Brown, Simon M. (2014) Phobos - The design and implementation of embedded software for a low cost radar warning receiver. EngD thesis

http://theses.gla.ac.uk/7288/

Copyright and moral rights for this thesis are retained by the author

A copy can be downloaded for personal non-commercial research or study, without prior permission or charge

This thesis cannot be reproduced or quoted extensively from without first obtaining permission in writing from the Author

The content must not be changed in any way or sold commercially in any format or medium without the formal permission of the Author

When referring to this work, full bibliographic details including the author, title, awarding institution and date of the thesis must be given.

Glasgow Theses Service
http://theses.gla.ac.uk/
theses@gla.ac.uk

Phobos

The Design and Implementation of Embedded Software for a Low Cost Radar Warning Receiver

Simon Brown

December 2014

VOLUME ONE OF TWO

A portfolio thesis submitted to the Universities of

Edinburgh

Glasgow

Heriot-Watt and

Strathclyde

In fulfilment of the requirements for the degree of

Doctor of Engineering in System Level Integration

Institute for System Level Integration

Abstract

This portfolio thesis describes work undertaken by the author under the Engineering Doctorate program of the Institute for System Level Integration. It was carried out in conjunction with the sponsor company Teledyne Defence Limited.

A radar warning receiver is a device used to detect and identify the emissions of radars. They were originally developed during the Second World War and are found today on a variety of military platforms as part of the platform's defensive systems. Teledyne Defence has designed and built components and electronic subsystems for the defence industry since the 1970s. This thesis documents part of the work carried out to create Phobos, Teledyne Defence's first complete radar warning receiver.

Phobos was designed to be the first low cost radar warning receiver. This was made possible by the reuse of existing Teledyne Defence products, commercial off the shelf hardware and advanced UK government algorithms. The challenges of this integration are described and discussed, with detail given of the software architecture and the development of the embedded application. Performance of the embedded system as a whole is described and qualified within the context of a low cost system.

Acknowledgements

I would like to thank Adrian Metcalfe for his guidance and leadership during this project and his determination to bring Phobos to market.

I would also like to thank Paddy Forrest, James Irvine and all the staff at Teledyne and ISLI.

Contents

List of Figures

List of Tables

Nomenclature

802.16j	The multihop and relay subgroup of the IEEE 802.16 workgroup
AOA	Angle of Arrival
AOC	Association of Old Crows
COTS	Commercial Off The Shelf
DF	Direction Finding
DSEi	Defence Security and Equipment International - A biennial defence and security trade show held in London
DSP	Digital Signal Processor
Dstl	Defence Science and Technology Laboratory
EEPROM	Electrically Erasable Programmable Read-Only Memory
ELINT	Electronic Intelligence
EngD	Engineering Doctorate
ESM	Electronic Support Measures
EW	Electronic Warfare
FIFO	First In First Out, a buffer that preserves the order of the received data
FPGA	Field Programmable Gate Array, a run time reconfigurable device used to implement digital logic circuits
GUI	Graphical User Interface

IFM	Instantaneous Frequency Measurement receiver
ISLI	Institute for System Level Integration
JED	Journal of Electronic Defense
MoD	Ministry of Defence
MTI	Moving Target Indicator
PCB	Printed Circuit Board
PDW	Pulse Descriptor Word
PRF	Pulse Repetition Frequency
PRI	Pulse Repetition Interval
RAF	Royal Air Force
rms	Root Mean Square
RN	Royal Navy
RWR	Radar Warning Reciever
sensitivity	The weakest input signal that a receiver can detect or operate correctly with
SSD	Solid State Drive
SWaP	Size, Weight and Power
TDL	Teledyne Defence Limited
TOA	Time Of Arrival
UAV	Unmanned Aerial Vehicle
UDP	User Datagram Protocol
WiMAX	Worldwide Interoperability for Microwave Access - The commercial name for all of the IEEE 802.16 standards

For the Chief

1 Executive Summary

Phobos is a novel radar sensor that is being developed by Teledyne Defence Limited. It listens for the emissions of radars that may be present and can identify the emitter of the pulses received by comparing the pulses against a library of known radar emission patterns. It is designed to perform the role of a radar warning receiver or to be used as an electronic intelligence gathering device. A radar warning receiver is a device that only warns the operator about radars that are present and are known to be found on platforms that are considered to be threats. The role of an intelligence gathering device is to gather as much data as possible about all the radars that are present, whatever their host platform may be.

Phobos is novel because instead of offering increased performance compared to existing products, it offers a significant reduction in size, weight and power and is available at a low cost.

Historically radar warning receivers and similar electronic intelligence gathering devices have been fitted to platforms that are large or expensive, or both. This is because existing systems are not low cost, so can only be justified when used to protect high value platforms or as part of a dedicated electronic intelligence capability. They commonly make use of custom pulse processors which require large amounts of power and cooling. Such systems have been available for many decades and their roles, performance, and size, weight and power (SWaP) requirements are well established. By making a smaller, lighter and cooler low power system available at low cost, Teledyne is hoping to challenge the status quo in the market and change the expectations of their customers as to how and when such a system could be used. Teledyne hopes to widen the market for such systems to include platforms such as armoured vehicles, inshore patrol boats and unmanned aerial vehicles (UAVs). At the same time Teledyne aims to diversify beyond military customers and traditional military platforms to include applications such as border monitoring and fighting piracy.

Teledyne has achieved these reductions in size, weight and power whilst at the same time reducing cost by taking advantage of advances in commercial processing technology and through using new radar identification software licensed from the UK Ministry of Defence (MoD). The commercial success in recent years of smart phones and tablet computers has driven great improvements in both the computational power and power efficiency of commercial microprocessors. By using commercial off the shelf (COTS) processing boards based around these new microprocessors in place of custom pulse processors, Teledyne can take full advantage of both the technological advances and the economies of scale. The radar identification software that Teledyne has licensed from the MoD complements this approach by offering high performance and minimal resource usage, making it suitable for a modern mobile platform.

Phobos is supplied as two parts: the sensor unit and a ruggedised laptop computer which are linked together by a network cable. The laptop computer runs the user interface application which receives data from the sensor unit, showing the results on screen and allows the operator to control the sensor unit. The sensor unit consists of antennas, a RR017 pulse characteriser, a single board computer, a digital compass and a GPS receiver. Radar pulses are received by the antennas, measured by the pulse characteriser and compared against the library of known radar emission patterns by the single board computer. The single board computer then sends the results along with the current location and heading across the network link to the laptop computer.

This portfolio thesis documents the work carried out by the author as part of his Engineering Doctorate (EngD) project in helping Teledyne to bring Phobos to market. The author's principal contributions to Phobos were writing almost every line of the software that controls the sensor unit half of Phobos and devising the algorithm that calculates the angle of arrival of incident radar pulses. The project is documented from the very start up until and including the final prototype. At the end of this period we were able to demonstrate that the product worked, that it could identify complex emitters in a realistic environment and provide the operator with that information correctly. Problems were found with the case design, data acquisition times and data integrity, which although did not prohibit system operation did limit system performance, making the performance targets set unreachable. Now that these problems have been identified, Teledyne can set about addressing them for the production models. Phobos is a notable product for Teledyne as it is the first time that they have marketed a complete system of their own to end customers rather than components or subsystems to other defence companies. The prototype described in this document is hoped to be the first in the Phobos family of products.

2 Portfolio Organisation

This portfolio thesis is divided up into two volumes: Volume One provides an overview of the product and what was achieved, whilst Volume Two documents the work done by the author to create the product and contains the papers published and reports written during the course of the research project.

2.1 Volume One

Chapter Four describes the history of radar warning receivers, what one is and why someone might want to buy one. Products that are currently available that could be considered competitors to Phobos are discussed along with why there is a gap in the market for a product such as Phobos and why Teledyne is well placed to exploit that opportunity. Chapter Five documents the history of the EngD project, how a project to develop a novel high speed data link for a radar sensor became the system integration work for the Phobos sensor unit. Chapter Six describes how the Phobos sensor unit operates today in detail. This includes all operational aspects of the sensor unit, both hardware and software, and the facilities that exist to aid development. Chapters Seven, Eight and Nine summarise the results, detail the conclusions and list the opportunities for future improvement of the sensor unit.

2.2 Volume Two

Volume Two starts with an introduction, Chapter Ten, that describes the context in which the component documents of Volume Two were written. Chapter Eleven documents the work done to develop a method of calculating the angle of arrival of incident pulses. Chapter Twelve documents the results of the first attempt to measure the system performance in detail and the consequences of that attempt. Chapter Thirteen contains the published papers that have resulted from this work. Chapters Fourteen and Fifteen contain reports written about conferences attended and other pieces of work relating to Phobos and software development within Teledyne. Chapter Sixteen is a personal reflection on the EngD project and Appendix A contains the PHOBOS-R product brochure.

3 Taught Modules

The required taught modules were taken in three phases. In the first phase 46 credits worth of technical modules were taken, studying on-site at the Institute for System Level Integration. In the second phase the remaining technical modules were studied via distance learning, which accrued a further 75 credits for a total of 121 credits from technical modules. The business modules were taken in the final phase, all through distance learning with weekend courses at the Edinburgh Business School. This provided a further 60 credits, resulting in a total of 181 credits.

3.1 Technical Modules

As the research project was expected to be concerned with the implementation of a wireless data link for a radar sensor network, the technical modules were chosen to be relevant to that area where possible. The chosen modules are shown in Table 3.1. DSP for Communications, Broadband and Digital Networks, and FPGAs for DSP were of particular relevance and the knowledge gained proved to be very useful during the investigation into the IEEE 802.16 wireless standards. Embedded Software 1 and Microprocessors and Microcontrollers, were also benficial as they were aligned with what the research project became. The concepts of reuse and testability studied in the silicon IP block related courses could also be applied to software development.

3.2 Business Modules

Of the business modules chosen, which are shown in Table 3.2, only Project Management was chosen for its expected direct relevance to the research project. The knowledge gained was used to better structure the research activity and project goals. Economics, although taken out of personal interest, proved to be very relevant with the research

Module	Credit Value
DSP for Communications	15
System Partitioning	15
Broadband and Digital Networks	8
FPGAs for DSP	8
Residential Total	46
IP Block Authoring	15
IP Block Integration	15
VLSI	15
Microcontrollers and Microprocessors	15
Embedded Software 1	15
Distance Learning Total	75

Table 3.1: Technical Modules

Module	Credit Value
Economics	20
Negotiation	20
Project Management	20

Table 3.2: Business Modules

project beginning at the start of the banking crisis. The defence industry is driven by government spending and the course explained the effects of and possible remedies for a sudden drop in aggregate demand or an economic shock. The negotiation course taught how to decide upon an initial position and how to defend it when entering into negotiations, which is directly relevant to launching a new product. The course also taught the difference between the high level aims of a negotiation and the means used to achieve them, and taught methods to discover them through dialog. This was beneficial when answering customer questions at product demonstrations as it was more important to understand why a feature was being enquired about than the questions about the feature itself.

4 The Need For Low Cost Radar Warning Receivers

4.1 Introduction

Although experiments into using radio waves to locate objects had been carried out decades before, it wasn't until the Second World War that effective radar systems were produced in large numbers. The possibility of either gaining a significant advantage in remote sensing over the enemy or having to suffer a significant disadvantage led to a rapid pace of development for both radar systems and their countermeasures in all belligerent countries. A key advance of the war was the development of the cavity magnetron by Randall and Boot [1] which allowed the construction of radar systems that operated at much higher frequencies than before. This allowed the size of the system to be greatly reduced as the higher frequencies required a much smaller antenna, which in turn allowed radars to be fitted to much smaller platforms. The higher frequency also improved the resolution of the system, allowing it to distinguish much smaller targets.

The Allied forces used B-24 Liberator bombers to protect convoys from German submarines during the Battle of the Atlantic. The Liberators were fitted with the ASV MkII 1.7 m wavelength radar which allowed them to locate a surfaced submarine at beyond visual range and in complete darkness. The German Navy quickly felt the disadvantage and deployed their first countermeasure, the FuMB-1 Metox. Metox was one of the first Radar Warning Receivers (RWRs) and could detect the emissions from the ASV MkII. As RWRs only have to detect the direct emission from the radar rather than the much weaker reflection from the target, they have a significant advantage in detection range over the radar itself. This would allow the submarine to submerge long before the Liberator could attack it. Metox could however only detect radars, not identify them, which could lead to false alarms when operating near friendly forces. The first systems had a Biscay Cross antenna which gave no information about the angle of arrival of the received radar pulse, so the crew didn't know where the threat was approaching from.

When the Liberators were upgraded with the ASV MkIII the Metox was rendered obsolete. The ASV MkIII was a magnetron based radar which allowed it to operate at centimetre wavelengths that the Metox was not able to detect. The Germans, although aware of the ineffectiveness of Metox, were baffled as to what had changed as they did not believe that centimetric radar was possible until they found one in a crashed Allied bomber[1]. In response they developed the FuMB-7 Naxos [2] which was tuned to the right wavelength and the submarine crews were once again warned of approaching aircraft.

The arms race between radar and RWR system developers has continued ever since with neither side gaining a decisive advantage for long. Since the war the uses of radar technology have proliferated far beyond the detection of ships and aircraft. Radars are now used as automobile reversing sensors, to monitor weather and to detect airport runway debris. RWRs today, with the exception of speed camera detectors, are largely restricted to the same roles as in the Second World War. They are found on naval vessels, military aircraft and used by specialist ground based Electronic Warfare (EW) units.

4.2 Technical Overview

This section provides a technical overview of what role each part of the system performs and what the constraints are on its performance. RWR systems can be modelled from a pulse processing perspective as containing the six components shown in Figure 4.1.

4.2.1 Antenna

The choice of antenna controls which pulses will be detected and whether their angle of arrival (AOA) can be determined. An RWR system is designed to have a certain probability of intercept for pulses of interest. Pulses of interest will have certain frequencies, durations, amplitudes and angles of incidence to the system. The set of pulses of interest is determined by the nature of the host platform and the purpose of the RWR, along with the range of emitters it would like to detect. The requirements of an intelligence gathering system are far greater than that of a simple threat warner and although both

[1] 3^{rd} February 1943 a British Stirling bomber crashed south of Rotterdam, near Hardinxveld-Giesendam. On board was a H2S 9 cm radar, serial number 6.[2]

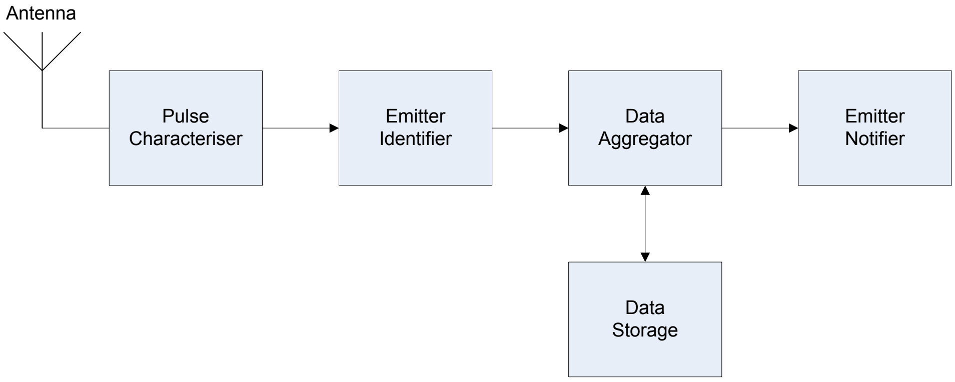


Figure 4.1: Generic Radar Warning Receiver System Diagram

types of system may be interested in the same emitter, the parameters of their pulses of interest can be quite different due to the differing level of detail required. The probability of intercept is also limited by the sensitivity of the Pulse Characteriser and the throughput of the Emitter Identifier, however the type, number and orientation of the antennas are the most significant factors.

If a RWR can detect a radar beyond the detection range of that radar, it can allow its host platform to avoid detection altogether. This is possible due to the range advantage of the interceptor. Consider an intercept station illuminated by the emissions of a radar. All of the incident radiation is available to the intercept station whereas the radar will only receive the fraction of that energy which is reflected back toward it, further reduced by the path loss of the return journey. This difference in available energy is the basis of the range advantage of the interceptor. Platforms designed to reflect back very little radar energy using stealth techniques may be able to exploit the range advantage to the extent that they can detect an emitter and engage it before being detected.

The intercept station also has some disadvantages with respect to the radar whose emissions it is trying to intercept. The radar station knows the approximate characteristics and direction of the reflected pulse which allows the use of a high gain directional antenna and a narrowband receiver which improves the sensitivity of the system. A practical intercept station will require a broadband antenna and receiver, as it will have been designed to intercept more than just one mode of one emitter. To overcome this, directional antennas can be used as even though the AOA of the pulse is not known in advance, if enough antennas are employed, all possible AOAs can be covered. Not only does this extend the range of the intercept station, but it also allows the AOA of the incident pulse to be determined. This can be done by comparing how each antenna receives the same pulse. Using multiple antennas, depending on their type and number, can greatly increase the system size and cost which may make them unsuitable for some platforms and roles.

As an alternative to using multiple antennas of the same type to benefit from antenna gain and the ability to calculate the AOA of the incident pulse, antennas of differing types can be used. A typical configuration would be an omnidirectional antenna to provide adequate probability of intercept and a rotating dish antenna to provide high gain and AOA information. This can be cheaper than using multiple antennas of the same type, but is more restrictive with regard to host platform, ruling out any platform that is either small or fast moving.

The choice of antenna is dependent upon the set of pulses of interest, the desired probability of intercept for those pulses at a given range, the need to determine the AOA through multiple reception, the size, quantity and type of antennas that the platform can accommodate and the financial constraints.

4.2.2 Pulse Characteriser

The Pulse Characteriser is the source of digital information about the received pulses, the bridge between the analogue waveform and the digital pulse descriptor word (PDW). Along with the antenna, the sensitivity of the characteriser limits the detection range of the system. The characteriser must also have enough ports and channels to support the chosen number of antennas and the desired probability of intercept.

Data is generated about every pulse received in as much detail as is required, but only at the scale of the individual pulse. The level of required detail is determined by the role of the system and the requirements of the emitter identifier. An intelligence gathering platform will wish to record everything with as much precision as possible. A pure threat warner will need nothing more than what is required by its emitter identifier. An overview of the type of data that a pulse characteriser might record is shown in Table 4.1.

4.2.3 Emitter Identifier

The role of the Emitter Identifier is to analyse the individual PDWs received from the Pulse Characteriser and attribute them to radars or radar types found in the emitter

Pulse Width	Width of the received pulse. Required accuracy determined by emitter identifier and role.
Time of Arrival	System time assigned to the pulse. May be absolute or in system ticks.
Amplitude	Peak pulse amplitude. May include a detailed amplitude profile of the entire pulse.
Angle of Arrival	Calculated AOA of the pulse normally in the azimuth plane. May also include elevation.
Frequency	The carrier frequency of the pulse.
Pulse on Pulse	When pulses overlap there is a rapid increase in amplitude. Normally invalidates any pulse width and amplitude measurements.
Modulation	Indicates whether the carrier frequency of the pulse was modulated over its duration in phase or frequency.
Bad Data	The characteriser was unable in some way to correctly measure the pulse. This flag indicates which values are known to be unreliable.

Table 4.1: Pulse Characteristics

library. This is traditionally a two stage process where the pulses are first grouped into pulse trains that are believed to have come from the same emitter and then the pulse train patterns are compared against the library of known patterns. The first part of this section explains the constraints on useful radar waveforms that cause them to have recognisable patterns. The latter part discusses how a stream of received pulses could be practically matched against the pattern library.

4.2.3.1 Constraints on Useful Radar Waveforms

The use of modern signal generation techniques could allow military radars to emit a very agile signal, which would make them very difficult for an intercept station to identify. Fortunately for the intercept station, a very agile signal is both difficult to generate and unsuitable for use as a radar waveform. Radar systems have a peak output power in the range of tens to hundreds of kilowatts and above. Frequency agility is limited by the ability of the system to generate a high power signal efficiently at multiple frequencies. Using multiple frequencies is advantageous as lower frequencies have a longer detection range and higher frequencies are able to provide more accurate target bearings and target resolution. Each extra frequency comes at a cost as the system needs either an extra transmitter and receiver to preserve the transmit power and receiver sensitivity, or the system has to accept the performance loss of using broadband transmitter and receiver. The host also has to accommodate the size, power and cooling requirements of each additional frequency.

Variation in pulse duration is constrained not only by how rapidly the emitter can be turned on and off, but also by the fact that the generated waveform must be able to detect targets. A useful waveform is able to determine the range of a target and its radial velocity for all targets of interest in the detection range of the radar. To calculate a target's range, the system must be able to attribute a received return to an outgoing pulse in order to know the time of flight of the pulse. If the system has multiple pulses in flight at the same time, unless it is able to attribute returns it will not be able to differentiate between a return from a near target due to the most recent pulse and a return from a distant target due to an earlier pulse. As the effective radar cross section of the targets will not be known in advance, the maximum detection range of the radar will be unknown, making it difficult to restrict the system to one pulse in flight, by reducing the radar's output power. As a consequence the radar may operate with a maximum detection range that is greater than its unambiguous range. As described in Chapter Twelve of "Introduction to Airborne Radar" by Stimson [3], the unambiguous range R_u can be expressed as

$$R_u = \frac{c(PRI)}{2}$$

where c is the speed of light and PRI is the pulse repetition interval. Range ambiguity can be resolved without changing the output power by changing the PRI a small amount. At the new PRI, if the target is within the unambiguous range the apparent range will not change. If it is beyond the unambiguous range, the change in apparent range will be proportional to how many pulses ago, shown as n, the returned pulse was emitted. This is equivalent to multiples of the R_u of the original PRI in range.

$$n = \frac{\Delta R_{apparent}}{\Delta R_u}$$

The true range is then

$$R_{true} = nR_u + R_{apparent}$$

Adding one extra PRI does not extend the unambiguous range to infinity however as there will be a distance at which integer multiples of the two respective unambiguous ranges align. As an example, if $R_{u2} = 1.1R_{u1}$ then the new ambiguous range will be $R_u^{'} = 10R_{u2} = 11R_{u1}$. The unambiguous range can then be extended as required by adding further PRIs. The ambiguity may return however in the presence of multiple targets beyond ambiguous range on the same bearing. In the presence of two targets when the radar changes from one PRI to the next, the apparent range will change and there will be two sets of range pairs that could have caused the observed change. One pair will be the actual ranges, whilst the other pair are only *Ghosts*, but without further information it cannot be determined which is which. This arises from the fact that the true ranges of the targets may be different multiples of the unambiguous range of that PRI. So when the PRI changes the new apparent ranges may display the targets in a different order. A worked example from Stimson p.158 [3] is shown in Table 4.2.

PRI	First Target $R_{apparent}$	Second Target $R_{apparent}$	ΔR_u to PRI 1
1	6	6.5	0
2	5.5	6	$+\frac{1}{4}$
3	6.5	7	$-\frac{1}{4}$

PRI Change	$\Delta R_{apparent}$		$\frac{\Delta R_{apparent}}{\Delta R_u}$		True Range	
	Same	Swapped	Same	Swapped	Same	Swapped
1 → 2	-0.5, -0.5	0, -1	-2, -2	0, -4	26, 26.5	6, 46.5
1 → 3	0.5, 0.5	1, 0	-2, -2	-4, 0	26, 26.5	46, 6.5
2 → 3	1, 1	1.5, 0.5	-2, -2	-3, -1	26, 26.5	36, 16.5

Table 4.2: Radar Ghosts

In the example, the first PRI was chosen to give an unambiguous range of 10 nautical miles and the other two PRIs to increase or reduce the unambiguous range by a quarter of a nautical mile respectively. At any of the PRI changes, the targets may then appear in the same order or swapped. The lower half of the table shows the calculated true range indicated at each transition. The true range of the same order pair is constant at 26 and 26.5 nautical miles whereas the swapped range pair varies, which identifies it as the ghost pair.

Instead of using multiple PRIs to attribute pulses, the pulses can be keyed by increasing the carrier frequency of the pulses in a known manner over many pulses in the pulse train out to a distance well beyond the maximum detection range. By measuring the frequency of the reflected pulse, its time of emission can be looked up against the transmitted pattern and the range of the target calculated. This allows a single pattern to identify any number of static targets. However if either the host platform or any of the targets are not static the problem of ghosting occurs just as with PRI switching. In this case the reflected pulses will have a Doppler shift and without multiple measurements it may be ambiguous as to which target has undergone which shift. With linear modulation, one measurement would be made on each segment and the system would need N segments to resolve $N - 1$ targets. This technique, known as FM Ranging, is therefore suited to applications such as altimeters rather than air-to-air applications where PRI switching performs better [3].

The detection range of a radar is limited by the energy reflected by the targets. The detection range can therefore be increased by both emitting and receiving more energy, however both are problematic. The emitted energy can be increased by simply increasing the pulse duration. The range resolution of a radar is however inversely proportional to the pulse width. For closely spaced targets such as aircraft flying in formation, as the pulse width increases there will come a point where the returns from the two aircraft start to overlap and the radar is then no longer able to resolve the two targets. This problem can be reduced by using pulse compression, a technique commonly known as 'Chirp'. A pulse can be compressed by increasing its carrier frequency over its duration at transmission. Then the received reflected pulse is passed through a frequency dependent delay line which delays the start of the pulse with respect to its end. Very high compression ratios can be used with only the slight disadvantage of increased Doppler sensitivity and increased sidelobes. To receive more energy, the receiver can integrate over the received pulses and use the aggregate output. If the system and environment allow for coherent integration then the received pulses will combine constructively in the receiver, while random noise will combine destructively. Non-coherent integration is also possible but produces inferior results. The improvement in signal to noise ratio that coherent integration can achieve is dependent upon the number of pulses received or integration period. The maximum integration period will however often be limited by other system criteria. The integration time is only valid for one mode of operation, so whenever the system switches to a new PRI or FM Ranging segment, a new integration period begins. So the more modes the radar has, the shorter the integration time. The integration period also determines the bandwidth for the Doppler filters in the Doppler filter bank, with the result that the longer the integration period, the narrower the filter bandwidth and the greater the frequency resolution.

Measurement of the Doppler shift of a received pulse can be used to determine the radial velocity of the target; which is the rate at which it is approaching or receding from the radar. As well as providing velocity information about the target, Doppler shift can be used to separate targets of interest from radar clutter. Radar clutter is the name for the pulse energy reflected back from the terrain and atmosphere in the radar's detection range. This is obviously a signal of interest for a radar altimeter or a weather radar, but of no interest at all to an airport surveillance radar. Thankfully for the airport surveillance radar, the clutter will be stationary or slow-moving and therefore the returned pulses will have a low Doppler shift. For mobile radars the situation is more complicated as the Doppler shift of the clutter will be variable and the range of expected Doppler shifts will be greater as the target could be approaching or receding. The range of acceptable Doppler shifts is limited by the pulse repetition frequency (PRF) of the mode the radar is in. If the Doppler passband is greater than the PRF then components from the next spectral line of the carrier will be included, duplicating targets. A high PRF radar can support a large Doppler passband which will allow targets with a large radial velocity to be easily differentiated from the clutter. It will however have a short unambiguous range. A low PRF radar will have a large unambiguous detection range but a very narrow range of unambiguous radial velocities. Such a radar is often call a Moving Target Indicator (MTI) as it would not be expected to be able to resolve the velocity ambiguity to give more information. Even the MTI functionality will not be reliable as a stationary radar will have radial velocities with a Doppler shift equal to the PRF which will alias to zero and be indistinguishable from clutter. These velocities are called blind speeds. For a moving radar, the clutter may be aliased across the entire Doppler passband making targets of interest hard to isolate from the clutter. A summary of the advantages and disadvantages of the various PRFs, taken from Skolnik [4], is given in Table 4.3.

This shows that whilst it may be possible to generate an arbitrary waveform, useful radar waveforms must be tailored to the targets that they wish to detect. The radar system parameters involved are often interrelated making it difficult to change one aspect of system performance without adversely affecting others. Although it may be possible for radars to make use of broadband, single use noise like waveforms in the future, the majority of systems in service today were designed against these constraints. It is these radars that Phobos was designed to detect. Further information on the constraints on radar waveforms can be found in "ELINT The interception and Analysis of Radar Signals" by Wiley, Chapter Two, Section Two [5]. In addition "Introduction to Airborne Radar" by Stimson is a very accessible introduction for non-specialists [3].

PRF	Advantages	Disadvantages
Low	Can sort clutter from targets on basis of range. No range ghosts. Front-end STC suppresses sidelobe detections and reduces dynamic range requirements	Low Doppler visibility due to multiple blind speeds. Poor slow-moving target rejection. Cannot measure radial target velocity.
Medium	Good performance at all target aspects. Good slow-moving target rejection. Measures radial velocity. Less range eclipsing than in High PRF.	Range ghosts. Sidelobe clutter limits performance. High stability requirements due to range folding.
High	Can be sidelobe clutter-free for some target aspects. Single Doppler blind zone at zero velocity. Good slow-moving target rejection. Measures radial velocity. Velocity-only detection can improve detection range.	Sidelobe clutter limits performance. Range eclipsing. Range ghosts. High stability requirements due to range folding.

Table 4.3: Advantages and Disadvantages of High, Medium and Low PRF Operation

4.2.3.2 Emitter Identification

At this point in the system, the received radar pulses are represented by digital pulse descriptor words in place of analogue signals. The data that is included in each PDW will be chosen according to the needs of the operator and the emitter identification software. Emitter identification techniques can be divided into single pulse and multiple pulse techniques. Single pulse techniques rely upon a very detailed characterisation of each pulse so that any quirks, that the transmit chain of the emitter might have, may be noticed. For example, if the amplitude envelope profile is very detailed it may reveal ringing at the start and end of the pulse. Knowing the extent of the ringing along with other pulse parameters such as frequency and pulse width may be enough to reveal the radar type. The more detailed the characterisation, the further this technique can be taken. Once the radar type has been identified there may be enough variation between models of the same type to identify a specific instance of that radar type. This obviously requires the interceptor to have detailed library entries for each known instance of that radar type which makes it only suitable for low volume platforms such as ships. More details can be found in D'Agostino *et al.* [6].

Multiple pulse techniques do not require the same level of characterisation detail as single pulse techniques, but do bring the extra burden of deinterleaving the pulse trains. When the interceptor is in the presence of multiple emitters, the pulses from each emitter will be received in a potentially overlapping and jumbled manner. Before the pulse train of each emitter can be compared against the library of known emitters, the pulse trains first have to be reconstructed. The received pulses are interleaved and the process of reconstruction is called deinterleaving. It is done by using the above described restrictions that radar signals have and other physical restrictions that can be deduced. Two suitable criteria are frequency and angle of arrival. Pulses that are similar in frequency are more likely to have come from the same emitter than those that are dissimilar and pulses that have a similar AOA are more likely to have come from the same emitter than those that are dissimilar. An emitter that changes frequency from pulse to pulse that is mounted on a fast moving platform may break both assumptions, but the interceptor should know how likely an encounter with such an emitter is. Other pulse parameters can be used such as pulse width, but they are more vulnerable to multipath effects, effects due to the scan pattern of the emitter and pulse on pulse effects.

Once the pulse trains have been deinterleaved they can then be compared to the known emitter pulse patterns, however this comparison is not simple. The received pulse train must be synchronised in time with the library entry as the received pulse train may be only part of or many repetitions of the pattern in the library. The match will not be always perfect, pulses may be missing and extra pulses may have been included through deinterleaving error. This leads to a large processing burden as each deinterleaved pulse train may have to be compared against many library entries many times with different time shifts and after which the process may have to be repeated for incorrectly deinterleaved pulses. This is for a pulse density of over a hundred thousand pulses per second in real time. Consequently many papers can be found in the open literature on the themes of assessing and improving pulse throughput for differing deinterleaving techniques [7, 8, 9, 10, 11].

Instead of deinterleaving using single pulses, radars can also be deinterleaved by looking for patterns in larger quantities of data. Radar emission patterns can be observed by analysing the time difference of arrival patterns of the received pulses. The pulses can be clustered using coarse pulse parameters as an initial filter and the radars then reveal themselves through changes in the time difference of arrival of their pulses when they change between their modes. Once deinterleaved, the patterns can then be compared to the known patterns. This technique by its use of time difference avoids the problem of synchronisation as the absolute time of arrival (TOA) is no longer relevant. Hassan proposes a new technique of combined deinterleaving and recognition which also avoids the problem of synchronisation [12]. The coarsely filtered pulses are correlated against the emitter library and in doing so are deinterleaved and identified in a single step. He suggests that the technique would be well suited to reducing the number of false positives generated by the deinterleaver, thereby reducing the load on the threat identifier.

Due to their sensitive nature few actual RWR systems have been described in the open literature. One example is the Cutlass system described by Rogers in 1985 [7]. Variants of this system are still in service with the Bangladeshi and Turkish Navy despite its age, which demonstrates the long life of military systems [13, 14]. Further details on the Cutlass family of systems can be found in "The Naval Institute Guide to World Naval Weapons Systems, 1997-1998" by Friedman [15].

4.2.4 Data Aggregator and Storage

In the previous stages of the system, the data being processed has changed from analogue waveforms to pulse descriptor words, to pulse trains and to identified pulse trains. It is at the data aggregator stage where all of this information is condensed into an operational picture where in place of pulse information, information about emitters is sent to the operator. For each emitter type detected the aggregator needs to extract all the remaining data from the pulses so that it can be passed to the emitter notifier. The primary concern is estimating the number of instances of each radar type. AOA information can be used to discriminate instances of emitters, but if they are close together, such as aircraft in formation, the system may not have sufficient angular resolution. The emitter identifier may be able to explicitly tag different instances of the same emitter simplifying this process.

Depending upon the role of the system, the aggregator may also be used to cluster unidentified pulses before they are sent to the emitter notifier. Pulses can be clustered using parameters such as AOA, frequency and pulse width. Statistical information about the pulses in the clusters such as minimum, maximum and average PRF can also be calculated. In the case of a remotely operated platform, it may also be appropriate to include information about the host platform such as current location and heading.

For many systems it is neither beneficial for the operator nor possible to send the raw pulse data beyond this point. It may however be useful if the data could be retained so that it could be analysed later when time and facilities are available. So it is at this point where the raw and processed pulse data can be saved for later analysis.

4.2.5 Emitter Notifier

The role of the Emitter Notifier is to present the operational picture constructed by the aggregator to the operator. This is where the difference between a RWR and an intelligence gathering system become clear. The needs of a pilot in a fast jet are very different to those of a border agent receiving data from a remote border monitoring device. In the case of a warship there may be no human operator at all as the system may be simply directly connected to the defensive systems. Emitter Notifiers can be regarded as fulfilling two roles: that of threat warning and electronic intelligence. Actual systems may perform one or both of them.

4.2.5.1 Threat Warning

In a threat warning role the system is only concerned with known threats and its output need only include the level of detail that can be exploited by the operator. The system must be able to rank threats so that the operator can make best use of the information. For a fast jet this may mean a combination of audible and visual alarms for a limited number of targets, with a coarse indication as to where the threats are. A threat warning system on a ship might be used to guide the ship's defensive systems, therefore it will be concerned with all concurrent targets and will provide the most detailed location information it can.

4.2.5.2 Electronic Intelligence

In this role all of the received pulses are of interest and all of the information gleaned will be made available to the operator. This allows the operator to not only build a detailed operational picture, but to identify and analyse unknown emitters through the use of other sensor data. This may require the ability to access and visualise both the raw and the processed pulse data along with the emitter information. The operator must be able to add to and edit emitter libraries and then reprocess previous data. In contrast to the threat warning role, the data is often analysed not in real time but after the event. These systems are called Electronic Support Measures or ESM systems.

4.3 Market Overview

Since the Second World War, the market for RWR and ESM systems has grown and the systems have become much more reliable and capable. The military remains the dominant customer and product development is driven according to their needs. In the British armed forces, the Army only has a modest radar EW capability, which is a small part of 14 Royal Signals based at Cawdor Barracks. The Army only buys RWR systems for its helicopters such as the HIDAS [26] system fitted to the AH-1 Apache. Consequently most RWR and ESM system development work is done for the Royal Navy (RN) and the Royal Air Force (RAF). The RAF buys ESM systems for its specialist EW platforms such as Nimrod and the recently acquired RC-135W Rivet Joint aircraft [27]. RWR systems are fitted to fast jet aircraft such as Tornado and Eurofighter. The RN buys integrated systems that perform both EW and defensive roles such as the UAT system [28] found on the Type 45 Destroyer and the UAP (4) [29] system found on Astute class submarines.

The capabilities of these systems are ever improving. Frequency ranges have widened, minimum pulse widths have narrowed, pulse density ceilings have risen and contemporary systems can accommodate an ever increasing number of concurrent emitters. Table 4.4 shows some of the system specifications for a selection of current systems. The information has been taken from publicly available data sheets and a technology survey published in the Journal of Electronic Defense (JED)[21]. Where there are gaps in the table the information was not available.

