

X-10[®] Home Automation Using the PIC16F877A

*Author: Jon Burroughs
Microchip Technology Inc.*

INTRODUCTION

X-10 is a communication protocol designed for sending signals over 120 VAC wiring. X-10 uses 120 kHz bursts timed with the power line zero-crossings to represent digital information. Plug-in modules available from various vendors enable users to create home automation systems by using the AC wiring already installed within a home. Readers who would like an overview of the X-10 signal format may refer to Appendix A.

PICmicro[®] microcontrollers can easily be used in conjunction with X-10 technology to create home automation applications. The specific PICmicro microcontroller (MCU) used should be selected based on RAM, ROM, operating frequency, peripheral, and cost requirements of the particular application. The PIC16F877A was selected for this application because of its versatility as a general purpose microcontroller, its FLASH program memory (for ease of development), data EEPROM, and ample I/O.

This application note discusses the implementation of X-10 on a PICmicro MCU to create a home controller that can both send and receive X-10 signals. The reader may implement the home controller as is, or adapt the circuits and firmware to other applications. A library of X-10 functions is provided to facilitate development of other X-10 applications using PICmicro MCUs (see Appendix E).

Operating instructions for the home controller are included in Appendix B.

HARDWARE OVERVIEW

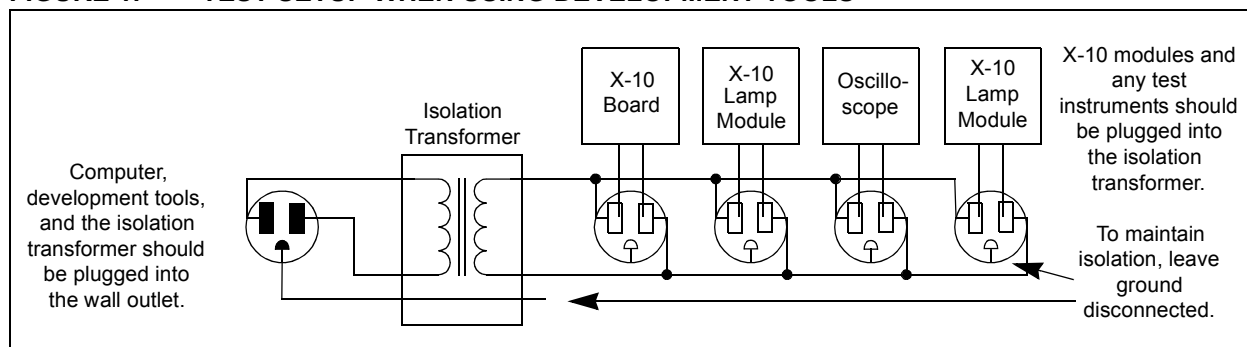
The home controller application described in this application note allows the user to program on and off times for up to sixteen devices, using a 2 x 16 liquid crystal display and five push buttons. A built-in light sensor can be used to turn on lights at dusk, and turn them off at dawn.

The home controller is designed to facilitate experimentation with home automation using the PIC16F877A. In addition to the PIC16F877A, the board will accept any other PICmicro MCU that shares the same pinout, such as the PIC18F452. Therefore, experimenters may expand on the application using the higher performance of the PIC18 family of parts without changing the hardware.

With care, engineers and home control enthusiasts can experiment with home automation using the MPLAB[®] ICD and MPLAB[®] ICD 2 development tools or in-circuit emulator. However, proper circuit isolation precautions must be taken to avoid damage to your computer or development tools. See Figure 1 and the warning note!

WARNING: Vss or ground on the application circuit is tied to neutral of the 120 VAC. To safely connect your development tools or computer to the home controller, you must power it through an isolation transformer and leave wall ground (the green wire in most cases) disconnected. Any test instruments (such as an oscilloscope) that you hook up to the application circuit, should be powered through the isolation transformer as well, with wall ground disconnected. In addition, the entire circuit should be enclosed within a suitable case to prevent unintentional contact with the mains voltage!

FIGURE 1: TEST SETUP WHEN USING DEVELOPMENT TOOLS



HARDWARE DESCRIPTION

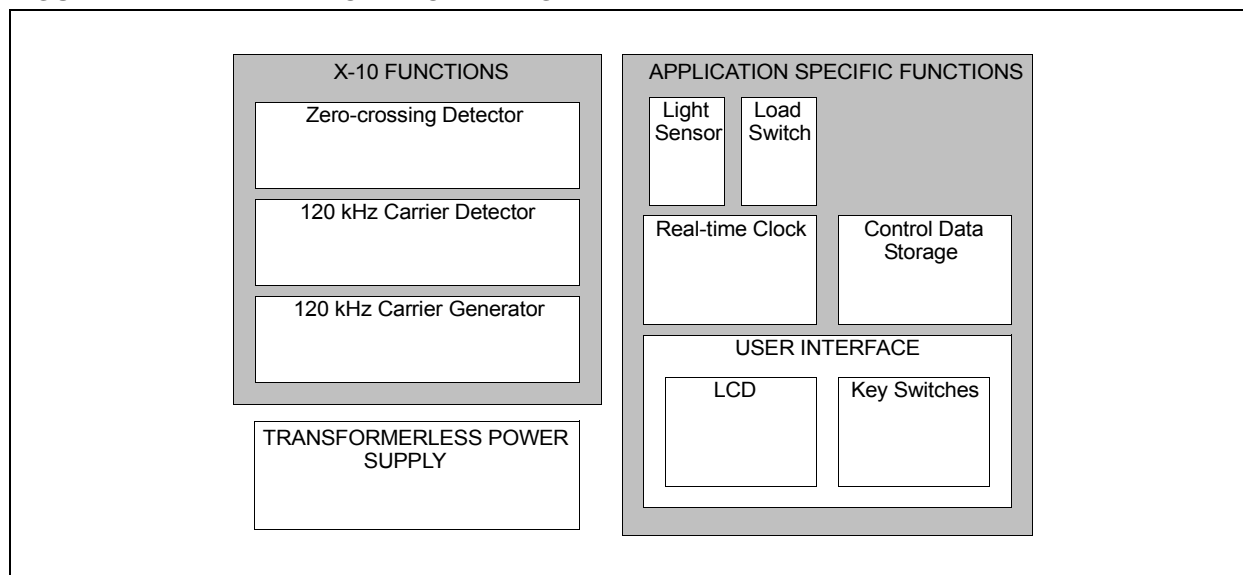
An overview of the home controller application hardware is shown in Figure 2.

The hardware functionality of X-10 circuitry can be divided into four functional blocks:

- Zero-crossing detector
- 120 kHz carrier detector
- 120 kHz signal generator
- Transformerless power supply

There are several application functions that are not directly associated with the X-10 interface. User interface functions are accomplished with an LCD display and five push buttons. A real-time clock is created using Timer1 and an external 32 kHz oscillator. User modified control data, such as unit on and off times, are stored in the PICmicro MCU's built-in EEPROM. A light sensor and load switch are also used in this application.

FIGURE 2: APPLICATION BLOCK DIAGRAM



A summary of resource use can be seen in Table 1.
Details of the functional sections are discussed below.

TABLE 1: SUMMARY OF MICROCONTROLLER RESOURCE USE

Resource	Function	Description
External interrupt on RB0	Zero-crossing Detect	Generates one interrupt every zero-crossing.
CCP1/Timer2 in PWM mode	120 kHz Modulation	TRISC is used to enable/disable 120 kHz output. Main oscillator is 7.680 MHz.
Timer2 interrupt through postscaler	Triac Dimmer Timing	Generates dimmer timing increments for controlling Triac.
Timer1 interrupt	Real-time Clock	Used as time keeping clock and key scan clock. One interrupt/25 ms, 40 interrupts/1 sec.
Timer0 interrupt	120 kHz Envelope Timing	Times duration of 1 ms bursts and onset of second and third phase bursts.
ADC	Light Sensor	Used to detect dawn and dusk.
PORTB<1:5>	Key Press Inputs	Five push buttons are used for menu navigation.
PORTB<6:7>	Reserved for ICD	Isolation precautions required. See warning note!
PORTD<0:7>	LCD Data pins	8 data lines for LCD.
PORTE<0:2>	LCD Control pins	3 control lines for LCD.
DATA EEPROM	Non-volatile Control Data Storage	Stores on and off times and other user programmable information.

Zero-Crossing Detector

In X-10, information is timed with the zero-crossings of the AC power. A zero-crossing detector is easily created by using the external interrupt on the RB0 pin and just one external component, a resistor, to limit the current into the PICmicro MCU (see Figure 3).

In the United States, $V_{rms} = 117\text{ VAC}$, and the peak line voltage is 165V. If we select a resistor of $5\text{ M}\Omega$, $I_{peak} = 165\text{V}/5\text{ M}\Omega = 33\text{ }\mu\text{A}$, which is well within the current capacity of a PICmicro MCU I/O pin.

Input protection diodes (designed into the PICmicro MCU I/O pins) clamp any voltage higher than V_{DD} or lower than V_{SS} . Therefore, when the AC voltage is in the negative half of its cycle, the RB0 pin will be clamped to $V_{SS} - 0.6\text{V}$. This will be interpreted as a logic zero. When the AC voltage rises above the input threshold, the logical value will become a '1'.

In this application, RB0 is configured for external interrupts, and the input buffer is a Schmitt trigger. This makes the input threshold $0.8\text{ }V_{DD} = 4\text{V}$ on a rising edge and $0.2\text{ }V_{DD} = 1\text{V}$ on a falling edge.

