INTRODUCTION

The PIC16C62X devices create a new branch in Microchip's PIC16CXXX 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 PIC16CXXX features such as:

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

<table>
<thead>
<tr>
<th>Device</th>
<th>Program Memory</th>
<th>Data Memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIC16C620</td>
<td>512 x 14</td>
<td>80 x 8</td>
</tr>
<tr>
<td>PIC16C621</td>
<td>1K x 14</td>
<td>80 x 8</td>
</tr>
<tr>
<td>PIC16C622</td>
<td>2K x 14</td>
<td>128 x 8</td>
</tr>
</tbody>
</table>

This family of devices also introduce on-chip brown-out reset circuitry and a filter on the reset input (MCLR) to the PIC16CXXX mid-range microcontrollers. Brown-out Reset holds the device in reset while $V_{DD}$ is below the Brown-out Reset voltage of 4.0V ± 0.2V. The reset filter is used to filter out glitches on the 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

```
<table>
<thead>
<tr>
<th>bit7</th>
<th>bit6</th>
<th>bit5-4</th>
<th>bit3</th>
<th>bit2-0</th>
</tr>
</thead>
<tbody>
<tr>
<td>C2OUT</td>
<td>C1OUT</td>
<td>—</td>
<td>CIS</td>
<td>CM2:CM0</td>
</tr>
</tbody>
</table>

| bit7: C2OUT: Comparator 2 output |
| 1 = C2 VIN+ > C2 VIN– |
| 0 = C2 VIN+ < C2 VIN– |

| bit6: C1OUT: Comparator 1 output |
| 1 = C1 VIN+ > C1 VIN– |
| 0 = C1 VIN+ < C1 VIN– |

| bit5-4: Unimplemented: Read as '0' |

| bit3: CIS: Comparator Input Switch |
| When CM2:CM0 = 001: |
| 1 = C1 VIN– connects to RA3 |
| 0 = C1 VIN– connects to RA0 |
| When CM2:CM0 = 010: |
| 1 = C1 VIN– connects to RA3 |
| 2 VIN– connects to RA2 |
| 0 = C1 VIN– connects to RA0 |
| 2 VIN– connects to RA1 |

| bit2-0: CM2:CM0: Comparator mode |
```

R = Readable bit
W = Writable bit
U = Unimplemented bit, read as '0'
- n = Value at POR reset
A single comparator is shown in Figure 2. The relationship between the inputs and the output is also shown. When the voltage at \( V_{IN+} \) is less than the voltage at \( V_{IN-} \), the output of the comparator is at a digital low level. When the voltage at \( V_{IN+} \) is greater than the voltage at \( V_{IN-} \), 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 \( V_{SS} \) and \( V_{DD} \) and one input must be in the Common Mode Range (CMR). The CMR is defined as \( V_{DD}-1.5 \text{ volt} \) to \( V_{SS} \). 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 \( V_{DD} \) or below \( V_{SS} \) by more than 0.6 volt, the microcontroller may draw excessive current. A maximum source impedance to the comparators of 10 k\( \Omega \) 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.
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**

<table>
<thead>
<tr>
<th>CIS</th>
<th>C1 VIN-</th>
<th>C2 VIN-</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>RA0</td>
<td>RA1</td>
</tr>
<tr>
<td>1</td>
<td>RA3</td>
<td>RA2</td>
</tr>
</tbody>
</table>

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.

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.

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

![Figure 10: Three Inputs Multiplexed to Two Comparators](image)

The comparators will remain active if the device is placed in sleep mode, except for the Comparators Off Mode (CM2:CM0 =111) and Comparators Reset Mode (CM2:CM0=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.

### 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.

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.

### EXAMPLE 1: INITIALIZING THE COMPARATOR MODULE

```assembly
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 comparator interrupts
BSF STATUS,RP0 ;go to Bank 0
BSF INTCON,PEIE ;enable global interrupts
BSF INTCON,GIE ;and peripheral interrupts
```

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.
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

\[
\begin{array}{cccccccc}
\text{bit7} & \text{VREN} & \text{VROE} & \text{VRR} & \text{VR3} & \text{VR2} & \text{VR1} & \text{VR0} \\
\end{array}
\]

\text{Register: VRCON}  \\
\text{Address: 9Fh}  \\
\text{POR Value: 00h}  \\
\text{R: Readable}  \\
\text{W: Writable}  \\
\text{U: Unimplemented, read as '0'}

\text{VR3:VR0: VREF value selection } 0 \leq V_{<3:0>} \leq 15 \\
\text{when VRR = 1: } V_{REF} = (V_{<3:0>} / 24) \times V_{DD} \\
\text{when VRR = 0: } V_{REF} = 1/4 \times V_{DD} + (V_{<3:0>} / 32 \times V_{DD})

\text{VRR: VREF Range selection}  \\
1 = \text{Low Range}  \\
0 = \text{High Range}  \\

\text{VROE: VREF Output Enable}  \\
1 = \text{VREF is output on RA2 pin}  \\
0 = \text{VREF is disconnected from RA2 pin}  \\

\text{VREN: VREF Enable}  \\
1 = \text{VREF circuit powered on}  \\
0 = \text{VREF circuit powered down, no I\textsubscript{DD} drain}

FIGURE 12: VOLTAGE REFERENCE BLOCK DIAGRAM

Note: The voltage reference is V\textsubscript{DD} derived and therefore, the V\textsubscript{REF} output changes with fluctuations in V\textsubscript{DD}. 
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 (VROE). 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 VREF 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 VREF and the value of the input signal or the digital output.

The full range of VSS to VDD 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 VREF from approaching VSS or VDD. Equation 1 and Equation 2 are used to calculate the output of the voltage reference.

EQUATION 1: VOLTAGE REFERENCE EQUATION, VRR = 1

$$V_{REF} = \frac{VR_{<3:0>}}{24} \times V_{DD}$$

EQUATION 2: VOLTAGE REFERENCE EQUATION, VRR = 0

$$V_{REF} = \frac{V_{DD}}{4} + \frac{VR_{<3:0>}}{32} \times V_{DD}$$

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 VDD 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 ; 10 \mu s 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 \mu s to allow VREF to stabilize
4. Delay 10 \mu s 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 VREF 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 VREF is used as an output, an external buffer must be used.

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 \( V_{\text{AIN}} \). 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 \( V_{\text{AIN}} \).

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 \( V_{\text{AIN}} \) 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 \( V_{\text{AIN}} \) slope. The length of time the reference slope requires to return to zero is proportional to the value of \( V_{\text{AIN}} \). 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.
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 $C_1$ 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 switch $S_1$ 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 integrating capacitor $C_1$ to discharge completely. The other input is multiplexed with ground, reference, and $V_{AIN}$ through switch $S_2$. 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 $V_{AIN}$ signals. Assuming that the integrator ramps are highly linear, Equation 3 is used to determine the value of $V_{AIN}$.

