AN611

Resistance and Capacitance Meter Using a PIC16C622

Author: *Rodger Richey*
Logic Products Division

INTRODUCTION

The PIC16C62X devices create a new branch in Microchip's PIC16CXX 8-bit microcontroller family by incorporating two analog comparators and a variable voltage reference on-chip. The comparators feature programmable input multiplexing from device inputs and an internal voltage reference. The internal voltage reference has two ranges, each capable of 16 distinct voltage levels. Typical applications such as appliance controllers or low-power remote sensors can now be implemented using fewer external components thus reducing cost and power consumption. The 18-pin SOIC or 20-pin SSOP packages are ideal for designs having size constraints.

The PIC16C62X family includes some familiar PIC16CXX features such as:

- 8-bit timer/counter with 8-bit prescaler
- PORTB interrupt on change
- 13 I/O pins
- Program and Data Memory

Device	Program Memory	Data Memory
PIC16C620	512 x 14	80 x 8
PIC16C621	1K x 14	80 x 8
PIC16C622	2K x 14	128 x 8

This family of devices also introduce on-chip brown-out detect circuitry and a filter on the reset input ($\overline{MCLR}$) to the PIC16CXX mid-range microcontrollers. Brown-out Detect holds the device in reset while V_{DD} is below the Brown-out Detect voltage of 4.0V, ± 0.2V. The reset filter is used to filter out glitches on the $\overline{MCLR}$ pin.

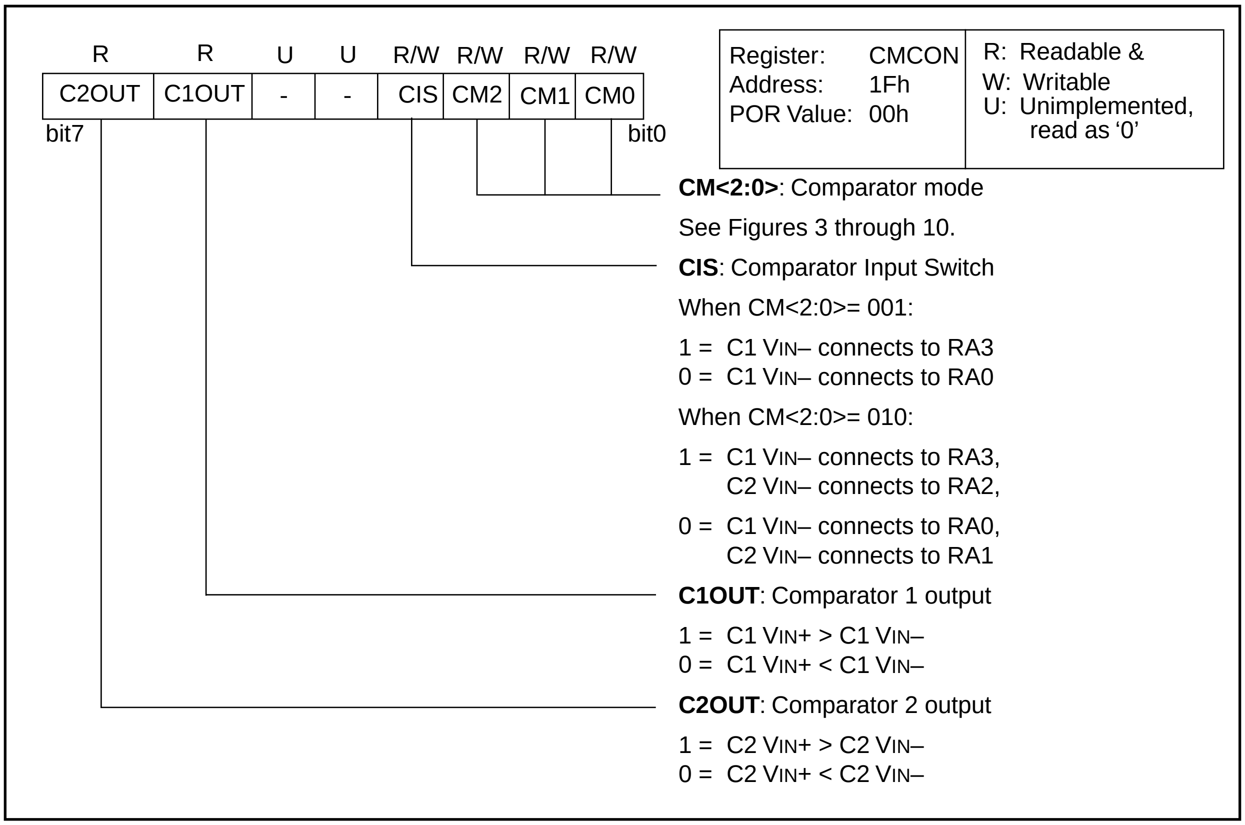
This application note will describe:

- Comparator module
  - operation
  - initialization
  - outputs
- Voltage Reference module
  - operation
  - initialization
  - outputs
- Linear slope integrating Analog to Digital conversion techniques
  - advantages
  - disadvantages
- Overview of the application circuit
- Detailed description of the measurement techniques used in the application circuit

COMPARATOR MODULE

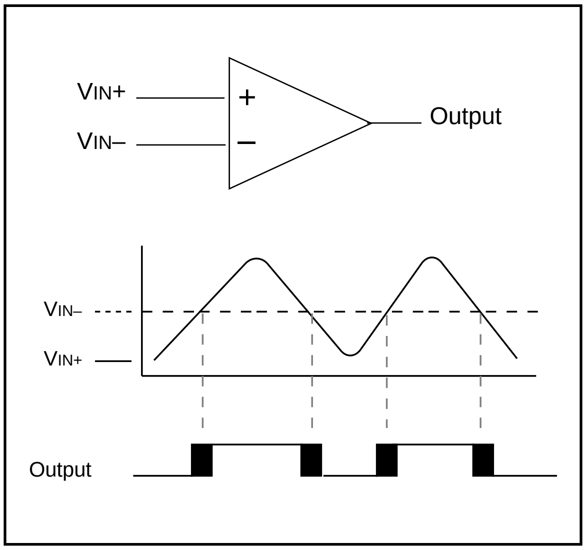
The comparator module contains two analog comparators with eight modes of operation. The inputs to the comparators are multiplexed with the RA0 through RA3 pins. The on-chip voltage reference can also be selected as an input to the comparators. The Comparator Control Register (CMCON) controls the operation of the comparator and contains the comparator output bits. Figure 1 shows the CMCON register.

FIGURE 1: CMCON REGISTER



A single comparator is shown in Figure 2. The relationship between the inputs and the output is also shown. When the voltage at VIN+ is less than the voltage at VIN-, the output of the comparator is at a digital low level. When the voltage at VIN+ is greater than the voltage at VIN-, the output of the comparator is at a digital high level. The shaded areas of the comparator output waveform represent the uncertainty due to input offsets and response time.

FIGURE 2: SINGLE COMPARATOR

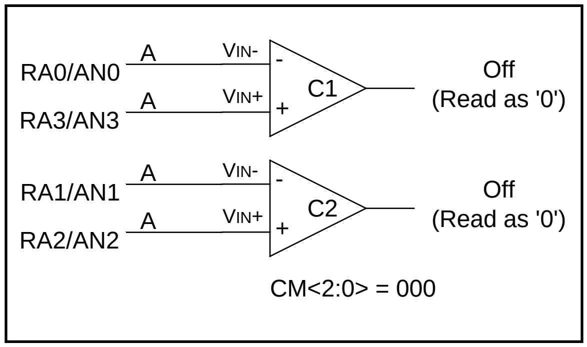


The TRISA register controls the I/O direction of the PORTA pins regardless of the comparator mode. If the comparator mode configures a pin as an analog input and the TRISA register configures that pin as an output, the contents of the PORTA data latch are placed on the pin. The value at the pin, which can be a digital high or low voltage, then becomes the input signal to the comparators. This technique is useful to check the functionality of the application circuit and the comparator module.

Comparator Operating Modes

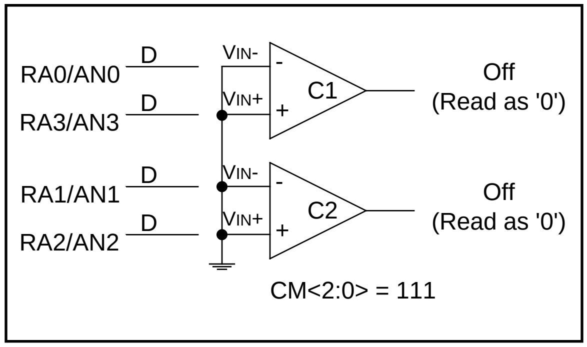
The analog inputs to the comparator module must be between VSS and VDD and one input must be in the Common Mode Range (CMR). The CMR is defined as VDD-1.5 volt to VSS. The output of a comparator will default to a high level if both inputs are outside of the CMR. If the input voltage deviates above VDD or below VSS by more than 0.6 volt, the microcontroller may draw excessive current. A maximum source impedance to the comparators of 10 kΩ is recommended. Figure 3 through Figure 10 show the eight modes of operation.

FIGURE 3: COMPARATORS RESET



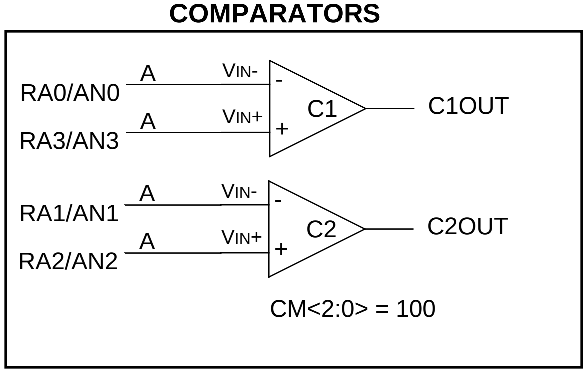
The Comparators Reset Mode (Figure 3) is considered the lowest power mode because the comparators are turned off and RA0 through RA3 are analog inputs. The comparator module defaults to this mode on Power-on Reset.

FIGURE 4: COMPARATORS OFF



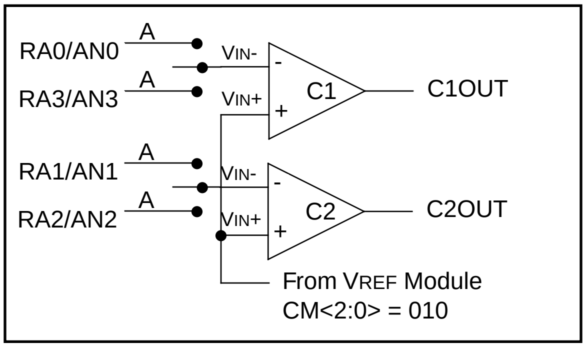
The Comparators Off Mode (Figure 4) is the same as the Comparators Reset Mode except that RA0 through RA3 are digital I/O. This mode may consume more current if RA0 through RA3 are configured as inputs and the pins are left floating.

FIGURE 5: TWO INDEPENDENT COMPARATORS



The Two Independent Comparators Mode (Figure 5) enables both comparators to operate independently.

FIGURE 6: FOUR INPUTS MULTIPLEXED TO TWO COMPARATORS

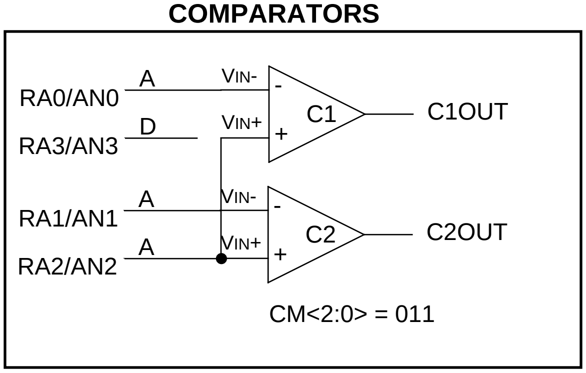


The Four Inputs Multiplexed to Two Comparators Mode (Figure 6) allows two inputs into the VIN- pin of each comparator. The internal voltage reference is connected to the VIN+ pin input of each comparator. The CIS bit, CMCON<3>, controls the input multiplexing to the VIN- pin of each comparator. Table 1 shows this relationship.

TABLE 1: COMPARATOR INPUT MULTIPLEXING

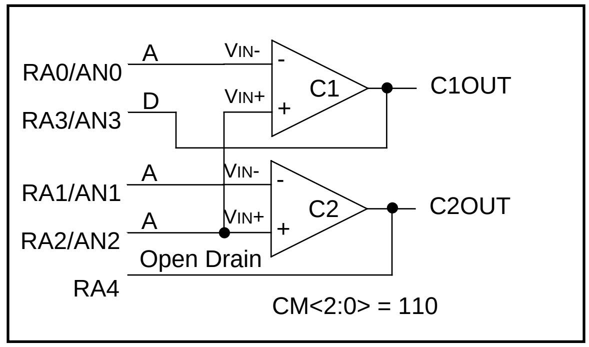
CIS	C1 VIN-	C2 VIN-
0	RA0	RA1
1	RA3	RA2

FIGURE 7: TWO COMMON REFERENCE COMPARATORS



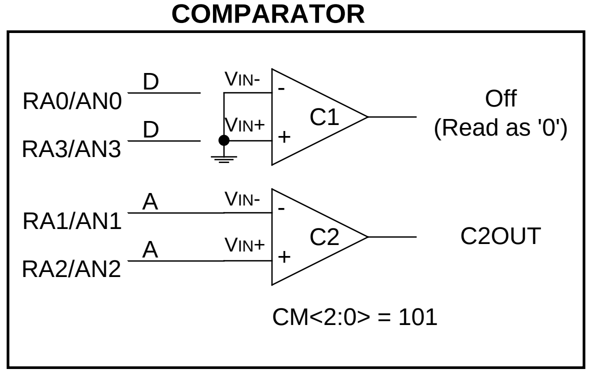
The Two Common Reference Comparators Mode (Figure 7) configures the comparators such that the signal present on RA2 is connected to the VIN+ pin of each comparator. RA3 is configured as a digital I/O pin.

