1. Abstract

An autonomous flying platform is an Autonomous Unmanned Aerial Vehicle (AUAV) with the ability to hover. In previous years, research has been undertaken to build an autonomous flying platform and in the main have been unsuccessful due to stability issues. The previous years approach of electrically powered ducted fans provided a rapidly responding method of providing thrust for stability and height control, by regulating the rotational speed, and successfully managed to hover.

This year, through more extensive investigations and using previous research knowledge, an attempt was made to further develop the flying platform to improve the stability control, and to remove the need for a separate off-board power supply. The platform utilises previous years plans for five ducted fan units, but replaced the central fan with an Internal combustion engine in order to improve lift, and reduce the overall electrical power required. The electric ducted fans were powered from brushless permanent magnet DC motors and electronic speed controllers utilised in last year's project, and larger variants of these motors were researched and used as a generator in combination with a internal combustion engine to form an on-board power supply.

In a previous year's project, an internal combustion engine was used to develop the lift, but had been unsuccessful. This year saw a return to an IC engine, with additional research in order to find the most effective static and dynamic response performance.

Stability problems have been a major concern in many of the previous year's projects, which has led to a step change in technology this year. This year an IMU was made available from BAE Systems in order to gain a reliable control signal of the accurate position of the platform. A modular electronic control system was developed to allow each part of the control system to be researched, designed and tested separately. All speed controllers utilised a Pulse Width Modulated logic signal, similar to that required by radio control servos used for the throttle control on the IC engines. The system was designed to allow for correct set up procedures to be completed, and for an emergency position to be made available to switch off the platform in the event of a problem in flight.

Power sources were researched to power the electronic control fans, and the control system. Different battery types were investigated continuing on from previous year's work, and additionally this year, super capacitors were included. However they were not found to be suitable and hence the IC generator was continued to be investigated with the intent to replace the umbilical lead used by last year's group.

2. Acknowledgements

The author would like to thank the following people for their support and enthusiasm throughout this project…

- Dr. Martin Jenkins – the project supervisor
- Dr. Garry Lester – the project co-supervisor
- Dr. Mike Horwood for his assistance with PIC programming
- David Weightman for his assistance with acquiring various parts and equipment
- Whipton Craft and Model centre for assistance and enthusiasm with “Glow plug” technology
- Mr Tony Moy, BAE Systems for the IMU and advice given
- Mr Keith Nock, The Hobby Box, Uckfield, for assistance with short notice supply, and benefit of modelling knowledge
- www.ic-prog.com for high quality free PIC programmer software
- Liam Dushynsky, James Mackenzie-Burrows, Kevin Lowis, Richard Holbrook, Richard Forder, Rebecca Hughes, Chris Poczka, and Jody Muelaner.

It is the belief of this project group that any further model products that are required should be sourced through Whipton Craft and Model centre due to his overwhelming help and enthusiasm with this project.

3. Table of contents

1 Abstract……………………………………………………………………………….. 1

2 Acknowledgements………………………………………………………………… 2

3 Table of Contents………………………………………………………………….. 3

4 Introduction & Background…………………………………………………….5

5 Stability propulsion research……………………………………………………7

i Introduction…………………………………………………………………… 7

ii Theory…………………………………………………………………………..7

iii Results and Analysis…………………………………………………………8

v Conclusion……………………………………………………………………...9

6 Speed Controller Operation……………………………………………………10

i Introduction……………………………………………………………………10

ii Schulze 32.55 start up sequence……………………………………………… 10

iii Schulze 32.55 settings………………………………………………………11

iv Warning notes and cautions……………………………………………….. 11

v Conclusion…………………………………………………………………… 12

7 Control System – Speed controller interface……………………………………13

i Introduction……………………………………………………………………13

ii Theory……………………………………………………………………...13

iii Analogue circuit consideration……………………………………………...14

iv QBasic™ Program construction and use……………………………………15

v Microchip™ PIC16F873 design and build…………………………………18

vi Conclusion…………………………………………………………………… 22

8 Propulsion system Testing…………………………………………………….. 23

i Stability Fans…………………………………………………………………. 23

ii Central Ducted Fans………………………………………………………….. 25

9 Electrical Power Systems……………………………………………………… 27

i Introduction……………………………………………………………………27

ii Generator Set Research & Selection………………………………………….27

iii Platform Power Requirements………………………………………………...30

iv Analysis…………………………..………………………………………….31

v Conclusion……………………………………………………………………31

10 Radio Control systems……………………………………………………………32

i Introduction……………………………………………………………………32

ii Theory………………………..………………………………………….. 32

iii Development………………..…………………………………………….. 33

iv Analysis…………………………..………………………………………….35

v Conclusion……………………………………………………………………35

11 IMU Emulator circuit……………………………………………………………………. 36

i Introduction………………………………………………………………………36

ii Theory…………………………..…………………………………….….…….. 36

iii Development……………..……………………………………………….….….. 37

iv Analysis…………………………..………………………………………………38

v Conclusion…………………………………………………………………….38

12 Implementation……………………………………………………………………39

13 Final conclusions, further work……………………………………………………. 40

14 Appendices…………………………………………………………………………41

14.1 Appendix 1 Product Design Specification………………………………………41

14.2 Appendix 2 Electronic Interfaces……………………………………………….. 43

14.3 Appendix 3 QBasic™ program listing and instructions…………………………46

14.4 Appendix 4 PWM Interface circuit diagrams and program listing………………49

14.5 Appendix 5 Propulsion Test Results……………………………………………..59

14.6 Appendix 6 Genset Results………………………………………………………64

14.7 Appendix 7 Speed controller Interface Instructions…………………………….. 69

14.8 Appendix 8 Genset Test Instructions……………………………………………72

14.9 Appendix 9 Radio control Systems………………………………………………76

14.10 Appendix 10 IMU Emulator Program code and circuit diagram..………………78

14.11 Appendix 11 Example Meeting Agenda, and Minutes…………….……….……97

15 References……………………………………………………………………...102

4. Introduction & Background

Flying platforms are a specific class of Unmanned Aerial Vehicle (UAV) with the ability to hover. UAVs have recently been in the public eye with "Predator" and its successful military application in reconnaissance and attack roles. More recent military campaigns in the Middle East have led to more enthusiasm and pressure being placed on manufacturers to produce a stable and reliable vehicle to reduce the risks placed on pilots during the height of battle conditions. They can eradicate the risks posed to pilots whilst simultaneously reducing the cost of the aircraft and of pilot training. There are also advances to be made with the removal of the cockpit reducing weight, and size within the aircraft and also allows the forces placed upon the aircraft during flight to exceed those of human survivability.

Further possibilities for an UAV exist in the civilian sector. There are many inspections and monitoring services that are currently carried out by manned aircraft including the inspection of overhead electric cables, and the monitoring of traffic flows on motorways etc. The vehicle can also be used for police surveillance purposes operating remotely from a central base, allowing many vehicles to be controlled with fewer ground staff, and with less costs involved, more can be available in a wider area. However any proposed civilian unmanned vehicle will first have to gain authority to fly in British airspace.

Flying platforms are very complex vehicles as they have little or no aerodynamic stability, making hovering impossible without some form of human, mechanical or electrical control system. There have been a number of projects similar to this in the past, with the most notable being the "Hiller flying platform", shown in figure 4.1, where by a relatively small platform was designed and constructed to carry a human, which also formed the control system. The advances in technology has allowed electronic sensors together with advanced electronic control systems to replace the human altogether, and one of the first companies to achieve this was RC Toys in the USA, in the form of Draganflyer III, and shown in figure 4.2.

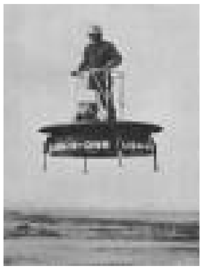


Figure 4.1 Hiller flying platform [1]

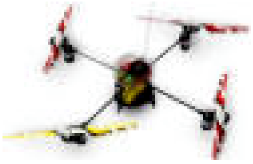


Figure 4.2: Draganflyer III, RC Toys [2]

There have been a number of previous projects at the University of Exeter to attempt this problem, and with the exception of the 2002-3 group none have yet managed to get a stable flying platform. Last year's group managed to get the platform to remain fairly stable in flight, but were not entirely happy, with a number of glitches. Previous groups have looked at inherently stable designs, as well as vectored thrust designs, but they have all been ultimately unsuccessful.

This year's group took onboard the points raised by the previous year's research efforts. The use of electronic ducted fan units was integrated into the new designs as they were proved to be reliable, light weight, have good response times, and are relatively safe. Ordinary propellers have less thrust than the ducted variety and given that the edges of the blades are exposed; they are more dangerous to the user. Last year's group used five ducted fans arranged with one in each corner of the platform and a centrally mounted duct, all of which were of the same diameter. The group found that although there was sufficient thrust to allow the platform to hover, there was insufficient to carry a power source. This year it was decided to replace the centre duct with a larger diameter duct with a internal combustion engine to power it, with the intention to gain additional lift to allow some form of power supply to be carried for the outer stability fans. A number of IC engine technologies were considered, with a large capacity glow plug engine being chosen to operate the main duct.

The issue surrounding power supplies from last year was progressed with investigations into battery power finding no viable systems that could manage the duration required, without an exceptional amount of weight to be carried. It was decided to investigate small generators, powered by glow plug internal combustion engine. The power to weight ratio was of primary concern, as was the amount of fuel that would be consumed during the specified flight.

The final main goal of this year's project was to replace the accelerometers and gyros used by the previous group with a higher quality IMU, incorporating gyros for all three axis and accelerometers for all three axis. BAE Systems donated this, together with full datasheets and with helpful advice for installing and using the device.

A full product design specification for this years project is given in appendix 1.

The group was split into four sections, with the main posts also being filled. Liam Dushynsky chaired the group with James Mackenzie-Burrows as secretary and Kevin Lowis as treasurer. Alex Tombling was responsible for the agenda's and chaired one meeting in the absence of the chairman. Kevin Lowis also chaired the control subgroup, Richard Holbrook chaired the Propulsion subgroup, and Alex Tombling chaired the Electrical Power Systems subgroup. The structure subgroup consisted of only Chris Poczka for the majority of its exsistance, and so did not have a separate chairman.

All electronic designs used a standard procedure for the program it was produced in, and for its numbering. All designs were labelled with the makers' initials and a three-digit number, e.g. AT001. All designs were produced in Multisim 2001, and kept on file on WebCT on the university of Exeter server.

5. Stability propulsion research

5.i Introduction

From the conclusions reached from the previous year's report, it was decided to continue with electric motor powered ducted fans to form the stability thrust for the platform. [3] The 2002-3 group had shown that the Plettenberg HP220/30/A4 S P4 brushless DC permanent magnet motors were around 90% efficient giving a maximum of 20N of thrust when used with the WeMoTec Midi fan. [3] The motors were controlled using the Schulze Future 32.55 brushless speed controller, designed to operate with this particular motor.

It was decided to briefly research further into motors for the stability control fans in order to ensure no newer technology has become available, and to ensure all possible alternatives had been exhausted. One of the major concerns from the previous groups work was that not enough power was available to lift the platform and its power supply, so this needed attention at an early stage.

5.ii Theory

Various motor technologies were re-investigated, but no new technology was found. The permanent Magnet Brushless DC motor has only become popular in recent times as the electronics required to operate them has become more freely available. These motors are efficient as they remove the largest loss in a traditional motor – the brushes. The windings, therefore, need to be placed on the casing and not on the rotor; instead the permanent magnets are fitted to the rotor. This has a number of benefits, including: -

- The length of windings can be extended,
- Smaller losses, and less heat build up,
- Better heat dissipation.

As the windings are placed outside the rotor, they have more available area, and so can be lengthened to fill the additional space, giving rise to larger torque values. This more powerful motor can be of benefit on applications were torque is preferable over speed. Without brushes there are no losses of power due to the arcing that was previously inevitable, giving the secondary benefit that without arcing, there is no additional heat energy being wasted. As the windings are powered on any motor, they tend to dissipate heat, when this is in the centre of the motor, it can cause problems heating up the entire device causing distortion and failure. However, with the brushless motor, the windings are on the outside, and can utilise the casing as a heat sink to remove the heat from the bearings and main drive shaft, reducing the wear, and fatigue problems that can be caused by overheating. The main disadvantage of these types of motor is the initial cost, and the cost of the associated electronics required to power them.

Commutation on these motors is performed electronically, and so expensive power MOSFETs are required. At high speed, the back emf produced by the Hall effect as the rotor rotates is used to synchronise the control electronics with the rotor position. This is difficult to achieve at low revolutions, as the back emf is too low to be detected, and thus the motor will fail to rotate smoothly if at all, and so will normally have a minimum operating speed to avoid this region.

These brushless motors need controlling by a specific type of controller, as they effectively work as a three-phase motor. The most common method of DC motor control is to regulate the voltage feeding the motor, normally by means of a Variable resister. However this has a number of disadvantages: -

- Very Effective
- Generates Heat
- Wastes Power

This example of speed controlling is inappropriate as three-phase motors were being used on this project.

The second most common method of speed control is by Pulse Width Modulation. By using a simple PWM circuit, the heat generated can be reduced, and the power wastage minimised. Pulse Width Modulation works by varying the proportion of on to off times on a simple square waveform. Common practice is to keep the cycle time constant, and vary the on/off proportion, by extending the on time. An alternative is to keep the on time constant, and vary the length of the off time, hence extending the cycle length. With slow cycle lengths, generally greater than ¼ second, it is possible to see the motor change speed with the cycle, but with fast cycle times of 20mS, this becomes invisible. A basic example of a PWM circuit for DC Motor control is given in figure 5.1.

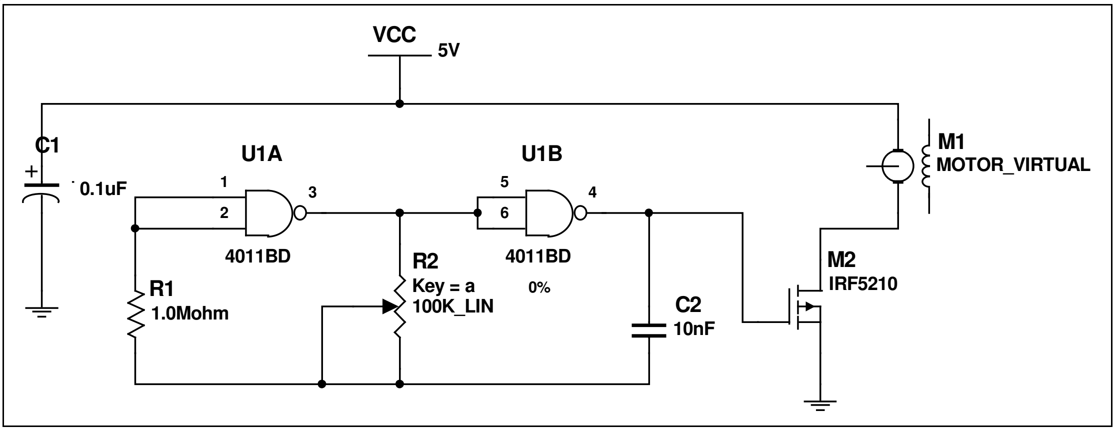


Figure 5.1: Pulse width modulation speed controller

In this example, varying the pot changes the length of the oscillation, and hence the speed of the motor. It should be noted the MOSFET would get hot. Again, due to the three-phase nature of the motor this circuit is inappropriate. Brushless motors require a complex control system, and the best route to operate these motors is to purchase of the shelf controllers. These were already purchased by the 2002-3 group and so were continued to be used. The controllers purchased were the Schulze Future 32.55 brushless speed controller at £225 each. These devices utilised a PWM DC input signal, identical to that used by traditional radio control servos, to control the eventual speed of the motor.

5.iii Results and Analysis

On the WeMoTec datasheet, the Plettenberg HP220/30/A4 S P4 was shown to give 21N of static thrust at a rotational speed of 32,400rpm, using an input power of 1.1Kw, (40.5A at 27.3v), [4] when used with the Schulze speed controller. The speed controller motor timing can affect the performance of the motor dramatically, with each motor designed to operate with an optimum timing setting [4] and it is important that speed controllers are matched for efficient use. For the Plettenberg motor used, it is recommended to have a switching frequency of approximately 19kHz.

The ordinary Plettenberg motor was restricted to speeds of 25,000rpm, [5] however as these units have the 'S' suffix this speed may be exceeded as they are made of hardened material. The previous group report stated that the S rating allowed the motor to rotate at up to 80,000rpm, due to the Kevlar reinforcing material used to keep the magnets attached to the shaft. [3] The motor should stabilise at 100^{o}C when run at 25,000rpm. The motors use neodymium magnets, which can demagnetise at temperatures above 150oC. [15]

Given that all these components for the stability side of the propulsion system had already been purchased and tested by the previous group, it was decided to continue using the same items on this years project. Electrically the motors had only one interface, the three phase wires to the speed controller, while the controller itself had a power supply pair, and three control wires from the control system. These control wires were 5v supply, reference ground and a PWM control signal, designed to operate with radio control equipment within the model market. This will be covered in more depth in section 6.

5.iv Conclusions

The Plettenberg/Schulze set up (see figure 5.2) appeared to be effective in last years project, and given the outlay already made by the university it was inappropriate not to use them on this project. The motors were adequately powered for stability purposes, and were efficient in their use of power. There were no reliable specifications of the response times, or of the thrust output at different power levels, and so these needed to be reinvestigated by this years' team. In order to keep an operational platform in one piece, the control interface used by last year's group was not reused this year, and a new updated version constructed, and is covered in detail in section 6.

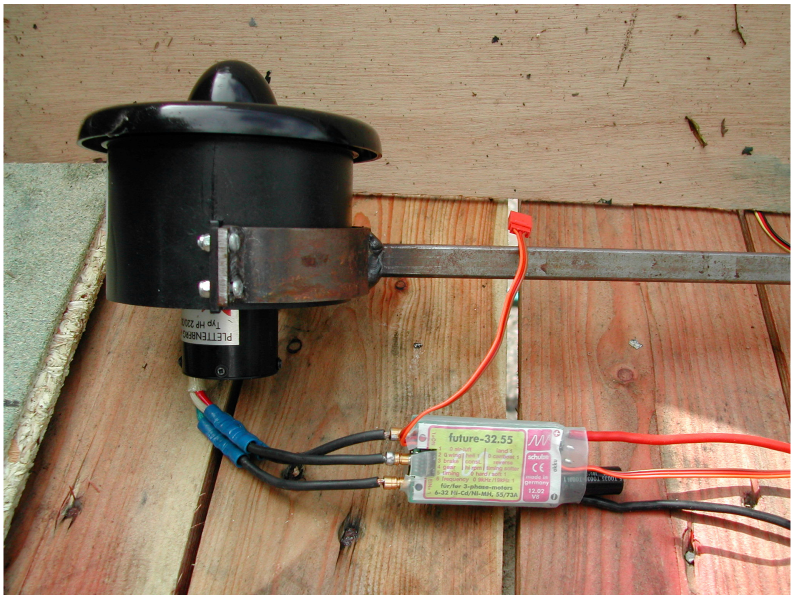


Figure 5.2: Plettenberg HP220/30/A4 S P4, and Schulze Future 32.55 controller

6. Speed controller operation

6.i Introduction

To control the rotational speed of the motors, the speed controller requires a signal to state the required speed. These speed controllers are primarily designed for use with radio controlled receivers and so require a signal identical to that produced by radio control equipment.

For initial testing a Basic™ program was written to imitate the signals given by a radio receiver, but this was unreliable as Microsoft Windows™ interrupts caused poor repeatability of the cycle times of the control pulse. A new interface was designed using the circuits developed by previous years', however these were deemed too large and bulky to be useable. A Micro controller was used to interface between the control system on the platform and each speed controller, and is discussed in full in section 7.

The speed controllers require a specific start up sequence in order for the electronics in the speed controller to set up the 100% and 0% points of the linear speed scale.

6.ii Schulze 32.55 start up sequence

Not all radio control equipment is perfect, and not all equipment is trimmed accurately, and so most speed controllers are designed and constructed with an intelligent programming system, which allows it to be uniquely adjusted for the particular incoming signal. In order for full performance to be realised from the motors, whilst maximising the resolution, the controllers need to be given the maximum and minimum settings, and the controllers told when these are being sent.

There are two different methods of loading these settings in to the controller:

1. Send the 'stop' command to the controller, followed by the 'full-speed' signal, or
2. Send the 'full-speed' signal, followed by the 'stop' signal.

Both methods will set up the controllers to give the same range and start/stop points but have different results immediately after set up is complete. The first command must be being sent to the controller at the point when transmission first starts from the interface, or when the speed controller power is connected.

When main power is applied to the speed controller, it emits a series of bleeps, through the motor itself, in a raising series of tones. Once one end of the scale is detected, it emits either a single beep for 0% or a double beep for 100%. The first reading is stored immediately once it falls within the boundaries of either end of the scale. The controller does nothing until the second end stop is sent, whereby it will emit the second appropriate tone. Once both settings have been stored, the controller will power the motor to whatever speed is being fed in at that time. Therefore in the options given above, if the first method were used, once the controller had received the 100% power reading, the motor would start up at full power, whereas with option 2, the motor would remain stationary, as it would be reading 0%. The second option is favourable for a number of reasons. Primarily once fitted to the platform, all 4 fans changed state from off to 100% simultaneously, the immediate current draw would be in excess of 120A, from a previous 2A. This dramatic change in current draw would cause considerable problems with the generator, or excessive strain on any battery used to supply it. It would also cause undue stress on the motors themselves and the platform structure. The other major concern with the control fans changing state so quickly during a set-up cycle is that it would cause alarm to those operators surrounding it. This could be dangerous with such potentially lethal devices. In order to ensure set up was completed safely and reliable, the start up procedure was written in to the interface, so that it could not be avoided.

There are a number of in-built monitors with the Schulze speed controller. The temperature monitor will switch off the motor in the event of the controller getting too hot. If the motor windings are shorted, this monitor will operate quick enough to prevent damage, and so this cannot be relied upon to detect this fault condition. In the event of overheating, the system can be reset by sending the 0% signal to the controller for around 2seconds. This will re-arm the controller, and will continue as before. The controller is very susceptible to overheating when being used at high speed, and thus high power. The previous group fond that by placing the controller in the air flow enough cooling took place to remove any problems encountered. This must be considered again this year, and adopted unless there are considerable issues with doing this.

A voltage monitor is also fitted to these controllers that will power down the motor in the event the supply voltage drops below a preset threshold. The Schulze controllers are set to switch off at 5V, and can be re-armed in the same way as for the temperature monitor.

The devices also include a speed monitor, for both maximum and minimum speeds. If either extreme is breached the device shuts the motor down, until back within its limits. Below a certain speed, the controller cannot correctly detect the actual speed, and so stops the motor altogether. This protective feature can make the motor reluctant to start of there is too much load on it, which can also cause the current to be exceeded, tripping the current monitor. When the current drawn by the motor exceeds the controllers rated maximum, it will limit its output. However if it remains in current limited mode for some time, the device will switch off to protect both the motor and the controller. It is important to note that this will only detect over current on the controller, and so if a smaller motor is used, this will become damaged before the controller will take note.

The final monitor on the Schulze controller detects the signal from the interface or radio control receiver. In the event this signal is longer or shorter than normal, it will switch to disarm mode. This prevent interference from giving wrong signals to the device, but means that the designer of any interface must remain within tolerances and keep a repeatable signal being fed into it.

6.iii Schulze 32.55 settings

There are a number of DIL switches on the Schulze speed controllers for various different functions. DIL switch 1 and 2 should be set to off, giving the “winged aircraft” setting. DIL switch 3 enables the brake feature; this is not used and thus should be set to off. DIL switches 4 and 5 selects a belt driven gearbox, and the responsiveness of the motor; a belt drive is not used on the fans, and so they should both be off to give a highly responsive fan unit.

6.iv Warning notes and cautions

Electric motors fitted with propellers are dangerous and require proper care for safe operation. Keep well clear of the propeller or fan blades when the power supply is connected. These speed controllers are not protected against reverse polarity connection, and incorrect connection will cause irreparable damage, as was proven during this year's project. Also connection of the power supply to the motor terminals will cause equal damage.

Speed controllers are sensitive to humidity, and should not be used in wet environments without correct protection. They must also be protected against mechanical loads, vibration and dirt contamination.

The cables on the controller should be kept as short as possible. The motor cables should not exceed 10cm, and the power supply cables should not exceed 20cm. When measuring the current drawn from these high power controllers, always use a clamp type ammeter, not a series meter with shunt resistor, as this will effect the operation significantly. When the controller is not in use, the power supply must be disconnected, and the 0% command not relied upon. This is especially important when persons may need to gain access to the platform in the vicinity of the ducted fan units for adjustments. When the speed controllers are slowing the motor down, the motor will be producing more voltage (through the power generation rules) than is being supplied by the power source. This can be significantly higher than the power supply, and so must have some form of reservoir to 'dump' this excess power. When using a battery, it must be able to cope with short duration of high voltage charging, such as a lead acid battery, or with a generator, a circuit able to deal with these excessive power fluctuations. This power must always be used, and cannot be left floating in the controller, i.e. if the battery was not connected, it would cause damage to the controller.

6.v Conclusions

The Schulze speed controllers are very simple to operate and this was a good choice made by the previous group. Due to the way in which all speed controllers operate, when connecting the power supply, there is a large spark, due to the high peak current charging the internal capacitors. Care must be taken in connecting the polarity of the power supply correctly, and the resulting spark emitted as the connection is made should not be mistaken for that of the capacitors charging.

The controllers must be placed in the airflow from the ducted fan in order to keep them cool; this may reduce thrust slightly, but is the only feasible method of cooling without resorting to additional cooling. The structure could be used as a heatsink, but may not be sufficient for prolonged use.

A picture of the Schulze future-u 32.40k, a member of the same family of controllers as the 32.55 is given in figure 6.1

Figure 6.1: Schulze 32.40k controller

7. Control System – Speed controller interface

7.i Introduction

There are two control systems used to stabilise the height, and pitch/roll of the platform. Each control system uses the IMU as its main sensor, and outputs all control levels for the stabilisation-ducted fans and the central lift ducted fan as a 0-5V analogue DC signal.

In order for all model radio control manufactures to work together, they utilise a standard Pulse Width Modulated DC signal to give a specific setting. Thus the Schulze speed controllers and the servos for the throttle control all need a PWM digital DC signal. Therefore the output from the control system needs to be converted to a PWM signal. This was a simpler method than having the control system derive the PWM signal itself.

The outputs of the control systems are DC voltages variant on the input to the control system. In a change form the previous project where a 0V was defined as no change, a +5V signal as full thrust, and –5V as minimum thrust, this year it was decided to output 0V as minimum thrust, and +5V as maximum thrust. The control module rather than the interface would then determine the normal operating thrust, as it is more appropriate for it to be defined by the controller rather than the link.

