Title: QUAV Project,

*“A Project”*

Prepared by Date \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

,

Checked by Date \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

,

Approved by Date \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

, Student Manager 2010

Authorised for use by Date \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Dr Luis Mejias, Project Coordinator

**QUT Avionics**

Queensland University of Technology

CRCSS-EESE, GPO Box 2434

Gardens Point Campus

Brisbane, Australia, 4001.

Telephone (+61 7) 3864 1772

Facsimile (+61 7) 3864 1517

e-mail luis.mejias@qut.edu.au

web <http://code.google.com/p/ahns10/>

This document is Copyright 2010 by the QUT. The content of this document, except that information which is in the public domain, is the proprietary property of the QUT and shall not be disclosed or reproduced in part or in whole other than for the purpose for which it has been prepared without the express permission of the QUT

**Revision Record**

|  |  |  |  |
| --- | --- | --- | --- |
| Document Issue/Revision Status | **Description of Change** | **Date** | **Approved** |
|  | Initial Issue |  |  |

**Distribution List**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Affiliation** | **Distribution Date** | **Approved** |
| Avionics Lab File Archive | QUT Avionics |  |  |

**Foreword**

Foreword.

**Table of Contents**

Paragraph Page No.

[1 Introduction 8](#_Toc256110267)

[1.1 Scope 8](#_Toc256110268)

[1.2 Background 8](#_Toc256110269)

[2 Reference Documents 9](#_Toc256110270)

[2.1 QUT Avionics Documents 9](#_Toc256110271)

[2.2 Non-QUT Documents 9](#_Toc256110272)

[3 Project Outline 11](#_Toc256110273)

[3.1 Project Aim 11](#_Toc256110274)

[3.2 Project Design Methodology 11](#_Toc256110275)

[4 System Architecture 12](#_Toc256110276)

[4.1 Hardware Architecture 12](#_Toc256110277)

[4.2 Software Architecture 14](#_Toc256110278)

[5 System Design 15](#_Toc256110279)

[5.1 Platform/Airborne Systems 15](#_Toc256110280)

[5.1.1 Airframe 15](#_Toc256110281)

[5.1.2 Enclosure 15](#_Toc256110282)

[5.1.3 Power 16](#_Toc256110283)

[5.1.4 Interfaces 17](#_Toc256110284)

[5.1.5 Flight Computer 19](#_Toc256110285)

[5.1.6 Airborne Software 19](#_Toc256110286)

[5.1.7 Flight Control 20](#_Toc256110287)

[5.1.8 State Estimation 21](#_Toc256110288)

[5.1.9 Localisation 21](#_Toc256110289)

[5.2 Ground Control Systems 23](#_Toc256110290)

[5.2.1 Ground Control Software 23](#_Toc256110291)

[5.3 Communications 23](#_Toc256110292)

[6 Conclusions 24](#_Toc256110293)

[7 Recommendations 25](#_Toc256110294)

[8 Appendices 26](#_Toc256110295)

**List of Figures**

Figure Page No.

[Figure 4.1 - Mode Control Unit 12](#_Toc256110296)

[Figure 4.2 - Hardware Architecture 13](#_Toc256110297)

[Figure 4.3 - High Level Block Diagram of Proposed Software Architecture 14](#_Toc256110298)

[Figure 5.1 - 3D Render of Final System 15](#_Toc256110299)

[Figure 5.2 - Component Mounting Inside Enclosure 16](#_Toc256110300)

[Figure 5.3 - Power System 17](#_Toc256110301)

[Figure 5.4 - Vibration Dampeners (15mm and 8mm) [RD/4] 18](#_Toc256110302)

[Figure 5.5 - Airframe Interface Connector [RD/5] 19](#_Toc256110303)

[Figure 5.6 - Gumstix Overo Fire COM [RD/11] 19](#_Toc256110304)

[Figure 5.7 - Quadrotor motor control input effects [RD/13] 21](#_Toc256110305)

[Figure 5.8 - Surveyor SRV-1 Blackfin Camera [RD/10] 22](#_Toc256110306)

**List of Tables**

Table Page No.

