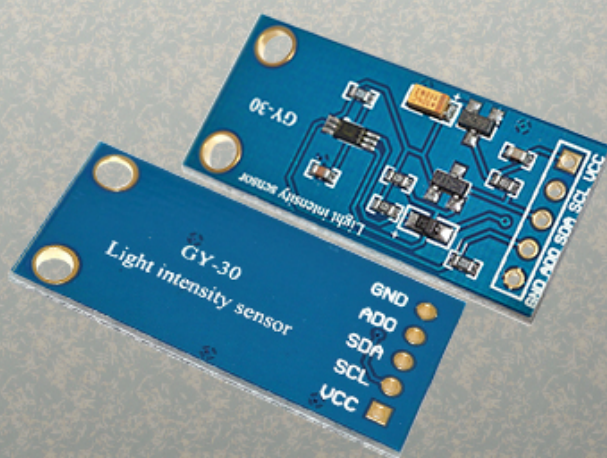
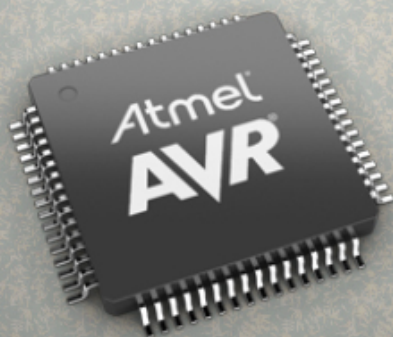




راه اندازی ماژول GY-30 و ساخت لوکس متر با AVR کدویژن

راه اندازی ماژول GY-30 و ساخت لوکس متر با AVR کدویژن



<https://blog.microele.com>

تاریخ انتشار ۱۶ آذر، ۱۴۰۰ توسط آرش فتاحی

با عرض سلام خدمت همراهان سایت میکروالکام. در مطالب قبلی **نحوه راه اندازی سنسور دمای MAX30205** بررسی گردید. در این مطلب نحوه راه اندازی سنسور لوکس متر GY-30 و ساخت دستگاه لوکس متر با میکروکنترلر AVR و کامپایلر کدویژن (CodeVision) مورد بررسی قرار خواهد گرفت. پس با من تا انتهای مطلب همراه باشید. همچنین میتونید سایر مطالب من رو از **این قسمت** مطالعه کنید.



ماژول لوکس متر GY-30

ماژول لوکس متر GY-30 مجهز به سنسور BH1750 بوده که حساس به شدت نور می‌باشد. این ماژول دارای دقت و رزولوشن بالایی است و داده‌های خروجی آن بر حسب واحد لوکس (lx) می‌باشد. این سنسور دارای یک مبدل آنالوگ به دیجیتال 16 بیتی بوده و می‌تواند مقادیر مربوط به شدت روشنایی را به مقادیر دیجیتال، در خروجی تبدیل کند. همچنین پروتکل ارتباطی این ماژول I2C است.



ماژول لوکس متر GY30

هدف آزمایش

در این مطلب قصد داریم با میکروکنترلر AVR و کامپایلر Codevision ماژول لوکس متر GY-330 را راه‌اندازی کرده و



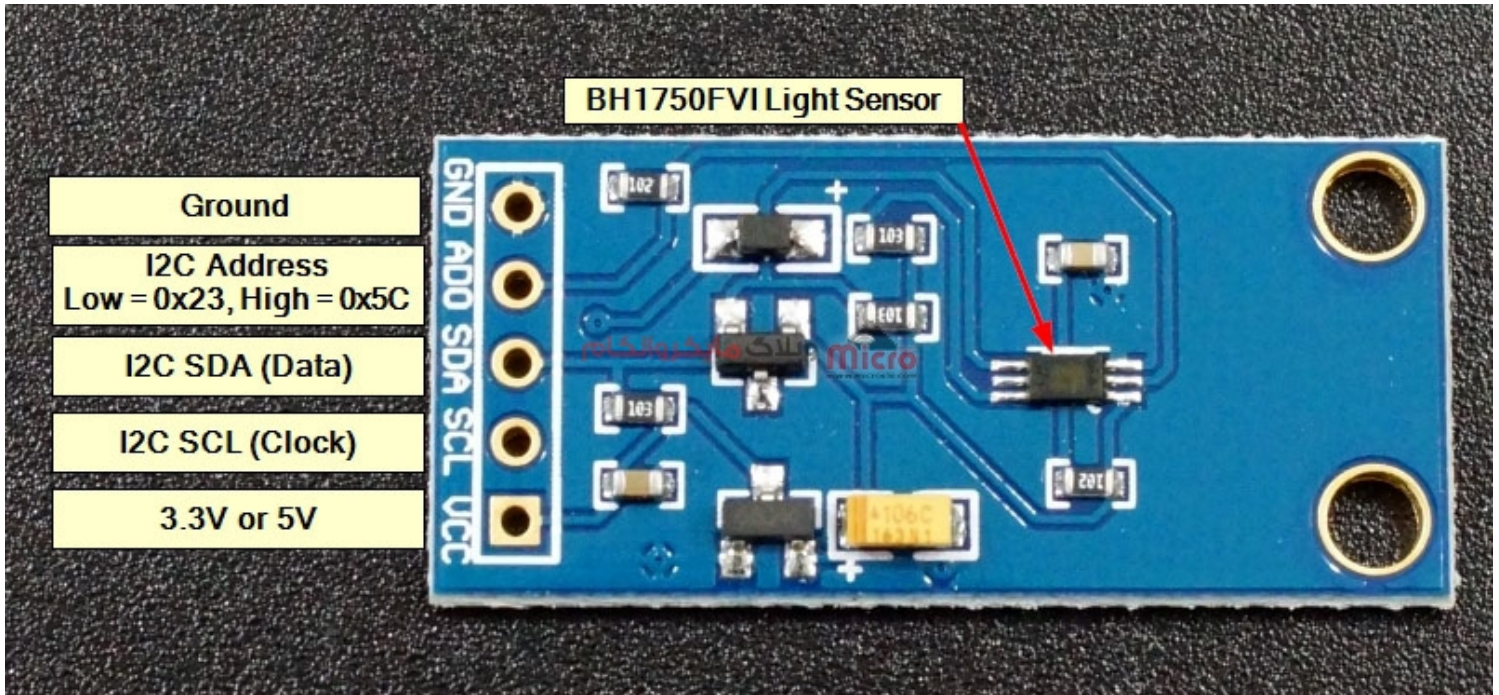
مقادیر مربوط به شدت روشنایی محیط را با واحد لوکس (lx) بر روی یک LCD کاراکتری 2x16 نمایش دهیم. برای این کار از دیتاشیت قطعه BH1750 که سنسور اصلی ماژول GY-30 می باشد کمک گرفته و با پیکربندی واحد I2C در کدویزارد نرم افزار کدویژن، برنامه را خواهیم نوشت.

