راه اندازی و اندازه گیری دما با سنسور دمای DS18B20 به صورت تک و چند سنسور با AVR

تاریخ انتشار۱۵ تیر, ۱۴۰۰ توسط آرش فتاحی

با عرض سلام خدمت همراهان سایت مایکروالکام. در این مطلب قصد داریم ابتدا یک سنسور دمای DS18B20 را به میکروکنترلر Atmega32 متصل کرده و سپس با کامپایلر کدویژن برنامه مربوط به آن را جهت قرائت دما و نمایش آن بر روی LCD کاراکتری انجام دهیم. در مرحله بعد به جای 1 عدد سنسور دما، 8 عدد از آن را به میکرو متصل کرده و برنامه‌ای خواهیم نوشت که با نگه داشتن یک کلید که به میکرو متصل است، به ترتیب دمای سنسور 1 تا 8 بر روی

LCD کاراکتری نمایش داده شود. پس با من تا انتهای مطلب همراه باشید. همچنین میتونید سایر مطالب من رو از این قسمت مطالعه کنید.

معرفی سنسور دما DS18B20

سنسور DS18B20، یک سنسور دمای دیجیتال دقیق و ارزان قیمت بوده که می‌توان از آن برای سنجش دمای محیط‌های گوناگون استفاده نمود. این سنسور دقتی 12 بیتی و حدودِ 0.0625 درجه سانتی گراد داشته و براساس دیتاشیت ارائه شده توسط شرکت سازنده، می‌تواند دمای بین 55- تا 125+ درجه سانتی‌گراد را اندازه‌گیری کند.

سنسور دما *DS18B20*

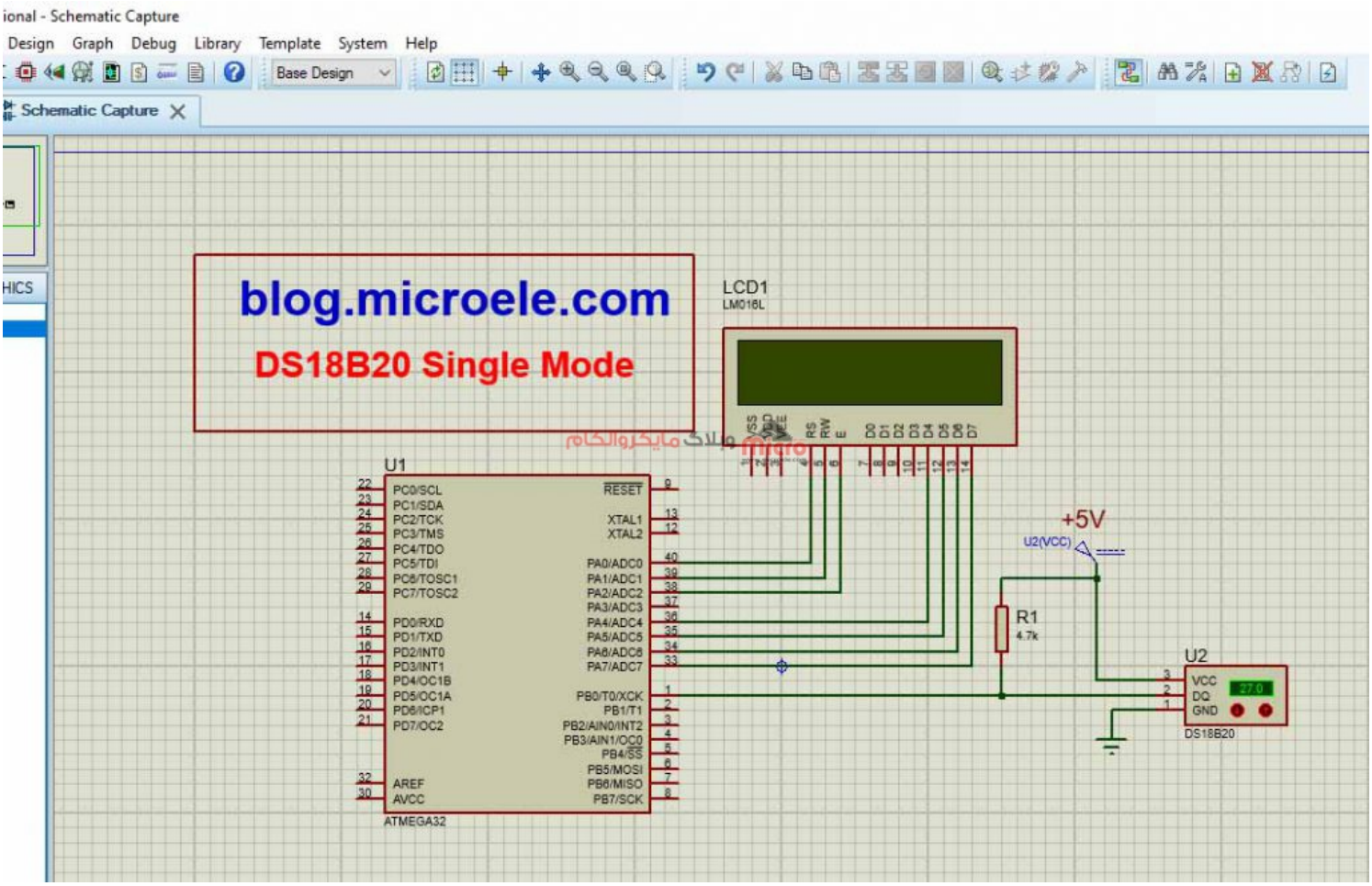
برای اتصال DS18B20 به میکروکنترلر، از پروتکل یک سیمه (One Wire) استفاده می‌شود. از آن‌جایی که هر کدام از سنسورهای DS18B20 آدرس اختصاصی خود را دارند، می‌توان تا 8 عدد سنسور را از طریق یک سیم کنترل کرده و دمای هر کدام از این سنسورها را به صورت مجزا توسط میکروکنترلر قرائت کرد.

وسایل مورد نیاز

- میکروکنترلر Atmega32
- سنسور دمای DS18B20
- مقاومت 4.7 کیلواهم
- LCD کاراکتری
- پتانسیومتر 10 کیلواهم برای تنظیم کنتراست LCD
- برد برد
- سیم برد بردی

بخش اول: راه اندازی سنسور DS18B20 به صورت تک سنسور (Single)

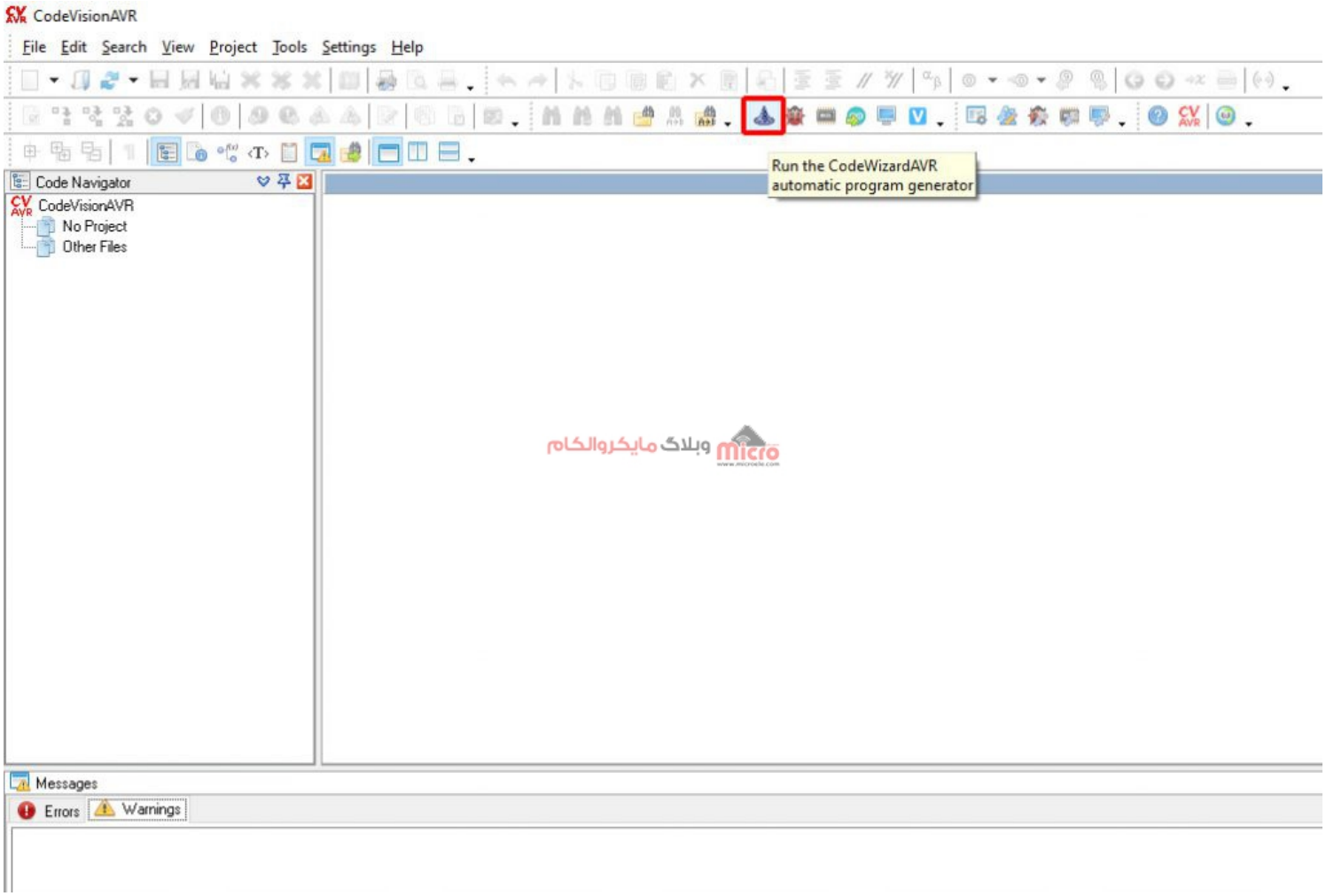
ابتدا مدار مورد نظر را مانند تصویر زیر در نرم افزار پروتئوس پیاده‌سازی می‌کنیم. برای اتصال سنسور دما، حتما باید پایه‌ای که قرار است برای داده‌های سنسور اختصاص دهیم، **با یک مقاومت 4.7 کیلواهم Pull-Up گردد**.



مدار بسته شده در پروتئوس برای شبیه سازی *DS18B20*

تظیمات CodeWizard در نرم افزار CodeVision

نرم افزار کدویژن را باز کرده و Wizard را اجرا می‌کنیم.

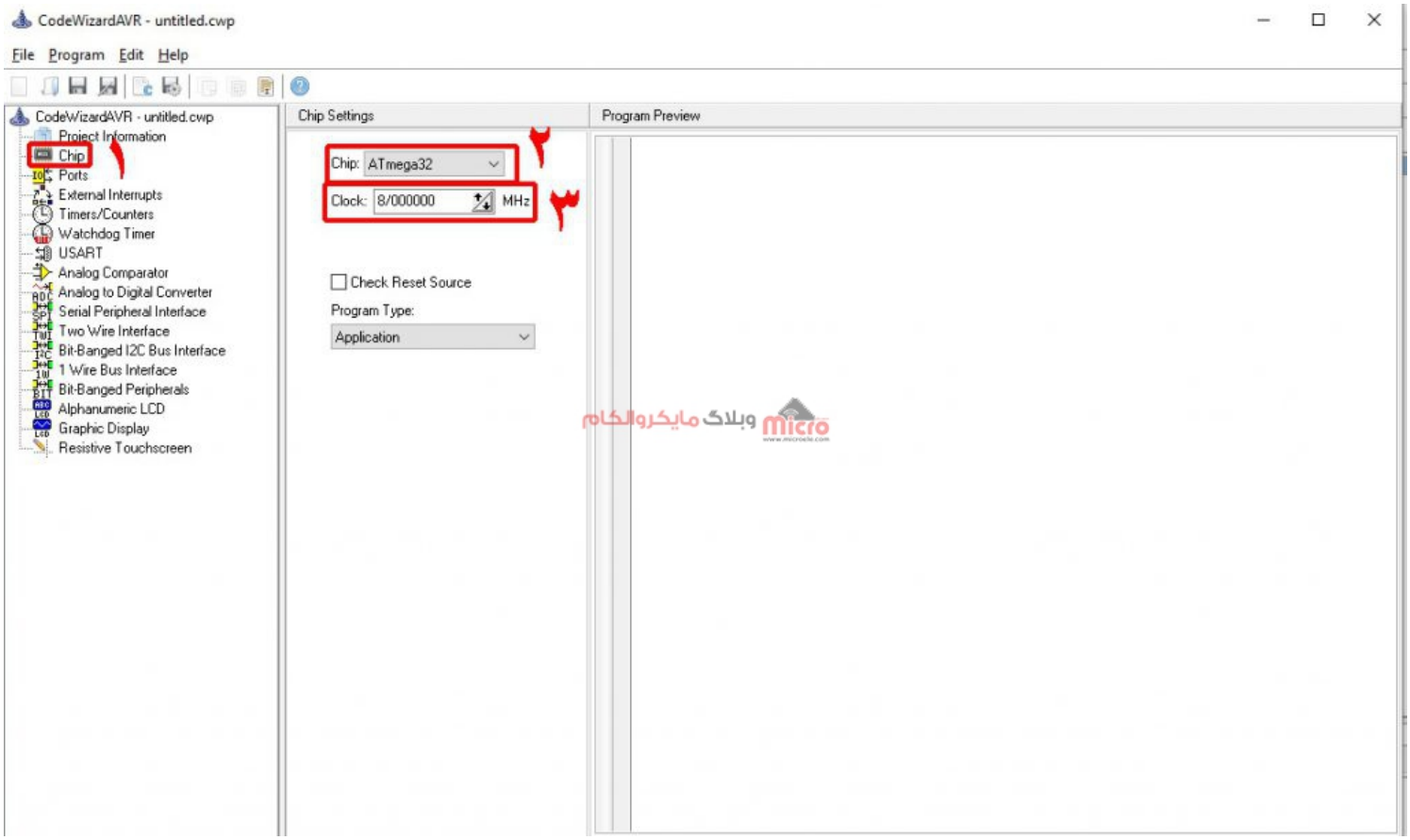


CodeWizard

در پنجره باز شده برای انتخاب Target، گزینه اول را انتخاب کرده و بر روی OK کلیک کنید تا وارد محیط Wizard شوید.

