به نام خدا

گزارش کار آز ریز شماره 9

امیرمحمد پیرحسین لو 9531014

محمد رهبر 9422024

**شرح کلی آزمایش:**

در این آزمایش قصد داریم عملکرد زمانسنج/شمارنده 0 را در حالت CTC و حالت عملکرد عادی بررسی کنیم.

به این منظور، به صورت تناوبی یک LED را به کمک زمانسنج روشن و خاموش می کنیم.

**الف)**

*به دلیل اینکه زمانسنج 4 بار* overflow *شده است و هر بار زمانسنج تا 255 شمرده است و تقسیم کننده هم روی 1024 تنظیم شده است، صورت کسر به شکل بالا درآمده است.*

*به دلیل اینکه کلاک میکرو را روی* 1MH *تنظیم کرده ایم، مخرج کسر به شکل بالا درآمد.*

*تنظیمات زمانسنج در حالت عادی:*

TCNT0 = 0;

OCR0 = 0;

TCCR0 |= (1<<CS02) |(0<<CS01) |(1<<CS00);

TIMSK |= (1<<OCIE0);

تنظیمات زمانسنج در حالت CTC:

TCNT0 = 0;

OCR0 = 127; //frequency = 2Hz

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

TIMSK |= (1<<OCIE0);

|  |
| --- |
| #include <avr\io.h>  #include <avr\interrupt.h> |
| #include <util\delay.h> |
| #include <stdio.h> |
| #include <string.h> |
| #include <stdlib.h> |
| #include <string.h> |
|  |
| #define degree\_sysmbol 0xdf |
| /\* |
| LCD16x2 4 bit ATmega16 interface |
| http://www.electronicwings.com |
| \*/ |
|  |
| #define LCD\_Dir DDRA /\* Define LCD data port direction \*/ |
| #define LCD\_Port PORTA /\* Define LCD data port \*/ |
| #define RS PA1 /\* Define Register Select (data reg./command reg.) signal pin \*/ |
| #define EN PA3 /\* Define Enable signal pin \*/ |
| volatile int currentRow = 1; |
| volatile char dogment[10]; |
| volatile int column = 1; |
| volatile int overflowCounter = 0; |
|  |
| void LCD\_Command( unsigned char cmnd ) |
| { |
| LCD\_Port = (LCD\_Port & 0x0F) | (cmnd & 0xF0); /\* sending upper nibble \*/ |
| LCD\_Port &= ~ (1<<RS); /\* RS=0, command reg. \*/ |
| LCD\_Port |= (1<<EN); /\* Enable pulse \*/ |
| \_delay\_us(1); |
| LCD\_Port &= ~ (1<<EN); |
|  |
| \_delay\_us(200); |
|  |
| LCD\_Port = (LCD\_Port & 0x0F) | (cmnd << 4); /\* sending lower nibble \*/ |
| LCD\_Port |= (1<<EN); |
| \_delay\_us(1); |
| LCD\_Port &= ~ (1<<EN); |
| \_delay\_ms(2); |
| } |
|  |
|  |
| void LCD\_Char( unsigned char data ) |
| { |
| LCD\_Port = (LCD\_Port & 0x0F) | (data & 0xF0); /\* sending upper nibble \*/ |
| LCD\_Port |= (1<<RS); /\* RS=1, data reg. \*/ |
| LCD\_Port|= (1<<EN); |
| \_delay\_us(1); |
| LCD\_Port &= ~ (1<<EN); |
|  |
| \_delay\_us(200); |
|  |
| LCD\_Port = (LCD\_Port & 0x0F) | (data << 4); /\* sending lower nibble \*/ |
| LCD\_Port |= (1<<EN); |
| \_delay\_us(1); |
| LCD\_Port &= ~ (1<<EN); |
| \_delay\_ms(2); |
| } |
|  |
| void LCD\_Init (void) /\* LCD Initialize function \*/ |
| { |
| LCD\_Dir = 0b11111110; /\* Make LCD command port direction as o/p \*/ |
| \_delay\_ms(20); /\* LCD Power ON delay always >15ms \*/ |
|  |
| LCD\_Command(0x33); |
| LCD\_Command(0x32); /\* send for 4 bit initialization of LCD \*/ |
| LCD\_Command(0x28); /\* Use 2 line and initialize 5\*7 matrix in (4-bit mode)\*/ |
| LCD\_Command(0x0c); /\* Display on cursor off\*/ |
| LCD\_Command(0x06); /\* Increment cursor (shift cursor to right)\*/ |
| LCD\_Command(0x01); /\* Clear display screen\*/ |
| \_delay\_ms(2); |
| LCD\_Command (0x80); /\* Cursor 1st row 0th position \*/ |
| } |
|  |
|  |
| void LCD\_String (char \*str) /\* Send string to LCD function \*/ |
| { |
| int i; |
| for(i=0;str[i]!=0;i++)/\* Send each char of string till the NULL \*/ |
| { |
| LCD\_Char (str[i]); |
| } |
| } |
|  |
| void LCD\_String\_xy (char row, char pos, char \*str) /\* Send string to LCD with xy position \*/ |
| { |
| if (row == 0 && pos<16) |
| LCD\_Command((pos & 0x0F)|0x80); /\* Command of first row and required position<16 \*/ |
| else if (row == 1 && pos<16) |
| LCD\_Command((pos & 0x0F)|0xC0); /\* Command of first row and required position<16 \*/ |
| LCD\_String(str); /\* Call LCD string function \*/ |
| } |
|  |
| void LCD\_Clear() |
| { |
| LCD\_Command (0x01); /\* Clear display \*/ |
| \_delay\_ms(2); |
| LCD\_Command (0x80); /\* Cursor 1st row 0th position \*/ |
| } |
|  |
| int main() |
| { |
| TCNT0 = 0; |
| OCR0 = 255; |
| //ctc |
| TCCR0 |=(1<<CS02)|(0<<CS01)|(0<<CS00)|(1<<WGM01); |
|  |
| sei(); |
|  |
| DDRB = 0b11111111;  LCD\_Init(); |
| while(1){} |
| return 0; |
| } |
|  |
|  |
| ISR(TIMER0\_COMP\_vect) |
| { |
| overflowCounter++; |
| if(overflowCounter == 2){ |
| if(PORTB & (1 << PB3)){ |
| PORTB &= ~(1 << PB3); |
| }else{ |
| PORTB |= (1 << PB3); |
| } |
| overflowCounter = 0; |
| } |
| } |

**چالش ها:**

طبق شکل دستور کار، پایه RW که در LCD است باید به PA2 وصل شود که در این صورت کار LCD کار نمی کند.  
با جستجو مشخص شد که این پایه باید به GND وصل شود.

