Final Report

Yang Liu

Introduction

We designed and developed a webcam nearly from scratch over the past 9 weeks as a part of our course EECS 395/495 – Engineering System Design.

In addition to designing the webcam, we also created our own website and fabricated a 3D printed enclosure for the webcam.

The webcam features an Atmel SAMS48B microcontroller, AMW136 Zentri Wifi Chip and Omni vision OV2640 camera. The SAM4S8B features a maximum operating speed of 120MHz with 512kB of flash memory and 128kB of SRAM.

The AMW136 can communicate through UART, SPI, I2C, and USB and has 128kb RAM and 512Kb flash drive. The Wi-Fi chip has a feature where it acts as a server, so the website designed for the webcam can be hosted directly on the chip, which broadcasts the images captured by the camera.

The Omni Vision OV2640 was chosen as the desired camera, with a maximum possible frame rate of up to 15 frames per second.

We had initially made all the connections in a breakout board. This gives us easy access to the pins of the various peripherals and make the necessary connections. Once the code was developed and the camera was working with the breakout board, we had designed the final schematics and board using Eagle PCB.

The website was created using HTML and styled the page using CSS. To make the webpage interactive we used Javascript. The entire design of the webpage was fairly simple with limited interactions such as clicking a button to start/stop the camera and navigating between pages.

The 3D enclosure was designed using Onshape. Onshape is an online CAD software and one of the main reasons we used it because it was easily accessible anywhere and we could share our designs with anyone. This also eliminates the need to install a software in our devices to access the file.

Once the enclosure was designed, the zcode was generated using Z-suite and printed using Zortrax 3D printer. The Z-Suite is a fairly simple software to use and takes us through the various steps right from placing the object on the printer to generating supports, selecting the material, pattern, thickness. Average printing time was between 2-3 hours.

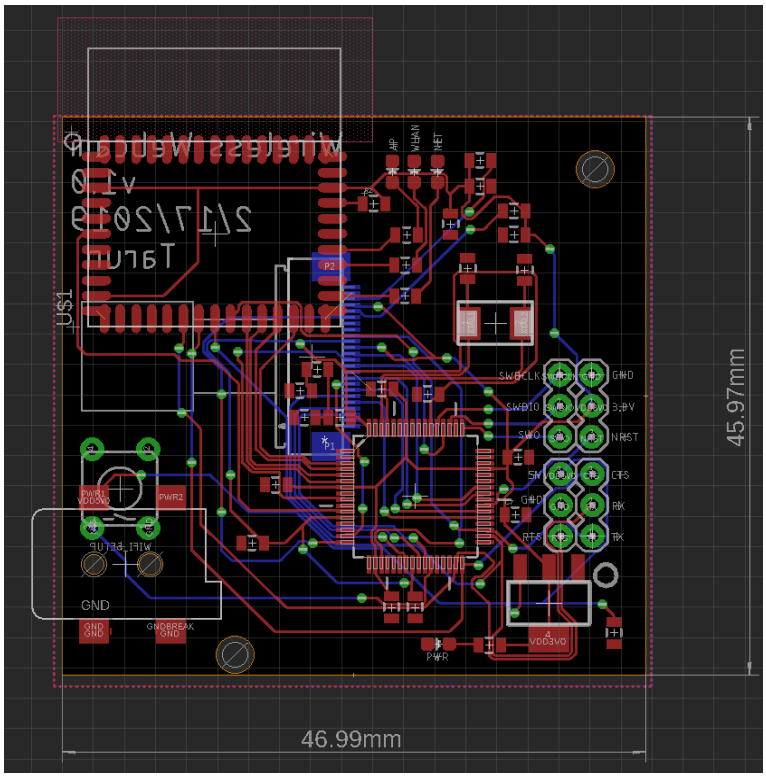
PCB Design

The final PCB was decided after several iterations. Our goal was to make the board as small as possible. The maximum dimension of the board which we were allowed to use was 50mm x 50mm. Although this seems to be a fairly large board area, the problem starts to arise when we start routing all the components. I started designing the board and after several iterations was able to only reduce it to 46mm x 46mm. Also, I wanted to keep the camera, the buttons and the LED indicators on one side and the Wifi chip and the microcontroller on the other side. This way the user gets to see only the relevant components and the processing happens in the other side.

