



UNIVERSIDADE LUTERANA DO BRASIL
PRÓ-REITORIA DE GRADUAÇÃO
DEPARTAMENTO DE ENGENHARIA ELÉTRICA



ANDRÉ MARTINS DE OLIVEIRA

DESENVOLVIMENTO DE UM DISPOSITIVO PARA COLETA E
ARMAZENAMENTO MASSIVO DE FALHAS

Canoas, dezembro de 2010



ANDRÉ MARTINS DE OLIVEIRA

**DESENVOLVIMENTO DE UM DISPOSITIVO PARA COLETA E
ARMAZENAMENTO MASSIVO DE FALHAS**

Trabalho de Conclusão de Curso
apresentado ao Departamento de
Engenharia Elétrica da ULBRA como um
dos requisitos obrigatórios para a obtenção
do grau de Engenheiro Eletricista

Departamento:

Engenharia Elétrica

Área de Concentração

Telecomunicações e Microprocessadores

Professor Orientador:

[Msc.] Eng. Eletr. Dalton Vidor– CREA-RS: 079005-D

Canoas - RS

2010



FOLHA DE APROVAÇÃO

Nome do Autor: André Martins de Oliveira

Matrícula: 992101751-9

Título: Desenvolvimento de dispositivo para coleta e armazenamento massivo de falhas

Trabalho de Conclusão de Curso
apresentado ao Departamento de
Engenharia Elétrica da ULBRA como um
dos requisitos obrigatórios para a obtenção
do grau de Engenheiro Eletricista

Professor Orientador:

[Msc.] Eng. Eletr. Dalton Vidor

CREA-RS: 079005-D

Banca Avaliadora:

[MSc. Eng. Eletr. Augusto A. D de Mattos

CREA-RS: 088003-D

Conceito Atribuído (A-B-C-D):

[MSc Eng. Eletr. Miriam Noemi Cáceres Villamayor

CREA-RS: 067231-D

Conceito Atribuído (A-B-C-D):

Assinaturas:

Autor
André Martins de Oliveira

Orientador
Dalton Vidor

Avaliador
Augusto A. D. de Mattos

Avaliador
Miriam Noemi Cáceres Villamayor

Relatório Aprovado em:



DEDICATÓRIA

Dedico a minha esposa Cristiane,
pelo carinho e por suportar minha ausência.



AGRADECIMENTOS

Aos meus pais Eleunice e Mário, pelo carinho e por incentivarem a busca do conhecimento desde minha infância

Aos meus familiares e amigos que suportaram e entenderam a minha ausência em muitos eventos importantes de nossas vidas.

Aos meus colegas de trabalho André Schneider e Rodrigo pelo apoio e compreensão.

Ao Professor orientador Dalton Vidor pelo esforço pessoal em me auxiliar e pelo aprendizado proporcionado

Aos professores André Bianchi e Augusto Mattos pelas importantes dicas.

Aos colegas Ricardo, Carlos e Gilmar, do Curso de Engenharia Elétrica, pelas observações valiosas para elaboração deste trabalho



EPÍGRAFE

“Um tempo que aprendi a entender as coisas do mar, a conversar com as grandes ondas e não discutir com o mau tempo. A transformar o medo em respeito, o respeito em confiança. Descobri como é bom chegar quando se tem paciência. E, para se chegar, onde quer que seja, aprendi que não é preciso dominar a força, mas a razão. É preciso, antes de mais nada, querer.”

Amyr Klink



RESUMO

OLIVEIRA, André Martins de. **Desenvolvimento de Dispositivo para coleta e armazenamento massivo de eventos de falhas**. Trabalho de Conclusão de Curso em Engenharia Elétrica - Departamento de Engenharia Elétrica. Universidade Luterana do Brasil. Canoas, RS. 2010.

O presente trabalho apresenta o desenvolvimento de um sistema de coleta e armazenamento massivo de falhas em memória do tipo SD Card, utilizando comunicação SPI e relógio independente. Para simulação desses eventos, foi desenvolvida uma interface eletrônica que analisa falhas em linhas digitais de telecomunicações. O dispositivo é ajustado para uma resolução de 1 segundo para as falhas, obtendo-se assim uma ferramenta útil de auxílio técnico, através do arquivamento seguro do histórico das falhas, para reparo em meios de telecomunicações com anomalias.

Palavras chave: SD Card, Codificação HDB3, G703, PIC18F4520, SPI



ABSTRACT

OLIVEIRA, André Martins de. **Development of Device for collecting and storing massive fault events.** Work of Conclusion of Course in Electrical Engineering Electrical Engineering Department. Lutheran University of Brazil. Canoas, RS. 2010.

This paper presents the development of a system for collect and storage of massive failures in memory type SD Card using SPI communication and independent clock. To simulate these events, we developed an electronic interface that analyzes failures in digital telecommunications lines. The device is set to a resolution of 1 second for the failures, thus obtaining a useful technical aid through the secure archiving of the history of failures, repair of telecommunications facilities with deficiencies

Keywords: SD Card, HDB3 Code, G703, PIC18F4520, SPI



LISTA DE FIGURAS

FIGURA 2-1 – LINHA DE TRANSMISSÃO PARALELA.....	4
FIGURA 2-2 – ANALOGIA DE UMA LINHA DE TRANSMISSÃO COM CIRCUITO ELÉTRICO.....	4
FIGURA 2-3 – ESTRUTURA TRANSVERSAL DE UM CABO COAXIAL.....	6
FIGURA 2-4 – SINAL ANALÓGICO COM RESPECTIVAS HARMÔNICAS.....	8
FIGURA 2-5 – SINAL ANALÓGICO COM RESPECTIVAS HARMÔNICAS.....	8
FIGURA 2-6 – REPRESENTAÇÃO DE UM SINAL DIGITAL.....	9
FIGURA 2-7– REPRESENTAÇÃO DA TAXA DE BITS POR UNIDADE DE TEMPO.....	9
FIGURA 2-8 – TRANSMISSÃO TIPO BANDA BASE.....	10
FIGURA 2-9– ETAPAS BÁSICAS DO PROCESSO DE CONVERSÃO DE SINAIS.....	10
FIGURA 2-10 – SINAL ANALÓGICO EM UM CANAL TELEFÔNICO.....	11
FIGURA 2-11– REPRESENTAÇÃO DA LARGURA DE BANDA.....	12
FIGURA 2-12– REPRESENTAÇÃO DO SINAL ANALÓGICO APÓS ETAPA DE AMOSTRAGEM (PAM).....	12
FIGURA 2-13– REPRESENTAÇÃO DO SINAL PAM APÓS A QUANTIZAÇÃO	13
FIGURA 2-14 – CODIFICAÇÃO DE LINHA DIGITAL	14
FIGURA 2-15 – CODIFICAÇÃO AMI	16
FIGURA 2-16 – CODIFICAÇÃO DE LINHA DIGITAL	17
FIGURA 2-17– REPRESENTAÇÃO DO PROCESSO DE MULTIPLEXAÇÃO	17
FIGURA 2-18– ESTRUTURA TDM UTILIZADA NO BRASIL	19
FIGURA 2-19– REPRESENTAÇÃO DO PROCESSO DE SINCRONISMO.....	20
FIGURA 2-20– MÁSCARA PADRÃO DE UM PULSO HDB-3.....	22
FIGURA 2-21 – TELA DO OSCILOSCÓPIO PARA TRÁFEGO NORMAL.....	23
FIGURA 2-22 – TELA DO OSCILOSCÓPIO PARA ALARME TIPO LOS.....	23
FIGURA 2-23 – TELA DO OSCILOSCÓPIO PARA ALARME TIPO AIS.....	24
FIGURA 2-24– ATENUAÇÃO.....	25
FIGURA 2-25– DISTORÇÃO.....	25
FIGURA 2-26 - RUÍDO.....	26
FIGURA 3-27– DIAGRAMA EM BLOCOS DO DISPOSITIVO.....	27
FIGURA 3-28 – FOTOGRAFIA COM VISÃO FRONTAL DO DISPOSITIVO.....	28
FIGURA 3-29 – FOTOGRAFIA COM TELA DE MANUTENÇÃO.....	28
FIGURA 3-30 – TELA DE GRAVAÇÃO DE DADOS NA MEMÓRIA SD CARD.....	29
FIGURA 3-31 – DIAGRAMA DE PINOS CHIP SN75176.....	30
FIGURA 3-32 – ESTRUTURA DE FUNCIONAMENTO DO RECEIVER DIFERENCIAL.....	30



FIGURA 3-33 – DIAGRAMA DE PINOS CIP SN74LS123.....	31
FIGURA 3-34 – CIRCUITO TÍPICO UTILIZADO NO CHIP SN74LS123.....	31
FIGURA 3-35 – DIAGRAMA DE SINAIS ENTRADA E SAÍDA NO CHIP SN74LS123.....	32
FIGURA 3-36 – ESTRUTURA DE FUNCIONAMENTO DO MULTIVIVRADOR MONOESTÁVEL.....	32
FIGURA 3-37– CIRCUITO COMPLETO COM AS 2 ETAPAS PARA DETECÇÃO DE FALHAS.....	33
FIGURA 3-38– FOTOGRAFIA COM A PLACA DE CIRCUITO DESENVOLVIDO.....	33
FIGURA 3-39– DIAGRAMA DE PINOS CHIP DS 1302.....	34
FIGURA 3-40– CIRCUITO TÍPICO DE LIGAÇÃO PARA CHIP DS 1302.....	35
FIGURA 3-41– CIRCUITO DE LIGAÇÃO CHIP DS 1302.....	35
FIGURA 3-42 – ETAPAS DE COMANDO CHIP DS 1302.....	36
FIGURA 3-43– BYTE DE COMANDO SPI CHIP DS1302.....	36
FIGURA 3-44– ETAPAS DE COMANDO SPI REDUZIDAS.....	37
FIGURA 3-45– PROCESSO DE LEITURA E ESCRITA SIMPLIFICADO NO DS1302.....	38
FIGURA 3-46 – FOTOGRAFIA COM DIMENSÃO DO TAMANHO DE UMA MICRO SD.....	39
FIGURA 3-47 – FOTOGRAFIA COM DIMENSÃO DO TAMANHO DE UM ADAPTADOR MICRO SD	40
FIGURA 3-48 – PLACA ACE SD CARD.....	40
FIGURA 3-49 – CIRCUITO DE CONVERSÃO DE SINAIS ELÉTRICOS	41
FIGURA 3-50 – DIAGRAMA DE PINOS PARA COMUNICAÇÃO SPI SD CARD.....	42
FIGURA 3-51 – ESTRUTURA DE COMUNICAÇÃO SPI MEMÓRIA SD CARD.....	43
FIGURA 3-52 – BITS DOS REGISTRADORES DE RESPOSTA SPI PARA SD CARD.....	44
FIGURA 3-53– FLUXOGRAMA DE INICIALIZAÇÃO MEMÓRIA SD CARD COM PROTOCOLO SPI.....	46
FIGURA 3-54 – EXEMPLO DE LEITURA REGISTRADOR CID	47
FIGURA 3-55 – EXEMPLO DE LEITURA REGISTRADOR CSD	47
FIGURA 3-56 – DIAGRAMA DE PINOS PIC18F4520.....	49
FIGURA 3-57 – MODELO DE PLANILHA EXCEL DESENVOLVIDA PARA CONSOLIDAÇÃO DOS EVENTOS.....	50
FIGURA 3-58 – FLUXOGRAMA DE SOFTWARE DESENVOLVIDO PARA MODO COLETA.....	52
FIGURA 3-59 – FLUXOGRAMA DE SOFTWARE DESENVOLVIDO PARA MODO MANUTENÇÃO	54
FIGURA 4-60– TELA DO SOFTWARE PARA ANÁLISE DE SETORES MEMÓRIA SD CARD.....	55
FIGURA 4-61 – TELA DO OSCILOSCÓPIO PARA TRÁFEGO NORMAL.....	57
FIGURA 4-62 – TELA DO OSCILOSCÓPIO NA SAÍDA DO RECEIVER DIFERENCIAL.....	58
FIGURA 4-63 – TELA DO OSCILOSCÓPIO NA SAÍDA DO MONOESTÁVEL RETRIGÁVEL	59
FIGURA 4-64– TOPOLOGIA DE TESTES COM PABX FABRICANTE ALCATEL-LUCENT.....	60
FIGURA 4-65 – TOPOLOGIA DE TESTES COM PABX FABRICANTE SIEMENS.....	61
FIGURA 4-66 – FOTOGRAFIA DO DISPOSITIVO INSTALADO EM AMBIENTE DE PRODUÇÃO..	61
FIGURA 4-67 – TELA COM DADOS GRAVADOS NO CARTÃO SD PARA FALHA LOS DETECTADA.....	62
FIGURA 4-68– TELA COM DADOS GRAVADOS PABX PARA FALHA LOS DETECTADA.....	62



FIGURA 4-69 – TELA COM DADOS GRAVADOS SD CARD PARA FALHA LOS NORMALIZADA.	62
FIGURA 4-70 – TELA COM DADOS GRAVADOS NO PABX PARA FALHA LOS NORMALIZADA.	63
FIGURA 4-71 – FOTOGRAFIA DO DISPOSITIVO INSTALADO EM AMBIENTE DE PRODUÇÃO..	63
FIGURA 4-72 – GRÁFICO QUANTITATIVO DE FALHAS AVALIADAS POR 72 HORAS.....	64
FIGURA 4-73 – GRÁFICO QUALITATIVO DAS FALHAS AVALIADAS POR 72 HORAS.....	64



LISTA DE TABELAS

TABELA 1 – PERÍODOS DE TEMPO TDM.....	19
TABELA 2 – CARACTERÍSTICAS ELÉTRICAS INTERFACE G.703.....	21
TABELA 3 – TABELA COM VARIÁVEIS PARA DETECÇÃO DE FALHAS.....	29
TABELA 4 – TABELA DE FUNCIONAMENTO RECEIVER DIFERENCIAL.....	30
TABELA 5 – ESTRUTURA DOS REGISTRADORES UTILIZADOS DS1302.....	38
TABELA 6 – ESTRUTURA DOS BYTES DE COMANDO SPI PARA CARTÕES SD.....	43
TABELA 7 – ESTRUTURA DOS BYTES DE COMANDO SPI PARA CARTÕES SD.....	45



