برای دانلود مطالب آموزشی بیشتر به وبسایت Melec.ir مراجعه کنید.

بنام خدا

ارتباط متلب با میکروکنترلر AVR

نویسنده : حمیدرضا بایرامی

سایت ناشر : www.Melec.ir

خلاصه پروژه :

ارتباط matlab با میکروکنترلر (Atmega 16,32)

در این پروژه میخواهیم دو تا سنسور دما یا هر سنسور دیگه ای رو به میکروکنترلر وصل کنیم و از و از طریق میکروکنترلر مقدار این دو تا سنسور رو بخوانیم و بعد از یک سری محاسبات مربوط به پارامترهای سنسور آنرا در نمایشگرlcd چاپ کنیم و همین مقدار چاپ شده رو به GUI "متلب" بفرستیم تا به صورت نرم افزاری این مقدار را به چشم خود ببینیم یا مانورهای لازمه رو انجام بدیم.

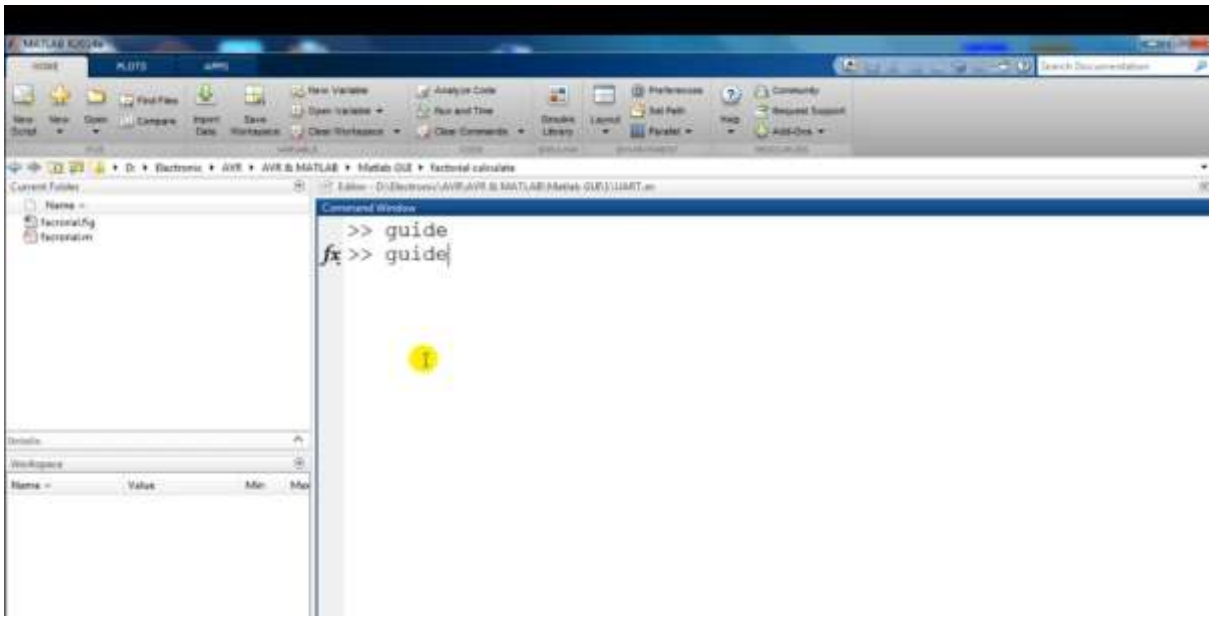
GUI(graphical user interface)همانطور که از اسم این ابزار مشخص است میتوانیم با این ابزار به راحتی تابع ها و مانورهای لازمه رو در خود "متلب " یعنی در common window یا در Mfile این نرم افزار این مانورها را انجام داد و با چند سری از کلیدها و نمایشگرها تسلط کاملی رو از طریق این GUIبه "متلب " داشته باشیم .از شکل زیر دو تا کلید برای میکرو قرار دادم و وظیفه ی یکی از کلید ها این است که :اگر ما این کلید را فشار دهیم مشخصات میکرو کنترلر را بر روی LCDچاپ خواهد کرد و همین مشخصات رو از طریق پورت یوزارت به طرف مقابل خود میفرستد تا طرف مقابل هم از این تنظیمات خبر دار شود این تنظیمات یوزارت میتواند از جمله baud Rate , Data Bit , Parity , و غیره باشد.

و کلید دومی رو گذاشتیم که با فشردن این کلید از میکروکنترلر درخواست میکنیم که مقدار سنسورها رو قراعت کند و روی LCDچاپ کند و به طرف مقابل بفرستد تا طرف مقابل هم از این مقدار سنسورها اطلاع حاصل کند.

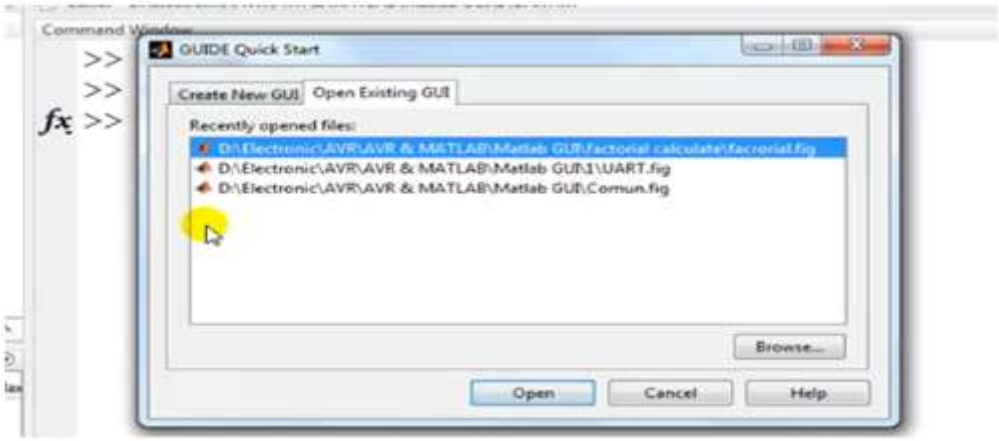
اگر میکروکنترلر ما در فاصله ای دوری از GUI داشته باشد نیازی به دوتا کلید گفته شده فوق نیست چون من از طریق خود GUIمیتوانم به میکرو درخواست بدم تا مقدار سنسورها را قراعت کند و به متلب برگرداند .پس ناگفته نماند که متلب یکی از پرکاربردترین نرم افزارها دنیاست و میتوانیم پردازش لازم را بر روی پارامترهای خودمان داشته باشیم یعنی در این پروژه ما از پروتکل یوزارت متلب استفاده کردیم و میتوانستیم از SPI , TWI, I2C نیز استفاده کنیم.متلب برای این پروتکل ها دستورات استاندارد زبان C را دارد یعنی نیازی به یادگیری دستورات جدید نیست و دستورات ارسال و دریافت از یوزارت یا دیگر پروتکل ها دقیقا مثل زبان Cمیباشد

در ابتدا به اموزش نرم افزار متلب که در این پروژه استفاده کردیم می پردازیم :

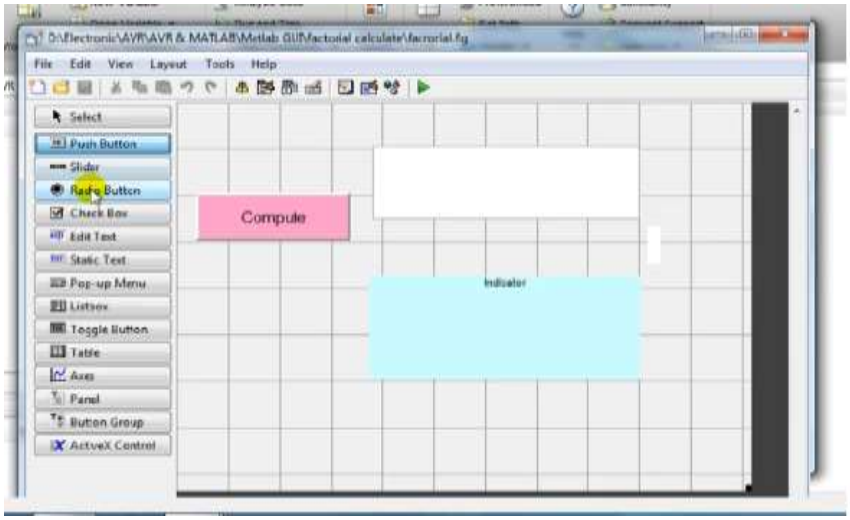
در نرم افزار متلب برای سهولت کار به جای استفاده از پنجره ی command window از یک TOL BOX در ابتدا با نوشتن GUIDE مانند شکل زیر



سپس با زدن دکمه اینتر وارد صفحه زیر می شویم

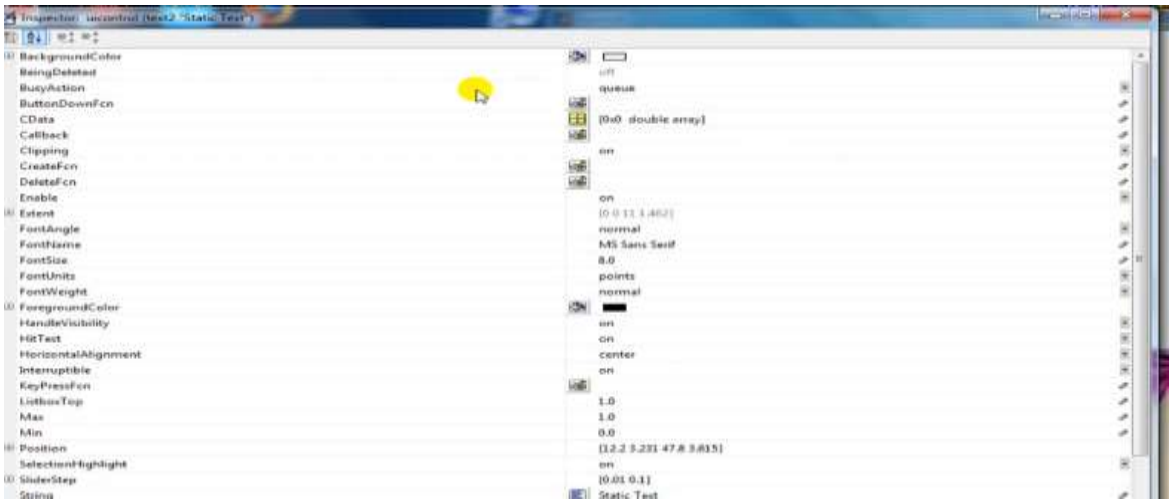


در مرحله بعد با زدن CREAT NEW GUI وارد صفحه ی جدیدی شکل زیر می شویم

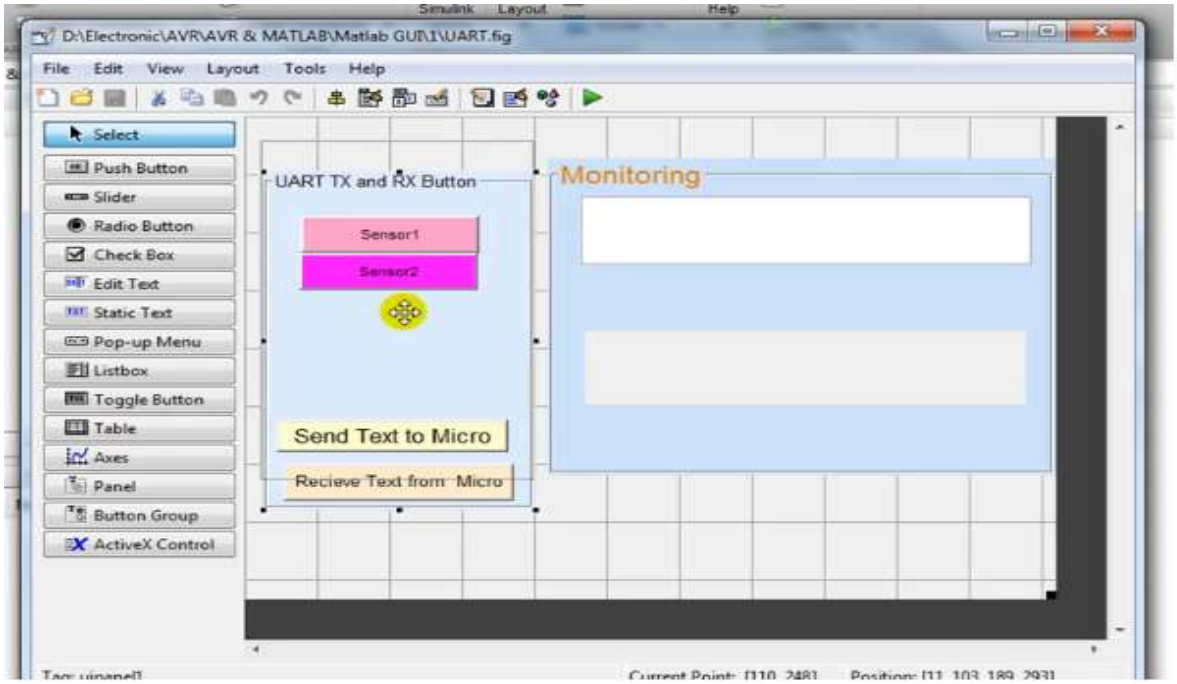


در اینجا قبل اینکه به ادامه متلب بپردازیم لازم است با معنی GUI اشنا شویم GUI مخفف (graphical user interface) یعنی صفحه ای گرافیکی تا کار ما را اسان کند

در box اول به رنگ سفید اطلاعات را از ان می گیریم EDIT TEXT نام دارد BOX قرمز رنگ میتواند کلید یا پوش باتن باشد اما BOX اخر ابی کم رنگ خروجی نمایشگر ماست ،برای تغییر رنگ و تنظیمات دیگر آن کافی است دوبار کلیک کنیم بر روی BOX مورد نظر و وارد صفحه ی زیر می شویم

