Web-Enabled Alarm System

Contents

Description

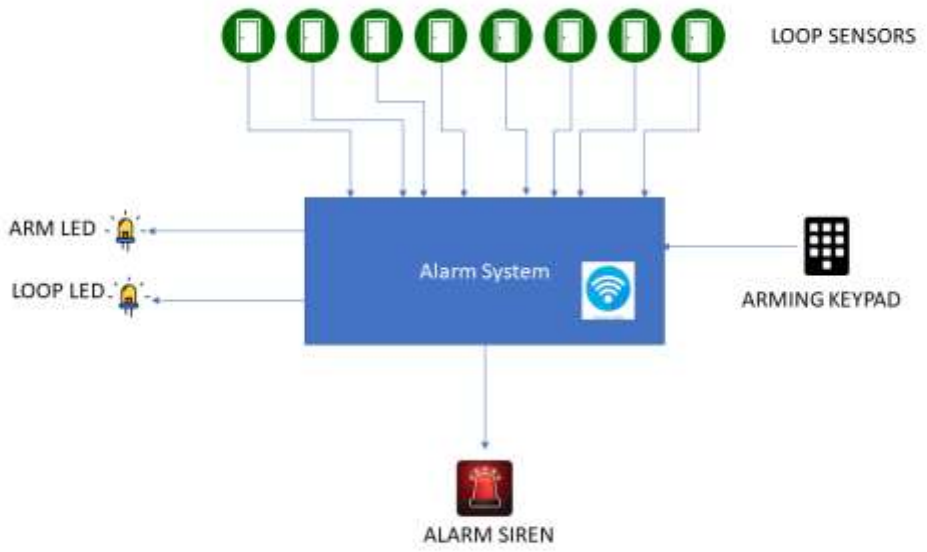

Figure 1 Alarm System Overview

This document describes the hardware and software details of a DIY (Do It Yourself) Web-Enabled Alarm System. This project is built around a NodeMCU module, which is a development board incorporating Wi-Fi and multiple GPIO into a small, Arduino-programmable unit. 8 loop inputs are provided, as well as a toggle-style arming input and outputs for an alarm siren, warning buzzer, and loop and Arm LED displays.

Figure 1 shows an overview of the system.