The JED survey divides the market into four sectors: ground, surface, submarine and air, and there is a wide variety of systems available for those markets. The lightest system is the AQ211 (V)10 RWR from ITT which can be configured at 16 lb and is designed for an airborne platform. The heaviest system is the MRSR-800/MRGR-800 ESM from Indra which can be configured at 195 kg and is designed for naval platforms. The survey shows that whilst weight is a consideration for current airborne platforms, it is not a critical one. The heaviest airborne system is the HES-21 from Saab which is 100 kg, which can not be regarded as light weight. The heavier systems are also not just fitted to transport aircraft, the ALR-67 (V) 3 from Raytheon which weighs 79 lb is fitted to the F-18 fast jet. Power requirements show no obvious link to the host platform type. The lowest power

Manufacturer **System**	**Frequency** **DF Accuracy**	**Pulse Density** **Reaction Time**	**Power Weight** **Size**	**Platform** **Notes and Reference**
Northrop Grumman AN/ALR-93(V)1	C/D, E-J Band 15° rms (E-J Bands)		198 W 27.2 kg	[16]
Rockwell Collins CS-3600	2-18 GHz 6° rms	2.5+ Mpps <1 s	<400 watts 75 lbs 2U chassis	Small Without Antennas [17]
SELEX Galileo Sky Guardian 2500	C-K Band Better than 10° rms		224 W 23kg	air [18]
SELEX Galileo SEER	E-J Band <10° rms			air [19]
Thales CATS	E-K Band better than 10°		144 W 10 kg 194 x 157 x 318mm	air Processing unit only [20, 21]
Thales Vigle Lightweight	2-18 GHz 6° rms	1 Mpps < 1s	20 kg + Antennas	sea [22, 21]
Indra ALR-400	0.5-42 GHz 4 GHz BW Amplitude Monopulse		>200W 10.2 kg	air [23]
Argon ST WBR-2000	2-18 GHz present unspecified		<350 W <100 lbs 6U chassis	ground and sea Amplitude and phase DF [21]
Saab U-SME-200	2-18 GHz 2° rms	2.5 Mpps	350 W 45 kg 16 x 17 x 10.5 inch	sea [24, 21]
Elettronica ELT/160 family	E-J Band Amplitude Monopulse		15-20 kg 1 ATR	air Single Channel Recording [21]
Teledyne Defence Phobos	2-18 GHz 10° rms	1+ Mpps < 1s	24 W 7.2 kg 320 x 320 x 105mm	Multiple Complete System [25]

Table 4.4: Functional Specification of Competing Products

device, the 44 W ALR-69A made by Raytheon is designed for airborne platforms, whilst the 5 kW AN/ALQ 178 V(3) from MiKES is also designed for airborne platforms.

Despite the differences in weight and power, there is a largely common set of core features. The core operating frequency range is 2 - 18 GHz or E - J band with many vendors offering optional extensions down to 0.5 GHz (C band) and up to 40 GHz (K band). Instantaneous frequency measurement receivers (IFMs) are used as wide open receivers to provide high probability of intercept across the frequency range with super heterodyne or digital receivers used for channelisation. Three Direction Finding (DF) methods are used: amplitude comparison monopulse with 4 - 8 antennas, phase or time comparison and rotating antennas. The airborne systems are supplied in custom housings for that platform, whilst ground and naval systems are generally supplied in rack mountable chassis.

Amongst all of these details two things are notably missing: price and capability. This makes it difficult to assess the relative merits of the systems. The utility of the system can be thought of as a function of the value of the information it can provide, the cost of the system and its reliability. The cost of the system is not only the initial cost and maintenance costs of the system, but also the opportunity cost of budget and sensor bay space that *could* have been used for a different system. The value of the information received can be thought of in terms of the disadvantage of failing to receive it. In the case of an intelligence gathering application measuring the value of something that was not received is difficult as it may never have been there. In the case of platform protection the calculation is simpler. It is the replacement cost of the platform and crew, the cost of the platform not being available until it can be replaced and any secondary costs that occur as a result of the platform being lost. This is however tempered by the fact that the platform will have more than one protection system and in the event of the platform being lost, the responsibility will be divided across the systems that were not able to neutralise the threat.

A RWR not only has to compete against all the other RWR systems, it must also compete against all the other platform protection systems for part of the platform protection budget. In the case of signals intelligence gathering there are similar concerns as ESM systems must compete against other types of electronic intelligence (ELINT) systems and other means of intelligence gathering. As the size of the platform protection budget increases with the cost of the platform this has meant that RWRs have only been fitted to high value platforms such as fast jets, large warships and helicopters.

Despite the modest size of the British Army's EW capability there are still plenty of opportunities for RWR systems on ground based platforms. Armoured vehicles of all types are vulnerable to air power and may be able to benefit from knowing about threats they can't yet see or hear. They are not the only platforms where to date a RWR system has been unaffordable, impractical or both. Inshore patrol boats and commercial shipping vessels also do not routinely have RWR systems fitted. For commercial vessels it was not thought that a RWR would be beneficial, however the increase in piracy is changing that view, especially in waters near Somalia.

The market for RWR and ESM systems is also changing for airborne platforms. The unmanned aerial vehicle has turned parts of the military aircraft market upside down. Instead of ever more capable and expensive aircraft which are bought in ever fewer numbers, UAVs are comparatively low cost, low performance and plentiful. They are also not routinely fitted with a RWR system for cost reasons. This increase in the size of the military aircraft market raises an interesting possibility. If all UAVs were to be fitted with a RWR or an ESM system and an appropriate data link, would such a distributed radar sensor network be able to offer greater performance than specialist EW platforms through its spatial diversity? The challenge would be to produce a system that offers sufficient performance to allow the exploitation of the sensor network, at a cost suited to the platform protection budget of an UAV.

The RWR and ESM system market is still dominated by the traditional platforms of fast jet and large warship. The advances in commercial processor technology made in the past decades now allow a system to offer acceptable performance without the cost of previous systems and with greatly reduced SWaP requirements. This may lead to RWRs being fitted to other platforms such as armoured vehicles and inshore patrol craft that previously weren't valuable enough to merit an RWR system. As the product continues to move from being custom made to a commodity, completely new markets may appear such as border monitoring and commercial shipping. In the future RWR systems may become so common on the battlefield that, with an appropriate data link, they could form a radar sensor network. Such a network may be able, through its spatial diversity, to outperform specialist ESM systems at ELINT data collection as well as providing platform protection.

4.4 The Teledyne Advantage

As the RWR moves from being a bespoke product toward becoming a commodity product, the competitive landscape will change bringing both advantages and disadvantages for both the established system vendors and any new entrants to the market. The established vendors have invested heavily in their existing product ranges and will want to maximise the return on that investment. They will also be reluctant to cannibalise sales of those products by bringing out lower cost products too soon.

New entrants have no existing products to endanger so are positioned to be able to push further and faster with cost reduction through the use of commercial off the shelf components. This could lead to a significant competitive advantage if they can disrupt the market for existing products before the established vendors have a low cost product available. Their main challenges will be credibility and customer access. A new entrant's product will not be in service with any military and will not be based upon an existing product, it will never have been to war. This means that it is vulnerable to being regarded as unproven and therefore a high risk purchase. Potential purchasers can only buy systems that they are aware of. When a military decides upon the requirements for its next generation of RWR systems, the requirements will be guided in part by what they believe system vendors can actually deliver within the budget available. New entrants with a disruptive product will first have to make contact with various purchasing bodies within the target militaries and demonstrate their product and show how it is different and better than existing products. Only then is the purchasing body in a position to be able to ask for such a device as their next RWR system. This is easy for the existing vendors, they will already know the purchasing teams well through past projects and they will know who to demonstrate a new product to and when the opportunities are.

In part though these weaknesses are also a strength for the new entrant. The fact that their product is so different to the existing ones makes direct comparison difficult, especially so if the differences are very visible. The most visible differences of the system are its cost, size and user interface, the things a prospective purchaser would best appreciate through a demonstration rather than read on a specification sheet.

The only custom electronics still required by a RWR or an ESM system is the pulse characteriser. These devices in both analogue and digital form are part of Teledyne's core product offering. Teledyne has been supplying such devices to the larger defence companies for integration into their products for many years. This allows Teledyne to create a product that is composed of internal and commercial components, whereas the larger defence companies who have positioned themselves as system integrators will still be dependent upon companies like Teledyne for this custom part.

Another important consideration for Teledyne is that Phobos will only be one part of Teledyne's product range and this may lead to conflict as the customers of their other products may be their competitors in this market. The company's stated aim is to 'move up the food chain' of defence companies and this may allow the company to change its relationship with its customers, becoming more of a collaborator rather than a nameless subcontractor[2].

4.5 Contributions

With reference to Figure 4.1 the author's contributions to the Phobos system are as follows. The antennas are off the shelf commercial parts. The author collected the data for the performance analysis of the lids described in detail in Chapter Eleven with the help of others and was solely responsible for the analysis. The author designed the algorithm and wrote the implementation of the software that calculates the angle of arrival of radar pulses that is associated with the antennas. The Pulse Characteriser is a Teledyne product. The author worked with others to write the firmware that configures the USB interface of the Pulse Characteriser and was solely responsible for the software that collects and decodes the pulse data from the Pulse Characteriser. The Emitter Identifier is a software library licensed from DSTL. The author was solely responsible for the software that prepares the input data for the Emitter Identifier and the software that interprets the output. The Data Aggregator is software that was solely written by the author. It composes the output data in a form that is suitable for the Emitter Notifier and runs a server that the Emitter Notifier can connect to, to collect the data. Data storage is an abstract part of the system that represents the ability of the system to save pulse data with the option of replaying the data later. The author was solely responsible for the software that selects and saves the desired pulse data to disk and the code that allows pulse data files to be used as system input. In the case of Phobos all of the parts of the system mentioned so far are contained within the sensor unit, with the Emitter Notifier being a software application that runs on a separate computer. This is explained in chapter 6. The author's only contribution to the Emitter Notifier is the client software that connects to the Data Aggregator, collects the output data and sends commands to

[2] Stated by Keith Ferguson, Managing Director of TDL, in many internal company presentations.

the sensor unit.

5 Project History

5.1 Next Generation Data Link

When Teledyne Defence entered into an agreement with the Institute for System Level Integration (ISLI) to sponsor an EngD student in 2006 the company was in a different position to that which it found itself in at the end of the project four years later. At the start, the company was part of the Filtronic group and traded as Filtronic Components Limited (FCL). Filtronic Components was the original company of the Filtronic group started by Professor Rhodes from Leeds University in 1977. FCL was founded to supply microwave components to specialist markets, but diversified and grew rapidly in the 1990s, supplying components for mobile communications handsets and base stations. The Dot Com crash and increasing commoditisation of those markets led the company to divest itself of both its handset business in 2005 and its wireless infrastructure business in 2006. Very quickly the company went from being a global business with thousands of employees to being very UK focused with only a few hundred staff.

During this time FCL was also trying to reposition itself. The large European and American defence contractors relied upon companies such as FCL for specialist components and subsystems and as such were also able to demand ever lower prices. To try and maintain profit margins the company decided to “move up the food chain” by producing larger, more complex and more profitable systems. This would allow the company to bid for large contracts as part of a consortium rather than just being used as a subcontractor, changing significantly the negotiating position of the company. As part of this strategic change, the company renamed itself Filtronic Defence Limited (FDL) in 2007.

FDL’s technology road map for 2007 included three development activities which were intended to produce one of the new more complex products, diminutively referred to as the ’Sensor in a Tin.’ FDL had a long history of producing high performance IFMs and the first activity was to produce a new analogue IFM, the RR017. This was to be a disruptive product because rather than higher performance, the objective was instead

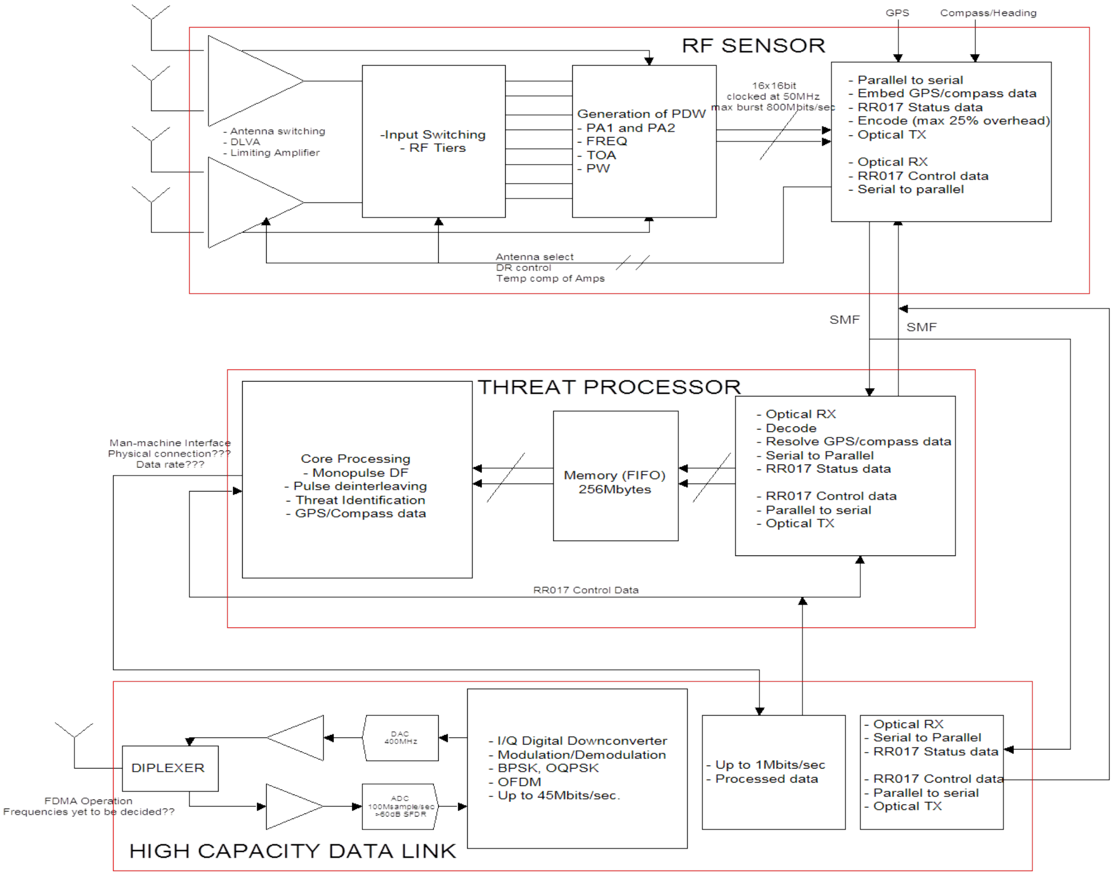


Figure 5.1: Sensor Data Link

to provide acceptable performance with much lower size, weight, power and cost. The second activity was 'Threat Processing', the development of appropriate software to turn the RR017 into a low cost RWR. The final activity was the 'Data Link', which was to allow the threat data collected by the low cost RWR and other data collected by the 'Sensor in a Tin' to be conveyed by an appropriate wireless network back to where it was needed.

During the restructuring of Filtronic, staff had moved between the groups and FDL now had staff from Filtronic's former Global Technology Group who were familiar with parts of various mobile communications standards and could advise on what might be possible with a modern data link. On that basis, they had sketched out what they thought was possible. With this design they then sought to engage with outside experts and academia to complete the design. This design is shown in Figure 5.1, image created by Mustafa Akkul, copyright Teledyne Defence Ltd. The EngD project was set up with ISLI to give FDL direct access not only to the latest advances in telecommunications research that academia could offer, but also the wealth of experience that ISLI's academic institutions had in designing novel wireless links. As well as ISLI, FDL also engaged with the more local Wireless Centre of Industrial Collaboration (Wireless CIC). The Wireless CIC was a joint venture between the Universities of Bradford and Leeds and the regional development agency Yorkshire Forward to capitalise on the research and expertise of the Universities' wireless research centres. The two collaborations were designed to be complementary, using ISLI to focus on the research aspects of the project whilst using the Wireless CIC's skills and expertise to accelerate product development.

The data link activity on the technology roadmap was divided up into three phases. The first phase, which was to be driven by the Wireless CIC, was the Commercial off the Shelf (COTS) Lab Model. This was a concept demonstrator where an existing COTS wireless standard (IEEE 802.11) would be used to transmit data from multiple sensors back to an operator's terminal. The sensor data would be a video stream and simulated RWR data, as the RR017 would not be complete by this stage. This phase was to be complete by November 2007. The second phase was the completion of the 'Sensor in a Tin'. This required the completion of the 'Threat Processing' activity, the completion of the RR017, the fusing of threat data with other sensor data and the choice of the physical layer of the 'Data Link'. The second phase was due to be complete by November 2008. The final phase was to upgrade the data link from a fixed point-to-point network to an ad hoc network. This phase had a target completion date of May 2009

5.2 Metamorphosis to Phobos

In the second half of 2007 the project began to change direction. Some of the former Global Technology Group staff left the company and the interim engineering director at FDL was replaced by a permanent appointee, Peter Forrest, who started a review of all projects. It was decided that the Low Cost RWR was a commercially viable product in itself and that ad hoc networking functionality, if required, could be added later. The only missing part of the Low Cost RWR system was the threat identification software and there was no effort under way to develop any. The Defence Science and Technology Laboratory (Dstl) had new threat identification software that they were trying to commercialise and FDL entered into licensing negotiations. This change coincided with the Milcom 2007 conference and the completion of the data link strategy report, both reports are included in Volume Two. The data link strategy report concluded that the planned 802.16j extension to the WiMAX radio networking system held promise for military applications, but that FDL should wait until suitable COTS equipment was available. The commercial exhibits at Milcom 2007 showed that other companies were working on military applications for WiMAX. So there was a reasonable chance that a pure COTS data link or a COTS data link that could be customised would be available in the near future.

Consequently the EngD project was changed to match this new situation. Instead of being centred around data link research and development, the project would be focused on the system integration required to bring Phobos to market. This would be the integration of FDL technology, COTS components and threat identification software from Dstl. A team was formed within FDL to develop the product by early 2008 led by Adrian Metcalfe. As this was going to be the first product in FDL's history where software would make up a large part of the delivered product, the internal software development tools were upgraded. Microsoft Visual SourceSafe was replaced with Apache Subversion and Trac bug tracking software was added. The report on this tool upgrade is included in Chapter Fifteen in Volume Two.

5.3 From Prototype to Product

Two prototypes were developed, each as a result of the maturation of the product and of Teledyne's understanding of the needs of the market. The first prototype was used to show that it could be done: that a RR017 together with the radar identification software could correctly identify an emitter. It was built out of preexisting equipment, it was never intended to represent a system that a customer might buy. The single board computer used was taken from the Wireless CIC project and the antennas used were left over from a previous project. The mast was a re-purposed DJ's lighting stand. The only custom work done was the manufacture of the aluminium housings, which was completed through FDL's preferred contractor. The radar identification software was made available to FDL in the summer of 2008 and with assistance from Dstl staff the system successfully identified a synthetic emitter in October 2008.

After the success of the first prototype, work started in 2009 on the second prototype. The intention was to develop something that could be taken to trade shows and demonstrated to customers. It had to be much closer to the performance requirements and in a form factor that both showcased the system's low SWaP characteristics and that a customer might consider purchasing. The target was to have something to show at both Defence Security and Equipment International (DSEi) and at the Association of Old Crows (AOC)

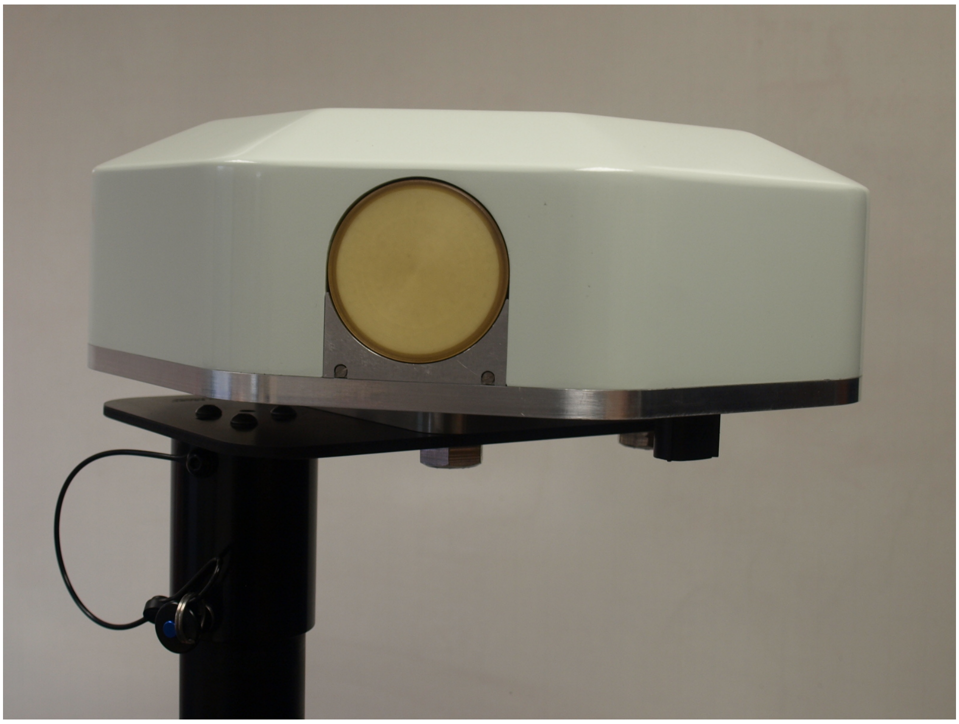


Figure 5.2: Unpainted Second Prototype

convention – two important trade shows that would occur in the Autumn of 2009. DSEi is held biennially, organised by the UK MoD and held in London. It covers all aspects of defence and security and is used to showcase UK expertise to the world. The AOC is a US based association of EW specialists which formed in the 1960s to represent and further their speciality. Crow is derived from the Second World War code name "Raven" which was used for ECM specialists. The AOC annual convention is the major technical conference and equipment exhibition for EW. A picture of the second prototype is shown in Figure 5.2 copyright Teledyne Defence Ltd.

The system is now housed on an octagonal base plate with a removable lid. The only external connections are for the power connector, Ethernet and the optional Wi-Fi antenna. The aluminium base plate is designed to fit a NATO standard mounting plate. The mounting plate shown is attached to the top of a rugged mast. The system was exhibited in a variety of different colours to reflect its suitability for land and sea operation.

In 2008 FDL had contracted the Advanced Digital Institute [30] to produce a user interface for the Phobos system. There were originally two applications: one a standard Microsoft Windows application and the other for handheld devices running Microsoft Windows CE. They also created a scenario generator which would provide the UI with synthetic data so that it could be demonstrated without the need for a Phobos system and real emitters. Along with updating and completing the Phobos unit, integration with and completion of the user interface were the main tasks completed by the team. The first functional demonstration was made to a European customer after the two shows in the Autumn of 2009 and the system was demonstrated to another European customer in early 2010.

The shows and demonstrations generated a lot of interest in the product and in the company as a whole. FDL had been acquired by the Teledyne Corporation by this point and was trading as Teledyne Defence Limited (TDL) and TDL had managed to surprise and intrigue many other members of the defence community. Whilst generating this interest in early 2010, the system was still not finished and the final push began to complete the software and to make the housing suitable for production. This included the antenna characterisation work, system performance evaluation, making the system robust and completing the user interface. The housing was redesigned with several new objectives. The antenna shielding was improved in order to improve AOA calculation. The housing was redesigned to ensure that it was cost effective to produce, suitably sealed for the lead customer's naval environment, and compatible with the electronic components such as compasses that would be used in production units.

TDL entered into a contract with a foreign military customer in 2010 to supply Phobos systems for naval platforms [31]. The first unit was delivered in the Autumn of 2010 for operational evaluation.

6 Phobos System Architecture

6.1 Overview

The Phobos system is initially available from Teledyne in two forms: the QR020 shown in Figure 6.1 and the QR020-M1 shown in Figure 5.2, both pictures copyright Teledyne Defence Ltd. The QR020-M1 is a complete product supplied as a sealed unit that requires only power, a data connection and a suitable NATO standard mounting point. The QR020 is a stripped down QR020-M1, with no housing, GPS, compass or antennas. This is to allow it to be tightly integrated into the customer's chosen host platform where it can make use of preexisting antennas and their cable runs. The host would also be expected to provide the required positioning information normally provided by the GPS and compass. This document focuses on the QR020-M1 and does not distinguish between the two, except in Table 6.1 below, as the QR020-M1 was developed as a continuation of

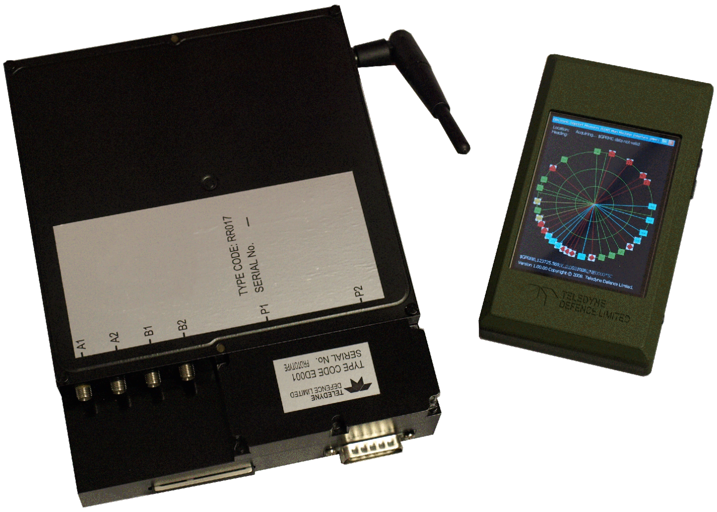


Figure 6.1: QR020 and Handheld Display

	QR020-M1[32]	QR020[33]
Antennas	Four Cavity backed Spirals	none
Receiver channels	4 Port, 2 channel (electronic switching)	
Frequency Range	2-18 GHz instantaneous coverage	
Azimuth Coverage	360 degrees	
AOA Bearing accuracy	10 degrees RMS	
System Response Time	< 1 second	
Emitter Library Size	> 5000 entries (expandable)	
Concurrent Tracks	500	
Displayed Tracks	500 (30 with hand held display)	
Power	24 W, 9-36 V DC	
Size (mm)	320 x 320 x 105	190 x 150 x 43
Volume (l)	8.5	1.25
Weight (kg)	7.2	1.9

Table 6.1: System Specification

the QR020. Both systems use the same embedded software so have the same software architecture. The pulse characteriser, single board computer and user interface options are also common to both systems. Any discussion of the extra components not found in the QR020 only applies to the QR020-M1.

The system has two halves: the sensor unit and the user interface. The sensor unit is in the shape of a rounded octagon with four antennas facing horizontally, spaced ninety degrees apart from each other, around the housing. The housing has connectors for power, Ethernet, and a Wi-Fi antenna, with the chosen data link being used to communicate with the user interface. The base plate of the system is shaped to allow it to be bolted onto a NATO standard mounting point.

Inside the housing the four antennas are attached to an optional signal conditioning unit, which is used to attenuate known interfering emissions. The output of the conditioning unit is passed to the RR017 pulse characteriser which creates pulse descriptor words that describe the received pulses. The RR017 is connected to the single board computer which analyses the pulse descriptor words to see if any of the observed emitters have entries in the emitter library. The emitter library contains descriptions of the emission patterns of known radar systems. The single board computer then creates a list of the observed radar systems in track table form, adds to the table the system's current location and heading, and sends the table over the data link to the operator. The single board computer is a Windows PC that runs the pulse processing application Phobos.exe. The received pulses are checked against the emitter library using software licensed from the commercial arm of Dstl, the technology agency of the UK MoD.

The second half of the system is either a laptop computer or a handheld device running the user interface application. This application receives the track tables and displays them to the operator. The laptop user interface is more sophisticated than the handheld, showing the operator more information about the observed emitters and their host platforms. It allows the operator to reconfigure the sensor unit.

The top level system parameters taken are shown in Table 6.1 and the relevant parameters of the RR017 pulse characteriser are shown in Table 6.2 with the figures taken from the product data sheets [33, 32, 25]. Teledyne makes a number of options available for customers who wish to extend the capabilities of the system. The number of antennas can be increased to eight and the number of receiver channels can be increased to four. The operating frequency range can be extended down to 500 MHz and up beyond 18 GHz. The frequency accuracy and resolution can be improved to < 4.5 MHz and < 12.5 MHz. A solar power pack can be included for unattended remote operation. The software options include raw data capture and pulse analysis tools.

Frequency Measurement	< 25 MHz resolution, < 10 MHz accuracy
Dynamic Range	42 dB, 62 dB with attenuator
Minimum Pulse Width	75 ns
TOA Measurement Resolution	10 ns
Recovery Time	500 ns maximum, 300 ns typical
Environment Pulse Density	> one million pulses per second

Table 6.2: Pulse Measurement Capabilities

The remainder of this chapter describes the architecture of both the system hardware and software in detail.

6.2 Hardware

The Phobos hardware can be divided up into three logical parts: the housing, pulse detection and measurement, and pulse processing. These divisions are shown in Figure 6.2.

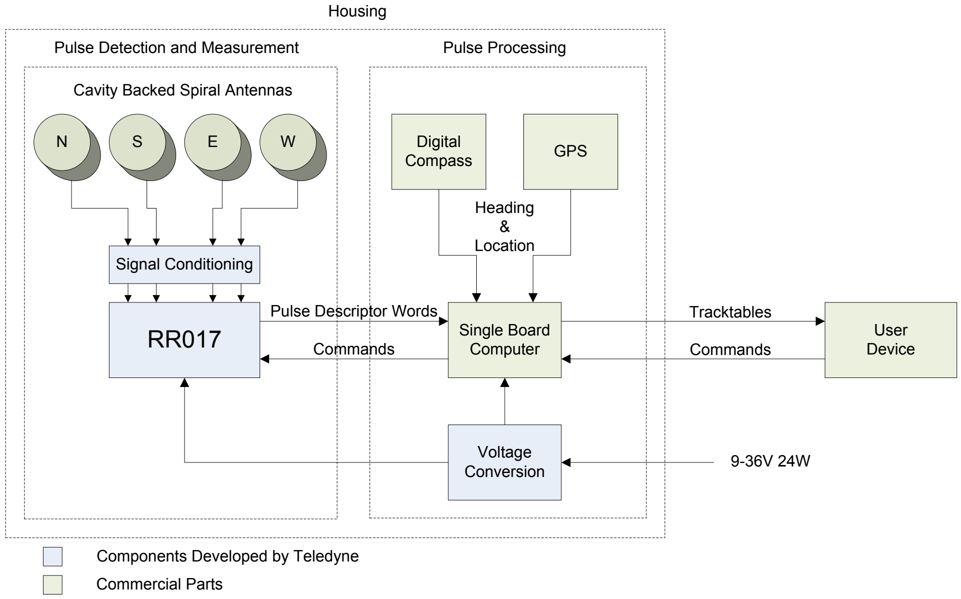


Figure 6.2: Hardware Block Diagram

6.2.1 Housing

The housing has to perform two roles for the system: to allow the system to be installed on a platform of choice and to protect the system from possible harsh external conditions, so it can continue to operate. The design has to achieve both, whilst remaining appropriate for a low cost system.

The prototype housing was machined out of aluminium. Aluminium is frequently used in Teledyne's products which makes it attractive as a design can be realised very quickly, compared to using other materials. The base plate is a rounded octagon, approximately 32 centimetres in diameter. It has mounting holes for a NATO standard mounting plate and connectors for power, Ethernet, USB debug and a Wi-Fi antenna. The lid for the housing was originally made from nylon rather than aluminium. This was to allow the GPS receiver an unobstructed view of the sky. The material had to be changed to aluminium when it was discovered that the nylon lid prevented antenna calibration. Being able to calibrate the antennas and obtain accurate AOA information was much more important for system development than accurate GPS information. An analysis of the lids is included in Chapter Eleven in Volume Two.

During operation the expected environmental hazards are vibration, water and dust. The extent of these is dependent upon the host platform and where it is operating. On a naval platform the device will be exposed to high winds, high seas and salt water, whereas a

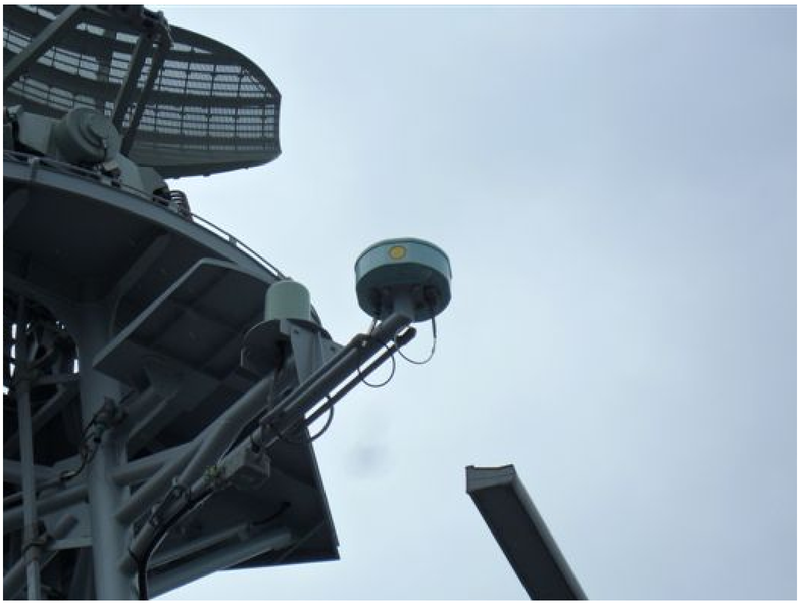


Figure 6.3: Phobos on a Warship

tracked platform might be partially submerged and exposed to sand storms. Both platforms will have internal vibrations due to engines and other equipment and experience sudden shocks caused by weapons systems. As the system has no moving parts the vibration resistance is dependent upon how the system components are attached to each other and how those connections are damped. The resistance to water and dust ingress is determined by how well all the joints in the system are sealed. The prototype was not designed with these requirements in mind, its vibration resistance is unknown and the unit is in no way sealed. During testing the unit had to be protected with a temporary waterproof cover to prevent water ingress.

Toward the end of the period of research and development described in this document other teams at Teledyne were working on adapting the housing for installation on a warship – the host platform for the first customer. The first production standard version of the housing has sides that are made from metal as well as the base plate. Recesses in the metal sides have been made to house the antennas with the face of the antenna positioned slightly proud of the housing to solve the antenna calibration problems. The lid has returned to being manufactured from nylon to allow the GPS to operate and is now much smaller as it no longer encloses the sides. The new design is also completely sealed against water. This later model is shown in Figure 6.3, copyright Adrian Metcalfe.

End Point	Direction	Buffering	Purpose
2	out	double	command input
4	in	double	command response
8	in	triple	PDW data

Table 6.3: USB Endpoints

6.2.2 Pulse Detection and Measurement

The system has four cavity backed spiral antennas with which to receive radar pulses. The antennas are designed for the frequency range of the system and were chosen as they suit its low cost and compact nature. They are connected by equal length cables to the signal conditioning unit which is then connected to the four inputs of the RR017 pulse characteriser. The signal conditioning unit is an optional component that is used to temporarily blank predetermined frequency ranges of the customer's choosing. Examples are to blank mobile communication bands for naval vessels when operating close to shore and to blank a co-sited emitter that will interfere when active. On the prototype the signal conditioning unit was required to blank the emissions of another system that was also in operation at the primary testing location, Portsdown West, a Dstl facility.

The RR017 is connected to the single board computer by a USB bus. USB was chosen for the RR017 for reasons of simplicity and cost over Ethernet. The bus interface is provided by an FX2 USB interface chip from Cypress Semiconductor [34] that supports the 480 Mbps USB 2.0 protocol. At 'power on' this USB interface controller needs to be configured in order to interface correctly with the single board computer. The FX2 device contains an 8051 compatible microcontroller that is used to perform this task. The software is loaded from an EEPROM and has to configure the required end points before placing the device in slave FIFO mode as described in Chapter Fifteen in Volume Two. In this mode the microcontroller is inactive and data is written directly to the device which then makes it available on the USB interface once the packet has been committed. Packets can be either manually or automatically committed when 512 bytes have been received. The data bus and the packet commit control line are driven by the RR017's FPGA. The USB output used for pulse data is triple buffered to reduce the amount data lost between USB reads by the single board computer.

This USB interface is used to send commands from the single board computer to the RR017 and to send command responses along with pulse data back to the single board computer. The configuration of the USB endpoints is shown in Table 6.3

6.2.3 Pulse Processing

At the centre of the pulse processing unit is the Single Board Computer. It is a Compulab FitPC2 with a single core Intel Atom processor [35]. The system boots from and stores all of its data on a single flash memory SSD connected using a SATA interface. The two peripherals, the compass and the GPS, are both connected using USB and they are both bus powered eliminating the need for extra power interfaces.

In the early stages of the project a different single board computer [36] was used, and other boards were considered to replace it as well as the FitPC2. At the time low cost development boards based around ARM architecture processors such as the Beagle Board [37] had become available and the ARM architecture is well suited to low cost and low power applications. The OMAP3 system on chip that the Beagle is based around, as well as an ARM Cortex A8 processor, also contains a DSP co-processor which would allow the computational work of the system to be partitioned between them. This would have allowed the threat identification software to run on its own dedicated processor leaving the CPU to manage the system and the data flow. The openness of the development board also provided a reference as to how the OMAP processor could be integrated into a PCB. This would have greatly simplified integrating it into the RR017 if that had been desired.

However the agreement between Teledyne and the license holders of the radar identification software precluded these options as the software was only going to be made available in binary form, compiled for an x86 Windows PC.

