PYGATE FIRMWARE REFERENCE.

The Pygate firmware was modified to enable LoRa packets to be transmitted and received from the Semtech SX1308 concentrator by sending and receiving UDP packets on ports on the loopback network interface of the development board used. It is important to note that the Pycom WLAN class *must* be initialised before the loopback interface, and therefore the UDP packet forwarder, will function.

Sending LoRa Packets from the concentrator

In order for some payload data D with length $|D|$ to be transmitted with frequency f (MHz), Bandwidth bw (KHz), Spreading Factor sf, Code Rate cr (e.g. $4/5$) and power p (dBm) on radio module r ($r \in \{0,1\}$ for the Pygate which is equipped with two Semtech 1257s), the following JSON object must be constructed. δ is constructed by creating a string of format SF$< sf >$BW$< bw >$ (e.g. SF7BW125 for $sf = 7, bw = 125$). β is the payload encoded in Base64.

```
{
“txpk”: {
            “freq”:f,
            “rfch”:r,
            “powe”:p,
            “datr”: δ,
            “codr”: cr,
            “size”: |D|,
            “data”: β },
}
```

Figure 1 | Transmitting LoRa UDP packet format.

The JSON object is then sent as a string to port 6001 via UDP on the loopback interface. Provided the Pygate Firmware is running, the packet will immediately be transmitted by the LoRa concentrator.

Receiving LoRa Packets

In order to receive LoRa packets from the concentrator, the application must listen to port 6000 on the loopback interface. UDP packets will be received on this port however the process of decoding the packets is slightly more involved than that of transmission. The packet structure is shown in Figure 2.

Name	Size / Bytes	Description
Packet Type	1	Indicates whether the UDP packet contains a LoRa packet or the Time On Air of the previously transmitted packet. LoRa Packet: 0x01.
Payload Size (S_p)	1	Size in Bytes of the payload.

Header Size (S_h)	1	Size in Bytes of the header which contains packet receive information.
Header	S_h	Header containing packet metadata in a JSON object encoded as UTF-8.
Payload	S_p	LoRa packet payload encoded as UTF-8

Figure 2 | Structure of packets received via UDP containing Time On Air of previously sent packet or a received LoRa packet.

Packet Type	Code
TOA	0x02
LoRa	0x01

Figure 3 | Shows Byte codes of the two packet types which may be received from the Pygate firmware on port 6000.

Receiving Time On Air of Previously Transmitted Packet

When a LoRa packet is transmitted by the concentrator, the firmware immediately sends a UDP packet containing the Time On Air of the packet to port 6000 on the loopback interface. The packet structure is given by Figure 4.

Name	Size / Bytes	Description
Packet Type	1	Indicates whether the UDP packet contains a LoRa packet or the Time On Air of the previously transmitted packet. TOA Packet: 0x02.
Time On Air (ms)	4	Time On Air of previously transmitted packet in milliseconds. Field is a long type encoded as little-endian.

Figure 4 | Packet structure of Time On Air packet which is sent by the firmware immediately after a LoRa packet is transmitted by the concentrator.

Firmware Thread Flowchart

The firmware utilises two threads. The first is for LoRa transmission and forwards packets received via UDP. The second is for LoRa receptions and forwards LoRa packets received to the loopback interface via UDP. A future firmware version will see the two threads combined since socket functions used are non-blocking. The operation of the firmware threads is detailed in Figure 5.

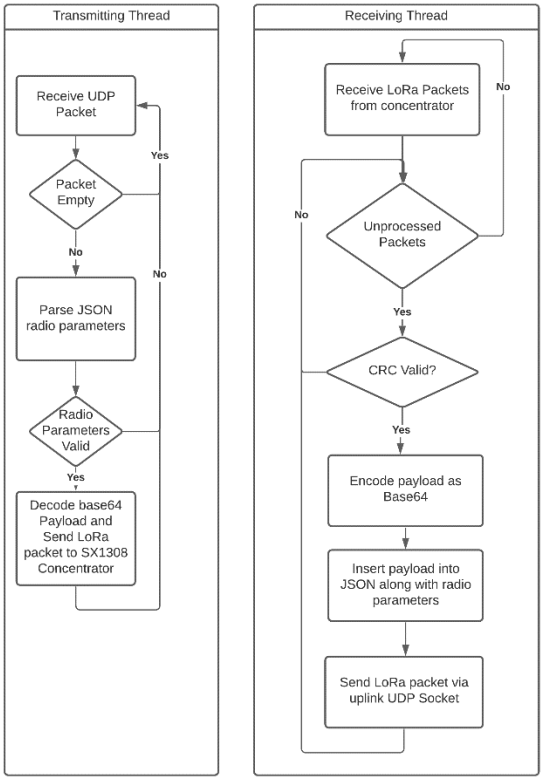

Figure 5 | Flowchart presenting firmware thread peration. 'Receive UDP Packet' is a non-blocking operation which will return an empty UDP packet if the buffer is empty. 'Receive LoRa Packets from concentrator' is a non-blocking operation which returns a list of packets from the LoRa concentrator buffer.

Future Development

There are two main areas of development at the time of writing:

1. Consider modifying the firmware so that the loopback interface is not required and the concentrator is directly interfaced with the MicroPython driver.
2. Provide configuration of the ports and addresses used by the firmware to send and receive LoRa packets via UDP. This would enable the concentrator to be controlled by other devices connected on the WLAN.