CodeWizard Target

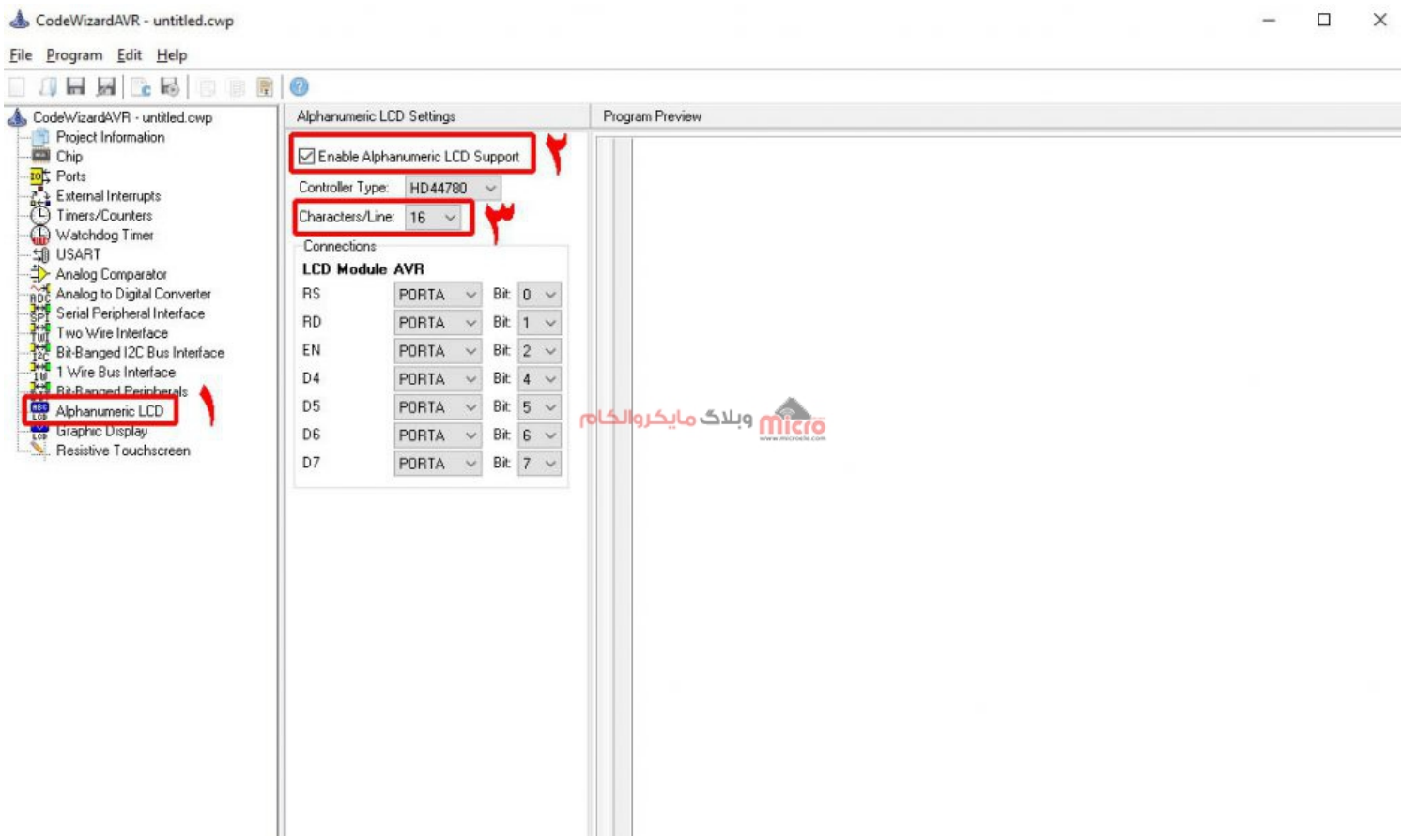
در Wizard، ابتدا چیپ مورد نظر خود، که در اینجا برای ما Atmega32 می‌باشد را انتخاب کرده و سپس در قسمت Clock، فرکانس کاری میکرو را تعیین می‌کنیم. در این پروژه می‌خواهیم میکرو با فرکانس داخلی 8 مگاهرتز کار کند، بنابراین Clock را بر روی 8/000000 تنظیم می‌کنیم. شما می‌توانید فرکانس‌های بالاتر و یا پایین تر را انتخاب کنید.



تنظیمات میکرو در *CodeWizard*

تنظیمات LCD کاراکتری

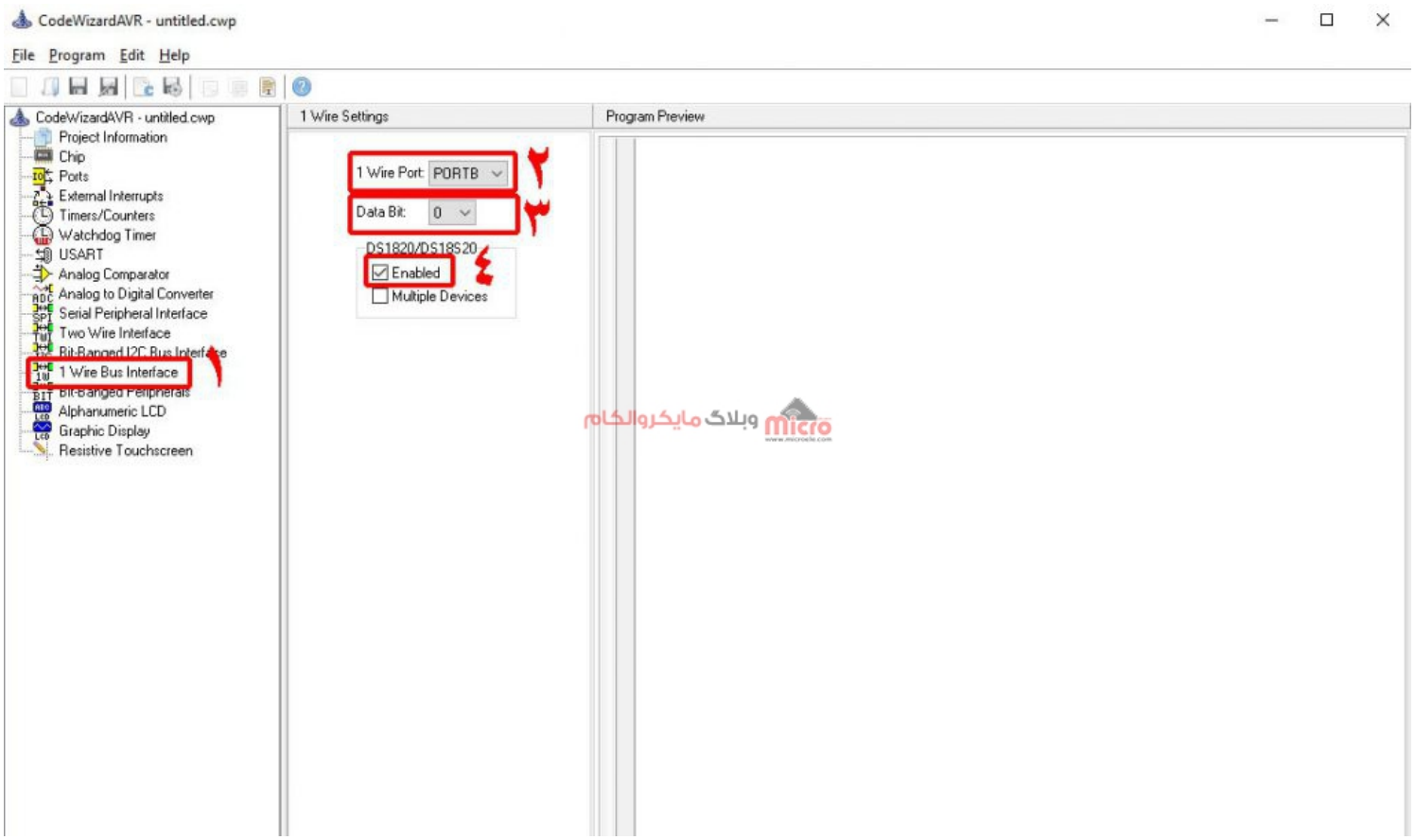
برای راه اندازی LCD، از بخش Alphanumeric LCD گزینه Enable Alphanumeric LCD Support را فعال کنید. از آن‌جایی که LCD استفاده شده، 16 کاراکتری می‌باشد، Characters/Line را روی 16 قرار می‌دهیم.



تنظیمات *LCD* در *CodeWizard*

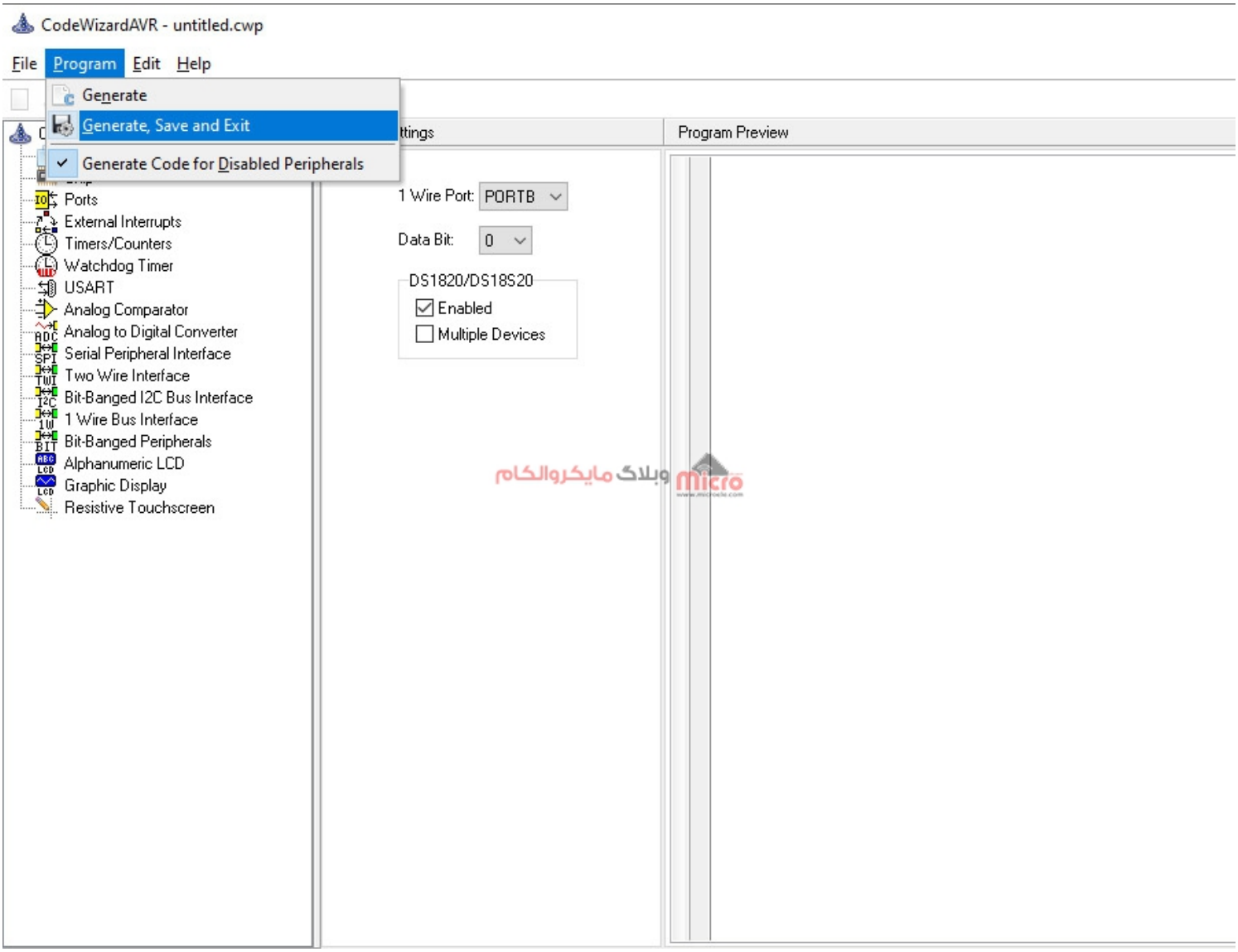
تنظیمات One Wire

بخش آخر تنظیمات مربوط به راه اندازی DS18B20 است. برای این کار به بخش 1 Wire Bus Interface رفته و از قسمت 1 Wire Port، پورتی که سنسور به آن متصل است را انتخاب کنید. در اینجا ما سنسور را به PORTB متصل کرده‌ایم. در قسمت Data Bit، شماره پایه‌ای که سنسور به میکرو متصل شده است را انتخاب کنید، که برای ما پایه صفرمِ پورت B می‌باشد. در نهایت از بخش DS1820/DS18B20 گزینه Enable را فعال کنید.



