S2Bot QuickStart Introduction

S2Bot is available in two variants – Chrome App or Native Executable. Robots supported include:

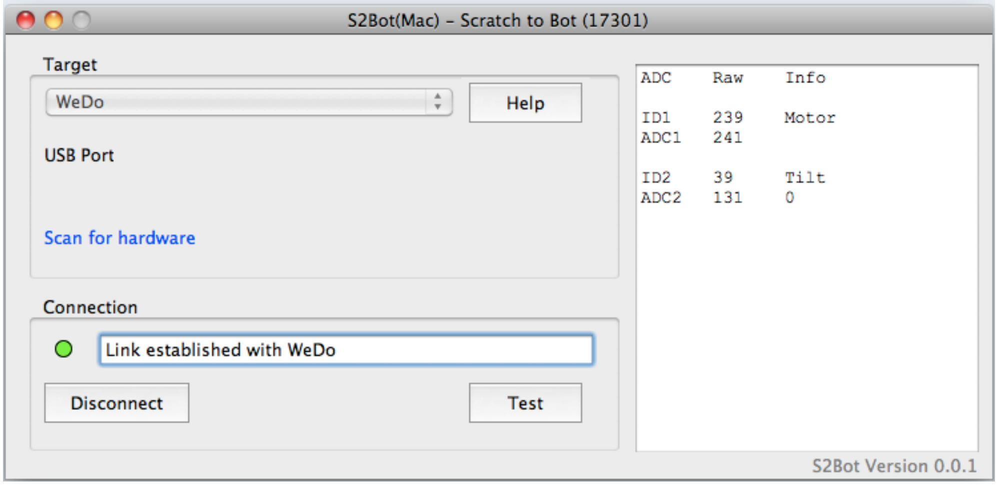
Device	**Connection**	**S2Bot Chrome App**	**S2Bot Native**
LEGO WeDo 2.0	BLE	✓	✓
LEGO WeDo 1.0	USB HID	✓	✓
LEGO NXT	classic bluetooth	–	✓
LEGO EV3	classic bluetooth	–	✓
Vengit SBrick	BLE	✓	✓
Orbotix Sphero	classic bluetooth	✓	✓
Orbotix BB8	BLE	✓	✓
Orbotix Ollie	BLE	✓	✓
PicoBoard	USB VCP	✓	✓
Vernier GoTemp	USB HID	✓	–
Vernier GoMotion	USB HID	✓	–

This manual is available in two formats:

Chrome Version	see www.picaxe.coms/docs/s2bot_app.pdf
Native Version	(this document)

S2Bot - Native Versions:

The native version is a traditional ‘installed’ program and runs on Windows, Mac or Linux.


S2Bot(Mac) – Scratch to Bot (17301)

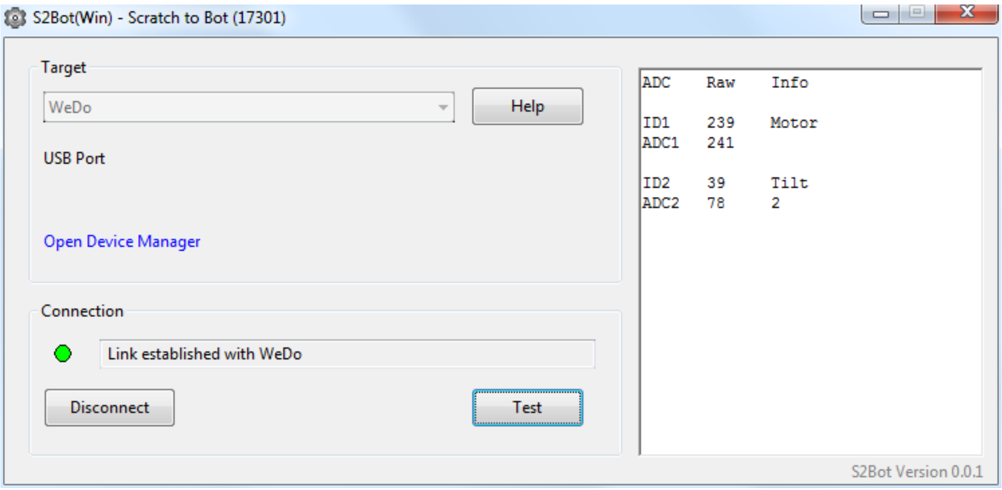
Target: WeDo

USB Port

Connection: Link established with WeDo

```
ADC     Raw     Info

ID1     239     Motor
ADC1    241

ID2     39      Tilt
ADC2    131     0
```

S2Bot Version 0.0.1


S2Bot(Win) - Scratch to Bot (17301)

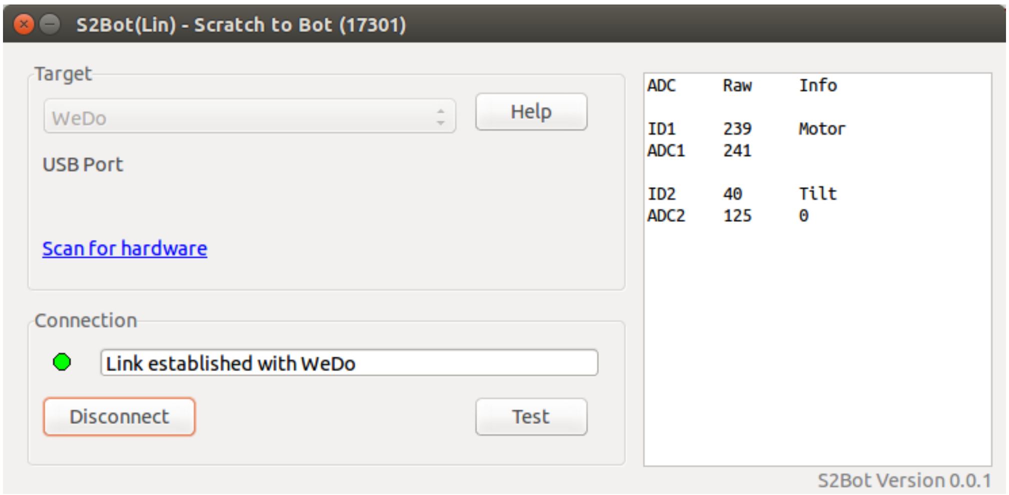
Target: WeDo

USB Port

Connection: Link established with WeDo

```
ADC     Raw     Info

ID1     239     Motor
ADC1    241

ID2     39      Tilt
ADC2    78      2
```

S2Bot Version 0.0.1


S2Bot(Lin) - Scratch to Bot (17301)

Target: WeDo

USB Port

Connection: Link established with WeDo

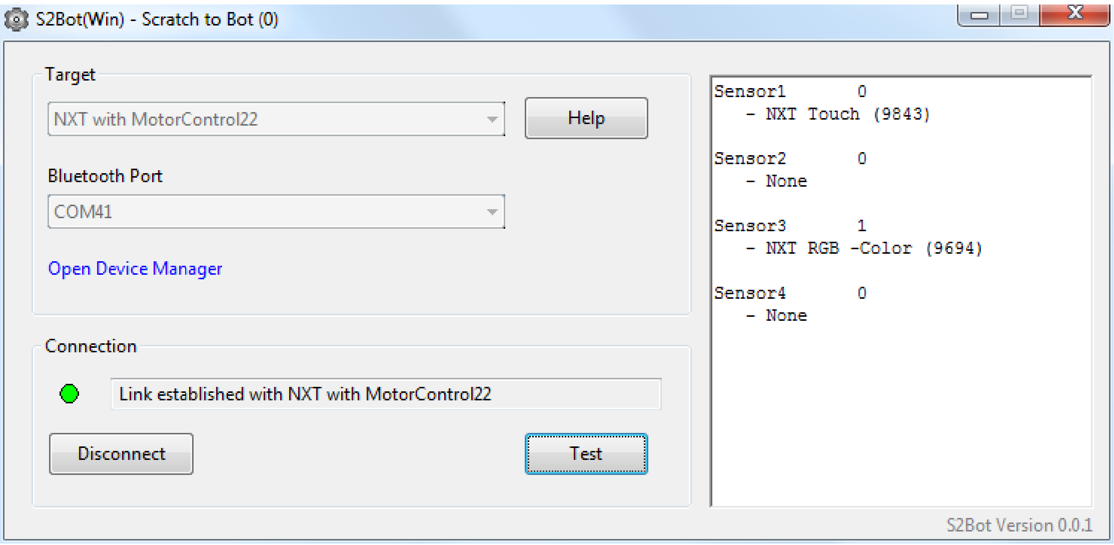
```
ADC     Raw     Info

ID1     239     Motor
ADC1    241

ID2     40      Tilt
ADC2    125     0
```

S2Bot Version 0.0.1

Note that any firewall must allow localhost http communication on ports 17300 to 17320. This is to allow Scratch to 'talk' to S2Bot on the local computer

Contents

1.0 Quick Start Notes (Native Version)



Target: NXT with MotorControl22

Bluetooth Port: COM41

Connection: Link established with NXT with MotorControl22

```
Sensor1      0
   - NXT Touch (9843)

Sensor2      0
   - None

Sensor3      1
   - NXT RGB -Color (9694)

Sensor4      0
   - None
```

S2Bot is a free helper app to allow control of robotics systems from Scratch 2 (both the online and offline versions of Scratch are supported). It is a free download from http://www.picaxe.com/s2bot

After installing the S2Bot helper app double click on the 'gear' icon to run it (under Linux make sure it has executable permission - right click>Properties>Permissions>Execute).

