# 实验3.4 矩阵键盘扫描实验

姓名：孟麟芝 学号：201800121050 实验时间：2020.6.23

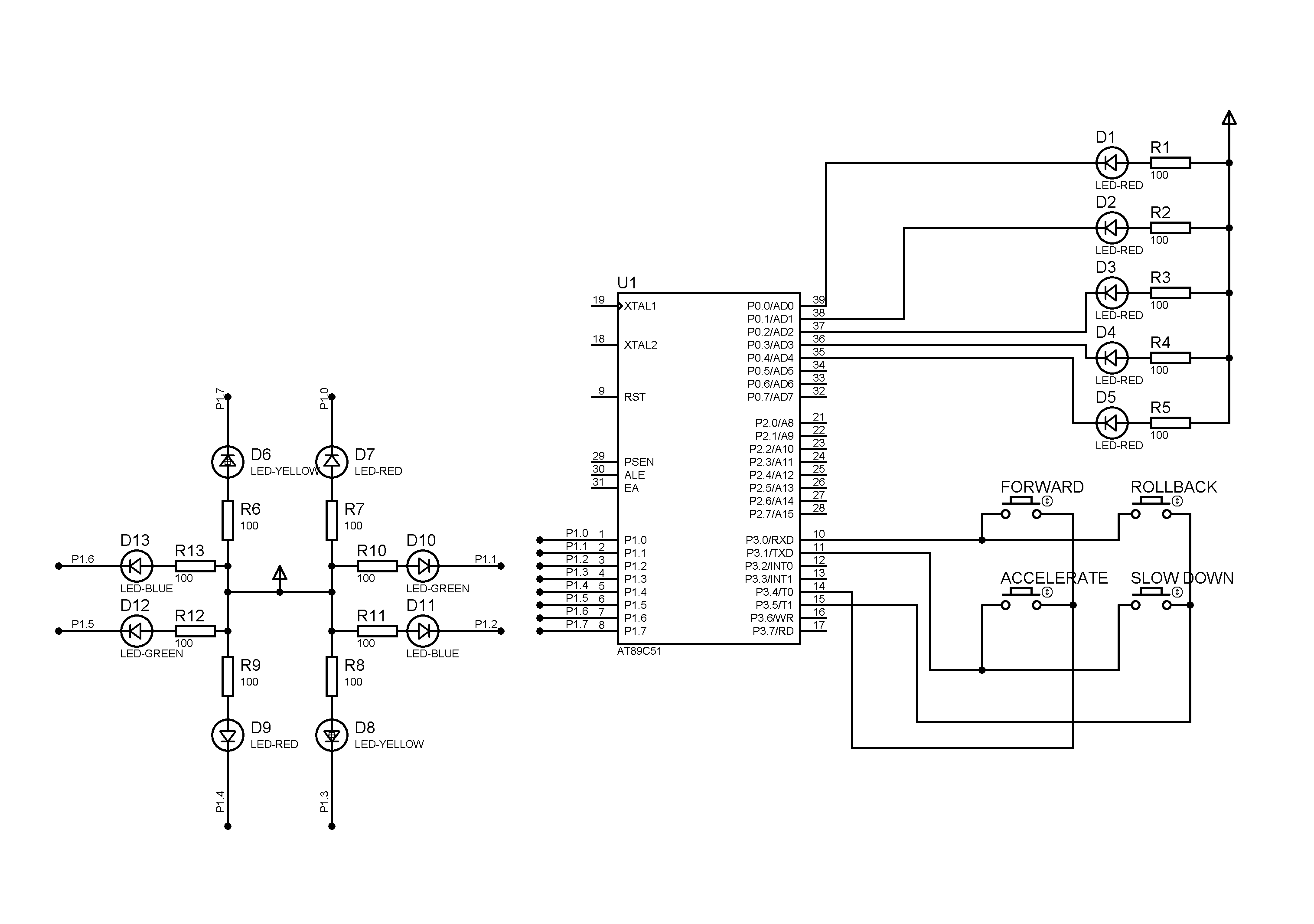
## 实验要求

（1）编写源程序并进行注释，叙述实验原理，画出流程图

（2）记录实验过程

（3）记录Proteus仿真结果

## 实验电路及功能



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **选件编号** | **元件名称** | **参数** | **所在元件库类名** |  | **生产厂家** |
| U1 | AT89C51 |  | Microprocessor ICs | 8051 Family | ATMEL |
| 微处理器 |
| 8051 家族 |
|  |  |  |  |  |  |
| D1-D5,D7,D9 | LED-RED |  | Optoelectronics | LEDs |  |
| 光电子器件 | 发光二极管 |
| D10,D12 | LED-GREEN |  | Optoelectronics | LEDs |  |
| 光电子器件 | 发光二极管 |
| D11,D13 | LED-BLUE |  | Optoelectronics | LEDs |  |
| 光电子器件 | 发光二极管 |
| D6,D8 | LED-YELLOW |  | Optoelectronics | LEDs |  |
| 光电子器件 | 发光二极管 |
| R1-R13 | RES | 100 | Resistors | Generic |  |
| 电阻 | 一般的 |
| FORWARD  ROLLBACK  ACCELERATE  SLOW DOWN | BUTTON |  | Switches & Relays | Switches |  |
| 开关与继电器 | 开关 |

