

WirelessThings LaunchPad

User Guide

1. Introduction

The WirelessThings LaunchPad is a free download which comprises of a number of small applications that deal with sharing of a wireless transceiver, the setup of the devices and some example snippets of code which you may find useful in writing your own code. It is written entirely in Python.

2. Requirements

You will as a minimum need to have Python 2.7 and PySerial installed on your machine.

Currently the Launchpad works on a Raspberry Pi, there are some changes needed for running under Windows, these will be available as an update shortly.

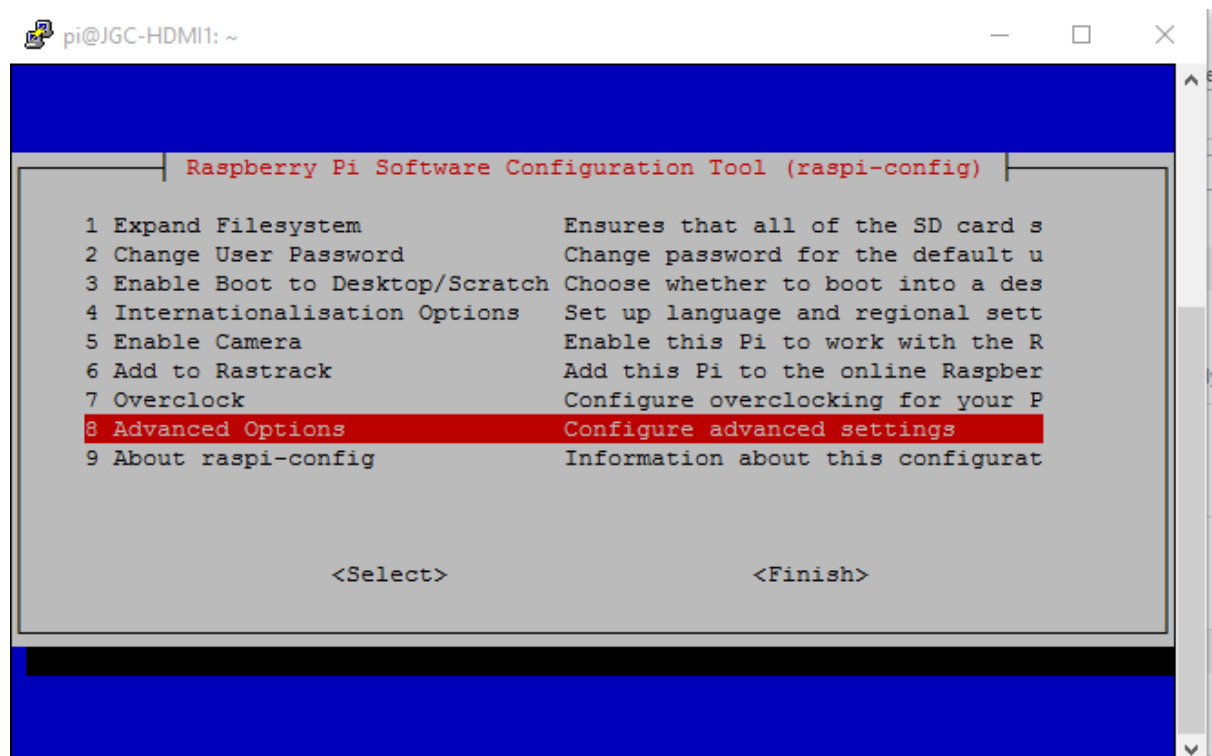
Preparing your Raspberry Pi for use with a Slice of Radio

To use a Raspberry Pi with a slice of Radio if you have a stock Rasbian install you will need to detach the console from the Serial port so that the Slice of Radio can use it.

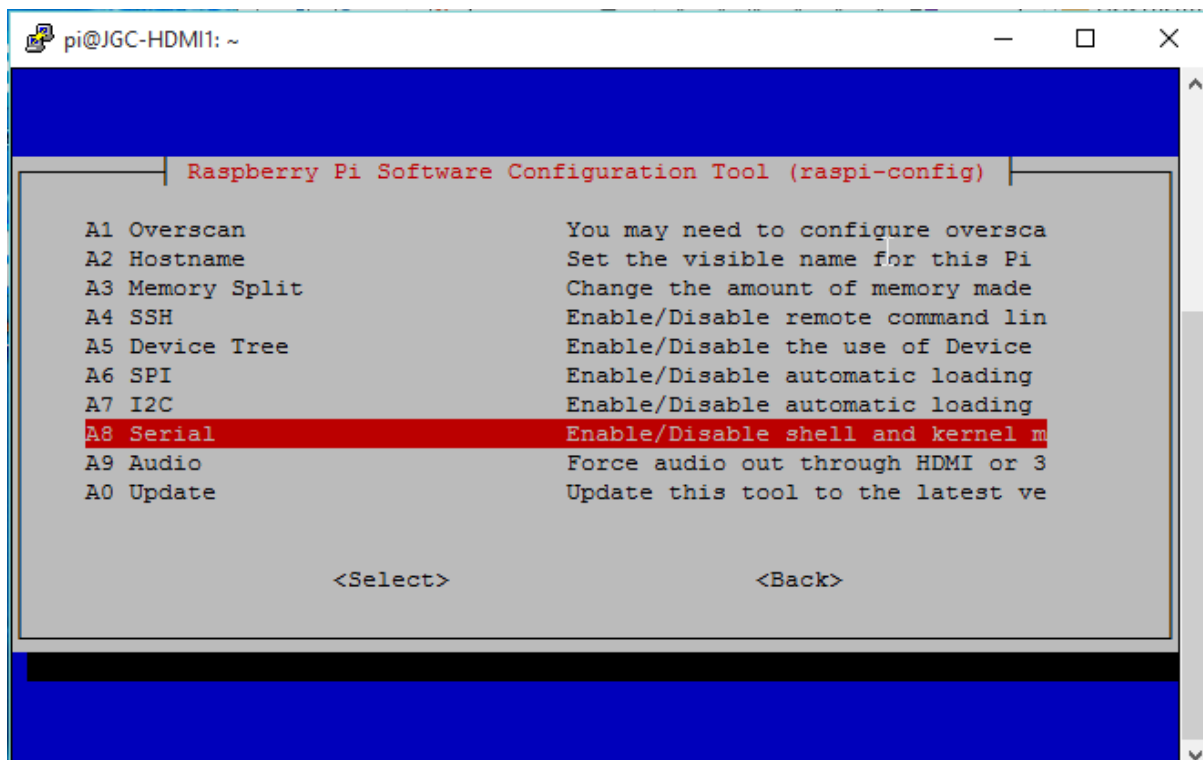
Run the configuration utility from a terminal:

```
sudo raspi-config
```

