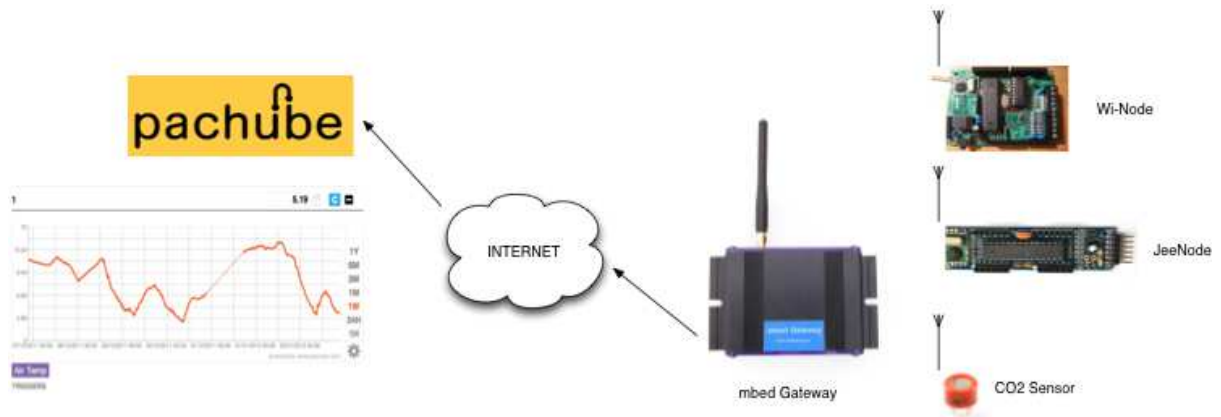


IoT Gateway



By Andrew Lindsay

Web: blog.thiseldo.co.uk

Twitter: @AndrewDLindsay

Project sponsor: SK Pang Electronics Ltd, www.skpang.co.uk

Contents

Introduction	3
Main features.....	3
Roadmap	3
Technical description	4
Hardware	4
Mbed I/O port usage.....	4
Software	5
Sensor Input modules	5
Routing.....	5
Output modules	6
Management interface	7
Appendices.....	8
Data formats	8
Simple Jeenode and Openenergymonitor RFM12B payload	8
RFM12B Payload – V1	8
RFM12B Payload – V2	9
File Formats.....	10
Main setup file – IOTSETUP.TXT.....	10
Node identification file – IOTNODE.TXT	10
Sensor Routing file – IOTRTG.TXT	11
References	12

Introduction

This document describes the design and technical details for a gateway for the *Internet of Things* (IoT). The gateway enables multiple wireless sensor devices to communicate with external services that will capture, record and display the data.

Main features

The main features of the IoT gateway are:

- Single low powered hardware gateway using mbed microcontroller platform
- Collect readings from multiple sensors
- Detect new sensors
- Send sensor readings to selected destinations
- Web based management and control interface
- RFM12B 868MHz, 433MHz is option
- Optional RFM22B module instead of RFM12B
- Socket for XBee, XRF or WiFly module
- Configuration files saved to local file system or μ SD card
- Open Source

Roadmap

The initial version of the gateway will only support file based configurations as described in this document, further enhancements are planned in addition to those already available:

V1 – Initial file based configuration with Pachube and MQTT output modules

V2 – Web based management interface

Further unplanned developments include WiFi instead of Ethernet, XRF LLAP and Xbee support.

Technical description

Hardware

The hardware for the gateway is based on the mbed rapid prototyping board and a custom designed IoT gateway board from SK Pang. The mbed board is based on the NXP LCP1768 microcontroller which is a 960MHz ARM based cpu with 512K flash and 64K RAM although 32K is available for mbed applications.

The IoT Gateway board includes:

- Socket for mbed
- Ethernet RJ45
- RFM12B 868MHz module, can use 433MHz module as alternative
- Pads for RFM22B as alternative to RFM12B
- Socket for XBee, Ciseco XRF or WiFly module.
- μ SD socket
- Power and status LEDs

Mbed I/O port usage

The IoT gateway utilises a number of the mbed interfaces to connect to the LEDs, radio modules, Ethernet and uSD socket. The port usage is shown in the table below.

