实　验　报　告

实验课程：　《电子工艺实训》

实验项目：　单片机小车设计与制作

系　　别：　计算机科学与技术

年　　级：　2022

学生姓名：　陈瑾

学生学号：　37220222203552

实验日期：　2024　年　7　月　8　日--　19　日

电子工艺实训实验报告

姓名：陈瑾　　学号：37220222203552　　总分：

一、PCB 制作部分

1. 简述 PCB 设计过程及遇到的主要问题，并附上原理图、元件库、PCB、封装、丝印的截图。

PCB 设计过程：

1. 绘制原理图：在 EDA 软件中创建或选择现有的元器件符号，确保符号正确反映器件的功能和引脚配置。

2. 封装：为每个器件设计或选择合适的 PCB 封装。封装应与实际器件的尺寸和引脚排列相匹配。

3. 绘制 PCB：原理图转 PCB 文件，并根据电气性能和实际安装需求调整器件位置。

4. 布线：根据电路的连接关系，进行布线操作，连接各个器件的引脚。

5. 布线优化和检查：对布线进行优化，并进行网络检查和 DRC 检查。

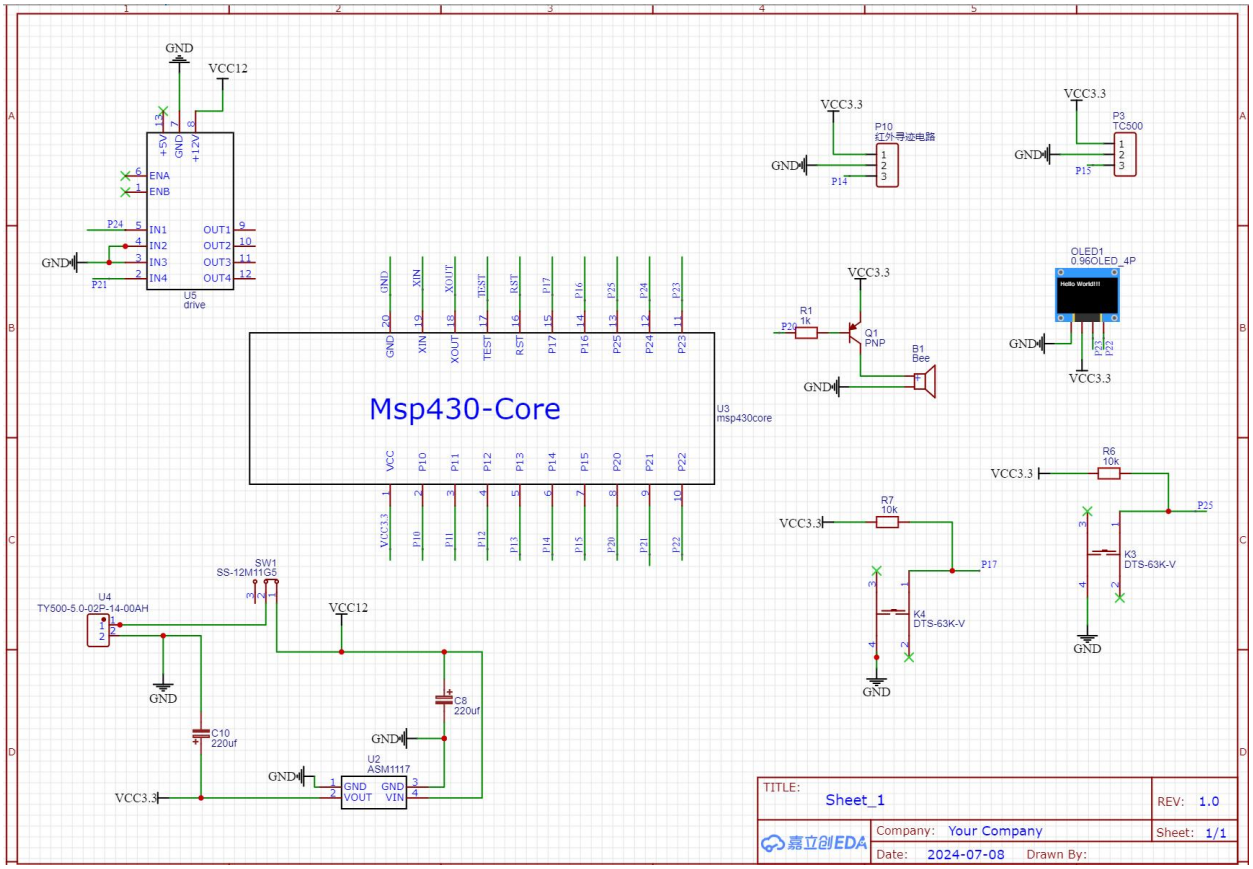
6. 铺铜，丝印：在 PCB 上铺设铜皮，设计丝印。

7. 最后生产：完成设计后，生成制造文件，并提交给制造商进行生产。

