

Plataforma para processamento de imagem com dsPIC com exemplo para reconhecimento de caractere usando rede neural.

Trabalho de Conclusão de Curso

Engenharia da Computação

Autor: Ismael Carvalho Mascarenhas
Orientador: Prof. Dr. Sergio Campello Oliveira



UNIVERSIDADE
DE PERNAMBUCO

Ismael Carvalho Mascarenhas

Plataforma para processamento de imagem com dsPIC com exemplo para reconhecimento de caractere usando rede neural.

Monografia apresentada como requisito parcial para obtenção do diploma de Bacharel em Engenharia da Computação pela Escola Politécnica de Pernambuco – Universidade de Pernambuco.

Recife, Dezembro de 2010.

Dedico aos meus Pais.

Agradecimentos

Agradeço primeiramente a Deus por tudo, agradeço a meus pais por toda orientação que me deram, a meu orientador pela paciência, aos outros professores do curso por todo conhecimento repassado e a meus colegas de faculdade que me acompanharam na caminhada.

Resumo

Para o processamento de imagens existem varias abordagem com relação ao hardware utilizado. Os mais comuns são o uso de computadores, FPGAs e DSPs. Para atender as necessidades de novas aplicações, a utilização do processamento de imagens vem crescendo a cada dia, necessitando cada vez mais de novos algoritmos, aplicações e novas abordagens. Tendo como objetivo cumprir esses requisitos foi criado neste trabalho uma plataforma para processamento digital de imagem utilizando como hardware para o processamento o dsPIC 33FJ16GP304 um microcontrolador DSP de baixo custo junto com uma aplicação para o reconhecimento de caracteres utilizando rede neural. Para poupar recursos, foi utilizado a abordagem de processamento de imagem em tempo real sem o armazenamento da mesma. Para a validação da plataforma, foram realizados testes em que foram apresentadas imagens à plataforma e como resultados alcançados o reconhecimento dos caracteres presentes na imagem mesmo a imagem estando com ruído ou faltando píxeis. O principal objetivo deste trabalho é a junção de processamento de imagens por rede neural em sistemas embarcados de baixo custo.

Abstract

There are several approach with the hardware for image processing such as computers, FPGAs, and DSPs. The Use the Image processing is growing ever because of the news applications, requiring new algorithms and new approaches. For these goal we created a platform for digital image processing a microcontroller dsPIC 33FJ16GP304, that presents a low cost DSP with an application for character recognition by Network Neural. In order to save resources, we used an real time image processing approach. To validate the platform was conducted tests in which the images were presented as platform and results had the recognition of characters in the image even the image being noisy or missing pixels. The main goal this work was the combination of image processing by neural network with low cost embedded systems.

Sumário

| | |
|--|-----------|
| CAPÍTULO 1 | 1 |
| INTRODUÇÃO | 1 |
| 1.1 ESTRUTURA DO DOCUMENTO | 2 |
| CAPÍTULO 2 | 3 |
| 2.1 DSPs..... | 3 |
| 2.2 DSPIC | 4 |
| 2.3 dsPIC 33FJ16GP304..... | 5 |
| 2.4 PIC18F4550 | 7 |
| 2.5 DISPLAY LCD | 7 |
| 2.6 RECONHECIMENTO DE CARACTERES | 7 |
| 2.7 REDES NEURAIS..... | 9 |
| 2.8 ADALINE..... | 10 |
| CAPÍTULO 3 | 11 |
| 3.1 DIAGRAMA..... | 11 |
| 3.2 CIRCUITO..... | 12 |
| 3.3 ENVIO DA IMAGEM..... | 16 |
| 3.4 PROCESSAMENTO DA IMAGEM | 17 |
| 3.5 RESULTADOS | 21 |
| CONCLUSÃO E TRABALHOS FUTUROS..... | 24 |
| 4.1 CONCLUSÃO | 24 |
| 4.2 TRABALHO FUTUROS | 25 |
| BIBLIOGRAFIA..... | 26 |
| APÊNDICE A CÓDIGO FONTE DO DSPIC 3FJ16GP304 | 27 |
| APÊNDICE B..... | 29 |
| CÓDIGO FONTE DO PIC18F4550..... | 29 |

Índice de Figuras

| | | |
|-------------------|---|----|
| Figura 1. | Diagrama de processamento de sinais. | 4 |
| Figura 2. | Arquitetura Dspic | 6 |
| Figura 3. | Instruções dsp do dsPIC. | 6 |
| Figura 4. | Display LCD 16x2 | 8 |
| Figura 5. | Neurônio McCulloch e Pittis..... | 9 |
| Figura 6. | Diagrama da plataforma | 11 |
| Figura 7. | Diagrama elétrico da plataforma..... | 13 |
| Figura 8. | Visão superior 3D da plataforma. | 15 |
| Figura 9. | Vista inferior 3D da plataforma. | 15 |
| Figura 10. | Aplicativo em Java para enviar imagem..... | 17 |
| Figura 11. | Código PIC18F4550. | 18 |
| Figura 12. | Conjunto de treinamento letras T e H. | 19 |
| Figura 13. | Erro no processo de aprendizagem. | 19 |
| Figura 14. | a) Teste com a letra t b) Teste com a letra h. | 22 |
| Figura 15. | a) Teste com a letra t com ruído. B) Teste da letra h com ruído. | 23 |
| Figura 16. | a) Letra t incompleta. B) Letra h incompleta. | 23 |

Índice de Tabelas

| | |
|--|----|
| Tabela 1. Matriz de pesos da rede neural..... | 20 |
|--|----|

Tabela de Símbolos e Siglas

ADC – Conversor analógico digital

DAC – Conversor digital analógico

DSP – Processador digital de sinais

dsPIC - Microcontrolador dsp da Microchip

RNA – Rede Neural Artificial

TI – Texas Instruments

Capítulo 1

Introdução

A utilização do processamento digital de imagens vem crescendo continuamente [1] com a abertura de novas áreas, como também no melhoramento de processos em áreas já conhecidas. Entre essas áreas, destaca-se o melhoramento de imagens, a compressão de imagens e a visão computacional com a criação de algoritmos capazes de detectar objetos e o movimento de objetos. O processamento digital de imagens pode ser aplicado em áreas como: segurança, médica [2] e na indústria.

Sistemas embarcados são sistemas dedicados para uma determinada atividade, geralmente possuem um microprocessador. Diferenciam-se de computadores, que são sistemas genéricos, por serem compactos e disponibilizarem menos recursos de hardware. São exemplos de sistemas embarcados: Semáforos de trânsito, celulares e uma plataforma para monitoração corrente de fuga [3].

Para a realização do processamento de sinais, podem ser usados os DSPs que são processadores desenvolvidos visando operações mais usadas em processamento digital de sinais, assim possuem instruções específicas para este propósito, que aliadas à alta capacidade de processamento de instruções por segundo [4], viabilizam o processamento de sinais como imagens [5].

Para processamento embarcado de imagem também, são utilizados FPGAs. FPGAs são dispositivos de lógica programável, que comparados aos DSPs possuem um custo mais elevado e dificuldade em cálculos de ponto flutuante. Porém, pelo poder de definição da arquitetura, possuem maior velocidade de processamento. Existem algumas propostas para o uso de FPGA com DSP para processamento de imagens, como apresentado em [6], no qual o FPGA por ter alta velocidade e flexibilidade é utilizado para amostrar e mostrar a imagem e o DSP realiza o processamento da imagem.

Visando processar imagem de forma remota e com baixo custo, foi desenvolvida neste trabalho uma plataforma que possui a capacidade de receber uma imagem por uma porta de comunicação serial TTL RS-232, que pode ser oriunda de uma câmera, por exemplo, com comunicação serial e ser processada em tempo real, ou seja, sem o armazenamento no DSP. Para o uso em computadores com porta USB, foi incluído um microcontrolador auxiliar PIC18F4550 para fazer a conversão, tanto para uso como conversor USB-Serial para aquisição de imagem na plataforma como também para a plataforma servir de Dispositivo de Interface Humana HID.

Como microcontrolador para o processamento de imagem, é utilizado o dsPIC 33FJ16GP304, um microcontrolador de baixo custo fabricado pela Microchip. Para o teste da plataforma foram realizados testes para o reconhecimento de caracteres por redes neurais. Assim foi criada uma plataforma para o processamento de imagem embarcada, capaz de processar imagem sem a necessidade do uso de um computador.