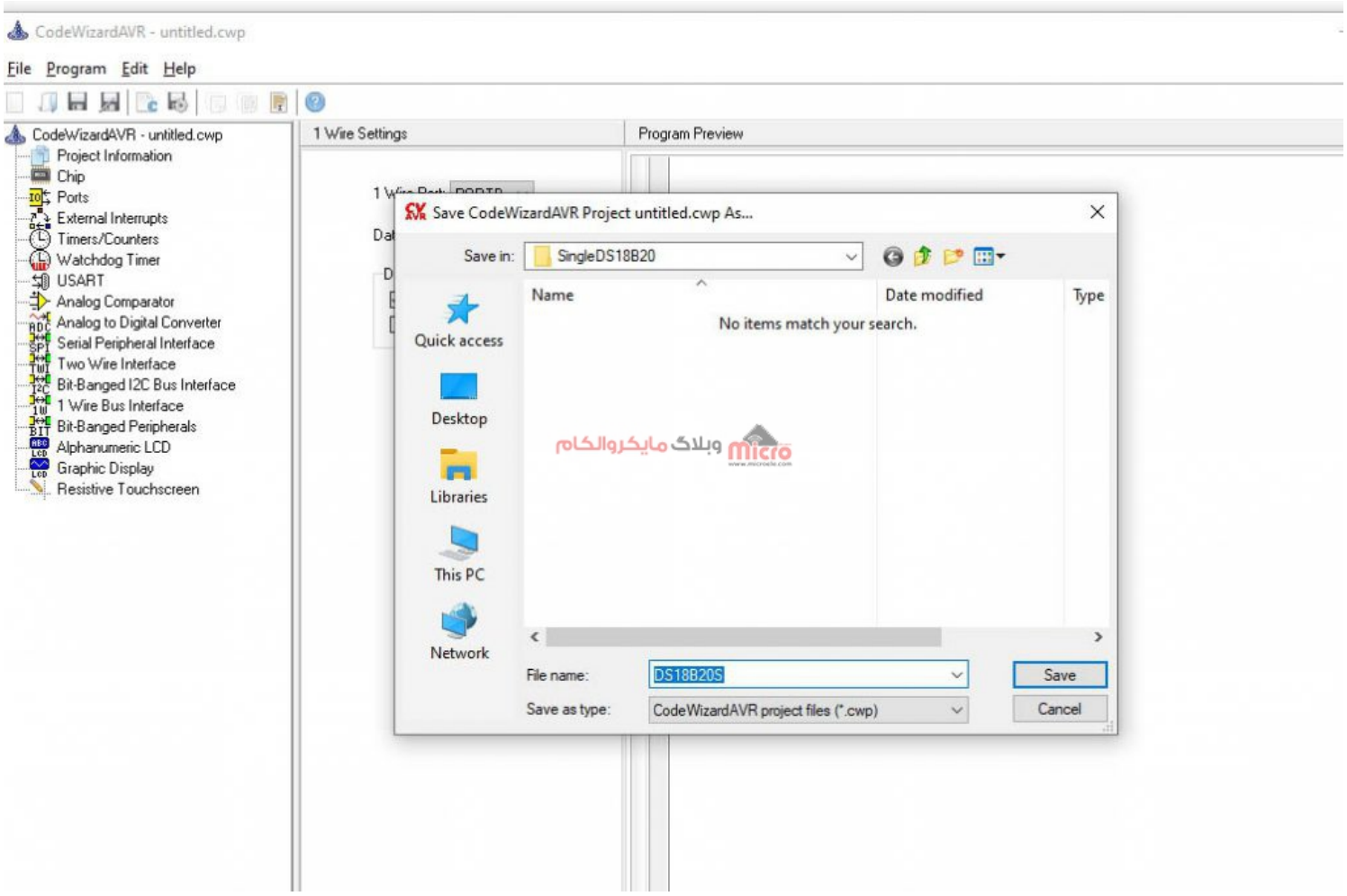
تنظیمات One Wire و DS18B20 در CodeWizard

حال نوبت به ذخیره سازی و Generate کردن پروژه می‌رسد. برای این کار از منوی Program در کدویزارد گزینه Generate, Save and Exit را انتخاب کنید.



تولید کد در *CodeWizard* در کدویژن

در نهایت برای ذخیره سازی فایل‌ها نام انتخاب کرده و آن‌ها را Save کنید. در این‌جا، فایل c. را به نام main ذخیره کردیم.

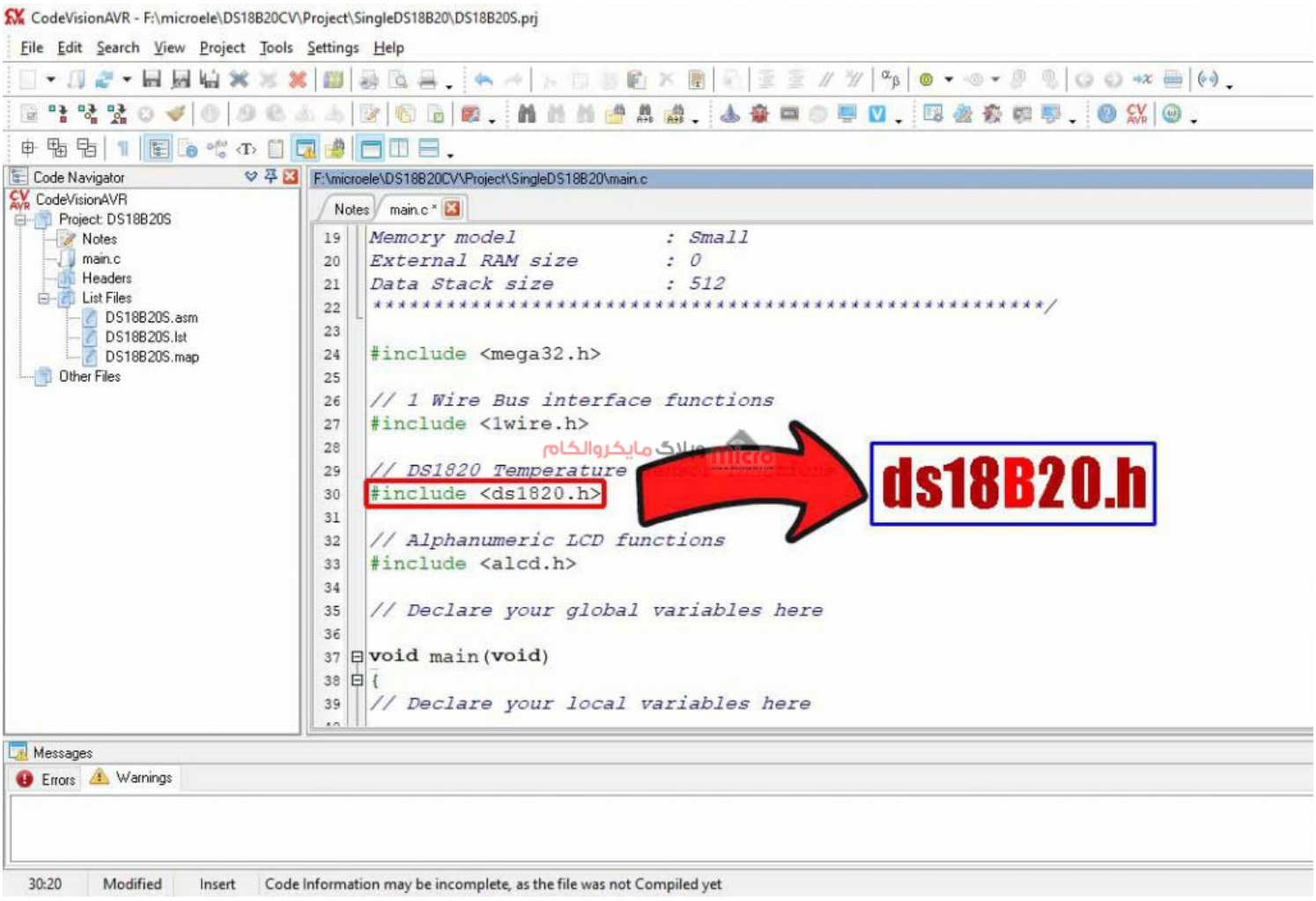


ذخیره سازی پروژه در *CodeWizard*

شروع کد نویسی برای حالت تک سنسور در کدویژن

بعد از باز شدن فایل c. که در این‌جا برای ما main.c می‌باشد، ابتدا باید نحوه فراخوانی کتاب‌خانه سنسورِ دما را اصلاح کنیم. از آن‌جایی که ما از سنسور DS18B20 استفاده می‌کنیم، بنابراین ds1820.h را باید به ds18B20.h تغییر دهیم. حال برنامه را Build کرده تا فایل ds18B20.h توسط کامپایلر به پروژه اضافه گردد.

```
#include <ds18B20.h>
```

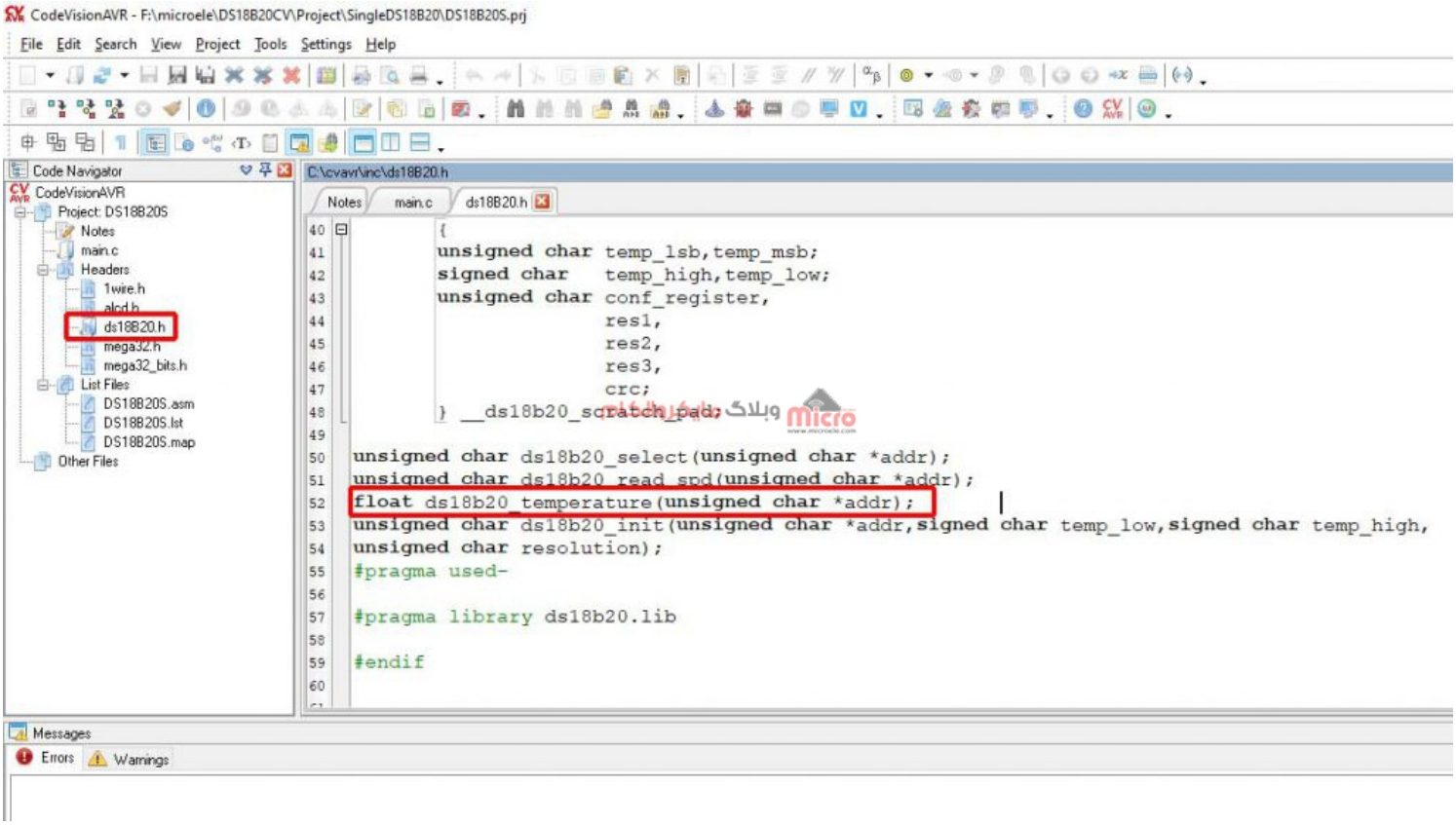


جایگزینی کتاب‌خانه *DS18B20*

بعد از Build کردن برنامه، در قسمت Code Navigator در سمت چپ، ds18B20.h توسط کامپایلر لود شده و نمایان می‌شود.