遇到的问题：

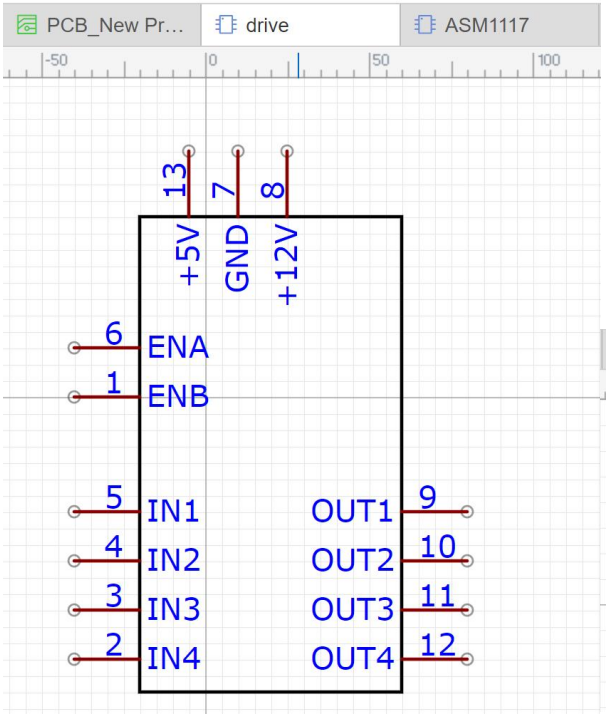
器件引脚设置错误，封装错误。

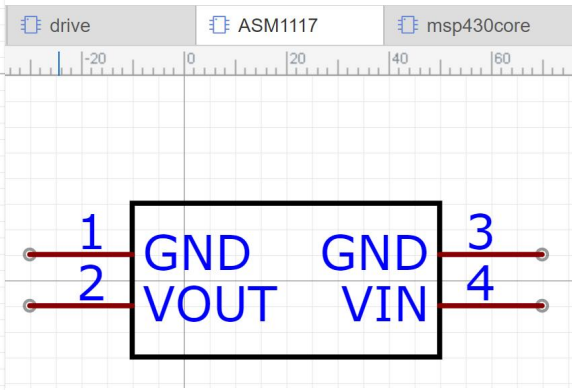
原理图：

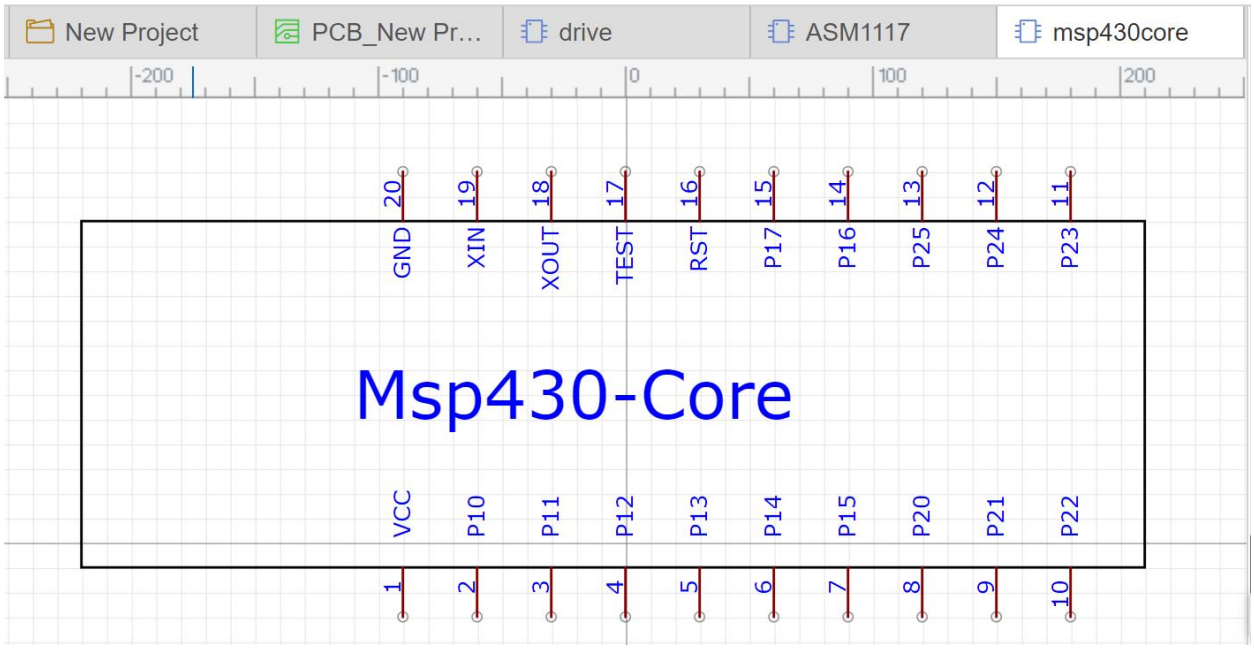


元件库：

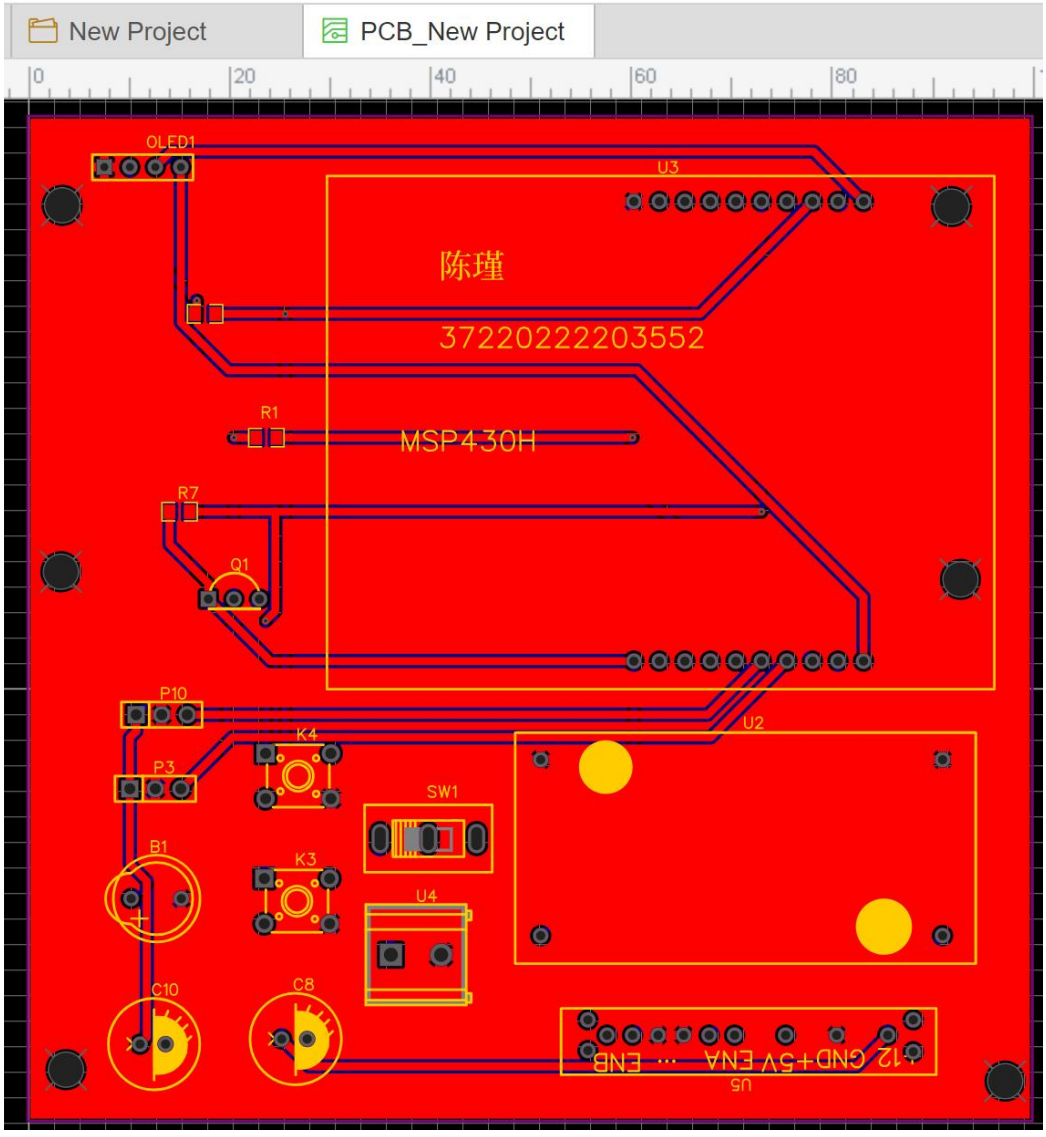
标题(零件名称)	所有者
drive	happycj
ASM1117	happycj
msp430core	happycj

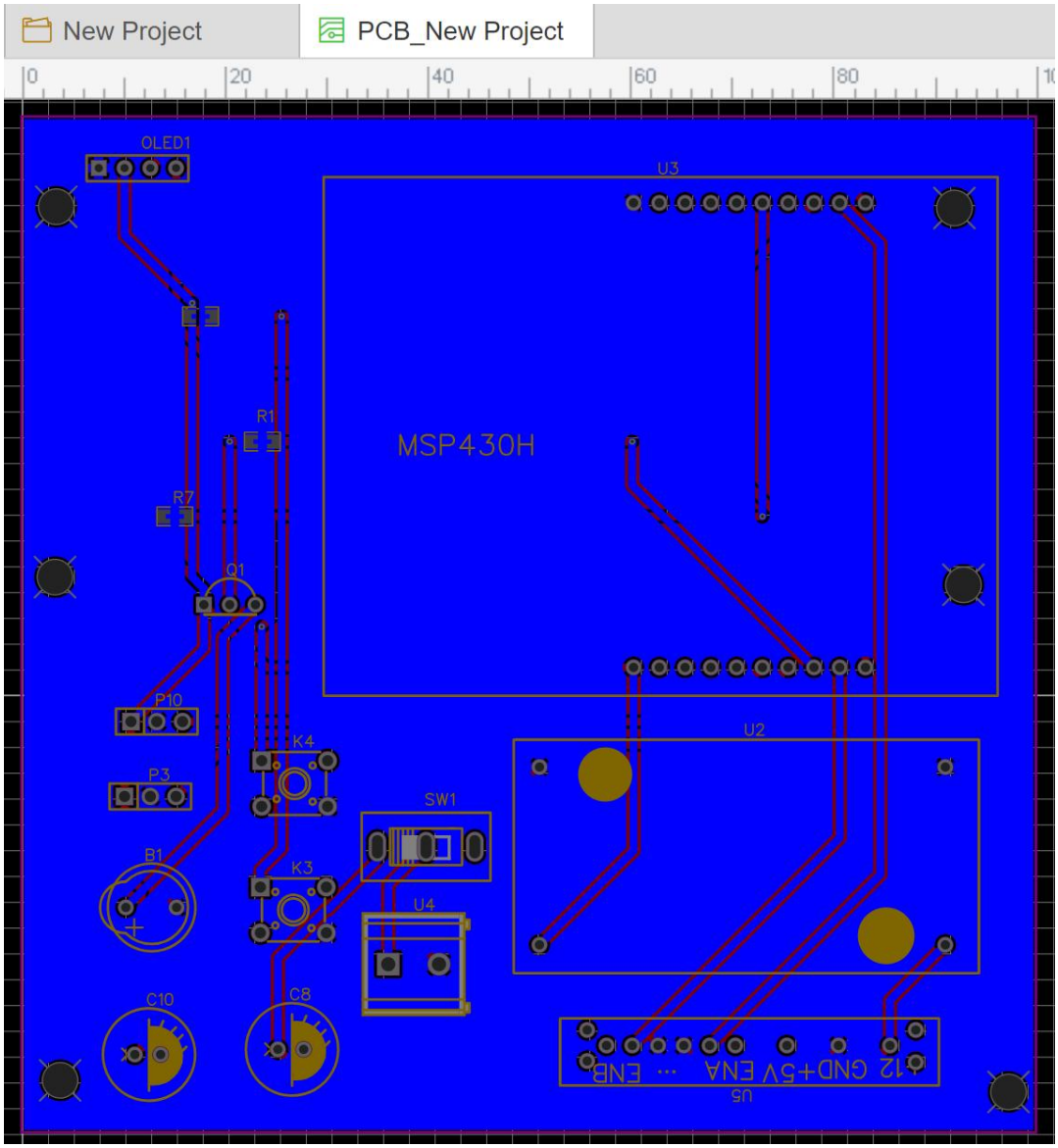




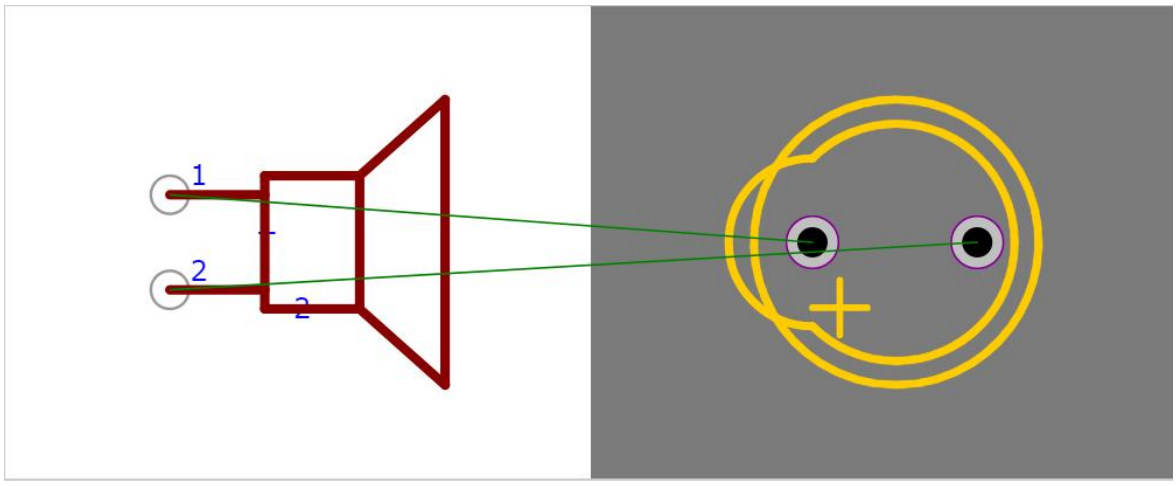

New Project
PCB_New Pr...
drive
ASM1117
msp430core
-200
-100
0
100
200
20
19
18
17
16
15
14
13
12
11
GND
XIN
XOUT
TEST
RST
P17
P16
P25
P24
P23
Msp430-Core
VCC
P10
P11
P12
P13
P14
P15
P20
P21
P22
1
2
3
4
5
6
7
8
9
10

