

Tutorial : interfacing an HD44780 LCD Display with STM32L4

The HD44780 is a widespread common LCD controller allowing to drive various formats of LCD displays made of 1 line with 8 characters (1X8), 2X16, 2X20, 4x20 or others.

The goal is to show how a GPIO library originally developed for a STM8s MCU (AN4032) can be easily modified for a STM32 MCU.

The following equipment is used

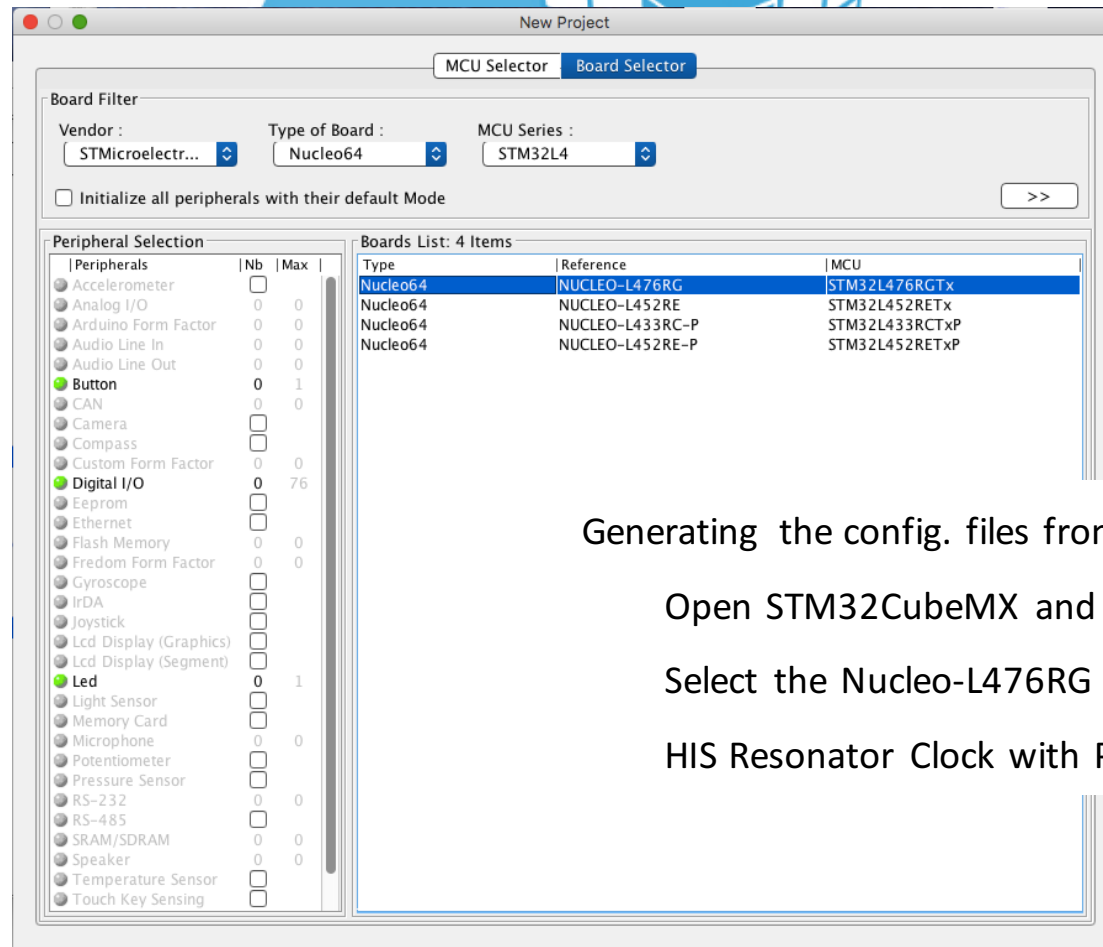
- NUCLEO-L476RG Board or equivalent and a 2X16 HD44780 LCD display
- Eclipse with the necessary packages for Nucleo boards installed
- OpenOCD or STLink USB Driver - STM32CubeMX

Jean-Christophe Toussaint Phelma Grenoble-INP

New Project

Load Project

Help



Generating the config. files from STM32CubeMX.