اضافه کردن کتاب‌خانه‌ها و متغیرها

با کلیک بر روی آن می‌توانید توابع این کتاب‌خانه را مشاهده کنید. برای خواندن مقادیر دما، تابع ds18b20_temperature که خروجی آن به صورت Float است را استفاده خواهیم کرد.



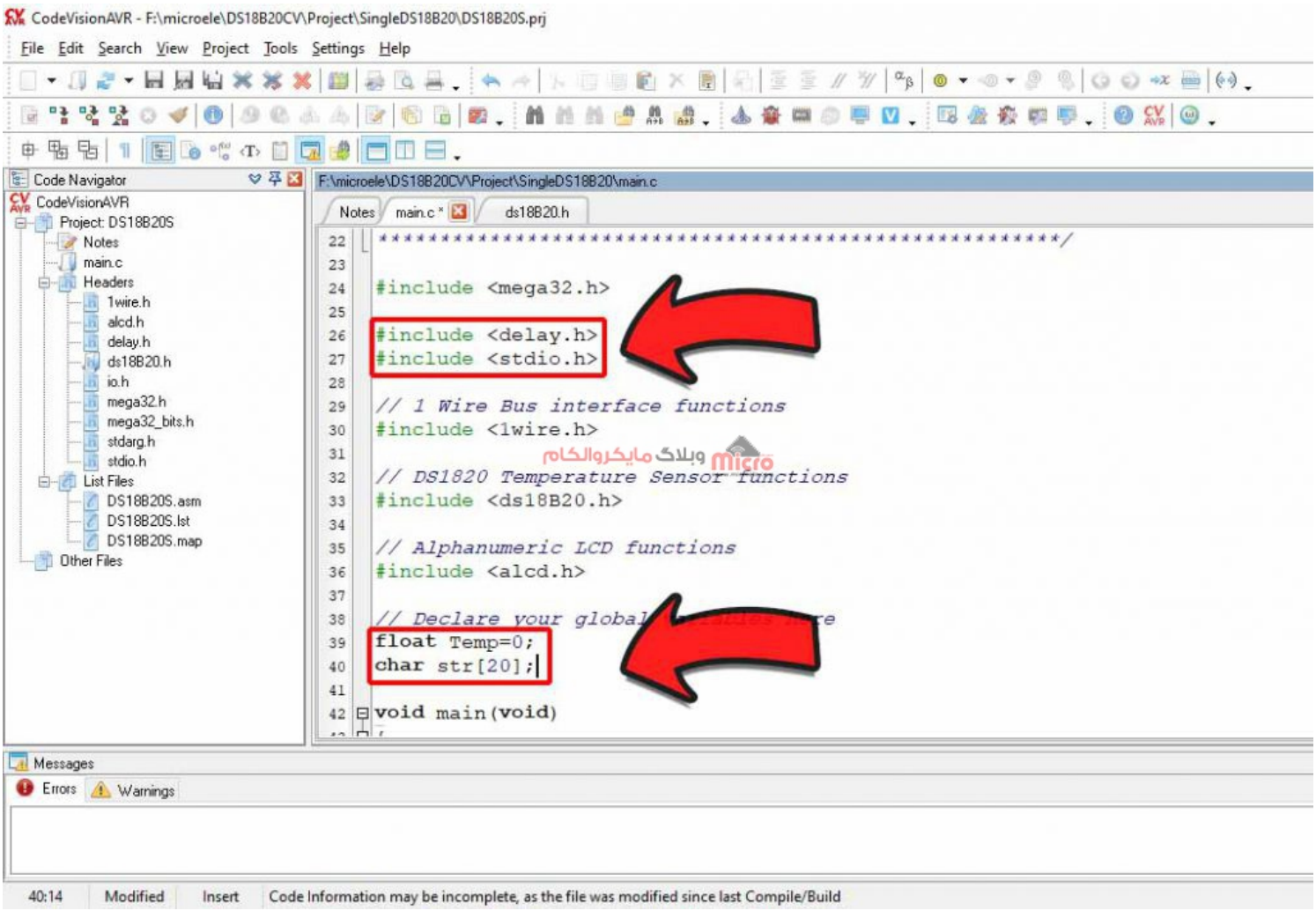
کتاب‌خانه *DS18B20*

```
ds18b20_temperature();
```

مانند تصویر زیر، در فایل main.c کتاب‌خانه‌های delay.h و stdio.h را اضافه کنید. از کتاب‌خانه stdio جهت استفاده از تابع sprintf برای تبدیل مقادیر float به کاراکتر جهت چاپ مقادیر دما بر روی LCD استفاده خواهیم کرد.

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

همچنین متغیرهای Temp از نوع Float جهت انتقال مقادیر خوانده شده از سنسور و آرایه رشته‌ای str را جهت تبدیل مقادیر عددی به رشته برای نمایش بر روی LCD کاراکتری در برنامه تعریف می‌کنیم.



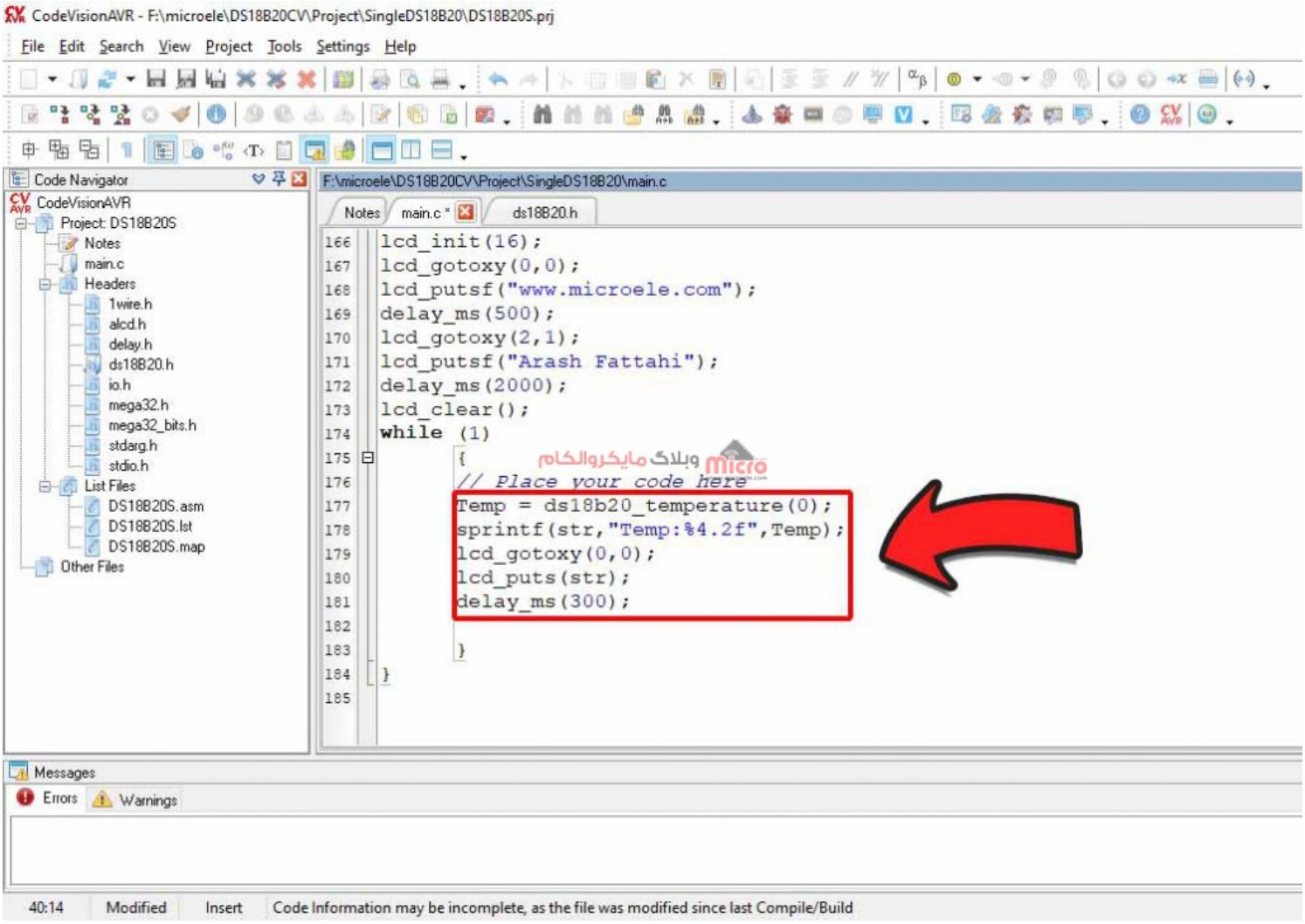
افزودن کتاب‌خانه‌ها در فایل.c

```
float Temp=0;
char str[20];
```

تکمیل برنامه در حلقه While در کدویژن

حال برای نوشتن ادامه برنامه به حلقه While می رویم. تابع ds18b20_temprature دارای یک ورودیِ آدرس است. از آن‌جایی که هدف ما در این بخش راه اندازی یک تک سنسور است، ورودی این تابع را مقدار 0 قرار می‌دهیم. در آخر

برای انتقال خروجی این تابع، آن را برابر با متغیر Temp که قبل‌تر آن را تعریف کرده بودیم قرار می‌دهیم.



تکمیلِ برنامه نوشته شده در حلقه *while*

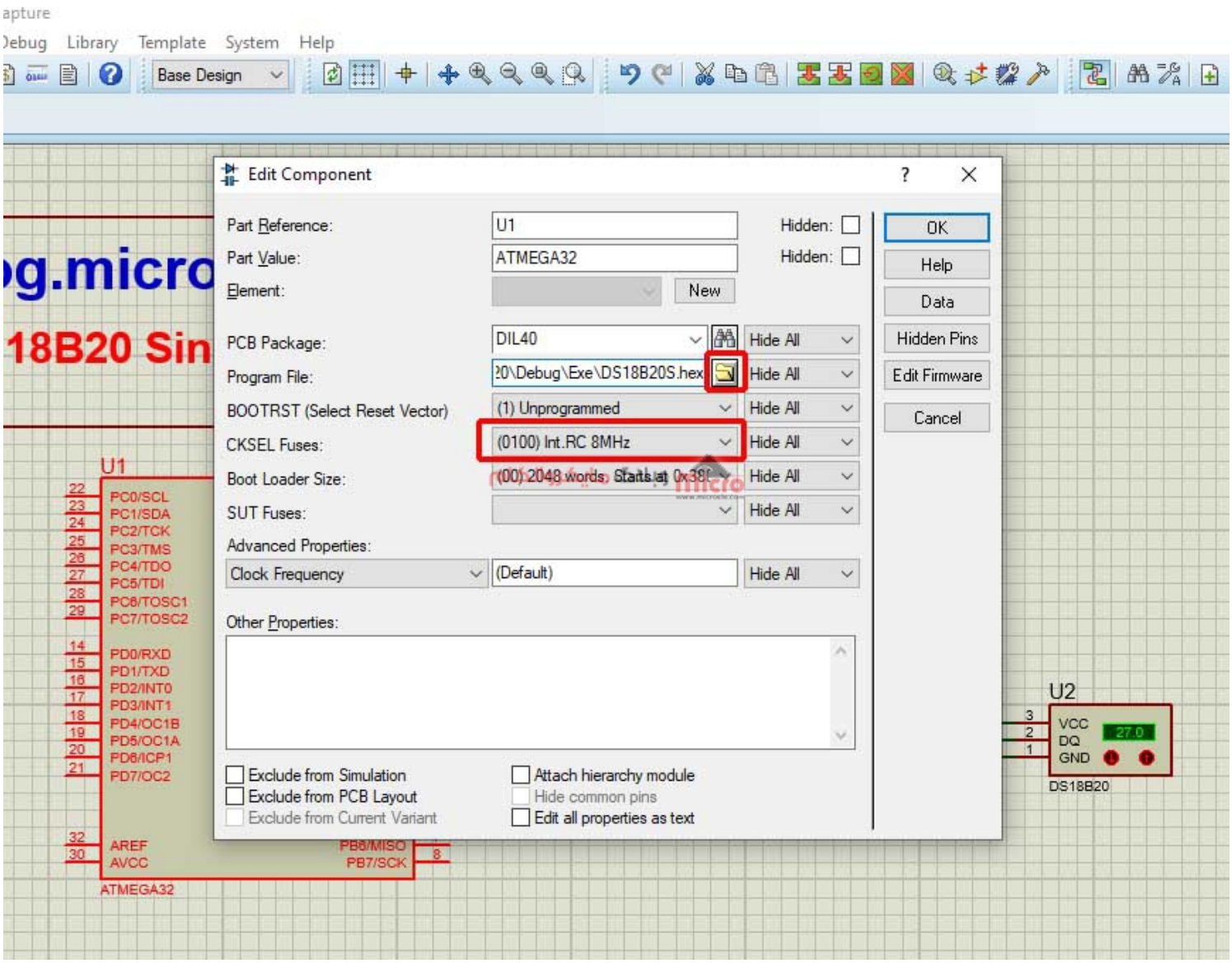
```
Temp = ds18b20_temperature(0);
sprintf(str,"Temp:%4.2f",Temp);
lcd_gotoxy(0,0);
lcd_puts(str);
```

```
delay_ms(300);
```