وسایل و قطعات مورد نیاز

- ماژول لوکس متر GY-30
- میکروکنترلر AVR (Mega32 یا Mega8 یا ...)
- LCD کاراکتری
- دو عدد مقاومت 10k اهم
- برد برد
- سیم برد بردی

اتصال ماژول لوکس متر GY-30 به میکروکنترلر AVR

با توجه به جدول زیر پایه های ماژول را به تغذیه و میکروکنترلر AVR متصل می کنیم:

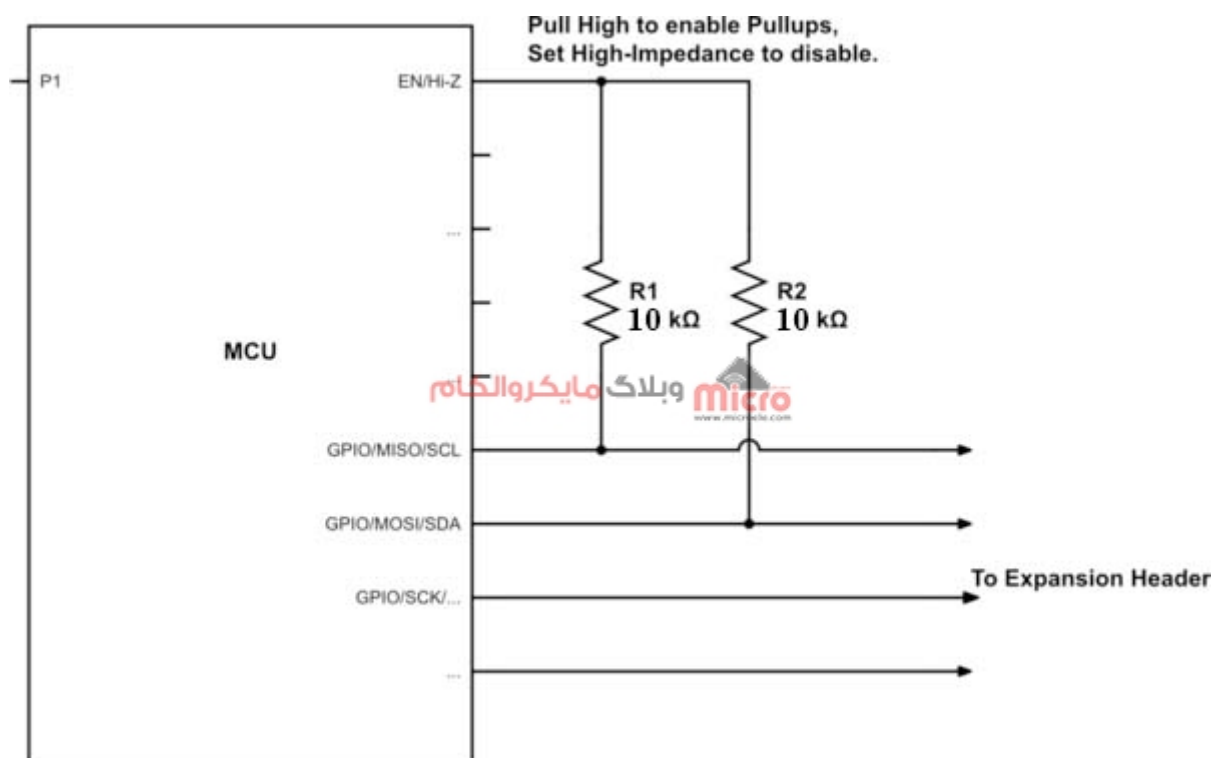


اتصال پایه‌های ماژول لوکس متر به میکرو

ماژول	میکروکنترلر
GND	GND
ADO	بدون اتصال
SDA	PC1
SCL	PC0
VCC	+5 ولت

* از آن جایی که می‌خواهیم از I2C نرم افزاری در ایجاد این پروژه استفاده کنیم، می‌توانید هر پایه دیگری از میکرو را نیز به غیر از PC0 و PC1 به عنوان پایه‌های SDA و SCL انتخاب کنید.

