



راه اندازی سنسور دمای MAX30205 با میکروکنترلر AVR و کدویژن

راه اندازی سنسور دما MAX30205

(آموزش قدم به قدم مطابق با دیتاشیت)



CV
AVR



<https://blog.microele.com>

تاریخ انتشار ۱۱ مهر، ۱۴۰۰ توسط آرش فتاحی

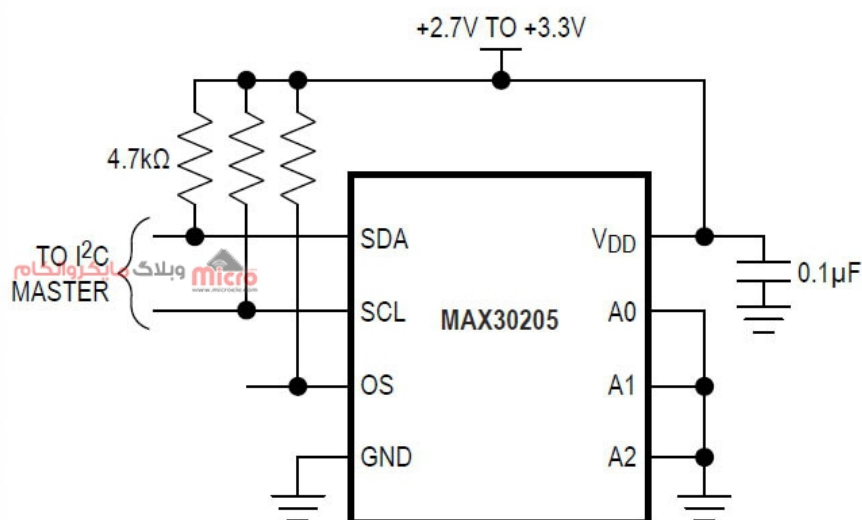
با عرض سلام خدمت همراهان سایت میکروالکام. در مطالب قبلی نحوه راه اندازی و استفاده از سنسور دما DS18B20 بصورت تکی و چند تایی با میکروکنترلر AVR مورد بررسی قرار گرفت. در این مطلب نحوه راه اندازی سنسور دمای MAX30205 با میکروکنترلر AVR و کامپایلر کدویژن مورد بررسی قرار خواهد گرفت. پس با من تا انتهای مطلب همراه باشید. همچنین میتونید سایر مطالب من رو از [این قسمت](#) مطالعه کنید.



معرفی سنسور دما MAX30205

MAX30205 یک سنسور سریع و دقیق، مناسب برای اندازه‌گیری دمای بدن انسان می‌باشد. این سنسور، دمای اندازه‌گیری شده را با استفاده از یک مبدل آنالوگ به دیجیتال (ADC) سیگما-دلتا، که تفکیک پذیری بسیار بالایی دارد به فرم دیجیتال تبدیل می‌کند. همچنین ارتباط با این سنسور، از طریق پروتکل I2C صورت می‌گیرد. MAX30205 سه خط انتخاب آدرس دارد که با ترکیبی از دو خط SDA و SCL طبق جدول (Table 1) دیتا شیت مجموعاً 32 آدرس را شامل می‌شود. ولتاژ کاری سنسور، بین 2.7V تا 3.3V بوده و جریان کل مصرفی آن حدود 600uA است.

به دلیل کوچک، دقیق و سریع بودن این سنسور و همچنین استفاده از پروتکل I2C و تعداد کم سیم‌های مورد نیاز جهت برقراری ارتباط با آن، استفاده از MAX30205 برای پروژه‌های مربوط به تجهیزات پزشکی بسیار ایده آل می‌باشد. این قطعه در پکیج TDFN 8-pin موجود بوده و دمای کاری آن بین 0 درجه سانتیگراد تا +50 درجه سانتیگراد می‌باشد. در شکل زیر، نحوه سیم‌بندی این قطعه نمایش داده شده است.



سنسور دما MAX30205



مزایا و ویژگی‌های سنسور دمای MAX30205

1- دقت و سرعت بالای این سنسور و همچنین ولتاژ کاری پایین آن، به طراحان این اجازه را می‌دهد تا بتوانند در استفاده از منابع مورد نیاز جهت طراحی مدار خود، صرفه جویی کنند.

0.1 °C Accuracy (37 °C to 39 °C)

16Bit (0.00390625 °C) Temperature Resolution

2.7V to 3.3V Supply Voltage Range

2- وجود حالت‌های One-shot و Shutdown که به کاهش مصرف توان کمک می‌کند.

3- جریان کل مصرفی 600uA.

4- داشتن توابع دیجیتالی که امکان استفاده از این سنسور در هر سامانه‌ای را به راحتی ممکن می‌سازد.

هدف آزمایش

در این مطلب قصد داریم با استفاده از میکروکنترلر AVR و استفاده از کامپایلر CodeVision 3.12، سنسور دمای MAX30205 را بدون استفاده از کتابخانه آماده و صرفاً از طریق توابع I2C موجود در خود کامپایلر کدویژن و با مطالعه دیتاشیت این قطعه، راه اندازی و مقادیر دما را بر روی یک LCD کاراکتری 2 در 16 نمایش دهیم.

