

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
ESCOLA DE ENGENHARIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

RICARDO DE AZAMBUJA

PROJETO DE DIPLOMAÇÃO

TEXVID

Porto Alegre
2005

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
ESCOLA DE ENGENHARIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

TEXVID

INTERFACE SERIAL PARA SAÍDA DE TEXTO EM SINAL DE VÍDEO COMPOSTO

Projeto de Diplomação apresentado ao
Departamento de Engenharia Elétrica da Universidade
Federal do Rio Grande do Sul, como parte dos
requisitos para Graduação em Engenharia Elétrica.

ORIENTADOR: Tiaraju Vasconcellos Wagner

Porto Alegre
2005

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
ESCOLA DE ENGENHARIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

RICARDO DE AZAMBUJA

TEXVID
INTERFACE SERIAL PARA SAÍDA DE TEXTO EM SINAL DE
VÍDEO COMPOSTO

Este projeto foi julgado adequado para fazer jus aos créditos da Disciplina de “Projeto de Diplomação”, do Departamento de Engenharia Elétrica e aprovado em sua forma final pelo Orientador e pela Banca Examinadora.

Orientador: _____

Prof. Tiaraju Vasconcellos Wagner, UFRGS

Banca Examinadora:
Prof.

Prof.

Coordenador da Comissão de Graduação: _____

Chefe do Departamento: _____

Porto Alegre, novembro de 2005.

DEDICATÓRIA

Dedico este trabalho a todos aqueles que sempre me apoiaram em tudo que eu fiz.
Graças a eles, consegui dar mais este passo na estrada sem-fim do conhecimento.

AGRADECIMENTOS

Agradeço aos meus pais, por sempre terem me incentivado nos estudos, nos negócios, enfim, em tudo que eu fiz até hoje.

Ao meu orientador, por ter tido paciência e por ter compartilhado a sua experiência para me ajudar neste trabalho.

Aos meus colegas, pelas dicas e todo apoio que me deram.

Ao ensino público federal, pois sem ele eu não teria a oportunidade de estudar nesta fabulosa universidade.

A todos aqueles que acreditaram na minha capacidade de realizar mais essa tarefa.

RESUMO

Este relatório apresentará, de forma sucinta, todos os passos – da idéia inicial até os testes finais – do desenvolvimento da Interface Serial para Saída de Texto em Sinal de Vídeo Composto – TEXvid. Nele encontrar-se-ão explicações básicas sobre a geração de sinais de vídeo composto monocromáticos no padrão PAL-M utilizando um microcontrolador PIC16F628A, os algoritmos utilizados para vencer os desafios propostos, os passos para montar o hardware necessário e as motivações que me levaram a escrever este trabalho. Uma breve discussão sobre a escolha dos componentes, configuração da serial e das simplificações feitas no projeto inicial também poderão ser vistas.

Palavras-chaves: Engenharia Elétrica; Automação Industrial; Automação Residencial; Automação Comercial; Microcontroladores; Sinais de Vídeo; PAL-M.

ABSTRACT

This work will introduce all the steps – from the beginning until the final tests – developing the Composite Video Serial Interface TEXvid. Here, find out the basic of generating monochrome PAL-M video using just a microcontroller PIC16F628A, the algorithms, hardware and the causes let me develop this system. A brief discussion about serial interface and choosing the right parts for a project could be seen too.

Keywords: Electrical Engineering; Home, Commercial and Industrial Automation; Microcontrollers; Video Signals; PAL-M.

SUMÁRIO

1 INTRODUÇÃO.....	11
2 BREVE HISTÓRICO DA TELEVISÃO.....	13
3 FORMAÇÃO DA IMAGEM DE TELEVISÃO.....	15
3.1. O Sinal de Vídeo Composto Monocromático.....	18
3.1.1. Tempo de apagamento Horizontal.....	21
3.1.2. Pórtico Anterior e Posterior.....	22
3.1.3. Sincronismo horizontal.....	23
3.1.4. Tempo de apagamento vertical.....	23
3.1.5. Sincronismo vertical.....	24
4 ESPECIFICAÇÕES E IDÉIAS INICIAIS.....	26
4.1. A idéia inicial.....	26
4.2. Público Alvo.....	27
4.3. Limitações do protótipo inicial.....	27
5 PROJETO E IMPLEMENTAÇÃO.....	29
5.1. Resolução.....	29
5.2. Escolha do microcontrolador.....	31
5.3. Algoritmos, memória e, novamente, resolução.....	32
5.4. Interface Serial.....	34
5.5. Montagem da placa protótipo.....	35
5.6. Geração do sincronismo horizontal.....	41
5.7. Testes do sincronismo horizontal.....	41
5.8. Geração e testes do sincronismo vertical.....	42
5.9. Resolução alcançada e memória consumida.....	43
5.10. Geração das linhas de texto.....	44
5.11. Recebimento dos comandos pela serial.....	45
6 TESTES FINAIS.....	47
7 CUSTOS FINAIS.....	51
8 MELHORIAS FUTURAS.....	53
8.1. Microcontrolador.....	53
8.2. Sobreposição dos caracteres sobre um sinal de vídeo já existente.....	53
8.3. Caixa externa e placa.....	54
9 CONCLUSÕES.....	55
10 REFERÊNCIAS	56
11 APÊNDICE.....	57

LISTA DE ILUSTRAÇÕES

ILUSTRAÇÃO 1: IMAGEM DE TELEVISÃO FORMADA POR ELEMENTOS CHAMADOS PIXELS.....	15
ILUSTRAÇÃO 2: ESQUEMA DE FUNCIONAMENTO DA VARREDURA HORIZONTAL COM RETRAÇOS.....	16
ILUSTRAÇÃO 3: DISTORÇÃO NA IMAGEM DEVIDO AO ATRASO NO RETRAÇO.....	17
ILUSTRAÇÃO 4: SINCRONISMO HORIZONTAL E IMAGEM.....	20
ILUSTRAÇÃO 5: PULSOS DE APAGAMENTO E SINCRONISMO.....	22
ILUSTRAÇÃO 6: LOCALIZAÇÃO DOS PÓRTICOS ANTERIOR E POSTERIOR..	22
ILUSTRAÇÃO 7: SOBREPOSIÇÃO DO APAGAMENTO HORIZONTAL SOBRE O PULSO DE SINCRONISMO.....	23
ILUSTRAÇÃO 8: PULSOS DE EQUALIZAÇÃO E DE SINCRONISMOS.....	24
ILUSTRAÇÃO 9: ÁREA ÚTIL DISPONÍVEL APÓS CONTABILIZAR OS SINCRONISMOS E OS APAGAMENTOS.....	25
ILUSTRAÇÃO 10: DIAGRAMA DE BLOCOS DO SISTEMA.....	26
ILUSTRAÇÃO 11: EXEMPLOS DE CARACTERES CRIADOS.....	30
ILUSTRAÇÃO 12: LAYOUT DA PLACA: PIC16F628 E COMPONENTES PRÓXIMOS.....	37
ILUSTRAÇÃO 13: LAYOUT DA PLACA: MAX232 E CONECTOR DB9.....	38
ILUSTRAÇÃO 14: LAYOUT DA PLACA: ENTRADA DA ALIMENTAÇÃO.....	38
ILUSTRAÇÃO 15: DISTRIBUIÇÃO DOS COMPONENTES NA PLACA.....	39
ILUSTRAÇÃO 16: MÁSCARA USADA NA CONFECÇÃO DA PLACA.....	39
ILUSTRAÇÃO 17: PADRÃO GERADO USANDO 10101010.....	42
ILUSTRAÇÃO 18: SAÍDA DE VÍDEO NO OSCILOSCÓPIO TRIGANDO AS LINHAS HORIZONTAIS.....	42
ILUSTRAÇÃO 19: LINHA VERTICAL GERADA COM O SINCRONISMO VERTICAL.....	43
ILUSTRAÇÃO 20: TEXTO GERADO COM RUÍDOS.....	44
ILUSTRAÇÃO 21: TEXTO FUNCIONANDO CORRETAMENTE.....	45
ILUSTRAÇÃO 22: EXEMPLO DE TELA GERADA.....	46
ILUSTRAÇÃO 23: SINAL DA SAÍDA DO TEXVID.....	47
ILUSTRAÇÃO 24: SINAL NA SAÍDA DA FILMADORA.....	48
ILUSTRAÇÃO 25: 1º TESTE DOS CARACTERES.....	49
ILUSTRAÇÃO 26: 2º TESTE DOS CARACTERES.....	49
ILUSTRAÇÃO 27: 3º TESTE DOS CARACTERES.....	50

LISTA DE TABELAS

TABELA 1: ESCALA IRE.....	20
TABELA 2: CONVERSÃO IRE X MV.....	21
TABELA 3: CARACTERÍSTICAS DESEJADAS E SIMPLIFICAÇÕES.....	28
TABELA 4: COMANDOS ENVIADOS PELA SERIAL.....	46
TABELA 5: PLANILHA DE CUSTOS FINAIS, SEM CONSIDERAR O CUSTO DE DESENVOLVIMENTO.....	52

1 INTRODUÇÃO

Foi-se o tempo que somente a propaganda era a alma dos negócios. Com a explosão da internet, sistemas celulares de 3ª geração, localizadores com gps, o onipresente (ou onipotente?) Google, tocadores de música e vídeo com mais de 40GB de memória sendo usados com podcasting e muitas outras fontes de informação, lidar com esse verdadeiro maremoto, sem se afogar, é o novo desafio dos negócios.

Tendo isso na cabeça e com lápis e papel na mão – sim, eles ainda são a primeira linha para separar a criatividade do esquecimento – coloquei-me a pensar no que poderia ser feito para o projeto de diplomação. É claro que não foi uma tarefa fácil. O curso de Engenharia Elétrica abre todos os campos imagináveis de trabalho para o aluno. Mas, tendo em vista os prazos e a minha preferência por soluções simples e eficientes que envolvam o uso de microcontroladores, passei a procurar por algo que valesse a pena ser feito.

Apesar de eu ter citado somente novas tecnologias no primeiro parágrafo, há uma que não foi colocada lá de propósito, mas ainda é a mais difundida no nosso país: a televisão. O que poderia ser feito para aproveitá-la para dar vazão a essa necessidade de se compactar todas as informações que devem chegar até nós?

A solução para essa dúvida me veio à mente na fila de um restaurante. Apesar do estabelecimento ter dois aparelhos de 29 polegadas, as pessoas ficavam entreolhando-se para saber qual a mesa estava vaga e quem da fila deveria dirigir-se a ela. Porém, um simples sistema que utilizasse os equipamentos existentes junto com o computador do restaurante poderia fazer fluir as informações de mesas livres, tempo de espera e ordem de chegada com um custo extra muito baixo.

Juntando essa idéia com a lembrança de um projeto de microcontroladores AVR utilizados para geração de sinais de vídeo que eu havia visto em uma página da Universidade

de Cornell (nos EUA), nasceu a idéia do TEXvid. A partir desse ponto, comecei a estudar o que seria necessário para gerar sinais de vídeo com o mínimo custo e utilizando componentes que fossem possíveis de serem adquiridos aqui no Brasil. Também iniciei a batalha entre capacidade do hardware disponível e “vendabilidade” do produto final.

Neste relatório irei apresentar todos os passos que eu tive de dar para conseguir transformar a idéia de um produto – a Interface Serial para Saída de Texto em Sinal de Vídeo Composto TEXvid – que também é o meu Projeto de Diplomação do curso de Engenharia Elétrica da Universidade Federal do Rio Grande do Sul, em algo factível tendo em vista os prazos, a burocracia, a falta de equipamentos, quatro disciplinas e os custos finais do projeto.

