



PONTIFÍCIA UNIVERSIDADE CATÓLICA DO PARANÁ – PUC-PR

CCET – CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA

ENGENHARIA DE COMPUTAÇÃO

Luiz Renato Hey Schmidt

Rafael da Silva Neves Serrano

PROJETO INTEGRADO

CCAA – CALHA DE CHUVA AUTOMOTIVA AUTOMÁTICA

Curitiba

30/11/2009

Luiz Renato Hey Schmidt

Rafael da Silva Neves Serrano

PROJETO INTEGRADO

CCAA – CALHA DE CHUVA AUTOMOTIVA AUTOMÁTICA

Documentação do projeto integrado
referente à disciplina de
Microprocessadores II, valendo como
nota parcial das PA citada, sob
supervisão do professor Afonso Ferreira
Miguel, da Pontifícia Universidade
Católica do Paraná.

Curitiba
30/11/2009

Resumo - As calhas de chuva são principalmente instaladas em carros que não possuem sistema de ar condicionado para que se possa abrir a janela com o objetivo de desembaçar os vidros ou simplesmente para a circulação do ar em um dia chuvoso. Além desses motivos há também aqueles que não gostam de estar o tempo inteiro com o ar condicionado ligado pois o mesmo prejudica o desempenho do motor e aumenta o consumo de combustível. Apesar dessas vantagens, grande parte dos consumidores não adotam a calha de chuva por um simples motivo, é esteticamente feia e faz com que o carro perca parte de seu design. A principal idéia do sistema de Calha de Chuva Automotiva Automática, ou simplesmente CCAA, além de manter sua funcionalidade original, é ser esteticamente aceitável, isto é, que não prejudique o design do carro nem sua aerodinâmica, que são dentre outros fatores decisivos na hora da compra de um veículo. O sistema CCAA depois de instalado permite a ativação da calha automaticamente quando há detecção de chuva no ambiente, ou manualmente através do estímulo do usuário.

Palavras - chave: CCAA, calha, chuva, design, e veículo.

Abstract – The Rain Gutter are mainly installed in cars that don't have air-conditioning system, allowing to open the windows on rainy days when the windshield gets foggy. There are also other reasons, like those who just don't like the fact that the Rain Gutters makes the car esthetically ugly, making the car losing its original design, decreases engine's performance and increases fuel consumption. The main idea of the Automatic Automotive Rain Gutter or simply the CCAA, (acronym in Portuguese) beyond keeping the original functionalities of a normal Rain Gutter, it is that it doesn't harm the car's aerodynamic and preserves the original design of the car, which is mostly the decisive factor. After installed, the CCAA offers to ways of control, the automatic one, which is activated when there is rain detection or the manual one, wich is controlled by the user.

Keywords: CCAA, gutter, rain, design and vehicle.

SUMÁRIO

1. Introdução	4
2. Objetivos do Projeto.....	4
3. Desenvolvimento do Protótipo.....	5
3.1 Projeto Mecânico.....	8
3.1.1 Lista de Materiais Utilizados	10
3.2 Projeto do Hardware.....	10
3.2.1 Lista de Materiais Utilizados	13
3.3 Projeto do Software	13
3.3.1 Código Fonte	16
4. Conclusão	25
5. Referências	25

1. INTRODUÇÃO

Diariamente nas ruas é possível observar que muitas pessoas possuem o sistema convencional de “calhas de chuva” instalado em seus carros. Esse tipo de equipamento possui uma grande demanda dentre os consumidores, pois quando instalado em carros que não possuem sistema de ar condicionado, permite a abertura parcial da janela acelerando o desembaçamento dos vidros e resultando numa melhor circulação do ar em um dia chuvoso. Além dessas vantagens, para aqueles que possuem ar condicionado em seus veículos, as calhas de chuva continuam sendo uma boa alternativa pois não aumentam o consumo de combustível e nem reduzem o desempenho do motor, que são fatores que ocorrem com o ar-condicionado ligado constantemente.

Mesmo com todos esses fatores positivos, a grande maioria das pessoas não aderem esse sistema simplesmente por ser esteticamente feio e fazer com que o carro perca parte de seu design e aerodinâmica. Sendo assim, observa-se a necessidade da produção de um equipamento semelhante ao existente que só seria acionado e projetado para fora do veículo quando houvesse intervenção do usuário em um controle manual, ou automaticamente através de um sensor de chuva.

2. OBJETIVOS DO PROJETO

O projeto CCAA, tem como objetivo a integração e ampliação dos conhecimentos adquiridos até o presente momento no curso de Engenharia da Computação, enfatizando as matérias de eletrônica e microprocessadores.

Além da parte técnica desenvolvida, como hardware e software, existem fatores primordiais para a conclusão efetiva do projeto que baseam-se na divisão e organização de tarefas, execução de trabalho em grupo, implementação dos conhecimentos práticos e

teóricos adquiridos tanto no meio acadêmico quanto no cotidiano de cada integrante da equipe.

3. DESENVOLVIMENTO DO PROTÓTIPO

O desenvolvimento do protótipo inicia-se com a pesquisa e criação de uma parte mecânica para que seja possível a movimentação da calha, seguido do projeto de hardware e paralelamente o do software que controlará todo o sistema.

Abaixo seguem algumas imagens dos rascunhos feitos para a confecção das peças que futuramente seriam utilizadas e dos circuitos a serem desenvolvidos.

