اموزش میکرو کنترولر های XMEGA

با کامپایلر CODEVISION AVR

نویسنده : پوریا علیزاده

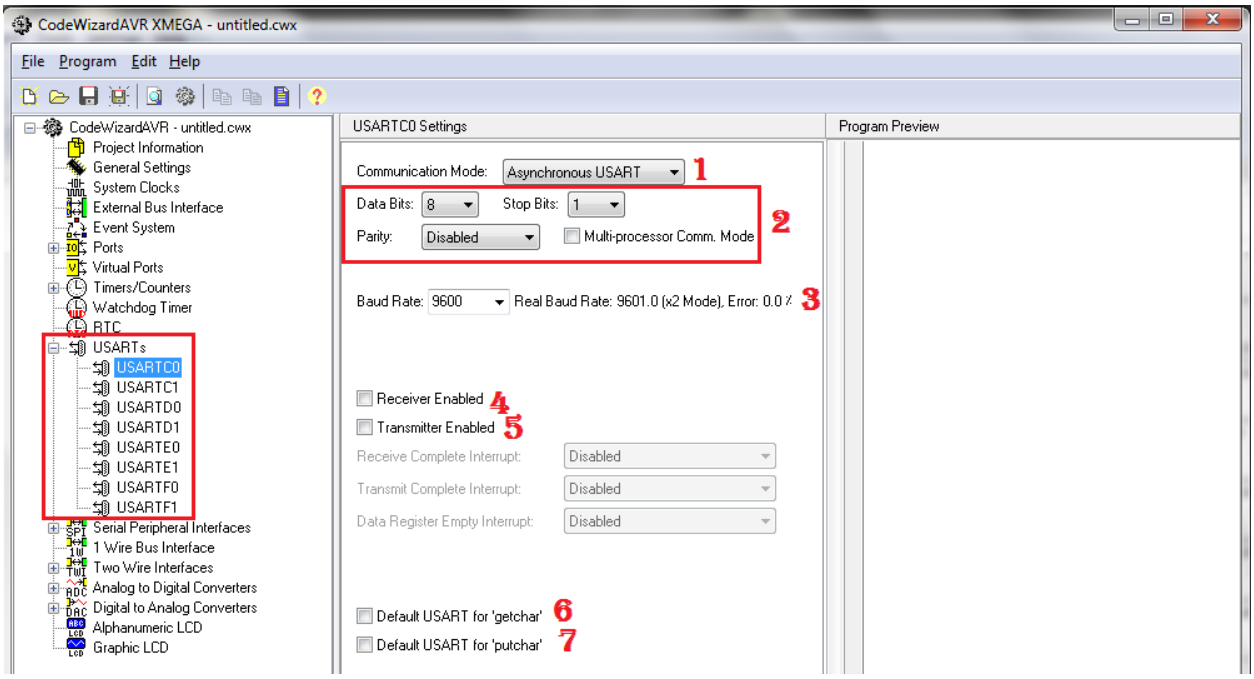
ارتباط USART:

ارتباط سریال USART یکی از پر کاربرد ترین و معروف ترین پروتکلهائی است که توسط اکثر کامپیوتر ها حمایت می شود.و لذا برای برقراری ارتباط بین میکروکنترولر و کامپیوتر اغلب مورد استفاده قرار می گیرد.

در میکرو کنترولر های سری XMEGA بسته به نوع ان ،2 واحد USART در روی چند پورت ان می تواند قرار داشته باشد که مثلا در میکرو ATXMEGA128 A1 در روی هر کدام از پورت های (C,D,E,F) 2 واحد USART موجود می باشد که با نامهای 0و1 مشخص می گردند.مثلا USART موجود روی پورت C : USARTC0,USARTC1

تنظیمات واحد USART در CODEWIZARD:

بعد از انتخاب قسمت USART در CODEWIZARD تنظیمات این قسمت باز می شوند.



بعداز انتخاب USART مورد نظر(درمنوی سمت چپ قسمت CODEWIZARD) می توانید تنظیمات مربوط به ان قسمت را انجام دهید.