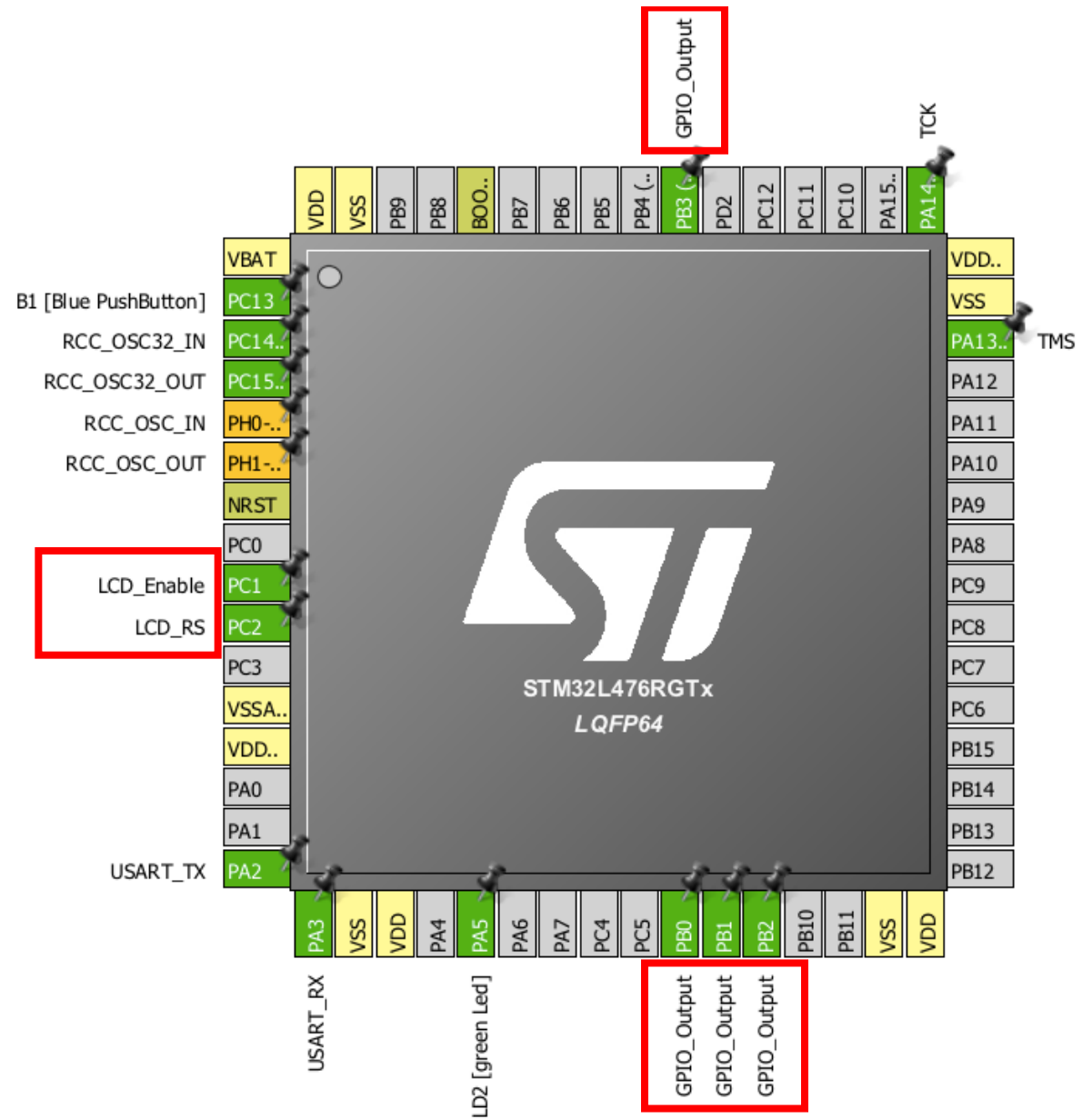
Open STM32CubeMX and open a new project.

Select the Nucleo-L476RG from the Boards tab

HIS Resonator Clock with PLLCLK is selected by default

Pin Name	Signal Name	User Label
PC13	GPIO_EXTI13	B1 [Blue PushButton]
PC14-OSC32_IN (PC14)	RCC_OSC32_IN	
PC15-OSC32_OUT (PC15)	RCC_OSC32_OUT	
PH0-OSC_IN (PH0)	RCC_OSC_IN	
PH1-OSC_OUT (PH1)	RCC_OSC_OUT	
PC1	GPIO_Output	LCD_Enable
PC2	GPIO_Output	LCD_RS
PA2	USART2_TX	USART_TX
PA3	USART2_RX	USART_RX
PA5	GPIO_Output	LD2 [green Led]
PB0	GPIO_Output	
PB1	GPIO_Output	
PB2	GPIO_Output	
PA13 (JTMS-SWDIO)	SYS_JTMS-SWDIO	TMS
PA14 (JTCK-SWCLK)	SYS_JTCK-SWCLK	TCK
PB3 (JTDO-TRACESWO)	GPIO_Output	

Apply OK Cancel



Project Settings

Project Code Generator Advanced Settings

Project Settings

Project Name
Nucleo_L4_HD44780_V2

Project Location
/Users/toussain/Documents/STM32dev

Toolchain Folder Location
/Users/toussain/Documents/STM32dev/Nucleo_L4_HD44780_V2/

Toolchain / IDE
SW4STM32

Code Generation

Linker Settings

Minimum Heap Size

Minimum Stack Size

The Code is successfully generated under /Users/toussain/Documents/STM32dev/nucleoL4_Timer1TriggeredByTimer2

Close Open Project Open Folder

Mcu and Firmware Package

Mcu Reference
STM32L476RGTx

Firmware Package Name and Version
STM32Cube FW_L4 V1.11.0

Ok Cancel

Go to Project Settings > Generate code

Enter a project name and select SW4STM32

Generate the code and open the project in Eclipse IDE

Example of Application :

code to be inserted in main.c

```
/* USER CODE BEGIN Includes */
#include <stdarg.h>
#include <stdio.h>
#include <HD44780.h>
/* USER CODE END Includes */
```

```
/* USER CODE BEGIN PV */
```

```
/* Private variables -----*/
```

