

DESENVOLVIMENTO DE UMA METODOLOGIA PARA CAPTURA DE IMAGENS USANDO MICROCONTROLADORES

Trabalho de Conclusão de Curso

Engenharia da Computação

Erick Franklin Coelho da Silva
Orientador: Prof. Abel Guilhermino da Silva Filho

Recife, dezembro de 2005.

DESENVOLVIMENTO DE UMA METODOLOGIA PARA CAPTURA DE IMAGENS USANDO MICROCONTROLADORES

Trabalho de Conclusão de Curso

Engenharia da Computação

Este Projeto é apresentado como requisito parcial para obtenção do diploma de Bacharel em Engenharia da Computação pela Escola Politécnica de Pernambuco – Universidade de Pernambuco.

Erick Franklin Coelho da Silva
Orientador: Prof. Abel Guilhermino da Silva Filho

Recife, dezembro de 2005.

Erick Franklin Coelho da Silva

**DESENVOLVIMENTO DE UMA
METODOLOGIA PARA CAPTURA DE
IMAGENS USANDO
MICROCONTROLADORES**

Resumo

A maioria dos sinais encontrados na natureza são contínuos e podem ser transformados em imagens para a apreciação do olho humano. Uma das formas de imagem mais comum no cotidiano é a imagem de vídeo obtida a partir de uma fonte geradora de vídeo composto, como por exemplo, uma câmera de vídeo. Este estudo apresenta uma metodologia para captura de imagens usando microcontroladores. Microcontroladores são microprocessadores com arquiteturas simples e recursos limitados, contudo, a principal vantagem dos microcontroladores são essencialmente as características encontradas de um computador pessoal comum em um único *chip* integrado a preço acessível. Neste trabalho é proposta uma arquitetura de implementação de protótipo de um circuito condicionador para captura, armazenamento e transmissão serial de um quadro de imagem de vídeo composto. O maior desafio deste trabalho refere-se às limitações de recursos de memória e processamento disponíveis nos microcontroladores. Contudo, uma solução satisfatória para problemas complexos usando um circuito integrado simples com um custo relativamente baixo torna a metodologia proposta atrativa.

Abstract

Most of the signals found in the nature are continuous and can be changed into images for the appreciation of the human eye. Nowadays, one of the most common form of image is the vídeo image gotten from a generating source of composite video, for example, a video camera. This study presents a methodology developed to capture images using microcontrollers. Microcontrollers are limited microprocessors with simple architectures and resources, however, the main advantage of the microcontrollers is essentially the similar characteristics found of a common personal computer in a single chip offered by an accessible price. In this work is considered an architecture implementation prototype of a conditioning circuit for capture, storage and serial transmission of a frame of image of composed video. The biggest challenge of this work is about the limitations of resources of available memory and processing in the microcontrollers. However, the proposed methodology is attractive because it is a satisfactory solution for a complex problem using a simple and low cost circuit integrated.

Sumário

Índice de Figuras	v
Índice de Tabelas	vii
Tabela de Símbolos e Siglas	viii
1 Introdução	11
1.1 Objetivo	11
1.2 Organização	12
2 Conceitos Básicos	13
2.1 Vídeo Composto	13
2.1.1 Fundamentos	13
2.1.2 Partes do sinal de vídeo	15
2.1.3 Formatos de Vídeo	17
2.1.4 Codificação de cores	18
2.1.5 Níveis de cores	18
2.1.6 Conceito de varredura entrelaçada	19
2.1.7 Imagem ativa	20
2.1.8 Formação da imagem	20
2.2 Microcontrolador	22
2.2.1 Por que usar um microcontrolador?	23
2.2.2 Microcontroladores vs Microprocessadores	23
2.2.3 Unidade de memória	23
2.2.4 Unidade de processamento central	24
2.2.5 Barramento	25
2.2.6 Unidade de entrada/saída	26
2.2.7 Comunicação serial	26
2.2.8 Temporizador geral (<i>Timer</i>)	27
2.2.9 Temporizador de programa (<i>Watchdog</i>)	27
2.2.10 Conversor analógico-digital	28
2.2.11 Programa	28
3 Etapas do Projeto	31
3.1 Definição da arquitetura	31
3.2 Definição dos componentes	32
3.3 Elaboração do esquema elétrico	32
3.4 Validação do esquema	32
3.5 Elaboração do layout	33
3.6 Confecção do circuito impresso	33
3.7 Montagem da placa	34
3.8 Testes sobre a arquitetura completa	34
4 Implementação	36

4.1	Arquitetura	36
4.2	Componentes	38
4.2.1	Microcontrolador	38
4.2.2	Conversor A/D	39
4.2.3	Separador de Sincronismo	39
4.2.4	Memória SRAM	40
4.2.5	Contador	41
4.2.6	Transceptor	41
4.3	Esquema elétrico	42
4.4	Validação do esquema elétrico	45
4.5	Layout do circuito	46
4.6	Circuito impresso	46
4.7	Montagem da placa	49
4.8	Testes da arquitetura	50
5	Software para Captura de Imagens	52
5.1	Máquina de Estados	52
5.2	Módulos do programa	54
5.3	Formato de armazenamento	56
5.4	Formato de saída	56
5.4.1	Estrutura	57
5.4.2	Cabeçalho do arquivo	57
5.4.3	Cabeçalho da imagem	58
5.4.4	Índice de cores	58
5.5	Transmissão serial	59
6	Conclusões e Trabalhos Futuros	60
6.1	Contribuições	60
6.2	Dificuldades	61
6.3	Trabalhos futuros	61
	Apêndice A – Lista de Componentes	62
	Apêndice B – Experimentos	64
	Apêndice C – Pinos dos Componentes	66

Índice de Figuras

Figura 1. Sincronismo horizontal [6].	11
Figura 2. Referência de <i>burst</i> de cor [6].	14
Figura 3. Sincronismo horizontal e referência de <i>burst</i> de cor [6].	15
Figura 4. Partes do sinal de vídeo [7].	15
Figura 5. Sinal de vídeo entrelaçado no padrão RS-170 [7].	16
Figura 6. Níveis de cores [7].	18
Figura 7. Varredura entrelaçada [6].	19
Figura 8. Apagamento vertical e sinal de sincronismo vertical para NTSC [6].	21
Figura 9. Apagamento vertical e sinal de sincronismo vertical para PAL e SECAM [6].	21
Figura 10. Varredura de um quadro completo NTSC [7].	22
Figura 11. Modelo de memória [8].	24
Figura 12. Modelo de unidade central de processamento com três registradores [8].	25
Figura 13. Conectando a memória e a unidade central de processamento [8].	25
Figura 14. Modelo de uma unidade de entrada/saída [8].	26
Figura 15. Modelo de unidade serial [8].	27
Figura 16. Marcador de tempo para gerar sinais em intervalos de tempo regulares [8].	27
Figura 17. Modelo de um <i>watchdog</i>	28
Figura 18. Bloco para converter uma forma analógica para digital [8].	28
Figura 19. Elementos básicos e conexões internas de um microcontrolador da Microchip[8].	29
Figura 20. Diagrama de bloco do projeto.	36
Figura 21. <i>Pipeline</i> do conversor A/D LTC1406 [13].	39
Figura 22. Diagrama de tempo do componente LM1881 [15].	40
Figura 23. Diagrama de tempo do contador.	41
Figura 24. Esquema elétrico do circuito condicionador.	43
Figura 25. Sinal de vídeo composto (a) e sinal de sincronismo horizontal (b), ambos extraído com uma ferramenta de osciloscópio.	45
Figura 26. (a) Máscara de trilhas da face superior e (b) máscara de trilhas da face inferior do circuito.	47
Figura 27. (a) Máscara de ilhas de soldagem da face superior e (b) máscara de ilhas de soldagem da face inferior do circuito.	48
Figura 28. Máscara de contornos de componentes do circuito.	48
Figura 29. (a) Visão de um furo metalizado e (b) visão de um furo não metalizado.	49

Figura 30. Máquina de estados do programa para captura de um quadro de imagem de vídeo.	53
Figura 31. Controle da captura do sinal de vídeo pelo programa.	53
Figura 32. Organização dos módulos do programa com mapeamento nos respectivos arquivos “.c”.	54
Figura 33. Barramento de dados compartilhado e os sinais: OE_TRA e OE_MEM.	55
Figura 34. Formato de armazenamento do quadro de imagem na memória.	56
Figura 35. Estrutura de um arquivo <i>bitmap</i> (.bmp).	57
Figura 36. Fluxograma da transmissão serial através do microcontrolador.	59

Índice de Tabelas

Tabela 1.	Formatos padrões de vídeo analógico.	17
Tabela 2.	Diferentes níveis de vídeo.	18
Tabela 3.	Imagem ativa.	20
Tabela 4.	Operações possíveis com o projeto do circuito.	36
Tabela 5.	Resumo das características do microcontrolador PIC18F4455.	37
Tabela 6.	Tabela verdade da memória.	39
Tabela 7.	Tabela de função do contador.	40
Tabela 8.	Tabela de função do transceptor.	41
Tabela 9.	Relação de pinos entre os componentes do circuito.	43
Tabela 10.	Lista de testes sobre a parte elétrica do circuito.	49
Tabela 11.	Lista de testes sobre as funcionalidades dos componentes.	49
Tabela 12.	Definição dos sinais OE_TRA e OE_MEM conforme operação desejada.	53
Tabela 13.	Campos do cabeçalho de um arquivo <i>bitmap</i> .	55
Tabela 14.	Campos do cabeçalho de imagem de um arquivo <i>bitmap</i> .	56

Tabela de Símbolos e Siglas

- ↑ – Low-to-High (Nível de transição de subida)
- ↓ – High-to-Low (Nível de transição de descida)
- /CS – Not Chip Select (Não Seleção de Componente)
- /OE – Not Output Enable (Não Permitir Saída)
- /WE – Not Write Enable (Não Permitir Escrita)
- A/D – Analog/Digital (Analogico/Digital)
- AC – Alternate Current (Corrente Alternada)
- ADC – Analog Digital Converter (Conversor Analógico Digital)
- BMP – Bitmap (Mapa de Dígitos Binários – formato de arquivo de imagem)
- C – Sinal de Crominância
- CCD – Charge Coupled Device (Dispositivo de Carga Acoplada)
- CCIR – Comité Consultatif International pour la Radio (Comitê do Conselho de Rádio Internacional)
- CPU – Central Processing Unit (Unidade Central de Processamento)
- CVBS - Color Video, Blank and Sync (Cor de Vídeo, Apagamento e Sincronismo)
- DC – Direct Current (Corrente Contínua)
- DIR – Direction (Direção)
- DSP – Digital Signal Processing (Processamento de Sinal Digital)
- DVD – Digital Versatile Disk (Disco Versátil Digital)
- EEPROM – Electrically-Erasable Programmable Read-Only Memory (Memória Somente de Leitura Programável Apagável Eletronicamente)
- EUSART – Enhanced Universal Synchronous Asynchronous Receiver-Transmitter (Transmissor-Receptor Avançado Assíncrono e Síncrono Universal)
- FS-USB – Full Speed Universal Serial Bus (Barramento Serial Universal de Velocidade Máxima)
- GPRS – General Packet Radio Service (Serviço de Rádio de Pacotes Gerais)
- H – High (Nível alto)
- HCOUNT – Horizontal Count (Comprimento Total de uma Linha)
- HSYNC – Horizontal Sync (Sincronismo Horizontal)
- I/O – Input/Output (Entrada/Saída)
- I²C – Inter-Integrated Circuit (Circuito Inter-Integrado)
- IRE – Institute Radio Engineers (Instituto Engenheiros de Rádio)
- ITS – Insertion Test Signals (Inserção de Sinais de Teste)
- JPG – Joint Photographic Experts Group (Grupo Experiente da União Fotográfica – formato de arquivo de imagem)
- L – Low (Nível baixo)
- NRZ – Non-Return to Zero (Sem Retorno de Zero – formato de pulso para transmissão)
- NTSC – National Television Standards Committee (Comitê de Padrões da Televisão Nacional)
- OE_MEM – Memory Output Enable (Permitir Saída da Memória)
- OE_TRA – Transceptor Output Enable (Permitir Saída do Transceptor)
- PA \bar{L} – Phase Alteration Line (Linha de Fase Alternada)
- PCB – Printed Circuit Board (Placa de Circuito Impresso)
- PCLK – Pixel per Clock (*Pixel* por Frequência de *Clock*)

- PDIP** – Plastic Dual-In Line Package (Linha de Empacotamento Relativo a Plástico)
- QAM** – Quadrature Amplitude Modulation (Modulação pela Amplitude da Quadratura)
- RAM** – Random Access Memory (Memória de Acesso Aleatório)
- RGB** – Red, Green and Blue (Vermelho, Verde e Azul – padrão de cores)
- ROM** – Read Only Memory (Memória Somente de Leitura)
- RS170** – Padrão de vídeo usado nos EUA.
- RS232** – Serial Interface to Receiver/Sender (Interface Serial para Enviar/Receber – padrão para comunicação serial)
- SECAM** – Systeme Electronic Pour Couleur Avec Memoire (Sistema Eletrônico de Cores com Memória)
- SPI** – Serial Peripheral Interface (Interface Periférica Serial)
- SRAM** – Static Random Access Memory (Memória Estática de Acesso Aleatório)
- SSOP** – Small Shrink Outline Package (Desenho de Pacote de Pequeno Encolhimento)
- TCP/IP** – Transmission Control Protocol/Internet Protocol (Protocolo de Controle de Transmissão/Protocolo de Internet)
- USB** – Universal Serial Bus (Barramento Serial Universal – padrão de comunicação serial)
- VSYNC** – Vertical Sync (Sincronismo Vertical)
- W/R** – Write/Read (Escrita/Leitura)
- X** – Don't care (Indiferente)
- Y** – Sinal de Luminância
- Z** – High Z (Nível de alta impedância – 3º nível lógico)

Agradecimentos

Sou grato a todos aqueles que contribuíram para este trabalho. Em particular, a todos os professores do curso de graduação em Engenharia da Computação da Universidade de Pernambuco por suas valiosas informações transmitidas durante o período de aprendizado as quais serviram como base de conhecimento e fonte de inspiração.

Também desejo agradecer especialmente ao meu professor orientador, Prof. Abel Guilhermino, pela dedicação que empregou seu conhecimento, supervisão, incentivo e sugestões aplicadas para a conclusão deste trabalho. Gostaria de agradecer ao professor Carlos Alexandre de Mello pelas orientações iniciais sobre a produção de textos científicos e explicações sobre a importância do trabalho de conclusão de curso.

Sou muito grato aos colaboradores da empresa Segsat Industria Ltda [27]: Newton van der Linden e Carlos Fernando Meira Lins, por me ajudarem nos assuntos sobre eletrônica e desenvolvimento de projeto de *hardware*. Por fim, desejo agradecer a Segsat Industria Ltda por disponibilizar o espaço e as ferramentas do seu laboratório de desenvolvimento os quais foram de fundamental importância para a realização dos meus testes e validações.

Capítulo 1

Introdução

A maioria dos sinais encontrados diretamente na natureza são contínuos. Esses sinais podem ser transformados em imagens que podem ser consideradas descrições de parâmetros que variam sobre uma superfície [1]. Por exemplo, imagens podem resultar da variação da intensidade de luz sobre um plano bidimensional. Entretanto, a intensidade de luz não é o único parâmetro usado em imagens. Uma imagem pode ser formada pela temperatura de um circuito integrado, velocidade do sangue na artéria de um paciente, emissão de raios-X de uma galáxia distante, dentre outros exemplos. Essas imagens são usualmente convertidas em figuras convencionais, por isso elas podem ser avaliadas pelo olho humano.

Em um sistema computacional, por exemplo, mais de 10 MB podem ser necessários para armazenar um segundo de um vídeo de televisão. Isso é mais de mil vezes maior que um sinal de voz de tamanho semelhante. Essas características têm feito o processamento de imagem um subgrupo distinto dentro do Processamento Digital de Sinais [2] - DSP.

O Processamento Digital de Sinais é distinguido das outras áreas dentro da engenharia elétrica pelo tipo único de dados usados: sinais. Na maioria dos casos, esses sinais são originados de sensores de dados do mundo real: vibração sísmica, imagens visuais, ondas sonoras, etc. O DSP se constitui de fundamentação matemática, algoritmos, e de técnicas usadas para manipular esses sinais depois que eles são convertidos em uma forma digital. Isso inclui uma grande variedade de objetivos, como: melhoramento de imagens visuais, reconhecimento e geração de fala, compressão de dados para armazenamento e transmissão, dentre outros.

1.1 Objetivo

A captação da imagem através de um circuito condicionador é um dos processos que antecedem o Processamento Digital de Sinais. Sendo assim, o objetivo deste trabalho é desenvolver uma metodologia para captura de imagens usando microcontroladores. O processo de captura de imagens oriundo da metodologia desenvolvida serve de trabalho preliminar para futuras aplicações nos mais diversos campos, tais como: monitoramento de ambientes, controle de segurança, pesquisas em ambientes que não pode haver intervenção humana, visão em dispositivos remotos, robótica, dentre outras aplicações.

Uma das grandes vantagens do uso de microcontroladores é a flexibilidade e o custo envolvido no desenvolvimento de aplicações que utilizam esta abordagem. A arquitetura de um microcontrolador é composta de periféricos tais como: conversores A/D (*Analog/Digital*),

temporizadores, portas de I/O (*Input/Output*), interfaces seriais e paralelas, entre outros periféricos que auxiliam o projetista no desenvolvimento de projetos de sistemas embarcados [3].

Um dos grandes desafios deste trabalho refere-se primeiro as limitações de recursos disponíveis nos microcontroladores as quais são caracterizadas pelo baixo desempenho de processamento e memória de trabalho pequena em comparação com computadores pessoais comuns. Em segundo, a implementação do circuito condicionador para captura dos sinais. Contudo, uma solução satisfatória para problemas complexos usando um circuito integrado simples com custo relativamente baixo torna esta abordagem atrativa.

1.2 Organização

A metodologia proposta para captura de imagens usando microcontroladores é constituída de seis etapas: (1) Aquisição do sinal analógico de vídeo composto, (2) Conversão do sinal de entrada em digital, (3) Separação dos sinais de sincronismo de vídeo, (4) Armazenamento, (5) Pré-processamento da imagem e (6) Transmissão do sinal digital processado em uma saída serial. A seguir está descrita a organização deste documento.

No Capítulo 2 são apresentados os conceitos básicos sobre as características da formação do sinal de vídeo composto, uma vez que é de fundamental importância conhecer como esse sinal transporta informações sobre sincronismo vertical, sincronismo horizontal, croma e luminância, como também seus padrões e frequências de operação. Um outro conceito básico, também abordado refere-se as características básicas de um microcontrolador como a formação da sua arquitetura e seus componentes como portas I/O, memória RAM (*Random Access Memory*), memória ROM (*Read Only Memory*), saída serial entre outras.

Em seguida, nos Capítulos 3 e 4, as etapas e a implementação do projeto são apresentadas desde a etapa onde a arquitetura da metodologia é elaborada, passando pela escolha dos componentes, elaboração do esquema elétrico, validação dos componentes, elaboração do *layout* da placa até os testes da arquitetura. Esta parte é importante para identificar cada um dos passos utilizados para definir o método de captura de imagens, como também para descrever sua validação através da prototipação de um circuito.

Uma vez que a abordagem escolhida utiliza um microcontrolador para controlar e gerenciar as funções de cada um dos componentes que compõem o circuito condicionador, o Capítulo 5 deste documento é dedicado a apresentar o *software* embarcado. Esse sistema é responsável pelo controle da interação dos recursos de *hardware* para aquisição, armazenamento, processamento e transmissão da imagem digitalizada.

Finalmente, no Capítulo 6 são descritas as conclusões obtidas sobre o projeto e algumas considerações são apresentadas para realização de trabalhos futuros.

Capítulo 2

Conceitos Básicos

Este capítulo tem o objetivo de apresentar os dois principais conceitos acerca do trabalho proposto que estão relacionados à formação do sinal de vídeo composto e a arquitetura básica dos microcontroladores.

2.1 Vídeo Composto

Existem diversos mecanismos para aquisição de imagens. Os mais conhecidos são o scanner, o digitalizador de apenas um quadro de imagem de vídeo (*frame grabber*), o digitalizador do sinal de vídeo (*video grabber*) e a câmera digital [4]. Uma imagem de vídeo gerada por uma câmera ou um videocassete pode ser digitalizada diretamente a partir do sinal elétrico que a representa, que em geral está disponível como uma saída do equipamento.

Embora existam muitas variações e técnicas de implementação, sinais de vídeo são apenas uma forma de transferência de informação visual de um ponto para outro. A informação pode vir de um videocassete, um DVD *player*, um canal de radiodifusão local, ou um sistema de satélite, da Internet ou muitas outras fontes [5].

2.1.1 Fundamentos

A compreensão dos fundamentos do sinal de vídeo é essencial para entender a metodologia proposta neste trabalho.

Inicialmente, o vídeo continha apenas o sinal de luminância (informação em escala de cinza, também conhecida como preto e branco). Enquanto a radiodifusão em cores estava sendo desenvolvida, tentativas para transmitir vídeo colorido usando dados RGB (*Red, Green and Blue*) analógico estavam sendo feitas. Porém, essa técnica ocupava três vezes mais largura de banda que a atual solução em escala de cinza, assim métodos alternativos foram desenvolvidos usando os dados Y (luminância + sincronismo), R-Y and G-Y para representar a informação da cor [5]. A técnica foi então desenvolvida para transmitir as informações Y, R-Y e G-Y usando um único sinal, ao invés de três sinais separados, e dentro da mesma largura de banda como o sinal de vídeo em escala de cinza original. Esse sinal de vídeo composto é o que os padrões de vídeo NTSC (*National Television Standards Committee*), PAL (*Phase Alteration Line*) e SECAM (*Systeme Electronique Pour Couleur Avec Memoire*) ainda tomam por base atualmente.

Em resumo, um sinal de vídeo composto é um sinal no qual todos os componentes requeridos para gerar o sinal de vídeo estão embutidos em um único sinal. Os três componentes principais que juntos formam um sinal composto são os seguintes:

- **Sinal de luminância** - contém a informação de intensidade (brilho e contraste) da imagem do vídeo.
- **Sinal de crominância** - contém a informação de cor da imagem do vídeo.
- **Sinal de sincronismo** - controla a varredura do sinal sobre uma tela, como uma tela de TV.

O sinal composto monocromático é construído com dois componentes: luminância e sincronismo. Esse sinal, usualmente designado de sinal Y, é mostrado na Figura 1, a qual representa uma linha de vídeo. A ordenada do plano de coordenadas cartesianas, na Figura 1, representa o nível do sinal de vídeo cuja unidade de medida é IRE (*Institute Radio Engineers*), essa unidade será descrita com mais detalhes em seções futuras, ainda neste capítulo. Também pode-se observar, na Figura 1, o sinal de sincronismo horizontal, conhecido como HSYNC, que indica o início de uma linha de vídeo.

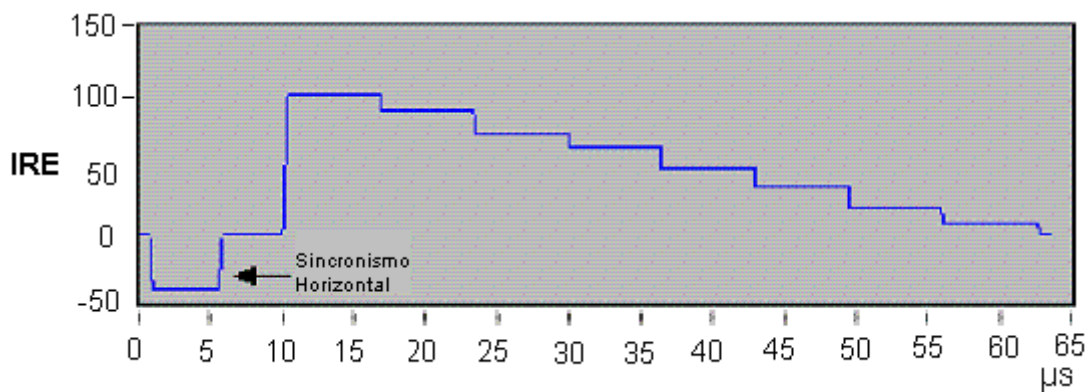


Figura 1. Sincronismo horizontal [6].

