



Bare Metal C Programming for STM32 Devices

Day 1:

Bare Metal NUCLEO-F207ZG :

The STM32CubeIDE Visual Studio Code Extension Package

Sponsored by

DigiKey

Webinar Logistics

- Turn on your system sound to hear the streaming presentation.
- If you have technical problems, click “Help” or submit a question asking for assistance.
- Participate in ‘Attendee Chat’ by maximizing the chat widget in your dock.



Fred Eady

Visit 'Lecturer Profile' in your console for more details.

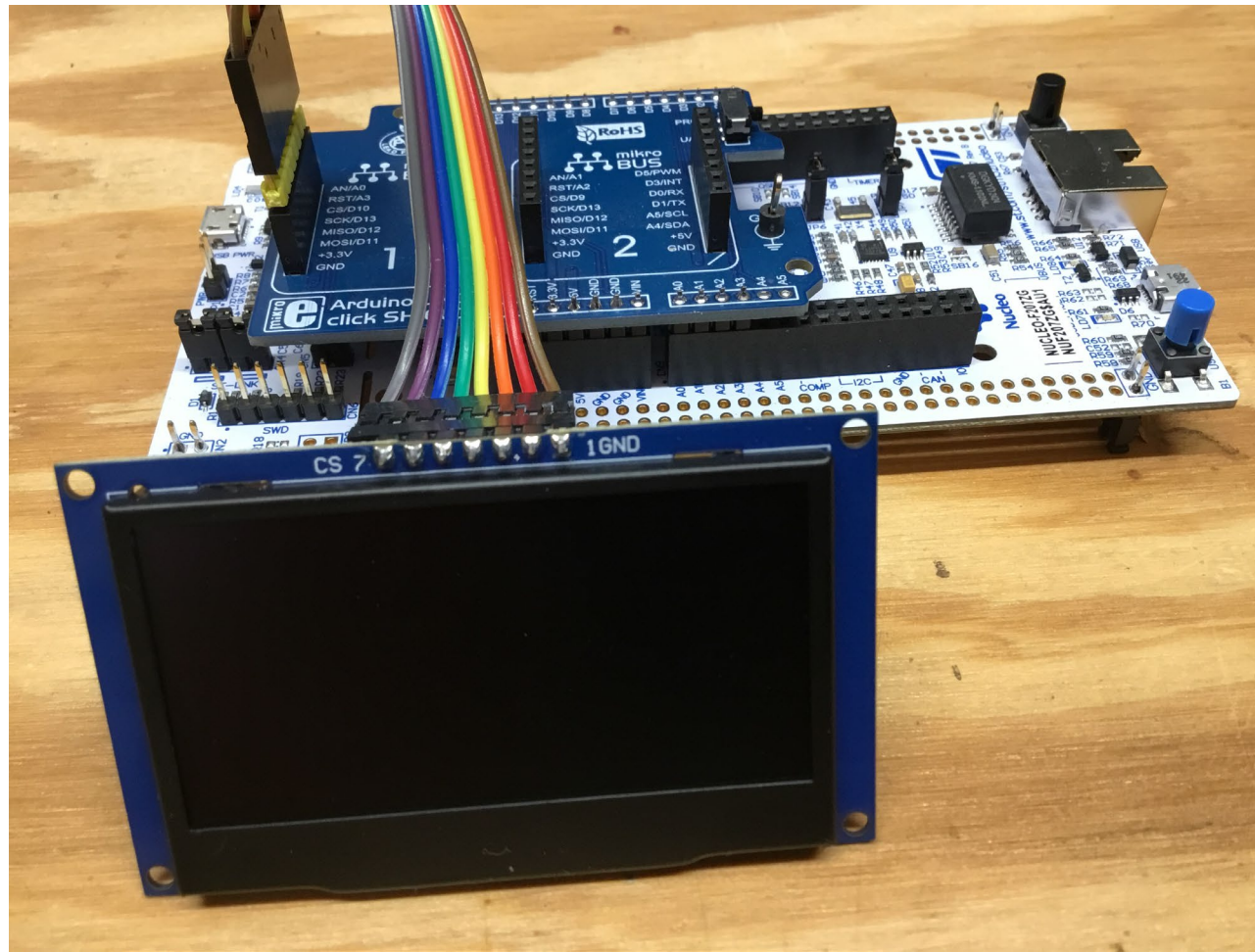
MAX



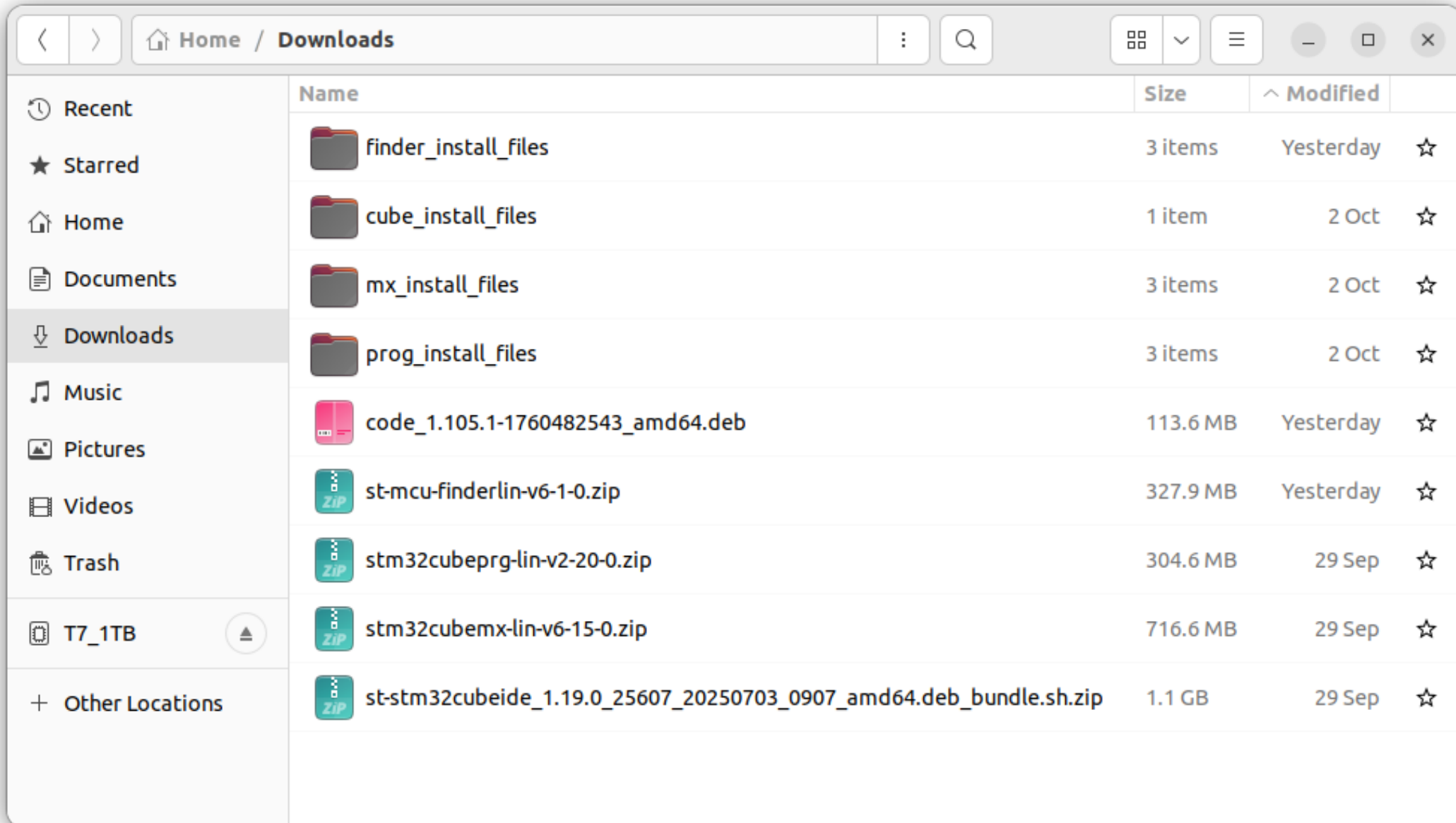
I am the ULTIMATE debugger!!

AGENDA

- **Install the Core STM32 Development Tools**
- **Create a Bare Metal SSD1309 Application**



Download and Install the Core STM32 Development Tools





Launch STM32CUFinder – Select NUCLEO-F207ZG

The screenshot displays the STM32CUFinder web application interface. On the left, the 'Board Filters' sidebar is visible, showing the 'Commercial Part Number' set to 'NUCLEO-F207ZG'. The main content area features a large image of the NUCLEO-F207ZG board with the STM32F207ZGT6 chip. The top navigation bar includes tabs for 'MCU/MPU Selector', 'Board Selector', 'Example Selector', and 'Cross Selector'. The top right of the interface has social media icons and a 'Buy' button. A 'Welcome' dialog box is open on the right side, listing 'STM32CUBE KEY ACTIONS' with 'Launch STM32CUFinder' highlighted by a red arrow. Below the board image, a 'Boards List' table is partially visible with columns for 'Overview', 'Commercial Part No', 'Type', 'Marketing Status', 'Unit Price (US\$)', and 'Mounted Device'.