6.3 Software

The design for the Phobos application software was originally done by Adrian Metcalfe at the start of the project. The design has evolved over the lifetime of the project into the structure shown in Figure 6.4. The diagram has been simplified by removing some members and methods entirely and not showing some parameters in order to make it readable, but the structure is unchanged. The software is object oriented C++ where significant effort has been made to make the application as easily portable as practicable. The software is designed around the flow of the data rather than the flow of control. This means that in the main loop instead of data being passed from stage to subsequent stage, the main loop calls into methods of member objects with references to data held by other

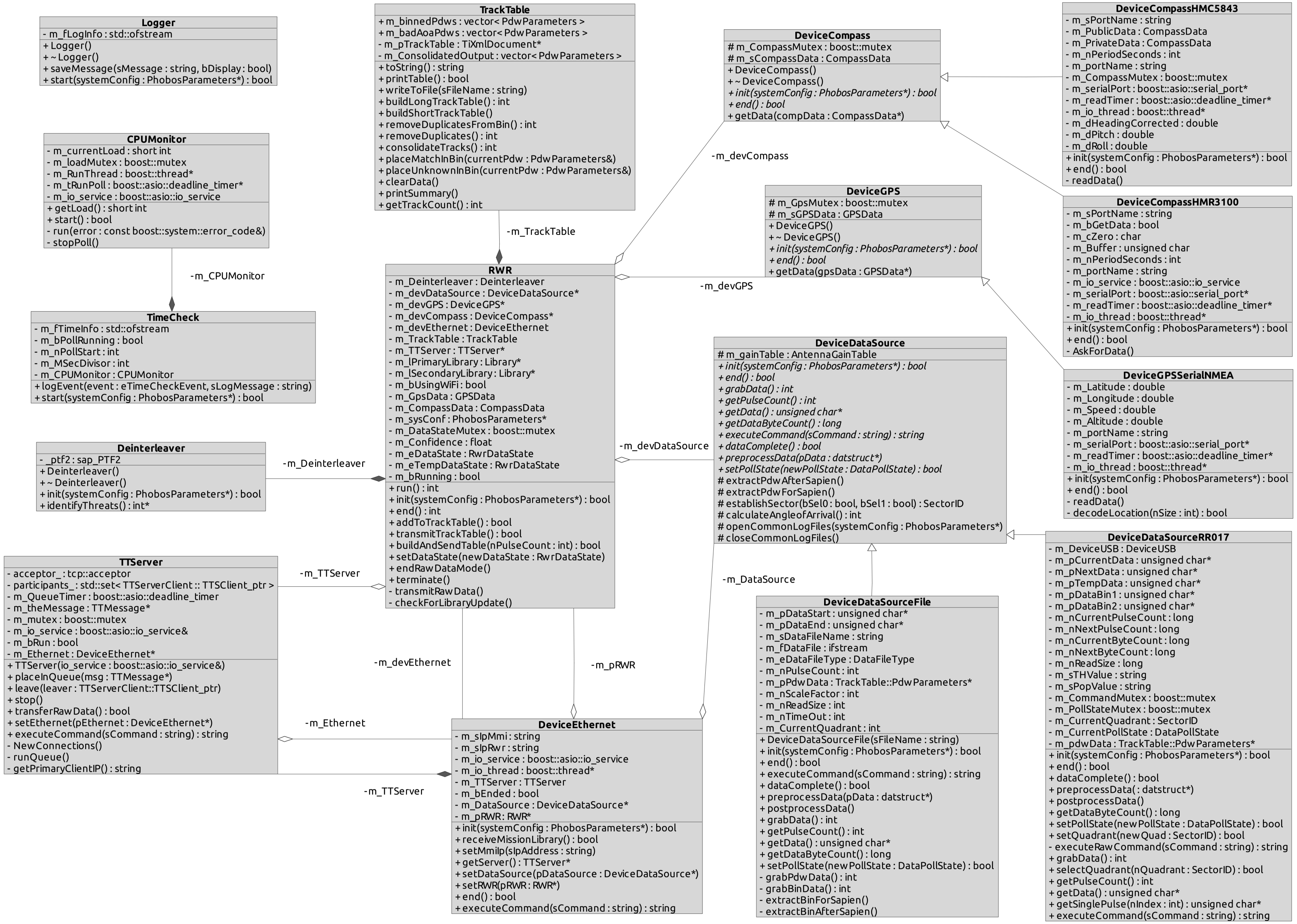

Logger
- m_fLogInfo : std::ofstream
+ Logger()
+ ~ Logger()
+ saveMessage(sMessage : string, bDisplay : bool)
+ start(systemConfig : PhobosParameters*) : bool
CPUMonitor
- m_currentLoad : short int
- m_loadMutex : boost::mutex
- m_RunThread : boost::thread*
- m_tRunPoll : boost::asio::deadline_timer*
- m_io_service : boost::asio::io_service
+ getLoad() : short int
+ start() : bool
- run(error : const boost::system::error_code&)
- stopPoll()
-m_CPUMonitor
TimeCheck
- m_fTimeInfo : std::ofstream
- m_bPollRunning : bool
- m_nPollStart : int
- m_MSecDivisor : int
- m_CPUMonitor : CPUMonitor
+ logEvent(event : eTimeCheckEvent, sLogMessage : string)
+ start(systemConfig : PhobosParameters*) : bool
Deinterleaver
- _ptf2 : sap_PTF2
+ Deinterleaver()
+ ~ Deinterleaver()
+ init(systemConfig : PhobosParameters*) : bool
+ identifyThreats() : int*
-m_Deinterleaver
TTServer
- acceptor_ : tcp::acceptor
- participants_ : std::set< TTServerClient :: TTSClient_ptr >
- m_QueueTimer : boost::asio::deadline_timer
- m_theMessage : TTMessage*
- m_mutex : boost::mutex
- m_io_service : boost::asio::io_service&
- m_bRun : bool
- m_Ethernet : DeviceEthernet*
+ TTServer(io_service : boost::asio::io_service&)
+ placeInQueue(msg : TTMessage*)
+ leave(leaver : TTServerClient::TTSClient_ptr)
+ stop()
+ transferRawData() : bool
+ setEthernet(pEthernet : DeviceEthernet*)
+ executeCommand(sCommand : string) : string
- NewConnections()
- runQueue()
- getPrimaryClientIP() : string
-m_TTServer
-m_Ethernet
-m_TTServer
TrackTable
+ m_binnedPdws : vector< PdwParameters >
+ m_badAoaPdws : vector< PdwParameters >
- m_pTrackTable : TiXmlDocument*
- m_ConsolidatedOutput : vector< PdwParameters >
+ toString() : string
+ printTable() : bool
+ writeToFile(sFileName : string)
+ buildLongTrackTable() : int
+ buildShortTrackTable()
+ removeDuplicatesFromBin() : int
+ removeDuplicates() : int
+ consolidateTracks() : int
+ placeMatchInBin(currentPdw : PdwParameters&)
+ placeUnknownInBin(currentPdw : PdwParameters&)
+ clearData()
+ printSummary()
+ getTrackCount() : int
-m_TrackTable
RWR
- m_Deinterleaver : Deinterleaver
- m_devDataSource : DeviceDataSource*
- m_devGPS : DeviceGPS*
- m_devCompass : DeviceCompass*
- m_devEthernet : DeviceEthernet
- m_TrackTable : TrackTable
- m_TTServer : TTServer*
- m_lPrimaryLibrary : Library*
- m_lSecondaryLibrary : Library*
- m_bUsingWiFi : bool
- m_GpsData : GPSData
- m_CompassData : CompassData
- m_sysConf : PhobosParameters*
- m_DataStateMutex : boost::mutex
- m_Confidence : float
- m_eDataState : RwrDataState
- m_eTempDataState : RwrDataState
- m_bRunning : bool
+ run() : int
+ init(systemConfig : PhobosParameters*) : bool
+ end() : int
+ addToTrackTable() : bool
+ transmitTrackTable() : bool
+ buildAndSendTable(nPulseCount : int) : bool
+ setDataState(newDataState : RwrDataState)
+ endRawDataMode()
+ terminate()
- transmitRawData()
- checkForLibraryUpdate()
-m_devEthernet
-m_pRWR
DeviceEthernet
- m_sIpMmi : string
- m_sIpRwr : string
- m_io_service : boost::asio::io_service
- m_io_thread : boost::thread*
- m_TTServer : TTServer
- m_bEnded : bool
- m_DataSource : DeviceDataSource*
- m_pRWR : RWR*
+ init(systemConfig : PhobosParameters*) : bool
+ receiveMissionLibrary() : bool
+ setMmiIp(sIpAddress : string)
+ getServer() : TTServer*
+ setDataSource(pDataSource : DeviceDataSource*)
+ setRWR(pRWR : RWR*)
+ end() : bool
+ executeCommand(sCommand : string) : string
-m_devCompass
DeviceCompass
# m_CompassMutex : boost::mutex
# m_sCompassData : CompassData
+ DeviceCompass()
+ ~ DeviceCompass()
+ init(systemConfig : PhobosParameters*) : bool
+ end() : bool
+ getData(compData : CompassData*)
DeviceGPS
# m_GpsMutex : boost::mutex
# m_sGPSData : GPSData
+ DeviceGPS()
+ ~ DeviceGPS()
+ init(systemConfig : PhobosParameters*) : bool
+ end() : bool
+ getData(gpsData : GPSData*)
-m_devGPS
-m_devDataSource
DeviceDataSource
# m_gainTable : AntennaGainTable
+ init(systemConfig : PhobosParameters*) : bool
+ end() : bool
+ grabData() : int
+ getPulseCount() : int
+ getData() : unsigned char*
+ getDataByteCount() : long
+ executeCommand(sCommand : string) : string
+ dataComplete() : bool
+ preprocessData(pData : datstruct*)
+ setPollState(newPollState : DataPollState) : bool
# extractPdwAfterSapien()
# extractPdwForSapien()
# establishSector(bSel0 : bool, bSel1 : bool) : SectorID
# calculateAngleofArrival() : int
# openCommonLogFiles(systemConfig : PhobosParameters*)
# closeCommonLogFiles()
-m_DataSource
DeviceDataSourceFile
- m_pDataStart : unsigned char*
- m_pDataEnd : unsigned char*
- m_sDataFileName : string
- m_fDataFile : ifstream
- m_eDataFileType : DataFileType
- m_nPulseCount : int
- m_pPdwData : TrackTable::PdwParameters*
- m_nScaleFactor : int
- m_nReadSize : int
- m_nTimeOut : int
- m_CurrentQuadrant : int
+ DeviceDataSourceFile(sFileName : string)
+ init(systemConfig : PhobosParameters*) : bool
+ end() : bool
+ executeCommand(sCommand : string) : string
+ dataComplete() : bool
+ preprocessData(pData : datstruct*)
+ postprocessData()
+ grabData() : int
+ getPulseCount() : int
+ getData() : unsigned char*
+ getDataByteCount() : long
+ setPollState(newPollState : DataPollState) : bool
- grabPdwData() : int
- grabBinData() : int
- extractBinForSapien()
- extractBinAfterSapien()
DeviceCompassHMC5843
- m_sPortName : string
- m_PublicData : CompassData
- m_PrivateData : CompassData
- m_nPeriodSeconds : int
- m_portName : string
- m_CompassMutex : boost::mutex
- m_serialPort : boost::asio::serial_port*
- m_readTimer : boost::asio::deadline_timer*
- m_io_thread : boost::thread*
- m_dHeadingCorrected : double
- m_dPitch : double
- m_dRoll : double
+ init(systemConfig : PhobosParameters*) : bool
+ end() : bool
- readData()
DeviceCompassHMR3100
- m_sPortName : string
- m_bGetData : bool
- m_cZero : char
- m_Buffer : unsigned char
- m_nPeriodSeconds : int
- m_portName : string
- m_io_service : boost::asio::io_service
- m_serialPort : boost::asio::serial_port*
- m_readTimer : boost::asio::deadline_timer*
- m_io_thread : boost::thread*
+ init(systemConfig : PhobosParameters*) : bool
+ end() : bool
- AskForData()
DeviceGPSSerialNMEA
- m_Latitude : double
- m_Longitude : double
- m_Speed : double
- m_Altitude : double
- m_portName : string
- m_serialPort : boost::asio::serial_port*
- m_readTimer : boost::asio::deadline_timer*
- m_io_thread : boost::thread*
+ init(systemConfig : PhobosParameters*) : bool
+ end() : bool
- readData()
- decodeLocation(nSize : int) : bool
DeviceDataSourceRR017
- m_DeviceUSB : DeviceUSB
- m_pCurrentData : unsigned char*
- m_pNextData : unsigned char*
- m_pTempData : unsigned char*
- m_pDataBin1 : unsigned char*
- m_pDataBin2 : unsigned char*
- m_nCurrentPulseCount : long
- m_nNextPulseCount : long
- m_nCurrentByteCount : long
- m_nNextByteCount : long
- m_nReadSize : long
- m_sTHValue : string
- m_sPopValue : string
- m_CommandMutex : boost::mutex
- m_PollStateMutex : boost::mutex
- m_CurrentQuadrant : SectorID
- m_CurrentPollState : DataPollState
- m_pdwData : TrackTable::PdwParameters*
+ init(systemConfig : PhobosParameters*) : bool
+ end() : bool
+ dataComplete() : bool
+ preprocessData(: datstruct*)
+ postprocessData()
+ getDataByteCount() : long
+ setPollState(newPollState : DataPollState) : bool
+ setQuadrant(newQuad : SectorID) : bool
- executeRawCommand(sCommand : string) : string
+ grabData() : int
+ selectQuadrant(nQuadrant : SectorID) : bool
+ getPulseCount() : int
+ getData() : unsigned char*
+ getSinglePulse(nIndex : int) : unsigned char*
+ executeCommand(sCommand : string) : string

Figure 6.4: Simplified Class Diagram for the Phobos Application

member objects when required. The member objects manage their own data and in doing this the amount of copying is minimised. The software is required to get as much data from the RR017 as is available and deliver a list of identified emitters in XML track table form in less than one second.

This section is divided up into five parts: Configuration, Main Loop, Hardware Interfaces, Software Interfaces and Telemetry. Configuration describes how the software can be configured at compile time using the preprocessor, at run time using command line arguments and by the configuration file. Main Loop describes the flow of the programs main loop which is independent of how the software has been configured. The two Interfaces sections describe how the software is connected to other parts of the system in a portable and performant manner. The telemetry section describes the software facilities available for debugging the system, logging and measuring the system's performance.

6.3.1 Configuration

The software is deliberately very configurable. This allows the system to operate whilst various hardware components or their inputs are unavailable. Originally this was to facilitate software development before system hardware was available, but subsequently made it simpler to adapt the system as and when the hardware changed. There are three ways to change the software configuration: at compile time, using command line arguments and using the configuration file.

There are two build profiles for the software which follow the Microsoft convention by being called Release and Debug. The Release build is the build that would be shipped to customers. It is an optimised build with all debugging features disabled. The Debug build is used for software development and as well as not being optimised it contains two other development features: it enables the performance measurement code and turns on data logging. The performance measurement code is designed to estimate where the software is spending most of its time and is described in detail in Chapter Twelve in Volume Two. This is to allow not only the performance to be assessed but also to predict the potential of the platform and to record a more accurate picture of the radar environment. The data logging code saves the raw pulses and the processed pulse data to file. As well as facilitating the resolution of software issues, the saved data files provide valuable source data for when the system is being tested and developed far from actual emitters. These options were made compile time options for reasons of performance and safety. In the release build the code is not built in, so has no affect on performance and it cannot be mistakenly enabled. The Release profile generates an executable called Phobos.exe whilst the Debug profile generates an executable called PhobosD.exe. The name of the executable is used to choose the name of the directory hierarchy in which the log files are stored.

The command line arguments are a convenient way to change a few settings at run time that are not suited to the configuration file. All options chosen on the command line will be used in preference to defaults or to options set in the configuration file. Up to three command arguments may be given in any order however none are required. The valid options are: the path to a pulse data file, the path to an emitter library file and the path to a configuration file. The only way to run the system using a prerecorded data file instead of getting pulse data from the RR017 is to specify the data file on the command line. It cannot be included in the configuration file. For historical reasons the system supports two formats of pulse data file. When the radar identification software was initially delivered by Dstl, it came with a reference data file for testing. At this point in the project, it was the only source of realistic pulse data, so a large part of the software was developed using it. The second pulse data file format is that of the RR017 which quickly became the most commonly used one once RR017 hardware was available. The data file type is determined from the file name extension. This functionality is not expected to be exposed in the final product.

The configuration file is an XML document parsed using the TinyXML library [38]. This was an obvious choice as the TinyXML library was already being used to generate the track tables. Either the default configuration file must exist, or an alternative must be specified on the command line, for the program to start correctly. The configuration file is designed to hold all of the system parameters that may be unique to that platform and are unlikely to be changed frequently, such as the system hardware configuration. Not all of the elements have to be present for the configuration file to be parsed correctly. Default values will be used in some cases if required. The configuration file options are shown in Table 6.4.

6.3.2 Main Loop

After successful initialisation the main loop begins and runs until either an unrecoverable error occurs or an exit command is received. The main loop is illustrated in Figure 6.5. The nature of the main loop is not dependent upon any of the configuration options, but it does have its own state. At the start of each loop iteration a private

Log Directory	Path to where the log files should be saved
Emitter Library	Path to the initial Emitter Library
GPS	Port and type
Compass	Port, type and host platform offset
Antenna Patterns	Path to Antenna calibration files
Bearing Accuracy	Bearing calculation accuracy in degrees
Default Threshold	Default RR017 detection threshold
Polling	Flag to set polling mode and initial quadrant
Confidence	Emitter identification confidence threshold
Data Read Size	How many bytes to read from the data source
Data Read Timeout	Data source read timeout in ms
Playback Scale Factor	Scale factor to alter the data file replay rate

Table 6.4: Configuration File Options

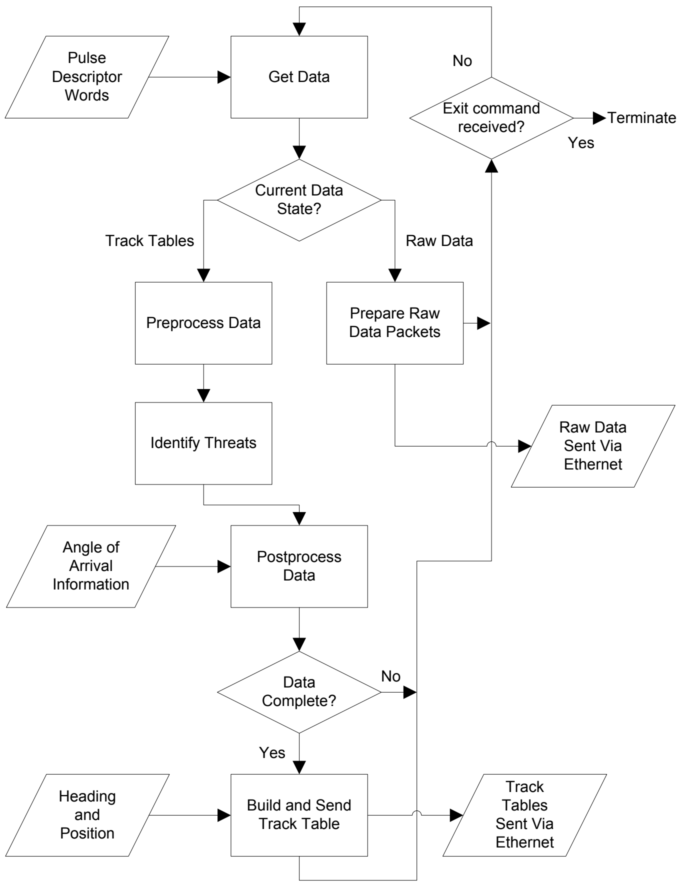


Figure 6.5: The Main Loop

copy of the current state is taken, to ensure the entire iteration is completed with a consistent state. The only current state variable is the data state, which controls whether the system should send raw pulse data to the operator, or process the pulses and send a list of emitters in the form of a track table instead. This is an option that the operator can change at run time by sending a command from the user interface.

If raw data is selected then the loop is very tight. Once the data has been received from the data source preexisting code is called to send the raw pulse data to the operator as UDP packets. When complete, if no exit command has been received, then the next iteration of the loop can begin, otherwise the program terminates.

When track table generation is selected the loop contains several more stages. First the data is preprocessed according to the needs of the emitter identifier. Only the required information is extracted from the binary pulse information at this stage, to minimise the processing required. This extracted information is then loaded into the data structure required by the emitter identifier, and the identification process starts. Once the threats have been identified the emitter identifier returns a structure with the threat tag information. The emitters only have a numeric identifier, not a name. Another library is required on the operators terminal to convert the numeric identifier into a platform name. This means that no one part of the system has the complete mapping from radar pattern to emitter name, which makes it easier to keep the contents of emitter library secret and reduces the impact of part of the system falling into enemy hands.

Once the emitter information has been combined with the original parametric data, all of the remaining information is extracted from the binary pulse data and the AOA calculated for pulses of interest. Pulses of interest are either pulses with emitter tags or all pulses. This flag can currently only be changed in the code, but in the future it could be moved out to the configuration file and made changeable at run time. The AOA is calculated by comparing the two amplitudes recorded in each pulse descriptor word. The emitter is assumed to be far enough away that the difference in amplitude at each antenna is solely down to the directionality of each antenna and their differing orientation. The system has been calibrated in an anechoic chamber where the response of the four antennas was measured through 360 degrees across the frequency range. This data is then turned into difference tables of the expected difference in amplitude at each angle of arrival. To find a match, the difference table for each antenna pair at that frequency is walked from both ends. The two walkers are not expected to finish in the same place and there are two factors to allow for this: tolerance and angle tolerance. The tolerance is how close the walkers must get to the difference value before they declare a match. Angle tolerance is how far apart in degrees those two matches can be before the match is declared unsuccessful. If both matches are within the angle tolerance the resulting AOA value is the average of the two. It is however quite likely that the match is unsuccessful. This could occur because the emitter is in an adjacent quadrant to the one being observed and RR017 has only received the pulse on one channel, the other measurement just being noise. In the case of an unsuccessful match the pulse is assigned the AOA equal to boresight on the antenna that received the strongest signal.

Once this is complete, the `DataSource`[1] is asked whether there is more data to come as part of this set. This is to allow for systems where the data is received in stages, such as quadrant by quadrant instead of all the data from the system's field of view at once. Although the RR017 does poll around its four ports it does not use this feature. Instead the polling is managed in the `DataSource` instead. This change was made to the `DataSource` in preparation for multi-threaded data grabs, but this work is not yet complete.

The final stage of the track table branch contains many actions. Build and send track table takes the parametric pulse data, condenses it, extracts further statistical information, gets information about the position and heading of the host platform, generates a XML track table and places it in the queue of the track table server. The system will have received many pulses from each emitter and this needs to be reduced to one track table entry per emitter whilst retaining as much information about the pulses as is useful. To do this, the field of view of the system is divided up into bins which by default are ten degrees in size. Any pulse that has the same emitter tag as another pulse in a bin is assumed to be a duplicate. The variation of frequency and pulse width for the emitter is recorded as minimum, maximum and average values. This technique is crude and has two major disadvantages: emitters whose AOA is near a bin boundary will be recorded differently to emitters aligned with the centre of a bin due to straddling and that there can only be one unknown emitter in each bin.

The data is now a list of emitters rather than pulses and this list is then transformed into the body of an XML format track table document. The header contains information about the host platform, obtained from the GPS and compass. This is the location, altitude, speed and heading of the platform. The XML document is then placed in the queue of the track table server and the next iteration of the loop can begin. Before that happens

[1] A monospace font is used here and subsequently for source code class names. Refer to the class diagram Figure 6.4 for more information.

the program checks that an exit command has not been received, the reception of which would cause the program to terminate rather than start the next iteration.

6.3.3 Hardware Interfaces

There are three components in the system that have hardware that the system software must drive directly: the RR017 pulse characteriser, the compass and the GPS. This section covers those hardware interfaces.

Pulse Characteriser There are three USB endpoints on the RR017 that are used. One for receiving the pulse data, one for issuing commands and one for receiving the response to the issued commands. The data endpoint is configured using the Cypress EZ-USB library functions using two parameters set in the configuration file, read size and timeout. The read size is the maximum number of bytes to collect in one transaction. The timeout is the maximum amount of time any one read transaction should take. Altering the read size variable changes the behaviour as expected, but changing the timeout does not. Shortening the read timeout does seem to shorten the read time, but not in a constant or predictable way. Setting the read size to be a large value in order to always catch all the available data and relying upon the timeout to ensure the system can meet its timing requirements proved to be an unworkable strategy. Attempts to limit the maximum possible main loop iteration time are described in Chapter Twelve in Volume Two.

Three buffers are created for the pulse data: bin one, bin two and a temporary buffer. The RR017 is set to the North East quadrant and four data reads are made, one for each quadrant, whilst issuing the quadrant change commands in between. The data received is kept in the temporary buffer. The two data bins are accessed through two pointers called Current and Next. Each PDW is 96 bits in size which does not align with the 512 bytes USB packet size. This means only 504 bytes of the packet are actual data, with the remaining space in the packet padded out with zeros. The data is then copied from the temporary buffer into the buffer pointed to by the Next pointer with the padding being stripped out at the same time. When this is complete the Next and Current pointers are swapped. This was done to allow for multithreaded data grabbing. It was hoped that being able to perform the next data grab whilst the current data was being processed would provide a significant performance improvement. Initial testing suggested that the CPU was not being heavily loaded. The work described in the Performance Considerations chapter in Volume Two has cast doubt on the usefulness of this improvement.

All commands received from the UI must be issued to the RR017, except two. The exceptions are switching between raw data mode and track tables, and the exit command. Rather than send actual RR017 commands, the commands received from the UI are short strings that are used to refer to an action. The system then issues the corresponding command or commands to the RR017. This has the advantage of there being a fixed number of valid commands. The commands are listed in Table 6.5.

The RR017 data source maintains its own state as to whether it is in fixed quadrant or polling mode. When a quadrant command is received it changes the state appropriately as well as issuing the corresponding command to the RR017. As the antennas are physically attached to the data source the antenna gain tables are regarded as the property of and are maintained by the RR017 data source. At system start up the gain tables need to be initialised whereby difference tables are created for the gain of each antenna relative to its neighbours for each frequency. The gain tables can then be used to calculate the AOA of a pulse by looking up the difference in amplitude between the two receiver channels in the table. This abstracts the details of the postprocessing step from the main loop as the data source just has to call the appropriate function for one of its members.

Location and Heading Both the GPS and compass are USB devices that are accessed through a virtual serial port. Accessing the devices was found to be very slow, taking several seconds to return the required data. This meant that they could not be part of the main loop. Also unless Phobos was fitted to very fast moving platform the values would not be expected to change very rapidly. Instead data was obtained in a loop running in a separate thread that would sleep for one second after each read, using a timer. A copy of the current values for location and heading would then be taken by the main loop which would only have to wait for the acquisition of the mutex which guards the shared value. In order to maintain platform independence the Boost library was used instead of operating system functionality for the serial port access, timers, threads and mutexes.

6.3.4 Software Interfaces

Data File As well as the RR017 there is another source of pulse data – the data file data source. In the early stages of the project when the RR017 was not yet available, initial software development was done with the test data file provided with the emitter identifier. This allowed the development of the track table generation code, Ethernet

Command	Parameters	Meaning
RAW	IP Address + Start/Stop	Start or stop sending raw data to the given address
EXIT	none	The program should exit at the end of this loop
QUAD	quadrant name or auto	Switch between fixed quadrants or auto polling mode
IFM RESET	none	Full IFM reset including thresholds
IFM SOFT RESET	none	Reset IFM hardware but retain thresholds
THOLD	new threshold value	Change the pulse detection threshold to the given value
POPTH	new threshold value	Change the pulse on pulse threshold to the given value
ATTEN	IN or OUT	Enable or disable the attenuator

Table 6.5: RR017 Commands

link and Operator interface in the absence of any actual pulses. The code was subsequently changed to allow the loading of data file in RR017 PDW format and this is now the main method of software testing as signal generators do not create realistic radar environments. The raw data files from every field test against real emitters can now be used as realistic input with a number of shortcomings.

When using a data file as input there are two main differences as to how the data flows. Firstly, it is impossible to know in what size chunks the data originally arrived as there are no markings in the file. This means that the environment cannot be reproduced accurately. Secondly, the data file will be finite in size so there must be a facility to either stop or play the file again. Accessing the data file is also much quicker than accessing the RR017 so the systems speed of operation will be much greater. From the pulse timestamps the system attempts to get one second's worth of data but that and observing the antenna polling patterns proved to be unreliable.

Ethernet Communication between the Phobos unit and the operators computer is done using Ethernet. The link was originally developed by the Advanced Digital Institute [30], but has been reimplemented and substantially changed over the course of the project. The track table server, an instance of `TTServer`, listens on port 30000 and waits for clients to connect. When a client, running an instance of `TTClient`, connects it is sent a HELLO message containing the server version number. The client is then added to the list of current clients by the server and the client itself must acknowledge the HELLO message by sending an ACK message in response. Once the connection has been established the client can expect to receive regular TABLE messages and the server may be sent COMMAND messages. The protocol also includes a HEARTBEAT message for when the server has no data to send, but this functionality is not in use. The receipt of these messages must be signalled by sending an ACK message back to the other party. ACK messages themselves are not acknowledged. ACK messages for TABLE messages contain the length of the TABLE message, whilst the ACK messages for COMMANDs contain the command response and whether it was successful. The COMMAND messages are passed to the `DeviceEthernet` which passes them on to the `DataSource` to be executed. This requires the `DataSourceFile` to support commands, as the pointer is abstract, which makes little sense as nothing can be changed. Accordingly in response to a command, the `DataSourceFile` reports that the command failed, with a message that commands are not supported.

The ACK messages are used to validate that the connections are still alive. Every time the server receives a message from one of the clients it resets that clients timer back to five seconds. If a client timer does expire the client is removed from the list, the connection closed and it will no longer receive messages. The client has a similar timer for the server. If it receives no messages from the server for three seconds, it will close the connection to the server. After that it will wait for one second and then try to reconnect to the server. This is to give the link resiliency and to make it possible to restart both the Operator's UI and the Phobos system independently.

The code for `TTServer` and `TTClient` is implemented using the Boost ASIO library [39] and is derived from the example applications that come with it. The Boost library was chosen to avoid any operating system dependency. All of the transactions are carried out using asynchronous call backs run by an instance of the ASIO library's IO Service which runs in its own thread. This isolates the main loop from the network activity ensuring that neither one will block the other needlessly.

6.3.5 Telemetry

The software contains a number of features that act as loggers, development aids and performance measurement tools. With the exception of the system log their inclusion is optional as they provide no benefit in normal operation and their actions may reduce overall system performance. Accordingly they are normally only included in development builds. The amount of logging that is enabled can have a significant impact on the duration of the main loop, so the amount of logging performed is very configurable. The system log file, which is always enabled, is a plain text log file that records system events. It is used to record the system start up time, initial parameters, shut down time and any errors that occur in between. There is one global instance of the `logger` class allowing it to be easily called from any other class. The `saveMessage` method has a Boolean flag which determines if the message should be displayed on the terminal window of the Phobos application as well as saved to file. This is to warn a developer who may be watching at the time that there is a problem. All of the log files are saved into a directory structure hierarchy so that they can be easily linked and attributed to a particular period of operation. The root directory of the hierarchy is specified in the configuration file and the full path of any log file is

root/day-month/(executable name)/(log name)-HHMM-SS.extension

Each log file from the same run will have the same time in the file name. Saving the executable name indicates whether it was a release or a debug build as they have different names.

Pulse data logging is done by the data source, not a separate object as all the pulse data is contained within the data source. There are three logging options for pulse data: Raw, Tagged and All. Enabling Raw data logging saves the binary pulse data to file in an unprocessed form as it is received. The Tagged and All options are mutually exclusive and record the parameters of the chosen set of pulses in text form to file. This is done as part of the post processing step of the main loop. All of the pulse logging options are enabled by preprocessor directives, so are completely absent if not enabled at compile time.

During development it is also useful to be able to check that pulses are being received without monitoring log files or using the operator's interface. There are two ways that this can be done. For a debug version of the Phobos application, each time a track table is built before it is placed in the queue of the `TTServer`, a summary is printed in the terminal window of the application. If there are less than sixteen emitters it prints the emitter and bearing pairs, otherwise – just the number of tracks in the table. The `TTServer` and `TTClient` classes were developed using test applications with artificial data to keep their development orthogonal to the Phobos application. The `TTClient` test application can be used as a second way to determine whether any build of the Phobos application is actually detecting emitters and generating tables. The information is reduced as the test application only prints the size of the table, but if that value changes it is a good indication that the system is detecting and identifying pulses.

The last part of the telemetry is the `TimeCheck` class. The usefulness of the system depends upon its performance. As a low cost system the software has to be able to both meet the performance targets and get the most out of the hardware. The `TimeCheck` class is an optional global object that is used to measure the duration of tasks. Each task of interest has a corresponding start and stop event, the occurrence of which is recorded by a call to the `logEvent` member function with the event name as a parameter. The system time when the start event is received is recorded in system ticks. When the matching stop event is received, the duration for the event in milliseconds is written to a text log file. The log event function has enough logic to check that events are received in the correct order. There is also an extra parameter that may be recorded along with the time for some tasks. The time required to build a track table is expected to be related

Data: 1563	PrePro: 0	ID: 78	Density: 4028	PostPro: 78
Build: 47	Tracks: 32	Loop: 1766	Load: 4	

Table 6.6: Example Time Log Entry

to the number of tracks, so the timing for build track table also records the number of tracks present in the table. In the same way the pulse density is recorded along with the identify time. The logging is arranged so that one line in the log file captures all the events in one iteration of the main loop. The last item on the line is an exception as it is record of the system load and is not connected to a timing event. The system load is measured using the Microsoft Performance Data Helpers and in common with the compass and GPS does so once a second in its own thread. A line from a timing log is shown in Table 6.6, the entry has been split over two lines for space. The units for all items are milliseconds, apart from Density which is in number of pulses, Tracks which is in number of tracks and Load which is in percent.

6.3.6 Contributions

The author's contributions to the system software in the context of the five parts mentioned above are as follows. The author was solely responsible for the software that loads the system configuration from the configuration file and the choice of the fields in the file. This software uses the TinyXML library [38] to parse the XML. The author was solely responsible for all parts of the Main Loop except Identify Threats and Raw Data Sent Via Ethernet. The author was solely responsible for the hardware interfaces to the Pulse Characteriser and for Location and Heading. This includes the software and algorithm that calculates the AOA of the received pulses. The author was solely responsible for the software interfaces for data file parsing and replay, and the track table server. The track table server and client are based upon a example included in the Boost library documentation [39]. The author was solely responsible for the telemetry code that provides logging, performance measurement and development aids.

7 Summary of Results

The embedded software running on the Phobos sensor unit had to fulfil four high level requirements – one that ensured correct operation and three related to system performance:

1. To provide correctly formatted track tables, containing accurate information, in a reliable and robust manner to the operator.
2. The latency between a pulse being received and the corresponding track table being received by the operator shall be less than one second.
3. The pulse data throughput of the sensor unit shall be greater than one million pulses per second.
4. The error in the calculation of the angle of arrival of the received pulses shall be be no more than five degrees rms.

The first requirement ensures that the system is operating correctly, performing all of the tasks in the main loop. This includes collecting the pulse data from the RR017, analysing the pulses using the pulse identification software, summarising the results and transmitting those results, along with information about the host platform, collected from the GPS and compass, to the operator. That this requirement has been met has been demonstrated by testing the device using both synthetic and real emitters where the results were verified against a known input or the ground truth. Field testing also verified the robustness of the link between the operator and the sensor unit. In the event of one party becoming unavailable, the other will notice and the link will be automatically re-established when both parties become available again.

The second requirement ensures that the produced track tables arrive in a timely manner, allowing the operator to make best use of the information. This latency is known as the 'Antenna to Glass' time. Measuring it is difficult as it requires measurements across three devices: the RR017, the sensor unit and the operator's terminal. So as a starting approximation of the latency, the duration of the main loop in the embedded software has been used instead, with the knowledge that the actual time will be longer than that. The main loop duration averages for the five data sets collected for performance evaluation were all above one second. The shortest was 1.3 seconds and the longest was 2.6 seconds. The breakdown of the main loop task durations showed that this slowness was due to the time taken to collect the pulse data from the RR017. The data collection times for the five runs represented between 84% and 99% of the main loop duration. This requirement has not been met, but the area that requires improvement has been revealed.

The third requirement is a measure of how dense the radar pulse environment can be before the device becomes overloaded with pulse data. It is closely related to the system latency, as throughput is improved by having fewer longer periods of data collection whereas latency is improved by having more shorter periods of data collection. The five data collections performed for performance analysis had a maximum average throughput of 5046 pulses per second. This is a long way from the one million pulses per second target, but that is expected as it is very unlikely that the actual pulse density was that high at the test location during the tests. It had been hoped to calculate the actual pulse density using the timestamps in the raw data, but that was impossible due to the amount of bad data the pulse characteriser was generating. This requirement awaits further testing in the lab where arbitrary pulse densities can be created.

The fourth requirement is to ensure that the angle of arrival accuracy is consistent with what the market expects for a four channel amplitude comparison system. The major finding was that the main constraint on the accuracy was not the receiver or the antennas, but the physical design of the system housing. Initially testing showed that AOA calculation would be impossible due the amount of pulse energy scattered within the housing when using a nylon lid. Changing the material of the lid to aluminium allowed the system to be calibrated, but prevented operation of the GPS receiver. With the receiver having an amplitude tracking error of 0.5 dB the results show that the target of better than ten degrees can be achieved. The performance is noticeably better in the lower half of the frequency range than in the upper half. As the frequency increases the antenna gain patterns become more noisy with more ambiguous regions and flat spots. As the instantaneous angle error is proportional to the gradient of the gain difference curve when that difference is close to zero the error tends toward infinity. These points have been ignored in calculating the average, but the number of such points was counted and shows that the quality of the gain difference curves degrades as frequency increases due to scattering effects.

8 Conclusions

The purposes of making a prototype are to validate the design, to estimate the level of performance to expect from the finished product and to illuminate the path to that finished product. The work described in this thesis has helped Teledyne to achieve all of those goals. The embedded software that controls the sensor unit had to fulfil 4 requirements:

- To provide correctly formatted track tables, containing accurate information, in a reliable and robust manner to the operator.
- The latency between a pulse being received and the corresponding track table being received by the operator shall be less than one second.
- The pulse data throughput of the sensor unit shall be greater than one million pulses per second.
- The error in the calculation of the angle of arrival of the received pulses shall be be no more than five degrees rms.

The first requirement is by far the most important as the prototype had to be able to show that the product could work. The system has progressed from being able to identify synthetic emitters in a lab environment, to producing correctly formatted and complete track tables containing identified complex emitters in a realistic environment at Portsdown West. This shows that the prototype has met this requirement.

The second and third requirements have not been shown to be met. The maximum system throughput remains unknown due to data quality problems during the first attempt to measure it, and the system latency was found to be in excess of one second despite attempts to limit it. Timing analysis has revealed that the latency is due to the time taken to gather the pulse data from the RR017 and that the CPU is only lightly loaded. This leaves the system with significant potential for performance improvement with the existing hardware once the issues with the USB interface have been resolved.