LISTA DE ABREVIATURAS E SIGLAS

ITU: International Telecommunication Union

HDB3: High Density Bipolar of order 3

AMI: Alternate Mark Inversion

TDM: Time Division Modulation

SD Card: Secure Digital Card

SPI: Serial Peripheral Interface

PCM: Pulse Code Modulation

TS: Time Sample

DC: Direct Current

NOS: No of Signal

LOS: Loss of Sincronism

CRC: Cyclic Redundant Check

AIS: Alarm Indication Signal

RDI: Remote Defect Indication

FAS: Frame Alignment Signal

SNR: Signal to Noise Ratio

RISC: Reduced Instruction Set Computer



SUMÁRIO

1. INTRODUÇÃO.....	1
2. REFERENCIAL TEÓRICO.....	3
2.1. Linha de Transmissão de Telecomunicações.....	3
2.1.1 Descrição geral.....	3
2.1.2 Modelagem de uma linha de transmissão coaxial.....	4
2.1.3 Cálculo de impedância característica em cabos coaxiais.....	5
2.2. Dados e Sinais.....	6
2.2.1 Sinais Analógicos.....	7
2.2.2 Sinais Digitais.....	8
2.2.3 Conversão de Sinais Analógicos para Digitais.....	10
2.2.4 Multiplexação por Divisão de Tempo (TDM).....	17
2.3. Transmissão de Sinais Digitais.....	20
2.3.1 Interface a 2.048kbp/s – Recomendação ITU-T G.703.....	20
2.3.2 Tipos de Alarmes de anomalias em 2048kbps.....	23
2.4. Perdas na Transmissão.....	24
3. MATERIAIS E MÉTODOS.....	26
3.1. Descrição Geral do Sistema.....	26
3.2. Descrição dos Sistemas Eletroeletrônicos.....	29
3.2.1 Dispositivo para detecção de falhas em linhas G703.....	29
3.2.2 Dispositivo para geração da base de tempo.....	33
3.2.3 Dispositivo para armazenamento massivo das falhas.....	39
3.2.4 Microcontrolador PIC 18F4520.....	48
3.3. Descrição dos Sistemas de Informática e Computacionais.....	49
3.3.1 Software compilador.....	49
3.3.2 Software para gravação do PIC18F4520.....	49
3.3.3 Interface para visualização dos resultados.....	50
3.4. Fluxograma do software desenvolvido – Modo de Coleta.....	50
3.5. Fluxograma do software desenvolvido – Modo de manutenção via porta serial.....	53
4. APRESENTAÇÃO E DISCUSSÃO DOS RESULTADOS.....	55
4.1. Testes Preliminares.....	55
4.1.1 Simulações da Capacidade de Memória SD.....	55
4.1.2 Simulações de falhas de energia durante etapa de Gravação.....	56
4.1.3 Monitoração da precisão do relógio fornecido pelo RTC.....	57
4.1.4 Testes de compatibilidade com outros tipo de cartões SD.....	57
4.1.5 Testes no Osciloscópio.....	57
4.2. Metodologia dos Testes Integrados no Sistema Real.....	60
4.2.1 Testes Central PABX Fabricante Alcatel-Lucent.....	60
4.2.2 Testes Central PABX Fabricante Siemens.....	60
4.3. Comparações dos Resultados das Simulações, Modelos e o Sistema Real.....	61
4.3.1.1 Testes Central PABX Fabricante Alcatel-Lucent.....	61
4.3.2 Testes Central PABX Fabricante Siemens.....	63
5. CONSIDERAÇÕES FINAIS.....	65
5.1. Problemas Encontrados.....	65
5.2. Avaliação dos Objetivos Propostos.....	65
5.3. Avaliação dos Índices de Qualidade.....	65



5.4. Avaliação do Cronograma Previsto e Executado.....	66
5.5. Conclusões.....	66
5.6. Sugestões para Trabalhos Futuros.....	67
6. REFERÊNCIAS.....	68
OBRAS CONSULTADAS.....	69
APÊNDICE A – CÓDIGO FONTE EM LINGUAGEM C.....	70
APÊNDICE B – DIAGRAMA ESQUEMÁTICO DO DISPOSITIVO.....	1
APÊNDICE C – FOTO DO PROTÓTIPO DESENVOLVIDO.....	2

1. INTRODUÇÃO

A sociedade vive a chamada “Era da Informação”, onde todos os sistemas de telecomunicações exercem um papel muito abrangente neste contexto. O 1º sistema de telefonia foi apresentado oficialmente pelo inventor escocês Alexander Graham Bell no ano de 1876. Desde esta data houve muitos avanços e pesquisas, de forma a automatizar a comunicação, como também oferecer mobilidade de comunicação para as pessoas. Muitos estudos desde então foram também voltados a aumentar a taxa de comunicação por unidade de tempo, surgindo novas tecnologias de telecomunicações.

Dentro do contexto de novas tecnologias, é fundamental para as pessoas terem uma comunicação confiável e com alta qualidade. Não basta pessoas e empresas possuírem dispositivos móveis de última geração, centrais telefônicas baseadas em tecnologia de ponta, recursos de vídeo-conferência, se não possuem uma infra-estrutura de telecomunicações bem dimensionada e confiável para tornar a comunicação realmente eficiente. Esta infra-estrutura geralmente está baseada em chamados meios de transmissão, que podem abranger estruturas de fibra óptica, enlaces de rádio frequência, cabos coaxiais e pares trançados, entre outros

Os consumidores de serviços de telecomunicações, muitas vezes não conseguem antecipar uma falha grave que possa ocorrer em seu meio de telecomunicações. Muitas vezes uma falha grave ocorre precedida de falhas mais simples e de duração muito curta. Geralmente os consumidores dos serviços oferecidos pelas operadoras de telecomunicações não possuem uma base de dados confiável para poder avaliar e tomar providências com base no histórico de falhas ocorridas.

O presente trabalho apresenta um protótipo eletrônico de gerenciamento, que auxilia equipes responsáveis pela operação e manutenção de sistemas de telecomunicações, a levantar de forma eficiente as reais causas da degradação na qualidade de serviço em um sistema de telefonia baseado em centrais telefônicas privadas do tipo PABX. O protótipo proposto possui eficiência máxima para análise

quando as falhas ocorrem de forma aleatória, pois o mesmo registra o tipo de falha detectada, e também é capaz de armazenar a data e hora em que houve esse tipo de falha, facilitando a indexação para a extração de relatórios históricos.

Além do exposto, quando um consumidor opta por mudar seu fornecedor de serviços de telecomunicações, praticamente é impossível realizar uma comparação efetiva na melhoria na qualidade dos serviços, devido à ausência dos dados históricos armazenados, e de uma forma segura e indexada.

Os principais fabricantes de centrais telefônicas do tipo PABX, muitas vezes não possuem módulos específicos para esta função de supervisão de falhas, ou muitas vezes esses recursos são bem limitados a informar somente o horário da falha, e não a duração, e com capacidade de memória restrita para gravação de períodos históricos mais amplos, bem como geração de estatísticas mais detalhadas sobre qualidade de serviço.

O projeto possui como objetivo o desenvolvimento de um dispositivo eletrônico capaz de avaliar permanentemente uma linha digital de telecomunicações do tipo E1, com tipo de codificação G703 (ITU-T).

Como objetivo específico o dispositivo deverá, em caso de alguma anomalia ou falha no sistema, registrar o tipo de falha ocorrida, bem como indicar um detalhamento de data e hora que ocorreu a falha, salvando em um local seguro, neste caso uma memória flash do tipo SD Card, para consultas e diagnósticos futuros. O dispositivo deverá executar todas as tarefas anteriormente descritas bem como deverá ser um dispositivo de alta confiabilidade, não podendo sofrer interferências externas, nem podendo ter sua base de tempo alterada por fatores externos ao ambiente. Além disso, o dispositivo não poderá interferir no funcionamento de um sistema de telecomunicações real em operação. O protocolo de comunicação entre os periféricos que contemplam o dispositivo será o SPI (*Serial Protocol Interface*).

O principal resultado previsto no funcionamento do dispositivo é uma resolução de um segundo para as falhas, devendo avaliar pelo menos dois tipos de falhas ocorridas no sistema de telecomunicações, sendo que os dados armazenados possam ser extraídos de forma fácil e interativa pelo operador.

2. REFERENCIAL TEÓRICO

2.1. *Linha de Transmissão de Telecomunicações*

2.1.1 Descrição geral

Toda informação para ser transmitida de um ponto ao outro através de uma distância, necessita de uma linha de transmissão, também chamada de um meio de transmissão. Para haver uma transmissão efetiva, a informação original necessita ser convertida em sinais eletromagnéticos.

Uma linha de transmissão pode ser definida como um dispositivo capaz de transmitir ou conduzir energia de um ponto a outro. Geralmente se deseja que, numa linha de transmissão, a energia seja transportada com a máxima eficiência, sendo as perdas em calor ou radiação as menores possíveis. (KRAUS, 1986, p.454)

Kraus (1986) destaca que linhas de transmissão podem ser divididas em 2 grandes tipos, quanto ao tipo de propagação eletromagnética:

- Propagação no modo TEM, onde o campo elétrico (E) e magnético (H) são inteiramente transversais, ou seja, utilizada em meios que utilizam no mínimo 2 condutores

- Propagação em modo superior, onde o campo elétrico (E) e magnético (H) possuem componentes na direção de propagação, sendo condutores dos tipos guias de onda, geralmente utilizados em sistemas de transmissão via ondas eletromagnéticas.

O objeto desse estudo será direcionado à propagação no modo TEM, através de cabos coaxiais, utilizados amplamente em linhas digitais de telecomunicações *multiplexadas*.

Conforme Forouzan (2008), os cabos coaxiais possuem uma atenuação do sinal muito mais elevada, se comparada a cabos metálicos trançados, em outras palavras, embora o cabo coaxial tenha uma largura de banda muito maior, o sinal enfraquece rapidamente, exigindo assim o uso freqüente de repetidores nas linhas de transmissões.

2.1.2 Modelagem de uma linha de transmissão coaxial

Conforme Kraus (1986), realizando uma modelagem matemática, um cabo coaxial pode ser expresso por 2 condutores paralelos e uniformes conforme figura abaixo:

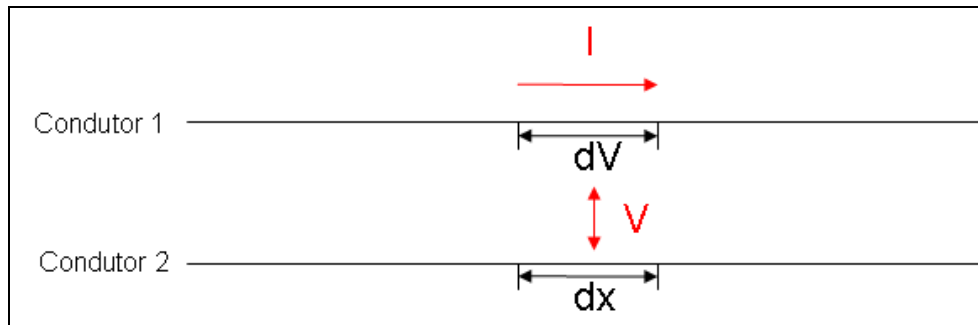


Figura 2-1 – Linha de transmissão paralela

Fonte: Elaborada pelo autor

Efetuada uma analogia a um circuito elétrico, podemos representar uma linha de transmissão no seguinte equivalente:

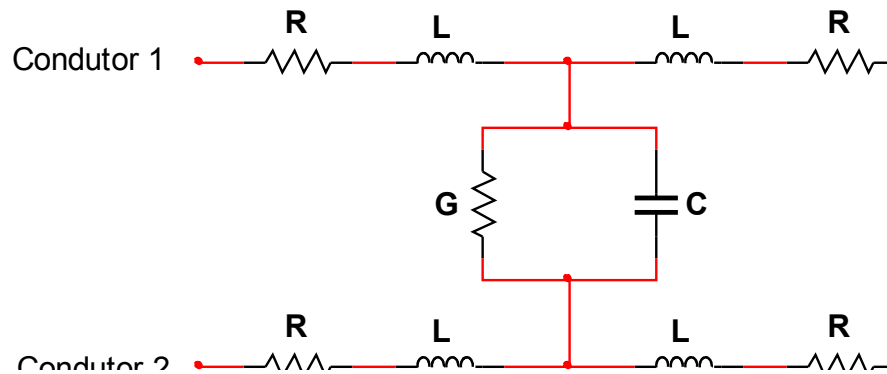


Figura 2-2 – Analogia de uma linha de transmissão com circuito elétrico

Fonte: Elaborada pelo autor

A impedância série desta estrutura pode ser modelada pela seguinte equação:

$$Z_{serie} = R + j\omega L = R + jX \therefore [\Omega/m]$$

A admitância em paralelo desta estrutura pode ser modelada pela seguinte equação:

$$Y_{paralelo} = G + j\omega C = G + jB \therefore [\sigma/m]$$

Portanto a impedância característica de uma linha de transmissão pode ser aproximada por:

$$Z_o = \sqrt{\frac{R + j\omega L}{G + j\omega C}}$$

Quando o R e o G possuem valores próximos a zero ou muito menores quando comparados a ωL e ωC , ou quando a frequência tende ao infinito, a equação pode ser reduzida a:

$$Z_o = \sqrt{\frac{L}{C}}$$

Onde:

Z_o = Impedância característica da linha [Ω]

L = Indutância em série [H/m]

C = Capacitância em paralelo [C/m]

2.1.3 Cálculo de impedância característica em cabos coaxiais

Especificamente para o cabo coaxial, o parâmetro de impedância característica, pode ser aproximado por:

$$Z_o = \frac{138}{\sqrt{\epsilon_r}} \log \frac{b}{a}$$

Onde:

ϵ_r = Permissividade relativa do meio (dielétrico)

a= Raio do condutor

b=Raio da malha externa

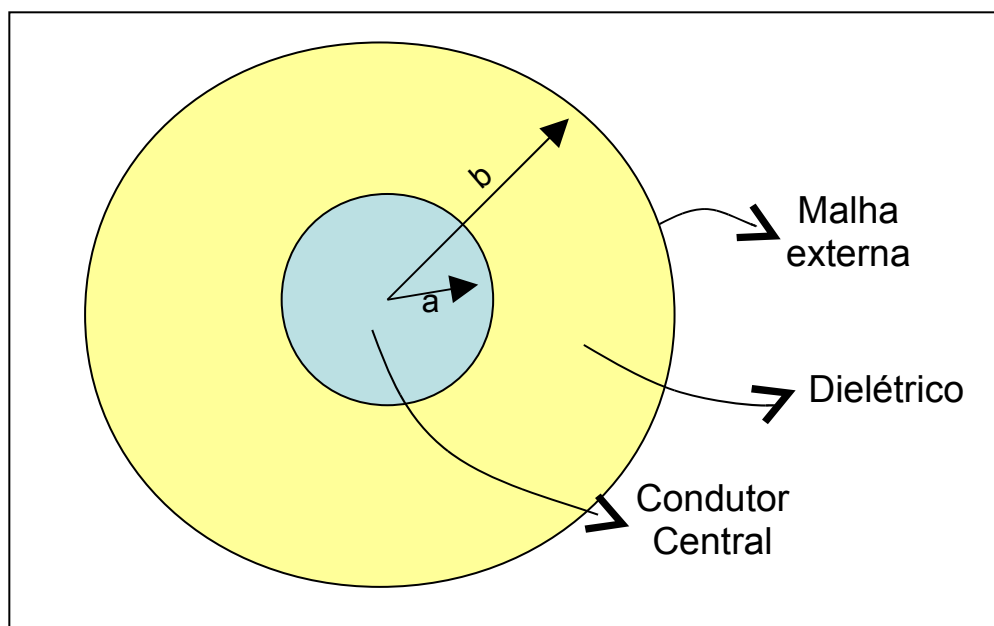


Figura 2-3 – Estrutura transversal de um cabo coaxial

Fonte: Elaborada pelo autor

2.2. *Dados e Sinais*

Para haver um estabelecimento de comunicação em uma linha de transmissão, é necessário haver um transporte dos dados na forma eletromagnética, ou seja, um sinal.

