**Lab 4: EAGLE Design**

Eric Pires

John Lamanuzzi

Richard Harrison

ECE 388

Team 3

Lab Date: October 12, 2016

Due Date: November 9, 2016



**Table of Contents**

[Introduction 2](#_Toc466396603)

[Methods and Testing Procedures 3](#_Toc466396604)

[Prototype Finalization 3](#_Toc466396605)

[Eagle Schematic Design 5](#_Toc466396606)

[Eagle Layout Design 8](#_Toc466396607)

[Results 9](#_Toc466396608)

[Prototype Finalization 9](#_Toc466396609)

[Eagle Schematic Design 11](#_Toc466396610)

[Eagle Layout Design 11](#_Toc466396611)

[Discussion 12](#_Toc466396612)

[Prototype Enhancements 12](#_Toc466396613)

[Button Panel 12](#_Toc466396614)

[Design Process 13](#_Toc466396615)

[Conclusion 14](#_Toc466396616)

[Reflection 14](#_Toc466396617)

[References 15](#_Toc466396618)

[Appendix 16](#_Toc466396619)

Abstract

The goal of this laboratory was to fabricate a printed circuit board design using EAGLE schematic and layout editors. The previous breadboard prototype was used as a model for building the design schematic along with the Metro Mini 328 board. The board was designed in the layout editor and passed through several iterations before finalizing and processing the design. The board passed several iterations before the required gerber files were generated and sent for evaluation.

# Introduction

The objective of this lab was to use EAGLE for designing the climate control and room occupancy system onto a PCB. Before fabrication, the prototype was finalized and modeled onto the schematic editor. The following components were tested prior to the fabrication process:

* SRD-05VDC-SL-C Relay
* 2N3904 Transistor
* IRF510 MOSFET
* IR Obstacle Avoidance Sensor

Various resources were used to design the schematic with the provided component requirements. The components were then arranged and carefully traced. Error tests were ran to ensure that the design met the given fabrication house rules. Once these requirements were met, the CAM processor was executed to create the necessary Gerber files to be sent to the fabrication house.

# Methods and Testing Procedures

## Prototype Finalization

Prior to the circuit board design, the prototype was finalized with additional testing procedures. The first procedure was using the relay to switch a high voltage signal. The purpose was to ensure that the larger loads of the design would be sufficiently powered with the given relay. To test this, a desk lamp was used as the switching load of the relay. Using a wall outlet switch provided by Professor Viall, the desk lamp was connected to the outlet on the switching mechanism. This mechanism can be found in Figure 1 of the Appendix. The two exposed wires of the mechanism were then fed into the screw terminals of the relay board (COM and NO). The relay coil was then provided a 5V signal and grounded to turn the relay on. The testing matrix can be found below:

|  |  |  |  |
| --- | --- | --- | --- |
| Test Number | Test Name | Test Description | Procedure Description |
| 1 | Relay Switching | Does the relay switch when the coil is powered? | Supply the relay with a 5V signal from the microcontroller. Observe the relay to ensure that the coil is activated. This should cause the on-board led to turn on. |
| 2 | Light Load Test | Will the relay power a 40W light bulb? | Attach the desk lamp to the outlet switch mechanism. One wire will connect to the COM port while the other to the NO port. Repeat procedure #1 and observe the state of the light bulb. Calculate the maximum possible current draw. |
| 3 | Heater Load Test | Will the relay power a 170-250W space heater? | Repeat procedure #2. Observe the state of the heater. Calculate the maximum possible current draw for the low and high settings. |

The next test was to power the cooling fans with a higher voltage. To control the 12V load using logic signals, an N-channel BJT and MOSFET were chosen to act as a switch. To accomplish this, a 9V battery was used to simulate a high voltage load. The ground of the 12V fan was tied to the collector/drain pin of the transistors. The base/gate pins were given the logic signal while the emitter/source pins were grounded. The testing matrix can be found below:

|  |  |  |  |
| --- | --- | --- | --- |
| Test Number | Test Name | Testing Question | Procedure Description |
| 1 | Battery Check | Does the battery for this test still have a charge? | Measure the voltage of the batter using the Etekcity MSR-R500 Digital Multimeter. |
| 2 | BJTand MOSFET Load Test | Will the BJT handle the 12VDC Fan load? | Connect the load to the collector pin (or drain), the logic signal to the base (or gate), and ground to the emitter (or source). Power the base and observe the state of the fan. |

The final step was to test a new type of IR sensor. The IR obstacle avoidance sensor was acquired and tested in a counting application to determine the accuracy and speed of the sensor. The sensor contained only three pin connections: power, ground, and out. A simple counting code loop was written to print an incremented value every time the sensor triggered. The sensor was triggered every second ten times and compared to the given output on the serial console. The code can be found in Figure 1 of the Appendix. A table of the testing procedures can be found below.

