

G16

Atomic



ສມາຈັກ

នັ້ນຈາ ພຣະມໂຍຮາ 2111311367

ນວພລ ສຸວະຮຣະນກູງກົມ 2111310021

ປົງລິພລ ເລີຄລໍ້າ 2111310997

ກັກຮພງຫຼີ ກັນວຳ 2111310773

ປລື້ມປີຕີ ອຣກໍຍ 2111310765

ຕະນຸນັ້ນທີ່ ໃຈດີ 2011311491

Overview

Requirement

ใช้บอร์ด STM32L152B เป็นพื้นฐานในการพัฒนารถที่สามารถติดตามเส้นทางที่กำหนดได้ โดยมีเซ็นเซอร์ตรวจจับสิ่งกีดขวางเพื่อป้องกันการชน หากพบสิ่งกีดขวางระบบจะทำการรีวอร์ดและหยุดเคลื่อนไหว เพื่อความปลอดภัยมากขึ้น อีกทั้งยังสามารถเปลี่ยนทิศทางการเคลื่อนไหวได้ตามความต้องการของผู้ใช้ ดังนั้น การสร้างรถแบบนี้จะช่วยเพิ่มประสิทธิภาพและความปลอดภัยในการเดินทางอย่างมีประสิทธิภาพมากยิ่งขึ้น

Specification

สร้างโปรแกรมให้กับไมโครคอนโทรลเลอร์ STM32L1 เพื่อควบคุมการทำงานของรถที่ใช้เซ็นเซอร์อินฟราเรดและเซ็นเซอร์อัลตราโซนิก (ultrasonic sensor) เพื่อตรวจจับเส้นทางและการหลบหลีกอุปสรรค โดยสร้างระบบควบคุมการทำงานของมอเตอร์ ส่วนหนึ่งของโค้ดยังใช้สำหรับการวัดระยะทางโดยใช้เซ็นเซอร์อัลตราโซนิก และทำการคำนวณระยะทางจากสัญญาณที่ได้มา

โดยแบ่งเป็นหลายไฟล์ดังนี้

FUNCTION

SystemClock_Config()

TIM_OC_GPIO_conf,

Configl293d_GPIO,

TIM_OC_Config,

Configl293d_2_GPIO

TIM_OC_GPIO_2_conf,

Timbase_conf,

TIM_OC_2_Config

Timbase_2_conf

SystemClock_Config()

- มีหน้าที่กำหนดค่าต่าง ๆ เพื่อเริ่มต้นระบบ โดยการตั้งค่านาฬิการะบบ (System Clock) และการตั้งค่าพลังงานสำหรับการเข้าถึงหน่วยความจำแฟลช เพื่อให้ระบบทำงานอย่างเสถียรและเชื่อถือได้

```

void SystemClock_Config(void){
    /* Enable ACC64 access and set FLASH latency */
    LL_FLASH_Enable64bitAccess();;

    LL_FLASH_SetLatency(LL_FLASH_LATENCY_1);

    /* Set Voltage scale1 as MCU will run at 32MHz */
    LL_APB1_GRP1_EnableClock(LL_APB1_GRP1_PERIPH_PWR);
    LL_PWR_SetRegulVoltageScaling(LL_PWR_REGU_VOLTAGE_SCALE1);

    /* Poll VOSF bit of in PWR_CSR. Wait until it is reset to 0 */
    while (LL_PWR_IsActiveFlag_VOSF() != 0)
    {
    };

    /* Enable HSI if not already activated*/
    if (LL_RCC_HSI_IsReady() == 0)
    {
        /* HSI configuration and activation */
        LL_RCC_HSI_Enable();

        while(LL_RCC_HSI_IsReady() != 1)
        {
        };
    }

    /* Main PLL configuration and activation */
    LL_RCC_PLL_ConfigDomain_SYS(LL_RCC_PLLSOURCE_HSI, LL_RCC_PLL_MUL_6, LL_RCC_PLL_DIV_3);

    LL_RCC_PLL_Enable();
    while(LL_RCC_PLL_IsReady() != 1)
    {
    };

    /* Sysclk activation on the main PLL */
    LL_RCC_SetAHBPrescaler(LL_RCC_SYSCLK_DIV_1);
    LL_RCC_SetSysClkSource(LL_RCC_SYS_CLKSOURCE_PLL);
    while(LL_RCC_GetSysClkSource() != LL_RCC_SYS_CLKSOURCE_STATUS_PLL)
    {
    };

    /* Set APB1 & APB2 prescaler*/
    LL_RCC_SetAPB1Prescaler(LL_RCC_APB1_DIV_1);
    LL_RCC_SetAPB2Prescaler(LL_RCC_APB2_DIV_1);

    /* Set systick to 1ms in using frequency set to 32MHz */
    /* This frequency can be calculated through LL RCC macro */
    /* ex: __LL_RCC_CALC_PLLCLK_FREQ (HSI_VALUE, LL_RCC_PLL_MUL_6, LL_RCC_PLL_DIV_3); */
    LL_Init1msTick(32000000);

    /* Update CMSIS variable (which can be updated also through SystemCoreClockUpdate function) */
    LL_SetSystemCoreClock(32000000);
}

