

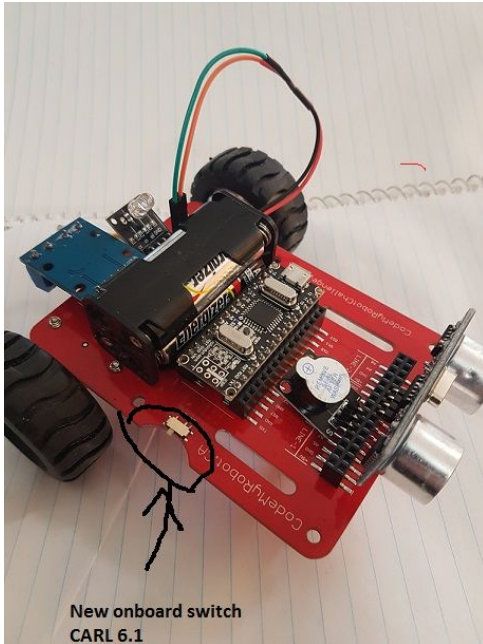
Battery storage, please remove batteries from the holder when robot not in use

See above for battery connection terminals. Small white switch is at the side of the Robot

Copyright: There is none, take anything from our manual, website, parts list, make your own materials, even. It is all open source.

CARL 6.1 Living Manual on Google

Open source manual, cut and paste what you need, share edits with us



GOAL:

1. To support teachers of youth by giving them a small coding robot
2. This starts the conversation, and the problem solving steps needed to code the robot
3. We provide some beginning code to ensure initial success
4. Students get the opportunity to learn at their own pace
5. Teachers can upload video clips to the students robot dancing, see rules on website

This manual is open for everyone to write, short helpful ideas, join us with suggestions and solutions, share your best practices, email us to join codemyrobot@mail.com



CodeMyRobot made possible by our sponsors.

Current version of this live document Version 1.2 Nov 10 2018

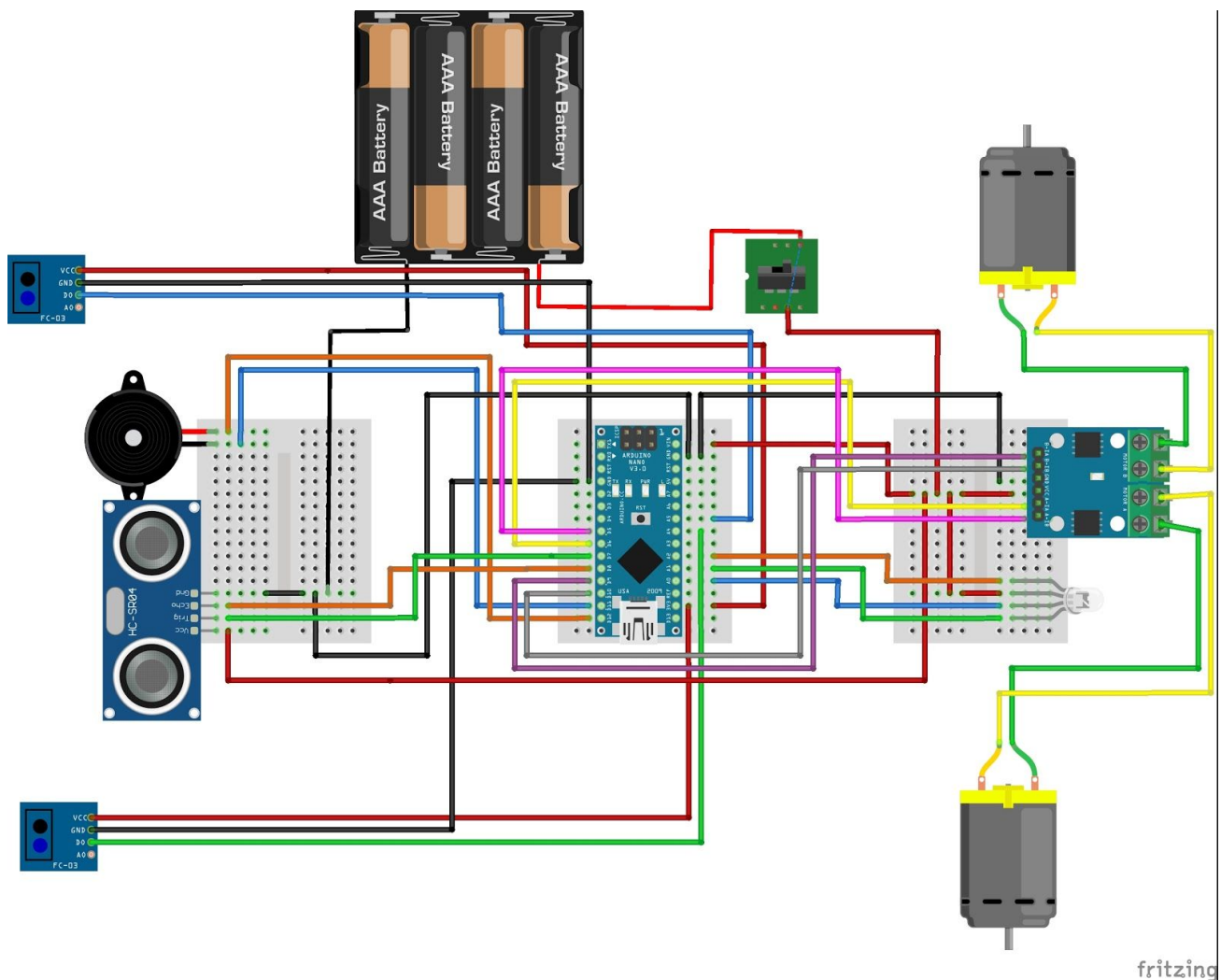
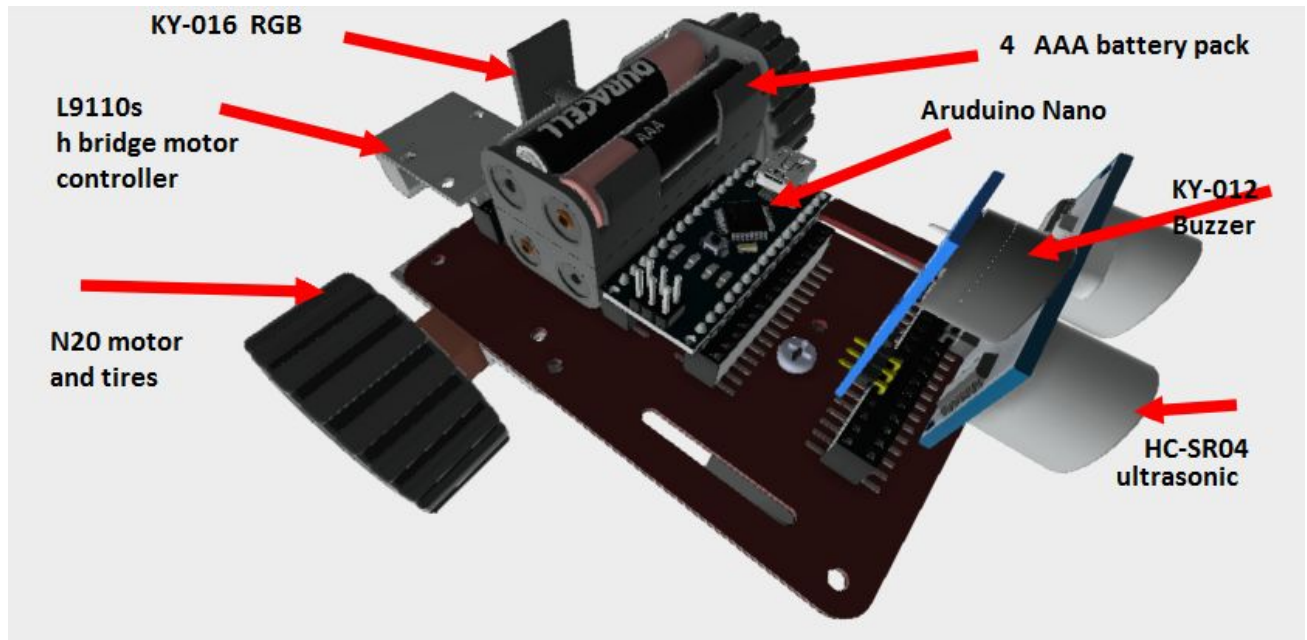
Where do I start?	take it slow, each idea built on the top of the last idea, and in no time, all will make sense. Trust me I went through this.
IT is about	making, and empowering, and collaborating.
IT is not about	answering all the questions and being an expert . We have moved from a sage on a stage, to a guide on the side, or marauder in the middle if you will..
And most important	it is all about problem solving and finding ways to make things work while learning how to slow down, analysis, compare, contrast, diagnose, test and solve.
	it is all about problem solving and finding ways to make things work while learning how to slow down, analysis, compare, contrast, diagnose, test and solve.
Other cool things	1. Make posters on how to use the robot 2. Write out the code 3. Create a robot club 4. Draw the robot in tinkercad.com 5. Make cool video clips on how to use the robot 6. Research the sensors 7 Research careers in computational thinking.
it is not about dancing the robot and you're all done	IT's about learning how you control the robot with code. It's about being inspired, creating problem solving skills, while being kind, helpful, and collaborating with other.
Safety	Take batteries out of holder when robot not in use
How to attach power	Red wire goes into the + red pin header and the black - wire goes into the black pin header, switch is at the side
	Ok lets get on to some ideas