از تابع sprintf جهت تبدیل مقدار Float به رشته استفاده کرده و مقدار متغیر Temp را به صورت دو رقمی با دو رقم اعشار از طریق المانِ 4.2f به متغیر رشته‌ای str تبدیل و به آن منتقل می‌کنیم. درصورتی که می‌خواهید مقادیر سه رقمی نیز قابل چاپ بر روی LCD باشند، کافیست به جای عبارت 4.2f از 5.2f در تابع sprintf استفاده کنید. در نهایت با دستور lcd_gotoxy(0,0) به ابتدای سطر اول LCD رفته و از طریق دستور lcd_puts(str) مقادیر دما را بر روی LCD چاپ می‌کنیم.

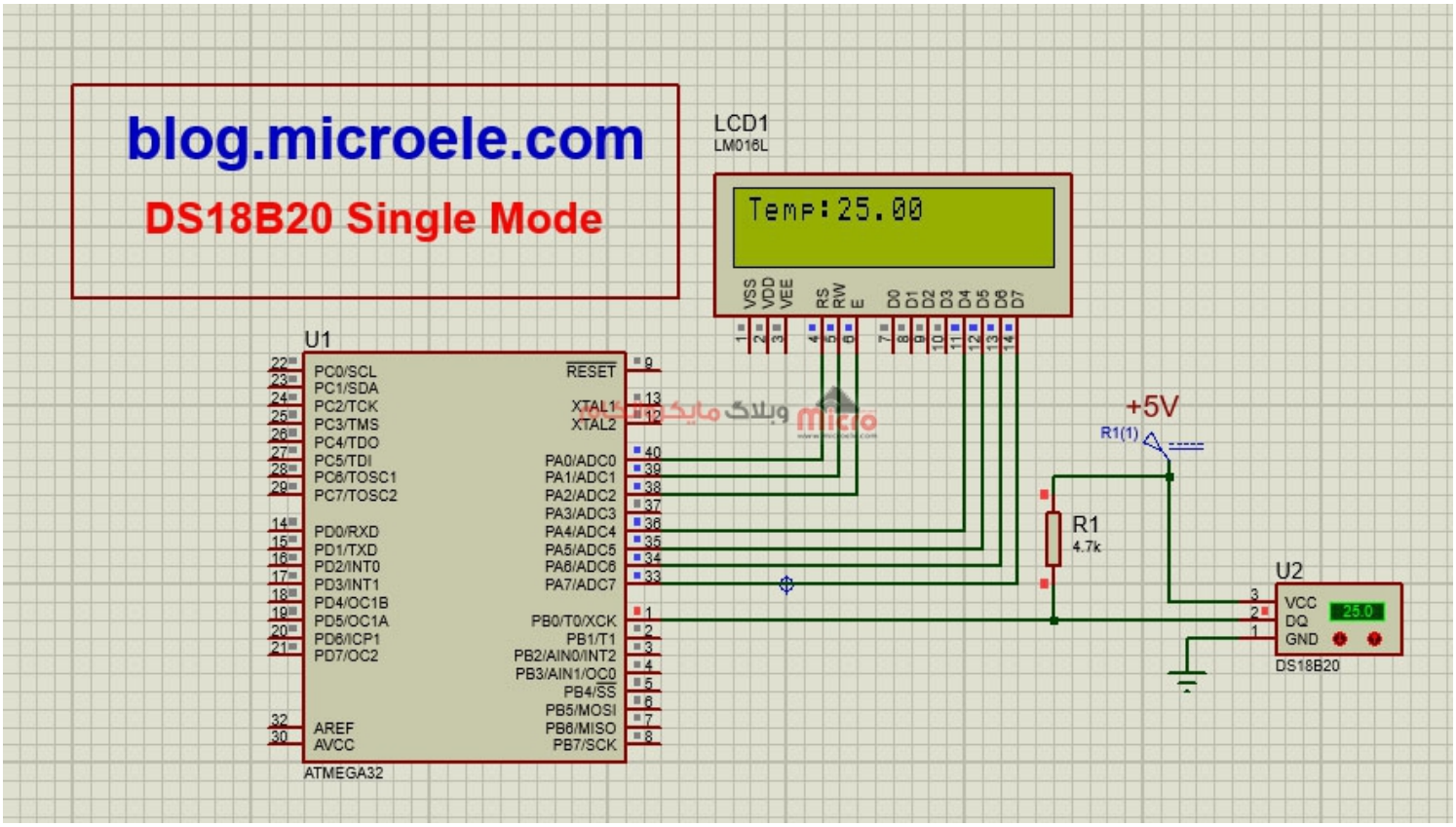
شبیه سازی حالت تک سنسور DS18B20

به محیط پروتئوس رفته و با دو بار کلیک بر روی میکرو، برنامه را به آن لود کرده و همچنین فرکانس داخلی میکرو را بر روی 8Mhz داخلی تنظیم می‌کنیم.



لود کردن برنامه به میکرو در محیط پروتئوس

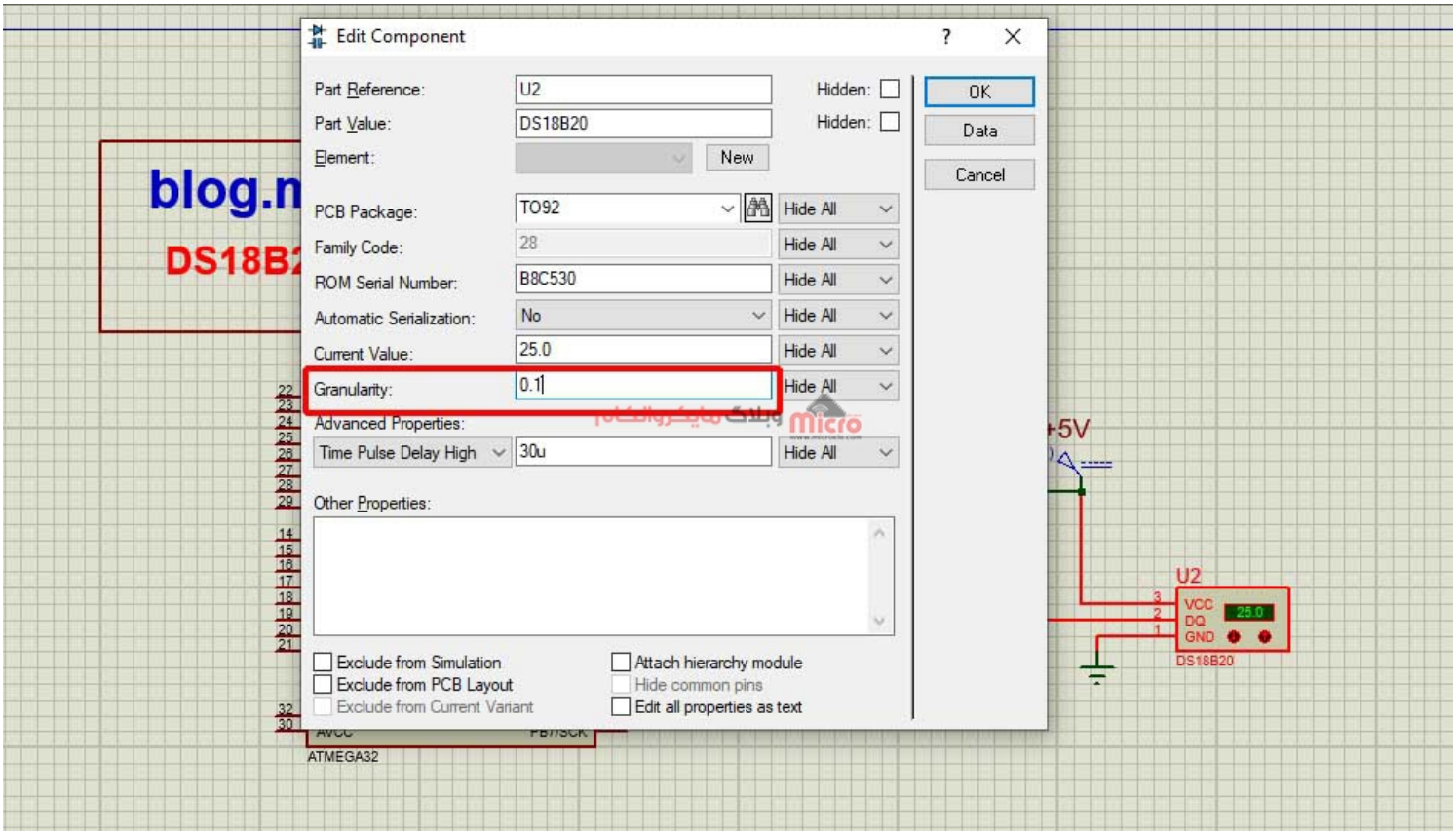
پروتئوس را Run کرده و شبیه سازی را اجرا می‌کنیم. همچنین برای تغییر دما می‌توانید از دو کلیدِ قرمز رنگِ فلش مانند بر روی سنسور استفاده کنید.



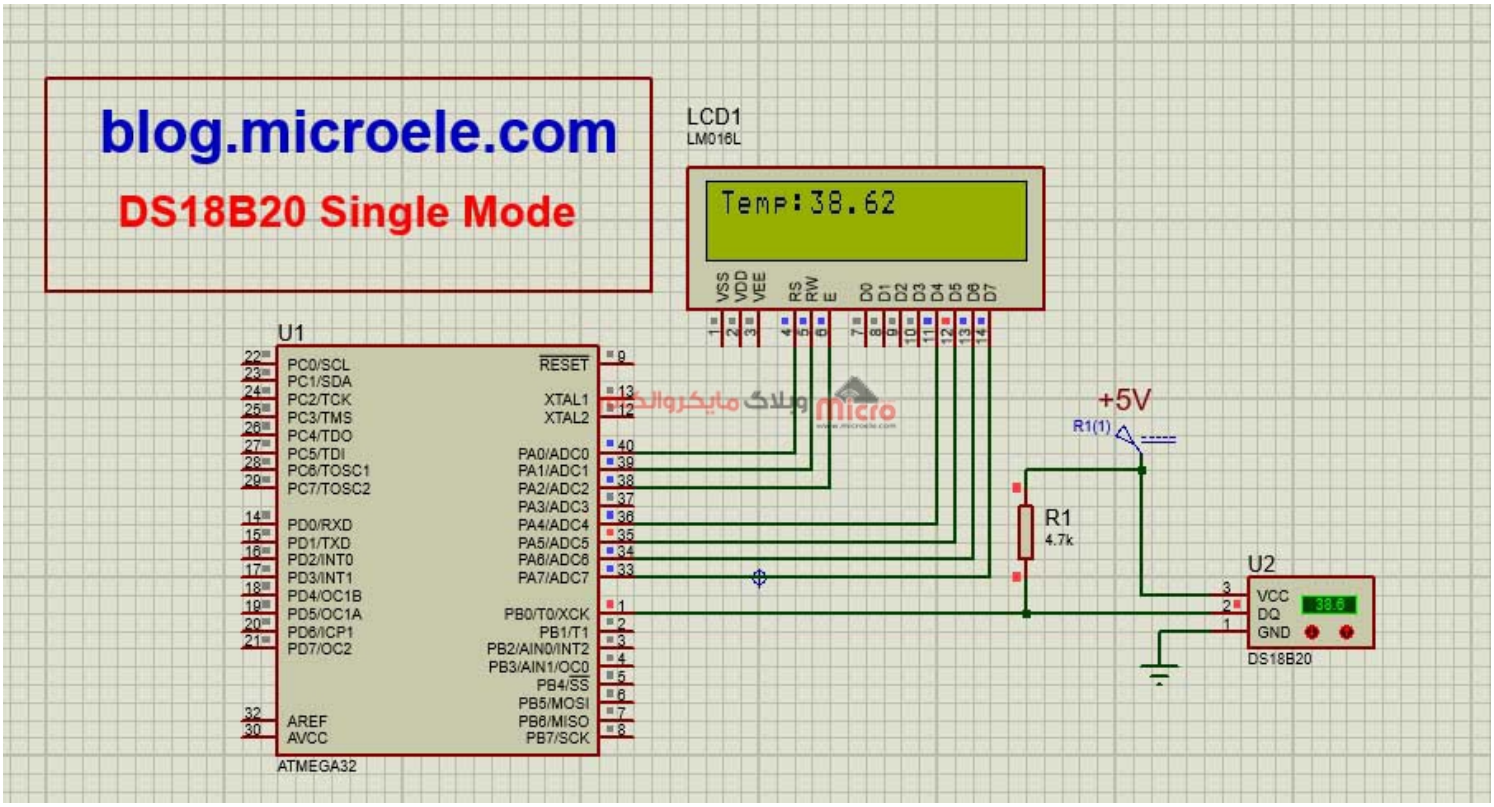
شبیه سازی *Single Mode* در پروتئوس

تظنیم Granularity در پروتئوس

برای بالاتر بردن دقت سنسورِ DS18B20 در پروتئوس و تولید مقادیر اعشاری، کافیست بر روی سنسور DS18B20 کلیک راست، و سپس کلیک چپ کرده تا پنجره تنظیمات سنسور باز شود. با قرار دادن مقادیر 0.1 یا 0.01 در قسمتِ Granularity می‌توان دقت اعمالِ دما بر روی سنسور را افزایش داد.



