AeroPy Documentation:

Last Edit: Chan

Date: 5/26/20

**Trigno Communications:**

**Connecting to the Trigno Base:**

public void ValidateBase(string key, string license,string dataType):

Initial call to the Trigno Base. Sets up a connection to the base using the user’s key and license strings. Initializes the Trigno Base in the desired data source configuration given by the dataType variable (e.g “RF”,”sim”)

**Connection and Searching for Sensor Functions (RF connection):**

public System.Threading.Tasks.Task ScanSensors():

Main method for beginning sensor scan. Will scan for unpaired sensors via RF

public void PairSensors():

This sets the base into pairing mode, allowing for a user to pair a new sensor to the base to be found in the scan. Call scan after pairing to ensure that pipeline is set up properly for streaming

public bool ConnectSensors():

Connects and initializes all the sensors that have been currently found in the scan

If you only want to connect a specific sensor, use SelectSensor method

public void SelectSensors(int sensorNum):

Selects and initializes the sensor at index sensorNum

**Sensor Streaming(RF connection):**

public void ClearSensorList():

Clears the internal list of found sensors, call this first to refresh internal state before data streaming

public void AddSensortoList(DelsysAPI.Components.TrignoRf.SensorTrignoRf SelectedSensor):

Adds a sensor the internal list to give to stream data. Use the GetSensorObject() function call to get the internal TrignoRf sensor object of the desired sensor. See example code for more information.

public DelsysAPI.Components.TrignoRf.SensorTrignoRf GetSensorObject(int sensorNo):

Returns the sensor object of the sensor at the index sensorNo

public DelsysAPI.Transforms.Transform CreateTransform(string type):

Method to create a basic transform to be passed into streamData. This returns a DelsysApi Transform object to be passed into the stream data function call.

public void StreamData(List<int> index, DelsysAPI.Transforms.Transform newTransform, int frameThroughput):

This calls the stream data function. It will stream data from the sensor indices passed in index at a frame throughput specified by the user, default is set to 2 frames. This tells the Trigno base to begin streaming data to the data buffers. Use CheckDataQueue() function to determine if new data is available and PollData() function to retrieve the data.

public void StopData():

Tells the Trigno base to stop the data stream, and clean up the data pipeline after data streaming has completed.

**Data Management (RF connection):**

public bool CheckDataQueue():

This returns a flag that denotes whether there is new data in the internal data buffer that is ready to be extracted. Use the PollData() function to return the data structure.

public List<List<double>>[] PollData():

This retrieves the data from the data buffer fed by the StreamData data stream. Returns data the form of packets, length denoted by the currently set sample mode of the sensor. Each row in the data packet contains the returned data for a different channel in the sensor (eg. EMG, IMU, ACC, GYRO, etc.).

**Helper Functions (RF connection):**

public void SetDataLog(bool dataSave):

This function sets a flag that tells the base whether to save the data at the end of collection or not. True being to save the data output, and False to not.

public bool GetDataLog():

Returns the current state of the DataLog Bool

public void SetOutputDir(string outputDir):

Sets the output directory for data to be saved to

public string GetOutputDir():

Returns the current output directory

public int GetTotalPackets():

Returns the total number of data packets collected from the current streaming session

public List<string> GetSampleMode():

Getter for the current samplemodes of the sensors

public void SetSampleMode(int componentNum, string sampleMode)

Sets the sample mode for the sensors. Will set the sensor at index componentNum to the mode given by sampleMode

public string[] ListSensorNames():

This will return a string array of the current sensors found

public string[] ListSensorModes(int sensorSelected):

This will list out the list of different sensor modes for the sensor at index sensorSelected

**Trigno Communications (Simulated Data):**

public void LoadSensorFromShpf(string fileLoc)

Loads in the .shpf file data from the file path given by the variable “fileLoc”. This function basically mimics the scan, and connect functions used in the RF data source.

public void StreamSim()

Begins the simulated data source stream. Data will be streamed as if it is being collected in real time. Data is polled for and retrieved in the same method used in the RF data collection pipeline, i.e CheckDataQueue() and PollData().

public List<List<double>> GetSimData()

This retrieves all of the data at once from the .shpf file. Useful if the user wants the data immediately instead of simulating it.

public bool CheckStreamStatus()

This returns a Boolean that indicates the current status of the simulated data stream. When the simulated data stream is finished, it will return True, otherwise, if the data stream is still running, it will return False.

public List<string> GetSimFrequencies()

This returns a list of strings of the returned simulated data stream comprised of the sensor type and the sampling frequency, separated by a colon. (E.G [“EMG:2222.22”])