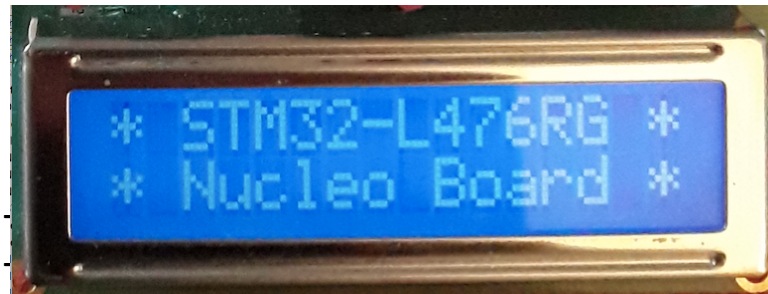
```
/* Exported types -----*/
```

```
/* Define 6 custom characters to display bar graph*/
```

```
char STCustom[48] =
```

```
{
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, // Blank
    0x10, 0x10, 0x10, 0x10, 0x10, 0x10, 0x10, 0x10, // 1column |
    0x18, 0x18, 0x18, 0x18, 0x18, 0x18, 0x18, 0x18, // 2columns ||
    0x1c, 0x1c, 0x1c, 0x1c, 0x1c, 0x1c, 0x1c, 0x1c, // 3columns |||
    0x1e, 0x1e, 0x1e, 0x1e, 0x1e, 0x1e, 0x1e, 0x1e, // 4columns ||||
    0x1f, 0x1f, 0x1f, 0x1f, 0x1f, 0x1f, 0x1f, 0x1f, // 5columns |||||
};
```

```
/* USER CODE END PV */
```



```

/* USER CODE BEGIN 2 */
  LCD_INIT();
  LCD_CLEAR_DISPLAY();

  /* Set @CGRAM address start */
  LCD_CMD(CGRAM_address_start);
  /* Loading 6 characters @CGRAM address from STCustom tab */
  LCD_LOAD_CGRAM(STCustom, 6);

  /* Set cursor to the chosen position*/
  LCD_LOCATE(1, 1);
  /* Print string on LCD (must be ended with \n)*/
  LCD_printstring("* STM32-L476RG \1\2\3\4\5 *\n");

  /* Print string on LCD (must be ended with \n)*/
  LCD_LOCATE(2, 1);
  LCD_printstring("* Nucleo Board \1\2\3\4\5 *\n");
/* USER CODE END 2 */

```

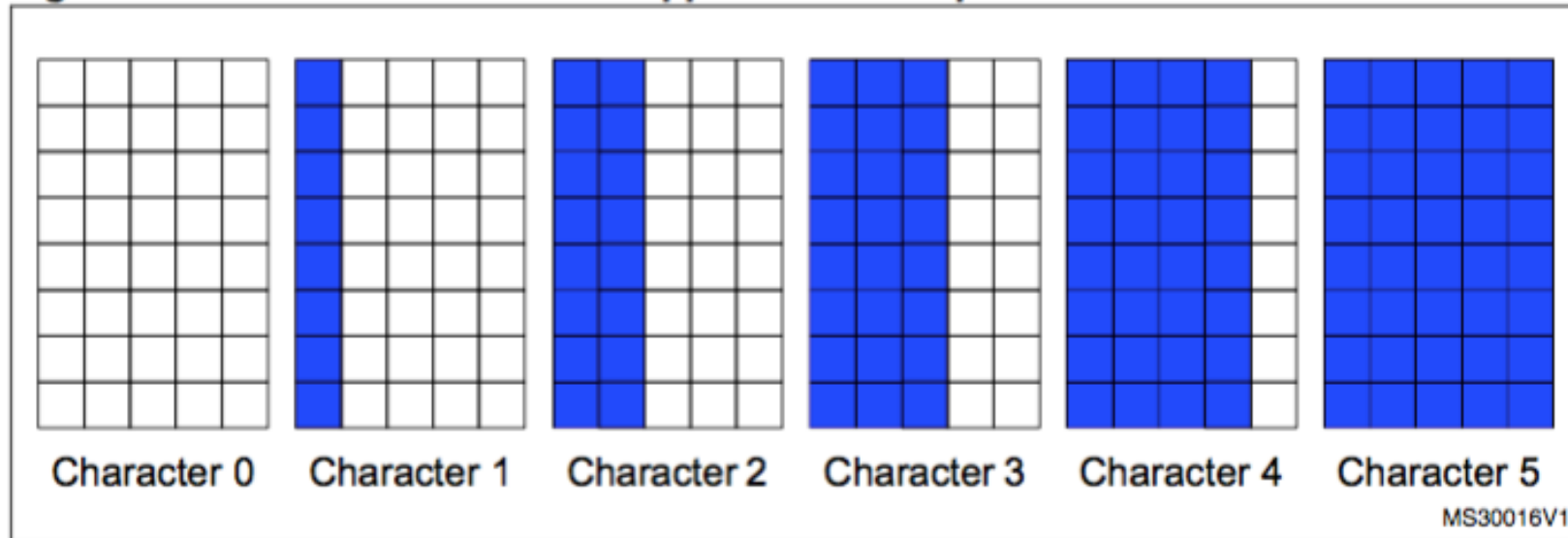
LCD_printing allows to print common characters and user defined ones.

```

/* USER CODE BEGIN 3 */
    LCD_RSHIFT();
    HAL_Delay(1000);
}
/* USER CODE END 3 */

```

Figure 5. Custom characters of application example



These characters are then written into the HD44780 LCD display module at the start address of CGRAM @0x40. These characters are defined from number 0 to 5 and can be called directly by printing their hexadecimal code.

For instance, the C instruction `LCD_printstring("\3\n")` displays Character 3 at the current cursor position. Displaying the bar graph prints a string containing the characters' numbers (progressively changing from Character 0 to Character 5) at the same LCD display location.

