

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 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. You'll need to have that already installed and PySerial as a minimum

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 issues with running under Windows, these will be corrected shortly.

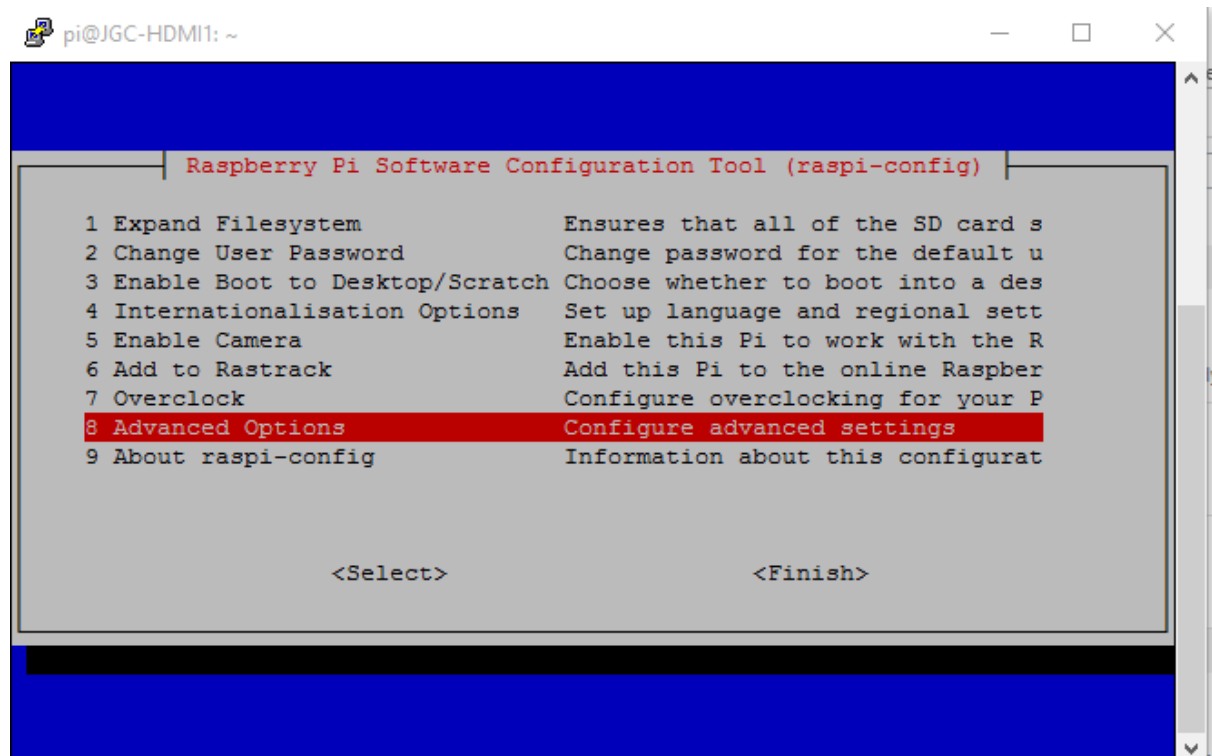
Preparing your Raspberry Pi for use with a Slice of Radio

