CPE3500: Embedded Digital Signal Processing

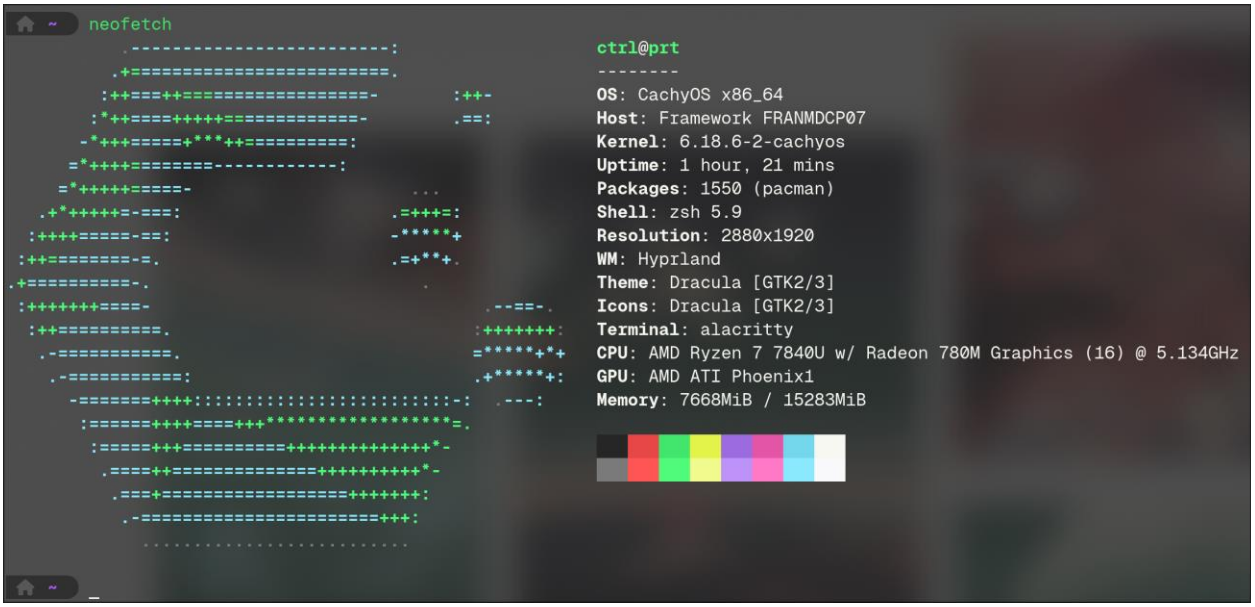
Lab-1 Datasheet: Getting Started with STM32CubeIDE – LED Blinking with HAL

