Bangladesh University of Engineering and Technology
Department of Electrical and Electronic Engineering

EEE 416 (July 2023)
Microprocessor and Embedded Systems Laboratory

Final Project Report

Section: C2 Group: 07

STM32 Black Pill Development Board

Course Instructors:

Dr. Sajid Muhaimin Choudhury, Assistant Professor

Ayan Biswas Pranta, Lecturer

Signature of Instructor: ____

Academic Honesty Statement:

IMPORTANT! Please carefully read and sign the Academic Honesty Statement, below. <u>Type the student ID and name, and put your signature</u>. ***You will not receive credit for this project experiment unless this statement is signed in the presence of your lab instructor.***

"In signing this statement, We hereby certify that the work on this project is our own and that we have not copied the work of any other students (past or present), and cited all relevant sources while completing this project. We understand that if we fail to honor this agreement, We will each receive a score of ZERO for this project and be subject to failure of this course."

Signature: ____________________ **Full Name: Md. Ridwanul Haque** **Student ID: 1906165**	**Signature:** ____________________ **Full Name: Md. Toky Tazwar** **Student ID: 1906174**
Signature: ____________________ **Full Name: Md. Julkar Naim Joy** **Student ID: 1906181**	**Signature:** ____________________ **Full Name: Raihan Amin Rana** **Student ID: 1906186**

Table of Contents

1 Abstract

Our Project is a compact design and implementation of a development board with stm32 black pill microprocessor which can perform all the experiments of EEE416 microprocessor and embedded systems laboratory. The peripheral devices are placed in a way so that it takes minimum floor area and the GPIO pins are used efficiently to make sure that all the devices are connected to a suitable pin.

2 Introduction

Compact design is extremely important for cost reduction, space efficiency, reliability and so on. This project is an example of efficient compact design and implementation using stm32 black pill. This project can be a good application for wearable health monitoring devices, smart home automation systems, IOT environmental monitoring stations, educational electronics kits, robotics and automation and so on.

3 Design

3.1 Problem Formulation (PO(b))

Our goal is to develop an STM32 based Development Board for laboratory use that will help us in better understanding of STM32 microprocessors. The board should contain all the peripheral devices to accommodate all the features of microprocessors including timer, GPIO ports, Interrupt, Systick Interrupt, ADC & DAC, I2C , SPI and other communication modules etc.

3.1.1 Identification of Scope

The trainer board has the following characteristics:

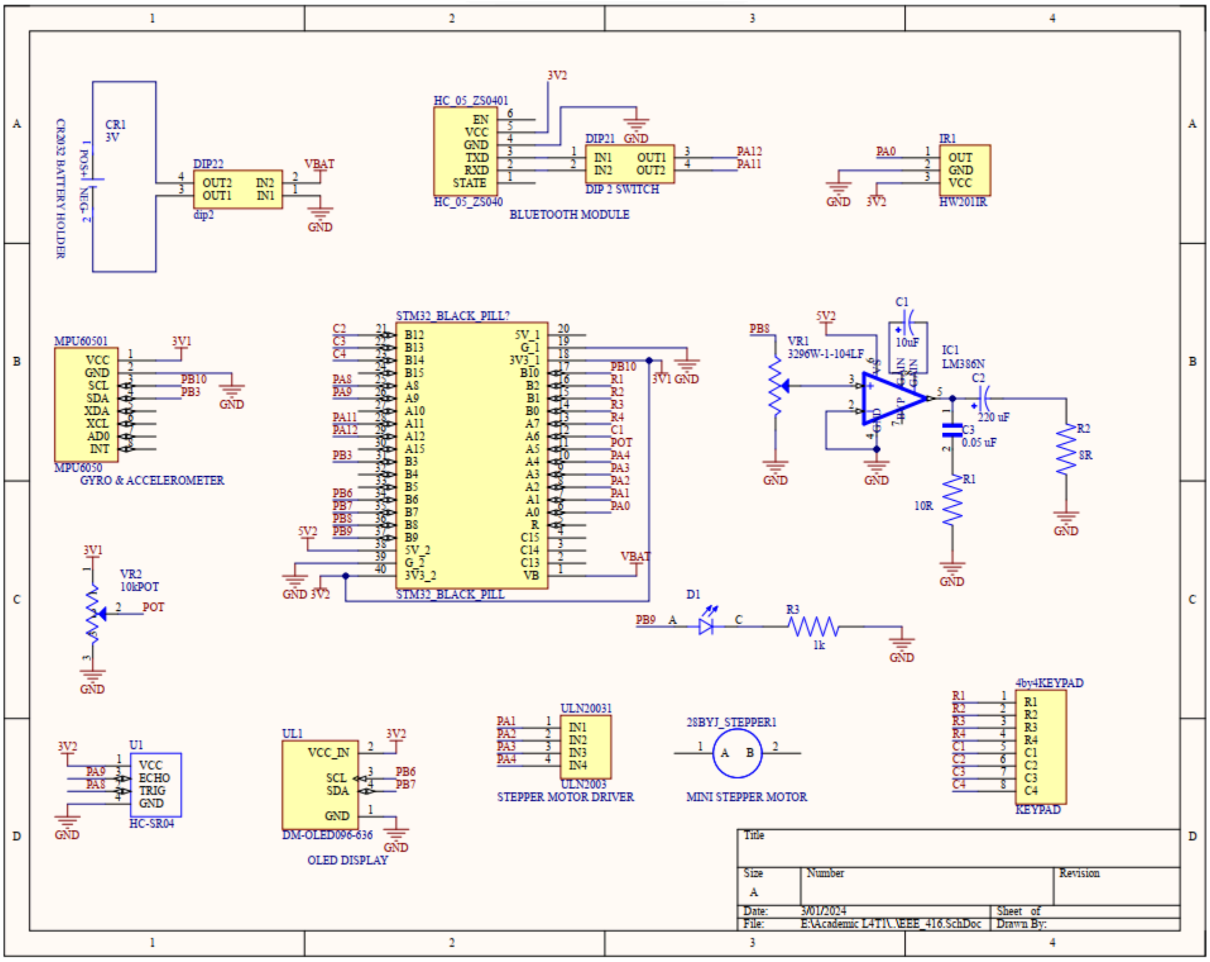
1. STM32 Black Pill Board that has minimum area on the PCB
2. It should encompass all experiments of EEE 416
3. Additionally, it should contain a LCD display with SPI, Gyroscopic sensor with I2C etc.
4. The Black Pill should be detachable from the PCB.
5. The board should have a separate power supply unit for the peripheral devices.