实验功能为，电路图中D6～D13 八个发光二极管构成彩色旋转灯，D1～D5为档位指示灯，一档旋转速度最慢（周期1s，D5亮），二档较快（周期0.8s，D4亮），三档更快（周期0.6s，D3亮），四档再快（周期0.4s，D2亮），五档最快（周期0.2s，D1亮）。四个按键如其英文名称所示，FORWARD和ROLLBACK用于设定旋转方向为顺时针旋转或者逆时针旋转，ACCELERATE和SLOW DOWN用于增快或则减慢旋转速度。

## 实验原理

本例使用C语言编写，并使用线反转法。

程序控制流程是：首先初始化设置默认运行参数，然后读取按键，识别键码，并根据键码的不同执行运行参数调整，最后根据当前的运行参数执行发光二极管D6-D13的轮流旋转。

按键的识别中的键码可以根据电路图中的连接情况，总结出其键值表，即：正向按钮为：0x22,反向按钮为：0x12，加速按钮为0x21，减速按钮为0x11，因此可以定义按键键码表（C语言方式）：

uchar code KEY\_TABLE[]={0x22,0x12,0x21,0x11}; //按键键值表

速度的控制通过控制调用延时程序的次数来决定，假设延时程序的延时长度为5ms。延时程序可以按如下方式实现（假设晶振频率为12MHz）：

void delays()

{

uchar t,ms;

ms=5; //延时5ms

while(ms--) for(t=0;t<120;t++);

}

调用延时程序次数为200次时，周期为1s；次数为160次时，则周期0.8s；次数为140次时，则周期0.7s；次数为120次时，则周期0.6s；次数为100次时，则周期0.5s；次数为80次时，则周期0.4s；次数为40次时，则周期0.2s。这样周期值表（C语言方式）为：

uchar code T\_TABLE[]={200,160,120,80,40}; //周期值表

档位输出也可以按查表形式给出，1档时应给出的档位输出值为0x0F；2档时应给出的档位输出值为0x17；3档时应给出的档位输出值为0x1B；4档时应给出的档位输出值为0x1D；5档时应给出的档位输出值为0x1E。因此档位输出值表为：

