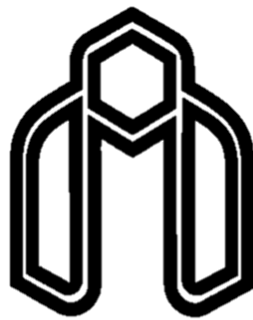


هو العليم



دانشگاه صنعتی شاهرود

درس آزمایشگاه ریزپردازنده

نیم سال دوم ۹۹-۹۸

دانشکده برق

شبیه سازی فرکانس متر

تهیه و تنظیم: حسن رضائی نسب - ۹۶۲۲۷۴۳

فرکانس کلاک را ۴ مگاهرتز در نظر می‌گیریم و از Atmega32 استفاده می‌کنیم.

از تایمر صفر استفاده می‌کنیم و تایمر را در این مد طراحی می‌کنیم.

n را برابر ۸ در نظر می‌گیریم و clock value را برابر با ۵۰۰ کیلو هرتز قرار می‌دهیم و خروجی را به صورت toggle تعریف می‌کنیم.

مقدار TCNT0 (مقدار اولیه) را مشابه مثال گفته شده در ویدیو برابر با 0x83 قرار می‌دهیم و D.C. را همان ۵۰٪ در نظر می‌گیریم.

برای اندازه‌گیری فرکانس در مد نرمال، از پایه ۸ پورت B کمک می‌گیریم و فرکانس متر را به آن متصل می‌کنیم.

متن کد برنامه در حالت Normal به صورت زیر است:

```
#include <mega32.h>

interrupt [TIM0_OVF] void timer0_ovf_isr(void)
{
    TCNT0=0x83;
    PORTB.7=~PORTB.7;
}

void main(void)
{
    DDRB=0x80;
    PORTB=0;

    // Timer/Counter 0 initialization
    // Clock source: System Clock
    // Clock value: 500.000 kHz
    // Mode: Normal top=0xFF
    // OC0 output: Toggle on compare match
    // Timer Period: 0.25 ms
    // Output Pulse(s):
```

```
// OC0 Period: 0.5 ms Width: 0.25 ms

TCCR0=(0<<WGM00) | (0<<COM01) | (1<<COM00) | (0<<WGM01) | (0<<CS02)
| (1<<CS01) | (0<<CS00);

TCNT0=0x83;

OCR0=0;

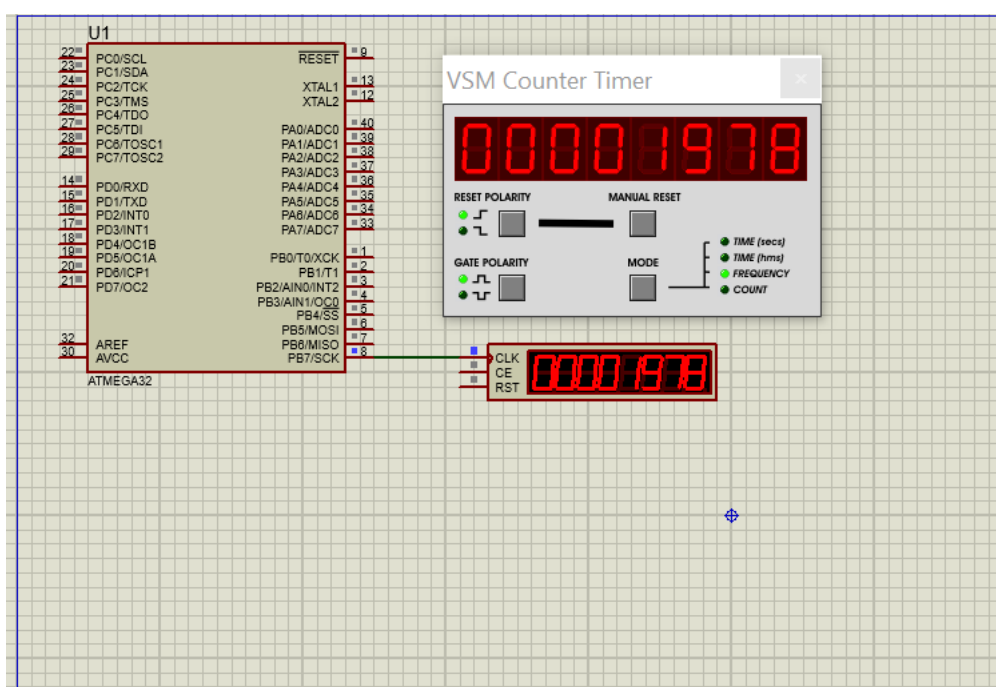
TIMSK=0x01;

// Global enable interrupts

#asm("sei")

while (1)
{
}
}
```

