Security System

NanoTech Final Project Report

Tyler M. and Helen P.
Rowan University
Dr. Tang

December 21, 2018

1 Introduction to Design Overview

The Security System project is a simple implication of a home security system, where the device have the capabilities of sensing motions, sounds, and send an alert to the owner through an app on their phone. There are several objectives in this project, one is being able to receive different signals from each of the sensors. From there, the device needs to be able to set off the alarm, which is the buzzer, set off the continuous flashing of the red LED, and alert the household occupier of potential intruder. The system should be able to reasonably detect if a person is in the room using auditory and visual cues. By utilizing a sounds sensor and infrared motion sensor, the Mixed Signal Processor (MSP) is able to take in the signals from the sensors and analyze this data to understand its surroundings. The data the MSP takes in is processed and used to determine whether or not the device should trip the alarm and signal the user.

1.1 Design Features

The design features are the following:

- Ability to detect motions and body heat
- Ability to detect sounds
- Able to sound the alarm (buzzer)
- Abe to set the red LED to flashing on the MSP430
- Send alerts through the ESP to an app on your phone

1.2 Featured Applications

Some applications of these features are:

- Security System
- Robots
- Industrial machinery

1.3 Design Resources

A link to the GitHub repository used for this project can be found directly bellow:

https://github.com/RU09342-F18/intro-to-embedded-final-project-nano-tech-1

1.4 Block Diagram

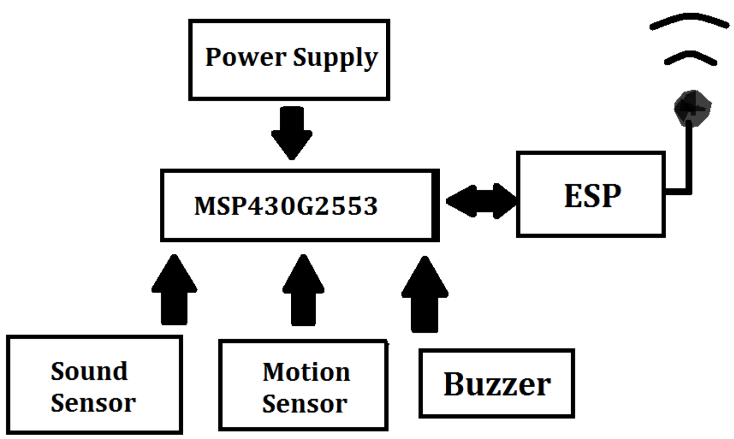

Figure 1: Overall Network of Devices

In figure 1, it represented a simple overview of the general functionality of the device. Power is supplied via the USB port to the MSP430G2553, which enables the power source pins on the MSP430 LaunchPad. In addition, the LaunchPad also provides pin connections to the motion sensor, the sound sensor, and the buzzer. Each sensor is connected to a 3.3 voltage pin, a GND pin, and to their corresponding signal feedback pins on the MSP430. Communications between the MSP and the ESP8266 module is made to send information for alerting the user of living organism detected in the presence of the device.

1.5 Board Image

The MSP430G2553 board is shown in figure 2.

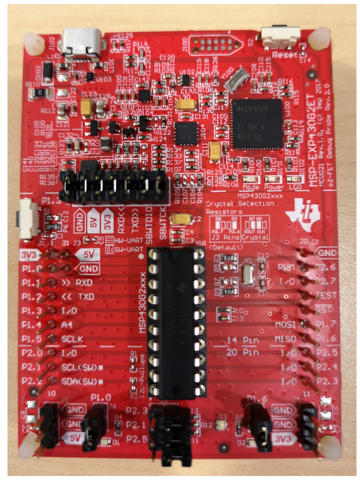

Figure 2: MSP430G2553

In figure 3, the MSP430G2553 is hooked up to each sensors through their designated pins. Both the PIR Motion Sensor and the KY-037 Microphone Sensor are connected to two 3.3V source pins and to two GND pins. The output pins from the motion and sound sensors are connected to P1.3 and P1.4 respectively. The Electromagnetic Active Buzzer is connected to GND and P2.5 to regulate frequency.