[Table 5.1 - Platform states and sensor measurement source 21](#_Toc256121996)

**Definitions**

|  |  |
| --- | --- |
| AHNS | Autonomous Helicopter Navigation System |
| IMU  RC  MCU  ESC  EMI  PCB  AIC | Inertial Measurement Unit  Remote Control  Mode Control Unit  Electronic Speed Controller  Electro-Magnetic Interference  Printed Circuit Board  Aircraft Interface Connector |
| MAV | Micro Air Vehicle |
| GPS | Global Positioning System |
| GCS | Ground Control Station |
| HMI | Human Machine Interface |
| 3D | Three dimensional |
| IR | Infrared |

# Introduction

This document outlines the proposed system architecture, hardware solutions and design decisions that were chosen to successfully meet the system requirements of the AHNS project as mentioned in [RD/1]. The motivation for these choices revolves around the project aim which is to develop an autonomous small size helicopter which can be used indoors. To achieve this aim, a system architecture was developed which addresses both hardware and software interfaces of the project. The implementation of this architecture is explained in subsequent system design sections which are split into platform (airborne) and ground control system components. Each design section details what hardware has been chosen and any relevant design decisions that have needed to be made. It is anticipated that many of these preliminary design choices and decisions will need to be modified to met the system requirements and ultimately render the AHNS project a success.

## Scope

The preliminary design document represents the initial research conducted by the members of the AHNS project. The outcomes of this research forms the basis of the system architecture and hardware solutions presented in this document. The scope of the project will be limited to these design decisions until the need for them to change arises.

## Background

The AHNS project is a continuation of the 4th year Avionics AHNS projects which started in 2007. Much of the design methodology presented here has relied upon the work of AHNS 2009 which managed to successfully achieve altitude hold on a small size electric helicopter. It is hoped that this year’s project can extent on last year’s success and achieve full autonomy on the helicopter platform.

# Reference Documents

## QUT Avionics Documents

|  |  |  |
| --- | --- | --- |
| RD/1 | AHNS-2010-SY-HL-001 | AHNS 2010 High Level Objectives |
| RD/2 | AHNS-2010-SY-SR-001 | AHNS 2010 System Requirements |
| RD/3 | AHNS-AP-DD-003 | QUAV Project, 2009, Autonomous Helicopter Navigation System, Autopilot Software, Design Document for |
| RD/14 | AHNS-2010-PL-TS-001 |  |
| RD/15 | AHNS-SE-DD-001 | QUAV Project, Autonomous Helicopter Navigation System, State Estimator, Attitude Estimation Detailed Design for |

