# EECS 113
# Final Project Report


| Members: | Student ID#: |
|---|---|
| Ziqi Ding | 82177304 |
| Zhengyi Zhu | 86499703 |
| Devin Quoc-An Pham | 96005056 |
| Amauri Villegas Garcia | 14366956 |


6/14/2019

Group 13

## Introduction:

In this assignment, our group was tasked to create a Weather Monitoring System with an IoT device based off of Raspberry Pi. We achieved this by first connecting a temperature and humidity sensor to record the local temperature and humidity every minute, generate hourly averages of those values and displayed them on the LCD module. Then, we used an API to retrieve CIMIS values from the local Irvine weather station and compared those values to our own in order to determine the amount of water to be “irrigated” via a relay. Lastly, after determining the time it takes to “irrigate” the water, we used a PIR sensor to sense motion and turn off the “irrigation system” and resume when there’s no motion until the time that we calculated for the relay has been reached.

## Contribution:

Ziqi Ding: Setup, Motion Sensor Module, Code Debugging, 5V Relay Module, Lab Report
Zhengyi Zhu: Setup, Motion Sensor Module, Code Debugging
Devin Quoc-An Pham: Setup, Motion Sensor Module, 5V Relay Module, Video Presentation, Lab Report
Amauri Villegas Garcia: General project structure design, Setup, DHT 11 Module, LCD 1602 Display Module, Cimis api connection and parsing, video recorder

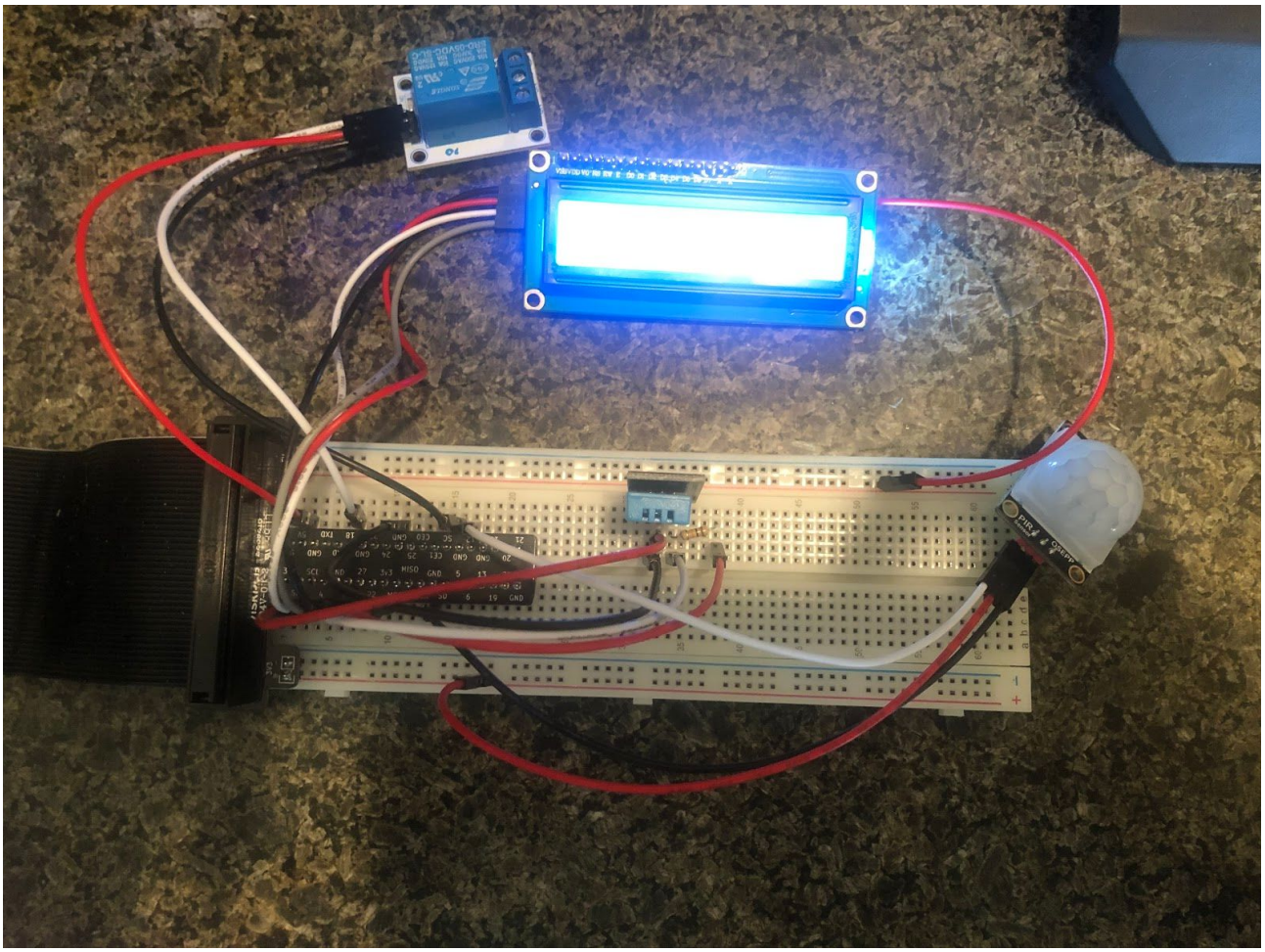
## Procedure:

Our final project setup consisted of four peripherals connected to a Raspberry Pi Model 3 B+ and controlled by it: DHT11 Module (Temp & Humidity Sensor), Passive Infrared (PIR) sensor, LCD1602 Display Module, and a 5v Relay Module (see photo below)

## Setup:

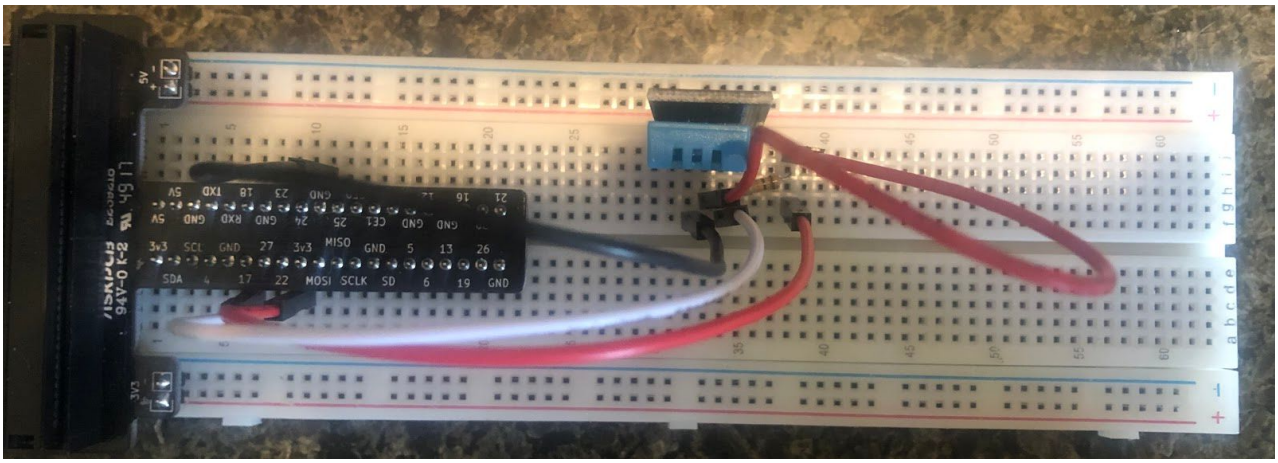
Wire Colors:

- Red (source/power)
- Black (ground)
- White (signal) or SDA for LCD Display Module
- Gray (SCL for LCD)