وسایل مورد نیاز

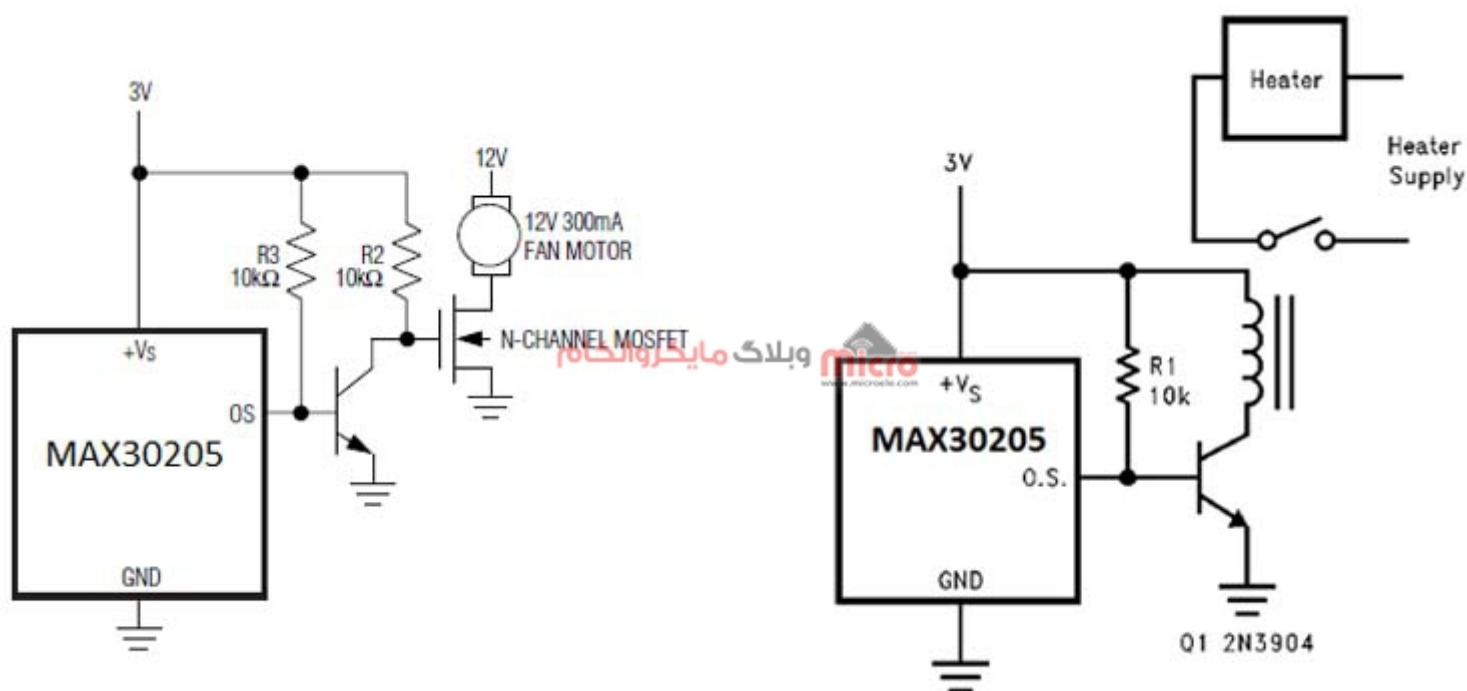
- میکروکنترلر AVR
- LCD کاراکتری 2 در 16
- پناسیومتر 10 کیلو اهم جهت تنظیم کنتراست LCD
- ماژول سنسور دمای MAX30205 (CJMCU)
- برد برد
- سیم برد بردی



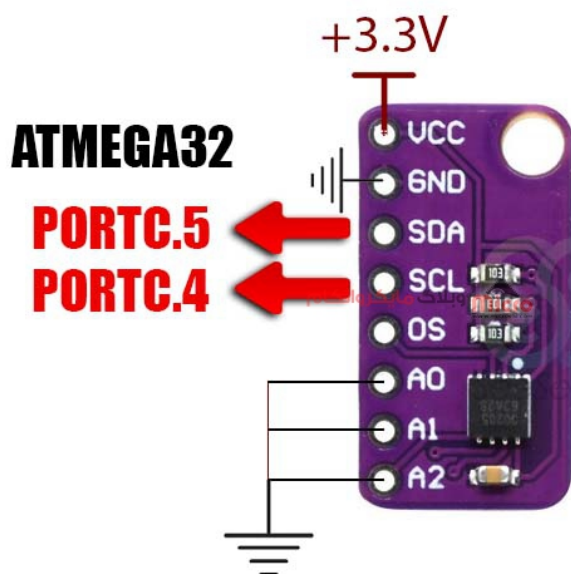
اتصال ماژول به میکروکنترلر

مانند تصویر زیر، پایه های تغذیه ماژول (MAX30205) را متصل کرده و همچنین پایه های SDA و SCL آن که مربوط به پروتکل I2C می باشد، به پایه های I2C میکروی AVR مورد نظر خود که در این جا برای ما ATMEGA32 می باشد وصل می کنیم. (دقت شود که حداکثر ولتاژ کاری سنسور 3.3 ولت می باشد).

همچنین پایه های A1، A0 و A2 مربوط به آدرس سنسور MAX30205 بوده که در این مدار به GND متصل شده. پایه OS نیز برنامه پذیر بوده و می توانید این پایه را طوری برنامه ریزی کنید که بر روی دو دمای مورد نظر شما فعال و غیر فعال شود. این پایه می تواند به یک مدار هشدار دهنده یا گرمایشی و یا سرمایشی متصل شده و روی دماهای تعیین شده روشن و خاموش شود. در شکل های زیر نمونه هایی از این نوع مدارها آورده شده است.



مدار های راه اندازی ماژول MAX30205



اتصالات سنسور MAX30205 به میکروکنترلر

Pin Description

PIN	NAME	FUNCTION
1	SDA	Serial-Data Input/Output Line. Open-drain. Connect SDA to a pullup resistor. High impedance for supply voltages from 0 to 3.3V.
2	SCL	Serial-Data Clock Input. Open-drain. Connect SCL to a pullup resistor. High impedance for supply voltages from 0 to 3.3V.
3	OS	Overtemperature Shutdown Output. Open-drain. Connect OS to a pullup resistor.
4	GND	Ground
5	A2	I ² C Slave Address Input. Connect A2 to GND or V _{DD} to set the desired I ² C bus address. Do not leave unconnected (Table 1).
6	A1	I ² C Slave Address Input. Connect A1 to GND, V _{DD} , SDA, or SCL to set the desired I ² C bus address. Do not leave unconnected (Table 1). High impedance for supply voltages from 0 to 3.3V.
7	A0	I ² C Slave Address Input. Connect A0 to GND, V _{DD} , SDA, or SCL to set the desired I ² C bus address. Do not leave unconnected (Table 1). High impedance for supply voltages from 0 to 3.3V.
8	V _{DD}	Positive 3.3V Supply Voltage Input. Bypass to GND with a 0.1μF bypass capacitor.
—	EP	Exposed Pad (Bottom-Side of Package). Connect EP to GND.

مشخصات پایه های سنسور MAX30205

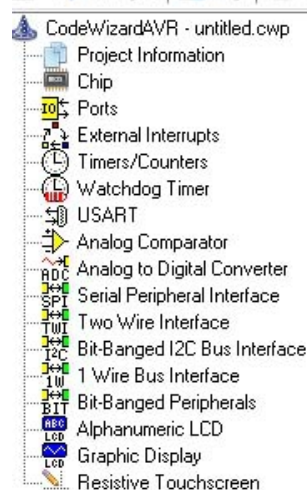


برنامه نویسی در محیط کدویژن

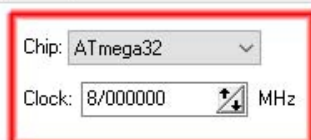
کدویژن را باز کرده و CodeWizard را اجرا کنید. در قسمت Chip Settings میکروی مورد نظر خود را که در این جا برای ما ATMEGA32 است را انتخاب و سپس فرکانس کاری میکرو را تنظیم کنید. در این مطلب ما فرکانس میکرو را بر روی 8 مگاهرتز قرار داده ایم.

