



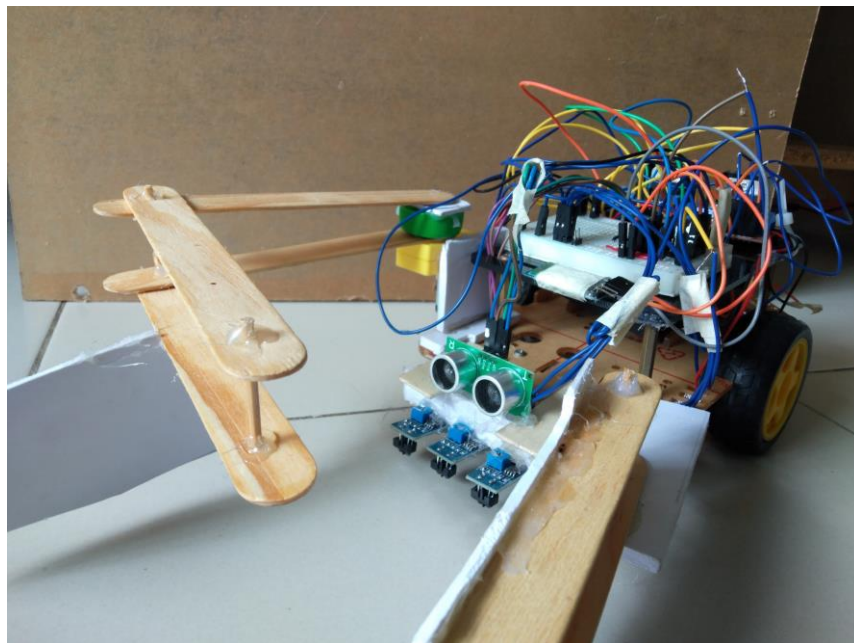
HCMC UNIVERSITY OF TECHNOLOGY AND EDUCATION
FACULTY OF MECHANICAL ENGINEERING



MICO236929

MICROCONTROLLER PROJECT REPORT

TWO-WHEELED MOBILE ROBOT



Ho Chi Minh City, 6/2022

List of members

No.	Student name	Student ID	Sign
01	Trần Ngọc Hiểu	20146127	
02	Nguyễn Bá Vũ Thạch	20146530	
03	Lê Văn Tâm	20146526	
04	Bùi Đăng Khoa	20146108	
05	Đào Thanh Trọng	20146130	

Contribution

ACTIVITY	TIME (From date to date)		RESPONSIBILITY
Schematic design	11/6/2022	20/6/2022	Hiếu, Thạch, Trọng, Khoa.
Hardware building	11/6/2022	19/6/2022	Hiếu, Tâm, Trọng, Khoa.
Coding: Project setup and code library	13/6/2022	16/6/2022	Hiếu, Thạch.
Coding: Algorithm development	18/6/2022	20/6/2022	Hiếu, Thạch, Tâm, Khoa.
Coding: Debug and final test	18/6/2022	20/6/2022	Hiếu, Thạch, Tâm, Khoa, Trọng.
Report	23/6/2022	24/6/2022	Hiếu, Thạch.



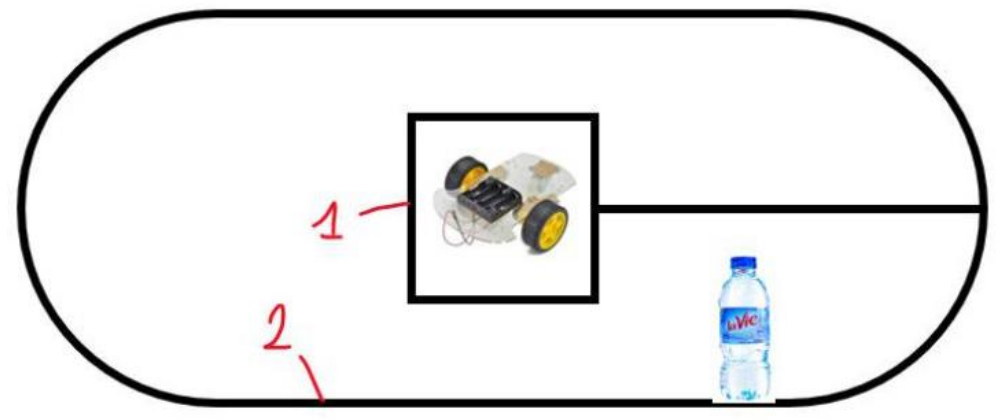
Members of MPLABX team

Introduction

Thông báo dự án cuối kỳ xây dựng và lập trình robot di động hai bánh xe theo yêu cầu sau:

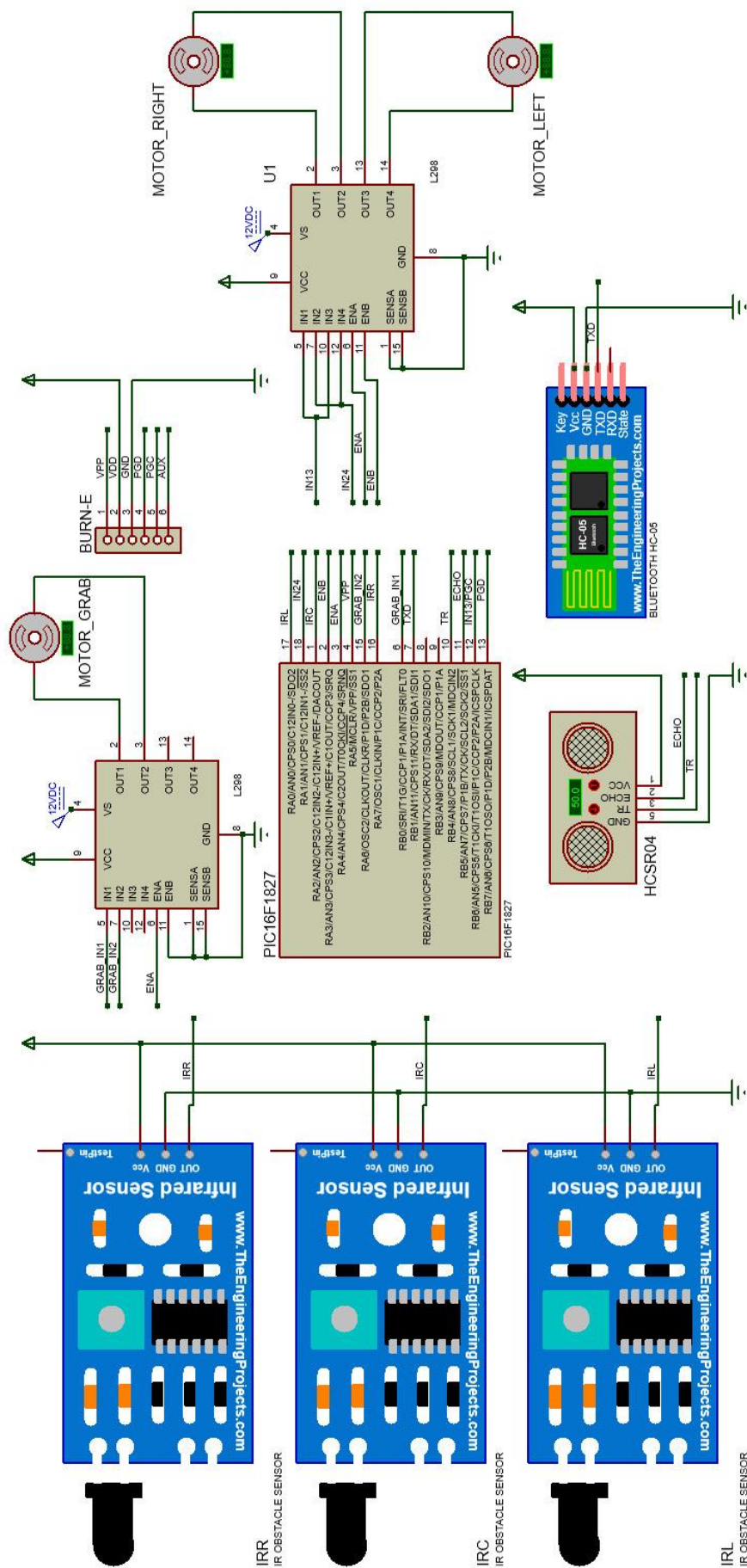
1. Robot di động hai bánh xe di chuyển **tự động** từ hình vuông 1 đến kẹp chai nước thì dừng lại. (Vị trí chai nước trên đường 2 là ngẫu nhiên)
2. Người điều khiển (có dây hoặc không dây) cho robot đặt chai nước vào hình vuông 1 trong tổng thời gian tối đa **30 giây**.
3. Đường line là băng keo điện màu đen trên nền gạch lớp học. Chai nước 350ml rỗng (không có nước)
4. Sử dụng vi điều khiển PIC.