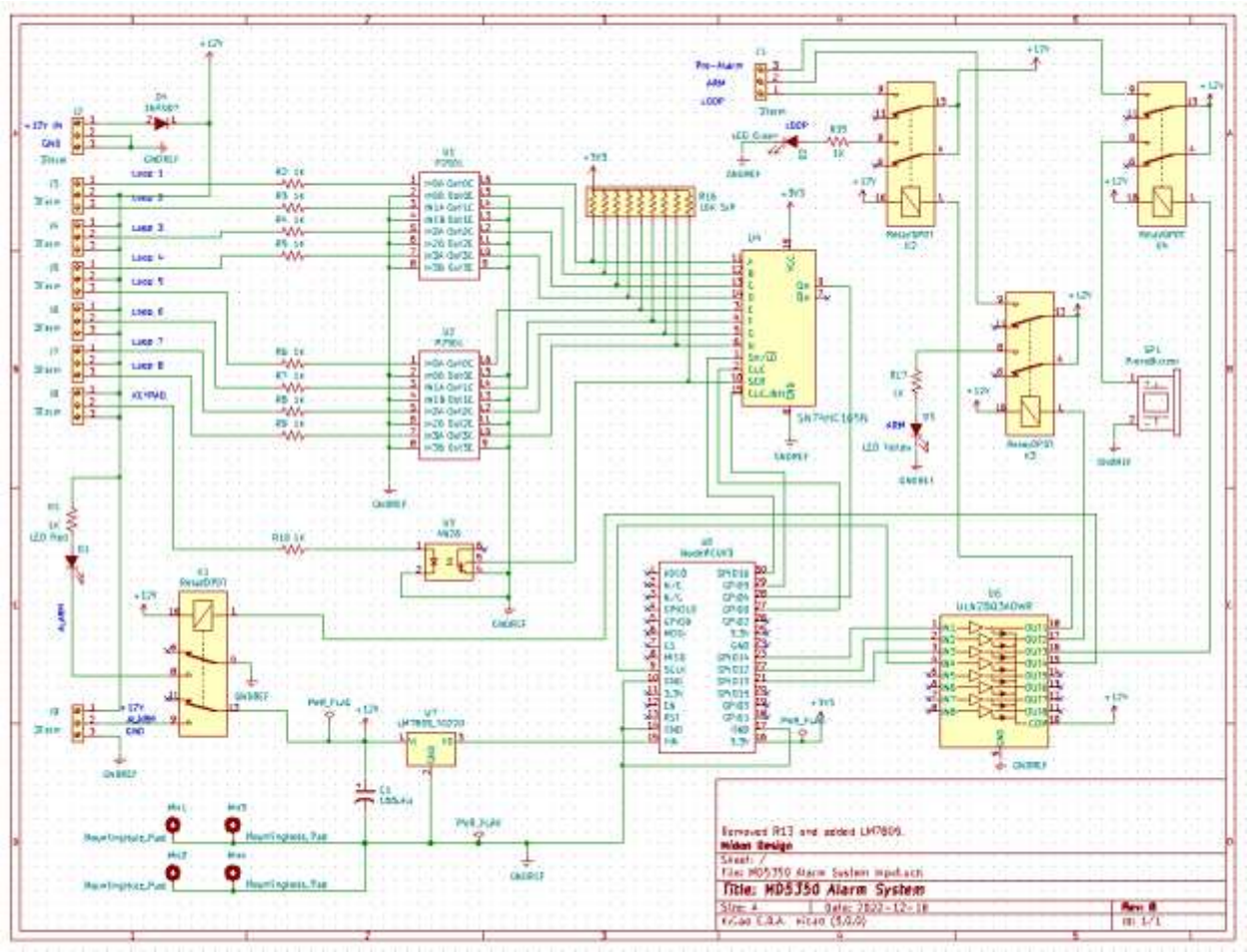

Figure 2 Alarm System Schematic

Hardware Details

The core of this design is U5, the NodeMCU development board. Connected to its GPIO's are the loop sensor inputs, via the 74HC165 parallel to serial chip U4. This was done to conserve the useable GPIO's since not all of them are useable without some limitations. The loop sensor inputs are opto-isolated by U1 and U2 in order to ensure that transients on the long loops do not affect the central processor. Similarly, the keypad input is protected by U3. Outputs from U5 are connected to U6 which level-converts the 3.3V outputs to 12V in order to drive the relays and LED's. Since U5 is not necessarily 12V tolerant, it is powered via U7 which provides it with 9V. Diode D4 is used to protect the power input from being connected in reverse. Note that R16 is required since the outputs from U1, U2 and U3 are open-collector.

You can order PCB's using the referenced Gerber file from just about any PCB house. I use JLCPCB in China and OshPark in the USA for PCB's.

Software Details

I would like to acknowledge Rui Santos for her work in developing the Wi-Fi and timer interrupt code used in this project.

I would also like to acknowledge Manoj R. Thakur for his guidance on a web server for the NodeMCU.

Please reference the link provided for the code details.

Like all Arduino software, the code begins by setting up global parameters and initializes the various libraries with required settings. Environment constants are then setup. There are many that need to be adjusted to your local environment such as email addresses, static IP addresses for the required servers, and names for the loops that will be connected to the system. Global variables are then setup. This includes the wrapper for the web page that will be used to connect to the system.

The timer interrupt is then configured. This interrupt fires every 1000mS (1 Second) and increments one of 3 timers used by the rest of the system: an Arming Timer, a Pre-Alarm Timer, and an Alarm Timer. The delays for these are all configurable in the global variable settings. The Arming timer is the delay between arming the timer, and actually being in the alarmed state. This delay allows you to arm the system, and then exit the area without setting off the alarm. The Pre-Alarm timer provides a delay to allow you to enter the area prior to setting off the alarm siren. During this time, a buzzer will be sounding to alert you (and others) that an alarm has been tripped. The Alarm timer is the period during which a siren will be sounding. I keep this short (2 minutes) since that is usually enough time to scare off an intruder but not long enough to annoy neighbors if a false alarm occurs. When each of these timers is started, an email will be sent to your chosen recipient indicating a state change.