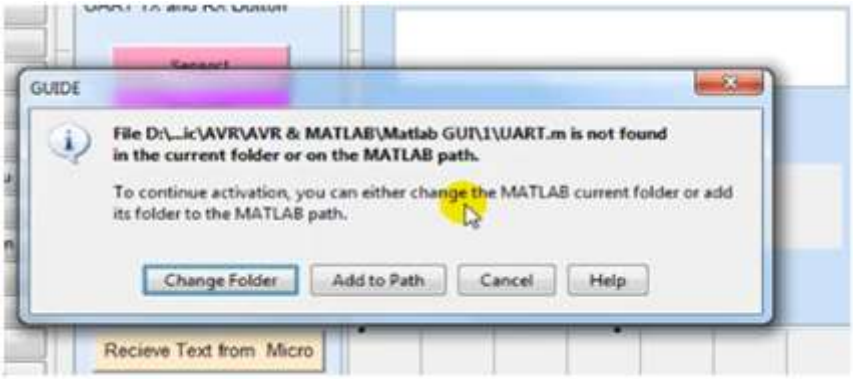


سپس با RUN کردن کلید سبز رنگ وارد صفحه ی زیر می شویم



در اینجا با ساخت Panel، UART TX می توان BOX ها رو در داخل ان قرار داد که می توان انها رو جابه جا کرد و Panel دیگر به اسم MONITORING

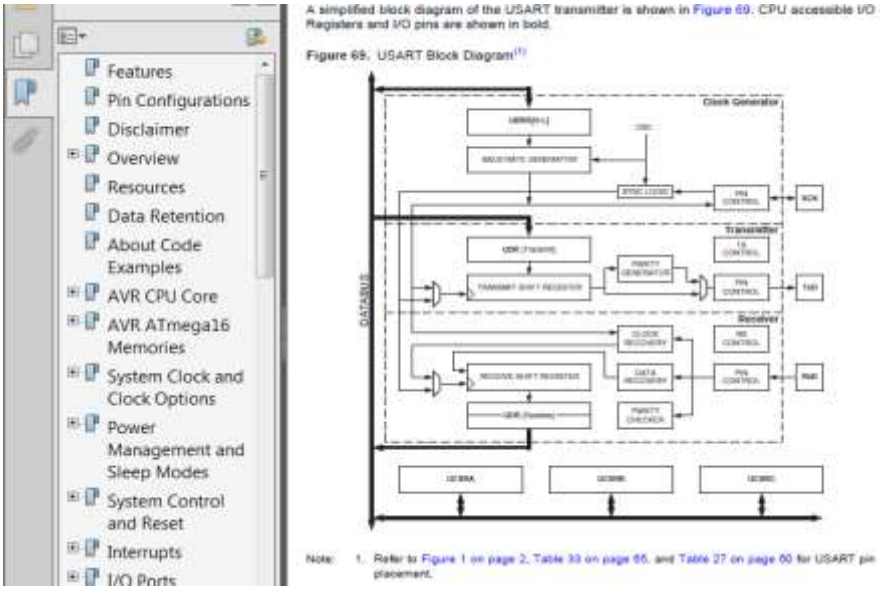
در پروژه از دو سنسور استفاده شده یعنی اگر سنسور شماره یک را فشار دهیم از میکرو میخواهد اطلاعاتش رو به متلب بفرستد و سنسور شماره دو نیز اطلاعاتش رو به میکرو می فرستد بین اینها یک سری رابط وجود دارد که زبان هم دیگه رو میخوانند با فشردن هر کدام از این کلید ها در قسمت مانیتورینگ منظور BOX پایین نشان می دهد ، اگر مرحله بعد با RUN کردن پیغامی رو می دهد که در کجا ذخیره کنیم



پس در نتیجه GUI کار ما را اسان تا با ابزار مفید و مختصری سرو کار داشته باشیم .

دراینجا با ساخت GUI بیشتر آشنا خواهیم شد و ارتباط ان با کلید ها و پوش باتن ها و فرستادن یک سری اطلا عات از طریق پروتکل UART

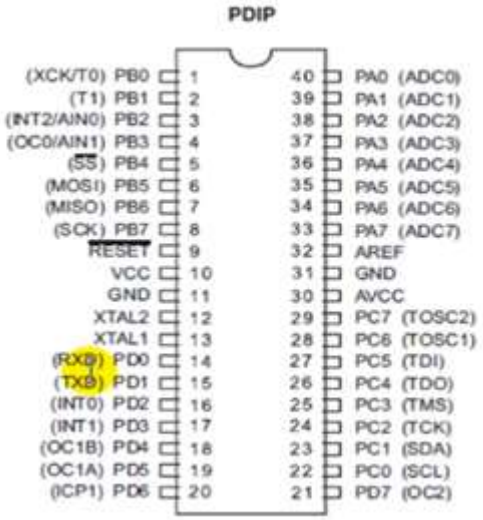
در شکل زیر با دیتا شیت AT MEGAH16 رو مشاهده می کنید .



در اینجا 3 پایه به نام های فرستنده ،گیرنده و کریستال وجود دارد .

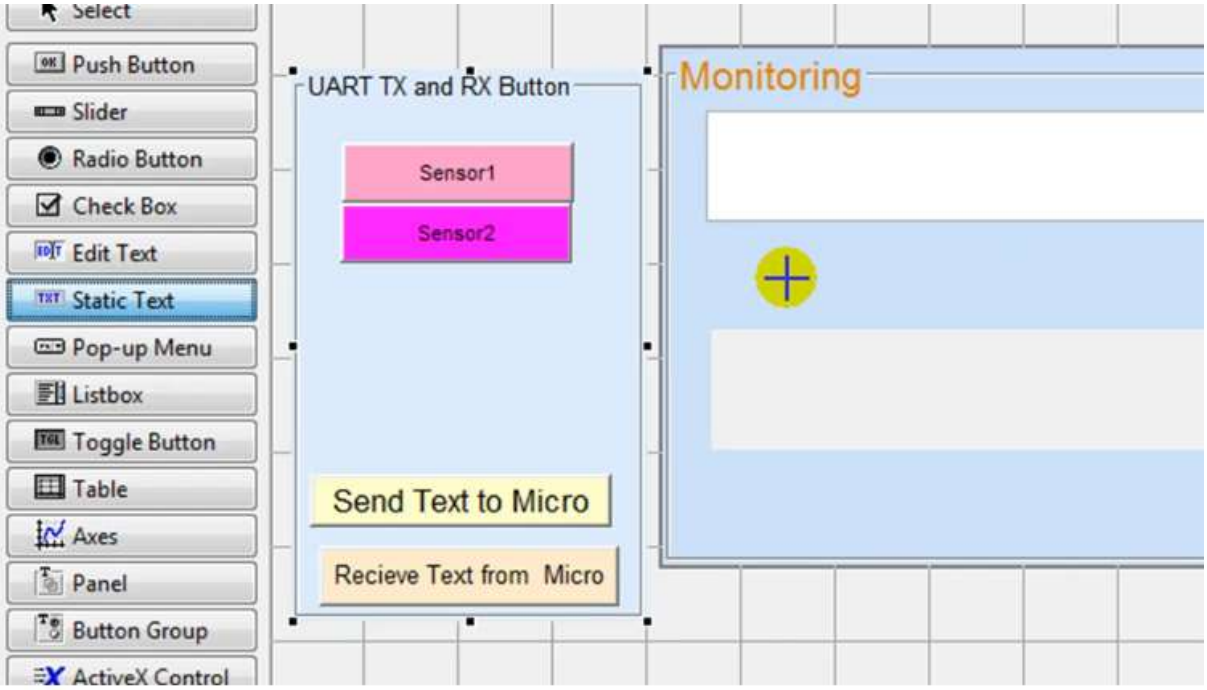
در میکرو کنترلرزیر می توان پایه ها رو مشاهده کرد .

Figure 1. Pinout ATmega16



در شکل بالا که با رنگ زرد مشخص شده فرستنده و گیرنده و پایه PB0 پایه کریستال ماست .نکته اگر به صورت سنکرون استفاده کنیم نیاز به این پایه است ،اما اگر به صورت

اسنکرون نیاز به این پایه نیست.برای یاداوری لازم است ،شکل زیررا مرور کنیم

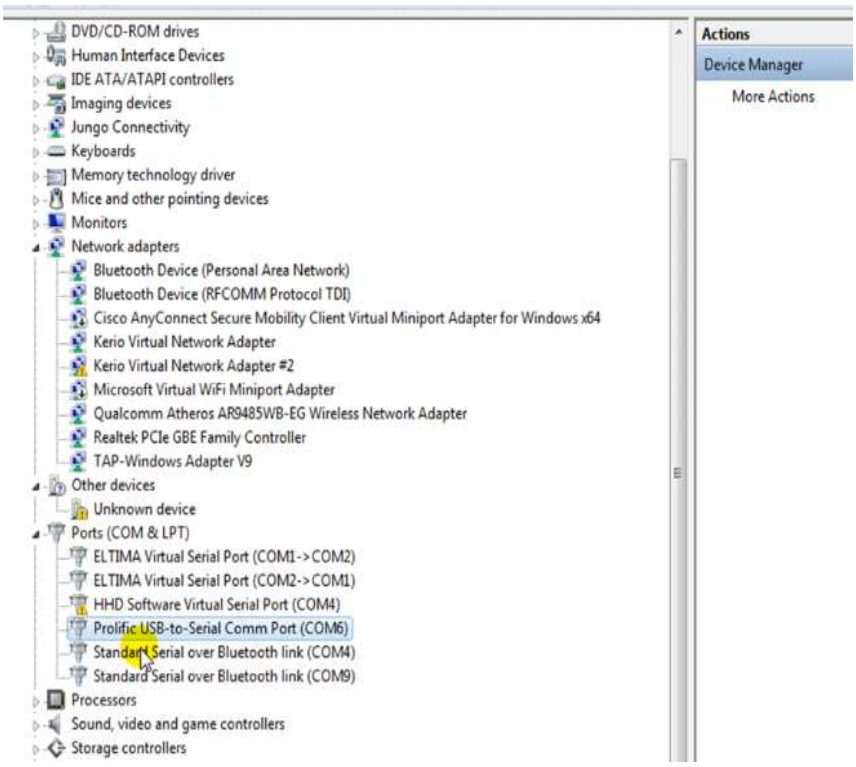


همان طور که قبلا نیز گفته شده بود ،چهار BOX که در داخل PANEL قرار دارند PUSH BUTTON هستند .و در Panelقسمت monitoring کادر اول ان Edit text و کادر پایینی آن static text نام دارد .

با راست کلیک کردن بر روی سنسور 1 گزینه view call back و گزینه call back را کلیک کرده وارد قسمت Editor در نرم افزار متلب می شویم ،که در اینجا ما رجوع می کنیم به تابع همین کلید در متلب .

در ادامه به بررسی قسمت برنامه ی متلب می پردازیم :

در اینجا سریال com6 می باشد .به این علت که سریال به USB در COM6 است ،به این علت به پورت 6 وصل می کنیم

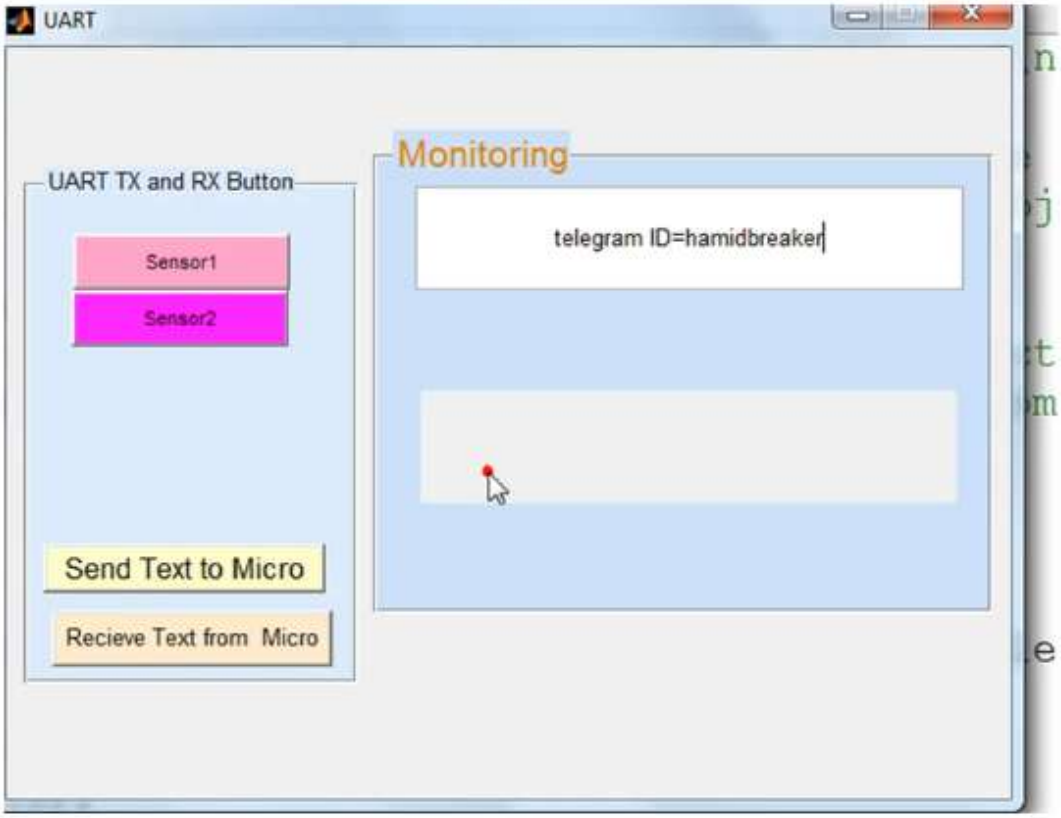


سریال را COM6 ،و 9600 BAUDRATE،که این عدد ،عدد پیش فرض ما می باشد ،و کل تابع را می نامیم .

