TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

****

BÁO CÁO

HỆ NHÚNG (IT4210)

BÀI THỰC HÀNH SỐ 3

Xây dựng ứng dụng trên hệ nhúng ARM (Online)

Nhóm sinh viên thực hiện: **3B**

1. Nguyễn Duy Khai	**20183771**
2. Đặng Quang Thắng	**20183829**

Giảng viên hướng dẫn: **TS. Đỗ Công Thuần**

Hà Nội, tháng 1 năm 2021

MỤC LỤC

Về phân công:

Cả 2 thành viên cùng làm và chữa bài cho nhau.

I. Các bài tập lập trình

3.1 Tìm hiểu CPU STM32F103C6

Xác định các thông số:

Dung lượng ROM	16 hoặc 32 Kbytes of Flash memory
Dung lượng RAM	6 hoặc 10 Kbytes of SRAM
Tần số clock tối đa của CPU	72 MHz
Điện áp hoạt động	2.0V -> 3.6V
Số chân vào ra	51
Số bộ timer	6
Số cổng USART	2
Dòng điện sử dụng khi CPU chạy ở 72 MHz	Chế độ Run, code xử lý dữ liệu ở trong ROM: • Đồng hồ ngoài, tất cả thiết bị ngoại vi bật: 45-46 mA • Đồng hồ ngoài, tất cả thiết bị ngoại vi tắt: 30-31 mA Chế độ Run, code xử lý dữ liệu ở trong RAM: • Đồng hồ ngoài, tất cả thiết bị ngoại vi bật: 41-42 mA • Đồng hồ ngoài, tất cả thiết bị ngoại vi tắt: 27-28 mA Mức tiêu thụ điện cao nhất trong chế độ Sleep: • Đồng hồ ngoài, tất cả thiết bị ngoại vi bật: 26-27 mA • Đồng hồ ngoài, tất cả thiết bị ngoại vi tắt: 7,5-8 mA
Dòng điện sử dụng khi CPU chạy ở 8 MHz	Chế độ Run, code xử lý dữ liệu ở trong ROM: • Đồng hồ ngoài, tất cả thiết bị ngoại vi bật: 7-8 mA • Đồng hồ ngoài, tất cả thiết bị ngoại vi tắt: 6-7 mA

	Chế độ Run, code xử lý dữ liệu ở trong RAM: • Đồng hồ ngoài, tất cả thiết bị ngoại vi bật: 6-7 mA • Đồng hồ ngoài, tất cả thiết bị ngoại vi tắt: 5-6 mA Mức tiêu thụ điện cao nhất trong chế độ Sleep: • Đồng hồ ngoài, tất cả thiết bị ngoại vi bật: 4-5 mA • Đồng hồ ngoài, tất cả thiết bị ngoại vi tắt: 3-4 mA
Danh sách chân ADC input	PA0-PA7, PB0, PB1, PC0-PC5
Dánh sách chân ngắt ngoài	37 chân GPIOs
Các chân RX/TX của USART1	Chân PA9, PA10

3.2 Tìm hiểu mã nguồn mẫu

Main.c => nơi chứa code, khởi tạo biến, hàm cần dùng.

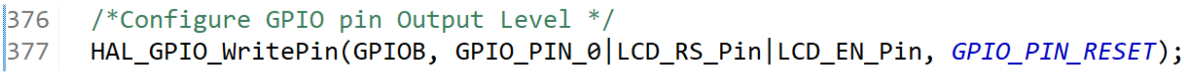
Drivers => nơi chứa các file thư viện cần thiết để chạy chương trình

Project1.ioc => chỉnh cấu hình project.

3.3 Thiết lập và lập trình GPIO.

a. Chân PB0 thuộc cổng B được cấu hình ở dòng code nào, hàm nào?

Code nằm ở dòng số 377:



```
376   /*Configure GPIO pin Output Level */
377   HAL_GPIO_WritePin(GPIOB, GPIO_PIN_0|LCD_RS_Pin|LCD_EN_Pin, GPIO_PIN_RESET);
```

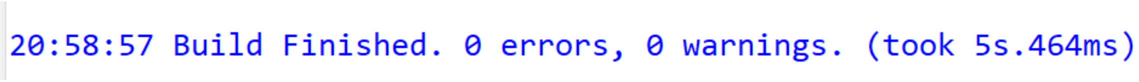
Nằm ở hàm: `static void MX_GPIO_Init(void)`

b. Chức năng của vòng lặp main loop trong hàm main() là gì?

