Vol 8. No. 3 – September, 2015

African Journal of Computing & ICT

www.ajocict.net

CC3000 Wifi Base Configuration Of Gateway For Internet Accessibility in Monitoring Variables Via Wireless Sensor Systems

[1]J. Agajo, [2]I. C. Obiora-Dimson, [3]O.B. Chukwuejekwu & [4]N. Nathaniel

[1]Federal University of Technology Dept. of Computer Engineering, P.M.B 65 Minna Nigeria

[2]Department of Electronic and Computer Engineering Nnamdi Azikiwe University Awka

[3]Department of Information Management Technology, Federal University of Technology Owerri Imo

[4]Department of Electrical/Electronic Imo State Polytechnic Umuagwo Imo State

ABSTRACT

CC3000 WiFi base configuration of gateway for Internet access in monitoring variables via Wireless sensor systems tends to develop interconnection of various interface for physical parameters to be sent to internet, this work developed an interface based on the CC3000 WIFI Shield. The setup is made up of Sensor, ADC, Processor WIFI Shield and the gateway. Physical Parameter are collected and converted to digital electrical signal, processed and sent to the WIFI Shield which converts them to radio waves using Orthogonal Frequency Division Multiplexing (OFDM). To transmit to the internet the service set identifier SSID and Password of the gateway is provided, the SSID was included in the source code to enable connection to the gateway. This was accomplished by running various test like Scan test, Ping test, connection test, web Client test smart Configuration and self-connection test In summary this work developed a platform for data transfer. This application has the ability to retrieve sensed data directly from a wireless sensor system composed of sensors ADC WIFI and gateway.

Keywords: Wi-Fi, gateway, Sensor SSID WPA.**Keywords**: ICT, internet, risk, students, tertiary institutions

African Journal of Computing & ICT Reference Format:

J. Agajo, I. C. Obiora-Dimson, O.B. Chukwuejekwu & N. Nathaniel (2015): CC3000 Wifi Base Configuration Of Gateway For Internet Accessibility in Monitoring Variables Via Wireless Sensor Systems. Afr J. of Comp & ICTs. Vol 8, No. 3. Pp 219-227.

1. INTRODUCTION

Various parameters can be monitored, which include temperature, energy, tank levels, speed or just controlling switches, this wireless remote monitoring solution puts you informed and control in a matter of seconds. Heretofore, wireless monitoring was only possible outdoor using satellite and terrestrial networks. Indoor monitoring however. was stressful for these system owing to problems of multipath, signal impairment and so on, therefore the need for a system that can serve as a supporting network for accurate monitoring became necessary.. Wireless Sensor Systems are spatially distributed autonomous sensors to monitor physical or environmental conditions, such as temperature, sound, pressure, etc. and to cooperatively pass their data through the network to a main location , this Location could be indoor or outdoor.

The more modern networks are bi-directional, also enabling control of sensor activity. The development of wireless sensor networks was motivated by military applications such as battlefield surveillance. Recent advances in wireless communications and micro electro-mechanical systems have motivated the development of extremely small, low-cost sensors that possess sensing, signal processing and wireless[2] communication capabilities. Hundreds and thousands of these inexpensive sensors work together to build a Wireless Sensor System, which can be used to collect useful information (i.e. temperature, humidity) from a variety of environment.

The collected data must be transmitted to remote base station (BS) for further processing. Wireless Sensor Systems (WSSs) have been used in many application domains such as habitat monitoring [1], infrastructure protection [1], and scientific exploration [2]. Smart environments represent the next evolutionary development step in building, utilities, industrial, home, shipboard, and transportation systems automation. Like any sentient organism, the smart environment relies first and foremost on sensory data from the real world. Sensory data comes from multiple sensors of different modalities in distributed locations.

Wireless Sensor Systems generally consist of a data acquisition network and a data distribution network, monitored and controlled by a management centre. The sensor node can measure data from any physical system and send it, usually via radio transmitter, to a command centre or sink node, either using a single hop or multiple hops operation. The data traverses through a number of data concentration devices (or gateways) and networks until it gets to its destination.[6] In order to improve viewership, this work goes beyond the present convention of having to arrange a set-up within a local area by providing an interface between a local area and the internet called the gateway. Such a setup might not be flexible and robust to meet up with the current demand.