Function	Port Description	Mbed Port
LEDs	LED1 on mbed	led1
	LED2 on mbed	led2
	LED3 on mbed	led3
	LED4 on mbed	led4
	Magjack Green (Low = On)	p30
	Magjack Yellow (Low = On)	p29
	Status (High = On)	p25
RFM12B/RFM22B (SPI 1)	MOSI	P11
	MISO	p12
	SCK	p13
	CS	p14
	Interrupt	p18
XBee, XRF, WiFly (Serial)	RXD	p27
	TXD	p28
	RESET	p26
μ SD (SPI 2)	MOSI	p5
	MISO	p6
	SCK	p7
	CS	p8

Software

The software for the IoT gateway is designed to be flexible, easy to configure and control. It has been designed to allow additional input and output modules to be easily coded and configured.

The gateway software consists of a number of parts:

- Sensor Input modules
- Routing
- Output modules
- Management interface

Sensor Input modules

Sensor nodes transmit readings at regular intervals and the readings are captured, processed and forwarded onto the output module by means of a routing configuration for each sensor.

The IoT gateway will have a number of sensor input modules defined that know how to deal with messages from a particular type of sensor. Example sensors using the RFMB12 interface are JeeNodes, OpenEnergyMonitor sensor and my own sensor data format. XBee support could be added later.

The sensor definition configuration depends on the hardware installed within the gateway and allows the user to setup the various parameters required by the input module.

Input module parameters that can be updated are:

- Payload format
- ID
- Channel/group

Routing

Routing is the process of taking the sensor reading and directing it to the correct output module.

This is performed by simple lookup on the sensor node configuration data. The routing data consists of a list of sensors, their input type, the output module and the output module parameters.

- Sensor ID
- Sensor Input module name
- Sensor Output module name
- Adjustment factor (e.g. multiply by 10, 100, 0.1, 0.01, 1)
- List of parameters for output module, e.g. feed and stream numbers or MQTT topic