1- در این قسمت می توانید نوع عملکرد این قسمت را انتخاب کنید که به ترتیب از بالا به پائین در مد سنکرون،اسنکرون،infared، master spi می تواند عمل کند.

2- در این قسمت می توانید چندین نکته مهم در این ارتباط را مشخص کنید:

Data bit: در این قسمت می توانید تعداد بیتهای اطلاعات ارسالی یا دریافتی داده ها را تنظیم کنید.

Stop bit: یک یا 2 بیت در انتهای اطلاعات ارسال می شود که مشخص کننده پایان داده است.

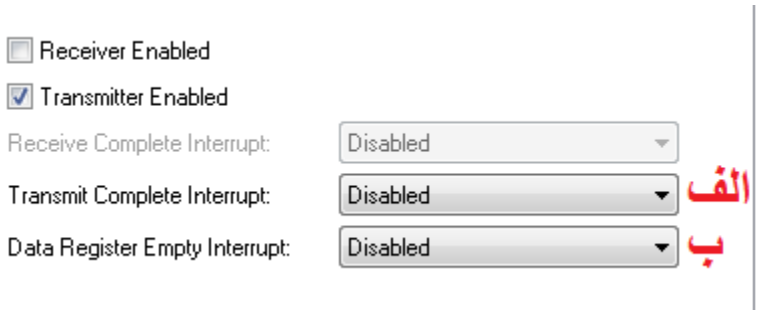
Parity: این بیت در انتهای داده امده و مشخص کننده صحت داده است.

Multi_processor comm:از این مد در ارتباط برقرار کردن چند mcu با هم از طریق چند باس سریال استفاده می شود.که استفاده از این مد بطور موثری فریم های ورودی را کاهش می دهد.در این روش باید از مد 9bit ارسال و دریافت داده استفاده شود.

3- در این قسمت نرخ ارسال دیتا مشخص می شود گه بستگی به فرکانس کاری میکرو دارد.و در هنگام تنظیم، در صد خطا محاسبه شده نیز جلوی این گزینه به نمایش در می اید که نباید بیشتر از 0.2% باشد.

4- اگر این قسمت فعال شود میکرو در مد گیرنده (receiver) عمل میکند.با انتخاب این گزینه قسمت زیر نیز به نمایش در می اید که با فعال کردن ان بعد از دریافت کامل اطلاعات وقفه ایجاد می شود که در این قسمت سطح وقفه مورد نظر تعیین میگردد.

5- اگر این قسمت فعال شود میکرو در مد فرستنده (transmitter) عمل میکند.با انتخاب این گزینه قسمت های زیر به نمایش در می ایند که به ترتیب به این صورت عمل میکنند.



الف- با تنظیم این قسمت اگر دیتا به صورت کامل فرستاده شود وقفه اتفاق می افتد که می توان سطح ان را در این قسمت تعیین کرد.

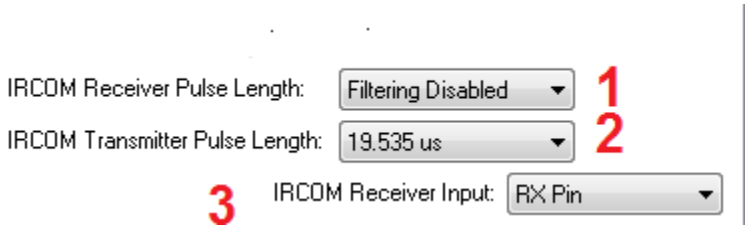
ب- با تنظیم این قسمت اگر رجیستر دیتا خالی باشد وقفه ای اتفاق می افتد که میتوان سطح ان را در این قسمت تعیین کرد.

نکته: اگر هر 2 قسمت receiver enable,transmitter enable فعال باشد میکرو در هر 2 مد فرستنه و گیرنده عمل می کند.