Run STM32CubeMX – Specify Project Name and Toolchain

Pinout & Configuration | Clock Configuration | **Project Manager**

Project

Code Generator


Project Settings

Project Name  nucleo_f207zg_ssd1309

Project Location

Application Structure Do not generate the main()

Toolchain Folder Location

Toolchain / IDE  CMake Generate Under Root

Default Compiler/Linker



Run STM32CubeMX – Configure the GPIO

Pinout & Configuration

GPIO Mode and Configuration

Mode

Configuration

Group By Peripherals

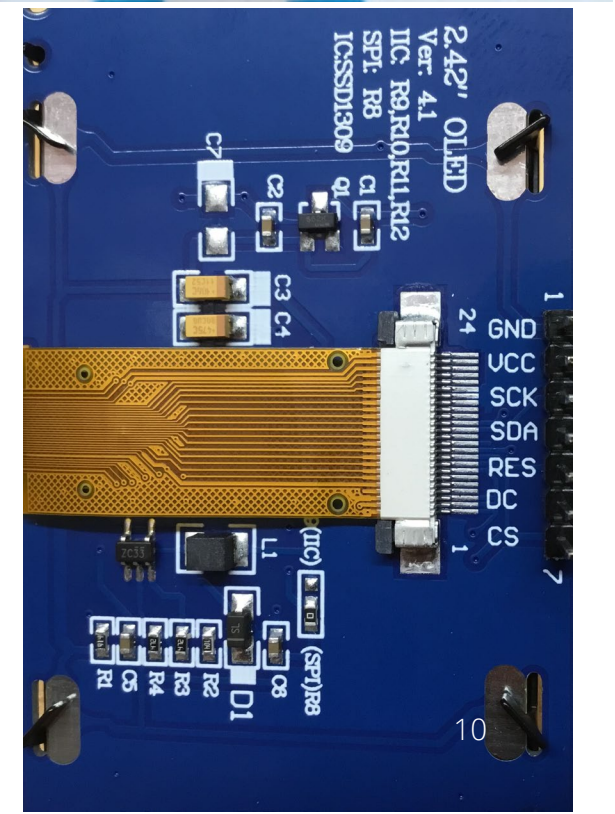
SYS USART USB NVIC
 GPIO Single Mapped Signals I2C RCC SPI

Search Signals

Search (Ctrl+F)

Show only Modified Pins

Pin ...	Signal ...	GPIO o...	GPIO m...	GPIO P...	Maximu..	User La...	Modified
PA3	n/a	Low	Output ...	No pull-...	Low	DC	<input checked="" type="checkbox"/>
PB0	n/a	Low	Output ...	No pull-...	Low	LED1G	<input checked="" type="checkbox"/>
PB7	n/a	Low	Output ...	No pull-...	Low	LED2B	<input checked="" type="checkbox"/>
PB14	n/a	Low	Output ...	No pull-...	Low	LED3R	<input checked="" type="checkbox"/>
PC13	n/a	n/a	Externa...	No pull-...	n/a	BTN1	<input checked="" type="checkbox"/>
PD14	n/a	Low	Output ...	No pull-...	High	SPI1_CS	<input checked="" type="checkbox"/>
PF3	n/a	High	Output ...	No pull-...	Low	RST	<input checked="" type="checkbox"/>
PG6	n/a	Low	Output ...	No pull-...	Low	USB_P...	<input checked="" type="checkbox"/>
PG7	n/a	n/a	Input m...	No pull-...	n/a	USB_O...	<input checked="" type="checkbox"/>