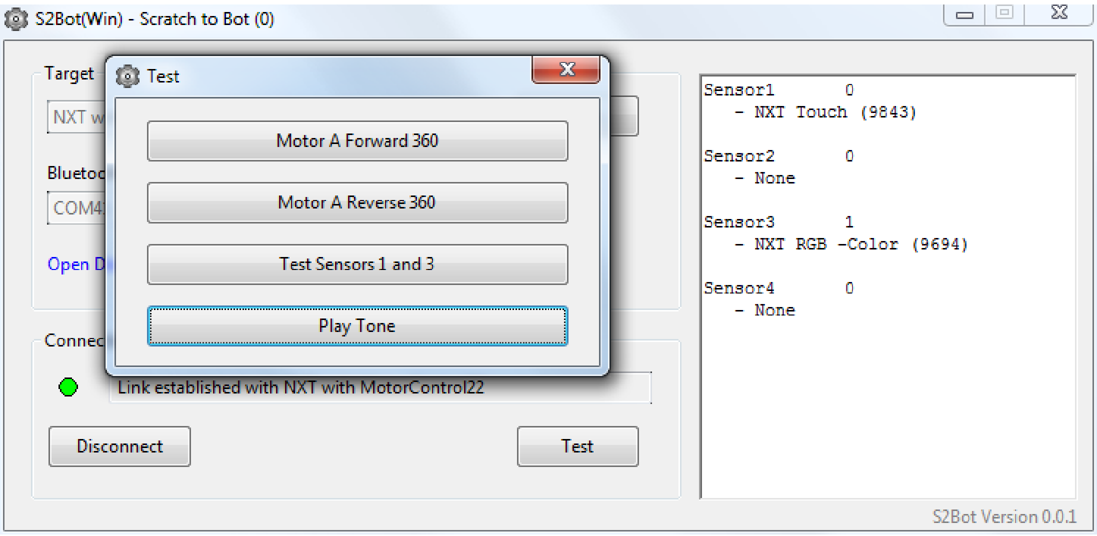
Select the desired robot interface type from the 'target' list.

(If using a classic bluetooth device make sure it is now switched on and paired and connected – see the appendices at the end of this document).

Select the correct port from the list (or type it in) and then click 'Connect'.

If you do not know the correct port then use the 'Device Manager' (Windows) or 'Scan' (Mac / Linux) link to help find the port name. More details on this are in the appendices.

If all is well you will then start to see sensor readings in the right hand box. You can also use the 'Test' button to see if your robot outputs are working correctly.



Link established with NXT with MotorControl22

```
Sensor1      0
   - NXT Touch (9843)

Sensor2      0
   - None

Sensor3      1
   - NXT RGB -Color (9694)

Sensor4      0
   - None
```

From this point on you generally just leave S2Bot running in the background and now work within Scratch instead.

Start Scratch 2 (either online or offline version – make sure you have the latest version).

Open the appropriate .sb2 template file for the interface you are using.

In the Offline version select	File > Open
In the Online version select	File > Upload from my computer

(Note that the templates are supplied as 'write-protected' files, so you must save your project with a new name).

The special extra interface blocks will now be available in the 'Other Blocks' palette. If Scratch is 'talking' correctly to S2Bot the dot beside the extension name in 'Other Blocks' will turn green.

DO NOT USE 'ADD AN EXTENSION' BUTTON IN SCRATCH. This is not required as the special blocks come from the template file instead.

Sensor values can be easily tested by checking the checkbox beside the reporter block on the 'Other Blocks' tab. When checked the values will be constantly displayed/updated on the Stage.

That's it - have fun and be creative!

1.1 Sharing Projects

Unfortunately you cannot currently share projects that use hardware extensions on the Scratch website. Therefore please keep all extension based projects private.

1.2 Install the BLED112 Dongle (if required)

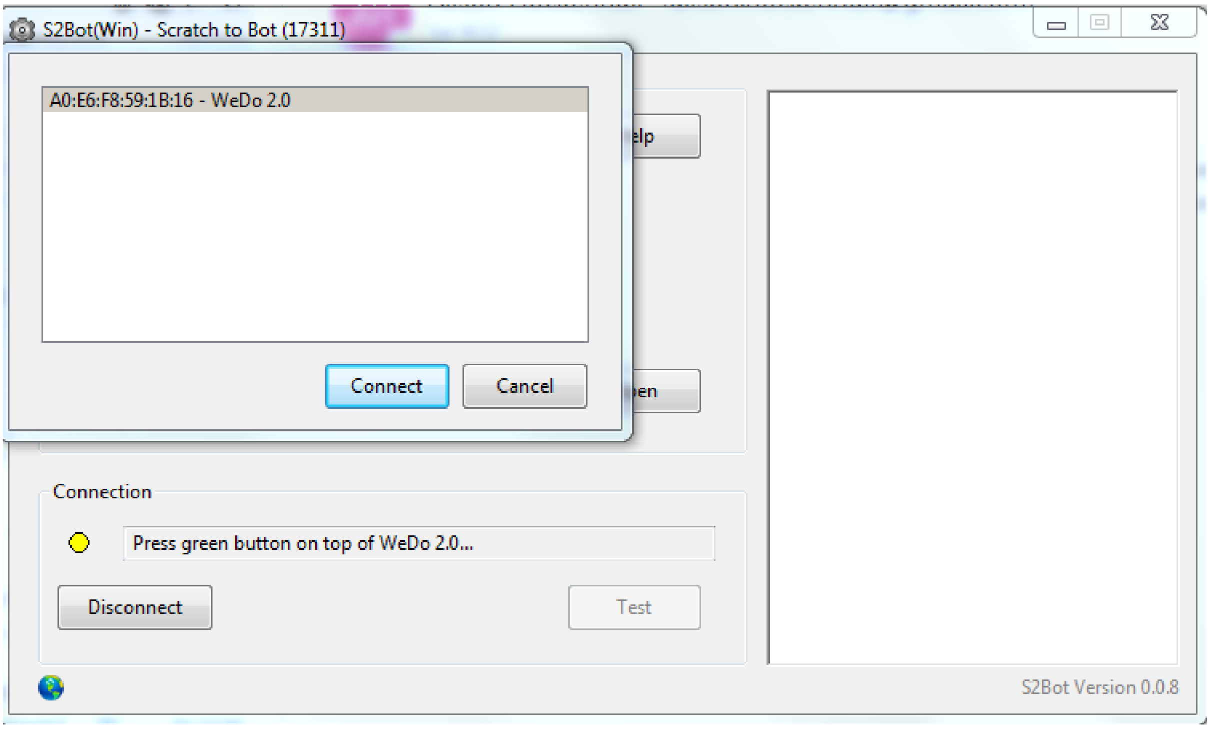
Make sure the BLED112 dongle is installed (if required).

On Windows, the very first time the BLED112 dongle is used a driver will install (see appendix A) and then the computer must be restarted.

On Mac OSX / Linux no driver is required.

1.3 Connecting a BLE Device

1. Insert the BLE112 dongle
2. Start the S2Bot app
3. Select the robot device from the S2Bot drop down list
4. Select the 'Bluegiga Bluetooth Low Energy' COM port from the COM port drop down list
5. Click the 'Connect' button
6. A list of available robots will be shown.
7. Click on the desired robot and then click 'Connect'


S2Bot(Win) - Scratch to Bot (17311)

A0:E6:F8:59:1B:16 - WeDo 2.0

Connection

Press green button on top of WeDo 2.0...

The robot is now ready for use. If desired it can be tested using the buttons within the 'Test' dialog.

Now:

1. Click the .sb2 template file to save it onto your desktop.
2. Start Scratch 2.0 and open this new template file. Opening the template file will automatically configure Scratch correctly and add the extra sensor/motor blocks to the 'More Blocks' palette.

In the **Offline** version select

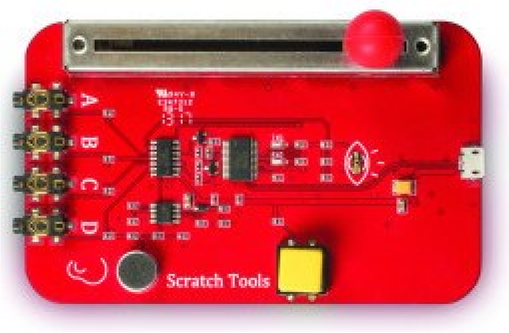
File > Open

In the **Online** version select

File > Upload from my computer

S2Bot should now show the robot device and Scratch as both being connected. The system is now ready for use.

2.0 Notes on using the PicoBoard with S2Bot



The Picoboard normally has a USB connector and connects to the computer via a USB cable. Installation of the USB driver is described in the appendix at the end of this document.

If you have a very old ScratchBoard with a 9 pin D connector you can either use a USB<>Serial Convertor cable (e.g. USB010) or even a traditional 9 pin D serial port if your computer has one.

The PicoBoard can only be used by one piece of software at a time (e.g. do not try to use Scratch 1.4 at the same time as Scratch 2/S2Bot).

The Picoboard does not have any outputs, so the S2Bot 'Test' button is not enabled.

The S2Bot software does not use the in-built 'Add Extensions' feature of Scratch 2, so do not use the 'Add Extensions' button. Instead simply open the .sb2 template file provided, as opening this file will automatically configure Scratch correctly and add the extra sensor blocks to the 'More Blocks' palette.

In the Offline version select	File > Open
In the Online version select	File > Upload from my computer

3.0 Notes on using the Lego WeDo 2.0 (Bluetooth LE) with S2Bot

A BLED112 dongle is **always** required for Windows or Mac use, even if you have bluetooth inside your computer. See further details below.

WeDo 2.0 is a bluetooth smart / bluetooth low energy (BLE) device.

See section 1.3 for details on how to connect the WeDo 2.0 via a bluetooth low energy connection.

The S2Bot software does not use the in-built 'Add Extensions' feature of Scratch 2, so do not use the 'Add Extensions' button. Instead simply open the .sb2 template file provided, as opening this file will automatically configure Scratch correctly and add the extra sensor/motor blocks to the 'More Blocks' palette.