FOPEN(S) اومده در اینجا پروتکل را باز کرده

در خط پایین fprint(s) کد استاندار زبان C است ،یعنی اگر بخواهیم از طریق پروتکل USART بفرستیم یعنی در USART بنویسیم از طریق fprint میفرستد ،کارکترh را از طریق S بفرستد .جوابی که از طریق میکرو برگرده

به سمت متلب برگرده از طریق fscanf هر چی که وارد شد را اسکن می کنه و کارکتر f یعنی اعداد اشاری را نیز نشان می دهد ،که به عنوان خروجی معرفی شده ،برای نشان دادن خروجی در باکس پایین قسمت monitoring مطابق شکل زیر

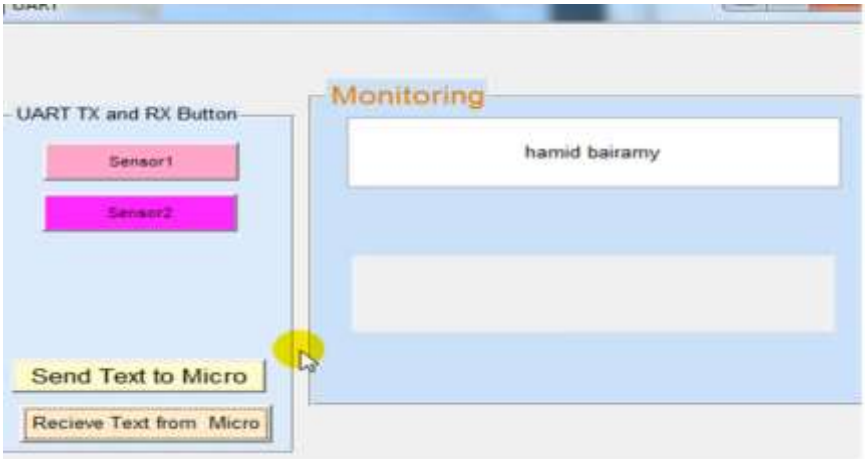


اسم باکس در اینجا با توجه به تنظیمات آن indicate نام دارد یا همان static box نام دارد.

```
Editor - D:\Electronic\AVR\AVR & MATLAB\Matlab GUI\1\UART.m*
UART.m*
85      %%fwrite(s,13);     %new line command (13) or '\n' ins
86      %//////This Section is for Indicating///
87 -    out = fscanf(s,'%f')          %The default value is 10
88      %getdata = fscanf(s,'%c',20);    % A = fscanf(obj,'for
89
90 -    set(handles.indicate,'string',out);
91      %fclose(s);        %Disconnect serial port object from
92 -    delete(s);        %Remove serial port object from memo
93
94
95
96      % --- Executes on button press in RXD.
97      function RXD_Callback(hObject, eventdata, handles)
98
99 -    s=serial('COM6','BaudRate',9600);
00 -    fopen(s);
```

برای نوشتن مطلبی در indicate با توجه به شکل بالا در قسمت set کن handles رو روی indicate باکس به صورت string با این فرمت را به عنوان خروجی .پس این اطلاعات در داخل باکس indicate قرار می گیرد.در خط پایین set برای بستن پروتکل است.

به عنوان مثال اگر ما در باکس اول قسمت monitoring کلمه ای را یاداشت کنیم ،می توانیم اطلا عاتی رو توسط sent text to micro می توان ارسال کرد.و اگر اطلا عاتی از طرف مقابل بیاید توسط Receive text from micro گرفته می شود ،و در بکس دوم قسمت monitoring نشان می دهد که در شکل زیر مشاهده می کنید.



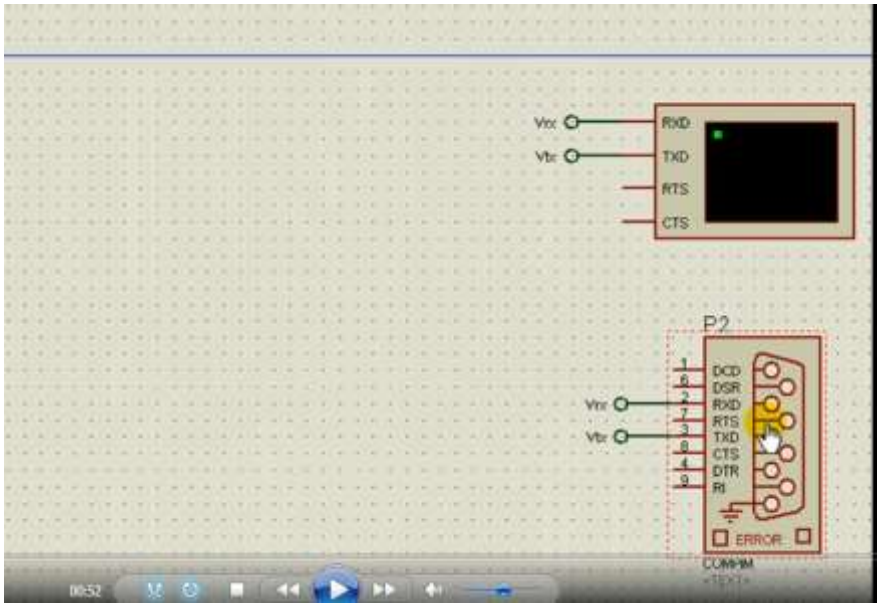
اگر بخواهیم به قسمت مربوط به هر باکس در متلب بشویم با کلیک راست بر روی باکس مربوط و زدن گزینه call back وارد قسمت مربوط به ان باکس می شویم .

نکته :برای اینکه اطلاعات ما پشت سر هم نباشه با زدن کلید اینتر ،یا برای فرستادن تک دستور مثلا در زبان c کلید اینتر همان 13 می باشد .و کد Sهمان پروتکل UART می باشد.

```
95 -        set(hObject,'backgroundcolor','white');
96 -    end
97
98
99      % --- Executes on button press in TXD.
100     function TXD_Callback(hObject, eventdata, handle
101 -   tx=get(handles.Input,'string'); %tx=hamid bairam
102 -   s=serial('COM6','BaudRate',9600);
103 -   fopen(s);        %opens UART protocol anfter setti
104 -   fprintf(s,tx);
105     %fwrite(s,13);     %f code
106 -   fclose(s);        %Disconnect serial port object
107 -   delete(s);
108
```

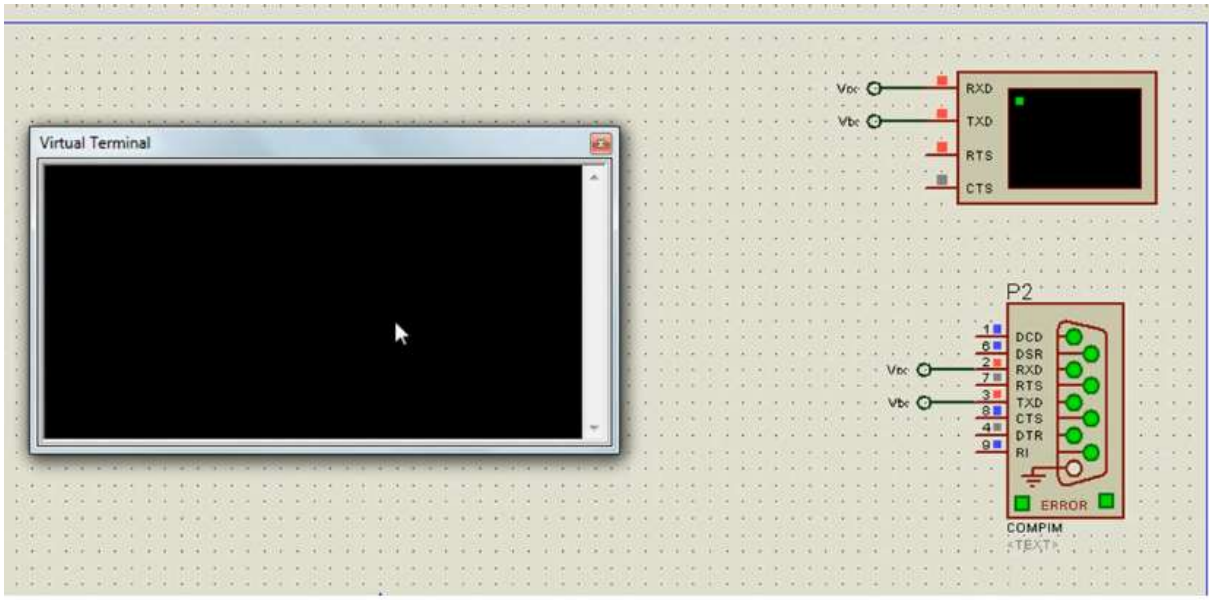
در ادامه به بررسی درست کار کردن GUI متلب در پروتئوس:

قبل از بستن کامل مدار از قسمت Virtual INSTRUMENT MODE رفته و VIRTUAL TERMINAL را انتخاب کرده و در قسمت کتاب خانه COMPIMرا انتخاب می کنیم. مانند شکل زیر

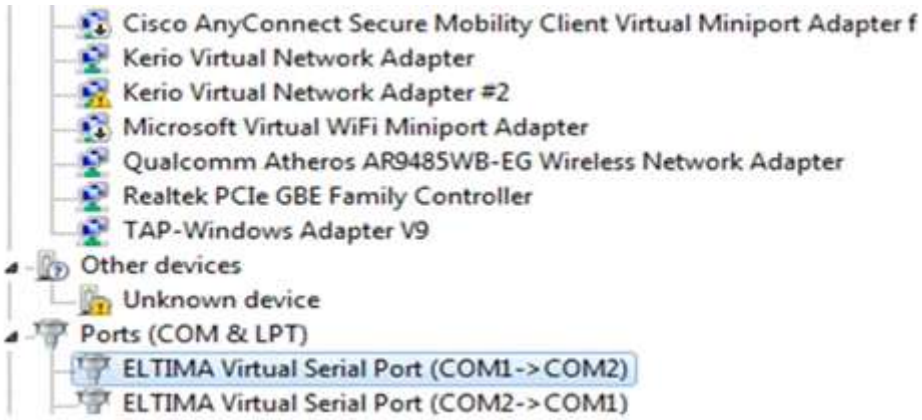


همان طور که در شکل ملاحضه می کنید ،هم COMPIM و virtual terminal هر دو دارای فرستنده و گیرنده می باشند ،که پایه های RXDو TXD هستند،که از داخل بهم وصل هستند.در مرحله بعد با RUN کردن پروتئوس صفحه ای مشکی رنگی که همان

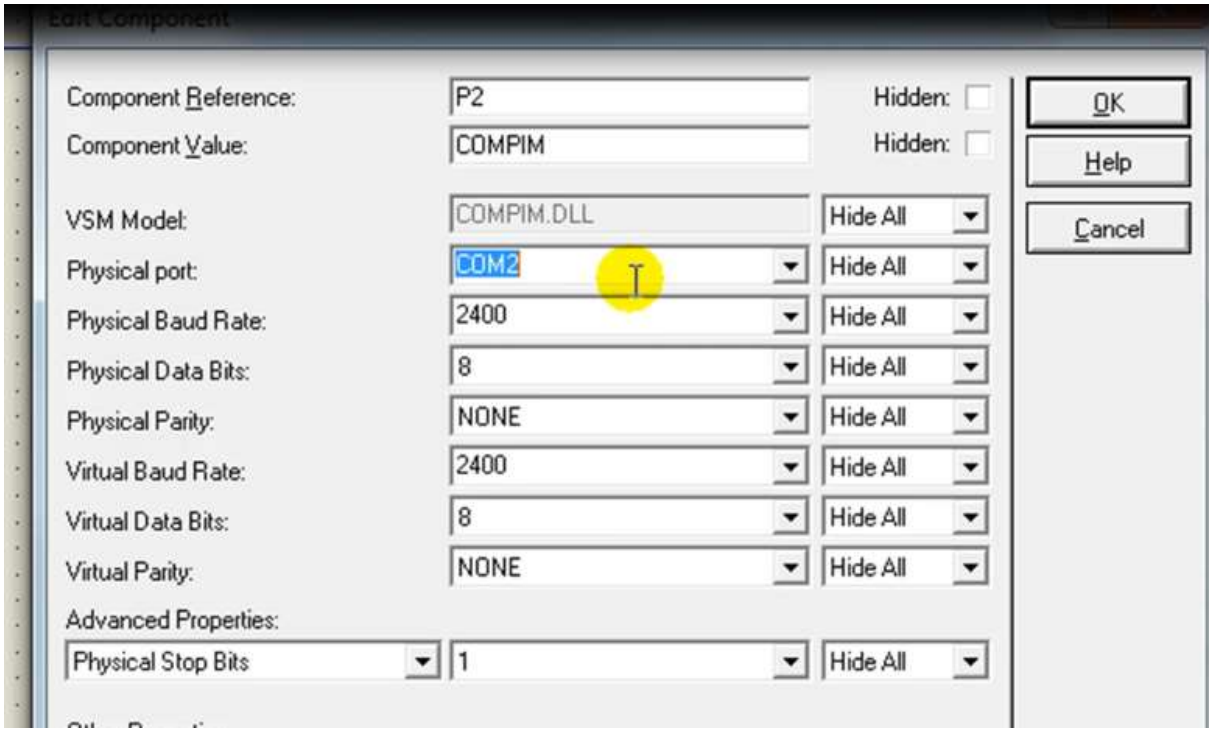
Virtual Terminal نام دارد ،باز می شود که همین میکرو کنترلر ماست.



