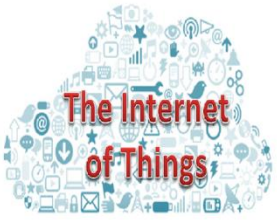
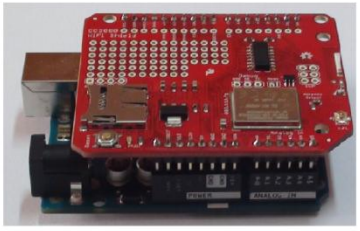
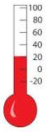
TENET TECHNETRONICS

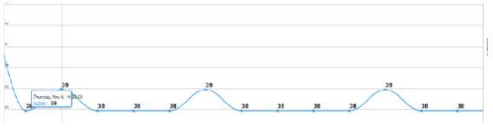
2014

Plotting Temperature Sensor Values on TENET TECHNETRONICS Cloud Using Wi-Fi Module (CC3000) with Arduino









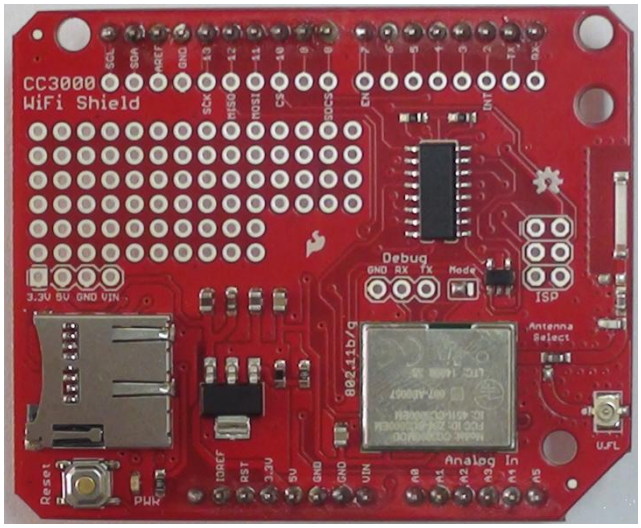
"Simplifying

Author: Mr. Prajwal R

Introduction:

The CC3000 Wi-Fi module from Texas Instruments is a small silver package which finally brings easy-to-use, affordable Wi-Fi functionality to your Arduino projects. It uses SPI for communication (not UART!) so you can push data as fast as you want or as slow as you want. It has a proper interrupt system with IRQ pin so you can have asynchronous connections. It supports 802.11b/g, open/WEP/WPA/WPA2 security, TKIP & AES. A built in TCP/IP stack with a "BSD socket" interface supports TCP and UDP in both client and server mode, with up to 4 concurrent socket connections.

"e-megha.tenettech.net" is aTenet Technetronics cloud solution. That can set up number of "streams" on https://e-megha.tenettech.net/ and post, fetch, delete data from any number of Internet-connected devices.



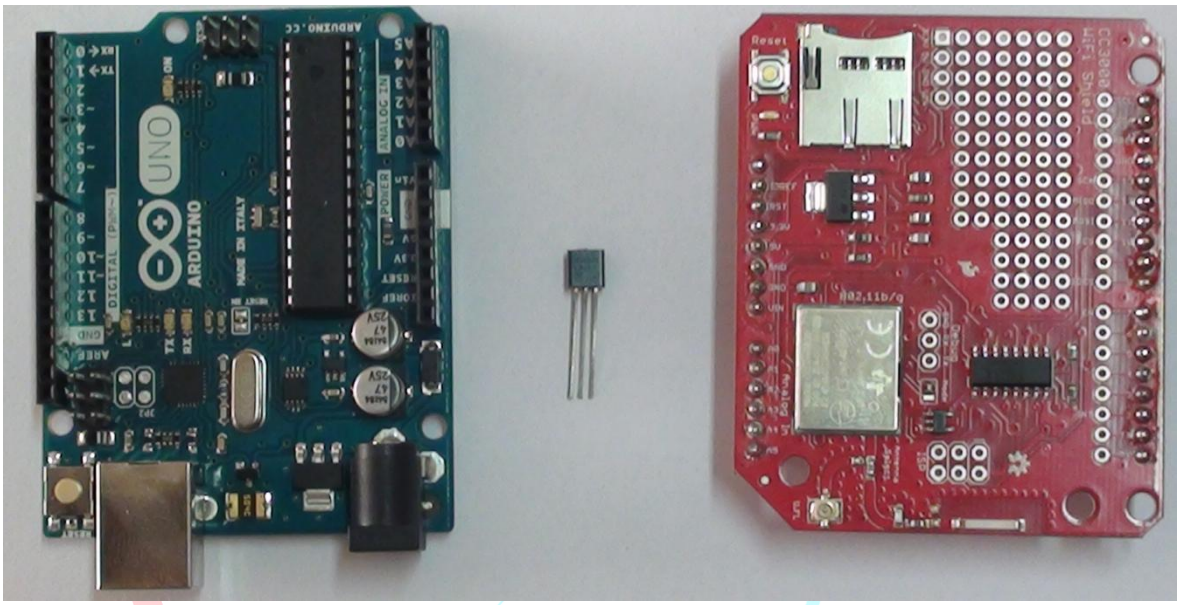
All of this falls under the umbrella of Internet of Things (IOT), in which Internet-connected embedded systems can communicate with Internet services, such as web sites or other connected devices.