**EQUATION 3: OUTPUT EQUATION FOR THE MODIFIED-SLOPE CONVERTER**

$$V_{AIN} = \frac{t_{AIN} - t_{GND}}{t_{VREF} - t_{GND}} \times K \text{ mV}$$

where $t_{AIN}$ is the measured time for the $V_{AIN}$ signal, $t_{VREF}$ is the measured time for the voltage reference signal, $t_{GND}$ 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 converter. The linear slope and integrator of Figure 13 are replaced with the exponential charge waveform of an RC. 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 1Ω to 999Ω 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

\[ R_{UNK} = \frac{t_{UNK} \times R_{KN}}{t_{KN}} \]

EQUATION 5: CAPACITANCE EQUATION

\[ C_{UNK} = \frac{t_{UNK} \times C_{KN}}{t_{KN}} \]

RUNK and CUNK are the unknown resistor and capacitor values. RKN and CKN are the known resistor and capacitor values. tUNK and tKN 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 VREF 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 $t_{UNK}$.

The analog modules are now disabled. The comparator interrupts are disabled and the comparators are turned off (CM2:CM0=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, $t_{UNK}$, 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, $t_{KN}$. 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 TMR0”. 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 ns. Therefore, the highest count that the time registers should reach is 999,000. This is a 20-bit number. If the 21st 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>
<thead>
<tr>
<th>Operation</th>
<th>Result</th>
<th>Temp. Storage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add</td>
<td>ACCa + ACCb</td>
<td>ACCb</td>
</tr>
<tr>
<td>Subtract</td>
<td>2's Comp ACCa then</td>
<td>ACCa</td>
</tr>
<tr>
<td></td>
<td>ACCa + ACCb</td>
<td>ACCa</td>
</tr>
<tr>
<td>Multiply</td>
<td>ACCa x ACCb</td>
<td>ACCb (MSB's)</td>
</tr>
<tr>
<td>Divide</td>
<td>ACCb:ACCc</td>
<td>ACCa</td>
</tr>
<tr>
<td>2's Comp</td>
<td>NOT(ACCa) + 1</td>
<td>ACCa</td>
</tr>
</tbody>
</table>
Computer Program

The program that receives data from the PICMETER was written in Visual Basic® from Microsoft® for the Windows® environment. Figure 17 shows 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>
<thead>
<tr>
<th>Marked Value</th>
<th>Fluke Value</th>
<th>PICMETER Value</th>
<th>Error %</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 nF</td>
<td>2.3 nF</td>
<td>2.2 nF</td>
<td>4.3</td>
</tr>
<tr>
<td>2.5 nF</td>
<td>2.63 nF</td>
<td>2.5 nF</td>
<td>4.9</td>
</tr>
<tr>
<td>20 nF</td>
<td>16.5 nF</td>
<td>16.3 nF</td>
<td>1.2</td>
</tr>
<tr>
<td>33 nF</td>
<td>35.2 nF</td>
<td>35.8 nF</td>
<td>1.7</td>
</tr>
<tr>
<td>47 nF</td>
<td>45 nF</td>
<td>44.5 nF</td>
<td>1.1</td>
</tr>
<tr>
<td>50 nF</td>
<td>52 nF</td>
<td>52.9 nF</td>
<td>1.7</td>
</tr>
<tr>
<td>100 nF</td>
<td>99.7 nF</td>
<td>93 nF</td>
<td>6.7</td>
</tr>
<tr>
<td>0.1 µF</td>
<td>95 nF</td>
<td>96.1 nF</td>
<td>1.2</td>
</tr>
<tr>
<td>0.22 µF</td>
<td>215 nF</td>
<td>215.2 nF</td>
<td>0.1</td>
</tr>
<tr>
<td>470 nF</td>
<td>508 nF</td>
<td>518.9 nF</td>
<td>2.1</td>
</tr>
<tr>
<td>940 nF</td>
<td>922 nF</td>
<td>983.1 nF</td>
<td>6.6</td>
</tr>
</tbody>
</table>

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

An 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**

PIC16C62X devices add two significant analog features to the PIC16CXXX 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 PIC16CXXX mid-range family core create a new path to achieve high resolution results.
APPENDIX A: PICMETER FIRMWARE

MPASM 01.40 Released PICMETER.ASM 1-16-1997 17:49:07 PAGE 1

LOC OBJECT CODE LINE SOURCE TEXT
VALUE

00001 TITLE "PICMETER Firmware for PIC16C622"
00002 LIST P - 16C622
00003 ERRORLEVEL -302
00004
00005 INCLUDE <P16C622.INC>
00006
00007 __CONFIG __BODEN_OFF & __CP_OFF & __PWRT_OFF & __WDT_OFF & __XT_OSC
00008
00009 ;--------------------------------------------------------------------------------
00010 ;-----------------------------------------------------------*
00011 ;-----------------------------------------------------------*
00012 ; PICMETER - Resistance and Capacitance Meter
00013 ;-----------------------------------------------------------*
00014 ;-----------------------------------------------------------*
00015 ;-----------------------------------------------------------*
00016 ; Author: Rodger Richey
00017 ; Applications Engineer
00018 ; Filename: picmeter.asm
00019 ; Revision: 1 May 1995
00020 ; 1-16-97 ; For compatibility with MPASMWIN 1.40
00021 ;-----------------------------------------------------------*
00022 ;-----------------------------------------------------------*
00023 ;-----------------------------------------------------------*
00024 ; PICMETER is based on a PIC16C622 which has two comparators and a
00025 ; variable voltage reference. Resistance and capacitance is
00026 ; calculated by measuring the time constant of a RC network. The
00027 ; toggle switch selects either resistor or capacitor input. The
00028 ; pushbutton switch starts a measurement. The time constant of the
00029 ; unknown component is compared to that of known component to
00030 ; calculate the value of the unknown component. The following
00031 ; formulas are used:
00032 ;-----------------------------------------------------------*
00033 ;Resistance: Ru = ( Rk * Tu ) / Tk
00034 ;Capacitance: Cu = ( Ck * Tu ) / Tk
00035 ;-----------------------------------------------------------*
00036 ;-----------------------------------------------------------*
00037 ;-----------------------------------------------------------*
00038 ;-----------------------------------------------------------*
00039 ;-----------------------------------------------------------*
00040 ;--------------------------------------------------------------------------------
00041 ;-----------------------------------------------------------*
00042 ; RS232 code borrowed from Application Note AN593
00043 ; "Serial Port Routines Without Using Timer0"
00044 ; Author: Stan D'Souza
00045 ;-----------------------------------------------------------*
00046 ;--------------------------------------------------------------------------------
003D0900 xtal equ .4000000
00002580 baud equ .9600
000F4240 fclk equ xtal/4
00050 ; The value baudconst must be a 8-bit value only
00051 ;-----------------------------------------------------------*
00052 baudconst equ ((fclk/ baud)/3-2)
Bit Equates

