Document: Datasheet Date: 21-Feb-13 Model #: 1166 Product's Page: www.sunrom.com/p-710.html

Ultrasonic Distance Sensor – Serial Out

Its compact size, higher range and easy usability make it a handy sensor for distance measurement and mapping.

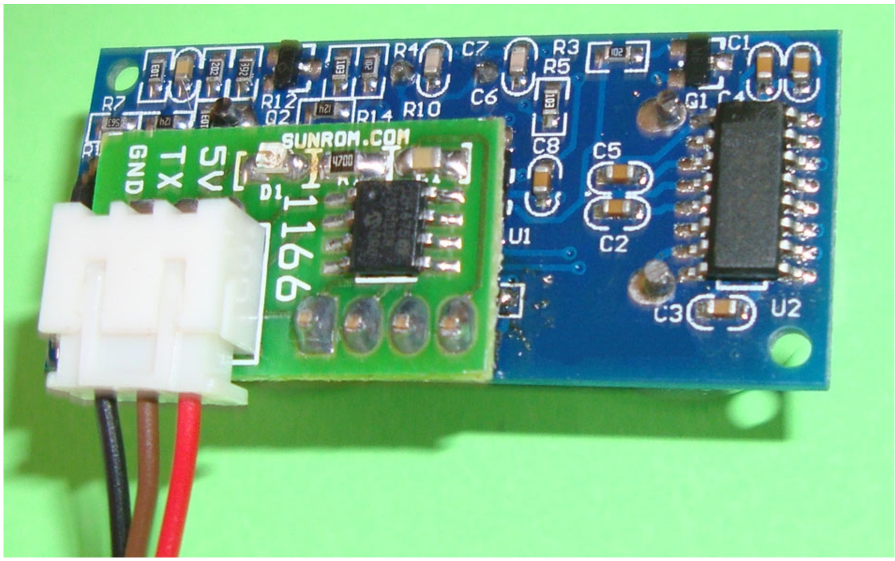
Features

- Minimum range 10 centimeters
- Maximum range 400 centimeters (4 Meters)
- Accuracy of +-1 cm
- Resolution 1 cm
- 5V DC Supply voltage
- Compact sized SMD design
- Modulated at 40 kHz
- Serial data of 9600 bps TTL level output for easy interface with any microcontroller

Specification

Parameter	Value	Unit
Supply Voltage	5	V
Supply Current	15	mA
Output Data speed	9600	Bps
Output Data Format	8-N-1	8 data bytes, no parity, 1 stop bit

Pin Details



Pin	Value
GND	Supply Ground
TX-OUT	Serial output data (TTL 5V level) at 9600 baud rate
+5V	Supply +5V

Output format

The serial output data consist of nine ASCII bytes as per table below

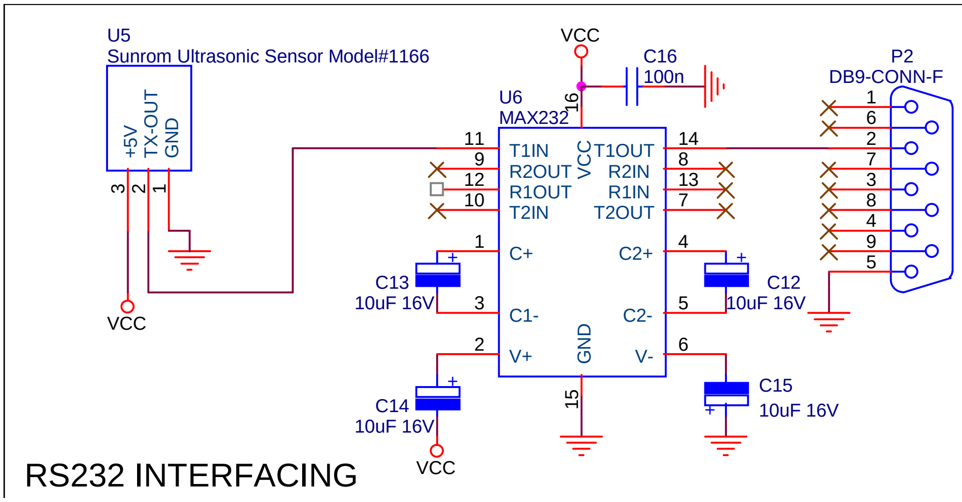
Sample outputs strings
100.00cm
080.01cm
075.96cm
010.56cm

