KING MONGKUT'S INSTITUTE OF TECHNOLOGY LATKRABANG
SCHOOL OF ENGINEERING

Final Project - Micro

INSTRUCTED BY : Dr. Poom Konghuayrob

NAME : Varis Saligupta, Akesit Akkharsaksiri, Krittin Sakharin
ID NUMBER : 65011619, 65110131,65011356
GROUP NO : 2
DATE OF SUB : 27/5/2023

Purposes

- Understanding concepts of PCB and Robot
- Application of STM 32
- To experience the workspaces and routines of engineers.
- Encouraging teamwork; combining different ideas and perspectives.

Scope

- A line following robot that has the ability and agility to overcome .

Background and research

- PID Control:

PID control (Proportional-Integral-Derivative) is a widely used control algorithm in engineering and robotics. It is a feedback control mechanism that continuously adjusts the output based on the error between the desired setpoint and the measured process variable. The PID controller consists of three components: the proportional term (P), the integral term (I), and the derivative term (D).

The proportional term provides a control action proportional to the current error, the integral term accounts for accumulated past errors, and the derivative term anticipates future errors by considering the rate of change of the error. By combining these three terms, the PID controller can effectively regulate the system and reduce steady-state errors, improve response time, and enhance stability.

- STM32 Microcontroller:

The STM32 microcontroller is a family of microcontrollers developed by STMicroelectronics. It is based on the ARM Cortex-M processor architecture, offering a wide range of features and capabilities suitable for embedded systems and microcontroller applications. The STM32 microcontroller series provides a scalable platform with different variations, offering varying levels of performance, power efficiency, and peripheral integration.

The STM32 microcontrollers are known for their high-performance computing capabilities, low power consumption, extensive connectivity options, and rich peripheral set. They offer features such as timers, communication interfaces (UART, SPI, I2C), analog-to-digital converters (ADC), digital-to-analog converters (DAC), and general-purpose input/output (GPIO) pins. The STM32 microcontrollers are often used in applications such as robotics, industrial automation, Internet of Things (IoT) devices, and consumer electronics.

- **EasyEDA:**

EasyEDA is a web-based Electronic Design Automation (EDA) tool that provides a user-friendly platform for designing and prototyping electronic circuits. It offers a suite of design and simulation tools accessible through a web browser, eliminating the need for complex software installations.

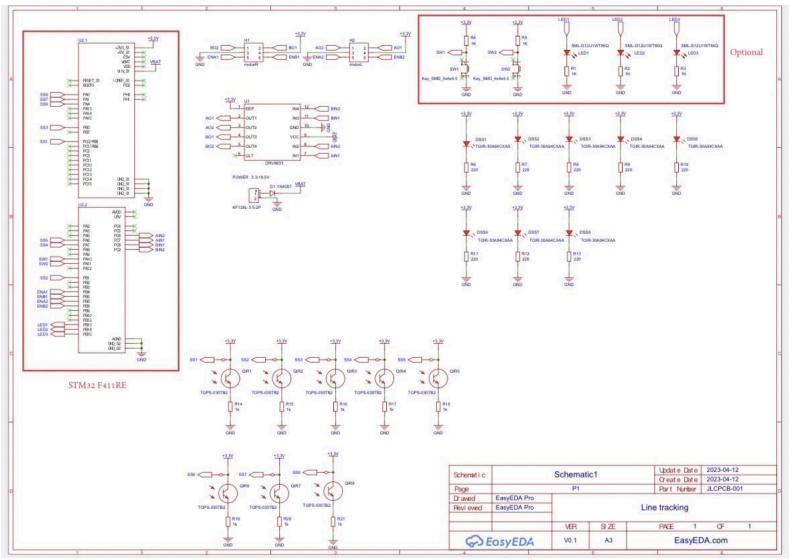
EasyEDA allows users to create and design schematic diagrams, PCB layouts, and even simulate circuit behavior. It provides a vast component library with a wide range of electronic components, symbols, and footprints, making it easy to create circuit designs. EasyEDA also offers collaboration features, allowing multiple users to work on the same project simultaneously, making it suitable for team-based projects.