It has been shown that the fourth requirement can be met and that the method used by the embedded software is both practical and performant. There are however problems with scattering of pulse energy within the sensor unit housing that mean that certain angles of incidence have a very high theoretical angle error which skews the result. The housing for the production models will be designed to correct this problem and the embedded software is then expected to provide the correct results unchanged.

As a product Phobos has generated interest from both Teledyne's existing customer base and new customers. The chances of success for the low cost radar warner concept appear good. Advanced militaries are looking to save money in the current financial climate, emerging militaries are looking for new capabilities, and new markets are appearing in both the private and state sectors. The fact that the first customer for the system is an advanced military gives the product more credibility in those new markets. All the while, through advances in semiconductor technology, microprocessors become ever more computationally capable and power efficient. Making RWRs and ESM systems designed around COTS processing solutions even more compelling.

The research project as a whole has furthered the knowledge of the author in many ways. The benefit of the technical modules and the chance to apply that knowledge to Phobos is shown by the technical progress made with the product. The combination of business modules and attendance at Milcom 2007 and the 2008 AOC convention clarified the commercial positioning of Phobos, showed who the customers might be and where it could take Teledyne. The chance to present Phobos to both customers and at a conference provided valuable feedback on that positioning.

9 Future Work

Future work for this product can be divided up into three areas: general tasks, tasks relating to AOA measurement and tasks relating to system performance.

9.1 General Tasks

Improving Data File Replay The current data file replay capability for using a prerecorded data file as pulse input is very basic. The file is played as rapidly as possible with no regard to how much elapsed time a data grab represents. When using a data file as the source, the system will always be able to make full data grabs until the end of the file. This is very unlikely to correspond to how the data was captured making it an unfaithful reproduction. Being able to replay a data file second for second, grab for grab, would be very useful for software development. It would allow different methods for other tasks in the system such as data deduplication or how the track table is presented to the user to be trialled and compared.

Binary data checking To minimise the chances of more problems with bad binary data a simple health check application could be written that could appraise the health of a binary data file and the incoming data stream for a debug build.

Improve Software Architecture The relationships between the `RWR` class, the `TTServer` class and the `DeviceEthernet` class are overly complicated with excessive linkage. This is due to the need to respond to commands from the operator received by the `TTServer` class that must be executed by the `DeviceDataSourceRR017`. The code to send raw data is also completely separate from the `TTServer` code relying upon fixed IP addresses rather than active connections.

9.2 Angle of Arrival Tasks

Rewrite pulse deduplication code based upon emitter tag rather than fixed bins The current deduplication code allocates fixed bins of ten degree size and assumes all pulses with the same tag in each bin belong to the same emitter. This means that an emitter near a bin boundary is very likely to appear twice in each track table making it harder for the system GUI to keep track of emitters from track table to track table. Instead of the emitters being placed into bins, the bins could be placed around the emitters. Each cluster of emitters of the same type would have a bin centred around the mean AOA and any pulses that fall into the bin would be regarded as duplicates. This would require more computation and would require care to make the result not dependent upon the order in which the pulses arrive, but should provide a more accurate representation of the radar environment.

Re-evaluate the AOA calculation performance with the new housing If the flat spots still remain, there are several options left to remove them. The housing could be filled with radar absorbent material to reduce internal scattering. The effect of smoothing the gain difference curves would still need to be determined, as would the effect of changing the search algorithm.

More realistic testing After calibrating the antennas in an anechoic chamber, the AOA calculation performance could be tested using real known fixed radars. The antennas are calibrated with the system in free space, a situation that customers may not be able to provide on their host platform. The effect of having other objects nearby on both AOA performance and identification performance could be investigated.

Investigate the effect of radar beam patterns and the impact of wrong quadrant pulses An intercepted radar might have multiple antennas which may rotate. This might result in the received pulses having a varying pulse amplitude although everything else is constant. A pulse received when slightly away from the centre of the main beam might be of low enough amplitude to have only been correctly measured on one channel for example. This would create an erroneous result assigned to the AOA of boresight for the channel where the amplitude was measured. Pulses with a single valid amplitude measurement occur often when a pulse is received with a true AOA that lies in a neighbouring quadrant to the one being listened to. One strategy to deal with these pulses would be to aggregate them with pulses that have the same emitter tag and two valid amplitude measurements. A confidence indicator would be needed in order to choose when to aggregate a single value pulse.

9.3 System Performance Tasks

Improving data collection Data collection has been found to be the dominating slow task of the system. It is not yet known whether this is due to actual slowness or the effect of sparse quadrants. If it is due to sparse quadrants, the impact might be reduced by automatically raising and lowering the grab size for each quadrant depending upon missing data in the last grab for that quadrant. If the data collection time can be reduced, then it may become possible to take advantage of threaded data grabs to improve throughput.

Logging Improvements The system log should record data grabs quadrant by quadrant with the grab size and the amount of missing data if there is any. C++ 11 includes functionality for measuring time including using a high resolution clock if the system has one. This is available in the standard library as `std::chrono` and would be good basis for the improvement of the time measurement code.

Bibliography

[1] Boot H, Randall J. The cavity magnetron. Journal of the Institution of Electrical Engineers-Part IIIA: Radiolocation. 1946;93(5):928–938.

[2] Bauer AO. Naxos, The History of a German Mobile radar detection Finder 1943 - 1945; 2004. Foundation for German communication and related technologies. Available from: `http://www.cdvandt.org/naxos.htm`.

[3] Stimson GW. Introduction to airborne radar. SciTech Pub.; 1998.

[4] Skolnik MI. Radar handbook. McGraw-Hill (New York); 1990.

[5] Wiley RG. ELINT The Interception and Analysis or Radar Signals. Artech House; 2006.

[6] D'Agostino S, Foglia G, Pistoia D. Specific Emitter Identification: Analysis on real radar signal data. In: Proc. European Radar Conf. EuRAD 2009; 2009. p. 242–245.

[7] Rogers JAV. ESM processor system for high pulse density radar environments. IEE Proceedings F Communications, Radar and Signal Processing. 1985;132(7):621–625.

[8] Turner SP. A parallel processing solution to ESM. In: Proc. IEE Colloquium Signal Processing for ESM Systems; 1988. .

[9] Roe J, Cussons S, Feltham A. Knowledge-based signal processing for radar ESM systems. IEE Proceedings F Radar and Signal Processing. 1990;137(5):293–301.

[10] Maier MW. Processing throughput estimation for radar intercept receivers. IEEE Transactions on Aerospace and Electronic Systems. 1998;34(1):84–92.

[11] Chan YT, Chan F, Hassan HE. Performance evaluation of ESM deinterleaver using TOA analysis. In: Proc. MIKON-2002 Microwaves, Radar and Wireless Communications 14th Int. Conf. vol. 2; 2002. p. 341–350.

[12] Hassan HE. A new algorithm for radar emitter recognition. In: Proc. 3rd Int. Symp. Image and Signal Processing and Analysis ISPA 2003. vol. 2; 2003. p. 1097–1101.

[13] Wikipedia. BNS Bangabandhu;. Available from: http://en.wikipedia.org/wiki/BNS_Bangabandhu.

[14] Wikipedia. Barbaros Class Frigates;. Available from: http://en.wikipedia.org/wiki/Barbaros-class_frigate.

[15] Friedman N. US Naval Institute Guide to World Naval Weapons Systems, 1997–1998. US Naval Institute Press Annapolis, Maryland; 1997.

[16] Northrop Grumman Corporation. AN/ALR-93(V)1 Radar Warning Receiver/Electronic Warfare Suite Controller;. Available from: http://www.northropgrumman.com/Capabilities/ANALR93/Documents/ALR-93.pdf.

[17] Rockwell Collins. CS-3600 Tactical Surveillance System;. Available from: http://www.rockwellcollins.com/~/media/Files/Unsecure/Products/Product%20Brochures/EW-Intelligence/SIGINT/CS-3600/CS-3600%20data%20sheet.aspx.

[18] Selex Galileo. Sky Guardian 2500 Compact and Lightweight ESM. 300 Capability Green, Luton, Bedfordshire, LU1 3PG, United Kingdom; 2010. Available from: http://www.selex-es.com/documents/737448/4748929/body_mm07739_SEER_LQ_.pdf.

[19] Selex ES. SEER a digital RWR for modern fighters,. 300 Capability Green, Luton, Bedfordshire, LU1 3PG, United Kingdom;. Available from: http://www.selex-es.com/documents/737448/4748929/body_mm07739_SEER_LQ_.pdf.

[20] Thales. CATS Compact airborne threat surveyor for helicopter. 2 Avenue Gay-Lussac, 78851 Elancourt Cedex, France;. Available from: https://www.thalesgroup.com/sites/default/files/asset/document/cats_helicopter.pdf.

[21] Holt O. Technology Survey Sampling of RWR/ESM Systems. The Journal of Electronic Defense. 2009 February;32(2):39–48.

[22] Thales. VIGILE LW The fully automatic R-ESM System for small and Offshore Patrol Vessels. Manor Royal, Crawley West Sussex, RH10 9HA;. Available from: https://www.thalesgroup.com/sites/default/files/asset/document/bat_fiche_vigile_lw_2_-_a4.pdf.

[23] Indra Security and Defense. ALR-400 Radar Warning Receiver. Carretera Loeches 9, 28850 Torrejon de Ardoz, Madrid, Spain;. Available from: http://www.

indracompany.com/sites/default/files/alr-400_1.pdf [cited 14 November 2013].

[24] Saab Electronic Defence Systems. Surface Tactical ESM and ELINT Systems. PO Box 39347, 7948 Capricorn Square, Cape Town, South Africa;. Available from: http://www.saabgroup.com/Global/Documents%20and%20Images/Naval/Electronic%20Warfare/SME100/SME100%20-%20SME200%20product%20sheet.pdf.

[25] Teledyne Defence Ltd. Technical Datasheet RR017. Airedale House, Shipley, West Yorkshire, UK, BD17 7SW; 2011. Rev 3.0. Available from: http://www.teledynedefence.co.uk/pdf/RR017_Teledyne.pdf.

[26] Selex ES Ltd. HIDAS Helicopter Integrated Defensive Aids System. 300 Capability Green, Luton, Bedfordshire, LU1 3PG, United Kingdom; 2013. Available from: http://www.selex-es.com/documents/737448/19890350/body_mm07737_HIDAS_LQ_.pdf.

[27] Royal Air Force. RC-135W Rivet Joint;. Available from: http://www.raf.mod.uk/equipment/rc135wrivetjoint.cfm.

[28] Thales UK. Thales to upgrade Royal Navy fleet with new digital UAT system; 2012. Available from: https://www.thalesgroup.com/en/content/thales-upgrade-royal-navy-fleet-new-digital-uat-system.

[29] naval-technology com. SSN Astute Class Nuclear Submarine, United Kingdom;. Available from: http://www.naval-technology.com/projects/astute/.

[30] Advanced Digital Institute. Salts Mill, Victoria Road, Saltaire, West Yorkshire, BD18 3LA;. Available from: http://www.adi-uk.com/.

[31] Teledyne Defence Ltd. PHOBOS General Press Release; 2011. Available from: http://www.teledynedefence.co.uk/pdf/pressreleases/Phobos.pdf.

[32] Teledyne Defence Ltd. Technical Datasheet QR020-M1. Airedale House, Shipley, West Yorkshire, BD17 7SW, UK; 2010. Rev 1.0. Available from: http://teledynedefence.co.uk/pdf/QR020-M1.pdf.

[33] Teledyne Defence Ltd. PHOBOS Low Cost Tactical Threat Warning System QR020. Airedale House, Shipley, West Yorkshire, UK, BD17 7SW; 2009. Available from: http://teledynedefence.co.uk/pdf/PHOBOS_QR020%20DS_Teledyne.pdf.

[34] Cypress Semiconductor. EZ-USB Technical Reference Manual. Cypress Semiconductor, 198 Champion Court, San Jose, CA 95134-1709; 2011. Available from: http://www.cypress.com/?docID=27095.

[35] Compulab Ltd. FitPC 2 Datasheet. 17 HaYetsira Street, Moradot HaCarmel Industrial Park, Yokneam Elite, Israel 20692;. Available from: `http://www.compulab.co.il/fitpc2/html/fitpc2-sb-datasheet.htm`.

[36] Compulab Ltd. iGLX Datasheet. 17 HaYetsira Street, Moradot HaCarmel Industrial Park, Yokneam Elite, Israel 20692;. Available from: `http://compulab.co.il/products/sbcs/sbc-iglx/`.

[37] G Coley, C Cooley and J Kridner. Beagle Board and Beagle Bone Community. 1380 Presidential Dr, Suite 100, Richardson, TX 75081-2437;. Available from: `http://beagleboard.org/`.

[38] L Thomason, Y Berquin and A Ellerton. TinyXML a simple, small, C++ XML parser that can be easily integrated into other programs.;. Available from: `http://www.grinninglizard.com/tinyxml/`.

[39] Kohlhoff CM. Boost Asio Library;. Available from: `http://www.boost.org/libs/asio`.

Phobos

The Design and Implementation of Embedded Software for a Low Cost Radar Warning Receiver

Simon Brown

December 2014

VOLUME TWO OF TWO

A portfolio thesis submitted to the Universities of

Edinburgh

Glasgow

Heriot-Watt and

Strathclyde

In fulfilment of the requirements for the degree of

Doctor of Engineering in System Level Integration

Institute for System Level Integration

Contents

[1] This paper has been reformatted for inclusion in this thesis. The original can be found on the ARMMS website [1]

List of Figures

List of Tables

Nomenclature

AES	Advanced Encryption Standard
AOA	Angle of Arrival
AOC	Association of Old Crows
ARMMS	Automated RF and Microwave Measurement Society
boresight	Antenna boresight is the axis of maximum gain of a directional antenna
deinterleaving	The attribution of individual pulses to the pulse trains of different emitters.
DOD	Department of Defense
EAP	Extensible Authentication Protocol
ELINT	Electronic Intelligence
EW	Electronic Warfare
Free Software	Free software is software that gives the user the freedom to share, study and modify it. http://www.fsf.org
HF	High Frequency
LPI	Low Probability of Intercept
LTE	Long-Term Evolution
MANET	Mobile Ad hoc Network
mutex	Mutual exclusion. A software lock that prevents more than one thread from accessing a section of code concurrently

NLOS	Non Line Of Sight
PDW	Pulse Descriptor Word
PIR	Passive Infrared
RF	Radio Frequency
rms	root mean square
TDL	Teledyne Defence Limited
TOA	Time Of Arrival
UAV	Unmanned Aerial Vehicle
UMB	Ultra Mobile Broadband (EV-DO Rev. C)
UMTS	Universal Mobile Telecommunications System

10. Introduction

This volume contains all of the papers and reports that were written by the author during the research phase of the project. Unlike in the previous volume each report, paper or chapter in this volume is a standalone document that reflected the situation at the time that it was written. This is a situation that may have changed by the time that the research phase was complete. This introduction is intended to help place each report into the chronology of the project so that the motivation for writing it, the problems found and the solution chosen can be more easily understood.

Angle of Arrival Calculation for Electronic Warfare Receivers This chapter documents in detail the work done to measure the antenna gain of each antenna in the prototype across the operating frequency range. The antenna manufacturer had supplied calibration data for each antenna in isolation and until this phase that data was being used whilst the system was being developed. No attempt had been made to quantify how realistic that data was. The first trips to the anechoic chamber were at the start of 2010 and it soon became clear that system calibration would not be straight forward. After many trips to the anechoic chamber we had a rough calibration for the prototype and enough information to guide the redesign of the housing that was done in the summer of 2010 for the first production models.

Performance Considerations This chapter documents the work done to make the first detailed measurement of the system's performance. Up until this point performance had only been judged by eye to see that track tables were arriving roughly every second and only major problems such as getting data from slow GPS and compass devices had been dealt with. The data was collected in August 2010 but not analysed in detail until 2011. The field trip during which the measurements were taken had been mostly devoted to development work rather than measurement as the pressure of the first customer delivery was starting to mount. It had also been assumed that the radar environment and the types and number of both known and unknown emitters would play a big part in determining the system performance making lab testing unrealistic. The initial analysis of the results was enough to confirm the suspected problems with the RR017 firmware in time for rectification for the first production units.

Publications The first paper, Design and Realization of Linux Based Wireless Data Acquisition System, was published at the instigation of our colleagues at the Wireless CIC. The paper was presented at NIMC 2008 held between the 22^{nd} and the 24^{th} of July in Glasgow. It was published in the journal Communications of SWIN Volume 4 in 2008. This paper documents work done on the concept demonstrator for the radar sensor network before the project changed into the system integration work for Phobos.

The second paper, Antenna Characterisation for Amplitude Comparison in Electronic Warfare Systems, was presented at the Autumn conference of the Automated RF & Microwave Measurement Society (ARMMS) in 2010 at the end of the project and published in their proceedings. Due to the applied nature of the research project the work was unsuitable for publication in the more theoretical academic radar journals, whereas the ARMMS is very much interested in practical details. The event was small enough to encourage in depth discussion of challenges and the attendees came from a wide range of applied backgrounds ranging from test and measurement companies to motor racing organisations.

Conferences Milcom is the leading annual conference on military communications. It has strong backing from both the IEEE and the major defence companies. The 2007 conference was held in Florida and attracted several four star generals as well as the state governor as invited speakers. This conference was chosen for attendance in order to appraise the state of the art in military data links through what was being shown on the exhibition floor and what was being presented in the technical sessions

The Association of Old Crows Annual Convention is the major annual conference and exhibition devoted to electronic warfare. The 2008 convention was held in Nevada and unlike Milcom it did not have a strong academic backing, instead it just focused on the US military, their allies and their equipment suppliers. The research project had changed by this point from sensor data links to Phobos and Teledyne had intended to show the Phobos concept on their stand at the exhibition. This convention was attended in order to improve the author's understanding of the market for electronic warfare products and to become more familiar with the electronic warfare community.

Internal Reports The Data Link Strategy report was written in the Autumn of 2007 to document what had been learned about sensor data links. Filtronic Defence had recently appointed a new Director of Engineering and this coincided with his review of existing projects. The research project was changed to the system integration work for Phobos as a result of this review.

The Cypress EZ-USB FX2 Firmware and HDL customisation report was written at the start of 2008. Production of the RR017 had been delayed and a realistic source of data was needed for both the Wirless CIC project and Phobos system development. The USB interface also had to be proved, to show that it could handle the high data rates that the RR017 was expected to provide.

The Version Control Systems and Defect Trackers report was written in the Spring of 2008 as a motivation document for upgrading the software development infrastructure at Filtronic Defence. The existing version control system did not encourage collaboration and there was a need to better manage change requests for the production test software. It also coincided with the introduction of new software in the company for requirements management.

11. Angle of Arrival Calculation for Electronic Warfare Receivers

11.1. Introduction

For RWR systems to be able to determine the angle of arrival (AOA) of incident pulses has long been an important requirement. The AOA, as well as being used to locate the emitter, is also often used in identifying the emitter. A common technique of identifying emitters is to compare the received pulse train from an observed emitter against a library of known pulse trains. Before this can be done, the received pulses have to be reassembled into their original pulse trains, distinguishing the pulses from the different emitters that may be present. One way to do this is to sort the incoming pulses on the basis of AOA. This is because unless the relative velocity of the emitter and the RWR is great, pulses from the same emitter will come from the same direction. This process of attributing individual pulses to the pulse trains of different emitters is called deinterleaving.

The extra cost of generating this AOA information is dependant upon the AOA information itself being useful to the user. ESM systems require high accuracy AOA information to allow them to precisely match the pulse data with other intelligence. So having to generate it for deinterleaving purposes adds no extra cost. This may not always be true for RWR systems – a coarse indication of direction may be all that is required or perhaps all that can be fully utilised.

AOA can be derived from pulse data in a number of ways. The simplest method is to use a rotating directional antenna. When the received signal strength is at a maximum the antenna is likely to be staring at the emitter. Such an antenna would however add greatly to the size, weight and power (SWaP) requirements making it unsuitable for SWaP sensitive or fast moving platforms. Alternatively several antennas could be used and AOA calculated by making use of how different antennas receive the same pulse. This could be the difference in phase due to spatial separation of the antennas or the difference in amplitude due to the directionality of the antennas. Phase comparison systems have the advantage of excellent accuracy, being able to achieve an accuracy of less than one degree. Their disadvantages are size and cost. Phase comparison requires several antennas at each antenna site to resolve ambiguity across the frequency range and the antennas and their cable assemblies have to be phase matched which significantly increases the cost. The increased number of antennas makes them larger which makes them far more difficult to retrofit to existing platforms. This is especially true for platforms with stringent aerodynamic requirements. Amplitude comparison systems are much lower cost but are commonly advertised as only having an accuracy of ten degrees rms.

The RR017 pulse characteriser which Phobos has been created around was designed as a low cost product and accordingly it echews the extra complexity and cost required for phase comparison. It has four external ports each attached to an antenna for three hundred and sixty degree azimuth coverage. There are only two internal measurement channels which are switched between the four ports. For each pulse the amplitude of the pulse envelope is measured on both channels which can then be compared. To create the Phobos system cavity backed spiral antennas were chosen for several reasons. They are compact, easily available for the frequency ranges of interest and have a monotonic fall off in gain either side of bore sight for angles of interest. This antenna gain characteristic greatly simplifies amplitude comparison.

As the emitter identification software used by Phobos does not use AOA to assist in identifying emitters, the AOA calculation accuracy does not affect emitter identification performance and need only be as good as required by the operator. Designing the system around amplitude comparison using a low cost pulse characteriser and low cost antennas allows the system cost to be greatly reduced. This makes the system very attractive in scenarios where precise AOA information is not required and may open up more cost sensitive platforms to RWR systems. The rest of this portfolio document describes how AOA is calculated by the Phobos system and compares the results to the ten degree target expected by the market.

11.2. Method

To minimise the amount of calculation required during operation, AOA is calculated using a look up table which is generated at program start up. The program requires a table of amplitude response for each antenna in azimuth, in one degree and one GHz steps. This is then converted into a table of expected amplitude difference for each frequency for every AOA in one degree and one GHz steps. Then during operation AOA calculation is a two stage process. Firstly the quadrant that is currently being observed is determined and the amplitude difference calculated with the results passed to the next stage. The second stage is a simple search from both ends of the look up table for a matching amplitude difference value. Ideally both searches will find a match at the same place and that result, once compensated for quadrant position, could be used as the result. Searching from both ends would actually be unnecessary duplication in this case. Two searches are done – one from each end of the table – to try to reduce the impact of amplitude measurement inaccuracy. The searches are done with a defined match tolerance and each search will declare a match only when the value in the difference table matches the measured value within the tolerance. If the two searches have declared a match, but not at the same location, the average of the two results is used.

Problems arise when the amplitude of the recorded pulse was above the noise floor on only one channel. This could be due to receiving a weak signal from a distant emitter or the emitter not being in the quadrant currently being observed. As that pulse may belong to an emitter that represents a grave and immediate threat it was decided not to discard the pulse as bad data. Instead, before the AOA look up is performed, the input is checked to make sure that both values are above the noise floor of the system. If they are not, the pulse is given the AOA that corresponds to boresight on the antenna which measured a valid amplitude. There is also the possibility that if the gain difference is too flat or misshapen, or an inappropriate difference tolerance was chosen, the two searches might match at very different points. To limit the possibility of this, the difference between the two successful search results is compared against the chosen angle tolerance. If the result is greater than the chosen tolerance, the AOA is assigned to boresight of the strongest antenna as before.

11.3. Expected Results

The antennas were supplied with performance data measured by the manufacturer. This data was collected from each antenna in isolation at the manufacturer's test facility and was used as the initial calibration until a system calibration could be performed. That data is used here to illustrate the method and it was also used to set initial expectations

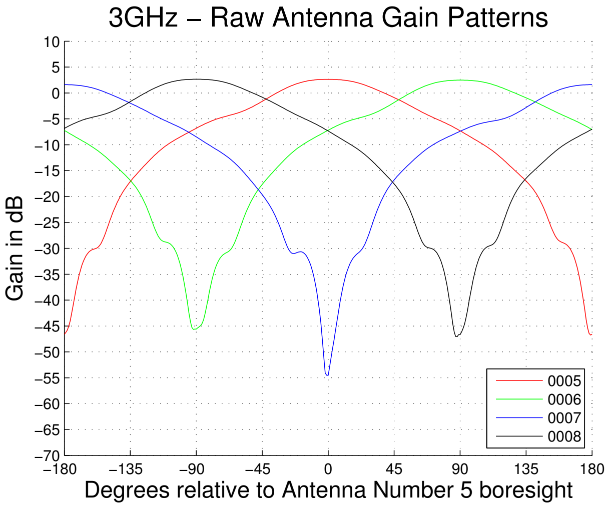


Figure 11.1.: Manufacturer Provided Performance Data for Each Antenna of the Four at 3GHz Overlaid

of system performance.

In Figure 11.1 the antenna gain information for the four antennas is shown on the same graph with the sets of data transposed so that each antenna is ninety degrees apart. Antenna number five is notionally pointing North, six – East, seven – South and eight – West. The curves are very similar with the gain monotonically decreasing either side of boresight until the ninety degree region opposite boresight. The principal point of concern is the shoulder that each antenna has in the region seventy degrees anticlockwise of boresight.

In order to calculate the AOA of a pulse it is assumed that the emitter is far enough away and that any variation in received pulse amplitude at the antennas is solely due to the directionality of the antennas. This is shown in the gain difference plot shown in Figure 11.2. The unusual names for the quadrants are due to the data they represent. The difference is always calculated by subtracting the value received from the antenna ninety degrees away in a clockwise direction. Hence the North East gain difference is the North value minus the East value and the East South gain difference is the East value minus the South value. The effects of the problematic shoulder region can be seen here in the increased flatness at the start of each curve. The RR017 has an average amplitude tracking error between the two channels of 0.5 dB for signals at the same power level and 1.5 dB for signals up to twenty dB apart [2]. Once the system has been calibrated, this amplitude tracking error is assumed to be the only remaining source of error. The

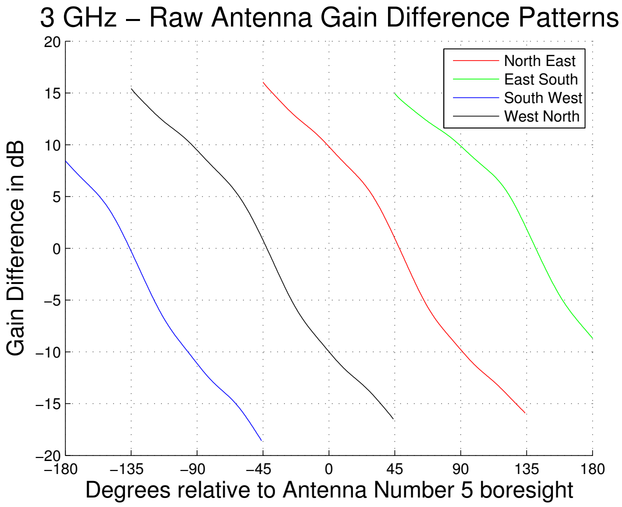


Figure 11.2.: Gain Difference Plot for the Overlaid Antenna Patterns

steeper the gradient of the amplitude difference curve, the less effect a 0.5 dB error has. This is shown in Figure 11.3. The graph shows that the point of least error for each antenna pair is always approximately at the forty five degree point halfway between the two and the worst region is just off boresight at the flattest part of the shoulder.

As frequency increases the main beam of the antenna becomes narrower and the antenna more directional. As amplitude tracking error is not specified as being frequency dependant, this improves the AOA accuracy as the tracking error will have a smaller impact. This is shown in Figure 11.4.

11.4. Initial Experimentation

The Phobos prototype shown in Figure 11.5 (copyright Teledyne Defence Ltd) was designed to be compact and quick to produce. This meant that almost all of the case work was made from aluminium as this is the material TDL usually uses for cases. The only exception was the lid which was made using a rapid prototyping technique out of nylon. This was to allow the GPS unit visible on top of the aluminium boxes to function. As TDL did not have its own anechoic chamber, all of the early calibration work was carried out in facilities at Bradford University. It was hoped that the most challenging part would be automating the data measurement process as data needed to be collected for each antenna, at seventeen frequencies, in one degree steps. However initial data collection showed there were serious problems.

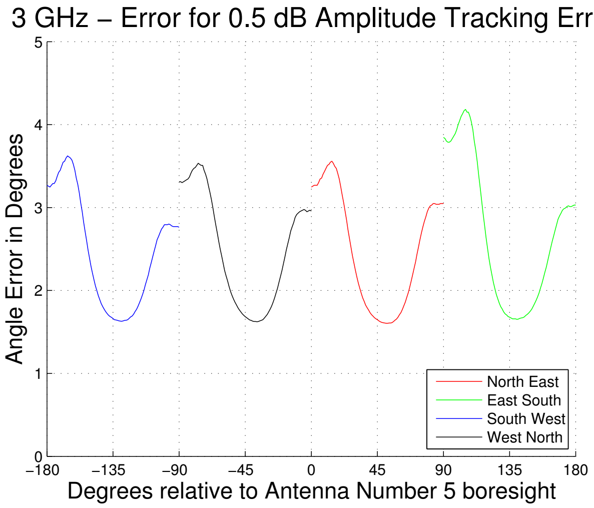


Figure 11.3.: Angle Error for 0.5 dB Amplitude Tracking Error for the Overlaid Antenna Patterns at 3Ghz

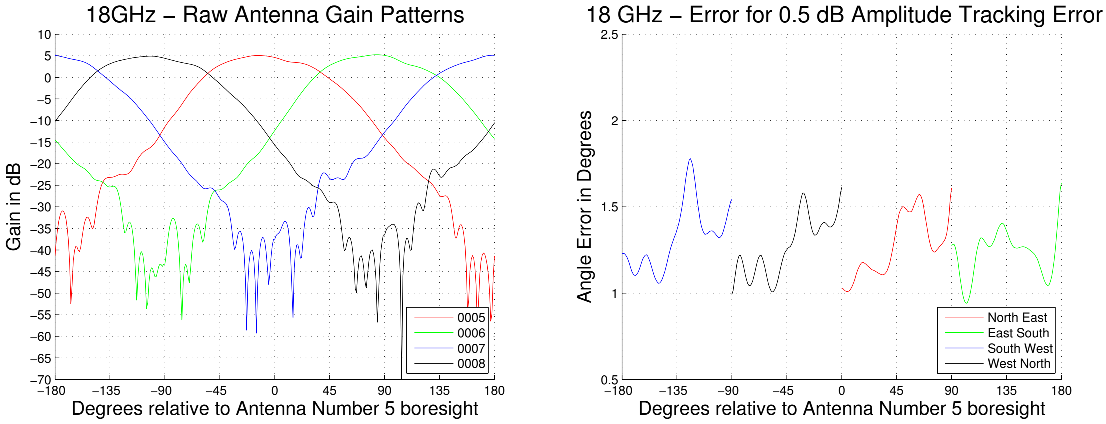


Figure 11.4.: Antenna Patterns and Angle Error at 18 GHz

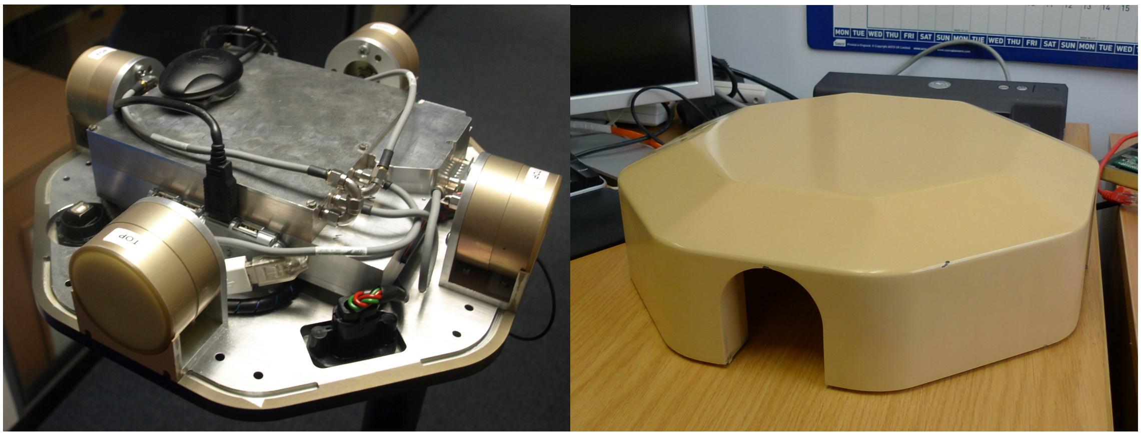


Figure 11.5.: Phobos Prototype With Lid Removed

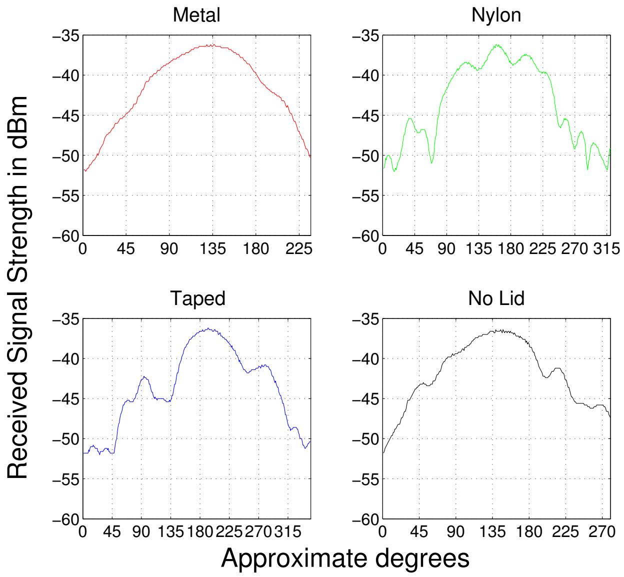


Figure 11.6.: Lid Performance Comparison

The gain fall off either side of boresight was no longer monotonic, which would make AOA calculation much more difficult. The results were discussed with the RF engineers at TDL and they advised that the modulation on the received signal strength was likely due to scattering and reflection within the case. As a possible remedy they suggested changing the lid to a metallic one that was electrically connected to the base plate of the case. An aluminium lid was made to the same design as the nylon lid and the vertical walls of a second nylon lid were lined with copper tape in an attempt to see if the scattering and reflection could be minimised whilst still allowing the GPS to function. The performance of the three lids is shown on Figure 11.6. Clearly the only usable result is that with the metal lid. Interestingly, the results with no lid were better than both the taped lid and the nylon lid.

11.5. Results

Measurements were collected at all frequencies from 3 to 17 GHz except 7 GHz. Two GHz could not be used due to interference from other emitters presumed to be mobile communications. At 18 GHz the RF source could not deliver enough power for a complete data set and the 7 GHz data set was incomplete.

The results for 3, 10 and 17 GHz are shown in Figure 11.7, along with overall results. As

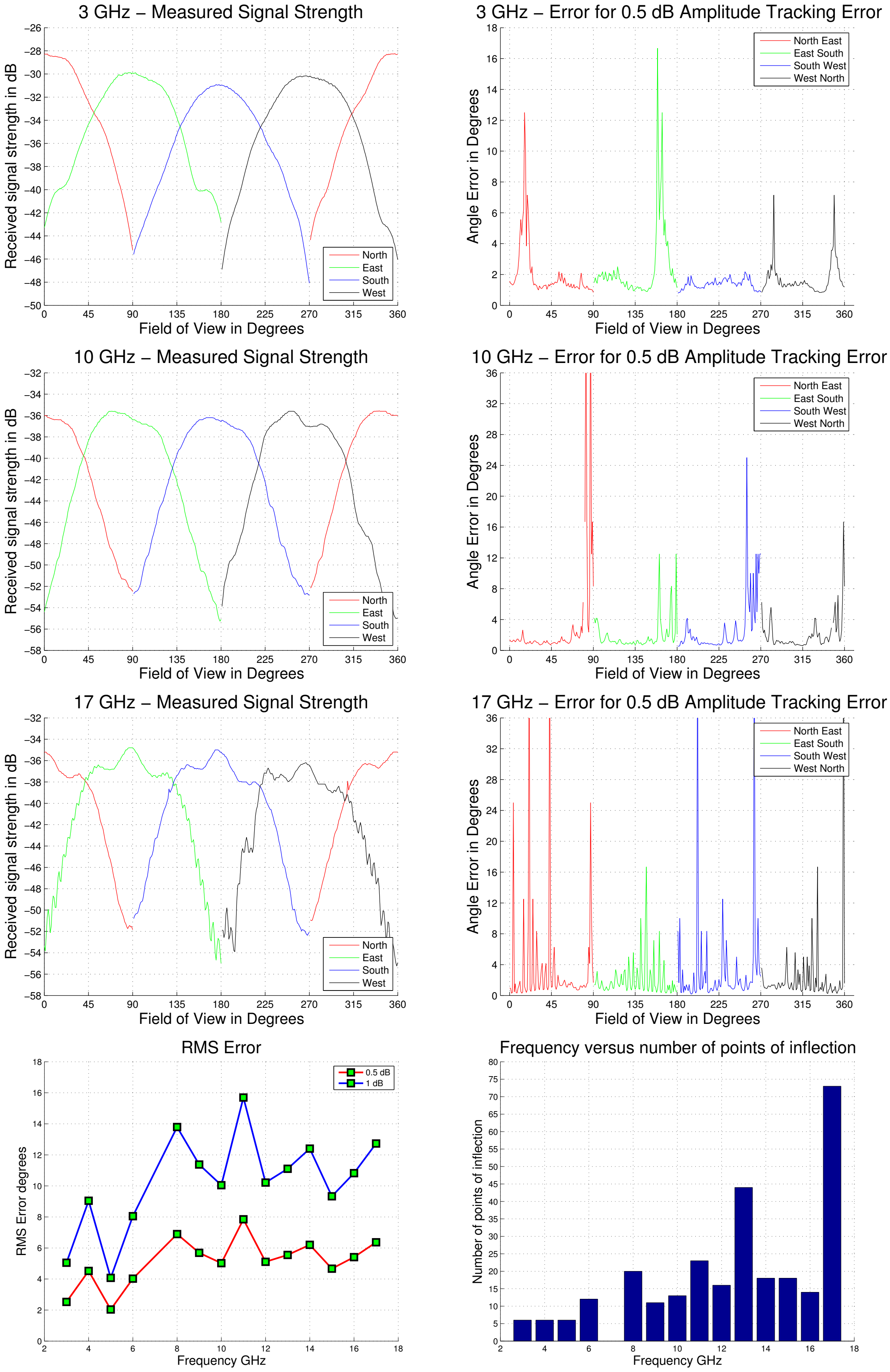


Figure 11.7.: Antenna Patterns and Angle Error

the frequency increases, the measured signal strength curves start to deviate from the expected smooth curves. They develop flat spots and jagged sides which are problematic. At each point where the gradient difference is zero the angle error will tend toward infinity and be very large for values close to that. This led to problems in calculating the average error. Should the entire data set be disregarded due to one instance of the error being undefined or could the number of such instances be used to indicate quality? At the bottom of Figure 11.7 the average rms error is shown for each frequency as well as the number of points discarded due to having zero gradient difference. The rms error is reasonably constant from 8 GHz to 17 GHz, but the number of discarded points is on an upward trend over this frequency range reaching a peak at 17 GHz where twenty percent of the data points have been discarded.

