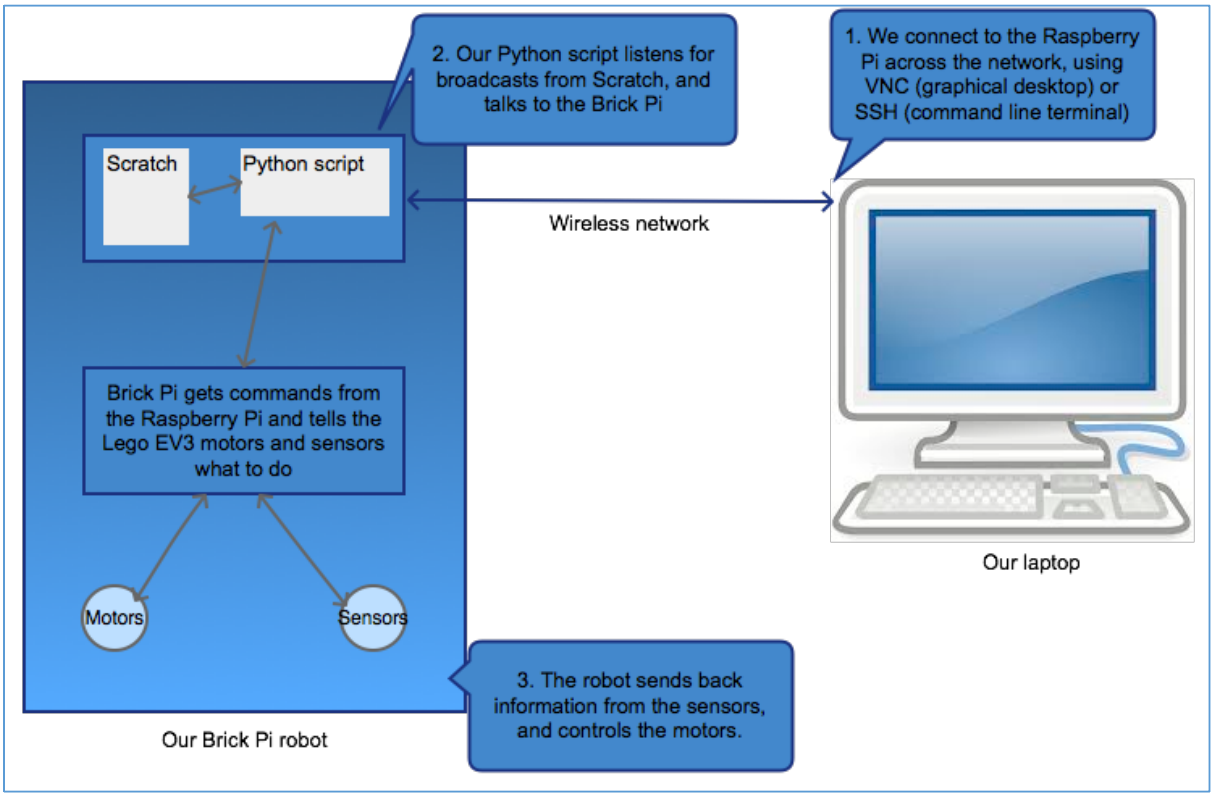
Scratch Brick Pi robot workshop

Here are the steps we will take with this workshop:

- Connect to the Brick Pi desktop, over the network, using VNC
- Learn how to use the keyboard to control the Brick Pi robot in Scratch
- Learn how the robot's sensors work
- Modify the scripts to allow the robot to control itself using information from the sensors