In the Offline version select	File > Open
In the Online version select	File > Upload from my computer

3.1 Lego Input/Output Devices Currently Supported by WeDo 2.0

- WeDo 2.0 Tilt Sensor
- WeDo 2.0 Distance Sensor
- WeDo 2.0 'M' Motor
- Push button (on top of WeDo 2.0)
- RGB LED Light (inside WeDo 2.0)
- Sounder/Speaker (inside WeDo 2.0)

WeDo 1.0 sensors are not supported.

*WeDo 1.0 motors and lights **are** supported, however you must manufacture a simple adapter cable as Lego do not currently sell an 'official' adapter cable.*

3.2 Windows / Mac

You **must** use an **external** BLED112 ‘BLE to serial’ smart dongle. This is due to the Chrome app development system not currently supporting ‘internal bluetooth LE’ on any platform apart from Chromebooks. The BLED112 smart USB adapter overcomes this app limitation via communicating with the operating system as a ‘virtual RS232 Com Port’ instead of as ‘native BLE’.

However one great advantage of this system is that, as the BLED112 is a smart ‘bluetooth to serial COM port’ adapter, it can also be used on older operating systems such as Windows XP or Windows 7 which do not normally support BLE technology. All versions of Windows from XP up to 10 are therefore supported.

You do not need to manually configure any bluetooth stack or ‘pair’ the bluetooth device. The S2Bot app and BLED112 dongle handle all the pairing and communication automatically.

For more details on purchasing and installing a BLED112 dongle please see:

www.picaxe.com/bled112

Do not try to use a cheaper (non-Smart) bluetooth USB dongle from auction sites such as eBay. These will simply not work with the S2Bot system.

4.0 Notes on using the Lego WeDo 1.0 (USB) with S2Bot

The WeDo 1.0 connects to the computer via a USB cable. There are no drivers to install on Windows or Mac, so WeDo should work straight away. For Linux please see the appendix.

The WeDo can only be used by one piece of software at a time (e.g. do not try to use Scratch 1.4 at the same time as Scratch 2/S2Bot).

The Chrome App version of S2Bot supports up to 13 WeDos (motor/sensor A to Z)

The WeDo sometime resets its USB connection (and so disconnects from S2Bot) when a new motor/sensor is snapped onto it. If this occurs you will need to click the S2Bot ‘ Connect’ button again.

The S2Bot software does not use the in-built ‘Add Extensions’ feature of Scratch 2, so do not use the ‘Add Extensions’ button. Instead simply open the .sb2 template file provided, as opening this file will automatically configure Scratch correctly and add the extra sensor/motor blocks to the ‘More Blocks’ palette.

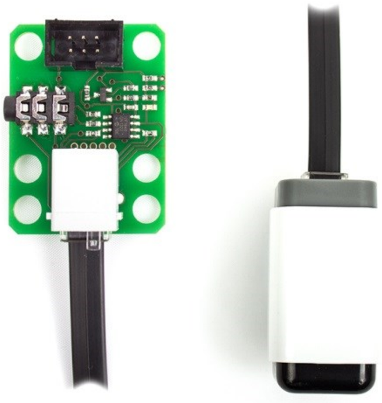
In the Offline version select   File > Open

In the Online version select   File > Upload from my computer

4.1 Lego Input/Output Devices Currently Supported by WeDo 1.0

- WeDo Tilt Sensor
- WeDo Distance Sensor
- Power Functions ‘M’ Motor (or RCX style motor via Power Functions extension cable)
- Power Functions Lights
- Power Functions Servo
- Custom Home Made Sensors

4.2 Advanced WeDo 1.0 use only - custom sensor block



As we have also made our own ‘home-made’ sensors and NXT/Wedo adapters to allow NXT sensors to work with WeDo 1.0 you will also see an extra ‘custom’ reporter block available beside the normal tilt and distance reporters. This allows Scratch to display/use the raw ADC reading from any home-made sensor (as opposed to the scaled tilt/distance values). If you also want to make your own sensors that use the custom reporter block then simply use a 10k resistor from ID to 0V to get S2Bot to recognise this type of custom sensor.

The 4 wires of a cut in half Power Functions cable are: 5V ADC ID 0V

Note also that the ADC already has an internal 10k/200k potential divider inside the WeDo, so the ADC values displayed by the custom block will be the reading of the internal potential divider and your external sensor circuit in parallel.

5.0 Notes on using the Lego NXT with S2Bot

S2Bot does not currently support USB cable connection to the NXT, so you must use classic bluetooth.

The S2Bot software does not use the in-built ‘Add Extensions’ feature of Scratch 2, so do not use the ‘Add Extensions’ button. Instead simply open the .sb2 template file provided, as opening this file will automatically configure Scratch correctly and add the extra sensor blocks to the ‘More Blocks’ palette.

In the Offline version select File > Open

In the Online version select File > Upload from my computer

5.1 Bluetooth

Before using the NXT it must be bluetooth ‘paired’ with your computer. Bluetooth instructions can be found in the appendix at the end of this document.

The default NXT bluetooth pin is 1234, and Bluetooth is switched on via the NXT ‘Bluetooth’ menu. We also recommend you set the NXT Settings>Sleep setting to 30 minutes or more.

The correct bluetooth port to use within S2Bot should be shown within the properties of the Bluetooth Wizard connection. If two ports are shown use the ‘outgoing’ COM port.

Always bluetooth connect the PC to the NXT before starting S2Bot.

5.2 Recommended Classic Bluetooth USB Adapter (if required)

Kensington
idea: The Bluetooth adapter that stays in to stay on
idée : Comme votre adaptateur Bluetooth, restez branché
Idee: Der Bluetooth-Adapter, der immer in Verbindung bleibt
Bluetooth® 2.1 USB Micro Adapter
Bluetooth

Not all bluetooth adapters are the same, some very cheap models can be quite unreliable and hence frustrating to use.

We use a Kensington branded EDR v2.1 miniature bluetooth USB dongle, based on the Cambridge Silicon radio chipset. This cost about £5 ($US 8) from eBay and works very reliably with Windows, Linux and Mac.

It is compatible with Sphero, Lego NXT and Lego EV3.

For bluetooth LE devices use the BLED112 instead.

See the appendices at the end of this document for setup instruction.

5.3 NXT Firmware

S2Bot expects the NXT to be running the normal default firmware (version 1.28 to 1.31). This was a deliberate decision to allow easy swapping of hardware between the Scratch and the Mindstorms 2 software in schools. We may in the future also add a leJOS firmware version option as well, but for now you must use the default Lego firmware.

5.4 Using Sensors

Before using a sensor you need to tell Scratch which sensor is connected to which input port on your NXT brick. To do this drag out a ‘setup sensor’ block and configure it as required. Add these blocks at the top of your program after your ‘green flag’ block, or even simply double click on the block to send the information to the NXT. It is also possible to change the sensor type as the program runs for more complex NXT sensors (e.g. to read X, Y and Z position from the Hitechnic accelerometer).

Use the checkbox beside the 4 sensor reporter blocks in Scratch to display the current sensor value onto the Stage for testing.

5.5 Controlling motors

For simple applications you can use the motor commands to switch individual motors on and off for various amounts of time. Power can be set between 0 and 100 to adjust the motor speed.

However the NXT motors normally ‘coast’ (keep rotating) for a small amount of time after they are switched off. Therefore instructions such as ‘rotate 90 degrees’ simply don’t work properly. This makes accurate control from Scratch over a bluetooth link very difficult (e.g. a robot will turn a little bit more than 90 degrees when instructed, and hence appears to move erratically).

The solution to this problem was developed a few years back by the clever people at RWTH Aachen University. You first download a special ‘MotorControl22’ software file that lives permanently on your NXT. Then all the Bluetooth motor commands are processed directly by the ‘MotorControl22’ program onboard the NXT brick itself (instead of by the computer via direct bluetooth commands). This allows much more accurate control of the motors via feedback from the position sensors in the NXT and so the motor can then be moved much more accurately. Therefore within S2Bot accurate motor commands are only available within the ‘NXT with MotorControl22’ interface option.

The ‘move’ commands are designed for mobile robots that use two motors, one on the left (motor A) and one on the right (motor B). These commands allow the robot to move in a straight line by switching on both motors at exactly the same time.

5.6 Downloading MotorControl22

MotorControl22.rxe is provided as a Lego Mindstorms software file that must be downloaded onto the NXT brick. This is a one-off process, as the program is retained in the NXT memory when it is switched off. You can download MotorControl22 to the NXT brick using the standard LEGO MINDSTORMS NXT 2.0 software or via NeXTTool by following these steps:

First download and unzip MotorControl22 from here.

Method 1 - Using MINDSTORMS NXT 2.0 software

1. Select "Remote Control" from the "Tools" menu to open the "Remote Control" dialog

2. Select "Connections" from the "Remote Control" dialog to open the connection dialog

3. Scan for your NXT and connect via USB or Bluetooth in the usual way

4. Select the "Memory" tab in the connection dialog

5. Select "Download" from the "Memory" tab and use the file dialog to select the MotorControl22.rxe file to download to the NXT

Method 2 - Using NeXTTool by USB cable

(Note the Mindstorms Fantom driver must also have been installed to use NeXTTool, this already occurs by default when the Mindstorms NXT 2.0 software is installed)

1. Download NeXTTool.

2. Unzip NeXTTool into the same folder as the MotorControl22.rxe file.

3. Double click on the 'TransferMotorControlBinaryToNXT.bat' file in that folder.

MotorControl22 was developed by the clever people at RWTH Aachen University and is not altered or changed in any way. The original source code and protocol can be found here.

6.0 Notes on using the Lego EV3 with S2Bot

S2Bot does not currently support WiFi or USB cable connection to EV3, so you must use classic bluetooth.