A aplicação de reconhecimento de caractere pode ser útil para o reconhecimento de placas de automóveis. Outra aplicação para a plataforma é a criação de um mouse ocular, que pode ser controlado pela posição da Iris.

1.1 Estrutura do documento

Este trabalho está dividido em 4 capítulos, O primeiro contém a parte introdutória. O Capítulo 2 contém uma revisão das ferramentas e componentes utilizados como DSP, dsPIC e Rede Neural. O Capítulo 3 apresenta o desenvolvimento da plataforma, com as abordagens usadas e os resultados e por fim O Capítulo 4 apresenta as conclusões e os trabalhos futuros sugeridos.

Capítulo 2

2.1 DSPs

Processadores de Sinais Digitais (DSPs) são microprocessadores com melhor capacidade de processamento de sinais comparado aos microprocessadores para aplicações genéricas quando ambos possuem o mesmo clock. Inicialmente Lançado pela Bell Labs em 1979, o MAC 4 foi o primeiro DSP a ser disponibilizado no mercado. Para suprir uma necessidade do mercado por microprocessadores dedicados a sinais. As primeiras aplicações foram na área de cancelamento de ruído em áudio.

Os DSPs mais populares são os da Texas Instruments ® (TI), que tiveram a sua produção iniciada em 1983 com o TMS32010. Atualmente existem outros fabricantes como: Microchip, Motorola, AT&T, NEC e Altera [7].

No mercado atual existem DSPs de variados modelos desde os mais simples, com menor capacidade de processamento, aos modelos com maior capacidade como o C6000 da Texas que possui clock de 1GHz . O preço dos DSPs variam de R\$ 5,00 até R\$ 800,00.

A aplicação principal dos DSPs é o processamento em tempo real de sinais como áudio e vídeo. Para isto, os DSPs são desenvolvidos com a capacidade de executar operações matemáticas mais comuns em processamento de sinais que são adição, multiplicação e transferência de memória consecutiva.

Instruções como *Multiply-Accumulate* (MAC) $A = A + (X * Y)$, que é a realização da multiplicação de dois números junto com soma do valor presente no acumulador, podem ser executadas em um ciclo de máquina. Assim, operações como a transformada rápida de Fourier podem rapidamente calculada. A transformada rápida de Fourier é um conjunto algoritmos para cálculo da transformada discreta de Fourier e sua inversa os quais dependem fortemente da instrução MAC. A

transformada rápida de Fourier é bastante utilizado em processamento de sinais por permitir a resolução de equações diferenciais.

Outra utilização da instrução MAC é em redes neurais em que são realizadas sucessivas multiplicações do peso do neurônio com as entradas dos mesmos e o resultado é somado com o acumulador. Assim utilizando um microcontrolador DSP é realizado em um ciclo de máquina o que em microcontroladores genéricos levaria mais instruções conseqüentemente mais tempo para ser executada.

Um exemplo típico do processamento de sinais digitais é mostrado na Figura 1, nela tem um ADC que é um conversor analógico digital que transforma um sinal analógico em digital (bits) depois é processado pelo DSP e, logo em seguida, passa pelo DAC que é um conversor digital analógico. Este é um exemplo típico de um filtro contra ruídos.



Figura 1. Diagrama de processamento de sinais.

2.2 DSPIC

Os dsPIC são microcontroladores DSPs fabricados pela Microchip, uma empresa Norte americana que é líder na fabricação de microcontroladores com a arquitetura RISC de 8 e 16 bits.

Os dsPIC possuem a arquitetura Harvard na qual o caminho de dados é diferente do caminho de dados de programa. São de 16 bits e diferenciam da família 24H que também são de 16 bits por possuírem no núcleo central uma central para processamento de instruções DSP e 2 acumuladores de 40 bits para instruções MAC. Outras características que também são comuns aos da família 24H é a presença de 16 registradores de 16bits (w0 – w15) e hardware dedicado para multiplicação e divisão.

A arquitetura dos microcontroladores dsPIC da família 33F é apresentada na Figura 2. No núcleo central tem a *engine dsp*, responsável por realizar as operações envolvendo instruções dsp, para a execução de instruções comuns o núcleo contém uma ALU. No núcleo também contem um banco de 16 registradores de 16 bits, um hardware dedicado para multiplicação e divisão, 2 acumuladores de 40 bits e outros componentes utilizado para execução de instruções.

Na parte inferior da Figura 2, tem diversos periféricos como porta serial Rs-232, CAN e SPI conversores ADC e DAC de alta velocidade assim como também timers e geradores de PWM para permitir uma melhor utilização dos dsPICs provendo comunicação externa e recursos extras para serem utilizados em software. Na parte direita da Figura 2 está apresentada tem a memória RAM e na parte esquerda a memória de programa.

Atualmente existem duas famílias de dsPIC, a 30F e a 33F. A 30F chega a ter a capacidade de processar 30 MIPS (Milhões de instruções por segundo – medida de desempenho utilizada em informática) e alguns modelos possuem EEPROM interna (do inglês, *Electrically-Erasable Programmable Read-Only Memory*) que é uma memória não volátil para armazenamento de dados. Outro periférico presente é o controle integrado de motor. A família 33F chega a ter 40 MIPS e é própria para ser utilizada como filtro digital.

As principais instruções para o processamento digital de sinais pode ser visto na Figura 3. Um exemplo é a instrução ED (do inglês, Euclidean distance) apresentada na equação (2.1). Nela o quadrado da subtração de dois números é realizado em apenas um ciclo de maquina. Esta operação é útil em calculo de erro médio quadrático e calculo de distâncias.

$$ED = (x-y)^2 \quad (2.1)$$

2.3 dsPIC 33FJ16GP304

O dsPIC 33FJ16GP304 é um microcontrolador fabricado pela Microchip da linha DSP com o caminho de dados 16 bits e caminho de instruções de 24 bits.

Possui memória para programa de 16KB e memória RAM de 2KB, consegue operar em até 40MIPS, Sua tensão de alimentação é entre 3 e 3,6 volts e possui oscilador interno. Possui como periférico 1 UART, 1 SPI e 1I2c. Uma importante característica é a sua capacidade de permitir que os pinos de saída sejam configurados, os pinos para comunicação serial pode ser escolhido entre os pinos das diversas portas, existe mais de uma opção para escolha dos pinos que será ligado o cristal externo. Isto facilita o roteamento da placa.

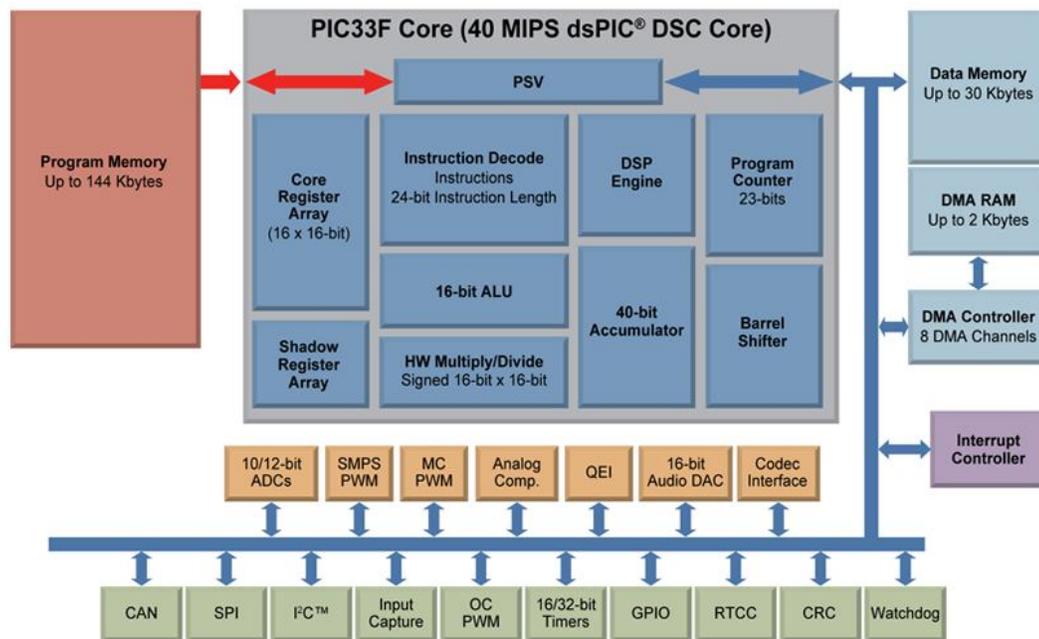


Figura 2. Arquitetura Dspic [8].

| Instrução | Operação |
|-----------|---------------------|
| CLR | $A=0$ |
| ED | $A = (x - y)^2$ |
| EDAC | $A = A + (x - Y)^2$ |
| MAC | $A = A + (X * Y)$ |
| MAC | $A = A + 2X$ |
| MOVSAC | Não muda A |
| MPY | $A = x * y$ |
| MPY | $A = 2X$ |
| MPY.N | $A = -X * Y$ |
| MSC | $A = A - X * Y$ |

Figura 3. Instruções dsp do dsPIC.

