**School of Electrical and Electronic Engineering**



Embedded Systems Project

DESIGN REPORT #1

Title: ?

Group Number: ?

|  |  |  |
| --- | --- | --- |
| Group members name: | ID Number | I confirm that this is the group’s own work. |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Tutor: Click here to enter text.

Date: Click here to enter a date.

Contents

[1. Introduction 1](#_Toc525895373)

[2. Motor characterisation 1](#_Toc525895374)

[3. Load measurements 1](#_Toc525895375)

[4. Gear ratio selection 1](#_Toc525895376)

[5. Summary 1](#_Toc525895377)

[6. References 1](#_Toc525895378)

# Introduction

* See the chapter on Reports (near the start of the ESP Procedures Handbook).
* See the specific marking scheme for this section of the report.

You may want to reference books and papers that you have researched. Give their reference like this [1] (see the References section at the end of this document).

# Motor characterisation

* See the chapter on Reports (near the start of the ESP Procedures Handbook).
* See the specific marking scheme for this section of the report.

# Load measurements

* See the chapter on Reports (near the start of the ESP Procedures Handbook).
* See the specific marking scheme for this section of the report.

# Gear ratio selection

* See the chapter on Reports (near the start of the ESP Procedures Handbook).
* See the specific marking scheme for this section of the report.

# Summary

* Design recommendations
* Summary of key results and assumptions.

# References

1. See the section on Citations and Referencing Styles in the ESP Procedures Handbook.

Make sure that you have **read the top** of the marking scheme to look for report length etc.

Make sure that you have **read the bottom** of the marking scheme for Presentation and Penalties.

Remember to update your table of contents before submitting the report.

Aim to submit the report long before the deadline, to mitigate last minute problems with the internet and with Blackboard.

The AEAT-601B-F06 measures motion and direction by using two magnetic sensors that measure change across a disk and is included with the motor. When there’s movement, if a rising edge is detected and count is incremented. This allows the measurement of wheel speed using the counts per second. The detection of direction uses two sensors, A and B, that are placed so that when there’s motion, they detect a rising edge at a 90 ° phase difference. A will detect first when there’s anticlockwise movement and B will detect first when there’s clockwise movement. To use the quadrature interface mode, pin PA\_5 and pin PB\_3 will be used to input channel A and B, 5V connected to +5V and GND to GND both on CN6 and the index is not needed so it will remain disconnected **[1,2]**. The TIM 2 timer peripheral will increment the CNT register every rising edge without disrupting the CPU and we use a software timed interrupt to sample the number of counts every specified period to get the count rate. This can be used to calculate , angular speed using the counts per revolution, where then ω can then be used to calculate wheel speed in using wheel radius and gear ratio.

The motor drive board has a current sensing circuit that is used to input voltage across a 0.1Ω resistor, in series with the motors, into the microcontroller for measurement **[3]**. First, The APLC-784 will amplify the voltage of this resistor using the isense+ and isense- paths. The gain being determined by VDDin of the amplifier which is fixed by the voltage regulator LM3480 – 5 at 5V if the voltage input is above 6 V **[4]**, to get a gain of 8.029 in order to get an output in the range of 0 – 5V for currents of 0 – 1.4 A. The microcontroller will receive these measurements through Analog in pins Pin 10 Avago A(+) of jp1a to PB\_0 and Pin 9 Avago A(–) of jp1a to PA\_4 for Motor right, Pin 12 Avago B+ of jp1a to PA\_0 and Pin 11 Avago B- of jp1a to PA\_1 for Motor left **[5,6]**.To get an accurate relationship constant between voltage and current, we have to collect data of current through motor turned on using an Ammeter and voltage using a volt meter across TP20 and TP18 for motor 1 for example, plot it and find the gradient. The software can then calculate the motor current from voltage input multiplying the gradient by input voltage.

The DS2781 chip will measure the current and voltage from the battery, place it in a register and output it to the microcontroller upon a software command. The current is measured across the 0.01 Ω resistor called Rsns, and the voltage is measured at Vin, where voltage of battery is inputted to ground **[7]**. The voltage is measured down to a resolution of 9.76 mV and up to a maximum of +9.9902 V and the 10-bit voltage register is updated every 440 ms **[7]**. The chip is capable of reading the current down to a resolution of 1.56 μV/Rsns, therefore the resolution is 156 μA, and up to a maximum of +- 5.11 A as 16 bits are used for current measurement with the current measurement is updated every 3.515 s. The microcontroller Digital in PB\_8 pin will connect to drive board pin 8 of Jp1a [5,6]. So, Using the DS2781 and the one wire method classes, software can read in the battery current and voltage.