Requirement

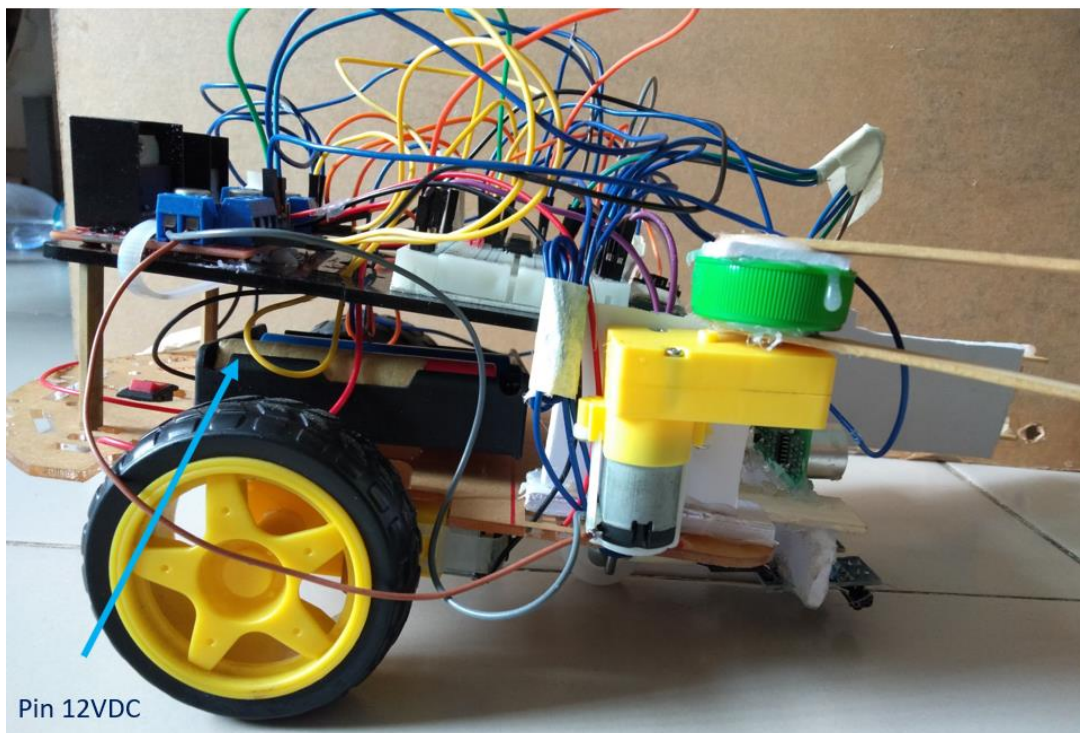
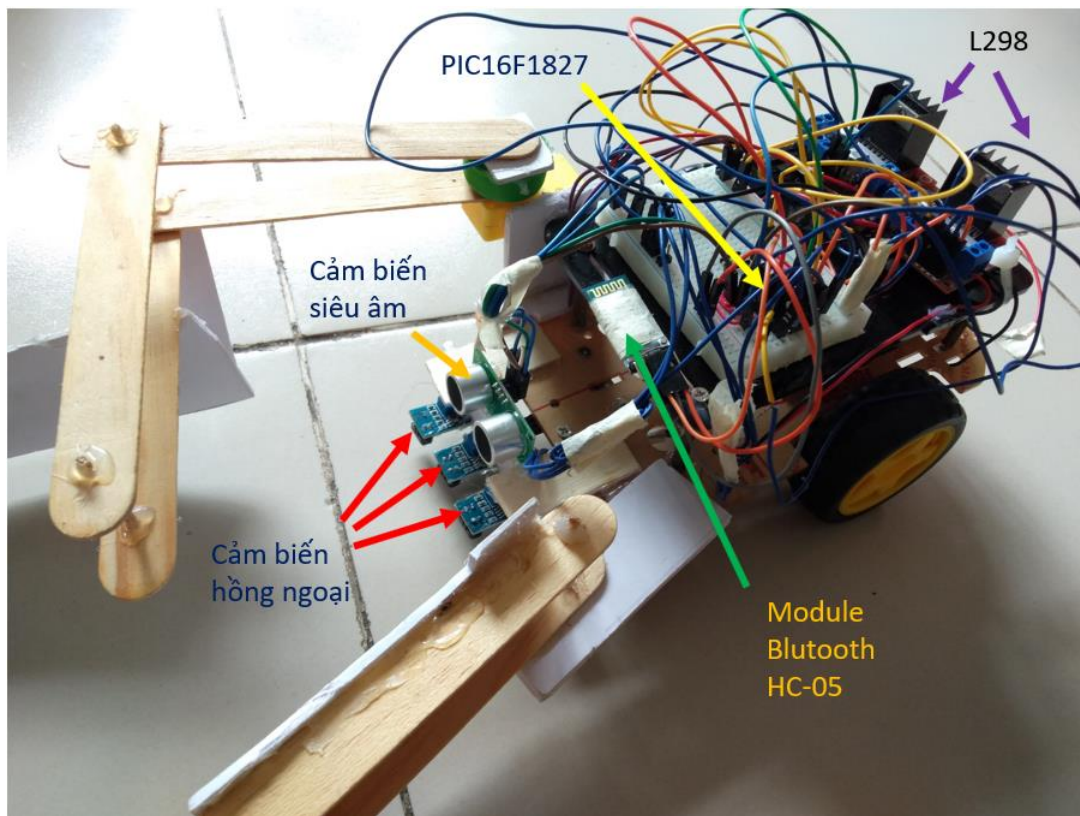


