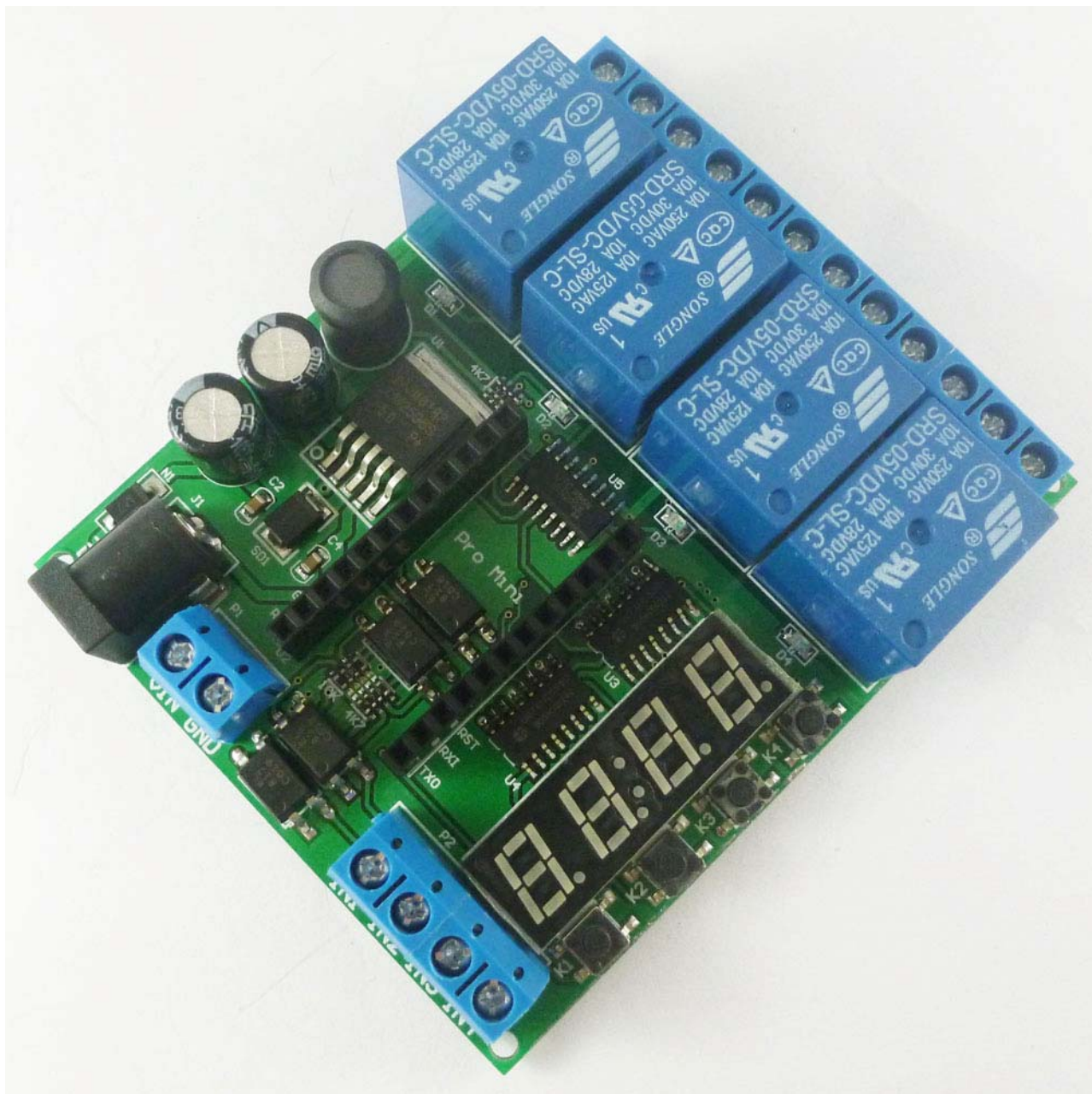
