

فصل سوم

برنامه های مربوط به پروژه

Codevision AVR



برای نوشتن برنامه ها باید در ابتدا پورت سریال بلد بود

من ابتدا برنامه فرستنده را بصورت کلی بیان میکنم:

ما ۶ کلید داریم که به ازای هر کلید یک حرکت در روبات انجام میشود. و در اینجا برای هر کلید یک کد هگز غیر از صفر قرار داده ام که با هر بار فشار دادن کلید این کار انجام شود و یک سری تنظیمات یوزات هم هست مثل باند و نوع ماژول که باید تنظیم شود و در نهایت یک حلقه گذاشته ام که به ازای هر کلید یک کد بفرستد.

برنامه فرستنده به شرح زیر است:

ابتدا به تعریف کتابخانه ها و میکرو پرداختیم

```
#include <mega16.h>
```

```
#include <delay.h>
```

```
//Standard Input/Output functions
```

```
#include <stdio.h>
```

برای هر کلید یک کد هگز قرار داده ام

```
int up=0x01;
```

```
int down=0x02;
```

```
int left=0x03;
```

```
int right=0x04;
```

```
int rl=0x05;
```

```
int ll=0x06;
```

```
//Declare your global variables here
```

شروع برنامه

```
void main(void(
```

```
)
```

```
//Declare your local variables here
```

```
//Input/Output Ports initialization
```

```
//Port A initialization
```

```
//Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In  
Func0=In
```

```
//State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
```

تعریف نوع ورودی ها و خروجی ها

پورت آ بعنوان ورودی برای کلید های کنترل

```
PORTA=0xff;
```

```
DDRA=0x00;
```

```
//Port B initialization
```

```
//Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In  
Func0=In
```

```
//State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
```

```
PORTB=0x00;
```

```
DDRB=0x00;
```

```
//Port C initialization
```

```
//Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In  
Func0=In
```

```
//State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
```

```
PORTC=0x00;
```

```
DDRC=0x00;
```

```
Port D initialization //
```

```
Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In //  
Func0=In
```

```
State7=T State6=T State5=T State4=T State3=T State2=T State1=T //  
State0=T
```

پورت دی بعنوان خروجی برای ماژول

```
;PORTD=0x00
```

```
;DDRD=0x00
```

```
Timer/Counter 0 initialization //
```

```
Clock source: System Clock //
```

```
Clock value: Timer 0 Stopped //
```

```
Mode: Normal top=FFh //
```

```
OC0 output: Disconnected //
```

```
;TCCR0=0x00
```

```
;TCNT0=0x00
;OCR0=0x00

Timer/Counter 1 initialization //
Clock source: System Clock //
Clock value: Timer 1 Stopped //
Mode: Normal top=FFFFh //
.OC1A output: Discon //
.OC1B output: Discon //
Noise Canceler: Off //
Input Capture on Falling Edge //
Timer 1 Overflow Interrupt: Off //
Input Capture Interrupt: Off //
Compare A Match Interrupt: Off //
Compare B Match Interrupt: Off //
;TCCR1A=0x00
;TCCR1B=0x00
;TCNT1H=0x00
;TCNT1L=0x00
;ICR1H=0x00
;ICR1L=0x00
;OCR1AH=0x00
;OCR1AL=0x00
;OCR1BH=0x00
;OCR1BL=0x00

Timer/Counter 2 initialization //
Clock source: System Clock //
Clock value: Timer 2 Stopped //
Mode: Normal top=FFh //
OC2 output: Disconnected //
;ASSR=0x00
;TCCR2=0x00
;TCNT2=0x00
;OCR2=0x00

External Interrupt(s) initialization //
INT0: Off //
INT1: Off //
```

```
INT2: Off //
;MCUCR=0x00
MCUCSR=0x00
```

تنظیمات یوزارت

```
Timer(s)/Counter(s) Interrupt(s) initialization //
;TIMSK=0x00
```

```
USART initialization //
Communication Parameters: 8 Data, 1 Stop, No Parity //
USART Receiver: Off //
USART Transmitter: On //
USART Mode: Asynchronous //
USART Baud Rate: 110 //
;UCSRA=0x00
;UCSRB=0x08
;UCSRC=0x86
;UBRRH=0x08
;UBRRL=0xE0
```

```
Analog Comparator initialization //
Analog Comparator: Off //
Analog Comparator Input Capture by Timer/Counter 1: Off //
;ACSR=0x80
;SFIOR=0x00
```

```
{while (1
}
Place your code here //
```

اگر پورت آ ۰ مساوی صفر شد آنگاه کد هگز جلو را میفرستد

```
}while(PINA.0==0
;delay_ms(100
if(PINA.0==0
;printf("%x",up
{ ;delay_ms(100
```

اگر پورت آ ۱ مساوی صفر شد آنگاه کد هگز عقب را میفرستد

```
} while(PINA.1==0  
;delay_ms(100  
{ if(PINA.1==0  
;printf("%x",down  
;delay_ms(100  
{
```

اگر پورت آ ۲ مساوی صفر شد آنگاه کد هگز چپ را میفرستد

```
} while(PINA.2==0  
;delay_ms(100  
{ if(PINA.2==0  
;printf("%x",left  
;delay_ms(100  
{
```

اگر پورت آ ۳ مساوی صفر شد آنگاه کد هگز راست را میفرستد

```
} while(PINA.3==0  
;delay_ms(100  
{ if(PINA.3==0  
;printf("%x",right  
;delay_ms(100  
{
```

اگر پورت آ ۴ مساوی صفر شد آنگاه کد هگز بلند کردن را میفرستد

```
} while(PINA.4==0  
;delay_ms(100  
{ if(PINA.4==0  
;printf("%x", rl  
;delay_ms(100  
{
```