3.2 Design Method (PO(a))

The following steps were followed for implementing the project:

- Familiarize with STM32F401CDU6 Microcontroller
- Familiarize with STM32 Black Pill & CubeIDE
- Buying Hardware
- Coding & Debugging with STM32 Black Pill
- PCB Design
- PCB Manufacturing
- Testing Final Circuit

3.3 Circuit Diagram



3.4 PCB Design

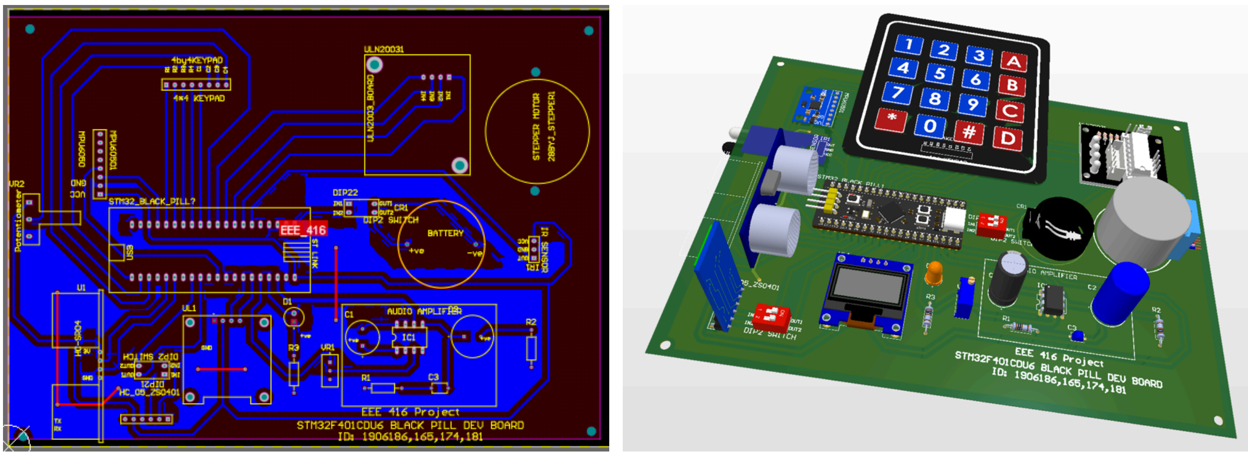


Fig: PCB Design & 3D Rendering

4 Implementation

4.1 Stepper Motor:

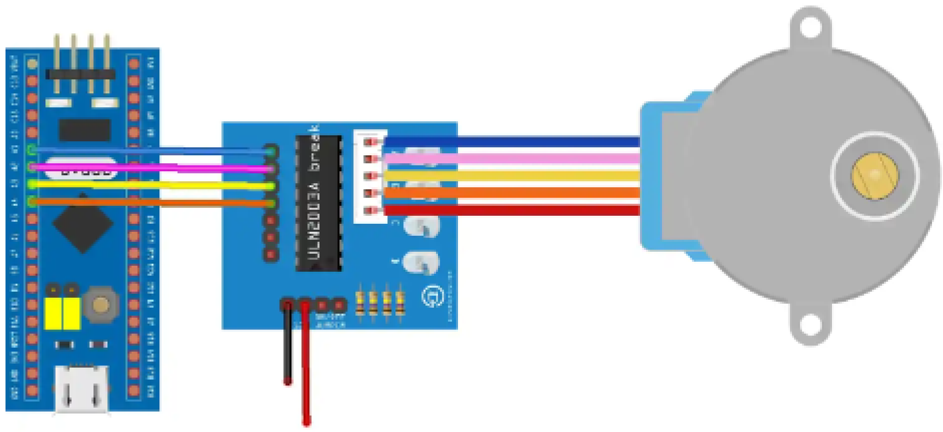


Fig: Connection of the stepper motor with microprocessor

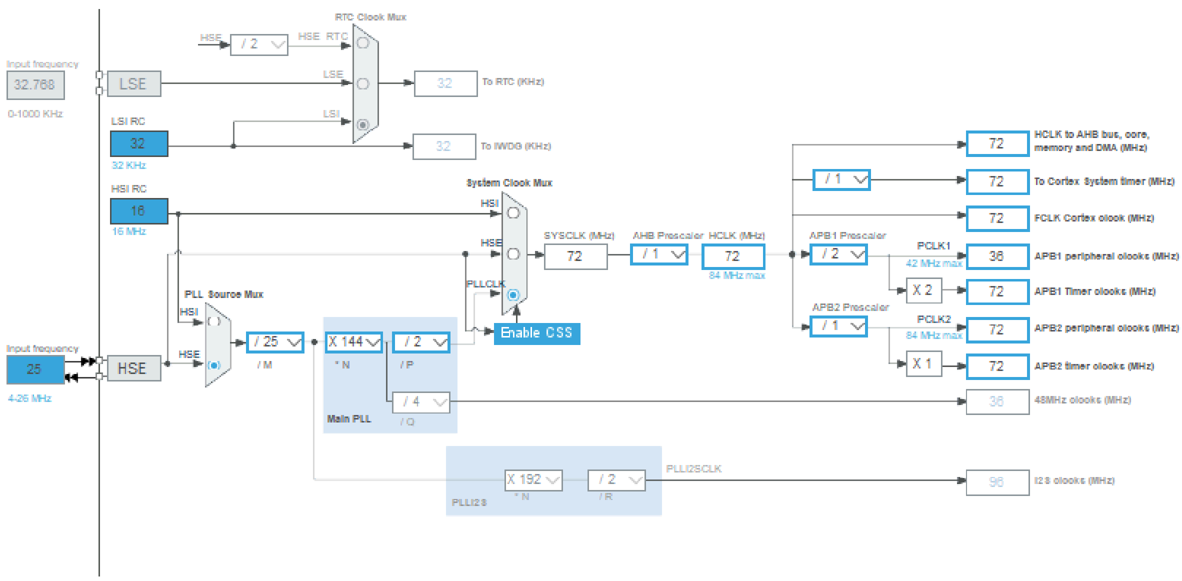


Fig: Clock configuration of the stepper motor

4.1.1 Implementation Code:

```c
/* USER CODE BEGIN 0 */
void delay (uint16_t us)
{
  __HAL_TIM_SET_COUNTER(&htim1, 0);
  while (__HAL_TIM_GET_COUNTER(&htim1) < us);
}

#define stepsperrev 4096

void stepper_set_rpm (int rpm)  // Set rpm--> max 13, min 1,,,  went to 14 rev/min
{
    delay(60000000/stepsperrev/rpm);
}

void stepper_half_drive (int step)
{
    switch (step){
        case 0:
              HAL_GPIO_WritePin(GPIOA, GPIO_PIN_1, GPIO_PIN_SET);   // IN1
              HAL_GPIO_WritePin(GPIOA, GPIO_PIN_2, GPIO_PIN_RESET);   // IN2
              HAL_GPIO_WritePin(GPIOA, GPIO_PIN_3, GPIO_PIN_RESET);   // IN3
              HAL_GPIO_WritePin(GPIOA, GPIO_PIN_4, GPIO_PIN_RESET);   // IN4
              break;

        case 1:
              HAL_GPIO_WritePin(GPIOA, GPIO_PIN_1, GPIO_PIN_SET);   // IN1
              HAL_GPIO_WritePin(GPIOA, GPIO_PIN_2, GPIO_PIN_SET);   // IN2
              HAL_GPIO_WritePin(GPIOA, GPIO_PIN_3, GPIO_PIN_RESET);   // IN3
              HAL_GPIO_WritePin(GPIOA, GPIO_PIN_4, GPIO_PIN_RESET);   // IN4
              break;

```

```c
  /* Infinite loop */
  /* USER CODE BEGIN WHILE */
  while (1)
  {
    /* USER CODE END WHILE */
      for(int i=0;i<512;i++)
      {
        for(int i=0;i<8;i++)
        {
            stepper_half_drive (i);
            stepper_set_rpm (8);

            }
        }