FIGURE 8: TWO COMMON REFERENCE COMPARATORS WITH OUTPUTS



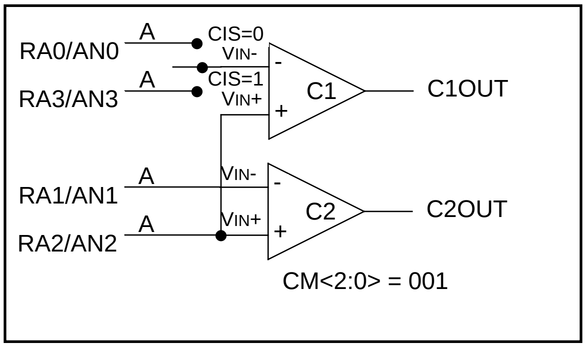
The Two Common Reference Comparators with Outputs Mode (Figure 8) connects the outputs of the comparators to an I/O pin. These outputs are digital outputs only with RA3 defined as a CMOS output and RA4 defined as an open drain output. RA4 requires a pull-up resistor to function properly. The value of resistance used for the pull-up will affect the response time of comparator C2. The signal present on RA2 is connected to the VIN+ pin of both comparators.

FIGURE 9: ONE INDEPENDENT COMPARATOR



The One Independent Comparator Mode (Figure 9) turns comparator C1 off making both RA0 and RA3 digital I/O. Comparator C2 is operational with analog inputs from RA1 and RA2.

FIGURE 10: THREE INPUTS MULTIPLEXED TO TWO COMPARATORS



The Three Inputs Multiplexed to Two Comparators Mode (Figure 10) connects the VIN+ pin of each comparator to RA2. The VIN- pin of comparator 2 is connected to RA1. The CIS bit, CMCON<3>, controls the input to the VIN- pin of comparator 1. If CIS = 0, then RA0 is connected to the VIN- pin. Otherwise RA3 is connected to the VIN- pin of comparator 1.

Note:	Each comparator that is active will consume less power when the output is at a high level.

Clearing the Comparator Interrupt Flag

The comparator interrupt flag, CMIF, is located in the PIR1 register. This flag must be cleared after changing comparator modes. Whenever the comparator mode or the CIS bit is changed, the CMIF may be set due to the internal circuitry switching between modes. Therefore, comparator interrupts should be disabled before changing modes. Then, a delay of 10 µs should be used after changing modes to allow the comparator circuitry to stabilize.

The steps to clear the CMIF flag when changing modes are as follows:

- Change the comparator mode or CIS bit
- 10 µs delay
- Read the CMCON register to end the “mismatch” condition
- Clear the CMIF bit of the PIR1 register

The value of C1OUT and C2OUT are internally latched on every read of the CMCON register. The current values of C1OUT and C2OUT are compared with the latched values, and when these values are different a “mismatch” condition occurs. The CMIF interrupt flag will not be cleared if the CMCON register has not been read.

Using the Comparator Module

The CMCON register contains the comparator output bits C1OUT and C2OUT, CMCON<7:6>. These bits are read only. C1OUT and C2OUT follow the output of the comparators and are not synchronized to any internal clock edges. Therefore, the firmware will need to maintain the status of these output bits to determine the actual change that has occurred. The PIR1 register contains the comparator interrupt flag CMIF, PIR1<6>. The CMIF bit is set whenever there is a change in the output value of either comparator relative to the last time the CMCON register was read.

Note:	If a change in C1OUT or C2OUT should occur when a read operation on the CMCON register is being executed (start of the Q2 pcycle), the CMIF interrupt flag may not be set.

When reading the PORTA register, all pins configured as analog inputs will read as a '0'. Analog levels on any pin that is defined as a digital input may cause the input buffer to consume more current than is specified.

The code in Example 1 shows the steps required to configure the comparator module. RA3 and RA4 are configured as digital outputs. RA0 and RA1 are configured as the VIN- inputs to the comparators and RA2 is the VIN+ input to both comparators.

EXAMPLE 1: INITIALIZING THE COMPARATOR MODULE

```
CLRF    PORTA          ;init PORTA
MOVLW   0X03           ;Two Common
MOVWF   CMCON          ;Reference
                       ;Comparators
                       ;mode selected
BSF     STATUS,RP0     ;go to Bank 1
MOVLW   0X07           ;Set RA<2:0> as
MOVWF   TRISA          ;inputs,RA<4:3>
                       ;as outputs
BCF     STATUS,RP0     ;go to Bank 0
CALL    DELAY10        ;10µs delay
MOVF    CMCON,F        ;read the CMCON
BCF     PIR1,CMIF      ;clear the CMIF
BSF     STATUS,RP0     ;go to Bank 1
BSF     PIE1,CMIE      ;enable compar-
                       ;ator interrupt
BCF     STATUS,RP0     ;go to Bank 0
BSF     INTCON,PEIE    ;enable global
BSF     INTCON,GIE     ;and peripheral
                       ;interrupts
```

The comparators will remain active if the device is placed in sleep mode, except for the Comparators Off Mode (CM<2:0>=111) and Comparators Reset Mode (CM<2:0>=000). In these modes the comparators are turned off and are in a low power state. A comparator interrupt, if enabled, will wake-up the device from sleep in all modes except **Off** and **Reset**.

Comparator Timings

The comparator module has a response time and a mode change to output valid timing associated with it. The response time is defined as the time from when an input to the comparator changes until the output of that comparator becomes valid. The response time is faster when the output of the comparator transitions from a high level to a low level. The mode change to output valid time refers to the amount of time it takes for the output of the comparators to become valid after the mode has changed. The internal voltage reference may contribute some delay if used in conjunction with the comparators (see Voltage Reference Settling Time).

VOLTAGE REFERENCE MODULE

The voltage reference is a 16-tap resistor ladder network that is segmented to provide two ranges of VREF values. Each range has 16 distinct voltage levels. The voltage reference has a power-down function to conserve power when the reference is not being used. The voltage reference also has the capability to be connected to RA2 as an output. Figure 11 shows the Voltage Reference Control Register (VRCON) register which controls the voltage reference. Figure 12 shows the block diagram for the voltage reference module.

FIGURE 11: VRCON REGISTER

R/W	R/W	R/W	U	R/W	R/W	R/W	R/W
VREN	VROE	VRR	-	VR3	VR2	VR1	VR0

bit7 … bit0

Register: VRCON
Address: 9Fh
POR Value: 00h

R: Readable
W: Writable
U: Unimplemented, read as ‘0’

VR<3:0>: VREF value selection 0 ≤ VR [3:0] ≤ 15

when VRR = 1: VREF = (VR<3:0>/ 24) * VDD
when VRR = 0: VREF = 1/4 * VDD + (VR<3:0>/ 32 * VDD)

VRR: VREF Range selection

1 = Low Range
0 = High Range

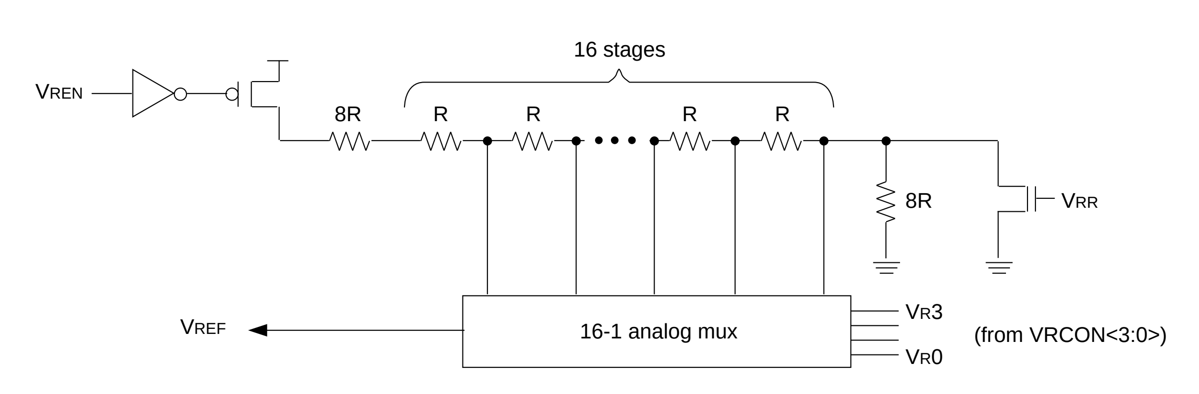
VROE: VREF Output Enable

1 = VREF is output on RA2 pin
0 = VREF is disconnected from RA2 pin

VREN: VREF Enable

1 = VREF circuit powered on
0 = VREF circuit powered down, no IDD drain

FIGURE 12: VOLTAGE REFERENCE BLOCK DIAGRAM



Note: The voltage reference is VDD derived and therefore, the VREF output changes with fluctuations in VDD.

Using the Voltage Reference

The voltage reference module operates independently of the comparator module. The output of the voltage reference may be connected to the RA2 pin at any time by setting the TRISA<2> bit and the VRCON<6> bit (V_{ROE}). It should be noted that enabling the voltage reference with an input signal present will increase current consumption. Configuring the RA2 pin as a digital output with the V_{REF} output enabled will also increase current consumption. The increases in current are caused by the voltage reference output conflicting with an input signal or the digital output. The amount of increased current consumption is dependent on the setting of V_{REF} and the value of the input signal or the digital output.

The full range of V_{SS} to V_{DD} cannot be realized due to the construction of the module (Figure 12). The transistors on the top and bottom of the resistor ladder network keep V_{REF} from approaching V_{SS} or V_{DD}. Equation 1 and Equation 2 are used to calculate the output of the voltage reference.

EQUATION 1: VOLTAGE REFERENCE EQUATION, V_{RR}=1

$$V_{REF}=(V_R\langle 3:0\rangle/24) \times V_{DD}$$

EQUATION 2: VOLTAGE REFERENCE EQUATION, V_{RR}=0

$$V_{REF}=(V_{DD}/4) + (V_R\langle 3:0\rangle/32) \times V_{DD}$$

An example of how to configure the voltage reference is given in Equation 2. The reference is set for an output voltage of 1.25V at a V_{DD} of 5.0V.

EXAMPLE 2: VOLTAGE REFERENCE CONFIGURATION

```
MOVLW   0X02         ;4 Inputs Muxed
MOVWF   CMCON        ;to 2 comps.
BSF     STATUS,RP0   ;go to Bank 1
MOVLW   0x07         ;RA3-RA0 are
MOVWF   TRISA        ;outputs
MOVLW   0XA6         ;enable VREF,
MOVWF   VRCON        ;low range
                     ;set VR<3:0>=6
BCF     STATUS,RP0   ;go to Bank 0
CALL    DELAY10      ;10µs delay
```

If the voltage reference is used with the comparator module, the following steps should be followed when making changes to the voltage reference.

1. Disable the comparator interrupts
2. Make changes to the voltage reference
3. Delay 10 µs to allow V_{REF} to stabilize
4. Delay 10 µs to allow comparators to settle
5. Clear the comparator interrupt flag
   – Read the CMCON register
   – Clear the CMIF bit
6. Enable comparator interrupts

The output of the voltage reference may be used as a simple DAC. However, the V_{REF} output has limited drive capability when connected to the RA2 pin. In fact the amount of drive the voltage reference can provide is dependent on the setting of the tap on the resistor ladder. If V_{REF} is used as an output, an external buffer must be utilized.

Voltage Reference Settling Time

Settling time of the voltage reference is defined as the time it takes the output voltage to settle within 1/4 LSB after making a change to the reference. The changes include adjusting the tap position on the resistor ladder, enabling the output, and enabling the reference itself. If the voltage reference is used with the comparator module, the settling time must be considered.

MAKING SIMPLE A/D CONVERSIONS

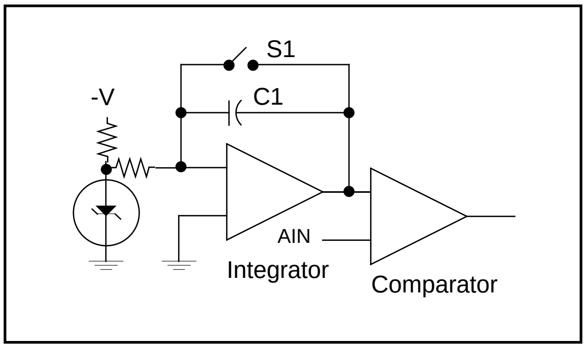
Linear slope integrating A/D converters are very simple to implement and can achieve high linearity and resolution for low conversion rates. The three types of converters that will be discussed are the single-slope, dual-slope, and modified single-slope converters. The following material was referenced from application note AN260, *"A 20-Bit (1ppm) Linear Slope-Integrating A/D Converter"*, found in the Linear Applications Handbook from National Semiconductor®.

Single-Slope Integrating Converter

A single-slope integrating converter is shown in Figure 13. In a single-slope converter, a linear ramp is compared against an unknown input AIN. When the switch S1 is opened the ramp begins. The time interval between the opening of the switch and the comparator changing state is proportional to the value of AIN.

The basic assumptions are that the integrating capacitor C1 and the clock used to measure the time interval remain constant over time and temperature. This type of converter is heavily dependent on the stability of the integrating capacitor.

