University of Southern Queensland

Faculty of Engineering & Surveying

Development of an Unmanned Aerial Vehicle for Use During Disaster Situations

A dissertation submitted by

Bede Wilson

in fulfilment of the requirements of

ENG4112 Research Project

towards the degree of

Bachelor of Engineering (Mechatronics)

Submitted: October, 2010

Abstract

The rise in natural disasters, combined with a decrease in the cost of miniaturised avionics has resulted in a significant interest in the use of Unmanned Aerial Vehicles (UAV's) within the disaster management field. This project seeks to serve as a first step towards developing a fully autonomous UAV targeted at this application. A comprehensive literature review was undertaken to assess the current state of the art in small UAV research and establish the context for this project. Legislation and standards related to the development and operation of small UAV's in Australia was also reviewed.

On completion of the review, an existing remote controlled aeroplane was purchased to be used as the parent vehicle and the control system hardware and software designed. The goal of this stage was to develop a UAV stabilisation system as a step towards fully autonomous flight. Some future work is required to rectify some problems that were encountered but significant progress was made. A fly-by-wire or co-pilot mode was also developed to facilitate easy manual control of the aircraft and lead in to the integration of a navigation module.

On completion of the final adjustments the UAV will undergo extensive flight trials to tune the control system ready for further research.

University of Southern Queensland

Faculty of Engineering and Surveying

ENG4111/2 *Research Project*

Limitations of Use

Prof F Bullen

Dean

Faculty of Engineering and Surveying

Certification of Dissertation

I certify that the ideas, designs and experimental work, results, analyses and conclusions set out in this dissertation are entirely my own effort, except where otherwise indicated and acknowledged.

I further certify that the work is original and has not been previously submitted for assessment in any other course or institution, except where specifically stated.

Bede Wilson

0050048951

Signature

Date

Acknowledgments

I must begin by thanking my supervisor Mark Phythian. Mark's expert guidance and assistance has been crucial in helping me to complete this project. Thanks also to USQ technical staff Terry and Brett and colleagues Dan and Brett for making their remote control aircraft and UAV experience available.

Finally, special thanks must go to my girlfriend Cass and my family and friends. Without their support this project would not have been possible.

Bede Wilson

University of Southern Queensland

October 2010

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction

The rise in natural disasters, combined with a decrease in the cost of miniaturised avionics has resulted in a significant interest in the use of UAV's. The response and recovery phases of the emergency management cycle often require some air support. On high fire danger days, aircraft are mobilised to spot fires before they get out of control. Once fires break out water bombing aircraft are used to combat them. Aircraft are also used after flood and storm events for damage assessment and to provide supplies and access to isolated areas. Smaller scale events such as searches also use air support, using both trained air observers and sophisticated imaging technology to assist in locating missing people.

The effect of using UAV's is twofold. The automation of tasks that can be considered as "dull, dirty and dangerous" improves the safety of air crews and emergency workers. Secondly, having UAV's perform aerial surveillance missions frees other aircraft for urgent supply drops, evacuations and deployments. Despite these advantages, the use of UAV's in natural disasters has been limited.

There are already a number of UAV's commercially available or in development that are capable of performing these functions. Companies such as Boeing, Lockheed Martin, Aerosonde and AeroVironment have all developed small UAV platforms. Research organisations such as the CSIRO are actively engaged in developing UAV's, both as an exercise in its self and as a platform for further research in areas such as machine vision. Many universities are researching UAV's, also for these reasons. Finally, significant work has been undertaken by hobbyists particularly in the development of cheap, disposable UAV's. Competitions such as the UAV Outback Challenge and SparkFun Electronics Autonomous Vehicle Competition as well as the DIY Drones on line community showcase the results.

In response to the activity in the civil UAV sector the Civil Aviation Safety Authority (CASA) has developed legislation governing the operation of UAV's. Although there are a number of restrictions relevant to this project, the overall legislative framework is considered to be conducive to further research without jeopardising public safety and is not likely to have a significant impact on this project.

The clear space for improvements in emergency air operations, combined with an active UAV research and development community and decreasing costs of miniaturised avionics means that the development of a stable UAV platform is a reasonable and very achievable objective.

1.2 Problem definition and research objectives

In order to guide the development of research objectives, an overall aim for the project was considered. While the design, building and testing of a fully functional UAV and its payload is beyond the scope of an undergraduate research project, it was considered that reasonable steps toward this goal could be made. Bearing this in mind, the development of any payload or ancillary systems not specifically related to UAV control was discounted. Rather, it was considered important that a functional UAV be built considering the intended use of the aircraft. This would allow for the UAV to form the base of any project wishing to consider the specific applications of UAV's to disaster situations. It is in this light that the overall aim of the project was developed.

- This project seeks to develop an unmanned aerial vehicle that could be used to obtain footage or photographs of areas affected by natural disasters.

In defining the specific research objectives a number of factors had to be considered. This primarily concerned the balancing of the vision for a finished product, a UAV capable of fully autonomous flight and navigation, and the time and resources available to achieve that. It was immediately clear that it would not be possible to design and build a fully functional UAV within the time constraints of this research project. Consideration then fell to which elements could be completed that would best support further research and development of the UAV.

Of high importance was the investigation of the legislative and regulatory framework as applicable to the UAV. In order to facilitate UAV research, the ability to fly and test the aircraft is crucial. This led to the first key objective which is:

- Research legislation and standards relating to the use of Unmanned Aerial Vehicles (UAV's) in Australia.

After evaluating the legislation and establishing the viability of the research, a comprehensive study of the existing literature would have to be completed. Hence the second key objective of the project is:

- Research and evaluate existing methods of UAV control.

After deciding on a control methodology, a parent vehicle needs to be selected in order to test the prototype design. Given the limitations regarding time and budget, this parent vehicle would be purchased in a kit or ready-to-fly state in order to eliminate the airframe design stage. This allows for the development and testing of a solid control system. If the requirements of the project develop such that a custom made airframe is required, this can be considered in any future work. As such the third objective is:

- Select an existing kit (fixed wing or rotary) for use as a parent vehicle.

With the initial stages of the project complete, the primary task then becomes the design an implementation of a control system capable of flying the aircraft. Again, available time became an important factor in determining the level of autonomy to be achieved by the control system. The most fundamental component of autonomous flight was considered to be some form of attitude stabilisation and so this became the primary goal of the control system. Extra levels of control would be added to the UAV if time permits.

- Design an electronic control system (hardware and software) capable of maintaining stable, horizontal flight and verify its operation external to the parent vehicle.

On completion of all of the above objectives, the final step in the process must be to adapt the fully functioning control system to the parent vehicle in order to complete tuning of the control system and extensive flight trials.

- Adapt the electronic control system for use in the chosen parent vehicle and verify its operation.

As discussed earlier, extra levels of control would be added to the UAV if time permits. These potential areas for additional work are:

- Integrate GPS way-point navigation into the control system to allow for fully autonomous flight.

  OR

- Adapt the control system to allow fully autonomous take off and landing.

These project objectives were documented at the beginning of the project and have been included in appendix A for reference.

1.3 Outline

This chapter seeks to present a background to the project, looking at the reasons for beginning the project in the first place and providing some vision as to what the end goal may be, not just for this project but, hopefully, anyone who continues this work in the future. As this is a research project the specific objectives to be tackled are also defined.

Chapter 2 will look at the available literature regarding UAV control and present the key elements as they relate to this project. Background on key points that are not specific to UAV's is also included to ensure that the reader is aware of the important underlying theory as well as the terminology that will be used throughout.

Chapter 3 looks at the legislation and standards relevant to UAV operations in Australia. This includes an examination of operating standards, particularly with regard to safety, that are currently in use in the UAV research arena.

Chapter 4 This chapter discusses the methodology to be employed throughout this research project. It also presents the broad system layout for the UAV and presents design criteria for each of the sub-systems. Finally, the process of major component selection is documented in this chapter.

Chapter 5 discusses the design of the UAV in detail. In this chapter, important elements of the design are exposed. Full schematics of the final system as well as software listings are presented in the appendices.

Chapter 6 then seeks to evaluate the design through both qualitative and quantitative methods and outline any important testing that was undertaken as part of the project. Where time permitted, solutions to any problems encountered have been developed and presented. Any changes that were not completed have been included as recommendations for future work.

Chapter 7 serves to summarise the achievements of the project and relate these both to the specific goals of this research project as well as the broader context of a completed UAV for use in natural disasters. Any problems highlighted in chapter 6 have also been included as recommendations for further work.

1.4 Conclusion

This chapter has given some background to the project, explaining the inspiration for the topic as well as some indication of its context in terms of the wider UAV industry. It has also defined the research problem and set out the key objectives to be addressed throughout the project. Finally, it has introduced the dissertation itself, outlining the focus of each chapter.

Chapter 2

Background

2.1 Introduction

This chapter serves to inform the reader of the key concepts related to the design of a small UAV. To do this, a review of the available literature has been conducted and its application to the current research problem discussed. This literature is related to a range of key concepts such as an introduction to flight dynamics and its applicability to aircraft control, sensing methods employed on UAV's and common UAV control methods. Legislation and standards relevant to the UAV project are not outlined here and are instead discussed in detail in chapter 3.

2.2 Flight Dynamics

2.2.1 Definition of symbols

While a detailed exposition of flight dynamics will not be presented in this dissertation, consideration must be given to the relevant terminology as well as any significant aerodynamic considerations related to the UAV design. Throughout this dissertation, the directions left and right with reference to the aircraft are taken facing the aircraft's direction of travel, in the same way that port and starboard are defined for a ship.

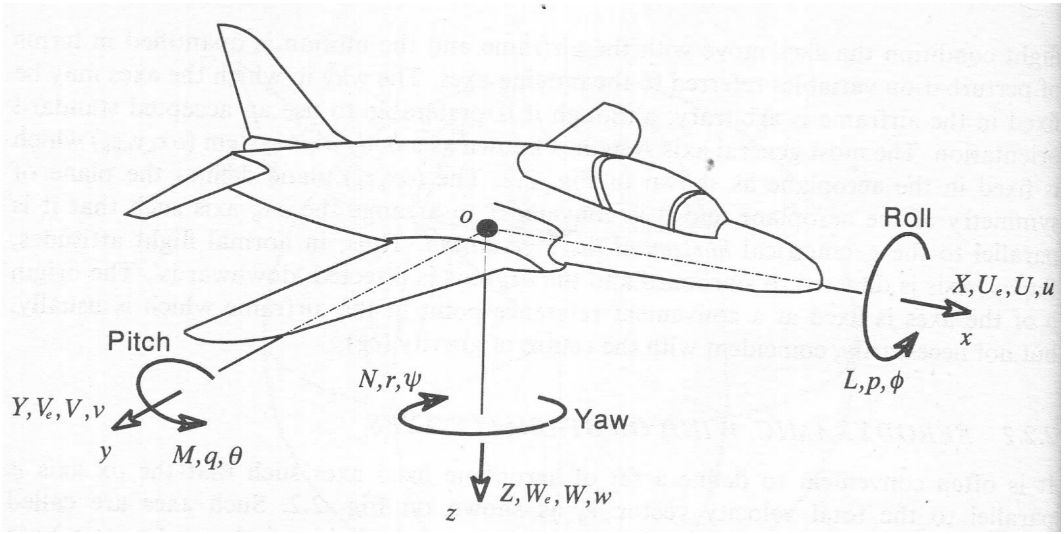


Figure 2.1: Definition of flight axes and variables (Cook 1997)

Cook (1997) denotes the axes and control surface deflections as follows (see figure 2.1). The roll or x axis can be seen passing from the centre of gravity through the nose of the aircraft. Rotation about this axis will cause one wing of the aircraft to move upwards and the other downwards. The angle of the wings is expressed as the roll angle ϕ, with a positive roll being to the right. The corresponding roll rate is expressed as the value p.

The pitch or y axis is the one passing from the centre of gravity through the right hand wing tip. Positive rotation about this axis will result in the nose of the aircraft pointing upwards and the tail of the aircraft downwards. This pitch angle is assigned the variable θ and pitch rate q.

The last remaining axis is z or yaw axis. Its direction is from the centre of gravity towards the ground. Rotation around the yaw axis is assigned the variable Ψ and the yaw rate r, with positive rotation being a turning of the nose towards the right hand side of the aircraft.

2.2.2 Flight axis control

In a four axis aircraft, these flight axes are controlled by ailerons, elevators and a rudder. The fourth axis refers to the forward motion from the motor and propeller. A four axis aircraft with these control surfaces labelled is shown in figure 2.2.

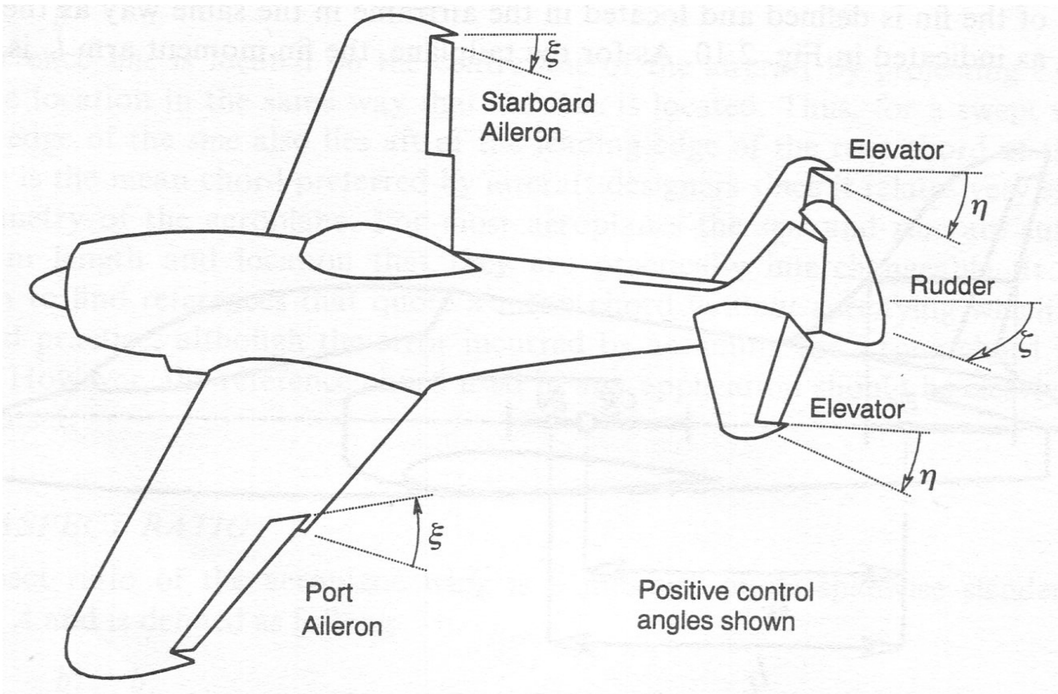


Figure 2.2: Control surfaces of a four axis aircraft (Cook 1997)

Note that a positive displacement ξ on the ailerons means the left aileron will move upwards and the right aileron downwards. This will effect a negative roll rate p and hence a negative roll angle ϕ.

Similarly, a positive elevator deflection η pushes the elevator towards the ground, resulting in a negative pitch rate q and hence negative pitch θ.

Finally, a positive rudder deflection ζ moves the rudder to the left of the aircraft, resulting in a negative yaw rate r and negative yaw Ψ.

It is possible to control the aircraft using only three control axes. In this case the ailerons are removed or left fixed, leaving the elevator, rudder and throttle as the three control axes. The pitch axis control operates in the same manner as that for four axis control. The yaw and roll axes however are both controlled by the rudder. Although the rudder primarily controls yaw, it also affects the roll of the aircraft. For example, consider the application of some negative rudder. The aircraft will turn about the yaw axis at a positive rate q. This will make the left wing of the aircraft move through the air faster that the right wing. This increase in air speed will result in an increased lift on the left wing and hence a positive roll moment. This control method, although less elegant and aggressive that four axis control is more than suitable for the control of small aircraft. Indeed, many ultralight aircraft also use this method for control.

Figure 2.3: 3-axis ultralight aircraft (Wiebe 2010)

2.2.3 Equations of motion

In order to undertake a numerical analysis of the flight dynamics, Cook (1997) outlines the derivation of the equations of flight for a standard four axis aircraft. These equations are formulated by considering Newton's second law. In terms of force this is expressed as $F = ma$ and in terms of moment or torque, $M = I\frac{d\omega}{dt}$. That is, the sum of all forces acting on the aircraft F is equal to its mass m multiplied by its acceleration a and the sum of all moments acting on the aircraft M is equal to its moment of inertia I multiplied by its angular acceleration $\frac{d\omega}{dt}$. With the mass and moment of inertia of the aircraft known[1], for a given applied force and moment the linear and angular acceleration of the aircraft can be determined. Integrating these values of acceleration will yield the relevant velocities and displacements. An image of the pitching moment model is presented in figure 2.4.

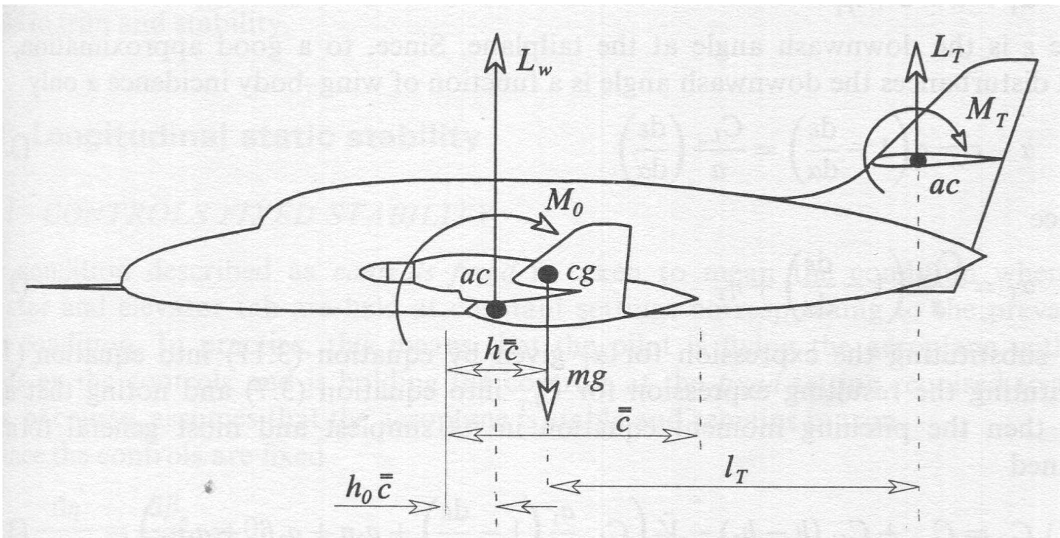


Figure 2.4: Pitching moment model (Cook 1997)

Without wishing to go into too much detail, the equations for force and moment can be derived based on the aerodynamic properties of the aircraft and combined with the equations of motion to give a set of differential equations describing the dynamics of the aircraft. These equations can also be linearised to simplify their solution.

[1] These quantities can be measured and calculated with relative ease, particularly for a small UAV.

The difficulty with this approach lies in a couple of areas. Firstly, determining the aerodynamic characteristics of an airframe in order to calculate applied forces and moments is not a trivial task. A significant amount of analysis, using a wind tunnel or Computational Fluid Dynamics (CFD) techniques would be required to obtain realistic aerodynamic coefficients. Secondly, the flight dynamics of an aircraft are inherently non-linear. As such the linearised equations of motion can only provide realistic results for small perturbations around the steady state. Indeed Cook (1997) refers to the equations as the 'small perturbation equations of motion'. Given the very large displacements the UAV is likely to be subjected to given its small weight and low airspeed, this detailed analysis approach is unlikely to warrant the effort required to pursue it. Nevertheless, the linearised flight equations do provide insight into the flight characteristics of an aircraft and at the very least may be used to inform some less detailed or qualitative analysis of the UAV.

2.2.4 Significant aerodynamic effects

Although numerical analysis of the aerodynamics of the airframe will not be undertaken, consideration must be given to the major aerodynamic effects that will be encountered in flight.

Stability

While Cook (1997) outlines a vast array of static and dynamic stability issues, only two key concepts will be considered here. These are classified broadly as lateral stability issues, or stability about the roll axis, and longitudinal stability, or stability about the pitch axis. Specifically this will consider the ability of the airframe to 'self-right' when subjected to atmospheric disturbances or other destabilising forces.

