**ENGG1500 Assessment 2 Title Page**

**This document must be typed. Do not print it then fill in by hand!**

Student number: C3349607

Discipline: Electrical and Electronic engineering

Workshop class: Workshop number / Day / Time

W01/06/Friday/0800

Your workshop number can be found by googling ENGG1500, getting the 2020 online timetable and finding your class

Team number: 34

How many hours did this assignment take: ~30

What mark would you honestly give yourself: 10/10

Putting a low mark here will not negatively affect your mark

**Compliance**

|  |  |
| --- | --- |
| *Staple in the top left hand corner (or bound)* |  |
| *UoN coversheet attached to hard copy after this page. Try googling UoN cover page* |  |
| *Font is 12 point Times New Roman* |  |
| *Online submission made by 3pm Monday week 5 to* ***BB*** *as a .docx with the correct naming convention* |  |
| *Online submission is identical to this report* |  |
| *Submitted as a hard copy to the workshop leader* ***before*** *5 min past the beginning of your week 5 workshop* |  |

**Do not put your name anywhere on this document.**

# Introduction

## Project Overview

The project overview must be read and understood by everyone attempting the ENGG1500 Robot task. In short, the group must construct a machine that navigates a course containing specific obstacles via the constraints outlined in the project overview.

The group has been given a kit of Arduino components to construct this machine, however, are not limited to these components and can use anything else required.

## Skills required

### Lab Induction

Before entering the workshop, all students must have completed the SEEC lab induction on Blackboard. This outlines all safety considerations when working in the EE103A workshop.

### Arduino Code

Knowledge of Arduino code (similar to C) is required for this task. The main principles of C are taught within the ENGG1003 course, however, there are some further considerations when working with Arduino that have not yet been covered in ENGG1003. For example:

1. Including and utilising 3rd party libraries and functions.
2. Creating and using custom functions.
3. For loops, switch statements and other flow control objects not yet covered in ENGG1003 may be required.

### Constructing the machine

At this stage, the group robot is mostly built. That being said, there will likely be times when components will have to be added, removed or replaced on the machine. This will require basic skills such as unscrewing components, plugging and unplugging low voltage connections, etc. This must be done in such a way as to not damage the components of the machine.

The group also keeps a table of which component connections are wired to which ports on the Arduino. When changing any of these connections the table should be updated so the alterations can be made in code.

Please note also, that as of 20 Mar only one member of the group is to handle the robot, including modifying its configuration. This member has already been designated, Nevertheless, knowledge of how the robot is constructed and connected is still useful to possess.

# Research conducted

## Coding Arduinos

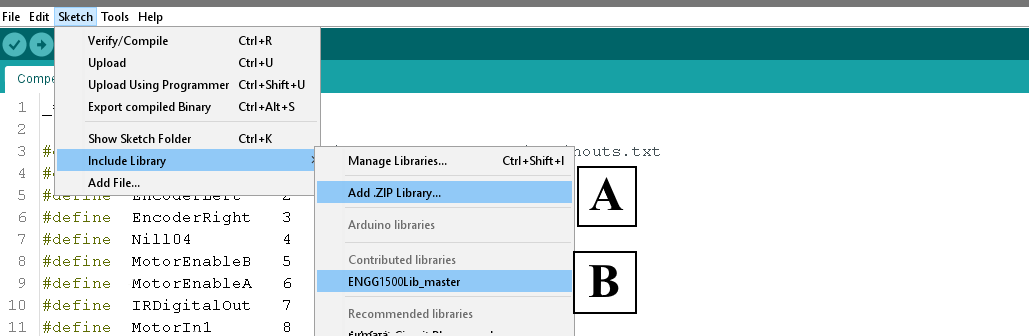
For ease of access, the group has decided to utilise Git Hub to store the relevant project files. This is to help facilitate several individuals working on different aspects of the code.

### Installing Arduino

The Arduino IDE should be downloaded from [www.arduino.cc/en/Main/Software](http://www.arduino.cc/en/Main/Software) and installed as per the included instructions for your system.

### Including Libraries

There are several C libraries built into the Arduino IDE, 3rd party libraries are also able to be installed manually. In the week 2 lab [1], a custom library for the encoder (page 8) is included as the file “ENGG1500Lib\_master.zip”. To install this file, first, download it onto your computer then, int the Arduino IDE navigate to “Sketch → Include Library → Add .ZIP Library” (Figure 1 - A) and select the downloaded .zip file.