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

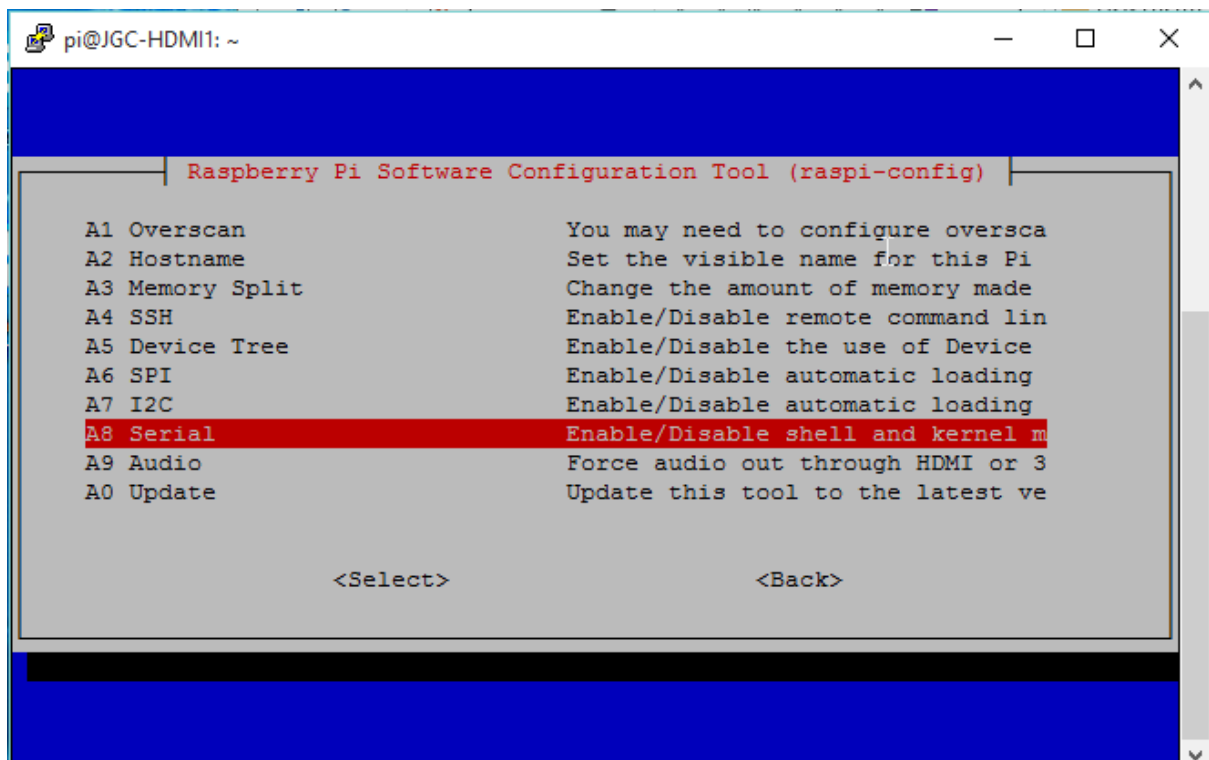
Run the configuration utility from a terminal:

```
sudo raspi-config
```

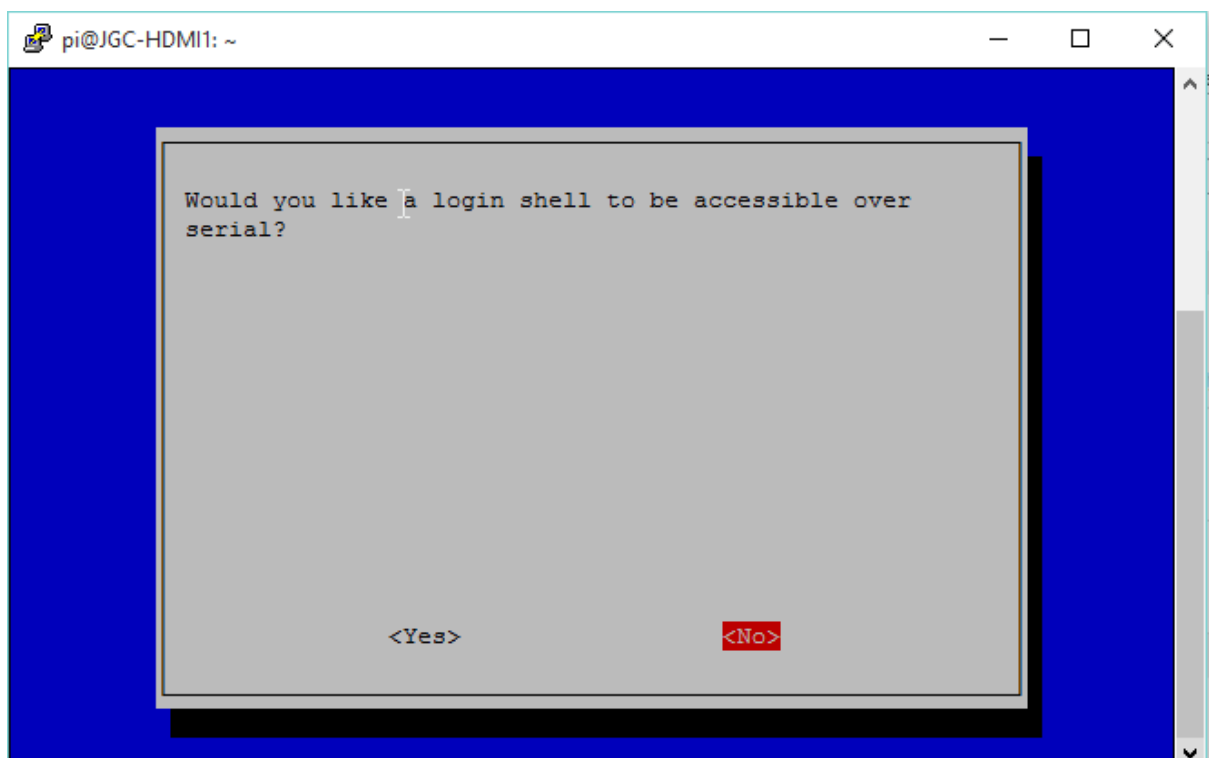
Select "Advanced Options"



