

hss Design

Project requirements:





1. It will detect intruders and will detect motion.
2. System has three states ... DISARMED, ARMED, ALERT
3. System allows for entry of PIN for authentication of user

Project Items:

1. LCD
2. PIR/Ultrasonic sensor
3. Matrix keypad
4. Arduino NANO
5. LEDs
6. Buzzer

Project Progress:

10/7/2023

1. Still getting used to the idea of timers and interrupts.
2. Matrix keypad sketch up and running.
3. Got the fucking 1602 LCD to work. Had an issue with backlight (pins 15, 16) because voltage was too low. Used the 3V3 pin on Nano for backlight.
4. Next steps:
 - a. Play around with other LCD example sketches 
 - b. Use I2C for LCD and free up pins 
 - c. Integrate keypad and LCD 
 - d. Draw IO representation of the system (Allocate pins properly). 

11/7/2023

1. Played with LCD example sketches. Not too difficult to program. Should be easy to integrate the matrix keypad with help from the example sketches.

12/7/2023

1. Clean LCD wiring (No jumpers).

13/7/2023

1. Watched RobotDroneWorkshop video on LCD. Learned about Nick Gammon sketch for I2C device scanning to get slave addresses. **We need <Wire.h> whenever I2C is involved.**

15/7/2023

1. LCD and Keypad can now talk to each other. LCD still uses a parallel bus (not serial I2C bus).
2. Next steps:
 - a. Implement **Password Protection**. ✓
 - b. Solder I2C Connector? ✗
 - c. Implement PIR Sensor ✓
 - d. Implement Ultrasonic Sensor ✗
 - e. Go through Arduino Forum (start from my post)
 - f. Understand **State Machines**

17/7/2023

1. **Password Protection** fully implemented. LCD continues in an infinite loop ("ENTER PIN: ") until the correct PIN is entered ("SUCCESS").

18/7/2023

1. Got the PIR sensor set up. Code looks easy, but needs more fiddling with.
2. Watched some tutorials on Arduino vs Pi. Procrastinated ?

22/7/2023

1. PIR is properly working. Time delay ("lockout") and trigger mode is important, sensitivity not so much.
2. Learned the importance of "state" in the sample PIR code. Especially important in infinite loops.
3. Active buzzer easy to use - needs only a DC voltage.
4. **Successfully integrated the buzzer and PIR.**

23/7/2023

1. DrawIO representation of HSS complete. Includes State diagram and Pin Allocation/layout.
2. Started integration of individual programs. This is the last lap!
3. I have noticed a pattern. There are 3 states in my system and each state, upon initial guessing, is stuck in its own infinite loop. So, I simply could have a function for each state and run infinite loops inside each of them. To break out of the infinite loop (and thus, the function) I could simply return a flag variable. By checking for flag variable status inside the main loop() function, I can change the state of the system (using a state variable). This is my pre-meditative guess on how to run the system.
4. LCD fucked up. Only a single white stripe on the top row. Needs rewiring. ✓

26/7/2023

1. Some doubt about how to monitor **both** for a PIR signal and a PIN entry. Is this multi-tasking? (this design has been abandoned -> see next entry)

01/8/2023

1. New design (😓) implemented due to coding logic difficulties. Now, state 2 (ARMED) will **only** monitor for a signal from the PIR. Once a signal is detected, `eternalcheck()` returns to `main()`.
2. PIR and buzzer (wiring and pin allocation) have been included in the final circuit.
3. PIR and buzzer code snippets have been integrated into final system code.
4. Wiring is getting messy. Will probably **NOT** use LEDs.

03/8/2023

1. Project substantially complete. Code uploaded to Github.