**华中科技大学**

**《信息技术导论》实验报告**

|  |  |
| --- | --- |
| 实验名称： | 智能小车循迹 |
| 院（系）： | 电子信息与通信工程学院 |
| 专业班级： | 提高2101班 |
| 姓名： | 杨筠松、王泳丹、王润鹏 |
| 学号： | U202115980、U202114206、U202116237 |
| 时间： | 2021.12.11 |
| 地点： | 实验室204 |
| 实验成绩： |  |
| 指导教师： | 胡蓉 |

**一、实验目的**

通过学习信息技术导论相关知识，掌握基础有关理论，通过编写C语言程序控制小车，沿着5cm的黑线进行循迹，到达指定的终点；意在培养学生的系统观和软硬协同观念，最终实现对本专业的专业研究方向认知。

**二、实验内容**

自行组装小车，通过已经学习的C语言知识，自行编写运行逻辑完成小车循迹任务。

**三、主要设备仪器**

CCS软件、TI-RSLK机器人系统教学套件

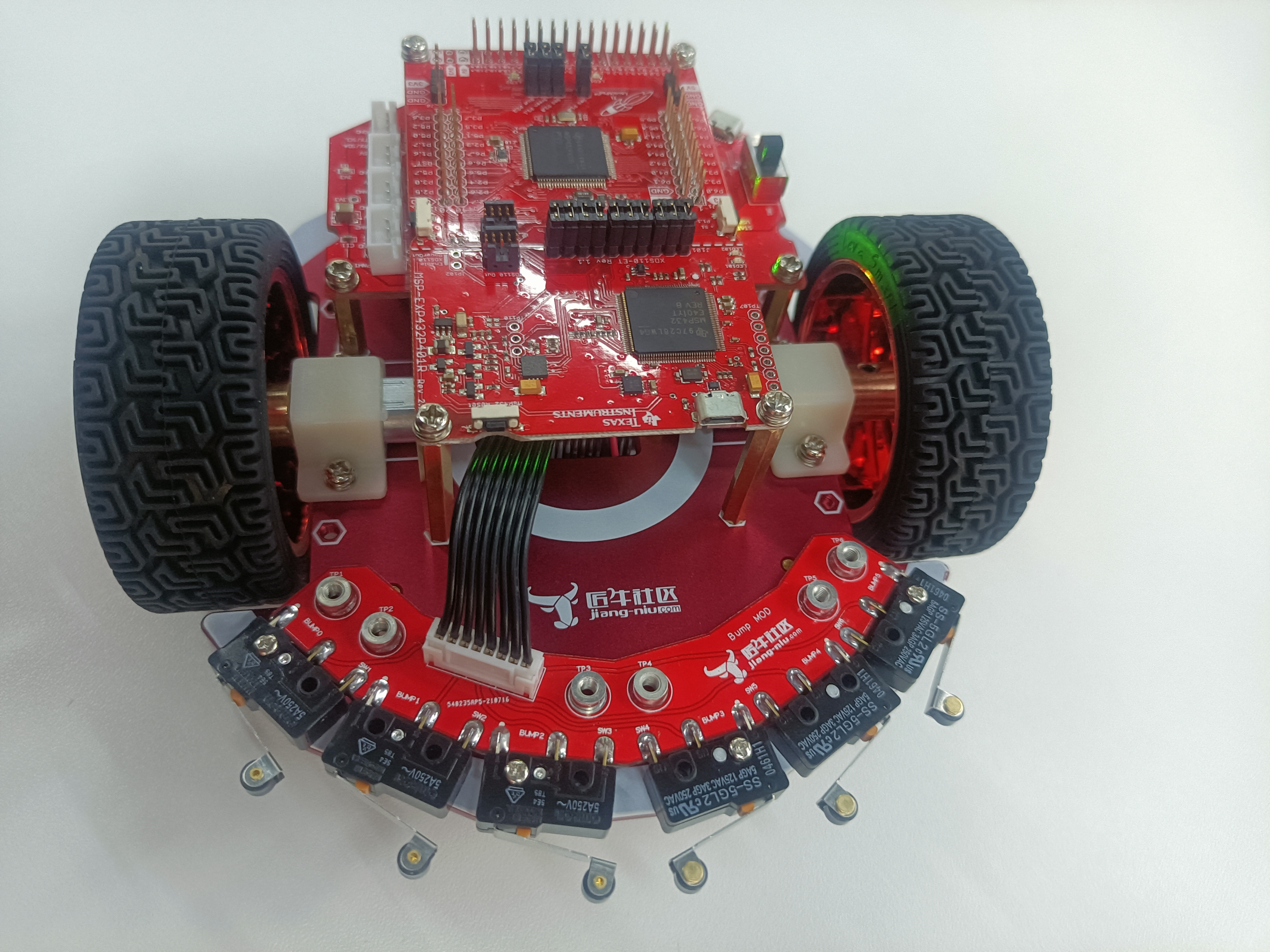
**四、操作方法和实验步骤**

任务一：确定试验场地的情况，并构思基本循迹逻辑

由于上课用的是3cm的胶带，而实际场地中使用的却是5cm，对于传感器的影响比较大。同时考虑到小车在运行中实现方向校正的情景，难以完整判断所有情况，于是决定只是对于最高位最低位置进行判断来确定小车运行方向以及前进。逻辑简单，可实施性强，同时情况相对较少，对于后期前进参数的调整大有裨益。

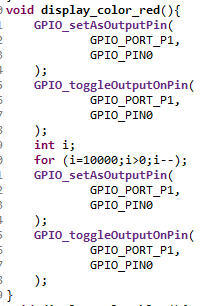
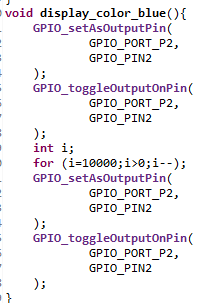
任务二：组装好系统教学套件，并实现基本的直行

按照PPT中一步一步进行安装和调整最后实现了小车的成功组装。

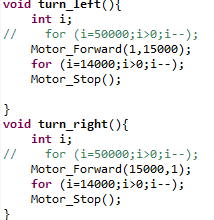


任务三：编写左右转弯的程序，并且编写不同场景下灯光的变化程序

由于场地灯光情况复杂，考虑对于不同场景下的数据进行判断，并且以灯光变化的方式呈现。

通过小车上的灯光便可以判断，是场地光线原因还是程序逻辑原因，有利于对左转右转的情况进行调整。下面是左转右转代码的展示：



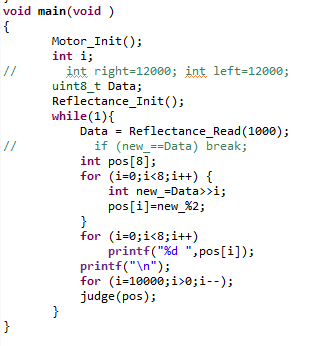