```

4.2 Gyroscope:

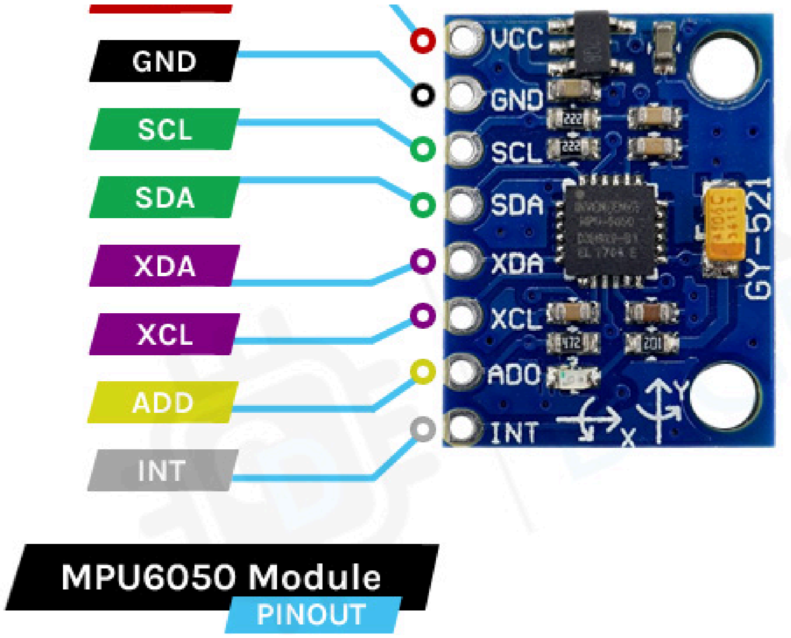


Fig: Gyro MPU6050 module

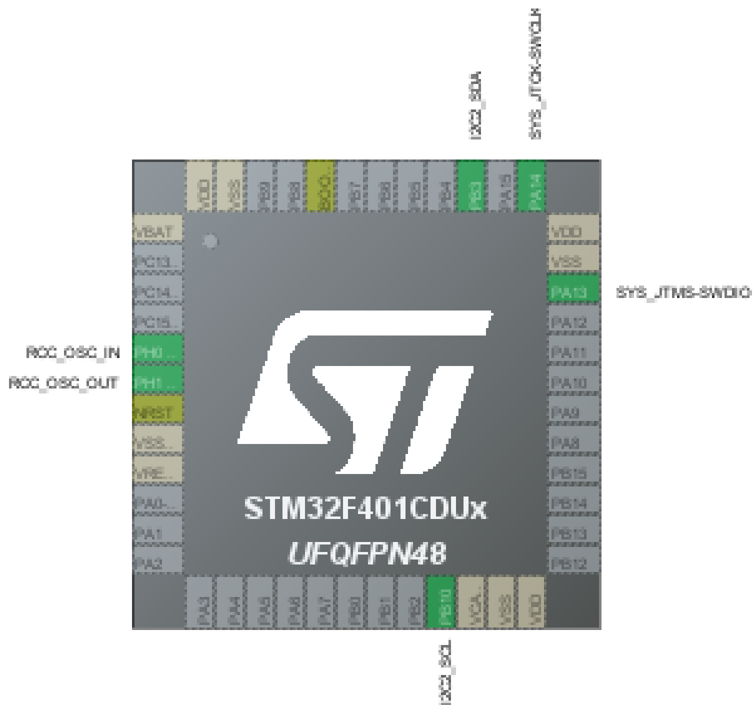


Fig: Pin configuration for the gyroscope

4.2.1 Implementation Code:

```c
void MPU6050_Read_Gyro (void)
{
    uint8_t Rec_Data[6];

    // Read 6 BYTES of data starting from GYRO_XOUT_H register

    HAL_I2C_Mem_Read (&hi2c2, MPU6050_ADDR, GYRO_XOUT_H_REG, 1, Rec_Data, 6, 1000);

    Gyro_X_RAW = (int16_t)(Rec_Data[0] << 8 | Rec_Data [1]);
    Gyro_Y_RAW = (int16_t)(Rec_Data[2] << 8 | Rec_Data [3]);
    Gyro_Z_RAW = (int16_t)(Rec_Data[4] << 8 | Rec_Data [5]);

    /*** convert the RAW values into dps (°/s)
         we have to divide according to the Full scale value set in FS_SEL
         I have configured FS_SEL = 0. So I am dividing by 131.0
         for more details check GYRO_CONFIG Register              ****/

    Gx = Gyro_X_RAW/131.0;
    Gy = Gyro_Y_RAW/131.0;
    Gz = Gyro_Z_RAW/131.0;
}
```

4.3 Ultrasonic Sonar Sensor And Oled:

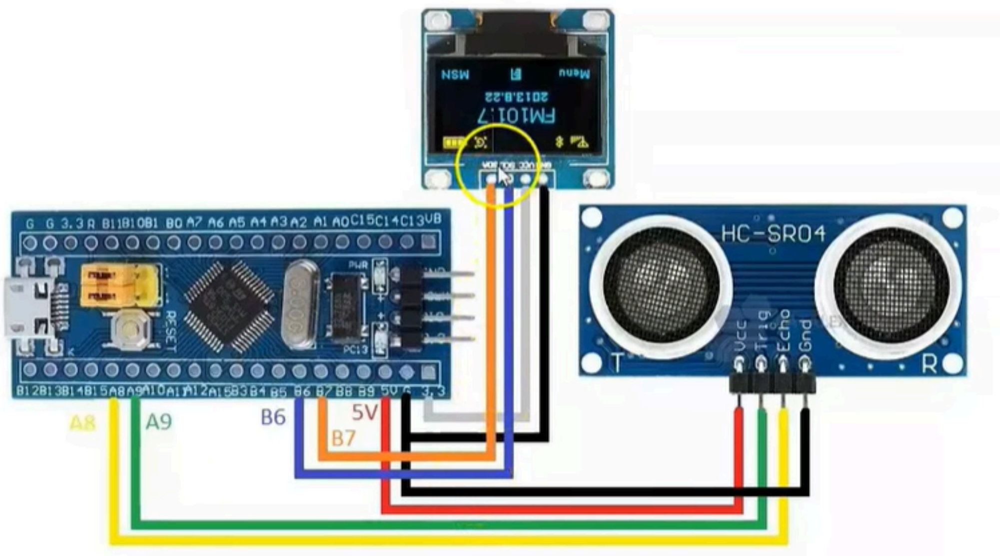
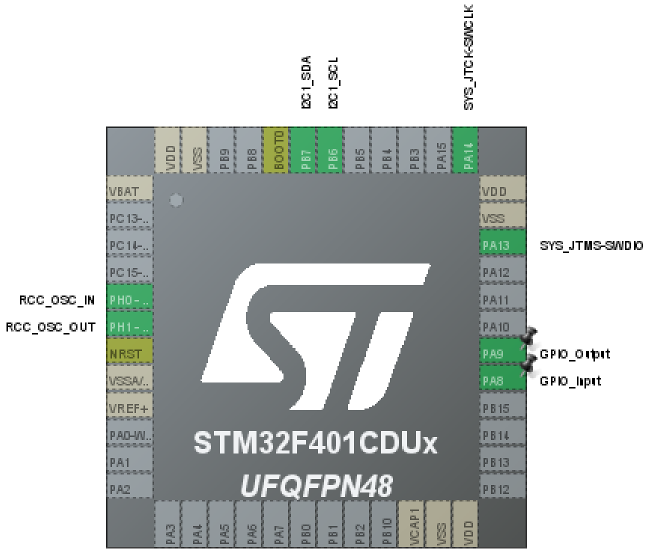


Fig:Stm32 Blackpill with Sonar Sensor and Oled Display



4.3.1 Implementation code:

```c
#include "main.h"

/* Private includes ----------------------------
/* USER CODE BEGIN Includes */
#include "fonts.h"
#include "ssd1306.h"
#include "stdio.h"
/* USER CODE END Includes */

```

```c
#define TRIG_PIN GPIO_PIN_8
#define TRIG_PORT GPIOA
#define ECHO_PIN GPIO_PIN_9
#define ECHO_PORT GPIOA
uint32_t pMillis;
uint32_t Value1 = 0;
uint32_t Value2 = 0;
uint16_t Distance  = 0;  // cm
char strCopy[15];
char str[2] = "  ";