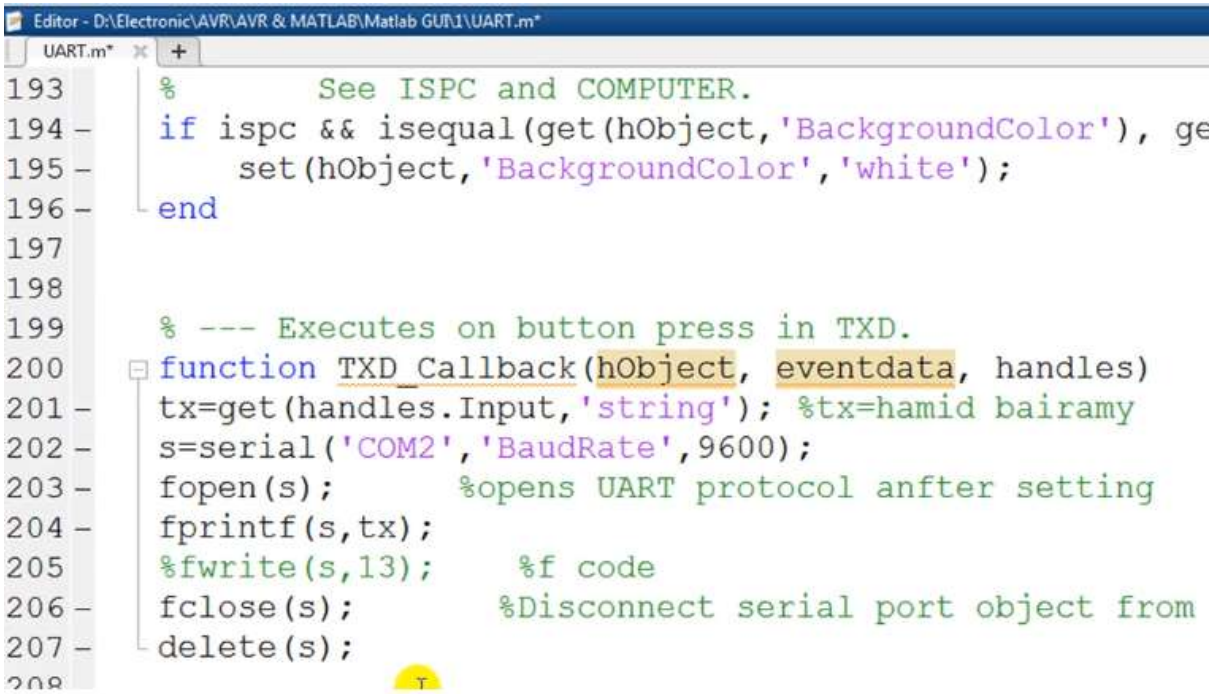
برای اینکه پایه های معرفی شده ،به وصل باشند ،باید ادرس پورت آنها یکی باشد ،برای این کار به تنظیمات پورت virtual serial port رفته



و با کلیک کردن بر روی COMPIM می توان پورت انها را مشخص کرد

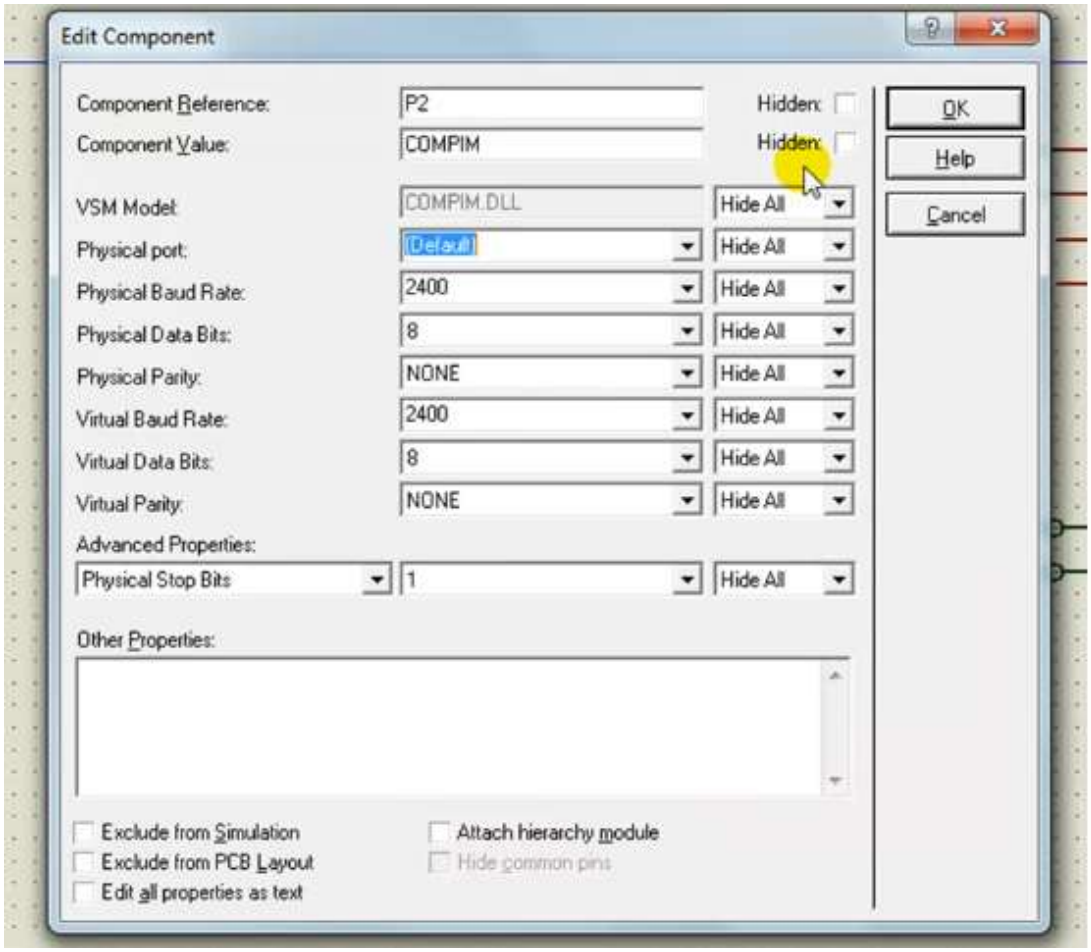


و در مرحله بعد با رفتن به قسمت متلب و با زدن گزینه COMBACK می توان COM سریال را عوض کرد

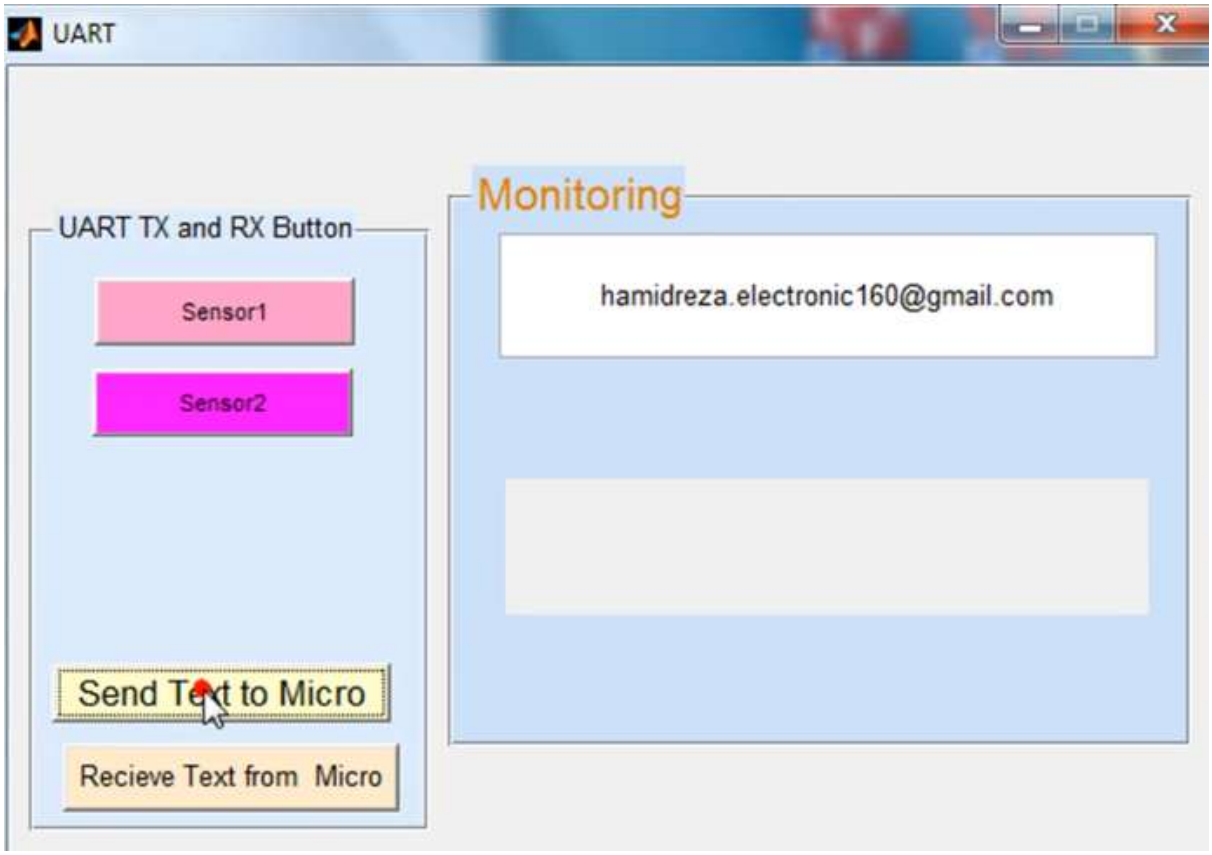


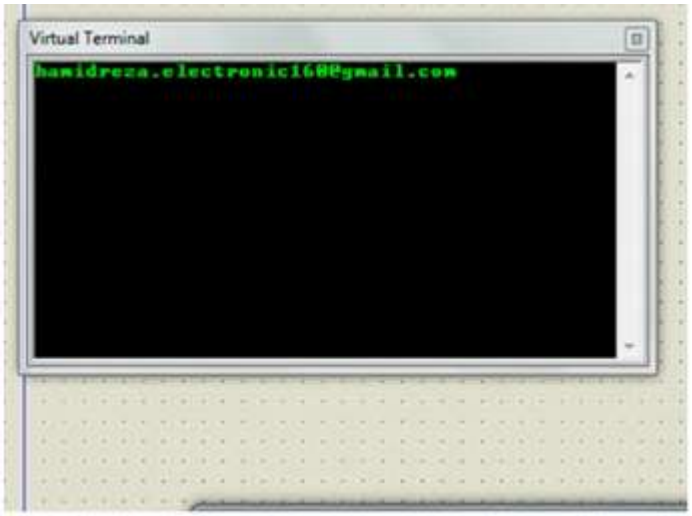
پس تمامی قسمت تابع GUI سریال را COM2 تنظیم می کنیم .

و BAUDRATE تمام انها را بر روی 9600 می گذاریم .که در شکل زیر مشاهده می کنید.



مثال:در این مثال در UART در باکس اول با نوشتن جمله ای و با زدن کلید SEND TO TEXT MICRO اطلاعات از متلب به میکرودر صفحه ی پروتئوس انتقال داده می شود.





یعنی نوشته ی ما ازطریق fprintf(s,tx) نوشته ی ما در قسمت tx نوشته می شود ،و از طریق سریال s فرستاده می شود. fprint f در زبان c یعنی فرستادن است .

با زدن کلید 1 یا همان سنسور 1 کاراکتر 1 ،همان حرف h را در صفحه ظاهر می کند .و با فشردن سنسور 2 ،حرف f را می فرستد.که نشان از صحت برنامه دارد.برای درستی قسمت Receive text from micro در مطلب که به عنوان گیرنده می باشد ، با زدن کلید Recive text from جمله را در قسمت میکرو صفحه ی

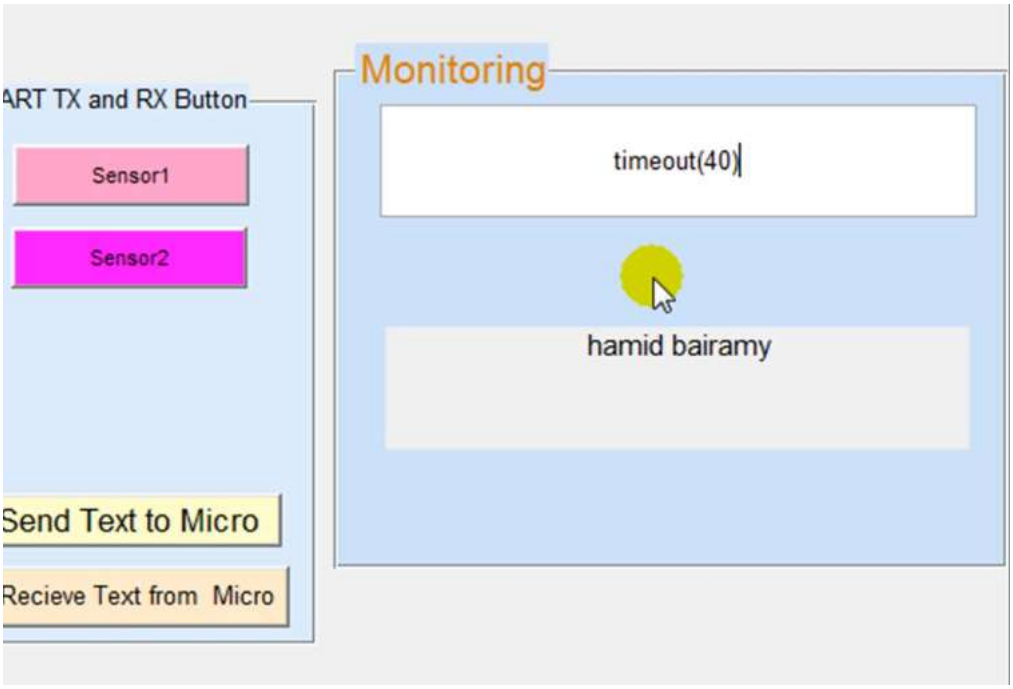
پروتئوس وبعد از چند ثانیه جمله به قسمت متلب در باکس دوم یا ststic textچاپ می کند.