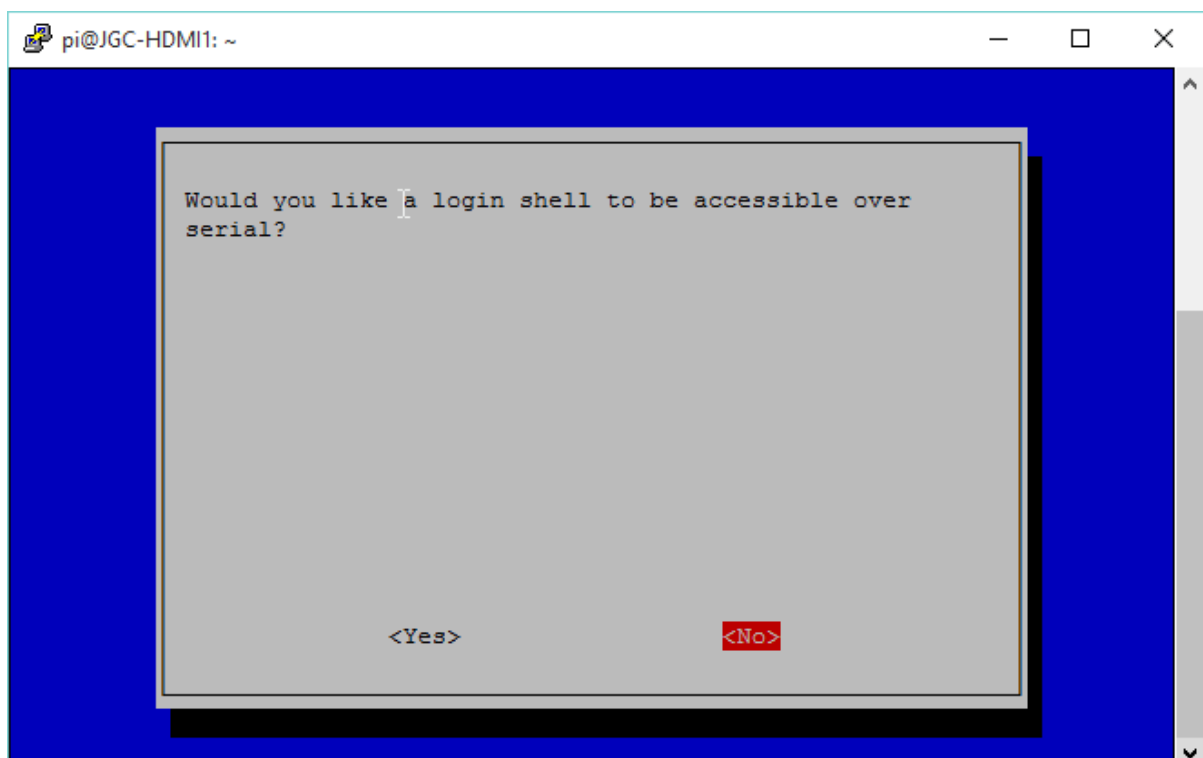
Select "Advanced Options"



Then select Serial



And finally select "No"



Then exit the config utility by selecting OK and Finish.

3. Installation

On the target machine (e.g. Raspberry Pi) extract the files from the zip to a directory of your choice.

It is important not to unzip on a Windows machine and then copy the files over as line endings and file permissions can be incorrectly transferred.

Select a directory to unpack to (change to V0.15 and above, as these versions do not have a containing folder in the zip).

4. Encryption

If you have not already set up encryption then now is the time. This only needs to be done once.

In the LaunchPad directory there is a folder called “Tools”, and in there one called “encryptionSetup”. Navigate to that directory in a terminal window. The following instructions are for a Raspberry Pi running Raspbian, but a similar sequence can be followed on other operating systems.

The python script in the folder called “encryptionSetup.py” will set up a random PanID and encryption key on your device. These will then be copied by the MessageBridge and Configuration Wizard onto any devices that you configure. If your device has already been setup then the tool will report that to you.

The command:

```
python encryptionSetup.py
```

will run the tool using the port /dev/ttyAMA0 and a baud rate of 9600, This is suitable for the Slice of Radio.

If you are using an SRF-Stick or other WirelessThings USB device then you will need to use the following command:

```
python encryptionSetup.py -p /dev/ttyACM0
```

Command line switches used by this tool are

-d	Enable debug output to the console
-p	Specify the port to use Default /dev/ttyAMA0
-b	Specify the baud rate to use Default 9600
-f	Force overwrite setting even if encryption is already setup

Example:

```
pi@jgctest ~/WirelessThings-LaunchPad/Tools/encryptionSetup pi@jgctest
~/WirelessThings-LaunchPad/Tools/encryptionSetup $ python encryptionSetup.py
2015-08-17 10:55:48,292 - This app will attempt to read the current PANID and
encryption setting from the radio on port /dev/ttyAMA0.
2015-08-17 10:55:48,293 - If factory default setting are found we will generate a
new PANID and encryption key to setup your radio network
2015-08-17 10:55:48,295 - Attempting to read the current settings
2015-08-17 10:55:50,411 - Default settings found
2015-08-17 10:55:50,411 - Generating new Settings
2015-08-17 10:55:50,412 - Applying setting to radio
2015-08-17 10:55:54,661 - New setting have been successfully applied
2015-08-17 10:55:54,662 - Your radio network settings are
2015-08-17 10:55:54,663 - PANID: 86A5
2015-08-17 10:55:54,664 - Encryption is: Enabled
2015-08-17 10:55:54,664 - Encryption Key: 97F800D0CBB421BE475DD2A7E8422C36
2015-08-17 10:55:54,665 - Exiting
pi@jgctest ~/WirelessThings-LaunchPad/Tools/encryptionSetup $
```

Example where the encryption setting have already been done:

```
pi@jgctest ~/WirelessThings-LaunchPad/Tools/encryptionSetup pi@jgctest
~/WirelessThings-LaunchPad/Tools/encryptionSetup $ python encryptionSetup.py
2015-08-17 10:57:02,875 - This app will attempt to read the current PANID and
encryption setting from the radio on port /dev/ttyAMA0.
2015-08-17 10:57:02,876 - If factory default setting are found we will generate a
new PANID and encryption key to setup your radio network
2015-08-17 10:57:02,878 - Attempting to read the current settings
2015-08-17 10:57:04,985 - Non default settings found, no changes have been made
2015-08-17 10:57:04,986 - Your radio network settings are
2015-08-17 10:57:04,987 - PANID: 86A5
2015-08-17 10:57:04,988 - Encryption is: Enabled
2015-08-17 10:57:04,990 - Encryption Key: 97F800D0CBB421BE475DD2A7E8422C36
2015-08-17 10:57:04,991 - Exiting
pi@jgctest ~/WirelessThings-LaunchPad/Tools/encryptionSetup $
```