## Non-QUT Documents

|  |  |  |
| --- | --- | --- |
| RD/4 | HiSystems Vibration Dampeners | HiSystems. 2010. Vibration Dampeners. Available: [https://www.mikrocontroller.com/images/Vibrationsdaempfer\_8mm.jpg](https://www.mikrocontroller.com/images/Vibrationsdaempfer_8mm.jpg%20) (accessed March 10 2010). |
| RD/5 | Jaycar 26 Pin IDC Right Angle Header | Jaycar Electronics. 2010. 26 Pin IDC Locking Right Angle Header, Available: [http://jaycar.com.au/products\_uploaded/productLarge\_372.jpg](http://jaycar.com.au/products_uploaded/productLarge_372.jpg%20) (accessed March 10 2010). |
| RD/6 | Embedded Linux Primer: A Practical Real-World Approach | Hallinan, C. 2007. Embedded Linux Primer: A Practical Real-World Approach: Prentice Hall PTR Upper Saddle River, NJ, USA. |
| RD/7 | OpenEmbedded | OpenEmbedded. 2009. Welcome to OpenEmbedded. Available: <http://wiki.openembedded.net/index.php/Main_Page> (accessed March 10 2010). |
| RD/8 | GNU Pth - The GNU Portable Threads | Engelschall, R. S. 2010. GNU Pth - The GNU Portable Threads. Available: <http://www.gnu.org/software/pth/> (accessed March 10 2010). |
| RD/9 | Native POSIX Threads Library (NPTL) Support for uClibc | Hill, S. J. 2006. "Native POSIX Threads Library (NPTL) Support for uClibc." Available: <http://www.kernel.org/doc/ols/2006/ols2006v1-pages-409-420.pdf> (accessed March 10 2010). |
| RD/10 | Surveyor SRV-1 Blackfin Camera | Surveyor Corporation. 2010. Surveyor SRV-1 Blackfin Camera. Available: <http://www.surveyor.com/blackfin/> (accessed March 9 2010). |
| RD/11 | Overo Fire COM | Gumstix 2010. Overo Fire COM. Available: <http://www.gumstix.com/store/catalog/product_info.php?products_id=227>  (accessed March 7 2010) |
|  |  |  |
| RD/12 | Design and Control of an Indoor Micro Quadrotor | Bouabdallah, S, P Murrieri and R Siegwart. 2007. "Design and Control of an Indoor Micro Quadrotor." In Advances in unmanned aerial vehicles: state of the art and the road to autonomy, ed. K. P. Valavanis: Springer Verlag. |
| RD/13 | IEEE Design and control of an indoor micro quadrotor | Bouabdallah, S., P. Murrieri and R. Siegwart. 2004. "Design and control of an indoor micro quadrotor." In IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION: IEEE. |
|  |  |  |

In the event of any conflict between this document and any RD referenced herein, such conflict shall be notified to .

In the following text, RD/x identifies referenced documents, where "x" denotes the actual document.

# Project Outline

## Project Aim

The aim of the AHNS project is to develop a hardware and software solution to automate a small size electric helicopter for indoor environments. Traditionally, the automation of a Micro Air Vehicle (MAV) such as a small size electric helicopter has relied on GPS as its primary tool for navigation. However due to the environmental restrictions on the project, other navigational means are required since GPS cannot be used indoors. The proposed navigational solution to achieve automation is a combinational approach consisting of inertial measurements and computer vision. Utilising these two concepts will allow the states of the platform to be estimated and ultimately controlled through the use of an onboard flight computer.

## Project Design Methodology

This document will outline the preliminary design decisions that have been made for the AHNS project. The inspiration for the majority of these decisions has come from the previous work of AHNS 2009 and their numerous recommendations for future work on the AHNS project. Other design decisions have stemmed from trade studies, baseline system requirements and conversations with professional and experienced researchers in the field of automation.

Critical design decisions that impact heavily on the system architecture have needed to be made in the preliminary design phase of the project. These decisions deviate extensively to the work conducted by AHNS 2009 and include:

* Analysis of alternative airframe configurations and types
* Placement of the flight control system to onboard the platform with implemented thread control
* Unification of the Ground Control Station (GCS) and the Human Machine Interface (HMI) software
* Use of an IEEE 802.11 network connection to achieve wireless communication between the platform (airborne) and the GCS
* Movement of the camera and image processing to onboard the platform
* Redesign of the platform payload to accommodate new hardware and address vibration concerns

# System Architecture

The AHNS project architecture can be divided into 2 critical sections: hardware and software architecture.

## Hardware Architecture

The hardware architecture can be seen in Figure 4.2 below. The basis of the system is the Gumstix Overo Fire (flight computer) which processes all the control code and has a wireless network adaptor built in for communications. This is connected to a breakout board to enable easy interface with the other hardware systems. The systems connected directly to the break out board are the IMU, camera, range finder, temperature sensor, coulomb counter, external lights relay, and magnetic compass.

In addition to this, there is a mode control unit that interfaces the RC receiver and electronic speed controllers with the flight computer. The job of the MCU is to read pulse widths from the RC receiver for use in stability augmentation mode as well as multiplexing the signals from the RC receiver directly to the ESCs when the aircraft is being flown in manual mode. Details of this can be found in Figure 4.1.



