Lab Cortex-M4: Serial WIFI
Cortex-M4 Board

- STM32F407 Discovery Board
  - RS232C
  - Serial WIFI
  - 0.96 inch OLED graphic display

- On board ST-LINK JTAG debugging interface
Start STM32CubeIDE
Click X to close
New STM32 Project
Select Board (Not MCU)
Select STM32F4DISCOVERY and click Next
Project Name

STM32 Project

Project Setup

- Directory with specified name already exists

Project Name: Serial
- Use default location
- Location: C:/Users/limdjl/STM32CubeIDE/workspace_1.1.0

Options:
- Targeted Language: C
- Targeted Binary Type: Executable
- Targeted Project Type: STM32Cube

Finish
Enable USART2, USART3
Check if I2C1 is enabled
Generate Code
Open main.c

```c
/* USER CODE BEGIN Header */

/***************************************************************
* @file
* @brief : Main program body
***************************************************************
* @attention
* ...
* This software component is licensed by ST under Ultimate Liberty license
* SLA0044, the "License"; You may not use this file except in compliance with
* the License. You may obtain a copy of the License at:
* www.st.com/SLA0044
* ...
*/

/* USER CODE END Header */

/* Includes ---------------------------------------------*/
#include "main.h"
#include "usb_host.h"

/* Private includes --------------------------------------*/
/* USER CODE BEGIN Includes */
/* USER CODE END Includes */

/* Private typedef --------------------------------------*/
/* USER CODE BEGIN PTD */
/* USER CODE END PTD */
```
Build Project

```
10:37:00 ***** Incremental Build of configuration Debug for project Serial *****
make -j4 all
arm-none-eabi-gcc "../Core/Src/main.c" -mcpu=cortex-m4 -std=gnu11 -g3 -DUSE_HAL
arm-none-eabi-gcc -o "Serial.elf" @"objects.list" -mcpus=cortex-m4 -T"C:\Users
Finished building target: Serial.elf

arm-none-eabi-objdump -h -S Serial.elf > "Serial.list"
arm-none-eabi-size Serial.elf

text data bss dec hex filename
32464 152 3648 36264 8da8 Serial.elf
Finished building: default.size.stdout

Finished building: Serial.list

10:37:12 Build Finished. 0 errors, 0 warnings. (took 5s.800ms)
```
/* Private user code -----------------------------------------------------------------------------*/
/* USER CODE BEGIN 0 */
void PrintString(uint8_t * string)
{
    HAL_UART_Transmit(&huart2, (uint8_t *)string, strlen((char *)string), 0xffff);
}
/* USER CODE END 0 */

/* USER CODE BEGIN 2 */
    PrintString((uint8_t *)"Hello Cortex-M
\n\r");
/* USER CODE END 2 */
#include "main.h"
#include "usb_host.h"
#include "string.h"
int main(void)
{
    /* USER CODE BEGIN 1 */
    /* USER CODE END 1 */

    /* MCU Configuration-----------------------------------------------*/

    /* Reset of all peripherals, Initializes the Flash interface and the Systick. */
    HAL_Init();

    /* USER CODE BEGIN Init */
    /* USER CODE END Init */

    /* Configure the system clock */
    SystemClock_Config();

    /* USER CODE BEGIN SysInit */
    /* USER CODE END SysInit */

    /* Initialize all configured peripherals */
    MX_GPIO_Init();
    MX_I2C1_Init();
    MX_I2S3_Init();
    MX_SPI1_Init();
    MX_USB_HOST_Init();
    MX_USART2_UART_Init();
    MX_USART3_UART_Init();

    /* USER CODE BEGIN 2 */
    PrintString((uint8_t *)"Hello Cortex-M\n\n");
    /* USER CODE END 2 */
Run Debug
Select a way to debug 'WifiWeather':