Student Name:	Chance
Student ID:	001095302
Date:	26/01/27

Task-1: Software Installation and Setup

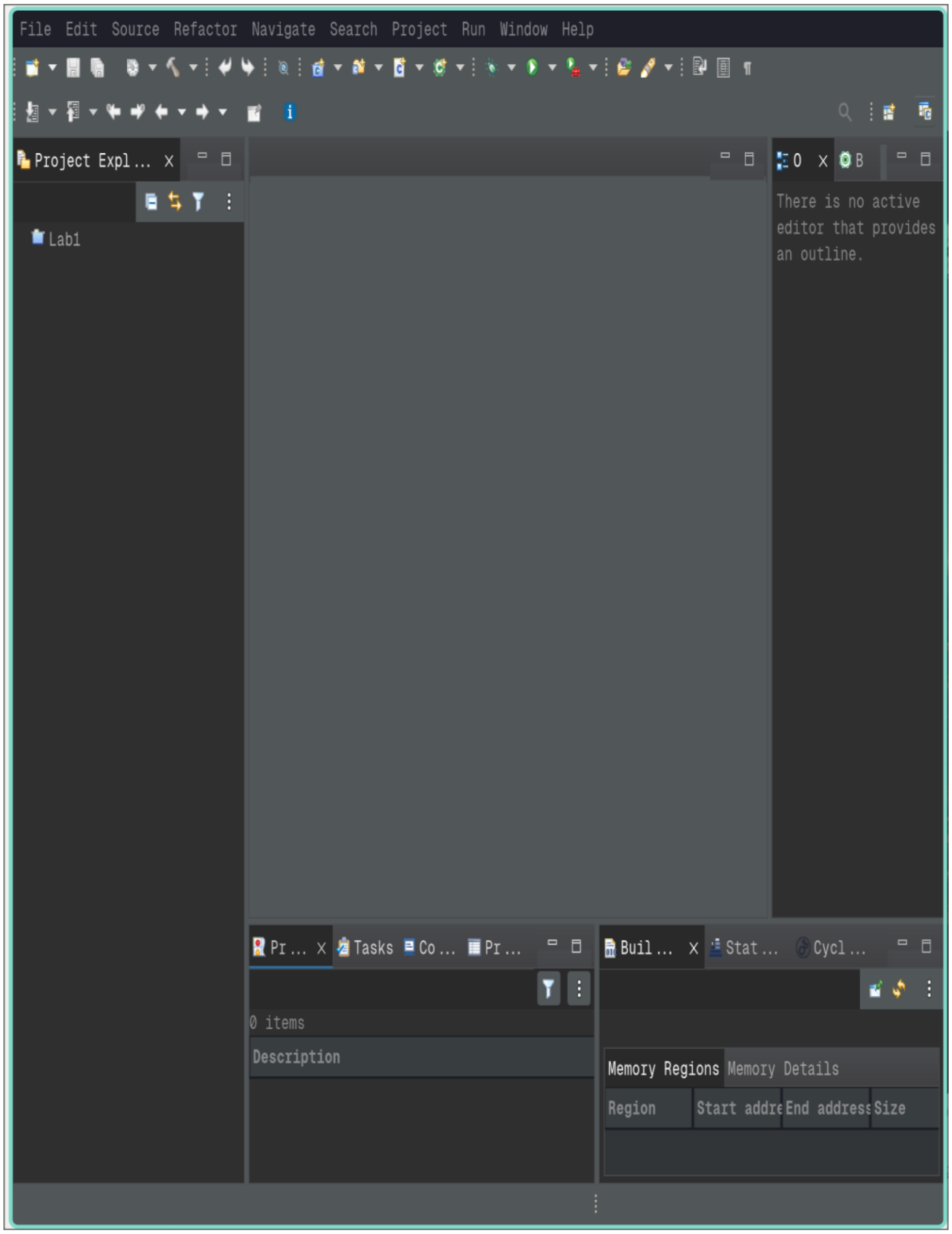
1) **What operating system are you using?**

Linux



```
~ neofetch
ctrl@prt
--------
OS: CachyOS x86_64
Host: Framework FRANMDCP07
Kernel: 6.18.6-2-cachyos
Uptime: 1 hour, 21 mins
Packages: 1550 (pacman)
Shell: zsh 5.9
Resolution: 2880x1920
WM: Hyprland
Theme: Dracula [GTK2/3]
Icons: Dracula [GTK2/3]
Terminal: alacritty
CPU: AMD Ryzen 7 7840U w/ Radeon 780M Graphics (16) @ 5.134GHz
GPU: AMD ATI Phoenix1
Memory: 7668MiB / 15283MiB
~ _
```

2) **Insert a screenshot of your “Welcome to STM32CubeIDE” Page below:**



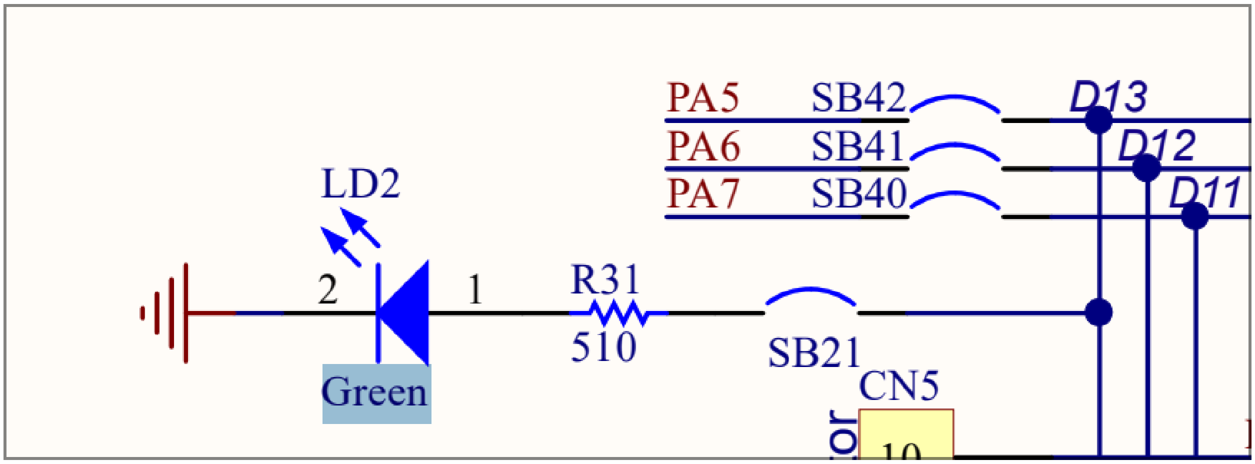
There is no active editor that provides an outline.

