

# SM5033: Internet of Things

## Embedded C Coding Assignment

Deevanshu Mukesh Gupta

SM21MTECH12014

---

**GitHub Link for the Assignment:**

<https://github.com/deviith/SM5033>

### Programming Assignment

# Program for Embedded C. Add components and write the code accordingly to perform the following system. <https://simulator.mbed.com>

To the given board, connect the following components.

- 1) an sht31 temperature and humidity sensor
- 3) a red LED.
- 4) a blue LED.
- 5) a white LED.
- 7) a button.

The system should perform the following tasks continuously.

- 1) White LED should be blinking every 2 seconds. (0.5 seconds on time, 2 seconds off time)  
**(3 marks)**

- 2) Read temperature and humidity values every 10 seconds (not continuously). The intensity of red LED and blue LED should change depending on temperature and humidity values, respectively. (higher the value higher intensity the corresponding led should have)  
**(7 marks)**

- 3) On button press, take new readings immediately, and update both LED intensity.  
**(5 marks)**

Submit a report (slides or document) explaining your code. Also submit the video of code working. **Plagiarism will result in strict action.**

## Video Link for Assignment:

<https://drive.google.com/file/d/1J104gEaYLbEw1DzSoaURD8do8BrHJ-qT/view?usp=sharing>

## Code :

```
#include "mbed.h"
#include "Sht31.h"

Ticker Tensecond;

DigitalOut white_led(p5);

PwmOut red_led(p6);
PwmOut blue_led(p7);

InterruptIn pushbutton(p8);

Sht31 sht31(I2C_SDA, I2C_SCL);

void reading_value(){

    float temp = sht31.readTemperature();
    float humidity = sht31.readHumidity();

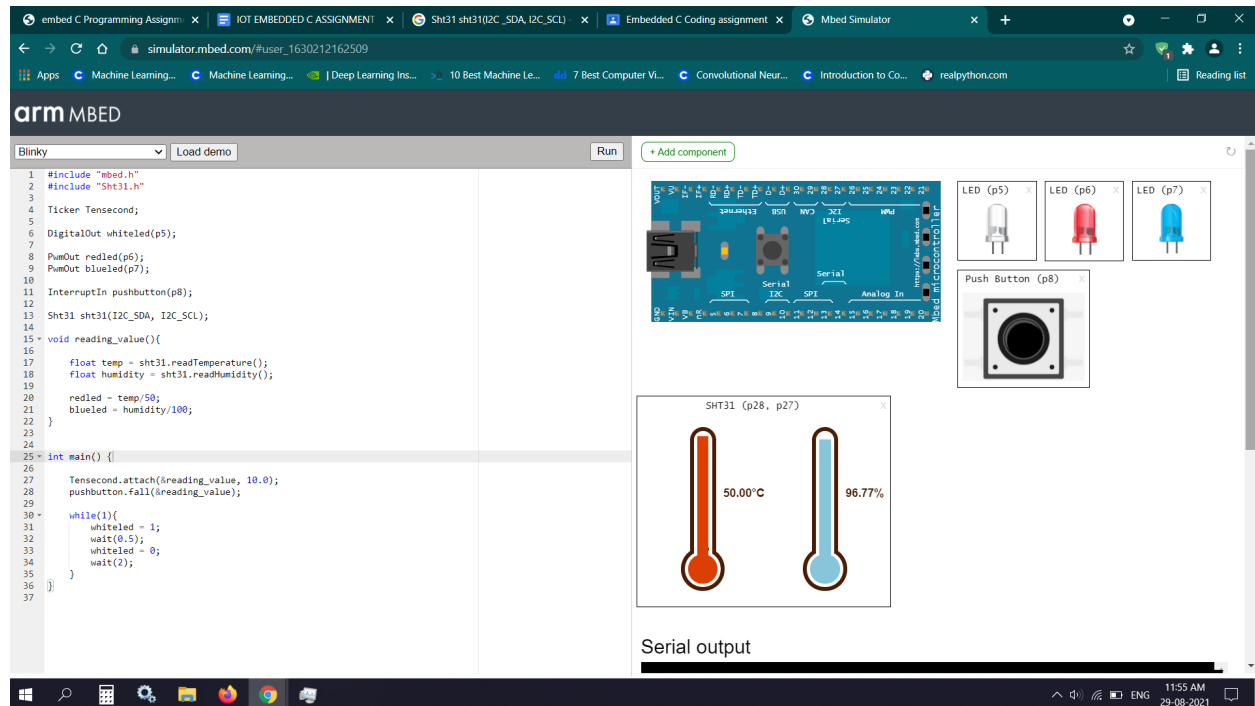
    red_led = temp/50;
    blue_led = humidity/100;
}

int main() {

    Tensecond.attach(&reading_value, 10.0);
    pushbutton.fall(&reading_value);

    while(1){
        white_led = 1;
        wait(0.5);
        white_led = 0;
        wait(2);
    }
}
```

## Mbed Online Simulator Screen:



## Explanation :

<pre>#include "mbed.h" #include "Sht31.h"</pre>	Following command copy everything from that library 'mbed.h' and 'Sht31.h' and paste it here.
<pre>Ticker Tensecond;</pre>	The Ticker interface is used to set up a recurring interrupt by <i>Tensecond</i> to repeatedly call a function at a specified rate.
<pre>DigitalOut white_led(p5);</pre>	The DigitalOut pin <i>p5</i> is initialize with ' <i>white_led</i> ' variable to control the pin output
<pre>PwmOut red_led(p6); PwmOut blue_led(p7);</pre>	The PwmOut is used to control the frequency of output <i>p6</i> and <i>p7</i> are initialized using the <i>red_led</i> and <i>blue_led</i> variable.

InterruptIn <code>pushbutton(p8);</code>	The interrupt interface is used to trigger an event when an digital input pin changes <i>p8</i> button is initialized using <i>pushbutton</i> variable.
Sht31 <code>sht31(I2C_SDA, I2C_SCL);</code>	It is used to initialize the Sht31 temperature and humidity sensor using the <i>sht31</i> variable.
<code>void reading_value(){     float temp = sht31.readTemperature();     float humidity = sht31.readHumidity();     red_led = temp/50;     blue_led = humidity/100; }</code>	We created a function <i>reading_value()</i> to take the reading and put the value of pwm led red and blue when called.
<code>float temp = sht31.readTemperature(); float humidity = sht31.readHumidity();</code>	We initialize the float variable <i>temp</i> and <i>humidity</i> to read the temperature and humidity value using the sht31 sensor.
<code>red_led = temp/50; blue_led = humidity/100;</code>	We had created a logic to fix the led intensity according to the temp and humidity max value. e.g if <i>temp</i> is 25 deg then <i>red_led</i> output is 0.5 means 50% of its full intensity. Similarly for humidity.
<code>Tensecond.attach(&amp;reading_value, 10.0);</code>	With the help of ticker, this command calls the <i>reading_value()</i> function every 10 seconds.
<code>pushbutton.fall(&amp;reading_value);</code>	With the help of Interrupt, this command calls the <i>reading_value()</i> function whenever the push button is pressed.
<code>while(1){     white_led = 1;     wait(0.5);     white_led = 0;     wait(2); }</code>	This will create the infinite loop in which the white led will get ON for 0.5 second and OFF for 2 second continuously