Para facilitar o acompanhamento do desenvolvimento deste trabalho, irei inicialmente dar um breve histórico deste fascinante aparelho, que é a televisão, para depois explicar o seu funcionamento básico necessário para a formação da imagem e, só então, iniciar o detalhamento de todo o meu trabalho. Acredito que seguindo essa ordem, o leitor conseguirá aproveitar ao máximo este relatório.

2 BREVE HISTÓRICO DA TELEVISÃO

Em 1873, um jovem irlandês operador de telégrafo, Joseph May, descobriu que barras de selenium, quando expostas ao sol, tinham a sua resistência elétrica variada. Ou seja, variações de luz poderiam ser captadas e transformadas em sinais elétricos. Já em 1875, George Carey propôs um sistema que cada ponto da imagem fosse formado por um elemento sensível – a imagem seria lida de forma simultânea. Na época, esse sistema se mostrou impraticável para uma resolução que o torna-se útil. Após, foi publicado, em 1881, na França, por Constantin Senlecq um esquema de um sistema que utilizava uma idéia inovadora que seria utilizar apenas um elemento sensível e enviar a informação da imagem de forma serial. Essa foi a base da televisão moderna: ao invés de um grande sensor que lia todos os pontos de uma só vez, apenas um feixe que varre toda a área da imagem. O alemão Paul Nipkow conseguiu, em 1884, patentear um sistema de escaneamento de imagem que utilizava discos rotativos com uma série de furos dispostos em uma espiral. Um raio de luz que passasse através dos furos iluminaria cada linha da imagem. Após uma rotação completa dos discos, a imagem inteira era escaneada. Com uma velocidade suficientemente rápida, o olho humano não conseguiria perceber que a imagem era formada por pontos. Apesar da idéia ser simples, a tecnologia da época não possibilitou que fosse colocada em prática.

Quando o tubo de raios catódicos com uma tela fluorescente foi inventado, em 1897, por Karl Ferdinand Braun, a abordagem mecânica para a televisão teve seus dias contados (apesar da competição entre elas ter durado até 1937). No início de 1908, A. A. Campbell Swinton esboçou um sistema que usava tubo de imagem tanto para capturar quanto para mostrar a imagem na tela. Esse foi o primeiro sistema puramente eletrônico. Porém, como as células disponíveis ainda não eram sensíveis o suficiente (e não haviam amplificadores), os sinais ficavam muito fracos.

Vladimir Zworykin inventou em 1923 o Iconoscope. A primeira imagem produzida veio em 1925, por um engenheiro elétrico escocês – John Logie Baird. Era um sistema de dois discos montados em um mesmo eixo que produzia uma imagem com 30 linhas e 5 telas por segundo. Por volta de 1930, muitos pesquisadores, independentemente, desenvolveram o princípio da imagem entrelaçada. Num encontro de engenheiros em Nova York, Zworykin apresentou o seu primeiro protótipo do Iconoscope. O aparelho foi construído pela RCA. Ele escaneava as imagens com 120 linhas a 24 quadros por segundo. Com o interesse da indústria no negócio, receptores a tubo de raios catódicos estavam no mercado já em 1933.

O primeiro sistema televisivo teve início em Berlim, Alemanha, em março de 1935. Eles transmitiam 180 linhas / quadro a 25 quadros por segundo. As imagens eram feitas com filmes e depois escaneadas por um sistema de discos rotativos. As câmeras eletrônicas foram desenvolvidas em 1936 e utilizadas já nas olimpíadas de Berlim. Em novembro do mesmo ano, Paris teve a sua primeira transmissão de televisão.

A partir daí, os sistemas evoluíram para 240 linhas (sistema mecânico), 405 linhas em 1937 (Inglaterra), 455 linhas ainda no mesmo ano (França). Alemanha e Itália adotaram 441 linhas. De 1937 até 1939, as vendas de receptores em Londres saltaram de 2.000 para 20.000 aparelhos.

Em 1941, nos Estados Unidos, foi adotado o padrão de 525 linhas e 60 quadros por segundo. E, a partir de 1952, progressivamente seria adotado na Europa o padrão de 625 linhas e 50 quadros por segundo. Chegávamos na era da televisão moderna.

3 FORMAÇÃO DA IMAGEM DE TELEVISÃO

Para conseguir gerar sinais de vídeo que possam ser utilizados nos aparelhos de televisão, é necessária a compreensão do funcionamento desses equipamentos primeiro. Portanto, nesta parte do relatório, serão revisados alguns aspectos básicos da formação da imagem na TV com algumas simplificações.

A televisão é um equipamento que reproduz imagens estáticas em alta velocidade. Essas imagens são uma combinação de pontos claros e escuros numa tela sensível. Os elementos de imagem são conhecidos por “pixel”. A sobreposição de diversas imagens estáticas durante um curto espaço de tempo é o que dá a ilusão de algo em movimento.

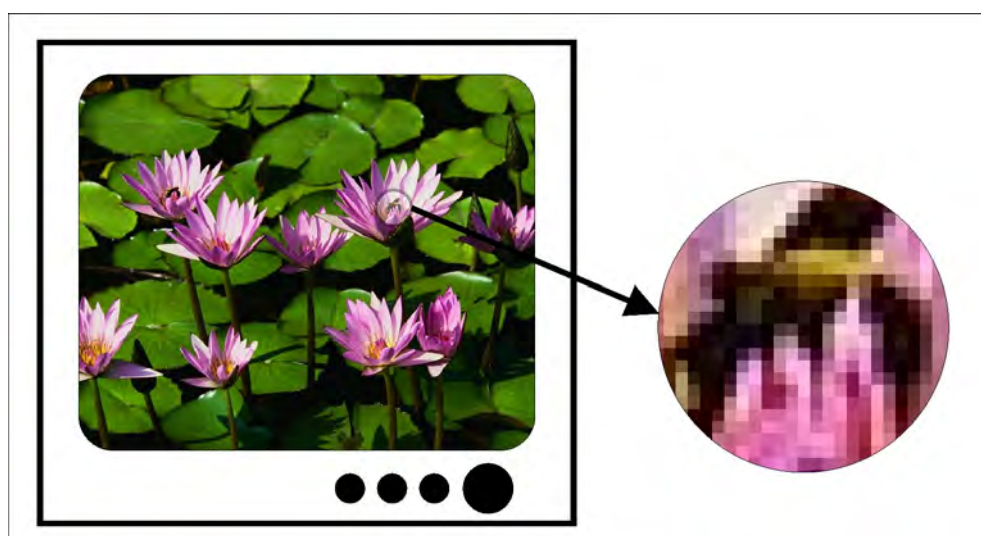


Ilustração 1: Imagem de televisão formada por elementos chamados pixels.

A sensibilidade dos nossos olhos à luz faz a resposta a variações rápidas na quantidade recebida por eles não ser percebida instantaneamente. Basta ficar olhando para uma lâmpada, por exemplo, que logo após desligá-la a imagem persistirá na retina, fazendo parecer que a ela ainda encontra-se acesa. Essa característica do olho humano é muito explorada em shows de mágica e brinquedos de ilusão de ótica. Desse modo, imagens mostradas com uma taxa maior que 16 Hz já nos dão uma ilusão de movimento. Nas projeções de cinema é usada a taxa de 24 quadros por segundo para criar a ilusão de movimento das imagens.

A taxa de 24 quadros por segundo, apesar de já conseguir a ilusão de movimento, não é suficientemente rápida para que o brilho das imagens sofra uma passagem suave entre dois quadros consecutivos. Quanto maior o nível de iluminação, mais acentuado é esse efeito de cintilação. Nas projeções do cinema esse problema foi resolvido mostrando duas vezes o mesmo quadro e, com isso, alcançando uma taxa de 48 quadros por segundo. Com o aumento da frequência do apagamento das imagens, o efeito de cintilação é eliminado. Na televisão esse problema é resolvido com o entrelaçamento das imagens e a utilização de 60Hz.

A imagem na TV é o resultado das diversas colisões dos elétrons arremessados contra a tela numa varredura da esquerda para a direita (Varredura Horizontal), linha após linha (Varredura Vertical).

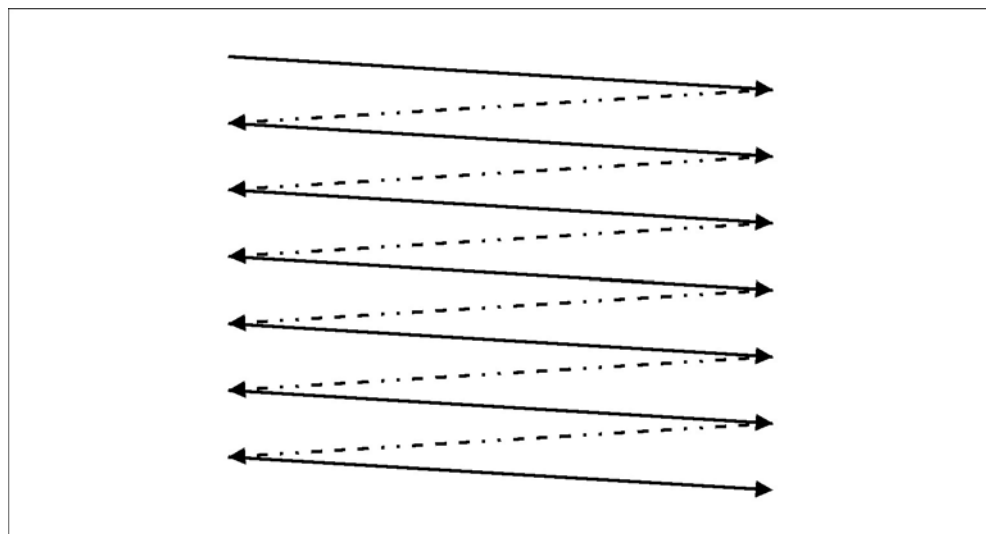


Ilustração 2: Esquema de funcionamento da varredura horizontal com retraços.

Diferentemente dos sistemas de reprodução de imagens baseados em fotografia, que mostram uma imagem inteira a cada quadro, o televisor vai formando a imagem a ser exibida ponto a ponto, linha após linha e quadro após quadro. Portanto, a temporização desses sinais é muito importante, pois se uma linha demorar um pouco mais para chegar ao receptor, aparecerá uma mancha negra no lugar onde ela deveria estar, e a imagem ficará distorcida.

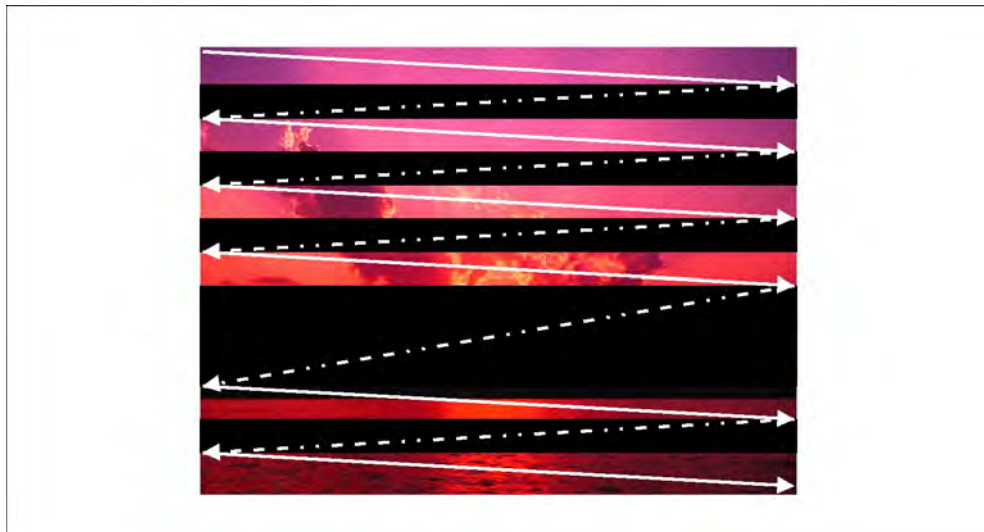


Ilustração 3: Distorção na imagem devido ao atraso no retraço.

