#include "mbed.h"

#include "hcsr04.h"

#include "ESP8266.h"

#include "math.h"

#define CloudIP "184.106.153.149"           //Raw IP Address of ThingSpeak Cloud Server

DigitalOut RLed(LED1);                      //Onboard Red LED = Shelf Out of Stock

DigitalOut GLed(LED2);                      //Onboard Green LED = All OK

DigitalOut BLed(LED3);                      //Onboard Blue LED for Wifi Tx Indication

HCSR04 usensor1(D8,D9);                     //ECHO Pin=D9, TRIG Pin=D8

HCSR04 usensor2(D7,D6);                     //ECHO Pin=D7, TRIG Pin=D6

Serial pc(USBTX,USBRX);                     //Serial Communication with PC

ESP8266 wifi(PTC17, PTC16, 115200);         //Tx Pin:PTC17; Rx Pin:PTC17; Baud rate:115200

void wifi\_send(void);;                      //Connect and Push Data Channel to Cloud Server

int num = 0;

int distance1, distance2;

float dist\_remaining1, dist\_percent1, dist\_remaining2, dist\_percent2;

char snd[255],rcv[1000];                    //snd: send command to ESP8266

                                            //rcv: receive response from ESP8266

int main()

{

    pc.baud(115200);                        //Baud Rate of 115200 for Tera Term

    pc.printf("########  ######  ####  ######      ######  ##    ##  ######  ######## ######## ##     ##\n\r");

    pc.printf("##       ##    ##  ##  ##    ##    ##    ##  ##  ##  ##    ##    ##    ##       ###   ###\n\r");

    pc.printf("##       ##        ##  ##          ##         ####   ##          ##    ##       #### ####\n\r");

    pc.printf("######   ##        ##   ######      ######     ##     ######     ##    ######   ## ### ##\n\r");

    pc.printf("##       ##        ##        ##          ##    ##          ##    ##    ##       ##     ##\n\r");

    pc.printf("##       ##    ##  ##  ##    ##    ##    ##    ##    ##    ##    ##    ##       ##     ##\n\r");

    pc.printf("########  ######  ####  ######      ######     ##     ######     ##    ######## ##     ##\n\r");

    pc.printf("-----------------------------------------------------------------------------------------\n\r");

    pc.printf("Project By: Priyank Kalgaonkar, Sahil Kumar, Linknath Surya Balasubramanian\n\r");

    pc.printf("-----------------------------------------------------------------------------------------\n\r\n\r");

    pc.printf("Initial Setup\r\n");

    wifi.SetMode(1);                        //Set ESP mode to 1

    wifi.RcvReply(rcv, 1000);               //Receive a response from ESP

    pc.printf("%s\r", rcv);

    pc.printf("Conneting to WiFi\r\n");     //AP Setup Initialization

    wifi.Join("Priyank's iPhone", "audi1155");

    wifi.RcvReply(rcv, 1000);

    pc.printf("%s\n", rcv);

    wait(8);

    wifi.GetIP(rcv);                        //Obtains an IP address from the AP

    while (1)

    {

        wifi\_send();

        RLed = 1;

        GLed = 1;

        BLed = 0;

        wait(2.0f);

    }

}

void wifi\_send(void)

{

    while(num<1000000000000)

    {

        num=num+1;

        pc.printf("Cloud Sync Instance #: %d\n\r", num);

        pc.printf("Syncing Data with Cloud, Please Wait.\n\r");

    //Ultrasound Sensor (HC-SR04) #1 Initialization

        int a = 30;

        usensor1.start();

        wait\_ms(500);

    //Calculating Distance Percentage Remaining for Sensor # 1

        distance1 = usensor1.get\_dist\_cm();

        dist\_remaining1 = a-distance1;

        dist\_percent1 = (dist\_remaining1/30)\*100;

    //Ultrasound Sensor (HC-SR04) #2 Initialization

        int b = 30;

        usensor2.start();

        wait\_ms(500);

    //Calculating Distance Percentage Remaining for Sensor # 2

        distance2 = usensor2.get\_dist\_cm();

        dist\_remaining2 = b-distance2;

        dist\_percent2 = (dist\_remaining2/30)\*100;

    //LED and Tera Term Output

        if (distance1<30 && distance2<30) {

            RLed = 1;

            BLed = 1;

            GLed = 0;

            //printf("Percent remaining: %f\r", dist\_percent1 && dist\_percent2);

        } else {

            GLed = 1;

            BLed = 1;

            RLed = 0;

            printf("Shelves Empty! Replenish Stock.\n\r");

        }

    //Sending Data to the Cloud Server via ESP8266 WiFi Module

        strcpy(snd,"AT+CIPMUX=0\n\r");        //AT+CIPMUX: Enabling Single Channel Mode

        wifi.SendCMD(snd);

        wait(1);

        wifi.RcvReply(rcv, 1000);

        wait(1);

        sprintf(snd,"AT+CIPSTART=4,\"TCP\",\"%s\",80\n",CloudIP); //Establish TCP connection w/ Cloud Server

        wait(1);

        wifi.RcvReply(rcv, 1000);

        wait(1);

        strcpy(snd,"AT+CIPSEND=100\n\r");    //Set length of the data that will be sent

        wifi.SendCMD(snd);

        pc.printf("%s\r", rcv);

        wait(1);

        wifi.RcvReply(rcv, 1000);

        pc.printf("%s\r", rcv);

        wait(1);

    //Pushing the data acquired from HC-SR04 Ultrasonic Sensor to Cloud Server via API

        pc.printf("Product X - Sensor 1: ");

        sprintf(snd,"GET [https://api.thingspeak.com/update?api\_key=O59NSRSQZCJ2G6WK&field1=%f](https://api.thingspeak.com/update?api_key=O59NSRSQZCJ2G6WK&field1=%25f)\r", dist\_percent1);

        printf("Product X: Percent Stock Remaining: %f\n\r", dist\_percent1);

        wifi.SendCMD(snd);

        pc.printf("%s\r",snd);

        wait(1);

        wifi.RcvReply(rcv, 1000);

        pc.printf("%s\r", rcv);

        pc.printf("Product Y - Sensor 2: ");

        sprintf(snd,"GET [https://api.thingspeak.com/update?api\_key=O59NSRSQZCJ2G6WK&field2=%f](https://api.thingspeak.com/update?api_key=O59NSRSQZCJ2G6WK&field2=%25f)\n\r\n\r", dist\_percent2);

        wifi.SendCMD(snd);

        pc.printf("%s\r",snd);

        printf("Product Y: Percent Stock Remaining: %f\n\r\n\r", dist\_percent2);

        wait(1);

        wifi.RcvReply(rcv, 1000);

        pc.printf("%s\r", rcv);

    }

}