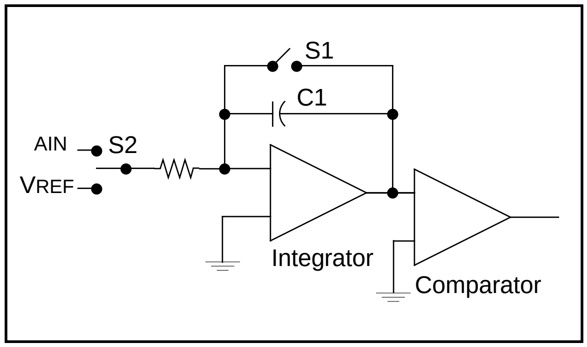
FIGURE 13: SINGLE-SLOPE INTEGRATING CONVERTER



Dual-Slope Integrating Converter

Figure 14 shows a dual-slope integrating converter. The dual-slope converter integrates the AIN input for a predetermined length of time. The voltage reference is then switched into the integrator input, using S2, which integrates in a negative direction from the AIN slope. The length of time the reference slope requires to return to zero is proportional to the value of AIN. Both slopes are made with the same integrating capacitor C1 and measured with the same clock, so they need only to be stable over one conversion cycle.

FIGURE 14: DUAL-SLOPE INTEGRATING CONVERTER



The dual-slope converter essentially removes the stability factor of the integrating capacitor from a conversion, however, the dielectric absorption of C1 has a direct effect. Dielectric absorption not only creates residual non-linearity in the dual-slope converter, but causes the converter to output different values for a fixed input as the conversion rate is varied. Dielectric absorption is defined as the capacitor dielectric's unwillingness to accept or give up charge instantaneously. This effect is modeled as a parasitic RC network across the main capacitor. A charged capacitor will require some time to discharge, even through a dead short, due to the parasitic RC network and some amount of charge will be absorbed by the parasitic C after charging of the main capacitor has stopped. Typically, Teflon, polystyrene and polypropylene dielectrics offer better performance than paper, mylar, or glass. Electrolytics have the worst dielectric absorption characteristics and should be avoided for use in slope integrating converters.

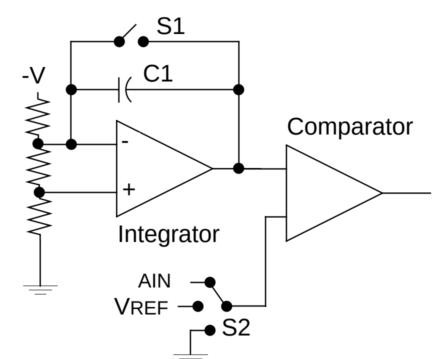
National Semiconductor is a Registered Trademark of National Semiconductor Corporation.

Modified Single-Slope Converter

The modified single-slope converter has been designed to compensate for the effects present in the previous converters. Resolutions of up to 16-bits can be achieved using high precision components and voltage reference source. Figure 15 shows the modified single-slope converter. Some features of this converter are:

- Continuously corrects for zero and full-scale drifts in all components of the circuit.
- The integrating capacitor C1 is charged periodically and always in the same direction. The error induced from dielectric absorption will be small and can be compensated by using an offset term in the calibration procedure.
- The ramp voltage always approaches the comparator trip point from the same direction and slew rate.
- There is no noise rejection capability because the input signal is directly coupled to the comparator input. A filter at the comparator input would cause a delay due to the settling time of the filter.

FIGURE 15: MODIFIED SINGLE-SLOPE INTEGRATING CONVERTER



The microcontroller sends a periodic signal to the switch S1 regardless of the operating mode of the system. The output of the integrator is a fixed frequency, period and height signal which is fed into the input of the comparator. The time between ramps is long enough to allow the integrating capacitor C1 to discharge completely. The other input is multiplexed with ground, reference, and the AIN through switch S2. When the microcontroller starts a conversion, the ground signal is switched into the comparator and the time for the ramp to cross zero is measured and stored. The same measurements are repeated for the reference and AIN signals. Assuming that the integrator ramps are highly linear, Equation 3 is used to determine the value of AIN.

EQUATION 3: OUTPUT EQUATION FOR THE MODIFIED-SLOPE CONVERTER

$$AIN = \frac{\tau_{AIN} - \tau_{GND}}{\tau_{VREF} - \tau_{GND}} \times K \quad \mu V$$

where τ_{AIN} is the measured time for the AIN signal, τ_{VREF} is the measured time for the voltage reference signal, τ_{GND} is the measured time for the ground signal, and K is a constant (typically 10^7).

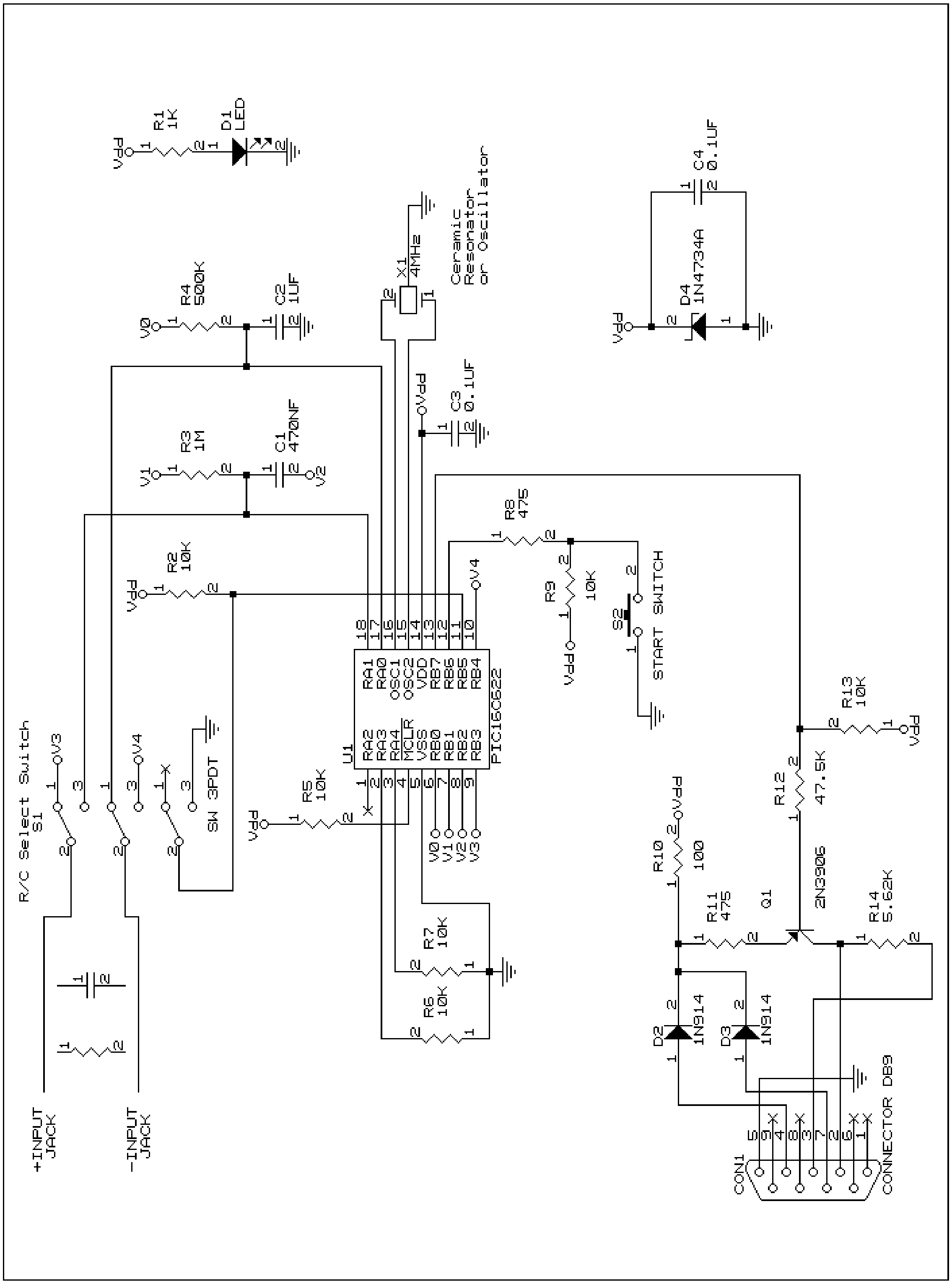
APPLICATION CIRCUIT

The application circuit, called PICMETER, uses a PIC16C622 as a resistance and capacitance meter. The PICMETER uses a variation of the single-slope integrating convertor. The linear slope and integrator of Figure 13 are replaced with the exponential charge waveform of a RC Network. The charge time of a known component is compared against the charge time of an unknown component to determine the value of the unknown component.

A schematic of the PICMETER is shown in Figure 16. All reference designators cited in this section refer to this schematic. Results are transmitted to a PC which displays the value measured. The PICMETER can measure resistance in the range 1 KΩ to 999 KΩ and capacitance from 1 nF to 999 nF.

The following sections describe, in detail, the hardware, firmware, and PC software used in the application circuit. Appendix A shows the PICMETER firmware and Appendix B has the PC software. Appendix C contains the PCB layout.

FIGURE 16: PICMETER SCHEMATIC



Power

The RS-232 serial port provides power to the PICMETER. The RTS and DTR lines from the serial port output 3V to 11V to the PICMETER. The diodes D2 and D3 prevent any damage to the PC's serial port. Resistor R10 is used to current limit the Zener diode, D4. D4 is used to regulate the RTS and DTR voltage to 5.6V. Capacitors C3 and C4 provide power supply filtering to the Zener diode and the PIC16C622. This method of supplying power to devices using a serial port, such as a trackball or mouse, is very simple considering that the PICMETER requires approximately 7 mA to function.

Switches

Switch S1 is used to select either a resistor or capacitor measurement. RB5 of the PIC16C622 is used to detect what type of component is being measured. This switch also swaps the unknown component into the RC network.

If a resistor is the unknown component and a capacitor measurement is requested, the circuit reduces to a resistor divider on the VIN- pin of the comparator. This would result in a measured value of 0 pF if the voltage on the resistor divider network is greater than the voltage reference setting. Otherwise an error is detected. If a capacitor is the unknown component and a resistor measurement is selected, the circuit reduces to a capacitor divider network on the VIN- pin of the comparator. This case will also produce an error message.

Resistor measurements that are started without any component connected to the measuring terminals will cause an error. Capacitor measurements without a component connected to the measuring terminals will give a result of 0 pF.

Switch S2 is used to initiate a measurement. The switch is connected to RB6 of the PIC16C622 and the PORTB wake-up on change interrupt is used to detect a key press. A modified version of the firmware in AN552, *"Implementing Wake-up on Key Stroke"* was used to control the interrupt.

Measuring the Charge Time

The procedures for measuring a resistor or capacitor are the same except for the I/O pins used to control the RC networks. This also applies when measuring a known or unknown component.

Measurement Overview

The charge time of the unknown RC network is measured using Timer0. This value is multiplied by the known value of resistance or capacitance and stored in an accumulator. Then the charge time of the known RC network is measured. The accumulator is divided by the known RC network charge time to give the value of resistance or capacitance of the unknown component. Equation 4 shows the equation used to calculate resistance and Equation 5 shows the capacitance equation.

EQUATION 4: RESISTANCE EQUATION

$$R_{UNK} = \frac{\tau_{UNK} \times R_{KN}}{\tau_{KN}}$$

EQUATION 5: CAPACITANCE EQUATION

$$C_{UNK} = \frac{\tau_{UNK} \times C_{KN}}{\tau_{KN}}$$

R_{UNK} and C_{UNK} are the unknown resistor or capacitor values. R_{KN} and C_{KN} are the known resistor and capacitor values. τ_{UNK} and τ_{KN} are the charge times for the unknown and known components.

Detailed Measurement Description

The first step in measuring the charge time of either the known or the unknown RC networks is to reconfigure the I/O pins. The default state of the PORTA and PORTB pins connected to the RC network are all grounded outputs. This discharges all capacitors in the RC networks. The unknown component is measured first, so the known component, R4 or C1, is removed from the RC network. This is accomplished by making RB0 or RB2 on the PIC16C622 an input. Connections to the other RC network are kept grounded.

The analog modules are now initialized. The mode of the comparators is set to Four Inputs Multiplexed to Two Comparators (Figure 6). The CIS bit, CMCON<3> is cleared to select RA0 as the VIN- input to comparator 1 and RA1 as the VIN- input to comparator 2. The voltage reference is enabled, the output is disabled, and the high range is selected. The tap on the resistor ladder is set to 12. The value of 12 was selected because it is the lowest value of V_{REF} that will trip the comparators, yet gives a time constant long enough to achieve good resolution for the measurement. After a 20 msec delay, which allows the analog modules to stabilize, the comparator flag is cleared. Comparator interrupts are enabled and Timer0 is cleared. Finally, the PEIE bit is set to enable comparator interrupts and the GIE bit is set to enable interrupts.

Now that the analog systems are ready, Timer0 is cleared again and power is applied to the unknown RC network by setting RB1 or RB3 high. Timer0 begins to increment a set of three registers which are cascaded together. These registers contain the charge time of the component. While waiting for the DONE flag, the ERROR flag is checked. See the Error Message section for an explanation of error detection. When the capacitor voltage trips the comparator, Timer0 is prevented from further incrementing the time registers and the DONE flag is set. The value in the time registers is τ_{UNK}.

The analog modules are now disabled. The comparator interrupts are disabled and the comparators are turned off (CM<2:0>=111). RA0 through RA3 and RB0 through RB4 are set up as grounded outputs to discharge the capacitors in the RC networks. This prevents a false reading during the next measurement. The voltage reference is disabled to conserve power and all interrupt flags are cleared. Extra delay loops are added at this time to ensure that the capacitors are discharged.

The charge time, τ_{UNK}, is then multiplied by the value of known resistance or capacitance. These values, in pF or Ω, were obtained by measuring the known RC networks with a Fluke meter. Each of these values is a 24-bit number. The result of multiplication is a 56-bit number which is stored in accumulators ACCb (most significant 24-bits) and ACCc (least significant 24-bits).

The process now repeats itself, except this time the charge time of the known RC network is measured. Now the unknown component is removed from the RC network by making the connections from the PIC16C622 inputs. The analog modules are initialized and the same procedure explained above is followed to measure the charge time of the known RC network. The 56-bit result previously stored in accumulators ACCb and ACCc is now divided by the charge time of the known component, τ_{KN}. This result is a 24-bit number which has the units of pF or Ω. This value is then transmitted to the PC.