0 items

Description

Region	Start addre	End address	Size

3) Insert the screenshot of the circuit diagram of the on-board (user) LED from the STM32 Nucleo Board Schematic:

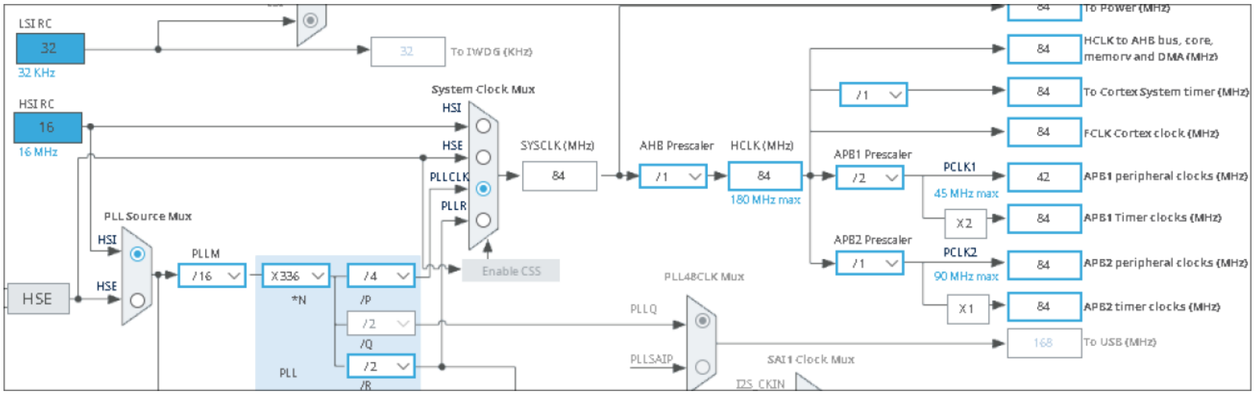


4) **Why is GPIO mode for PA5 (LED) set to “Output Push Pull” by default.**

Because that gives natural handling where high/1 is on and low/0 is off.

5) **What is the default clock source used (in Clock Configuration) to achieve an 80 MHz system clock?**