The interface is required to convert the +5V DC signal into a 1.1-1.9ms 5V pulse as defined in section 7.ii below. The specifications of the control systems were that each part of the system should be independent of one another. The pitch part of the control system is connected to two opposite fans, while the roll control is connected to the other two opposite fans. The height control is to be connected only to the central IC powered ducted fan. A base thrust level has to be established in order to get the platform to hover. However, it was deemed that the central fan will not be able to provide all the thrust required to provide neutral buoyancy, and so the outer fans will need to operate at a nominal speed to provide additional thrust. It is therefore considered that one of each pair of control fans will operate at a preset thrust, while the other is adjusted to speed up or slow down as required to correct any tilt that is detected by the IMU.

The interface must incorporate the start up procedure for the speed controllers, and be able to shut down all the controllers in the event of an emergency. This system must be able to respond to a shut down request immediately in order to provide a safe environment in which to work following an emergency condition. The interface must remain within the boundaries set out in section 6 above in order to prevent the controllers from shutting down prematurely.

Appendix 2 shows the electrical relationships between each of the sections on the platform. This was used to insure each group new what signals were to communicate with each other so that a common ground could be set. This prevented the group from working to two different standards and not realising they would not work until it was too late.

7.ii Theory

In keeping with the rest of the electronics it was first considered to remain with analogue electronics. However the large component count used by the previous project, over 100 items, it was deemed to be quite large and bulky for the platform, and possibly a higher current drain than was necessarily needed. It was therefore considered to look at microprocessor control with a Microchip™ PIC controller.

Each company involved in the manufacture of radio control equipment for the model market follows the same standard for signals to control the equipment. These signals are in the form of a pulse with duration of nominally 1ms to 2ms repeated every 20ms, a duty cycle of between 5% and 10%. The full range of on time between full power and stop is thus 1ms, giving the off time to be between 18 and 19ms in total.
The accepted signals for the controller are: -

- Low pulse 0-0.3v
- High Pulse 3.3-6.0v.
- Cycle time 10-30ms
- Pulse range 0.8-2.5ms.

This gives a small amount of margin to be used when designing the final control interface. The duration of off 'dwell' time is important for most radio control uses. This is due to when the signals are sent by radio frequency from the transmitter to the on board receiver. In order to multiplex up to eight channels of information they are placed on the carrier one after another, hence a large gap between each on pulse of each channel. The 1ms minimum pulse is to allow the receiving module to count between each channel, and to discount noise that may be picked up. Figure 7.1 shows the standard signal sent to radio control equipment such as the throttle servo and speed controllers.

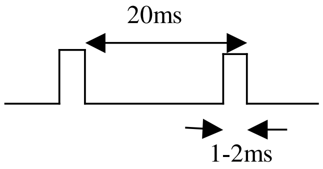


Figure 7.1: PWM signal for radio control modules

The 1-2ms pulse is not converted exactly linearly by the speed controllers, but is within very fine tolerances, and may be regarded as linear.

In order to allow initial testing of the stabilisation fans, a short QBasic™ program was written to replicate the signals from a RC receiver. The digital circuit then followed this.

7.iii Analogue circuit consideration

The basis of the purely analogue circuit was the previous years' project. The principle was to use an integrator to produce a positive linear voltage ramp (see figure 7.2). The output voltage of the control system was then compared with the ramped voltage to generate the required pulse width signals (see figure 7.3).

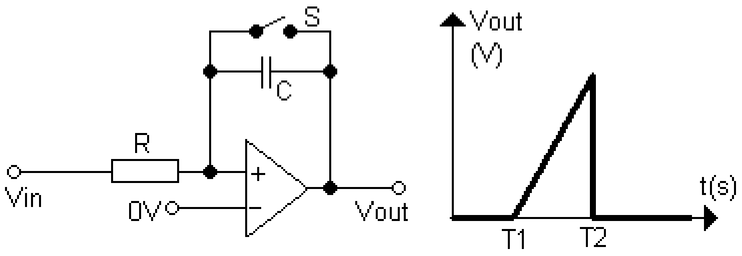


Figure 7.2: Integrator schematic and output [7,8]

Assuming that switch S is initially closed, once opened (t=T1), the capacitor will begin to charge. As it does so, the output voltage will rise until it reaches the magnitude of Vin, but inverse in sign. Once switch S is re-closed (t=T2), the capacitor is immediately discharged and the output voltage returns to 0V. The gradient of the slope, m, is given by equation 7.1.

$$m=V_{in}/(C.R) \qquad (7.1)$$

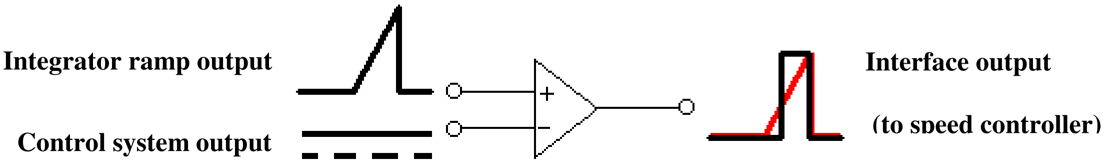


Figure 7.3: Fundamental principle of interface

The pulse widths are determined when the integrator ramp is greater in magnitude than the control system output, as shown in figure 7.4. [9]

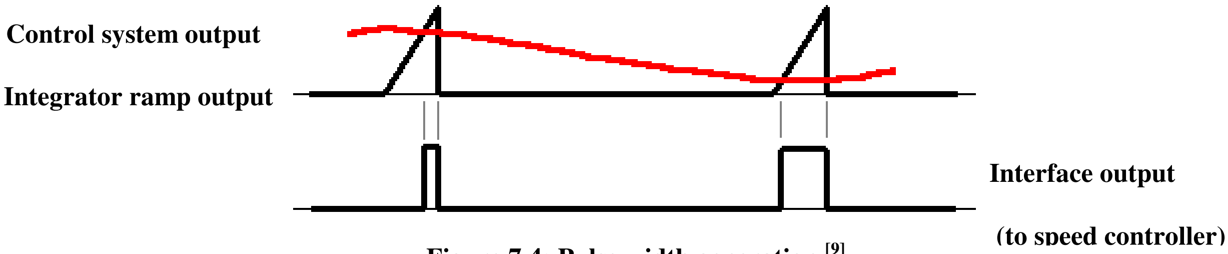


Figure 7.4: Pulse width generation [9]

The ramps were then separated into 20ms intervals using an astable 555 timer and a monostable. The 555 produces a square wave with a period of 20ms, and a duty cycle of approximately 50%. This triggers the monostable to generate a pulse <2ms. This set up was used last year as the 555 timer cannot produce duty cycles of <50%, and the monostable needs to have completed the 2ms pulse before the reset pulse is sent otherwise the system saturates at 5V.

The previous group utilised pull up and pull down switches to give the start up signals to the controller. This appears to have worked for them, but was not foolproof, and it was possible to start up the motors on full power as described in section 6.ii. The safety device on the platform also made use of pull down networks, effectively telling the device to operate at minimum speed, or in this case 0%.

7.iv QBasic™ program construction and use

In order to test the stability propulsion ducted fans, a quick and easy to operate and control circuit was required. There was also the specification of being able to accurately control the percentage of power being requested of it, i.e. the PWM signal being fed in. The dynamic testing required by the propulsion group required the interface to be able to step between two distinct ratings, and to give a timing output at the point the controller was informed of the change. These specifications pointed the group in the direction of a PC controlled system.

The simplest control output to utilise from a PC was the parallel port, which under Windows 98™ can be directly controlled from the relative registers. The system was designed so that the PWM signal was sent on Data bit 0 (pin 2 on a SPP parallel port), and the timing output on Data bit 1 (pin 3). Pin 25 is used as the ground connection. Each data output of the parallel port is designed to sink or source no more than 20mA. Given that the timing output will only be connected to an oscilloscope, and the PWM output will feed the TTL logic levels on the controller, this limit should not be exceeded under normal conditions. The output buffer for PC LPT1 is Hex0378, input buffer is Hex0379, and the control buffer is Hex037A. [10]

Full QBasic™ program listing, together with instructions on the program use is given in appendix 3.

It was found to be simpler, and quicker to produce the program, if rather than use the system timer for accurate timings, pulse widths were produced using delay routines, calibrated with an oscilloscope. The timings for the delay were calculated by trial and error, and utilised a variable for the full cycle period. The variable for the on time duration was set to be from 0 to 100, so that it could be used as a simple percentage, but was adjusted in software to give 1 to 2ms. This was done by trial and error again, until both 1ms and 2ms was found.

```
newlevel=350+level+level+level+level
```

(Figure 7.5)

The QBasic™ program entry given as figure 7.5, sets up the on time of the cycle to be four times the percentage added to 350 fixed duration. This gives just under 1ms for 0% and just over 2ms for 100%. The actual 'on' cycle variable is called '`oncyc`' and is different to '`newlevel`' to allow for the timing trigger to be activated when the two are not the same. The 'off' cycle time '`offcyc`' is then calculated by deducting the on time from the full cycle time.

The keyboard is the main interface between the user and the interface. There are presets for **S**top, **F**ull, **I**dle, and **P**reset, giving 0%, 100%, 10%, and 65% respectively. The system also accepts "+" for step up by large interval, ">" for step up by small interval, "-" for step down large interval, and "<" for step down small interval. Each interval is predefined by variables at the program start up using registers '`laint`' and '`smint`'.

The program restricts the level to be 0-100% by reading the level before passing through the control system, and if it is found to be outside the limits, the level is amended to be 0 or 100% as appropriate.

The on pulse is controlled by a loop, which incorporates an entry to output the timing pulse if required; this section of code is given as figure 7.6. If the trigger has been called, then a different integer is sent to the port buffer placing both data 0, and 1 lines high for the duration, while if it has not been called, only Data line 0 is placed high. The loop repeats for '`oncyc`' repetitions, and is placed on the port buffer at memory location Hex0378 reserved for LPT1. The loop controller '`i`' is reset to 0 after the loop to ensure correct operation for the off loop.

```
IF triger = 1 GOTO trigloop
'----------On loop
FOR i = i TO oncyc
OUT &H378, 1
NEXT
i = 0
GOTO offloop
trigloop:
FOR i = i TO oncyc
OUT &H378, 3
NEXT
i = 0
```

(Figure 7.6)

The off loop is similar to the first, but includes a screen output to allow the user to know what power level the controller is receiving. This output drains time from the PC and so is additional dwell time for the loop, thus reducing the number of repetitions required. The off loop also removes the trigger pulse from the timing line. On the display part of the program, the screen is cleared to remove the scrolling effect that would otherwise be seen. Again the loop counter is reset to 0 after the loop completes. The off loop is given as figure 7.7.

```
'----------Off Loop
offloop:
FOR i = i TO offcyc
OUT &H378, 0
NEXT
i = 0

'-------------------Print output level to screen and restart
CLS
PRINT level; "%"
```

(Figure 7.7)

```
'---------------------Shut down routine
shutdown:
CLS
PRINT "Shutting down motor, making safe"
g = 0
reshutdown:
FOR i = i TO 350
OUT &H378, 1
NEXT
i = 0
FOR i = i TO 6900
OUT &H378, 0
NEXT
i = 0
g = g + 1
IF g = 50 THEN END ELSE GOTO reshutdown
```

(Figure 7.8)

The shutdown procedure for the program, ‘S’ on the keyboard, ensures that the motor is placed in the stop position before closing. This part of the code, given as figure 7.8, replicates the main program but repeats the 0% command 50 times, before **end**ing the program.

Although QBasic™ allows for an Executable file to be produced, but this affects the timings. This is due to the QBasic program itself taking up operating processor time, so without it running, the EXE file will run faster, and require longer delay loops. It was therefore left as a QBasic™ file and a short batch file written to open the both the program, and the file. This batch file is shown in figure 7.9.

```
cd c:\qbasic\
qb.exe /run platform\fulltest.bas
```

(Figure 7.9)

An additional problem was found that the motors appeared to be hunting, and unable to keep a constant speed. However this was later found to be the unreliability of the Windows™ operating system, giving differing amounts of processing time to the QBasic™ file, and thus depending on which point in the cycle the delay came, could affect the length of the on pulse, and thus the overall speed of the motor.

7.v Microchip™ PIC16F873 design and build

7.v.a Introduction

It was considered to replace the analogue electronics involved in the interface with a microprocessor-controlled circuit. This would have the effect of reducing the number of components from over 100, to around 10.

The PIC16F87x series was selected from the large range of PIC devices manufactured by Microchip™. These devices have up to 8k x 14 words of FLASH program memory, up to 368 x 8 bytes of data memory, and up to 256 x 8 bytes of EEPROM data memory. These devices offer a number of additional features including:

- Direct, indirect and relative addressing modes,
- Power on reset,
- Power up timer,
- Programmable code protection,
- Selectable oscillator options,
- In-circuit serial programming™ (ICSP) via two pins
- Wide operating range, 2.2-5.5V,
- Sink/Source current 25mA,
- Universal Synchronous Asynchronous Receiver Transmitter (USART),
- Up to 8 10bit A to D converters,
- Up to 33 digital I/O lines,
- PWM outputs,
- Low power consumption, <0.6mA typical @3V, 4MHz [11]

The Microchip™ PIC range has the advantage that there are only 35 single word instructions to learn for programming. Having given consideration to the amount of A to D converters required, the cost of the individual items, and the amount of memory required, it was decided to go with the lower power 20MHz version of the PIC16F873.

7.v.b Method

Although compilers are available in C and a number of other programming languages, Alex Tombling, appointed by the group to look after all PIC programs, preferred to write in assembly language. In order to program the PIC a programmer was required, and an assembler. Microchip™ offer freeware software “MPASM™” which takes assembly language files and converts them to a .HEX file for directly programming into a PIC. This software is available online. [12] A programmer was constructed labelled AT001, and given in appendix 4 as figure 14.4.1, and pictured as figure 14.4.2. This programmer was called the JDM programmer and sourced from the Internet. [13] The programming software was sourced from the same location on the Internet and was found to be very reliable and user friendly.

In order to reduce the time taken to reprogram the devices, the USART was used together with a MAX232 Integrated Circuit to directly connect to the serial port of the programming PC using a PIC downloader. This required the target device to have been pre-programmed using the standard programmer with a bootloader. These were both sourced from the Internet. [14]

The Bootloader was programmed into the PIC originally using the IC Programmer.

The programs for the PIC are written in assembly language in a standard text editor such as Microsoft WordPad™. This was then assembled using MPASM, and directly sent into the PIC by way of the Downloader. This is a very simple window platform, which requires the filename to be entered into the program, and "Download" operated. This then starts to communicate with the target board, and awaits the PIC to be reset. Once this is reset it will start to send the new program to the target chip, and when complete will acknowledge completion, and the PIC will start running the new program.

7.v.c Circuit design

There are very few components required to operate a PIC. For basic operation only 4 external components are required. There is an oscillator and a pair of 22pF capacitors to start oscillation. The master clear connection needs to be tied high normally, and in order to allow the clear to be used, a 47kOhm resister should pull this input high. In addition to the basic circuitry there is the bootloader components for the USART, the MAX232, and five associated 100nF capacitors. Each of these capacitor values was chosen following guidance from the appropriate datasheets. Most of the I/O ports on the PIC are of the open collector variant, and so are either open circuit or connected to ground. Thus each of the PWM outputs need to be pulled high, and this is done using a DIL package of 4.7kOhm resisters, with similar resister package used to pull high the digital control inputs. Each PWM output is group with a +5V supply and ground connection to power the servo for the throttles, and for references for the speed controller.

Care must be exercised when building this circuit to ensure that the oscillator connections are kept as short as possible and any spare track removed, due to the additional capacitance that exists between the tracks and will cause problems. This was one fault that was found during initial testing of the circuit.

The full interface circuit diagram can be found in appendix 4 as figure 14.4.3, and the relevant features of the circuit are shown on figure 7.10.

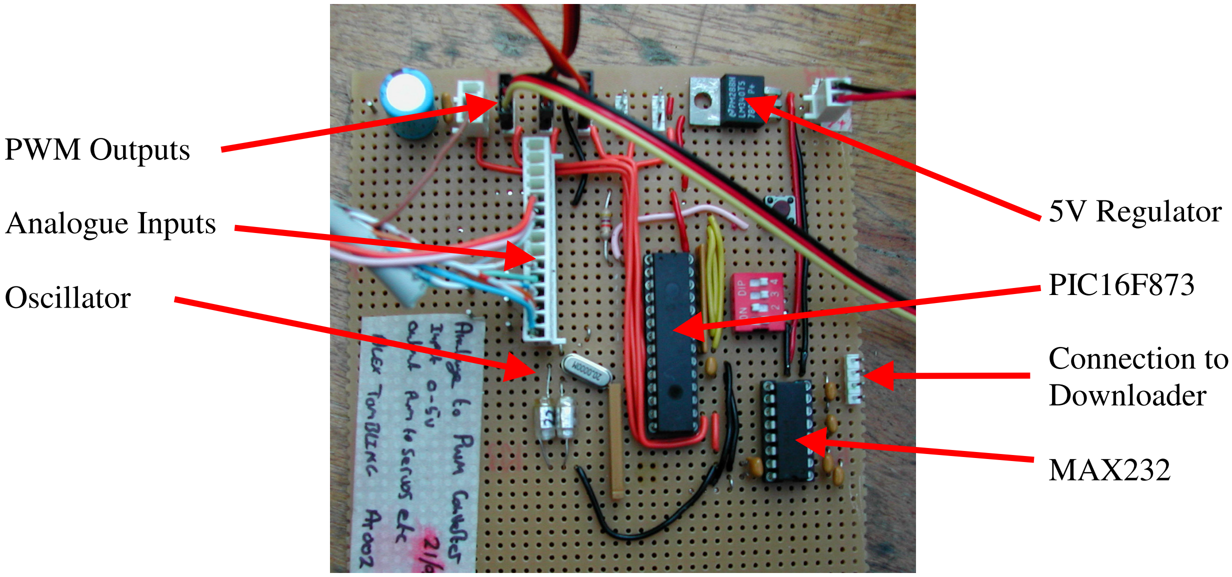


Figure 7.10: Circuit features of AT002

In order to replicate the control system so that the speed controller may be used with known power levels, and so that the throttles on the IC engine tests can be controlled, a separate circuit was built, AT003. The circuit diagram for this is given in appendix 4, as figure 14.4.4, and detail of the connections given in figure 7.11.

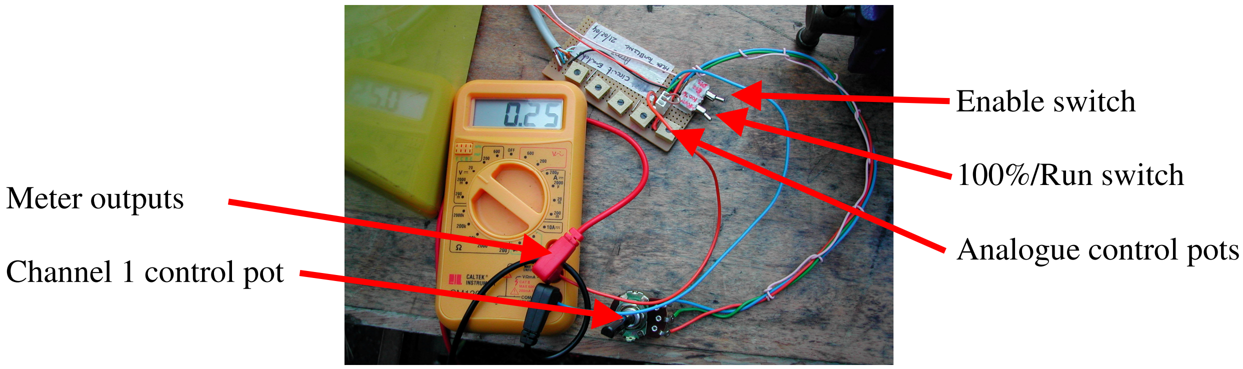


Figure 7.11: Circuit features of AT003

In order to simplify operation of the control system emulator (AT003), an additional pot was fitted which was easier to use than the preset variety. A selector jumper was fitted to enable the user to choose which control method to use for channel 1. Additionally wires were connected to the board to link to a Digital Multi Meter to enable the user to accurately see what power setting was being fed into the interface board. The meter read between 0 and 5V for 0 to 100% respectively.

7.v.d Program code

The complete listing of all 35 commands for the PIC is given in Appendix 4. All source codes to be assembled with MPASM™ must have filenames of less than eight characters plus the extension .ASM.

This program was written to operate at 20MHz, and with a bootloader. The software version number was stored in the program source code as a remark in the program header. The final version was ver3.2, but required small amendments as described in section 7.v.e. Once all registers, variables and constants are declared, the program starts with some bootloader vectors. This is so when the PIC is reset, it checks for a new program being loaded. In the event one is not found, the processor returns to the start of the user program. The first part of the user program sets up the ports so that port A operates in analogue input mode, Port B as digital inputs, and port C as digital outputs.

This interface can only cope with 5 PWM outputs and respective analogue control inputs. This prevents all the necessary controls being interfaced on the one board, but allows all propulsion connections to be achieved together. It was considered that extra control may be required to govern the generator IC engine, respective of the load placed upon it, and this could incorporate its own interface and directly drive the servo for the throttle control.

The program then starts by checking for the primary set up feed. This input on Port B bit 0, is initially high, which sends the program into a loop placing all outputs on the PWM connections of Port C to be at 2ms every 20ms (100%). This subroutine called ‘`allhi`’ is separate from the normal operating subroutine, and is timed uniquely as there are no analogue conversions to take place.

Each pulse cycle is identical in principle. Firstly the output of one PWM connection is placed high and an integer value is loaded into the timer level register ‘`timlev`’, and this is passed into the delay subroutine. This subroutine then delays for 1ms, then delays for a set period controlled by the timer level register. This register is set so that Hex 'FF' creates a delay of 1ms, and Hex '00' creates a delay of 0ms. The delay subroutine then exits, and the timer level is deducted from Hex 'FF' to give the inverted level. The PWM output is the placed low, and the delay subroutine called again, delaying for 1ms, followed by a delay described by the timer delay register. This makes each PWM routine 3ms in length. Once one output is completed, the next is performed in an identical way. Once all 5 outputs have been completed a delay of 5ms is required to complete the 20ms cycle.

Once the primary set up feed is taken low, the device sends out all PWM signals at 0% for 2seconds to ensure all speed controllers complete the start up routine. The enable input (Port B bit 1) is then read, and if high, all outputs remain at 0% and effectively disabled. Once this input is placed low, the interface is enabled. The enable input is used by the radio control safety device to switch off all inputs in an emergency and make the platform safe, as it also reduces the throttle controls to 0% and should shut down both IC engines. It was intended that there would also be a Idle input on Port B bit 2, which would set all speed controllers to 10% so that the ducted fans could cool the speed controllers, and place the IC servos to a point where each IC engine would tick over without stalling. This feature was not implemented on the current version but it is hoped to add this feature at a later date.

Once the interface is enabled, the program routine has to convert all the analogue inputs and calculate the pulse width to be sent to the controllers. The first stage of this process is to set the analogue converter to read the appropriate channel and call a 1ms delay to allow the voltage to stabilise within the conversion module inside the PIC. Then conversion is started with the '`GO_DONE`' flag on the AtoD converter. This flag is then cleared when conversion is complete; and is an appropriate method of waiting for completion. The converted level is 10bits in length and is spread over two locations, and can be justified left or right. It was selected to be left justified so that the least significant bits are ignored. This value was then read into the delay routine as described above giving a single cycle time of approximately 4ms, due to the analogue conversion delays. The total cycle for all 5 conversions is therefore just short 20ms, and only a small additional delay is required.

Once the process is completed, it returns to check the enable input and repeats the process. The primary set up input is not rechecked and is ignored on all future repetitions.

A full set of operating instructions are given as appendix 7.

7.v.e Conclusion

The PIC16F873 circuit worked reliably, and was subjected to mechanical stresses, and chemical ingress without any noticeable decay in operation. It was used on the test bench with both IC engines subjecting it to an extreme amount of vibration, and covered in fuel and oil from the exhaust on a number of occasions. The connectors for the servos etc. were subjected to mechanical strain when servos were disconnected, causing them to be bent in various directions. The battery was connected incorrectly on a number of occasions, as were the servos, all with no damage resulting. The outputs were found to be very stable and reliable giving consistent results for both servo operation and speed controller operation.

The circuit could have been improved with the addition of an idle input as previously mentioned. This would have involved a new subroutine where the values of 'on' time and 'off' time were preset as with the 0% and 100% functions. There is sufficient space within the program memory to add this feature, it was only time that prevented it from being completed before the termination of the project this year. The PIC16F87x series was found to be very versatile and more than adequate for this purpose and is recommended to all future groups for interfacing the servos and speed controllers.

7.vi Conclusion

The analogue electronics method was rejected at an early stage due to the complexities in setting up such a circuit, and its large component count. It was also considered that with so much fuel and oil being in the operating environment this may effect the resistance and capacitance values enough to have a large effect on the operating characteristics.

The QBasic™ program was inherently unreliable due to the methods used for timing the pulse durations. By using a Windows™ operating system, the operator could not guarantee the length of pulse, or the cycle duration, and on a number of occasions the speed controller cut out activated, as it believed it had lost the incoming signal. This system was a quick fix for a dilemma but should not be relied upon for future use. However it did have the advantage of giving a timing reference pulse when the power rating was changed.

The best interface by far was the PIC16F873 controlled device. This gave reliable repeatable results, and was able to stand up to the environment it was placed into. It is the belief of the author that this interface could be adapted to provide a timing reference pulse as the QBasic™ version did, giving a higher degree of accuracy for dynamic testing of the ducted fans, however step changing from one level to another could be more challenging with the current circuit.

8. Propulsion System Testing

8.i Stability Fans

8.i.a Introduction

The outer fans were tested using the QBasic™ speed controller interface. Tests were carried out to find out the relationships between thrust and pulse widths, and many other attributes.

The tests were carried out using the test rig designed and constructed by the previous years' group. The previous group completed a number of tests on the outer-ducted fan units, but a number of group members suspected that their results were inaccurate. The group decided, therefore, to retest one ducted fan for static performance, and for dynamic response.

8.i.b Theory

Before testing could commence, an interface had to be constructed to control the speed controllers, and was discussed fully in section 7. During the test a number of variables were recoded, including rotational speed, thrust output, power in, and pulse width supplied. The pulse width was immediately available through the controlling interface, and the thrust output was calculated using a set of scales, and effect of moments due to the length of the rig arm. Full details of this test jig can be found in the Hughes report. [16] The power in was calculated from the voltage across the battery power supply multiplied by the current drawn from the battery, measured with a clamp on Ammeter. The rotational speed was more difficult to measure due to its high speed.