- Local C/C++ Application
- STM32 Cortex-M C/C++ Application

Description
STM32 Cortex-M C/C++ Application
```c
int main(void) {
    /* USER CODE BEGIN 1 */
    /* USER CODE END 1 */

    /* MCU Configuration ---------------------------------------------*/
    /* Reset of all peripherals, Initializes the Flash interface and the Systick. */
    HAL_Init();

    /* USER CODE BEGIN Init */
    /* USER CODE END Init */

    /* Configure the system clock */
    SystemClock_Config();

    /* USER CODE BEGIN 2 */
    /* USER CODE END 2 */
}
```
Resume

```c
int main(void)
{
    /* USER CODE BEGIN 1 */
    /* USER CODE END 1 */

    /* MCU Configuration */
    /* Reset of all peripherals, Initializes the Flash interface */
    HAL_Init();

    /* USER CODE BEGIN Init */
    /* USER CODE END Init */
```
STMicroelectronics STLink Virtual COM Port (COM13)
USB S/N: 0672F565257867767114235
Baud rate: 115200

Prolific USB-to-Serial Comm Port (COM12)
USB S/N: 981CCB9356&G084
Baud rate: 115200

Connected to COM12 (115200 bps)

Hello Cortex-M
Hello Cortex-M
Exercise 1: Weather Forecast using WIFI Module

STM32F407 Board

Serial WIFI Module

Wireless Broadband Router

www.kma.go.kr
Project Setup

Setup STM32 project

Project Name: WifiWeather

- Use default location

Location: C:/Users/lmldj/STM32CubeIDE/workspace_1.1.0

Options

- Targeted Language
  - C  C++

- Targeted Binary Type
  - Executable  Static Library

- Targeted Project Type
  - STM32Cube  Empty

Finish  Cancel
Serial WIFI Module

ESP8266 ESP-01
## ESP8266 AT Command Set

<table>
<thead>
<tr>
<th>Function</th>
<th>AT Command</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>Working</td>
<td>AT</td>
<td>OK</td>
</tr>
<tr>
<td>Restart</td>
<td>AT+RST</td>
<td>OK [System Ready, Vendor:www.ai-thinker.com]</td>
</tr>
<tr>
<td>Firmware version</td>
<td>AT+GMR</td>
<td>AT+GMR 0018000902 OK</td>
</tr>
<tr>
<td>Join Access Point</td>
<td>AT+CWJAP?</td>
<td>Query AT+CWJAP? +CWJAP:&quot;RochefortSurLac&quot; OK</td>
</tr>
<tr>
<td></td>
<td>AT+CWJAP=&quot;SSID&quot;,&quot;Password&quot;</td>
<td></td>
</tr>
<tr>
<td>Quit Access Point</td>
<td>AT+CWQAP=?</td>
<td>Query OK</td>
</tr>
<tr>
<td></td>
<td>AT+CWQAP</td>
<td></td>
</tr>
<tr>
<td>Get IP Address</td>
<td>AT+CIFSR</td>
<td>AT+CIFSR 192.168.0.105 OK</td>
</tr>
</tbody>
</table>

AT+CIFSR
### ESP8266 AT Command Set

| WiFi Mode | AT+CWMODE?  
| AT+CWMODE=1  
| AT+CWMODE=2  
| AT+CWMODE=3  | Query  
| STA  
| AP  
| BOTH  |

| Set up TCP or UDP connection | AT+CIPSTART=?  
| (CIPMUX=0) AT+CIPSTART = <type>,<addr>,<port>  
| (CIPMUX=1) AT+CIPSTART= <id><type>,<addr>,<port>  | Query  
| id = 0-4, type = TCP/UDP, addr = IP address, port= port  |

| TCP/UDP Connections | AT+ CIPMUX?  
| AT+ CIPMUX=0  
| AT+ CIPMUX=1  | Query  
| Single  
| Multiple  |

| Check join devices' IP | AT+CWLIF  |

| TCP/IP Connection Status | AT+CIPSTATUS  | AT+CIPSTATUS? no this fun  |

| Send TCP/IP data | (CIPMUX=0) AT+CIPSEND=<length>;  
| (CIPMUX=1) AT+CIPSEND= <id>,<length>  |

| Close TCP / UDP connection | AT+CIPCLOSE=<id> or AT+CIPCLOSE  |
Copy Files

- Copy wifi_weather.c file to C:\Users\UserName\STM32CubeIDE\workspace_1.1.0\WifiWeather\Core\Src