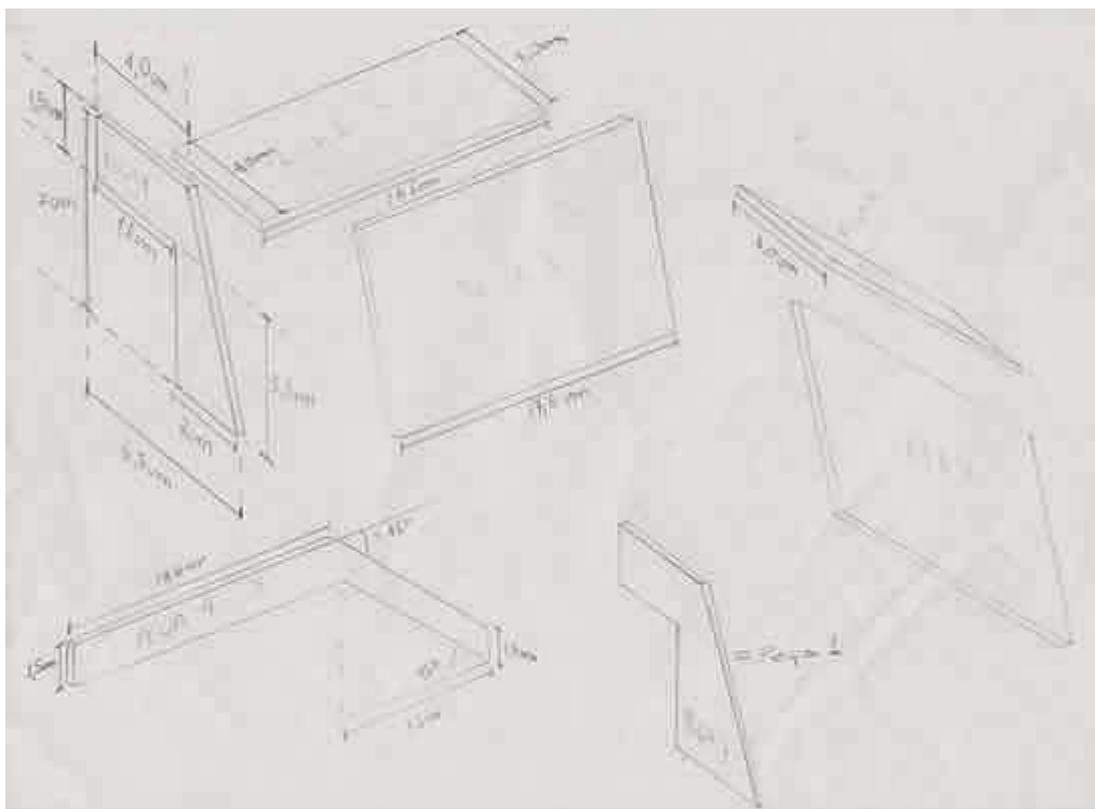


Figura 01 – Projeto da calha de chuva. Fonte: Autores do projeto.

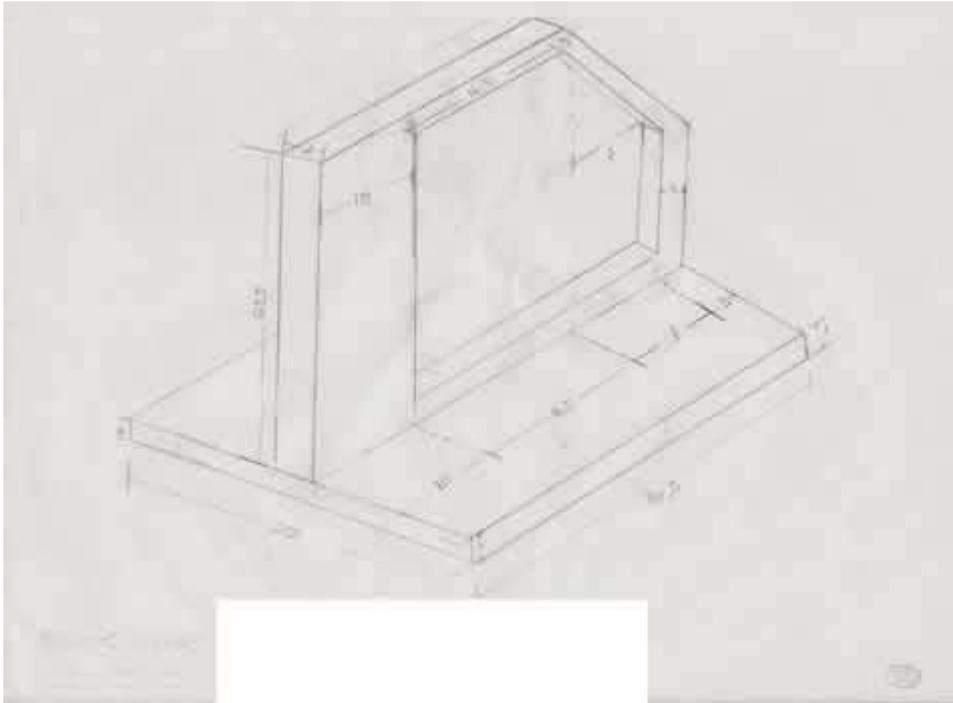


Figura 02 – Projeto da porta em MDF. Fonte: Autores do projeto.

