# CoderDojo Bray : Robo Sushi

The basic Arduino program to start you off with the CoderDojo Bray Robo Sushi challenges is provided: **MechRobotMinimal**.

“mechrobotminimal” is a simple program which controls a Mechatrons Ultimate robot, using the MechRobot class we are working with to wrap some of the robot functions. The “brains” of the Mechatrons robot is a standard Arduino Uno board, so we use the Arduino IDE to write code for it, and use the same techniques as we would use for a regular Arduino board.

“mechrobotminimal” doesn’t do anything very interesting: it will just drive your robot forwards, forever. It will not stop when it runs into an obstacle, it will continue to try to drive forwards. It is basically not very bright at all.

Using this “mechrobotminimal” program as a base, we have a series of challenges to go deeper into controlling the robot, learn how the sensors work, and learn some programming techniques.

This is just the first version of this doc … more challenges will be added later.

If you can complete these Challenges, then you have mastered the basics of using Mechatrons robot. Congratulations!

## Challenge #0 – Getting Started

### Level

Beginner: You need to complete this challenge to get started with working with the Mechatrons robot and your Arduino.

### Challenge

* Get the Arduino IDE installed on your computer
* Install the AFMotor, NewPing, IRRemote and MechRobot libraries  
  (USB stick, or github)
* Copy the Arduino examples too – they will be useful for experiments later
* Build the MechRobotMinimal example and download it to your robot using the USB cable
* Watch it drive along the floor!  
  **NB It will NOT stop when it bumps into something!**So follow it about and pick it up to stop it getting damaged!

### Extra Credit

* Make a copy of the MechRobotMinimal sketch
* Modify the code so that it runs forwards for only a certain time and then stops
* Modify the code so that it runs forwards for a certain time, stops, and then runs backwards for the same time, and does this over and over
  + Does it end up back where it started? Why?
* Modify the code so that it runs forwards and backwards for a certain number of times, and then stops
* Remember to save each sketch you get working separately!

### Aims

Get the Arduino IDE installed, and talking to the Mechatrons robot correctly. Verify that you can build and download code.

When you have this working, your environment is up and running and you are ready for the rest of the Robot Sushi challenges.

### Resources

Laptop, USB cable, Mechatrons robot

### Notes for Mentors

See the coderdojobray site for troubleshooting information, if the Arduino IDE will not communicate with the Arduino board

Note that we have modified our Mechatrons robots to have an on/off switch for running the motors (so that you can play with it without it jumping off the desk). If the motors don’t run, check the switch is in the “on” position

## Challenge #1 – Looking at the MechRobot Code

### Level

Beginner

### Challenge

We want to take a look at the code for the MechRobot class and get familiar with it, so that you can experiment with the code later.

* Find the folder on your computer where the Arduino IDE is installed
  + Under this folder, there is a “libraries” directory
  + In there is a MechRobot folder
  + Open up in your code editor the MechRobot.h file. This defines the “interface” to the MechRobot class:
* Find the **public:** functions. These are the functions you can call from your sketches
* Can you find the functions which the MechRobotMinimal example uses?
* Can you find the functions which control the servo motor (the sonar sensor mount)?
* Open up the MechRobot.cpp file. This is the code which actually does the work.
  + Can you find the function which sets the robot speed?
  + Can you explain how this works?

### Extra Credit

* Make a copy of the MechRobotMinimal example
* Modify your copy of the code so that the left and right wheels move at different speeds
  + What happens?
* Can you see how to run the wheels forwards and backwards?
* Modify your copy of the code so that you try the different combinations of left/right speeds and forward/backwards
  + Note down how this makes your robot move

### Aims

Learn about where the MechRobot code is, how to inspect it, and learn about the some of the functions provided by the Mechatrons library

### Resources

Laptop, Notepad++ (or equivalent for Mac/Linux machine), Arduino IDE, Mechatrons robot

### Notes for Mentors

## Challenge #2 – Moving Parts

### Level

Intermediate

### Challenge

There are various functions in the MechRobot class which control the “moving parts” of the robot: the wheels and the servo. This challenge is to get familiar with them.

* Make a copy of the MechRobotMinimal sketch
* Experiment with the functions which control the movement of the wheels:
  + setSpeed(), setLeftSpeed(), setRightSpeed()
  + forward(), backward()
  + turnLeft(), turnRight(), turnLeft(degrees), turnRight(degrees)
  + backup(), backup(speed)
  + halt()
* Can you see what all these functions do?
* How many times do you have to call turnLeft() to leave the robot facing in the same direction it started off in?
  + Does it end up in exactly the same place?
  + Why?
  + Do you get the same if you use turnRight()?

### Extra Credit

* See if you can use the rotation trim functions **getRotateTime()** and **trimRotationAngle()** to calibrate your robot so that turnLeft() does what you would expect

### Aims

Getting familiar with the functions which control the movement of your robot.

### Resources

Laptop, Arduino IDE, USB cable, Mechatrons robot

### Notes for Mentors

## Challenge #3 – Looking at the Distance Sensors

### Level

Intermediate

### Challenge

Our robot has two distance sensors on it: a sonar sensor, and an IR sensor. We want to see if they are accurate and if they return useful information

* Switch off your robot motors so that the robot will not run
* Make a copy of your example sketch
* Modify the code to use the sonarDistance() and irDistance() functions in the MechRobot class to
  + Get the disastance
  + Use **Serial.print()** and **Serial.println()** to print out the
* Observe the output on the Adruino IDE Serial Monitor
  + Are the sensors accurate?
  + Do they agree?
  + How far away from an obstacle can the sensors detect something

### Extra Credit

* Look at the “bump” sensor: can you write a program to detect when the “bump” sensor hits something
  + Is there a difference between the “left” and the “right” bump sensors?
  + Why is that?

### Aims

Learn how to read the distance and bump sensor values

### Resources

Laptop, Arduino IDE, USB cable, Mechatrons robot

### Notes for Mentors

## Challenge #4 – Using the distance sensors to control movement

### Level

Intermediate

### Challenge

Take a copy of the MechRobotMinimal sketch.

* Modify the sketch so that instead of driving forever, the robot regularly looks at the distance in front of it, and when it detects that it is approaching an obstacle
  + It stops driving forward
  + It turns in random direction
  + It drives off in that direction instead

### Extra Credit

* Add in support for the “bump” detector: sometimes the robot can bump into something which is below the sensor’s field of view. So the sensor does not see it.
  + Update your code so that the robot will change direction **either if** it sees that it is too close to something, **or** it has bumped into an obstacle

### Super Extra Credit

* Look in the MechRobot.h file to find the functions which control the servo (the sensor mount for the distance sensor)
* Modify your algorithm for what happens when the robot detects an obstacle
  + Look around with the serve
  + Take the distance in different directions
  + Pick the distance to travel based on the direction with the most “free space”
  + Drive off in that direction

### Aims

Learn about making decisions based on sensor input

### Resources

Laptop, Arduino IDE, USB cable, Mechatrons robot

### Notes for Mentors

See the coderdojobray site for troubleshooting information, if the Arduino IDE will not communicate with the Arduino board.