Here are some things to try:

Learning about the parts	Print out the picture of the robot and the name of the parts: ask the student to do a slideshow with one part on each slide 1. Arduino Nano 2. RGB LED, 3. HBridge motor controller, Ultra Sonic and Buzzer
Learn how to 3D model	SketchUP is a free modeling software and students can download it and draw the robots, we can send you a short sketchup tutorial, to get you started.
Teachers can choose the degree of engagement	<ol style="list-style-type: none"> 1. High problem solving, just handout the box and ask the student to provide video of clock and maze dance 2. Advanced PS, provide computer and box 3. Medium PS, Illustrate how code works, and arduino IDE 4. Starter PS, provide and overview of all the steps 5. Instructional PS, work with the student, guiding learning
Train the trainer	Have each successful student, teach or train the next student, a lesson in collaboration
CARL 6.1	<p>You can code the robot to dance and do the clock and maze challenge</p> <p>From there you can add the RGB LED lights and color code those</p> <p>From there you can add the ultrasonic sensor and the robot will avoid hitting the wall.</p> <p>We hope to have other challenges with the ultrasonic sensor in 2019</p>
What is next	<p>You will find the CARL 6.1 allows you to snap out the Nano and the battery to be used on other projects</p> <p>Research these keywords for possible next steps</p> <p>Arduino.cc - OSEPP - VEX - First robotics - Hackaday.com - instructables.com - thingiverse.com - make.com</p>

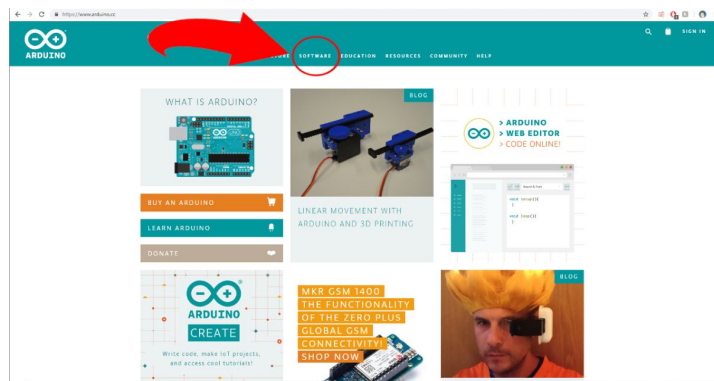
Index

Battery terminal	1
Where to start	2
Things to try	3
Index	4
The robot	5
Arduino download	6
Arduino Setup settings	10
Loading Code	12
Code settings	16
Share Best Practices	10
Challenge and prizes	11

