

**ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH  
TRƯỜNG ĐẠI HỌC BÁCH KHOA  
KHOA ĐIỆN – ĐIỆN TỬ**



**MÔN ĐO LƯỜNG ĐIỀU KHIỂN BẰNG MÁY TÍNH (EE3017)**

---

Đề tài Bài tập lớn:

**LẬP TRÌNH CHO KIT STM32 GIAO TIẾP VỚI MÁY TÍNH**

---

Giáo viên hướng dẫn: **Nguyễn Hoàng Giáp**  
Sinh viên thực hiện : **Nguyễn Gia Phúc**  
MSSV : **2010528**

*TP. Hồ Chí Minh, Ngày 7 Tháng 8 Năm 2023*

## MỤC LỤC

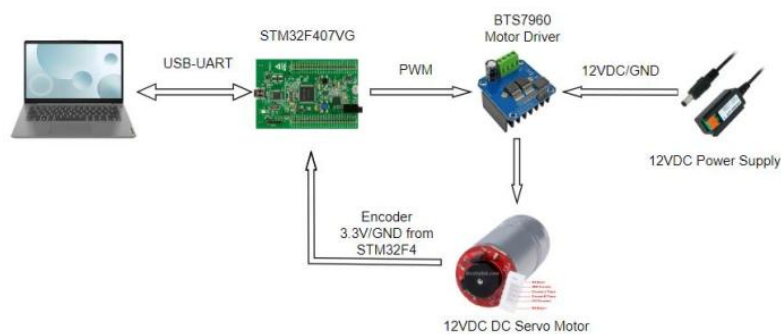
<b>CHƯƠNG 1: GIỚI THIỆU TỔNG QUAN VỀ ĐỀ TÀI.....</b>	<b>2</b>
1.1. Giới thiệu về phần cứng. ....	2
1.2. Giới thiệu về phần mềm lập trình cho MCU STM32 và PC. ....	2
<b>CHƯƠNG 2: LẬP TRÌNH FIRRMWARE TRÊN MCU STM32F407 .....</b>	<b>4</b>
2.1. Lập trình thuật toán PID điều khiển động cơ. ....	4
2.1.1. Bộ PWM. ....	4
2.1.2. Đọc tốc độ qua bộ Encoder. ....	6
2.1.3. Giải thuật PID rời rạc. ....	7
2.3. Lập trình sáng LED. ....	7
2.4. Lập trình nút nhấn. ....	9
2.5. Lập trình giao tiếp nối tiếp RS232. ....	10
<b>CHƯƠNG 3: LẬP TRÌNH GUI TRÊN C# .....</b>	<b>14</b>
<b>CHƯƠNG 4: KẾT QUẢ THỰC HIỆN .....</b>	<b>15</b>

## CHƯƠNG 1: GIỚI THIỆU TỔNG QUAN VỀ ĐỀ TÀI

### 1.1. Giới thiệu về phần cứng.



*Sơ đồ thực tế về hệ thống*



*Sơ đồ tổng quan về hệ thống*

Pin	Chức năng	Thiết bị kết nối
PA0	Đọc Encoder	Encoder phase A/C1
PA1	Đọc Encoder	Encoder phase B/C2
PA2	STM32 UART TX	PC UART RX
PA3	STM32 UART RX	PC UART TX
PA8	Xuất xung PWM	Enable Driver
PC6	Digital Output	Ngõ vào thứ nhất Driver/IN3
PC7	Digital Output	Ngõ vào thứ hai Driver/IN4
PB0	Ngắt ngoài (Digital Input)	Nút nhấn
PD12	Digital Output	Led
PD13	Digital Output	Led
PD14	Digital Output	Led
PD15	Digital Output	Led

*Chức năng Pin được sử dụng trong STM32F4*

### 1.2. Giới thiệu về phần mềm lập trình cho MCU STM32 và PC.

STT	IDE + Compiler
-----	----------------

1	<b>STM32CubeMX</b>
	Package for STM32F4
2	<b>MDK-AKM Professional</b>
	Package STM32F4xx_DFP

*Các phần mềm lập trình cho MCU STM32F4*

**Note:** Toàn bộ source code lưu trong Folder: DCMotorController với STM32CubeMX.ioc project của STM32CubeMX và STM32CubeMX.uvprojx là của MDK-ARM

