UNIVERSIDADE FEDERAL DE SANTA CATARINA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA E ELETRÔNICA
EEL7802 - PROJETO EM ELETRÔNICA II

RELATÓRIO FINAL

SISTEMA DE SEGURANÇA ADICIONAL E DE FÁCIL USO PARA ESTABELECIMENTOS COMERCIAIS

CAMERA TRAP

Autores:
Luciano Silva do Lago
Pedro Henrique Kappler Fornari

Florianópolis, 10 de Dezembro de 2015.

Sumário

1. Introdução

Esse relatório irá abordar o desenvolvimento do projeto “Camera Trap”, realizado pelos estudantes de engenharia eletrônica da UFSC Luciano Silva do Lago e Pedro Henrique Kappler Fornari. Inicialmente será apresentada a motivação do trabalho, explicando a ideia inicial para solucionar o problema. Após isso, serão abordados os problemas enfrentados e algumas etapas do projeto, dando maior ênfase na etapa final, visto que as etapas anteriores já foram melhor detalhadas em relatórios passados. Por fim, serão apresentados os resultados obtidos assim como os aprendizados adquiridos com esse trabalho, mostrando os principais erros e o que ainda pode ser feito para melhorar o projeto e quem sabe transformar o protótipo desenvolvido durante esse período em um produto real.

2. Motivação

Como apresentado na proposta inicial deste trabalho, a numerosa quantidade de assaltos que ocorre em empresas e estabelecimentos comerciais faz com que empreendedores estejam constantemente melhorando seus sistemas de segurança, os quais estão cada vez mais complexos e mais caros. Além disso, mesmo com sistemas de segurança ativados, muitos comerciantes e empresários continuam sendo assaltados. Isso ocorre porque os assaltantes conhecem os detalhes de cada sistemas e a instalação, sendo que, dessa forma, conseguem desativar os equipamentos e sabem quais deles são suficientemente seguros e não tentam invadir o local.

Como, em maioria, micro e pequenos empreendedores não tem orçamento suficiente para contratar os sistemas de segurança que conseguem evitar prejuízos por furtos e assaltos, o foco desse trabalho se encontra em desenvolver uma tecnologia que seja de fácil instalação, para que o próprio proprietário consiga instalar o equipamento, e que não seja exageradamente caro para manter os empreendedores a par do que está ocorrendo em seus negócios em sua ausência. Assim, com informações suficientes, as providências necessárias poderão ser tomadas, acionando o órgão responsável por cada problema que o equipamento apresentar aos seus usuários, que no caso são os próprios empreendedores.

3. Proposta Inicial

No inicio do desenvolvimento desse projeto foi decidido que a melhor forma de manter os usuários desse sistema de segurança informados do que acontece em suas empresas de forma rápida e barata é enviar uma foto do local para um aplicativo celular cada vez que um movimento fosse detectado. Além disso, para facilitar a instalação é necessário um equipamento portátil, que deve ser simplesmente conectado a uma tomada ou, melhor ainda, utilizar baterias.

Com isso em mente, a primeira ideia do sistema foi utilizar o sensor câmera OV7670 juntamente ao módulo WiFi ESP8266 e um microcontrolador externo mais uma sensor de movimento PIR, de forma a captar a imagem e enviá-la pelo módulo WiFi. Assim, com os dados em um processador suficientemente capaz de processar os bytes da imagem, como um computador ou um celular, seria feita a conversão dos bytes para imagem e assim apresentada ao usuário.

Para armazenar os dados, inicialmente foi pensado em utilizar uma memória flash, visto o tamanho consideravelmente grande de uma imagem se comparada à memória do microcontrolador. Também foi levantada a ideia de se utilizar uma placa de FPGA para ter um processamento mais rápido e uma memória grande, porém devido ao pequeno conhecimento dos desenvolvedores em linguagens de descrição de hardware (HDL) e ao pequeno tempo de desenvolvimento se comparado à complexidade do projeto, foi decidido utilizar um microcontrolador, sendo ele especificamente o Cypress PSoC 4 CY8C4245AXI-483, o qual possui além das funções básicas de um microcontrolador certa flexibilidade de hardware, permitindo interconectar pinos e utilizar portas lógicas.

4. Desenvolvimento

Foi desenvolvido um cronograma na primeira etapa do projeto o qual foi readaptado ao longo do semestre, porém ele foi utilizado como base para o controle das atividades, sendo modificado somente quando algo inesperado ocorria, o que não ocorreu do segundo para o presente relatório.