The S2Bot software does not use the in-built ‘Add Extensions’ feature of Scratch 2, so do not use the ‘Add Extensions’ button. Instead simply open the .sb2 template file provided, as opening this file will automatically configure Scratch correctly and add the extra sensor blocks to the ‘More Blocks’ palette.

In the Offline version select File > Open

In the Online version select File > Upload from my computer

6.1 Bluetooth

Before using the EV3 it must be bluetooth ‘paired’ with your computer. Bluetooth instructions can be found in the appendix at the end of this document. The default EV3 bluetooth pin is 1234, and Bluetooth is switched on via the EV3 ‘Settings>Bluetooth’ menu. We also recommend you set the Settings>Sleep setting to 30 minutes or more.

The correct BT COM port to use within S2Bot should be shown within the properties of the Bluetooth Wizard connection. If two COM ports are shown use the ‘outgoing’ COM port.

Always bluetooth connect the PC to the EV3 before starting S2Bot.

6.2 Recommended Classic Bluetooth USB Adapter (if required)

Kensington
idea: The Bluetooth adapter that stays in to stay on
idée : Comme votre adaptateur Bluetooth, restez branché
Idee: Der Bluetooth-Adapter, der immer in Verbindung bleibt
Bluetooth® 2.1 USB Micro Adapter
Bluetooth

Not all bluetooth adapters are the same, some very cheap models can be quite unreliable and hence frustrating to use.

We use a Kensington branded EDR v2.1 miniature bluetooth USB dongle, based on the Cambridge Silicon radio chipset. This cost about £5 ($US 8) from eBay and works very reliably with Windows, Linux and Mac.

For bluetooth LE devices use the BLED112 instead.

See the appendices at the end of this document for setup instruction.

6.3 EV3 Firmware

S2Bot expects the EV3 to be running the normal default Lego firmware (v1.01H or later). This allows easy swapping between Scratch and the Mindstorms software in schools.

6.4 Using Sensors

The EV3 brick automatically recognises sensors when they are attached and immediately uses them in their 'default' mode. Therefore in many case you do not need to configure the sensors, they will just work as soon as they are plugged into the EV3.

However some sensors also have more than 1 mode (e.g. the NXT temperature sensor has both Celsius and Fahrenheit modes), so in some projects you may wish to change to a different mode.

Therefore when using an advanced sensor you can also instruct Scratch to switch to a different 'mode' if desired. To do this drag out a 'setup sensor' block and configure it as required. Add these blocks at the top of your program after your 'green flag' block, or even simply double click on the block to send the information immediately to the EV3. It is also possible to change the sensor type as the program runs for more complex EV3 sensors (e.g. to read angle and rate from the EV3 giro sensor).

The sensor name displayed in S2Bot is the internal name as reported from the EV3 itself.

Use the checkbox beside the 4 sensor reporter blocks in Scratch to display the current sensor value onto the Stage for testing.

6.5 Controlling motors

For simple applications you can use the motor commands to switch individual motors on and off for various amounts of time. Power can be set between 0 and 100 to adjust the motor speed.

However the EV3 motors normally 'coast' (keep rotating) for a small amount of time after they are switched off. Therefore the monitored precision commands such as 'rotate 90 steps' are better to use for more accurate control, as these carefully count the number of steps moved by using the motor's internal tacho counter.

The 'move' command is designed for mobile robots that use two motors, one on the left (motor A) and one on the right (motor B). These commands allow the robot to move in a straight line by switching on both motors at exactly the same time.

6.6 Controlling LED backlight colour

There is also a LED backlight behind the keyboard switches that can be turned red, green or orange.

6.7 Playing sounds

The EV3 can produce 'beep' noise directly and can also play .rsf sound files that have been already downloaded onto the EV3 using the Lego Mindstorms software.

6.8 Keyboard switches

The keyboard arrow switches are not currently accessible in direct bluetooth communication mode. Therefore they cannot currently be used within Scratch programs – this may be a feature added in the future.

6.9 LCD Screen

The LCD screen (for displaying text messages) is not currently accessible in direct bluetooth communication mode. Therefore it cannot currently be used within Scratch programs – this may be a feature added in the future.

7.0 Notes on using the Orbotix Sphero with S2Bot

The Sphero can be changed colour or moved in various directions at various speeds. The direction is always relative to the current 'heading'. The only way to see the current heading is to switch the blue tail light on, the current heading (0 degrees) is then directly opposite the tail light position.

Note that the Sphero can move very quickly, so a slow speed is recommended. Also remember that you can use the Scratch Stage red 'Stop' button to quickly stop the Sphero if it tries to escape!

The S2Bot software does not use the in-built 'Add Extensions' feature of Scratch 2, so do not use the 'Add Extensions' button. Instead simply open the .sb2 template file provided, as opening this file will automatically configure Scratch correctly and add the extra sensor blocks to the 'More Blocks' palette.

In the Offline version select	File > Open
In the Online version select	File > Upload from my computer

7.1 Classic Bluetooth

Before using the Sphero it must be bluetooth 'paired' with your computer. Classic bluetooth instructions can be found in the appendix at the end of this document. The Sphero does not require a PIN, but simply accept 1234 if your bluetooth pairing software still insists on using a PIN!

The Sphero cannot be paired/used when it is still in the charging cradle. Take out of the cradle and tap twice to wake it up before pairing.

The correct bluetooth port to use within S2Bot should be shown within the properties of the Bluetooth Wizard connection. If two ports are shown use the 'outgoing' COM port.

7.2 Recommended Classic Bluetooth USB Adapter (if required)

Kensington
idea: The Bluetooth adapter that stays in to stay on
idée : Comme votre adaptateur Bluetooth, restez branché
Idee: Der Bluetooth-Adapter, der immer in Verbindung bleibt
Bluetooth® 2.1 USB Micro Adapter
Bluetooth

Not all bluetooth adapters are the same, some very cheap models can be quite unreliable and hence frustrating to use.

We use a Kensington branded EDR v2.1 miniature bluetooth USB dongle, based on the Cambridge Silicon radio chipset. This cost about £5 ($US 8) from eBay and works very reliably with Windows, Linux and Mac.

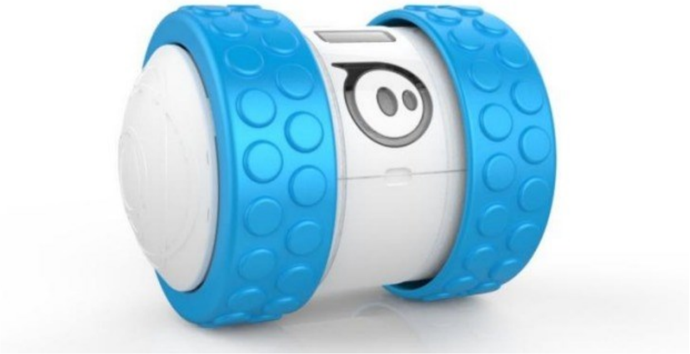
For bluetooth LE devices use the BLED112 instead.

See the appendices at the end of this document for setup instruction.

7.3 Sphero Firmware

S2Bot expects the Sphero to be running the normal default firmware.

8.0 Notes on using the Orbotix Ollie or BB8 with S2Bot



The Ollie and BB8 robots both use an identical bluetooth low energy communication protocol and have the same features, so are regarded as being the 'same' robot via S2Bot.

A BLED112 dongle is **always** required for Windows or Mac use, even if you have bluetooth inside your computer. See further details below.

The Ollie/BB8 can be changed colour or moved in various directions at various speeds. The direction is always relative to the current 'heading'. The only way to see the current heading is to switch the blue tail light on, the current heading (0 degrees) is then directly opposite the tail light position.

Note that the Ollie/BB8 can move very quickly, so a slow speed is recommended. Also remember that you can use the Scratch Stage red 'Stop' button to quickly stop the Ollie/BB8 if it tries to escape!

See section 1.3 for details on how to connect the WeDo 2.0 via a bluetooth low energy connection.

The S2Bot software does not use the in-built 'Add Extensions' feature of Scratch 2, so do not use the 'Add Extensions' button. Instead simply open the .sb2 template file provided, as opening this file will automatically configure Scratch correctly and add the extra sensor blocks to the 'More Blocks' palette.

In the Offline version select	File > Open
In the Online version select	File > Upload from my computer

Note that the robot will not move when the charging cable is connected!

8.1 Windows / Mac

You **must** use an **external** BLED112 ‘BLE to serial’ smart dongle. This is due to the Chrome app development system not currently supporting ‘internal bluetooth LE’ on any platform apart from Chromebooks. The BLED112 smart USB adapter overcomes this app limitation via communicating with the operating system as a ‘virtual RS232 Com Port’ instead of as ‘native BLE’.

However one great advantage of this system is that, as the BLED112 is a smart ‘bluetooth to serial COM port’ adapter, it can also be used on older operating systems such as Windows XP or Windows 7 which do not normally support BLE technology. All versions of Windows from XP up to 10 are therefore supported.

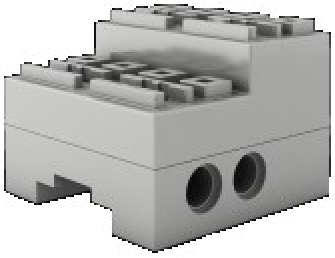
You do not need to manually configure any bluetooth stack or ‘pair’ the bluetooth device. The S2Bot app and BLED112 dongle handle all the pairing and communication automatically.

For more details on purchasing and installing a BLED112 dongle please see:

www.picaxe.com/bled112

Do not try to use a cheaper (non-Smart) bluetooth USB dongle from auction sites such as eBay. These will simply not work with the S2Bot system.

9.0 Notes on using the Vengit SBrick with S2Bot



A BLED112 dongle is **always** required for Windows or Mac use, even if you have bluetooth inside your computer. See further details below.

The SBrick can control up to 4 Lego motors.