Part of the AN4032 application note

```

/**
***** HD4470 library for STM32
* @file      HD44780.c
* @author    MCD Application Team
* @version   V1.0.0
* @date      16_January-2012
* @brief     HD44780.c
* originally developed for STM8s
*
* adapted by JC Toussaint Phelma Grenoble-INP for STM32
* @date      18_February-2018
*****
* @attention
*
* THE PRESENT FIRMWARE WHICH IS FOR GUIDANCE ONLY AIMS AT PROVIDING CUSTOMERS
* WITH CODING INFORMATION REGARDING THEIR PRODUCTS IN ORDER FOR THEM TO SAVE
* TIME. AS A RESULT, STMICROELECTRONICS SHALL NOT BE HELD LIABLE FOR ANY
* DIRECT, INDIRECT OR CONSEQUENTIAL DAMAGES WITH RESPECT TO ANY CLAIMS ARISING
* FROM THE CONTENT OF SUCH FIRMWARE AND/OR THE USE MADE BY CUSTOMERS OF THE
* CODING INFORMATION CONTAINED HEREIN IN CONNECTION WITH THEIR PRODUCTS.
*
* FOR MORE INFORMATION PLEASE READ CAREFULLY THE LICENSE AGREEMENT FILE
* LOCATED IN THE ROOT DIRECTORY OF THIS FIRMWARE PACKAGE.
*
* <h2><center>&copy; COPYRIGHT 2011 STMicroelectronics</center></h2>
*****
*/

```


A small part of the library adapted for STM32

```
/* Includes -----*/
#include "main.h"
#include "stm32l4xx_hal.h"
#include <stdarg.h>
#include <stdio.h>
#include <HD44780.h>
```

```
uint16_t GPIO_ReadOutputData (GPIO_TypeDef *GPIOx){
    return GPIOx->ODR;
}
```

GPIO_ReadOutputData allows us to read the GPIOx port value and

```
void GPIO_Write (GPIO_TypeDef *GPIOx, uint16_t PortVal){
    GPIOx->ODR=PortVal;
}
```

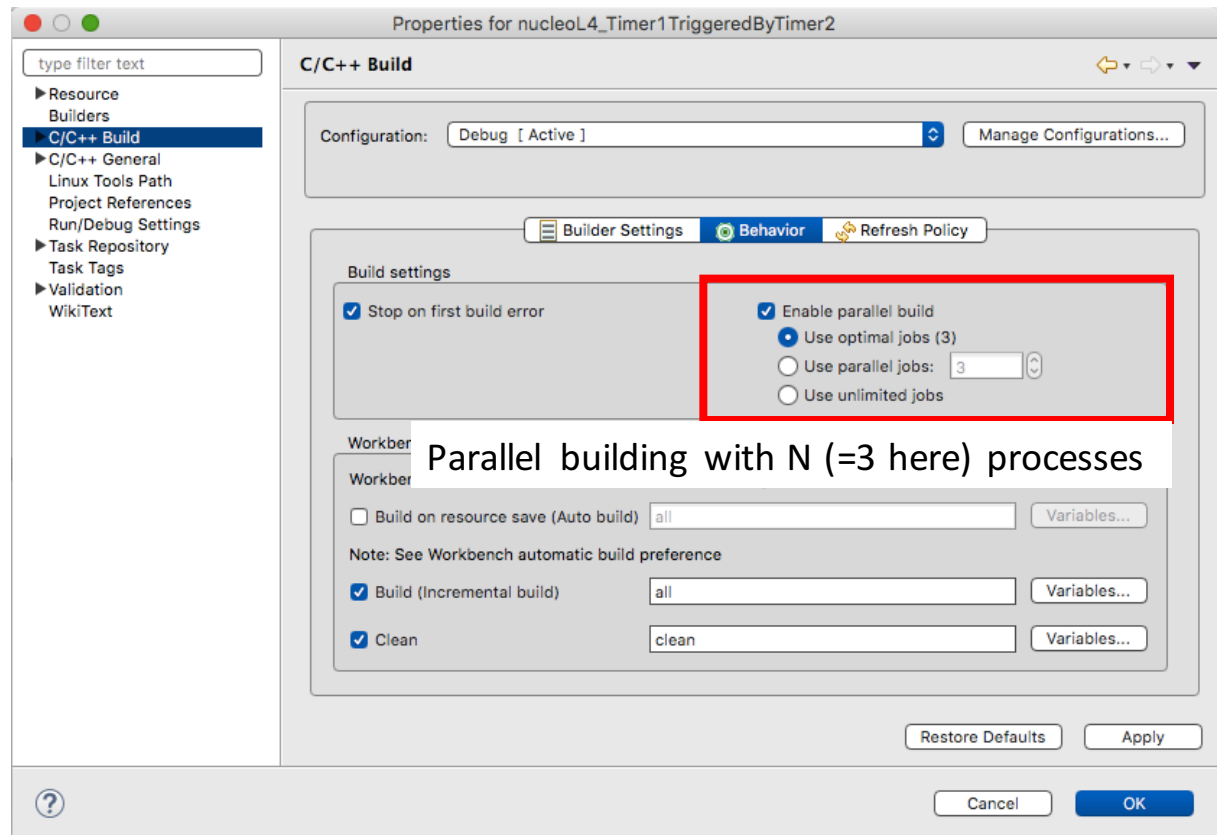
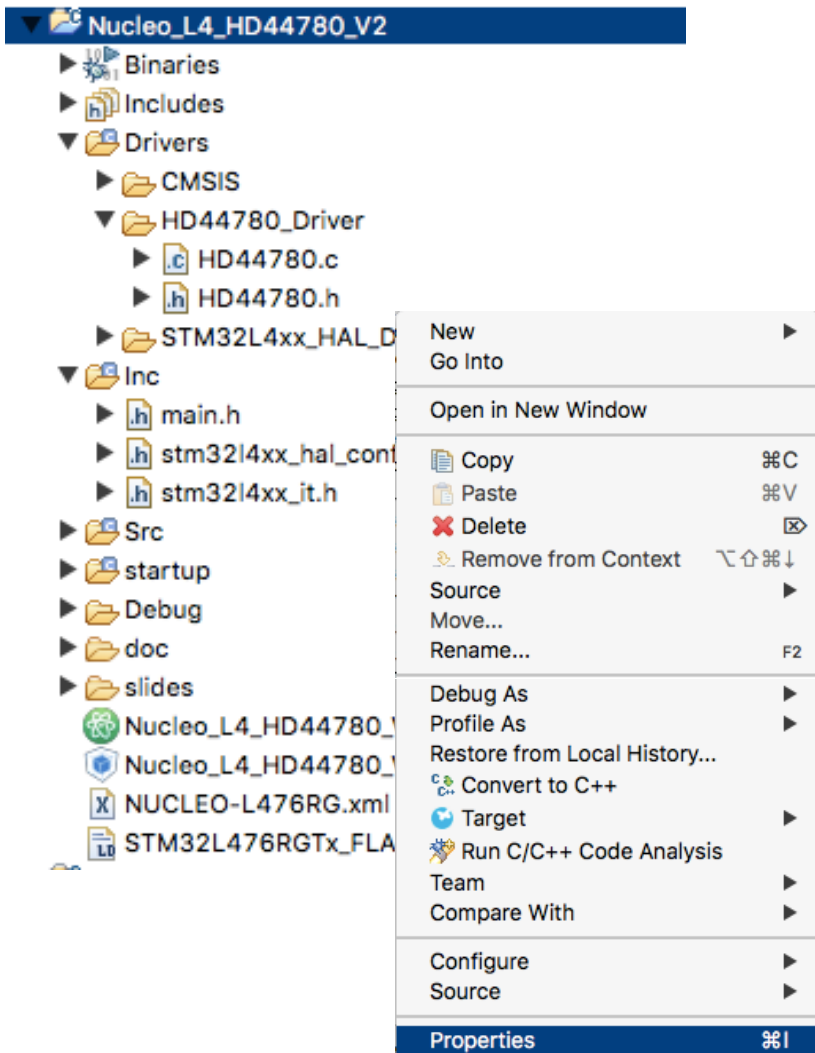
GPIO_Write to assign it a value

```
void LCD_CMD(unsigned char cmd_data)
{
    HAL_GPIO_WritePin(LCDControlPort, LCD_RS, GPIO_PIN_RESET);
```

