

Computer programming with MATLAB
Worksheet 1: Buggy construction

Introduction

This series of four labs will take you through the process of building a computer controlled robot buggy. In essence there are two parts to the buggy. The mechanical buggy with its motors, gear boxes and wheels, and the electronic system which controls the buggy. Our buggies will be controlled by a Raspberry PI computer. This is a fully fledged computer on a single board. It has screen output, USB ports, Network ports and everything you would expect from a normal desktop computer. It is our hope that by introducing you to this computer early on in your degree, you will be able to use it in your projects later, or indeed use it in your own projects at home. First we are going to configure the Raspberry PI computer, then we are going to build the buggy.

Your robot buggy kit

Your robot buggy kit should look like figure 1.



Figure 1: The robot buggy kit

It should contain the following items:

1. Acoustic sensor, 1. Red bank of switches, 1. 5v Voltage regulator, 1. Bank of LEDs, Many Batteries, 1. SD card, 1. Raspberry PI, 1. Raspberry PI holder, 4. Buggy wheels, 4. Metal supports, 1. USB cable, 8. motor brackets, lots of wires, 1. screwdriver, 1. 3D printed camera holder, lots of resistors, 1. x4 AA battery holder, 1 x8 AA battery holder, 1. battery clip, 1. on/off switch, 1. Raspberry PI camera, lots of junction boxes.

In in 4 weeks time, you will have to disassemble your buggy and return the kit in the same state that it has been given to you in.

Raspberry Pi set-up

Step 1: First go and find the Raspberry Pi (Pi) in your kit. It should look like the image shown in figure 1. Also in your buggy kit there should be a box labelled “Premium Raspberry Pi Enclosure”. This is just a plastic box to protect the Raspberry Pi. Place the Pi in the enclosure, before you do anything else. Circuit boards can be damaged by static electricity, grease from your hands and liquids.



Figure 2: The Raspberry Pi 3.0.

Before we can turn it on, we need to install some software on to the Pi to make it work, currently there is no software on it at all, not even an operating system. Once we have installed the software onto the Pi, it will be able to interact with the various sensors and motors. Let's install the software onto the Pi. ***Note: For the 2019 class, we have already installed the software for you, so you can skip to step 6. If you want to buy your own raspberry Pi to play with at home (or use it in your 3/4th year projects, steps 2-5 will be useful to you later)... skip to step 6.***

Step 2: Download the software *Win32DiskImager* to a PC from <https://sourceforge.net/projects/win32diskimager/>

Step 3: We have bundled all the software you will need to run the Pi, in a single file called *image.iso*. You will be able to find this on Moodle, it's about 1Gb in size, so you may have to wait a moment while it downloads. This will contain the operating system of the Pi, as well as all the relevant code needed to control the motors, ultra-sound distance sensor and the camera.

Computer programming with MATLAB
Worksheet 1: Buggy construction

Step 4: Go and find the SD card and SD card reader from your buggy kit. If they are in packing, unpack them and insert the SD card into the SD card reader. Then insert the SD card reader into the PC in the computer room.

Step 5: Using *win32diskimager*, write the .iso file you downloaded from moodle to the SD card. You will have to double click on *win32diskimager* to start it. Ensure you select the image you have downloaded and the SD card using the boxes shown in figure 2. Once you press “Write”, it will take around 30 minutes to write the image to the SD card, this is because it's big file and the USB ports are slow. At this point you can begin the next section and return once the image write is complete. Don't wait for it to finish writing – you don't have time!

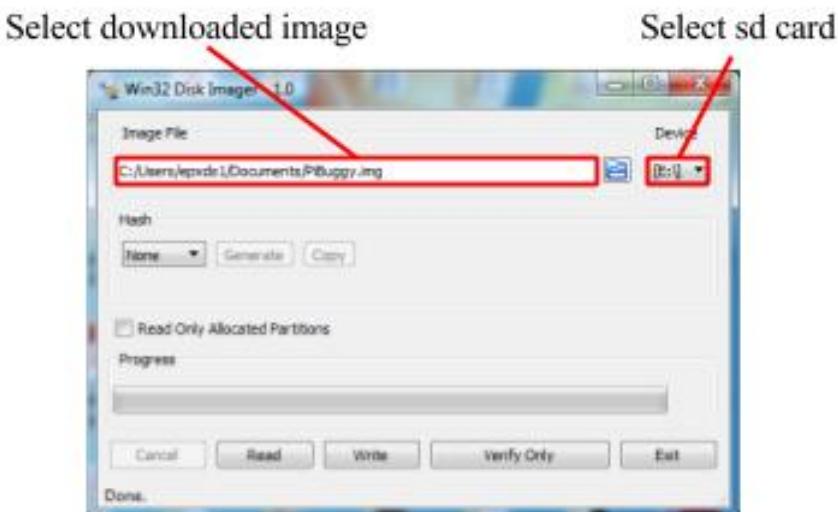


Figure 3: The disk imaging software.

Step 6: Now on the computer next to the one which is writing the image to the SD card (i.e. using another computer). Connect the PI up to the PC monitor using the HDMI cable which you will find in the buggy kit. Also, carefully, disconnect keyboard and mouse from the PC and reconnect them to your PI. Now, insert the SD card into your PI. Your PI is now set up and ready to go. All we need now is power! We will get this from the USB port of the computer. In your buggy kit, you should see a white, USB cable, with a micro USB end, this looks like a phone charging cable. Connect one end to a PC, and the other end to the PI. We are using the computer as a power supply, at the moment. Later on we will power the PI off batteries. Be very careful not to power the PI off the computer and batteries at the same time, as

Computer programming with MATLAB

Worksheet 1: Buggy construction

you could damage the computer. I also suggest, you try to power the PI off the computer as much as you can, so you don't drain your batteries.

Step 7: Once the power is connected, the PI should begin to boot up. After a minute or so, you should be able to see the desktop of the raspberry pi and be able to interact with its installed programs as shown in figure 3. If this doesn't happen, ensure the SD card is correctly inserted, and ask for help!

Step 8: Have a quick play with the PI's desktop and start a few programs. You can see that it is really a fully functioning computer. Try to find the word processor (hint: It's called Libreoffice)

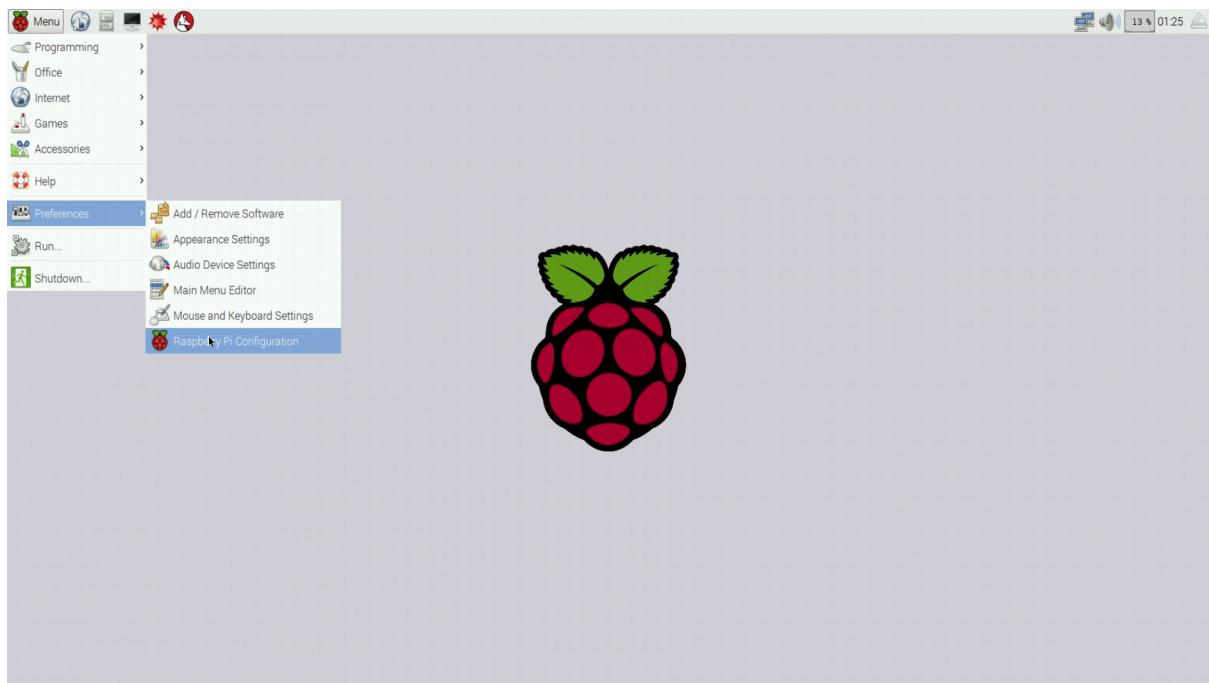


Figure 4: Raspberry pi desktop

Building the buggy

Now that we have the PI working, let's build the buggy.