Figure . - Mode Control Unit



Figure . - Hardware Architecture

## Software Architecture

The software architecture can be seen in Figure 4.3 below. It consists of threaded airborne and ground based segments. The airborne segment is almost exclusively to be run on the flight computer whilst all ground control station code is to run on the GCS computer.



Figure . - High Level Block Diagram of Proposed Software Architecture

# System Design

## Platform/Airborne Systems

### Airframe

A suitable airframe was selected in trade study RD/14, which recommended that the Mikrokoper MK40 airframe be acquired for the project. This airframe was selected due to its low mechanical complexity, and relaxed maintenance regime. Figure 5.1 shows a 3D render of this airframe will all necessary components installed between the landing. This layout will maximise protection in the event of a crash. The landing gear will protect all hardware components when the aircraft in is its normal flying orientation and the motors will protect the hardware in the event of an inverted crash. In Figure 5.1 the four red blocks near the centre represent the ESCs, the two large blue blocks represent the two batteries, and the large black block represents the avionics enclosure.

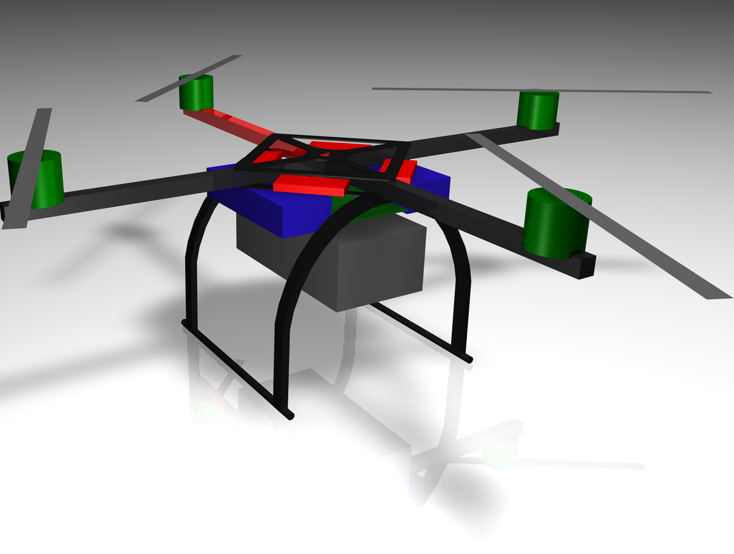


Figure . - 3D Render of Final System

### Enclosure

In order to provide protection against shock damage and to increase immunity against electromagnetic interference, all avionics components will be mounted inside an enclosure. The enclosure will be attached to the airframe using vibration dampeners. All the components will be fixed to a stack of custom made printed circuit boards and the stack will be mounted inside the enclosure using vibration dampeners. To ensure that all components would fit into the enclosure, they were all modelled to scale within a 3D package and assembled together in a frame representing the enclosure dimensions. Figure 5.2 shows the 3D render of one possible layout for the inside of the enclosure. With the layout depicted, the main stack is mounted on small vibration dampeners to reduce vibration in the camera, where as the IMU is suspended on larger vibration dampeners to ensure vibration free signals. Protruding out of the bottom of the enclosure are the camera and the range finder which both face down. A rectangular hole will be cut in the top of the enclosure to allow the RC receiver to protrude out and get a better signal. The chosen regime for reducing EMI will be lining the inside of the enclosure with aluminium tape that will be grounded. The wireless network antenna will also be mounted external to increase signal strength. The enclosure will interface with the airframe via a single Airframe Interface Connector. Refer to the interface management section for more details on the AIC.