خط readasync(s) در متلب UART را به صورت اسنکرون می خونه ودر بافر خود قرار می دهد .

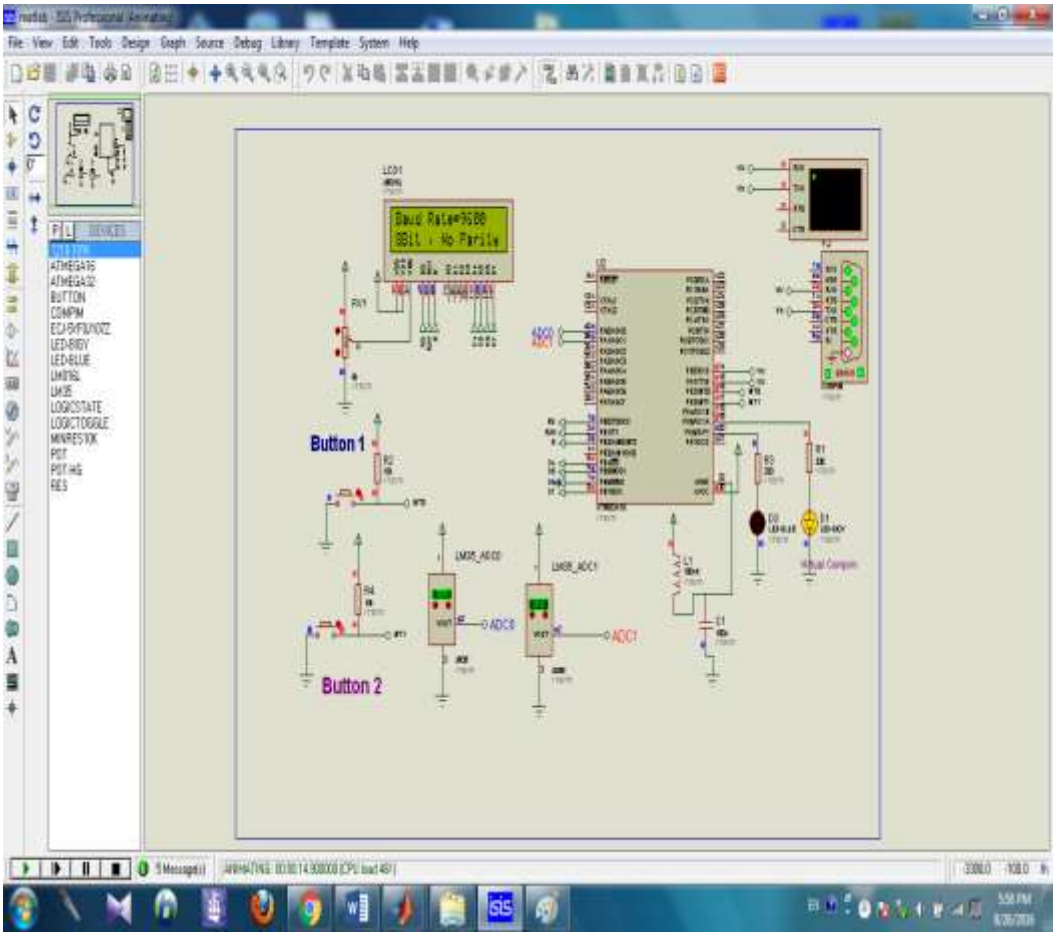
```
Editor - D:\Electronic\AVR\AVR & MATLAB\Matlab GUI\1\UART.m
UART.m
97      function RXD_Callba
98
99 -    s=serial('COM2','Ba
100 -   fopen(s);
101 -   readasync(s);  %rea
102     %pause(3) %executic
103
104 -   out = fscanf(s)
```

و بعد از ان قرار می دهد ،در متغیر OUT که اسکن می کند .

نکته مهم:با نوشتن هر نوشته ای در قسمت میکرو اطلاعات به قسمت STATIC BOX منتقل نمی شود.به دلیل اینکه متلب به صورت پیش فرض یه تایم اوتی را تعریف کرده ،که ما 10 ثانیه وقت داریم تا اطلا عات را از پورت یوزارت بگیرید ،در غیر اینصورت پورت را می بندد.که ما می توانیم مقدار تایم ان را در باکس Edit Text افزایش داد. مانند شکل زیر



در این بخش به قسمت عملی مدار در پروتئوس می پردازیم .در شکل زیر مدار عملی را مشاهده می کنید.



در این مدار برای جلوگیری از شلوغ شدن ،و سیم کشی مدار از پین ورودی ،وخروجی استفاده کرده ایم .

پین هایی که در میکرو وصل شده اند ،به عنوان ورودی ،و پین های استفاده در ال سی دی به عنوان خروجی می باشد.

و در سایر موارد از بر چسب استفاده کرده ایم .

پوش پاتن 1 و 2 را به INT1 و INT0 در میکرو وصل کرده ایم .

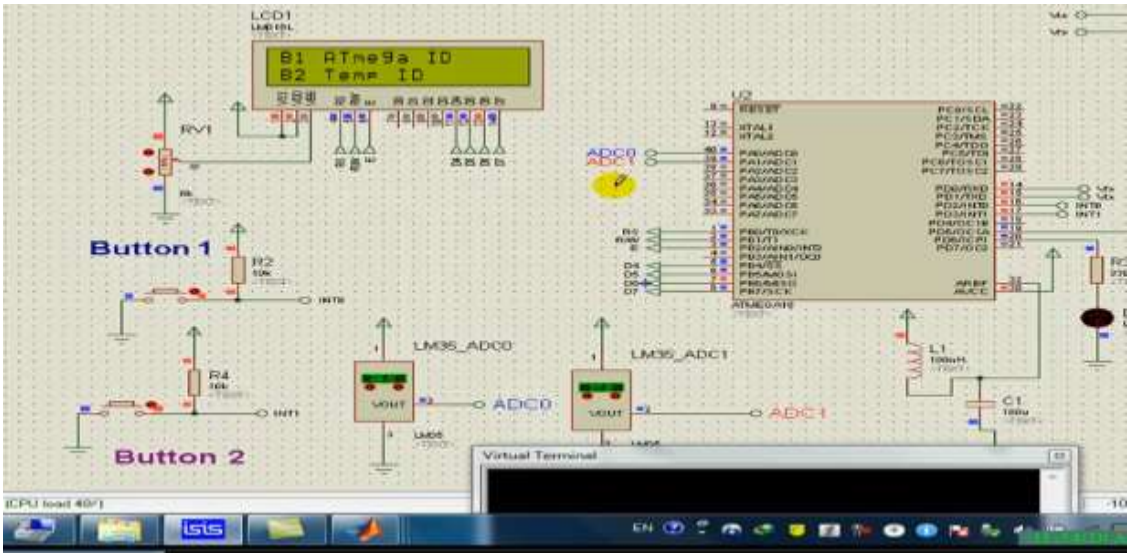
دو سنسور LM35 را به ADC0 و ADC1 از پورت A وصل کرده ایم .

ال سی دی را به پورت ,B وصل می کنیم .دو LED در این مدار LED اول در صورت روشن نشدن به دلیل تغذیه نشدن ان است ،یا میکرو ما به درستی پروگرم نشده است .و LED دوم موقعی که از طرف مقابل میکرو کنترلر می گیرد به صورت چشمک زن در می اید ،مبنی بر فرستادن و گرفتن اطلا عات است.همان طور که در قسمت های قبل دید متلب ما اطلاعات را به درستی ارسال و دریافت می کند ،تغذیه مربوط به AVCC را به صورت جداگانه وصل کنید.چون میکرو کنترلر با فرکانس بالا کار می کند همان فرکانس باعث ایجاد نویز در دمای ما می شود،پس از تغذیه ی چدا استفاده می کنیم .

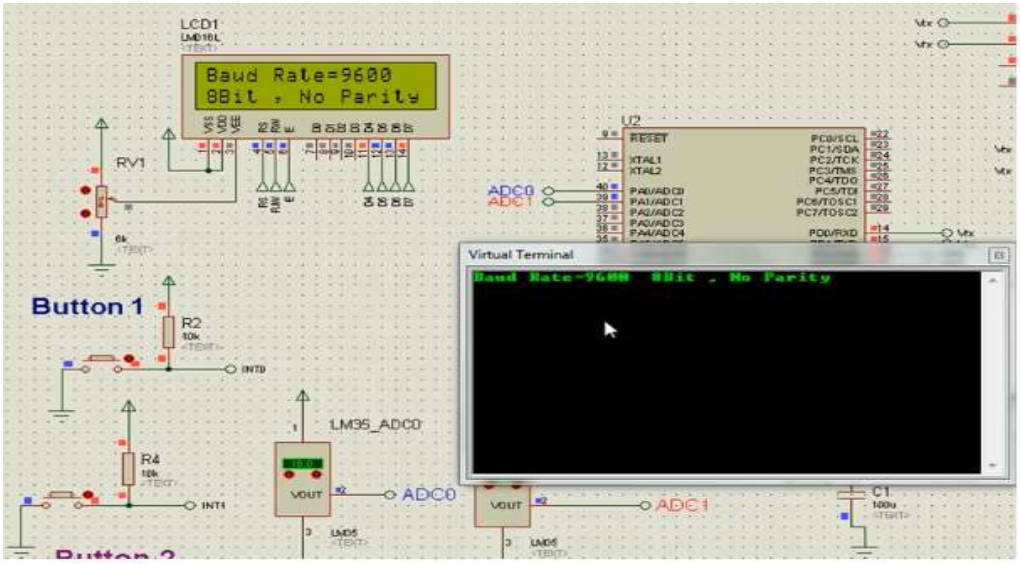
و از یک فیلتر برای جلوگیری از فرکانسهای ناخواسته که از یک سلف و خازن را به زمین وصل کرده ایم .

در مرحله بعد با RUN کردن پروتئوس می بینیم که LCD مدار روشن می شود.

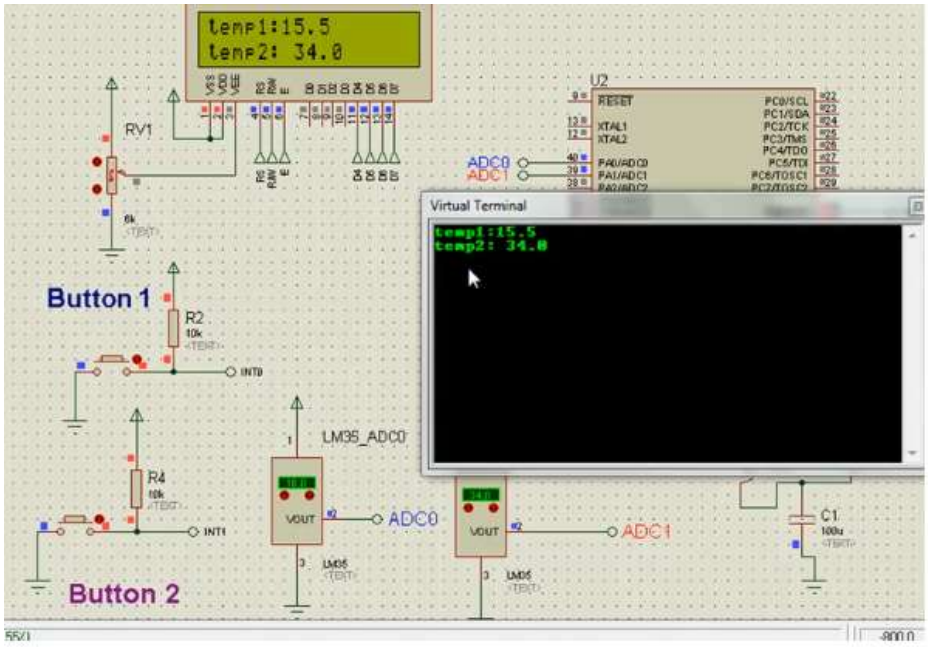
همان طور که در شکل زیر مشاهده می کنید ،با فشردن پوش پاتن 1 اطلاعات مربوط به AT Mega 16 ظاهر می شود.با فشردن پوش باتن 2 یعنی دما های ما را نشان خواهد داد.



در اینجا ما با فشردن Button1 علاوه بر اینکه در ال سی دی نمایش می دهد ،وهم یک نسخه از ان را به طرف مقابل می فرستد که همان متلب ما باشد ،در اینجا virtual terminal ما می باشد .



و در شکل پایین قسمت دما ما را نشان می دهد.



همان طور که ملاحضه می کنید سنور1 را دمای 16 و سنسور 2 دمای 34 به صورت دستی داده بودیم ، بر روی ال سی دی دمای سنسور 1 را با اختلاف 0.5 یعنی 15.5 نشان و دمای سنسور2 همان دمای 34 را نشان داده.در اینجا دما ها پشت

سر هم نشان داده نشده زیرا در کد ویژن بعد از فرستادن اولی کد اسکس 13 را می فرستد ،به منزله کلید اینتر می باشد .

