Lab Cortex-M4

I2C Oled Display
Open STM32CubeMX and click New Project
Board Selector

- Select Board Selector Tab
- Select Discovery and STM32F4
- Double click STM32F407VGTx MCU

<table>
<thead>
<tr>
<th>Peripherals</th>
<th>Nb</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accelerometer</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Analog I/O</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Arduino Form Factor</td>
<td>32</td>
<td></td>
</tr>
<tr>
<td>Audio Line In</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Audio Line Out</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Button</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>CAN</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Camera</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Compass</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Custom Form Factor</td>
<td>66</td>
<td></td>
</tr>
<tr>
<td>Digital I/O</td>
<td>128</td>
<td></td>
</tr>
<tr>
<td>Eeprom</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Ethernet</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Type</th>
<th>Reference</th>
<th>MCU</th>
</tr>
</thead>
<tbody>
<tr>
<td>Discovery</td>
<td>STM32F401C-DISCO</td>
<td>STM32F401VCTx</td>
</tr>
<tr>
<td>Discovery</td>
<td>STM32F411E-DISCO</td>
<td>STM32F411VETx</td>
</tr>
<tr>
<td>Discovery</td>
<td>STM32F4DISCOVERY</td>
<td>STM32F407VGTx</td>
</tr>
<tr>
<td>Discovery</td>
<td>STM32F4DISCOVERY</td>
<td>STM32F407VGTx</td>
</tr>
<tr>
<td>Discovery</td>
<td>STM32F429I-DISCO</td>
<td>STM32F429ZITx</td>
</tr>
<tr>
<td>Discovery</td>
<td>STM32F429I-DISC1</td>
<td>STM32F429ZITx</td>
</tr>
<tr>
<td>Discovery</td>
<td>STM32F469I-DISCO</td>
<td>STM32F469NIHx</td>
</tr>
<tr>
<td>Discovery</td>
<td>STM32F412G-DISCO</td>
<td>STM32F412ZGTx</td>
</tr>
<tr>
<td>Discovery</td>
<td>STM32F413H-DISCO</td>
<td>STM32F413ZHTx</td>
</tr>
</tbody>
</table>
- Select Settings from Project menu
### Project Name: Oled

#### Project Settings

<table>
<thead>
<tr>
<th>Project Name</th>
<th>Oled</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Project Location</th>
<th>D:\work\STM32Cube</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Toolchain Folder Location</th>
<th>D:\work\STM32Cube\Oled</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Toolchain / IDE</th>
<th>EWARM</th>
</tr>
</thead>
</table>

#### Linker Settings

<table>
<thead>
<tr>
<th>Minimum Heap Size</th>
<th>0x200</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimum Stack Size</td>
<td>0x400</td>
</tr>
</tbody>
</table>

#### Mcu and Firmware Package

<table>
<thead>
<tr>
<th>Mcu Reference</th>
<th>STM32F407VGTx</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Firmware Package Name and Version</th>
<th>STM32Cube FW_F4 V1.18.0</th>
</tr>
</thead>
</table>

| Use latest available version | ✔ |

<table>
<thead>
<tr>
<th>Use Default Firmware Location</th>
<th>C:\Users\linda\STM32Cube\Repository\STM32Cube_FW_F4_V1.18.0</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Ok</th>
<th>Cancel</th>
</tr>
</thead>
</table>
- Select Code Generator Tab
- Select Add necessary library files as reference in the toolchain project configuration file
- Click OK
Pinout Selection

- CAN1: Master Mode
- I2C1: I2C
- I2C3: I2C
- USART2: Asynchronous
- USART3: Asynchronous
- Save from File menu
- Generate Code from Project menu
- Do not click Open Project
- Open folder or Close
Open Visual Studio 2015

- Start New Project
- Select VisualGDB and Embedded Project Wizard
- Check the location and change if necessary
- Change name to SerialWifi
- Click OK
- Import a project built with command-line-tools
- Generate a makefile to build the project
- **Uncheck** Copy imported files to project directory
- Click Next
- Type STM32F407VG and double click the device
- Or select the device and click Next
- Directory for imported source
  D:\work\stm32cube\Oled

  New Embedded Project

  Source Directory To Import
  Directory with imported sources:
  D:\work\stm32cube\Oled

  Importing Sources to Visual Studio Project

  - Import preserving directory structure
    VisualGDB will create virtual subfolders inside "source files" and "header files" in Project Explorer.