Output modules

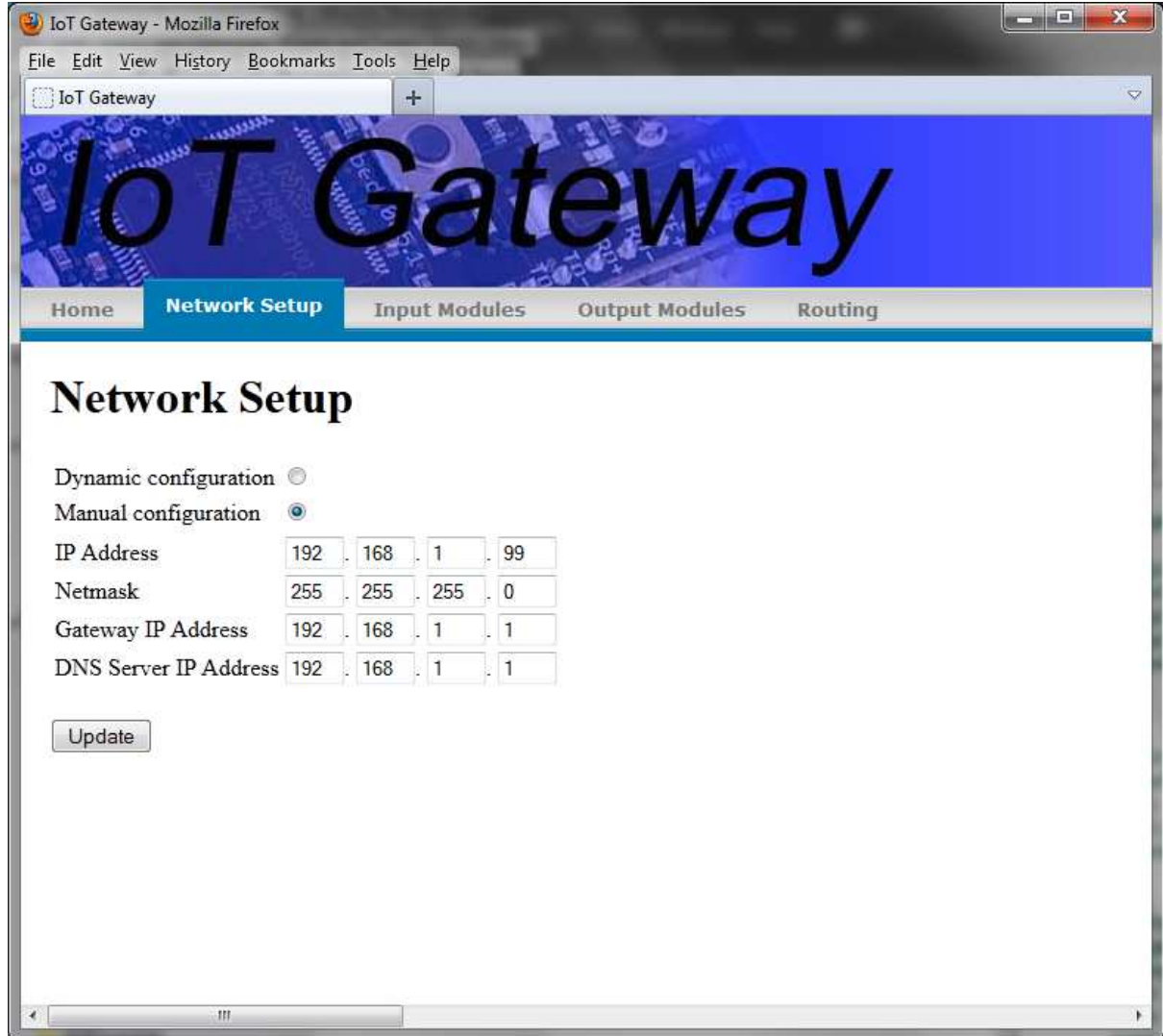
The output modules define a destination for sensor readings. This would normally be an online API call. This could be as simple as updating a Pachube feed, publishing to MQTT topic, sending the reading to a database or sending a twitter message.

Output module parameters:

- Name
- API URL
- API Key
- Parameters, e.g. for Pachube this is Feed ID and Stream ID

Management interface

The management interface is a web based tool that is used to create the configuration data used by the IoT gateway. It is menu driven with an overall dashboard or status screen.



Appendices

Data formats

The gateway supports a number of data formats that the various types of sensor can send. These are detailed below.

Simple Jeenode and Openenergymonitor RFM12B payload

This is a simple 2 bytes per reading with the number of readings worked out as being the payload length divided by two. Sensor ID is the position number of the reading within the payload. The payload layout is:

Reading 1	...	Reading n
-----------	-----	-----------

The format of the readings is:

Reading low byte	Reading high byte
------------------	-------------------

This is the default format for when the more advanced payload formats are not identified.

RFM12B Payload – V1

This is the format used by my Jeenodes and other sensors. It defines the payload format for the readings. Each block is in the diagrams a single byte. The payload layout is:

Status	Reading 1	...	Reading n
--------	-----------	-----	-----------

The format of the readings is:

Sensor ID	Reading low byte	Reading high byte
-----------	------------------	-------------------

Sensor Id is 1 byte and refers to the sensor on the particular node.

Status byte:

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	0	0	0	0	0	0	Low Battery

Low Battery = 1 when low battery condition is detected in sensor. This is currently detected by RFM12B modules present in the remote nodes.

RFM12B Payload – V2

This is the updated format to be used by my Jeenode and other sensors. It defines the payload format for the readings. Each block is a single byte. The payload layout is:

Status	Reading 1	...	Reading n
--------	-----------	-----	-----------

The format of the readings is:

Bits 4-7 Data Type	Bits 0-3 Sensor ID high byte	Sensor ID Low byte	Reading Data
-----------------------	------------------------------------	-----------------------	--------------

The format of the reading data is dependent on the data type being used

Data Type	Value	Reading Data length (Bytes)	Notes
Byte (8bits)	0	4	
Int (16 bits)	1	5	
Long (32bits)	2	7	
String	3	4 + string length	String is defined as length plus characters. Max length is 40 characters.