STT	IDE + Compiler
1	Microsoft Visual Studio Enterprise
	Visual C#

*Các phần mềm lập trình GUI trên PC*

**Note:** Toàn bộ source code lưu trong Folder WindowsFormsApplication1 với project là: WindowsFormsApplication1.sln

## CHƯƠNG 2: LẬP TRÌNH FIRMWARE TRÊN MCU STM32F407

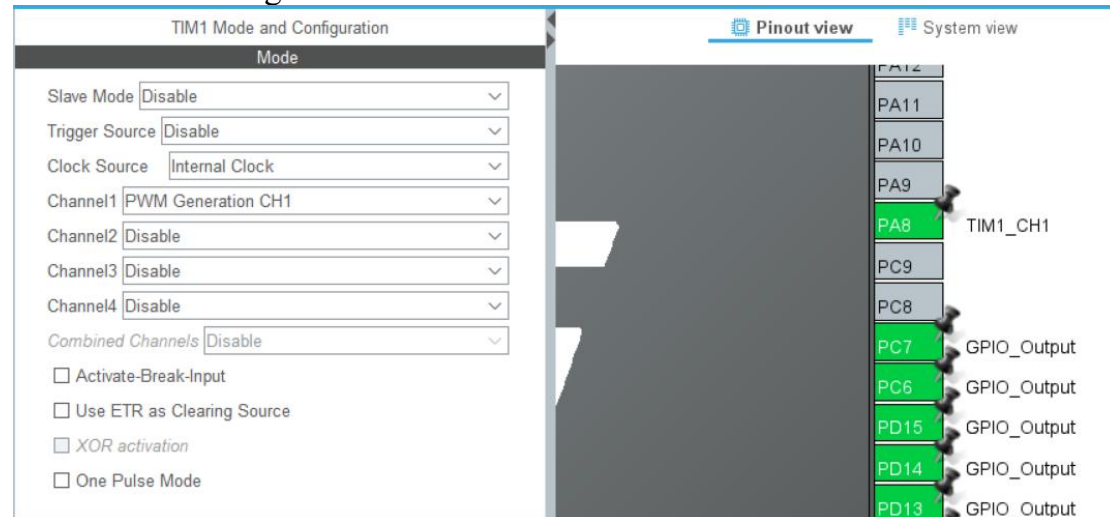
### 2.1. Lập trình thuật toán PID điều khiển động cơ.

#### 2.1.1. Bộ PWM.

Chọn Timer 1 cấu hình như sau:

Clock source: internal clock

Channel 1: PWM generation CH1



Từ công thức:

$$UpdateEvent = \frac{Timer_{clock}}{(Prescaler + 1)(Period + 1)}$$

Trong mục configuration

Prescaler: 159 => mỗi lần đếm là 10μs

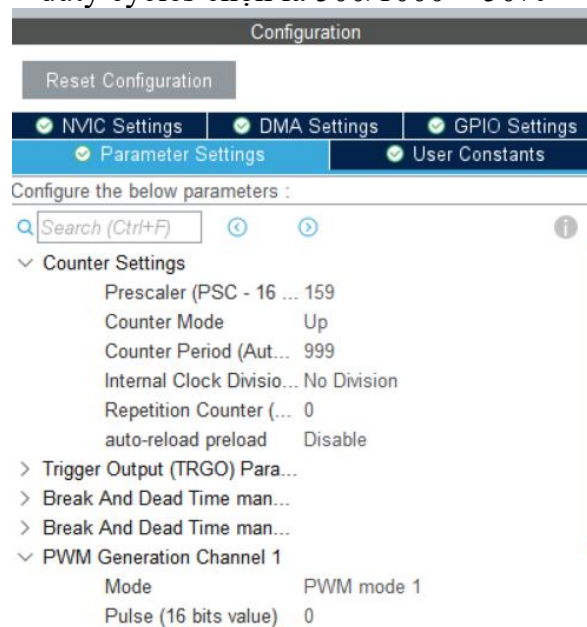
Counter Mode: Up

Counter Period: 999 => ở đây mình tạo chu kì xung là 10ms

Mode: PWM mode 1

Pulse: 0 => duty cycles chọn là 0/1000 = 0%

Note: Pulse: 499 => duty cycles chọn là 500/1000 = 50%



## Các hàm quan trọng

### - Hàm khởi động PWM:

```
HAL_TIM_PWM_Start(&htim1, TIM_CHANNEL_1);
```

Tham số:

- *htim*: Bộ timer sử dụng chức năng PWM, nếu sử dụng Timer 1 thì sẽ là &htim1,...
- *Channel*: Channel đã cấu hình cho PWM, nếu là Channel 1 thì gọi TIM\_CHANNEL\_1

### - Hàm xuất xung PWM:

```
HAL_TIM_SET_COMPARE(&htim1, TIM_CHANNEL_1, duty);
```

Tham số: · *htim*: Bộ timer sử dụng chức năng PWM, nếu sử dụng Timer 1 thì sẽ là &htim1, ... · *Channel*: Channel đã cấu hình cho PWM, nếu là Channel 1 thì gọi TIM\_CHANNEL\_1