تصویری از شبیه سازی در این مد در پروتئوس به صورت زیر است:



همانطور که مشاهده می شود فرکانس ۲ کیلوهرتز اندازه گرفته شده است. (با اندکی چشم پوشی)

حال تایمر را در حالت CTC بررسی می‌کنیم.

فرکانس کلاک را ۴ مگاهرتز در نظر می‌گیریم و از Atmega32 استفاده می‌کنیم. مشابه بخش فوق یک پالس با فرکانس ۲ کیلو هرتز تولید می‌کنیم.

دوباره از تایمر صفر استفاده می‌کنیم و تایمر را در این مد طراحی می‌کنیم.

n را برابر ۸ در نظر می‌گیریم و clock value را برابر با ۵۰۰ کیلو هرتز قرار می‌دهیم و خروجی را به صورت toggle تعریف می‌کنیم.

برای محاسبه OCR0 داریم:

$$2k = \frac{4M}{2 * 8(1 + OCR0)} \quad , \quad OCR0 = (0x7C)$$

برای اندازه‌گیری و نمایش فرکانس در مد CTC، از پایه ۴ پورت B استفاده می‌کنیم.

متن کد برنامه در حالت CTC به صورت زیر است:

```
#include <mega32.h>
void main(void)
{
    DDRB=0x08;
    PORTB=0;

    // Timer/Counter 0 initialization
    // Clock source: System Clock
    // Clock value: 500.000 kHz
    // Mode: CTC top=OCR0
    // OC0 output: Toggle on compare match
    // Timer Period: 0.25 ms
    // Output Pulse(s):
    // OC0 Period: 0.5 ms Width: 0.25 ms
    TCCR0=(0<<WGM00) | (0<<COM01) | (1<<COM00) | (1<<WGM01) | (0<<CS02)
    | (1<<CS01) | (0<<CS00);
```

```
TCNT0=0x00;
```

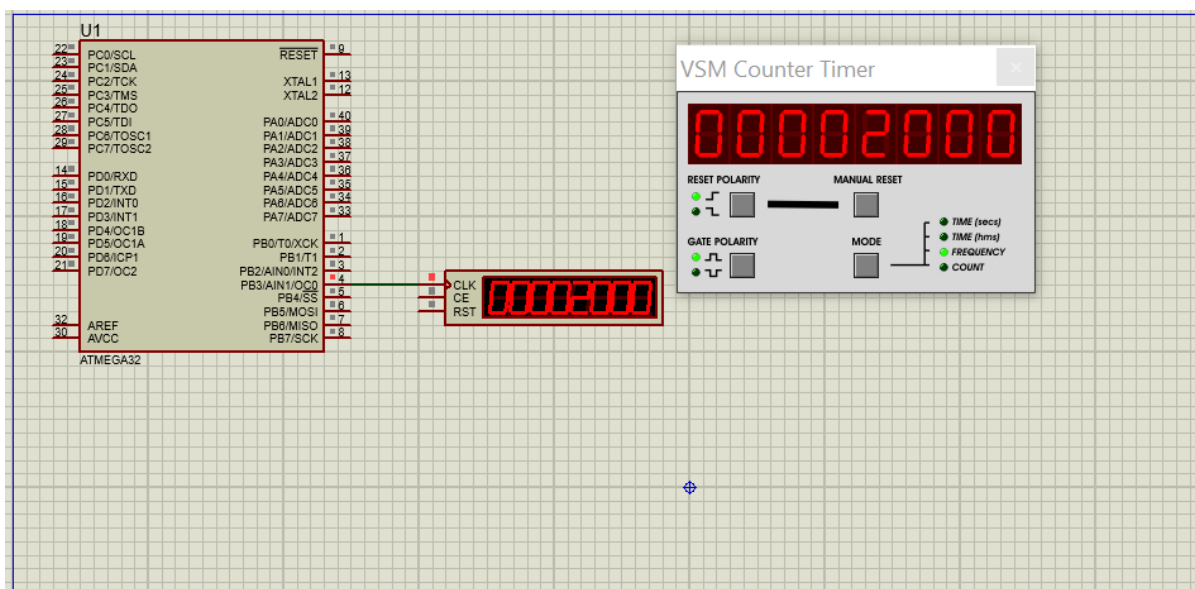
```
OCR0=0x7C;
```

```
while (1)
```

```
{};
```

```
}
```