Run STM32CubeMX – Configure SPI1

Pinout & Configuration | Clock Configuration | Software Packs

Search: []

Categories: A->Z

System Core

- DMA
- GPIO
- IWDG
- NVIC
- RCC
- SYS
- WWDG

Analog >

Timers >

Connectivity >

Multimedia >

Security >

GPIO Mode and Configuration

Mode

Configuration

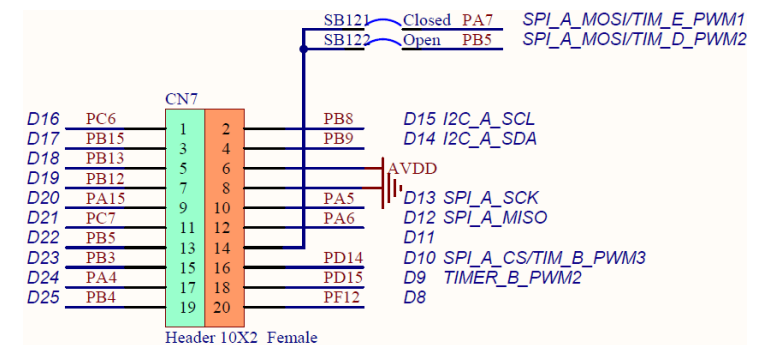
Group By Peripherals

SYS USART USB NVIC

GPIO Single Mapped Signals I2C RCC SPI

Search Signals: [Search (Ctrl+F)] Show only Modified Pins

Pin ...	Signal on Pin	GPIO o...	GPIO m...	GPIO P...	Maxim...	User L...	Modified
PA5	SPI1_SCK	n/a	Alterna...	No pull...	High		<input type="checkbox"/>
PA6	SPI1_MISO	n/a	Alterna...	No pull...	High		<input type="checkbox"/>
PA7	SPI1_MOSI	n/a	Alterna...	No pull...	High		<input type="checkbox"/>



Run STM32CubeMX – Generate the Skeleton Code

GENERATE CODE

MX

The screenshot shows the Visual Studio Code interface with the STM32CubeMX skeleton code in the main.c file. The code is as follows:

```

88 int main(void)
89 {
90     /* USER CODE BEGIN 1 */
91     /* USER CODE END 1 */
92     /* MCU Configuration-----*/
93     /* Reset of all peripherals, Initializes the Flash interface and the Systick. */
94     HAL_Init();
95     /* USER CODE BEGIN Init */
96     /* USER CODE END Init */
97     /* Configure the system clock */
98     SystemClock_Config();
99     /* USER CODE BEGIN SysInit */
100    /* USER CODE END SysInit */
101    /* Initialize all configured peripherals */
102    MX_GPIO_Init();
103    MX_USART3_UART_Init();
104    MX_USB_OTG_FS_PCD_Init();
105    MX_SPI1_Init();
106    /* USER CODE BEGIN 2 */
107    //*****
108    // INITIALIZE THE SSD1309 DISPLAY AND PRINT TO THE SSD1309
109    //*****
110    initSSD1309();
111    setTextXY(row: 0, col: 0);
112    sprintf(spiTxBuf, "SSD1309 PROJECT");
113    displayString(string: spiTxBuf);
114}

```

The terminal output shows the build process:

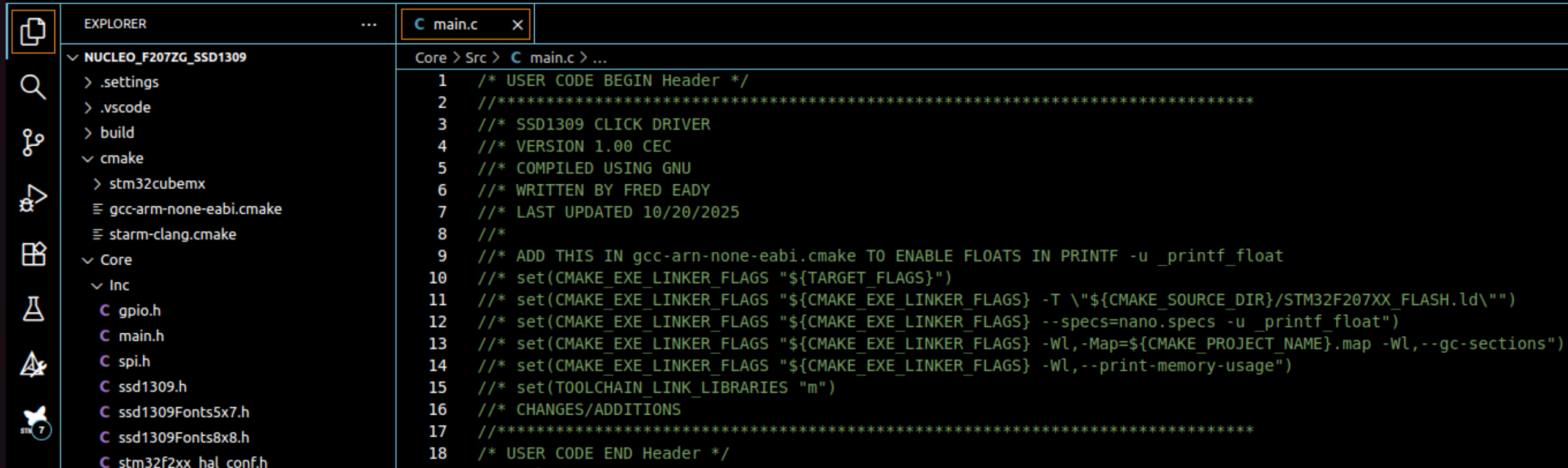
```

[main] Building folder: /home/fred/nucleo_f207zg_ssd1309/build/Debug
[build] Starting build
[driver] NOTE: You are building with preset Debug, but there are some overrides being applied from your VS Code settings.
[proc] Executing command: cube-cmake --build /home/fred/nucleo_f207zg_ssd1309/build/Debug --
[build] ninja: no work to do.
[driver] Build completed: 00:00:00.014
[build] Build finished with exit code 0

```

A notification at the bottom right asks: "Do you want to install the recommended 'C/C++ Extension Pack' extension from Microsoft for the C language?" with buttons for "Install" and "Show Recommendations".