BEGIN equ 0 ;begin a measurement flag
DONE equ 7 ;done measuring flag
WHICH equ 5 ;R or C measurement flag
F_ERROR equ 3 ;error detection flag
EMPTY equ 5 ;flag if component is connected
V0 equ 0 ;power for R reference ckt
V1 equ 1 ;power for C reference ckt
V2 equ 2 ;ground for C reference ckt
V3 equ 3 ;power for unknown R ckt
V4 equ 4 ;ground for unknown C ckt
msb_bit equ 7 ;define for bit 7
lsb_bit equ 0 ;define for bit 0
RkHI equ 0x07 ;value of the known resistance, R4, in ohms
RkMID equ 0x9D ;measured by a Fluke meter
RkLO equ 0x38
CkHI equ 0x07 ;value of the known capacitance, C1, in pF
CkMID equ 0x47 ;measured by a Fluke meter
CkLO equ 0x48

User Registers

Bank 0
W_TEMP equ 0x20 ;Bank 0 temporary storage for W reg
STATUS_TEMP equ 0x21 ;temporary storage for STATUS reg
Ttemp equ 0x23 ;temporary Time register
flags equ 0x24 ;flags register
count equ 0x25 ;RS232 register
txreg equ 0x26 ;RS232 data register
delay equ 0x27 ;RS232 delay register
offset equ 0x28 ;table position register
msb equ 0x29 ;general delay register
offset equ 0x2A ;general delay register
TimeLO equ 0x40 ;Time registers
TimeMID equ 0x41
TimeHI equ 0x42

Math related registers
ACCaHI equ 0x50 ;24-Bit accumulator a
ACCaMID equ 0x51
ACCaLO equ 0x52
ACCbHI equ 0x53 ;24-Bit accumulator b
ACCbMID equ 0x54
ACCbLO equ 0x55
ACChHI equ 0x56 ;24-Bit accumulator c
ACChMID equ 0x57
ACChLO equ 0x58
ACCdHI equ 0x59 ;24-Bit accumulator d
ACCdMID equ 0x5A
ACCdLO equ 0x5B
temp equ 0x5C ;temporary storage