- *Duty*: Duty Cycle muốn xuất ra, ví dụ 0%, 50%, 100%

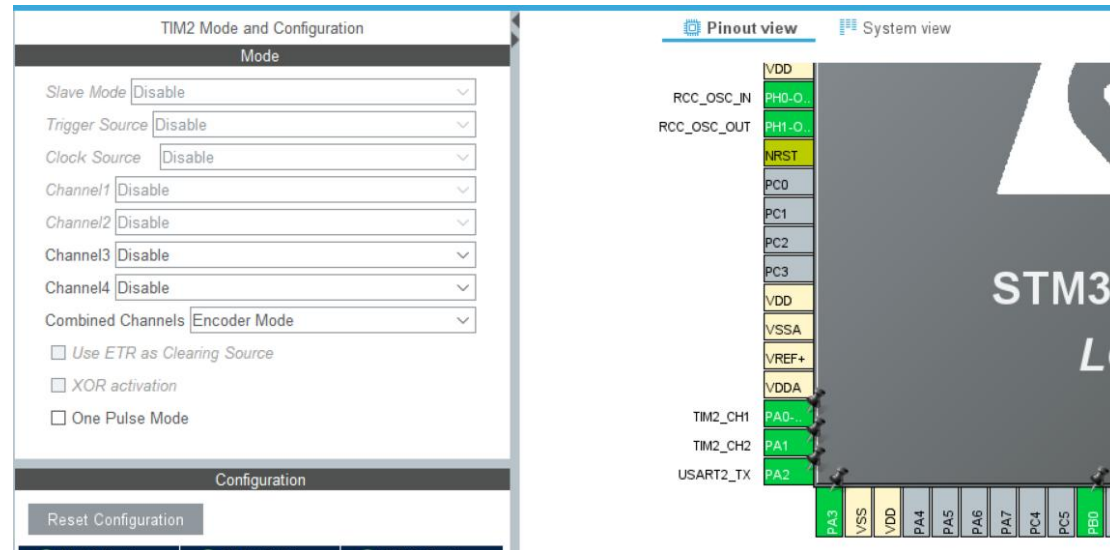
### Áp dụng vào project:

```
if (!state)
{
    duty=PID_control (speed_set, speed);
    if(direct)
    {
        HAL_GPIO_WritePin(GPIOC, GPIO_PIN_6,GPIO_PIN_RESET);
        HAL_GPIO_WritePin(GPIOC, GPIO_PIN_7,GPIO_PIN_SET);
    }
    else
    {
        HAL_GPIO_WritePin(GPIOC, GPIO_PIN_7,GPIO_PIN_RESET);
        HAL_GPIO_WritePin(GPIOC, GPIO_PIN_6,GPIO_PIN_SET);
    }
    __HAL_TIM_SET_COMPARE(&htim1, TIM_CHANNEL_1, duty);
}
else
{
    __HAL_TIM_SET_COMPARE(&htim1, TIM_CHANNEL_1, 0);
}
```

### 2.1.2. Đọc tốc độ qua bộ Encoder.

Chọn Timer 2 cấu hình như sau:

Combined Channels: Encoder Mode.



CubeMX sẽ tự config chân PA0 VÀ PA1 ứng với CH1 và CH2 của TIMER2 để đọc encoder mode 4

#### Các hàm quan trọng

- Hàm khởi động Encoder Mode của Timer 2:

```
HAL_TIM_Encoder_Start(&htim2,TIM_CHANNEL_ALL);
```

- Áp dụng vào project:

```
void HAL_TIM_PeriodElapsedCallback(TIM_HandleTypeDef *htim)
{
    if(htim->Instance== TIM3)
    {
        encoder_cnt= TIM2->CNT;

        encoder=(float)((encoder_cnt*60)/(4*374));
        if (encoder_cnt>=0)
        {
            speed=encoder;
            send_data(VELO_F,encoder);
        }
        else if(encoder_cnt<0)
        {
            speed=-encoder;
            send_data(VELO_R,encoder);
        }
    }
    if (!state){} //Trong phần PWM có giải thích
    else {}
    TIM2->CNT=0;
}
```