O sinal de crominância sozinho, que é usualmente chamado de sinal C, é mostrado na Figura 2. Esse sinal contém uma referência de cor, denominada *burst* de cor, que é utilizada para decodificação da cor com intuito de garantir a fidelidade e estabilidade da cor da imagem. A Figura 2 também mostra a modulação da cor no período de duração de uma linha.

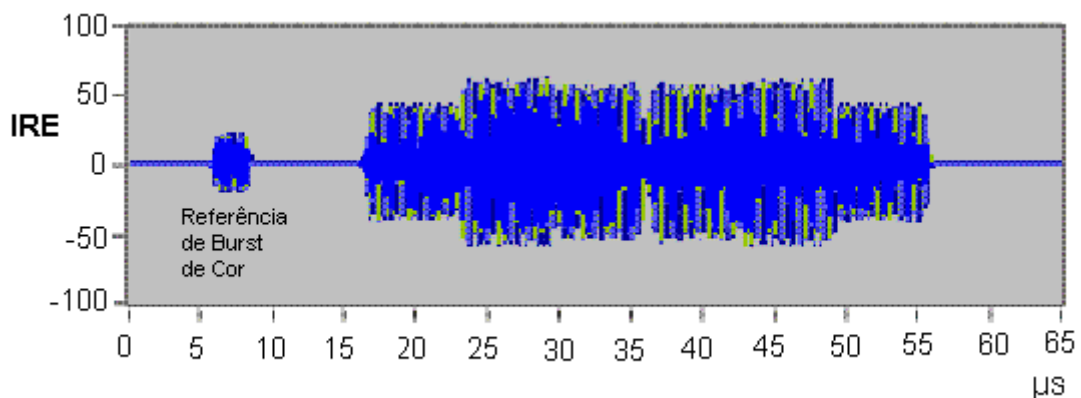


Figura 2. Referência de *burst* de cor [6].

O sinal de vídeo colorido composto, frequentemente chamado de vídeo colorido, apagamento, e sincronismo (CVBS - *Color Vídeo, Blank, and Sync*), é a soma de Y e C conforme mostra a Figura 3 (CVBS = Y + C).

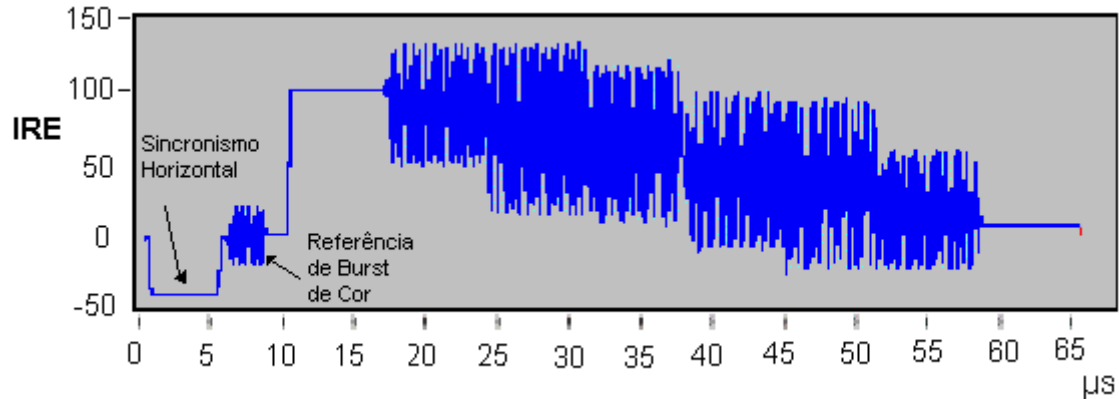


Figura 3. Sincronismo horizontal e referência de *burst* de cor [6].

Os dois componentes Y e C também podem ser distribuídos separadamente como dois sinais independentes. Esses dois sinais juntos são chamados ou de Y/C ou S-vídeo [5].

2.1.2 Partes do sinal de vídeo

O sinal para uma única linha de vídeo horizontal consiste de um sinal de sincronismo horizontal, nível de sinal de referência e vídeo ativo como mostrado na Figura 4.

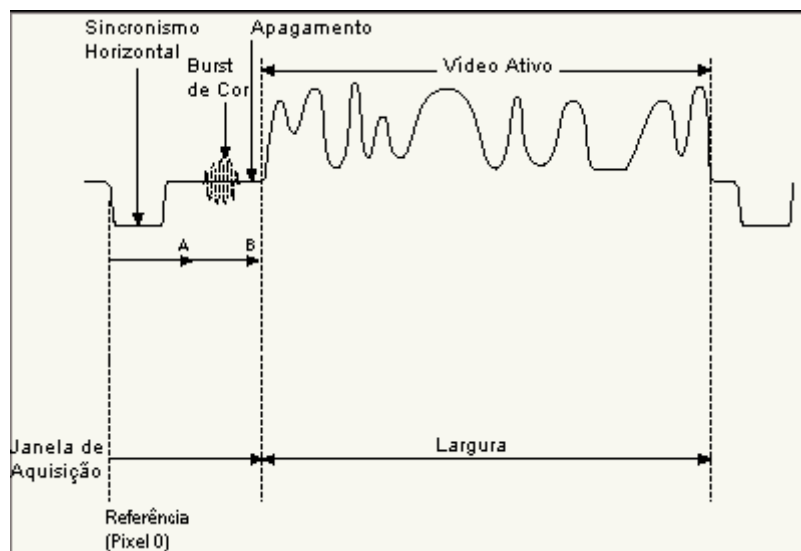


Figura 4. Partes do sinal de vídeo [7].

O sinal de sincronismo horizontal (HSYNC) é inserido a cada nova linha de vídeo. Esse sinal é utilizado pelo dispositivo receptor para identificar o início de cada linha de vídeo. Ele é seguido por um nível de sinal A, conforme a Figura 4, que é usado como um nível de referência para remover quaisquer componentes DC a partir do sinal de vídeo instável (acoplado AC). Isso é consumado durante o intervalo AB. Para sinais coloridos compostos a maior parte do intervalo AB é usado para o *burst* de cor que fornece informação para decodificar o conteúdo da cor do sinal.

Informação de cor pode ser incluída junto com o sinal de vídeo monocromático. Um sinal colorido composto consiste de um sinal monocromático padrão (RS-170 ou CCIR [28]) com a adição dos seguintes componentes:

- *Burst* de cor – localizado na entrada traseira, essa é uma região de alta frequência que fornece uma referência de fase e amplitude para a informação de cor subsequente.
- Sinal de crominância – esse é a informação de cor real. Este consiste de duas quadraturas de componentes modulados sobre uma portadora na frequência do *burst* de cor. A fase e a amplitude desses componentes determinam o conteúdo de cor de cada *pixel*.

Outro aspecto do sinal de vídeo são os pulsos de sincronismo vertical (VSYNC). Esses são de fato uma série de pulsos que ocorrem entre os campos pares e ímpares para sinalizar ao monitor para executar um retraçar vertical e preparar para rastrear o próximo campo, ou seja, indica para o monitor o momento de deslocar o ponto de apresentação da imagem para a parte superior esquerda do vídeo (aponta para o início de um quadro de vídeo). Existem diversas linhas entre cada campo que não contêm informação de vídeo ativo. Algumas contêm apenas pulsos HSYNC, enquanto diversas outras contêm uma série de pulsos de equalização (igualar) e VSYNC. Esses pulsos foram definidos nas primeiras radiodifusões da TV e têm sido parte do padrão desde então, embora a tecnologia de *hardware* recente tenha eliminado a necessidade de alguns pulsos extras. Um sinal entrelaçado composto RS-170 é mostrado na Figura 5, incluindo os pulsos de sincronismo vertical.

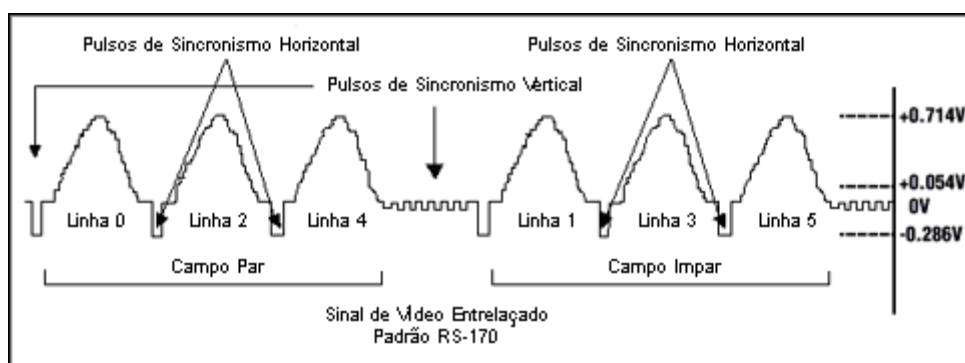


Figura 5. Sinal de vídeo entrelaçado no padrão RS-170 [7].

É importante perceber que o tamanho horizontal (em *pixels*) de uma imagem obtida a partir de uma câmera analógica é determinado pela taxa na qual o *frame grabber* tira amostras de cada linha de vídeo horizontal. A taxa, em fila, é determinada pela taxa de linha vertical e a arquitetura da câmera. A estrutura das câmeras de vetor CCD [4] (*Charge Coupled Device*) determina o tamanho de cada *pixel*. Para evitar distorção da imagem, deve-se fazer a amostragem na direção horizontal em uma taxa que corta a região de vídeo ativo horizontal no número de *pixels* correto. Em seguida, se apresenta um exemplo com números do padrão RS-170:

Parâmetros de interesse:

- Número de linhas por quadro (linhas/quadro): 525 (que incluem 485 para tela; e o resto são linhas VSYNC para cada um dos dois campos)
- Frequência da linha: 15,734kHz
- Duração da linha: 63,556 μ s
- Duração horizontal ativa: 52,66 μ s
- Número de *pixels* por linha (*pixels*/linha): 640

A partir desses dados, alguns cálculos podem ser realizados:

Pixel por frequência de *clock* (PCLK, frequência que cada *pixel* chega ao *frame grabber*):

$$\frac{640 \text{ pixels / linha}}{52,66 \cdot 10^{-6} \text{ s / linha}} \approx 12,15 \cdot 10^6 \text{ pixels / s (12,15MHz)} \quad (1.1)$$

Comprimento total da linha em *pixels* do vídeo ativo + informação de tempo (referido como HCOUNT):

$$63,556 \cdot 10^{-6} \text{ s} \times 12,15 \cdot 10^6 \text{ pixels / s} \approx 772 \text{ pixels} \quad (1.2)$$

Taxa por quadro:

$$\frac{15,734 \cdot 10^3 \text{ linhas / s}}{525 \text{ linhas / quadro}} \approx 30 \text{ quadros / s} \quad (1.3)$$

2.1.3 Formatos de Vídeo

A Tabela 1 descreve algumas características dos formatos padrões de vídeo analógico comumente usados:

Tabela 1. Formatos padrões de vídeo analógico.

Formato	Onde é usado	Modo	Nome do sinal	Taxa de quadros, veloc. de varredura (quad/seg)	Resolução de linha vertical	Taxa de linhas (linhas/seg)	Tamanho da imagem (LxA)
NTSC	América do Norte, América Central e Japão	<i>Mono</i>	RS-170	30	525	15.750	640x480
		<i>Color</i>	NTSC <i>Color</i>	29,97	525	15.734	
PAL	Europa (exceto França), Austrália, partes da África e América do Sul	<i>Mono</i>	CCIR	25	405	10.125	768x576
		<i>Color</i>	PAL <i>Color</i>	25	625	15.625	
SECAM	França, Europa Oriental, Rússia, partes do Oriente Médio e África	<i>Mono</i>		25	819	20.475	N/A
		<i>Color</i>		25	625	15.625	

NTSC: *National Television Standards Committee*.

PAL: *Phase Alteration Line*, o Brasil é o único país no mundo transmitindo no formato PAL com

um sistema de 525/30.

SECAM: *Système Electronic Pour Couleur Avec Memoire*.

2.1.4 Codificação de cores

Para todos os formatos PAL [5] e NTSC [5], a codificação é baseada no conceito da modulação pela amplitude da quadratura (QAM – *Quadrature Amplitude Modulation*) [33], onde componentes de 2-cores são modulados pela amplitude da quadratura e então combinados. A modulação deve ser decodificada, então para guardar a trilha da fase absoluta necessária para decodificar a informação da cor, um sinal de referência chamado *burst* de cor, é inserido no início de cada linha, a direita depois do pulso de sincronização horizontal.

Para o formato SECAM [5], os componentes 2-cores são modulados freqüentemente usando duas freqüências de subportadora diferentes e são distribuídos em seqüência em linhas de vídeo alternadas. O formato SECAM não precisa de um sinal de *burst* de cor.

2.1.5 Níveis de cores

Os níveis de vídeo definem os níveis e variações para as diferentes partes do sinal de vídeo. A unidade para definir níveis de vídeo é a IRE (*Institute Radio Engineers*). O nível de apagamento refere-se a 0 IRE e o nível de branco refere-se a +100 IRE. O nível de apagamento, que é um nível de referência para o sinal de vídeo (comumente 0 V), pode ser diferente do nível de preto conforme mostra a Figura 6.

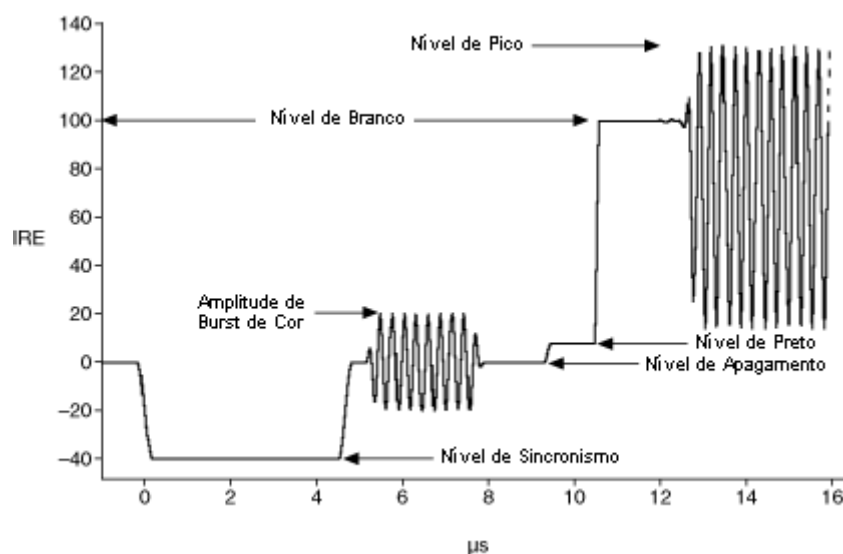


Figura 6. Níveis de cores [7].

Para o formato NTSC, uma configuração de +7,5 IRE é comumente aplicada, que se move do nível de preto até +7,5 IRE. Para os formatos PAL e SECAM, o nível de preto é alinhado com o nível de apagamento em 0 IRE.

A Tabela 2 mostra os diferentes níveis de vídeo dependendo do seu formato:

Tabela 2. Diferentes níveis de vídeo.

Formato de Vídeo	Nível Sinc.	Nível de Apagamento	Nível de Preto	Nível de Branco	Nível de Pico	Amplitude de Burst
NTSC	-40 IRE	0 IRE	+7,5 IRE	+100 IRE	+120 IRE	20,0 IRE
PAL	-43 IRE	0 IRE	0 IRE	+100 IRE	+133 IRE	21,5 IRE
SECAM	-43 IRE	0 IRE	0 IRE	+100 IRE	+130 IRE	N/A

O sinal de vídeo composto é definido como uma fonte de voltagem com uma impedância de 75Ω . O nível *sync-to-white* é normalmente 1 Vpico-a-pico quando carregado com uma resistência de 75Ω . Então, o sinal descarregado é 2 Vp-p nominal.

2.1.6 Conceito de varredura entrelaçada

Todos os sistemas de vídeo composto mostram a imagem de vídeo em uma tela de TV usando uma técnica de varredura entrelaçada. A Figura 7 apresenta o conceito de varredura entrelaçada.

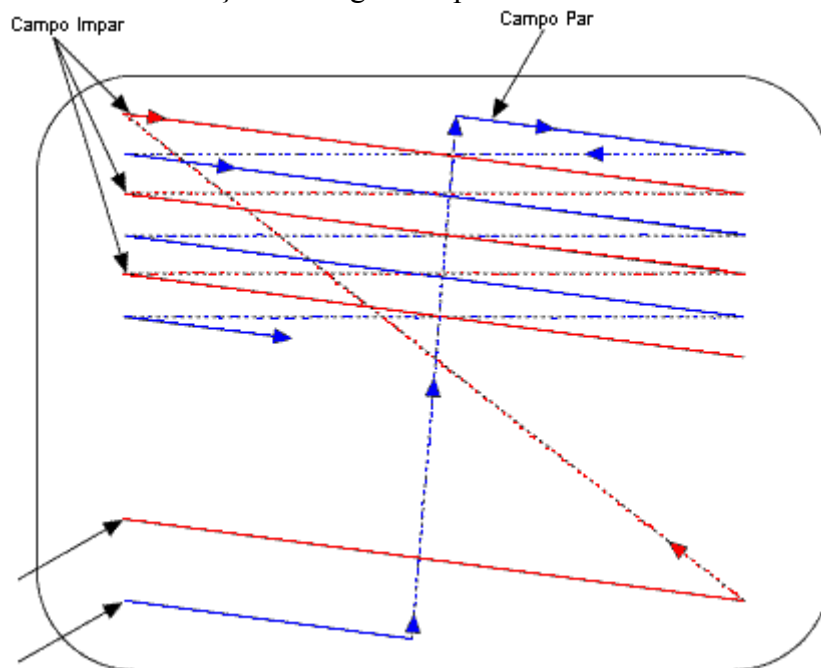


Figura 7. Varredura entrelaçada [6].

O sinal de vídeo analógico inclui pulsos de sincronização que controlam a varredura linha-a-linha da esquerda para direita e campo-a-campo de cima para baixo. Os pulsos que controlam a varredura linha-a-linha são chamados de pulsos de sincronização horizontal (HSYNC). Os pulsos que controlam a varredura vertical são chamados de pulsos de sincronização vertical (VSYNC).

Dois campos entrelaçados compõem um quadro completo. O primeiro campo, chamado de campo ímpar, vane as linhas ímpares da imagem do vídeo. O segundo campo, chamado de campo par, vane a linhas pares da imagem do vídeo. O processo se repete para todos os quadros.

Devido ao entrelaçamento do vídeo, o efeito de imagem piscando na tela é menos aparente que uma imagem apresentada em uma sistema de vídeo não entrelaçado, porque a taxa em que campos sucessivos aparecem na tela é duas vezes maior que a taxa na qual quadros sucessivos podem aparecer em uma tela não entrelaçada contendo o mesmo número de linhas de varredura.

2.1.7 Imagem ativa

A imagem de vídeo ativa resultante da varredura sempre tem uma proporção de aspecto (horizontal/vertical) de 4/3, independente do formato de vídeo [6]. O sinal de vídeo composto colorido mostra que o processo de varredura requer algum espaço adicional na esquerda e na direita de cada uma das linhas, como também no topo e na base da região de imagem de vídeo ativo. Esse espaço adicional inclui os sinais de sincronismo, *burst* de cor, e outra informação de formato específico, como ITS (*Insertion Test Signals*), que não fazem parte da imagem de vídeo ativa. Aproximadamente 90% de todas as linhas transmitem informação de imagem ativa, sendo que cada linha contém apenas 80% em média de informação durante sua transmissão. O valor exato depende do formato de vídeo, como apresentado na Tabela 3.

Tabela 3. Imagem ativa.

Formato de Vídeo	Linhas/Quadro	Linhas Ativas	Taxa de Quadro	Duração de Linha	Duração de Linha Ativa
NTSC	525	480/486	29,97 quad/s	63,55 μ s	52,2 μ s
PAL/SECAM	625	576	25,00 quad/s	64,00 μ s	52,0 μ s

As linhas ativas representam o número de linhas que são realmente usadas para transmitir a informação de imagem. Por exemplo, apenas 480 linhas dentre as 525 linhas por quadro transmitem informação de imagem no padrão NTSC. De forma semelhante, em cada linha, a informação de imagem é transmitida somente durante a seqüência de linhas ativas, que é menor que a duração da linha inteira. Por exemplo, dos 63,55 μ s, apenas 52,2 μ s são a duração da linha ativa em NTSC. A taxa de quadro é a velocidade de varredura.

2.1.8 Formação da imagem

A varredura inicia (linha-a-linha do topo-para-base) com o número de linhas que representam o padrão de sincronização vertical para o campo ímpar. Imediatamente após o padrão de sincronização vertical para o campo ímpar, inserção de sinais de testes (ITS) podem ser incluídos opcionalmente: finalmente, o campo ímpar ativo real aparece. O processo se repete para os campos pares, formando um quadro completo.

A maioria das linhas inicia com um pulso de sincronização horizontal seguida pelo padrão do *burst* de cor. Então, a imagem ativa aparece com uma mudança de intensidade, onde um alto nível de sinal corresponde à intensidade de brilho.

O exemplo de perfil de linha extraído que é exibido nas Figuras 8 e 9, apresentam linhas extraídas de um sinal de vídeo real a partir dos seus campos ímpares e pares. Pulsos de sincronismo horizontal são basicamente pulsos negativos simples. Porém, os sinais de sincronismo vertical são compostos de trens de pulsos distribuídos em diversas linhas, e esses trens de pulso são diferentes para campos ímpares e pares. As Figuras 8 e 9 mostram os padrões de sincronismo vertical para ambos os campos e para os três formatos de vídeo principais.

Existem pulsos que precedem e que sucedem os pulsos de sincronismo vertical. Esses pulsos são denominados pulsos equalizadores cuja freqüência é igual a duas vezes a freqüência de linha. O conjunto formado pelos pulsos equalizadores e pelos pulsos de sincronismo vertical é denominado de apagamento vertical.

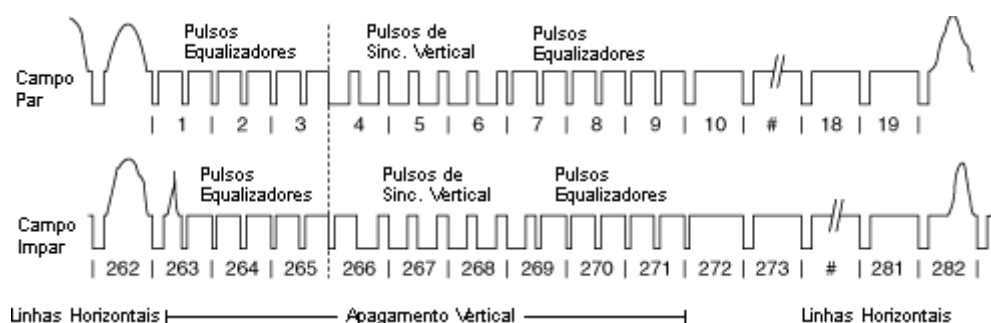


Figura 8. Apagamento vertical e sinal de sincronismo vertical para NTSC [6].

