Weeder Technologies

Pro-Kit

PO Box 2426, Ft Walton Beach, FL 32549

850-863-5723

RS-232 Analog Input

By Terry J. Weeder

Nearly every facet of our day to day life, in some way or another, is greatly dependent upon our ability to sense conditions in the physical environment. We obtain information regarding temperature and pressure via the nerve endings under our skin, light with our eyes, sounds with our ears, smells, etc. Regardless of the computing power of our brains, life's management would be very difficult, to say the least, if it not for the data provided by these human sensors. In the same respect, there are many computer related applications which are absolutely dependent upon data input reflecting this same physical environment. Albeit, data acquisition, or various forms of closed-loop controlling systems.

Many component-level electronic sensors are available which meet these needs. Discreet devices that measure temperature, pressure, and of course, mechanical position such as a potentiometer, are among a few. Unfortunately, the data output from most of these sensors is in the form of an analog voltage which is not compatible with the binary "1"s and "0"s used in the computer world.

To pass analog voltage data to a digital environment such as a PC, some form of conversion is necessary. The RS-232 Analog Input kit described here uses a self-calibrating 12-bit plus sign A/D converter which measures the voltage levels at its input pins and converts them to a number in the range of 0 to 4095, represented in 1 mV steps. Eight input pins are available and can be set up for either single-ended conversion - read the voltage potential between each pin and a common return, or differential conversion - read the difference in voltage between pin pairs. Because the result is scaled to 1 mV steps, further adjustments or math calculations are not required to yield the direct voltage measured at an input pin. For instance, a 2.499 volt input will be converted to the number 2499.

A powerful feature of this circuit is its ability to store alarm trip-point levels for each of its eight inputs. Even during times when the host computer is not requesting conversions, the RS-232 Analog Input module is continually scanning its input pins and reports if any pass over their user defined trip points. These trip points can be changed on the fly by the host. This is ideally suited for applications where a hundred or so sensors need to be monitored for over-range conditions without tying up the communications line/port.

Due to the special design of the RS-232 interface being used, the module can share the same port with others of its kind. A DIP switch setting on board allocates its own unique address, and allows up to 16 units to be stacked together to provide a total of 128 single-ended inputs, or 64 differential inputs.

The set of commands which can be sent from your PC's com port to initiate a conversion or set an alarm trip point is shown in listing 1. For versatility, each input pin can be read individually in either single-ended or differential mode, or all pins can be read as a group. The later being convenient for continuous data acquisition applications. There is also a command which will instruct the A/D converter to initiate its auto-zero function. This is useful when the RS-232 Analog Input kit is located in an environment subject to broad ambient temperature changes which may cause an offset error.

Circuit Theory

Refer to the schematic diagram shown in figure 1. The key component is the ADC12138 (IC2), a self-calibrating 12-bit plus sign A/D converter manufactured by National Semiconductor. This ADC features an 8 channel multiplexer and can be configured to operate in various combinations of single-ended, differential, or pseudo-differential modes. Control is accomplished via a 3-wire serial I/O interface which makes it ideal for a microcontroller based project such as this. Of the three wires there is a digital out (DO), a digital in (DI) and a serial clock pin (SCLK). The end of conversion pin (EOC) is used to indicate when the ADC is ready to receive another command. IC2 uses a precision voltage reference set by an LM4040-4.1 (D1). Using a 4.096 volt reference such as this, allows the 12-bit result to be scaled to 1 mV steps. Filter capacitors C6 thru C11 provide bypassing of noise spikes on the power supply and reference pins.

The intelligence in the circuit is provided by a PIC16C54 (IC1), an EPROM-based 8-bit CMOS microcontroller manufactured by Microchip. The crystal (XTAL1) sets the microcontroller's internal oscillator at 4 MHz, which is also used to supply the conversion clock (CCLK) required by IC2. IC1 has the responsibility of communicating with the A/D converter - instructing it to begin a conversion on a specified input pin then reading the result, or telling it to perform a function such as auto-calibrate or auto-zero. This is initiated with a set of commands transferred via the 3-wire interface mentioned above. Resistor R7 is required to hold the SCLK pin low during the microcontroller's 18 ms Power-On Reset delay when its corresponding driving pin is still in a high impedance state.

