# Motor Control

R. Shek

## **Introduction**

We require our rover to be able to move freely and be able to find its way through the obstacle course, to do this we used two DC motors controlled by the Arduino, using an integrated H bridge circuit.

Diagram, schematic

Description automatically generated**How does the H-Bridge work?**

The H-bridge is the piece of the circuitry which allows the Arduino to communicate with the motors [1]. The priciple of how the H-bridge operates is through the use of transistors as switches, by turning transistors in diagonals the current will flow through the motor, the motor will start to spin in a certain direction, by reversing the current we revers the motion. Once we connect the H-bridge IC we are suppliying a 3.3V as Vm and through the use of the voltage outputs from the microcontroller we can send pulse width modulated (PWM) signals to the transistors which will decide how much current will drive the motors.

Fig. 1 Circuit Diagram of H-Bridge [2]

Diagram

Description automatically generated The PWM signal is the key factor which determines the speed of the motor [3]. To change the speed we changed the voltage driving the transistors, as we know digital signals are either high or low; however by switching the signal on or off at very rapid intervals, changing the duty cycle, the voltage can be tuned from the minimum of 0V to the maximum voltage, 3.3V.

Fig. 2 Diagram of different PWM signals [4].

**Implementation in code:**

Graphical user interface, text, application

Description automatically generatedIn the beginning, we connected the Arduino board to the H-bridge, achieved by connecting 4 digital pins to the H-bridge module. The module can control the wheels by sending 2 direction signals and 2 enable signals. In the IDE we setup the pins into output mode so that the Arduino can send the four digital signals as output, then we defined some variables; a constant variable for the time, a variable to set the start time and one for the end time. To structure the code for the movement of the rover we sectioned the code and defined a function for each direction, forwards, backwards, left, and right. Each function was structured as the code in figure 4.

Fig. 3 Original function for forward movement.

First a signal is sent to the DIR pins for the left and right motors. Then, to time the motors correctly we made use of the millis() function, an inbuilt timer inside the board which counts the time the program has been running, to use the millis() function we built a while loop which runs for a set time interval, once the program enters this while loop it will set the speed of the wheels, done by using the analogWrite() function with the enable pin and an integer ranging from 0-255 for the input parameters, this function is what sends the enable pwm signal; in our testing we discovered that a value of 200 for the speed was sufficient to travel a decent distance during a 2000 millisecond time interval, when turning a value of 150 was used.

**Improvements:**

Text

Description automatically generatedOur final implementation adds improvements to the control, we decided to use two sliders to control the motion of the rover. The sliders have an output range between -255 and 255, these two values are taken in by the function as input parameters, then by using multiple if statements the direction of the wheels is decided, finally the absolute value of the input values is calculated and sent to the motors as a pwm enable signals.

Fig. 4 Final iteration of the motion function.

This implementation offers many advantages, it simplifies the controls into one function instead of having separate function for each movement. It also offers the rover greater dexterity as it will constantly update its speed and direction as the sketch will loop through the motion function infinitely.

## **References:**

1. Build electronic circuits. What is a H-bridge [Internet]. Build electronic circuits; n.d. [cited 2022 June 08]. Available from: https://www.build-electronic-circuits.com/h-bridge/.
2. EEERover technical guide. [Internet]. EEERover technical guide; n.d. [cited 2022 June 08]. Available from: https://github.com/edstott/EEERover/blob/main/doc/README.md.
3. Arduino Docs. Basics of PWM [Internet]. Arduino Docs; n.d. [cited 2022 June 08]. Available from: <https://docs.arduino.cc/learn/microcontrollers/analog-output>.
4. ^