**Main**

//Function Direction Speed (PWM)

//Motor 1 D 4 D 5

//Motor 2 D 7 D 6

//Motor 1

**int directionL = 4;**  //Pin controlling Left motor direction

**int speedL = 5;**  //Pin controlling Left motor speed

//Motor 2

**int directionR = 7;** //Pin controlling Right motor direction

**int speedR = 6;**  //Pin controlling Right motor speed

**void setup() {**

//Left Motor setup

**pinMode(directionL, OUTPUT);**

**pinMode(speedL, OUTPUT);**

//Right Motor setup

**pinMode(directionR, OUTPUT);**

**pinMode(speedR, OUTPUT);**

//Turns motor pins off until needed

**digitalWrite(speedL, LOW);**

**digitalWrite(speedR, LOW);**

}

**void loop() {**

//run test() functions

**// test1();**

**// test2();**

**// test3();**

**delay(3000);**

**}**

**Helpers**

/\*\*\*

\* This creates an easy way to set the speed of both motors

\* We assign the name 's' to be our integer input speed for the function

\* Instead of digitalWrite(), we'll use analogWrite()

\* digitalWrite() can only assign pins to two states: HIGH or LOW

\* analogWrite() can give pins a range of values, from 0-255 or 0-1023

\* In our case, the range of speed spans from 0 to 255

\*/

**void setspeed(int s){**

**analogWrite(speedL, s);**  //Set the speed for the Left motor to 's’

**analogWrite(speedR, s);** //Set the speed for the Right motor to 's'

}

/\*\*\*

\* This allows us to easily control the brakes of both motors

\* Here we set the pin controlling speed to LOW, which means the pin is currently off

\* Since it receives no power, the motors won't rotate

\*/

**void brakeON(){**

// Setting the speed pin to be LOW

// In LOW, it receives 0V, and so can't move

**digitalWrite(speedL, LOW);**

**digitalWrite(speedR, LOW);**

**}**

/\*\*\*

\* This allows us to easily control the brakes of both motors

\* Here we set the pin controlling speed to HIGH

\* Which means it's able to receive power and rotate at the given speed

\*/

**void brakeOFF(){**

// Setting the speed pin to be HIGH

// In HIGH, it receives +5V, and so can move

**digitalWrite(speedL, HIGH);**

**digitalWrite(speedR, HIGH);**

}

/\*\*\*

\* This synchronizes the motors to turn in the same direction

\* The direction pin uses HIGH or LOW to represent the direction in which the motors will spin

\* In our case, LOW means forward

\*/

**void forward(){**

**brakeOFF();**

**digitalWrite(directionL, LOW);**

**digitalWrite(directionR, LOW);**

**setspeed(100);**

**}**

/\*\*\*

\* This synchronizes the motors to turn in the same direction

\* The direction pin uses HIGH or LOW to represent

\* the direction in which the motors will spin

\* In our case, HIGH means backward

\*/

**void backward(){**

**brakeOFF();**

**digitalWrite(directionL, HIGH);**

**digitalWrite(directionR, HIGH);**

**setspeed(100);**

**}**

/\*\*\*

\* This tells the motors to turn in opposite directions

\* This rotates the robot

\* In our case, rotating right involves

\* directionL = LOW and directionR = HIGH

\*/

**void rotateRight(){**

**brakeOFF();**

**digitalWrite(directionL, LOW);**

**digitalWrite(directionR, HIGH);**

**setspeed(100);**

**}**

/\*\*\*

\* This tells the motors to turn in opposite directions

\* This rotates the robot

\* In our case, rotating left involves

\* directionL = HIGH and directionR = LOW

\*/

**void rotateLeft(){**

**brakeOFF();**

**digitalWrite(directionL, HIGH);**

**digitalWrite(directionR, LOW);**

**setspeed(100);**

**}**

**Tests**

**void test1(){**

// Setting the speed pin to be HIGH

// In HIGH, the speed pin receives +5V, and is able can move

**digitalWrite(speedL, HIGH);**

**digitalWrite(speedR, HIGH);**

//Synchronizes rotation direction of both motors

//Rotating in the same direction drives in that direction

//Rotating in opposite directions makes it rotate

**digitalWrite(directionL, LOW);**

**digitalWrite(directionR, LOW);**

**analogWrite(speedL, 100);**  //Set the speed for the Left motor to 's'

**analogWrite(speedR, 100);** //Set the speed for the Right motor to 's'

//Freezes the states of pins for the amount of time

//Using delay here allows the robot to drive for 2.5 seconds before exiting the loop

**delay(2500);**

//Setting the speed pin to be LOW. In LOW, the speed pin receives 0V

//We take advantage of that feature by using it as our way to 'brake' the motors

**digitalWrite(speedL, LOW);**

**digitalWrite(speedR, LOW);**

}

**void test2(){**

**forward();**  //Does the same exact set of instructions as test1

**delay(2500);**

}

**void test3(){**

**forward();**

**delay(2000);**

**brakeON();**

**delay(1000);**

**backward();**

**delay(2000);**

**rotateRight();**

**delay(2000);**

**rotateLeft();**

**delay(2000);**

**brakeON();**

}

**Tabs**

/\*\*\*

\* Tabs are convenient for organizing related functions

\* The main tab is the tab with the name of the sketch

\* In this case, twoMotorIntro is the main tab

\*

\* Only one thing to remember about tabs

\* The side tabs use somewhat independent

\* Variables used in the main tab must be declared inside the main tab

\* Variables from the main tab can be used in side tabs

\* BUT variables declared in side tabs exists only in that side tab

\* Variables used in the main tab

\*

\* If I declared a variable in this side tab,

\* it would be a 'local' variable because it can

\* only be used locally, in this tab

\*

\* If I declared a variable in the main tab,

\* it would be a 'global' variable because it can

\* be used by any other tabs or functions

\*/