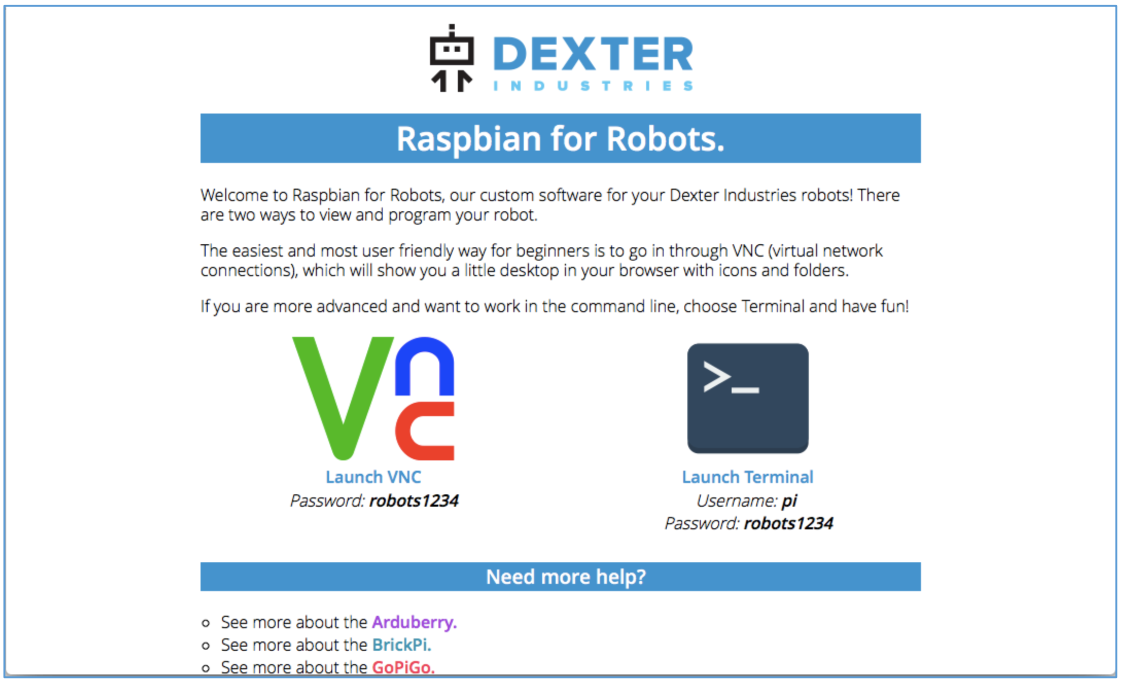
Connect to the Brick Pi desktop, over the network, using VNC

The diagram below shows how we initially work with the Brick Pi robot:


2. Our Python script listens for broadcasts from Scratch, and talks to the Brick Pi
1. We connect to the Raspberry Pi across the network, using VNC (graphical desktop) or SSH (command line terminal)
Scratch
Python script
Wireless network
Brick Pi gets commands from the Raspberry Pi and tells the Lego EV3 motors and sensors what to do
Our laptop
Motors
Sensors
3. The robot sends back information from the sensors, and controls the motors.
Our Brick Pi robot

So let's get started. First, boot your computer into OS X. Open the Safari browser and type the IP address of the robot, into the address bar. Your robot should be set up already and labelled with the IP address.

You should see the following displayed in Safari:



Raspbian for Robots.

Welcome to Raspbian for Robots, our custom software for your Dexter Industries robots! There are two ways to view and program your robot.

The easiest and most user friendly way for beginners is to go in through VNC (virtual network connections), which will show you a little desktop in your browser with icons and folders.

If you are more advanced and want to work in the command line, choose Terminal and have fun!

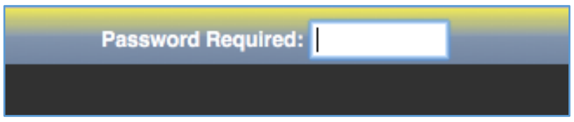
Launch VNC
Password: ***robots1234***

Launch Terminal
Username: ***pi***
Password: ***robots1234***

Need more help?