Enable Printf Floats and Rebuild



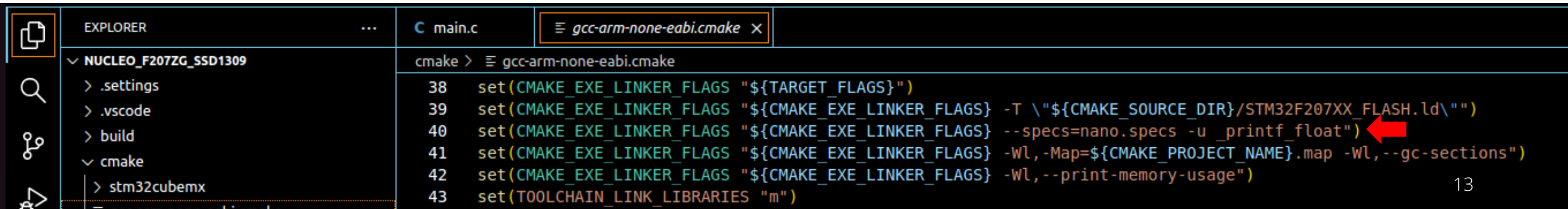
EXPLORER ... C main.c

Core > Src > C main.c > ...

```

1  /* USER CODE BEGIN Header */
2  /**
3  /** SSD1309 CLICK DRIVER
4  /** VERSION 1.00 CEC
5  /** COMPILED USING GNU
6  /** WRITTEN BY FRED EADY
7  /** LAST UPDATED 10/20/2025
8  /**
9  /** ADD THIS IN gcc-arm-none-eabi.cmake TO ENABLE FLOATS IN PRINTF -u _printf_float
10 /** set(CMAKE_EXE_LINKER_FLAGS "${TARGET_FLAGS}")
11 /** set(CMAKE_EXE_LINKER_FLAGS "${CMAKE_EXE_LINKER_FLAGS} -T \"${CMAKE_SOURCE_DIR}/STM32F207XX_FLASH.ld\")
12 /** set(CMAKE_EXE_LINKER_FLAGS "${CMAKE_EXE_LINKER_FLAGS} --specs=nano.specs -u _printf_float")
13 /** set(CMAKE_EXE_LINKER_FLAGS "${CMAKE_EXE_LINKER_FLAGS} -Wl,-Map=${CMAKE_PROJECT_NAME}.map -Wl,--gc-sections")
14 /** set(CMAKE_EXE_LINKER_FLAGS "${CMAKE_EXE_LINKER_FLAGS} -Wl,--print-memory-usage")
15 /** set(TOOLCHAIN_LINK_LIBRARIES "m")
16 /** CHANGES/ADDITIONS
17 /**
18 /* USER CODE END Header */

```



EXPLORER ... C main.c gcc-arm-none-eabi.cmake

cmake > gcc-arm-none-eabi.cmake

```

38 set(CMAKE_EXE_LINKER_FLAGS "${TARGET_FLAGS}")
39 set(CMAKE_EXE_LINKER_FLAGS "${CMAKE_EXE_LINKER_FLAGS} -T \"${CMAKE_SOURCE_DIR}/STM32F207XX_FLASH.ld\")
40 set(CMAKE_EXE_LINKER_FLAGS "${CMAKE_EXE_LINKER_FLAGS} --specs=nano.specs -u _printf_float")
41 set(CMAKE_EXE_LINKER_FLAGS "${CMAKE_EXE_LINKER_FLAGS} -Wl,-Map=${CMAKE_PROJECT_NAME}.map -Wl,--gc-sections")
42 set(CMAKE_EXE_LINKER_FLAGS "${CMAKE_EXE_LINKER_FLAGS} -Wl,--print-memory-usage")
43 set(TOOLCHAIN_LINK_LIBRARIES "m")

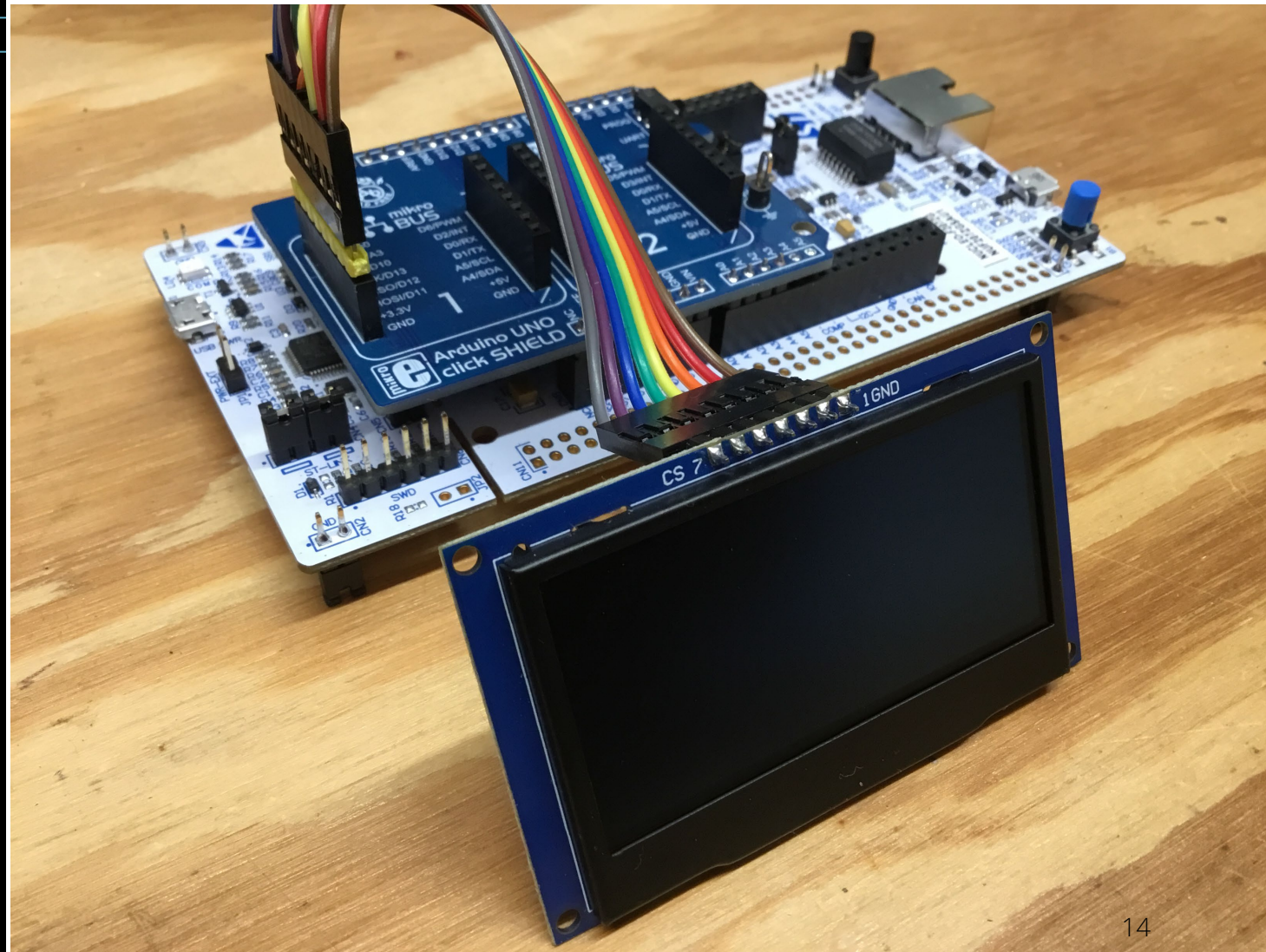
```