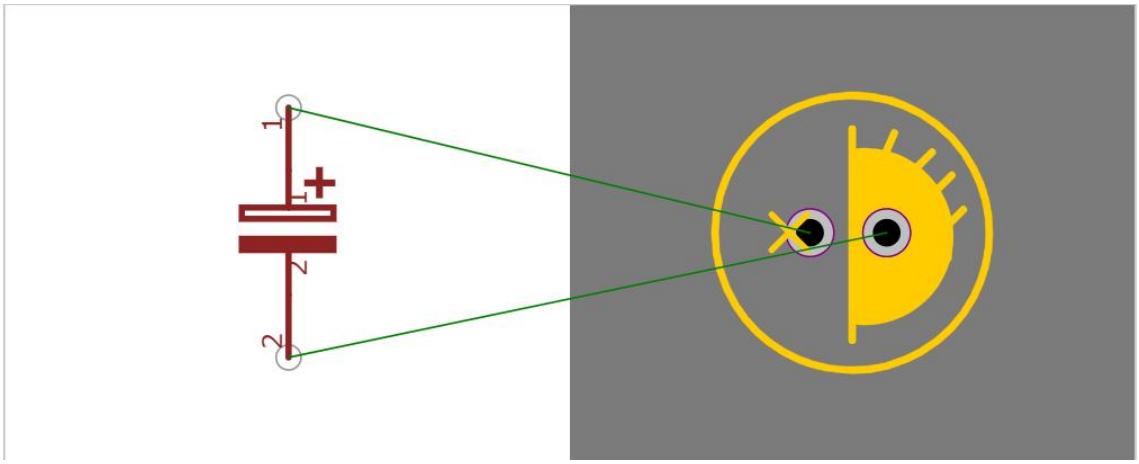
PCB：

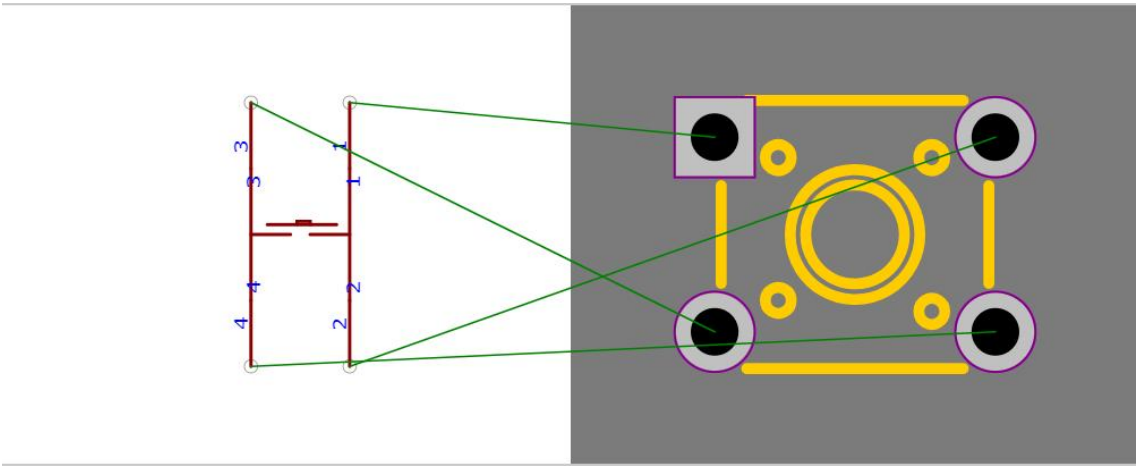

New Project
PCB_New Project
0
20
40
60
80
OLED1
U3
陈瑾
37220222203552
R1
MSP430H
R7
Q1
P10
K4
P3
U2
SW1
B1
K3
U4
C10
C8
U5

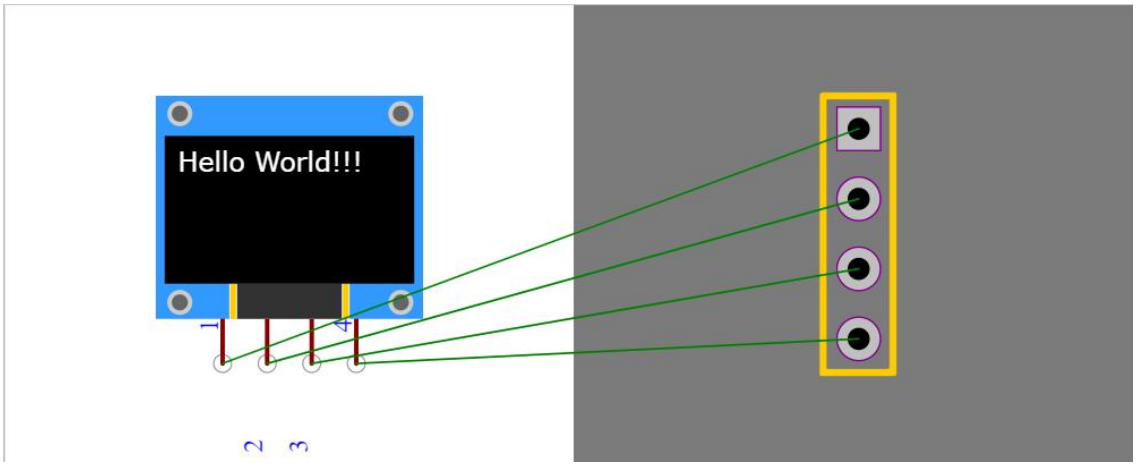

New Project
PCB_New Project
OLED1
U3
MSP430H
R1
R7
Q1
P10
K4
P3
SW1
B1
K3
U4
U2
C10
C8
U5