CodeWizardAVR - untitled.cwp

File Program Edit Help



Chip Settings



☐ Check Reset Source

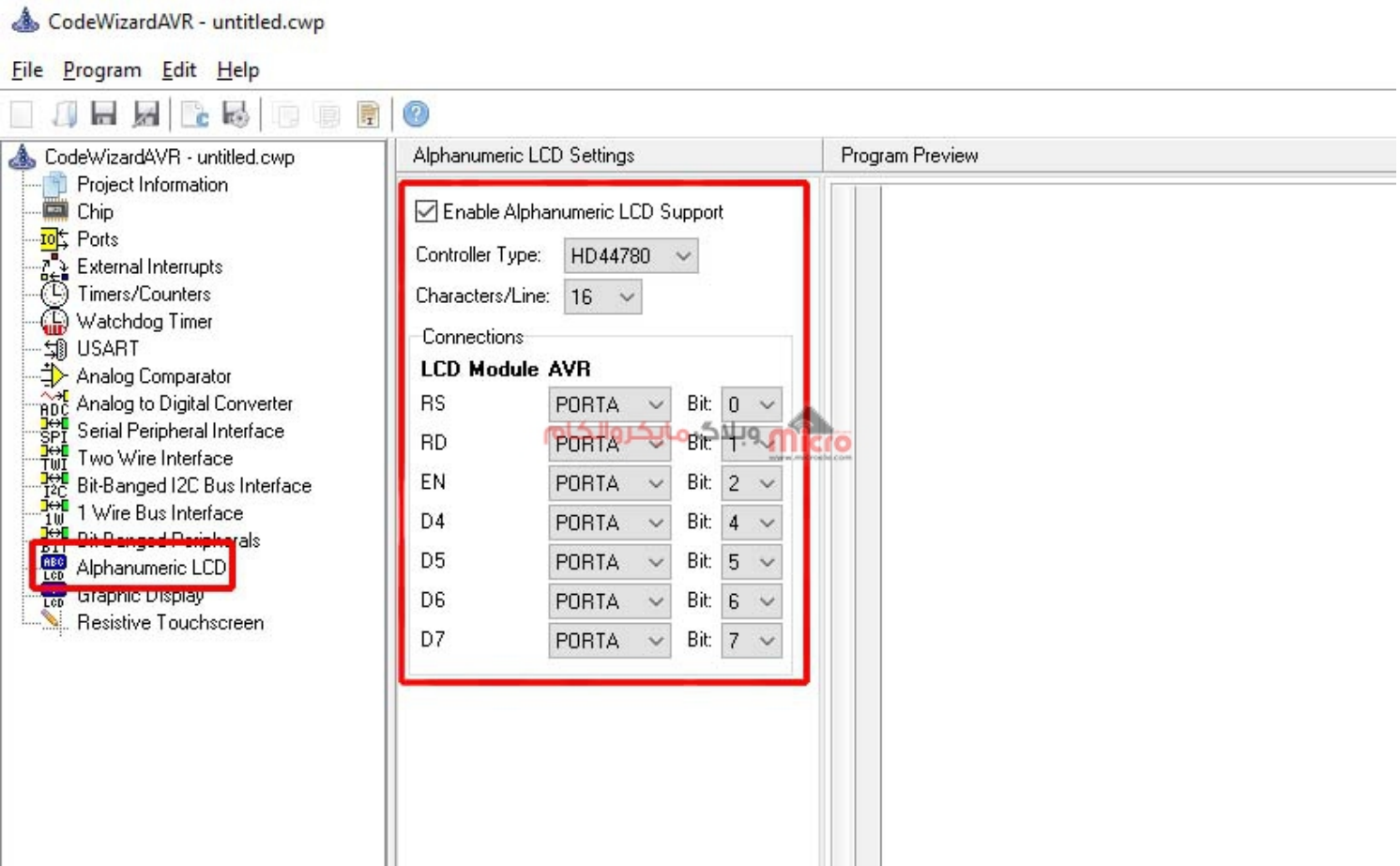
Program Type:

Application

Program Preview

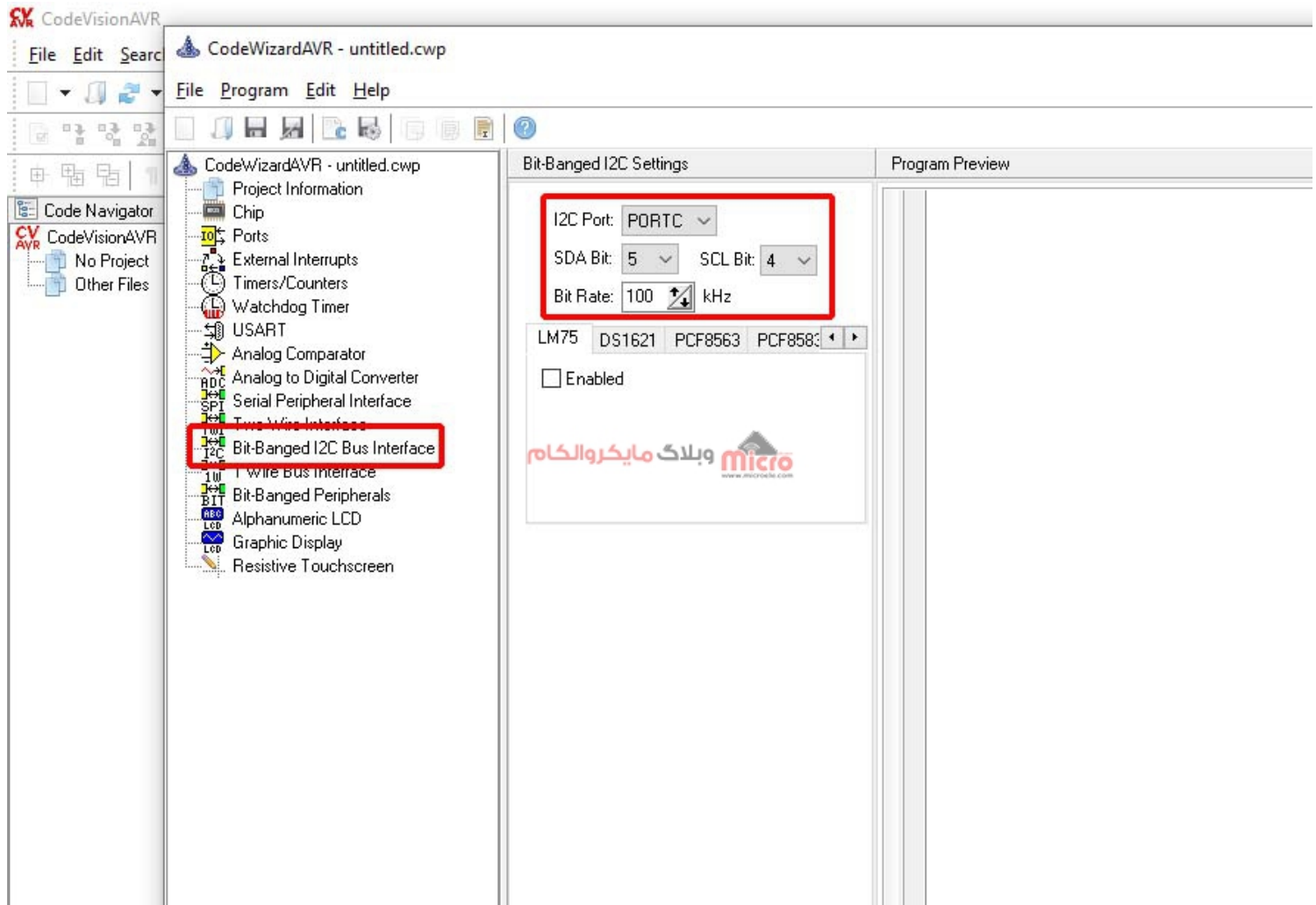
تنظیمات انتخاب چیپ و فرکانس میکرو

در ادامه، به بخش Alphanumeric LCD رفته و آن را فعال کرده و تنظیمات مربوط به پورت و نحوه اتصال آن به پایه‌های را انجام دهید.