O projeto foi iniciado com alguns diagramas, que serviram para auxiliar no desenvolvimento, identificando *bugs* inesperados, e identificar quaisquer erros de projeto antes de iniciar qualquer teste ou codificação. Com esses diagramas prontos e validados iniciamos os estudos dos componentes, os quais foram bastante detalhados nos dois relatórios anteriores, mas é valido lembrar dos pontos mais importantes de cada componente.

O sensor de presença foi configurado para ter um tempo de resposta rápido, porém com um atraso entre acionamentos máximo, visando diminuir o envio de imagens rápido demais, ainda a resolução foi ajustada para ter máxima percepção de movimentos do ambiente, sendo que essas regulagens foram feitas por meio de dois potenciômetros contidos no sensor escolhido.

Foram verificados os sinais de sincronização com a câmera utilizando o osciloscópio do laboratório, os resultados foram comparados aos da folha de dados, confirmando os sinais recebidos, com esses sinais será feita a captação da imagem pelo sistema. Também foi estudado como configurar a câmera, a qual utiliza um protocolo SCCB e dessa forma, conforme explicado no Relatório II foi ajustada sua resolução e protocolo de compressão das imagens captadas.

O estudo do módulo WiFi, que possui um microcontrolador Xtensa LX106 embarcado, foi feito inicialmente verificando como realizar a comunicação entre ele e o microcontrolador externo, sendo que existia a possibilidade de se utilizar uma comunicação serial entre os dois, utilizando comandos AT, programar o microcontrolador do ESP8266 em LUA ou utilizar a interface de programação do Arduino, que, conforme explicado no segundo relatório, foi a opção escolhida.

4.1. Mudanças no decorrer do projeto

Foi observado ao longo do projeto que algumas partes da ideia inicial não funcionariam como o esperado, logo, algumas partes foram alteradas. Algumas dessas alterações foram citadas em outros relatórios, porém esta parte do trabalho visa apresentar todas as alterações do projeto inicial e os motivos que levaram a essas alterações.

A velocidade necessária para se captar a imagem foi crucial ao longo de todo o desenvolvimento do projeto, fazendo com que as maiores alterações acontecessem em função disso. Inicialmente, quando foi levantada a ideia de se utilizar uma memória flash externa, com alguns testes e leituras de *datasheets* foi possível observar que a velocidade de comunicação, tanto serial quanto SPI com as memórias eram insuficientes para obtermos uma imagem com qualidade, logo essa opção foi descartada.

Para substituir essa memória a decisão tomada foi de utilizar o módulo WiFi como memória, visto que ele possui memória maior do que o microcontrolador externo, porém os

pinos que o ESP8266 possui para GPIO's são escassos. Foi estudado possibilidades de utilizar os pinos de BOOT (GPIO0, GPIO2 e GPIO15) e os pinos serial (GPIO1 e GPIO3), junto de uma multiplexação de sinais, onde seriam enviados 4 bits de dados da câmera por vez, sendo que para isso, seria enviado um sinal de controle PCLK com o dobro da frequência que a câmera fornecia, além disso utilizar os pinos de BOOT e de serial não é muito confiável, pois dependendo dos sinais lógicos nestes pinos durante a inicialização do módulo, o mesmo inicia de modos diferentes (download de código pela serial, boot pela memória flash externa ou boot por cartão SD), logo adaptamos essa ideia para nosso primeiro protótipo.

Também foi estudada a possibilidade de criar um controle por maquina de estados no ESP8266, sendo que essa máquina de estados seria controlada pela leitura do pino analógico que o módulo possui, porém, devido a velocidade de leitura do pino analógico ser bastante lenta e este tipo de implementação não nos permitiria entrar em modos de baixo consumo de energia automaticamente essa solução também foi descartada.

A solução encontrada foi utilizar uma função da câmera que cria um controle dos sinais de sincronização, que fazia uma espécie de AND com os sinais PCLK e HREF (PCLK era válido somente quando HREF fosse alto), desta maneira seria necessário enviar apenas o VSYNC e o PCLK como sinais de sincronização para o ESP8266. Além disso, foi necessário diminuir a resolução da câmera de 640x480 (600kB de memória) para 176x144 (49,5kB de memória), tamanho ainda insuficientemente pequeno para caber no nosso módulo. A câmera envia uma média de 16bits por pixel no modo YUV422, que envia primeiro o canal Y, o Gama, e depois Y ou U, canais Chroma (a ordem de envio podem variar e dependem das configurações nos registradores da câmera. Por final, para fazer a imagem caber no módulo WiFi foi decidido capturar apenas a imagem preto e branco, ou seja, apenas o canal Gama, para fins de testes.