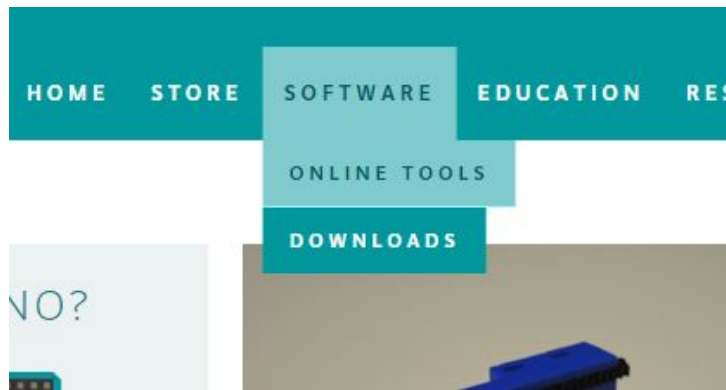


How to Download Arduino

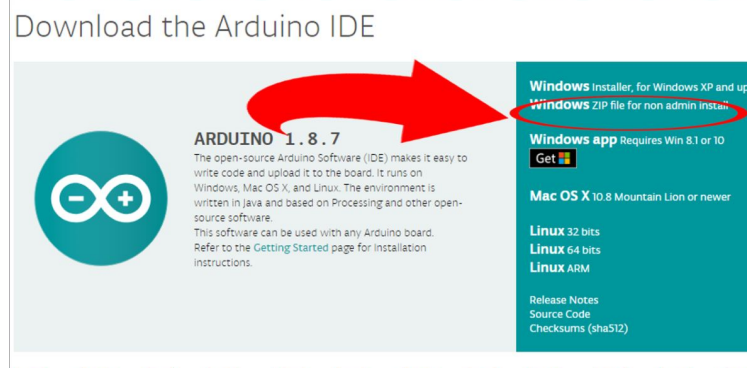
1. Open up the arduino website (<https://arduino.cc>) and hover over the software tab.



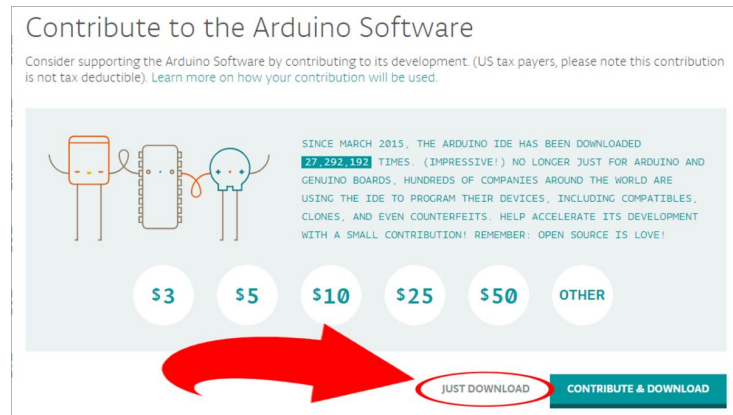
2. Click on Downloads.



3. Download the windows ZIP file.



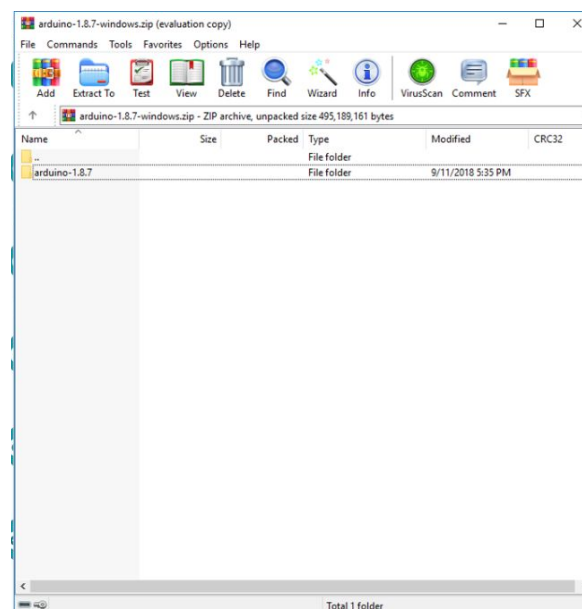
4. Click on Just Download.



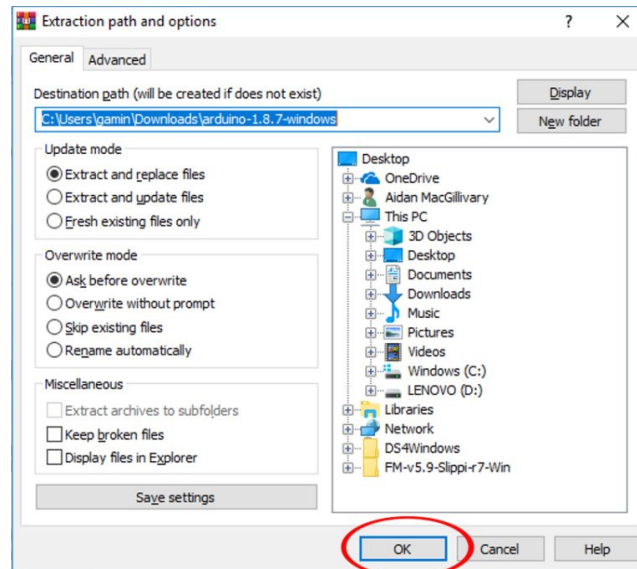
5. Open the download in the bottom left once it is done.



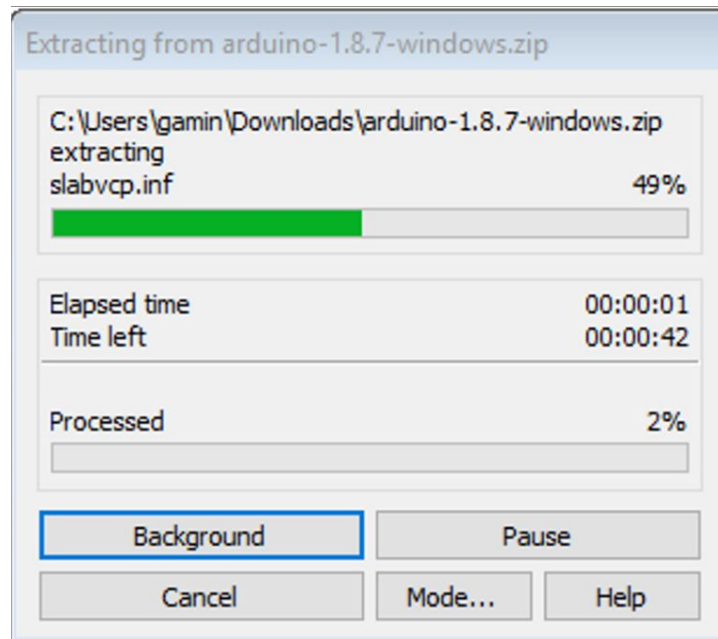
6. Right click the arduino folder, then click extract to specified folder.