Note: The CC3000 does not support "AP" mode, it can connect to an access point but it cannot be an access point.

Here is a quick demo on how to Plot Temperature Sensor Values on a Cloud Using Wi-Fi Module (CC3000) with Arduino

Hardware Required:

- Arduino board.
- Arduino cable.
- Wi-Fi module Shield (CC3000).
- Temperature Sensor LM35.

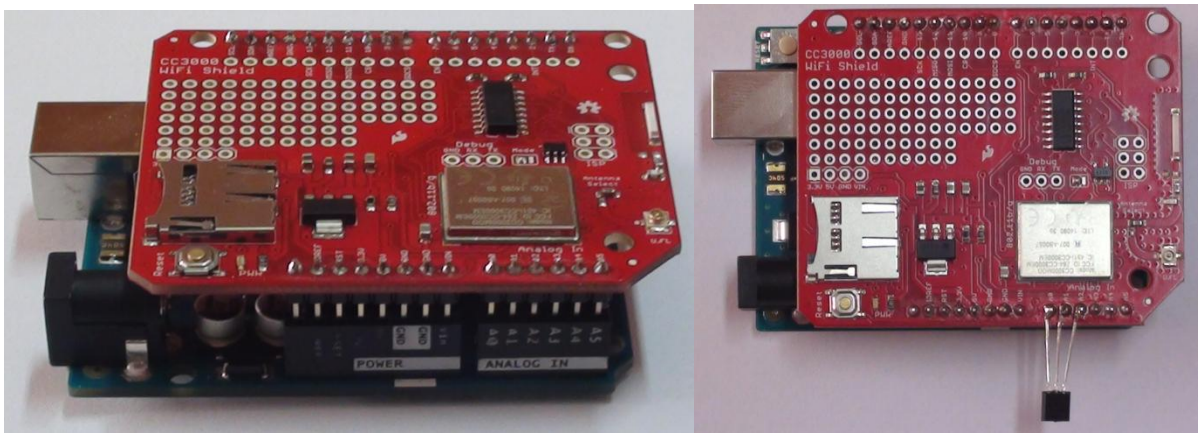


Step one:

The Wi-Fi CC3000 is a shield that piggybacks onto an Arduino board and Temperature Sensor Is connected as following.

Upon Wi-Fi Shield	Temperature Sensor
Digital 14 (A0)	GND
Analog 1 (A1)	Output
Digital 16 (A2)	VCC

Note: Gnd and Vcc pins of Temperature Sensor can be directly connected to the Gnd and 5v pins of the Arduino board respectively.



Step two:

Connect the Arduino board to the computer using Arduino USB cable. Launch the Arduino IDE and select the appropriate serial port and board.

Here's a sketch that posts the Temperature sensor data to "e-megha.tenettech.net" using the Wi-Fi Shield.