/* USER CODE END PV */

/* Private function prototypes ---
void SystemClock_Config(void);
static void MX_GPIO_Init(void);
static void MX_I2C1_Init(void);
static void MX_TIM1_Init(void);
/* USER CODE BEGIN PFP */
```

```
while (1)
    //SSD1306_Puts ("Distance:", &Font_11x18, 1);
{
    HAL_GPIO_WritePin(TRIG_PORT, TRIG_PIN, GPIO_PIN_SET);  // pull the TRIG pin HIGH
        __HAL_TIM_SET_COUNTER(&htim1, 0);
        while (__HAL_TIM_GET_COUNTER (&htim1) < 10);  // wait for 10 us
        HAL_GPIO_WritePin(TRIG_PORT, TRIG_PIN, GPIO_PIN_RESET);  // pull the TRIG pin low

        pMillis = HAL_GetTick(); // used this to avoid infinite while loop  (for timeout)
        // wait for the echo pin to go high
        while (!(HAL_GPIO_ReadPin (ECHO_PORT, ECHO_PIN)) && pMillis + 10 >  HAL_GetTick());
        Value1 = __HAL_TIM_GET_COUNTER (&htim1);

        pMillis = HAL_GetTick(); // used this to avoid infinite while loop (for timeout)
        // wait for the echo pin to go low
        while ((HAL_GPIO_ReadPin (ECHO_PORT, ECHO_PIN)) && pMillis + 50 > HAL_GetTick());
        Value2 = __HAL_TIM_GET_COUNTER (&htim1);

        Distance = (Value2-Value1)* 0.034/2;

        SSD1306_GotoXY (0, 0);
        SSD1306_Puts ("Distance:", &Font_11x18, 1);
        sprintf(strCopy,"%d    ", Distance);
        itoa(Distance, strCopy,10);
        SSD1306_GotoXY (0, 30);
        SSD1306_Puts ("   ", &Font_16x26, 1);
        SSD1306_Puts (strCopy, &Font_16x26, 1);
        SSD1306_Puts ("   ", &Font_16x26, 1);
        SSD1306_UpdateScreen();
        HAL_Delay(50);
        //SSD1306_Clear();
```