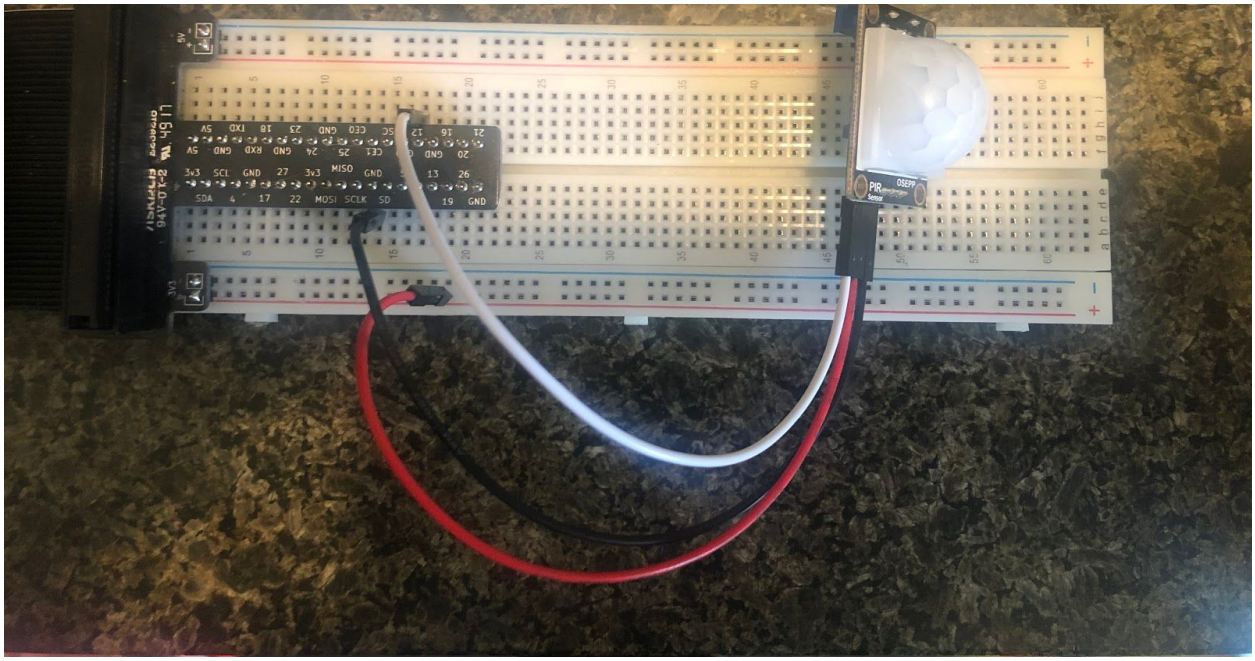
**DHT11 Module:**

The purpose of the DHT11 module is to read the temperature and humidity values when asked by the program and operates between 3.3V - 5.5V. For our setup, we connected the power ($V_{CC}$) and GND to their respective “3v3” and “GND” on the GPIO extension board and connected the signal wire to “GPIO 17” (pin 11). We also connected $V_{CC}$ in parallel with the signal wire and included a 10kΩ resistor in series with the $V_{CC}$ branch.



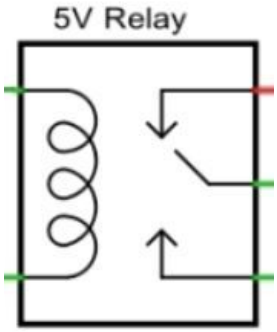
**Passive Infrared (PIR) Sensor:**

The PIR sensor passively waits and tracks for movement. When movement is detected, the action is indicated by the signal cable and an LED will light up on the back of the module. We setup the PIR module with $V_{CC}$ connected to “3v3” and GND to “GND” and our signal to “GPIO 12” (pin 32).

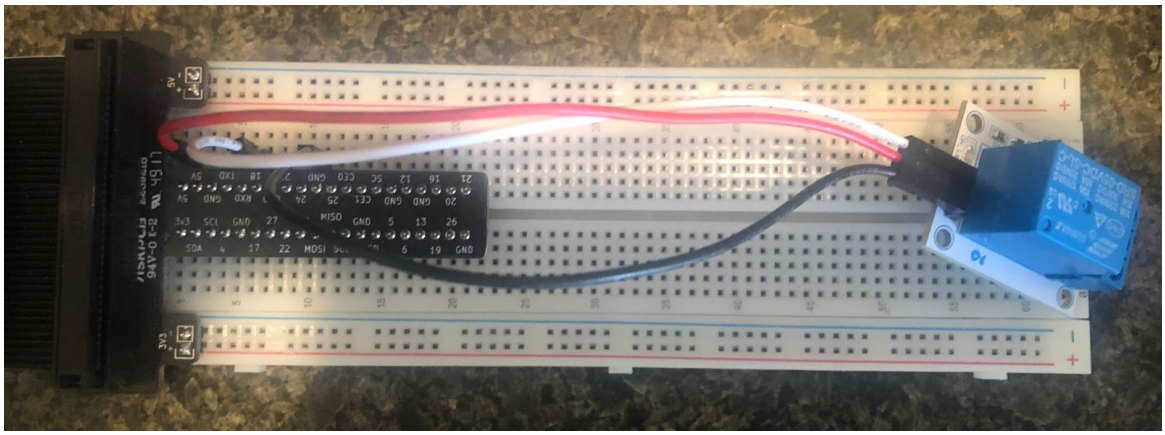


**5V Relay Module:**

The 5V relay is designed where a connection in the inputs will activate an electromagnet that moves a switch on the other side to make a closed connection (see schematic provided).

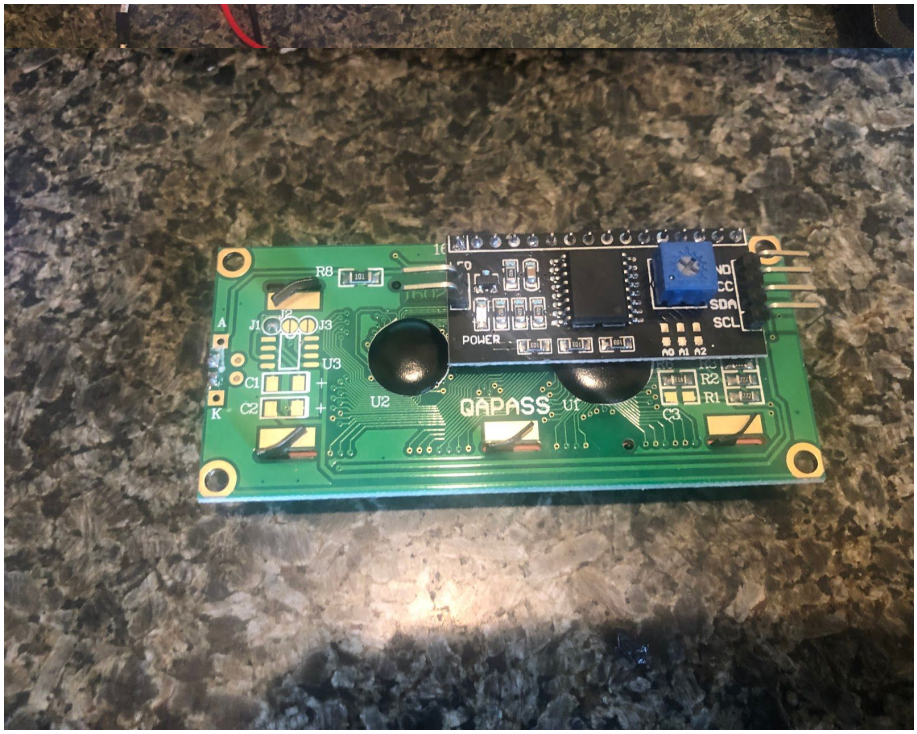


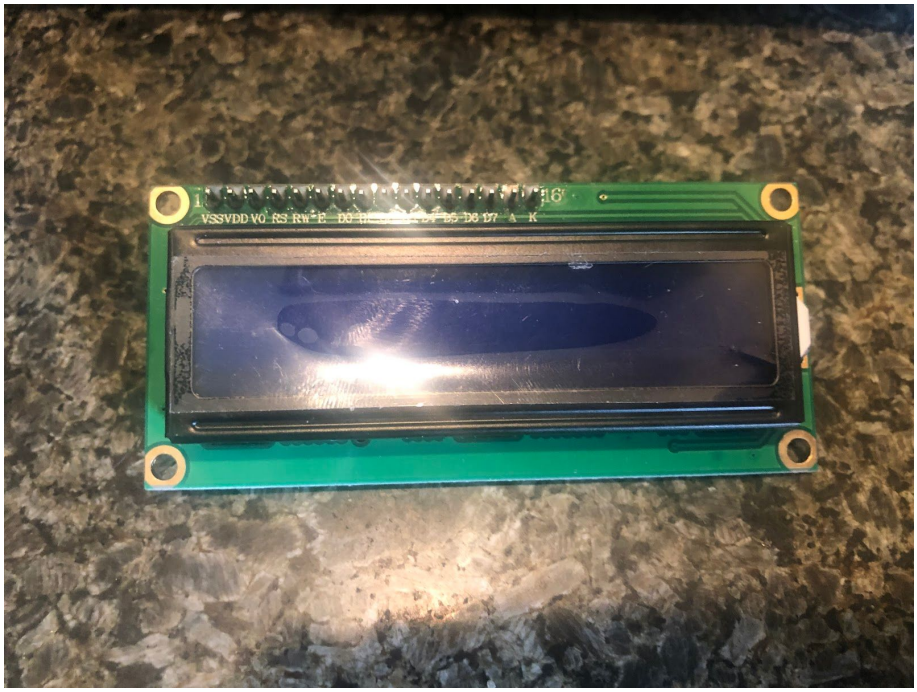

Very similarly to the other modules, we set up or V and GND to “5V” and “GND” respectively and connected the signal cable to “GPIO 18” (pin 12)



**LCD1602 Display Module:**

The LCD display module serves to output values on a screen based on the commands of the program. This specific module used for our project came with an I2C interface soldered to the back of the module, which makes operating the LCD module more simple and easy. We connected our VCC and GND to their respective “5V” an “GND” on the GPIO Board and connected the SDA (white wire) and SCL (gray wire) to “SDA” and “SCL” on the board respectively as well.



