Đại học Sư phạm Kỹ thuật Thành phố Hồ Chí Minh

Khoa Cơ khí Chế tạo Máy

Bộ môn Cơ Điện tử

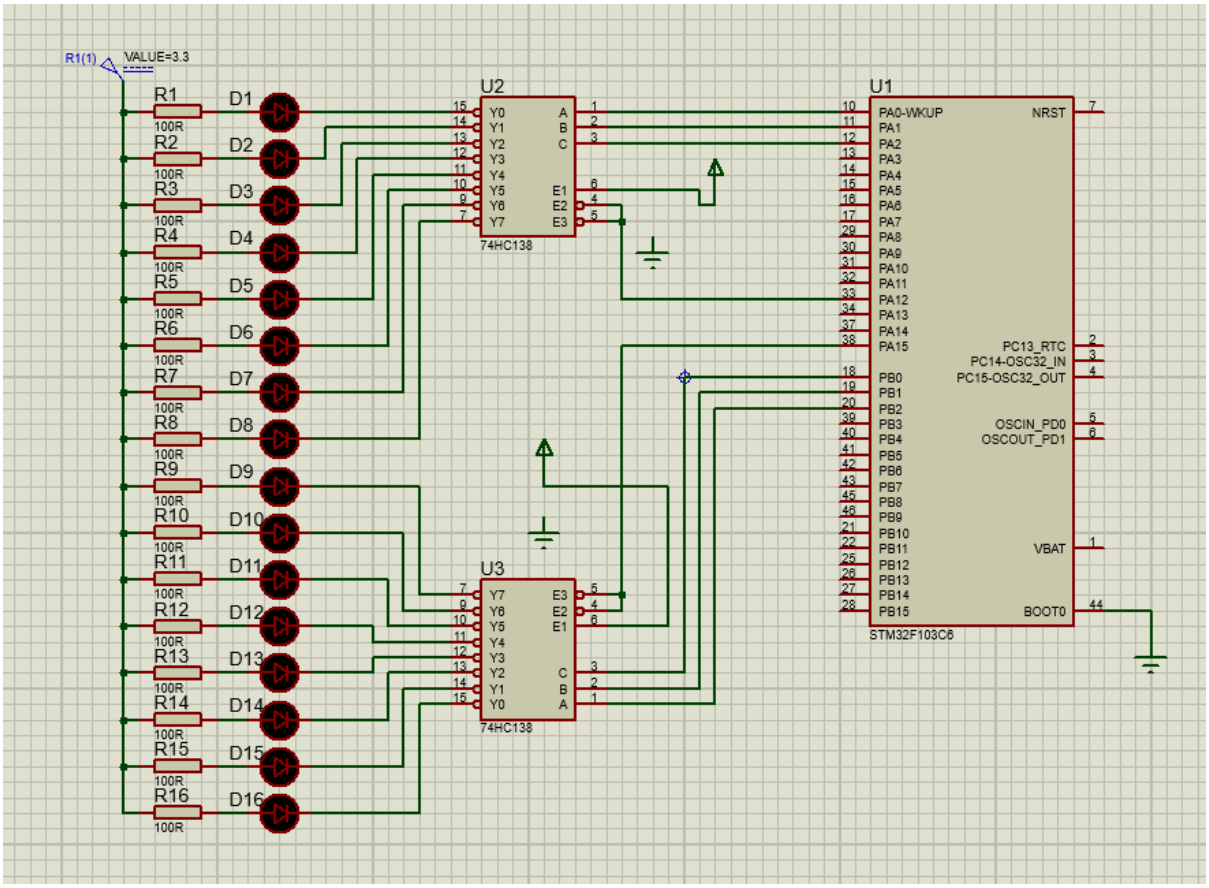
BÁO CÁO DỰ ÁN

1. Thông tin người thực hiện:

STT	Người thực hiện	MSSV	Ngày
01	Nguyễn Hữu Chí	20146479	31/3/2023

2. Yêu cầu dự án

- Lập trình 1 điểm sáng chạy từ D1 sang D16 và lặp lại. Trì hoãn giữa 2 trạng thái 0.3s



