EECS 3201 Finale Project Report: Home Security System

Syed Mustafa Jamal, Student Number: 217761651
Arthur Sabadini Nascimento, Student Number: 220007175

December 4, 2024

Contents

Abstract

Our project is a home security system. Some of the features of this project are: a mode system to change the security mode, an alarm system that goes off when the door is opened and password saving/changing for the security (passcode is 4 bits long). We should be able to create a physical representation of the idea and run some trials. Here is the link to our demonstration video (https://youtu.be/GGY3ZGjNfwI).

Modules

HCSRO4.v

The design of this file was inspired by @tuansydau's project [1], specifically the sonic.v file [2]. You can see in our code, however, that we implemented the communication utilizing an ASM (Moore) design instead, since this suited the course content better. We have five states as follows.

FSM States

IDLE (Reset)

Resets instances and signals, and prepares for a measurement request.

SEND_PULSE

Makes a measurement request by driving the sensor input high for 10us.

WAIT_ECHO

Wait for sensor output (echo) to respond (echo goes high).

MEASURE

Count elapsed microseconds the sensor output has been high; the datasheet [3] specifies distance is proportional to elapsed time.

WAIT_CYCLE

Wait for next measurement; datasheet specifies 60ms delay.

FSM Inputs

1	0
echo	CLSEL

echo is an output of the sensor, and CLSEL is the 'Counter Limit Select' bit. If CLSEL = 0, then the counter limit is TRIG_DELAY; else if CLSEL = 1, the counter limit is DELAY_60_MS.

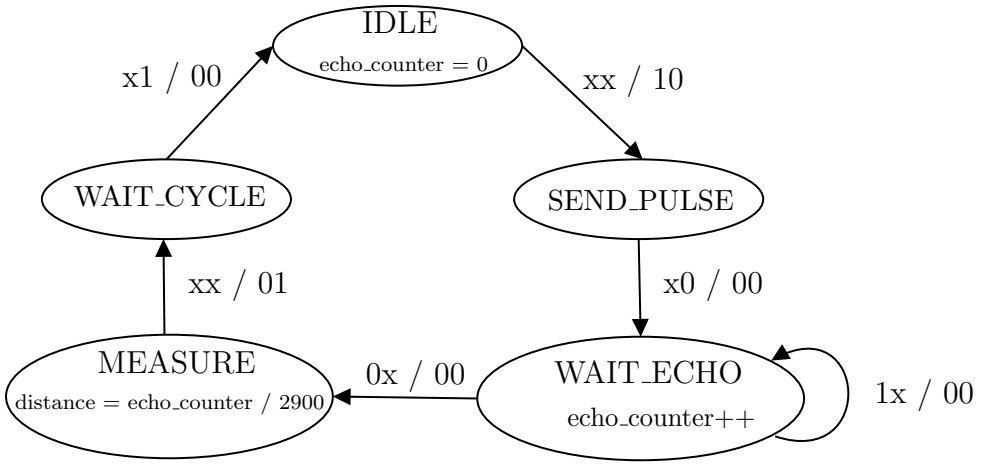
FSM Outputs

1	0
trig	LDO

trig is a control signal if the sensor, and LDO is the 'Latch Distance to Output' bit. If LDO = 1, then we latch the calculated distance to the output; else, the output is not changed.

FSM Design

If any input is marked as 'x', it means it is not important/used. Follows the FSM for this module



ModeMessage.v

This module is responsible for displaying different messages on the seven-segment display depending on the System State and Input. We implemented it using a decoder design, where the state is passed to the module as an input, and it writes the corresponding messages to the seven-segment displays.

FSM States

UNARM

Displays 'unarm' on the seven-segment display.

ARMS

Displays 'armS' on the seven-segment display.

ARMA

Displays 'armA' on the seven-segment display.

RESET

Displays 'reset' on the seven-segment display.

DISPLAY

Displays the contents of the 4-bit input on the seven-segment display (responsive to changes in input).