2.4 PIC18F4550

O PIC 18F4550 é um microcontrolador fabricado pela Microchip, da linha de 8 bits, possui como principal periférico um porta USB Full Speed de capacidade de transferência de até 12Mbps. Chega a alcançar até 12 MIPS possui 32 KB de memória de programa e 2 KB de memória RAM. Possui periféricos como 1 UART e 1 SPI/I2C.

2.5 Display LCD

Displays de cristal liquido, acrônimo de LCD (do inglês, *liquid crystal display*) são amplamente utilizados em sistemas embarcados, como telefones públicos, maquinas de café expresso, para mostrar informações ao usuário.

Seu funcionamento consiste de um liquido polarizador de luz, controlado eletricamente, que se encontra comprimido dentro de pixels entre duas lâminas transparentes. Existem vários modelos de LCD, as diferenças entre os modelos são a quantidade de linhas, quantidade de colunas e o modelo do controlador responsável pelo controle do display. Um exemplo é apresentado na Figura 4, o qual possui duas linhas cada uma com 16 caracteres conhecido como display 16x2.

A maioria dos displays 16x2 utilizam um controlador compatível com o com o controlador HD44780. Na especificação deste controlador define que para o seu funcionamento é necessário que o display seja configurado, os comandos e dados enviado são de 8bits enviados através dos pinos d0 à d7 do display, pode ser utilizado a opção de envio de 4 bits e depois os outros 4 bits, utilizando assim os pinos de d4 à d7.

2.6 Reconhecimento de caracteres

Para que a imagem seja processada e extraído o caractere, é necessário a sua digitalização e depois a sua discretização que determina a resolução da imagem. Depois a imagem é colocada no formato de uma matriz $M \times N$, onde cada

ponto representa um pixel da imagem. Cada pixel tem informação referente a luminosidade e a cor. Geralmente cada pixel é armazenado numa área da memória do sistema computacional que irá tratá-la.



Figura 4. Display LCD 16x2.

Para o reconhecimento de padrões é possível aplicar diversas técnicas como técnicas estatísticas, técnicas sintáticas e técnicas baseadas em redes neurais. As técnicas sintáticas precisam de um conjunto de atributos que são constantes para cada padrão. As regras sintáticas os padrões são expressos em operadores e primitivas unidos por regras que descrevem o relacionamento. Sua principal vantagem é necessitar de uma pequena amostra. As técnicas estatísticas assim como as baseadas em redes neurais precisam de um grande número de amostras. As técnicas estatísticas funcionam sobre modelos estatísticos os quais classificam os caracteres com informações obtidos pela manipulação de medidas puramente numéricas ou booleanas.

2.7 Redes Neurais

Redes neurais artificiais (RNA) começaram a ser estudadas na década de 40, o primeiro neurônio artificial foi proposto no artigo “A logical calculus of the ideas immanent in nervous activity” escrito por McCulloch e Pittis. O modelo é apresentado na Figura 5.

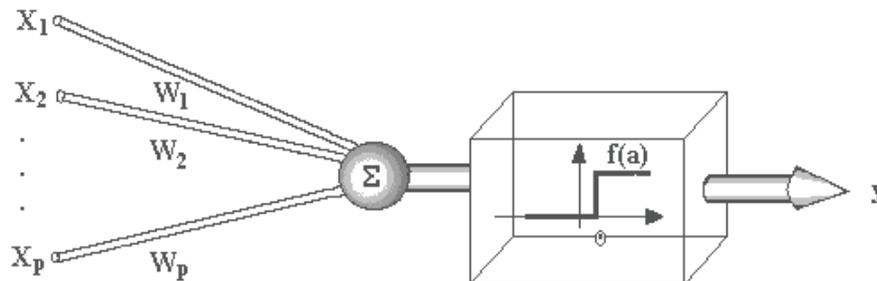


Figura 5. Neurônio McCulloch e Pittis.

RNAs são inspiradas nos neurônios biológicos e são constituídas de elementos chamados neurônios. O nível de conexão entre os neurônios é definido pelos pesos sinápticos. Um neurônio celular pode ser definido por um corpo mais as ramificações, os dendritos. Na região de sinapse, que é formada pelo axônio de um neurônio e os dendritos do outro, ocorre a passagem do impulso nervoso.

Para que um neurônio repasse o impulso nervoso, é necessário que a soma dos estímulos recebidos pelo neurônio ultrapasse um limiar, este processo é conhecido como a lei do tudo ou nada e o neurônio possui função de ativação degrau.

A influência de um neurônio no outro nas RNA DAE dada pelo peso sináptico entre eles quanto maior o peso mais um influenciaria o outro neurônio. O neurônio de McCulloch e Pittis tem a sua saída baseada na soma das entradas multiplicada pelos pesos, caso passe de um limiar a saída é ativa. Em uma RNA, o processo de treinamento é baseado no cálculo dos pesos dos neurônios, onde é buscado um conjunto de pesos que torne a rede neural capaz de classificar uma classe baseada nas entradas.

Assim, uma rede neural constitui uma ferramenta utilizada para resolução de problemas de classificação como o reconhecimento de caracteres. O processo de execução da rede neural constitui o somatório do produto das entradas pelos respectivos pesos e depois passar o valor pela função de ativação que determinará qual será a saída da rede neural.

2.8 Adaline

A rede neural do tipo Adalaine foi uma melhoria da Rede Perceptron por possuir saída com valores contínuos, foi apresentada por Widrow e Hoff. A diferença da Adaline para Perceptron esta no calculo da função de saída. A Perceptron possui a função degrau conhecida como tudo ou nada. Já a Adaline possui uma função continua, exemplo de funções de ativação continua utilizada: Linear (Equação 2.2), Sigmoidal Logística (Equação 2.3) e Tangente Hiperbólica (Equação 2.4). As equações são apresentadas a seguir:

$$f(net) = net, \quad (2.2)$$

$$f(net) = \frac{1}{1 + e^{-net}} \quad (2.3)$$

$$f(net) = \frac{e^{net} + e^{-net}}{e^{net} + e^{-net}} \quad (2.4)$$

Capítulo 3

Neste capítulo é apresentado como foi desenvolvida a plataforma e realizados os testes, começando pelo diagrama, o circuito, depois o envio da imagem, como é realizado o processamento e por ultimo os resultados alcançados.

3.1 Diagrama

Na Figura 6 é apresentado o diagrama da plataforma. Nela pode ser visto o elemento central de processamento que é o dsPIC 33FJ16GP304, o qual recebe uma imagem na porta serial, processa a mesma através de uma rede neural e tem como saída um display LCD 16x2. Para a aquisição da imagem é necessário uma câmera que disponha de comunicação serial RS-232, pelo fato de ser de difícil aquisição da câmera e por não ser vendida no Brasil, foi escolhida a abordagem da utilização de um PC (computador) mais um PIC18F4550 como conversor USB-Serial.

