

Serial Interface With ADXL362

Now that you have experience with the Serial interface, we will apply the same methodology as we did in session 4 using the input shields. Your tasks are the following:

- Using the provided skeleton/pseudo code, create a sending and receiving sketch.
- In the sending sketch, because the functions provided by the ADXL362 library do a serial print on their own accord, we can't use them directly because they will provide noise when we will want to do a `Serial.write()` with the actual value we want to send.
 - Look in the CPP file using WordPad or NotePad and find the read functions. It's a good practice to explore these libraries in such a manner to discover the mechanics of a library's functions, and what functions it has to begin with.
 - Because the paddle only moves up and down, we can cheat a little bit and only worry about using one axis of movement to provide the input. Thus, we will use the Y axis data.
 - Once you find the function that reads in the Y axis movement, look at how it works and simply mimic what it does to get the Y value (to get a better understanding of what is really going on in these functions, watch the video provided for this session).
 - Implement this in your sending sketch to extract the Yaxis values, and print them to the terminal to ensure its working properly.

- Once you have this working, comment out the print statement and continue to write the sending sketch's code with the guidance of the comments provided in the skeleton code.

- In the receiving sketch, you will need to find a way to synchronize the incoming data with what you are reading it otherwise the buffer that you read in may be incorrect because new data hasn't come in yet, or you read in the data before it all was there. Try to mess with the delay or come up with another way to synchronize the data transmission. Remember, both RX/TX pin sets are connected to each other; the receiver can talk to the sender as well. Use this to your advantage for synchronization.

- Once you have your code written, verify that it works correctly by printing out the read in value to the serial terminal on the receiving side and compare with the values you got when you printed out the values originally on the sending side.

Now that you have this code completed, once again, implement it into a separate function in your PONG game that will be called in the void loop. Use a global variable to store your value and use this value to change the direct of the paddle as you move the accelerometer.