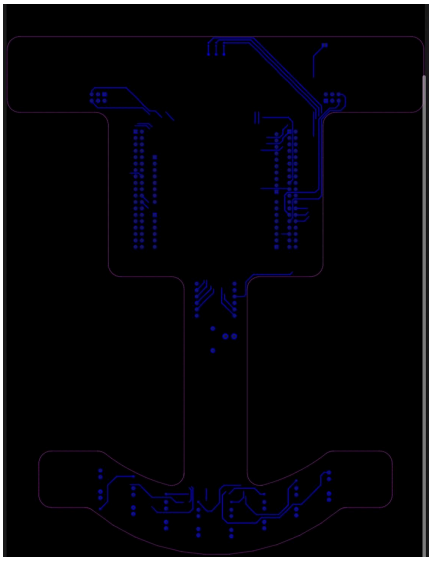
The platform provides integration with various manufacturers, enabling seamless ordering of PCB prototypes directly from the tool. EasyEDA has gained popularity due to its ease of use, accessibility, and community-driven support.

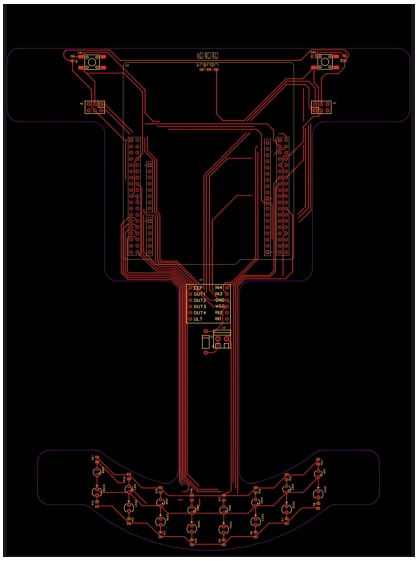
List of component

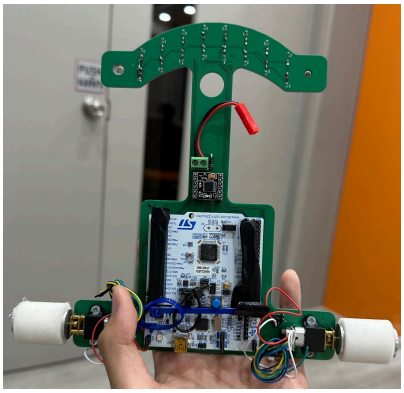
- 1x DRV8833 Motor Driver
- 1x STM32F411RE Microcontroller
- 2x CH-N20-3 Geared DC Motors
- 2x C23873 SMD Tactile Switch
- 4x LED 0402 SMD
- 8x 220Ω 0603 SMD Resistors
- 14x 1kΩ 0603 SMD Resistors
- 8x TOIR-30A94CXAA Infrared LEDs
- 8x TOPS-030TB2 Infrared Phototransistors
- 1x 1N4007 Rectifier Diode
- 1x GNB5502S70A 2S 7.4V 70C/140C 550 mAh LiPo Battery