تصویری از شبیه سازی در این مد در پروتئوس به صورت زیر است:



در این بخش تایمر را در حالت Fast PWM بررسی می‌کنیم و فرکانس را اندازه می‌گیریم.

فرکانس کلاک را ۴ مگاهرتز در نظر می‌گیریم و از Atmega32 استفاده می‌کنیم. مشابه بخش فوق یک پالس با فرکانس ۲ کیلو هرتز تولید می‌کنیم.

دوباره از تایمر صفر استفاده می‌کنیم و تایمر را در این مد طراحی می‌کنیم.

در این حالت n را برابر ۱۶ در نظر می‌گیریم و clock value را برابر با ۵۰۰ کیلو هرتز قرار می‌دهیم و خروجی را به صورت non inverting تعریف می‌کنیم.

در این حالت هم D.C. را ۵۰٪ در نظر می‌گیریم.

برای محاسبه TOP داریم:

$$2k = \frac{4M}{16 * (1 + TOP)} \quad , \quad TOP = (0x7C)$$

برای محاسبه OCR0 داریم:

$$50\% = \frac{OCR0}{1 + 124 - 0} * 100 \quad , \quad OCR0 = (0xFA)$$

متن کد برنامه در حالت Fast PWM به صورت زیر است:

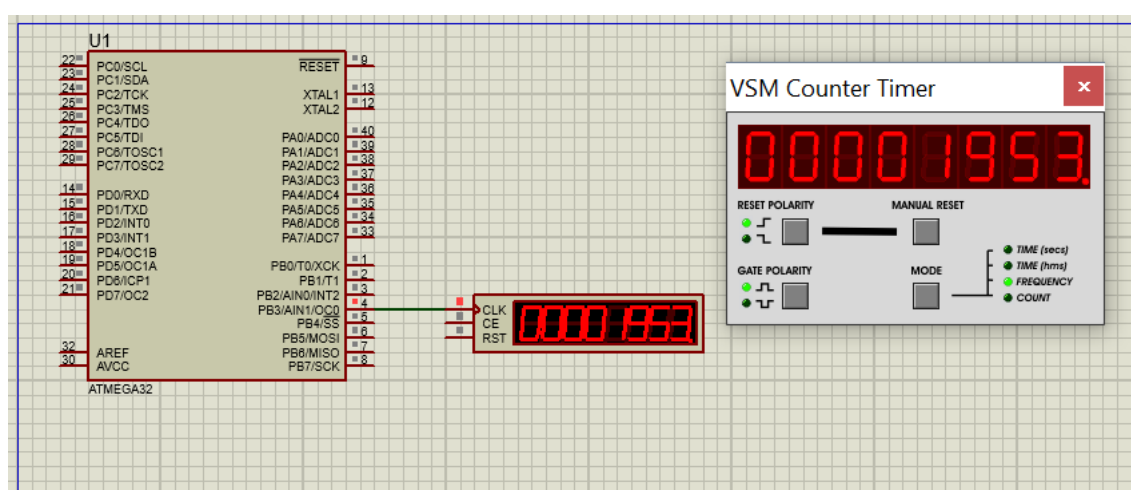
```
#include <mega32.h>
void main(void)
{
  DDRB=0x08;
  PORTB=0;
  // Timer/Counter 0 initialization
  // Clock source: System Clock
  // Clock value: 500.000 kHz
  // Mode: Fast PWM top=0xFF
  // OC0 output: Non-Inverted PWM
  // Timer Period: 0.512 ms
  // Output Pulse(s):
  // OC0 Period: 0.512 ms Width: 0.512 ms
```

```

TCCR0=(1<<WGM00) | (1<<COM01) | (0<<COM00) | (1<<WGM01) | (0<<CS02)
| (1<<CS01) | (0<<CS00);
TCNT0=0;
OCR0=0x7C;
while (1)
{
}
}

```

تصویری از شبیه سازی در این مد در پروتئوس به صورت زیر است:



همانطور که مشاهده می شود فرکانس ۲ کیلوهرتز اندازه گرفته شده است. (با اندکی چشم پوشی)

در بخش آخر تایمر را در حالت Phase Currect بررسی می کنیم و فرکانس را اندازه می گیریم.