IC1 also handles the communications to and from the RS-232 serial port of the host computer. The voltage levels defined for serial communications on an RS-232 interface are +3V to +25V for a logic 0, and -3V to -25V for a logic 1. Most RS-232 devices use +12V and -12V respectively. Bit 2 of port-B is used to send data to the serial port. A logic 1 is generated by placing bit 2 at a high level which turns off Q1, thus allowing the -12V from the TD (Transmit Data) pin to be applied to the RD (Receive Data) pin thru R2. Bit 2 is sent low to produce a logic 0 which turns on Q1, pulling the RD pin to +5V. Because the TD pin of an RS-232 port is normally at a marking level (-12V), it is possible to "steal" from it the negative voltage needed for communications at RS-232 levels and a separate supply or negative charge pump is not required. Bit 0 of port-B is tied to the DTR (Data Terminal Ready) pin thru R5 and determines when the unit is plugged into an active RS-232 port. Bit 1 of port-B is tied to the RD pin thru R4 and is used to verify an idle RS-232 state prior to sending any serial data. This will avoid a collision with the data sent from any other projects which are sharing the same RS-232 port.

Command Set

TITLE	COMMAND	DESCRIPTION
READ	R pin	Perform conversion on a single input pin. (pin = 1-8 or A-D). Returns -4095 to +4095.
SINGLE-ENDED	S	Perform single-ended conversion on input pins 1-8. Returns "result1 result2 result3 result4 result5 result6 result7 result8" (resultX = -4095 to +4095).
DIFFERENTIAL	D	Perform differential conversion on input pins A-D. Returns "resultA resultB resultC resultD" (resultX = -4095 to +4095).
ALARM	A pin {-}value	Set alarm trip-point on an input pin. (pin = 1-8 or A-D, value = 0 to 4095). Returns "pinH" or "pinL" when transition of trip-point.
CLEAR	C	Clear all alarm trip-point settings.
AUTO-ZERO	Z	Perform Auto-Zero function on ADC.

Note, all commands must be preceeded by the WTADC header charcter A, and address character A-P. All responses from the WTADC will also contain this preface.

Listing 1

LED1 is used to indicate communications activity with the com port. The DIP switch (S1) together with the pull-down resistors R8 thru R11 set the address of the RS-232 Analog Input board. Power for the circuit is supplied from a standard wall transformer with an output in the range of 7 to 15 VDC. IC3, a 78L05 voltage regulator drops this voltage to the 5 volts required by the rest of the circuit. C1 and C2 stabilize the operation of the regulator and provide filtering.

The PIC Firmware

A PIC16C54 programmed with the "RS-232 Analog Input" firmware is available from the source mentioned in the parts list.

Upon power-up, the PIC sends a series of commands to the ADC to set its acquisition time, perform auto-calibrate and auto-zero. It then enters the main program loop which monitors the "TD" pin of the com port looking for a start bit. If a start bit is detected, the program jumps to a routine which gets the first character from the com port and checks to see if it is the "RS-232 Analog Input" header character "A". This will determine if the data to follow is meant for this kit or some other kit sharing the same port. Immediately following this header character is the address character (A-P). The PIC reads the DIP switch setting to determine if there is an address match and in turn either fetches or discards the remainder of the data stream.