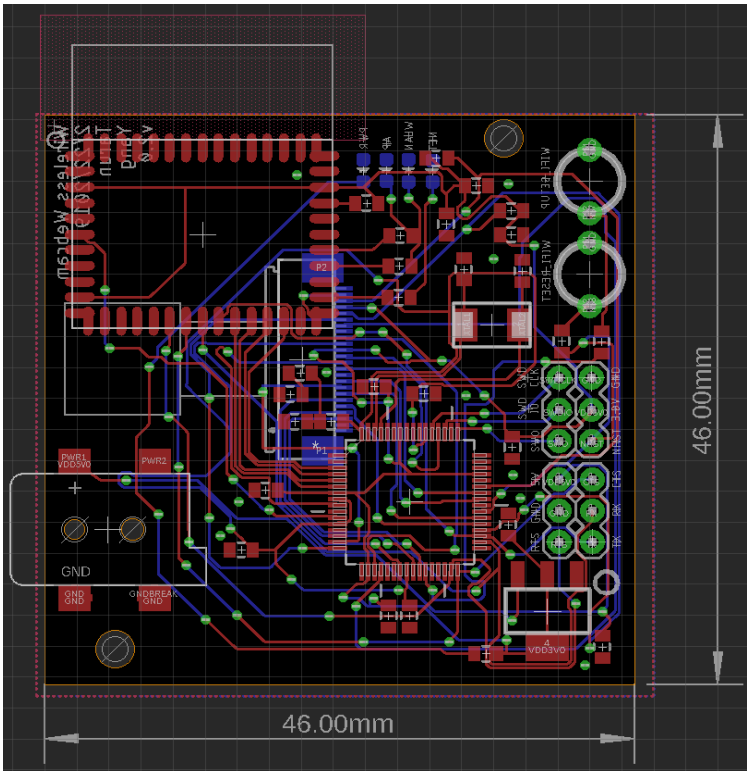
In the first few iterations on the PCB there were a few l problems I had encountered.

- The first problem is that the button was kept very close to the barrel jack and this could have posed potential problems during the soldering of the components on the PCB.
- The second problem was I didn't give sufficient space for the screw heads.

All these were fixed in the following iterations and this was the design of our final PCB. Below is the diagram of our final PCB.



Initial Design



Final Design

C Code

Since me and my partner are trying to split the work during our first few iteration of the C code. I am writing up the c code with There were totally 19 functions we had to implement out of which we had to write 5 functions from scratch and make few or no changes for the remaining 14 functions. These functions were obtained from the example projects in Atmel Studio.

The basic flow of the program is as follows. First, the GPIO of the Wifi chips are configured and the root folder is set. Then the clock, camera, Wi-Fi chip and all the pins are initialized and then the program then enters the main loop, where the microcontroller checks for a Web Setup request, and if connected to the internet, waits for a websocket connection. It continues to wait until a connection is available. Once there is a connection, it takes the picture, calculates the length, creates a file on the Wi-Fi chips flash, then writes to the stream so that the website knows to update its image.

The two functions that took most of the time were write_image_to_file and find_image_len. The snippets of the code are given below:

This function essentially checks the entire buffer for the Hex value FFD8 which is the

```
uint32_t find_image_len(void)
{
    //The first loop is to find the starting address of the JPEG (0xFFD8)
    for(int i=0;i<CAM_BUFFER; i++){
        if((image_dest_buffer_ptr[i]==0xFF) && (image_dest_buffer_ptr[i+1]==0xD8)){
            start_buff = i;
            break;
        }
        else
        image_length = 0;
    }

    // Once the start of the JPEG image has been found, the end of the image is found (0xFFD9)
    int j=0;
    for (int i=start_buff;i<CAM_BUFFER;i++){
        if((image_dest_buffer_ptr[i]==0xFF) && (image_dest_buffer_ptr[i+1]==0xD9)){
            image_length = j+1 ;
            break;
        }
        j++;
    }
    if(image_length==0)
        return 0;
    else
        return image_length;
}
```

starting of the JPEG file. Once it gets the starting of the image, the location is stored in a

variable (in this case start_buff) and then the loop is run from the starting value up to the point where the loop locates the Hex value FFD9 which is the end of the JPEG file.