**Implementation:**

Our code for our project consisted of multiple functions called by our main function when needed. Each individual function serves a specific purpose to the setup as a whole

**def call_Cimis_Api():**

This function served as an API to retrieve data from the local weather station such as hourly temps and humidity along with $ET_O$ values using the url. The program will also indicate whether the retrieval of the CIMIS data was successful or not

**def parse_cimisdict():**

This function is called to parse the dictionary obtained from cimis api response called. It reads data for HlyRelHum (humidity), HlyAirTmp (temperature), and HlyEto (Eto) from within the dictionary and it populates three list, one for humidity values, one for temperature values and one for Eto values. It populates each list as long as each entry in the dictionary has data, if the entry is equal to None, null it doesn't add those entries to the list

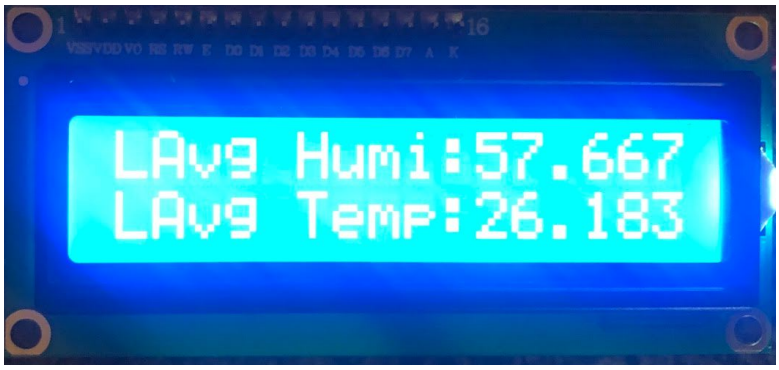
**def get_Humidity_avg():**

This function serves to take all of the humidity values from the list of humidity values read by the DHT11 module and returns and prints the average humidity value to LCD (see LCD below)

**def get_Temp_avg():**

Very similarly to the above function, this one will instead take in all of the temperature values read by the DHT11 module in the list and return and print an average temperature value to LCD (see LCD below)

**Average Humidity and Temperatures on LCD:**



**def readTempandHum():**

This function is called to read the values from our DTH11 sensor, if there is no error and we are able get a reading then we populate a List for temperature and for Humidity which values are eventually average every x interval.

**def destroy():**

This function is served to clear the LCD of any values so that when it's called, it will allow the program to display different values

**def calculate():**

This function serves as a formula to return the amount of time the irrigation system needs to run based on the formula provided by the assignment PDF. After calculating the value with the provided formula, we converted the value from gallons/day to seconds based on the rate of the irrigation system per hour and the amount of seconds in a day. (see LCD below)

**def EToValue():**

This function throws in all of the temperature and humidity along with the $ET_O$ values from our setup and the CIMIS API and used them to calculate the local $ET_O$ based on the formula provided by the project PDF (see LCD below)

**Local ETO Value and Irrigation Time on LCD:**


**def displayHomescreen():**

This function will control the LCD to blink "Weather Station sampling data" every second to indicate that the program is working on retrieving data and values until the minute and hour is up

**LCD Displaying Home Screen:**


**def irrigation():**

This function will use event detection for the PIR sensor and call the loop() function to implement motion detection to turn on and off irrigation

**def loop():**

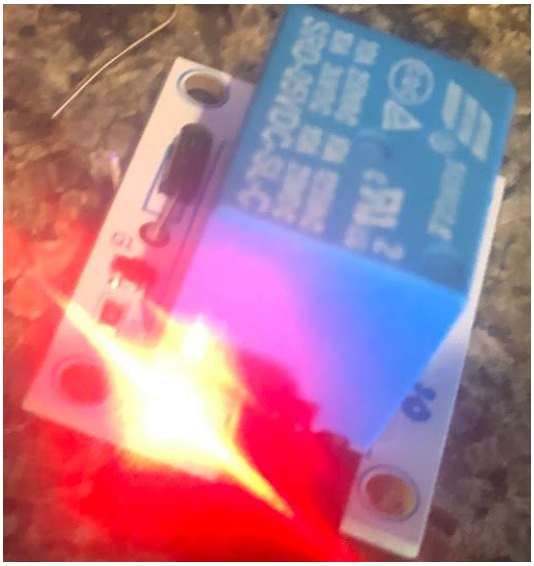
This function is called when there are changes to the signal from the PIR sensor. The function will turn off the relay when no movement is detected and turn it off when movement is detected. In addition, the relay will be turned on if the PIR sensor has been detecting movement for more than 1 minute (60 s) when the relay is turned off, the function will print on the LCD display “No movement detected. Resuming irrigation” or “Resuming irrigation” when the PIR sensor detects movement for over a minute. When the relay is on, the LCD will print “Movement detected. Irrigation Off”.  (see LCD images below)

