### Packet Structure (4 Byte MAC Address)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Offset | Octet | 0 | 1 | 2 | 3 |
| Octet | Bit | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| 0 | 0 | Destination MAC Address |
| 4 | 32 | Source MAC Address |
| 8 | 64 | Checksum |
| 12 | 96 | Sequence Number | REL | Type | ACK | Length |
| 16 | 128 |  |
|  |  | Data |
|  |  |  |

### MAC Address

|  |  |  |  |
| --- | --- | --- | --- |
| DC | xx | xx | xx |

Byte 0 = DC (Langlo devices; Other prefixes will be allocated as required)

Bytes 1-3 = Last three bytes of default device MAC address or similar

### Checksum

|  |
| --- |
|  |

32-bit checksum calculated on the remainder of the packet (from Byte 12 to end-of-packet)

### Sequence Number

|  |
| --- |
| n |

n = 1..65535

Sequence number simply wraps after reaching 65535.

### REL bit

|  |
| --- |
| n |

n = 0..1

0 => Unreliable delivery
1 => Reliable delivery—ACK required

### Type

|  |
| --- |
| n |

n = 0..127

Packet payload Type will define the structure of the date provided in the payload. At this point, it will probably define the length, making the Length byte redundant, but who knows what we might need this for in the future.

For the moment, the following packet payload Types are defined:

#### ACK (0x00)

No longer used

#### Voltage (0x01)

Module supply voltage in millivolts (uint16\_t) (2 bytes)

#### Power (0x02)

Module battery, solar panel (if relevant) and load voltage in millivolts (uint16\_t) and current in milliamps (uint16\_t) status (12 bytes in total).

#### Tank Level (0x11)

Single byte pressure reading (uint8\_t) (HDL300 sensor).

#### Pump Status (0x12)

Single bit on/off status (but currently coded in a whole byte—uint8\_t). Will need to report this periodically, to confirm off-state under normal conditions. Will also need to report more frequently (Every 10 seconds? More/less often?) when running, once again to verify state.

#### Weather Report (0x20)

Should ultimately include temperature as °C  10 (uint16\_t), pressure in kPa (uint16\_t), humidity as a percentage (uint8\_t), rainfall as mm  10 (uint16\_t), wind direction in degrees (uint16\_t) and wind speed in kph (uint8\_t) (10 bytes in total).

#### Atmosphere Report (0x21)

A subset of the Weather Report, including atmospheric temperature (uint16\_t), pressure (uint16\_t) and humidity (uint8\_t) only (5 bytes in total).

#### Rainfall Report (0x22)

A subset of the Weather Report, including rainfall (uint16\_t) only (2 bytes).

#### Wind Report (0x23)

A subset of the Weather Report, including wind direction (uint16\_t) and wind speed (uint8\_t) only (3 bytes in total).

#### VOX Report (0x24)

Volatile organics level (uint16\_t) (CCS811 Sensor) (2 bytes in total).

#### Ambient Light Report (0x25)

Ambient light level as lux  100 (uint16\_t) (MAX44009 Sensor) (2 bytes in total).

#### UV Report (0x26)

UV status, including UVA (uint16\_t), UVB (uint16\_t) and UV Index (uint16\_t) (VEML6075 Sensor) (6 bytes in total).

#### Sprinkler Controller (0x30)

Reports available per OpenSprinkler API, via WiFi, so may not even need to be included here, but structure undefined at this point in any case.

#### GPS Report (0x40)

Report includes latitude (float), longitude (float), altitude (float) and HDOP (float) (16 bytes in total).

#### AWTS Report (0x50)

Report designed specifically for a particular Aerated Wastewater Treatment System (AWTS). The report currently includes blower pressure (uint16\_t) and [influent] tank level (uint8\_t) (3 bytes in total).

#### Reset Report (0x7E)

Report includes a reset code (uint8\_t). Codes other than 0 (normal processor start-up) are yet to be defined. (1 byte in total).

All reports could also include an equipment status report, including battery condition, uptime and anything else that seems like a good idea, but the mechanism for including these is yet to be defined.

### ACK bit

|  |
| --- |
| n |

n = 0..1

0 => Normal packet (payload included)
1 => ACK packet (no payload)

### Length

|  |
| --- |
| n |

n = 1..48

Packet payload length. In line with the whole LoRa philosophy, this should never get anywhere near maxing out. Packet payload length will, for the moment at least, be defined by the packet payload Type.