تغییر *Granularity* سنسور *DS18B20* در پروتئوس

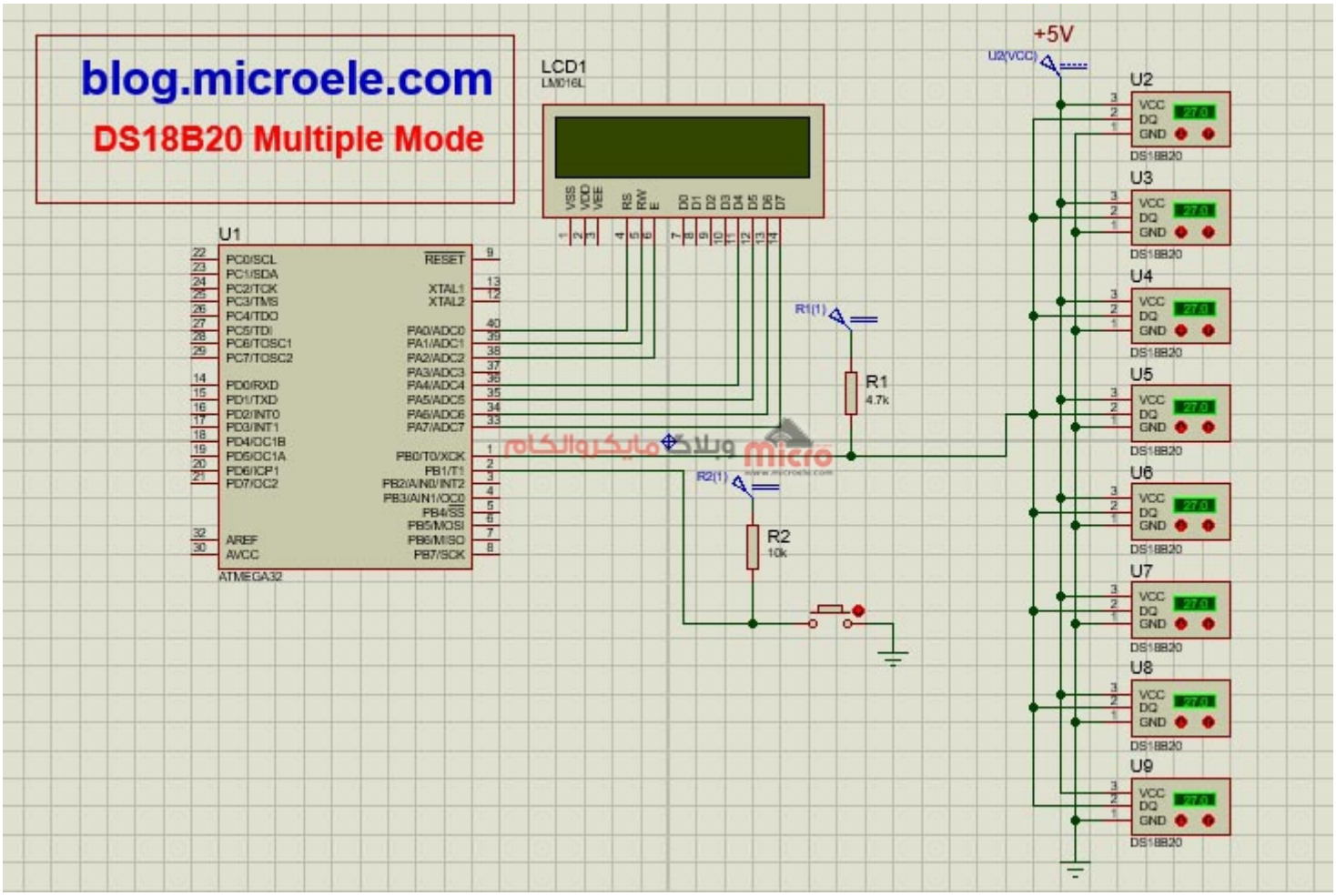


شبیه سازی *DS18B20* در پروتئوس

بخش دوم: راه اندازی سنسور DS18B20 به صورت چند سنسور(Multiple)

در این بخش به راه اندازی تعداد بیشتری سنسور دمای DS18B20 به صورت هم زمان بر روی یک پین میکرو خواهیم پرداخت.

مطابق تصویر زیر مدار را در نرم افزار پروتئوس می‌بندیم:



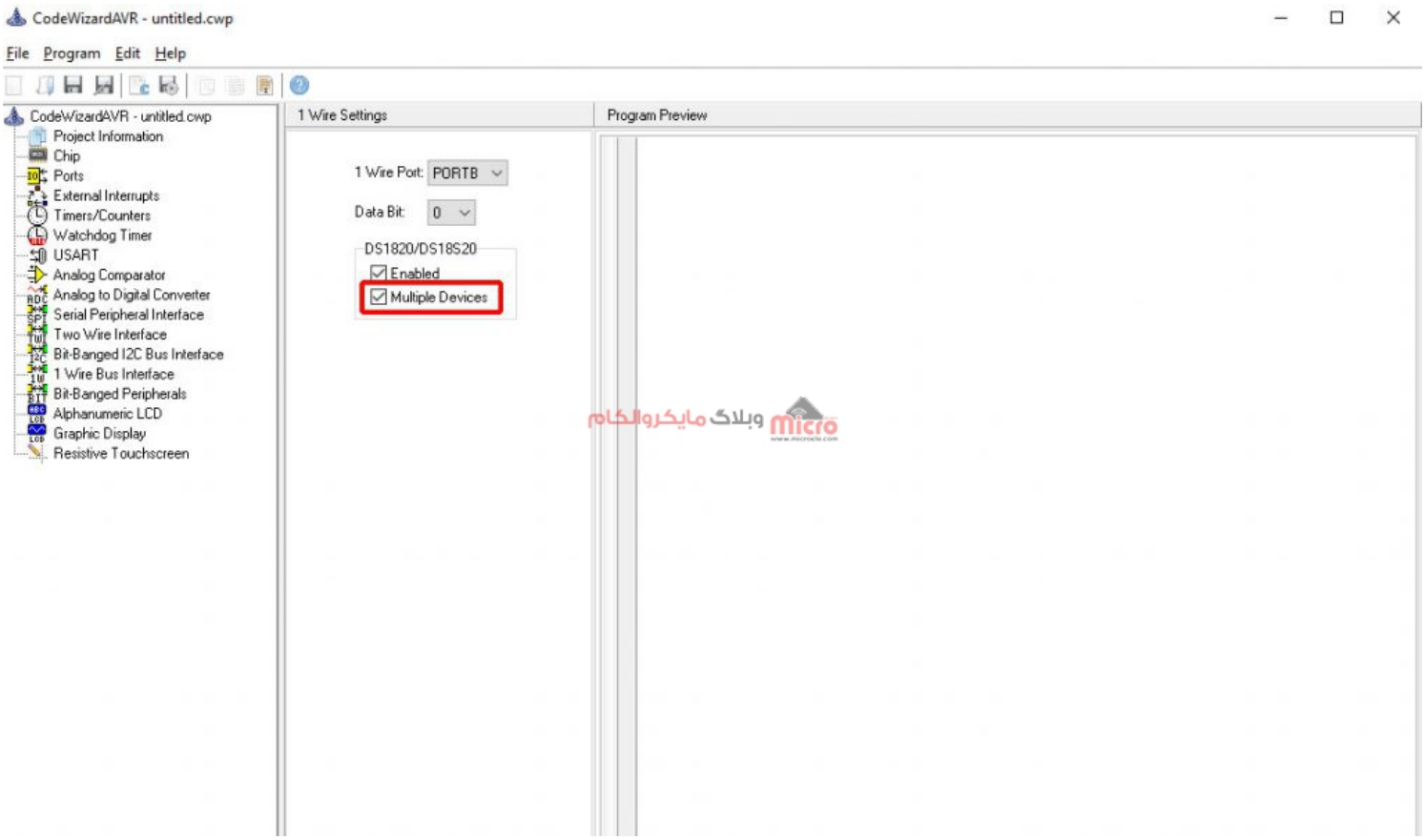
بستن مدار *DS18B20* به صورت چند سنسور

در این مدار، یک کلید به پایه PB1 میکرو اضافه می‌کینم که با نگه داشتن آن، نمایش دمای مرتبط با هر سنسور بر روی LCD به صورت چرخشی عوض شود.

اعمال تنظیمات Code Wizard برای حالت چند سنسور

مطابق بخش قبل، پیکربندی‌های مربوط به انتخاب میکرو و LCD را در کدویزارد نرم افزار کدویژن انجام داده و به بخش مربوط به تنظیمات One Wire می‌رویم.

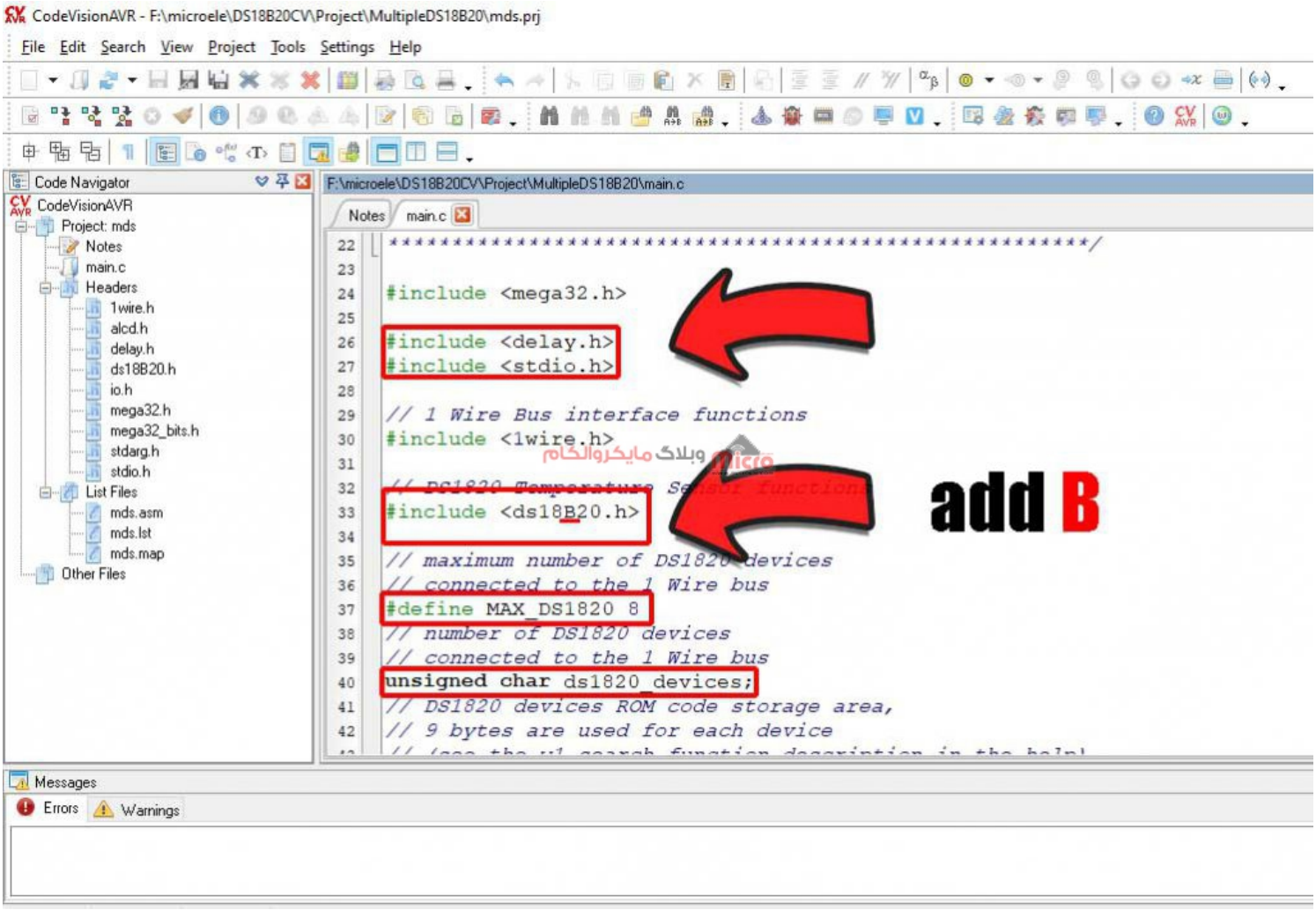
تنظیمات را مانند دفعه قبل انجام داده با این تفاوت که این بار گزینه Multiple Devices را نیز فعال می‌کنیم.در آخر پروژه را Generate کرده و Save کنید.