The HM – 10 is a BLE module that allows the microcontroller to connect to other BLEs using a unique service profile the HM-10 in which is a collection of services with different purposes. The HM-10 capabilities include the ability to run at very high speeds of 32 MHz, allow both Slave and Master roles, send 20-bit character words ie. Hello would be 5 characters and can be put in sleep mode when not in use where it consumes much less power [8]. The HM-10’s ability to act as slave allows the buggy to be controlled from another and only one BLE, our phones, that will be master to send a command to be processed by software of the microcontroller. To control the BLE in software, we use AT commands. The HM – 10 uses 32-bit AT commands that are sent to the module by software serially through the UART strip and software can then control the module using SoftSerial.h [9]. To connect the HM-10 to the STM32, CN6 3.3 V to VCC, CN6 GND to GND, PA\_2 to TX and PA\_3 to RX are used [6,8].

Motor A = right

Motor B = left

To use the quadrature interface mode, pin PA\_5 and pin PB\_3 will be used to input channel A and B, 5V connected to +5V and GND to GND both on CN6 and the index is not needed so it will remain disconnected

The microcontroller will connect to the pins Pin 10 Avago A(+) of jp1a to PC\_2 and Pin 9 Avago A(–) of jp1a to PC\_3 for Motor right, Pin 12 Avago B+ of jp1a to PC\_4 and Pin 11 Avago B- of jp1a to PC\_5 for Motor left **[?]**.

The microcontroller pin PB\_8 will connect to pin 8 of Jp1a belonging DS2781.

The HM-10 connections to the STM32 are CN6 3.3 V to VCC, CN6 GND to GND, PA\_2 to TX and PA\_3 to RX.

We need Slave because it stops interference as slave can only connect to one master and it has to exchange data at the same rate as master.

LQFP64

References:

Ref 2: [1] pg 65 tech handbook and

[2] Available at: <https://www.st.com/resource/en/datasheet/DM00086815.pdf> table 8

[3] Ref 3: pg 74 tech handbook

[4] Ref 4: Texas instruments (2015, September) LM3480 Datasheet available at Available at: <http://www.ti.com/lit/ds/symlink/lm3480.pdf>

[5] Ref 5: motor drive board data sheet and micro data sheet <https://os.mbed.com/platforms/ST-Nucleo-F401RE/#board-pinout>

[7] Ref 7: ds2781 datasheet

Available at: <https://datasheets.maximintegrated.com/en/ds/DS2781.pdf>

Ref 8: ds2781 datasheet

[6] Ref 9: micro and drive board data sheet <https://os.mbed.com/platforms/ST-Nucleo-F401RE/#board-pinout>

[8] Ref 10: martyncurrey website <http://www.martyncurrey.com/hm-10-bluetooth-4ble-modules/>

Ref 12: Micro and martyncurrey website

Available at:

<https://os.mbed.com/platforms/ST-Nucleo-F401RE/#board-pinout>

Available at:

<http://www.martyncurrey.com/hm-10-bluetooth-4ble-modules/>

Ref [9]: Pg 72 – 73 of Technical handbook

Pg 72 – 73 of Technical handbook

Blackboard/Embeddedsysytemsproject/hardware and circuitry/Motor drive board/ drive board connections

Blackboard/Microcontroller engineering II/laboratories/Technical documentation/UoM documents/ Pin out table

[1] Podd,F (2018-2019). ESP Technical Handbook: University of Manchester. 65

[2] Microcontroller pins definitions table 8, Available at: <https://www.st.com/resource/en/datasheet/DM00086815.pdf>

[3] Podd,F (2018-2019). ESP Technical Handbook: University of Manchester. 74

[4] Ref 4: Texas instruments (2015, September) LM3480 Datasheet available at Available at: <http://www.ti.com/lit/ds/symlink/lm3480.pdf>

[5] Drive board connections Available at: <https://online.manchester.ac.uk/bbcswebdav/pid-6247170-dt-content-rid-26307173_1/courses/I3027-EEEN-21000-1181-1YR-027927/Motor%20Drive%20board%20connections%281%29.pdf>

[6] Ref 9: microcontroller pinout <https://os.mbed.com/platforms/ST-Nucleo-F401RE/#board-pinout>

[7] ds2781 datasheet Available at: <https://datasheets.maximintegrated.com/en/ds/DS2781.pdf>

[8] Ref 10: martyncurrey website <http://www.martyncurrey.com/hm-10-bluetooth-4ble-modules/>

[9] Podd,F (2018-2019). ESP Technical Handbook: University of Manchester. 72 - 73