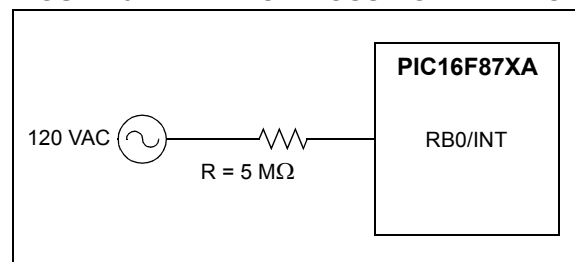
Upon each interrupt, the Interrupt Edge Select bit within the OPTION_REG register is toggled, so that an interrupt occurs on every zero-crossing. Using the following equation, it is possible to calculate when the pin state will change relative to the zero-crossing:

$$V = V_{pk} \cdot \sin(2\pi \cdot f \cdot t), \text{ where } V_{pk} = 165\text{V and } f = 60\text{ Hz}$$

On a rising edge, RB0 will go high about $64\text{ }\mu\text{s}$ after the zero-crossing, and on a falling edge, it will go low about $16\text{ }\mu\text{s}$ before the zero-crossing.

More information on interfacing PICmicro MCUs to AC power lines can be found in the application note AN521, "Interfacing to AC Power Lines", which is available for download from the Microchip web site.

FIGURE 3: ZERO-CROSSING DETECTOR



120 kHz Carrier Detector

To receive X-10 signals, it is necessary to detect the presence of the 120 kHz signal on the AC power line. This is accomplished with a decoupling capacitor, a high-pass filter, a tuned amplifier, and an envelope detector. The components of the carrier detector are illustrated in Figure 4.

Because the impedance of a capacitor is: $Z_c = 1/(2\pi fC)$, a 0.1 μF capacitor presents a low impedance (13 Ω) to the 120 kHz carrier frequency, but a high impedance (26.5 k Ω) to the 60 Hz power line frequency. This high-pass filter allows the 120 kHz signal to be safely coupled to the 60 Hz power line, and it doubles as the coupling stage of the 120 kHz carrier generator described in the next section.

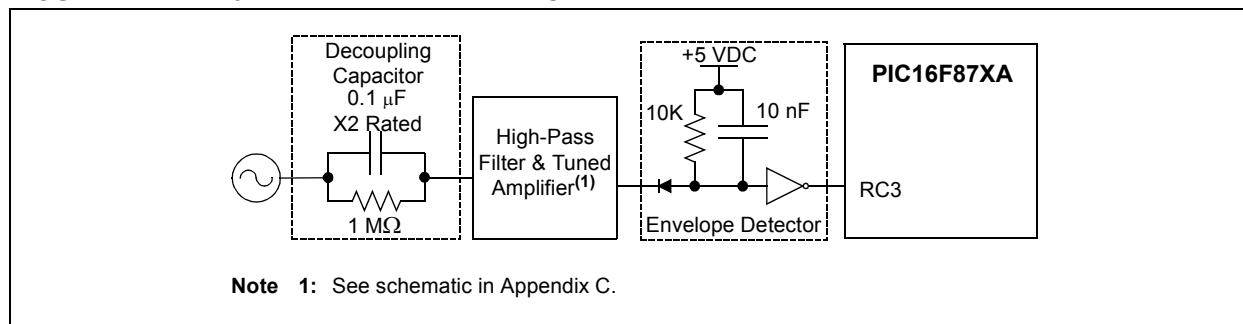
Since the 120 kHz carrier frequency is much higher than the 60 Hz power line frequency, it is straightforward to design an RC filter that will pass the 120 kHz signal and completely attenuate the 60 Hz. A high-pass filter forms the first stage of the High-Pass Filter and Tuned Amplifier Block, shown on sheet 5 of the schematics in Appendix C.

For a simple high-pass filter, the -3 db breakpoint is: $f_{3\text{ db}} = 1/(2\pi R C)$. For $C = 150\text{ pF}$ and $R = 33\text{ k}\Omega$, $f_{3\text{ db}} = 1/(2\pi * 150\text{ pF} * 33\text{ k}\Omega) = 32\text{ kHz}$.

This $f_{3\text{ db}}$ point assures that the 60 Hz signal is completely attenuated, while the 120 kHz signal is passed through to the amplifier stages. Next, the 120 kHz signal is amplified using a series of inverters configured as high gain amplifiers. The first two stages are tuned amplifiers with peak response at 120 kHz. The next two stages provide additional amplification. The amplified 120 kHz signal is passed through an envelope detector, formed with a diode, capacitor, and resistor. The envelope detector output is buffered through an inverter and presented to an input pin (RC3) of the PIC16F877A.

Upon each zero-crossing interrupt, RC3 is simply checked within the 1 ms transmission envelope to see whether or not the carrier is present. The presence or absence of the carrier represents the stream of '1's and '0's that form the X-10 messages described in Appendix A.

FIGURE 4: 120 kHz CARRIER DETECTOR



120 kHz Carrier Generator

X-10 uses 120 kHz modulation to transmit information over 60 Hz power lines. It is possible to generate the 120 kHz carrier with an external oscillator circuit. A single I/O pin would be used to enable or disable the oscillator circuit output. However, an external oscillator circuit can be avoided by using one of the PICmicro MCU's CCP modules.

The CCP1 module is used in PWM mode to produce a 120 kHz square-wave with a duty cycle of 50%. Because X-10 specifies the carrier frequency at 120 kHz (+/- 2 kHz), the system oscillator is chosen to be 7.680 MHz, in order for the CCP to generate precisely 120 kHz. Calculations for setting the PWM period and duty cycle are shown in the code listing comments for the function `InitPWM`.

After initialization, CCP1 is continuously enabled, and the TRISC bit for the pin is used to gate the PWM output. When the TRISC bit is set, the pin is an input and the 120 kHz signal is not presented to the pin. When the TRISC bit is clear, the pin becomes an output and the 120 kHz signal is coupled to the AC power line through a transistor amplifier and capacitor, as depicted in Figure 5.

Since the impedance of a capacitor is $Z_c = 1/(2\pi fC)$, a 0.1 μF capacitor presents a low impedance to the 120 kHz carrier frequency, but a high impedance to the 60 Hz power line frequency. This high-pass filter allows the 120 kHz signal to be safely coupled to the 60 Hz power line, and it doubles as the first stage of the 120 kHz carrier detector, described in the previous section.

To be compatible with other X-10 receivers, the maximum delay from the zero-crossing to the beginning of the X-10 envelope should be about 300 μs . Since the zero-crossing detector has a maximum delay of approximately 64 μs , the firmware must take less than 236 μs after detection of the zero-crossing to begin transmission of the 120 kHz envelope.

Transformerless Power Supply

The PIC16F877A and other board circuits require a 5V supply. In this application, the X-10 controller must also transmit and receive its data over the AC line. Since X-10 components are intended to be plugged into a wall outlet and have a small form factor, a transformerless power supply is used. Two characteristics of transformerless supplies that should be kept in mind are limited current capacity, and lack of isolation from the AC mains (see the warning note)!

WARNING: This circuit is not isolated from 120 VAC. Act with caution when constructing or using such a circuit, and ensure that it is contained within a suitable insulated enclosure. Follow isolation precautions to avoid personal injury or damage to test equipment and development tools.

Figure 6 illustrates the transformerless power supply used in this application. To protect the circuit from spikes on the AC power line, a 130V VDR (voltage dependent resistor) is connected between Line and Neutral. A Positive Temperature Coefficient (PTC) device acts as a resettable fuse, which limits current between Ground and Neutral. The 47 Ω resistor limits current into the circuit, and the 1 M Ω resistor provides a discharge path for the voltage left on the capacitor when the circuit is unplugged from the wall. Two diodes rectify the voltage across the 1000 μF capacitor and 5.1V Zener diode to produce a 5V supply.

The reader may wish to refer to the technical brief TB008, "Transformerless Power Supply", available for download from the Microchip web site, for additional information on transformerless power supply design.