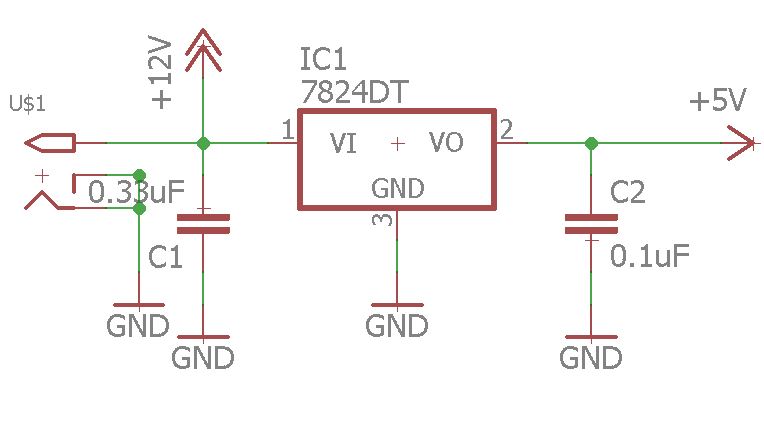
|  |  |  |  |
| --- | --- | --- | --- |
| Test Number | Test Name | Test Description | Procedure Description |
| 1 | Trigger Response | Does the sensor react accordingly and increment the given counter? | Compile and run a basic loop that increments an integer variable whenever triggered. |
| 2 | Trigger Speed | How fast can the sensor pick up a rising edge? | Trigger the IR sensor in rapid succession ten times in a row. Calculate the accuracy of the sensor by dividing ten by the number displayed on the serial console. |

## Eagle Schematic Design

After the prototyping phase was complete, the design phase began. Before using EAGLE schematic editor, research was done on how to work in the software. Design schematics for the Metro Mini 328, the controller used in the prototype phase, and the Arduino UNO were carefully examined to learn how to model an atmega328p onto a PCB. The UNO’s barrel jack had provided insight on how to power the board. The determined input signal for the board was decided on 12V. This signal would be enough to power the case fans and the microcontroller. To utilize logic signals, a voltage regulator was added to the design. Since the system will require a timer, a 16MHz crystal was also added, based on the Metro Mini design.

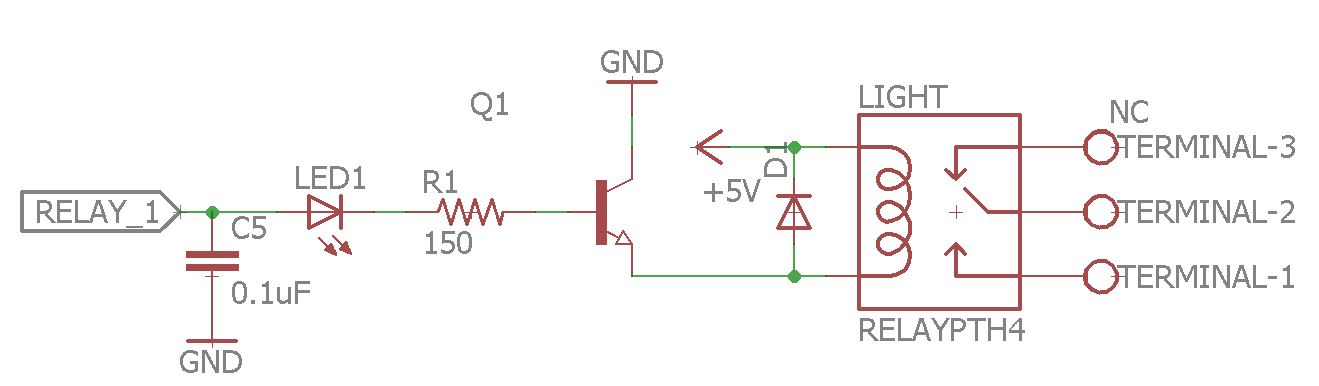
Following the research behind EAGLE, a new project was created. The design rule checks and CAM processing files in the class drive were imported to EAGLE. Various libraries were downloaded from the element14 community blog. With these files loaded into the software package, designing soon began.