The next character received is the "command" character and will indicate to the PIC the action being requested by the host. For a single-ended or differential conversion, the PIC sends the appropriate multiplexer addressing

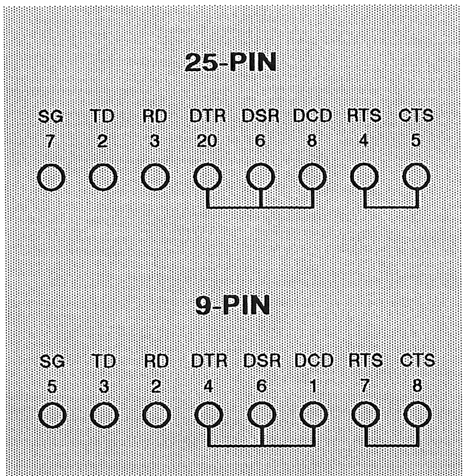


Figure 3

data to the ADC instructing it to initiate a conversion on the desired input pin. The result is returned to the PIC in the form of a 12-bit plus sign binary number. This number is converted to a decimal format represented in ascii characters, then transmitted to the host via the RS-232 interface. If the request was to read "all" input pins, the PIC repeats this conversion process for each pin, transmitting a "space" character between results.

If the "Alarm" command is received, the PIC reads the pin number which follows, and its new trip-point setting which can be in the range of 0 to plus or minus 4095 mV. This value is converted to a 13-bit binary number which includes the sign bit, and stored in two of sixteen 8-bit registers which have been set aside to hold the alarm trip-point settings for each input pin. The extra three bits in these registers are used to indicate "alarm active", "single or differential mode", and "current state" - above or below. After an alarm trip-point is set on any given input pin, the A/D conversion of that pin is included in the main program loop. Whenever the result of that pin's conversion is the opposite of its "current state", the PIC sends an indication to the host which consists of the pin number followed by the new state - "H" for high, or "L" for low. There is a built-in 10 mV hysteresis used with these alarm trip-points to prevent constant reporting when the voltage at an input pin is residing at the threshold.

When data is to be returned to the host, whether reporting an alarm trip or conversion result, the PIC first listens to the communications line to determine its availability. If or when it becomes idle, the header and address characters are sent followed by the response in the format shown in listing 1. This data stream is terminated with a carriage return and the program returns to the main loop.

Construction

The complete circuit fits nicely on a PC board measuring just under 3" x 2.5". Refer to the parts placement diagram shown in figure 2, identify the component side of the PC board which is marked and begin by soldering in the two IC sockets. Next, solder in the resistors and capacitors paying particular attention to the orientation of the polarized caps C1, C2, C7, C9 and C11. When mounting the LED, identify the anode which is the long lead; this should correspond with the pad labeled "A" on the parts placement diagram.

Care should be taken when soldering in the transistor (Q1), the regulator (IC3), and the voltage reference (D1) to avoid a solder bridge between the closely spaced pads. The crystal (XTAL1) should be mounted leaving a small gap between the bottom of its case and the PC board to avoid the chance of its metal case shorting the two pads together. Finish by mounting the DIP switch (S1).

Obtain a piece of 4-conductor telephone cord to be used as your RS-232 cable, and the appropriate connector to mate with your computer's serial port. Figure 3 shows the hook-up diagram for both a 9-pin and 25-pin RS-232 connector. Match the "SG", "DTR", "TD" and "RD" connections on the PC board with their corresponding pins on the connector you are using as shown in the diagram. Solder a jumper wire on the RS-232 connector between pins "RTS" and "CTS", and between pins "DTR", "DSR" and "DCD" as shown.