The previous group measured this using a monostable circuit, which was modified by Rebecca Hughes and labelled as RCH001. This circuit is given in Appendix 5. The maximum speed expected from the ducted fan unit was 33,000rpm, and so a maximum speed for the circuit was given as 35,000rpm to give a safety margin. Each fan had 6 blades, giving 210,000 blades per min, or 3,500 blades per second. It was decided to use a photo diode, and an infrared LED mounted either side of the fan blades to detect the blades movement. Every time the blade passed the sensor, the light beam between the two would be broken, and the RCH001 circuit would register this. The variables would need to be adjusted to cope with this speed of signal. 3,500 blades per second, is the same as a 3.5kHz square wave, with a small amount of distortion on the positive going edge. The time constant for the monostable is given in equation 8.1.

$$T=R\,.C \tag{8.1}$$

The time constant, calculated by 1 over frequency, is 285.7us and with a 1uF capacitor, the required resistor is 286kOhms. This value is not available, the nearest being 270kOhm, and so this value was chosen. Under test conditions with a perfect square wave generated from a signal generator gave a voltage output of 4.85V. Conversion from Hz input signal to rpm is given by signal multiplied by 10. Full details on the origins of this circuit can be found in the Tulloch report from the 2002-03 project group. [17]

The motor temperature was monitored continuously during all the testing to ensure it did not breach 100°C. At temperatures beyond100°C the insulating coating begins to breakdown, and the motor may become irrecoverably damaged. The motor uses neodymium magnets, which begin to demagnetise at temperatures above 150°C, and so must obviously be avoided.

The response times of the motors were to be recorded. These were measured on a digital storage 'scope. When the controller was passed a new pulse width from the interface, a timing reference pulse was sent on a separate line from the PC. The output from the monostable speed detector circuit was placed on the second channel of the 'scope. The storage capability was activated on the positive edge of the reference pulse, and one screen width recorded.

8.i.c Results

1Kg of thrust was expected to be required from each of the stability fans. This was reached with a rotational speed of 21,600rpm, with a pulse width of 1.6ms (60% level). This speed drew 16.8A, with a total electrical power of 408W. Table 14.5.1 in appendix 5 shows all the results from the static stability fan tests. All figures referred to below can be found in appendix 5.

Figure 14.5.1 shows the thrust output against the pulse width fed into the speed controllers.

Figure 14.5.2 shows the total electrical power required to produce each thrust level.

Figure 14.5.3 shows the thrust against current drawn from the power supply at an approximate steady 24V continuous.

Figure 14.5.4 shows the temperature of the Plettenberg HP220/20/A4 S P4 when run at a constant 60% pulse width level.

It should be noted that the temperature increased significantly when the fan was switched off rising 25^{o}C. It was therefore advised that the fans be run at 10% idling to pass an airflow over the motor and controller after use for a period of time to cool them down without placing excessive currents through them. When power is reduced from 60% to 10%, the temperature increases by 10^{o}C, but quickly returns to its previous temperature. After seven minutes of idling, the temperature is below 40^{o}C. It was therefore recommended that the controllers be idled for 5minutes after use to cool off.

The peak electrical current drawn was also measured when the fans were increased from 10% to 65%, and from 60% to 70%. The maximum current drawn in both cases was 28A. Dynamic tests were also carried out for response times. For each 10% step change, both positive and negative, response times of 0.3s were noted, while for a 5% step change, response times of 0.14s were noted.

8.i.d Analysis

The ducted fans using the brushless Plettenberg motor gives a very fast response time suitable for the stability of the platform, as the platform time constant is expected to be longer. It is unlikely that a step response would be required on the platform and a smoother response would be expected linear to the angular rotation of the platform. This type of response would be very difficult to investigate without additional expensive equipment, and therefore the step response was used to test this system.

The pulse widths cycle every 20ms, and so the main factor in the time response will be this delay. If the call for a change in power is received just after the signal had been converted, it will be 20ms before a response is sent to the controller. This must be taken into account when designing the control system, and cannot be reduced in any way without different controllers specific for this purpose, as most will comply with radio control standards.

8.i.e Conclusion

From the test results it was concluded that the fans should not be taken above 27,000 rpm due to the limitations of the WeMoTec Midifan. This means that the maximum PWM level is to be 80%, with a maximum current draw of 35A for each motor. At this speed, 1.6Kg of thrust should be possible. It is anticipated that the outer fans will run at 1Kg of thrust normally, and be adjusted up or down from this point to achieve stability. This gives a normal power take off of 68A @24V. It may be possible to incorporate height control into the outer fans, as they will be easier to control than the central IC fan.

8.ii Central Ducted Fans

8.ii.a Introduction

In the previous years' project the central fan was an electric operated ducted fan identical to the outer stability fans. It was found that these fans were not capable of lifting the platform if it had to carry. It was recommended that this year's group consider the use of an IC engine with a larger ducted fan.

Table 8.1 shows a comparison of various IC engines available for use with the central fan.

Man'	Model Number	Displacement (cc)	RPM (1000's)	Weight (g)	Power		Cost
					BHP	kW	
Saito	FA-120S/FA-120SGK	20	2 to 11	840	2.2	1.6	300
Saito	FA-180/FA-180GK	29	2 to 10.5	900	2.8	2.1	410
Moki		22 - 90	8 to 12	0.8 to 2.9	3.5 to 7.5	2.6 to 5.6	299 - 675
Irvine	120 ring	19.82 (1.2 cubic inches)	1.7 to 12	1020	2.35	1.8	
Irvine	150 ring	1.5 cubic inches			2.55	1.9	
OS	.70 SZ-H Ring 3D heli	11.5 cc	2 to 18	553.5g	2.5	1.9	
OS	BGX-1 3500	35cc	1.5 to 10	1340	4.1	3.1	350
OS	1.4 RX-FI	23	1.8 to 10	807	3.5	2.6	
OS	os FT-300 (TWIN)	24.38 *2	1.8 to 9	1828	4	3.0	900
OS	OSMAX 91 FX	14.95	2 to 16	550	2.8	2.1	190
OS	OSMAX 160FX FI	26.23	1.8 to 9	945	3.7	2.8	700
OS	OS Max 46VX-DF	7.45	2.5-28	462	2.5	1.9	250
OS	OS MAX 91 SX-H C-Sped	14.95	2 to 16	547	3	2.2	270
Super Tigr	G2300	23.21	?	877	3.7	2.8	125
Rossi	Aero 60	10	17	590	2.5	1.9	120-170

Table 8.1: IC engine comparison

Full details on the engine selection can be found in the Holbrook report. [18]

The OS MAX 91 was selected, and due to its imminent cessation of production, two were purchased. This engine was capable of 25,000rpm, and had a capacity of 15cc. The ducted fan purchased at the same time claimed to produce a thrust of 6Kg at 16,000rpm.

8.ii.b Theory

Testing of the ducted fan was required in order to confirm the maximum thrust possible. The head temperature should also be monitored to ensure that it does not overheat causing possible distortion of the head, and premature failure of the glow plug.

Before testing can take place the engine should be run in, so that all the moving parts are free to move before the engine revs are increased towards the upper limit.

8.ii.c Method

Running in the IC engine was completed with a 12" diameter propeller with a well proportioned hub as per the instructions from OS. [19] The OS 91 engine utilised an O.S. No 8 glow plug, and a fuel containing 10% nitromethane, increasing to 16% for extra performance. The fuel contained at least 18% lubricant, of which was mainly caster oil. The natural oil was preferred over synthetic as it had a higher temperature point in which it started to breakdown. The running in period totalled 20minutes, in short 5minute bursts. The needle valve started at one and a half turns from fully closed, and gradually adjusted for maximum performance.

Once run in was completed, the ducted fan was attached to the IC engine, and all settings re-adjusted to improve performance. The engine was then further run in for 10minutes in the horizontal position before rotating ready for testing.

The tests consisted of running the IC engine at preset throttle settings, and measuring the thrust produced, and the rotational speed of the fan. The detail of the test jigs, and procedures are detailed in the Holbrook report. [20] The IC engine had to run with a tuned exhaust pipe to improve performance and to reduce noise.

The rotational speed was recorded with a commercially available strobe used for model aircraft propellers.

Due to the limited test area within the University of Exeter, testing took place outside the rear loading bay of Engineering. This is due to the fact that Computer Science now occupies the old Engineering engine bay.

8.ii.d Results & Analysis

Full results together with graphical representations can be found in appendix 5. Full details can again be found in the Holbrook report. [18]

It was found that the ducted fan was able to produce more lift with a small duct on top of the fan to direct the air into the main unit. It also appeared to be necessary to fit a short duct under the fan unit to direct airflow over the head of the IC engine for cooling purposes. During testing, a problem materialised with the head of the engine, and so a new larger head was purchased. This new head had more surface area for cooling, and resulted in extending the life span of the glow plugs together with a reduction in overall outside temperature.

The maximum thrust achieved from the IC ducted fan was 5.1Kg at a rotational speed of 20,000rpm. The maximum speed recorded was 20,800rpm. Table 14.5.2 in appendix 5, shows the weights expected to be carried on the platform this year. It can be seen from this that with the 4 Kg expected from the outer fans, and a maximum of 5Kg from the central duct, a total of 9Kg is available to lift the platform.

8.ii.e Conclusion

Given that there is only 9Kg of lift available maximum, and it is predicted that the platform will weigh in excess of 10Kg, it is not believed to be enough to fly the platform with all the electronics associated with the generator. Figure 14.5.5 in appendix 5 show the Thrust IC under test.

9. Electrical Power Systems

9.i Introduction

In the previous years projects the power supply utilised was a large capacity battery on the end of an umbilical cable. This prevented the platform from being truly autonomous. This year it was intended to produce all electrical power on board. A small amount of power will be available for starting up procedures via an umbilical from the ground rig. Once systems are active and running, the cable can be removed, and the platform becomes totally autonomous.

There are a number of different power supplies required by the various control systems throughout the platform.

9.ii Generator Set Research & Selection

9.ii.a Introduction

The generator intended for this platform was to consist of a small IC engine, with a direct drive to a small electric motor. This motor needs to produce a DC stabilised voltage at greater than 24V. It needs to be able to supply enough current for the 4 stabiliser ducted fans, and all on-board electronics. From section 8, this is believed to be approximately 68A, with an additional load of approximately 5A for control systems, giving a total of 73A @24V. If the voltage is increased to 36V, it is believed this current draw will be reduced to around 50A.

9.ii.b Theory

There are two alternatives to producing DC power, either by utilising a DC generator, or an AC generator with appropriate rectification. In either case electronics are required to smooth the waveform produced and to take account of any surges, or outages.

Power is to be sourced from an IC engine directly coupled to an electrical generator. In order to calculate the maximum power obtainable from a generator, the output will be loaded with a resistive load manufactured from a length of resistance wire.

There were a number of larger generators available, but all were far too heavy for the application. Small motors were investigated to discover how much power could be generated when used as a dynamo.

9.ii.c Method

Initial tests were carried out using the Plettenberg HP220/20/A4 motor directly coupled to a second motor under test. The electrical power fed into the primary motor was recorded and compared to the electrical power obtained from the tested motor.

Two motors were tested, a second Plettenberg HP220/20/A4, and a Graupner Speed 600 BB Turbo 12v. These were placed on a test rig produced by Richard Holbrook, and fully detailed in the Holbrook report. [19] Each motor was rotated at speeds controlled by the Schulze controller, and the output loaded with a variable resistance below 5Ohms. The variable pot was rated at 10A, so currents above this were to cause problems.

Care was exercised to ensure the maximum speed of each motor type was not exceeded.

9.ii.d Results

A number of electric motors were rejected due to weight considerations before testing commenced. These included the EMD PM50/38ew rated at 12A @ 5,000rpm but weighed 1.25Kg. [21] A 2.2Kw alternator was sourced from A Davis & Sons Ltd, but costing £165+VAT, and weighing in excess of 2Kg, and a GR6317 12V 700 motor was considered but again was too heavy. [22] Consideration was also given to Dyno-starters.

Full test results and graphs can be found in appendix 6. All figures referred to below can be found in appendix 6.

Figure 14.6.1 shows that the 3 phase HP220/20/A4 can give 90% efficiency when running with a 1Ohm load, whereas figure 14.6.2 shows the 12V DC motor can only give a maximum of 40% efficiency.

It was found that three-phase generators are approximately 150% more efficient than single-phase units. The Plettenberg motors used for the ducted fans were found to be highly efficient with a relatively high power output. Research was conducted and found that Plettenberg manufactured larger models with increased power output. The Plettenberg HP370/30/A2 S was recommended by the manufacturers to meet the project specification, with an anticipated output of 1.4Kw at 13,000rpm. [23] This was obtained.

Tests proved that the small Plettenberg motor was capable of producing 19.3A at 36,000rpm, in excess of the capacity of the test load. This created overheating in the test load, and due to the length of time the device had been disused in storage considerable smoke was emitted.

The new Plettenberg HP370/30/A2 was tested in the same manner, driven by the HP220/20/A4. Unfortunately the small motor was not powerful enough to turn the larger motor under load, and testing had to be aborted.

The new motor was connected directly to a IC engine. The IC engine used was the second OS91 purchased as a spare thrust engine, but was able to meet the specifications of the Genset. [19] A coupling, and flywheel was developed in order to connect the two together. The test rig for the loading of the motor was made from resistance wire. The motor was connected to a rectifier constructed from 60A diodes and pictured in figure 9.1. This rectifier was connected to a 5Ohm load, and a voltmeter attached at this point. This same point was connected to a connector to allow smaller loads to be attached. These loads were 2Ohm, 1Ohm, ½Ohm, and ¼Ohm. The test rig was permanently connected to a load to reduce the risk of the IC engine over revving when switching between loads.

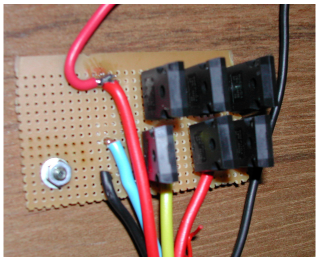


Figure 9.1:60A 3-Phase Rectifier

Testing took place outside the rear of Engineering at Exeter University, during fixed times to reduce the disturbance to Computer Science. The Test rig is pictured in Figure 9.2, and the overheating resistance wire is shown in figure 9.3.

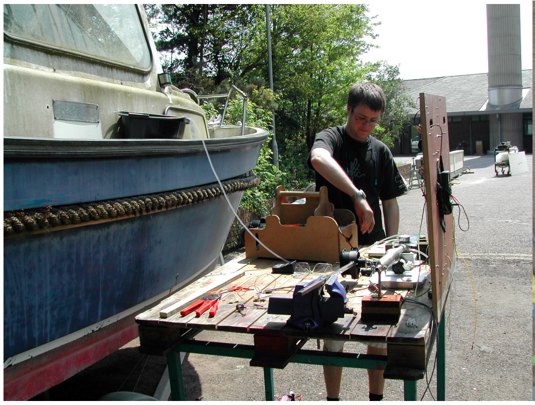


Figure 9.2: Genset Testing

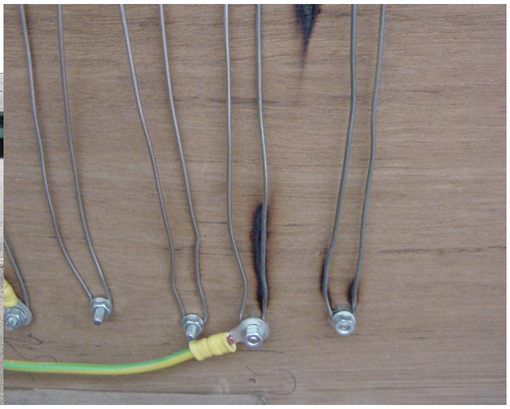


Figure 9.3: Resistance wire overheating

Following overheating, it was believed that the rectifier may have failed due to the excessive currents flowing from the generator. A new 120A rectifier was constructed, but this weighed in at 1.5Kg when finished, due to the heat sinks fitted, and the large components used. The Mackenzie-Burrows [24] report detailed the research and construction information for the second rectifier circuit, and is pictured as figure 9.4.

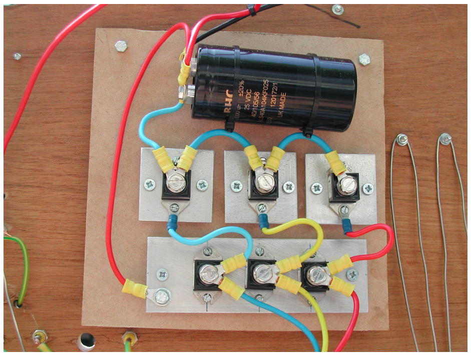


Figure 9.4:120A 3-Phase Rectifier

At one point during testing the flywheel became detached from the engine shaft and was thrown clear of the Genset. This caused a great deal of concern due to the energy that was contained in the flywheel, which took ½ minute before stopping rotating after leaving the shaft. During this failure, the main drive shaft also became bent, causing significant catastrophic failure of the engine. In order for testing to resume as soon as possible, a new engine was sourced, but was water-cooled, making it impossible to be flown this year. However, the new engine remained cool throughout all subsequent tests, and the glow plug, which had previously expired after short runs, lasted for the complete duration of all subsequent tests, proving the water-cooling to be very effective.

A full table of results can be found in appendix 6 as table 14.6.3.

Although it was believed that 60A had been exceeded, it has not been possible to repeat this output. It has been possible to produce 20.5V across a ½Ohm load, giving 41A.A total power output of over 800W. Further investigations are required, but are outside the time limitations of this project.

9.ii.e Analysis

The Plettenberg HP370/30/A2 was deemed the best small motor to be used for power generation, and should be capable of producing in excess of 1.5Kw. However, it was not possible to achieve this with the set up used. A number of flexible couplings were broken during tests due to the excessive torques placed upon the coupling.

A photo of the completed Genset including IC engine, coupling, and electric motor is given as figure 9.5.

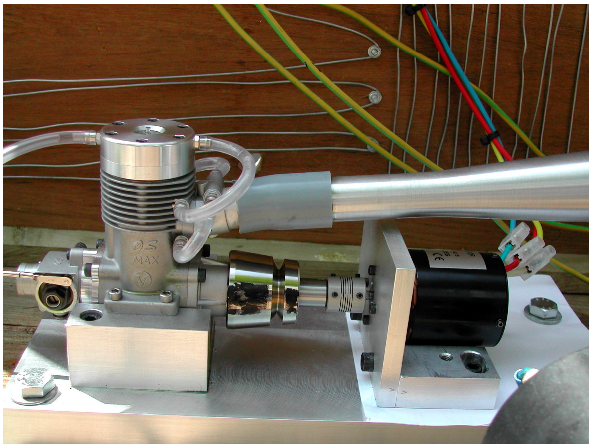


Figure 9.5: Genset with water-cooled head

9.ii.f Conclusion

The onboard generator is a feasible design, which needs fine-tuning. The prospect of a water-cooled onboard system is plausible, and should not be dismissed by the next research effort without full investigation. The one major factor that caused most problems was the coupling between the IC engine, and electric motor. Mechanical time needs to be spent here to produce a strong reliable coupling, and aligned so that the flexible coupling is no longer required.

9.iii Platform Power Requirements

The platform required a number of different power supplies, including the 24V high power feeds for the speed controllers for outer stability. These included:

- +5V for the control system
- +8V for the IMU
- +15V for control systems
- -15V for control systems
- Ground

These different users required different power levels, and it was very difficult to find this information out from the group members. The different interfaces and know power requirements were given in appendix 2.

The Genset was planned to give an output of 24V, and this was to be used by a power control system to provide all positive voltages for the platform. However, negative voltages were more problematic. It was believed possible to increase the voltage from the Genset to 36V, and create a false ground by way of a summing circuit. This would allow for all required voltages to be derived from the supplied power. The independent voltages were to be produced with a fixed voltage regulator, but none were available with a high current capacity. A circuit was therefore developed using a LM370 variable voltage regulator, controlled by a resister network.

9.iv Analysis

The electrical power system developed for this platform was untested due to the time constraints and late running of the Genset development. What testing was carried proved successful, and that the main regulation system would function correctly, if supplied with enough power. The circuitry was in total very heavy, and needs to be reduced in order to fly in the future.

9.v Conclusion

This system can be made to function well, but more investigative research is required in to producing a light weight power supply system, and improved power generation from the Plettenberg HP370/30/A2.

Water-cooling should not be totally discounted, without proper investigation.

10. Radio Control Systems

10.i Introduction

There are a number of different methods of communicating with the platform while in flight. While most revolve around radio frequency transmission, there is also infrared transmission, and high frequency satellite transmission. Satellite transmission is inappropriate for this project due to the extensive cost of licences, and necessary equipment. The additional weight required for such a transmission network also precludes its use. Infrared would be a cheap and simple transmission method for test purposes, but would be inappropriate once the platform was out of line-of-sight from the controlling transmitter. The Infrared method also requires a narrow beam to be accurately aligned, which would be impossible to operate with the natural yaw moment of the platform. Radio frequency is the natural choice for communications to and from the platform while in flight.

Start up routines may be controlled through an umbilical cable from the ground starter rig, which would also supply power for control systems prior to starting of the on board generator.

10.ii Theory

Radio is most familiar in the application of transmission of speech and music by ‘stations’ and the reception of these signals on ‘receivers’. This type of system is called a broadcast system, which can be picked up by many receivers. Every home must contain at least one such receiver, often with the user giving little thought to the myriad of other applications of the basic principles that exist.

The basic principles of the broadcast system involves, the data or speech to be transmitted is sent to a transmitter, and sent into space. At this point the signals are relatively high powered and in simple terms radiate in all directions from the transmitter aerial. The receiver, which may be placed some distance from the transmitter, has an aerial which picks up a tiny portion of the transmitted signal, and passes it to the radio receiver itself. This has the job of converting the radio signal back into a signal that can be used to drive a loudspeaker, or used by digital processes to decode the initial data.

The platform will utilise a radio data system, where information in data form is passed to and from the platform via wireless means. The platform may be able to send data back to the ground station, with information on the IMU sensor positions or the current pulse widths being sent to the speed controllers and throttle controls.

In the system to be used by the platform, the aerial may be orientated to restrict its transmission power to the area of landing pad, and space above ground. There is no requirement to send signals along ground level.

There are additional uses for radio frequency, including Global Positioning Satellite systems, or satellite navigation systems. Here the distance between the transmitter and receiver is measured and by using the known locations of each transmitter, the physical location of the receiver can be calculated in 3D space. In the GPS system, a number of transmitters are precisely located in space around the earth and each equipped with an atomic clock of extreme accuracy. The transmitters are operating continuously, giving out a coded signal including their location, and the precise time the transmission was made. The receiver, knowing the speed at which radio signals travel, can then calculate the distance between satellite and the receiver. With current technology, a low price receiver can be purchased for around £95, giving accuracies within 20m in both position and altitude. [25]

The transmitter unit generates a sinusoidal waveform at the desired frequency and accepts a signal, which may be analogue or digital in nature, and uses this signal to change the characteristics of the sine wave. This latter process is called modulation. The initial sine wave produced is termed the carrier frequency, and the signal fed in to the transmitter containing the data is termed the modulating signal. [26] The carrier frequencies used on radio systems cover a wide range and some typical uses of different parts of the spectrum are as follows:

- 60KHz – Very low frequency signals for specialised data transmission
- 200KHz – ‘Long wave’ signals such as those transmitted by the BBC World Service
- 1MHz – ‘Medium wave’ signals such as those transmitted by public broadcast services
- 20MHz – ‘Short wave’ signals used by radio amateurs
- 27MHz and 35MHz – Model control transmitters
- 70MHz – Emergency Services use
- 80-108MHz – Frequency Modulated radio transmissions
- 200MHz – Television transmitters
- 3GHz – Satellite based communications systems [27]

The wavelength of a radio signal can be found by dividing 300,000,000 by the signal frequency. As an example, a radio signal of 70MHz has a wavelength of 4.286 meters, and is given as equation 10.1. [28]

$$\begin{aligned}\text{Wavelength} &= 300{,}000{,}000 / 70{,}000{,}000 \\ &= 4.286 \text{ meters}\end{aligned} \qquad (10.1)$$

The majority of radio frequencies have become subject to extensive controls by Government agencies, and in the early days this caused a slowing in the rate of development of radio.

10.iii Development

There are a number of techniques used to transmit data through radio communications including frequency modulation, amplitude modulation, and pulse width modulated signals. In addition there are a number of techniques for ensuring data is received as it was transmitted, and or errors corrected. There is also the methods used to transmit the actual data, for instance ASCII characters are transmitted as an 8-bit code word, normally with a start bit, and stop bit.

In addition to these different variables, there is also the speed in which data is transmitted, which needs to be controlled.

The design and construction of a radio transmitter and receiver is extremely difficult and would be expensive and time consuming to accurately match the pair together operating in a licensed part of the radio spectrum. However, a number of companies have now developed a ready built module that eliminates all the setting up required previously.

Many of these devices operate within the 418MHz, and 433MHz frequency bands, which are available for all users without the need for a licence, as long as they comply with strict standards. These devices incorporate power restrictions, which make them very simple to set up and operate without the need to gain further authorisations. They also incorporate their own error checking techniques to ensure all data received is exactly as sent, but only to a limited level of encryption. Any higher level of correction should be incorporated into external control systems.

The design for the platform is to be based around the 433MHz AM super-regenerative receiver module supplied by RF Solutions Ltd based in Lewes, East Sussex. These modules are readily available from Maplin, at a cost of £9.99 for a matched pair. [25] This is a simple to use pair of devices, where the data stream can be directly fed into the transmitter module, and read from the receiver module with no additional components required for their operation.

The design for the Platform system incorporates a number of controls in order to make the platform safe in the event of an emergency. The main transmitted data contains the enable feed to actually allow the platform to operate, and an idle input to reduce the power of the outer fans to 10% pulse width, and to reduce the IC engine throttle settings to idling speed. There is also a 'land' command which was intended to be connected to the height control system, to reduce the intended flight height to 0, bringing the platform back to ground level. The control system adopted would need to allow for a number of other commands to be sent, allowing future groups to send 'return' commands or details of direction of travel requested of the platform.

In order to encode the information to be sent to the transmitter a number of options were available. Either construct a digital encoding circuit, utilise a PIC to produce a data stream, or use a readily available purpose made device manufactured by Holtek. The Holtek HT12E uses a twelve-bit codeword defined by 12 address/data lines. The first 8-bit code defines the address of the receiver to be contacted, while the remaining 4-bits contain the data to be sent. This allows a number of devices to operate on the 433MHz band, including many car alarm transmitter modules. The 12bit address code must be set the same on both encoder and decoder modules. Encoding is recommended to be via digital control to enable the sending of up to 16 different requests using the 4 data lines. An additional transmit line will need to be operated in order to tell the Holtek device to output the signal to the transmitter module.

In order to read the received signal, a Holtek HT12D is recommended which has outputs for each data line, and a valid transmission line that is placed high for a short period to indicate a valid transmission has been received and decoded. This will be used by the processing PIC to start acting on the received data. The data lines remain latched at the last received state until the next valid transmission is received. The PIC outputs will then be connected to the controller interface to enable the system, or to set the system to idle, and will be connected to the control system to adjust the height control when ready to land.