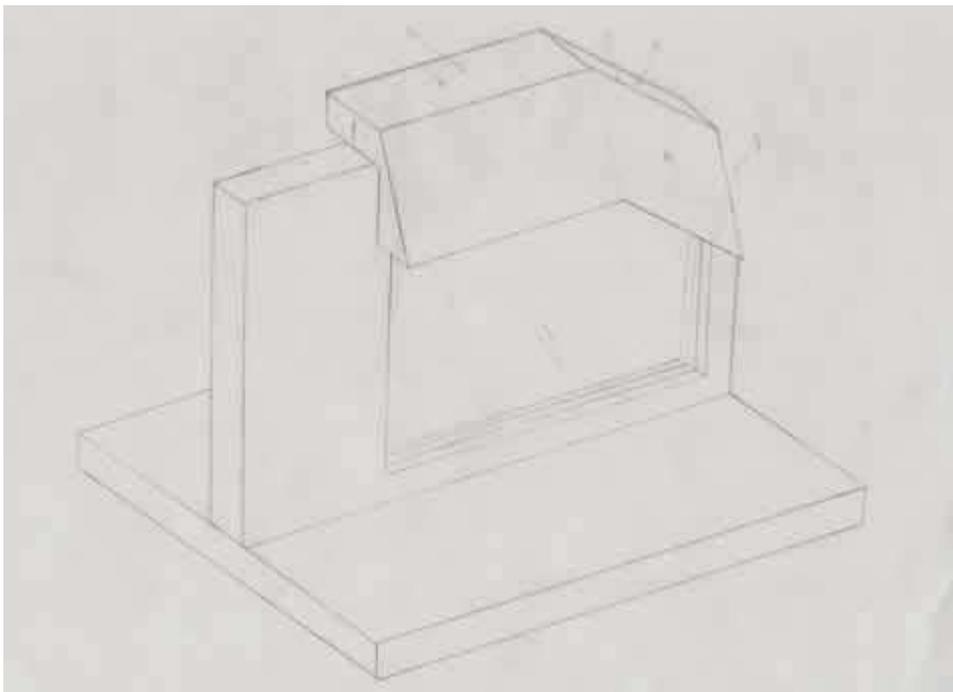


Figura 03 – Projeto da porta em MDF com a calha acoplada. Fonte: Autores do projeto.

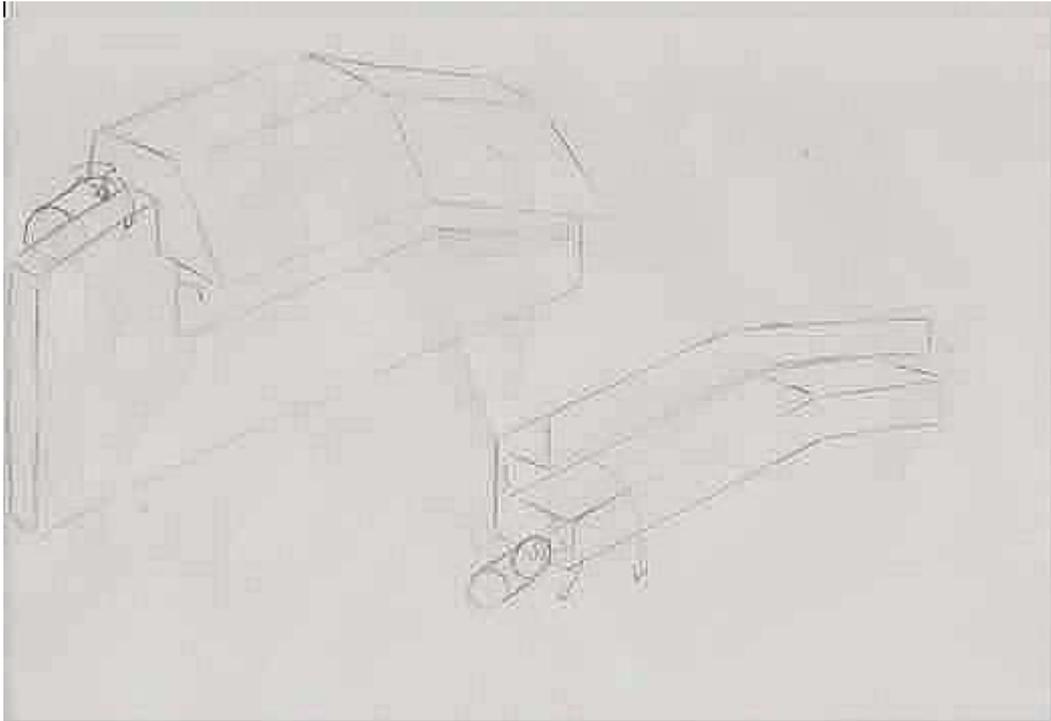


Figura 04 – Figura demonstrando a movimentação da calha. Fonte: Autores do projeto.

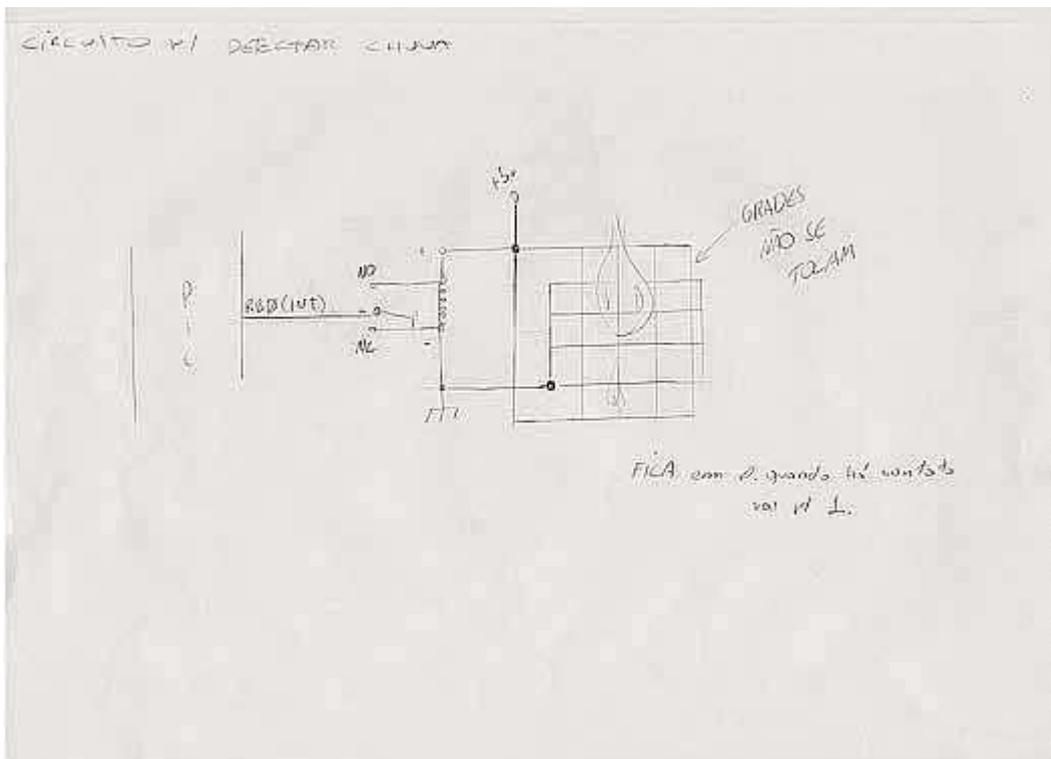


Figura 05 – Figura demonstrando o funcionamento do sensor de chuva. Fonte: Autores do projeto.