4.4 Bluetooth Module:

In our project, we used Bluetooth HC-05 module. Bluetooth serial modules allow all serial enabled devices to communicate with each other using Bluetooth. It has 6 pins. HC-05 module has two modes. Command mode and data mode, Data mode and command mode. We implemented data mode here. The baud rate we set is 9600 bps.

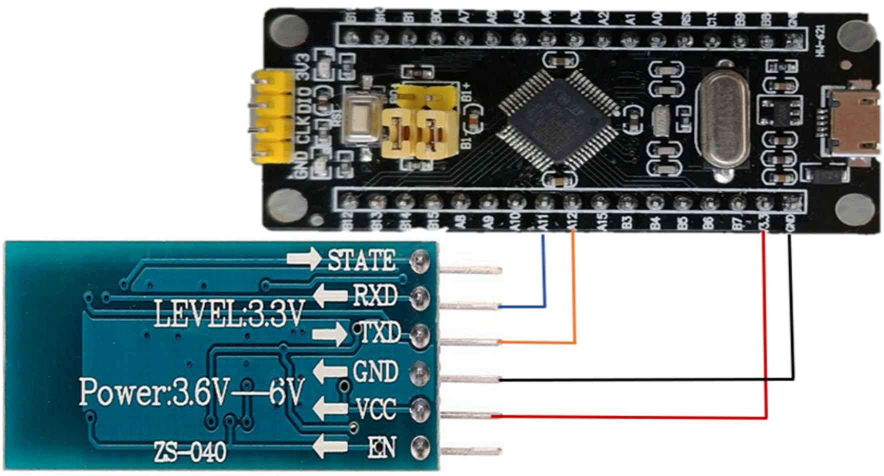


Figure: Connection of Bluetooth HC-05 with STM32 Black Pill

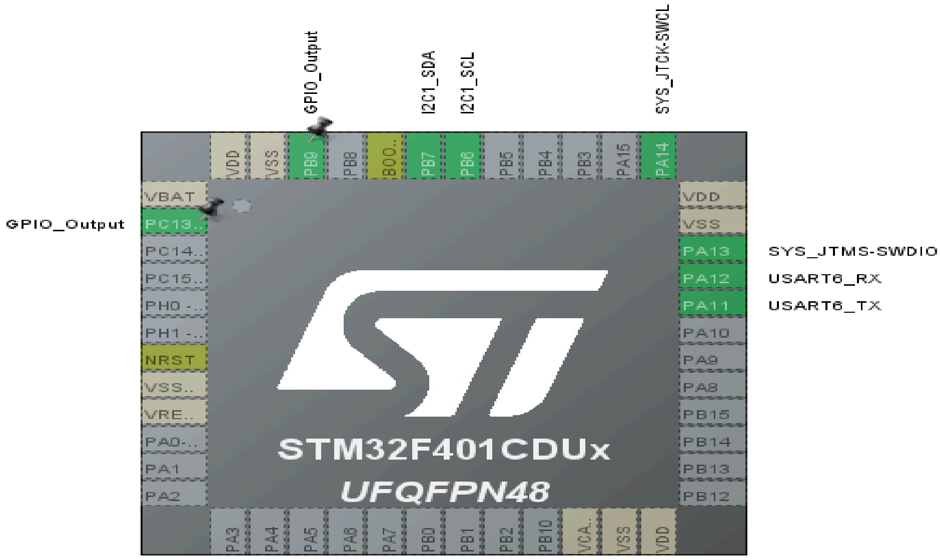


Fig: Pin configuration for the bluetooth module

4.4.1 Implementation Code:

```
/* USER CODE BEGIN PV */
uint8_t rxData;
/* USER CODE END PV */
```

```
  /* USER CODE BEGIN 2 */
  HAL_UART_Receive_IT(&huart6,&rxData,1);
  SSD1306_Init();
    char snum[5];

    SSD1306_GotoXY (0,0);
    SSD1306_Puts ("Ridwanul", &Font_11x18, 1);
    SSD1306_GotoXY (0, 30);
    SSD1306_Puts ("Haque", &Font_11x18, 1);
    SSD1306_UpdateScreen();
    HAL_Delay (1000);

    SSD1306_ScrollRight(0,7);
    HAL_Delay(3000);
    SSD1306_ScrollLeft(0,7);
    HAL_Delay(3000);
    SSD1306_Stopscroll();
    SSD1306_Clear();
    SSD1306_GotoXY (35,0);
    SSD1306_Puts ("SCORE", &Font_11x18, 1);
  /* USER CODE END 2 */
```

```c
/* USER CODE BEGIN 4 */
void HAL_UART_RxCpltCallback(UART_HandleTypeDef *huart)
{
  if(huart->Instance==USART6)
  {
    if(rxData==78) // Ascii value of 'N' is 78 (N for NO)
    {
        HAL_GPIO_WritePin(GPIOC, GPIO_PIN_13, 1);
        HAL_GPIO_WritePin(GPIOB, GPIO_PIN_9, 0);
    }
    else if (rxData==89) // Ascii value of 'Y' is 89 (Y for YES)
    {
        HAL_GPIO_WritePin(GPIOC, GPIO_PIN_13, 0);
        HAL_GPIO_WritePin(GPIOB, GPIO_PIN_9, 1);
    }
    HAL_UART_Receive_IT(&huart6,&rxData,1); // Enabling interrupt receive again
  }
}
/* USER CODE END 4 */
```

4.5 IR Sensor Module

We used HW201 IR sensor module. This module has 3 pins. Power supply input pin VCC, Power supply ground pin GND and active high output pin named OUT. Besides it has a IR transmitter LED, IR receiver photodiode, distance adjusting knob, power LED and obstacle LED.