以及判断场景情况并且做出对应反应的代码：



任务四：完成最终场地的布置，对于运行参数调整，并拍摄最终视频

鉴于不同情况下对于场地要求不同，需要对场地进行遮光，以及在不同情况下的转速进行调整，并且最终实现了小车的运行。

主调函数：main的代码展示



**五、实验结果和分析**

小车可以按照一定路线利用既定设计好的程序进行循迹行驶

**六、心得体会**

在本次实验中，充分感受到了软硬协同观念的重要性，不同情况下的实验对象的表现，要从软件逻辑，硬件获取是否成功两个方向进行考虑，不可偏颇。同时进一步提升成员的编写代码能力以及抗挫能力，在困难面前不气馁；小组成员一起合作精神同样彰显的淋漓尽致，对接下来专业方向的学习有了具体可感的感受。虽然没有完全弄懂硬件实现逻辑，但是对于今后学习方向有了指导作用。

**附录**

完整的代码如下所示：

#include <ti/devices/msp432p4xx/driverlib/driverlib.h>

#include<stdio.h>

void Motor\_Init(void)

{

P1->SEL0 &= ~0xC0;

P1->SEL1 &= ~0xC0; // configure as GPIO

P1->DIR |= 0xC0; // make P1.6 & P1.7 out

P1->OUT &= ~0xC0;

P3->SEL0 &= ~0xC0;

P3->SEL1 &= ~0xC0; // configure as GPIO

P3->DIR |= 0xC0; // make P3.6 & P3.7 out

P3->OUT &= ~0xC0; // low current sleep mode

}//对电机进行初始化

void PWM\_Init34(uint16\_t period, uint16\_t duty3, uint16\_t duty4)

{

if(duty3 >= period) return; // bad input

if(duty4 >= period) return; // bad input

P2->DIR |= 0xC0; // P2.6, P2.7 output

P2->SEL0 |= 0xC0; // P2.6, P2.7 Timer0A functions

P2->SEL1 &= ~0xC0;