This work, implements the stage which provide the interface that will connect to the web. This interface between the Local Area Network LAN and Wide Area Network WAN is called the Gateway. The gateway makes it possible for data to be sent to the internet which can be viewed via web-based application

2. WEB-BASED APPLICATION

A Web-based application refers to any program that is accessed over a network connection using HTTP, rather than existing within a device's memory. Web-based applications often run inside a Web browser. However, Web-based applications also may be client-based, where a small part of the program is downloaded to a user's desktop, but processing is done over the Internet on an external server.[9] There is a lot of confusion created by the use of terms like Web-based, Internet-based and cloud-based when referring to applications. Web-based applications actually encompass all the applications that communicate with the user via HTTP. This includes light applications like Flash games, online calculators, calendars, and so on, as well as more intensive applications that use HTTP.

3. GATEWAY

An internet gateway essentially means a system that is acting as a router for other devices on the network to receive internet access.

- A gateway may contain devices such as protocol translators, impedance matching devices, rate converters, fault isolators, or signal translators as necessary to provide system interoperability. It also requires the establishment of mutually acceptable administrative procedures between both networks.
- A protocol translation/mapping gateway interconnects networks with different network protocol technologies by performing the required protocol conversions.
- Loosely, a computer or computer program configured to perform the tasks of a gateway. For a specific case, see default gateway.

Gateways, also called protocol converters, can operate at any network layer. The activities of a gateway are more complex than that of the router or switch as it communicates using more than one protocol.

4. MODEM

The Origin of Modems

The word "modem" is a contraction of the words **modulator-demodulator**. A modem is typically used to send digital data over a phone line. The sending modem **modulates** the data into a signal that is compatible with the phone line, and the receiving modem **demodulates** the signal back into digital data .**Wireless modems** convert digital data into radio signals and back.

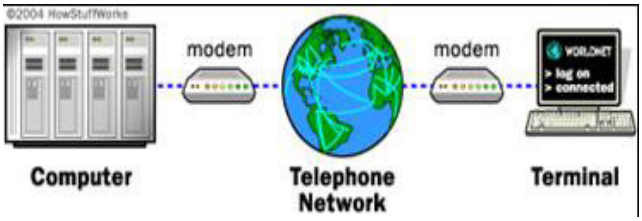

Figure 1 Setup of a MODEM[8]

Modems came into existence in the 1960s as a way to allow terminals to connect to computers over the phone lines. When a modem first makes a connection, you will hear screeching sounds coming from the modem. These are digital signals coming from the computer to which you are connecting being modulated into audible sounds. The modem sends a higher-pitched tone to represent the digit 1 and a lower-pitched tone to represent the digit 0 as shown in figure 1 and 2.

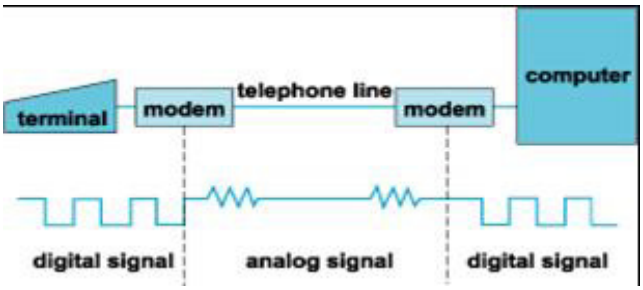

Figure 2 Working Principle of MODEM[10]

Communication between LAN and WAN is shown in figure 3 and figure 4, which is made possible by the gateway. A number of variables can be measured via sensor for onward conditioning via the Analog to Digital Converter (ADC) then to the Processor, unlike other setup where GPRS is used as gateway which rather appear as being too slow. This work adopts the use of Wifi Shield that has higher data delivery rate..

