Programming Robot Car Elegoo robot



Building and Programming robots in TINKERCAD

This week you be returning to TinkerCad Circuits to start programming the selfdriving car. This workshop aims to revisit the text programming done previously and continue to develop your text programming skills.

Please complete the bronze requirements before starting the silver, and complete the silver before starting the gold.



BRONZE Challenge: Building your robot circuit

At the challenge day, you'll be using the Elegoo Robot. The components listed here can be used to build an equivalent robot in TinkerCAD allowing code to be transferred across on the challenge day.

You can either match the exact component list or adapt to match your robot design. Note, the Elegoo is a 4-wheel robot but the motor controls are paired up so can be simulated with only 2 motors.

Important: To transfer your code to the real robot for the challenge day, you will need to make sure components are connected to the correct pins. Components and pins are specified on the next slide.

	Components
5	Arduino Uno
	9V Battery
	IR sensor
	IR remote
	H-Bridge motor driver
	2x Hobby Gear Motor
	3x PIR sensor
	Micro servo
	Ultrasonic Distance Sensor (4-pin)



BRONZE Challenge: Elegoo Robot Motors

The robot gear motors are connected via a motor driver





BRONZE Challenge: Other components

from	to
PIR 1 signal	Arduino 2
PIR 2 signal	Arduino 4
PIR 3 signal	Arduino ~10
Servo signal	Arduino ~3
Ultrasonic Trig	Arduino A5
Ultrasonic Echo	Arduino A4
IR sensor out	Arduino 12

The power pins on components can be connected to the same pins on the Arduino, like a breadboard

Power	to
* power	Arduino 5V
* ground	Arduino GND
9V battery positive	Arduino Vin
9V battery negative	Arduino GND



SILVER Challenge:

Aims:

- Write functions to drive the robot forward, left, right, backwards and stop
- Remember to setup the relevant pins as outputs
- In the main loop, set the speed for your car to drive in different directions
- Can you get the car to slowly accelerate and decelerate (slow down) to a target speed?
 - Hint: a for loop will be useful to do this for(int i=0; i<target; i++)

Notes:

- In void setup(), use pinMode(pin, OUTPUT); to specify a pin for output
- The H-Bridge for the Elegoo allows us to specify how fast the motor turns, rather than just on/off. This is done using analogWrite to the drive pin, with a value between 0-255
- To recap on programming the robot, watch the video for this week
- Example movement function:

}

```
void forward(){
    analogWrite(leftDrivePin, carSpeed);
    analogWrite(rightDrivePin, carSpeed);
    digitalWrite(leftForwardPin, HIGH);
    digitalWrite(leftReversePin, LOW);
    digitalWrite(rightReversePin, LOW);
    digitalWrite(rightForwardPin, HIGH);
    Serial.println("Forward");
```

ABERYSTWYTH

GOLD Challenge:

Aims:

- Write a function to obtain a distance from the ultrasonic sensor. The function should return a number (distance in cm)
 - Remember to set the trigger pin as output and echo pin as input
- Add the servo library
 - #include <Servo.h> //servo library
 - Servo myservo;
- In the setup function initialise the servo:
 - myservo.attach(servoPin);

Notes:

• Remember to watch the video to guide you through this task.

```
//Ultrasonic distance measurement
int Distance_test() {
  digitalWrite(Trig, LOW);
  delayMicroseconds(2);
  digitalWrite(Trig, HIGH);
  delayMicroseconds(10);
  digitalWrite(Trig, LOW);
  float distance = pulseIn(Echo, HIGH);
  distance = distance / 2 / 29;
  return (int)distance;
}
```



GOLD Challenge:

Aims:

- In the main loop, move the servo ready to the take three ultrasonic readings around the robot e.g. myservo.write(90);
- Move the servo to these three positions (remembering to allow time for the servo to move before taking the reading)
 - 0 degrees left distance
 - 90 degrees centre distance
 - 180 degrees right distance
- Call the function to get ultrasonic readings after each servo movement and store the readings in three variables, e.g.

int leftDistance = getDistance();

Notes:

- To move the servo you can use the function servo.write(angle);
- You can create a new variable called and assign it to readUltrasonic, e.g. int leftDistance = readUltrasonic();
- Remember to watch the video to recap how do do this task.



GOLD Challenge:

Aims:

- Using a set of if-statements, identify which direction has the 'shortest' distance and which has the longest. Print the answers to the serial monitor
 - Remember to use Serial.begin(9600); in the setup function to be able to print to the Serial monitor

Notes:

- If statements can be nested inside each other
- if-else(condition) can be used to add addition logic to if-statements
- Use the serial monitor to help solve problems by printing out what part of the code is running e.g. Serial.println("driving forward"); or Serial.println("found obstacle");



Extension Challenge:

What else can you add to this program?

 Can you implement a function to take readings from the PIR sensors and report which sensors detect the line?



Thank You