- Copy wifi_weather.h file to C:\Users\UserName\STM32CubeIDE\workspace_1.1.0\WifiWeather\Core\Inc
Project Explorer

workspace_1.1.0 - WifiWeather/Core/Src/wifi_weather

File Edit Source Refactor Navigate Search Project

Project Explorer

Serial

WifiWeather

Binaries

Includes

Core

Inc

main.h
stm32f4xx_hal_conf.h
stm32f4xx_it.h
wifi_weather.h

Src

main.c
stm32f4xx_hal_msp.c
stm32f4xx_it.c
syscalls.c
sysmem.c
system stm32f4xx.c
wifi_weather.c

Startup
wifi_weather.c에서 무선 공유기 이름 (SSID), 비밀번호 변경
**ESP8266 AT Instructions**

**AT+CWIAP — Connects to an AP**

[@deprecated] This command is deprecated. Please use AT+CWIAP_CUR or AT+CWIAP_DEF instead.

<table>
<thead>
<tr>
<th>Commands</th>
<th>Query Command:</th>
<th>Set Command:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>AT+CWIAP?</td>
<td>AT+CWIAP=&lt;ssid&gt;,&lt;pwd&gt;[,&lt;bssid&gt;]</td>
</tr>
<tr>
<td></td>
<td>Function: to query the AP to which the ESP8266 Station is already connected.</td>
<td>Function: to set the AP to which the ESP8266 Station needs to be connected.</td>
</tr>
</tbody>
</table>
/* USER CODE BEGIN Includes */
#include "string.h"
#include "wifi_weather.h"
/* USER CODE END Includes */

/* USER CODE BEGIN 0 */
void PrintString(uint8_t * string)
{
    HAL_UART_Transmit(&huart2, (uint8_t *)string, strlen((char *)string), 0xffff);
}
/* USER CODE END 0 */

/* USER CODE BEGIN 2 */
PrintString((uint8_t *)"Hello Cortex-M\n\r");
WifiSetup();
WifiWeather();
/* USER CODE END 2 */
프로그램 실행 결과

SmarTTY - Raw Terminal
Connected to COM11 (115200 bps)  Baud rate: 115200

AT+CIPSTART="TCP","www.kma.go.kr",80
busy p...
OK

AT+CIPSTART="TCP","www.kma.go.kr",80
CONNECT
OK

08:16:03 GMT 21<temp>13.0<wfEn>Clear
08:16:05 GMT 21<temp>13.0<wfEn>Clear
08:16:08 GMT 21<temp>13.0<wfEn>Clear
기술 뉴스

RSS 서비스

RSS는 Really Simple Syndication의 약자로, 웹사이트에서 업데이트된 정보를 쉽게 구독자들에게 제공하기 위해 XML을 기반으로 만들어진 데이터 형식입니다. RSS 서비스를 이용하면 업데이트된 정보를 쉽게 받아볼 수 있습니다.