uchar code OUT\_TABLE[]={0x0F,0x17,0x1B,0x1D,0x1E}; //档位输出表

下面是思路的流程图：



## 实验过程

（1）根据上述实验内容，参考1.2.2，在Proteus环境下建立图3.11所示原理图。

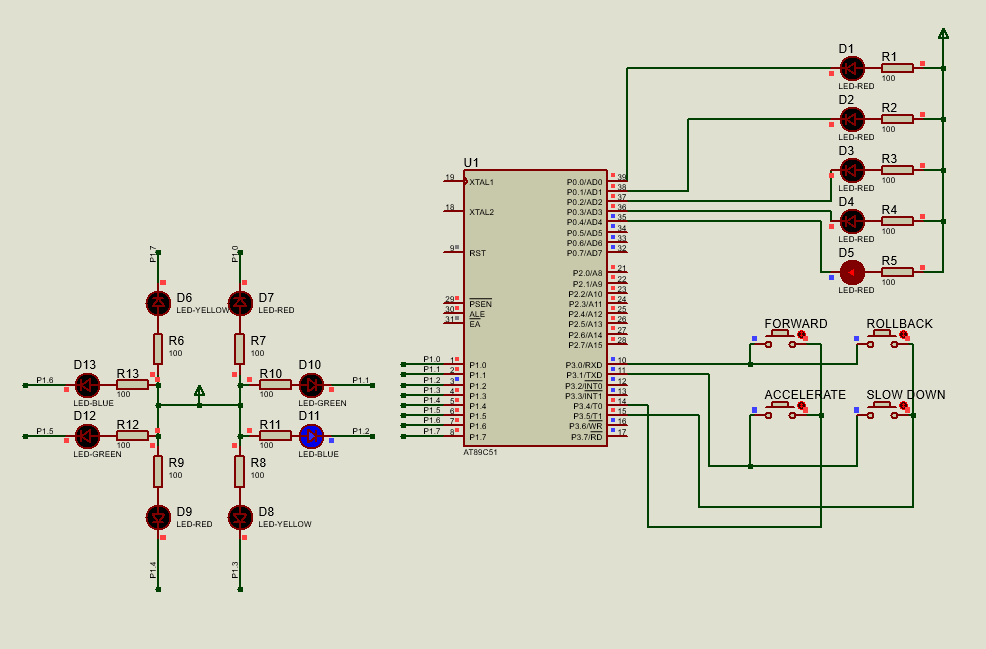
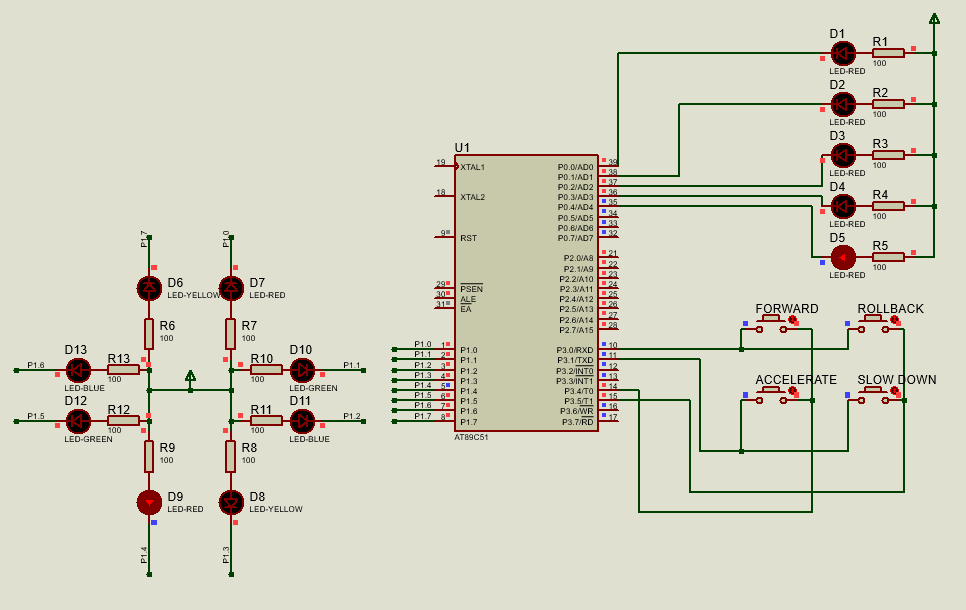
（2）根据（2）和（3）编写控制源程序。