The Application Code – ssd1309.h

C ssd1309.h x

Core > Inc > C ssd1309.h > ...

```
25 //*****
26 /* FUNCTION PROTOTYPES
27 /* PREVENTS HAVING TO MANAGE THE ORDER OF ssd1309.c FUNCTONS
28 //*****
29 void displayString(char *string);
30 void sendCmd(char cmd);
31 void sendData(char bite);
32 void setFont(char* font);
33 void displayChar(char ch);
34 void setTextXY(uint8_t row, uint8_t col);
35 void clearDisplay(void);
36 void resetSSD1309(void);
37 void initSSD1309(void);
38
39 #define ON          1
40 #define OFF         0
41
42 #define SSD1309_Max_X    127    //128 Pixels
43 #define SSD1309_Max_Y    63     //64 Pixels
44
45 #define PAGE_MODE       01
46 #define HORIZONTAL_MODE 02
47
48 #define SSD1309_Command_Mode    0x80
49 #define SSD1309_Data_Mode       0x40
50 #define SSD1309_Display_Off_Cmd 0xAE
51 #define SSD1309_Display_On_Cmd  0xAF
52 #define SSD1309_Normal_Display_Cmd 0xA6
53 #define SSD1309_Inverse_Display_Cmd 0xA7
54 #define SSD1309_Activate_Scroll_Cmd 0x2F
55 #define SSD1309_Dectivate_Scroll_Cmd 0x2E
56 #define SSD1309_Set_Brightness_Cmd 0x81
```