For the Nucleo-F446RE its 16MHz for 84MHz from the PLLCLK/HSI RC



Task-2: Blink LED

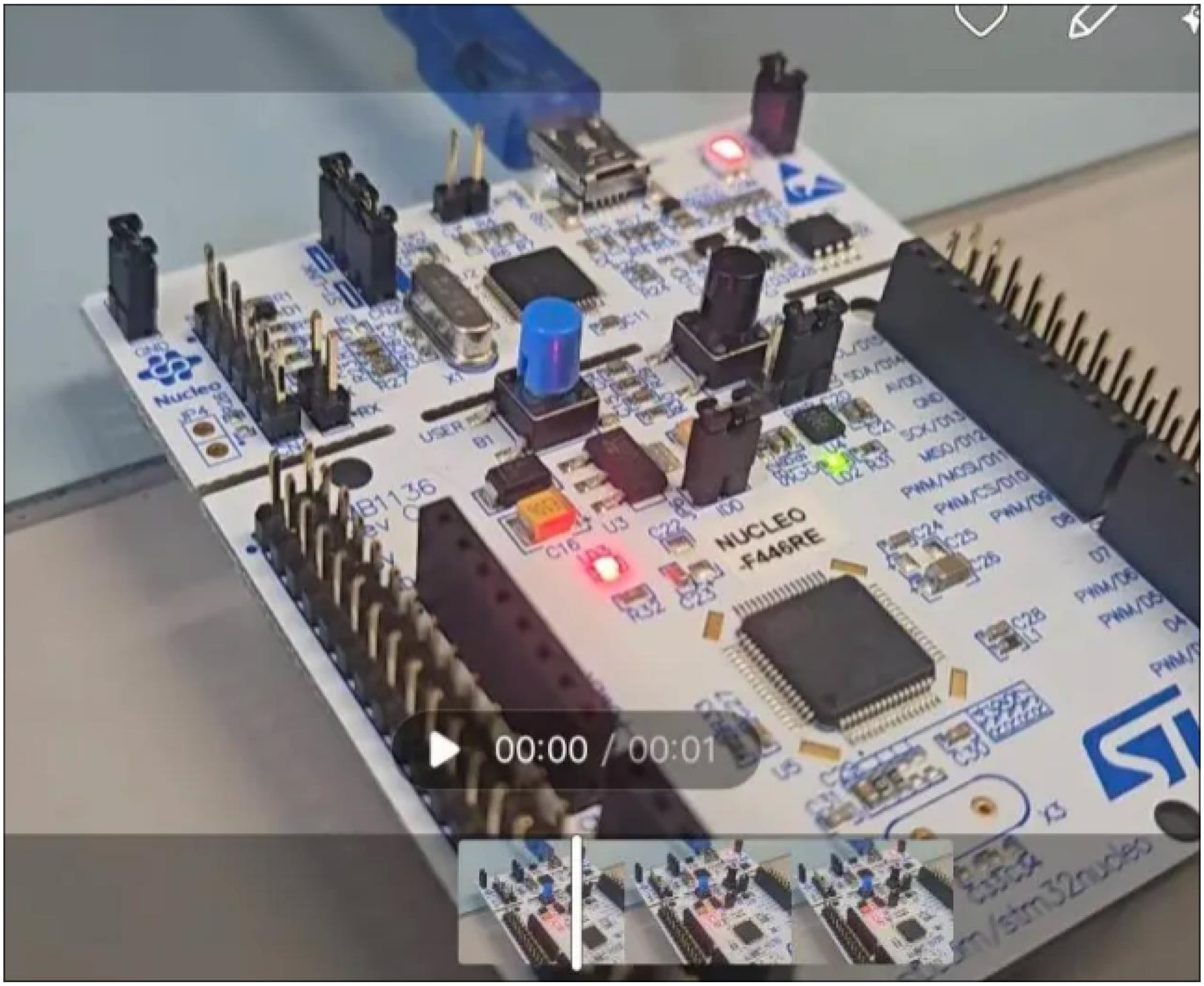
6) **Insert the code in main() function which toggles the LED every 1 second:**