（3）将源程序添加到U1中，并构造（build）该程序。

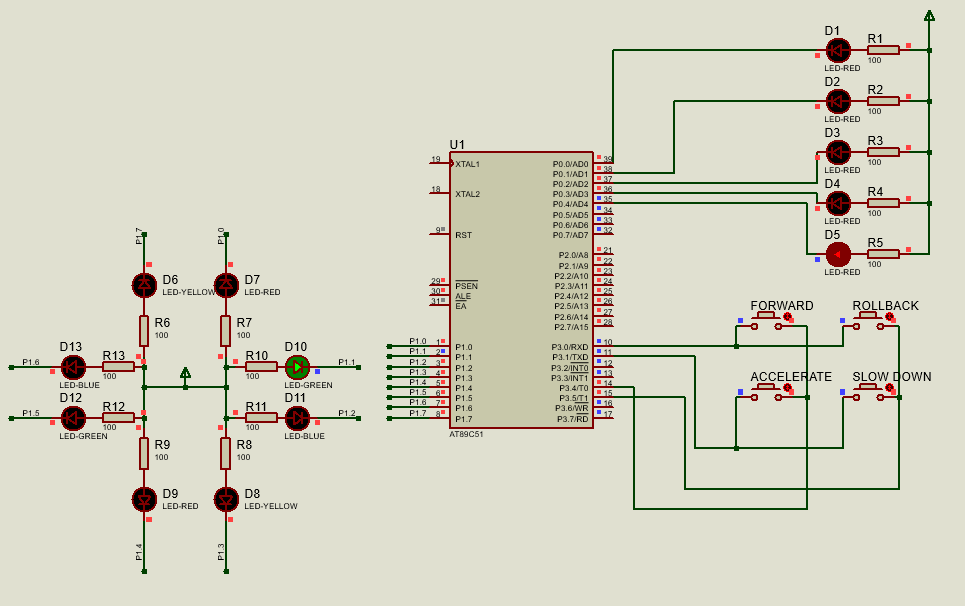
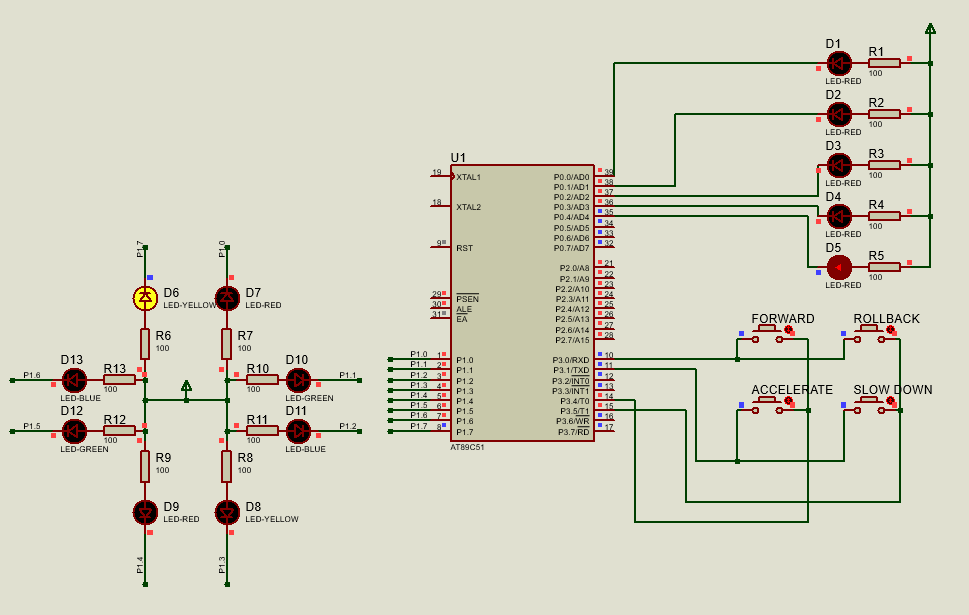
（4）执行仿真过程观察秒表程序功能是否正确。

