L-COM communication interface design

Stephen Amey

June 5th, 2021

Contents

1. Physical interface

The L-COM module utilizes a serial interface to communicate with an attached device. This serial interface is part of a 4-wire physical interface that may be connected to by an external device, as seen in Figure 1.

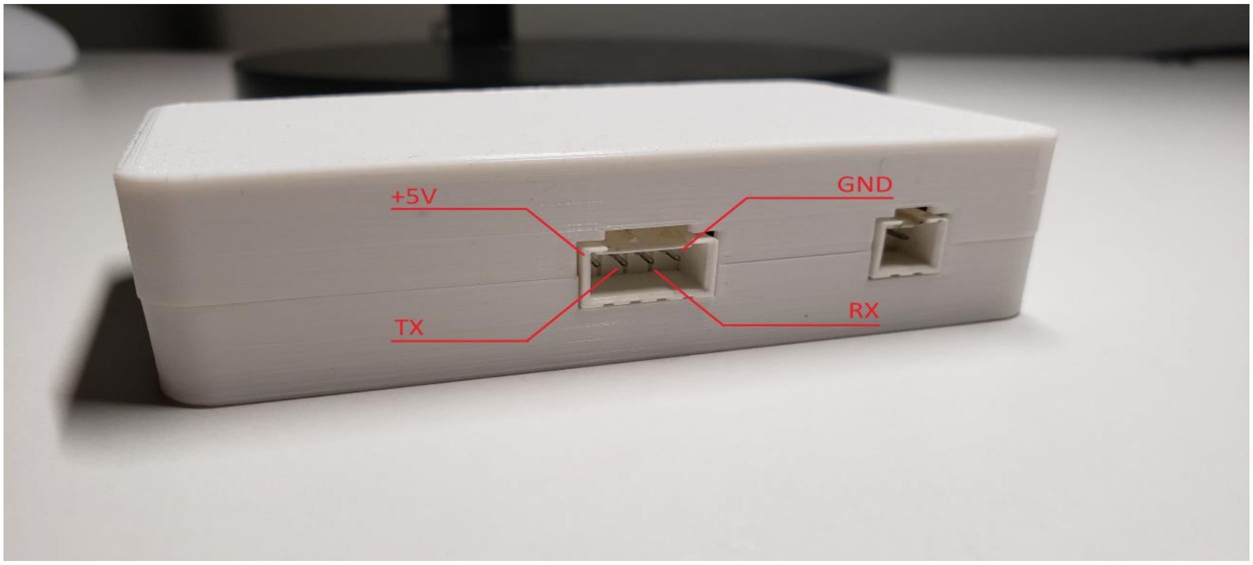

Figure 1, physical communication interface of L-COM

Providing +5V via the interface is not required if powering via other methods, however the GND connection is required. While the L-COM operates at 3.3V and and TX pin switches between GND and 3.3V, the RX pin is 5V tolerant. The interface is not immediately compatible with USB communication, but a separate serial-USB converter module may be connected to the interface to serve as a bridge.

In addition to the external connector interface, the L-COM also provides an internal pin header that may also be utilized for serial communication. However, care must be taken in regards to connecting the right pins and physically securing it so that it does not disconnect. The internal view and pin header may be seen in Figure 2.