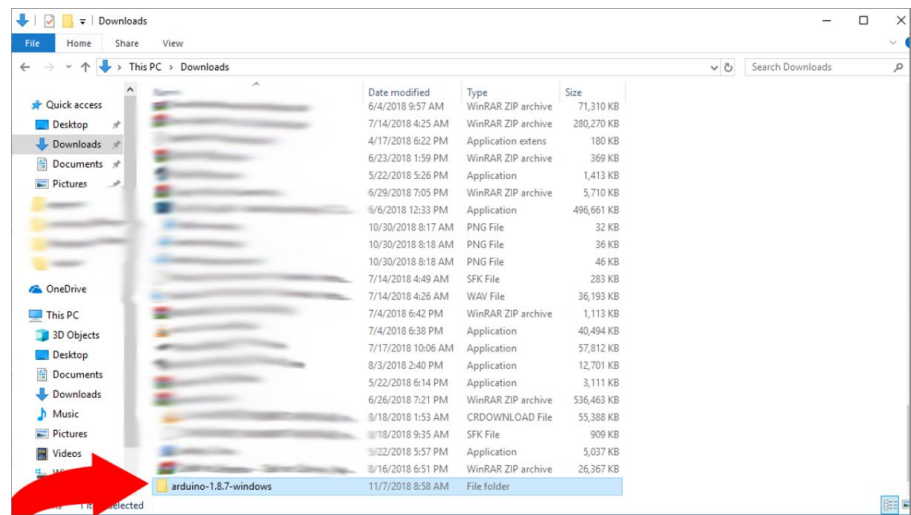
7. Choose the directory then click 'OK.'



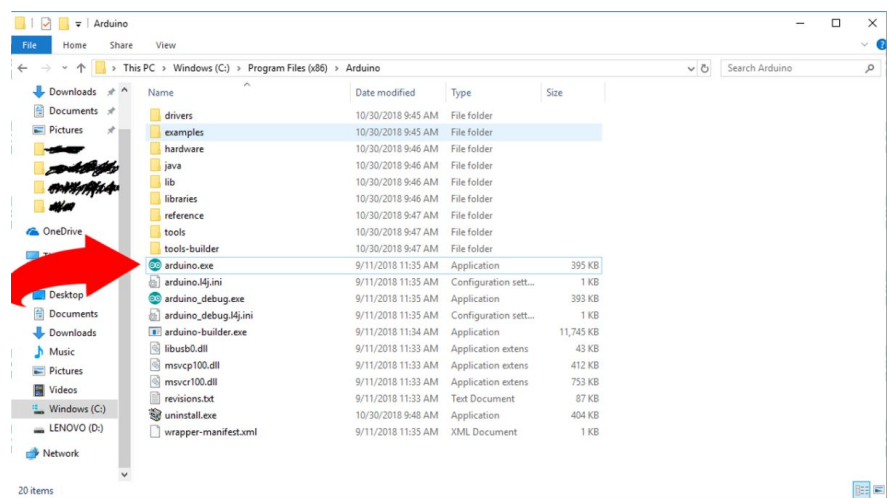
8. Wait for it to extract.



9. Open the folder it extracted to.

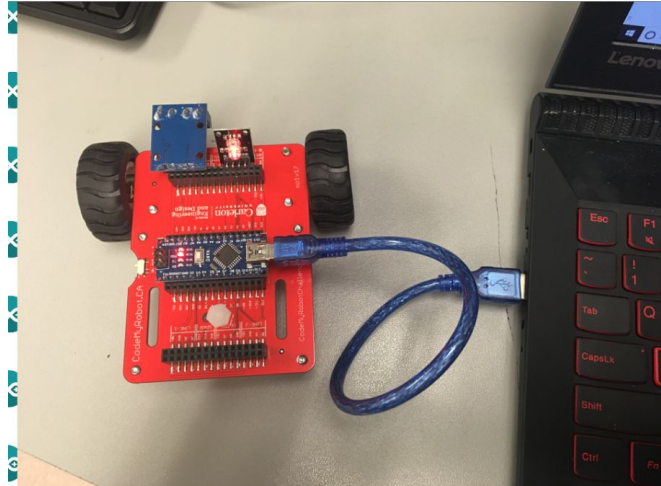


10. Open arduino.exe

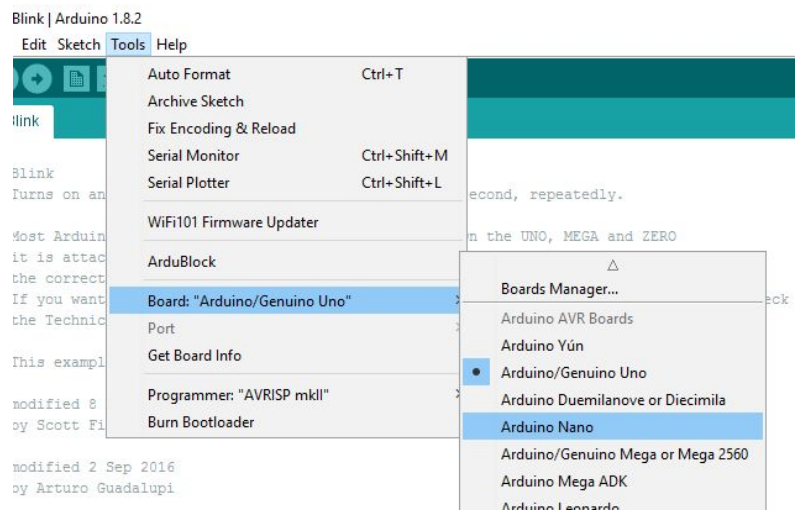


How to Change Settings for Robot

1. 4 Plug in the robot using a USB A to B adapter. You may have to turn it over as the wheels may turn as power is now plugged in.