Figure

Now to include the library simply select it from the “Sketch → Include Library” (Figure 1 - B) dropdown. see Figure 1 for further details.

## Components

The following items are currently being used on the robot.

### Arduino and Sensor Shield

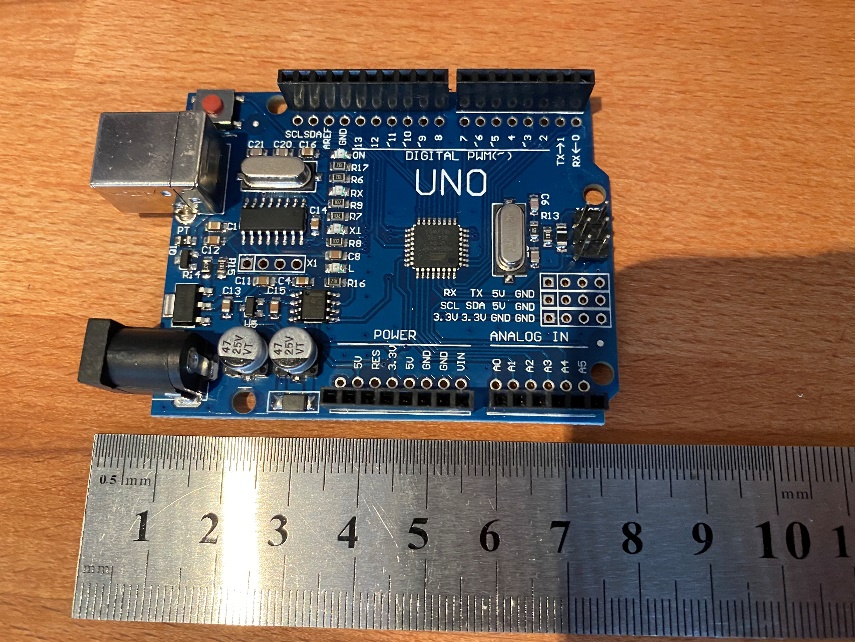


Figure – Arduino UNO

The Arduino has fourteen digital and five analogue I/O pins. These are what the components of the robot are connected to, allowing them to interface with the Arduino’s microcontroller.

The Arduino also has pins for +5 V, +3.3 V and ground connections as well as a variety of other specialised pins. As most of our components require a power, ground and variety of data connections we utilise a “sensor shield” that provides a more organised format and extra power and ground pins to better manage this.

In addition, six of the digital pins are designated as pulse width modulation (PWM) pins, designated by a (~).

#### Pulse Width Modulation

PWM refers to a method of approximating an analogue signal with a digital output. This is done by repeatedly switching the signal on and off to create the appearance of a value between fully off and fully on. This is analogous to switching a light that is white when on and black when off. If the light is switched on and off very fast, then it would appear to be a middle grey. If you kept switching the light very fast but only kept it on for a quarter of the time and off for three quarters then the light would appear a darker grey.



Figure 3

This introduces the concept of duty cycle [2]. the longer the relative time a PWM signal is kept on rather than off, the stronger the signal appears. As represented in Figure 3 the higher the duty cycle (as a percentage of time) the brighter the light. This allows us to use a digital signal that can only be either 0 or 5 V to approximate 3.3 V or any other voltage between 0 and   
5. This will be utilised later when working on the motor controller.

### Infrared Sensor

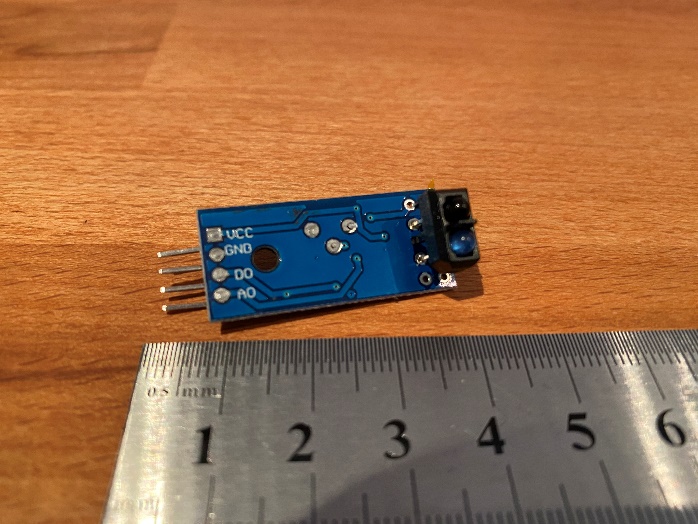


Figure – Infrared sensor

The infrared sensor measures how much infrared light reflects off a surface [3]. This can be used to detect changes in reflectivity. An infrared sensor will most likely be used to detect the strong black lines of the track.