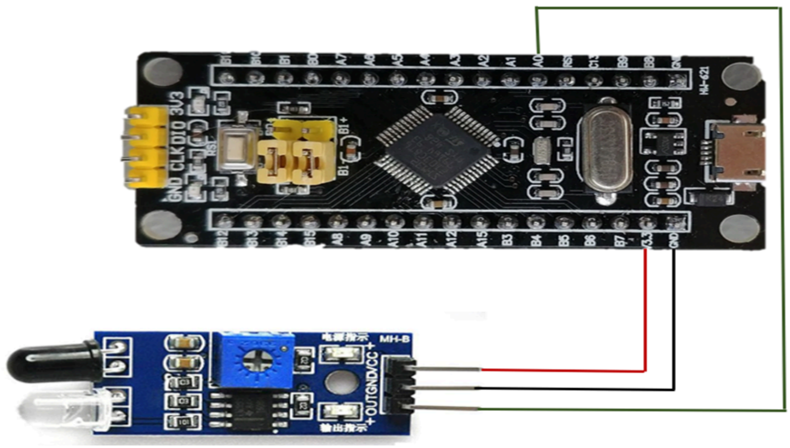


Figure: Connection of HW-05 IR sensor module with STM32 Black Pill

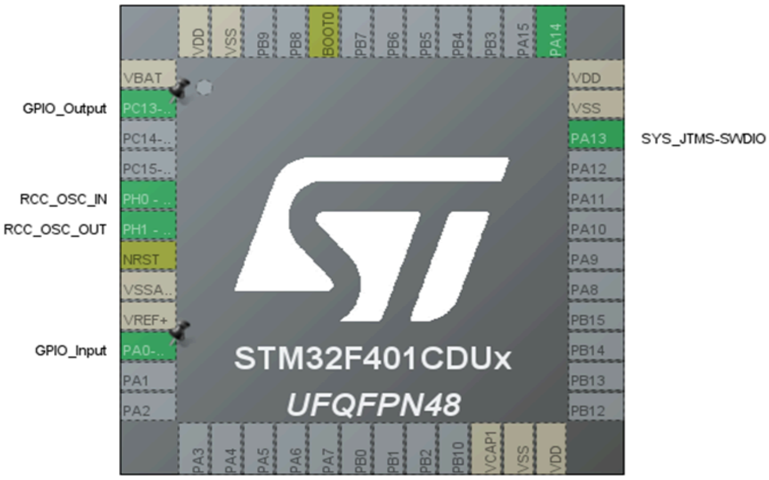


Fig: Pin configuration for the IR sensor module

4.5.1 Implementation Code:

```c
#include "main.h"

int main(void)
{
  /* USER CODE BEGIN 1 */

  /* USER CODE END 1 */

  /* MCU Configuration----------------------------------

  /* Reset of all peripherals, Initializes the Flash int
  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();
```

```
  while (1)
  {
    /* USER CODE END WHILE */

    /* USER CODE BEGIN 3 */
      int x = HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_0);
         HAL_Delay(300);
         if ((x == 0)) {
           HAL_GPIO_WritePin(GPIOC, GPIO_PIN_13, 0);
         } else {
           HAL_GPIO_WritePin(GPIOC, GPIO_PIN_13, 1);
         }

   }
  /* USER CODE END 3 */
}
```

4.6 Speaker:

4.6.1 Audio Amplifier:

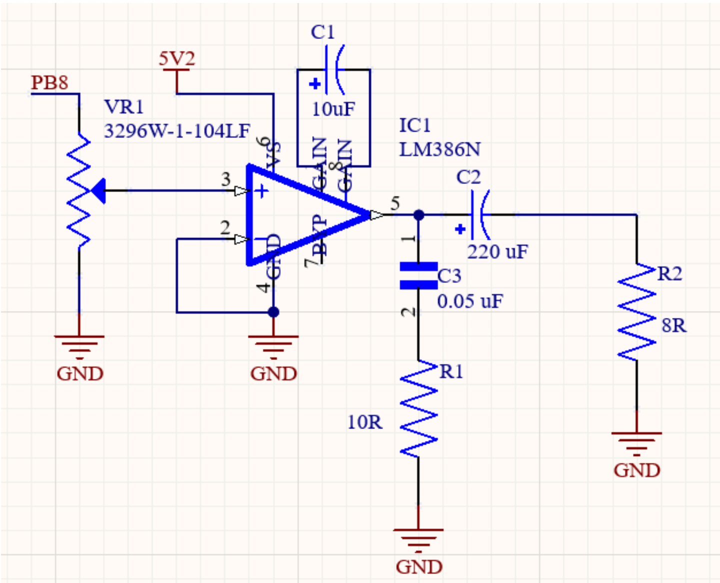


Fig: Audio Amplifier circuit for Speaker

4.6.2 Implementation Code:

```
HAL_TIM_PWM_Start(&htim4, TIM_CHANNEL_3);


int i;
int n = 1;
uint16_t current_note = 0;




static uint32_t note_freq[8] = {261,  294, 329, 349, 392, 440, 494, 522}; //Hz
static uint16_t song_notes[32]= {2, 2, 3, 2, 4, 4, 4, 4,
                                  1, 1, 2, 1, 3, 3, 3, 3,
                                  2, 2, 2, 2, 1, 1, 1, 1,
                                  0, 0, 0, 0, 0, 0, 0, 0};  // 0=C, 1=D ....