7,6- یکی از usart ها می تواند به عنوان پیش فرض(default) نرم افزار قرار گیرد تا از توابع استانداردputchar,getchar استفاده کند،که در هر کدام از usart ها این 2 گزینه انتخاب شود،انusart به عنوان پیش فرض انتخاب می شود.

مد infrared:

تنظیمات این واحد نیز به جز موارد زیر مانند قسمت بالا می باشد.



1- در این قسمت پهنای پالس infrared در مد گیرنده تنظیم می شود.که باید قسمت receiver enable نیز فعال شده باشد.

2- در این قسمت پهنای پالس infrared در مد فرستنده تنظیم می شود.که باید قسمت transmitter enable نیز فعال شده باشد.

3- در این قسمت ورودی ircom مشخص می شود.که می تواند پایه rx قسمت usart مورد نظر یا یکی از کا نالهای event channel باشد.

** برای دستگا هائی با بیشتر از یک usart قسمت ircom در یک زمان می تواند با یکی از usart های مورد نظر ارتباط برقرار کند.

توابع و دستورات ارسال و دریافت:

تابع void usart_disable(USART_t *pu) :

این تابع usart مورد نظر را disable می کند.

```
usart_disable(&USARTC0);
```

در این مثال usartc0 غیر فعال می شود.

1-تابع void putchar(char c) :

این تابع یک کاراکتر را به کامپیوتر ارسال می کند.

```
putchar('p');
```

در این مثال حرف p ارسال می شود.

2-تابع void putchar_usartmn(char c) :

عملکرد این تابع نیز مانند تابع بالا بوده با این تفاوت که در این تابع کاراکتر مورد نظر از طریق usart تعیین شده فرستاده می شود.

```
putchar_usartc0('p');
```

در این مثال حرف p از طریق usartc0 فرستاده می شود.

تابع putsf(); :

این تابع یک رشته ثابت مثل یک جمله را ارسال می کند.

```
putsf("POORIA");
```

در این مثال کلمه pooria ارسال می شود.

تابع puts(); :

این تابع یک رشته ارایه ای مثل رشته هائی که با تابع sprint(); ساخته شده را ارسال می کند.

```
Sprint(str, "temp1:%u"temp1);

Puts(str);
```

تابع printf(); :

این تابع می تواند هم به عنوان putsf(); عمل کند و هم مقدار متغییر های قرار داده شده در ان را ارسال کند.

Printf("pooria");

Printf("temp1:%u"temp1);

تابع (char getchar(void :

این تابع کاراکتر مورد نظر را دریافت کرده و داخل متغیر تعیین شده قرار می دهد.و در هر خط از برنامه قرار گیرد برنامه را در ان خط متوقف می سازد تا یک کاراکتر دریافت شود.

a=getchar();

در این مثال کاراکتردریافت شده داخل متغییر a قرار داده می شود.

تابع (char getchar_usartmn(void :

عملکرد این تابع نیز مانند تابع بالا بوده با این تفاوت که در این تابع کاراکتر مورد نظر از طریق usart تعیین شده دریافت می شود.

a=getchar_usartc0();

در این مثال حرف p از طریق usartc0 دریافت شده و داخل متغیر a قرار می گیرد.

تابع ;(scanf :

این تابع با قالبی مشخص متغییری را دریافت می کند و تا دریافت یک مقدار متوقف می شود.

Scanf("%u",&n);

در این مثال مقدار دریافت شده در متغییر n قرار داده می شود.

تابع ;(gets :

این تابع یک مقدار را درون یک متغییر قرار می دهد.

Char str[4];

Gets(str,4);

در این مثال قرار است عدد 2.48 در یافت شود،که پس از دریافت داخل ارایه str قرار می گیرد.

توجه 1:در هنگام استفاده از مد سریال، پایه (TX) ،پایه ارسال دیتا باید به صورت خروجی تعریف شود و پایه (RX)،پایه دریافت دیتا به صورت ورودی تعریف شود.