The Application Code – `ssd1309.c`

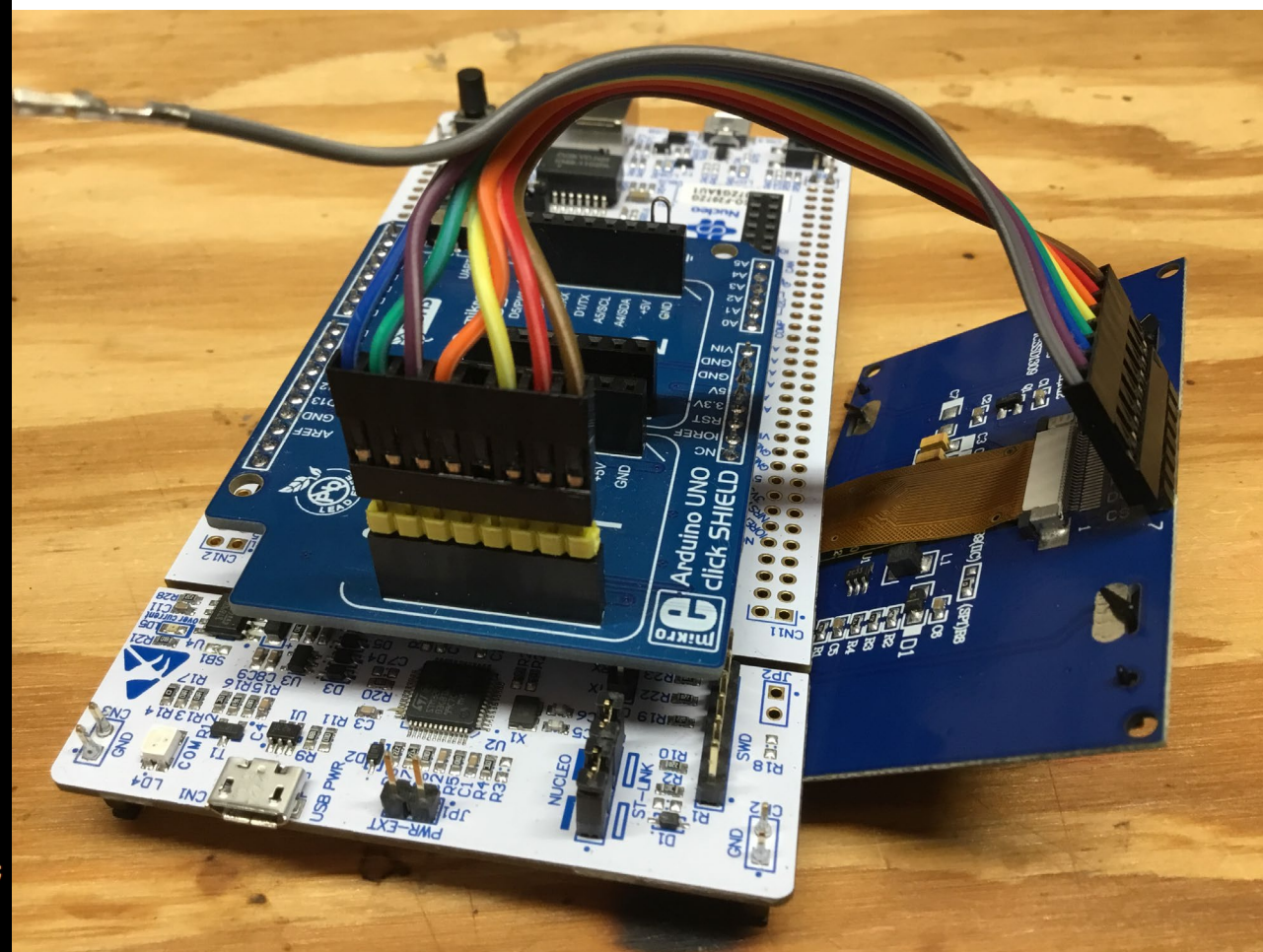
C ssd1309.c 4 X

Core > Src > C ssd1309.c > sendCmd

```

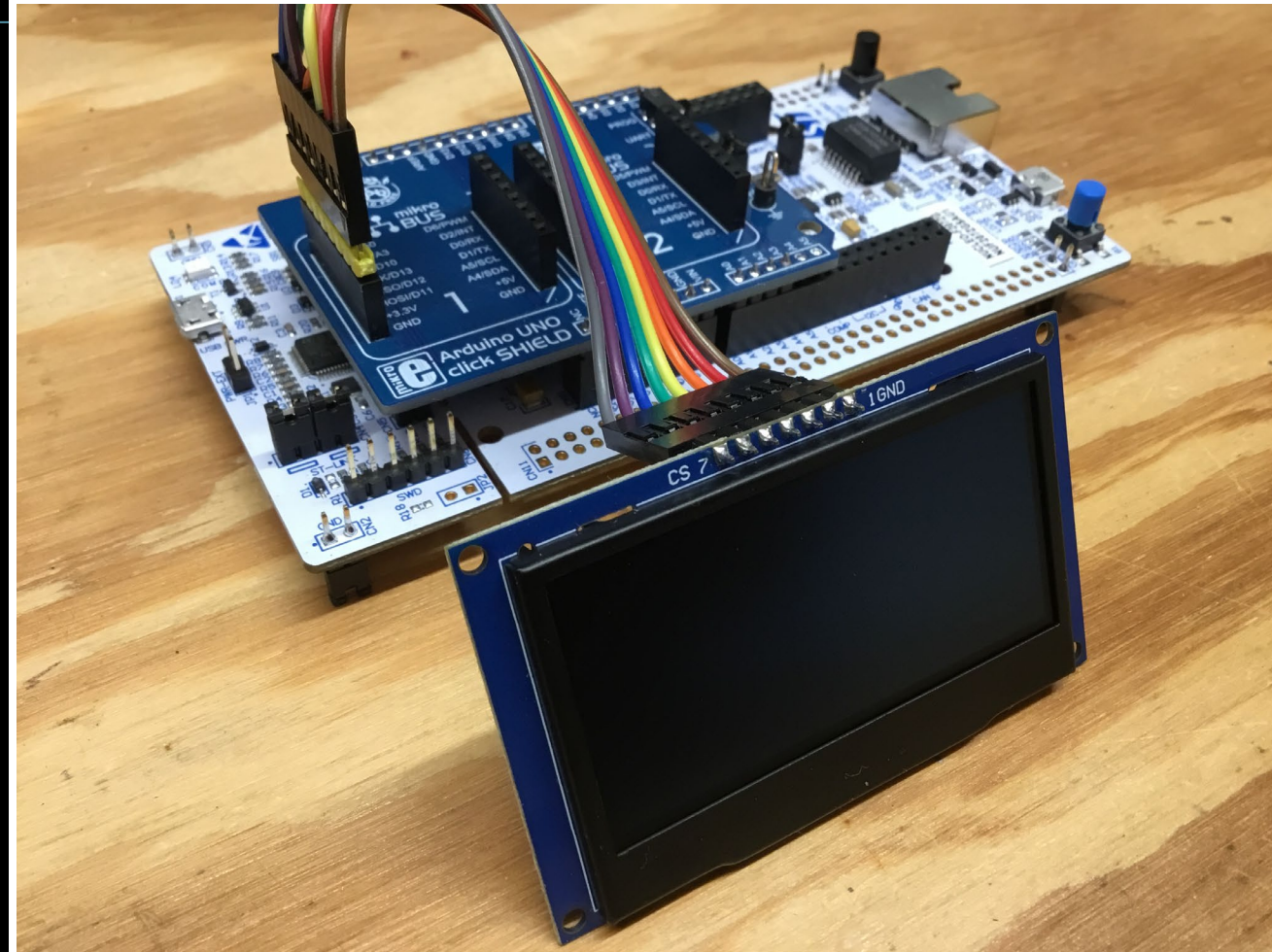
1  //*****
2  /** SSD1306 CLICK DRIVER
3  /** VERSION 1.00 A CEC
4  /** COMPILED USING GNU
5  /** WRITTEN BY FRED EADY
6  /** LAST UPDATED 10/20/2025
7  /** CHANGES/ADDITIONS
8  //*****
9  //*****
10 /** INCLUDES
11 //*****
12 #include "main.h"
13 #include "stm32f2xx_hal_def.h"
14 #include "ssd1309.h"
15
16 //*****
17 /** SSD1309 VARIABLES
18 //*****
19 char* curFontPtr;           // Current font.
20 uint8_t fontOffset = 2;    // Font bytes for meta data.
21 uint8_t fontWidth;        // Font width.
22 uint8_t col;              // Cursor column.
23 uint8_t row;              // Cursor row (RAM).
24
25 extern char spiTxBuf[128];
26 extern SPI_HandleTypeDef hspi1;
27
28 void cmdMode(void)
29 {
30     HAL_GPIO_WritePin(GPIOx: DC_GPIO_Port, GPIO_Pin: DC_Pin, PinState: GPIO_PIN_RESET);
31 }
32
33 void dataMode(void)
34 {
35     HAL_GPIO_WritePin(GPIOx: DC_GPIO_Port, GPIO_Pin: DC_Pin, PinState: GPIO_PIN_SET);
36 }