11.6. Conclusions

The results show that the target of ten degrees rms error across the desired frequency range with an amplitude tracking error of 0.5 dB is a realistic target for a production device. They also show that the case plays a significant role in limiting the achievable accuracy and that further work is required upon the case design as the product moves from the prototype stage to being a saleable item.

The method used to calculate the AOA described in section 11.2 relies upon two valid amplitude measurements. The results above show that the difference in antenna gain for two antennas ninety degrees apart can be up to 19 dB. So for an emitter at 17 GHz that is staring straight at one antenna the received signal strength would need to be at least 19 dB above the noise floor at that antenna before the signal would be detectable on either antenna ninety degrees away allowing a valid second amplitude to be measured. Emissions of interest to EW receivers tend to come from distant emitters and as a consequence be low power. Although this has no effect upon identification, the consequences of loss of AOA accuracy and the possible creation of ghost emitters requires further investigation.

11.7. Further work

The accuracy of the AOA calculation can be improved in several ways depending upon customer interest. These improvements can be divided up into three areas: improvements to the case design, improved search algorithms and better testing. Specific tasks for each section have been given where appropriate along with general directions.

All work done so far on AOA performance has been focused on measuring and addressing problems with the case design. The original case design prevented AOA calibration as too much pulse energy was being scattered around within the case, which made the measured amplitude values ambiguous. Consequently the case design for the Phobos demonstrator has been modified in the hope of reducing this scattering for the first production standard units. The performance of the new design should be compared to that of the old design to check that the scattering has been reduced to point that the antennas can be calibrated. Along with the case redesign the cost and practicality of filling the case, or surrounding the antennas within the case, with radar absorbent material should be investigated. Even if the benefit is small, if AOA calibration is borderline with the new case, it may be enough to allow calibration.

Due to the focus on the case, the performance of the AOA calculation software has not yet been properly evaluated. This could be done with some computer generated input data that represents one or more radar emitters that move in known ways. This would also be the first step in building a full system input simulator. Noise could also be added to the amplitude values to better replicate real world conditions. The search algorithm used is very simple, but whether it successfully settles on an AOA value has a great impact on the rest of the system. There are two factors which decide whether a match is declared: the angle tolerance and the gain difference tolerance. The gain difference tolerance is how close the value has to be to a table entry before the table search stops in that direction. The angle tolerance is how close the the two table searches have to be to each other before a match is declared. Other search algorithms could be evaluated for both the time that they take to reach an result and the accuracy of that result. As well as the AOA value the algorithm could output a confidence value for the result. This would allow the user interface application to better combine duplicates and to recognise spurious results.

- Investigate the effect of the two tolerance values using synthetic data with the aim of discovering optimal defaults.
- Re-implement the search with a different algorithm such as binary search and measure the speedup.
- Add an AOA confidence value to indicate whether the pulse has two valid amplitude values and whether a match was successful.

The AOA calculation software has been tested in anechoic chambers using signal generators as it was impractical to use a real emitter. A signal generator transmitting a continuous waveform is a poor approximation of a radar and the differences between them need to be better understood. As the radar scans around the system will be swapping between its four quadrants and depending upon the respective speeds of rotation there may be strange effects. For example the system might receive pulses from the main lobe of the radar then pulses from a side lobe. If the weaker side lobe pulses only have one valid amplitude value the AOA calculation may fail resulting in the system reporting two emitters with different AOAs when there is only one, or the emitter may appeared to have moved. If the system were to be tested in an environment with a known ground truth, these anomalies would reveal themselves and the experience could be used to improve the realism of generated synthetic pulse data.

- Test the system in a busy maritime environment where the ground truth can be recorded from AIS[1] broadcasts. Measure the AOA accuracy and note any anomalous results.

[1] Automatic Identification System. A radio beacon that ships above 300 gross tons and all passenger ships must broadcast that includes their position and heading.

12. Performance Considerations

12.1. Introduction

An accurate measurement of the system performance is required to better understand how the system can be improved and to predict whether the system will be able to satisfy the performance targets set for it. In the product brochure, which is included with this volume as Appendix A there are several system performance parameters listed to show prospective customers that Phobos will be able to identify their targets of interest. The targets of interest for each customer along with the expected usage scenario are generally a closely guarded secret, which means that the brochure instead has to suggest system performance levels through technical system parameters rather than list detectable emitter types. The prospective customer will then compare these parameters against what they believe would be required by a radar identification algorithm to identify their targets of interest. Traditional radar identification methods can be classified as *interval-only* or *multiple-parameter* [3]. Interval-only techniques use just the TOA difference between pulses to discover emitter patterns whereas multiple-parameter techniques also make use of other pulse information. A complete description of these methods can be found in Chapter 13 of Richard Wiley's book [4] and in several published papers [5, 6, 7]. Interval-only techniques have the advantage of having the entire pulse population available for analysis, but this creates a large pulse processing burden in dense radar environments. By grouping pulses using other pulse parameters the processing power requirements can be reduced, but if the group boundaries are chosen badly, important pulses may be disregarded. Armed with the knowledge of emitter identification method and pulse characterisation accuracy the customer can then extrapolate if the system might be suitable for their needs.

This established system of performance measurement is a disadvantage for Phobos in two ways. Firstly, the pulse identification software has only recently been developed and the performance that it can deliver with its minimal set of inputs is not yet widely known. Secondly, Phobos is designed to be a low cost system which means that certain features which may be ubiquitous on other systems have not been included or have been reduced, as they are not as important in this system. The precision of the AOA measurement is a good example of such a feature. AOA is often used as a grouping criterion for radar pulses. Pulses which have come from the same direction are more likely to have been emitted by the same emitter than ones that have come from wildly different directions. The AOA measurement accuracy for Phobos is specified as better than ten degrees. For the comparable CS-3600 from Rockwell Collins [8] it is better than five degrees and for the Meerkat [9] system from Thales it is better than 1.5 degrees. High accuracy AOA calculation is normally done with either a highly directional dish antenna or by using phase comparison. Choosing either over simple amplitude comparison will add to the cost and the bulk of the system. As Phobos does not depend upon AOA accuracy for emitter identification this extra cost and bulk becomes optional, allowing the system to be used in many more roles and on many more platforms. Although a disruptive product Phobos still has to be broadly comparable with the existing products in order to be credible so that it can establish itself in the market.

A lot of the performance data items in the brochure come from the RR017 which was designed as a stand alone product before Phobos was conceived and although a low cost product it still offers very high performance. Of the remaining performance data items in the brochure, this document will focus on System Response Time and Environment Pulse density. The RR017 can deliver pulses at a very high rate to the rest of the system and these two items measure the latency and the throughput of the system. These parameters are not independent as when there are more pulses present it will take longer to process them and when processing takes longer there is a risk that pulses might be dropped if the output buffer of the RR017 becomes full. They are also completely determined by the performance of the pulse processing software and the hardware that it runs upon as the inputs and output of the system are high capacity. As well as the sales perspective, being able to measure the system's performance is vital from a technical perspective. Without measured evidence of which parts of the software process are quick and which are slow it is very difficult to improve performance or predict future performance.

CPU	Intel Atom Processor: Z530
Memory	1 GB DDR2
Network	1 Gigabit Ethernet interface
Storage	SATA-II interface
USB Ports	6
Weight	90 g
Power Consumption	5 W
Operating Temperature	Industrial: -40° to 85° C

Table 12.1.: fitPC2 Specifications

12.2. Single Board Computer

ARM architecture processors have long been dominant in the embedded market, but compared with desktop PC processors, although they are more power efficient, there was a significant performance divide. With the success of the Apple iPhone this has changed. There is suddenly great interest in higher performance ARM architecture processors and a number of processor vendors, such as Texas Instruments and Freescale, have made low cost development boards for their processors available. The Beagle board from Texas Instruments [10] with its single core ARM Cortex A8 processor and low size weight and power consumption was a natural fit for a low cost, low power product.

Using such a board did not however fit with the rest of the project. Teledyne has experience using single board computers that run Microsoft's operating systems from past projects and a stock of x86 architecture single board computers. Using PC hardware also allowed the use of standard removable flash storage devices and eliminated any concerns about the board being able to supply enough power to its USB ports. When it became clear that the pulse identification software would only be made available in a form compiled for x86 Microsoft Windows it was clear that an ARM architecture processor could not be used. The decision was taken to use the fitPC2 from Compulab [11] which, although intended as a home theatre PC, was available in an industrial variant, designed to operate over a much wider temperature range. Brief specifications of the fitPC2 are shown in Table 12.1. The embedded application software was designed to be platform independent where possible with only the emitter identification software and the USB access library being platform specific. This is to allow the single board computer to be easily changed should the need arise in the future.

12.3. Software Architecture

The Phobos application, which runs on the single board computer, collecting pulse descriptor words and generating track tables was originally designed around the flow of control. In this design the data for each iteration of the main loop is passed from task to task and transformed as required by each stage. This was chosen as it is simple and it fitted in well with how the class hierarchy had been constructed.

Initial testing however, showed that the performance was poor and more importantly the code proved difficult to extend as at any point in the main loop, only the data that was required was guaranteed to be available. This meant for example that all pulse data had to be completely pre-processed to parametric form before being passed to the Identify Threats task as afterwards the binary data was unavailable, making selective processing based on emitter identification impossible. This was temporarily overcome with extra linkage between the classes but this made the code more complex. To remedy this, the design was turned on its head. Instead of the data being arranged around the control flow, the control flow was arranged around the data. The aim was to improve performance by minimising the amount of a data manipulation and the number of copies, whilst making the data more accessible. The data is now kept in two classes, `DataSource` and `TrackTable`, where all tasks within the class have full access to the data, but the data is not available to other classes. A simplified version of the main loop is shown in Figure 12.1 along with the class boundaries and the data held.

The `DataSource` holds the binary pulse data and the parametric data which is an array of C structures where each element can hold the data from a single PDW. On arrival the binary data is saved to file and the padding required to align with USB packet sizes is removed. The pulse processing software only requires the data from a small number of the fields in a PDW. This data is extracted and saved both in parametric form and the required data structure for Identify Threats, with the binary data only being walked once. After Identify Threats the emitter tags need to be saved with their corresponding pulse in the parametric data. As this is done the remaining information is extracted from the binary data and secondary data such as the angle of arrival is calculated for pulses of interest. The pulses are then placed in bins according to their emitter tag and AOA.

The `TrackTable` class has to transform the data from parametric form to XML text. Pulses with the same tag that are in the same bin are regarded as duplicates and need to be condensed into one whilst preserving the extra information of all the pulses that were present. At present the values are averaged and the mean value is stored along with the

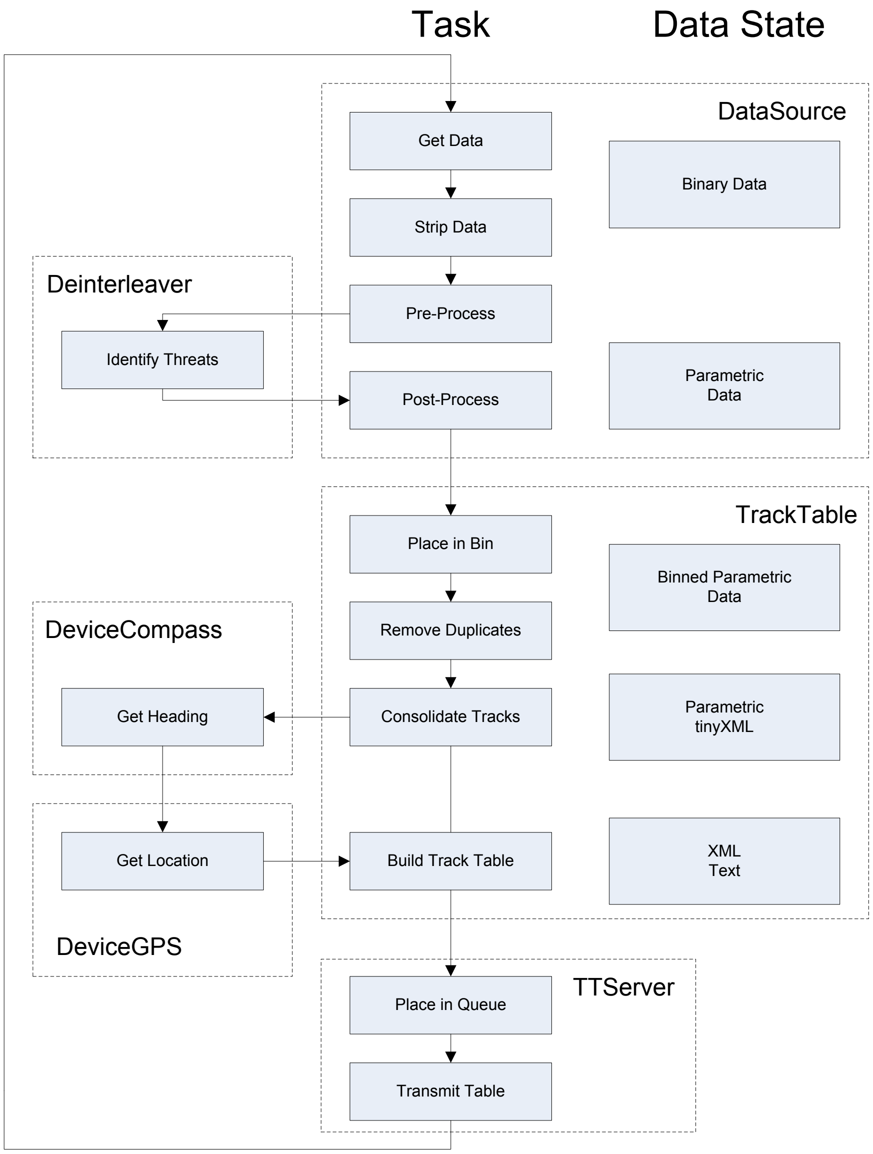


Figure 12.1.: Data Centric Design

maximum and the minimum value where appropriate. Once the duplicates have been removed the bins are consolidated into one and the parametric data loaded into a data structure from the TinyXML library. To build the track table, as well as the pulse data, the host platform data is needed. The current heading and location are retrieved from the `DeviceGPS` and `DeviceCompass` class. Once all the data has been loaded into the TinyXML structure it is then exported as a string of XML text and placed in the queue of the `TTServer`.

Beyond reducing allocations and copies to maximise data throughput, any task that may block on device access was moved from the main loop if possible. During the first test of the complete system with real data, it was identified through trial and error that collecting data from the compass was responsible for a delay of tens of seconds. As heading and location data do not change quickly with time for most platforms, the updating of heading and location data was moved to separate threads which only updated every five seconds. This reduced the maximum potential waiting time from the hardware access time to the time it took for the other thread to update the shared variable and release the mutual exclusion (mutex). For the same reason the `TTServer` also operates in a separate thread allowing the transmission of track tables and reacting to commands to happen independently. The state of the main loop is protected so that it cannot be changed whilst an incomplete track table is being processed. This leaves just the acquisition of data from the RR017 on the main loop. Functionality for threaded data acquisition was added but disabled as it was difficult to determine in the absence of reliable timing data whether it was worthwhile. Answering this question was one of the main motivations for this work.

12.4. Measurement Technique

In normal operation the Phobos application does not generate enough log information to allow any sort of timing analysis. In order to provide this information, simple time measurement functionality was added in the form of the `TimeCheck` class. As any attempt to measure the time taken to complete a task will also affect the time taken, the class eschews complexity in order to minimise this impact. There is one global instance of `TimeCheck` which is enabled through a compile time switch and it only records data on events chosen by other compile time switches. This eliminates the code entirely when it is not desired. At application start-up the object is initialised in order to open the log file and then the commencement and completion of tasks can then be recorded using the `logEvent` method. Events are an enumerated list of matching start and stop events. A start event saves the current system time and a stop event writes out the time taken between the two events. For example, a start event for data collection would be logged before the call to the `DataSource` to get the data. A stop event would then be logged when that function returns. Starts and stops for different event pairs can overlap and there is checking for start and stop events received out of sequence such as a stop event with no preceding start event. There is no thread safety and events may only be logged from the main thread.

The duration of some events is expected to vary according to other parameters. The amount of time taken to build a track table will depend upon how many tracks are present. The duration of a USB command may depend upon which command was issued. Consequently the `logEvent` method has a string as an optional parameter, which is written to the log file along with the duration of that event. As well as time, the CPU load is measured once per second and recorded in the log file once per main loop iteration. The combination of this information will show which tasks are requiring the most time and provide a much clearer picture of general system loading and how external factors affect the duration of the main loop.

As well as the timing data, the system can also record the raw pulse data. This is to allow real pulse data to be used as input data when the system is being developed or tested away from any radar emitters. The data is transferred from the RR017 to the host PC in USB packets which contain 42 pulse descriptor words and each PDW contains a time stamp. Analysis of the PDWs should allow verification of the main loop time, estimation of the actual pulse density and estimation of the number of pulses missed through quadrant switching. The main loop time is the time between data grabs from the same quadrant. The true pulse density is the pulse density within a single data grab. The number of pulses missed whilst switching is the time stamp difference between the last pulse of one grab and the first pulse of the next grab multiplied by the true pulse density. This assumes that the RR017 is capable of detecting every pulse and that the pulse density is constant. These assumptions are not safe, but allow a starting point and can be revisited when they are better understood.

Start Time	0951	1151	1438D	1450D	1458D
Data Collection (ms)	1660	1250	1825	1903	2240
Pre-processing (ms)	2.52	0.66	10.00	8.32	9.20
Pulse Identification (ms)	15.32	51.75	203.85	204.30	231.62
Post-processing (ms)	1.76	1.88	95.85	90.83	123.88
Track Table Creation(ms)	0.52	3.57	31.95	31.96	35.09
Unaccounted for (ms)	0.76	1.39	5.90	6.98	8.59
Total (ms)	1681	1309	2173	2245	2649
Standard Deviation (ms)	63	316	1269	1300	1137
Pulses processed per second	4791	3014	5046	4922	4537
Grab Size (bytes)	24576	12288	51200	51200	51200
Pulse Detection Threshold	300	400	400	400	400
Average Pulse Count	8054	3768	10964	11049	12018
Average Tracks	Not recorded	21	21	22	26
Under Reads (%)	0.4	21	45	44	36
CPU Load (%)	8.1	6.8	16	17	15

Table 12.2.: Timing Results

12.5. Results

The data presented here was captured on the 6th of August 2010 at the Dstl facility at Portsdown West near Portsmouth. This location was chosen because of its dense and varied radar environment. There are pleasure craft, small commercial vessels, larger commercial vessels such as ferries, military vessels and land installations, all operating at the same time. The purposes of this trip were to prove the timing code and to measure the initial system performance.

12.5.1. Timing Results

The results from the timing log are shown in Table 12.2. Five sets of data were collected and are identified by their respective start times. The top half of the table contains the average timing information and the number of pulses processed per second whilst the bottom part of the table contains system information and other recorded information. The two runs from the morning are both release builds and represent a similar configuration to what a customer might actually use, whilst the runs from the afternoon are debug builds which record extra pulse information and are as a consequence slower. The average time taken for each main loop task relative to the average length of the main

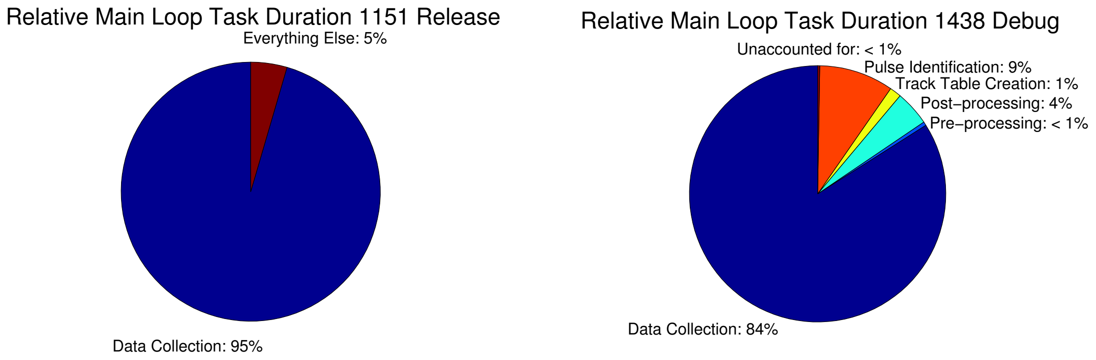


Figure 12.2.: Relative Main Loop Durations

loop is shown in Figure 12.2.

The first run was recorded with a low pulse detection threshold. This was to provide an abundance of data so that the system would not be affected by the fact that the radar environment to the north of where the system was sited was relatively sparse. The abundance of pulses comes from mobile communications emissions which are in band at the lower end of the systems frequency range. The second run has the threshold set to exclude mobile communications emissions but with the grab size reduced in order minimise the time spent in the sparse quadrants. The system parameters for both of the first two data captures were chosen with latency as the primary consideration. The three data captures from the afternoon were all made with the same system parameters. Here the grab size was increased to investigate the effects of increasing pulse throughput on latency. When the requested amount of data has not been received, the USB interface waits for a timeout to occur before returning the data it has received. The timeout is configured during USB endpoint initialisation and set to one hundred milliseconds, though this timeout appears to be ineffectual.

For all of the data captures the time taken performing data collection dominates. For the 1151 data capture it is on average 95% of the main loop duration and for the 1438D data capture – 84%. The duration of the main loop during the 0951 data capture is almost constant having a standard deviation of just 63 ms which is only four percent of the mean duration. The data captures with a higher threshold have a much higher main loop duration standard deviation. The coefficient of variance was twenty four percent for the 1151 capture, rising to fifty eight percent for the 1438D and 1450D captures. The shortest main loop duration for the afternoon data captures was less than half a second and the longest almost six seconds. All of the data captures with the high threshold encountered a significant number of under reads. This value does not represent the amount of missing data, but rather the number of data grabs which did not return the full amount of data.

The measurements did reveal an unexpected stepping in the data. As the USB interface only transmits full five hundred and twelve byte packets which contain forty two pulse descriptor words, the pulse data was expected to be stepped in forty two pulse steps which it was. Surprisingly, the timing data was also stepped in steps of fifteen or sixteen microseconds. This was discovered to be a feature of Microsoft Windows. Although the timer is millisecond accurate, it's value is only updated every fifteen to sixteen milliseconds [12].

12.5.2. Raw Data Inspection

For all five data captures, the binary pulse data was saved so it could be examined after the event. It was hoped that this data could be then used to verify the log data, to provide an estimate of the actual pulse density and to provide an insight into the amount of time lost during quadrant switching.

Unfortunately subsequent analysis of the data has shown it to be of very poor quality making these objectives impossible. At the time of the data collection it was known that there were some issues with the RR017 firmware as pulses with very high frequency values had been observed. What was unknown was the extent of the problems. The results of the data analysis are shown in Table 12.3.

	0951	1151	1438D	1450D	1458D
Good	58%	57%	57%	57%	57%
Bad	42%	43%	43%	43%	43%
% of bad that would be good if swapped	66%	76%	76%	76%	76%
Total	491526	615720	2881872	2371068	2094078
Good	35%	33%	33%	33%	33%
Bad	65%	67%	67%	67%	67%
Transitions	61304	111975	522103	430606	379892

Table 12.3.: Raw Data Analysis

Each pulse, after it was decoded from the binary data, was then assessed for authenticity. A pulse would be declared as bad if either its frequency value was out of range or if both

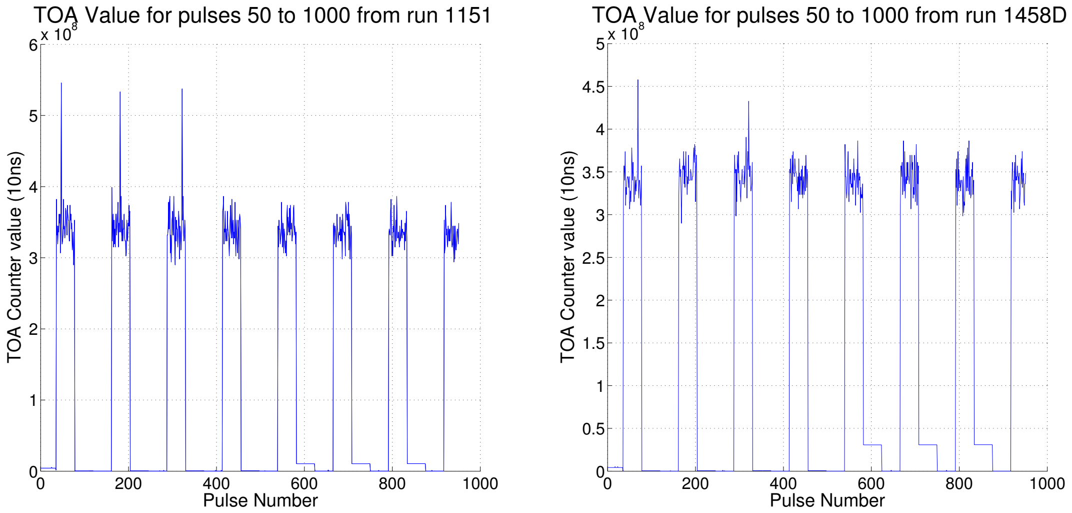


Figure 12.3.: Time of Arrival Values for Consecutive Pulses from Two Data Collections

amplitude values were below the noise floor. The binary pulse descriptor word is made of three thirty two bit words with the first two containing a variety of pulse characteristics and the third word containing the time of arrival value. During inspection of the binary pulse data it was noticed that sometimes the first two words seemed to be in the wrong order. This was tested by swapping the first two words for bad pulses and reassessing the pulse. The swapped row in the table is the percentage of bad pulses that would have been scored as good if their first two words had been reversed. The almost constant rate of bad pulses across the five runs suggests that there is something systematically wrong with the way the pulse descriptor words are constructed. The fact that the bad pulses were often better with their first two words swapped was only part of the problem.

Pulses in succession in the data were observed with the same TOA value or with a TOA that appeared to have gone backwards. The TOA counter is a 32 bit unsigned integer value that counts in ten nanosecond steps. This means the counter will wrap every 42.9 seconds. Knowing from the first set of data that the system was polling all four quadrants in times much less than this, it is very unlikely that two consecutive pulses in the binary data data stream would have the same TOA value or that the following pulse would have a TOA value of just less than the previous pulse. The TOA values for the first nine hundred pulses of the 1151 collection and the 1458D collection are shown in Figure 12.3. The first fifty pulses have been discarded as due to buffering there is an expected TOA discontinuity after the first USB packet of forty two pulses. The average pulse processing rates for the two data sets are 3014 and 4537 pulses per second, so the change in TOA value over one thousand pulses would be expected to be less than one second for both sets. This does not happen in either set, instead the TOA value moves from close to zero to around a value that represents three and a half seconds later, where it stays for a brief and noisy period before returning to zero. After staying around zero for a number of pulses there is then another period of three and a half seconds later noise. This pattern repeats eight times for both data sets with a very similar period. Sometimes on the way down there is a flat section, this can be seen on both graphs at approximately pulse 600, 725 and 850. This is a period of normal operation and the periods where the TOA value appears to be at zero on the graphs also represent normal operation. The graph appears flat due to the large scale of the TOA counter value axis, when in fact the value is actually rising. The system appears to work correctly then enter a fault state, then somehow recover only to later re-enter the fault state. With limited data further analysis of the fault was impossible in the period of the project. The issue was reported to the firmware team at TDL, who solved the problem for the first production standard model.

12.6. Conclusions

The objective of this work was to make the first credible measurement of system performance. This measurement was expected to show how far away from the performance targets the system was, which areas should be chosen for improvement and to allow subsequent improvements to be quantitatively assessed. The timing results show that the system can process three thousand pulses per second with an average latency of 1.3 seconds and reveal that the time taken to collect the data from the RR017 is the task where improvement effort should be concentrated. As well as achieving the objectives the work has also revealed the true extent of the known data quality issues of the RR017 device used in the prototype.

The results are a long way off the performance targets set, but it is important to note that the results presented here represent a single snapshot of system performance in an unknown environment and as such the data can not be used to estimate the upper bound of performance for this hardware configuration. Due to the data quality problems it is impossible to estimate the true pulse density in each quadrant so the number of missed pulses remains unknown. Until the true density is known it cannot be determined if the very long data grab times are due to inherent slowness or because the device is simply waiting for pulses to arrive to complete the data packet.

It was decided to make the first performance measurement in a realistic scenario primarily to benefit the overall project. As well as making these measurements, the system as a whole was tested and improved over the course of the two days. It was also hoped that valuable pulse data could be recorded in order to be used as simulation data for subsequent system development. A hoped for secondary benefit was that the real emitters would provide a more realistic workload for the pulse identification software with the assumption that the time required for pulse identification would be a dominant factor. As that assumption has now been shown to be incorrect, it is clear that until pulse identification time does become dominant, using synthetic data will allow better testing of system performance.

12.7. Future Work

Reducing the data collection time is the first task as it dominates the main loop duration. Before this work begins, it would be greatly beneficial to enhance the logging code to provide extra information. The logging of the amount of received data should also include per quadrant information as well as total per cycle. The amount of missing data should also be recorded with reference to the grab size. This will allow sparse quadrants and their effect to be much more easily recognised and measured.

If the USB interface is to be retained on a production device, there are problems to address and already planned improvements. When the USB interface is configured a timeout value is chosen – currently one hundred milliseconds. Changing this value seems to have little to no affect upon system performance and needs to be further investigated. In order to improve system latency in sparse environments automatic data grab size modification is planned. This will grow and shrink the data grab size between configured minimum and maximum values depending upon whether there was missing data in the last grab, which will reduce the impact of having to wait for data in sparse environments in the absence of an effective timeout. If the data collection time can be reduced to nearer half the main loop time, throughput could be improved by moving the data collection to a separate thread. This would allow the other main loop tasks to run concurrently to data collection.

Feedback from potential customers suggests that the RR017 would be a more attractive product if it came with an Ethernet interface in place of USB. Consequently, if interfaces other than USB are to be evaluated, Ethernet should be the first.

13. Publications

13.1. Design and Realisation of Linux Based Wireless Data Acquisition System

Design and Realization of Linux Based Wireless Data Acquisition System

Y. Cheng[1], Y. Fun Hu[1], P. Jiang[2], S. Brown[3], A. Metcalfe[3]

[1]School of Engineering, Design and Technology, [2]School of Informatics
University of Bradford
Bradford, West Yorkshire, BD7 1DP, UK
Emails: {y.cheng4, y.f.hu, p.jiang}@bradford.ac.uk
[3]Filtronic Defence Ltd.
Airedale House, Acorn Park, Shipley, BD17 7SW, UK
Emails: { s.brown, a.metcalfe }@fcl.com

Abstract: This paper presents a wireless sensor system developed for environment monitoring. Both live streaming video (1Mbyte/s) and data sensor signals with a high sampling rate (10Mbytes/s) are required to be transmitted from a mobile station to a monitoring PC in real-time. In order to obtain a 12Mbytes/s wireless data sampling rate, an embedded computer links a video camera and a Cypress signal acquisition board through USB 2.0 ports on the sensor side, whilst IEEE 802.11g is used for wireless communication of sensor signals to the monitoring PC. It is significant in wireless sensor system that the communication link should be reliable for high data throughput. The design and implementation of wireless communication for reliable and real-time sensing applications using the commercial off-the-shelf techniques are reported and discussed in this paper. The factors affecting the performance of data transmission such as the data rate, time delay, packets loss impacted by distance are experimentally investigated through the developed prototype system.

Keywords: data acquisition, Linux based, USB device operation, wireless sensor.

1. Introduction

Over the past years, a lot of research attention has been paid to wireless sensor networks (WSN) [1]-[3] due to its wide range of applications in engineering, military, healthcare, etc. Many design aspects in WSN such as the network architecture, protocols, signal processing, software and hardware platforms [6] have been extensively researched. Wireless sensor networks have provided vast opportunities for the development of wireless telemetry systems for remote monitoring and control for low-rate and short frame sensor information. However they are not applicable for wireless sensor systems requiring mass sensor data to be transmitted in real-time, such as video or radar surveillance.

This project is aiming at monitoring environment via high data rate wireless data sensors and cameras. Sensor data sampling rate is 10Mbytes/s and video stream data rate is approximately 1Mbytes/s. Meanwhile, sensors are equipped with end effectors so that they are able to execute actions in real time according to received commands from remote control terminals. Although various wireless cameras are available in the market [4], they use proprietary interfaces to transmit video. This makes it extremely difficult for users to integrate them with other wireless sensors or add more functions for long distance transmission. For most of the existing environmental monitoring systems with intelligent front end unit which are able to access the off-the-shelf digital cameras or sensors, they adopt embedded windows system or use Ethernet interface devices [5] with embedded web server. One reason for doing this is because no official driver is provided for USB device, such as digital cameras (unofficial Linux drivers can be found in [14] [15]), PT (Pan/Tilt) controllers, data sensors [7] [8]. This either prolongs the real-time response time which may have an adverse effect in case of emergency control or reduces the range of sensor options, especially the majority of digital cameras have only USB interface.

In this paper, the development of a wireless data acquisition and real-time control system for high rate data and video stream transmission from remote sensors across a radio link, faithful display on a remote terminal and execution of received commands on sensors is described. To achieve a high data rate and real-time wireless sensor/actuator system with the capability of future expansion, IEEE 802.11g wireless communication protocol and Linux operation system based design and realization are considered. Specifically, the design and realization of the proposed practical remote data acquiring and video capturing system are based on the embedded Gentoo Linux operation system [11] in the sensor module which links a digital camera and a Cypress signal acquisition board through USB2.0 ports and the Windows XP operation system in the user side monitoring PC. It adopts two different ways to access data sensor and real-time video [15], i.e. via libusb [13] and VLC [12] respectively for improving system flexibility. The UDP is adopted to transmit data upon reception of high priority real-time demands whilst TCP is established to send commands for reliable transmission.

A series of experimental tests have been carried out to evaluate the system performance in terms of bit rate and the packet loss ratio against the transmission distance. The results obtained can be used as a reference for development of other similar systems.

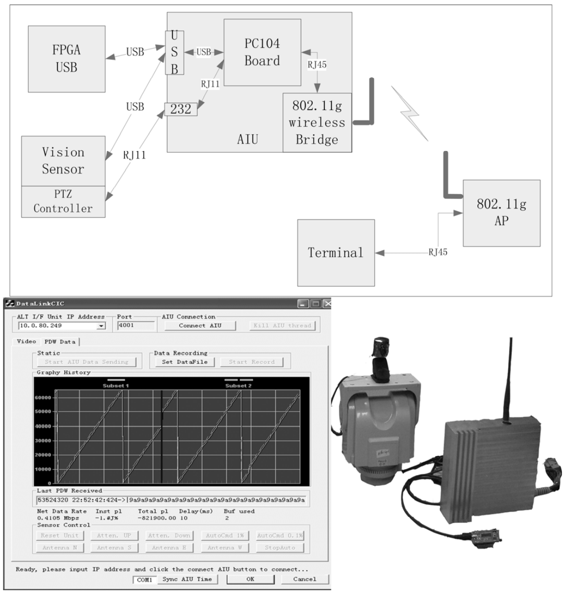
The remaining parts of the paper are organized as following: section two describes in detail the system design and implementation with sample software codes; section three discusses the implementation issues; following experimental evaluation and analysis in section four, conclusions are given in section five.

2. System design and realization

This section will share the key issues in implementing the three main modules from system level design to module level.

2.1 System hardware platform and networking

The whole system is divided into three main parts: The sensor units, the Air Interface Unit (AIU) and the Remote Terminal (RT). Two kinds of sensors are used for data acquisition and video surveillance simultaneously: A Cypress sensor [10] for data acquisition and a digital camera mounted on a Pan/Tilt (PT) motion controller [9] for real-time video surveillance respectively. The AIU is a PC-104 based module with AMD LX800 500MHz CPU, 256M embedded DDR memory and front panel connectors providing two USB2.0 ports, one RS232 serial port, two mini PCI connecting hard disk and two 100Mbps Ethernet ports. Comparing to other processors, this module has powerful processing capability, Gentoo Linux operating system and abundant interfaces with a single 5V power supply and less than 5 Watt power consumption with a dimension of only 111 x 91 x 10 mm. To achieve better communication effect, one Ethernet to wireless bridge with external antenna is used with AIU. The RT can be any computer running windows XP system with Ethernet connection. The system structure, software screen snapshot and prototype are shown in Fig. 1.



```
ALT I/F Unit IP Address: 10.0.80.249    Port: 4001
Last PDW Received: 53524320 22:52:42:424->|9a9a9a9a9a9a9a9a9a9a9a9a9a9a9a9a9a9a9a9a9a9a
Net Data Rate   Inst pl   Total pl     Delay(ms)   Buf used
0.4105 Mbps     -1.#J%    -821900.00   10          2
Ready, please input IP address and click the connect AIU button to connect...
```

Figure 1: System structure, hardware and software snapshot

IEEE802.11 is chosen for the wireless link between the AIU and the RT to enable clients to access sensors wirelessly via the Internet. All sensors are associated with nearby APs and then connect to RTs directly or via the Internet. For reliable connections to exchange commands between the AIU and the RT, TCP socket connection is established while UDP socket connection is used to achieve the best effort data rate.