A Figura 8 apresenta o padrão de apagamento vertical para o formato NTSC. Nesse padrão existem seis pulsos equalizadores que antecedem e sucedem os pulsos de sincronismo vertical. O campo ímpar possui um pequeno deslocamento para direita antes do primeiro pulso equalizador em comparação ao campo par.

Em seguida, a Figura 9 apresenta o padrão de apagamento para os formatos PAL e SECAM. Nesses formatos, os pulsos equalizadores que antecedem e sucedem os pulsos de sincronismo vertical são apenas cinco. O campo par possui um pequeno atraso à esquerda em relação ao campo ímpar, diferente do formato NTSC.

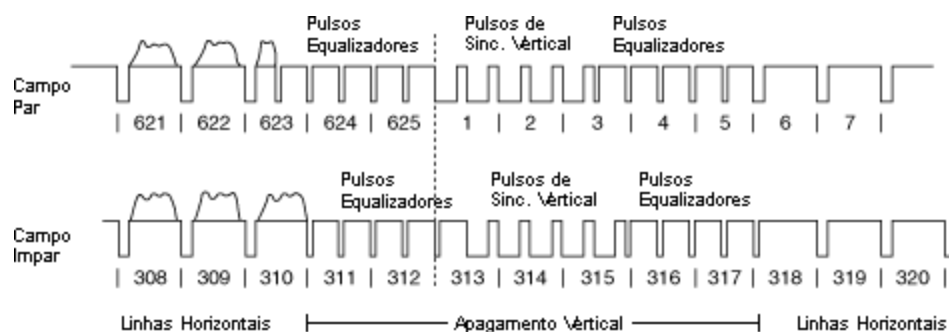


Figura 9. Apagamento vertical e sinal de sincronismo vertical para PAL e SECAM [6].

A imagem mapeada de um quadro completo no padrão de vídeo NTSC é apresentada na Figura 10 e poderia ser exibida em uma tela de televisão se as seguintes condições forem satisfeitas:

- A televisão pode mostrar uma linha inteira, ao invés apenas da parte ativa da imagem.
- A televisão não entrelaça os dois campos para formar o quadro completo da imagem, mas ao invés disso, faz uma varredura progressiva, linha a linha do quadro inteiro.

A Figura 10 mostra o resultado da varredura das 525 linhas que compõe um quadro completo de vídeo no formato NTSC.

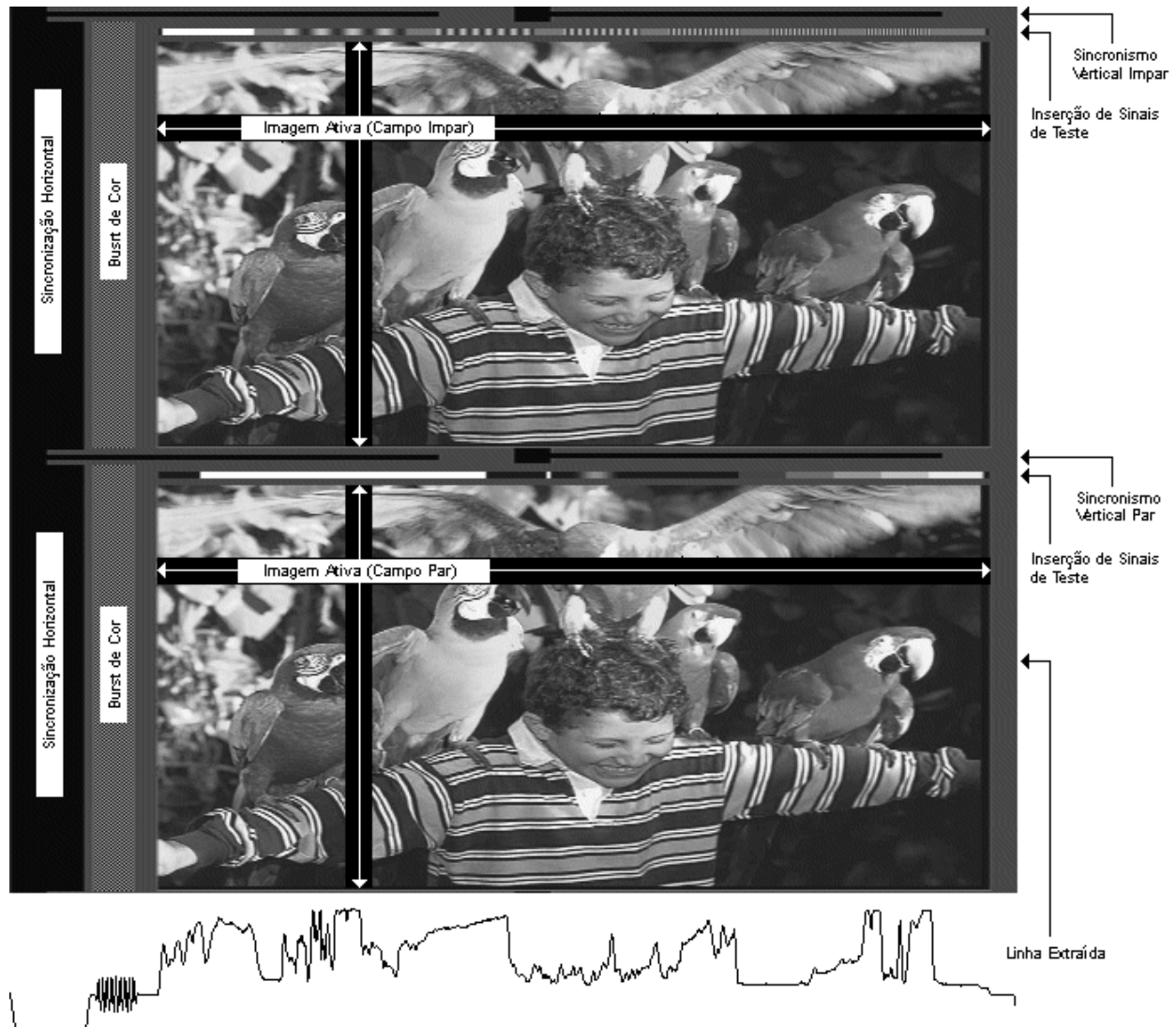


Figura 10. Varredura de um quadro completo NTSC [7].

A Figura 10 é uma imagem em escala de cinza, pois representa a intensidade gráfica de uma forma de onda de vídeo NTSC. A informação de cor está embutida nessa forma de onda, porém ainda não foi decodificada. É possível constatar o *burst* de cor dos sinais no lado esquerdo da Figura 10. O padrão pontilhado, na Figura 10, logo após o sincronismo horizontal, representa a intensidade gráfica do tom do seno que está na forma de onda do *burst* de cor. Depois de decodificada, o *burst* de cor poderia ser visto como uma superfície sólida colorida.

2.2 Microcontrolador

Um microcontrolador é essencialmente um computador de arquitetura simples em um único *chip* a preço acessível. O termo único *chip* significa que um sistema de computador inteiro está dentro de uma área limitada sobre uma pastilha de silício e está encapsulada em um circuito integrado [3].

Um microcontrolador possui características similares às encontradas em um computador pessoal comum. O microcontrolador possui um CPU (*central processing unit*), RAM (*random access memory*), ROM (*read only memory*), portas de I/O (*input/output*), portas serial e paralela,

temporizadores (*timers*), e às vezes alguns outros periféricos embutidos como conversores analógico-digital e digital-analógico. O microcontrolador também têm a capacidade de carregar, armazenar e executar um programa, tornando-o uma solução simples e eficiente para controle.

2.2.1 Por que usar um microcontrolador?

Sendo um computador barato em um único *chip*, os microcontroladores são facilmente embutidos dentro de projetos de circuitos eletrônicos maiores. Sua habilidade de armazenar e executar programas únicos torna-os extremamente versáteis. Por exemplo, um programa de um microcontrolador pode tomar decisões e executar funções baseado em situações (portas de I/O) e eventos. Suas funções matemáticas e lógicas permitem ao microcontrolador imitar sofisticadas lógicas e circuitos elétricos.

Os programas também podem fazer o microcontrolador se comportar como uma rede neural e/ou um controlador de lógica *fuzzy*. Os microcontroladores fazem parte do mercado consumidor de eletrônica e são responsáveis pela “inteligência” dos atuais dispositivos eletrônicos inteligentes.

2.2.2 Microcontroladores vs Microprocessadores

O microcontrolador difere de um microprocessador em diversas formas. A primeira, e mais importante, está nas suas funcionalidades. Um ou mais microprocessadores comumente servem como a unidade de processamento central de um computador.

Quando um projeto de *hardware* usa um microprocessador, componentes como memória RAM, memória ROM, unidades de entrada/saída precisam ser adicionados ao projeto, uma vez que o microcontrolador provê apenas a unidade lógica-aritmética, registradores e a unidade de processamento central.

Por outro lado, o microcontrolador está projetado para ser tudo isso em um só. Nenhum componente externo de memória RAM, memória ROM, portas I/O, comunicação serial são necessários para a utilização de um microcontrolador, porque todos os periféricos necessários já estão embutidos nele. Então, tempo e espaço necessários para construir uma aplicação são economizados.

É importante observar que mesmo com a utilização de um microcontrolador, alguns componentes externos são necessários dependendo da natureza do projeto a ser construído. Uma aplicação que utiliza um microcontrolador pode necessitar de uma memória RAM externa para armazenar uma grande quantidade de dados, por exemplo: vários quadros de imagem de vídeo.

2.2.3 Unidade de memória

A memória é a parte do microcontrolador que é utilizada para armazenar os dados, programa e pilha. A forma mais simples de explicá-la é descrevendo-a como um grande armário com muitas gavetas. Se supusermos que as gavetas estão marcadas de uma forma que não possam ser confundidas, qualquer um dos seus conteúdos pode ser facilmente acessado. É suficiente conhecer a nomeação da gaveta para com certeza conhecer o seu conteúdo.

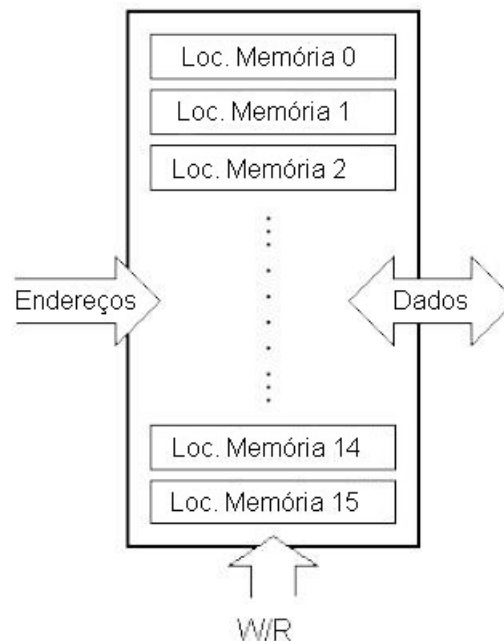


Figura 11. Modelo de memória [8].

Os componentes da memória são semelhantes aos do modelo apresentado na Figura 11. Para uma determinada entrada obtemos o conteúdo de uma localização de memória endereçada. Nesse momento, são trazidos dois novos conceitos: endereçamento e localização de memória. A memória consiste de todas as localizações de memória, e endereçamento é simplesmente a seleção de uma dessas localizações. Isso significa, por um lado, que precisamos selecionar a posição de memória desejada, e por outro lado, precisamos aguardar pelo conteúdo dessa localização. Junto à leitura de uma localização de memória, a memória também deve fornecer uma maneira de gravar dados nela. Então, isso é permitido através de uma linha de sinal adicional chamada linha de controle. Estamos designando-a como linha W/R (*Write/Read*).

A linha de controle é usada da seguinte forma: se $W/R = 1$, a leitura está pronta. Caso contrário a escrita ocorre na localização de memória.

Esse tipo de memória é conhecido como memória de dados ou memória RAM, contudo existe também um tipo de memória utilizado para armazenar o programa do microcontrolador que é conhecida como memória ROM.

2.2.4 Unidade de processamento central

A CPU (Unidade Central de Processamento) é responsável pelo processamento do microcontrolador. Basicamente, o processamento consiste na execução das instruções do programa, nas operações aritméticas e no controle de algumas localizações de memória auxiliar interna.

Considere, por exemplo, três localizações de memória adicionais que especificam um bloco que tem a capacidade embutida de multiplicar, dividir, subtrair, e mover o conteúdo de uma localização de memória para outra. Estas tarefas são processadas pela unidade central de processamento. As localizações de memória auxiliar interna são chamadas de registradores. A Figura 12 mostra um modelo básico de CPU com três registradores.

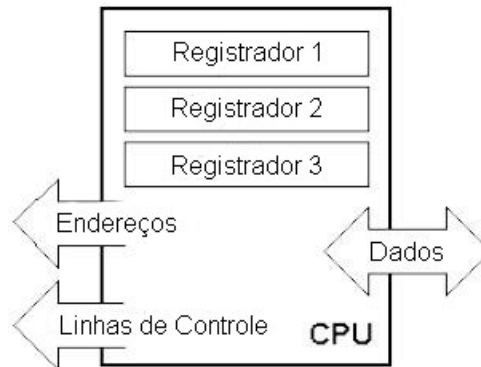


Figura 12. Modelo de unidade central de processamento com três registradores [8].

Registradores são, portanto, localizações de memória com a função de ajudar com as execuções de diversas operações matemáticas ou qualquer outra operação com dados sempre que dados estejam incluídos. A CPU e a memória são duas entidades independentes que estão interligadas. Se, por exemplo, desejarmos somar os conteúdos de duas localizações da memória RAM e devolver o resultado para uma outra localização da memória RAM, é necessário uma conexão entre a CPU e a memória. Simplesmente, precisamos de algum “caminho” pelo qual os dados passam de um bloco para outro.

2.2.5 Barramento

Esse “caminho” é chamado de barramento, conforme mostra a Figura 13. Fisicamente, isso representa um grupo de 8, 16 ou mais fios. Existem dois tipos de barramentos: endereçamento e dados. O primeiro, consiste de tantas linhas quanto o montante de memória que se deseja endereçar, e o segundo tipo depende dos dados. O primeiro serve para transmitir endereços a memória da CPU e o segundo para interligar todos os blocos dentro de um microcontrolador.

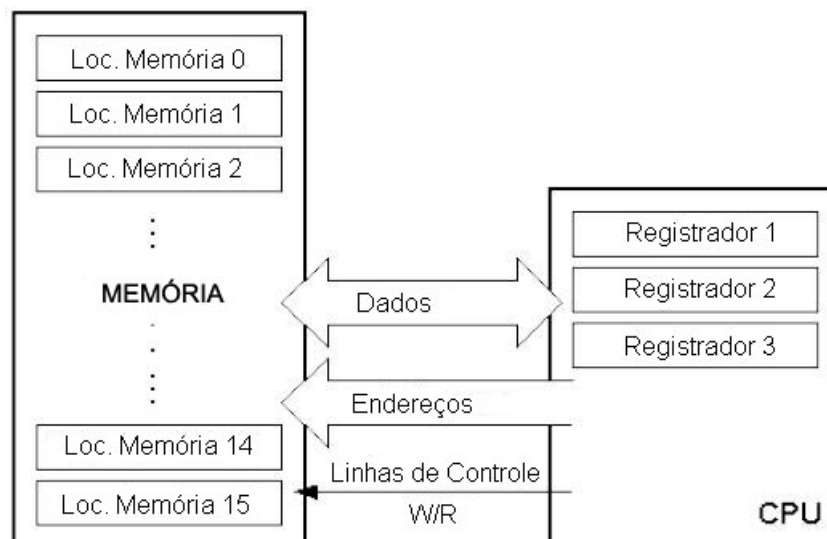


Figura 13. Conectando a memória e a unidade central de processamento [8].

Foi apresentada a unidade de CPU que é capaz de trabalhar, mas que não tem nenhum contato com o mundo externo. Com intuito de resolver essa deficiência, adicionalmente existe uma unidade de entrada/saída que contém diversas localizações de memória à qual uma

extremidade está ligada ao barramento de dados e a outra está ligada as linhas de saída do microcontrolador que podem ser vistas como pinos do componente eletrônico.

2.2.6 Unidade de entrada/saída

As localizações citadas na seção anterior são chamadas de portas. Existem diversos tipos de portas: entradas, saídas e portas bidirecionais, conforme mostra a Figura 14. Quando se trabalha com portas, primeiramente é necessário escolher qual porta atende às necessidades de transmissão, e então enviar dados para, ou receber dados a partir da porta.

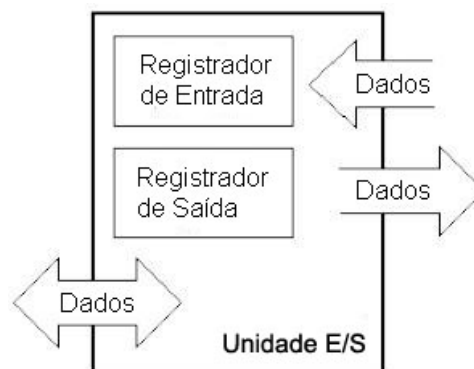


Figura 14. Modelo de uma unidade de entrada/saída [8].

As portas de entrada/saída agem como localizações de memória, ou seja, alguma coisa está simplesmente sendo escrita ou lida, e isso pode ser observado nos pinos do microcontrolador.

2.2.7 Comunicação serial

Conforme descrito acima, existe mais de uma unidade que possibilita a comunicação com o mundo externo. Porém, essa forma de comunicação tem suas desvantagens. Uma desvantagem básica é o número de linhas necessárias para transmissão de dados.

Suponha que se trabalhe apenas com três linhas. Uma linha seria usada para enviar dados, a outra linha usada para receber dados e a terceira linha seria usada em ambos os casos como uma linha de referência de entrada/saída, conforme mostra a Figura 15. Existe também comunicação serial com apenas duas linhas: linha de recepção e linha de transmissão. Para que essa abordagem funcione, é necessário definir algumas regras para troca de dados. Essas regras são chamadas de protocolo. O protocolo é definido antecipadamente, assim não existirá qualquer desentendimento entre os lados que estão se comunicando.

Por exemplo, pode-se considerar o seguinte protocolo: o valor lógico 1 é definido na linha de transmissão até que a transferência inicie. Baixa-se o valor lógico da linha de transmissão para 0 por um período de tempo T , então o lado receptor saberá que está recebendo dados, e logo ativará seu mecanismo receptor. Voltando para o lado transmissor e iniciada a colocação de níveis lógicos baixos e altos na linha de transmissão na ordem do *bit* menos significativo para o *bit* mais significativo, cada *bit* permanece na linha por um período de tempo T , e no final, ou depois do oitavo *bit*, o nível lógico 1 é trazido de volta para linha que marcará o final da transmissão de dados. O conjunto de passos descritos é conhecido como código de linha NRZ (*Non-Return to Zero*).

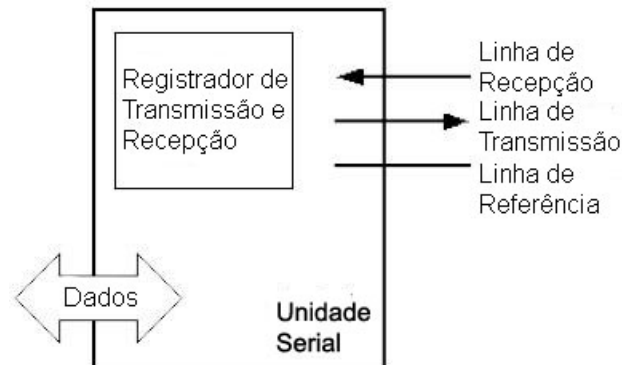


Figura 15. Modelo de unidade serial [8].

Como as linhas estão separadas para o recebimento e o envio de dados, é possível receber e enviar dados ao mesmo tempo. Conhecida como modo *full-duplex*, essa forma de comunicação habilita o bloco chamado de comunicação serial. Diferente da transmissão paralela, os dados movem-se *bit a bit*, ou em séries de *bits*, que definem o termo comunicação serial originalmente.

Depois da recepção dos dados, é necessário lê-los a partir de uma localização de recebimento e armazená-los em uma memória. Por outro lado, quando o processo tem o objetivo de enviar bytes, os dados são obtidos da memória através do barramento de dados e são transferidos para a localização de envio. Em seguida, os dados são enviados para unidade de recebimento conforme o protocolo.

2.2.8 Temporizador geral (*Timer*)

Uma vez que a comunicação serial foi explicada, pode-se enviar, receber e processar dados.

Porém, com intuito de utilizar o microcontrolador na indústria é necessário adicionar alguns poucos blocos. Um deles é o bloco de *timer* como mostra a Figura 16, ou bloco marcador de tempo, que é importante por fornecer informações sobre tempo, duração, protocolo etc. A unidade básica de um *timer* é um contador livre de execução que é de fato um registrador o qual incrementa um valor numérico de uma unidade em intervalos, assim captura seu valor durante o período t_1 e t_2 e com base em suas diferenças, é possível determinar quanto tempo se passou. Essa é uma parte do microcontrolador muito importante que solicita requisições a maior parte do tempo.

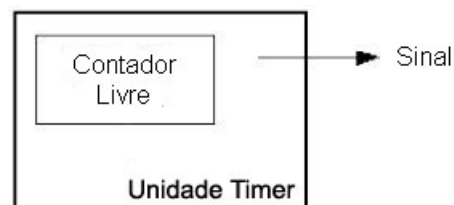


Figura 16. Marcador de tempo para gerar sinais em intervalos de tempo regulares [8].

2.2.9 Temporizador de programa (*Watchdog*)

Uma coisa a mais que requer atenção é a ausência de falha funcional do microcontrolador durante seu tempo de execução. Suponha que como resultado de uma interferência (que frequentemente ocorre na indústria) o microcontrolador páre a execução do programa, ou pior, ele reinicie trabalhando incorretamente.

Certamente, quando isso acontece com um computador, simplesmente o usuário reinicia o sistema e, normalmente, ele volta a trabalhar. Porém, não existe um botão para reiniciar que pode ser pressionado no microcontrolador para resolver o problema. Para esse obstáculo, mais um bloco precisou ser adicionado ao microcontrolador, o qual é chamado de *watchdog*, conforme mostra a Figura 17. Esse bloco é de fato outro contador que executa livremente, onde o programa precisa escrever zero o tempo todo para que ele execute corretamente.

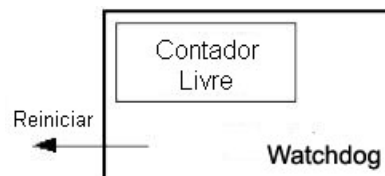


Figura 17. Modelo de um *watchdog* [8].

Nos casos em que o programa fica “preso”, o zero não será escrito. Logo, o contador, que executa livremente, alcançará seu valor máximo e conseqüentemente enviará um sinal que irá reiniciar o microcontrador automaticamente. Isso resultará na execução do programa novamente.

Esta característica é um elemento importante de todo programa que seja confiável sem supervisão humana [8].

2.2.10 Conversor analógico-digital

Como os sinais periféricos são substancialmente diferentes daqueles que o microcontrolador pode entender (zero's e um's), eles precisam ser convertidos em um padrão que um microcontrolador possa compreender. Essa tarefa é desempenhada por um bloco para conversão analógico-digital, ou simplesmente, um bloco ADC. Esse bloco, conforme mostra a Figura 18, é responsável por converter uma informação sobre algum valor analógico para um número binário e encaminhá-lo ao bloco CPU. Então, o bloco CPU pode posteriormente processá-lo.

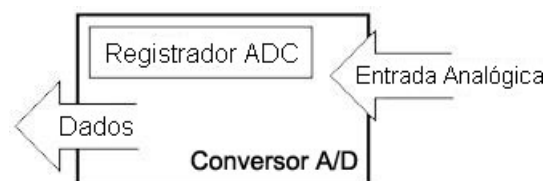


Figura 18. Bloco para converter uma forma analógica para digital [8].