tx = PORTB,7 ;define for RS232 TXD output pin
0000 00119  org 0x0
0000 2810 00120  goto init
00121
0004 00122  org 0x4
0004 28B9 00123  goto ServiceInterrupts
00124
0010 00125  org 0x10
0010 00126 init
0010 1283 00127  bcf STATUS,RP0 ;select bank 0
0011 0185 00128  clrf PORTA ;clear PORTA and PORTB
0012 0186 00129  clrf PORTB
0013 1786 00130  bsf tx ;set TXD output pin
0014 01A4 00131  clrf flags ;clear flags register
0015 3010 00132  movlw 0x10 ;load table offset register
0016 00A8 00133  movwf offset
0017 00B7 00134  clrf INTCON ;clear interrupt flags and disable interrupts
0018 3007 00135  movlw 0x07 ;turn off comparators, mode 111
0019 009F 00136  movwf CMCON
001A 2140 00137  call delay20 ;wait for comparators to settle
0018 008F 00138  movf CMCON,F
001B 3088 00139  movlw 0x88 ;WDT prescaler, internal TMR0 increment
001C 0081 00140  movwf OPTION_REG
001D 138B 00141  bcf TRISA ;PORTA all outputs, discharges RC ckt
001E 3060 00142  movlw 0x60 ;PORTA<7,4:0> outputs, PORTA<6:5> inputs
001F 0185 00143  clrf TRISB
0020 1283 00144  bcf STATUS,RP0 ;select bank 1
0021 3008 00145  movlw 0x10 ;WDT prescaler, internal TMR0 increment
0022 0012 00146  goto start ;wait for a start measurement key press
0023 0012 00147  bcf STATUS,RP0 ;select bank 1
0024 0013 00148  bsf TX ;set TXD output pin
0025 0014 00149  movlw 0x08 ;enable RBIE interrupt
0026 0015 00150  movwf INTCON
0027 213D 00151  call vlong ;delay before transmitting boot message
0028 213D 00156  call vlong ;to allow computer program to set up
0029 213D 00157  call vlong ;allow computer program to set up
002A 213D 00158  call vlong ;transmit boot message
002B 213D 00159  call vlong ;transmit boot message
002C 178B 00160  bsf INTCON,GIE ;enable global interrupt bit
00156
0021 0007 00161  start
0022 1C24 00162  btfs flags,BEGIN ;wait for a start measurement key press
0023 282D 00163  goto start
0024 00164  btfs flags,BEGIN ;clear start measurement flag
0025 00165  bcf INTCON,GIE ;transmit a start measurement message
0026 3053 00166  movlw 'S' ;to the PC
0027 20AD 00167  call Send
0028 178B 00168  bsf INTCON,GIE
00166
0029 01C2 00169  clrf TimeHI ;reset Time registers
0030 01C1 00170  clrf TimeMID
0031 01C0 00171  clrf TimeLO
0032 18E6 00172  btfs PORTB,WHICH ;detect if resistor or capacitor measure
0033 2862 00173  goto Capacitor
00172
0034 19A4 00174  bcf STATUS,RP0 ;set V0 to input
0035 1406 00175  bcf TRISB,V0
0036 1283 00176  bcf STATUS,RP0
0037 20FB 00177  call AnalogOn ;turn analog on
0038 0181 00178  clrf TMR0
0039 0000 00179  nope
003A 1586 00180  bcf PORTB,V3 ;turn power on to unknown RC ckt
0040 0181 00181  btfsc flags,F_ERROR ;detect if an error occurs
0041 288B 00182  goto ErrorDetect
0042 1FA4 00183  btfs flags,DONE ;measurement completed flag
0043 2840 00184  goto RwaitU
0044 13A4 00185  bcf  flags,DONE  ;clear measurement completed flag
0045 2111 00186  call  AnalogOff  ;turn analog off
0046 2126 00188  call  SwapTtoA  ;swap Time to accumulator a
0047 3007 00189  movlw  RkHI  ;swap known resistance value
0048 0003 00190  movwf  ACCbHI  ;to accumulator b
0049 309D 00191  movlw  RkMID
004A 00D4 00192  movwf  ACCbMID
004B 3038 00193  movlw  RkLO
004C 00D5 00194  movwf  ACCbLO
004D 2230 00195  call  Mpy24  ;multiply accumulator a and b
004E 1683 00197  bsf  STATUS,RP0  ;set V3 to input
004F 1586 00198  bsf  TRISB,V3
0050 1283 00199  bcf  STATUS,RP0
0051 20FB 00200  call  AnalogOn  ;turn analog on
0052 0181 00201  clrf  TMR0
0053 0000 00202  nop
0054 1406 00203  bsf  PORTB,V0  ;turn power on to known RC ckt
0055 19A4 00204  RwaitK  btfsc  flags,F_ERROR  ;detect if an error occurs
0056 288B 00205  goto  ErrorDetect
0057 1FA4 00206  btfss  flags,DONE  ;measurement completed flag
0058 2855 00207  goto  RwaitK
0059 13A4 00208  bcf  flags,DONE  ;clear measurement completed flag
005A 2111 00209  call  AnalogOff  ;turn analog off
005B 2126 00211  call  SwapTtoA  ;swap Time to accumulator a
005C 224B 00212  call  Div24  ;divide multiply by known time
005D 138B 00214  bcf  INTCON,GIE  ;disable all interrupts
005E 3052 00215  movlw  'R'  ;transmit, for R measurement
005F 20AD 00216  call  Send
0060 178B 00217  bsf  INTCON,GIE  ;enable global interrupt bit
0061 282D 00218  goto  start  ;restart
0062 0220 00219  Capacitor
0063 1683 00221  bsf  STATUS,RP0  ;set V2 to input
0064 1506 00222  bcf  TRISB,V2
0065 1283 00223  bcf  STATUS,RP0
0066 0181 00225  clrf  TMRO
0067 0000 00226  nop
0068 1486 00227  bsf  PORTB,V1  ;turn power on to unknown RC ckt
0069 19A4 00228  CwaitU  btfsc  flags,F_ERROR  ;detect if an error occurs
006A 288B 00229  goto  ErrorDetect
006B 1FA4 00230  btfss  flags,DONE  ;measurement completed flag
006C 2869 00231  goto  CwaitU
006D 13A4 00232  bcf  flags,DONE  ;clear measurement completed flag
006E 2111 00233  call  AnalogOff  ;turn analog off
006F 2126 00235  call  SwapTtoA  ;swap Time to accumulator a
0070 3007 00236  movlw  CkHI  ;swap known resistance value
0071 0003 00237  movwf  ACCbHI  ;to accumulator b
0072 3047 00238  movlw  CkMID
0073 0044 00239  movwf  ACCbMID
0074 3048 00240  movlw  CkLO
0075 0005 00241  movwf  ACCbLO
0076 2230 00242  call  Mpy24  ;multiply accumulator a and b
0077 1683 00244  bsf  STATUS,RP0  ;set V3 to input
0078 1606 00245  bcf  TRISB,V4
0079 1283 00246  bcf  STATUS,RP0
007A 20FB 00247  call  AnalogOn  ;turn analog on
007B 0181 00248  clrf  TMRO
007C 0000 00249  nop
007D 1486 00250  bsf  PORTB,V1  ;turn power on to known RC ckt
007E 19A4 00251 CwaitK btfsc flags, F_ERROR ;detect if an error occurs
007F 288B 00252 goto ErrorDetect ;measurement completed flag
0080 1FA4 00253 btfss flags, DONE ;clear measurement completed flag
0081 287E 00254 goto CwaitK
0082 13A4 00255 bcf flags, DONE
0083 2111 00256 call AnalogOff ;turn analog off
0084 2126 00257 call SwapTtoA ;swap Time to accumulator a
0085 224B 00258 call Div24 ;divide multiply by known time
0086 138B 00259 bcf INTCON, GIE ;disable all interrupts
0087 3043 00260 movlw 'C' ;transmit, for C measurement
0088 20AD 00261 call Send
0089 178B 00262 bcf INTCON, GIE ;enable global interrupt bit
008A 282D 00263 goto start ;restart
008B 00266 ErrorDetect
008C 00267 ;************************************************************************
008D ;*----------------------------------------------------------------------*
008E ;* RS232 Transmit Routine                                            *
008F ;* Borrowed from AN593, "Serial Port Routines Without Using Timer0" *
008G ;* Author: Stan D'Souza                                           *
008H ;* This is the routine that interfaces directly to the hardware     *
008I ;************************************************************************
008J ;************************************************************************
008K 00288 Transmit
008L 00289 bcf STATUS, RP0 ;disable TMR0
008M 00290 movwf txreg ;send start bit
008N 00291 bcf tx
008O 00292 movlw baudconst
008P 00293 movwf delay
008Q 00294 movlw 0x9
008R 00295 movwf count
008S 00296 txbaudwait
008T 00297 decfsz delay, F
008U 00298 goto txbaudwait
008V 00299 movlw baudconst
008W 00300 movwf delay
008X 00301 decfsz count, F
008Y 00302 goto SendNextBit
008Z 00303 movlw 0x9
008A 00304 movwf count
008B 00305 bcf tx ;send stop bit
008C 00306 return
008D 00307 SendNextBit
008E 00308 rrf txreg, F
008F 00309 btfss STATUS, C
0090 00310 goto Setlo
0091 00311 bcf tx
0092 00312 goto txbaudwait
0093 00313 Setlo bcf tx
0094 00314 goto txbaudwait
0095 ;________________________________________________________________________
;***********************************************************************
;*---------------------------------------------------------------------*
;*-     Borrowed from AN552, "Implementing Wake-up on Key Stroke"          -*
;*-     Author: Stan D’Souza                                          -*
;*----------------------------------------------------------------------*
;************************************************************************

00B9 0043 ServiceKeystroke
00B9 00A0 0034 movwf W_TEMP ;Pseudo push instructions
00BA 0E03 00345 swapf STATUS,W
00BB 1283 00346 bcf STATUS,RPO
00BC 00A1 00347 movwf STATUS_TEMP
00348
00BD 0801 00349 movf TMRO,W
00BE 00A3 00350 movwf Ttemp
00BF 190B 00351 btfsc INTCON,T0IF ;Service Timer0 overflow
00C0 20E5 00352 call ServiceTimer
00C1 1B0C 00353 btfsc PIR1,CMIF ;Stops Timer0, Records Value
00C2 20EC 00354 call ServiceComparator
00C3 180B 00355 btfsc INTCON,RBIF ;Service pushbutton switch
00C4 20CB 00356 call ServiceKeystroke ;Starts a measurement
00357
00C5 1283 00358 bcf STATUS,RPO
00C6 0E21 00359 swapf STATUS_TEMP,W ;Pseudo pop instructions
00C7 0083 00360 movwf STATUS
00C8 0EA0 00361 swapf W_TEMP,F
00C9 0E20 00362 swapf W_TEMP,W
00363
00CA 0009 00364 retfie
00365 ;_______________________________________________________________________
00366
00367 ;***********************************************************************
00368 ;*---------------------------------------------------------------------*
00369 ;*-     Borrowed from AN552, "Implementing Wake-up on Key Stroke"          -*
00370 ;*-     Author: Stan D’Souza                                          -*
;*----------------------------------------------------------------------*
;************************************************************************