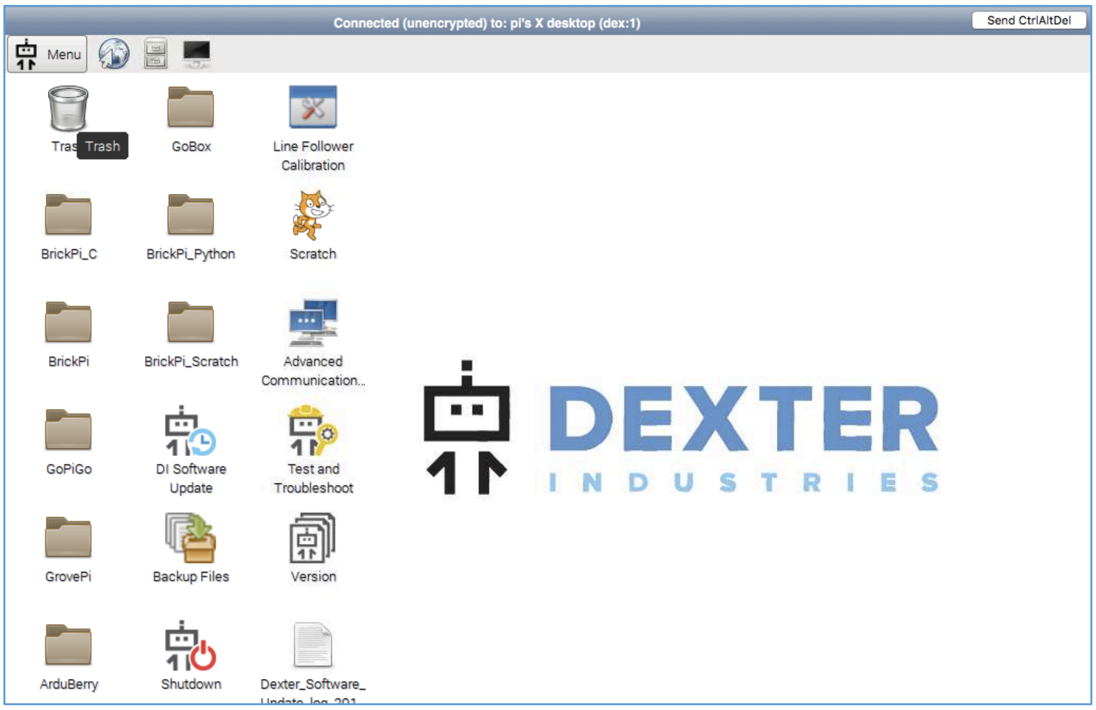
- See more about the Arduberry.
- See more about the BrickPi.
- See more about the GoPiGo.

Click on the icon that says 'Launch VNC'. This will let us connect to the Linux desktop on the Raspberry Pi computer that drives the Brick Pi robot. You'll see the VNC password on this page. Make a note of that as you'll need it to get to the next step (below).



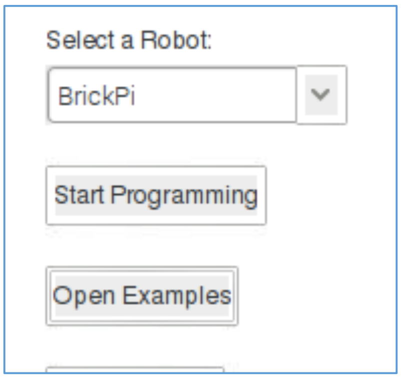
Password Required:

Now you should be able to see the Linux desktop running on the Raspberry Pi computer part of our robot.



Learn how to use the keyboard to control the Brick Pi robot in Scratch

Click on the Scratch cat icon. You should see a terminal window open (don't close it), and a menu open up in front. Select 'BrickPi' then click on 'Open Examples'.



You'll see a window open with some Scratch example projects. Click on Car.sb2 then select 'Brick Pi' and click on 'Start programming'. Say ok and Scratch will open. You'll see a popup saying 'Remote sensor connections enabled'. This means the Python script is running in the background and is connected to Scratch. Click 'ok' and you should see the Car example project open.

Have a look at the scripts and you'll see the broadcasts that send instructions to the Python script, that control the robot motors. The motors have labels that being with 'M' e.g. MA, MB, MC, MD. These are labelled on the Brick Pi board, where it connects to the motors. Although this project doesn't use sensors, it's worth mentioning that these will use a similar labelling pattern but just starting with 'S', followed by a number e.g. S1, S2, S3, S4 and so on.

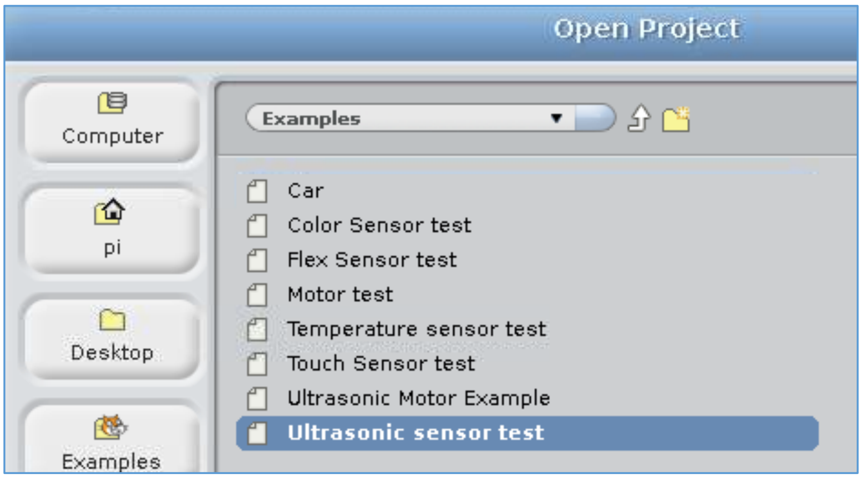
Make sure that the robot is on a stand, so the wheels can spin freely, or is placed where it will not drop off the side of a table. The 'space' key will stop all motors, and the arrow keys will run the motors. Try pressing the keys and you should see the robot move.