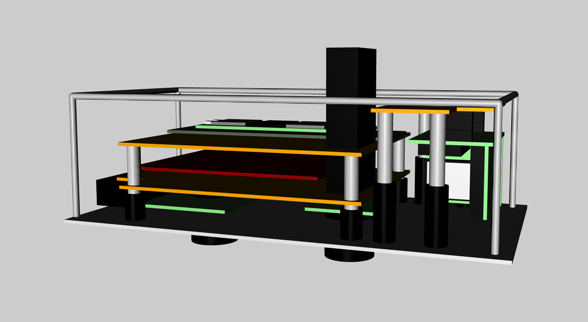


Figure . - Component Mounting Inside Enclosure

### Power

The power system on the AHNS project will consist of a dual power bus system as depicted in Figure 5.3. All electrical loads will be split between two independent power buses that are diode isolated from each other. Each bus provides power to two of the four motors as well as all the avionics and RC components. If either power bus fails, the functioning power bus will continue to provide power to all avionics as well as opposite motors. This would render the aircraft uncontrollable in the roll and pitch axes but would be sufficient to facilitate a controlled landing. The autopilot and MCU each have their own set of connections to the bus so that if the autopilot fails the MCU will be able to power the RC receiver facilitating a manual landing.



Figure . - Power System

### Interfaces

All hardware interfaces that exist on the platform can be separated into two basic categories, these are mechanical and electrical. This section of the document details the specific type and management of these interfaces.

#### Mechanical interfaces

Mechanical methods of reducing vibration will be incorporated into the design. Accurate IMU data is important in maintaining stability of the system and vibrations caused by the motors can degrade the accuracy to a point where the data can be rendered unusable. In addition to this it is important to ensure that vibrations are minimised on the camera to enable accurate state estimation through computer vision.

The scope of vibration dampening will be broken up into two categories, these are global and local. Global vibration dampening will entail the use of vibrations dampeners to mount the enclosure to the airframe. As all the avionics will be mounted inside the enclosure, the exposure of all devices to vibration will be reduced. Local vibration dampening entails the mounting of specific parts within the enclosure to result in a further decrease in vibration. The camera will be mounted on 8mm local vibration dampeners and the IMU will be suspended on more effective 15mm vibration dampeners. Figure 5.4 shows the 8mm and 15mm vibration dampeners that will be acquired with the airframe.

The enclosure will be mounted up-side down in the airframe with all the components mounted to the lid. As was seen in Figure 5.2, the avionics are attached to the lid. This means that by only removing the lid, the entire avionics suite is removed from the airframe and all components are easily accessible. This modular approach is in line with good systems engineering principles.

Standard M3 screws and PCB spacers will be used for all other mounting.



Figure . - Vibration Dampeners (15mm and 8mm) [RD/4]

#### Electrical Interfaces

In order to minimise the risk of vibration induced wire failure, measures will be taken to minimise the wire count on the project. This will be achieved by mounting all components onto PCBs so that there are no wires between the components. This will significantly reduce the wire count and hence the risk of wire failure. To simplify the process of removing the avionics from the airframe, a single airframe interface connector will be used. The AIC receptacle has been designed to only accept the plug in the correct orientation and will lock the plug once it is inserted. Figure 5.5 shows the airframe interface connector.

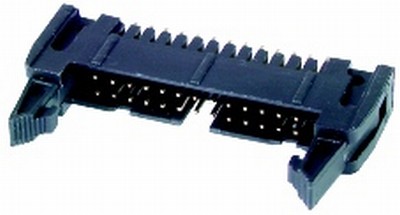


Figure . - Airframe Interface Connector [RD/5]

### Flight Computer

To enable autonomous operation of the AHNS platform considerable airborne computing power and hardware interfacing is required. This was decided to be achieved with a single-board flight computer. A trade study was therefore undertaken to consider the precise system requirements and high level objects of the system. The Gumstix Overo Fire COM was selected from a field of four single-board computer alternatives based on its processor type and power, mass, operating system support, memory, hardware interfaces and cost. Its processor support for digital image processing and computations proved its suitability. To expand its hardware interfacing options two expansion boards, the Pinto-TH and the Summit were also selected.



Figure . - Gumstix Overo Fire COM [RD/11]

### Airborne Software

The flight computer is an embedded device which is required to interface with the hardware sensors, actuators and communications. The requirement for considerable hardware interfacing has lead to the decision to program all embedded code with the C programming language. The overhead of object-oriented code is unwarranted thus an object-oriented programming language is not required. The importance of C is also clear when considering that the Linux kernel may have to be rebuilt for specific hardware configurations [RD/7].