Step 1: The main components of the buggy are two large plastic plates. Go and find these in the buggy kit (see figure 5). Choose one of the plates to be the *base* to your buggy, and one to be the *top*.

Computer programming with MATLAB
Worksheet 1: Buggy construction

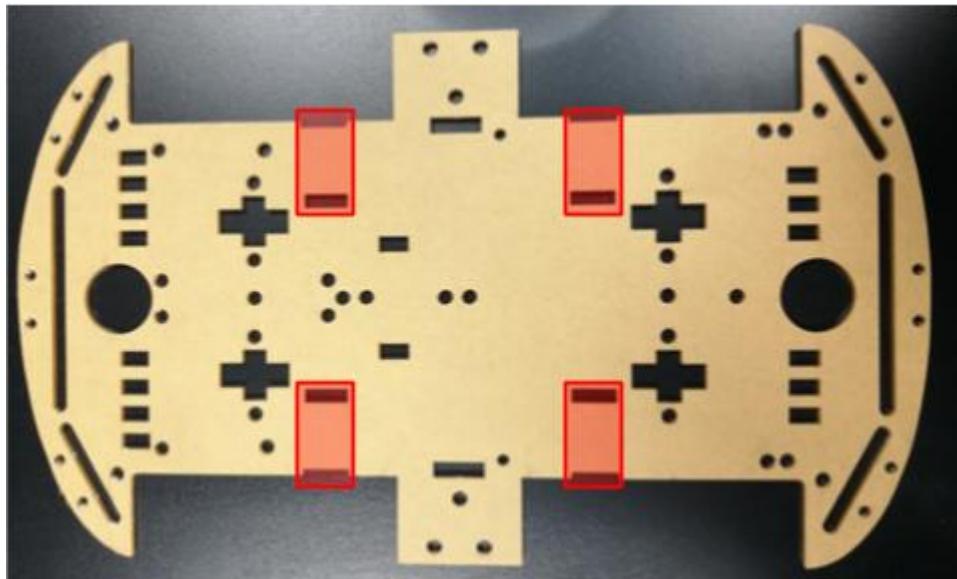


Figure 5: Plates that make up the structure of the Buggy. The holes highlighted in red are where the motors are to be mounted.

Step 2: Now go and find the gear boxes for the buggy, they are bright yellow and look what is shown in figure 6, there should be two with wires coming out of them, and two without any wires. The gear boxes with wires have motors, the gear boxes without wires don't have any motors. At each corner of the base plate, there are slots to attach each of the four gear boxes, as highlighted in red in figure 5. Using two of the t-shaped plastic pieces shown in figure 6, the mount the motor to the bottom of the base plate by placing one on each side. Also look at figure 7 to show you how the finished product should look. Note, in figure 6 the two gear boxes with wires are mounted together, and the ones with no wires are mounted together. You should do the same.

Computer programming with MATLAB
Worksheet 1: Buggy construction



Figure 6: Attachment of motor to bottom plate

Step 3: As shown in figure 6, the motor case can be attached to the base plate by threading one of the motor mounts through the board and placing the other in the groove on the outer edge of the plate. The bolts in the kit can then be threaded through the motor mounts and motor casing to hold the motor in place.

Step 4: Once the bolts are in place, the nuts can be used to securely fasten the motor casing in place on the base. Note, the nuts must go on the inside of the buggy, and the screw heads on the outside, or the bots will stick into the wheels.

Step 5: This process can be repeated with all of the motor mounts in all four corners of the base plate. Ensure that the motor casings containing the motors are both attached to the same end of the base. Once all motors are attached, thread the motor wires through the opposite side of the base plate as shown in figure 6.

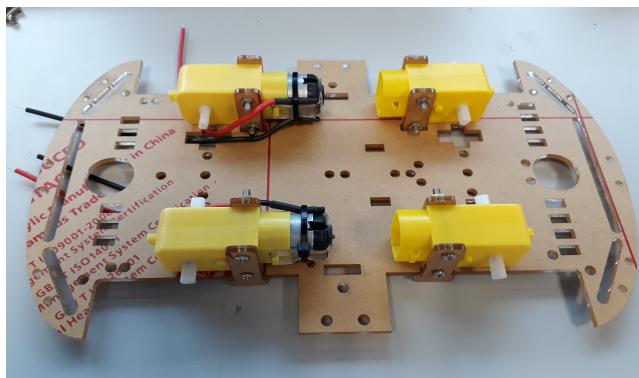


Figure 7: All four motors attached in correct orientation

Computer programming with MATLAB
Worksheet 1: Buggy construction

Step 6: All the wheels can now be attached to the motor casings and the base flipped over. This forms the base of the buggy and the starting point of the design. The wheels should just push fit on.

Step 7: In your box you will have two power battery packs, one which can take four AA cells and one which can take five AA cells, we are going to join them together, so they can produce about 12V, enough to power the buggies motors. Using a few bits of sticky pad (you will have to layer them to make them high enough to stick), join the two packs together (back to back), as shown in figure 8.

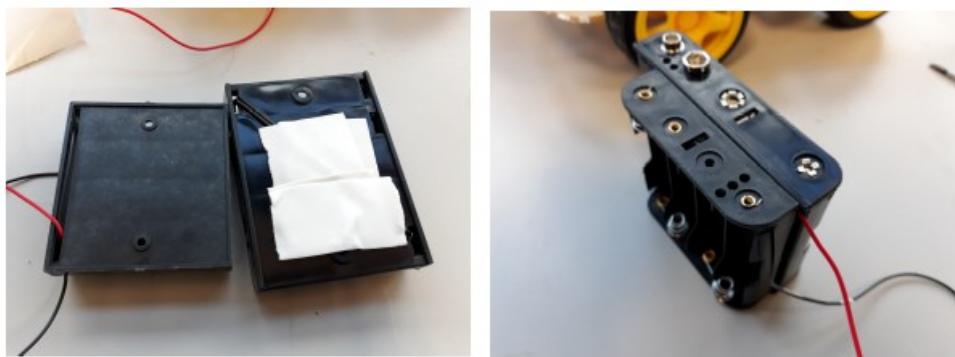


Figure 8: Building the battery pack

Step 8: You should, have a battery clip in your box (it's the sort of clip you would expect to attach to a 9V battery), attach two single junction boxes to it as shown in the figure 9. The junction boxes come in strips of 10, to get a single one, just twist or cut one off the strip. You may also have to use the wire cutters to remove a bit of wire so the wires can be easily screwed in to make a good contact.