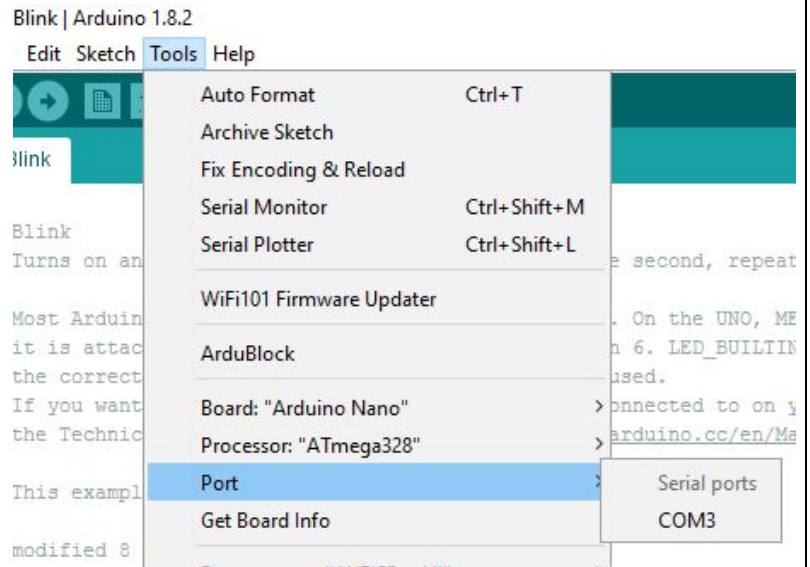
2. Click on Tools>Board:>Arduino Nano



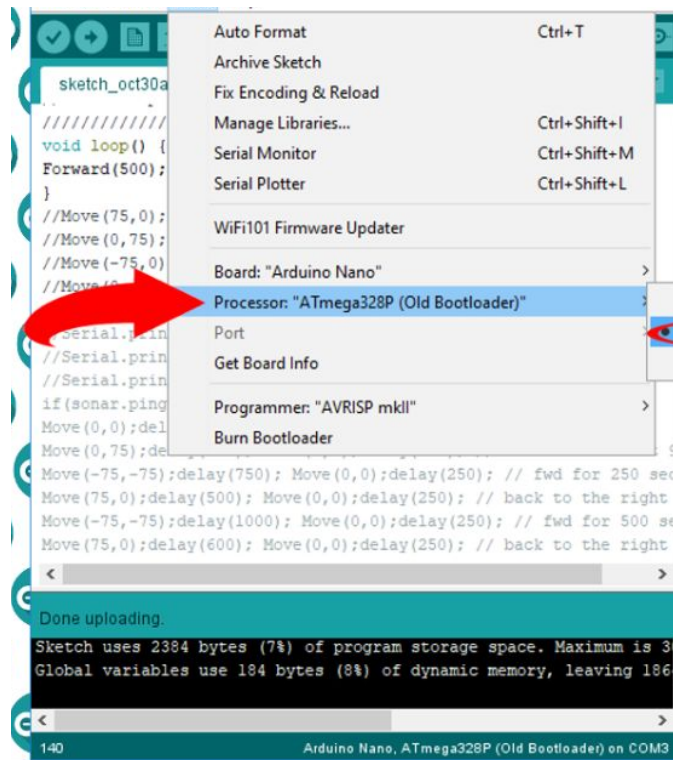
3. Select the port you've plugged the robot into.

This example is Com 3 but it can be Com 5 or Com 7

You can tell by unplugging the cord and check tools, whatever Com port that is missing is the correct port.



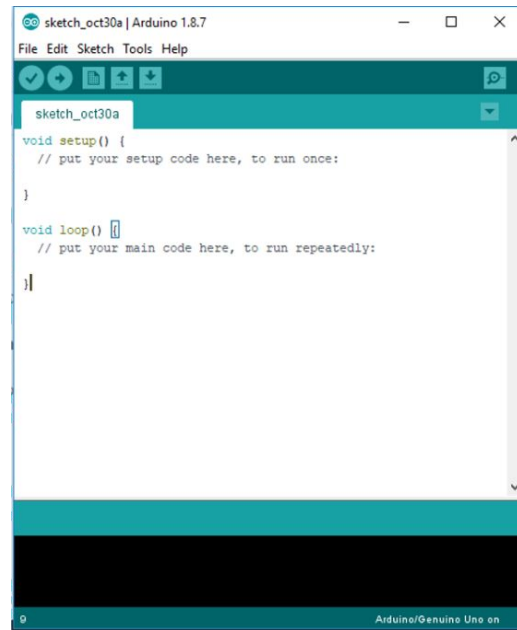
4. Select ATmega328 (Old Bootloader) for the Processor



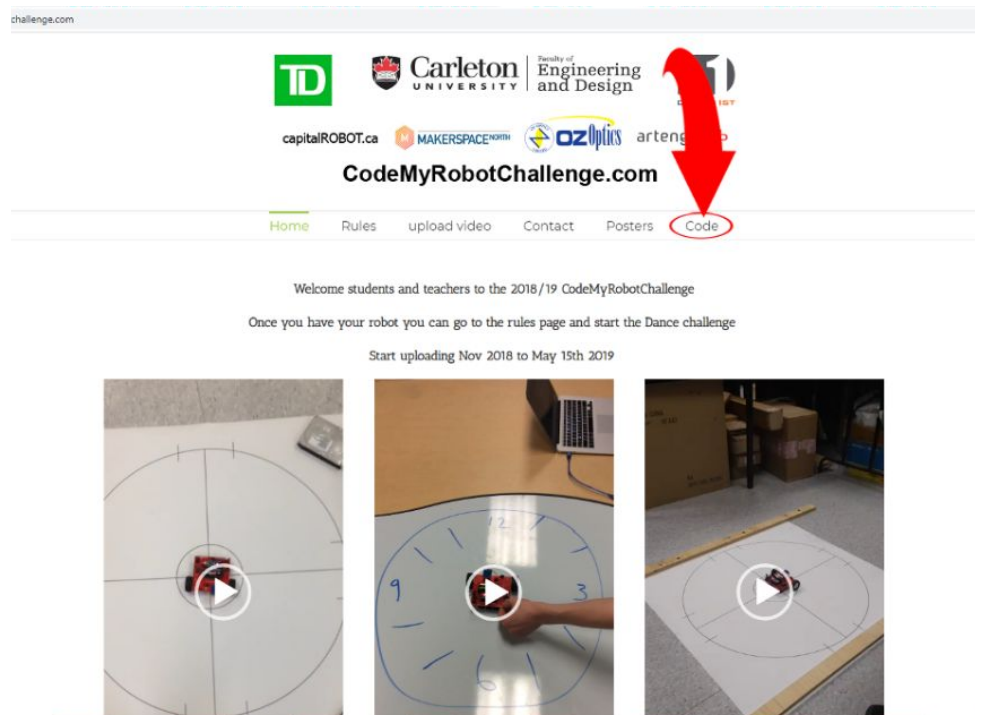
That is it You now have the settings to use the Arduino IDE

