

**بسمه تعالی**



دانشگاه صنعتی خواجه نصیر الدین طوسی

**دانشکده مهندسی برق و کامپیوتر**

**پروژه پایانی آزمایشگاه ریزپردازنده**

**ارتباط گوشی با میکرو از طریق بلوتوث**

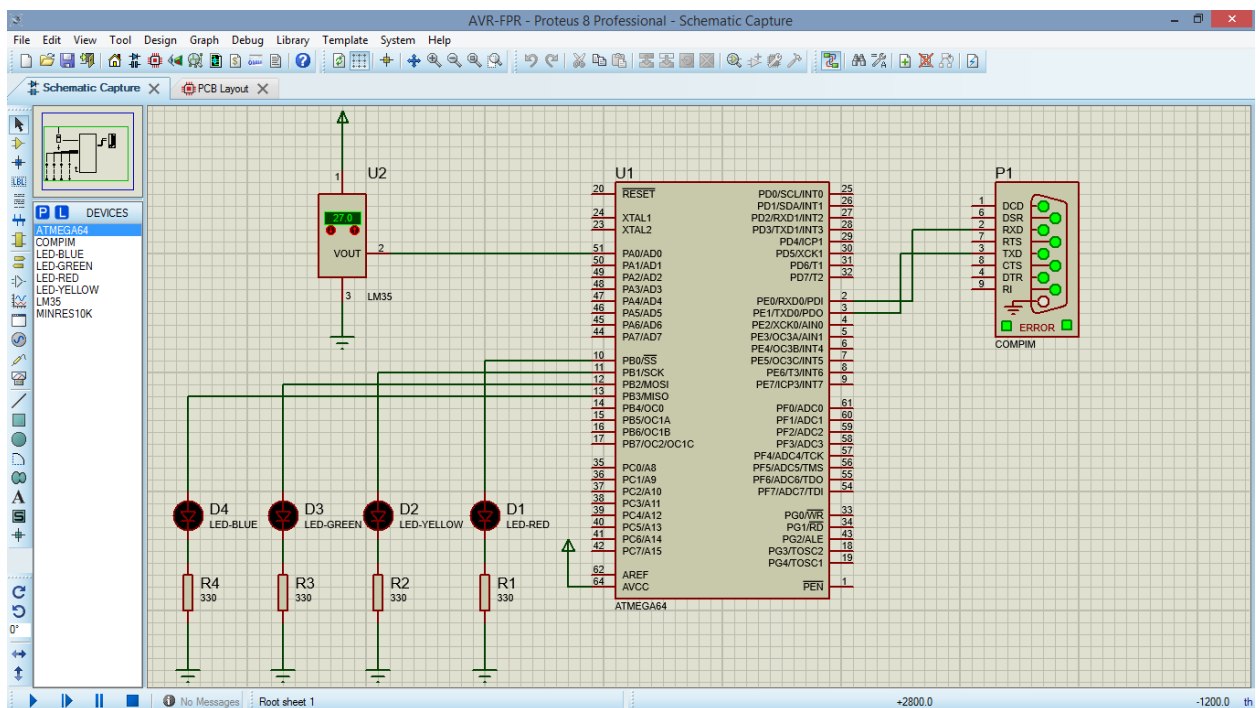
**مدرس آزمایشگاه ریزپردازنده:**

**مهندس حمید اصلانی**

**ارائه دهنده:**

**شایان علی پور**

- شرح آزمایش:  
ماژول بلوتوث را به پورت سریال میکرو وصل میکنیم.  
نرم افزار terminal for Bluetooth را بر روی گوشی آندرویدی خود نصب کرده و تلاش میکنیم به بلوتوث میکرو وصل شویم.  
سپس چهار led و یک سنسور lm35 را به ترتیب به PORTB و ADC میکرو وصل میکنیم. باید بتوانید از طریق گوشی led ها را کنترل کنید.(مثلا با ارسال عبارت on1 اولین led را روشن کنید)  
همچنین با ارسال دستور report باید دمای خوانده شده از سنسور را در گوشی دریافت کنید.