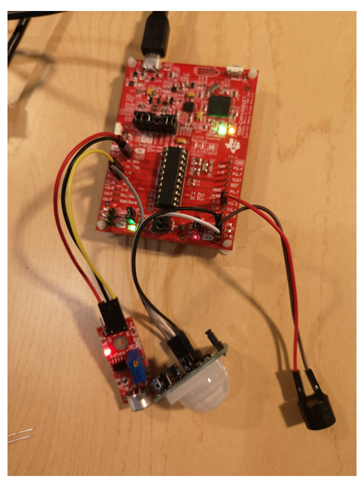

Figure 3: Overall connections of the different sensors

2 Key System Specifications

The following table details the parameters of the project. The different attributes are outlined with further details provided in the detail column.

Parameter	Specification	Details
12 bit ADC	Convert analog signals to digital values	ADC reads in an analog value (the voltage) and converts that to a digital, 12 bit number. This is used to measure the volume of noise detected by the microphone
Microphone	Measure the noise in a room	The short range microphone is used to detect how loud a noise is in a room. The microphone has two output modes. Digital, outputs high when noise passes a defined threshold, and analog, which outputs a voltage proportional to the noise in the room.
PWM	Pin P2.5 generates a PWM signal to control the Buzzer	In order for the buzzer to create a noise that is both audible and continuous, the MSP must generate a PWM signal to the P2.5 pin. The buzzer clicks together whenever it receives a high value and releases whenever it turns low. The PWM signal must be constant and frequent in order to make a loud, steady tone.
Infrared motion sensor	Compares the changes in infrared levels in order to detect motion.	The PIR Motion Sensor measures the infrared levels within its field of vision. The data is than stored on to the internal memory and will be compared to the next reading in order to detect changes. The on-board processor is used to detect this change, so no additional hardware or software is needed to make this work. If change is detected, the motion sensor outputs a high value. If no change is detected the sensor outputs low.

3 System Description

3.1 Problem

The device must serve as a security system and must be able to detect when a person is in it's presence using either auditory or visual cues. Upon receiving the feedback signals from the sensors, the device needs to be able to send a message to the household residents.

3.2 Solution

The problems that the MSP430 have to solve are divided into sections and tackled individually. The first problem is figuring how does the Passive Infrared (PIR) Motion Sensor works. Based on the datasheets [1, 2], the sensor will initially store the infrared level emitted by surrounding objects from the calibration setup of no movements into the internal memory. It would then compare the changes in infrared level when motion is detected. If the compared infrared level is different from the calibration setup, a high signal will be sent to the I/O pin P1.4 on the MSP.

The second problem is setting the KY-037 Microphone Sensor Module to detect sound. This problem is solved by adjusting the potentiometer on the sensor to a level where it can detect the sound of snapping a finger. From there, it is set to listened for volume of sound nearby and outputs analog signal proportional to sound level.