Use a DC wall transformer with an output in the range of 7 to 15 VDC. Cut off the connector at the end of the wires, use a voltmeter to determine positive and negative, and solder those wires to the "POS" and "NEG" connections on the PC board. After all components and wires have been soldered, closely examine both sides of the PC board for solder bridges and/or cold solder joints and re-solder if necessary. Carefully plug IC1 and IC2 into their sockets using the orientation as shown in the parts placement diagram.

```
CLS
OPEN "COM2:1200,N,8,1" FOR RANDOM AS #1
ON COM(2) GOSUB RECEIVE
COM(2) ON

DO
   KEY$ = INKEY$
   IF UCASE$(KEY$) = "C" THEN
      INPUT "Enter Command ", OUT$
      GOSUB TRANSMIT
   END IF
LOOP UNTIL KEY$ = CHR$(27)
CLOSE #1
END

RECEIVE:
   COM(2) OFF
   IF INPUT$(2, #1) = "AA" THEN
      LINE INPUT #1, IN$
      PRINT IN$
   ELSE LINE INPUT #1, DISCARD$
   END IF
   COM(2) ON
   RETURN

TRANSMIT:
   COM(2) OFF
   OUT$ = "AA" + OUT$
   PRINT #1, OUT$
   LINE INPUT #1, DISCARD$
   COM(2) ON
   RETURN
```

Listing 2

```
CLS
OPEN "COM2:1200,N,8,1" FOR RANDOM AS #1

DO
    KEY$ = INKEY$
    OUT$ = "S"
    GOSUB TRANSMIT
    GOSUB RECEIVE
    LOCATE 5, 1
    PRINT A%, B%, C%, D%, E%, F%, G%, H%
LOOP UNTIL KEY$ = CHR$(27)
CLOSE #1
END

RECEIVE:
    IF INPUT$(2, #1) = "AA" THEN
    INPUT #1, A%, B%, C%, D%, E%, F%, G%, H%
    ELSE LINE INPUT #1, DISCARD$
    END IF
    RETURN

TRANSMIT:
    OUT$ = "AA" + OUT$
    PRINT #1, OUT$
    LINE INPUT #1, DISCARD$
    RETURN
```

Listing 3

Operation

The RS-232 Analog Input kit can share the same serial port as other kits of its kind simply by wiring each kit in parallel on the same RS-232 cable. You must however, remove R1 and R2 on any subsequent kit (ie: of all the kits parallelled on the same line, only one kit should have R1 and R2 installed). Doing this will allow you to piggyback up to 16 analog input modules and use them on the same port which currently supports any of the other RS-232 kits offered by **Pro-Kit**.

To test out the circuit, hook up a 15-turn 1K potentiometer (available from Radio Shack) to each of the eight analog inputs in a voltage divider scheme as shown in figure 4. Plug the wall transformer into an electrical outlet, then plug the RS-232 connector into the com port of a PC. Note, always apply power to the analog input kit prior to plugging it into an active com port.

Use the simple QBASIC program shown in listing 2 to read the voltages contrived by the POT settings. Start the program then hit "C" on the keyboard. A prompt will appear asking you to type in a command. Type "R1" - using caps - then hit enter. The voltage seen at analog input pin #1 will be read and displayed on the computer monitor in millivolts. Typing in "R2", "R3", "R4", etc., in the same manner as above will read the voltages at each of the other input pins. Use pin numbers A thru D to read the "difference" in voltage potential between pin pairs as allocated in the schematic diagram. Depending on which pin of a pair is more positive then the other, will stipulate whether the result is returned as a positive or negative number.

To set an alarm trip point on an input pin, type in "A", the pin number, followed by the setting value in millivolts, then hit enter. To use a negative number, simply include the negative sign. Experiment by setting alarm trip points on various pins or pin pairs, then adjusting the POTs to transcend those points. When the voltage at an input pin exceeds its alarm trip-point, the pin number along with the letter "H" is echoed on the monitor. After the voltage returns to a level 10 mV less then the trip-point - used for hysteresis, the pin number along with the letter "L" is echoed. Keep in mind that setting trip points on pin pairs A thru D will cancel those settings on the corresponding individual pins 1, 3, 5 or 7, and vice versa - due to the sharing of registers in the PIC.

If you wish to use two broadly spaced high and low trip points on one sensor, simply connect two input pins to the same sensor and set their trip points to the high and low extremes. Remember to take into account the 10 mV hysteresis when defining the low trip point.