3.1 Projeto Mecânico

O motor escolhido para desenvolver o projeto foi o servo-motor, para que fosse possível determinar a posição desejada precisamente. Para poder controlar tal motor, geramos sinais do tipo PWM (Pulse-Width Modulation) e fixamos o motor no protótipo da calha de chuva, a qual foi feita de acrílico de 2mm de espessura, para que além de leve fosse tão funcional quanto calha de chuva convencional.



Figura 06 – Servo-motor utilizado da marca Futaba, modelo S3003. Fonte: Futaba®, link: <http://www.futaba-rc.com/servos/servos.html> .

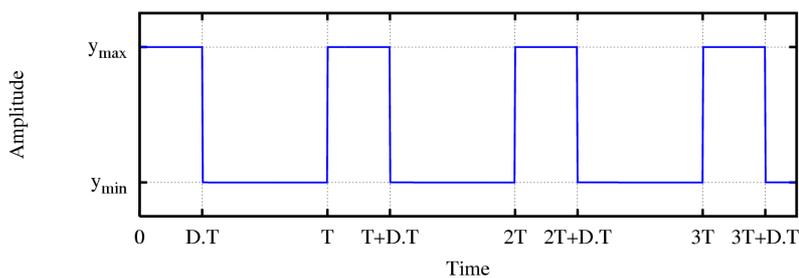


Figura 07 – Demonstração de um pulso PWM. Fonte: Desconhecida, link: <http://www.iua.upf.es/~jlozano/interfaces/interfaces5.html> .



Figura 08 – Servo-motor fixado na calha de chuva. Fonte: Autores do projeto.

Toda essa estrutura foi fixada no protótipo que simula a porta de um carro, cuja foi feita de madeira MDF branca, e o vidro feito de acrílico de 3mm de espessura, assim obtivemos a posição desejada do servo-motor e da calha de chuva para executar todas as funções necessárias.

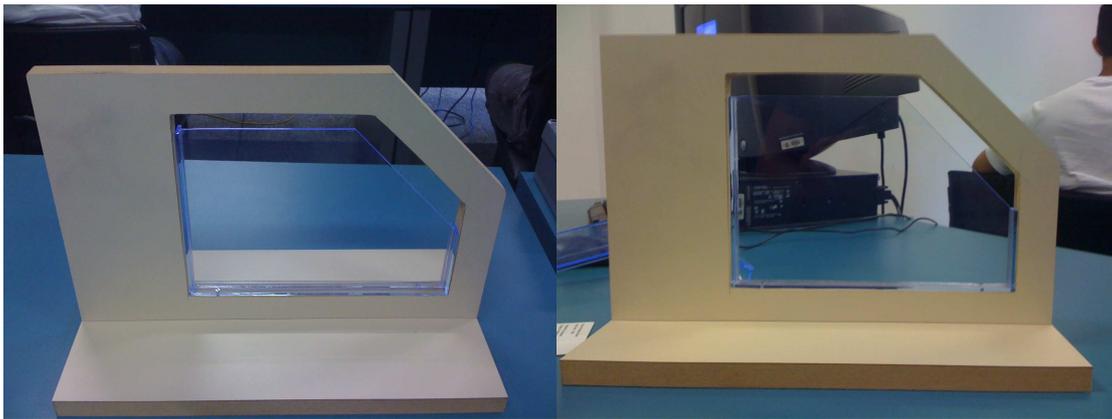


Figura 09 – Protótipo da porta de um carro feita em MDF. Fonte: Autores do projeto.

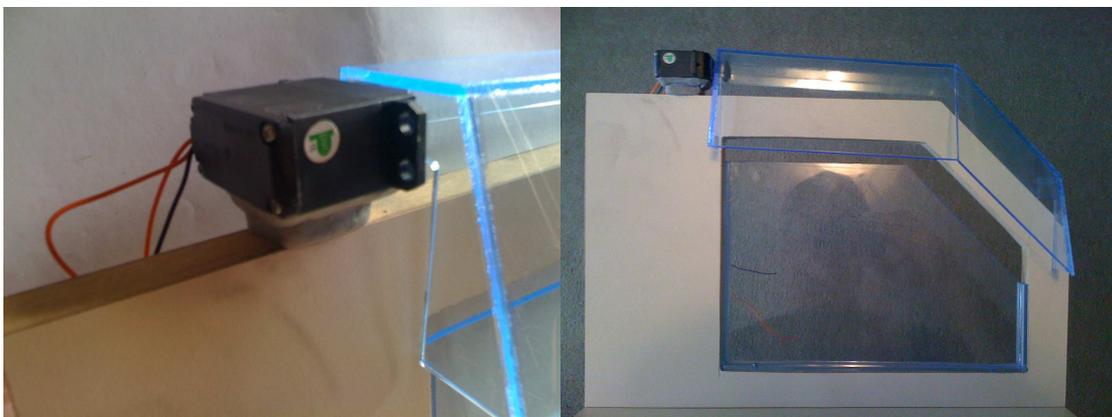


Figura 10 – Servo motor fixado na calha e na porta. Fonte: Autores do projeto.

3.1.1 Lista de Materiais Utilizados

Os materiais utilizados para concluir essa etapa do projeto foram:

- Servo-motor Futaba modelo s3003;
- Acrílico de 2mm de espessura;
- Acrílico de 3mm de espessura;
- Parafusos;
- Pregos;
- Durepox.