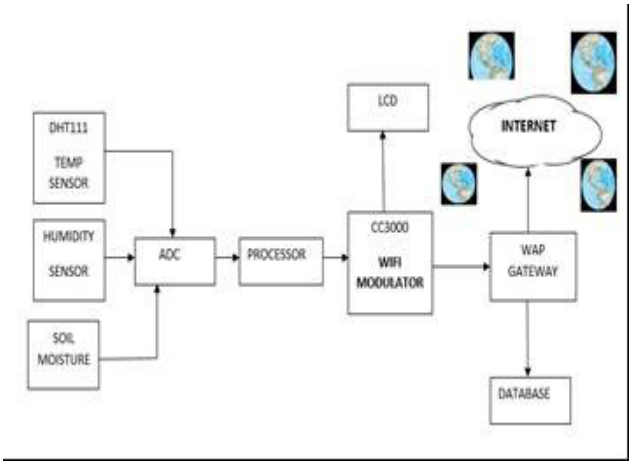

Figure 3 Web-based remote monitoring setup

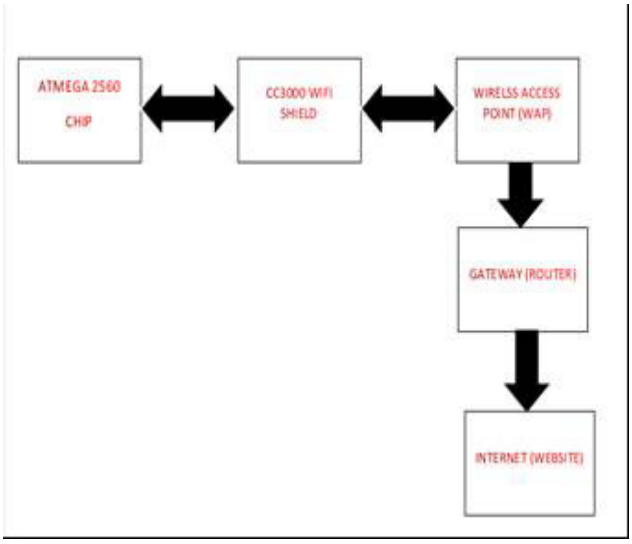

Figure 4 Block Diagram of the connection[12]

5. WIFI SHIELD

The Arduino **WiFi shield** allows an Arduino board to connect to the internet using the **WiFi** library and to read and write an SD card using the SD library. The **WiFi** Library is included with the most recent version of the Arduino IDE. The firmware for the **WiFi shield** has changed in Arduino IDE The Arduino WiFi shield allows an Arduino board to connect to the internet using the WiFi library and to read and write an SD card using the SD library. The WiFi Library is included with the most recent version of the Arduino IDE. ***The firmware for the*** WiFi ***shield has changed in Arduino IDE.*** ***It is strongly recommended to install this update per*** these instructions The WiFi library is similar to the Ethernet library and many of the function calls are the same. To use the shield, mount it on top of an Arduino board (e.g. the Uno). To upload sketches to the board, connect it to your computer with a USB cable as usual. Once the sketch has been uploaded, one can disconnect the board from the computer and power it with an external power supply. Digital pin 7 is used as a handshake pin between the WiFi shield with the block diagram in figure 5 and the Arduino, and should not be used

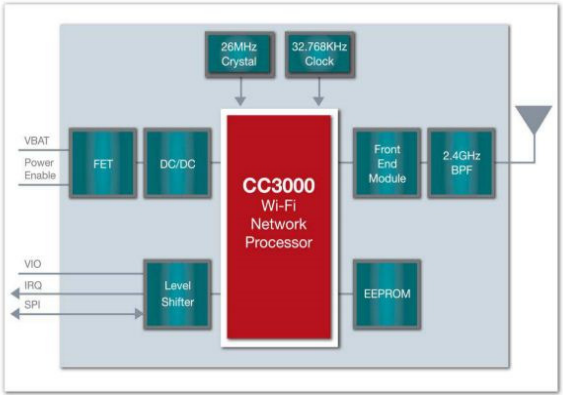

Figure 5 Wifi Shield Architecture [4]

6. SMARTCONFIG

SmartConfig is the special functionality in the CC3000 that allows setting the SSID and password settings without having to type or re-program the module. Any iOS/Android device can be used to set the configuration - solving the annoying deployment problem of how to set the connection details for a new device.

7.SMARTCONFIGCREATE AND SMARTCONFIGRECONNECT

These two SmartConfig sketches should be used together to demonstrate how the SmartConfig app was used on the smartphone to pass connection details to the CC3000.

8. SMARTCONFIGCREATE

This sketch will initialise the CC3000, erasing any previous connection details stored on the device. It will then enter SmartConfig mode with a 60 second timeout where it waits for configuration data to arrive from the Smart Phone.

If a connection was successfully established, the connection details will be stored in the non-volatile memory of the CC3000, and the module will be configured to automatically reconnect to this network on startup (meaning no need to run the SmartConfig app unless the AP details change or one erase the stored connection details on the module).