See section 1.3 for details on how to connect the WeDo 2.0 via a bluetooth low energy connection.

The S2Bot software does not use the in-built ‘Add Extensions’ feature of Scratch 2, so do not use the ‘Add Extensions’ button. Instead simply open the .sb2 template file provided, as opening this file will automatically configure Scratch correctly and add the extra sensor blocks to the ‘More Blocks’ palette.

In the Offline version select File > Open

In the Online version select File > Upload from my computer

9.1 Windows / Mac

You **must** use an **external** BLED112 ‘BLE to serial’ smart dongle. This is due to the Chrome app development system not currently supporting ‘internal bluetooth LE’ on any platform apart from Chromebooks. The BLED112 smart USB adapter overcomes this app limitation via communicating with the operating system as a ‘virtual RS232 Com Port’ instead of as ‘native BLE’.

However one great advantage of this system is that, as the BLED112 is a smart ‘bluetooth to serial COM port’ adapter, it can also be used on older operating systems such as Windows XP or Windows 7 which do not normally support BLE technology. All versions of Windows from XP up to 10 are therefore supported.

You do not need to manually configure any bluetooth stack or ‘pair’ the bluetooth device. The S2Bot app and BLED112 dongle handle all the pairing and communication automatically.

For more details on purchasing and installing a BLED112 dongle please see:

www.picaxe.com/bled112

Do not try to use a cheaper (non-Smart) bluetooth USB dongle from auction sites such as eBay. These will simply not work with the S2Bot system.

Appendix A – BLED112 Driver Installation

These instructions are for WeDo 2.0, Ollie, BB8, SBrick.
They do not apply to NXT, EV3 and Sphero which uses the kensington dongle instead.

Installing BLED112 Driver for Windows

Windows 7, 8, 10 and later should automatically download and install the BLED112 driver the very first time the BLED112 is inserted via the 'New Hardware Wizard'. For XP / Vista (or if the driver installation failed on 7/8/10) the driver must be installed manually from this link.

When the driver is correctly installed you will see an entry within Device manager in the 'Ports (COM & LPT)' section that states 'Bluegiga Bluetooth Low Energy'. This entry gives you the correct COM ort number to use within S2Bot.

The computer **MUST be restarted** after the driver has been installed.

Other operating systems

No driver installation is required for Mac, Linux or Chromebook.

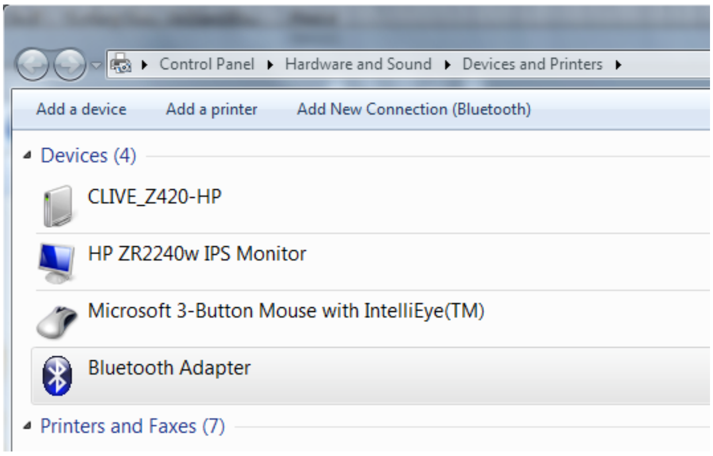
Appendix B – Bluetooth on Windows

These instructions are for NXT, EV3 and Sphero.
They do not apply to WeDo 2.0, Ollie, BB8, SBrick which uses the BLED112 smart dongle instead.

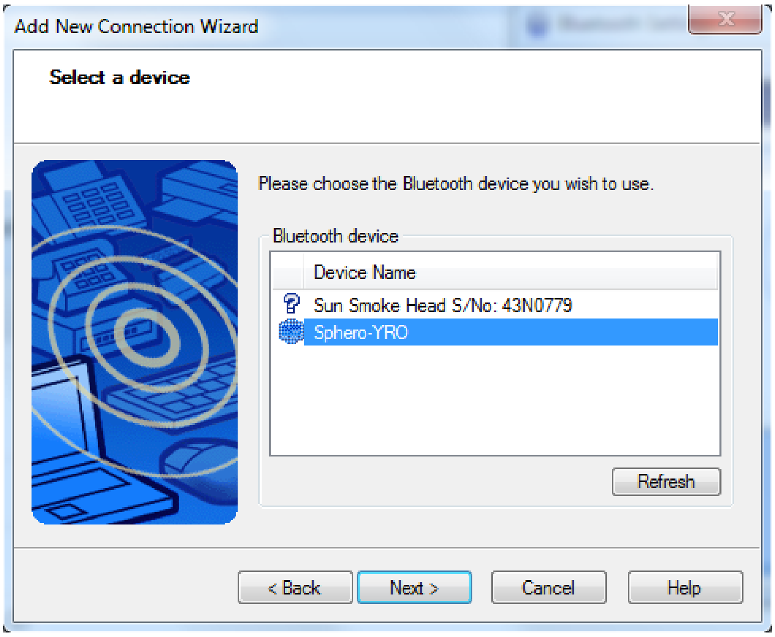
We use a Kensington branded EDR v2.1 miniature bluetooth USB dongle, based on the Cambridge Silicon radio chipset. This was about £5 from eBay, and was supplied with a CD containing the 'Bluetooth Stack for Windows by Toshiba v7.00.05' (which will work on any brand of computer, not just Toshiba!). We used Windows 7 Pro (64-bit) for these screenshots. The appearance of the bluetooth wizard will vary depending on which bluetooth stack software you are using, but the process is generally very similar.

1) Install the bluetooth software (if required) and restart the computer.

2) Insert the bluetooth dongle into the USB port. Either click on the 'bluetooth' icon in your taskbar, or from the Control Panel > Hardware and Sound > Devices and Printers select Bluetooth Adapter and then click 'Add New Connection (Bluetooth).


Control Panel ▸ Hardware and Sound ▸ Devices and Printers ▸
Add a device
Add a printer
Add New Connection (Bluetooth)
Devices (4)
CLIVE_Z420-HP
HP ZR2240w IPS Monitor
Microsoft 3-Button Mouse with IntelliEye(TM)
Bluetooth Adapter
Printers and Faxes (7)

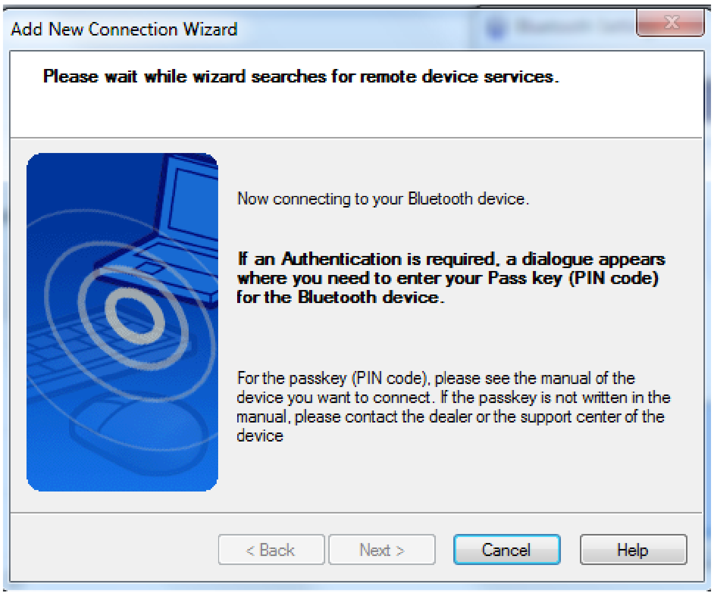
3) Highlight the new bluetooth device required and click Next.


Add New Connection Wizard

Select a device

Please choose the Bluetooth device you wish to use.

4) When prompted empty the pass key (PIN) required on the actual device (the default for NXT/EV3 is 1234). Some devices like Sphero do not use a PIN, so just accept the default value on screen.



Add New Connection Wizard

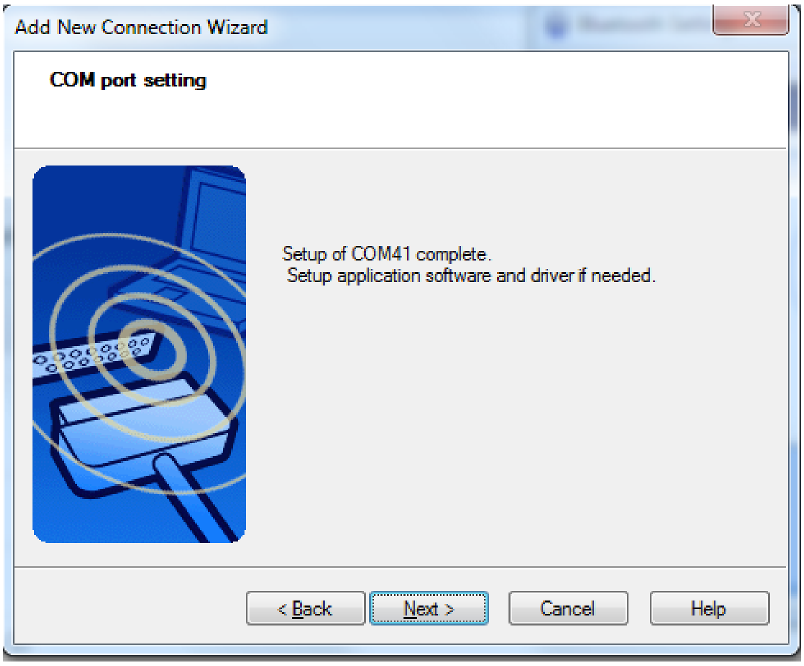
Please wait while wizard searches for remote device services.