A major factor in lateral stability is the dihedral angle of the wings, as shown in figure

2.5. Dihedral refers to the angle of the wing above the horizontal. Consider an aircraft subjected to a small upwards draught on its left hand wing. The aircraft will roll to the right. In this position the lift on the right hand wing will act purely in the upwards direction, whereas the lift on the left hand wing will push the aircraft both upwards *and* to the right. This effect is called sideslip. As the aircraft is pushed sideways, the aircraft will yaw slightly, the angle of attack of the right hand wing will be increased and a rolling moment against the disturbance will result. In this manner the dihedral angle contributes to the ability of the aircraft to self-stabilise in the lateral axis.

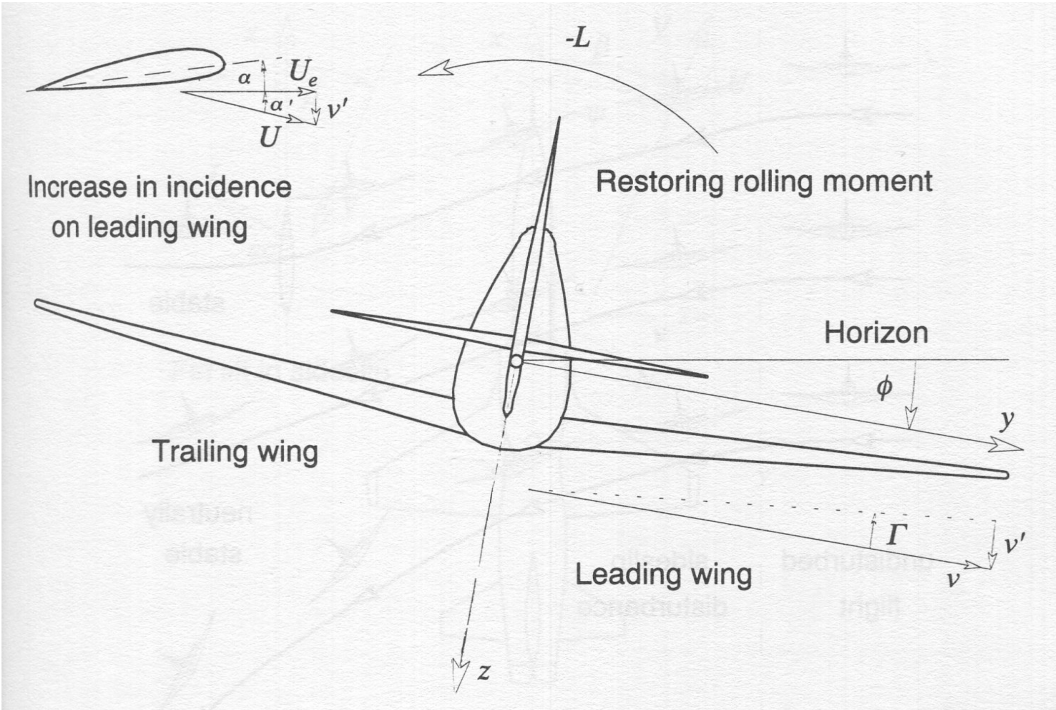


Figure 2.5: Dihedral angle Γ (Cook 1997)

Although there is no equivalent ‘self-righting‘ property for the longitudinal axis, Cook (1997) does outline some key contributors to the stability or otherwise of an aircraft in this axis. For example, Cook discusses the effect of the centre of thrust of the aircraft's engine. Aircraft with a thrust line further away from the centre of mass will experience a greater pitching moment due to the output of the engine. Cook goes on to state that factors such as flexibility in the airframe and a large backward sweep on the wings all ‘generally tend to reduce the available stability’.

Coupling of flight axes

One case of the coupling of flight axes has already been examined in section 2.2.2 regarding three-axis control of an aircraft. That is, the tendency of the aircraft to roll due to the yawing motion of the aircraft. In a similar vein, this effect will occur in reverse; an aircraft that is in roll will tend to yaw as well. This yaw is caused by the sideslip effect already discussed in section 2.2.4. This yawing effect will be considered later as a method of detecting an error in the attitude detection system.

2.3 Method of Sensing

2.3.1 Infrared horizon detection

Taylor et al. (2003) give a detailed discussion on the use of an array of infra-red thermopile sensors to detect the earth's horizon and use this as a means to orientate the aircraft in the pitch and roll axes. Infra-red thermopiles can be used to measure temperature over a field of view typically in the order of 100° (Adiprawita, Ahmad & Sembiring 2007*a*). The earth is warmer than the sky so a thermopile on the tip of a wing will indicate a higher temperature as it points towards the ground and a lower temperature as it points towards the sky. If a thermopile is placed on the tip of each wing, the difference in temperature between the two can be used to determine the roll angle of the aircraft. An illustration of this concept is shown in figure 2.6. Crucially this requires the maximum (ground) and minimum (sky) temperatures, or the maximum difference between these values, to be known. Due to this the system must be re-calibrated prior to each flight. Placing an extra pair of sensors on the z axis allows for automatic recalibration and also allows the aircraft to operate at bank and pitch angles greater than the sensor's field of view (Taylor et al. 2003). Iscold et al. (2010) and Adiprawita, Ahmad & Sembiring (2007*b*) also use this approach, citing cost and weight savings as well as simpler hardware than other methods as advantages.

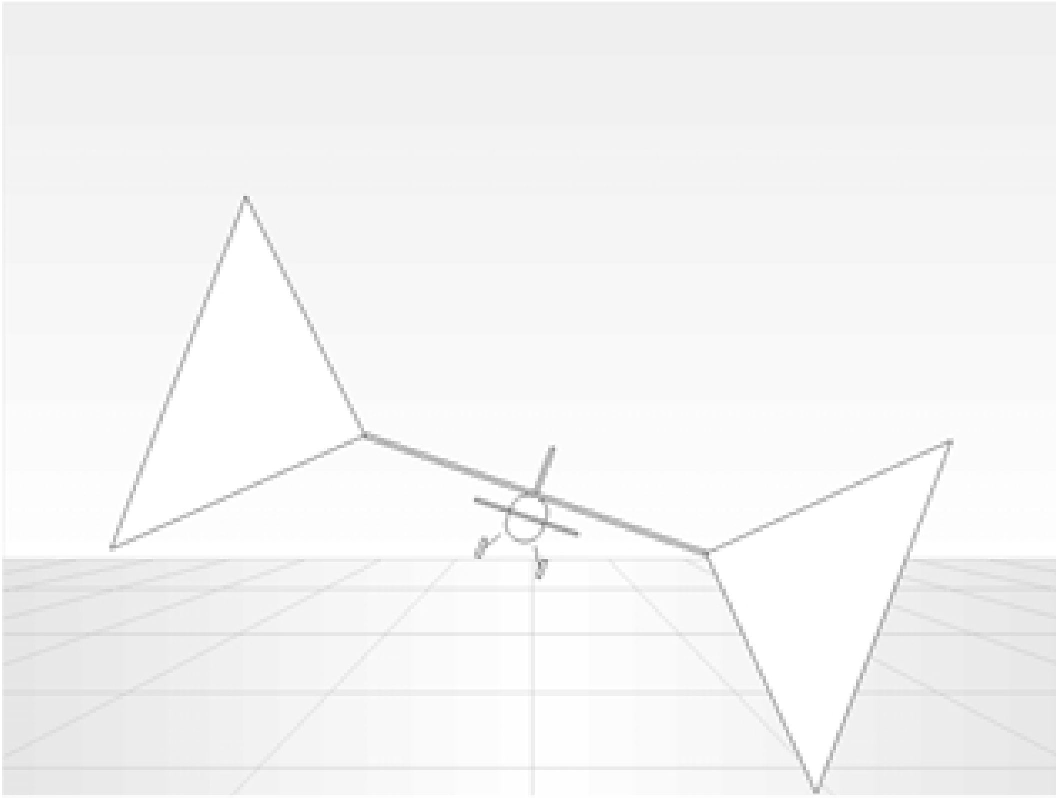


Figure 2.6: Thermopile sensor configuration for detecting the roll angle of an aircraft (Taylor et al. 2003)

The effectiveness of this approach relies on some assumptions. Egan & Taylor (2007) discusses in detail some of the characteristics of infrared thermopile sensors pertaining to their use in UAV control systems. One critical assumption used in Taylor et al. (2003) is that the sensor window is a square shape and so the angle of the aircraft can be calculated directly from the average temperature across the sensor. This is evident in the formula presented for the calculation of an absolute roll angle.

$$T_{avg} = \frac{\theta_{sky} \times T_{sky} + \theta_{ground} \times T_{ground}}{\theta_{total}} \tag{2.1}$$

Knowing the average temperature over the sensor T_{avg} and knowing that $\theta_{sky} + \theta_{ground} = \theta_{total}$, the roll angle can be determined. In fact, the work completed by Egan & Taylor

(2007) found that the temperature/angle was not linear and could be better modelled by a modified sine curve. This improved model is illustrated in figure 2.7.

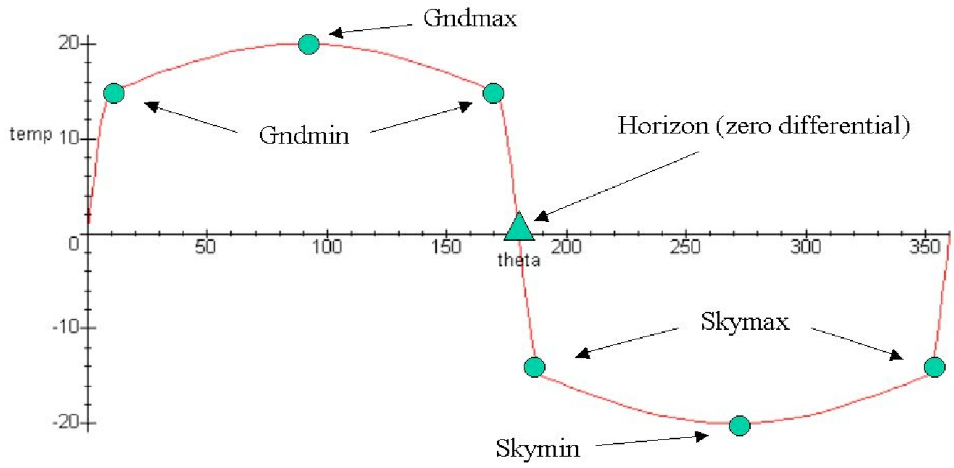


Figure 2.7: Revised model of temperature relative to roll angle for infrared thermopile sensors (Egan & Taylor 2007)

Although this revised model does provide a better estimation of attitude, the results of Taylor et al. (2003) do indicate that the simplified approach is more than adequate for aircraft control.

The main limitation of using thermopiles as a method of attitude detection is that the system cannot operate outside of Visual Meteorological Conditions (VMC). Weather conditions such as low cloud or precipitation can result in errors in attitude readings, hampering the control system. Egan & Taylor (2006) suggests the addition of a yaw gyroscope to detect inconsistencies in attitude measurements. The principle behind this method of detecting error relies on the sideslip effect outlined in section 2.2.4. Any error introduced by cloud cover or moisture affecting the thermopile sensor will result in an incorrect reading of attitude and a pitch or roll from the aircraft in an attempt to correct this. The resulting sideslip will be detected by the yaw axis gyroscope. At this point, the control system will detect the inconsistency between the 'stable flight' state as detected by the attitude sensor and the yawing state detected by the gyroscope. This inconsistency could be used to commence some form of error correction routine or flag to the operator that manual control is required to fly the aircraft out of the cloudy conditions. It should be noted that flying into the cloud in the first place would be a breach of the Civil Aviation Safety Regulations. The offence regarding the flight into cloud or in non VMC conditions is one of 'strict liability' meaning that regardless of the intent of the operator, changes in conditions or other factors, *any* unauthorised flight into cloud is an offence and must be prevented by adequate flight planning and procedures in the first instance.

Note that the actual performance of the control system outside of VMC cannot be determined as non VMC flights of UAV's have been prohibited (CASA 2003).

2.3.2 Inertial measurement

An alternative approach to determine the UAV's attitude is by using a combination of gyroscopes, magnetometers and accelerometers. These components are combined to form an Inertial Measurement Unit (IMU) responsible for detecting the attitude and heading of the aircraft. These systems are available as proprietary modules (Iscold et al. 2010, Mendelow, Muir, Boshielo & Robertson 2007) or custom built designs (Adiprawita et al. 2007*a*). Egan & Taylor (2006) contends that although an inertial measurement approach has previously been the default option for flight control, it is not necessarily an appropriate solution for small UAV's. The low mass of small UAV's means that they are particularly vulnerable to turbulence. Importantly, "[if] for any reason the aircraft adopts a vertical flight attitude, computational singularities can lead to computed pitch/roll angles which are 180° in error" (Egan & Taylor 2006).

The work in Adiprawita et al. (2007*a*) also highlights some of the extra complexities involved in using inertial measurement systems compared to the infrared detection scheme already discussed. Firstly, accelerometers are used to detect the direction of acceleration due to gravity and hence the attitude of the aircraft. They will also detect the acceleration due to the motion of the aircraft so this must be filtered out. The susceptibility of the sensors to noise is also greater than that of the thermopiles and as such a more comprehensive and robust filtering approach is required. Finally, the challenge of combining information from a range of different sensors requires more complex computations, typically in the form of a Kalman filter, and hence greater processing capability on board.

2.3.3 Navigation

All of the publications previously discussed use a GPS receiver to determine the latitude and longitude of the UAV. The GPS receiver is also used to detect the heading of the aircraft by comparing its current position with its position in previous updates. This data is fed into the UAV's navigation system to determine flight paths. It is important to note that the reliance on this differential approach means that when the aircraft is not moving, such as prior to take off, the direction it is facing can not be determined. This is also an inherent difficulty in rotary wing platforms that, during hover, will not be moving relative to the earth.

Typical refresh times for small GPS receivers are in the order of 1 to 5 Hz, inadequate for firm control of the UAV. To combat this, the yaw gyroscope already installed to detect attitude reading errors can be co opted to provide extrapolated headings between updates (Egan & Cooper 2006).

As navigation is not a key objective of the project at this stage, consideration will not be given to methods of calculating flight paths and other more detailed topics in this sphere.

2.4 Servo Motor Control

Although there are a range of methods for controlling servo motors, only one method will be considered here. This method is known as Pulse Position Modulation, or PPM. In actual fact, both pulse position and pulse width affect the control of the servo. Endurance RC (2010) gives an introduction to this process as it applies to remote control servo motors.

Each servo will have a neutral position, typically labelled as its 0° position. Sending a 1.5 mS pulse to the servo control line will command the servo to hold this neutral position. This pulse is typically applied to the control line every 20 mS, a consideration that will be examined in more detail later. This is typical of all analogue servo motors in the remote control arena.

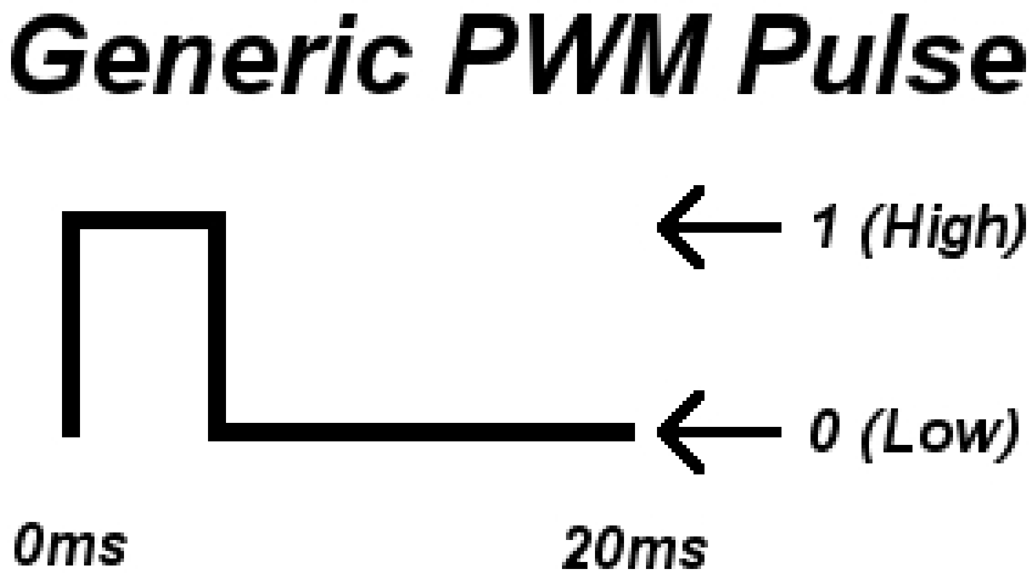


Figure 2.8: Servo motor control signal pulse (Endurance RC 2010)

The servo will also have a minimum and maximum value either side of this neutral position. To drive the servo to its minimum position, the servo control signal pulse width should be reduced to 1 mS (nominal). Conversely, to drive the servo to its maximum position, the servo control signal pulse width should be increased to 2 mS (nominal). The true duration of these pulse widths is dependent on the manufacturer of the servo motor. For example the minimum pulse width is as low as 0.5 mS on some common brands of remote control servo motor. This component of the signal is known as Pulse Width Modulation (PWM) i.e., the servo position is encoded as a function of the pulse width of the signal.

As has already been mentioned, the time between pulses for a single servo is 20 mS. This value is also only nominal and is not particularly important to the operation of the servo. Endurance RC (2010) indicates that being anywhere within a few milliseconds of this value is precise enough. The reason for this delay is a crucial one. One radio signal is required to control a number of servos. For a standard four-axis aircraft, four servo control signals are required. Any additional features added to the aircraft could increase the required number of signals to six or even eight channels. Figure ?? illustrates how this signal can be expanded to control a number of servos. In this example, the first pulse width corresponds to the first servo channel, the second pulse width to the second servo and so on. With a nominal time between pulses of 20 mS per servo, up to ten 2 mS pulses can be delivered via this PPM scheme. As such, 10 channels is the nominal channel limit for this method of servo control.

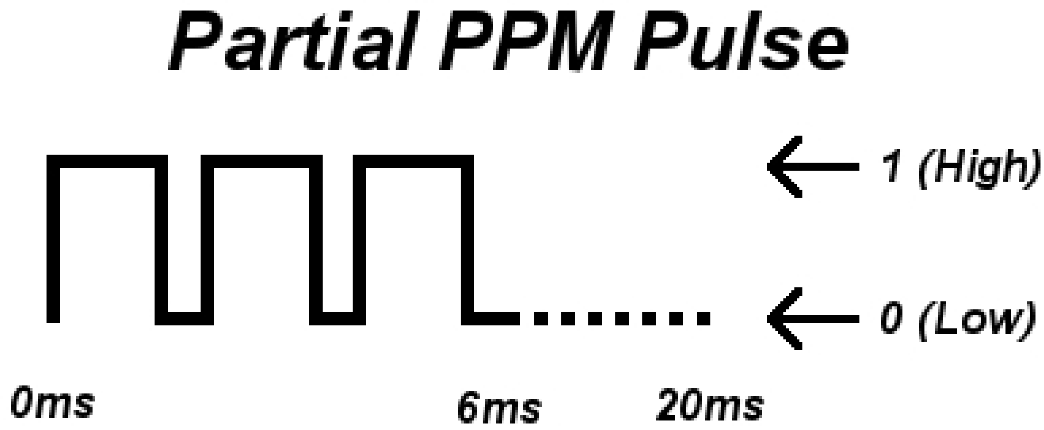


Figure 2.9: Using Pulse Position Modulation (PPM) to multiplex the servo control signals (Endurance RC 2010)

This information will be crucial to the development of the UAV control system, particularly to the fly-by-wire control mode of the UAV.

2.5 Control Methodology

All of the literature discussed to this point uses PID loops to facilitate flight control. Neural networks and fuzzy control methods such as those described in Puttige & Anavatti (2008) and Kurnaz, Kaynak & Konakoğlu (2007) have been discounted due the the significant processing overheads and the time required to develop and train a functional controller.