A basic circuit diagram can be found in appendix 9. As the circuit was never constructed, no pictures are available. However, a previous experiment with this system has proven successful at up to 200meters.

Systems are also now available to directly connect to RS232 systems. This type of transmitter module would be very useful for sending sensor readings back to a ground control PC. This would allow the IMU outputs to be encoded and sent along with the speed controller pulse widths, and possibly temperature of each motor to the ground station. Although it would not be possible to update the information within the same time frame it is made available by the IMU, it is hoped that it is possible to send data every second, which should be more than enough definition for any monitoring requirement. The system would not be intended to control the platform, as this would mean it is not autonomous, just to monitor each control system.

10.iv Analysis

Although the system was never constructed, previous experiments with the Holtek devices, and the RF solutions radio module have proved reliable and very easy to control, and set up.

Although continuous transmission of data on this frequency is not ideal, due to the low power transmissions used under University experimental conditions, this is not believed to be a problem.

A hand controller should contain a toggle switch to enable transmission. The decoder should be set so that as soon as the valid transmission is lost, the enable output sent to the controller interface is stopped, immediately shutting down the platform.

In the previous years' project, a kill switch was utilised to effectively pull out the speed controller main battery power feed. This would not be appropriate this year, as the throttle controls in this instance would become uncontrolled, and will be pulled open by the force from IC drawing air through the carburettor, increasing the revs of all IC engines possible to dangerous levels. In this circumstance the Generator will not be under load, as the stability fans would be stopped, and the IC engine will increase in speed to levels in excess of its design limits.

10.v Conclusion

The use of radio control is the only sensible method of gaining remote control of the platform. The RF Solutions module takes the hard work out oft the design of the RC module, and the Holtek devices simplify the encoding of the data.

11. IMU Emulator circuit

11.i Introduction

Previous research efforts have utilised accelerometers and gyros to detect the platforms location in 3D space, without a great deal of reliability. This year, the group was generously donated an IMU from BAE Systems, in order to gain reliability from the sensors for the control system. The IMU was valued in excess of £15,000, and therefore all electrical systems to be connected to it, had to be rigorously tested to remove any possibility of damage.

The IMU constructed to military specification, with a cut-down software program installed. All required datasheets were given to the University, but strict controls were put in to place to ensure the security of the device. Although copies were made of the IMU documentation, all copies were numbered and logged against the group members' name.

The IMU had a number of digital outputs in which to interface with external control equipment, a clock pulse to synchronise the data into the control system, the data stream, and a sync pulse, in order to acknowledge completion of data transfer for each sensor. In order to use the data given by the IMU, an interface was devised to convert the digital signals into analogue voltages.

The group did not want to connect the IMU directly to the Interface without prior testing to insure no damage could result with the IMU, and so an emulator was devised. Due to the fast speeds involved, a PIC was chosen to provide the signals from the emulator.

11.ii Theory

The IMU outputs all 11 sensor outputs in 11, 24-bit data streams. The data stream consisted of a 4-bit encoded sensor ID, followed by 4 bits of unused data. Then the value of the sensor output in 16bits, using twos-compliment method for negative numbers. Twos-compliment shows that zero is given by Hex 00, and 1 by Hex 01. However –1 is given as Hex FF. Using a 16bit level of accuracy, the largest number possible is Hex 7FFF, while the largest negative number is Hex 8000.

Each clock pulse is 400ns in length, with the data being valid for the positive edge of the clock pulse. There is a sync pulse that is sent after the 24bit of data is transmitted. The sync pulse confirms the arrival of all data and allows the interface to convert the digital values into a useable analogue voltage, and output the value on the correct sensor output.

A photograph of the IMU interface, LD002, is given as figure 11.1.

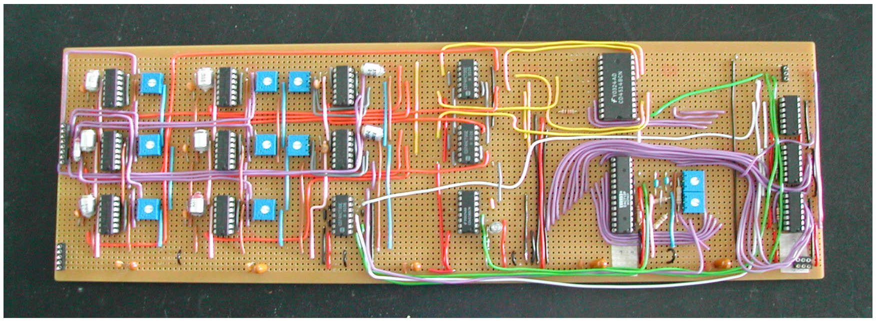


Figure 11.1: IMU Interface, LD001

11.iii Development

The Interface needed to be tested under identical conditions to what it will expect under IMU conditions, and therefore all timings on any emulator had to exactly match that of the IMU. Given the unreliability of Windows™ seen in sections 7 and 8, the decision was made to construct the emulator around a PIC16F873 microprocessor.

A test program was written using a 10MHz oscillator, with a short delay loop in order to assess how long the loop needs to be to produce 400ns. Unfortunately it was unattainable, the shortest possible pulse being 750ns, and so a 20MHz oscillator was introduced. The same program was run again, and no delay was found to be the answer. That is, placing a port output high, and immediately returning it low, produced a 400ns pulse. This meant that the program code could not utilise one single loop, calling data from a look up table, or other means, indeed it would have to be one continuously run program, loading the value to be sent as a literal each time.

After the sync pulse, there is a short delay in order for the connected interface to work on the received data. This delay is approximately 6.4us in length. However, due to the poor quality of the photocopy of the timing diagram, this was originally believed to be 84us. The PIC program was developed to include a delay of this length between each sensor output.

The program was written to be used with a bootloader in order to speed up any changes with the software. Although the circuit built to run the IMU emulator did not have the additional circuitry to operate the bootloader facility, it was used on the prototyping board manufactured previously by Alex Tombling for testing work using the Microchip™ PIC16F873 device. This prototype unit was very useful for creating small test circuits, and for running test programs such as discovering the shortest possible pulse length. It is highly recommended any future research efforts that intend to work with the PIC16F873, or any similar device construct a simple prototyping board.

The 400ns pulse was recorded on an oscilloscope and is given as figure 11.2. Figure 11.3 shows a second oscilloscope display of one sensor output being sent through.

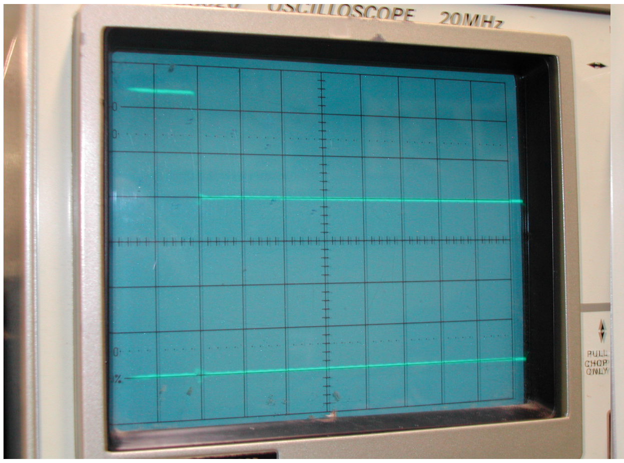


Figure 11.2:400ns Pulse from emulator

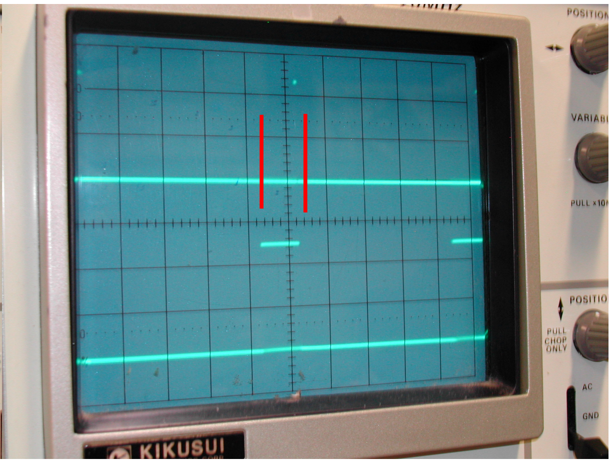


Figure 11.3: Emulator Output

The upper trace in figure 11.3 shows the sync pulse from the emulator, while the lower trace is the data output. In this view, in order to show the original delay of 64us between sensors, and the sync pulse, it is impossible to see the detail of the 24bits of data on the lower trace.

For the purposes of testing, all sensor outputs were given a value predetermined by the group, so that the interface can be correctly checked for errors. Each sensor was given a different value to check different points of the digital to analogue converter, and in order to ensure each output was correctly sent to each interface output. Given that of the 11 devices available, only 8 were to be used, the three spare outputs were given 0v values, while the other eight, values of –10V, -7.5V, -5V, -2.5V, +2.5V, +5V, +7.5V, and +10V.

The emulator showed up a number of faults with the interface, including incorrectly wired capacitors, all grounds on power supplies not tied together, and simple connections made incorrectly.

The full program listing, and circuit diagram of AT005 (IMU interface) is given in appendix 10.

11.iv Analysis

The emulator performed remarkably well. The use of the development board was a great benefit, and eased the process of developing the final emulator program. Figure 11.4 shows the clock pulse against the sync pulse in more detail. This perfect square wave could not have been produced any other way and still have the data controlled, without more expensive equipment, or dedicated PC software.

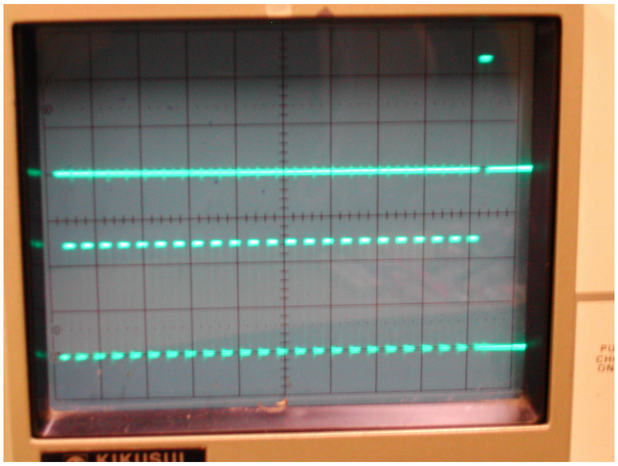


Figure 11.4: Clock Vs Sync, on IMU Emulator

11.v Conclusion

The IMU performed well when connected to the interface. Having known values produced by the emulator allowed faultfinding on the interface simple, which would not have been possible with the IMU directly.

This use for the PIC16F873 once again proved how useful the microprocessor family are. Consideration should be given in future to finding a faster PIC, and using it to manipulate the information produced by the IMU digitally before being converted to analogue for the control system to utilise.

12. Implementation

In order to be effective within the time span available the group was split into sections, as discussed in section 4. The group layout can be seen more clearly in figure 12.1.

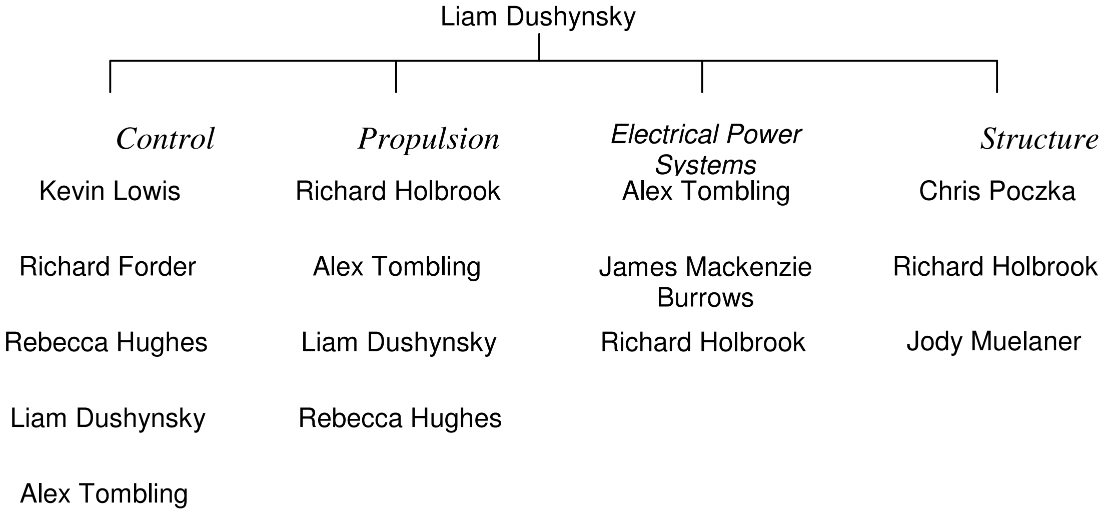


Figure 12.1: Group structure

Each group member was responsible for giving full proposals for each purchase they intended to make, which had to be accepted by the group, and remain within the agreed budget. An example copy of the meeting minutes is given in appendix 11.

After successful testing of the Plettenberg HP220/20/A4 S P4, it was decided to use these this year for stabilising the platform. The majority of the thrust was to be produced by the central IC powered ducted fan, with power being provided for the electric fans by an on board generator. The structure was not designed until very late on in the project and there was no time left to manufacture the platform. All results were therefore based upon test rigs and not physical in flight tests.

The speed controllers would need to be placed in the airflow of the stability ducts, and the Genset would need cooling on both the IC engine and the motor. A water cooled head was purchased for the IC engine following catastrophic failure of the original IC, and found to be very able to keep the outer body temperature low, and so consideration should be given to incorporating a water based cooling system in future years for both IC engines, and possible speed controllers should the need arise to purchase more.

The Genset was tested only on static loads, and a dynamic test with the stability fans not conducted before the project deadlines. Future testing should show that the current Genset is able to cope with the small changes in power demand expected from the stability fans.

In order to increase security of all tools in the workshop, a cupboard was adopted in the student workshop area, and locked with a hasp and padlock. Most group members held keys, and a list of all key holders posted on the wall behind the work area. A PC connected to the university network was also placed in the workshop to allow information to be shared efficiently and quickly.

13. Final Conclusions, Discussion & Further work

The new propulsion system appeared to be sufficient to lift the platform without payload, and without the additional electronics such as the main rectifier. For testing purposes it was intended to leave the generator on the ground, and connect an umbilical to feed power to the platform.

To interconnect the control system to the motor speed controllers, a microprocessor-controlled interface has been designed, constructed, and thoroughly tested in harsh operating conditions, including snowfall. A safety system was designed incorporating radio control so that in emergencies the platform can be made safe without approaching it.

Testing was carried out on the Generator set, and believed to be sufficient to power the platform. Due to the nature of the test environment it was not possible to investigate how much fuel would be required for a flight time of 20mins as requested in the PDS. Further investigations need to be completed by forthcoming research groups to reduce the weights involved in the power system, and to calculate the amount of fuel, and effect of the weight of fuel, required for the designed flight time.

The IMU was fully utilised for measuring all direction changes and found to be very reliable with small considerations to drift. The IMU interface, after initial teething problems, was also found to be reliable, but consideration should be given to replacing this analogue circuit with a high-speed digital variant. At present the digital signal is converted to analogue, and manipulated, whereas the manipulation could be performed far more quickly and economically digitally before conversion.

The yaw moment of the platform was ignored again this year, although the effect was presumed to be negligible. Further work is required to discover the extent of the yaw moment of the completed platform, and if necessary, processes put into place to remove the problem.

It is the intention of this group to place all information, including all reports on to CD Rom to assist future research efforts, to enable information to be quickly available and searched. This is to be carried out by Alex Tombling after the main deadline, and attached to the Group report [24].

Appendix 1

Product Design Specification

14.i Product Design Specification

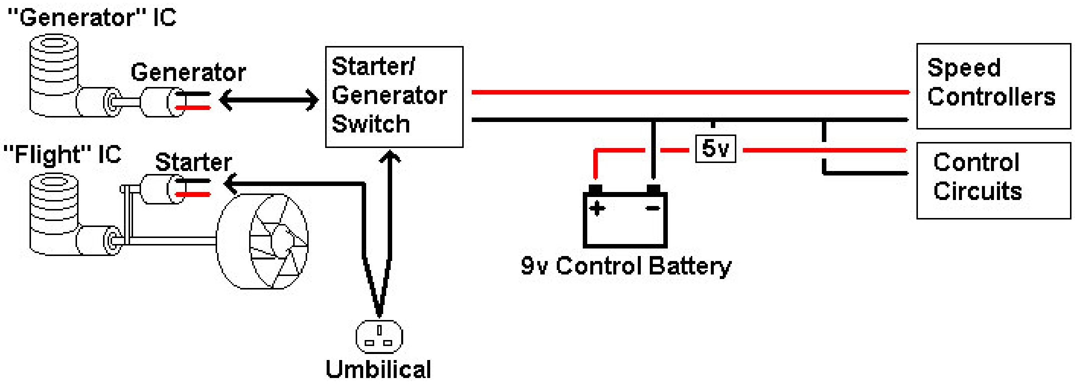
	This design
Performance	1. Must hover approximately 1 m above the ground. 2. Flight duration to be approximately 20 minutes. 3. Must remain Stable. 4. Must provide viable operating platform. 5. Must be able to carry a payload of up to 3 kg. 6. Must have the capability to have the On/Off controlled by remote.
Environment	1. Must be capable of operating in a temperature range of -10°C to 50°C. 2. Must be capable of operating in humid conditions and to be water resistant when operating in light rain. 3. Must be operated in minimal airflow disturbances i.e. minimal wind speeds.
Maintenance	1. Onboard battery must be easily attainable for possible replacement, and recharging. 2. Fuel tank for internal combustion engine must also be easily accessible for refuelling.
Life in Service	1. Products life in service is to be approximately 5 years.
Target Production Cost	1. A budget of £1000 has been assigned to this project.
Size	1. The flying platforms dimensions to be similar to the dimensions specified in the previous groups report.
Weight	1. Yet to be determined but should be designed for minimum weight possible. Estimated weight including payload is approx 10 kg.
Materials	1. Materials used must have a high mechanical tolerance, and must have as lower density as possible.
Design Constraints	1. The flight must be completely autonomous. 2. The design must not a helicopter-based design. 3. Must use an onboard IC engine for power generation.

Appendix 2

Electronic Interface Diagram

14.ii Electronic Interface Diagram

Initial power system block diagram.
(05/02/04)



The above diagram sketches out the block diagram of the main Electrical power system features. These include: -

- Control Battery. Small PP3 9volt battery to provide smooth voltage for control circuits.
- Umbilical cord. High current supply for starting IC engines.
- Generator. It is intended to produce ALL the electrical power required during flight from this source.
- Generator IC starter. This is the generator, used in reverse. A ‘switch’ controlled from the umbilical controls this hand over.
- Flight IC Starter. Small motor to start IC from umbilical supply.
- 5v Regulator. For Control circuits.

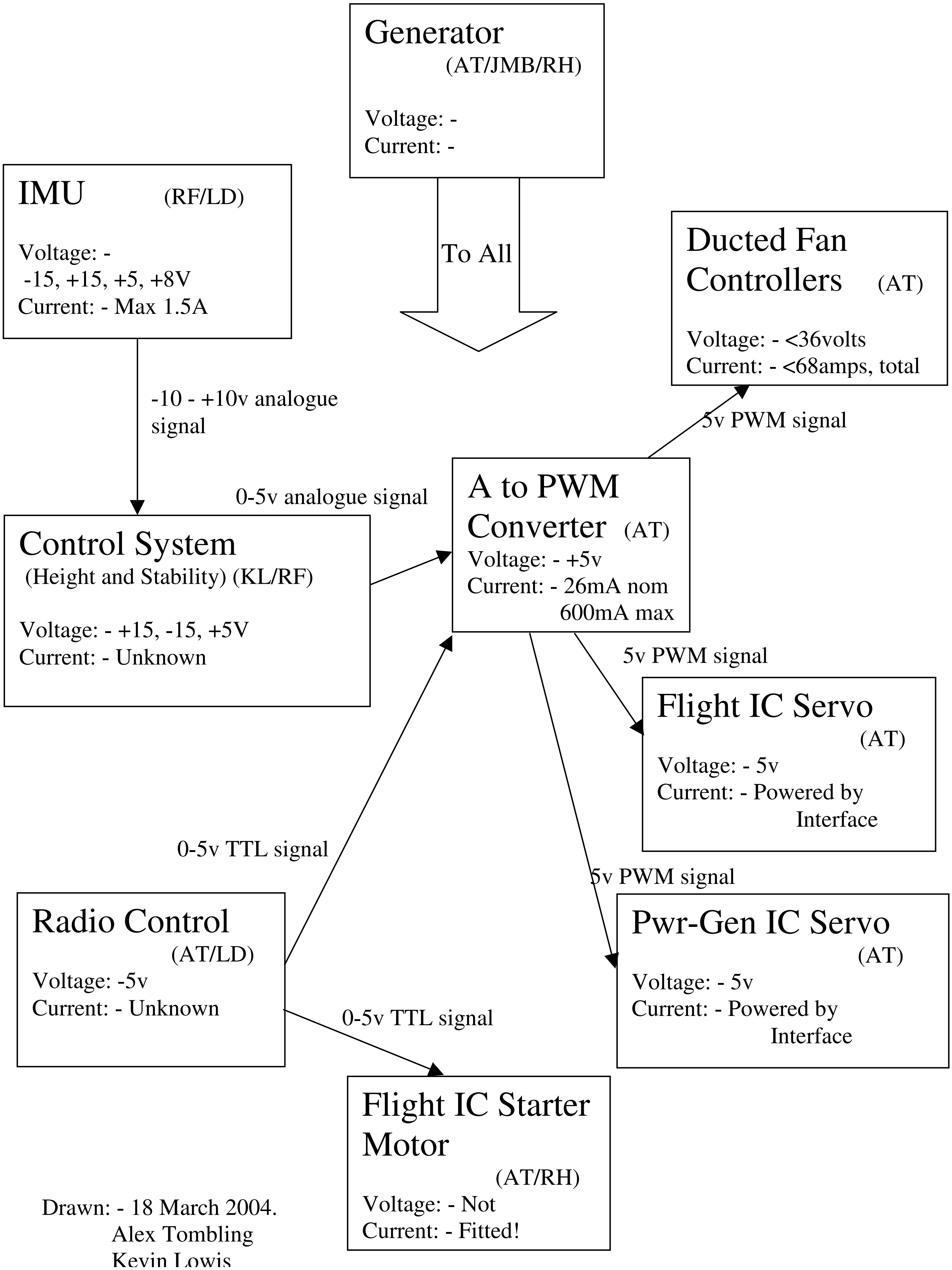
The start up procedure, from a power perspective, would be as follows.

1 Using the umbilical, the generator IC will be started, and run up to operating power.
2 The generator will then switch over to generation mode, and take control of the main platform power system.
3 The umbilical then starts the Flight IC, and should be run up to check it functions correctly, for approx 5mins until up to operating temperature.
4 Once all power systems are proved to function correctly, the umbilical is disconnected, and full control is handed over to the control systems.

Specifications known so far:

Estimates of worst-case requirements of outer fans is 1kg thrust each, which equates to 17A each, a total of 68A @24V. It is assumed that a generator would be 80% efficient, based on test results recorded, and data from the manufacturers. This requires 2KW to be produced from the IC engine.

Electronic Interface Diagram
(Forth draft, 18 March 2004)



Drawn: - 18 March 2004.
Alex Tombling
Kevin Lowis

Appendix 3

QBasic™ Program Listing

For PWM creation for Stability fan testing

14.iii.a Test program operating instructions

Speed Controller Test Software (REV 02)

FULLTEST.BAT

By Alex Tombling (08/12/03)

This software is a QBasic file, which runs in a dos window. Any changes that are required, e.g. to the preset value should be directed through Alex Tombling. In order to ease running of the software, please double click on “FULLTEST.BAT” which will load the correct file into QBasic, and start the software running. Due to the nature of the timings, this is accurate on the machine in the student workshop and is not guaranteed to work on a different machine.

This piece of software has two outputs on the PC Parallel port, LPT1. Data bit 0 (pin 2) gives a PWM output to RC standard ready for direct connection to the speed controller. The Speed controller signal ground (on the 3-core lead) should be connected to pin 18 on the parallel port. Data bit 1 (pin 3) gives a pulse output every time the user changes the power level. This high pulse is between 1 and 2ms in duration.

Upon starting the software the power level is set to 100%.

For safety reasons, before connecting the speed controller to the battery supply, the 5v supply should be ON, and the PC control should be set to output the 100% level signal. The controller should then be powered-up, ensuring correct polarity is observed from the battery, and will bleep twice to accept the *Full power* setting. The computer should then be set to send 0%, *Stop*, and the controller will bleep once to acknowledge this. This method is far safer than the alternative whereby stop is given first, then full power during which time the fan will go to full power.

The Computer has a number of keys that can be used. They are: -
s = Stop
f = Full power (100%)
i = Idle power (10%)
p = Preset Power level (65% currently)
+ = Increase power by 10%
- = Decrease power by 10%
> = Increase power by 5%
< = Decrease power by 5%
e = End Program

End program sends the stop command for 50 cycles to ensure that the motor is stopped before the control program disengages. It is advisable to re-teach the controller the 0, and 100% points after interruption of the signal. The control program will not allow the user to go above 100% power, or below 0%.

The screen will continuously display the current power setting being transmitted to the motor controller.

To avoid overheating the motor, after running above 40%, the motor should be idled for 2minutes to allow it to cool in the draft produced. In any case, the motor temperature must not exceed 100^{o}C.