Now connecting to your Bluetooth device.

If an Authentication is required, a dialogue appears where you need to enter your Pass key (PIN code) for the Bluetooth device.

For the passkey (PIN code), please see the manual of the device you want to connect. If the passkey is not written in the manual, please contact the dealer or the support center of the device

5) Once pairing is complete the target COM port now reserved for that device will be shown, remember this number for future reference.

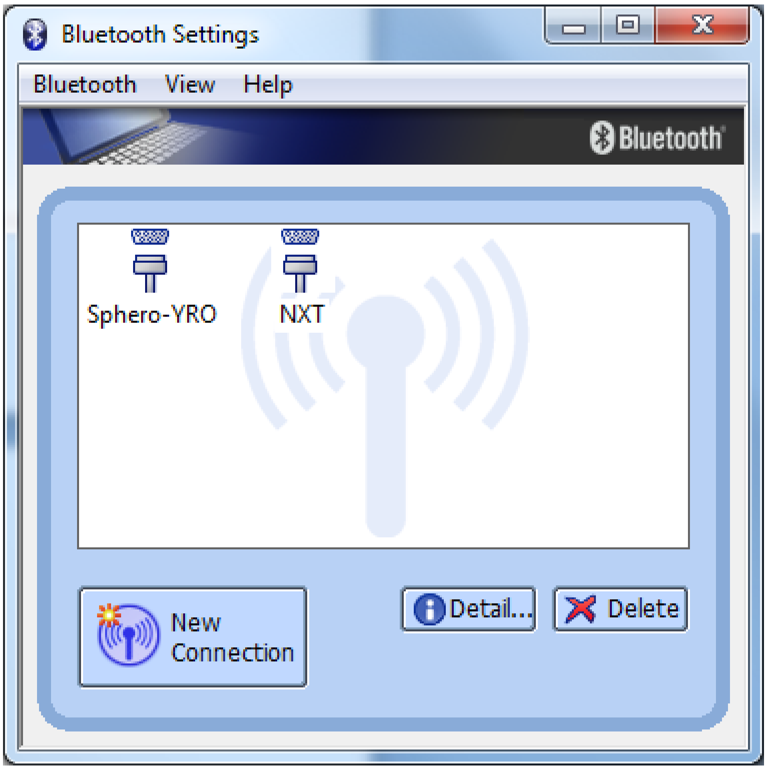


Add New Connection Wizard

COM port setting

Setup of COM41 complete.
Setup application software and driver if needed.

6) The paired devices will now be shown in the bluetooth window (which can be activated from the icon in the task bar). Note that a bluetooth device can be 'paired' but not yet 'connected', but to use with S2Bot it must also be switched on and 'connected'.



When you are ready to use the device simply double click on it's icon within the bluetooth window. It will then connect and the icon will change to show that it is now connected. Click on the 'detail' button to see which COM port number to use within S2Bot if you have forgotten.

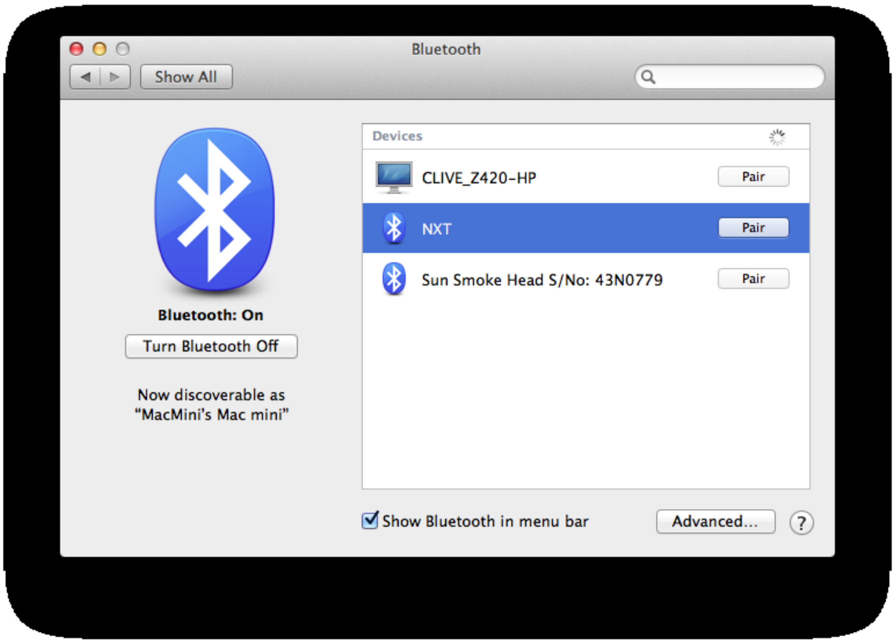
Different bluetooth stack software will vary in appearance from the images shown above, you may need to right click over the icon and select Properties to see the COM port. If two different COM ports are shown for the same device then you should select the 'Outgoing' COM port number.

Appendix C – Bluetooth on Mac

These instructions are for NXT, EV3 and Sphero.
They do not apply to WeDo 2.0, Ollie, BB8, SBrick which uses the BLED112 smart dongle instead.

On OS-X Mavericks and later (10.9 shown in screenshots):

Select Bluetooth from Apple menu>System Preferences:

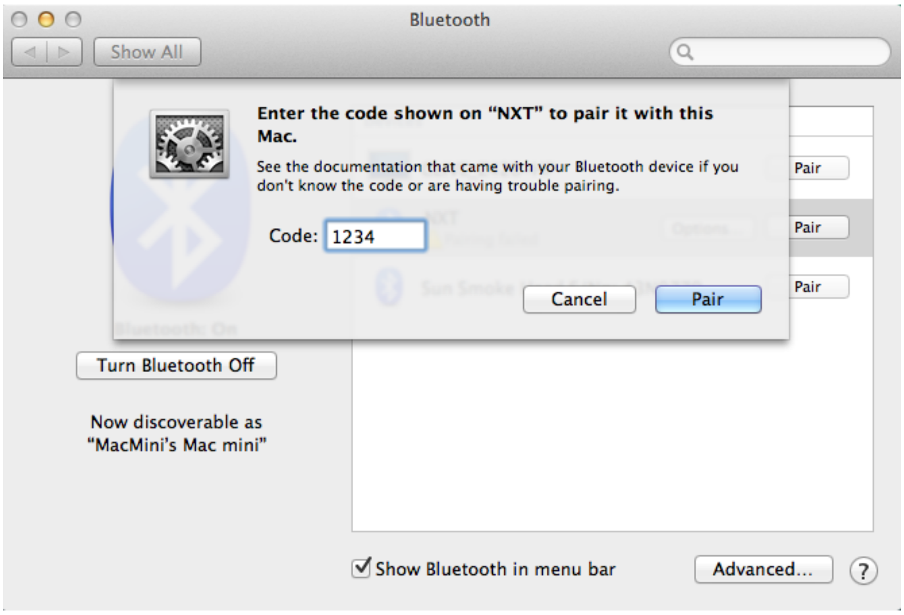


Bluetooth: On

Now discoverable as
"MacMini's Mac mini"

☑ Show Bluetooth in menu bar

Highlight the device required and click 'Pair'. When prompted enter the PIN pass code onto the device (note that Sphero doesn't require a PIN, so just click Pair anyway).



Enter the code shown on "NXT" to pair it with this Mac.

See the documentation that came with your Bluetooth device if you don't know the code or are having trouble pairing.

Code: 1234

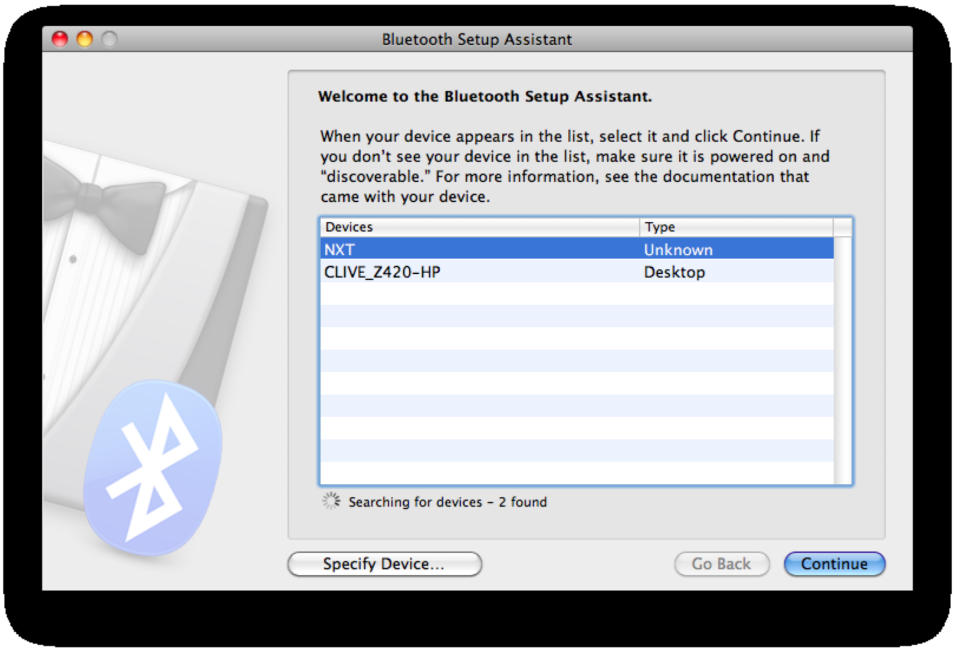
Now discoverable as
"MacMini's Mac mini"

☑ Show Bluetooth in menu bar

That's it! Use the 'Scan' link within S2Bot to find the automatically generated serial port name (e.g. /dev/NXT-DevB)

On earlier OS X Versions (10.8 and earlier, 10.6 shown in screenshots):