Computer programming with MATLAB
Worksheet 1: Buggy construction

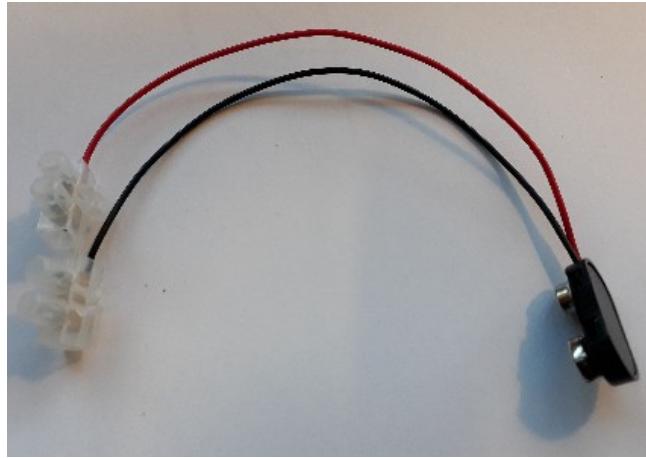


Figure 9: Adding junction boxes to the battery clip

Step 9: The black wire coming out of the x4 AA battery pack is a bit too long, cut it to around 7cm (if it has not already been cut). Once this is done, connect it to the red wire with from the battery clip, using the junction box. Then, clip the battery clip onto the x5 AA cell battery pack. This is all shown in figure 10.

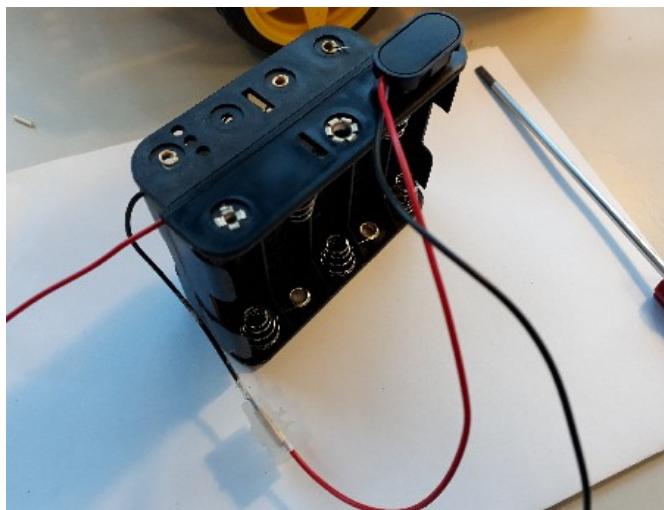


Figure 10: Connecting the battery pack to the battery clip.

Step 10: The battery pack should, now have two wires coming out of it, a black one with a junction box on it and a red one with no junction box on it. Find another single junction box and attach it to the battery pack, so both wires coming out for the battery pack have junction boxes on them. Then get a single jumper lead, with two male ends on it. Cut it in half, strip the ends and screw them into the junction boxes

Computer programming with MATLAB
Worksheet 1: Buggy construction

attached to the battery pack. The final battery pack, assembly should look as shown in figure 11.

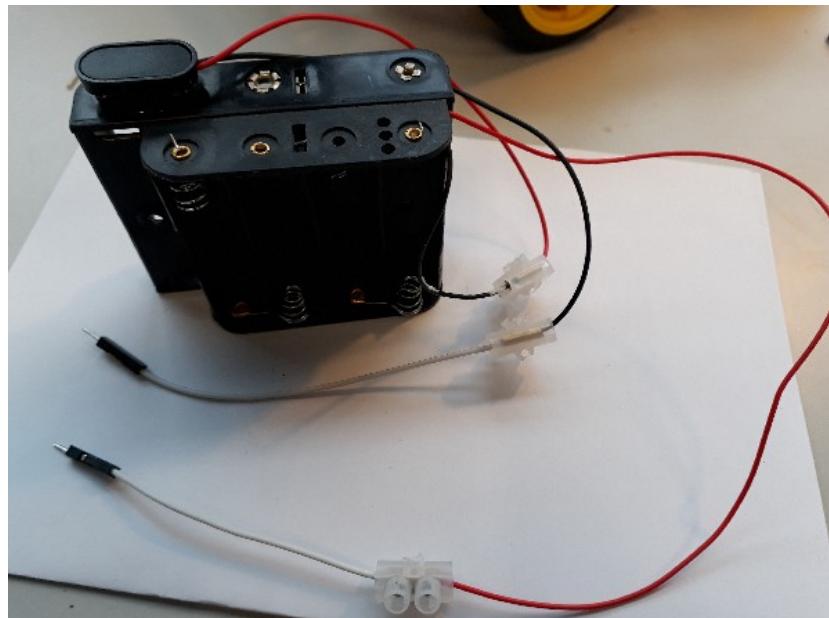


Figure 11: Final battery pack assembly.

Step 11: You should have a big pack of AA batteries in your kit, find 8 of them, and fill up the x5 battery holder with them, and x3 slots in the x4 battery holder. As shown in figure 12. **NOTE: WE ARE INTENTIONAL NOT FILLING THE x4 BATTERY HOLDER FULLY, AS WE DO NOT YET WANT TO POWER THE BUGGY. ONCE YOU HAVE WIRED UP THE BUGGY, GET A DEMONSTRATOR TO CHECK THE WIRING. THEN PUT IN THE FINAL BATTERY. :)**

Computer programming with MATLAB
Worksheet 1: Buggy construction

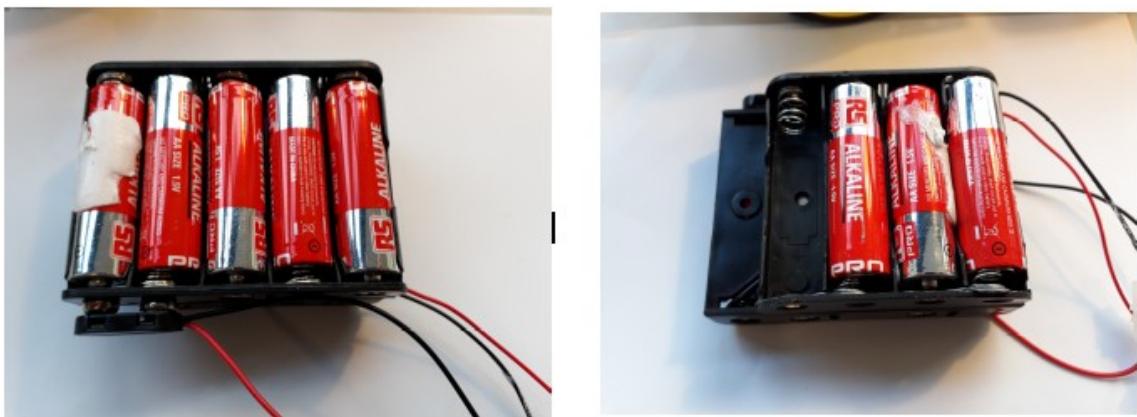


Figure 12: Filling the battery holder with batteries. [Note one battery has been left out. This is because we don't want the buggy to start working yet!]

Once you have done this, use some more stick pads to attach the battery holder to the buggy, over the axles with the motors as shown in figure 13. Ensure the battery pack is placed on the end above the motors to put the weight of the buggy over the powered wheels.

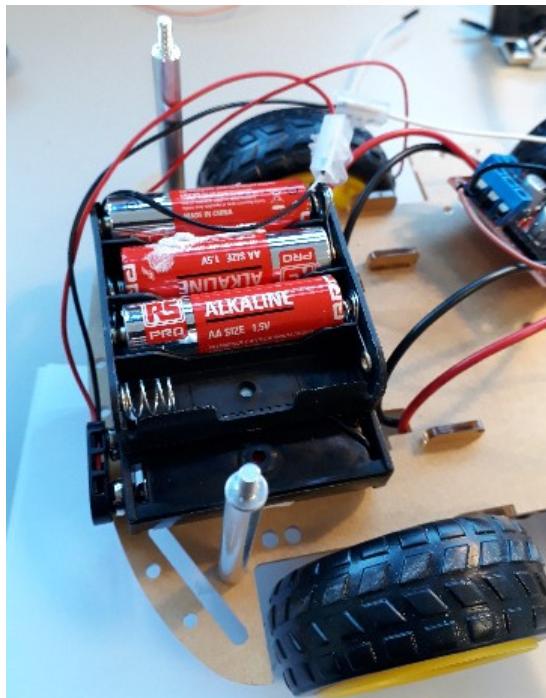


Figure 13: The battery pack attached to the buggy, with sticky pads. Note the battery pack sits over the driving wheels.