There won't be need to edit the sketch to add the SSID and password - the SmartConfig app does that.

9. SMART CONFIG RECONNECT

This sketch shows the process involved of using CC3000 in 'reconnect' mode, and avoid erasing all stored connection profiles, which is unfortunately necessary with other sketches where manual config data is provided.

- Initialize the CC3000 with a special SmartConfig flag so it doesn't erase the profile data
- Access Point connection (based on saved AP details)
- DHCP address assignment
- Disconnect

Using the SmartConfigCreate Sketch

Step One: Install the SmartConfig App

Before one can use SmartConfig to provide an AP connection details, there is need to install the SmartConfig app:

Step Two: Configure the SmartConfig App on your Phone

Once installed the SmartConfig app, connect to the AP that the CC3000 will be using (HOMENETWORK) as shown in figure 6 and then load the app. Figure 6 shows the device configuration with the AP's **SSID**, **Gateway IP Address** and **Device Name** fields already populated:

Figure 6 Device Configuration

1. Hello, CC3000!
2. RX Buffer : 131 bytes
3. TX Buffer : 131 bytes
4. Free RAM: 595
5. Initialising the CC3000 ...
6. Firmware V. : 1.24
7. MAC Address : 0x08 0x00 0x28 0x01 0xA8 0x1F
8. Waiting for a SmartConfig connection (~60s) ...

The code on the Arduino will send messages back to your computer over the USB connection and these are visible in the Arduino Serial Monitor. Open the Serial Monitor by clicking on the icon at the top right of the Arduino window. This resets the Arduino and allows you to see the messages.

Serial Monitor not opening?

Make sure you've got the same Serial Port you used to Upload the code. You can selected it in **Tools→Serial Port**.
After a few seconds you should see--- reset ---
BERGCloudCC3000 version: 203
BERGCloud: Connecting to WiFi network...
BERGCloud: Using SmartConfig.
BERGCloud: CC3000 firmware version 1.28
BERGCloud:
CC3000 MAC address XX:XX:XX:XX:XX:XX

In place of the Xs you should see the CC3000's MAC Address. Your CC3000 firmware version may also be different. Leave the Serial Monitor open so you can check the progress of the Smart Config.

10. OPEN THE TI SMART CONFIG APP ON IOS DEVICE

The iOS device will need to be connected to the network you would like your device to join. It will set the SSID to the network name automatically. In this case it's "Test 2.4GHz".

Figure 7 Configuring SSID and Pasword

Enter your Wi-Fi Password [3]
This will be shown as plain text.

Figure 8 Configuration with Password Entered

Press start to attempt configuration
The button will change colour to red in figure 8

Check if configuration is complete
On completion the button with change back to blue.

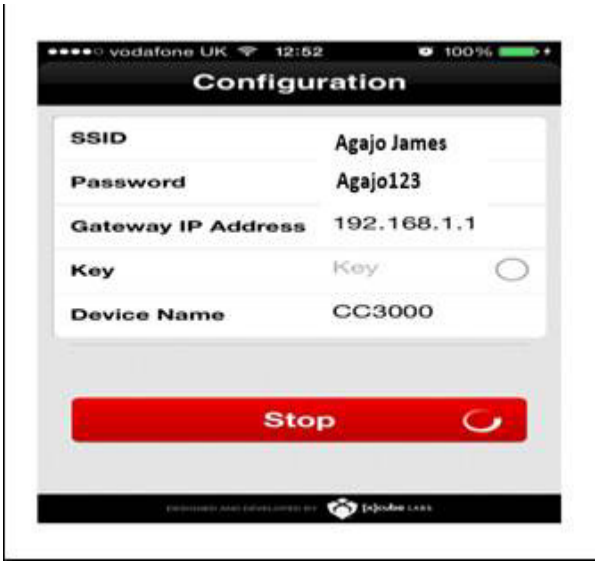

Figure 9 Checking if Configuration is Complete

The Arduino Serial Monitor should now display:

```
--- reset ---
BERGCloudCC3000 version: 203
BERGCloud: Connecting to WiFi network...
BERGCloud: Using SmartConfig.
BERGCloud: CC3000 firmware version 1.28
BERGCloud: CC3000 MAC address
XX:XX:XX:XX:XX:XX
BERGCloud: Waiting for DHCP...
BERGCloud: IP address: XX.XX.XX.XXX
BERGCloud: Netmask: XXX.XXX.XXX.0
BERGCloud: Gateway: XX.XX.XX.X
BERGCloud: DHCP server: XX.XX.XX.X
BERGCloud: DNS server: XX.XX.XX.X
BERGCloud: Looking up host:
bridge.bergcloud.com
BERGCloud: Host IP address: 46.137.83.12
Waiting...
Connecting took 21 seconds.
That's all for this example.
```

Now a connection to WiFi is established
Not secure enough?
Because of the way Smart Config transmits the data between the phone and the CC3000 one may want to encrypt the config data using a pre-shared key set in the app and in your Arduino code.