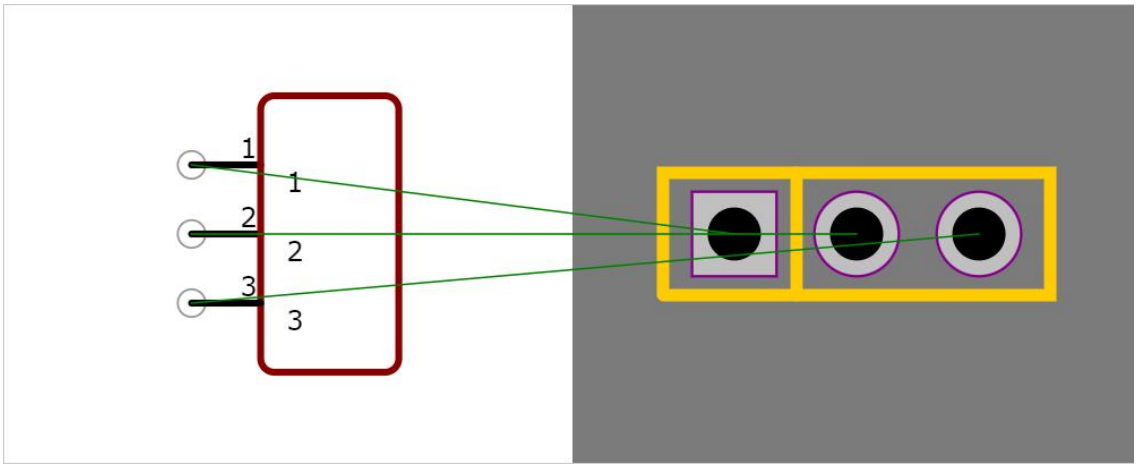
封装：

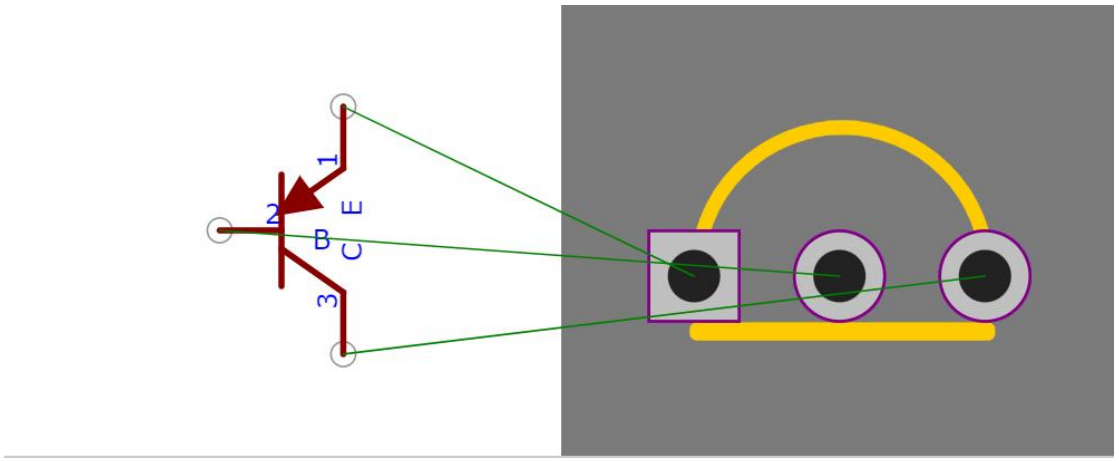

1
2
1
2

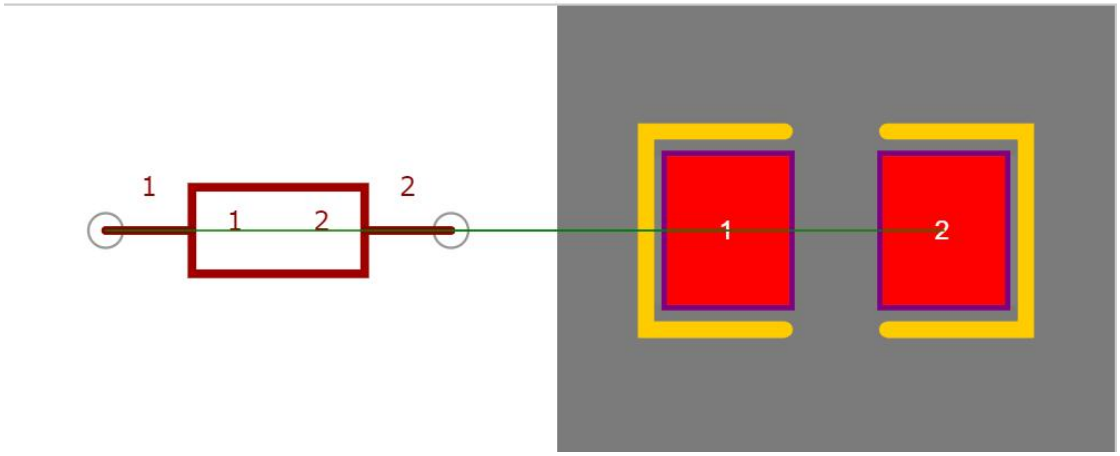

1
2
2
1

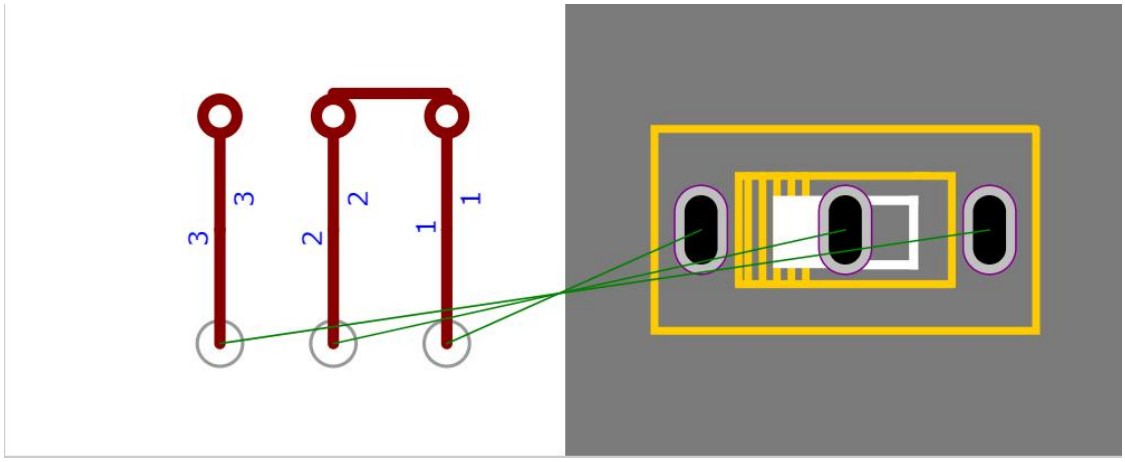

3
3
4
4
1
1
2
2

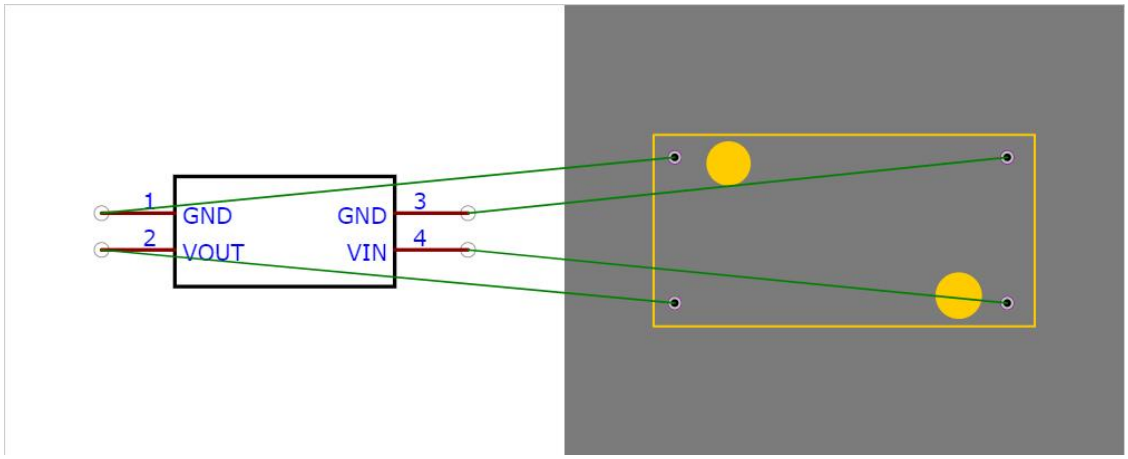

Hello World!!!
1
4
2
3

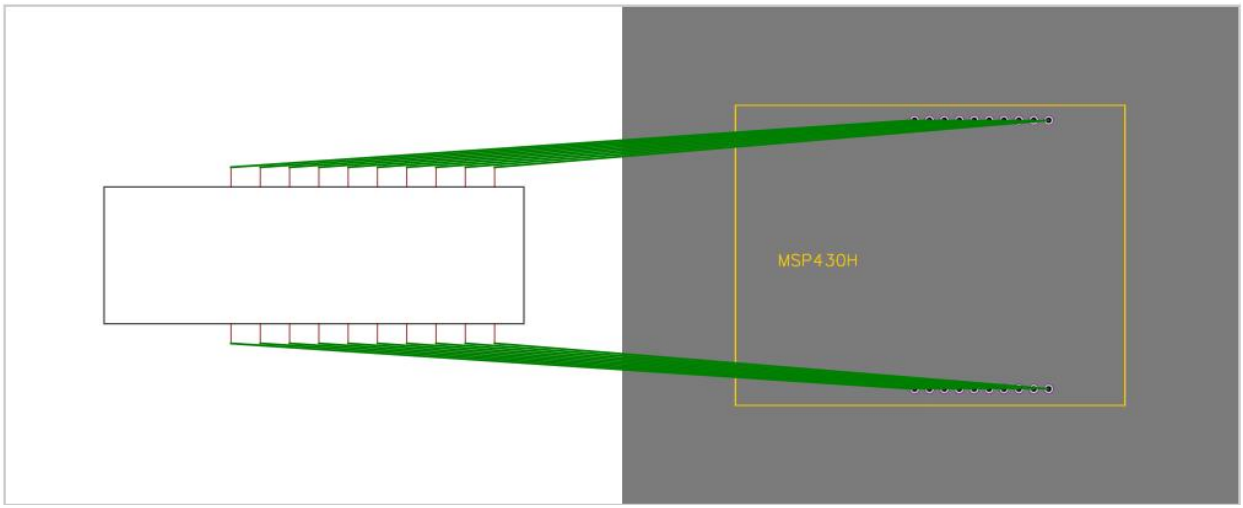

1
2
3
1
2
3

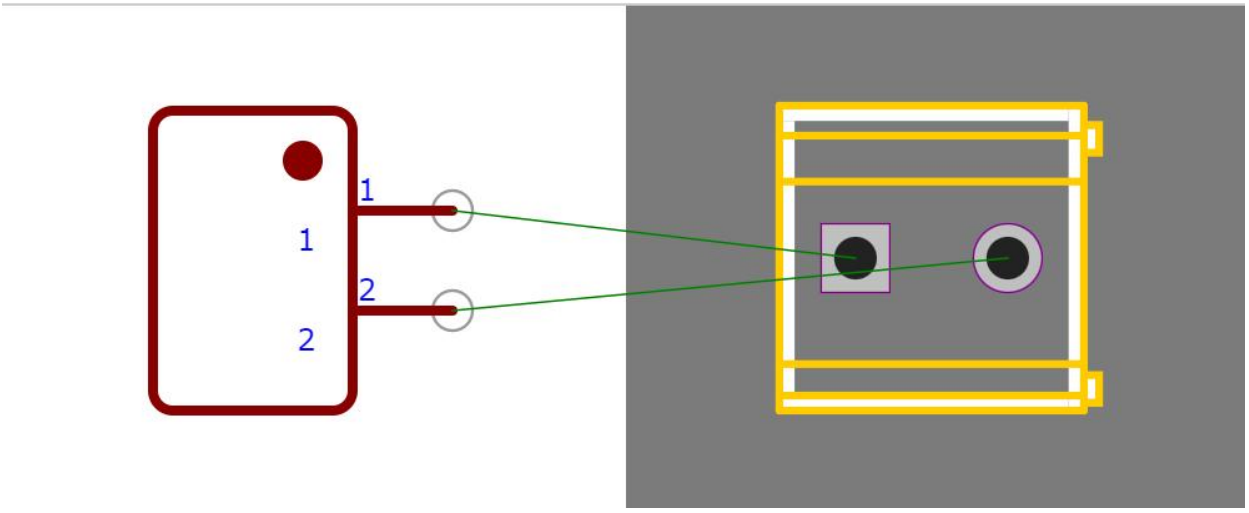

1
2
3
E
B
C

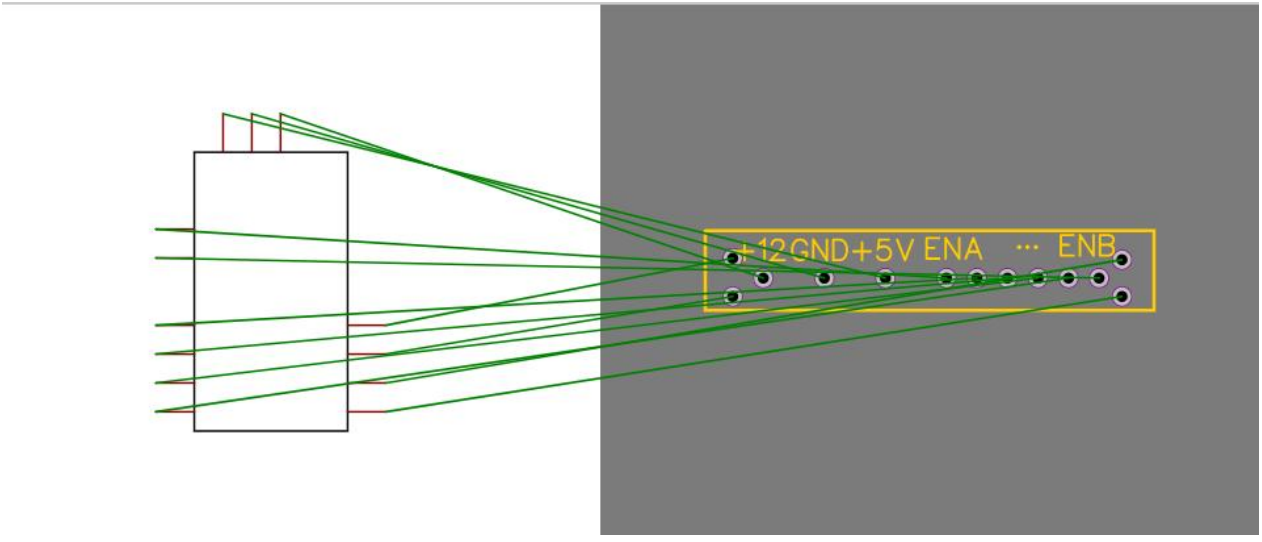

1
2
1
2
1
2


3
3
2
2
1
1

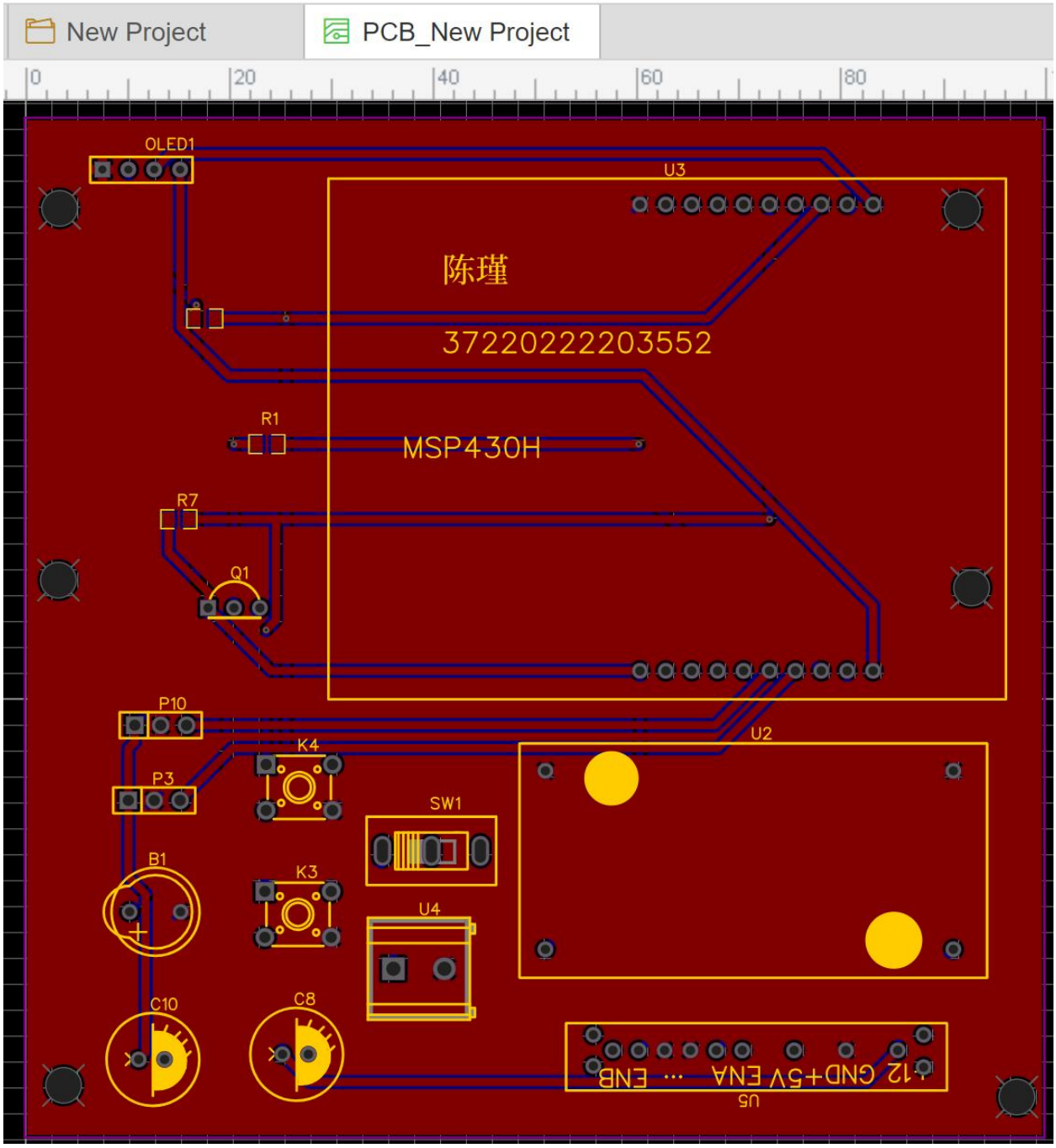

1
GND
2
VOUT
GND
3
VIN
4







丝印：



2. 简述 PCB 布局布线要注意哪些问题(以本次课程的案例为例分析)?

布局注意事项：

考虑元件的装配顺序和方向，以便于焊接和维修。

布线注意事项：

尽量保持走线在同一平面内，减少层间穿越。

根据电流大小选择合适的线宽，确保线间距满足制造要求和防止短路。

过孔应尽可能小，但也要考虑制造工艺的限制。

焊盘大小应适合元件引脚，避免焊接不良。

避免走线问题，如避免走线过长、过窄或急剧转弯，避免走线直接相邻或交叉

3．简述 PCB 设计的主要挑战？

初次学习并实操，操作不熟练，容错率低，且需要在有限的时间和空间内合理的布局布线。

4. PCB 经验总结与个人感受。

通过此次课程学习，我初次接触 PCB 设计并完成项目。我明白在课程中学到的理论知识是 PCB 设计的基础，只有通过实际操作，才能真正理解和掌握这些知识。学习使用 PCB 设计软件是非常重要的，熟悉软件的各种功能和工具对于提高设计效率至关重要。在设计过程中，选择合适的元件和对应的封装对于 PCB 设计的成功至关重要。此外，布局时要注意元件的合理分区和摆放，布线时要尽量减少走线长度和交叉，这些都是提高 PCB 性能的关键。在设计完成后，进行审查可以发现许多初步设计时忽略的问题，这有助于提高设计的质量和可靠性。

第一次完成 PCB 设计，感觉非常新奇和兴奋。看到自己设计的电路板变成实物，是一种独特的体验。在设计过程中难免遇到各种问题，但是通过寻求老师同学帮助、自己努力查阅资料解决这些问题，成就感还是很强烈的。实际操作中发现，理论知识和实际设计之间存在着一定的差距，这让我更加明白了实践的重要性。此外，和同学们一起讨论设计、解决问题，也让我感受到了团队合作的乐趣。

通过这次 PCB 设计的经历，我不仅学到了专业知识，还锻炼了动手能力和解决问题的能力，这对我未来的学习和职业发展都是宝贵的财富。

二、MSP430 单片机部分

作业 1、读取 MSP430G2553 LaunchPad 上 S2 的按键状态，并用该按键控制 LED1。按键按下时让 LED 1 亮起，按键松开时让 LED 1 熄灭。

（1）代码

```
#include <msp430.h>

/**
 * main.c
 */
int main(void)
{
    WDTCTL = WDTPW + WDTHOLD;   // stop watchdog timer
    DCOCTL=0;
    P1DIR |=BIT0;
    P1DIR &=~BIT3;
    P1REN|=BIT3;
    P1OUT|=BIT3;
    while(1)
    {
        if(!(BIT3&P1IN)){
            P1OUT|=BIT0;
        }
        else{
            P1OUT&=~BIT0;
        }
    }
}
```

（2）现象及讨论

按键按下时让 LED 1 亮起，按键松开时让 LED 1 熄灭（视频转动图后无法正常显示，所以现象视频另外上传，不在报告中体现）。

作业 2、在上一节 Blink 程序的基础上，将 MCLK 分别设置为 1MHz 和 8MHz，观察 LED1 闪烁的频率有何变化。

（1）代码

```
#include <msp430.h>


/**
 * main.c
```

```
 */
int main(void)
{
    WDTCTL = WDTPW + WDTHOLD;
    P1DIR |=0x01;
    //BCSCTL1=CALBC1_1MHZ;
    //DCOCTL=CALDCO_1MHZ;
    BCSCTL1=CALBC1_8MHZ;
    DCOCTL=CALDCO_8MHZ;

    for(;;)
    {
        volatile unsigned int i;
        P1OUT ^=0x01;
        i=50000;
        do(i--);
        while(i!=0);
    }
}
```

（3）现象及讨论

MCLK 设置为 8MHz 时，LED1 闪烁的频率明显比 MCLK 设置为 1MHz 快。

作业 3、通过外接矩阵键盘的方式控制 LED1 和 LED2 的亮灭。在矩阵键盘上任选两个按键 A 和 B 通过中断的方式分别控制 LED1 和 LED2，每按 A 或者 B 一次，对应的 LED 亮灭状态翻转一次。

（1）代码

```
#include <msp430.h>
#include <msp430g2553.h>
#include "stdint.h"

int main()
{
    WDTCTL = WDTPW | WDTHOLD;

    P1DIR |= BIT0;
    P1DIR |= BIT6;

    P1REN |= (BIT1 | BIT2);
    P1DIR &= ~(BIT1 | BIT2);
```

```
    P1IE = (BIT1 | BIT2);
    P1IES = (BIT1 | BIT2);
    P1IFG &= ~(BIT1 | BIT2);

    __bis_SR_register(GIE);

    while (1)
    {
        ;
    }
}
```

（2）现象及讨论

按下矩阵键盘上特定的两个按钮，对应的 LED 亮灭状态翻转一次。