3. Nội dung:

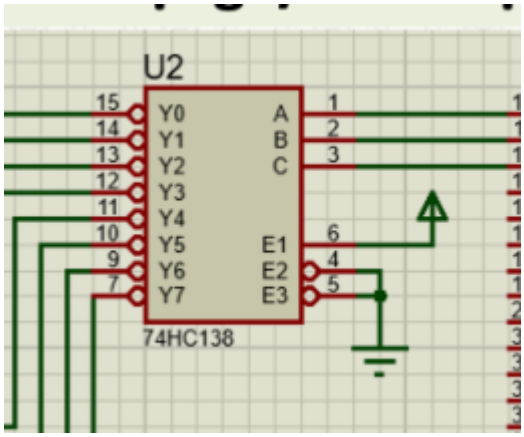
3.1. Phân tích dự án:

- Để Led sáng được thì chân Yx của IC 74HC138 tương ứng với Led phải cấp mức 0

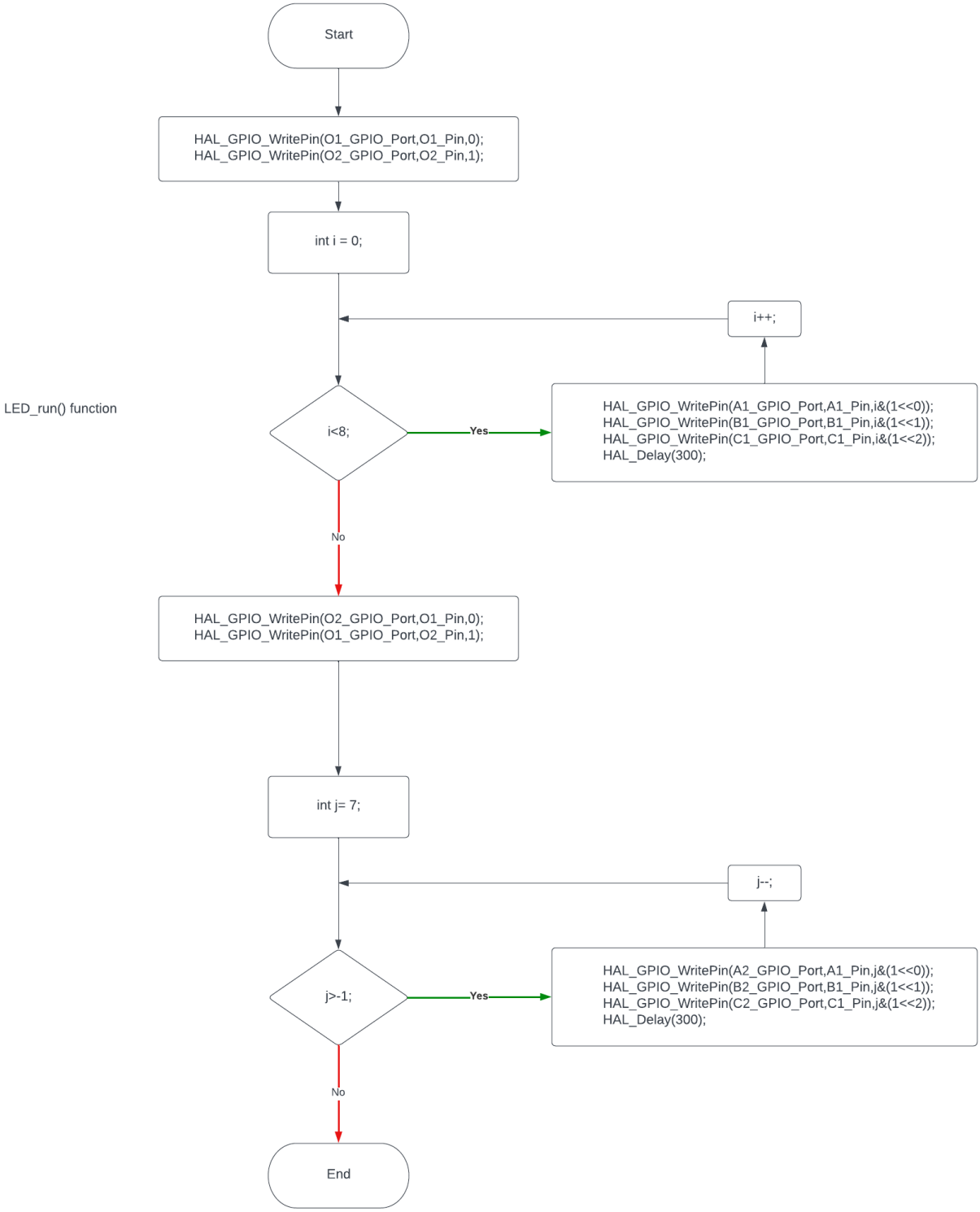
- Tại 1 thời điểm chỉ có 1 Led sáng tương ứng chỉ có 1 chân Yx của IC 74HC138 được xuất mức 0 còn lại xuất mức 1

