SENDING SNIFFER MESSAGES:

You command the sniffer for sending RF messages to the network by just sending a character on the serial port (the firmware reads it and sends the proper RF message).

The commands (ASCII characters, without \n or \r) are:

      '0'   - deactivate sniffer mode on ALL the possible gluons on the network
      '1'   - activate normal sniffer mode on all the gluons (no double notice for connections or data packets)
      '2'   - activate verbose sniffer mode
      '3'   - synchronize gluons clock
      '4'   - scan network (report all gluons and describe their connections)
      '5'   - delete all connections in the network

NOTE: if DEBUG_MODE is defined (in file utils.h), then the sniffer will answer back with the name of the command just broadcasted. For example, when sending '1', you will receive back on the serial port:

      Snd RF: 254 -> 255
      Msg: SNIFF_ON

If you don't want to deactivate the debug mode, just ignore Snd and Msg during the parising of the receiver sniffer messages.

RECEIVING SNIFFER MESSAGES:

There are of two kinds: (a) normal packets and (b) connection transactions. For easier parsing, I am using this RPN convention using a binary or ternary OPERATOR:

           X,Y,...,COMMAND (followed by EOL)

Where the arguments can be either a number, a pair of numbers, a combination of both, or a string (payload data)
Example sniffed strings for a typical communication between gluon 3 (sender) and gluon 4 (receiver):

- Creation of an outlet link:

      [3,0],4,NO   *New link created on outlet 0 of gluon 3, TO gluon 4.*

- Deletion of an outlet link:

      [3,0],4,DO   *Deleting link on outlet 0 of gluon 3, that was connected TO gluon 4*

- Creation of an inlet link:

      3,[4,2],NI   *New link created on inlet 2 of gluon 4, FROM gluon 3*

- Deletion of an inlet link:

      3,[4,2],DI   *Deleting link on inlet 2 of gluon 4, that was connected FROM gluon 3*

- Move of INLET link:

      3,[4,2],DI\n(NEW LINE)**3,[4,1],NI**   *As you can see, there is no need of a new code and the parser will work.*

- A SENT packet:

      [3,0],4,UPDATE+4536+ROT/10/1/15500,S *(parsing the payload is different - I want to use OSC like packets)*

- A RECEIVED packet:

      3,[4,2],UPDATE+4536+ROT/10/1/15500,R

- Connection data:

      3,[4,5],[1,6],[],[4],NET

This means that the message refers to gluon with ID3, and:
-    *The first list contains the nodes to which the (unique) output is connected (here 4 and 5 - there can be up to six connections, or none, in that case it will read []).*

- *The three other lists correspond to inlets 0 to 2, and indicate the nodes to which they are connected (here first inlet is connected to nodes 1 and 6, second inlet to none, and third to 4. Note: maximum number of connections per inlet is 6.*

- Disconnecting ALL inlets/outlets:

  3,CLR_CON (first the index of the gluon)

- Disconnecting all OUTLETS or all INLETS:

  3,CLR_CON_OUT or **3,CLR_CON_IN**