00CB 0037 ServiceKeystroke
00CB 118B 00374 bcf INTCON,RBIE ;disable interrupt
00CC 0906 00375 comf PORTB,W ;read PORTB
00CD 100B 00376 bcf INTCON,RBIF ;clear interrupt flag
00CE 3940 00377 andlw B’01000000’
00CF 1903 00378 btfsc STATUS,Z
00D0 2806 00379 goto NotSwitch
00D1 2143 00380 call delay16 ;de-bounce switch for 16ms
00D2 0906 00381 comf PORTB,W ;read PORTB again
00D3 2009 00382 call KeyRelease ;check for key release
bsf flags,BEGIN
return

NotSwitch ;detected other PORTB pin change
bcf INTCN,RBIF ;reset RBI interrupt
bsf INTCN,RBIE

return

NotSwitch                       ;detected other PORTB pin change
bcf INTCN,RBIF     ;reset RBI interrupt
bsf INTCN,RBIE     ;enable interrupt
andw B’01000000’
btfsc STATUS,Z        ;key still pressed?
return                  ;if no, then return
sleep                   ;else, save power
bcf INTCN,RBIE     ;disable interrupts
comf PORTB,W         ;read PORTB
bcf INTCN,RBIF     ;clear flag
andlw B’01000000’
btfsc STATUS,Z        ;key still pressed?
return                  ;if no, then return
sleep                   ;else, save power
bcf INTCN,RBIE     ;disable interrupts
comf PORTB,W         ;read PORTB
bcf INTCN,RBIF     ;clear flag

;_______________________________________________________________________
;***********************************************************************
;*---------------------------------------------------------------------*
;*-     ISR to service a Timer0 overflow                              -*
;*---------------------------------------------------------------------*
;***********************************************************************
incf TimeMID,F       ;increment middle Time byte
btfsc STATUS,Z        ;if middle overflows, 
incf TimeHI,F       ;increment high Time byte
bsf TimeHI,EMPTY    ;check if component is connected
bsf flags,F_ERROR   ;set error flag
bcf INTCN,T0IF     ;clear TMR0 interrupt flag
return

;_______________________________________________________________________
;***********************************************************************
;*---------------------------------------------------------------------*
;*-     ISR to service a Comparator interrupt                         -*
;*---------------------------------------------------------------------*
;***********************************************************************
bcf STATUS,RP0      ;select bank 0
btfss PORTB,WHICH     ;detect which measurement, R or C?
goto capcomp
btfsc CMCON,C2OUT     ;detect if C ckt has interrupted
 goto    scend
btfsc CMCON,C2OUT     ;detect if C ckt has interrupted
 goto    scend
bcf INTCN,T0IF     ;clear TMR0 interrupt flag

;_______________________________________________________________________
;***********************************************************************
;*---------------------------------------------------------------------*
;***********************************************************************
;*---------------------------------------------------------------------*
;*---------------------------------------------------------------------*
;*---------------------------------------------------------------------*
00449 ;*- Turn Comparators and Vref On -*
00450 ;*---------------------------------------------------------------------*
00451 ;***********************************************************************
00452 AnalogOn
00453 1283 bcf STATUS,RP0 ;select bank 0
00454 3002 movlw 0x02 ;turn comparators on, mode 010
00455 009F movwf CMCON ;4 inputs multiplexed to 2 comparators
00456 1683 bcf STATUS,RP0 ;select bank 1
00457 300F movlw 0x0F ;make PORTA<3:0> all inputs
00458 0085 movwf TRISA
00459 179F bcf VRCON,VREN
00460 1283 bcf STATUS,RP0 ;select bank 0
00461 2140 call delay20 ;20msec delay
00462 089F movf CMCON,F ;clear comparator mismatch condition
00463 130C bcf PIR1,CMIF ;clear comparator interrupt flag
00464 1683 bcf STATUS,RP0
00465 170C bcf PIE1,CMIE ;enable comparator interrupts
00466 1283 bcf STATUS,RP0
00467 170B bcf INTCON,PEIE ;enable peripheral interrupts
00468 114A bcf flags,F_ERROR
00469 1283 bcf STATUS,RP0
00470 3007 movlw 0x07
00471 009F movwf CMCON ;disable comparators
00472 2140 call delay20 ;20msec delay
00473 089F movf CMCON,F ;clear comparator mismatch condition
00474 130C bcf PIR1,CMIF ;clear comparator interrupt flag
00475 1283 bcf STATUS,RP0 ;select bank 0
00476 0008 return

00477 ;***********************************************************************
00478 ;*---------------------------------------------------------------------*
00479 ;*- Turn Comparators and Vref Off -*
00480 ;*---------------------------------------------------------------------*
00481 ;***********************************************************************
00482 AnalogOff
00483 1283 bcf STATUS,RP0
00484 1283 bcf INTCON,PEIE
00485 3080 movlw 0x80 ;reset PORTB value
00486 0086 movwf PORTB
00487 1683 bcf STATUS,RP0 ;select bank 1
00488 130C bcf PIE1,CMIE ;disable comparator interrupts
00489 0185 clrf TRISA ;set PORTA pins to outputs, discharge RC ckt
00490 3060 movlw 0x60 ;set PORTB 7,4-0 as outputs, 6,5 as inputs
00491 0086 movwf TRISB
00492 139F bcf VRCON,VREN ;disable Vref
00493 1283 bcf STATUS,RP0 ;select bank 0
00494 3007 movlw 0x07
00495 009F movwf CMCON ;disable comparators
00496 2140 call delay20 ;20msec delay
00497 089F movf CMCON,F ;clear comparator mismatch condition
00498 130C bcf PIR1,CMIF ;clear comparator interrupt flag
00499 1283 bcf STATUS,RP0
00500 110B clrf INTCON,T0IF ;clear Timer0 interrupt flag
00501 213D call vlong ;long delay to allow capacitors to discharge
00502 213D clrf vlong
00503 213D call vlong
00504 0008 return

00505 ;***********************************************************************
00506 ;*---------------------------------------------------------------------*
00507 ;*- Swap Time to Accumulator a -*
00508 ;*---------------------------------------------------------------------*
00509 ;***********************************************************************
00510 ;***********************************************************************
00511 SwapTtoA
00512 1283 bcf STATUS,RP0
00513 0842 movwf TimeHI,W
00514 000D movwf ACCaHI

DS0611B-page 24 © 1997 Microchip Technology Inc.
0129 0841 00515   movf   TimeMID,W
012A 00D1 00516   movwf   ACCaMID
012B 0840 00517   movf   TimeLO,W
012C 00D2 00518   movwf   ACCaLO
012D 01C2 00519   clrf   TimeHI
012E 01C1 00520   clrf   TimeMID
012F 01C0 00521   clrf   TimeLO
0130 0008 00522   return