### 2.1.3. Giải thuật PID rời rạc.

Hàm truyền bộ điều khiển PID

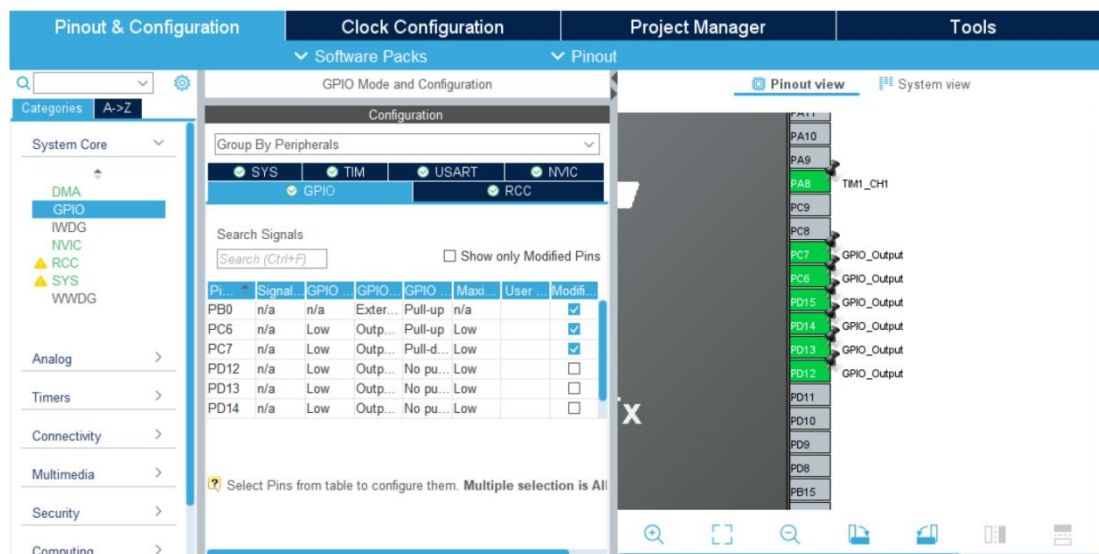
$$G_{PID}(z) = K_P + \frac{K_I z + 1}{2} \frac{1}{z - 1} + \frac{K_D z - 1}{T} \frac{1}{z}$$

float PID\_control (float setpoint, float measure)

```
{
    ek_2=ek_1;
    ek_1=ek;
    ek=setpoint-measure;
    uk_1=uk;
    uk=uk_1+Kp*(ek-ek_1) +Ki*(T/2)*(ek+ek_1)+ (Kd/T)*(ek_2*ek_1+ek_2);
    if (uk>umax) uk=umax;
    if (uk<umin) uk=umin;
    return (uk);
}
```

### 2.3. Lập trình sáng LED.

Chọn chân PD15, PD14, PD13, PD12 ở chế độ Output để tận dụng 4 LED có sẵn trên kit



#### Các hàm quan trọng

- Hàm sáng/tắt LED:

HAL\_GPIO\_WritePin(GPIOD, GPIO\_PIN\_12, 1); //1: LED sáng, 0: LED tắt

Áp dụng vào project

```
switch (buffer[1])
{
    case LED12:
    {
        if(data_float == 0)
        {
            HAL_GPIO_WritePin(GPIOD, GPIO_PIN_12, 0);
        }
    }
}
```



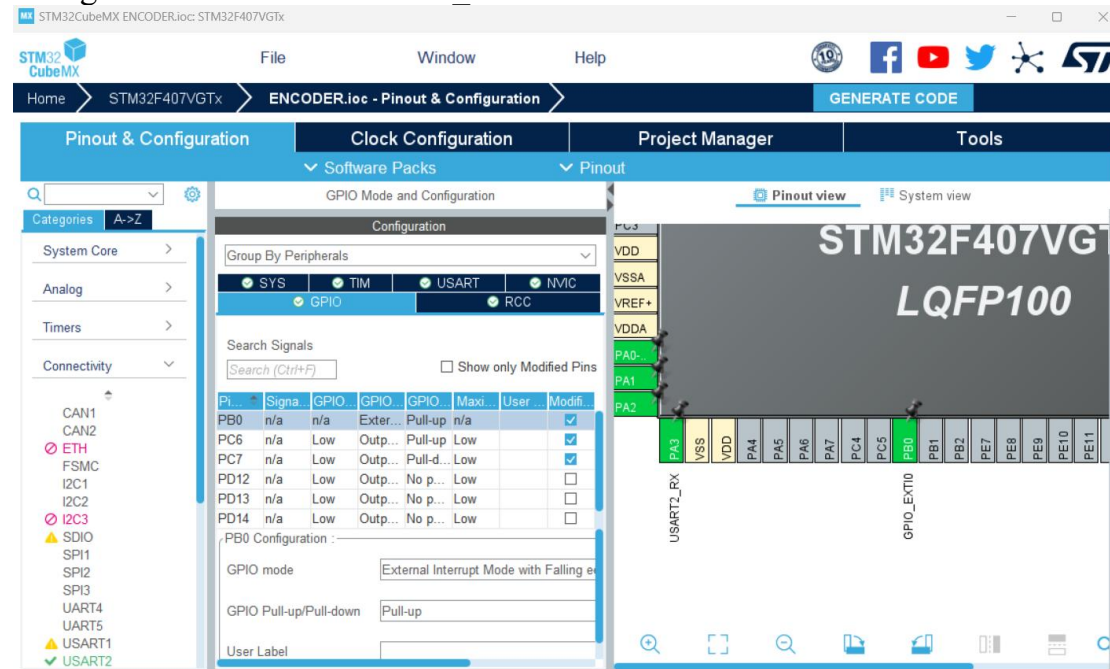
```

        break;
    }
    else
    {
        HAL_GPIO_WritePin(GPIOD, GPIO_PIN_12, 1);
        break;
    }
}
case LED13:
{
    if(data_float == 0)
    {
        HAL_GPIO_WritePin(GPIOD, GPIO_PIN_13, 0);
        break;
    }
    else
    {
        HAL_GPIO_WritePin(GPIOD, GPIO_PIN_13, 1);
        break;
    }
}
case LED14:
{
    if(data_float == 0)
    {
        HAL_GPIO_WritePin(GPIOD, GPIO_PIN_14, 0);
        break;
    }
    else
    {
        HAL_GPIO_WritePin(GPIOD, GPIO_PIN_14, 1);
        break;
    }
}
case LED15:
{
    if(data_float == 0)
    {
        HAL_GPIO_WritePin(GPIOD, GPIO_PIN_15, 0);
        break;
    }
    else
    {
        HAL_GPIO_WritePin(GPIOD, GPIO_PIN_15, 1);
        break;
    }
}