- Sau đó thay đổi chân Yx xuất mức 0 lần lượt theo thứ tự từ trên xuống (như hình trên) tương ứng với Led D1 -> D16

- Vấn đề là nếu mắc mạch như hình dưới đây thì các mức ban đầu của ABC là 000 (do PAx tương ứng set mức LOW) thì lúc này Y0 xuất mức 0 (D1, D16 sáng) và các chân còn lại xuất mức 1. Vì vậy ta cần tạo ra trạng thái tắt của tất cả các LED bằng cách kéo chân E2 và E3 lên mức 1. Do vậy ta sẽ nối E2, E3 vào 1 chân vi xử lý để có thể điều chỉnh lên 1 hoặc xuống 0 khi cần



3.2. Lưu đồ lập trình:



3.3. Mã nguồn chương trình:

```
/* USER CODE BEGIN Header */
/**

******************************************************************************
  * @file           : main.c
  * @brief          : Main program body

******************************************************************************
  * @attention
  *
  * Copyright (c) 2023 STMicroelectronics.
  * All rights reserved.
  *
  * This software is licensed under terms that can be found in the LICENSE file
  * in the root directory of this software component.
  * If no LICENSE file comes with this software, it is provided AS-IS.
  *

******************************************************************************
  */
/* USER CODE END Header */
/* Includes ------------------------------------------------------------------*/
#include "main.h"

/* Private includes ----------------------------------------------------------*/
/* USER CODE BEGIN Includes */

/* USER CODE END Includes */

/* Private typedef -----------------------------------------------------------*/
/* USER CODE BEGIN PTD */
```

```
/* USER CODE END PTD */

/* Private define ------------------------------------------------------------*/
/* USER CODE BEGIN PD */
/* USER CODE END PD */

/* Private macro -------------------------------------------------------------*/
/* USER CODE BEGIN PM */

/* USER CODE END PM */

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

/* USER CODE BEGIN PV */

/* USER CODE END PV */

/* Private function prototypes -----------------------------------------------*/
void SystemClock_Config(void);
static void MX_GPIO_Init(void);
/* USER CODE BEGIN PFP */

/* USER CODE END PFP */

/* Private user code ---------------------------------------------------------*/
/* USER CODE BEGIN 0 */
void Led_run() {
	HAL_GPIO_WritePin(O1_GPIO_Port,O1_Pin,0); //ON1
	HAL_GPIO_WritePin(O2_GPIO_Port,O2_Pin,1); //OFF2
	for (int i = 0; i < 8; i++) {
		HAL_GPIO_WritePin(A1_GPIO_Port,A1_Pin,i&(1<<0));
		HAL_GPIO_WritePin(B1_GPIO_Port,B1_Pin,i&(1<<1));
		HAL_GPIO_WritePin(C1_GPIO_Port,C1_Pin,i&(1<<2));
```

```
            HAL_Delay(300);
        }
        HAL_GPIO_WritePin(O2_GPIO_Port,O2_Pin,0);  //ON2
        HAL_GPIO_WritePin(O1_GPIO_Port,O1_Pin,1); //OFF1
            for (int j = 7; j > -1; j--) {
            HAL_GPIO_WritePin(A2_GPIO_Port,A2_Pin,j&(1<<0));
            HAL_GPIO_WritePin(B2_GPIO_Port,B2_Pin,j&(1<<1));
            HAL_GPIO_WritePin(C2_GPIO_Port,C2_Pin,j&(1<<2));
            HAL_Delay(300);
        }

}
/* USER CODE END 0 */

/**
  * @brief  The application entry point.
  * @retval int
  */
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();
  /* USER CODE BEGIN 2 */

  /* USER CODE END 2 */

  /* Infinite loop */
  /* USER CODE BEGIN WHILE */
  while (1)
  {
    /* USER CODE END WHILE */
                    Led_run();
    /* USER CODE BEGIN 3 */