Then select Serial



And finally select "No"



Then exit the config utility by selecting OK and Finish.

3. Installation

On the target machine (Raspberry Pi) extract the files from the zip.

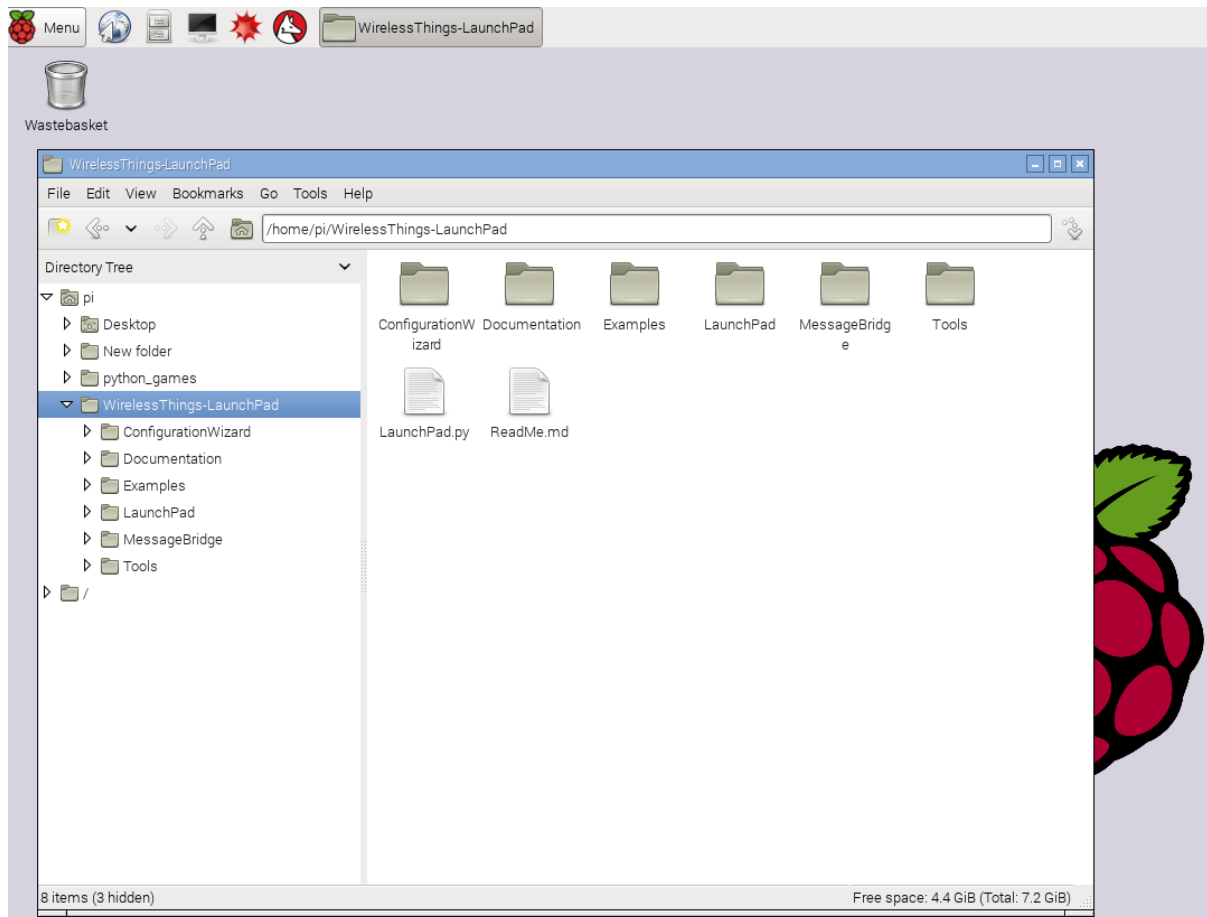
It is important not to unzip on a Windows machine and then copy the files over as line endings and file permissions can be incorrectly transferred.

