– ECE 09.321: Final Project –

Santa's Fire Detector

Nick Klein, Kieran O'Connor,
Shane Price
Rowan University
Version 1

December 21, 2018

1 Design Overview

During the Santa Fire Detector project a series of sensors a buzzer and an ESP01 chip are able to warn Santa, or others, of a fire in a building. It was discovered that Santa would often approach a chimney and descend, at the bottom of his descent he would be burned by the fire in the fire place; the Santa Fire Detector is a way for Santa to know if a fire is in the fire place as he approaches the chimney. Once Santa approaches the chimney the system will evaluate if a fire is present and warn Santa via an audible alarm and a message sent to his phone.

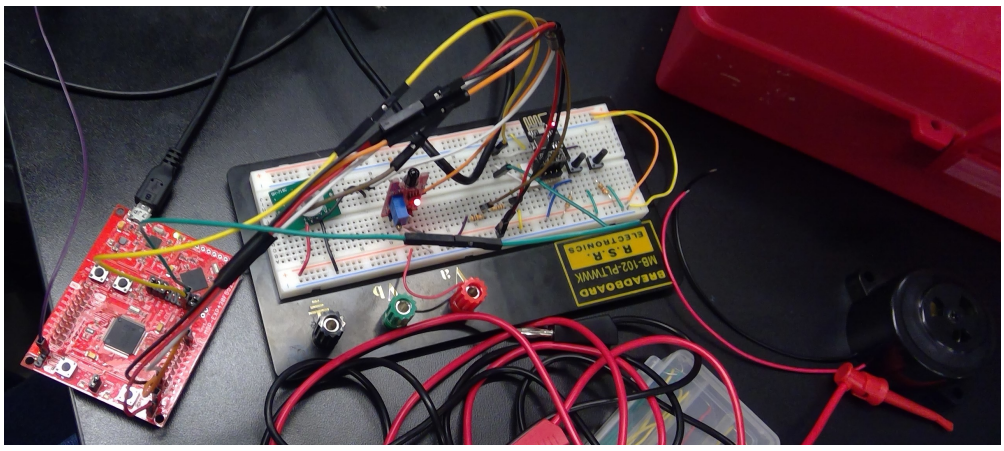

Figure 1: Santa Fire Detector

1.1 Design Features

The design features are as follows:

- Flame Detector
- Motion Sensor
- 100dB Buzzer
- ESP01

1.2 Featured Applications

The Santa Fire Detector can be used for all applications listed below:

- Added feature in current alarms
- Add on to a system utilizing sensors and actuators
- Santa on Christmas Eve
- Fire Departments
- Residential Buildings
- Commercial Buildings

1.3 Design Resources

To view the code used in the Santa Fire Detector project, such as the code for the MSP430F5529 and the ESP01 follow this Github link.
`https://github.com/RU09342-F18/intro-to-embedded-final-project-russells-muscle`

1.4 Block Diagram

The below block diagram depicts how the hardware functions as a singular system. This will be discussed in greater detail in the Detailed Block Diagram Section.

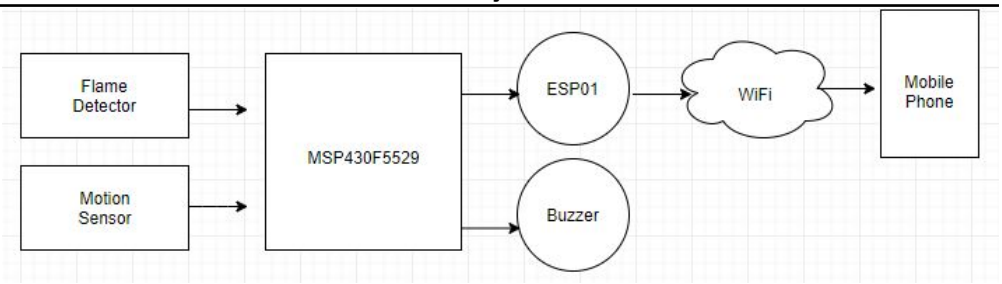

Figure 2: Santa Fire Detector Block Diagram

1.5 Board Image

Below is a close view circuit design of the Santa Fire Detector. This shows how all the hardware is assembled and where the individual components connect to the MSP430F5529.