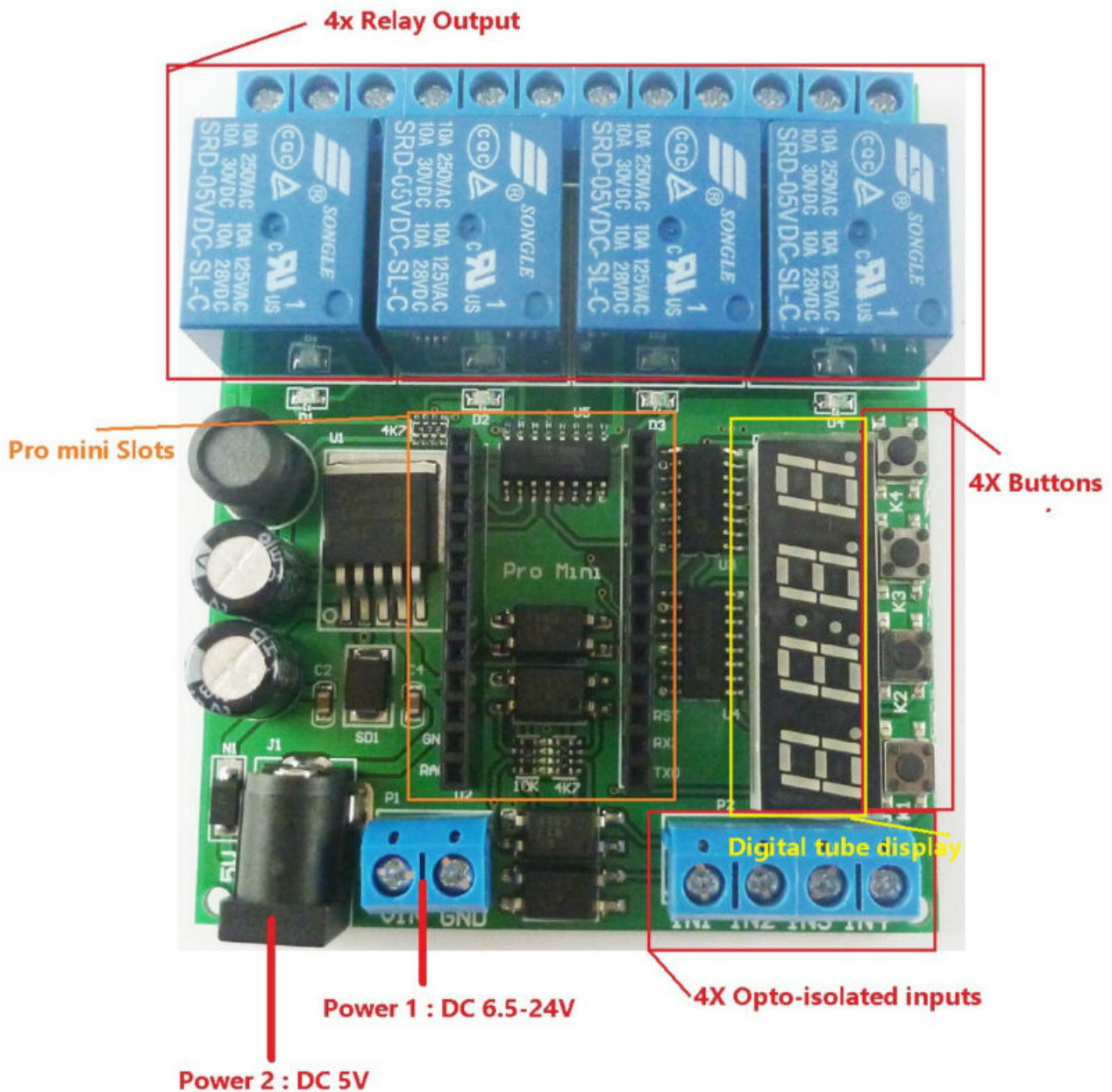