Line Tracking robot

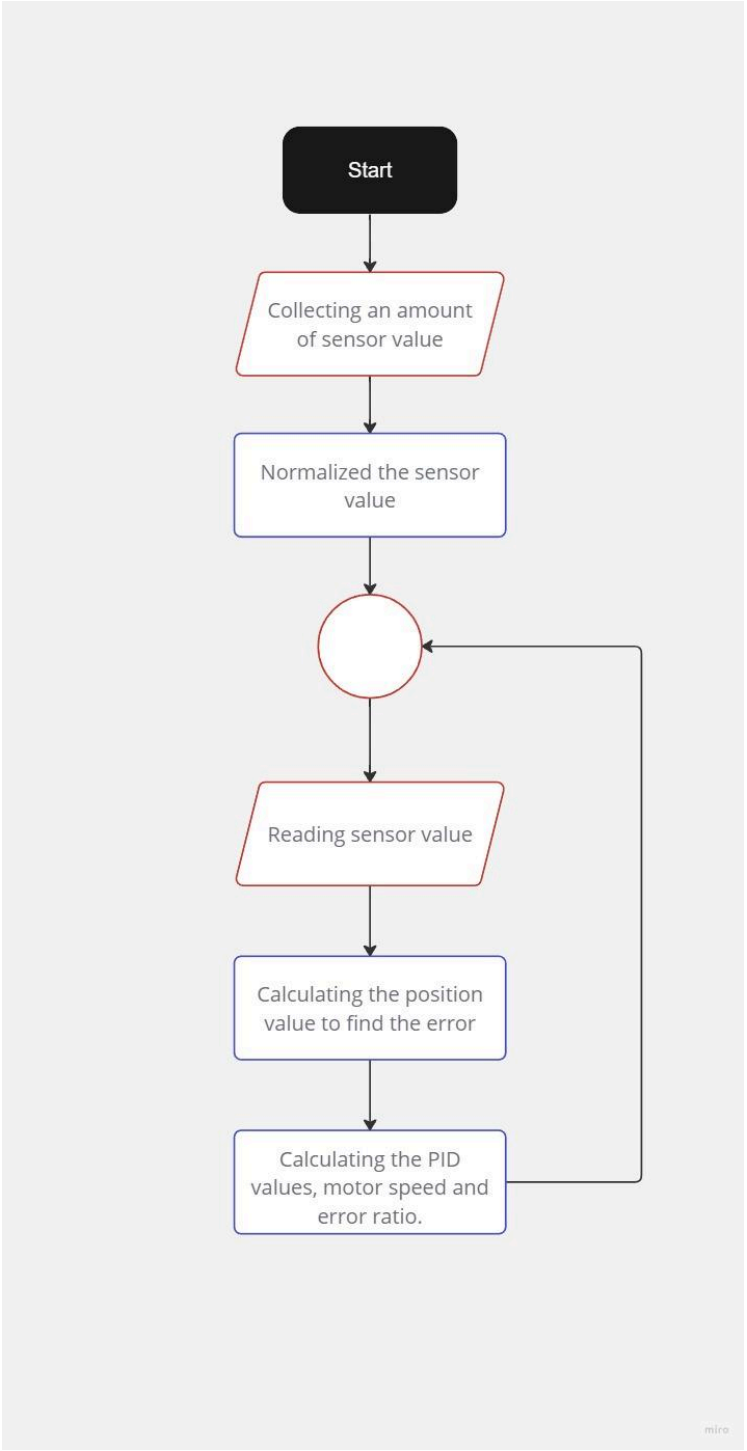


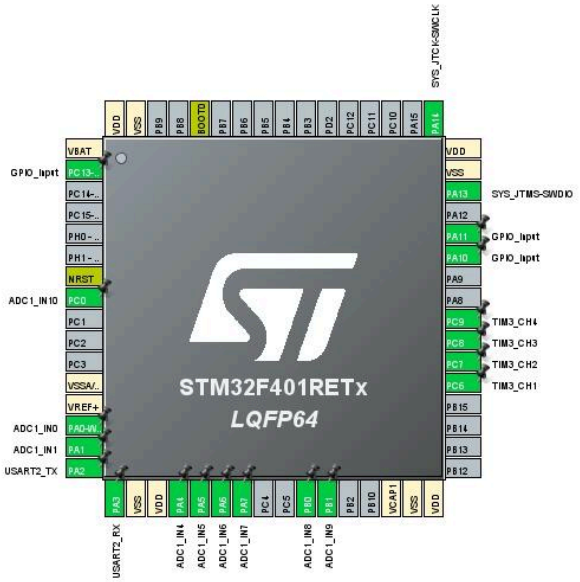






Code


Start
Collecting an amount of sensor value
Normalized the sensor value
Reading sensor value
Calculating the position value to find the error
Calculating the PID values, motor speed and error ratio.
miro