The infrared sensor has four pins: VCC (+5 V power), GND (ground), DO (digital out), AO (analogue out). At this stage, pin AO is unused as to detect between black and white values DO is sufficient. The value of the DO pin is accessed in code though a simple digitalRead() call. The DO pin output a high value when the receiver detects more light than a certain threshold. The tolerance of the sensor is calibrated manually using a potentiometer on the board.

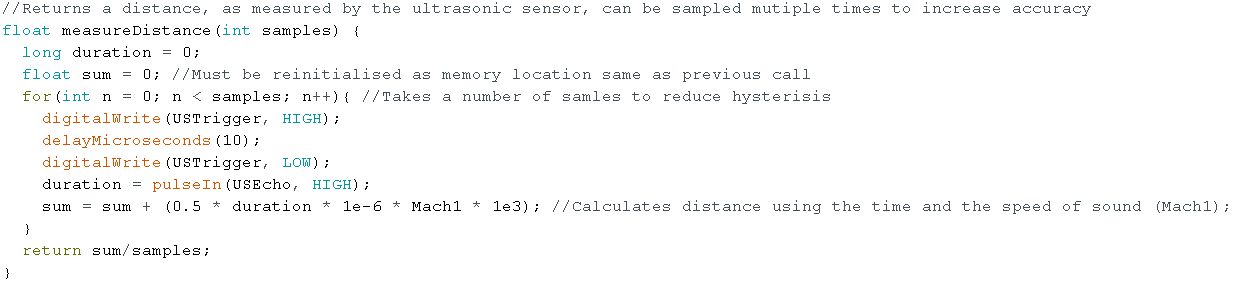
### Ultrasonic Sensor



Figure – Ultrasonic sensor

The ultrasonic sensor is used to detect distances from hard surfaces by sending out a sound wave in the ultrasonic range and recording the time it takes for the echo to return [4]. This will be useful when avoiding hard walls along the track.

The ultrasonic sensor has four pins Vcc, Gnd, Trig (Trigger) and Echo. The Trigger pin is passed a 10 μs high pulse once the sound wave bounces back the Echo pin outputs a high pulse. To use the ultrasonic sensor, a custom function measureDistance() is used (see Listing 1).



Listing

### Motor Controller and Power Distribution

#### Power Distribution Board (PDB)

To power the two DC motors that drive the wheels of the robot we require a 12 V connection and much higher current then the outputs on the Arduino can provide. The solution to this is to plug +12 V power not into the Arduino directly but into a PDB. This board splits the input voltage into several 12 V & 5 V lines which can then be used to power the Arduino and motor controller separately.