```

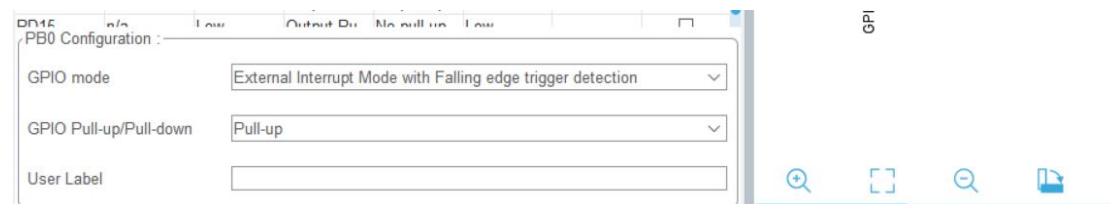
## 2.4. Lập trình nút nhấn.

Trong GPIO chân PB0 là GPIO\_EXT0



GPIO mode: External Interrupt Mode Falling edge: Ngắt ngoài chế độ phát hiện xung xuống

GPIO Pull: Có dùng trở kéo lên hoặc xuống không. Ta chọn trở kéo lên Pull UP



Vì ta đã kéo trở lên nên khi nhấn nút nhấn sẽ có tích cực cạnh xuống, chương trình sẽ nhảy vào ngắt trong hàm

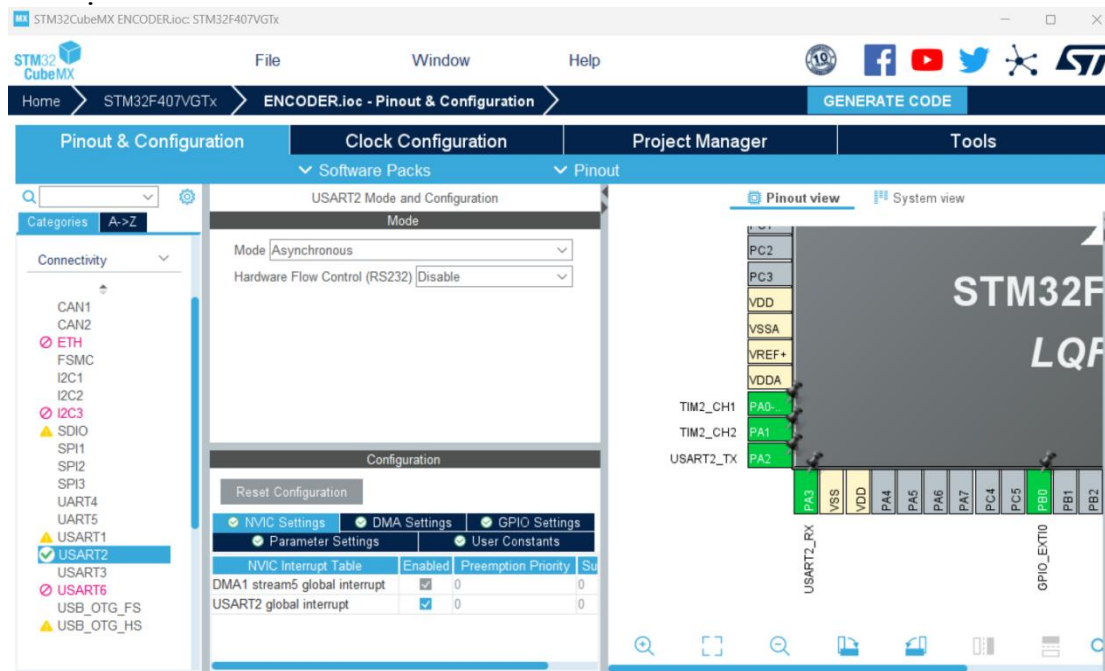
```
void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin)
```