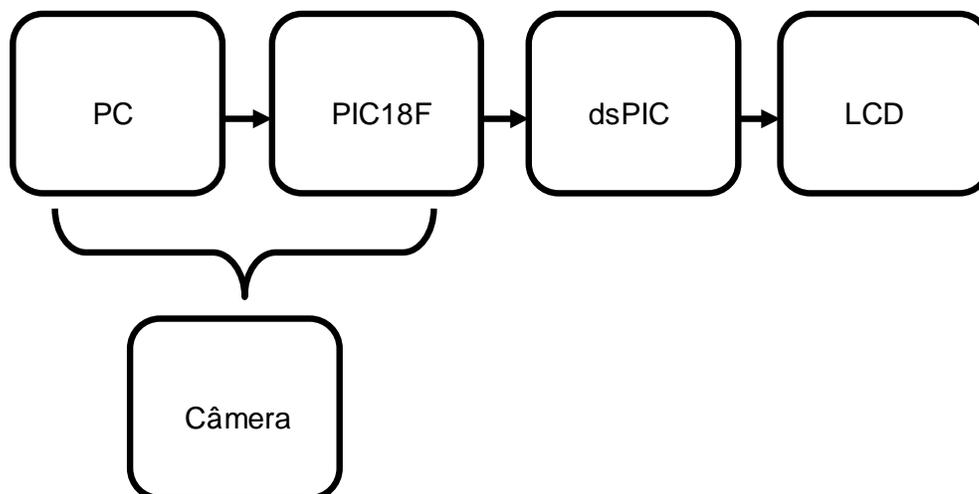


Figura 6. Diagrama da plataforma.

Com a utilização de um computador mais um PIC18F4550 como conversor simulando uma câmera serial, assim tem a possibilidade de a plataforma ser totalmente utilizada em ambiente computacional através do uso do simulador, pelo fato de não existir um componente conversor USB-Serial no simulador. Gerando facilidade no seu desenvolvimento para estudo.

3.2 Circuito

O diagrama elétrico da plataforma pode ser visto na Figura 7, nela tem os dois microcontroladores, o PIC18F4550 e o dsPIC 33FJ16GP304, as conexões com a porta USB e a conexão com um display 16x2, que informa qual foi caractere reconhecido pela rede neural.

A alimentação da plataforma é feita pelo conector J1, após o conector tem um diodo D1 para proteção caso a alimentação seja ligada invertida e também para isolar o circuito. A plataforma utiliza internamente a tensão de 3.3V para a sua alimentação. Para regular a tensão, é utilizado o CI regulador de tensão LM317 com base ajustável.

O Ajuste da tensão é feita pela equação 3.1. Neste caso $R2 = 390 \Omega$ e $R1 = 240 \Omega$, assim tendo como saída a tensão de 3,3 Volts. Para filtro são utilizados dois capacitores eletrolíticos C5 e C6 ambos de 10 uF x 16V.

$$v_o = \left(\frac{R2}{R1} + 1 \right) * 1,25 \quad (3.1)$$

Como característica de um circuito sequencial é necessário a geração de sinal de clock. No PIC18F4550 foi utilizado o oscilador interno com o uso de cristal e capacitores de filtros externos sendo X1, C1 e C2 respectivamente. Outra abordagem seria a utilização de cristal interno, a opção do uso do externo se deu pelo fato de ter uma maior precisão para a comunicação USB. O sinal de clock utilizado no dsPIC 33FJ16GP304 também é do oscilador interno com uso de cristal externo com dois capacitores de filtro sendo X2, C3 e C4 respectivamente.

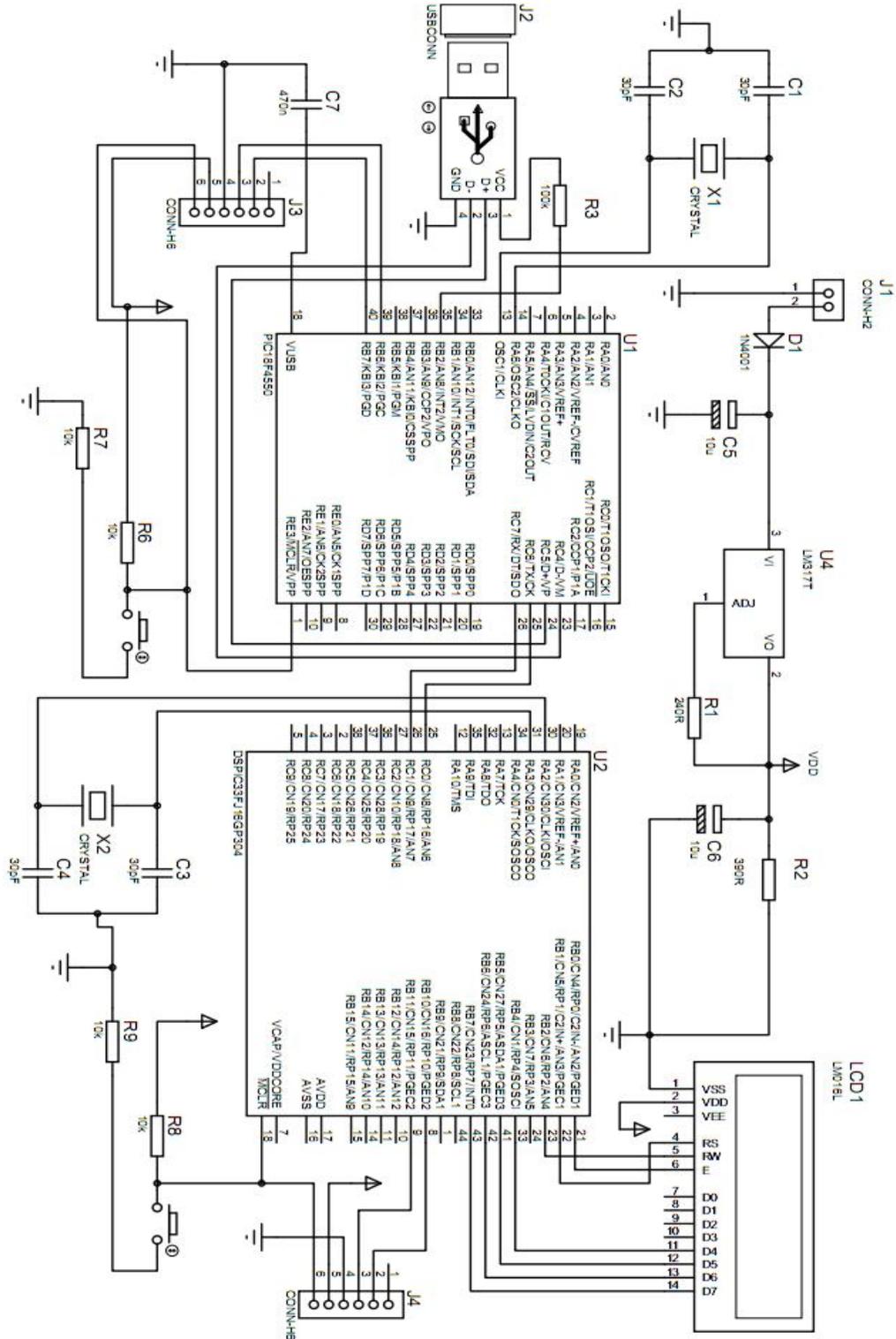


Figura 7. Diagrama elétrico da plataforma.

Caso seja necessário reset dos microcontroladores, são utilizados na plataforma dois circuitos para reset, um para o PIC18F e outro para o dsPIC33F. O circuito é igual para ambos, e é composto de dois resistores ($R6$ e $R7$ para o PIC18F e $R8$ e $R9$ para o dsPIC33F) para limitar a corrente, sendo os seus valores atribuídos pela especificação dada pela Microchip. Também é utilizado um botão, que ao ser acionado causa o reset na plataforma. O circuito de reset é ligado ao microcontrolador pelo pino de MCLR. Quando o pino MCLR é colocado em nível lógico 0, ocorre o reset do circuito.

Para a gravação do software que é executado dentro do PIC18F, a plataforma possui o conector ISCP J3 e para o dsPIC33F o conector ISCP J4. Os conectores seguem o padrão do gravador PICKIT 3. O PICKIT 3 é um gravador desenvolvido pela Microchip integrado ao ambiente de desenvolvimento Mplab, também desenvolvido pela Microchip com este gravador é possível também a realização da depuração do código.

O resistor R3 ligado ao conector USB J2 é utilizado para informar ao PIC18F que um dispositivo foi conectado. O capacitor C7 funciona como filtro para o circuito interno do microcontrolador que regula a tensão para a comunicação USB.