Ainda existe outra solução para este problema, que seria comprar uma versão da câmera que contém um buffer FIFO (First In, First Out) e facilita a comunicação com módulos ou transmissões mais lentas, essa versão é, em média R$10,00 mais cara do que a proposta inicial, logo seria a melhor solução, porém como o tempo de chegada do componente seria maior do que o tempo para a apresentação final foi decidido usar a técnica anterior.

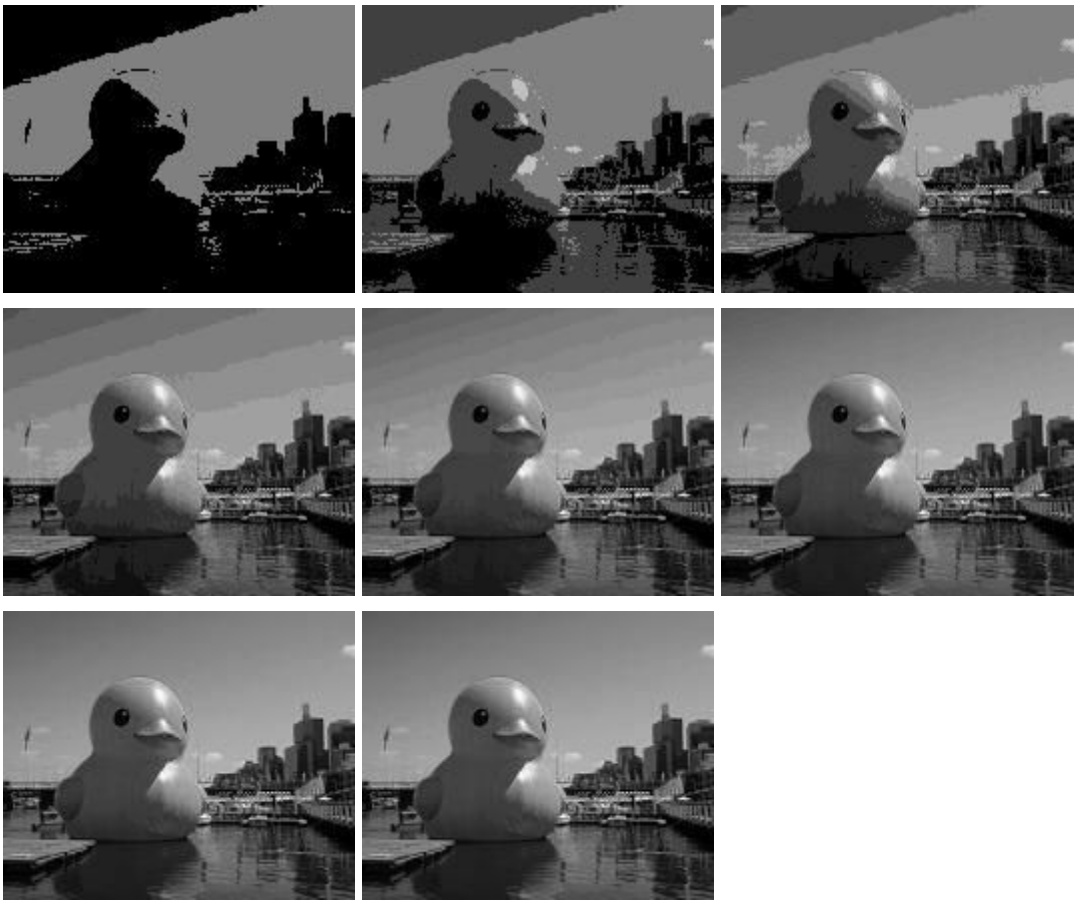
Outra possível solução é utilizar um microprocessador e não um microcontrolador, como, por exemplo, um Raspberry PI, onde poderíamos utilizar um sistema operacional embarcado Open Source como o GNU/Linux tendo memória e velocidade suficiente para armazenar os dados da câmera com maior velocidade e facilidade, porém, também devido ao tempo de entrega do projeto final, deixamos isso para possíveis trabalhos futuros e focamos nosso desenvolvimento na solução apresentada acima.

4.2. Testes da nova solução

Inicialmente, a fim de validar a ideia de utilizar apenas 4bits pois o módulo ESP8266 carece de GPIOs para captar a imagem, foram feitos testes com uma imagem qualquer da internet para verificar quais seriam as perdas reais de todas essa modificações que foram feitas até o momento.

Original colorida.

Esta é a imagem original utilizada para os testes, foi escolhida uma imagem de forma a facilitar a visualização de formas quando os captados eram alterados. Agora serão postas todas as imagens, de 1bit a 8bits de resolução.