Define a pre-shared key in your Arduino code
To set up Smart Config with a pre-shared key it can either open and upload the **File→Examples→BergCloudCC3000→Smart_Config_With_Key** example or modify the **Smart_Config** code you should already have open.
If you want to modify your Arduino code, find the line
If you want to modify your Arduino code, find the line
`#define WLAN_SMARTCONFIG_KEY NULL`
and replace it with
`#define LAN_SMARTCONFIG_KEY "1234567abcdefgh"`
The key can be set to any 16 characters. **But it must be exactly 16 characters**.
Upload the code and open the Serial Monitor as you did in steps 4 and 5.

Set the key in the Smart Config app
After opening the Smart Config app set the Password and enter the pre-shared key `abcdefghijklmnop` in the Key field and press **Done**. Make sure you enter all 16 characters.

Enable the key
Touch the small circle to the right of the Key to use it.

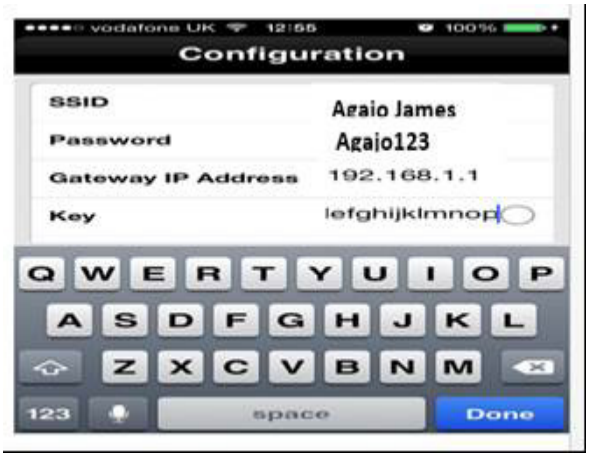

Figure 10 Set the key in the Smart Config app

Figure 11 Enable the key

The circle will become a tick and if your key is 16 characters long you will be able to press the blue **Start** button to begin the Smart Config process as in figure 11.

Configure the CC3000
Press the **Start** button, it will turn red and become blue when it's complete as shown in figure 12.

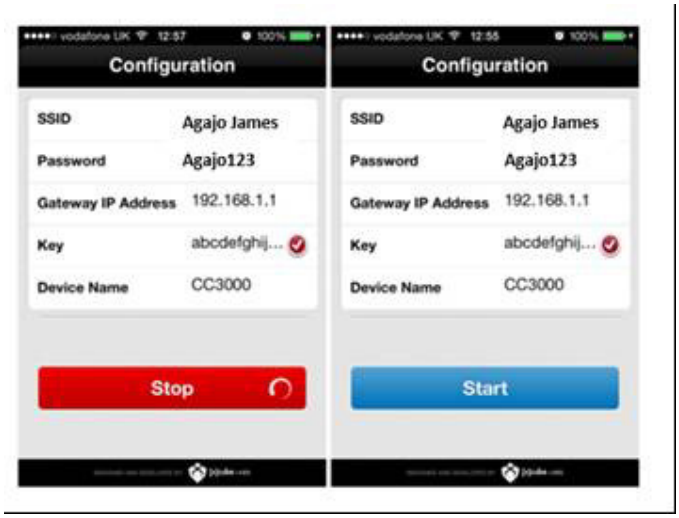

Figure 12 Configure the CC3000

You can check the state of your device in the Serial Monitor, just like in step 9.
The connection will be established again
That's all this example code will do.

11. TEST RESULTS

The CC3000 was interfaced with an ArduinoMega 2560 and the following test were carried out

- Board Test
- Scan Test
- Ping test
- Connection test
- Web client test
- Smart configuration and Fast connect test

11.1 Board Test

The board test was carried out purposely to show that the CC3000 wifi shield is working. That is to test its workability. The test result is shown in figure 13.