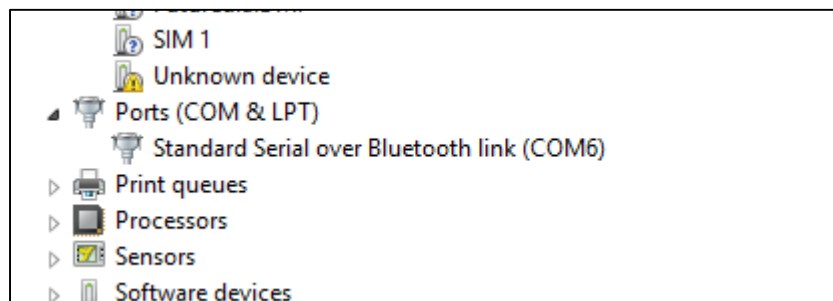


نرم افزار پروتئوس را باز کرده و پروژه‌ای جدید ایجاد میکنیم. سپس از قسمت pick devices قطعات زیر را انتخاب میکنیم:

- ATMEGA64
- LM35
- LED-COLOR
- COMPIM
- RESISTORS

این قطعات را مطابق شکل صفحه قبل به هم وصل میکنیم. حال باید تغییراتی در برخی از این قطعات ایجاد کنیم.

برای اتصال بلوتوث به کامپیوتر لازم است ابتدا آن دو را با یکدیگر pair کنیم. بعد از اینکار به قسمت device manager کامپیوتر رفته و بدنبال گزینه ports میگردیم. اتصال بلوتوث را پیدا میکنید. عبارت داخل پرانتز را یادداشت میکنیم.



در اینجا ما COM6 را داریم. سپس به پروتئوس باز میگردیم و بروی COMPIM کلیک میکنیم و سپس قسمت های physical port, physical baud rate و virtual baud rate را مطابق شکل صفحه بعد تغییر میدهیم. بادریت پورت فیزیکی و مجازی باید یکسان باشد.

**Edit Component**

Part Reference:  Hidden: ☐

Part Value:  Hidden: ☐

Element:  New

VSM Model:  Hide All

Physical port:  Hide All

Physical Baud Rate:  Hide All

Physical Data Bits:  Hide All

Physical Parity:  Hide All

Virtual Baud Rate:  Hide All

Virtual Data Bits:  Hide All

Virtual Parity:  Hide All

Advanced Properties:

Physical Stop Bits  Hide All

Other Properties:

☐ Exclude from Simulation      ☐ Attach hierarchy module  
☐ Exclude from PCB Layout      ☐ Hide common pins  
☐ Exclude from Bill of Materials      ☐ Edit all properties as text

OK Help Cancel

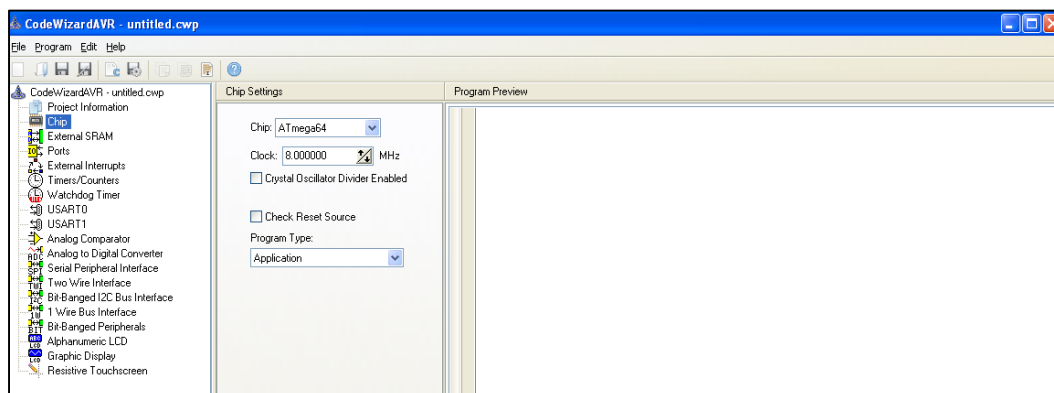
بعد از تنظیم COMPIM به سراغ مقاومت ها میرویم. مقدار پیش فرض آن ها ۱۰ K میباشد. بروی مقدار مقاومت کلیک میکنیم و آن را به ۳۳۰ تغییر میدهیم.

