XBee UART Python functions

Chris Vincent Densing

October 28, 2019

Part I
Packet encode functions: Generic

Includes functions for formation of packets that conform to the XBee API format. Included in file *packet_encode.py*.

1 AT Command Query (atcom_query)

Assemble a byte stream for querying AT parameter values. Use on local XBee device connected to serial.

1.1 Returns:

1. *bytestr* - (type: Bytes) Formatted XBee API frame (0x08: AT Command frame).

1.2 Arguments:

1. *param* - (type: String) Two-character AT parameter.

2 AT Command Set (atcom_set)

Assemble a byte stream for setting AT parameter values. Use on local XBee device connected to serial.

2.1 Returns:

1. *bytestr* - (type: Bytes) Formatted XBee API frame (0x08: AT Command frame).

2.2 Arguments:

1. *param* - (type: String) Two-character AT parameter.
2. *value* - (type: Bytes) New AT parameter value.

3 Transmit request (gen_txreq)

Assemble a byte stream for a transmit request payload. Use on local XBee device connected to serial. Returned stream still needs to have headers and checksum appended (use in conjunction with gen_headtail()). Refer to XBee documentation for details on arguments.

3.1 Returns:

1. *bytestr* - (type: Bytes) Semi-formatted XBee API frame (0x10: Transmit request frame).

3.2 Arguments:

1. *fid* - (type: String) 1-byte Frame ID in hexadecimal.
2. *dest* - (type: String) 8-byte Destination address in hexadecimal.
3. *brad* - (type: String) 1-byte Broadcast radius in hexadecimal.
4. *opts* - (type: String) 1-byte Transmit options in hexadecimal.
5. *data* - (type: String) Variable length payload in hexadecimal.

4 Generate Header and Checksum (gen_headtail)

Append headers (0x7e and 2-byte length) and checksum to a generic payload. Use on local XBee device connected to serial.

4.1 Returns:

1. *bytestr* - (type: Bytes) XBee API frame.

4.2 Arguments:

1. *bytestr* - (type: Bytes) Assembled payload without headers and checksum.

Part II
Packet encode functions: Specific

Includes functions for formation of command packets specific to the RESE2NSE node 1 firmware. Included in file *packet_encode.py*.

5 Remote node unicast (debug_unicast)

Assembles command packet for issuing a unicast command to a remote node in standby/debug mode.

5.1 Returns:

1. *bytestr* - (type: Bytes) XBee API frame.

5.2 Arguments:

1. *n* - (type: Integer) Number of unicast packets remote node will transmit back.
2. *dest* - (type: Bytes) 8-byte address of remote node.

6 Remote node set channel (debug_channel

Assembles command packet for changing the radio channel of a remote node. Command is processed by the MSP430.

6.1 Returns:

1. *bytestr* - (type: Bytes) XBee API frame.

6.2 Arguments:

1. *ch* - (type: Bytes) 1-byte new channel.
2. *dest* - (type: Bytes) 8-byte address of remote node.

7 Remote node set power level (debug_power

Assembles command packet for changing the radio transmit power of a remote node. Command is processed by the MSP430.

7.1 Returns:

1. *bytestr* - (type: Bytes) XBee API frame.

7.2 Arguments:

1. *pow* - (type: Bytes) 1-byte new power level.
2. *dest* - (type: Bytes) 8-byte address of remote node.

8 Remote node start sensing (start_sensing

Assembles command packet for transitioning from stanby/debug mode into sensing mode of a remote node.

8.1 Returns:

1. *bytestr* - (type: Bytes) XBee API frame.

8.2 Arguments:

1. *period* - (type: Integer) 1-byte sampling period.
2. *dest* - (type: Bytes) 8-byte address of remote node.

9 Remote node stop sensing (stop_sensing

Assembles command packet for transitioning from sensing mode to stanby/debug mode of a remote node.

9.1 Returns:

1. *bytestr* - (type: Bytes) XBee API frame.

9.2 Arguments:

1. *dest* - (type: Bytes) 8-byte address of remote node.

10 Remote node query parameter (debug_query

Assembles command packet for querying remote node parameters. This includes AT parameters (must be implemented in remote node firmware) and MSP430 parameters.

10.1 Returns:

1. *bytestr* - (type: Bytes) XBee API frame.

10.2 Arguments:

1. *atcom* - (type: String) Parameter to query.
   Supported parameters:
   (a) PL - power level
   (b) CH - channel
   (c) A - aggregator address
   (d) T - sampling period
2. *dest* - (type: Bytes) 8-byte address of remote node.