The image length is calculated and returned by the function.

```
void write_image_to_file()
{
    // if there is an image detected where the length exists
    if(image_length!= 0)
    {
        // create a string that contains 20 chars can hold the command
        char image_length_transfer[20];

        // put "image transfer" and image length together to make the final command
        sprintf(image_length_transfer,"image_transfer %d\r\n",image_length);

        // write the command to the WIFI moduel
        write_wifi_command(image_length_transfer,2);
        while(transfer_start)
        {
            // create an array for transfer content with a length that detected for that specific image
            uint8_t image_transfer[image_length];
            char tem_char;
            uint8_t tem_int;
            for(uint32_t i = 0 ;i<image_length;i++)
            {
                // put the content on the address starting at the starting point to the tem variable
                tem_int = image_dest_buffer_ptr[start_buff+i];
                tem_char = tem_int;
                // output the char tem to the terminal
                usart_putchar(BOARD_USART,tem_char);
            }
            transfer_start = false;
        }
    }
}
```

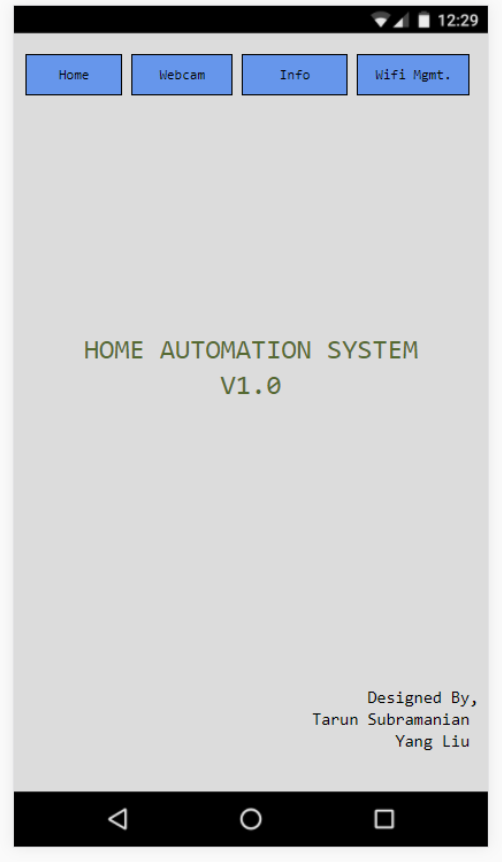
This function transfers the image from the processor to the server where it is displayed. Firstly, we check if there is an image or not. This can be done by simply checking the length of the image, the we write the image_transfer function and specify the length of the image.

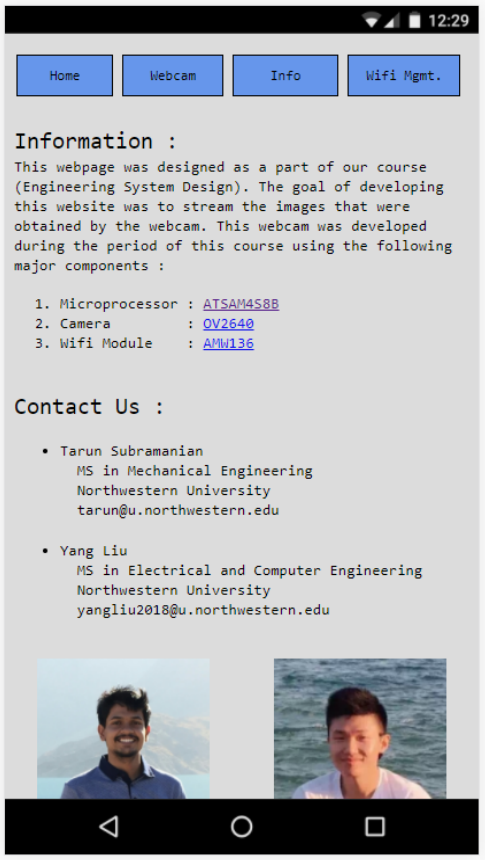
Once it acknowledges it, the processor then sends the image character by character using the usart_putchar function.

