Home Automation Communication System I

By

Brett McNerney
Tim Evans

Submitted to:

Dr Gary Dempsey

EE451 Senior Laboratory

September 26, 2003
# Table of Contents

Abstract | p.ii
---|---
Objective | p.1
Specifications | p.1
Division of Labor | p.2
Software Status | p.2
Hardware Status | p.3
Problems | p.3
Conclusion | p.3

Appendix A

Hardware Schematic | a-1
PSPICE Simulation | a-2
Abstract
This article contains the objectives, specifications, status, and current problems of a laboratory project to design a home automation communication system. An EMAC 80515 microcontroller development board will be used to transmit a signal over a phone wire. This signal must be converted to TTL logic levels and sent to another EMAC 80515 microcontroller development board. Utilizing the onboard A/D chip, the microprocessor must translate these signals to commands to turn on various devices. The entire system must run in real-time and be designed for mass-production (i.e. Monte Carlo analysis will be performed in PSPICE).
Objectives:
The purpose of this project is to develop a Home Automation Communication System. The system will receive a signal from the transmitter which will then turn on various devices connected to the receiver. The signal received from the transmitter must be converted to TTL levels in order to be processed by the EMAC 80515 microcontroller development board. All code will be compiled using the Keil uVision software package, and all hardware designs will undergo Monte Carlo analysis in PSPICE.

Specifications:
1) Power Supply: ±5VDC (± 5%) for hardware interface, provided to students.
2) Product Temperature: 0 to 120 degrees F.
3) Phone line wiring: 24 AWG copper, maximum length to be determined by students.
4) Port 4.0: flip pin to show fastest interrupt timing.
5) Port 4.1: Turn on LED for 2 seconds when time bit is received.
6) Port 4.2: Toggle LED on/off when frame bit received.
7) Other Port 4 pins: show other timing generated by interrupts.
8) Keypad: After initialization, key=command for LCD display or override control of the output devices.
9) Output Devices: Turn devices on when corresponding received bit D7-D0 is logic 1: D7: turn on outdoor lights (a 5V relay will be provided to switch on/off 120VAC light). D6: turn on sprinkler system water pump (a 5V relay will be provided to switch on/off 120VAC pump). D5: activate security lock (a 5V relay will be provided to switch on/off 120V solenoid). D4: turn on indoor lights (a 5V relay will be provided to switch on/off 120VAC light). D3: turn on security alarm (a Sonalert device will be provided). D2: turn on stereo (a 5V relay will be provided). D1: turn on coffee maker (a 5V relay will be provided). D0: turn on TV (a 5V relay will be provided).

10) 2-Line LCD Format (See Table 1):

<table>
<thead>
<tr>
<th>CMD or Key</th>
<th>Line 1</th>
<th>Line 2</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>after startup</td>
<td>ENTER A-D FOR CMD</td>
<td>OR 0-7 FOR OVERRIDE</td>
<td>main menu</td>
</tr>
<tr>
<td>CMD B</td>
<td>D7 6 5 4 3 2 1 0</td>
<td>actual received bits</td>
<td></td>
</tr>
<tr>
<td>CMD C</td>
<td>800 AM</td>
<td></td>
<td>Show current time in hr/min</td>
</tr>
<tr>
<td>CMD A</td>
<td>SW reset</td>
<td></td>
<td>Hold on display for 1sec</td>
</tr>
<tr>
<td>CMD D</td>
<td>ENTER A-D FOR CMD</td>
<td>OR 0-7 FOR OVERRIDE</td>
<td>Back to main menu</td>
</tr>
<tr>
<td>CMD 0-7</td>
<td>OUTPUT OVERRIDE X</td>
<td></td>
<td>Toggle output device on/off, X=device (0-7)</td>
</tr>
</tbody>
</table>

Table 1:
LCD Display Content Based on Command Entered
**Division of Labor:**
Tim Evans – Hardware Design
- Convert the signal to TTL logic
- Buffer the signal for the microcontroller
- Design for Mass production (Extreme PSPICE simulations)
Brett McNerney – Software Design
- A/D processing
- Keyboard Support
- LCD Display
- 24 hour clock

**Software Status:**

![Software Flow Chart](image)

**Figure 1: Software Flow Chart**
Currently, about 25% of the code has been developed; however, 90% of the code is expected to be completed and tested on 9/30/2003.
Hardware Status:
Currently, the entire circuit has been designed in PSPICE. The Schematic is included in appendix a-2. The output when using ideal components is also included in the appendix a-2. The input to the comparator has an average voltage of 1V, and its $V_{\text{ref}} = 0.5V$.

![Hardware Block Diagram](image)

Figure 2: Hardware Block Diagram
The input signal may need to be amplified for better noise filtering. The comparator is used to amplify the output signal to TTL levels, as well as make prompt transitions.

Problems:
- Signal processing may prove difficult due to difficulties in deciphering bits.
- Noise could flip the comparator at undesired times.

Conclusion:
Currently the project is approximately 50% complete, with hardware being 75% done and software being 25% done. A lot of code still needs to be written and tested, and the hardware portion must be built, tested, and verified that it is ready for mass-production.