作业 4（选做）在实验 3 中，大家有没有发现当按键按下或者抬起时，LED 会出现短暂的快速亮灭现象。这个现象是由于按键的抖动引起的，大家可以思考下如何消除抖动呢？提示：按键抖动的持续时间一般在 10ms 之内

（1）代码

（2）现象及讨论

作业 5、编程实现：利用定时器编写呼吸灯。所谓呼吸灯是指 LED 在一个周期内先逐渐变亮，再逐渐变暗。

（1）代码

```
#include <msp430.h>

volatile unsigned int brightness = 0;  // 亮度值
volatile int direction = 1;           // 亮度变化方向：1 表示变亮，-1 表示变暗

void main(void)
{
    WDTCTL = WDTPW | WDTHOLD;   // 停止看门狗定时器

    // 配置 LED 引脚
    P1DIR |= BIT6;              // 将 P1.6 设为输出
    P1SEL |= BIT6;              // 将 P1.6 设为定时器输出

    // 配置定时器 A0 用于 PWM
```

```
    TA0CCR0 = 1000-1;           // PWM 周期
    TA0CCTL1 = OUTMOD_7;        // TA0CCR1 重置/设置模式
    TA0CCR1 = brightness;       // 初始亮度
    TA0CTL = TASSEL_2 + MC_1;   // SMCLK, up mode

    // 配置定时器 A1 用于中断
    TA1CCTL0 = CCIE;            // 允许 CCR0 中断
    TA1CCR0 = 7000;             // 中断周期
    TA1CTL = TASSEL_2 + MC_1;   // SMCLK, up mode

    __bis_SR_register(GIE);     // 开启全局中断

    while (1)
    {
        // 主循环什么也不做，所有工作在中断中完成
    }
}

// 定时器 A1 CCR0 中断服务程序
#pragma vector = TIMER1_A0_VECTOR
__interrupt void Timer1_A0(void)
{
    // 更新亮度值，步长为 5
    brightness += direction * 5;

    if (brightness >= 1000)
    {
        direction = -1;  // 变暗
        brightness = 1000; // 防止亮度超出范围
    }
    else if (brightness <= 0)
    {
        direction = 1;   // 变亮
        brightness = 0; // 防止亮度低于 0
    }

    TA0CCR1 = brightness; // 更新 PWM 占空比
}
```

（2）现象及讨论

灯光亮度会逐渐增加然后逐渐减少，形成一个周期性的变化，类似于呼吸的动作。

作业 6、编写接收程序和发送程序，当开发板串口接收到 PC 机发来的字符“1”时，点亮 LED1，并向 PC 机发送“LED_ON”；当当开发板串口接收到 PC 机发来的字符“0”时，熄灭 LED1，并向 PC 机发送“LED_OFF”；当收到其它字符时，翻转 LED1 状态，并向 PC 机发送“LED_ON”或者“LED_OFF”，表明 LED1 当前的状态。

（1）代码

```
#include <msp430.h>
#include <msp430g2553.h>
#include "stdint.h"

/**
 * main.c
 */

void send_string(char str[])
{
    unsigned int i = 0;
    while (str[i] != '\0')
    {
        while (!(IFG2 & UCA0TXIFG))
            __no_operation();
        UCA0TXBUF = str[i];
        i++;
    }
}

int main(void)
{
    WDTCTL = WDTPW | WDTHOLD; // 停止看门狗定时器

    P1DIR |= BIT0;
    P1OUT &= ~BIT0;

    P1SEL = BIT1 + BIT2;  // P1.1 = RXD, P1.2 = TXD
    P1SEL2 = BIT1 + BIT2; // P1.1 = RXD, P1.2 = TXD

    UCA0CTL1 |= UCSSEL_2; // 使用 SMCLK
    UCA0BR0 = 104;
    UCA0BR1 = 0;
    UCA0MCTL = UCBRS0;
    UCA0CTL1 &= ~UCSWRST;
    IE2 |= UCA0RXIE;
```

```
    __bis_SR_register(GIE);

    send_string("Hello World!\n");

    while (1)
    {
        __no_operation();
    }
}

#pragma vector = USCIAB0RX_VECTOR
__interrupt void USCI0RX_ISR(void)
{
    volatile unsigned char ch;
    ch = UCA0RXBUF;
    switch (ch)
    {
    case '1':
        P1OUT |= BIT0;
        send_string("LED_ON\n");
        break;
    case '0':
        P1OUT &= ~BIT0;
        send_string("LED_OFF\n");
        break;
    default:
        P1OUT ^= BIT0;
        if (P1OUT & BIT0)
            send_string("LED_ON\n");
        else
            send_string("LED_OFF\n");
        break;
    }
}
```