Program:

```
/*****************************************************************

 * WebClient.ino

 * CC3000 WebClient Test

 *

 * Manually connects to a WiFi network and performs an HTTP GET

 * request on a web page. Prints the contents of the page to

 * the serial console.

 *

 * The security mode is defined by one of the following:

 * WLAN_SEC_UNSEC, WLAN_SEC_WEP, WLAN_SEC_WPA, WLAN_SEC_WPA2

 *

 * Hardware Connections:

 *
```

```
 * Uno Pin    CC3000 Board    Function
 *
 * +5V       VCC or +5V      5V
 * GND       GNDGND
 * 2         INT             Interrupt
 * 7         EN              WiFi Enable
 * 10        CS              SPI Chip Select
 * 11        MOSI            SPI MOSI
* 12         MISO            SPI MISO
 * 13        SCK             SPI Clock
 *
 * Resources:
 * Include SPI.h, SFE_CC3000.h, and SFE_CC3000_Client.h
 ******************************************************************/


#include <SPI.h>
#include <SFE_CC3000.h>
#include <SFE_CC3000_Client.h>

// Pins
#define CC3000_INT    2   // Needs to be an interrupt pin (D2/D3)
#define CC3000_EN     7   // Can be any digital pin
#define CC3000_CS     10  // Preferred is pin 10 on Uno


// Connection info data lengths
```

```
#define IP_ADDR_LEN     4   // Length of IP address in bytes


// Constants
char ap_ssid[] = "ssid";              // SSID of network
char ap_password[] = "password";      // Password of network
unsignedint ap_security = WLAN_SEC_WPA2; // Security of network
unsignedint timeout = 30000;          // Milliseconds
char server[] = "e-megha.tenettech.net";      // Remote host site
int temp;
// Global Variables
SFE_CC3000 wifi = SFE_CC3000(CC3000_INT, CC3000_EN, CC3000_CS);
SFE_CC3000_Client client = SFE_CC3000_Client(wifi);


void setup() {


ConnectionInfoconnection_info;
inti;
pinMode(14, OUTPUT);
pinMode(16, OUTPUT);
digitalWrite(14, LOW);
digitalWrite(16, HIGH);


 // Initialize Serial port
Serial.begin(115200);
```

```
Serial.println();
Serial.println("-------------------------");
Serial.println("CC3000 is ready");
Serial.println("-------------------------");

 // Initialize CC3000 (configure SPI communications)
if ( wifi.init() ) {
Serial.println("CC3000 initialization complete");
 }
else {
Serial.println("Something went wrong during CC3000 init!");
 }

 // Connect using DHCP
Serial.print("Connecting to SSID: ");
Serial.println(ap_ssid);
if(!wifi.connect(ap_ssid, ap_security, ap_password, timeout)) {
Serial.println("Error: Could not connect to AP");
 }

 // Gather connection details and print IP address
if ( !wifi.getConnectionInfo(connection_info) ) {
Serial.println("Error: Could not obtain connection details");
 }
else {
```

```
Serial.print("IP Address: ");
for (i = 0; i< IP_ADDR_LEN; i++) {
Serial.print(connection_info.ip_address[i]);
if ( i< IP_ADDR_LEN - 1 ) {
Serial.print(".");
    }
  }
Serial.println();
 }


}
void loop() {
int a=analogRead(A1);
temp = ((a*100.0*5.0)/1023.0);
Serial.println(temp);
Serial.println("Posting!");
postData();
delay(1000);
 // If there are incoming bytes, print them
if ( client.available() ) {
char c = client.read();
Serial.print(c);
 }
```

```
}
voidpostData()
{

 // Make a TCP connection to remote host
Serial.print("Performing HTTP GET of: ");
Serial.println(server);
if ( !client.connect(server, 80) ) {
Serial.println("Error: Could not make a TCP connection");
 }

 // Make a HTTP GET request
 // Serial.println(temp);
client.print("GET /update.php?id=T310071&value=");
client.print(temp);
client.println(" HTTP/1.1");
client.print("Host: ");
client.println(server);
client.println("Connection: close");
client.println();
Serial.println();
delay(1000);
}
```

Note: Before uploading the sketch,download & install the CC3000 Arduino Library and there are a few global variables need to adjustto alter some constants to connect to your WiFI network.

The ap_ssid and ap_password strings should be adjusted to match your WiFi network. If your network is open (no passkey), that second variable should be blank (ap_password [] = "";). Also make sure to adjust the ap_security variable to match the security of your network. See the comments for possible values.

Step three:

Once you've made those adjustments, you can safely upload the code. After you've sent the sketch to your Arduino, open up the Serial Monitor to get an idea of what's going on.