14.iii.b QBasic™ Program Listing

Program listing for Fulltest.BAS

```
fullcycle = 7250
level = 100
laint = 10
smint = 5

'----------------Check keyboard
start:
a$ = INKEY$
IF a$ = "e" THEN GOTO shutdown
IF a$ = "s" THEN level = 0
IF a$ = "f" THEN level = 100
IF a$ = "i" THEN level = 10
IF a$ = "p" THEN level = 65
IF a$ = "+" THEN level = level + laint
IF a$ = "-" THEN level = level - laint
IF a$ = "." THEN level = level + smint
IF a$ = "," THEN level = level - smint

'------------Decide on level
IF level > 100 THEN level = 100
IF level < 0 THEN level = 0
newlevel = 350 + level + level + level + level
IF oncyc = newlevel THEN triger = 0 ELSE triger = 1
oncyc = newlevel
offcyc = fullcycle - oncyc

'------------------On/off pulse loop
IF triger = 1 GOTO trigloop
'----------On loop
FOR i = i TO oncyc
OUT &H378, 1
NEXT
i = 0
GOTO offloop

trigloop:
FOR i = i TO oncyc
OUT &H378, 3
NEXT
i = 0


'----------Off Loop
offloop:
FOR i = i TO offcyc
OUT &H378, 0
NEXT
i = 0

'-------------------Print output level to screen and restart
CLS
PRINT level; "%"
GOTO start

'---------------------Shut down routine
shutdown:
CLS
PRINT "Shutting down motor, making safe"
g = 0
reshutdown:
FOR i = i TO 350
OUT &H378, 1
NEXT
i = 0
FOR i = i TO 6900
OUT &H378, 0
NEXT
i = 0
g = g + 1
IF g = 50 THEN END ELSE GOTO reshutdown
```

Appendix 4

Pulse Width Modulation

Interface Circuit and Program Listing

14.iv Speed Controller Interface

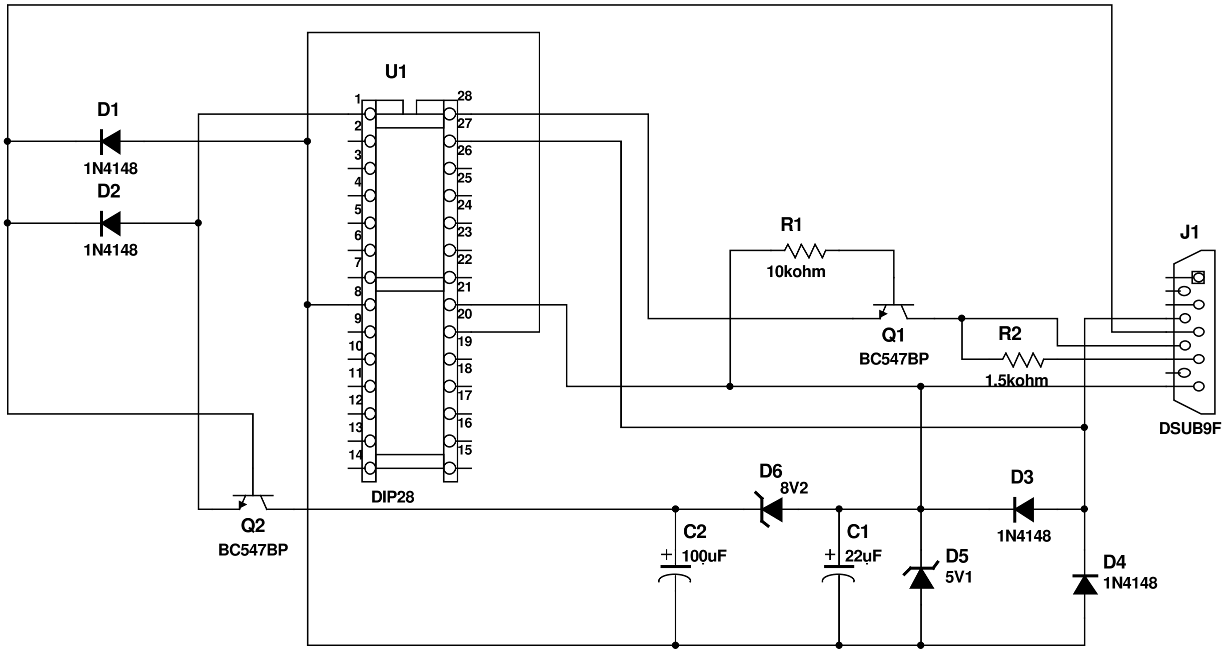


Figure 14.4.1:JDM PIC Programmer Circuit Diagram

Title: AT001 Circuit Diagram
JDM PIC Programmer
Designed by: Unknown
Revision: 1.0

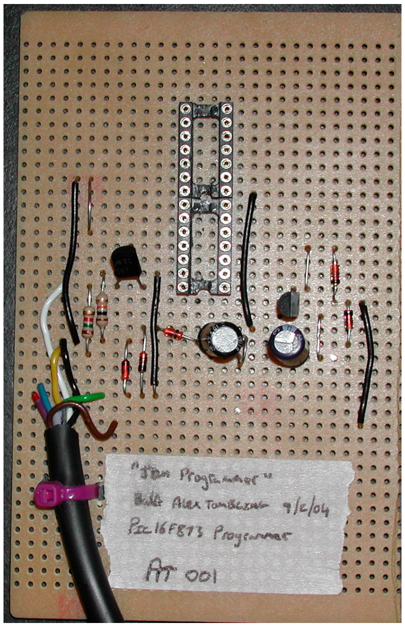


Figure 14.4.2: JDM PIC Programmer

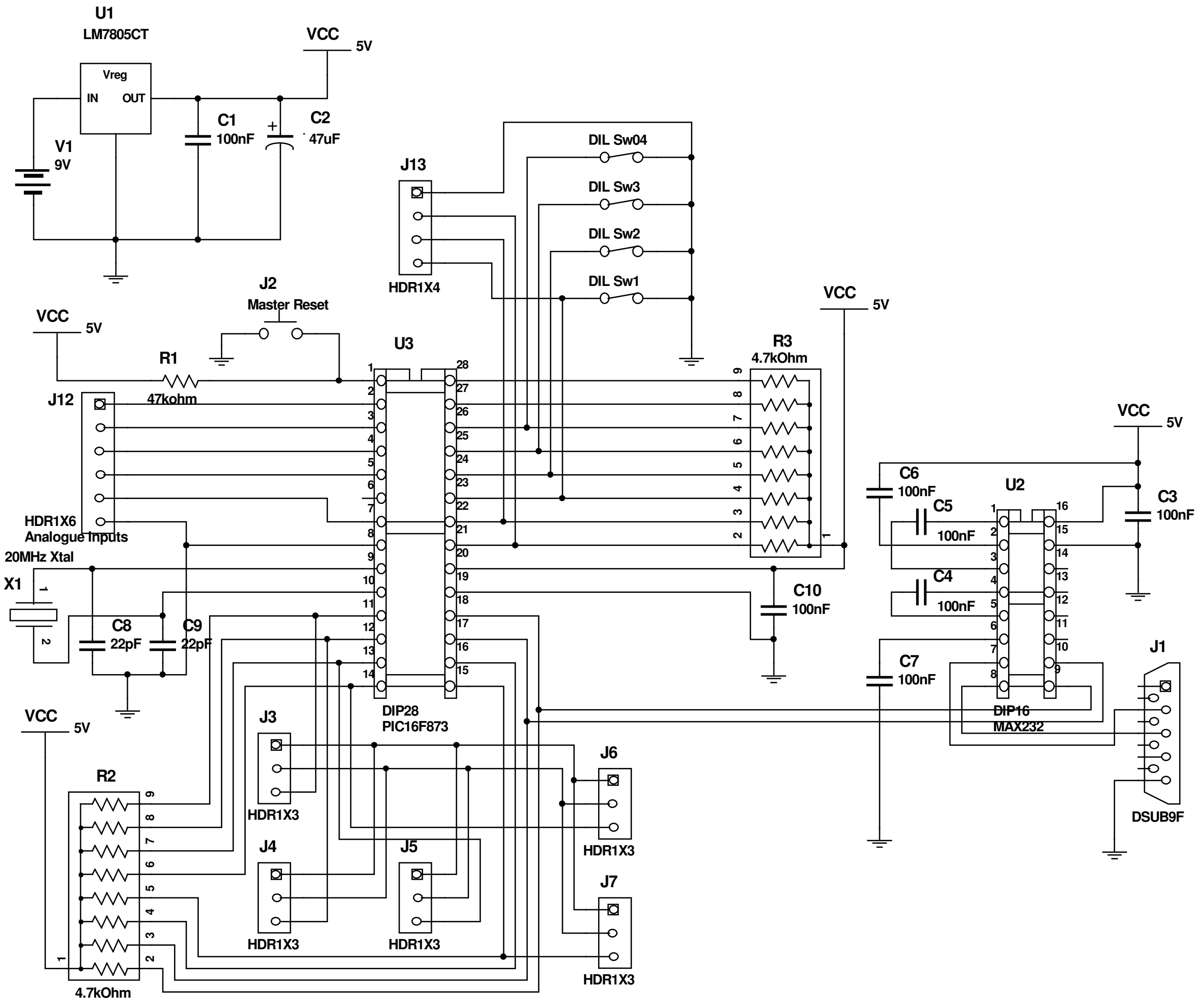


Figure 14.4.3:Contrtller Interface Circuit Diagram. PWM Generator.

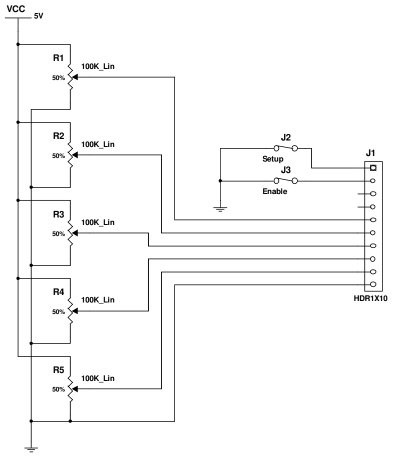


Figure 14.4.4: Analogue voltage controller. (Control System Emulator).

Speed controller interface program source code

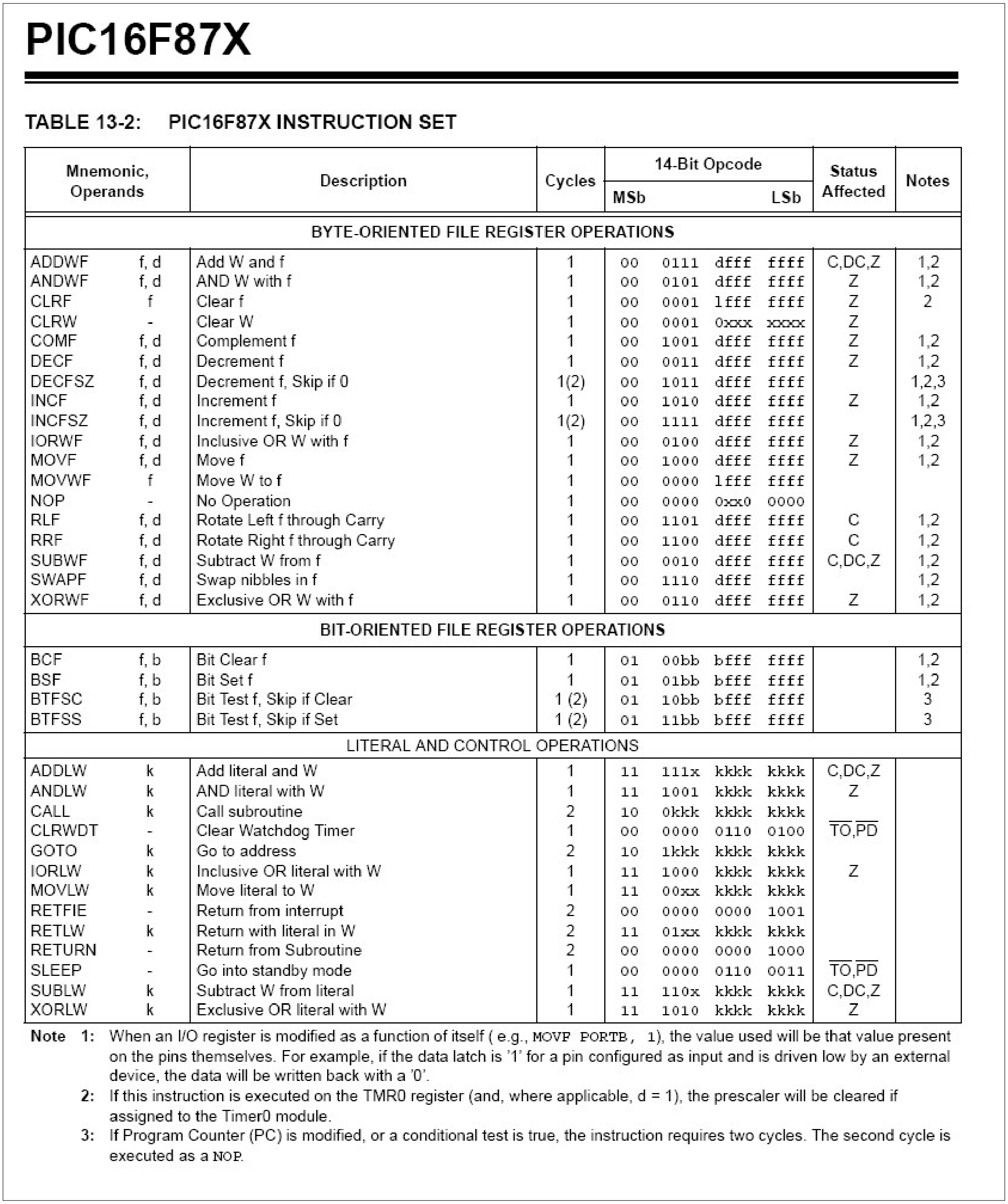
```
;******************************************************************************
;
;      A_PWM_C4.ASM  Source Code for PIC 16F873 Microcontroller
;
;      A.Tombling    Ver 3.2             11/03/2004
;
;      Function...............................................................
;
;             Read in 4 analouge voltage levels 0-5v and convert to digital level.
;             Convert this level into a PWM signal suitable for RC Control.
;             Output on port B(3:0) as PWM signals. (4 channel A to PWM conveter)
;             Designed for 4th year MENG Flying Platform
;             ----Bootloader version----
;             ----20MHz Version----
;...............................................................................
;
;             Small inaccuracies in timing corrected
;
;..............................................................................
       LIST P=PIC16F873    ; Declare processor type to assembler
       #include "p16f873.inc"
;*******************************************************************************
;
;      Declare lables for registers, constants and variables.....
;
ddca   EQU    H'FF'        ; Data Direction codes
ddcb   EQU    H'FF'        ; A inputs, B inputs, C Outputs
ddcc   EQU    H'00'        ;
timlev EQU    H'20'        ; Register for level sent to delay
count1 EQU    H'21'        ; Registers for delay subroutine
count2 EQU    H'22'
count3 EQU    H'23'
lount  EQU    H'24'        ; Register for 0% repeat count
;--------------------------------------------------------------------------------
; ----------Start of header for bootloaded prog-----------------
;
       org    0x0000 ; Reset Vector is always at location 00 for this chip!
       clrf   h'03'        ;zero status
       movlw  h'00'        ;high byte of long jump
       movwf  h'0A'        ;zero PCLATH
       goto   Setup1
; ---------------------------------End booted prog--------------
;
;      Startup routine
;
       ORG    04h
Setup1 BCF    STATUS,RP0
       BCF    STATUS,RP1          ; Page 0
       CLRF   PORTA
       CLRF   PORTB
       CLRF   PORTC               ; Clear all output buffers to initialise
       BSF    STATUS,RP0          ; Page1
       MOVLW  H'02'
       MOVWF  ADCON1              ; Set up A(5,3:0) as analouge inputs
       MOVLW  ddca                ; Setup port direction registers
       MOVWF  TRISA
       MOVLW  ddcb
       MOVWF  TRISB
       MOVLW  ddcc
       MOVWF  TRISC
       BCF    STATUS,RP0          ; Page0
;
;             Program routines...
;             Detect B0, (setup primary) set all outputs as FF  (2ms)
;             Detect B0 low, and set all outputs as 00 (1ms) for 1sec
;             Detect B1 (Enable) if high keep all outputs as 00 (1ms)
;             Detect B1 if low start AtoD conversion
;             B2 = 0 servo2 @ preset%, 1 servo2 @ 0%
;             B3 = 0 Invert Servo1
;             B4 = 0 Invert Servo2
;
;============================Main Program===============================================
;
;             Run setup program
;
start  BTFSC  PORTB,0                    ; Check B(0) if 1 set all outputs to 2ms
       GOTO   allhi              ;
       MOVLW  H'64'              ;Set low repeat counter to 2 seconds (100*20ms)
       MOVWF  lount              ; 100dec is 64hex
       GOTO   alllo              ; Set all controllers to 0%
       CLRF   lount
ckEnab BTFSC  PORTB,1                    ; Check Enable Pin, if high start ADC
       GOTO   alllo              ; Not enabled so set all outs as 0%
;
       GOTO   runpro             ; If skipped check for enable
;------------------------------------------------------------------------------
;
;============================Sub Programs===============================================
;                    Set all controllers to 100%
;
allhi  BSF    PORTC,0                    ; Fan 1 high
       MOVLW  H'FF'
       MOVWF  timlev
       CALL   sdelay             ; Call delay with level
       BCF    PORTC,0                    ; Fan 1 low
       MOVF   timlev,0
       SUBLW  H'00'              ; W=L-W
       MOVWF  timlev             ; Level = FF-Level
       CALL   sdelay             ; Call delay with level
;                    Next fan/servo
       BSF    PORTC,1                    ; Fan 2 high
       MOVLW  H'FF'
       MOVWF  timlev
       CALL   sdelay             ; Call delay with level
       BCF    PORTC,1                    ; Fan 2 low
       MOVLW  H'00'              ;
       MOVWF  timlev             ;
       CALL   sdelay             ; Call delay with level
;                    Next fan/servo
       BSF    PORTC,2                    ; Fan 3 high
       MOVLW  H'FF'
       MOVWF  timlev
       CALL   sdelay             ; Call delay with level
       BCF    PORTC,2                    ; Fan 3 low
       MOVLW  H'00'              ;
       MOVWF  timlev             ;
       CALL   sdelay             ; Call delay with level
;                    Next fan/servo
       BSF    PORTC,3                    ; Fan 4 high
       MOVLW  H'FF'
       MOVWF  timlev
       CALL   sdelay             ; Call delay with level
       BCF    PORTC,3                    ; Fan 4 low
       MOVLW  H'00'              ;
       MOVWF  timlev             ;
       CALL   sdelay             ; Call delay with level
;                    Next fan/servo
       BSF    PORTC,4                    ; Servo 1 high
       MOVLW  H'FF'
       BTFSS  PORTB,3
       XORLW  H'FF'              ; Invert the W if inv high
       MOVWF  timlev
       CALL   sdelay             ; Call delay with level
```

```
      BCF   PORTC,4                 ; Servo 1 low
      MOVLW H'00'             ;
      MOVWF timlev            ;
      CALL  sdelay            ; Call delay with level;
;                 Next fan/servo
      BSF   PORTC,5                 ; Servo 2 high
      MOVLW H'FF'
      BTFSS PORTB,4
      XORLW H'FF'             ; Invert the W if inv high
      MOVWF timlev
      CALL  sdelay            ; Call delay with level
      BCF   PORTC,5                 ; Servo2 low
      MOVLW H'00'             ;
      MOVWF timlev            ;
      CALL  sdelay            ; Call delay with level;
;
;                 Delays to force cycle to 20ms total
      MOVLW H'FF'
      MOVWF timlev
      CALL  sdelay
;
      GOTO  start
;
;----------------------------------------------------------------
;                 Set all controllers to 0%
;
alllo INCF  lount,F           ; Increment the counter incase of 0 start.
alllo1 BSF  PORTC,0                 ; Fan 1 high
      MOVLW H'00'
      MOVWF timlev
      CALL  sdelay            ; Call delay with level
      BCF   PORTC,0                 ; Fan 1 low
      MOVF  timlev,0
      SUBLW H'00'             ; W=L-W
      MOVWF timlev            ; Level = FF-Level
      CALL  sdelay            ; Call delay with level
;                 Next fan/servo
      BSF   PORTC,1                 ; Fan 2 high
      MOVLW H'00'
      MOVWF timlev
      CALL  sdelay            ; Call delay with level
      BCF   PORTC,1                 ; Fan 2 low
      MOVLW H'FF'             ;
      MOVWF timlev            ;
      CALL  sdelay            ; Call delay with level
;                 Next fan/servo
      BSF   PORTC,2                 ; Fan 3 high
      MOVLW H'00'
      MOVWF timlev
      CALL  sdelay            ; Call delay with level
      BCF   PORTC,2                 ; Fan 3 low
      MOVLW H'FF'             ;
      MOVWF timlev            ;
      CALL  sdelay            ; Call delay with level
;                 Next fan/servo
      BSF   PORTC,3                 ; Fan 4 high
      MOVLW H'00'
      MOVWF timlev
      CALL  sdelay            ; Call delay with level
      BCF   PORTC,3                 ; Fan 4 low
      MOVLW H'FF'             ;
      MOVWF timlev            ;
      CALL  sdelay            ; Call delay with level
;                 Next fan/servo
      BSF   PORTC,4                 ; Servo 1 high
      MOVLW H'00'
      BTFSS PORTB,3
```

```
       XORLW  H'FF'                ; Invert the W if inv high
       MOVWF  timlev
       CALL   sdelay               ; Call delay with level
       BCF    PORTC,4                     ; Servo 1 low
       MOVLW  H'FF'                ;
       MOVWF  timlev               ;
       CALL   sdelay               ; Call delay with level;
;                    Next fan/servo
       BSF    PORTC,5                     ; Servo 2 high
       MOVLW  H'00'
       BTFSS  PORTB,4
       XORLW  H'FF'                ; Invert the W if inv high
       MOVWF  timlev
       CALL   sdelay               ; Call delay with level
       BCF    PORTC,5                     ; Servo2 low
       MOVLW  H'FF'                ;
       MOVWF  timlev               ;
       CALL   sdelay               ; Call delay with level;
;
;                    Delays to force cycle to 20ms total
       MOVLW  H'FF'
       MOVWF  timlev
       CALL   sdelay
       MOVLW  H'FF'
       MOVWF  count3
       CALL   fdelay
;
       DECFSZ lount,F
       GOTO   alllo1
       GOTO   ckEnab        ; Return and check if enabled
;---------------------------------------------------------------------------
;                    AtoD Conversion program
;
runpro MOVLW  B'10000001'          ; 10 fosc/32, 000 ch0, 001 nogo, on
       MOVWF  ADCON0               ; Switch on AtoD
       MOVLW  H'00'
       MOVWF  timlev
       CALL   sdelay               ; 1ms delay to stabilise signal
       BSF    ADCON0,GO_DONE              ; Start conversion
       NOP
       NOP
       BTFSC  ADCON0,GO_DONE
       GOTO   $-1
       BSF    PORTC,0                     ; Fan 1 high
       MOVF   ADRESH,W             ; Get D value into W
       MOVWF  timlev
       CALL   sdelay               ; Call delay with level
       BCF    PORTC,0                     ; Fan 1 low
       MOVF   timlev,0
       SUBLW  H'FF'                ; W=L-W
       MOVWF  timlev               ; Level = FF-Level
       CALL   sdelay               ; Call delay with level
;                    Next Fan
       MOVLW  B'10001001'          ; 10 fosc/32, 001 ch1, 001 nogo, on
       MOVWF  ADCON0               ; Switch on AtoD
       MOVLW  H'00'
       MOVWF  timlev
       CALL   sdelay               ; 1ms delay to stabilise signal
       BSF    ADCON0,GO_DONE              ; Start conversion
       NOP
       NOP
       BTFSC  ADCON0,GO_DONE
       GOTO   $-1
       BSF    PORTC,1                     ; Fan 2 high
       MOVF   ADRESH,W             ; Get D value into W
       MOVWF  timlev
       CALL   sdelay               ; Call delay with level
       BCF    PORTC,1                   ; Fan 2 low
       MOVF   timlev,0
       SUBLW  H'FF'               ; W=L-W
       MOVWF  timlev              ; Level = FF-Level
       CALL   sdelay              ; Call delay with level
;                   Next Fan
       MOVLW  B'10010001'         ; 10 fosc/32, 010 ch2, 001 nogo, on
       MOVWF  ADCON0              ; Switch on AtoD
       MOVLW  H'00'
       MOVWF  timlev
       CALL   sdelay              ; 1ms delay to stabilise signal
       BSF    ADCON0,GO_DONE            ; Start conversion
       NOP
       NOP
       BTFSC  ADCON0,GO_DONE
       GOTO   $-1
       BSF    PORTC,2                   ; Fan 3 high
       MOVF   ADRESH,W            ; Get D value into W
       MOVWF  timlev
       CALL   sdelay              ; Call delay with level
       BCF    PORTC,2                   ; Fan 3 low
       MOVF   timlev,0
       SUBLW  H'FF'               ; W=L-W
       MOVWF  timlev              ; Level = FF-Level
       CALL   sdelay              ; Call delay with level
;                   Next Fan
       MOVLW  B'10011001'         ; 10 fosc/32, 011 ch3, 001 nogo, on
       MOVWF  ADCON0              ; Switch on AtoD
       MOVLW  H'00'
       MOVWF  timlev
       CALL   sdelay              ; 1ms delay to stabilise signal
       BSF    ADCON0,GO_DONE            ; Start conversion
       NOP
       NOP
       BTFSC  ADCON0,GO_DONE
       GOTO   $-1
       BSF    PORTC,3                   ; Fan 4 high
       MOVF   ADRESH,W            ; Get D value into W
       MOVWF  timlev
       CALL   sdelay              ; Call delay with level
       BCF    PORTC,3                   ; Fan 4 low
       MOVF   timlev,0
       SUBLW  H'FF'               ; W=L-W
       MOVWF  timlev              ; Level = FF-Level
       CALL   sdelay              ; Call delay with level
;                   Next Fan
       MOVLW  B'10100001'         ; 10 fosc/32, 101 ch4, 001 nogo, on
       MOVWF  ADCON0              ; Switch on AtoD
       MOVLW  H'00'
       MOVWF  timlev
       CALL   sdelay              ; 1ms delay to stabilise signal
       BSF    ADCON0,GO_DONE            ; Start conversion
       NOP
       NOP
       BTFSC  ADCON0,GO_DONE
       GOTO   $-1
       BSF    PORTC,4                   ; Fan 5 high
       MOVF   ADRESH,W            ; Get D value into W
       BTFSS  PORTB,3
       XORLW  H'FF'               ; Invert the W if inv high
       MOVWF  timlev
       CALL   sdelay              ; Call delay with level
       BCF    PORTC,4                   ; Fan 5 low
       MOVF   timlev,0
       SUBLW  H'FF'               ; W=L-W
       MOVWF  timlev              ; Level = FF-Level
       CALL   sdelay              ; Call delay with level
```

```
;
       MOVLW  H'50'
       MOVWF  count3
       CALL   fdelay              ; fine delay to tweak 20ms cycle
;
       GOTO   ckEnab
;
;===========================Delay Routine===========================================
;             This routine should take 1ms + level to complete
;
;             sdelay takes 1ms + count3 (count3= 0-1.25ms)
;
;
sdelay MOVF   timlev,0
       MOVWF  count3
       MOVLW  H'FF'
       MOVWF  count2              ;and copy it into the COUNT registers
;
D1     MOVLW  H'05'               ;Load  into the working register
       MOVWF  count1
       DECFSZ count1,F            ;Decrement the counter and check for zero
       GOTO   $-1                 ;If not zero, jump back one instruction
;
       DECFSZ count2,F
       GOTO   D1
;
fdelay INCF   count3,1            ;Increment count3 so 00 is correctly read at end.
D2     MOVLW  H'06'               ;Load  into the working register
       MOVWF  count1
       DECFSZ count1,F            ;Decrement the counter and check for zero
       GOTO   $-1                 ;If not zero, jump back one instruction
;
       DECFSZ count3,F
       GOTO   D2

       RETURN                     ;and finally, return from the subroutine.
;
;**********************************************************************************
       END
```



PIC16F87X

TABLE 13-2: PIC16F87X INSTRUCTION SET

Mnemonic,	Operands	Description	Cycles	14-Bit Opcode MSb			LSb	Status Affected	Notes
BYTE-ORIENTED FILE REGISTER OPERATIONS									
ADDWF	f, d	Add W and f	1	00	0111	dfff	ffff	C,DC,Z	1,2
ANDWF	f, d	AND W with f	1	00	0101	dfff	ffff	Z	1,2
CLRF	f	Clear f	1	00	0001	lfff	ffff	Z	2
CLRW	-	Clear W	1	00	0001	0xxx	xxxx	Z	
COMF	f, d	Complement f	1	00	1001	dfff	ffff	Z	1,2
DECF	f, d	Decrement f	1	00	0011	dfff	ffff	Z	1,2
DECFSZ	f, d	Decrement f, Skip if 0	1(2)	00	1011	dfff	ffff		1,2,3
INCF	f, d	Increment f	1	00	1010	dfff	ffff	Z	1,2
INCFSZ	f, d	Increment f, Skip if 0	1(2)	00	1111	dfff	ffff		1,2,3
IORWF	f, d	Inclusive OR W with f	1	00	0100	dfff	ffff	Z	1,2
MOVF	f, d	Move f	1	00	1000	dfff	ffff	Z	1,2
MOVWF	f	Move W to f	1	00	0000	lfff	ffff		
NOP	-	No Operation	1	00	0000	0xx0	0000		
RLF	f, d	Rotate Left f through Carry	1	00	1101	dfff	ffff	C	1,2
RRF	f, d	Rotate Right f through Carry	1	00	1100	dfff	ffff	C	1,2
SUBWF	f, d	Subtract W from f	1	00	0010	dfff	ffff	C,DC,Z	1,2
SWAPF	f, d	Swap nibbles in f	1	00	1110	dfff	ffff		1,2
XORWF	f, d	Exclusive OR W with f	1	00	0110	dfff	ffff	Z	1,2
BIT-ORIENTED FILE REGISTER OPERATIONS									
BCF	f, b	Bit Clear f	1	01	00bb	bfff	ffff		1,2
BSF	f, b	Bit Set f	1	01	01bb	bfff	ffff		1,2
BTFSC	f, b	Bit Test f, Skip if Clear	1 (2)	01	10bb	bfff	ffff		3
BTFSS	f, b	Bit Test f, Skip if Set	1 (2)	01	11bb	bfff	ffff		3
LITERAL AND CONTROL OPERATIONS									
ADDLW	k	Add literal and W	1	11	111x	kkkk	kkkk	C,DC,Z	
ANDLW	k	AND literal with W	1	11	1001	kkkk	kkkk	Z	
CALL	k	Call subroutine	2	10	0kkk	kkkk	kkkk		
CLRWDT	-	Clear Watchdog Timer	1	00	0000	0110	0100	$\overline{TO}$,$\overline{PD}$	
GOTO	k	Go to address	2	10	1kkk	kkkk	kkkk		
IORLW	k	Inclusive OR literal with W	1	11	1000	kkkk	kkkk	Z	
MOVLW	k	Move literal to W	1	11	00xx	kkkk	kkkk		
RETFIE	-	Return from interrupt	2	00	0000	0000	1001		
RETLW	k	Return with literal in W	2	11	01xx	kkkk	kkkk		
RETURN	-	Return from Subroutine	2	00	0000	0000	1000		
SLEEP	-	Go into standby mode	1	00	0000	0110	0011	$\overline{TO}$,$\overline{PD}$	
SUBLW	k	Subtract W from literal	1	11	110x	kkkk	kkkk	C,DC,Z	
XORLW	k	Exclusive OR literal with W	1	11	1010	kkkk	kkkk	Z	

Note 1: When an I/O register is modified as a function of itself (e.g., `MOVF PORTB, 1`), the value used will be that value present on the pins themselves. For example, if the data latch is '1' for a pin configured as input and is driven low by an external device, the data will be written back with a '0'.