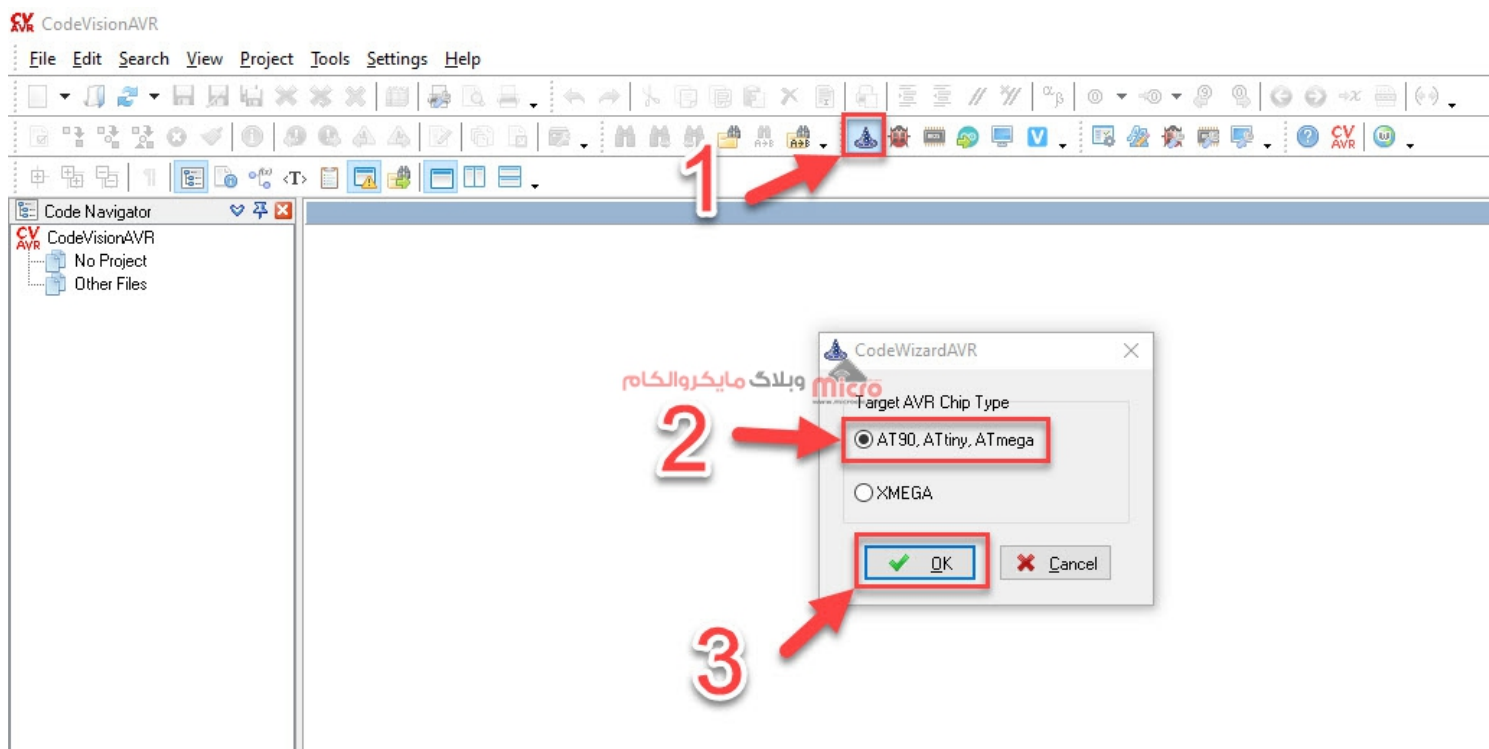
* برای عملکرد بهتر I2C، بهتر است پایه‌های SDA و SCL میکرو را بر روی برد برد با یک مقاومت 10k اهم Pull Up کنید.



pull up کردن پایه‌های SCL و SDA میکرو

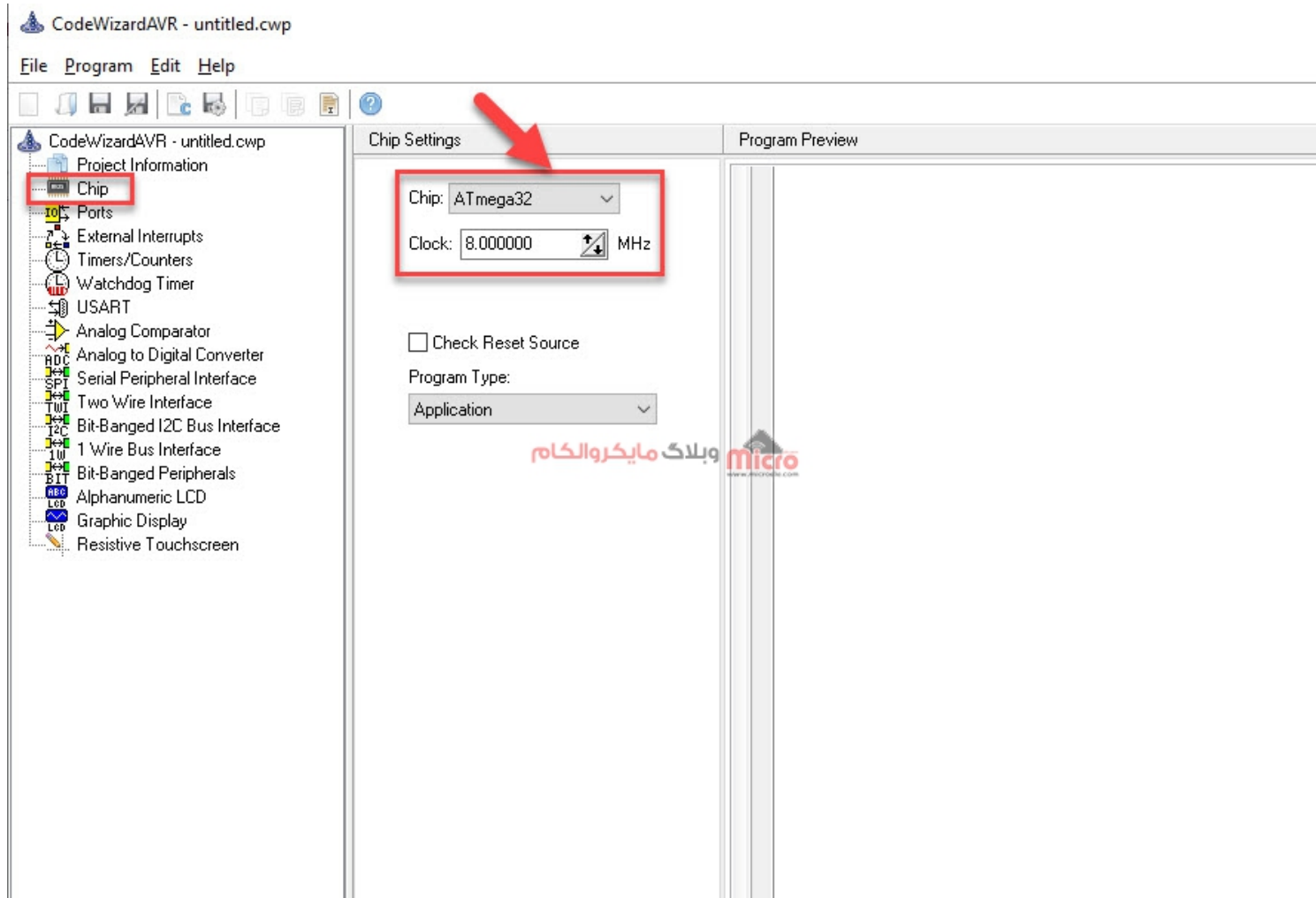
اعمال تنظیمات در محیط Codewizard در کدویژن