FIGURE 5: 120 kHz CARRIER GENERATOR

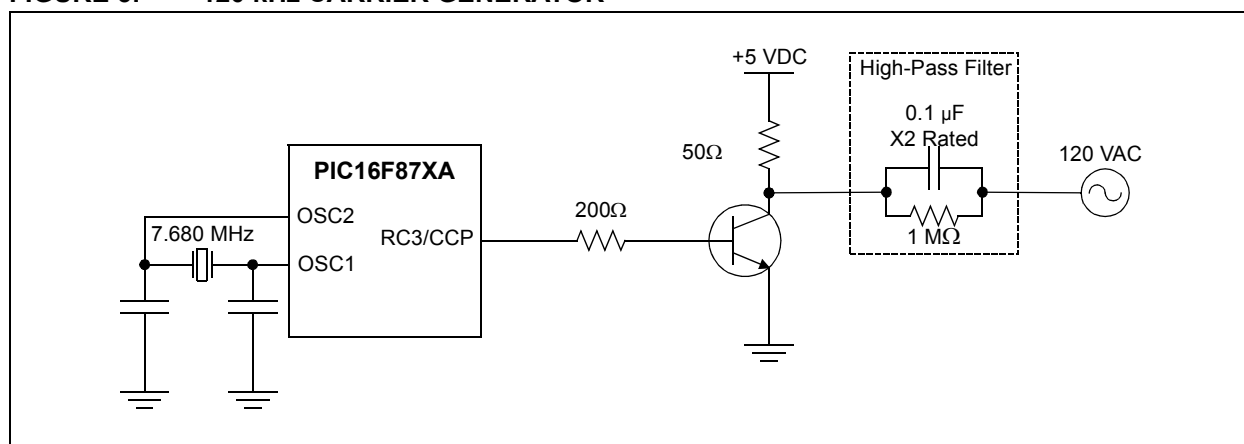
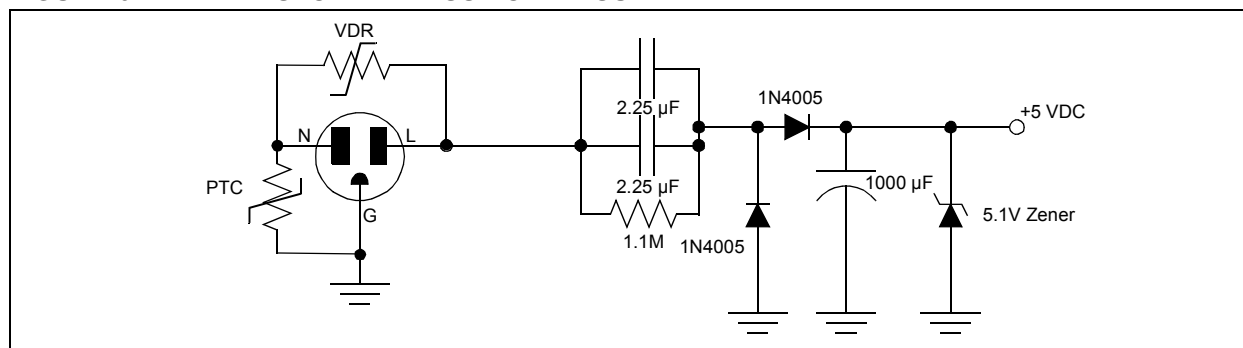


FIGURE 6: TRANSFORMERLESS POWER SUPPLY



Load Switch

A load switch is included on the home controller so that it may act as a lamp module, with its own house and unit address. A Triac was selected as the load switch, because its medium power switching capacity and rapid switching capability make it well-suited for lamp control and dimming.

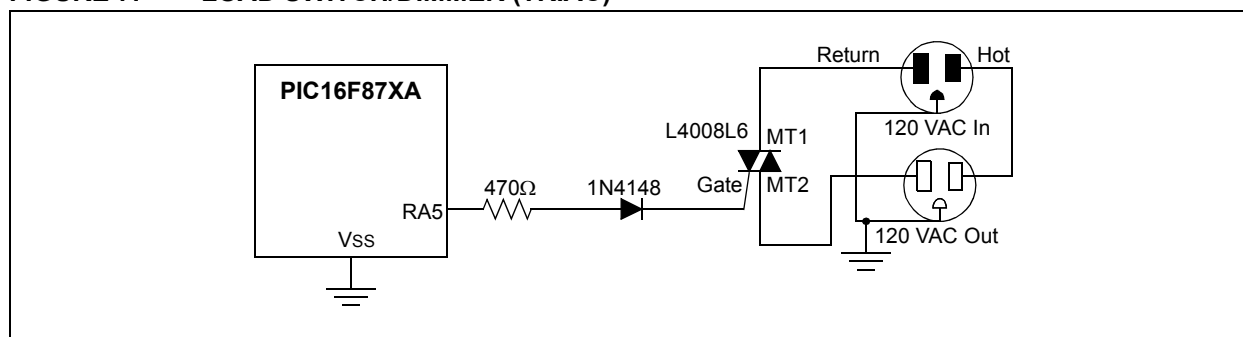
A Triac is an inexpensive, three-terminal device that basically acts as a high speed, bi-directional AC switch. Two terminals, MT1 and MT2, are wired in series with the load. A small trigger current between the gate and MT1 allow conduction to occur between MT1 and MT2. Current continues to flow after the gate current is removed, as long as the load current exceeds the latching value. Because of this, the Triac will automatically switch off near each zero-crossing as the AC voltage falls below the latching voltage.

A Teccor® L4008L6 Triac was selected because it has a sensitive gate that can be directly controlled from the logic level output of the PICmicro MCU I/O pin. The sensitive gate Triac can control AC current in both directions through the device, even though the PICmicro MCU can provide only positive voltages to the gate.

A variable dimmer is created by including a delay between the time of each zero-crossing and the time that the trigger current is provided to the Triac from the PICmicro MCU.

The design and control of a lamp dimmer using a PICmicro MCU is discussed in detail in PICREF-4 Reference Design, "PICDIM Lamp Dimmer for the PIC12C508".

FIGURE 7: LOAD SWITCH/DIMMER (TRIAC)



LCD Module

The 2-line x 16-character display uses the HD44780U Display Controller. Eight data lines and three control lines are used to interface to the PICmicro MCU. If fewer I/O pins are available, the LCD can be operated in Nibble mode using only four data lines, with some additional software overhead. A basic LCD library is included in this application, which provides the necessary functions for controlling this type of LCD.

Real-Time Clock

A real-time clock is implemented using Timer1. The real-time clock keeps track of the present time using a routine called `UpdateClock`. It also determines the rate that the buttons are read by a routine called `ScanKeys`.

Timer1 is set to cause an interrupt each time it overflows. By adding a specific offset to Timer1 each time it overflows, the time before the next overflow can be precisely controlled. The button reading routine, `ScanKeys`, is called each time a Timer1 interrupt occurs. Since `ScanKeys` performs debouncing of the button presses, a suitable rate to check the buttons is once every 25 ms.

With a 32 kHz crystal, the counter increments once every 31.25 μ s when the prescaler is set to 1:1. In order for Timer1 to generate an interrupt once every 25 ms, TMR1H:TMR1L are pre-loaded with 0xFCE0h.

The Timer1 interrupt interval, or tick, can be seen in the following equation:

$$(FFFFh - FCE0h) * 1/32 \text{ kHz} = .025 \text{ s} = 1 \text{ tick}$$

Each time `ScanKeys` is called (every 25 ms), it calls `UpdateClock`. `UpdateClock` keeps track of the time unit variables: ticks, seconds, minutes, and hours. Since every 25 ms equals one tick, seconds are incremented every 40 ticks. Minutes and hours are incremented in a similar fashion.

Push Buttons

Five push buttons, connected to RB1-RB5, are used for user interaction with the application. Each normally open push button will pull a port pin low when it is pressed.

Light Sensor

To detect the ambient light level, a CdS photoresistor is used in conjunction with an 820 Ω resistor to create a voltage divider. The voltage on the divider varies with the intensity of ambient light and is connected to an analog channel (AN0) of the microcontroller.

In-Circuit Debugger

RB6 and RB7 have been reserved for In-Circuit Serial Programming™ (ICSP™) and the in-circuit debugger (ICD). However, do not connect the ICD or any other

development tool, without taking first isolating the entire application from wall power (see the previous warning notes)!

Control Data Storage

Certain control data that is programmable by the user must be stored in non-volatile memory. The PICmicro MCU's built-in EEPROM is well-suited to this task.

To use EEPROM memory space most efficiently (by avoiding wasted bits), on/off times and light sensor control flags are stored using the format shown in Figure 8. Figure 9 shows the location of on/off times and other information within the data EEPROM. Using this data organization, only 48 bytes of EEPROM are required to store the on/off times and light sensor control flags for 16 units.

FIGURE 8: ON/OFF TIME STORAGE

EEHours	4 bits		4 bits	
	On Hour		Off Hour	
EEOnMinutes	1	1	6 bits	
	A	B	OnMin	
EEOffMinutes	1	1	6 bits	
	C	D	Off Min	

A = AM/PM bit for On Hour
B = Control bit for On at Dusk
C = AM/PM bit for Off Hour
D = Control bit for Off at Dawn

FIGURE 9: EEPROM DATA

Address	Unit	Data	
0x001	System	House Address	
0x002	System	Unit Address	
0x010	Unit 1	OnHour	OffHour
0x011	Unit 2	OnHour	OffHour
0x012	Unit 3	OnHour	OffHour
•	•	•	•
•	•	•	•
•	•	•	•
0x020	Unit 1	A B	OnMin
0x021	Unit 2	A B	OnMin
0x022	Unit 3	A B	OnMin
•	•	•	•
•	•	•	•
•	•	•	•
0x030	Unit 1	A B	OffMin
0x031	Unit 2	A B	OffMin
0x032	Unit 3	A B	OffMin
•	•	•	•
•	•	•	•
•	•	•	•

Each time that minutes are incremented within the `UpdateClock` routine, a flag is set that enables a routine called `CheckOnOffTimes` to be called from the main loop. `CheckOnOffTimes` compares the present time with the unit on and off times stored in EEPROM memory. If there is a match, then a flag is set to either turn the unit on or off, by sending it the appropriate X-10 command when the routine `ControlX10Units` is called.

APPLICATION FIRMWARE OVERVIEW

The firmware is divided into several different files to facilitate adaptation of the code to other applications. Following is a summary of the files associated with this application note:

- `x10lib.asm` Defines X-10 functions.
- `x10lib.inc` Defines X-10 constants and macros.
- `x10hc.asm` Main application code for the home controller.
- `x10demo.asm` Example code that shows how to use the X-10 library macros.
- `lcd.asm` Defines the routines necessary for driving the LCD.
- `p16f877A.lkr` Standard linker file for PIC16F877A parts.
- `p16f877A.inc` Standard include file for PIC16F877A parts.

Detailed descriptions of operation can be found in the comments within the code listing. The X-10 library functions and macros are described in the next section.