Para a ligação do display LCD 16x2 é utilizado a porta B do dsPIC. Para facilitar o roteamento na fase de desenvolvimento do layout da placa da plataforma. Foi optado pela utilização da transmissão de dados e comando do dsPIC para o LCD ser feita por uma via de 4 bits, sendo utilizados os pinos de RB4 à RB7 do dsPIC ligados aos pinos de D4 à D7 do LCD. Os pinos de RB0 à RB2 do dsPIC são ligados aos pinos do LCD: RS, RW e E respectivamente.

A Figura 8 mostra uma visão superior da plataforma, criada em ambiente computacional, nela pode ser vista a disposição dos componentes e as trilhas do circuito impresso.

A colocação do dsPIC na face inferior (Figura 9) do circuito foi feita para facilitar a sua construção em uma placa de face única, sendo a outra face construída através de *jumps*. Os conectores são colocados as bordas para facilitar a ligação dos mesmos.

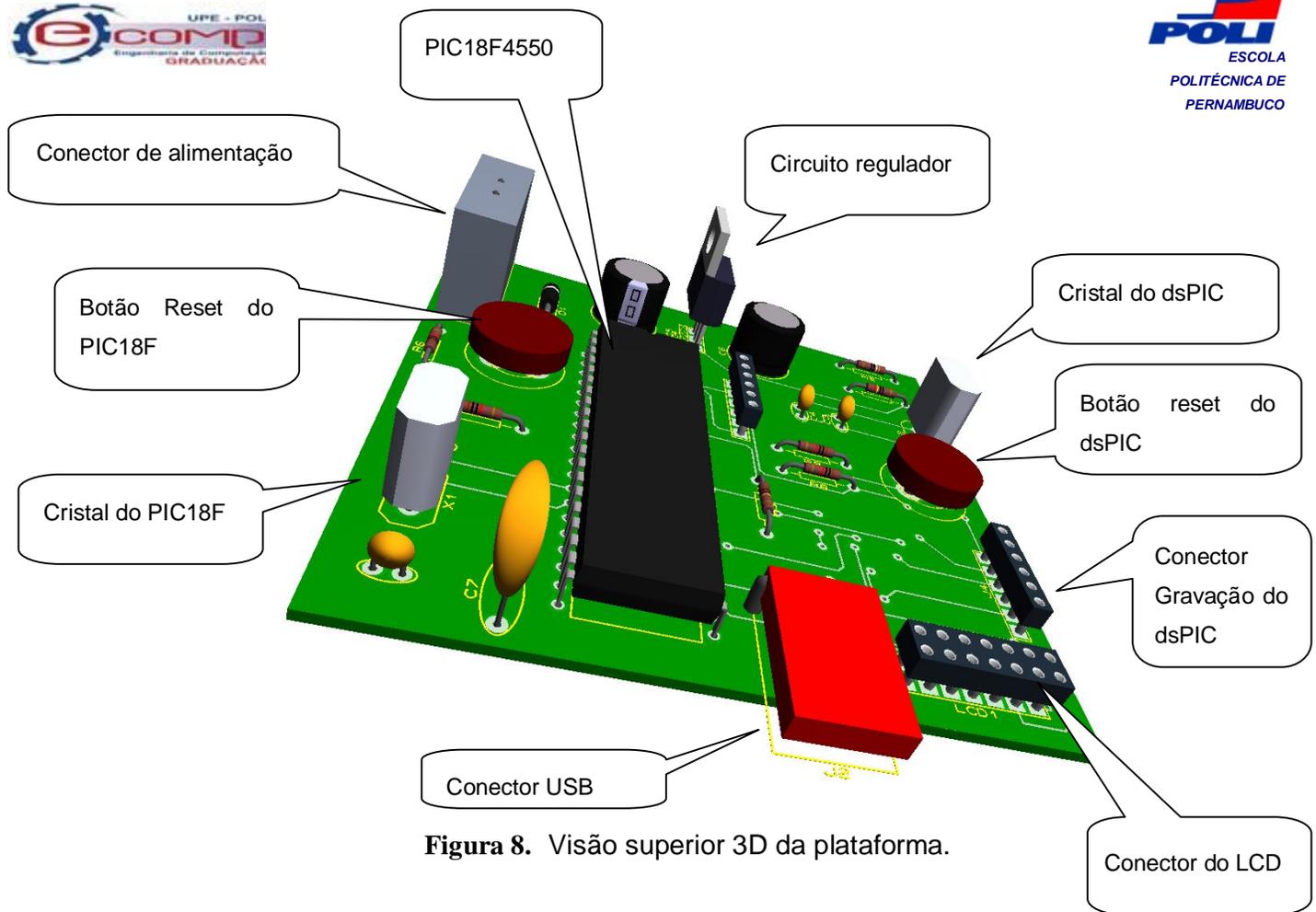


Figura 8. Visão superior 3D da plataforma.

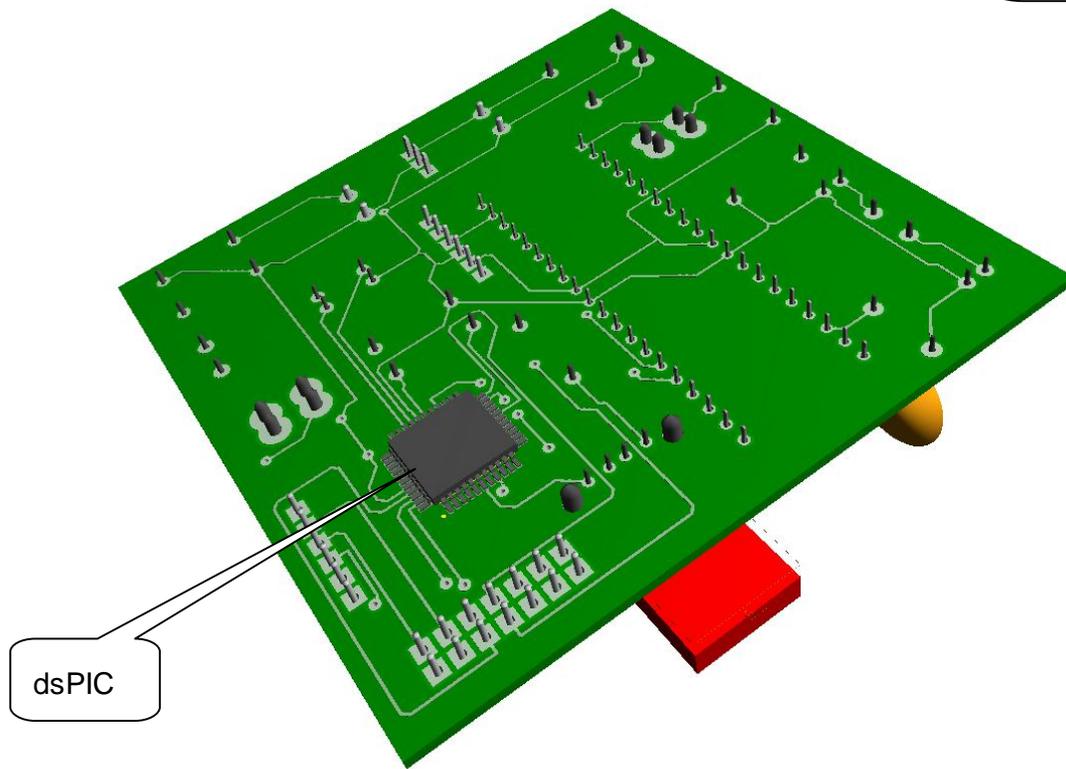


Figura 9. Vista inferior 3D da plataforma.

3.3 Envio da imagem

Para a geração da imagem no PC foi desenvolvido um software na linguagem Java, o qual que envia um array de 25 bits, cada um representando um pixel da imagem. Para o uso da porta serial na maquina virtual Java foi optado pelo uso da biblioteca OpenSource rxtx. A imagem 10 apresenta a interface gráfica do software, neste exemplo esta sendo enviada a imagem que representa o caractere T. No software é possível escolher quais pixels serão 1 ou 0 para serem enviado para a plataforma.

