

به نام خدا

آموزش میکروکنترلر AVR به زبان C (WinAVR)

(جلسه چهارم)



مقدمه:

در جلسه قبل به راه اندازی LCD گرافیکی و توابع مربوطه پرداختیم. در این جلسه تصمیم داریم به معرفی پورت سریال میکرو و نحوه ارتباط میکرو با میکروی دیگر و نیز ارتباط با کامپیوتر توسط پروتکل سریال و استاندارد (واسط) RS232 بپردازیم.

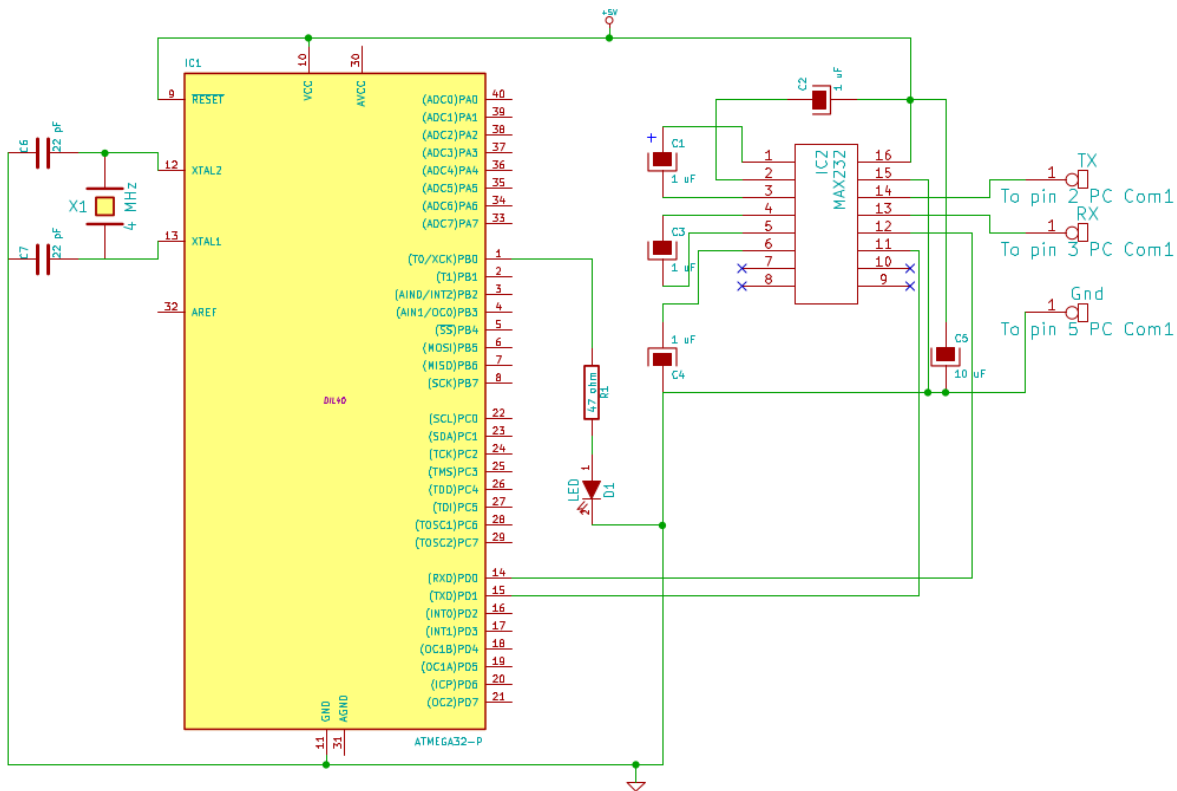
تصویر ابتدای این مقاله مربوط به کامپیوتر جیبی ZQ290II می باشد که اولین کامپیوتر نویسنده این مقاله بود (سال ۱۳۸۳). این کامپیوتر یا PDA قادر بود اطلاعات (شماره تلفن ها، یادداشت ها و ایمیل ها) را از طریق پورت سریال با نرم افزار MS Outlook هماهنگ کند و به اصطلاح به تبادل اطلاعات توسط پورت سریال کامپیوتر و تنها با سه رشته سیم بپردازد. هرچند هم اکنون از این کامپیوتر فقط عکس آن باقی مانده است ولی شاید علاقه و اشتیاق فراوان به یادگیری چگونگی انتقال اطلاعات بین این دستگاه و کامپیوتر و در نهایت خراب کردن آن جرقه ای بوده است برای رفتن به دنبال علم دیجیتال، میکروکنترلر 8051 و AVR و در نهایت نوشتن این سری مقالات به شیوا ترین روش ممکن برای علاقه مندان به این علم.