X-10 LIBRARY

A simple library of commands was developed and used for the home controller. It can be used with little or no modification in a user's application. The library consists of two files: `x10lib.asm` and `x10lib.inc`.

To use the library, a user need only understand the function of the macros defined in `x10lib.inc`. The macros greatly simplify the use of the library by eliminating the need for the user to understand every X-10 function in `x10lib.asm`. Examples of how the macros are used are included in the file `x10demo.asm`.

The macros are explained below:

InitX10

This macro is used to initialize the peripherals that provide X-10 functionality. It must be called in the application program before any of the below macros will work. It is used as follows:

```
InitX10
```

SkipIfTxReady

Before sending an X-10 message, it is necessary to make sure that another message is not already being sent, which is signified by the `X10TxFlag` being set. This macro simply checks that flag and skips the next instruction if it is okay to begin a new transmission. Otherwise, there is a chance that a new transmission will interrupt an ongoing transmission.

It is used as follows:

```
SkipIfTxDone
GOTO    $-1    ;loop until ready to
                ;transmit next message
```

SendX10Address (House, Unit)

This macro is used to send an X-10 address for a particular unit. It requires two arguments, a house address and unit address. The definitions for all house and unit addresses are defined in `x10lib.inc`. To use this macro to send the address for unit 16 at house P, one simply types:

```
SendX10Address HouseP, Unit16
```

SendX10AddressVar

This macro is used to send an X-10 address, defined by variables rather than constants. To send an address contained in the user variables `MyHouse` and `MyUnit`, the following sequence would be applied:

```
MOVWF    MyHouse, W    ;contains a value
                        ;from 0-16

MOVWF    TxHouse

MOVWF    MyUnit, W      ;contains a value
                        ;from 0-16

MOVWF    TxUnit

SendX10AddressVar
```


SendX10Command (House, Function)

This macro is used to send an X-10 command. It requires two arguments, the house address and function code. The definitions for all house addresses and function codes are defined in `x10lib.inc`. To use this macro to send the command 'All Lights On' to all units at house A, one types:

```
SendX10Command HouseA, AllLightsOn
```

SendX10CommandVar

This macro is used to send an X-10 command, defined by a variable rather than a constant. To use this macro to send the command stored in the user variable `MyCommand` to all units at `MyHouse`, one types:

```
MOVF    MyHouse, W    ;contains a value
                        ;from 0-16

MOVWF   TxHouse

MOVF    MyCommand, W  ;any X-10
                        ;function
                        ;defined in
                        ;x10lib.inc

MOVWF   TxFunction

SendX10CommandVar
```

SkipIfRxDone

Before reading an X-10 message, it is necessary to make sure that a complete message has been received. This is signified by the `X10RxFlag` being set. This macro simply checks that flag and skips the next instruction if a new X-10 message has been received.

It is used as follows:

```
SkipIfRxDone

GOTO    $-1           ;loop until message
                        ;received
```

SkipIfAddressRcvd

It may be necessary to make sure that an address was received by using this macro, which checks to see if the `RxCommandFlag` is clear.

It is used as follows:

```
SkipIfAddressRcvd

GOTO    $-1           ;loop until address
                        ;received
```

SkipIfCommandRcvd

Or, it may be necessary to make sure that a command was received by using this macro, which checks to see if the `RxCommandFlag` is set.

It is used as follows:

```
SkipIfCommandRcvd

GOTO    $-1           ;loop until command
                        ;received
```

ReadX10Message

This macro is called to read a received X-10 message, which may be either an address or a command. If the message was an address, then the received house and unit codes will be stored in the variables `RxHouse` and `RxUnit`, respectively. If the message was a command, then the received house address and function code will be stored in the variables `RxHouse` and `RxFunction`. It is simply called as follows:

```
ReadX10Message
```

Please refer to the example code in `x10demo.asm` to see how each of these macros is used in a simple application.

Memory Usage

Memory usage for the X-10 portion of the application is summarized in Table 2.

TABLE 2: SUMMARY OF MEMORY USAGE FOR X-10 FUNCTIONALITY

Memory Type	Used	Available on PIC16F877A	Percent Used
FLASH Program Memory	437 words	8453 words	5%
Data Memory (RAM)	62 bytes	368 bytes	17%
EEPROM Data Memory	0 bytes	256 bytes	0%

Memory usage for the entire home controller application is summarized in Table 3.

TABLE 3: SUMMARY OF MEMORY USAGE FOR THE HOME CONTROLLER

Memory Type	Used	Available on PIC16F877A	Percent Used
FLASH Program Memory	3762 words	8453 words	44.5%
Data Memory (RAM)	168 bytes	368 bytes	45.6%
EEPROM Data Memory	51 bytes	256 bytes	20%

CONCLUSION

The PICmicro MCU is well-suited to X-10 applications. With its plethora of on-chip peripherals and a few external components, a PICmicro MCU can be used to implement an X-10 system that can transmit and receive messages over the AC power line wiring. The small code size of the X-10 library leaves ample space for the user to create application specific code. PICmicro MCUs, such as the PIC16F877A, have plenty of additional resources for creating more complex X-10 applications, while smaller PICmicro MCUs can be selected for economical use in simpler X-10 applications.

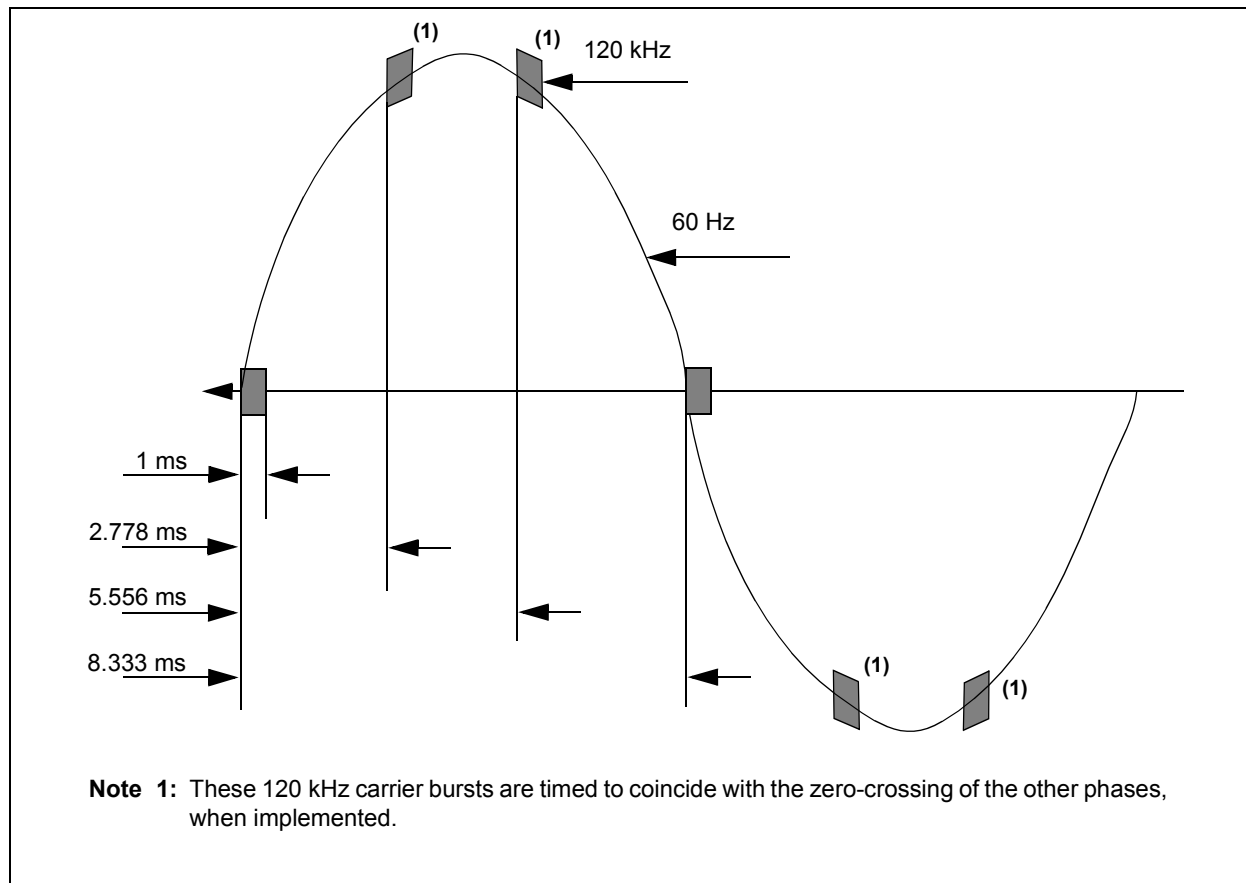
USEFUL WEB REFERENCES

- <http://www.abacuselectrics.com/x10.htm>
This web site describes how to build an appliance module that utilizes the PIC16C52 or PIC16F84. Parts of this project's receiver circuit, designed by Phil Plunkett, were adapted to the home controller application.
- <http://www.microchip.com>
The Microchip web site features data sheets, product information, and more. Helpful technical documentation available here include:
AN521 "Interfacing to AC Power Lines"
TB008 "Transformerless Power Supply"
PICREF-4 "PICDIM Lamp Dimmer for the PIC12C508"
- <http://www.x10.com/support>
The X10 Wireless Technology, Inc.TM web site features technical information and FAQs pertaining to the X-10 communication protocol.

APPENDIX A: HOW DOES THE X-10 PROTOCOL WORK?