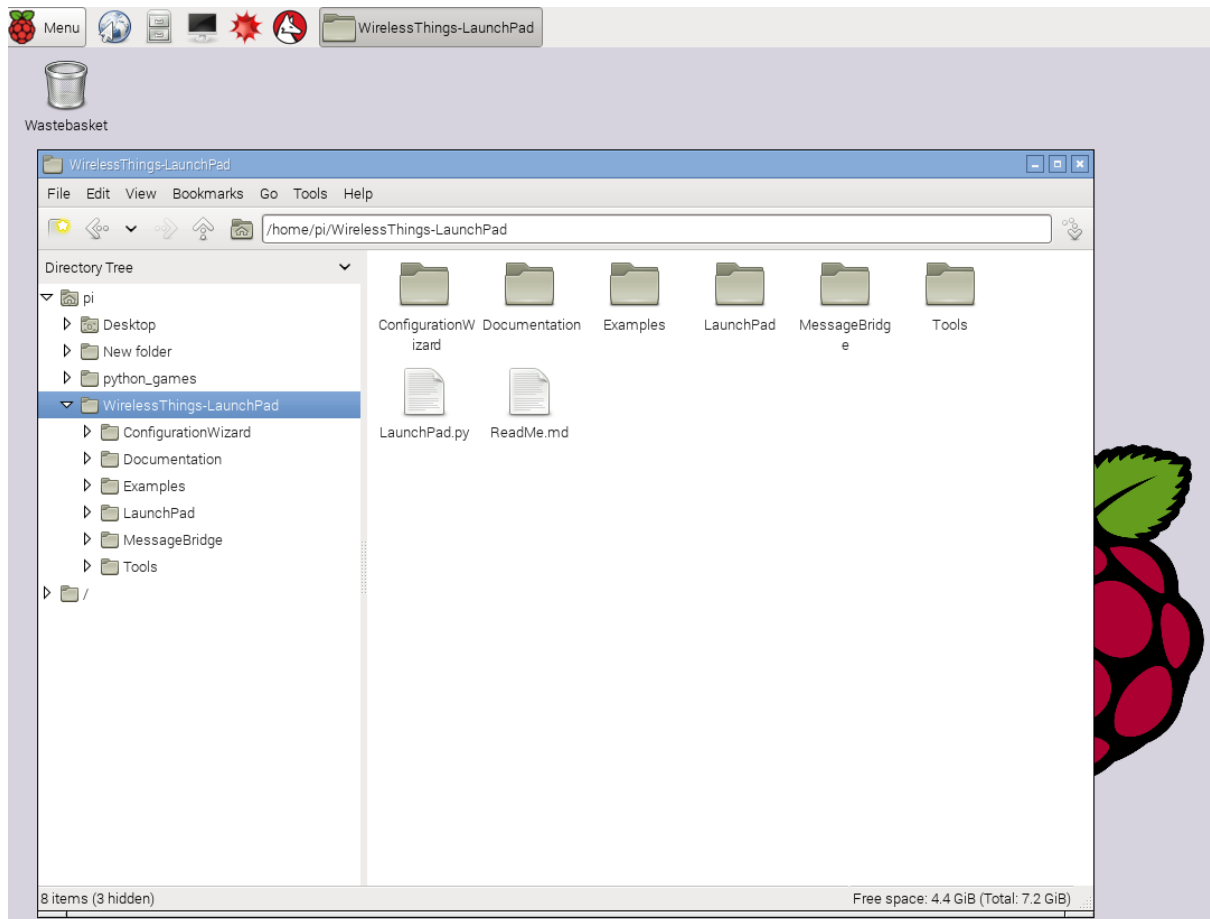
The tool will display the settings to you on completion. It can be useful to note these settings down for later use (e.g. in troubleshooting).

Note: Once encryption has been changed then you will not be able to talk to any devices that have already been configured with different settings. These will need re-configuring via the Configuration Wizard.

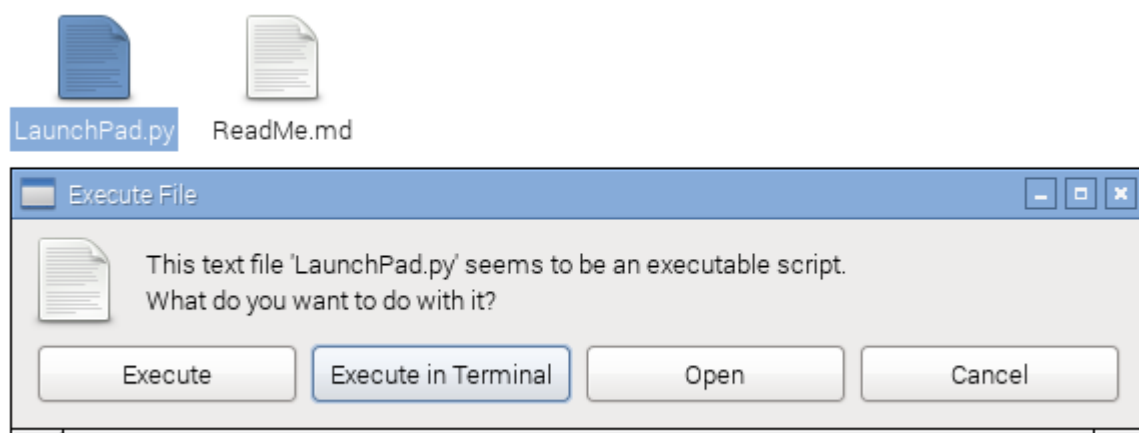
5. The LaunchPad

You will need to be running a graphical interface, this can be started on the Pi by one of the following: the Pi interface (startx), vnc or xrdp.

Open the folder containing the LaunchPad:



Select and open LaunchPad.py – either double click it or right click and select 'open', you should be asked if you want to execute it.