**When PIR Senses Motion:**

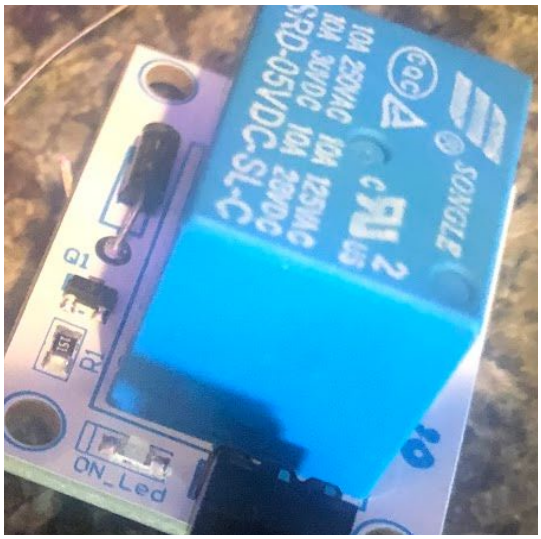

**When PIR Doesn’t Sense Motion:**


**When PIR Has Been Sensing Motion Over a Minute:**


**Relay is On (LED ON) When Motion is Detected:**



**Relay is Off (LED OFF) When no Motion Detected or PIR Senses Movement for Over a Minute:**



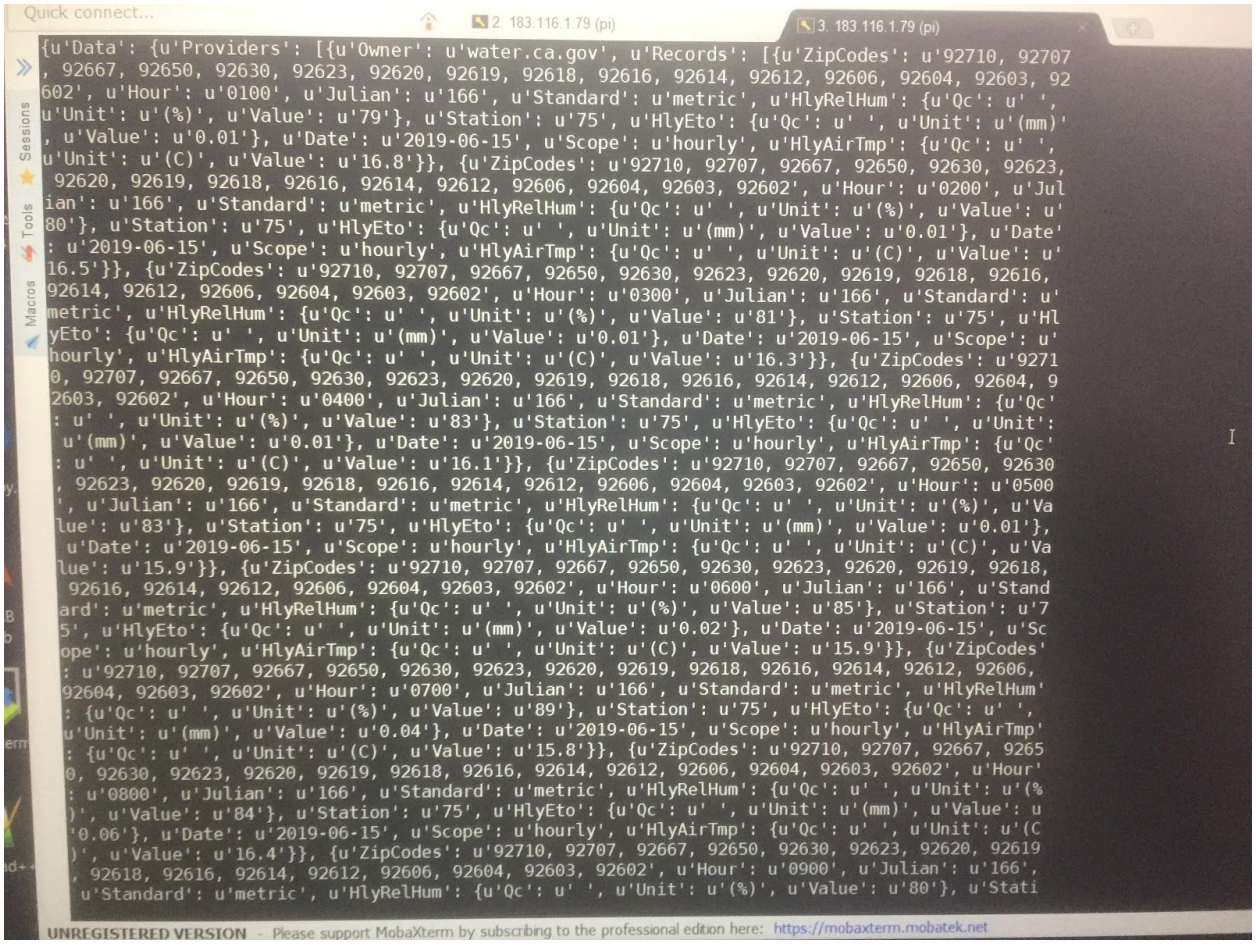

## Data:

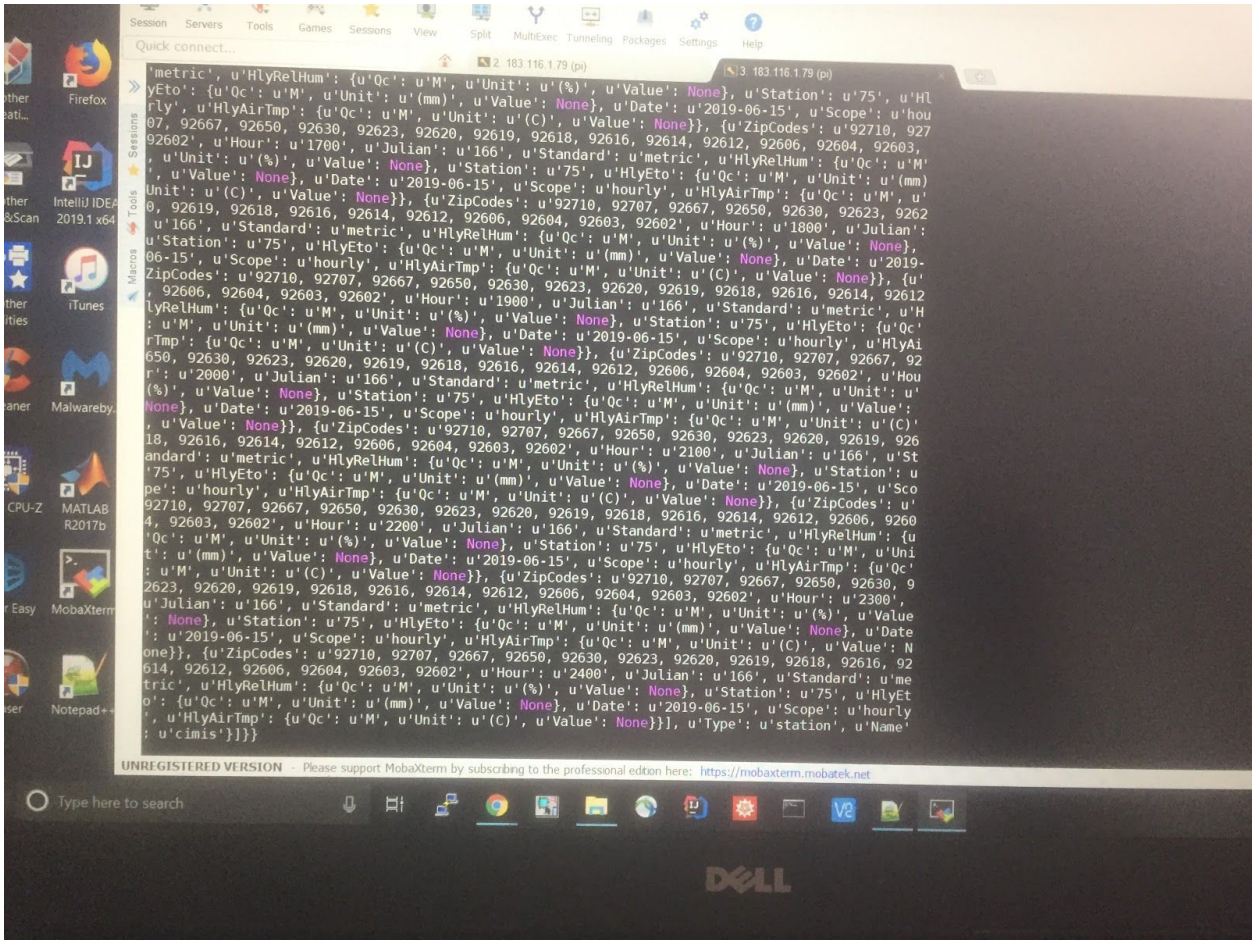
Following lines is the response obtained from cimis api request and saved in a file.txt:

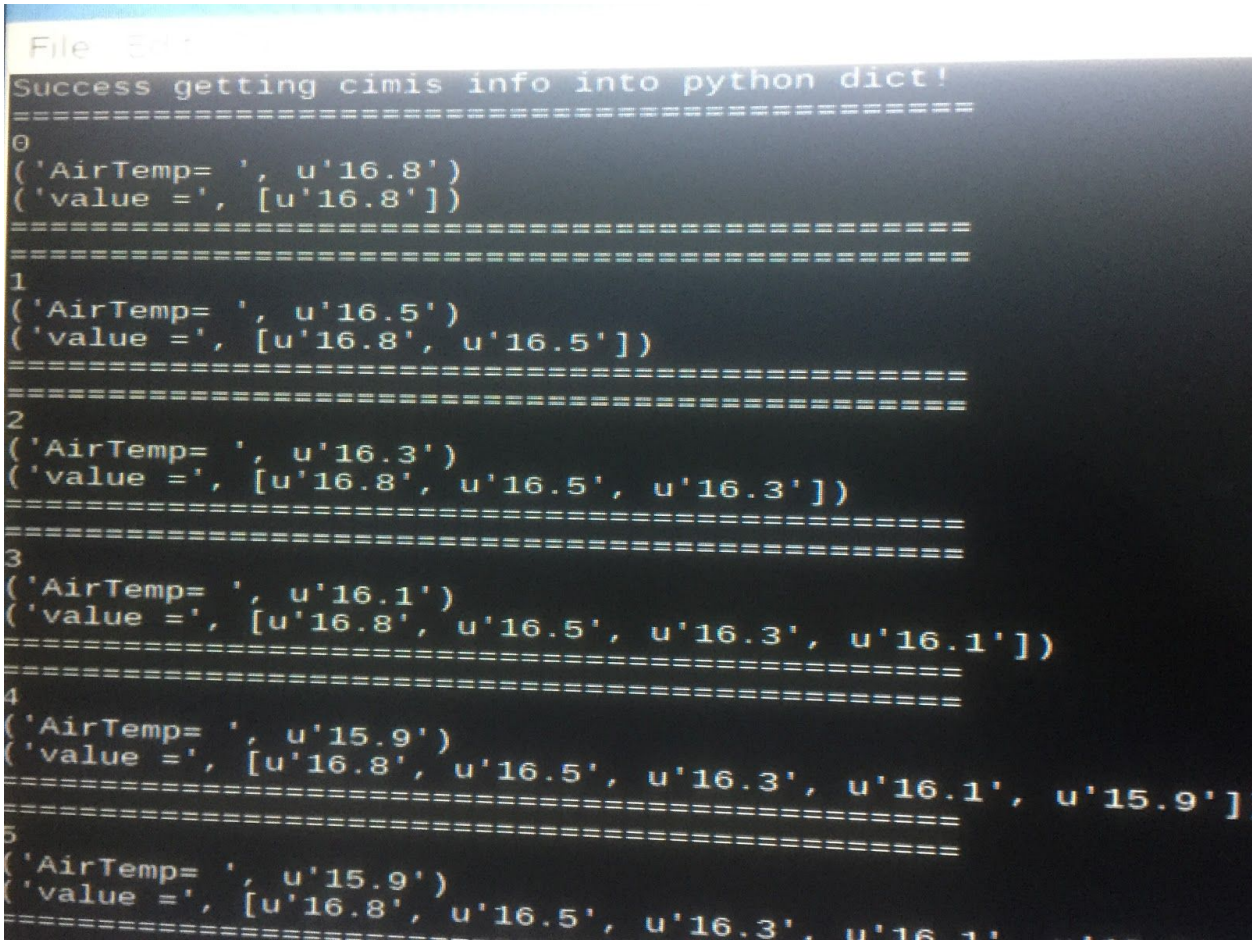
{"Data": {"Providers": [{"Owner": "water.ca.gov", "Records": [{"ZipCodes": "92710, 92707, 92667, 92650, 92630, 92623, 92620, 92619, 92618, 92616, 92614, 92612, 92606, 92604, 92603, 92602", "Hour": "0100", "Julian": "165", "Standard": "metric", "HlyRelHum": {"Qc": " ", "Unit": "(%)", "Value": "85"}, "Station": "75", "HlyEto": {"Qc": " ", "Unit": "(mm)", "Value": "0"}, "Date": "2019-06-14", "Scope": "hourly", "HlyAirTmp": {"Qc": " ", "Unit": "(C)", "Value": "16.6"}}, {"ZipCodes": "92710, 92707, 92667, 92650, 92630, 92623, 92620, 92619, 92618, 92616, 92614, 92612, 92606, 92604, 92603, 92602", "Hour": "0200", "Julian": "165", "Standard": "metric", "HlyRelHum": {"Qc": " ", "Unit": "(%)", "Value": "85"}, "Station": "75", "HlyEto": {"Qc": " ", "Unit": "(mm)", "Value": "0"}, "Date": "2019-06-14", "Scope": "hourly", "HlyAirTmp": {"Qc": " ", "Unit": "(C)", "Value": "16.6"}}, {"ZipCodes": "92710, 92707, 92667, 92650, 92630, 92623, 92620, 92619, 92618, 92616, 92614, 92612, 92606, 92604, 92603, 92602", "Hour": "0300", "Julian": "165", "Standard": "metric", "HlyRelHum": {"Qc": " ", "Unit": "(%)", "Value": "84"}, "Station": "75", "HlyEto": {"Qc": " ", "Unit": "(mm)", "Value": "0"}, "Date": "2019-06-14", "Scope": "hourly", "HlyAirTmp": {"Qc": " ", "Unit": "(C)", "Value": "16.6"}}, {"ZipCodes": "92710, 92707, 92667, 92650, 92630, 92623, 92620, 92619, 92618, 92616, 92614, 92612, 92606, 92604, 92603, 92602", "Hour": "0400", "Julian": "165", "Standard": "metric", "HlyRelHum": {"Qc": " ", "Unit": "(%)", "Value": "80"}, "Station": "75", "HlyEto": {"Qc": " ", "Unit": "(mm)", "Value": "0.01"}, "Date": "2019-06-14", "Scope": "hourly", "HlyAirTmp": {"Qc": " ", "Unit": "(C)", "Value": "16.9"}}, {"ZipCodes": "92710, 92707, 92667, 92650, 92630, 92623, 92620, 92619, 92618, 92616, 92614, 92612, 92606, 92604, 92603, 92602", "Hour": "0500", "Julian": "165", "Standard": "metric", "HlyRelHum": {"Qc": " ", "Unit": "(%)", "Value": "77"}, "Station": "75", "HlyEto": {"Qc": " ", "Unit": "(mm)", "Value": "0.02"}, "Date": "2019-06-14", "Scope": "hourly", "HlyAirTmp": {"Qc": " ", "Unit": "(C)", "Value": "16.8"}}, {"ZipCodes": "92710, 92707, 92667, 92650, 92630, 92623, 92620, 92619, 92618, 92616, 92614, 92612, 92606, 92604, 92603, 92602", "Hour": "0600", "Julian": "165", "Standard": "metric", "HlyRelHum": {"Qc": " ", "Unit": "(%)", "Value": "76"}, "Station": "75", "HlyEto": {"Qc": " ", "Unit": "(mm)", "Value": "0.03"}, "Date": "2019-06-14", "Scope": "hourly", "HlyAirTmp": {"Qc": " ", "Unit": "(C)", "Value": "16.9"}}, {"ZipCodes": "92710, 92707, 92667, 92650, 92630, 92623, 92620, 92619, 92618, 92616, 92614, 92612,