（2）现象及讨论

Led 状态随输入发送变化，视频中是开启后关闭。

作业 7、按如下格式在 OLED 屏上显示自己的姓名、学号和系别信息。

“姓名：***”

“学号：****”

“系别：****”

（1）代码

```
#include <msp430.h>
#include "oledfont.h"
#include "oled.h"       //里面有 SDA 和 SCL 引脚定义，移植时需注意修改

//延时函数，IAR 自带，经常使用到
#define CPU_F ((double)8000000)   //外部高频晶振 8MHZ
//#define CPU_F ((double)32768)   //外部低频晶振 32.768KHZ
#define delay_us(x) __delay_cycles((long)(CPU_F*(double)x/1000000.0))
#define delay_ms(x) __delay_cycles((long)(CPU_F*(double)x/1000.0))

void main(void)//主函数
{
    WDTCTL = WDTPW + WDTHOLD;       //关闭看门狗

    BCSCTL1 = CALBC1_8MHZ;              // Set range
    DCOCTL = CALDCO_8MHZ;

    OLED_Init(); //OLED 初始化
    delay_ms(200);

     OLED_ShowCHinese(0, 0, 0);
     OLED_ShowCHinese(16,0,1);
     OLED_ShowChar(32,0,':');
     OLED_ShowCHinese(48,0,2);
     OLED_ShowCHinese(64,0,3);
     OLED_ShowCHinese(0,2,4);
     OLED_ShowCHinese(16,2,5);
     OLED_ShowChar(32,2,':');
     OLED_ShowChar(8, 4, '3');
     OLED_ShowChar(16, 4, '7');
     OLED_ShowChar(24, 4, '2');
     OLED_ShowChar(32, 4, '2');
     OLED_ShowChar(40, 4, '0');
     OLED_ShowChar(48, 4, '2');
     OLED_ShowChar(56, 4, '2');
     OLED_ShowChar(64, 4, '2');
     OLED_ShowChar(72, 4, '2');
     OLED_ShowChar(80, 4, '0');
     OLED_ShowChar(88, 4, '3');
     OLED_ShowChar(96, 4, '5');
     OLED_ShowChar(104, 4, '5');
     OLED_ShowChar(112, 4, '2');
```

```
    OLED_ShowCHinese(0,6,6);
    OLED_ShowCHinese(16,6, 7);
    OLED_ShowChar(32,6,':');
    OLED_ShowCHinese(48,6, 8);
    OLED_ShowCHinese(64,6, 9);

}
```

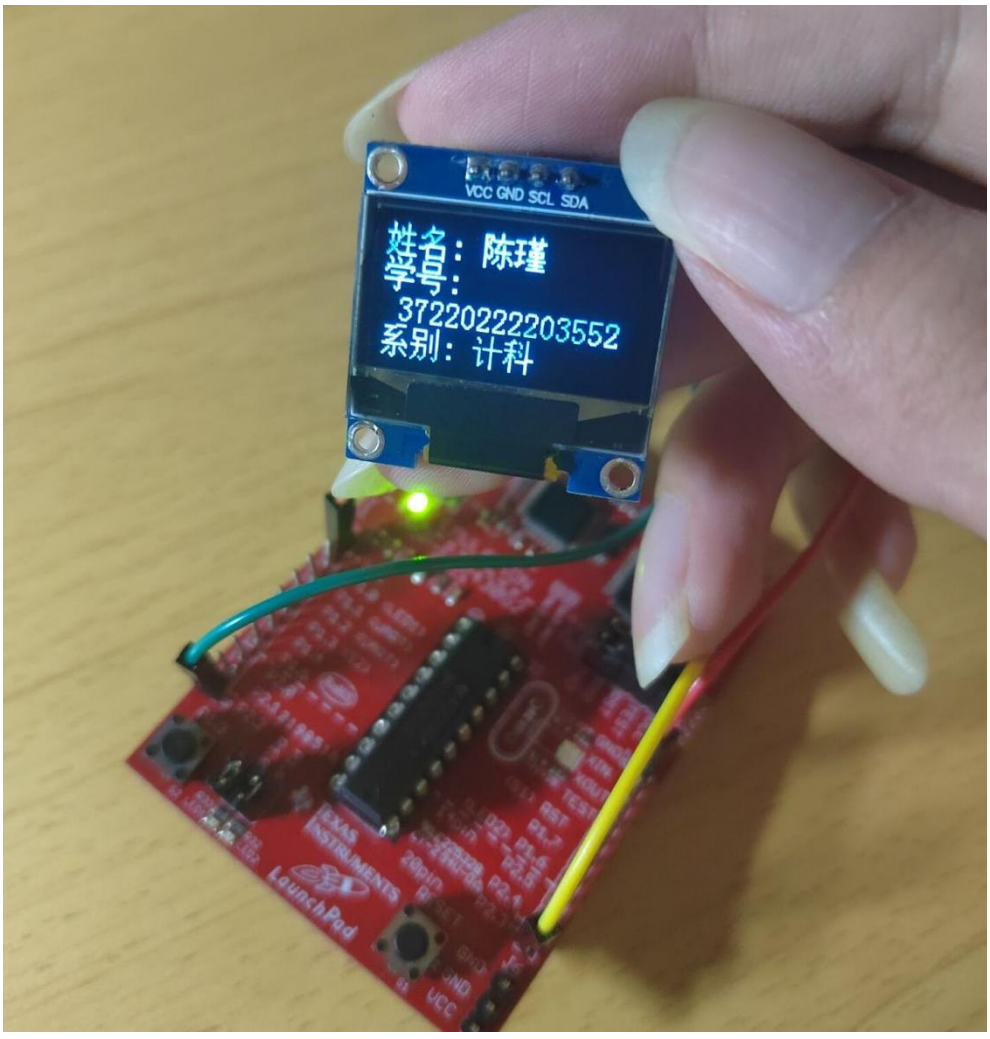
字模文件：

```
const char Hzk[][32]=
{
 {0x10,0x10,0xF0,0x1F,0x10,0xF0,0x40,0x3C,0x10,0x10,0xFF,0x10,0x10,0x10,0x00,0x00},

{0x40,0x22,0x15,0x08,0x16,0x21,0x40,0x42,0x42,0x42,0x7F,0x42,0x42,0x42,0x40,0x00},/*"
姓",0*/

 {0x00,0x20,0x20,0x10,0x08,0x14,0x67,0x84,0x44,0x24,0x14,0x0C,0x00,0x00,0x00,0x00},

{0x04,0x04,0x04,0x02,0xFE,0x43,0x43,0x42,0x42,0x42,0x42,0x42,0xFE,0x00,0x00,0x00},/*"
名",1*/

 {0x00,0xFE,0x22,0x5A,0x86,0x08,0x88,0x68,0x18,0x0F,0xE8,0x08,0x08,0x08,0x08,0x00},

{0x00,0xFF,0x04,0x08,0x07,0x20,0x11,0x0D,0x41,0x81,0x7F,0x01,0x05,0x09,0x30,0x00},/*"
陈",2*/

 {0x84,0x84,0xFC,0x84,0x84,0x00,0x04,0xC4,0x5F,0x54,0xF4,0x54,0x5F,0xC4,0x04,0x00},

{0x10,0x30,0x1F,0x08,0x08,0x00,0x44,0x55,0x55,0x55,0x7F,0x55,0x55,0x55,0x44,0x00},/*"
瑾",3*/

 {0x40,0x30,0x11,0x96,0x90,0x90,0x91,0x96,0x90,0x90,0x98,0x14,0x13,0x50,0x30,0x00},

{0x04,0x04,0x04,0x04,0x04,0x44,0x84,0x7E,0x06,0x05,0x04,0x04,0x04,0x04,0x04,0x00},/*"
学",4*/

 {0x80,0x80,0x80,0xBE,0xA2,0xA2,0xA2,0xA2,0xA2,0xA2,0xBE,0x80,0x80,0x80,0x00},

{0x00,0x00,0x00,0x06,0x05,0x04,0x04,0x04,0x44,0x84,0x44,0x3C,0x00,0x00,0x00,0x00},/*"
号",5*/

 {0x00,0x00,0x22,0x32,0x2A,0xA6,0xA2,0x62,0x21,0x11,0x09,0x81,0x01,0x00,0x00,0x00},

{0x00,0x42,0x22,0x13,0x0B,0x42,0x82,0x7E,0x02,0x02,0x0A,0x12,0x23,0x46,0x00,0x00},/*"
系",6*/
```

```
 {0x00,0x3E,0x22,0xE2,0x22,0x22,0x22,0x3E,0x00,0x00,0xF8,0x00,0x00,0xFF,0x00,0x00},

{0x81,0x41,0x31,0x0F,0x41,0x81,0x41,0x3F,0x00,0x00,0x0F,0x40,0x80,0x7F,0x00,0x00},/*"别",7*/

 {0x40,0x40,0x42,0xCC,0x00,0x40,0x40,0x40,0x40,0xFF,0x40,0x40,0x40,0x40,0x40,0x00},

{0x00,0x00,0x00,0x7F,0x20,0x10,0x00,0x00,0x00,0xFF,0x00,0x00,0x00,0x00,0x00,0x00},/*"计",8*/

 {0x24,0x24,0xA4,0xFE,0xA3,0x22,0x00,0x22,0xCC,0x00,0x00,0xFF,0x00,0x00,0x00,0x00},

{0x08,0x06,0x01,0xFF,0x00,0x01,0x04,0x04,0x04,0x04,0x04,0xFF,0x02,0x02,0x02,0x00},/*"科",9*/
};
#endif
```

