Final Presentation
Cell Phone Controlled Security System
16 April 2008

Team Members
Kevin Brown
Don DeLaMare
Brian Faires
6812 Microcontroller

Size 2.2" X 1.6"
6812 Microcontroller

32K Bytes Flash EEPROM
2K Bytes Ram
31 I/O Lines
8-Ch 16-bit Timers
SCI/SPI Ports
Key Wake-up port
8MHz Internal Bus
25MHz Operation
40 pin connector
RS-232 Serial Port
3 push buttons (2 user / reset)
3 LEDs (2 user / VDD)
We know how to use these 24 pins as I/O. We can use 5 more pins.
Step 1

Security System

Keypad → Alarm Box (MCU) → LCD Screen
Step 2

Security System

Keypad → Alarm Box (MCU) → LCD Screen
  |                  3                  |
  V  Text to Speech V
        2

Speaker
Step 3

Security System

Keypad 7 Alarm Box (MCU) 9 LCD Screen

Speaker

Text to Speech

Telephone Line

phone unit

Speaker

Telephone Line

phone unit
Step 5

Security System

Keypad
Alarm Box (MCU)
Motion Sensor
Door Sensor
Fire Sensor
Window Sensor
Text to Speech
Telephone Line
phone unit

LCD Screen
Relay Speaker
Relay Microphone
Relay Microphone
Relay Microphone

Diagram showing the connections between different components of a security system.
Step 6

Security System

Keypad
Motion Sensor
Door Sensor
Fire Sensor
Window Sensor

Alarms Box (MCU)

Text to Speech

LCD Screen
Speaker
Microphones

Telephone Line
Phone Unit

Relays
Answering Machine
Keypad

7 parallel wires with four inputs and three outputs. Each key corresponds to a row and a column.
LCD Display - LCD1031 16X2 Characters

- 16x2 LCD
- Yellow-green LED backlight
- Includes the HD44780 controller
- Machine pin male headers come out of back of board
8-bit MCU Interface with LCD Controller

Busy Flag Timing Sequence
# HD44780 Character Map

<table>
<thead>
<tr>
<th>Code</th>
<th>Characters</th>
</tr>
</thead>
<tbody>
<tr>
<td>XXX0000</td>
<td>ΔΩΠερ</td>
</tr>
<tr>
<td>XXX0001</td>
<td>!1AΩαη δαθενγη</td>
</tr>
<tr>
<td>XXX0010</td>
<td>&quot;2BRbr</td>
</tr>
<tr>
<td>XXX0011</td>
<td>#3CScs</td>
</tr>
<tr>
<td>XXX0100</td>
<td>$4DTdt</td>
</tr>
<tr>
<td>XXX0101</td>
<td>%5EUeu</td>
</tr>
<tr>
<td>XXX0110</td>
<td>&amp;6FVfv</td>
</tr>
<tr>
<td>XXX0111</td>
<td>'7GWgw</td>
</tr>
<tr>
<td>XXX1000</td>
<td>(8HXhx</td>
</tr>
<tr>
<td>XXX1001</td>
<td>)9IYiy</td>
</tr>
<tr>
<td>XXX1010</td>
<td>*:JZjz</td>
</tr>
<tr>
<td>XXX1011</td>
<td>+;KIkk</td>
</tr>
<tr>
<td>XXX1100</td>
<td>,&lt;L¥II</td>
</tr>
<tr>
<td>XXX1101</td>
<td>-=MJm</td>
</tr>
<tr>
<td>XXX1110</td>
<td>.&gt;Nⁿn→</td>
</tr>
<tr>
<td>XXX1111</td>
<td>/?O_0→</td>
</tr>
</tbody>
</table>
MCU interface with Text to Speech chip

Figure 15. Type IV SPI Transaction.
MCU interface with Phone Line

The microcontroller will have to pick up the phone on incoming calls if there is no answering machine. The microcontroller will constantly listen to the phone line for a ring (high voltage pulse) and answer the phone (allowing a current to flow back on the phone line) after the desired number of rings. Then it will send out a string to the text to speech chip to say, “enter code.” Then the microcontroller will listen for a low and a high frequency to detect a row and a column of a four by four matrix to match a button press. If the correct code was entered, it will go to the menus of the security system, then continue sending the menus’ strings to the text to speech chip and listen for button presses to switch between menus.
## Schedule