پروتکل سریال برای ارتباط

یکی از مسائلی که در پروژه های بزرگ به وجود می آید ارتباط دو میکرو با بکدیگر و یا ارتباط میکرو با وسایل جانبی و ماژولهایی است که با پروتکل سریال کار می کنند. همچنین ارتباط میکرو با کامپیوتر و ارسال و دریافت اطلاعات نیز یکی از موارد لازم در پروژه های ارتباط با PC و کنترل از طریق PC و یا ذخیره اطلاعات در کامپیوتر است. برای ارتباط بین میکرو و کامپیوتر از پروتکل سریال و رابط RS-232 و نیز یک آی سی خاص به نام MAX232 استفاده می شود که سطوح ولتاژ و منطق TTL میکرو را با PC سازگار می نماید. برای ارتباط دو میکرو در فواصل کوتاه می توان از این چیپ صرفنظر کرد و برای فواصل بیشتر از چند متر بایستی در کنار هر میکرو یک MAX232 قرار داد تا با تبدیل سطح ولتاژ نویز کمتری روی کابل ارتباطی اثر بگذارد. در این مقاله در پروژه اول برنامه ای برای ارتباط دو طرفه میکرو و کامپیوتر می نویسیم. در این برنامه در ابتدا متنی توسط میکرو برای ترمینال کامپیوتر ارسال می شود و سپس برنامه وارد حلقه ای می شود و منتظر دریافت کاراکتر از PC می ماند؛ در صورتی که کاربر کاراکتر q را وارد کند LED موجود در مدار میکرو روشن می شود و در صورتی که کاراکتر w را وارد کند دیود نورانی LED خاموش می شود. همچنین کاراکترهای تایپ شده توسط کاربر مجدداً توسط میکرو به ترمینال ارسال می شود تا کاربر کاراکتر تایپ شده را ببیند. (در حالت عادی اگر در ترمینال چیزی بنویسیم دیده نمی شود و فقط ارسال می گردد، محیط ترمینال مانند یک ادیتور مثل Notepad نیست).

شماتیک پروژه ارتباط میکروکنترلر با کامپیوتر

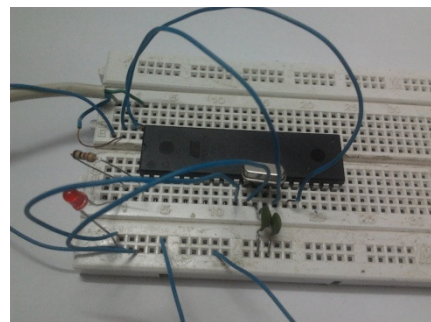
قبل از هر چیز شماتیک زیر را بر روی برد بورد ببینید، در این شماتیک از آی سی MAX232 برای ارتباط با پورت سریال کامپیوتر بهره گرفته شده است. ترمینال های سمت راست شماتیک به ترتیب از بالا به پایین به پایه های 2 و 3 و 5 کانکتور مادگی DB9 متصل می شود که پایه اول که به پین شماره 2 کانکتور پورت سریال کامپیوتر متصل می شود ارسال (TX) از طرف میکرو و دریافت از طرف PC می باشد، پایه بعدی که به پایه شماره 3 پورت سریال PC وصل می شود ارسال از طرف PC و دریافت (RX) از طرف میکرو است. پایه 5 نیز زمین مشترک می باشد که سطوح ولتاژ نسبت به آن سنجیده می شود. منطق و سطح ولتاژ در سمت PC متفاوت با منطق و ولتاژ TTL میکرو بوده و معمولاً در محدوده ای بالاتر از 5 ولت قرار دارد.