A abordagem na plataforma para a conversão USB-Serial utilizou o drive CDC que abstrai todo o protocolo USB e disponibiliza uma interface Serial RS-232, é utilizada a velocidade de 9.600 Kbps para trabalho do CDC. Outra vantagem da utilização do PIC18F4550 como conversor é que o mesmo pode se comportar como dispositivo HID, permitindo assim que a plataforma tenha mais funcionalidades. Um exemplo de dispositivo HID é um mouse, assim a plataforma pode se transforma em um mouse no qual o ponteiro do mouse é guiado pela imagem da câmera.

PIC18f4550 funciona como uma ponte, ao receber dados pela sua porta USB retransmite para sua porta serial para ser encaminhado para a porta serial do dsPIC, como pode ser visto no código na Figura 11. O mesmo fica verificando dados na porta USB em um loop e envia para serial, todo o código foi implementado na linguagem C. Atualmente a imagem, recebida pela plataforma, é do tamanho 5x5 composta de 25 pixels monocromáticos de 1 bit por pixel.



Figura 10. Aplicativo em Java para enviar imagem.

3.4 Processamento da imagem

Para o teste da plataforma foi utilizado um exemplo simples de uma Rede Neural Adaline, com função de ativação Sigmoidal Logistic, para o reconhecimento de dois caracteres T e H. A escolha da rede Adaline foi baseada pelo fato de ela ser de simples execução comparada a MLP, por não possuir camada escondida. A rede Adaline também é mais aprimorada comparada a rede Perceptron por possuir função de ativação contínua. Para o treinamento da rede neural que roda no dsPIC foi implementado um algoritmo para treinamento na linguagem Java executando em um computador comum.

A arquitetura da rede é composta de 1 neurônio com 26 entradas, 25 (X_1 à X_{25}) entradas uma para cada pixel mais 1 (X_0) para bias. O valor do bias é de 1. A taxa de aprendizado foi escolhida igual a 1, pelo fato do problema ser simples e permitir uma convergência mais rápida na fase de treinamento. Na Figura 12 é apresentado o conjunto de teste com as letras H e T tanto maiúscula quanto minúscula.

Na Figura 13 é apresentado o gráfico do erro médio quadrático. Ele foi usado como critério de parada no treinamento. O treinamento foi interrompido na 20ª iteração. Como pode ser visto na seção 3.5 a rede neural não decorou os exemplos de treinamento, pelo fato da mesma ser capaz de reconhecer o caractere mesmo em

formato não presente no conjunto de treinamento. Como é mostrado no gráfico para o problema de reconhecimento de caractere a rede neural consegue aprender e a resolver o problema. Os pesos calculados estão na tabela 1. Nela tem os pesos da linha 2 colunas 2 e 3 iguais a 0, isso significa que não contribuem em nada para a identificação do caractere, os pesos negativos são os que influenciam para reconhecimento como T e os positivos para reconhecimento com H.

```
#use rs232(baud=9600, xmit=PIN_C6, rcv=PIN_C7)
#include <usb_cdc.h>
#include <string.h>
void usb_debug_task(void);
void main(void) {
    char c;
    disable_interrupts(GLOBAL);
    while(true)
    {
        usb_task();
        usb_debug_task();
        while (usb_cdc_kbhit()) {
            c=usb_cdc_getc();
            if (c=='\n') {putc('\r'); putc('\n');}
            else if (c=='\r') {putc('\r'); putc('\n');}
            else {putc(c);}
            if (kbhit())
            {
                c=getc();
                usb_cdc_putc(c);
            }
        }
    }
}
```

Figura 11.Código PIC18F4550.

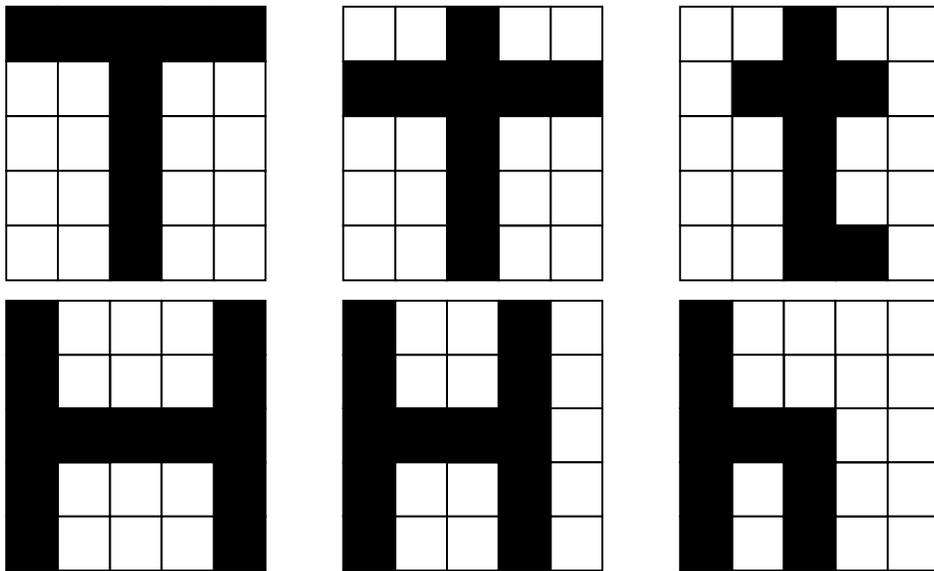


Figura 12. Conjunto de treinamento letras T e H.

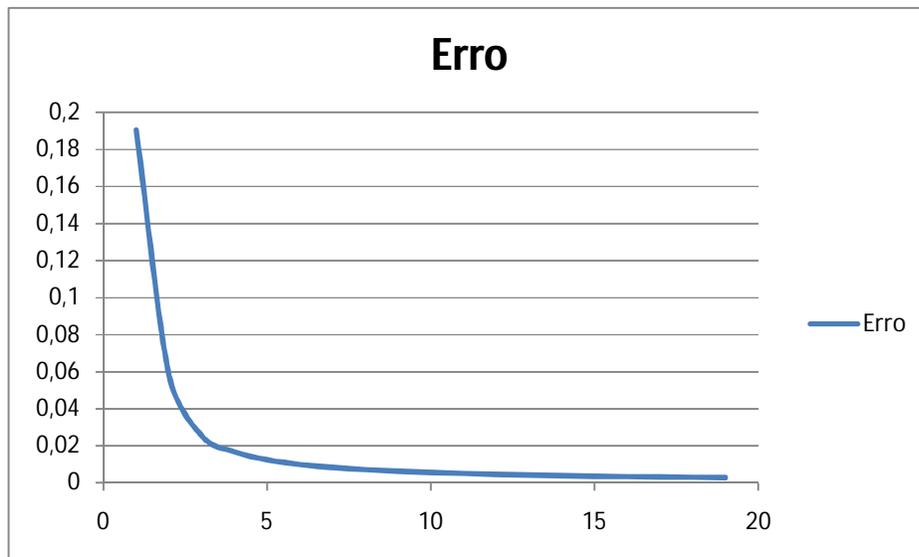


Figura 13. Erro no processo de aprendizagem.

Tabela 1. Matriz de pesos da rede neural.

| Neurônio/Entrada | Entrada 1 | Entrada 2 | Entrada 3 | Entrada 4 | Entrada 5 |
|------------------|-----------|-----------|-----------|-----------|-----------|
| Neurônio 1 | -0,45169 | -0,52973 | -0,89581 | -0,89581 | -0,89581 |
| Neurônio 2 | 0,44412 | 0,55216 | -0,89581 | 0,00000 | 0,00000 |
| Neurônio 3 | 0,99628 | 0,99628 | 0,10046 | 0,48486 | 0,48486 |
| Neurônio 4 | 0,24983 | 0,35786 | -0,38440 | -0,19430 | -0,00823 |
| Neurônio 5 | 0,25402 | 0,17599 | -0,19010 | -0,19010 | -0,19010 |

Com os pesos ajustados para a implementação do código fonte na linguagem C para o dsPIC. Uma primeira preocupação foi poupar recursos do microcontrolador, para isto os pesos foram colocados como um array constante, como isso os dados ficam na memória flash, sem ocupar espaço na RAM, gerando uma economia de 5% da memória RAM segundo dados do compilador usado.