وارد محیط کدویژن شده و CodeWizard را اجرا کنید.



اعمال تنظیمات در Codewizard

در کدویزارد و از قسمت Chip میکروی مورد نظر خود و Clock میکرو را انتخاب کنید.



اعمال تنظیمات در محیط Codewizard

به بخش Alphanumeric LCD رفته و تنظیمات مربوط به LCD را انجام دهید. در این جا ما LCD را بر روی PORTA تنظیم کرده‌ایم.



CodeWizardAVR - untitled.cwp

File Program Edit Help

CodeWizardAVR - untitled.cwp

Project Information

- Chip
- Ports
- External Interrupts
- Timers/Counters
- Watchdog Timer
- USART
- Analog Comparator
- Analog to Digital Converter
- Serial Peripheral Interface
- Two Wire Interface
- Bit-Banged I2C Bus Interface
- 1 Wire Bus Interface
- Bit-Banged Peripherals
- Alphanumeric LCD**
- Graphic Display
- Resistive Touchscreen

Alphanumeric LCD Settings

Enable Alphanumeric LCD Support

Controller Type: HD44780

Characters/Line: 16

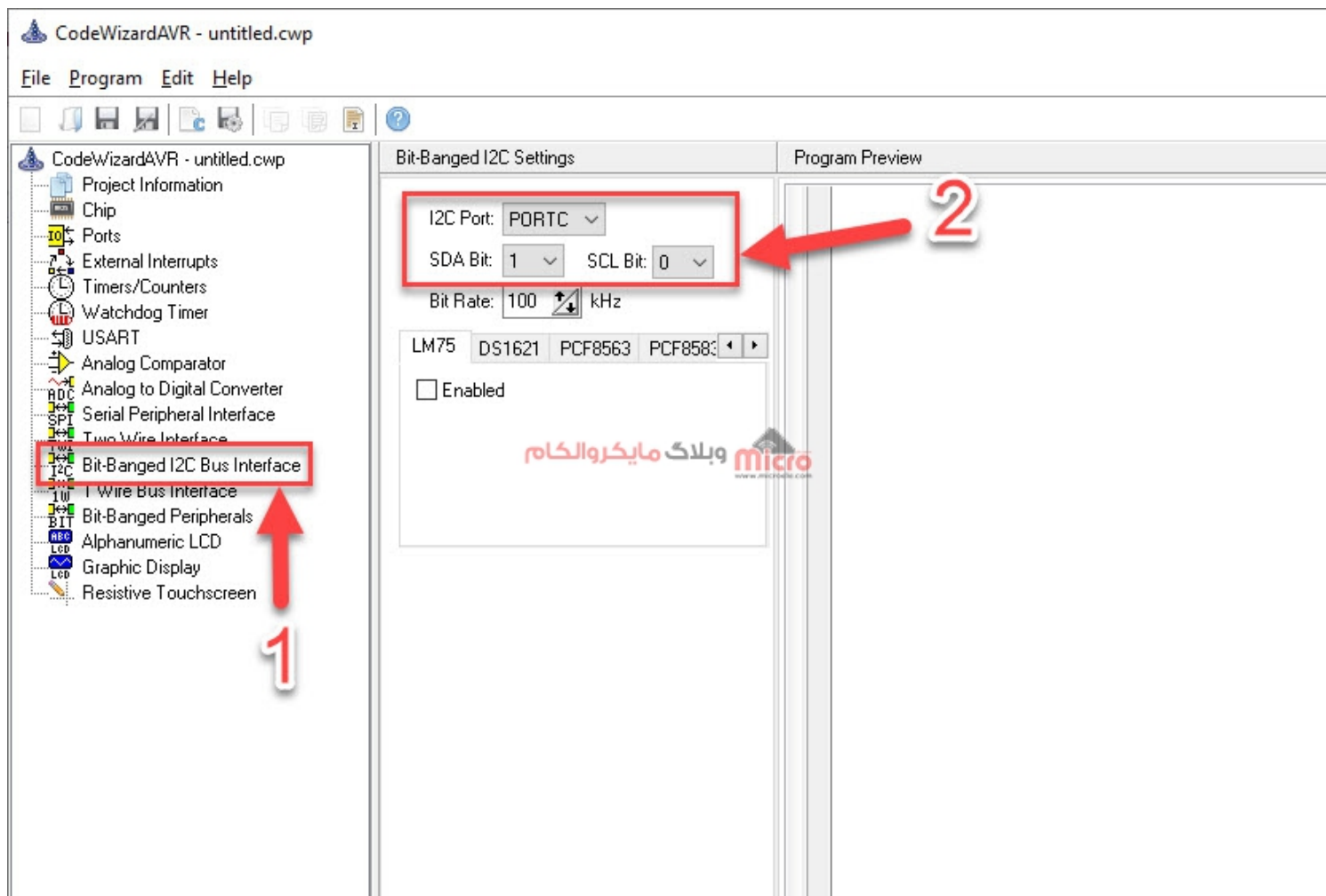
Connections

LCD Module AVR		
RS	PORTA	Bit: 0
RD	PORTA	Bit: 1
EN	PORTA	Bit: 2
D4	PORTA	Bit: 4
D5	PORTA	Bit: 5
D6	PORTA	Bit: 6
D7	PORTA	Bit: 7