Geralmente, os dados enviados de um local para outro, não se encontram em um formato que pode ser transmitido diretamente na linha de transmissão, para serem transmitidos os dados, ou melhor, a informação precisa ser transformada em sinais eletromagnéticos. Os dados, assim como os sinais, podem ser em sua forma, analógicos ou digitais.

[...]Dados analógicos são contínuos e assumem valores contínuos. Os dados digitais possuem estados discretos e assumem valores discretos. [...]Sinais analógicos podem ter um número infinito de valores em um período de tempo; os sinais digitais podem ter apenas um número limitado de valores. (FOROUZAN, 2007, p.58)

Nos sistemas de telecomunicações, comumente são utilizados sinais analógicos periódicos e sinais digitais não periódicos

2.2.1 Sinais Analógicos

Conforme Forouzan (2008), os sinais analógicos periódicos podem ser classificados como simples ou compostos. Um sinal analógico simples, uma onda senoidal, não pode ser decomposta em sinais mais simples.

Uma onda senoidal pode ser representada por 3 parâmetros, onde estes parâmetros descrevem totalmente uma onda senoidal. Os parâmetros são: **Amplitude Máxima, Frequência e Fase.**

A amplitude máxima de um sinal é o valor absoluto da máxima intensidade, proporcional à energia que ela transporta no caso de sinais elétricos este parâmetro é medido em *Volts*

A frequência (f) corresponde ao número de ciclos realizados por unidade de tempo. O período (T) é o inverso da frequência, sendo medido em segundos.

O termo fase descreve uma forma de onda relativa ao instante de tempo zero. A fase está ligada a deslocamento no tempo.

Portanto, uma onda senoidal pode ser caracterizada, no domínio tempo por:

$$s(t) = A \cdot \sin(\omega t + \phi)$$

Onde:

A = Amplitude do sinal

ω = Frequência angular ($2\pi f$)

ϕ = Ângulo de fase

Uma comunicação de dados com apenas um sinal analógico simples não é possível de ser estabelecida, é necessário trafegar em uma linha de transmissão um sinal analógico composto, que é formado de várias ondas senoidais simples.

No início do século XX, o matemático francês Jean-Baptiste Fourier demonstrou que um sinal composto é uma combinação de ondas senoidais simples, com diferentes frequências, amplitudes e fases. Ele provou que um sinal periódico composto, com frequência f , pode ser decomposto em uma série de funções senoidais e co-senoidais, na qual cada função é uma harmônica inteira da frequência fundamental f do sinal composto.

$$s(t) = A_1 \cdot \sin(2\pi f_1 t + \phi_1) + A_2 \cdot \sin(2\pi f_2 t + \phi_2) + A_3 \cdot \sin(2\pi f_3 t + \phi_3) + \dots$$

$$s(t) = A_0 + \sum_{n=1}^{\infty} A_n \cdot \sin(2\pi n f t) + \sum_{n=1}^{\infty} B_n \cos(2\pi n f t)$$

Onde:

$$A_0 = \frac{1}{T} \int_0^T s(t).d(t)$$

$$A_n = \frac{2}{T} \int_0^T s(t). \cos(2\pi nft).d(t)$$

$$B_n = \frac{2}{T} \int_0^T s(t). \sin(2\pi nft).d(t)$$

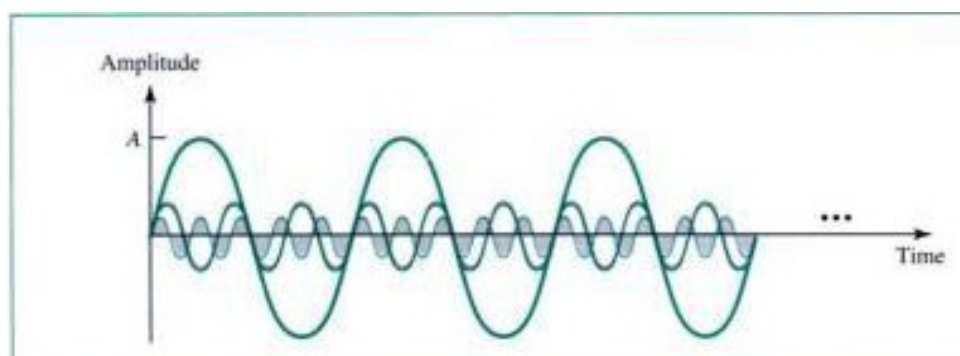


Figura 2-4 – Sinal analógico com respectivas harmônicas
Fonte: (FOROUZAN, 2004, p.58)

A onda com maior amplitude na figura acima é denominada a frequência fundamental ou primeira harmônica. Esta onda possui frequência f , ou seja, é a mesma frequência do sinal composto. Nota-se também que a amplitude das harmônicas tende a zero à medida que a frequência tende para infinito.

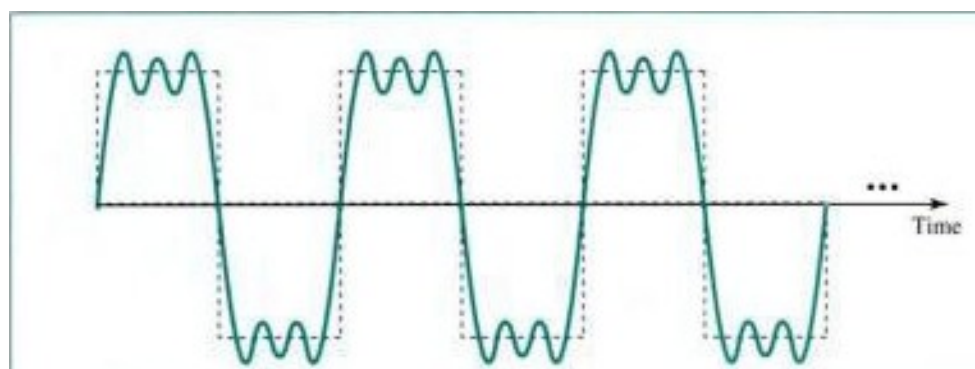


Figura 2-5 – Sinal analógico com respectivas harmônicas
Fonte: (FOROUZAN, 2004, p.58)

2.2.2 Sinais Digitais

As informações podem ser também representadas por sinais digitais. Os sinais digitais podem ter mais que 2 níveis lógicos (0 e 1), fazendo assim com que em uma transmissão possam ser transmitidos mais do que 1 bit por nível.

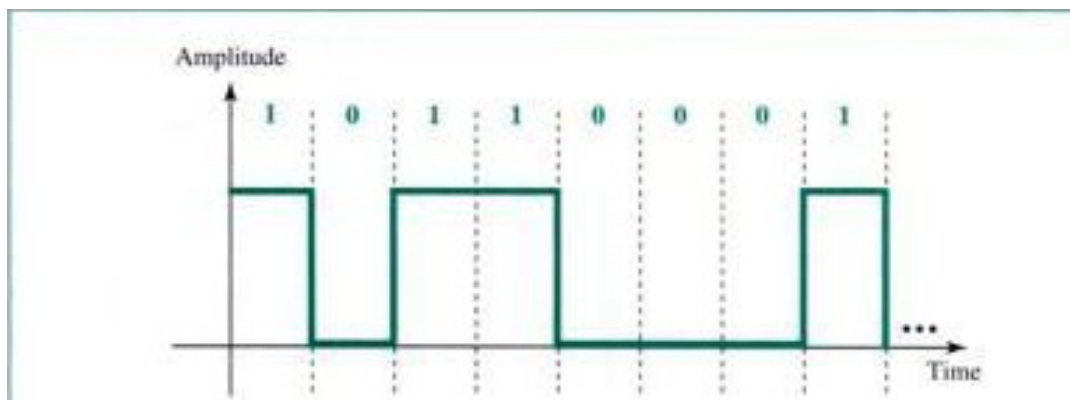


Figura 2-6 – Representação de um sinal digital

Fonte: (FOROUZAN, 2004, p.62)

Como a maioria dos sinais digitais não é periódica, conseqüentemente o parâmetro de frequência do sinal não é relevante, sendo este substituído por taxa de transferência. A taxa de transferência é um parâmetro que expressa a quantidade de bits enviados por unidade de tempo, sendo este tempo geralmente 1 segundo, ficando *Bits por Segundo*, ou bps.

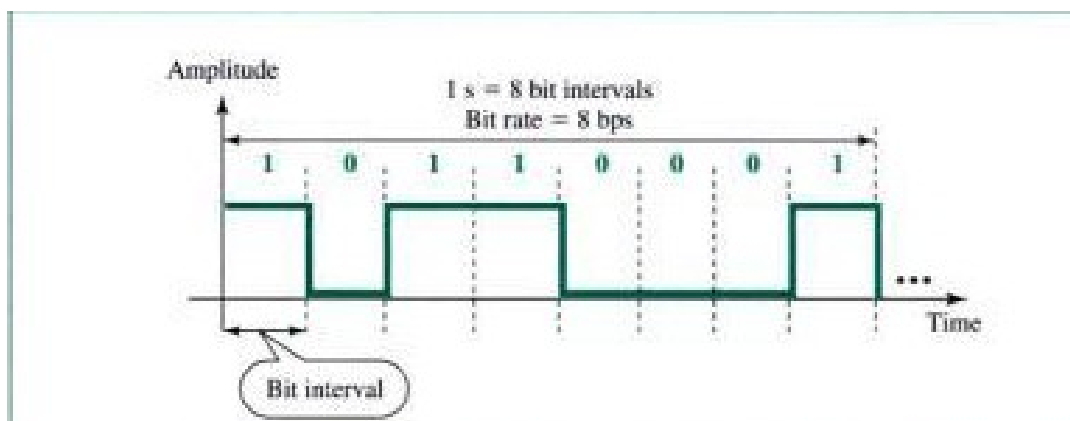


Figura 2-7– Representação da taxa de bits por unidade de tempo

Fonte: (FOROUZAN, 2004, p.63)

Segundo Forouzan (2008), um sinal digital, periódico ou não, é um sinal analógico composto por frequências entre zero e infinito.

Para sistemas de comunicação de dados, os sinais digitais não periódicos são os mais utilizados. Para a transmissão de sinais digitais, existem 2 grandes tipos de transmissões: transmissão banda larga e transmissão banda base.

A transmissão do tipo banda base, o sinal digital é enviado bit a bit por um canal passa-baixa, onde a largura de banda começa na frequência 0. Este tipo de transmissão não requer modulação, pois geralmente é empregado em pequenas distâncias.

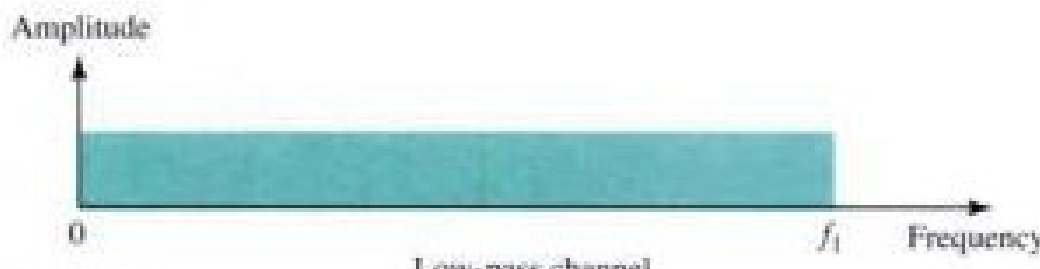


Figura 2-8 – Transmissão tipo banda base
Fonte: (FOROUZAN, 2004, p.66)

A transmissão do tipo banda larga, significa transformar um sinal digital em sinal analógico para a transmissão, através de um processo chamado de modulação. O receptor no outro lado da linha deverá realizar uma demodulação do sinal para obter os dados originais (informação).

Para a comunicação de dados em canais E1, ao invés de ser transformada a informação em sinal analógico para a transmissão, a informação é transformada em sinais digitais, através de um processo denominado codificação.

2.2.3 Conversão de Sinais Analógicos para Digitais

Existem várias técnicas de conversão de sinais analógicos para sinais digitais. A técnica mais comum utilizada em telecomunicações para a conversão de sinais analógicos para dados digitais é a PCM (modulação por código de pulsos). A modulação por código de pulsos possui 3 etapas distintas para a completa digitalização, que são: Amostragem, Quantização e Codificação

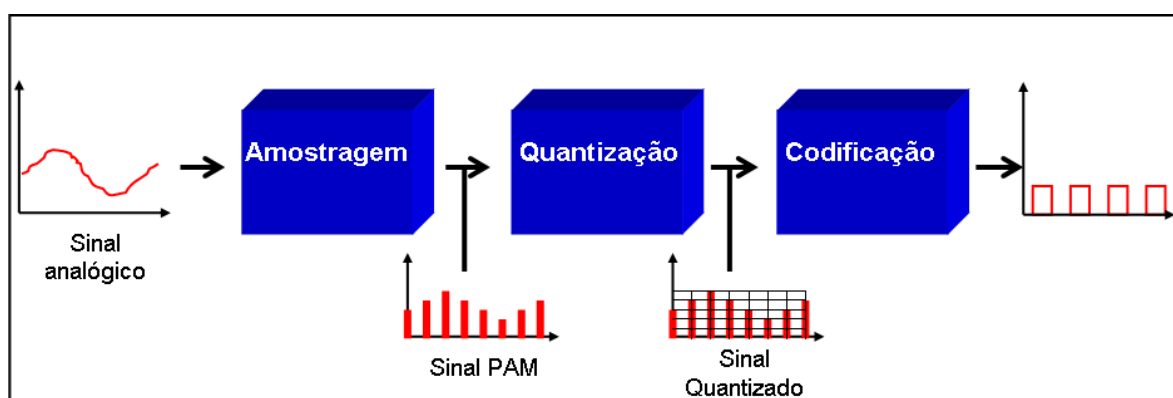


Figura 2-9– Etapas básicas do processo de conversão de sinais
Fonte: Elaborada pelo autor

O sinal analógico de entrada no dispositivo que realiza a conversão analógica digital é um sinal na faixa 300 Hz a 3400 Hz. Segundo Pines (1978), esta faixa de frequências para fonia foi definida com base em um compromisso entre

parâmetros chamados de intelegibilidade e energia da voz. Intelegibilidade é o percentual de palavras perfeitamente reconhecidas em uma conversação. A faixa de frequência de áudio localizada entre 300 Hz a 3400, utilizada nas comunicações telefônicas, garante 85% de intelegibilidade e 68% de energia de voz recebida pelo ouvinte.