The extracted files will be in a directory called “WirelessThings-LaunchPad”.

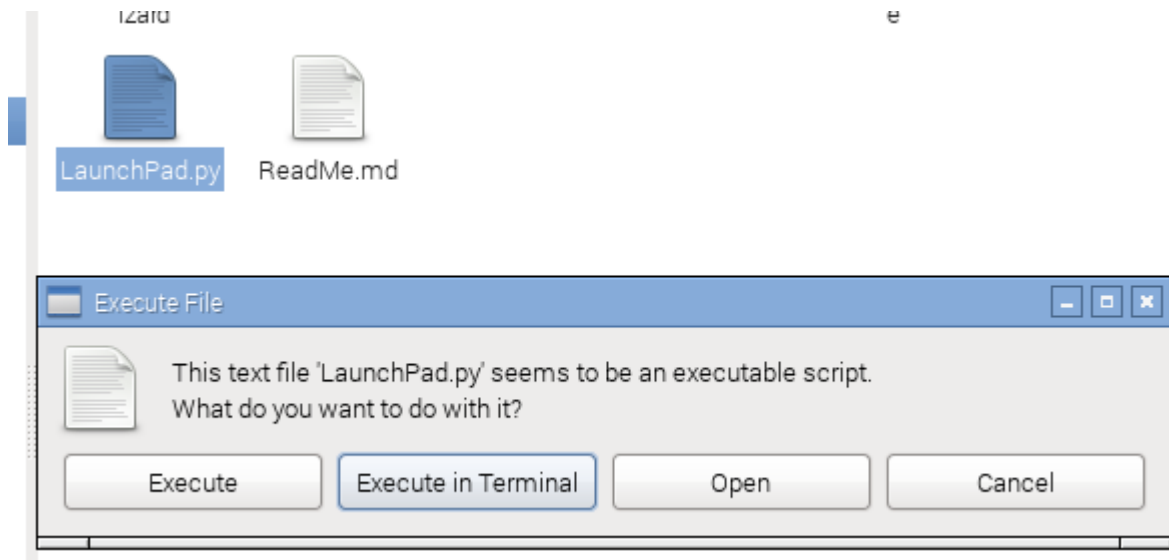
4. The LaunchPad

You will need to be running a graphical interface, this can be though 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 WirelessThings-LaunchPad folder):

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

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

5. WirelessThings Message Bridge

This small app has two functions, firstly it allows more than a single application to send and receive messages to a locally connected USB or serial transceiver such as an SRF Stick, URF, XRF, Voyager+ or other WirelessThings radio device. (requires firmware x and above on USB devices and firmware x on serial devices to support the message bridge). The second function is to log all incoming and outgoing messages for 7 days. The log files are saved in CSV format so can be imported into a spreadsheet or database. The time and the radio message are the two fields recorded.

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"

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.