Áp dụng vào chương trình

```
void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin)
{
    cnt++;
    state = 1;
    uint8_t buffer[10] = {0};
    buffer[0] = EMER;
    HAL_UART_Transmit(&huart2, buffer, 8, 1000);
    //Chong rung phim
    for(int i = 100000; i>0; i--);
}
```

## 2.5. Lập trình giao tiếp nối tiếp RS232.

Ta chọn 2 chân PA2 và PA3 ứng với Tx và Rx UART2 của MCU và thiết lập chế độ DMA



### Các hàm quan trọng

- Hàm khởi tạo DMA

```
HAL_UARTEx_ReceiveToIdle_DMA(&huart2, receive, 10);
```

```
__HAL_DMA_DISABLE_IT(&hdma_usart2_rx, DMA_IT_HT);
```

Khi có tín hiệu đến, sẽ vào ngắt DMA có hàm sau

```
void HAL_UARTEx_RxEventCallback(UART_HandleTypeDef *huart, uint16_t Size)
```

Trong hàm đó ta sẽ thực hiện công việc xử lý chuỗi, tất cả có trong chương trình

```
if(huart->Instance== huart2.Instance)
{
    uint8_t buffer[10] = {0};
    HAL_UARTEx_ReceiveToIdle_DMA(&huart2, receive, 10);
    size = Size;

    len = strlen((char *)receive);
    int index=0;
    int position =0;
    for (int i=0; i < len; i++)
    {
        if (receive[i] == ETX)
        {
            index ++;
            if (index == 1)
            {
                position = i;
                break;
            }
        }
    }
}
```

```

    }
}
uint8_t data_bytes[4] = {0};
for (int j=0; j<=position; j++)
buffer[j]=receive[j];
if(buffer[0]==STX & buffer[position]==ETX)
{

    for (int j=0; j<=position-3;j++)
    data_bytes[j]=buffer[j+2];

    data_float = (float)atof((char *)data_bytes);

    switch (buffer[1])
    {
        case SET_POINT:
        {
            if(data_float >0)
            {
                speed_set=data_float;
                direct=0;//forward
                break;
            }
            else
            {
                speed_set=-data_float;
                direct=1;//reverse
                break;
            }
        }
        case START:
        state=0;
        break;
        case STOP:
        __HAL_TIM_SET_COMPARE(&htim2, TIM_CHANNEL_1, 0);
        state=1;
        break;
        case KP:
            Kp=data_float;
            break;
        case KI:
            Ki=data_float;
            break;
        case KD:
            Kd=data_float;

```

```

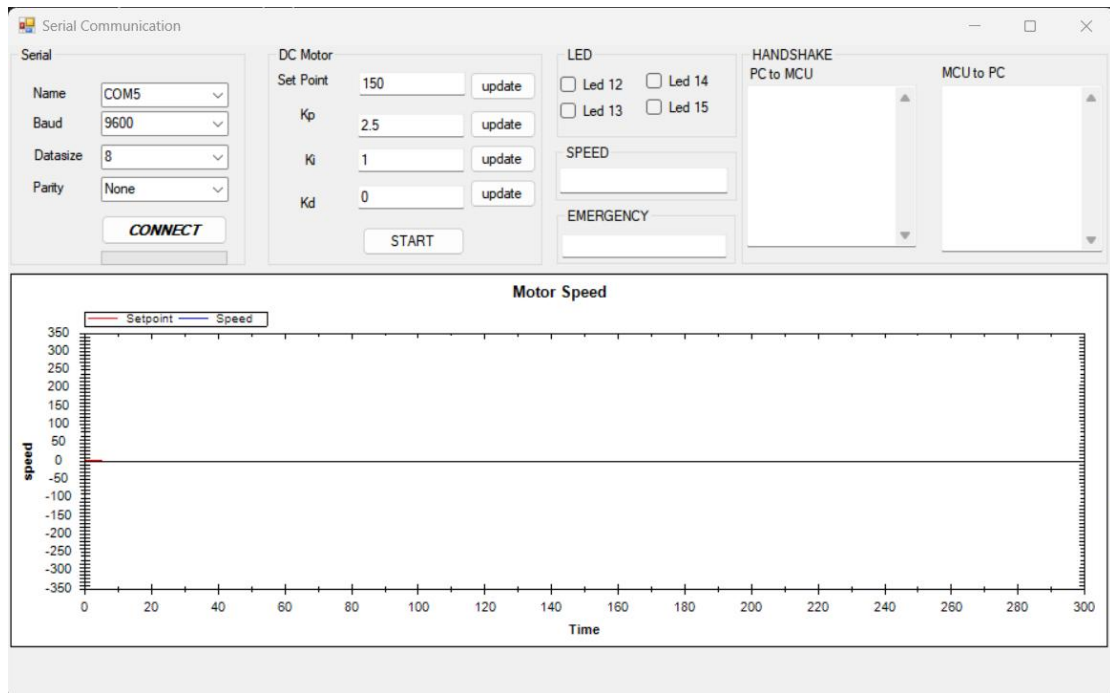
break;
case LED12:
{
if(data_float == 0)
{
HAL_GPIO_WritePin(GPIOD, GPIO_PIN_12, 0);
break;
}
else
{
HAL_GPIO_WritePin(GPIOD, GPIO_PIN_12, 1);
break;
}
}
case LED13:
{
if(data_float == 0)
{
HAL_GPIO_WritePin(GPIOD, GPIO_PIN_13, 0);
break;
}
else
{
HAL_GPIO_WritePin(GPIOD, GPIO_PIN_13, 1);
break;
}
}
case LED14:
{
if(data_float == 0)
{
HAL_GPIO_WritePin(GPIOD, GPIO_PIN_14, 0);
break;
}
else
{
HAL_GPIO_WritePin(GPIOD, GPIO_PIN_14, 1);
break;
}
}
case LED15:
{
if(data_float == 0)
{
HAL_GPIO_WritePin(GPIOD, GPIO_PIN_15, 0);
break;
}
}