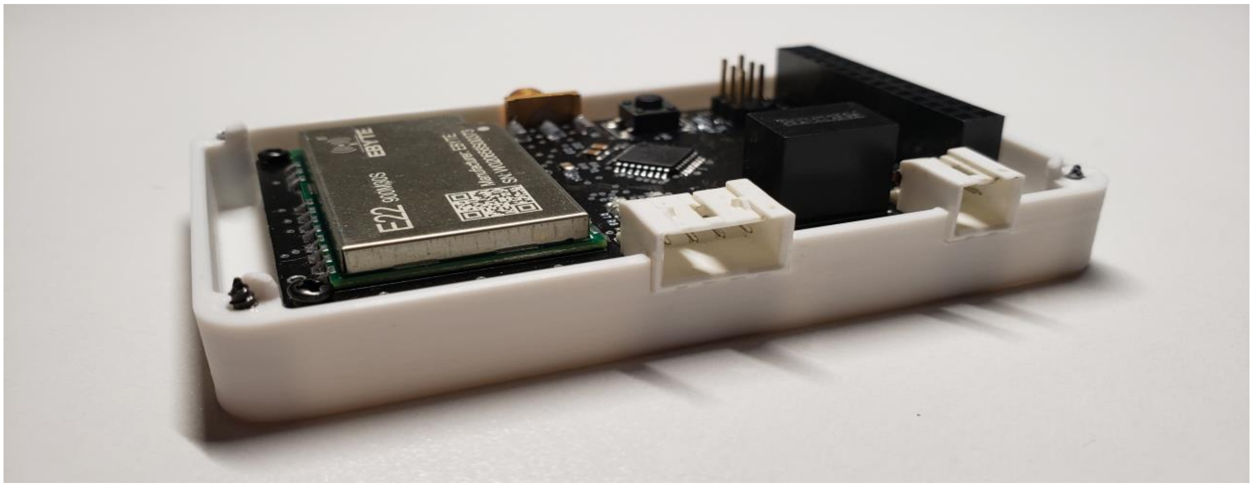

Figure 2, internal view of L-COM

2. Serial packet formatting

There are four types of packets that may be sent over the serial connection:

- Status
- Command
- Ack
- Message

These packets are purely for the serial connection between the L-COM and the attached device, and are not used for over-the-air transmission between L-COM devices. Over-the-air transmission uses the format defined in the RadioLib library, with a programmer defined payload.

2.1 Common packet header

Each type of packet shares a common header with several fields, as seen in Figure 3:

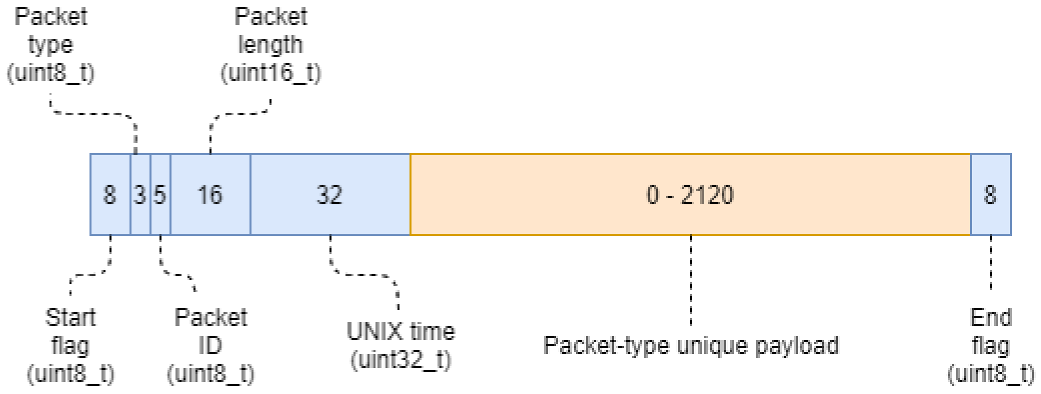

Figure 3, common packet header of serial transmission

- **Start and end flag** fields indicate the start and the end of the packet respectively, and both flags should be received in-order before attempting to process the packet data.
- **Packet type** field indicates the type of packet received, e.g. Status, Command, Ack, Message, and how the packet-type unique payload should be interpreted.
- **Packet ID** field provides an ID for the packet, and may be used to tell different packets apart and if any have been missed. This ID is unique to the L-COM device and is not used as a 'shared ID'; both sides of the communication link may send packets with the same ID, but must keep track of the other side's IDs.
- **Packet length** field indicates how long the entire packet is (including the common header) in bytes. This value should also be used to confirm that the end flag field is received in the correct location within the packet and that no bytes are missing, and ensure the validity of the packet.
- **Unix time** contains the current UNIX time as tracked by the L-COM device. Until the UNIX time is set by a command, this field will contain zero as the L-COM cannot determine the current UNIX time by itself.

- **Packet-type unique payload** contains up to 265 bytes of different data depending on the type of packet (as specified by the Packet type field). Refer to the subsections below.

2.2 Command packet

A command packet is sent from the attached device to the L-COM, and is used to run built-in commands. A command packet contains two sub-fields: Command and Parameters, as seen in Figure 5. A command packet does not need to utilize all of the space allocated for parameters.

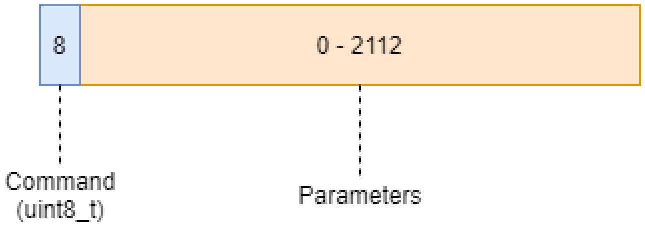

Figure 4, Command packet

- **Command** indicates which specific command is to be run by the L-COM.
- **Parameters** contains any extra data needed by specific commands, and is possible to be empty. Refer to section 2. Commands for the parameters required by specific commands.