The airborne software operating system will be developed on the flight computer with the aid of the OpenEmbedded Linux build environment [RD/7]. There is extensive support available for its use with the Overo Fire since it is recommended by the manufacturer. OpenEmbedded provides many pre-compiled programs and utilities. Aside from the Linux operating system, the airborne autopilot software has been formulated as a single process but its tasks have been divided into three logical threads; state estimation, control and downlink. The roles of each thread are shown in . It is anticipated that the state estimation and control threads will have the highest priority, approaching the requirement for soft real-time scheduling. The downlink thread will be scheduled at a considerably lower priority.

Implementation of the threads has been chosen to conform to the POSIX standard to guarantee functionality under the Linux operating system. The thread library desired and most widely distributed is the Native POSIX Threads Library (NPTL) [RD/6]. It is preferable because of its kernel creation of the threads, thus enabling pre-emptive scheduling which will switch between threads without waiting for events [RD/8]. The thread library to be used will need to be finalised based on the C library chosen as the base of the distribution.

The choice of C –libraries is essentially between GNU C (glibc) and micro C (uclibc) [RD/7]. From an embedded system perspective uclibc is the standard and hence preferable and does offer support for the desired POSIX NTPL thread library [RD/7;RD/9]. The C library is chosen at the time of operating system deployment from OpenEmbedded. The feasibility of using uclibc, NTPL and OpenEmbedded will become apparent early in the process of airborne code development nevertheless a threaded airborne software architecture is proposed.

### Flight Control

The autopilot control has been described in SR-B-10 as taking the form of a cascaded proportional, integral and derivative controller. That is, the autopilot will consist of cascaded guidance and control PID control loops. At a preliminary level these cascaded elements may be position control and attitude control [RD/12]. In stability augmentation mode the attitude controller is to maintain the quadrotor at a level attitude, without considering the altitude of the aircraft or its position. This mode is primarily to be used as a means of tuning the autopilot’s attitude control loops. In autonomous station keeping the position control will use the information from the localisation subsystem to drive the attitude control loops.

The quadrotor arrangement means regardless of the control implementation the outputs will be reference signals for the four engines as shown in Figure 5.7. Vertical motion can be controlled by simultaneously increasing or decreasing the motor speeds. To vary pitch or roll, asymmetric engine thrust is produced between engines 1 and 3 or 2 and 4. Pitch and roll variations also lead to longitudinal and lateral motion respectively [RD/12; RD13]. Yaw is controlled by exploiting the opposing torque of the two engine pairs, 1 and 3 and 2 and 4 [RD/12]. Thus decreasing the speed of one pair will cause the aircraft to yaw in the direction of their propeller rotations. Although these effects are well understood, gain tuning of the controller will still be achieved during on-line flight testing. Provisions have been included in the design of the software and hardware systems to log all state and control data for facilitation of linear system model development.

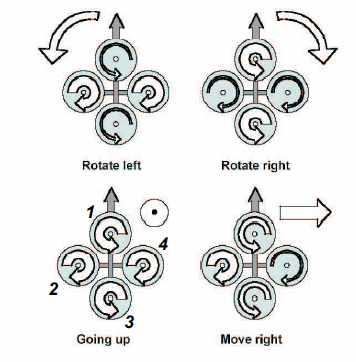


Figure . - Quadrotor motor control input effects [RD/13]

### State Estimation

The state estimation design will follow and extend on the state estimation implementation of AHNS 2009. 9 states of the platform will be measured by various sensors located on the platform itself (refer to Table 5.1).