```

```
        }
        else
        {
            HAL_GPIO_WritePin(GPIOD, GPIO_PIN_15, 1);
            break;
        }
    }
}

HAL_UART_Transmit(&huart2, buffer, 8, 1000);
}
```

### CHƯƠNG 3: LẬP TRÌNH GUI TRÊN C#



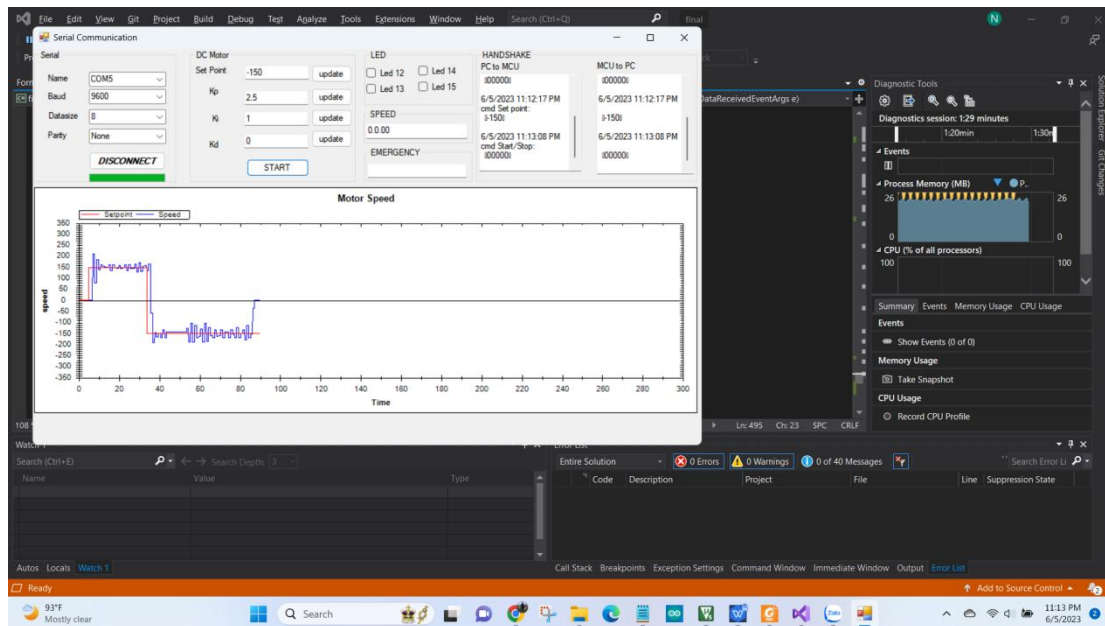
#### Frame truyền

STX	Command	Data	ETX
-----	---------	------	-----

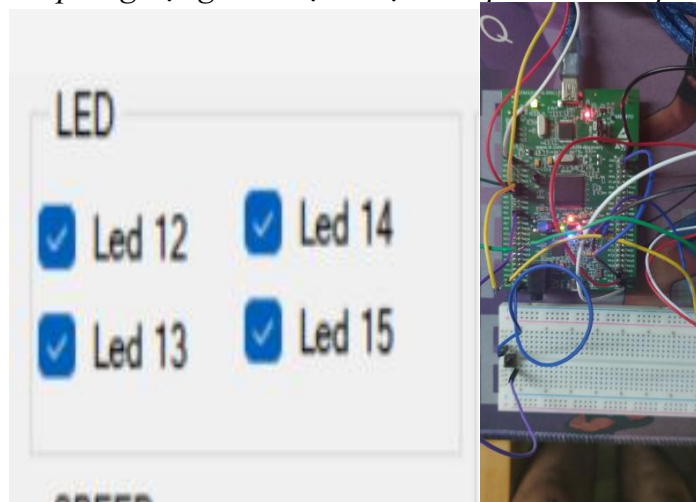
#### Function truyền nhận trên C#

	PC to MCU	MCU to PC
RS232	serialPort1.Write();	serialPort1.ReadExisting();