Lặp vô hạn thao tác gửi giá trị tương ứng GPIO_PIN_SET và GPIO_PIN_SET ra chân PB0 (chân pin_0 cổng GPIO_B), mỗi thao tác cách nhau 0.5s.

3.4 Cấu hình ngoại vi

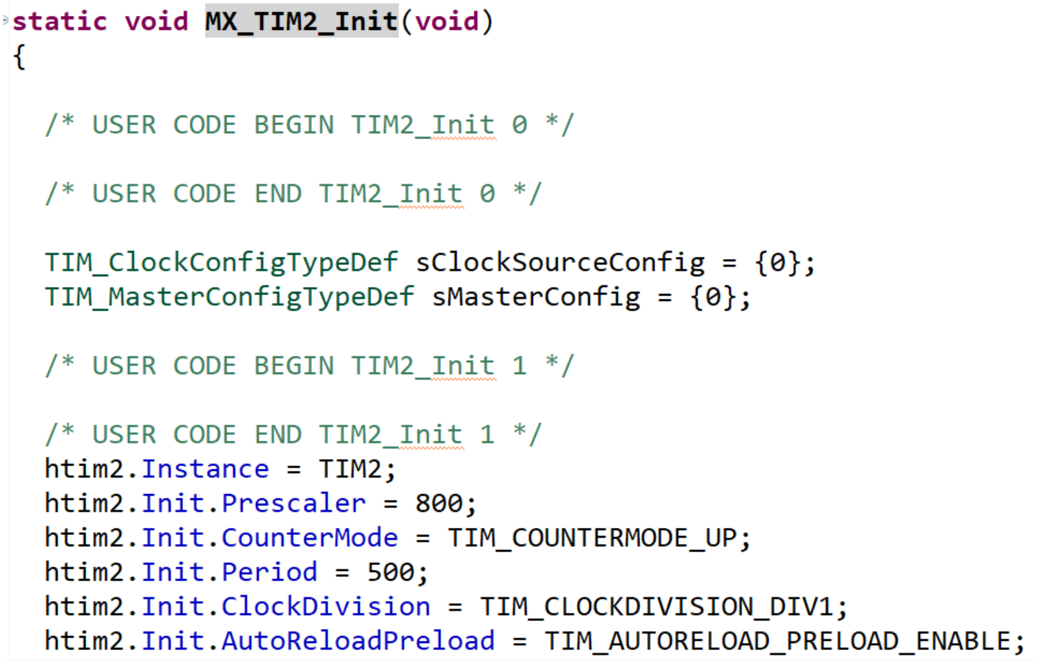
Không có lỗi xảy ra, build thành công



```
20:58:57 Build Finished. 0 errors, 0 warnings. (took 5s.464ms)
```