There are 6 distinct states of operation for this system. Those states dictate the responses for all areas of code. These are shown in Table 1.

Table 1 System States

State	0	1	2	3	4	5
Dis-Armed	X	X				
Pre-Arm		X				
Armed			X	X	X	X
Pre-Alarm				X		
In Alarm					X	
End of Alarm						X

As the software loops through the “Loop” procedure, it looks for timer expirations, if they are active, or loop changes, or keypad changes. If a timer expires, the corresponding state change is made, and action taken accordingly. For example, if the system is in State 1 and the Arming Timer has expired, the system is forced into State 2. In that state, a loop opening will cause an alarm. Similarly, if in State 5, a loop opening will also cause an alarm, that is, a state change to 3. Toggling the Keypad will cause a state

change from 0 to 1 or, if in any other state, will cause the state to change to 0. To exit State 5, a keypad toggle is required.

There is no on-board real time clock, so time is obtained by querying an NTP server. Adjustment for time zone is done when setting the environment. Adjustment for Daylight Savings Time is done whenever the NTP server is queried. If your location does not observe DST, then change the procedure "IsDST" to always return false.

How to Use

Before compiling, change all the environment variables to suit your locale. These are:

1. NumberLoops
2. LoopName[NumberLoops]
3. FromEmail
4. FromEmail2
5. HashFromEmail
6. HashPassword
7. MailSubject
8. MailTo
9. MailTo2
10. IPAddress LocalIP
11. IPAddress gateway
12. IPAddress subnet
13. IPAddress dns
14. IPAddress dns2
15. SMTPserver
16. deviceName
17. ssid
18. password
19. ServerPort
20. DelayTime
21. AlarmTime

Note the restrictions about these variables highlighted in the code. Also, be advised that any emails sent by this system will always have the same subject line, but that the message will contain the detailed information. Also, emails can only be sent to one email address and the sending email address must be valid.

Once this is done, compile the program using the toolset indicated at the beginning of the software listing. After powering up, connect to the system using any browser. If you don't use 80 as your ServerPort, make sure to connect via the following syntax: 192.168.1.2:xxxx, where xxxx is your chosen port number. If you want to expose the system to the internet, make sure to forward that port on your router. The resulting web page will look something like Figure 3. You can play around with fonts and

colours in the code if so desired. Just look for the section called “Our HTML webpage contents in program memory”.

Figure 3 Example Output

Conclusion

I hope that this project has given you some ideas to work on a similar project of your own.

73

Mitch NØDIM

Appendix 1 – Bill of Material

Designator	Quantity	Description
C1	1	100uFd
D1	1	LED Red
D2	1	LED Green
D3	1	LED Yellow
D4	1	1N4007
J1,J2,J3,J4,J5,J6,J7,J8,J9	9	Screw Terminal Connector
K1,K2,K3,K4	4	Relay DPDT 12V
R1,R2,R3,R4,R5,R6,R7,R8,R9,R10,R15,R17	12	1K Resistor
R16	1	10K SIP Resistor
SP1	1	12V Piezo Buzzer
U1,U2	2	P2501 Opto-Isolator
U3	1	4N26 Opto-Isolator
U4	1	74HC165 Parallel to Serial Converter
U5	1	NodeMCUV3
U6	1	ULN2803A Darlington Array
U7	1	LM7809 9V Voltage Regulator

Appendix 2 - Sources

1. PCB Gerbers for this project: http://n0dim.com/Documents/MD5350 Alarm System.zip
2. PCB manufacturers: https://jlcpcb.com/ and https://oshpark.com/
3. NodeMCU description: https://www.nodemcu.com/index_en.html
4. Software Listing: http://n0dim.com/Documents/MD5350 Alarm System INO.zip