

## پروژه ساعت دیجیتال با کمک ATmega16 و سون سگمنت



در این پروژه ما می خواهیم یک ساعت دیجیتال طراحی و پیاده سازی کنیم با کمک Atmel AVR ATmega16 و سون سگمنت.

به همین دلیل بهتر است برای تکمیل آموزش روش اتصال و کار با سون سگمنت را یاد بگیرید.

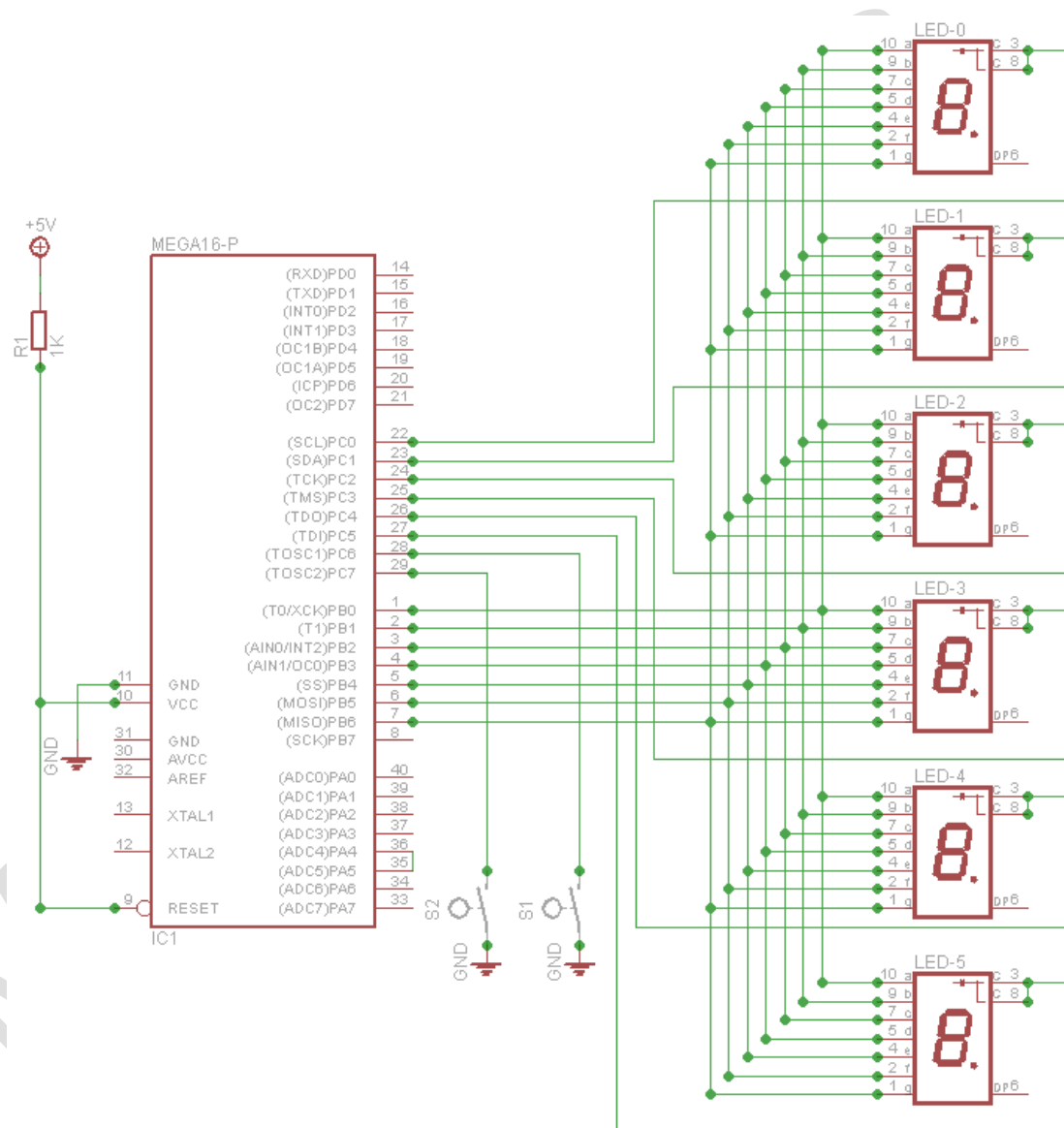
توجه : اگرچه این پروژه با ATmega16 طراحی شده است اما شما می توانید از سایر میکروکنترلر ها مثل ATmega32 و ... نیز استفاده کنید.

### روش کار ساعت دیجیتال با کمک ATmega16

ساعت دیجیتال ما به این شکل عمل می کند ، هنگامی که جریان وارد مدار می شود (روشن می شود) ساعت روی "۰۰:۰۰:۰۰" تنظیم می شود ("ثانیه:دقیقه:ساعت") ، برای تنظیم زمان دو دکمه وجود دارد . دکمه SW1 برای تنظیم دقیقه ، هنگامی که این دکمه را فشار دهیم تا زمانی که به عدد ۵۹ برسد افزایش می یابد و بعد از آن دوباره از صفر شمارش را آغاز می کند ( همانند ساعت های دیجیتال معمولی ) . دکمه SW2 نیز برای تنظیم ساعت است .

## سخت افزار ساعت دیجیتال با کمک ATmega16

شکل زیر دیاگرام اتصال ATmega16 و سون سگمنت ساعت دیجیتال را نمایش می دهد. یکی از ویژگی های این پروژه این است که تمامی سون سگمنت ها به وسیله یک پورت (PORTB) درایو (راه اندازی) می شوند. میکروکنترلر از طریق PortC مشخص می کند که کدام سون سگمنت باید عمل کند. توجه داشته باشید که سون سگمنت ها از نوع کاتد مشترک هستند.