توجه 2: برای وصل میکرو به کامپیوتر به علت عدم تطبیق سطوح ولتاژ میکرو و کامپیوتر از تراشه های واسط استفاده کنید،که به دلیل اینکه ولتاژ کاری میکروکنترولر های xmega تا 3.6v می باشد میتوانید از تراشه هائی مانند max3232 که محدوده کاری ان 3v-5v میباشد و همچنین از مبدلهای usb به سریال مانند ft232rl استفاده نمائید.

مثال: برنامه ای بنویسید که دو میکرو متصل به هم از طریق خطوط ارتباط سریال(rx,tx) که به میکرو اول سنسور دما ds1820 متصل می باشد،دما را خوانده و به میکرو دوم ارسال کند تا میکرو دوم دما را روی lcd به نمایش در اورد.

برنامه میکرو اول:

```
float p;
while (1)
    {
    p= ds18b20_temperature(0);
     putchar_usartc0(p);
        }
```

برنامه میکرو دوم:

```
char p1[8];
float p;
while (1)
    {
    lcd_clear();
    lcd_gotoxy(3,0);
    p=getchar_usartc0();
    itoa(p,p1);
    lcd_putsf("TEMP IS:");
    lcd_puts(p1);
    delay_ms(200);
   }
```

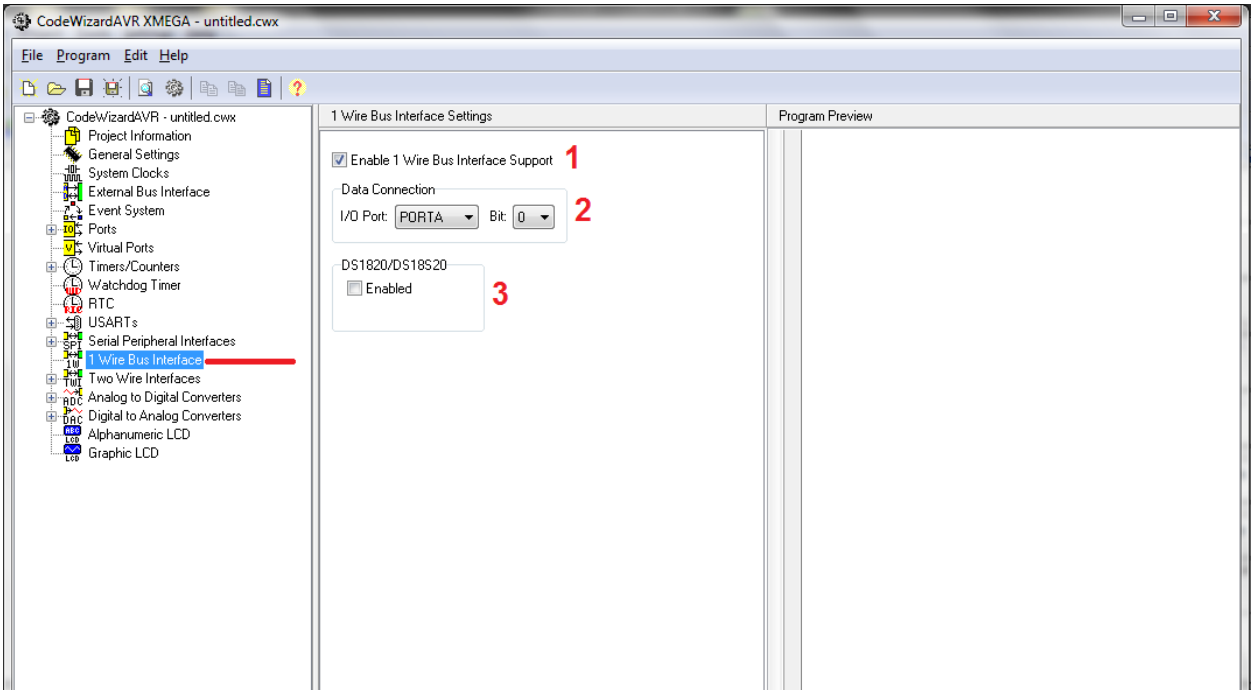
پروتکل 1wire :