STM32F401RETx
LQFP64
SYS_JTCK-SWCLK
SYS_JTMS-SWDIO
GPIO_Input
ADC1_IN10
TIM3_CH4
TIM3_CH3
TIM3_CH2
TIM3_CH1
ADC1_IN0
ADC1_IN1
USART2_TX
USART2_RX
ADC1_IN4
ADC1_IN5
ADC1_IN6
ADC1_IN7
ADC1_IN8
ADC1_IN9

```
while (1)
  {
                TIM3->CCR3=0;//backward.L
                 TIM3->CCR4=0;//forward.L
                 TIM3->CCR1=0;//backward.R
                 TIM3->CCR2=0;//forward.R
while(HAL_GPIO_ReadPin(GPIOC,GPIO_PIN_13) == 0){
                 for(int i=0;i<8;i++){
                                if (sensors_max[i]<adc_current[i]){
                                                sensors_max[i]=adc_current[i];
                                }
                                if(sensors_min[i]>adc_current[i]){
                                                 sensors_min[i]=adc_current[i];
                                }
                  }
}
for(int i=0;i<8;i++){
                   sensors_avg[i]=((sensors_max[i]+sensors_min[i])/2);
}
if (HAL_GPIO_ReadPin(GPIOA,GPIO_PIN_10) == 0){
                while(HAL_GPIO_ReadPin(GPIOA,GPIO_PIN_10) == 1);
                //while(1){
                // sprintf(&text2,"%.2f,%.2f,%.2f,%.2f,%.2f,%.2f,%.2f,%.2f\r\n",(float)adc_current[0],(float)adc_current[1],(float)adc_current[2],(float)adc_current[3]
                ,(float)adc_current[4],(float)adc_current[5],(float)adc_current[6],(float)adc_current[7]);

                //sprintf(&text2,"%.2f,%.2f,%.2f,%.2f,%.2f,%.2f,%.2f,%.2f\r\n",(float)sensors_avg[0],(float)sensors_avg[1],(float)sensors_avg[2],(float)sensors_avg[3],
                (float)sensors_avg[4],(float)sensors_avg[5],(float)sensors_avg[6],(float)sensors_avg[7]);
                //  HAL_UART_Transmit(&huart2,text2, strlen(text2),1000);
                  //}
                TIM3->CCR3=0;//backward.L
                TIM3->CCR4=350;//forward.L
                TIM3->CCR1=0;//backward.R
                TIM3->CCR2=350;//forward.R
                HAL_Delay(400);
                PID22(250,0.21,0,0,550,-550,600);//230
  }
  if (HAL_GPIO_ReadPin(GPIOA,GPIO_PIN_11) == 0){
                while(HAL_GPIO_ReadPin(GPIOA,GPIO_PIN_11) == 1);
                TIM3->CCR3=0;//backward.L
                TIM3->CCR4=350;//forward.L
                TIM3->CCR1=0;//backward.R
                TIM3->CCR2=350;//forward.R
                HAL_Delay(400);
                while(1){
                 if(sensors_avg[1]<adc_current[1]&&sensors_avg[2]<adc_current[2]&&sensors_avg[3]<adc_current[3]&&sensors_avg[4]<adc_current[4]
                &&sensors_avg[5]<adc_current[5]&&sensors_avg[6]<adc_current[6]){
                                Error=Lasterror
                }//allwhite
                if(sensors_avg[1]>adc_current[1]&&sensors_avg[2]>adc_current[2]&&sensors_avg[3]>adc_current[3]&&sensors_avg[4]>adc_current[4]
                &&sensors_avg[5]>adc_current[5]&&sensors_avg[6]>adc_current[6]){
                                Error=0*500;
                }//allblack
                if(sensors_avg[1]<adc_current[1]&&sensors_avg[2]<adc_current[2]&&sensors_avg[3]>adc_current[3]&&sensors_avg[4]>adc_current[4]
