Raspberry (Python)

## **recording** (**Transducer**)

Constantly recieves data from the transducers and sends it to the sending subprogram via Queue.

## StartRecording(Transducers,Polling Rate, Duration)

Sets the parameters for the recording and duration – using the recording subproram and automatically stops recording after a given duration.

## PollConfig(Address)

Goes to the FTP server and downloads the .config file to check if it has been updated and sends the data to the Set config subprogram.

## SetConfig(Config File)

Reads the downloaded config file and then calls the startrecording/pause till start time subprogram if necessary.

## EstablishSocket(Address)

Simply connects to the socket on the server. (TCP socket on X port).

## PauseTillStartTime(Start Time)

Holds the entire program till the start time has been reached and then resumes it to allow accurate data collection. **Sync tim**e

## Sending (bytes,Socket Connection)

Takes data from JSON file and the calculates the size of bytes needed to be send. Which is then sent to the server to ensure all data is sent.

## Jsonify(Data from Transducers,RPI ID)

Combines the multiple transducer’s data into one JSON file along with the raspberrypi’s identifier

## File Writing Thread ( Live Data ) {Either Server or Raspberry| Undecided}

Server (Python)

## SetupConnections(Socket)

Connects to all raspberries with a message from them that denotes their ID. Client computer connects and sends message to show that it is a client.

## **GetTransducerCount**(Socket,data)

Program that is run during setupconnections to find the amount of tranducers being dealt with.

## ClientConnectionThread(Queue,Instruction Queue)

Takes packaged JSONdata- converts to bytes and uses the sending data subprogram to send data to the client.

## RpiConnectionThread(Queue,Instruction Queue)

Gets raw bytes from the raspberry and sends to a Queue which is then unpacked in the unpackJSON subprogram.

## AlignJson(JsonQueue,TransducerCount)

Aligns the JSON files by sampleTime. (e.g. all 0-100 samples from every transducer is aligned) and then sent to packaging thread to be packaged. SORT BY SAMPLE TIME – ONCE the first {transducercount} are all the same sample time then we send to packaging thread.

## SortJSONData(Jsondata)

Sort the array of jsondata by sample number (i.e. which sample it currently is )

return to alignJSON

## UnpackJSON(Raw Bytes)

Converts raw JSON bytes into JSON object and puts into secondary JSON Queue.

## PackagingThread(Array with JSON data)

Combines all the different JSON files from a

## RecievingData(Socket)

Get number of bytes to be recieved and then recieve message and return

## GenerateStartRecordingTime()?

# SendingData(**Bytes,socket**)

Sends a byte code first then the data

Client Comupter (Python)

## RecievingData(Socket)

Get number of bytes to be recieved and then recieve message and return

# SendingData(**Bytes,socket**)

Sends a byte code first then the data

# **InitializeWithServer**(**Bytes,socket**)

Handshake with server using sendingdata program

**ConnecttoLabview**(**Bytes,socket**)

TBC

Client Computer (LabView)