**AVR UART**

**Example Code**

You can also find the code on [pastebin](http://pastebin.com/7kfELPgv), and the [AVR code gallery](https://github.com/maxEmbedded/avr-code-gallery/tree/master/USART%20of%20AVR). The AVR code gallery contains the LCD files as well. To learn how to write, build, compile and execute your code using Atmel Studio 6, view [this](http://maxembedded.com/2012/06/25/using-atmel-studio-6/) tutorial.

[?](http://maxembedded.com/2013/09/the-usart-of-the-avr/)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57 | #include <avr/io.h>  #include <util/delay.h>  #include <stdlib.h>  #include "lcd.h"                            // include LCD library    #define BAUD 9600                           // define baud  #define BAUDRATE ((F\_CPU)/(BAUD\*16UL)-1)    // set baudrate value for UBRR    #ifndef F\_CPU  #define F\_CPU 16000000UL                    // set the CPU clock  #endif    // function to initialize UART  void uart\_init (void)  {      UBRRH=(BAUDRATE>>8);      UBRRL=BAUDRATE;                         //set baud rate      UCSRB|=(1<<TXEN)|(1<<RXEN);           //enable receiver and transmitter      UCSRC|=(1<<URSEL)|(1<<UCSZ0)|(1<<UCSZ1);// 8bit data format  }    // function to send data - NOT REQUIRED FOR THIS PROGRAM IMPLEMENTATION  void uart\_transmit (unsigned char data)  {      while (!( UCSRA & (1<<UDRE)));            // wait while register is free      UDR = data;                             // load data in the register  }    // function to receive data  unsigned char uart\_recieve (void)  {      while(!(UCSRA) & (1<<RXC));       // wait while data is being received      return UDR;                             // return 8-bit data  }    // main function: entry point of program  int main (void)  {      unsigned char a;      char buffer[10];        uart\_init();                            // initialize UART      lcd\_init(LCD\_DISP\_ON\_CURSOR);           // initialize LCD      lcd\_home();                             // goto LCD Home        while(1)      {          a=uart\_recieve();                   // save the received data in a variable          itoa(a,buffer,10);                  // convert numerals into string          lcd\_clrscr();                       // LCD clear screen          lcd\_home();                         // goto LCD home          lcd\_puts(buffer);                   // display the received value on LCD          \_delay\_ms(100);                     // wait before next attempt      }        return 0;  } |