کانکتور مادگی شبیه شکل زیر می باشد:



چیدمان قطعات بر روی برد به صورت زیر است:



البته ما قسمت MAX232 را بر روی برد مونتاژ نکرده ایم و از ماژول آماده ای که قبلاً برای این کار ساخته بودیم استفاده کردیم، تصویر این ماژول در زیر نشان داده شده است:



پس از مونتاژ قطعات بر روی برد نوبت به کد نویسی پروژه می رسد. در مرحله بعد برنامه ای توسط WinAVR می نویسیم و با استفاده از بخش پروگرامر Bascom بر روی میکرو پروگرام می کنیم. در جلسات قبل نحوه پروگرام کردن میکرو توسط بسکام با یک فایل Hex آموزش داده شده است که در این جلسه از تکرار آن خودداری می کنیم. برای پروژه های این مقاله بایستی مطابق نقشه از کریستال خارجی ۴ مگاهرتز بر روی پایه های XTAL1 و XTAL2 میکرو استفاده کرده و فیوز بیت کریستال را روی گزینه آخر یعنی 111111 قرار دهید. (فیوزبیت کریستال خارجی برای تمام فرکانس ها از ۱ مگاهرتز تا ۱۶ مگاهرتز بر روی یک تنظیم قرار می گیرد).

Fusebit KLA987

111111:Ext. Crystal Osc.

کدنویسی پروژه ارتباط میکروکنترلر با کامپیوتر

برای برنامه نویسی این پروژه به کتابخانه ای برای پورت سریال به نام Uart نیاز داریم. متأسفانه کتابخانه Uart موجود در avrlib به هیچ وجه کار نکرد و مجبور شدیم از یک کتابخانه free دیگر که از اینترنت دانلود کردیم استفاده کنیم. این کتابخانه در پوشه ای جدا در پوشه این مقاله به همراه سورس پروژه آورده شده و نام آن نیز دقیقاً uart.h و uart.c می باشد، پس آن را با کتابخانه استاندارد avrlib اشتباه نگیرید.

```

main.c
#include <avr/io.h>
#include <avr/interrupt.h>
#include "printf.h"
#include "lcd.h"
#include "uart.h"
#include <util/delay.h>

#include <avr/pgmspace.h>
#include <stdio.h>

/* define CPU frequency in Mhz here if not defined in Makefile */
#ifndef F_CPU
#define F_CPU 4000000UL
#endif

/* 9600 baud */
#define UART_BAUD_RATE 9600

int main(void)
{
    DDRB = 0b00000001;
    unsigned int i;
    char buffer[7];
    int num=134;
    PORTB = 0b00000000;
  
```

کد این پروژه در زیر آورده شده است:

```

#include <avr/io.h>
#include <avr/interrupt.h>
#include "rprintf.h"
#include "uart.h"
#include <util/delay.h>

#include <avr/pgmspace.h>
#include <stdlib.h>

/* define CPU frequency in Mhz here if not defined in Makefile */

#ifndef F_CPU
#define F_CPU 4000000UL
#endif

/* 9600 baud */
#define UART_BAUD_RATE      9600

int main(void)
{
    DDRB = 0b00000001;

    unsigned int c;
    char buffer[7];
    int num=134;
    PORTB = 0b00000000;
    _delay_ms(500);
    PORTB = 0b00000001;
    uart_init( UART_BAUD_SELECT(UART_BAUD_RATE,F_CPU) );
    sei();
    uart_puts("Send from Micro...\n");
    //uart_puts_P("String stored in FLASH\n");
    //itoa( num, buffer, 10); // convert interger into string (decimal format)
    //uart_puts(buffer); // and transmit string to UART
    //uart_putc('\r');

    _delay_ms(500);

    PORTB = 0b00000000;
    _delay_ms(500);

    while(1){
    c = uart_getc();
    if ( c & UART_NO_DATA )
    {
        /*
        * no data available from UART
        */
    }
    else
    {
        /*
        * new data available from UART
        * check for Frame or Overrun error
        */
        if ( c & UART_FRAME_ERROR )
        {
            /* Framing Error detected, i.e no stop bit detected */
            //uart_puts_P("UART Frame Error: ");
        }
        if ( c & UART_OVERRUN_ERROR )
        {
            /*
            * Overrun, a character already present in the UART UDR register was
            * not read by the interrupt handler before the next character arrived,
            * one or more received characters have been dropped
            */
            //uart_puts_P("UART Overrun Error: ");
        }
        if ( c & UART_BUFFER_OVERFLOW )