The first circuit to be designed was the power supply. To accomplish this, a power jack and voltage regulator was added to the schematic. The jack supplies 12V to the input of the voltage regulator. The output supplied the rest of the board with a 5V signal. Both the input and output pins were tied down to ground with a capacitor. This would act as a high frequency filter to ensure that the noise levels of the input signal are minimal. The capacitor values were chosen based on the datasheet recommendation. The schematic layout can be viewed below.



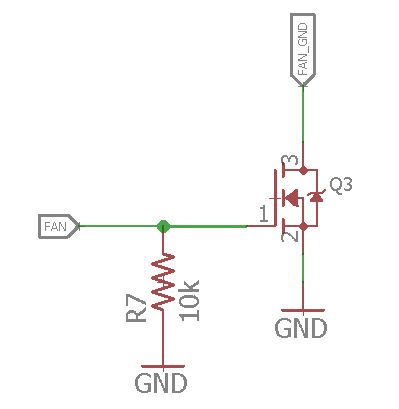
The microcontroller was then placed with labels on each used pin. These labels would then be routed for the given sub designs. The crystal was also added to pins 7 and 8.

The next circuit was the relay design. The schematic found for the given relay board was modeled onto the editor. A grounded capacitor was added to ensure that any potential high noise frequency would be filtered. The schematic layout can be viewed below.



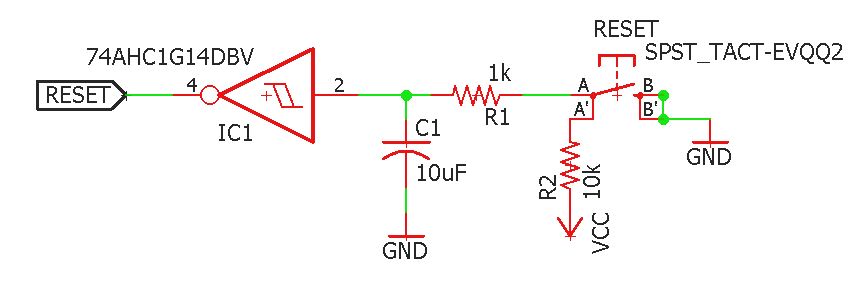
Note that the relay signal is switched using a transistor. When activated, the coil is grounded and the relay is activated. The diode that ties the coil pins ensures that current flows back to the coil in the event of a potential power surge that could damage the microcontroller.

The next design was the fan control. The fan connector was sent a 12V signal and its ground was tied to the source pin of a MOSFET. The MOSFET’s base was then fed to an I/O pin. This was also tied to a pull down resistor to prevent any floating signals from activating the fan. This would act as a high load switch, turning the fans on and off based on the logic signal. Note that multiple fans will be wired in parallel and that the power requirements were accounted for. The schematic for this design can be found below.



The final schematic step was to add the appropriate headers for the remaining sensors and displays. The DHT-22, LCD, ISP, IR sensors, and user interface board were all provided with headers on the schematic.

In addition to the main board, the user interface board was designed. This consisted of three buttons: Reset, up, and down. These buttons will allow the user to control the set temperature displayed on the LED. It will also allow the user to reset the system to a default state. The button schematic was designed to produce a clear input signal. This is done with a RC pulldown network. This design choice is explained in the Discussion. The schematic for this board can be found below.



## Eagle Layout Design

A table of steps performed during the layout design can be found below:

|  |  |
| --- | --- |
| Step Description | Procedure Description |
| Mount the Components | Use the *Move* tool to drag the components onto the board space. Pair up all relevant components based on the schematic layout. Drag the components around to ensure that they contain clear routes toward one another. |
| Route the Traces | Use the *Route* tool to connect the airwires to their designated paths. Frequently hit the *Ratsnest* tool to refresh the airwire paths. Avoid right angle bends. |
| Adjust Board Space | Shrink any unused board space and move components around to fill gaps. Use *ripup* to redo the traces that have been altered from moving components. |
| Run Error Checking | Use the DRC tool to ensure that all fabrication specification have been met |
| Fix Errors | Observe the error list and adjust the spacing of components and trace bends. Ensure that all components contain enough room for traces to pass through. |
| Export and Observe Gerbers | Run *Cam Processing* and zip gerber files to be sent to fabrication house. Observe the generated gerber files using gerbv. |