Esse procedimento foi para fim de teste mesmo, pois não pode-se dizer que a imagem recebida perderá qualidade de imagem tal como a imagem presente do pato, um fator que muda dessa imagem para a recebida câmera é a qualidade, esta é uma imagem de alta qualidade que foi redimensionada, a do OV7670 será uma de aproximadamente 0.3MP, baixíssima qualidade para os dias de hoje.

Como pode-se observar, a imagem com os 4bits mais significativos, apesar de bem distorcida, ainda é reconhecível e por isso essa configuração foi utilizada, visto que é possível utilizar os pinos 12, 13, 14 e 15, acelerando o processo de captura dos dados por uma única leitura de porta e não pino a pino pois os bits capturados já estão na sequência certa. Além disso os 4 its menos significativos parecem controlar o contraste e a

luminosidade da imagem, por isso escolhemos uma constante para esses pinos, para que isso interfira o mínimo possível na imagem final.

Abaixo está o script em python para a manipulação desta imagem.

```python
from PIL import Image

im = Image.open("rubber-duck9.jpg")
pix = im.load()
mask_table = [0x80, 0xc0, 0xe0, 0xf0, 0xf8, 0xfc, 0xfe, 0xff]

y_table = []
for y in range(144):
    for x in range(176):
        r, g, b = pix[x, y]
        yc = int((r + g + b) / 3)

        y_table.append(yc)

cnt = 0
for i in range(8):
    for y in range(144):
        for x in range(176):
            yc = y_table[cnt]
            mask = mask_table[i]
            cnt += 1

            pix[x, y] = (yc & mask), (yc & mask), (yc & mask)

    im.save("rubber-duck9-{0}bitsPB.jpg".format(str(i + 1)))
    cnt = 0
```

4.3. Integração dos módulos

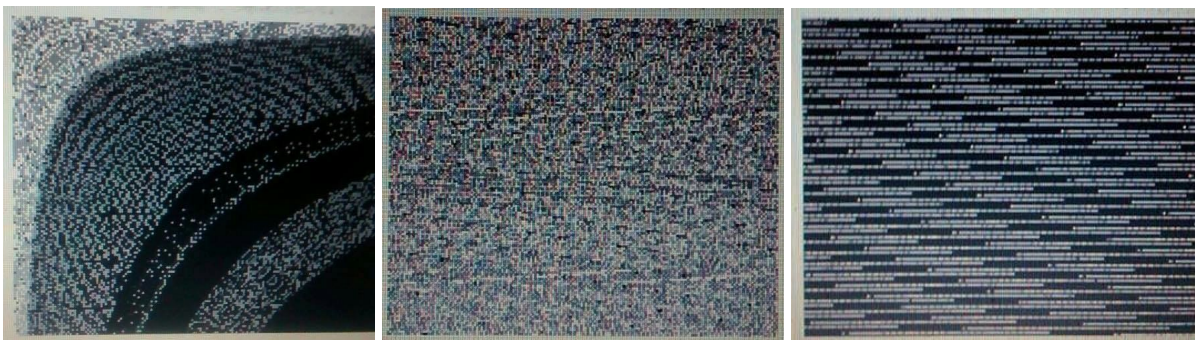
Com todas as partes finalmente prontas foi iniciada a integração entre os módulos, infelizmente, como o sistema é dependente de todos os módulos, não houve nenhum teste individual da câmera, para visualizar a imagem recebida.

Foi utilizado o microcontrolador para inicialmente configurar a câmera, visto que cada vez que ela é reiniciada a configuração deve ser refeita, após isso ele é responsável por fornecer o sinal de XCLK para a câmera. O ESP8266 recebe o sinal de sincronização, sendo ele uma AND entre HREF e PCLK, e o VSYNC diretamente da câmera para verificar o inicio e fim de cada imagem e 4bits de dados provindos da câmera, D7, D6, D5 e D4. O firmware rodando no ESP8266 trata o VSYNC como uma interrupção, sendo que essa interrupção habilita a função que lê os dados sincronizados pelo sinal de sincronização fornecido pelo microcontrolador, onde foi utilizada a técnica de *polling* para receber a imagem por ter um código razoavelmente extenso. Esses dados são armazenados em uma matriz e, posteriormente, são enviados por um servidor TCP/IP. O método de *polling* foi escolhido após o método por interrupções do sinal ter falhado, o módulo ESP8266 apresentou-se bastante instável quando com sinais com frequências razoavelmente altas (12MHz no caso), mas este método, o de *polling* também não funcionou, pois não conseguia capturar os pixel de maneira correta, e enquanto ele mandava uma imagem estava na verdade mandando pixels aleatórios de vários frames.