Program Preview

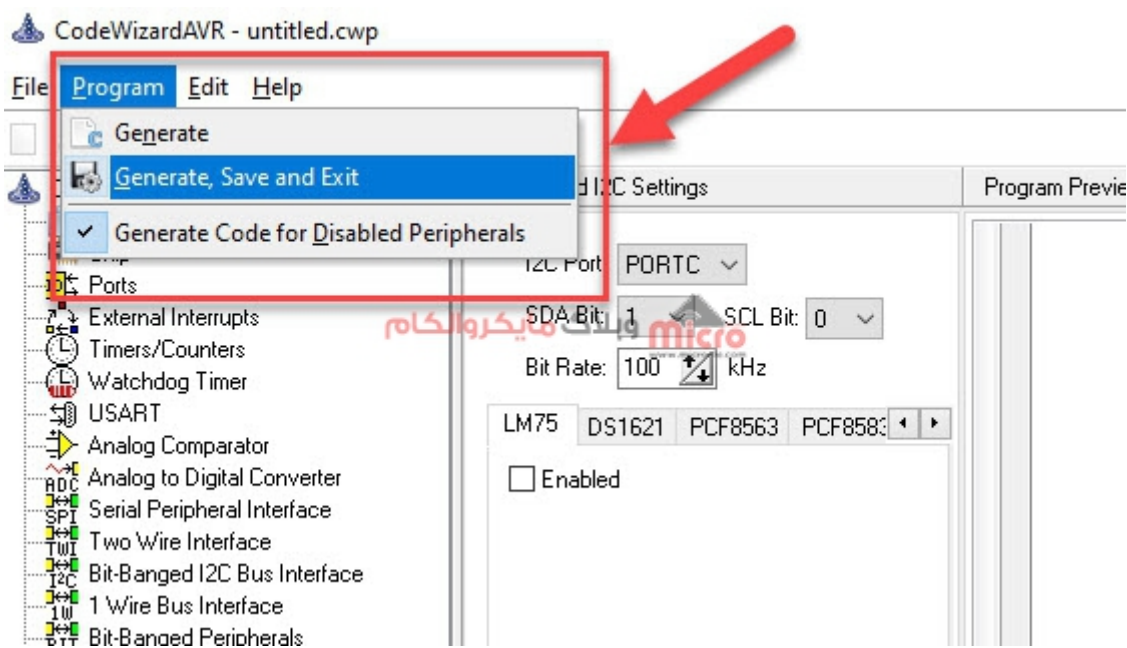
تنظیمات مربوط به LCD کاراکتری در Codewizard کدویژن

در آخر نوبت به تنظیمات مربوط به واحد I2C در ویزارد می‌رسد. به بخش Bit-Banged I2C Bus Interface رفته و پورت و پایه‌های مربوط به I2C را تنظیم کنید.



اعمال تنظیمات I2C در Codewizard

از منوی Program برنامه را Generate و Save کنید.



نخیره سازی تنظیمات و تولید کد در Codewizard

شروع کدنویسی در محیط کدویژن (Codevision)

مطالعه دیتاشیت BH1750 و استخراج دستورات

برای کدنویسی ابتدا لازم است تا دیتاشیت قطعه BH1750 را مطالعه کرده و تنظیمات مربوط به ماژول و آدرس‌های مربوطه را از داخل آن استخراج کنیم.

جدول زیر کدهای مربوط به دستورات مختلفی را که می‌توانیم به ماژول ارسال کنیم را نشان می‌دهد:



● Instruction Set Architecture

Instruction	Opecode	Comments
Power Down	0000_0000	No active state.
Power On	0000_0001	Waiting for measurement command.
Reset	0000_0111	Reset Data register value. Reset command is not acceptable in Power Down mode.
Continuously H-Resolution Mode	0001_0000	Start measurement at 11x resolution. Measurement Time is typically 120ms.
Continuously H-Resolution Mode2	0001_0001	Start measurement at 0.51x resolution. Measurement Time is typically 120ms.
Continuously L-Resolution Mode	0001_0011	Start measurement at 41x resolution. Measurement Time is typically 16ms.
One Time H-Resolution Mode	0010_0000	Start measurement at 11x resolution. Measurement Time is typically 120ms. It is automatically set to Power Down mode after measurement.
One Time H-Resolution Mode2	0010_0001	Start measurement at 0.51x resolution. Measurement Time is typically 120ms. It is automatically set to Power Down mode after measurement.
One Time L-Resolution Mode	0010_0011	Start measurement at 41x resolution. Measurement Time is typically 16ms. It is automatically set to Power Down mode after measurement.
Change Measurement time (High bit)	01000_MT[7,6,5]	Change measurement time. ※ Please refer "adjust measurement result for influence of optical window."
Change Masurement time (Low bit)	011_MT[4,3,2,1,0]	Change measurement time. ※ Please refer "adjust measurement result for influence of optical window."

※ Don't input the other opecode.

