1. Introduction

Our project goal was to create a low cost home monitoring system that can be accessed from anywhere from in the world. The UI for this system is a webpage compatible with any HTML5 capable browser. This means you can access the page from your mobile device or computer. We set out to create a system capable of supplying a live video feed, configurable alarm system with motion sensor, and temperature and light monitoring systems. As of today, we have achieved most of our design goals with the exception of the temperature and light sensing modules.

1. Project Design

Our design consists of 3 distinct pieces. A custom PCB that houses our motion sensor, alarm system, and temperature and light sensing circuits. Our raspberry pi microcontroller which acquires and processes raw data, and hosts our web server. And lastly, our website, that communicates bidirectionally between our server(pi) and client.

Our custom PCB essentially consists of 4 different circuits, all working independently of one another. We used a BJT with a relay in order to control the alarm. A motion sensor hooked up directly to the pi. An analog circuit consisting of an LM305 temperature sensor and a photodiode. Because the pi doesn’t have on board Analog to digital conversion, we also had to include an ADC in our circuit that can convert our analog signals to digital that the pi can interpret. Our data processing on board the pi was handled by 2 separate Python scripts. Each of these scripts controlled GPIO pins that configured our alarm controls. We also installed an Apache web server on the pi so that we could host our website from home. Our index.html page (the page that you see when accessing the pi’s external IP address), was saved in our apache website folder. In order to communicate between the Pi and the website (remote client), we used websockets. Websockets are a new feature in HTML5 that are implemented within the webpage using JavaScript. The websockets allow us to send commands from the website to the pi using a push button. In order to have a live video feed, we used a program call MJPG streamer. This program is installed on the pi. Our camera captures video in real time and saves the captured stills to a temporary folder. MJPG points to this folder and streams the individual captures over port 8080. By using an iframe redirect, we were able to embed the stream into our index.html. The webpage also dynamically changes size to fit the device that is being used clientside. This provides a superior user experience.

1. Results

We had initially specified a live video stream at 1080p 60 fps. Our camera was capable of capturing at 1080p 60fps but other technical limitations forced us to decrease the video resolution and the framerate. MJPG has a maximum video resolution of 900p. So we limited our camera capture to this resolution. Additionally, we had to decrease the framerate to 15 fps because streaming live data over the internet is limited by the bandwidth of your upload speed determined by your ISP. Our home network is speced at 5 Mbps upload. The average file size of each jpg image was on the order of a couple of 100s of Kbytes, so this limited us. We had also set out to be able to control the alarm from the website. We were able to this using the websocket we had implemented. One shortfall of our alarm was that we used a PWM signal to control the alarm output. The pin we used for the PWM is a 3.3V output. For the frequency, we set the PWM to 2.4 KHz. However, because the signal level was only 3.3V, it is not that loud of an alarm. We encountered major problems in implementing our ADC circuit. We used an i2c protocol that used two separate lines to communicate with the pi. These lines require pull up resistors between themselves and the logic level (3.3V). Our custom pcb did not include such resistors and so we were unable to read data off of these lines. Additionally, we sought to send this data from the pi to the webpage using a websocket. But because we were unable to read data off of our ADC, we could not implement this feature.

1. Summarize

One change we would obviously make would be including pull up resistors between the SDA/SCL and 3.3V. This would have allowed us to implement additional features in our monitoring system. We also would have liked to make the PCB smaller. This would have reduced overall production costs significantly. Adding a text/email system to notify the user when the alarm is triggered. The design experience was very close to what we expected. We set weekly goals for completing just a small part of the project. If we had more time, we believe we could have completed all of our the initial design goals that we set forth.