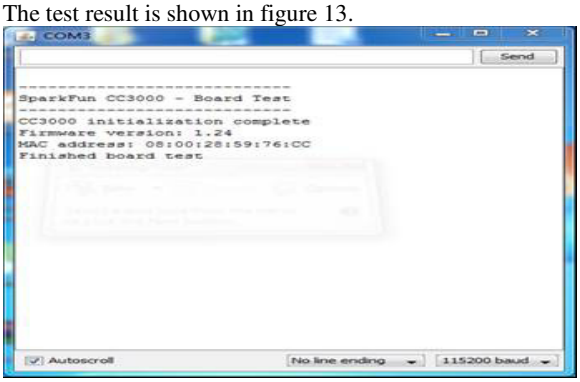

Figure 13 Board test of the CC3000 wifi shield

One major function of the wifi module is it ability to identify and detect networks within its surrounding. These networks could be open or closed irrespective of their security implementation or protocols.

11.2 Scan Test

These test was carried out in in figure 14 in order for the wifi module to scan or search and display available network with their properties within a confined environment.

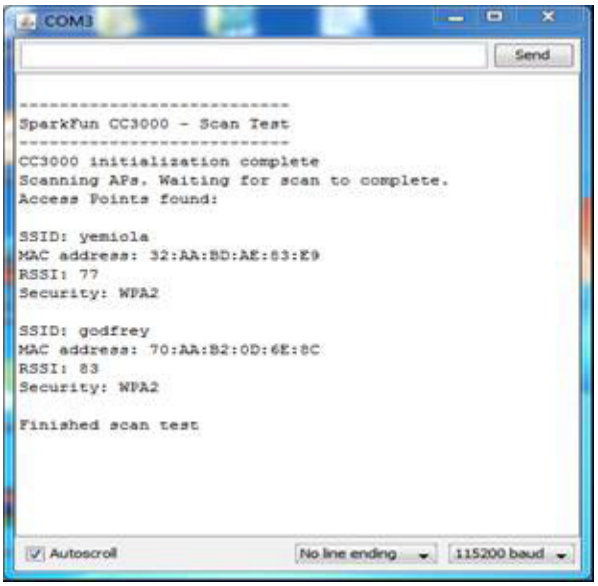

Figure 14 Scan test

The CC3000 WIFI shield scans it environment for available networks irrespective of the security levels and display on the serial monitor the properties of that network which includes the name or SSID of the network, The MAC address of the network, the RSSI port number and the security mode either WPA2, WPA or unsecured network.

11.3 Connection Test

Aside scanning a network, one major ability of the WIFI module is to connect to that network. This can only be achieved if the WIFI shield has all the necessary credentials to connect to the network access point as in figure 15.