00523 ;_______________________________________________________________________
00524
00525 ;***********************************************************************
00526 ;*---------------------------------------------------------------------*
00527 ;*-     Transmit the Boot Message                                     -*
00528 ;*--------------------------------------------------------------------*
00529 ;***********************************************************************
0131         00530 BootMSG
0131 1283 00531   bcf     STATUS,RP0     ;select bank 0
0132 3002 00532   msg     movlw   HIGH Table   ;init the PCH for a table call
0133 008A 00533   movwf   PCLATH
0134 0828 00534   movf    offset,W        ;move table offset into W
0135 2200 00535   call    Table           ;get table value
0136 2095 00536   call    Transmit        ;transmit table value
0137 2146 00537   call    delay1          ;delay between bytes
0138 0BA8 00538   decfsz  offset,F        ;check for end of table
0139 2932 00539   goto    msg
013A 3010 00540   movlw   0x10            ;reset table offset
013B 00A8 00541   movwf   offset
013C 0008 00542   return

00543 ;_______________________________________________________________________
00544
00545 ;***********************************************************************
00546 ;*---------------------------------------------------------------------*
00547 ;*-     Delay Routines                                                -*
00548 ;*---------------------------------------------------------------------*
00549 ;***********************************************************************
013D 30FF 00550   vlong   movlw   0xff     ;very long delay, approx 200msec
013E 00A9 00551   movwf   msb
013F 2948 00552   goto    d1
0140 00553 00552   delay20     ;20 msec delay
0140 301A 00554   movlw   .26
0141 00A9 00555   movwf   msb
0142 2948 00556   goto    d1
0143 00557 00556   delay16     ;16 msec delay
0143 3015 00558   movlw   .21
0144 00A9 00559   movwf   msb
0145 2948 00560   goto    d1
0146 00561 00560   delay1     ;approx 750nsec delay
0147 3001 00562   movlw   .1
0147 00A9 00563   movwf   msb
0148 30FF 00564   movlw   0xff
0149 00AA 00565   movwf   lab
014A 0BAA 00566   decfsz  lab,F
014B 294A 00567   goto    d2
014C 0BAA 00568   decfsz  msb,F
014D 2948 00569   goto    d1
014E 0008 00570   return

00571 ;_______________________________________________________________________
00572
00573
0200 00574   org    0x200
00575
00576 ;***********************************************************************
00577 ;*---------------------------------------------------------------------*
00578 ;*-     Table for Boot Message                                         -*
00579 ;*---------------------------------------------------------------------*
00580 ;***********************************************************************
00581 ;******************************************************************************
00582 0200 0782    00583 Table                           ;boot message "PICMETER Booted!"
00584 0201 3400    00585         addwf   PCL, F          ;add W to PCL
00585 0202 3421    00586         retlw   0
00586 0203 3464    00587         retlw   '!'          ;add W to PCL
00587 0204 3465    00588         retlw   'd'
00588 0205 3474    00589         retlw   'e'
00589 0206 346F    00590         retlw   't'
00590 0207 346F    00591         retlw   'o'
00591 0208 3442    00592         retlw   'B'
00592 0209 3420    00593         retlw   ' '          ;add in carry if necessary
00593 020A 3452    00594         retlw   'R'
00594 020B 3445    00595         retlw   'E'
00595 020C 3454    00596         retlw   'T'
00596 020D 3445    00597         retlw   'E'
00597 020E 344D    00598         retlw   'M'
00598 020F 3443    00599         retlw   'C'
00599 0210 3449    00600         retlw   'I'
00600 0211 3450    00601         retlw   'P'
00601 ;_______________________________________________________________________
00602 ;************************************************************************
00603 ;*---------------------------------------------------------------------*
00604 ;*-     24-bit Addition                                               -*
00605 ;*-                                                                   -*
00606 ;*-     Uses ACCa and ACCb                                            -*
00607 ;*-                                                                   -*
00608 ;*-     ACCa + ACCb -> ACCb                                           -*
00609 ;*---------------------------------------------------------------------*
00610 ;***********************************************************************
00611 ;************************************************************************
00612 ;*---------------------------------------------------------------------*
00613 ;*-     Subtraction ( 24 - 24 -> 24 )                                      -*
00614 ;*-                                                                        -*
00615 ;*-     Uses ACCa, ACCb, ACCd                                              -*
00616 ;*-                                                                        -*
00617 ;*-     ACCa -> ACCd,                                                      -*
00618 ;*-     2's complement ACCa,                                               -*
00619 ;*-     call Add24 ( ACCa + ACCb -> ACCb ),                                -*
00620 ;*-     ACCd -> ACCa                                                       -*
00621 ;*---------------------------------------------------------------------*
00622 ;***********************************************************************
00623 ;*---------------------------------------------------------------------*
00624 ;*-     Transfer ACCa to ACCd                                           -*
00625 ;*-                                                                        -*
00626 ;*-     movf ACCAH,W ;Transfer ACCa to ACCd
00627 ;------------------------------------------------------------------------
00628 ;_______________________________________________________________________
00629 ;************************************************************************
00630 ;*---------------------------------------------------------------------*
00631 ;*---------------------------------------------------------------------*
00632 ;*---------------------------------------------------------------------*
00633 ;*---------------------------------------------------------------------*
00634 ;*---------------------------------------------------------------------*
00635 ;*---------------------------------------------------------------------*
00636 ;*---------------------------------------------------------------------*
00637 ;*---------------------------------------------------------------------*
00638 ;*---------------------------------------------------------------------*
00639 ;*---------------------------------------------------------------------*
00640 ;*---------------------------------------------------------------------*
00641 ;*---------------------------------------------------------------------*
00642 ;*---------------------------------------------------------------------*
00643 ;*---------------------------------------------------------------------*
00644 ;*---------------------------------------------------------------------*
00645 ;*---------------------------------------------------------------------*
00646 ;*---------------------------------------------------------------------*
0225 0852 00647  movf  ACCaLO,W
0226 00DB 00648  movwf  ACCdLO
0227 2275 00649  call  compA     ;2's complement ACCa
0228 2212 00650  call  Add24     ;Add ACCa to ACCb
0229 0859 00651  movf  ACCdHI,W  ;Transfer ACCd to ACCa
022A 00D0 00652  movwf  ACCaHI
022B 085A 00653  movf  ACCdMID,W
022C 00D1 00654  movwf  ACCaMID
022D 085B 00655  movf  ACCdLO,W
022E 00656  movwf  ACCaLO
022F 3400 00657  retlw  0

