APPENDIX A: Design Completion Form

THAMES

	Component of system/Milestone	Supervisor	Time/Date	Comments (all/part/none working; protoboard/constructed)
X	Understand RC Rx output			
	Get basic motor movement via ESC	QM	12:30 6th	Signal generator → ~~square~~ wave → ESC → motor. pulse
X	Successful signal generator test of PPM Decoder			
X	Successful RC test of PPM Decoder			
✓	Successful serial telemetry test (transmit), in software			
✓	Successful serial telemetry test (receive), in software			
	Successful transmission and receive via Bluetooth, hardware			
	Get raw data out of the IMU	QM	12:30 6th	IMU → Arduino → serial → PC program - data decoded and displayed 200Hz ← on breadboard
	Establish communication (I2C or SPI)	QM	"	"
	Process measurements with DMP or external filter	QM	"	filter running on arduino Mahony filter from library
✓	Controller outputs a ~~PPM~~/PWM signal for ESCs	QM	10th 10:00	14 discrete levels from terminal → 11 Matlab → ESCs & motors.
X	Controller receives converted PPM data			
✓	Controller receives serial telemetry	QM	10th 10:00	see above. terminal → 11 matlab. used.
	Test power distribution board	QM	10.37 09.03.17	
✓	Test power distribution board with motors	QM	10th 10:00	Motors running from ESCs from battery.
	Flight/ ~~Stable Flight~~	SRG	14:10 13/3	FLIGHT OFF TABLE DEMONSTRATED BY VIDEO. CONTROL NEEDS SOME WORK
	Target flight time achieved (1/2/3 minutes)			
	Hooked on to and carried empty lunchbox			
	Lift target weight			

Milestones finalised by supervisor: S. GUNN ..... Signed ..... Date 13/3/17

Prototype hardware handed over to: ..... Signed ..... Date

Other items returned to Lab support hatch and checked by: ..... Signed ..... Date

APPENDIX B: Project Completion Form

Appendix E: Project Completion Form

Cost Estimates

Please give detailed calculations and estimates of the overall cost of your actual design below. Take care to include person-hour estimates for your software, board production and debugging, as well as your components and consumables. You should also estimate the production cost of your final unit (you may assume a large quantity are to be produced), the market price and determine how many need to be sold to be profitable. Account for any differences between the actual values and the values given in your original project proposal form.

Total cost for prototype £98

Estimated cost for manufacture £70 with mass discount of ~30%.

Person-hours 450 x 75 £33750

Total cost £137,500

Market value £150 + VAT → £180 £80 profit per unit → 1720 units sold to make a profit

Differences: More person-hours

More expensive propellers, motors, power distribution board

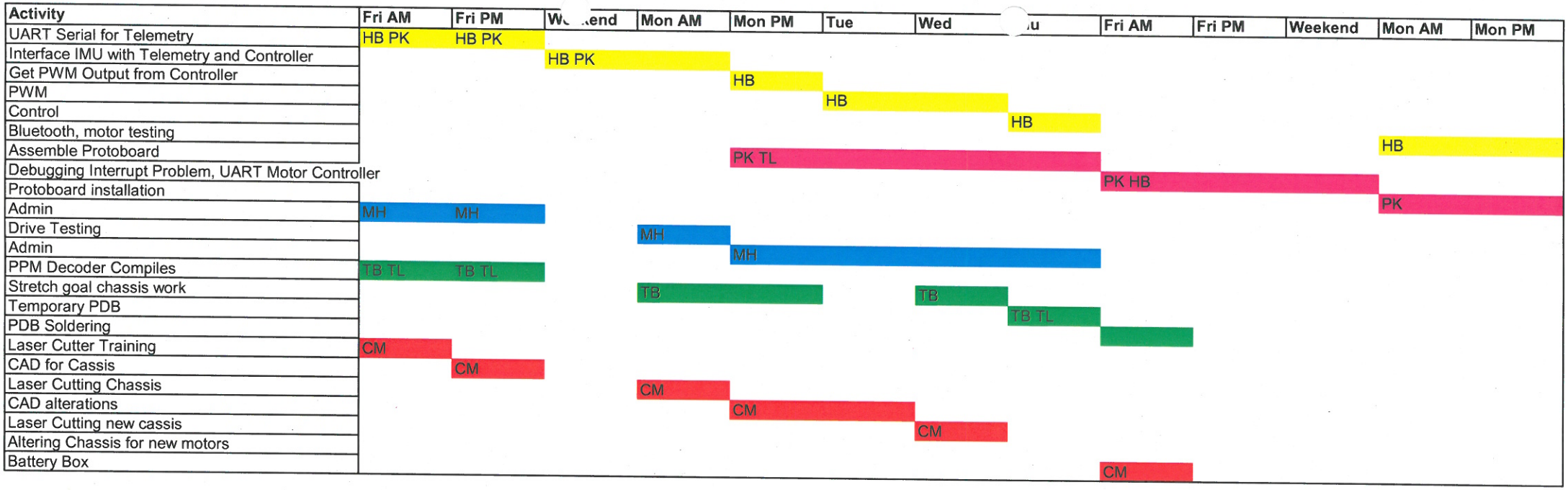
Design Changes

Briefly summarise any design changes your team had to make to the original design proposal, in order to get your system to work. Do not go into vast detail, as it is anticipated that this will be done by the individuals responsible for these components of the design in the formal report.