اگر پورت آ ۵ مساوی صفر شد آنگاه کد هگز پایین آوردن را میفرستد

```
} while(PINA.5==0  
;delay_ms(100  
{ if(PINA.5==0
```

```

;printf("%x",ll
;delay_ms(100
{

;printf("%x",0x00

;{
{

```

برای گیرنده هم ابتدا تنظیمات یوزارت را انجام داده‌ام ام باید باند فرستنده و گیرنده یکی باشد. و بعد گفته ام اگر کد هگز فرضاً ۲ رسید به عقب حرکت کند

برنامه گیرنده به شرح زیر است :

تعریف کتابخانه و میکرو

```

#include <mega32.h>

// Alphanumeric LCD Module functions

#asm

.equ __lcd_port=0x1B ;PORTA

#endasm

#include <lcd.h>

#include <delay.h>

// Standard Input/Output functions

#include <stdio.h>

```



```
// Declare your global variables here
```

```
//char text[15];
```

```
//int i;
```

```
char litter=0;
```

```
void main(void)
```

```
{
```

```
// Declare your local variables here
```

```
// Input/Output Ports initialization
```

```
// Port A initialization
```

```
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In  
Func0=In
```

```
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T  
State0=T
```

پورت آ بعنوان خروجی برای چراغهای لیفتراک

```
PORTA=0x00;
```

```
DDRA=0xff;
```

```
// Port B initialization
```

```
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In  
Func0=In
```

```
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T  
State0=T
```

پورت بی بعنوان خروجی برای موتورهای لیفتراک

```
PORTB=0x00;

DDRB=0xff;

// Port C initialization

// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In
Func0=In

// State7=T State6=T State5=T State4=T State3=T State2=T State1=T
State0=T

PORTC=0x00;

DDRC=0x00;

// Port D initialization

// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In
Func0=In

// State7=T State6=T State5=T State4=T State3=T State2=T State1=T
State0=T

PORTD=0x00;

DDRD=0x00;

// Timer/Counter 0 initialization

// Clock source: System Clock

// Clock value: Timer 0 Stopped

// Mode: Normal top=FFh

// OC0 output: Disconnected

TCCR0=0x00;
```

```
TCNT0=0x00;
OCR0=0x00;

// Timer/Counter 1 initialization
// Clock source: System Clock
// Clock value: Timer 1 Stopped
// Mode: Normal top=FFFFh
// OC1A output: Discon.
// OC1B output: Discon.
// Noise Canceler: Off
// Input Capture on Falling Edge
// Timer 1 Overflow Interrupt: Off
// Input Capture Interrupt: Off
// Compare A Match Interrupt: Off
// Compare B Match Interrupt: Off
TCCR1A=0x00;
TCCR1B=0x00;
TCNT1H=0x00;
TCNT1L=0x00;
ICR1H=0x00;
ICR1L=0x00;
OCR1AH=0x00;
OCR1AL=0x00;
OCR1BH=0x00;
OCR1BL=0x00;
```

```
// Timer/Counter 2 initialization
// Clock source: System Clock
// Clock value: Timer 2 Stopped
// Mode: Normal top=FFh
// OC2 output: Disconnected
ASSR=0x00;
TCCR2=0x00;
TCNT2=0x00;
OCR2=0x00;

// External Interrupt(s) initialization
// INT0: Off
// INT1: Off
// INT2: Off
MCUCR=0x00;
MCUCSR=0x00;

// Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=0x00;

// USART initialization
// Communication Parameters: 8 Data, 1 Stop, No Parity
// USART Receiver: On
// USART Transmitter: Off
```

تنظیمات یوزارت

```
// USART Mode: Asynchronous
// USART Baud Rate: 110
UCSRA=0x00;
UCSRB=0x10;
UCSRC=0x86;
UBRRH=0x08;
UBRRL=0xE0;

// Analog Comparator initialization
// Analog Comparator: Off
// Analog Comparator Input Capture by Timer/Counter 1: Off
ACSR=0x80;
SFIOR=0x00;

// LCD module initialization
lcd_init(16);

while (1)
{
    // Place your code here
    litter=getchar();
    while(litter=='1')
    {
        اگر ورودی ما کد هگز ۱ شد به ترتیب خروجی ها به صورت زی شوند پورت بی برای حرکت به جلو و
        پورت آ برای روشن شدن چراغها
    }
}
```

```
PORTB=0b00001010;

PORTA=0b00000011;

litter=getchar();

if( litter=='0'){

    PORTB=0x00;

    PORTA=0x00;

    break;

}

}

while(litter=='2')

{

    PORTB=0b000000101;

    PORTA=0b000001100;

    litter=getchar();

    if( litter=='0'){

        PORTB=0x00;

        PORTA=0x00;

        break;

    }

}

while(litter=='3')

{

    PORTB=0b000000010;

    PORTA=0b000000101;

    litter=getchar();
```

```
if( litter=='0'){  
    PORTB=0x00;  
    PORTA=0x00;  
    break;  
}  
}  
  
while(litter=='4')  
{  
    PORTB=0b00001000;  
    PORTA=0b00001010;  
    litter=getchar();  
    if( litter=='0'){  
        PORTB=0x00;  
        PORTA=0x00;  
        break;  
    }  
}  
  
while(litter=='5')  
{  
    PORTB=0b01010000;  
    litter=getchar();  
    if( litter=='0'){  
        PORTB=0x00;  
        PORTA=0x00;  
        break;  
    }  
}
```

```
        }  
    }  
  
while(litter=='6')  
{  
    PORTB=0b10100000;  
    litter=getchar();  
    if( litter=='0'){  
        PORTB=0x00;  
        PORTA=0x00;  
        break;  
    }  
}  
litter=0;  
delay_ms(100);  
};
```