  - Show all source files together
    All source files from all subdirectories will be put together in "Source files" view in Project Explorer.

  - Do not import source files to Visual Studio project
    The created project will not link to any source files. Building and debugging the project will be possible, however, the source files will not be shown in Project Explorer and won't be parsed by IntelliSense. You will be able to add the files to the project manually using the "Add existing files" command.
Check debugging interface and click Finish
Select VisualGDB Project Properties
- Uncheck **STM32F4 Default init File**
- Click Apply
- Set Include directories
- Click OK
- Build Solution and check successful build
- Copy ssd1306.c and fonts.c files to D:\work\stm32cube\Oled\Src
- Copy ssd1306.h and fonts.h files to D:\work\stm32cube\Oled\Inc
- Add copied files to Src group (right click) in the Solution Explorer
Double click main.c in Solution Explorer
Add the following code in main.c

```c
/* USER CODE END 0 */

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 BEGIN Includes */
#include "ssd1306.h"
#include "fonts.h"
/* USER CODE END Includes */

/* USER CODE BEGIN 2 */
ssd1306_Init();
HAL_Delay(1000);
ssd1306_Fill(Black);
ssd1306_UpdateScreen();

HAL_Delay(1000);

ssd1306_SetCursor(0, 0);
ssd1306_WriteString("Hello World", Font_11x18, White);
ssd1306_SetCursor(0, 50);
ssd1306_WriteString("ARM Cortex-M3", Font_7x10, White);
ssd1306_UpdateScreen();
/* USER CODE END 2 */
/ * USER CODE BEGIN WHILE */
    int counter;
    unsigned char string[10];
    while (1)
    {
    / * USER CODE END WHILE */

/ * USER CODE BEGIN 3 */
    string[0] = counter / 100 + 0x30;
    string[1] = (counter % 100) / 10 + 0x30;
    string[2] = (counter % 100) % 10 + 0x30;
    string[3] = 0;
    ssd1306_SetCursor(40, 20);
    ssd1306_WriteString(string, Font_16x26, White);
    counter++;
    if (counter > 999) counter = 0;
    ssd1306_UpdateScreen();
    HAL_Delay(10);
    HAL_Delay(1);
    }
/ * USER CODE END 3 */
```c
void ssd1306_WriteCommand(uint8_t command)
{
    HAL_I2C_Mem_Write(&hi2c1, SSD1306_I2C_ADDR, 0x00, 1, &command, 1, 10);
}

// Het scherm initialiseren voor gebruik

uint8_t ssd1306_Init(void)
{
    // Even wachten zodat het scherm zeker opgestart is
    HAL_Delay(100);

    /* Init LCD */
    ssd1306_WriteCommand(0xAE); // display off
    ssd1306_WriteCommand(0x20); // Set Memory Addressing Mode
```
Logic Analyzer Capture

- **ACK**
void ssd1306_UpdateScreen(void)
{
    uint8_t i;

    for (i = 0; i < 8; i++) {
        ssd1306_WriteCommand(0xB0 + i);
        ssd1306_WriteCommand(0x00);
        ssd1306_WriteCommand(0x10);

        // We schrijven alles map per map weg
        HAL_I2C_Mem_Write(&hi2c1, SSD1306_I2C_ADDR, 0x40, 1, &SSD1306_Buffer[SSD1306_WIDTH * i], SSD1306_WIDTH, 100);
    }
}
Exercise

- WifiWeather.c의 프로그램을 수정하여 OLED display에 일기예보를 나타내는 프로그램을 작성한다.
- 한국 현재 시간(GMT보다 9시간 빠름), 기온, 습도, 일기예보(영문)를 나타낸다. 글자 크기, 화면 배치, 그래픽 등의 세부적인 사항은 창의력을 발휘하여 자유롭게 선택한다.
- 결과는 사진을 찍어서 보고서에 포함한다.
WiFi OLED Mini Weather Station
with ESP8266

Posted on September 9, 2017 by Erich Styger

I’m convinced that this ‘Internet of Things’ thing-thing is not real. Pure marketing and buzz words without any added value, right? The IoT hype is so bizar; it must be originated by aliens which have taken over the brains of all the Pointy-haired Bosses of the world? There is no useful application or use case out there!

But wait! There ‘is’ actually good use case, at least for the geeks of this world. We all love clocks as we want to know the time, and we all love the weather forecast so we can plan accordingly. At least I usually do :-).