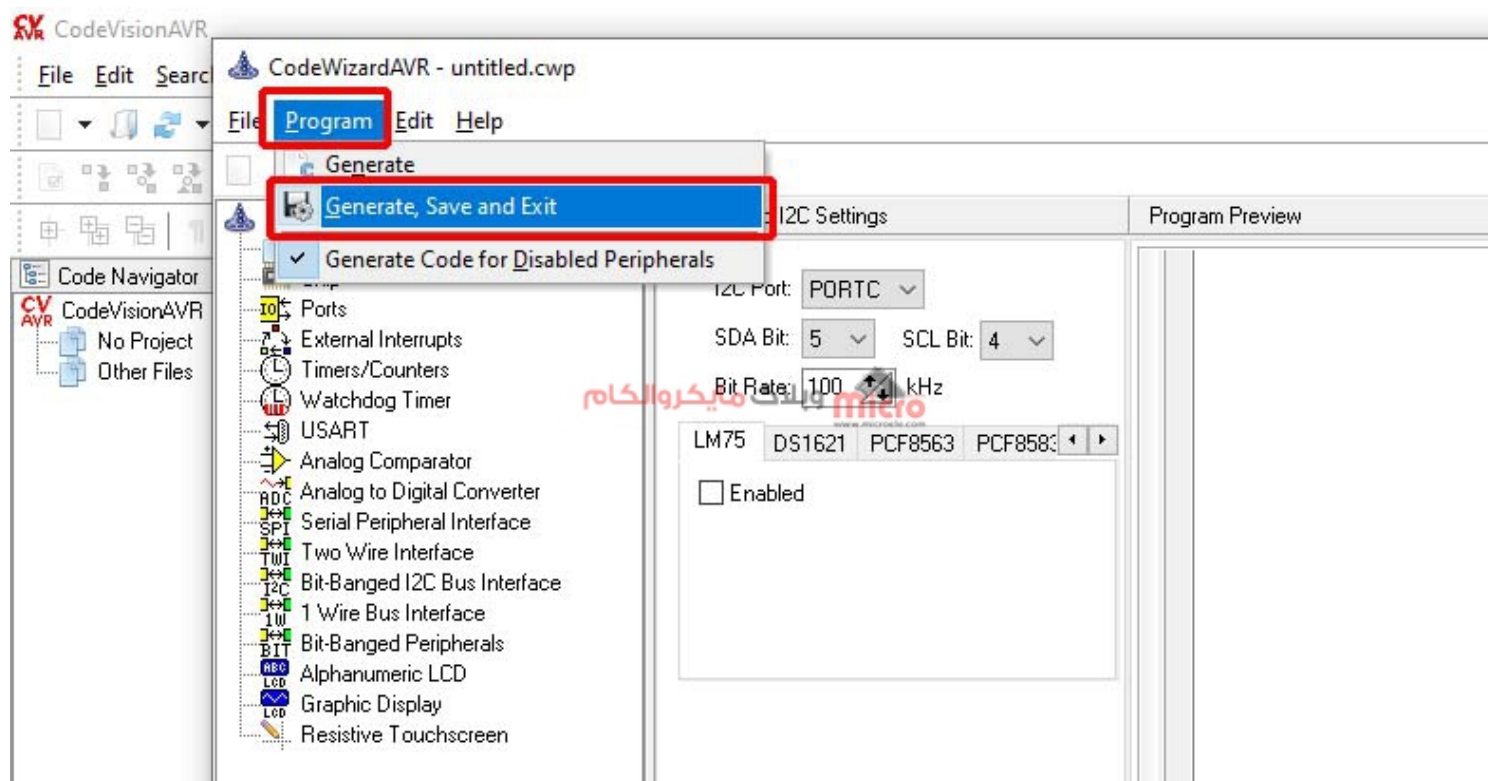
تنظیمات مربوط به کانفیگ LCD

حال در کد ویزارد، بخش Bit-Banged I2C Bus Interface را انتخاب و تنظیمات I2C را مطابق شکل انجام دهید. (بهتر است نرخ ارسال و دریافت دیتا بر روی 100khz قرار داده شده باشد. همچنین می‌توانید پورت و پایه‌های میکرو را جهت اتصال پایه‌های SDA و SCL سنسور به میکرو، تغییر دهید.)



تنظیمات مربوط به بخش I2C

تا این جای کار تنظیمات مربوط به ویزارد تمام شده است. به منوی Program رفته و با انتخاب گزینه Generate, Save and Exit پروژه را تولید و ذخیره کنید.



تولید و ذخیره پروژه در کدویژن

شروع برنامه نویسی در محیط کدویژن

قدم اول پیدا کردن آدرس سنسور MAX30205 با استفاده از دیتاشیت آن است. با توجه به جدول ارائه شده در دیتاشیت سنسور (Table 1)، و از آن جایی که ما هر سه پایه آدرس سنسور (A0، A1 و A2) را به زمین متصل کردیم، بنابراین آدرس قطعه برای نوشتن 0x90 خواهد بود که به آن آدرس Write می گوئیم. توجه داشته باشید که بیت صفرم Device Address در قطعاتی که با پروتکل I2C به میکرو متصل می شوند مشخص کننده نوشتن یا خواندن از دیوایس مورد نظر می باشد. اگر این بیت صفر باشد یعنی می خواهیم در قطعه اطلاعاتی را بنویسیم و اگر یک باشد یعنی می خواهیم از قطعه بخوانیم. بنابراین برای Read کردن از سنسور، آدرس مربوطه 0x91 خواهد بود.