Computer programming with MATLAB
Worksheet 1: Buggy construction

Step 12: Although the majority of the wiring will be addressed in the next section, the motor wires will be attached now. Go and find the motor driver board from the buggy kit. It is the circuit board shown in figure 14. The motor wires are attached to the motor driver board at the positions shown in the figure. Use the screwdriver to loosen the terminals, place the wires in the terminals and then tighten the screws up clamping the wires in place. Make sure they are in tightly.



Figure 14: Motor wires connected to the motor driver board, both front and back to their corresponding sides

Step 13: Once you have wired up the wires carrying current to the motors, you then need to add the wires carrying power to the circuit board from the battery. Choose two wires with two male ends and add them screw them into the circuit board as show in figure 15. I've used brown and black wires, but you can use any color wires you want.

Computer programming with MATLAB
Worksheet 1: Buggy construction

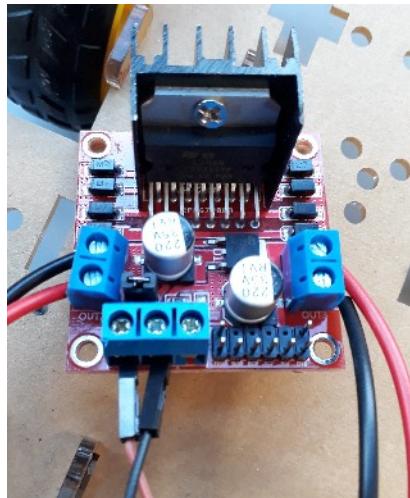


Figure 15: Brown and black wires added to the motor driver board (bottom left). You can use any color wire.

Step 14: Finally, find six female to female, jumper leads and attach them to the front most pins on the motor driver board, shown in figure 16. It does not matter what color they are. There may be some jumpers on some of the pins, if there are, just remove them. The final wired up jumper board should look like figure 16. The three wires on the left are to control the left hand motor and the three wires on the right are to control the right hand motor. These are called the motor control wires.

Computer programming with MATLAB
Worksheet 1: Buggy construction

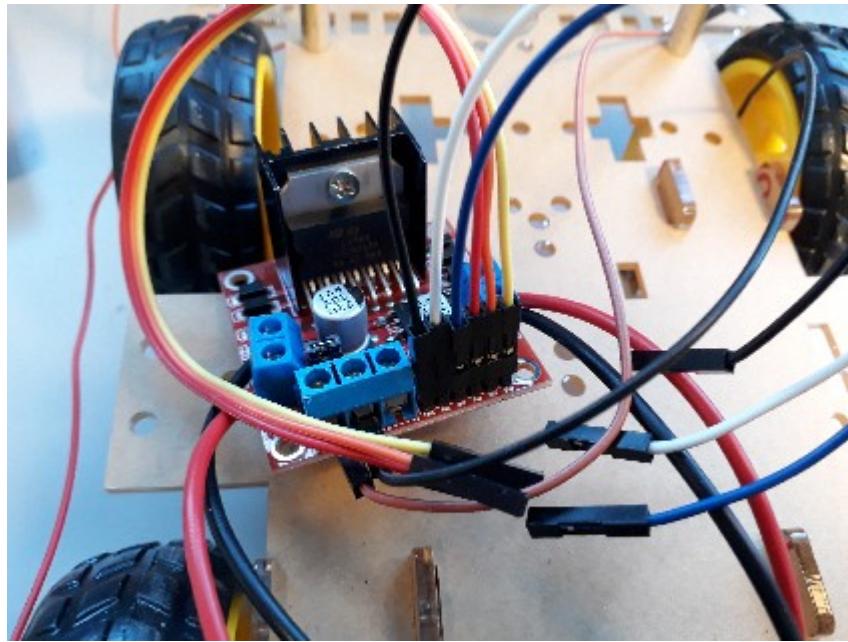


Figure 16: Final wired up motor driver board, with the six female to female cables attached. These are the wires which the computer will use to control the motors.

Step 15: The threaded posts can now be attached to each corner of the base plate. The posts can be placed in all four corners of the base plate as shown in figure 9, and then secured in place with nuts as shown in figure 18.

Computer programming with MATLAB
Worksheet 1: Buggy construction



Figure 17: Threaded post attached to the base plate using a nut on the underside of the plate

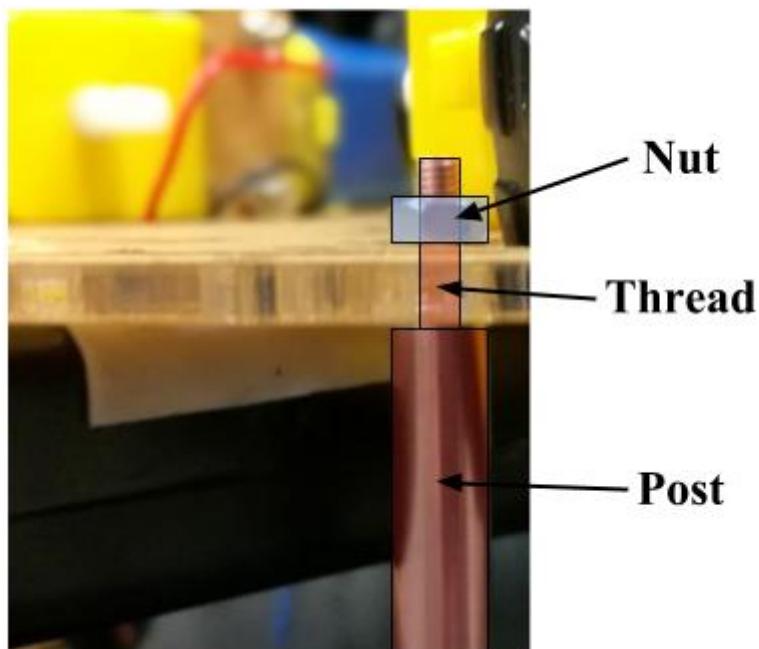


Figure 18: The posts are attached by placing the threaded end through the hole in the corner of the plates and securing with a nut

Computer programming with MATLAB
Worksheet 1: Buggy construction

Step 16: Once all posts have been securely attached to the base plate, the upper plate can be placed on top of the posts and secured with additional nuts. As the plates are not symmetrical, the plate may need rotating until it can fit properly. Thread the wires from the battery from below up through the top plate as shown in figure 19.

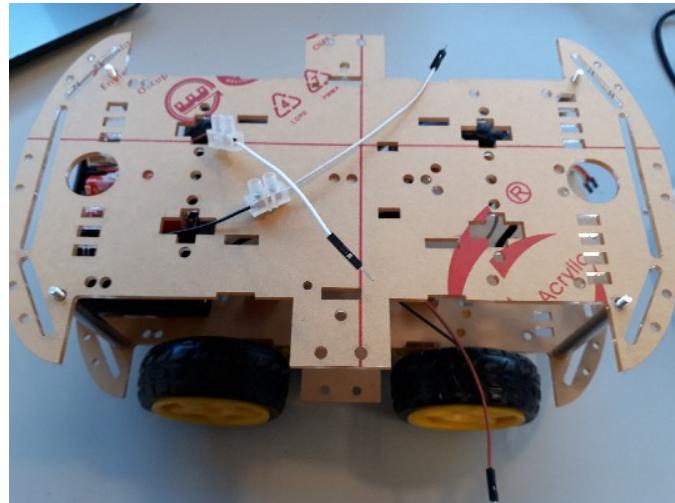


Figure 19: The wires from the bottom of the buggy threaded through the top plate.