Status byte has been updated with the format version so that packets can be identified easily:

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	0	0	0	0	Version1	Version0	Low Battery

Low Battery = set to 1 when low battery detected.

Version: 00 – version 1 (For compatibility with V1 format), 01 – version 2

File Formats

Main setup file – IOTSETUP.TXT

The main IoT Gateway configuration is defined in the file IOTSETUP.TXT, this can be edited by hand or updated via the web management interface. Any changes made by the management interface will overwrite previous changes. A restart is needed for the new parameters to be read.

```
ip.mode=fixed (or dhcp)
ip.address=192.168.1.99
ip.netmask=255.255.255.0
ip.gateway=192.168.1.1
ip.dns=192.168.1.1
rfm12b.band=868
rfm12b.id=30
rfm12b.group=212
time.timezone=GMT
time.dst=yes
pachube.key=your pachube api key
mqtt.host=192.168.1.12
mqtt.port=1883
mqtt.username=any username if host requires one
mqtt.password=password if host requires one
```

Node identification file – IOTNODE.TXT

It should be possible for the IoT gateway to identify payloads itself if only the V1 and V2 type are used. It is also possible to use the default simple format too but these may be miss-interpreted as a V1 format if the correct combination of bytes is present in the payload. In view of this, there is a node identification file. This is used to build up the internal data structures that determine how payloads are interpreted and dealt with. It basically maps the first three bytes in the payload, namely the group ID, node ID and length. If new nodes are detected then they are added to the file automatically. They will have a payload type of 99. This can then be updated by hand to the correct format.

Column	Name	Type	Description
1	Group ID	Byte	First byte from the received payload, the group Id.
2	Node ID	Byte	Second byte from the received payload, the node Id.
3	Data length	Byte	Third byte from the received payload, the data length.
4	Payload type	Byte	Payload type for this node 0 = Simple, 1 = V1, 2 = V2, 99=unknown

Values are expressed as decimal values only.

Example file:

```
5,10,4,0
5,5,7,2
```

Sensor Routing file - IOTRTG.TXT

The sensor routing table determines where a reading from a sensor node is to be directed to. The file can be edited by hand or updates for new nodes and sensors can be added via the management interface. Unknown node/sensor combinations are added to the file automatically when they are detected. These can then be updated manually or via the management interface with the correct parameters. The format of the file is as a .csv with the columns defined as follows:

Column	Name	Type	Description
1	Node ID	Short	ID of the Node.
2	Sensor ID	Short	ID of the sensor or reading number within the node payload. Starts at 0 for multiple readings in simple payload. Set to -1 for whole node low battery status monitor.
3	Multiplication factor	Float	Value to multiply reading by to get actual value, 1, 10, 100, 0.1, 0.01. e.g. to convert from 12340 to 123.40 use factor 0.01. Use 1 for no change.
4	Output Type	Short	Output type. Pachube = 1, MQTT = 2
5	Output Param 1	String	Parameter 1 value, e.g. Pachube Feed, Mqtt topic
6	Output Param 2	String	Parameter 2 value, e.g. Pachube Datastream
7	Output Param 3	String	Parameter 3 value
8	Output Param 4	String	Parameter 4 value

Not all values would be used by all output modules.

Example file:

```
1,1,0.01,1,,2227,0,
1,3,1,2,sensor/temperature,,,
1,2,0.1,1,,9506,3,
```

References

mbed.org – mbed development boards, info and online compiler.

www.skpang.co.uk/catalog/mbed-iot-gateway-full-kit-p-1052.html - SK Pang IoT Gateway

pachube.com – IoT logging, graphing, triggering etc.

openenergymonitor.org - Home of the Open Energy Monitor project

jeelabs.org – Home of the JeeNode

mqtt.org - MQTT information.

<http://ceit.uq.edu.au/content/mqttclient-mbed-version-20> MQTT for mbed