How to Get the Code for the Robot

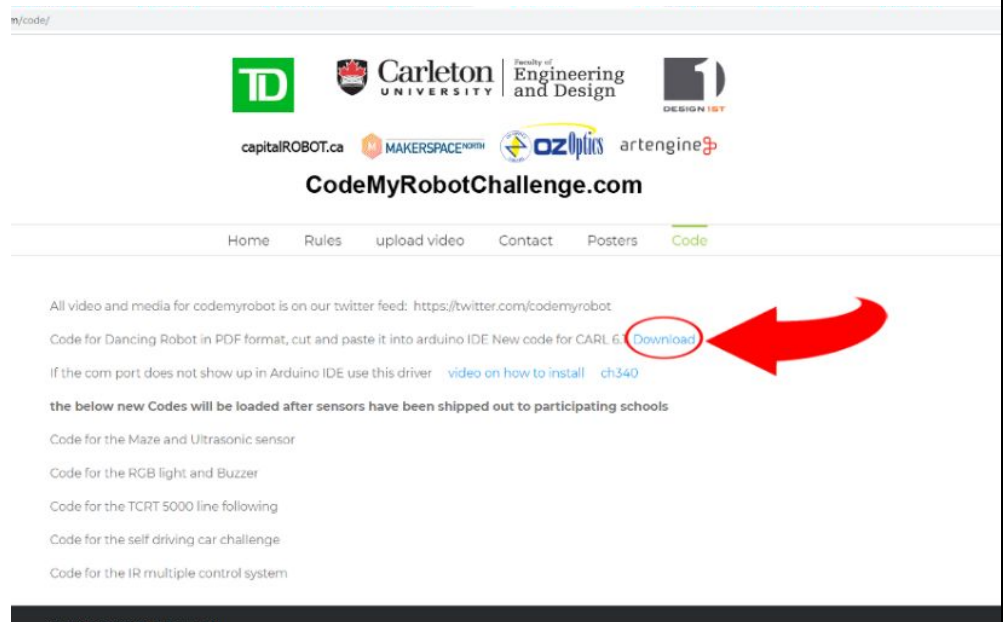
1. Make sure you already have Arduino open.



2. Go to <https://codemyrobotchallenge.com> and click on Code.



3. Click on Download.



my/code/

TD Carleton UNIVERSITY Faculty of Engineering and Design DESIGN 101

capitalROBOT.ca MAKERSPACE NORTH OZ Optics artengine

CodeMyRobotChallenge.com

Home Rules upload video Contact Posters **Code**

All video and media for codemyrobot is on our twitter feed: <https://twitter.com/codemyrobot>

Code for Dancing Robot in PDF format, cut and paste it into arduino IDE New code for CARL 6.1 **Download**

If the com port does not show up in Arduino IDE use this driver [video on how to install](#) [ch340](#)

the below new Codes will be loaded after sensors have been shipped out to participating schools