Map

1. Schematic design:



2. Hardware building:



3. Field test:





- Link youtube video chạy xe hoàn thành map:

<https://youtu.be/xZAPyyajufk>

4. Code Appendix:

```
#define IRL PORTAbits.RA0 //Left sensor
#define IRR PORTAbits.RA7 //Right sensor
#define IRC PORTAbits.RA2 //Center sensor
#define IN1_3 LATBbits.LATB6
#define IN2_4 LATABbits.LATA1
#define echo PORTBbits.RB5
#define trig PORTBbits.RB4
#define servo_IN1 LATBbits.LATB0
#define servo_IN2 LATABbits.LATA6
#include "mcc_generated_files/mcc.h"

char str[50];
```

```

unsigned int mode_manual = 1,mode_auto = 0;

int Saved_Level,PWM_Left,PWM_Right,Level_Right,Level_Left;

int Error,Previous_Error = 0;

unsigned int distance_us;

float distance_cm;


void Stop()
{
    IN1_3 = 0;
    IN2_4 = 0;
}

void Go_Ahead()
{
    IN1_3 = 0;
    IN2_4 = 1;
}

void Go_Back()
{
    IN1_3 = 1;
    IN2_4 = 0;
}

void Servo_Open()
{
    servo_IN1 = 1;
    servo_IN2 = 0;
}

void Servo_Close()
{

```



```

servo_IN1 = 0;

servo_IN2 = 1;
}

```

```

void Servo_Stop()

{

servo_IN1 = 0;

servo_IN2 = 0;

}

```

```

void main(void)

{

while(1){

Servo_Close();

Level_Right = Level_Left = 5;

Saved_Level = 1;

SYSTEM_Initialize();

while (mode_manual == 1)

{

PWM_Left =Level_Left*100+99;

PWM_Right =Level_Right*100+99;

PWM3_LoadDutyValue(PWM_Left);

PWM4_LoadDutyValue(PWM_Right);

for (int i=0;i<10;i++){

while ((EUSART_is_rx_ready()== 0));

str[i]=EUSART_Read();

if(str[i]=='B')//Go back

{

```

```

    if (Level_Right > Level_Left ) {Level_Left = Level_Right;}

    else {Level_Right = Level_Left;}

    Saved_Level = Level_Right;

    Go_Back();

    __delay_ms(20);
}

if (str[i]=='F')//Go ahead
{
    if (Level_Right > Level_Left ) {Level_Left = Level_Right;}

    else {Level_Right = Level_Left;}

    Saved_Level = Level_Right;

    Go_Ahead();

    __delay_ms(20);
}

//Control velocity

if (str[i]=='0') Saved_Level = Level_Left = Level_Right = 0;
if (str[i]=='1') Saved_Level = Level_Left = Level_Right = 1;
if (str[i]=='2') Saved_Level = Level_Left = Level_Right = 2;
if (str[i]=='3') Saved_Level = Level_Left = Level_Right = 3;
if (str[i]=='4') Saved_Level = Level_Left = Level_Right = 4;
if (str[i]=='5') Saved_Level = Level_Left = Level_Right = 5;
if (str[i]=='6') Saved_Level = Level_Left = Level_Right = 6;
if (str[i]=='7') Saved_Level = Level_Left = Level_Right = 7;
if (str[i]=='8') Saved_Level = Level_Left = Level_Right = 8;
if (str[i]=='9') Saved_Level = Level_Left = Level_Right = 9;
if (str[i]=='q') Saved_Level = Level_Left = Level_Right = 10;

if (str[i]=='L')//Turn left

