ESP32 components library with BMP180

Component Library -> https://github.com/UncleRus/esp-idf-lib
Documentation -> https://esp-idf-lib.readthedocs.io/en/latest/
Examples -> https://github.com/UncleRus/esp-idf-lib/tree/master/examples

Example how to implement components library with BMP180 sensor:

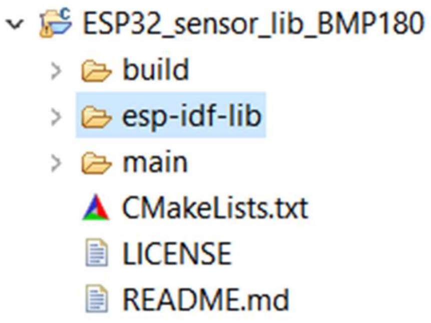
Create ESP-IDF project in Eclipse

- Create a new Espressif IDF Project: menu *File -> New -> Espressif IDF Project*.

- Enter a project name, click *Next*

- On the next window leave *the Create a project using one of the templates* unchecked and click on finish. New project with simple application code example will be created.

- Clone ESP IDF components/sensors library and to the project
  - Copy to the project folder.
  - Refresh project to view new folder.



ESP32_sensor_lib_BMP180
- build
- esp-idf-lib
- main
- CMakeLists.txt
- LICENSE
- README.md

- Adjust the top level CMakeLists.txt file to include the sensor library component. (*Check Examples folder*)

```
# The following lines of boilerplate have to be in your project's
# CMakeLists in this exact order for cmake to work correctly
cmake_minimum_required(VERSION 3.5)

set(EXTRA_COMPONENT_DIRS esp-idf-lib/components)
include($ENV{IDF_PATH}/tools/cmake/project.cmake)
project(app-template)
```

- Adjust the "*Kconfig.projbuild*" file or add one from example to the *'main'* folder.

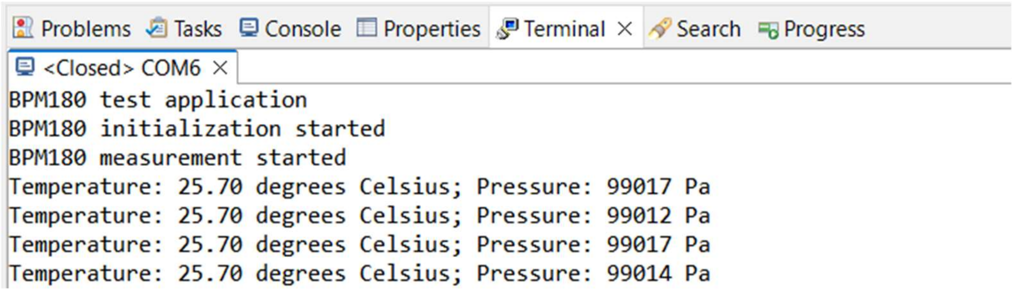
```
menu "Example configuration"
    config EXAMPLE_I2C_MASTER_SCL
        int "SCL GPIO Number"
        default 5 if IDF_TARGET_ESP8266
        default 6 if IDF_TARGET_ESP32C3
        default 22 if IDF_TARGET_ESP32 || IDF_TARGET_ESP32S2 || IDF_TARGET_ESP32S3
        help
            GPIO number for I2C Master clock line.

    config EXAMPLE_I2C_MASTER_SDA
        int "SDA GPIO Number"
        default 4 if IDF_TARGET_ESP8266
        default 5 if IDF_TARGET_ESP32C3
        default 21 if IDF_TARGET_ESP32 || IDF_TARGET_ESP32S2 || IDF_TARGET_ESP32S3
        help
            GPIO number for I2C Master data line.
endmenu
```