AlarmDrive.v

This module is used to perform a pattern on LEDS and play the buzzer for each Alarm State. We implemented it using a Moore machine with four states as follows

FSM States

ALARMOFF (Alarm Off)

Buzzer and all LEDS are off.

ALARMON (Alarm On)

Red leds and buzzer blink/buz every second, blue led is off.

AWAYSEQ

Red LEDS and buzzer are on, blue LED is off.

PLCCLD

Blue LED and buzzer turn on and off every second, red LEDS are off.

SecuritySystem.v

This module is the main file and is responsible for the whole system. Drives all operations by controlling the states and inputs/outputs of the above modules. We implemented it as a Mealy machine with four states as follows

FSM States

UNARM

Displays 'unarm' message, alarm is off.

ARMS

Displays 'armS' message. Checks periodically if the distance measured is smaller than the threshold, if so alarm state is set to ALARMON. If the correct passcode is not entered in less than 10 seconds, an alarm state is set to PLCCLD to symbolize police being called.

ARMA

Displays 'armA' message. In the first 2 seconds, the alarm state is set to AWASEQ, giving a time window for the user to leave the area. After 2 seconds, the operation is identical to the ARMS state above.

RESET

Displays 'reset' message. If the correct passcode is entered, it allows the user to reset the passcode; the new passcode is displayed.

Appendix A: Video Script/Procedure

1. Show the board and setup of the security system. Realistically, be on a wall with the sensor being placed higher up so that there is no disturbance from people going near the door.

2. Going into the reset mode and changing the password, showcasing that the passcode cannot be changed without having the old passcode. Once the correct passcode is inserted and confirmed (via the confirm button), the owner is able to change the passcode. The confirm button is pressed twice to update the passcode; we showcase this and the new passcode being updated.

3. The first security mode we show is the Unarm mode. This mode is designed for simple access to come in and out without triggering anything. It will not trigger the LEDs, buzzer or the police LED (which signals if the police are called for a potential break-in).

4. The second security mode (first armed one) is Armed S (Armed Stay). This mode is designed for when a user is at home and not planning on going outside. It would trigger if the door is opened after setting it to this mode. When triggered, the alarm would go off (red LEDs and buzzer), and if in a situation where it was not a break-in (owners returning home), the passcode can be entered within a certain amount of time to stop the alarm. This time is set to 60 seconds on the actual system; for the video demonstration, it is set to 10s. If the time is exceeded with no passcode, it is assumed to be a break-in and the police led (signalling police notified) starts blinking with the buzzer.

5. The next mode is similar to the armed stay in how it is triggered, however, it has a different purpose. Armed A (away) is meant when no one is in the house. This mode would be set, and a light buzzer with flat LEDS (not blinking) would signal everyone to leave before the system is armed. This will not trigger an alarm during the time, giving adequate time for people to leave; for the demo, it is 2 seconds, but in reality, it would be 15-30s. After the duration, it is armed similarly to Armed S and can be disabled by entering the passcode; in the demo, we show the alarm going off to its full extent to showcase the police led.

References

[1] Adam Silverman. *ajs0429/EECS3201Project*. original-date: 2020-11-09T00:32:21Z. Mar. 24, 2023. URL: https://github.com/ajs0429/EECS3201Project.

[2] Adam Silverman. *EECS3201Project/sonic.v at master · ajs0429/EECS3201Project*. GitHub. URL: https://github.com/ajs0429/EECS3201Project/blob/master/sonic.v.

[3] *Ultrasonic Ranging Module HC-SRO4*. URL: https://cdn.sparkfun.com/datasheets/Sensors/Proximity/HCSR04.pdf.

External Resources

All design files and proposal reports can be found in this GitHub directory (https://github.com/ArthurSabadini/eecs3201-home-security-system), and the demonstration video can be found here.

(https://www.youtube.com/watch?v=GGY3ZGjNfwI&feature=youtu.be)