3.3 Detailed Block Diagram

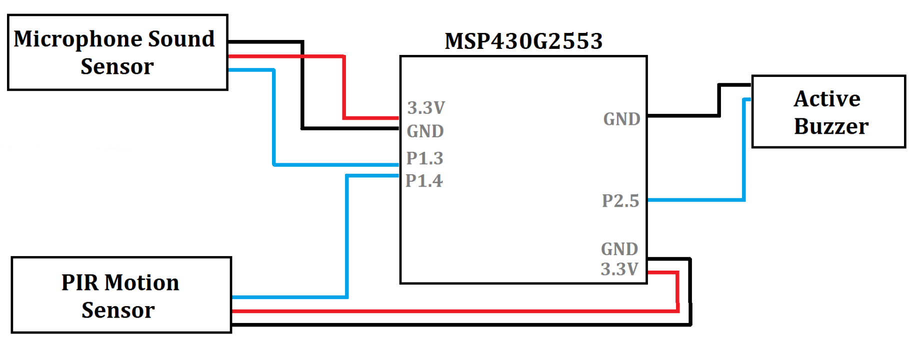

Figure 4: The pins connections to each of the sensors

3.4 Highlighted Devices

- MSP430G2553
  - Essential the ”Brains” of the project. It is the location for where code is compiled and run. It is in charge of reading in the signals from the sensors and do something with it according to how it is coded.
- MSP430 LaunchPad
  - The LaunchPad provides pin outs for numerous useful I/O pins, some of which are used in this project for connecting to ground, power source, and communicating with the different sensors and UART.
- Passive Infrared (PIR) Motion Sensor
  - The PIR Motion Sensor is able measure the infrared light radiating from objects in it’s field of range and sense movements. There are also pins to control the sensitivity of the sensor.
- KY-037 Microphone Sensor Module (High Sensitivity)
  - The Microphone Sensor is able to detect sound when directly played in front of it. It also have a potentiometer for adjusting the sensitivity level.
- Electromagnetic Active Buzzer PB-1226PEAQ
  - The buzzer is able to emits a frequency that will resembles an alarm.
- ESP8266 Module
  - The ESP8266 module is able to send information to alert the user of movements or sounds detected through an app on their phone.
- Breadboard
  - The Breadboard allows for the connection of numerous electronic components in a neat and organized fashion, without the risk of two components touching. It was used for testing in this project.

3.5 MSP430G2553

The brains of this project was the MSP430G2553 [7]. This processor was chosen for its simplicity, on-board features, and because the team already has programming experience with it. This MSP comes equip with an on board analog to digital converter (ADC) as well as two timers with 3 capture compare registers each. As for the processor itself, it is a bit lacking, being one of the slowest in the family of available processors that the team could use. It also has very limited memory space, but this was very unlikely to be a problem during development.

3.6 MSP430 LaunchPad

The MSP430 LaunchPad contains the breakout pins for the device which allows direct connection of pre-labeled pins to the breadboard. This is extremely helpful in terms of easy access and identification of each pins for connections to the sensors. The LaunchPad also provided a 3.3 volt rail, which was used to provide power to the different sensors.

3.7 Passive Infrared (PIR) Motion Sensor

The passive-infrared motion sensor is used to sense motion in the room. The sensor captures data on the infrared levels of it's surroundings [1, 2]. These data points are recorded and used to compare to the next set of data the sensor captures. The sensor's on-board processor analyzes these data points and compares them to see if there was any changes between the two. If the sensor detects any motion, it outputs a high value on the ”OUT” port.

3.8 KY-037 Microphone Sensor Module (High Sensitivity)

The KY-037 Microphone is a sensor part of an Arduino sensor kit [5]. Its appliance, in general, can be used for many different projects. The microphone has two output modes that can either be used individually, or at the same time. The first output mode sends a high signal if the microphone senses a volume above a certain threshold. This threshold can be specified by adjusting the potentiometer on the sensor. The second output mode is analog. The voltage at the analog port is proportional to the volume level detected by the microphone. The range of volume that the microphone will react to can also be adjusted using the potentiometer.

3.9 Electromagnetic Active Buzzer PB-1226PEAQ

The electromagnetic buzzer used is a very simple device. The device is a twin plate design, where each plate is connected to the two terminals of the device (plus and minus) [6]. When voltage is applied to the device, the two plates push together and make an audible ”click” sound. If the device receives alternating high low signals at a fast enough rate, it can be perceived to make a buzzing sound. The rate of the PWM can be adjusted to make the buzzer create a different sound or pitch. Higher PWM rates create a higher pitch. Pitches that are too high may be outside the range of human hearing depending on individual ability.

3.10 ESP8266 Module

This Arduino module is a self contained WiFi module [3, 4]. The Arduino device has a full contained network TCP/IP stack. The ESP is useful for network communication when a device doesn't have built in support. The device itself utilizes a typical 2.4 GHz antenna to connect to the routers. Therefore, it is unable to connect to the newer, faster 5 GHz band. The modules on-board processor and memory allows it to be self contained and it can interface directly with internet connected devices.

3.11 Breadboard

In this project, the breadboard was used to test whether the sensors are sending the output signals to their respective pins. Each of the sensor is tested by using an LED. The code is configured to fired an interrupt each time an output signal is received from the sensor. For each interrupt, it is set to toggle the LED and than clears the interrupt flag for the next interrupt.

4 SYSTEM DESIGN THEORY

The major design components of the project are the uses of the sensors. This allows the device to have eyes into the real world and know what is going on around it. The timers were used extensively to synchronize events on the processor. The last design feature of the project is the use of UART, which was used to communicate with the ESP8266 so that the MSP could connect to the internet.