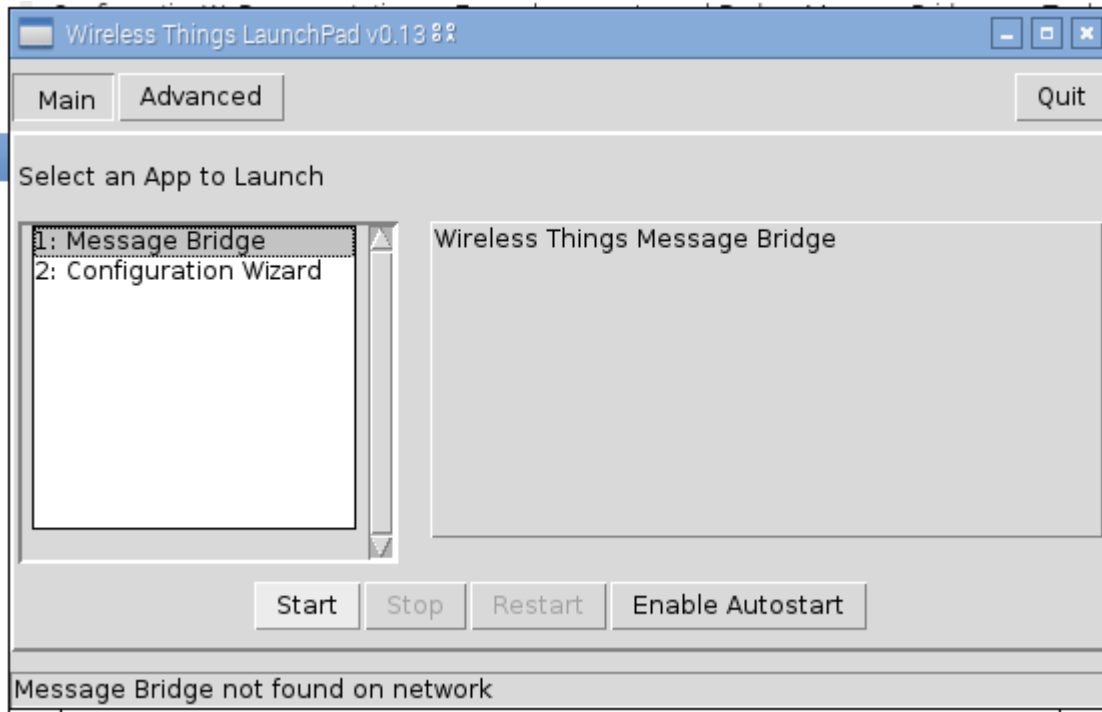
1. port: The serial port to use. Different devices (e.g.SRF Stick) may use a different serial port. The default Serial port is "/dev/ttyAMA0". For a WirelessThings USB based device this would be "/dev/ttyACM0", whilst under Windows it would be of the form "COM4" where the number reflects the COM port 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

Running the Message Bridge

When the LaunchPad runs you should see the following screen:



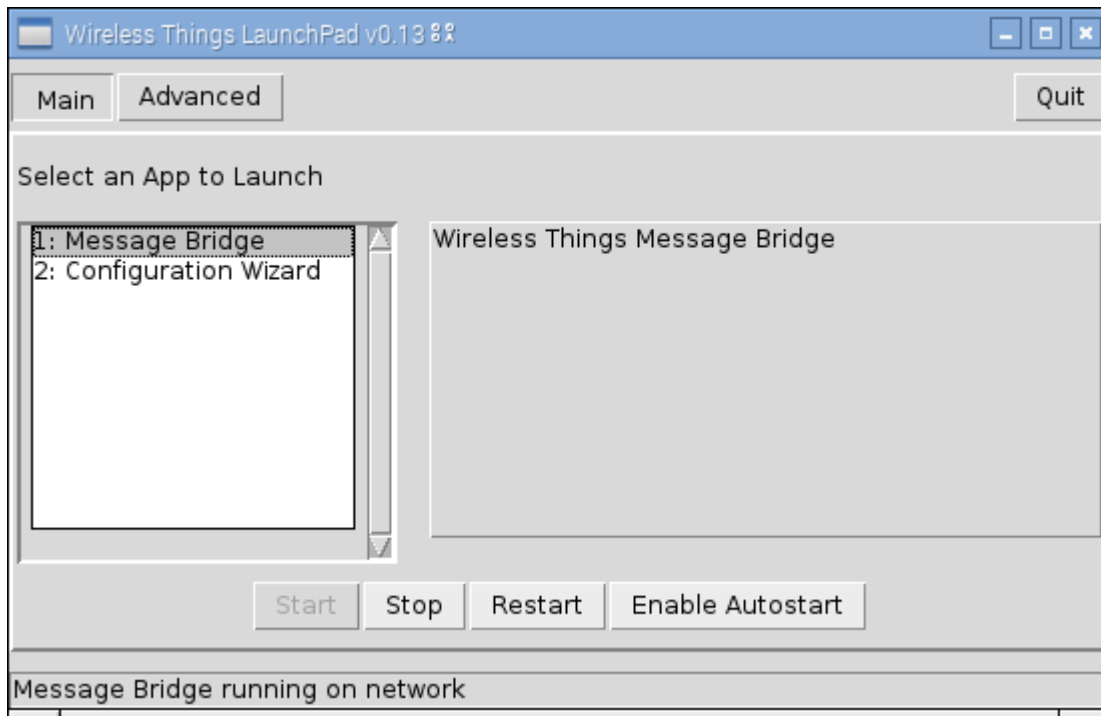
If "MessageBridge" is not selected then you will need to select it.