```

Configl293d_GPIO, Configl293d_2_GPIO

ใช้สำหรับกำหนดการตั้งค่าของพอร์ต GPIO ที่ใช้สำหรับควบคุมมอเตอร์ โดยกำหนดให้มีการทำงานของขา GPIO เป็นโหมด OUTPUT และตั้งค่า Output Type เป็น PUSH-PULL และค่าอื่น ๆ ตามความเหมาะสม

```
***** IR Config *****

void IR_Config(void) {
    LL_GPIO_InitTypeDef IR_Config;
    LL_AHB1_GRP1_EnableClock(LL_AHB1_GRP1_PERIPH_GPIOA);
    LL_AHB1_GRP1_EnableClock(LL_AHB1_GRP1_PERIPH_GPIOD);

    IR_Config.Mode = LL_GPIO_MODE_INPUT;
    IR_Config.OutputType = LL_GPIO_OUTPUT_PUSHPULL;
    IR_Config.Speed = LL_GPIO_SPEED_FREQ_VERY_HIGH;
    IR_Config.Pull = LL_GPIO_PULL_NO;
    IR_Config.Pin = LL_GPIO_PIN_3;      // set IR1 pin to PB3
    LL_GPIO_Init(GPIOA, &IR_Config);

    IR_Config.Pin = LL_GPIO_PIN_2;      // set IR2 pin to PB4
    LL_GPIO_Init(GPIOD, &IR_Config);
}

void IR_Check(void) {
    IR1 = LL_GPIO_IsInputPinSet(GPIOA, LL_GPIO_PIN_3);
    IR2 = LL_GPIO_IsInputPinSet(GPIOD, LL_GPIO_PIN_2);
}
```

Timbase_conf, Timbase_2_conf

ใช้สำหรับกำหนดค่าเริ่มต้นสำหรับตัวจับเวลา (Timer) กี่ใช้สำหรับควบคุม PWM ของมอเตอร์ โดยกำหนดโหมดการทำงานของ Timer การแบ่งคล็อก (Clock Division) และการตั้งค่าอื่น ๆ ตามความเหมาะสม

```
void Timbase_conf(void)
{
    LL_TIM_InitTypeDef timbase_initstructure;

    LL_APB1_GRP1_EnableClock(LL_APB1_GRP1_PERIPH_TIM3);
    //Time-base configure
    timbase_initstructure.ClockDivision = LL_TIM_CLOCKDIVISION_DIV2;
    timbase_initstructure.CounterMode = LL_TIM_COUNTERMODE_UP;
    timbase_initstructure.Autoreload = 100 - 1;
    timbase_initstructure.Prescaler = 16000 - 1;
    LL_TIM_Init(TIM3, &timbase_initstructure);

    LL_TIM_EnableCounter(TIM3);
}
```

```
void Timbase_2_conf(void){
    LL_TIM_InitTypeDef timbase_initstructure;

    LL_APB1_GRP1_EnableClock(LL_APB1_GRP1_PERIPH_TIM2);
    //Time-base configure
    timbase_initstructure.ClockDivision = LL_TIM_CLOCKDIVISION_DIV2;
    timbase_initstructure.CounterMode = LL_TIM_COUNTERMODE_UP;
    timbase_initstructure.Autoreload = 100 - 1;
    timbase_initstructure.Prescaler = 16000 - 1;
    LL_TIM_Init(TIM2, &timbase_initstructure);

    LL_TIM_EnableCounter(TIM2);
}
```

ฝังก์ชันเหล่านี้ใช้ในการกำหนดการตั้งค่าของตัวควบคุมการเอาท์พุตของ
Timer สำหรับควบคุมความเร็วของมอเตอร์

`TIM_OC_GPIO_conf, TIM_OC_Config`
`TIM_OC_GPIO_2_conf, TIM_OC_2_Config`

```

void TIM_OC_GPIO_conf(void)
{
    LL_GPIO_InitTypeDef gpio_initstructure;

    LL_AHB1_GRP1_EnableClock(LL_AHB1_GRP1_PERIPH_GPIOB);

    gpio_initstructure.Mode = LL_GPIO_MODE_ALTERNATE;
    gpio_initstructure.Alternate = LL_GPIO_AF_2;
    gpio_initstructure.OutputType = LL_GPIO_OUTPUT_PUSHPULL;
    gpio_initstructure.Pin = LL_GPIO_PIN_4;
    gpio_initstructure.Pull = LL_GPIO_PULL_NO;
    gpio_initstructure.Speed = LL_GPIO_SPEED_FREQ_VERY_HIGH;
    LL_GPIO_Init(GPIOB, &gpio_initstructure);
}

```

```

void TIM_OC_Config(void)
{
    Timbase_conf();
    TIM_OC_GPIO_conf();
    LL_TIM_OC_InitTypeDef tim_oc_initstructure;

    tim_oc_initstructure.OCState = LL_TIM_OCSTATE_DISABLE;
    tim_oc_initstructure.OCMode = LL_TIM_OCMODE_PWM1;
    tim_oc_initstructure.OCPolarity = LL_TIM_OCPOLARITY_HIGH;
    tim_oc_initstructure.CompareValue = 100; //change duty cycle here
    LL_TIM_OC_Init(TIM3, LL_TIM_CHANNEL_CH1, &tim_oc_initstructure);

    /*Start Output Compare in PWM Mode*/
    LL_TIM_CC_EnableChannel(TIM3, LL_TIM_CHANNEL_CH1);
    LL_TIM_EnableCounter(TIM3);
}