After running the provided error check in the schematic editor, the board file was generated and modeled in the layout editor. The ratsnest tool was frequently used to calculate the best possible routes for traces. The traces width was configured to be 10mm and bend no more than 45 degrees. Vias were also placed to route overlapping traces to the bottom of the board. When all data and power traces were connected, the polygon tool was used to generate a ground plane on both ends of the board. The ground plane would make it easier to route the ground connections, greatly reducing the number of traces. The ratsnest tool was run to assure that no airwires were left on the design. The design rules check provided in the class drive was executed to ensure that the board met the fabrication house specifications. This process was repeated to assure that the design was ready to be exported.

When the DRC detected no errors, the design was then proceeded with the given cam file from the class drive. The generated gerber files were then inspected using gerbv software. The traces were once more examined and adjusted to provide clear and short paths throughout the design. The gerbers were then zipped and sent to Professor Viall for evaluation

# Results

## Prototype Finalization

The final components proved to have operated as expected. The relay was able to handle the given loads without any complications. The transistor tests however proved more challenging. The first test resulted in a component malfunction. To determine the cause of this, the datasheet was pulled for the transistor [1]. The problem was concluded to have occurred due to the excess of current being drawn from the transistor. The maximum rate was about 0.2A, while the load in test drew 0.25A. After realizing this issue, a new transistor was selected and performed to expectations. However, a new problem arose. The fan would still be on despite the gate voltage being disconnected. To counter this, a pull down resistor was added to the design. This adjustment was accounted for in the circuit design. The IR sensor proved to provide much faster counter readings then the previous sensor. However, the sensor cannot distinguish between a human presence entering and a random object. All measured results can be found in the text matrixes below:

Relay Control

|  |  |  |  |
| --- | --- | --- | --- |
| Test Name | Test Question | Pass/Fail | Additional Notes |
| Relay Switching | Does the relay switch when the coil is powered? | Pass | When the coil is powered, the on board led illuminated and a clicking sound was made by the device, signifying that the magnet switched positions when powered. |
| Light Load Test | Will the relay power a 40W light bulb? | Pass | When the relay was powered, the lightbulb was provided with 40W/110V = 363mA of current. |
| Heater Load Test | Will the relay power a 170-250W space heater? | Pass | When the relay was powered, the space heated drew a minimum of 170W/110V or 1.5A and a maximum of 250W/110V = 2.27A of current. |

Fan Control

|  |  |  |  |
| --- | --- | --- | --- |
| Test Name | Testing Question | Pass/Fail | Additional Notes |
| Battery Check | Does the battery for this test still have a charge? | Pass | The voltage was about 8.05V, which is still greater than the logic signal (5V). |
| BJT Load Test | Will the BJT handle the 12VDC Fan load? | Fail | The device failed to handle the load and was destroyed. The maximum rated current (200mA) was below the current rating of the fan (250mA). |
| MOSFET Load Test #1 | Will the MOSFET handle the 12VDC Fan? | Fail | The device is rated to handle up to 100VDS, therefore the 8.05V load was no burden on the device. However, the gate threshold is much lower and triggers the fan while the jumper is floating. |
| MOSFET Load Test 2 | Will a pull-down resistor remove the floating signal? | Pass | The pull down network ensures that only the logic signal will be able to activate the transistor switch. |

IR Sensor Reading

|  |  |  |  |
| --- | --- | --- | --- |
| Test Name | Test Description | Pass/Fail | Additional Notes |
| Trigger Response | Does the sensor react accordingly and increment the given counter? | Pass | The sensor can detect any obstacle within its infrared beam. This includes non-human entities. |
| Trigger Speed | How fast can the sensor pick up a rising edge? | Pass | Through repeated testing, the sensor proved to be 90% accurate when triggering in rapid succession. Occasionally, the sensor would miss one gesture. This error is admissible and will be out of scope for the final demonstration |

## Eagle Schematic Design

The schematic design was run through the error-checking tool provided in the editor software. The schematic only returned several warnings, all of which pertained to missing values for certain components. These missing values were unimportant to the design, as all pertinent values were charted in the bill of parts. Polarity and header pinouts were added onto the silkscreen near the end of the design phase. The entire schematic for the main board and the button panel can be found in Figures 3A and 4A respectively.