Como os pesos são números fracionários menores que zero, foram multiplicados todos por 100 e excluído a parte fracionária, ficando apenas com a parte inteira, esta decisão foi tomada visando a não utilização de divisão de números flutuantes o que seria mais custoso computacionalmente para o microcontrolador.

Na Figura 14 tem o código da rede neural no DSPIC, nela pode ser visto que o pacote com a imagem chega pela interrupção da porta serial e em tempo real, ou seja, sem o armazenamento da imagem o que tornaria o processo mais custoso, é realizada a execução da rede neural determinando qual o caractere que é reconhecido.

```
#int_RDA
void RDA_isr(void)
{
    disable_interrupts(INT_RDA);
    saida1=bias1;
    i=0;
    while(i<25)
    {
        entrada = getc();
        saida1 += entrada*pesos1[i++];
    }
    if(saida1>0)
        printf(lcd_putc, "\fCaractere = T\n Net = %d",saida1);
    else
        printf(lcd_putc, "\fCaractere = H\n Net = %d",saida1);
    enable_interrupts(INT_RDA);
}
```

Tabela 2. Código da Rede Neural no dsPIC.

3.5 Resultados

Para validação da plataforma primeira foram executados testes enviando pacotes com imagens através da aplicação em Java, os primeiro pacotes foram com as leras usadas no treinamento, como pode ser visto nas Figuras 15a e 15b a entrada e a saída obtida no display LCD. No display LCD é mostrado o caractere que foi reconhecido mais o valor do Net da rede. O Net é o valor obtido pela soma das multiplicações dos pesos pelas entradas do neurônio.

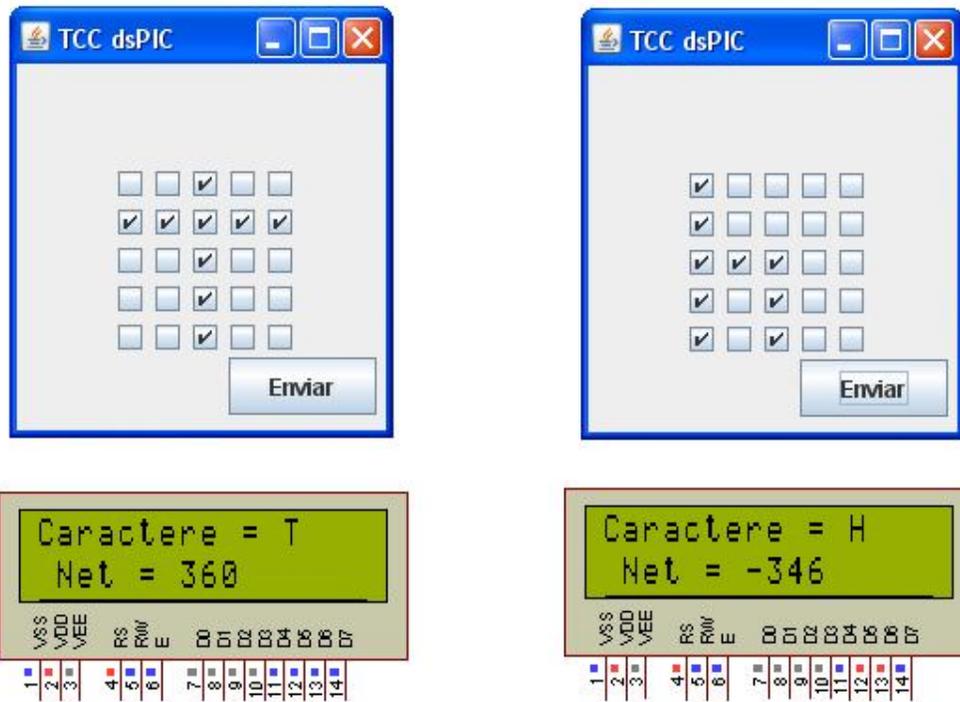


Figura 14. a) Teste com a letra t b) Teste com a letra h.

O uso de uma rede neural para o reconhecimento de caractere foi feita pela sua capacidade de generalização, ao ser capturada uma imagem pode conter ruídos, pra verificar a capacidade da rede neural foi realizado o teste em imagens de letras com ruídos como pode ser visto nas Figuras 16(a). e 16(b). Mesmo com ruído foi reconhecido o caractere correto, porém o valor do net ficou mais próximo de 0 comparado aos casos das Figuras 15a e 15b.

Outro problema que pode ser apresentado em imagens capturadas é a ausência de pixels, para esta simulação foram criadas imagens como pode ser vista nas Figuras 17(a) e 17(b). Mesmo com a ausência de pixels, os caracteres foram devidamente reconhecidos, porém com um valor de net mais próximo de 0 comprado aos das figuras 15(a) e 15(b).

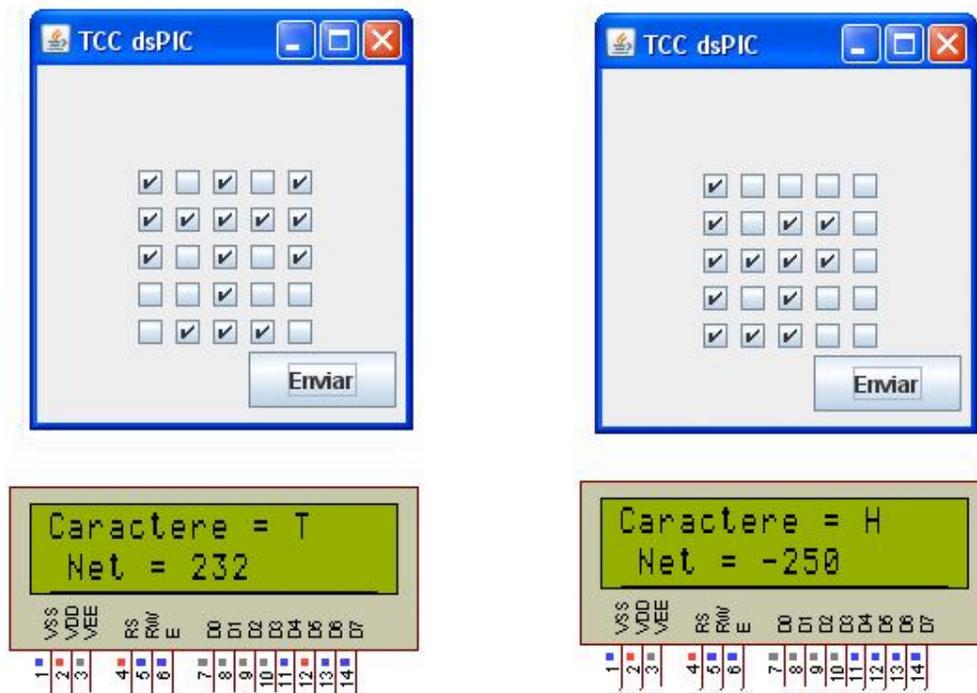


Figura 15.a) Teste com a letra t com ruído. B) Teste da letra h com ruído.

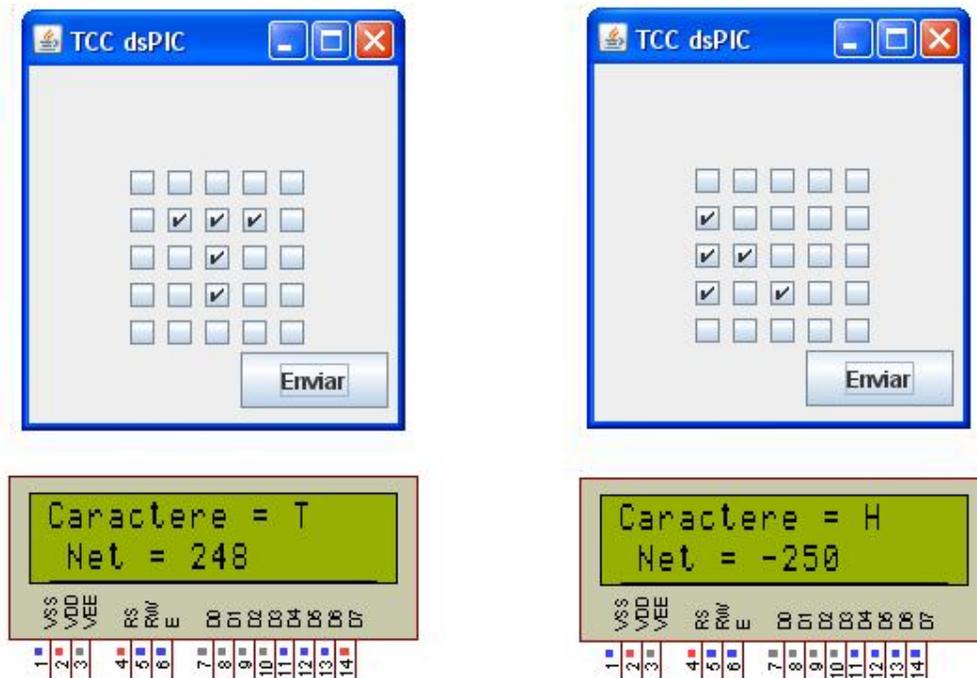


Figura 16.a) Letra t incompleta. B) Letra h incompleta.