After the L-COM receives a command packet, it will attempt to execute the command. Following the execution attempt, the L-COM will then send an Ack packet to the attached device indicating the result of the command.

2.3 Ack packet

An Ack packet is sent from the L-COM to the attached device either after the attached device sends a command packet to the L-COM, or the attached device sends a message packet to the L-COM to transmit. An Ack packet consists of a two sub-fields: Result and Return data. The Ack packet is seen in Figure 6.

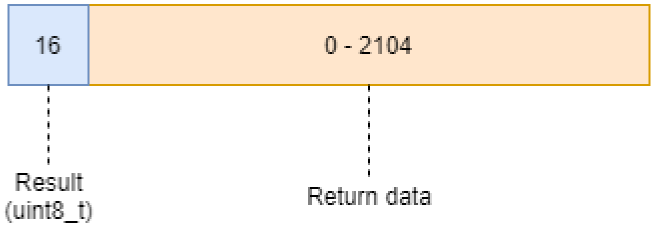

Figure 5, Ack packet

- **Result** is the result code that indicates whether the command or transmission occurred successfully, or ran into an error. This field uses the RadioLib error codes in addition to L-COM specific ones.
- **Return data** is used to return any additional data along with the result, e.g. return values from a command.

2.4 Message packet

A message packet can be sent either from the L-COM to the attached device, or from the attached device to the L-COM. A message packet has four sub-fields: RSSI, SNR, Status, and Message payload, as seen in Figure 7.

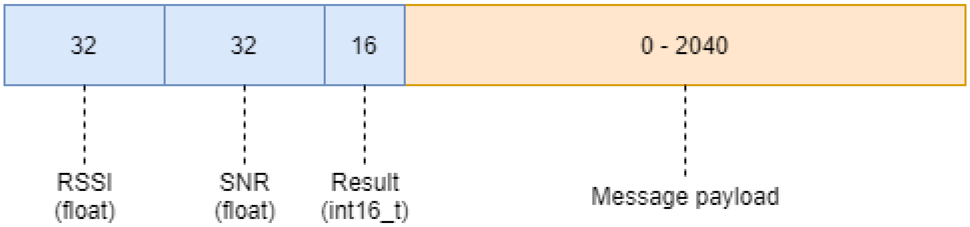

Figure 6, Message packet

- **RSSI** represents the received-signal-strength-indicator. This field is only used when the L-COM receives a message and sends it to the attached device. When the attached device sends a message to the L-COM (i.e. to transmit), this field should be set to zero.
- **SNR** represents the signal-to-noise ratio. This field should be used in the same way as the RSSI field, where it should be set to zero when sent by the attached device.
- **Result** indicates the health of the received packet using the RadioLib defined error codes. This field should be used in the same way as the RSSI and SNR fields, where it should be set to zero when sent by the attached device. When a packet is received over-the-air, a value of zero (i.e. ERR_NONE) indicates the packet is healthy. Other values may indicate CRC mismatches, timeouts, and other errors. Refer to the RadioLib status codes documentation.
- **Message payload** holds the received or to-be-transmitted message. When transmitting, the contents of this field are at the discretion of the programmer the data that needs to be transmitted.

After sending a message packet from an attached device to the L-COM to transmit, the L-COM will then send an Ack packet back to the attached device to indicate the result of the transmission. An Ack packet **is not** sent when the L-COM receives a message over-the-air and sends it to the attached device.

3. Commands

The L-COM module is capable of executing a variety of commands. These commands are sent by an attached device over the serial interface, and are executed as soon as they are received. These are divided into set, get, and miscellaneous commands, each with their own set of parameters. Following the execution of a command, an acknowledge packet will be transmitted containing the result code, and any extra data that would be returned by the command

3.1 Set commands

3.1.1 setLoRaParameters

This command is used to set the LoRa parameters for the EBYTE transceiver module. Upon power-on or reset, the parameters are by default unset, and must be set prior to attempting communication.