Nise (2008) gives a good explanation of the fundamental principles of PID control. PID stands for Proportional, Integral and Derivative, giving some indication as to the operation of this control method. A diagram of a typical PID loop is shown in figure 2.10. The summing junction in the control loop calculates the instantaneous error as given by the difference between the current output subtracted from the desired output. This error signal is then fed into the PID element of the loop which calculates the required intervention to achieve the desired output. The proportional element of the controller effects a response that is proportional to the error i.e., a larger error results in a larger response. In the time domain, $g(t) = K_P \cdot e(t)$, where K_P is the proportional gain term. The derivative term delivers a response proportional to the rate of change of the error. That is, $g(t) = K_D \cdot \frac{de(t)}{dt}$. Finally, the integral term delivers a response proportional to the sum of all previous errors, or $g(t) = K_I \cdot \int_0^t e(t)dt$. When all of these feedback terms are combined, the mathematical representation of the controller is as shown in equation 2.2.

$$g(t) = K_P \cdot e(t) + K_I \cdot \int_0^t e(t)dt + K_D \cdot \frac{de(t)}{dt} \tag{2.2}$$

Despite all of the above UAV's using PID loops, there are some important differences

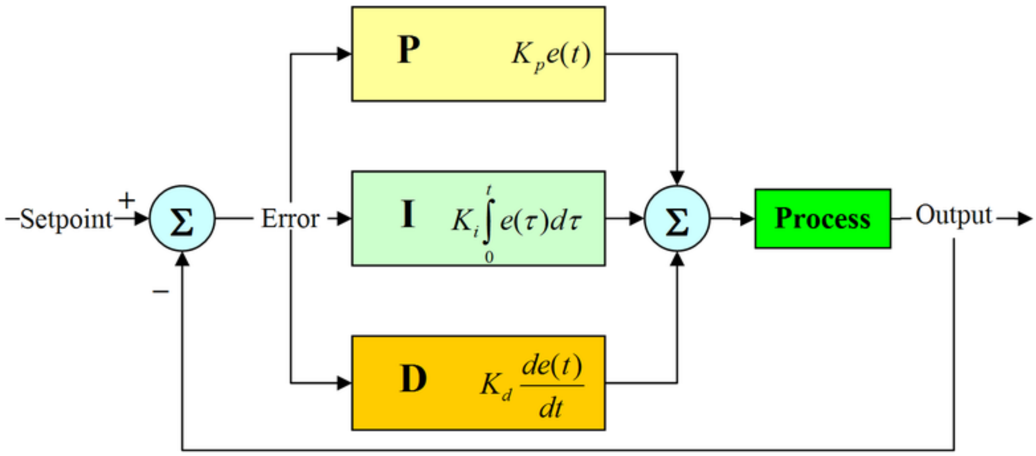


Figure 2.10: Basic PID control loop structure (SilverStar 2006)

in the way these have been implemented. Control systems can consist of a basic PID structure (Taylor et al. 2003) or include an array of feed forward paths to ensure appropriate control is applied during all flight conditions (Iscold et al. 2010). Figure 2.11 shows a comparison of the two different approaches.

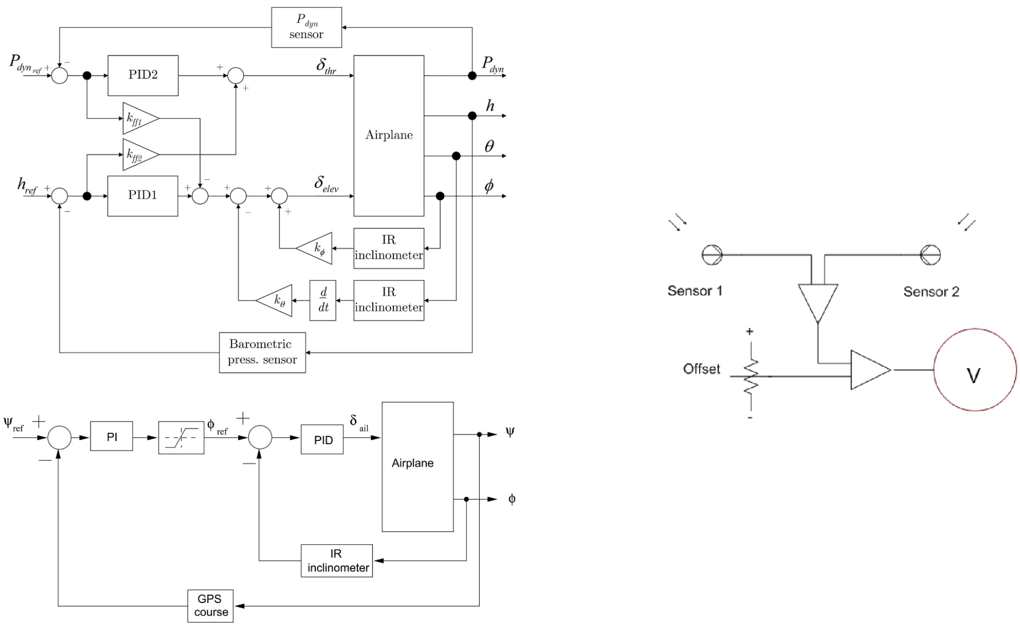


Figure 2.11: Comparison of control loop topologies left (Iscold et al. 2010), and right (Taylor et al. 2003)

Regardless of the detailed design of the control loops, no solution will be functional without appropriate tuning. Although detailed simulations and system identification can be undertaken (Johnson & Fontaine 2001), Egan, Cooper & Taylor (2004) advocates the use of in flight auto tuning using the Ziegler-Nichols method. The primary reason given is that the behaviour of small aircraft is particularly difficult to simulate reliably. Note that this relies on the aircraft using electric powered propulsion so that the mass of the UAV does not change with fuel consumption (Egan et al. 2004). If the UAV is to be powered by an internal combustion engine and the fuel tank is sufficiently removed from the aircraft's centre of mass the moments acting in the longitudinal flight axis could change significantly. This effect can be mitigated by the integral control term in the control system but this could result in the UAV having less capacity to respond to disturbances of flight.

Nise (2008) also gives a more detailed explanation of the Ziegler-Nichols tuning approach. The first step is to set all gains to zero and then increase K_P until the output begins to oscillate. The gain at which the output oscillates is called K_U or the ultimate gain, and the period of the oscillation T_U. The values for K_P, K_I and K_D can then be calculated according to table 2.1.

	K_P	K_I	K_D
PID Coefficients	$0.65 \cdot K_U$	$0.5 \cdot T_U$	$0.12 \cdot T_U$

Table 2.1: Ziegler-Nichols tuning coefficient calculation (Nise 2008)

Egan et al. (2004) also notes that setting control gains as a function of airspeed is important to achieve reliable performance. This is because a fixed control surface deflection will have more effect at high air speeds compared to low air speeds. This is because aerodynamic lift $L \propto V_0$ (Cook 1997).

2.6 Conclusion

A range of literature relating both specifically to UAV's and to key concepts underlying UAV design has been considered. In doing so consideration has been given to the fundamental concepts of flight to a level required for initial prototype design. These concepts will form the basis of the parent vehicle selection process describe in chapter 5. A more in detailed investigation was also undertaken to determine the dominant methods of UAV control, related specifically to sensing and control system design. These investigations will be central to the development of the UAV's electronics platform.

Chapter 3

Legislation and Standards

3.1 Introduction

This chapter seeks to identify and examine any legislation applicable to UAV operations within Australia. Consideration is also given to any standards or common industry practises to identify any safety improvements that could be achieved in addition to complying with legislative requirements.

3.2 Civil Aviation Safety Regulations

Civil Aviation Safety Regulations (CASR) 1998 is the governing piece of federal legislation for all civil aviation activities. *Part 101 Unmanned aircraft and rocket operations* within this legislation details requirements specific to unmanned aircraft. *Part 101* has a wide scope covering everything from model aircraft to full scale UAV's. As such, a summary of legislation relevant only to this project will be given. To do this, a description of *one* area that would be suitable for UAV testing is presented along with relevant considerations for that area only. Although this area could be extended by applying for the relevant exemptions and approvals from CASA, this is not considered necessary at this stage of the project. Note also that this is only a summary. *Part 101* must be read as a whole by any person considering undertaking UAV operations of any sort.

Area of operation

With regard to the area the UAV may be operated, a number of considerations are relevant. This area must be:

- Outside of any prohibited or restricted area (subject to *Airspace Regulations 2007*)
- Outside of controlled airspace
- Below 400ft Above Ground Level (AGL)
- Not over a populous area
- In an area such that aircraft are not obstructed from approaching or landing at an aerodrome

Weather and time of day

The UAV must not be operated in the following conditions:

- In or into cloud
- At night
- In weather other that Visual Meteorological Conditions (VMC)

Aircraft specific regulations

In *Subpart 101.F UAVs*, CASA makes clear the difference between a small UAV and a model aircraft. That is:

> There is no practicable distinction between a small UAV and a model aircraft except that of use - model aircraft are flown only for the sport of flying them.

Given that the express purpose of this aircraft is for research and although the parent vehicle *is* in fact a model aircraft, this system must be considered a 'UAV' as defined in *Part 101* and hence is subject to all of the regulations of *Subpart 101.F UAVs*. As such the relevant considerations are:

- The UAV must be an aeroplane of less than 150 kg in weight or a rotary wing aircraft of less that 100 kg in weight [1]
- The UAV stays more that 30m away from people not directly associated with its operation

Most of the rest of *Part 101* deals with certification and other requirements not required for a small, non-commercial UAV and as such will not be detailed in this dissertation.

3.3 Other Legislation

National parks or other reserves may also pose some restriction to flight activities and these must be assessed on a case by case basis as required. At this point no

[1] It is presumed that the UAV will exceed 100g in weight and hence is not exempt from these regulations

other legislative requirements have been identified that would apply specifically to UAV operations.

3.4 UAV Specific Standards

Some research organisations have also developed rules in addition to the requirements of the *CASR*. Once such example is given in Taylor & Egan (2006). This is a comprehensive flight manual for UAV operations at Monash University. It covers many potential safety issues not directly legislated for in the *CASR* and should be considered prior to any extensive in-flight testing at USQ. Another example is the rules for the 2010 UAV Outback Challenge (UAV Outback Challenge 2010).

3.4.1 Monash UAV Operations Flight Manual

The Monash regulations require any UAV operator to be members of the Model Aeronautical Association of Australia (MAAA). This has a number of advantages. Firstly, the MAAA is an approved aviation administration organisation and is responsible for controlling a number of regulations that apply to model aircraft. For example, *Part 101* stipulates that:

> A person may operate a model aircraft at night only in accordance with the written procedures of an approved aviation administration organisation.

Being a member of the MAAA allows for the use of the UAV under these regulations, allowing for a wider range of test conditions than would otherwise be available when flying under the *Part 101* UAV regulations.

MAAA also operates a number of airfields and 'approved areas' designated for the flight of model aircraft. Membership of the MAAA allows access to these areas for testing and reduces the need to find land outside of populous areas to conduct flight testing.

Taylor & Egan (2006) also outlines a range of Occupational Health and Safety (OHS) regulations specific to Monash university. While these considerations are not directly relevant, USQ does have equivalent Workplace Health and Safety policies and procedures and these must be considered at all times.

The operations manual also imposes a range of additional requirements. These include the fitting of a Flight Termination System (FTS) allowing a flight to be manually ended. Egan et al. (2004) describes this FTS in more detail. If the radio control signal is lost for a duration of 2.5 s, the aircraft will release a parachute. This parachute deployment also physically breaks the power connection to the motor, preventing the UAV from 'running away'. Egan et al. (2004) also discusses potential future developments for the FTS. For example, on long range flights the UAV will move out of the range of the standard remote control. For this reason a 'low power VHF beacon' will be developed to provide the dead-man signal to the aircraft and it will be the loss of this VHF signal that will result in the FTS deploying. Crucially, the FTS is *not* designed to protect the aircraft from damage and deployment of the FTS is likely to result in significant damage to the UAV.

Some of the other requirements include restrictions on weight additional to those outlined in *Part 101* and specific limitations on propulsion methods.

Another important element of the regulations is that a number of positions have been appointed with specific responsibilities with regard to UAV flight operations. Taylor & Egan (2006) list these responsibilities in detail so they will not be repeated here, but a summary of the positions has been compiled.

Safety Officer

The safety officer is responsible for ensuring that appropriate safety equipment such as fire extinguishers, a first aid kit, mobile telephone etc. are available on site. The safety officer is also responsible for ensuring compliance with a range of regulations regarding issues such as site layout, and Personal Protective Equipment (PPE). The safety officer is also responsible for formal reporting of safety breaches and incidents.

Duty Pilot

The duty pilot is responsible for controlling all flight operations. The duty pilots responsibilities include spectrum management to ensure that there is no radio interference between UAV's. They also control the timing and sequence of flights and are responsible for a range of record keeping requirements imposed by the Monash UAV research group.

3.4.2 UAV Outback Challenge 2010

UAV Outback Challenge (2010) also requires pilots to be 'MAAA gold-wing standard or equivalent'. Equivalency is determined by a number of proficiency tests for those participants who are not MAAA members.

Prior to taking part in the UAV Outback Challenge a number of safety checks are required. These include an inspection and demonstration of operation of all critical components in the aircraft, followed by a demonstration flight.

During the challenge only one UAV is operating at any one time so spectrum management is not a prime consideration. That said, the challenge organisers have introduced restrictions on the radio frequencies used to prevent systems within the UAV from interfering with each other. Accordingly, the radio control, video and data elements of the UAV must operate on different frequencies. A UAV operating with a 900 MHz data link, 5.8 GHz video link and 2.4 GHz radio control link is given as an example of an acceptable solution for use in the competition.

There are a range of rules concerning the handling of in flight failures and emergencies. Firstly, it must be possible for the UAV operator to override the UAV control system if required. Some further action must be taken in the event that any one of a number of failure conditions are met. These conditions include the aircraft moving outside of the mission boundary, losing data or GPS connections or if the judges deem the aircraft to be 'out of control'. If GPS, radio or data connections are lost, two options for flight termination are available. The first is for the aircraft to automatically return to a 'comms hold' rally point in order to regain communication. In the event that communication is not regained the aircraft can return to the airfield in a further attempt to regain the communication link. Continued failure to regain communications means that the operator is then required to terminate the flight. A number of other failure conditions and their associated control procedures are included but they are primarily variations of the return to way point method already described. If all safety systems ultimately fail or the aircraft breaches the mission boundary, the aircraft must enter the flight termination mode and return to the ground. This can be via a parachute as already described or by deliberately 'ditching' the aircraft.

3.5 Conclusion

This chapter has examined the legislation and standards relevant to UAV operations in Australia. *Civil Aviation Safety Regulations Part 101 Unmanned aircraft and rocket operations* was discussed and the legislative requirements specific to this project were examined. The standards for operation for both the Monash University UAV Group and the UAV Outback Challenge 2010 competition were also examined in order to identify relevant standards that may enhance the safety of UAV operations beyond the base line legislative requirements. The need for a manual override system as well as a method for terminating flight were recurring themes in both standards, as well as the need for appropriate spectrum management and general safety considerations consistent with workplace health and safety procedures. While not all of these considerations will be immediately relevant to this project due to the time restraints on flight testing, all points must be kept in mind in order to ensure that any testing that may occur as part of a future project can be done so safely and in accordance with the law.

Chapter 4

Methodology and Prototype System Design

4.1 Introduction

This chapter will outline the methodology used throughout the project including the processes used for the system design, detailed design, evaluation and testing. The system design process will then be examined in more detail, looking at the criteria for major components and how they were selected. The major components considered are the parent vehicle, including supporting components such as the remote control, and the microcontroller and associated development system.

4.2 Methodology

The first step in the research process, after defining the research problem and objectives, was to lay out the UAV system in block format. This broad system design would give an indication of the major components required in the aircraft. Key design criteria were also defined in this step in order to assist in the evaluation and testing stage.

The detailed design phase primarily considered the control system as no significant modifications to the parent vehicle were required. The control system includes both the hardware and the software required to control the aircraft. Having already selected the microprocessor for the control system in the system design stage, hardware design was related primarily to the supporting circuitry required for the control handover process as well as the method of interfacing the remote control receiver, attitude sensors and servo motors. Once this hardware was designed and constructed the final step in the process was to write the software.

During the evaluation and testing stage of the project, a number of distinct elements are under inspection. Firstly, it must be established that the element being tested performs as it was designed to. Secondly, in must be ensured that the element under inspection interacts appropriately with other subsystems in the UAV. Finally, it must be confirmed that the design concept is appropriate to the UAV. In other words, is the chosen approach a suitable way of addressing the problem at hand or are there more appropriate methods of performing the same task. Following this process will ensure that the UAV solution addresses the ultimate design intent as well as simply being free of bugs or errors.

Once all of the conceptual and design issues are addressed extensive flight testing can commence, with the intent of both tuning the control system for autonomous flight and further evaluating the performance of the UAV.

4.3 System Topology

As an initial step towards the UAV design, the topology of the UAV system has been defined. This may ultimately change as the project progresses but will form the basis of development work at this stage. The UAV has been split into two main subsystems.

These are the parent vehicle and the control system. Future development of the UAV beyond this project will see this expand as payloads such as imaging modules would be added at this level.

The parent vehicle subsystem consists of a number of major components. These are the airframe itself, the motor, speed controller and propeller, the remote control system, servo motors and batteries. All of these components will be purchased off the shelf if available and suited to the UAV application.

The other main element to the UAV is the control system. Broadly speaking this can be separated into hardware and software subsystems. The hardware elements will include the microcontroller(s) and any required supporting hardware, sensors, interfacing and handover control. The software components will overlap these areas. Important modules will be the reading of remote control signals, control of servo motors, handover control, reading of sensors and the control loop itself.

This system is depicted as a flowchart in figure 4.1.

4.4 Parent Vehicle Selection

The first consideration in selecting a parent vehicle was whether to use a fixed wing (aeroplane) or rotary wing (helicopter) platform. Ultimately this decision was made primarily on consideration of the intended application of the aircraft. Using a fixed wing platform would allow for greater flight times than an equivalent rotary wing solution. In terms of a UAV used to perform a rapid impact assessment for example, this would allow for the collection of more information per launch, simplifying the operation. Although a rotary wing platform could allow for greater manoeuvrability this was not considered to be a big advantage for this application. Also, although the ability to take off and land vertically would be useful, many small fixed wing platforms are able to be launched by simply throwing them into the air and they do not require much

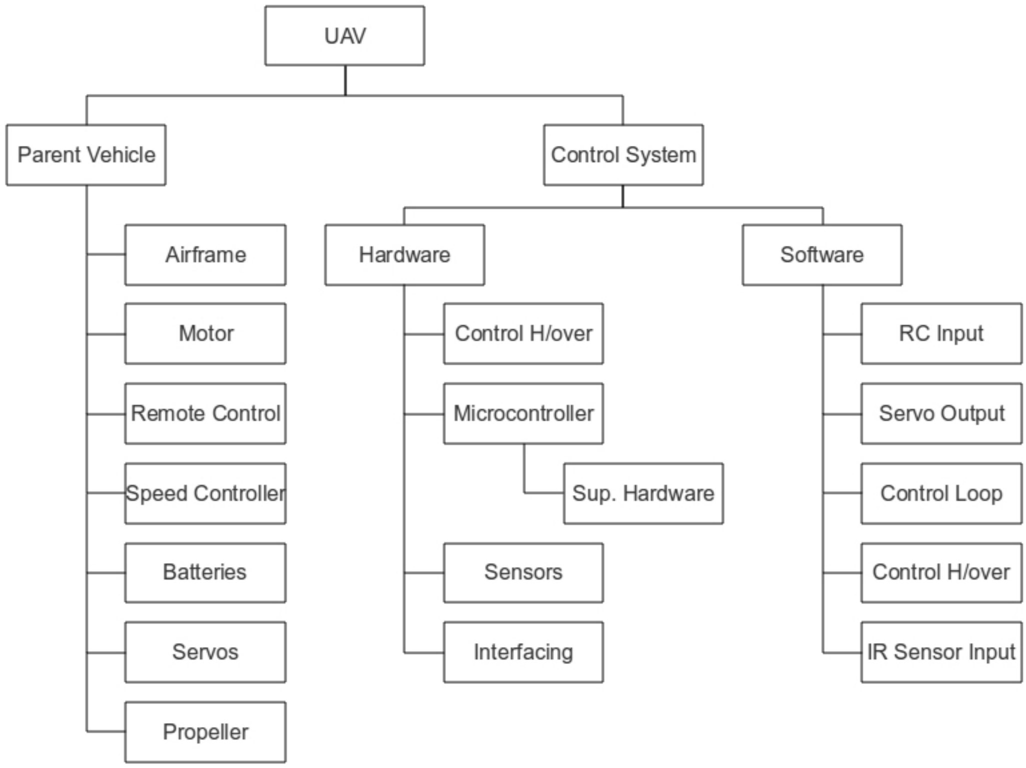


Figure 4.1: UAV system topology

horizontal space to achieve a reasonable height to urban obstacles such as trees and buildings. Due to this it was decided to use a fixed wing aircraft for the development of the UAV.