فعال کردن گزینه *Multiple Devices* در *CodeWizard*

شروع کد نویسی برای حالت چند سنسور DS18B20 در کدویژن

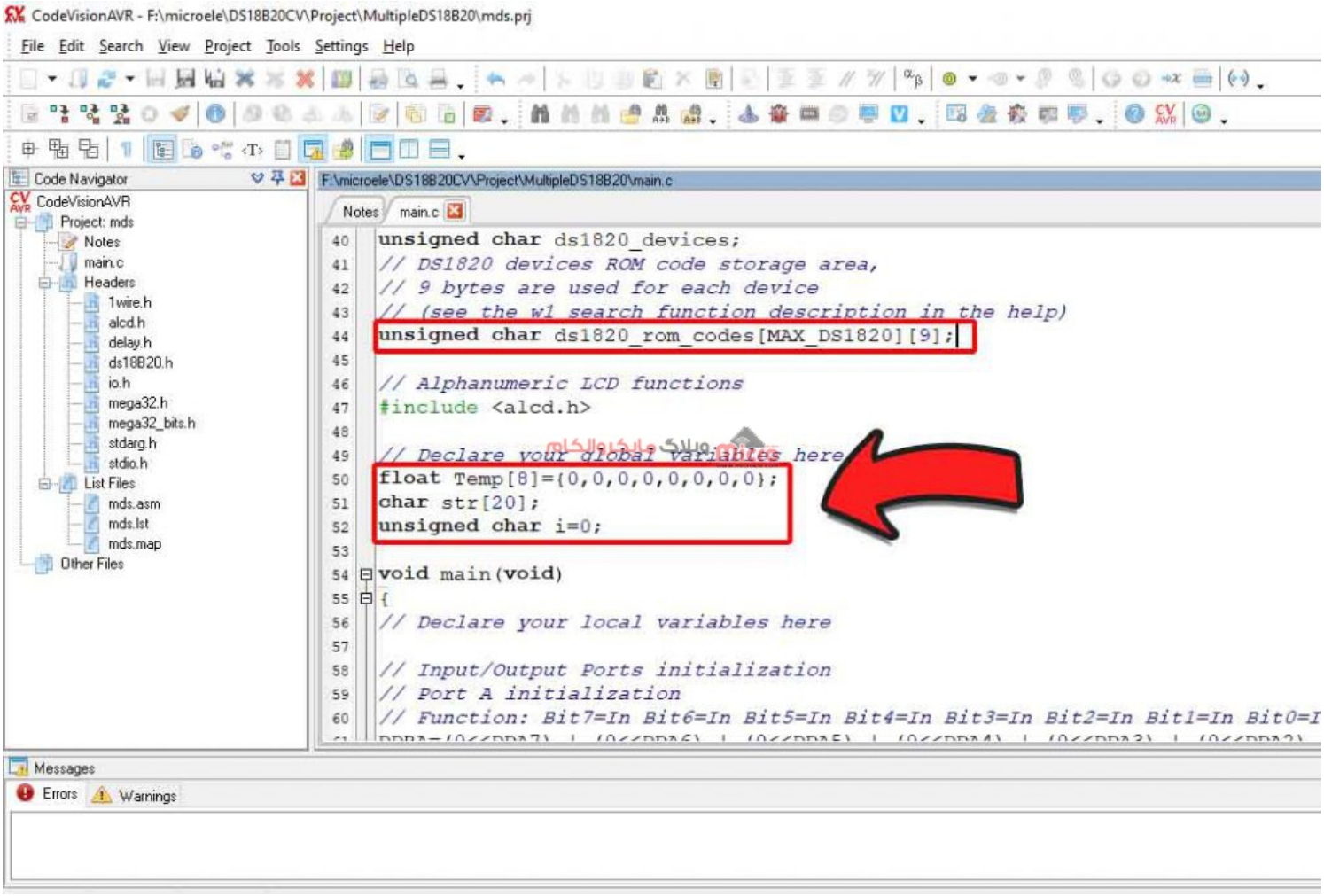
مانند تصویر زیر مجدداً در فایل main.c، کتاب‌خانه‌ها را اضافه کنید. همچنین حرف B را در کتاب‌خانه ds1820 اضافه کنید. به صورت پیش فرض یک تعریف با عنوان MAX_DS1820 8 توسط کدویژن ایجاد شده است که ماکزیمم تعداد سنسورهای DS18B20، که می‌توانیم به میکرو متصل کنیم را نشان می‌دهد. در حال حاضر این تعریف برای 8 عدد سنسور تنظیم شده است. اگر تعداد سنسورهایی که به میکرو متصل کرده‌اید کمتر است، می‌توانید عدد متناظر با تعداد سنسورهای مدنظر خود را به جای 8 قرار دهید. در قسمت بعدیِ کد، یک متغیر به نام ds1820_devices به صورت پیش فرض تعریف شده است که آدرس سنسورهای DS18B20 در آن قرار می‌گیرد.



اضافه کردن کتاب‌خانه به فایل *main.c*

آرایه ds1820_rom_codes

کمی پایین‌تر، یک آرایه به نام ds1820_rom_codes[MAX_DS1820][9] به صورت پیش فرض توسط کدویژن در کد قرار داده شده است که تمامی آدرس‌های مربوط به سنسورها، در این آرایه ذخیره می‌گردد. در تصویر زیر، یک آرایه به نام Temp با 8 خانه جهت ذخیره سازی مقادیر دمای سنسورها تعریف کردیم. همچنین یک متغیر i به عنوان شمارنده جهت گزینش سنسورها هنگام فشردن کلید تعریف شده است.



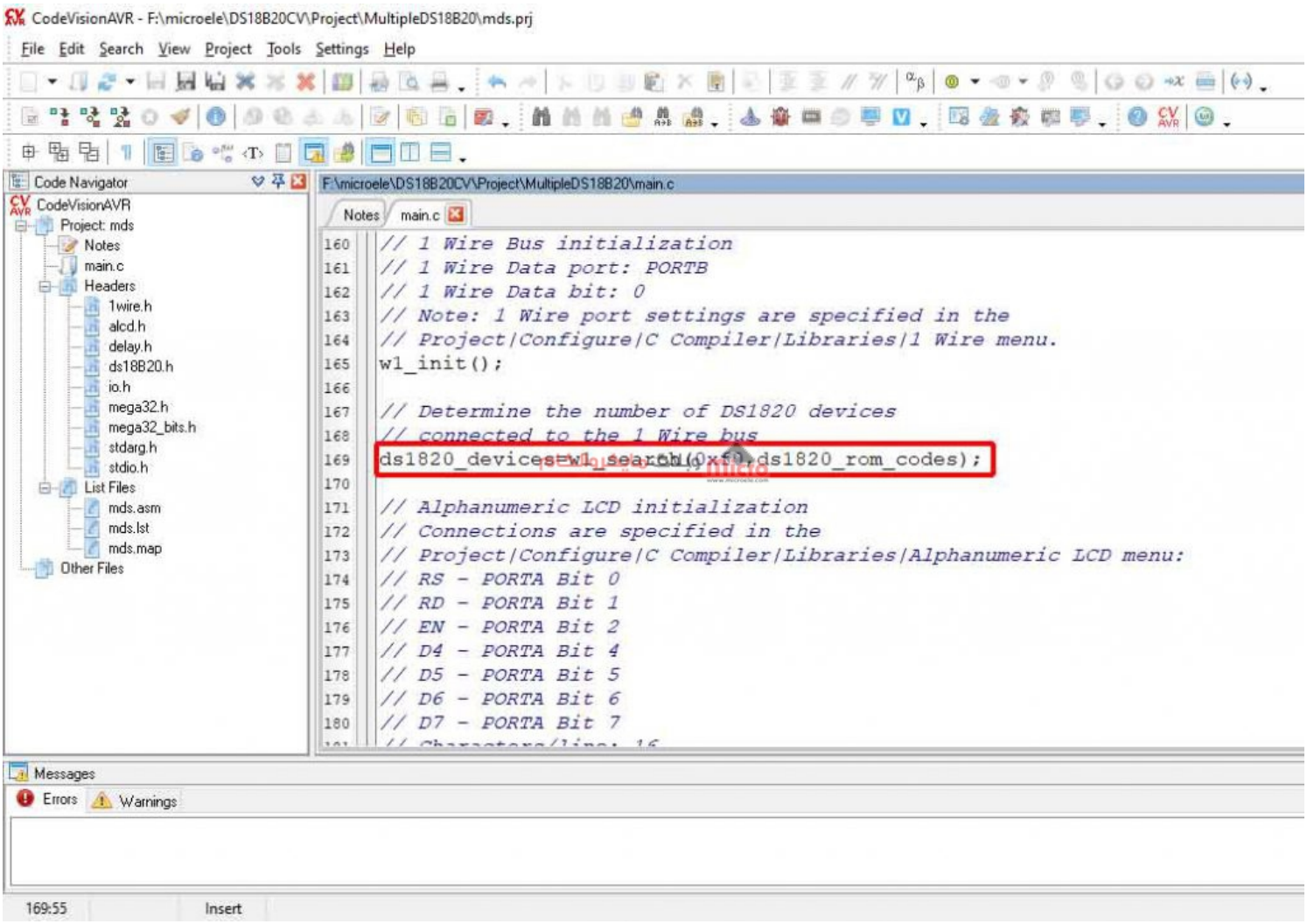
تعریف متغیرها در فایل *main*

```
float Temp[8]={0,0,0,0,0,0,0,0};
char str[20];
unsigned char i=0;
```

نحوه جست و جوی آدرس‌های سنسورهای DS18B20 توسط کدویژن

از طریق دستور (ds1820_devices=w1_search(0xf0,ds1820_rom_codes که به صورت پیش فرض توسط کدویژن در فایل

main.c ایجاد شده است، عملیات جست و جوی آدرس‌های سنسورهای DS18B20 متصل شده به میکرو انجام شده و در نهایت این آدرس‌ها در آرایه ds1820_rom_codes ذخیره می‌گردند.