Table 1. MAX30205 Slave Address Selection

A2 CONNECTION	A1 CONNECTION	A0 CONNECTION	SLAVE ADDRESS BYTE (hex)
GND	GND	GND	90h
GND	GND	V _{DD}	92h
GND	GND	SCL	82h
GND	GND	SDA	80h
GND	V _{DD}	GND	94h
GND	V _{DD}	V _{DD}	96h
GND	V _{DD}	SCL	86h
GND	V _{DD}	SDA	84h

آدرس های سنسور دما MAX30205

حال در ابتدای برنامه خود آدرس های Read و Write را define می کنیم. ADDR آدرس خواندن و ADDW آدرس نوشتن خواهد بود.

```
#define ADDR      0x91 // Address to Read from MAX30205
#define ADDW      0x90 // Address to Write from MAX30205
```



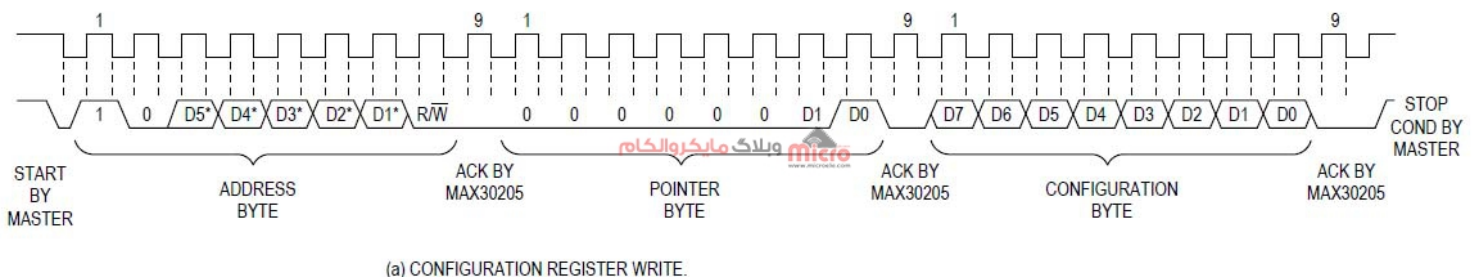
CodeVisionAVR - F:\microele\MAX30205\Project\MAX30205_MICROELECTRONICS_CodeVision\MAX30205.prj

File Edit Search View Project Tools Settings Help

```
22  *****/
23
24  #include <mega32.h>
25
26  // I2C Bus Functions
27  #include <i2c.h>
28
29  // Alphanumeric LCD functions
30  #include <alcd.h>
31
32  #define ADDR      0x91  // Address to Read from MAX30205
33  #define ADDW      0x90  // Address to Write from MAX30205
34
35  // Declare your global variables here
36
37  void main(void)
38  {
39  // Declare your local variables here
```

معرفی آدرس های خواندن و نوشتن

اعمال تنظیمات اولیه و Initialize کردن سنسور MAX30205



تنظیمات مربوط به راه اندازی MAX30205



Table 2. Register Functions and POR State

REGISTER NAME	ADDRESS (Hex)	POR STATE		POR STATE (°C)	READ/ WRITE
		Hex	BINARY		
Temperature	00	0000h	0000 0000 0000 0000	0	Read-only
Configuration	01	00h	0000 0000	—	R/W
T _{HYST}	02	4B00h	0100 1011 0000 0000	75	R/W
T _{OS}	03	5000h	0101 0000 0000 0000	80	R/W

جدول مربوط به تنظیمات سنسور دما MAX30205

با توجه به (Table 2)، ابتدا باید بر اساس فریم دیتای مربوط به اعمال تنظیمات، دیوایس را به حالت Config برده و مقدار مناسب را در رجیستر پیکر بندی قطعه قرار دهیم. برای این منظور باید در آدرس 0x01 که Configuration می باشد، مقدار مناسب، که در جدول 0x00 می باشد را قرار دهیم. یک تابع در برنامه به نام MAX30205_begin ایجاد کرده و موارد گفته شده را مطابق با فریم موجود در دیتاشیت، پیاده می کنیم.

```
void MAX30205_begin(void)
{
    i2c_start();
    i2c_write(ADDW);
    i2c_write(0x01);
    i2c_write(0x00);
    i2c_stop();
    delay_ms(50);
}
```



CodeVisionAVR - F:\microele\MAX30205\Project\MAX30205_MICROELECOM_CodeVision\MAX30205.prj

File Edit Search View Project Tools Settings Help

Code Navigator

Project: MAX30205

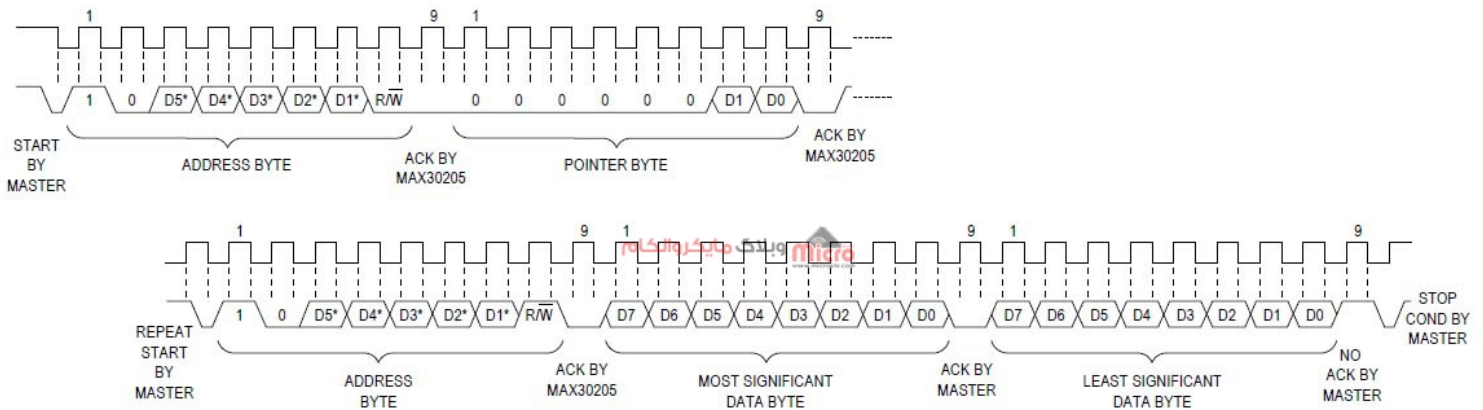
- Notes
- main.c
- Headers
 - alcd.h
 - delay.h
 - i2c.h
 - io.h
 - mega32.h
 - mega32_bits.h
 - stdarg.h
 - stdio.h
- List Files
 - MAX30205.asm
 - MAX30205.lst
 - MAX30205.map
- Other Files

main.c

```
37
38 // Declare your global variables here
39
40
41 void MAX30205_begin(void)
42 {
43     i2c_start();
44     i2c_write(ADDW);
45     i2c_write(0x01);
46     i2c_write(0x00);
47     i2c_stop();
48     delay_ms(50);
49 }
50
51
52
53 void main(void)
54 {
55     // Declare your local variables here
56
57     // Input/Output Ports initialization
```

معرفی تابع MAX30205_begin

نوشتن تابع برای خواندن دما از سنسور MAX30205



*SEE TABLE 1.