Como se fosse a leitura de um texto, a imagem vai sendo formada a cada linha que é mostrada. Iniciando no topo da esquerda, todos os elementos são varridos da esquerda para direita até chegar ao final da tela. Esse método é conhecido por Varredura Linear Horizontal. Ele é utilizado tanto na câmera quanto no televisor, o que cria uma consistência para a geração e a reprodução utilizando o mesmo sinal.

A sequência de eventos para varrer todos os pixels da tela é a seguinte:

1. O feixe de elétrons percorre toda uma linha horizontal.
2. Após chegar ao final da linha, o feixe retorna rapidamente para a esquerda para iniciar o desenho da linha abaixo. Esse tempo de retorno é chamado de retraço e nele nenhuma imagem é mostrada na tela. Na verdade, durante esse tempo a imagem fica em nível de preto.
3. Como são feitos dois movimentos simultâneos, vertical e horizontal, quando o feixe retorna ao lado esquerdo após o retraço, ele estará um pouco abaixo de onde havia começado a linha. Ao chegar no final da tela, ocorre o retraço novamente, mas agora para voltar para o topo.

Quanto maior a resolução que se quer obter, maior o número de elementos de imagem. Como a televisão funciona linha a linha, o número de linhas define a altura do pixel e, por conseguinte, a resolução da imagem. Porém a qualidade da imagem não foi o único fator na definição do padrão de 525 linhas por quadro (RS-170). Entre os diversos fatores, este número foi usado por ser ideal para a largura de banda de 6MHz ocupada por um canal de televisão.

Conforme visto no parágrafo sobre a persistência visual, para criar a ilusão de movimento sem o problema da cintilação são necessários 48 variações no brilho por segundo. O movimento do feixe para baixo é o que nos dá essa frequência de quadros por segundo. Com o sistema de imagem entrelaçada, cada quadro é dividido em dois campos e esses são mostrados 60 vezes por segundo.

3.1. O SINAL DE VÍDEO COMPOSTO MONOCROMÁTICO

O padrão norte-americano RS-170 foi desenvolvido na década de 40 para transmissões de vídeo monocromáticas. A evolução desse padrão é o RS-170A que tem como principal diferença a capacidade de carregar a informação de cor. Porém a parte relativa ao monocromático ficou quase igual. Apenas os sincronismos e a luminescência fazem parte do sinal monocromático. Como este trabalho utiliza somente os valores de luminescência para gerar os caracteres na tela, a partir daqui sempre que for mencionado vídeo composto, estará se referindo ao sinal monocromático. Existem umas pequenas diferenças nas frequências de sincronismo entre os dois padrões. No RS-170 a frequência horizontal é de 15.750Hz e a vertical é de 60Hz. Já no RS-170A temos 15.734,5Hz e 59.94Hz respectivamente. O padrão utilizado no Brasil, o PAL-M, é exatamente igual ao RS-170A (NTSC) no que diz respeito a parte monocromática e, portanto, foi usado como padrão para este projeto.

Todo o funcionamento da TV é baseado no sinal de vídeo recebido. Neste sinal constam os tempos de início e fim de linha, apagamento, retraço, campo par e ímpar além, é óbvio, da imagem propriamente dita.

O sinal de vídeo composto pode ser dividido em três componentes:

1. Sinal da Câmera. Este sinal analógico varia conforme a intensidade luminosa de cada ponto da imagem. Devido a limitação do espectro utilizado na transmissão, ele fica limitado a 6MHz. Ele varia de completamente branco até preto (sem imagem). É muito importante ressaltar que a resolução da imagem está atrelada a capacidade do dispositivo variar essa parte do sinal – variando do nível de preto até o de branco. Um dispositivo que consiga apenas trocar de preto para branco e novamente para preto durante uma única linha horizontal vai gerar apenas uma grande barra vertical branca na tela. Para gerar uma imagem mais complexa são necessárias trocas na casa dos megahertz.

2. Pulso de apagamento. Toda vez que ocorre o retraço, vertical ou horizontal, é necessário que se desligue a saída de imagem para que ela não apresente um risco cortando-a. Esse desligamento é obtido com o pulso de apagamento, que nada mais é que um pulso no nível de preto. Assim quando ocorrem os retraços, nada é desenhado na tela.

3. Pulso de sincronismo. Esse pulso tem a finalidade, como o nome já diz, de manter o sincronismo entre o sinal recebido e a varredura da tela pelo canhão de elétrons.

Somando-se os três sinais, temos o sinal de vídeo composto monocromático:

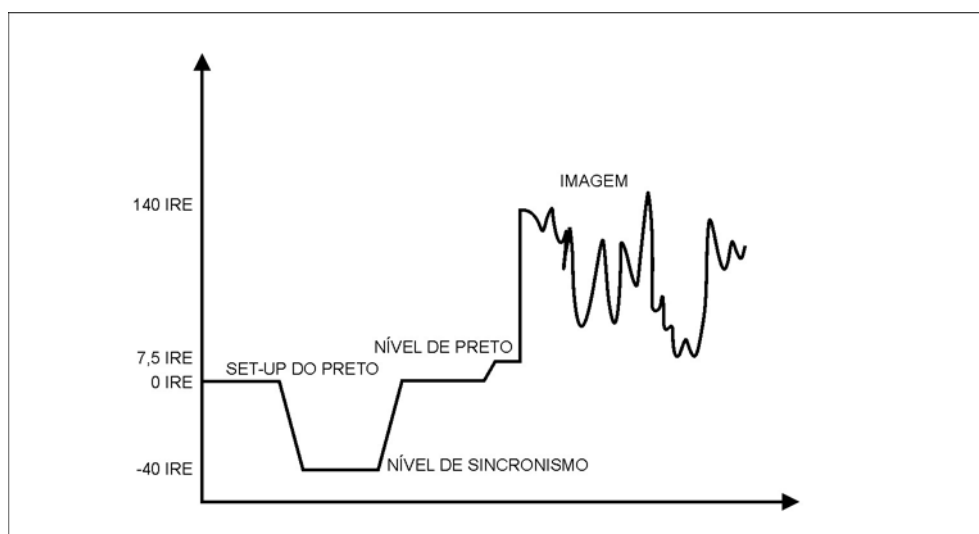


Ilustração 4: Sincronismo Horizontal e imagem.

Para mensurar os sinais de vídeo, foi criada a escala IRE (Institute of Radio Engineers, atual IEEE). Nessa escala, ficam estabelecidas as amplitudes de cada parte do sinal de vídeo. A escala IRE cobre 140 unidades e equivale a 1Vp-p sobre uma carga de 75Ω.

<i>ESCALA IRE</i>	
Amplitude do pulso de sincronismo	-40 IRE
Amplitude do set-up do preto	7,5 IRE
Amplitude do nível de branco	100 IRE

Tabela 1: Escala IRE

Existem equipamentos chamados de monitores de forma de onda, que são osciloscópios que trabalham diretamente na escala IRE.

Para utilizar um osciloscópio comum, é necessária a conversão da escala IRE para níveis de tensão. Como 140 IRE são equivalentes a 1Vp-p, para passar de IRE para mV é necessário dividir o valor por 0,14 e para passar de mV para IRE é só multiplicar.

CONVERÃO ESCALA IRE PARA TENSÃO		
	ESCALA IRE (IRE)	TENSÃO (mV)
Amplitude do pulso de sincronismo	-40	286
Amplitude do set-up do preto	7,5	54
Amplitude do nível de branco	100	714

Tabela 2: Conversão IRE x mV

Os pontos mais importantes no estudo do sinal de vídeo composto são a temporização e a quantidade de pulsos. Os padrões de vídeo definem tempos e quantidades para cada uma das partes do sinal.

3.1.1. Tempo de apagamento Horizontal

O tempo de apagamento horizontal é responsável pela não interferência do retraço com a imagem que está sendo gerada. O tempo necessário para varrer uma linha inteira é dado por $1/15.750\text{Hz}$ que é aproximadamente $63,5\mu\text{S}$. O pulso de apagamento ocupa uns 16% (que é um valor um pouco maior que o tempo gasto no retraço) desse tempo e, portanto, é $10,2\mu\text{S}$.

Subtraindo esse tempo do tempo total para uma linha, temos o tempo disponível para a geração da imagem na tela, que é de $53,3\mu\text{S}$.

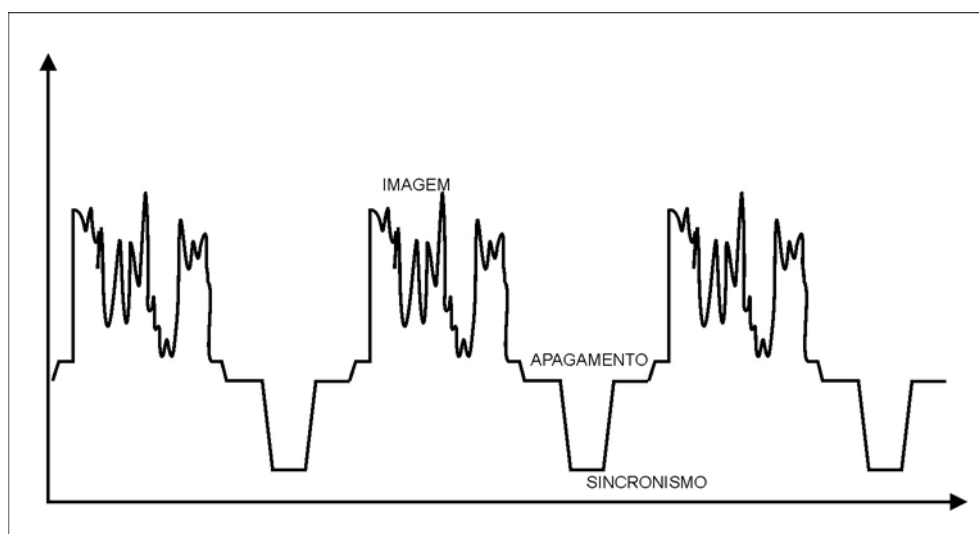


Ilustração 5: Pulsos de apagamento e sincronismo.

3.1.2. Pórtico Anterior e Posterior

O tempo de apagamento engloba o pulso de sincronismo e, para ter garantia de que o retraço ocorrerá sem problemas, além do sincronismo temos um período de apagamento antes (pórtico anterior) e depois (pórtico posterior) do sincronismo. O tempo mínimo do pórtico anterior é de $1,27\mu\text{S}$ e do posterior $3,81\mu\text{S}$.