Command - ID	setLoRaParameters– 0x00								
Bytes	**0**	**1-4**	**5-8**	**9**	**10**	**11**	**12**	**13-14**	**15-18**
Data type	**uint8_t**	**float**	**float**	**uint8_t**	**uint8_t**	**uint8_t**	**uint8_t**	**uint16_t**	**float**
Field	**Command ID**	**Frequency**	**Bandwidth**	**Spreading factor**	**Coding rate**	**Sync word**	**Power**	**Preamble length**	**Current limit**
Returns	**Result code (uint16_t)**								

3.1.2 setUNIX

This command is used to set the UNIX timestamp of the L-COM, and enables time-keeping so that the module can provide timestamps of its actions and transmissions.

Command - ID	setUNIX – 0x01	
Bytes	**0**	**1-4**
Data type	**uint8_t**	**uint32_t**
Field	**Command ID**	**Timestamp (UNIX)**
Returns	**Result code (uint16_t)**	

3.1.3 setModeMessage

This command is used to set the mode and repeater message of the L-COM module. Zero (0x00) is the normal operating mode capable of receiving and transmitting. One (0x01) is the repeater operating mode, where when a packet is received over-the-air, it will transmit a set message. The repeater message may be a maximum of 255 characters (depending on usage, remember to take account of carriage returns, newlines, and null terminators), as LoRa allows a maximum message size of 255 bytes. By default, the repeater message is an empty string (all zeros). Repeater mode is intended to aid in recovery conserve power by only transmitting when it receives packets from a mobile ground station. By default, the L-COM is in normal operation mode (0).

Command - ID	setModeMessage – 0x02		
Bytes	**0**	**1**	**2-257**
Data type	**uint8_t**	**uint8_t**	**char × 255**
Field	**Command ID**	**Mode**	**Message**
Returns	**Result code (uint16_t)**		

3.2 Get commands

3.2.1 getLoRaParameters

This command is used to get the LoRa parameters of the EBYTE transceiver module. By default, the parameters will be unset on power-up or restart.

Command - ID	getLoRaParameters– 0x10
Bytes	**0**
Data type	**uint8_t**
Field	**Command ID**
Returns	**Result code (uint16_t), Frequency (float), Bandwidth (float), Spreading factor (uint8_t), Coding rate (uint8_t), Sync word (uint8_t), Power (uint8_t), Preamble length (uint16_t), Current limit (float)**

3.2.2 getUNIX

This command is used to get the UNIX timestamp from the L-COM. If the UNIX timestamp was not already set by command and time-keeping enabled, then this command will return the default value of zero.

Command - ID	getUNIX – 0x11
Bytes	**0**
Data type	**uint8_t**
Field	**Command ID**
Returns	**Result code (uint16_t), Timestamp (uint32_t)**

3.2.3 getModeMessage

This command is used to get the current mode and repeater message of the L-COM. By default, the L-COM is in normal operation mode (0), and the repeater message is an empty string (all zeros).

Command - ID	getModeMessage – 0x12
Bytes	**0**
Data type	**uint8_t**
Field	**Command ID**
Returns	**Result code (uint16_t), Mode (uint8_t) , Message (char × 255)**

3.2.4 getModuleStatus

This command is used to trigger the L-COM to send an Ack packet containing status info on several parts of the L-COM hardware and software*.

Command - ID	getModuleStatus – 0x13

Bytes	**0**
Data type	**uint8_t**
Field	**Command ID**
Returns	**Result code (uint16_t)*, LoRa set (bool), UNIX set (bool), Uptime (uint32_t), Temperature (float), TOA (uint32_t), Data rate (float)**

3.3 Miscellaneous commands

3.3.1 radioReset

This command is used to trigger a reset of the EBYTE transceiver module.

Command - ID	radioReset– 0x20
Bytes	**0**
Data type	**uint8_t**
Field	**Command ID**
Returns	**Result code (uint16_t)**

3.3.2 systemReset

This command is used to trigger a full reset of the L-COM software. This requires that the watchdog be enabled, entering an infinite loop, and the watchdog causing a reset.

Command - ID	systemReset– 0x21
Bytes	**0**
Data type	**uint8_t**
Field	**Command ID**
Returns	**Result code (uint16_t)**

3.3.3 negotiateLoRaParameters

This command is used to trigger the L-COM to begin a LoRa parameter negotiation process with the other L-COM module that it is paired with by code-word. The LoRa parameters should have already been set and communication with the other module achieved before attempting to use this command.

Command - ID	negotiateLoRaParameters– 0x22
Bytes	**0**
Data type	**uint8_t**
Field	**Command ID**
Returns	**Result code (uint16_t)**