X-10 transmissions are synchronized with the zero-crossings on the AC power line. By monitoring for the zero-crossings, X-10 devices know when to transmit or receive X-10 information. A binary '1' is represented by a 1 ms long burst of 120 kHz, near the zero-crossing point of the AC. A binary zero is represented by the lack of the 120 kHz burst.

FIGURE A-1: X-10 TRANSMISSION TIMING



A complete X-10 message is composed of a start code (1110), followed by a house code, followed by a key code. The key code may be either a unit address or a function code, depending on whether the message is an address or a command. Table A-1 and Table A-2 show the possible values of the house and key codes.

TABLE A-1: HOUSE CODES

House Addresses	House Codes			
	H1	H2	H4	H8
A	0	1	1	0
B	1	1	1	0
C	0	0	1	0
D	1	0	1	0
E	0	0	0	1
F	1	0	0	1
G	0	1	0	1
H	1	1	0	1
I	0	1	1	1
J	1	1	1	1
K	0	0	1	1
L	1	0	1	1
M	0	0	0	0
N	1	0	0	0
O	0	1	0	0
P	1	1	0	0

TABLE A-2: KEY CODES

Unit Addresses	Key Codes				
	D1	D2	D4	D8	D16
1	0	1	1	0	0
2	1	1	1	0	0
3	0	0	1	0	0
4	1	0	1	0	0
5	0	0	0	1	0
6	1	0	0	1	0
7	0	1	0	1	0
8	1	1	0	1	0
9	0	1	1	1	0
10	1	1	1	1	0
11	0	0	1	1	0
12	1	0	1	1	0
13	0	0	0	0	0
14	1	0	0	0	0
15	0	1	0	0	0
16	1	1	0	0	0
Function Codes	All Units Off	0	0	0	1
	All Units On	0	0	0	1
	On	0	0	1	0
	Off	0	0	1	1
	Dim	0	1	0	0
	Bright	0	1	0	1
	All Lights Off	0	1	1	0
	Extended Code	0	1	1	1
	Hail Request	1	0	0	0
	Hail Acknowledge	1	0	0	1
	Pre-set Dim	1	0	1	X
	Extended Code (Analog)	1	1	0	0
	Status = On	1	1	0	1
	Status = Off	1	1	1	0
	Status Request	1	1	1	1

When transmitting the codes in Table A-1 and Table A-2, two zero-crossings are used to transmit each bit as complementary bit pairs (i.e., a zero is represented by 0-1, and a one is represented by 1-0). For example, in order to send the house code A, the four-bit code in Table A-1 is 0110, and the code transmitted as complimentary bit pairs is 01101001. Since house and key codes are sent using the complimentary format, the start code is the only place where the pattern 1110 will appear in an X-10 data stream.

The key code, which is 5-bits long in Table A-2, takes 10 bits to represent in the complimentary format. Because the last bit of the key code is always zero for a unit address and one for a function code, the last bit of the key code can be treated as a suffix that denotes whether the key code is a unit address or function code.

A complete block of data consists of the start code, house code, key code and suffix. Each data block is sent twice, with 3 power line cycles, or six zero-crossings, between each pair of data blocks.

For example, to turn on an X-10 module assigned to house code A, unit 2, the following data stream would be sent on the power line, one bit per zero-crossing.

First, send the address twice:

```

1110      01101001    10101001      01
START     HOUSE A      UNIT 2        Suffix

```

```

1110      01101001    10101001      01
START     HOUSE A      UNIT 2        Suffix

```

Next, wait for three cycles (six zero-crossings):

```
000000
```

Then, send the command twice:

```

1110      01101001    01011001      10
START     HOUSE A      ON            Suffix

```

```

1110      01101001    01011001      10
START     HOUSE A      ON            Suffix

```

Lastly, wait for three cycles (six zero-crossings) before sending the next block:

```
000000
```

There are exceptions to this format. For example, the bright and dim codes do not require the 3-cycle wait between consecutive dim commands or consecutive bright commands. For a complete discussion of all X-10 messages, please refer to the X10 Wireless Technology, Inc. web site (see the "USEFUL WEB REFERENCES" section).

APPENDIX B: HOME CONTROLLER OPERATING INSTRUCTIONS

Welcome Screen

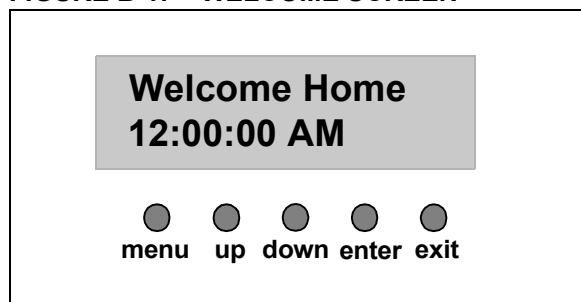
The home controller user interface consists of five buttons and a 2 x 16 LCD. Upon power-up, the Welcome screen is displayed. This screen displays a welcome message and the time. Immediately, the seconds begin incrementing and the PICmicro MCU begins keeping track of the time.

Figure B-1 shows the Welcome screen and the location and functionality of each button. Depending on the screen viewed, each of the five buttons performs a different function.

When the Welcome screen is displayed, the buttons enable access to the following functions:

- Press **menu** to enter the Select Function screen.
- Press **up** to brighten the lamp that is plugged into the home controller.
- Press **down** to dim the lamp.
- Press **enter** to turn the lamp on.
- Press **exit** to turn the lamp off.

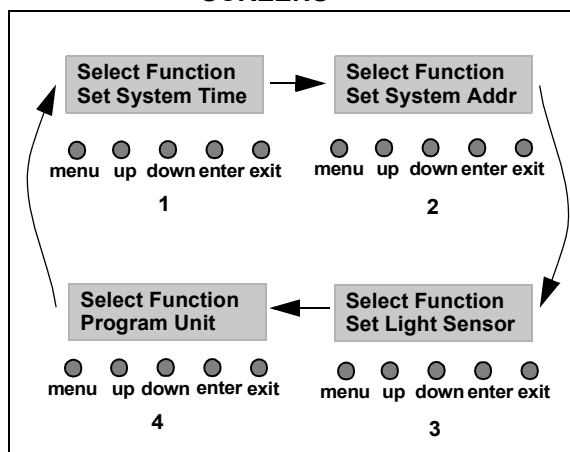
FIGURE B-1: WELCOME SCREEN



Select Function Screen

When viewing the Welcome screen, the **menu** button enables access to the Select Function screen. Each successive press of the **menu** button cycles through the four main functions of the user interface: setting the system time, setting the system address, setting the light sensor, or programming the unit on and off times, as illustrated in Figure B-2.

FIGURE B-2: SELECT FUNCTION SCREENS



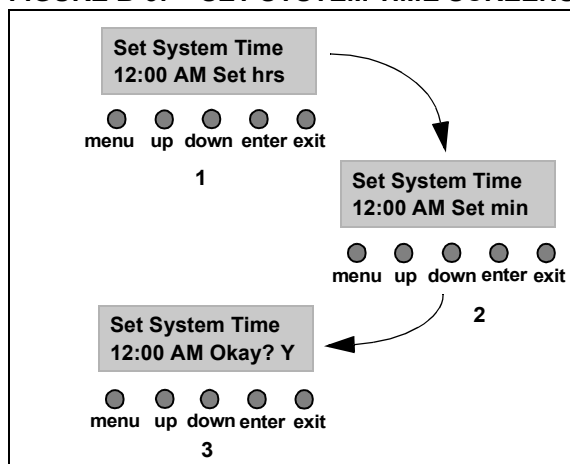
Set System Time Screen

Use the **Set System Time** screen to set the time.

SETTING SYSTEM TIME

1. Starting from the Welcome screen, press **menu** until the **Set System Time** screen is displayed and press **enter**.
2. Press **up/down** to set the hours.
3. Press **enter** when the correct hour, including AM or PM, has been selected.
4. Repeat this process to set the minutes.
5. If the time is correct, select **Y** (the default) using the **up/down** buttons and press **enter**. This returns to the Welcome screen with the new time displayed.
6. If the time is not correct, select **N** and press **enter**. This will return the user to step 2 so the correct time can be entered.
7. Press **exit** at any time to return the user to the Welcome screen without saving the new time.

FIGURE B-3: SET SYSTEM TIME SCREENS



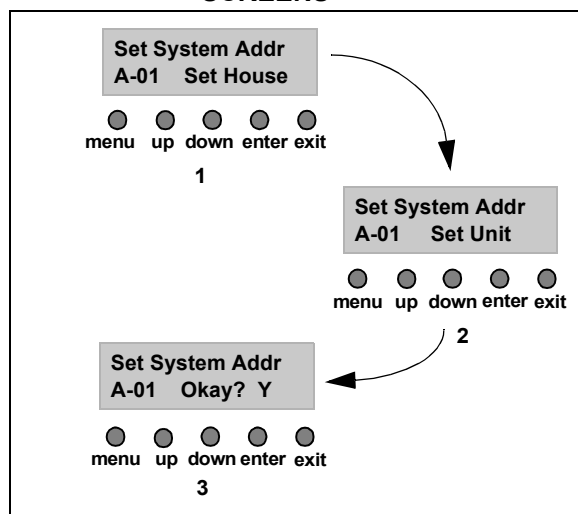
Select System Address Screen

Use the **Set System Address** screen to set the house address and unit address of the home controller.