```

```
void TIM_OC_GPIO_2_conf(void){  
    LL_GPIO_InitTypeDef gpio_initstructure;  
  
    LL_AHB1_GRP1_EnableClock(LL_AHB1_GRP1_PERIPH_GPIOB);  
  
    gpio_initstructure.Mode = LL_GPIO_MODE_ALTERNATE;  
    gpio_initstructure.Alternate = LL_GPIO_AF_1;  
    gpio_initstructure.OutputType = LL_GPIO_OUTPUT_PUSHPULL;  
    gpio_initstructure.Pin = LL_GPIO_PIN_10;  
    gpio_initstructure.Pull = LL_GPIO_PULL_NO;  
    gpio_initstructure.Speed = LL_GPIO_SPEED_FREQ_VERY_HIGH;  
    LL_GPIO_Init(GPIOB, &gpio_initstructure);  
}
```

```
void TIM_OC_2_Config(void){  
    Timbase_conf();  
    TIM_OC_GPIO_conf();  
    LL_TIM_OC_InitTypeDef tim_oc_initstructure;  
  
    tim_oc_initstructure.OCState = LL_TIM_OCSTATE_DISABLE;  
    tim_oc_initstructure.OCMode = LL_TIM_OCMODE_PWM1;  
    tim_oc_initstructure.OCPolarity = LL_TIM_OCPOLARITY_HIGH;  
    tim_oc_initstructure.CompareValue = 100; //change duty cycle here  
    LL_TIM_OC_Init(TIM2, LL_TIM_CHANNEL_CH3, &tim_oc_initstructure);  
  
    /*Start Output Compare in PWM Mode*/  
    LL_TIM_CC_EnableChannel(TIM2, LL_TIM_CHANNEL_CH3);  
    LL_TIM_EnableCounter(TIM2);  
}
```

Utr_GPIO_Config, TIMx_IC_Config:

ฟังก์ชันเหล่านี้ใช้ในการกำหนดค่า GPIO และ Timer สำหรับการต่อตัววัด
ระยะทางแบบอัลตราโซนิก

```

void Utr_GPIO_Config(void){
    LL_GPIO_InitTypeDef timic_gpio;
    LL_AHB1_GRP1_EnableClock(LL_AHB1_GRP1_PERIPH_GPIOA);

    //Config PA2 to input
    timic_gpio.Mode = LL_GPIO_MODE_ALTERNATE;
    timic_gpio.Pull = LL_GPIO_PULL_NO;
    timic_gpio.Pin = LL_GPIO_PIN_2;
    timic_gpio.OutputType = LL_GPIO_OUTPUT_PUSHPULL;
    timic_gpio.Speed = LL_GPIO_SPEED_FREQ_VERY_HIGH;
    timic_gpio.Alternate = LL_GPIO_AF_3;

    LL_GPIO_Init(GPIOA, &timic_gpio);

    //Config PA1
    timic_gpio.Mode = LL_GPIO_MODE_OUTPUT;
    timic_gpio.Pin = LL_GPIO_PIN_1;

    LL_GPIO_Init(GPIOA, &timic_gpio);
}

```

```

void TIMx_IC_Config(void){
    //Config IC Mode
    LL_TIM_IC_InitTypeDef timic;
    LL_APB2_GRP1_EnableClock(LL_APB2_GRP1_PERIPH_TIM9);

    timic.ICActiveInput = LL_TIM_ACTIVEINPUT_DIRECTTI;
    timic.ICFilter = LL_TIM_IC_FILTER_FDIV1_N2;
    timic.ICPolarity = LL_TIM_IC_POLARITY_BOTHEDGE; //capture both edge
    timic.ICPrescaler = LL_TIM_ICPSC_DIV1;
    LL_TIM_IC_Init(TIM9, LL_TIM_CHANNEL_CH1, &timic); // PA1 use timer2 channel2

    NVIC_SetPriority(TIM9_IRQn, 0);

    NVIC_EnableIRQ(TIM9_IRQn);
    LL_TIM_EnableIT_CC1(TIM9);
    LL_TIM_CC_EnableChannel(TIM9, LL_TIM_CHANNEL_CH1);

    LL_TIM_EnableCounter(TIM9);
}

```

`speaker_TIM_BASE_Config`, `speaker_TIM_OC_GPIO_Config`,
`speaker_TIM_OC_Config`:

ฟังก์ชันเหล่านี้ใช้ในการกำหนดการตั้งค่าของตัวควบคุมเวลาและ GPIO
สำหรับสร้างสัญญาณเสียง

```
void speaker_TIM_BASE_Config(uint16_t ARR){  
    //config Time base  
    LL_TIM_InitTypeDef timbase_initstructure;  
    //Enable clock for TIM4  
    LL_APB1_GRP1_EnableClock(LL_APB1_GRP1_PERIPH_TIM4);  
  
    timbase_initstructure.ClockDivision = LL_TIM_CLOCKDIVISION_DIV1;  
    timbase_initstructure.CounterMode = LL_TIM_COUNTERMODE_UP;  
    timbase_initstructure.Autoreload = ARR - 1;  
    timbase_initstructure.Prescaler = TIMx_PSC - 1;  
  
    //use TIM4      for timer  
    LL_TIM_Init(TIM4, &timbase_initstructure);  
  
    //TIM4 start count  
    LL_TIM_EnableCounter(TIM4);  
}
```

```
void speaker_TIM_OC_GPIO_Config(void){  
    // Declare struct for GPIO config  
    LL_GPIO_InitTypeDef gpio_initstructure;  
    // Enable GPIO clock  
    LL_AHB1_GRP1_EnableClock(LL_AHB1_GRP1_PERIPH_GPIOB);  
  
    //config PB6 to OUTPUT in ALTERNATE mode (AF2)  
    gpio_initstructure.Mode = LL_GPIO_MODE_ALTERNATE;  
    gpio_initstructure.Alternate = LL_GPIO_AF_2;  
    gpio_initstructure.OutputType = LL_GPIO_OUTPUT_PUSHPULL;  
    gpio_initstructure.Pin = LL_GPIO_PIN_6;  
    gpio_initstructure.Pull = LL_GPIO_PULL_NO;  
    gpio_initstructure.Speed = LL_GPIO_SPEED_FREQ VERY_HIGH;  
  
    // Write configuration to GPIO register  
    LL_GPIO_Init(GPIOB, &gpio_initstructure);  
}
```

```
void speaker_TIM_OC_Config(uint16_t note){
    LL_TIM_OC_InitTypeDef tim_oc_initstructure;

    speaker_TIM_OC_GPIO_Config(); //SET OUTPUT PIN
    speaker_TIM_BASE_Config(note); //SETTING and USING TIM-BASE

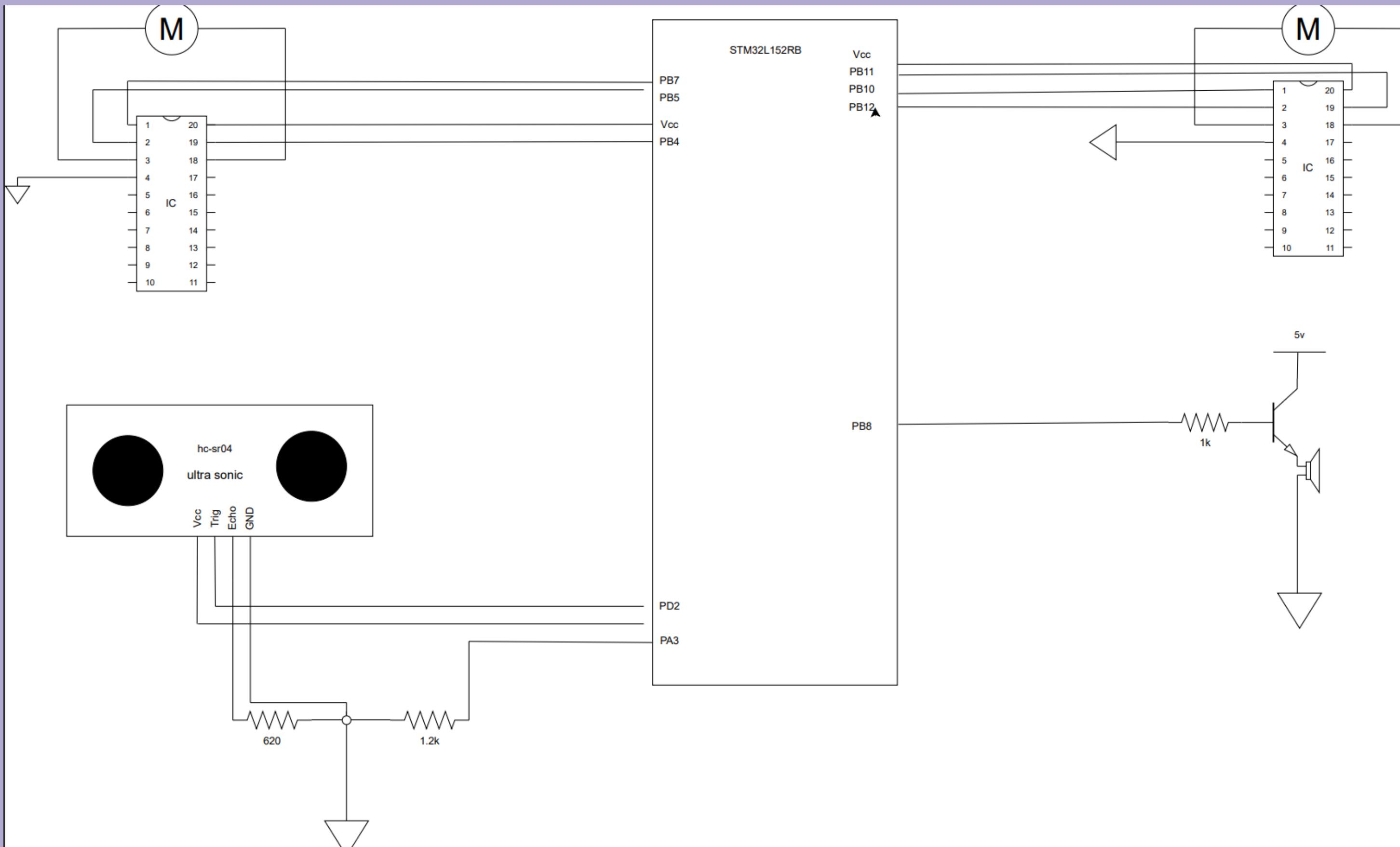
    tim_oc_initstructure.OCState = LL_TIM_OCSTATE_DISABLE;
    tim_oc_initstructure.OCMode = LL_TIM_OCMODE_PWM1;
    tim_oc_initstructure.OCPolarity = LL_TIM_OCPOLARITY_HIGH;
    tim_oc_initstructure.CompareValue = LL_TIM_GetAutoReload(TIM4) / 2;
    LL_TIM_OC_Init(TIM4, LL_TIM_CHANNEL_CH1, &tim_oc_initstructure);

    /*Interrupt Configure*/
    NVIC_SetPriority(TIM4_IRQn, 1);
    NVIC_EnableIRQ(TIM4_IRQn);
    LL_TIM_EnableIT_CC1(TIM4);

    /*Start Output Compare in PWM Mode*/
    LL_TIM_CC_EnableChannel(TIM4, LL_TIM_CHANNEL_CH1);
    LL_TIM_EnableCounter(TIM4);
}
```