92606, 92604, 92603, 92602", "Hour": "0700", "Julian": "165", "Standard": "metric", "HlyRelHum": {"Qc": " ", "Unit": "(%)", "Value": "74"}, "Station": "75", "HlyEto": {"Qc": " ", "Unit": "(mm)", "Value": "0.06"}, "Date": "2019-06-14", "Scope": "hourly", "HlyAirTmp": {"Qc": " ", "Unit": "(C)", "Value": "17.1"}}, {"ZipCodes": "92710, 92707, 92667, 92650, 92630, 92623, 92620, 92619, 92618, 92616, 92614, 92612, 92606, 92604, 92603, 92602", "Hour": "0800", "Julian": "165", "Standard": "metric", "HlyRelHum": {"Qc": " ", "Unit": "(%)", "Value": "72"}, "Station": "75", "HlyEto": {"Qc": " ", "Unit": "(mm)", "Value": "0.09"}, "Date": "2019-06-14", "Scope": "hourly", "HlyAirTmp": {"Qc": " ", "Unit": "(C)", "Value": "17.3"}}, {"ZipCodes": "92710, 92707, 92667, 92650, 92630, 92623, 92620, 92619, 92618, 92616, 92614, 92612, 92606, 92604, 92603, 92602", "Hour": "0900", "Julian": "165", "Standard": "metric", "HlyRelHum": {"Qc": " ", "Unit": "(%)", "Value": "71"}, "Station": "75", "HlyEto": {"Qc": " ", "Unit": "(mm)", "Value": "0.1"}, "Date": "2019-06-14", "Scope": "hourly", "HlyAirTmp": {"Qc": " ", "Unit": "(C)", "Value": "17.9"}}, {"ZipCodes": "92710, 92707, 92667, 92650, 92630, 92623, 92620, 92619, 92618, 92616, 92614, 92612, 92606, 92604, 92603, 92602", "Hour": "1000", "Julian": "165", "Standard": "metric", "HlyRelHum": {"Qc": " ", "Unit": "(%)", "Value": "74"}, "Station": "75", "HlyEto": {"Qc": " ", "Unit": "(mm)", "Value": "0.09"}, "Date": "2019-06-14", "Scope": "hourly", "HlyAirTmp": {"Qc": " ", "Unit": "(C)", "Value": "17.7"}}, {"ZipCodes": "92710, 92707, 92667, 92650, 92630, 92623, 92620, 92619, 92618, 92616, 92614, 92612, 92606, 92604, 92603, 92602", "Hour": "1100", "Julian": "165", "Standard": "metric", "HlyRelHum": {"Qc": " ", "Unit": "(%)", "Value": "75"}, "Station": "75", "HlyEto": {"Qc": " ", "Unit": "(mm)", "Value": "0.13"}, "Date": "2019-06-14", "Scope": "hourly", "HlyAirTmp": {"Qc": " ", "Unit": "(C)", "Value": "17.9"}}, {"ZipCodes": "92710, 92707, 92667, 92650, 92630, 92623, 92620, 92619, 92618, 92616, 92614, 92612, 92606, 92604, 92603, 92602", "Hour": "1200", "Julian": "165", "Standard": "metric", "HlyRelHum": {"Qc": " ", "Unit": "(%)", "Value": "73"}, "Station": "75", "HlyEto": {"Qc": " ", "Unit": "(mm)", "Value": "0.39"}, "Date": "2019-06-14", "Scope": "hourly", "HlyAirTmp": {"Qc": " ", "Unit": "(C)", "Value": "19"}}, {"ZipCodes": "92710, 92707, 92667, 92650, 92630, 92623, 92620, 92619, 92618, 92616, 92614, 92612, 92606, 92604, 92603, 92602", "Hour": "1300", "Julian": "165", "Standard": "metric", "HlyRelHum": {"Qc": " ", "Unit": "(%)", "Value": "72"}, "Station": "75", "HlyEto": {"Qc": " ", "Unit": "(mm)", "Value": "0.35"}, "Date": "2019-06-14", "Scope": "hourly", "HlyAirTmp": {"Qc": " ", "Unit": "(C)", "Value": "19.5"}}, {"ZipCodes": "92710, 92707, 92667, 92650, 92630, 92623, 92620, 92619, 92618, 92616, 92614, 92612, 92606, 92604, 92603, 92602", "Hour": "1400", "Julian": "165", "Standard": "metric", "HlyRelHum": {"Qc": " ", "Unit": "(%)", "Value": "67"}, "Station": "75", "HlyEto": {"Qc": " ", "Unit": "(mm)", "Value": "0.55"}, "Date": "2019-06-14", "Scope": "hourly", "HlyAirTmp": {"Qc": " ", "Unit": "(C)", "Value": "20.8"}}, {"ZipCodes": "92710, 92707, 92667, 92650, 92630, 92623, 92620, 92619, 92618, 92616, 92614, 92612, 92606, 92604, 92603, 92602", "Hour": "1500", "Julian": "165", "Standard": "metric", "HlyRelHum": {"Qc": " ", "Unit": "(%)", "Value": "64"}, "Station": "75", "HlyEto": {"Qc": " ", "Unit": "(mm)", "Value": "0.55"}, "Date": "2019-06-14", "Scope": "hourly", "HlyAirTmp": {"Qc": " ", "Unit": "(C)", "Value": "21.7"}}, {"ZipCodes": "92710, 92707, 92667, 92650, 92630, 92623, 92620, 92619, 92618, 92616, 92614, 92612, 92606, 92604, 92603, 92602", "Hour": "1600", "Julian": "165", "Standard": "metric", "HlyRelHum": {"Qc": " ", "Unit": "(%)", "Value": "61"}, "Station": "75", "HlyEto": {"Qc": " ", "Unit": "(mm)", "Value": "0.45"}, "Date": "2019-06-14", "Scope": "hourly", "HlyAirTmp": {"Qc": " ", "Unit": "(C)", "Value": "22.2"}}, {"ZipCodes": "92710, 92707, 92667, 92650, 92630, 92623, 92620, 92619, 92618, 92616, 92614, 92612, 92606, 92604, 92603, 92602", "Hour": "1700", "Julian": "165", "Standard": "metric", "HlyRelHum": {"Qc": " ", "Unit": "(%)", "Value": "63"}, "Station": "75", "HlyEto": {"Qc": " ", "Unit": "(mm)", "Value": "0.32"}, "Date": "2019-06-14", "Scope": "hourly", "HlyAirTmp": {"Qc": " ", "Unit": "(C)", "Value": "21.3"}}, {"ZipCodes": "92710, 92707, 92667, 92650, 92630, 92623, 92620, 92619, 92618, 92616, 92614, 92612, 92606, 92604, 92603, 92602", "Hour": "1800", "Julian": "165", "Standard": "metric", "HlyRelHum": {"Qc": " ", "Unit": "(%)", "Value": "66"}, "Station": "75", "HlyEto": {"Qc": " ", "Unit": "(mm)", "Value": "0.15"}, "Date": "2019-06-14", "Scope": "hourly", "HlyAirTmp": {"Qc": " ", "Unit": "(C)", "Value": "20.2"}}, {"ZipCodes": "92710, 92707, 92667, 92650, 92630, 92623, 92620, 92619, 92618, 92616, 92614, 92612, 92606, 92604, 92603, 92602", "Hour": "1900", "Julian": "165", "Standard": "metric", "HlyRelHum": {"Qc": " ", "Unit": "(%)", "Value": "73"},

"Station": "75", "HlyEto": {"Qc": " ", "Unit": "(mm)", "Value": "0.03"}, "Date": "2019-06-14", "Scope": "hourly", "HlyAirTmp": {"Qc": " ", "Unit": "(C)", "Value": "18.2"}}, {"ZipCodes": "92710, 92707, 92667, 92650, 92630, 92623, 92620, 92619, 92618, 92616, 92614, 92612, 92606, 92604, 92603, 92602", "Hour": "2000", "Julian": "165", "Standard": "metric", "HlyRelHum": {"Qc": " ", "Unit": "(%)", "Value": "78"}, "Station": "75", "HlyEto": {"Qc": " ", "Unit": "(mm)", "Value": "0"}, "Date": "2019-06-14", "Scope": "hourly", "HlyAirTmp": {"Qc": " ", "Unit": "(C)", "Value": "17.3"}}, {"ZipCodes": "92710, 92707, 92667, 92650, 92630, 92623, 92620, 92619, 92618, 92616, 92614, 92612, 92606, 92604, 92603, 92602", "Hour": "2100", "Julian": "165", "Standard": "metric", "HlyRelHum": {"Qc": " ", "Unit": "(%)", "Value": "79"}, "Station": "75", "HlyEto": {"Qc": " ", "Unit": "(mm)", "Value": "0"}, "Date": "2019-06-14", "Scope": "hourly", "HlyAirTmp": {"Qc": " ", "Unit": "(C)", "Value": "17.1"}}, {"ZipCodes": "92710, 92707, 92667, 92650, 92630, 92623, 92620, 92619, 92618, 92616, 92614, 92612, 92606, 92604, 92603, 92602", "Hour": "2200", "Julian": "165", "Standard": "metric", "HlyRelHum": {"Qc": " ", "Unit": "(%)", "Value": "79"}, "Station": "75", "HlyEto": {"Qc": " ", "Unit": "(mm)", "Value": "0"}, "Date": "2019-06-14", "Scope": "hourly", "HlyAirTmp": {"Qc": " ", "Unit": "(C)", "Value": "17.3"}}, {"ZipCodes": "92710, 92707, 92667, 92650, 92630, 92623, 92620, 92619, 92618, 92616, 92614, 92612, 92606, 92604, 92603, 92602", "Hour": "2300", "Julian": "165", "Standard": "metric", "HlyRelHum": {"Qc": " ", "Unit": "(%)", "Value": "78"}, "Station": "75", "HlyEto": {"Qc": " ", "Unit": "(mm)", "Value": "0"}, "Date": "2019-06-14", "Scope": "hourly", "HlyAirTmp": {"Qc": " ", "Unit": "(C)", "Value": "17.3"}}, {"ZipCodes": "92710, 92707, 92667, 92650, 92630, 92623, 92620, 92619, 92618, 92616, 92614, 92612, 92606, 92604, 92603, 92602", "Hour": "2400", "Julian": "165", "Standard": "metric", "HlyRelHum": {"Qc": "M", "Unit": "(%)", "Value": null}, "Station": "75", "HlyEto": {"Qc": "M", "Unit": "(mm)", "Value": null}, "Date": "2019-06-14", "Scope": "hourly", "HlyAirTmp": {"Qc": "M", "Unit": "(C)", "Value": null}}], "Type": "station", "Name": "cimis"}]}}