```



The Application Code – main.c

```
C main.c x
Core > Src > C main.c > main
96  int main(void)
118
119  /* Initialize all configured peripherals */
120  MX_GPIO_Init();
121  MX_USART3_UART_Init();
122  MX_USB_OTG_FS_PCD_Init();
123  MX_SPI1_Init();
124  /* USER CODE BEGIN 2 */
125  //*****
126  // INITIALIZE THE SSD1309 DISPLAY AND PRINT TO THE SSD1309
127  //*****
128  initSSD1309();
129  setTextXY(row: 0,col: 0);
130  sprintf(spiTxBuf,"SSD1309 PROJECT");
131  displayString(string: spiTxBuf);
132  setTextXY(row: 2,col: 0);
133  sprintf(spiTxBuf,"NUCLEO-F207ZG ");
134  displayString(string: spiTxBuf);
135  setTextXY(row: 4,col: 0);
136  sprintf(spiTxBuf,"BareMetal App");
137  displayString(string: spiTxBuf);
138  setTextXY(row: 6,col: 0);
139  sprintf(spiTxBuf,"CEC ROCKS!");
140  displayString(string: spiTxBuf);
141  /* USER CODE END 2 */
142
143  /* Infinite loop */
144  /* USER CODE BEGIN WHILE */
145  while (1)
146  {
147      /* USER CODE END WHILE */
148
149      /* USER CODE BEGIN 3 */
150  }
151  /* USER CODE END 3 */
152 }
```



The Application Code – Add `ssd1309.c` to `CMakeLists.txt`

The screenshot shows the Visual Studio Code interface with the Explorer view on the left and the Code editor on the right. The Explorer view shows the project structure for `NUCLEO_F207ZG_SSD1309`, with the `Core > Src` directory expanded. The file `ssd1309.c` is highlighted in the Explorer view, and a red arrow points to it. The Code editor shows the `CMakeLists.txt` file, which has been updated to include `ssd1309.c` as a source file for the executable target.

```
EXPLORER
...
C ssd1309.c 2 x
M CMakeLists.txt x

Core > Src > C ssd1309
M CMakeLists.txt
1 //*****
2 /** SSD1306
3 /** VERSION
4 /** COMPILE
5 /** WRITTEN
6 /** LAST UP
7 /** CHANGES/ADDITIONS
8 //*****
9 //*****
10 /** INCLUDES
11 //*****
12 #include "main.h"
13 #include "stm32f2xx_hal_def.h"
14 #include "ssd1309.h"
15
16 //*****
17 /** SSD1309 VARIABLES
18 //*****
19 char* curFontPtr; // Current font.
20 uint8_t fontOffset = 2; // Font bytes for meta data.
21 uint8_t fontWidth; // Font width.
22 uint8_t col; // Cursor column.
23 uint8_t row; // Cursor row (RAM).
24
25 extern char spiTxBuf[128];
26 extern SPI_HandleTypeDef hspi1;
27
28 void cmdMode(void)
29 {
```

Open the Serial Port

>

Open Serial **1**

recently used ⚙

File: Clear Recently Opened...

Developer: Clear Notebook Editor Type Cache

CMake: Quick Start

CMake: Configure

CMake: Delete Cache and Recon

Debug: Select and Start Debuggi

Please select a serial port

/dev/ttyS0

/dev/ttyS1

/dev/ttyS10

/dev/ttyS11

/dev/ttyS12

/dev/ttyACM0 (STMicroelectronics) **2**

Select a baud rate

115200 **3**

57600

38400

19200

9600

4800

PROBLEMS 6 OUTPUT DEBUG CONSOLE **TERMINAL** PORTS STM32CUBE RTOS

Opened with baud rate: 115200

Unleash MAX

The screenshot displays the Visual Studio Code environment. The main editor shows a C file named `main.c` with the following code:

```

83 PUTCHAR_PROTOTYPE
89 }
90 /* USER CODE END 0 */
91
92 /**
93  * @brief The application entry point.
94  * @retval int
95  */
96 int main(void)
97 {
98
99     /* USER CODE BEGIN 1 */
100
101     /* USER CODE END 1 */
102
103     /* MCU Configuration-----*/
104
105     /* Reset of all peripherals, Initializes the Flash interface and the Systick. */
106     HAL_Init();
107
108     /* USER CODE BEGIN Init */
109
110     /* USER CODE END Init */
111
112     /* Configure the system clock */
113     SystemClock_Config();
114
115     /* USER CODE BEGIN SysInit */
116
117     /* USER CODE END SysInit */
118
119     /* Initialize all configured peripherals */
120     MX_GPIO_Init();
121     MX_USART3_UART_Init();
122     MX_USB_OTG_FS_PCD_Init();
123     MX_SPI1_Init();
    
```

The interface also shows a file explorer on the right with the project structure:

- NUCLEO_F207ZG_SSD1309
 - .settings
 - .vscode
 - build
 - cmake
 - stm32cubemx
 - gcc-arm-none-eabi.cmake
 - starm-clang.cmake
 - Core
 - Inc
 - gpio.h
 - main.h
 - spi.h
 - ssd1309.h
 - ssd1309Fonts5x7.h
 - ssd1309Fonts8x8.h
 - stm32f2xx_hal_conf.h
 - stm32f2xx_it.h
 - usart.h
 - usb_otg.h

The Run menu is open, showing the following options:

- Start Debugging (F5)
- Run Without Debugging (Ctrl+F5)
- Stop Debugging (Shift+F5)
- Restart Debugging (Ctrl+Shift+F5)
- Open Configurations
- Add Configuration...
- Step Over (F10)
- Step Into (F11)
- Step Out (Shift+F11)
- Continue (F5)
- Toggle Breakpoint (F9)
- New Breakpoint
- Enable All Breakpoints
- Disable All Breakpoints
- Remove All Breakpoints
- Install Additional Debuggers...

