The TcpListener is a C language script written to automatically listen to the sensor nodes at the automatic weather station and insert the data into a database table on the wimea server or any server as specified in config.h file.

The sensor nodes broadcasts data every minute which means that a lot of data is captured by this script and then inserted into the table. This program runs as a daemon process on the server listening on the TCP port set using the config.h file.

In order to send data to the database using C-programming, MySQL-c-connector must be installed on the computer where this program runs. This is mostly recommended for Linux users since it is simple to setup and configure as compared to windows user.

There are 3 major files involved in the process of reading data from the sensor nodes and then insert it into the database. These are tcpListener.c ,nodes.h and config.h

“nodes.h” contains parameters from each node and the keys used to identify them. It defines the parameters attached the TXT variable eg 10m, 2m, gnd, sink etc in the sensors.dat file which uniquely identifies a given node on the station. It's REQUIRED that the TXT value from the node contains 10m or 2m or gnd or sink string depending on the node. For instance TXT=makg\_10m or TXT=10m-Makg2 to represent the 10m node. In case different keys are used to report sensor values, they should be updated in nodes.h

“tcpListener.c” listens on port 10020 or any specified and receives a string sent by a particular node on the weather station, processes it, sorts and filters it ,and then sends the extracted data it to the database table(s) on the wimea server for more processing and analysis.

Below is the flow of the program.

Algorithm for the TCP Listener in C

1. Start

2. Fork a child process

3. If fork is successful

3.1. terminate the parent

4. create a socket

5. wait for any connection from sensd or any client

6. If connection is successful

6.1. Accept connection

7. While true

7.1. receive string record from node

7.2. determine the node the sent the string

7.3. Extract the data values it has

7.4. Store node status information into the database [nodestatus]

7.5. Store the data from the attached sensors into the database [observationslip]

7.5. Log the SQL query into the file