Um programa em Python se conecta a esse servidor e então começa a receber pacotes contendo partes da matriz de dados que o módulo WiFi está enviando. Com isso o programa trata os dados separando os bits que ainda devem ser separados e utiliza métodos de uma biblioteca pronta, PILLOW, a mesma utilizada para passar de RGB para PB, para transformar estes bytes em RGB, e então é gerada uma imagem BMP, a qual é apresentada na tela do computador.

Um problema que não teve solução a tempo da entrega do projeto foi que os dados recebidos, quando convertidos para imagem, não apresentam algo concreto, não é possível ainda identificar a imagem se comparada ao ambiente original. Isso pode estar ocorrendo por vários fatores e por isso alguns deles foram testados, corrigindo erros no firmware. A primeira intuição foi de que a manipulação dos bits feita estava errada, pois após ler a porta, que tem 32 bits, o código fazia uma manipulação por deslocamentos e operações booleanas, aplicando algumas máscaras, para armazenar apenas 8bits de dados em cada posição da matriz. Verificando essa manipulação foram encontrados alguns problemas nas máscaras aplicadas e assim que consertados foi identificado que os bytes recebidos pelo programa em phyton eram todos 0x00, indicando que ainda existiam problemas na codificação. Para ter maior certeza de que as interrupções estavam realmente sendo identificadas foi utilizado a técnica de polling com cada uma delas, VSYNC e sinal de sincronização, onde o sinal de sincronização mostrou um comportamento melhor quando o polling estava sendo utilizado se comparado a uma interrupção externa. Também foi percebido que a indicação dos pinos GPIO4 e GPIO5 do módulo WiFi estavam trocados, o que já havia sido informado por alguns sites, porém foi esquecido. Outro fator importe são os registradores do sensor câmera que possuem poucos e até nenhuma documentação, para isto foram usados exemplos do kernel Linux e exemplos da internet e feitos alguns testes.

Com todas essas correções, além de várias tentativas, os resultados começaram a melhorar, porém as imagens captadas ainda estão longe de ter boa qualidade, como podemos observar abaixo, onde elas estão apresentadas. Inicialmente, quando os dados transmitidos eram, sem exceção, 0x00, a imagem obtida era totalmente preta, como pode-se observar abaixo.



Abaixo segue o script para recepção de dados, em python.

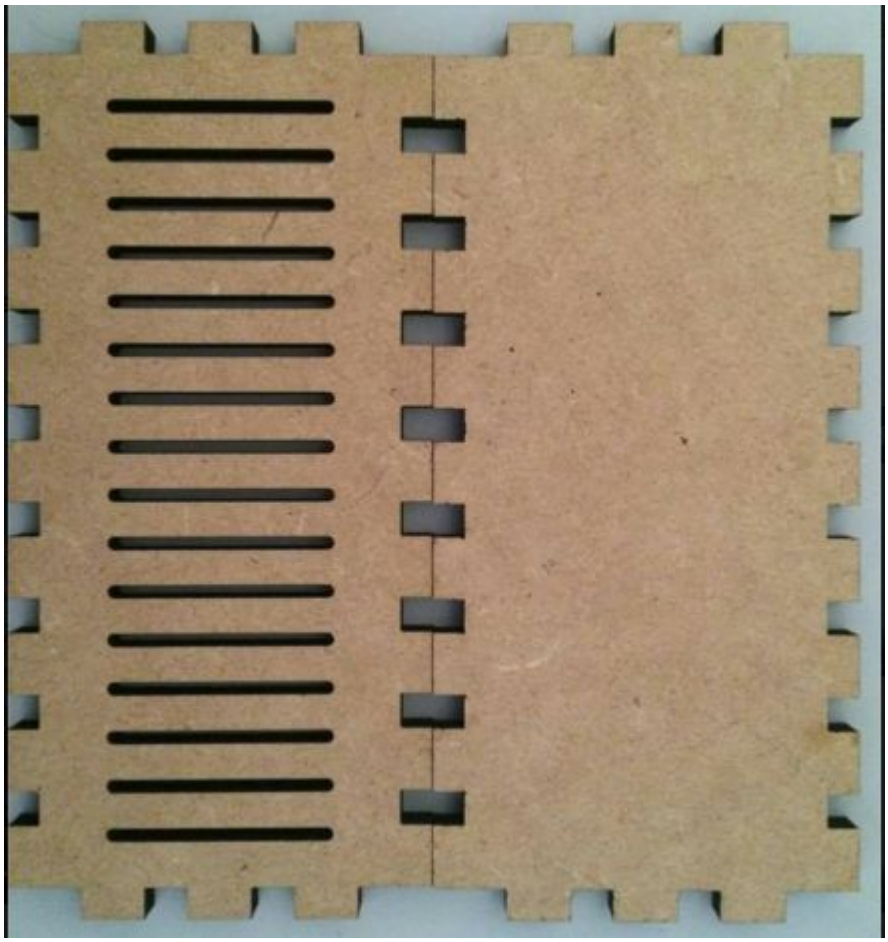
```
import socket
from PIL import Image

im = Image.new("RGB", (176, 144), "white")
pix = im.load()

sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
```

```
sock.connect(("192.168.1.3", 5000))
sock.send(b"client")

data_in = []
for i in range(18):
    data = sock.recv(1408)
    print(data)
    for c in data:
        # faz escala de 0x10,0xf0 para 0x00,0xff
        c = c & 0xf0
        c = int((c - 0x10) * 0xff / 0xe0)
        data_in.append(c)

sock.close()

cnt = 0;
for y in range(144):
    for x in range(176):
        yc = data_in[cnt]
        cnt += 1

        # uma boa aproximacao
        pix[x, y] = int(yc), int(yc), int(yc)

im.show()
```