If more time was available the the system would have been made into a PCB for more convenient usage. Due to time restrictions the system had to remain on a breadboard, neatly wired and connected.

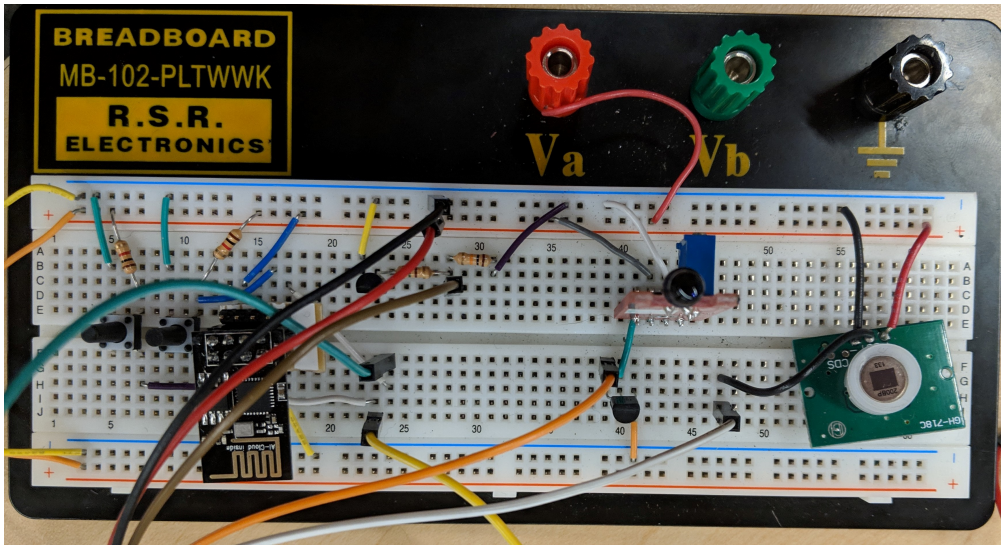

Figure 3: Close View Circuit Design

2 Key System Specifications

PARAMETER	SPECIFICATIONS	DETAILS
Flame Detector	3.3V, Sends a high signal	Operates between 3.3V & 5V. Detects light between 760nm and 1100nm. Sends a high signal needing a MOSFET to send a low signal.
Motion Sensor	3.3V, Sends a low signal	Can operate up to 20V & 7m from sensor. Sends a low signal to be directly connected to MSP.
Buzzer	3.3V, Operating at 80% power	Operates up to 12V being sent a 80% PWM from the MSP through the MOSFET
ESP01	3.3V	Operates between 3.0 and 3.6V transmits at 2.4GHz
2N7000 MOSFET	Powered from PWM & High Signal	Operates up to 60V drives signal low from flame detector. Receives PWM from the gate and sends to the drain (Buzzer).

3 System Description

The problem that is solved by the Santa Fire Detector is a common issue for Santa when he approaches a chimney and is unsure if there is a fire down below. Now Santa can approach the chimney and be told whether it is safe or not for him to descend and leave presents or if he should sneak in through the front door. Using a few basic sensors Santa can now prevent major burns and pain to help keep him in the Christmas spirit while delivering presents to the families.

3.1 Detailed Block Diagram

The block diagram below shows how the sensors connect to the micro-controller then how the micro-controller communicates with the actuator (buzzer alarm) and the ESP01 device.