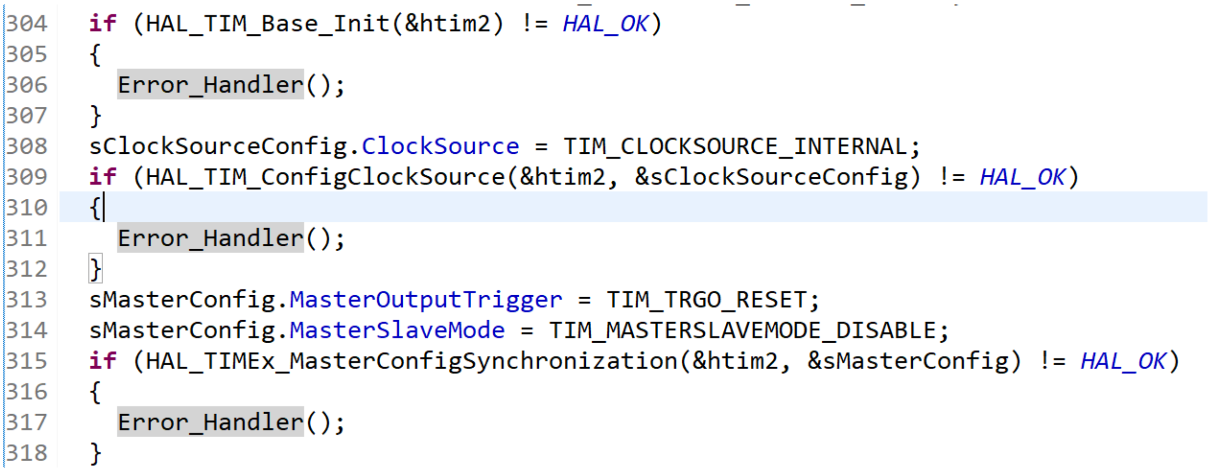
3.5 Lập trình xử lý nút bấm

Đoạn code cấu hình TIM2:



```
static void MX_TIM2_Init(void)
{

  /* USER CODE BEGIN TIM2_Init 0 */

  /* USER CODE END TIM2_Init 0 */

  TIM_ClockConfigTypeDef sClockSourceConfig = {0};
  TIM_MasterConfigTypeDef sMasterConfig = {0};

  /* USER CODE BEGIN TIM2_Init 1 */

  /* USER CODE END TIM2_Init 1 */
  htim2.Instance = TIM2;
  htim2.Init.Prescaler = 800;
  htim2.Init.CounterMode = TIM_COUNTERMODE_UP;
  htim2.Init.Period = 500;
  htim2.Init.ClockDivision = TIM_CLOCKDIVISION_DIV1;
  htim2.Init.AutoReloadPreload = TIM_AUTORELOAD_PRELOAD_ENABLE;
```

Xác định hander của TIM2:



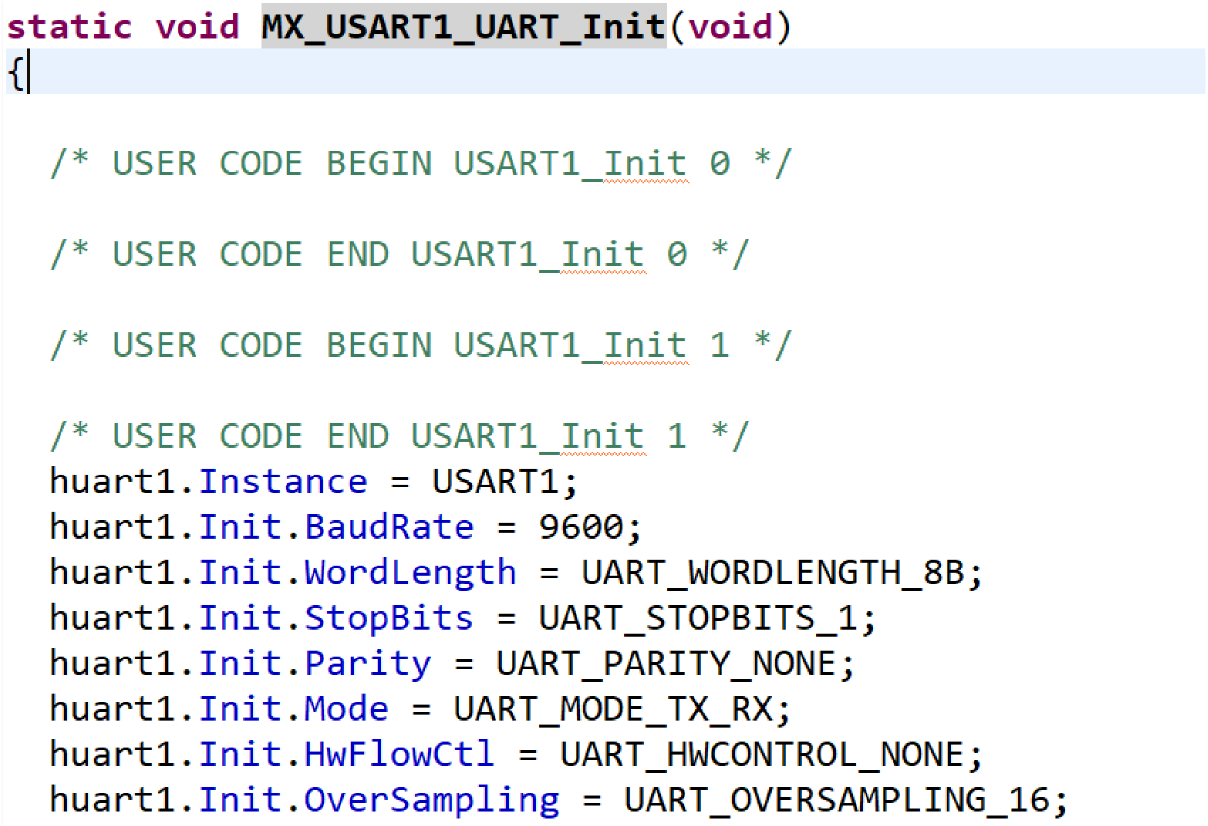
```
  if (HAL_TIM_Base_Init(&htim2) != HAL_OK)
  {
    Error_Handler();
  }
  sClockSourceConfig.ClockSource = TIM_CLOCKSOURCE_INTERNAL;
  if (HAL_TIM_ConfigClockSource(&htim2, &sClockSourceConfig) != HAL_OK)
  {
    Error_Handler();
  }
  sMasterConfig.MasterOutputTrigger = TIM_TRGO_RESET;
  sMasterConfig.MasterSlaveMode = TIM_MASTERSLAVEMODE_DISABLE;
  if (HAL_TIMEx_MasterConfigSynchronization(&htim2, &sMasterConfig) != HAL_OK)
  {
    Error_Handler();
  }
```