## Eagle Layout Design

The layout design was run through the design rules check tool. The service found many errors within the design. Many of the errors detected were caused by the angles of the traces from the microcontroller pins. These issues will be explained in the Discussion. The entire board layout for the main board and button panel can be found in Figures 3B and 4B respectively.

# Discussion

## Prototype Enhancements

During this laboratory, the prototype was finalized to model the expected design as best as possible. The fans and lights were tested with a high load and confirmed to operate with the given components. Those components were then matched with surface mount versions to build on the board editor. Furthermore, the IR sensor was replaced with a simpler and smaller solution. The older sensor was inadequate for repeatedly counting entries and exits. This is due to the design of the PIR sensor used. The output signal was delayed for a minimum of five seconds. This delay is far too large to meet the design requirements. A new IR sensor was bought and tested in a counter system. The sensor reacted much faster than the older one. Fortunately, the sensor has a similar pinout and software implantation to the original sensor. Despite the success of the new sensor, it still had a major flaw. The sensor would still trigger even when a non-human entity passed through it. This was determine by waving a piece of paper in front of the sensor. This scenario will be out of the testing scope for the final presentation, as there are better but more costly ways of accomplishing this task on a larger scale.

## Button Panel

To ensure that the button input was properly debounced, a hardware debouncing implementation was designed on the panel board. The button attaches to ground and the input line. When pressed, the button will complete the circuit and send a high signal to the microcontroller. The RC pulldown network acts as a low-pass filter, which cleans the input signal. To calculate the resistor and capacitor values, the following formula was used to estimate an optimal RC circuit for a 10ms response:

The inverting schmitt trigger also helps filter the noisy input signal to produce a single edge trigger. The device limits the rising and falling edges to avoid issues such as bouncing [2]. This debouncing implementation was researched and compared to software solutions. This method was implemented due to its simplicity compared to the software solution. YouTube videos from Derek Molloy [3] and Jeremy Blum [4] were used as a reference to this debouncing solution,

## Design Process

The design process was a long and tedious procedure. Originally, the autorouter tool was utilized to help route the microcontroller traces. This was done in order to speed up the design process. Unfortunately, this was a poor choice. When running the design rules check, over fifty errors had been detected. Every single trace routed through the autorouter contained incompatible angles. To fix this mistake, every trace routed to the microcontroller was ripped up and rerouted manually. The result was much more acceptable. This process took an extensive amount of precision and patience, but was much more useful in the long run than rushing the design with the autorouter tool. This mistake was very costly, but it in turn forced our group to use our own intellect and skills to design the board. In the end, this mishap proved to be quite beneficial.

# Conclusion

In conclusion to the laboratory, the EAGLE software was thoroughly utilized to produce a circuit board design of the climate control and room occupancy system. Through careful and precise mounting and routing, design skills were developed and improved throughout the duration of the laboratory. The board design also brought to light many design requirements that were previously overlooked: power requirements, noise filtering, size limitations, and output denouncing. Overall, despite the many hardships faced, the design process proved beneficial to enhancing our hardware design skills.

# Reflection

Hardware implementation was heavily overlooked by our group. Since the prototyping was simple and intuitive, we predicted that the board design would be similar. However, through trial and error, we soon realized this to be wrong. As a result, the design requirements and prototype were improved on. We were given an opportunity to explore a real example of an embedded design procedure. Using EAGLE was a steep learning curve that required precision and patience to comprehend. As a result, there was much research involved prior to finalizing the design. The element14 community offered the most helpful resources in learning the software and provided useful tips for proper designing. Benjamin Heckendorn’s video *Getting Started with Cadsoft EAGLE* was a great resource that examined prior to the design process [5]. The lab assistants were also equally resourceful and helped clarify many topics regarding the design process. There was much to take from this laboratory, and more is expected for the upcoming laboratory.

# References

[1] “2n3904,” in SparkFun, 2003. [Online]. Available:

<https://www.sparkfun.com/datasheets/Components/2N3904.pdf>. Accessed: Nov. 5, 2016

[2] C. Cockrill, "Understanding Schmitt Triggers," in *Texas Instruments*, 2011. [Online]. Available: http://www.ti.com/lit/an/scea046/scea046.pdf. Accessed: Nov. 5, 2016.