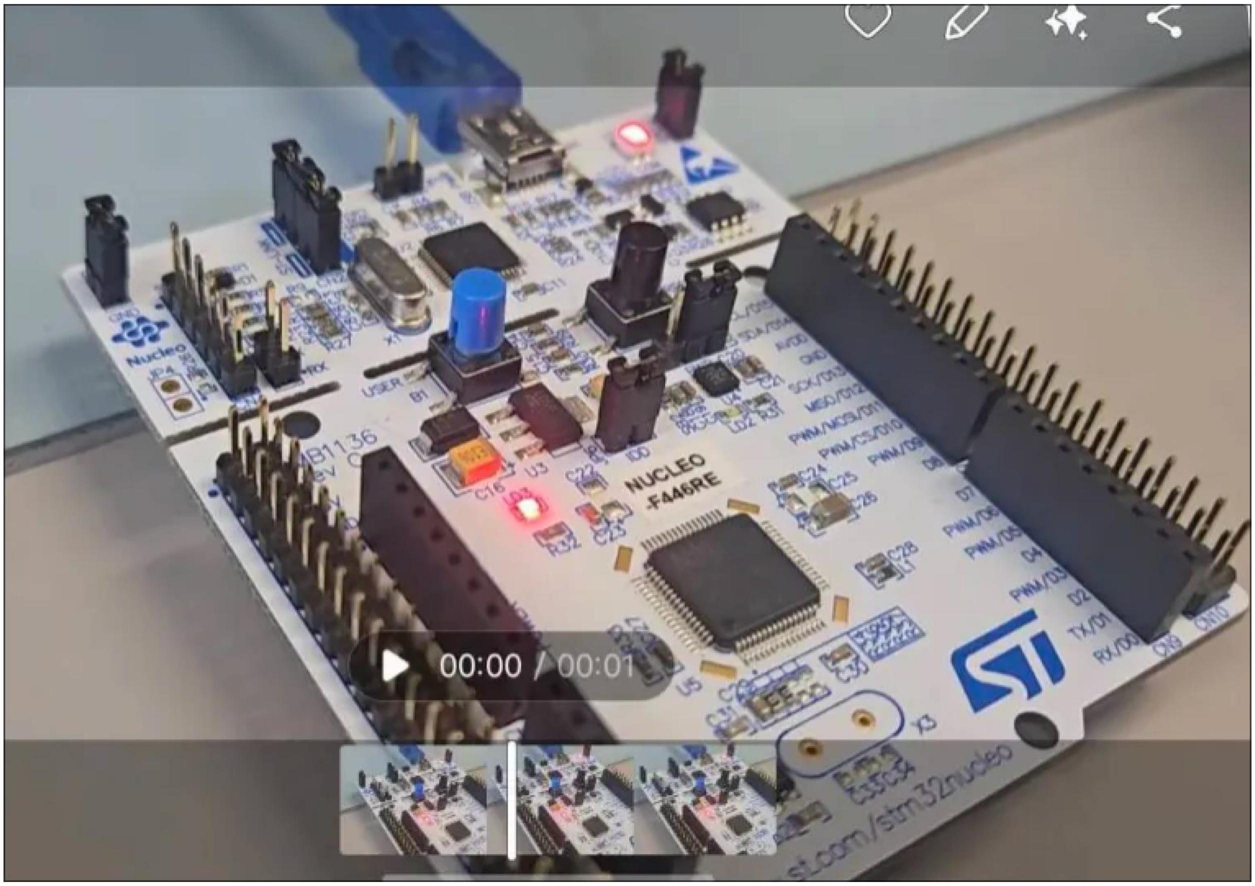
```
int main(void) {

/* 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_USART2_UART_Init(); / USER CODE
BEGIN 2 */

/* USER CODE END 2 */

/* Infinite loop / / USER CODE BEGIN WHILE / while (1) { / USER CODE END WHILE /
HAL_GPIO_TogglePin(GPIOA, GPIO_PIN_5); HAL_Delay(1000); / USER CODE BEGIN 3 / } / USER
CODE END 3 */ }
```

Or in less aids formatting

```
While (1) {
        HAL_GPIO_TogglePin(GPIOA, GPIO_PIN_5);
        HAL_Delay(1000);
}
```

7) **Capture a short video of your on-board LED blinking and insert two still frames:**
   - Frame 1 : LED ON
   - Frame 2 : LED OFF


NUCLEO
-F446RE
Nucleo
USER
B1
00:00 / 00:01



Task-3: Modify Delay

8) **Insert the modified code which toggles the LED every 0.2 seconds:**

```
While (1) {
        HAL_GPIO_TogglePin(GPIOA, GPIO_PIN_5);
        HAL_Delay(200);
}
```

Task-4: Blink LED

9) **Insert the code in main() function that turns on/off the LED every 1 sec using HAL_GPIO_WritePin():**

```
While (1) {

        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_5, GPIO_SET);

        HAL_Delay(500);

        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_5, GPIO_RESET);

        HAL_Delay(500);

}
```

Reflections

10) **What part of this lab did you find most helpful for understanding embedded systems development?** *(i.e. IDE setup, HAL functions, debugging, etc.)*

    Understanding that toolchains are always going to be an issue.

11) **What challenges did you encounter during this lab, and how did you overcome them?** (i.e. installation issues, driver problems, hardware issues, code errors, etc.)

My build env wasn’t using gcc-arm-no-eabi or whatever, so I set an env var so it would look for it properly.