Quick connect...
2. 183.116.1.79 (pi)
3. 183.116.1.79 (pi)
Sessions
Tools
Macros
{u'Data': {u'Providers': [{u'Owner': u'water.ca.gov', u'Records': [{u'ZipCodes': u'92710, 92707
, 92667, 92650, 92630, 92623, 92620, 92619, 92618, 92616, 92614, 92612, 92606, 92604, 92603, 92
602', u'Hour': u'0100', u'Julian': u'166', u'Standard': u'metric', u'HlyRelHum': {u'Qc': u' ',
u'Unit': u'(%)', u'Value': u'79'}, u'Station': u'75', u'HlyEto': {u'Qc': u' ', u'Unit': u'(mm)'
, u'Value': u'0.01'}, u'Date': u'2019-06-15', u'Scope': u'hourly', u'HlyAirTmp': {u'Qc': u' ',
u'Unit': u'(C)', u'Value': u'16.8'}}, {u'ZipCodes': u'92710, 92707, 92667, 92650, 92630, 92623,
92620, 92619, 92618, 92616, 92614, 92612, 92606, 92604, 92603, 92602', u'Hour': u'0200', u'Jul
ian': u'166', u'Standard': u'metric', u'HlyRelHum': {u'Qc': u' ', u'Unit': u'(%)', u'Value': u'
80'}, u'Station': u'75', u'HlyEto': {u'Qc': u' ', u'Unit': u'(mm)', u'Value': u'0.01'}, u'Date'
: u'2019-06-15', u'Scope': u'hourly', u'HlyAirTmp': {u'Qc': u' ', u'Unit': u'(C)', u'Value': u'
16.5'}}, {u'ZipCodes': u'92710, 92707, 92667, 92650, 92630, 92623, 92620, 92619, 92618, 92616,
92614, 92612, 92606, 92604, 92603, 92602', u'Hour': u'0300', u'Julian': u'166', u'Standard': u'
metric', u'HlyRelHum': {u'Qc': u' ', u'Unit': u'(%)', u'Value': u'81'}, u'Station': u'75', u'Hl
yEto': {u'Qc': u' ', u'Unit': u'(mm)', u'Value': u'0.01'}, u'Date': u'2019-06-15', u'Scope': u'
hourly', u'HlyAirTmp': {u'Qc': u' ', u'Unit': u'(C)', u'Value': u'16.3'}}, {u'ZipCodes': u'9271
0, 92707, 92667, 92650, 92630, 92623, 92620, 92619, 92618, 92616, 92614, 92612, 92606, 92604, 9
2603, 92602', u'Hour': u'0400', u'Julian': u'166', u'Standard': u'metric', u'HlyRelHum': {u'Qc'
: u' ', u'Unit': u'(%)', u'Value': u'83'}, u'Station': u'75', u'HlyEto': {u'Qc': u' ', u'Unit':
u'(mm)', u'Value': u'0.01'}, u'Date': u'2019-06-15', u'Scope': u'hourly', u'HlyAirTmp': {u'Qc'
: u' ', u'Unit': u'(C)', u'Value': u'16.1'}}, {u'ZipCodes': u'92710, 92707, 92667, 92650, 92630
, 92623, 92620, 92619, 92618, 92616, 92614, 92612, 92606, 92604, 92603, 92602', u'Hour': u'0500
', u'Julian': u'166', u'Standard': u'metric', u'HlyRelHum': {u'Qc': u' ', u'Unit': u'(%)', u'Va
lue': u'83'}, u'Station': u'75', u'HlyEto': {u'Qc': u' ', u'Unit': u'(mm)', u'Value': u'0.01'},
u'Date': u'2019-06-15', u'Scope': u'hourly', u'HlyAirTmp': {u'Qc': u' ', u'Unit': u'(C)', u'Va
lue': u'15.9'}}, {u'ZipCodes': u'92710, 92707, 92667, 92650, 92630, 92623, 92620, 92619, 92618,
92616, 92614, 92612, 92606, 92604, 92603, 92602', u'Hour': u'0600', u'Julian': u'166', u'Stand
ard': u'metric', u'HlyRelHum': {u'Qc': u' ', u'Unit': u'(%)', u'Value': u'85'}, u'Station': u'7
5', u'HlyEto': {u'Qc': u' ', u'Unit': u'(mm)', u'Value': u'0.02'}, u'Date': u'2019-06-15', u'Sc
ope': u'hourly', u'HlyAirTmp': {u'Qc': u' ', u'Unit': u'(C)', u'Value': u'15.9'}}, {u'ZipCodes'
: u'92710, 92707, 92667, 92650, 92630, 92623, 92620, 92619, 92618, 92616, 92614, 92612, 92606,
92604, 92603, 92602', u'Hour': u'0700', u'Julian': u'166', u'Standard': u'metric', u'HlyRelHum'
: {u'Qc': u' ', u'Unit': u'(%)', u'Value': u'89'}, u'Station': u'75', u'HlyEto': {u'Qc': u' ',
u'Unit': u'(mm)', u'Value': u'0.04'}, u'Date': u'2019-06-15', u'Scope': u'hourly', u'HlyAirTmp'
: {u'Qc': u' ', u'Unit': u'(C)', u'Value': u'15.8'}}, {u'ZipCodes': u'92710, 92707, 92667, 9265
0, 92630, 92623, 92620, 92619, 92618, 92616, 92614, 92612, 92606, 92604, 92603, 92602', u'Hour'
: u'0800', u'Julian': u'166', u'Standard': u'metric', u'HlyRelHum': {u'Qc': u' ', u'Unit': u'(%
)', u'Value': u'84'}, u'Station': u'75', u'HlyEto': {u'Qc': u' ', u'Unit': u'(mm)', u'Value': u
'0.06'}, u'Date': u'2019-06-15', u'Scope': u'hourly', u'HlyAirTmp': {u'Qc': u' ', u'Unit': u'(C
)', u'Value': u'16.4'}}, {u'ZipCodes': u'92710, 92707, 92667, 92650, 92630, 92623, 92620, 92619
, 92618, 92616, 92614, 92612, 92606, 92604, 92603, 92602', u'Hour': u'0900', u'Julian': u'166',
u'Standard': u'metric', u'HlyRelHum': {u'Qc': u' ', u'Unit': u'(%)', u'Value': u'80'}, u'Stati
UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net