                &&sensors_avg[5]<adc_current[5]&&sensors_avg[6]<adc_current[6]){
                                Error=0*500;
                }//black3,4
                if(sensors_avg[1]<adc_current[1]&&sensors_avg[2]<adc_current[2]&&sensors_avg[3]<adc_current[3]&&sensors_avg[4]>adc_current[4]
                &&sensors_avg[5]<adc_current[5]&&sensors_avg[6]<adc_current[6]){
                                Error=1*500;
                }//black4
                if(sensors_avg[1]<adc_current[1]&&sensors_avg[2]<adc_current[2]&&sensors_avg[3]<adc_current[3]&&sensors_avg[4]>adc_current[4]
                &&sensors_avg[5]>adc_current[5]&&sensors_avg[6]<adc_current[6]){
                                Error=2*500;
                }//black4,5
                if(sensors_avg[1]<adc_current[1]&&sensors_avg[2]<adc_current[2]&&sensors_avg[3]<adc_current[3]&&sensors_avg[4]<adc_current[4]
                &&sensors_avg[5]>adc_current[5]&&sensors_avg[6]<adc_current[6]){
                                Error=3*500;
                }//black5
```

```
            if(sensors_avg[1]<adc_current[1]&&sensors_avg[2]<adc_current[2]&&sensors_avg[3]<adc_current[3]&&sensors_avg[4]<adc_current[4]
            &&sensors_avg[5]>adc_current[5]&&sensors_avg[6]>adc_current[6]){
                        Error=4*500;
            }//black5,6
            if(sensors_avg[1]<adc_current[1]&&sensors_avg[2]<adc_current[2]&&sensors_avg[3]<adc_current[3]&&sensors_avg[4]<adc_current[4]
            &&sensors_avg[5]<adc_current[5]&&sensors_avg[6]>adc_current[6]){
                        Error=5*500;
            }//black6
            if(sensors_avg[1]<adc_current[1]&&sensors_avg[2]<adc_current[2]&&sensors_avg[3]>adc_current[3]&&sensors_avg[4]<adc_current[4]
            &&sensors_avg[5]<adc_current[5]&&sensors_avg[6]<adc_current[6]){
                         Error=(-1*500);
            }//black3
            if(sensors_avg[1]<adc_current[1]&&sensors_avg[2]>adc_current[2]&&sensors_avg[3]>adc_current[3]&&sensors_avg[4]<adc_current[4]
            &&sensors_avg[5]<adc_current[5]&&sensors_avg[6]<adc_current[6]){
                         Error=(-2*500);
            }//black2,3
            if(sensors_avg[1]<adc_current[1]&&sensors_avg[2]>adc_current[2]&&sensors_avg[3]<adc_current[3]&&sensors_avg[4]<adc_current[4]
            &&sensors_avg[5]<adc_current[5]&&sensors_avg[6]<adc_current[6]){
                         Error=(-3*500);
            }//black2
            if(sensors_avg[1]>adc_current[1]&&sensors_avg[2]>adc_current[2]&&sensors_avg[3]<adc_current[3]&&sensors_avg[4]<adc_current[4]
            &&sensors_avg[5]<adc_current[5]&&sensors_avg[6]<adc_current[6]){
                         Error=(-4*500);
            }//black1,2
            if(sensors_avg[1]>adc_current[1]&&sensors_avg[2]<adc_current[2]&&sensors_avg[3]<adc_current[3]&&sensors_avg[4]<adc_current[4]
            &&sensors_avg[5]<adc_current[5]&&sensors_avg[6]<adc_current[6]){
                        Error=(-5*500);
            }//black1
            P=Error;
            I=I+Error;
            D=Error-Lasterror;
