

Swarm Algorithms: The Roomba Experiment

Brian West

Introduction

The Roomba Experiment was started with the goal to show two robots could map out a small area so to expand the Experiment for large rooms. This was to make iRobot's Roomba more efficient at cleaning room by reference a map with all the obstacles already recorded. This report will go over how the Roomba are hooked, how to run the Roombas, and how to understand the code.

Hooking Up Roombas With Arduino

Depending on which model of Roomba you are using there are different ways to connect the Arduino.

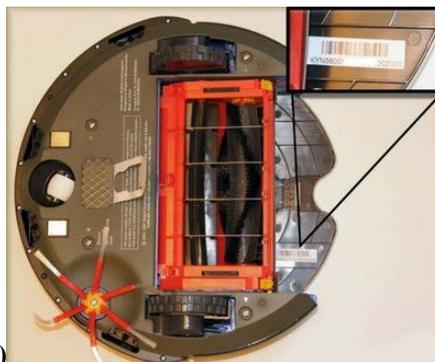
For a non-Create Roomba make sure the model is after 2005.

Checking Roomba's "birth" date

To check just look at the Serial Number either:



Under the battery



Or above the dirt bin (500 series)

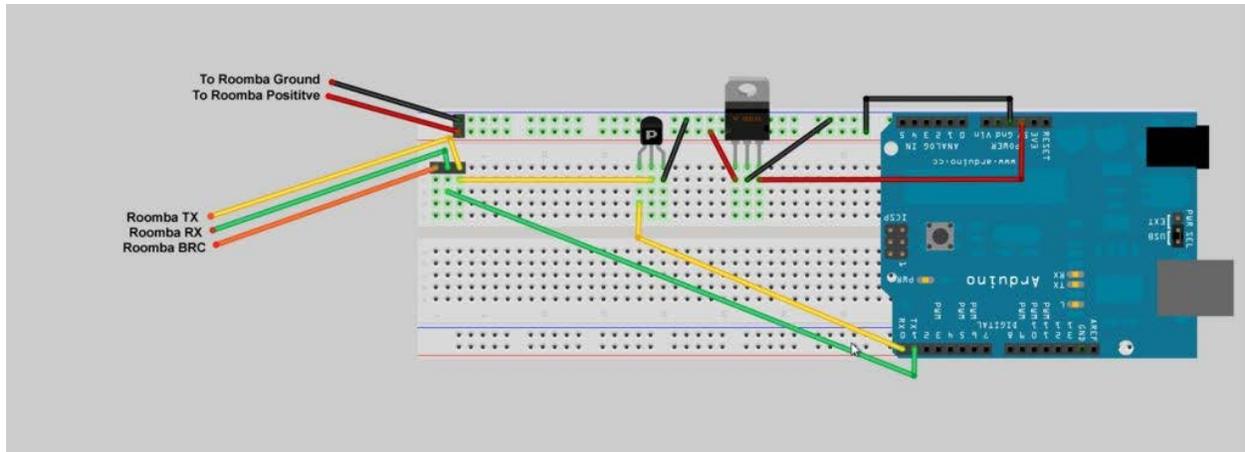


■ Year
■ Month
■ Day

How to interpret the serial number.

Hooking Up Arduino

Non-Create Roomba hook up schematic:



These use a miniDIN connection due to the way they take input through the serial port. They also do not have a 25-pin serial port like the Create.

To avoid confusion the

Red wire -> 5v socket (you can also try the Vin socket to make sure it is regulated by the Arduino)