- Removal of PPM-to-Digital Decoder due to Spectrum not arming, replaced with bluetooth module.
- Moved IMU interface to an Arduino Mini to remove compute stress from the main controller.
- Added a MSP microcontroller to fix issue with data integrity because of a high interrupt ~~rate~~ rate on the controller. (Motor control)*
- Holes from chassis removed due to restrictions on the laser cutter hardware, used tape for mounting instead.

* MSP used to control motors via UART, also a better choice due to 6 COM outputs from 2 * 16-bit timers vs 4 COM on 2 8-bit timers on the mbed.

Actual


Activity
Fri AM
Fri PM
Weekend
Mon AM
Mon PM
Tue
Wed
Fri AM
Fri PM
Weekend
Mon AM
Mon PM
UART Serial for Telemetry
Interface IMU with Telemetry and Controller
Get PWM Output from Controller
PWM
Control
Bluetooth, motor testing
Assemble Protoboard
Debugging Interrupt Problem, UART Motor Controller
Protoboard installation
Admin
Drive Testing
Admin
PPM Decoder Compiles
Stretch goal chassis work
Temporary PDB
PDB Soldering
Laser Cutter Training
CAD for Cassis
Laser Cutting Chassis
CAD alterations
Laser Cutting new cassis
Altering Chassis for new motors
Battery Box
HB PK
HB PK
HB PK
HB
HB
HB
HB
PK TL
PK HB
PK
MH
MH
MH
MH
TB TL
TB TL
TB
TB
TB TL
CM
CM
CM
CM
CM
CM

Discrepancy in Project Activities

Comment on any major differences between the planned and actual project activities.

Transmitter and receiver did not arrive

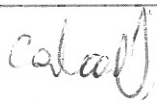
Unexpected activities had to be completed

Broken arm on chassis

Kept to plan reasonably well

Assessment of Effort

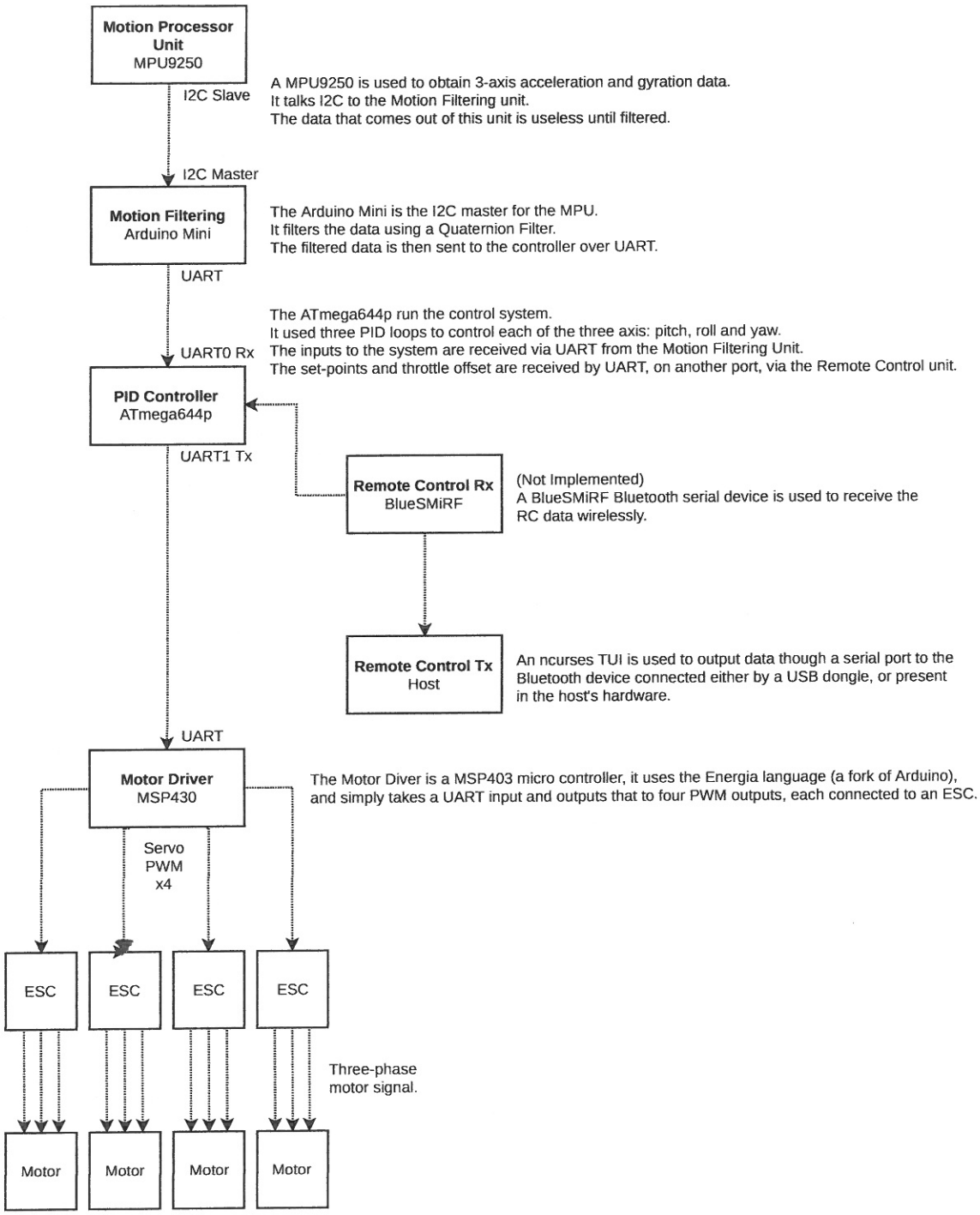
The table below will be used as an indication how team marks should be allocated across the team.