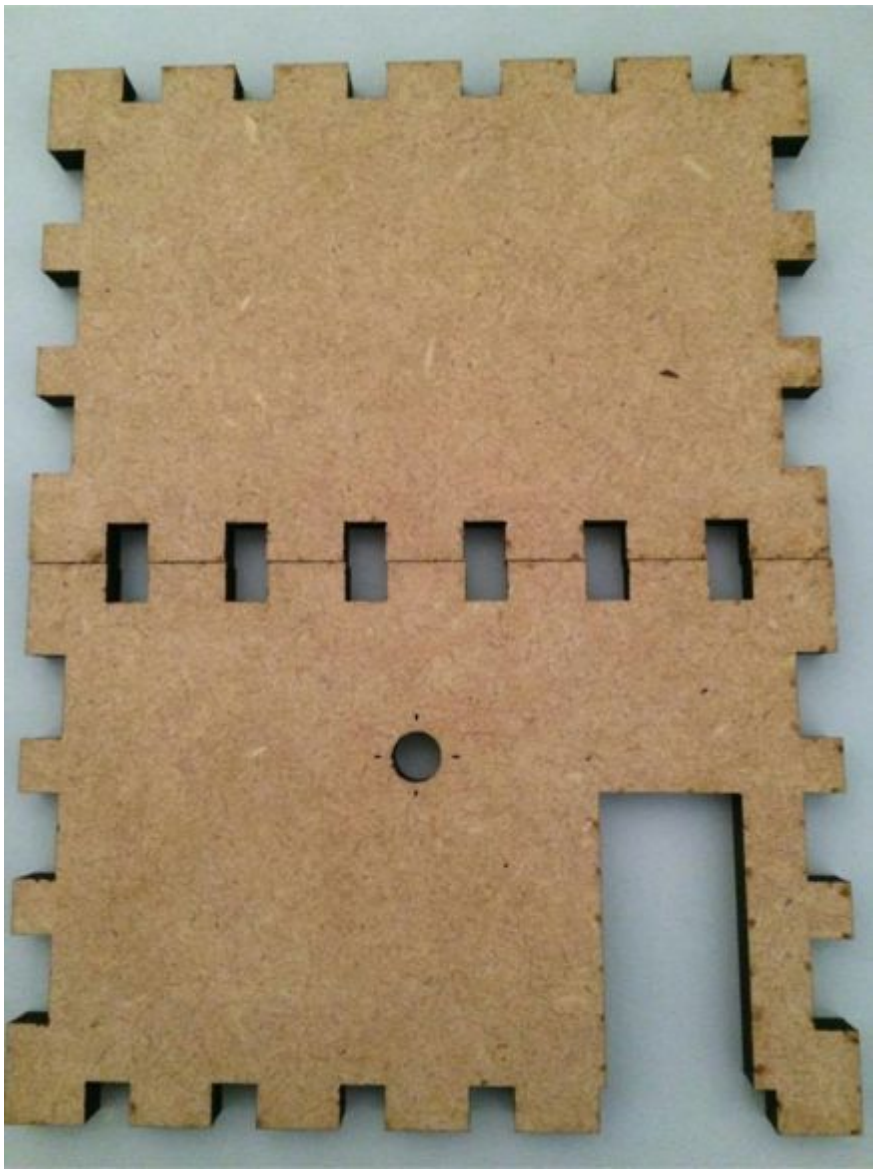
5. Design do produto

Para proteger o circuito e deixar o protótipo mais apresentável, foi desenvolvido, em parceria com o Instituto Senai de Inovação uma caixa em MDF, utilizando uma máquina de corte a laser. O desenho da caixa foi feito por meio do software SolidWorks, o qual permite o desenvolvimento de objetos 3D.