Listing 3 shows a simple QBASIC program which can be used for continuous data acquisition on all single-ended inputs at one time. To read the four differential inputs instead, change the "S" in line 5 to "D", and use four variables in lines 9 and 15. This mode of operation ties up the communications line/port, so if there are other kits sharing the same RS-232 line, you will have to insert a short delay (10 ms) in the program loop to give the other kits a chance to report their data; such as alarm trips on additional analog input boards, switch or button closures on a digital I/O board, Caller ID data,

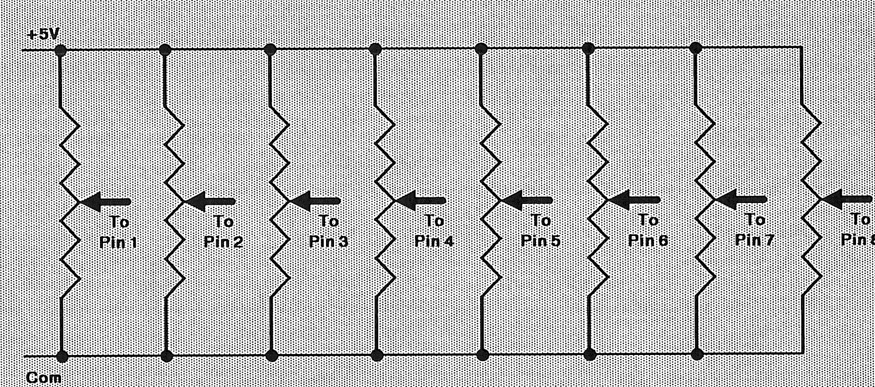


Figure 4

DTMF data, X-10 reception, etc.

Creating Your Own Program

The RS-232 Analog Input module communicates at 1200 baud, no parity, 8 data bits and 1 stop bit. Your program should contain the line OPEN "COM1:1200,N,8,1" FOR RANDOM AS #1, or similar. Also the ON COM GOSUB statement should be used as shown in the sample programs to handle branching to a subroutine when data is received from the module. All data sent by the analog input module to the serial port is preceded by the header character (A) and the board address character (A-P), then ending with a carriage return. All commands sent by the PC to the analog input module must also be preceded by these same header and address characters.

Two important notes here: Because of the structure of the RS-232 interface used by the analog input module, all characters that are sent to the module are also echoed back to the PC. Therefore your program must use the COM(1) OFF statement prior to using the PRINT #1, and then a LINE INPUT #1 statement to dump all echoed characters in the buffer before issuing a COM(1) ON statement. Also, always use the COM(1) OFF statement at the beginning of your subroutine branched to by the ON COM GOSUB statement, then a COM(1) ON statement at the end after all characters have been received. Failure to turn off event trapping as mentioned above will cause communications errors between the PC and the analog input module.

Parts List

Resistors (All are 1/4-watt, 10% units)

R1, R7, R8, R9, R10, R11 - 10,000 ohm

R2, R12 - 1000 ohm

R3, R4, R5 - 47,000 ohm

R6 - 620 ohm

Capacitors

C1 - 47 uF, 35-WVDC, electrolytic

C2 - 10 uF, 35-WVDC, electrolytic

C3, C4 - 15 pF, ceramic disc

C5, C6, C8, C10 - 0.1 uF monolithic ceramic

C7, C9, C11 - 10 uF, 10-WVDC, tantalum

Semiconductors

D1 - LM4040-4.1 precision 4.096-volt reference (National Semiconductor)

IC1 - PIC16C54-XT/P (pre-programmed) 8-bit microcontroller (Microchip)

IC2 - ADC12138CIN self-calibrating, 12-bit plus sign, A/D converter with 8-channel multiplexer (National Semiconductor)

IC3 - 78L05 low power 5-volt regulator

LED1 - light emitting diode, red

Q1 - 2N4403, general-purpose PNP silicon transistor

Other Components

S1 - DIP switch, 4-pole

XTAL1 - 4 MHz crystal

Miscellaneous: PC board, IC sockets, DC wall transformer, RS-232 connector & cable, analog sensors, solder, etc.