2.2.11 Programa

A Figura 19 apresenta a arquitetura completa de um microcontrolador. Essa arquitetura mostra a conexão entre os componentes internos citados nas seções anteriores: unidade serial, conversor A/D, unidade de E/S, memória, CPU, *timer* e *watchdog*.

De acordo com a natureza da aplicação que será implementada com o uso da arquitetura dos microcontroladores, alguns componentes externos devem ser inseridos no projeto. Além disso, um microcontrolador necessita de um programa que pode seja executado.

Esse programa serve para informar a seqüência lógica de instruções que serão executadas com intuito de alcançar algum objetivo.

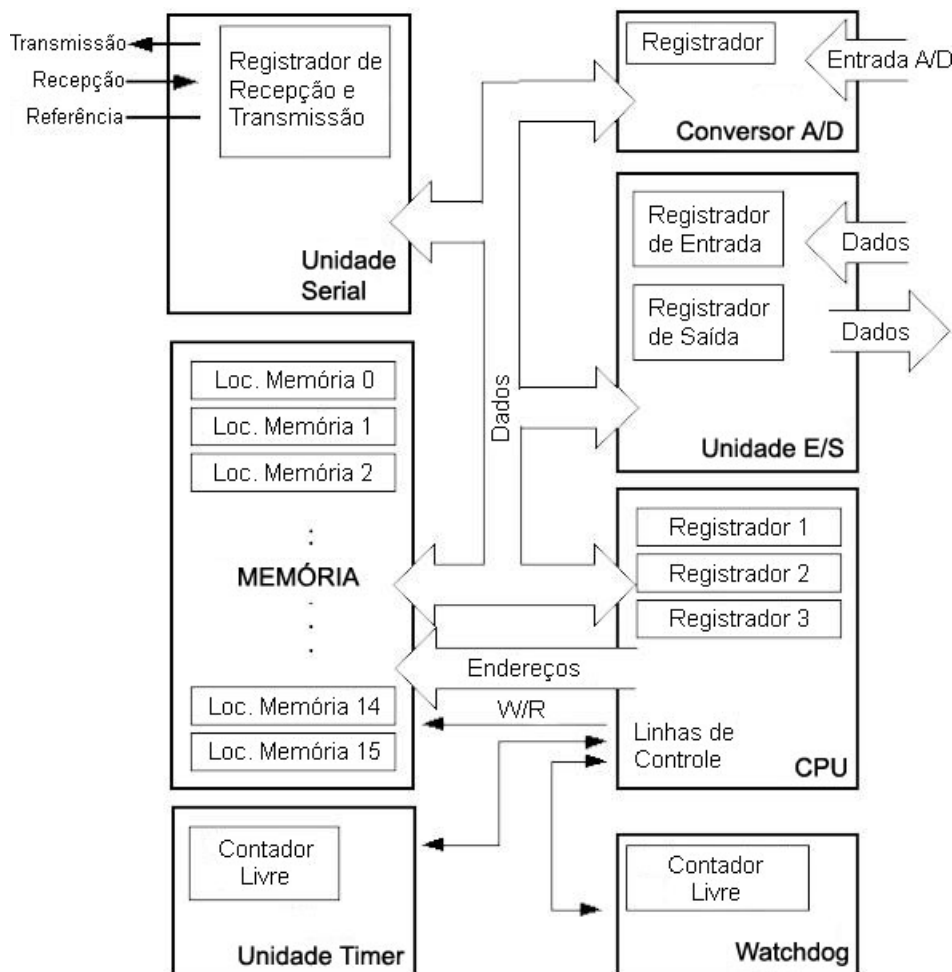


Figura 19. Elementos básicos e conexões internas de um microcontrolador da Microchip[8].

Escrever programas é um campo especial do trabalho com microcontroladores que é chamado de programação. A programação pode ser feita, dependendo do microcontrolador, em diversas linguagens como *Assembly*, *C* [24] ou *Basic* que são as linguagens mais comumente usadas. *Assembly* pertence as linguagens de baixo nível que são programadas com menor produtividade, mas conseguem o menor montante de espaço em memória e dão os melhores resultados, onde a velocidade do programa é uma preocupação. Programas na linguagem *C* são mais fáceis de escrever, mais fáceis de compreender, contudo são lentos em execução comparados aos programas em *Assembly*.

Basic é a mais fácil de aprender, e suas instruções estão mais perto do modo de pensar humano, mas como na linguagem de programação *C*, é mais lenta que *Assembly* devido ao código compilado ser maior e, conseqüentemente, menos eficiente. Em qualquer caso, antes de fazer um julgamento sobre essas linguagens é necessário considerar cuidadosamente a demanda de velocidade de execução, o tamanho da memória e a quantidade de tempo disponível para sua construção.

Depois de escrito o programa, deve-se instalá-lo dentro do dispositivo microcontrolador e executá-lo. Com intuito de fazer isso, é necessário adicionar alguns poucos componentes externos para essa tarefa. Primeiro, deve-se fornecer energia ao microcontrolador, como também ao oscilador que tem a função similar a função de um coração no corpo humano. Baseado nos ciclos de *clock*, o microcontrolador executa as instruções do programa.

Quando receber energia, o microcontrolador irá executar uma pequena auto-verificação, irá apontar para o início do programa e então iniciará sua execução. Como o trabalho do dispositivo depende de muitos parâmetros, o mais importante deles é a experiência do desenvolvedor de *hardware*, e a habilidade do programador em obter o máximo sobre esse dispositivo com seu programa.

Capítulo 3

Etapas do Projeto

A elaboração desse projeto foi planejada para um prazo de 12 meses de trabalho. Com intuito de facilitar o andamento do projeto e aumentar as chances de conclusão com sucesso, foi realizado um planejamento que resultou na divisão do projeto em diversas tarefas que foram agrupadas em etapas distintas. A seguir é apresentada uma descrição das etapas do projeto, bem como as responsabilidades e importância de cada uma delas.

3.1 Definição da arquitetura

A etapa que define a arquitetura do projeto é responsável por organizar a forma como os conceitos se relacionam para obtenção de um propósito, em particular, a captura do sinal de vídeo analógico.

Inicialmente, foi proposto que o microcontrolador realizasse a maioria das tarefas sozinho através de seus componentes embutidos, ou seja, seria utilizado o conversor analógico-digital embutido, o sinal digitalizado seria processado e repassado diretamente para uma saída de comunicação externa do microcontrolador, e os quadros de imagens seriam passados sequencialmente em tempo real para causar a sensação de movimento.

Entretanto, as análises sobre as arquiteturas dos microcontroladores disponíveis para o projeto mostraram que esses microcontroladores não tinham capacidade de processamento suficiente para realizar todas essas tarefas em tempo suficientemente hábil. Então, foram adicionados ao projeto os componentes externos: um conversor analógico-digital, um separador de sincronismo de sinal de vídeo, uma memória SRAM (*Static Random Access Memory*) e dois contadores para endereçamento da memória. Além disso, foi reduzido o escopo do projeto para capturar apenas um quadro de imagem do sinal de vídeo analógico e transmiti-lo em uma porta de comunicação serial quando a captura do quadro estivesse completa.

Essa nova abordagem limita a atuação da aplicação do projeto, porém, o circuito poderia ser projetado para receber componentes mais eficientes, mais rápidos, com maior capacidade de armazenamento, com mais opções de comunicação com o mundo externo que podem ser implementados em trabalhos futuros.

Em resumo, a função do circuito para este projeto foi delimitada apenas para captura de um quadro de imagem do sinal de vídeo, porém a forma como o projeto do circuito foi elaborado permite modificar a especificação dos componentes que formam o circuito para se obter

resultados melhores, como por exemplo, a transmissão de imagens em tempo real em trabalhos futuros.

3.2 Definição dos componentes

A etapa da escolha dos componentes é responsável por definir que circuitos integrados disponíveis no mercado poderiam ser utilizados neste projeto. Essas escolhas ocorreram para todos os componentes desde um simples transceptor até o tipo de microcontrolador. Esta etapa tem uma relação estreita com a definição da arquitetura, pois a partir da especificação dos componentes está definida a capacidade máxima de processamento do circuito.

Basicamente, o impacto dos componentes escolhidos tem influência no tempo disponível para se realizar uma tarefa, por exemplo, em apenas 52,2 μ s uma linha ativa em NTSC é transmitida, então é necessário que o conversor analógico-digital consiga realizar o máximo de amostras nesse intervalo de tempo se a qualidade da imagem é uma tarefa crítica. Um outro exemplo pode ser considerado a capacidade de armazenamento da memória externa que determina quantos *pixels* podem ser armazenados e conseqüentemente a resolução da imagem.

Para cada um dos componentes usados no projeto do circuito condicionador, foram considerados, para diversos fabricantes: tempos de funcionamento, condições recomendadas para utilização, formas de onda, tabelas de função, tabelas verdade, pinos e encapsulamento. Além disso, dois importantes fatores também foram utilizados como critério de seleção: o custo e a disponibilidade no mercado consumidor.

Os componentes definidos foram: transceptor (74HC245), contador (74HC4040), memória (HY62256), separador de sincronismo (LM1881), conversor A/D (LTC1406) e microcontrolador (18F4455). Esses componentes serão abordados no Capítulo 4.

3.3 Elaboração do esquema elétrico

A etapa de elaboração do esquema elétrico é responsável por transformar o diagrama de blocos do circuito condicionador em um diagrama capaz de estruturar a comunicação entre os componentes elétricos escolhidos. Nesta etapa, as abstrações sobre as funções dos componentes são detalhadas em função dos seus pinos.

Para elaborar o esquema elétrico é necessário conhecer as condições recomendadas sobre tensão e temperatura para utilização dos componentes. Elas são fornecidas pelos fabricantes em documentos denominados *datasheets*. Além disso, é exigido um conhecimento básico sobre eletrônica para adicionar ao esquema do circuito componentes elementares como resistores, capacitores, transistores, diodos etc.

3.4 Validação do esquema

Uma vez projetado o esquema elétrico do circuito condicionador, foi necessário realizar alguns testes para garantir o funcionamento conforme o desejado para os principais componentes utilizados no circuito. Esta etapa foi designada como validação do esquema e é importante porque a confecção de uma placa de circuito impresso pode ser dispendiosa, logo, gastos extras com retrabalho de circuito impresso devem ser evitados.

Esta etapa também serve para experimentar em laboratório a capacidade especificada pelos fabricantes e conseqüentemente construir protótipos básicos. Após os testes experimentais

alguns componentes podem ser substituídos por outros mais adequados, uma nova organização pode ser definida, valores de componentes elementares como resistores e capacitores são redefinidos etc.

O maior desafio desta etapa está na utilização de uma placa exclusiva para experimentos de circuitos eletrônicos bastante conhecida na área de eletrônica: uma placa *proto-board*. Na verdade, a utilização dessa placa plástica de prototipação é relativamente simples, pois as ligações dos componentes são realizadas através de pequenos fios introduzidos nos diversos furos dessa placa, para formar os barramentos. Contudo, a organização do circuito nesta placa está sujeita a contatos ruins e interferências provocadas pela própria natureza da placa *proto-board*, como capacitância parasita dos fios em paralelo.

Além disso, dependendo do encapsulamento dos componentes, o teste não pode ser realizado. Esses casos são mais comuns com encapsulamentos onde os pinos estão muito próximos um dos outros e o tamanho completo do componente é muito pequeno, como acontece para os encapsulamentos do tipo SSOP.

A maior parte dos circuitos integrados escolhidos por este trabalho possuem encapsulamento PDIP, com exceção apenas do conversor analógico-digital que é do tipo SSOP.

3.5 Elaboração do layout

A etapa de elaboração do *layout* consiste da definição física da posição dos componentes, ou seja, nesta etapa são definidos os locais físicos que cada componente irá ocupar na placa de circuito impresso. A definição do *layout* do circuito reduz ainda mais a abstração fornecida pelo esquema elétrico do circuito condicionador.

Alguns cuidados merecem atenção nesse momento:

- A distância entre as trilhas que ligam os componentes devem seguir os padrões recomendados pelos fabricantes dos componentes para evitar curto-circuitos e garantir capacitâncias associadas.
- A inspeção detalhada das trilhas deve ser realizada com atenção para evitar que conflitos causados pela interligação de duas ou mais trilhas ocorram.

Como o tamanho final do circuito impresso do projeto não é um requisito crítico, houve um certo conforto para se projetar esse *layout*. Mesmo assim, alguns projetos de *layout* requerem a utilização de mais de uma camada de placa. Existem placas com duas camadas que são conhecidas como placas de dupla face e também placas com várias camadas que são conhecidas como placas de múltiplas faces. Para o projeto do circuito condicionador, utilizou-se a técnica de dupla face com intuito de facilitar a interligação dos pinos e de evitar conflitos.

3.6 Confeção do circuito impresso

A etapa de confecção do circuito impresso consiste em aplicar técnicas industriais ou artesanais para imprimir o desenho contendo as ligações elétricas entre os componentes de um determinado circuito eletrônico sob uma chapa de material resistente recoberta por uma fina camada de cobre [29].

Basicamente, uma placa de circuito impresso é construída com um lado cobreado em cima de um substrato isolante (fenolite ou fibra de vidro). As conexões entre os componentes são feitas do lado do cobre através de caminhos condutores no cobre. As conexões terminam nos pontos de

conexão com os componentes, os quais são denominados de ilhas ou *pads* que normalmente possuem furos onde são inseridos os terminais dos componentes.

Uma grande vantagem da placa de circuito impresso é o fato de poder ser duplicada quantas vezes for necessário o que permite uma produção em larga escala.

3.7 Montagem da placa

Esta etapa é uma das mais simples entre todas as etapas do projeto, uma vez que já existem a especificação do projeto, esquema do circuito elétrico, *layout* do circuito elétrico e a placa de circuito impresso. A etapa de montagem da placa, como o nome já diz, é responsável pela inserção dos componentes eletrônicos na placa de circuito impresso.

Entenda-se por inserção de componentes, a tarefa de colocar adequadamente soldado o componente na placa de circuito impresso. Essas atividades requerem experiência com o manuseio de ferramentas para soldar componentes e prática para por a quantidade de solda e temperatura adequada. Alguns cuidados devem ser considerados: não fornecer calor acima do necessário no processo da solda para evitar eventuais danos tanto no próprio componente quanto nas trilhas condutoras de cobre, evitar tempo excessivo para soldar e ter cuidados com eletricidade estática.

Pode-se evitar soldar diretamente alguns circuitos integrados como os contadores, memória, microcontrolador, etc; acrescentando componentes auxiliares conhecidos como *sockets* para suportar componentes. Esses elementos auxiliares servem como uma base com entradas para a quantidade de pinos do componente em si, dessa forma, a tarefa consiste em soldar esse componente auxiliar e inserir usando um pouco de pressão manual o componente principal sobre os furos preparados para este fim.

3.8 Testes sobre a arquitetura completa

Finalmente, a última etapa do processo físico do projeto do circuito condicionador consiste em realizar testes logo após a montagem da placa de circuito impresso com seus respectivos componentes.

Esta etapa é responsável por verificar se ocorreu alguma falha de montagem e se todos os componentes elétricos estão ligados corretamente, uma vez que erros no processo de montagem podem ocorrer, tais como:

- Inserção de um componente diferente da especificação do projeto.
- Danos em um componente elétrico.
- Rompimento de um caminho condutor do circuito.
- Conflitos entre os pinos de um mesmo componente causados por um erro no processo de soldagem.

Antes de iniciar qualquer teste, deve-se verificar se o fornecimento de energia para o circuito atende ao especificado para o projeto, pois um erro nesta característica pode causar danos sérios aos componentes do circuito.

Recomenda-se a utilização de ferramentas adequadas para verificar se os pinos estão ligados corretamente, entre elas podemos citar: um multímetro e um osciloscópio. Entre os testes mais comuns estão:

- Verificação de continuidade dos caminhos condutores.
- Verificação do nível de energia fornecido em diversas partes do circuito.
- Verificação de conflitos entre os pinos de um mesmo componente.
- Análise da forma de onda do sinal gerado por alguns componentes.

Logo após a conclusão dos testes físicos, os testes que incluem o uso de *software* embarcado com o microcontrolador podem ser iniciados. Essa categoria de teste é relativamente mais simples, pois envolve a escrita de diversos trechos de código de programa para testar a comunicação de entrada/saída com o microcontrolador e os demais componentes do circuito. O resultado desses testes serve como base para implementação do *software* embarcado que será utilizado na aplicação a qual se destina o projeto do circuito condicionador.

Capítulo 4

Implementação

Este capítulo tem o objetivo de apresentar a implementação da metodologia para captura de vídeo usando microcontroladores. Em outras palavras, são apresentadas as decisões tomadas e as tarefas executadas em cada uma das etapas do projeto.

4.1 Arquitetura

A arquitetura do circuito condicionador é composta de vários componentes eletrônicos interligados fisicamente com intuito de tornar possível a captura, armazenamento e transmissão de uma imagem de vídeo. Esta seção descreve a forma como os componentes estão organizados para atender tal requisito.

A Figura 20 mostra a organização dos componentes do circuito condicionador proposto por este trabalho.

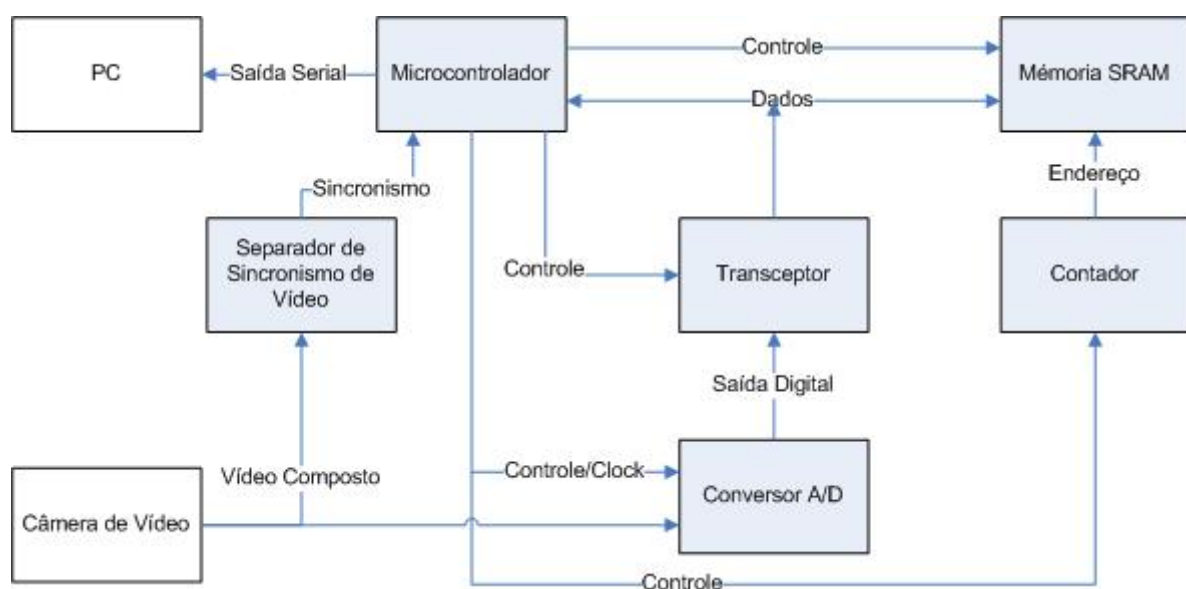


Figura 20. Diagrama de bloco do projeto.

Uma câmera analógica de vídeo monocromático é utilizada como fonte geradora do sinal de vídeo composto. Esse sinal de entrada é enviado para dois componentes do circuito: o conversor A/D e o separador de sincronismo de vídeo. O primeiro tem a capacidade de fazer a aquisição e a conversão para digital das amostras do sinal analógico de entrada, logo após essa conversão, um valor digital (dados da imagem) é disponibilizado na saída da porta paralela de 8-*bits* do conversor A/D. O segundo componente, o separador de sincronismo de vídeo, é projetado basicamente para detectar os sinais de sincronismo vertical e horizontal contidos no sinal de vídeo composto. Esse componente indica a ocorrência dos sinais de sincronismo definindo o valor de nível alto no seu pino de saída correspondente.

Um componente de memória SRAM externa é utilizado para armazenar os dados da imagem digital. A porta de entrada desse componente possui conexão tanto com uma porta de I/O do microcontrolador quanto com a porta de saída digital do conversor A/D, isso significa que os dados que a memória armazenará serão fornecidos ou pelo microcontrolador ou pelo conversor A/D. Quanto ao endereçamento da memória, este é realizado por um componente contador também inserido no circuito. Na verdade, foram utilizados dois componentes contadores de 12-*bits* em cascata que permitiria endereçar até 2^{24} posições.

O principal componente desta arquitetura é o microcontrolador, uma vez que este componente é responsável por controlar o funcionamento de todos os demais componentes do circuito, com exceção apenas do separador de sincronismo de vídeo. O controle das tarefas do microcontrolador é coordenado pela especificação do *software* embarcado. Mas, para que seja possível realizar tais controles devem existir conexões entre o microcontrolador e os componentes que se desejam comandar.

Sendo assim, a arquitetura proposta contempla conexões entre: os sinais de controle de escrita/leitura da memória e o microcontrolador, os sinais de controle de *clock/reset* dos contadores e o microcontrolador, o sinal de controle de habilitar do transceptor e o microcontrolador.

Conforme pode ser observado na Figura 20, o barramento de dados é compartilhado entre uma porta de I/O do microcontrolador, a porta de I/O da memória e a porta de saída do transceptor. A razão para existir um componente transceptor nesse circuito é permitir que a saída do conversor A/D seja isolada do restante do circuito, uma vez que o conversor A/D não tem esse sinal de controle. Então, quando a porta de saída do transceptor está desabilitada, apenas a memória e o microcontrolador usam o barramento de dados.

Para isolar a porta de I/O do microcontrolador usada no barramento de dados, basta configurá-la como uma porta de apenas entrada, através de *software*. De forma semelhante, para isolar a porta de I/O memória do barramento de dados, basta definir seus sinais de controle para desabilitar a porta de saída.

Esta arquitetura foi projetada para permitir operações de leitura e escrita conforme mostra a Tabela 4.

Tabela 4. Operações possíveis com o projeto do circuito.

Operação	Porta de origem	Porta de destino
Escrever	Conversor A/D	Memória
Escrever	Microcontrolador	Memória
Ler	Memória	Microcontrolador
Ler	Conversor A/D	Microcontrolador

Uma característica da estrutura do circuito que precisa ser observada com atenção refere-se a ligação do sinal de *clock* do conversor A/D com o microcontrolador. Como o conversor A/D realiza a operação de aquisição/conversão baseado no período do ciclo do sinal de *clock*, esta

ligação implica que o microcontrolador é responsável por gerar a frequência do sinal de controle que, conseqüentemente, determina a taxa de conversão das amostras do sinal de vídeo.

Estrategicamente o sinal de *clock* do conversor A/D e os sinais de *clock* dos contadores estão ligados ao mesmo pino de saída do microcontrolador, ou seja, o sinal enviado pelo microcontrolador para comandar a captura do sinal analógico, também comanda o incremento do contador de endereçamento da memória. A razão para essa estratégia é o fato do conversor A/D só precisar realizar amostras a medida que o endereçamento consegue ser computado, em outras palavras, não é necessário converter mais sinais do que se pode endereçar na memória.

Como foi citado anteriormente, o componente de sincronismo é o único componente que não é controlado pelo microcontrolador, esse componente necessita apenas da entrada do sinal de vídeo para desempenhar suas funcionalidades. Contudo, os sinais de sincronismo extraídos pelo separador são repassados para o microcontrolador através de conexões entre os pinos de saída do separador de sincronismo e alguns pinos de entrada do microcontrolador.