- Adjust application files (*main.c,* ...).

```
#include <stdio.h>
#include <stdbool.h>
#include <unistd.h>

#include <inttypes.h>
#include <freertos/FreeRTOS.h>
#include <freertos/task.h>
#include <esp_system.h>
#include <bmp180.h>
#include <string.h>


#ifndef APP_CPU_NUM
      #define APP_CPU_NUM PRO_CPU_NUM
#endif

void bmp180_test(void *pvParameters)
{
    bmp180_dev_t dev;
    memset(&dev, 0, sizeof(bmp180_dev_t)); // Zero descriptor

    printf("BPM180 initialization started\n");
    ESP_ERROR_CHECK(bmp180_init_desc(&dev, 0, CONFIG_EXAMPLE_I2C_MASTER_SDA, 
CONFIG_EXAMPLE_I2C_MASTER_SCL));
    ESP_ERROR_CHECK(bmp180_init(&dev));

    printf("BPM180 measurement started\n");
    while (1)
    {
        float temp;
        uint32_t pressure;

        esp_err_t res = bmp180_measure(&dev, &temp, &pressure, BMP180_MODE_STANDARD);
        if (res != ESP_OK)
            printf("Could not measure: %d\n", res);
        else
            /* float is used in printf(). you need non-default configuration in
             * sdkconfig for ESP8266, which is enabled by default for this
             * example. see sdkconfig.defaults.esp8266
             */
            printf("Temperature: %.2f degrees Celsius; Pressure: %" PRIu32 " Pa\n", 
temp, pressure);

        vTaskDelay(pdMS_TO_TICKS(2000));
    }
}


void app_main(void)
{
	printf("BPM180 test application\n");

	ESP_ERROR_CHECK(i2cdev_init());
	xTaskCreatePinnedToCore(bmp180_test, "bmp180_test", configMINIMAL_STACK_SIZE * 
15, NULL, 5, NULL, APP_CPU_NUM);
}
```

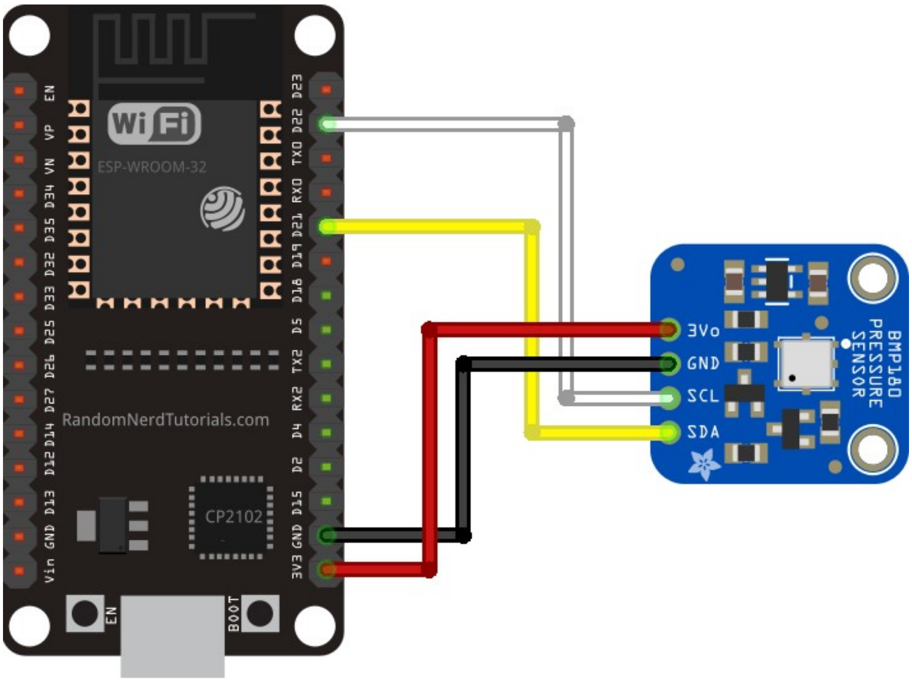
- Build the project and upload to the ESP32

- Open terminal window *Window -> Show View -> Terminal* . In the opened window, select "*Open a Terminal*" icon. Select *Project name*, *Serial Port*. Results in the terminal window should appear.



```
BPM180 test application
BPM180 initialization started
BPM180 measurement started
Temperature: 25.70 degrees Celsius; Pressure: 99017 Pa
Temperature: 25.70 degrees Celsius; Pressure: 99012 Pa
Temperature: 25.70 degrees Celsius; Pressure: 99017 Pa
Temperature: 25.70 degrees Celsius; Pressure: 99014 Pa
```

The BMP180 barometric sensor connection to the ESP32 shown in the following schematic diagram.


Wi Fi
ESP-WROOM-32
EN
VP
VN
D34
D35
D32
D33
D25
D26
D27
D14
D12
D13
GND
Vin
D23
D22
TXO
RXO
D21
D19
D18
D5
TX2
RX2
D4
D2
D15
GND
3V3
RandomNerdTutorials.com
CP2102
EN
BOOT
3Vo
GND
SCL
SDA
BMP180 PRESSURE SENSOR