Select Bluetooth from Apple menu>System Preferences:



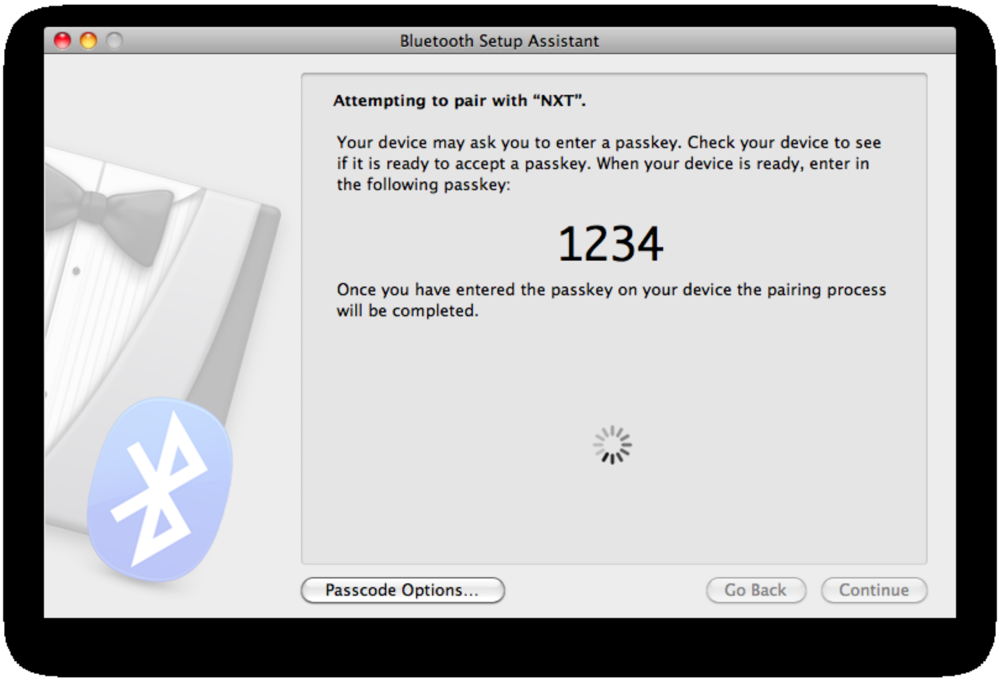
Welcome to the Bluetooth Setup Assistant.

When your device appears in the list, select it and click Continue. If you don't see your device in the list, make sure it is powered on and "discoverable." For more information, see the documentation that came with your device.

Devices	Type
NXT	Unknown
CLIVE_Z420-HP	Desktop

Searching for devices – 2 found

Highlight the device required and click ‘Continue’. When prompted enter the PIN pass code onto the device (note that Sphero doesn’t require a PIN, so just click Continue anyway).



Attempting to pair with "NXT".

Your device may ask you to enter a passkey. Check your device to see if it is ready to accept a passkey. When your device is ready, enter in the following passkey:

1234

Once you have entered the passkey on your device the pairing process will be completed.

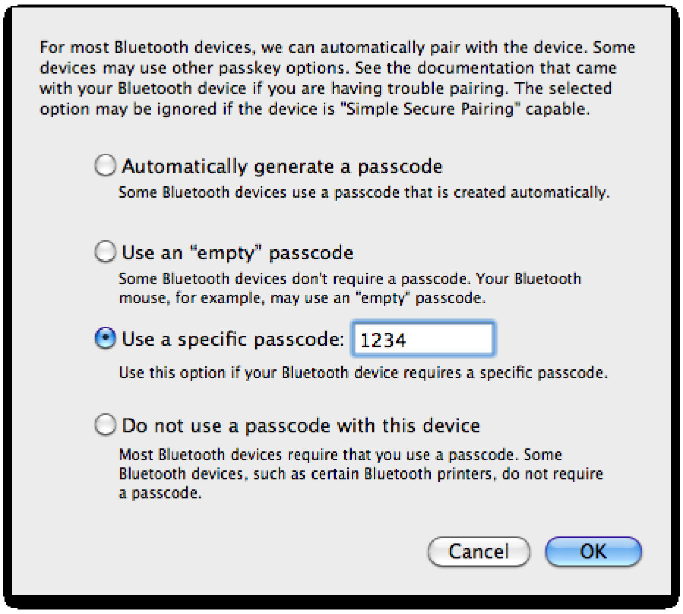
On the first attempt pairing will probably fail.

If this happens click the ‘Passcode Options’ button.

For NXT / EV3 select ‘Use a specific passcode’ and type 1234

For Sphero select ‘Do not use a passcode with this device’

Click OK and then try to pair again by clicking ‘Continue’.



For most Bluetooth devices, we can automatically pair with the device. Some devices may use other passkey options. See the documentation that came with your Bluetooth device if you are having trouble pairing. The selected option may be ignored if the device is "Simple Secure Pairing" capable.

○ Automatically generate a passcode
Some Bluetooth devices use a passcode that is created automatically.

○ Use an “empty” passcode
Some Bluetooth devices don't require a passcode. Your Bluetooth mouse, for example, may use an "empty" passcode.

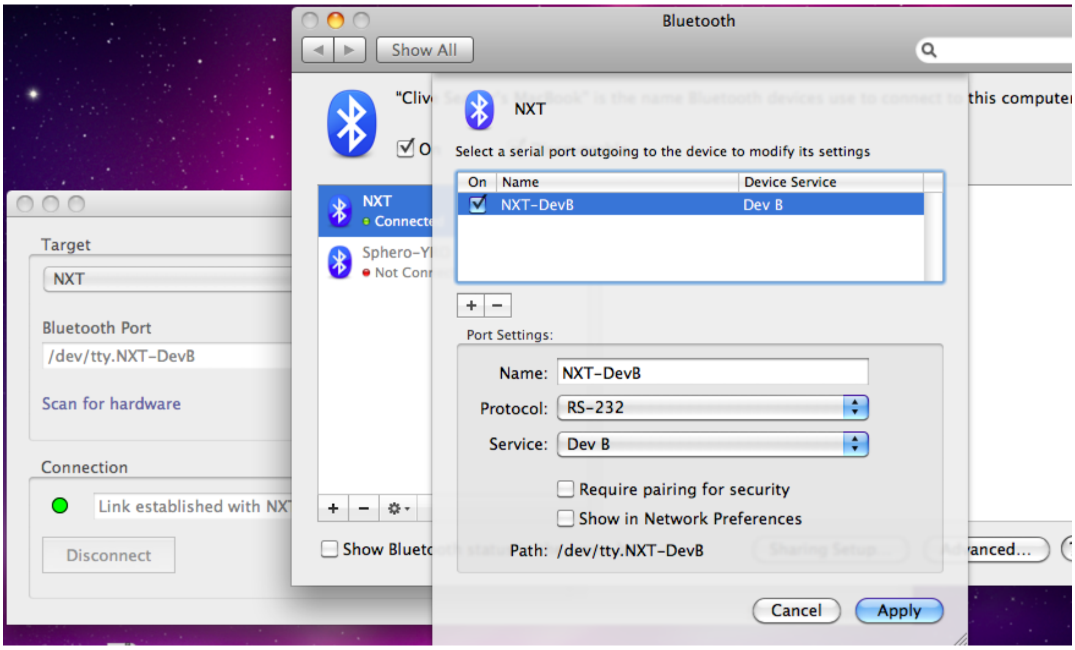
● Use a specific passcode: 1234
Use this option if your Bluetooth device requires a specific passcode.

○ Do not use a passcode with this device
Most Bluetooth devices require that you use a passcode. Some Bluetooth devices, such as certain Bluetooth printers, do not require a passcode.

Once the device is ‘paired’ it is not yet quite ready, as it does not have a serial port allocated.

So highlight the device in the list and click the ‘gear’ bottom left of window and select ‘Edit Serial Ports’. A RS232 protocol serial port will then be allocated to the connection as shown below. The name will be generated automatically e.g. for a NXT it is often /dev/tty.NXT-DevB

Click ‘Apply’ and then the device is then ready to use with S2Bot.



Target: NXT
Bluetooth Port: /dev/tty.NXT-DevB
Connection: Link established with NX

NXT

Select a serial port outgoing to the device to modify its settings

On	Name	Device Service
☑	NXT-DevB	Dev B

Port Settings:

Name: NXT-DevB
Protocol: RS-232
Service: Dev B
☐ Require pairing for security
☐ Show in Network Preferences
Path: /dev/tty.NXT-DevB

}

Appendix D – Bluetooth on Linux (Ubuntu)

These instructions are for NXT, EV3 and Sphero.
They do not apply to WeDo 2.0, Ollie, BB8, SBrick which uses the BLED112 smart dongle instead.

We use a Kensington brand EDR v2.1 miniature bluetooth USB dongle, based on the Cambridge Silicon radio chipset. This works well with Ubuntu 14.04 with both NXT/EV3 and Sphero.

Bluetooth is normally already preinstalled on Ubuntu (we tested with 14.04 LTS) , but on other distros you could use

```
sudo apt-get install bluetooth
```

Make sure the bluetooth device is switched on and then scan for bluetooth devices using

```
sudo hcitool scan
```

Copy the bluetooth address number of the devices found, they are unique to your device and you'll need them later e.g. our Sphero and NXT look like this:

```
00:06:66:4A:3F:FB       Sphero-YRB
00:16:53:0C:F4:6B       NXT
```

Now we need to add these to /etc/bluetooth/rfcomm.conf so that these devices create rfcomm ports for use by S2Bot.

```
sudo gedit  /etc/bluetooth/rfcomm.conf
```

Cut and paste in these sections (here we use both rfcomm0 and rfcomm1 so we can use both a NXT and a Sphero on the same computer, but if you only have 1 device you could simply use rfcomm0). Remember to edit with **YOUR** bluetooth address numbers (found earlier with the hcitool scan)

```
rfcomm0 {
        # Automatically bind the device at startup
        bind yes;
        # Bluetooth address of the device
        device 00:06:66:4A:3F:FB;
        # RFCOMM channel for the connection
        channel 1;
        # Description of the connection
        comment "Sphero-YRB";
}
rfcomm1 {
        # Automatically bind the device at startup
        bind yes;
        # Bluetooth address of the device
        device 00:16:53:0C:F4:6B;
        # RFCOMM channel for the connection
        channel 1;
        # Description of the connection
        comment "NXT";
}
```

Save the file and exit gedit.