Os pinos de entrada que recebem os sinais de sincronismo, no microcontrolador, possuem uma característica especial: são pinos de interrupções. Sendo assim, quando forem detectados os sinais de sincronismo vertical e horizontal, o microcontrolador irá lançar interrupções correspondentes que podem notificar o *software* embarcado da ocorrência desses eventos.

Por fim, a arquitetura também provê a comunicação do microcontrolador com o mundo externo através de uma porta de saída serial. No projeto implementado para protótipo foi adicionada uma interface RS232 para este tipo de comunicação.

4.2 Componentes

A escolha dos componentes que constituem a arquitetura do projeto do circuito condicionador consumiu diversas horas de projeto, uma vez que a gama de modelos e fabricantes é bastante abrangente. Contudo, o foco principal do critério de seleção está na adequação dos recursos disponíveis no componente com a especificação requerida pelo projeto. Os componentes escolhidos estão listados nas seções abaixo, bem como o resumo sobre suas especificações. Uma descrição completa sobre os pinos dos componentes pode ser encontrada no Apêndice C.

4.2.1 Microcontrolador

O microcontrolador usado neste projeto é fabricado pela *Microchip* [10] e pertence a família 18F4455 [11]. Este componente é ideal para aplicações com baixo consumo porque possui tecnologia nW, e aplicações que precisam de conectividade porque possui três tipos de portas seriais: FS-USB (12Mbit/s), I2C e SPI (10Mbit/s), além de uma porta serial assíncrona EUSART.

A grande disponibilidade de memória RAM para armazenamento e a memória de programa do tipo *Enhanced Flash* tornam-no adequado para aplicações embarcadas de controle e monitoramento que requerem uma conexão periódica com um computador pessoal através da sua porta USB que pode ser usada tanto para enviar quanto para receber dados.

A Tabela 5 apresenta um resumo das características do microcontrolador 18F4455.

Tabela 5. Resumo das características do microcontrolador PIC18F4455.

Característica	Valor
Tipo da memória de programa	<i>Enhanced Flash</i>
Memória de programa	24 <i>Kbytes</i>
Memória de dados SRAM	2.048 <i>bytes</i>
Memória de dados EEPROM	256 <i>bytes</i>

Pinos de Entrada/Saída (I/O)	34
Encapsulamento	40 – PDIP
Temporizadores 8/16 bits	1/3
Interface USB 2.0 (12Mbit/s)	Sim
Portas seriais assíncrona	EUSART
Frequência de operação máxima	48MHz
Ciclo de instrução	4 Ciclos de <i>clock</i>

4.2.2 Conversor A/D

O conversor A/D é um circuito integrado capaz de adquirir sinais analógicos e transformá-los em sinais digitais. Nesta aplicação, em particular, o conversor utilizado é um LTC1406CGN [13], fabricado pela *Linear Technology* [12], que possui capacidade máxima para realizar 20Mps (20 milhões de amostras por segundo) de 8 bits sobre uma fonte geradora de sinal analógico com até 250MHz de largura de banda.

Esse conversor A/D possui uma saída paralela de 8-bits e a arquitetura de *pipeline* tem uma latência de dados de cinco ciclos de *clock*.

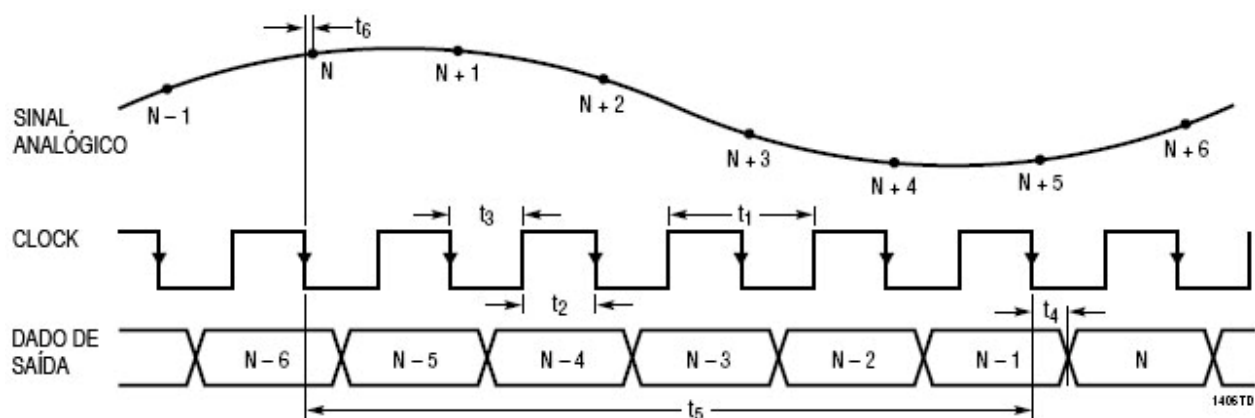


Figura 21. Pipeline do conversor A/D LTC1406 [13].

A conversão dos dados inicia na borda de descida do sinal de *clock* e os dados estão disponíveis quando o sinal de *clock* está alto, considerando um *pipeline* com latência de dados de cinco ciclos de *clock*.

A Figura 21 ilustra a arquitetura de *pipeline*, onde t_1 , t_2 , t_3 , t_4 , t_5 e t_6 representam período de *clock* (mínimo 50ns), largura de pulso alto (mínimo 20ns), largura de pulso baixo (mínimo 20ns), atraso de saída (máximo de 25ns), atraso de *pipeline* (5 ciclos), atraso de abertura (normalmente 3ns) respectivamente.

4.2.3 Separador de Sincronismo

O componente separador de sincronismo usado foi LM1881 [15], fabricado pela *National Semiconductor* [14], que é um circuito integrado projetado para tirar os sinais de sincronização a partir de uma fonte geradora de sinal de vídeo composto que estão nos formatos NTSC, PAL e SECAM. Os sinais extraídos incluem sinal de sincronismo vertical, sinal de sincronismo composto, sinal de *burst* de entrada e sinal de campo de informação de linha ímpar ou par.

O diagrama de tempo abaixo ilustra as informações sobre as informações de tempo deste componente.

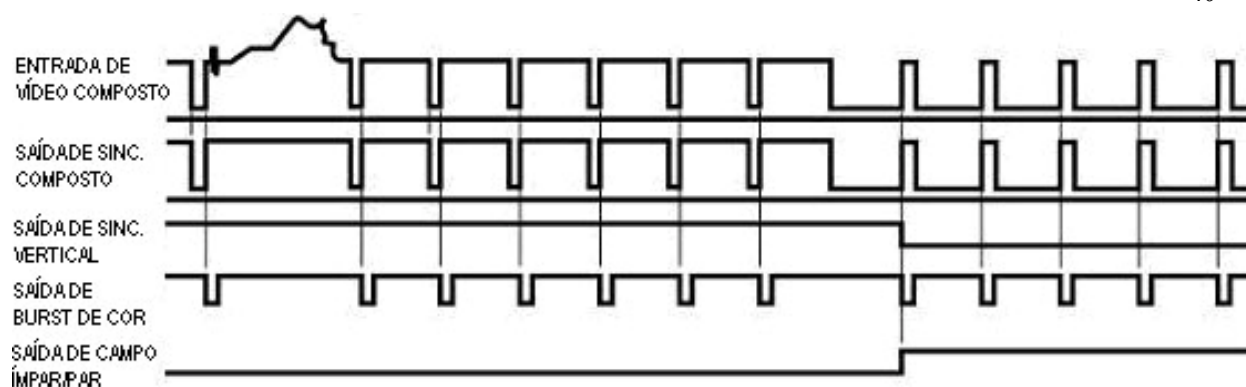


Figura 22. Diagrama de tempo do componente LM1881 [15].

Conforme pode ser observado na Figura 22, a saída do sinal de sincronismo horizontal (composto) é simplesmente a reprodução da forma de onda do sinal de vídeo composto em seu nível de preto, com o vídeo completamente removido. A saída do sinal de sincronismo vertical é uma derivação do sinal de sincronismo composto.

As saídas dos sinais: *burst* de entrada e campo de informação ímpar e par não são abordados neste trabalho. Esses sinais geralmente não são usados, entretanto, o primeiro pode ser útil para aplicações que necessitam de informação sobre o entrelaçamento da fonte de vídeo composto. E o segundo, a saída do sinal de *burst*, é útil em aplicações que precisam de informação sobre as cores da imagem.

4.2.4 Memória SRAM

O componente de memória usado na implementação deste trabalho é especificado como HY62256-A [17] que é fabricada com tecnologia de alto desempenho pela *Hyundai* [16]. Esta memória tem velocidade de operação razoável, baixo consumo de energia e possui 32.768x8-bits posições de memória de acesso aleatórios do tipo SRAM para armazenamento.

A memória HY62256 tem um modo de retenção de dados que garante a recuperação de dados válidos com uma tensão mínima de alimentação de apenas 2V. Logo, este componente é aplicável em sistemas que usam baixa tensão ou sistemas com bateria de reserva (*backup*). O encapsulamento escolhido foi 28-pinos PDIP.

Os sinais de controle da memória são denominados /CS, /WE, /OE que significam seleção de componente, permitir escrita e permitir saída respectivamente. A Tabela 6 apresenta a lista de estados que a memória pode assumir. A legenda L, H e X é usada para representar o nível lógico do sinal em alto, baixo ou indiferente.

Tabela 6. Tabela verdade da memória.

/CS	/WE	/OE	Modo	Operação E/S
H	X	X	Repouso	Alta impedância
L	H	H	Saída desabilitada	Alta impedância
L	H	L	Leitura	Saída de dados
L	L	X	Escrita	Entrada de dados

Como pode ser observado na Tabela 6, o funcionamento da memória é simples. O ciclo de leitura ocorre quando os sinais de controle assumem esta configuração /CS=L, /WE=H e /OE=L e o ciclo de gravação ocorre com a configuração /CS=L e /WE=L, note que para a escrita o valor do sinal /OE é indiferente.

4.2.5 Contador

O componente contador usado na implementação deste trabalho é chamado 74HC4040 [19] que é fabricado pela *Texas Instruments* [18]. Este componente é um contador binário de 12 estágios com uma entrada de */clock*, uma entrada de *reset* para limpar a contagem de forma assíncrona e doze saídas paralelas. O encapsulamento deste componente é 16-DIP.

A contagem avança quando a transição do nível alto para o nível baixo do sinal de */clock* acontece, e a saída paralela assume o valor zero quando o sinal de controle *reset* para o nível alto. A Tabela 7 apresenta um resumo das funções deste componente contador, as legendas L, H, X, ↑ e ↓ são usadas para representar os estados de nível alto, baixo, indiferente, transição do nível baixo para alto e transição do pulso de nível alto para o pulso de nível baixo.

Tabela 7. Tabela de função do contador.

Clock	Reset	Estado da Saída
↑	L	Nenhuma mudança
↓	L	Avança a contagem
X	H	Nível baixo (contador zerado)

A seguir é mostrado o diagrama de tempo do componente contador, basicamente a Figura 23 ilustra o nível de saída em cada uma dos doze sinais de saída paralela, nomeados de Q1 a Q12 de acordo com o ciclo do sinal de *clock*. Além disso, na Figura 23 é apresentado o sinal de *reset* em seu estado alto, apenas no início do ciclo de *clock*.

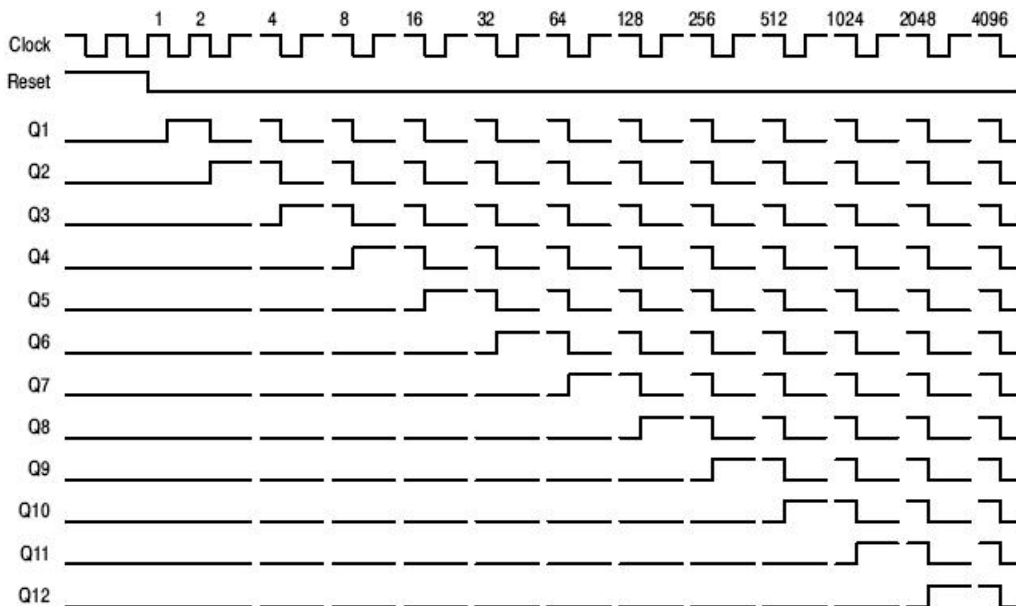


Figura 23. Diagrama de tempo do contador.

4.2.6 Transceptor

O componente transceptor usado na implementação deste trabalho é chamado 74HC245 [20] que é fabricado pela *Philips Semiconductors* [21]. Este componente é um transceptor de barramento de 8-bits compatível com três estados de saída em ambas as direções: envio e recebimento. Os sinais de controle são */OE* e *DIR* que são responsáveis pelo controle de permissão de saída e pelo

controle das direções de envio e recebimento respectivamente. O encapsulamento deste componente é 20 pinos-DIP.

O transceptor 74HC245 possui também duas portas de dados de entrada e saída, denominadas de A e B. A idéia principal deste componente é transferir para a porta de saída os sinais que são recebidos por em uma porta de entrada. A direção de transferência pode ocorrer nos dois sentidos: da porta A para a porta B ou vice-versa. Além disso, esse componente pode assumir um estado de alta impedância na saída das suas portas, esse estado é utilizado para isolar o componente do resto do circuito.

A Tabela 8 resume as funções deste transceptor. As legendas L, H, Z e X são usadas para representar os níveis lógicos alto, baixo, alta impedância e indiferente.

Tabela 8. Tabela de função do transceptor.

Entrada		Entrada/Saída	
/OE	DIR	An	Bn
L	L	A = B	Entrada
L	H	Entrada	B = A
H	X	Z	Z

Conforme mostra a Tabela 8, o sinal /OE=H define as duas portas A e B para um nível de tensão de alta impedância, isolando-as do resto do circuito elétrico. Quando o sinal /OE está definido igual a /OE=L, a função de transferência ocorre da porta A para B quando o sinal DIR é igual a DIR=H, caso contrário, DIR=L, a transferência ocorre no sentido contrário.

É importante lembrar que a função de transferência define as portas A e B para um dos três estados: entrada, saída ou alta impedância.

4.3 Esquema elétrico

A implementação do esquema elétrico da arquitetura para o circuito condicionador definiu a ligação elétrica entre os diversos sinais utilizados para comunicação entre os componentes. A tarefa principal realizada no esquema elétrico foi identificar em cada um dos componentes seus pinos de entrada e saída e relacioná-los conforme a proposta da arquitetura.

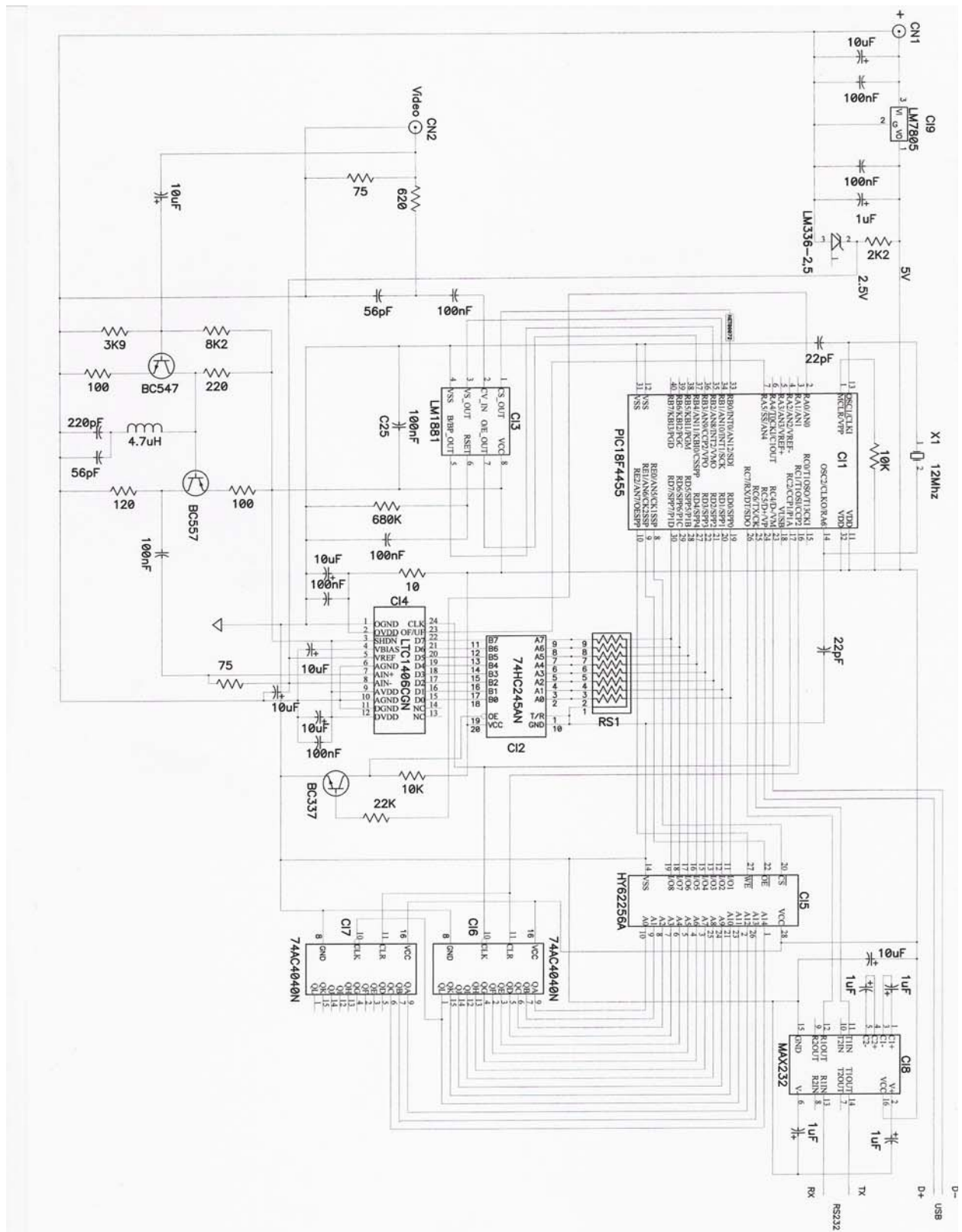


Figura 24. Esquema elétrico do circuito condicionador.

A Figura 24 apresenta o esquema elétrico do circuito condicionador proposto pela arquitetura de projeto deste trabalho. A listagem de componentes está disponível no Apêndice A deste trabalho. Para a elaboração desse esquema elétrico, informações sobre as condições,

recomendadas pelos fabricantes, sobre tensão de alimentação e temperatura de cada um dos componentes são importantes para garantir o funcionamento adequado dos componentes. Foi usado a ferramenta P-CAD 2002 Schematic para construção desse esquema elétrico.

Como pode ser observado na Figura 24, componentes elétricos comuns, como resistores, capacitores, diodos e transistores foram inseridos no esquema para garantir condições adequadas de diferença de potencial, corrente, capacitância, impedância, polarização relativas a natureza do circuito elétrico. Dessa forma, esta implementação foi realizada com a contribuição de um profissional qualificado em eletrônica, tendo em vista o pouco conhecimento do assunto em questão.

Na parte central da Figura 24 do esquema, pode-se facilmente identificar o barramento de dados entre o microcontrolador, a memória e o transceptor. Também é possível encontrar os pinos dos sinais de controle que o microcontrolador envia para os demais componentes. Quanto ao componente da memória é possível identificar os pinos de endereçamento ligados a saída dos dois contadores do circuito. A Tabela 9 resume a relação de origem e destino entre os pinos dos componentes principais do esquema, como também a função dos sinais na ligação.

Tabela 9. Relação de pinos entre os componentes do circuito.

Origem		Destino		Função do(s) Sinal(is)
Componente	Pinos	Componente	Pinos	
Microcontrolador	19, 20, 21, 22, 27, 28, 29, 30	Memória	11, 12, 13, 15, 16, 17, 18, 19	Dados
Microcontrolador	8, 9, 10	Memória	20, 22, 27	Controle
Microcontrolador	16, 17	Contador 1	11, 10	Controle
Microcontrolador	16	Contador 2	11	Controle
Microcontrolador	2	Transceptor	19	Controle
Microcontrolador	17	Conversor A/D	24	Controle
Separador Sinc.	1, 3, 5, 7	Microcontrolador	33, 34, 35, 37	Interrupção
Contador 1	9, 7, 6, 5, 3, 2, 4, 13, 12, 14, 15, 1	Memória	10, 9, 8, 7, 6, 5, 4, 3, 25, 24, 21, 23	Endereçamento
Contador 2	9, 7, 6	Memória	2, 26, 1	Endereçamento
Contador 1	1	Contador 2	10	Controle
Conversor A/D	22, 21, 20, 19, 18, 17, 16, 15	Transceptor	11, 12, 13, 14, 15, 16, 17, 18	Dados
Transceptor	2, 3, 4, 5, 6, 7, 8, 9	Microcontrolador	19, 20, 21, 22, 27, 28, 29, 30	Dados
Transceptor	2, 3, 4, 5, 6, 7, 8, 9	Memória	11, 12, 13, 15, 16, 17, 18, 19	Dados

Devido a natureza de capacidade da memória usada, foram usados apenas alguns pinos de saída dos contadores para realizar o endereçamento. Também é possível visualizar o cascadeamento entre os dois contadores para formação de endereços maiores. Basicamente, o *bit* mais significativo do primeiro contador é ligado ao sinal de controle de *clock* do segundo contador, dessa forma, a cada estouro de contagem do primeiro contador um incremento ocorre no segundo.

Esse projeto de esquema elétrico contempla também alguns circuitos complementares para regular a entrada de alimentação para 5V, regular o nível de referência 2,5V exigido pelo conversor A/D, transmitir dados na saída serial do microcontrolador através de uma interface

RS232 e amplificar o sinal de vídeo na entrada. Os detalhes sobre esses circuitos complementares não são tratados neste trabalho e estão além do escopo de assunto estudado.

4.4 Validação do esquema elétrico

A implementação da validação do esquema elétrico consistiu em montar alguns componentes do circuito em um arranjo semelhante ao definido no esquema sobre uma placa de prototipação elétrica, *proto-board*. Apenas os componentes do microcontrolador, separador de sincronismo, contador e memória foram testados na placa de prototipação.

A validação do microcontrolador foi realizada através da implementação de um pequeno programa de teste com objetivo de enviar uma mensagem de texto através da porta serial RS232. Para se certificar que a mensagem tinha sido enviada com sucesso, foi utilizado o programa *HyperTerminal* do sistema operacional *Windows* para receber esta mensagem. Essa validação foi útil para definir as configurações de portas de I/O, temporizadores, interrupções e a taxa de transmissão da porta serial EUSART.

A validação do contador foi relativamente simples, uma vez que o microcontrolador foi usado para gerar os sinais de *clock* e *reset*. O valor da saída do contador foi verificada medindo-se o nível de tensão que estava definido em cada um dos pinos de saída com o uso de uma ferramenta de multímetro.