            Lasterror=Error;
            float PID =(0.54*P)+(0*I)+(0.32*D); // last p = 0.4 last d = 0.4
            sprintf(&text1,"%.2f\r\n",(float)PID);
            HAL_UART_Transmit(&huart2,text1, strlen(text1),1000);
            L=650+PID;
            R=650-PID;
            if(L>=0&&R>=0){
                        TIM3->CCR3=0;//backward.L
                        TIM3->CCR4=L;//forward.L
                        TIM3->CCR1=0;//backward.L
                        TIM3->CCR2=R;//forward.L
            }
            if(L>=0&&R<0){
                        TIM3->CCR3=0;//backward.L
                        TIM3->CCR4=L;//forward.L
                        TIM3->CCR1=-R;//backward.L
                        TIM3->CCR2=0;//forward.L
            }
            if(L<0&&R>=0){
                        TIM3->CCR3=-L;//backward.L
                        TIM3->CCR4=0;//forward.L
                        TIM3->CCR1=0;//backward.L
                        TIM3->CCR2=R;//forward.L
            }
            if(L<0&&R<0){
                        TIM3->CCR3=-L;//backward.L
                        TIM3->CCR4=0;//forward.L
                        TIM3->CCR1=-R;//backward.L
                        TIM3->CCR2=0;//forward.L
            }
           }
          }
         }
}
```

```
void PID22(int bs,float x,float y, float z,int max,int min,int ratio){
	while(1){
		if(sensors_avg[1]<adc_current[1]&&sensors_avg[2]<adc_current[2]&&sensors_avg[3]<adc_current[3]&&sensors_avg[4]<adc_current[4]
	&&sensors_avg[5]<adc_current[5]&&sensors_avg[6]<adc_current[6]){
		Error=Lasterror
	}//allwhite
	if(sensors_avg[1]>adc_current[1]&&sensors_avg[2]>adc_current[2]&&sensors_avg[3]>adc_current[3]&&sensors_avg[4]>adc_current[4]
	&&sensors_avg[5]>adc_current[5]&&sensors_avg[6]>adc_current[6]){
		Error=0*500;
	}//allblack
	if(sensors_avg[1]<adc_current[1]&&sensors_avg[2]<adc_current[2]&&sensors_avg[3]>adc_current[3]&&sensors_avg[4]>adc_current[4]
	&&sensors_avg[5]<adc_current[5]&&sensors_avg[6]<adc_current[6]){
		Error=0*500;
	}//black3,4
	if(sensors_avg[1]<adc_current[1]&&sensors_avg[2]<adc_current[2]&&sensors_avg[3]<adc_current[3]&&sensors_avg[4]>adc_current[4]
	&&sensors_avg[5]<adc_current[5]&&sensors_avg[6]<adc_current[6]){
		Error=1*500;
	}//black4
	if(sensors_avg[1]<adc_current[1]&&sensors_avg[2]<adc_current[2]&&sensors_avg[3]<adc_current[3]&&sensors_avg[4]>adc_current[4]
	&&sensors_avg[5]>adc_current[5]&&sensors_avg[6]<adc_current[6]){
		Error=2*500;
	}//black4,5
	if(sensors_avg[1]<adc_current[1]&&sensors_avg[2]<adc_current[2]&&sensors_avg[3]<adc_current[3]&&sensors_avg[4]<adc_current[4]
	&&sensors_avg[5]>adc_current[5]&&sensors_avg[6]<adc_current[6]){
		Error=3*500;
	}//black5


	if(sensors_avg[1]<adc_current[1]&&sensors_avg[2]<adc_current[2]&&sensors_avg[3]<adc_current[3]&&sensors_avg[4]<adc_current[4]
	&&sensors_avg[5]>adc_current[5]&&sensors_avg[6]>adc_current[6]){
		Error=4*500;
	}//black5,6
	if(sensors_avg[1]<adc_current[1]&&sensors_avg[2]<adc_current[2]&&sensors_avg[3]<adc_current[3]&&sensors_avg[4]<adc_current[4]