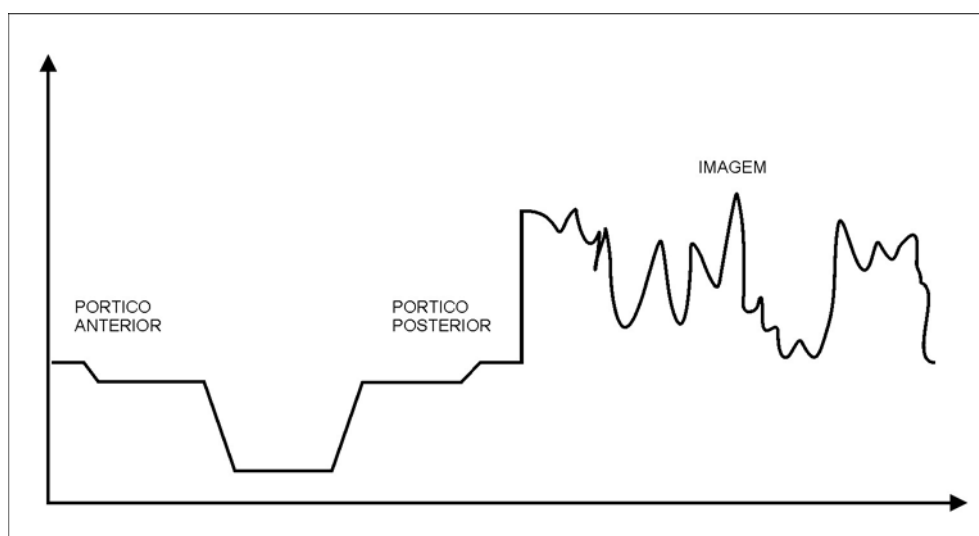


Ilustração 6: Localização dos pórticos anterior e posterior.

3.1.3. Sincronismo horizontal

Descontados os pórtricos anterior e posterior do tempo de apagamento horizontal, sobram $5,12\mu\text{s}$ para o pulso de sincronismo. Porém, como os tempos anteriores eram tempos mínimos, o pulso de sincronismo fica padronizado em $4,7\mu\text{s}$ de duração.

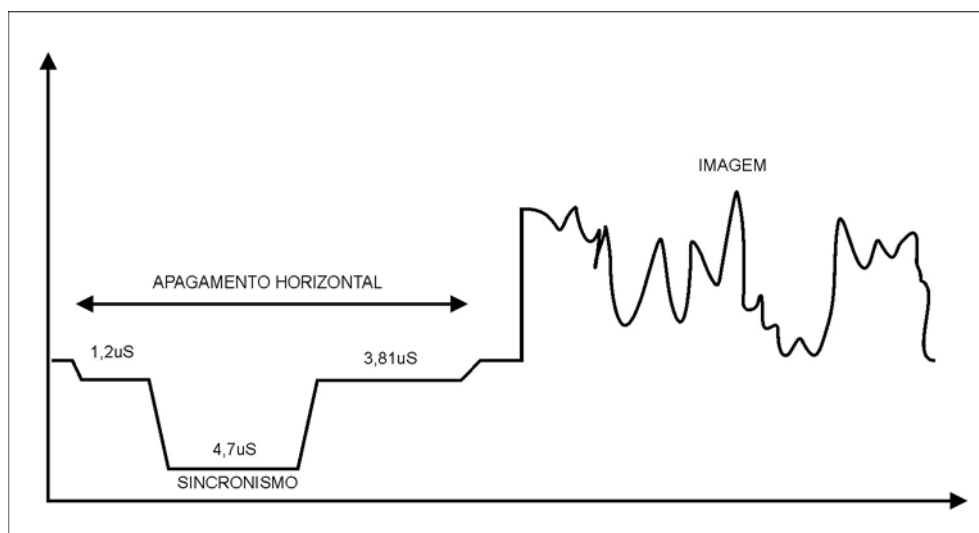


Ilustração 7: Sobreposição do apagamento horizontal sobre o pulso de sincronismo.

3.1.4. Tempo de apagamento vertical

Assim como eram necessários tempos para os retraços horizontais, o retraço vertical também exige um tempo em que a imagem não deverá ser mostrada na tela. Tendo como padrão que o retraço ocupe no máximo 8% do tempo de varredura vertical, teremos um total de $1,33\text{ms}$. Durante o retraço vertical, 21 linhas horizontais são desenhadas na tela (sem que apareçam).

3.1.5. Sincronismo vertical

O sincronismo vertical é feito durante o tempo de apagamento vertical (da mesma forma que ocorria no horizontal) para que não sejam geradas sujeiras na tela. Ele é composto de diversas linhas horizontais especiais que forçam os circuitos de deflexão vertical a começarem o retraço. Porém, como o retraço não começa com o início do pulso de sincronismo vertical (pois ele deve carregar um capacitor para disparar os circuitos de varredura) algumas linhas não são desenhadas no início e no final da tela.

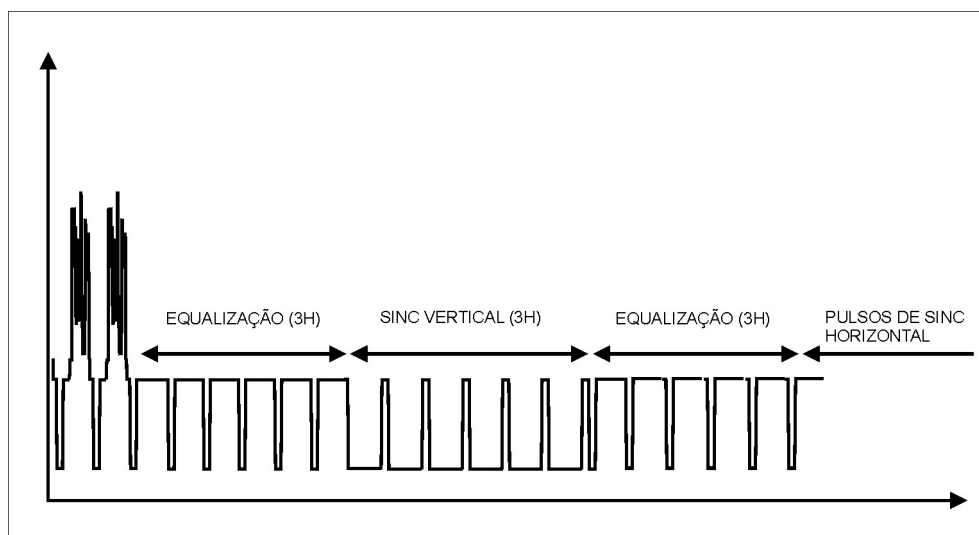


Ilustração 8: Pulsos de equalização e de sincronismos.

A tela fica, descontando os tempos mortos, com a seguinte configuração:

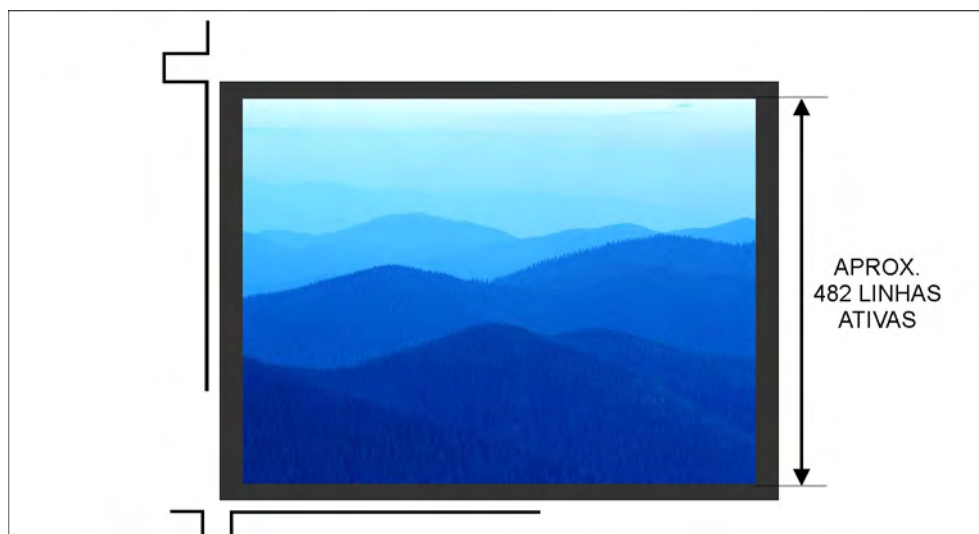


Ilustração 9: Área útil disponível após contabilizar os sincronismos e os apagamentos.

4 ESPECIFICAÇÕES E IDÉIAS INICIAIS

Além de uma boa idéia, um projeto de engenharia necessita de especificações para que os objetivos sejam traçados e seja possível, no final de tudo, saber se estes foram ou não alcançados. Muitos projetos feitos sem essa preocupação chegam ao extremo de perder o foco inicial e não ser possível nem avaliar os resultados. Portanto, para não correr esse risco com o TEXvid, esta seção mostrará como foi traçado o esqueleto básico que foi usado durante todo o andamento do meu projeto.

4.1. A IDÉIA INICIAL

Os primeiros esboços do TEXvid eram no sentido de desenvolver um sistema que possibilitasse sobrepor caracteres e elementos gráficos em um sinal de vídeo já existente para ser utilizado em automação residencial, comercial e industrial. Também era importante que ele tivesse uma interface para comunicação com o mundo exterior que fosse amplamente usada e conhecida. Além disso, era necessário utilizar componentes que fossem facilmente encontrados no mercado nacional e que tivessem um baixo custo de aquisição para que o projeto final fosse comercializável.

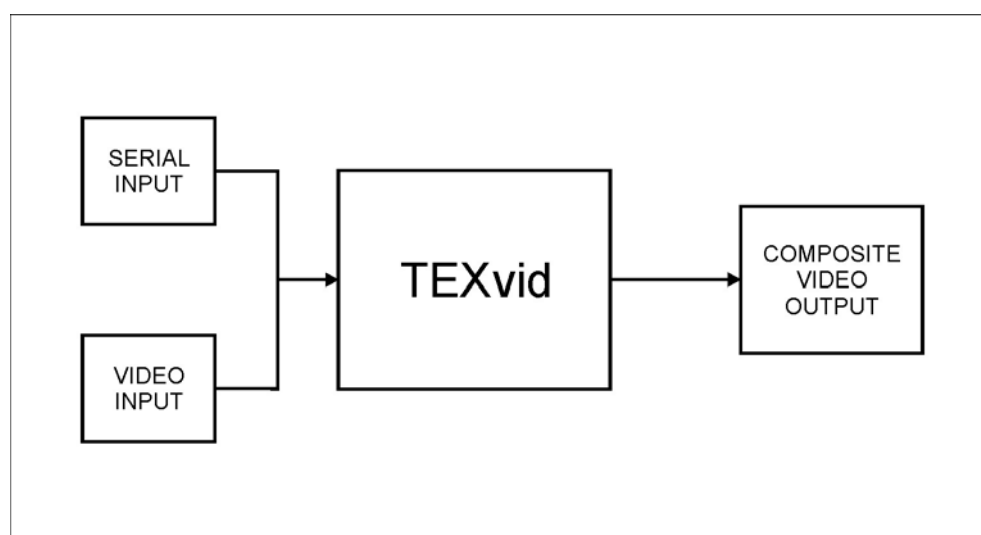


Ilustração 10: Diagrama de blocos do sistema

4.2. PÚBLICO ALVO

Conforme já foi mencionado na introdução, o TEXvid poderá ser utilizado em diversos projetos de automação comercial, residencial e industrial. Para citar alguns:

- Automação de pontos de venda.
- Quiosques multimídia.
- Sistema de avisos.
- Monitoramento de processos industriais.
- Circuito fechado de TV para identificação de pessoas.