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:



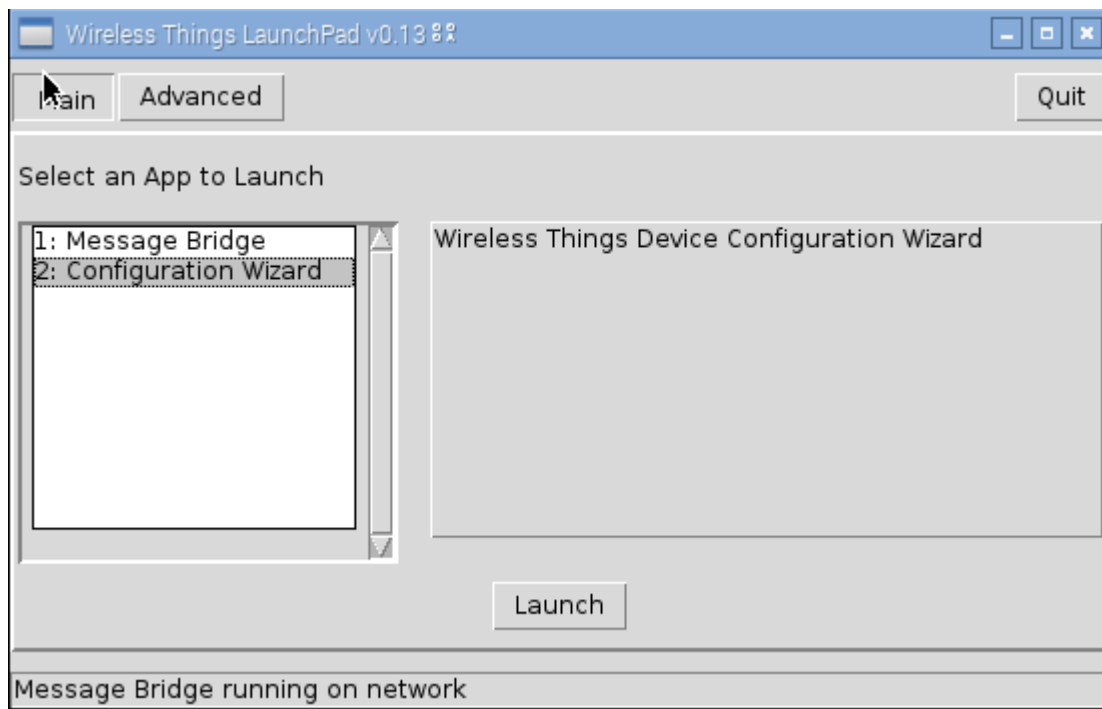
The Message Bridge will remain running even if you exit the LaunchPad.

6. 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:

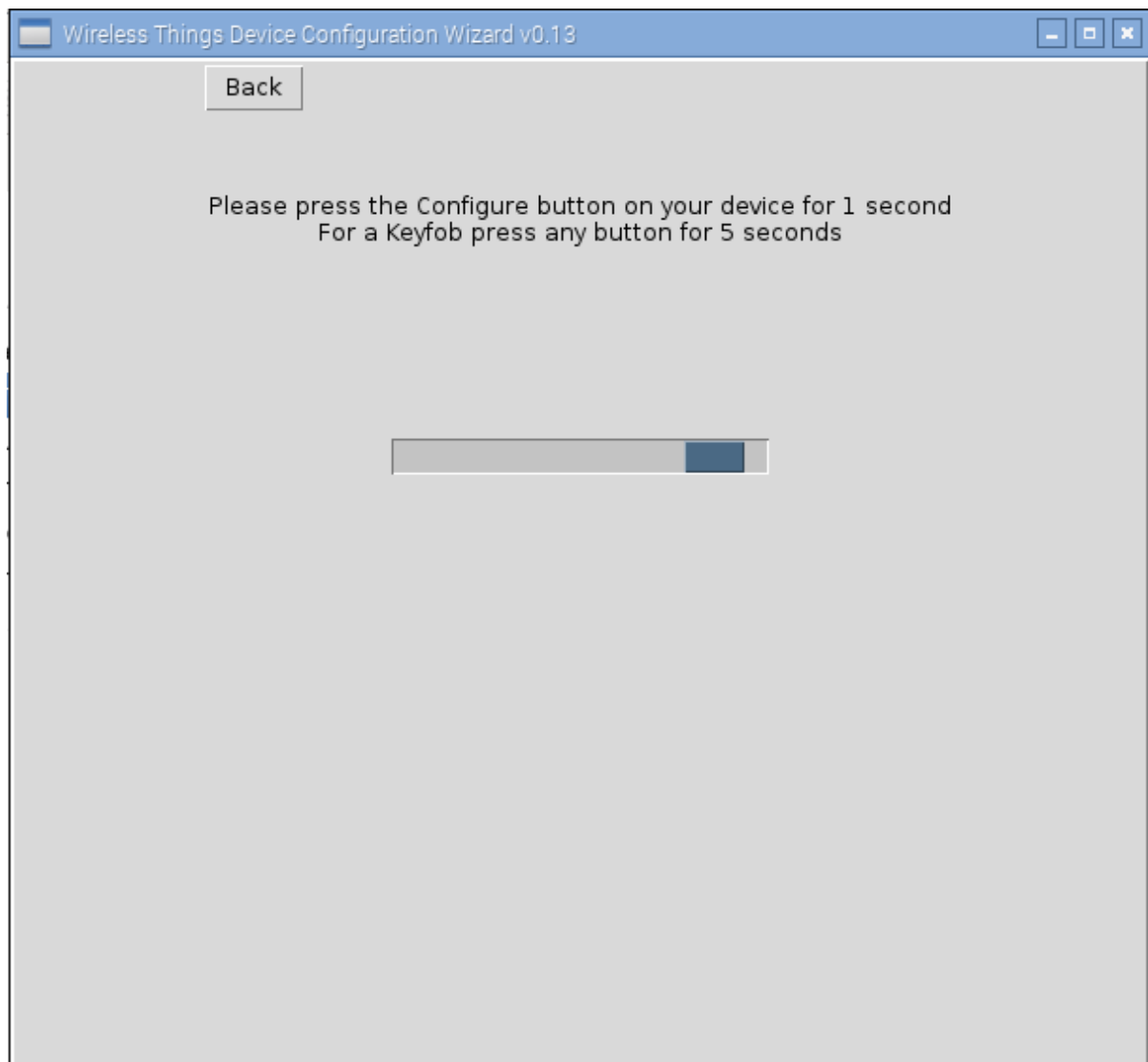


Press "Launch"



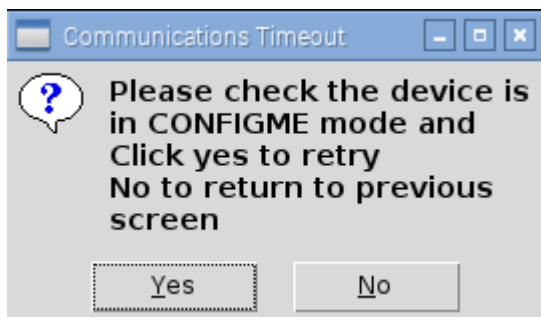
The Configuration Wizard will list the MessageBridges that it has detected running on your network. In this case there is only one – “JohnsSerial”.

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.