RSS 서비스 이용하기

RSS 뉴스 루프

1. 구독하기
2. RSS 뉴스 루프보기
3. RSS 뉴스 루프 업데이트

RSS 서비스 이용하기

RSS 뉴스 루프

1. RSS 뉴스 루프 보기
2. RSS 뉴스 루프 업데이트
3. RSS 뉴스 루프 업데이트
동네예보(도표) : 경기도 인산시상록구 사동 [X=58,Y=121]

기상청

201805052000 6 58 121 24 0 16.0 -999.0 -999.0 4 0 흐림 Cloudy 30 0.0 0.0 2.0 4 남 S 75 0.0 0.0 3 1 15.0 18.0 14.0 4 1 비 Rain 60 0.0 0.0 1.8 4 남 S 85 5.0 0.0 6 1 14.0 18.0 14.0 4 1 비 Rain 60 0.0 0.0 2.0 4 남 S 90 5.0 0.0 9 1 15.0 18.0 14.0 4 1 비 Rain 70 0.0 0.0 2.1 2 동 E 85 8.0 0.0 12 1 16.0 18.0 14.0 4 1 비 Rain 80 0.0 0.0 2.8000000000000003 2 동 E 80 8.0 0.0 15 1 17.0 18.0 14.0 4 0 흐림 Cloudy 30 0.0 0.0 2.0 2 동 E 80 8.0 0.0 18 1 18.0 18.0 14.0 4 0 흐림 Cloudy 30 0.0 0.0 2.5 2 동 E 75 0.0 0.0 21 1 16.0 18.0 14.0 4 0 흐림 Cloudy 30 0.0 0.0 1.8 1 북동 NE 80 0.0 0.0 24 1 15.0 18.0 14.0 4 0 흐림 Cloudy 30 0.0 0.0 1.1 1 북 N 85 0.0 0.0 3 2 14.0 25.0 13.0 4 0 흐림 Cloudy 30 0.0 0.0 1.6 0 북 N 85 0.0 0.0 6 2 13.0 25.0 13.0 4 0 흐림 Cloudy 30 0.0 0.0 1.4000000000000001 0 북 N 85 0.0 0.0 9 2 18.0 25.0 13.0 3 0 구름 많음 Mostly Cloudy 20 0.0 0.0 1.0 0 북 N 55 0.0 0.0 12 2 23.0 25.0 13.0 3 0 구름 많음 Mostly Cloudy 20 0.0 0.0 2.1 0 북 N 35 0.0 0.0 15 2 25.0 25.0 13.0 3 0 구름 많음 Mostly Cloudy 20 0.0 0.0 3.5 7 북서 NW 30 0.0 0.0 18 2 21.0 25.0 13.0 2 0 구름 조금 Partly Cloudy 10 0.0 0.0 2.8000000000000003 7 북서 NW 45 0.0 0.0 21 2 17.0 25.0 13.0 1 0 맑음 Clear 0 0.0 0.0 2.6 7 북서 NW 65 0.0 0.0 24 2 14.0 25.0 13.0 1 0 맑음 Clear 0 0.0 0.0 1.8 7 북서 NW 75 0.0 0.0
Temperature, Weather Forecast, Humidity
TCP Connection

```c
strcpy((char *)string, (char *)"AT+CIPSTART="TCP","www.kma.go.kr",80");
length = strlen((char *)string);
string[length] = 0x0D;
string[length + 1] = 0x0A;
string[length + 2] = 0;
HAL_UART_Transmit(&huart3, (uint8_t *)string, length + 2, 0xFFFF);
HAL_UART_Receive(&huart3, (uint8_t *)buffer, 100, 4000);
HAL_UART_Transmit(&huart2, (uint8_t *)buffer, 100, 0xFFFF);
string[0] = 0x0D;
string[1] = 0x0A;
HAL_UART_Transmit(&huart2, (uint8_t *)string, 2, 0xFFFF);
```