Architectural design

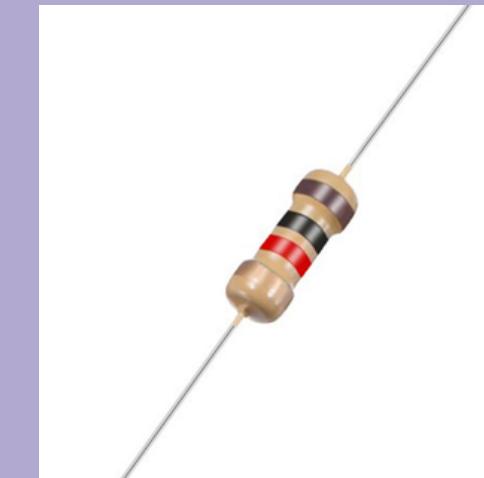
integrator test



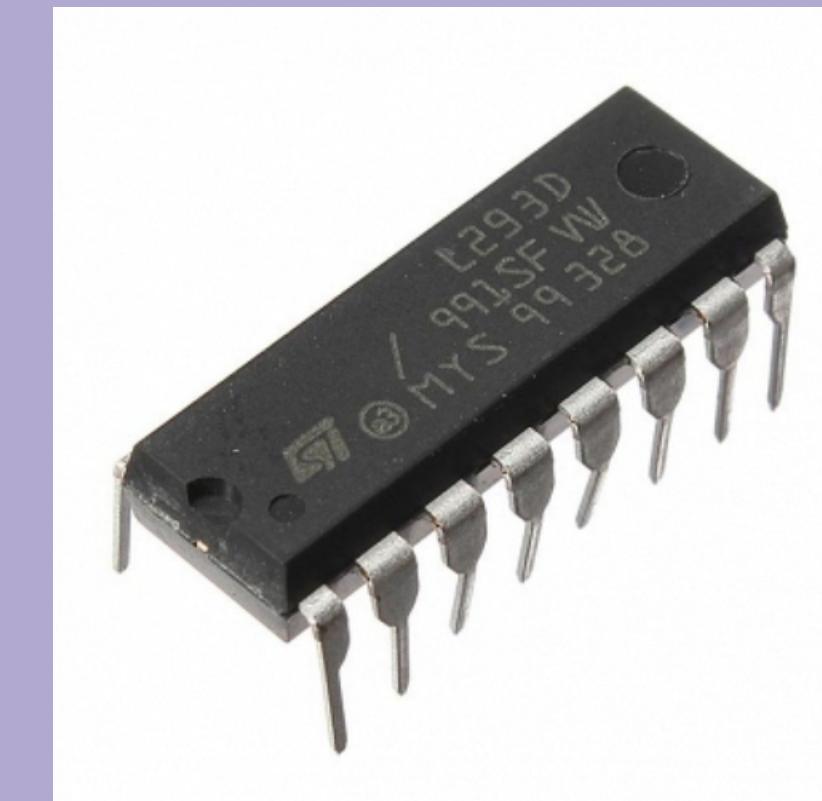
Hardware detail selection



STM32L152RB



resistor 630 1k 1,2k ohm



l293d motor driver ic



Ultrasonic HC-04 SR



infared sensor



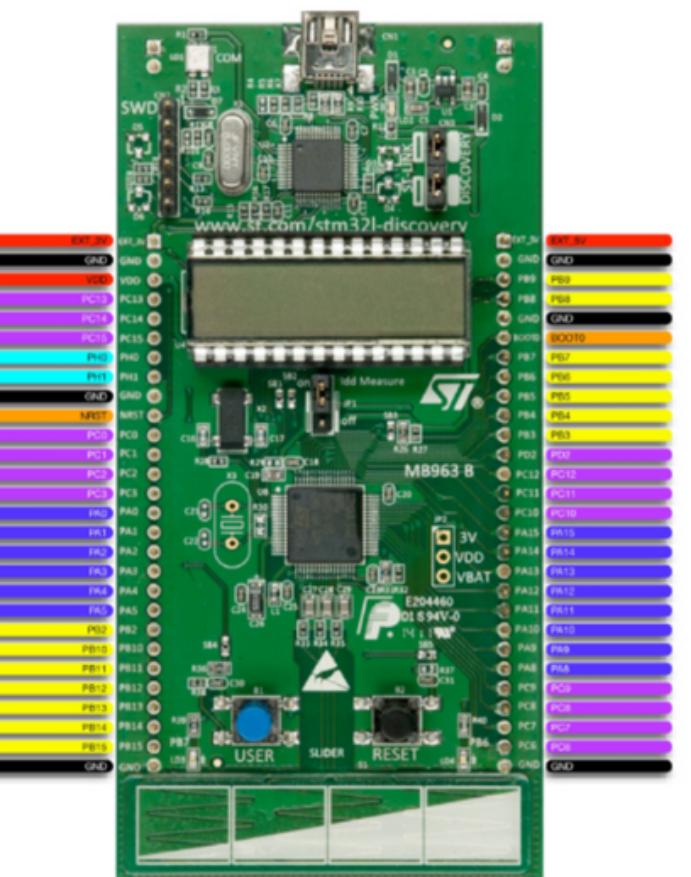
speaker



motor gear DC

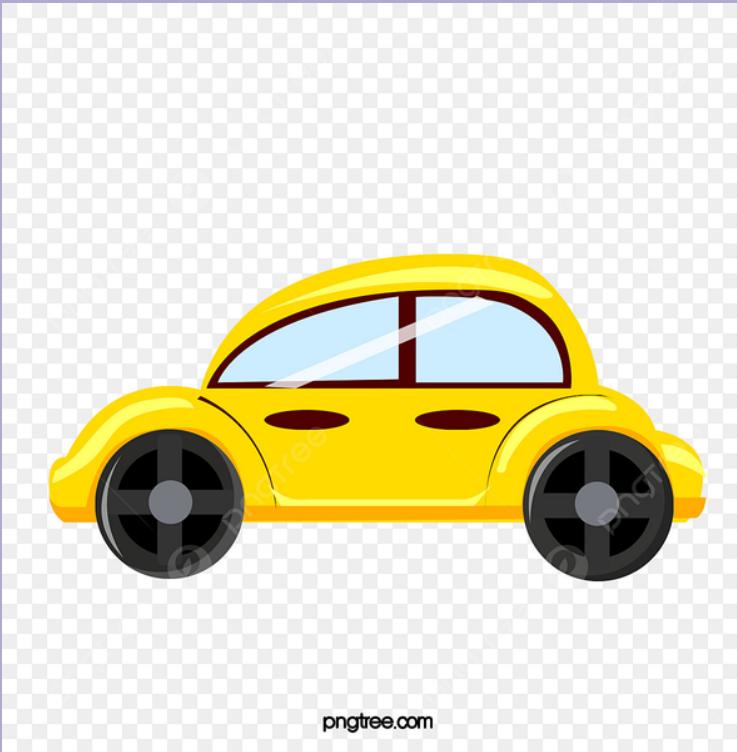
STM32 Pin allocation

Used Pin	
EXT_3V	
GND	
VDD	
PC13	
PC14	
PC15	
PH0	
PH1	
GND	
NRST	
PC0	
PC1	
PC2	
PC3	
PA0	
PA1	trigger pin
PA2	echo pin
PA3	IR1
PA4	
PA5	
PB2	
PB10	Motor2
PB11	Motor2
PB12	Motor2
PB13	
PB14	
PB15	
GND	



Used Pin	
EXT_5V	
GND	
PB9	
PB8	
GND	
BOOT0	
PB7	Motor1
PB6	speaker output
PB5	Motor1
PB4	Motor1
PB3	
PD2	IR2
PC12	
PC11	
PC10	
PA15	
PA14	
PA13	
PA12	
PA11	
PA10	
PA9	
PA8	
PC9	
PC8	
PC7	
PC6	
GND	

