**آزمایش 1**

**شمارش وقفه**

**1. در تنظیمات اولیه، LCD را چگونه config کردید؟ (2 نمره)**

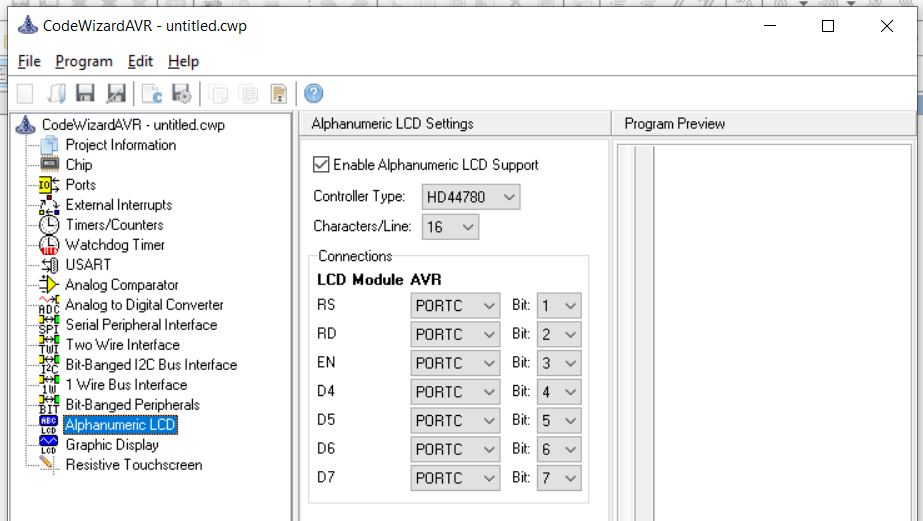
در پنجره CodeWizard از منوی سمت چپ گزینه Alphanumeric LCD را انتخاب می‌کنیم.

گزینه Enable Alphanumeric LCD Support را فعال می‌کنیم.

گزینه Characters را بر روی 16 قرار می‌دهیم؛ چون LCD که از قبل انتخاب کردیم، 16 ستون دارد.

همه مقادیر پورت را روی PORTC می‌گذاریم و از شماره 1 تا 7 برای بیت‌ها انتخاب می‌کنیم.

مطابق تصویر زیر:



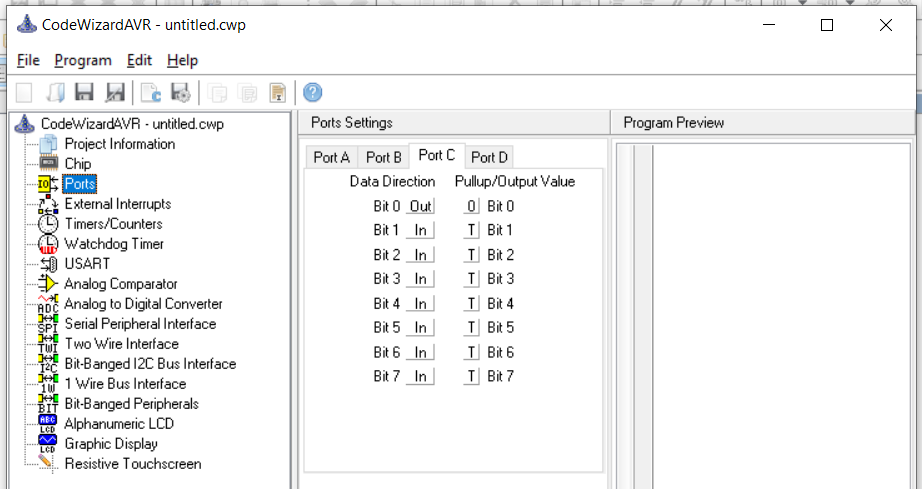
**2. در تنظیمات اولیه، LED ها را چگونه config کردید؟ (2 نمره)**