Once the transfer is complete, the flag is set to false and it is completed.

Website:

Designing the website was very important since the user will be interacting with the webcam through this website. We wanted to make sure the design of the website was simple but very user friendly. We provided the user with a navigation bar so that he/she can move through all the pages easily.

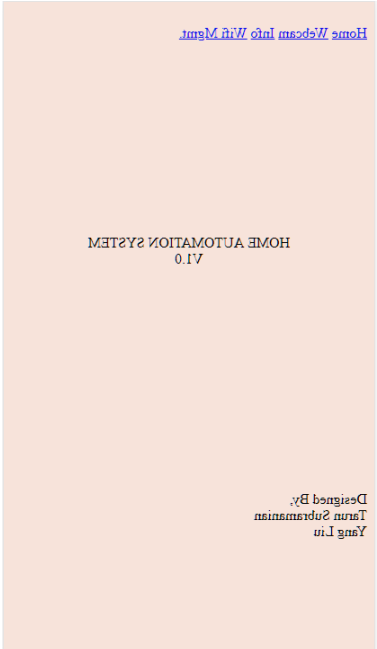




We developed this website for a Nexus 6p / Nexus 5s. A challenge in developing the website was the Javascript where we had to create a mouse button for starting/stopping the webcam.

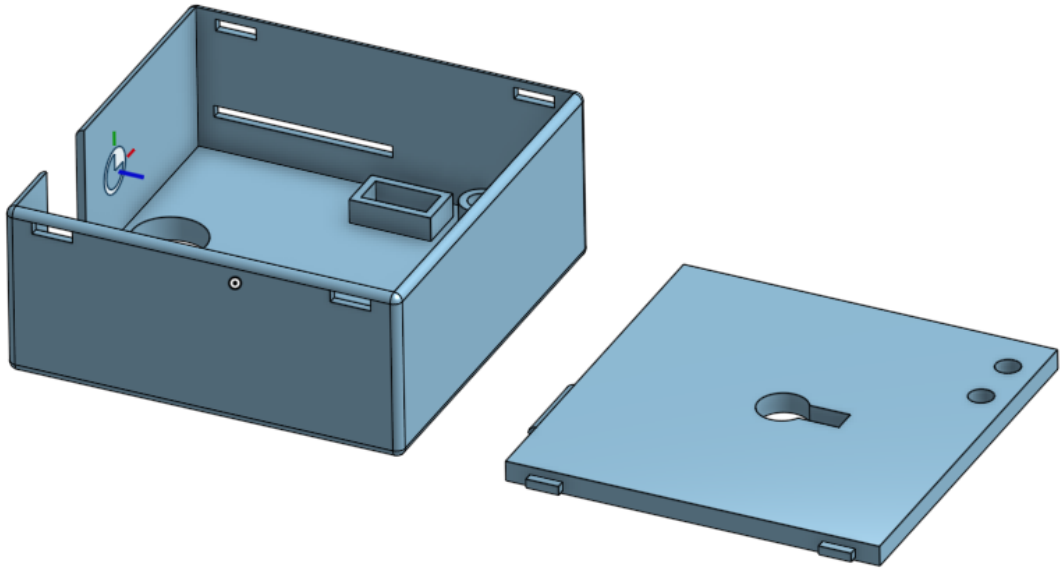
The functions onOpen() and onClose() determine the state of the button and change the text of the button accordingly. Depending on the state of the Websocket, the connection was made using the doConnect() function or closed using websocket.close.

The previous version of this website that I designed looks like this final one but we make it nicer and functional finally.