فرکانس کلاک را ۴ مگاهرتز در نظر می گیریم و از Atmega32 استفاده می کنیم. مشابه بخش فوق یک پالس با فرکانس ۲ کیلو هرتز تولید می کنیم.

دوباه از تایمر صفر استفاده می کنیم و تایمر را در این مد طراحی می کنیم.

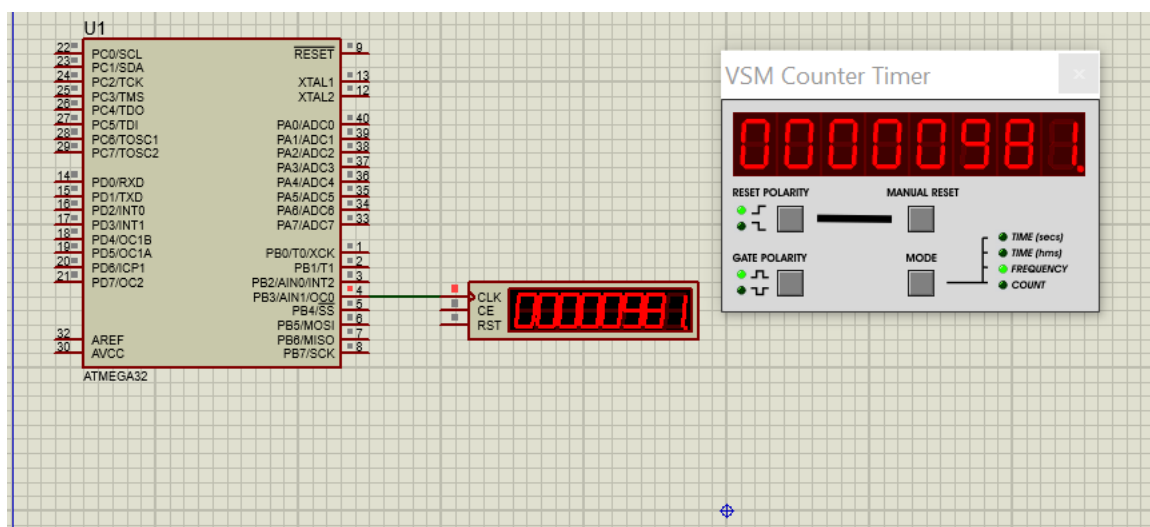
در این حالت n را برابر 8 در نظر می گیریم و clock value را برابر با ۵۰۰ کیلو هرتز قرار می دهیم و خروجی را به صورت non inverting تعریف می کنیم.

در این حالت هم D.C. را ۵۰٪ در نظر می گیریم که در نهایت به OCR0 برابر با 0xE6 می رسم.

متن کد برنامه در حالت Phase Currect به صورت زیر است:

```
#include <mega32.h>
void main(void)
{
    DDRB=0x08;
    PORTB=0;
    // Timer/Counter 0 initialization
    // Clock source: System Clock
    // Clock value: 500.000 kHz
    // Mode: Phase correct PWM top=0xFF
    // OC0 output: Non-Inverted PWM
    // Timer Period: 1.02 ms
    // Output Pulse(s):
    // OC0 Period: 1.02 ms Width: 0.92 ms
    TCCR0=(1<<WGM00) | (1<<COM01) | (0<<COM00) | (0<<WGM01) | (0<<CS02)
    | (1<<CS01) | (0<<CS00);
    TCNT0=0;
    OCR0=0xE6;
    while (1)
        {};
}
```


تصویری از شبیه سازی در این مد در پروتئوس به صورت زیر است:



همانطور که مشاهده می شود فرکانس ۱ کیلوهرتز اندازه گرفته شده است. (با اندکی چشم پوشی)