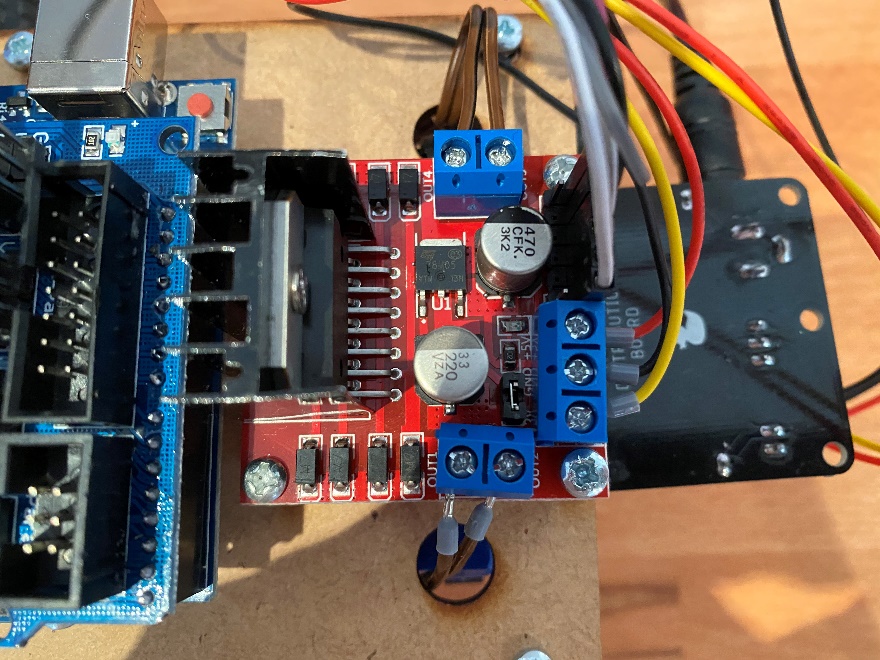


Figure - Power distribution board

Figure - Motor controller

#### Motor Controller

The motor controller takes +5 V, +12 V and ground connections, as well as six control inputs from the Arduino. The six control inputs are as follows ENA, ENB (Enable A&B), IN1, IN2, IN3, IN4 (Input 1,2,3&4).

If we wanted the first motor to spin at half strength, we would set ENA to a 50% duty cycle, IN1 to high, IN2 to low. This creates a voltage differential at motor A of ~6 V – 0 V. If we instead swapped the values of IN1&2 the motor would spin the other direction. To do the same to motor B we would use pins ENB, IN3 and IN4.

### Encoder

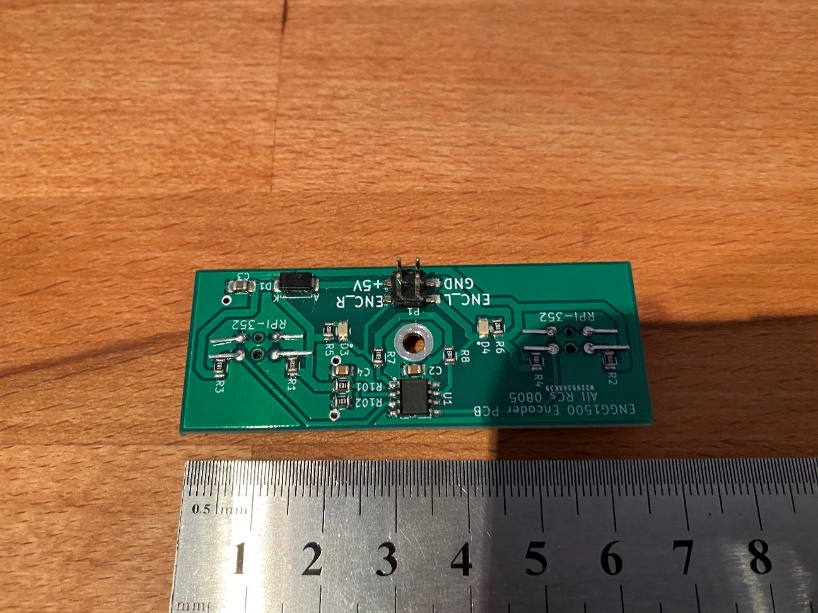


Figure 9 - Photo interrupter board

Figure 10 - Encoder wheel

The encoder board (Figure 9) has two photo-interrupters that sit over two encoder wheels (Figure 10) attached to the wheel axle. These photo-interrupts detect the changes in light as the slots of the encoder wheels rotate past them. As the wheels have 20 slots and our wheels are ~65 mm in diameter, we can use some maths to determine how many mm our robot moves per encoder slot.

Equation

From here the encoder is used to approximate how far the robot has moved by counting the number of pulses detected by the photo-interrupters. This is also useful to detect if the wheels are spinning at all.