AT+CIPSTART—Establishes TCP Connection, UDP Transmission or SSL Connection

<table>
<thead>
<tr>
<th>Set Command</th>
<th>Single TCP connection (AT+CIPMUX=0):</th>
<th>Multiple TCP Connections (AT+CIPMUX=1):</th>
</tr>
</thead>
<tbody>
<tr>
<td>AT+CIPSTART=&lt;type&gt;,&lt;remote IP&gt;,&lt;remote port&gt;[,&lt;TCP keep alive&gt;]</td>
<td>AT+CIPSTART=&lt;link ID&gt;,&lt;type&gt;,&lt;remote IP&gt;,&lt;remote port&gt;[,&lt;TCP keep alive&gt;]</td>
<td></td>
</tr>
</tbody>
</table>
```c
sprintf((char *)query, "GET /wid/queryDFSRSS.jsp?zone=4127152500 HTTP/1.1\r\nHost: www.kma.go.kr
queryLength = strlen((char *)query);

for (int i = 0; i < 50; i++)buffer[i] = ' ';
my_itoa(queryLength, queryStringLength, 3);
sprintf((char *)string, "AT+CIPSEND=%s", queryStringLength);
length = strlen((char *)string);
string[length] = 0x0D;
string[length + 1] = 0x0A;
string[length + 2] = 0;
HAL_UART_Transmit(&huart3, (uint8_t *)string, length + 2, 0xFFFF);
HAL_UART_Receive(&huart3, (uint8_t *)buffer, 23, 4000);

for (int i = 0; i < 2000; i++)buffer[i] = ' ';
HAL_Delay(1);
HAL_UART_Transmit(&huart3, (uint8_t *)query, queryLength, 0xFFFF);
HAL_UART_Receive(&huart3, (uint8_t *)buffer, 2000, 4000);
//HAL_UART_Transmit(&huart2, (uint8_t *)buffer, 2000, 0xFFFF);
string[0] = 0x0D;
string[1] = 0x0A;
HAL_UART_Transmit(&huart2, (uint8_t *)string, 2, 0xFFFF);
```

**AT+CIPSEND—Sends Data**

- **Set Command:**
  1. Single connection: (+CIPMUX=0)
     
  AT+CIPSEND=<length>
  2. Multiple connections: (+CIPMUX=1)
     
  AT+CIPSEND=<link ID>,<length>
  3. Remote IP and ports can be set in UDP transmission:
     
  AT+CIPSEND=[<link ID>,][<length> [,<remote IP>,<remote port>]]

- **Function:** to configure the data length in normal transmission mode.
ret1 = (int)strstr((char *)buffer, (char *)"GMT");
HAL_UART_Transmit(&huart2, (uint8_t *)(ret1 -9), 12, 0xFFFF);

ret1 = (int)strstr((char *)buffer, (char *)"<hour>");
ret2 = (int)strstr((char *)buffer, (char *)"</hour>");
HAL_UART_Transmit(&huart2, (uint8_t *)(ret1), ret2 - ret1, 0xFFFF);

ret1 = (int)strstr((char *)buffer, (char *)"<temp>");
ret2 = (int)strstr((char *)buffer, (char *)"</temp>");
HAL_UART_Transmit(&huart2, (uint8_t *)(ret1), ret2 - ret1, 0xFFFF);

ret1 = (int)strstr((char *)buffer, (char *)"<wFEn>");
ret2 = (int)strstr((char *)buffer, (char *)"</wFEn>");
HAL_UART_Transmit(&huart2, (uint8_t *)(ret1), ret2 - ret1, 0xFFFF);

string[0] = 0x0D;
string[1] = 0x0A;
HAL_UART_Transmit(&huart2, (uint8_t *)string, 2, 0xFFFF);
/* wait for a while to slow down */
HAL_Delay(2000);
```c
sprintf((char *)string, "AT+CIPSEND=%s", queryLengthString);
length = strlen((char *)string);
string[length] = 0x0D;
string[length + 1] = 0x0A;
string[length + 2] = 0;
HAL_UART_Transmit(&huart3, (uint8_t *)string, length + 2, 0xFFFF);
HAL_UART_Receive(&huart3, (uint8_t *)buffer, 23, 4000);

for (int i = 0; i < 2000; i++) buffer[i] = ' ';  
HAL_Delay(1);
HAL_UART_Transmit(&huart3, (uint8_t *)(query), queryLength, 0xFFFF);
HAL_UART_Receive(&huart3, (uint8_t *)buffer, 2000, 4000);
//HAL_UART_Transmit(&huart2, (uint8_t *)buffer, 2000, 0xFFFF);
string[0] = 0x0D;
string[1] = 0x0A;
HAL_UART_Transmit(&huart2, (uint8_t *)string,
```

```xml
<?xml version="1.0" encoding="UTF-8"?>
  <rss version="2.0">
    <channel>
      <title>
        동네예보 웹서비스 - 경기도 안산시성북구 사동 도박예보</title>
      <link>
        ma.go.kr/weather/main.jsp</link>
      <description>동네예보 웹서비스</description>
```
Exercise 2: IoT Exercise using WIFI Module

Wireless Broadband Router

STM32F407 Board

Serial WIFI Module

Ubidots.com
IoT Website

- Ubidots.com
- Select STEM from INDUSTRIES menu
Build our connected future, today.