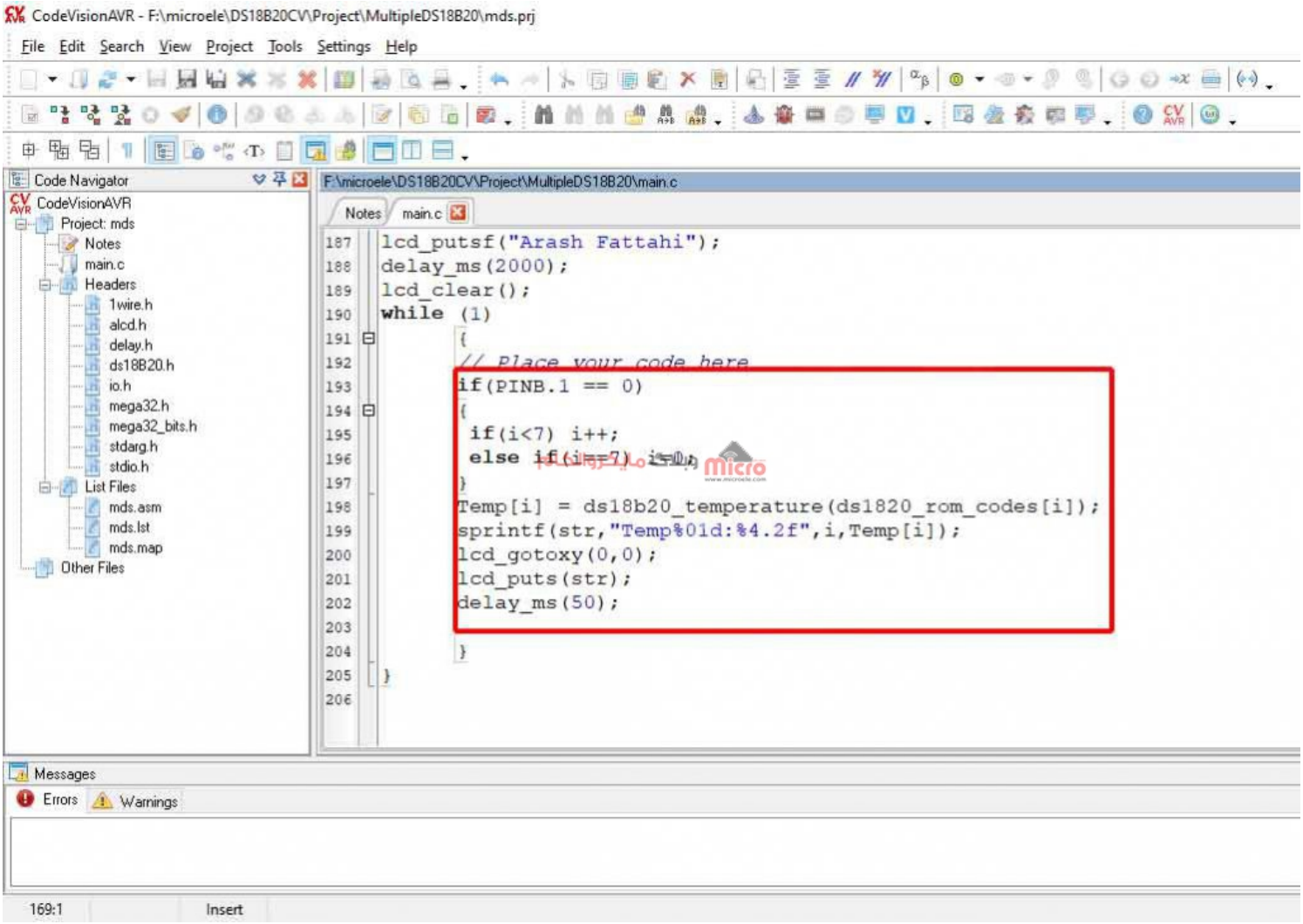


جست و جوی آدرس‌ها در *DS18B20*

قرائت دما با سنسور دما DS18B20 در حالت Multiple

مجدداً با استفاده از تابع ds18B20_temprature می‌توانیم مقادیر سنسورها را بخوانیم. با این تفاوت که برای ورودی این

تابع به جای عدد 0، آرایه ds18B20_rom_codes، که حاویِ آدرس‌های سنسورها می‌باشد را قرار می‌دهیم.



تکمیل برنامه در حلقه اصلی *While*

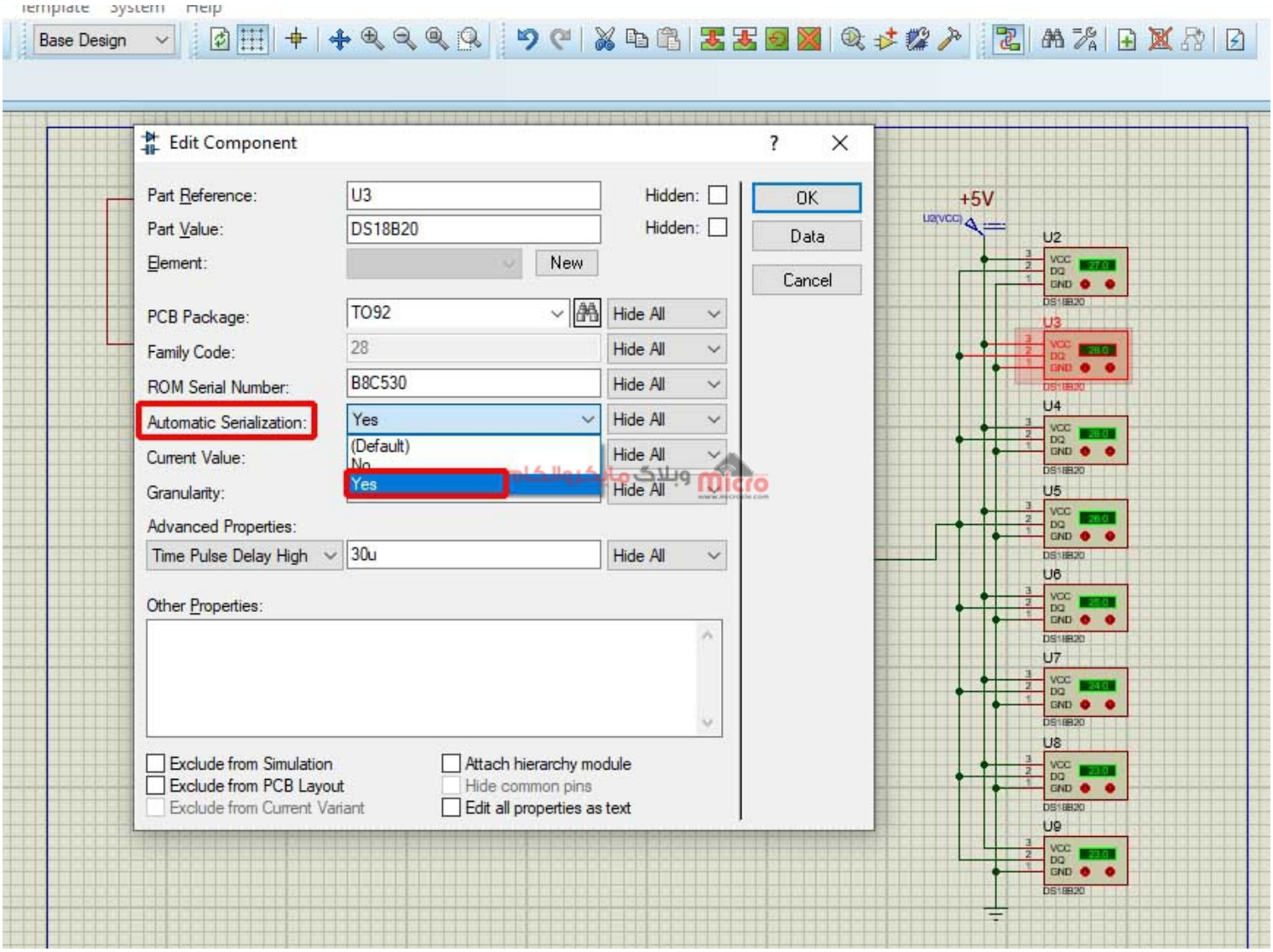
```
if(PINB.1 == 0)
      {
       if(i<7) i++;
       else if(i==7) i=0;
      }
```

```
Temp[i] = ds18b20_temperature(ds1820_rom_codes[i]);
sprintf(str,"Temp%01d:%4.2f",i,Temp[i]);
lcd_gotoxy(0,0);
lcd_puts(str);
delay_ms(50);
```

برنامه را Build کرده و در محیط پروتئوس برنامه را بر روی میکرو لود می‌کنیم.

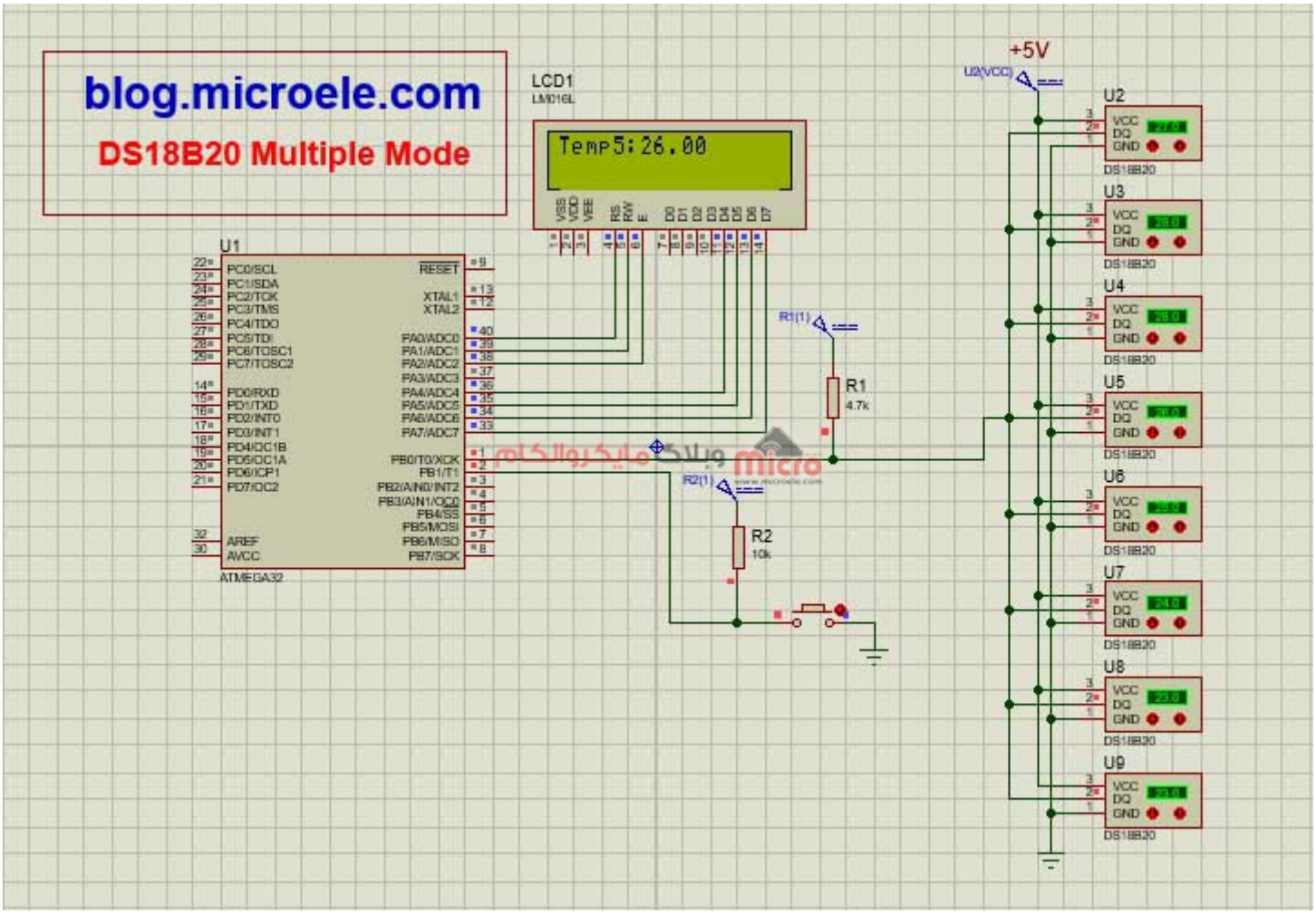
شبیه سازی حالت Multiple سنسور دما DS18B20 در پروتئوس

در محیط پروتئوس بر روی سنسور های دما یک بار کلیک راست و سپس کلیک چپ کرده تا پنجره تنظیمات سنسور باز شود. منوی Automatic Serialization را باز کرده و آن را بر روی Yes قرار دهید. این تنظیم باید برای تمامیِ سنسورها اعمال شود.



فعال سازیِ *Automatic Serialization* سنسور دما در پروتئوس

حال با نگه داشتن کلید در پروتئوس، دمای سنسورها به ترتیب بر روی LCD نمایش داده خواهد شد.



شبیه سازی *Multiple Mode* در پروتئوس

برای دریافت فایل‌های پروژه و شبیه سازی سنسور دمای DS18B20، از طریق این لینک اقدام نمایید.

توجه شود که برای اجرای فایل‌های پروتئوس، از ورژن 8.11 به بالا استفاده شود.

جمع بندی

در این مطلب ویژگی های سنسور دمای DS18B20 که یکی از پرکاربردترین سنسورهای دما در پروژه‌های الکترونیکی و صنعتی می‌باشد معرفی و نحوه راه اندازی سنسور دمای DS18B20، با میکروکنترلر AVR و کامپایلر کدویژن، به صورت

تک سنسور و چند سنسور و نحوه شبیه سازیِ آن در محیط پروتئوس شرح داده شد.

امیدوارم از این آموزش کمال بهره را برده باشید. در صورتی که هرگونه نظر یا سوال داشتید درباره این آموزش لطفا اون رو در انتهای همین صفحه در قسمت دیدگاه ها قرار بدید. در کوتاه ترین زمان ممکن به اون ها پاسخ خواهم داد.
اگر این مطلب براتون مفید بود، اون رو حتما به اشتراک بگذارید. همینطور میتونید این آموزش را پس از اجرای عملی توی اینستاگرام با هشتگ microelecom# به اشتراک بگذارید و پیج مایکروالکام (microelecom@) رو هم منشن کنید.