

APPENDIX

Code:

```
void uploadstatus(){//calling RESTful API to upload datapoint to MCS  
to report LED status
```

```
while (!c2.connect(SITE_URL, 80))
```

```
{
```

```
  Serial.print(".");
```

```
  delay(500);
```

```
}
```

```
delay(100);
```

```
if(digitalRead(13)==1)
```

```
  upload_led = "LED_DISPLAY,,1";
```

```
else
```

```
  upload_led = "LED_DISPLAY,,0";
```

```
int thislength = upload_led.length();
```

```
HttpClient http(c2);
```

```
c2.print("POST /mcs/v2/devices/");
```

```
c2.print(DEVICEID);
```

```
c2.println("/datapoints.csv HTTP/1.1");
```

```
c2.print("Host: ");
```

```
c2.println(SITE_URL);
c2.print("deviceKey: ");
c2.println(DEVICEKEY);
c2.print("Content-Length: ");
c2.println(thislength);
c2.println("Content-Type: text/csv");
c2.println("Connection: close");
c2.println();
c2.println(upload_led);
delay(500);

int errorcount = 0;
while (!c2.available())
{
  Serial.print(".");
  delay(100);
}

int err = http.skipResponseHeaders();
int bodyLen = http.contentLength();

while (c2)
{
  int v = c2.read();
  if (v != -1)
```

```
{  
  Serial.print(char(v));  
}  
else  
{  
  Serial.println("no more content, disconnect");  
  c2.stop();  
}  
  
}  
Serial.println();  
}
```

```
void outzone(){  
  
while (!c2.connect(SITE_URL, 80))  
{  
  Serial.print(".");  
  delay(500);  
}  
  
delay(100);
```

```
if(digitalRead(8)==1)
    upload_ledzone = "OUT_ZONE,,1";
else
    upload_ledzone = "OUT_ZONE,,0";

int zonelength = upload_ledzone.length();
HttpClient http(c2);
c2.print("POST /mcs/v2/devices/");
c2.print(DEVICEID);
c2.println("/datapoints.csv HTTP/1.1");
c2.print("Host: ");
c2.println(SITE_URL);
c2.print("deviceKey: ");
c2.println(DEVICEKEY);
c2.print("Content-Length: ");
c2.println(zonelength);
c2.println("Content-Type: text/csv");
c2.println("Connection: close");
c2.println();
c2.println(upload_ledzone);
delay(500);

int errorcount = 0;
while (!c2.available())
```

```
{
  Serial.print(".");
  delay(100);
}

int err = http.skipResponseHeaders();
int bodyLen = http.contentLength();

while (c2)
{
  int v = c2.read();
  if (v != -1)
  {
    Serial.print(char(v));
  }
  else
  {
    Serial.println("no more content, disconnect");
    c2.stop();
  }

}

Serial.println();
}
```

```
void uploadGPS(){

    while (!c2.connect(SITE_URL, 80))

    {

        Serial.print(".");

        delay(500);

    }

    delay(100);

    float latitude_post=latitude;

    float longitude_post=longitude;

    Serial.printf("latitude=%.4f\tlongitude=%.4f\n",latitude,longitude);

    if(latitude>-90 && latitude<=90 && longitude>=0 &&

longitude<360){

        sprintf(buffer_latitude, "%.4f", latitude);

        sprintf(buffer_longitude, "%.4f", longitude);

    }

    String upload_GPS =

"GPS,,"+String(buffer_latitude)+","+String(buffer_longitude)+","+"0"+"

\n"+"LATITUDE,,"+buffer_latitude+"\n"+"LONGITUDE,,"+buffer_lon

gitude;//null altitude

    int GPS_length = upload_GPS.length();

    HttpClient http(c2);
```

```
c2.print("POST /mcs/v2/devices/");
c2.print(DEVICEID);
c2.println("/datapoints.csv HTTP/1.1");
c2.print("Host: ");
c2.println(SITE_URL);
c2.print("deviceKey: ");
c2.println(DEVICEKEY);
c2.print("Content-Length: ");
c2.println(GPS_length);
c2.println("Content-Type: text/csv");
c2.println("Connection: close");
c2.println();
c2.println(upload_GPS);
delay(500);

if(latitude<15.5343){
  digitalWrite(8,HIGH);
}
else{
  digitalWrite(8,LOW);
}

LSMS.beginSMS("0925376867");
```

```
int errorcount = 0;

while (!c2.available())
{
    Serial.print(".");
    delay(100);
}

int err = http.skipResponseHeaders();
int bodyLen = http.contentLength();

while (c2)
{
    int v = c2.read();