TIMER\_A0->CCTL[0] = 0x0080; // CCI0 toggle

TIMER\_A0->CCR[0] = period; // Period

TIMER\_A0->EX0 = 0x0000; // divide by 1

TIMER\_A0->CCTL[3] = 0x0040; // CCR3 toggle/reset

TIMER\_A0->CCR[3] = duty3; // CCR3 duty cycle is duty3/period

TIMER\_A0->CCTL[4] = 0x0040; // CCR4 toggle/reset

TIMER\_A0->CCR[4] = duty4; // CCR4 duty cycle is duty4/period

TIMER\_A0->CTL = 0x02F0; // SMCLK=12MHz, divide by 8, up-down mode

}

void Motor\_Forward(uint16\_t leftDuty, uint16\_t rightDuty)

{

P1->OUT |= 0xC0; // set direction of motors

PWM\_Init34(15000, 15000 - rightDuty, 15000 - leftDuty);

P3->OUT |= 0xC0; // activate motors

}

//功能控制小车后退

void Motor\_Backward(uint16\_t leftDuty, uint16\_t rightDuty)

{

P1->OUT &= ~0xC0; // set direction of motors

PWM\_Init34(15000, rightDuty, leftDuty);

P3->OUT |= 0xC0; // activate motors

}

//功能控制小车停止

void Motor\_Stop(void)

{

P1->OUT &= ~0xC0;

P2->OUT &= ~0xC0; // off

P3->OUT &= ~0xC0; // low current sleep mode

}

void Reflectance\_Init(void){

P5->SEL0&= ~0xFF;

P5->SEL1&= ~0xFF;

P5->DIR|= 0x08;

P5->OUT&= ~0x08;

P7->SEL0&= ~0xFF;

P7->SEL1&= ~0xFF;

P7->DIR&= ~0xFF; // P7.0-7.7 input

}

uint8\_t Reflectance\_Read(uint32\_t time){

uint8\_t result;

int i;

P5->OUT|= 0x08; // Turn on IR light

P7->DIR|= 0xFF; // P7.0-7.7 output

P7->OUT|= 0xFF; // Set P7.0-7.7 high

for(i=100; i>0; i--); // loop delay

P7->DIR&= ~0xFF; // P7.0-7.7 input

P7->REN&= ~0xFF;

for(i=time; i>0; i--); // loop delay

result = P7->IN;

P5->OUT&= ~0x08; // Turn off IR light

return result;

}

void turn\_left(){

int i;

// for (i=50000;i>0;i--);

Motor\_Forward(1,15000);

for (i=14000;i>0;i--);

Motor\_Stop();

}

void turn\_right(){

int i;

// for (i=50000;i>0;i--);

Motor\_Forward(15000,1);

for (i=14000;i>0;i--);

Motor\_Stop();

}

void display\_color\_red(){

GPIO\_setAsOutputPin(

GPIO\_PORT\_P1,

GPIO\_PIN0

);

GPIO\_toggleOutputOnPin(

GPIO\_PORT\_P1,

GPIO\_PIN0

);

int i;

for (i=10000;i>0;i--);

GPIO\_setAsOutputPin(

GPIO\_PORT\_P1,

GPIO\_PIN0

);

GPIO\_toggleOutputOnPin(

GPIO\_PORT\_P1,

GPIO\_PIN0

);

}

void display\_color\_blue(){

GPIO\_setAsOutputPin(

GPIO\_PORT\_P2,

GPIO\_PIN2

);

GPIO\_toggleOutputOnPin(

GPIO\_PORT\_P2,

GPIO\_PIN2

);

int i;

for (i=10000;i>0;i--);

GPIO\_setAsOutputPin(

GPIO\_PORT\_P2,

GPIO\_PIN2

);

GPIO\_toggleOutputOnPin(

GPIO\_PORT\_P2,

GPIO\_PIN2

);

}

void judge(int s[]){

int i;

WDT\_A\_hold(WDT\_A\_BASE);

if ((s[7]==0&&s[0]==0)||(s[7]==1&&s[0]==1)) {

// display\_color\_red();

Motor\_Forward(8000,8000);

for (i=20000;i>0;i--);

Motor\_Stop();

}//如果当前数据获取是0XXXXXX0

if ((s[7]==1&&s[0]==0)) {

display\_color\_blue();

turn\_right();

//如果当前数据获取的是1XXXXXX0}进行右转

}

if (s[7]==0&&s[0]==1) {

display\_color\_red();//如果当前数据获取的是0XXXXXX1

turn\_left();

}

}

void main(void )

{

Motor\_Init();

int i;

// int right=12000; int left=12000;

uint8\_t Data;

Reflectance\_Init();

while(1){

Data = Reflectance\_Read(1000);

// if (new\_==Data) break;

int pos[8];

for (i=0;i<8;i++) {

int new\_=Data>>i;

pos[i]=new\_%2;

}

for (i=0;i<8;i++)

printf("%d ",pos[i]);

printf("\n");

for (i=10000;i>0;i--);

judge(pos);

}

}