2: If this instruction is executed on the TMR0 register (and, where applicable, d = 1), the prescaler will be cleared if assigned to the Timer0 module.

3: If Program Counter (PC) is modified, or a conditional test is true, the instruction requires two cycles. The second cycle is executed as a `NOP`.

Appendix 5

Propulsion Test Results

14.v Propulsion Test Results

Pulse Width (ms)	RPM	Current In (A)	Power In (W)	Thrust (g)
1.00	0	0	0	0.00
1.07	4800	0.4	10.12	90.00
1.10	4933.33	0.5	12.6	45.45
1.15	7000	1	25.1	93.42
1.20	8866.67	1.8	45	164.11
1.25	10466.7	2.7	67.23	229.75
1.30	12000	3.6	89.28	318.12
1.35	13600	4.8	118.56	386.29
1.40	14866.7	6.8	167.28	477.18
1.45	16666.7	9	220.5	605.94
1.50	18466.7	10.5	257.25	717.03
1.55	19800	13.3	323.19	858.42
1.60	21600	16.8	408.24	1009.90
1.65	23066.7	20	486	1161.39
1.70	24666.7	24.1	580.81	1322.97
1.75	25933.3	28	672	1476.98
1.80	27333.3	34	812.6	1656.24
1.85	28733.3	39	928.2	1817.82
1.90	29866.7	46	1090.2	1984.46
1.95	30466.7	48	1137.6	2080.40
2.00	30533.3	48.8	1156.56	2080.40

Table 14.5.1: Stability Propulsion Test Results

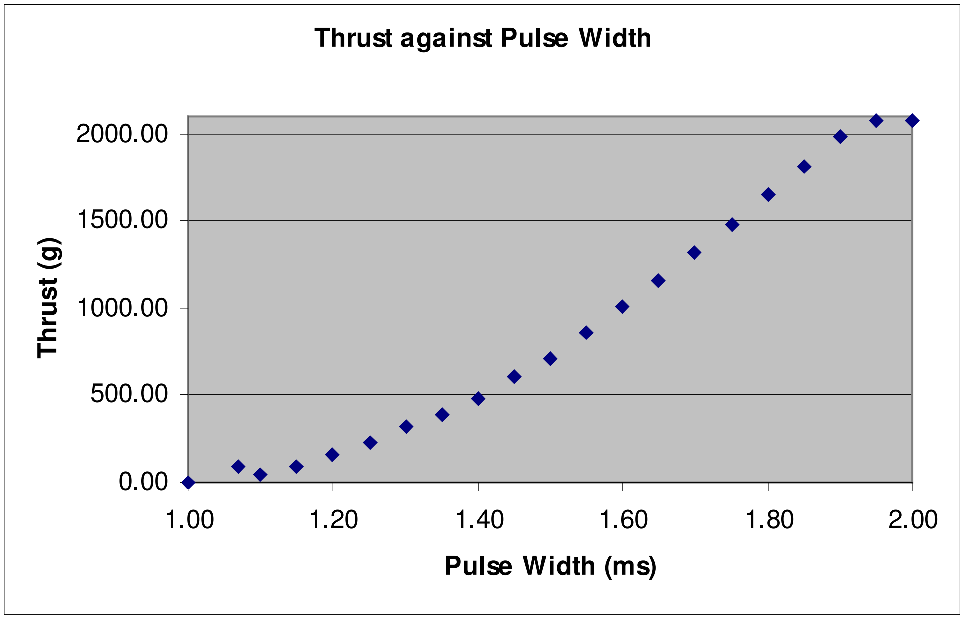


Figure 14.5.1: Stabilisation fans test graphical results

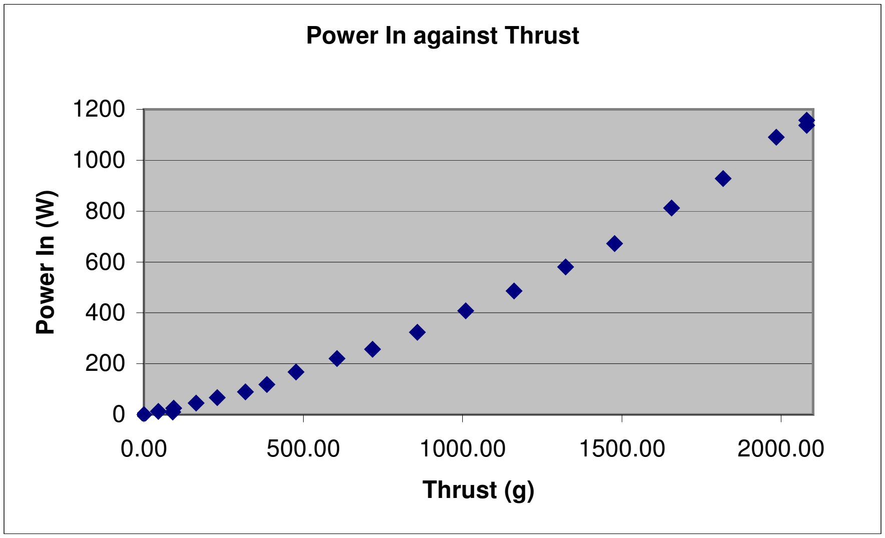


Figure 14.5.2: Stabilisation fans test graphical results

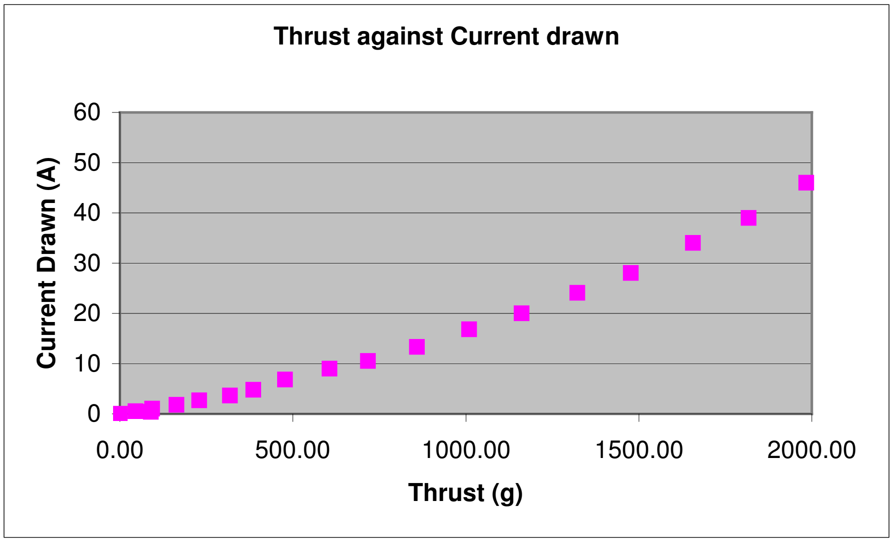


Figure 14.5.3: Stabilisation fans test graphical results

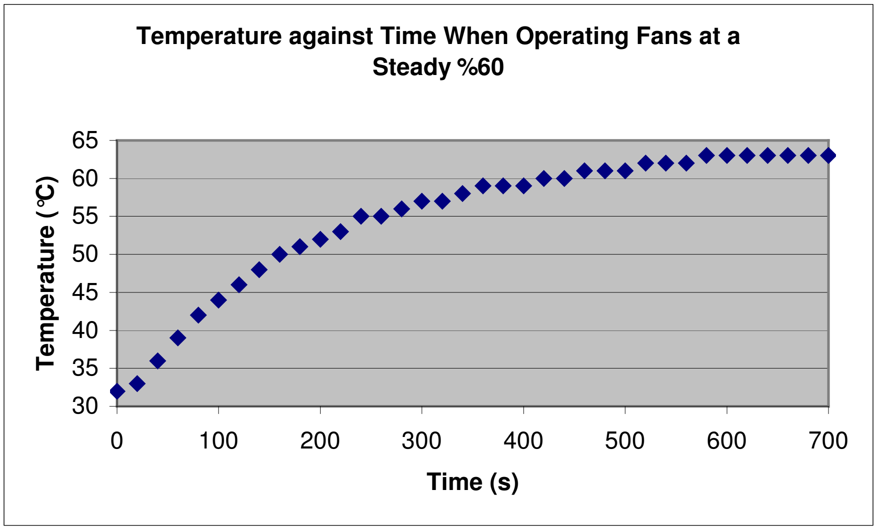


Figure 14.5.4: Stabilisation fans temperature readings for steady thrust output

Date	Thrust	RPM	Comments
27/02/04	4.1	20000	no duct
01/03/04	1.34	12000	no duct
	3.9	20000	no duct
	4	20800	no duct
09/03/04	3.8	20000	no duct
10/03/04	3.46	19200	no duct
	2.94	16400	no duct
	3.7	19800	no duct
	3.2	18000	no duct
	3.5	19200	no duct
	3	17400	no duct
	3.63	20000	no duct
	3.2	18400	no duct
	3.7	19600	no duct
	3.8	20000	no duct
16/03/04	2.5	19600	no duct
	2.8	20800	no duct
	2.65	20000	no duct
	0.6	9800	no duct
	2.475	18200	no duct
	2.9	19800	no duct
18/03/04	2.8	19600	cardboard test 3
	3.1	20000	cardboard test 4
	3.3	20000	cardboard test 4
	3.6	19800	cardboard test 4
04/05/04	4.645	19400	duct 3
	4.145	20000	duct 3
	4.145	19800	duct 3
	4.345	20400	duct 5
	4.245	19200	duct 4
	3.345	20000	no duct
	3.62	20400	-4
	3.4	19600	-8
	3.2	19200	-2
	3.53	20000	2
	3.73	20400	4
	3.82	20600	8
	3.94	20400	6
	3.26	20000	0
14/05/04	0.855	10000	no duct
	0.705	10000	no duct
	1.245	12200	no duct
	2.335	16000	no duct
	0.34	6600	no duct
	0.58	8200	no duct
	0.95	10000	no duct
	1.335	12000	no duct
	1.68	13400	no duct
	2.5	16200	no duct
	3.63	19800	no duct
	1.57	13600	no duct

Table 14.5.2: Central Thrust Ducted Fan results

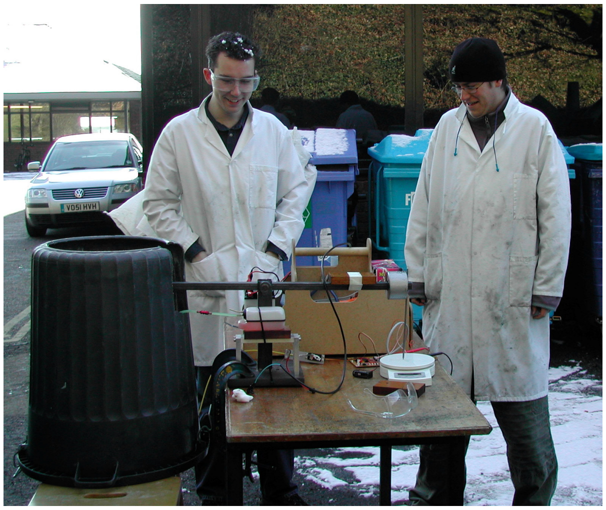


Figure 14.5.5: Thrust IC in safety cage under test

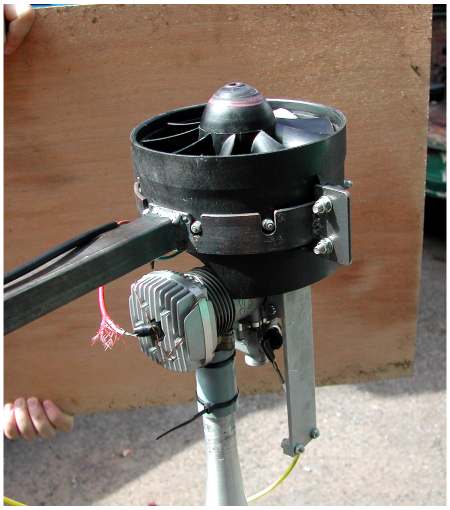


Figure 14.5.6: Thrust IC with Ducted Fan

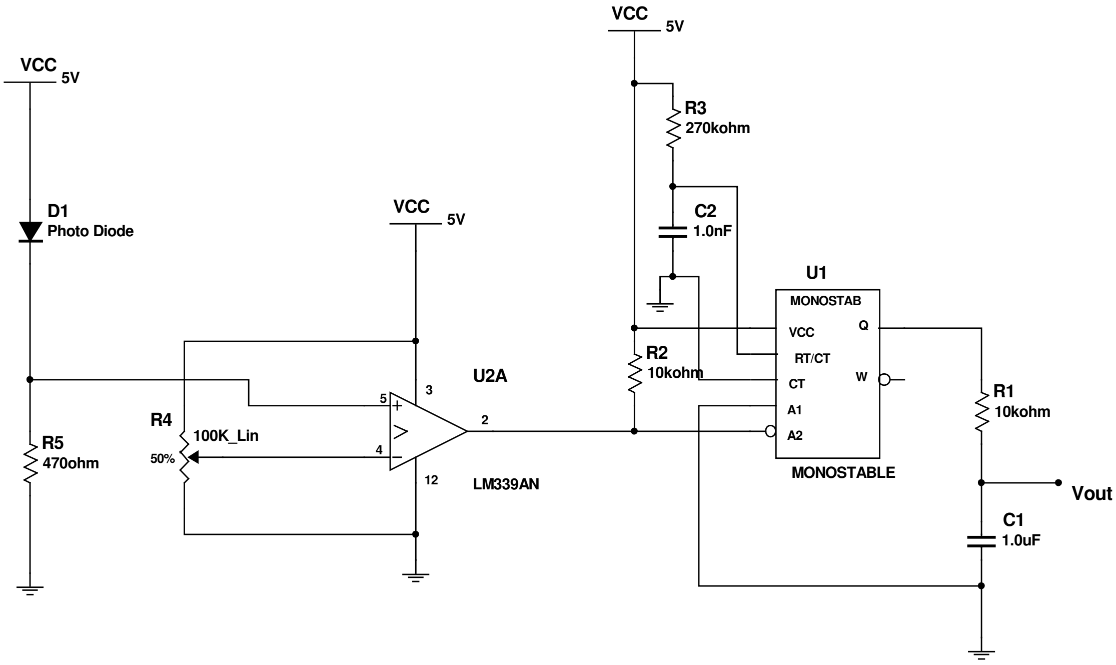


Figure 14.5.7:RCH001 Frequency to Voltage converter for rotational speed measurement

Appendix 6

Genset Test Results

14.vi Genset Test Results

Electrical Power In (W)	Speed (rpm)	V out (V)	R load (?)	Electrical Power Out (W)	Efficiency
10.08	6000	2.4	5	1.15	11.43%
32.63	13600	7.1	5	10.08	30.90%
57.50	19600	10.9	5.1	23.30	40.51%
82.17	24000	13.4	5.2	34.53	42.02%
94.24	27200	15.4	5.3	44.75	47.48%
108.68	30000	17	5.4	53.52	49.24%
123.00	32000	18.2	5.4	61.34	49.87%
127.92	34000	19.3	5.5	67.73	52.94%
157.44	37200	21.2	5.6	80.26	50.98%
157.44	38000	21.9	5.6	85.64	54.40%
57.73	15800	8.3	4	17.22	29.83%
112.05	26400	14.5	4	52.56	46.91%
160.55	32400	18.3	4	83.72	52.15%
196.80	38000	21.4	4	114.49	58.18%
57.50	15000	7.2	3	17.28	30.05%
124.00	25200	13.6	3	61.65	49.72%
179.58	30000	17.5	3	102.08	56.85%
232.75	37600	21	3	147.00	63.16%
57.04	13600	6.9	2	23.81	41.73%
133.92	23600	12.7	2	80.65	60.22%
221.40	30800	16.7	2	139.45	62.98%
275.72	37600	20.6	2	212.18	76.95%
57.50	11600	5.5	1	30.25	52.61%
167.96	21000	10.9	1	118.81	70.74%
296.40	29200	15.4	1	237.16	80.01%
386.37	36400	19.3	1	372.49	96.41%

Table 14.6.1: Plettenberg 3-phase test results.

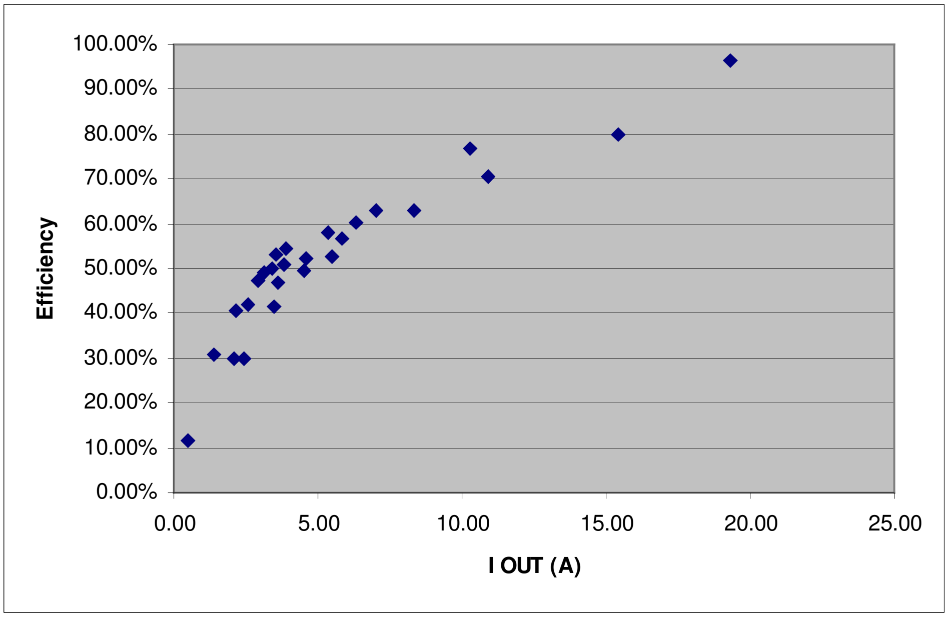


Figure 14.6.1: Electrical Efficiency of 3-phase Plettenberg.

Electrical Power In (W)	Speed (rpm)	V out (V)	R load (?)	Electrical Power Out (W)	Efficiency
18.14	6600	4	5	3.20	17.64%
40.16	12600	7.7	5	11.86	29.53%
62.50	17400	10.6	5	22.47	35.96%
15.06	6400	4	4	4.00	26.56%
42.50	12200	7.5	4	14.06	33.09%
64.74	16800	10	4	25.00	38.62%
89.64	20000	11.98	4	35.88	40.03%
17.57	6200	3.7	3	4.56	25.97%
42.50	11600	6.9	3	15.87	37.34%
70.00	15600	9.2	3	28.21	40.30%
99.60	17400	11	3	40.33	40.50%
17.57	5200	3	2	4.50	25.61%
47.50	10000	6	2	18.00	37.89%
75.00	14000	7.8	2	30.42	40.56%
109.56	18000	9	2	40.50	36.97%
12.60	3000	1.5	1	2.25	17.86%
22.59	5000	2.3	1	5.29	23.42%
45.18	6800	3.6	1	12.96	28.69%
60.00	9800	4.5	1	20.25	33.75%
84.66	11800	5.6	1	31.36	37.04%

Table 14.6.2: Efficiency of Graupner speed 600BB Turbo 12V

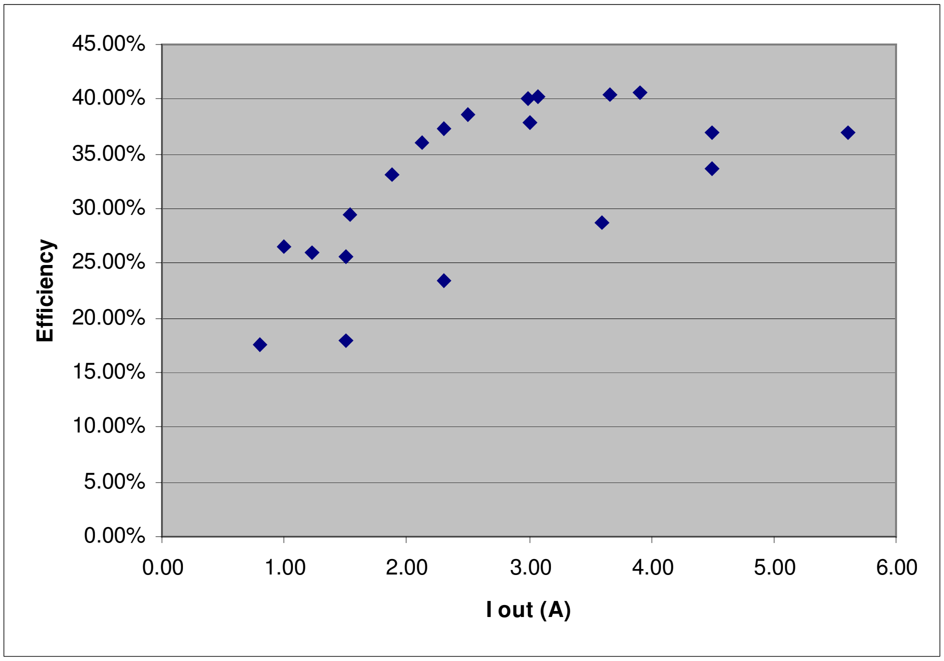


Figure 14.6.2: Efficiency of Graupner speed 600BB Turbo 12V

Speed (±100) (rpm)	Load Resistance (?)	V out (V)	I out (A)	Electrical Power Out (W)	Integrity of result
15500	0.5	18.0	36.0	648.00	Good
16000	0.5	18.5	37.0	684.50	Good
18200	0.5	22.0	44.0	968.00	Good
18500	0.5	20.0	40.0	800.00	Good
19000	0.5	21.9	43.8	959.22	Good
19000	0.5	22.0	44.0	968.00	Good
21000	0.5	24.0	48.0	1152.00	Good
18000	1.0	2.4	2.4	5.76	Poor
15000	5.0	20.7	4.1	85.70	Good
18300	5.0	25.0	5.0	125.00	Good
17000	Open Circuit	30.0	0.0	0.00	Good
19000	Open Circuit	33.0	0.0	0.00	Good

Table 14.6.3: Genset Test results using Plettenberg HP370/30/A2

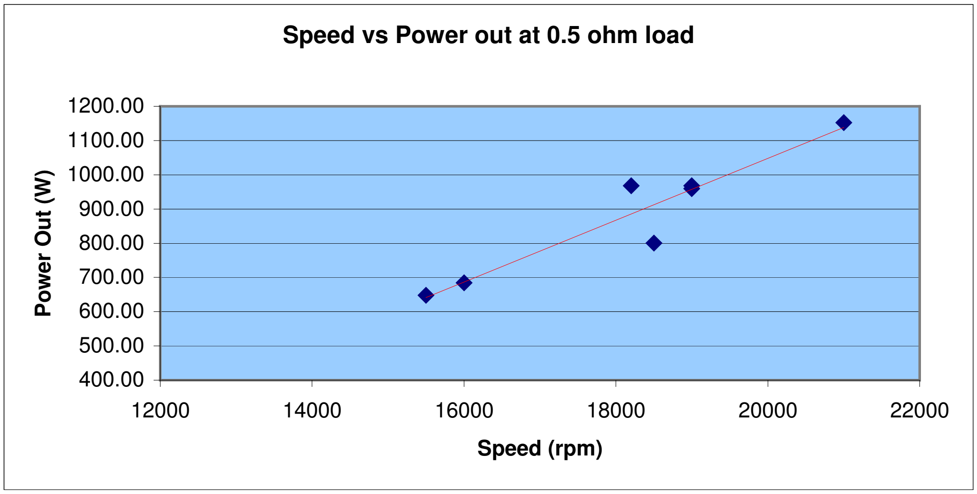


Figure 14.6.3: Speed of motor Vs Power produced by Genset powered by IC.