Session
Servers
Tools
Games
Sessions
View
Split
MultiExec
Tunneling
Packages
Settings
Help
Quick connect...
2. 183.116.1.79 (pi)
3. 183.116.1.79 (pi)
Sessions
Tools
Macros
Firefox
IntelliJ IDEA 2019.1 x64
iTunes
Malwarebyt...
MATLAB R2017b
CPU-Z
MobaXterm
Notepad++
'metric', u'HlyRelHum': {u'Qc': u'M', u'Unit': u'(%)', u'Value': None}, u'Station': u'75', u'Hl
yEto': {u'Qc': u'M', u'Unit': u'(mm)', u'Value': None}, u'Station': u'75', u'HlyEto': {u'Qc': u'M', u'Unit': u'(mm)
rly', u'HlyAirTmp': {u'Qc': u'M', u'Unit': u'(C)', u'Value': None}}, u'Date': u'2019-06-15', u'Scope': u'hou
07, 92667, 92650, 92630, 92623, 92620, 92619, 92618, 92616, 92614, 92612, 92606, 92604, 92603,
92602', u'Hour': u'1700', u'Julian': u'166', u'Standard': u'metric', u'HlyRelHum': {u'Qc': u'M'
, u'Unit': u'(%)', u'Value': None}, u'Station': u'75', u'HlyEto': {u'Qc': u'M', u'Unit': u'(mm)
', u'Value': None}, u'Date': u'2019-06-15', u'Scope': u'hourly', u'HlyAirTmp': {u'Qc': u'M', u'
Unit': u'(C)', u'Value': None}}, {u'ZipCodes': u'92710, 92707, 92667, 92650, 92630, 92623, 9262
0, 92619, 92618, 92616, 92614, 92612, 92606, 92604, 92603, 92602', u'Hour': u'1800', u'Julian':
'166', u'Standard': u'metric', u'HlyRelHum': {u'Qc': u'M', u'Unit': u'(%)', u'Value': None},
u'Station': u'75', u'HlyEto': {u'Qc': u'M', u'Unit': u'(mm)', u'Value': None}, u'Date': u'2019-
06-15', u'Scope': u'hourly', u'HlyAirTmp': {u'Qc': u'M', u'Unit': u'(C)', u'Value': None}}, {u'
ZipCodes': u'92710, 92707, 92667, 92650, 92630, 92623, 92620, 92619, 92618, 92616, 92614, 92612
, 92606, 92604, 92603, 92602', u'Hour': u'1900', u'Julian': u'166', u'Standard': u'metric', u'H
lyRelHum': {u'Qc': u'M', u'Unit': u'(%)', u'Value': None}, u'Station': u'75', u'HlyEto': {u'Qc'
: u'M', u'Unit': u'(mm)', u'Value': None}, u'Date': u'2019-06-15', u'Scope': u'hourly', u'HlyAi
rTmp': {u'Qc': u'M', u'Unit': u'(C)', u'Value': None}}, {u'ZipCodes': u'92710, 92707, 92667, 92
650, 92630, 92623, 92620, 92619, 92618, 92616, 92614, 92612, 92606, 92604, 92603, 92602', u'Hou
r': u'2000', u'Julian': u'166', u'Standard': u'metric', u'HlyRelHum': {u'Qc': u'M', u'Unit': u'
(%)', u'Value': None}, u'Station': u'75', u'HlyEto': {u'Qc': u'M', u'Unit': u'(mm)', u'Value':
None}, u'Date': u'2019-06-15', u'Scope': u'hourly', u'HlyAirTmp': {u'Qc': u'M', u'Unit': u'(C)'
, u'Value': None}}, {u'ZipCodes': u'92710, 92707, 92667, 92650, 92630, 92623, 92620, 92619, 926
18, 92616, 92614, 92612, 92606, 92604, 92603, 92602', u'Hour': u'2100', u'Julian': u'166', u'St
andard': u'metric', u'HlyRelHum': {u'Qc': u'M', u'Unit': u'(%)', u'Value': None}, u'Station': u
'75', u'HlyEto': {u'Qc': u'M', u'Unit': u'(mm)', u'Value': None}, u'Date': u'2019-06-15', u'Sco
pe': u'hourly', u'HlyAirTmp': {u'Qc': u'M', u'Unit': u'(C)', u'Value': None}}, {u'ZipCodes': u'
92710, 92707, 92667, 92650, 92630, 92623, 92620, 92619, 92618, 92616, 92614, 92612, 92606, 9260
4, 92603, 92602', u'Hour': u'2200', u'Julian': u'166', u'Standard': u'metric', u'HlyRelHum': {u
'Qc': u'M', u'Unit': u'(%)', u'Value': None}, u'Station': u'75', u'HlyEto': {u'Qc': u'M', u'Uni
t': u'(mm)', u'Value': None}, u'Date': u'2019-06-15', u'Scope': u'hourly', u'HlyAirTmp': {u'Qc'
: u'M', u'Unit': u'(C)', u'Value': None}}, {u'ZipCodes': u'92710, 92707, 92667, 92650, 92630, 9
2623, 92620, 92619, 92618, 92616, 92614, 92612, 92606, 92604, 92603, 92602', u'Hour': u'2300',
u'Julian': u'166', u'Standard': u'metric', u'HlyRelHum': {u'Qc': u'M', u'Unit': u'(%)', u'Value
': None}, u'Station': u'75', u'HlyEto': {u'Qc': u'M', u'Unit': u'(mm)', u'Value': None}, u'Date
': u'2019-06-15', u'Scope': u'hourly', u'HlyAirTmp': {u'Qc': u'M', u'Unit': u'(C)', u'Value': N
one}}, {u'ZipCodes': u'92710, 92707, 92667, 92650, 92630, 92623, 92620, 92619, 92618, 92616, 92
614, 92612, 92606, 92604, 92603, 92602', u'Hour': u'2400', u'Julian': u'166', u'Standard': u'me
tric', u'HlyRelHum': {u'Qc': u'M', u'Unit': u'(%)', u'Value': None}, u'Station': u'75', u'HlyEt
o': {u'Qc': u'M', u'Unit': u'(mm)', u'Value': None}, u'Date': u'2019-06-15', u'Scope': u'hourly
', u'HlyAirTmp': {u'Qc': u'M', u'Unit': u'(C)', u'Value': None}}], u'Type': u'station', u'Name'
: u'cimis'}]}}
UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net
Type here to search
DELL




Picture above shows text displaying when connectiong to api was a success. Also population of the List holding the values for Temp from cimis.

## Code:

A few pieces of code:

```
## Function in charge of calling cimis api and getting our data from station 75: ###
def call_Cimis_Api():
    global response_success
    global cimis_response
    print("++++++++++++++++++++++++++++++++++++++++++++++++")
    print("inside call_Cimis_Api(): ")

url="http://et.water.ca.gov/api/data?appKey=55a1b0c5-298b-4fd5-9c42-2576c839580d
```

```
&targets=75&startDate=2019-06-15&endDate=2019-06-15&unitOfMeasure=M&dataIte
ms=hly-eto,hly-air-tmp,hly-rel-hum"
   try:
      response=requests.get(url,timeout=15,header={'Active': 'application/json'})
      #if the response was successful, no Exception will be raised
   except HTTPError as http_err:
         response_success=False
      print("HTTP error ocurred:",http_err)
   except Exception as err:
         response_success=False
      print("Other error occurred:", err)
   else:
         if(response):
                  cimis_response=response.json() #turn cimis response into a pythong dict
                  print("Success getting cimis info into python dict!")
         response_success=True


######## This function extracts the records inside cimis_response dictionary and it
##   sends it to temAPiList, humiAPiList, EtoAPiList:
def parse_cimisdict():
   global response_success
   global tempAPiList
   global humiAPiList
   global EtoAPiList
   if(response_success):
         for index in range(0,len(cimis_response['Data']['Providers'][0]['Records'])):

print("================================================")
                  print(index)

if(cimis_response['Data']['Providers'][0]['Records'][index]['HlyAirTmp']['Value'] is not
None):

value.append(cimis_response['Data']['Providers'][0]['Records'][index]['HlyAirTmp']['Value
'])
                           print("value =" ,value)
                           y=float(value)
         tempAPiList.append(y)
```

```
elif(cimis_response['Data']['Providers'][0]['Records'][index]['HlyRelHum']['Value'] is not
None):

value2.append(cimis_response['Data']['Providers'][0]['Records'][index]['HlyRelHum']['Val
ue'])
                        print("value2 =", value2)
                        x=float(value2)
        humiAPiList.append(x)
                elif(cimis_response['Data']['Providers'][0]['Records'][index]['HlyEto']['Value']
is not None):

value3.append(cimis_response['Data']['Providers'][0]['Records'][index]['HlyEto']['Value'])
                        print("value3 =", value3)
                        z=float(value3)
        EtoAPiList.append(z)

   //print("TempAPiList length= ",len(tempAPiList))
        //print ("TempAPIlist: ",tempAPiList)
        //print("humiAPiList length= ",len(humiAPiList))
        //print ("humiAPIlist: ",humiAPiList)
   //print("EtoAPiList length= ",len(EtoAPiList))
        //print ("EtoAPIlist: ",EtoAPiList)




## This function  gets the average humidity from our DHT11  readings ############
def get_Humidity_avg():
   print("humidity average: ")
   sum=0.0
   if (len(humidityList)>0):
      for y in humidityList:
```

```
            sum = sum+y
        return(round( (sum/ len(humidityList)),3))
    return 0.00

## This function  gets the average  temperature from our DHT11 readings #######
def get_Temp_avg():
    print("Temp average: ")
    sum=0.0
    if(len(tempList)>0):
        for x in tempList:
            sum=sum+x
        return (round((sum /len(tempList)),3))
    return 0.00

##  This function   gets the average humidity from our API data list readings###
def get_HumidityAPI_avg():
    print("humidity average: ")
    sum=0.0
    if (len(humiAPiList)>0):
        for y in humiAPiList:
            sum = sum+y
        return(round( (sum/ len(humiApiList)),3))
    return 0.00

## This function  gets the average  temperature from API data list readings #######
def get_TempAPI_avg():
    print("Temp average: ")
    sum=0.0
    if(len(tempAPiList)>0):
        for x in tempAPiList:
            sum=sum+x
        return (round((sum /len(tempAPiList)),3))
    return 0.00


## This function  gets the average  Eto from API data list readings #######
def get_EtoApi():
    print("ETo average: ")
    sum=0.0
```

```
if(len(EtoAPiList)>0):
    for x in EtoAPiList:
        sum=sum+x
    return (round((sum /len(EtoAPiList)),3))
return 0.00
```