    if (v != -1)
    {
        Serial.print(char(v));
    }
    else
    {
        Serial.println("no more content, disconnect");
        c2.stop();
    }
}
```



```
    }  
    Serial.println();  
  }  
  
void uploadspeed(){  
  
    while (!c2.connect(SITE_URL, 80))  
    {  
        Serial.print(".");  
        delay(500);  
    }  
  
    delay(100);  
  
    float speed_post=speeda;  
    Serial.printf("speeda=%.4f\n",speeda);  
    sprintf(buffer_speed, "%.4f", speeda);  
  
    float time=5;  
    float speed_temp=sqrt((latitude*latitude-longitude*longitude))/time;  
  
    String upload_speed = "SPEED,,"+String(buffer_speed);
```

```
int speed_length = upload_speed.length();
c2.println("POST /mcs/v2/devices/");
c2.println(DEVICEID);
c2.println("/datapoints.csv HTTP/1.1");
c2.println("Host: ");
c2.println(SITE_URL);
c2.println("deviceKey: ");
c2.println(DEVICEKEY);
c2.println("Content-Length: ");
c2.println(speed_length);
c2.println("Content-Type: text/csv");
c2.println("Connection: close");
c2.println();
c2.println(upload_speed);
delay(500);
int errorcount = 0;

while (!c2.available())
{
  Serial.print(".");
  delay(100);
}

int err = http.skipResponseHeaders();
```

```
int bodyLen = http.contentLength();

while (c2)
{
    int v = c2.read();
    if (v != -1)
    {
        Serial.print(char(v));
    }
    else
    {
        Serial.println("no more content, disconnect");
        c2.stop();
    }

}

Serial.println();
}
```

GPS:

```
static unsigned char getComma(unsigned char num,const char *str)
{
    unsigned char i,j = 0;
    int len=strlen(str);
```

```
for(i = 0;i < len;i ++)  
{  
    if(str[i] == ',')  
        j++;  
    if(j == num)  
        return i + 1;  
}  
return 0;  
}  
  
static double getDoubleNumber(const char *s)  
{  
    char buf[10];  
    unsigned char i;  
    double rev;  
  
    i=getComma(1, s);  
    i = i - 1;  
    strncpy(buf, s, i);  
    buf[i] = 0;  
    rev=atof(buf);  
    return rev;  
}
```

```
static double getIntNumber(const char *s)
```

```
{  
    char buf[10];  
    unsigned char i;  
    double rev;  
  
    i=getComma(1, s);  
    i = i - 1;  
    strncpy(buf, s, i);  
    buf[i] = 0;  
    rev=atoi(buf);  
    return rev;  
}
```

```
void parseGPGGA(const char* GPGGAstr)
```

```
{  
    int tmp, hour, minute, second, num ;  
    if(GPGGAstr[0] == '$')  
    {  
        tmp = getComma(1, GPGGAstr);  
        hour   = (GPGGAstr[tmp + 0] - '0') * 10 + (GPGGAstr[tmp + 1] -  
'0');  
        minute = (GPGGAstr[tmp + 2] - '0') * 10 + (GPGGAstr[tmp + 3] -  
'0');
```

```
second = (GPGGAstr[tmp + 4] - '0') * 10 + (GPGGAstr[tmp + 5] - '0');
```

```
printf(buff, "UTC timer %2d-%2d-%2d", hour, minute, second);
```

```
//Serial.println(buff);
```

```
tmp = getComma(2, GPGGAstr);
```

```
latitude = getDoubleNumber(&GPGGAstr[tmp])/100.0;
```

```
int latitude_int=floor(latitude);
```

```
double latitude_decimal=(latitude-latitude_int)*100.0/60.0;
```

```
latitude=latitude_int+latitude_decimal;
```

```
tmp = getComma(4, GPGGAstr);
```

```
longitude = getDoubleNumber(&GPGGAstr[tmp])/100.0;
```

```
int longitude_int=floor(longitude);
```

```
double longitude_decimal=(longitude-longitude_int)*100.0/60.0;
```

```
longitude=longitude_int+longitude_decimal;
```

```
printf(buff, "latitude = %10.4f, longitude = %10.4f", latitude, longitude);
```

```
//Serial.println(buff);
```

```
tmp = getComma(7, GPGGAstr);
```

```
num = getIntNumber(&GPGGAstr[tmp]);
```

```
printf(buff, "satellites number = %d", num);
```

```
//Serial.println(buff);
```

```
    }  
    else  
    {  
        Serial.println("Not get data");  
    }  
}  
  
void GPS_receive() {  