Now restart bluetooth

```
sudo /etc/init.d/bluetooth restart
```

Now make sure your rfcomm0 etc. are appearing in /dev/

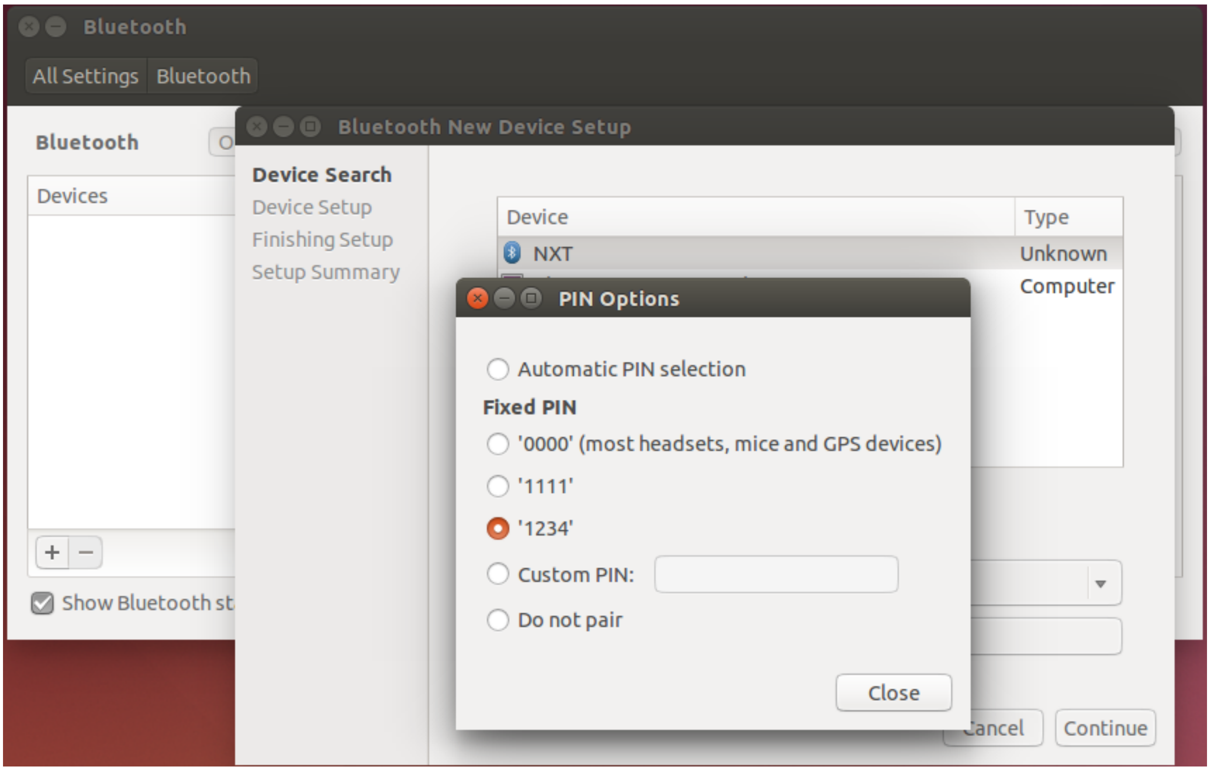
```
ls -l /dev/rfcomm0
```

This will also tell you which group rfcomm0 is in, probably dialout. So make sure you are a member of that group too.

```
sudo adduser your_user_name dialout
sudo reboot
```

Once the computer has restarted you can now use the simpler Ubuntu graphical bluetooth interface to more simply pair the bluetooth device with the computer. (If you had already paired the devices before adding the rfcomm rules then now unpair them and re-pair them again).

Search for ‘Bluetooth’ to find the connection wizard/interface.



Device	Type
NXT	Unknown
	Computer

PIN Options

○ Automatic PIN selection

Fixed PIN

○ '0000' (most headsets, mice and GPS devices)

○ '1111'

● '1234'

○ Custom PIN:

○ Do not pair

Click the ‘+’ button bottom left to add a new device. Select it from the list and click continue.

For Sphero just accept the default PIN that is shown on screen. Pairing can often take 3 or 4 attempts, don’t give up after 1 attempt!

For NXT/EV3 enter the PIN shown on the NXT/EV3 and accept. We did try to force the wizard to use the Lego default of 1234 as shown above, but that just didn’t seem to work (on Ubuntu 14.04), so we gave up and just entered the random 6 digit PIN displayed into the actual NXT instead!

Now you are ready to use with S2Bot – simply use the appropriate /dev/rfcommX as the port to open.

Note you can also easily see the state of any rfcomm port by simply typing

```
rfcomm
```

‘Clean’ means ready to use, ‘closed’ means it is not currently available (e.g. the bluetooth device is switched off).

Appendix E – WeDo 1.0 Installation with Linux (Ubuntu)

1) Wedo 1.0 (USB) is currently only supported by S2Bot on 32–bit versions of Linux (this is because S2Bot is a 32-bit application and there are no 64 bit 'compatibility' files for USB communication). We tested using Ubuntu 14.04 LTS 32-bit.

2) WeDo requires libhid, which is NOT installed by default with Ubuntu 14.04. So you need to install it via

```
sudo apt-get install libhid-dev
```

Note that, strangely, we had to add Germany to our repository sources. list, as the default UK repository on a fresh Ubuntu 14.04 LTS install did not find libhid-dev

3) You also need to add a udev rule to give the S2bot application permission to use WeDo. Assuming WeDo is not being used to launch a nuclear missile it is probably simplest to just give permission to everyone (or your own new group or even the dialout group, which is the group also used by default by the PicoBoard etc). To do this:

Remove the WeDo cable from the computer then type

```
sudo gedit /etc/udev/rules.d/99-legousb.rules
```

Right click in gedit and paste in this text. You only actually need the WeDo line if not also using a RCX/NXT/EV3, but this is what we use for development to support all the various Lego parts.

```
# WeDo everyone rule
SUBSYSTEM=="usb", ATTRS{idVendor}=="0694", ATTRS{idProduct}=="0003",
MODE="0666"

# WeDo dialout rule
# SUBSYSTEM=="usb", ATTRS{idVendor}=="0694", ATTRS{idProduct}=="0003",
GROUP="dialout", MODE="0666"

# RCX IR USB Tower rule
SUBSYSTEM=="usb", ATTRS{idVendor}=="0694", ATTRS{idProduct}=="0001",
SYMLINK+="legoir-%k", GROUP="dialout", MODE="0666"

# NXT brick rule
SUBSYSTEM=="usb", ATTRS{idVendor}=="0694", ATTRS{idProduct}=="0002",
SYMLINK+="legonxt-%k", GROUP="dialout", MODE="0666"

# NXT brick (firmware boot mode) rule
SUBSYSTEM=="usb", ATTRS{idVendor}=="03eb", ATTRS{idProduct}=="6124",
SYMLINK+="legonxt-%k", GROUP="dialout", MODE="0666"

#EV3 brick rule
SUBSYSTEM=="usb", ATTRS{idVendor}=="0694", ATTRS{idProduct}=="0005",
SYMLINK+="legoev3-%k", GROUP="dialout", MODE="0666"
```

Then save and exit gedit.

If required, add yourself to the dialout group and reboot

```
sudo adduser your_user_name dialout
sudo reboot
```

or just restart the udev service instead

```
sudo service udev restart
```

and then reinsert your WeDo back into the computer.

4) If desired, you can test if the WeDo hub is attached and found via

```
sudo lsusb -d 0x0694:0x003
```

Appendix F – PicoBoard Driver Installation

Installing FTDI Driver for Windows

Windows 7, 8 and later should automatically download and install the PicoBoard driver the very first time the Picoboard is inserted via the 'New Hardware Wizard'. For XP / Vista (or if the driver installation failed on 7/8) the driver must be installed manually from this link.

When the driver is correctly installed you will see an entry within Device manager in the 'Ports (COM & LPT)' section that states 'USB Serial Port'. This entry gives you the correct COM port number to use within S2Bot.

As long as you always plug the PicoBoard into the same USB socket on your computer the COM port number will remain the same.

Installing FTDI Driver For Mac OS X (Intel)

OS X Mavericks and later already have the AppleFTDI driver for the PicoBoard installed, so nothing needs to be installed. For older versions of OS X the FTDI driver is available and must be installed manually from this link.

The COM port symbolic link name to use within S2Bot will look like this

```
/dev/tty.usbserial-ABCD1234
```

where ABCD1234 will be a unique number for your particular PicoBoard. Connect your USB cable and then click the 'Scan' link in S2Bot to find this unique number.

Installing Driver for Linux

There are no drivers to install on Linux, so the PicoBoard should automatically appear as device `/dev/ttyUSB0` (or `/dev/ttyS0` for a traditional serial port (very old style ScratchBoard)).

When using USB, you can use ls -l to check which group ttyUSB0 is allocated to

```
ls -l /dev/ttyUSB0
```

The group will probably be dialout, in which case you must make sure that you are also a member of the dialout group (you probably won't be by default). To add yourself to the dialout group:

```
sudo adduser your_user_name dialout
sudo reboot
```