3.2 Projeto do Hardware

Toda a parte de elaboração do hardware foi desenvolvido utilizando-se do software EAGLE 5.6 da empresa CadSoft. Para o funcionamento do projeto foi necessário apenas uma placa, que contém o processador MICROCHIP™ modelo PIC16F877A, dois push-buttons, para a ativação manual de subida e descida da calha de chuva, um pino de saída que gera o sinal PWM para o servo e dois pinos para serem usados pelo sensor de chuva, o qual foi feito manualmente.

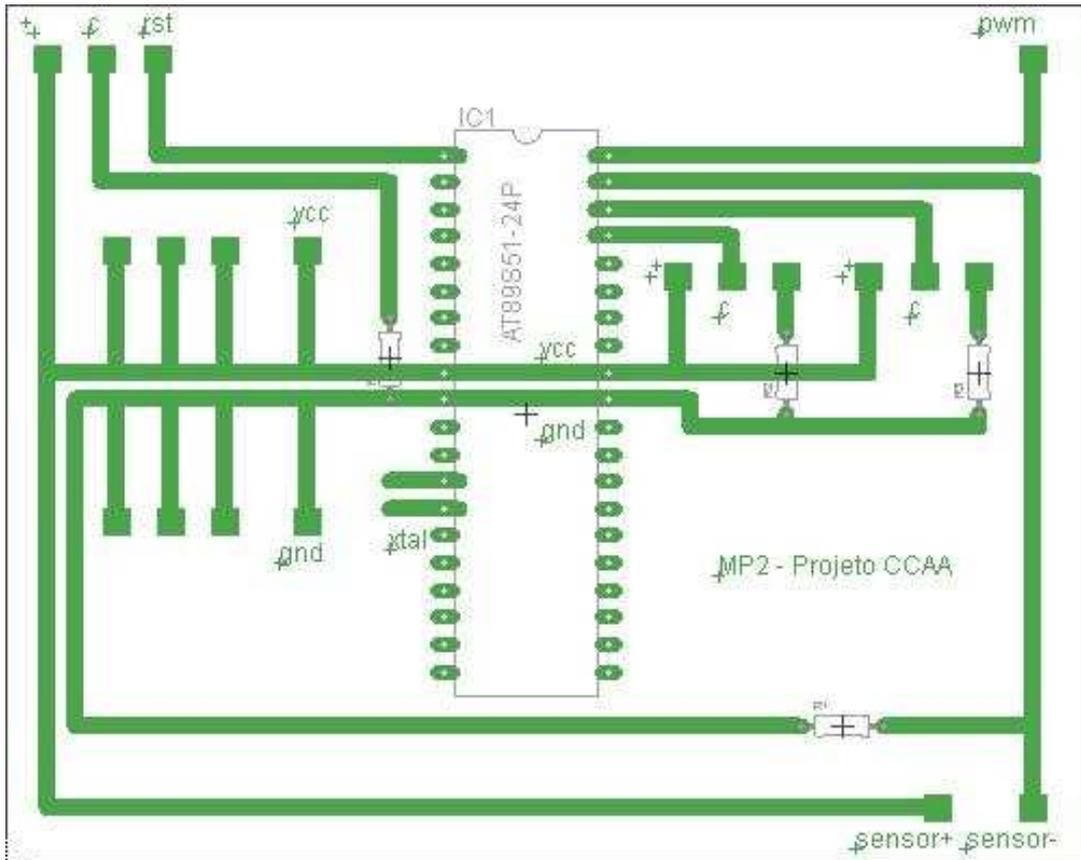


Figura 11 – Confecção da placa no software EAGLE 5.6. Fonte: Autores do projeto.

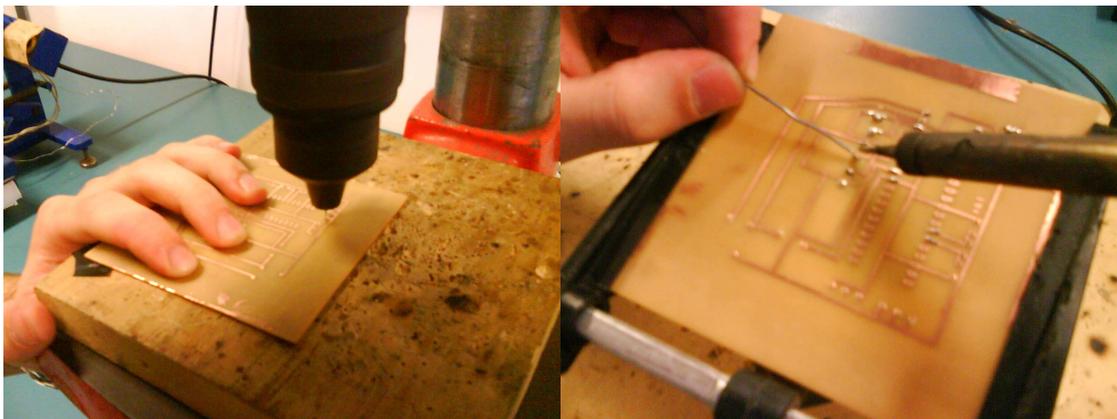


Figura 12 – Confecção da placa. Fonte: Autores do projeto.

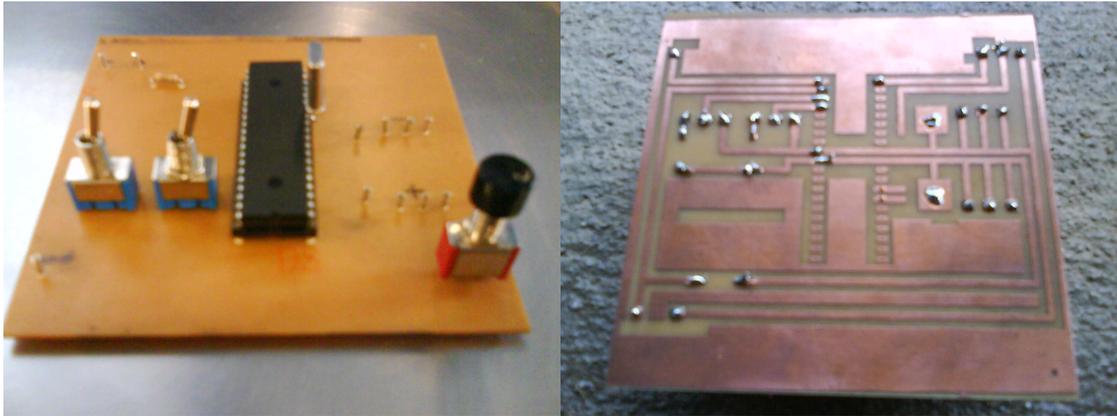


Figura 13 – Imagem da parte superior e inferior da placa confeccionada. Fonte: Autores do projeto.