2.2 Video capturing and data acquiring

Normally, videos are captured from commercial off-the-shelf digital cameras while data are collected by specifically designed sensors tailored to customers' need. In our system, both devices use USB interface. To enable the capturing of video streams from the USB digital camera (vision sensor) and the data sensor unit, the AIU module is designed with two USB2.0 ports and a wireless link for data transmission to the RT.

Two different ways can be used to interface with standard USB devices: VideoLan Client (VLC) plug-ins for video capturing, streaming and displaying; user level USB devices access library libusb for data acquisition from sensors[1].

Although more and more digital cameras have drivers natively supported by the Linux kernel, in most cases suitable drivers need to be downloaded from and compiled with the kernel again. After the driver is installed, a device named video0 will appear in the /dev directory. VCL will use this device as entry to capture videos.

The action VLC will carry out functions according to the commands passed by the initiated function. After that, 'start' or 'stop' function can be called whenever they are needed by different scenarios. The IP address and port numbers are represented by integer variables, *RTIP* and *videoport* respectively but they may change when different RTs try to connect to the AIU. The codes for capturing and streaming videos via UDP protocol in Linux is as shown in appendix A.1.

When there is a need to stop the capturing and streaming, the function *libvlc_playlist_stop(inst,&excp)* can be called.

Meanwhile, in the Windows-based RT, similar codes are used to receive and display videos. To hook the video display window to the desired one, one window handle should be kept and passed to the *libvlc_video_set_parent* API function. Notice that in the receiving part, only the port number should be passed to initiate function and it is always running while leaving the start and stop control to AIU. The benefit for doing this is obvious: if the RT stops receiving video from its own side, the AIU will still stream out video and waste the bandwidth.

The biggest advantage of using VLC is that video is coded/decoded and UDP socket is implicitly established when passing arguments to initiation functions in both AIU and RT.

To collect data from sensors by the libusb API, there are two options of reading data: bulk read and interrupt read. The option to be selected depends on sensor capability. The procedures of these two methods are similar although bulk

[1] VLC is an open source software by the VideoLAN project, supporting many audio and video codec and various streaming protocols; libusb is also an open source libraries to access USB devices in user level.

read is adopted in this paper. Specifically, three steps will realize USB devices reading and/or writing function:

(i) get the usb bus;
(ii) find USB device from USB bus;
(iii) open the specific USB device for reading and writing.

The procedure described below shows how USB devices are initiated, how all the USB bus and devices by *libusb* APIs can be accessed and how the information can be stored into a *usb_bus* structure:

```
struct usb_bus *usbBus;
usb_init();   //initialize USB device
usb_find_busses();
usb_find_devices();
usbBus = usb_get_busses();
```

Each USB device manufacturer has one exclusive manufacture ID issued by the Manufacturers Association and different kinds of devices have different unique product IDs. These two IDs are used to identify USB devices in operation systems. So the second step is to walk through all the buses and devices, match the given manufacture ID and vendor ID with the ones got from the system. This process will keep the identified USB device pointer for later usage or a NULL will return if there is no device matching the given one. The sample codes are shown in A.3.

Before read/write data from/to USB device, the *usb_bulk_read* function needs to know the device handle, which can be obtained as described in the steps above. The read entry point, the buffer memory to store the read data can be viewed from the log information of USB devices when plug in. 512 bytes are read in one loop; this is defined by the USB device capability. Finally, a time out value is set at 1000 milliseconds.

```
DevHandle = usb_open(UsbDevice);
char buf [512]
readCount = usb_bulk_read(DevHandle, READ_EP, buf, 512, 1000);
```

2.3 Networking and Communication protocols

Based on TCP sockets and considering the requirement of the information to be exchanged between AIU and RT, the communication protocol is defined by a seven-byte length packet. The AIU is implemented as a TCP server, which listens to the socket connection from clients in the RT. All sensor motion control, PT control, status control and RT status feedback information are contained in these seven bytes packets. As mentioned in earlier section, the data and video port numbers will be negotiated during the connection process, so there are two categories of commands: with or without associated data. Details of the protocol are shown in the following:

Table 1. Communication protocol.

Byte	B0	B1	B2	B3	B4	B5	B6
				Packets from RT to AIU			
Value	1/0	1/0	1/0	CMD	Data High	Data Low	Checksum
				Packets from AIU to RT			
Value	‘C’	‘I’	‘C’	CMD	Data High	Data Low	Checksum

Three-byte header is used to synchronize the packets between the AIU and the RT. Meanwhile, these three bytes are also used for the AIU to send feedback information on receiving status. When data are received correctly, these three bytes are set to *1* otherwise *0*. B3 is a one-byte command type identifying different commands. B4 and B5 are used for containing command data associated with B3, i.e. command “start data transmit” needs the port number to send data from the AIU to the RT. If the command is “initiate antenna position”, then B4 and B5 are filled with zeroes. B6 is the checksum, which equals to the sum of B3, B4 and B5 modulo 128. The values in B4 and B5 are defined as follows:

Data High = (Value-Value%128) %128
Data Low = Value%128

3. Implementation Issues

3.1 Digital Zooming Function

As described in section 2.2, the benefit to adopt VLC as the video process module is that the details of coding and decoding, streaming and display are encapsulated, leaving our implementation to focus on the data acquisition and other areas. However, there is no other way to access the video frames other than modifying the source codes. This means that the traditional methods for digital zooming will not be suitable anymore. This paper proposes a partial displaying window method to realize the zooming function by first creating two objects, one (*m_videoframe*) has fixed dimension and right location in the main window and another one (*m_videowin*) has variable dimension, as shown in Fig. 2, whose handle will be passed to *libvlc_video_set_parent* API function for displaying video (VLC will put video full size to fit this window through the received handle). After that, the desired window width, height and position will be calculated according to the zooming factor along with the offset; and the variable window *m_videowin* will have to be resized and moved to the newly calculated position. Finally, the appropriate portion which has the same dimension with the fixed object window will be taken out of the new *m_videowin* for display on the computer screen.

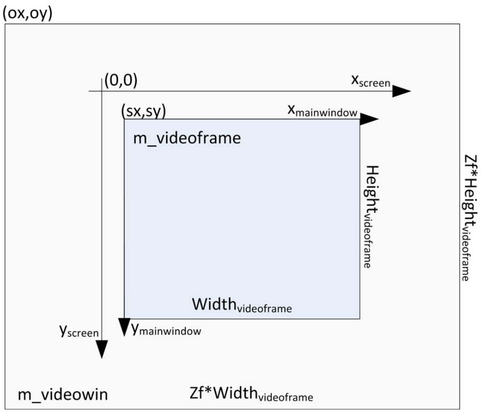


Figure 2: Digital zooming

Resize and move is realized by calling the Microsoft Foundation Class (MFC) function *MoveWindow* and specify a region by calling the MFC function *SetWindowRgn*. This method realizes the digital zooming function while avoiding the need to access video frames or graphic card memory. A.4 shows the codes to calculate position and shows the specified regions.

3.2 Program compiling under Linux

To compile the libusb and VLC libraries under Linux, *-lusb* and *-lvlc* options should be added into the Makefile. In some cases, some additional options like *–lmemcpymmx, -lmux* etc. are needed as well. The Makefile is shown in A.5.

3.3 AIU and RT synchronization

In addition to achieve the highest data rate, short time delay is another important requirement of the system. Time delay includes the data processing time in the AIU from the time of getting the data from the sensor to the time transmitting it, network delay in transmitting between the sender and the receiver, processing time delay in the RT from receiving the data to actually recording and displaying them. Before the time difference is recorded by AIU and RT becomes meaningful, time synchronization between these two modules must be done. Although the NTP (network time protocol) time synchronization servers are available from Internet, the Internet itself has inevitably unstable time delay and when the system is only used in local network, this option has its intrinsic shortcomings. The most practical way to build a NTP server in the local network is to set up a stratum 2 or stratum 3 clock. Unfortunately, this kind of NTP server needs to synchronize to higher stratum clock, which is not always available in the local network.

Since the most concerned matter is the time difference between the AIU and the RT, serial port hardware synchronization is proposed in this paper. Not only because it is easier to implement but also because the precision is better and more stable than the NTP server due to the shorter time delay and simpler networking only via serial cables. The following Table 2 compares these two methods in terms of time delay (including preprocessing time), stability and physical requirements.

Table 2. Comparison between the serial cable synchronization and NTP synchronization.

	Serial	NTP
Time delay	20ms	300~500ms
stability	No interference by environment	Interfered by network environment a lot
requirement	Serial cable	Internet NTP server higher than stratum 3

The time synchronization flow of serial port hardware time synchronization is shown in Fig. 3 and summarized as below:

(i) The RT initiates one byte and sends it to the AIU;

(ii) The RT records the system time just after sending the byte and the AIU records system time just after the serial interrupt event. In theory, the AIU and the RT should take action to record the system time at the same time;

(iii) The RT sends the recorded time to the AIU;

(iv) The AIU compares the received time with the recorded time itself;

(v) The AIU deducts the time difference and set its time to the deducted time.

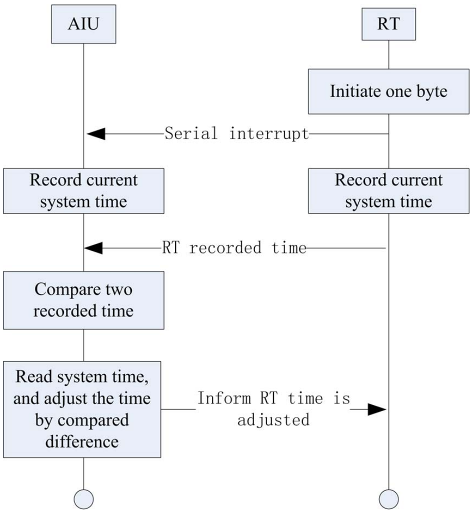

AIU
RT
Initiate one byte
Serial interrupt
Record current system time
Record current system time
RT recorded time
Compare two recorded time
Read system time, and adjust the time by compared difference
Inform RT time is adjusted

Figure 3: Clock synchronization flow

4. Performance analysis

4.1 Throughput, time delay

Adopting UDP protocol means packet loss is inevitable, this section is to find out the relationship between throughput and packet loss. Although the maximum signaling rate of 802.11g is 54Mbit/s, the ideal net throughput is only about 19Mbit/s. When coexist with 802.11b network, due to different modulation techniques and extra transmitting power to overcome interference, the net data throughput is less than 19Mbit/s. The maximum practical net data rate achieved in this paper is 16Mbit/s. Data packet length is 512 bytes which consist of 32 sensor pulse descriptor words with 128 bits each. So each successfully received packet means 32 correct pulse descriptor words. By comparing the received data with original ones, the received data incorrect rate is equal to packet loss. Time delay is measured by comparing time stamps recorded in both the AIU and the RT. Table 3 shows the relations of the data rate and packet loss.

Table 3. RT ←cable→AP←wireless→AIU

Transmitting Data rate	Receiving Data rate	Packet loss	Time delay
24Mbps	16Mbps	~10%	~50[1]ms
16Mbps	16Mbps	~3%	~25ms
12Mbps	12Mbps	~3%	~25ms
6Mbps	6Mbps	~1%	~25ms
3Mbps	3Mbps	~1%	~25ms

1. The data in the table are average of the obtained values

When the AIU transmits data at a data rate more than the Access Point (AP) can accept, the data rate drops down, packet loss increases quickly and the average time delay becomes longer due to congestion at the AP. There is not much difference in packet loss and time delay when the transmitting data rate is low enough for the AP to handle. The maximum data rate is the overall data rate of the radio link, when both transmitting and receiving by wireless link, the data rate drops to nearly half of the maximum throughput, when the transmitting data rate becomes too high, the AP is too congested to transfer data. The result is shown in Table 4.

Table 4. RT ←wireless→AP←wireless→AIU

Transmitting Data rate	Receiving Data rate	Packet loss
16Mbps	AP down	~100%
8Mbps	8Mbps	~3%
6Mbps	6Mbps	~1%
3Mbps	3Mbps	~1%

The UDP data flow is from AIU to RT without acknowledgement. To test the effect on data rate and packet loss caused by TCP commands sending back from the RT to the AIU, an experiments were carried out: create TCP information request commands, and force the RT to send them at the ratio of 0.1% and 1% of the UDP data received from AIU. The result is shown in Table 5.

When the data rate is lower than 6Mbps, there is almost no effect from TCP commands on data rate and packet loss. However when the data rate increases to higher than 12Mbps, the effect of sending TCP commands on both data rate and packet loss becomes bigger. Around 0.3% to 0.5% more packets are lost and data rate becomes 1Mbps to 2Mbps less than before when 0.1% to 1% percentage commands applied.

Table 5. Achievable data rate and packet loss

Data rate	Packet Loss	Commands %	Data rate with commands	Packet Loss with commands
16Mbps	~3%	1%	2Mbps lower	~3.5%
16Mbps	~3%	0.1%	1Mbps lower	~0.5% higher
12Mbps	~3%	1%	2Mbps lower	~0.3% higher
12Mbps	~3%	0.1%	1Mbps lower	~0.3% higher
6Mbps	~1%	1%	0.5Mbps lower	No effect
6Mbps	~1%	0.1%	<0.1Mbps lower	No effect

4.2 Distance effect on performance

The maximum transmission range the system can achieve depends a lot on the environment. A test is carried out in a real work environment along with a long corridor with fire doors closed at each 10 meters as shown in Fig. 4.

The target is to find out the maximum transmission range the system can work in an indoor environment. A transmitting data rate of 2Mbps is selected so that the time delay and packet loss can be tested in the same data rate condition in all distances. The result is shown in Table 6.

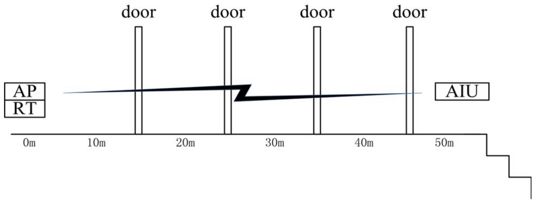


Figure 4: Test scenarios along the corridor

Table 6. The effect of distance in the transmission range

Distance (m)	Signal (dBm)	Data rate (Mbps)[1]	Packet loss	Delay (ms)
0	-15~-20	2.2	<1%	25[2]
10	-40~-50	2.03	~1%	~(0m delay)+20
20	-55~-60	2.03	~8%	~(10m delay)+20
30	-65~-70	2.03	~8%	~(20m delay)+20
40	-70~-75	2.0	~8%	~(30m delay)+10
50	-75~-80	2.0	~30%	~(40m delay)+10

[1] When there is some interference like people moving around, the data rate may drop down to about 1.3Mbps or even lower; This data rate is obtained when the wireless link become stable and data transferring is stable, the same for packet loss and delay.

[2] Due to synchronization precision, the delay in 0m distance may be different.

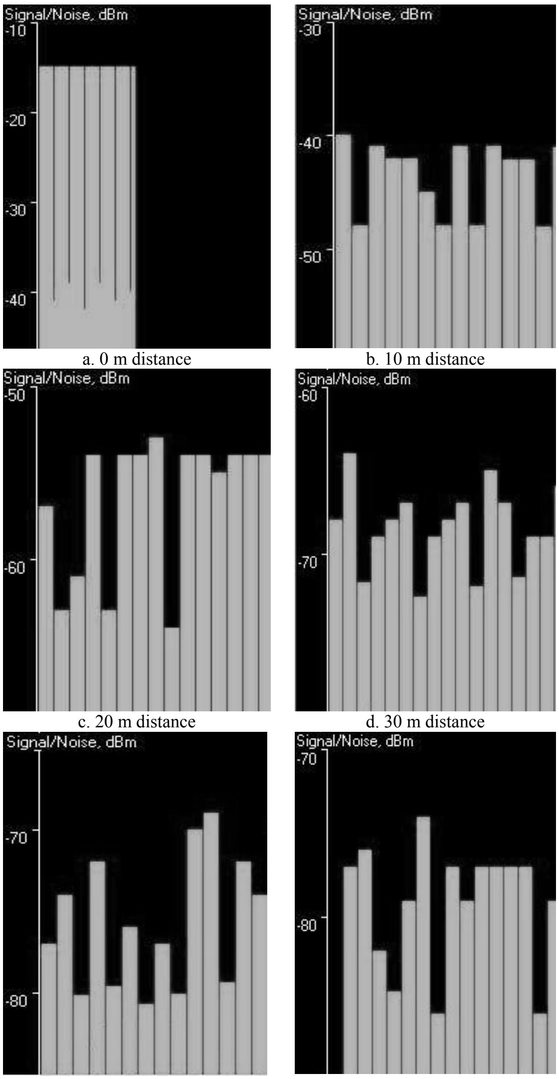
As shown in Fig. 5, within the range of 30 meters, the signal strength drops at 10dBm/10meter on average, and the time delay increases at 20ms/10meter on average. From 30 to 50 meters, the signal strength drops at 5dBm/10meter and the average time delay increases at 10ms/10meters. Fig. 5 shows the signal samples in different distances from AP (RT) to AIU.

5. Conclusion

The paper presents a high data throughput data acquisition and video streaming system based on embedded Linux computer interfacing with sensors by USB 2.0. Because of its flexible structure and open USB connection, this platform can be used for a wide range of applications for data acquiring and streaming. Distance sensitive factors such as data rate, time delay and packets loss are experimentally investigated. Future research will be focus on ad-hoc networking, more efficient packaging, security management and power consumption reducing.

Acknowledgement

This project is funded by Filtronic Defence Ltd. And the authors would like to thank Mr Craig Marshall and his team from Filtronics for their valuable advice and suggestions throughout the duration of the project.



a. 0 m distance

b. 10 m distance

c. 20 m distance

d. 30 m distance

e. 40 m distance

f. 50 m distance

Figure 5: Signal strength (dBm) between the RT and the AIU for 0m, 10m, 20m, 30m, 40m and 50 distance.

References

[1] I F Akyildiz, W Su, Y Sankarasubramaniam and E Cayirci, A Survey on Sensor Networks. IEEE Communication Magazine, Vol. 40, No. 8, 2002, pp. 102-114.

[2] B M Sadler, Fundamentals of Energy-Constrained Sensor Network Systems. IEEE Aerospace and Electronics Systems Magazine, Vol. 20, No. 8, 2005, pp. 17-35.

[3] K Romer and F Mattern, The Design Space of Wireless Sensor Networks. IEEE Wireless Communications, Vol. 11, No. 6, 2004, pp. 54-61.

[4] http://www.pinecomputer.com/ztvzt24chwic.html

[5] Y Kakimura, R Hirayama, H Hishikura, M Matsuzawa and M. Tago, A real-time MPEG1 Ethernet camera system, IEEE Transactions, Consumer Electronics, Vol 45, No. 3, 1999, pp. 925-931.

[6] A Mainwarning, D Culler, J Polastre, R Szewczyk and J Anderson, Wireless Sensor Networks for Habitat Monitoring. Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications, 2002, pp. 88-97.

[7] B Bi, S Sun and C Wang, Design of Data Acquisition Equipment Based on USB, 8th International Conference on Electronic Measurement and Instruments, 2007, pp. 1-866-1-869.

[8] J Hyde, USB Design by Example a practical Guide to Building I/O Device. Intel Press, 2nd Edition, May 2002.

[9] http://en.minking.cc/newEbiz1/EbizPortalFG/portal/html/ProductInfoExhibit.html?ProductInfoExhibit_ProductID=c373e911b528f08b8f6e0a2d4ae981dd&ProductInfoExhibit_isRefreshParent=false.

[10]http://www.cypress.com/products/?gid=9

[11]http://www.gentoo.org/

[12] http://www.videolan.org/vlc/download-gentoo.html

[13] http://libusb.sourceforge.net/doc/

[14]http://linux-uvc.berlios.de/

[15]http://www.exploits.org/v4l/

Appendices

A.1 Capturing and streaming videos via UDP protocol in Linux

```
char strvp[10];
libvlc_exception_t excp;
libvlc_instance_t *inst;
char *arg0 = "AIU_video";        //name, not important
char *arg1 = "v4l://";           //indicator for driver
char *arg2 = ":v4l-vdev=/dev/video0";//video device
char *arg3 = ":v4l-adev=/dev/dsp";  //audio device
char *arg4 = ":udp-caching=0";    //reduce caching delay
char arg5[150]="";
strcpy(arg5,":sout=#transcode{vcodec=mp4v,vb=1024,scale=1}:duplicate{
dst=std{access=udp,mux=ts,dst=");
strcat(arg5,RTIP);             //adding IP address to argument
strcat(arg5,strvp);            //add port number to argument
char *args[6] = {arg0, arg1, arg2, arg3, arg4, arg5};
libvlc_exception_init (&excp);
inst = libvlc_new (6, args, &excp);
quit_on_exception (&excp);
//start video capturing and streaming
libvlc_playlist_play (inst, -1, 0, NULL, &excp);
```

A.2 Sample codes for the API function.

```
char MRL[30];
char strvp[10];
libvlc_exception_t excp;
libvlc_instance_t *inst;
libvlc_media_instance_t *mi;
libvlc_media_descriptor_t *md;
strcpy(MRL,"udp://@:");
sprintf(strvp,":%d}}",videoport);
strcat(MRL,strvp);
char *arg0 = "-I";
char *arg1 = "dummy";
char *arg2 = "";
char *arg3 = ":udp-caching=50";
char *args[4] = {arg0, arg1, arg2, arg3};
```

```
libvlc_exception_init (&excp);
inst = libvlc_new(4, args, &excp);
quit_on_exception (&excp);
md = libvlc_media_descriptor_new( inst, MRL, &excp );
quit_on_exception( &excp );
mi = libvlc_media_instance_new_from_media_descriptor(md, &excp );
quit_on_exception( &excp );
libvlc_video_set_parent(inst,(libvlc_drawable_t)pParam,&excp);
quit_on_exception (&excp);
libvlc_media_instance_play( mi, &excp );
quit_on_exception (&excp);
while(1) Sleep(20000); //keep receiving
```

A.3 Find a specific USB device

```
struct usb_bus *tempUsbBus;
struct usb_device *tempUsbDevice, *UsbDevice;
usb_dev_handle *DevHandle;
char description[256],char str[256];
// manufacture ID and product ID.
int VendorId, productId;
// Walking the USB busses and devices
for (tempUsbBus = usbBus; tempUsbBus != 0;
  tempUsbBus = tempUsbBus->next)
 for(tempUsbDevice=tempUsbBus->devices;tempUsbDevice!=0;
      tempUsbDevice = tempUsbDevice->next){
        DevHandle = usb_open(tempUsbDevice);
        if (tempUsbDevice->descriptor.iManufacturer != 0) {
           usb_get_string_simple(DevHandle,
             tempUsbDevice->descriptor.iManufacturer, str, sizeof(str));
              snprintf(description,sizeof(description),"%04X-",
                tempUsbDevice->descriptor.idVendor);
            // convert string to unsigned long data for comparing.
              manId = strtoul(description, NULL, 16);
            // VENDOR_ID is defined to the actual Manufacturer ID
              if (VendorId == VENDOR_ID) {
                if (tempUsbDevice->descriptor.iProduct != 0){
                                          usb_get_string_simple(DevHandle,
                           tempUsbDevice->descriptor.iProduct, str, sizeof(str));
                           snprintf(description,  sizeof(description),  " - %04X",
                         tempUsbDevice->descriptor.idProduct);
                        productId = strtoul(description, NULL, 16);
                     // PRODUCT_ID is defined to the actual product ID
                        if (productId == PRODUCT_ID){
                          usb_close(DevHandle);
                          UsbDevice = tempUsbDevice;
                        //device found, break the iteration and store device
pointer.
                          break;
             }   }    }   }
    //close the opened USB devices when not used
     usb_close(DevHandle);
     }
```

A.4 Calculate position and show a specified region.

```
double zf;//zoom factor according to user's requirement
CRect fr,vfr,frmrect;//window position in screen, object window position
and fixed window position.
m_videoframe.GetClientRect(&vfr);
m_videoframe.GetParent()->GetClientRect(&fr);
int ox = fr.CenterPoint().x-(int)(vfr.Width()/2.0*zf);
int oy = fr.CenterPoint().y-(int)(vfr.Height()/2.0*zf);
int w = (int)(vfr.Width()*zf);
int h = (int)(vfr.Height()*zf);
m_videowin.MoveWindow(ox,oy,w,h,false); //resize and move
m_videowin.GetClientRect(&frmrect);
int sx = frmrect.CenterPoint().x-vfr.Width()/2;
int sy = frmrect.CenterPoint().y-vfr.Height()/2;
HRGN showrect = CreateRectRgn(sx,sy,sx+ vfr.Width(), sy+ vfr.Height());
m_videowin.SetWindowRgn(showrect,false); //show new portion of the
resized window
Invalidate(true);
```

A.5 Makefile example

```
KERNELV = `uname -r`
CC=gcc
INS=install
INSDIR=/usr/local/bin
LIBDIR=-L/usr/X11R6/lib -L/usr/local/lib -L/usr/local/lib/vlc
includes=-I/usr/src/linux-$(KERNELV)/include
LIBS=-lusb -lvlc -lmemcpymmx -lmemcpymmxext -lmemcpymmx -ldvbpsi -
ldl -lmux_ts -lmemcpymmxext -lmemcpy3dn -li422_yuy2_mmx -
li420_yuy2_mmx -li420_ymga_mmx -li420_rgb_mmx -lpthread
SRC=projectname.c
OBJS=projectname.o
PROG=projectname
projectname: ${OBJS}
                  ${CC}      $(includes)-o         ${PROG}  ${SRC}
          ${LIBDIR}${LIBS}
install:${PROG}
    ${INS}           -g         root       -o         root        ${PROG}
          ${INSDIR}
```

13.2. Antenna Characterisation for Amplitude Comparison in Electronic Warfare Systems[1]

Simon Brown$^{\alpha}$, James Irvine$^{\beta}$, Adrian Metcalfe$^{\gamma}$

$^{\alpha}$*Institute for System Level Integration*
Heriot-Watt University Research Park, Edinburgh, EH14 4AP, UK
simon.brown@sli-institute.ac.uk

$^{\beta}$*University Of Strathclyde*
Royal College Building, 204 George Street, Glasgow, G1 1XW, UK
j.m.irvine@strath.ac.uk

$^{\gamma}$*Teledyne Defence*
Airedale House, Acorn Park, Shipley, BD17 7SW, UK
ametcalfe@teledyne.com

22nd October 2010

13.2.1. Abstract

Radar warning receivers listen for pulses from radars of interest. The angle of arrival of each pulse is needed for pulse processing and location purposes. It can be accurately calculated using phase comparison, but this method is often impractical and expensive. This paper describes the authors attempts to use the less accurate method of amplitude comparison to provide adequate accuracy for use in a novel low cost radar warning receiver.

13.2.2. Introduction

The Radar is regarded as perhaps the biggest advance in remote sensing since the invention of the telescope [13]. Consequently since then methods have been sought to impair

[1] This paper has been reformatted for inclusion in this thesis. The original can be found on the ARMMS website [1]

radar performance and to turn radar operation into a weakness. Such activity is referred to as electronic warfare (EW) and plays a major part in modern warfare. An ideal radar has enough transmit power to be able to illuminate targets at the desired operating range, enough resolution to be able to separate targets of interest and is able to measure both the range and velocity of those targets. These requirements place constraints on the type of signal a radar can emit and thereby provide a useful set of characteristics an eavesdropper can use to identify a radar.

Such listening devices fall into two categories, radar warning receivers (RWR) and Electronic Support measures (ESM). RWRs are the simpler of the two and perform a platform protection role. They warn the operator of the presence of a radar of interest and may be able to indicate if they are being tracked by the radar. As an example RWRs are often fitted to military jets and are used to warn of the presence of enemy air defences, whether they are being tracked by the defences and whether an anti-aircraft missile has been launched in their direction. ESMs are more sophisticated in that they can provide all of the functionality of a RWR but also the ability to detect unexpected and characterise previously unknown radars. Their role is primarily intelligence gathering rather than protection.

As the exact nature and location of the radars of interest is not known in advance, RWR and ESM systems have to listen across the entire radar band and have antennas that can receive emissions from all directions. A radar pulse characteriser is able to measure the characteristics of individual radar pulses such as their width or frequency but some important data can only be inferred such as pulses repetition frequency and angle of arrival. The more data that can be gathered, the more exactly the radar can be characterised.

Angle of arrival (AOA) is often used as the key parameter for deinterleaving the received pulses. That is attributing a received pulse to the pulse train of one emitter rather than another. If it is known that two pulses have come from the same direction, the chances have improved that they have come from the same source. Once the pulse trains have been separated it is much easier to identify the emitter. There are two techniques commonly used to calculate angle of arrival, both of which rely upon comparing how the same pulse was received by multiple antennas. If the antennas are phase matched and are situated a known distance apart, phase comparison can be used. This technique has an accuracy of less than 1 degree but is less commonly used. As well as being more expensive it requires the siting of three or more antennas at multiple points around the platform. Expensive protection is fitted to expensive platforms which normally have a

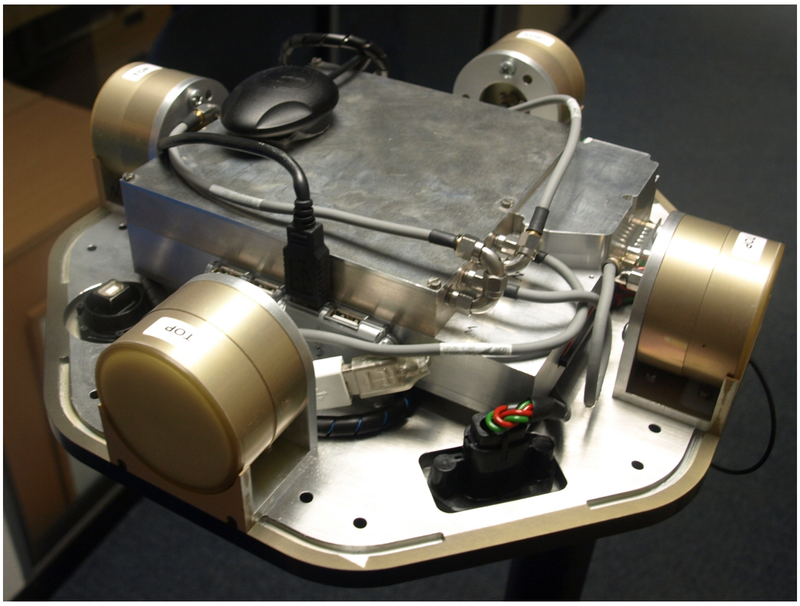


Figure 13.1.: Phobos Prototype

long service life. This makes it very hard to fit to existing high value platforms which were not designed around such antennas, and platforms such as jets with strict aerodynamic constraints. The comparatively inexpensive option is amplitude comparison which is normally quoted as having a 10 degree error [14]. It requires single antennas placed at multiple points on the platform and uses the known antenna gain patterns to calculate the AOA of the pulse.

Teledyne Defence Ltd has developed a new RWR/ESM system called Phobos. Phobos was designed to challenge the accepted view of the role of RWR and ESM systems by being significantly smaller, lighter, lower power and lower cost than anything currently available. This allows it to be fitted to lower cost platforms and to be used in new ways. This paper describes the authors' work to characterise the antennas on a prototype Phobos system to allow the measurement of angle of arrival accuracy.

13.2.3. System Description

The Phobos prototype consists of a RR017 [2] pulse characteriser, processing board, digital compass, GPS and 4 antennas. The prototype is shown in Figure 13.1 with the lid removed, photograph copyright Teledyne Defence Ltd. The RR017 is the device connected to the four antennas and rests on top of the computer. The RR017 has four ports but only two active channels. The two active channels are switched around the four

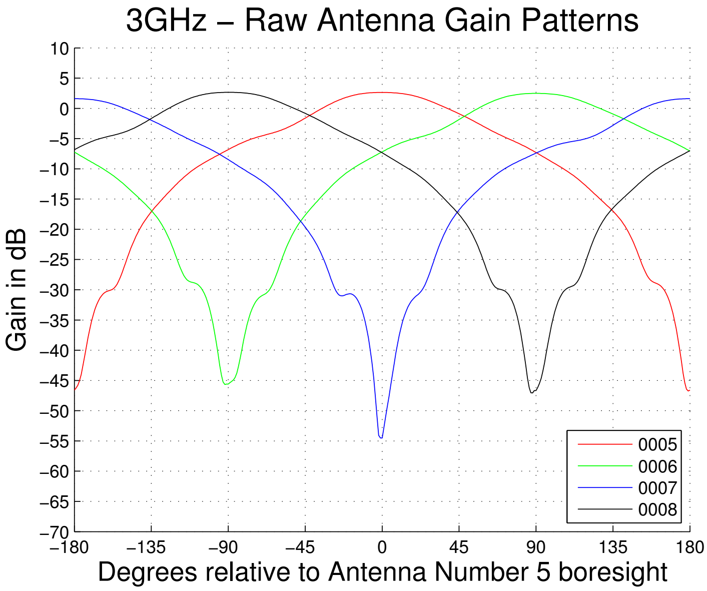


Figure 13.2.: Antenna Gain Patterns

ports to provide full 360 degree azimuth coverage. When a pulse is detected, the RR017 sends to the processing board a pulse descriptor word (PDW) which contains a variety of information including width and frequency of the pulse and the amplitude measurements from each channel. The processing board then passes the pulse descriptor words through an algorithm which performs both the deinterleaving and the identification step. For pulses of interest the AOA then needs to be calculated, duplicates removed and the resulting data passed to the operator.

The angle of arrival is calculated in a very simple manner. It is assumed that the only reason for a variation in amplitude between the two channels is due to the difference in antenna gain caused by antenna variation and orientation. Once the antennas have been characterised, the AOA can then be calculated. The antenna gain patterns at 3GHz are shown in Figure 13.2. Gain difference tables are calculated for the antennas for each degree in GHz steps across the operating band using the known antenna characterisations. The results for 3GHz are are shown in Figure 13.3 and over each ninety degree region of interest or quadrant the result is an approximation of a straight line. So to find the AOA in a quadrant two searches are started one from each end of the antenna difference table for that quadrant. If both searches find a match within the match tolerance and difference between the two AOA values is within the AOA tolerance a match is declared and the average of the two AOA values used.

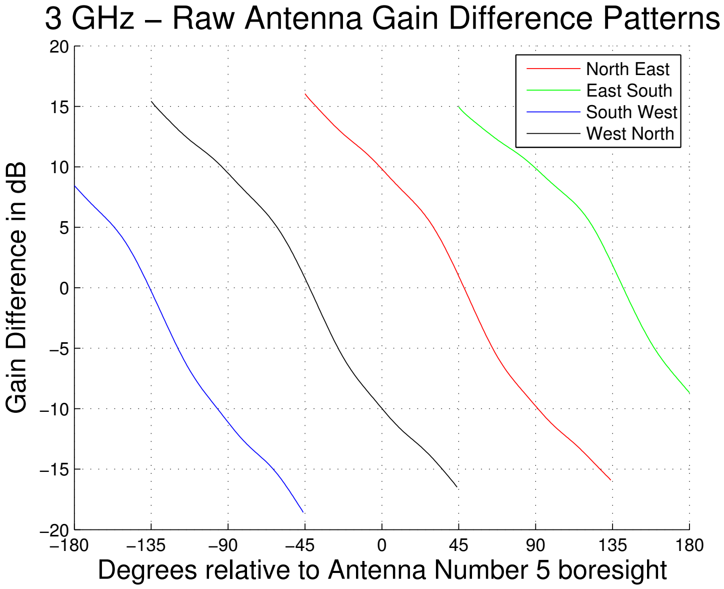


Figure 13.3.: Antenna Gain Difference

13.2.4. Antenna Characterisation

The antennas were supplied with amplitude and phase performance data. This allowed for a rough calibration for initial development work and as a reference for the system characterisation. The amplitude response is shown in Figure 13.2. The response for the four antennas although measured in isolation has been overlaid to aid comparison. The curves for each antenna are very similar, but shoulders can be seen for all antennas approximately eighty degrees either side of each antennas boresight. The effects of these shoulders can be seen in the gain difference plots shown in Figure 13.3. The shoulders seen in Figure 13.2 now correspond to flatter sections on the difference curve which leads to greater error as the flatter the curve the greater the number of degrees of error each dB of amplitude error equates to.

Absolute amplitude measurement error within the RR017 will not cause error in the AOA calculations as long as both channels are in error to the same extent. The amplitude tracking error for most situations is expected to be less than 0.5 dB based on the current RR017 build standard. This allows the AOA calculation error to be predicted assuming amplitude tracking error is the only component. The results for the antennas in isolation are shown in Figure 13.4.