Wireless Things Device Configuration Wizard v0.13

Back Temperature Sensor SB-CA RSSI: -53

This Temperature Sensor device sends periodic readings in degrees centigrade. It is accurate to +/- 1 degree between -10 and +40 degrees centigrade

Device ID: JJ Change

Reading Interval: 10 Seconds

10 Second reading period.
Expected life will be ?

Apply settings to device

Reset to defaults

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 5 or 10 minutes is quite fast enough. A shorter reporting time will mean a reduced battery life.

Pressing “Apply settings to device” will write the currently selected settings to the device.

Advanced Configuration

The image shows a software window titled "Wireless Things Device Configuration Wizard v0.13". At the top, it displays "Temperature Sensor SB-CA" and "RSSI: -64". 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 JJ (with a help icon)
- Pan ID**: PANID 5AA5 (with a help icon)
- Retries for Announcements**: RETRIES 05 (with a help icon)

Cyclic Commands

- Sleep Interval**: INTVL 010S (with a help icon)
- Battery Wake Count**: WAKEC 010 (with a help icon)
- Enable Cyclic Sleep**: CYCLE ☒ (with a help icon)

At the bottom, there is an "Encryption Settings" button.

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. The Pan ID shown is read from the MessageBridge.

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.

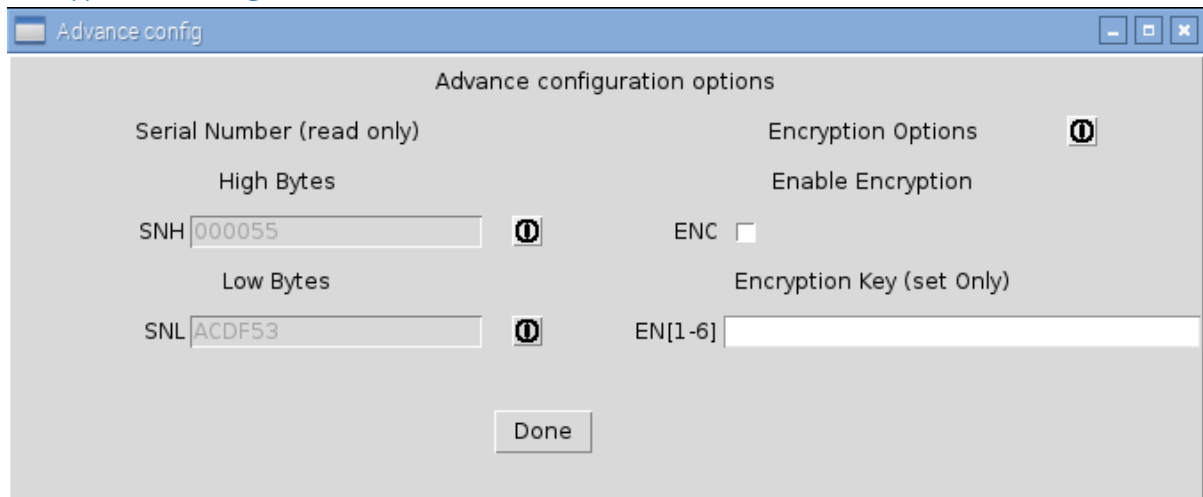
Cyclic commands: This give a finer grained way of controlling the reporting time.

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

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



The screenshot shows a window titled 'Advance config' with a subtitle 'Advance configuration options'. It contains two columns of settings. The left column, under 'Serial Number (read only)', has 'High Bytes' with a text box containing '000055' and a lock icon, and 'Low Bytes' with a text box containing 'ACDF53' and a lock icon. The right column, under 'Encryption Options', has 'Enable Encryption' with an unchecked checkbox labeled 'ENC', and 'Encryption Key (set Only)' with a text box labeled 'EN[1-6]' containing a single character. A 'Done' button is at the bottom center.

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. If a key is shown it has been read from the MessageBridge.

The encryption key is specified as 32 hexadecimal characters

e.g. ABCDEF0123456789ABCDEF0123456789

7. Software Examples

8. Downloading the WirelessThings LaunchPad

Download the zip file and extract to your desktop, this will create a folder called WirelessThings-LaunchPad, 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>

9. Updating the WirelessThings LaunchPad

When the LaunchPad is run it will check the url <https://www.wirelessthings.net/launchpad> for a later version and if one is found you will be prompted to download it.