```

```
TIM4 ->CCR1 = 500;
TIM4->ARR = (16000000 / 10 /  freq_giveup[current_note]);

/* USER CODE END 2 */

/* Infinite loop */
/* USER CODE BEGIN WHILE */
while (1)
{
    //TIM4->ARR = (16000000UL/10/ note_freq[current_note]) - 1UL;

    TIM4->ARR = (16000000UL/10/ note_freq[song_notes[current_note]]) - 1UL;
    freq = TIM4->ARR;
    current_note = current_note+1;
    if (current_note > 27 ||  current_note < 0) current_note = 0; // means,song_notes matrix
   HAL_Delay(60);         // delay
  /* USER CODE END WHILE */

  /* USER CODE BEGIN 3 */
}
```

5 Design Analysis and Evaluation

5.1 Novelty

Our model is Blackpill that was launched a year ago. So, there is almost no resource for this microcontroller, like codes, websites and videos. We used the resources available for bluepill and then wrote our code, set the clock and other library functions. Nobody in the EEE 416 lab currently working with blackpill except us. Our projects entirely focus on the lab exercises. We have built a compact system for the lab exercises. It includes an ultrasonic sonar sensor and shows the data on the OLED display. It also includes speaker, stepper motor, infrared sensor, bluetooth sensor. The PCB has made the project look much more compact, less messy and easy to understand. PCB has also made the project look simple and gorgeous.

5.2 Design Considerations (PO(c))

5.2.1 Considerations to public health and safety

The model is basically designed with stm32f401cdu blackpill and other relevant sensors. The components or the system has no effect on the public health and doesn't involve any safety concern.

5.2.2 Considerations to environment

Environmental safety was given priority during the development of the product and the final product and the whole procedure had no adverse effect on the environment. The components can be recycled after being damaged.

- Very few electronic waste
- Very few safety hazards

5.2.3 Considerations to cultural and societal needs

Cultural and societal concerns were taken into consideration. The model was developed to support the societal and cultural aspects accordingly.

- User friendly device for all age and culture
- No personal life breach and No overpriced elements used

5.3 Investigations (PO(d))

5.3.1 Design of Experiment

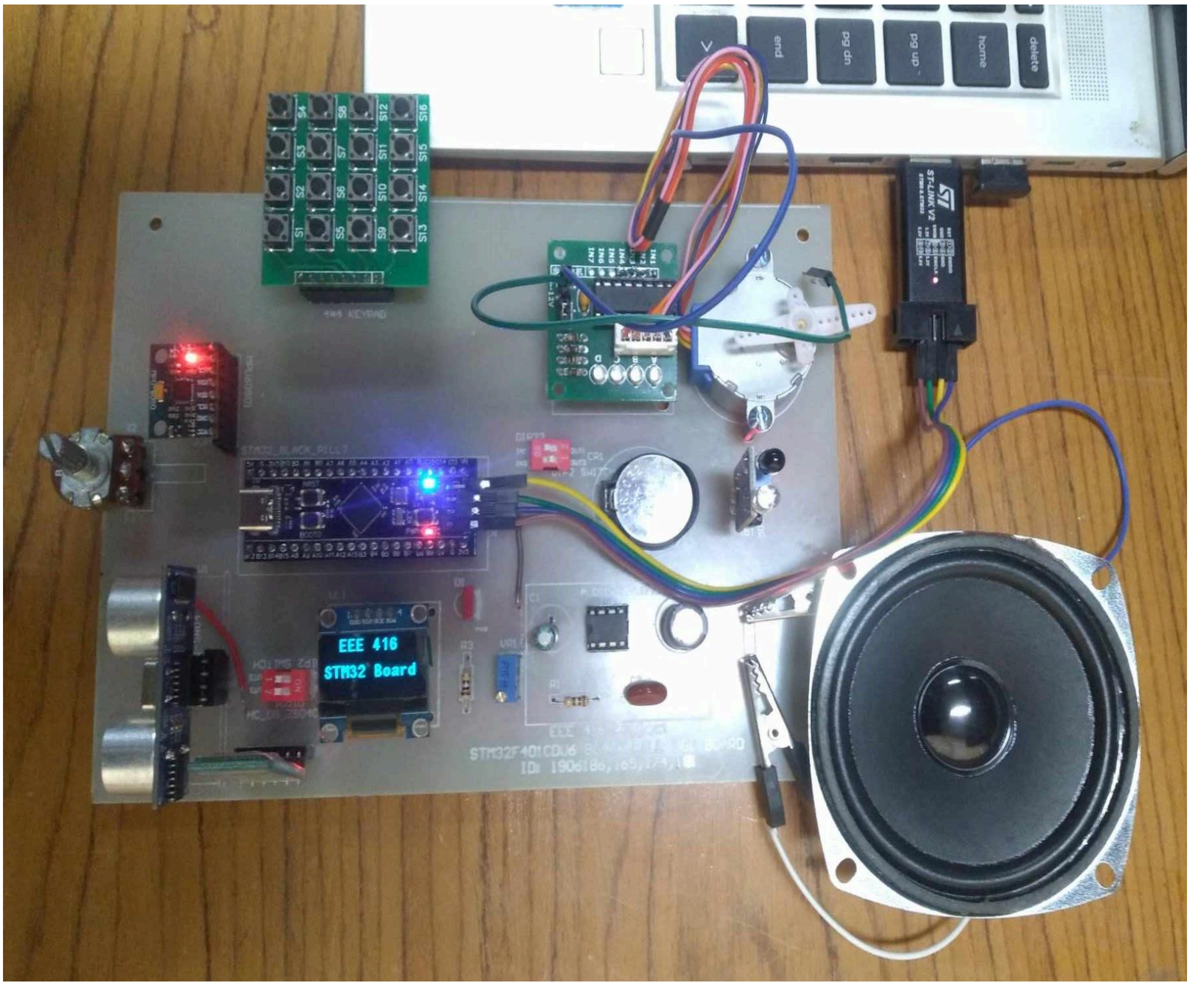


Fig: PCB

5.3.2 Data Collection

Data Collection was not the primary focus of our experiment.