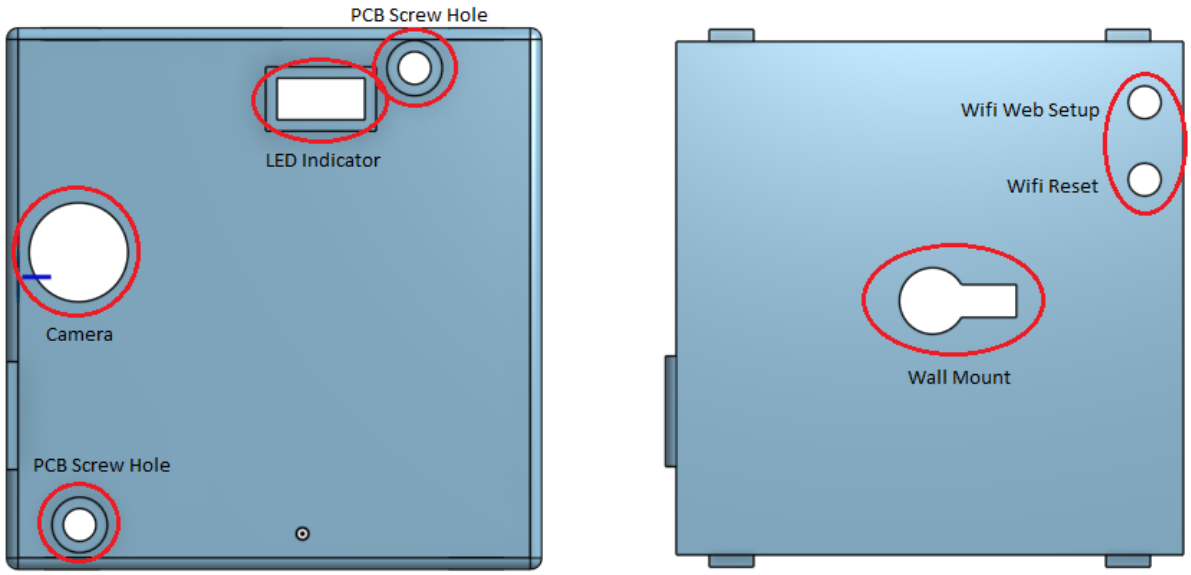


3D Design:

The final part of the webcam project was to design a case to hold all the components together.

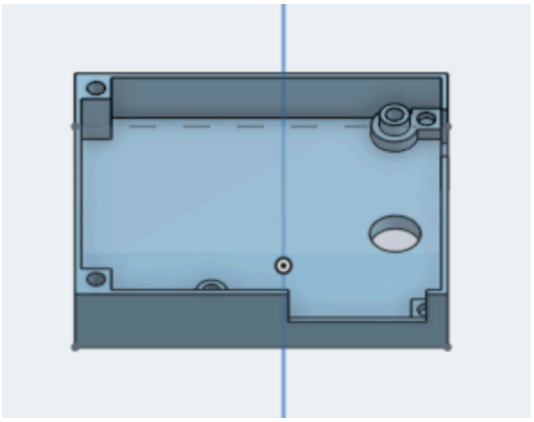


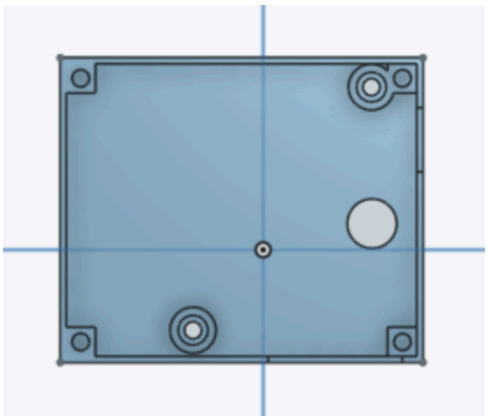
We had to go through several iterations of the case to compensate for the tolerances in the 3d printers and make sure all the holes/cutouts aligned with each other. Initially we had made the case a little big and made sure the pcb fit properly. Then we iterated the design with very small tolerances and below is a snapshot of our final design.