Lập trình 2 đoạn code quan trọng của:

3.6 Lập trình xử lý ngắt timer

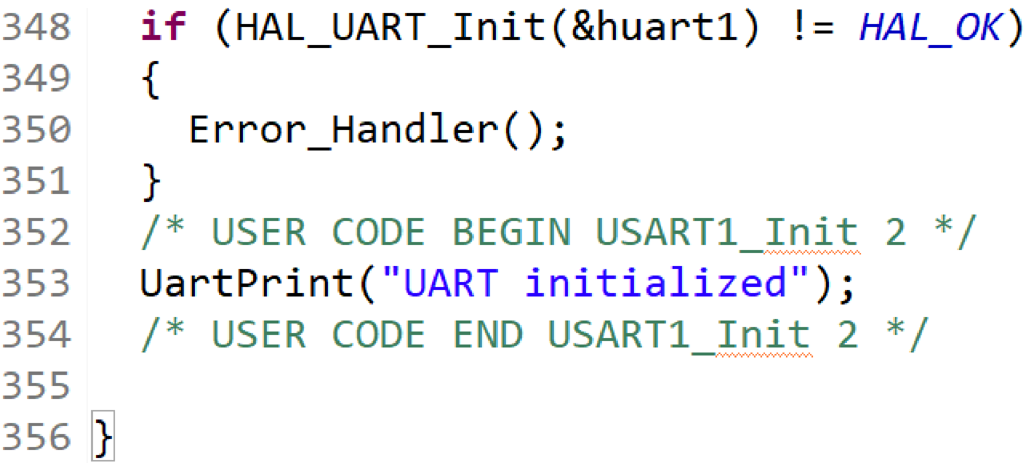
3.7 Lập trình ghép nối cổng nối tiếp UART

Code cấu hình USART1:



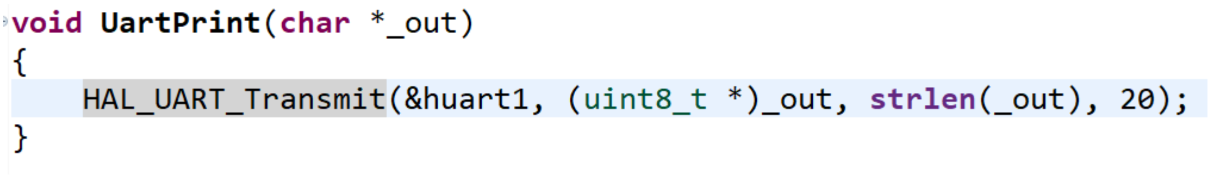
```
static void MX_USART1_UART_Init(void)
{

  /* USER CODE BEGIN USART1_Init 0 */

  /* USER CODE END USART1_Init 0 */

  /* USER CODE BEGIN USART1_Init 1 */

  /* USER CODE END USART1_Init 1 */
  huart1.Instance = USART1;
  huart1.Init.BaudRate = 9600;
  huart1.Init.WordLength = UART_WORDLENGTH_8B;
  huart1.Init.StopBits = UART_STOPBITS_1;
  huart1.Init.Parity = UART_PARITY_NONE;
  huart1.Init.Mode = UART_MODE_TX_RX;
  huart1.Init.HwFlowCtl = UART_HWCONTROL_NONE;
  huart1.Init.OverSampling = UART_OVERSAMPLING_16;
```

Xác định handler của USART1:



```
  if (HAL_UART_Init(&huart1) != HAL_OK)
  {
    Error_Handler();
  }
  /* USER CODE BEGIN USART1_Init 2 */
  UartPrint("UART initialized");
  /* USER CODE END USART1_Init 2 */

}
```

Xác định 2 hàm quan trọng của USART driver:



```
void UartPrint(char *_out)
{
    HAL_UART_Transmit(&huart1, (uint8_t *)_out, strlen(_out), 20);
}
```

=> Dùng để gửi dữ liệu qua USART (như trong ảnh là gửi 1 xâu ký tự). Tương tự hàm receive được dung để nhận dữ liệu từ USART.

3.8 Lập trình điều khiển LCD 16x2

3.9 Code lập trình để tin ra 2 dòng chữ trên LCD:

(3.8 và 3.9 có trong phần code)