Automated Home Security System

Chester Lau

Table of Contents

Introduction **2**

Hardware **2**

Parts List 2

Pinout 3

Software **3**

Complexities **4**

Completed Complexities: 4

Youtube Link **4**

Known Bugs and Shortcomings **4**

Future work **5**

Introduction:

The automated security system consists of a bluetooth module, wifi module, lcd screen, magnetic sensor module, and a servo. The system works by communicating with a bluetooth enabled device. One can toggle between the different modes by sending a 8 bit string at a time. There is a magnetic module to detect whether a door is opened, a servo to automatically unlock/lock the door, lcd screen to display the current status, and also a wifi module to update the user with various statuses of the security system.

At startup the alarm automatically boots into ARMED mode after the wifi module has connected. Next to deactivate the alarm, the user must send the passcode in 8 bit segments (0x12 then 0x34 then 0x56) via a bluetooth app. This will then deactivate the alarm , the lcd will display that the alarm is indeed deactivated and the servo will move to unlock the door. Next, to reactivate the alarm one sends (0x00) and the alarm will reactivate. This will cause the wifi module to send a POST HTTP request and the user's phone will receive a text message that the alarm is indeed active, and the lcd will also display it is active as well. The servo will also move to lock the door. Next, whenever the magnet module is activated while the alarm is active, the security system will detect there has been a breach. An alarm will sound and the wifi module will complete a POST HTTP request which will then call the user's phone to leave a message there has been a breach, the lcd will also display this status. To reset, one sends (0x88) and the alarm will reset back to active mode once again.

HARDWARE:

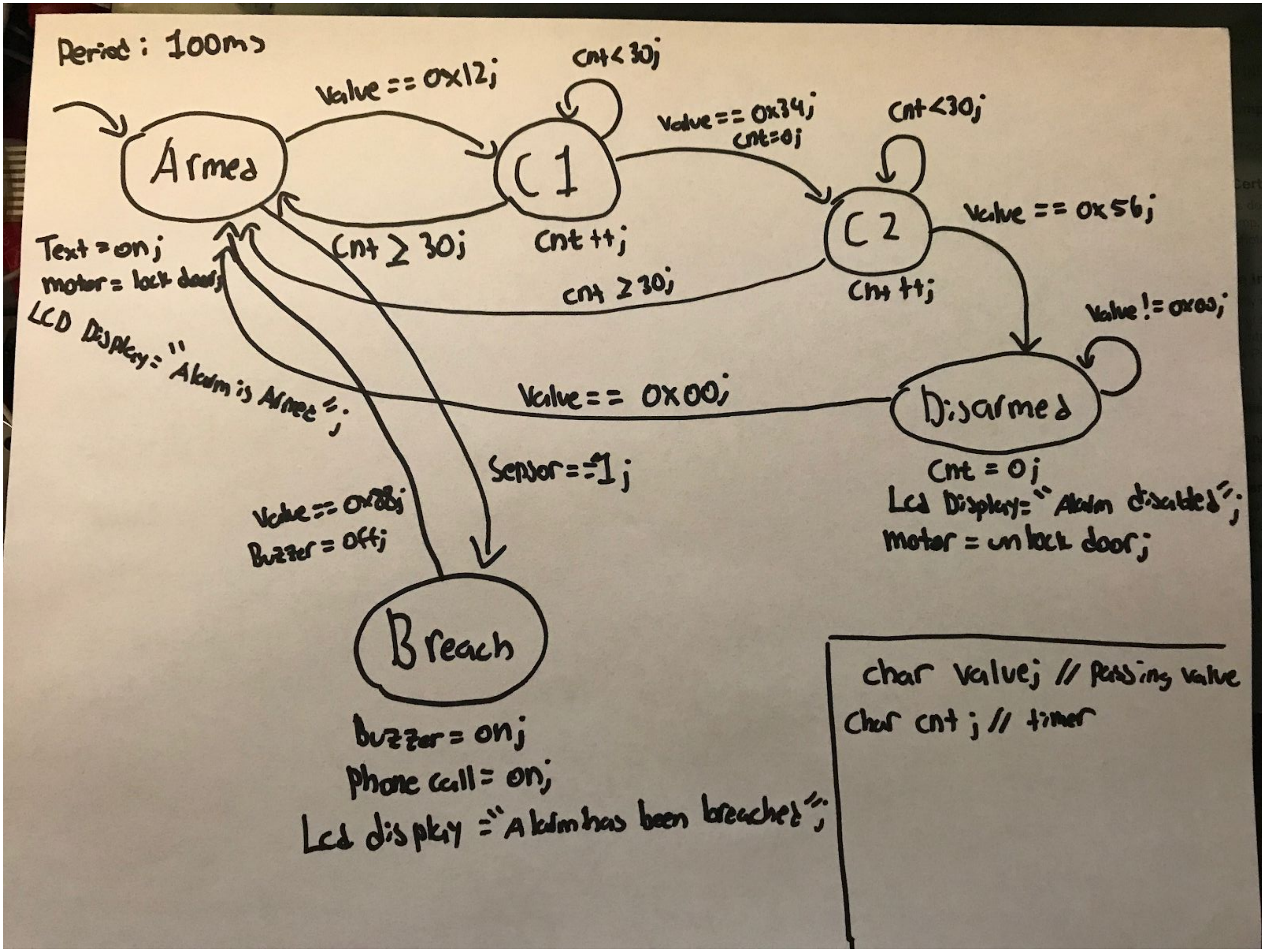
PARTS LIST:

- ATMEGA1284P microcontroller
- LCD Screen
- Buzzer/Speaker
- **SG51R Servo**
- **Magnetic Reed Switch Module**
- **HM-10 Bluetooth Module**
- **ESP8266 Esp-01 Wifi Module**

PINOUT:

LCD Screen: PC0-PC7, PD6-PD7
HM-10 Module: PD0-PD1
ESP8266: PD2-PD3
SG57R SERVO: PB6
Buzzer: PA0

SOFTWARE:



```
char value; // passing value
Char cnt ; // timer
```

COMPLEXITIES:

Completed Complexities:

- Utilizing UART to communicate with HM-10 module
- Utilizing UART to communicate with ESP8266 wifi module
- Using EEPROM to remember wifi login on ESP8266 wifi module
- Using EEPROM to remember previously connected devices on HM-10 module
- Getting ESP8266 and HM-10 to properly communicate and to receive/send all data with no arduino
- SG51R Servo motor
- Magnetic Reed Module

This project took me 40 hours to complete. The bluetooth module was stable and took me about 8 hours to research and configure it to properly remember devices and to send/receive data from my phone. The Wifi module however took me 30 hours to research and finally get it to work properly. I knew nothing about it , and it took me 30 hours to learn and complete these steps. First, I had to find a way to get the ATMEGA1284 to communicate with the module, this then lead me to having to buy an ftdi cable to debug the module. I discovered my module came blank and that it did not work on its own. I then had to compile binary files on Ubuntu and had to flash the wifi module with the ftdi cable so it could even turn on. This included having to try different wiring schemes and to individually partition the files on the ESP8266 eeprom by selecting which memory location each file was to be placed (0xFFEE000 etc.). Next, the module was able to accept AT commands via, RealTerm serial terminal, but it would not connect to wifi, had no ip, and kept getting a DNS fail. I had to reflash 4 different versions in the end and it finally worked. I then also had to learn how to port to the server that was completing my requests and how to properly send POST requests. I then had to configure the ATMEGA1284 to communicate to both wifi and bluetooth using 2 seperate baud rates and to send/receive data without conflicts.

YOUTUBE LINK:

https://www.youtube.com/watch?v=3pO7nYV1VaY

CODE REFERENCES:

http://www.electronicwings.com/avr-atmega/atmega16-interface-with-esp8266-module

http://www.martyncurrey.com/hm-10-bluetooth-4ble-modules/

KNOWN BUGS/SHORTCOMINGS:

The server that completes the text message and phone call might sometimes take up to 15 minutes to poll, therefore; there may be a delay in receiving notifications once in a while.

FUTURE WORK:

I would like to create a mobile app that is easy to use and visually appealing to control the security system. I would also like to transform the security system into a total home automation system. The user will be able to control his house and check on statuses when away from home. I can utilize HTTP protocols to have the user wirelessly control devices while at or away from home. For example, change the air conditioning settings, open garage doors, turn on/off lights, turn on the stove or oven, control the TV, etc.