To aid in selecting the parent vehicle a set of selection criteria was developed. Each candidate aircraft was then assessed according to these criteria and given a ranking which ultimately determined the vehicle chosen.

For the purpose of this project, only off the shelf remote control aircraft were considered. There were a number of reasons for this. Firstly, it was beyond the scope of the project to design and construct an airframe. Cost and time were also limiting factors in the project and purchasing a 'ready-to-fly' or 'almost-ready-to-fly' aircraft kit allowed for a fast and relatively cheap solution.

4.4.1 Selection criteria

Stability

The stability of the parent vehicle was of prime concern. Having an aircraft that is stable in flight decreases the workload on the flight controller, requiring less interventions to correct aircraft attitude. Also, having some degree of self correction such as that provided by the dihedral angle means that the control system is not required to make harsh corrections as it is given some assistance by the airframe itself.

Propulsion

Consideration of the aircraft's propulsion system involved two major components. Firstly, whether the aircraft is powered by an electric motor or internal combustion engine and secondly, the configuration of the propeller.

Although an internal combustion engine allows for greater flight times, the use of fuel brings the added complexity of having the mass of the aircraft change throughout the flight as it is consumed. This in turn affects the tuning of the control system and some compensation may be necessary. On the other hand use of an electric motor resolves this issue but the batteries used for power do have a lower energy density that fuel based systems.

The two most popular ways of mounting the propeller are to place it on the front of the aircraft, 'pulling' the aircraft through the air, or behind the fuselage, 'pushing' the aircraft. Using a pusher arrangement the propeller is more protected in the event of a crash or hard landing and so this is considered to be preferable.

Gliding characteristics

The ability of the parent aircraft to glide was considered to be an important characteristic in terms of the selection process. This gliding ability allows for the reduction of throttle during flight to extend aircraft range. Importantly, it also suggests that the aircraft is less likely to plummet to the ground in the event of a loss of power or other similar event.

Payload capacity

Although important, the payload capacity was not considered as a critical factor in the selection process. The expected weight of any flight stabilisation controller is minimal and at this stage no cameras or other heavy equipment were to be fitted to the aircraft. Information regarding the payload capacity of the aircraft was also particularly difficult to find and hence a qualitative assessment of the payload capacities was undertaken after speaking with experienced model aircraft pilots.

Payload capacity also included a consideration of the space available to install extra electronics, batteries and any other components that may be required.

Ease of control

Ease of control essentially boiled down to a decision between three-axis and four-axis control. That is, whether the aircraft is controlled using a rudder-elevator-throttle combination or if control of ailerons was included as well. Under four-axis control, the roll of the aircraft is achieved using the ailerons. Three-axis aircraft use the coupling of the yaw and roll axes to achieve the same result. For simplicity, three-axis control is favoured.

Ease of modification

Although significant modifications to the airframe were not expected to be required, the ability to easily fit control circuitry and sensors to the aircraft was important.

Durability

The durability of the aircraft was a key factor in the selection process. Although failure resulting in a crash landing was not considered to be likely, even a hard landing could result in significant damage to the aircraft and added components. The primary construction methods for off the shelf aircraft were a ply/balsa combination or Expanded Poly-Propylene (EPP) foam. EPP foam is far more resistant to damage and as such is favoured over the ply/balsa combination.

Cost

The cost of the aircraft system was not used as part of the ranking process as such. Rather, the focus was on selecting the 'right' system in terms of the other selection criteria, with the condition that the parent vehicle and all supporting components such as remote controls, batteries etc. should be under $1000.

4.4.2 Aircraft details

The three aircraft considered for use as the UAV parent vehicle were the Multiplex EasyStar, the Phoenix Boomerang 60 and Phoenix Classic EP. A summary of their specifications is given in table 4.1. This table also shows how the characteristics of each aircraft scored on a relative scale of one to ten, their combined score and finally, which aircraft was chosen as the UAV platform.

On considering all of the criteria already discussed, the Multiplex EasyStar is the preferred option and hence has been chosen as the parent vehicle for the UAV.

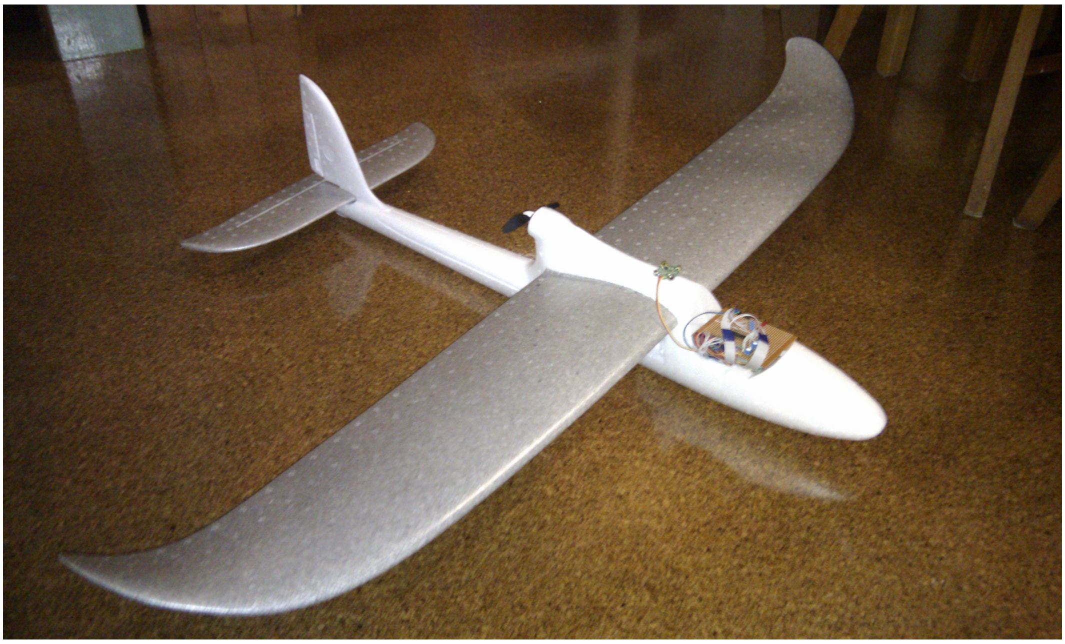


Figure 4.2: Multiplex EasyStar remote control foam glider (shown with control system fitted)

4.4.3 Additional components

After selecting the parent vehicle a range of additional components needed to be chosen. The Multiplex EasyStar kit includes a number of these including the motor, propeller and speed controller. Other components that were not included were the remote control system, batteries and servo motors.

Remote control

The requirements for the remote control in this project were relatively simple. The primary concern was ensuring the remote control system had enough channels to fly the aircraft manually as well as control the functions of the attitude stabilisation system.

The aircraft itself requires three channels for control of the rudder, elevator and throttle. An extra channel is required to switch control from manual to stabilisation mode.

After considering a number of available remote controls, the Spektrum DX6i was chosen. This is a 6 channel remote control system, allowing for control of a four-axis aircraft with a control hand over switch and one auxiliary channel. As the EasyStar glider only requires three channels for manual control, any input from the aileron stick can simply be ignored.

A further advantage of the chosen remote control is that receivers can be bound to a specific transmitter, meaning that in situations where a number of aircraft may be in operation, remote control signals should not interfere with each other.

Figure 4.3: Spektrum DX6i remote control transmitter and receiver set

Batteries

Having chosen an electric motor as the propulsion system for the UAV, a battery or batteries must also be selected. The most common types of battery for remote control aircraft are Nickle Metal Hydride (NiMH), Nickle Cadmium (NiCd) and Lithium Polymer (LiPo). Of the three, Lithium Polymer batteries provide the greatest energy density and as such were chosen for use in this UAV. The actual model of battery chosen was a Turnigy Power Systems 3 cell 2200 mAh High Discharge Li-Po Battery. This battery is of the appropriate voltage for the Multiplex motor and at 2200 mAh should allow for reasonable flight times.

Figure 4.4: Turnigy Power Systems 3 cell 2200 mAh High Discharge Li-Po Battery

Servo Motors

The selection of servo motors was primarily dictated by the available space in the EasyStar glider. Two HiTec HS-81 servo motors were selected. These motors weight 16 g, which is the size of the general class of servo motors that the EasyStar glider was intended to be fitted with. The pulse width required to control these servos is 0.6 mS to 2.4 mS, significantly wider than the nominal 1 mS to 2 mS standard. Although a device called a servo stretcher can be fitted to allow these servos to interface with standard control signals, the customised nature of the control system means that this can be allowed for in software instead.

4.5 Development Platform

At the outset of the project the MikroElektronika EasyPIC5 development platform was made available by USQ for the purposes of this project. This development platform is made up of two components. Firstly, the EasyPIC5, which is the hardware component of the development system. The EasyPIC5 board contains all of the supporting hardware required to run the PIC Processor itself. This includes the power supply and oscillator. The development board also has a number of extra features useful for the testing and development of PIC software. This includes an in-circuit programmer and debugger, LED's for displaying port outputs, switches for generating inputs, potentiometers connected to analogue to digital converter pins, alpha-numeric and graphical LCD displays and expansion headers for connecting other hardware. A range of add on modules are also available to expand the already extensive capabilities of the development system. This wide range of functionality makes the EasyPIC5 board a good choice for development of the UAV system.

The second part of the development system is the MikroElektronica MikroC Compiler and Integrated Development Environment (IDE). The MikroC compiler comes with

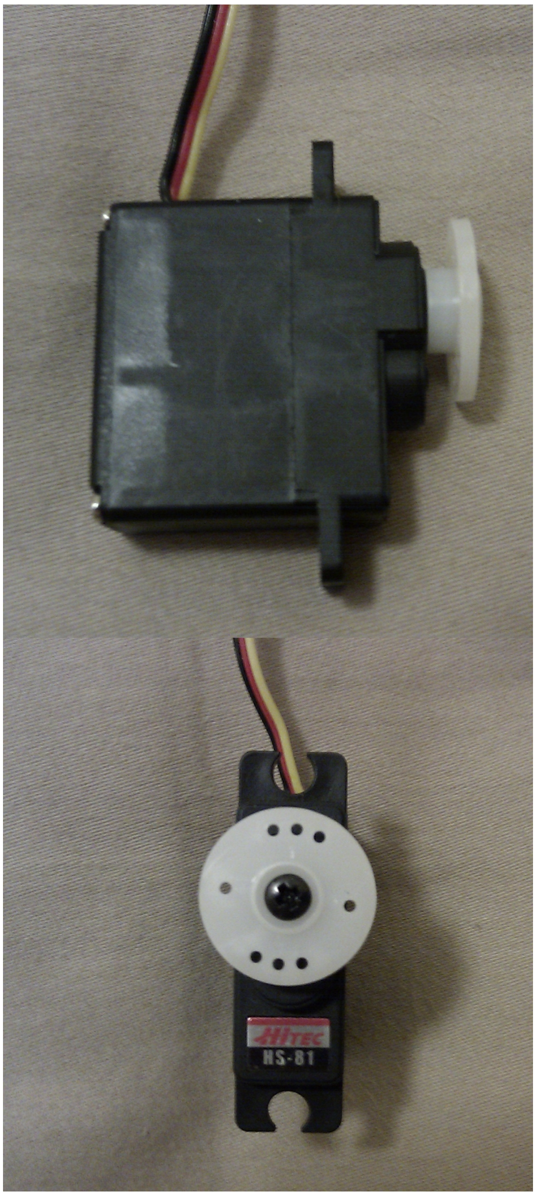


Figure 4.5: HiTec HS-81 servo motor

a wide array of library functions to simplify the development of PIC software. For example, the command Delay_ms() can generate the required delay in software without having to implement more complex timing loops. This makes the software easier to write, debug and read and will contribute significantly to the success of the project. The MikroE IDE is the final piece in the puzzle, providing an environment to develop, debug, install and test any software that is developed.

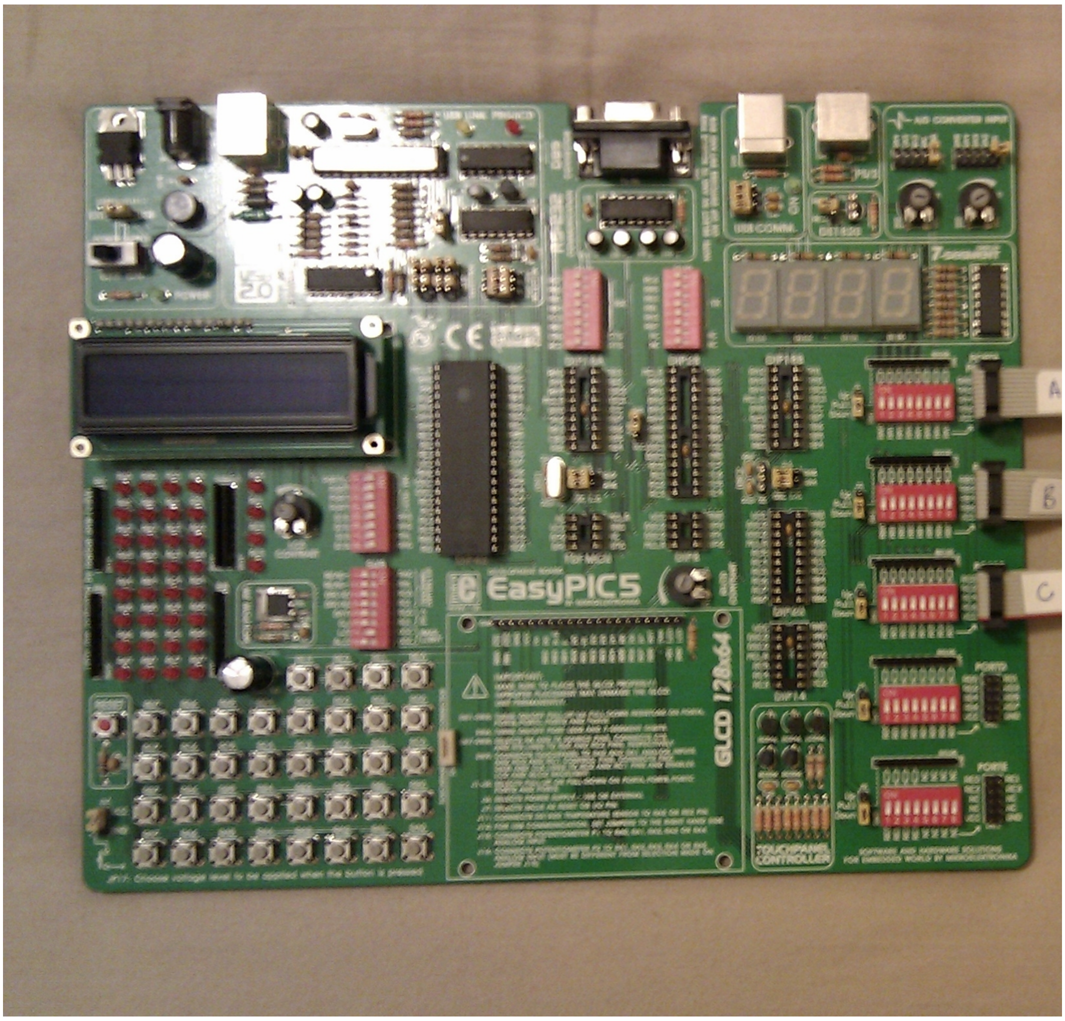


Figure 4.6: MikroElektronika EasyPIC5 Development Board

4.6 Microcontroller Selection

Having already decided to go with the EasyPIC5 development board the choice of processor was limited to a Microchip PIC. The EasyPIC5 board can accommodate PIC processors ranging from a small eight pin PIC to a large 40 pin one. In order to choose the processor, the requirements of the objectives for this project as well as the potential expansion required in future development was considered. Due to the extensive range of PIC processors available from Microchip a comparison of the many chips will not be presented. Rather, the features of the chosen chip will be presented as they relate to the UAV project. The microcontroller used for this project is the PIC16F887, a 40-pin

PIC processor.

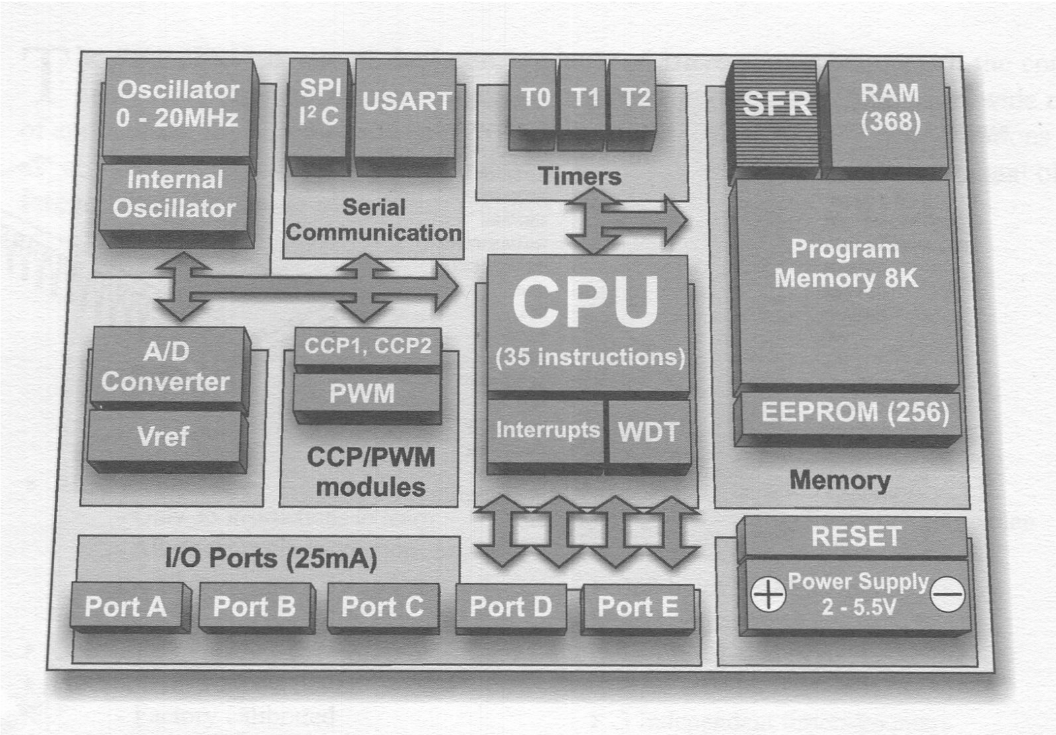


Figure 4.7: PIC16F887 internal block diagram (Verle 2008)

Some of the key features of the PIC16F887 processor are:

- **Clock Speed** The maximum clock speed of the processor is 20 MHz, towards the upper end of processors in this range. This will allow for fast computation of the control loops ensuring that servo signals can be updated within 20 mS.
- **IO Pins** 35 Input/Output pins are available on the chip (although some of these pins serve multiple purposes), allowing for plenty of expansion of the UAV hardware. This will be particularly useful during further development of the UAV when the integration of electronic payloads may be necessary.
- **Analogue Inputs** Analogue inputs are required for most of the UAV sensors. The PIC16F887 has 14, 10-bit analogue inputs, allowing for a wide range of sensors to be attached.

- **Timer Modules** The availability of timers is crucial to the decoding and encoding of PPM/PWM signals as well as to the operation of the control loop.

- **Serial Communications** Serial communications is not likely to be required for this phase of the project but will allow for communication between the processor and external modules via the I^2C or SPI bus.

- **In-Circuit Serial Programming** The ability to program the processor in-circuit means that modifications can be made to the software throughout the development and testing process with ease.