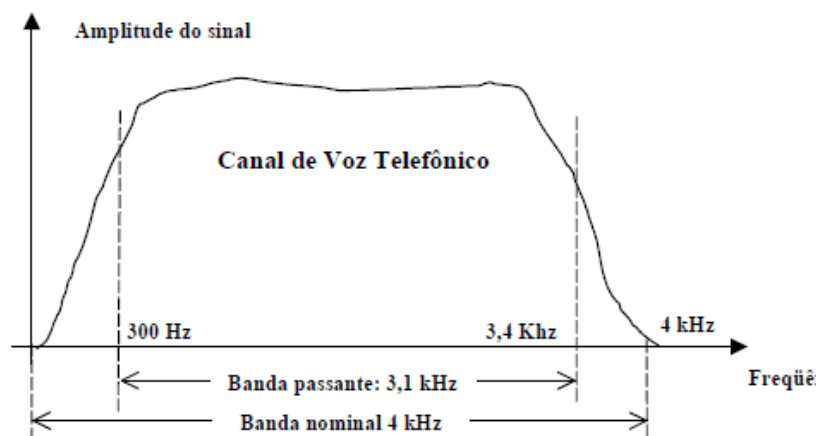


Figura 2-10 – Sinal analógico em um canal telefônico
Fonte: (ROCHOL, 1999, p.96)

2.2.3.1 Etapa de Amostragem

Na etapa de amostragem, o sinal analógico de entrada é amostrado a cada T_s segundos, onde T_s é o período de amostragem. Inversamente a este período temos a frequência de amostragem, ou taxa de amostragem. Porém temos restrições quanto a valores de T_s na prática, pois este parâmetro deve ter um valor estipulado máximo enquanto que a frequência de amostragem deve ter um valor mínimo, valores estes sempre baseados na frequência da informação original analógica.

De acordo com o teorema de Nyquist, a taxa de amostragem deve ser pelo menos o dobro da frequência mais elevada contida no sinal. (FOROUZAN, 2007, p.122)

Como a largura de banda de fonia varia de 300 a 3400 Hz, a largura de banda é:

$$B = fh - fl$$

Onde:

fh = Frequência Alta (Frequência mais elevada)

fl = Frequência Baixa

Logo:

$$B = 3400 - 300 = 3100 \text{ Hz}$$

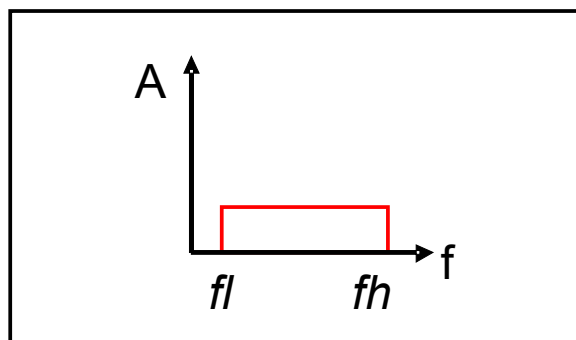


Figura 2-11– Representação da largura de banda
Fonte: Elaborada pelo autor

Seguindo aos critérios de Nyquist, e através do padrão ITU-T, temos para os canais de voz telefônicos uma taxa de amostragem de 8000Hz ou 8kHz. Este valor atende sobremaneira o critério, e foi aumentado e arredondado para facilitar o desenvolvimento eletrônico nos dispositivos associados.

$$\begin{aligned} f_0 &\geq 2 \times fh \\ 8000 &\geq 2 \times 3400 \\ 8000 \text{ Hz} &\geq 6400 \text{ Hz} \end{aligned}$$

O resultado da amostragem em um canal de comunicação é uma série de pulsos com valores de amplitude localizados entre as máximas e mínimas amplitudes do sinal de áudio original, portanto o conjunto dessas amplitudes pode ser infinito. O tempo de amostragem, para cada canal básico digital de telefonia, corresponde a 2,1μs em uma frequência de amostragem de 8kHz. A frequência de amostragem gera um período T de 125 μs.

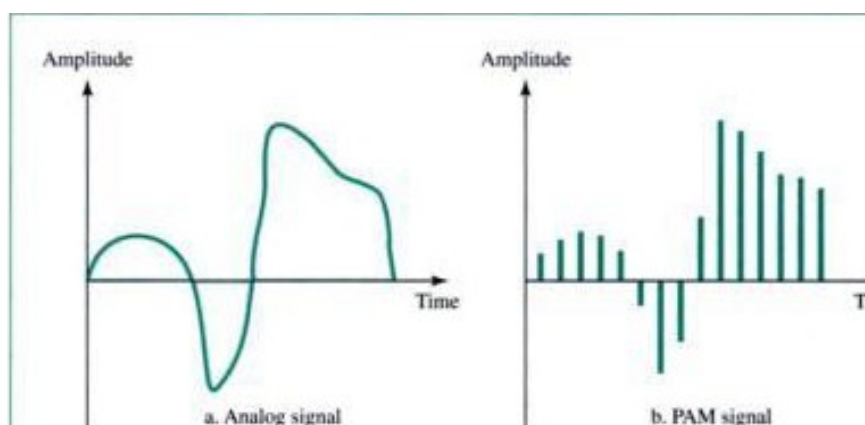


Figura 2-12– Representação do sinal analógico após etapa de amostragem (PAM)
Fonte: (FOROUZAN, 2004, p.99)

2.2.3.2 Etapa de Quantização

O objetivo da quantização é deixar estes valores de amplitudes finitas. Para a etapa de quantização, o sinal é dividido em intervalos onde é realizado um arredondamento para o próximo valor acima ou para baixo.

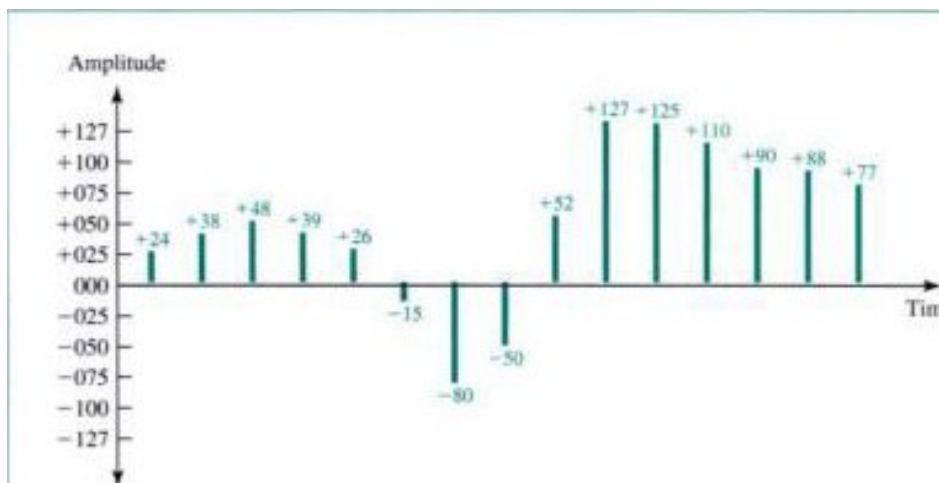


Figura 2-13– Representação do sinal PAM após a quantização
Fonte: (FOROUZAN, 2004, p.100)

2.2.3.3 Etapa de Codificação

A última etapa na PCM (Modulação por Código de Pulsos) é a codificação, que consiste em gerar códigos binários para cada valor de amplitude. Para sistemas de voz telefônica, são gerados códigos de 8bits para cada nível de sinal, totalizando um total de 256 níveis inteiros. O parâmetro mais importante a ser considerado na etapa de codificação é a taxa de bits, que pode ser calculada pela seguinte fórmula:

$$Tx_{bits} = f_{amostr} \times bits$$

Para o caso de linhas digitais de telecomunicações, temos :

$$f_{amostr} = 8000 \text{ kHz}$$

$$bits = 8$$

Portanto a Tx_{bits} para um canal digital básico de telecomunicações é de 64000 bps ou 64kbps.

Segundo Forouzan (2008), a codificação de linha é um processo de conversão dos dados na forma digital-digital. Todos os tipos de dados, ou informações, são armazenados nos dispositivos digitais (computadores, memórias), como uma sequência de bits. A codificação de linha é uma técnica largamente

utilizada nos sistemas de telecomunicações para converter uma sequência de bits em um sinal digital adequado para ser transmitido em um meio de transmissão.

No emissor, a sequência dos dados digitais é codificada em um sinal digital; e no receptor a sequência de dados digitais é recriada, reconvertendo-se a sequência de dados digitais original.

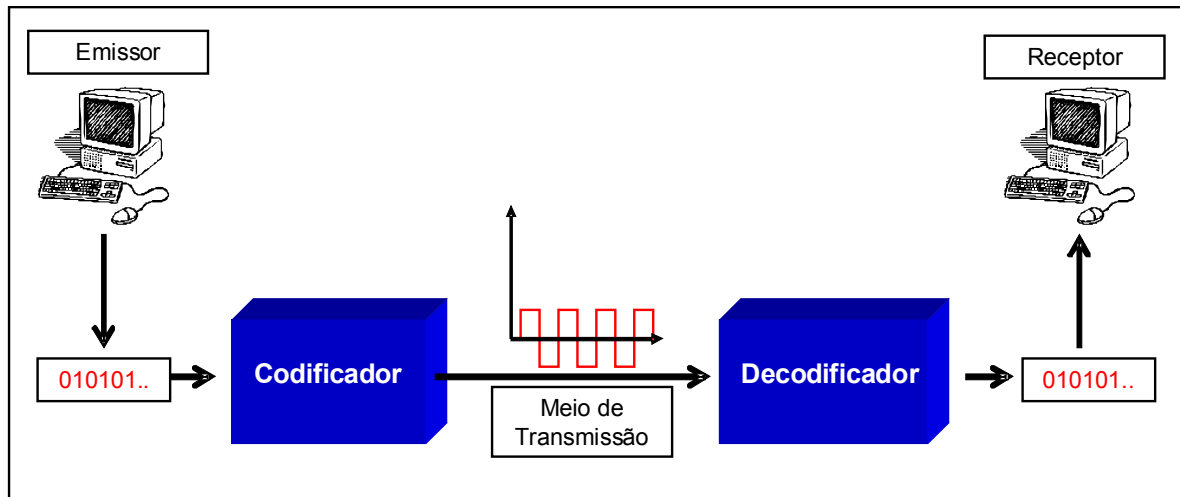


Figura 2-14 – Codificação de linha digital
Fonte: Elaborada pelo autor

Uma importante característica na transmissão de dados é a relação entre taxa de dados e taxa de sinal, representada pela letra r . Taxa de dados define o número de elementos de dados (bits) enviados na forma de 1. A unidade é bits por segundo (bps). A taxa de sinal é o número de elementos de sinal enviados na forma de 1. A unidade é baud.

Outra importante característica é a largura de banda de um sinal, conforme citado anteriormente, a largura de banda para sinais digitais é infinita, porém na prática não é exatamente isto que ocorre, pois muitos dos componentes do sinal têm amplitudes tão pequenas, que podem ser desconsiderados. O cálculo da largura de banda, para um sinal de 64kbps, pode ser calculado como:

$$B = c \times N \times \frac{1}{r}$$
$$B = \frac{1}{2} \times 64000 \times \frac{1}{1}$$
$$B = 32000 \text{ kHz}$$

Onde:

B = Largura de banda do sinal efetivo

C = Fator de caso, no caso $\frac{1}{2}$ para cálculo de taxa média de dados

N =Taxa de dados (bps)

r =Relação entre elemento de dados (informação) x elemento de sinal.

Para o caso de transmissões com codificação AMI / HDB-3 esta relação r possui valor 1, ou seja, para cada elemento de informação, necessitamos de um elemento de sinal.

Para a obtenção de alta qualidade na codificação/ decodificação dos sinais de informação, os métodos de codificação devem atender outras importantes características, tais como:

- Evitar o afastamento do sinal em relação a referência inicial – Causada por uma longa string de 0s ou 1s, que induz um desvio na referência inicial o sinal

- Supressão da componente DC do sinal – Ocorre quando um nível de sinal digital for constante por muito tempo, acabam aparecendo freqüências muito baixas, em torno de zero. Se o sistema possuir um casador de impedâncias utilizando transformadores, este tipo de sinal será perdido.

- Auto-sincronização – Na própria transmissão de dados, são enviados bits distintos com o objetivo de sincronizar os relógios (*clock*) entre o emissor e o receptor. Estes bits distintos de informação, geralmente são transições no sinal que alertam o receptor sobre o início, meio ou mesmo o final de um bloco de informação.

- Detecção de erros embutidos – É desejável para uma codificação possuir recursos de detecção de erros embutidos no código.

- Imunidade a ruídos ou interferências

Existem atualmente vários tipos de codificações para linhas digitais, cada qual com suas funcionalidades, bem como vantagens e desvantagens. A codificação HDB3 (Alta Densidade Bipolar com 3 Zeros) é muito utilizada no mundo, sendo apenas nos EUA que seu uso é muito restrito. A codificação HDB3 é uma codificação oriunda da codificação AMI.

A codificação AMI (Inversão de Marca Alternada) é uma codificação do tipo bipolar, ou seja, além do nível zero de sinal utiliza o positivo e o negativo. O maior ganho de potência neste tipo de codificação ocorre na freqüência $N/2$, ou seja, na metade da freqüência equivalente à taxa de dados

Na codificação AMI, quando o sinal apresenta uma grande seqüência de bits com nível lógico igual a 1, o mesmo sempre estará alternando entre as polaridades positivas e negativas da linha, fazendo assim com que a freqüência nunca chegue a tender a zero (Sinal DC). Portanto, a grande vantagem desse tipo de codificação é a supressão da componente DC do sinal, que é um grande problema para transmissões em longas distâncias. A grande desvantagem da codificação AMI é quando na linha se tem uma grande seqüência de bits 0s, pois há uma grande dificuldade de se conseguir sincronizar o emissor e o receptor no sistema.

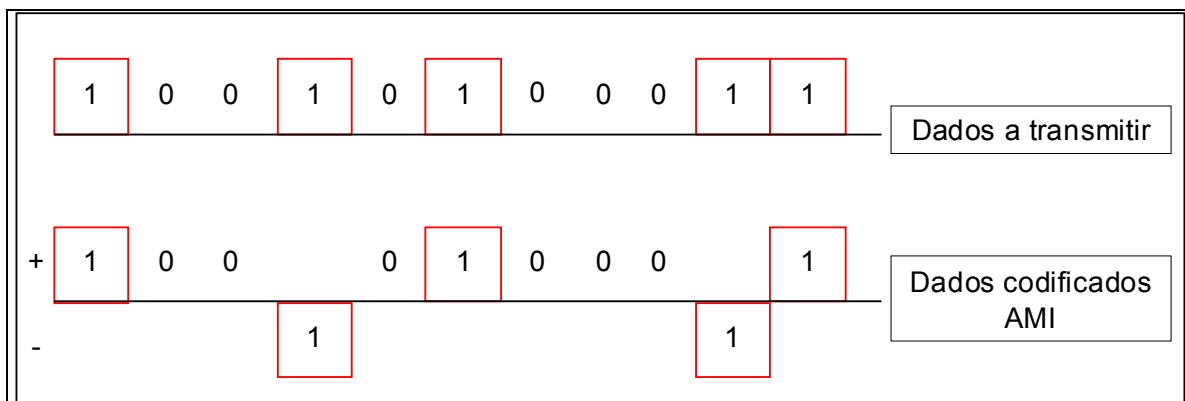


Figura 2-15 – Codificação AMI

Fonte: Elaborada pelo autor

A codificação HDB3 resolve justamente este problema de sincronizar o emissor e receptor no sistema de transmissão. Para gerar uma codificação HDB3 é realizada uma técnica de mistura de sinais, utilizando como base a codificação AMI. Na codificação HDB3 quatro voltagens consecutivas de nível 0 são substituídas por uma seqüência de **000V** ou **B00V**, onde a letra V significa violação da Regra AMI e B significa bipolar.