## CHƯƠNG 4: KẾT QUẢ THỰC HIỆN

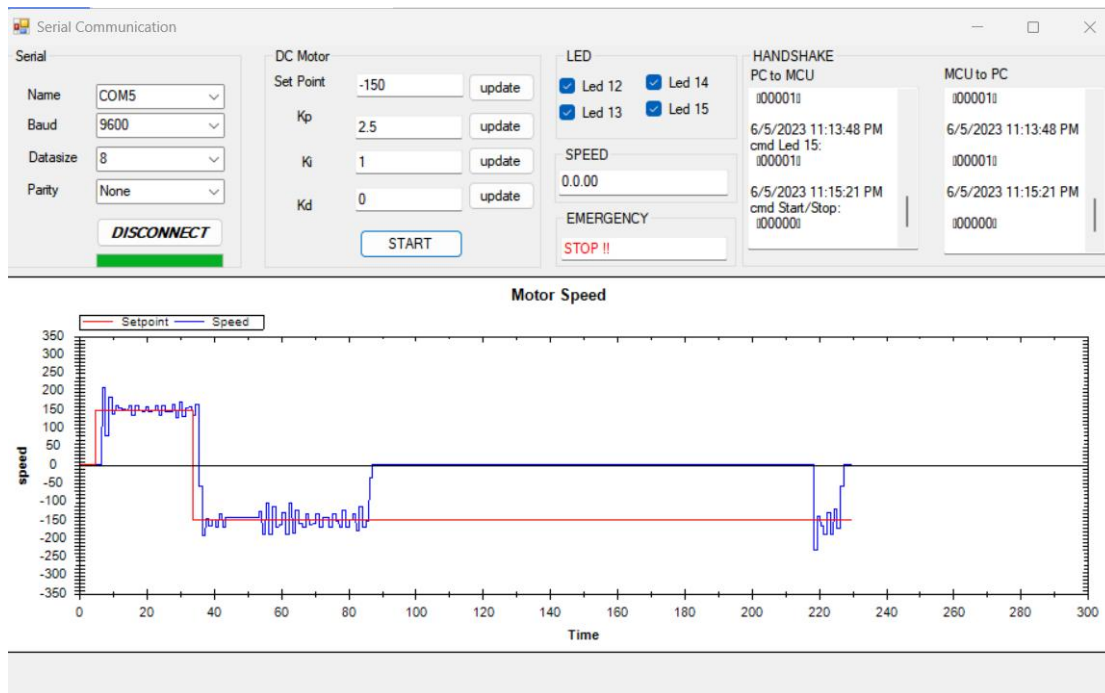


*Đáp ứng động cơ ở vị trí đặt 150rpm và -150rpm*

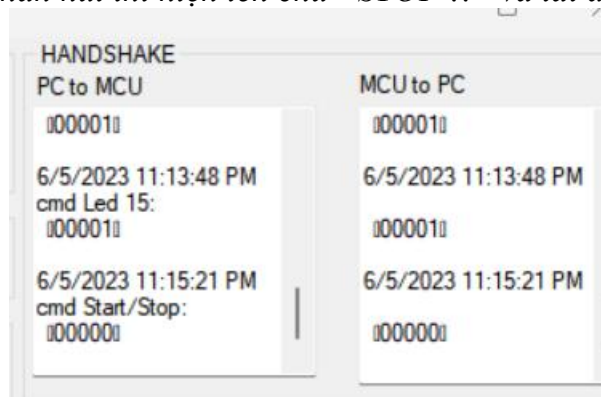


*Sáng 4 LED*





*Khi nhấn nút thì hiện lên chữ “STOP !!” và tắt động cơ*



*Handshake*

## TÀI LIỆU THAM KHẢO

1. Khaled Magdy, STM32 DC Motor Speed Control PWM With L293D  
<https://deepbluembedded.com/stm32-dc-motor-speed-control-pwm-l293d-examples/>
2. Pay It Forward Club, Lập trình C# và ứng dụng với Vi Điều Khiển  
<https://payitforward.edu.vn/wordpress/tutorials/lap-trinh-csharp-va-ung-dung-voi-vi-dieu-khien/>