    LGPS.getData(&info);  
    //Serial.println((char*)info.GPGGA);  
    parseGPGGA((const char*)info.GPGGA);  
}
```

connect:

```
void AP_connect(){  
    Serial.print("Connecting to AP...");  
    while (0 == LWiFi.connect(WIFI_AP))  
    {  
        Serial.print(".");  
        delay(500);  
    }  
    Serial.println("Success!");  
}
```

```
Serial.print("Connecting site...");

while (!c2.connect(SITE_URL, 80))
{
  Serial.print(".");
  delay(500);
}
Serial.println("Success!");
delay(100);
}

void getconnectInfo(){
  //calling RESTful API to get TCP socket connection
  c2.print("GET /mcs/v2/devices/");
  c2.print(DEVICEID);
  c2.println("/connections.csv HTTP/1.1");
  c2.print("Host: ");
  c2.println(SITE_URL);
  c2.print("deviceKey: ");
  c2.println(DEVICEKEY);
  c2.println("Connection: close");
  c2.println();
```



```
delay(500);

int errorcount = 0;
Serial.print("waiting for HTTP response...");
while (!c2.available())
{
  Serial.print(".");
  errorcount += 1;
  delay(150);
}
Serial.println();
int err = http.skipResponseHeaders();

int bodyLen = http.contentLength();
char c;
int ipcount = 0;
int count = 0;
int separater = 0;
while (c2)
{
  int v = (int)c2.read();
  if (v != -1)
  {
    c = v;
```

```
//Serial.print(c);
connection_info[ipcount]=c;
if(c==' ')
  separater=ipcount;
  ipcount++;
}
else
{
  Serial.println("no more content, disconnect");
  c2.stop();

}

}

//connection_info[ipcount]=NULL;
int i;
for(i=0;i<separater;i++)
{ ip[i]=connection_info[i];
}
int j=0;
separater++;

for(i=separater;i<21 && j<5 && i < ipcount;i++)
```

```
{ port[j]=connection_info[i];
  j++;
}
//port[j] = NULL;

portnum = atoi (port);

} //getconnectInfo

void connectTCP(){
  //establish TCP connection with TCP Server with designate IP and Port
  c.stop();
  Serial.print("Connecting to TCP...");
  while (0 == c.connect(ip, portnum))
  {
    Serial.println("Re-Connecting to TCP");
    delay(1000);
  }
  c.println(tcpdata);
  c.println();
  Serial.println("Success!");
} //connectTCP
```

```
void heartBeat(){
  Serial.println("send TCP heartBeat");
  c.println(tcpdata);
  c.println();
} //heartBeat
```

Upload:

```
void uploadstatus(){//calling RESTful API to upload datapoint to MCS
to report LED status
```

```
while (!c2.connect(SITE_URL, 80))
{
  Serial.print(".");
  delay(500);
}
```

```
delay(100);
```

```
if(digitalRead(13)==1)
  upload_led = "LED_DISPLAY,,1";
else
  upload_led = "LED_DISPLAY,,0";
```

```
int thislength = upload_led.length();

HttpClient http(c2);

c2.print("POST /mcs/v2/devices/");

c2.print(DEVICEID);

c2.println("/datapoints.csv HTTP/1.1");

c2.print("Host: ");

c2.println(SITE_URL);

c2.print("deviceKey: ");

c2.println(DEVICEKEY);

c2.print("Content-Length: ");

c2.println(thislength);

c2.println("Content-Type: text/csv");

c2.println("Connection: close");

c2.println();

c2.println(upload_led);

delay(500);

int errorcount = 0;

while (!c2.available())

{

  Serial.print(".");

  delay(100);

}

int err = http.skipResponseHeaders();
```

```
int bodyLen = http.contentLength();

while (c2)
{
    int v = c2.read();
    if (v != -1)
    {
        Serial.print(char(v));
    }
    else
    {
        Serial.println("no more content, disconnect");
        c2.stop();
    }

}

Serial.println();
}

void outzone(){

while (!c2.connect(SITE_URL, 80))
{
```

```
Serial.print(".");  
  
delay(500);  
  
}  
  
delay(100);  
  
if(digitalRead(8)==1)  
    upload_ledzone = "OUT_ZONE,,1";  
else  
    upload_ledzone = "OUT_ZONE,,0";  
  
int zonelength = upload_ledzone.length();  
HttpClient http(c2);  
c2.print("POST /mcs/v2/devices/");  
c2.print(DEVICEID);  
c2.println("/datapoints.csv HTTP/1.1");  