پس LED اول تغذیه میکرو و LED دوم اطلاعاتی وارد یا خارج یوزارت می شود.پس با نوشتن کاراکتر h در متلب یعنی دمای سنسور 1 را بگیر و به طرف مقابل بفرست ،و کاراکتر f دما سنسور 2 را دریافت و به طرف مقابل می فرستد.

نکته : BUDA RATE در COMPIM 9600 می باشد ،اما در VIRTUAL TERMINALعدد 2400 است.

پس از RUN کردن داخل سنسور 1 در متلب کاراکتر h می باشد .

یعنی hرامی فرستد واز طرف مقابل دمای سنسور 1 را دریافت می کند.

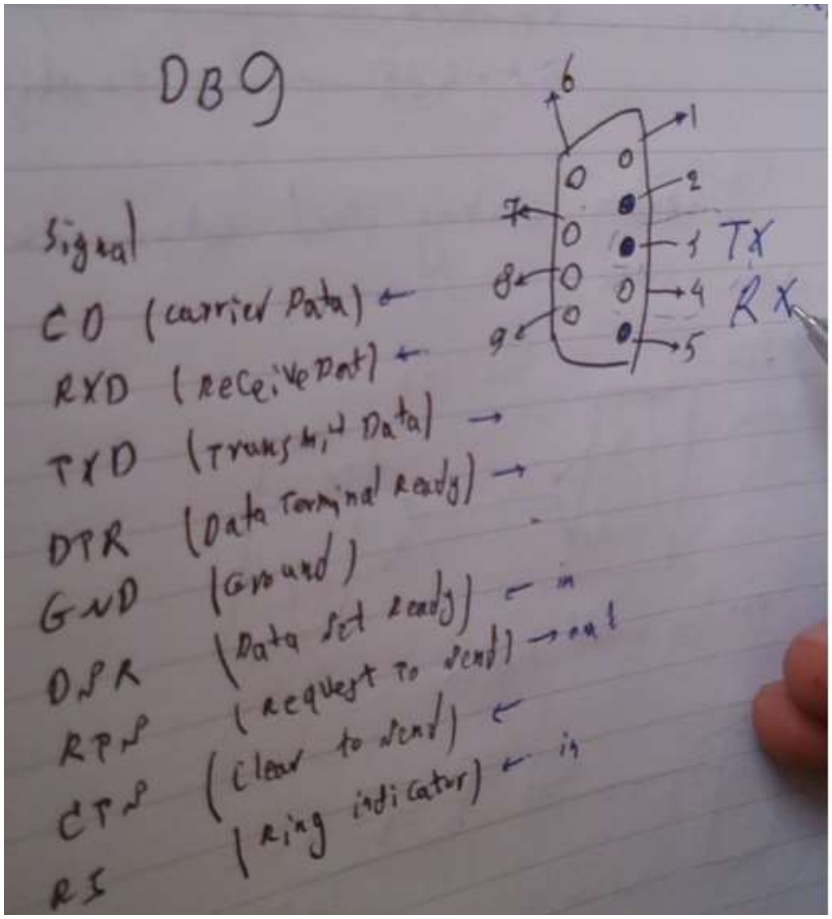
وسنسور شماره دوم f را می فرستد واز طرف مقابل دمای سنسور 2 را دریافت می کند.

در ادامه می خواهیم میکرو کنترلررا از طریق پروتکل یوزارت ارسال کنیم به کامپیوتر.در کامپیتوتر ما باید برنامه ای نظیر SERIAL PORT UTIBITY که در این پروژه ما از این برنامه استفاده می کنیم .

اگر بخواهیم میکرو کنترلر را به کامپیوتر وصل کنیم ،برای این کار از DEVICE به نام USB به سریال استفاده می کنیم .

چون در PC های قدیمی خودشون یه پورت داشتند ،به نام DB9 یعنی 9 پین داشتند ،اما امروزه در لپ تاب ها که DN9 ها در داخل انها قرار گرفته و از طریق USB به لپ تاب ما وصل می شوند.که در شکل زیر مشاهده می کنید.

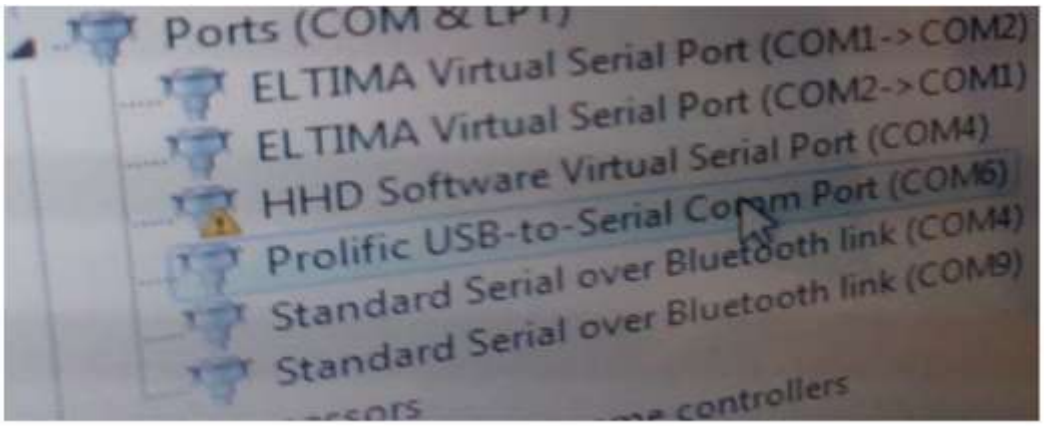
و در شکل زیر ادرس 9 پین را مشاهده می کنید .



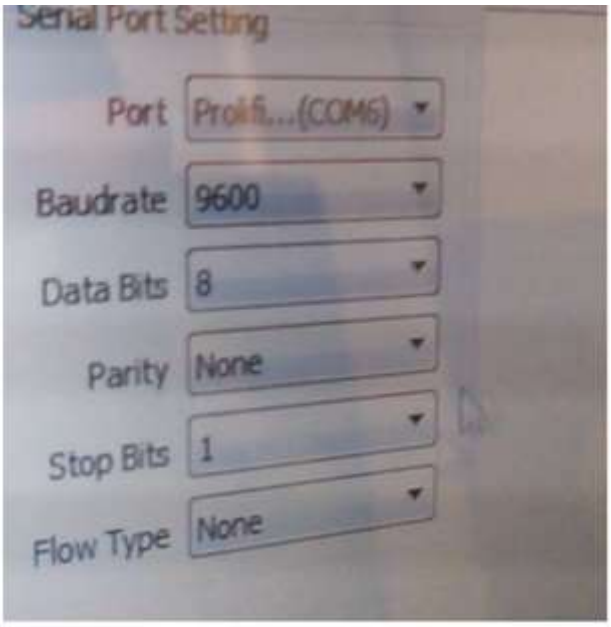
در سریال به USB که پنج تا سیم وجود دارد،که در شکل زیر به معرفی انها پرداخته ایم :

همان طور که مشاهده می کنید سیم اول از بالا تغذیه 5 ولت می باشد،دومی گراند ،سومی گیرنده ،چهارمی گیرنده ،و پنجمی 3.3 ولت برای تغذیه ی خود لپ تاب می باشد.

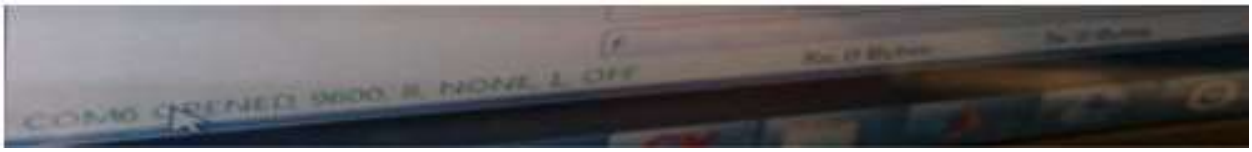
پس از وصل کردن USB به لپ تاب وارد قسمت شکل زیر می شویم .



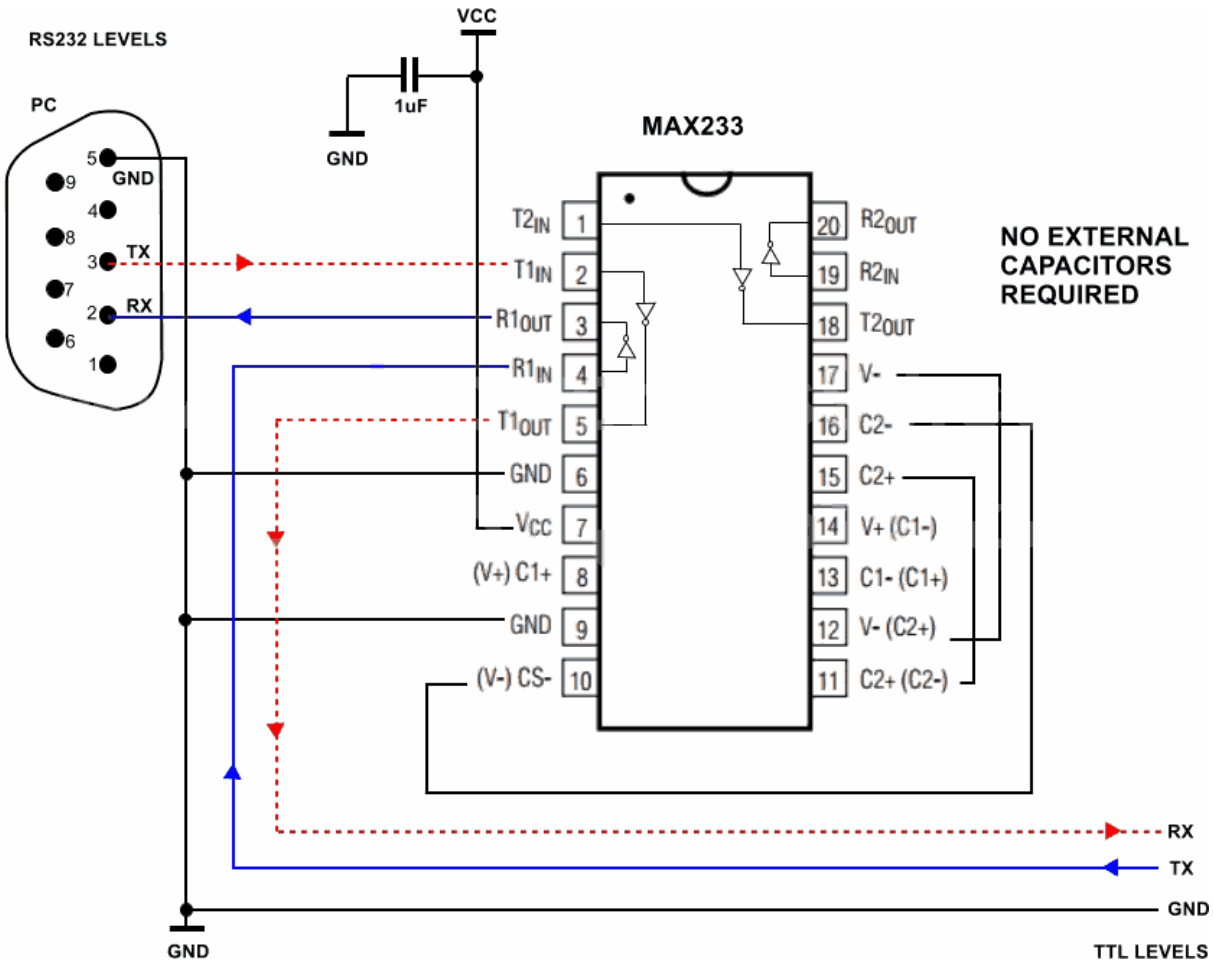
همان طور که مشاهده می کنید USB TO SERIAL به COM 6 وصل است.و تنظیمات مربوط به SERIAL PORT UTIBITY را مانند شکل زیر انجام می دهیم .



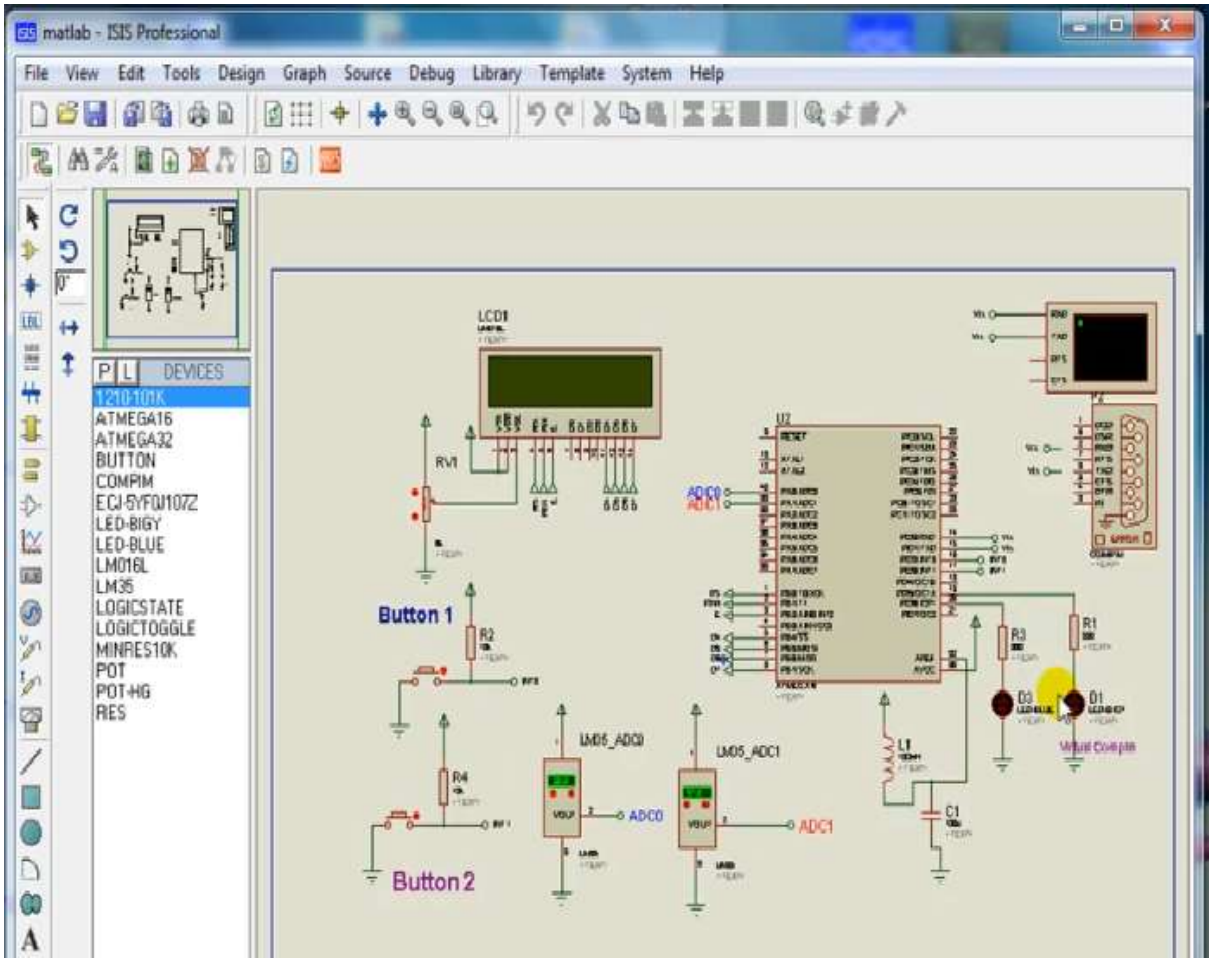
با RUN قسمت بالا نوشته سبز رنگی در پایین صفحه وجود دارد ،که نشان از درستی ان می باشد.