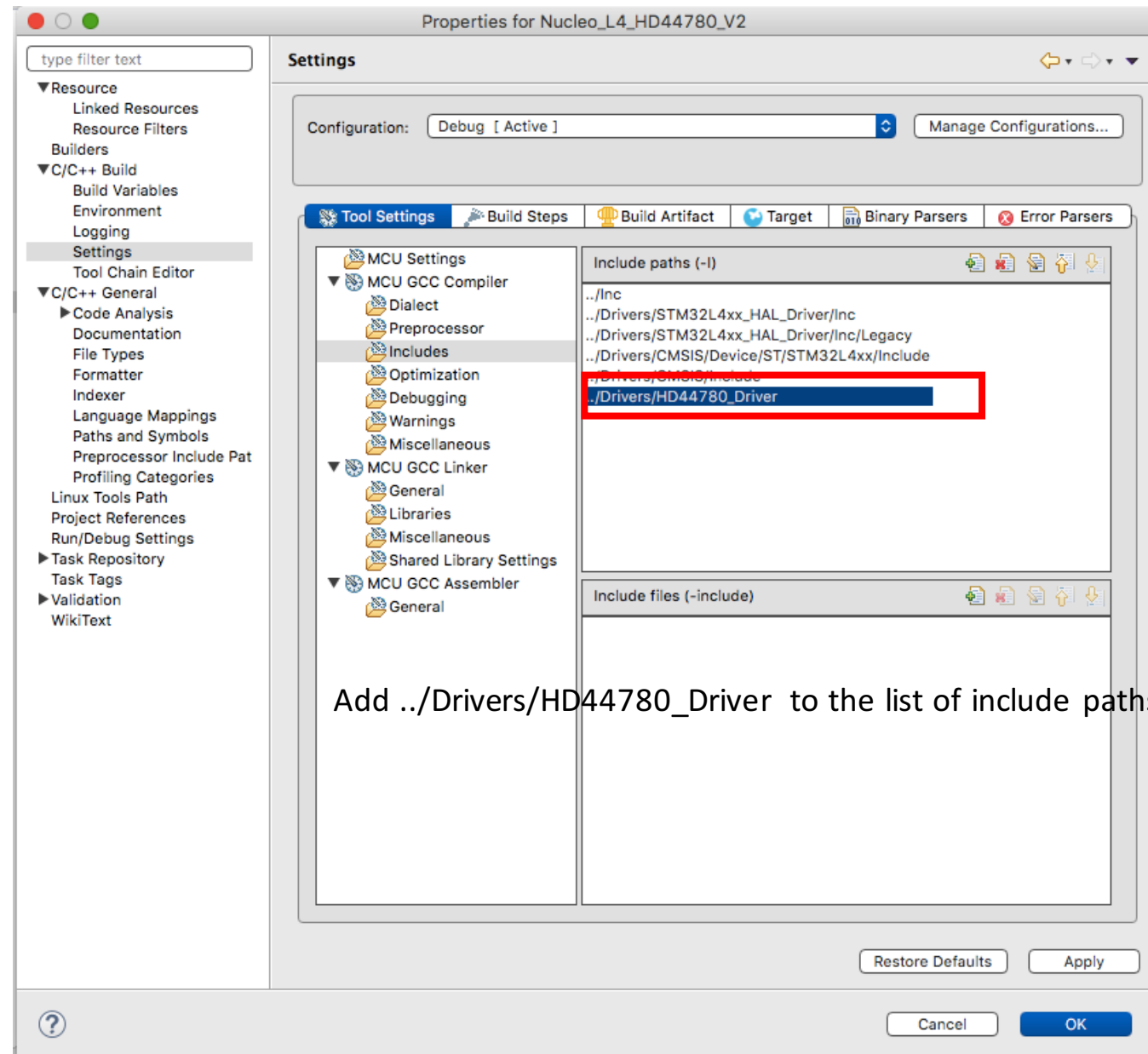
After that, the translation of the LCD_CMD function for STM32 is straightforward

```
/*
 * When the I/O port is programmed as output
 * a read access to the output data register gets the last written value.
 */
GPIO_Write(LCDPort, (GPIO_ReadOutputData(LCDPort) & 0xF0) | ((cmd_data >> 4) & 0x0F));
LCD_ENABLE();
GPIO_Write(LCDPort, (GPIO_ReadOutputData(LCDPort) & 0xF0) | (cmd_data & 0x0F));
LCD_ENABLE();
HAL_Delay(2);
}
```