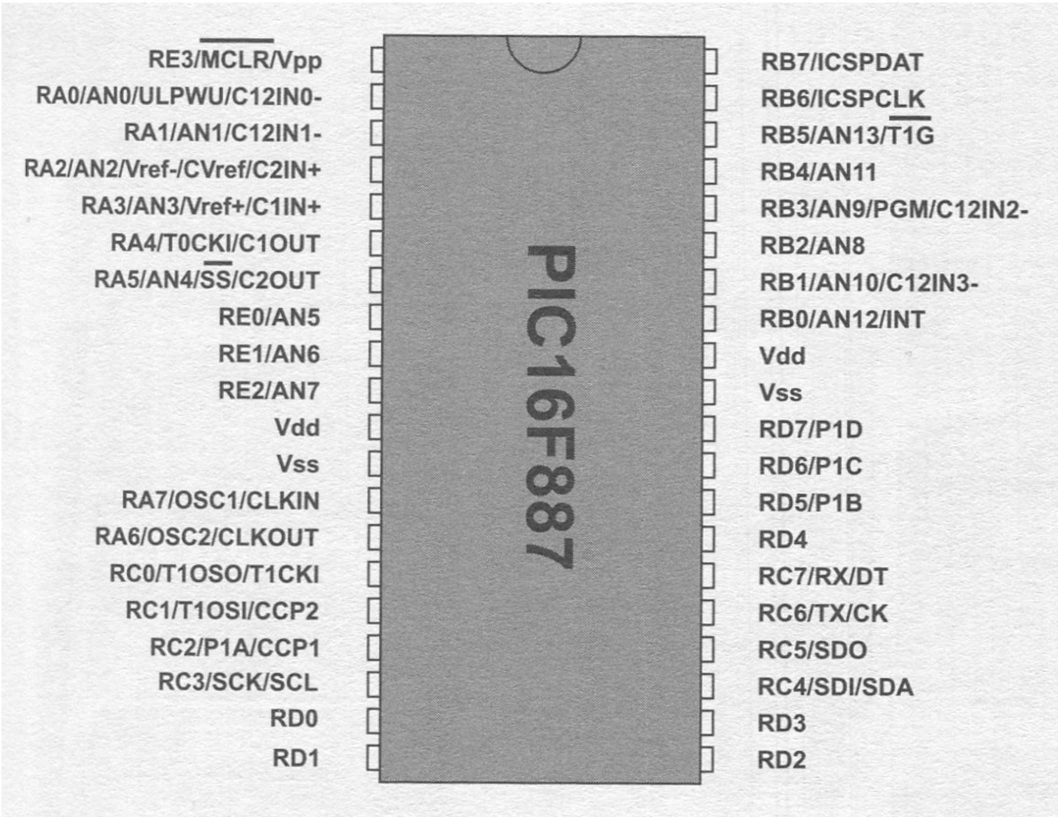


Figure 4.8: PIC16F887 pin out diagram (Verle 2008)

4.7 Conclusion

This chapter has outlined the approach to the design that will be used for the remainder of the project. An outline of the structure of the UAV has also been developed, detailing the key components both in the parent vehicle and control system. Finally, the details of the selection process for the parent vehicle and related components, development platform and microprocessor have been examined to form a solid base for the detailed design of the UAV.

	Weight	Multiplex EasyStar	Phoenix Boomerang 60	Phoenix Classic EP
Stability	10	4	3	3
Propulsion	8	4	2	2
Gliding	4	3	2	2
Payload	4	2	4	2
Ease of control	6	4	3	3
Ease of modification	8	3	2	3
Durability	6	4	2	4
Cost	0	< 1000	< 1000	< 1000
Weighted score		164	116	128

Table 4.1: Selection process weighted scoring table

Chapter 5

Detailed Design

5.1 Introduction

This chapter will outline the detailed design of the UAV project. As the parent vehicle has been constructed using purely off-the-shelf components, consideration is only given to the control system. Firstly the hardware design is examined, looking primarily at the methods of interfacing with key subsystems and the approach to handling the switch of control from manual to automatic mode. Secondly the design of the control system software is considered, looking at the key routines within the software and how these are integrated to form both the stabilisation mode and fly-by-wire mode software.

5.2 Control System Hardware

The hardware design for the UAV is based around a PIC16F887 microcontroller. A block diagram showing the layout of the system is presented in figure 5.1. In this system, the PIC processor performs all processing and control system functions and is supported by a number of subsystems. Each of these elements will be considered in detail below. Schematics of the entire system are included for reference in appendix B.

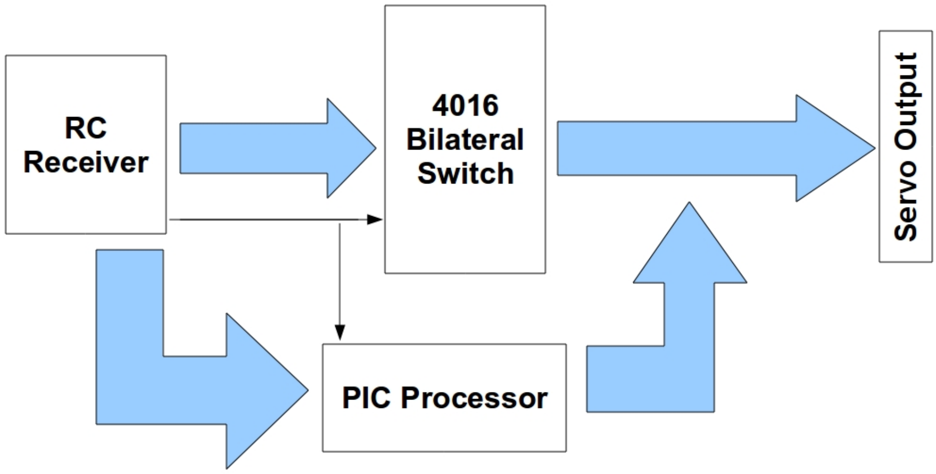


Figure 5.1: Hardware system block diagram

5.2.1 Sensor interface

Rather than constructing all of the support circuitry required to interface the thermopile sensors, pre-assembled sensor boards made by AttoPilot (figure 5.2) were purchased. The first of these boards contains four infra-red thermopile sensors and is used to detect the UAV's attitude in the pitch and roll axes. The second sensor board has only two thermopile sensors and is used primarily to calibrate the system. The boards have an differential amplifier for each pair of thermopile sensors, giving a differential output signal centred about 2.5V. This output signal undergoes some filtering through a resistor-capacitor network on the sensor board before being delivered to the analogue inputs of the PIC processor. The output of the X-Y sensor is connected to pins RA0 and RA1, and the output of the Z sensor to RA2.

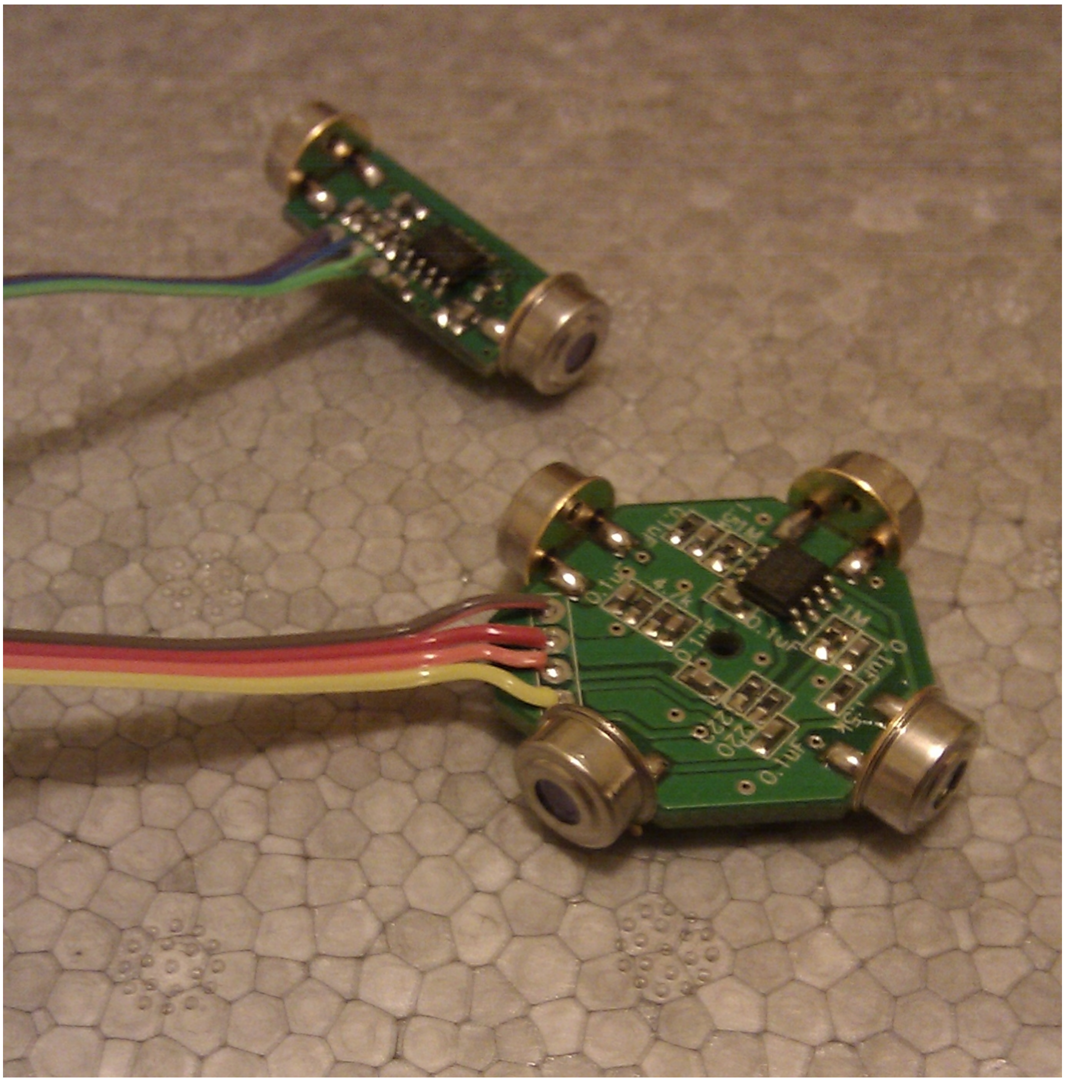


Figure 5.2: Infra-red thermopile sensor boards

5.2.2 Control input

All remote control signals are received by the Spektrum receiver module. These signals are pulse width modulated waveforms designed to control the servos directly and as such do not require any supporting circuitry. The servo control signals are connected to port B of the microcontroller as follows: Aileron to RB0, elevator to RB1 and control handover to RB2.

5.2.3 Failsafe control handover

The failsafe control handover sub-system is centred around a 4016 CMOS bi-lateral switch. During manual operation the switches in the 4016 IC are turned on by the handover signal from the remote control. In this state, the servo control signals are passed directly through the 4016 to the servos themselves. Once the handover signal changes the switches are all turned off, stopping any signals reaching the servos directly from the receiver.

As well as switching the 4016 IC, the handover signal alerts the PIC processor that it is now in control of the aircraft and it begins processing sensor information and sending the calculated control signals to the servos. This signal arrives at the PIC via the pin RB2.

Importantly, the PIC processor is not controlling the handover process itself so control can be regained in the event of a processor brown out or other fault causing the processor to fail.

5.2.4 Attitude controller

The attitude controller itself is based around the PIC16F887 microprocessor. For the development of the control system, the EasyPIC5 development board from Mikroelektronika has been used, containing all supporting hardware required for the operation of the PIC. Other hardware was built on a veroboard and connected to the EasyPIC development system via ribbon cables, as shown in figure 5.3.

Future development will see these two boards become one custom designed printed circuit board in order to save space.

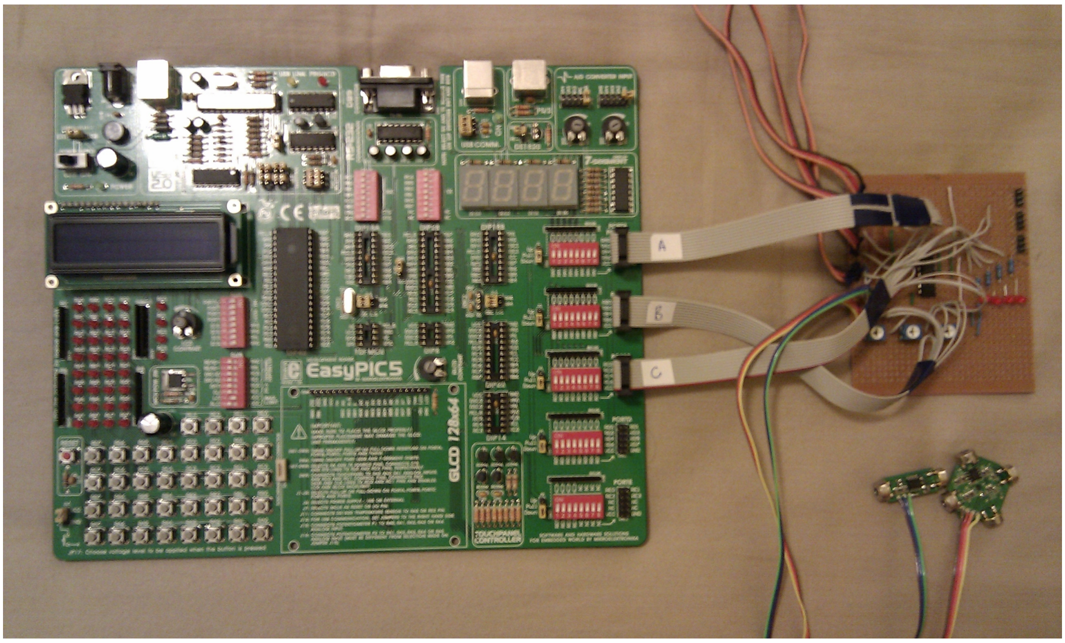


Figure 5.3: EasyPIC5 development board and UAV hardware prototype

5.2.5 Microcontroller Support Hardware

The support hardware required for the microcontroller is actually located on the EasyPIC5 board. This includes the power supply, reset circuit and oscillator. These components have been included on the schematic as they will ultimately have to be replicated in hardware once the control system is implemented as a Printed Circuit Board (PCB). The design for these components has been adapted from the PIC data sheet to suit the requirements of the UAV.

5.3 Control System Software

Consideration of the available literature presented two main options for the development of the control systems. The most common approach is illustrated by Iscold et al. (2010). That is, a control system is designed 'on paper' and implemented in software. Next an aerodynamic model of the aircraft is developed and combined with a six degree of freedom model, describing the rigid body dynamics of the aircraft in terms of aerodynamic conditions. These models are combined with an atmospheric model such that the flight of the aircraft can then be simulated. Using the simulation the PID gains of the aircraft's control loops can be tuned prior to flight.

The other method of control system development is characterised by the work in Egan et al. (2004). In this method, the control system is designed to give separate control loops in the pitch and roll flight axes. Instead of creating a model for simulation of the aircraft, the control system is implemented in hardware and installed in the aircraft. Tuning of the control loops then occurs in flight. Egan et al. (2004) used the tried and tested Ziegler-Nichols tuning method, implementing this as an in-flight automated tuning function.

On consideration of the two approaches, the latter is favoured for a few reasons. Firstly, the significant overhead required to measure the aerodynamic coefficients used in the aerodynamic model of the aircraft puts this approach out of the reach of this project. Also if this work was to be completed, all results would be specific to the current parent vehicle and any significant changes to the airframe or the installation of the control system in a new aircraft would require all of this work to be re-done. Finally, the flight equations used in the aerodynamic model are linearised and only hold true for small perturbations around the steady state. As such the accuracy (or inaccuracy) of the model means that tuning the PID gains in the simulation is not likely to yield results good enough to warrant the effort.

5.3.1 Control system design

The control system is shown in figure 5.4 and 5.5 in block diagram format. Note that the pitch and roll axes are not coupled and operate independently of each other.

Two distinct software programs have been developed for the UAV project. The first

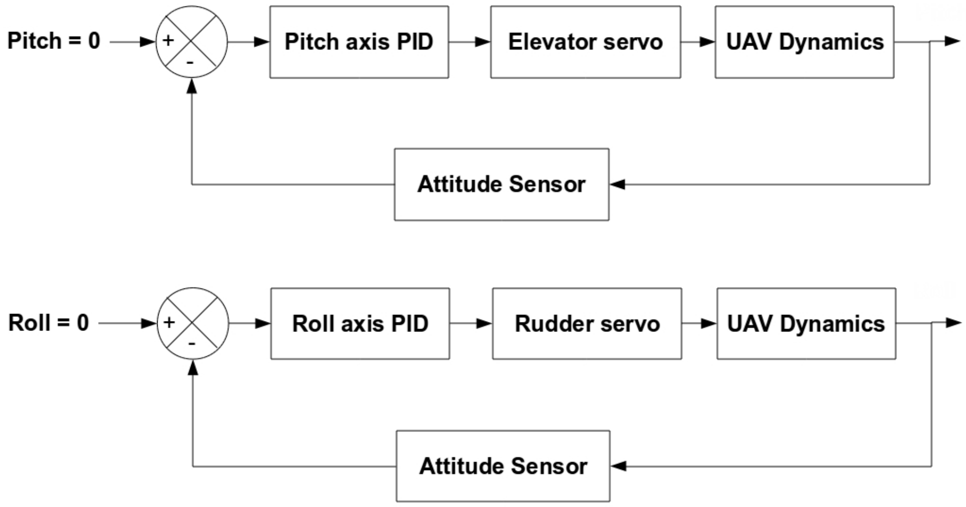


Figure 5.4: Control system block diagram (stabilisation mode)

is for the UAV operating in stabilisation mode. When enabled, the flight controller reads the attitudinal information from the infra-red thermopile sensors and calculates the required servo outputs based on a target attitude of 0° (level in both the pitch and the roll axes).

The second software program operates as a co-pilot or fly-by-wire interface. Operation of the software is the same as the stabilisation mode but the pilot is able to set the target attitude of the aircraft using the remote control. The distinction between this flight mode and normal manual operation is important. Under manual control the remote is used to set pitch and roll rates. For example, moving the elevator joystick to a fully down position would result in the aircraft attempting loop-the-loop manoeuvre. In the co-pilot mode the elevator joystick sets the absolute pitch angle. In this case, moving the elevator joystick to a fully down position would put the aircraft into a constant climb at a maximum pitch angle set in the control system.

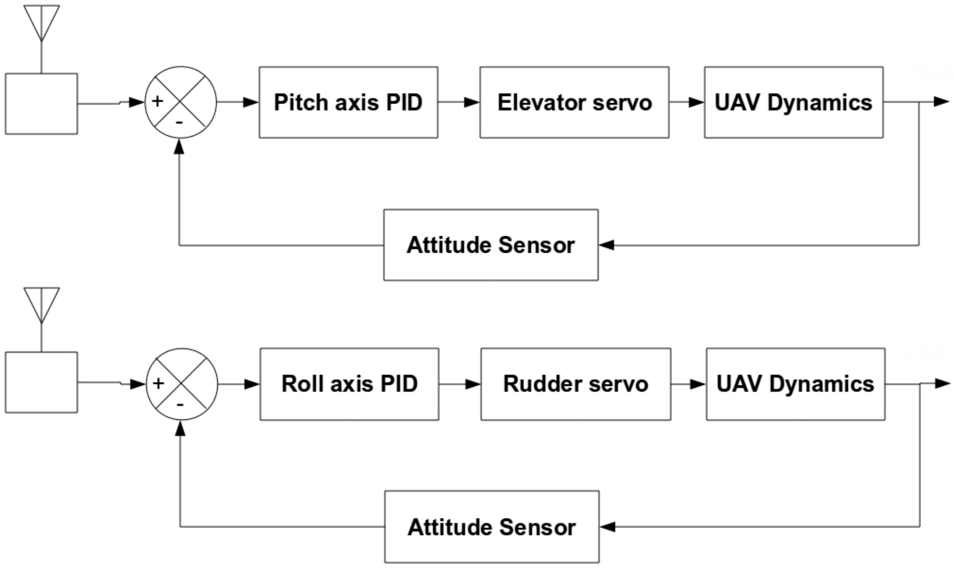


Figure 5.5: Control system block diagram (fly-by-wire mode)

For simplicity, roll angles are set by the aileron axis on the remote control rather than the rudder axis, with pitch set by the elevator axis. The reason for this is that aileron and elevator control both use the same joystick, so full control of the aircraft can be achieved using only one joystick. This greatly simplifies flying the aircraft and means that very little training will be required to control the UAV.