（2）现象及讨论

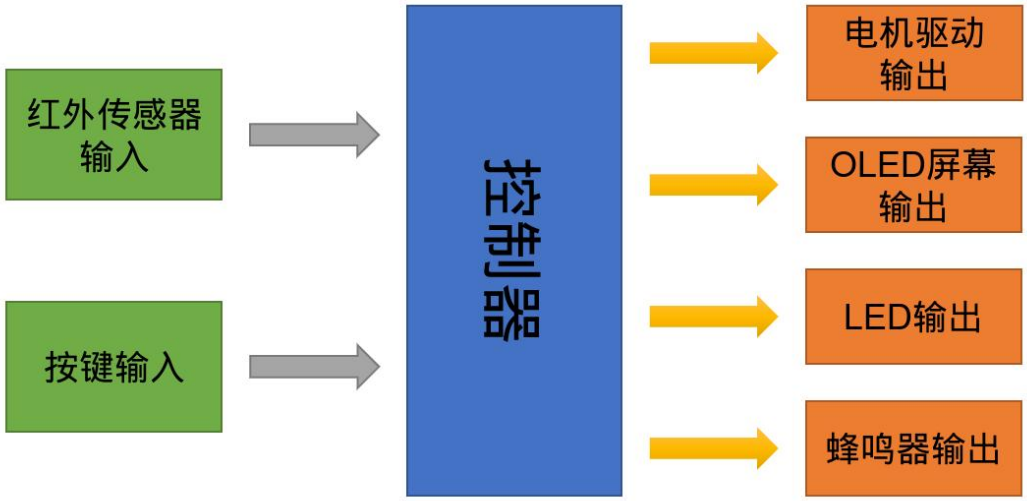
Oled 上显示了姓名，学号，系别信息。



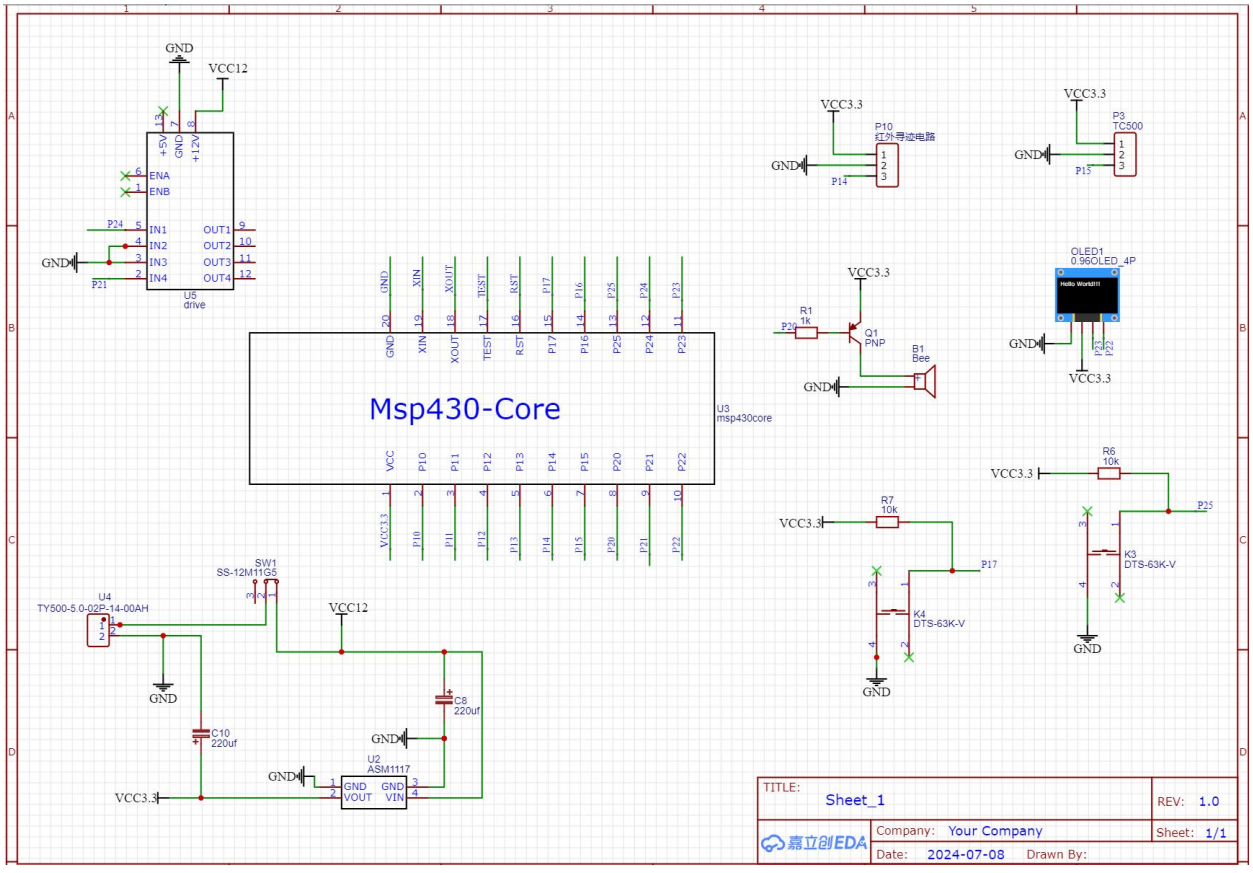
三、基于 MSP430 单片机的智能小车行驶

1. 电路与程序设计：系统组成、原理框图与各部分的电路图，系统软件与流程图。

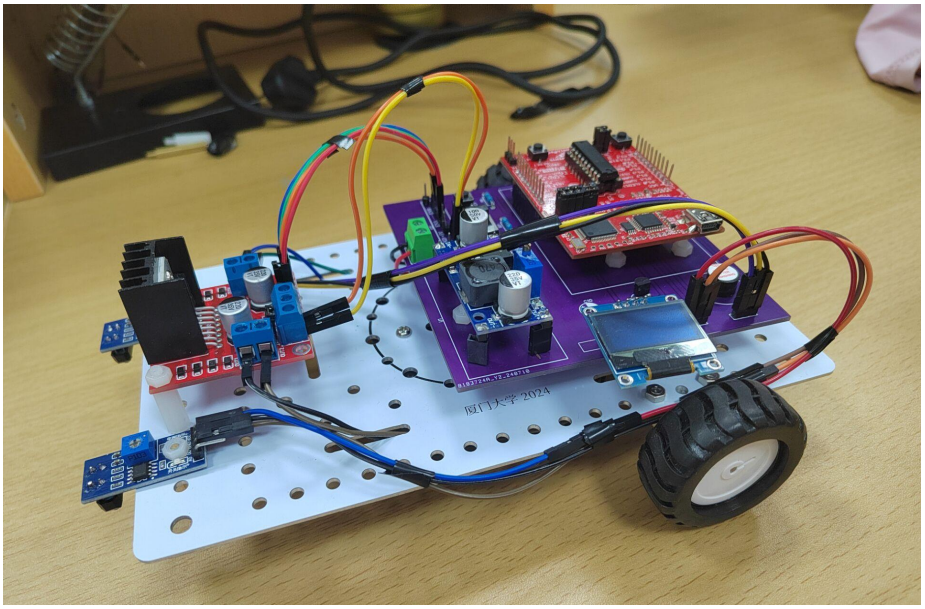
系统组成：

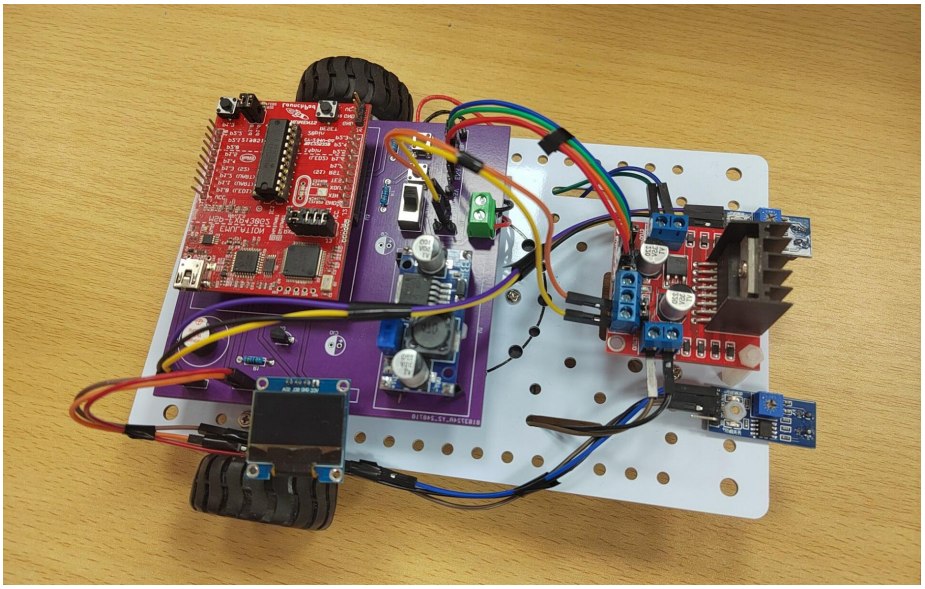


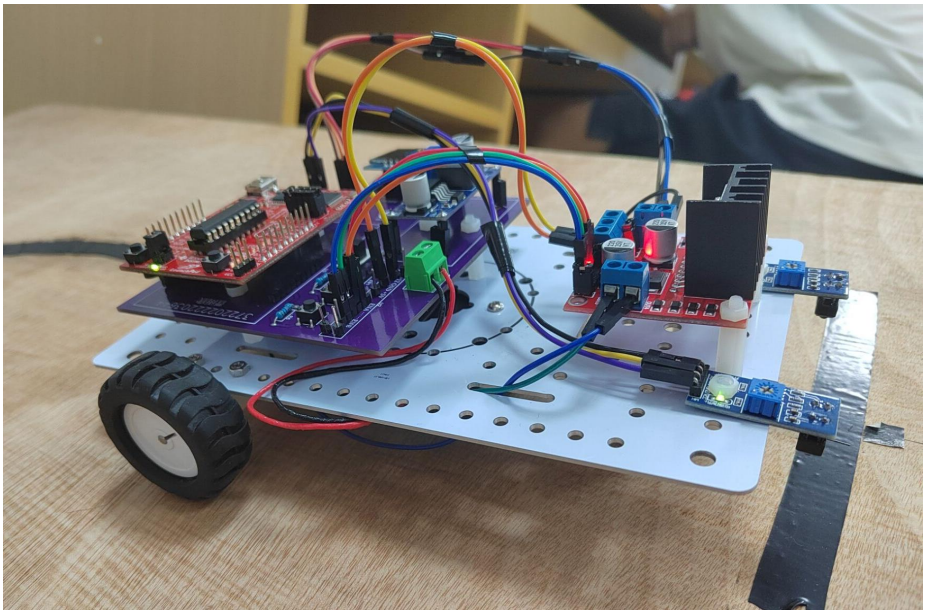
原理框图：



电路图：







系统软件：

Code Composer Studio 10.3.1 

核心代码：

```
#include <msp430.h>
#include "oledfont.h"
#include "oled.h"

#define CPU_F ((double)8000000)
#define delay_us(x) __delay_cycles((long)(CPU_F*(double)x/1000000.0))
#define delay_ms(x) __delay_cycles((long)(CPU_F*(double)x/1000.0))

#define hy1 (P1IN&BIT4)
#define hy2 (P1IN&BIT5)

#define key3 (P1IN&BIT3)
#define key2 (P1IN&BIT7)
#define key1 (P2IN&BIT5)

#define FM_L P2OUT &= ~BIT0
#define FM_H P2OUT |= BIT0

long Left = 6800;
long Right = 6800;
long Turn_speed = 6800;
int Rev_speed = 0;
const int Turn_Time = 5000;
int stop_time = 10;
const int Delay_Time = 40;
const int Change_turn_speed = 400;
const int Change_up_speed = 400;
const int Change_down_speed = 400;
int change_num = 0;
const int xiaodou = 100;

int t_miao = 0, count = 0;
volatile int start_time = 0, stop_time1 = 0;
volatile int timing_started = 0;

void TIMERA_Init(void) {
    TA0CCTL0 = CCIE;
    TA0CTL |= (TASSEL_2 + ID_3 + MC_2);
    TA0CCR0 = 50000;
}
```

```
#pragma vector=TIMER0_A0_VECTOR
__interrupt void Timer_A(void) {
    TA0CCR0 += 50000;
    count++;

    if (count >= 20) {
        count = 0;
        t_miao++;
        P1OUT ^= BIT6;
    }

    if (timing_started) {
        stop_time++;
    }
}

void oled_show_run() {
    OLED_ShowCHinese(32, 3, 10);
    OLED_ShowCHinese(48, 3, 11);
    OLED_ShowCHinese(64, 3, 14);
}

void oled_show_turnleft() {
    OLED_ShowCHinese(32, 3, 15);
    OLED_ShowCHinese(48, 3, 17);
    OLED_ShowCHinese(64, 3, 14);
}

void oled_show_turnright() {
    OLED_ShowCHinese(32, 3, 16);
    OLED_ShowCHinese(48, 3, 17);
    OLED_ShowCHinese(64, 3, 14);
}

void oled_show_stop() {
    OLED_ShowCHinese(32, 3, 12);
    OLED_ShowCHinese(48, 3, 13);
    OLED_ShowCHinese(64, 3, 14);
}

void Go() {
    TA1CTL = (TASSEL_2 + ID_0 + MC_1);
    unsigned int i;

    start_time = t_miao; // Record start time
```

```
timing_started = 1;

while (1) {
    if ((P1IN & BIT5) && (P1IN & BIT4)) {
        TA1CCR2 = 0;
        TA1CCR1 = 0;
        timing_started = 0; // Record stop time
        stop_time1 = t_miao;
        return;
    } else if (P1IN & BIT5) {
        TA1CCR2 = Turn_speed;
        TA1CCR1 = Rev_speed;

        oled_show_turnright();
        P1OUT |= BIT6;

        for (i = 0; i < Turn_Time; ++i) {
            if ((P1IN & BIT5) && (P1IN & BIT4)) {
                TA1CCR2 = 0;
                TA1CCR1 = 0;
                timing_started = 0;
                stop_time1 = t_miao;
                return;
            } else if (!(P1IN & BIT5)) {
                for (i = 0; i < Delay_Time; ++i) {
                    if ((P1IN & BIT5) && (P1IN & BIT4)) {
                        TA1CCR2 = 0;
                        TA1CCR1 = 0;
                        timing_started = 0;
                        stop_time1 = t_miao;
                        return;
                    }
                    delay_ms(1);
                }
                break;
            }
        }
    } else if (P1IN & BIT4) {
        TA1CCR2 = Rev_speed;
        TA1CCR1 = Turn_speed;

        oled_show_turnleft();
        P1OUT |= BIT0;

        for (i = 0; i < Turn_Time; ++i) {
            if ((P1IN & BIT5) && (P1IN & BIT4)) {
```

```
                    TA1CCR2 = 0;
                    TA1CCR1 = 0;
                    timing_started = 0;
                    stop_time1 = t_miao;
                    return;
                } else if (!(P1IN & BIT4)) {
                    for (i = 0; i < Delay_Time; ++i) {
                        if ((P1IN & BIT5) && (P1IN & BIT4)) {
                            TA1CCR2 = 0;
                            TA1CCR1 = 0;