Thousands of makers, students, and researchers use Ubidots STEM to test, learn, or teach IoT

- 3 forever free devices
- 200+ open source device libraries and tutorials
- Real-time dashboards with 30+ types of widgets (and the tools to code your own!)
- If-Then triggers with Email, SMS, Telegram, Voice call, Webhooks or Slack notifications.

Looking for our commercial product? Click here to start a free trial.
By signing up you agree to our Terms of Service and Privacy Policy.
DHT22 온/습도 센서 연결

VCC → 3V, GND → GND, DAT → PE0
DHT22 Protocol

DHT11 / DHT22 Protocol

- MCU Pulls High and waits for response
- MCU Pulls Low to send Start
- DHT responds & pulls LOW

- DHT Pulls HIGH to indicate 'get ready'
- Bit "0" 26-28us
- Bit "1"
- Each Data Bit starts with 50us LOW

- 18ms
- 80us
- 50us
- 70us

MCU Initiates Read (Start condition)
DHT Responds (Acknowledgment)
DHT sends 40 data bits (Data Transfer)
New STM32 Project

Project Name: Ubidots
STM32CubeMX: Pinout & Configuration

- USART2: Asynchronous
- USART3: Asynchronous
- TIM9: Internal Clock, Prescaler: 84
  Counter Period: 0xFFFF
Copy wifi_ubidots.c, dht22.c, main.c file to 
C:\Users\limdj\STM32CubeIDE\workspace_1.3.0\Ubidots\Core\Src

Copy wifi_ubidots.h, dht22.h file 
C:\Users\limdj\STM32CubeIDE\workspace_1.3.0\Ubidots\Core\Inc
WiFi 이름 변경

```c
void WifiSetup(void)
{
    uint8_t string[200];
    uint8_t buffer[200];
    int length;

    for (int i = 0; i < 200; i++) buffer[i] = ' ';  
    strcpy((uint8_t *)string, "AT+CWMODE=3");
    length = strlen((uint8_t *)string);
    string[length] = 0x0D;
    string[length + 1] = 0x0A;
    string[length + 2] = 0;
    HAL_UART_Transmit(&huart3, (uint8_t *)string, length + 2, 0xFFFF);
    HAL_UART_Transmit(&huart3, (uint8_t *)buffer, 100, 1000);
    HAL_UART_Transmit(&huart2, (uint8_t *)buffer, 100, 0xFFFF);
    string[0] = 0x0D;
    string[1] = 0x0A;
    HAL_UART_Transmit(&huart2, (uint8_t *)string, 2, 0xFFFF);
    strcpy((uint8_t *)string, (uint8_t *)"AT+CWMAP="iptime_limdj","xxxxx");
    length = strlen((uint8_t *)string);
    string[length] = 0x0D;
    string[length + 1] = 0x0A;
    string[length + 2] = 0;
    HAL_UART_Transmit(&huart3, (uint8_t *)string, length + 2, 0xFFFF);
    
```
DHT22_start();
check_response();
Rh_byte1 = read_data();
Rh_byte2 = read_data();
Temp_byte1 = read_data();
Temp_byte2 = read_data();
//sum = read_data();
//if (sum == (Rh_byte1+Rh_byte2+Temp_byte1+Temp_byte2))
{
    TEMP = (((Temp_byte1 << 8) | Temp_byte2);
    RH = (((Rh_byte1 << 8) | Rh_byte2);
}
temp = (float)TEMP / 10.0;
humid = (float)RH / 10.0;
FloatToString(temp_string, temp, 4, 1);
FloatToString(humid_string, humid, 4, 1);
sprintf(message, "%"temperature": %s, "humidity": %s", temp_string, humid_string);
HAL_UART_Transmit(&huart2, (uint8_t *)message, strlen(message), 0xFFFF);
sprintf(sendBuffer, "POST /api/v1.6/devices/%s/?token=%s HTTP/1.1\r\nHost: things.ubidots.com\nsendBufferLength = strlen((char *)sendBuffer);
for (int i = 0; i < 50; i++)buffer[i] = ' ';
my_itoa(sendBufferLength, sendBufferLengthString, 3);
sprintf((char *)string, "AT+CIPSEND=%s", sendBufferLengthString);
length = strlen((char *)string);
string[length] = 0x0D;
string[length + 1] = 0x0A;
string[length + 2] = 0;
HAL_UART_Transmit(&huart3, (uint8_t *)string, length + 2, 0xFFFF);
HAL_UART_Receive(&huart3, (uint8_t *)buffer, 24, 4000);