5.3.2 Attitude detection

The first step in the attitude detection process is to calibrate the system. The aircraft must be placed on the ground and the z-axis sensor board must have a clear view of the sky and the ground. The control system is then turned on at which point the processor will read the output of the z-axis sensor and store this value as the 'maximum difference'. In stabilisation mode, if the output of the pitch or bank signals reaches this value, the aircraft has moved out of the field of vision of the sensors and the control system will refer to the z-axis sensor to determine the attitude. In co-pilot mode the joystick can be used to demand a difference of up to this maximum value.

In practice the demanded pitch and roll angles will be limited to a value less than this maximum. For example, if the field of vision of each thermopile is 100° it is able to measure pitch and roll angles of ±50°. For this reason the maximum demanded pitch and roll angles should be limited to some value less than this. Although the z-axis sensor could be used to fly outside of these limits a more conservative approach is considered to be favourable, particularly during testing.

In the software the calibration function is performed by the *calibrate()* routine. This reads the value seen by RA2 and stores it as *max_error*. At this stage calibration is only undertaken at the start of each flight but in-flight calibration can be included at a later stage if the duration of the UAV's flight is extended such that it will fly over an extended part of the day.

5.3.3 Control decoding

Control signals from the remote control receiver arrive at the PIC processor in the form of Pulse Width Modulated (PWM) waveforms. A pulse width of 1.5ms corresponds to the neutral or 90° position of the servo i.e., no control surface deflection. Decreasing the pulse width to 0.6ms will move the servo to the 0° or minimum position, and increasing the pulse width to 2.4ms will move the servo to the 180° or maximum position.

Although the PIC16F887 does have dedicated CCP or capture mode to measure PWM inputs, there are not enough CCP enabled inputs to read all of the required information. To overcome this, the process is completed manually. When a rising edge is received on a port B input pin, an interrupt is generated. At this point, the interrupt routine will reset TMR0. At the next falling edge, the value of the timer is read and saved in the appropriate variable. Due to modulation scheme used in remote control aircraft, only one servo pulse will arrive at the processor at a time and so only one timer is required. This greatly simplifies the process of decoding servo signals.

The two variables that store the read control inputs are *aileron_width* and *elevator_width*. Remember that although the aircraft does not have ailerons, the aileron channel is used as it shares the elevator joystick on the remote control. This greatly simplifies the task of the UAV operator.

5.3.4 Stabilisation controller

The stabilisation controller is simply the software implementation of the control loop described in section 5.3.1. The differential signals as measured by the ADC module are used as the error input to the control loop. This error value is offset by the aileron and elevator inputs in the fly-by-wire mode to achieve co-pilot control.

The gains for the PID control routine are set in the software before loading onto the aircraft. Proportional, integral and derivative gains must be set separately for the pitch and roll axes as the control loops are not coupled.

The mathematics used in the stabilisation controller are derived from the digital PID implementation presented in Charais & Lourens (2004). That is:

$$C(n) = K_p E(n) + K_i T_s \sum_{0}^{n} E(n) + K_d \frac{E(n) - E(n-1)}{T_s} \tag{5.1}$$

5.3.5 Servo control output

The servo control output section of the software is triggered by the completion of the control loop calculations. This ensures that servo signals are generated approximately every 20 mS as required. The *Delay_us* routine provided by the MikroC libraries is not able to handle variable delay times so an alternative approach had to be found. *Vdelay_ms* can accept variable delays but only as whole numbers of milliseconds so was unsuitable. Instead the *Delay_Cyc* command was used. This function accepts a value of 'tens of CPU cycles' to determine the delay. For this reason the *servo_out* function converts the delay in microseconds to tens of clock cycles. The appropriate output pin is set to 1, the delay function performed and then the output pin reset to 0. This generates the required pulse width for control of the servo motor.

5.4 Conclusion

This chapter has looked in detail at the design of the UAV control system. Important features and those requiring some though as to implementation have been presented here, but a complete schematic is available in appendix B and code listing in appendix C. At this point the layout of the UAV system, detailed hardware design and the implementation of stabilisation and fly-by-wire software has been completed. The next chapter will address the testing and evaluation of the UAV system and suggest remedies and improvements where necessary.

Chapter 6

Evaluation and Testing

6.1 Introduction

This chapter will outline the process of testing the UAV. Discussion about the appropriateness of the design as well as simply the operation of the UAV will be presented and improvements suggested where required. These suggestions will be considered for further work in the following chapter.

6.2 Attitude measurement

During initial tests of the attitude measurement system, some problems were encountered measuring the pitch of the aircraft. This was due to the mounting arrangement used for the thermopile sensors. In the current arrangement the rear pitch axis sensor points directly at the electric motor. As such the temperature seen by this sensor when the motor is running is elevated, falsely indicating to the control system that this sensor is pointed more towards the ground than the front pitch axis sensor.

To alleviate this problem, the sensor has been rotated 45° so that it is not pointing directly at the motor. This does however require some modification to the attitude calculation process. Previously a downwards pitch of the aircraft would result in a decrease in the output of only one axis of the sensor board. In the 45° orientation, both axes will exhibit an equal, lower value. Similarly, whereas a roll putting the left-hand wing of the aircraft towards the ground only decreased the value of one axis using the previous configuration, the new orientation will result in the two axes changing by the same amount but in opposite directions. By re-writing the code that calculates the attitude of the aircraft to incorporate this new mixing effect, the problem of heat generated by the motor can be resolved.

6.3 Control input and output

To verify the servo control signal encoding/decoding routines, a modified version of the control system was loaded onto the PIC microcontroller. This modified program simply read and decoded the servo signals, re-encoded the signals and output them to the servo motors. Using this testing software, the aircraft would operate as if it were under direct remote control. For example, a full down on the elevator joystick would be decoded and re-encoded as the same value. This process uncovered some minor bugs in the software program but once the appropriate changes were made, the servos moved through their full range of operation when commanded by the remote control.

One potential problem with the control input and output process is the time required to encode and decode signals. Ultimately the control system will have complete control over the throttle as well as the rudder and elevator axes. This would require the control system to read two inputs and supply three outputs. The total time for handling signals in this configuration would be nominally 5×2.4 mS $= 12$ mS. This is a significant portion of the 20 mS refresh time of the servos. The addition of navigation and other functions could push the microcontroller close to this 20 mS limit resulting in less that optimal control of the UAV. To improve this, a second processor could be added purely to handle the servo signal processing functions. This second processor would communicate with the main control processor via SPI or I^2C bus. This approach has been used successfully by Egan et al. (2004), who noted that this effectively introduced a modular element to the design, increasing the flexibility and reliability of the control system.

6.4 Failsafe control handover

Initial discussions with the remote control supplier indicated that the remote control receiver provided two digital output channels and that these would be suitable for switching the control handover and other functions directly. In fact, on receiving the remote control system, closer inspection revealed that the auxiliary channels were pulse width modulated in the same manner as the control surface channels. In order to use these axillary channels for the control handover process all control signal decoding must occur outside of the main PIC processor. The two options were either to include a secondary processor for the distinct purpose of controlling the handover process or to continue with the existing design and introduce some form of hardware decoding for the axillary channel only. The addition of a secondary processor for control decoding has already been suggested as an important improvement on the current design. Adding the control handover function to this processor would be trivial and is considered the best approach at this stage.

To continue testing of the control handover process, a manual switch was added to replace the remote control signal. On switching control from the remote control to the PIC processor, servo signals from the remote control were successfully turned of by the 4016 bi-lateral switch. The processor also took control of the aircraft once the control signal changed.

Although not encountered during testing it is conceivable that the PIC processor could fail to stop generating servo control signals, most likely as the result of a software bug. In the current design, switching control back to the remote control does not prevent the PIC processor's outputs from reaching the servos. Clearly a more robust switching design is required.

Such an arrangement could be achieved using a data selector or multiplexer. For example, consider the 74LS157 quadruple 2-line to 1-line multiplexing IC. Rather than simply blocking remote control signals from reaching the servos, the servo control lines are controlled by a decisive switching process. In the event that the PIC processor fails to stop generating control signals, re-engaging manual control can now be effected successfully. An illustration of this revised process is included in figure 6.1.

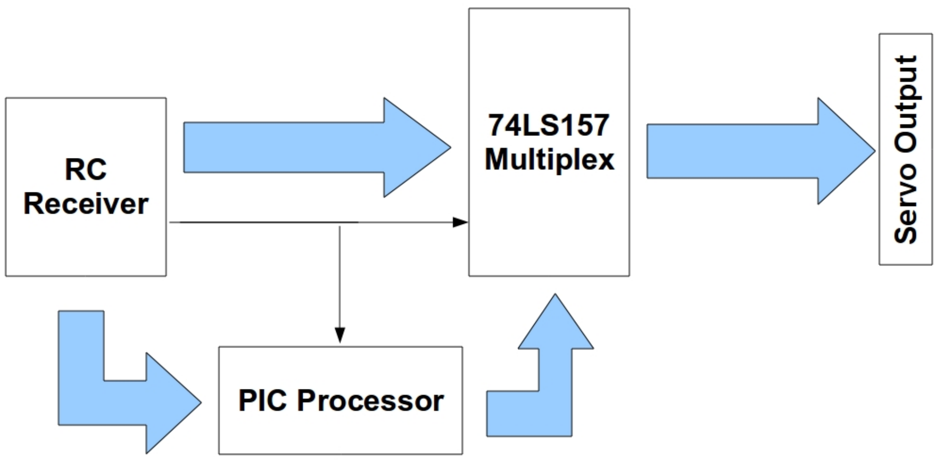


Figure 6.1: Revised control handover process using the 74LS157 multiplexer

One flaw in the failsafe design is that it does not have any automated error detection process. If the PIC processor were to fail for any reason, manual intervention is required from the pilot to return the UAV to manual control. While this may be an acceptable process during the testing phase when the UAV is in sight, the intention is that the UAV would be able to fly beyond the line of sight of the operator. In such a situation, there is no feedback mechanism to detect or recover from a fault.

The solution to this problem must consider two aspects. The first is some sort of on board error detection and the second is supplying feedback to the operator. The addition of on board error detection implies that a second 'watchdog' processor is required. This adds a level of redundancy required to maintain safe operation of the UAV. Anderson (2009) describes such an approach applied to a similar UAV control system. The larger processor, roughly equivalent to the PIC used in this project, handles the control loops as well as navigation and a range of other functions. The smaller processor controls only the handover process and control signal decoding. It is also able to detect the absence of this control signal, such as would occur if the aeroplane flew out of range of the remote control. After a specified time out period it may do a number of things including returning the UAV to the launch site, loiter about its current position or engage a parachute or crash landing configuration. The smaller processor may also be used to monitor the operation of the larger processor itself. If the larger processor fails to generate a 'still running' signal the smaller processor can be used to reboot it in mid air. Multiplexing of the servo control signals would occur in the same manner as it does currently. A block diagram illustration of this design is shown in figure 6.2. This approach adds further weight to the view that a second processor should be added to the control system. The implementation of this improved solution will be considered in chapter 7 as a possible future development.

6.5 Stabilisation control

Because of the size of the electronics and time limitations, in-flight testing of the stabilisation control processes was not possible. The operation of the control loops was able to be validated however. By setting all PID gains to zero and then increasing

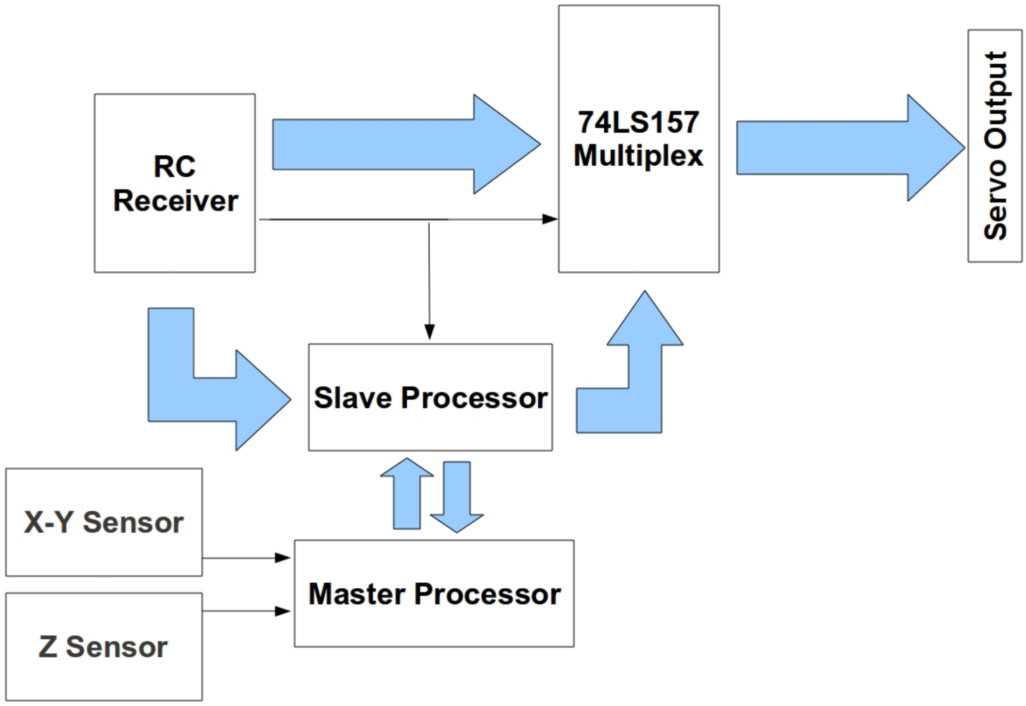


Figure 6.2: Introduction of a secondary processor to the control system

each gain one at a time, the effect on the control surfaces was able to be observed and hence a qualitative assessment of the systems operation could be made. Three potentiometers were included in the prototype so that these gains could be changed without recompiling the software with new gain values each time.

The performance of the lateral control loop was assessed first. With all gains set to zero, no control surfaces deflected. Increasing the proportional gain resulted in some deflection of the rudder when the aircraft was rolled. Rudder deflection increased as the roll angle increased, and also increased as the gain increased. This would indicate that the proportional element of the control loop is functioning as expected.

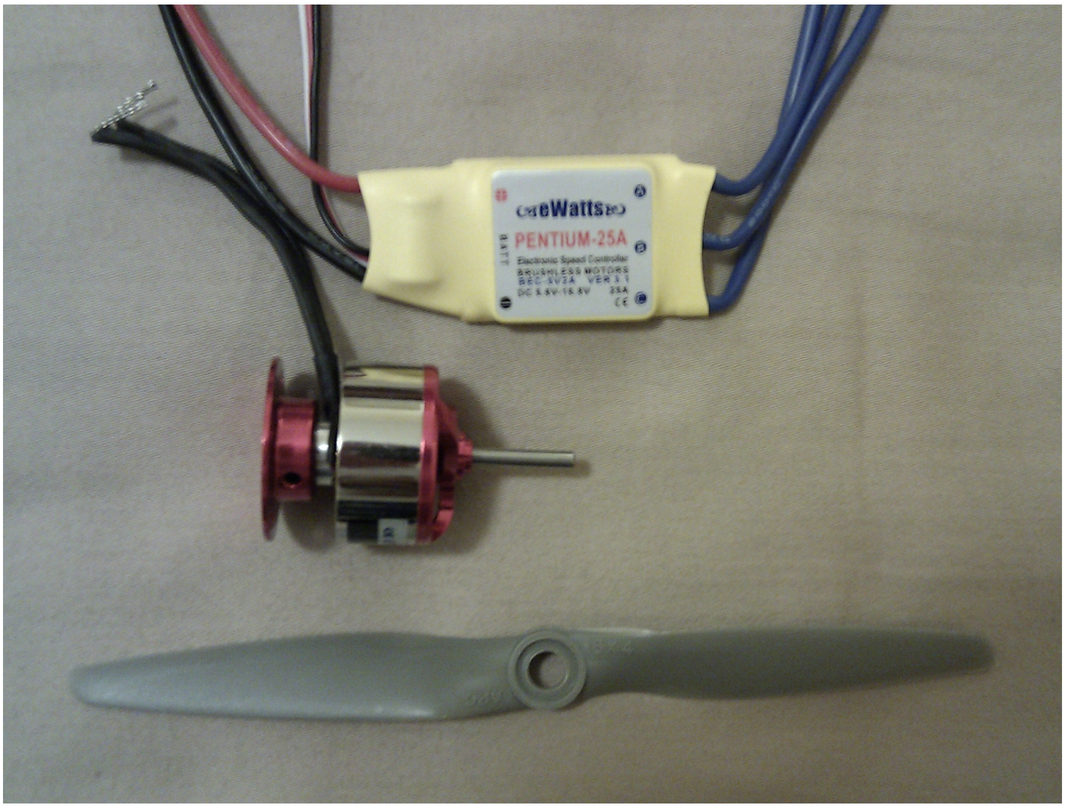
To test the derivative element of the control loop, all gains were again set to zero and the D term slowly increased. Rolling the aircraft slowly resulted in a minimal change in the rudder, whereas a fast roll gave a larger change in the rudder. This indicated that the derivative element of the control loop was also functioning correctly.

Finally the integral element was tested in the same manner as the two previous examples. Holding the aircraft in a non-zero position resulted in a steady change in the rudder deflection. This rate of change increased as the integral gain term was increased.

After re-assigning the potentiometers to control the longitudinal control loop gains, the control system response to the pitch of the aircraft was also tested. After some minor changes the longitudinal loop performed in the same manner.

The final step in this process, after rectifying problems with the control handover system is to begin flight tests and tune the control system. This flight testing will initially focus on tuning the flight controller. Consideration must also be given to placing some limits on the PID terms. This will be important in preventing 'integral windup', where the integral term becomes so large it dwarfs the P and D terms and renders the control system useless. These limits will have to be determined experimentally however.

There are some important points to consider for further development of the UAV control system. At this point, the control system has been designed primarily as a stabilisation controller. That is, it is able to stabilise the aircraft about the steady state condition. Control of the thrust is still performed manually by the operator. A vast improvement would be to install a pitot tube to measure airspeed. This device would be used firstly to scale the gain terms used in the PID loop to provide appropriate levels of feedback at all speeds. Secondly, it would be able to detect a stall, where the aircraft loses airspeed to a point where no lift is being generated.



In its current state the control system would attempt to correct this by tilting the nose of the aircraft upwards, decreasing the airspeed further and exacerbating the problem. On detecting a stall condition the control system could tilt the nose of the aircraft downwards to gain the required airspeed before returning to a suitable cruising altitude. The addition of a barometric pressure sensor to determine the height of the aircraft would also be important so that the aircraft can maintain itself at a constant cruising altitude.

6.6 Other considerations

Discussions with USQ technical staff (Byrne, T 2010; Richards, B 2010, pers. comm., 2 June) indicated that their previous efforts in developing a UAV found that noise from the aircraft's electric motor could create significant reliability issues with regard to the PIC processor. Two measures were implemented in order to deal with this. Firstly it was decided to power the UAV's avionics from a separate power source. The main motor and servo's would continue to operate using the Lithium Polymer battery and the control circuitry from a standard 9V battery. While this is sufficient for testing purposes, some investigation is still required to quantify the power requirements of the control board and whether this arrangement can provide for adequate flight times. To further reduce noise a brushless motor was selected to replace the standard motor supplied with the Multiplex kit. As well as reducing the noise that may be experienced this is expected to provide an improved efficiency.

The new brushless motor for the aircraft is the eWatts R2212 model. Moving to this motor also meant that a new motor controller, the eWatts Pentium-25A Electronic Speed Controller was required. To capitalise on the improved performance of the motor, a new 6 x 4 composite propeller was also purchased.