- کد برنامه



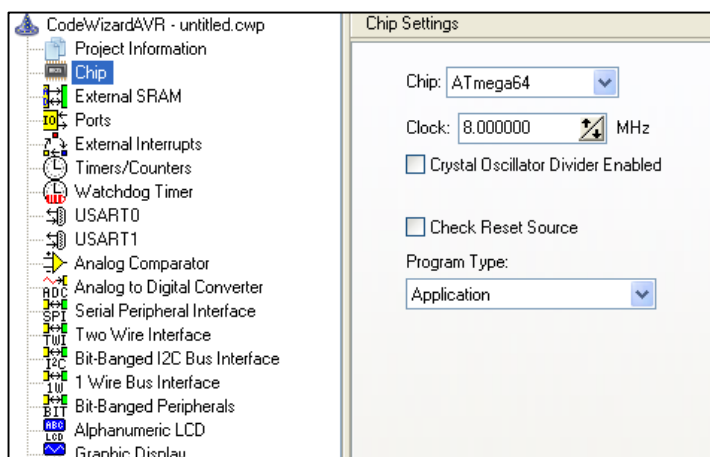
کد را با استفاده از CODE VISION کامپایل میکنیم. برای شروع برنامه را باز کرده و گزینه NEW PROJECT را انتخاب میکنیم. با پنجره روبرو مواجه میشویم، OK میزنیم و به قسمت بعد میرویم.

بعد از انتخاب OK با پنجره زیر مواجه می شویم:

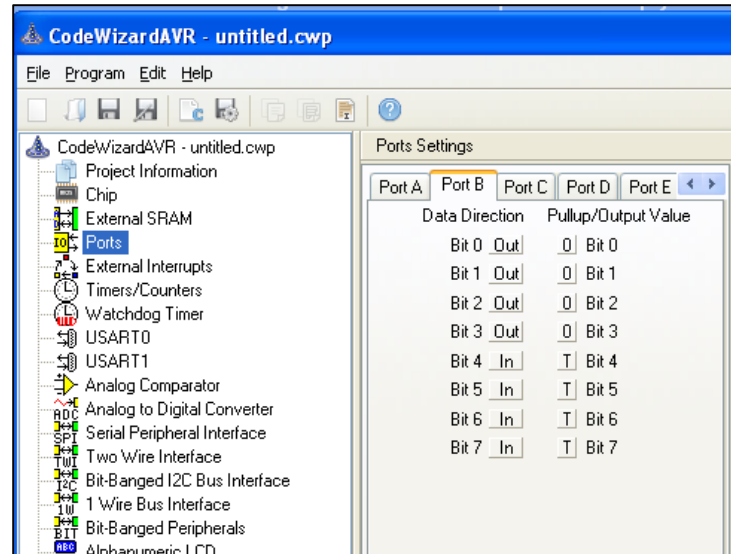


در این قسمت باید تغییراتی در بخش Chip, Ports, USART0 و Analog to Digital Converter ایجاد کنیم. هر بخش را مطابق شکل آن تغییر دهید.