xxx.xxcm<CR>								
‘0’ to ‘9’	‘0’ to ‘9’	‘0’ to ‘9’	.	‘0’ to ‘9’	‘0’ to ‘9’	‘c’	‘m’	<CR>
ASCII	ASCII	ASCII	Fixed Decimal	ASCII	ASCII	Fixed ASCII character	Fixed ASCII character	New line

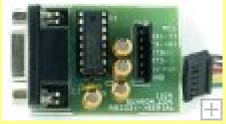
Position	**ASCII character**	**HEX**	**DEC**	**Description**
9	‘0’ to ‘9’	0x30 to 0x39	48 to 57	**X**xx.xxcm Hundred character
8	‘0’ to ‘9’	0x30 to 0x39	48 to 57	x**X**x.xxcm Tens character
7	‘0’ to ‘9’	0x30 to 0x39	48 to 57	xx**X**.xxcm Ones character
6	‘.’	0x2E	46	xxx.xxcm Decimal character
5	‘0’ to ‘9’	0x30 to 0x39	48 to 57	xxx.**X**xcm character
4	‘0’ to ‘9’	0x30 to 0x39	48 to 57	xxx.x**X**cm character
3	‘c’	0x63	99	‘c’ character
2	‘m’	0x6D	109	‘m’ character
1	<CR>	0x0D	13	New Line character

Interfacing with RS232

If you wish to interface the module with RS232 level like a PC serial port or any other device you need a level convertor such as MAX232 as shown below.

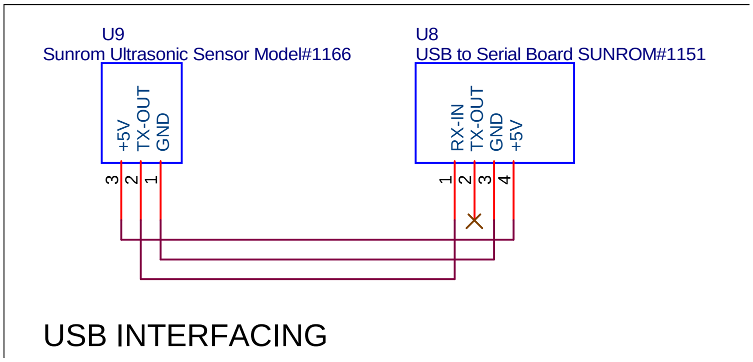


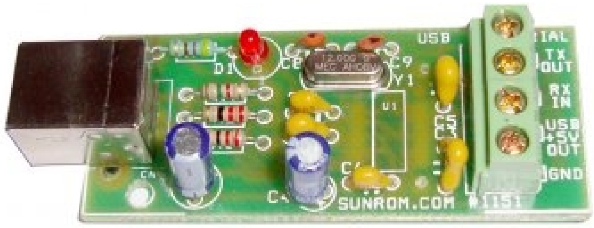
You can also use our Max232 Board Model 1104



http://www.sunrom.com/p-245.html

Interfacing to USB Port and Powering from USB Port



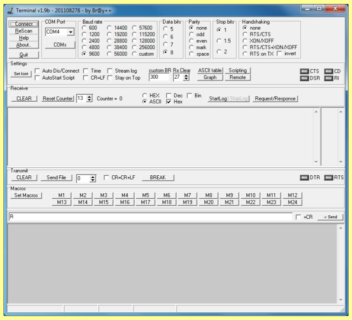


It will appear as virtual serial port on PC to which you can communicate through any software which can transmit receive by this serial port like hyperterminal or custom made software.