نکته مهم: برای اینکه ساعت دیجیتال ما به درستی کار کند باید اسیلاتور (نوسان ساز) داخلی میکروکنترلر ATmega16 بر روی 4 MHz قرار داده شده و فیوز بیت آن به درستی تنظیم شود.

## نرم افزار ساعت دیجیتال با کمک ATmega16

در زیر کد پیاده سازی شده ساعت دیجیتال با زبان C و کامپایلر AVR Studio 5 را می توانید مشاهده نمایید.

```

/*
 * DigitalClock.c
 * Written in AVR Studio 5
 * Compiler: AVR GNU C Compiler (GCC)
 *
 * Created: 12/10/2011 1:17:19 PM
 * Author: AVR Tutorials
 * Website: www.AVR-Tutorials.com
 */
#define F_CPU 4000000UL
#include <avr/delay.h>
#include <avr/io.h>
#include <avr/interrupt.h>

#define SegDataPort PORTB
#define SegDataPin PINB
#define SegDataDDR DDRB

#define SegCntrlPort PORTC
#define SegCntrlPin PINC
#define SegCntrlDDR DDRC

/*Global Variables Declarations*/
unsigned char hours = 0;
unsigned char minutes = 0;
unsigned char seconds = 0;

/*Function Declarations*/
/*****
***/
/*Decimal Digit (0-9) to Seven Segment Values Encoder*/
unsigned char DigitTo7SegEncoder(unsigned char digit, unsigned char
common);

/*Timer Counter 1 Compare Match A Interrupt Service Routine/Interrupt
Handler*/
ISR(TIMER1_COMPA_vect);

```

```

/*Main Program*/
/*****
***/
int main(void)
{
    SegDataDDR = 0xFF;
    SegCntrlDDR = 0x3F;
    SegCntrlPort = 0xFF;

    TCCR1B = (1<<CS12|1<<WGM12);
    OCR1A = 15625-1;
    TIMSK = 1<<OCIE1A;
    sei();

    while(1)
    {
        /* Set Minutes when SegCntrl Pin 6 Switch is Pressed*/
        if((SegCntrlPin & 0x40) == 0 )
        {
            _delay_ms(200);
            if(minutes < 59)
                minutes++;
            else
                minutes = 0;
        }
        /* Set Hours when SegCntrl Pin 7 Switch is Pressed*/
        if((SegCntrlPin & 0x80) == 0 )
        {
            _delay_ms(200);
            if(hours < 23)
                hours++;
            else
                hours = 0;
        }

        SegDataPort = DigitTo7SegEncoder(seconds%10,1);
        SegCntrlPort = ~0x01;
        SegDataPort = DigitTo7SegEncoder(seconds/10,1);
        SegCntrlPort = ~0x02;
        SegDataPort = DigitTo7SegEncoder(minutes%10,1);
        SegCntrlPort = ~0x04;
        SegDataPort = DigitTo7SegEncoder(minutes/10,1);
        SegCntrlPort = ~0x08;
        SegDataPort = DigitTo7SegEncoder(hours%10,1);
        SegCntrlPort = ~0x10;
        SegDataPort = DigitTo7SegEncoder(hours/10,1);
        SegCntrlPort = ~0x20;

    }

    return 0;
}

/*
* Function Description:
* Encode a Decimal Digit 0-9 to its Seven Segment Equivalent.
*
* Function Arguments:
* digit - Decimal Digit to be Encoded
* common - Common Anode (0), Common Cathode(1)

```

```

* SegVal - Encoded Seven Segment Value
*
* Connections:
* Encoded SegVal is return in the other G-F-E-D-C-B-A that is A is the
least
* significant bit (bit 0) and G bit 6.
*/
unsigned char DigitTo7SegEncoder(unsigned char digit, unsigned char common)
{
    unsigned char SegVal;

    switch(digit)
    {
        case 0: if(common == 1) SegVal = 0b00111111;
                else SegVal = ~0b00111111;
                break;
        case 1: if(common == 1) SegVal = 0b00000110;
                else SegVal = ~0b00000110;
                break;
        case 2: if(common == 1) SegVal = 0b01011011;
                else SegVal = ~0b01011011;
                break;
        case 3: if(common == 1) SegVal = 0b01001111;
                else SegVal = ~0b01001111;
                break;
        case 4: if(common == 1) SegVal = 0b01100110;
                else SegVal = ~0b01100110;
                break;
        case 5: if(common == 1) SegVal = 0b01101101;
                else SegVal = ~0b01101101;
                break;
        case 6: if(common == 1) SegVal = 0b01111101;
                else SegVal = ~0b01111101;
                break;
        case 7: if(common == 1) SegVal = 0b00000111;
                else SegVal = ~0b00000111;
                break;
        case 8: if(common == 1) SegVal = 0b01111111;
                else SegVal = ~0b01111111;
                break;
        case 9: if(common == 1) SegVal = 0b01101111;
                else SegVal = ~0b01101111;
                break;
    }
    return SegVal;
}

/*Timer Counter 1 Compare Match A Interrupt Service Routine/Interrupt
Handler*/
ISR(TIMER1_COMPA_vect)
{
    seconds++;

    if(seconds == 60)
    {
        seconds = 0;
        minutes++;
    }
    if(minutes == 60)
    {
        minutes = 0;
        hours++;
    }
}

```

```
}  
if(hours > 23)  
    hours = 0;  
}
```

منبع : [www.avr-tutorials.com](http://www.avr-tutorials.com)

ترجمه : [www.mohandes.org](http://www.mohandes.org)

MOHANDES.ORG