Step 17: In step 14 you attached six female-female jumper cables to the motor control board. Take the left hand set of three wires and thread them through the top plate, then take the right hand set of three wires and thread them through the top plate. This is shown in figure 20.

Computer programming with MATLAB
Worksheet 1: Buggy construction

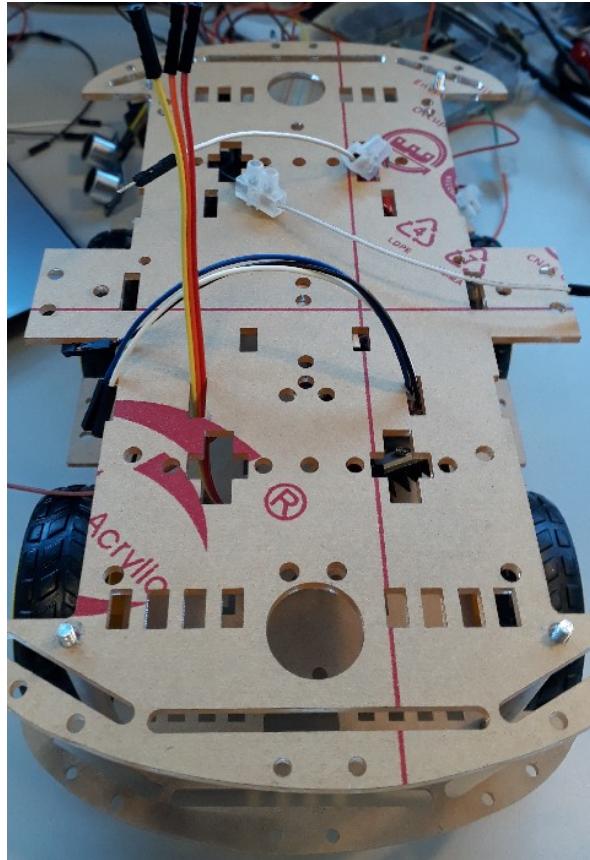


Figure 20: Motor control wires threaded through the top plate.

Step 18: With the top plate in position, attach the remaining components (raspberry pi, breadboard) using the double-sided sticky pads. The components must be attached in the orientation shown in figure 21. **Note: Only use a 1cm² of sticky material, or it will be very hard to dissemble the buggy later on.**

Computer programming with MATLAB
Worksheet 1: Buggy construction

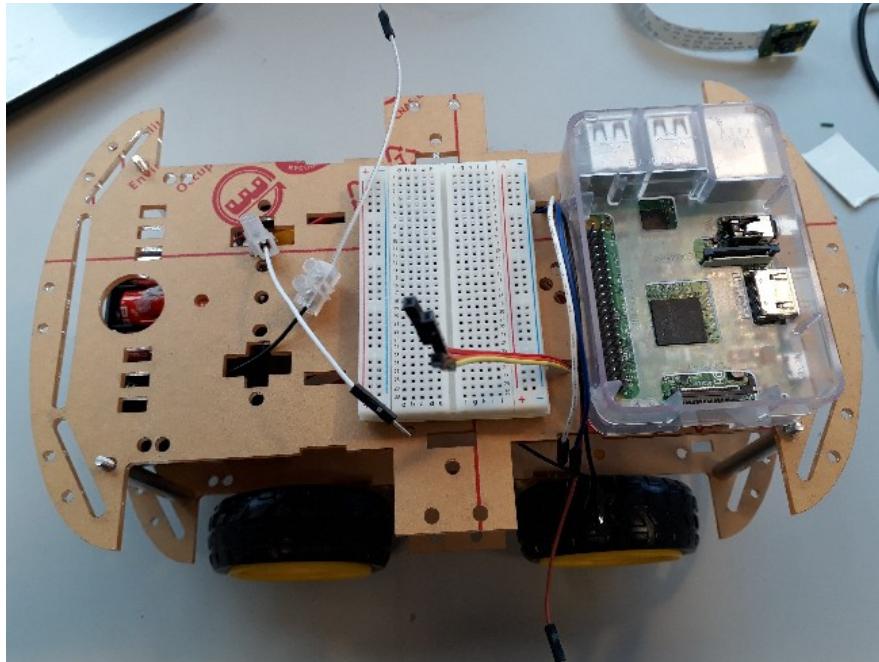


Figure 21: Top plate with all components attached in correct positions

Wiring the buggy

Congratulations you have now built your buggy!! With all the components in position, they can now all be wired up in order for the buggy to work. Due to the complexity of the wiring, each section will be addressed individually in order to reduce the complexity of the process. The buggy won't work until all parts are wired up.

Step 1: In this section, you will be wiring up each system of the buggy individually according to a series of wiring diagrams. Most of the wiring will be based on using an electronic breadboard as shown in figure 18. Breadboards allow you to quickly and easily build circuits by placing wires and components into the holes in the board. Each hole in the breadboard is also electrically connected to a series of other holes (as shown by the red lines in figure 12).

Computer programming with MATLAB
Worksheet 1: Buggy construction

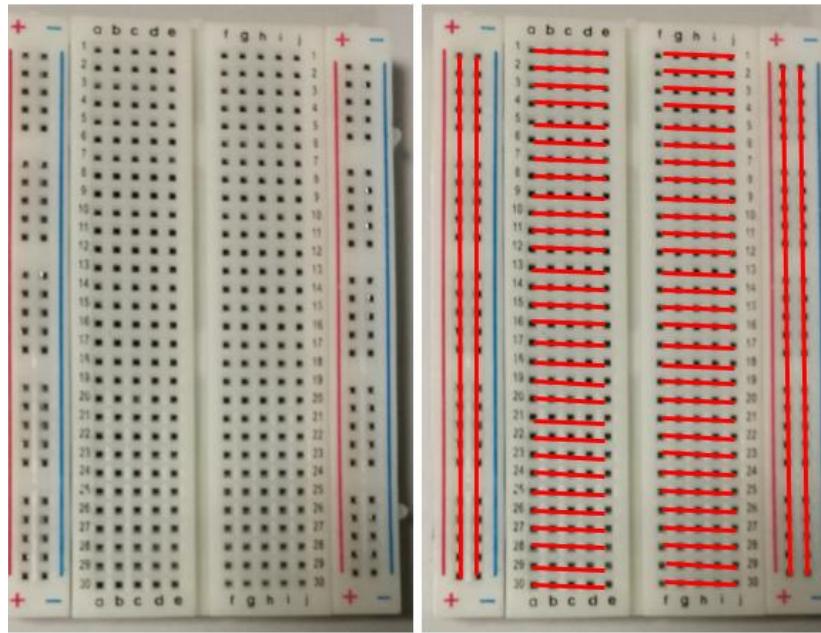


Figure 22: Layout of the breadboard, with the image on the left showing the common connections between pins.

Step 2: The first system to be wired is the power supply for both the motors and the PI. In order to connect all the components, you will need jumper cables, (it does not matter which color jumper cables you use), a voltage regulator, switch and a battery connector. An image of the voltage regulator is shown in figure 23. The voltage regulator takes the voltage supplied by the battery (which can vary depending on how used the batteries are but should be around 12V), and ensures it's a nice steady 5V for the PI to use. Now go and find these components in the buggy kit.



Figure 23: The voltage regulator

Computer programming with MATLAB
Worksheet 1: Buggy construction

Step 3: Once you have the correct components. First mount the voltage regulator on the board as shown in figure 24. Make sure it is the right way around, and in exactly the same position as shown in the figure. Then connect the wires as shown in figure 25, using the jumper cables.