### Motors

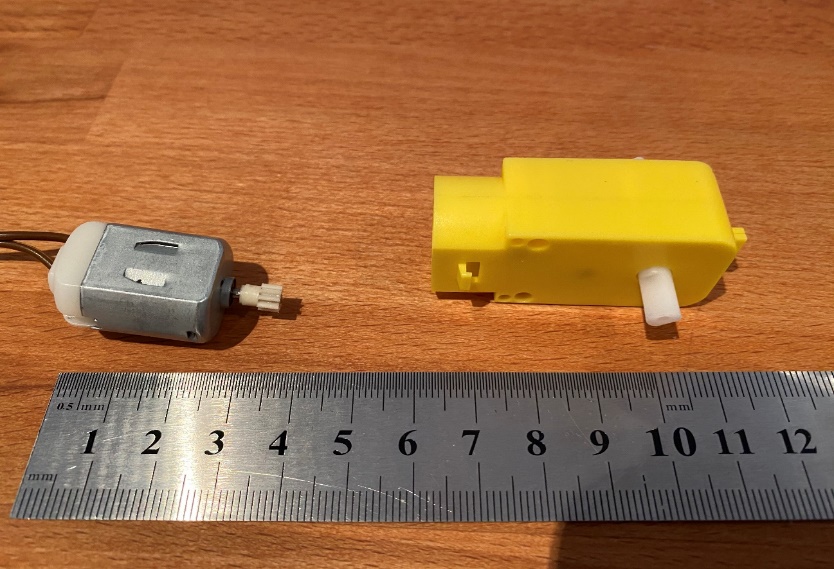


Figure - Motor and motor housing

As stated above two, 12 V DC motors are used to control the movement of the robot. We use two individual motors as this allows both more power and greater control as we can rotate the motors at different speeds and in different directions to turn the chassis. The motors are inserted into a motor housing (Figure 8) that changes the holds a gearbox and axle to attach the wheel and encoder wheel to.

# Assumptions

We assume we have to build a robot specifically.

## Track

It is assumed that the track conforms to all the conditions outlined in the project overview. The track is also short enough to permit at least one attempt on a full battery. It is possible to complete the track with the hardware provided.

## Components

The following assumptions are made about specific components; The speed of sound is consistently 343.21 m s-1 [5]. All vertical surfaces on the course will reflect 100% of the ultrasonic trigger. The ultrasonic sensor will always be positioned far enough from a wall to function correctly. The infrared sensor has enough ambient light to accurately detect black lines. The motors spin at the same rotational velocity when supplied the same voltage differential. The wheels will never slip on the track.

# Competency

## Overview

The competency was not completed in the required time as the robot could not find the black box consistently without hitting the walls of the area. The robot did manage to find the box but contacted the walls several times.

### Configuration

The robot utilised one infrared sensor and one ultrasonic sensor during the competency, with the ultrasonic sensor detecting the walls of the box and the infrared sensor detecting the black box on the ground.

### Code

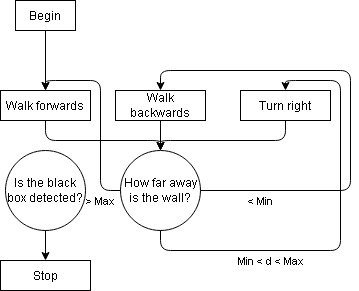


Figure - Competency UML

The code used during the competency was that of a simple state machine [6]. A state machine has a variety of functions (states) and makes decisions on which state it should be in based on a given input. These states and decisions are represented in Figure 11. The robot begins moving forward, then measures the distance reported by the ultrasonic sensors. If the distance is greater than some max value, the robot continues forwards, if it is less than some min value it moves backwards otherwise it turns. The robot stops when it detect the black box.

## A close up of a device Description automatically generatedLessons and skills learnt

The robot struggles to detect a wall when approaching from a 45° angle. This can be avoided by using a servo attached to the motor to allow the sensor to view walls from different angles.

We could also utilise the encoder to detect whether or not the wheels are moving, allowing the robot to back away when stuck.

# Conclusion

This report should provide all the information required to begin work on the ENGG1500 robot.

# References

|  |  |
| --- | --- |
| [1] | N. O'Dell, B. Schulz and D. Cuskelly, “Lab 2,” 2020. |
| [2] | Wikipedia, “ Pulse-width modulation,” Wikipedia, March 2020. [Online]. Available: https://www.wikiwand.com/en/Pulse-width\_modulation. [Accessed 23 March 2020]. |
| [3] | N. O'Dell, B. Schulz and D. Cuskelly, “Lab 1,” 2019. |
| [4] | N. O'Dell, B. Schulz and D. Cuskelly, “Lab 3,” 2020. |
| [5] | M. Bannon and F. Kaputa, “The Newton-Laplace Equation & Speed of Sound,” Thermaxx Jackets, 12 December 2014. [Online]. Available: http://www.thermaxxjackets.com/newton-laplace-equation-sound-velocity/. [Accessed 20 March 2020]. |
| [6] | M. Shead, “State Machines – Basics of Computer Science,” Mark Shead, 14 Februray 2011. [Online]. Available: https://blog.markshead.com/869/state-machines-computer-science/. [Accessed 20 March 2020]. |