RS-232 Transmission

PICMETER uses a transmit only, software implemented serial port adapted from AN593, *"Serial Port Routines Without Using the RTCC"*. Hardware hand-shaking is not used. Since the serial port is realized in software, all interrupts must be disabled during transmission otherwise the baud rate can get corrupted.

On power-up, PICMETER sends a boot message to the PC which is "PICMETER Booted!". Otherwise, a four byte packet structure with a command byte and 3 data bytes is used. The command byte contains one of four possible commands:

- ASCII 'S' signifies that a measurement has been initiated
- ASCII 'E' tells the PC that an error has been detected
- ASCII 'R' tells the PC that resistance data is contained in the three data bytes
- ASCII 'C' tells the PC that capacitance data is contained in the three data bytes

The first data byte for the 'R' and 'C' commands contain the MSB of the measured value. The last data byte contains the LSB of the measured value. The three data bytes for the commands 'S' and 'E' do not contain any useful information at this time.

An 'S' command is issued every time the start switch, S2, is pressed. PICMETER then sends an 'R' or 'C' command for a valid measurement or an 'E' command when an error is detected.

Since the PICMETER operates from a single supply voltage, a discrete transistor is used as a level shifter. This insures that a low output on the RS-232 TXD line is between -3V and -11V. When the TXD line, RB7, from the PIC16C622 is at a logic high level, the transistor Q1 is off. The RXD line of the computer will then be at approximately the same voltage as the TXD line, -11V to -3V. A logic low level from RB7 of the PIC16C622 will turn on transistor Q1. This will bring the RXD line of the computer to about the same voltage of the DTR or RTS line, +3V to +11V.

The pins of interest on the DB9 connector CON1 are:

- pin 2 - RXD
- pin 3 - TXD
- pin 4 - DTR
- pin 5 - GND
- pin 7 - RTS

RTS, DTR, and GND provide power and ground to the PICMETER. RXD is connected to the collector of transistor Q1. TXD is connected to RXD through resistor R14. Since hardware hand-shaking is not implemented on the PICMETER, DSR (pin 6) and CTS (pin 8) are left disconnected.

The demo board developed by Microchip was intended to connect directly to a 9-pin serial port. A 9-pin male-to-female cable may also be used. These boards were manufactured by Southwest Circuits located in Tucson, Arizona (Appendix C). The PCB layout for this demo board is shown in Appendix C.

Error Message

The error message is sent only when the PICMETER is making a measurement and detects an error. The range of resistance that the PICMETER measures is 1 kΩ to 999 kΩ. Using the value of C2, 1 μF, the range of charging times for resistance measurements is 1msec to 999 ms. The range of capacitor charging times is also 1 ms to 999 ms using the resistance value of R3, 1MΩ, and a capacitor measuring range of 1 nF to 999 nF. A ceramic resonator of 4 MHz gives Timer0 a resolution of 1 μsec. Therefore, the highest count that the time registers should reach is 999,000. This is a 20-bit number. If the 21^{st} bit should ever be set, it is assumed that the PICMETER is trying to measure the air gap between the measuring terminals, a component that is out of range, or switch S1 is not set correctly for the component in the measuring terminals.

24-Bit Math Routines

The 24-bit math routines were developed using simple algorithms found in any computer math book. These math routines include addition, subtraction, multiplication, division, and 2's complement. Four 24-bit accumulators located in the general purpose RAM area of the PIC16C622 are used by the math routines: ACCa, ACCb, ACCc, and ACCd. Table 2 shows the relationship between the math routines and the accumulators.

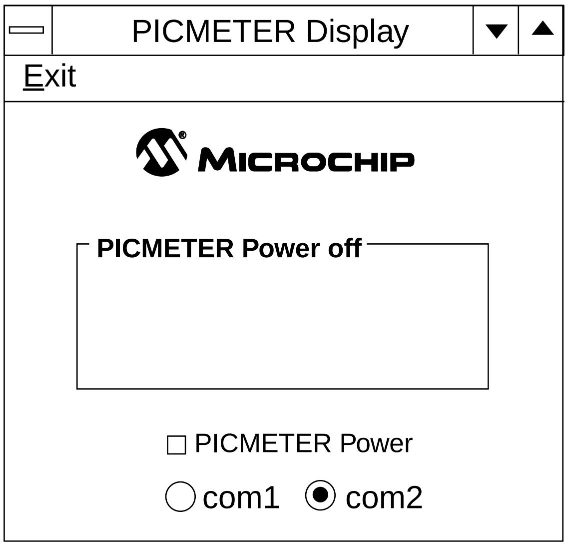
TABLE 2: MATH ROUTINE ACCUMULATORS

Name	Operation	Result	Temp. Storage
Add	ACCa + ACCb	ACCb	N/A
Subtract	2's Comp ACCa then	ACCa	N/A
	ACCa + ACCb	ACCa	
Multiply	ACCa x ACCb	ACCb (MSB's) ACCc (LSB's)	ACCd
Divide	ACCb:ACCc ACCa	quotient in ACCc remainder in ACCb	ACCd
2's Comp	NOT(ACCa) + 1	ACCa	N/A

Computer Program

The program that receives data from the PICMETER was written in Visual Basic® from Microsoft® for the Windows® environment. Figure 17 show the display of the Windows based PICMETER program.

FIGURE 17: PICMETER PC PROGRAM



The operation of this program is simple. A functional description is given below:

a) Select the appropriate COM port by clicking on the COM1 or COM2 buttons.

b) Turn power on to the PICMETER by clicking on the PICMETER Power button.

c) The frame message should read “PICMETER Booted!”, the frame contents will be cleared, and the LED on the PICMETER should be on.

d) The switch S1 selects the type of component that is in the measuring terminals.

e) Pressing the START button, S2, on the PICMETER will initiate a measurement. The frame message should read “Measuring Component” and the contents of the frame will be cleared.

f) When the measurement is complete, the frame message will read “Resistance” or “Capacitance” depending on the position of switch S1. The value of the component will be displayed in the frame as well as the units.

g) If an error is detected, the frame message will read “Error Detected”. This is only a measurement error. Check the component on the measuring terminals and the position of switch S1.

h) Turn off the PICMETER by clicking on the PICMETER Power button. The frame message will change to “PICMETER Power OFF”, the frame contents will be cleared, and the LED on the PICMETER will turn off.

Appendix B contains a complete listing of the Visual Basic program.

PICMETER ACCURACY

The PICMETER measures capacitance in the range of 1 nF to 999 nF. Table 3 shows a comparison of various capacitors. All capacitors have a tolerance of 10% and have various dielectrics. The average error percentage is 3%.

TABLE 3: CAPACITANCE MEASUREMENTS

Capacitance Accuracy			
Marked Value	**Fluke Value**	**PICMETER Value**	**Error %**
2.2 nF	2.3 nF	2.2 nF	4.3
2.5 nF	2.63 nF	2.5 nF	4.9
20 nF	16.5 nF	16.3 nF	1.2
33 nF	35.2 nF	35.8 nF	1.7
47 nF	45 nF	44.5 nF	1.1
50 nF	52 nF	52.9 nF	1.7
100 nF	99.7 nF	93 nF	6.7
0.1 µF	95 nF	96.1 nF	1.2
0.1 µF	99.4 nF	102.8 nF	3.4
0.22 µF	215 nF	215.2 nF	0.1
470 nF	508 nF	518.9 nF	2.1
940 nF	922 nF	983.1 nF	6.6

The 2.5 nF, 100 nF and 940 nF capacitors all have polyester dielectric material. The Equivalent Series Resistance (ESR) of polyester capacitors is typically high which would cause the PICMETER to have a larger error than other dielectrics. If the error percentages for these capacitors is ignored, the average error decreases to 1.9%.

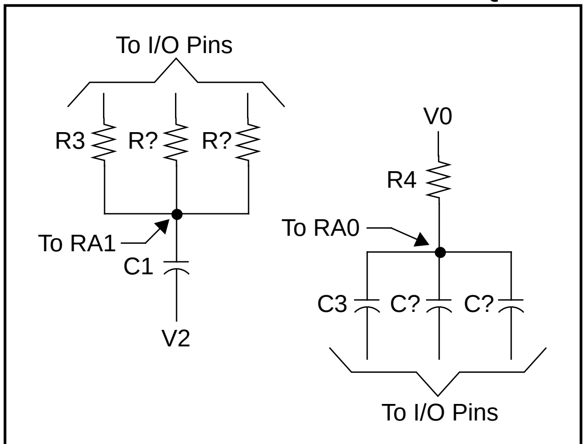
The resistance range of the PICMETER is 1 kΩ to 999 kΩ. Table 4, Resistance Measurements, shows a comparison of various resistors in this range. All resistors have a tolerance of 5%. The average error percentage is 1%.

TABLE 4: RESISTANCE MEASUREMENTS

Resistance Accuracy			
Marked Value	**Fluke Value**	**PICMETER Value**	**Error %**
1.2K	1.215K	1.2K	1.3
5.1K	5.05K	5.0K	1.0
8.2K	8.47K	8.3K	2.0
10K	10.2K	10K	2.0
15K	15.36K	15.1K	1.7
20K	20.8K	20.5K	1.5
30K	30.4K	30K	1.4
51K	50.3K	49.8K	1.0
75K	75.5K	74.8K	1.0
91K	96.4K	95.9K	0.6
150K	146.3K	145.6K	0.5
200K	195.5K	195K	0.3
300K	309K	309.5K	0.2
430K	433K	434.5K	0.4
560K	596K	599.6K	0.6
680K	705K	709.8K	0.7
820K	901K	907.3K	0.7
910K	970K	977.8K	0.8

The accuracy of the PICMETER is dependent on the range of components being measured. If auto-ranging could be implemented, the accuracy of the PICMETER could be improved. The addition of capacitors in parallel with C2 of Figure 16 would allow auto-ranging for resistor measurements. Additional resistors in parallel with R3 would give auto-ranging capability to capacitor measurements. Figure 18 shows a simple implementation of auto-ranging given that the I/O pins are available. The R? and C? are the extra components that are added to the PICMETER circuit. These components should be optimized for a particular range of devices.

FIGURE 18: AUTO-RANGING TECHNIQUE



Another addition to the PICMETER that would increase the accuracy of components being measured is a constant current source. The source would feed into the resistor of the RC networks. This provides the same charging current to all RC networks being measured. Figure 19 shows a bilateral current source and Figure 20 shows a precision current source.

FIGURE 19: BILATERAL CURRENT SOURCE

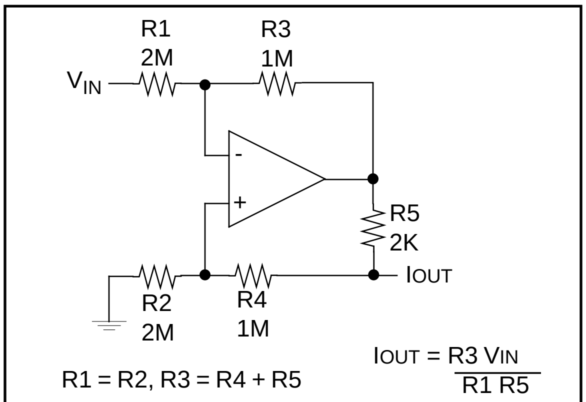
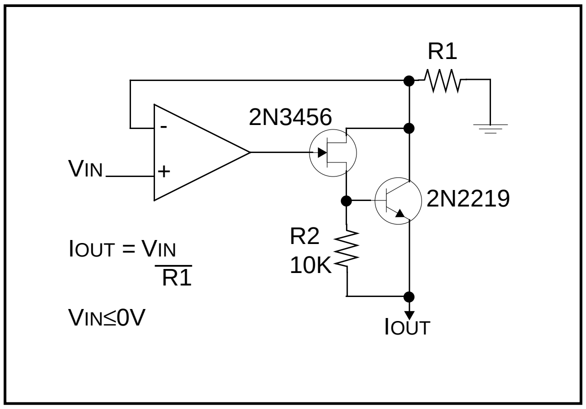
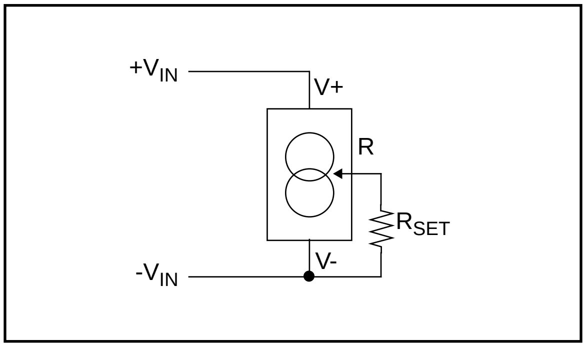


FIGURE 20: PRECISION CURRENT SOURCE



The alternative to the previous current sources is a single chip solution. A 3-terminal adjustable current source, such as a LM134/LM234/LM334 from National Semiconductor, is an ideal choice. This output current is programmable from 1 µA to 10 mA and requires a single external resistor to set the value of current. Figure 21 shows a block diagram of the LM334Z.

FIGURE 21: LM334Z BLOCK DIAGRAM



CONCLUSION

The PIC16C62X devices add two significant analog features to the PIC16CXX mid-range family: comparators and a voltage reference. The flexibility of eight operating modes for the comparator module allows the designer to tailor the PIC16C62X device to the application. The addition of an on-chip voltage reference simplifies the design by removing at least one external component and power consumption. These analog modules coupled with the PIC16CXX mid-range family core create a new path to achieve high resolution results.

Please check the Microchip BBS for the latest version of the source code. For BBS access information, see Section 6, Microchip Bulletin Board Service, page 6-3.