Name	Signature	% of effort
Matt Hunter		15
Harry Beadle		20
Petros Kargologiannis		20
Trong Luong		15
Tim Bills		15
Callum Marshall		15

If the breakdown is not equal please provide a short explanation below:

Harry and Petros were focused on software

APPENDIX C: Circuit Diagrams


Motion Processor Unit
MPU9250
I2C Slave
A MPU9250 is used to obtain 3-axis acceleration and gyration data.
It talks I2C to the Motion Filtering unit.
The data that comes out of this unit is useless until filtered.
I2C Master
Motion Filtering
Arduino Mini
The Arduino Mini is the I2C master for the MPU.
It filters the data using a Quaternion Filter.
The filtered data is then sent to the controller over UART.
UART
The ATmega644p run the control system.
It used three PID loops to control each of the three axis: pitch, roll and yaw.
The inputs to the system are received via UART from the Motion Filtering Unit.
The set-points and throttle offset are received by UART, on another port, via the Remote Control unit.
UART0 Rx
PID Controller
ATmega644p
UART1 Tx
Remote Control Rx
BlueSMiRF
(Not Implemented)
A BlueSMiRF Bluetooth serial device is used to receive the RC data wirelessly.
Remote Control Tx
Host
An ncurses TUI is used to output data though a serial port to the Bluetooth device connected either by a USB dongle, or present in the host's hardware.
UART
Motor Driver
MSP430
The Motor Diver is a MSP403 micro controller, it uses the Energia language (a fork of Arduino), and simply takes a UART input and outputs that to four PWM outputs, each connected to an ESC.
Servo
PWM
x4
ESC
ESC
ESC
ESC
Three-phase
motor signal.
Motor
Motor
Motor
Motor

APPENDIX D: Software Listings

D4 Code Listings

Team Thames

March 17, 2017

Contents

1 ThamesControl

1.1 Communications

1.1.1 Header

```
1  /*
2
3  Harry Beadle
4  D4 Thames
5  Communications (comms.h)
6
7  Headerfile for UART ISRs and initalisation.
8
9  */
10
11 #ifndef _COMMS_H_
12         #define _COMMS_H_
13
14         #include <avr/io.h>
15         #include <avr/interrupt.h>
16
17         #include "rc-symbols.h"
18         #include "mpu-symbols.h"
19         #include "drone.h"
20
21         #define BAUD 9600
22
23         // MPU Decoder Communications
24         uint8_t temp_byte, control_byte, low_byte, high_byte, bad_control;
25         ISR(USART0_RX_vect);
26
27         // Remote Controller Communications
28         enum {control, high, low} state, nstate;
29         uint8_t temp_byte_rc, control_code, data_byte;
30         ISR(USART1_RX_vect);
31
32         #include "../comms.c"
33
34 #endif
```

1.1.2 Source

```
1  /*
2
3  Harry Beadle
4  D4 Thames
5  Communications (comms.c)
6
7  Input                    Perf    Pin
8  ------------------------ ------- -------
9  UART from MPU Decoder    UART0   PD0
10 UART from Controller     UART1   PD2
11 UART to Motor Controller UART1   PD3
12
13 */
14
15 #include "inc/comms.h"
16
17 void init_comms(void)
18 {
19         // Initialise UART0 and UART1
20         // Set Badu Rate.
```