(b) TYPICAL POINTER SET FOLLOWED BY IMMEDIATE READ FOR 2-BYTE REGISTER SUCH AS TEMPERATURE, T_{OS} , AND T_{HYST} .

*SEE TABLE 1.

تنظیمات مربوط به راه اندازی MAX30205

حال می‌خواهیم تابعی جهت خواندن دما از سنسور ایجاد کنیم. با توجه به دیتاشیت و مشاهده فریم مربوط به دریافت دما که در تصویر بالا نیز آورده شده است، مراحل به شرح زیر می‌باشد. شایان ذکر است این تابع دارای خروجی از نوع float بوده تا بتواند مقادیر اعشاری دما را نیز برگرداند.

1. Start کردن I2C.
2. نوشتن آدرس Write.
3. نوشتن آدرس Temperature Mode که 0x00 می‌باشد.
4. Stop کردن I2C.
5. Start مجدد I2C.
6. نوشتن آدرس خواندن دما.
7. خواندن بخش Most Significant byte دیتا که مربوط به قسمت صحیح دما می‌باشد با acknowledge یک.
8. خواندن بخش Least Significant byte دیتا که مربوط به قسمت اعشار دما می‌باشد با acknowledge یک.
9. Stop کردن I2C.

```
float MAX30205_getTemperature(void)
{
    unsigned char readRaw[2] = {0};
    int raw;
    float temperature;
```



```
readRaw[0]=readRaw[1]=23;
i2c_start();
i2c_write(ADDW);
delay_ms(5);
i2c_write(0x00);
i2c_stop();
i2c_start();
i2c_write(ADDR);
readRaw[0]=i2c_read(1);
readRaw[1]=i2c_read(1);
i2c_stop();
raw = (readRaw[0] << 8) | readRaw[1];
temperature = raw * 0.00390625;
return temperature;
}
```



CodeVisionAVR - F:\microele\MAX30205\Project\MAX30205_MICROELECTRONICS_CodeVision\MAX30205.prj

File Edit Search View Project Tools Settings Help

Code Navigator

Project: MAX30205

- Notes
- main.c
- Headers
 - alcd.h
 - delay.h
 - i2c.h
 - io.h
 - mega32.h
 - mega32_bits.h
 - stdarg.h
 - stdio.h
- List Files
 - MAX30205.asm
 - MAX30205.lst
 - MAX30205.map
- Other Files

Notes main.c

```
52
53 float MAX30205_getTemperature(void)
54 {
55     unsigned char readRaw[2] = {0};
56     int raw;
57     float temperature;
58     readRaw[0]=readRaw[1]=23;
59     i2c_start();
60     i2c_write(ADDW);
61     delay_ms(5);
62     i2c_write(0x00);
63     i2c_stop();
64     i2c_start();
65     i2c_write(ADDR);
66     readRaw[0]=i2c_read(1);
67     readRaw[1]=i2c_read(1);
68     i2c_stop();
69     raw = (readRaw[0] << 8) | readRaw[1];
70     temperature = raw * 0.00390625;
71     return temperature;
72 }
```

تابع مربوط به خواندن دما از سنسور MAX30205

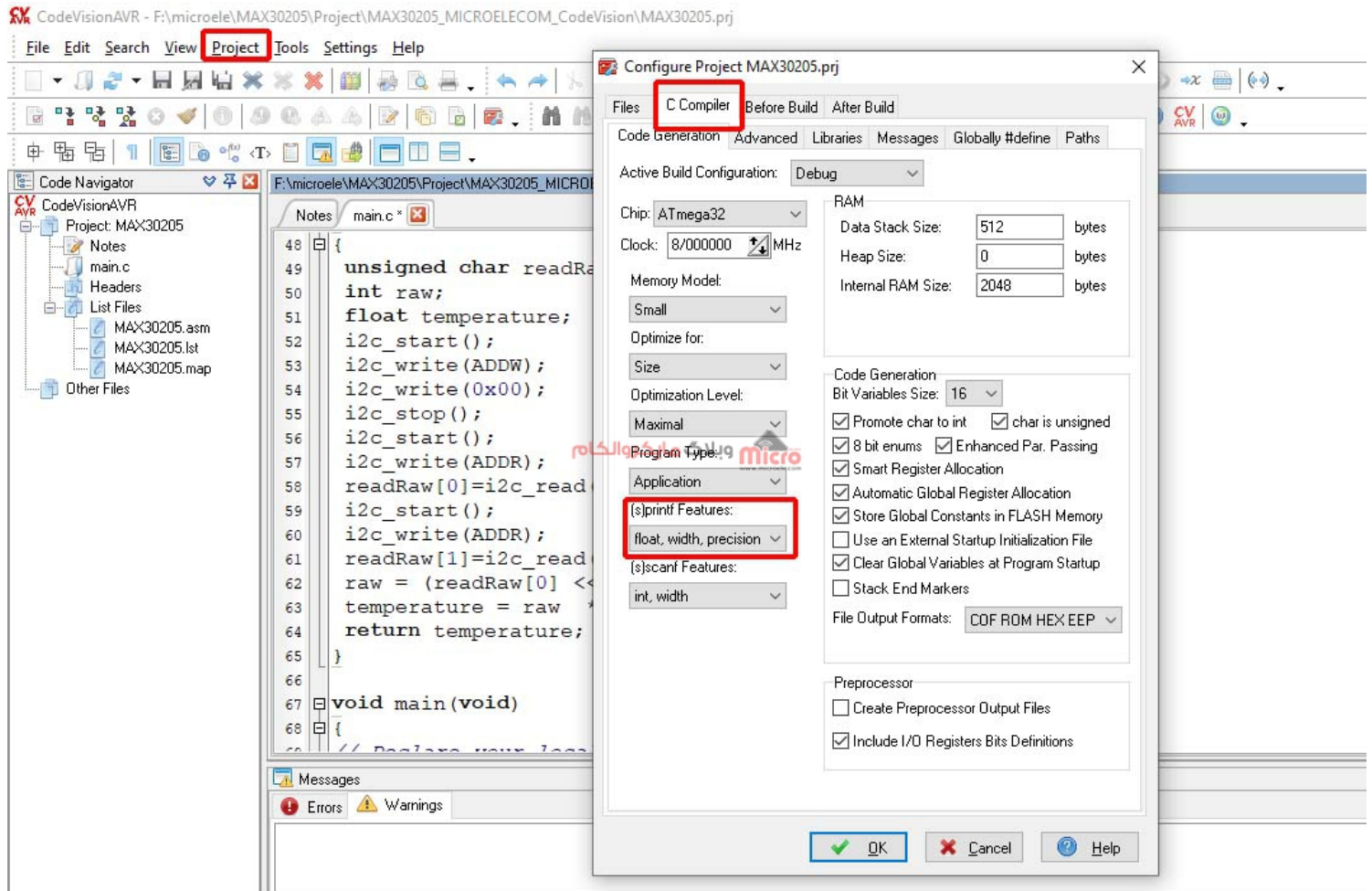
تشریح تابع نوشته شده برای خواندن دما

در خط اول یک آرایه با دو خانه به نام readRaw برای خواندن دیتای خام از نوع unsigned char تعریف شده است تا بخش های صحیح و اعشاری دما در این آرایه قرار گیرند. در خط دوم متغیر raw از نوع int تعریف شده است که بعد از کنار هم قرار دادن هر دو دیتای خوانده شده در آرایه readRaw مقدار دیتای خام در آن ذخیره می گردد. دیتای مربوط به بخش صحیح ابتدا 8 بار به سمت چپ شیف داده شده و با دیتای مربوط به بخش اعشار OR منطقی می شود و در متغیر raw ذخیره می گردد.