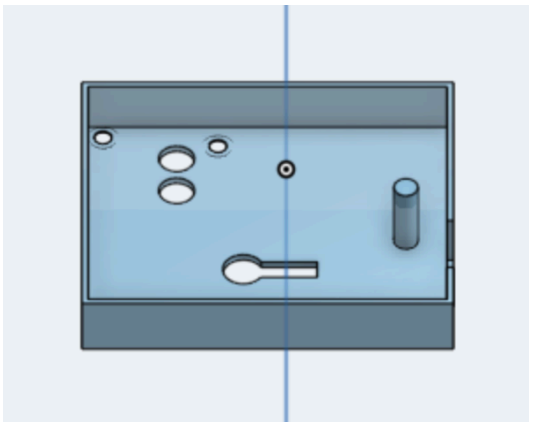


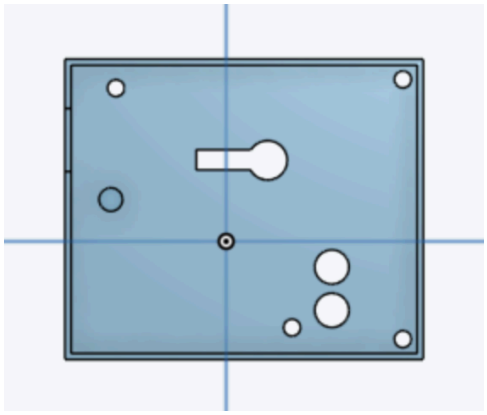
We designed it in such a way that the lid is snap fits on the case and hence eliminated the use of screws. There are screw holes to hold the pcb in place and cutouts for the LED indicators and the camera. The lid consists of cutouts for accessing the buttons and the wall mount cutout.

Since the 3D printing takes about 10 iteration until we reach our final design, I designed the first few version of the 3D printing. We make my first version design small to be just fit the board and adding the cut off/ holes and hanger hole to its correct position.

