

---

# Aberystwyth Robotics Club - Elegoo Robot

## Driving the Robot

---

### Introducing Arduino Programming

The instructions for our robot are always written inside Functions. Some values are defined above the functions. There are two essential functions for every program `void setup()` and `void loop()`. When we create a new program it will look like this:

```
void setup() {  
    // put your setup code here, to run once:  
  
}  
  
void loop() {  
    // put your main code here, to run repeatedly:  
  
}
```

All the instructions in `void setup()` are run when we first turn on the robot. This is good for any initialisation we need to do. After that, all the instructions in `void loop()` are run in a continuous loop until we turn off the robot. We can define some of our own functions to help make the code more readable. Text that comes after `//` is comments for us that will be ignored by the robot. Use these to remind yourself what the code is doing.

### Driving the Robot

The Elegoo is a 4-wheel robot, but the wheels are controlled in pairs. The motors are connected to numbered pins on the Arduino, so to start with, we need to define which pins are used.

For each pair of wheels we have 3 pins. Digital signals `HIGH` or `LOW` are sent to the pins to perform actions. Let's look at the left wheels.

Pin 5 enables the left motors. If this is set to `LOW` (off) the motors are off. If it is set to `HIGH` (on) then we can drive the motors. The driving is then done using pins 7 and 8. For the wheels to move, one pin needs to be set to `HIGH` and the other to `LOW`. If they are both the same the wheels will not move.

1. First define the pins above `void setup()`, and enable the left motors

```
#define LeftEnable 5  
#define LeftForward 7  
#define LeftReverse 8
```

2. Next initialise the pins to expect output inside `void setup()`

```
pinMode(LeftEnable, OUTPUT);  
pinMode(LeftForward, OUTPUT);  
pinMode(LeftReverse, OUTPUT);  
  
digitalWrite(LeftEnable, HIGH); //Enable left motor
```

3. Below the loop function, we're going to create three functions to turn the left wheels forward, reverse and stop. The first function is given as an example, replace the `?????`s in the second and third to match the function names.



```

void setLeftForward() {
    digitalWrite(LeftForward, HIGH);
    digitalWrite(LeftReverse, LOW); //Left wheel turning forwards
}

void setLeftReverse() {
    digitalWrite(LeftForward, ?????);
    digitalWrite(LeftReverse, ?????); //Left wheel turning
    backwards
}

void setLeftStop() {
    digitalWrite(LeftForward, ?????);
    digitalWrite(LeftReverse, ?????); //Left wheel stopped
}

```

digitalWrite means we want to send a digital signal to whatever is inside this function. LeftForward is the pin we would like the digitalWrite to act on.

HIGH is what we would like the pin to do. Digital signals work on 1s and 0s, 1 being on (HIGH) and 0 being off (LOW). In our case, 1 will be go and 0 will be stop.

4. Inside the loop function, we are going to call our new functions to drive forward, stop, drive back and stop. Between each action, we need to tell the program to pause so we can see the changes, otherwise they would happen too fast. The pause duration is defined in milliseconds (ms), and there are 1000 ms in a second.

```

setLeftForward();
delay(500); //pause 500ms
setLeftStop();
delay(500);
setLeftReverse();
delay(500);
setLeftStop();
delay(500);

```

5. Upload the program to your robot. Whilst holding the robot off the ground, turn on the battery and check the left wheels move in the correct directions.
6. Given the pins defined below, work with your partner to drive the right wheels, based on steps 2–4.

```

#define RightEnable 6
#define RightForward 11
#define RightReverse 9

```

7. Currently we are just driving one pair of wheels at a time. If we want to go straight, we need to control both pairs of wheels. Calling the functions like we did in step 4, work together to fill in the functions below.

```

void setForward() {
    ?????
}

void setReverse() {
    ?????
}

```



```
void setStop() {  
    ?????  
}
```

Hint: setting left forward and right forward without a pause between will enable the robot to drive straight.

8. Test out these new functions by calling them in the loop function. Can you get your robot to drive in a square? What other shapes can you produce? Alternatively, get your robot to dance.
9. Discuss with your partner how you could use this to navigate a maze. What sort of problems might you encounter?