4.1 Sensors

The sensors in this project allows the MSP to detect and sense things in the real world. For the purposes of this project, this is particularly useful, because this allows the MSP to ”know” when a person is in the room. The Microphone Sound sensor used, can listen for auditory cues and then send signals based on what it hears. The signals it sends are sent to the ADC on the MSP in order to gauge how loud the sound was, and whether or not this sound is worth tripping the alarm.

The second sensor used was the Passive Infrared Motion sensor. As detailed in an earlier section, this sensor allows the MSP to know when there is any motion within its field of view of the sensor. The sensor sends a high value when it detects motion and a low value when nothing is detected. The sensor is wired up to Port 1.4 on the MSP. This port is configured to fire an interrupt when ever it receives a high value. The interrupt would then call a function to sound the alarm for when motion has been detected.

4.2 Timers

Timers were used extensively in this project. One such use of timers is to PWM the buzzer. The buzzer was PWM’d to make a continuous buzzing noise loud enough for humans to hear, but also within the auditory range for the average person to hear. Another timer is being used for polling the ADC for changes in its memory banks. This timer polls the ADC every 100ms, but could be setup to poll more often. On each interval, the MSP checks the memory bank to see if the value in the ADC exceeds a predetermined value. The value is based on how loud the average noise level in the room is, as to not trip the alarm from the ambient noise in the room.

A second set of timers are being used to keep track of how long the alarm has been going off. This timer is run by a separate clock, so that it can be slower and the alarm can be on longer. Although this timer is constantly running, the value of the internal clock is set to 0 when ever the alarm is set off. When the timer interrupt fires, the alarm is shut off, and because the timer value is set to 0 every time, the alarm goes off as well. The alarm will always sound for the same amount of time. The last timer used in the design is for the recording LED. Many cameras and security devices have a red flashing LED to indicate that they are recording and operating properly. The timer used here is similar, and will PWM a red LED that indicates that the security system is armed and working. The LED is set to only flash during normal operation, and will turn off if the security system is disarmed.

4.3 UART

For this project, UART is used exclusively for communicating with the ESP8266. The ESP is capable of communicating over UART, so the UART line on the MSP was used to handle information exchange between the two components. The main task that the ESP handles is network communications. In order for the ESP to do this, it must know what servers to connect to, what to do when it gets there, and what information to exchange. It also have to grab information from the network and return it to the user. In order to instruct the ESP properly on what tasks to carry out, UART is used. UART communication is carried out with very simple character based commands. This is made possible because of the custom code on the ESP that has been setup to only interface with the one MQTT server that this project deals with. The table below details the 4 commands used for this project.

Command	Task	Syntax
$	Subscribe	$Topic\n
#	Subscribe	#Topic Payload\n
!	Received Message	!Topic: Message\n
∼	Unsubscribe	∼Topic\n

Beyond just the simplicity of only having to use 4 basic commands to interface with the MQTT server, it is also a better, faster design. With less characters to send, the MSP has to spend less time sending data over UART in order to make the ESP do something. It is also a simpler design from a development perspective. Simpler code that is easier to read, and takes up less space on the already very limited resources that the MSP is equipped with.

5 Getting Started/How to use the device

5.1 Hardware Overview

The full bill of materials for this project can be found in section 9.1. The main components and hardware used were the MSP430G2553, which handles all the processing. The MSP keeps track of the current device state and interfaces with all the sensors. The two sensors used are the Infrared Motion sensor and a Sound sensor. The infrared motion sensor is a simple 3.3 volt infrared sensor that sends a high bit when it detects motion. The sound sensor is a useful and robust 5 or 3.3 v sensor, that can output a digital or analog signal based on the volume of sound in the room. For network communications, the ESP8266 was used. This device comes equip with a 2.4 GHz band antenna, allowing it to connect to most Wifi networks. All these devices combined will allow the security system to do its task without over complicating the design.