Code for the Maze and Ultrasonic sensor

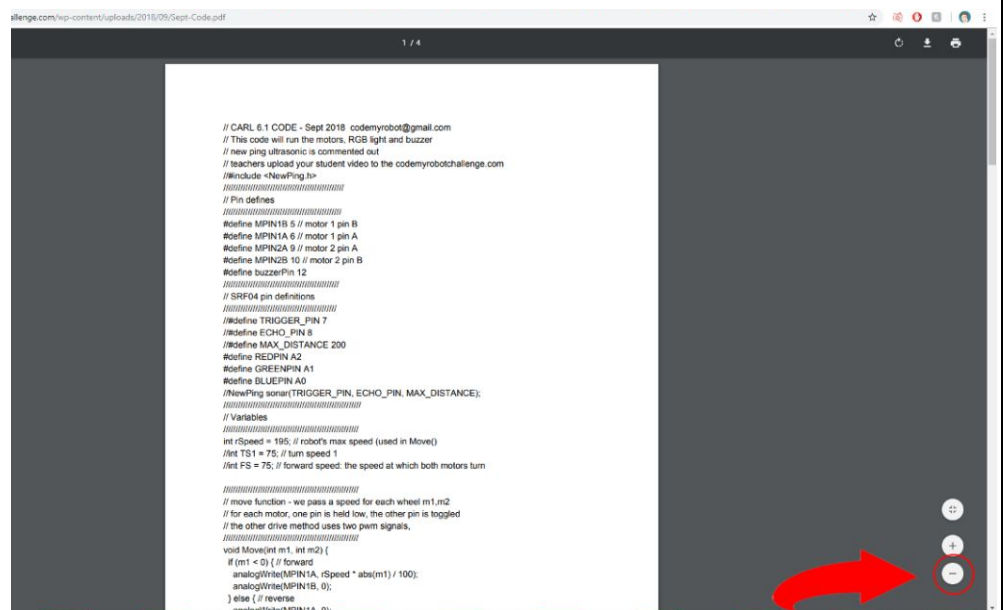
Code for the RGB light and Buzzer

Code for the TCRT 5000 line following

Code for the self driving car challenge

Code for the IR multiple control system

4. Zoom out by clicking the button in the bottom right.

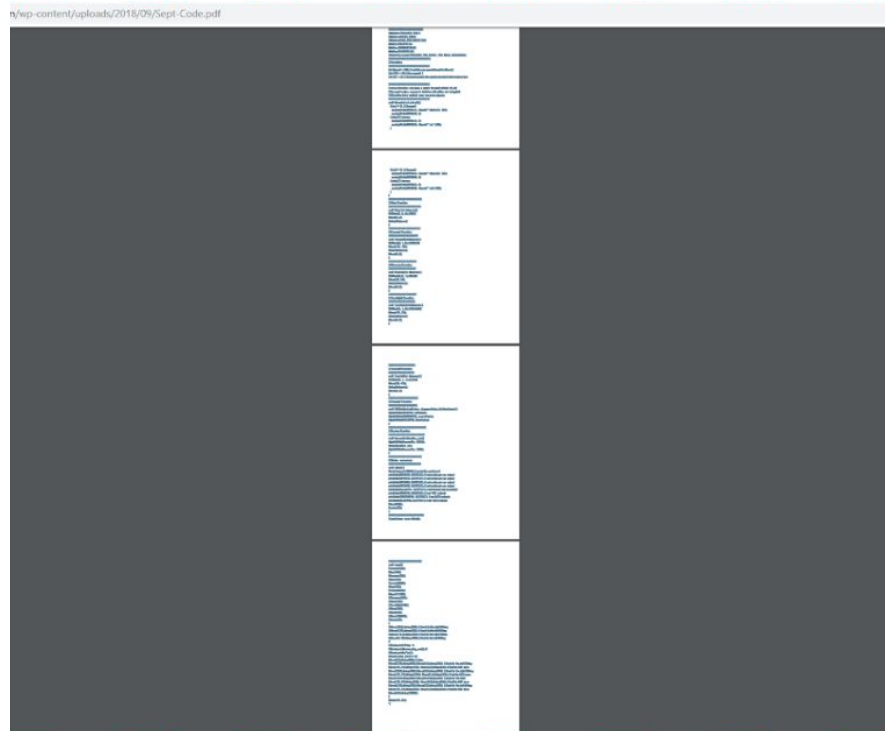


allenge.com/wp-content/uploads/2018/09/Sept-Code.pdf

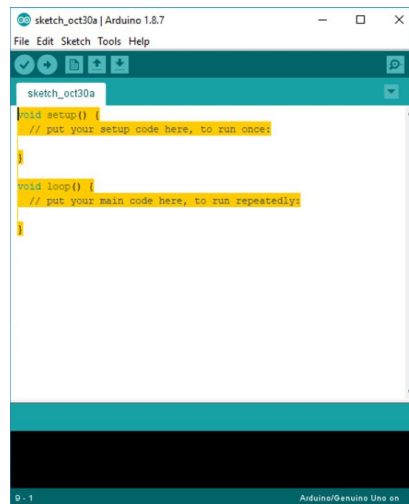
1 / 4

```
// CARL 6.1 CODE - Sept 2018 codemyrobot@gmail.com
// This code will run the motors, RGB light and buzzer
// new ping ultrasonic is commented out
// teachers upload your student video to the codemyrobotchallenge.com
#include <NewPing.h>
// Pin defines
//=====
#define MPIN1B 5 // motor 1 pin B
#define MPIN1A 6 // motor 1 pin A
#define MPIN2A 9 // motor 2 pin A
#define MPIN2B 10 // motor 2 pin B
#define buzzerPin 12
//=====
// SRF04 pin definitions
//=====
#define TRIGGER_PIN 7
#define ECHO_PIN 8
#define MAX_DISTANCE 200
#define REDPIN A2
#define GREENPIN A1
#define BLUEPIN A0
//NewPing sonar(TRIGGER_PIN, ECHO_PIN, MAX_DISTANCE);
//=====
// Variables
//=====
int fSpeed = 195; // robot's max speed (used in Move())
//int TS1 = 75; // turn speed 1
//int FS = 75; // forward speed: the speed at which both motors turn
//=====
// move function - we pass a speed for each wheel m1,m2
// for each motor, one pin is held low, the other pin is toggled
// the other drive method uses two pwm signals,
//=====
void Move(int m1, int m2) {
  if (m1 < 0) { // forward
    analogWrite(MPIN1A, fSpeed * abs(m1) / 100);
    analogWrite(MPIN1B, 0);
  } else { // reverse
    analogWrite(MPIN1A, 0);
```

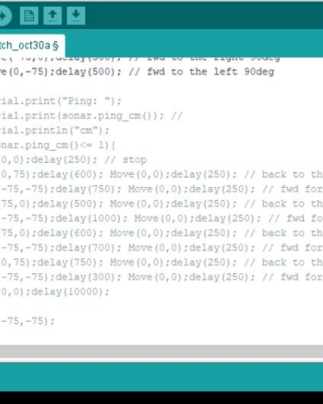
5. Select everything and copy it (Ctrl + A then Ctrl + C).



6. Go back to arduino and select all the text.



7. Paste the code (Ctrl + V).



The screenshot shows the Arduino IDE interface. The title bar indicates the sketch is named 'sketch_oct30a' and is being edited on an Arduino Uno. The menu bar includes 'File', 'Edit', 'Sketch', 'Tools', and 'Help'. The toolbar contains icons for opening files, saving, compiling, uploading, and monitoring. The code editor displays the following C++ code:

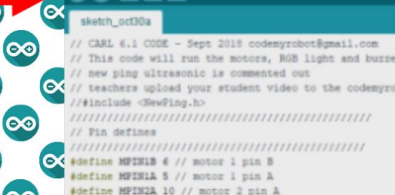
```

//Move (0,-75);delay(500); // fwd to the right 90deg
//^
//Serial.print("Ping: ");
//Serial.print(sonar.ping_cm()); //
//Serial.println("cm");
if(sonar.ping_cm()!= 1){
  Move(0,0);delay(250); // stop
  Move(0,75);delay(600); Move(0,0);delay(250); // back to the left
  Move(-75,-75);delay(750); Move(0,0);delay(250); // fwd for 250 sec
  Move(75,0);delay(500); Move(0,0);delay(250); // back to the right
  Move(-75,-75);delay(1000); Move(0,0);delay(250); // fwd for 500 sec
  Move(75,0);delay(600); Move(0,0);delay(250); // back to the right
  Move(-75,-75);delay(700); Move(0,0);delay(250); // fwd for 250 sec
  Move(0,75);delay(750); Move(0,0);delay(250); // back to the left
  Move(-75,-75);delay(300); Move(0,0);delay(250); // fwd for 500 sec
  Move(0,0);delay(10000);
}
Move(-75,-75);
}

```

The code is a C++ sketch for an Arduino Uno, likely controlling a sonar sensor. It includes comments in Chinese and C++ code for pin definitions, variable declarations, and a loop that sends ping commands and prints the results. The code is currently running, as indicated by the 'Serial Monitor' window at the bottom, which shows the output 'Ping: 100'.