  }
  /* USER CODE END 3 */
}

/**
  * @brief System Clock Configuration
  * @retval None
  */
void SystemClock_Config(void)
{
  RCC_OscInitTypeDef RCC_OscInitStruct = {0};
  RCC_ClkInitTypeDef RCC_ClkInitStruct = {0};

  /** Initializes the RCC Oscillators according to the specified parameters
  * in the RCC_OscInitTypeDef structure.
```

```
  */
  RCC_OscInitStruct.OscillatorType = RCC_OSCILLATORTYPE_HSI;
  RCC_OscInitStruct.HSIState = RCC_HSI_ON;
  RCC_OscInitStruct.HSICalibrationValue = RCC_HSICALIBRATION_DEFAULT;
  RCC_OscInitStruct.PLL.PLLState = RCC_PLL_NONE;
  if (HAL_RCC_OscConfig(&RCC_OscInitStruct) != HAL_OK)
  {
    Error_Handler();
  }

  /** Initializes the CPU, AHB and APB buses clocks
  */
  RCC_ClkInitStruct.ClockType = RCC_CLOCKTYPE_HCLK|RCC_CLOCKTYPE_SYSCLK
                              |RCC_CLOCKTYPE_PCLK1|RCC_CLOCKTYPE_PCLK2;
  RCC_ClkInitStruct.SYSCLKSource = RCC_SYSCLKSOURCE_HSI;
  RCC_ClkInitStruct.AHBCLKDivider = RCC_SYSCLK_DIV1;
  RCC_ClkInitStruct.APB1CLKDivider = RCC_HCLK_DIV1;
  RCC_ClkInitStruct.APB2CLKDivider = RCC_HCLK_DIV1;

  if (HAL_RCC_ClockConfig(&RCC_ClkInitStruct, FLASH_LATENCY_0) != HAL_OK)
  {
    Error_Handler();
  }
}

/**
  * @brief GPIO Initialization Function
  * @param None
  * @retval None
  */
static void MX_GPIO_Init(void)
{
  GPIO_InitTypeDef GPIO_InitStruct = {0};
```

```
  /* GPIO Ports Clock Enable */
  __HAL_RCC_GPIOA_CLK_ENABLE();
  __HAL_RCC_GPIOB_CLK_ENABLE();

  /*Configure GPIO pin Output Level */
  HAL_GPIO_WritePin(GPIOA, A1_Pin|B1_Pin|C1_Pin|O1_Pin
                    |O2_Pin, GPIO_PIN_RESET);

  /*Configure GPIO pin Output Level */
  HAL_GPIO_WritePin(GPIOB, C2_Pin|B2_Pin|A2_Pin, GPIO_PIN_RESET);

  /*Configure GPIO pins : A1_Pin B1_Pin C1_Pin O1_Pin
                     O2_Pin */
  GPIO_InitStruct.Pin = A1_Pin|B1_Pin|C1_Pin|O1_Pin
                    |O2_Pin;
  GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP;
  GPIO_InitStruct.Pull = GPIO_NOPULL;
  GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;
  HAL_GPIO_Init(GPIOA, &GPIO_InitStruct);

  /*Configure GPIO pins : C2_Pin B2_Pin A2_Pin */
  GPIO_InitStruct.Pin = C2_Pin|B2_Pin|A2_Pin;
  GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP;
  GPIO_InitStruct.Pull = GPIO_NOPULL;
  GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;
  HAL_GPIO_Init(GPIOB, &GPIO_InitStruct);

}

/* USER CODE BEGIN 4 */

/* USER CODE END 4 */

/**
```

```
  * @brief  This function is executed in case of error occurrence.
  * @retval None
  */
void Error_Handler(void)
{
  /* USER CODE BEGIN Error_Handler_Debug */
  /* User can add his own implementation to report the HAL error return state */
  __disable_irq();
  while (1)
  {
  }
  /* USER CODE END Error_Handler_Debug */
}

#ifdef  USE_FULL_ASSERT
/**
  * @brief  Reports the name of the source file and the source line number
  *         where the assert_param error has occurred.
  * @param  file: pointer to the source file name
  * @param  line: assert_param error line source number
  * @retval None
  */
void assert_failed(uint8_t *file, uint32_t line)
{
  /* USER CODE BEGIN 6 */
  /* User can add his own implementation to report the file name and line number,
     ex: printf("Wrong parameters value: file %s on line %d\r\n", file, line) */
  /* USER CODE END 6 */
}
#endif /* USE_FULL_ASSERT */
```

4. Ghi chú khác (nếu có)