Se o número de pulsos não-zero (positivo ou negativo) após a última substituição for ímpar, o padrão de substituição será 000V, o que torna o número total de pulsos não-zero par. Se o número de pulsos não zero (positivo ou negativo) após a última substituição for par, o padrão de substituição será B00V, o que torna

o número total de pulsos não-zero par. O fato de usar 2 tipos de substituição, B00V ou 000V, faz com que sempre se tenha um número par de pulsos não-zero após cada substituição. (P. S. 1978)

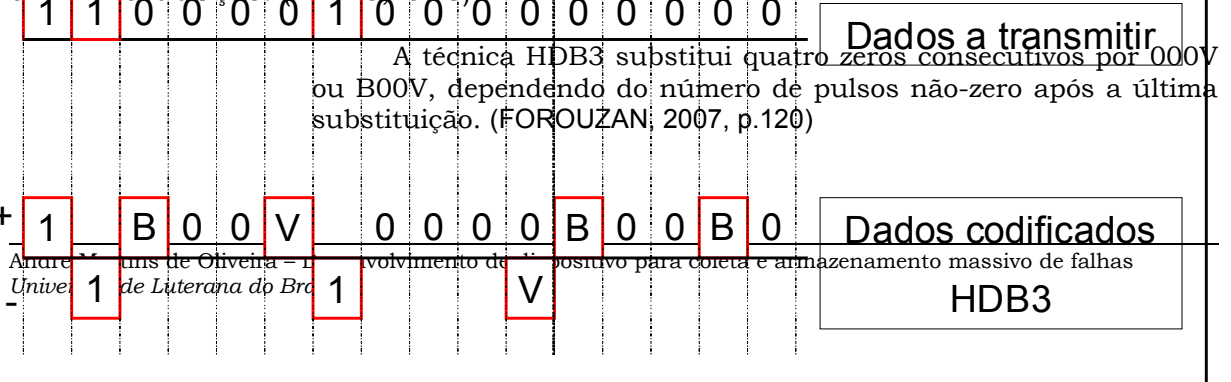


Figura 2-16 – Codificação de linha digital
Fonte: Elaborada pelo autor

2.2.4 Multiplexação por Divisão de Tempo (TDM)

Conforme Pines (1978), a multiplexação é uma técnica que possibilita a transmissão de vários canais de comunicação em um mesmo meio de transmissão.

A figura 2-17 apresenta de uma forma simplificada o processo de multiplexação por divisão de tempo, onde através do mesmo meio de transmissão, são transmitidas as informações de 4 computadores diferentes, porém em tempos diferentes.

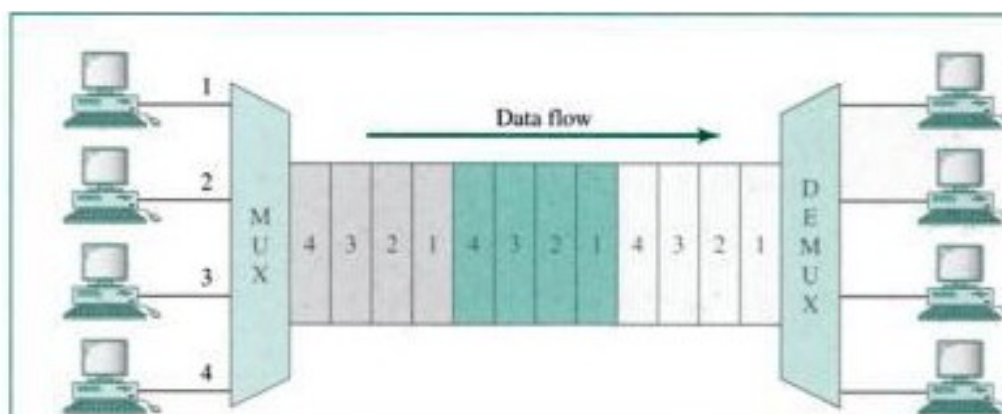


Figura 2-17- Representação do processo de multiplexação
Fonte: (FOROUZAN, 2004, p.157)

A multiplexação do tipo TDM (Time Division Multiplex) é atualmente a mais utilizada, porém existem outros tipos, bem menos empregados.

Conforme descrito anteriormente, os tempos de duração de cada amostragem possuem duração de $2,1\mu\text{s}$ para cada canal digital com taxa de transmissão de 64kbps. Cada amostragem no sinal está localizada dentro de uma janela de tempo, com duração total de $3,9\mu\text{s}$. Fazendo uma análise, verifica-se que este tempo é muito pequeno se comparado a um período T_s de $125\mu\text{s}$. Este período T_s é referente a uma frequência de amostragem de 8kHz de um sinal de áudio analógico, ou seja, o período de amostragem de um sinal de áudio teria uma ociosidade de $121,9\mu\text{s}$ por ciclo de amostragem. A técnica de multiplexação justamente racionaliza o uso deste tempo ocioso, pois aproveita para abrir outras janelas destinadas a transmissão de outros sinais amostrados. Esta abertura de janelas é que permite a transmissão de diferentes sinais através de um único meio de transmissão. O período T_s da frequência de amostragem corresponde a 1 quadro, sendo que cada quadro possui várias janelas, conforme a quantidade de canais utilizada seguindo a padronização do sistema vigente. No Brasil é utilizada a estrutura TDM-PCM européia, sendo 32 canais para cada quadro.

Outra característica relevante na estrutura TDM, são as informações de sincronismo e alarmes. Essa característica é baseada em uma estrutura chamada de multiquadro, onde no padrão Europeu utilizado no Brasil, o multiquadro possui 16 quadros, totalizando um tempo de 2 ms para cada multiquadro. A partir desta estrutura que é formado um canal E1, que possui 32 canais com taxa de transmissão de 64kbps cada, totalizando uma taxa de transmissão de 2,048Mbps.

EXEMPLO DE ESTRUTURA TDM UTILIZADA NO BRASIL
Os canais são numerados de 0 a 31, sendo 30 destes canais utilizados efetivamente para transporte de dados (informação). Os canais 0 e 16 são utilizados para sincronismo entre o emissor e o receptor. (PINES,1978)

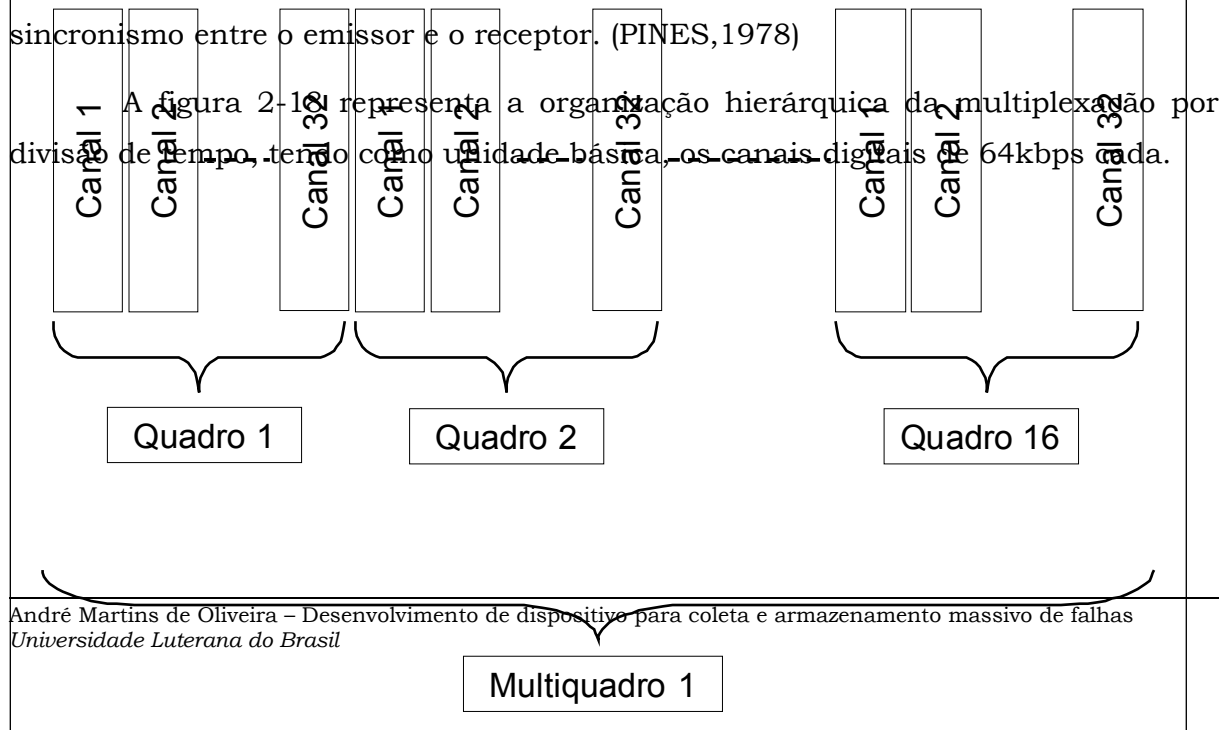


Figura 2-18– Estrutura TDM utilizada no Brasil
Fonte: Elaborada pelo autor

PERÍODO DE TEMPO DE CADA COMPONENTE DA MULTIPLEXAÇÃO TDM	
T Bit	488ns
T Amostragem	2,1 μ s
T Canal ou Janela ou <i>Time Slot</i> (8bits)	3,91 μ s
T Quadro (32 janelas)	125 μ s
T Multiquadro (16 quadros)	2 ms

Tabela 1– Períodos de tempo TDM
Fonte: Elaborada pelo autor

Pines (1978), destaca ainda, que para a obtenção da multiplexação por divisão de tempo, é necessário haver um sincronismo entre os seletores do emissor e do receptor, que devem obedecer as seguintes características de:

- a) Perfeito sincronismo de frequência e fase entre os quadros, característica obtida a partir de um sinal de alinhamento de multiquadro, disponível nas janelas do canal 16.
- b) As janelas de cada quadro devem ser exploradas na mesma ordem, característica obtida a partir de um sinal de alinhamento de quadro, obtida nas janelas do canal 0.

- c) Sincronismo entre os osciladores do emissor e receptor, também chamados de relógios de PCM, característica obtida através de uma técnica chamada de recuperação do sinal de relógio.

A figura 2-19 representa a estrutura de sincronismo comum entre o transmissor e o receptor para um sistema com multiplexação por divisão de tempo.

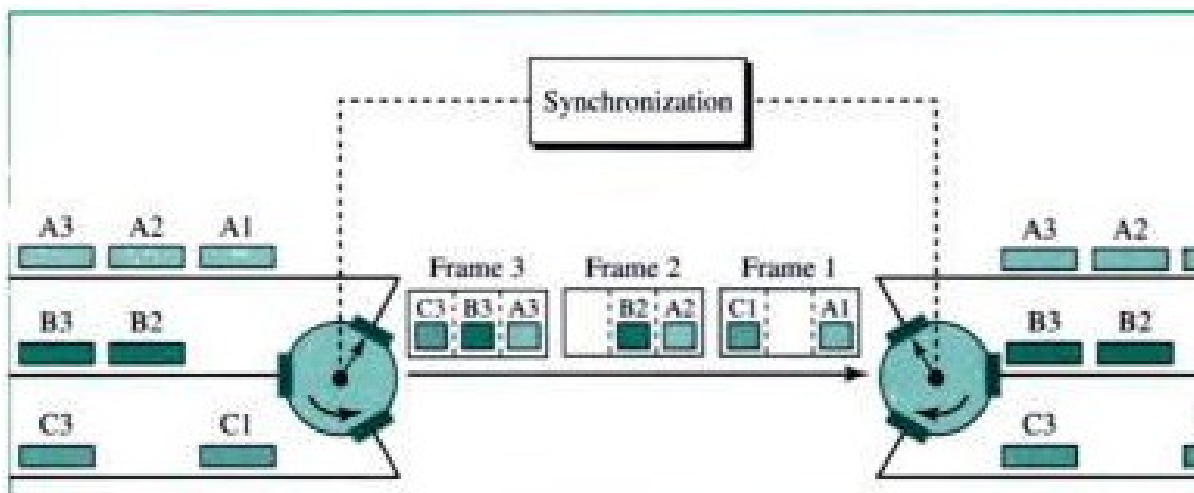


Figura 2-19– Representação do processo de sincronismo
Fonte: (FOROUZAN, 2004, p.157)

2.3. Transmissão de Sinais Digitais

2.3.1 Interface a 2.048kbp/s – Recomendação ITU-T G.703

O ITU-T é uma área da ITU (União Internacional de Telecomunicações), responsável por coordenar padronizações relacionadas a telecomunicações. O ITU-T atualmente é uma agência intergovernamental que congrega mais de 700 organizações públicas e privadas em 191 países. O ITU-T cria recomendações que, após aprovadas pelos membros, são empregadas como referência para o desenvolvimento de soluções tecnológicas envolvendo telecomunicações e redes.

(Fonte: <http://www.itu.int> Acesso em 10/10/10)

O presente projeto analisa uma linha de transmissão digital, padrão ITU-T G.703 com codificação HDB3, utilizando como meio de transmissão 1 par de cabos coaxiais com impedância de 75 Ω resistiva. Esse tipo de meio de transmissão é utilizado para conectar entre si ou entre outros tipos de equipamentos centrais PABX, roteadores, sistemas de gravação digital de voz, entre outros sistemas de telecomunicações.

2.3.1.1 Características elétricas para 2.048 Mbps

A recomendação G.703 (11/2001) recomenda as seguintes especificações físicas e elétricas, para a porta de saída, em uma linha de transmissão de formada de um par de cabos coaxiais com impedância de 75Ω :

Formato do Pulso (Retangular)	Todas as marcas (bits 1) de um sinal válido devem ser conforme a máscara da figura , independente do sinal. O valor V corresponde ao valor nominal da tensão de pico.
Tensão nominal de pico para uma marca (bit 1)	2,37 V
Tensão nominal de pico para um espaço (bit 0)	$0 \pm 0,237 \text{ V}$
Largura nominal de pulso	244 ns
Razão entre as amplitudes dos pulsos positivo e negativo no centro do intervalo de pulso	0.95 a 1.05
Razão entre as larguras dos pulsos positivo e negativo no centro da amplitude nominal	0.95 a 1.05

Tabela 2 – Características elétricas interface G.703

Fonte: ITU-T (11/2001)

A recomendação aborda ainda as seguintes características físicas:

- Taxa de Transmissão: 2048 kbit/s ± 50 ppm (partes por milhão)
- Codificação de Linha: HDB3 (Alta Densidade Bipolar 3 Zeros)

A figura 2-20 abaixo, extraída da recomendação G.703 (11/2001), apresenta a máscara de tolerância para os pulsos presentes na linha com codificação HDB-3.