O teste do componente da memória foi realizado parcialmente, pois foram testadas apenas as operações de escrita e leitura no endereço zero. Além disso, a definição de um valor constante foi forçado eletronicamente na entrada de dados da memória. O valor resgatado da memória também foi verificado com o uso de um multímetro.

Um dos testes mais simples, entre os testes realizados, aconteceu com o componente de separador de sincronismo do sinal de vídeo. Este componente foi adicionado a placa de *proto-board*, o sinal de vídeo foi fornecido como entrada e as saídas notificando a detecção dos sincronismos foram verificadas com o uso de uma ferramenta de osciloscópio.

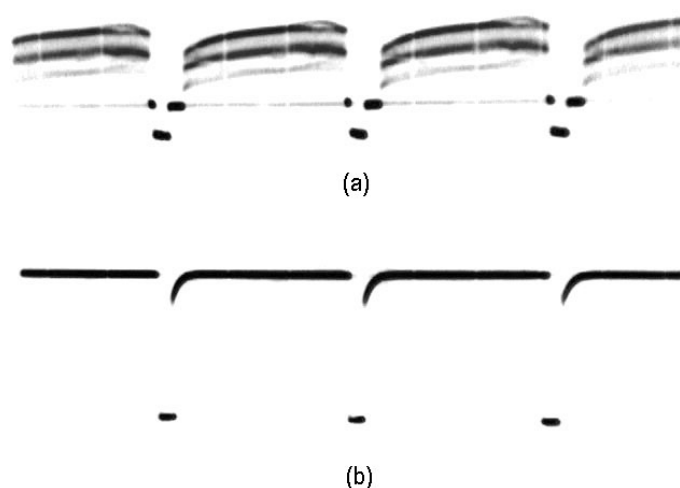


Figura 25. Sinal de vídeo composto (a) e sinal de sincronismo horizontal (b), ambos extraído com uma ferramenta de osciloscópio.

A Figura 25 mostra a validação do sinal de sincronismo horizontal extraído a partir do componente separador de sincronismo de vídeo. Na parte superior da Figura 25 foi utilizado um canal do osciloscópio para mostrar o sinal de vídeo original e na parte inferior da Figura 25 foi

utilizado o segundo canal do osciloscópio para mostrar o sinal extraído do pino de saída de sincronismo horizontal. Uma lista completa de figuras com os sinais do separador de sincronismo está disponibilizada no Apêndice B.

A razão para não realizar testes com o componente do conversor A/D refere-se ao tipo de encapsulamento que este componente possui, encapsulamento SSOP. Esse tipo de encapsulamento não permite usar o componente em placas de prototipação *proto-board*, devido as suas dimensões minúsculas. Já os testes com o transceptor não foi necessário pois o seu funcionamento é bastante simples, e a sua função no circuito não ocupa uma posição crítica. Portanto, optou-se por não usar o tempo de projeto com atividades de resultados poucos significativos.

4.5 Layout do circuito

A implementação do *layout* do circuito elétrico consiste em projetar a organização física dos componentes que compõe o circuito, ou seja, são definidos os locais onde cada componente ficará na placa de circuito impresso. Essa parte da implementação precisou da contribuição de um profissional especializado em projetos de *layout* PCB (*Printed Circuit Board*). A ferramenta utilizada para construir o desenho do *layout* foi o *software* de projeto assistido por computador P-CAD 2002 PCB.

Um conjunto de informações foram reunidas para possibilitar a criação do projeto de *layout* PCB e foram entregues ao profissional especializado. As informações são as seguintes:

- Esquema elétrico do circuito condicionador.
- Lista com a referência dos componentes.
- Encapsulamento e dimensões dos componentes.
- Restrições de espaço ocupado pelo circuito.
- Recomendações específicas do fabricante sobre o componente.

Sendo assim, o esquema elétrico e a lista de componentes com suas dimensões e encapsulamentos foram fornecidas para realização desta tarefa. Como a natureza deste projeto não exige restrições tamanho da placa de circuito impresso, não foi necessário solicitar restrições de espaço. Quanto as recomendações específicas dos fabricantes de cada componente foram repassados os documentos conhecidos como *datasheets*.

Uma restrição importante que deve ser citada nesta implementação refere-se a divisão dos planos de aterramento em digital e analógico recomendado pela *Linear Technology* para seu conversor A/D LTC1406 que é usado neste circuito condicionador. Maiores informações sobre essas restrições estão disponíveis no manual deste componente.

O *layout* projetado empregou duas faces de trilhas condutoras para placa de circuito impresso. Essa estratégia facilita o roteamento das trilhas condutoras entre os componentes. É importante lembrar que o arquivo de resultado dessa implementação, arquivo .PCB, é fundamental para a confecção do circuito impresso, como será descrito a seção seguinte.

4.6 Circuito impresso

A função básica desta etapa da implementação foi proporcionar suporte mecânico e interligação elétrica para os componentes utilizados no circuito eletrônico. Apesar de uma empresa local ter sido contratada para confeccionar esta placa, os processos utilizados por ela foram levantados e estão descritos nesta seção.

A confecção do circuito impresso foi realizada com técnicas artesanais de impressão para traçados condutores entre componentes, também conhecidos como trilhas, de circuito elétrico. Foi utilizada uma placa constituída de um laminado de fibra de vidro com ambas as faces recobertas por uma fina camada de cobre.

Esse material possui boa rigidez e ótima isolamento elétrica, dessa forma ele também pode ser utilizado em circuitos impressos profissionais, conseguindo-se produzir circuitos com alta densidade de trilhas, devido as suas características.

Uma vez que o processo utilizado foi artesanal, o circuito impresso foi construído com o uso de várias máscaras produzidas através de um processo serigráfico. As matrizes em material fotolítico das máscaras serigráficas foram impressas a partir do arquivo de *layout* .PCB. As impressões necessárias para confecção das máscaras serigráficas estão descritas a seguir.

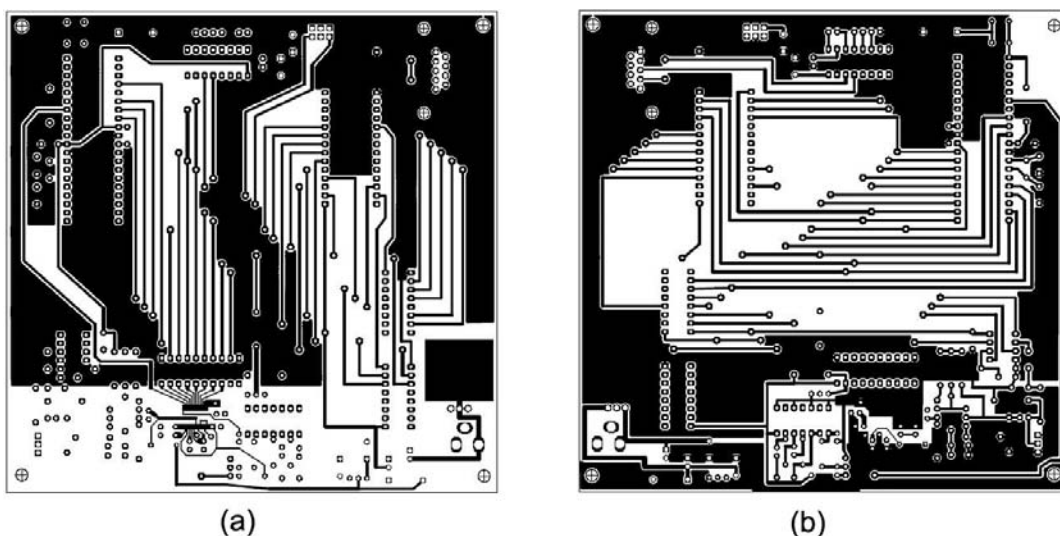


Figura 26. (a) Máscara de trilhas da face superior e (b) máscara de trilhas da face inferior do circuito.

A Figura 26 apresenta as matrizes para confecção das máscaras de trilhas das faces superior e inferior. Essas matrizes foram extraídas do arquivo de *layout* PCB através da ferramenta P-CAD 2002 e impressas em um fotolito. As máscaras serigráficas resultantes dessas matrizes servem para aplicar uma fina camada de estanho-chumbo sobre a camada de cobre nos locais onde as trilhas condutoras passam. Em seguida, através de corrosão química, retira-se o cobre em excesso.

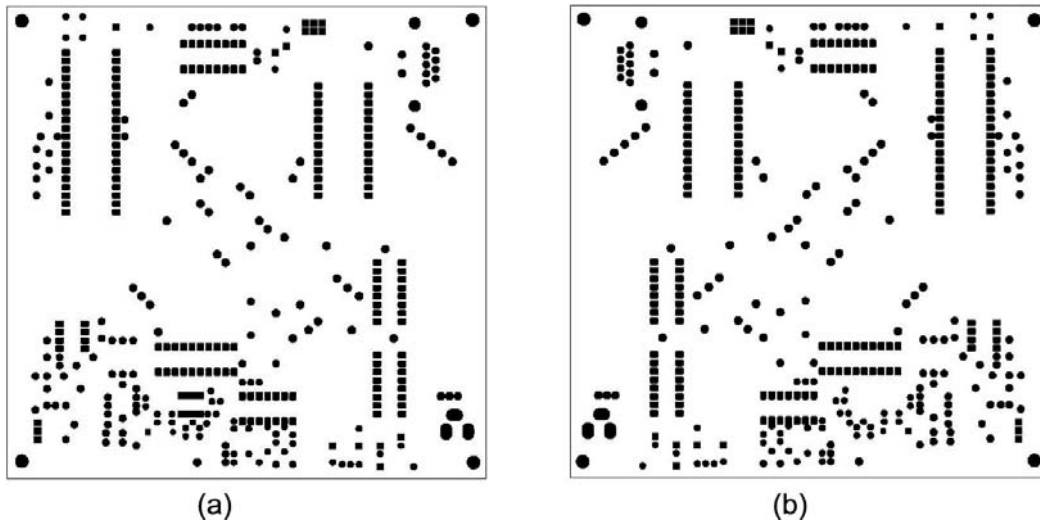


Figura 27. (a) Máscara de ilhas de soldagem da face superior e (b) máscara de ilhas de soldagem da face inferior do circuito.

O processo seguinte consiste em criar duas máscaras serigráficas (superior e inferior) para proteger os contatos elétricos, também conhecidos como ilhas de soldagem, da aplicação de um verniz que recobre toda a superfície da placa. Esse verniz possui uma característica que é utilizada para proteger a placa de oxidação, normalmente possui uma cor esverdeada [30]. As matrizes impressas em fotolito dessas máscaras também são extraídas a partir do arquivo de *layout* PCB, as quais geralmente são chamadas de máscaras de ilhas conforme mostra a Figura 27.

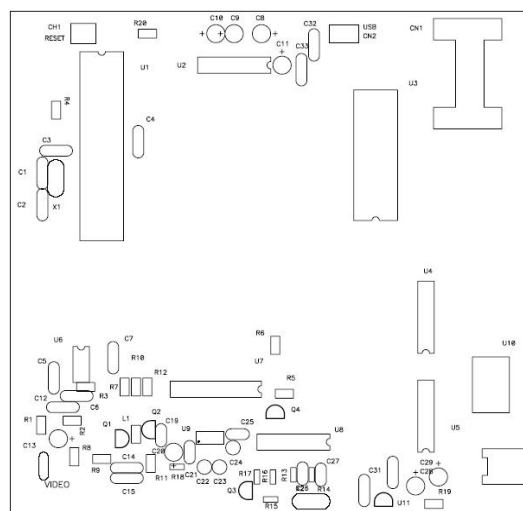


Figura 28. Máscara de contornos de componentes do circuito.

A última matriz para construção de máscara serigráfica, conforme mostra a Figura 28, refere-se a impressão dos contornos dos componentes e suas respectivas referências, na superfície da placa, visando facilitar a sua montagem e manutenções futuras. Este desenho geralmente é chamado de serigrafia de componentes. Como nas matrizes anteriores, essa matriz também foi extraída do arquivo de *layout* PCB e impressa em um fotolito para confecção da máscara serigráfica.

A última fase do processo artesanal utilizado para confeccionar a placa consiste em furar a placa nos locais onde existem ilhas de soldagem. Esses furos permitem a inserção dos terminais

dos componentes para a soldagem. Existem vários processos de furação, entretanto, nesta implementação foi utilizada uma furadeira manual de bancada para realizar esta tarefa.

É importante lembrar que devido a natureza de dupla face da placa, alguns furos chamados furos de passagem são necessários para interligar as trilhas condutoras entre uma face e outra da placa de acordo com o especificado no projeto do *layout* PCB.

4.7 Montagem da placa

A implementação da montagem da placa, como o nome já diz, é responsável pela inserção dos componentes eletrônicos na placa de circuito impresso devidamente soldados.

Os materiais usados nessa implementação além dos próprios componentes foram: a solda e o ferro de soldar. As tarefas de soldagem exigiram uma pequena experiência com o manuseio desses materiais. Porém, alguns cuidados foram tomados para evitar danos tanto nas trilhas condutoras da placa de circuito impresso quanto aos componentes eletrônicos. Esses cuidados são: não fornecer quantidade excessiva de solda, não utilizar o ferro de soldar em temperatura muito alta, evitar tempo prolongado para soldar e cuidados sobre energia estática.

Para evitar o contato do ferro de soldar diretamente com os terminais condutores dos componentes do microcontrolador, memória, contadores e transceptor, foram inseridos componentes auxiliares conhecidos como *sockets* que são usados como suporte para receber os terminais dos componentes. Sendo assim, o que realmente é soldado na placa são os *sockets* de suporte ao invés dos próprios componentes, em seguida, basta inserir o componente no *socket* já soldado.

Como a placa foi confeccionada a partir de um processo artesanal, é importante lembrar que todos os furos criados na placa de circuito impresso não possuem metalização. Isso significa que os furos de passagem, responsáveis por ligar uma face a outra da placa, precisam receber uma solda com uma pequena haste de metal para ligar ambos os lados.

A Figura 29 ilustra a diferença entre um furo metalizado e um furo não metalizado, este último, como ocorre em processos artesanais.

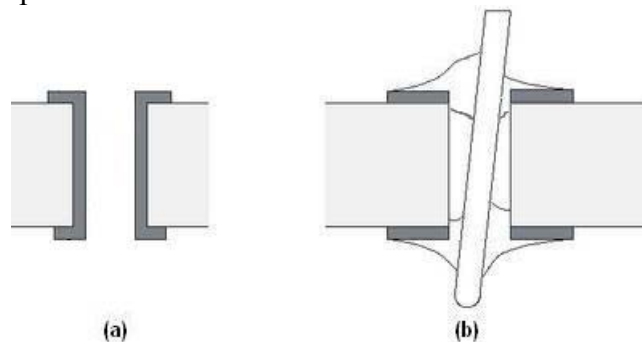


Figura 29. (a) Visão de um furo metalizado e (b) visão de um furo não metalizado.

Pode-se observar na figura que os furos metalizados ligam uma face a outra devido a existência de um anel condutor dentro do furo. Por outro lado, nos furos não metalizados a condução elétrica ocorre em lados distintos, logo, para ligá-los é necessário soldar uma haste condutora dentro do furo.

Uma tarefa que também mereceu destaque na implementação da montagem da placa foi a tarefa de soldar o componente do conversor A/D. Este componente possui encapsulamento SSOP que é destinado ao manuseio por máquinas automatizadas. Sendo assim, soldar manualmente um componente com a largura de um pino medindo pouco mais de 0,203 milímetros e tendo uma

distância entre centros de pinos medindo 0,635 milímetros foi uma tarefa muito difícil e cautelosa.

O custo para construção de um protótipo deste projeto foi de aproximadamente R\$ 120,00. Nesse valor estão incluídos: o custo dos componentes, o custo de confecção das máscaras e o custo de impressão da placa do circuito. É importante destacar que o custo do protótipo de projeto, geralmente, é muito mais elevado do que o custo de uma produção industrial em grande escala para mesmo projeto.

4.8 Testes da arquitetura

A implementação dos testes da arquitetura consistiu em verificar se o circuito condicionador montado estava pronto para receber o *software* embarcado, ou seja, o objetivo foi se certificar de que não havia falhas ou irregularidades elétricas após a montagem do circuito. Os testes e análises realizados estão resumidos na Tabela 10, como também as ferramentas utilizadas:

Tabela 10. Lista de testes sobre a parte elétrica do circuito.

Descrição do teste	Ferramenta utilizada
Teste da continuidade das trilhas condutoras entre os componentes	Multímetro
Teste de conflitos entre os pinos de um mesmo componente	Multímetro
Teste se existe alimentação chegando nos componentes	Multímetro
Análise da qualidade do sinal de vídeo na entrada do circuito	Osciloscópio
Análise da forma de onda do sinal de sincronismo vertical	Osciloscópio
Análise da forma de onda do sinal de sincronismo horizontal	Osciloscópio
Análise da forma de onda do oscilador de frequência do circuito	Osciloscópio

É importante lembrar que antes de iniciar qualquer teste com o circuito condicionador, certificou-se se a tensão de alimentação estava apropriada para o projeto do circuito (5V para o circuito e 2,5V para referência do conversor A/D). Este teste foi realizado removendo dos *sockets* os componentes principais como microcontrolador, memória, contador, e transceptor para evitar dados elétricos nesses componentes.

Concluído os testes e as análises da parte elétrica do circuito condicionador, foram realizados testes sobre a funcionalidade dos componentes através de pequenos programas, semelhantes aos aplicados na implementação da validação do esquema elétrico.

A Tabela 11 apresenta um resumo dos testes realizados sobre as funcionalidades dos componentes.

Tabela 11. Lista de testes sobre as funcionalidades dos componentes.

Descrição do teste	Componentes
Programa teste para zerar e incrementar os contadores	Contador
Programa teste para detectar as interrupções causadas pelos sinais de sincronismo vertical e sincronismo horizontal	Separador de sincronismo
Programa teste para ler/escrever na memória	Memória
Programa teste para converter uma amostra do sinal de vídeo e escrever o resultado na memória	Conversor A/D e Memória

Todos os testes implementados em *software* foram utilizá-los como base fundamental para o desenvolvimento do *software* embarcado para captura de imagens. Devido a importância deste *software* embarcado, o capítulo seguinte foi dedicado exclusivamente a este assunto.

Capítulo 5

Software para Captura de Imagens

O *software* embarcado desenvolvido para controlar o circuito condicionador tratado em capítulos anteriores é o principal assunto abordado neste capítulo. A linguagem de programação adotada é uma adaptação da linguagem C-ANSI adaptada pela empresa *Hi-Tech* [26] para seu compilador PIC-C18 [25] cuja plataforma de destino são os microcontroladores da família 18, fabricados pela *Microchip*.

A elaboração de sistemas embarcados utilizando a linguagem C tem a vantagem de reduzir o tempo de projeto, pois suas características sintáticas e semânticas facilitam o desenvolvimento e os testes do programa. Por outro lado, o tamanho do programa compilado ocupa um espaço maior na memória de programa do microcontrolador se comparado ao tamanho de um programa obtido através de uma linguagem montadora como *Assembly*.

Considerando que a qualidade de resolução da imagem obtida através do circuito condicionador está além do objetivo deste projeto, o *software* foi desenvolvido para obter apenas um quadro da imagem de vídeo com resolução de 40x30 *pixels* para validar a metodologia empregada na captura de imagens usando microcontroladores.

5.1 Máquina de Estados

A Figura 30 apresenta a máquina de estados do programa. Como pode ser observado na Figura 30, a idéia principal do programa é detectar as interrupções devido a presença dos sinais de sincronismo para realizar as operações referentes a captura do quadro de imagem de vídeo.

O primeiro estado é responsável por definir as configurações iniciais, tais como: definir o sentido das portas de entrada/saída, habilitar/desabilitar interrupções, habilitar/desabilitar temporizadores, ajustar a taxa de transmissão serial, entre outras.

O segundo estado consiste em aguardar pelo sinal de sincronismo vertical que será capturado por uma interrupção, esse estado é responsável por definir o momento de início de um quadro de imagem. Ainda nesse estado, uma variável é ajustada para definir o número máximo de linhas que serão capturadas.

O terceiro estado simplesmente aguarda pelo sinal de sincronismo horizontal que também será capturado por uma interrupção. Esse estado é utilizado para identificar o início de uma nova linha do quadro de imagem. Semelhante ao estado anterior, uma variável é ajustada para definir o número máximo de colunas (*pixels*) por linha que serão capturadas.

O quarto estado é responsável por comandar as operações que realizam a captura e armazenamento de todos os *pixels* de uma linha. Um dos fatores que contribuem para a qualidade da resolução da imagem é a quantidade de amostras realizadas nesse momento, por isso, o conjunto de instruções que realizam esta operação deve ser cuidadosamente analisado para que se obtenha o menor tempo possível por *pixel* capturado e armazenado.

Ao término da captura de uma linha completa, verifica-se se existem linhas restantes para serem capturadas, caso afirmativo a máquina de estados assume seu estado de espera de próximo sinal de sincronismo horizontal, caso contrário, a máquina de estados assume seu quinto e último estado.

Uma vez que o quadro da imagem de vídeo está armazenado na memória, o quinto estado é responsável apenas pelo resgate e transmissão dos dados da memória através de uma saída serial do microcontrolador os quais serão recebidos por um outro dispositivo capaz de processá-los, em particular, um computador pessoal comum foi utilizado neste projeto.

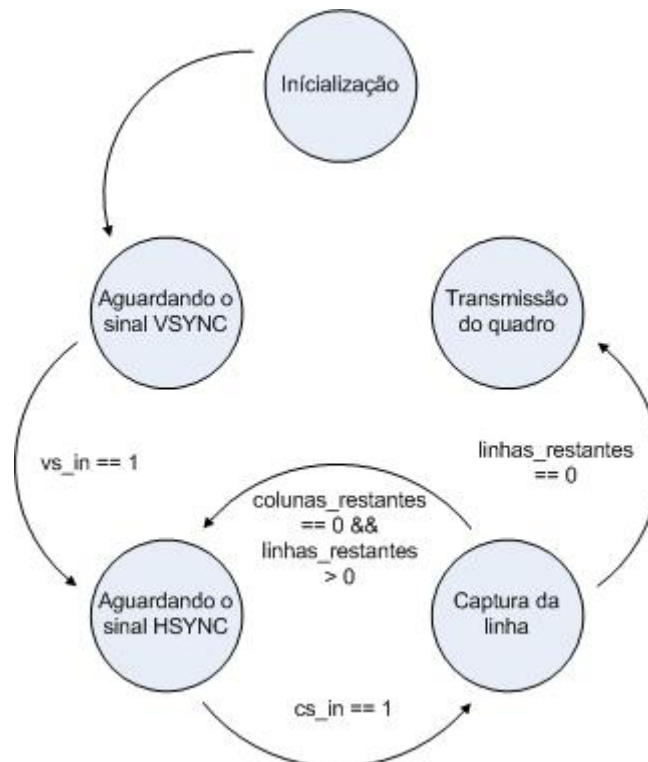


Figura 30. Máquina de estados do programa para captura de um quadro de imagem de vídeo.

Devido à sua importância, o processo de captura e armazenamento realizado pelo quinto estado está apresentado com mais detalhes na Figura 31. A captura e armazenamento consistem em enviar sinais para o conversor A/D adquirir e converter o sinal de vídeo, sinais para o contador de endereço assumir seu próximo valor de endereço de memória e sinais para a memória escrever o valor disponível nos pinos de dados.

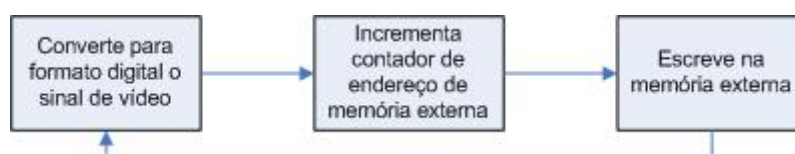


Figura 31. Controle da captura do sinal de vídeo pelo programa.