Can you work out what each of the broadcasts do? What is wrong with the controls? Try and fix the controls by editing the code, and then check that it worked correctly. The arrow sprite will display some information related to the commands sent to the robot.

Save the project with a new name, then make a video or make some notes on your progress!

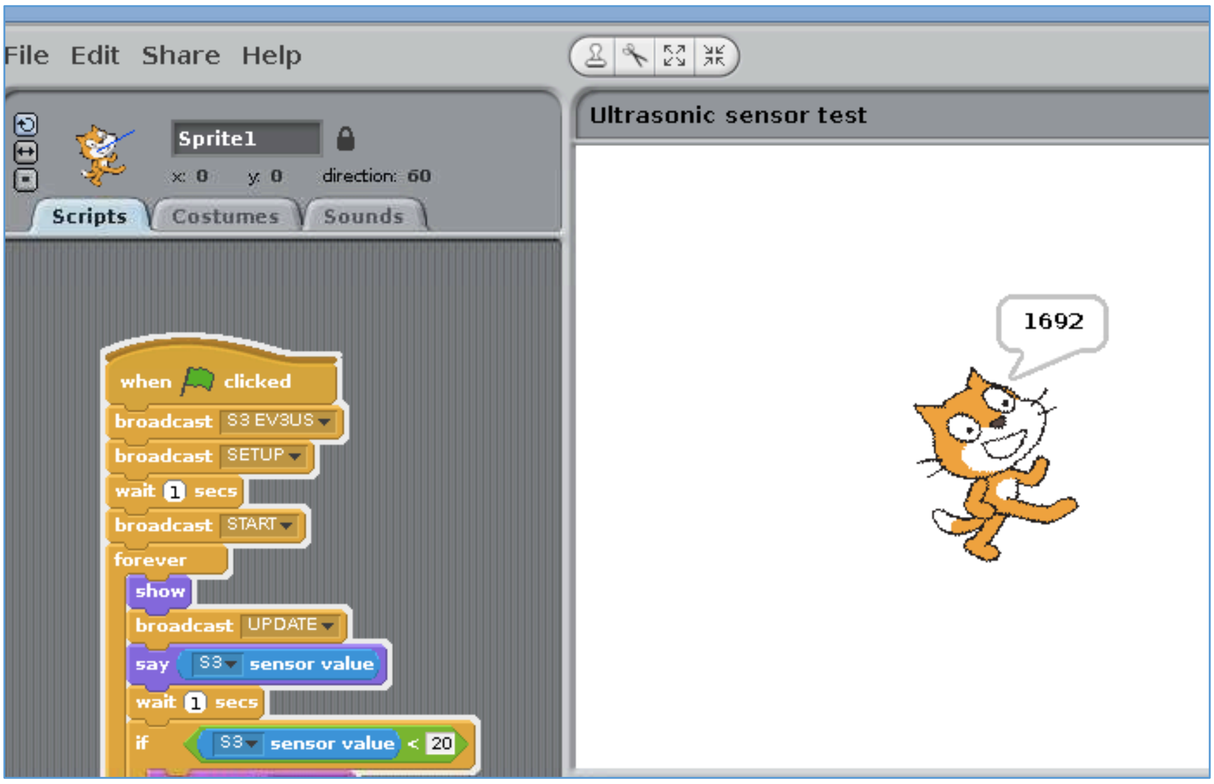
Learn how the robot's sensors work

Now go to the 'File' menu and load the Ultrasonic sensor test:



Click ok to the remote sensor connection popup again, and take a look at the code in Scratch.

The sensors take a few seconds to set up when the script runs. Once it is running, you should see a number next to Scratch. This is the distance that the Ultrasonic sensor is reading. It uses sound to bounce off objects to see how far away they are.



```
when [flag] clicked
broadcast S3 EV3US
broadcast SETUP
wait 1 secs
broadcast START
forever
  show
  broadcast UPDATE
  say S3 sensor value
  wait 1 secs
  if S3 sensor value < 20
```

When you look at the script, you will see that there are 3 broadcasts before the main forever loop.