پروتکل 1wire نوعی پروتکل ارتباطی است که در آن بین میکروکنترلر و وسیله جانبی توسط یک سیم ارتباط برقرار می شود و ارسال و دریافت اطلاعات از طریق تنها یک سیم انجام می پذیرد.امروزه اکثر قطعات از این روش ارتباطی استفاده می کنند مانند انواع سنسورها مثل Ds18x20

ویژگیهای این سیستم :

- دارای هزینه کم
- کم ترین سرعت انتقال اطلاعات بالای 16 Kbps
- فاصله مناسب بین master و slave توسط کابل بیشتر از 300 متر می تواند باشد.
- ارسال اطلاعات به صورت small package است.

تنظیمات قسمت 1wire:



1- با انتخاب این گزینه تنظیمات این قسمت فعال می شوند.

2- در این قسمت پورت و پایه (بیت) مورد نظر جهت اتصال به وسیله مورد نظر انتخاب می شود.

3- اگر قصد استفاده از سنسورهای ds1820/ds18s20 داریم با انتخاب این گزینه می توانیم از کتابخانه اماده نوشته شده برای این سنسور ها استفده کنیم.

Multiple devices:اگر بیش از یک سنسور ds1820 به باس 1wire متصل باشد،این گزینه را باید فعال کنیم و حداکثر تعداد سنسور مورد نظر حداکثر می تواند 8 عدد باشد.کد شناسائی (rom) مربوط به هر سنسور داخل ارایه ای به نام ds1820_rom_codes ذخیره می شود.

وسیله متصل شده به باس 1WIRE باید PULL UP شود با یک مقاومت 4.7K

توابع 1WIRE:

تابع unsigned char w1_init(void) :

این تابع تنظیمات اولیه مربوط به باس 1WIRE را انجام می دهد.اگر مقدار برگشتی تابع 1 باشد وسیله ای وجود دارد.واگر مقدار برگشتی صفر باشد وسیله ای وجود ندارد.

تابع unsigned char w1_read(void) :

این تابع یک بایت را از روی باس 1WIRE می خواند.

```
temp=w1_read();
```

در این مثال دیتا مورد نظر از روی باس 1wire خوانده و داخل متغییر temp قرار می دهد.

تابع unsigned char w1_write(unsigned char data) :

این تابع دیتا موردنظر (یک بایت) را روی باس 1WIRE قرار می دهد.

و اگر مقدار برگشتی 1 باشد دیتا مورد نظر بصورت کامل روی باس قرار داده شده، در غیر این صورت مقدار برگشتی 0 است.

```
w1_write(0xcc);
```

در این مثال دیتا 0xcc روی باس 1wire قرار داده می شود.

تابع unsigned char w1_search(unsigned char cmd,void *p) :

این تابع تعداد وسائل متصل شده به باس 1WIRE را بر می گرداند.اگر وسیله ای پیدا نشود مقدار برگشتی تابع صفر است. بجای cmd کد شناسائی(rom) وسیله مورد نظر و اشاره گر p به ناحیه ای از فضای sram اشاره می کند که در انجا کد شناسائی(rom) برگشتی از وسیله مورد نظر ذخیره شده است(که برای هر تعداد وسیله میتواند متفاوت باشد).

```
devices=w1_search(0xf0,rom_codes);
```

مثال:برنامه ای بنویسید که دمای محیط را توسط سنسور ds18b20 خوانده و روری lcd نشان دهد.(سنسوردما به PORTA.3 متصل بوده وlcd به PORTJ)

(این سورس با استفاده از کتابخانه ds18b20 نوشته شده است)

```
char buffer[32];
float temp;
while(1){
        temp=ds18b20_temperature(0);
        sprintf(buffer,"T=%5.3f\xdfC",temp);
        lcd_clear();
        lcd_puts(buffer);
        delay_ms(300);
   }
```

پایان قسمت سوم

منابع:

1–**مهندس اوژن کی نژاد**

2–**سایت ATMEL**

3–**سایت AVRFREAKS**

4–حمید نجفی

P-A