Плата [IO22C04](#) Pro-Mini-Delay-Relay представляет собой шилд для Arduino Pro mini. По сути это четырехканальный реле шилд с дополнительной обвязкой и функциональностью.



По центру платы расположен слот для подключения платы Arduino pro mini. На плате также расположены:

- четыре реле SRD-05VDC;
- блок из четырех семисегментных индикаторов;
- четыре кнопки;
- четыре входа с опторазвязкой (низкий уровень триггера, NPN типа);
- штекер для подключения стабилизированного питания +5В;
- входы для питания 6-24В.



Надо понимать, что данная плата является шилдом и без контроллера Arduino pro mini и загруженного в него скетча работать не может.

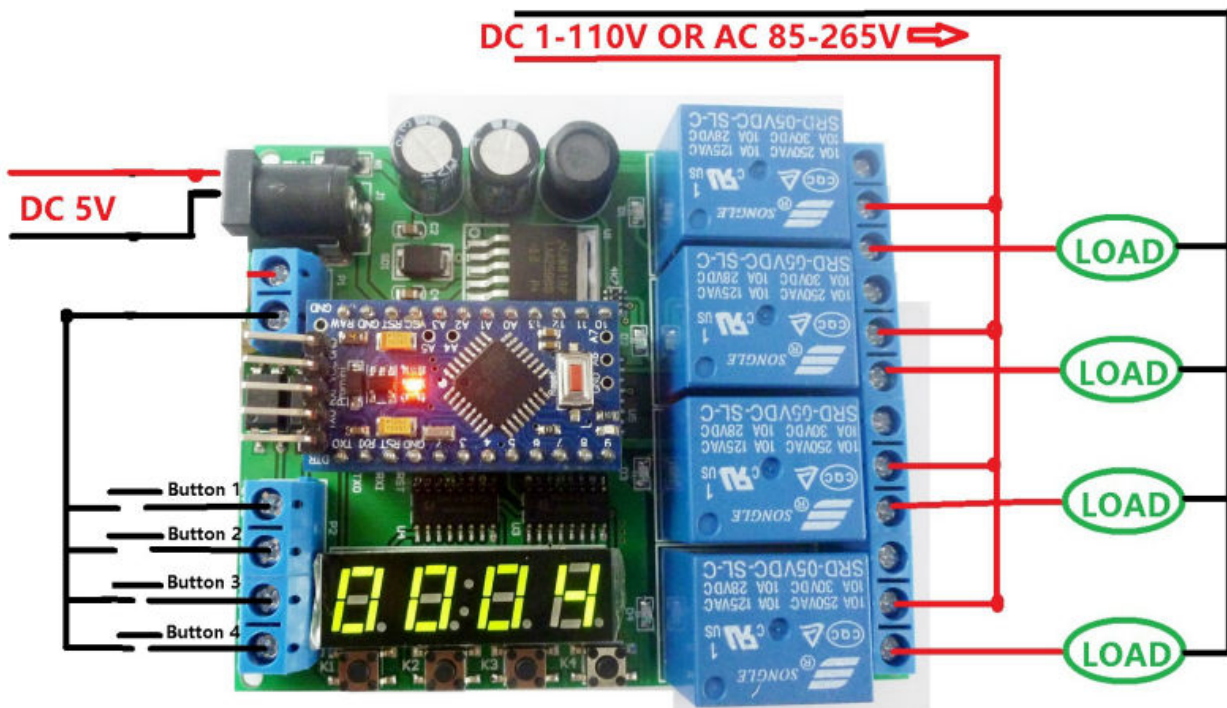
Реле подключены к 4 выходам Arduino через микросхему ULN2003 – набор мощных составных ключей для применения в цепях индуктивных нагрузок. Блок семисегментных индикаторов управляется 3 выводами Arduino по протоколу SPI (2 микросхемы 74HC595 – расширитель выводов).