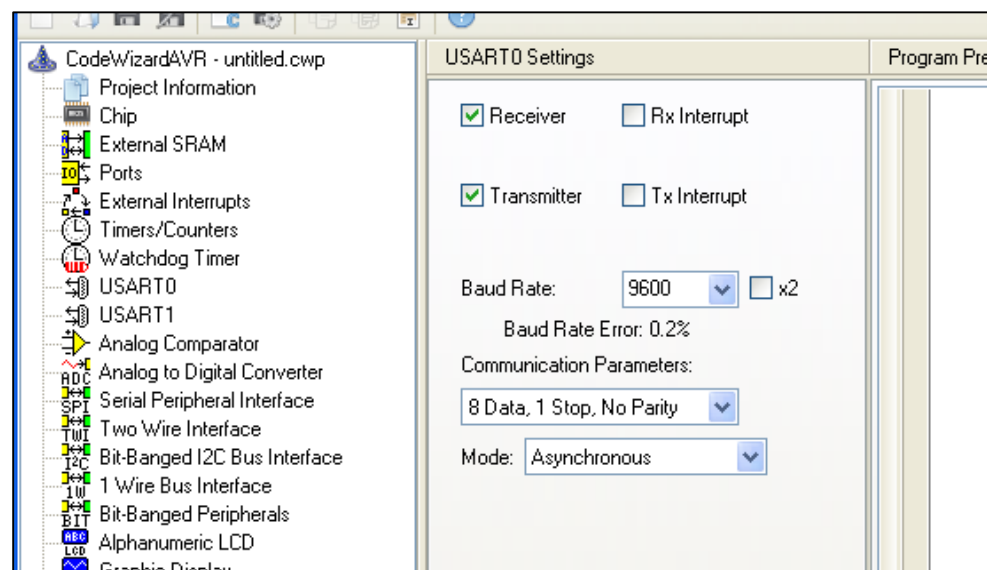
- Chips



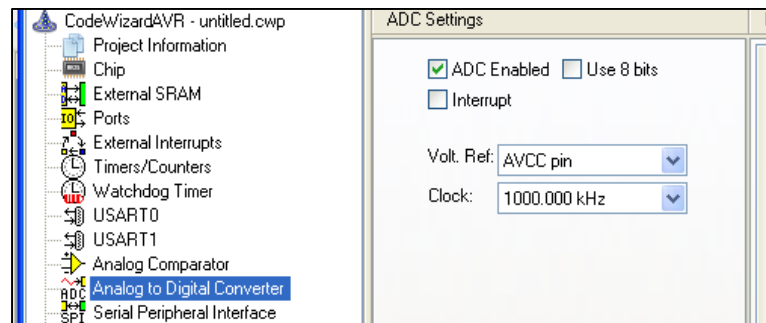
- Ports



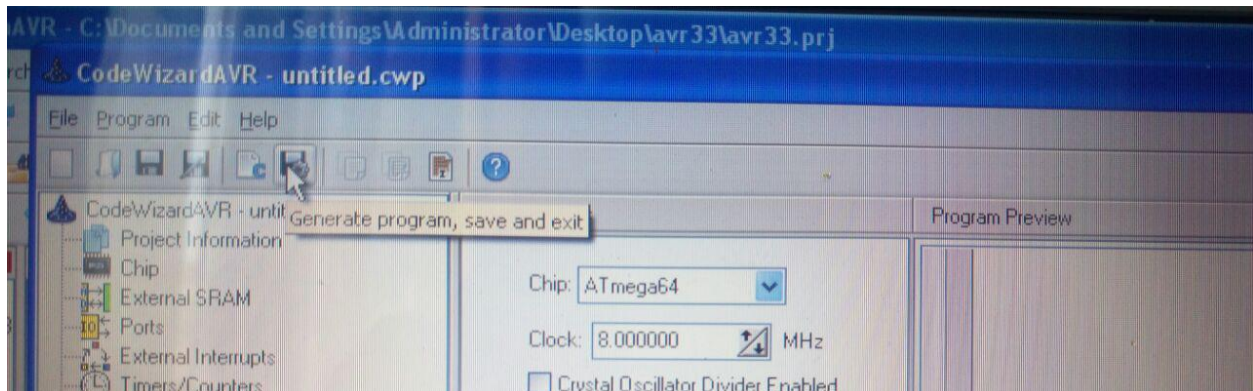
- USART0



- Analog to Digital Converter



پس از اعمال تغییرات فوق گزینه generate programe را انتخاب میکنیم.



حال میتوانیم کد زدن را شروع کنیم!

کد کامل برنامه در ضمیمه آمده است. در این قسمت تنها بخشی از کد آورده شده است.

```
while (1) {  
    gets(str, 3);  
    if (str[0] == 'o' & str[1] == 'n' & str[2] == '1') {  
        PORTB.0 = 1;  
        printf("led 1 is 1");  
    }  
    else if (str[0] == 'o' & str[1] == 'n' & str[2] == '2') {  
        PORTB.1 = 1;  
        printf("led 2 is 1");  
    }  
}
```

متغیر `str` یک آرایه ای از جنس کارکتر میباشد که سه حرف دارد. کاربر ورودی را در گوشی خود وارد میکند، به طور مثال "on2"، سپس متغیر `str` در شرطهای متعدد بررسی میشود تا مشخص شود کدام `led` باید خاموش یا روشن شود. برای ارسال گزارش نیز از دستور `puts()` استفاده میکنیم.

سنسور `lm35` ولتاژ را گرفته و در عددی ثابت ضرب میکند و در اسکیل ۰ تا ۱۰۰ نمایش میدهد. مراحل اینکار در کد مشخص است.