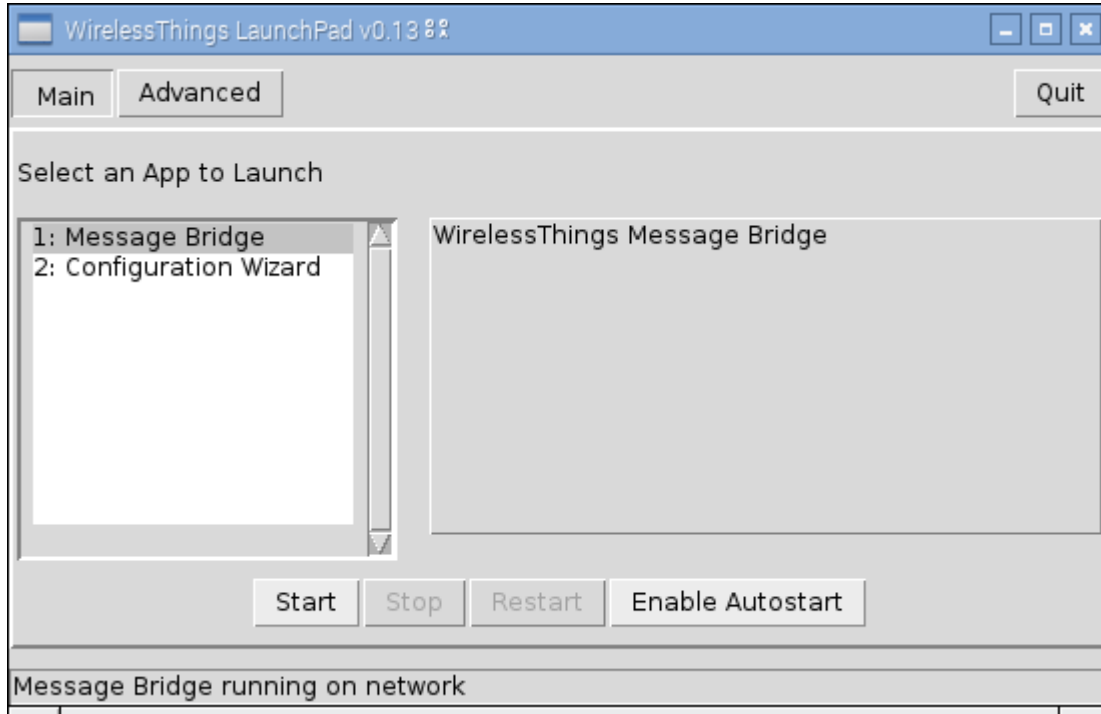


If this does not appear then the file permissions may be incorrect. You can fix the permissions with this command (run from a terminal in the LaunchPad folder):

```
find ./ -name "*.py" -exec chmod +x {} \;
```

If you still have problems then it is probably a good idea to delete the files then download and re-extract the zip on the Raspberry Pi itself.

Then press “Execute” and the LaunchPad will start up and present the following screen:



6. WirelessThings Message Bridge

This small app has two functions, firstly it allows more than a single application to send and receive Language of Things messages with a locally connected USB or serial transceiver such as an SRF Stick, URF, XRF, Voyager+ or other WirelessThings radio device (this requires firmware V53 and above on USB devices and firmware V88 on serial devices to support the message bridge). The second function is to log all incoming and outgoing messages for 7 days.

Configuration File

The configuration file is set up for running on a Raspberry Pi with a Slice of Radio. If you are using a different device then you will need to edit this file.

The configuration file for the MessageBridge is in the folder MessageBridge. The file is called "MessageBridge.cfg". If this file does not exist then it will be created on the first run from MessageBridge_defaults.cfg.

A couple of settings that you may wish to change. These settings are in the file section "[Serial]". To change a setting scroll down the file to find the entry you want to change and then replace the text.

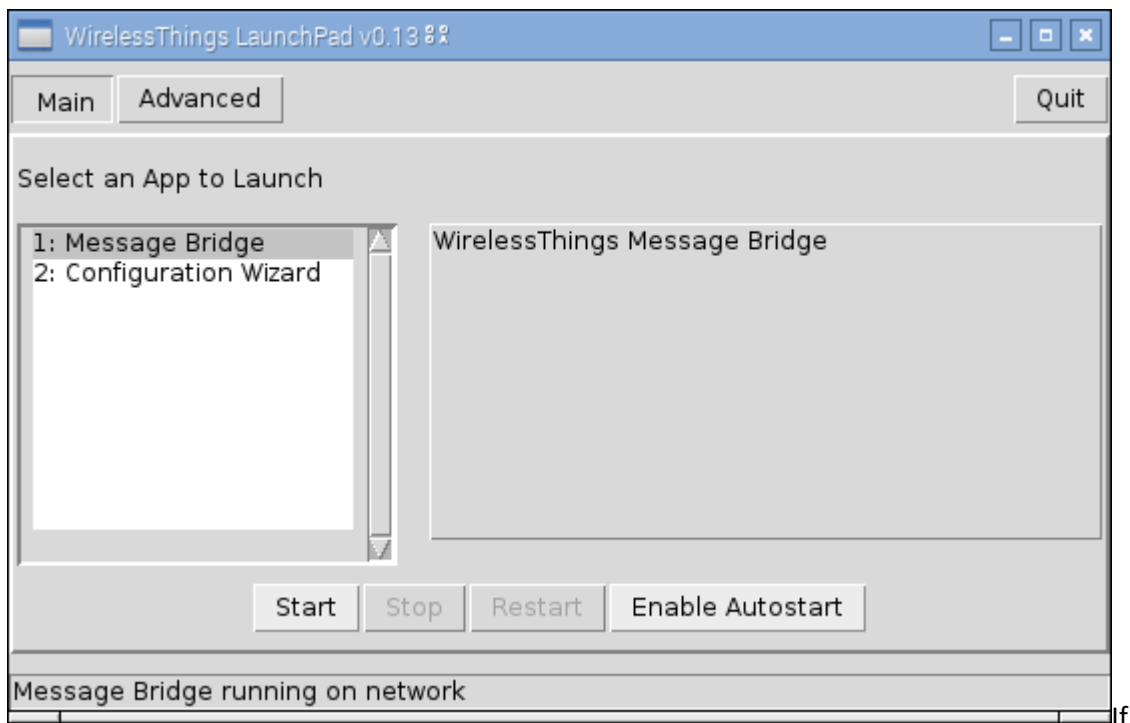
e.g. for a setting "baudrate" look for the line then begins with "baudrate = " and change the text after the "=" to the new setting value.