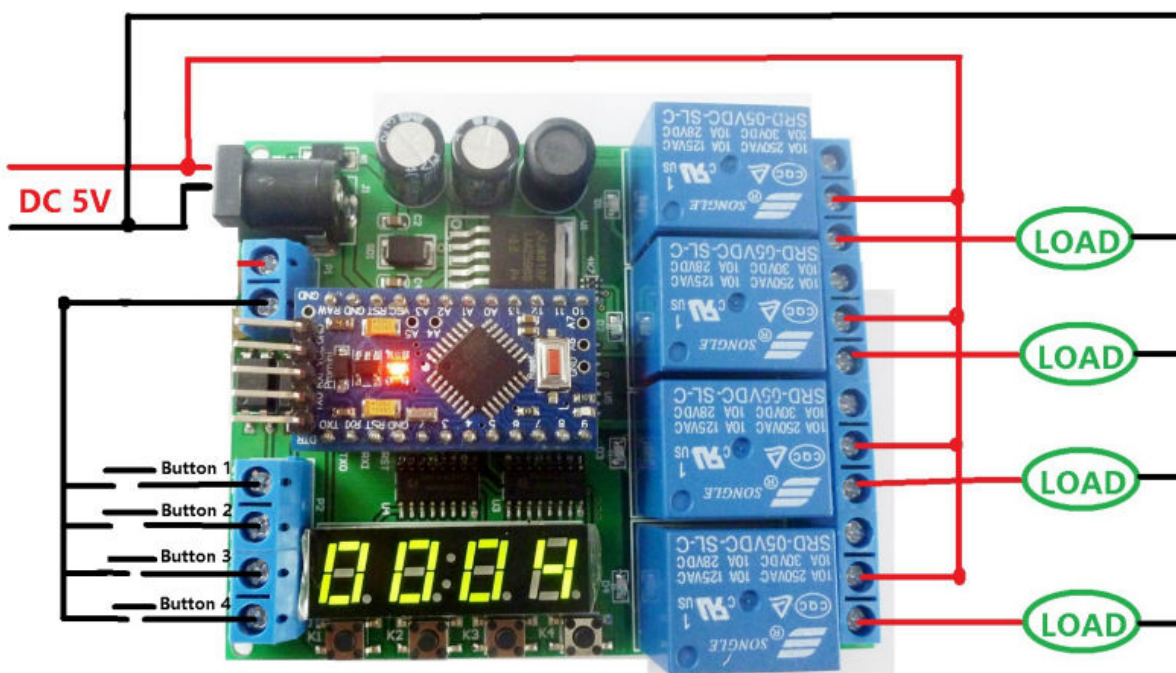
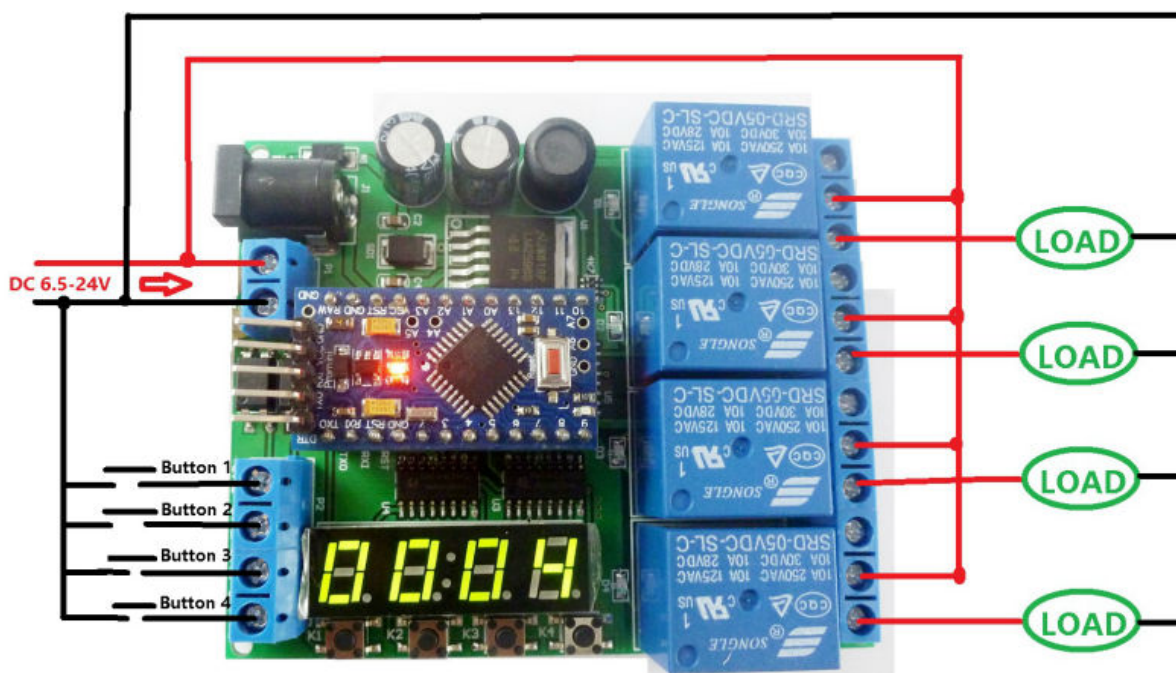
Карта подключения