	&&sensors_avg[5]<adc_current[5]&&sensors_avg[6]>adc_current[6]){
		Error=5*500;
	}//black6
	if(sensors_avg[1]<adc_current[1]&&sensors_avg[2]<adc_current[2]&&sensors_avg[3]>adc_current[3]&&sensors_avg[4]<adc_current[4]
	&&sensors_avg[5]<adc_current[5]&&sensors_avg[6]<adc_current[6]){
		 Error=(-1*500);
	}//black3
	if(sensors_avg[1]<adc_current[1]&&sensors_avg[2]>adc_current[2]&&sensors_avg[3]>adc_current[3]&&sensors_avg[4]<adc_current[4]
	&&sensors_avg[5]<adc_current[5]&&sensors_avg[6]<adc_current[6]){
		 Error=(-2*500);
	}//black2,3
	if(sensors_avg[1]<adc_current[1]&&sensors_avg[2]>adc_current[2]&&sensors_avg[3]<adc_current[3]&&sensors_avg[4]<adc_current[4]
	&&sensors_avg[5]<adc_current[5]&&sensors_avg[6]<adc_current[6]){
		 Error=(-3*500);
	}//black2
	if(sensors_avg[1]>adc_current[1]&&sensors_avg[2]>adc_current[2]&&sensors_avg[3]<adc_current[3]&&sensors_avg[4]<adc_current[4]
	&&sensors_avg[5]<adc_current[5]&&sensors_avg[6]<adc_current[6]){
		 Error=(-4*500);
	}//black1,2
	if(sensors_avg[1]>adc_current[1]&&sensors_avg[2]<adc_current[2]&&sensors_avg[3]<adc_current[3]&&sensors_avg[4]<adc_current[4]
	&&sensors_avg[5]<adc_current[5]&&sensors_avg[6]<adc_current[6]){
		Error=(-5*500);
	}//black1
	P=Error;
	I=I+Error;
	D=Error-Lasterror;
	Lasterror=Error;
	float PID =(0.54*P)+(0*I)+(0.32*D); // last p = 0.4 last d = 0.4
	sprintf(&text1,"%.2f\r\n",(float)PID);
	HAL_UART_Transmit(&huart2,text1, strlen(text1),1000);
	L=650+PID;
	R=650-PID;
```

```
if(L>max&&R>max){
                        L=max;
                        R=max;
        }
        if(L<min&&R>max){
                        L=min;
                        R=max;
        }
        if(L>max&&R<min){
                        L=max;
                        R=min;
        }
        if(L<min&&R<min){
                        L=min;
                        R=min;
        }
        if(L>=0&&R>=0){
                        TIM3->CCR3=0;//backward.L
                        TIM3->CCR4=L;//forward.L
                        TIM3->CCR1=0;//backward.L
                        TIM3->CCR2=R;//forward.L
        }
        if(L>=0&&R<0){
                        TIM3->CCR3=0;//backward.L
                        TIM3->CCR4=L;//forward.L
                        TIM3->CCR1=-R;//backward.L
                        TIM3->CCR2=0;//forward.L
        }
        if(L<0&&R>=0){
                        TIM3->CCR3=-L;//backward.L
                        TIM3->CCR4=0;//forward.L
                        TIM3->CCR1=0;//backward.L
                        TIM3->CCR2=R;//forward.L
        }
        if(L<0&&R<0){
                        TIM3->CCR3=-L;//backward.L
                        TIM3->CCR4=0;//forward.L
                        TIM3->CCR1=-R;//backward.L
                        TIM3->CCR2=0;//forward.L
        }
    }
}
```

Result

Track A			Track B		
11.1	13.03	11.4	45.8	**38.35**	52.73

Video:

https://youtube.com/shorts/mmrjnHj__n0?feature=share

Working Percentage

Name	ID	%	Note	Sign
Krittin Sakharin	65011356	33.3	Design PBC	
Varis Saligupta	65011619	33.3	Soldering	
Akesit Akkharasaksiri	65110131	33.3	Programming	