## 实验结果

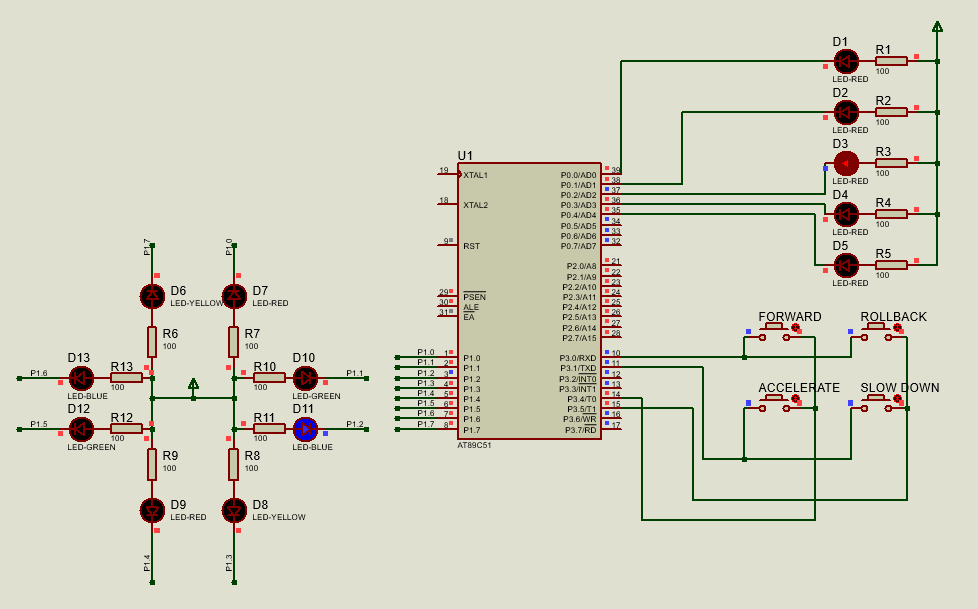
（1）最初，可见处于最低档顺时针旋转，旋转较为缓慢

（2）点击ROLLBACK，可以变成逆时针旋转，当然再点击FORWARD也可回归顺时针

（3）点击ACCELERATE和SLOW DOWN可以改变速度档位



## 实验源程序

#include<reg51.h>

#include<intrins.h>

#define uchar unsigned char

#define uint unsigned int

uchar code T\_TABLE[] = { 200,160,120,80,40 };//周期值表

uchar code OUT\_TABLE[] = { 0x0f,0x17,0x1b,0x1d,0x1e };//档位输出表，分别为1~5档

uchar code KEY\_TABLE[] = { 0x22,0x12,0x21,0x11 };//按键键码表，分别为正向，反向，加速，减速

//延时程序

void delays(uint ms)//参数为多少即为多少毫秒

{

uchar t;

while (ms--)for (t = 0; t < 120; t++);

}

//主程序

void main()

{

uchar DIR = 1;

uchar speed = 0;

uchar temp, key, j;

P0 = OUT\_TABLE[speed];

P1 = 0x0fe;

key = 0;

while (1)

{

P3 = 0x30;//行输出0

temp = P3;//读取列输入

temp = temp & 0x30;//保留有效位

if (temp != 0x30)

{

delays(10);//延时10毫秒

P3 = 0x30;//重新行输出0

temp = P3;//读取列输入

temp = temp & 0x30;//保留有效位

if (temp != 0x30)

{

key = temp;//记录按键所在列

temp = temp | 0x03;//变为列输出0

P3 = temp;

temp = P3 & 0x03;//记录按键所在行

key = temp | key;//合并键码

for (j = 0; j < 4; j++)

{

if (key == KEY\_TABLE[j])

{

key = j;

break;

}

}

}

}

else

{

if (key == 0)DIR = 1;//正转

if (key == 1)DIR = 0;//反转

if (key == 2)

{

if (speed == 4)

speed = speed; else

speed++;

key = ~DIR;

}

if (key == 3)

{

if (speed == 0)

speed = speed; else

speed--;

key = ~DIR;

}

if (DIR == 1)//正转时

{

for (j = 0; j < T\_TABLE[speed]; j++)

{

delays(5);

}

P1 = \_crol\_(P1, 1);//左移

}

if (DIR == 0)//反转时

{

for (j = 0; j < T\_TABLE[speed]; j++)

{

delays(5);

}

P1 = \_cror\_(P1, 1);//右移

}

P0 = OUT\_TABLE[speed];

}

}

}