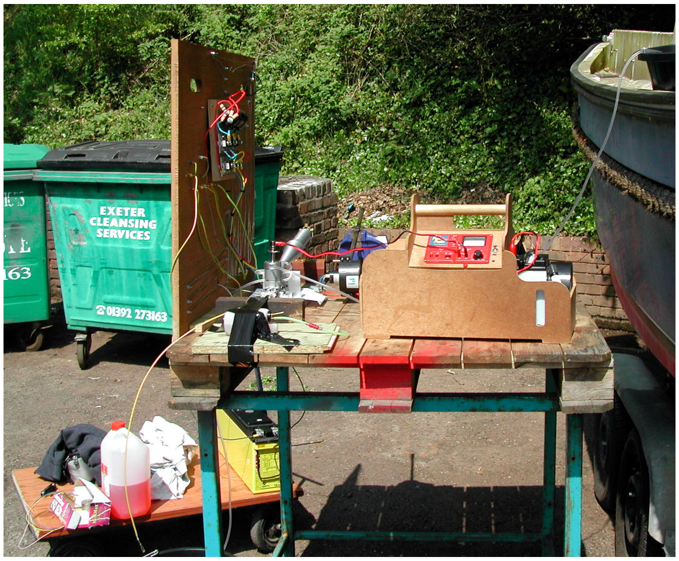


Figure 14.6.4: Genset Test rig

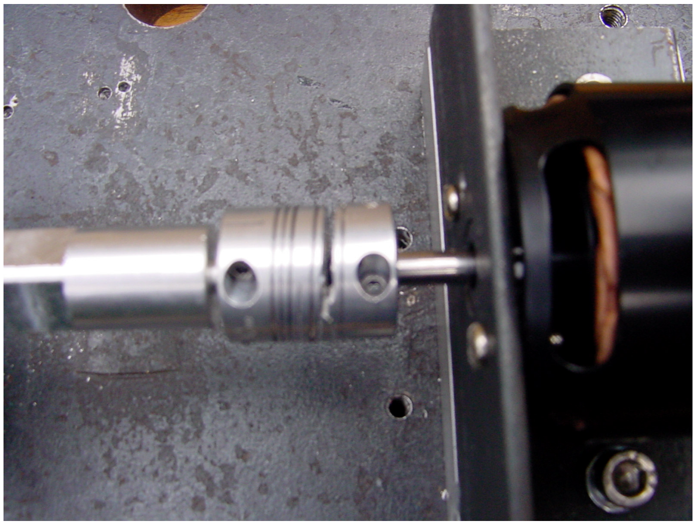


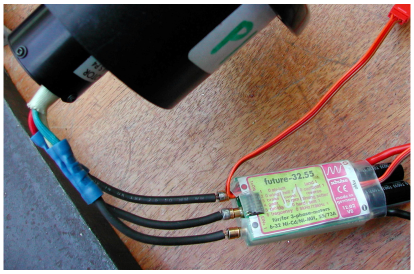
Figure 14.6.5: Genset Coupling - Failed

Appendix 7

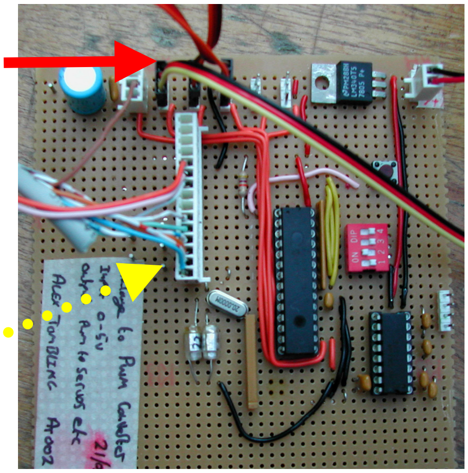
PWM Interface Instructions

14.vii PWM Interface Instructions

Instructions for use of the Analogue to PWM converter circuit.



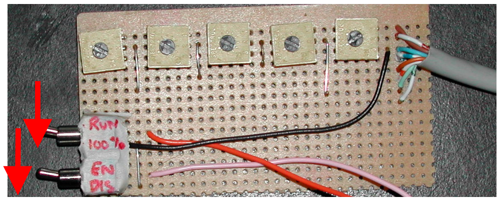
1. Connect speed controller to fan motor correctly. Ensure blue cable is connected to socket with blue dot etc.



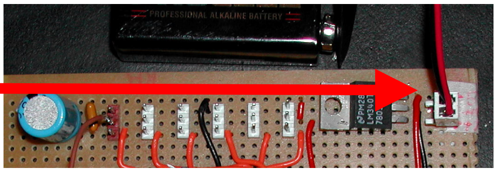
2. Connect the 3-way control cables to the converter circuit. Ensure the black wire is nearest the edge of the board. There are 5 possible positions, numbered 1 to 5, 1 being on the right. The 6th connection coloured brown is not used for servo/controller used.

NOTE: The 0% and 100% signal on Position 5 can be reversed by use of dipswitch 2. This is used for the servos, and should be OFF if channel used for speed controller, and on for servos.

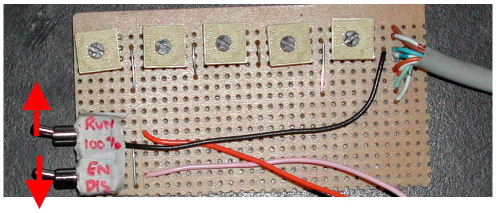
3. Ensure the analogue circuit is connected to this board as shown by the dashed arrow.



4. Set switches on analogue board to 100% and DISabled.

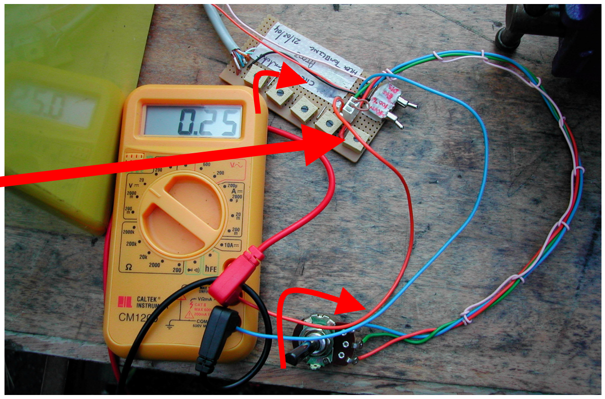


5. Connect 9v battery as shown. Ensure pin nearest board edge is not connected.



6. Connect Speed controller power cables to Battery. If instructions have been correctly followed, the motor should emit a burst of tones, followed by a double beep. If a single beep is heard, disconnect the main power pack, and check through instructions above.

7. Place Run/100% switch into the Run position, leaving the other switch in the DISabled position. The motors should now emit a single beep to acknowledge the 0% position.



8. Once the Enable switch is placed into the enable position, the analogue input will be converted to a signal for the controller/servo. The analogue voltages can be adjusted using the respective potentiometer. The arrow indicates channel 1 (see step 2 above). The rotation arrows indicate the direction for 0%. Channel 5 has the option of using a hand held pot, with jumper set as shown. When reversed, the onboard pot is used. Two leads are shown connected to a multimeter; these give the voltage output between 0 and 5V DC of channel 5. 5V is 100% pulse width. The figure shows channel 5 at 0.25V (5%), and the system disabled.

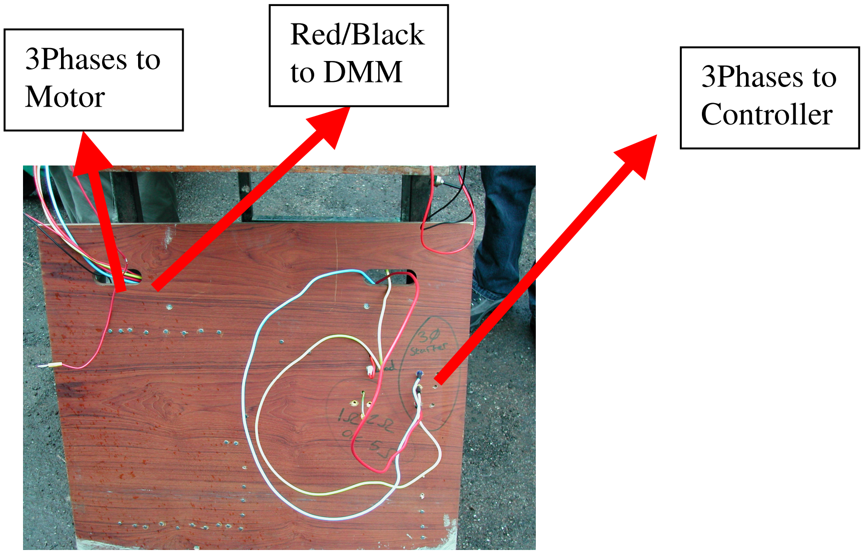
Switching the Enable/Disable switch to DISABLE will switch all pulse width outputs to 0%. Switching the run/100% switch back to 100% will have no effect until the rest button is pressed, or the battery disconnected.

Appendix 8

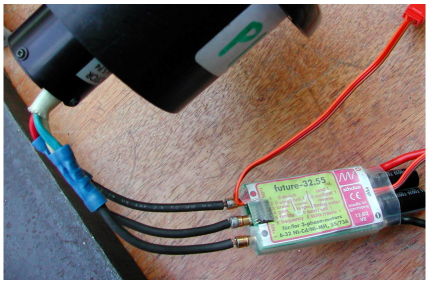
Genset Test Instructions

14.viii Genset Test Instructions

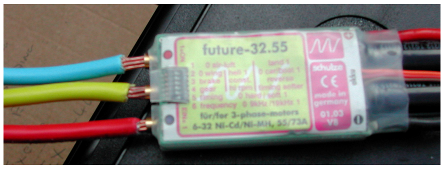
Instructions for use of the Analogue to PWM converter circuit.



1. Connect speed controller to fan motor correctly. Ensure blue cable is connected to socket with blue dot etc.



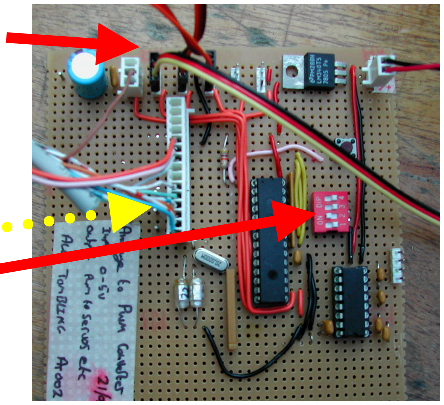
2. Connect Starter cables to second speed controller, ensuring the red and blue wires are reversed in relation to the coding on the controller.



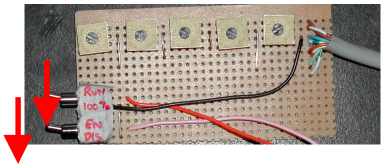
3. Connect the 3-way control cables to the converter circuit. Ensure the black wire is nearest the edge of the board. There are 5 possible positions, numbered 1 to 5, 1 being on the right. Channel 3 should be connected to the GENSET controller; Channel 4 is for the cooling fan controller, and 5 for the servo.

4. Ensure the analogue circuit is connected to this board as shown by the dashed arrow.

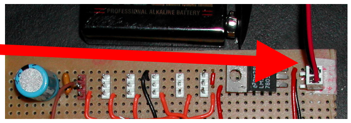
5. Ensure DIPswitch 2 is set to ON.



6. Set switches on analogue board to 100% and DISabled.

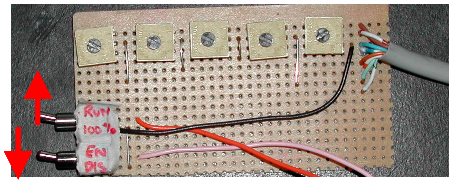


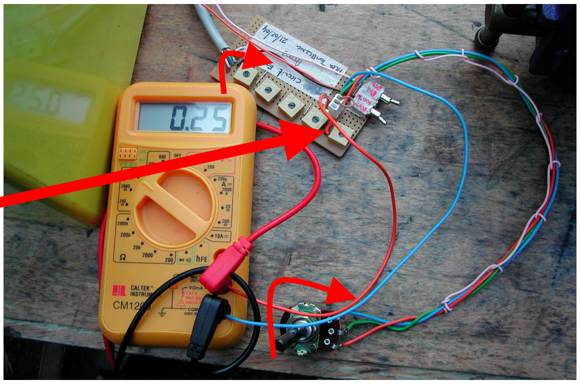
7. Connect 9v battery as shown. Ensure pin nearest board edge is not connected.



8. Connect Speed controller power cables to Battery. If instructions have been correctly followed, the motor should emit a burst of tones, followed by a double beep. If a single beep is heard, disconnect the main power pack, and check through instructions above.

9. Place Run/100% switch into the Run position, leaving the other switch in the DISabled position. The motors should now emit a single beep to acknowledge the 0% position.



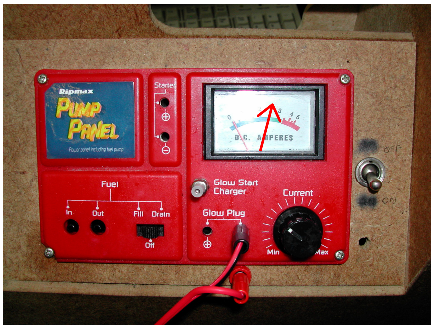


Once the Enable switch is placed into the enable position, the analogue input will be converted to a signal for the controller/servo. The analogue voltages can be adjusted using the respective potentiometer. The arrow indicates channel 1 (see step 2 above). The rotation arrows indicate the direction for 0%.

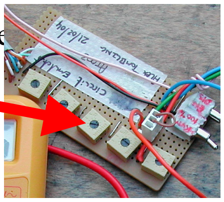
10. Channel 3 should be set to 0%; Channel 4 to 50% approx. Channel 5 uses the handheld pot, and is connected to the DMM output. 5V is interpreted as 100%, 0v as 0%, and is linear. Set channel 5 to 0.25v on DMM for starting the GENSET.

11. Set Enable switch to the Enable position, the cooling fan will run up to 50%, and the servo move to the ready position.

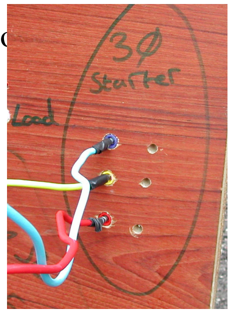
12. Connect Glow plug to Glow start charger, and ensure IC engine is primed with fuel. Switch on Glow plug, and raise current to just under 3A.



13. Turn Pot 3 until the GENSET turns repeatedly and hold. Once the engine fire
to 0%.



14. Remove 3 phase starter wires from starter controller. Do not allow them to t
remove these wires from test board as well.



Switching the Enable/Disable switch to DISABLE will stop the cooling fan, and put the servo into the stop position. This should be used in an emergency, as the cooling fan should be left running to cool the IC after it has stopped.

SHUTDOWN PROCEEDURE

To safely shut down the IC the following steps should be taken.

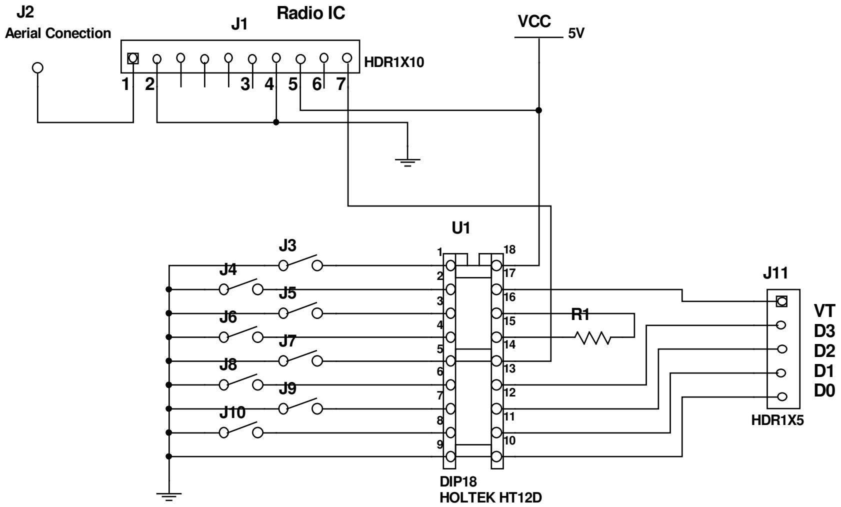
1. Turn channel 5 pot to 0% and ensure throttle is closed.
2. Once engine has stopped, and cooled, switch converter to DISABLE.
3. Disconnect the controllers from the power pack. Do not allow the battery terminals to short together.
4. Disconnect other connections in any order.

To kill the motors at anytime, switch the Enable/Disable switch to disable. Switching the Run/100% switch to 100% will not have any effect until the reset button is pressed or the battery disconnected.

Appendix 9

Radio Control System

14.ix Radio Control Systems

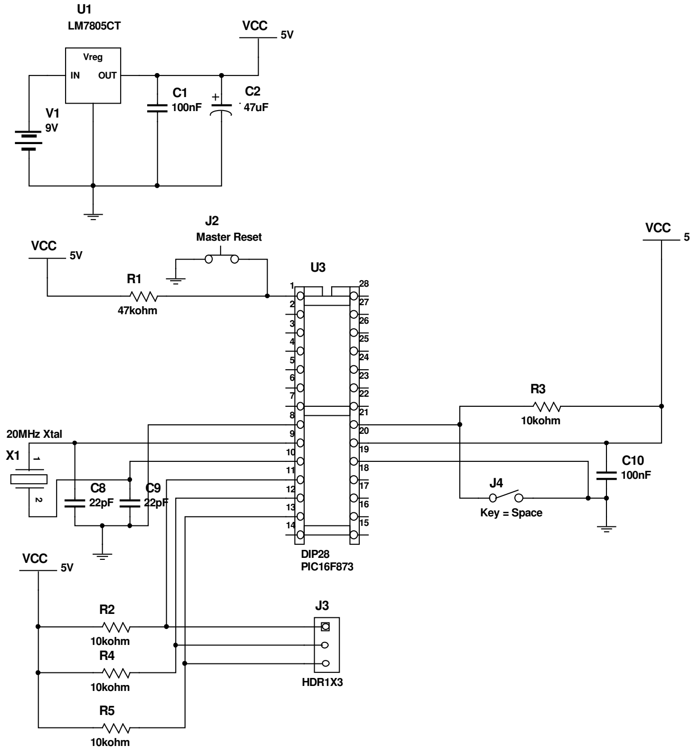


AT006
Radio Control Module (Receiver)
Alex Tombling
Apr 07, 2004

Appendix 10

IMU Emulator Circuit and Code

14.x IMU Emulator code and circuitry



AT005, IMU Emulator using PIC16F873
Alex Tombling
26 Feb 2004.