	Плата	Контакты Arduino
Реле	D1	10
	D2	11
	D3	12
	D4	6

Кнопки на плате	K1	2
	K2	3
	K3	4
	K4	5
Input входы	IN1	A1
	IN2	A0
	IN3	A3
	IN4	A2
SPI	latch	8
	clock	9
	data	7

Варианты схем подключения





Пример скетча, где по нажатию внешней кнопки на определенное время включается соответствующее реле. Значение таймера для каждого реле (от 1 до 9999 сек) устанавливается в скетче. Кнопками К1-К4 выбираем для какого таймера высвечиваются данные на индикаторах.

```
// подключение библиотеки таймера
#include <FlexiTimer2.h>
```

```
#define uchar unsigned char
#define uint unsigned int
```

```

int display_dat;
int key_value;

const int led = 13;

// Пин подключения SPI latch
const int latch = 8;
// Пин подключения SPI o clock
const int clock = 9;
// Пин подключения SPI data
const int data = 7;
// Пины подключения кнопок К1-К4
const int K1 = 2;
const int K2 = 3;
const int K3 = 4;
const int K4 = 5;
// пины подключения внешних кнопок
const int INPUT1 = A1;
const int INPUT2 = A0;
const int INPUT3 = A3;
const int INPUT4 = A2;
// Пины подключения реле
const int j1 = 10;
const int j2 = 11;
const int j3 = 12;
const int j4 = 6;
// Значение таймера (1-9999 секунд) для каждого реле
int relay1_delay_time = 1;
int relay2_delay_time = 2;
int relay3_delay_time = 3;
int relay4_delay_time = 4;

int relay1_time_left;
int relay2_time_left;
int relay3_time_left;
int relay4_time_left;

int relay1_display;
int relay2_display;
int relay3_display;
int relay4_display;

// Значения data для символов
// Character :0,1,2,3,4,5,6,7,8,9,A, b, C, c, d, E, F, H, h, L, n, N, o, P, r, t, U, -, ,

```

```

uchar TUBE_SEG[] =
{0xc0,0xf9,0xa4,0xb0,0x99,0x92,0x82,0xf8,0x80,0x90,0x88,0x83,0xc6,0xa7,0xa1,0
x86,0x8e,0x89,0x8b,0xc7,0xab,0xc8,0xa3,0x8c,0xaf,0x87,0xc1,0xbf,0xff};
// Значения data выбора индикатора
uchar TUBE_NUM[8]={0xfe,0xfd,0xfb,0xf7,0xef,0xdf,0xbf,0x7f};

uchar dat_buf[8];

uchar dat;
uchar com_num;
//*****
// dat : данные для отображения на индикаторе
// com_num : выбор индикатора
//*****
void TubeDisplayOneBit()
{
uchar tube_dat;
uchar bit_num;
tube_dat = TUBE_SEG[dat];
bit_num = ~TUBE_NUM[com_num];

// разрешить передачу данных в 74HC595
digitalWrite(latch, LOW);
// сначала выбор индикатора, потом data
shiftOut(data, clock, MSBFIRST, bit_num);
shiftOut(data, clock, MSBFIRST, tube_dat);
// защелкнуть данные в 74HC595
digitalWrite(latch, HIGH);
}

uchar OneSecondCnt;
void TubeDisplay4Bit(void)
{
if(com_num<3) com_num ++;
else com_num=0;
dat = dat_buf[com_num];
// вывести данные в один индикатор
TubeDisplayOneBit();

OneSecondCnt--;
if(OneSecondCnt==0)
{
OneSecondCnt = 200;
// декремент показаний активных таймеров
if(relay1_time_left>0)