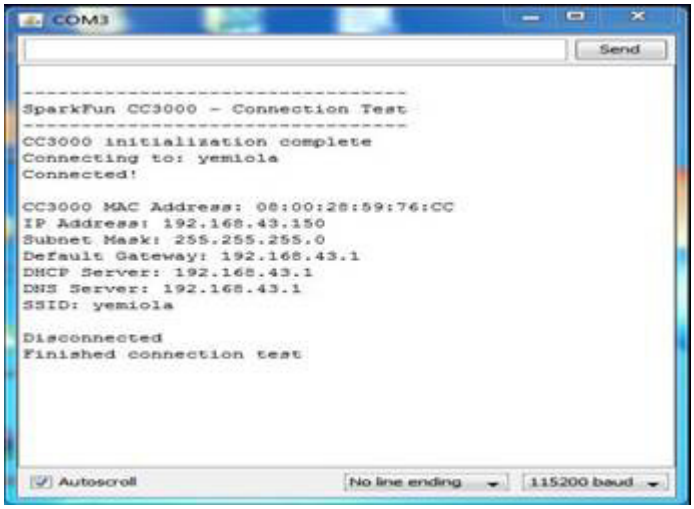

Figure 15 Connection test

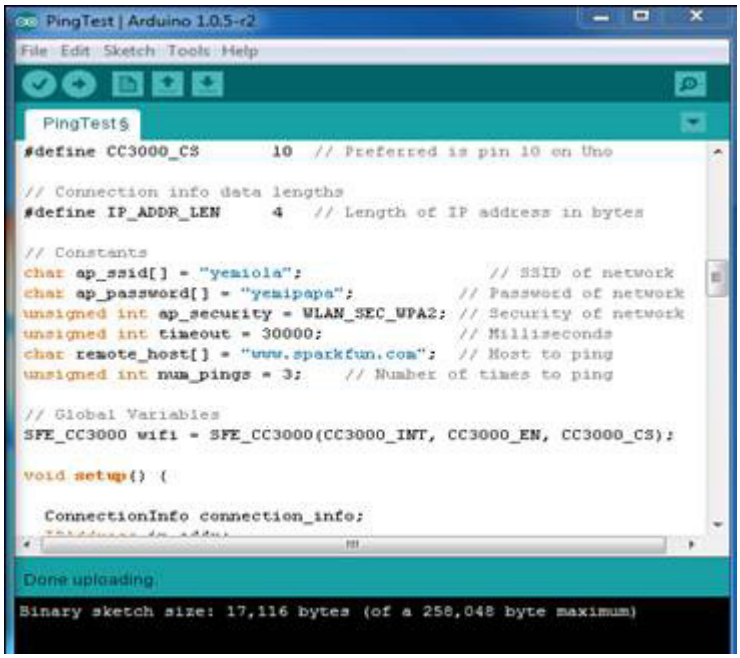

Figure 16 Arduino code for CC3000 connection test

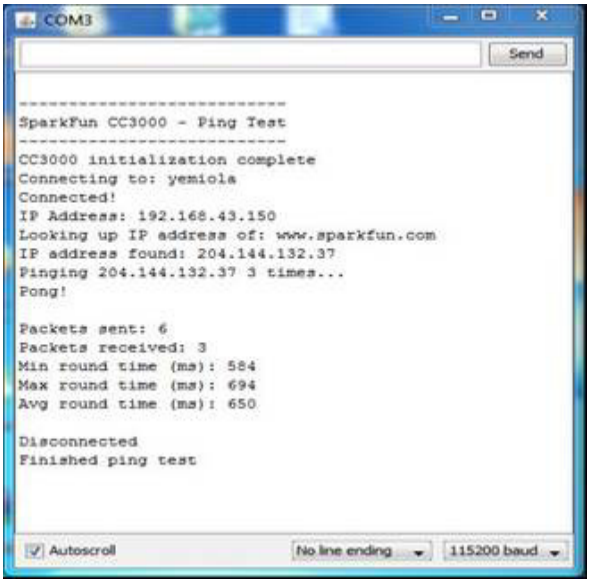

Figure 17 Ping test

The credentials required for these connection includes the network's SSID and the Password access only if the network is a secured network. The information include the MAC Address of the chip, assigned IP address, IP addresses of the DHCP Server, DNS Server, Defaut Gateway and the SSID of the network.

PING TEST

Ping is a network troubleshooting tools used to verify communication between two different hosts in the same network (intranet) or outside the network (Internet). It send an echo message to check if the host if reacheable or not. This test is very important as it assure us that our wifi chips can communicate with a server from diiferent network as in figure 17.

After a connection is achieved, the cc3000 through the access point can ping an IP address or web server and as well as transmit packets and receive packets too. This is only possible when a connection is established between the wifi shield and the access point.

11.4 Web Client Test

Having established a connection the Wi-Fi shield is capable of pinging a web address in figure 18 and retrieving information about the web address as shown in figure 19. The web address in this case is example.com after establishing a connection,