APPENDIX A: PICMETER FIRMWARE

```
MPASM 01.02.05 Intermediate PICMETER.ASM   5-1-1995 11:29:17             PAGE 1
PICMETER Firmware for PIC16C622

LOC OBJECT CODE    LINE SOURCE TEXT
VALUE

                          TITLE "PICMETER Firmware for PIC16C622"
                          LIST P = 16C622, F = INHX8M

                          INCLUDE "C:\PICMASTR\P16CXX.INC"
                     ; P16CXX.INC  Standard Header File, Version 0.2 Microchip Technology, Inc.


3FB9                      FUSES _BODEN_OFF&_CP_OFF&_PWDT_ON&_WDT_OFF&_XT_OSC

                     ;************************************************************************
                     ;*----------------------------------------------------------------------*
                     ;*-                                                                    -*
                     ;*-   PICMETER - Resistance and Capacitance Meter                      -*
                     ;*-                                                                    -*
                     ;*----------------------------------------------------------------------*
                     ;*-                                                                    -*
                     ;*-   Author:        Rodger Richey                                     -*
                     ;*-                  Applications Engineer                             -*
                     ;*-   Filename:      picmtr.asm                                        -*
                     ;*-   Revision:      1 May 1995                                        -*
                     ;*-                                                                    -*
                     ;*----------------------------------------------------------------------*
                     ;*-                                                                    -*
                     ;*-   PICMETER is based on a PIC16C622 which has two comparators and   -*
                     ;*-   a variable voltage reference. Resistance and capacitance is      -*
                     ;*-   calculated by measuring the time constant of a RC network. The   -*
                     ;*-   toggle switch selects either resistor or capacitor input. The    -*
                     ;*-   pushbutton switch starts a measurement. The time constant of the -*
                     ;*-   unknown component is compared to that of known component to      -*
                     ;*-   calculate the value of the unknown component.  The following     -*
                     ;*-   formulas are used:                                               -*
                     ;*-                                                                    -*
                     ;*-   Resistance:     Ru = ( Rk * Tu ) / Tk                            -*
                     ;*-   Capacitance:    Cu = ( Ck * Tu ) / Tk                            -*
                     ;*-                                                                    -*
                     ;*----------------------------------------------------------------------*
                     ;************************************************************************


                     ;************************************************************************
                     ;*----------------------------------------------------------------------*
                     ;*-   RS232 code borrowed from Application Note AN593                  -*
                     ;*-   "Serial Port Routines Without Using the RTCC"                    -*
                     ;*-   Author: Stan D'Souza                                             -*
                     ;*----------------------------------------------------------------------*
                     ;************************************************************************
003D 0900            xtal    equ     .4000000
2580                 baud    equ     .9600
000F 4240            fclk    equ     xtal/4
                     ;************************************************************************
                     ;The value baudconst must be a 8-bit value only
0020                 baudconst       equ     ((fclk/baud)/3-2)
                     ;************************************************************************
```

```
                ;***********************************************************************
                ;       Bit Equates
                ;***********************************************************************
0000            BEGIN   equ     0               ;begin a measurement flag
0007            DONE    equ     7               ;done measuring flag
0005            WHICH   equ     5               ;R or C measurement flag
0003            F_ERROR equ     3               ;error detection flag
0005            EMPTY   equ     5               ;flag if component is connected
0000            V0      equ     0               ;power for R reference ckt
0001            V1      equ     1               ;power for C reference ckt
0002            V2      equ     2               ;ground for C reference ckt
0003            V3      equ     3               ;power for unknown R ckt
0004            V4      equ     4               ;ground for unknown C ckt
0007            msb_bit equ     7               ;define for bit 7
0000            lsb_bit equ     0               ;define for bit 0
0007            RkHI    equ     0x07            ;value of the known resistance, R4, in ohms
009D            RkMID   equ     0x9D            ;measured by a Fluke meter
0038            RkLO    equ     0x38
0007            CkHI    equ     0x07            ;value of the known capacitance, C1, in pF
00C8            CkMID   equ     0xC8            ;measured by a Fluke meter
0030            CkLO    equ     0x30

                ;***********************************************************************
                ;       User Registers
                ;***********************************************************************
                ;       Bank 0
0020            W_TEMP  equ     0x20            ;Bank 0 temporary storage for W reg
0021            STATUS_TEMP equ 0x21            ;temporary storage for STATUS reg
0023            Ttemp   equ     0x23            ;temporary Time register
0024            flags   equ     0x24            ;flags register
0025            count   equ     0x25            ;RS232 register
0026            txreg   equ     0x26            ;RS232 data register
0027            delay   equ     0x27            ;RS232 delay register
0028            offset  equ     0x28            ;table position register
0029            msb     equ     0x29            ;general delay register
002A            lsb     equ     0x2A            ;general delay register
0040            TimeLO  equ     0x40            ;Time registers
0041            TimeMID equ     0x41
0042            TimeHI  equ     0x42

                ;       Math related registers
0050            ACCaHI  equ     0x50            ;24-Bit accumulator a
0051            ACCaMID equ     0x51
0052            ACCaLO  equ     0x52
0053            ACCbHI  equ     0x53            ;24-Bit accumulator b
0054            ACCbMID equ     0x54
0055            ACCbLO  equ     0x55
0056            ACCcHI  equ     0x56            ;24-Bit accumulator c
0057            ACCcMID equ     0x57
0058            ACCcLO  equ     0x58
0059            ACCdHI  equ     0x59            ;24-Bit accumulator d
005A            ACCdMID equ     0x5A
005B            ACCdLO  equ     0x5B
005C            temp    equ     0x5C            ;temporary storage

                ;       User Registers Bank 1
                ;W_TEMP  equ     0xA0           ;Bank 1 temporary storage for W reg

                ;       User defines
                 #define tx      PORTB,7        ;define for RS232 TXD output pin

                ;***********************************************************************

                        org     0x0
0000 2810               goto    init

```



```
                          org     0x4
0004 28B9                 goto    ServiceInterrupts

                          org     0x10
0010           init
0010 1283                 bcf     STATUS,RP0     ;select bank 0
0011 0185                 clrf    PORTA          ;clear PORTA and PORTB
0012 0186                 clrf    PORTB
0013 1786                 bsf     tx             ;set TXD output pin
0014 01A4                 clrf    flags          ;clear flags register
0015 3010                 movlw   0x10           ;load table offset register
0016 00A8                 movwf   offset
0017 018B                 clrf    INTCON         ;clear interrupt flags and disable interrupts
0018 3007                 movlw   0x07           ;turn off comparators, mode 111
0019 009F                 movwf   CMCON
001A 2140                 call    delay20        ;wait for comarators to settle
001B 089F                 movf    CMCON,F
001C 130C                 bcf     PIR1,CMIF
001D 1683                 bsf     STATUS,RP0     ;select bank 1
001E 3088                 movlw   0x88           ;WDT prescalar,internal TMR0 increment
001F 0081                 movwf   OPTION_REG
0020 0185                 clrf    TRISA          ;PORTA all outputs, discharges RC ckts
0021 3060                 movlw   0x60           ;PORTA<7,4:0> outputs, PORTA<6:5> inputs
0022 0086                 movwf   TRISB
0023 300C                 movlw   0x0C           ;setup Voltage Reference
0024 009F                 movwf   VRCON
0025 1283                 bcf     STATUS,RP0     ;select bank 0
0026 3008                 movlw   0x08           ;enable RBIE interrupt
0027 008B                 movwf   INTCON
0028 213D                 call    vlong          ;delay before transmitting boot message
0029 213D                 call    vlong          ;to allow computer program to setup
002A 213D                 call    vlong
002B 2131                 call    BootMSG        ;transmit boot message
002C 178B                 bsf     INTCON,GIE     ;enable global interrupt bit

002D           start
002D 1C24                 btfss   flags,BEGIN    ;wait for a start measurement key press
002E 282D                 goto    start
002F 1024                 bcf     flags,BEGIN    ;clear start measurement flag

0030 138B                 bcf     INTCON,GIE     ;transmit a start measurement message
0031 3053                 movlw   'S'            ;to the PC
0032 20AD                 call    Send
0033 178B                 bsf     INTCON,GIE

0034 01C2                 clrf    TimeHI         ;reset Time registers
0035 01C1                 clrf    TimeMID
0036 01C0                 clrf    TimeLO
0037 1E86                 btfss   PORTB,WHICH    ;detect if resistor or capacitor measure
0038 2862                 goto    Capacitor

0039           Resistor
0039 1683                 bsf     STATUS,RP0     ;set V0 to input
003A 1406                 bsf     TRISB,V0
003B 1283                 bcf     STATUS,RP0
003C 20FB                 call    AnalogOn       ;turn analog on
003D 0181                 clrf    TMR0
003E 0000                 nop
003F 1586                 bsf     PORTB,V3       ;turn power on to unknown RC ckt
0040 19A4      RwaitU     btfsc   flags,F_ERROR  ;detect if an error occurs
0041 288B                 goto    ErrorDetect
0042 1FA4                 btfss   flags,DONE     ;measurement completed flag
0043 2840                 goto    RwaitU
0044 13A4                 bcf     flags,DONE     ;clear measurement completed flag
0045 2111                 call    AnalogOff      ;turn analog off