You can use our USB to Serial Board Model 1151

http://www.sunrom.com/p-244.html

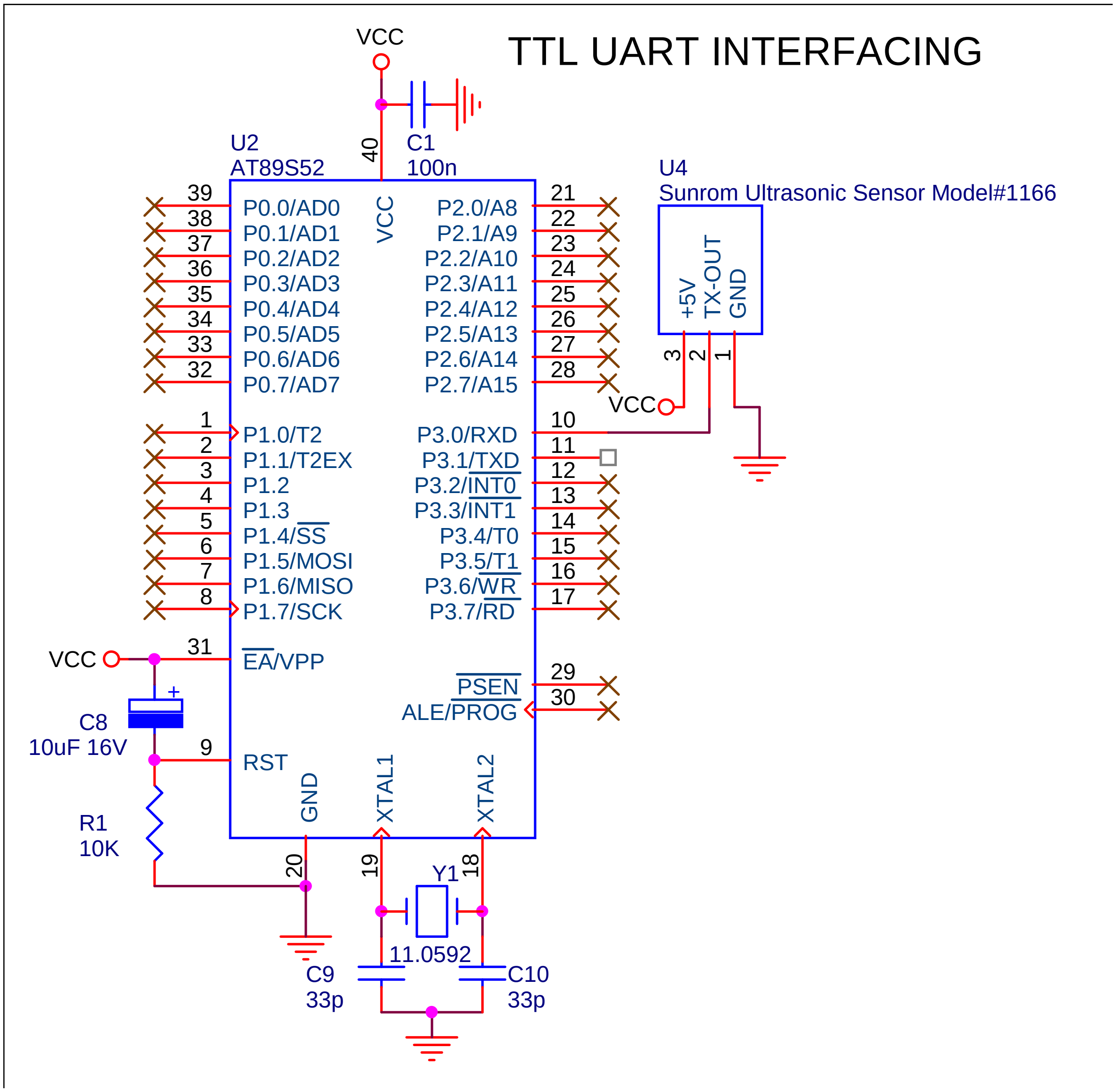
We recommend this terminal software on PC to view data



Download it from this link
http://www.sunrom.com/files/Terminal.exe

Application for reading through microcontroller

Connect Serial Out pin to RX pin of microcontroller and set the microcontroller to receive serial data at 9600 baud rate.