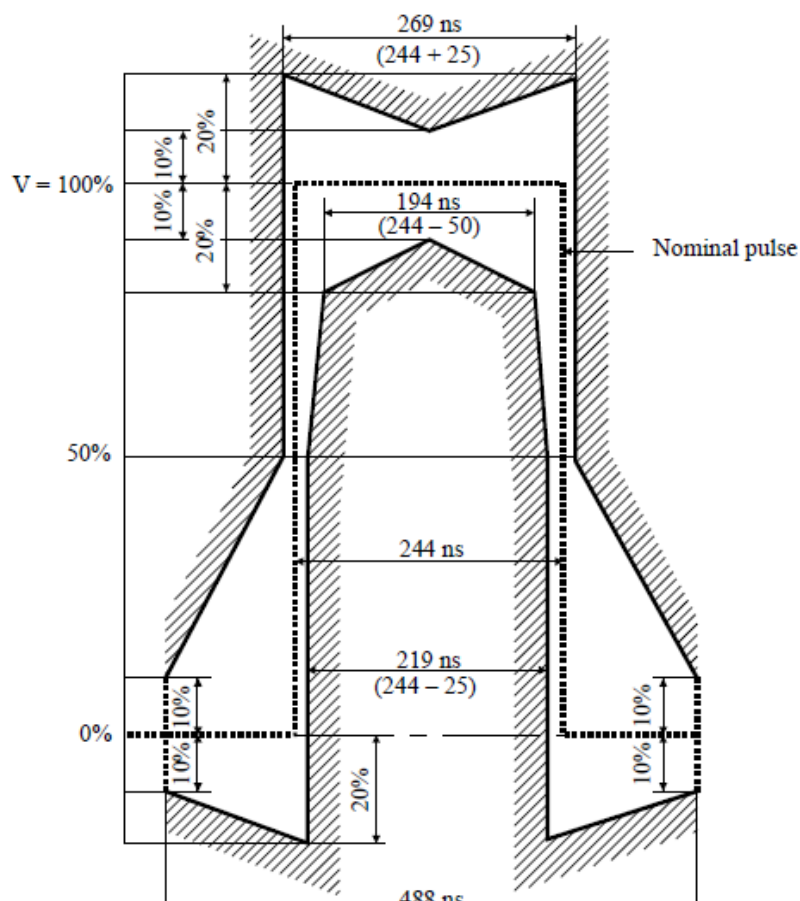


Figura 2-20– Máscara padrão de um pulso HDB-3

Fonte: ITU-T (11/2001)

Na figura 2-21 abaixo é apresentado um exemplo típico de um sinal codificado HDB-3 para a situação de tráfego normal, na entrada do dispositivo (com configuração do osciloscópio para 1V/div – Eixo Y e 500ns/div - Eixo X)

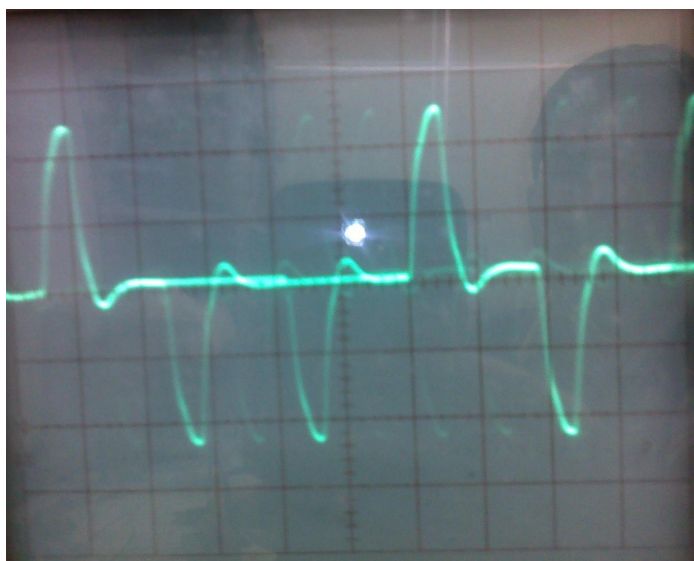


Figura 2-21 – Tela do osciloscópio para tráfego normal

Fonte: Elaborado pelo autor

2.3.2 Tipos de Alarmes de anomalias em 2048kbps

Conforme descrito anteriormente, alguns alarmes são enviados através dos *time slots* 0 e 16 de um quadro PCM, porém outros alarmes são detectados através das características físicas do sinal ao em função do tempo.

2.3.2.1. Detecção de Alarme LOS (Loss of Sincronism)

Conforme o ITU-T (10/1998), recomendação G.775, a condição de detecção de alarme LOS para uma interface de comunicação a 2048 kbps, é a ausência de transições no sinal de entrada por um período de 5 μ s a 1 ms.

Na figura 2-22 a seguir é apresentado um sinal característico presente na linha de transmissão para uma falha do tipo LOS (*Loss of Signal*)

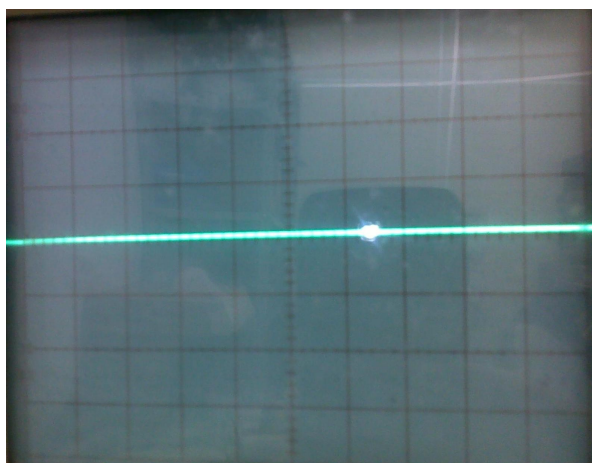


Figura 2-22 – Tela do osciloscópio para alarme tipo LOS

Fonte: Elaborado pelo autor

2.3.2.2. Detecção de alarme AIS (Alarm Indication Signal)

Conforme o ITU-T (10/1998). Recomendação G.775, a condição de alarme AIS para uma interface de comunicação a 2048 kbps pode ser por 3 maneiras:

- 1) Quando o sinal de entrada tem 2 (dois) ou menos zeros em uma sequência de 512 bits ($488\text{ns/bit} \times 512 \text{ bit} = 250 \mu\text{s}$)
- 2) Quando o sinal de entrada apresenta níveis lógicos uns (unitário) durante o tempo de 5 ms, contendo no máximo 0,2% a 0,1% de níveis 0, ou seja, em um intervalo de tempo de 5 ms, a distribuição de sinais com níveis lógicos 1 e níveis lógicos 0, deve ser em uma razão de 99,8 % para 0,2 % no pior caso.

3) Quando o sinal de entrada possui um ou menos níveis lógicos zero, em cada um dos quadros duplos consecutivos (512 bits por quadro duplo) e o sinal FAS (*Frame Alignment Signal*) não for detectado.

Na figura 2-23 a seguir é apresentado um sinal característico presente na linha de transmissão para uma falha do tipo AIS ou SIA. (Alarm Indication Signal)

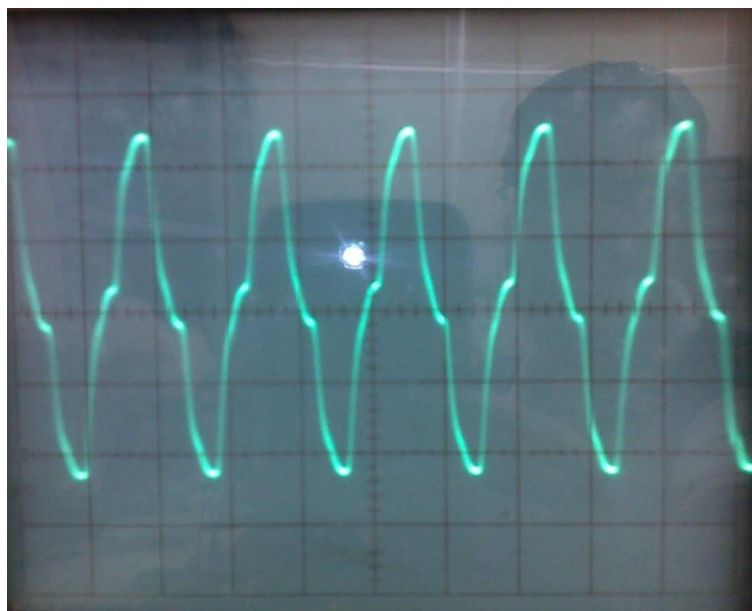


Figura 2-23 – Tela do osciloscópio para alarme tipo AIS

Fonte: Elaborado pelo autor

2.3.2.3. Detecção de alarme RDI (Remote Defect Indication)

Conforme o ITU-T (10/1998), recomendação G.775, a condição de detecção para o alarme RDI é quando o bit A estiver setado para 1 por mais de 5 ms. Este bit está localizado no *time slot* 0, bit 2 da palavra de serviço, amostrada nos quadros ímpares.

2.4. Perdas na Transmissão

As perdas na transmissão são os principais motivos para a geração dos alarmes descritos na seção anterior.

Conforme Forouzan (2008), todos os sinais que trafegam nos meios de transmissão não são perfeitos. Obviamente que um sinal enviado por um emissor nunca terá no receptor as características do sinal original.

Os principais tipos de perdas de transmissão são a atenuação, distorção e o ruído.

A figura 2-24 apresenta um exemplo de atenuação em uma linha típica de telecomunicações, onde o sinal original passa a ser atenuado, ou seja, sofre uma perda de amplitude, após trafegar em um meio de transmissão.

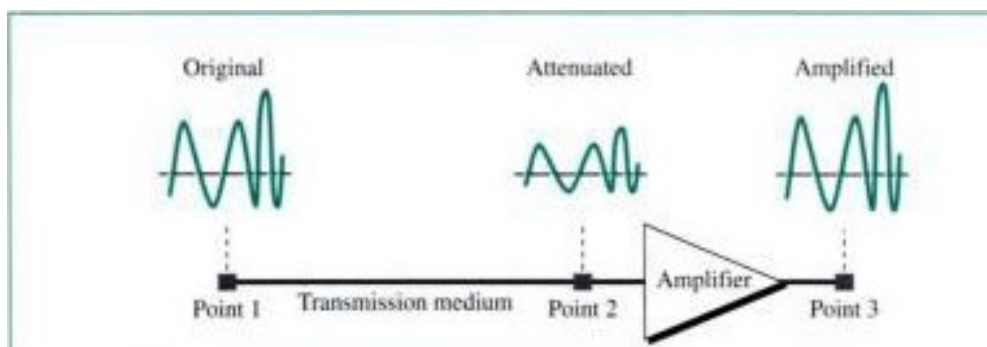


Figura 2-24– Atenuação
Fonte: (FOROUZAN, 2004, p.70)

Segundo Forouzan (2008), a distorção significa que o sinal original muda seu formato original. A figura 2-25 mostra as diferenças entre um sinal original e o sinal no destino que sofreu algum tipo de distorção

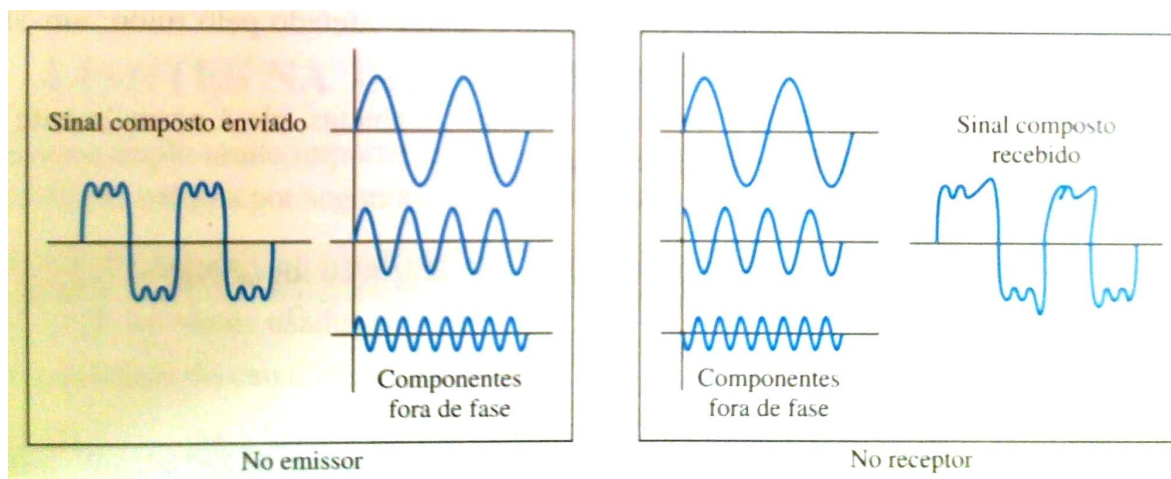


Figura 2-25– Distorção
Fonte: (FOROUZAN, 2008, p.83)

Segundo Pines [1978], ruído é um dos fatores de perda nas transmissões mais difícil de ser superado, e sua presença é inevitável. A figura 2-26 apresenta um exemplo de transmissão, onde é salientada a diferença de um sinal transmitido e recebido, quando sofre algum tipo de ruído.



Figura 2-26 - Ruído
Fonte: (FOROUZAN, 2008, p.84)

3. MATERIAIS E MÉTODOS

3.1. Descrição Geral do Sistema

O dispositivo eletrônico desenvolvido permite monitorar constantemente uma linha de transmissão padrão G.703 em um canal E1, trafegando a uma taxa 2.048 kbps. Através de um algoritmo desenvolvido em Linguagem C e utilizando um microcontrolador PIC 18F4520, conectado a três interfaces distintas, o dispositivo avalia permanentemente a linha de transmissão, gravando em uma memória

massiva do tipo *SD Card* os tipos de falhas ocorridas no sistema, bem como dados completos de data e hora em que ocorreram as mesmas. As três interfaces distintas são:

- Circuito para detectar falhas presentes no canal E1
- Relógio de Tempo Real
- Memória massiva do tipo *SD Card*

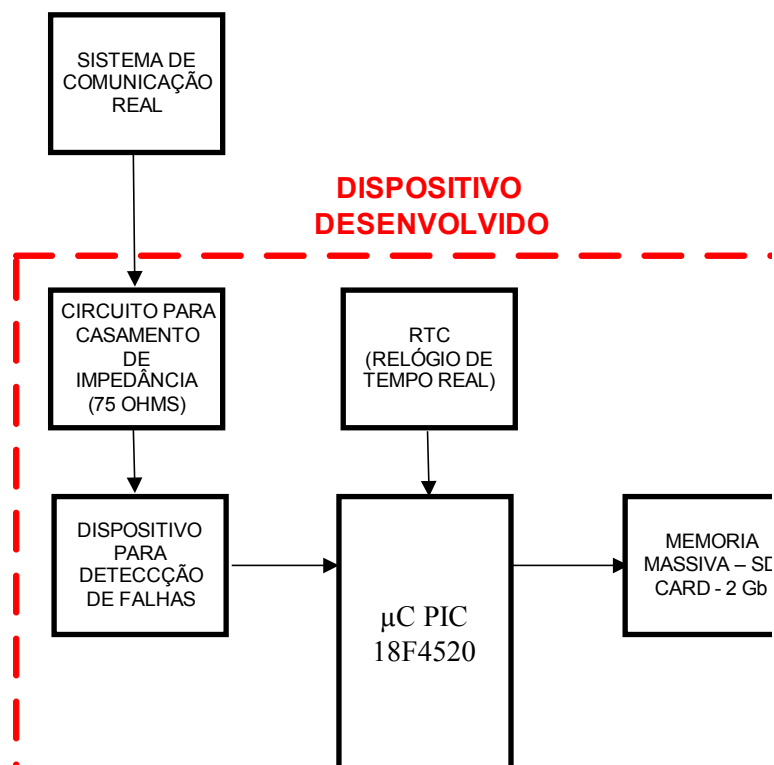


Figura 3-27– Diagrama em blocos do dispositivo
Fonte: Elaborado pelo autor

O dispositivo funciona como um “coletor de campo”, acoplado em paralelo com uma central telefônica em pleno funcionamento, não interferindo em hipótese alguma no funcionamento do sistema de comunicação. Possui na parte frontal um display LCD de 2 linhas por 16 colunas para monitoração visual da linha de transmissão, uma entrada para a memória *SD Card* e uma chave de seleção do modo de operação. Basicamente ele funciona em 2 modos: Manutenção e Coleta, selecionados durante o período de inicialização do sistema.