1. port: The serial port to use. Different devices (e.g.SRF Stick) may use a different serial port. The default Serial port is "/dev/ttyAMA0". For a WirelessThings USB based device this would be "/dev/ttyACM0", whilst under Windows it would be of the form "COM4" where the number reflects the COM port Windows assigned to the device.
i.e. port = /dev/ttyAMA0
2. network: The name of the MessageBridge. If you have more than one MessageBridge running on your local network then you will need to identify them individually. The default name is "Serial".
i.e. network = JohnsSerial

A full description of the configuration settings can be found at the end of this document.

Running the Message Bridge

When the LaunchPad runs you should see the following screen:



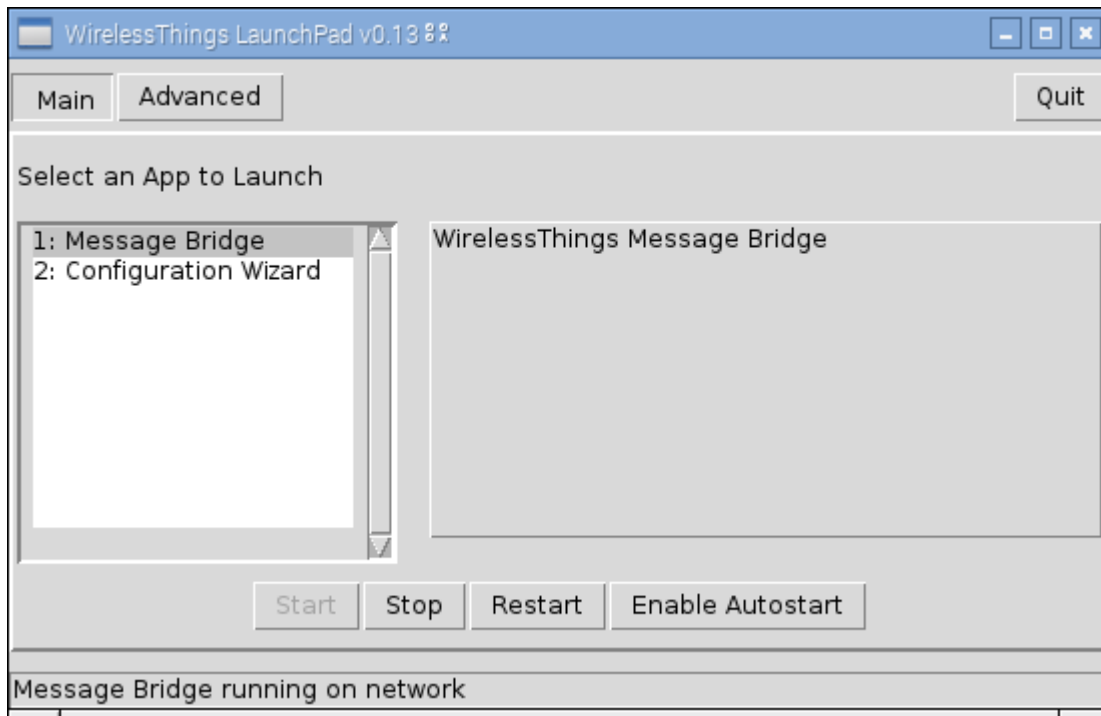
"MessageBridge" is not selected then you will need to select it.

You only need one MessageBridge running on a local network. If a MessageBridge is found then the message at the bottom will say so after a short while.

To run the MessageBridge press "Start".

Note: If you wish to set up this Raspberry Pi to always run the MessageBridge then you could instead press "Enable Autostart", in which case the MessageBridge will be installed into the system and will start whenever the Raspberry Pi is started.

Once the MessageBridge is running then the screen will change to show that:



The Message Bridge will remain running even if you exit the LaunchPad.

CSV Log files

The log files are saved in CSV format and can be imported into a spreadsheet or database. The time and the radio message (split into ID and Data) are the three fields recorded. The location of the log files is "WirelessThings-LaunchPad/MessageBridge/CSVLogs".

Example:

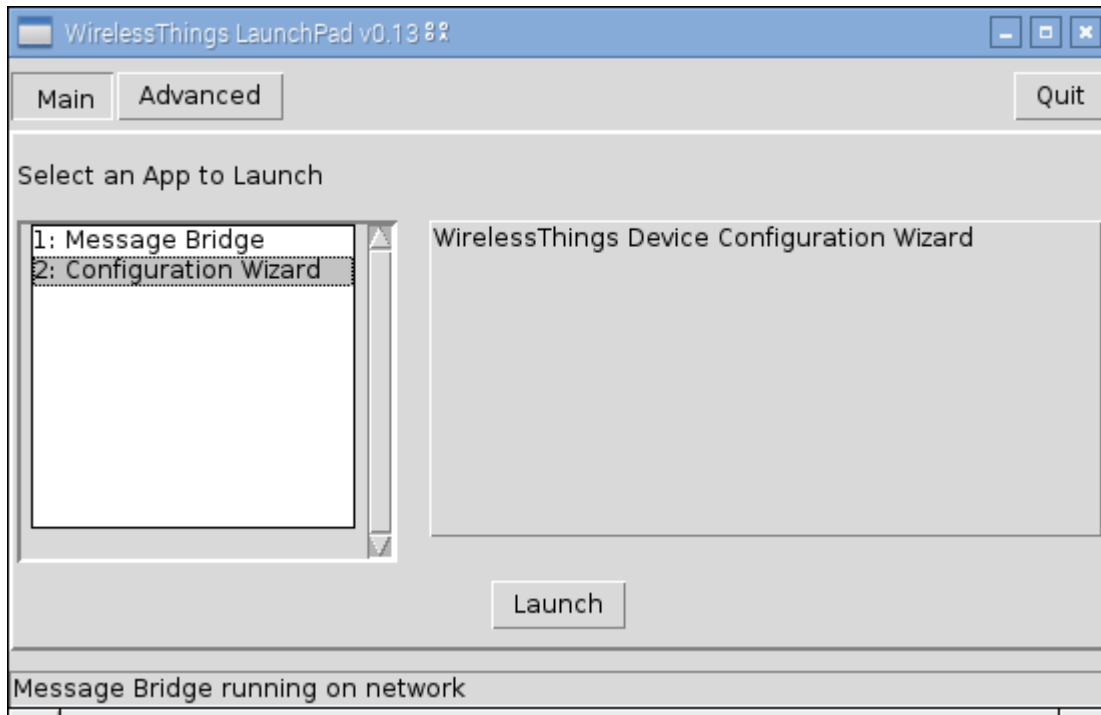
```
17 Aug 2015 08:32:14 +0000,EB,TEMP023.6
17 Aug 2015 08:32:16 +0000,JH,TEMP1060
17 Aug 2015 08:32:16 +0000,JH,TEMQ0071
17 Aug 2015 08:32:17 +0000,JZ,TEMP23.0
17 Aug 2015 08:32:18 +0000,AE,RHUM39.5
17 Aug 2015 08:32:18 +0000,AE,TEMP23.7
17 Aug 2015 08:32:19 +0000,JJ,TEMP025.9
17 Aug 2015 08:32:20 +0000,AA,VAL866259
17 Aug 2015 08:32:21 +0000,JW,ANA1056
```