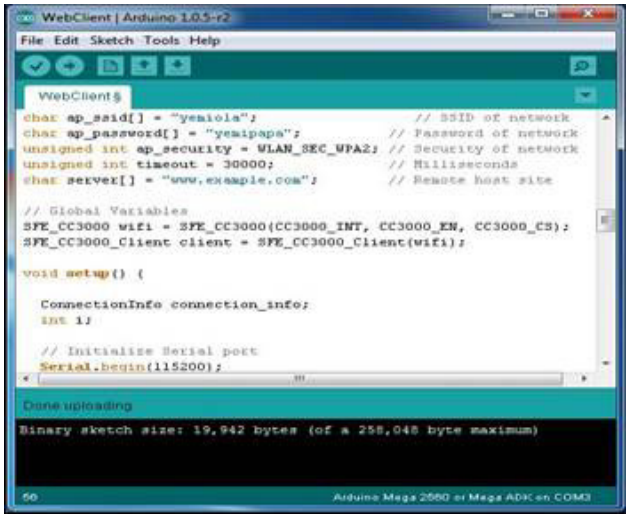

Figure: 18 Web client test source code

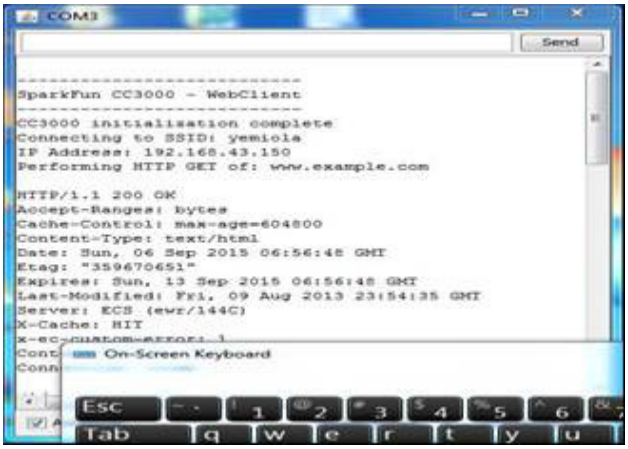

Fig 19 web client test result

12 CONCLUSION

Establishes the interface between the physical world and the web, the work succeeded in developing an interconnection of various stages, and a compatibility was accomplished by running various test like Scan test, Ping test, connection test, web Client test smart Configuration and self-connection test. The success of this test shows that the configuration of all the various stages brought about interconnection between all the stages, thereby making data transfer possible

REFERENCES

[1] James Agajo ,Okhaifoh J. , Onyebuchi Nosiri , Ekwueme Emmanuel Uchenna, Adaptive Monitoring and Localization of Faulty node in a wireless sensor network , American Scientific Research Journal for Engineering, Technology, and Sciences (ASRJETS) available online at http://asrjetsjournal.org/index.php/American_Scientific_Journal/article/view/813n Vol 14, No 1, (2015)

[2] James Agajo, H.C. Inyiama, A.L. Theophilus,) , Remote Monitoring and Estimation of Carbon Monoxide Pollution in Indoor Environment using Wireless Sensor Network via Satellite Pacific Journal of Science and Technology (Volume 12, Number 4) HAWAII (USA) University of Akamai Hawaii , available online at http://www.akamaiuniversity.us/PJST12_2_464.pdf (2012

[3] Smart Config in detail retrieve online www.Smart%20Config%20in%20detail%20(Berg%20Dev%20Center).html (2015)

[4] ADAFRUIT SmartConfig retrieve onlinewww.SmartConfig%20_%20Adafruit%20CC3000%20WiFi%20_%20Adafruit%20Learning%20System.html (2015)

[5] Rick Lesniak , **Adafruit CC3000 WiFi, Adafruit learning system retrieve online** https://learn.adafruit.com/adafruit-cc3000-wifi (2015)

[6] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, "A Survey on Sensor Networks", IEEE Communications Magazine, August,102-114(2002).

[7] P. Ferrari, A. Flammini, D. Marioli, E. Sisinni, A. Taroni, "Wired and wireless sensor networks for industrial applications",Microelectronics Journal, September, 2009, Vol. 40, N. 9, pp. 1322-1336.

[8] P. Ferrari, A. Flammini, M. Rizzi, E. Sisinni, "Improving simulation of wireless networked control systems based on WirelessHART",Computer Standards & Interfaces, November, 2013, Vol. 35, N. 6, pp. 605-615.

[9] R. Kay; F. Mattern, "The Design Space of Wireless Sensor Networks", IEEE Wireless communications, 11 (6), Pp.: 54–61.

[10] P. Mell, T. Grance, "The NIST Definition of Cloud Computing", NIST Special Publication 800-145. Available on-line at http://csrc.nist.gov/publications/nistpubs/800-145/SP800- 145.pdf

[11] Sanjit Kumar Dash, Subasish Mohapatra, Prasant Kumar Pattnaik, "A survey on Application of Wireless sensor network using Cloud Computing", IJCSET, Vol. 1,Issue 4, Pp.50- 55,December 2010.

[12] P. Jaworski, T. Edwards, J. Moore, K. Burnham, "Cloud Ccomputing concept for Intelligent Transportation Systems", 2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC), pp.391,936, 5-7 Oct. 2011.