8. Verify the code by clicking the checkmark button in the top left.



The screenshot shows the Arduino IDE interface. At the top, the menu bar includes 'File', 'Edit Sketch Tools', and 'Help'. A large red arrow points to the 'Sketch' menu. The main editor window displays the code for 'sketch_oct30a'. The code includes comments in German and C++ for pin definitions and motor control. The status bar at the bottom indicates 'Sketch uses 2462 bytes (8%) of program storage space. Maximum is 32768 bytes.' and 'Global variables use 184 bytes (8%) of dynamic memory, leaving 1116 bytes free.'

```

File Edit Sketch Tools Help

sketch_oct30a

// CARL 4.1 CODE - Sept 2018 codemyrobot@gmail.com
// This code will run the motors, RGB light and buzzer
// new ping ultrasonic is commented out
// teachers upload your student video to the codemyrobotchallenge.
#include <NewPing.h>

////////////////////////////////////
// Pin defines
////////////////////////////////////
#define MPIN1A 4 // motor 1 pin B
#define MPIN1A 5 // motor 1 pin A
#define MPIN2A 10 // motor 2 pin A
#define MPIN2B 9 // motor 2 pin B
#define buzzerPin 12

////////////////////////////////////
// TRF04 pin definitions
////////////////////////////////////
#define TRIGGER_PIN 7
#define ECHO_PIN 8
#define MAX_DISTANCE 200

void setup() {
  pinMode(MPIN1A, OUTPUT);
  pinMode(MPIN2A, OUTPUT);
  pinMode(MPIN2B, OUTPUT);
  pinMode(buzzerPin, OUTPUT);
}

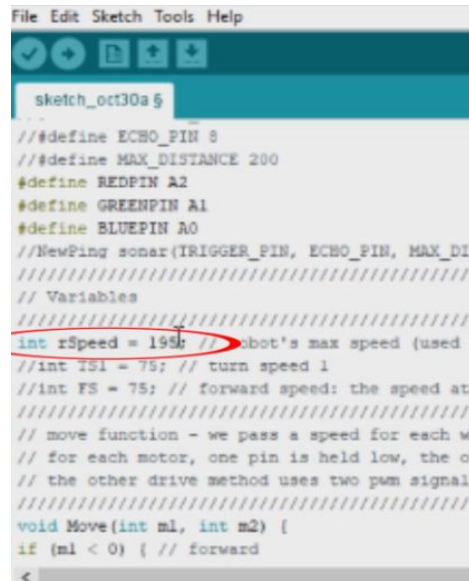
void loop() {
  // Ping
  NewPing sonar(MPIN1A, MPIN2A, MAX_DISTANCE);
  int dist = sonar.ping();
  if (dist > 0) {
    digitalWrite(buzzerPin, HIGH);
    delay(100);
    digitalWrite(buzzerPin, LOW);
    delay(100);
  }
  // Motor 1
  digitalWrite(MPIN1A, HIGH);
  digitalWrite(MPIN2A, LOW);
  delay(100);
  digitalWrite(MPIN1A, LOW);
  digitalWrite(MPIN2A, HIGH);
  delay(100);
  // Motor 2
  digitalWrite(MPIN2A, LOW);
  digitalWrite(MPIN2B, HIGH);
  delay(100);
  digitalWrite(MPIN2A, HIGH);
  digitalWrite(MPIN2B, LOW);
  delay(100);
}

Sketch uses 2462 bytes (8%) of program storage space. Maximum is 32768 bytes.
Global variables use 184 bytes (8%) of dynamic memory, leaving 1116 bytes free.
  
```

How to Change Code Settings

1. Scroll up to the line that says, "int rSpeed = 195;"

The speed can be edited by changing the number after the "=" sign.



```
File Edit Sketch Tools Help
sketch_oct30a $
// #define ECHO_PIN 8
// #define MAX_DISTANCE 200
// #define REDPIN A2
// #define GREENPIN A1
// #define BLUEPIN A0
// NewPing sonar(TRIGGER_PIN, ECHO_PIN, MAX_DISTANCE)
// Variables
// =====
int rSpeed = 195; // robot's max speed (used in Move)
// int TS1 = 75; // turn speed 1
// int FS = 75; // forward speed: the speed at which the robot moves
// move function - we pass a speed for each wheel
// for each motor, one pin is held low, the other is pulsed
// the other drive method uses two pwm signals
// =====
void Move(int m1, int m2) {
  if (m1 < 0) { // forward
```

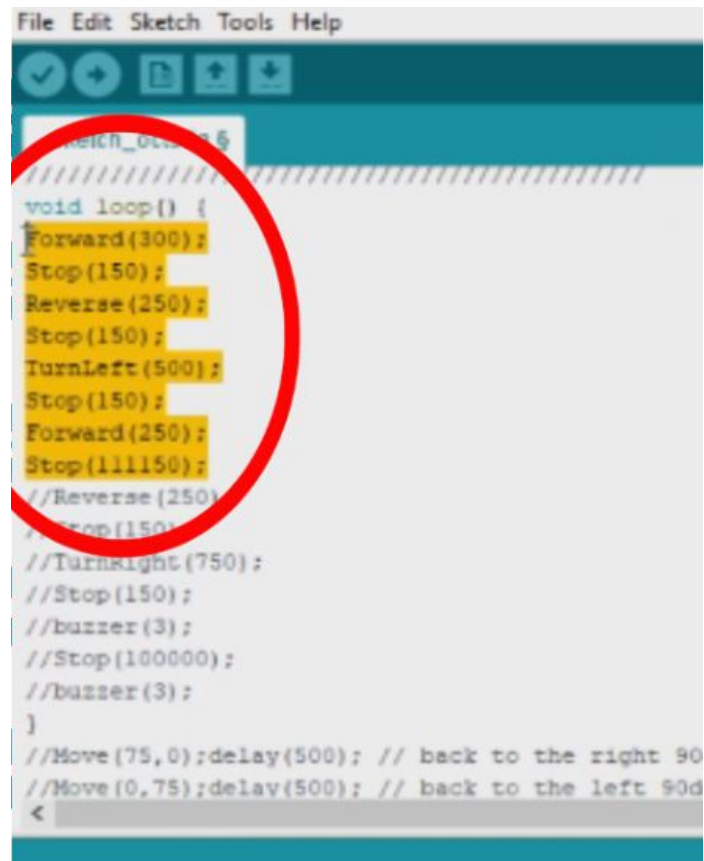
2. Scroll down to the line that says, "void loop() {"

Delete the commands there and then write in your own. This includes:

- Forward(x);
- Reverse(x);
- TurnLeft(x);
- TurnRight(x);
- Stop(x);

Where x = the number of milliseconds the command runs for.

NOTE: The lines of code with 2 slashes (//) before are just comments and will not change anything.



```
File Edit Sketch Tools Help
sketch_oct30a $
// =====
void loop() {
  Forward(300);
  Stop(150);
  Reverse(250);
  Stop(150);
  TurnLeft(500);
  Stop(150);
  Forward(250);
  Stop(111150);
  // Reverse(250);
  // Stop(150);
  // TurnRight(750);
  // Stop(150);
  // buzzer(3);
  // Stop(100000);
  // buzzer(3);
}
// Move(75,0);delay(500); // back to the right 90 degrees
// Move(0,75);delay(500); // back to the left 90 degrees
<
```