در شکل زیر مدار داخلی کامل USB به سریال را مشاهده می کنید.



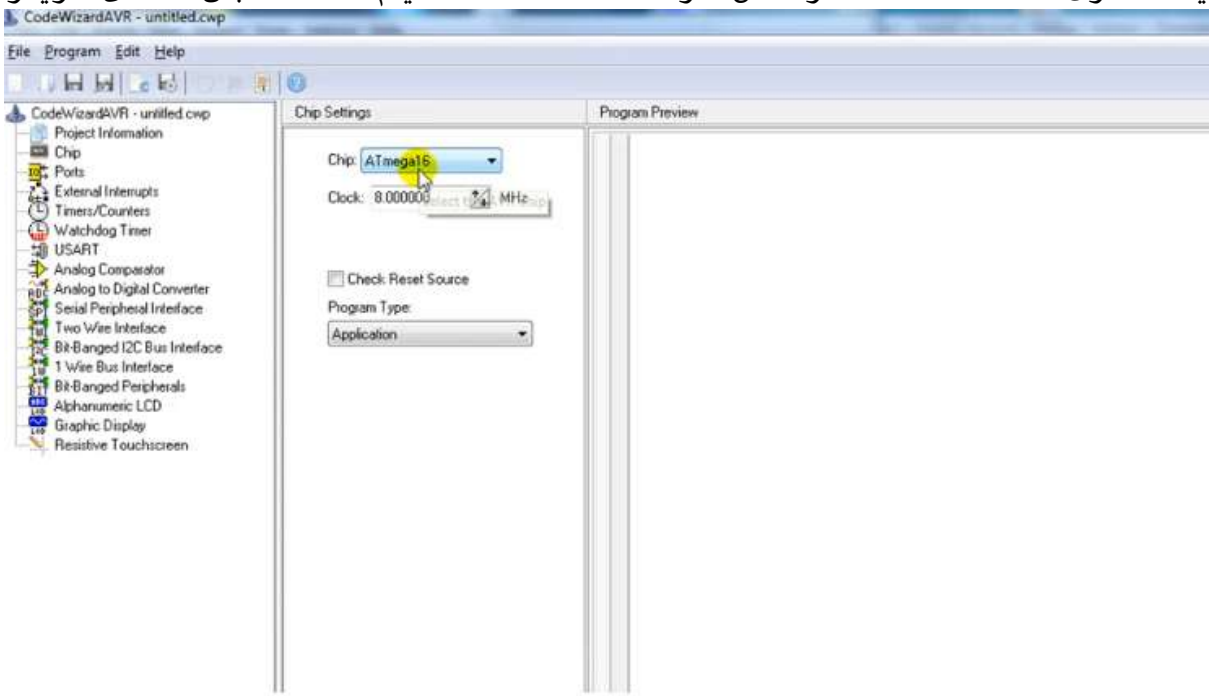
در اینجا به بعد به طور خلاصه به بررسی پروژه می پردازیم.



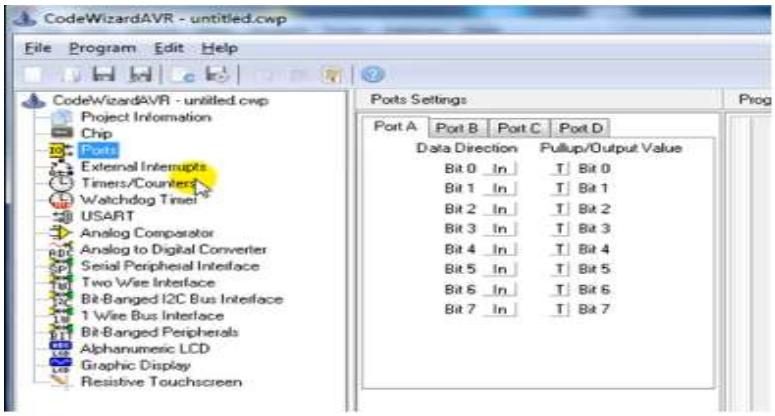
LED اول برای اطمینان از کار کرد میکرو و LED دوم دیتا رو میگره و می فرستد و چراغ ها به صورت چشمک زن می شوند.

به بررسی برنامه نویسی کد ویژن می پردازیم:

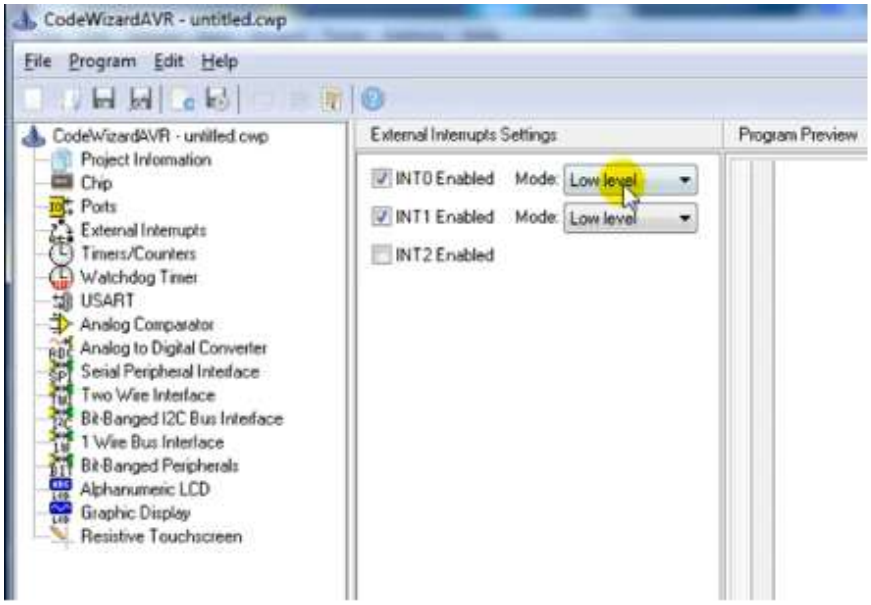
در ابتدا به قسمت کد ویزارد در برنامه کد ویژن رفته تا یک سری امکانات در ان را مشاهده کنیم مطابق شکل زیر



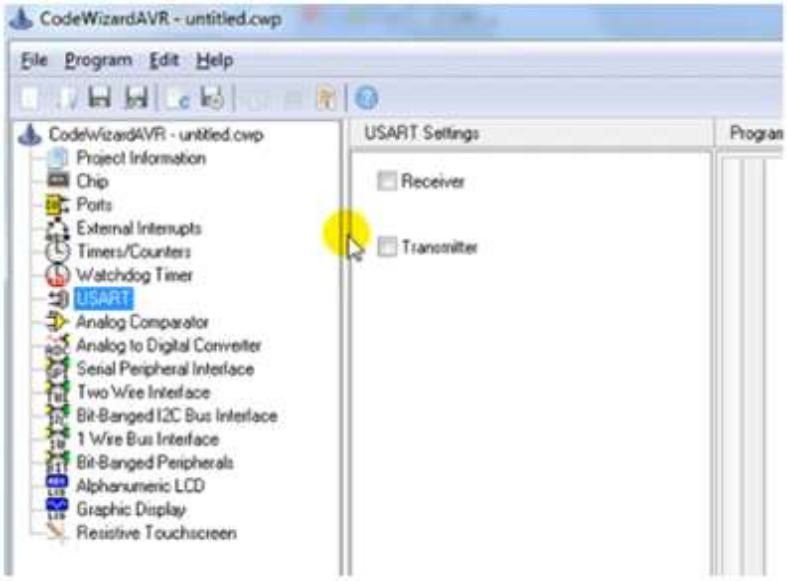
همان طور که مشاهده می کنید فرکانس ATmegah16 دارای فرکانس 8 مگاهرتز است .پورت ها رو با توجه به سمت چپ گزینه پورت تنظیم می کنیم



در قسمت بعد اینتراپت ها را فعال می کنیم در AT megah16 سه اینتراپت داریم ،ولی ما دو گزینه ی اول را فعال می کنیم



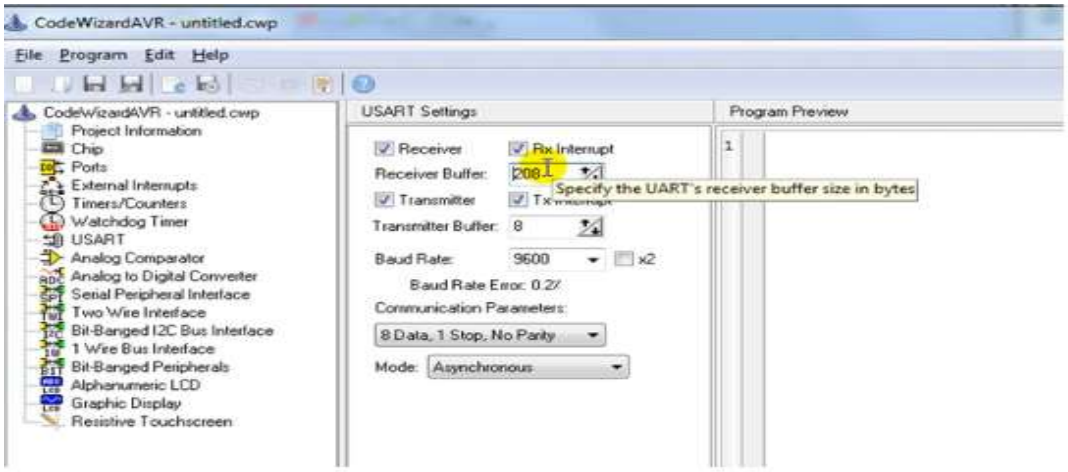
پروتکل یوزارت استفاده کردیم



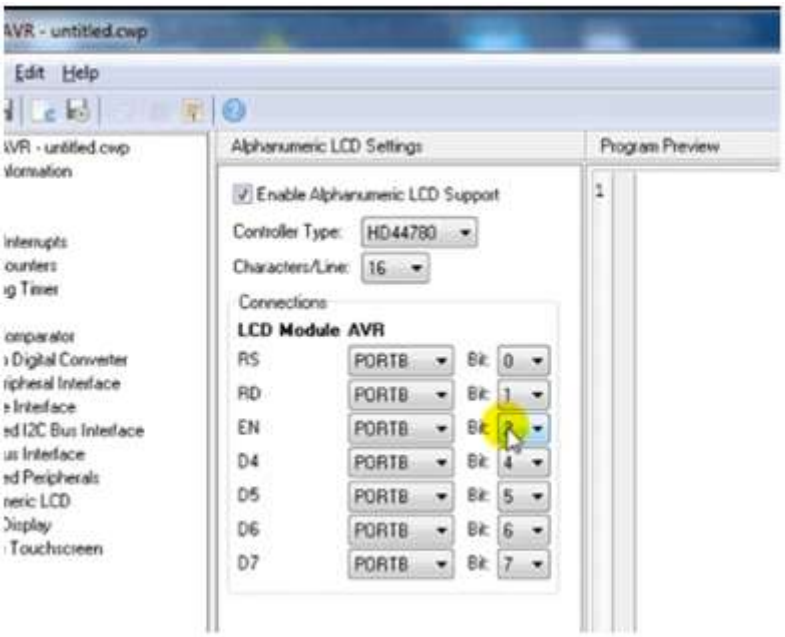
در یوزارت هم فرستنده استفاده شده و هم گیرنده ،در تنظیمات بافر گیرنده هر چقدر مقدار عدد را افزایش دهیم می توان مقدار بایت بیشتری را دریافت کرد ،به صورت پیش فرض عدد 8 می باشد که ما مقدار آن را 208 قرار می دهیم

،و مقدار فرستنده 8 کاراکترها را از طریق یوزارت می فرستد.