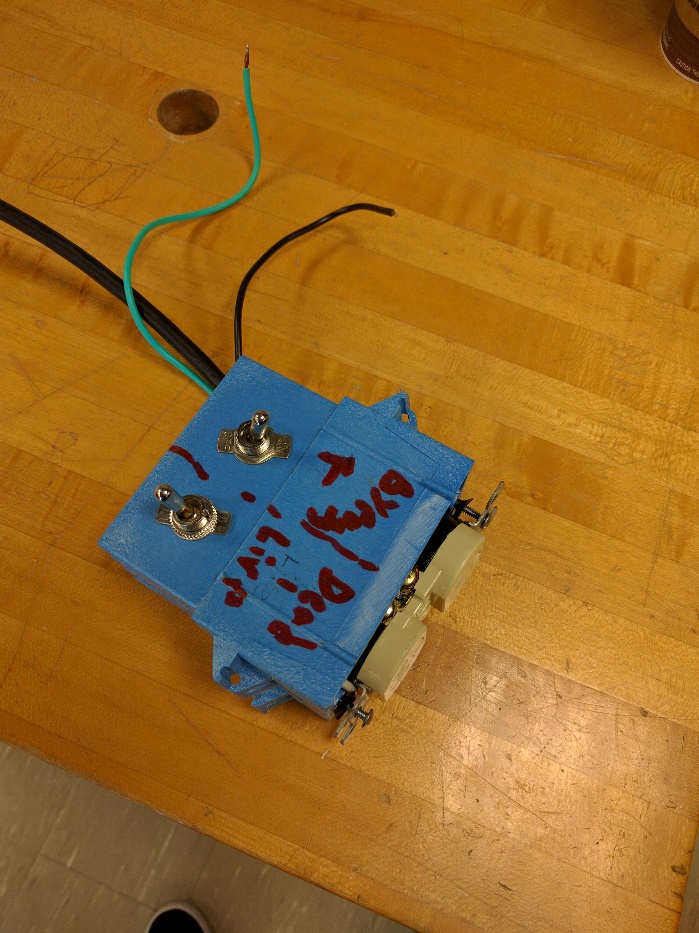
[3] Derek Molloy, "Digital electronics: Debouncing a push button switch (SPST)," YouTube, 2012. [Online]. Available: https://www.youtube.com/watch?v=tmjuLtiAsc0. Accessed: Nov. 5, 2016.

[4] Jeremy Blum, "Tutorial 10 for Arduino: Interrupts and hardware Debouncing," YouTube, 2011. [Online]. Available: https://www.youtube.com/watch?v=CRJUdf5TTQQ. Accessed: Nov. 5, 2016.

[5] The Ben Heck Show, "Getting started with CadSoft EAGLE," YouTube, 2013. [Online]. Available: https://www.youtube.com/watch?v=R4DYztYB6d4. Accessed: Nov. 5, 2016.