00658 ;_______________________________________________________________________
00659
00660 ;***********************************************************************
00661 ;*---------------------------------------------------------------------*
00662 ;*-     Multiply ( 24 X 24 -> 56 )                                 -*
00663 ;*---------------------------------------------------------------------*
00664 ;*---------------------------------------------------------------------*
00665 ;*---------------------------------------------------------------------*
00666 ;*---------------------------------------------------------------------*
00667 ;*---------------------------------------------------------------------*
00668 ;*---------------------------------------------------------------------*
00669 ;*---------------------------------------------------------------------*
00670 ;***********************************************************************
0230 00671 Mpy24
0230 223F 00672  call  Msetup
0231 0CD9 00673  mloop  rrf  ACCbHI, F       ;rotate d right
0232 0CDA 00674  rrf  ACCdMID, F
0233 0CDB 00675  rrf  ACCdLO, F
0234 1803 00676  btfsC  STATUS,C       ;need to add?
0235 2212 00677  call  Add24
0236 0CD3 00678  rrf  ACCbHI, F
0237 0CD4 00679  rrf  ACCbMID, F
0238 0CD5 00680  rrf  ACCbLO, F
0239 0CD6 00681  rrf  ACCcHI, F
023A 0CD7 00682  rrf  ACCcMID, F
023B 0CD8 00683  rrf  ACCcLO, F
023C 0BDC 00684  decfsz  temp, F       ;loop until all bits checked
023D 2A31 00685  goto  mloop
023E 3400 00686  retlw  0

00687
023F 00688 Msetup
023F 3018 00689  movlw  0x18            ;for 24 bit shifts
0240 00DC 00690  movwf  temp
0241 0853 00691  movf  ACCbHI,W        ;move ACCb to ACCd
0242 00D9 00692  movwf  ACCdHI
0243 0854 00693  movf  ACCbMID,W
0244 00DA 00694  movwf  ACCdMID
0245 0855 00695  movf  ACCcLO,W
0246 00DB 00696  movwf  ACCcLO
0247 01D3 00697  clrf  ACCcHI
0248 01D4 00698  clrf  ACCcMID
0249 01D5 00699  clrf  ACCcLO
024A 3400 00700  retlw  0

00701 ;_______________________________________________________________________
00702
00703 ;***********************************************************************
00704 ;*---------------------------------------------------------------------*
00705 ;*---------------------------------------------------------------------*
00706 ;*---------------------------------------------------------------------*
00707 ;*---------------------------------------------------------------------*
00708 ;*---------------------------------------------------------------------*
00709 ;*---------------------------------------------------------------------*
00710 ;*---------------------------------------------------------------------*
00711 ;*---------------------------------------------------------------------*
00712 ;*---------------------------------------------------------------------*

© 1997 Microchip Technology Inc.  DS00611B-page 27
024B 00715 Div24
024B 2272 00716 call Dsetup
024C 1003 00717
024D 0DD8 00718 dloop bcf STATUS,C ;Rotate dividend left 1 bit position
024E 0DD7 00719 rlf ACCcLO, F
024F 0DD6 00720 rlf ACCcMID, F
0250 0DD5 00721 rlf ACCcHI, F
0251 0DD4 00722 rlf ACCbLO, F
0252 0DD3 00723 rlf ACCbMID, F
0253 1803 00724 btfsc STATUS,C ;invert carry and exclusive or with the msb of the divisor then move this bit
0254 2A58 00725 goto clear ;into the lsb of the dividend
0255 1FD0 00726 btfss ACCaHI,msb_bit ;check the lsb of the dividend
0256 0AD8 00727 incf ACCcLO, F
0257 2A5A 00728 goto cont
0258 1BD0 00729 clear btfsc ACCaHI,msb_bit
0259 0AD8 00730 incf ACCcLO, F
025A 1858 00731 cont btfsc ACCcLO,lsb_bit ;check the lab of the dividend
025B 2A5E 00732 goto minus ;if = 0, then add divisor to upper 24 bits
025C 2212 00733 call Add24 ;of dividend
025D 2A5F 00734 goto check ;if = 1, then subtract divisor from upper 24 bits of dividend
025E 00735 ;24 bits of dividend
025F 0BDC 00736 check decfsz temp,f ;do 24 times
0260 2A4C 00737 goto dloop
0261 1003 00738
0262 0DD8 00739 bcf STATUS,C ;shift lower 24 bits of dividend 1 bit
0263 0DD7 00740 rlf ACCcMID, F ;position left
0264 0DD6 00741 rlf ACCcHI, F
0265 1BD3 00742 btfsc ACCbHI,msb_bit ;exclusive or the inverse of the msb of the dividend
0266 2A6A 00743 goto w1 ;with the msb of the divisor
0267 1FD0 00744 btfss ACCaHI,msb_bit ;store in the lsb of the dividend
0268 0AD8 00745 incf ACCcLO, F
0269 2A6C 00746 goto wzd
026A 1BD0 00747 btfsc ACCaHI,msb_bit
026B 0AD8 00748 incf ACCcLO, F
026C 1FD3 00749 btfss ACCbHI,msb_bit ;if the msb of the remainder is set and the msb of the divisor is not
026D 2A71 00750 goto wend
026E 1BD0 00751 btfsc ACCaHI,msb_bit ;the msb of the divisor is not
026F 2A71 00752 goto wend
0270 2212 00753 call Add24 ;add the divisor to the remainder to correct for zero partial remainder
0271 3400 00754 wend retlw 0 ;quotient in 24 lsb’s of dividend
0272 00763 ;remainder in 24 msb’s of dividend
0273 00764
0274 3400 00765 Dsetup
0275 3018 00766 movlw 0x18 ;loop 24 times
0276 00767 movwf temp
0277 00768
0278 3400 00769 retlw 0
0279 00770 ;
027A 00771
027B 00772 ;**********************************************************************
027C 00773 ;*---------------------------------------------------------------------*
027D 00774 ;* 2’s Complement -*
027E 00775 ;*-*
027F 00776 ;* Uses ACCa -*
0280 00777 ;-*
0281 00778 ;* Performs 2’s complement conversion on ACCa -*

DS00611B-page 28 © 1997 Microchip Technology Inc.
00779 ;-----------------------------------------------
00780 ;**********************************************************************************************
0275 compA
0275 09D2    comf ACCaLO, F ;invert all bits in accumulator a
0276 09D1    comf ACCaMID, F
0277 09D0    comf ACCaHI, F
0278 0AD2    incf ACCaLO, F ;add one to accumulator a
0279 1903    btfsc STATUS,Z
027A 0AD1    incf ACCaMID, F
027B 1903    btfsc STATUS,Z
027C 0AD0    incf ACCaHI, F
027D 3400    retlw 0

PICMETER Firmware for PIC16C622
MEMORY USAGE MAP ('X' = Used, '-' = Unused)

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 XXXXXXXXXXXXXXXX
2000 : -------X-------- ---------------- ---------------- ----------------

All other memory blocks unused.

Program Memory Words Used: 447
Program Memory Words Free: 1601

Errors : 0
Warnings : 0 reported, 0 suppressed
Messages : 0 reported, 15 suppressed
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

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
    End If
End Sub
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
Note the following details of the code protection feature on PICmicro® MCUs.