جدول دستورات BH1750

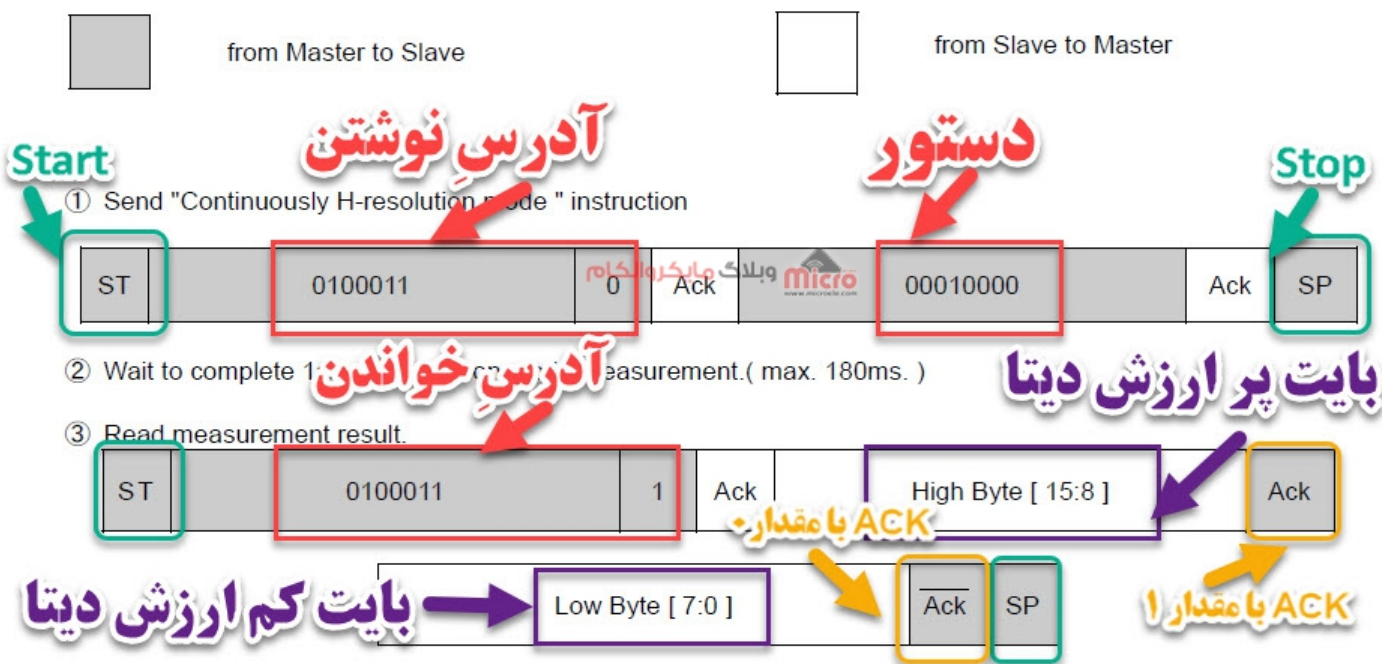
یکی از این دستورات، حالت Continuously H-Resolution می باشد که ماژول لوکس متر GY-30 را بر روی محاسبه شدت روشنایی محیط با دقت 1 لوکس تنظیم می کند. در این حالت مدت زمانی که طول می کشد تا اندازه گیری انجام گیرد 120 میلی ثانیه می باشد.

در داخل دیتاشیت نحوه ارسال این دستور و خواندن دیتا از سنسور نشان داده شده است:



● Measurement sequence example from "Write instruction" to "Read measurement result"

ex1) Continuously H-resolution mode (ADDR = 'L')



مثال مربوط به نحوه خواندن مقدار شدت روشنایی از سنسور BH1750

با توجه به مثال داخل دیتاشیت، آدرس نوشتن در ماژول لوکس متر 01000110 (0x46) و آدرس خواندن از آن، 01000111 (0x47) می باشد. همچنین کد دستور حالت Continuously H-Resolution مقدار 00010000 (0x10) است.

خواندن مقادیر از ماژول GY-30

روند کلی خواندن مقادیر مربوط به شدت روشنایی با ماژول GY-30 به شرح زیر می باشد:

ابتدا دستور Start را از طریق I2C ارسال می کنیم. سپس آدرس نوشتن در ماژول که 0x46 می باشد را Write کرده و دستور مدنظر، که در این جا حالت Continuously H-Resolution می باشد را با Write کردن 0x10 به ماژول ارسال می کنیم.

بعد از ارسال دستور خواندن، طبق دیتاشیت نیاز است تا 180 میلی ثانیه صبر کنیم تا اندازه گیری شدت نور توسط ماژول انجام شده و دیتا آماده خواندن شود.