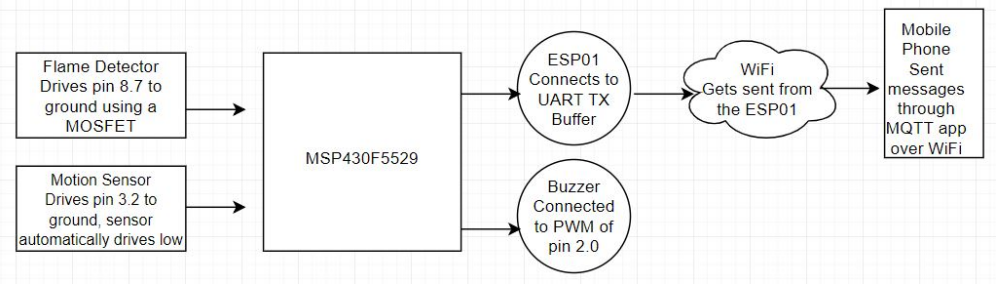

Figure 4: Santa Fire Detector Detailed Block Diagram

3.2 Highlighted Devices

- MSP430F5529
- ESP-01
- Flame Detector
- Motion Sensor
- Buzzer

3.3 MSP430F5529

The microcontroller is the main processing unit of Santa's Fire Detector, using the input of the sensors to change the state of the actuators in real time. The microcontroller detects input from sensors on a low signal.

3.4 ESP-01

The ESP-01, using the ESP8266 microchip, is connected to the microcontroller via serial UART connection sends commuications from Santa's Fire Detector to the user's phone app in the case of a fire. The TX (P4.5) and RX (P4.4) pins of the microcontroller connect to Pin 2 and Pin 7 on the ESP01 respectively. Pin 1 is tied to ground and Pins 4, 5, and 8 are tied to a 3.3V VCC. The other pins remain unconnected. When flashing the ESP, Pins 4 and 5 must be pulled to GND. For this reason, buttons were placed so that Pin 4 and 5 were connected to VCC when the buttons were not pressed but when they were, voltage drops across the button and a 1000 resistor to ground instead of through the ESP.

3.5 Arduino Flame Detector

The flame detector senses wavelengths of light from 760nm to 1100nm which covers the visible spectrum of a flame, sending a signal to the microprocessor when those wavelengths are present. Pin 2 is connected to GND, Pin 3 is connected to 3.3V VCC, and Pin 4 connects to the gate of an N-MOS which allows Pin 8.2 on the micro-controller to be pulled low. Pin 8.2 on the micro-controller is connected to the drain of the N-MOS and the source is connected to GND.

3.6 Mini PIR Motion Detector

Detects motion within a 7 meter range, sending a signal to the microprocessor while objects remain in motion. Pin 1 is connected to 3.3V VCC, Pin 3 is connected to GND, and Pin 2 connects to Pin 3.7 on the micro-controller.

3.7 Buzzer Alarm

In the event that there is a fire and motion near Santa's Fire Detector, the micro-controller sends a PWM signal with an 80% duty cycle to the alarm which makes an audible alarm sound. Pin 1 connects to 3.3V VCC, Pin 2 connects to the Drain of an N-MOS. The Source connects to GND, and the gate connects to a 150 resistor which in turn connects to a 10k resistor and Pin 2.0 on the micro-contoller. The 10k connects to GND on its other side. This voltage splitter keeps the gate pulled low when Pin 2.0 is not pushing a high signal.

4 SYSTEM DESIGN THEORY

This system had two important components that allowed our sensor and actuator to have proper functionality. The MOSFET driver and low side switch both utilize a MOSFET for different reasons as explained below.

4.1 MOSFET Driver

The MOSFET driver is a necessary component to the system for the fact that a micro-controller would have a burnt pin without it. The MOSFET driver was needed for the flame detector sensor, this sensor sent a high signal (3.3V) to the pin; since this was an input it would need to drive low (ground) to function properly.

A MOSFET driver is not necessary for the motion sensor as the digital output of the motion sensor was already sending low. This was determined by utilizing the oscilloscope to view how each of these components functioned and how they needed to be implemented into the system.

The MOSFET used was a signal MOSFET, 2N7000. This had source connected

to ground, and the drain was connected to the micro-controller while the gate was fed the digital output from the flame detector sensor.