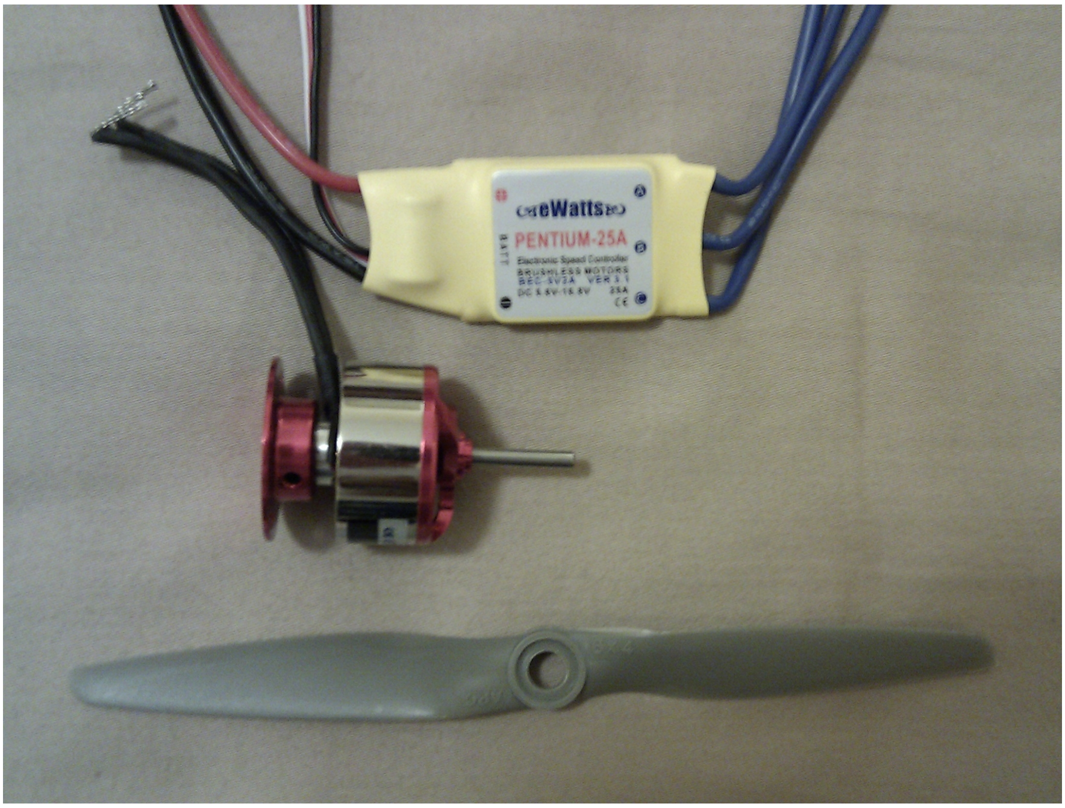


Figure 6.3: New brushless motor, speed controller and propeller

6.7 Conclusion

Although the correct operation of the control system has been verified in bench top testing, full flight tests have not been possible due to control handover problem. The solution to this, the addition of a second microcontroller, has been introduced and will be included as future work for the project. This new option also allows for a vast improvement in the safety of the UAV by allowing for in-flight error detection and intelligent error handling. In this scenario, both processors would check each other to determine their correct operation and could reboot the other if needed. The second processor could also be powered from a separate battery as described in UAV Outback Challenge (2010). This level of redundancy allows for much safer operation of the UAV, one of the prime considerations for this project.

Chapter 7

Conclusion

7.1 Introduction

This chapter serves a number of functions. Firstly it seeks to summarise further work required to complete this stage of the project. This further work is primarily the rectification of issues found in the preceding chapter as well as important improvements that should be made. Secondly, it will examine the current state of the project in the context of the longer term vision for the project. This may provide some inspiration for students wishing to continue the research carried out in this project. Finally, the original objectives of the project are revisited to ensure that the task set at the outset has been completed.

7.2 Recommendations for Further Work

The first piece of future work that must be undertaken is the addition of a second processor to the control system as described in chapter 6, as well as the implementation of the revised control handover process. This will result in vastly improved safety of the UAV and will rectify the problems encountered during testing.

At this point, extensive flight testing of the aircraft can begin and the aircraft's control system can be tuned. Consideration of the performance of the control system will inform the imposition of limits to prevent integral windup as has already been discussed.

Resolving these issues will pave the way for successful future development of the aircraft.

7.3 Future Directions

At this point the UAV is designed simply to stabilise flight in the lateral and longitudinal flight axes. The addition of static and dynamic pressure sensors to the aircraft would allow for control of the thrust axis and also the height of the aircraft. This would also be an important step in improving the operation of the control system to appropriately handle stall conditions and in tuning the control system to operate optimally at a range of air speeds.

Following this development, the addition of in-flight telemetry and recording would improve the ability to test the aircraft and facilitate future development.

The addition of GPS navigation is also a crucial part of many UAV systems and would be a significant task in itself. This would lead to the development of computer software to assist in the mission planning process to make the UAV easier to use. Also, the addition of automated take-off and landing would be a big achievement towards the ultimate goal of a fully autonomous system.

The other major component in any UAV system is the payload that the UAV will carry. In terms of the original vision for the project, a UAV for use during disaster situations, some form of video or still photography module would be the most obvious next step. There is no doubt that the existence of a robust UAV platform will prompt ideas for payloads from all quarters and this will drive the further development of the UAV for years to come.

7.4 Achievement of Project Objectives

The overall goal of this project was to develop an unmanned aerial vehicle that could be used to obtain footage or photographs of areas affected by natural disasters. For the most part this objective has been achieved. The specific objectives supporting this goal and the level to which they were achieved is as follows:

- **Research legislation and standards relating to the use of Unmanned Aerial Vehicles (UAV's) in Australia** The legislation and standards related to UAV's has been considered. A summary of the key considerations from the Civil Aviation Safety Regulations has been presented as they relate to this project. Finally, consideration of the standards for operation from two different organisations was used to inform the development of the safety systems of the UAV and provide guidance for future flight testing.

- **Research and evaluate existing methods of UAV control** Existing methods of UAV control were researched and evaluated, informing the development of the layout of the UAV system and ultimately the detailed design of the UAV.

- **Select an existing kit (fixed wing or rotary) for use as a parent vehicle** An existing fixed wing kit was selected for use as the parent vehicle as well as all of the additional components required to be installed on the aircraft. This included the remote control system, servo motors, electric motor and speed controller, propeller and batteries.

- **Design an electronic control system (hardware and software) capable of maintaining stable, horizontal flight and verify its operation external to**

the parent vehicle An electronic control system was designed and built and its operation external to the parent vehicle verified. Some problems were encountered with the initial design, with solutions to these presented for consideration as future work.

- **Adapt the electronic control system for use in the chosen parent vehicle and verify its operation** Due to the problems highlighted in the testing phase, this final step was not able to be completed. Once the problems identified have been resolved, the control system can be implemented in PCB form and extensive flight testing conducted.

One key outcome of the project that was not identified as an initial objective was the development of a fly-by-wire or co-pilot control mode. In lieu of fully autonomous control, this provides for very simple operation of the aircraft as it only requires one joystick for operation. Future development to include GPS navigation will be able to expand this co-pilot software such that the navigation system, rather than the remote control, provides these guiding inputs. As such it is an important achievement towards the goal of fully autonomous control.

7.5 Conclusion

For the most part this project has developed a UAV that could be used to obtain footage or photographs of areas affected by natural disasters. Where deficiencies in the current approach have been identified, so have improvements to resolve them. This project is only the first step in the long and involved process of developing a fully autonomous UAV, but it is a first step none the less. It is hoped that the achievements of this project will be continued by further researchers and that a robust solution for use during disaster situations will ultimately be developed.

References

Abzug, M. (1998), *Computational Flight Dynamics*, American Institute of Aeronautics and Astronautics, Virginia.

Adiprawita, W., Ahmad, A. & Sembiring, J. (2007*a*), Development of AHRS (Attitude and Heading Reference System) for Autonomous UAV (Unmanned Aerial Vehicle), *in* 'International Conference of Electrical Engineering and Information, Bandung, Indonesia'.

Adiprawita, W., Ahmad, A. & Sembiring, J. (2007*b*), Hardware in the Loop Simulation for Simple Low Cost Autonomous UAV (Unmanned Aerial Vehicle) Autopilot System Research and Development, *in* 'International Conference on Electrical Engineering and Informatics (ICEEI2007)'.

Anderson, C. (2009), *ArduPilot main page*, DIY Drones, viewed 12 September 2010, <http://diydrones.com/profiles/blogs/ardupilot-main-page>.

CASA (2003), *Civil Aviation Safety Regulations 1998 (CASR) Part 101*, 1st edn.

Charais, J. & Lourens, R. (2004), AN964: Software PID Control of an Inverted Pendulum Using the PIC16F684, Technical report, Microchip.

Cook, M. (1997), *Flight dynamics principles*, Arnold, London.

Egan, G., Cooper, R. & Taylor, B. (2004), Unmanned Aerial Vehicle Research at Monash University, *in* 'AIAC-11 Eleventh Australian International Aerospace Congress'.

Egan, G. K. & Cooper, R. J. (2006), An Unmanned Aerial Vehicle Autopilot, Technical report, Monash University.

Egan, G. K. & Taylor, B. (2006), The Use of Infrared Sensors for Absolute Attitude Determination of Unmanned Aerial Vehicles, Technical report, Monash University.

Egan, G. K. & Taylor, B. (2007), Characterisation of Infrared Sensors for Absolute Unmanned Aerial Vehicle Attitude Determination, Technical report, Monash University.

Endurance RC (2010), *Guide to PWM and PPM*, viewed 5 September 2010, `<http:/www.endurance-rc.com/ppmtut.html>`.

Iscold, P., Torres, G. & Leonardo, A. (2010), 'Development of a Hand-Launched Small UAV for Ground Reconnaissance', *IEEE Transactions on Aerospace and Electronic Systems* **46**(1), 335–348.

Johnson, E. & Fontaine, S. (2001), Use of flight simulation to complement flight testing of low-cost UAVs, *in* 'AIAA Modeling and Simulation Technologies Conference'.

Kurnaz, S., Kaynak, O. & Konakoğlu, E. (2007), 'Adaptive neuro-fuzzy inference system based autonomous flight control of unmanned air vehicles', *Advances in Neural Networks–ISNN 2007* pp. 14–21.

Mendelow, B., Muir, P., Boshielo, B. & Robertson, J. (2007), 'Development of e-Juba, a preliminary proof of concept unmanned aerial vehicle designed to facilitate the transportation of microbiological test samples from remote rural clinics to National Health Laboratory Service laboratories', *South African Medical Journal* **97**(11).

Microchip Technology (2009), *PIC16F882/883/884/886/887 Data Sheet*, U.S.A.

MikroElektronika (2008), *EasyPIC5 User Manual*, MikroElektronika, Beograd.

Nise, N. S. (2008), *Control Systems Engineering*, 5th edn, John Wiley & Sons, Inc., USA.

Puttige, V. & Anavatti, S. (2008), 'Real-time System Identification of Unmanned Aerial Vehicles: A Multi-Network Approach', *Journal of Computers* **3**(7), 31.

SilverStar (2006), *PID Feedback*, Wikimedia Commons, viewed 25 October 2010, <`http://upload.wikimedia.org/wikipedia/commons/4/40/Pid-feedback-nct-int-correct.png`>.

Taylor, B., Bil, C., Watkins, S. & Egan, G. (2003), Horizon sensing attitude stabilisation: A VMC autopilot, *in* '18th International UAV Systems Conference, Bristol, UK'.

Taylor, B. & Egan, G. K. (2006), Monash UAV Operations Manual, Technical report, Monash University.

UAV Outback Challenge (2010), *Search and Rescue Challenge Mission, rules and judging criteria*, viewed 5 September 2010, <`http://www.uavoutbackchallenge.com.au/2010/UAVChallengeRulesSearchAndRescue2010v1_3.pdf`>.

Verle, M. (2008), *PIC Microcontrollers With Examples in Assembly Language*, MikroElektronika, Beograd.

Wiebe, J. (2010), *We Be James & Belite Aircraft: Ultralight Aircraft & Avionics; FAR Part 103*, viewed 16 October 2010, <`http://jameswiebe.blogspot.com/2010/06/thorough-belite-superlite-review.html`>.

Appendix A

Project Specification

University of Southern Queensland

Faculty of Engineering and Surveying

ENG 4111/4112 Research Project
Project Specification

For: Bede Wilson

Topic: Development of an Unmanned Aerial Vehicle for Use During Disaster Situations

Supervisor: Mr Mark Phythian

Project Aim: This project seeks to develop an unmanned aerial vehicle that could be used to obtain footage or photographs of areas affected by natural disasters.

Programme (Issue A, 18 March 2010):

1. Research legislation and standards relating to the use of Unmanned Aerial Vehicles (UAV's) in Australia.

2. Research and evaluate existing methods of UAV control.

3. Select an existing kit (fixed wing or rotary) for use as a parent vehicle.

4. Design an electronic control system (hardware and software) capable of maintaining stable, horizontal flight and verify its operation external to the parent vehicle.

5. Adapt the electronic control system for use in the chosen parent vehicle and verify its operation.

As time permits:

6. Integrate GPS way-point navigation into the control system to allow for fully autonomous flight.

   OR

7. Adapt the control system to allow fully autonomous take off and landing.

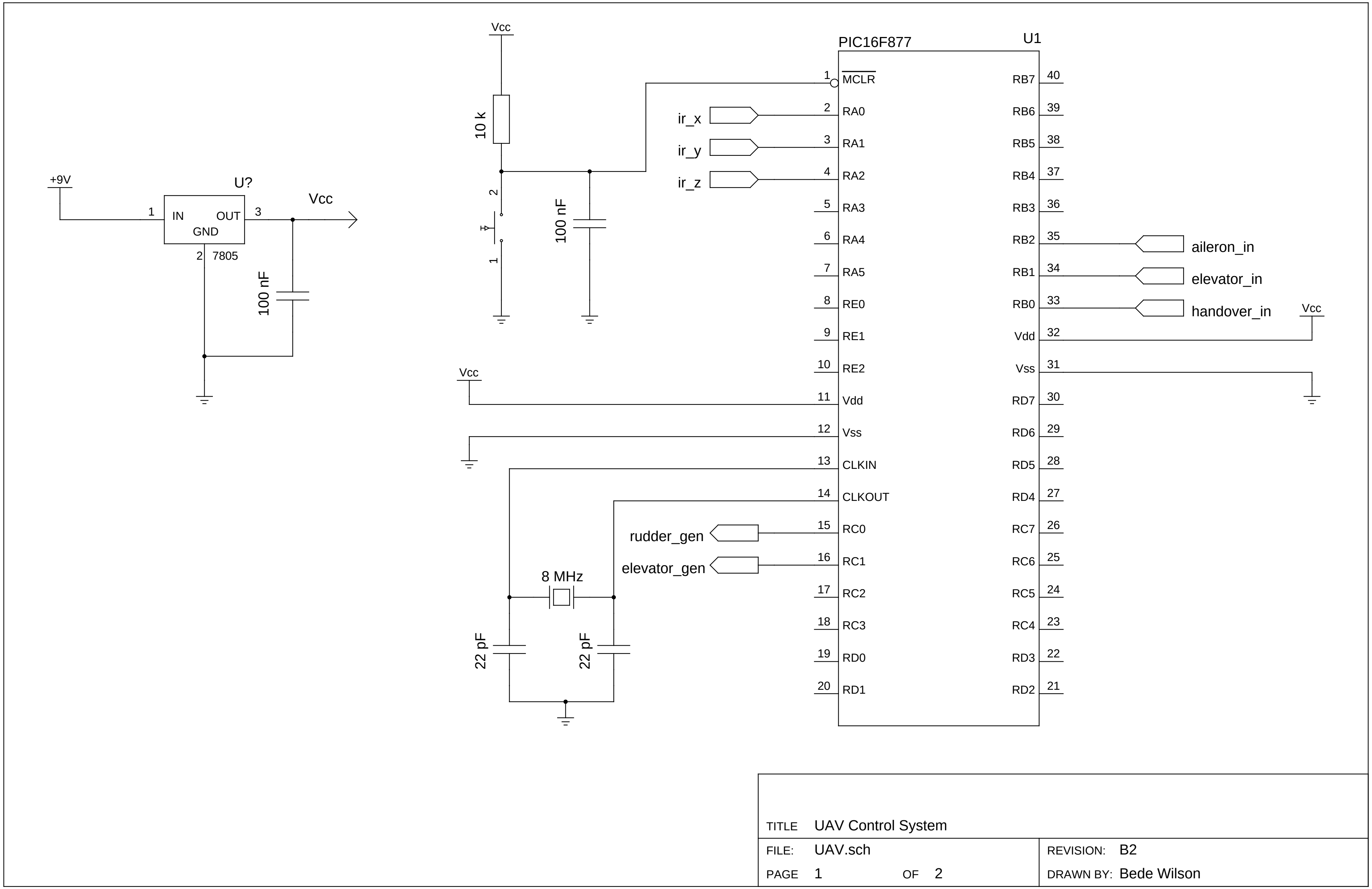
Agreed:

Bede Wilson (Student) Mark Phythian (Supervisor)

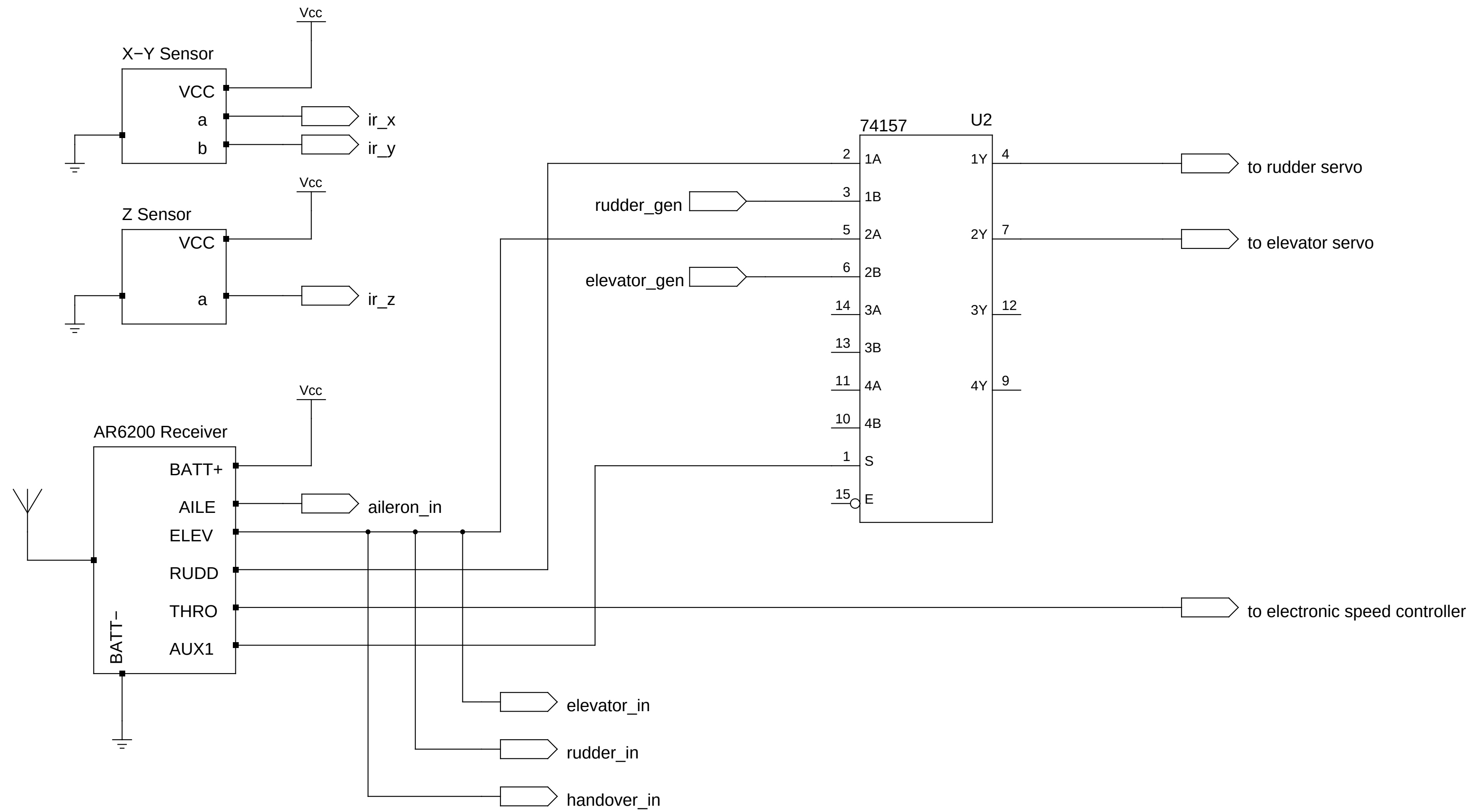
March 2010

Appendix B

Schematics



+9V
U?
IN
OUT
GND
7805
Vcc
100 nF
10 k
100 nF
PIC16F877
U1
MCLR
RA0
RA1
RA2
RA3
RA4
RA5
RE0
RE1
RE2
Vdd
Vss
CLKIN
CLKOUT
RC0
RC1
RC2
RC3
RD0
RD1
RB7
RB6
RB5
RB4
RB3
RB2
RB1
RB0
Vdd
Vss
RD7
RD6
RD5
RD4
RC7
RC6
RC5
RC4
RD3
RD2
ir_x
ir_y
ir_z
aileron_in
elevator_in
handover_in
rudder_gen
elevator_gen
8 MHz
22 pF
22 pF
TITLE UAV Control System
FILE: UAV.sch
REVISION: B2
PAGE 1 OF 2
DRAWN BY: Bede Wilson