SETTING HOUSE/UNIT ADDRESS

1. From the Welcome screen, press **menu** until the **Set System Addr** screen is displayed and press **enter**.
2. Press **up** or **down** to set the house address (a letter from A - P).
3. Press **enter** when the house address has been selected.
4. Repeat steps 2 and 3 to set the unit address (a number from 1 - 16).
5. If the house and unit addresses are correct, select **Y** (the default) using the **up/down** buttons and press **enter**. This returns to the Welcome screen with the new address stored in non-volatile memory.
6. If the address is not correct, select **N** and press **enter**. This will return the user to step 2.
7. Press **exit** at any time to return the user to the Welcome screen without saving the new address.

FIGURE B-4: SET SYSTEM ADDRESS SCREENS



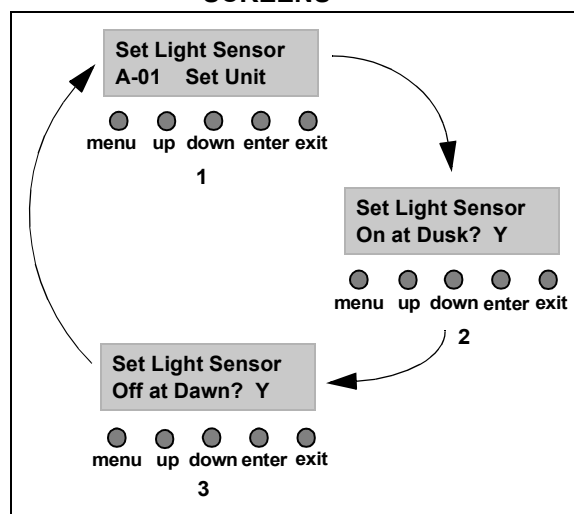
Set Light Sensor Screen

Use the **Set Light Sensor** screen to select whether units turn on at dusk, or off at dawn.

SETTING THE LIGHT SENSOR

1. From the Welcome screen, press **menu** until the **Set Light Sensor** screen is displayed and press **enter**.
2. Press **up** or **down** to select the desired unit. The house address will already be set to the system house address.
3. Press **enter** when the desired unit address has been selected.
4. Press **up** or **down** to select whether or not the unit should turn on at dusk, and press **enter**.
5. Repeat this process to set other units as desired.
6. Press **exit** to return to the Welcome screen. Pressing **exit** while the "On at Dusk" or "Off at Dawn" prompt is displayed will return the user to the Welcome screen without modifying that parameter.

FIGURE B-5: SET LIGHT SENSOR SCREENS



Program Unit Screen

Use the **Program Unit** screen to program on and off times for different units.

PROGRAMMING UNIT ON AND OFF TIMES

1. From the Welcome screen, press **menu** repeatedly until the **Program Unit** screen is displayed and press **enter**.
2. Press **up** or **down** to select the desired unit. The house address will already be set to the system house address.
3. Press **enter** when the unit address has been selected.
4. Press **up** or **down** to set the 'on' time hours. Hours set to '00' means that the unit will not be turned on at any time.
5. Press **enter** when the correct hour, including AM or PM, has been selected.
6. Repeat this process to set the 'on' time minutes. If the hour has been set to '00', then the minutes will be set to '00' automatically.
7. If the time is correct, select **Y** (the default) using the **up/down** buttons and press **enter**. The user will be prompted to program the 'off' time in a similar fashion.
8. If the time is not correct, select **N** and press **enter**. This allows the user to re-enter the hour and minutes by returning to step 2.
9. Repeat this process to set the 'on' and 'off' time for other units as desired.
10. Press **exit** to return to the Welcome screen. Pressing **exit** while the "Set Hours" or "Set Min" prompt is displayed will return the user to the Welcome screen without modifying any parameters.

