As part of the existing framework there is a remote control object. I originally designed this to take inputs from controllers like joysticks so they could inject currents into neurons. However, it works for a wide variety of sensor systems. It has a generic interface that converts incoming signals into currents for a user specified neuron. I will probably use that as a base and expand it so it can also handle generic outputs. The user will be able to pick a property from any of the objects in the simulation. That property value will be converted using a gain function and then transmitted to the receiver. So you could pick a joint position, a neuron conductance, or in your case the membrane voltage. This will make configuring the system easy and application agnostic. On top of this I will create a new simple serial controller type that will open a serial port and perform the actual serial IO. I suggest the data packets below. Please let me know if you feel it needs modifications. The same packet structure would be used to send data from animatlab to the RIO and back from that to animatlab. Within AnimatLab the user will assign a given data ID to each of the items that are to be transmitted or received. Within the RIO code it would need to have a corresponding mapping. Data will be included in a packet only if it has changed since the last time an update has been sent, and a packet will only be sent if at least one data item must be sent. If there is an error in the transmission where the checksum is incorrect then the receiver will send a Data Error packet. When the original transmitter receives this packet type it will resend all data values regardless of whether they have changed. Data packets will be streaming continuously between AnimatLab and RIO, so I do not believe that an acknowledge packet will be required for each receipt. Let me know if this sounds okay, or if you see any issues. I am trying to wrap something up that I am working on right now, so it will probably be a few weeks before I can get to this. Is that okay?

David

Data Send Packet

|  |  |  |
| --- | --- | --- |
| **Byte[s]** | **Description** | **Value** |
| 0-1 | Header Byte String | 0xFF, 0xFF |
| 2 | Data Send Message ID | 0x01 |
| 3-4 | Total size of message including header, message id, data id's, data value's, checksum, and footer |  |
| 5-6 | Data ID | ID associated with a given part in the GUI editor. This is user configurable. The receiving system will need to know what each ID corresponds to. |
| 7-11 | Data Value (Floating point value) | Data value corresponding to the preceding Data ID |
| ..... | Repeat Data ID and Data Value pairs as required for all data that will be transmitted in this packet |  |
| 12 | Checksum |  |

Data Error Packet

|  |  |  |
| --- | --- | --- |
| **Byte[s]** | **Description** | **Value** |
| 0-1 | Header Byte String | 0xFF, 0xFF |
| 2 | Data Error Message ID | 0x02 |
| 3-4 | Total size of message including header, message id, checksum, and footer |  |
|  |  |  |
| 5 | Checksum |  |
|  |  |  |

https://ssl.gstatic.com/ui/v1/icons/mail/images/cleardot.gif