4.2 Low Side Switch

The low side switch also utilized a 2N7000 signal MOSFET. This was used for the PWM of the buzzer alarm. The MOSFET would pull the buzzer low when the buzzer PWM was triggered, otherwise the voltage divider would pull low to prevent the buzzer from continuously being powered.

PWM from the micro-controller was sent to the voltage divider with the voltage divider connected to ground and the buzzer was on the drain of the MOSFET. Having the voltage divider grounded is how it was constantly pulled low until it was time to pull low from the drain (completing the circuit delivering power to the buzzer).

5 Getting Started/How to use the device

In order for this to function properly the MSP430F5529 must be connected to power with the sensor wire for the of the motion sensor connected to P3.2, wire to the flame detector to P8.7, wire for PWM connected to P2.0, and the wire to receive the message from UART to the TX pin on the MSP430F5529. Once fully connected and the MSP is powered on via USB, the system will be ready to be used. Simply light a match/lighter close to the black "LED" while the motion detector can sense motion of the lighter approaching. At this time an audible buzzer alarm will sound and send a message of "Fire" to a mobile phone connected to the ESP01 module as explained in the Communicating with the Device section.

6 Getting Started Software/Firmware

The software used to implement Santa's Fire Detector includes Code Composer Studio and the Arduino IDE as integrated development environments. In the case of using the code posted to github, they will only be used to flash the MSP430 and the ESP.

6.1 MSP430 using main.c in CCS

main.c must be run in a Code Composer Studio project that has the MSP-EXP430F5529 board selected. The board must be connected to the computer using a microusb cable. It then must be flashed using the main.c file.

6.2 ESP-01 using esp8266code.ino in Arduino IDE

Some setup is required to make the ESP-01 function using the Arduino IDE. To begin, you need to update the board manager with a custom URL. Open up Arduino, then go

to the Preferences (File >Preferences). Then, towards the bottom of the window, copy "http://arduino.esp8266.com/stable/package_esp8266com_index.json" without quotes into the Additional Board Manager URLs text box. Hit OK. Then navigate to the Board Manager by going to Tools >Boards >Boards Manager. There should be a couple new entries in addition to the standard Arduino boards. Look for esp8266. Click on that entry, then select Install. With the Board addon installed, all thats left to do is select ESP8266 Thing from the Tools >Boards menu.Then select your board's port number under the Tools >Port menu. This can be found by looking in device manager and checking under COM ports. Press the buttons that pull pin 4 and 5 on the ESP to GND and flash the device.

6.3 Communicating with the Device

There are a few different ways communication is accomplished over the device the first is with the use of UART. UART communicates from the MSP430F5529 to the ESP-01 using the TXD pin on the microcontroller sending to the ESP-01 RXD pin. When the program is originally open the microcontroller will send a message "$Fire" to the ESP-01 character by character subscribing the output of the system to the topic "Fire". After this is completed and the Flame detector detects a fire the microcontroller will send "#Fire Fire" to the ESP-01 transmitting the message "Fire" to the clients subscribed to the topic Fire connected to the wireless Internet.

To communicate over the wireless Internet an app called MQTT Dash is used. The IP Address has to correspond to the Internet used and to receive the message a text a GPIO operation must be created with the "Topic (sub)" set to "Fire". This should let you receive the message from the system.

7 Test Setup

There are a few measures of testing that were completed throughout the completion of this project. The most helpful test is done with the use of Realterm a terminal program used to test the UART configuration to receive the correct output from the MSP430F5529 to the ESP-01. The UART is tested because the ESP-01 is not receiving the data from the microcontroller. This was fixed by connecting the RX and TX pins to a laptop and testing with Realterm. After testing the proper "$Fire" and "#Fire Fire" were received properly after changing the baud rate of the UART and a while statement that lets the TX buffer clear before sending another character through UART.