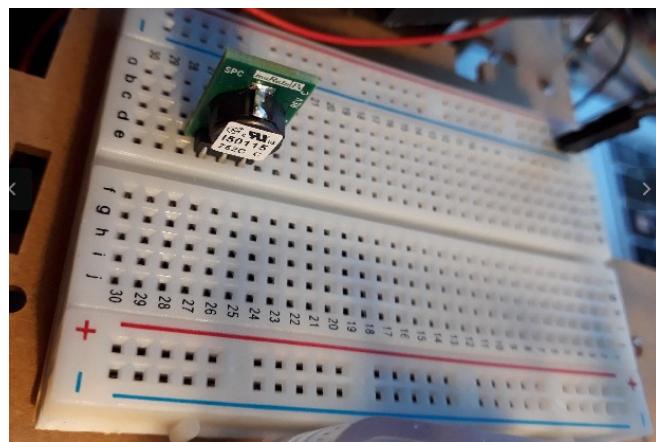


Figure 24: How to mount the voltage regulator on the board.

Note: You must ensure you connect the jumper cable to the correct pin on the raspberry pi in order for this to work

Computer programming with MATLAB

Worksheet 1: Buggy construction

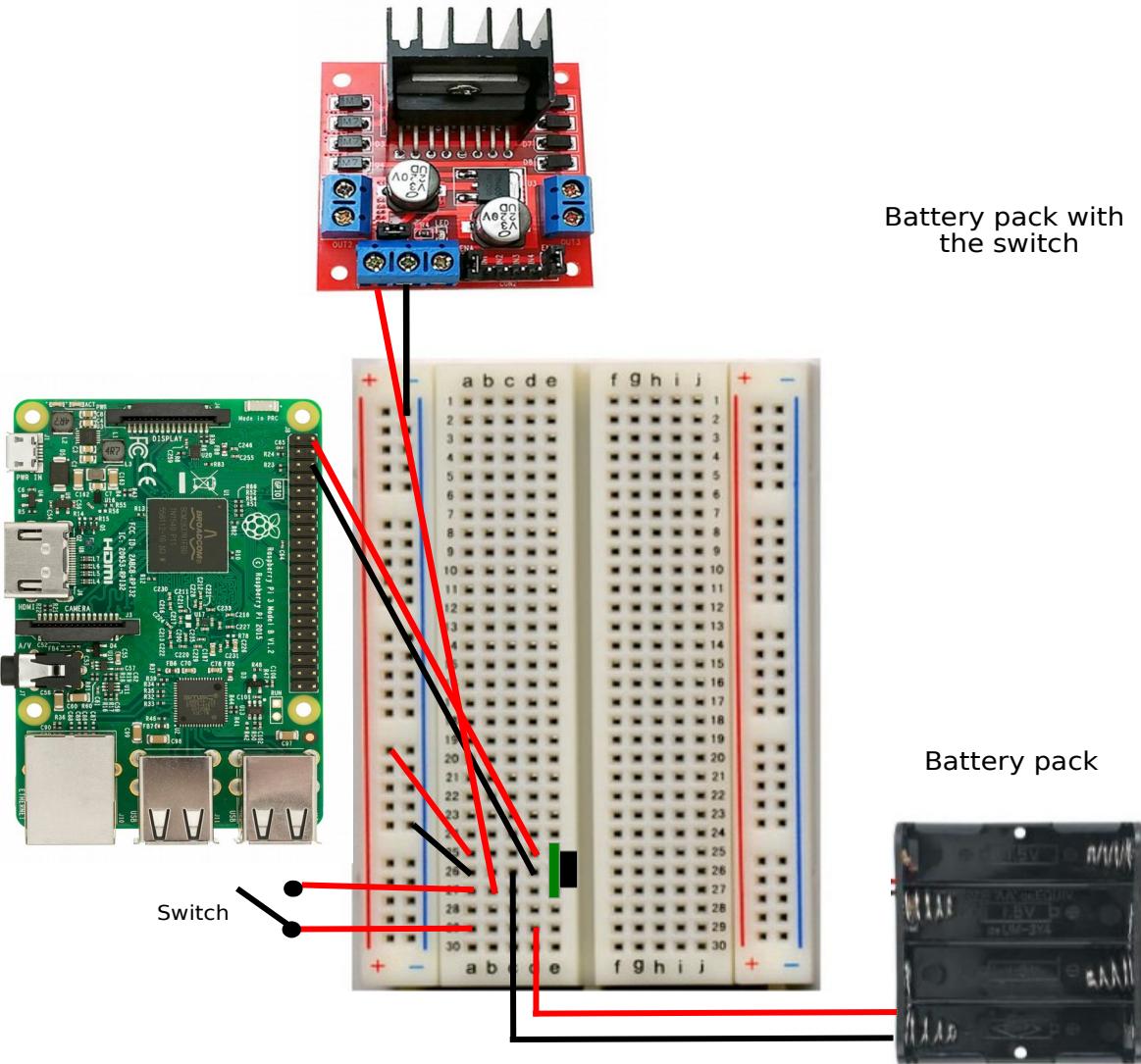


Figure 25: Circuit diagram for powering the raspberry pi and motors.

Computer programming with MATLAB
Worksheet 1: Buggy construction

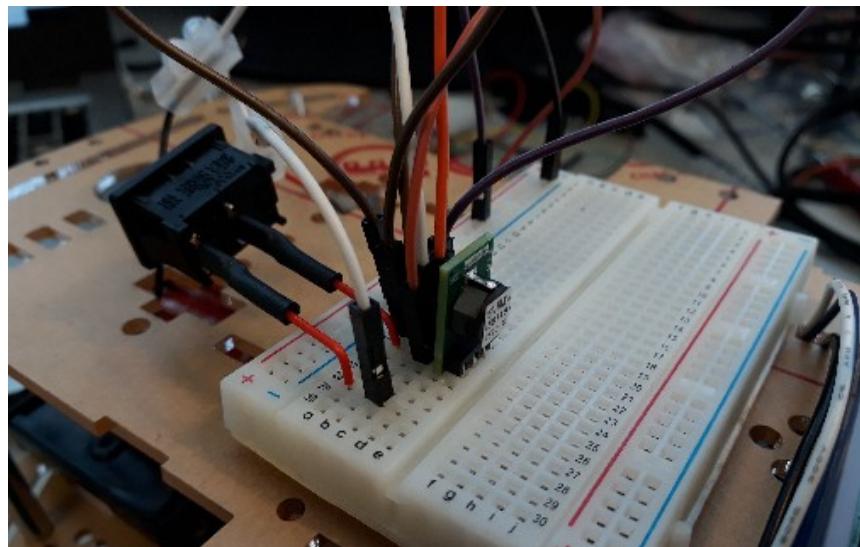


Figure 26: Close-up of the breadboard after the batteries have been wired

Step 4: Once the power supply to batteries and PI have been connected, you can now connect the output pins that will control the motors. Using the female-to-female jumper cables which you have already threaded through the top plate of the buggy, connect the raspberry pins to the motor controller board according to the diagram in figure 27. Again, ensure you attach the correct pins, otherwise the motors will not respond to commands.