Sample code for AT89S52 microcontroller working at 11.0592 Mhz crystal and reading serial data and displaying to LCD.

Following sample code shows how to read the serial data and use the serial data to display value on LCD and convert the serial data to integer.

Code is in C and can be adapted to any compiler after minor modifications.

```
#include <REGX51.H> // standard 8051 defines
#include <stdio.h> // printf
// -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
// -=-=-=-=- Include files -=-=-=-=-=-=-=-
// -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
#include "lcd.h"  //This LCD part you will have to create as per your hardware
```

```
#include "utils.h" // This file only contains delay, you have to create delay function
as per your hardware crystal.

// -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
// -=-=-=-=- Hardware Defines -=-=-=-=-=-=-=
// -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
sbit BUZZER=P1^3;
sbit RELAY = P1^5;

char sbuf[10], c, length;
char buf[16];
unsigned char pos;
int range;

//receive serial character from serial port
char mygetchar(void)
{
      char c;
      while(!RI);
      RI =0;
      c = SBUF;
      // myputchar(c); // echo to terminal
      return SBUF;
}

//convert serial buffer to integer range
void convert()
{
range = (sbuf[0]-0x30)*100;
range = range+ (sbuf[1]-0x30)*10;
range = range+ (sbuf[2]-0x30)*1;
//range
}
// -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
// -=-=-=-=- Main Program -=-=-=-=-=-=-=
// -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
void main()
{
      // -=-=- Intialize variables -=-=-=
      BUZZER = 0;
      RELAY = 0;

      SCON = 0x52; // 8-bit UART mode
      TMOD = 0x20; // timer 1 mode 2 auto reload
      TH1= 0xfd; // 9600 8-n-1
      TR1 = 1; // run timer1
      // -=-=- Intialise -=-=-=
      lcdInit();

      // -=-=- Welcome LCD Message -=-=-= lcdClear();
      lcdGotoXY(0,0); // 1st Line of LCD
      // "xxxxxxxxxxxxxxxx"
      lcdPrint("WELCOME TO ");
      lcdGotoXY(0,1); // 2nd Line of LCD
      // "xxxxxxxxxxxxxxxx"
      lcdPrint("COLLISION SYSTEM");
      delayms(5000); // 5 sec
      pos = 0;
      length = 0;
```

```c
// -=-=- Program Loop -=-=-=
while(1)
{
      c = mygetchar(); //loop till character received
      if(c==0x0D) // if received character is <CR> end of line, time to display
      {
            length = pos;
            pos = 0;

            convert(); // convert serial buffer to integer

            lcdClear();
            lcdGotoXY(0,0); // 1st Line of LCD

            sprintf (buf, "Range: %c%c%c%c%c%c cm", sbuf[0], sbuf[1], sbuf[2],
            sbuf[3], sbuf[4], sbuf[5]);

            lcdPrint(buf);
            if(range < 20) // we check if range is less than 20
            {
                  RELAY = 1;
                  lcdGotoXY(0,1); // 2nd Line of LCD
                  lcdPrint("Collision!!!!");
            } else
            {
                  RELAY = 0;
                  lcdGotoXY(0,1); // 2nd Line of LCD
                  lcdPrint("Preset:20");
                  //sprintf (buf, "Collision at:%d", 20);
                  //sprintf (buf, "Range: %d", range);
                  //lcdPrint(buf);
            }
      } else { //store serial data to buffer
                  sbuf[pos] = c;
                  pos++;
            }
   } // end while
}// end main
```