**CSEE5590-0005/490-0005**

**IOT / Robot Programming**

**Lesson Plan #3**

**Lesson Title:** Weather Station

**Lesson Description:** *Create your own weather station to monitor House temperature*

**In Class Exercise**

**ICP / ICE Part 1**

Assemble the following sensor according to schematics given in the lecture presentation slides:

* Temperature sensor Module
* Barometer Module
* Light sensor module
* UV sensor module
* Dust sensor module separately
* Display results in IDE monitor

**ICP / ICE Part 2:**

* Connect LCD with each component and display results
* Connect WIFI with each module
* Transmit the results for individual modules to ThingSpeak Server separately
* Visualize the results in Mobile

**ICP / ICE Part 3:**

Integrate all of the modules together and the complete circuit should be able to transmit the data to ThingSpeak for visualization.

**Bonus Part:**

The bonus part will be announced in the class according to class progress in the specific ICP or ICE.

**ThingSpeak Set up:**

[ThingSpeak](https://thingspeak.com/) provides very good tool for [IoT based projects](http://circuitdigest.com/internet-of-things-iot-projects). By using ThingSpeak site, we can monitor our data and control our system over the Internet, using the Channels and webpages provided by ThingSpeak. ThingSpeak ‘Collects’ the data from the sensors, ‘Analyze and visualize’ the data and ‘Acts’ by triggering a reaction.

* Create account
* Create new channel
* Name the field
* Copy the API key and use it in your code

### Using Android App

You'll be able to visualize data in any browser. But you might also check it on you Android based smart phone and visualize it whenever you want.

* Download and install **[ThingsView](https://play.google.com/store/apps/details?id=com.cinetica_tech.thingview)** app from Google Play store on your Android device
* On the app, insert your channel ID number and click add. You'll find the ID on your ThingSpeak channel configuration
* The current values of each variable will be displayed in a graph