c2.print("Host: ");  
c2.println(SITE_URL);  
c2.print("deviceKey: ");  
c2.println(DEVICEKEY);  
c2.print("Content-Length: ");  
c2.println(zonelength);  
c2.println("Content-Type: text/csv");  
c2.println("Connection: close");
```

```
c2.println();
c2.println(upload_ledzone);
delay(500);

int errorcount = 0;
while (!c2.available())
{
  Serial.print(".");
  delay(100);
}

int err = http.skipResponseHeaders();
int bodyLen = http.contentLength();

while (c2)
{
  int v = c2.read();
  if (v != -1)
  {
    Serial.print(char(v));
  }
  else
  {
    Serial.println("no more content, disconnect");
    c2.stop();
  }
}
```



```
    }

}

Serial.println();
}

void uploadGPS(){

while (!c2.connect(SITE_URL, 80))

{

Serial.print(".");

delay(500);

}

delay(100);

float latitude_post=latitude;

float longitude_post=longitude;

Serial.printf("latitude=%.4f\tlongitude=%.4f\n",latitude,longitude);

if(latitude>-90 && latitude<=90 && longitude>=0 &&
longitude<360){

sprintf(buffer_latitude, "%.4f", latitude);

sprintf(buffer_longitude, "%.4f", longitude);
```

```
    }

    String upload_GPS =
    "GPS,,"+String(buffer_latitude)+",,"+String(buffer_longitude)+",,"+"0"+"
    \n"+"LATITUDE,,"+buffer_latitude+"\n"+"LONGITUDE,,"+buffer_lon
    gitude;//null altitude

    int GPS_length = upload_GPS.length();

    HttpClient http(c2);

    c2.print("POST /mcs/v2/devices/");

    c2.print(DEVICEID);

    c2.println("/datapoints.csv HTTP/1.1");

    c2.print("Host: ");

    c2.println(SITE_URL);

    c2.print("deviceKey: ");

    c2.println(DEVICEKEY);

    c2.print("Content-Length: ");

    c2.println(GPS_length);

    c2.println("Content-Type: text/csv");

    c2.println("Connection: close");

    c2.println();

    c2.println(upload_GPS);

    delay(500);

if(latitude<15.5343){

    digitalWrite(8,HIGH);
```

```
}  
else{  
    digitalWrite(8,LOW);  
}  
  
LSMS.beginSMS("0925376867");  
  
int errorcount = 0;  
  
while (!c2.available())  
{  
    Serial.print(".");  
    delay(100);  
}  
  
int err = http.skipResponseHeaders();  
int bodyLen = http.contentLength();  
  
while (c2)  
{  
    int v = c2.read();  
    if (v != -1)  
    {  
        Serial.print(char(v));  
    }  
}
```

```
    }  
    else  
    {  
        Serial.println("no more content, disconnect");  
        c2.stop();  
    }  
  
    }  
    Serial.println();  
}
```

```
void uploadspeed(){  
  
    while (!c2.connect(SITE_URL, 80))  
    {  
        Serial.print(".");  
        delay(500);  
    }  
}
```

```
delay(100);

float speed_post=speeda;
Serial.printf("speeda=%.4f\n",speeda);
sprintf(buffer_speed, "%.4f", speeda);

float time=5;
float speed_temp=sqrt((latitude*latitude-longitude*longitude))/time;

String upload_speed = "SPEED,,"+String(buffer_speed);

int speed_length = upload_speed.length();
c2.print("POST /mcs/v2/devices/");
c2.print(DEVICEID);
c2.println("/datapoints.csv HTTP/1.1");
c2.print("Host: ");
c2.println(SITE_URL);
c2.print("deviceKey: ");
c2.println(DEVICEKEY);
c2.print("Content-Length: ");
c2.println(speed_length);
c2.println("Content-Type: text/csv");
```

```
c2.println("Connection: close");  
c2.println();  
c2.println(upload_speed);  
delay(500);
```

```
int errorcount = 0;
```

```
while (!c2.available())
```

```
{  
    Serial.print(".");  
    delay(100);
```

```
}
```

```
int err = http.skipResponseHeaders();
```

```
int bodyLen = http.contentLength();
```

```
while (c2)
```

```
{  
    int v = c2.read();
```

```
    if (v != -1)
```

```
    {
```

```
    Serial.print(char(v));  
  }  
  else  
  {  
    Serial.println("no more content, disconnect");  
    c2.stop();  
  }  
  
}  
Serial.println();  
}
```