8 Design Files

8.1 Schematics

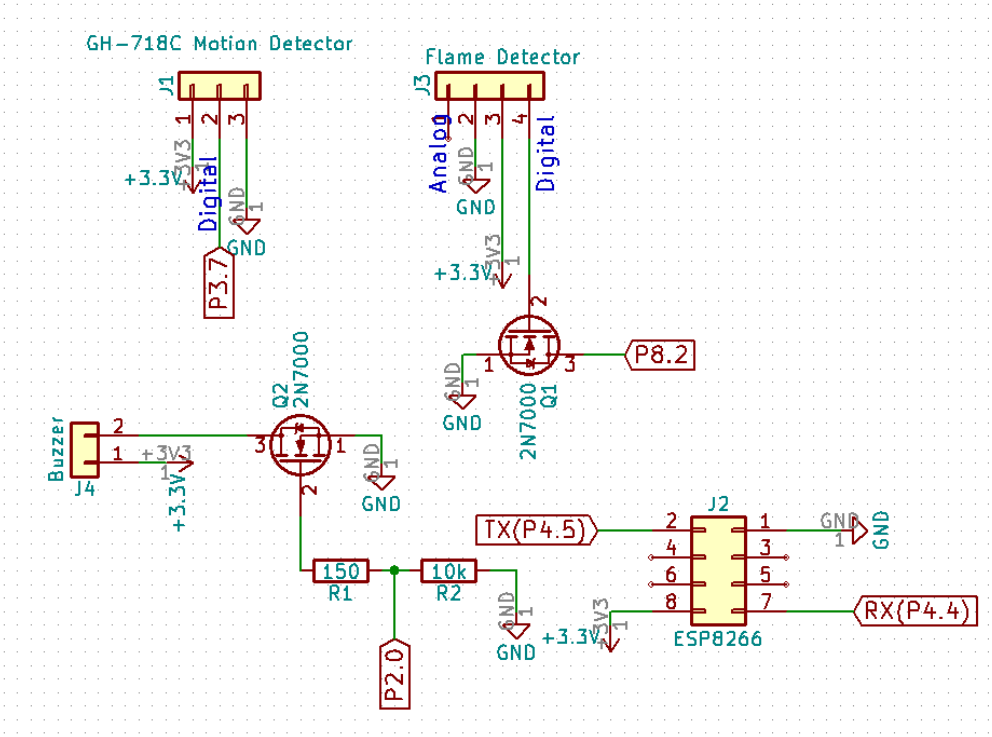

Figure 5: Santa Fire Detector Circuit

8.2 Bill of Materials

- 1 150 Ohm Resistor
- 1 10k Ohm Resistor
- 2 1000 ohm Resistors
- 1 MSP430F5529 Launch Board
- 2 2N7000 MOSFET
- 2 Buttons
- 1 Arduino Flame Detector

- 1 Mini PIR Motion Sensor
- 1 ESP8266
- 1 Breadboard

9 References

- Family User Guide
  http://www.ti.com/lit/ug/slau208q/slau208q.pdf
- MSP430F5529 Datasheet
  http://www.ti.com/lit/ug/slau533d/slau533d.pdf
- 2N7000 Mosfet Datasheet
  https://www.onsemi.com/pub/Collateral/2N7000-D.PDF
- TI Resource Explorer UART 115200 Echo Example Code
  http://dev.ti.com/tirex/#/Device/MSP430F5529/?link=Software%2FMSP430Ware%2FDevices%2FMSP430F5529%2FPeripheral%20Examples%2FRegister%20Level%2FMSP430F55xx_uscia0_uart_01.c
- ESP01 Datasheet
  https://cdn.sparkfun.com/datasheets/Wireless/WiFi/ESP8266ModuleV1.pdf
- Flame Detector Datasheet
  http://rogerbit.com/wprb/wp-content/uploads/2018/01/Flame-sensor-arduino.pdf
- Mini PIR Motion Sensor
  https://www.e-gizmo.net/oc/kits%20documents/mini%20PIR/mini%20PIR.pdf
- Example Code from Milestone II
  https://github.com/RU09342-F18/introtoembedded-f18-milestone2-russells-muscles