در نهایت متغیر raw را برای بدست آمدن مقدار صحیح دما در عدد 0.00390625 ضرب کرده و در متغیر temperature که از نوع float تعریف شده است قرار می دهیم. با توجه به این که می خواهیم از توابع sprintf برای چاپ مقادیر float بر



روی lcd استفاده کنیم لازم است که از منوی Project و سپس باز کردن بخش Configuration در کدویژن و رفتن به سربرگ C Compiler گزینه sprintf features را بر روی float, width, precision قرار دهیم.



تنظیمات مربوط به تابع sprintf برای نمایش مقادیر float

اضافه کردن کتابخانه‌ها و متغیرهای لازم جهت خواندن دما

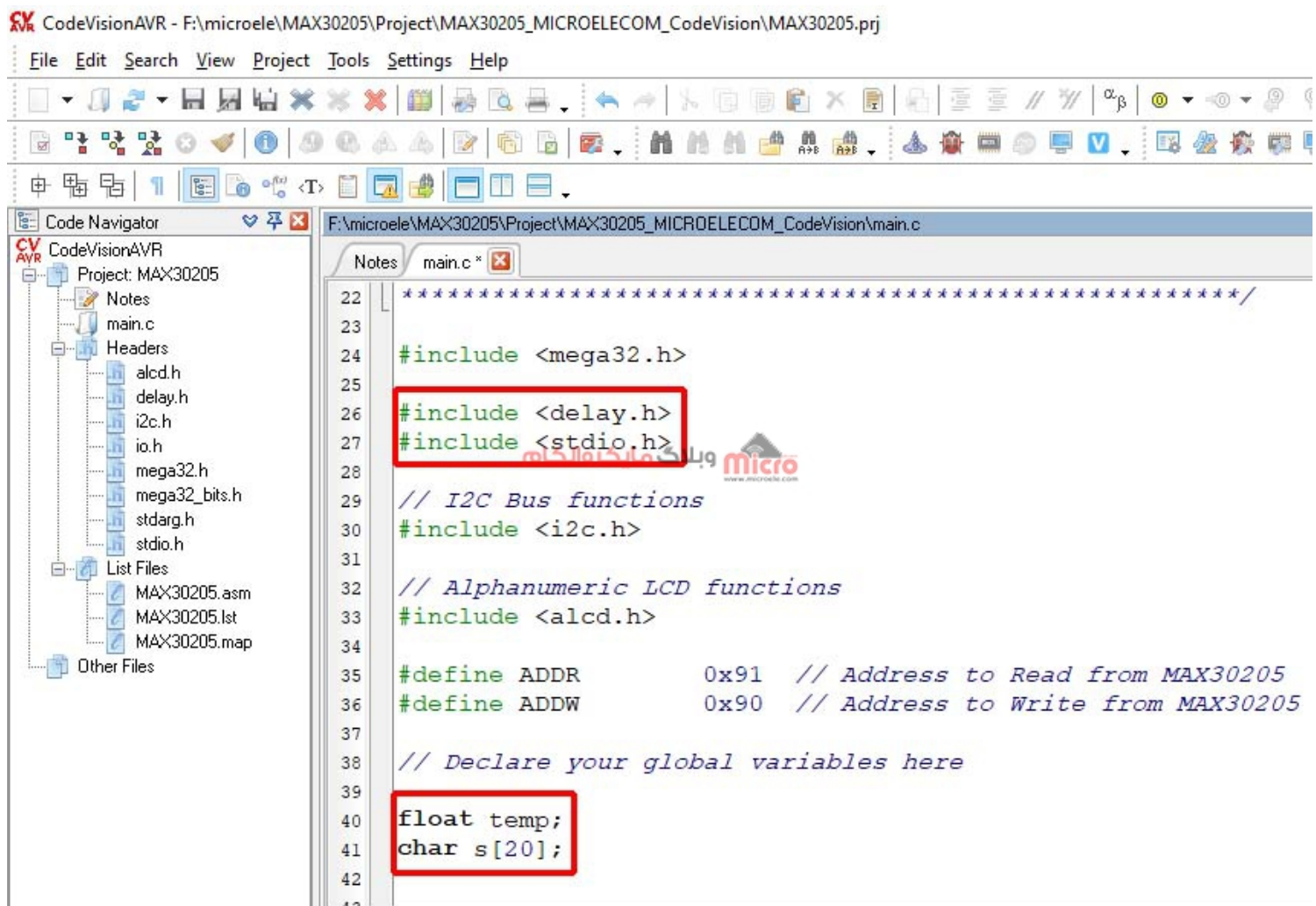
در ابتدای برنامه کتابخانه‌های delay و stdio و همچنین متغیرهای temp و s را به برنامه اضافه می‌کنیم. از متغیر s برای تبدیل مقادیر float به کاراکتر جهت چاپ بر روی LCD استفاده خواهد شد.

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



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

```
float temp;  
char s[20];
```



معرفی کتابخانه و متغیرهای مورد نیاز



تکمیل حلقه While و نحوه استفاده از توابع نوشته شده

قبل از حلقه while تابع MAX30205_begin را فراخوانی کرده و مانند تصویر زیر کدهای مربوطه جهت خواندن دما و چاپ مقادیر بر روی LCD را نوشته و تکمیل می‌کنیم.

```
MAX30205_begin();  
while (1)  
{  
    // Place your code here  
    temp = MAX30205_getTemperature();  
    sprintf(s, "Temp:%4.2f", temp);  
    lcd_gotoxy(0,0);  
    lcd_puts(s);  
    delay_ms(1000);  
}  
}
```



```
206 lcd_init(16);
207 lcd_clear();
208 lcd_gotoxy(4,0);
209 lcd_putsf("MAX30205");
210 lcd_gotoxy(0,1);
211 lcd_putsf("www.microele.com");
212 delay_ms(2500);
213 lcd_clear();
214 //-----
215 MAX30205_begin();
216 while (1)
217 {
218     // Place your code here
219     temp = MAX30205_getTemperature();
220     sprintf(s,"Temp:%4.2f",temp);
221     lcd_gotoxy(0,0);
222     lcd_puts(s);
223     delay_ms(1000);
224 }
225
226 }
```

تکمیل حلقه while

پروگرام کردن برنامه بر روی میکرو

برنامه را Build کرده و از بخش Chip Programmer در کدویژن، میکرو را پروگرام کنید. دقت شود که فیوزبیت‌های میکرو، متناسب با Clock ای که در برنامه تعیین شده است تنظیم گردد که در این جا ما برنامه را بر روی فرکانس 8Mhz تنظیم کرده بودیم بنابراین فیوزبیت‌های میکرو را بر روی 8 مگاهرتز داخلی قرار داده‌ایم.