Figura 14 –sensor de chuva concluído. Fonte: Autores do projeto.

O sensor de chuva funciona da seguinte maneira, são duas malhas de fios condutores postas bem próximas uma da outra, uma destas malhas está conectada ao terra e ao PIC, gerando sinal zero (nível lógico baixo), e a outra malha conectada ao VCC de 5V (nível lógico alto). Quando houver a passagem de água, ocorre a ligação entre essas malhas, fazendo a corrente percorrer as mesmas gerando sinal um ao processador, logo o mesmo entende como detecção de chuva.

3.2.1 Lista de Materiais Utilizados

Os materiais utilizados para concluir essa etapa do projeto foram:

- Placa de fenolite 10cm x 10cm;
- 3 push-buttons: NC-C-NA;
- Processador MICROCHIP™ modelo PIC16F877A;
- Resistor 4,5 M Ω ;
- 3 Resistores de 470 Ω ;
- Barra torneada de pinos fêmea;
- Estanho para solda.

3.3 Projeto do Software

Toda a parte lógica de programação foi desenvolvida na linguagem Assembly, utilizando o MPLAB IDE v8.36 da empresa MICROCHIP™ que é a mesma fornecedora do PIC utilizado, o PIC16F877A.

O software tinha que atender a todos os requisitos estipulados, os quais eram:

- Controle manual para descer a calha de chuva;
- Controle manual para subir a calha de chuva;
- Detecção automática da presença chuva.

O principal desafio foi gerar o sinal PWM correto, além de poder manipulá-lo na hora que fosse necessário. Para isso nos utilizamos da interrupção timer2, a qual contém um

módulo de 256 posições. Para que pudéssemos “setar” de maneira correta todas essas posições do módulo, foi necessário chamar essa interrupção à um frequência de 12.800Hz, assim o ciclo completo de configuração do módulo ocorreria a 50 Hz, que é o valor apropriado para o uso do servo-motor, após as configurações do fosc, post-scale/pre-scale, PR2 e a escolha de um cristal de oscilação de 4 MHz, obtivemos um valor bem próximo, 49.5 Hz.

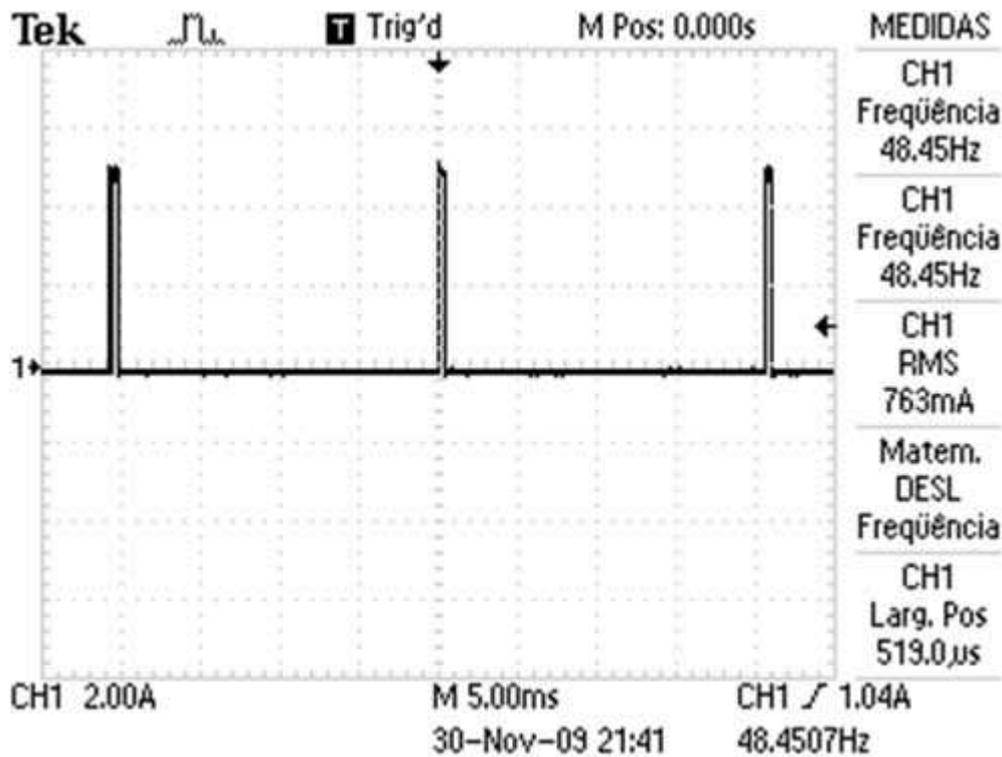


Figura 15 – Sinal PWM obtido na medição do Osciloscópio. Fonte: Autores do projeto.

Após tudo isso, a função geradora do sinal PWM, seta as posições do módulo em um (nível lógico alto) ou zero (nível lógico baixo) através de um registrador chamado ciclo_pwm, assim gerando o ciclo desejado, o qual pode ser modificado a qualquer momento, isto é, podendo modificar a posição inicial do servo-motor.