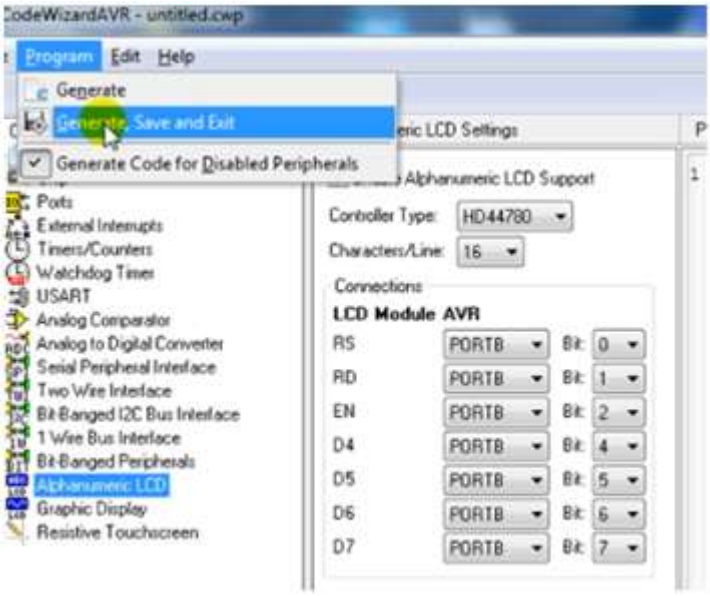
مقدار BUAD RATE ان را 9600 به صورت پیش فرض قرار می دهیم .و مد ان به صورت اسنکرون قرار می دهیم .



در مرحله بعد LCD ان را فعال می کنیم .در اینجا تمام پورت های B را فعال کرده ،و بیت های مربوط به ان را ،همان طور که مشاهده می کنید بیت 3 برای ADC می باشد.از پورت A فقط برای انالوگ به دیجیتال استفاده می کنیم .



بعد از تظیمات صورت گرفته ان را ذخیره کرده برای این کار به قسمت GENERATE SAVE AND EXIT رفته



Program Preview

```
1  #include <mega16.h>
2
3  // Alphanumeric LCD functions
4  #include <alcd.h>
5
6  // Declare your global variables here
7
8  // External Interrupt 0 service routine
9  interrupt [EXT_INT0] void ext_int0_isr(void)
10 {
11 // Place your code here
12
13 }
14
15 // External Interrupt 1 service routine
16 interrupt [EXT_INT1] void ext_int1_isr(void)
17 {
18 // Place your code here
19
20 }
21
22 #define DATA_REGISTER_EMPTY (1<<UDRE)
23 #define RX_COMPLETE (1<<RXC)
24 #define FRAMING_ERROR (1<<FE)
25 #define PARITY_ERROR (1<<UPE)
26 #define DATA_OVERRUN (1<<DOR)
```

در این قسمت اینتراپت 1 و 2 را تولید کرد.وقسمت های سبز رنگ برنامه خودش ایجاد کرده

```
}

#define DATA_REGISTER_EMPTY (1<<UDRE)
#define RX_COMPLETE (1<<RXC)
#define FRAMING_ERROR (1<<FE)
#define PARITY_ERROR (1<<UPE)
#define DATA_OVERRUN (1<<DOR)

// USART Receiver buffer
#define RX_BUFFER_SIZE 20
char rx_buffer[RX_BUFFER_SIZE];

#if RX_BUFFER_SIZE <= 256
unsigned char rx_wr_index=0,rx_rd_index=0;
#else
unsigned int rx_wr_index=0,rx_rd_index=0;
#endif

#if RX_BUFFER_SIZE < 256
```

در خط اول اطمینان حاصل می کند که بافرش فرستاده شده ،خط دوم گیرنده ،خط سوم فریم ERROR خط چهارم پریتی ERROR و خط پنجم آن دیتا اورران می باشد.در خط بعدی بافر گیرنده را که مقدار ان را 20 داده بودیم ،در خط پایین ان اسم برای بافر گذاشته در واقع همان خط بالایی می باشد .در خط بعدی شرطی گذاشته شده ،که اگر بافر کمتر 256 باشد ،فعال شود.

```
// This flag is set on USART Receiver buffer overflow
bit rx_buffer_overflow;

// USART Receiver interrupt service routine
interrupt [USART_RXC] void usart_rx_isr(void)
{
char status,data;
status=UCSRA;
data=UDR;
if ((status & (FRAMING_ERROR | PARITY_ERROR | DATA_OVERRU
   {
   rx_buffer[rx_wr_index++]=data;
#if RX_BUFFER_SIZE == 256
   // special case for receiver buffer size=256
   if (++rx_counter == 0) rx_buffer_overflow=1;
#else
   if (rx_wr_index == RX_BUFFER_SIZE) rx_wr_index=0;
   if (++rx_counter == RX_BUFFER_SIZE)
      {
      rx_counter=0;
      rx_buffer_overflow=1;
      }
#endif
```

در خط بعدی از برنامه مطابق شکل بالا وقتی Over flow شدن یک بیتی را نشان می دهد که ایا این به مقدار 20 رسیده یا خیر .در خط بعدی وقتی اینتراپتی می اید این دیتا ان می باشد ،که برای ان اسم status گذاشته شده است .اطلاعاتی که می اید در رجیستر پایینی آن ،رجیستر A را داخل status قرار می دهد.

در خط بعدی یک رجیستر داریم که اطلاعات درون آن قرار می گیرد که اسم ان را data گذاشته است.

در خط بعدی اگر فریم ،پریتی،دیتا آن ERROR نداشتیم ،بیا اینا را در بافر انجام بده .

در اینجا به بررسی کلی تر برنامه می پردازیم .

```
//Project : Communication Between UART and MATLAB
/*****************************************************
*
Chip type : ATmega16
Program type : Application
AVR Core Clock frequency: 8.000000 MHz
*****************************************************
**/
#include <mega16.h>
#include <delay.h>
// Alphanumeric LCD functions
#include <alcd.h>
#include <stdio.h>
#include <stdlib.h>
// Declare your global variables here
float temp1=0,temp2=0;
char lm1[20],lm2[20],;
unsigned char i=0;
// External Interrupt 0 service routine
interrupt [EXT_INT0] void ext_int0_isr(void)
{
// Place your code here
lcd_clear();
lcd_gotoxy(0,0);
lcd_putsf("Baud Rate=9600");
lcd_gotoxy(0,1);
lcd_putsf("8Bit , No Parity");
putsf("Baud Rate=9600 8Bit , No Parity");
for(i=0;i<=10;i++){PORTD.5=1;delay_ms(100);PORTD.5=0;
delay_ms(100);}
}
```

در خط اول فراخوانی کتاب خانه

در خط دوم دستورتی از یک پورت گرفته و فرستاده شود .

در خط سوم کتاب خانه

خط بعدی FLOAT دمای 1 و 2 را تعریف کردیم ،مقدار اولیه انه را صفر قرار می دهیم .برای اینکه از float استفاده

شده چون ما می خواهیم ،دماهای با اعشار را هم نشان دهیم .

در خط بعدی قرار است یک سری کارکتر را بر روی lcd نمایش دهیم ،و این کارکترها را به طرف مقابل بفرتد که همان متلب است.که 20 خانه را برای دمای 1 و 20 خانه برای دمای 2 اختصاص داده می شود .

در خط بعدی یک i تعریف شده که همان شمارنده است که قرار است 10 بار بشمارد چراغ چشمک زن غعال شود.

در خط بعدی اینتراپت را بزنیم بیا و lcd را پاک کن در اینجا همان کلید 1 ماست.

برو به lcd خانه ی اول و بنویس ("Baud Rate=9600");

خط بعدی برو خانه ی دومی از خط اولش بنویس ("8Bit , No put Parity");

حرف puts f حرف f در اینجا به این معنی است که این دستور را در حافظه فلش میکرو ذخیره کن .

در خط بعدی هم یک نسخه را هم می خواهیم که به طرف مقابل همین اطلاعات را بفرستد،که اطلا عات را از طریق یوزارت می فرستد.

طرف مقابل ما virtual terminal است .

```
// External Interrupt 1 service routine
interrupt [EXT_INT1] void ext_int1_isr(void)
{
// Place your code here
lcd_clear();
//////////////LM35 first////////
temp1=read_adc(0);
temp1 =(temp1/2.0)-1;
sprintf(lm1,"temp1:%3.1f",temp1);
lcd_gotoxy(0,0);
lcd_puts(lm1);
puts(lm1);
putchar(13);
for(i=0;i<=10;i++){PORTD.5=1;delay_ms(100);PORTD.5=0;
delay_ms(100);}
//////////////LM35 second////////
temp2=read_adc(1);
temp2=(temp2 /2.0)-1;
sprintf(lm2,"temp2: %3.1f",temp2);
lcd_gotoxy(0,1);
lcd_puts(lm2);
//putchar(13);
puts(lm2);
delay_ms(300);
////////////////////
}
void main(void) /* mian part on programming */
{
// Declare your local variables here
char ch;
DDRD=0b01100000;
```

در اینجا اینتراپت ها را بررسی می کنیم .

اینتراپت شماره 2 در اینجا خط اول lcd را پاک کن ،بعد بیا سنسور شماره یک را بخوان پایه صفر را بخوان

و بزار داخل TEMP1 با توجه به عددی که در داخل LCD نمایش می دهد 42 ان را بر 2 تقسیم کرده و منهای یک کرده ان را =(temp1/2.0)-1;

```
// External Interrupt(s) initialization
// INT0: On
// INT0 Mode: Falling Edge
// INT1: On
// INT1 Mode: Falling Edge
// INT2: Off
GICR|=(1<<INT1) | (1<<INT0) | (0<<INT2);
MCUCR=(1<<ISC11)   |   (0<<ISC10)   |   (1<<ISC01)   |
(0<<ISC00);
MCUCSR=(0<<ISC2);
GIFR=(1<<INTF1) | (1<<INTF0) | (0<<INTF2);
// USART initialization
//  Communication  Parameters:  8  Data,  1  Stop,  No
Parity
// USART Receiver: On
// USART Transmitter: On
// USART Mode: Asynchronous
// USART Baud Rate: 9600
UCSRA=(0<<RXC)  |  (0<<TXC)  |  (0<<UDRE)  |  (0<<FE)  |
(0<<DOR) | (0<<UPE) | (0<<U2X) | (0<<MPCM);
UCSRB=(1<<RXCIE)   |   (1<<TXCIE)   |   (0<<UDRIE)   |
(1<<RXEN)  |  (1<<TXEN)  |  (0<<UCSZ2)  |  (0<<RXB8)  |
(0<<TXB8);
UCSRC=(1<<URSEL) | (0<<UMSEL) | (0<<UPM1) | (0<<UPM0)
| (0<<USBS) | (1<<UCSZ1) | (1<<UCSZ0) | (0<<UCPOL);
UBRRH=0x00;
UBRRL=0x33;
// ADC initialization
// ADC Clock frequency: 1000.000 kHz
// ADC Voltage Reference: AREF pin
// ADC Auto Trigger Source: ADC Stopped
ADMUX=ADC_VREF_TYPE;
ADCSRA=(1<<ADEN) | (0<<ADSC) | (0<<ADATE) | (0<<ADIF)
| (0<<ADIE) | (0<<ADPS2) | (1<<ADPS1) | (1<<ADPS0);
SFIOR=(0<<ADTS2) | (0<<ADTS1) | (0<<ADTS0);
lcd_init(16);
//////////checking LCD and Microcontroller//////////
```

در قسمت های بالایی برنامه از اینتراپت خارجی را گفته شده ،که مد ان با لبه ی پایین رونده فعال می شود.

قسمت بعدی دارای یوزارت است .

قسمت بعدی ADC است .

قسمت بعدی برای اماده سازی LCD است .

```
ch=getchar();
if(ch=='h')
{
```

```
lcd_clear();
lcd_gotoxy(0,0);
lcd_putsf("GUI asked tmep1");
temp1=read_adc(0);
temp1 =(temp1/2.0)-1;
sprintf(lm1,"temp1:%5.1f",temp1);
lcd_gotoxy(0,1);
lcd_puts(lm1);
puts(lm1);
for(i=0;i<=10;i++){PORTD.5=1;delay_ms(100);PORTD.5=0;
delay_ms(100);}
}
//////////////////instruction with IF///////
if(ch=='f')
{
lcd_clear();
lcd_gotoxy(0,0);
lcd_putsf("GUI asked tmep2");
temp2=read_adc(1);
temp2=(temp2 /2.0)-1;
sprintf(lm2,"temp2: %3.1f",temp2);
lcd_gotoxy(0,1);
lcd_puts(lm2);
puts(lm2);
for(i=0;i<=10;i++){PORTD.5=1;delay_ms(100);PORTD.5=0;
delay_ms(100);}
}
/////////////
```

```
else
{
lcd_clear();
lcd_gotoxy(0,0);
lcd_putsf("B1 ATmega ID");
lcd_gotoxy(0,1);
lcd_putsf("B2 Temp ID");
}
}
}
```

در قسمت بعدی برنامه ;getchar() دستوری است که خود کد ویژن درست کرده است ،که میره پایه ی بافر یوزارت را می خواند.در دستور if اگر h را گرفت ، lcd را پاک کن و در خونه ی اول GUI متلب از میکرو کنترلر دمای سنسور 1 را می خواهد بعد دما را بخوان ،یک نسخه را روی LCD چاپ کن ویک نسخه ان را به طرف مقابل بفرست ،وقسمت چشمک زن آن را نشان می دهد.

در قسمت بعدی برنامه حرف f است ،که کارایی آن شبیه حرف h است ولی برای سنسور 2 می باشد.

در قسمت آخر برنامه else اگر دیتایی که از طرف مقابل می اید ،نه h باشد و نه حرف f باشد ،به صورت پیش فرض خط پایین را بنویس در سطر اول ;("B1 ATmega ID") ودر خانه ی دومی در سطر دوم ;("B2 Temp ID") رابنویس