design



pngtree.com



sensor



speaker

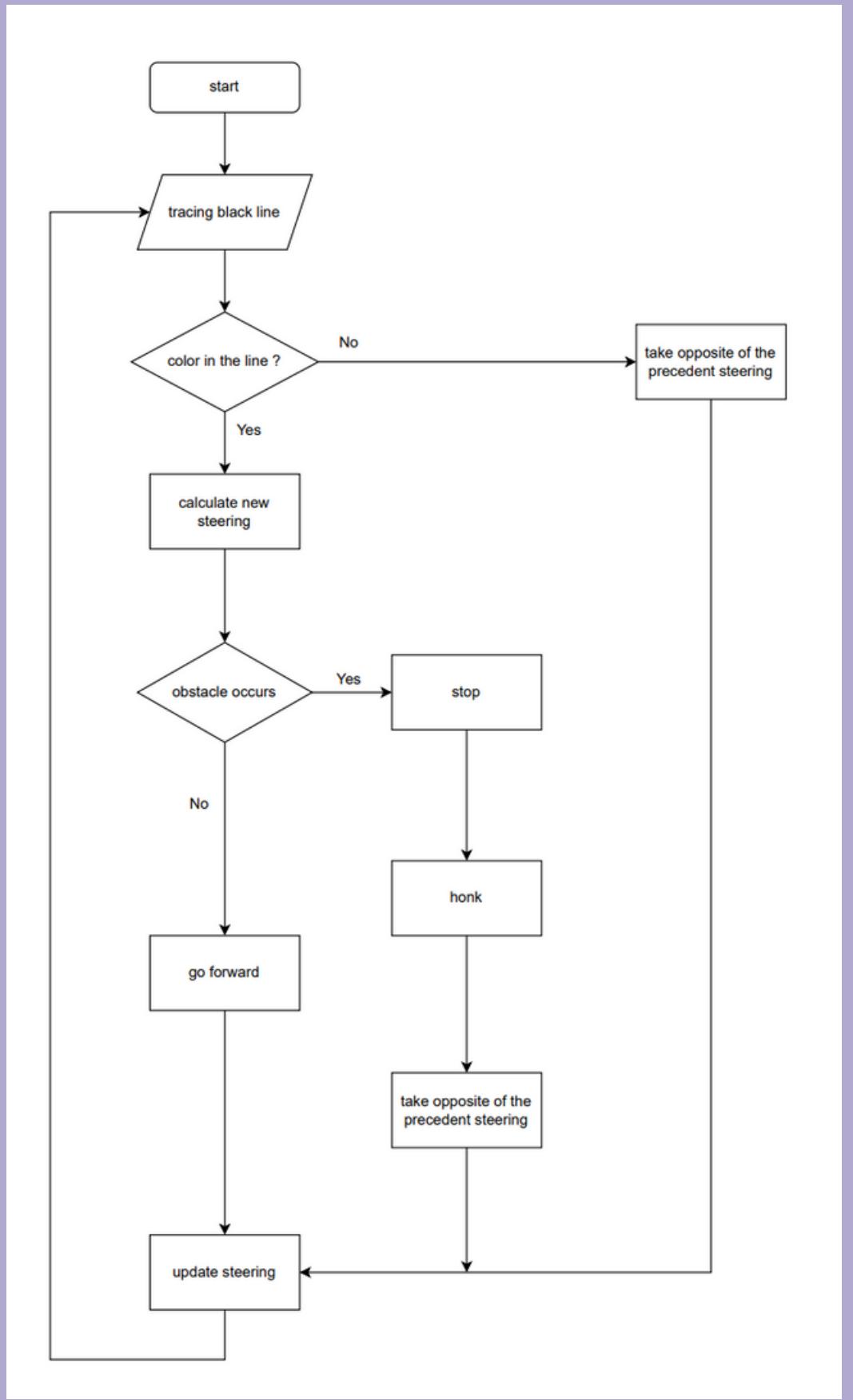


tracking

The background features a large, semi-transparent circle filled with a pink-to-purple gradient. This circle overlaps with several smaller, irregularly shaped blobs of the same color palette. The overall effect is organic and minimalist.