```

```

{
    Level_Left = 0;

    Level_Right = Saved_Level;

    __delay_ms(1);

    Go_Ahead();

    __delay_ms(20);
}

if (str[i]=='R')//Turn right
{
    Level_Right = 0;

    Level_Left = Saved_Level;

    __delay_ms(1);

    Go_Ahead();

    __delay_ms(20);
}


if (str[i]=='G')//Forward left
{
    Level_Right = Saved_Level;

    Level_Left = Saved_Level-2 ;

    if (Level_Left < 0) Level_Left = 0;

    Go_Ahead();

    __delay_ms(20);
}

if (str[i]=='T')//Forward right
{
    Level_Right = 0;

    Level_Left = Saved_Level;

```

```

Level_Right = Saved_Level-2 ;

if (Level_Right < 0) Level_Right = 0;

Go_Ahead();

__delay_ms(20);
}

if (str[i]=='V')//Mode Auto
{
    mode_manual = 0;
    mode_auto = 1;
}

if (str[i] == 'U')//Servo Close
{
    Servo_Close();
    __delay_ms(120);
    Servo_Stop();
    __delay_ms(10);
}

if (str[i] == 'u')//Servo Open
{
    Servo_Open();
    __delay_ms(120);
    Servo_Stop();
    __delay_ms(10);
}

else {Stop();}

```



```

    }
}

//Blind Running
PWM3_LoadDutyValue(600); PWM4_LoadDutyValue(600);

    __delay_us(1);

    Go_Ahead();

    __delay_ms(300);

    PWM3_LoadDutyValue(0); PWM4_LoadDutyValue(0);

    __delay_ms(20);

distance_cm = 5;

while(mode_auto == 1){

    //Ultrasonic sensor

    if ( distance_cm >= 5)

    {

        trig = 1;

        __delay_us(500);

        trig = 0;

        while (echo == 0);

        TMR0 = 0;

        while (echo == 1);

        distance_us = TMR0*32*4;

        distance_cm = (float)distance_us/58;

    }

    //Obstacle Sensor

    if ((IRL == 0)&&(IRC == 0)&&(IRR == 1)) Error = +1;

    else if ((IRL == 0)&&(IRC == 1)&&(IRR == 0)) Error = 0;

    else if ((IRL == 1)&&(IRC == 0)&&(IRR == 0)) Error = -1;

```

```

//Line Tracking Algorithm

if ((IRL == 1)&&(IRC == 0)&&(IRR == 1))
{
    PWM3_LoadDutyValue(0);

    PWM4_LoadDutyValue(0);

    Stop();

    __delay_ms(300);

    PWM3_LoadDutyValue(700);

    PWM4_LoadDutyValue(0);

    Go_Ahead();

    __delay_ms(800);

    PWM3_LoadDutyValue(0);

    PWM4_LoadDutyValue(0);

}

if ((IRL == 1)&&(IRC == 1)&&(IRR == 1))
{
    PWM3_LoadDutyValue(0);

    PWM4_LoadDutyValue(0);

    Stop();

    __delay_ms(300);

    PWM3_LoadDutyValue(700);

    PWM4_LoadDutyValue(0);

    Go_Ahead();

    __delay_ms(800);

    PWM3_LoadDutyValue(0);

```

```

    PWM4_LoadDutyValue(0);
}

if (Error == -1)
{
    PWM3_LoadDutyValue(0);
    PWM4_LoadDutyValue(850);
    __delay_us(1);
    Go_Ahead();
    __delay_ms(5);
    PWM3_LoadDutyValue(0);
    PWM4_LoadDutyValue(460);
    __delay_ms(2);
    Previous_Error = Error;
}

if (Error == +1)
{
    PWM4_LoadDutyValue(0);
    PWM3_LoadDutyValue(750);
    __delay_us(1);
    Go_Ahead();
    __delay_ms(5);
    PWM3_LoadDutyValue(460); PWM4_LoadDutyValue(0);
    __delay_ms(2);
    Previous_Error = Error;
}

else
{

```

```

    PWM3_LoadDutyValue(400); PWM4_LoadDutyValue(400);

    __delay_us(1);

    Go_Ahead();

}

}

else {mode_auto = 0;mode_manual = 1;Stop();

//Auto grab

PWM3_LoadDutyValue(750);

PWM4_LoadDutyValue(750);

Servo_Close();

__delay_ms(120);

Servo_Stop();

__delay_ms(10);}

}

}

}

```