TITLE UAV Control System

FILE: UAV1.sch

REVISION: B2

PAGE 2 OF 2

DRAWN BY: Bede Wilson

Appendix C

Code Listing

C.1 Stabilisation Control Mode

```
/*
  UAV control system - stabilisation mode
  Bede Wilson 2010
*/

// Variables for IR sensor measurement
double roll_error = 0;
double pitch_error = 0;
double max_error = 0;

// Servo pulse width determined by PID loops (uS)
long roll_out = 1500;
long pitch_out = 1500;

// PID persistant values
double roll_error_sum = 0;
double last_roll_error = 0;
double pitch_error_sum = 0;
double last_pitch_error = 0;
double cycle_time = 0;

void servo_out(){

  int tens_cycles;

  tens_cycles = ceil((roll_out*8)/10); // Calculate delay in
     tens of cycles
  PORTC.RC0 = 1; // Start rudder pulse
  Delay_Cyc(tens_cycles); // Delay for pulse width
  PORTC.RC0 = 0; // Finish rudder pulse
  tens_cycles = ceil((pitch_out*8)/10); // Calculate delay in
     tens of cycles
  PORTC.RC1 = 1; // Start aileron pulse
  Delay_Cyc(tens_cycles); // Delay for pulse width
  PORTC.RC1 = 0; // Finish aileron pulse

}

void calibrate(){

  unsigned tmp;
  tmp = Adc_Read(2); // Read Z sensor to get max
  max_error = tmp;

}

void get_attitude(){

  unsigned tmp;
  // Read IR sensors
  tmp = Adc_Read(0);
  // Determine error by scaling according to max (for 100 deg
     FOV)
  roll_error = ((tmp)/max_error)*50;
  // Repeat for pitch axis
  tmp = Adc_Read(1);
  pitch_error = ((tmp)/max_error)*50;

}
```

```
void pid(){

  // Roll PID gain terms
  int roll_kp = 150;
  int roll_ki = 30;
  int roll_kd = 30;
  // Pitch PID gain terms
  int pitch_kp = 150;
  int pitch_ki = 30;
  int pitch_kd = 30;

  cycle_time = (TMR1H * 0xFF + TMR1L)/1000000; // Get cycle
     time for calculation of derivatives and integrals

  roll_out = 1500 + ceil(roll_kp*roll_error + roll_ki*
     cycle_time*roll_error_sum + roll_kd*(roll_error -
     last_roll_error)*cycle_time); // Roll PID
  pitch_out = 1500 + ceil(pitch_kp*pitch_error + pitch_ki*
     cycle_time*pitch_error_sum + pitch_kd*(pitch_error -
     last_pitch_error)*cycle_time); // Pitch PID

  // Reset cycle timer
  TMR1L = 0;
  TMR1H = 0;

  // Limit max and min servo signal values
  if(roll_out >2400){roll_out=2400;}
  if(roll_out <600){roll_out=600;}
  if(pitch_out >2400){pitch_out=2400;}
  if(pitch_out <600){pitch_out=600;}

  // Save historic errors and calculate error sum
  last_roll_error = roll_error;
  last_pitch_error = pitch_error;
  roll_error_sum = roll_error_sum + roll_error;
  pitch_error_sum = pitch_error_sum + pitch_error;

}

void main(){

  // Initialisation
  // Set port directions
  TRISA = 0b00000111; // 3 inputs on port A (X-Y and Z sensor)
  TRISB = 0b00000100; // 1 inputs on port B (handover signal)
  TRISC = 0; // Port C as output (rudder and elevator)

  // Initialise analogue to digital converter
  ANSEL = 0b00000111; // Port A inputs are analogue
  ADCON1 = 0b10000000; // Right justified output, Vss and Vdd
     as references
  ADCON0 = 0; // Clock frequency is Fosc/2

  // Configure timer
  // TMR1 for measuring cycle times
  T1CON = 0b00010001; // Use internal clock and prescale for 1
     uS time increments

  // ******** BEGIN PROGRAM ********
  calibrate(); // Calibrates IR horizon detection system

  while(1){ // Infinite loop
```

```
    if(PORTB.RB2==1){ // If the UAV is in automatic mode...
      get_attitude();
      pid();
      servo_out();
      Delay_ms(16); // Delay so that servo pulses are
         generated every 20 mS
    }
  }
}
```

C.2 Fly By Wire Control Mode

```
/*
  UAV control system - fly by wire mode
  Bede Wilson 2010
*/

int aileron_width = 1500; // Stores aileron pulse width;
    default neutral
int elevator_width = 1500; // Stores elevator pulse width;
    default neutral
int pulse_count = 0; // Number of pulses recieved

// Variables for IR sensor measurement
double roll_error = 0;
double pitch_error = 0;
double max_error = 0;

// Servo pulse width determined by PID loops (uS)
long roll_out = 1500;
long pitch_out = 1500;

// PID persistant values
double roll_error_sum = 0;
double last_roll_error = 0;
double pitch_error_sum = 0;
double last_pitch_error = 0;
double cycle_time = 0;

void servo_out(){

  int tens_cycles;

  tens_cycles = ceil((roll_out*8)/10); // Calculate delay in
      tens of cycles
  PORTC.RC0 = 1; // Start rudder pulse
  Delay_Cyc(tens_cycles); // Delay for pulse width
  PORTC.RC0 = 0; // Finish rudder pulse
  tens_cycles = ceil((pitch_out*8)/10); // Calculate delay in
      tens of cycles
  PORTC.RC1 = 1; // Start aileron pulse
  Delay_Cyc(tens_cycles); // Delay for pulse width
  PORTC.RC1 = 0; // Finish aileron pulse

}

void calibrate(){

  unsigned tmp;
  tmp = Adc_Read(2); // Read Z sensor to get max
  max_error = tmp;

}

void get_attitude(){

  unsigned tmp;
  // Read IR sensors
  tmp = Adc_Read(0);
  // Determine error by scaling according to max (for 100 deg
     FOV)
  roll_error = ((tmp)/max_error)*50;
```

```
    // Repeat for pitch axis
    tmp = Adc_Read(1);
    pitch_error = ((tmp)/max_error)*50;

}

void pid(){

    // Roll PID gain terms
    int roll_kp = 150;
    int roll_ki = 30;
    int roll_kd = 30;
    // Pitch PID gain terms
    int pitch_kp = 150;
    int pitch_ki = 30;
    int pitch_kd = 30;

    cycle_time = (TMR1H * 0xFF + TMR1L)/1000000; // Get cycle
        time for calculation of derivatives and integrals

    roll_out = aileron_width + ceil(roll_kp*roll_error + roll_ki
        *cycle_time*roll_error_sum + roll_kd*(roll_error -
        last_roll_error)*cycle_time); // Roll PID
    pitch_out = elevator_width + ceil(pitch_kp*pitch_error +
        pitch_ki*cycle_time*pitch_error_sum + pitch_kd*(
        pitch_error - last_pitch_error)*cycle_time); // Pitch PID

    // Reset cycle timer
    TMR1L = 0;
    TMR1H = 0;

    // Limit max and min servo signal values
    if(roll_out >2400){roll_out=2400;}
    if(roll_out <600){roll_out=600;}
    if(pitch_out >2400){pitch_out=2400;}
    if(pitch_out <600){pitch_out=600;}

    // Save historic errors and calculate error sum
    last_roll_error = roll_error;
    last_pitch_error = pitch_error;
    roll_error_sum = roll_error_sum + roll_error;
    pitch_error_sum = pitch_error_sum + pitch_error;

}

void interrupt(){

    // Servo input handling
    if(INTCON.RBIF==1){ // Detect state change on port B
        if(IOCB.IOCB0==1){ // If change was on aileron input
            switch(PORTB.RB0){ // Check aileron input
                case 1: // On rising edge...
                    TMR0 = 0; // ...restart timer
                case 0: // On falling edge...
                    aileron_width = TMR0; // ...store pulse width
                    pulse_count++; // Increment pulse count
            }
        }
        if(IOCB.IOCB1==1){ // If change was on elevator input
            switch(PORTB.RB1){ // Check elevator input
                case 1: // On rising edge...
```

```
          TMR0 = 0; // ...restart timer
        case 0: // On falling edge...
          elevator_width = TMR0; // ...store pulse width
          pulse_count++; // Increment pulse count
      }
    }
  }

  // Reset interrupt flags
  INTCON.IOCB0 = 0;
  INTCON.IOCB1 = 0;
  INTCON.RBIF = 0;

}

void main(){

  // Initialisation
  // Set port directions
  TRISA = 0b00000111; // 3 inputs on port A (X-Y and Z sensor)
  TRISB = 0b00000111; // 3 inputs on port B (aileron and
     elevator RC, handover control)
  TRISC = 0; // Port C as output (rudder and elevator)

  // Initialise analogue to digital converter
  ANSEL = 0b00000111; // Port A inputs are analogue
  ADCON1 = 0b10000000; // Right justified output, Vss and Vdd
     as references
  ADCON0 = 0; // Clock frequency is Fosc/2

  // Configure timers
  // TMR0 for reading pulse widths
  OPTION_REG = 0b11010000; // Use internal clock and prescale
     for 1uS time increments
  // TMR1 for measuring cycle times
  T1CON = 0b00010001; // As above

  // ******** BEGIN PROGRAM ********
  calibrate(); // Calibrates IR horizon detection system

  // Enable interrupts
  INTCON.RBIE = 1; // Enable interrupt on port B change
  INTCON.GIE = 1; // Enable global interrupt

  while(1){ // Infinite loop while waiting for interrupt
    if(PORTB.RB2==1){ // If the UAV is in automatic mode...
      if(pulse_count==2){ // Wait until aileron and elevator
         signals have been received
        get_attitude();
        pid();
        servo_out();
        pulse_count=0; // Reset process
      }
    }
  }
}
```

Appendix D

Extracts From PIC16F887 Datasheet

- Key Features
- Pin Assignments
- Block Diagram

PIC16F882/883/884/886/887

28/40/44-Pin Flash-Based, 8-Bit CMOS Microcontrollers with nanoWatt Technology

High-Performance RISC CPU:

- Only 35 Instructions to Learn:
  - All single-cycle instructions except branches
- Operating Speed:
  - DC – 20 MHz oscillator/clock input
  - DC – 200 ns instruction cycle
- Interrupt Capability
- 8-Level Deep Hardware Stack
- Direct, Indirect and Relative Addressing modes

Special Microcontroller Features:

- Precision Internal Oscillator:
  - Factory calibrated to ±1%
  - Software selectable frequency range of 8 MHz to 31 kHz
  - Software tunable
  - Two-Speed Start-up mode
  - Crystal fail detect for critical applications
  - Clock mode switching during operation for power savings
- Power-Saving Sleep mode
- Wide Operating Voltage Range (2.0V-5.5V)
- Industrial and Extended Temperature Range
- Power-on Reset (POR)
- Power-up Timer (PWRT) and Oscillator Start-up Timer (OST)
- Brown-out Reset (BOR) with Software Control Option
- Enhanced Low-Current Watchdog Timer (WDT) with On-Chip Oscillator (software selectable nominal 268 seconds with full prescaler) with software enable
- Multiplexed Master Clear with Pull-up/Input Pin
- Programmable Code Protection
- High Endurance Flash/EEPROM Cell:
  - 100,000 write Flash endurance
  - 1,000,000 write EEPROM endurance
  - Flash/Data EEPROM retention: > 40 years
- Program Memory Read/Write during run time
- In-Circuit Debugger (on board)

Low-Power Features:

- Standby Current:
  - 50 nA @ 2.0V, typical
- Operating Current:
  - 11 μA @ 32 kHz, 2.0V, typical
  - 220 μA @ 4 MHz, 2.0V, typical
- Watchdog Timer Current:
  - 1 μA @ 2.0V, typical

Peripheral Features:

- 24/35 I/O Pins with Individual Direction Control:
  - High current source/sink for direct LED drive
  - Interrupt-on-Change pin
  - Individually programmable weak pull-ups
  - Ultra Low-Power Wake-up (ULPWU)
- Analog Comparator Module with:
  - Two analog comparators
  - Programmable on-chip voltage reference (CV_{REF}) module (% of V_{DD})
  - Fixed voltage reference (0.6V)
  - Comparator inputs and outputs externally accessible
  - SR Latch mode
  - External Timer1 Gate (count enable)
- A/D Converter:
  - 10-bit resolution and 11/14 channels
- Timer0: 8-bit Timer/Counter with 8-bit Programmable Prescaler
- Enhanced Timer1:
  - 16-bit timer/counter with prescaler
  - External Gate Input mode
  - Dedicated low-power 32 kHz oscillator
- Timer2: 8-bit Timer/Counter with 8-bit Period Register, Prescaler and Postscaler
- Enhanced Capture, Compare, PWM+ Module:
  - 16-bit Capture, max. resolution 12.5 ns
  - Compare, max. resolution 200 ns
  - 10-bit PWM with 1, 2 or 4 output channels, programmable "dead time", max. frequency 20 kHz
  - PWM output steering control
- Capture, Compare, PWM Module:
  - 16-bit Capture, max. resolution 12.5 ns
  - 16-bit Compare, max. resolution 200 ns
  - 10-bit PWM, max. frequency 20 kHz
- Enhanced USART Module:
  - Supports RS-485, RS-232, and LIN 2.0
  - Auto-Baud Detect
  - Auto-Wake-Up on Start bit
- In-Circuit Serial Programming™ (ICSP™) via Two Pins
- Master Synchronous Serial Port (MSSP) Module supporting 3-wire SPI (all 4 modes) and I^2C™ Master and Slave Modes with I^2C Address Mask

Pin Diagrams – PIC16F884/887, 40-Pin PDIP

40-pin PDIP

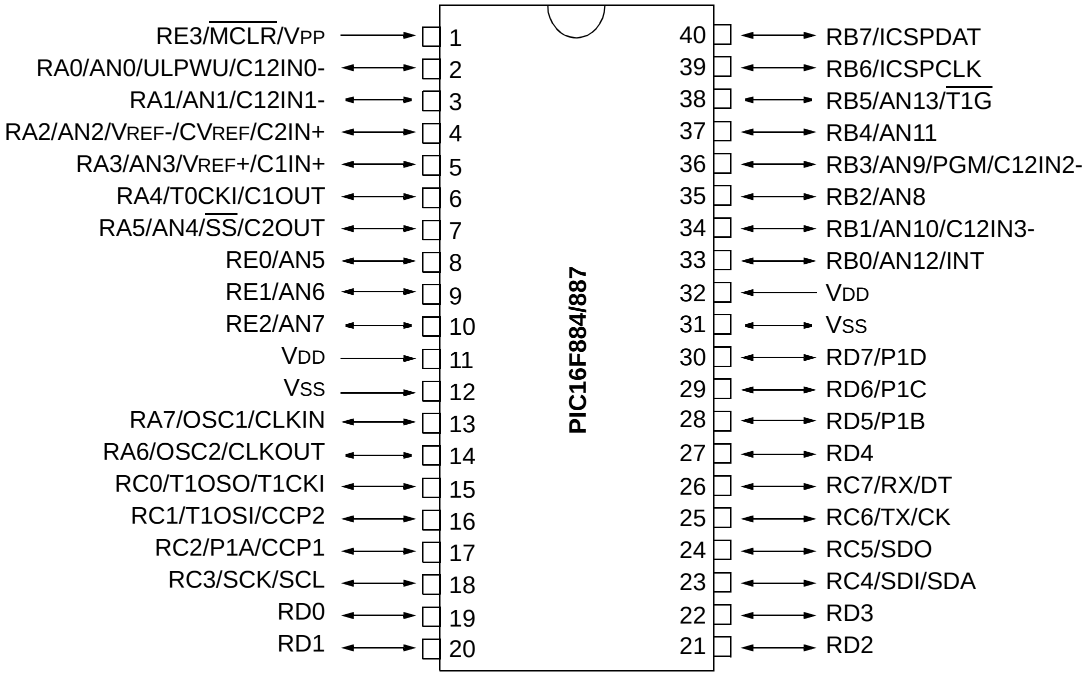
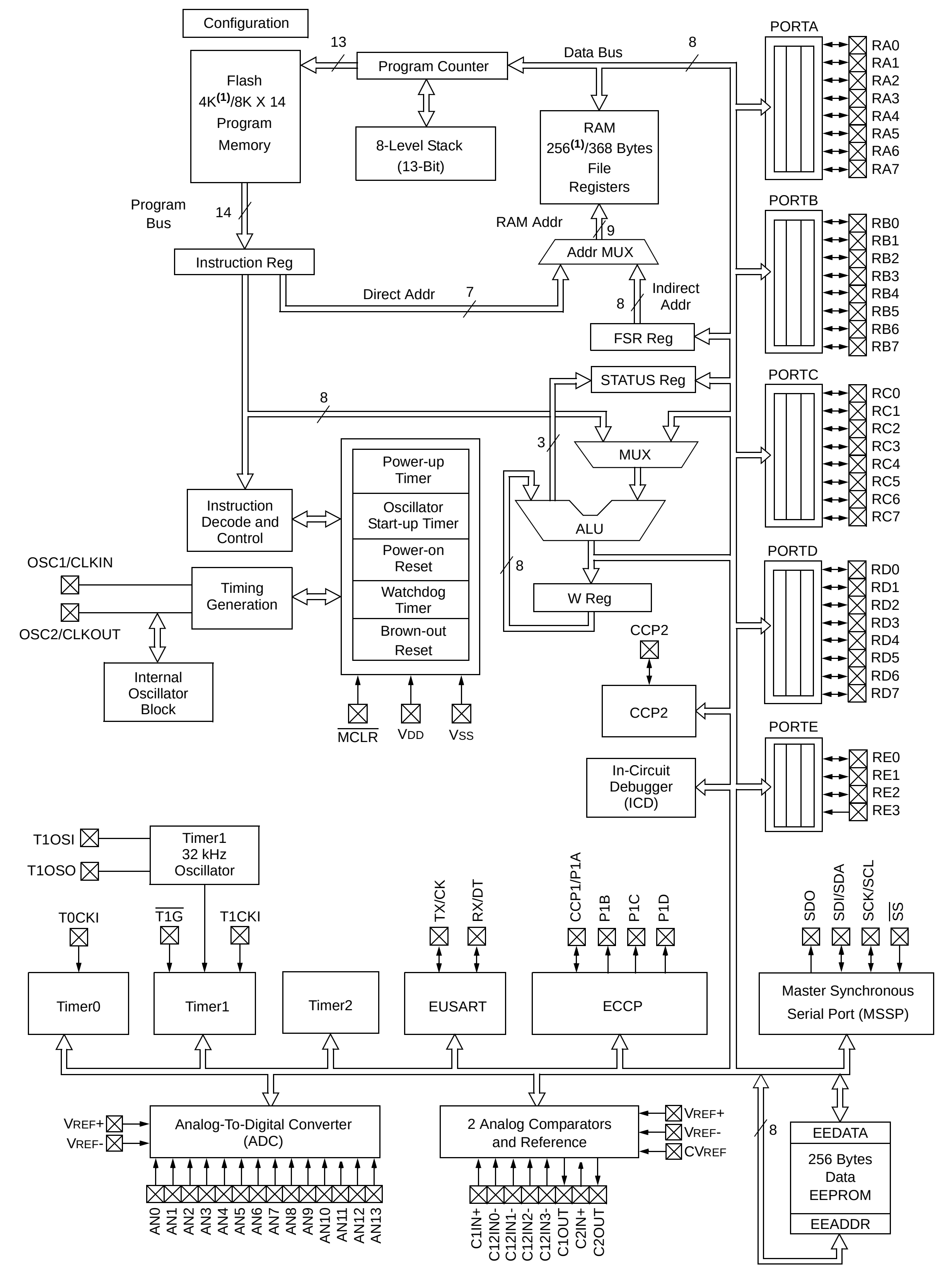


FIGURE 1-2: PIC16F884/PIC16F887 BLOCK DIAGRAM



Note 1: PIC16F884 only.