                            timing_started = 0;
                            stop_time1 = t_miao;
                            return;
                        }
                        delay_ms(1);
                    }
                    break;
                }
            }
        } else {
            oled_show_run();
            P1OUT &= ~BIT0;
            P1OUT &= ~BIT6;

            TA1CCR2 = Left;
            TA1CCR1 = Right;
        }
        int current_time = t_miao - start_time;
                OLED_ShowChar(0,0,current_time/100 + '0');
                OLED_ShowChar(8,0,(current_time/10) % 10 + '0');
                OLED_ShowChar(16,0,current_time % 10 + '0');
    }

}

void Switch() {
    while (1) {
        if (change_num >= 0) {
            OLED_ShowCHinese(0, 6, 23);
            OLED_ShowCHinese(16, 6, 25);
            OLED_ShowChar(32, 6, change_num / 10 + '0');
            OLED_ShowChar(40, 6, change_num % 10 + '0');
            OLED_ShowCHinese(48, 6, 26);
        } else {
            OLED_ShowCHinese(0, 6, 24);
            OLED_ShowCHinese(16, 6, 25);
```

```
            OLED_ShowChar(32, 6, (-change_num) / 10 + '0');
            OLED_ShowChar(40, 6, (-change_num) % 10 + '0');
            OLED_ShowCHinese(48, 6, 26);
        }

        if (!(P1IN & BIT3)) {
            delay_ms(10);
            if (!(P1IN & BIT3)) {
                break;
            }
        } else {
            if (!(P1IN & BIT7)) {
                delay_ms(xiaodou);
                if (!(P1IN & BIT7)) {
                    Right -= Change_down_speed;
                    Left = Right;
                    change_num--;
                }
            } else if (!(P2IN & BIT5)) {
                delay_ms(xiaodou);
                if (!(P2IN & BIT5)) {
                    Right += Change_up_speed;
                    Left = Right;
                    change_num++;
                }
            }
        }
    }
}

void buzzle() {
    P2OUT &= ~BIT0;
    delay_ms(500);
    P2OUT |= BIT0;
}

void main(void) {
    WDTCTL = WDTPW + WDTHOLD;
    BCSCTL1 = CALBC1_8MHZ;
    DCOCTL = CALDCO_8MHZ;

    TIMERA_Init();
    _EINT();

    OLED_Init();
    delay_ms(200);
```

```
P1DIR |= BIT0;
P1OUT &= ~BIT0;
P1DIR |= BIT6;
P1OUT &= ~BIT6;

P1DIR &= ~BIT3;
P1REN |= BIT3;
P1OUT |= BIT3;

P1DIR &= ~BIT4;
P1REN |= BIT4;
P1OUT |= BIT4;

P1DIR &= ~BIT5;
P1REN |= BIT5;
P1OUT |= BIT5;

P1DIR &= ~BIT7;
P1REN |= BIT7;
P1OUT |= BIT7;

P2DIR |= BIT0;
P2OUT |= BIT0;

P2DIR |= (BIT1 + BIT4);
P2SEL |= (BIT1 + BIT4);
P2OUT &= ~BIT1;
P2OUT &= ~BIT4;

P2DIR &= ~BIT5;
P2REN |= BIT5;
P2OUT |= BIT5;

TA1CCR0 = 20000;
TA1CCTL1 = OUTMOD_7;
TA1CCR1 = 0;
TA1CCTL2 = OUTMOD_7;
TA1CCR2 = 0;

Switch();
Go();
oled_show_stop();
buzzle();

    int current_time = t_miao - start_time;
```

```
        OLED_ShowChar(0,0,current_time/100 + '0');
        OLED_ShowChar(8,0,(current_time/10) % 10 + '0');
        OLED_ShowChar(16,0,current_time % 10 + '0');

}
```

流程图：

电路设计→元器件焊接→小车组装→小车调试

2. 测试方案与测试结果：测试过程记录，测试结果分析。

经过各种调试，解决存在的问题后，正式测试并记录结果：

平地

加速次数	时间	
0	22	
10	17	-4
20	14	-3
30	11	-3

爬坡

加速次数	时间	
0	26	
10	20	-6
20	16	-4
30	12	-4

可以看出，前十次加速效果最明显，到后面变得稳定，基本同次数加速效果相差不大，且相同情况下，爬坡耗时多于平地循迹。

3. 本人在小车项目中所做的工作。

辅助元器件焊接、电线连接与小车组装，编写代码，调试小车。

4. 小车项目经验总结与个人感受。

本次小车项目在实施过程中，我们团队遇到了诸多挑战，调试过程可谓是一波三折。然而，在团队成员的共同努力下，我们逐一克服了这些困难，共同成长。项目中让我印象最为深刻的两个问题是电机损坏与红外灵敏度不合适。由于我们拿到的电机有问题，导致小车两侧轮子驱动力不同，无法正常行驶，在调试过程中也没有察觉此问题，请助教帮助 debug 也没能找出问题（大概是坏了但没完全坏，所以一侧供电不足，但由于也有电流通过，所以不易发现问题），最后还是排除了其他可能出现的问题后，换了电机，解决了该问题；此外，红外灵敏度问题也给我们的项目带来了不小的困扰，在组装完成后，我们没有及时调整红外传感器的灵敏度，导致在测试过程中出现了一系列问题，幸运的是，我们团队迅速反应，及时调整了红外灵敏度，使得问题得到了妥善解决，没有对项目进度造成太大影响。

通过这次项目，我深刻体会到了团队协作的重要性。面对困难，我们相互支持、共同探讨，最终找到了解决问题的方法。同时，我也认识到了细节决定成败的道理，今后的工作中，我将更加注重细节，力求做到精益求精。这次项目经历对我来说是一次宝贵的成长机会，我将珍惜这段时光，继续努力，为团队贡献自己的力量。

四、实验改进建议

强烈建议：把坏了的元器件及时处理掉，不要再和好的元器件放在一起，像我们这次实验，拿到了一个坏的电机，调试过程多废了不少心思，严重影响了实验进度。

请把坏了的元器件及时处理掉！！！

请把坏了的元器件及时处理掉！！！

请把坏了的元器件及时处理掉！！！