5.3.3 Results and Analysis

We have showed the distance measured by the ultrasonic sensor on the oled display. Then we have configured the IR sensor and detected barrier using it and turn on the LED light on the black pill. We have also coded the bluetooth sensor and toggled light using the switch on our phone. We further configured the stepper motor in half stepping mode. The speaker was not giving us proper output. Rather it was giving us noise. ADC was also installed on the PCB.

5.3.4 Interpretation and Conclusions on Data

Data collection and interpretation was not our primary concern.

5.4 Limitations of Tools (PO(e))

- We can not add components to the PCB as the board has been fixed. We need to change PCB design everytime we want to integrate a component.
- We can not power up the board entirely on blackpill. We need extra power source.
- SPI is not a good system for blackpill as it needs 12 pins. Other peripheral devices also need RX,TX, PWM pins.

5.5 Impact Assessment (PO(f))

5.5.1 Assessment of Societal and Cultural Issues

The project isn’t against any social point of view and morality.

5.5.2 Assessment of Health and Safety Issues

This project doesn’t create any hazard waste or any form of danger . It is needed to ensure the disposal of our component. So we were able to ensure health and safety issues.

5.5.3 Assessment of Legal Issues

Everything used here permitted by govt. of Bangladesh . So the overall project don’t have any legal issues.

5.6 Sustainability Evaluation (PO(g))

Most components are stable and very much reliable. So it is expected that it will sustain more than year. Also it is suggested to check our setup every month to ensure optimization service quality.

5.7 Ethical Issues (PO(h))

The project work was solely done by us and we have mentioned the article from where we took help in the reference section. We have completed our project with moral standard and we abide by all the rules mention by regulation committee. All the softwares and files used are free.

6 Reflection on Individual and Team work (PO(i))

6.1 Individual Contribution of Each Member

- ID 1906186 - Led Blinking, Speaker, PCB Layout
- ID 1906181 - Stepper Motor, Gyroscope (I2C & SPI)
- ID 1906165 - Bluetooth, IR
- ID 1906174 - Sonar Sensor, Display LCD & LED

6.2 Mode of TeamWork

- Overall planning through a meeting
- Individual task assignment
- Work closely together, sharing ideas, responsibilities, and tasks
- Helping each other on technical problems
- Prioritizing transparency, accountability, and mutual respect
- Ensuring that everyone's contributions are valued and acknowledged
- Motivating one another
- Trying to maximize productivity, creativity, and overall project success.
- Integrating the individual tasks and completing the project

6.3 Diversity Statement of Team

As a team, we believe that diversity fuels our innovation and drives our success. We recognize that diversity encompasses a broad spectrum of identities, including but not limited to race, ethnicity, gender, sexual orientation, age, religion, disability, and socio-economic background. We value and celebrate the unique talents, perspectives, and contributions of each team member, understanding that diversity strengthens our team and enhances our ability to innovate.

6.4 Log Book of Project Implementation

Week	Task	Assessment
3rd	Collecting Ideas and searching necessary sources for proposal	
4th	Finalize idea of project	
5th	Study about Stm32 board & STM32 Cube IDE	
6th	Learn how to do basic code using STM32 Cube IDE	
7th	Making Lists of Components and distributing tasks among the group members	Proposal Presentation
8th	Performing Individual Tasks	
9th	Performing Individual Tasks	
10th	Discussing about problems faced during individual tasks and trying to solve them	
11th	Later work on individual tasks	
12th	Preparing for presentation, making hardware demonstration individually	Progress Presentation
13th	Work on PCB designing, preparing for final presentation	Final Presentation
14th	Assembling individual works together and get the PCB	Final Demonstration

7 Communication (PO(j))

7.1 Executive Summary

This trainer board can be used in the Laboratory for the learning of STM32 Microcontrollers and different Peripheral devices..

7.2 User Manual

- Open the STM32 CubeIDE Project files.
- Upload the code using ST Link V2 Debugger.
- Now you can disconnect the Debugger or you can keep powering the board using ST link. You can also use a USB or battery to power up the board instead.
- However, don't power up with the two devices (battery, USB or ST link) at the same time.

7.3 Github Link

https://github.com/RaihanAminRana/EEE_416_Project/

7.4 YouTube Link

https://youtu.be/1EUnkGZ2P24?si=km661cYm7T4fIZxc

8 Project Management and Cost Analysis (PO(k))

8.1 Bill of Materials

Product	Unit Price	Number of component	Total Price
STM32 Blackpill	1500	1	1500
Sonar sensor HC SR04	90	1	90
Oled Display 0.96inch	400	1	400
Bluetooth Sensor HC05	350	1	350
Stepper Motor	200	1	200
Gyroscope	750	1	750
Jumper Wire	40	1	40
IR Sensor HW 201	90	1	90
PCB Implementation	2000	1	2000
	Total		5420

9 Future Work (PO(l))

- Increasing the features and improving functionalities.
- Our Project encompasses almost all the tasks covered in the microprocessor lab. This project can be used as the prototype of various controlling applications like monitoring systems, IoT environmental monitoring stations, wearable health monitoring devices, etc.
- Analyze the performance of this project for a sufficient time duration and find out the areas of improvement.
- Run many test cases to find out issues that would be solved.
- Try to ensure safer and more efficient power supply.
- Improve mobility and adaptability more.
- Make our project more user friendly

10 References

1. https://controllerstech.com/interface-stepper-motor-with-stm32/
2. https://stm32-base.org/boards/STM32F103C8T6-Black-Pill.html
3. https://www.mouser.com/pdfDocs/ProductOverview_DFRobot-DFRobot-STM32F411-BlackPill-Development-Board.pdf