Display de informações

Slot *SD Card*

Manutenção
BNC para
ligar o E1



Figura 3-28 – Fotografia com visão frontal do dispositivo
Fonte: Elaborada pelo autor

Para o modo de manutenção, o dispositivo possui um *menu* de opções a serem selecionadas pelo operador do sistema, via conexão a porta serial RS-232 de um computador pessoal. As opções a serem selecionadas, permitem a realização da manutenção do dispositivo, tais como: limpeza da memória *SD Card*, descarga dos dados gravados, consultas aos registradores de informações da memória *SD Card*, bem como a escrita/leitura manual de uma determinada posição de memória do *SD Card*.

Na figura 3-3 a seguir, tem-se uma foto da tela de manutenção do dispositivo:

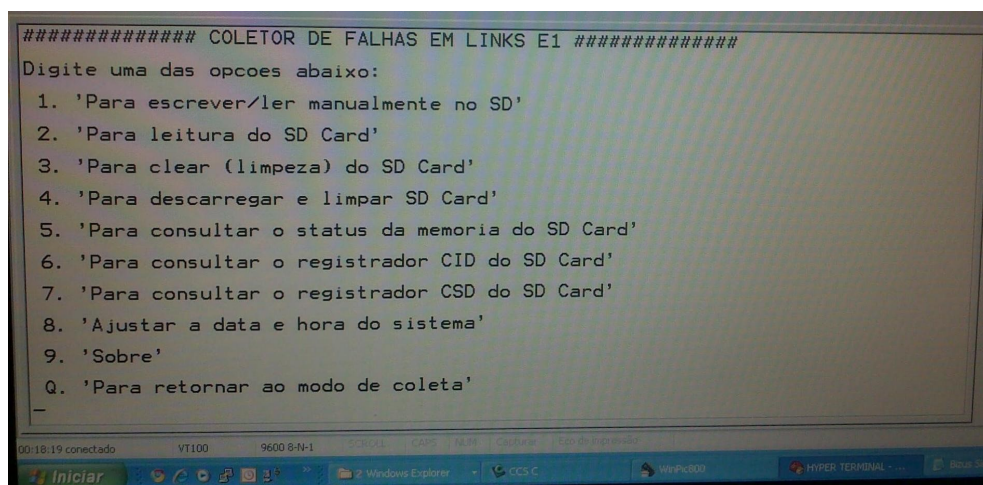


Figura 3-29 – Fotografia com tela de manutenção
Fonte: Elaborada pelo autor

Para o modo de coleta dos eventos de falhas, o dispositivo informa no display frontal o estado em que se encontra o link E1, bem como a data hora local.

Caso ocorra alguma falha o dispositivo informa no painel frontal o tipo de falha detectada, após é transmitido para a porta serial informações detalhadas sobre a falha detectada, bem como endereços em base hexadecimal sobre os bytes gravados na memória *SD Card*, conforme figura 3-4 abaixo:

Endereco: 00000038, Valor: 0
Endereco: 00000039, Valor: 0
Endereco: 0000003A, Valor: 13
Endereco: 0000003B, Valor: 11
Endereco: 0000003C, Valor: 10
Endereco: 0000003D, Valor: 19
Endereco: 0000003E, Valor: 4
Endereco: 0000003F, Valor: 12
Endereco: 00000040, Valor: 255
Gravacao concluida.

Figura 3-30 – Tela de gravação de dados na memória SD Card
Fonte: Elaborada pelo autor

Para quantificar as falhas, são utilizadas 2 variáveis no software - **status** e **falha**, conforme a tabela 3 abaixo:

ESTADO	DESCRIÇÃO	VARIÁVEL STATUS	VARIÁVEL FALHA
LINHA OK	Tráfego normalizado	0	0
FALHA SIA	Alarme detectado dispositivo remoto	1	1
FALHA LOS	Ausência total de transições elétricas no meio de transmissão	1	2

Tabela 3 – Tabela com variáveis para detecção de falhas

3.2. Descrição dos Sistemas Eletroeletrônicos

3.2.1 Dispositivo para detecção de falhas em linhas G703

O dispositivo utilizado para detecção de falhas na linha é formado por 2 componentes: Transceiver bidirecional e o Multivibrador Monoestável Retrigável

3.2.1.1. Transceiver bidirecional

Este circuito integrado ajustado com a função de receiver diferencial, é responsável por transformar os pulsos positivos e negativos presentes na interface G.703 em pulsos somente positivos, ou seja, para adequar os pulsos para o padrão TTL do microcontrolador PIC. Para esta função foi utilizado o circuito integrado SN75176.

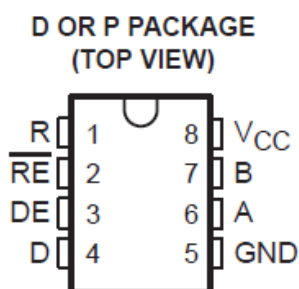


Figura 3-31 – Diagrama de pinos CHIP SN75176
Fonte: Datasheet SN75176

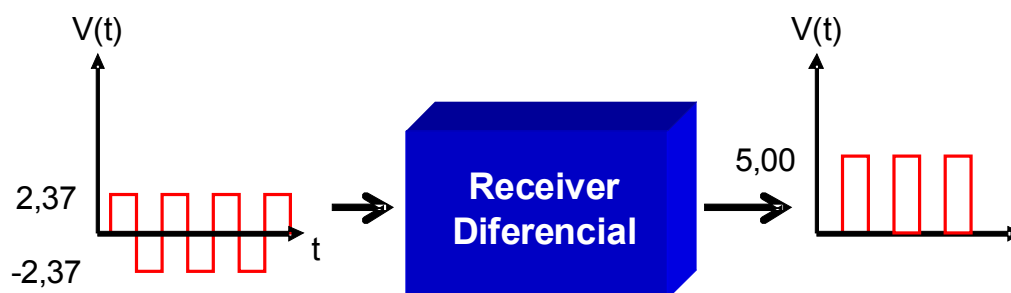


Figura 3-32 – Estrutura de funcionamento do receiver diferencial
Fonte: Elaborada pelo autor

No receiver as estradas diferenciais A e B são ligadas num circuito resistivo casador de impedâncias acoplado diretamente na linha digital de transmissão. Abaixo a tabela com de níveis lógicos na saída R em função do nível de sinal representa nas entradas diferenciais:

RECEIVER		
DIFFERENTIAL INPUTS A – B	ENABLE $\overline{RE}$	OUTPUT R
$V_{ID} \geq 0.2 \text{ V}$	L	H
$-0.2 \text{ V} < V_{ID} < 0.2 \text{ V}$	L	?
$V_{ID} \leq -0.2 \text{ V}$	L	L
X	H	Z
Open	L	?

H = high level, L = low level, ? = indeterminate,
X = irrelevant. Z = high impedance (off)

Tabela 4 – Tabela de funcionamento receiver diferencial
Fonte: Datasheet SN75176

3.2.1.2. Multivibrador Monoestável Retrigável

A utilização deste tipo de dispositivo é uma das formas disponíveis atualmente para detecção de falhas de sinais, pois ele gera na saída um sinal contínuo na saída em função das transições do sinal na entrada. No presente projeto foi utilizado o Circuito Integrado SN74LS123, fabricado pela Motorola.

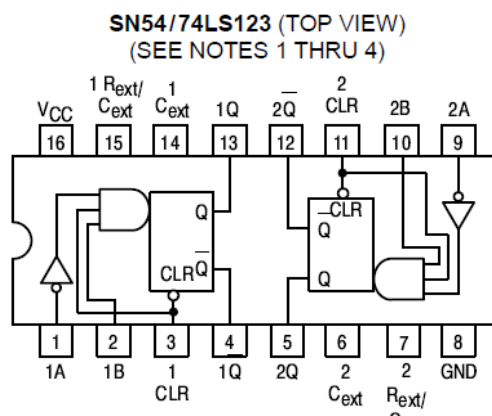


Figura 3-33 – Diagrama de pinos CIP SN74LS123

Fonte: Datasheet SN74LS123

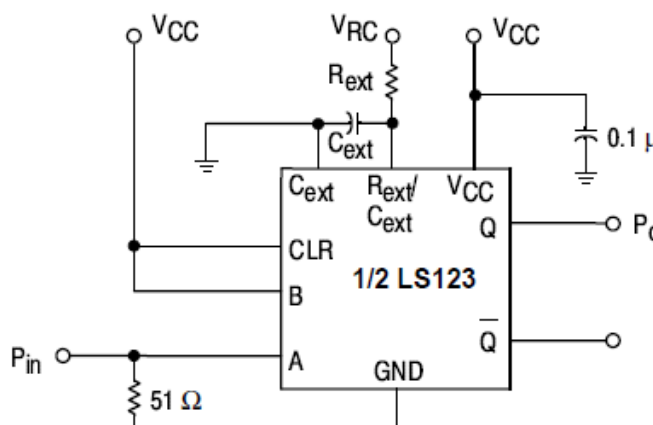


Figura 3-34 – Circuito típico utilizado no CHIP SN74LS123

Fonte: Datasheet SN74LS123

Quando há transições na entrada, que ocorrem em um período de tempo menor que T_w (largura de tempo do pulso de saída). Este tempo é parametrizado em função dos valores de capacitores e resistores utilizados externamente.

Quando não houver transições de sinal de entrada por um período de tempo maior que T_w , o sinal irá para nível lógico 0 na saída.

A função de trigger funciona como um gatilho de tempo, ou seja, a cada vez que é detectada uma transição de sinal de entrada, o temporizador T_w é zerado, fazendo assim com que a saída Q permaneça estável.

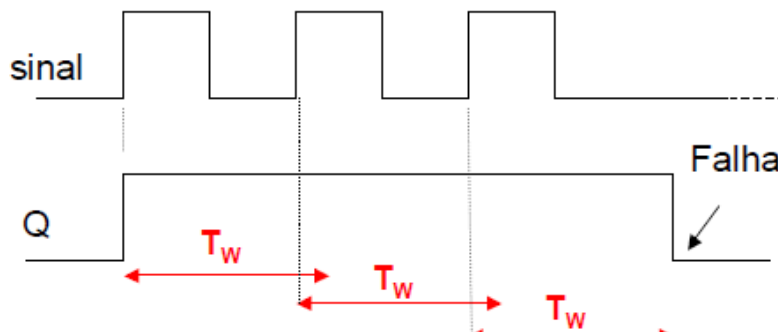


Figura 3-35 – Diagrama de sinais entrada e saída no CHIP SN74LS123

Fonte: Elaborada pelo autor



Figura 3-36 – Estrutura de funcionamento do multivivrador monoestável

Fonte: Elaborada pelo autor

Como o tempo T_w (Time Width) deve sempre ser maior que o período de tempo da frequência de transição do sinal de entrada, foi ajustado o T_w em torno de 1100ns, utilizando um capacitor (C_{ext}) de 100 pF foi possível efetuar o cálculo do resistor (R_{ext}), conforme fórmula para T_w fornecida pelo datasheet do componente:

$$Tw[ns] = 6 + 0,05 \times Cext[pF] + 0,45 \times Re\,xt[k\Omega] \times Cext[pF] + 11,6 \times Re\,xt[k\Omega]$$

$$1100 = 6 + 0,05 \times 100 + 0,45 \times Re\,xt[k\Omega] \times 100 + 11,6 \times Re\,xt[k\Omega]$$

$$Re\,xt[k\Omega] = \frac{1100 - 11}{45 + 11,6}$$

$$Re\,xt[k\Omega] = 19,24k\Omega \rightarrow 20k\Omega$$

3.2.1.3. Circuito eletrônico para detecção de falhas

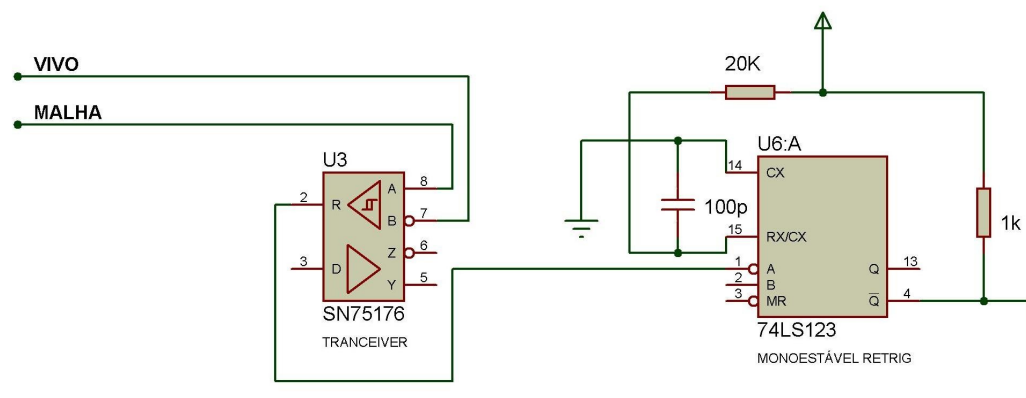


Figura 3-37– Circuito completo com as 2 etapas para detecção de falhas

Fonte: Elaborada pelo autor

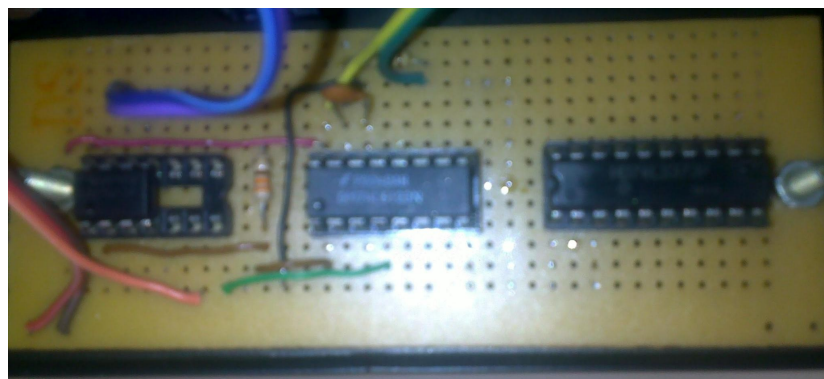


Figura 3-38– Fotografia com a placa de circuito desenvolvido

Fonte: Elaborada pelo autor

3.2.2 Dispositivo para geração da base de tempo

3.2.2.1 Descrição geral

O dispositivo utilizado para geração de uma base de tempo confiável possui como componente principal um RTC (Real Time Clock), modelo DS-1302 fabricado pela *Dallas Semiconductors*. Este dispositivo é responsável por fornecer informações

detalhadas de dia, mês, ano, horas, minutos e segundos, para o microcontrolador PIC anexar junto à palavra de dados que será formada para descrever algum evento de falha.