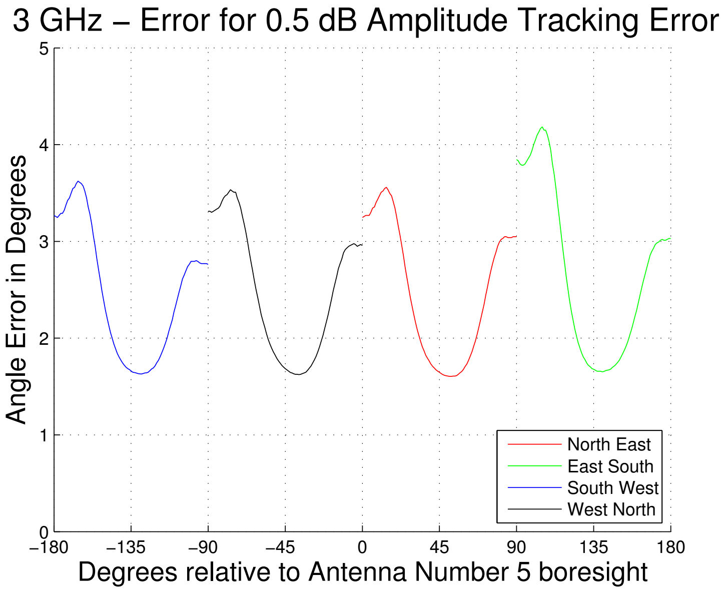


Figure 13.4.: AOA Calculation Error for the Antennas in Isolation at 3 GHz

13.2.5. Variation with the Lids

First attempts in an anechoic chamber to measure how the antennas performed once they were housed in the unit produced awful results. If the amplitude response for each antenna is not monotonic for the two ninety degree regions either side of boresight there maybe multiple places on the curve that correspond to the same amplitude difference. We supposed that the problems were due to scattering and reflection from the various metal parts within the unit. The unit contains a GPS receiver so must have a reasonably unobstructed view of the sky. To allow this the unit had a nylon lid. So to try and eliminate scattering and reflection the sides of the lid were covered with copper tape. This was only of marginal if any benefit. The measurements were then repeated with no lid at all and although still unusable this produced a better result than the first two lids. An aluminium lid was the next trial, knowing that the GPS would then have to be repositioned, which produced good results. The results for all four lids can be seen in Figure 13.5. A full characterisation was then attempted using the metal lid.

13.2.6. Results

The amplitude response for the system with the metal lid can be seen in Figure 13.6. The result is similar to before, but large shoulders can be seen on the East antenna's section

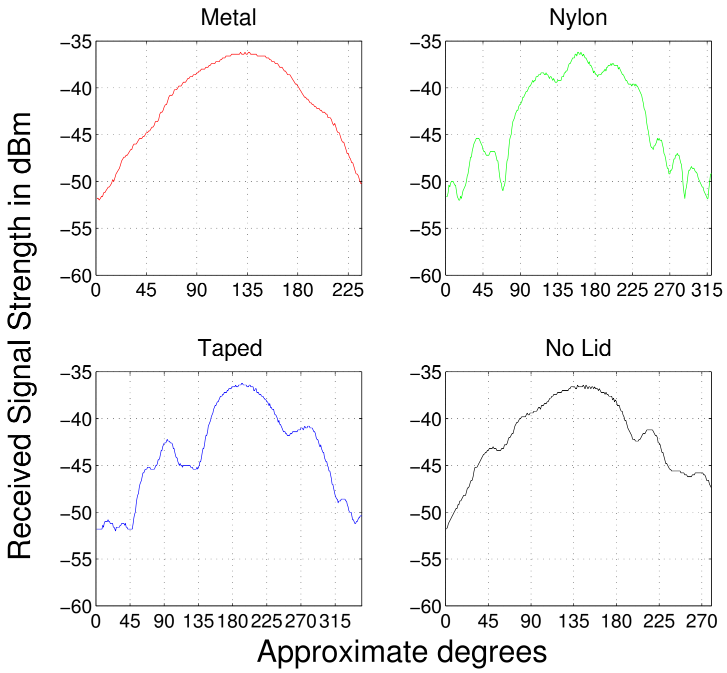


Figure 13.5.: Amplitude Measurements Taken at 3 GHz With Different Lids

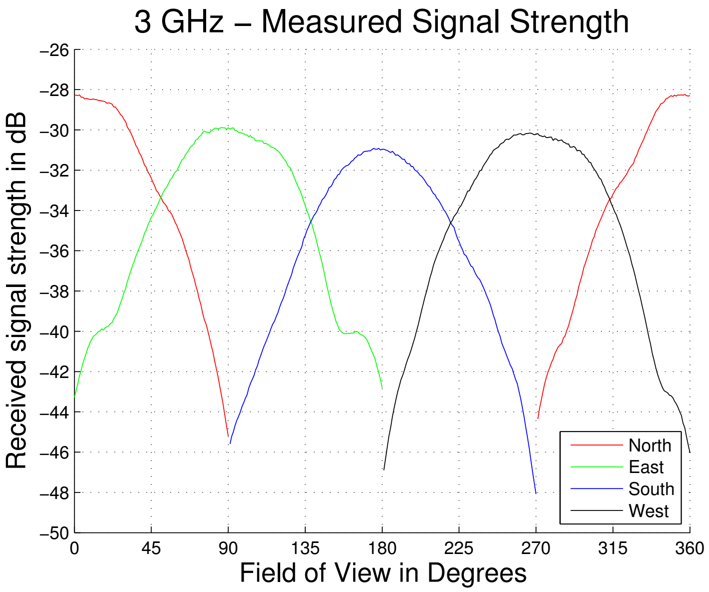


Figure 13.6.: System Amplitude Measurement at 3 GHz

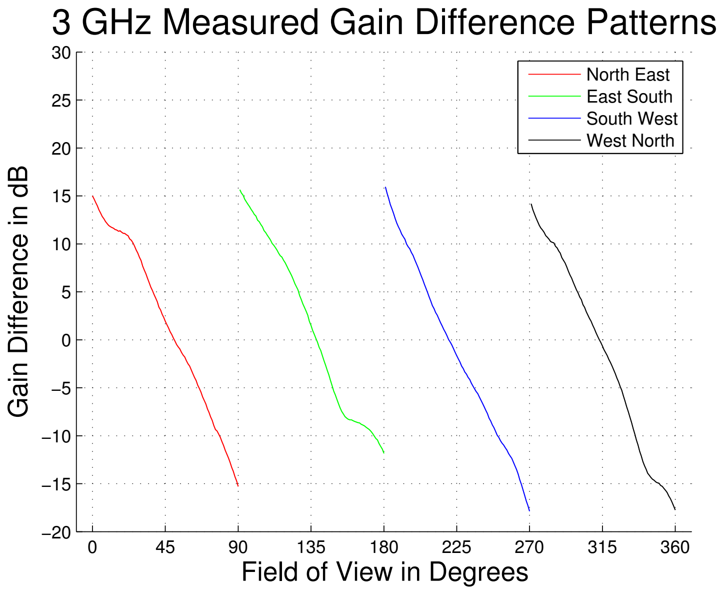


Figure 13.7.: System Gain Difference at 3 GHz

of the plot. Figure 13.7, the gain difference plot, shows the effect of the shoulders more clearly, with almost flat sections in 3 of the four quadrants. The resulting angle error plot is shown in Figure 13.8. Large spikes are shown which correspond to each flat spot on the gain difference plot. The error is less than three degrees for most of the field of view with an average of 1.88 and a worst case of almost seventeen degrees.

13.2.7. Conclusions

The purpose of this exercise was to better understand the process of AOA calculation and to provide a indication of the performance of a production unit. The design for the housing has been changed to take advantage of what was learnt when trialing the different lids. The production units will have an aluminium side with the antennas placed in aluminium holders to improve the shielding around them and to minimise internal reflection and scattering. The test equipment has been augmented with higher power signal sources and antennas for testing the higher frequencies to compensate for path loss in the test chamber.

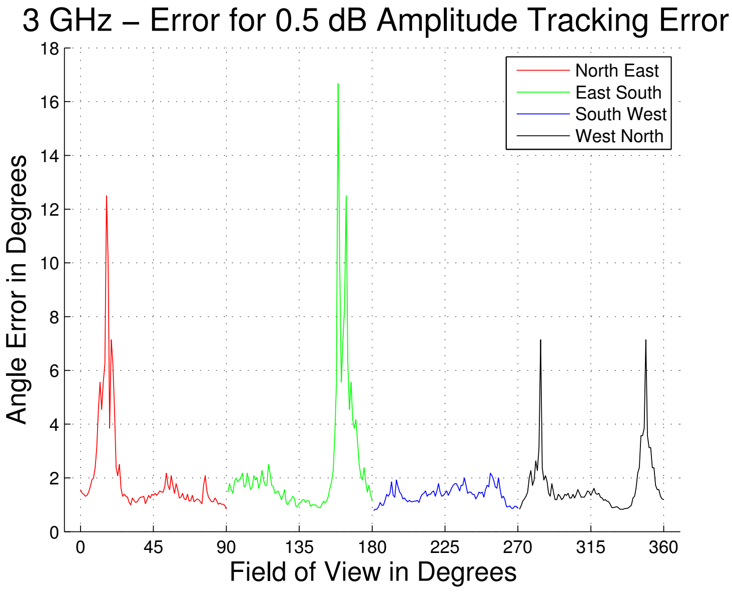


Figure 13.8.: System Angle Error for a 0.5 dB Tracking Error at 3 GHz

13.2.8. Further work

This work does not take into account the noise floor of the RR017 pulse characteriser. When the received signal strength on the weaker of the two channels is below the RR017's sensitivity, there is the potential for erroneous results as the apparent signal strength difference will be less than the actual difference. This is especially important when the emitter is not in the quadrant currently being observed. One channel may have a valid measure and the other is in the noise.

The performance of the current build standard of RR017s is significantly better than the one used in the Phobos prototype. Once the amplitude tracking error for the new units has been measured there maybe new information that could be taken advantage of. For example if it is shown to be strongly temperature dependant, temperature information could be used to set the tolerances used in the AOA matching search. As the gain difference curve is monotonic, the AOA matching search could also be improved by using a faster search, such as a binary search in place of the linear search.

13.2.9. Acknowledgements

The authors would like to thank Professor Raed Abd-Alhameed and Dr Chan See of Bradford University for their invaluable assistance with this work.

14. Conference Attendance Reports

14.1. Milcom 2007

14.1.1. Introduction

The Milcom 2007 conference was held from October 29 to October 31 at the Gaylord Palms convention centre in Orlando Florida in the USA. The purpose of attending this conference was to investigate the sensor network products available from other companies and the suitability of and interest in WiMAX as a military communications standard. Although there were numerous technical sessions, their content was in general too specialised to be of great relevance to this purpose so this report does not cover them. Detailed below are the two tutorials, WiMAX for the Warfighter and Tactical Wireless Networking Army's Requirements and Current/Future Force Capability Gaps, and a summary of the interesting products and companies from the exhibition.

14.1.2. WiMAX for the Warfighter

This tutorial was presented by Jim Orr who holds the title of Principal Network Architect at Fujitsu Network Communications. The presentation slides were also made available in software form. The material presented was not particularly true to the title – "WiMAX for the network operator" would have been a more accurate title, as there was very little reference made to war fighting. The standard was compared against its competitors from the telecommunications world: UMTS, LTE and UMB. This included higher level network infrastructure and spectrum allocation as well as the advantages and disadvantages of the air interface. The presentation was a good introduction to 802.16/WiMAX for those in the audience who were unfamiliar with it. The tutorial covered background information, the purpose of 802.16 and a summary of the two standards in use today – 802.16-2004 and 802.16e-2005.

The most informative part was the commercial information rather than the technical. The Mobile WiMAX certification process is just starting and Mr Orr predicts that Christmas 2008 will be when sales of consumer WiMAX equipment will take off. The Sprint Nextel Xohm network was briefly mentioned as an existing WiMAX network and Mr Orr regarded WiMAX's entry into the IMT-2000 group of 3G standards as very important. This should help to reduce spectrum allocation issues for commercial WiMAX network operators as they can now buy space allocated for 3G, but is of no direct relevance to the military. No mention was made during the presentation of the upcoming auction of the 700 MHz space in the USA. Operating WiMAX in the 2-5 GHz carrier range does not give a large enough usable NLOS range for a lot of applications and there is Military allocated spectrum at 400 MHz. A commercial network operating at 700 MHz would yield much useful information as to what could be achieved at these lower frequencies, particularly the range extension.

The security of WiMAX was touched upon and from a commercial perspective Mobile WiMAX provides more than adequate security using the EAP Authentication protocol and the AES algorithm for Payload encryption. There are a number of problems from the US military's perspective. AES is a suite B algorithm so can not be used for transmitting information of the higher classification types, though tunnelling may be permissible. More pressing is that the header information is unencrypted which makes the system more vulnerable to traffic analysis. It is preferred for some military communication systems for the entire air interface to be encrypted and for the system to be loaded with sufficient dummy data so that it operates at maximum capacity constantly. This is an attempt to make any meaningful traffic analysis impossible in the time frame that the information may be of use.

14.1.3. Tactical Wireless Networking Army's Requirements and Current/Future Force Capability Gaps

This tutorial was presented by Major Bryon Hartzog of the US Army who works at the Battle Command Battle Lab, Fort Gordon, GA. He has presented papers at previous Milcom conferences about trials he has carried out of WiMAX in an urban area [15]. The purpose of attending this tutorial was to discover what communication needs the US military may think it has and with the speakers background in the area, whether he thought WiMAX would be a good fit for any of those needs. Major Hartzog also offered the slides in electronic format for those interested once it had been approved for release.

The approval for release to non-US parties has not yet been given.

14.1.3.1. Finding of previous Studies with regard to 802.XX technologies

The 2001 Army Science Board (ASB) Ad Hoc Report recommended to "invest more in wireless infrastructure based on commercial advances." The ASB Task Force 2006 Report on Wireless Tactical Networking reported that 802.XX technologies are now being used successfully, if in a rather chaotic manner, despite their security short comings and unknown performance characteristics. They are attractive as they can be rapidly deployed at a lower cost and worrisome as they are well known to the enemy and driven by commercial rather than military priorities. The chosen route is one of increased involvement by the military in the standardisation process to make the standards suitable for military purposes and to ensure they remain so. This is to hopefully eliminate the need for military customisation below the service layer. The security problems remain and the situation is unlikely to change as there is no need for such security in commercial systems. To obtain all of the possible benefit that the 802.XX technologies can offer, the US military has realised that it must accept them as they are as the cost of maintaining their own version of a standard would eliminate most of the savings. This brings new problems as issues such as spectrum allocation are now far harder to solve. Consequently it seems that the intended use will be for low security networks where a large amount of data of strongly time limited value must be exchanged, rather than as a replacement for any core backbone. If there is success at this level though it is hard to see how 802.XX and other open systems could be ignored for future core network technologies. As the network complexity rises, the cost of maintaining a private version may ultimately be appealing against that of a bespoke standard, despite the compromises it entails.

14.1.3.2. Future Force Requirements

The Warfighter Information Network - Tactical (WIN-T) and the Joint Tactical Radio System (JTRS) are the two systems which will deliver this capability: WIN-T connecting the higher organisational units and JTRS linking the higher units to the dismounted soldier. JTRS will also be able to provide waveform compatibility for legacy and interim systems such as SINCGARS – the SINgle Channel Ground and Airborne Radio System. JTRS will however not be a single system. Three variants are planned: Airborne, Maritime/Fixed and Ground. Despite the obvious interest of the US military in 802.XX technologies it was not at all clear how it perceived making use of them. This suggests that they intend to carry on as they have been doing with units buying 802.XX equipment from operational and discretionary budgets whilst centrally investing more money in bespoke systems.

14.1.3.3. Product Evaluation

Major Hartzog's role in this is as a "Green suit" military adviser at the Battle Command Battle Lab, one of a number of evaluation centres where equipment vendors may bring their products for evaluation. The process is however biased toward larger companies who can participate in 10 or more projects as they are treated in a more collaborative manner.

14.1.4. Exhibition Floor

The commercial exhibition ran in parallel to the technical sessions. A large number of companies were exhibiting, amongst them a number of smaller companies, whose products were at the fringes of military communications, as well as the expected large defence companies.

14.1.4.1. WiMAX

The only company present that was heavily promoting WiMAX was Aeronix of Melbourne, Florida. They have developed an 802.16+ modem which is 802.16d-2004 based product which includes their own extensions such as AES header encryption, lengthened cyclic prefix to allow use at 200 knots and additional modulation modes. The version designed for UAVs, the UAVe Digital Data Link, will also be upgradeable to 802.16e-2005 when their solution is complete. The company is very positive about WiMAX and has obviously invested a lot of development effort in the products. To overcome the military problems their products are however no longer 802.16 compliant. So when operating in military mode, will not be able to inter-operate with other vendors equipment.

BAE Systems of Wayne, New Jersey, was the only other company with WiMAX products on display. They have developed an 802.16d-2004 solution that also has propriety extensions to solve the header encryption problem. Interestingly, they have also developed a multi-hop meta-MANET extension, but details were unavailable. The representative

I spoke to was not as enthusiastic or knowledgeable as the Aeronix representative and when I asked about their plans for 802.16e-2005 support, he had not heard of the standard. He also could not tell me if their meta-MANET extensions were connected to BAE Systems involvement with the 802.16j Mobile Multi-hop Relay (MMR) group. They have created a number of products based around 802.16d-2004 in different form factors, with and without meta-MANET and combined with another modem to create a bridge.

14.1.4.2. Ad Hoc Networks

Harris of Rochester, New York State, have a range of sensor network products under their Falcon Watch brand. The simpler units use a propriety ad hoc network system in the 30-108 MHz and 138-174 MHz bands and capture voice, seismic, PIR or Magnetic data. The battery life is between 1 month and 1 year dependant on the number and type of sensors connected. The units also report their health, battery status and whether they have been tampered with. Higher level gateway units also exist which provide the bridge to the satellite back haul. They also have a camera interface to which a camera can be connected and operated remotely. The camera provides infrared and low light level mode. Harris supply a PC based management application and there is a degree of integration with their other products.

Raytheon of Fullerton, California, were exhibiting their Microlight family of ad hoc voice and data radios. This family is capable of creating a self healing ad hoc network enhanced by automatic relay and multi-hop capability. It operates in the 420-450 MHz band and provides a capacity of 1Mbps with the air interface encrypted using AES. The units contain embedded GPS receivers to improve the situational awareness of commanders. The EPLRS-1 unit provides the command interface to this facility.

14.1.5. Conclusion

As somebody wholly unfamiliar with the US military's procurement strategy before the conference, the chaos of it all was surprising. With regard to the bigger picture, the breakfast and lunchtime speakers provided a lot of frank and useful information. The degree of incompatibility between the armed services was staggering. In his lunch time talk Jeb Bush, the former governor of Florida, talked about his experiences trying to coordinate the emergency response to two consecutive severe hurricane seasons. He described a very familiar situation of different groups from different levels: federal, state, city and county, who normally lead separate existences suddenly having to cooperate so they can respond together to the hurricanes. He recounted his arrival at the state's emergency response centre to be told they could communicate with exactly nobody outside the building. He then devoted a large amount of effort as governor to ensure that when the next hurricane arrived the response would be better. When hurricane Katrina threatened the gulf state area, the Florida emergency teams and law enforcement agencies were then able to assist their neighbouring states using the experience they had gained.

There are 1.4 million servicemen and women in the US armed forces and many more civilian staff and reservists. At all but the highest levels the services operate independently as they are of sufficient size and, with the exception of the Marines, designed to fulfil different roles. Their equipment cannot be entirely centrally planned and bought as the commanders in the theatres of operation need a large amount of flexibility as each deployment is different. Another lunch time talk was given by a Brigadier General who works at the US National Security Agency (NSA). He showed a film made by a US high school teacher about the changing pace of modern life entitled "Shift Happens."[1] He explained that historically electronic intelligence gathering was much easier, it just had to be decided which phone lines to tap. Today information transmission is plentiful and in many different forms, which makes it much more difficult to decide what to collect and how to arrange the collection. The military commander today is suffering from a similar information overload. The amount of battlefield information that can be collected is massive compared to a few years ago and getting it to where it may be of use, whilst it is still useful, is a great problem.

Independent procurement allows the evaluation of a number of similar systems in parallel, facilitating a direct comparison. The problem it seems is that, although this information can guide future policy, the US military is unable to then standardise on the best of the current generation due to the amount of time and money already invested in the competing systems. Instead that system can be standardised at the next generation, but compatibility must be introduced to support all of the previous systems. This is the reason for JTRS, the software defined system that can support a large number of standards.

Open technologies seem unlikely to be widely adopted until they are widely adopted, a situation only resolved by outside influence. IEEE 802.XX are disruptive wireless tech-

[1] Available on YouTube, search for "Shift Happens" (Original source unknown)

nologies for the US military. They relegate the military to being a second tier customer, do not follow traditional procurement methodology and are available to the enemy. They do however offer a price that can not be beaten. One conversation topic that could be heard often in the private discussion at Milcom was that of tightening budgets. Open standards may not be attractive today, but they are becoming more and more attractive as system complexity and hence cost rises. Not only are systems becoming more complex, but there are ever more of them. Price may be the outside influence capable of breaking the current adoption deadlock. The easiest route for the adoption of open technologies will be through isolated systems. A WiMAX equipped UAV would be a good example. Once they have been successful in isolation, there will be less resistance to their further adoption.

The exhibition floor demonstrated that there are available today a number of mature WiMAX, ad hoc and sensor network solutions but the market is far from open. The major players are entrenched with their own systems and with the exception of Aeronix there was little obvious enthusiasm for open standards. The principal advantage of WiMAX is cost and that benefit will not be fully realised until consumer equipment is being produced in large quantities to bring the economies of scale. That is not likely to happen until Autumn 2008. Sensor networks exist at the fringe of both the network and the price list of the major players in defence. Although there is innovation going on and a need for better systems, they will not be a headline item, the technology will follow rather than lead.

14.2. AOC Convention and Symposium 2008

14.2.1. Introduction

The 2008 Association of Old Crows (AOC) annual convention and symposium was held in Reno, Nevada, from the 20^{th} to the 23^{rd} of October. There were two purposes of attending this conference: firstly, to become more familiar with what is happening in both technological and commercial terms at the leading edge of electronic warfare (EW) and secondly, to assist in the launch of the QR020/Phobos. The QR020/Phobos launch was however, sadly postponed as the demonstration hardware that was sent to Reno was stopped at customs.

14.2.2. Radar for EW Engineers

This course was given by Dr Richard Wiley of Research Associates of Syracuse Inc. The course was based around the book "Introduction to Airborne Radar" by George Stimson [16] and used images from it extensively. The course as its title suggests provided a broad overview of radar systems but from an electronic warfare perspective rather than a radar perspective. It started with an overview of the physical characteristics of radar signals and the environment they operated in, frequency bands in radar and EW terminology, laser through to HF, the relationship between beam width and wavelength, atmospheric effects and rain, multipath effects, apertures and antenna gain. The discussion of antennas then lead onto mechanical and electronic beam steering and target detection. Dr Wiley explained that unless the radar's beam width was very narrow, simply getting a return was not enough to provide accurate bearing information. Sequential lobing and phase or amplitude comparison monopulse are used to provide much more accurate bearing by comparing the return from two channels that operate in parallel or sequentially. System design was the next topic, covering unambiguous range and Doppler, energy versus power, peak and average power, duty factor, receiver noise figure and of course the radar range equation.

Toward the end of the session as a numerical example, Dr Wiley talked about what he termed 'quiet' radars. He dislikes the term low probability of intercept (LPI) as there doesn't seem to be a standard definition of low. In his terminology a quiet radar is one where the target detection range for the radar is the same as the interception range for a ELINT receiver. As the received power at the ELINT receiver varies with the second negative power of range for a given transmit power and with the fourth negative power for the radar receiver, the ELINT receiver has an advantage. However, the ELINT receiver is unaware of the characteristics of the transmitted signal in advance and wants to preserve as much signal information as possible. It must listen over a wider channel bandwidth and is unable to use matched filtering, which puts it at a disadvantage with regard to receiver sensitivity. The range of a quiet radar taken from [4] p.216 is shown below. R represents range, S – sensitivity, G_R – radar receive gain and SLR – sidelobe to mainbeam ratio.

$$R_{Radar} = R_{ELINT} = \left[\frac{S_{ELINT}}{S_{Radar}} \times \frac{\sigma(SLR)G_R}{4\pi}\right]^{\frac{1}{2}}$$

Dr Wiley stated that current LPI radars can detect targets of military interest at ranges of up to about 20 kilometres, beyond that range equipment exists that can intercept such radar signals.

14.2.3. Conference Sessions

In contrast to Milcom very few of the sessions were concerned with technical detail. The main themes of the sessions I attended were maintenance, obsolescence and interoperability. Colonels Schwarze and VanderWerf of the United States Air Force talked about the fact that the vast majority of funds for equipment is spent on maintenance not new equipment. They talked about being held hostage by legacy equipment which can not be easily upgraded or replaced. The war in Iraq has also shown the army's EW capabilities to be lacking to the extent that they have had to call upon the other services for assistance which has also highlighted interoperability concerns. The US military believes it can solve most of these problems by moving toward software and digital technologies based around open architectures. The rest of this section is divided between the two themes of procurement and leveraging modern technology. This is an attempt to understand why money is spent in the way that it is and how recent technological advances can reduce the maintenance burden of both current and future systems.

14.2.4. Defence Procurement

Due to the advances being made every year in consumer electronics technology the armed forces of the world, despite their seemingly massive procurement budgets, are struggling to keep the equipment they field up to date. This manifests itself in the short term in two ways. The warfighters prefer to use consumer equipment as it performs better and the military equipment becomes ever more expensive to maintain due to part and technology obsolescence. The US air force still fields systems that are valve based and struggles both to source valves and recruit technicians familiar with valve technology. Replacing the outdated systems with modern equivalents is far from easy. Large military procurement programs may take a number of years to deliver by which point any consumer part used may have reached end of life and the technology may be obsolete so the equipment is out dated before it enters service. In the past ten years consumer Internet access technology in the UK has gone from dial-up to ADSL broadband to wireless broadband. The Clansman communications system was introduced to the British Army in the early 1980s and was planned to be replaced by the Bowman system in the early 1990s. The Bowman system came into service in the early 2000s, £500 million over budget and has been described as "astonishingly bad" [17]. Clansman equipment is still preferred by some soldiers despite being unencrypted and not providing location information. Bowman has entered into military slang as an abbreviation of "Better Off With Map And Nokia."

The reasons behind the dismal success rate of military procurement programs to deliver useful equipment on time and to budget have been investigated with interesting results. Defence planners are always fighting two wars: the war of today and the war of tomorrow, referred to as tactical and strategic planning. Tactical defence procurement is presently concerned with the wars in Afghanistan and Iraq, the EW portion of which is almost entirely concerned with CREW (Counter RCIED (Radio Controlled Improvised Explosive Device) Electronic Warfare). Strategic procurement is concerned with the equipment needed for the next war based upon what was learnt during the last war, the Cold War, which itself was predicted to be similar to the Second World War. The tactical equipment is needed within time frames of days, weeks and months to respond to an unforeseen immediate need and the strategic equipment is planned years in advance of when it might be needed and has to incorporate any advances made in tactical equipment and relevant technologies during the procurement process which tends to delay its delivery.

14.2.4.1. Tactical Procurement

It can never be entirely predicted what equipment will be actually needed until the war has begun or whether the provided equipment will function as intended. Tactical procurement attempts to quickly fill any capability gaps discovered by warfighters who are already in theatre. The immediacy of the need and the tendency for the items of equipment to be low in value in relative terms means that instead of a formal tendering process, the US DOD approaches a contractor directly that is known to have expertise in the required area. This immediate need creates a long term problem in that over time a large number of systems are developed for the four US armed forces that perform a similar function yet are completely incompatible. These are referred to as "stove pipe" systems.

14.2.4.2. Strategic Procurement

The bulk of military procurement funds are spent on strategic equipment, the core equipment that is used to fight all wars but has been designed for conventional wars like the Cold War and the Second World War. Billions of dollars are spent each year on equipment to fight the strategic war which may be of little current use or represent exceptionally poor value for money when used in current wars. Such equipment is however believed to be essential for fighting other enemies should the need arise. The procurement process for items of strategic equipment may take years or even decades to complete. Such a long time is required for many reasons and often the actual time required to manufacture, which may be years for an aircraft carrier, is not the biggest element. The militaries of the world are not technologists, they have delegated that role to private industry so when they want a new piece of equipment, they only have a functional specification and may not be aware of what can actually be achieved. The US military never sets out to buy the world's second best piece of equipment, they aim for not just the best they can, but the best in the world. This combined with the future in service date means that they must predict where the leading edge of technology will be. Consequently new equipment is designed around unproven and emerging technologies, the associated risk of which is a big contributor to delays, cost overruns and failures. As the procurement process lengthens, a larger number of tactical advances must be incorporated into the design which adds more delay and increases the risk that the requirements might have changed.

14.2.5. Modern Technology

Very few complex systems are ever designed from a clean sheet, they build upon knowledge gained from similar systems that have gone before. This technology reuse however remains within that particular company and is not shared. Colonel Schwarz states that the USAF currently fields "67+ Systems, 34+ Languages, 19+ Models/Platforms" of EW systems. Consequently if this maintenance burden can be reduced, there could be a lot more money for product development. Some parts of EW systems require bespoke hardware such as analogue IFMs, some however such as signal processing commonly now make use of modern commercial technologies and interfaces. If all defence contractors could start building systems around the same technologies and interfaces, physical systems could be reused between platforms. The USAF wants to consolidate the number of

EW systems fielded to the minimum needed and importantly when they become obsolete, be able to simply discard them and slot in a new one.

14.2.5.1. Open Architectures

It is envisioned that at some point in the future open architectures would make it possible, in a plug and play manner, to equip any flying platform with the EW systems needed for that mission immediately prior to take off. The advantage of platform based strategies to the prime contractor is that they control the platform completely and can behave as a monopolist with regards to any future changes. This means that although they may lose money developing the platform, there will be rich pickings available once it is fielded. Therefore it is likely that open architectures will only be introduced in the UK in a new platform as there is no incentive for the current platform owners to retrofit access for their competitors to their platform. In the event of an open platform being created, who will be in charge of integration and compatibility testing? The MOD has in recent years sold a significant amount of its science and engineering expertise to private industry. It seems unlikely that they will be willing to take on this responsibility, but if it is contracted out will they be capable of ensuring the objectivity of the testing?

If the platform becomes open, who will own the user interface? If the on-board systems are interchangeable at the hardware level, there must also be an open user interface standard, so that the user interface can be configured to make use of the new hardware, otherwise whoever controls the user interface will control the hardware. Cock pit display systems are regarded as safety critical which makes it very expensive to modify the code base due to the testing involved. The MOD would have to be prepared to specify systems to a greater level, breaking the platform down into subsystems and taking ownership of the interfaces in between. The MOD has its architecture framework MODAF [18], but it is very much at the system level and above. Open architectures would potentially open up direct access to MOD money for small contractors such as TDL creating much more competition in the market which should provide better value for money for the MOD.

The DOD has realised that an easy way to make COTS equipment more suitable for the military is to actively participate in standardisation processes. Extending consumer and industrial standards instead of recreating them removes the majority of the interoperability testing problems as the DOD only has to manage testing for the military extensions, the remainder can be carried out in commercial test houses. A COTS based open architecture has the possibility to offer low costs, but there is however a problem. Consumer standards have a very short lifetime in comparison with military systems. Part obsolescence is already a big problem in the maintenance of military systems and is likely to get worse as the rate of obsolescence increases in consumer electronics. As long as the interfaces are still supported open architectures should ease the problem as it should be easier to replace part obsolescent systems

14.2.5.2. The Importance of Performance

A point made by a number of the speakers was that performance is not that important. They bemoaned industry's fascination with bigger, better and more expensive when they would really prefer smaller, lighter, lower power and cheaper. SWaP is the term used to refer to size weight and power.

14.2.6. Advanced Tactical Electronic Support and RAF Spadeadam.

In general the presentations gave a valuable insight into what is currently perceived to be the problems and what might be the solutions. There was little that was of direct relevance to my work at TDL with two notable exceptions, the presentations by Steve Pizzo from the US Army's CERDEC and Wg Cdr Paul Wallace of the RAF. Steve's presentation was entitled Advanced Tactical Electronic Support (ATES) and described the need for a device very similar to the RWR except operating at VHF-UHF bands and used to identify radio communication emitters rather than radars. It may be worth pursuing this with DSTL to see if they have any algorithms for identifying radio communication emitters. A low band digital IFM could be used in place of the the RR017 and the device could be potentially very small.

Wg Cdr Wallace is the station commander of RAF Spadeadam which is the RAF's EW test range. They have a large amount of mostly Soviet air defence equipment which they use to provide realistic threats for EW training exercises. They even have a SCUD lurking somewhere in their 9,600 acre training area, which is located to the north east of Carlisle. Wg Cdr Wallace was at the conference to promote facilities available at RAF Spadeadam to industry. Once the RWR system is complete it may be appropriate to seek out more realistic test environments.

14.2.7. Exhibition

The exhibition contained a mix of the expected large defence contractors and a significant number of smaller companies who sell to markets other than defence. I set out to see what was being offered in the way of embedded processing solutions and to see if they could potentially be used in the RWR system to replace the PC104 computer.

14.2.8. Iveia

Iveia [19] sell a number of credit card sized processing modules with complementary IP to ease their use. The modules are called the Titan series and have either a Xilinx Virtex 4 or 5 FPGA, or a Freescale Power PC processor with a Xilinx Spartan FPGA. Their Velocity SoC core provides a number of controllers for the integrated peripherals and an interface to code running on the Power PC processing core that may be present. They were very positive about using the Power PC hard cores found on Xilinx FX series FPGAs and explained how their Velocity SoC core was designed to help the splitting of processing between software running on the processors and algorithms implemented in VHDL on the FPGA. The problem with buying a generic board is that it will always have unwanted features and it is hard to judge in advance if the convenience is worth the compromise. If TDL decides to develop a generic FPGA based processing card, this company would be one to further investigate.

14.2.9. Annapolis Micro Systems, Inc.

Annapolis Micro [20] have a completely different outlook to Iveia. Their products are similar but are larger with multiple FPGAs designed around standard PC interfaces such as PCI Express rather than targeting the embedded market. They also have a very low opinion of the Power PC cores found on Xilinx FX FPGAs describing them as useless. They have also reimplemented most of the standard cores such as Ethernet, FFTs and filters and sell a graphical system design tool called CoreFire to make use of them, without the need to write any VHDL. It became apparent that they were targeting customers with high performance signal processing needs, selling their boards as FPGA accelerators rather than as part of an embedded system. It is unlikely that such a board will be useful to TDL in the near future unless a board is needed in a SMART [21] like system with say a VME interface. Annapolis Micro is unlikely to be the only player in this market and it may be more suitable to seek out a UK board manufacturer.

14.2.10. Conclusion

This conference is much more aligned to the work of TDL than Milcom and was generally much more informative. Milcom however being a IEEE backed event had a much more academic feel and provided the usual presentation opportunities. All of the speakers at the AOC convention were invited and there were only 2 concurrent sessions. So although the AOC Convention is a much more relevant event, it does not provide any mechanism through which work from this EngD project could be published. There are no conference proceedings and the majority of non military speakers did not even provide their slides to be published on the AOC website. There was a small poster area from the Mercer engineering research centre in the exhibition hall, but it was largely neglected and in the absence of proceedings the work is unpublished. It may even have been just a simple shop window for their consulting activities. It would be great to go back next year with the completed RWR as was planned this year and talk about it. It may however be more prudent to find a different conference where at least there is a chance of a paper about the work being published.

15. Internal Reports

15.1. Data Link Strategy for Filtronic Defence Limited

15.1.1. Introduction

Within Filtronic Defence Limited (FDL) there has been identified a need for information about, and research into, data link technologies. This has arisen from the desire to improve the RR017 product offering and a strategic need to address the MoD's vision of a Network Enabled Capability (NEC) [22] or Network-centric Warfare using the American terminology. A cluster of RR017 units that were aware of their location and orientation, and could share information effectively would be able to make use of the spatial diversity of the cluster to create a more detailed picture of the radar environment. Solving the same problem of being able to share information effectively and being able to act upon it, but on a much larger scale, is the aim of NEC.

The recent advancements in commercial wireless technology such as UMTS and 802.11 WLAN make it very attractive, for reasons of cost, to reuse these technologies rather than for the MoD to fund the creation of a bespoke wireless standard. The problem is however that the primary aims of commercial and military networks differ. Features which may be required in a commercial network may be undesirable in a military network. Commercial networks require all links to pass through their infrastructure prohibiting direct user to user communication so that they can accurately bill for all services provided. Conversely ideal military networks need to be able to operate in an ad hoc infrastructure-less manner in order to provide the most resilient network.

15.1.2. The MANET problem and Co-operative Networking

A class of networks, known as mobile ad hoc networks (MANETs) are ideally suited to the needs of the military as they provide the resilience required. A lot of research effort has been devoted to this topic largely based around the 802.11 WLAN standard as it includes an ad hoc mode. The rate of progress has been slow and the networks created do not scale [23]. Significant work still remains in the areas of medium access control, routing, management and security [24]. As an example in a hierarchical network, at each level in the hierarchy, if the destination node for a packet is unknown, it is sent to the next level up. In a flat ad hoc network, each node must maintain a table of all known other nodes and gateways and how to reach them. This table must also be kept up-to-date as the network topology changes. MANETs also have a significant non-technical failing, that there is no guaranteed connectivity. Potential customers are likely to be reluctant to pay money for a service they may not be able to receive and an alternative business model has yet to be found.

Commercial wireless network operators are also experiencing difficulties as their customers demand ever higher data rates and better coverage. Both can be achieved by adding more base stations to the network, but this increases operating costs and the erection of base stations is becoming harder, due to the shortage of suitable sites and public opposition. Instead, the operators are looking to take advantage of their customers to provide coverage extension [25]. The coverage of a hot spot in a cafe could be extended to the bus stop outside by the person sat at the window allowing the operator to transfer data to the subscribers waiting for the bus by a another route. Co-operative communication networks share some of the advantages and disadvantages of MANETs but as the number of relay hops is constrained, so are the problems of scale. Currently the commercial operators are focusing on dedicated relays, a cut down, low cost, smaller base station, not combining them with the user terminal.

15.1.3. 802.16j Mobile WiMax with Mobile, Multi-hop and Relay Functionality

802.16 is the Wireless Metropolitan Area Networking (Wireless MAN) standard of the IEEE. There are currently two published versions, 802.16-2004 known as Fixed WiMax and 802.16e-2005 known as Mobile WiMax [26]. The 802.16j task group [27] is currently developing a modified version of the Mobile WiMax standard which will support a form of cooperative networking. The US military are actively participating in this working group, both directly and indirectly through the MITRE corporation and BAE Systems [28], so that the resulting standard is suitable for their needs. They have added a number of usage scenarios to the specification which are specific to the military. The standard is not yet complete but should deliver a good compromise of range, capacity and resilience if operated at a low enough frequency. The 802.16 standard has a lot of backing in the US with both Intel and Motorola developing hardware and the Sprint network is starting to roll out coverage in major US cities [29]. The US is also auctioning off the recently freed 700 MHz band [30], which if used would make WiMax very competitive as broadband wireless access technology. The 802.16j standard is currently at the first draft stage and not available to the public so its detailed contents are unknown. There are also likely to be further drafts before the standard is approved.