Black wire -> GND

Green Wire -> TX

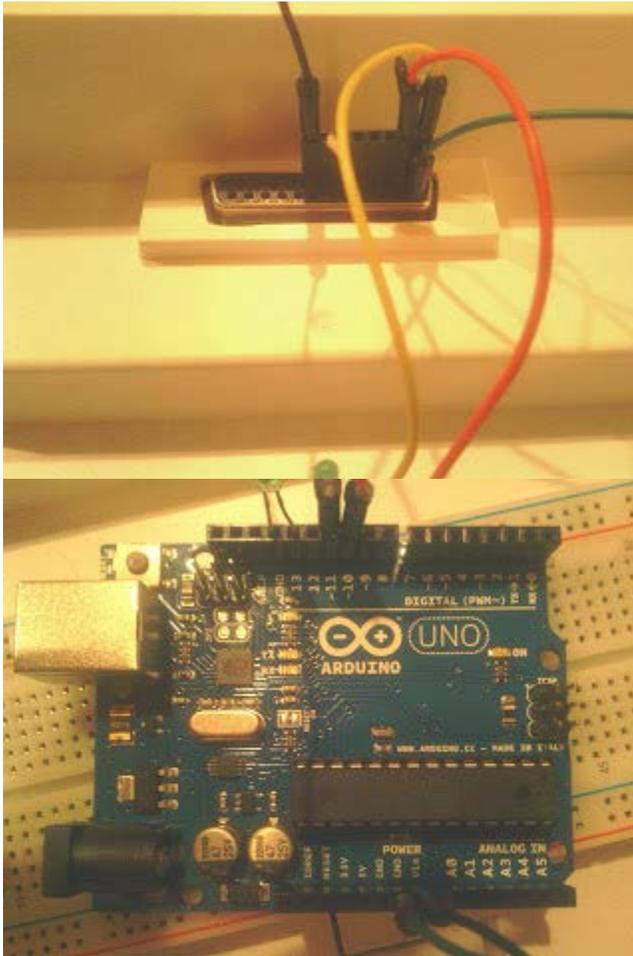
Yellow Wire -> RX

Now depending on your code the TX and RX pin can be different. (Also the wires I have already made do not match the schematic color wise, just make sure the correct goes to the correct place)

MiniDIN plug:



Create Roomba hook up Schematic:



The Create Roomba have to use the 25-pin serial port due to the miniDIN port does not give off enough power to power the Arduino nor does not accept commands from the port.

To clarify the connections:

- Arduino Vpwr --> Create pin 8 (Black Wire)
- Arduino GND --> Create pin 14 (Green Wire)
- Arduino 10 --> Create pin 2 (Red Wire)
- Arduino 11 --> Create pin 1 (Yellow Wire)

(Again also the wires I have already made do not match the schematic color wise, just make sure the correct goes to the correct place)



25-pin serial plug (aka 25-Position Male Solder D-Sub Connector)

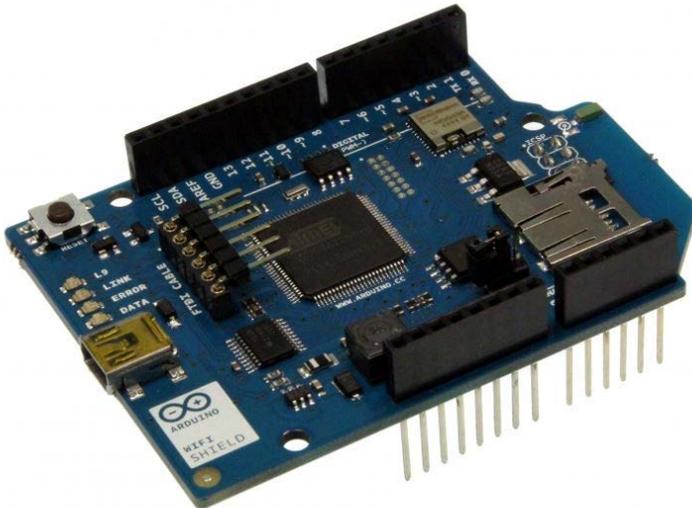
Other Attachments to Arduino:

There are shields that go on top of the Arduino:

Wireless Proto Shield (These should already be plugged into the Arduino with xBee in them) [Used for the xBee communication]



Arduino Wi-Fi Shield (integrated Antenna) [Used for Wi-Fi Communication]



Current codes that are on the Arduinos use the xBee communications and cannot currently get on Wi-Fi due to a few issues with the Wi-Fi on school. Further research needs to be done on these shields with the Roombas.

CAUTION: Not sure what ports the Wi-Fi shield uses so make sure you account for that when telling your code which pins you want for RX and TX, the two that tell the Roomba what to do.

Starting the Roomba

1. Plug Arduino into Roomba as demonstrated in the schemas above
2. Push the Roomba's Power button
3. Listen for a double beep from Roomba
 - a. (This means Arduino has taken over Roomba's functionality and is awaiting commands)
4. Press the reset button on Arduino if Roomba does not move after 5 seconds after the double beep
 - a. Sometimes the Arduino will start sending commands before the Roomba is ready thus why you have to reset the Roomba
5. Let the Roomba's work together to map out room until they are finished or get stuck.

About the Code:

Variables:

- rxpin
 - This is which pin you wish the rx commands to go through so make sure you coordinate this with your wiring when plugging into the Roomba.
- txpin
 - This is which pin you wish the tx commands to go through so make sure you coordinate this with your wiring when plugging into the Roomba.
- ledpin
 - This is which pin to power the led light (I would suggest keeping it to 13 so it's out of the way and right next to the ground which it also has to plug into to work.)
- roombaDir
 - This keeps the Roomba's direction and tells it which way to move on the map.
- sensorbytes
 - This is where the Roomba will store its bump sensor data when bumped.
- xPos & yPos
 - These are the starting point of the Roomba so make sure they are within the bounds of the map.
- maxX & maxY
 - These are used to make sure the Roomba doesn't stray too far from the map and keeps it in bounds
- mapArray
 - This is the two dimensional array which is the map for the Roomba to reference
- Roomba
 - This is the main variable that tells the Roomba what to do using the rxpin and txpin
- bumpRight
 - This is defined as the first sensor that the Roomba sends as shown by the 0x00
- bumpLeft
 - This is defined as the second sensor that the Roomba sends as shown by the 0x01

Methods:

- `setup()`
 - This method is used to prepare the pin for what they will be used for. Then it begins both Serial and Roomba connection at 57600 (Baud rate depends on the Roomba you are using but usually the default is 57600)
 - Finally it send 128(Start) and 132(Control) to tell the Roomba to now take commands from the Arduino
- `loop()`
 - This is where your main code will go to make the Roomba do everything
 - Currently the code only checks to see if it has bumped something on its left or right otherwise it will check to see if any communication is incoming. If nothing then it will drive forward.
 - If it does bump on the right then it will spin right, stop, update its own map, then transfer the map to the other Roomba and proceed.
 - If it does bump on the left then it will spin left, stop, update its own map, then transfer the map to the other Roomba and proceed.
 - If it does receive a signal from the other Roomba it checks to see if it is an update signal. If so will proceed to update its map. Otherwise proceeds.
- `isFinished()`
 - It is a method that tells when a Roomba has completed it map but does not work in the current implementation
- `changePosition(char dir)`
 - checks if dir(the direction turning) is either L (left) or R (right) and change the direction variable accordingly
- `updatePos()`
 - Updates the position of the Roomba on the map.
 - If it hits a boundary, if either x or y are at max or 0, it will back up and turn accordingly and proceed.
- `goForward()`
 - This makes the Roomba move forwards and update the map
- `goBackward()`
 - This makes the Roomba move backwards
- `halt()`
 - Stops the Roomba's motors so not to continue forward
- `ledBlink()`
 - Blinks the led light.
 - Used mostly to make sure commands are being called on time or being called at all.
- `spinLeft()`
 - Stops to motors and speeds up the opposite sides motors to turn the Roomba left
- `spinRight()`

- Stops to motors and speeds up the opposite sides motors to turn the Roomba right
- updateSensors()
 - Writes 142 to the Roomba which is the Sensor command
 - Then write 1 which retrieves the sensor packet which hold our bumper sensor data we use to detect bumps
 - From there it clears out the old sensor data and then stores the new data to be analyzed and check for bumps
- updatePoint(char clearPath)
 - Marks on the map where an obstacle has been detected with an X
- transferMap()
 - First starts a while loop to check if the transfer is done
 - Then start another loop that awaits a 'G' to be received to confirm the other Roomba is reading them
 - While it waits it continues to send 'U' over the connection to tell the other of an update to the map
 - Then follows with the x coordinate that needs to be updated
 - Next it waits for a 'G' confirmation and continues to send the y coordinate over
 - Finally once the confirmations are all made it sets done to 'T' (true) and exits out
- receiveMap()
 - creates two temporary variable to hold the new x and y coordinates
 - waits until the newX has been updated with a value
 - checks that the value is greater than -1
 - If so sets the newX equal to the received data, sends the confirmation 'G' over too proceed
 - waits until the newY has been updated with a value
 - checks that the value is greater than -1
 - If so sets the newY equal to the received data, sends the confirmation 'G' over too proceed
 - Finally updates its map with the new X for an obstacle and proceeds

Suggestions About the Code:

1. When it bumps make sure to turn the opposite way not the same way or else it will keep bumping into the obstacle
2. When both are trying to transfer data put a line in there for them to wait a random amount of time and check for a 'U' so that they break out of the infinite wait. This should allow one to transfer and one to update and then allow the other to transfer.
3. Make each Roomba update their map with where they have been and once it does hit an obstacle send the coordinates covered to the other in a batch.
 - a. This will create efficiency with the Roombas
4. Fix isFinished scenario to be more precise by checking that the whole map is marked with either obstacle marks or Roomba path marks, aka where the Roombas have marked its traveled path.
5. Add Code to check the Roomba's Battery Life and to signal where it left before its battery depletes.
6. Add Code to tell the Roomba to dock at a charging station when its battery hits a certain point.

Recommendations for Future Progression:

1. Figure out how to use the Wi-Fi shield with the Roomba so that it can be more easily checked for efficiency through a computer.
2. Build a website so that you can track the Roomba and give commands across the internet from anywhere.
3. Build a Mobile application to enable the user to view how the Roomba is doing, or to start or stop its work.

Resources

Arduino Uno to Create code and 25 pin hook up:

<https://groups.google.com/forum/#!topic/irobot-create-forum/Kgf5GFTaBhg>

How to make the miniDIN plug and Schematic for hooking up Arduino:

<http://www.instructables.com/id/Web-controlled-Twittering-Roomba/#step1>

Roomba Class API:

<http://www.airspayce.com/mikem/arduino/Roomba/classRoomba.html>

25-Position Male Solder D-Sub Connector:

<http://www.radioshack.com/product/index.jsp?productId=2103239>

Setting up the chips to program an xBee without the adapter:

<http://www.jeremyblum.com/2011/02/27/arduino-tutorial-9-wireless-communication/>

How to Program xBee chips:

<http://examples.digi.com/get-started/configuring-xbee-radios-with-x-ctu/>