```

```
0046 2126               call    SwapTtoA        ;swap Time to accumulator a
0047 3007               movlw   RkHI            ;swap known resistance value
0048 00D3               movwf   ACCbHI          ;to accumulator b
0049 309D               movlw   RkMID
004A 00D4               movwf   ACCbMID
004B 3038               movlw   RkLO
004C 00D5               movwf   ACCbLO
004D 2230               call    Mpy24           ;multiply accumulator a and b
004E 1683               bsf     STATUS,RP0      ;set V3 to input
004F 1586               bsf     TRISB,V3
0050 1283               bcf     STATUS,RP0
0051 20FB               call    AnalogOn        ;turn analog on
0052 0181               clrf    TMR0
0053 0000               nop
0054 1406               bsf     PORTB,V0        ;turn power on to known RC ckt
0055 19A4      RwaitK   btfsc   flags,F_ERROR   ;detect if an error occurs
0056 288B               goto    ErrorDetect
0057 1FA4               btfss   flags,DONE      ;measurement completed flag

0058 2855               goto    RwaitK
0059 13A4               bcf     flags,DONE      ;clear measurement completed flag
005A 2111               call    AnalogOff       ;turn analog off
005B 2126               call    SwapTtoA        ;swap Time to accumulator a
005C 224B               call    Div24           ;divide multiply by known time
005D 138B               bcf     INTCON,GIE      ;disable all interrupts
005E 3052               movlw   'R'             ;transmit, for R measurement
005F 20AD               call    Send
0060 178B               bsf     INTCON,GIE      ;enable global interrupt bit
0061 282D               goto    start           ;restart
0062           Capacitor
0062 1683               bsf     STATUS,RP0      ;set V2 to input
0063 1506               bsf     TRISB,V2
0064 1283               bcf     STATUS,RP0
0065 20FB               call    AnalogOn        ;turn analog on
0066 0181               clrf    TMR0
0067 0000               nop
0068 1486               bsf     PORTB,V1        ;turn power on to unknown RC ckt
0069 19A4      CwaitU   btfsc   flags,F_ERROR   ;detect if an error occurs
006A 288B               goto    ErrorDetect
006B 1FA4               btfss   flags,DONE      ;measurement completed flag
006C 2869               goto    CwaitU
006D 13A4               bcf     flags,DONE      ;clear measurement completed flag
006E 2111               call    AnalogOff       ;turn analog off
006F 2126               call    SwapTtoA        ;swap Time to accumulator a
0070 3007               movlw   CkHI            ;swap known resistance value
0071 00D3               movwf   ACCbHI          ;to accumulator b
0072 30C8               movlw   CkMID
0073 00D4               movwf   ACCbMID
0074 3030               movlw   CkLO
0075 00D5               movwf   ACCbLO
0076 2230               call    Mpy24           ;multiply accumulator a and b
0077 1683               bsf     STATUS,RP0      ;set V3 to input
0078 1606               bsf     TRISB,V4
0079 1283               bcf     STATUS,RP0
007A 20FB               call    AnalogOn        ;turn analog on
007B 0181               clrf    TMR0
007C 0000               nop
007D 1486               bsf     PORTB,V1        ;turn power on to known RC ckt
007E 19A4      CwaitK   btfsc   flags,F_ERROR   ;detect if an error occurs
007F 288B               goto    ErrorDetect
```

```
0080 1FA4                    btfss   flags,DONE      ;measurement completed flag
0081 287E                    goto    CwaitK
0082 13A4                    bcf     flags,DONE      ;clear measurement completed flag
0083 2111                    call    AnalogOff       ;turn analog off

0084 2126                    call    SwapTtoA        ;swap Time to accumulator a
0085 224B                    call    Div24           ;divide multiply by known time

0086 138B                    bcf     INTCON,GIE      ;disable all interrupts
0087 3043                    movlw   'C'             ;transmit, for C measurement
0088 20AD                    call    Send
0089 178B                    bsf     INTCON,GIE      ;enable global interrupt bit
008A 282D                    goto    start           ;restart

008B                 ErrorDetect
008B 1283                    bcf     STATUS,RP0      ;disable TMR0
008C 128B                    bcf     INTCON,T0IE
008D 110B                    bcf     INTCON,T0IF
008E 2111                    call    AnalogOff       ;turn analog off
008F 11A4                    bcf     flags,F_ERROR   ;clear error flag

0090 138B                    bcf     INTCON,GIE      ;disable all interrupts
0091 3045                    movlw   'E'             ;transmit, for C measurement
0092 20AD                    call    Send
0093 178B                    bsf     INTCON,GIE      ;enable global interrupt bit
0094 282D                    goto    start           ;restart

                     ;*******************************************************************
                     ;*-----------------------------------------------------------------*
                     ;*-      RS232 Transmit Routine
                     ;*-      Borrowed from AN593, "Serial Port Routines Without Using the RTCC"
                     ;*-      Author: Stan D'Souza
                     ;*-      This is the routine that interfaces directly to the hardware
                     ;*-----------------------------------------------------------------*
                     ;*******************************************************************
0095                 Transmit
0095 1283                    bcf     STATUS,RP0
0096 00A6                    movwf   txreg
0097 1386                    bcf     tx              ;send start bit
0098 3020                    movlw   baudconst
0099 00A7                    movwf   delay
009A 3009                    movlw   0x9
009B 00A5                    movwf   count
009C                 txbaudwait
009C 0BA7                    decfsz  delay
009D 289C                    goto    txbaudwait
009E 3020                    movlw   baudconst
009F 00A7                    movwf   delay
00A0 0BA5                    decfsz  count
00A1 28A6                    goto    SendNextBit
00A2 3009                    movlw   0x9
00A3 00A5                    movwf   count
00A4 1786                    bsf     tx              ;send stop bit
00A5 0008                    return
00A6                 SendNextBit
00A6 0CA6                    rrf     txreg
00A7 1C03                    btfss   STATUS,C
00A8 28AB                    goto    Setlo
00A9 1786                    bsf     tx
00AA 289C                    goto    txbaudwait
00AB 1386            Setlo   bcf     tx
00AC 289C                    goto    txbaudwait
                     ;___________________________________________________________________

                     ;*******************************************************************
                     ;*-----------------------------------------------------------------*
```

```
                ;*-     Generic Transmit Routine
                ;*-     Sends what is currently in the W register and accumulator ACCc
                ;*----------------------------------------------------------------*
                ;*****************************************************************
00AD            Send
00AD 2095               call    Transmit
00AE 2146               call    delay1              ;delay between bytes
00AF 0856               movf    ACCcHI,W            ;transmit high resistance byte
00B0 2095               call    Transmit
00B1 2146               call    delay1              ;delay between bytes
00B2 0857               movf    ACCcMID,W           ;transmit mid resistance byte
00B3 2095               call    Transmit
00B4 2146               call    delay1              ;delay between bytes
00B5 0858               movf    ACCcLO,W            ;transmit low resistance byte
00B6 2095               call    Transmit
00B7 2146               call    delay1              ;delay between bytes
00B8 0008               return
                ;_________________________________________________________________

                ;*****************************************************************
                ;*----------------------------------------------------------------*
                ;*-     Interrupt Service Routines
                ;*----------------------------------------------------------------*
                ;*****************************************************************
00B9            ServiceInterrupts
00B9 00A0               movwf   W_TEMP              ;Pseudo push instructions
00BA 0E03               swapf   STATUS,W
00BB 1283               bcf     STATUS,RP0
00BC 00A1               movwf   STATUS_TEMP

00BD 0801               movf    TMR0,W
00BE 00A3               movwf   Ttemp
00BF 190B               btfsc   INTCON,T0IF         ;Service Timer 0 overflow
00C0 20E5               call    ServiceTimer
00C1 1B0C               btfsc   PIR1,CMIF           ;Stops Timer0, Records Value
00C2 20EC               call    ServiceComparator
00C3 180B               btfsc   INTCON,RBIF         ;Service pushbutton switch
00C4 20CB               call    ServiceKeystroke    ;Starts a measurement

00C5 1283               bcf     STATUS,RP0
00C6 0E21               swapf   STATUS_TEMP,W       ;Pseudo pop instructions
00C7 0083               movwf   STATUS
00C8 0EA0               swapf   W_TEMP,F
00C9 0E20               swapf   W_TEMP,W

00CA 0009               retfie
                ;_________________________________________________________________

                ;*****************************************************************
                ;*----------------------------------------------------------------*
                ;*-    Borrowed from AN552, "Implementing Wake-up on Key Stroke"
                ;*-    Author: Stan D'Souza
                ;*----------------------------------------------------------------*
                ;*****************************************************************
00CB            ServiceKeystroke
00CB 118B               bcf     INTCON,RBIE         ;disable interrupt
00CC 0906               comf    PORTB,W             ;read PORTB
00CD 100B               bcf     INTCON,RBIF         ;clear interrupt flag
00CE 3940               andlw   B'01000000'
00CF 1903               btfsc   STATUS,Z
00D0 28D6               goto    NotSwitch
00D1 2143               call    delay16             ;de-bounce switch for 16msec
00D2 0906               comf    PORTB,W             ;read PORTB again
00D3 20D9               call    KeyRelease          ;check for key release
00D4 1424               bsf     flags,BEGIN
00D5 0008               return
```

```

00D6                NotSwitch                               ;detected other PORTB pin change
00D6 100B                   bcf     INTCON,RBIF             ;reset RBI interrupt
00D7 158B                   bsf     INTCON,RBIE
00D8 0008                   return

00D9                KeyRelease
00D9 2143                   call    delay16                 ;debounce switch
00DA 0906                   comf    PORTB,W                 ;read PORTB
00DB 100B                   bcf     INTCON,RBIF             ;clear flag
00DC 158B                   bsf     INTCON,RBIE             ;enable interrupt
00DD 3940                   andlw   B'01000000'
00DE 1903                   btfsc   STATUS,Z                ;key still pressed?
00DF 0008                   return                          ;if no, then return
00E0 0063                   sleep                           ;else, save power
00E1 118B                   bcf     INTCON,RBIE             ;disable interrupts
00E2 0906                   comf    PORTB,W                 ;read PORTB
00E3 100B                   bcf     INTCON,RBIF             ;clear flag
00E4 28D9                   goto    KeyRelease              ;try again
                    ;__________________________________________________________________

                    ;******************************************************************
                    ;*----------------------------------------------------------------*
                    ;*-     ISR to service a Timer0 overflow
                    ;*----------------------------------------------------------------*
                    ;******************************************************************
00E5                ServiceTimer
00E5 0AC1                   incf    TimeMID,F               ;increment middle Time byte
00E6 1903                   btfsc   STATUS,Z                ;if middle overflows,
00E7 0AC2                   incf    TimeHI,F                ;increment high Time byte
00E8 1AC2                   btfsc   TimeHI,EMPTY            ;check if component is connected
00E9 15A4                   bsf     flags,F_ERROR           ;set error flag
00EA 110B                   bcf     INTCON,T0IF             ;clear TMR0 interrupt flag
00EB 0008                   return
                    ;__________________________________________________________________

                    ;******************************************************************
                    ;*----------------------------------------------------------------*
                    ;*-     ISR to service a Comparator interrupt
                    ;*----------------------------------------------------------------*
                    ;******************************************************************
00EC                ServiceComparator
00EC 1283                   bcf     STATUS,RP0              ;select bank 0
00ED 1E86                   btfss   PORTB,WHICH             ;detect which measurement, R or C?
00EE 28F2                   goto    capcomp
00EF 1F1F                   btfss   CMCON,C1OUT             ;detect if R ckt has interrupted
00F0 28F4                   goto    scstop
00F1 28F9                   goto    scend
00F2                capcomp
00F2 1B9F                   btfsc   CMCON,C2OUT             ;detect if C ckt has interrupted
00F3 28F9                   goto    scend
00F4                scstop
00F4 128B                   bcf     INTCON,T0IE             ;disable TMR0 interrupts
00F5 110B                   bcf     INTCON,T0IF
00F6 0823                   movf    Ttemp,W
00F7 00C0                   movwf   TimeLO
00F8 17A4                   bsf     flags,DONE              ;set DONE flag
00F9                scend
00F9 130C                   bcf     PIR1,CMIF               ;clear comparator interrupt flag
00FA 0008                   return
                    ;__________________________________________________________________

                    ;******************************************************************
                    ;*----------------------------------------------------------------*
                    ;*-     Turn Comparators and Vref On
                    ;*----------------------------------------------------------------*
```

```
                    ;***********************************************************************
00FB                AnalogOn
00FB 1283                   bcf     STATUS,RP0          ;select bank 0
00FC 3002                   movlw   0x02                ;turn comparators on, mode 010
00FD 009F                   movwf   CMCON               ;4 inputs multiplexed to 2 comparators
00FE 1683                   bsf     STATUS,RP0          ;select bank 1
00FF 300F                   movlw   0x0F                ;make PORTA<3:0> all inputs
0100 0085                   movwf   TRISA
0101 179F                   bsf     VRCON,VREN
0102 1283                   bcf     STATUS,RP0          ;select bank 0
0103 2140                   call    delay20             ;20msec delay
0104 089F                   movf    CMCON,F             ;clear comparator mismatch condition
0105 130C                   bcf     PIR1,CMIF           ;clear comparator interrupt flag
0106 1683                   bsf     STATUS,RP0
0107 170C                   bsf     PIE1,CMIE           ;enable comparator interrupts
0108 1283                   bcf     STATUS,RP0
0109 170B                   bsf     INTCON,PEIE         ;enable peripheral interrupts
010A 11A4                   bcf     flags,F_ERROR
010B 0181                   clrf    TMR0                ;clear TMR0 counter
010C 0000                   nop
010D 0000                   nop
010E 110B                   bcf     INTCON,T0IF         ;clear TMR0 interrupt flag
010F 168B                   bsf     INTCON,T0IE         ;enable TMR0 interrupts
0110 0008                   return
                    ;_______________________________________________________________________

                    ;***********************************************************************
                    ;*---------------------------------------------------------------------*
                    ;*-     Turn Comparators and Vref Off
                    ;*---------------------------------------------------------------------*
                    ;***********************************************************************
0111                AnalogOff
0111 1283                   bcf     STATUS,RP0
0112 130B                   bcf     INTCON,PEIE
0113 3080                   movlw   0x80                ;reset PORTB value
0114 0086                   movwf   PORTB
0115 1683                   bsf     STATUS,RP0          ;select bank 1
0116 130C                   bcf     PIE1,CMIE           ;disable comparator interrupts
0117 0185                   clrf    TRISA               ;set PORTA pins to outputs, discharge RC ckt
0118 3060                   movlw   0x60                ;set PORTB 7,4-0 as outputs, 6,5 as inputs
0119 0086                   movwf   TRISB
011A 139F                   bcf     VRCON,VREN          ;disable Vref
011B 1283                   bcf     STATUS,RP0          ;select bank 0
011C 3007                   movlw   0x07
011D 009F                   movwf   CMCON               ;disable comparators
011E 2140                   call    delay20             ;20msec delay
011F 089F                   movf    CMCON,F             ;clear comparator mismatch condition
0120 130C                   bcf     PIR1,CMIF           ;clear comparator interrupt flag
0121 110B                   bcf     INTCON,T0IF         ;clear Timer0 interrupt flag
0122 213D                   call    vlong               ;long delay to allow capacitors to discharge
0123 213D                   call    vlong
0124 213D                   call    vlong
0125 0008                   return
                    ;_______________________________________________________________________

                    ;***********************************************************************
                    ;*---------------------------------------------------------------------*
                    ;*-     Swap Time to Accumulator a
                    ;*---------------------------------------------------------------------*
                    ;***********************************************************************
0126                SwapTtoA
0126 1283                   bcf     STATUS,RP0
0127 0842                   movf    TimeHI,W
0128 00D0                   movwf   ACCaHI
0129 0841                   movf    TimeMID,W
012A 00D1                   movwf   ACCaMID
```

```
012B 0840                  movf    TimeLO,W
012C 00D2                  movwf   ACCaLO
012D 01C2                  clrf    TimeHI
012E 01C1                  clrf    TimeMID
012F 01C0                  clrf    TimeLO
0130 0008                  return
                 ;_________________________________________________________________
                 ;*****************************************************************
                 ;*---------------------------------------------------------------*
                 ;*-       Transmit the Boot Message
                 ;*---------------------------------------------------------------*
                 ;*****************************************************************
0131             BootMSG
0131 1283                  bcf     STATUS,RP0       ;select bank 0
0132 3002        msg       movlw   HIGH Table       ;init the PCH for a table call
0133 008A                  movwf   PCLATH
0134 0828                  movf    offset,W         ;move table offset into W
0135 2200                  call    Table            ;get table value
0136 2095                  call    Transmit         ;transmit table value
0137 2146                  call    delay1           ;delay between bytes
0138 0BA8                  decfsz  offset,F         ;check for end of table
0139 2932                  goto    msg
013A 3010                  movlw   0x10             ;reset table offset
013B 00A8                  movwf   offset
013C 0008                  return
                 ;_________________________________________________________________
                 ;*****************************************************************
                 ;*---------------------------------------------------------------*
                 ;*-       Delay Routines
                 ;*---------------------------------------------------------------*
                 ;*****************************************************************
013D 30FF        vlong     movlw   0xff             ;very long delay, approx 200msec
013E 00A9                  movwf   msb
013F 2948                  goto    d1
0140             delay20                            ;20 msec delay
0140 301A                  movlw   .26
0141 00A9                  movwf   msb
0142 2948                  goto    d1
0143             delay16                            ;16 msec delay
0143 3015                  movlw   .21
0144 00A9                  movwf   msb
0145 2948                  goto    d1
0146             delay1                             ;approx 750nsec delay
0146 3001                  movlw   .1
0147 00A9                  movwf   msb
0148 30FF        d1        movlw   0xff
0149 00AA                  movwf   lsb
014A 0BAA        d2        decfsz  lsb,F
014B 294A                  goto    d2
014C 0BA9                  decfsz  msb,F
014D 2948                  goto    d1
014E 0008                  return
                 ;_________________________________________________________________
                           org     0x200
                 ;*****************************************************************
                 ;*---------------------------------------------------------------*
                 ;*-       Table for Boot Message
                 ;*---------------------------------------------------------------*
                 ;*****************************************************************
0200             Table                              ;boot message "PICMETER Booted!"
```

```
0200 0782                 addwf   PCL             ;add W to PCL
0201 3400                 retlw   0
0202 3421                 retlw   '!'
0203 3464                 retlw   'd'
0204 3465                 retlw   'e'
0205 3474                 retlw   't'
0206 346F                 retlw   'o'
0207 346F                 retlw   'o'
0208 3442                 retlw   'B'
0209 3420                 retlw   ' '
020A 3472                 retlw   'r'
020B 3465                 retlw   'e'
020C 3474                 retlw   't'
020D 3465                 retlw   'e'
020E 346D                 retlw   'm'
020F 3443                 retlw   'C'
0210 3449                 retlw   'I'
0211 3450                 retlw   'P'
                     ;_________________________________________________________________

                     ;*****************************************************************
                     ;*---------------------------------------------------------------*
                     ;*-      24-bit Addition
                     ;*-
                     ;*-      Uses ACCa and ACCb
                     ;*-
                     ;*-      ACCa + ACCb -> ACCb
                     ;*---------------------------------------------------------------*
                     ;*****************************************************************
0212                 Add24
0212 0852                 movf    ACCaLO,W
0213 07D5                 addwf   ACCbLO          ;add low bytes
0214 1803                 btfsc   STATUS,C        ;add in carry if necessary
0215 2A1D                 goto    A2
0216 0851            A1   movf    ACCaMID,W
0217 07D4                 addwf   ACCbMID         ;add mid bytes
0218 1803                 btfsc   STATUS,C        ;add in carry if necessary
0219 0AD3                 incf    ACCbHI
021A 0850                 movf    ACCaHI,W
021B 07D3                 addwf   ACCbHI          ;add high bytes
021C 3400                 retlw   0
021D 0AD4            A2   incf    ACCbMID
021E 1903                 btfsc   STATUS,Z
021F 0AD3                 incf    ACCbHI
0220 2A16                 goto    A1
                     ;_________________________________________________________________

                     ;*****************************************************************
                     ;*---------------------------------------------------------------*
                     ;*-      Subtraction ( 24 - 24 -> 24 )
                     ;*-
                     ;*-      Uses ACCa, ACCb, ACCd
                     ;*-
                     ;*-      ACCa -> ACCd,
                     ;*-      2's complement ACCa,
                     ;*-      call Add24 ( ACCa + ACCb -> ACCb ),
                     ;*-      ACCd -> ACCa
                     ;*---------------------------------------------------------------*
                     ;*****************************************************************
0221                 Sub24
0221 0850                 movf    ACCaHI,W        ;Transfer ACCa to ACCd
0222 00D9                 movwf   ACCdHI
0223 0851                 movf    ACCaMID,W
0224 00DA                 movwf   ACCdMID
0225 0852                 movf    ACCaLO,W
0226 00DB                 movwf   ACCdLO
```



```
0227 2275          call    compA          ;2's complement ACCa
0228 2212          call    Add24          ;Add ACCa to ACCb
0229 0859          movf    ACCdHI,W       ;Transfer ACCd to ACCa
022A 00D0          movwf   ACCaHI
022B 085A          movf    ACCdMID,W
022C 00D1          movwf   ACCaMID
022D 085B          movf    ACCdLO,W
022E 00D2          movwf   ACCaLO
022F 3400          retlw   0
              ;____________________________________________________________________
              ;*******************************************************************
              ;*-----------------------------------------------------------------*
              ;*-      Multiply ( 24 X 24 -> 56 )
              ;*-
              ;*-      Uses ACCa, ACCb, ACCc, ACCd
              ;*-
              ;*-      ACCa * ACCb -> ACCb,ACCc  56-bit output
              ;*-      with ACCb (ACCbHI,ACCbMID,ACCbLO) with 24 msb's and
              ;*-      ACCc (ACCcHI,ACCcMID,ACCcLO) with 24 lsb's
              ;*-----------------------------------------------------------------*
              ;*******************************************************************
0230          Mpy24
0230 223F          call    Msetup
0231 0CD9     mloop rrf    ACCdHI         ;rotate d right
0232 0CDA          rrf     ACCdMID
0233 0CDB          rrf     ACCdLO
0234 1803          btfsc   STATUS,C       ;need to add?
0235 2212          call    Add24
0236 0CD3          rrf     ACCbHI
0237 0CD4          rrf     ACCbMID
0238 0CD5          rrf     ACCbLO
0239 0CD6          rrf     ACCcHI
023A 0CD7          rrf     ACCcMID
023B 0CD8          rrf     ACCcLO
023C 0BDC          decfsz  temp           ;loop until all bits checked
023D 2A31          goto    mloop
023E 3400          retlw   0

023F          Msetup
023F 3018          movlw   0x18           ;for 24 bit shifts
0240 00DC          movwf   temp
0241 0853          movf    ACCbHI,W       ;move ACCb to ACCd
0242 00D9          movwf   ACCdHI
0243 0854          movf    ACCbMID,W
0244 00DA          movwf   ACCdMID
0245 0855          movf    ACCbLO,W
0246 00DB          movwf   ACCdLO
0247 01D3          clrf    ACCbHI
0248 01D4          clrf    ACCbMID
0249 01D5          clrf    ACCbLO
024A 3400          retlw   0
              ;____________________________________________________________________
              ;*******************************************************************
              ;*-----------------------------------------------------------------*
              ;*-      Division ( 56 / 24 -> 24 )
              ;*-
              ;*-      Uses ACCa, ACCb, ACCc, ACCd
              ;*-
              ;*-      56-bit dividend in ACCb,ACCc ( ACCb has msb's and ACCc has lsb's)
              ;*-      24-bit divisor in ACCa
              ;*-      quotient is stored in ACCc
              ;*-      remainder is stored in ACCb
              ;*-----------------------------------------------------------------*
              ;*******************************************************************
```

```
024B                Div24
024B 2272                   call    Dsetup
024C 1003           dloop   bcf     STATUS,C
024D 0DD8                   rlf     ACCcLO          ;Rotate dividend left 1 bit position
024E 0DD7                   rlf     ACCcMID
024F 0DD6                   rlf     ACCcHI
0250 0DD5                   rlf     ACCbLO
0251 0DD4                   rlf     ACCbMID
0252 0DD3                   rlf     ACCbHI
0253 1803                   btfsc   STATUS,C        ;invert carry and exclusive or with the
0254 2A58                   goto    clear           ;msb of the divisor then move this bit
0255 1FD0                   btfss   ACCaHI,msb_bit  ;into the lsb of the dividend
0256 0AD8                   incf    ACCcLO
0257 2A5A                   goto    cont
0258 1BD0           clear   btfsc   ACCaHI,msb_bit
0259 0AD8                   incf    ACCcLO
025A 1858           cont    btfsc   ACCcLO,lsb_bit  ;check the lsb of the dividend
025B 2A5E                   goto    minus
025C 2212                   call    Add24           ;if = 0, then add divisor to upper 24 bits
025D 2A5F                   goto    check           ;of dividend
025E 2221           minus   call    Sub24           ;if = 1, then subtract divisor from upper
                                                    ;24 bits of dividend
025F 0BDC           check   decfsz  temp,f          ;do 24 times
0260 2A4C                   goto    dloop
0261 1003                   bcf     STATUS,C
0262 0DD8                   rlf     ACCcLO          ;shift lower 24 bits of dividend 1 bit
0263 0DD7                   rlf     ACCcMID         ;position left
0264 0DD6                   rlf     ACCcHI
0265 1BD3                   btfsc   ACCbHI,msb_bit  ;exlusive or the inverse of the msb of the
0266 2A6A                   goto    w1              ;dividend with the msb of the divisor
0267 1FD0                   btfss   ACCaHI,msb_bit  ;store in the lsb of the dividend
0268 0AD8                   incf    ACCcLO
0269 2A6C                   goto    wzd
026A 1BD0           w1      btfsc   ACCaHI,msb_bit
026B 0AD8                   incf    ACCcLO
026C 1FD3           wzd     btfss   ACCbHI,msb_bit  ;if the msb of the remainder is set and
026D 2A71                   goto    wend
026E 1BD0                   btfsc   ACCaHI,msb_bit  ;the msb of the divisor is not
026F 2A71                   goto    wend
0270 2212                   call    Add24           ;add the divisor to the remainder to correct
                                                    ;for zero partial remainder
0271 3400           wend    retlw   0               ;quotient in 24 lsb's of dividend
                                                    ;remainder in 24 msb's of dividend
0272                Dsetup
0272 3018                   movlw   0x18            ;loop 24 times
0273 00DC                   movwf   temp
0274 3400                   retlw   0
                    ;_________________________________________________________________
                    ;*****************************************************************
                    ;*-------------------------------------------------------------*
                    ;*-     2's Complement
                    ;*-
                    ;*-     Uses ACCa
                    ;*-
                    ;*-     Performs 2's complement conversion on ACCa
                    ;*-------------------------------------------------------------*
                    ;*****************************************************************
```



```
0275                 compA
0275 09D2                    comf    ACCaLO          ;invert all bits in accumulator a
0276 09D1                    comf    ACCaMID
0277 09D0                    comf    ACCaHI
0278 0AD2                    incf    ACCaLO          ;add one to accumulator a
0279 1903                    btfsc   STATUS,Z
027A 0AD1                    incf    ACCaMID
027B 1903                    btfsc   STATUS,Z
027C 0AD0                    incf    ACCaHI
027D 3400                    retlw   0
                     ;____________________________________________________________________

                             END


0000 : X---X----------- XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX
0040 : XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX

0080 : XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX
00C0 : XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX

0100 : XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX
0140 : XXXXXXXXXXXXXXX- ---------------- ---------------- ----------------

0200 : XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX
0240 : XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX XXXXXXXXXXXXXX--

All other memory blocks unused.


Errors   :    0
Warnings :    0
Messages :    0
```