7. WirelessThings Device Configuration Wizard

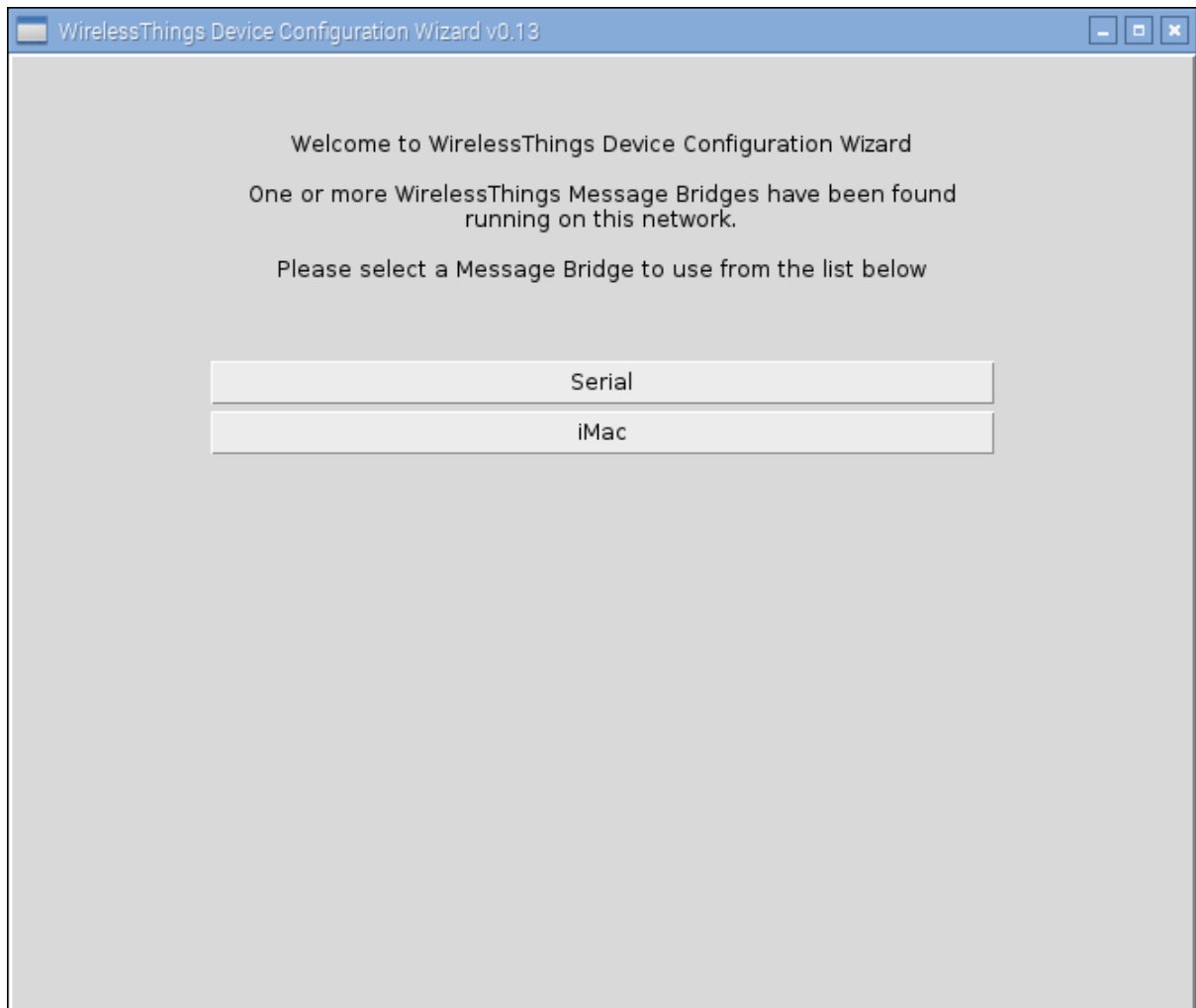
This step by step app allows you to setup an out of the box sensor or reconfigure an existing deployed device.

You need a MessageBridge running on your local network in order to run the Configuration Wizard.

When the LaunchPad runs select “Configuration Wizard” and you should see the following screen:

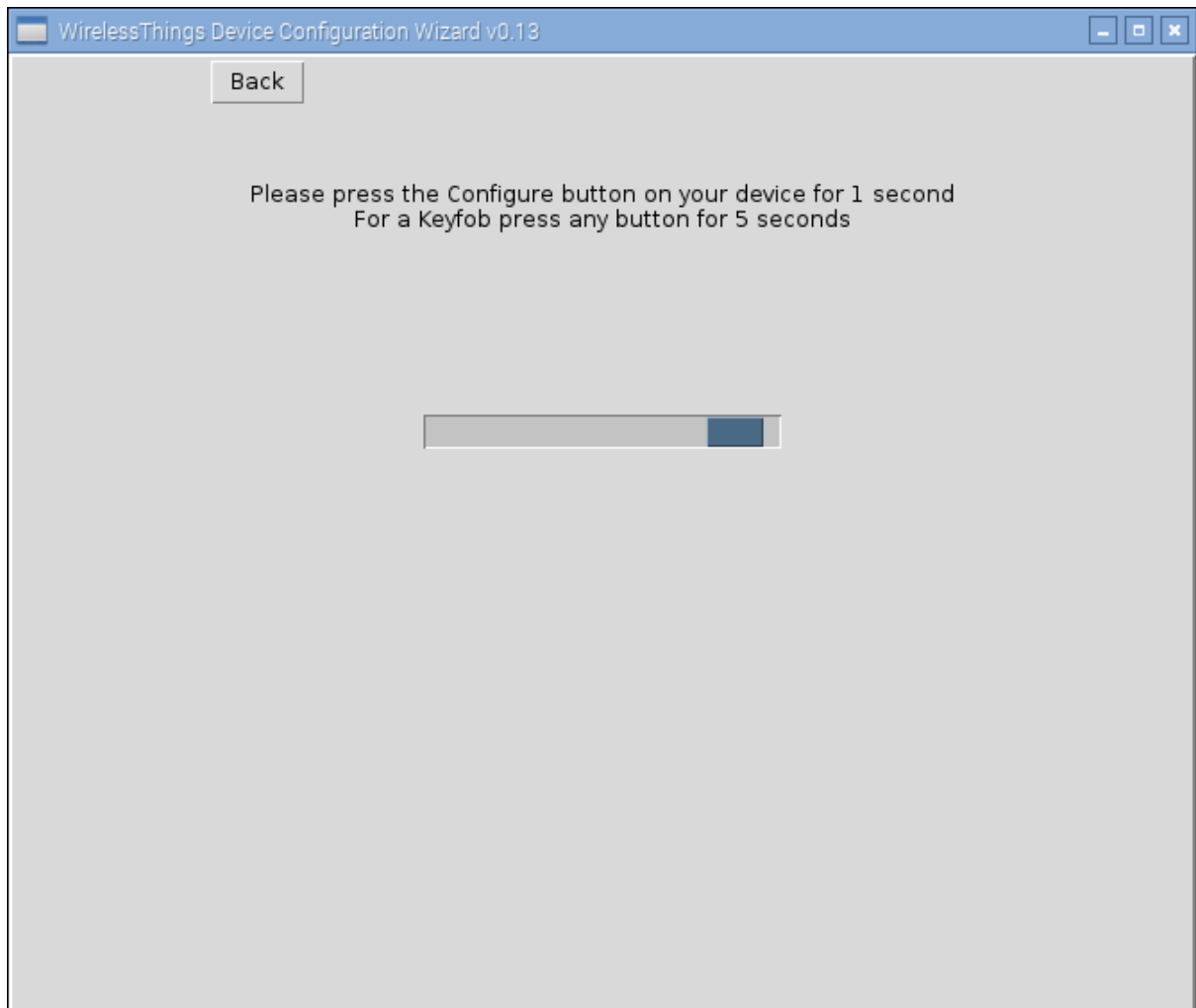


Press “Launch”



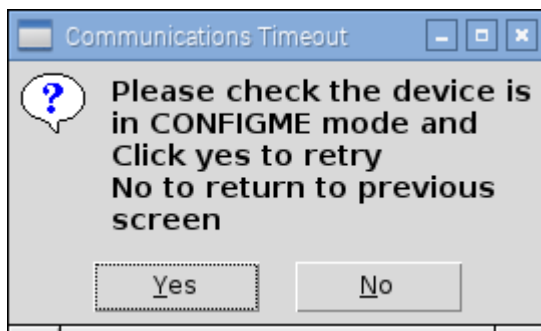
The Configuration Wizard will list the MessageBridges that it has detected running on your network. The default name is "Serial" and we will select that. Usually it will be the only one shown.

Click the MessageBridge that you wish to use:



The Configuration Wizard is now searching for your device. Press the "Configure button" on your device for 1 second to trigger Configuration Mode. – The Configure LED on your device should now flash briefly, then once every 5 seconds. The device will exit configuration mode after 1 minute if it has not heard from the Configuration Wizard.

If the Configuration Wizard did not see your device then you will see the following message:



Press "Yes" and press the "Configure" button on the device again. Make sure that you see the Configure LED flash.

When your device has been seen by the Configuration wizard the screen message will change to "Communicating with device" while the current device settings are downloaded.

Once the settings are downloaded then you will be presented with a screen specific to the device type that you are configuring. The following screen shot is for the thermistor temperature sensor.

The screenshot shows a window titled "WirelessThings Device Configuration Wizard v0.13". At the top, there is a "Back" button, the device name "Temperature Sensor SB-CA", and the RSSI value "-45". The main content area contains the following text: "This Temperature Sensor device sends periodic readings in degrees centigrade. It is accurate to +/- 1 degree between -10 and +40 degrees centigrade" (with an information icon 'i' to the right), "This is a new device. Network setting will be automatically set to match your hub", "Device ID: AA" (with a "Change" button and an information icon 'i'), and "A new ID has been automatically assigned, to override please click change". Below this is a "Reading Interval:" label followed by a slider bar set to "1 Minutes" (with an information icon 'i'). A confirmation message "1 Minute reading interval." is displayed below the slider. At the bottom, there are three buttons: "Apply settings to device", "Reset to defaults", and "Advanced config".

The "Device ID" will be allocated automatically by the Configuration Wizard. This ID identifies the device messages and needs to be unique within your WirelessThings network. You may change this if you wish.

When setting the reporting time consider the thing you are measuring. Room temperature, for instance, does not change that rapidly and a reporting time of 15 or 30 minutes is quite fast enough. A shorter reporting time will mean a reduced battery life.

If you see the option to "Update network settings" then the Wizard has found that the PanID and Encryption settings on your network do not match the local settings. You should normally select the checkbox, if it is not already checked, to update the settings.

Pressing "Apply settings to device" will write the currently selected settings to the device.

"Reset to defaults" will configure your device to its factory settings allowing you to configure again from the start.

Advanced Configuration

The image shows a software window titled "WirelessThings Device Configuration Wizard v0.13". At the top, it displays "Temperature Sensor SB-CA" and "RSSI: -45". Below this is a "Back" button and the instruction "Select your device config options". The window is divided into two main sections: "Generic Commands" and "Device specific options".

Generic Commands:

- Device ID:** CHDEVID
- Pan ID:** PANID
- Retries for Announcements:** RETRIES

Cyclic Commands:

- Sleep Interval:** INTVL
- Battery Wake Count:** WAKEC
- Enable Cyclic Sleep:** CYCLE ☒

At the bottom center is a button labeled "Encryption Settings".

Device ID – this can also be changed from the Advanced Configuration page

Pan ID – this can be used to separate different WirelessThings networks that are within wireless range of each other, a hexadecimal number between 0000 and EFFF can be given here. **You should not use this unless you know what you are doing as it will only set the Pan ID on the device and not at the MessageBridge end.**

Retries for Announcements: Some announcements (e.g. button pressed or door opened) are sent multiple times to ensure that the message gets through. You may wish to lower this to reduce the network traffic.

Cyclic commands: This give a finer grained way of controlling the reporting time.