Além disso, ele também terá utilidade junto a estudantes e hobbistas, já que custará muito menos que um display de cristal líquido que mostre caracteres com as mesmas dimensões.

4.3. LIMITAÇÕES DO PROTÓTIPO INICIAL

Devido aos prazos apertados, indisponibilidade de equipamentos e componentes, foram necessários alguns ajustes na minha idéia inicial para conseguir gerar uma proposta de projeto que fosse aceita pelo meu professor orientador. Para essa tarefa, foi necessário decidir como simplificar o projeto para ele se encaixar nos requisitos da disciplina do projeto de diplomação, sem perder o foco da idéia inicial, que era gerar um produto comercial.

Para que o meu projeto tivesse a possibilidade de evolução futura, ele deveria ser simplificado, mas sem prejudicar a estrutura básica dele. Então, o que eu fiz foi organizar, na forma de uma tabela, as características desejadas juntamente com as simplificações necessárias.

Características Iniciais	Simplificações	Mudanças necessárias no protótipo para uma implementação completa
Alta resolução para conseguir colocar o máximo de caracteres na tela.	Baixar a resolução para poder utilizar um processador mais barato e disponível no momento.	Utilizar um processador mais rápido.
Escrever na tela inteira.	Escrever somente duas linhas para conseguir utilizar somente a memória do processador.	Colocar um processador com memória RAM maior ou uma memória externa.
Baixo consumo.	Como o protótipo será montado com fonte externa ao invés de baterias (ou alimentação pela serial) esse ponto foi deixado de lado.	Utilizar um processador com baixo consumo (ou que possa reduzir o clock quando for possível) e talvez implementar algo utilizando o modo sleep.
Sobreposição dos caracteres sobre um sinal de vídeo já existente.	No protótipo não será possível a sobreposição de dois sinais.	Como o código fonte já contemplará essa possibilidade, só será necessário um circuito externo para separar os sinais de sincronismo do vídeo (LM1881).
Tamanho reduzido.	Para facilitar a confecção da placa, o tamanho final do protótipo não será o menor possível.	Refazer o layout da PCB para conseguir compactar mais o sistema.
Alta taxa de atualização dos valores para uso no monitoramento de variáveis em processos industriais.	A taxa de atualização ficou limitada a velocidade da serial e também do processador utilizado.	Trocar o processador por um mais rápido e talvez com interface usb.
Preço compatível com as características oferecidas e com a aceitação pelo mercado.	O menor preço possível já que se corre o risco de perder peças nas fases iniciais de testes.	Fazer uma pesquisa junto aos fornecedores para descobrir componentes equivalentes com menores preços.
Compatibilidade com o maior número possível de equipamentos.	O sistema precisará funcionar pelo menos nos equipamentos disponíveis.	Testar em vários modelos e fabricantes para descobrir e corrigir possíveis problemas.

Tabela 3: Características desejadas e simplificações.

Acredito que, mesmo com as simplificações feitas, o projeto não perdeu a sua utilidade. Ainda consigo mostrar mensagens na tela. Também não perdeu a sua complexidade – já que os algoritmos para gerar todos os sinais de sincronismo e para ler diretamente da memória as imagens não foram simples de fazer e o hardware ficou com todos os pontos principais implementados.

5 PROJETO E IMPLEMENTAÇÃO

Ao iniciar o projeto do TEXvid, várias dúvidas surgiram. Porém, a maior dificuldade foi decidir por onde começar. Como até este momento só haviam esboços a respeito do que o sistema deveria fazer e da forma dos sinais de vídeo composto que deveriam ser geradas, eu estava com medo que uma decisão errada pudesse comprometer alguma parte futura. Explicando melhor, como poderia ficar garantido que eu conseguiria gerar os sinais e receber os dados pela serial ou ler os caracteres da memória ao mesmo tempo que precisava manter o sincronismo na tela ou ainda processar os comandos que foram recebidos pela serial enquanto mantinha a imagem sem distorções?

Nesse ponto, uma decisão teve de ser tomada de forma empírica. Não havia tempo e nem recursos de engenharia de software disponíveis para que eu conseguisse fazer todos os testes para não deixar nada “solto” para o final antes de ter começado o trabalho em si. Portanto, as decisões foram tomadas tentando antecipar algumas coisas, mas conforme os problemas apareciam o rumo do projeto sofria mudanças.

Para mostrar uma idéia real de como foi o projeto e a implementação do TEXvid, vou fazer uma descrição também cronológica – e não só baseada diretamente nos blocos funcionais como seria esperado. Assim acredito que o leitor conseguirá perceber que algumas decisões tomadas no início são revistas em certo ponto e, algumas vezes, alteradas fazendo o ciclo de desenvolvimento não seguir uma trajetória em linha reta.

5.1. RESOLUÇÃO

O plano inicial, para um entusiasta da eletrônica como eu, é sempre tirar o máximo possível de tudo – levar os sistemas ao limite. Com a resolução não seria diferente. A busca pela maior resolução possível era uma meta. Mas qual a resolução necessária para mostrar

caracteres na tela? Dessa pergunta tirei duas conclusões: os caracteres precisam ter um número mínimo de pontos para serem legíveis e quanto menor o caracter, mais eu consigo escrever no mesmo espaço disponível.

Descobrir o menor tamanho de letra legível resolveria os dois problemas de uma só vez. Para tentar descobrir quão legível um caractere com poucos pontos pode ser, eu criei uma grande grade que, após ser impressa, serviu de base para teste dos caracteres. Após algumas tentativas, descobri que os caracteres com 3 pontos de largura por 5 pontos de altura eram os menores que possibilitavam diferenciar uma letra da outra sem dificuldade.

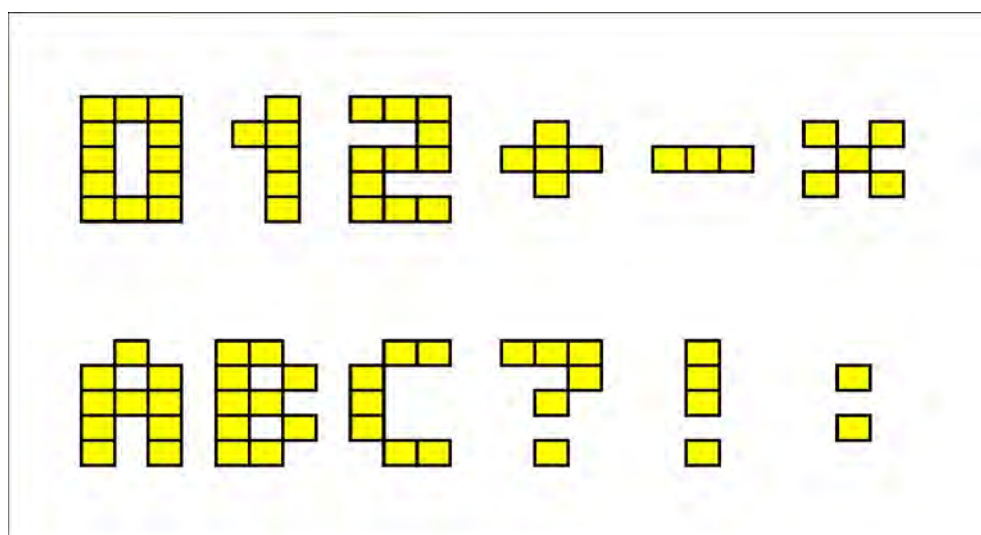


Ilustração 11: Exemplos de caracteres criados.

Nos sistemas de vídeo usados normalmente, o processador escreve em um espaço da memória e um hardware dedicado lê, escreve na tela e gera os sinais de sincronismos. No TEXvid, um microcontrolador, além das outras tarefas, terá de fazer tudo o que normalmente é deixado para o hardware dedicado de vídeo. Isso é o principal fator limitante pois a resolução da saída de vídeo está amarrada a capacidade de fazer a saída analógica variar no tempo.

Para gerar um sinal de vídeo composto usando um microcontrolador é necessária a utilização de um conversor digital – analógico (D/A). Um sinal de vídeo P&B tem o nível de

sincronismo, o de preto e quantos níveis forem possíveis até o branco. Como o TEXvid irá mostrar apenas caracteres monocromáticos, só serão necessários três níveis. Para conseguir gerar três níveis diferentes de tensão, vou utilizar um D/A de 2bits.

Pensando na versatilidade, a utilização de um sistema que leia qualquer coisa que esteja em um espaço de memória e escreva, bit a bit, em uma saída foi a alternativa escolhida. Outra opção seria ler diretamente da memória de programa cada caractere, mas assim não seria possível uma evolução para um sistema gráfico. Portanto, o TEXvid deverá ser capaz de ler bit a bit um espaço na memória e escrever esses bits em um pino específico para gerar os níveis de preto e branco (ou seja, a imagem propriamente dita). O nível de sincronismo será gerado com o segundo pino do D/A.

Conforme visto anteriormente, em uma linha o espaço disponível para a imagem é de aproximadamente $53,06\mu\text{S}$. Logo a resolução da imagem de saída do TEXvid será resultado da capacidade do microcontrolador variar a saída do D/A do nível de preto para o de branco neste tempo. Note que a resolução vertical já é definida pelo próprio padrão PAL-M, e é de 525 linhas.

5.2. ESCOLHA DO MICROCONTROLADOR

Neste ponto não adiantava mais continuar o estudo da resolução, pois seria necessário o início das experiências com algoritmos em assembly. Sem ter escolhido um microcontrolador para utilizar não havia como saber quais instruções estariam disponíveis e quantos ciclos cada uma levaria. Uma primeira estimativa mostrava que, para ler um bit da memória e escrever na saída, seriam necessários aproximadamente uns 5 passos. Por isso, nesse ponto do projeto foram definidas algumas diretrizes para a escolha do processador:

- Capacidade de rodar internamente à velocidade mínima de 5MHz e assim alcançar uma resolução horizontal de 53 pontos.

- Disponibilidade de bibliografia, do compilador / montador e do gravador.

- Facilidade de compra e baixo custo.

- Baixo consumo.

- Tamanho reduzido.

Há algum tempo atrás, eu havia adquirido um livro e um gravador de microcontroladores PIC. Assim, após descartar os modelos da linha 8051 disponíveis para compra, passei a procurar algum processador da Microchip que conseguisse preencher as exigências necessárias. Com essa decisão, aproveitei o projeto do TEXvid para agendar tempo de aprender a utilizar mais essa linha de microcontroladores.

Procurando no site do fabricante, encontrei o modelo PIC16F628A com 18 pinos e que conseguia rodar a 20MHz externamente e internamente a 5MHz. O único problema era que esse modelo tinha uma série que rodava no máximo a 4MHz externos e outra que rodava a 20MHz. Tive de comprar diretamente da Farnell para conseguir os que rodavam a 20MHz, mas isso foi feito via internet, com recebimento por sedex, e não se mostrou um problema.