Please check the Microchip BBS for the latest version of the source code. For BBS access information, see Section 6, Microchip Bulletin Board Service, page 6-3.

APPENDIX B: VISUAL BASIC PROGRAM

PICMTR.FRM

```
Sub Form_Load ()
    'Initialize the program
    Image1.Height = 600
    Image1.Width = 2700
    Frame1.Caption = "PICMETER Power Off"
    Label1.Caption = ""
    Label2.Caption = ""

    'Initialize Comm Port 1
    Comm1.RThreshold = 1
    Comm1.Handshaking = 0
    Comm1.Settings = "9600,n,8,1"
    Comm1.CommPort = 2
    Comm1.PortOpen = True

    'Initialize the global variable First%
    First% = 0
End Sub

Sub Form_Unload (Cancel As Integer)
    'Unload PICMETER
    Comm1.RTSEnable = False
    Comm1.DTREnable = False
    Comm1.PortOpen = False
    Unload PICMETER
End Sub

Sub Comm1_OnComm ()
    Dim Value As Double
    Dim High As Double
    Dim Medium As Double
    Dim Low As Double

    'Received a character
    If Comm1.CommEvent = 2 Then
        If First% = 0 Then
            If Comm1.InBufferCount = 16 Then
                Label1.FontSize = 10
                InString$ = Comm1.Input
                If InString$ = "PICMETER Booted!" Then
                    Frame1.Caption = "PICMETER Booted!"
                End If
                First% = 1
                Comm1.InputLen = 4
            End If
        Else
            If Comm1.InBufferCount >= 4 Then
                InString$ = Comm1.Input
                If Left$(InString$, 1) = "R" Then
                    Frame1.Caption = "Resistance"
                    Label2.FontName = "Symbol"
                    Label2.Caption = "KW"
                    Label1.FontSize = 24
                ElseIf Left$(InString$, 1) = "C" Then
                    Frame1.Caption = "Capacitance"
                    Label2.FontName = "MS Sans Serif"
                    Label2.Caption = "nF"
                    Label1.FontSize = 24
                ElseIf Left$(InString$, 1) = "E" Then
                    Frame1.Caption = "Error Detected"
                    Label2.Caption = ""
                ElseIf Left$(InString$, 1) = "S" Then
                    Frame1.Caption = "Measuring Component"
```

```
                Label2.Caption = ""
            Else
                Frame1.Caption = "Error Detected"
                Label2.Caption = ""
            End If

            If Frame1.Caption = "Error Detected" Then
                Label1.Caption = ""
            ElseIf Frame1.Caption = "Measuring Component" Then
                Label1.Caption = ""
            Else
                High = 65536# * Asc(Mid$(InString$, 2, 1))
                Medium = 256# * Asc(Mid$(InString$, 3, 1))
                Low = Asc(Mid$(InString$, 4, 1))
                Label1.Caption = Format$((High + Medium + Low) / 1000, "###0.0")
            End If
        End If
    End If
  End If
End Sub

Sub Check3D1_Click (Value As Integer)
    'Control Power to the PICMETER
    If Check3D1.Value = False Then
        Comm1.InputLen = 0
        Label1.Caption = ""
        Label2.Caption = ""
        Comm1.RTSEnable = False
        Comm1.DTREnable = False
        Frame1.Caption = "PICMETER Power Off"
        InString$ = Comm1.Input
    Else
        Frame1.Caption = ""
        First% = 0
        Comm1.InputLen = 0
        InString$ = Comm1.Input
        Comm1.RTSEnable = True
        Comm1.DTREnable = True
    End If
End Sub

Sub menExitTop_Click ()
    'Unload PICMETER
    Unload PICMETER
End Sub

Sub Option1_Click ()
    'Open COM1 for communications
    If Option1.Value = True Then
        If Comm1.CommPort = 2 Then
            Comm1.PortOpen = False
            Comm1.CommPort = 1
            Comm1.PortOpen = True
        End If
    End If
End Sub

Sub Option2_Click ()
    'Open COM2 for communications
    If Option2.Value = True Then
        If Comm1.CommPort = 1 Then
            Comm1.PortOpen = False
            Comm1.CommPort = 2
            Comm1.PortOpen = True
        End If
    End If
End Sub
```