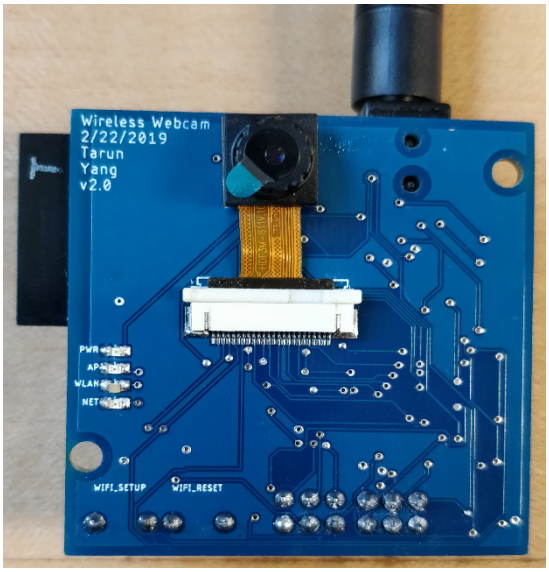


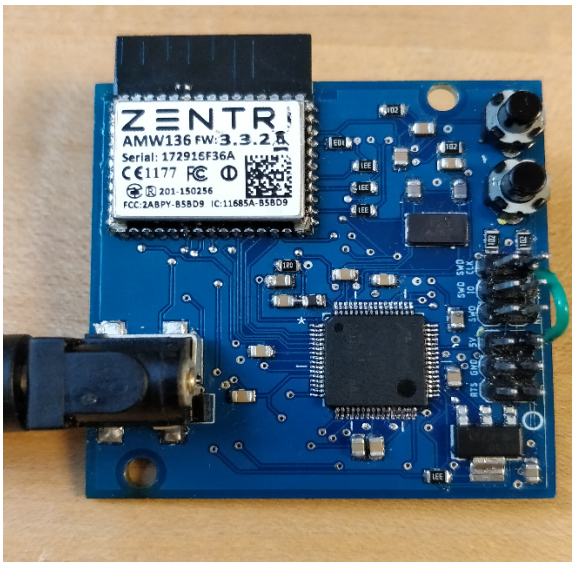


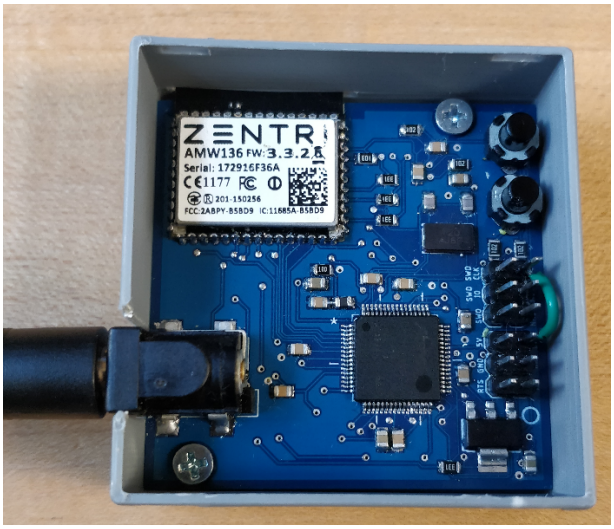


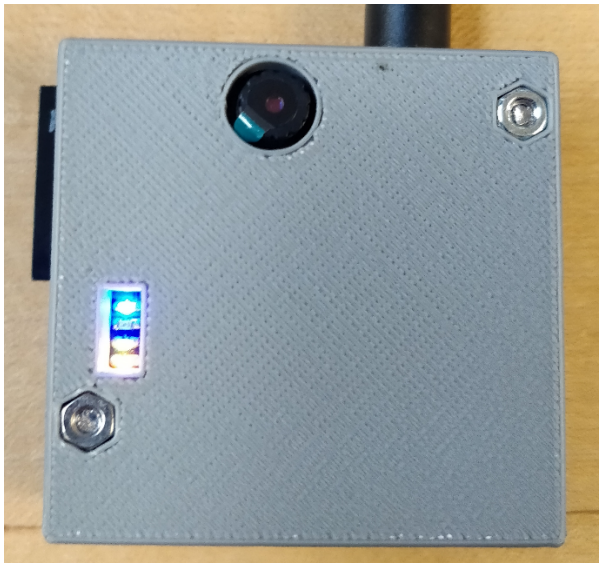


Final Webcam:









One mistake we did in the final PCB is that we did not ground the CTS pin. This was a pretty easy fix and we soldered a wire from the CTS pin to the ground since these pins were accessible as header pins.

Having made this change in our PCB, we used our board and it worked properly.

If I was given a chance to start over I would have spent more time optimizing the PCB design. When I was designing the PCB, my goal was to make sure the board worked, and size of the board really did not matter to me. However, after several iterations, I did try to make the board smaller but was not very successful in doing that because I did not place the components in an optimized manner. Rather, I grouped them according to their type so that I would better understand the board. If you look at the PCB, all the buttons,

the LED indications, the pins were all grouped together and a most of the top left and majority of the bottom side of the PCB is empty. I could have easily reduced by 5-6mm at least if I had used these spaces more efficiently.

Teamwork:

For the teamwork, me and my partner are working on ourselves best to this course and have a good approach to gain the final goal even there are some hard time, we still try split the work evenly and make the load balance with our other courses during the quarters. Since the some of the dues might be conflict to other courses due date, so we try learning to be better collaborating together and manager our time during the whole design. Fair enough, we learn to understand in each other's prospective and put more effort on this project if another one of us is struggling with other courses. That is the part I learn on this project in terms of teamwork. It is about understand and the ability to expose yourself to a work load if others are on other things.

Learning:

I was actually tried applying your individual project back in last Spring on a Bluetooth project. I am interested in learning hands-on experience on a combination of hardware and software which is what this course offers. The most part I am aiming to learn is not about just pure coding or by putting things together, but an experience of finishing a project which are practical and could be used in real life from scratch. By understanding the things behind it and putting time into debug and make things works are the things I am hoping to learn, especially in this course. I learnt a bit less than what I image but it is myself load issue since I have other courses like everybody else. I wish I could do more on coding the Javascript file and have a better understand on how website functions are applied. Personally, if the number of stations in the lab could be increased then that is the ideal case, but I know it is according to the budget, I am so surprised that we could use the 3D printing machine for this study. I think the number of the students registered on this class is good, two team members in one group is good, but I think the way build up team could be changed to assignments, in other words, the team members are chosen by prof. That will reduce the chance that someone is not putting effort to the project but choose a hardworking teammate.

Conclusion:

This course is so much fun, every point that I learnt is so practical and I gain a hands-on experience. In terms of course load and mental pressure, I have a different feeling of facing barrier compared to other courses, maybe it is because this course is building up something, but not by solving those calculation exercises and problems. The way I am learning thing on this course is completely different than other course, overall experience is 9.9 out of 10. I really wish I could enroll in this Design 2 for next quarter.