```

```

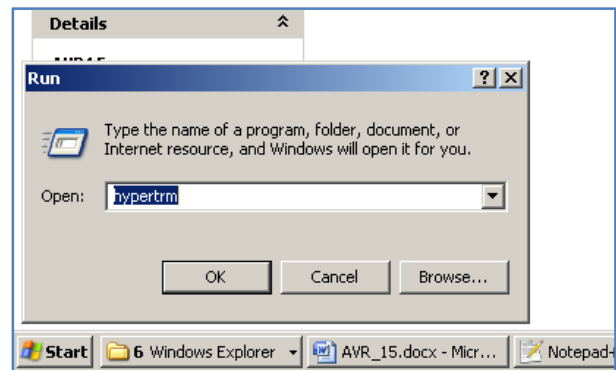
{
    /*
    * We are not reading the receive buffer fast enough,
    * one or more received character have been dropped
    */
    // uart_puts_P("Buffer overflow error: ");
}
/*
* send received character back
*/
uart_putc( (unsigned char)c );
    if (c == 'q')
    {
        PORTB = 0b00000001;
    }
    if (c == 'w')
    {
        PORTB = 0b00000000;
    }

}
}
return 0;
}

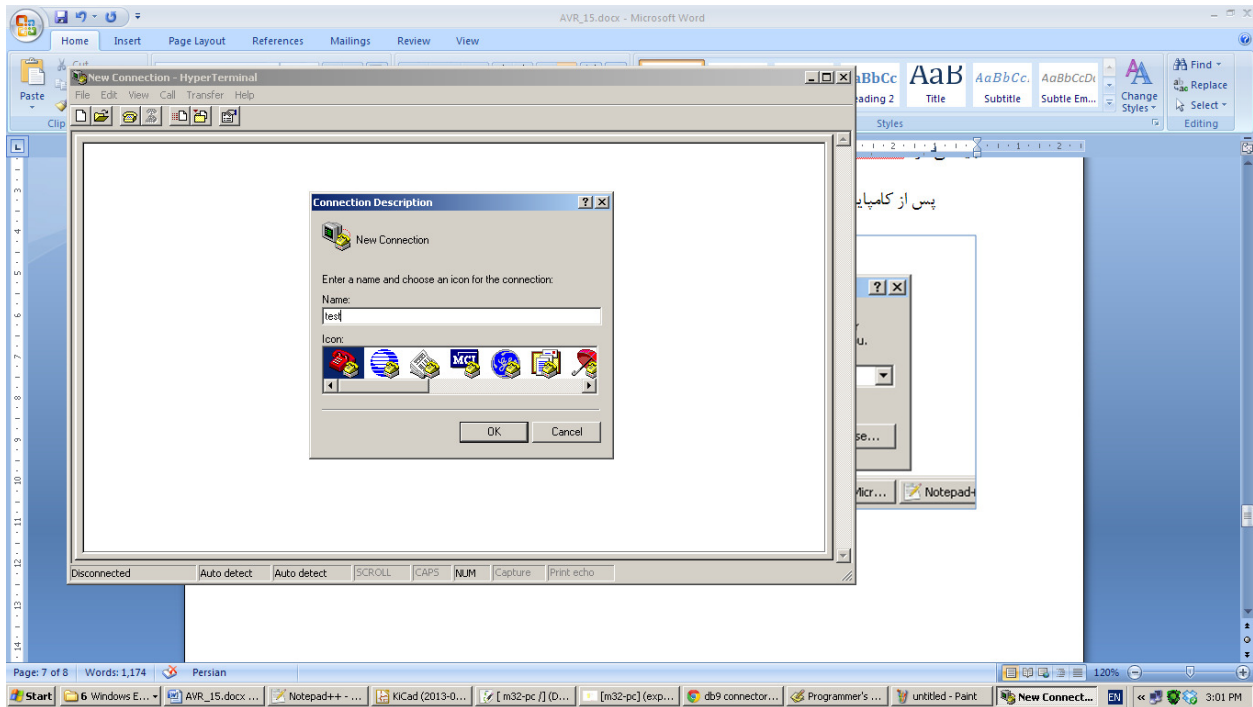
```

برای راحتی کار، تمام سورس پروژه در پوشه همراه این مقاله آورده شده است. توجه داشته باشید که نوع میکرو در این پروژه atmega32 می باشد و فرکانس کاری آن 4000000 هرتز است که در صورت تغییر بایستی در Makefile تنظیم شود.