PICMETER.BAS

```
Global I%
Global First%
```

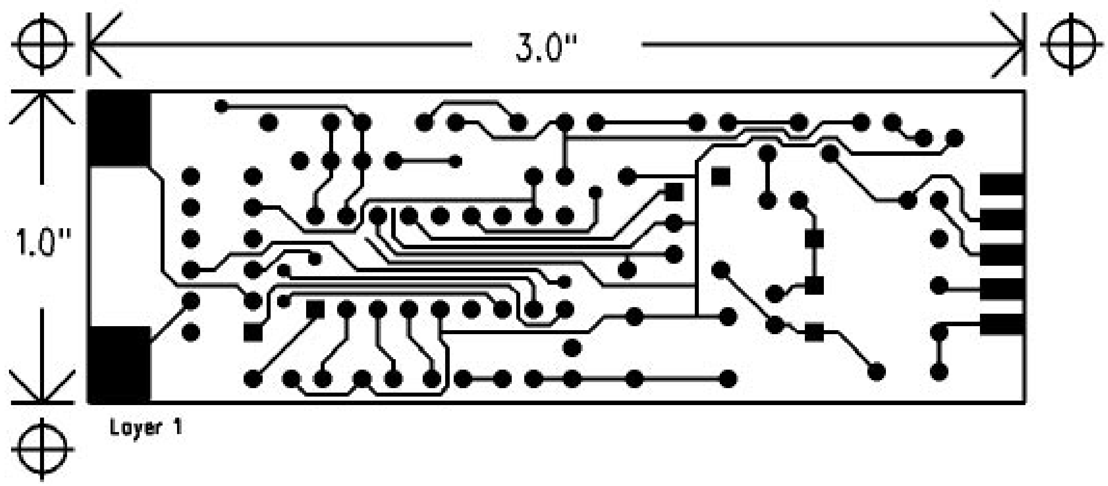
APPENDIX C: PICMETER PCB LAYOUT

Boards Manufactured by: Southwest Circuits

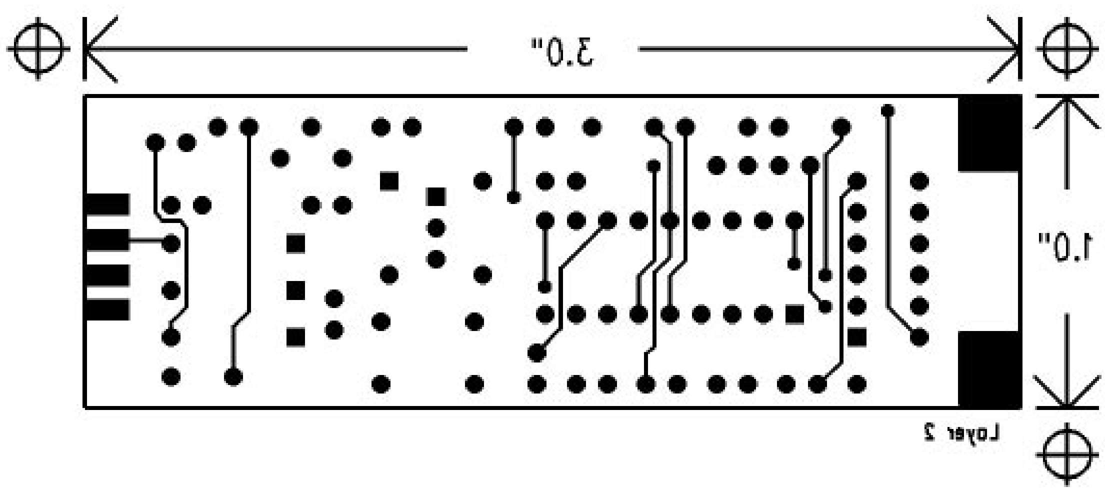
Contact: Perry Groves
3760 E. 43rd Place
Tucson, AZ 85713
1-520-745-8515

The following artwork is not printed to scale:

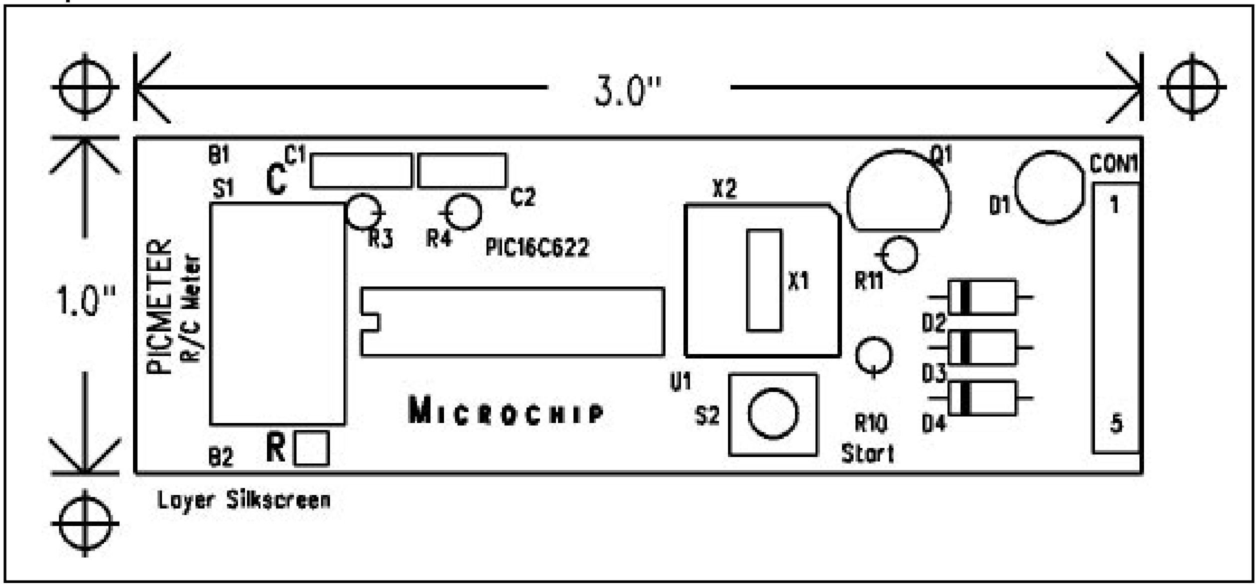
Component Side



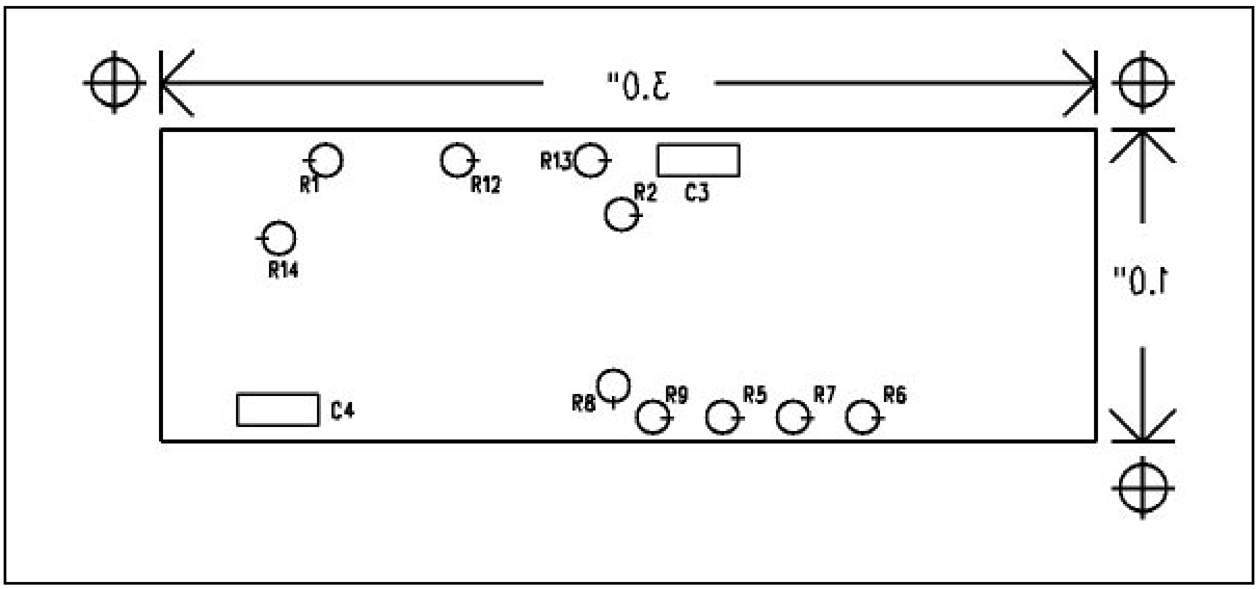
Solder Side



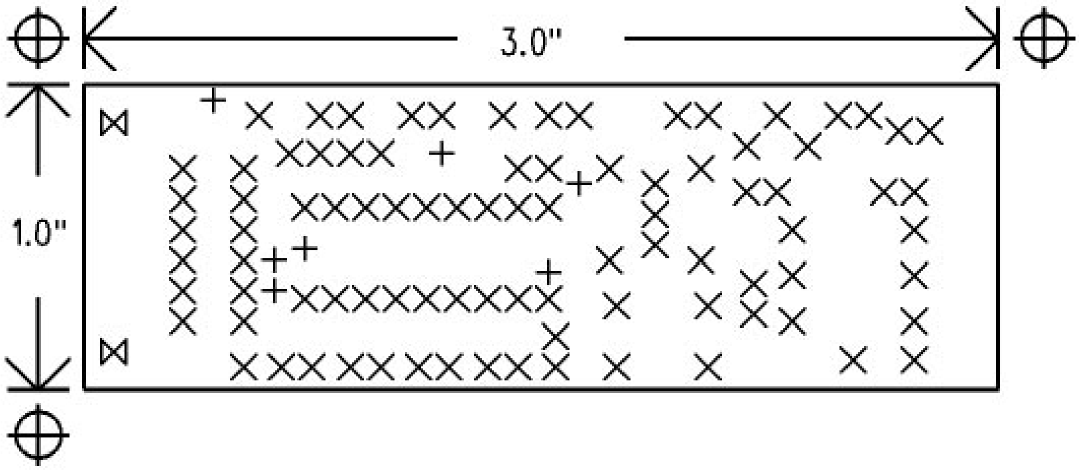
Component Side Silkscreen



Solder Side Silkscreen



Manufacturing Drawing



SIZE	QTY	SYM
18	7	+
37	89	×
95	2	⋈

NOTES:

WORLDWIDE SALES & SERVICE

AMERICAS

Corporate Office
Microchip Technology Inc.
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 602 786-7200 Fax: 602 786-7277
Technical Support: 602 786-7627
Web: http://www.mchip.com/microchip

Atlanta
Microchip Technology Inc.
500 Sugar Mill Road, Suite 200B
Atlanta, GA 30350
Tel: 770 640-0034 Fax: 770 640-0307

Boston
Microchip Technology Inc.
5 Mount Royal Avenue
Marlborough, MA 01752
Tel: 508 480-9990 Fax: 508 480-8575

Chicago
Microchip Technology Inc.
333 Pierce Road, Suite 180
Itasca, IL 60143
Tel: 708 285-0071 Fax: 708 285-0075

Dallas
Microchip Technology Inc.
14651 Dallas Parkway, Suite 816
Dallas, TX 75240-8809
Tel: 214 991-7177 Fax: 214 991-8588

Dayton
Microchip Technology Inc.
35 Rockridge Road
Englewood, OH 45322
Tel: 513 832-2543 Fax: 513 832-2841

Los Angeles
Microchip Technology Inc.
18201 Von Karman, Suite 455
Irvine, CA 92715
Tel: 714 263-1888 Fax: 714 263-1338

New York
Microchip Technology Inc.
150 Motor Parkway, Suite 416
Hauppauge, NY 11788
Tel: 516 273-5305 Fax: 516 273-5335

San Jose
Microchip Technology Inc.
2107 North First Street, Suite 590
San Jose, CA 95131
Tel: 408 436-7950 Fax: 408 436-7955

ASIA/PACIFIC

Hong Kong
Microchip Technology
Unit No. 3002-3004, Tower 1
Metroplaza
223 Hing Fong Road
Kwai Fong, N.T. Hong Kong
Tel: 852 2 401 1200 Fax: 852 2 401 3431

Korea
Microchip Technology
168-1, Youngbo Bldg. 3 Floor
Samsung-Dong, Kangnam-Ku,
Seoul, Korea
Tel: 82 2 554 7200 Fax: 82 2 558 5934

Singapore
Microchip Technology
200 Middle Road
#10-03 Prime Centre
Singapore 188980
Tel: 65 334 8870 Fax: 65 334 8850

Taiwan
Microchip Technology
10F-1C 207
Tung Hua North Road
Taipei, Taiwan, ROC
Tel: 886 2 717 7175 Fax: 886 2 545 0139

EUROPE

United Kingdom
Arizona Microchip Technology Ltd.
Unit 6, The Courtyard
Meadow Bank, Furlong Road
Bourne End, Buckinghamshire SL8 5AJ
Tel: 44 0 1628 851077 Fax: 44 0 1628 850259

France
Arizona Microchip Technology SARL
2 Rue du Buisson aux Fraises
91300 Massy - France
Tel: 33 1 69 53 63 20 Fax: 33 1 69 30 90 79

Germany
Arizona Microchip Technology GmbH
Gustav-Heinemann-Ring 125
D-81739 Muenchen, Germany
Tel: 49 89 627 144 0 Fax: 49 89 627 144 44

Italy
Arizona Microchip Technology SRL
Centro Direzionale Colleoni
Palazzo Pegaso Ingresso No. 2
Via Paracelso 23, 20041
Agrate Brianza (MI) Italy
Tel: 39 039 689 9939 Fax: 39 039 689 9883

JAPAN

Microchip Technology Intl. Inc.
Benex S-1 6F
3-18-20, Shin Yokohama
Kohoku-Ku, Yokohama
Kanagawa 222 Japan
Tel: 81 45 471 6166 Fax: 81 45 471 6122

9/22/95

MICROCHIP