FIGURE B-6: PROGRAM UNIT 'ON' TIME SCREENS

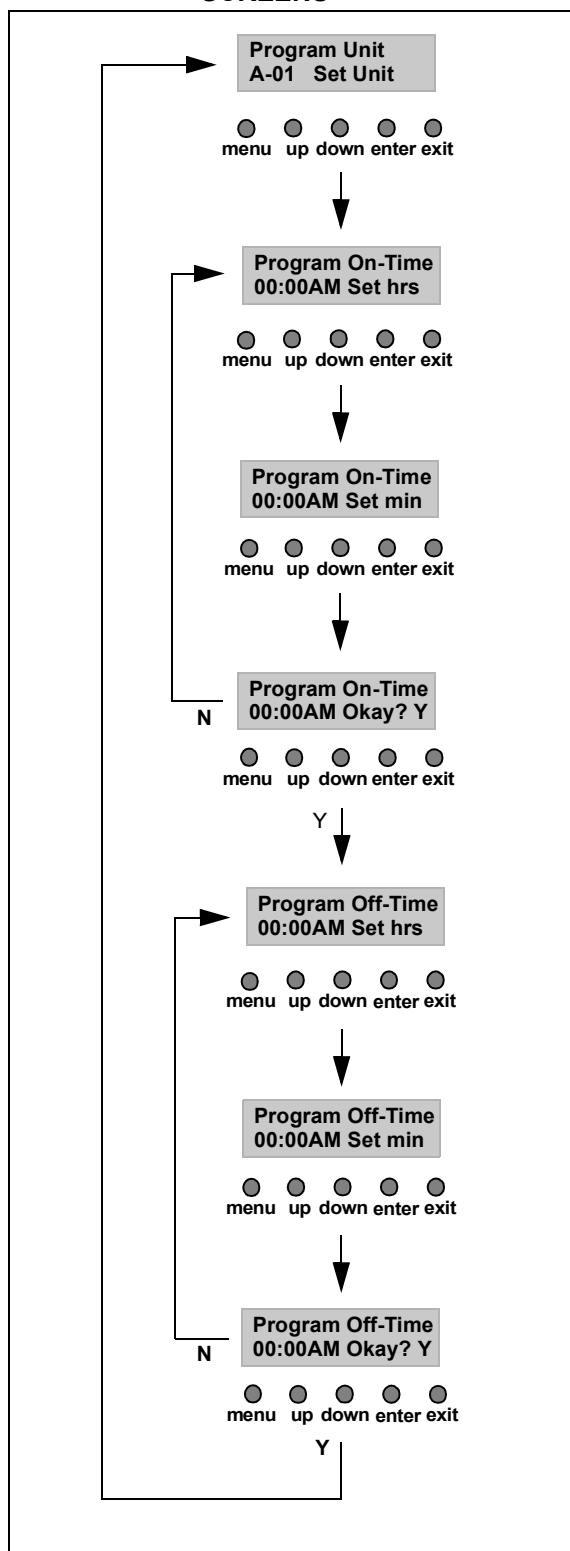


FIGURE C-2: SHEET 2 OF 5

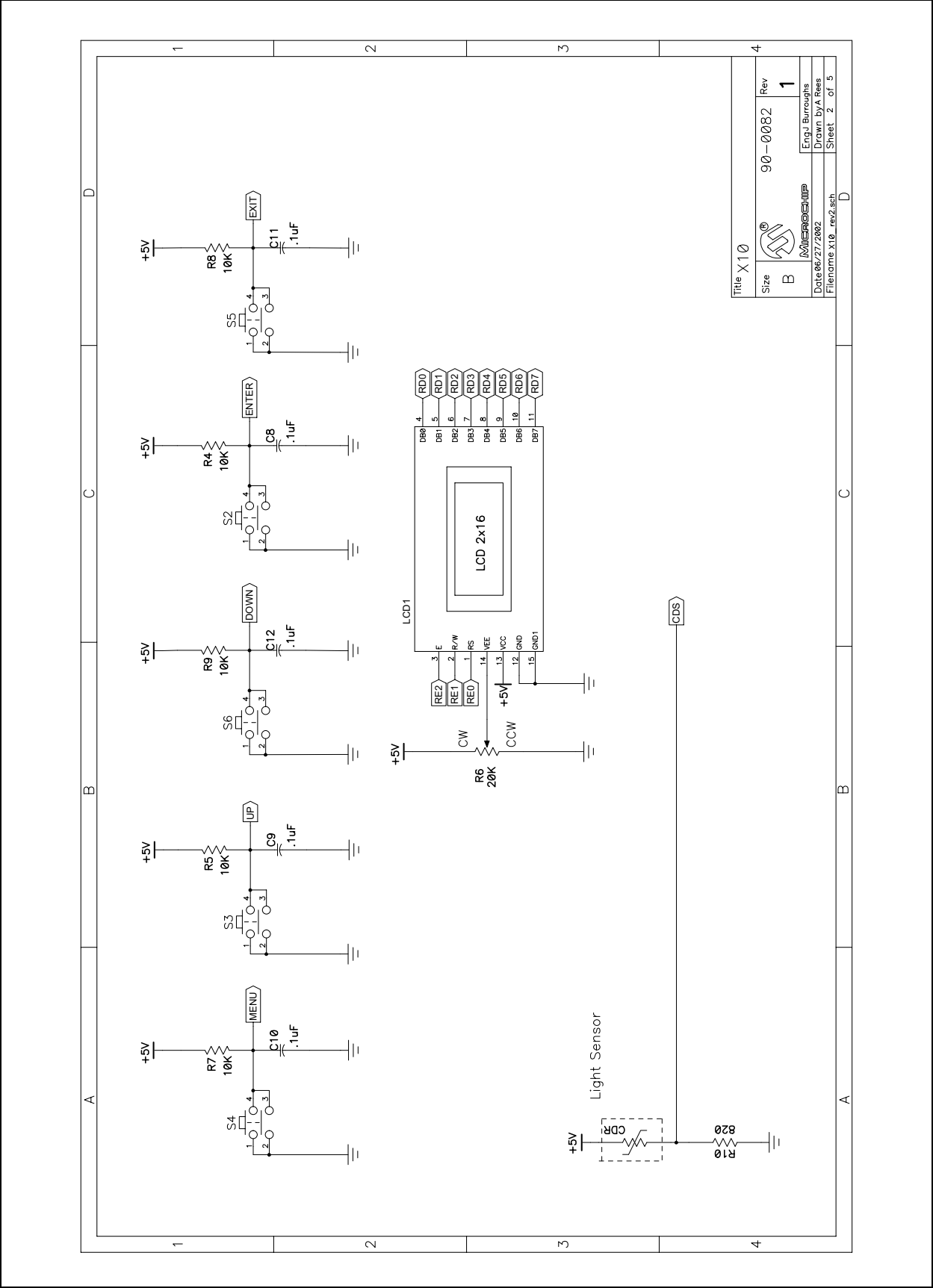


FIGURE C-3: SHEET 3 OF 5

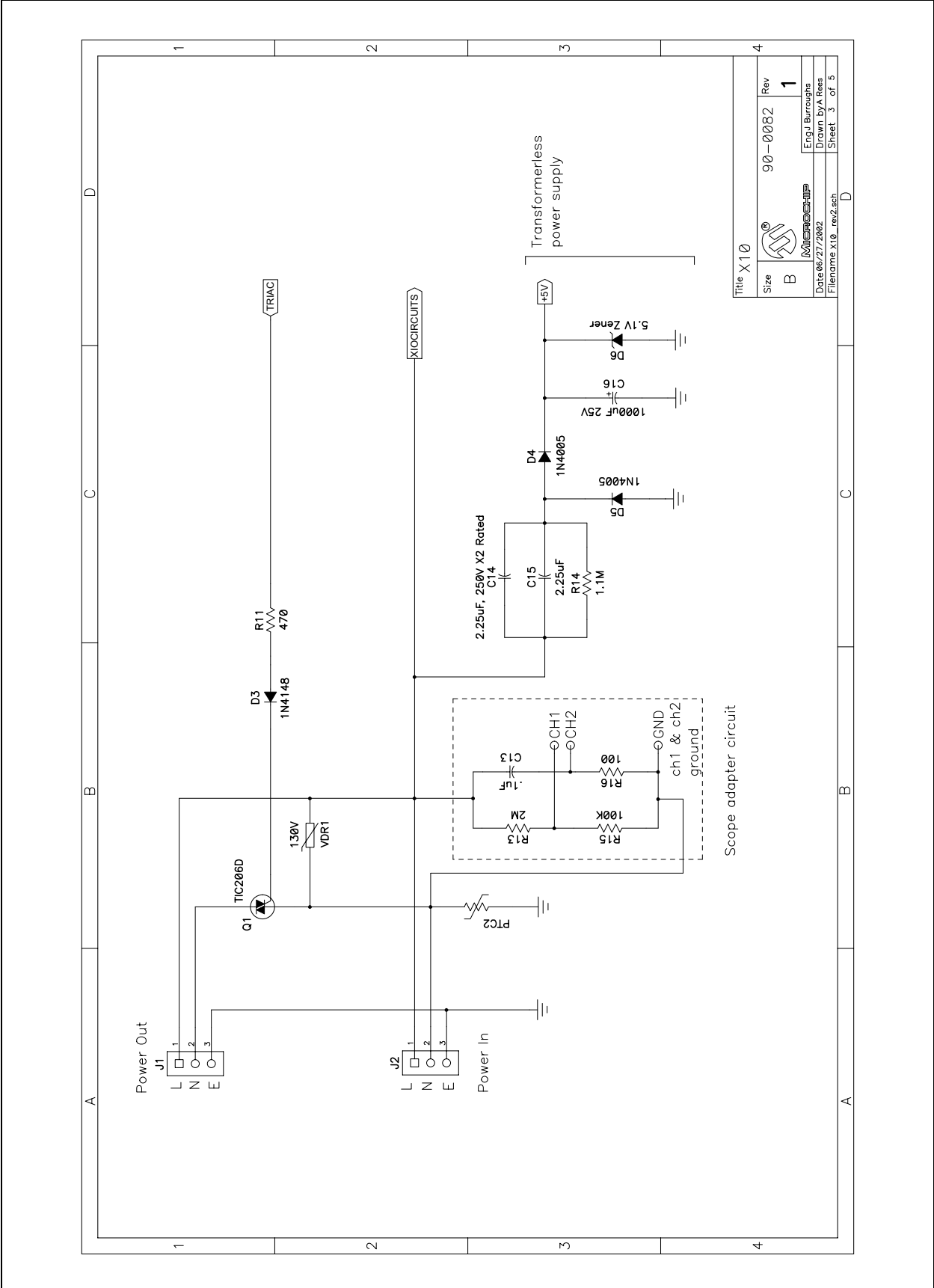


FIGURE C-4: SHEET 4 OF 5

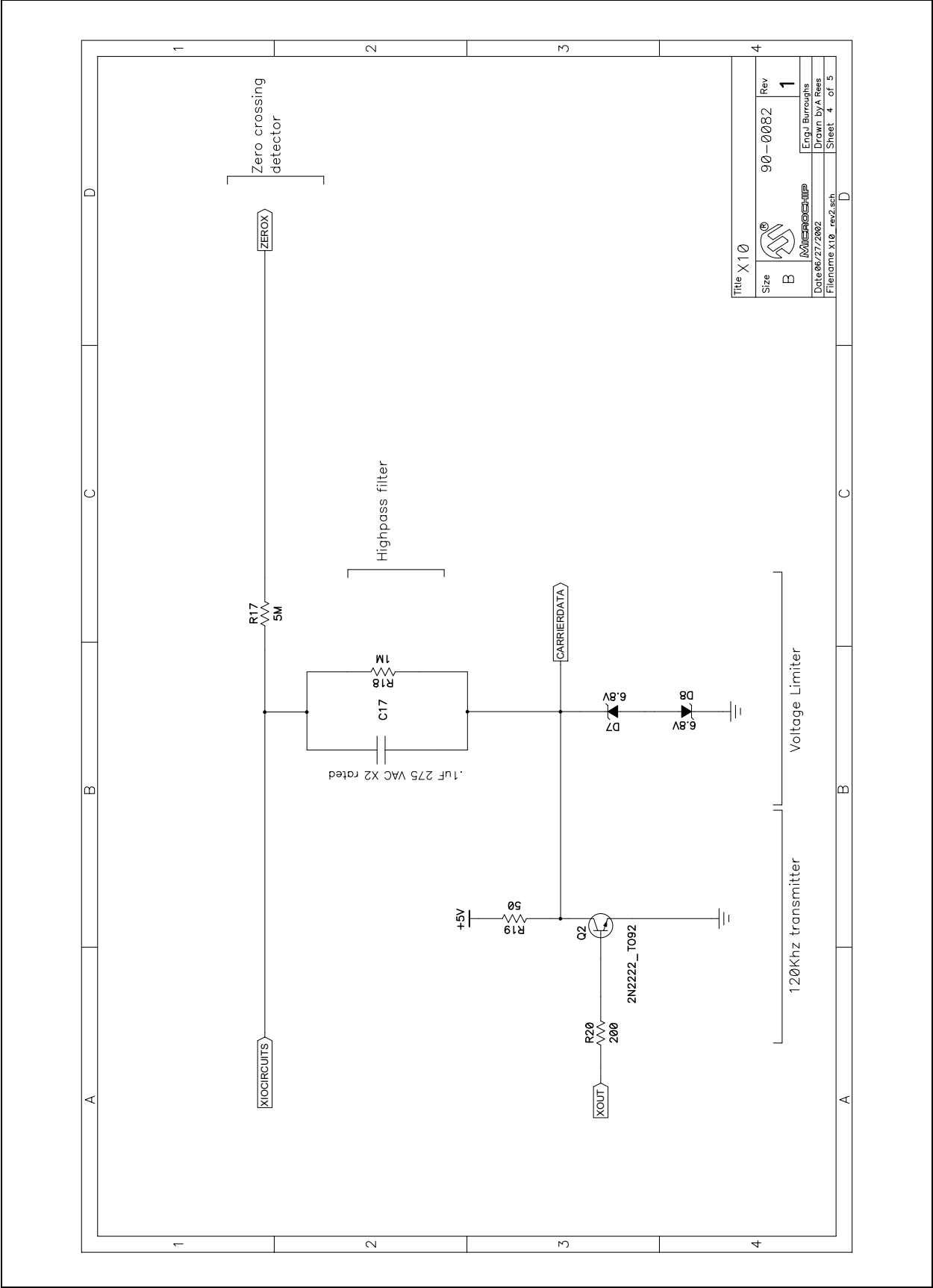
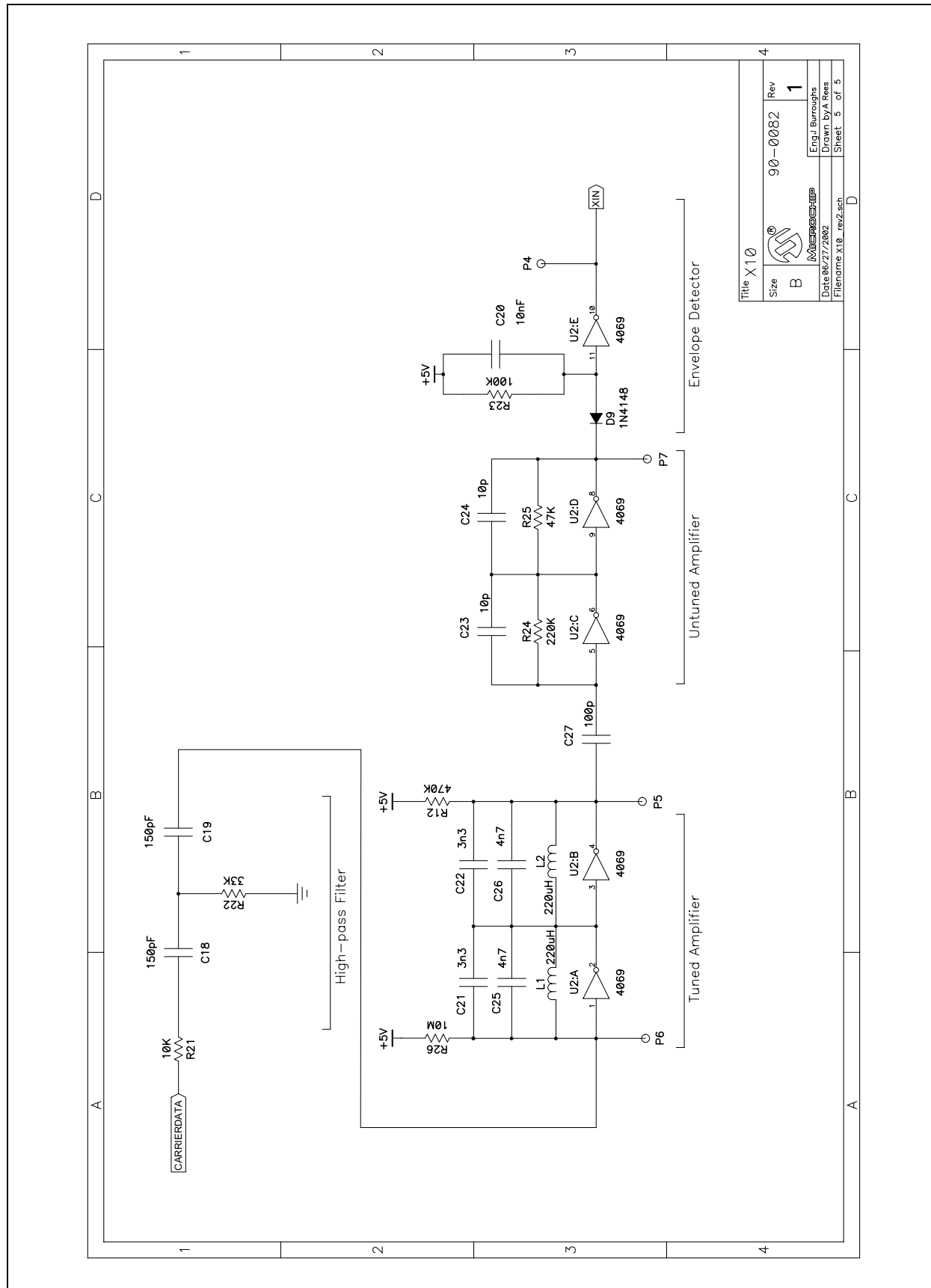