CodeVisionAVR - F:\microele\MAX30205\Project\MAX30205_MICROELECTRONICS\CodeVision\MAX30205.prj

File Edit **Program** Read Compare Help

Chip: Erase Chip
SCK Frequency: FLASH Hz
FLASH: EEPROM
Start: 0 h End: 3FF h
Checksum: 0xFC00 Program
Check: Fuse Bit(s)
Chip Program: All
FLASH Lock Bits: Programming disabled
Boot Lock Bit 0: B01=1 B02=1
Boot Lock Bit 1: B11=1 B12=1

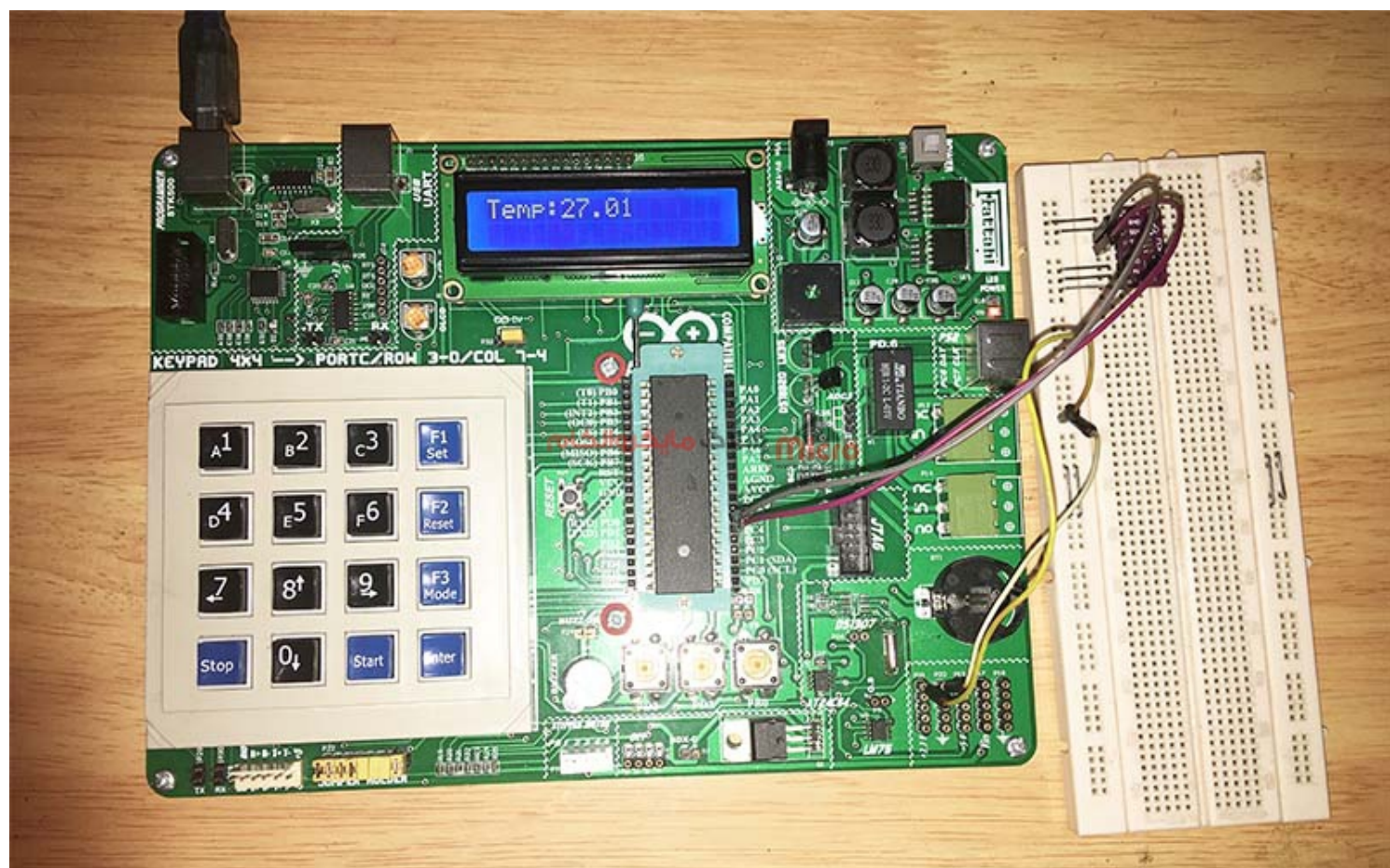
Internal 8Mhz

Program Fuse Bit(s):
☒ CKSEL0=0
☒ CKSEL1=0
☒ CKSEL2=0
☒ CKSEL3=0
☐ SUT0=0
☐ SUT1=0
☐ BODEN=0
☐ BODLEVEL=0
☐ BOOTRST=0
☐ BOOTSZ0=0
☐ BOOTSZ1=0
☐ EESAVE=0
☐ CKOPT=0
☐ JTAGEN=0
☐ OCDEN=0

☒ Check Signature ☐ Check Erasure ☐ Preserve EEPROM ☐ Verify

تنظیمات مربوط به پروگرام کردن برنامه روی میکروکنترلر

در تصویر زیر، برنامه بر روی میکرو پروگرام و اجرا شده است:



مدار کامل و مشاهده عملی نتایج

برای دریافت سورس کد مربوطه و پروگرام کردن میکروکنترلر از این [لینک](#) اقدام کنید.

مشاهده نتیجه



نتیجه گیری

در این مطلب مشخصات و راه اندازی سنسور دمای MAX30205 با میکروکنترلر توضیح داده شد. MAX30205 یک سنسور سریع و دقیق، مناسب برای اندازه‌گیری دمای بدن انسان، به همراه خروجی OS و با قابلیت برنامه ریزی می‌باشد. این سنسور، دمای اندازه‌گیری شده را با استفاده از یک مبدل آنالوگ به دیجیتال (ADC) سیگما-دلتا که تفکیک پذیری بسیار بالایی دارد به فرم دیجیتال تبدیل می‌کند. ارتباط با این سنسور، از طریق پروتکل I2C و با دو سیم صورت می‌گیرد.

امیدوارم از این آموزش کمال بهره را برده باشید. در صورتی که هرگونه نظر یا سوال داشتید درباره این آموزش لطفاً 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 حتماً 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.