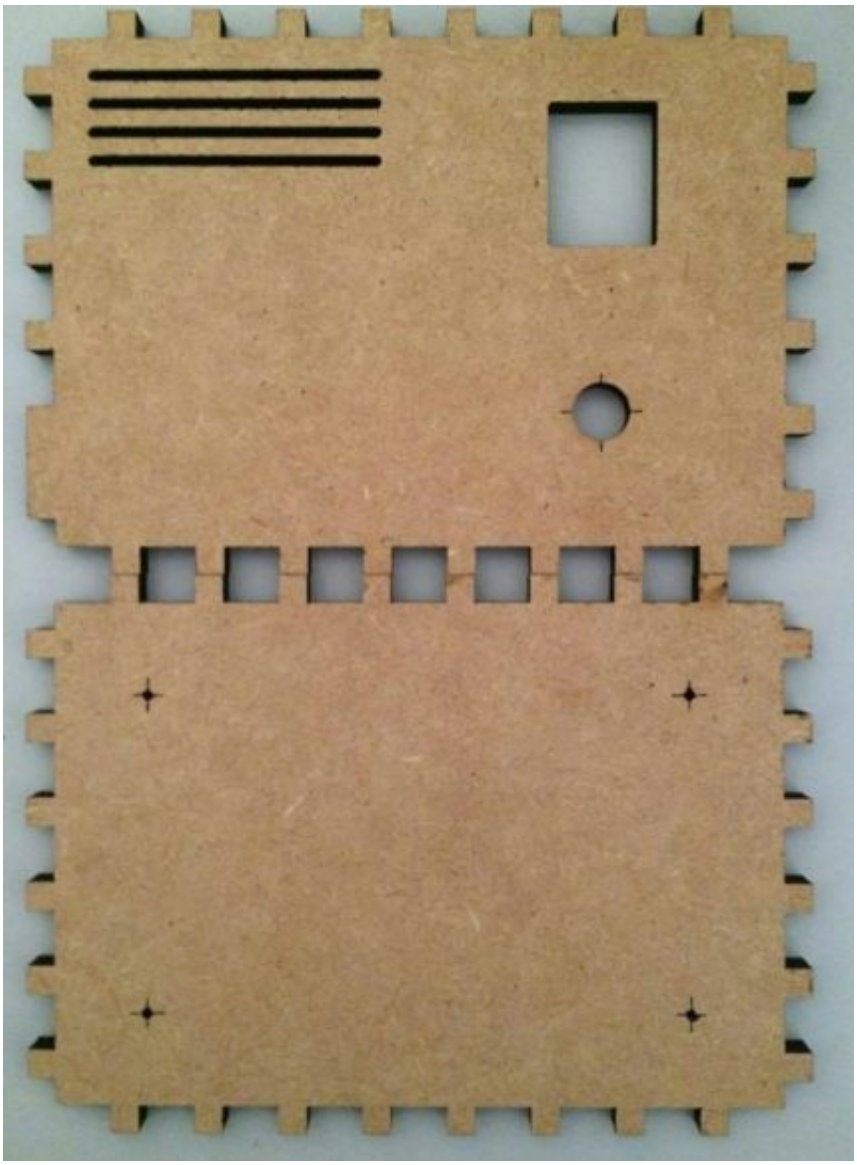
Para isso, foi desenhada cada peça da caixa, sendo elas: base, topo, laterais, frente e verso, sendo que para cada uma delas foi utilizada uma técnica para fixação por pressão, onde as bordas são cortadas de forma a se encaixarem, porém cada encaixe é feito um pouco maior do que deveria para ser encaixado perfeitamente, e dessa forma a pressão feita para encaixar as peças da caixa ajudam na fixação da mesma. As fotos de cada peça estão apresentadas nas imagens abaixo.



Estas peças são, respectivamente o topo e a base da caixa, a ventilação foi posta no topo visto que o ar quente tende a subir devido a sua densidade, evitando que o sistema aqueça demais.



Estas peças são as laterais da caixa, foram feitos buracos para a passagem do cabeamento de alimentação e outro para qualquer debug necessário, visto que esse é um protótipo de testes.



Finalmente são apresentadas a peça frontal e a peça posterior, respectivamente, foram feitos espaços para a câmera e para o sensor de presença na parte frontal e na peça posterior foram adicionados furos para facilitar a fixação, sendo que é possível escolher entre parafusos, pregos, ventosas, etc.