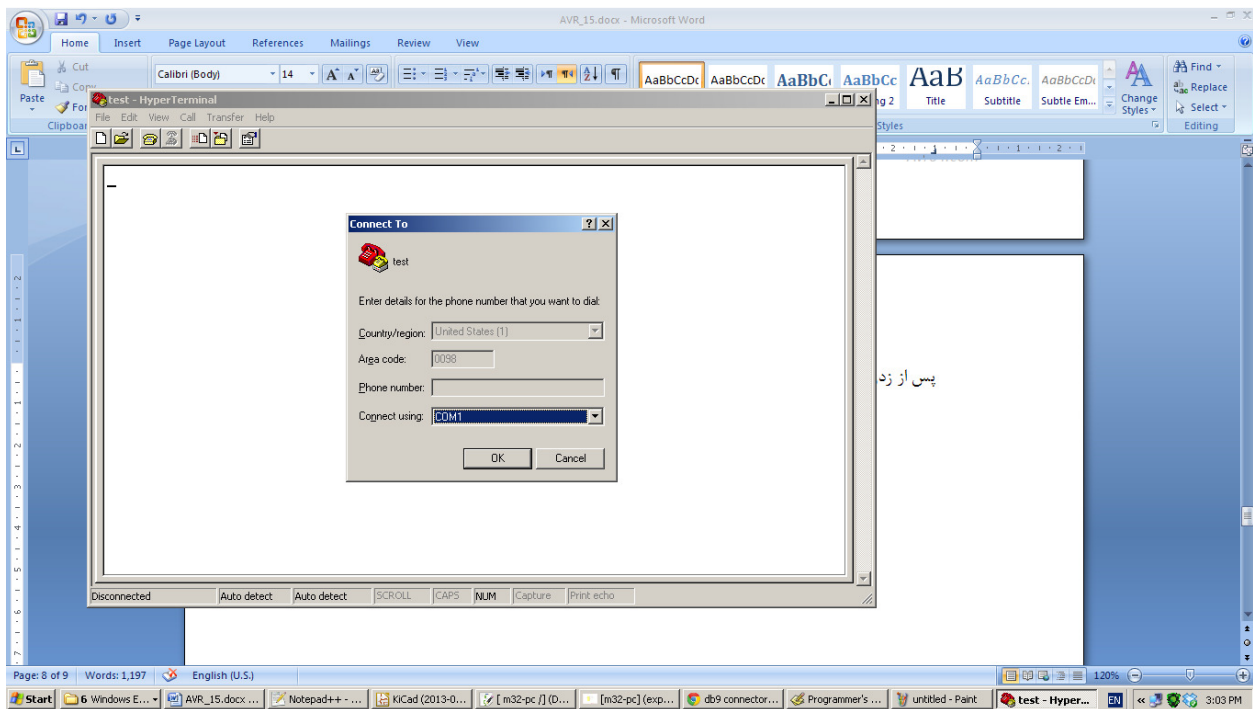
پس از کامپایل و پروگرام کردن میکرو، وارد Start شده و در Run تایپ کنید: Hypertrm



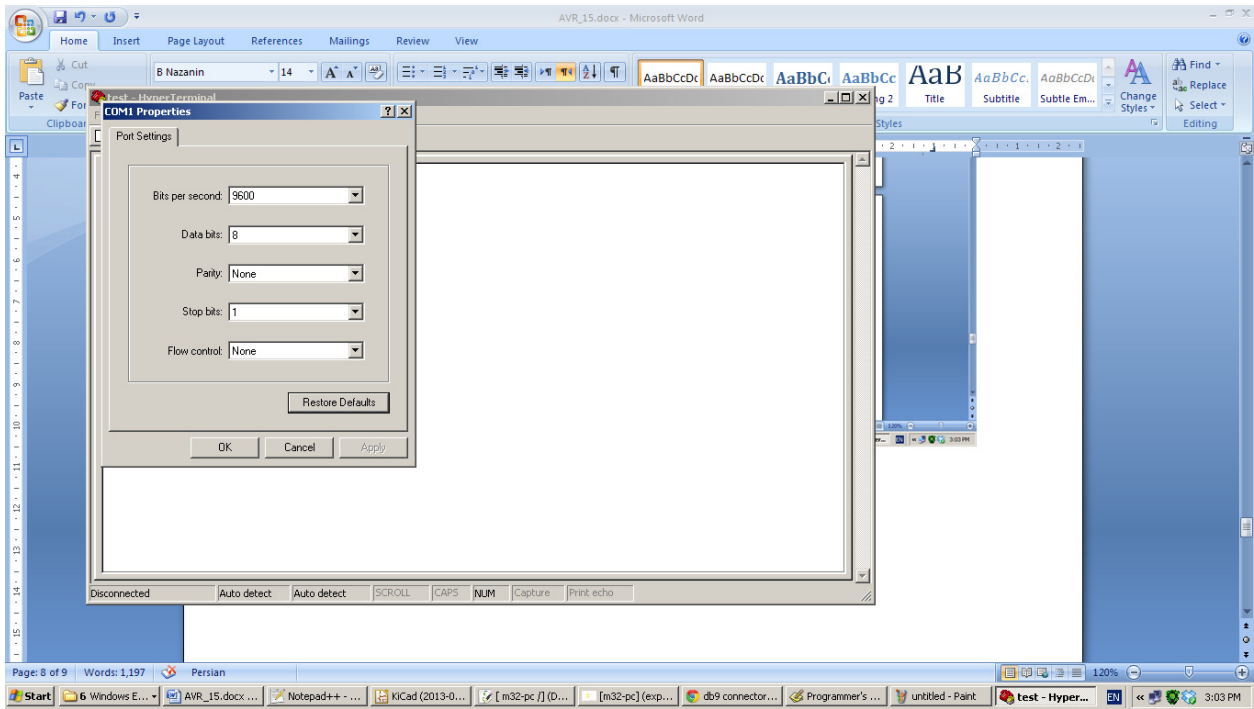
در پنجره بعدی یک نام دلخواه تایپ کنید:



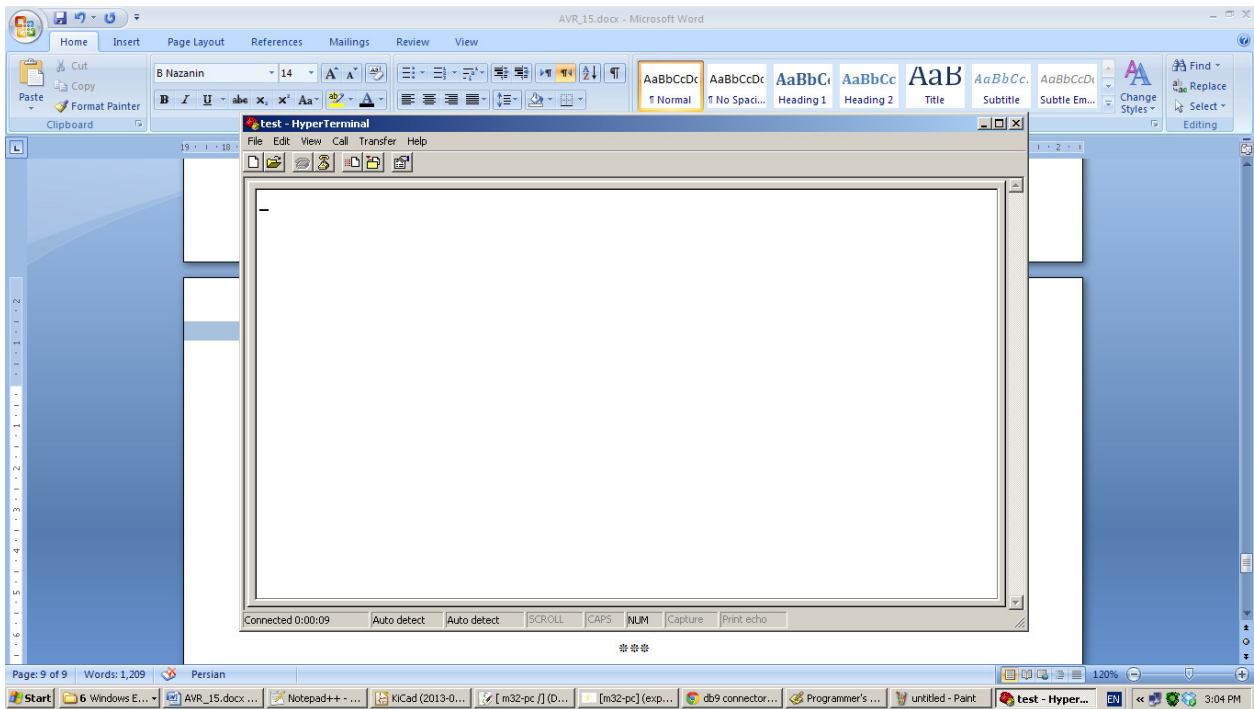
پس از زدن OK در پنجره جدید Com1 را انتخاب کرده و OK را بزنید:



در پنجره جدید روی Restore Defaults کلیک کنید و OK را بزنید:



پنجره ای مطابق زیر باز می شود:



هم اینک پورت Com1 کامپیوتر توسط هایپر ترمینال ویندوز با سرعت 9600 و به صورتی 8 بیتی بدون بیت توازن و یک بیت توقف باز شده است و آمده ارسال و دریافت کاراکتر می باشد. در این حالت پورت مادگی پروژه را به پشت case متصل کرده و تغذیه مدار را وصل کنید، مشاهده می کنید که عبارتی از طرف میکرو برای PC ارسال می شود. در این حالت اگر کلید q را بزنید LED روی برد روشن می شود و با زدن کلید w دیود نورانی خاموش می گردد. به این ترتیب ارتباط بین میکرو و کامپیوتر به وسیله پورت سریال برقرار شد.

در پروژه بعدی قصد داریم ارتباط بین دو میکرو را بررسی کنیم. در ارتباط بین دو میکرو معمولاً در فواصل کوتاه از آی سی MAX232 استفاده نمی شود و پایه های ارسال و دریافت دو آی سی میکرو به صورت معکوس به یکدیگر متصل می شوند. در پروژه بعدی یک میکروکنترلر ATmega16 کاراکتر های q و w را به ترتیب برای میکروی اول که همین پروژه جاری می باشد ارسال می کند و مشاهده می کنیم که LED به ترتیب روشن و خاموش می شود.

```

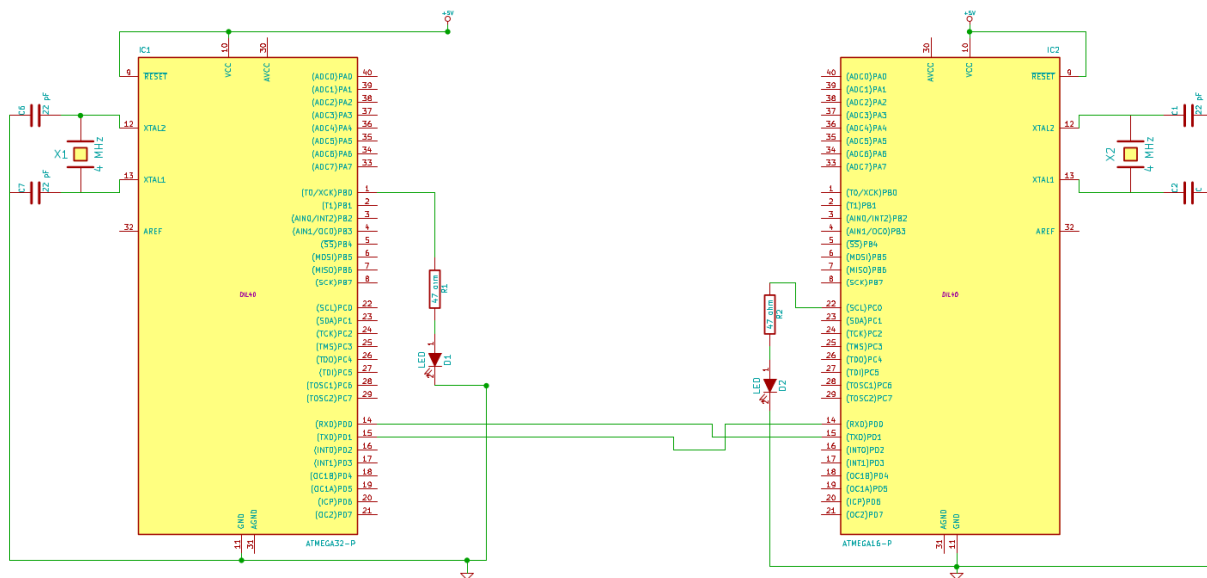
36
37
38
39 PORTB = 0b0000;
40
41 uart_init(UART1);
42 sei();
43 uart_puts("q");
44 //uart_puts_P("S");
45 //itoa(num, buff);
46 //uart_puts(buff);
47 //uart_putc("r");
48
49
50 _delay_ms(300);
51
52 PORTB = 0b0000;
53
54
55 _delay_ms(300);
56
57 while(1){
58
59
60
61 if (C &
62 {
63 /*
64 * no data avail
65 */