5.2 Módulos do programa

O programa é constituído de cinco módulos que possuem responsabilidades distintas: controle de contador de endereçamento, controle de memória, processamento de formato bmp, controle de transmissão serial e controle do fluxo principal. A seguir é apresentada a organização dos cinco módulos com seus respectivos mapeamentos em arquivos de código fonte.

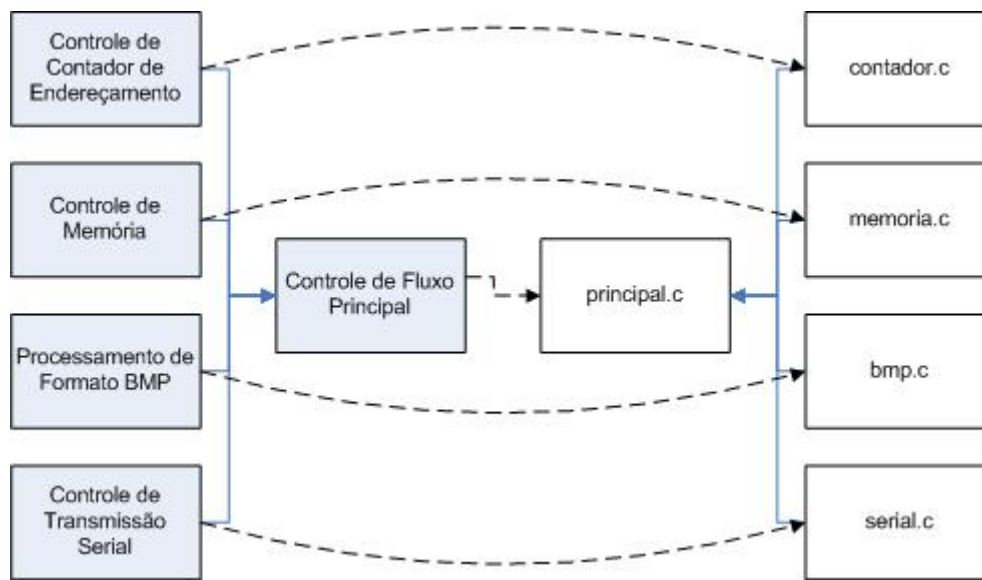


Figura 32. Organização dos módulos do programa com mapeamento nos respectivos arquivos “.c”.

O módulo para controle de contador de endereçamento possui duas funções *zerar_contador()* e *inc_contador()* que são responsáveis por definir o valor de saída do contador como zero ou com o valor anterior incrementado em uma unidade, respectivamente. Como se pode observar na Figura 32, não há um módulo responsável pelo controle do conversor A/D, pois o projeto físico do circuito condicionador foi projetado para acionar o conversor A/D automaticamente a cada envio de sinal para incremento de endereço.

O módulo para controle de memória possui as funções *memoria_init()*, *write_byte()* e *read_byte()* com responsabilidades para garantir as configurações iniciais, escrever um *byte* e ler um *byte* da memória, respectivamente. Neste módulo, é importante destacar que a função *write_byte()* é usada para gravar na memória o valor do *pixel* capturado pelo conversor A/D e por isso possui tempo de execução crítico, então esta função foi definida como uma macro a fim de evitar acréscimo de tempo por causa das operações de chamada e retorno da função.

O terceiro módulo, processamento de formato *bitmap* [22], é responsável por montar um cabeçalho de arquivo conforme os requisitos necessários para o formato de arquivo de imagem *bitmap* (.bmp). Esse cabeçalho é usado apenas no momento inicial da transmissão da imagem armazenada para a saída serial. Os detalhes dessa função que monta o cabeçalho *bitmap* serão abordados em uma próxima seção ainda neste capítulo.

O quarto módulo é responsável por configurar a taxa de transmissão que será utilizado pela EUSART do microcontrolador através da função *serial_init()*. Este módulo também implementa uma função para enviar, *phys_send()*, e uma função para receber bytes, *phys_recv()*. Como a operação de envio ocorre logo após o armazenamento do quadro da imagem de vídeo, e uma vez que não existem requisitos críticos para a transmissão dos dados a taxa de transmissão utilizada é de 9600bps. É importante lembrar que apesar do escopo deste projeto não incluir a

recepção de dados, esta funcionalidade está disponível para uso em trabalhos futuros, como por exemplo, implementação de um protocolo de envio de imagens.

O módulo principal deste programa possui o controle da máquina de estados, sendo responsável, portanto, pela lógica do fluxo de programa completo. Conforme visto na seção anterior, a máquina de estados é utilizada para identificar os sinais de sincronismo, armazenar e transmitir o quadro da imagem de vídeo capturado.

Uma característica importante que deve ser considerada na implementação do programa referente-se ao compartilhamento do barramento de dados no esquema elétrico do circuito condicionador, conforme mostra a Figura 33. Dois sinais de controle OE_TRA (*transceiver output enable*) e OE_MEM (*memory output enable*) são utilizados para indicar quais os componentes utilizarão o barramento.

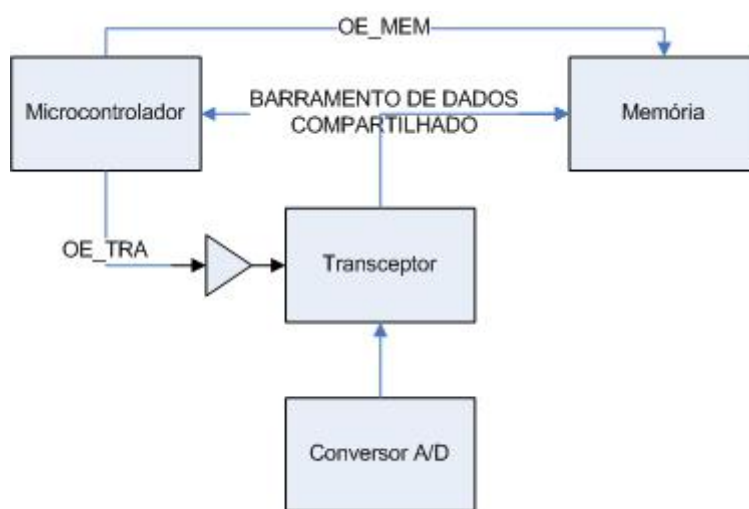


Figura 33. Barramento de dados compartilhado e os sinais: OE_TRA e OE_MEM.

O barramento de dados é compartilhado entre uma porta I/O do microcontrolador, a porta de saída do transceptor e a porta de I/O da memória. Conforme visto no capítulo anterior, o transceptor entre o conversor A/D e o barramento de dados é utilizado para isolar o conversor A/D do resto do circuito elétrico em determinados momentos. Esta operação é necessária para evitar danos nas portas dos componentes causados por conflitos entre sinais de saída de mais de um componente no mesmo barramento.

Em resumo, o programa evita os conflitos que podem ocorrer através do cumprimento das restrições listadas na Tabela 12:

Tabela 12. Definição dos sinais OE_TRA e OE_MEM conforme operação desejada.

Operação	Porta origem	Porta destino	OE_TRA	OE_MEM
Escrever	Conversor A/D	Memória	1	1
Escrever	Microcontrolador	Memória	0	1
Ler	Memória	Microcontrolador	0	0

O circuito condicionador realiza uma inversão externa no sinal OE_TRA antes dele alcançar o componente transceptor, conforme ilustra a Figura 33. Embora esta operação seja desnecessária, pois o programa tem controle sobre esses dois sinais, alternativamente apenas o sinal OE_MEM poderia ser controlado pelo programa, considerando a existência de uma ligação física entre os dois sinais de controle dos componentes.

Em outras palavras, o esquema elétrico foi projetado de forma que em trabalhos futuros uma instrução (OE_TRA) não seja necessária, otimizando assim o desempenho do sistema. Caso essa alternativa seja empregada, a operação de escrever do microcontrolador para a memória não poderia ser realizada, pois existiriam conflitos entre sinais de saída. Sendo assim, a relação entre o custo e benefício deve ser avaliada conforme o projeto.

Na abordagem do trabalho proposto, a operação de escrita do microcontrolador para a memória é necessária para inserir marcadores na imagem que está sendo armazenada, esses marcadores serão tratados na seção seguinte.

5.3 Formato de armazenamento

À medida que os *bytes* são capturados pelo conversor A/D, eles são repassados para as posições disponíveis na memória externa SRAM. A capacidade de armazenamento da memória utilizada neste projeto permite armazenar 32Kx8-bit, logo apenas 32x10⁶ *pixels* podem ser guardados no máximo. Contudo, para a proposta da implementação desse programa foi implementada a captura de um quadro de imagem com apenas 40x30 *pixels* de resolução que conseqüentemente pode ser armazenado na memória SRAM escolhida, com comodidade.

Cada *pixel* obtido através do conversor A/D é representado por 8-bits e será armazenado na memória a partir da posição zero. A única formatação existente nesse processo é a inserção de marcadores para indicar o início de cada uma das linhas do quadro. Um marcador é composto pela seqüência de caracteres r, o, w que é inserida antes de todos os *pixels* de cada uma das linhas do quadro de imagem. A Figura 34 ilustra uma representação do quadro de imagem armazenado na memória com os marcadores posicionados adequadamente.

Endereço	Marcador de linha			1º Pixel	2º Pixel	3º Pixel	...	40º Pixel
0x0000	0x72	0x6F	0x77					
0x002B	0x72	0x6F	0x77					
0x0056	0x72	0x6F	0x77					
.	Cabeçalho de Linha			Dados da Imagem				
.								
.								
0x04DF	0x72	0x6F	0x77					

0x72	'r'
0x6F	'o'
0x77	'w'

Figura 34. Formato de armazenamento do quadro de imagem na memória.

5.4 Formato de saída

Com o objetivo de transformar a imagem armazenada em um arquivo que pode ser compreendido por um programa de visualização de imagem comum, a construção do cabeçalho de uma imagem com formato *bitmap* (BMP) foi implementada neste programa. A construção desse cabeçalho é usada apenas quando a imagem é repassada pela saída serial do microcontrolador para um dispositivo externo ao circuito.

Arquivos BMP são um formato de arquivo histórico utilizado pelo sistema operacional *Windows*. Imagens BMP podem ter apenas as cores preto e branco ou até 16,7 milhões de cores

[23]. Este tipo de arquivo também pode possuir compressão dos dados, porém isso é uma tarefa raramente implementada e está além do escopo deste projeto. A seguir são apresentados resumidamente as características deste formato de arquivo.

5.4.1 Estrutura

A Figura 35 ilustra a estrutura de um arquivo *bitmap* (também conhecido como arquivo BMP). Este tipo de arquivo consiste de 3 ou 4 partes.

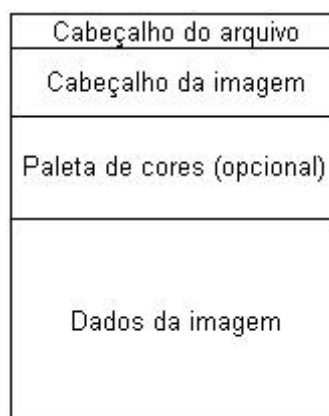


Figura 35. Estrutura de um arquivo *bitmap* (.bmp).

A primeira parte é o cabeçalho do arquivo, seguida pelo cabeçalho da imagem, se a imagem possui índice de cor então uma paleta de cores é a parte seguinte, e a última contém todos os *pixels* da imagem. A posição dos dados da imagem referente ao início do arquivo está informada no cabeçalho. Informações sobre largura, altura, tipo de compressão, números de cores estão contidas no cabeçalho da imagem.

5.4.2 Cabeçalho do arquivo

O cabeçalho de arquivo é constituído pelos campos da Tabela 13 e tem 14 bytes de comprimento.

Tabela 13. Campos do cabeçalho de um arquivo *bitmap*.

Byte de início	Tamanho (bytes)	Nome do campo	Descrição
1	2	Tipo	Deve ser sempre definido para BM (0x42, 0x4D) para informar que este é um arquivo .bmp
3	4	Tamanho	Especifica o tamanho do arquivo em <i>bytes</i> .
7	2	Reservado1	Deve ser sempre 0.
9	2	Reservado2	Deve ser sempre 0.
11	4	Deslocamento	Especifica o deslocamento a partir do início do arquivo para os dados da imagem .bmp

Os campos úteis dentro dessa estrutura de cabeçalho são o campo de tipo (que deve ser 'BM') que é uma pequena verificação que determina a legitimidade do arquivo BMP, e os campos de deslocamento que fornecem o número de *bytes* antes dos dados da imagem realmente, os *pixels*.

5.4.3 Cabeçalho da imagem

Os dados do cabeçalho da imagem possuem 40 *bytes* de comprimento, todos os campos estão descritos na Tabela 14. Os campos de maior interesse são largura e altura, número de *bits* por *pixel* (1 – preto e branco, 4 – para 16 cores, 8 - para 256 cores ou 24 – para 16,7 milhões de cores), o número de planos (o padrão assumido é 1) e o tipo de compressão.

Tabela 14. Campos do cabeçalho de imagem de um arquivo *bitmap*.

Byte de início	Tamanho (bytes)	Nome do campo	Descrição
15	4	Tamanho	Especifica o tamanho do cabeçalho da imagem, em <i>bytes</i> .
19	4	Largura	Especifica a largura da imagem, em <i>pixels</i> .
23	4	Altura	Especifica a altura da imagem, em <i>pixels</i> .
27	2	NumeroPlanos	Especifica o número de planos do dispositivo de destino.
29	2	NumeroBits	Especifica o número de <i>bits</i> por <i>pixel</i> .
31	4	Compressão	Especifica o tipo de compressão, geralmente 0 para indicar sem compressão.
35	4	TamanhoImagem	Especifica o tamanho dos dados da imagem, em <i>bytes</i> . Se não existir compressão, é válido o valor 0.
39	4	ResoluçãoX	Especifica o número de <i>pixels</i> horizontais por metro no projeto do dispositivo de destino. Usualmente, definido com o valor 0.
43	4	ResoluçãoY	Especifica o número de <i>pixels</i> verticais por metro no projeto do dispositivo de destino. Usualmente, definido com o valor 0.
47	4	NumeroCores	Especifica o número de cores usados no <i>bitmap</i> . Se estiver definido como zero, o número de cores é calculado através do campo NumeroBits.
51	4	CoresImportantes	Especifica o número de cores que são importantes para o <i>bitmap</i> . Se estiver definido como zero, todas as cores são importantes.

5.4.4 Índice de cores

O campo NumeroBits especifica o número de *bits* por *pixel*, conseqüentemente a resolução da imagem é especificada por este campo. Os valores possíveis são: 1 (preto/branco); 4 (16 cores); 8 (256 cores); 24 (16,7 milhões de cores). Este campo também determina se existirá a paleta de cores após o cabeçalho da imagem.

A implementação para o formato de saída deste projeto utilizou o número de *bits* por *pixel* definido para 8, conseqüentemente a imagem transmitida possui uma paleta de cores com 4 *bytes* para cada cor indexada. Os primeiros três *bytes* correspondem aos componentes B,G,R e o último *byte* é reservado e não é usado. Para imagens em escala de cinza de 8-*bits*, como para o projeto em questão (a câmera usada é preto e branco), esse índice de cores é geralmente uma rampa de escala de cinza.

Fazendo a soma, o tamanho do cabeçalho do arquivo mais o tamanho do cabeçalho da imagem mais 4 vezes o número de cores da paleta de cores devem ser igual ao deslocamento dos

dados de imagem propriamente. Em outras palavras: $14 + 40 + (4 * \text{NumeroCores}) = \text{Deslocamento}$. Onde $(4 * \text{NumeroCores})$ representa o tamanho da paleta de cores em bytes.

5.5 Transmissão serial

A transmissão do quadro de imagem armazenado na memória através da porta serial do microcontrolador é uma tarefa relativamente fácil, uma vez que existe uma EUSART implementada em *hardware* disponível para essa finalidade.

A Figura 36 apresenta a operação de transmissão:

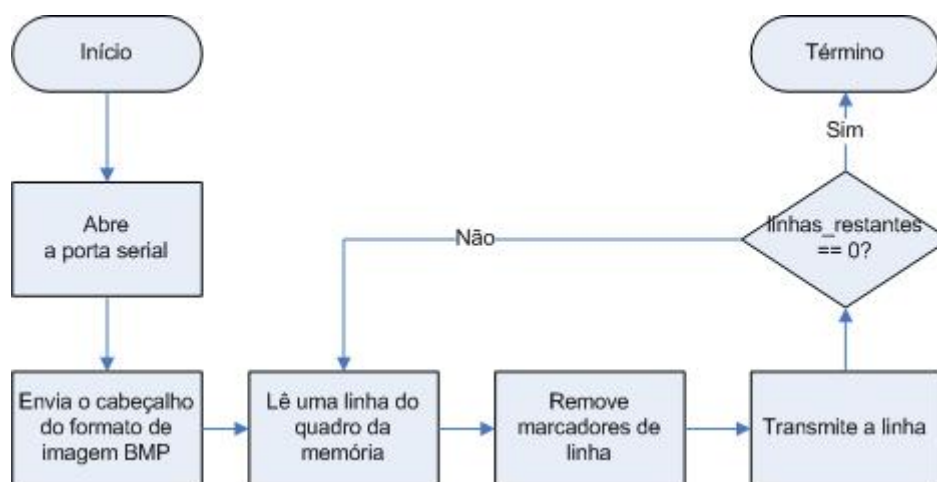


Figura 36. Fluxograma da transmissão serial através do microcontrolador.

Esta tarefa consiste inicialmente em habilitar a porta de transmissão serial do microcontrolador e em enviar um cabeçalho do formato de imagem *bitmap* conforme os parâmetros da imagem armazenada. Em seguida, cada uma das linhas é recuperada da memória, removem-se os marcadores de linha e então a linha é transmitida. Esta operação de transmissão de linha ocorre até que todas as linhas tenham sido enviadas pela porta serial. É importante destacar que a transmissão de uma linha consiste em escrever todos os bytes da linha na EUSART.

Capítulo 6

Conclusões e Trabalhos Futuros

Este trabalho propôs uma metodologia para captura de imagens usando microcontroladores. Uma câmera analógica monocromática foi utilizada como fonte geradora de imagem, e uma arquitetura para prototipação de um circuito condicionador foi implementada como forma de validação da metodologia empregada nesse projeto.

O enfoque principal deste trabalho foi a captura, armazenamento e transmissão de uma imagem digital através da saída serial do microcontrolador. Aspectos de desempenho foram analisados apenas para garantir os requisitos mínimos de tempo e de armazenamento de uma imagem, os quais contribuíram para a definição dos componentes usados no projeto do circuito condicionador.

É importante lembrar que este estudo não esperou resultados ótimos quanto à resolução da imagem capturada, contudo a arquitetura proposta foi projetada visando permitir melhorias que podem ser facilmente implementadas quanto à resolução da imagem, como, por exemplo, utilizando contadores, memória e microcontrolador com melhores desempenhos.

Este trabalho limitou-se a obter apenas um quadro de imagem de vídeo composto com a pequena resolução de 40×30 *pixels* devido às limitações de tempo de resposta da memória, do transceptor e dos contadores. Uma das limitações importantes refere-se ao tempo de instrução de 83,333ns do microcontrolador usando um *clock* de 48MHz de frequência, o qual impõe limitações quanto ao número de operações de amostragem, pois apesar do conversor A/D utilizado possuir ótimo desempenho, o microcontrolador limita-o devido ao tempo de resposta dos demais componentes.

Por fim, e não menos importante, o quadro de imagem digital obtido nos resultados mostrou que a abordagem usando microcontroladores para captura do sinal de vídeo composto é aplicável de forma satisfatória dependendo da escolha dos componentes eletrônicos e que independente do desempenho dos componentes escolhidos, a arquitetura do circuito condicionador foi validada com sucesso.

6.1 Contribuições

Atualmente aplicações para uso de imagens digitais são encontradas nas mais variadas áreas, como medicina, telecomunicações, segurança entre outras. Sendo assim, espera-se que a metodologia proposta contribua para realização de estudos sobre o desenvolvimento de ferramentas úteis em tais campos de atuação.

Outra contribuição deste trabalho é apresentar uma alternativa de implementação satisfatória para solução de problemas complexos usando um circuito integrado simples com um custo relativamente baixo – o uso de microcontroladores.

6.2 Dificuldades

Entre as dificuldades encontradas no decorrer deste trabalho, vale ressaltar a procura de fornecedores para aquisição dos componentes do circuito elétrico. Esta barreira foi transposta limitando as características dos componentes para satisfazer apenas os requisitos mínimos de tempo de resposta, capacidade de armazenamento e desempenho, de forma que pudessem ser obtidos no mercado fornecedor mais próximo.

Também é importante ressaltar as dificuldades na elaboração do esquema elétrico e o *layout* da placa devido à pequena experiência na implementação de projetos com circuitos eletrônicos. Esse obstáculo foi superado buscando-se a contribuição do conhecimento prático de um profissional experiente no desenvolvimento de projetos com circuitos eletrônicos.

A etapa de prototipação exigiu que todos os componentes fossem inseridos na placa de circuito impresso devidamente soldados. Contudo, o componente LTC1406 referente ao conversor A/D que foi utilizado possui encapsulamento de pinos SSOP cuja fabricação é projetada para máquinas automatizadas precisas realizem a solda dos pinos corretamente, logo uma das dificuldades foi soldar este componente na placa através de um processo manual e de pouca precisão. Essa barreira foi transposta através da utilização de uma lente de aumento e um ferro de solda com uma ponta bastante fina.

6.3 Trabalhos futuros

Acredita-se que este trabalho pode ser estendido em várias direções. O próximo passo deste trabalho seria a utilização de componentes com melhores desempenhos, o que permitiria a captura e armazenamento dos *pixels* mais rapidamente. A partir de um circuito com um tempo de resposta menor podem-se implementar projetos para a captura de vários quadros de imagem de vídeo composto e a transmissão através de uma saída serial USB, em tempo real.

Essas melhorias permitiriam a captura de mais amostras do sinal de imagem e conseqüentemente a resolução da imagem armazenada seria de maior qualidade. Além disso, técnicas de processamento de imagem poderiam ser aplicadas para gerar vídeos digitais em formatos com compressão de dados, como por exemplo, o formato MPEG [9].

O projeto desenvolvido pode servir, também, como base para o desenvolvimento de trabalhos com sistemas de monitoramento de ambientes, sistemas complementares de segurança remota, como também na elaboração de sistemas de visão computacional para robôs. Além disso, os trabalhos futuros poderiam incluir a transmissão das imagens por intermédio da rádio frequência, de redes *Ethernet* ou de redes móveis TCP/IP [31] usadas pela tecnologia celular GPRS [32].