Computer programming with MATLAB
Worksheet 1: Buggy construction

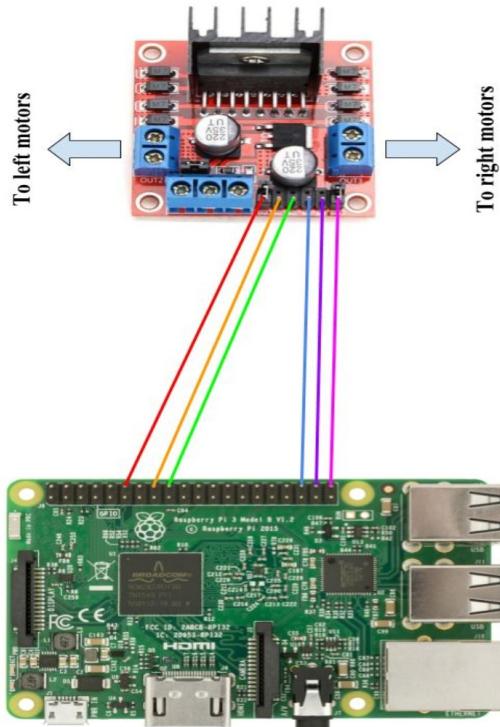


Figure 27: Circuit diagram for assigning GPIO pins to control the motors

Step 5: You will now connect the echo distance sensor to the raspberry pi. The echo sensor has four pins: 5V power, ground, trigger and echo. The sensor works by sending an ultra-sonic pulse, and then recording the time it takes to detect an echo. Using this time and the speed of sound, the distance to the nearest object in front of the sensor can be determined. As with the motor controller, you can connect the echo sensor the raspberry pi using the jumper cables. However, you must use two resistors (R1 and R2) in order to correctly use the sensor.

Step 6: In order to correctly connect the echo sensor, you must first place two resistors, R1 and R2, with resistances of $4.7\text{ k}\Omega$ [color code yellow, purple red, gold] and $10\text{ k}\Omega$ [brown ,red, black, black, brown] respectively. The resistors are to be placed onto the breadboard in the positions shown in figure 28. You will have to trim the end of the resistor wires in order to get them to sit flush to the board, as shown in figure 25.

Computer programming with MATLAB
Worksheet 1: Buggy construction

Step 7: Once the resistors are in place, you can connect the rest of the pins using the jumper cables according to the diagram in figure 16. Once the echo sensor is connected, you can then use blue tack to attach it to the top plate battery pack as shown in figure 27. If you don't have blue tack, it should just sit there on its own.

Computer programming with MATLAB
Worksheet 1: Buggy construction

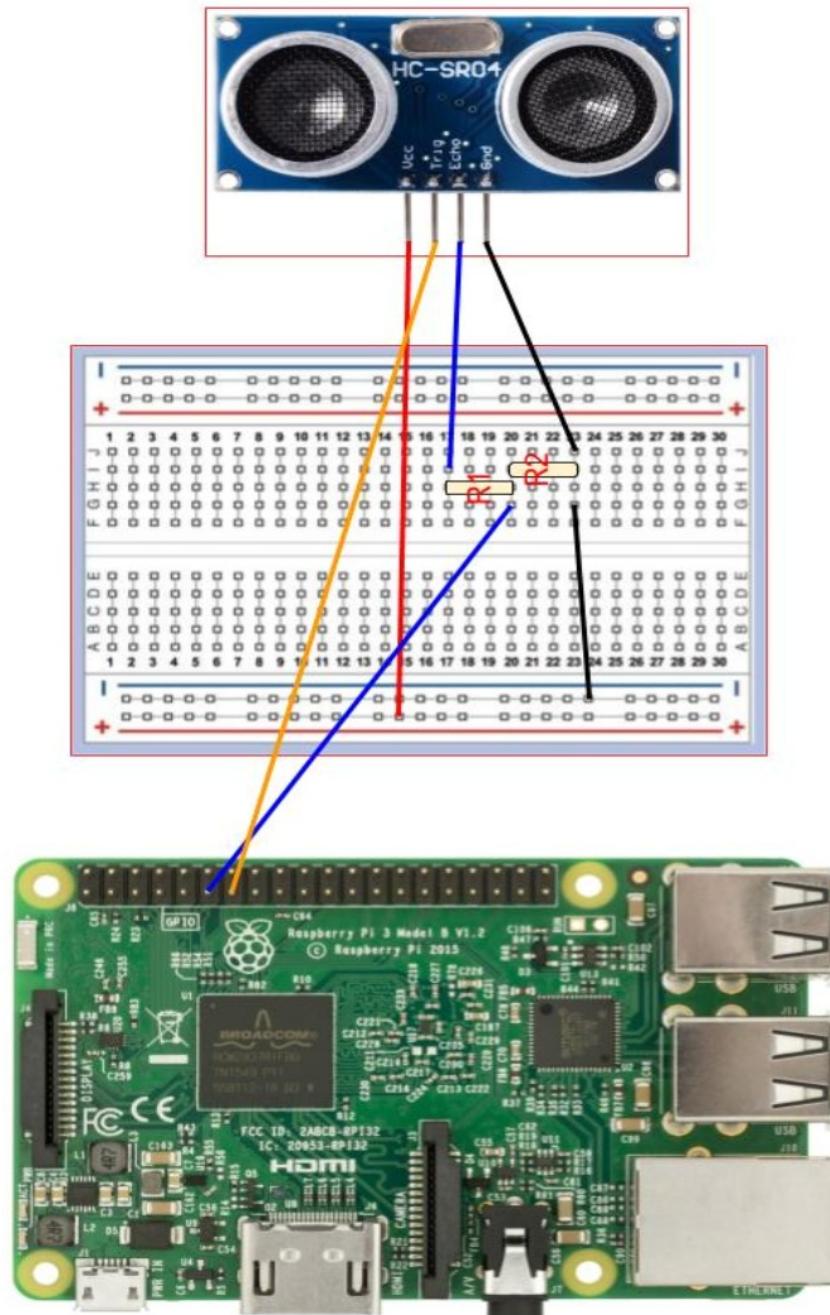


Figure 28: Circuit diagram for the ultrasonic sensor

Computer programming with MATLAB
Worksheet 1: Buggy construction

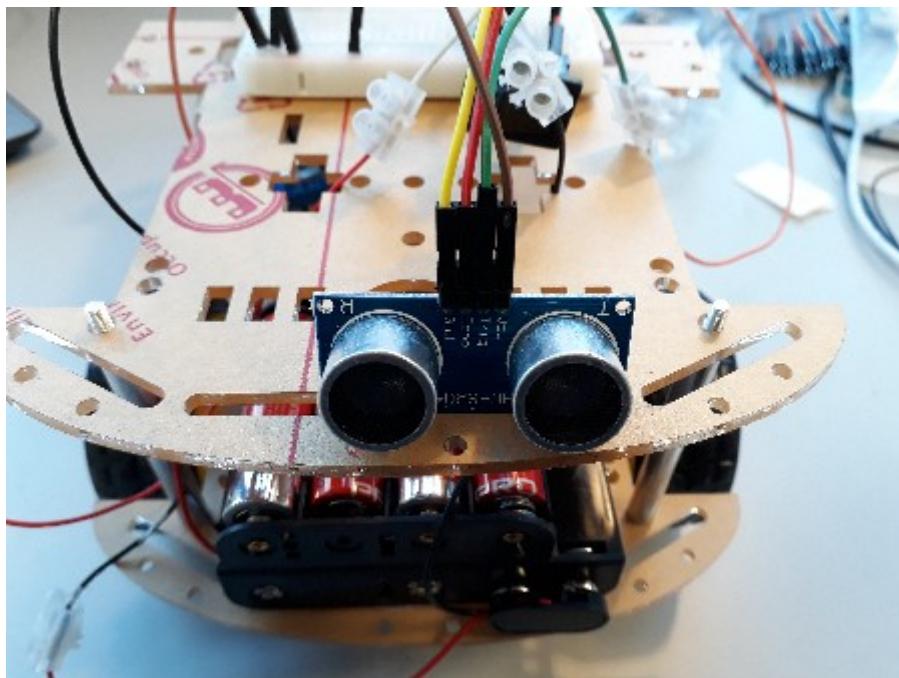


Figure 29: Image of ultrasonic sensor mounted about the battery pack. It should just sit in the grove.

Step 8: Once all the wiring is complete, it should look like the image shown in figure 28. At this point, the wiring is complete and the top plate can be secured with nuts four M3 nuts.