FIGURE C-5: SHEET 5 OF 5



APPENDIX D: PARTS LIST

Count	Reference	Value	Description
2	D7, D8	6.8V	Zener Diode
2	D4, D5	1N4005	Diode
2	D3, D9	1N4148	Diode
1	D6	5.1V	Zener Diode
1	Q2	2N2222	NPN Transistor
1	J2	Power In	Connector
2	J1	Power Out	Connector
1	U2	CD4069	HEX Inverters
8	C1, C2, C3, C8, C9, C10, C11, C12	0.1 μ F	Capacitor
4	C4, C5, C6, C7	15 pF	Capacitor
1	C13	0.1 μ F	Capacitor
2	C14, C15	2.25 μ F, 250V x2	Capacitor
2	C21, C22	3.3 nF	Capacitor
2	C25, C26	4.7 nF	Capacitor
1	C20	10 nF	Capacitor
2	C23, C24	10 pF	Capacitor
1	C27	100 pF	Capacitor
2	C18, C19	150 pF	Capacitor
1	C16	1000 μ F, 25V	Capacitor
1	C17	0.1 μ F, 275 VAC x2	Capacitor
1	Y2	7.680 MHz	Crystal
1	Y1	32 kHz	Crystal
2	L1, L2	220 μ H	Axial Lead Inductor
1	LCD1	CG161	HD44780-based 2x16 Liquid Crystal Display
2	D1, D2	LTL-94PEKTA	LEDs
1	U1	PIC16F877A	Microcontroller
1	R6	20 k Ω	Potentiometer
1	PTC1		CdS Cell
1	PTC2	Resettable PTC's	Resistor
1	R18	1 M Ω	Resistor
1	R14	1.1 M Ω	Resistor
1	R13	2 M Ω	Resistor
1	R17	5 M Ω	Resistor
7	R1, R4, R5, R7, R8, R9, R21	10 k Ω	Resistor
1	R22	33 k Ω	Resistor
1	R25	47 k Ω	Resistor
1	R19	50 k Ω	Resistor
1	R16	100 k Ω	Resistor
2	R15, R23	100 k Ω	Resistor
1	R20	200 Ω	Resistor
1	R24	220 k Ω	Resistor
2	R2, R3	680 Ω	Resistor
1	R10	820 Ω	Resistor
1	R11	470 Ω	Resistor
1	R12	470 k Ω	Resistor
1	R26	10 M Ω	Resistor
6	S1, S2, S3, S4, S5, S6		Push Button Switches
7	P1, P2, P3, P4, P5, P6, P7		Test Points
1	Q1	TIC206D	Sensitive Gate Triac
1	VDR1	130V	Varistor (Voltage Dependent Resistor)

APPENDIX E: SOURCE CODE

Due to size considerations, the complete source code for this application note is not included in the text. A complete version of the source code, with all required support files, is available for download as a Zip archive from the Microchip web site, at:

www.microchip.com

NOTES:

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.

Trademarks

The Microchip name and logo, the Microchip logo, KEELoQ, MPLAB, PIC, PICmicro, PICSTART and PRO MATE are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.


FilterLab, microID, MXDEV, MXLAB, PICMASTER, SEEVAL and The Embedded Control Solutions Company are registered trademarks of Microchip Technology Incorporated in the U.S.A.

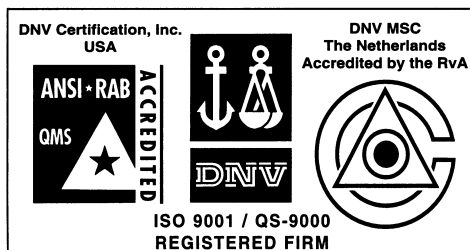
dsPIC, dsPICDEM.net, ECONOMONITOR, FanSense, FlexROM, fuzzyLAB, In-Circuit Serial Programming, ICSP, ICEPIC, microPort, Migratable Memory, MPASM, MPLIB, MPLINK, MPSIM, PICC, PICDEM, PICDEM.net, rPIC, Select Mode and Total Endurance are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

Serialized Quick Turn Programming (SQTP) is a service mark of Microchip Technology Incorporated in the U.S.A.

All other trademarks mentioned herein are property of their respective companies.

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

 Printed on recycled paper.



Microchip received QS-9000 quality system certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona in July 1999 and Mountain View, California in March 2002. The Company's quality system processes and procedures are QS-9000 compliant for its PICmicro® 8-bit MCUs, KEELoQ® code hopping devices, Serial EEPROMs, microperipherals, non-volatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001 certified.



WORLDWIDE SALES AND SERVICE

AMERICAS

Corporate Office

2355 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

2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7966 Fax: 480-792-4338

Atlanta

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

Boston

2 Lan Drive, Suite 120
Westford, MA 01886
Tel: 978-692-3848 Fax: 978-692-3821

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: 972-818-7423 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

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 Northam 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
Bei Hai Wan Tai Bldg.
No. 6 Chaoyangmen Beidajie
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: 86-28-86766200 Fax: 86-28-86766599

China - Fuzhou

Microchip Technology Consulting (Shanghai)
Co., Ltd., Fuzhou Liaison Office
Unit 28F, 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.
Room 701, Bldg. B
Far East International Plaza
No. 317 Xian Xia Road
Shanghai, 200051
Tel: 86-21-6275-5700 Fax: 86-21-6275-5060

China - Shenzhen

Microchip Technology Consulting (Shanghai)
Co., Ltd., Shenzhen Liaison Office
Rm. 1315, 13/F, Shenzhen Kerry Centre,
Renminnan Lu
Shenzhen 518001, China
Tel: 86-755-82350361 Fax: 86-755-82366086

China - Hong Kong SAR

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'Shaugnessy 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: 81-45-471-6166 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, 188980
Tel: 65-6334-8870 Fax: 65-6334-8850

Taiwan

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

EUROPE

Austria

Microchip Technology Austria GmbH
Durisolstrasse 2
A-4600 Wels
Austria
Tel: 43-7242-2244-399
Fax: 43-7242-2244-393

Denmark

Microchip Technology Nordic ApS
Regus Business Centre
Lautrup høj 1-3
Ballerup 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 - 1er Etage
91300 Massy, France
Tel: 33-1-69-53-63-20 Fax: 33-1-69-30-90-79

Germany

Microchip Technology GmbH
Steinheilstrasse 10
D-85737 Ismaning, Germany
Tel: 49-89-627-144 0 Fax: 49-89-627-144-44

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

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

10/18/02