for (int i = 0; i < 2000; i++)buffer[i] = ' ';
HAL_Delay(1);
HAL_UART_Transmit(&huart3, (uint8_t *)sendBuffer, sendBufferLength, 0xFFFF);
HAL_UART_Receive(&huart3, (uint8_t *)buffer, 500, 4000);
HAL_UART_Transmit(&huart2, (uint8_t *)buffer, 500, 0xFFFF);
string[0] = 0x0D;
string[1] = 0x0A;
HAL_UART_Transmit(&huart2, (uint8_t *)string, 2, 0xFFFF);
ret = (int)strstr((char *)buffer, (char *)"ERROR");
/* wait for a while to slow down */
HAL_Delay(10000);
Copy your Ubidots Token and paste in your source.
Paste the token in wifi_ubidots.c

```c
#include "stm32f4xx_hal.h"
#include "string.h"
#include "math.h"
#include "dht22.h"
extern UART_HandleTypeDef huart2;
extern UART_HandleTypeDef huart3;
#define UBIDOTS_TOKEN "BBFF-FNbbvAjk3TN96Bv2xxxxxxxxxx"
#define UBIDOTS_DEVICE "STM32F407"
```
Hello Cortex-M
AT+CWMODE=3
OK

AT+CWJAP="iptime_limdj","xxxxx"
CLOSED
WIFI DISCONNECT
WIFI CONNECTED
WIFI GOT IP

AT+CIPSTART="TCP","things.ubidots.com",80
busy p...

AT+CIPSTART="TCP","things.ubidots.com",80
CONNECT

OK

{"temperature": 25.1, "humidity": 52.5}
Recvd. 204 bytes

SEND OK

+IPD,311:HTTP/1.1 200 OK
Server: nginx
Date: Thu, 02 May 2019 01:34:10 GMT
Content-Type: application/json
Transfer-Encoding: chunked
Connection: keep-alive
Vary: Accept-Encoding
Vary: Cookie
Allow: GET, POST, HEAD, OPTIONS

4b
{"temperature": [{"status_code": 201}], "humidity": [{"status_code": 201}]}
Select Devices and wait for the device to appear.
Click humidity to change the color
Click brush

52.90
humidity

Description
Change description

API Label
humidity

ID
5dafde50e694a5108ef14ee

Allowed range
From: Min to: Max

Unit
Add unit

Tags
Add new tag

Last Activity
a few seconds ago

Oct 23 2019 - Oct 23 2019

New data available

2019-10-23 14:16:35 +09:00
52.90
{}
> Click desired color and click ➡️
Select Dashboards from Data menu and click “Add new Dashboard”. Then click.

No Dashboards created yet

Create Dashboards to visualize your data in realtime

Add new Dashboard
Add new Widget and select Line chart
Select Variables and click check mark.
Smartphone App