Serial Communication/R2Protocol Debugging Guide/FAQ

(Work in Progress)

By Stefen Pegels

Documentation for R2Protocol: R2Protocol Documentation
GitHub Link: R2Protocol.h R2Protocol2.py
Working Code Examples (for reference):
From Jetson To arduino:
Arduino Receiving .ino File
Jetson Sending .py File

From Arduino To Jetson:
Arduino Sending .ino File
Jetson Receiving .py File

Common Bugs - See Working Code Examples for Reference

Arduino Resets Itself

- If the serial monitor seems to be rebooting continuously, check for address errors resulting in a segmentation fault in the decode or encode calls. Make sure pointers vs values are used correctly (passing address using "&" or not; look into R2Protocol.h to verify the types of input parameters)
- Segmentation faults can also occur as a result of bad checksums, see **Checksum Returns -1 Errors**

No Data Shows Up Errors

- Check that number of bytes read == number of bytes written. Serial.read() is a blocking call: if 12 bytes are read and only 10 are written, the code will freeze as it waits for two more bytes
- Verify the sender is writing to the serial line - print the data being sent and verify its length to the expected length
- Verify the receiver is reading data - print the bytes as they arrive rather than printing only the decoded output
- Verify you are printing the data from the correct buffer that stored the read data
- Verify hardware TX/RX connections are properly set up

- On Arduino, ensure the correct Serial port is being used and is also initialized - ex **Serial** vs **Serial1** vs **Serial2**.

Data Shows Up as Zeros on Arduino

- Check that **baud rates** match in the serial initialization on **both** Jetson and Arduino
- Make sure that the print statements are **inside** the Serial.available() block. We should only read/decode/print in the cases where Serial.available() is >0. Most of the loop cycles on the arduino will do nothing, when this is written correctly.

Checksum Returns -1 Errors

- Reset input buffer on both devices in the initialization:
  - Arduino: while (Serial.available > 0) Serial.read();
  - Jetson: ser.reset_input_buffer()
- Continuously read the checksum as program runs and reset buffers if -1
- If checksum > 0 but incorrect data, see **Data is Correct at First…**
- Look for discrepancy in buffer sizes on arduino (ex data_buffer is only 4 bytes long but receiving 6 bytes of data)
- Verify number of bytes read == number of bytes written

Data is Correct at First, but then Incorrect Data

- Ensure size field in decode function AND the array length being written to match the message length. This value should be the length of the encoded message minus 16.