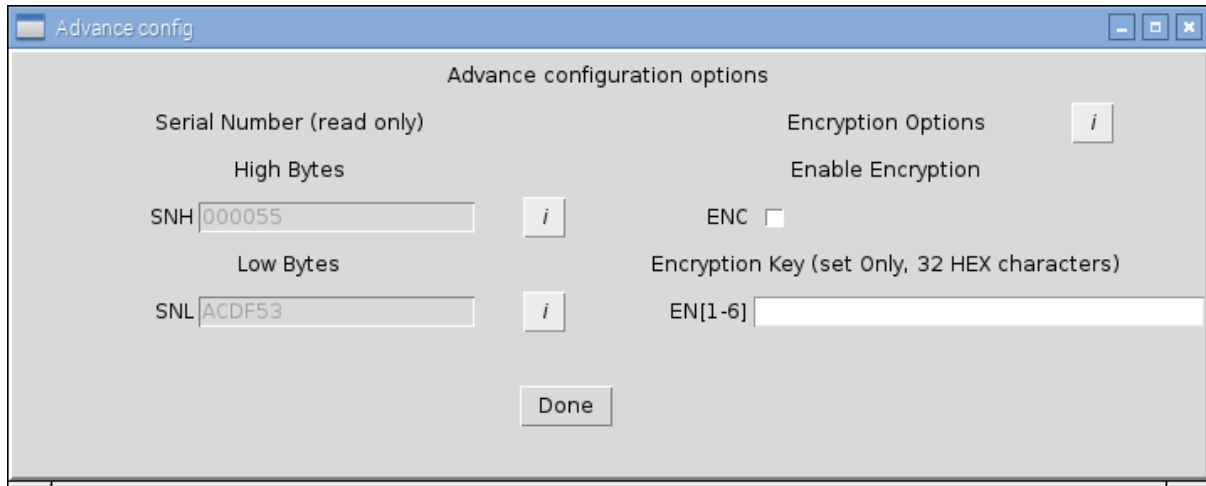
Sleep Interval – this is in the form 999M three digits followed by a letter. The letters are

- S – Seconds, range 000 - 999
- M – Minutes, range 000 - 999
- H – Hours, range 000 - 999
- D – Days, range 000 - 049

Note: a value of 000 switches the sleep interval off.

The Battery reporting interval, every so many readings the device will wake up and send the battery level. It will then wait a short period in case there are any commands for it and then go back to sleep. The default for this is every 10 readings, but this may be changed in order to save on battery consumption. A setting of 0 switches this functionality off completely. Maximum is 255.

Encryption Settings



This screen displays the Serial Number of the device and also allows you to set the encryption settings for the device. **You should not use this unless you know what you are doing as it will only set an encryption key on the device and not at the MessageBridge end.**

The encryption key is specified as 32 hexadecimal characters

e.g. ABCDEF0123456789ABCDEF0123456789

8. Software Examples

Please refer to the readme.md file in the Examples folder.

9. Downloading the WirelessThings LaunchPad

Download the zip file and extract to a folder of your choice, within this is an icon called LaunchPad.py, running this will start the main application, once loaded, from a simple menu you can select different applications to run.

<https://www.wirelessthings.net/launchpad>

10. Updating the WirelessThings LaunchPad

When the LaunchPad is run it will check for a later version and if one is found you will be prompted to upgrade. This will not work for V0.14 – see below.

Updating from V0.14 to V0.15 or later

This is not automatic as several things have changed that prevent the automatic upgrade from running.

- 1) Download the new zip file from <https://www.wirelessthings.net/launchpad>
- 2) Unzip it into a directory of your choice.
- 3) Stop any running message bridge, and disable Autostart if you enabled it.

- 4) If you have modified the message bridge configuration then copy WirelessThings-LaunchPad/MessageBridge/MessageBridge.cfg to the new folder/MessageBridge/MessageBridge.cfg
- 5) You can now run LaunchPad from the new directory, the old WirelessThings-LaunchPad directory can be deleted when you are satisfied that all is running correctly. Note: you may want to preserve the CSV logs (WirelessThings-LaunchPad/MessageBridge/CSVLogs/*.csv) if you are using them.

V0.15 changes

- **BREAKING CHANGE:** LaunchPad Auto update process has had some heavy rework so the update from 0.14 to 0.15 must be done as a full reinstall
- MessageBridge: New User and default config file setup
- LaunchPad: Catch Ctrl+C from console
- LaunchPad: Check the screen geometry to always use the main monitor to open the application
- LaunchPad: Add new post update routine
- ConfigurationWizard & MessageBridge: MessageBridge Name Clash (two or more MessageBridges have the same network identifier) shows a message to user and disables the button
- MessageBridge: Improve the ATlh1 tests
- LaunchPad: make UPD receive buffer larger

11. MessageBridge configuration options

Section	Option	Default	Description
[Debug]	console_debug	False	Enable / disable console debug
	console_level	DEBUG	Log Level for the console output (overridden by the --log command line option if given. Options are {DEBUG, INFO, WARNING, ERROR, CRITICAL}).
	file_debug	INFO	Enable / disable file logging debug. It is NOT recommended to use file debugging on the Raspberry Pi SD card for extended periods of time.
	log_file	./MessageBridge.log	Which file should we log to (file path).
	file_level		Log Level for the file output (overridden by the --log command line option if given. Options are {DEBUG, INFO, WARNING, ERROR, CRITICAL}).
[CSVLog]	csv_log	True	Should CSV logging be enabled {True, False}.
	csv_log_level	INFO	Options are {DEBUG, INFO, WARNING, ERROR, CRITICAL}.
	directory	CSVLogs/	Directory where the CSV

			Logs will be stored.
	csv_file_name	CSV_MessageBridge.csv	Which file should we log to (file path)
	days_to_keep	7	How many days the logger should keep the logs history
[Serial]	network	Serial	Network is the name used in the JSON packet it identify the interface which message are going to or coming from
	baudrate	9600	baud rate for the connected radio.
	port	/dev/ttyAMA0	Serial port on which your radio can be found. For Linux this will be a path like /dev/ttyAMA0 For Windows this will be the name eg. COM1, you may need to use \\.\COM8 on some machines. For Mac OSX this will be a path like /dev/tty.usbmodem000001
[UDP]	send_port	50140	UDP port the Message Bridge sends JSON out on.
	listen_port	50141	UDP port the Message Bridge listens to for incoming JSON
[DCR]	dcr_enable	True	Enable processing of 'DeviceConfigurationRequest' type JSON messages, if set to 'False' DCR messages will be ignored by the Message Bridge.
	timeout	60	Time out in seconds for a DCR if one not specified in the JSON.
	single_query_retry_count	3	Number of time to ask a single toQuery command before giving up.