15.1.4. FDL Strategy

FDL's stated aim is to move from being a component supplier to being a tier two systems and sub systems supplier whilst also growing the components business. In the short term these systems and sub systems are likely to be constructed solely from FDL components, but in future third party components of ever increasing complexity will have to be integrated to sustain growth. The need for data links, both wired and wireless, is only going to grow. As FDL's systems grow in complexity they will become a standard feature. Mobile WiMax support could be added to the RR017 product today, single chip WiMax processors are available, which system integrators can use to create PCMCIA cards and USB dongles, that could be adapted for the RR017 [31]. 802.16j solutions are not available yet, but companies such as PicoChip are developing them [32]. Crucially all relay stations may not be the same depending upon their role. The 802.16j documentation available so far lists a number of usage scenarios of differing complexity and it is likely that all relay stations will not be required to support all scenarios as this would have a detrimental effect on unit cost. A static relay providing coverage extension at the edge of a cell will be far simpler than say a train with a relay in each carriage and a continually changing radio environment as the train moves at high speed between base stations. Scenarios such as air dropping a number of combined Relay Station / Subscriber Station sensor nodes which will then self organise in the most battery efficient manner and connect to infrequently passing aircraft, may not receive much commercial attention. WiMax software companies may be reluctant to invest developer time in such a niche market, which is where FDL could add value.

FDL is currently creating a concept demonstrator for the RR017 with wireless data link in conjunction with the Wireless CIC to demonstrate its activity in this area. There is no FDL technology in the data link, it is just an off the shelf integration that show cases the possibilities and the supporting software that FDL has developed. The system is 802.11 based so is limited in that it can not provide the range or capacity required. If there is customer interest in this demo it could easily be upgraded to Mobile WiMax, the main foreseen cost would be the WiMax base station.

15.1.5. Conclusion

WiMax delivers the best combination of range and capacity of the commercial wireless standards available today. The current version of the standard does not facilitate the resilience required for a military network, but the 802.16j working group is addressing this problem. This new version of the standard is currently at the first draft stage and will hopefully be released in the next 12 months. Both BAE Systems and Selex are working on or have WiMax solutions, with BAE systems being actively involved in the 802.16j working group. The 802.16j standard is likely to include usage scenarios which are specific to the military and may not be supported by COTS equipment. FDL has a large amount of RF expertise and could integrate a third party WiMax processor into one of its systems or create a data link product. Integrating a third party WiMax processor would add wireless data link functionality whilst minimising risk, but FDL may not then have control over the feature set and scenarios supported. Licensing a 802.16j software implementation would give FDL the freedom to optimise the data link for their chosen scenarios and consequently a means of differentiating their product from their competitors.

In the short term FDL should use the Wireless CIC developed data link as a demonstration tool to gauge the level of interest in wireless data links of its customers.

15.2. Cypress EZ-USB FX2 Firmware and HDL Customisation

15.2.1. Purpose

Filtronic Defence Limited (FDL) is trialing the option of USB interfaces on its products, in place of the variety of serial and parallel interfaces currently in use. USB was chosen due to its ubiquity, low cost and because USB 2 High Speed can support the data rates required. The first product to be fitted with this interface will be the RR017 and the first use was to be the wireless data link project undertaken with Bradford University Wireless CIC.

15.2.2. Problems

- The chosen USB interface chip from Cypress Semiconductor contains an enhanced 8051 processor. This is to facilitate interfacing other interface standards to USB 2 in the absence of a host processor. For this purpose the 8051 will play no active role in the data transfer, operating in slave FIFO mode. The 8051 processor is only required to configure itself and the USB endpoints at start up. Firmware is required to carry out this configuration and initialisation.
- As the RR017 project was delayed, hardware was unavailable for testing the work done by the Wireless CIC. Consequently another board had to be sourced that could be used to provide a data stream for the Wireless CIC work whilst matching the actual RR017 hardware as closely as possible to minimise development effort.
- As a substitute RR017 board is being used the existing HDL will need to be re-targeted and modified.

15.2.3. Board Selection

A large number of development boards are available which are similar to the RR017. The requisite components are, the Cypress EZ-USB FX2 chip, a Xilinx Spartan 3E FPGA and a Xilinx PROM. The Selection was narrowed down to three which were the closest match, from Opal Kelly [33], Orange Tree Technologies [34] and Digilent [35]. The boards from Opal Kelly and Orange Tree Technology were expensive at $199.95 and £179 and contained a Spartan 3, not the 3E part. The Nexys 2 board from Digilent is fitted with the same FPGA as the RR017, just in a different package and costs only $99 so was without compromise the best substitute. As well as USB controller, FPGA and PROM the board is also fitted with Flash and SDRAM so may be of use in later projects.

15.2.4. Development

15.2.4.1. HDL Modification

Tim Roberts provided a cut down version of the VHDL he had written and successfully simulated for the RR017 which only contained the components required to operate the USB interface and test mode. As the FPGA used on the Nexys 2 is supported by the Xilinx Webpack [36] no new software licenses were required and the FPGA development environment could be downloaded from Xilinx's web site. Before the VHDL could be modified, the tool chain needed to be checked along with the device programing tools and the board. This revealed the first problem. Digilent provide windows software and pre-programed firmware for the Cypress chip to allow programming of the FPGA without a Xilinx JTAG programing cable. A Xilinx USB programmer is currently twice the cost of the Nexys 2 board. To facilitate this they have connected part of the upper byte of the FPGA → USB interface to the JTAG lines and programed the EPROM which the Cypress chip boots from with firmware. Consequently to operate as required a number of resistors were removed to disconnect the Cypress chip from the JTAG lines and the EPROM i2c line was cut and jumpered to stop it booting from that code. A negative side effect of this is that the interface between the FPGA and the Cypress chip is now only byte instead of word wide. After adjusting the VHDL to accommodate the byte wide interface the main area of development is the interface flags. Three flags, A, B and C, are available and can be programmed in the 8051 firmware to report whether an endpoint is full, empty or at any level in between. A flow control mechanism can be implemented using these flags. A simple scheme is currently in use where data is written if the end point is not full, and the command endpoint is read when it is not empty. Data is discarded if it can not be written to USB. Further work is required in developing a better scheme with buffering if required, it is not yet understood how and when to best discard data. Jon Hosie from FDL's firmware team is carrying out the implementation of the complete RR017 HDL.

15.2.4.2. 8051 Firmware

Writing the firmware for the cypress part should have been far easier than it was. Cypress provides example code that contains all of the required structure and interrupt service routines. Beyond modifying a small amount amount of assembly code so that the device reports the Filtronic vendor ID and the correct product ID all that remained to be done was to write the TD_init start up routine. TD_Poll the function called during the main loop isn't actually required to do anything when the device is in slave FIFO mode. It was also decided that the 8051 should accommodate the FPGA rather than the reverse where possible, so the 8051 must understand the polarity of the control lines.

In TD_Init, each end point is effectively configured twice, once by assigning to a register called EPXCFG and once to EPXFIFOCFG. A simple mistake easily made is to update one neglecting the other. It appears that order is important when assigning to registers. The 8051 must assign memory somehow to the endpoint buffers but it is not clear quite how this happens. The end point used to carry the PDW data back to the host PC operates correctly if triple buffered, but transmits bad data if quadruple buffered with no adequate explanation. There appears to be a number of known hardware faults with the device which have been fixed in newer revisions. These workarounds can be activated through the REVCTL register. They do however reveal further problems. In order to get an OUT end point to operate, the 8051 has to "Prime the pump" to use Cypress's terminology. This means disabling AUTOOUT committing however many packets as the buffer is deep, then re-enabling AUTOOUT, which questions how the system comes out of reset. A functioning firmware was developed by copying extracts from the application notes and other documentation available at Cypress's website [37] and modification through trial and error. Without a greater understanding of the internal operation of the part, it is difficult to understand some of the suggestions.

15.2.4.3. Windows Driver development

Adrian Metcalfe, the leader of FDL's software team, has written a windows driver using the supplied Cypress toolkit and a debugging application which displays the information passing in and out of the endpoints whilst allowing the user to send commands to the FPGA. Another command line based application is able to measure the rate of data throughput.

15.2.4.4. Linux Driver development

Linux USB driver development is different to windows in that a driver in the normal sense is not required. The Linux kernel exposes the USB end points and they can be accessed by any application from user space. A kernel mode driver could be created, but is not required. An abstraction layer for accessing the end points is available called libusb [38]. This makes accessing the device quite straight forward, but does impact upon performance as described later. As a result of this a simple application was written, making use of libusb that reads data from the device and measures throughput.

15.2.5. Throughput Results

The RR017 has an internal signal called CW_Meas that indicates that a measurement has occurred. Using the Nexys board this line was routed to an external pin and driven by a signal generator. This signal is then used by the test mode component to control the output data rate. Using Linux and the libusb layer, throughput is capped at 2MBps which is unacceptably slow. To use this device with a Linux host will require a different way of reading from the device. The GNURadio [39] project has achieved data rates of 30MBps using a different method and the same USB chip so it is possible. Under Windows XP much faster results were obtained. The initial test of 20,000 loops of the quick read program yielded a throughput of 16MBps, still not the target of 20MBps but a lot closer. When the number of loops was increased to 50,000 problems started to occur with the program reporting failures to read data. The maximum stable throughput over 50,000 loops was 7.69 MBps. Quickread discards the data after collection from USB so little is yet known about how much data is being lost, which data is being lost or why. This is however indicative of a worrying problem as the data path from the FPGA side of the USB FIFO to the output of the Cypress USB driver is a black box.

15.2.6. Remaining Problems

- Data loss at high speed over multiple loops.
- Despite firmware settings, the 8051 is unwilling to automatically commit an in packet of less than 512 bytes. 96 bit PDWs do not line up with a 512Byte boundary so currently the FPGA is committing the packet
- Unable to quad buffer the PDW stream without data corruption. Cause unknown, this may not be an issue.

15.3. Version Control Systems and Defect Trackers

15.3.1. Introduction

As software systems have grown in size and complexity a wide variety of software development tools have been created to ease the development process. Version control systems and defect trackers are two types of software development tools, but these tools are also known by other names depending on their exact specification and the developer's intentions. FDL is currently using Microsoft Visual Source Safe as its version control system and does not currently make use of any form of defect tracker. Source Safe is now an obsolete product having been replaced by Visual Studio Team System and is consequently no longer being actively developed. As FDL is now trying to move into the defence sub-system market, a move which will require a larger software development capacity, it was deemed timely to review the tools available and asses their suitability for FDL's needs.

15.3.2. Requirements

15.3.2.1. Version Control system

- **Both useful and easy to use**. The system should help the engineer to be more productive, integrate well into their development process and make minimal demands on their time and blood pressure.
- **Allow collaboration**. The system should allow more than one engineer to work concurrently on the same file, be able to resolve trivial conflicts and assist in resolving non-trivial conflicts. Branching and merging should be straight forward operations
- **Available**. The tool software should be available to whoever needs it on any machine. Access to source code through the tool should be of the same order of difficulty as using a shared drive or memory stick.

15.3.2.2. Defect tracker

Implementing a good defect tracking system is a difficult task as the bug information is often collected and submitted by users who may have limited general technical knowledge and no understanding of the system that has failed. The user interface therefore has to be carefully crafted, with the anticipated users in mind. Although it is envisaged that the defect tracker will initially only be used by FDL software developers it is highly likely that in the future the system will be extended to other FDL staff and maybe even customers.

- **Easy to install, configure and use**. The chosen system should place no unnecessary burden on IT or engineering resources. As with the version control system, it should integrate well with the engineers development process
- **Integration with DOORS**. To allow requirements to be linked to defects, e.g. web interface.
- **Integration with version control system**. To link changes to defects and hence requirements.

15.3.3. Available Software

15.3.3.1. Version Control systems

Most systems operate using either the client server model or the distributed model, though there are some exceptions such as source safe which uses a thick client model. Distributed systems are popular in large Free Software projects where there are no ownership or IP issues and the development process is less managed. In such a system, there is no central repository, no master copy and each developer maintains their own source tree. The Linux kernel, created by Linus Torvalds, uses a distributed model and although all official releases are made from the source tree maintained by Linus other developers maintain their own trees, making it easier for them to work in parallel before submitting their changes to Linus. The client server model is more popular in commercial projects as there is one authoritative copy of the software upon which all engineers work, which is held in one central location. The distributed model has only recently become popular, driven by the needs of the Linux kernel team. I have chosen to focus on the client server model as the available software is more mature and it is more suited to commercial environments.

- **IBM/Rational Clearcase**. Clearcase is the largest and most comprehensive system and is popular amongst larger companies. It is expensive at £2999.00 + VAT per floating user and requires a lot of administration. Integrates with most other software development tools and has a companion defect tracker called Clearquest.

- **Perforce**. Popular and advertised as fast. Integrates into a number of supporting tools. $800(USD) for the first year of each of the first 20 users and $160 (USD) maintenance per user per subsequent year. The user licenses do not float, they are simply user accounts on the server.
- **Subversion**. Designed as a better CVS, fixing most of the major shortcomings of CVS. Designed in the UNIX style of concentrating on one task. Consequently other tools are needed to complete the development environment. Widely used in the Free Software community, used by both the GNOME project and sourceforge. License cost £0, it is Free Software and many Free Software tools exist to complement it. TortoiseSVN provides windows explorer Shell integration and there are at least 2 choices of Visual studio plug-in. Also well supported by other editors/IDEs such as eclipse and EMACS. It can use the Apache run-time for network communication which integrates with Microsoft Active Directory for authentication.

15.3.3.2. Defect Trackers

- **IBM Rational Clearquest**. Companion to Clearcase and is again large, comprehensive and equally expensive at £3870.45 per floating user.
- **Bugzilla**. Initially developed by the mozilla team to track defects in the mozilla project, bugzilla is now a standalone project. It is used by many Free Software projects and commercial organisations. It is Free Software with a £0 license cost and is used via a web interface. There are many public bugzilla installations, http://bugzilla.kernel.org http://bugzilla.redhat.com are two examples.
- **Trac**. Is a light weight defect tracker with integrated wiki. It is Free Software with £0 license cost and plugs into the Apache web server. It provides simple release planning, defect tracking and a wiki in one package. It does not offer the scale or capability of Bugzilla or Clearquest, but equally does not require the installation and configuration expertise.

15.3.4. Conclusion

Free Software software development tools have an advantage over propriety tools in that they can be deployed as soon as the need is identified. There are no budgeting or licensing issues, or procurement delays which removes another barrier to using them.

The obvious choice on that basis alone is a combination of Subversion and Trac as they are both free software with £0 license cost and work well together. Subversion will scale to hundreds of users, by which point Trac may no longer be adequate, but deploying a more comprehensive defect tracker at this early stage may be counter productive by being needlessly onerous to use.

16. Reflection

At the end of the EngD Program the skills and experience that I have gained can be divided into four categories: technical skills, business skills, people skills and industrial experience. These categories are not distinct from each other as a large part of the industrial experience is the application of the skills gained to real world problems in a commercial setting.

The technical skills were largely gained through remote and on site study at ISLI. The courses of embedded software, microprocessors and microcontrollers gave me a much broader understanding of the field. Although I had worked at a leading firm for two and a half years prior to the course my knowledge was very focused and without a sound theoretical foundation, as my background was in electronics rather than computer science. The silicon system level integration courses of System Partitioning, IP Block Authoring and IP Block Integration were directly relevant to the architecture of the Phobos embedded software. The software was designed with attention to interfaces and re-usability, whilst allowing the processing load to be spread across multiple cores.

The business skills I gained at the Edinburgh Business School were much more generally applicable than I expected. Most of the products that Teledyne develops are sold with governments as the end customer, especially the UK government. With my joining of the EngD program coinciding with the banking crisis, the course on economics was very useful in understanding the spending decisions that governments had to make. The course on negotiation was the most illuminating and it complemented the people skills courses taught at the annual ISLI summer course held at Harburn House. At Phobos demonstrations and talking with customers at trade shows these two courses helped me to realise that the reason for the question was often more important than the question itself. Understanding the difference between what the customer wanted to achieve and the means that they expected to achieve it with, is important when selling an unconventional product. The perceived shortcomings of the product had to be defended in a credible way when compared to a conventional product. Both points are key to a successful negotiation and product marketing.

The people skills can then be carried over into teamwork at the sponsor company. The team of people working on Phobos came from various parts of the company and when problems occurred, the training on negotiation, persuasion and high performance teamwork helped me to both understand the situation and to make progress.

The project was not without unexpected success or problems. The biggest successes with the system were the general stability and reliability of the system. We spent very little time chasing unreproducible bugs or battling unexpected system behaviours. The system load was also much lower than predicted. At the start of the project it was feared that the pulse identification software would be very processor intensive, but in fact there was a lot of headroom left and with the continued improvement of single board computers the situation can only further improve, raising the pulse density ceiling of the system. The two major unexpected problems were antenna calibration and the unreliability of the RR017 pulse characteriser. It was thought from the beginning that calibrating the antennas would be time consuming, but we never expected it to be impossible with the original casing. This was always a high priority task and it took a lot of time away from other tasks. Phobos was one of the first real world uses of the RR017 and our work found problems with the firmware that had not been found during production test. As these issues could be worked around to some extent, they were not fully investigated until after the EngD project had ended and the RE was writing up.

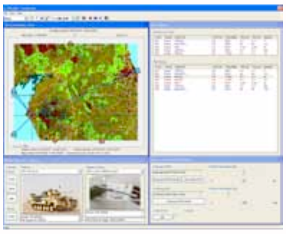
In retrospect it would have been better if two problems had been handled differently. The embedded software had a number of problems that were not quantified early enough in the project. We knew that there were problems, but as they could be worked around, or were not on the critical path, there was no great urgency to investigate them. There was always commercial pressure to demonstrate progress and we didn't always get the balance right between delivering functionality for the next demonstration and building a dependable system. Our ability to verify system operation was also limited, largely based around the high level outputs and the timing code was only written as an academic exercise rather than as product development objective. As the code grew in functionality and complexity, we didn't place enough importance on knowing the details of how the system was behaving. We should have stopped to quantify the risk of system problems as soon as they ceased to be high priority tasks. As this was not done we didn't know the extent of the risk we were taking. Quantifying them would have also made it much easier to make the case for the importance of resolving them. Telemetry should have been introduced much earlier in the project. Performance was always a key part of the product, but for most of the development period we only had a very coarse view of how our changes affected system performance. The embedded software is at its core a data processing application, but there was little ability to trace the flow of the data through the system. Data tracing should have been much more comprehensive and done much earlier.

A. PHOBOS-R Product Brochure

TELEDYNE
DEFENCE
A Teledyne Technologies Company

PHOBOS-R 2 to 18GHz Threat Warner/RESM System

Technical Datasheet QR020-M1



Features

- 2-18GHz Instantaneous Frequency Range
- Full 360° Azimuth Coverage
- Bearing Accuracy 10° rms
- Extremely small Size & Weight
- Very Low Power Consumption (<25Watt)
- Rapid Threat Warning (Emitter ID < 1sec)
- UK MoD Proven processing algorithms
- No External RF Cables
- Simple to Deploy & Operate
- Built in auto-positioning (Compass / GPS)
- Minimal Operator Workload
- Network enabled connectivity

Applications

- Unmanned Aerial Vehicles
- Man Portable / Deployable Sensors
- Remote Sensing (Unmanned)
- Armoured Fighting Vehicles
- Fast Patrol Boats – Littoral / EEZ Operations
- Reconnaissance Vehicles
- Light Utility Aircraft (FW & RW)
- Low Cost Combined Sensor Suites

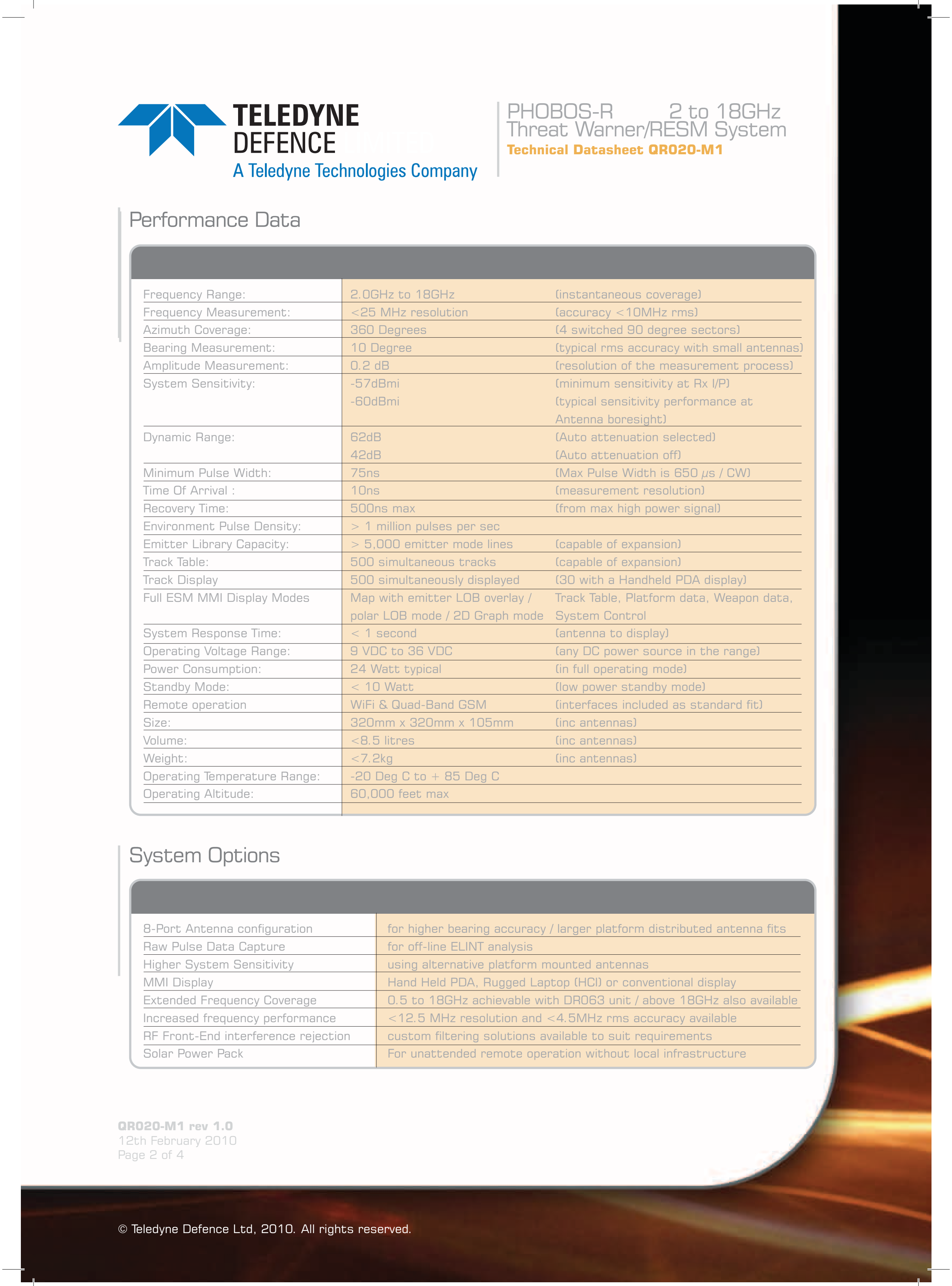
Product Description

The Phobos-R Threat Warner/RESM is an extremely compact, affordable, end-to-end integrated EW sensor system comprising: Antennas, RF Processing, Digital Processing, De-interleaving & Emitter ID/Library Matching, and Operator Interface. The system design employs a high degree of RF & digital signal processing integration, (based on established Teledyne RR017 and QR020 PHOBOS products), enabling the full 2-18GHz frequency coverage and 360° azimuth coverage to be achieved in a very small and light-weight unit. A key feature of the Phobos-R is that there are no external RF cables and no positional alignment requirements during set-up, making it extremely easy to deploy and operate on a wide variety of small platforms of all types, including those not thought previously feasible for such protection on the grounds of size, weight, power or cost. Only two external cables are required for system operation; DC power (9v–36v) and a network cable for data output/system control. The system includes both WiFi & quad band GSM interfaces, enabling either local or remote location of the sensor unit relative to the user interface. The sensor system incorporates established waveform based processing algorithms proven by UK MoD, also enabling the creation of a user interface which is both robust in dense signal environments and requires minimal operator workload or training. Simple to use hand-held ruggedised PDA MMI display & full ESM MMI running on a ruggedised laptop or conventional displays are also available. MIL-STD 2525 symbology ensures ease of object recognition. The rapid threat warning response (<1 sec) also facilitates use of the system in conjunction with self-protection measures such as automatically set-on responsive jammers (ECM / EA) and it can also be used in conjunction with CESM as a frequency extension for low cost combined EW sensor suites.

Teledyne Defence Ltd
Airedale House
Royal London Industrial Estate
Acorn Park, Charlestown
Shipley, West Yorkshire
UK, BD17 7SW

Tel: +44 (0) 1274 531 602
Fax: +44 (0) 1274 595 724
Email: tdl-phobos@teledyne.com
Web: www.teledynedefence.co.uk

© Teledyne Defence Ltd, 2010. All rights reserved.



TELEDYNE DEFENCE
A Teledyne Technologies Company

PHOBOS-R 2 to 18GHz
Threat Warner/RESM System
Technical Datasheet QR020-M1

Performance Data

Frequency Range:	2.0GHz to 18GHz	(instantaneous coverage)
Frequency Measurement:	<25 MHz resolution	(accuracy <10MHz rms)
Azimuth Coverage:	360 Degrees	(4 switched 90 degree sectors)
Bearing Measurement:	10 Degree	(typical rms accuracy with small antennas)
Amplitude Measurement:	0.2 dB	(resolution of the measurement process)
System Sensitivity:	-57dBmi -60dBmi	(minimum sensitivity at Rx I/P) (typical sensitivity performance at Antenna boresight)
Dynamic Range:	62dB 42dB	(Auto attenuation selected) (Auto attenuation off)
Minimum Pulse Width:	75ns	(Max Pulse Width is 650 µs / CW)
Time Of Arrival :	10ns	(measurement resolution)
Recovery Time:	500ns max	(from max high power signal)
Environment Pulse Density:	> 1 million pulses per sec	
Emitter Library Capacity:	> 5,000 emitter mode lines	(capable of expansion)
Track Table:	500 simultaneous tracks	(capable of expansion)
Track Display	500 simultaneously displayed	(30 with a Handheld PDA display)
Full ESM MMI Display Modes	Map with emitter LOB overlay / polar LOB mode / 2D Graph mode	Track Table, Platform data, Weapon data, System Control
System Response Time:	< 1 second	(antenna to display)
Operating Voltage Range:	9 VDC to 36 VDC	(any DC power source in the range)
Power Consumption:	24 Watt typical	(in full operating mode)
Standby Mode:	< 10 Watt	(low power standby mode)
Remote operation	WiFi & Quad-Band GSM	(interfaces included as standard fit)
Size:	320mm x 320mm x 105mm	(inc antennas)
Volume:	<8.5 litres	(inc antennas)
Weight:	<7.2kg	(inc antennas)
Operating Temperature Range:	-20 Deg C to + 85 Deg C	
Operating Altitude:	60,000 feet max	

System Options

8-Port Antenna configuration	for higher bearing accuracy / larger platform distributed antenna fits
Raw Pulse Data Capture	for off-line ELINT analysis
Higher System Sensitivity	using alternative platform mounted antennas
MMI Display	Hand Held PDA, Rugged Laptop (HCI) or conventional display
Extended Frequency Coverage	0.5 to 18GHz achievable with DR063 unit / above 18GHz also available
Increased frequency performance	<12.5 MHz resolution and <4.5MHz rms accuracy available
RF Front-End interference rejection	custom filtering solutions available to suit requirements
Solar Power Pack	For unattended remote operation without local infrastructure

QR020-M1 rev 1.0
12th February 2010
Page 2 of 4

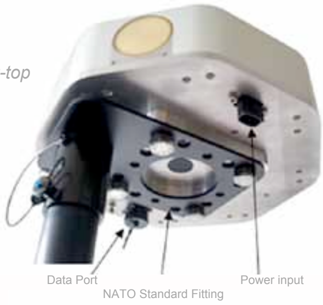
© Teledyne Defence Ltd, 2010. All rights reserved.

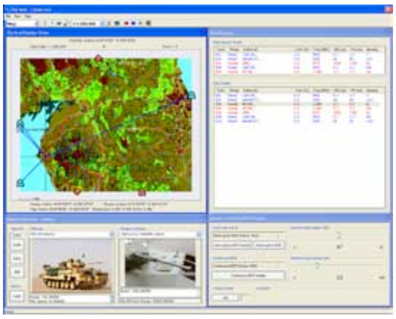
TELEDYNE
DEFENCE
A Teledyne Technologies Company

PHOBOS-R 2 to 18GHz
Threat Warner/RESM System

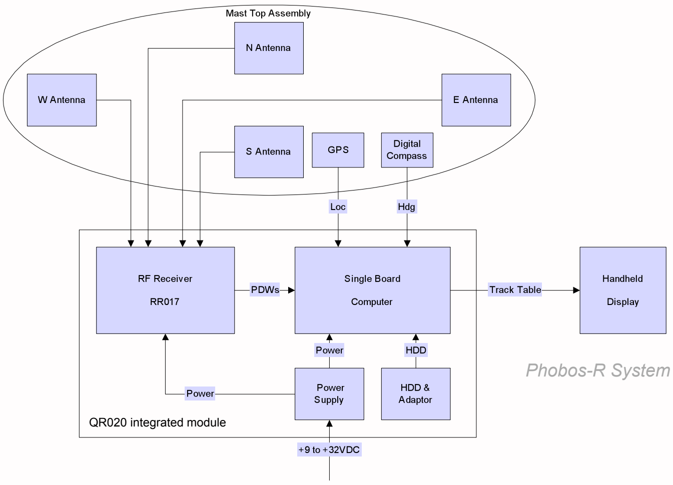
Technical Datasheet QR020-M1

Phobos-R integrated Sensor mounted on mast-top





Colour MMI Display on rugged lap-top



Phobos-R System functional block diagram

QR020-M1 rev 1.0
12th February 2010
Page 3 of 4

© Teledyne Defence Ltd, 2010. All rights reserved.

TELEDYNE
DEFENCE LIMITED
A Teledyne Technologies Company

PHOBOS-R 2 to 18GHz
Threat Warner/RESM System
Technical Datasheet QR020-M1

Teledyne Defence Ltd
Airedale House
Royal London Industrial Estate
Acorn Park
Charlestown
Shipley, West Yorkshire
UK, BD17 7SW

Tel: +44 (0) 1274 531 602
Fax: +44 (0) 1274 595 724
Email: tdl-phobos@teledyne.com
www.teledynedefence.co.uk

© Teledyne Defence Ltd, 2010. All rights reserved.

Bibliography

[1] Brown S, Irvine J, Metcalfe A. Antenna Characterisation for Amplitude Comparison in Electronic Warfare Systems; 2010. Available from: http://www.armms.org/media/uploads/1335954896.pdf.

[2] Teledyne Defence Ltd. Technical Datasheet RR017. Airedale House, Shipley, West Yorkshire, UK, BD17 7SW; 2011. Rev 3.0. Available from: http://www.teledynedefence.co.uk/pdf/RR017_Teledyne.pdf.

[3] Hassan HE. A new algorithm for radar emitter recognition. In: Proc. 3rd Int. Symp. Image and Signal Processing and Analysis ISPA 2003. vol. 2; 2003. p. 1097–1101.

[4] Wiley RG. ELINT The Interception and Analysis or Radar Signals. Artech House; 2006.

[5] Rogers JAV. ESM processor system for high pulse density radar environments. IEE Proceedings F Communications, Radar and Signal Processing. 1985;132(7):621–625.

[6] Mardia HK. New techniques for the deinterleaving of repetitive sequences. IEE Proceedings F Radar and Signal Processing. 1989;136(4):149–154.

[7] Milojevic DJ, Popovic BM. Improved algorithm for the deinterleaving of radar pulses. IEE Proceedings F Radar and Signal Processing. 1992;139(1):98–104.

[8] Rockwell Collins. CS-3600 Tactical Surveillance System;. Available from: http://www.rockwellcollins.com/~/media/Files/Unsecure/Products/Product%20Brochures/EW-Intelligence/SIGINT/CS-3600/CS-3600%20data%20sheet.aspx.

[9] Thales. MEERKAT-S: High Mobility Radar Band ESM and ELINT System;. Available from: http://www.thalesgroup.com/Portfolio/Defence/Aerospace_Product_Meerkat-S/.

[10] G Coley, C Cooley and J Kridner. Beagle Board and Beagle Bone Community. 1380 Presidential Dr, Suite 100, Richardson, TX 75081-2437;. Available from: http://beagleboard.org/.

[11] Compulab Ltd. FitPC 2 Datasheet. 17 HaYetsira Street, Moradot HaCarmel Industrial Park, Yokneam Elite, Israel 20692;. Available from: http://www.compulab.co.il/fitpc2/html/fitpc2-sb-datasheet.htm.

[12] Nilsson J. Implementing a Continuously Updating, High-resolution Time Provider for Windows. MSDN Magazine. 2004;p. 78–88. Available from: http://msdn.microsoft.com/en-us/magazine/cc163996.aspx.

[13] Barton DK. Radar Systems Analysis. Artech House; 1976.

[14] Selex Galileo. Sky Guardian 2500 Compact and Lightweight ESM. 300 Capability Green, Luton, Bedfordshire, LU1 3PG, United Kingdom; 2010. Available from: http://www.selex-es.com/documents/737448/4748929/body_mm07739_SEER_LQ_.pdf.

[15] Hartzog MBK, Brown TX. Wimax - Potential Commercial Off-The-Shelf Solution for Tactical Mobile Mesh Communications. In: Proc. IEEE Military Communications Conf. MILCOM 2006; 2006. p. 1–7.

[16] Stimson GW. Introduction to airborne radar. SciTech Pub.; 1998.

[17] Charette RN. Weapons Acquisition Problems Span the Globe; 2008. Available from: http://www.spectrum.ieee.org/nov08/6935.

[18] Ministry of Defence. MOD Architecture Framework;. Available from: https://www.gov.uk/mod-architecture-framework.

[19] iVeia LLC. 51 Franklin St, Suite 301, Annapolis, MD 21401;. Available from: http://www.iveia.com/.

[20] Annapolis Micro Systems. 190 Admiral Cochrane Drive, Suite 130, Annapolis, Maryland 21401 USA;. Available from: http://www.annapmicro.com/.

[21] Thales Group. Annual report 2007. 45 rue de Villiers, 92200 Neuilly-sur-Seine, France; 2007. SMART (SubMarine Advanced RESM Technology) p94. Available from: https://www.thalesgroup.com/sites/default/files/asset/document/thales2007_GB_V4C.pdf.

[22] Ministry of Defence. Joint Service Publication 777 - Network Enabled Capability;. Available from: http://webarchive.nationalarchives.gov.uk/20121026065214/http://www.mod.uk/NR/rdonlyres/E1403E7F-96FA-4550-AE14-4C7FF610FE3E/0/nec_jsp777.pdf.

[23] Liu JJN, Chlamtac I. In: Mobile Ad Hoc Networking with a View of 4G Wireless: Imperatives and Challenges. John Wiley & Sons, Inc.; 2005. p. 1–45.

[24] Burbank JL, Chimento PF, Haberman BK, Kasch WT. Key Challenges of Military Tactical Networking and the Elusive Promise of MANET Technology. Communications Magazine, IEEE. 2006 november;44(11):39–45.

[25] Dohler, Aghvami. Distributed and Cooperative Communication Networks with application to cellular, ad hoc and sensor networks; 2007. IEEE ICC 2007 Tutorial.

[26] IEEE 802 16 Working Group;. Available from: `http://www.wirelessman.org`.

[27] IEEE 802 16j Relay task group;. Available from: `http://www.wirelessman.org/relay/`.

[28] Sherman M, McNeill K, Shyy DJ, Spoenlein S. Contribution on 802.16j (Mobile Multihop Relay) Military Usage Models;. C802.16j-06/042. Available from: `http://www.ieee802.org/16/relay/contrib/C80216j-06_042.pdf`.

[29] Xohm WiMAX from Sprint;. Available from: `http://www.xohm.com`.

[30] Merritt. WiMax backers gear up for 700 MHz nets;. Available from: `http://www.eetimes.com/news/latest/showarticle.jhtml?articleID=201311262`.

[31] Sequans Communications. SQN1130 SoC for WiMAX Mobile Stations;. Available from: `http://www.sequans.com/site/sqn1130.html`.

[32] Pico Chip website;. Available from: `http://www.picochip.com`.

[33] Opal Kelly. XEM3001 Xilinx Spartan-3 Integration Module;. Available from: `http://www.opalkelly.com/products/xem3001`.

[34] Orange Tree Technologies. Zest SC1;. Available from: `http://www.orangetreetech.com/pdf/ZestSC1Flyer.pdf`.

[35] Digilent Inc. Digilent Nexys 2 Board Reference Manual;. Available from: `http://www.digilentinc.com/Data/Products/NEXYS2/Nexys2_rm.pdf`.

[36] Xilinx. Xilinx Web Pack;. Available from: `http://www.xilinx.com/ise/logic_design_prod/webpack.htm`.

[37] Cypress Semiconductor;. Available from: `http://www.cypress.com`.

[38] Drake D, Stuge P, et al.. libusb;. Available from: `http://libusb.org`.

[39] Blossom E, et al.. GNU Radio;. Available from: `http://www.gnuradio.org`.