The following items are available from Weeder Technologies, PO Box 2426, Ft Walton Beach, FL 32549. 850-863-5723.

- Double-sided etched and drilled PC board (WTADC-B), $9.50
- All board mounting components including a pre-programmed PIC16C54 (WTADC-C), $39.50
- A pre-programmed PIC16C54 only (PIC-ADC), $16.00

All orders must include an additional $4 for shipping and handling. Florida residents add sales tax. Visa and MasterCard welcome.

RS–232 ANALOG INPUT

WTADC

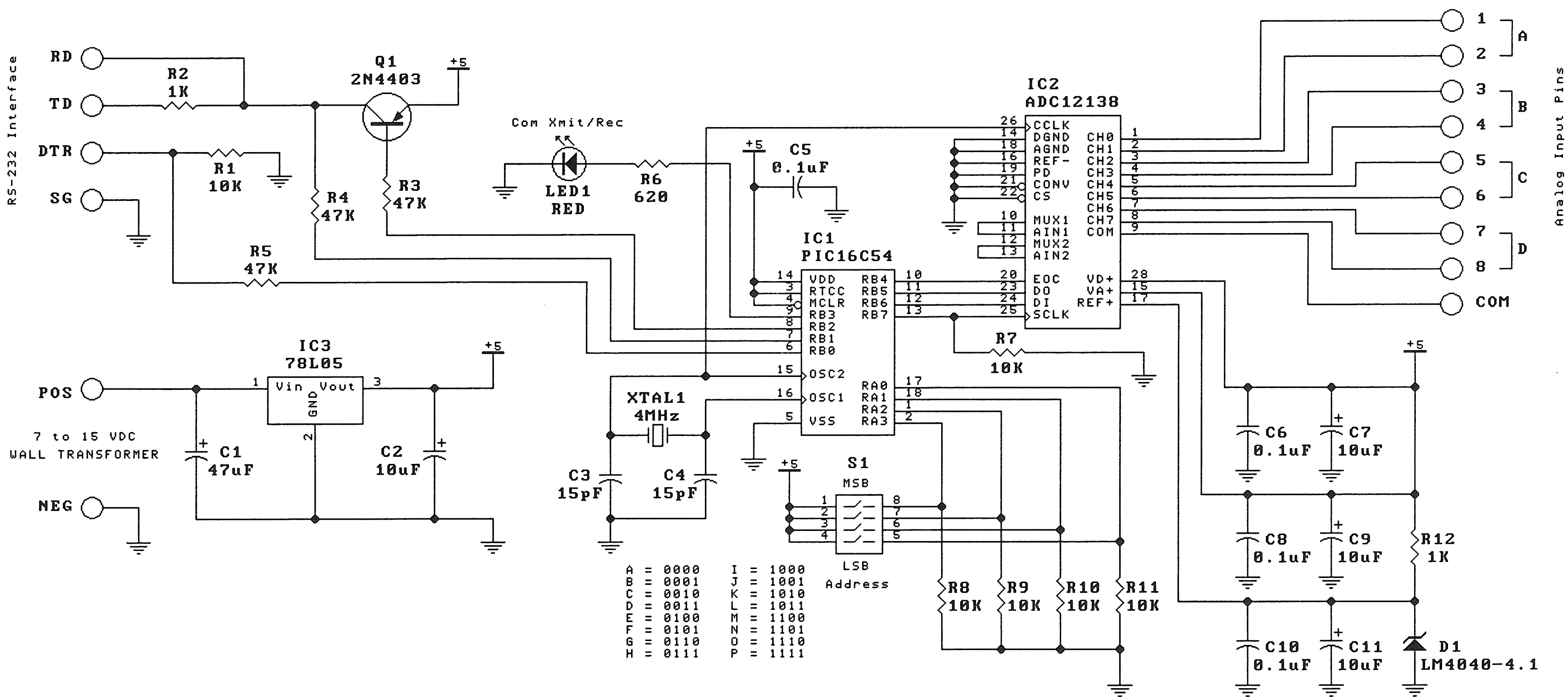
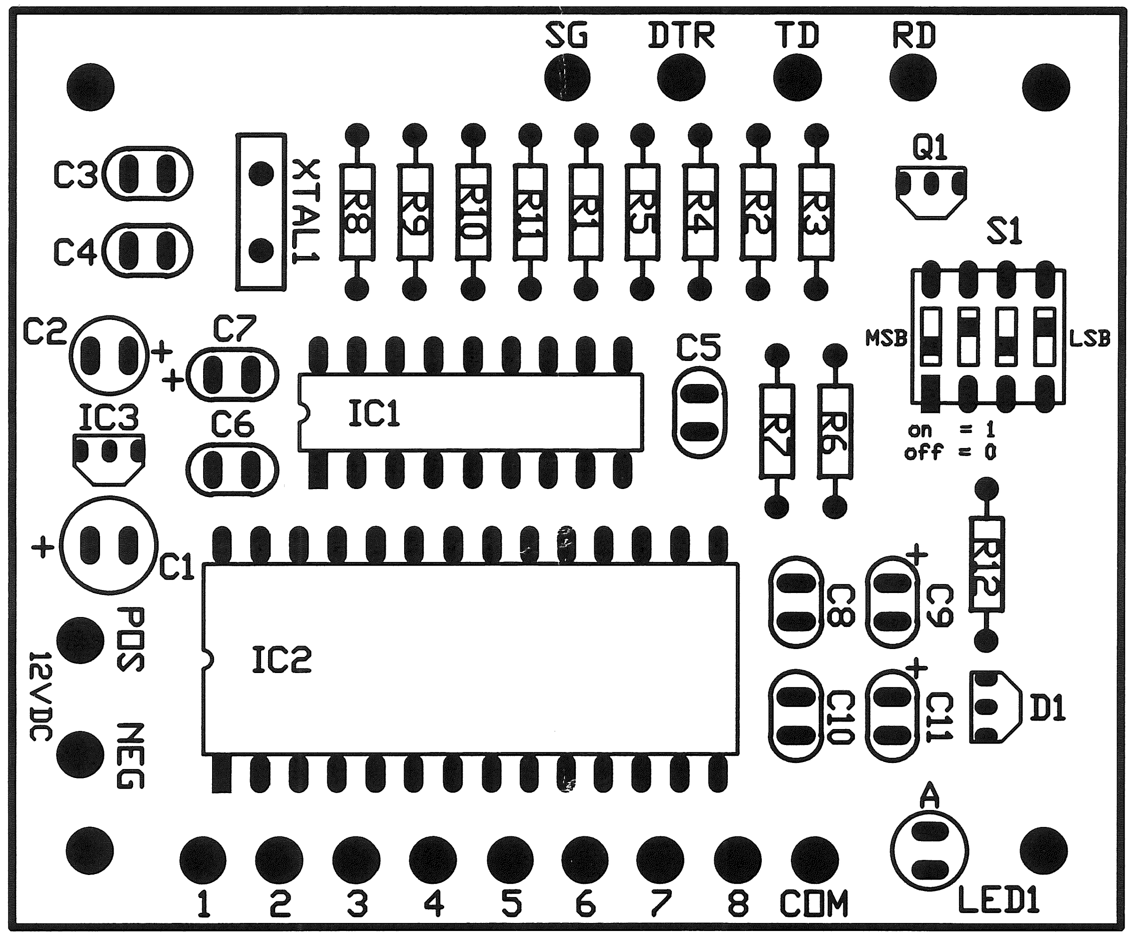


Figure 1



WTADC

Figure 2