Biometric Access Control System

Designed By: Paolo Bruno, Bilal Ahmed
Senior Design Summer 2009
Serial Communication

8 data bits, 1 stop bit, and no parity
Fingerprint Module Serial Protocol

*NOTE* Header Checksum = Command + Param1 + Param2 + Data Size + Error Code

* If data size is zero, then data and data check sum is not used.
* Start byte: 0x7E
**Request Connection Command – Determines if you are connected to the FIM3040**

<table>
<thead>
<tr>
<th>Command</th>
<th>0x01</th>
<th>Command</th>
<th>0x01</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter 1</td>
<td>X</td>
<td>Parameter 1</td>
<td>Result Succeeded</td>
</tr>
<tr>
<td>Parameter 2</td>
<td>X</td>
<td>Parameter 2</td>
<td>Fingerprint Count</td>
</tr>
<tr>
<td>Data Size</td>
<td>0</td>
<td>Data Size</td>
<td>0</td>
</tr>
<tr>
<td>Error Code</td>
<td>X</td>
<td>Error Code</td>
<td>Error Code</td>
</tr>
<tr>
<td>HDR Checksum</td>
<td>0x01</td>
<td>HDR Checksum</td>
<td>N / A</td>
</tr>
</tbody>
</table>

**NOTE** When a command packet is sent to the fingerprint module it returns an acknowledgement packet that informs you whether or not the command sent was successful.
The Structure of a Command Packet

<table>
<thead>
<tr>
<th>Request Connection Command</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Start Byte</strong></td>
</tr>
<tr>
<td>0x00 0x00 0x7E</td>
</tr>
</tbody>
</table>
AVR Programming Basic Framework

(1) Enable Rx / Tx on Uart1 and Uart0.

(2) Set Baud Rate by setting up UBRR registers.

(3) Decide between polling and using interrupts? Interrupts Chosen.

(4) Decide on terminal interface or GUI using Visual C#. Could not implement a GUI due to shortage of time.

(5) Store Fingerprint Module (FIM3040LV) Commands in EEPROM.

(6) Read and learn FIM3040 communications protocol manual.

(7) Buffer the incoming FIM3040 packets properly! Compare command packet with acknowledgement packet before proceeding.

(8) Make sure you have an updated version of the communication protocol. The fingerprint module had many quirks that needed to be figured out. Often times the communication protocol had wrong information and bad English!
PCB Advanced Circuits Barebone
Analog Security Breach Alarm

- An alarm was designed as a security measure, after several unsuccessful authentication attempts the alarm becomes active.

- Oscillator and NE555 timer were used to generate signals at 500MHZ.

- Mixer Analyzer circuit was designed to select certain output based on the input voltage using the N-Channel MOSFETs (IRF510).

- With the use of diodes non-ideal OpAmp can produce a steady state oscillation.

- Output of the oscillator is fed into a mixer amplifier.
PCB Security Breach Alarm
Finished Product
Possible Improvement

• One of the possible improvement is to design the Access system with keypad as an alternate entry solution

• Design GUI to make it easy to use interface