<table>
<thead>
<tr>
<th>Task Name</th>
<th>Duration</th>
<th>Start</th>
<th>Finish</th>
<th>April</th>
<th>May</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acquire Parts</td>
<td>28 days</td>
<td>Wed 4/23/08</td>
<td>Fri 5/30/08</td>
<td>4/23</td>
<td></td>
</tr>
<tr>
<td>Keypad Interface Working</td>
<td>4 wks</td>
<td>Mon 6/2/08</td>
<td>Fri 6/27/08</td>
<td></td>
<td>6/2</td>
</tr>
<tr>
<td>LCD Interface Working</td>
<td>1.05 mons</td>
<td>Mon 6/2/08</td>
<td>Mon 6/30/08</td>
<td></td>
<td>6/2</td>
</tr>
<tr>
<td>Interface Text to Speech Chip</td>
<td>1.05 mons</td>
<td>Tue 7/1/08</td>
<td>Tue 7/29/08</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Interface phone with MCU</td>
<td>1.05 mons</td>
<td>Fri 8/1/08</td>
<td>Fri 8/29/08</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Interface Microphone &amp; Speaker</td>
<td>1.1 mons</td>
<td>Mon 9/1/08</td>
<td>Tue 9/30/08</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Connect sensors</td>
<td>1 mon</td>
<td>Wed 10/1/08</td>
<td>Tue 10/28/08</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Design switch for Answering Ma</td>
<td>1 mon</td>
<td>Mon 11/3/08</td>
<td>Fri 11/28/08</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Demonstrate complete system</td>
<td>2 days?</td>
<td>Mon 12/1/08</td>
<td>Tue 12/2/08</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Gantt Chart**

- **June**
  - 5/30
  - 6/30
  - 7/1
- **July**
  - 6/27
  - 8/1
- **August**
  - 8/29
  - 9/1
- **September**
  - 9/30
  - 10/1
  - 10/30
  - 11/3
- **October**
  - 10/28
- **November**
  - 11/28
- **December**
  - 12/1
State Machine

Unarmed
Change Settings or Arm System

Green State
Check Sensors

Yellow State
Unarm System || goto Orange

Orange State
Contact Primary Phone

Red State
Sound Siren and/or Call Backup Numbers
State Machine

Unarmed
Change Settings or Arm System

Green State
Check Sensors

Yellow State
Unarm System || goto Orange

Orange State
Contact Primary Phone

Red State
Sound Siren and/or Call Backup Numbers

Keypad Object
State Machine

Unarmed
Change Settings or Arm System

Green State
Check Sensors

Yellow State
Unarm System || goto Orange

Orange State
Contact Primary Phone

Red State
Sound Siren and/or Call Backup Numbers

Incoming Call Object

Keypad Object

Outgoing Call Object
Check Sensors

Check Keypad

Update Incoming Phone Unit
Update Incoming Phone Unit

Check Sensors
Check Keypad

Beep and Prompt for Key Code

Poll Keypad, Buffer Input, Wait for 'Enter'

Incorrect Code, 1st or 2nd

3rd Incorrect Code

No Code Timeout
1. Phone line available
   - Call Primary Number
     - Play prerecorded message
       - Wait for any key press or timeout (Voice mail)
         - Red State
     - Connect speaker and mics
       - Prompt for code
         - Disconnect
   - Phone line busy with unconfirmed user
     - Prompt for code
     - Disconnect
   - Phone line busy with user
     - Process User Commands
Risks

Phone line input
- How quickly will our MCU be able to detect a button press?
- Will quickly pressing one key multiple times be detected as a single keypress?

LCD Support chip
- Needed for 16x2. We won't know if we can do this until we have the LCD display. May have to stick with 8x2 or 16x1.
Risks cont.

- Shortage of available pins: We will aim to use serial/serial to parallel communication where possible. If there is still a shortage we can use encoders and decoders.

- Program size: If the program cannot fit on 32KB of EEPROM, we may have to replace some of the user specified options with predefined functionality.
Risks cont.

- We want the user to be able to cut off a phone menu if they please. Since we don't know of a way to tell the TTS chip to 'stop talking' we may have to send it words at the rate of speech or ignore button presses mid-sentence.