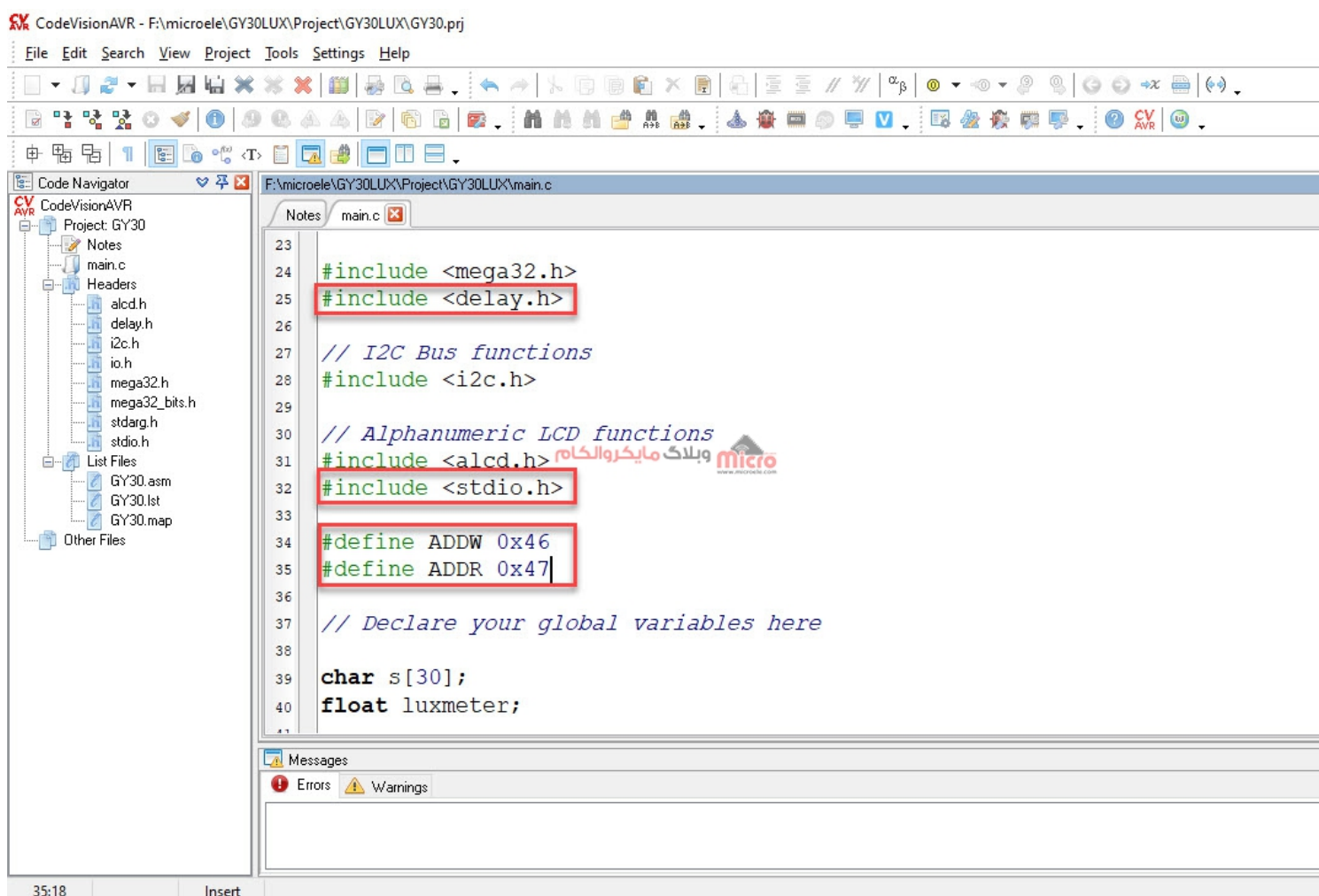
برای خواندن مقدار روشنایی، آدرس خواندن از ماژول را که 0x47 می باشد در آن Write می کنیم. ابتدا بخش پر ارزش



دیتا توسط ماژول به میکرو با ACK یک ارسال می‌شود و در آخر بخش کم ارزش دیتا با ACK صفر دریافت می‌گردد.

شروع کدنویسی

مانند تصویر زیر، در برنامه آدرس‌های نوشتن و خواندن را Define کرده و کتابخانه‌های delay.h و stdio.h را به برنامه اضافه می‌کنیم.



The screenshot shows the CodeVisionAVR IDE interface. The main window displays the code for main.c. The code includes headers for mega32.h, delay.h, and stdio.h, and defines constants ADDW and ADDR. The code is as follows:

```
23
24 #include <mega32.h>
25 #include <delay.h>
26
27 // I2C Bus functions
28 #include <i2c.h>
29
30 // Alphanumeric LCD functions
31 #include <alcd.h>
32 #include <stdio.h>
33
34 #define ADDW 0x46
35 #define ADDR 0x47
36
37 // Declare your global variables here
38
39 char s[30];
40 float luxmeter;
```

اضافه کردن کتابخانه‌ها و آدرس‌های I2C ماژول به برنامه



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

#define ADDW 0x46
#define ADDR 0x47
```

متغیرهای s از نوع char و luxmeter از نوع float را جهت چاپ مقدار روشنایی در برنامه تعریف می‌کنیم. همچنین یک تابع به نام readLux جهت خواندن میزان روشنایی ایجاد خواهیم کرد.



CodeVisionAVR - F:\microele\GY30LUX\Project\GY30LUX\GY30.prj

File Edit Search View Project Tools Settings Help

Code Navigator

CodeVisionAVR
Project: GY30
Notes
main.c
Headers
List Files
GY30.asm
GY30.lst
GY30.map
Other Files

F:\microele\GY30LUX\Project\GY30LUX\main.c

```
35  
36 // Declare your global variables here  
37 char s[30];  
38 float luxmeter;  
39  
40 float readLux (void)  
41 {  
42     unsigned int luxh, luxl;  
43     int mylux;  
44     float lux;  
45     i2c_start();  
46     i2c_write(ADDW);  
47     i2c_write(0x10);  
48     delay_ms(180);  
49     i2c_stop();  
50     i2c_start();  
51     i2c_write(ADDR);  
52     luxh=i2c_read(1);
```

Messages

Errors Warnings

57:15 Modified Insert Code Information may be incomplete, as the file was not Compiled yet

ایجاد تابع `readLux` جهت خواندن شدت روشنایی از سنسور

بدنه کامل تابع `readLux` در زیر آورده شده است:

```
char s[30];  
float luxmeter;  
  
float readLux (void)  
{  
    unsigned int luxh, luxl;
```



```
int mylux;  
float lux;  
i2c_start();  
i2c_write(ADDW);  
i2c_write(0x10);  
delay_ms(180);  
i2c_stop();  
i2c_start();  
i2c_write(ADDR);  
luxh=i2c_read(1);  
luxl=i2c_read(0);  
i2c_stop();  
mylux=(luxh<<8)|luxl;  
lux = mylux/1.2;  
return lux;  
}
```

تشریح تابع نوشته شده readLux

از آنجایی که دیتای دریافتی از ماژول دو بایت می‌باشد، بایت اول را در متغیر luxh و بایت دوم را در luxl قرار داده‌ایم. در نهایت luxh را 8 بار به سمت چپ شیفت داده و آن را با luxl OR بیتی کرده و در متغیر myLux قرار داده‌ایم تا دیتای دریافتی درون یک متغیر، یکپارچه شده و 16 بیتی گردد.

با توجه به مثال آورده شده در دیتاشیت، لازم است برای تبدیل عدد دریافتی به واحد lux، عدد به دست آمده بر 1.2 تقسیم شود.

How to calculate when the data High Byte is "10000011" and Low Byte is "10010000"
 $(2^{15} + 2^9 + 2^8 + 2^7 + 2^4) / 1.2 = 28067 [lx]$

نحوه به دست آوردن میزان روشنایی به واحد لوکس (lx)

تا این جای کار، نوشتن تابع جهت خواندن مقدار روشنایی انجام شده است.



تکمیل حلقه while

مانند تصویر زیر، حلقه while را تکمیل می‌کنیم. در برنامه نوشته شده، از طریق تابع readLux مقدار روشنایی از ماژول خوانده شده و به میکرو ارسال می‌گردد.

سپس با دستور sprintf از کتابخانه stdio، این مقادیر را به رشته تبدیل کرده و بر روی LCD چاپ می‌کنیم.