6. Conclusões e trabalhos futuros

Como já foi comentado, apesar de bastante trabalho do grupo, ao final do semestre não foi possível apresentar uma imagem coletada pela câmera e enviada para o computador, porém houve um imenso aprendizado dos desenvolvedores. Foram estudados três hardwares que vêm sendo cada vez mais utilizados pelo mundo, tanto pelo seu preço quanto por sua qualidade. O processamento da imagem trouxe grandes dificuldades que foram enfrentadas pela dupla, tendo um grande estudo na parte de configuração da câmera e de multiplexação de pinos para que fosse possível ao menos demonstrar o funcionamento do sistema de transmissão. Além disso, o módulo ESP8266 fez com que a dupla conhecesse diversas plataformas para o desenvolvimento de produtos com internet, o que é extremamente importante para trabalhar com IoT, mercado que, segundo a CISCO, terá 50 bilhões de equipamentos conectados até 2020. Além das plataformas, foi estudado como implementar protocolos de transmissão WiFi, como TCP/IP e UDP, e como receber os dados transmitidos no computador.

Foram estudadas linguagens de programação para desenvolver o programa dos módulos, sendo C para o microcontrolador, C++ para o ESP8266, um pouco de LUA também para o ESP8266 e o envio de comandos pelo protocolo I2C para a câmera, que é bastante utilizado em outros tipos de módulos, visto que economiza uma grande quantidade de pinos. Plataformas de desenvolvimento de hardware e de peças 3D também foram estudadas, sendo elas o Altium, para desenvolvimento de PCB's e o SolidWorks para desenvolver a caixa, softwares que são fundamentais na engenharia, sendo o Altium ainda mais importante para a engenharia eletrônica.

Ainda nesse projeto foi percebida a fundamental importância de um planejamento estratégico e funcional para gerenciar o desenvolvimento do projeto. Para isso foram estudados softwares para desenvolvimento de cronogramas como o Planner e o MS Project, de forma a acompanhar o andamento do projeto, cumprimento de metas e a verificação de pontos que ainda deviam ser realizados. Além disso o desenvolvimento de diagramas funcionais, estruturais e sequenciais auxiliaram para a verificação de erros de projeto antes mesmo de começarmos o desenvolvimento, introduzindo os integrantes aos conceitos da modelagem de sistemas, o que facilita e muito a validação das ideias iniciais, qualquer apresentação para possíveis clientes e evita o re-spin, como é chamada a alteração do projeto após boa parte dele já ter sido desenvolvida.

Como o trabalho ainda não foi concluído é possível ainda citar quais passos serão tomados daqui em diante, pois possivelmente esse projeto será levado a competições de empreendedorismo e inovação, além disso algumas soluções para os problemas que ocorreram ao longo do projeto foram apresentadas, porém a utilização de algum módulo de alta velocidade para receber a imagem é fundamental e será prioridade para os próximos passos do projeto, sendo que pode ser utilizada a câmera com buffer FIFO ou um microcomputador embarcado, ainda está sendo estudado quais são as vantagens e desvantagens de cada solução, como preço, velocidade, diminuição de tamanho, possíveis atualizações com maior número de funções do equipamento, entre outros. Um microcomputador embarcado já foi inclusive estudado, uma plataforma de desenvolvimento

da marca OLIMEX conhecido por RT5350F-OLinuXino, que possui um MIPS embarcado e tem módulos WiFi a disposição.

Também é necessário o desenvolvimento de um aplicativo celular para receber a imagem, para isso, durante o semestre já foram estudadas algumas plataformas de desenvolvimento de aplicativos celular, sendo que uma tentativa foi feita com a plataforma do MIT App Inventor, porém a falta de funções para processamento de imagens apresentou dificuldades no desenvolvimento do aplicativo, de forma que deverá ser utilizado outra plataforma para desenvolver o mesmo, como programação orientada a objetos em Java, Python ou C++.

Finalmente foram pesquisado alguns concursos nos quais o projeto poderá ser inscrito, de forma que investidores possam conhecer nosso trabalho e talvez auxiliarem no desenvolvimento com dinheiro ou equipamentos. Dentre os concursos que pesquisamos estão o Ideation Brasil, que ocorre em Florianópolis; a competição Intel de sistemas embarcados; a Telit Cup Brasil; e o Desafio IOT, sendo as duas ultimas encontradas no site embarcados.com.br. Com esse investimento poderá ser desenvolvido um design melhor para o produto final e materiais de marketing para, se tudo der certo, transformar esse projeto em um produto que realmente nos traga lucro e tenha utilidade para a sociedade, auxiliando principalmente micro e pequenos empreendedores e comerciantes.