Detail design

flowchart



Grantt chart

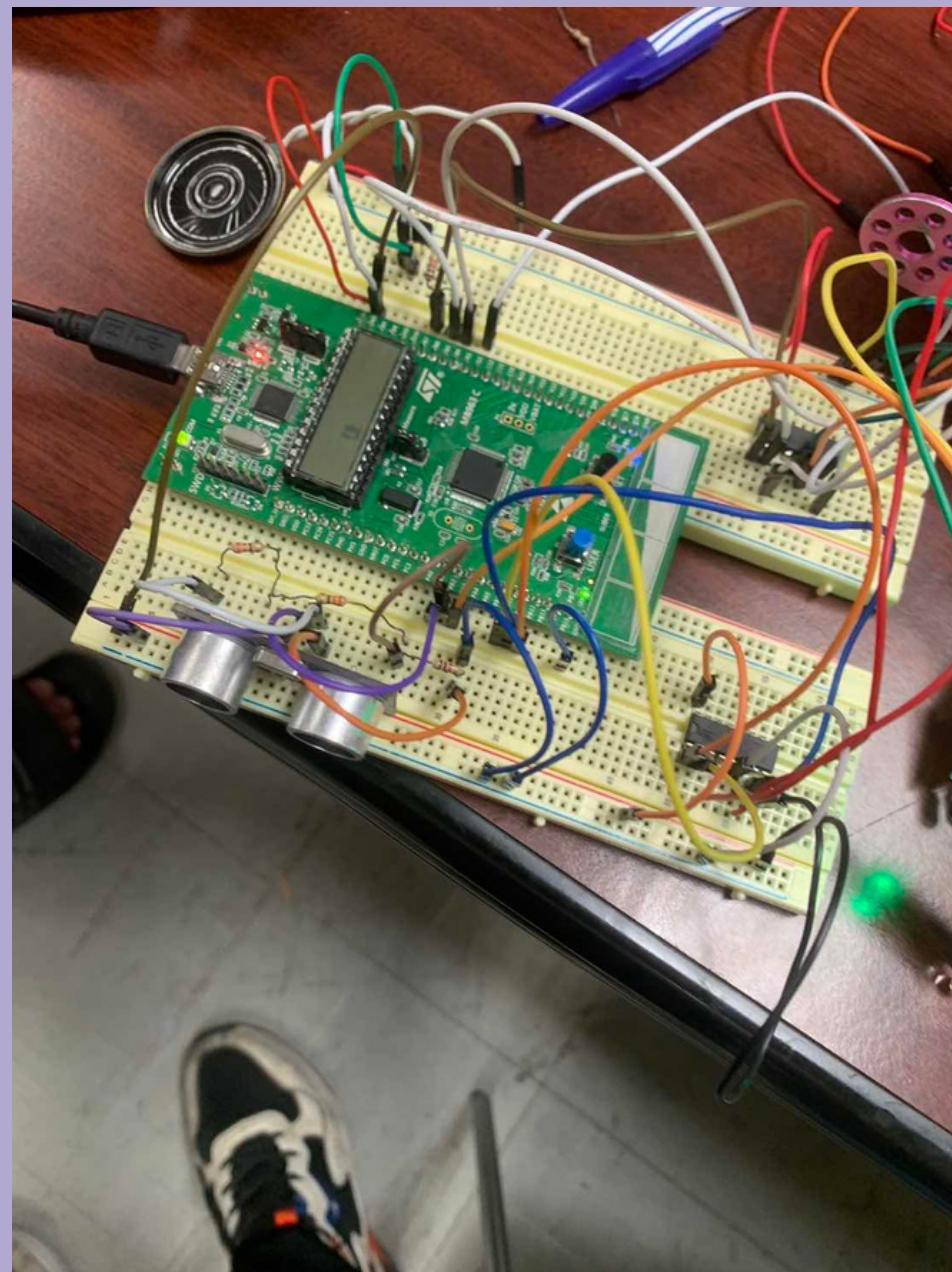
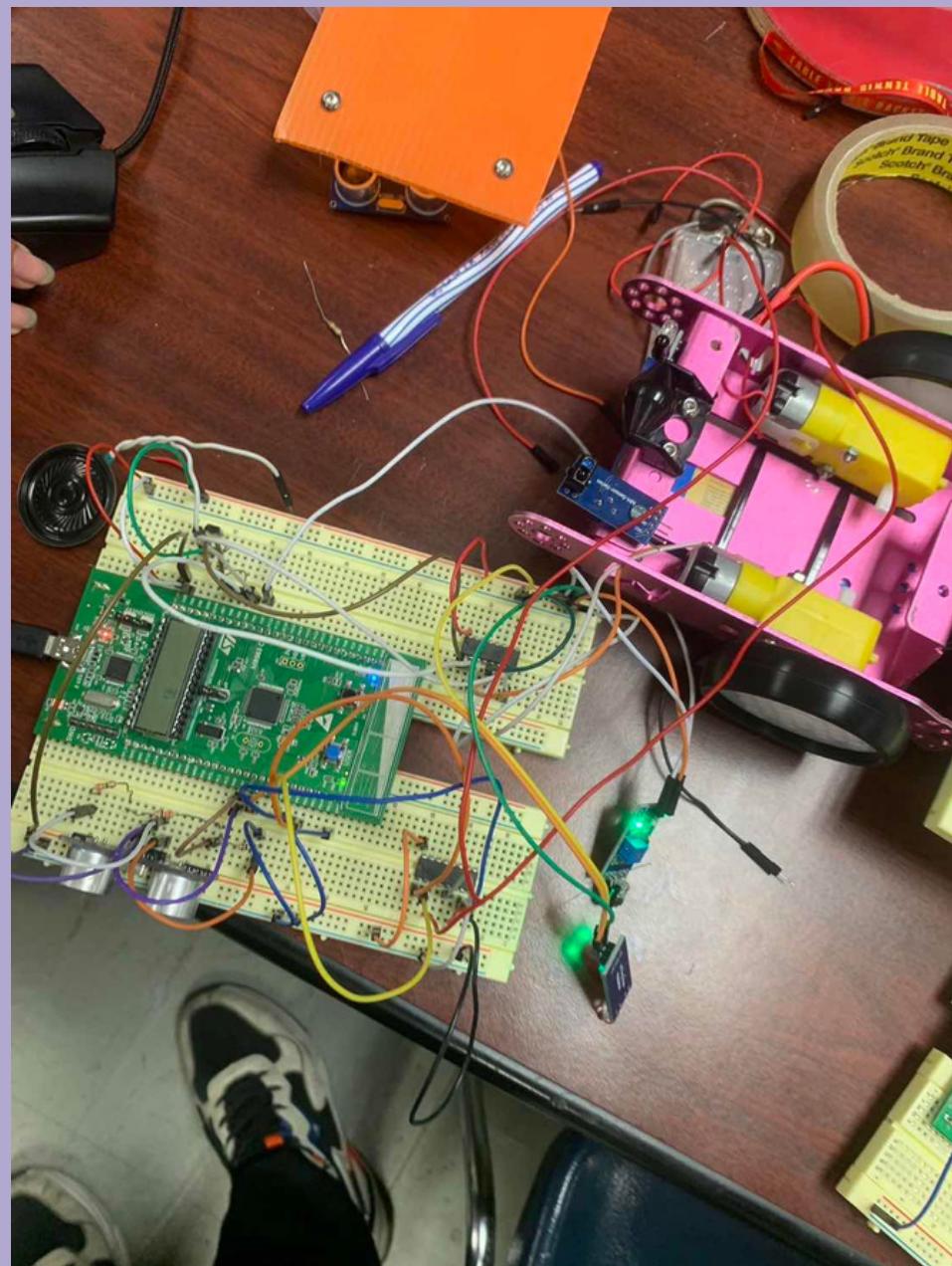
WBS NUMBER	TASK TITLE	TASK OWNER	START DATE	DUE DATE	DURATION	PCT OF TASK COMPLETE	PHASE ONE						PHASE TWO						PHASE THREE							
							WEEK 1			WEEK 2			WEEK 3			WEEK 4			WEEK 5			WEEK 6				
							M	T	W	R	F	M	T	W	R	F	M	T	W	R	F	M	T	W	R	F
1	Project Conception																									
1.1	คิดหัวข้อและประชุม	Everyone	12/20/23	12/21/23	1	100%																				
1.1.1	ออกแบบ		12/21/23	12/22/23	1	100%																				
1.2	ค้นคว้าข้อมูล		12/22/23	12/25/23	3	100%																				
1.3	รายละเอียดอุปกรณ์		12/24/23	12/31/23	7	100%																				
1.4	ทำเอกสาร		1/1/24	2/14/24	43	100%																				
1.5	แนงหน้าที่		1/2/24	1/2/24	0	100%																				
2	Update Project Plan																									
2.1	ส่งเอกสาร		1/15/24	1/15/24	0	100%																				
2.2	update presentation		1/16/24	1/16/24	0	100%																				
3	Progression																									
3.1	เตรียมอุปกรณ์		1/16/24	2/10/24	24	100%																				
3.2	ทำระบบultra sonic		1/16/24	2/15/24	29	100%																				
3.2.1	ทำระบบควบคุมมอเตอร์		2/3/24	2/15/24	12	100%																				
3.2.2	ทำระบบจับเส้นด้วย IR		2/3/24	2/15/24	12	100%																				
3.3	ทำระบบแคร์		2/3/24	2/15/24	12	100%																				
4	Final																									
4.1	ทำ slide		2/10/24	2/14/24	4	100%																				
4.2	ซ้อมพิธีเข็นต์		2/10/24	2/14/24	4	100%																				
4.3	Submit all file		2/11/24	2/14/24	3	100%																				
4.4	Presentation		2/12/24	2/14/24	2	0%																				

Demo video/ project demo

Demo video







Problem & Solution

Conclusion

Thankyou