Computer programming with MATLAB
Worksheet 1: Buggy construction

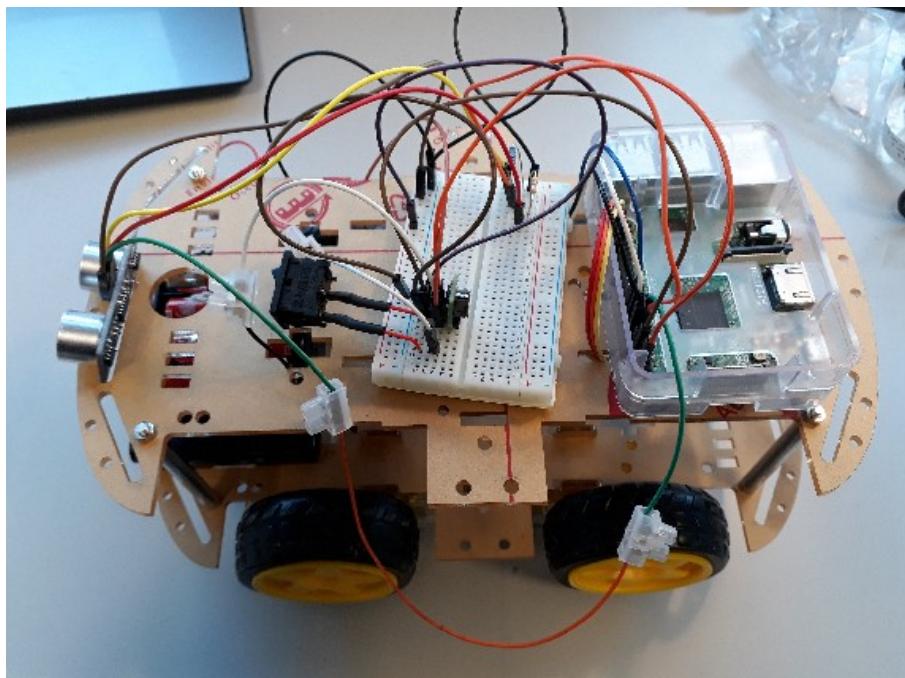


Figure 30: Image of how the buggy should look once fully wired.

Step 9: Finally the camera and camera mount can be attached to the buggy. First, slide the ribbon cable of the camera through the slot in the front of the buggy (figure 31). The camera is then connected to the raspberry pi using a ribbon cable. To connect the camera to the pi, you must lift the black clip shown in figure 31 and slide in the ribbon cable with the metal contacts facing away from the USB ports. Once the ribbon cable is in place, the black clip can be pressed down, locking the ribbon cable in. **If you struggle with this ask a demonstrator for help!! They ribbon cable is very easy to damage.**

Computer programming with MATLAB
Worksheet 1: Buggy construction

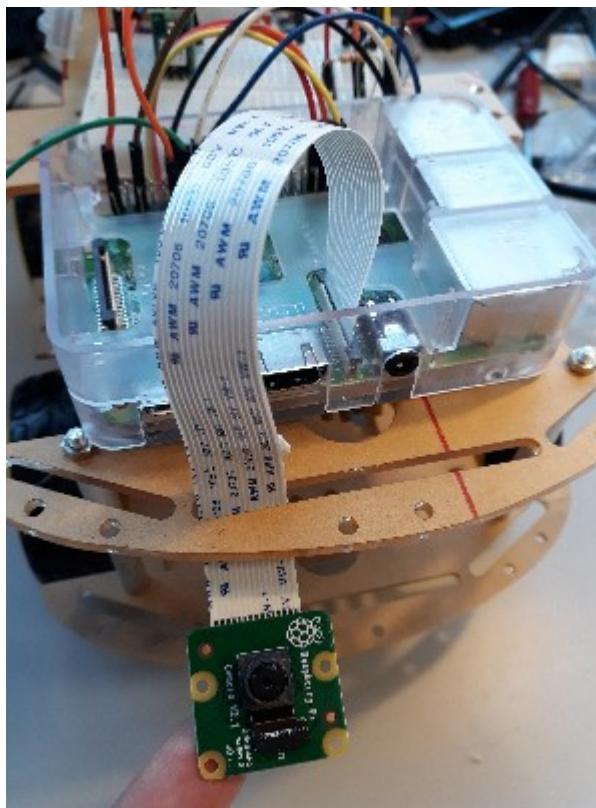


Figure 31: Camera board ribbon cable attached to the raspberry pi

Step 10: With the pi camera connected, you can now place it inside the camera mount as is shown in figure 32. The camera mount can then be attached to the raspberry pi using the M3 bolts and nuts, as can be seen in figures 30 and 31.

Computer programming with MATLAB
Worksheet 1: Buggy construction

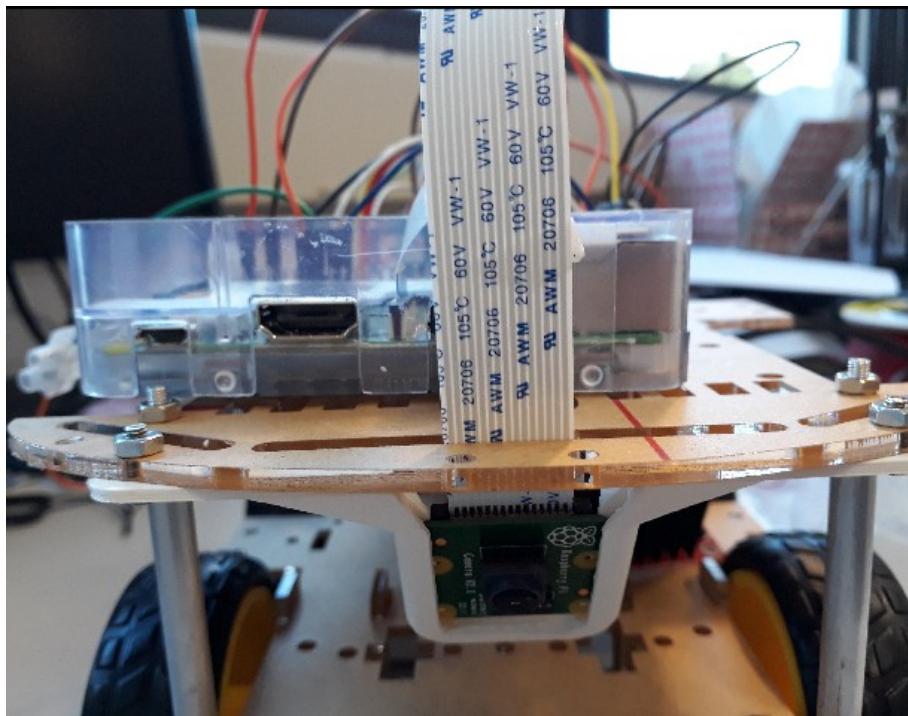


Figure 32: Camera board in the mount, attached to the underside of the top plate

Testing the Buggy

Now the Buggy is complete. Ask a demonstrator to come over and check all the wiring is OK. If all is well, place the buggy on the ground and turn it on. The image that was written to the PI contains a script that will automatically run when the raspberry pi is turned on. This stage will allow you to make sure that the buggy has been built properly, so ensure you look to make sure the motors are functioning correctly and in a straight line. If the echo sensor is working correctly, the buggy should reverse when in proximity to solid objects.

In the case of the buggy not functioning correctly, get a demonstrator to come over and check the connections.

Marked assignment due at the end of the lab. This is worth 20% of this series of labs:

Get a demonstrator to take a photo of your group with the completed buggy with a camera phone. You must then each individually upload this photo to moodle into the hand in box called “Buggy: Assignment 1”. Really this is just a nice way of checking that you were at the lab and were involved in building the buggy. Not all assignments in this lab will be that easy. ;)