The first broadcast is ‘S3 EVBUS’. This tells the Brick Pi that the EV3 Ultrasonic sensor is connected to the S3 connection (again, this is labelled on the robot). The SETUP and START broadcasts get the sensor set up and started.

Once this is done, every time the ‘UPDATE’ broadcast is issued, a value is read from the sensor. The ‘if’ block tells the script to perform an action if the sensor value is smaller than 20 centimeters.

Place your hand in front of the sensor and watch the sensor burn straight through your flesh... Only joking – you should be able to move it back and forth, and see the sensor value change.

Modify the scripts to allow the robot to control itself using information from the sensors

Now that you can see how the Ultrasonic sensor works, we can combine the sensor information and use it to help control the robot. Wow, our robot will soon be autonomous!

Load up your Car project again. Then go to the Scratch ‘File’ menu and select ‘Import Project’ to import project file that combines the motors and the sensor code. Have a discussion about what you would like to do e.g. make the robot stop if an object is in front of it, or make the robot turn if something is within a certain distance. Once your team has decided what you want to do, make some notes so that you can model the idea. This is the first step in creating a solution to a problem using programming.

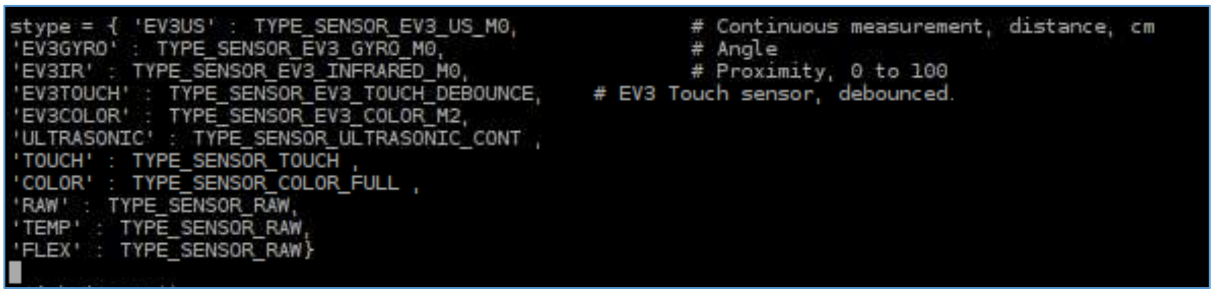
Your notes should identify the main parts of the code and what they do. It doesn’t matter if they are simple notes or aren’t very clear, everyone starts somewhere. Keep it simple, because we want it to work well, and then you can add more ideas later!

Make some changes to the code, and test it out. You will need to try different values to get it to work just right. Once it is working, save your project under a different name from the template script, and take videos and talk about what you’ve achieved.

Congratulations! You’ve successfully taken a remote controlled robot and made it autonomous!

Extra mile:

You may have noticed that there are more motors and sensors that you can use. It should be simple enough to figure out the motors, but for the sensors, you’ll need to assign more sensors to the right connections. The sensor types you can add (if you have them handy) can be seen in the section of the Python script shown below:



```
stype = { 'EV3US' : TYPE_SENSOR_EV3_US_M0,              # Continuous measurement, distance, cm
'EV3GYRO' : TYPE_SENSOR_EV3_GYRO_M0,                    # Angle
'EV3IR' : TYPE_SENSOR_EV3_INFRARED_M0,                  # Proximity, 0 to 100
'EV3TOUCH' : TYPE_SENSOR_EV3_TOUCH_DEBOUNCE,    # EV3 Touch sensor, debounced.
'EV3COLOR' : TYPE_SENSOR_EV3_COLOR_M2,
'ULTRASONIC' : TYPE_SENSOR_ULTRASONIC_CONT ,
'TOUCH' : TYPE_SENSOR_TOUCH ,
'COLOR' : TYPE_SENSOR_COLOR_FULL ,
'RAW' : TYPE_SENSOR_RAW,
'TEMP' : TYPE_SENSOR_RAW,
'FLEX' : TYPE_SENSOR_RAW}
```

Repeat the last exercise by discussing possible enhancements to add to the robot, write down the steps required, get testing, and see what you can create!