```

```

    {relay1_time_left--; relay1_display = relay1_time_left;}
else
    {digitalWrite(j1, LOW); relay1_display = relay1_delay_time;}
if(relay2_time_left>0)
    {relay2_time_left--; relay2_display = relay2_time_left;}
else
    {digitalWrite(j2, LOW); relay2_display = relay2_delay_time;}
if(relay3_time_left>0)
    {relay3_time_left--; relay3_display = relay3_time_left;}
else
    {digitalWrite(j3, LOW); relay3_display = relay3_delay_time;}
if(relay4_time_left>0)
    {relay4_time_left--; relay4_display = relay4_time_left;}
else
    {digitalWrite(j4, LOW); relay4_display = relay4_delay_time;}
}
// выбор показаний для вывода на индикаторы
if(key_value==0)
    display_dat = relay1_display;
else if(key_value==1)
    display_dat = relay2_display;
else if(key_value==2)
    display_dat = relay3_display;
else
    display_dat = relay4_display;
// показания таймера поразрядно
dat_buf[0] = display_dat/1000;
display_dat = display_dat%1000;
dat_buf[1] = display_dat/100;
display_dat = display_dat%100;
dat_buf[2] = display_dat/10;
dat_buf[3] = display_dat%10;
}

```

```

void setup() {
    // установка режимов для пинов
    pinMode(led, OUTPUT);
    // для SPI
    pinMode(latch, OUTPUT);
    pinMode(clock, OUTPUT);
    pinMode(data, OUTPUT);
    // для кнопок К1-К4
    pinMode(K1, INPUT);
    pinMode(K2, INPUT);
}

```

```
pinMode(K3, INPUT);
pinMode(K4, INPUT);
// для внешних кнопок
pinMode(INPUT1, INPUT);
pinMode(INPUT2, INPUT);
pinMode(INPUT3, INPUT);
pinMode(INPUT4, INPUT);
// для реле
pinMode(j1, OUTPUT);
pinMode(j2, OUTPUT);
pinMode(j3, OUTPUT);
pinMode(j4, OUTPUT);
// запуск таймера каждые 5 мс
FlexiTimer2::set(5, 1.0/1000, TubeDisplay4Bit);
FlexiTimer2::start();
// начальные значения для пинов
digitalWrite(j1, LOW);
digitalWrite(j2, LOW);
digitalWrite(j3, LOW);
digitalWrite(j4, LOW);
```

```
digitalWrite(INPUT1, HIGH);
digitalWrite(INPUT2, HIGH);
digitalWrite(INPUT3, HIGH);
digitalWrite(INPUT4, HIGH);
```

```
digitalWrite(K1, HIGH);
digitalWrite(K2, HIGH);
digitalWrite(K3, HIGH);
digitalWrite(K4, HIGH);
```

```
key_value = 0;
}
```

```
void loop() {
```

```
while(1) {
// определение нажатия внешних кнопок
if(digitalRead(INPUT1)==LOW)
{
relay1_time_left = relay1_delay_time;
digitalWrite(j1, HIGH);
digitalWrite(led, HIGH);}
else
```



```

    {digitalWrite(led, LOW);}
    if(digitalRead(INPUT2)==LOW)
    {
        relay2_time_left = relay2_delay_time;
        digitalWrite(j2, HIGH);
        digitalWrite(led, HIGH);
    }
    else
    { digitalWrite(led, LOW);}
    if(digitalRead(INPUT3)==LOW)
    {
        relay3_time_left = relay3_delay_time;
        digitalWrite(j3, HIGH);
        digitalWrite(led, HIGH);
    }
    else
    { digitalWrite(led, LOW);}
    if(digitalRead(INPUT4)==LOW)
    {
        relay4_time_left = relay4_delay_time;
        digitalWrite(j4, HIGH);
        digitalWrite(led, HIGH);
    }
    else
    { digitalWrite(led, LOW);}
    // определение нажатия кнопок K1-K4
    if (digitalRead(K1)==LOW) {key_value = 0; }
    else if(digitalRead(K2)==LOW) {key_value = 1; }
    else if(digitalRead(K3)==LOW) {key_value = 2; }
    else if(digitalRead(K4)==LOW) {key_value = 3; }
}
}

```