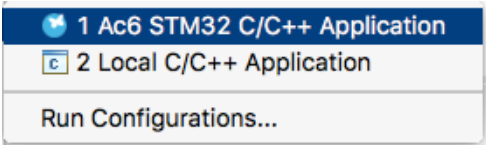
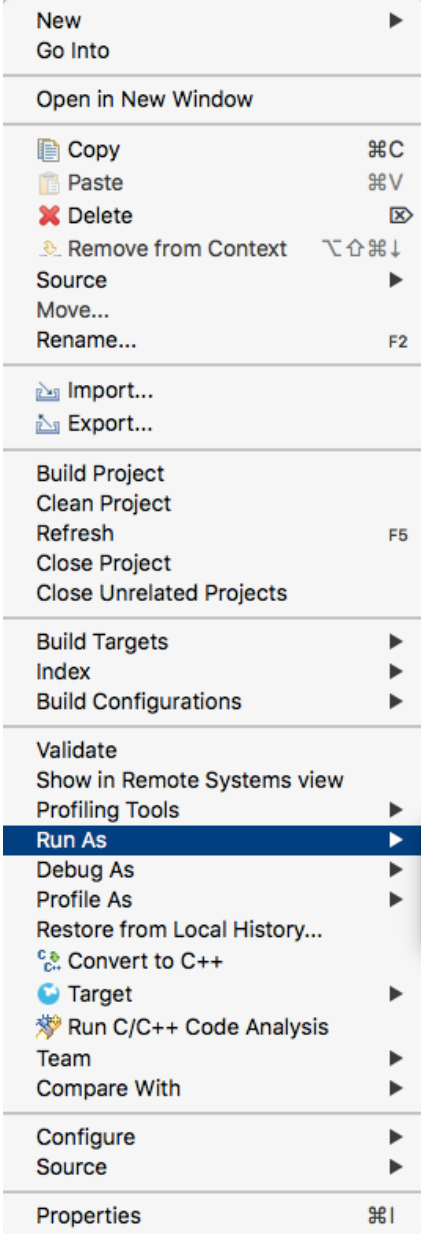
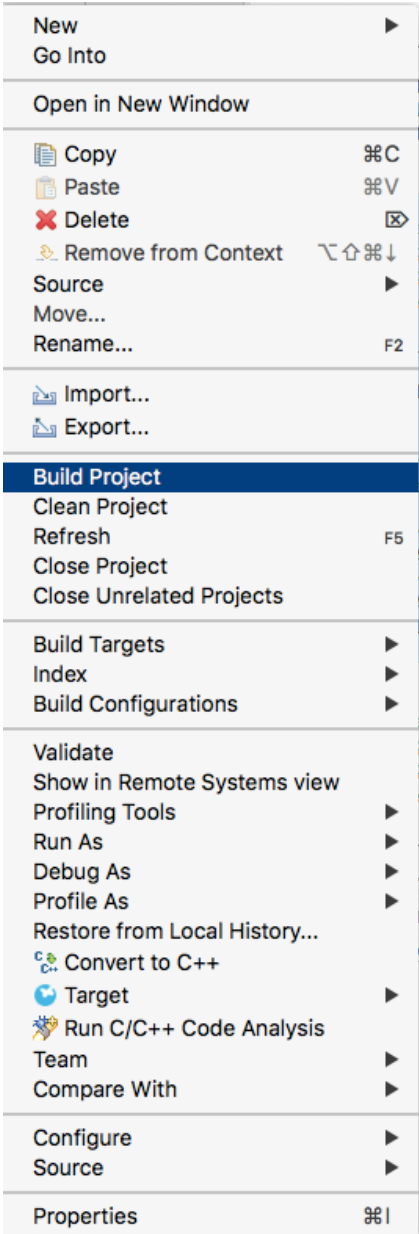
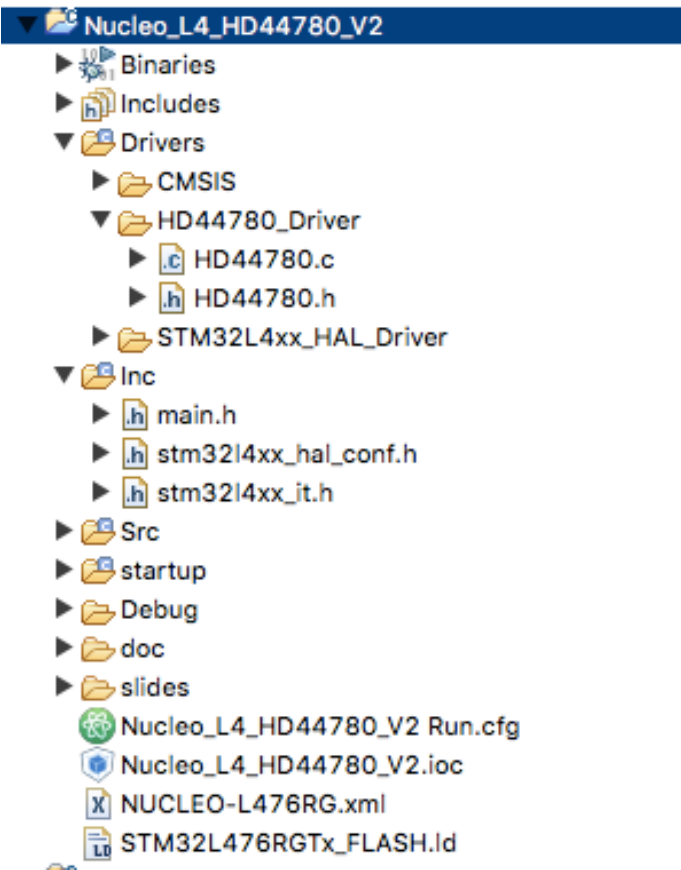
Eclipse Configuration