# Appendix

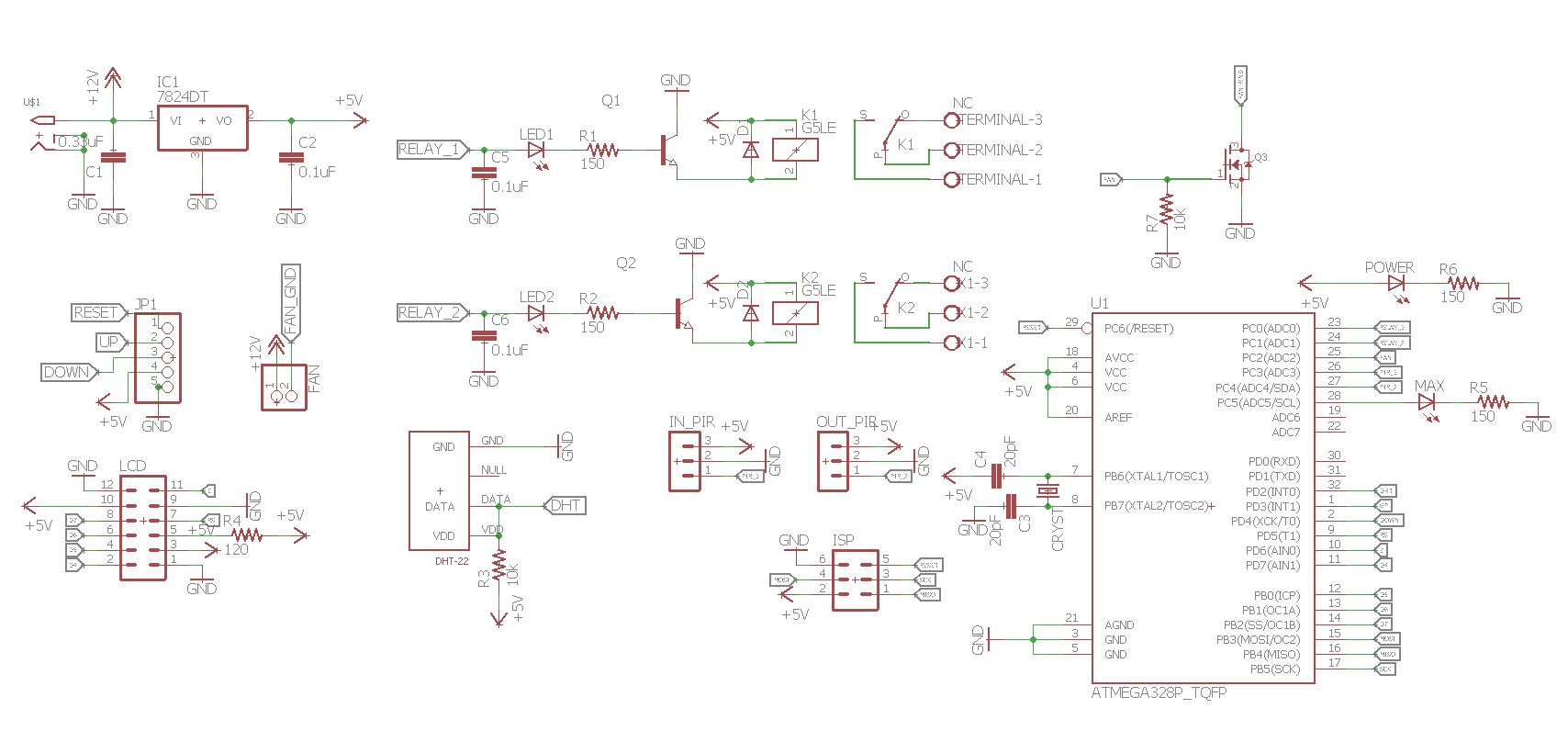
*Figure 1: Outlet Switching Mechanism*

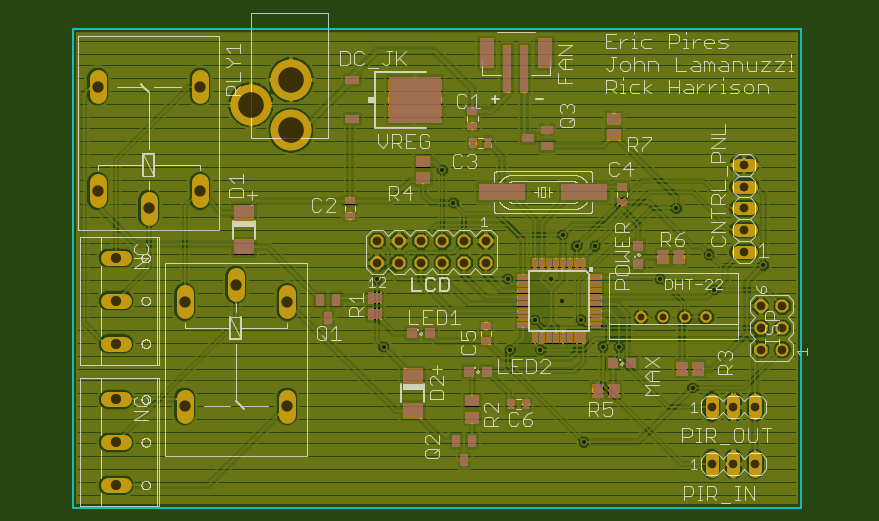
**

*Figure 2: IR sensor counting code*



*Figures 3A and 3B: Main Board Schematic and Layout*





*Figure 4A and 4B: Panel Board Schematic and Layout*