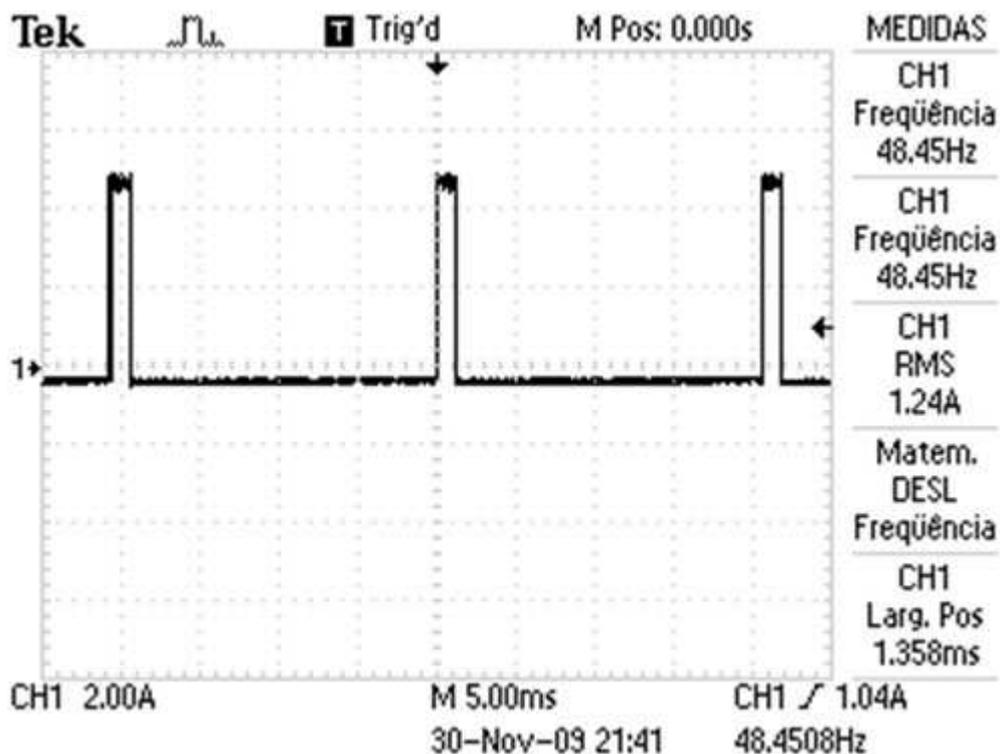


Figura 16 – Sinal PWM modificado, obtido na medição do Osciloscópio. Fonte: Autores do projeto.

Concluindo essa etapa, tínhamos o essencial para iniciar a criação das funções necessárias, para isso foi feito um loop que constantemente checa as portas utilizadas para as três funções, as quais são de subir e descer a calha manualmente e detectar a presença de chuva automaticamente através do sensor.

A função de descida manual opera da seguinte forma, quando o sinal lido é um (nível lógico alto), então há ativação manual para descer, esta função vai alterar o valor do ciclo_pwm usado na função geradora de PWM, logo com o ciclo variando, a posição do servo-motor varia, porém antes de ir para a posição desejada, é checado se a calha de chuva já está na mesma posição, se sim o programa ignora e volta para o loop, se não, vai para a posição desejada.

A função de subida manual opera da mesma forma que a de descida, a única diferença é apenas o valor do ciclo_pwm, que é alterado para um valor diferente para que a calha de chuva possa ir para a posição correta.

A função para a detecção automática de chuva através do sensor opera da seguinte forma, quando o valor lido na porta é um (nível lógico alto), então se inicia um contador que checa quantas vezes foi detectada a presença de chuva e se inicia uma contagem que restringe o tempo entre as mesmas, se for maior que o estipulado, essas detecções são descartadas, se for dentro do tempo estipulado, irá se confirmar a presença de chuva e a calha de chuva irá descer. Após isso, se inicia outra contagem para monitorar se ainda há detecção de chuva, se esta contagem for maior que o valor estipulado, então não há mais presença de chuva, logo a calha de chuva irá subir para a posição inicial, caso seja dentro do valor estipulado, a calha se mantém na posição sabendo que ainda há chuva.

3.3.1 Código Fonte

```
:: Projeto CCAA  
:: Microprocessadores 2  
:: Prof. Afonso Miguel  
:: Rafael da S. N Serrano e Luiz Renato H. S
```

```
#include "p16F877a.inc"  
#define pin_pwm          PORTB,7  
  
__CONFIG    3F31  
CBLOCK     0x20  
WORK_TEMP  
STATUS_TEMP  
FSR_TEMP  
PCLATH_TEMP  
ciclo_pwm  
aux_timer  
conta_chuva  
checa_tempo  
checa_tempo2  
timer  
timer2
```

ENDC

```
org      0
goto    main
org      4
```

```
call    push
bcf     PIR1,TMR2IF
clrf   TMR2
call    pwm
call    pop
retfie
```

```
push:
movwf  WORK_TEMP
swapf  STATUS,W
movwf  STATUS_TEMP
movf   FSR,W
movwf  FSR_TEMP
movf   PCLATH_TEMP,W
clrf   PCLATH
clrf   STATUS
return
```

```
pwm:
BANKSEL aux_timer
incf    aux_timer,F
movf    aux_timer,W
BANKSEL ciclo_pwm
xorwf   ciclo_pwm,W
BANKSEL STATUS
btfss   STATUS,Z
goto    pulso0
BANKSEL aux_timer
movf    aux_timer,W
xorlw   .255
BANKSEL STATUS
btfss   STATUS,Z
goto    pulso1
```

return

pulso1:

BANKSEL aux_timer

clrf aux_timer

BANKSEL PORTB

bsf PORTB,7

return

pulso0:

BANKSEL PORTB

bcf PORTB,7

return

pop:

movf PCLATH_TEMP,W

movwf PCLATH

movf FSR_TEMP,W

movwf FSR

swapf STATUS_TEMP,W

movwf STATUS

swapf WORK_TEMP,F

swapf WORK_TEMP,W

return

desce_calha_manual:

BANKSEL ciclo_pwm

movf ciclo_pwm,w

xorlw .16

BANKSEL STATUS

btfss STATUS,Z

goto desce_manual

goto loop

desce_manual:

BANKSEL ciclo_pwm

movlw .12

subwf ciclo_pwm,f

goto loop

desce_calha:

```
BANKSEL    ciclo_pwm
movf       ciclo_pwm,w
xorlw     .16
BANKSEL    STATUS
btfsc     STATUS,Z
goto      aguarda_para_descer
goto      loop
```

aguarda_para_descer:

```
BANKSEL    PORTB
btfss     PORTB,6
goto      aguarda_para_descer
```

desce:

```
BANKSEL    ciclo_pwm
movlw     .12
subwf     ciclo_pwm,f
```

continua_embaixo:

```
BANKSEL    checa_tempo2
movlw     .5
movwf     checa_tempo2
```

continua_embaixo2:

```
BANKSEL    PORTB
btfss     PORTB,6
goto      continua_embaixo3
goto      continua_embaixo
```

continua_embaixo3:

```
BANKSEL    aux_timer
Movf      aux_timer,w
xorlw     .0
BANKSEL    STATUS
btfss     STATUS,Z
goto      continua_embaixo2
goto      dec_timer2
dec_timer2:
```

```
BANKSEL    timer2
decfsz     timer2,f
goto       continua_embaixo2
BANKSEL    timer2
movlw     .50
movf       timer2
decfsz     checa_tempo2,f
goto       continua_embaixo2
goto       zera_tudo2
```

zera_tudo2:

```
BANKSEL    conta_chuva
movlw     .10
movwf     conta_chuva
movlw     .50
movwf     timer2
movlw     .5
movwf     checa_tempo2
goto      sobe_calha
```

sobe_calha_manual:

```
BANKSEL    ciclo_pwm
movf       ciclo_pwm,w
xorlw     .28
BANKSEL    STATUS
btfss     STATUS,Z
goto      sobe_manual
goto      loop
```

sobe_manual:

```
BANKSEL
ciclo_pwm
movlw     .12
addwf     ciclo_pwm,f
goto      loop
```

sobe_calha:

```
BANKSEL    ciclo_pwm
movf       ciclo_pwm,w
```

```

xorlw      .28
BANKSEL   STATUS
btfsc    STATUS,Z
goto     sobe
goto     loop
sobe:
BANKSEL   ciclo_pwm
Mowlw    .12
addwf    ciclo_pwm,f
goto     loop

configura:
BANKSEL   TRISB
movlw    B'01110000'
movwf    TRISB

BANKSEL   PORTB
clrf     PORTB

BANKSEL   OPTION_REG
movlw    B'11001111'
movwf    OPTION_REG

BANKSEL   PIE1
movlw    B'00000010'
movwf    PIE1

BANKSEL   PR2
movlw    .5
movwf    PR2

BANKSEL   INTCON
movlw    B'01000000'
movwf    INTCON

BANKSEL   T2CON
movlw    B'00011011'
movwf    T2CON

```

```

BANKSEL    ADCON1
movlw     B'00001110'
movwf     ADCON1

BANKSEL    ADCON0
movlw     B'00000101'
movwf     ADCON0

BANKSEL    ciclo_pwm
movlw     .28
movwf     ciclo_pwm
movlw     .0      ;
movwf     aux_timer
movlw     .10
movwf     conta_chuva
movlw     .50
movwf     timer
movlw     .50
movwf     timer2
movlw     .3
movwf     checa_tempo
movlw     .5
movwf     checa_tempo2
return

main:
call      configura
BANKSEL   INTCON
bsf       INTCON,GIE
BANKSEL   T2CON
bsf       T2CON, TMR2ON

loop:
BANKSEL   PORTB
btfss    PORTB,6
goto     loop2
goto     possivel_chuva1
loop2:
btfss    PORTB,4

```

```
goto    loop3
goto    desce_calha_manual
```

```
loop3:
btfss   PORTB,5
goto    loop
goto    sobe_calha_manual
```

```
possivel_chuva1:
BANKSEL   checa_tempo
movlw    .3
movwf    checa_tempo
BANKSEL   conta_chuva
decfsz   conta_chuva,f
goto     cont_possivel_chuva1
goto     desce_calha
```

```
cont_possivel_chuva1:
BANKSEL   PORTB
Btfss    PORTB,6
goto     cont_possivel_chuva1
```

```
possivel_chuva2:
BANKSEL   PORTB
btfss    PORTB,6
goto     cont_possivel_chuva2
goto     possivel_chuva1
```

```
cont_possivel_chuva2:
BANKSEL   aux_timer
movf     aux_timer,w
xorlw    .0
BANKSEL
STATUS
btfsc    STATUS,Z
goto     possivel_chuva2
goto     dec_timer
dec_timer:
BANKSEL   timer
```

```
decfsz    timer,f
goto      possivel_chuva2
BANKSEL   timer
movlw    .50
movf     timer
decfsz   checa_tempo,f
goto     possivel_chuva2
goto     zera_tudo
```

zera_tudo:

```
BANKSEL   conta_chuva
movlw    .10
movwf   conta_chuva
movlw    .50
movwf   timer
movlw    .3
movwf   checa_tempo
goto     loop
END
```

4. CONCLUSÃO

Com a finalização deste projeto, foi possível interagir todos os conhecimentos adquiridos nas mais variadas disciplinas presentes no curso de Engenharia de Computação.

A sincronia do grupo durante a execução do projeto foi um fator primordial na conclusão integral dos tópicos previamente estipulados. A divisão de tarefas foi essencial para se ter um bom gerenciamento do tempo e prova que seguir um cronograma é o ideal para que o projeto seja entregue na data previamente estipulada.

5. REFERÊNCIAS

MIGUEL, Afonso Ferreira. Módulo 16F877A. Disponível em: <http://www.afonsomiguel.com/MicrochipPIC>. Acesso em: 25 de Outubro de 2009.

PIC16F877A. Microchip, disponível em: <http://www.microchip.com/wwwproducts/Devices.aspx?dDocName=en010242>. Acesso em: 25 de Outubro de 2009.