5.2 Communication

User interaction is strictly restricted to the MQTT server interface, while user communication is mostly restricted to that same interface. Although the device has very few input options, it does have some that require a user. Users can request the security system to either arm or disarm itself using the MQTT server interface. Beyond this one simple task, there are no other user inputs. The device itself can, however, communicate with the user. The device, when the alarm is tripped, pushes an update to the MQTT server notifying it that the alarm is going off. The User can then use a compliant internet device to check the status on the server. The security system also tells the MQTT server which sensor tripped the alarm. For example, if the security system hears someone or something in the room, but doesn't see anything, the alarm will be tripped and the system will still send an update to the MQTT server saying the alarm is going off and it was set off by the microphone, not the infrared motion sensor.

6 Getting Started Software/Firmware

6.1 Software

There are two pieces of software required for this project. The first is Code Composer Studio, which is available for the three major operating systems and is free to download for all users. It also comes pre-packaged with all the resources needed to compile the board used in this project. The second piece is the MQTT, which is necessary to establish wireless communications from the board to the user. It is used to compile the ESP8266 Module to send alerts to the "MQTT DASH" app on the user's phone.

6.2 Firmware

In order for the device to run properly, an updated copy of the MSP430G2553 firmware must be installed to the board. To update the firmware, simply load up Code Composer Studio and compile the project to the board. Code Composer Studio will automatically detect that the boards firmware is out of date and prompt you to update it. Never unplug the device while updating firmware, as this could permanently damage the device.

7 Test Setup

In order to test whether or not the sensors are sending the output signals to the MSP430, an LED was used for the interrupts and the ADC function for the Motion Sensor and the Sound Sensor. The interrupt is setup so that each time a signal is received from the Motion Sensor output to pin P1.4, a interrupt would fire and toggle the green LED. Then clears the interrupt flag for the next interrupt. Similarly, the LED is used to test the ADC function of constantly polling the sound sensor every 1000 ms on P1.3 pin, in order to check for sounds that is above the threshold, which in event will toggle the LED.

8 Results and Conclusions

The conclusions of this project showed that the system, while difficult to setup and execute is very possible. The main functions of the system were successfully setup and tested. The microphone sensor was used and tested successfully. The buzzer and LED were properly PWM'd to make a loud sound and visible blink. These components were all successfully hooked up and by using the logic from the code that was written, the components were able to interact to form a basic security system. The ESP was also used and seemed to have been working correctly. However, there were complications with the network itself, not the ESP. The network that the ESP was attempting to connect to seemed to have been having problems on the days that testing was done. The only component that proved difficult was the infrared motion sensor. The tech office where this sensor was pulled from did not know what model it was and had no additional information on it. A datasheet could not be found on this device and all the information that the team currently have on the device is from sourcing similar products and testing the component in the real world.

References

[1] Parallax, "PIR Sensor," [Online], Available:
`http://eecs.oregonstate.edu/sites/eecs.oregonstate.edu/files/tekbots/docs/pir/pirsensor-v1.2.pdf`

[2] "HC-SR501 PIR MOTION DETECTOR," [Online], Available:
`https://www.mpja.com/download/31227sc.pdf`

[3] "ESP8266," [Online], Available: `https://nurdspace.nl/ESP8266`

[4] "ESP8266 Module (WRL-13678)," [Online], Available:
`https://cdn.sparkfun.com/datasheets/Wireless/WiFi/ESP8266ModuleV1.pdf`

[5] "Arduino KY-037 Sensitive Microphone Sound Sensor Module," [Online], Available:
`https://tkkrlab.nl/wiki/Arduino_KY-037_Sensitive_microphone_sensor_module`

[6] "PB-1226PEAQ," [Online], Available:
`https://www.mspindy.com/Specifications/PB-1226PEAQ.pdf`

[7] "Mixed Signal Microcontroller," [Online], Available:
`http://www.ti.com/lit/ds/symlink/msp430g2553.pdf`

9 Appendix

9.1 Bill of Materials

Item	Name	Amount	Cost
1	MSP-EXP430G2553 Micro-controller	1	$9.99
2	Passive Infrared (PIR) Motion Sensor	1	$10.00
3	KY-037 Microphone Sensor Module (High Sensitivity)	1	$1.29
4	Electromagnetic Active Buzzer PB-1226PEAQ	1	$0.45
5	ESP-01 Serial Wifi Module (ESP8266 Module)	1	$6.95
6	Green LED	1	$0.18
7	Solderless Breadboard	1	$3.99