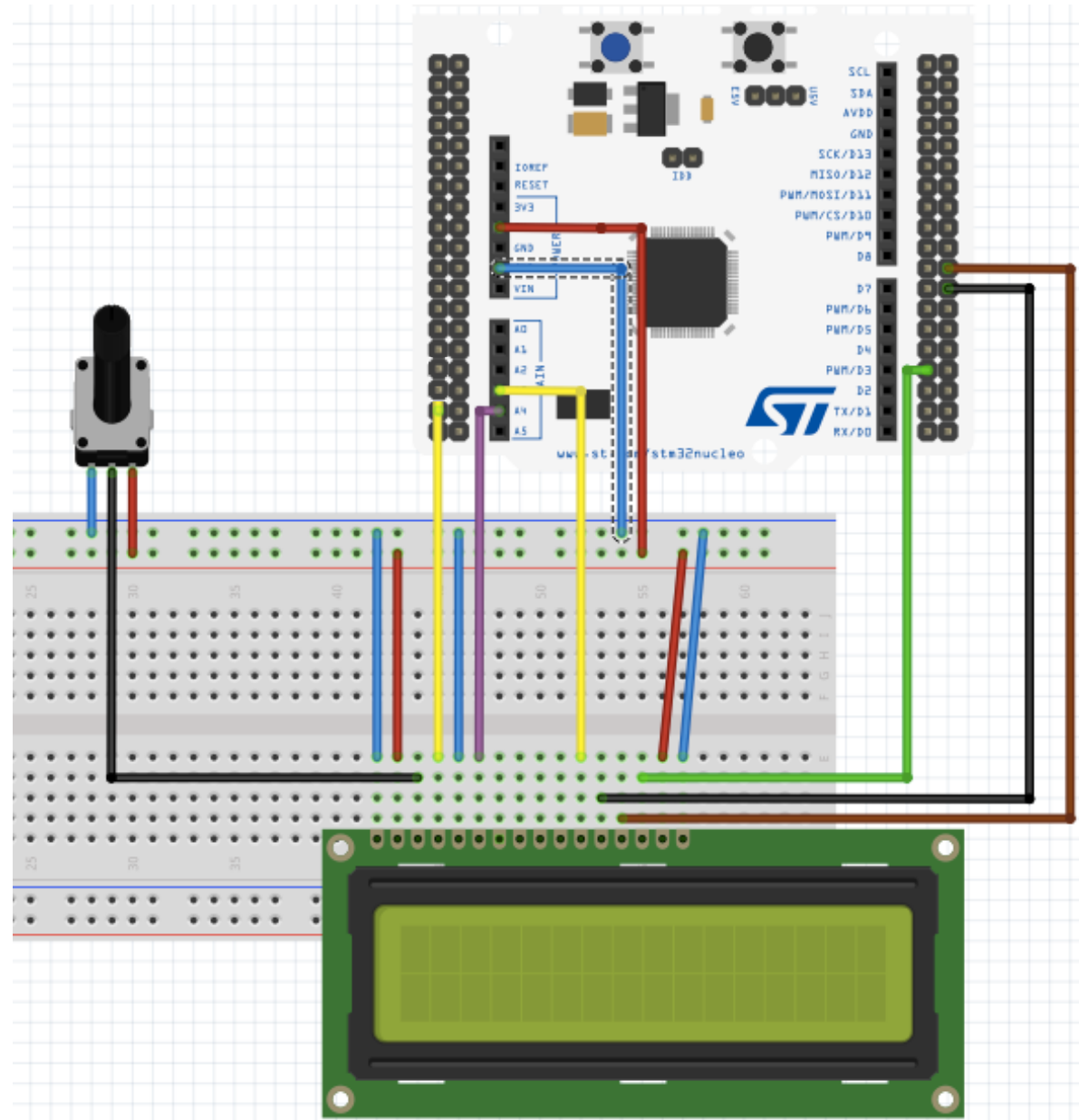
Include Paths



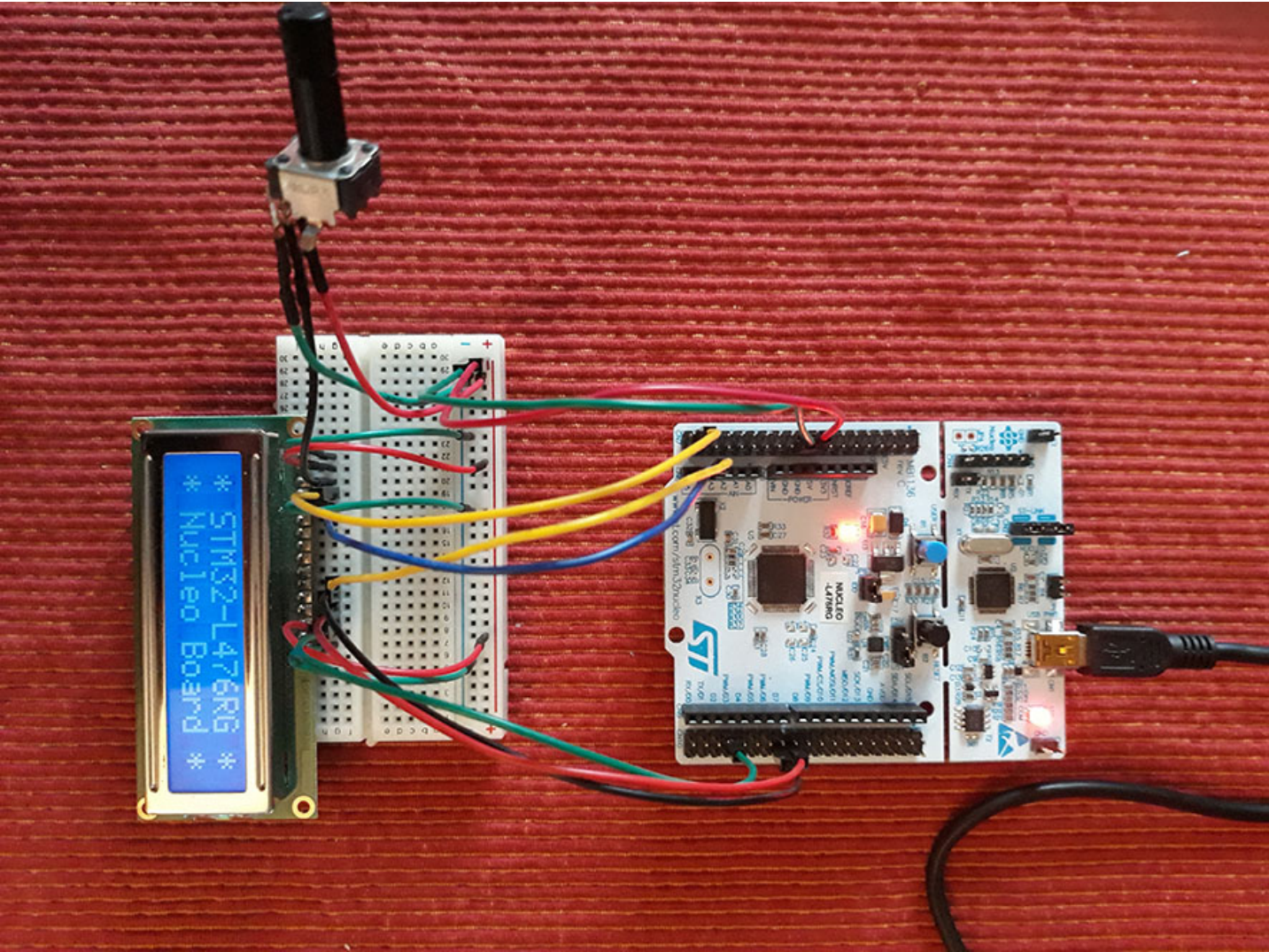
Compile and then
Run as a STM32 runtime



Wiring



In Practice



References

1. Carmine Noveillo, “Mastering STM32”, <https://leanpub.com/mastering-stm32>
2. Marc Laury, “A la découverte des cartes Nucleo”, Eyrolles, EAN13 : 9782212673692
3. Reference Manual for STM32L4, DM00083560.pdf
4. Application Note **AN4032**, “Interfacing an HD44780 2-line LCD display with the STM8SVLDISCOVERY”