Conclusão e Trabalhos Futuros

4.1 Conclusão

O principal objetivo deste trabalho foi o desenvolvimento de uma plataforma para processamento embarcado de imagem utilizando como microcontrolador o dsPIC 33FJ16GP304.

Na parte experimental deste trabalho foi possível realizar o processamento de imagem em uma plataforma utilizando um microcontrolador dsp de baixo custo. Durante os testes realizados foi verificado que o mesmo apresenta uma opção para processamentos de imagem mais simples, porém com aplicações bastante úteis.

A opção de utilizar uma rede neural Adaline para reconhecimento de caracteres, mostrou-se bastante satisfatório, sendo reconhecidos os caracteres presentes no conjunto de treinamento como também o de caracteres não presente no conjunto de treinamento.

Portanto concluímos que a plataforma é capaz de realizar o processamento de imagem podendo ser bastante útil em substituir um computador genérico.

4.2 Trabalho Futuros

Alguns trabalhos futuros.

1. Confecção do hardware da plataforma.

Este trabalho visa a confecção da plataforma com componentes reais e da placa de circuito impresso. Começando pela compra dos mesmos, confecção da placa de circuito impresso, montagem e soldagem dos componentes.

2. Adaptação de uma câmera real Serial.

Neste trabalho visa a acoplamento de uma câmera real na plataforma, atividade que envolve o estudo do protocolo e desenvolvimento do código que receberá o mesmo.

3. Mouse ocular.

Neste trabalho visa o desenvolvimento de um mouse ocular no qual o movimento do ponteiro do mouse é orientado pela posição da Iris do olho humano.

Bibliografia

[1] GONZALES C. Rafael; WOODS E. Richard. Processamento de Imagens Digitais. 3ª Ed., Pearson, 2010.

[2] Lima, S. M. L. de; SISTEMA DE AUXÍLIO AO DIAGNÓSTICO DE DOENÇAS INTRACRANIANAS E CÂNCER DE MAMA. 2009. Trabalho de conclusão de curso do curso de Engenharia da Computação, Escola Politécnica de Pernambuco, universidade de Pernambuco, Pag 13-62.

[3] Lima, R. A. ; Oliveira, S. C. ; FONTANA, Eduardo ; MARTINS FILHO, J. F. ; Cavalcanti, G. O. ; PRATA, T. L. ; LIMA, Renato Bandeira ; F. J. M. M. Cavalcanti . Satellite Telemetry System for Pollution Detection on Insulator Strings of High-Voltage Transmission Lines. In: International Microwave and Optoelectronics Conference - IMOC 2009, 2009, Belém. Proceedings of the International Microwave and Optoelectronics Conference, 2009.

[4] wikilingue. Disponível em: <http://pt.wikilingue.com/es/Processador_digital_de_sinal> Acesso em: 22 de julho de 2010.

[5] Pereira, M F L; Algoritmo Paralelo para Reconstrução Tridimensional de Imagens Tomográficas de Amostras Agrícolas em Arquitetura DSP com Técnicas Wavelets, 2001, Tese de mestrado em Ciência da Computação, Universidade de são Carlos.

[6] Duan Jinghong, Deng Yaling Liang Kun;Development of Image Processing System Based on DSP and FPGA; ICEMI'2007.

[7] DSP. Disponível em <http://pt.wikipedia.org/wiki/DSP>, acesso em 19 de dezembro 2010.

[8] 16-bit PIC24 MCUs and dsPIC® DSCs. Disponível em http://www.microchip.com/en_US/family/16bit/, acesso em 19 de dezembro 2010.

Apêndice A

Código fonte do dsPIC 3FJ16GP304

```
#include<33FJ16GP304.h>
#FUSES NOWDT,PR,HS
#use delay(clock=4000000)
#include <lcd.c>

//pinos da uart
#Pin_select U1TX=PIN_C0
#Pin_select U1RX=PIN_C1

//pinos do lcd
#define LCD_ENABLE_PIN PIN_B0
#define LCD_RS_PIN    PIN_B1
#define LCD_RW_PIN    PIN_B2
#define LCD_DATA4     PIN_B4
#define LCD_DATA5     PIN_B5
#define LCD_DATA6     PIN_B6
#define LCD_DATA7     PIN_B7

#use rs232(UART1,baud=9600,parity=n,bits=8, ERRORS)

int bias1 = 1;
int saida1;
const int16 pesos1[]={-45, 44, 99, 24, 25,
                    -52, 55, 99, 35, 17,
                    -89,-89, 10,-38,-19,
                    -89, 0, 48,-19,-19,
                    -89, 0, 48, 0,-19};
```

```
int i;
int entrada;
#define int_RDA
void RDA_isr(void)
{
    disable_interrupts(INT_RDA);
    saida1=bias1;
    i=0;
    while(i<25)
    {
        entrada = getc();
        saida1 += entrada*pesos1[i++];

    }
    if(saida1>0)
        printf(lcd_putc,"\fCaractere = T\n Net = %d",saida1);
    else
        printf(lcd_putc,"\fCaractere = H\n Net = %d",saida1);
    enable_interrupts(INT_RDA);
}
```

```
void main()
{
    setup_spi( FALSE );
    setup_wdt(WDT_OFF);
    setup_timer1(TMR_DISABLED);
    lcd_init();
    delay_ms(6);
    enable_interrupts(INT_RDA);

    while(1)
```

```
{  
  
}  
}
```

Apêndice B

Código fonte do PIC18F4550

```
#include <18F4550.h>  
  
#fuses  
HSPLL,NOWDT,NOPROTECT,NOLVP,NODEBUG,USBDIV,PLL5,CPUDIV1,VREGE  
N  
  
#use delay(clock=48000000)  
#use rs232(baud=9600, xmit=PIN_C6, rcv=PIN_C7)  
#define USB_CON_SENSE_PIN PIN_B2  
  
#include <usb_cdc.h>  
#include <string.h>  
void usb_debug_task(void);  
void main(void) {  
    char c;  
    disable_interrupts(GLOBAL);  
  
    while(true)  
    {  
        usb_task();  
        usb_debug_task();  
        while (usb_cdc_kbhit()) {  
            c=usb_cdc_getc();  
            if (c=='\n') {putc('\r'); putc('\n');}  
            else if (c=='\r') {putc('\r'); putc('\n');}  
            else {putc(c);}  
            if (kbhit())
```

```
{
    c=getc();
    usb_cdc_putc(c);
}
}
}

void usb_debug_task(void) {
    static int8 last_connected;
    static int8 last_enumerated;
    int8 new_connected;
    int8 new_enumerated;
    static int8 last_cdc;
    int8 new_cdc;
    new_connected=usb_attached();
    new_enumerated=usb_enumerated();
    new_cdc=usb_cdc_connected();
    last_connected=new_connected;
    last_enumerated=new_enumerated;
    last_cdc=new_cdc;
}
}
```

```
void usb_debug_task(void) {
    static int8 last_connected;
    static int8 last_enumerated;
    int8 new_connected;
    int8 new_enumerated;
    static int8 last_cdc;
    int8 new_cdc;

    new_connected=usb_attached();
```

```
new_enumerated=usb_enumerated();  
new_cdc=usb_cdc_connected();  
  
last_connected=new_connected;  
last_enumerated=new_enumerated;  
last_cdc=new_cdc;  
}
```