Professor Heckathorn

Computer Science 440

29 November 2022

Mini Security Alarm Analysis

The following documentation is an overall summary of the development process our group partook in when finishing and creating our project.

Our first steps when deciding what creation, we would construct started with research of different types of projects one can make using hardware such as the Raspberry Pi or Arduino. After finding various articles online about numerous hardware projects, we discovered that the projects including cameras were intriguing, as the project’s usage of the camera for security functions was seen as a very useful and productive system in the real world. After delving further security-oriented projects, we then found the usage of sensors with the Raspberry Pi interesting enough that we would like to explore and learn their usage in hardware development. This was ultimately a big reason for our projects emphasis on tinkering with sensors. We thought making an alarm system would be an interesting and completable enough project, and a fun way to implement sensors. Our final version of our project would detect motion and sound an alarm, using PIR sensors and a Piezo buzzer.

After mutually understanding the functions and purposes of our project, we then began on its construction; This began with us researching which parts we would need to complete our desirable functions, along with ensuring compatibility amongst all the parts. We began with purchasing the most crucial part of the project, the computer board. Initially, we had planned to buy a Raspberry Pi board, but soon began searching elsewhere after learning of the high prices of Raspberry Pi boards. We soon found a cheaper alternative with sufficient specs to satisfy our projects needs and functions. The name of this alternative is called AML-S905X-CC (Le Potato), which we were able to purchase on Amazon. In order to complete the detection portion of our system, we would have to incorporate sensors to our build. After researching various sensors, we ultimately decided to use a passive infrared sensor (PID sensor). PIR sensors measure infrared light radiating from objects in its field of view. To show the output of the PIR sensor, we decided to use a speaker to indicate motion from the PIR sensor. We used a piezoelectric speaker or a piezo speaker, which would provide sufficient sound feedback for our system. Both the PIR sensor and piezo speaker were purchased on Amazon. We also opted into using a breadboard, to allow for work on the pins to be more secure, easier, and efficient. A 100-ohm resistor was also used to limit the power to the PIR sensor.

The first step we did was …

Like the development of many projects, ours was not completed without hardships and obstacles faced. The first we faced was selecting the best and most efficient operating system to use the board with. We initially believed that using the Armbian operating system would be the best, as it allowed for an efficient connection to the board. Unfortunately, using this operating system caused most functions used in the script to not be detected for an unknown reason. After multiple efforts to debug this weird issue with Armbian, we decided to switch the environment to Raspian, as we have seen various other projects that use that environment. Unfortunately, we encountered more concerns with this operating system. The operating system was difficult to navigate and traverse through, so we attempted Armbian again. This was because Armbian was similar to a Linux environment, which was more familiar to all of us. We were able to find another library that included functions that were compatible with our board. After settling with the Operating system, the next problem we encountered was managing the electrical connections with the components of our project. Shortly after, we learned that a breadboard can be used to easily manage those connections in a much more convenient manner.

Our board we used was the culprit of the many tedious issues we faced when constructing this project. Some of it’s specs were manageable to work with, however some of its limitations would just be tedious to work with. For example, the board only supports USB 2.0 Type A, when USB 3.0 would allow for us to use the keyboard we had. This led to us spending some time looking for a compatible USB 2.0 keyboard that could be used. Another spec issue was that we could not use 5W, as it would lead to the operating system to crash. The lack of power forced us to increase its wattage to 10W to make it work. The most frustrating and difficult problem that we’d eventually overcome would be learning the libraries and functions that would be required to use our board. This was mostly due to our board having few documentation to help the user from using it. The website from our board’s creators was poorly managed; It included broken links, outdated tutorials/guides, inactive forums with scarce support, among other things. Us opting to use an alternative board led to us conducting heavy research for functions and libraries used by our board, which took a majority of the time used on this project. We essentially had to view scarce documentation on various libraries, and tested most of them until they contained functions that would work with our project/board.

Overall, this group project proved being a very informative as well as entertaining project that allowed for us to learn about the basics of hardware development…