5.3. ALGORITMOS, MEMÓRIA E, NOVAMENTE, RESOLUÇÃO

Com o microcontrolador definido, foi desenvolvido o algoritmo necessário para ler um bit da memória e escrever no D/A. Usando um algoritmo assim, a imagem ficaria definida como um grande conjunto de “zeros” e “uns” dentro da memória como se fosse a imagem que apareceria na tela. Após muitos testes, o algoritmo que conseguia ler com o menor número de ciclos, mantendo a simetria dos pontos brancos e pretos, ficou assim:

<i>btfs</i>	ENDEREÇO,Nº DO BIT
<i>goto</i>	TERMINA
<i>bsf</i>	PINO_SAIDA


```

goto      TERMINA
bcf       PINO_SAIDA
nop

```

Dessa maneira, consegui gerar leituras simétricas e com 5 ciclos do microcontrolador para ler cada bit. Com isso, a resolução máxima horizontal ficou limitada a 53 pontos. Esse algoritmo deverá ser repetido para cada bit. Portanto, para facilitar a manutenção do código, ele foi escrito na forma de uma macro.

```

SAIDABIT MACRO ENDERECO_INI, ENDERECO_FIN, NBIT, ROTULO
VARIABLE    i=0
;Define uma variável para o compilador
;NÃO ESTARÁ PRESENTE NO CODIGO DE PROGRAMA!
VARIABLE    it=0
;Usada para medir o total de bits
VARIABLE    ENDERECO=ENDERECO_INI
WHILE (ENDERECO<=ENDERECO_FIN) && (it<NBIT)
  WHILE (i<8) && (it<NBIT)
    btfss      ENDERECO,i
    goto      ROTULO+(6*(it+1))-2
    ;SETA PINO
    bsf       saida
    ;Seta o pino de saída pra 1 (nivel de branco)
    goto      ROTULO+(6*(it+1))
    ;LIMPA PINO
    bcf       saida
    ;Seta o pino de saída pra 0 (nivel de preto)
    nop
    i++
    it++
  ENDW ;fim do 2° while
  i=0
  ENDERECO++
ENDW ;fim do 1° while
ENDM ;fim da macro

```

Foi definido nas especificações que o sistema deveria escrever duas linhas de texto na tela. Logo, levando em conta a resolução de 53 pontos, a memória RAM ocupada para escrever duas linhas ficaria em 530 bits (5 x 53 x 2), que são aproximadamente 67 bytes. Como microcontrolador escolhido tem 224 bytes de RAM, não há motivo para preocupação.

Já que há memória RAM disponível, as duas linhas serão geradas e armazenadas na memória com seus valores bit a bit definidos. Depois será utilizado o algoritmo de leitura / escrita de bits para passar da memória para a saída de vídeo.

Para disponibilizar tempo de processamento para as tarefas de recebimento pela serial e de processamento dos dados recebidos, a imagem será gerada usando somente o campo ímpar. Isso significa que a imagem ficará apresentando riscos entre as linhas, pois o padrão PAL-M usa linhas entrelaçadas. Durante o tempo que seria usado para gerar as linhas pares, serão realizados os processamentos referentes a serial e quaisquer outras necessidades.

5.4. INTERFACE SERIAL

Pela minha experiência com microcontroladores, os maiores problemas ocorrem quando se tenta interfaceá-los com outros dispositivos. Foi por isso que atrolei o desenvolvimento do algoritmo de vídeo para iniciar os testes com a serial.

Irei mostrar duas linhas de 13 caracteres na tela. Portanto, precisarei receber no mínimo 26 bytes de informação pela serial. Para configurar a serial, alguns procedimentos são necessários:

- Configurar para Asynchronous Receiver.
- Ajustar o BaudRate (vou utilizar 2400 para manter o erro pequeno).
- A recepção é habilitada pelo CREN (RCSTA<4>).
- A interrupção é ligada / desligada pelo RCIE (PIE1<5>).
- Quando um byte é recebido, o bit RCIF (PIR1<5>) é setado. Esse bit é limpo por hardware quando RCREG é zerado.
- Se os dados não forem retirados do RCREG, ocorre um Overrun quando não houver espaço para o próximo byte, setando o bit OERR (RCSTA<1>).

Como ainda não havia sido montado o hardware do protótipo, foi tentada a alternativa de simulação da serial usando o software Mplab. Porém, após muitas tentativas frustrantes, descobri que a simulação não era funcional para esse modelo de microcontrolador. O

simulador só funcionava direito na simulação para o primeiro byte recebido pela serial, depois ele se perdia e nada mais funcionava.

Nesse ponto, foi necessária a montagem de uma placa de testes para poder verificar os algoritmos referentes ao recebimento pela serial. Para isso, desenvolvi uma pequena placa com um conector DB9, soquete de 18 pinos e um regulador de tensão. Utilizava essa placa junto com uma pequena protoboard para ligar o MAX232 e o PIC via serial com o PC.

O algoritmo para tratar os caracteres recebidos pela serial, simplificada, ficou o seguinte:

- Ocorreu a interrupção da serial.
- Joga o byte para um registrador temporário.
- Testa se o byte corresponde ao caracteres '@'.
- Armazena os caracteres seguintes em um buffer de comandos.

5.5. MONTAGEM DA PLACA PROTÓTIPO

Para fazer os testes da porta serial, com o clock correto, e da geração de vídeo, foi necessária a montagem de um protótipo. Primeiro montei em uma protoboard somente o processador rodando a 20MHz para ver se funcionava sem problemas. Devido ao sucesso dessa parte, passei para a montagem em placa de circuito impresso.

Para montar o layout da placa, precisava decidir alguns pontos pendentes. Um deles era como seria o conversor D/A. Será de 2 bits sendo que um dos bits será responsável pela geração do preto e branco e o outro do sincronismo. Como a geração de imagem não depende do pino do sincronismo, garanto maior simplicidade. No início, cheguei a pensar em criar algo usando o estado de alta-impedância dos pinos, em conjunto com um circuito externo,

para gerar o sincronismo. Porém, como o dreno de corrente era compatível com a capacidade do microcontrolador, decidi pelo uso somente de resistores.

Para fazer um conversor D/A de 2 bits usando resistores preciso de dois pinos livres. Esses, de preferência, precisavam agüentar a corrente drenada e ter um pull-up interno. Analisando os pinos já utilizados (cristal externo e serial), a escolha ficou nos pinos RB3 e RB5.

Como o televisor tem uma entrada de 75Ω e na escala IRE 140 corresponde a $1V_{pp}$ sobre 75Ω , preciso alcançar os seguintes valores:

- Nível de branco: $1V$ ($13,33mA$)
- Nível de preto: $0,33V$ ($4,4mA$)
- Nível de sincronismo: $0V$ ($0A$)

Fazendo os equacionamentos necessários, cheguei aos seguintes valores para os resistores R1 e R2:

- $R1 = 450\Omega$ ($10k\Omega // 470\Omega$)
- $R2 = 900\Omega$ ($2,2k\Omega // 1,5k\Omega$)

Foram verificadas as pinagens dos componentes junto aos datasheets e tomado o cuidado de separar os pinos de cada lado do microcontrolador, além de deixar acessos para todos os pinos usados na parte superior da placa. As ligações do MAX232 foram feitas para que o cabo serial não precisasse ter RX e TX cruzados, pois eles foram cruzados na própria placa. Um capacitor de $100nF$ foi colocado junto aos pinos de alimentação do PIC16F628 para filtrar possíveis ruídos. No final, o layout da placa ficou o seguinte:

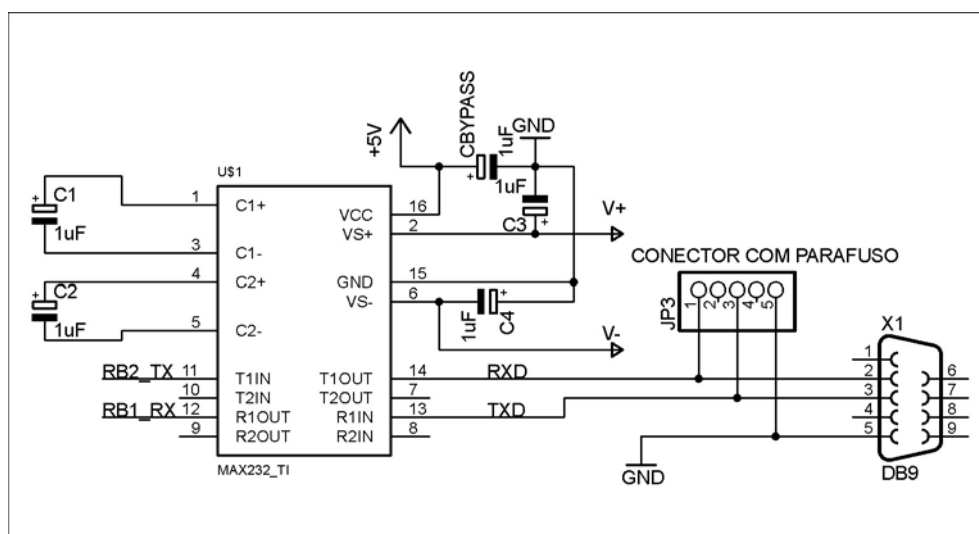


Ilustração 13: Layout da placa: MAX232 e conector DB9.

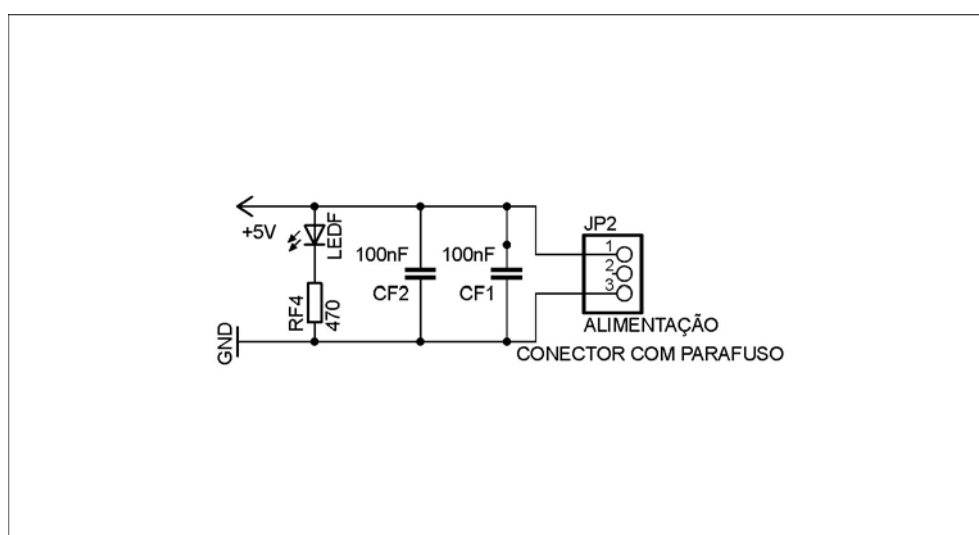
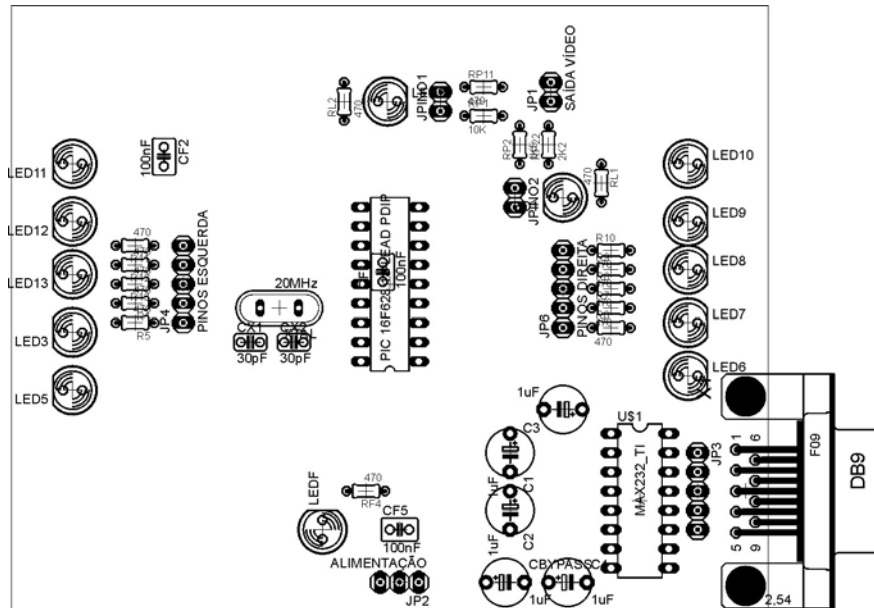


Ilustração 14: Layout da placa: entrada da alimentação.