- The PICmicro family meets the specifications contained in the Microchip Data Sheet.
- Microchip believes that its family of PICmicro microcontrollers is one of the most secure products of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the PICmicro microcontroller in a manner outside the operating specifications contained in the data sheet. The person doing so may be engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as "unbreakable".
- Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our product.

If you have any further questions about this matter, please contact the local sales office nearest to you.

Information contained in this publication regarding device applications and the like is intended through suggestion only and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. No representation or warranty is given and no liability is assumed by Microchip Technology Incorporated with respect to the accuracy or use of such information, or infringement of patents or other intellectual property rights arising from such use or otherwise. Use of Microchip's products as critical components in life support systems is not authorized except with express written approval by Microchip. No licenses are conveyed, implicitly or otherwise, under any intellectual property rights.

Microchip received QS-9000 quality system certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona in July 1999. The Company's quality system processes and procedures are QS-9000 compliant for its PICmicro® 8-bit MCUs, KeElLoq® code hopping devices, Serial EEPROMs and microperipheral products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001 certified.

© 2002, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

Microchip technology Incorporated, Printed on recycled paper.
AMERICAS
Corporate Office
2335 West Chandler Blvd.
Chandler, AZ  85224-6199
Tel:  480-792-7200  Fax:  480-792-7277
Technical Support: 480-792-7627
Web Address: http://www.microchip.com

Rocky Mountain
2335 West Chandler Blvd.
Chandler, AZ  85224-6199
Tel:  480-792-7966  Fax:  480-792-7456

Atlanta
500 Sugar Mill Road, Suite 2008
Atlanta, GA  30350
Tel:  770-640-0334  Fax:  770-640-0307

Chicago
333 Pierce Road, Suite 180
Itasca, IL  60143
Tel:  630-285-0071 Fax:  630-285-0075

Dallas
4570 Westgrove Drive, Suite 160
Addison, TX 75001
Tel:  972-818-7924  Fax:  972-818-2924

Detroit
Tri-Atria Office Building
32255 Northwestern Highway, Suite 190
Farmington Hills, MI  48334
Tel:  248-538-2250 Fax:  248-538-2260

Kokomo
2767 S. Albright Road
Kokomo, Indiana  46902
Tel:  765-864-8360 Fax:  765-864-8387

Los Angeles
18201 Von Karman, Suite 1090
Irvine, CA  92612
Tel:  949-263-1888 Fax:  949-263-1338

New York
150 Motor Parkway, Suite 202
Hauppauge, NY  11788
Tel:  631-273-0000 Fax:  631-273-0071

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

Toronto
6285 Northpark Drive, Suite 108
Mississauga, Ontario L4V 1X5, Canada
Tel:  905-673-0699 Fax:  905-673-6509

ASIA/PACIFIC
Australia
Microchip Technology Australia Pty Ltd
Suite 22, 41 Rawson Street
Epping 2121, NSW
Australia
Tel:  61-2-9868-6733 Fax:  61-2-9868-6755

China - Beijing
Microchip Technology Consulting (Shanghai) Co., Ltd., Beijing Liaison Office
Unit 915
Beijing, 100027, No. China
Tel:  86-10-85282100 Fax:  86-10-85282104

China - Chengdu
Microchip Technology Consulting (Shanghai) Co., Ltd., Chengdu Liaison Office
Rm. 2401, 24th Floor,
Ming Xing Financial Tower
No. 88 TIDU Street
Chengdu 610016, China
Tel:  86-28-6766200 Fax:  86-28-6766599

China - Fuzhou
Microchip Technology Consulting (Shanghai) Co., Ltd., Fuzhou Liaison Office
Unit 7F, World Trade Plaza
No. 71 Wusi Road
Fuzhou 350001, China
Tel:  86-591-7503506 Fax:  86-591-7503521

China - Shanghai
Microchip Technology Consulting (Shanghai) Co., Ltd., Shanghai Liaison Office
Room 701, Bldg D
Far East International Plaza
No. 317 Xian Xing Road
Shanghai, 200005
Tel:  86-21-6275-5700 Fax:  86-21-6275-5060

China - Shenzhen
Microchip Technology Consulting (Shanghai) Co., Ltd., Shenzhen Liaison Office
Rm. 1301, 13th Floor,
Shenzhen Center
Shenzhen 518030, China
Tel:  86-755-2389901 Fax:  86-755-2380008

Hong Kong
Microchip Technology Hongkong Ltd.
Unit 901-6, Tower 2, Metroplaza
223 Hing Fong Road
Kwai Fong, N.T., Hong Kong
Tel:  852-2401-1200 Fax:  852-2401-3431

India
Microchip Technology Inc.
India Liaison Office
Divyasree Chambers
1 Floor, Wing A (A3/A4)
No. 11, O’Shaugnesssey Road
Bangalore, 560 025, India
Tel:  91-80-2290061 Fax:  91-80-2290062

Japan
Microchip Technology Japan K.K.
Benex S-1 6F
3-18-20, Shinyokohama
Kohoku-Ku, Yokohama-shi
Kanagawa, 222-0033, Japan
Tel:  81-45-471-6186 Fax:  81-45-471-6122

Korea
Microchip Technology Korea
168-1, Youngbo Bldg. 3 Floor
Samsung-Dong, Kangnam-Ku
Seoul, Korea 135-882
Tel:  82-2-554-7200 Fax:  82-2-558-5934

Singapore
Microchip Technology Singapore Pte Ltd.
200 Middle Road
#07-02 Prime Centre
Singapore, 189980
Tel:  65-334-8870 Fax:  65-334-8850

Taiwan
Microchip Technology Taiwan
11F-3, No. 207
Tung Hua North Road
Taipei, 105, Taiwan
Tel:  886-2-2717-7175 Fax:  886-2-2545-0139

EUROPE

Denmark
Microchip Technology Nordic ApS
Regus Business Centre
Laurup høj 1-3
Ballerpur DK-2750 Denmark
Tel:  45 4420 9895 Fax:  45 4420 9910

France
Microchip Technology SARL
Parc d’Activite du Moulin de Massy
43 Rue du Saule Trapu
Batiment A - ler Etage
91900 Massy, France
Tel:  33-1-69-53-63-20 Fax:  33-1-69-30-90-79

Germany
Microchip Technology GmbH
Gustav-Heinemann Ring 125
D-81739 Munich, Germany
Tel:  49-89-6100-9  Fax:  49-89-6100-9

Italy
Microchip Technology SRL
Centro Direzionale Colleoni
Palazzo Taurus 1 V. Le Colleoni 1
20041 Agrate Brianza
Milan, Italy
Tel:  39-039-65791-1  Fax:  39-039-6899883

United Kingdom
Arizona Microchip Technology Ltd.
505 Eskdale Road
Winnersh Triangle
Wokingham
Berkshire, England RG41 5TU
Tel:  44 118 921 5869 Fax:  44-118 921-5820

01/18/02

© 2002 Microchip Technology Inc.