Table . - Platform states and sensor measurement source

|  |  |  |  |
| --- | --- | --- | --- |
| **State** | **Sensor** | **State** | **Sensor** |
| Roll | IMU | Yaw rate | IMU |
| Pitch | IMU | X position | Onboard camera |
| Yaw | IMU | Y position | Onboard camera |
| Roll rate | IMU | Z position | IR sensor |
| Pitch rate | IMU |  |  |

The Euler angles and rates will all be measured by the Inertial Navigation Unit (IMU). The IMU which will be used will be the Sensor Dynamics Minicube 6DOF SD755. This hardware was inherited from AHNS 2009 and also came with a USB evaluation board for debugging purposes. The X and Y positions will be measured by an onboard camera which will be discussed in the following section. Finally the Z position will be measured by an Infrared (IR) sensor. The IR sensor is a Sharp GP2Y0A710K0F which has a measurement distance of 100-550cm. The IR sensor was also inherited by AHNS 2009.

A Kalman filter will be used to provide state estimation outputs for the flight control system. Filtering is required since the sensors measuring the states will contain inherit noise characteristics which introduce inaccuracies to the sensor measurements perceived. These noisy measurements form the input to the Kalman filter which will provide an estimate of what the true value of the measurement should’ve been. It was recommended from AHNS 2009 in RD/15 that a simple Kalman filter be implemented for this purpose.

### Localisation

Localisation of the platform will be through image processing as stated in HLO-2 [RD/1]. The image processing will be implemented through an onboard camera solution that will track a stationary ground target to determine its position estimate through X and Y offset co-ordinates.

#### Onboard camera

The onboard camera will be the Surveyor SRV-1 Blackfin Camera (refer to Figure 5.6). This hardware solution consists of a Blackfin BF537 processor which interfaces with an Omnivision OV9655 1.3 megapixel camera. The primary advantage of choosing this hardware solution is that it integrates a digital signal processing (DSP) Blackfin processor and a camera into the one hardware package. This frees the flight computer of any image processing tasks since the Blackfin processor will analyse and interpret the visual sensor data directly from the camera. The localisation data will then be transferred from the Blackfin processor to the flight computer when polled.



Figure . - Surveyor SRV-1 Blackfin Camera [RD/10]

#### Position estimate

A stationary ground target will be placed on the ground such that it is visible to the Surveyor SRV-1 Blackfin Camera hardware. The ground target will be a distinct shape (such as a cross or circle) which will have a high colour contrast when compared with the rest of the surrounding ground surface. A high colour contrast is important so that the image processing hardware can successfully identify the ground target against the ground surface scenery.

The hardware itself will be mounted to the airframe payload bay such that the camera lens will be pointing downwards. This will allow the camera and Blackfin processor to identify the stationary ground target and perform position estimates based upon the targets identified location. These estimates will be saved into the Blackfin processor memory as X and Y offset co-ordinates.

## Ground Control Systems

### Ground Control Software

A ground control station GUI and HMI GUI was designed in 2009 for AHNS using the Qt C++ library. In operation these two GUIs were run on separate computes linked by UDP [RD/3]. The lack of HMI system requirements removes the need to develop a separate HMI for a non-expert users and the move to a more powerful on board flight computer removes the need for control and state estimation to take place on the ground. Whilst it necessary to heavily modify the code-base to remove the UDP, control and state estimation features it is desired to continue development of the autopilot GUI developed.

The new proposed architecture still calls for a C++ Qt thread based architecture to run on a Ubuntu Linux desktop. The tasks preformed by the two GCS threads and their information sharing flow shows that very little processing other than logging, display and user input needs to be achieved through the GCS. Indeed the tasks for the GCS are well defined in SR-B-8 and SR-B-09.

## Communications

How is the communications going to work with the platform (Hamilton)

# Conclusions

# Recommendations

It is recommended that the hardware solutions and design decisions stipulated in this document be implemented and rigorously tested against the system requirements as mentioned in [RD/1]. If it is found that a component in the AHNS project fails the numerous acceptance tests required to meet the system requirements then that component should be appropriately altered or removed. This should then be followed by a possible redesign of the subsystem and subsequent retesting of the failed acceptance tests. This process should continue until all acceptance tests have been passed indicating that the system requirement in [RD/1] have been met. It is of critical importance that all design decisions that deviate from this document be fully documented over the lifetime of the project and be presented at the conclusion of the AHNS project.

# Appendices

No Appendices.