Após alguns ajustes nas posições dos componentes, a placa de circuito impresso foi roteada com 100% de eficiência. Foi tomado o cuidado para criar uma malha de terra de blindagem na parte do cristal de 20MHz envolta de toda placa, porém não foi fechado nenhum anel de terra.



utilizando um ferro de passar roupa. Após apenas três tentativas, a transferência ficou perfeita e a placa foi corroída.

Nos primeiros testes utilizando o multímetro ajustado para teste de continuidade, a placa aparentou estar perfeita. Após a soldagem de todos os componentes, foi feito novamente esse teste e tudo parecia sem curtos circuitos ou trilhas rompidas.

Quando tentei montar o PIC na placa, ele não oscilava corretamente quando configurado com o oscilador externo (20MHz), mas quando estava usando o oscilador interno ele funcionava sem problemas. Testei com o osciloscópio e descobri que algumas trilhas que deveriam estar aterradas apresentavam 500mV. Utilizando a micro-retífica, alarguei os espaços entre algumas trilhas da placa e voltei a testá-la. Tudo estava funcionando bem. Consegui testar o processador sem problemas.

Nessa primeira fase de testes, percebi que havia esquecido de colocar no layout da placa opção para gravação In-circuit. Soldei os pinos necessários sobre as trilhas e junto ao conector para RJ11. Também adaptei um regulador de tensão LM7805 porque descobri que eu não tinha nenhuma fonte regulada de 5V, somente de 18V.

Após alguns atropelos, a placa protótipo estava pronta, rodando a 20MHz sem apresentar falhas. A partir daí, foi possível o início dos testes da geração dos sinais de sincronismo utilizando um osciloscópio e uma câmera de vídeo para fazer a comparação dos sinais gerados pelo TEXvid com os gerados pela câmera.

Também foi necessária a confecção de um cabo serial para ligar o TEXvid ao PC. Como eu já havia feito a inversão dos pinos TX e RX na própria placa, o cabo ficou somente uma ligação direta entre os pinos 2 (RX), 3 (TX) e o 5 (GND) de ambos os lados.

5.6. GERAÇÃO DO SINCRONISMO HORIZONTAL

Para possibilitar um upgrade futuro do sistema e, daí, conseguir sobrepor os caracteres gerados com um sinal de vídeo externo, foram necessários alguns cuidados na geração dos sinais de sincronismo. Estes não poderiam ficar misturados ao código principal, porque seria impossível fazer as alterações necessárias no futuro. Deveria ser possível trocar o disparador sem maiores problemas para poder fazer os sincronismo com qualquer outro sinal.

Utilizando o Timer 0 do microcontrolador para criar uma interrupção com o tempo de uma linha horizontal (aprox. $64\mu\text{S}$), foi possível fazer com que o sincronismo horizontal ocorresse independentemente do resto do software. Portanto, nunca ocorreria o risco de se perder o sincronismo. Devido ao tamanho do código gerado em assembly, optei por não colocá-lo junto ao corpo do trabalho deixando para consulta nos anexos. Como o código foi exaustivamente comentado, não haverá dificuldade de encontrar os itens mencionados no decorrer do relatório.

5.7. TESTES DO SINCRONISMO HORIZONTAL

Com a placa funcionando e o sincronismo implementado e testado no simulador, era hora de testar os sinais diretamente em um televisor. O funcionamento dos sinais de sincronismo horizontais possibilitam a geração de linhas verticais na tela sem maiores dificuldades.

Já que uma imagem vale mais do que mil palavras, acredito que as fotos tiradas dos sinais de testes do sincronismo horizontal - junto com as legendas - serão suficientes para mostrar como os testes foram feitos.



Ilustração 17: Padrão gerado usando 10101010

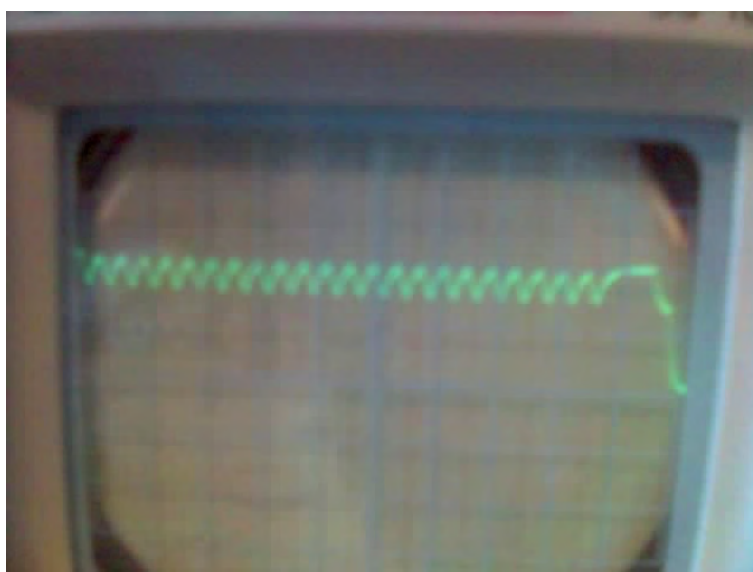


Ilustração 18: Saída de vídeo no osciloscópio trigando as linhas horizontais.

5.8. GERAÇÃO E TESTES DO SINCRONISMO VERTICAL

Após conseguir gerar o sincronismo horizontal, só estava faltando a geração do vertical. O sincronismo horizontal havia sido feito utilizando o Timer 0 para a temporização.

Já o vertical, para simplificar o desenvolvimento, foi feito contando-se o número de linhas de sincronismo horizontal e quando essas alcançavam 482 o Timer 0 era desligado e o sincronismo vertical era realizado. Com o término dos pulsos de sincronismo, recarregava-se o Timer 0 e ele assumia novamente a geração dos sincronismos horizontais.

Para geração de uma linha vertical, ainda foi necessário implementar um contador para poder decidir quando a imagem seria ligada e desligada durante um campo.



Ilustração 19: Linha vertical gerada com o sincronismo vertical.

5.9. RESOLUÇÃO ALCANÇADA E MEMÓRIA CONSUMIDA

Estando o código acertado em relação aos pulsos verticais e horizontais de sincronismo, tive de rever a resolução máxima alcançada e a memória que estava sendo ocupada. Como o algoritmo inicial de geração de imagens não contava com as instruções de tratamento das interrupções, a resolução horizontal caiu para 45 pontos por linha. Para ter um pixel do meu sistema com uma certa simetria, foi definido que cada ponto teria três linhas de altura vertical. Assim eles ficaram quase quadrados.

Na geração de duas linhas de texto serão gastos 45 bits por linha, sendo as linhas com cinco pontos de altura. No total, ocuparei na memória RAM aproximadamente 60 bytes.

5.10. GERAÇÃO DAS LINHAS DE TEXTO

Para gerar as linhas de texto, tive de criar na memória de programa todo um conjunto de caracteres utilizando o valor '1' para um ponto ligado e o '0' para um ponto desligado. Porém, como o sistema desenha uma linha de pontos inteira por vez, foi necessário dividir os caracteres em linhas e calcular o offset para que eles fossem desenhados corretamente linha após linha, formando, no final do desenho da tela, uma letra completa.

Em um primeiro teste da geração direta dos caracteres na tela, tive alguns problemas para definir quando a letra deveria iniciar. Assim, gerei alguns padrões com o texto correto mais algum ruído depois. Esses ruídos, na verdade, são valores na memória lidos na ordem incorreta.



Ilustração 20: Texto gerado com ruídos.

Ajustando o algoritmo de início e final do texto na tela, foi possível criar uma tela na qual o texto aparece na posição correta e não apresenta ruídos. Com mais esse avanço, tudo estava pronto para iniciar a fase do recebimento dos comandos via serial.



Ilustração 21: Texto funcionando corretamente.

5.11. RECEBIMENTO DOS COMANDOS PELA SERIAL

O recebimento dos comandos pela serial será o responsável por alterar os textos exibidos na tela. Essas alterações serão feitas uma única vez, diretamente na memória RAM reservada para a imagem. Para recebimento dos comandos pela serial, poderá ser utilizado qualquer programa emulador de terminal que envie caracteres em ASCII e utilize a configuração 2400-8N1.

Para receber um comando pela serial, este deverá iniciar com o caracter '@'. Após receber o '@', o algoritmo irá armazenar os próximos 12 caracteres em um espaço da memória que foi definido como um buffer de comandos.

Os comandos seguirão o padrão: @<NºdaLinha><texto com 11 letras>.

Quaisquer caracteres que não constem na tabela definida pelo programa serão substituídos por um quadrado preenchido. Caso o usuário pressione a tecla Enter antes de

alcançar os 11 caracteres correspondentes a uma linha, essa linha será preechida com espaços em branco. Como exemplo, para gerar a tela a seguir foram utilizados os seguintes comandos:

```
@110% DE DESC
@2TEXVID
```



Ilustração 22: Exemplo de tela gerada.

Tabela de comandos	
@1	Escreve na linha um
@2	Escreve na linha dois
@\$	Mostra a splashwindow
@R	Reinicializa o programa
@&	Entra no modo gráfico (não implementado)

Tabela 4: Comandos enviados pela serial.

6 TESTES FINAIS

A fase de testes finais iniciou-se com a comparação do sinal de vídeo composto gerado pelo TEXvid com um sinal gerado por uma filmadora portátil. Para esse teste, deixei a lente da filmadora completamente tapada e programei o TEXvid para gerar apenas uma tela sem nada escrito nela.

Foi possível obter um sinal praticamente idêntico – desconsiderando uma variação na amplitude e a portadora referente a cor da imagem que consta na saída da filmadora.

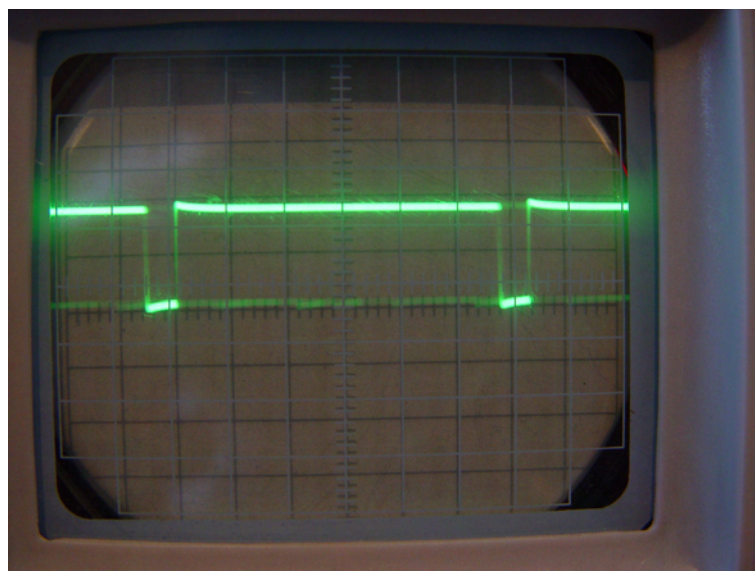


Ilustração 23: Sinal da saída do TEXvid.