IMU Emulator PIC16F873 Source code listing.

```
;**********************************************************
;* IMU Emulator for Flying Platform Group project 2004
;**********************************************************
;
;
;
;               Written for Bootloader
;               ----------------------
;;******************************************************************************
;
;       IMU_Emu2.ASM   Source Code for PIC 16F873 Microcontroller
;
;       A.Tombling     Ver 2.0        29/02/2004
;
;       Function...............................................................
;
;               Output signals as per IMU for test purposes
;
;               Input On/Off B0, 1 off, 0 on.
;               Ouput Clock C0, Data C1, Sync C2.
;
;...............................................................................
;
;               Corrected data stream, MSB is LAST!
;..............................................................................
        LIST P=PIC16F873      ; Declare processor type to assembler
        #include "p16f873.inc"
;*******************************************************************************
;
;       Declare lables for registers, constants and variables.....
;
ddca    EQU     H'FF'          ; Data Direction codes
ddcb    EQU     H'FF'          ; A inputs, B inputs, C Outputs
ddcc    EQU     H'00'          ;
timlev  EQU     H'2A'          ; Register for level sent to delay
count1  EQU     H'21'          ; Registers for delay subroutine
count2  EQU     H'22'
;--------------------------------------------------------------------------------
; ----------Start of header for bootloaded prog-----------------
;
        org     0x0000  ; Reset Vector is always at location 00 for this chip!
        clrf    h'03'          ;zero status
        movlw   h'00'          ;high byte of long jump
        movwf   h'0A'          ;zero PCLATH
        goto    Setup1
; ---------------------------------End booted prog--------------
;
;       Startup routine
;
        ORG     04h
Setup1  BCF     STATUS,RP0
        BCF     STATUS,RP1             ; Page 0
        CLRF    PORTA
        CLRF    PORTB
        CLRF    PORTC                  ; Clear all output buffers to initialise
        BSF     STATUS,RP0             ; Page1
        MOVLW   ddcb
        MOVWF   TRISB
        MOVLW   ddcc
        MOVWF   TRISC
        BCF     STATUS,RP0             ; Page0
;
;               Program routines...
;               Detect B0 (Enable) High Do nothing
;               Detect B0 low, Send data
;
;===========================Main Program============================================
;
;               Run setup program
;
start   BTFSC   PORTB,0        ; Check B(0) if 1 set all outputs High
        GOTO    allhi                  ;
;
```

```
;                               ID 00 set to 0V
        MOVLW   H'00'           ; ID bit 0
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'           ; ID Bit 1
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'           ; ID Bit 2
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'           ; ID Bit 3
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC           ; End of ID word
        MOVLW   H'00'
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC           ; End of ignored bits
        MOVLW   H'00'           ; Data bit 0
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'           ; Data bit 1
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'           ; Data bit 2
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'           ; Data bit 3
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'           ; Data bit 4
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'           ; Data bit 5
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'           ; Data bit 6
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'           ; Data bit 7
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC           ; End of first half of dataword
        MOVLW   H'00'           ; Data bit 8
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'           ; Data bit 9
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'           ; Data bit 10
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'         ; Data bit 11
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'         ; Data bit 12
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'         ; Data bit 13
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'         ; Data bit 14
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'         ; Data bit 15
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC         ; End of Datastream for Device 0
        MOVLW   H'FC'         ; Sync Pulse
        MOVWF   PORTC
        MOVLW   H'00'
        MOVWF   PORTC
        CALL    sdelay  ; delay after sync pulse
;
;                             ID 01 set to -10V
        MOVLW   H'00'         ; ID bit 0
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'         ; ID Bit 1
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'         ; ID Bit 2
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'02'         ; ID Bit 3
        MOVWF   PORTC
        MOVLW   H'03'
        MOVWF   PORTC         ; End of ID word
        MOVLW   H'00'
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC         ; End of ignored bits
        MOVLW   H'00'         ; Data bit 0
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'         ; Data bit 1
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'         ; Data bit 2
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'         ; Data bit 3
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'          ; Data bit 4
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'          ; Data bit 5
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'          ; Data bit 6
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'          ; Data bit 7
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC          ; End of first half of dataword
        MOVLW   H'00'          ; Data bit 8
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'          ; Data bit 9
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'          ; Data bit 10
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'          ; Data bit 11
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'          ; Data bit 12
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'          ; Data bit 13
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'          ; Data bit 14
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'02'          ; Data bit 15
        MOVWF   PORTC
        MOVLW   H'03'
        MOVWF   PORTC          ; End of Datastream for Device 1
        MOVLW   H'FC'          ; Sync Pulse
        MOVWF   PORTC
        MOVLW   H'00'
        MOVWF   PORTC
        CALL    sdelay  ; delay after sync pulse

;                              ID 03 set to -7.5v
        MOVLW   H'00'          ; ID bit 0
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'          ; ID Bit 1
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'02'          ; ID Bit 2
        MOVWF   PORTC
        MOVLW   H'03'
        MOVWF   PORTC
        MOVLW   H'02'          ; ID Bit 3
        MOVWF   PORTC
        MOVLW   H'03'
        MOVWF   PORTC          ; End of ID word
        MOVLW   H'00'
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'
```

```
MOVWF   PORTC
MOVLW   H'01'
MOVWF   PORTC
MOVLW   H'00'
MOVWF   PORTC
MOVLW   H'01'
MOVWF   PORTC
MOVLW   H'00'
MOVWF   PORTC
MOVLW   H'01'
MOVWF   PORTC         ; End of ignored bits
MOVLW   H'00'         ; Data bit 0
MOVWF   PORTC
MOVLW   H'01'
MOVWF   PORTC
MOVLW   H'00'         ; Data bit 1
MOVWF   PORTC
MOVLW   H'01'
MOVWF   PORTC
MOVLW   H'00'         ; Data bit 2
MOVWF   PORTC
MOVLW   H'01'
MOVWF   PORTC
MOVLW   H'00'         ; Data bit 3
MOVWF   PORTC
MOVLW   H'01'
MOVWF   PORTC
MOVLW   H'00'         ; Data bit 4
MOVWF   PORTC
MOVLW   H'01'
MOVWF   PORTC
MOVLW   H'00'         ; Data bit 5
MOVWF   PORTC
MOVLW   H'01'
MOVWF   PORTC
MOVLW   H'00'         ; Data bit 6
MOVWF   PORTC
MOVLW   H'01'
MOVWF   PORTC
MOVLW   H'00'         ; Data bit 7
MOVWF   PORTC
MOVLW   H'01'
MOVWF   PORTC         ; End of first half of dataword
MOVLW   H'00'         ; Data bit 8
MOVWF   PORTC
MOVLW   H'01'
MOVWF   PORTC
MOVLW   H'00'         ; Data bit 9
MOVWF   PORTC
MOVLW   H'01'
MOVWF   PORTC
MOVLW   H'00'         ; Data bit 10
MOVWF   PORTC
MOVLW   H'01'
MOVWF   PORTC
MOVLW   H'00'         ; Data bit 11
MOVWF   PORTC
MOVLW   H'01'
MOVWF   PORTC
MOVLW   H'00'         ; Data bit 12
MOVWF   PORTC
MOVLW   H'01'
MOVWF   PORTC
MOVLW   H'02'         ; Data bit 13
MOVWF   PORTC
MOVLW   H'03'
MOVWF   PORTC
MOVLW   H'00'         ; Data bit 14
MOVWF   PORTC
MOVLW   H'01'
MOVWF   PORTC
MOVLW   H'02'         ; Data bit 15
MOVWF   PORTC
MOVLW   H'03'
MOVWF   PORTC         ; End of Datastream for Device 3
MOVLW   H'FC'         ; Sync Pulse
```

```
        MOVWF   PORTC
        MOVLW   H'00'
        MOVWF   PORTC
        CALL    sdelay  ; delay after sync pulse
;                             ID 05 set to -5V
        MOVLW   H'00'         ; ID bit 0
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'02'         ; ID Bit 1
        MOVWF   PORTC
        MOVLW   H'03'
        MOVWF   PORTC
        MOVLW   H'00'         ; ID Bit 2
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'02'         ; ID Bit 3
        MOVWF   PORTC
        MOVLW   H'03'
        MOVWF   PORTC         ; End of ID word
        MOVLW   H'00'
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC         ; End of ignored bits
        MOVLW   H'00'         ; Data bit 0
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'         ; Data bit 1
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'         ; Data bit 2
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'         ; Data bit 3
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'         ; Data bit 4
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'         ; Data bit 5
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'         ; Data bit 6
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'         ; Data bit 7
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC         ; End of first half of dataword
        MOVLW   H'00'         ; Data bit 8
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'         ; Data bit 9
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'          ; Data bit 10
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'          ; Data bit 11
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'          ; Data bit 12
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'          ; Data bit 13
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'02'          ; Data bit 14
        MOVWF   PORTC
        MOVLW   H'03'
        MOVWF   PORTC
        MOVLW   H'02'          ; Data bit 15
        MOVWF   PORTC
        MOVLW   H'03'
        MOVWF   PORTC          ; End of Datastream for Device 5
        MOVLW   H'FC'          ; Sync Pulse
        MOVWF   PORTC
        MOVLW   H'00'
        MOVWF   PORTC
        CALL    sdelay ; delay after sync pulse
;                              ID 07 set to -2.5V
        MOVLW   H'00'          ; ID bit 0
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'02'          ; ID Bit 1
        MOVWF   PORTC
        MOVLW   H'03'
        MOVWF   PORTC
        MOVLW   H'02'          ; ID Bit 2
        MOVWF   PORTC
        MOVLW   H'03'
        MOVWF   PORTC
        MOVLW   H'02'          ; ID Bit 3
        MOVWF   PORTC
        MOVLW   H'03'
        MOVWF   PORTC          ; End of ID word
        MOVLW   H'00'
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC          ; End of ignored bits
        MOVLW   H'00'          ; Data bit 0
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'          ; Data bit 1
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'          ; Data bit 2
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'          ; Data bit 3
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'         ; Data bit 4
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'         ; Data bit 5
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'         ; Data bit 6
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'         ; Data bit 7
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC         ; End of first half of dataword
        MOVLW   H'00'         ; Data bit 8
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'         ; Data bit 9
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'         ; Data bit 10
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'         ; Data bit 11
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'         ; Data bit 12
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'02'         ; Data bit 13
        MOVWF   PORTC
        MOVLW   H'03'
        MOVWF   PORTC
        MOVLW   H'02'         ; Data bit 14
        MOVWF   PORTC
        MOVLW   H'03'
        MOVWF   PORTC
        MOVLW   H'02'         ; Data bit 15
        MOVWF   PORTC
        MOVLW   H'03'
        MOVWF   PORTC         ; End of Datastream for Device 7
        MOVLW   H'FC'         ; Sync Pulse
        MOVWF   PORTC
        MOVLW   H'00'
        MOVWF   PORTC
        CALL    sdelay ; delay after sync pulse
;                             ID 08 set to 0V
        MOVLW   H'02'         ; ID bit 0
        MOVWF   PORTC
        MOVLW   H'03'
        MOVWF   PORTC
        MOVLW   H'00'         ; ID Bit 1
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'         ; ID Bit 2
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'         ; ID Bit 3
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC         ; End of ID word
        MOVLW   H'00'
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC           ; End of ignored bits
        MOVLW   H'00'           ; Data bit 0
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'           ; Data bit 1
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'           ; Data bit 2
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'           ; Data bit 3
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'           ; Data bit 4
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'           ; Data bit 5
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'           ; Data bit 6
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'           ; Data bit 7
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC           ; End of first half of dataword
        MOVLW   H'00'           ; Data bit 8
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'           ; Data bit 9
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'           ; Data bit 10
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'           ; Data bit 11
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'           ; Data bit 12
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'           ; Data bit 13
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'           ; Data bit 14
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'           ; Data bit 15
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC          ; End of Datastream for Device 8
        MOVLW   H'FC'          ; Sync Pulse
        MOVWF   PORTC
        MOVLW   H'00'
        MOVWF   PORTC
        CALL    sdelay ; delay after sync pulse
;                              ID 09 set to +2.5V
        MOVLW   H'02'          ; ID bit 0
        MOVWF   PORTC
        MOVLW   H'03'
        MOVWF   PORTC
        MOVLW   H'00'          ; ID Bit 1
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'          ; ID Bit 2
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'02'          ; ID Bit 3
        MOVWF   PORTC
        MOVLW   H'03'
        MOVWF   PORTC          ; End of ID word
        MOVLW   H'00'
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC          ; End of ignored bits
        MOVLW   H'00'          ; Data bit 0
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'          ; Data bit 1
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'          ; Data bit 2
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'          ; Data bit 3
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'          ; Data bit 4
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'          ; Data bit 5
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'          ; Data bit 6
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'          ; Data bit 7
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC          ; End of first half of dataword
        MOVLW   H'00'          ; Data bit 8
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'          ; Data bit 9
        MOVWF  PORTC
        MOVLW  H'01'
        MOVWF  PORTC
        MOVLW  H'00'          ; Data bit 10
        MOVWF  PORTC
        MOVLW  H'01'
        MOVWF  PORTC
        MOVLW  H'00'          ; Data bit 11
        MOVWF  PORTC
        MOVLW  H'01'
        MOVWF  PORTC
        MOVLW  H'00'          ; Data bit 12
        MOVWF  PORTC
        MOVLW  H'01'
        MOVWF  PORTC
        MOVLW  H'02'          ; Data bit 13
        MOVWF  PORTC
        MOVLW  H'03'
        MOVWF  PORTC
        MOVLW  H'00'          ; Data bit 14
        MOVWF  PORTC
        MOVLW  H'01'
        MOVWF  PORTC
        MOVLW  H'00'          ; Data bit 15
        MOVWF  PORTC
        MOVLW  H'01'
        MOVWF  PORTC          ; End of Datastream for Device 9
        MOVLW  H'FC'          ; Sync Pulse
        MOVWF  PORTC
        MOVLW  H'00'
        MOVWF  PORTC
        CALL   sdelay ; delay after sync pulse
;                             ID 0B set to +5V
        MOVLW  H'02'          ; ID bit 0
        MOVWF  PORTC
        MOVLW  H'03'
        MOVWF  PORTC
        MOVLW  H'00'          ; ID Bit 1
        MOVWF  PORTC
        MOVLW  H'01'
        MOVWF  PORTC
        MOVLW  H'02'          ; ID Bit 2
        MOVWF  PORTC
        MOVLW  H'03'
        MOVWF  PORTC
        MOVLW  H'02'          ; ID Bit 3
        MOVWF  PORTC
        MOVLW  H'03'
        MOVWF  PORTC          ; End of ID word
        MOVLW  H'00'
        MOVWF  PORTC
        MOVLW  H'01'
        MOVWF  PORTC
        MOVLW  H'00'
        MOVWF  PORTC
        MOVLW  H'01'
        MOVWF  PORTC
        MOVLW  H'00'
        MOVWF  PORTC
        MOVLW  H'01'
        MOVWF  PORTC
        MOVLW  H'00'
        MOVWF  PORTC
        MOVLW  H'01'
        MOVWF  PORTC          ; End of ignored bits
        MOVLW  H'00'          ; Data bit 0
        MOVWF  PORTC
        MOVLW  H'01'
        MOVWF  PORTC
        MOVLW  H'00'          ; Data bit 1
        MOVWF  PORTC
        MOVLW  H'01'
        MOVWF  PORTC
        MOVLW  H'00'          ; Data bit 2
        MOVWF  PORTC
        MOVLW  H'01'
        MOVWF  PORTC
        MOVLW  H'00'          ; Data bit 3
        MOVWF  PORTC
        MOVLW  H'01'
        MOVWF  PORTC
        MOVLW  H'00'          ; Data bit 4
        MOVWF  PORTC
        MOVLW  H'01'
        MOVWF  PORTC
        MOVLW  H'00'          ; Data bit 5
        MOVWF  PORTC
        MOVLW  H'01'
        MOVWF  PORTC
        MOVLW  H'00'          ; Data bit 6
        MOVWF  PORTC
        MOVLW  H'01'
        MOVWF  PORTC
        MOVLW  H'00'          ; Data bit 7
        MOVWF  PORTC
        MOVLW  H'01'
        MOVWF  PORTC          ; End of first half of dataword
        MOVLW  H'00'          ; Data bit 8
        MOVWF  PORTC
        MOVLW  H'01'
        MOVWF  PORTC
        MOVLW  H'00'          ; Data bit 9
        MOVWF  PORTC
        MOVLW  H'01'
        MOVWF  PORTC
        MOVLW  H'00'          ; Data bit 10
        MOVWF  PORTC
        MOVLW  H'01'
        MOVWF  PORTC
        MOVLW  H'00'          ; Data bit 11
        MOVWF  PORTC
        MOVLW  H'01'
        MOVWF  PORTC
        MOVLW  H'00'          ; Data bit 12
        MOVWF  PORTC
        MOVLW  H'01'
        MOVWF  PORTC
        MOVLW  H'00'          ; Data bit 13
        MOVWF  PORTC
        MOVLW  H'01'
        MOVWF  PORTC
        MOVLW  H'02'          ; Data bit 14
        MOVWF  PORTC
        MOVLW  H'03'
        MOVWF  PORTC
        MOVLW  H'00'          ; Data bit 15
        MOVWF  PORTC
        MOVLW  H'01'
        MOVWF  PORTC          ; End of Datastream for Device B
        MOVLW  H'FC'          ; Sync Pulse
        MOVWF  PORTC
        MOVLW  H'00'
        MOVWF  PORTC
        CALL   sdelay  ; delay after sync pulse
;                             ID 0D set to 7.5V
        MOVLW  H'02'          ; ID bit 0
        MOVWF  PORTC
        MOVLW  H'03'
        MOVWF  PORTC
        MOVLW  H'02'          ; ID Bit 1
        MOVWF  PORTC
        MOVLW  H'03'
        MOVWF  PORTC
        MOVLW  H'00'          ; ID Bit 2
        MOVWF  PORTC
        MOVLW  H'01'
        MOVWF  PORTC
        MOVLW  H'02'          ; ID Bit 3
        MOVWF  PORTC
        MOVLW  H'03'
        MOVWF  PORTC          ; End of ID word
        MOVLW  H'00'
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC           ; End of ignored bits
        MOVLW   H'00'           ; Data bit 0
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'           ; Data bit 1
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'           ; Data bit 2
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'           ; Data bit 3
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'           ; Data bit 4
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'           ; Data bit 5
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'           ; Data bit 6
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'           ; Data bit 7
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC           ; End of first half of dataword
        MOVLW   H'00'           ; Data bit 8
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'           ; Data bit 9
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'           ; Data bit 10
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'           ; Data bit 11
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'           ; Data bit 12
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'02'           ; Data bit 13
        MOVWF   PORTC
        MOVLW   H'03'
        MOVWF   PORTC
        MOVLW   H'02'           ; Data bit 14
        MOVWF   PORTC
        MOVLW   H'03'
        MOVWF   PORTC
        MOVLW   H'00'           ; Data bit 15
```

```
       MOVWF   PORTC
       MOVLW   H'01'
       MOVWF   PORTC          ; End of Datastream for Device D
       MOVLW   H'FC'          ; Sync Pulse
       MOVWF   PORTC
       MOVLW   H'00'
       MOVWF   PORTC
       CALL    sdelay  ; delay after sync pulse
;                             ID 0E set to 10V
       MOVLW   H'02'          ; ID bit 0
       MOVWF   PORTC
       MOVLW   H'03'
       MOVWF   PORTC
       MOVLW   H'02'          ; ID Bit 1
       MOVWF   PORTC
       MOVLW   H'03'
       MOVWF   PORTC
       MOVLW   H'02'          ; ID Bit 2
       MOVWF   PORTC
       MOVLW   H'03'
       MOVWF   PORTC
       MOVLW   H'00'          ; ID Bit 3
       MOVWF   PORTC
       MOVLW   H'01'
       MOVWF   PORTC          ; End of ID word
       MOVLW   H'00'
       MOVWF   PORTC
       MOVLW   H'01'
       MOVWF   PORTC
       MOVLW   H'00'
       MOVWF   PORTC
       MOVLW   H'01'
       MOVWF   PORTC
       MOVLW   H'00'
       MOVWF   PORTC
       MOVLW   H'01'
       MOVWF   PORTC
       MOVLW   H'00'
       MOVWF   PORTC
       MOVLW   H'01'
       MOVWF   PORTC          ; End of ignored bits
       MOVLW   H'02'          ; Data bit 0
       MOVWF   PORTC
       MOVLW   H'03'
       MOVWF   PORTC
       MOVLW   H'02'          ; Data bit 1
       MOVWF   PORTC
       MOVLW   H'03'
       MOVWF   PORTC
       MOVLW   H'02'          ; Data bit 2
       MOVWF   PORTC
       MOVLW   H'03'
       MOVWF   PORTC
       MOVLW   H'02'          ; Data bit 3
       MOVWF   PORTC
       MOVLW   H'03'
       MOVWF   PORTC
       MOVLW   H'02'          ; Data bit 4
       MOVWF   PORTC
       MOVLW   H'03'
       MOVWF   PORTC
       MOVLW   H'02'          ; Data bit 5
       MOVWF   PORTC
       MOVLW   H'03'
       MOVWF   PORTC
       MOVLW   H'02'          ; Data bit 6
       MOVWF   PORTC
       MOVLW   H'03'
       MOVWF   PORTC
       MOVLW   H'02'          ; Data bit 7
       MOVWF   PORTC
       MOVLW   H'03'
       MOVWF   PORTC          ; End of first half of dataword
       MOVLW   H'02'          ; Data bit 8
       MOVWF   PORTC
       MOVLW   H'03'
```

```
        MOVWF   PORTC
        MOVLW   H'02'          ; Data bit 9
        MOVWF   PORTC
        MOVLW   H'03'
        MOVWF   PORTC
        MOVLW   H'02'          ; Data bit 10
        MOVWF   PORTC
        MOVLW   H'03'
        MOVWF   PORTC
        MOVLW   H'02'          ; Data bit 11
        MOVWF   PORTC
        MOVLW   H'03'
        MOVWF   PORTC
        MOVLW   H'02'          ; Data bit 12
        MOVWF   PORTC
        MOVLW   H'03'
        MOVWF   PORTC
        MOVLW   H'02'          ; Data bit 13
        MOVWF   PORTC
        MOVLW   H'03'
        MOVWF   PORTC
        MOVLW   H'02'          ; Data bit 14
        MOVWF   PORTC
        MOVLW   H'03'
        MOVWF   PORTC
        MOVLW   H'00'          ; Data bit 15
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC          ; End of Datastream for Device E
        MOVLW   H'FC'          ; Sync Pulse
        MOVWF   PORTC
        MOVLW   H'00'
        MOVWF   PORTC
        CALL    sdelay ; delay after sync pulse
;                              ID 0F set to 0V
        MOVLW   H'02'          ; ID bit 0
        MOVWF   PORTC
        MOVLW   H'03'
        MOVWF   PORTC
        MOVLW   H'02'          ; ID Bit 1
        MOVWF   PORTC
        MOVLW   H'03'
        MOVWF   PORTC
        MOVLW   H'02'          ; ID Bit 2
        MOVWF   PORTC
        MOVLW   H'03'
        MOVWF   PORTC
        MOVLW   H'02'          ; ID Bit 3
        MOVWF   PORTC
        MOVLW   H'03'
        MOVWF   PORTC          ; End of ID word
        MOVLW   H'00'
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC          ; End of ignored bits
        MOVLW   H'00'          ; Data bit 0
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'          ; Data bit 1
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'          ; Data bit 2
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'          ; Data bit 3
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'          ; Data bit 4
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'          ; Data bit 5
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'          ; Data bit 6
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'          ; Data bit 7
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC          ; End of first half of dataword
        MOVLW   H'00'          ; Data bit 8
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'          ; Data bit 9
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'          ; Data bit 10
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'          ; Data bit 11
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'          ; Data bit 12
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'          ; Data bit 13
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'          ; Data bit 14
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC
        MOVLW   H'00'          ; Data bit 15
        MOVWF   PORTC
        MOVLW   H'01'
        MOVWF   PORTC          ; End of Datastream for Device F
        MOVLW   H'FC'          ; Sync Pulse
        MOVWF   PORTC
        MOVLW   H'00'
        MOVWF   PORTC
        CALL    sdelay ; delay after sync pulse
        GOTO    start
;
;===========================Sub Programs============================================
;                      Set all Outputs High
;
allhi   MOVLW   H'FF'
        MOVWF   PORTC
        GOTO    start
;===========================Delay Routine=========================================
;              This routine should take 1ms + level to complete
;
;              sdelay takes 1ms + count3 (count3= 0-1ms)
;
;
sdelay
        MOVLW   timlev
        MOVWF   count2                 ;and copy it into the COUNT registers
;
D1      MOVLW   H'02'                  ;Load  into the working register
        MOVWF   count1
        DECFSZ  count1,F               ;Decrement the counter and check for zero
        GOTO    $-1                    ;If not zero, jump back one instruction
;
        DECFSZ  count2,F
        GOTO    D1
;
        RETURN                         ;and finally, return from the subroutine.
;
;**********************************************************************************
        END
```

Appendix 11

Meeting Agenda and Minutes,

14.xi Meeting agenda and minutes.

University of Exeter
School of Engineering

Agenda

for the
Flying Platform Project Meeting 37a
to be held on

Thursday 18 March 2004

commencing at **10.00** am, room 102a.

Circulated to: Liam Dushynsky (LD), Richard Forder (RF), Richard Holbrook (RH), Rebecca Hughes (RCH), Kevin Lowis (KL), James Mackenzie-Burrows (JMB), Jody Muelaner (JM), Chris Poczka (CP), Alex Tombling (AT).

15/03 37a.1		Apologies for absence
15/03 37a.2	i	Minutes of previous meeting
	ii	Matters arising
15/03 37a.3	i	Chairman's Report
	ii	Overall project review
15/03 37a.4		Secretary's Report
15/03 37a.5		Treasurer's Report
15/03 37a.6	i	Control / IMU
	ii	Propulsion
	iii	Electrical Power Systems
	iv	Platform structure
15/03 37a.7		Tasks For Easter
15/03 37a.8		Any Other Business
15/03 37a.9		Date, Time and Place of next meeting

Time Meeting Started____________ Time Meeting Finished__________

UNIVERSITY of EXETER

Fourth Year Group Project

FLYING PLATFORM

Minutes

Date: 18/03/04
Room: 102A
Meeting: 37A

Group Members		**Present**	**Apologies**
Liam Dushynsky	LD		✓
Richard Forder	RF	✓	(arrived at 10:12)
Richard Holbrook	RH	✓	
Rebecca Hughes	RCH	✓	(arrived at 10:28)
Kevin Lowis	KL	✓	
James Mackenzie-Burrows	JMB	✓	
Jody Muelaner	JM	✓	(arrived at 10:36)
Chris Poczka	CP		
Alex Tombling	AT	✓	

Due to the absence of LD, AT was appointed acting chair for this meeting.
The acting chair welcomed all to the flying platform meeting.

18/3 37A.1 Apologies:

18/3 37A.1.1 Apologies were received in advance from LD who was absent due to a job interview.

18/3 37A.2 Previous Minutes:

18/3 37A.2.1 The minutes from meeting 35 were accepted by all who were present. The minutes from meetings 36 and 37 are to be signed off on Friday, so all are requested to read them.

18/3 37A.2.2 AT has uploaded the report on the radio communication link to WebCT.

18/3 37A.2.3 KL, RF and AT have discussed temperature sensors.

18/3 37A.2.4 KL has ordered the op-amps for the control circuit.

18/3 37A.2.5 The genset testing is still not complete.

18/3 37A.2.6 KL has not been able to start the construction of control circuit as he is waiting on input from others.

18/3 37A.2.7 RF is the only member left to get the out of hours form signed by MAJ and GAL.

18/3 37A.2.8 RH has not been able to investigate options for vibration mounts as of yet.

18/3 37A.2.9 Since the last meeting, there have been 2 tests with the cardboard duct on the central ducted fan unit.

18/3 37A.3 Chairman's Report:

18/3 37A.3.1 On behalf of LD, AT emphasised that the project is a very long way behind. Given the fact that there has not been a 100% attendance at meetings for weeks, he asked for more group commitment. He also advised that when people call meetings, they need to turn up.

18/3 37A.3.2 AT mentioned that all access to the workshop over the Easter break is forbidden. This means that testing is not possible. LD is to investigate this.

18/3 37A.3.3 RF asked about access to the Kay Labs. LD is to investigate this as well.

18/3 37A.4 Secretary's Report:

18/3 37A.4.1 The secretary had no matters which needed discussing at the meeting.

18/3 37A.5 Treasurer's Report:

18/3 37A.5.1 The treasurer had no matters which needed discussing at the meeting.

18/3 37A.6 Matters Discussed:

i) Control / IMU:

18/3 37A.6.1 KL will start building the control circuit today but needs to investigate comparators prior to this. He requested discussing this with AT and RF at some point during the day.

18/3 37A.6.2 RF explained that he now has a new monostable circuit for use in the IMU circuit. He will speak to Mike Horwood today about this.

18/3 37A.6.3 KL, RCH and AT spoke to Mustafa Aziz about control theory and application which helped to clarify some issues.

ii) Propulsion

18/3 37A.6.4 RH explained that the test of the cardboard duct was carried out on Monday. During this, the exhaust coupling broke. After a discussion with MAJ, RH bought a section of high temperature hose. The experiment was run again on Wednesday afternoon. The hose melted. Therefore, RH and AT have been investigating alternative options for securing the exhaust.

18/3 37A.6.5 RH said that he ordered the new cylinder head for the engine. He informed KL that it was about £70.

iii) Power Systems

18/3 37A.6.6 AT has accepted the role of chair for this subgroup.

18/3 37A.6.7 The genset was tested on Tuesday but the coupling between the motor and the engine broke. A new coupling has been order which is expected today. This coupling is stronger and therefore should withstand a higher level of torque.

18/3 37A.6.8 Genset testing will continue today at 12 noon.

18/3 37A.6.9 AT informed the group that the current demand from propulsion is in the region of 100A.

iv) Structure

18/3 37A.6.10 No matters were discussed as the Structures group was absent.

18/3 37A.7 Tasks for Easter:

RCH Continue with her work on control. However, she needs values from CP.

KL Build and test the control system.

RH Depends on the outcome of whether the group can gain access to the workshop over the Easter. A final decision will be reached by tomorrow.

RF Continue with the algorithm work on the IMU.

JMB Design filter and power distribution circuit.

AT Design and build the data communication link. If access can not be made to the workshop, then he will build the structure as well.

JM Carry on with custom fan design.

18/3 37A.8 Any Other Business

18/3 37A.8.1 RCH will not be able to make the meeting tomorrow and apologises in advance.

18/3 37A.9 Task List:

Task	Action	Deadline
Test genset.	RH, JMB, AT	18/03/04
Speak to Mike Horwood about monostable circuit.	RF	18/03/04
Liaise about comparators.	KL, AT, RF	18/03/04
Get out of hours form signed by MAJ, GAL.	RF	18/03/04
Update the power link interface diagram.	AT, JMB	19/03/04
Get some definite answers about access to the building over Easter.	LD	19/03/04
Technical drawings of the structure.	CP	19/03/04
Provide values for the control system.	CP	19/03/04
Read minutes.	**ALL**	19/03/04

18/3 37A.9 Next Meeting:

The next group only meeting is currently **UNDECIDED**.
The next formal meeting with guests will be at **10:00am** on Friday **19/3/2004** in Room **102A.**

Meeting closed at 10:47am.

15. References

[1] www.unrealaircraft.com/wings/hillier_platform.php Many unusual aircraft

[2] www.rctoys.com RC Toys official website, Canada

[3] Gornall N, et al (2003). *The Design and Development of a Flying platform.* University of Exeter, Exeter, UK. P15.

[4] WeMoTec (Date unknown), Midi Ducted fan instructions.

[5] www.plettenberg-motoren.com Plettenberg official website

[6] Schulze Elektronik (2002), future-u speed controller instructions

[7] Hart B. (1997). *Introduction to analogue electronics*, Arnauld, Hodder Headline Group, London. Pp212-213

[8] Floyd T. L. (2001). *Electronics Fundamentals – Circuits, Devices, and Applications (fifth edition),* Prentice Hall, New Jersey, USA. Pp998-1000

[9] Tulloch N. (2003). *The Design and Development of a Flying platform.* University of Exeter, Exeter, UK. P33.

[10] www.snazzysigns.co.uk/frenchyelectronics/projects/circuitdiagrams/ports/spp.html Information on the SPP Parallel port.

[11] Microchip Technology Inc (2001), PIC16F87x datasheet

[12] www.microchip.com Microchip Technology Inc official website

[13] www.ic-prog.com Website of free IC programmer

[14] www.ehl.cv/pic Source of Freeware "PIC Downloader 1.08"

[15] Hughes A. (1995). *Electric motors and drives – Fundamentals, types and applications (second edition),* Butterworth Heinmenn Oxford, UK. P.84.

[16] Hughes R. (2004). *The Design and Development of a Flying Platform.* University of Exeter, Exeter, UK. Chapter 3.

[17] Tulloch N. (2003). *The Design and Development of a Flying platform.* University of Exeter, Exeter, UK. P28.

[18] Holbrook R. (2004). *The Design and Development of a Flying Platform.* University of Exeter, Exeter, UK. pp11-22

[19] Ibid. pp24-35

[21] O.S. (Date unknown). OS MAX 91 Datasheet

[21] Personal Correspondence, Mr B Miller, Compass House Ltd, Rotherfield, UK.

[22] www.hobby-lobby.com Hobby Company stocking motors etc.

[23] Personal E-Mail, Diane Plettenberg. Plettenberg Motoren, Germany.

[24] Mackenzie-Burrows, J. (2004). *The Design and Development of a Flying Platform.* University of Exeter, Exeter, UK.

[25] www.maplin.co.uk Electronic component website

[26] Dr W Houghton lecture notes (2003). *Communication systems*. University of Exeter, Exeter, UK.

[27] Birnie P, Fairall, J (1999). *An Introduction to Low Power Radio.* RF Solutions Ltd, Lewes, UK. P5

[28] Ibid. P6

[29] Dushynsky L, et al (2004). *The Design and Development of a Flying platform.* University of Exeter, Exeter, UK.