THE ARDUINO SECURITY SYSTEM

Ben Ackerman
CSC 172H

BACKGROUND: Security systems have been around for a very long time, even before the introduction of microcontrollers. Over the course of all that time, they have come quite a long way. They have gone from being simple analog circuits with keyswitches and mechanical bells to being sophisticated digital systems that can automatically report alarms and status information to a monitoring center and even offer home automation to some extent. However, many modern security systems have some hidden shortcomings. For instance, one system that is advertised as having many innovative features still uses a POTS (“plain old telephone service”) line for monitoring by default (Elk Products). While reporting alarms in this fashion may be reliable, it can also be quite slow, not to mention it cannot be used in homes that do not have landline phone service (which, with cellphones becoming so common, is seen more and more often). Other modern systems get around the no-landline limitation by using an internal “virtual cellphone” of sorts to emulate POTS-based reporting (“Home Security Systems Comparison”). Not only is this still phone-based reporting, it also introduced the cost of needing to connect to the cellular network, and it can be affected by poor cellular reception. A better method of reporting alarms is IP-based reporting, where the alarm system uses an Internet connection to report alarms to the monitoring center. Some alarm systems do offer this capability, but it hasn't caught on as much as one would think (Elk Products). By using IP-based reporting, the alarm system can potentially send a detailed alarm report to a monitoring center in a split-second, allowing for authorities to be dispatched sooner in a potentially life-or-death situation.

Additionally, with any alarm system, there is always that one feature you wish it had. Be it an output that can be used in a specific situation, the ability to control a certain device, or some other feature, there often seems to be something that the manufacturer forgot about. For instance, my family was getting a hold-up alarm installed in the house, and we wanted there to be an LED that would light up when the system needed to be reset. However, it turned out that because the system was configured for the panic button to trigger a silent alarm (i.e. only report it to the monitoring company; don't do anything else), it would not turn on any of its outputs when that alarm was in its memory, so there was no way to know if it needed to be reset or not.

WHAT IS THIS PROJECT? The objective here is to build a fully-featured alarm system using the open-source Arduino platform, then to use any resources remaining on the Arduino boards to add additional features that could be used in specialized applications. On top of all of this, some older technologies that are still commonly used in alarm systems will be replaced with newer ones, such as IP-based instead of phone-based alarm reporting, and possibly a battery backup system using a lithium polymer battery instead of a lead acid one. Hopefully, when all is said and done, this will all come out to be cheaper than most alarm systems on the market, but if that ends up not working out, the plan is to have it offer more features than commercial systems for a price similar to that of those systems.

PLANNED FEATURES:
- All standard alarm system features implemented on the Arduino platform
- Wide array of control outputs that can be used in many specialized applications
- Fast IP-based alarm reporting: alarms can be reported to the monitoring center in a split-second, and other communication (date/time updates, automatic communication tests, bill payment notifications, and much more) can occur without tying up a phone line
• Zone multiplexing: expand the system's native capacity of 16 zones to 32 or possibly more without any additional hardware (uses the Arduino's analog inputs)
• Non-lead acid battery: the system will use the newer lithium polymer battery technology to reduce both the cost and the safety hazard posed by the system's battery
• Keypad daisy-chaining: additional keypads beyond the first one, rather than being connected directly to ports on the central controller, will be connected to the previous keypad, which can potentially reduce wiring and increase the number of keypads that can be connected (because it isn't limited by a finite number of ports on the central controller)
• More features will be added as time and money permit

HOW WILL THIS BE DONE? The work involved in this project can be divided into two main categories: hardware and software. Most of the hardware work will need to be done first (although some software work should be happening simultaneously so that there's something to test the hardware with). The hardware work consists primarily of assembling the central controller and keypad (although the housings for these will come later). The software work involves actually writing the software that drives this whole thing. The software work is then divided into two sub-categories: controller and keypad. Since those two components run on their own microcontrollers, each of them needs its own software. The work in those two sub-categories needs to happen somewhat simultaneously, however, as they both need to be in place in order to test anything. Once all the base work is done and all the main features work correctly, then it's time for the fun part: using any remaining resources to add extra features to the system. This part will then continue as long as it can until the project is due.

CURRENT ASSETS: There are many tutorials online about how to make a security system using an Arduino, but none of them are for a system quite as fully-featured as this project will be. For example, the alarm system tutorial that my parts supplier's website links to shows how to build a portable system that only protects one door, rather than a full-building system like this project aims to create (“Portable Alarm System”). However, information from those tutorials could be used to help with portions of this project. Also, I know someone who works in the alarm business, so I could ask him for advice and/or obtain spare parts from him if necessary.

TENTATIVE SCHEDULE:

<table>
<thead>
<tr>
<th>MILESTONE</th>
<th>EXPECTED COMPLETION DATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arrival of parts</td>
<td>DONE</td>
</tr>
<tr>
<td>Assemble hardware</td>
<td>10/31/14 (In progress)</td>
</tr>
<tr>
<td>Build/debug initial version of controller and keypad software</td>
<td>11/25/14</td>
</tr>
<tr>
<td>Add features as time/money permit</td>
<td>12/6/14</td>
</tr>
</tbody>
</table>
Works Cited