The bottom status bar shows the current line and column: Ln 106, Col 1, Spaces: 2, UTF-8, CRLF.

MAX Delivers!

```
C main.c x
Core > Src > C main.c > main
96 int main(void)
118
119 /* Initialize all configured peripheral
120 MX_GPIO_Init();
121 MX_USART3_UART_Init();
122 MX_USB_OTG_FS_PCD_Init();
123 MX_SPI1_Init();
124 /* USER CODE BEGIN 2 */
125 //*****
126 // INITIALIZE THE SSD1309 DISPLAY AND P
127 //*****
128 initSSD1309();
129 setTextXY(row: 0,col: 0);
130 sprintf(spiTxBuf,"SSD1309 PROJECT");
131 displayString(string: spiTxBuf);
132 setTextXY(row: 2,col: 0);
133 sprintf(spiTxBuf,"NUCLEO-F207ZG ");
134 displayString(string: spiTxBuf);
135 setTextXY(row: 4,col: 0);
136 sprintf(spiTxBuf,"BareMetal App");
137 displayString(string: spiTxBuf);
138 setTextXY(row: 6,col: 0);
139 sprintf(spiTxBuf,"CEC ROCKS!");
140 displayString(string: spiTxBuf);
141 /* USER CODE END 2 */
142
143 /* Infinite loop */
144 /* USER CODE BEGIN WHILE */
145 while (1)
146 {
147     /* USER CODE END WHILE */
148
149     /* USER CODE BEGIN 3 */
150 }
151 /* USER CODE END 3 */
152 }
```



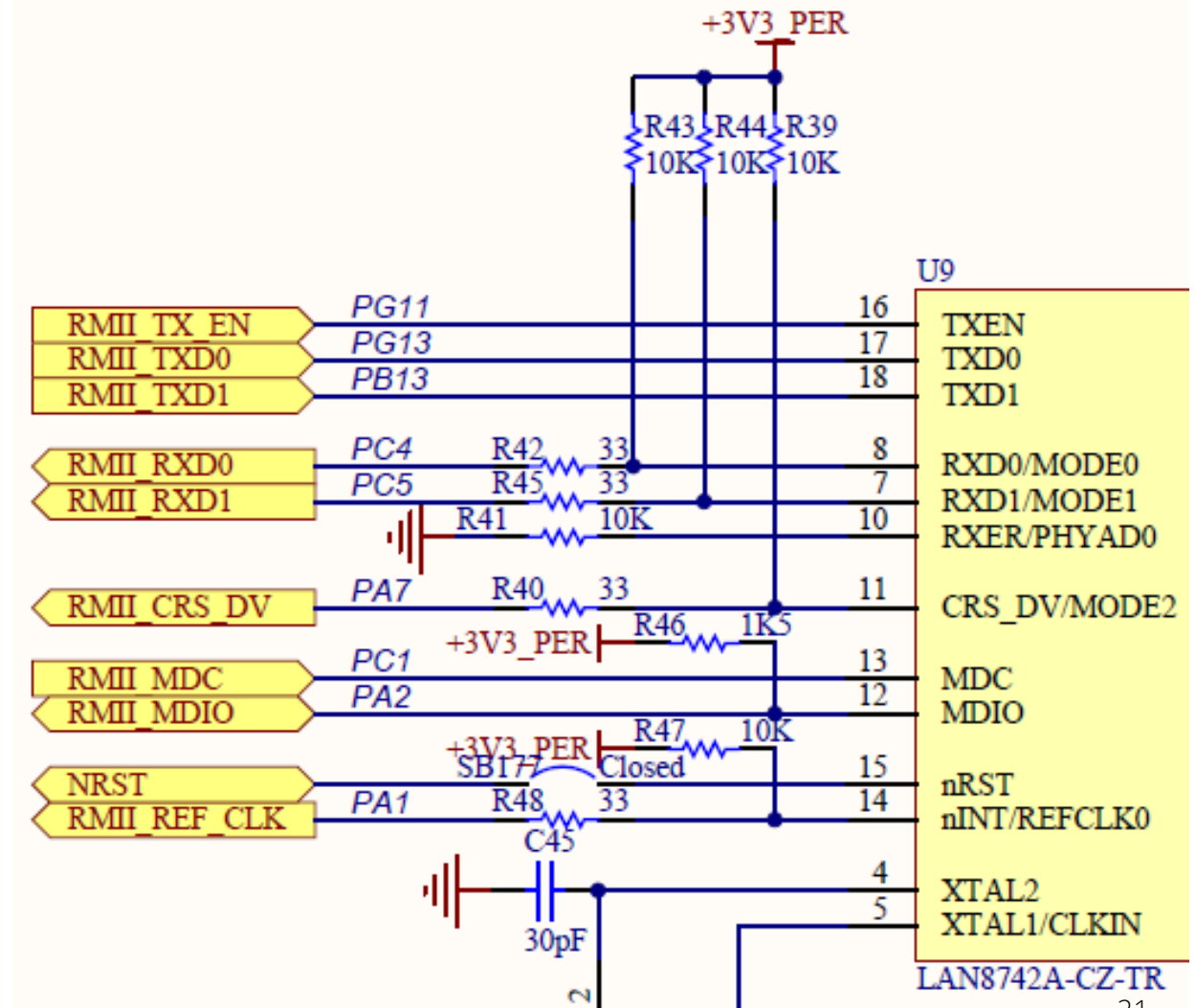
Next Time...

Thank you for attending!!!

Please consider the resources below:

- [Today's Download Package](#)
- www.st.com
- digikey.com

MORE TO COME..





DesignNews

Thank You

Sponsored by

DigiKey