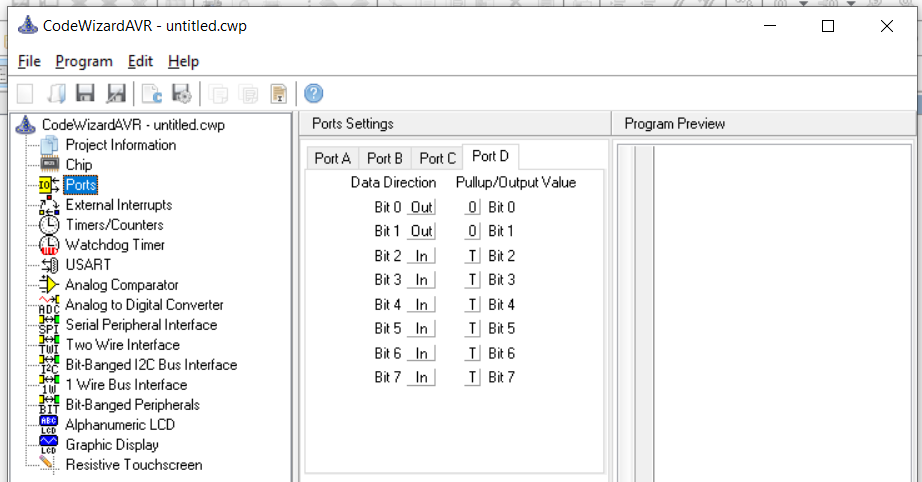
در پنجره CodeWizard از منوی سمت چپ گزینه Ports را انتخاب می‌کنیم.

بر روی بیت صفر و یک پورت D کلیک می‌کنیم تا حالت آن به خروجی تغییر کند. مقدار پیشفرض را نیز روی صفر قرار می‌دهیم. برای بیت صفرم پورت C نیز همین مراحل را انجام می‌دهیم.

چون سه ماژول LED ما به این پورت‌ها متصل شده‌اند، آنها را به حالت خروجی تغییر دادیم.

مطابق تصویر زیر:





**3. هر بار که وقفه خارجی صفر یا یک رخ می‌دهد، برنامه شما چه عملیاتی انجام می‌دهد؟ (4 نمره)**

برنامه از حلقه while(1) که در تابع main درحال اجرا است، به توابع ext\_int0\_isr یا ext\_int1\_isr منتقل می‌شود و پس از اجرای محتوای آنها دوباره ادامه حلقه while را از سر می‌گیرد.

در تابع وقفه صفر این برنامه، تابع setNum فراخوانی می‌شود و آرگومان num+2 به آن پاس داده می‌شود. این تابع همان طور که از نامش پیداست، مقدار num را آپدیت می‌کند، مقدار جدید را روی LCD چاپ می‌کند و در انتها تابع checkNum را فراخوانی می‌کند تا با بررسی مقدار جدید عدد، وضعیت LED ها تعیین شود.

در تابع وقفه یک نیز موارد مشابه رخ می‌دهد. تنها تفاوت، آرگومان num-1 می‌باشد.

**4. کد برنامه (2 نمره)**

*#include* <stdlib.h>

*// Declare your global variables here*

int num = 0;

char numstr[10] = "";

*// Functions*

void checkNum()

{

*// Blue: D1, Green: D0, Yellow: C0*

*if* (num < 0) {

        PORTD.0 = 0;

        PORTD.1 = 0;

        PORTC.0 = 1;

    } *else* *if* (num % 2 == 0) {

        PORTD.0 = 0;

        PORTD.1 = 1;

        PORTC.0 = 0;

    } *else* {

        PORTD.0 = 1;

        PORTD.1 = 0;

        PORTC.0 = 0;

    }

}

void setNum(int x)

{

    num = x;

    itoa(num, numstr);

    lcd\_clear();

    lcd\_puts(numstr);

    checkNum();

}

interrupt [EXT\_INT0] void ext\_int0\_isr(void)

{

    setNum(num + 2);

}

interrupt [EXT\_INT1] void ext\_int1\_isr(void)

{

    setNum(num - 1);

}

*// Ini*

setNum(0);

*while* (1)

      {

*// Place your code here*

      }

}