```
CodeVisionAVR - F:\microele\GY30LUX\Project\GY30LUX\GY30.prj
File Edit Search View Project Tools Settings Help

Code Navigator
CodeVisionAVR
Project: GY30
  Notes
  main.c
  Headers
  List Files
  GY30.asm
  GY30.lst
  GY30.map
  Other Files

F:\microele\GY30LUX\Project\GY30LUX\main.c
Notes main.c *
190 lcd_putsf("www.microele.com");
191 lcd_gotoxy(0,1);
192 lcd_putsf("GY30 BH1750 LUX");
193 delay_ms(2000);
194
195 while (1)
196 {
197 // Place your code here
198 luxmeter = readLux();
199 sprintf(s, "Lux:%6.2f(lx)", luxmeter);
200 lcd_gotoxy(0,0);
201 lcd_putsf("          ");
202 lcd_gotoxy(0,0);
203 lcd_puts(s);
204 delay_ms(200);
205
206 }
207 }
```

Messages
Errors Warnings

204:21 Modified Insert Code Information may be incomplete, as the file was not Compiled yet

تکمیل برنامه در حلقه while

```
while (1)
```



```
{  
luxmeter = readLux();  
sprintf(s, "Lux:%6.2f(lx)", luxmeter);  
lcd_gotoxy(0,0);  
lcd_putsf("          ");  
lcd_gotoxy(0,0);  
lcd_puts(s);  
delay_ms(200);  
}
```

* دقت شود، از آن جایی که از `sprintf` جهت تبدیل مقادیر `float` به رشته استفاده شده است، حتما از قسمت `Configure` project در کدویژن، قابلیت `float` برای `sprintf` فعال گردد.



CodeVisionAVR - F:\microele\GY30LUX\Project\GY30LUX\GY30.prj

File Edit Search View **Project** Tools Settings Help

Code Navigator

- CodeVisionAVR
 - Project: GY30
 - Notes
 - main.c
 - Headers
 - List Files
 - GY30.asm
 - GY30.lst
 - GY30.map
 - Other Files

Project Menu:

- Check Syntax
- Compile (F9)
- Build (Shift+F9)
- Build All (Ctrl+F9)
- Stop Compilation
- Clean Up
- Information
- Go to Next Error (F8)
- Go to Previous Error (Ctrl+F8)
- Go to Next Warning
- Go to Previous Warning (Ctrl+F4)
- Notes
- Export to a New CodeVisionAVR Project
- Export to CodeVisionAVR Extension for Atmel Studio
- Configure**

```

200 lcd_gotoxy(0,0);
201 lcd_putsf("                ");
202 lcd_gotoxy(0,0);
203 lcd_puts(s);
204 delay_ms(200);|
205

```

تنظیم قابلیت float در printf

بعد از باز کردن بخش Project Configure. به سربرگ C Compiler رفته و sprint Features را بر روی float, width , precision قرار داده و بر روی OK کلیک کنید.



The screenshot shows the CodeVisionAVR IDE interface. The main window displays a C program for an AVR microcontroller. The 'Configure Project GY30.prj' dialog box is open, with the 'C Compiler' tab selected. The 'Active Build Configuration' is set to 'Debug'. The 'Chip' is 'ATmega32' and the 'Clock' is '8.000000 MHz'. The 'Memory Model' is 'Small'. The 'Optimize for' is 'Size' and the 'Optimization Level' is 'Maximal'. The 'Program Type' is 'Application'. The '(s)printf Features' dropdown menu is open, showing options: 'int, width', 'int', 'int, width', 'int, width, precision', 'long, width', 'float, width, precision'. A purple arrow points to the 'float, width, precision' option. The 'Code Generation' section has 'Bit Variables Size' set to '16'. The 'File Output Formats' are 'COF ROM HEX EEP'. The 'Preprocessor' section has 'Include I/O Registers Bits Definitions' checked.

تنظیم قابلیت float در printf

برنامه را کامپایل کرده و بر روی میکرو پروگرام و اجرا می‌کنیم.



نمایش مقدار روشنایی بر روی LCD کاراکتری با واحد لوکس

همچنین عزیزان می‌توانند سورس کامل نوشته شده برای این مطلب و دیتاشیت قطعه BH1750 را از [این لینک](#) دانلود نمایند.

نحوه عملکرد

نحوه عملکرد این آموزش را در ویدیو زیر مشاهده می‌نمایید.

نتیجه گیری

ماژول لوکس متر GY-30 مجهز به سنسور BH1750 بوده که حساس به شدت نور می‌باشد. این ماژول دارای دقت و رزولوشن بالایی است و داده‌های خروجی آن بر حسب واحد لوکس (lx) می‌باشد. با استفاده از میکروکنترلر AVR و کامپایلر کدویژن Codevision و با مطالعه دیتاشیت BH1750، این سنسور را راه‌اندازی کرده و مقدار روشنایی محیط را بر



حسب واحد لوکس بر روی LCD کاراکتری نمایش دادیم.

امیدوارم از این آموزش کمال بهره را برده باشید. در صورت داشتن هرگونه نظر یا سوال درباره این آموزش 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.