```

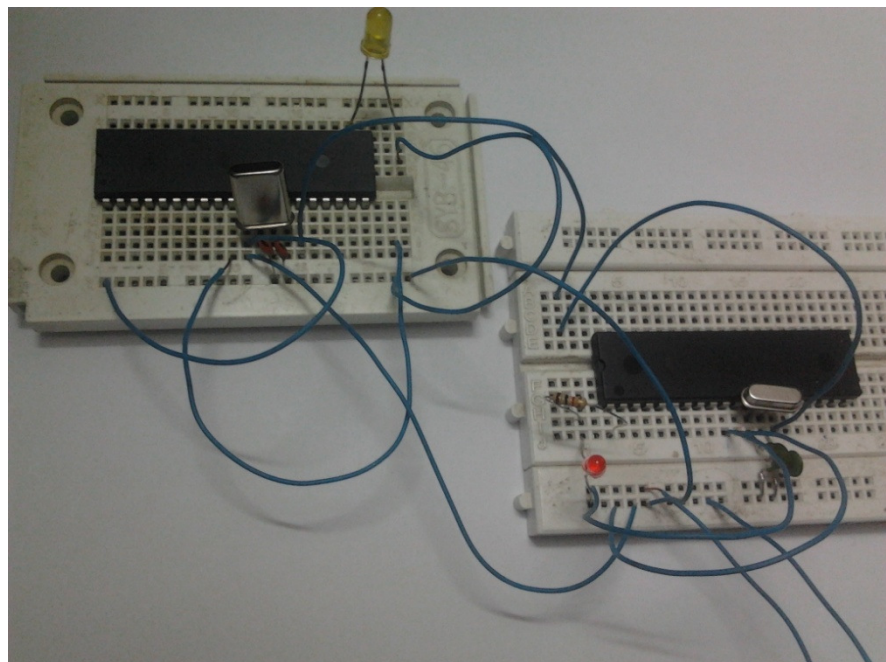
The screenshot shows a Notepad++ window with the following code and a HyperTerminal window overlaid on it. The HyperTerminal window displays the characters 'q' and 'w' received from the microcontroller. The status bar at the bottom of the HyperTerminal window shows 'Connected 0:03:38', 'Auto detect', '9600 8-N-1', 'SCROLL', 'CAPS', 'NUM', 'Capture', and 'Print echo'.

کدنویسی پروژه ارتباط دو میکرو با یکدیگر

برای ارتباط بین دو میکرو مداری مطابق شکل زیر بر روی دو برد جدا بنویسید، مدار پروژه قبلی را بدون تغییر نگه دارید و قسمت ATmega16 (سمت راست شماتیک زیر) را روی یک برد دیگر مونتاز نمایید:



چیدمان قطعات بر روی برد مورد به صورت زیر است:



برد کوچک سمت چپ که دارای بزرگ LED زرد می باشد همان میکروی سمت راست در شماتیک است. (فرستنده)

کدنویسی پروژه ارتباط دو میکرو با یکدیگر

در این قسمت فقط کد میکرو ATmega16 آورده شده است. میکرو مگا ۳۲ با همان برنامه قبلی مورد استفاده قرار می

گیرد. کد میکرو ای تی مگا ۱۶ در زیر آورده شده است:

```
#include <avr/io.h>
#include <avr/interrupt.h>
#include "rprintf.h"
#include "uart.h"
#include <util/delay.h>

#include <avr/pgmspace.h>
#include <stdlib.h>

/* define CPU frequency in Mhz here if not defined in Makefile */

#ifndef F_CPU
#define F_CPU 4000000UL
#endif

/* 9600 baud */
#define UART_BAUD_RATE 9600

int main(void)
{
    unsigned int c;
    //DDRB |= _BV(1);
    DDRC = 0xFF;
    PORTC = 0b00000001;

    _delay_ms(500);
    PORTC = 0b00000000;
    _delay_ms(500);

    uart_init( UART_BAUD_SELECT(UART_BAUD_RATE,F_CPU) );
    sei();

    PORTC = 0b00000001;

    uart_puts("Send from Micro...\n");

    while(1){
        _delay_ms(500);
        uart_putc('q');
        PORTC = 0b00000001;
        _delay_ms(500);
        uart_putc('w');
        PORTC = 0b00000000;
    }

    return 0;
}
```

سورس این پروژه نیز در پوشه این مقاله به نام ATmega16-src آورده شده است. پس از کامپایل و پروگرام کردن میکرو و

اتصال تغذیه مشاهده می کنید که هر دو دیود LED روشن و خاموش می شوند و در واقع میکرو ATmega16 میکرو Mega32

را کنترل می کند.

ادامه دارد...