Bibliografia

- [1] SMITH, S. W. *The Scientist and Engineer's Guide to Digital Signal Processing*, 2nd edition. California Technical Publishing: 1999.
- [2] GONZALEZ, R. C.; WINTZ, P. *Digital Image Processing*, 2nd edition. Addison Wesley: 1987.
- [3] IOVINE, J. *PIC Robotics - A Beginner's Guide to Robotics Projects Using the PICmicro*. McGrawHill, 2004.
- [4] SCURI, A.S. *Fundamentos da Imagem Digital*. Pontifícia Universidade Católica do Rio de Janeiro, 1999.
- [5] JACK, K. *Video Demystified, A Handbook for the Digital Engineer*, 4th edition. Elsevier: 2005.
- [6] NATIONAL INSTRUMENTS. *Video Signal Measurement Fundamentals*. Disponível em: <http://zone.ni.com/devzone/conceptd.nsf/webmain/0E5DAD8917692B5D86256F3500549509>, visitado em 18/10/2005.
- [7] NATIONAL INSTRUMENTS. *Anatomy of a Video Signal*. Disponível em: <http://zone.ni.com/devzone/devzoneweb.nsf/Opendoc?openagent&BB087524D4052C9E8625685E0080301B#3>, visitado em 18/10/2005.
- [8] MATIC, N. *The Microchip PIC Microcontroller, for beginners too*, vol. 1. Disponível em: <http://www.mikroelektronika.co.yu/english/product/books/PICbook/picbook.htm>, visitado em: 10/10/2005.
- [9] WESTWATER, R.; FURHT, B. *Real-Time Video Compression, Techniques and Algorithms*. Kluwer Academic Publishers, 1997.
- [10] MICROCHIP TECHNOLOGY INC. Disponível em: <http://www.microchip.com>, visitado em 05/03/2005.
- [11] MICROCHIP TECHNOLOGY INC. *PIC18F2455/2550/4455/4550 Data Sheet, 28/40/44-Pin High Performance, Enhanced Flash USB Microcontrollers with nanoWatt Technology*. 2004. 426 p.
- [12] LINEAR TECHNOLOGY CORPORATION. Disponível em: <http://www.linear.com>, visitado em 05/09/2005.
- [13] LINEAR TECHNOLOGY CORPORATION. *LTC1406, Low Power, 8-Bit, 20Msps, Sampling A/D Converter*. 1998. 16 p.
- [14] NATIONAL SEMICONDUCTOR CORPORATION. Disponível em: <http://www.national.com>, visitado em 02/10/2005.
- [15] NATIONAL SEMICONDUCTOR CORPORATION. *LM1881, LM1881-X Video Sync Separator*. 2003. 12 p.
- [16] HYUNDAI ELECTRONICS AMERICA. Disponível em: <http://www.heacom>, visitado em: 15/08/2005.
- [17] HYUNDAI ELECTRONICS AMERICA. *32Kx8-bit CMOS SRAM*. Disponível em: <http://www.datasheetcatalog.com>, visitado em: 16/09/2005, 15 p.

- [18] TEXAS INSTRUMENTS INCORPORATED. Disponível em: <http://www.ti.com>, visitado em: 02/11/2005.
- [19] TEXAS INSTRUMENTS INCORPORATED. *SN54HC4040, SN74HC4040 12-Bit Asynchronous Binary Counters*. 2003. 17 p.
- [20] PHILIPS SEMICONDUCTORS. Disponível em: <http://www.semiconductors.philips.com>, visitado em 02/11/2005.
- [21] PHILIPS SEMICONDUCTORS. *Product specification 74HC/HCT245 Octal bus transceiver; 3-state*. 1993. 7 p.
- [22] <http://astronomy.swin.edu.au/~pbourke/dataformats/bmp/>, visitado em: 05/11/2005.
- [23] <http://www.fortunecity.com/skyscraper/windows/364/bmpffrmt.html>, visitado em: 05/11/2005.
- [24] SCHILDT, H. C, completo e total. 3ª edição. São Paulo: Makron Books, 1996.
- [25] HI-TECH SOFTWARE, S. Picc-18 C Manual. 7th edition. Australia: 2003.
- [26] HI-TECH SOFTWARE. Disponível em: <http://www.htsoft.com>, visitado em: 02/08/2005.
- [27] SEGSAT IND. E COM. LTDA. Disponível em: <http://www.segsat.com>, visitado em: 10/11/2005.
- [28] WIKIPEDIA ENCYCLOPEDIA. Disponível em: <http://www.wikipedia.com>, visitado em: 02/08/2005.
- [29] <http://gedabr.projetos.etc.br/article/articleview/5/1/3/>, visitado em: 20/10/2005.
- [30] <http://gedabr.projetos.etc.br/article/articleview/13/1/3/>, visitado em 20/10/2005.
- [31] KUROSE, J. F.; ROSS, K. W. *Redes de Computadores e a Internet, uma nova abordagem*. Addison Wesley, p. São Paulo, 2003.
- [32] <http://www.gsmworld.com/technology/gprs/index.shtml>, visitado em 22/12/2005.
- [33] HO, S. W. *Adaptive Modulation (QPSK, QAM)*. Intel Literature Center, 2004.

Apêndice A

Lista de Componentes

Este apêndice lista todos os componentes utilizados no projeto do circuito condicionador, bem como seus respectivos valores, referências, quantidades e símbolos.

Quantidade	Símbolo	Valor	Referência
2	C	22pF	C1, C2
14	C	100nF	C3, C4, C5, C6, C7, C19, C21, C25, C26, C27, C30, C31, C32, C33
4	C	1uF	C8,C9,C10,C11
2	C	56pF	C12, C15
7	C	10uF	C13, C20, C22, C23, C24, C28, C29
1	C	220pF	C14
1	CH	RESET	CH1
1	CN	RS232	CN1
1	CN	USB	CN2
1	CN	12v	CN3
1	CN	VIDEO	CN4
1	L	4,7uH	L1
2	Q	BC547	Q1, Q3
1	Q	BC557	Q2
1	Q	BC337	Q4
1	R	620R	R1
2	R	75R	R2, R18
1	R	680K	R3
3	R	10K	R4,R5, R16
1	R	22K	R6
1	R	8K2	R7
1	R	3K9	R8
3	R	100R	R9, R12, R20
1	R	220R	R10
1	R	120R	R11
1	R	1K2	R13
1	R	1K	R14

2	R	22K	R15, R17
1	R	2K2	R19
1	U	PIC184455	U1
1	U	MAX232	U2
1	U	HY62256A	U3
2	U	74HC4040AN	U4,U5
1	U	LM1881	U6
1	U	74HC245AN	U7
1	U	74HC04	U8
1	U	LTC1406CGN	U9
1	U	7805	U10
1	U	LM336-2,5	U11
2	X	12MHz	X1,X2

Notas:

R – Resistor

C – Capacitor

CH – Chave

CN – Conector

L – Transistor

Q – Transistor

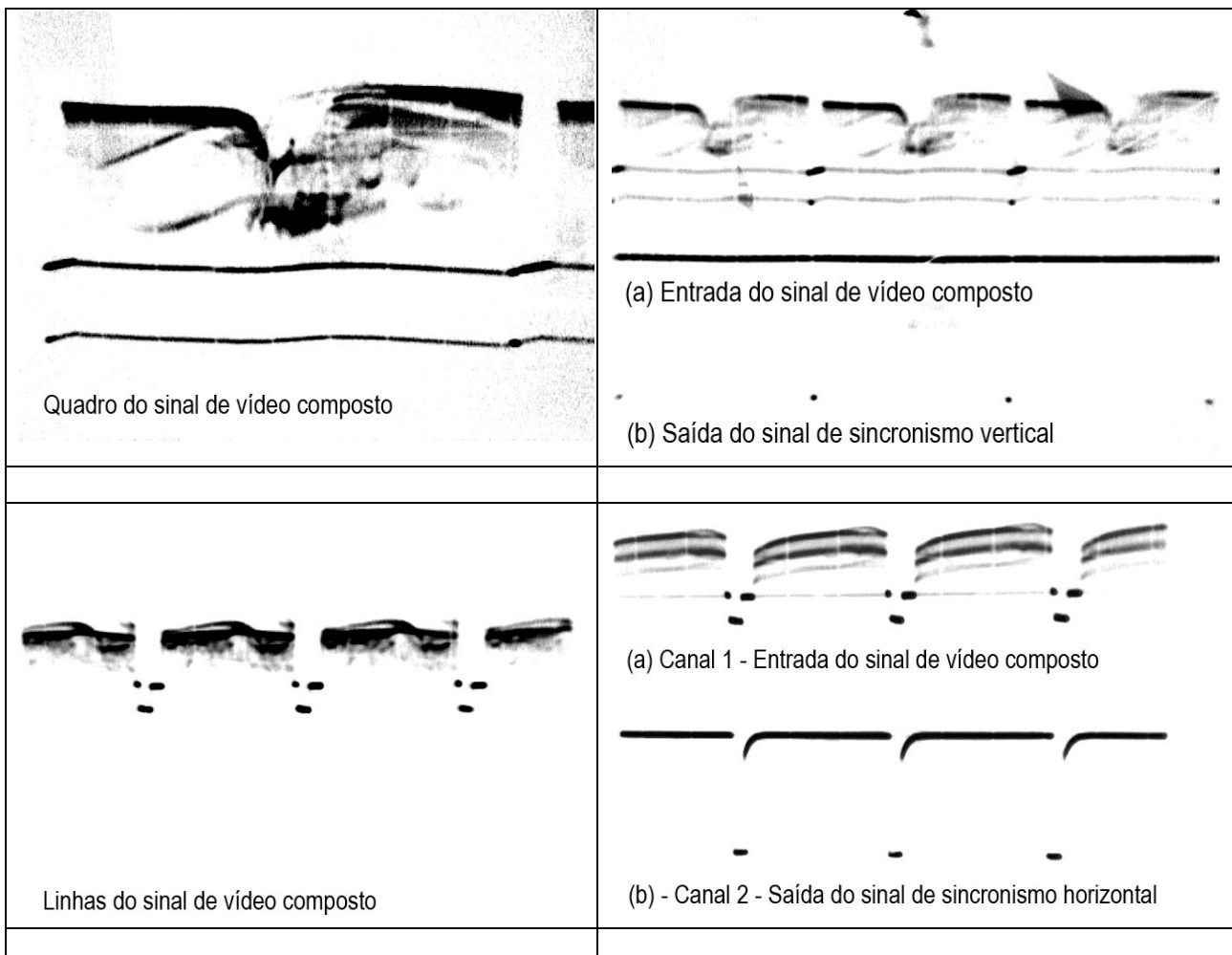
U – Circuito Integrado





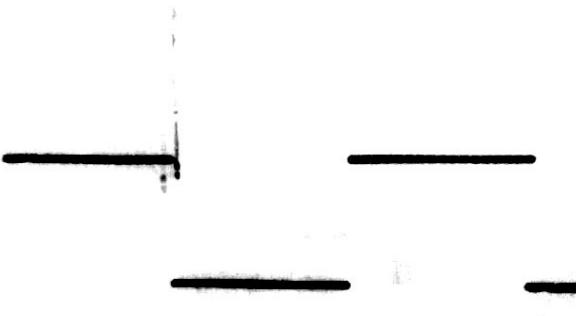
X – Cristal

Apêndice B

Experimentos

Este apêndice relaciona as imagens sobre sinais de sincronismo as quais foram analisadas com o uso de uma ferramenta de osciloscópio durante experimentos realizados na implementação do projeto do circuito condicionador.



 <p>(a) Canal 1 - Entrada do sinal de vídeo composto</p>  <p>(b) Canal 2 - Saída do sinal de campo ímpar/par</p>	 <p>(a) Canal 1 - Entrada do sinal de vídeo composto</p>  <p>(b) Canal 2 - Saída do sinal de campo ímpar/par</p>
 <p>Saída do sinal de campo ímpar/par</p>	

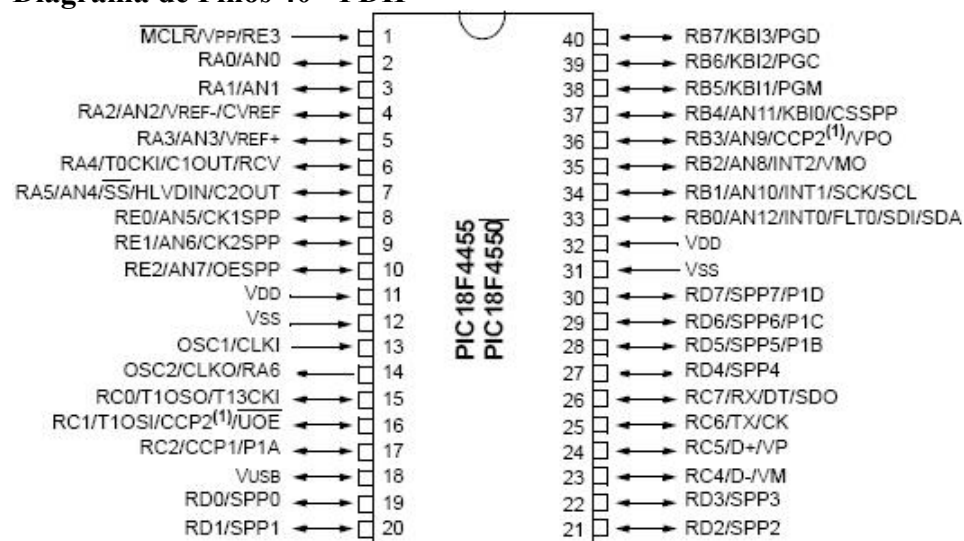
Apêndice C

Pinos dos Componentes

Este apêndice lista a descrição dos pinos dos componentes utilizados no projeto do circuito condicionador, como também o diagrama da posição dos pinos fisicamente.

Microcontrolador 18F4455

Diagrama de Pinos 40 - PDIP



Descrição dos Pinos

Pino	Símbolo	Descrição
1	/MCLR	Master Clear Input
	VPP	Programming Voltage Input
	RE3	Digital Input
2	RA0	Digital I/O
	AN0	Analog Input 0
3	RA1	Digital I/O
	AN1	Analog Input 1
	RA2	Digital I/O
	AN2	Analog Input 2

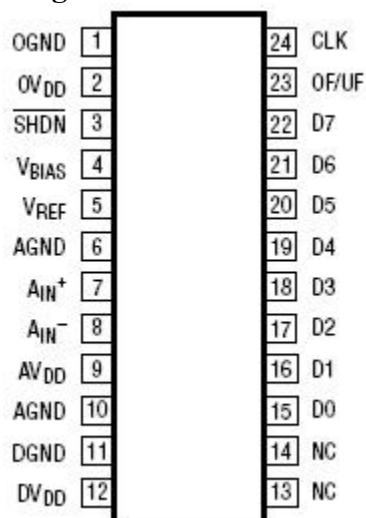
	VREF-	A/D Reference Voltage (low) Input
	CVREF	Analog Comparator Reference Output
5	RA3	Digital I/O
	AN3	Analog Input 3
	VREF+	A/D Reference Voltage (high) Input
6	RA4	Digital I/O
	T0CKI	Timer0 External Clock Input
	C1OUT	Comparator 1 Output
	RCV	External USB Transceiver RCV Input
7	RA5	Digital I/O
	AN4	Analog Input 4
	/SS	SPI Slave Select Input
	HLVDI	High/Low Voltage Detect Input
	C2OUT	Comparator 2 Output
8	RE0	Digital I/O
	AN5	Analog Input 5
	CK1SPP	SPP Clock 1 Output
9	RE1	Digital I/O
	AN6	Analog Input 6
	CK2SPP	SPP Clock 2 Output
10	RE2	Digital I/O
	AN7	Analog Input 7
	OESPP	SPP Output Enable Output
11	VDD	Positive Supply for Logic and I/O Pins
12	VSS	Ground Reference for Logic and I/O Pins
13	OSC1	Oscillator Crystal Input or External Clock Source Input
	CLKI	External Clock Source Input
14	OSC2	Oscillator Crystal Input
	CLKO	In RC mode, OSC2 pin outputs CLKO which has $\frac{1}{4}$ the frequency of OSC1 and denotes the instruction cycle rate.
	RA6	General Purpose I/O
15	RC0	Digital I/O
	T1OSO	Timer1 Oscillator Output
	T13CKI	Timer1/Timer3 External Clock Input
16	RC1	Digital I/O
	T1OSI	Timer1 Oscillator Input
	CCP2	Capture 2 Input/Compare 2 Output/PWM 2 Output
	/UOE	External USB Transceiver /OE Output
17	RC2	Digital I/O
	CCP1	Compare1 Input/Compare 1 Output/PWM 1 Output
	P1A	Enhanced CCP1 PWM Output, Channel A
18	VUSB	Internal USB 3.3V Voltage Regulator Output
19	RD0	Digital I/O
	SPP0	Streaming Parallel Port Data
20	RD1	Digital I/O
	SPP1	Streaming Parallel Port Data
21	RD2	Digital I/O

	SPP2	Streaming Parallel Port Data
22	RD3	Digital I/O
	SPP3	Streaming Parallel Port Data
23	RC4	Digital I/O
	D-	USB Differential Minus Line (Input/Output)
	VM	External USB Transceiver VM Input
24	RC5	Digital I/O
	D+	USB Differential Plus Line (Input/Output)
	VP	External USB Transceiver VP Input
	RC6	Digital I/O
	TX	EUSART Asynchronous Transmit
	CK	EUSART Synchronous Clock
26	RC7	Digital I/O
	RX	EUSART Asynchronous Receive
	DT	EUSART Synchronous Data
	SDO	SPI Data Out
27	RD4	Digital I/O
	SPP4	Streaming Parallel Port Data
28	RD5	Digital I/O
	SPP5	Streaming Parallel Port Data
	P1B	Enhanced CCP1 PWM Output, Channel B
29	RD6	Digital I/O
	SPP6	Streaming Parallel Port Data
	P1C	Enhanced CCP1 PWM Output, Channel C
30	RD7	Digital I/O
	SPP7	Streaming Parallel Port Data
	P1D	Enhanced CCP1 PWM Output, Channel D
31	VSS	Ground Reference for Logic and I/O Pins
32	VDD	Positive Supply for Logic and I/O Pins
33	RB0	Digital I/O
	AN12	Analog Input 12
	INT0	External Interrupt 0
	FLT0	Enhanced PWM Fault Input
	SDI	SPI Data In
	SDA	I2C Data Out
34	RB1	Digital I/O
	AN10	Analog Input 10
	INT1	External Interrupt 1
	SCK	Synchronous Serial Clock I/O for SPI Mode
	SCL	Synchronous Serial Clock I/O for I2C Mode
35	RB2	Digital I/O
	AN8	Analog Input 8
	INT2	External Interrupt 2
	VMO	External USB Transceiver VMO Output
36	RB3	Digital I/O
	AN9	Analog Input 9
	CCP2	Capture 2 Input/Compare 2 Output/PWM 2 Output

	VPO	External USB Transceiver VPO Output
37	RB4	Digital I/O
	AN11	Analog Input 11
	KBI0	Interrupt-on-change
	CSSPP	SPP Chip Select Control Output
38	RB5	Digital I/O
	KBI1	Interrupt-on-change
	PGM	Low-Voltage ICSP Programming Enable
39	RB6	Digital I/O
	KBI2	Interrupt-on-change
	PGC	In-Circuit Debugger and ICSP Programming Clock
40	RB7	Digital I/O
	KBI3	Interrupt-on-change
	PGD	In-Circuit Debugger and ICSP Programming Data

Conversor A/D LTC1406CGN

Diagrama de Pinos 24 - SSOP



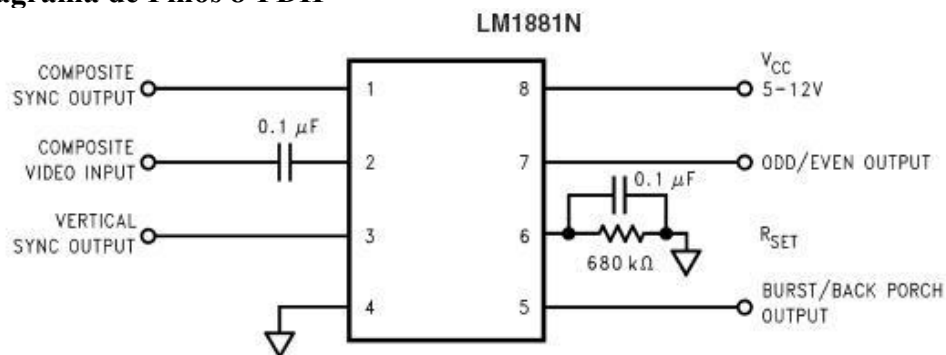
Descrição dos Pinos

Pino	Símbolo	Descrição
1	OGND	Digital Data Output Ground
2	OVDD	Digital Data Output Supply
3	/SHDN	Power Shutdown Input
4	VBIAS	Internal Bias Voltage
5	VREF	External 2.5V Reference Input
6	AGND	Analog Ground
7	AIN+	±1V Input
8	AIN-	±1V Input
9	AVDD	Analog 5V Positive Supply
10	AGND	Analog Ground
11	DGND	Digital Ground for Internal Logic

12	DVDD	Digital 5V Positive Supply
13-14	NC	No Internal Connection
15-22	D0-D7	Digital Data Output
23	OF/UF	Overflow/Underflow Bit
24	CLK	Clock Input

Separador de Sincronismo LM1881

Diagrama de Pinos 8-PDIP

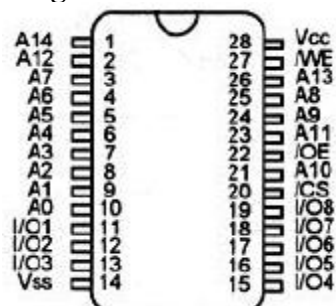


Descrição dos Pinos

Pino	Símbolo	Descrição
1	CS	Composite Sync Output
2	VIDEO	Composite Vídeo Input
3	VS	Vertical Sync Output
4	GND	Ground
5	BRST	Burst/Back Porch Output
6	RSET	Reset
7	OE	Odd/Even Output
8	VCC	Power Supply

Memória SRAM HY62256A

Diagrama de Pinos 20 - PDIP



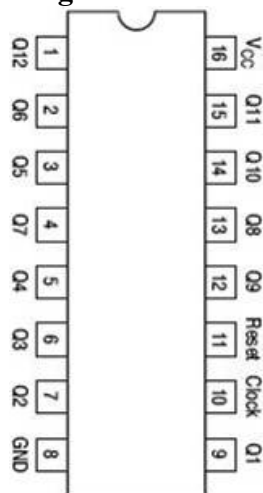
Descrição dos Pinos

Pino	Símbolo	Descrição
------	---------	-----------

1	A14	Address Input
2	A12	Address Input
3-10	A7-A0	Address Input
11-13	I/O1-I/O3	Data Input/Output
14	VSS	Ground
15-19	I/O4-I/O8	Data Input/Output
20	/CS	Chip Select
21	A10	Address Input
22	/OE	Output Enable
23	A11	Address Input
24	A9	Address Input
25	A8	Address Input
26	A13	Address Input
27	/WE	Write Enable
28	VCC	Power

Contador 74HC4040

Diagrama de Pinos 16 - PDIP



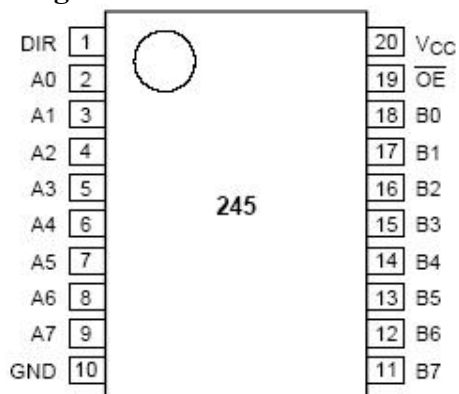
Descrição dos Pinos

Pino	Símbolo	Descrição
1	Q12	Parallel Output
2	Q6	Parallel Output
3	Q5	Parallel Output
4	Q7	Parallel Output
5	Q4	Parallel Output
6	Q3	Parallel Output
7	Q2	Parallel Output
8	GND	Ground
9	Q1	Parallel Output
10	CLOCK	Clock Input
11	RESET	Master Reset Input

12	Q9	Parallel Output
13	Q8	Parallel Output
14	Q10	Parallel Output
15	Q11	Parallel Output
16	VCC	Positive Supply Voltage

Transceptor 74HC245

Diagrama de Pinos 20 - PDIP



Descrição dos Pinos

Pino	Símbolo	Descrição
1	DIR	Direction Control
2-9	A0-A7	Data Input/Output
10	GND	Ground
11-18	B7-B0	Data Input/Output
19	\overline{OE}	Output Enable Input
20	VCC	Supply Voltage