Nesta primeira imagem, está o sinal da saída do TEXvid. Ele tem uma amplitude de 5Vpp e essa foto foi feita trigando o osciloscópio nos sinais de sincronismo horizontais. A amplitude pico-a-pico do sinal corresponde à saída com nível alto do microcontrolador, já que o osciloscópio tem uma alta impedância. Quando colocado sobre uma carga de 75Ω , ele fica com as amplitudes corretas.

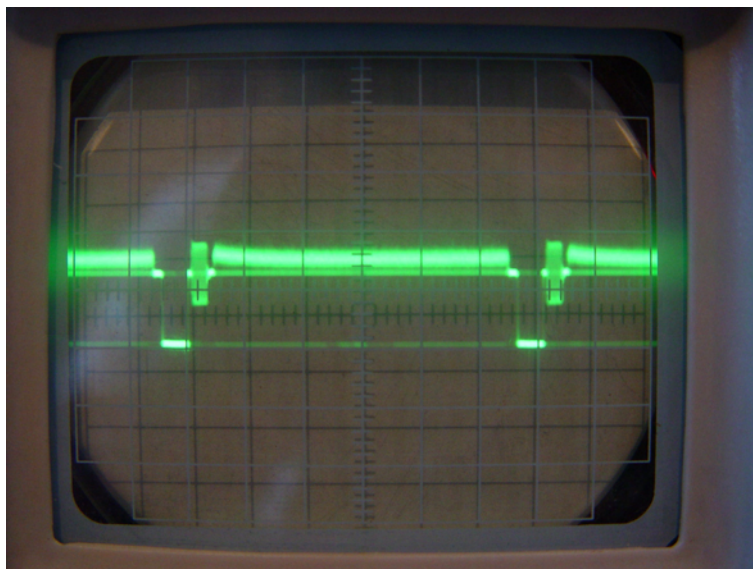


Ilustração 24: Sinal na saída da filmadora.

No sinal da saída da filmadora, a amplitude já está conforme a conversão da escala IRE para mV. Como não consigo gerar um sinal puramente monocromático, aparecem os sinais referentes a informação de cor sobrepostos.

O protótipo também foi testado em aparelhos de televisão das seguintes marcas: Philips, Sharp, Semp, LG, Spaltec e CCE. O único aparelho que apresentou deficiências na imagem gerada foi o da marca CCE. Nesta televisão, as linhas verticais e horizontais ficaram oscilando. Acredito que isso ocorra porque o TEXvid gera linhas horizontais com uma oscilação no tempo total de 200nS. Isso ocorre porque esse é o tempo de 1 ciclo a 5MHz e o estouro do Timer 0 tem essa variação dependendo do ponto do código que ele ocorre.

Para testar a capacidade de geração de todos os caracteres que o sistema tem mapeado, foram feitos testes e verificado que tudo funcionava conforme o projeto. O sistema apresentava algumas falhas no uso de caracteres repetidos se não fosse feita a correta configuração dos bits de parada junto a COM1 do PC.

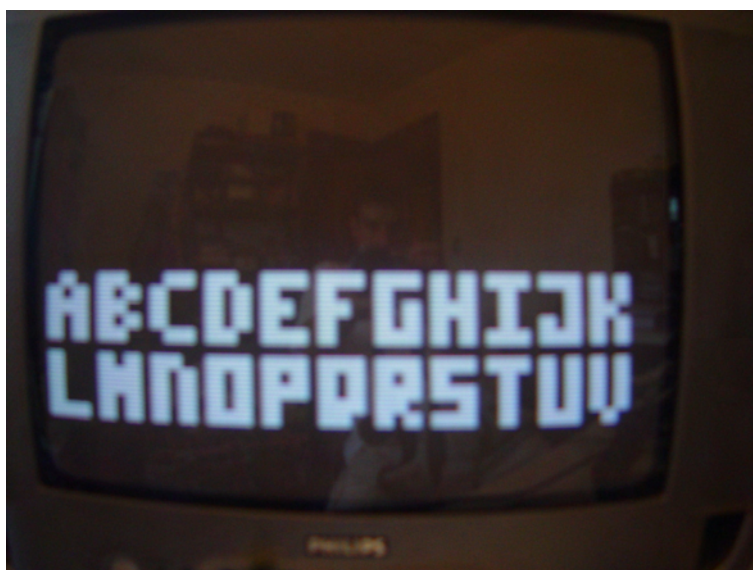


Ilustração 25: 1º Teste dos caracteres.

Na foto acima, foram usados os comandos:

```
@1 A B C D E F G H I J K
@2 L M N O P Q R S T U V
```



Ilustração 26: 2º Teste dos caracteres.

No 2º teste, foram utilizados os comandos:

```
@1 W X Y Z 1 2 3 4 5 6 7
@2 8 9 0 + - * / = ( ) ?
```

E no 3º teste, abaixo, foram utilizados:

```
@1 ! . : % # & & & & &  
@2 TEXVID
```



Ilustração 27: 3º Teste dos caracteres.

É interessante notar que os caracteres '&' não fazem parte do conjunto disponível no TEXvid, portanto foram desenhados com o caractere correspondente a um erro.

7 CUSTOS FINAIS

Uma das principais preocupações, desde o início do projeto, eram os custos. Para que este projeto tivesse a chance de tornar-se algo comercial, deveria ter um baixo custo de produção. Muitas decisões foram tomadas nesse sentido. A escolha do microcontrolador utilizado é um bom exemplo. Caso os custos não fossem um problema, poderia utilizar um componente com maior capacidade de processamento. Para a geração dos sinais de sincronismo também existem circuitos que poderiam fazer o trabalho, mas com isso talvez o projeto ficasse inviável comercialmente.

Outro motivo para se preocupar com os custos é o próprio lucro. Agregar valor ao produto aumenta o lucro, mas diminuir os custos é o mais fácil e que se verifica instantaneamente.

No total, foram gastos no projeto 125 reais. Porém, esse valor corresponde a compra de componentes sobressalentes e equipamentos que foram usados no projeto, mas não contam no custo individual de cada peça fabricada.

<i>Planilha de Custos</i>	
1 Microcontrolador PIC16F628A	R\$ 7,81
1 Cristal 20MHz	R\$ 6,00
1 Soquete torneado 18 pinos	R\$ 3,60
1 Conector para placa DB9	R\$ 1,90
1 Placa virgem de fenolite	R\$ 3,50
1 Caixa plástica	R\$ 3,30
1 Chave monopolar	R\$ 1,80
11 Leds 5mm	R\$ 2,20
1 Jack RCA	R\$ 1,00
1 Jack RJ11	R\$ 1,50
1 MAX232	R\$ 2,77
1 LM7805	R\$ 2,00
10 Resistores carvão	R\$ 0,30
6 Capacitores	R\$ 1,50
Custo Total Aproximado	R\$ 39,18

Tabela 5: Planilha de custos finais, sem considerar o custo de desenvolvimento.

Como é fácil perceber, o custo unitário ficou muito baixo. Levando em conta uma carga tributária de uns 25% e considerando também que o mercado, muitas vezes, é que impõe o preço final, acredito que o TEXvid poderia ser vendido por uns R\$ 200,00. É claro que para vender por esse preço, alguns itens como a sobreposição dos caracteres em um sinal de vídeo externo deveriam estar implementados.

8 MELHORIAS FUTURAS

Melhorias no projeto, normalmente, ocorrem após ele passar por um período de testes maior do que o disponível para este trabalho. Porém, desde o início, já haviam sido tomadas decisões com o intuito de simplificar o projeto que deveriam ser implementadas nas melhorias futuras.

8.1. MICROCONTROLADOR

Para alcançar uma resolução horizontal maior é necessário um componente que consiga trabalhar em uma frequência maior. Uma simples troca por um modelo de microcontrolador da família 18F (do mesmo fabricante) possibilitaria um aumento da frequência interna para, no mínimo, 10MHz. Logo, a resolução horizontal poderia ser dobrada sem um aumento considerável no custo final.

8.2. SOBREPOSIÇÃO DOS CARACTERES SOBRE UM SINAL DE VÍDEO JÁ EXISTENTE

Para conseguir sobrepor dois sinais de vídeo, mantendo o sincronismo correto, é necessário o uso de um circuito especial. Esse circuito se encarrega de extrair os sinais de sincronismo da entrada de vídeo e, assim, facilita o trabalho para o microcontrolador. O próprio microcontrolador poderia se encarregar dessa tarefa utilizando as entradas comparadoras, mas, dessa maneira, a exigência sobre a capacidade de processamento seria ainda maior.

Um circuito bastante usado para a extração dos sinais de sincronismo de um sinal de vídeo composto é o LM1881 da National (um equivalente mais moderno seria o EL4583 da Intersil). O custo adicional ficaria por volta de R\$13.

8.3. CAIXA EXTERNA E PLACA

A confecção da placa de circuito impresso em um processo artesanal não funcionaria para um produto em linha de produção. A utilização de processos industriais para a confecção da placa de circuito impresso garantiria uma diminuição no tamanho, um aumento da confiabilidade, redução dos custos em escala e capacidade de produção de um número elevado de peças. Deste modo, também seria possível a fixação dos resistores e conectores diretamente pelo fornecedor.

Um investimento na caixa externa – com um retrabalho terceirizado ou confecção de uma matriz para injetora – poderia ajudar a incrementar as vendas. É comprovado que produtos que investem em design vendem melhor.

9 CONCLUSÕES

Este trabalho tinha dois objetivos principais: o desenvolvimento de um projeto de engenharia – que na minha concepção não poderia ser outra coisa se não um produto – e o aprendizado. Acredito que ambos foram alcançados com êxito, pois ao final do trabalho o TEXvid funcionava perfeitamente e eu aprendi muito com tudo isso. Após dois meses e mais de duas mil linhas de código em assembly o trabalho foi concluído. Diversos foram os aprendizados que eu tirei dessa experiência. Organização, planejamento, visão empresarial, projeto eletrônico são apenas alguns deles.

Ao propor este projeto, mergulhei numa área que nunca havia estudado antes. Uma vaga idéia: essa seria a melhor descrição para os meus conhecimentos a respeito dos sinais de vídeo. Como não havia cursado a disciplina de Televisão, devido à falta de um horário compatível, tive de estudar tudo a partir do zero. Nessa hora, o auxílio do meu orientador foi imprescindível. Após driblar o susto inicial, encontrar a bibliografia correta e definir as metas, tudo correu bem.

Outra parte gratificante de tudo isso foi perceber que o curso de engenharia elétrica da Ufrgs havia me preparado para encarar os desafios sem medo. Apesar de eu ainda ter muito o que aprender, já que estou iniciando a minha carreira na engenharia, sei que recebi as ferramentas necessárias para construir o meu conhecimento daqui para a frente.

Espero que este trabalho também sirva como mais uma ferramenta. Que ele possa ser usado como base para outros alunos que queiram se aventurar nesta mesma área.

10 REFERÊNCIAS

GROB, B.; **Televisão e Sistemas de Vídeo**. Rio de Janeiro: Guanabara, 1989. 385 p.

MICROCHIP. **PIC16F627A/628A/648A**: Data Sheet. 2005. 178 p.

NINCE, U. S.; **Sistemas de Televisão e Vídeo**. Rio de Janeiro: LTC, 1991. 264 p.

SOUZA, D. J.; **Desbravando o PIC**. São Paulo: Érica, 2003. 267 p.

11 APÊNDICE

Código fonte completo em assembly desenvolvido para o TEXvid.