O DS1302 possui um RTC/calendário e 31 bytes de RAM estática. A comunicação desse dispositivo é realizada através de comunicação serial SPI (Serial Peripheral Interface) a 3 fios. Os 3 fios utilizados para a comunicação com o PIC são:

1° RST (restauração), mesmo que CS (Chip Select)

2° I/O (linha de dados)

3° SCLK (clock serial)

O DS1302 fornece segundos, minutos, horas, dia, data, mês, e ano. A extremidade da data do mês é ajustada automaticamente para meses com menos de 31 dias, incluindo correções para ano bissexto.

O DS1302 é compatível com tecnologia TTL ($V_{CC}=5V$), sendo um circuito integrado projetado para operar com um baixo consumo de energia e manter o relógio operando em menos de $1 \mu W$ de potência.

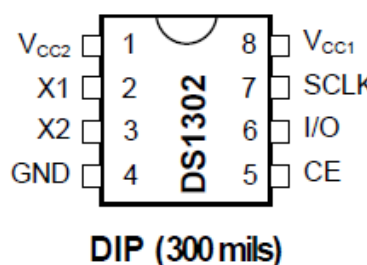


Figura 3-39– Diagrama de pinos CHIP DS 1302

Fonte: Datasheet DS-1302

Além das funções relativas à base de tempo, o DS1302 possui um pino adicional de alimentação (V_{CC1}), que pode ser utilizado por uma fonte de alimentação de backup. Neste projeto foi utilizada uma bateria recarregável de 3 V, para manter a confiabilidade no relógio do dispositivo.

O DS 1302 também possui a função de carregador inteligente da bateria de backup, que deve ser conectado no pino V_{CC2} (alimentação principal) e configurado através de registrador Trickle-Charge, com endereços 91h para leitura e 90h para escrita.

Abaixo circuito eletrônico típico para o funcionamento do RTC DS1302:

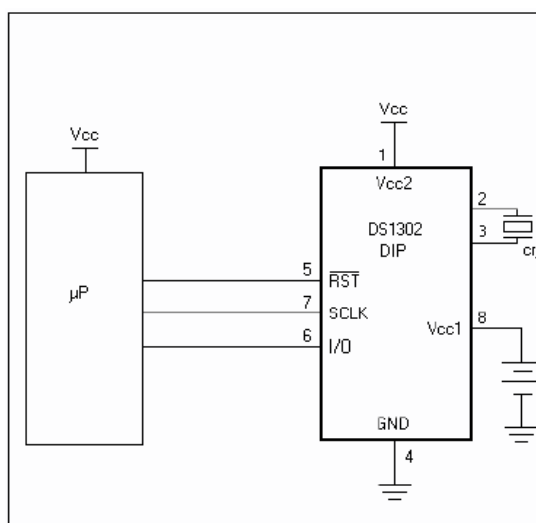


Figura 3-40– Circuito típico de ligação para CHIP DS 1302

Fonte: Datasheet DS1302

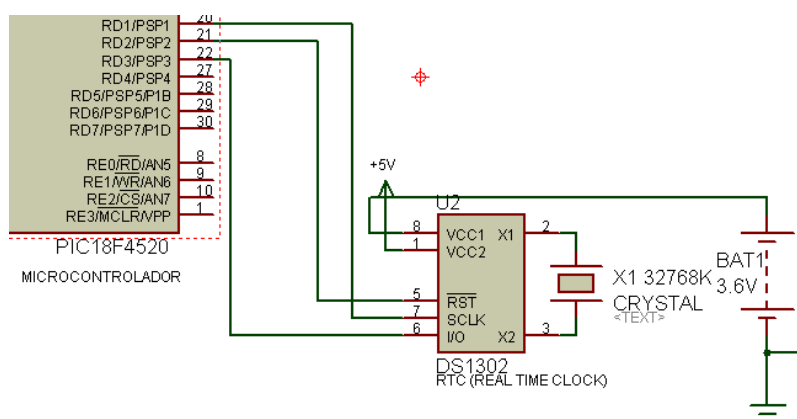


Figura 3-41– Circuito de ligação CHIP DS 1302

Fonte: Elaborada pelo autor

3.2.2.2. Considerações sobre comunicação SPI deste dispositivo

Qualquer processo de comunicação entre dispositivos utilizando o protocolo SPI, tanto para a leitura, tanto para a escrita, basicamente é realizado através de 4 etapas de comandos distintas, a serem realizadas e controladas a partir do dispositivo mestre do sistema. Para este projeto o dispositivo mestre é o microcontrolador PIC e o escravo o RTC DS 1302. A figura abaixo apresenta as 4 etapas:

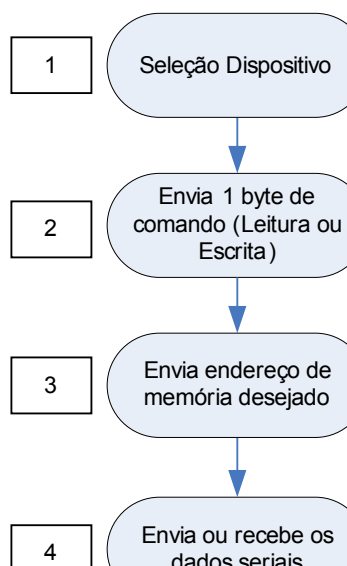


Figura 3-42 – Etapas de comando CHIP DS 1302

Fonte: Elaborada pelo autor

Porém, como o DS1302 possui apenas 10 informações disponíveis para escrita ou leitura, mais 31 endereços de RAM estática, as etapas 2 e 3 são agrupadas, fazendo com que o byte de comando já informe se a operação é de leitura ou escrita, através do bit menos significativo (0) , conforme figura abaixo:

7	6	5	4	3	2	1	0
1	RAM	A4	A3	A2	A1	A0	RD
	$\overline{CK}$						$\overline{WR}$

Figura 3-43– Byte de comando SPI CHIP DS1302

Fonte: Datasheet DS1302

As etapas do processo ficam reduzidas a três para o DS 1302, conforme figura abaixo:

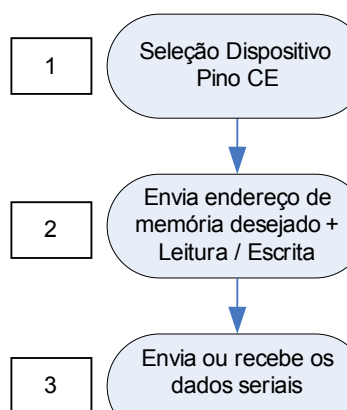


Figura 3-44– Etapas de comando SPI reduzidas

Fonte: Elaborada pelo autor

3.2.2.3. Funcionamento da escrita de dados

Todas as transferências de dados são iniciadas através da habilitação do pino CE (Chip Enable), ou seja, setando para nível lógico 1 comandado pelo microcontrolador PIC.

Após ser habilitado o pino CE, o PIC coloca no pino I/O o bit desejado (0 ou 1), após gera um pulso de clock no pino SCLK do RTC. O pulso de clock é um comando a partir do PIC para colocar o pino SCLK em nível alto, seguido de um comando para colocar o nível baixo nesse pino. Esse pulso de clock possui uma duração 1 ciclo de máquina, ou seja, 500ns conforme será visto mais detalhadamente na seção 3.2.4. A cada pulso de *clock*, os bits contidos no byte de envio desejado, são rotacionados para direita na saída do pino de I/O, um a um., sendo controlados sempre pela borda de descida do pino SCLK. Esta geração de *clock* é repetida por 8 vezes, formando assim um byte, sendo utilizado para escrita por exemplo de um byte de comando, operação indispensável para qualquer operação de transferência de dados. A operação de escrita também é realizada quando se deseja um ajuste de data/ hora manualmente pelo operador, através do menu de opções do dispositivo projetado.

3.2.2.4. Funcionamento da leitura de dados

O processo de leitura de dados sempre ocorre quando o PIC precisa realizar alguma consulta aos dados de data e hora, geralmente quando alguma anomalia é detectada no E1.

Similar ao processo de escrita, o processo de leitura é iniciado pela habilitação do pino CE pelo PIC e em seguida é escrito um byte de comando pelo mesmo, conforme explicado na seção anterior. Após o envio do byte de comando, o PIC recebe no pino de I/O os bits, um a um, enviados pelo DS 1302, formando um byte de informação a cada 8 bits, controlados pelas bordas de subida dos pulsos gerados no pino SCLK.

A figura abaixo explica de uma forma resumida o processo de escrita e leitura realizada no DS 1302 utilizado no projeto:

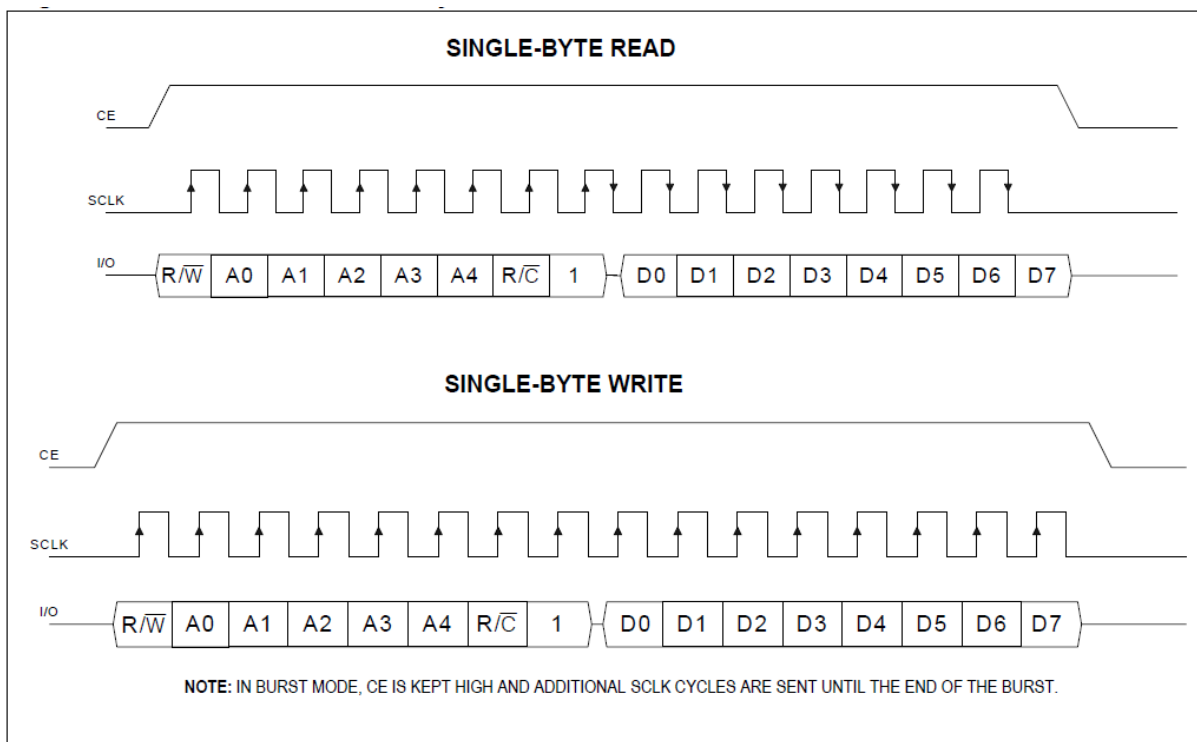


Figura 3-45– Processo de leitura e escrita simplificado no DS1302

Fonte: Datasheet DS1302

Para o ajuste de data/hora no dispositivo ou a consulta da data/hora que ocorrer alguma falha, software desenvolvido utiliza os seguintes bytes de comando, conforme as operações de escrita ou leitura descritas na tabela 5:

RTC

READ	WRITE	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0	RANG
81h	80h	CH	10 Seconds			Seconds				00–59
83h	82h		10 Minutes			Minutes				00–59
85h	84h	12/24	0	10 AM/PM	Hour	Hour				1–12/0–
87h	86h	0	0	10 Date		Date				1–31
89h	88h	0	0	0	10 Month	Month				1–12
8Bh	8Ah	0	0	0	0	0	Day			1–7
8Dh	8Ch	10 Year				Year				00–99
8Fh	8Eh	WP	0	0	0	0	0	0	0	—
91h	90h	TCS	TCS	TCS	TCS	DS	DS	RS	RS	—

CLOCK BURST

RFh	RFh
-----	-----

Tabela 5– Estrutura dos registradores utilizados DS1302

Fonte: Datasheet DS1302

3.2.3 Dispositivo para armazenamento massivo das falhas

3.2.3.1. Descrição Geral

A cada falha ocorrida no sistema de coleta, o microcontrolador PIC lê no DS1302 6 bytes de informações sobre data e hora e mais 2 bytes no dispositivo para detecção de falhas, para caracterizar o tipo de falha ocorrida no meio de transmissão. Como o sistema foi desenvolvido para operar por grandes períodos de tempo, é utilizada no projeto uma memória massiva do tipo SD-Card.

As memórias do tipo *SD- Card* (Secure Digital – Cartão) são amplamente utilizadas nas mais diversas aplicações eletrônicas para o armazenamento de informações em massa. No projeto foi utilizada uma memória do fabricante *Sandisk*, do tipo micro SD, com capacidade nominal de armazenamento de 2 Gbytes, ou aproximadamente 2 bilhões de bytes.



Figura 3-46 – Fotografia com dimensão do tamanho de uma micro SD
Fonte: Elaborada pelo autor

Para a conexão do cartão de memória do tipo micro SD, com o circuito, foi utilizado um adaptador de cartão micro SD para cartão SD de fabricante *Sandisk*



Figura 3-47 – Fotografia com dimensão do tamanho de um adaptador micro SD
Fonte: Elaborada pelo autor

Para a conexão adequada do cartão SD-Card com o PIC e conseqüente alimentação padrão da memória de 3.3 V, foi utilizado um módulo de conexão para cartões SD ACE SDCARD, desenvolvido pela ACEPIC Tecnologia e Treinamento Ltda

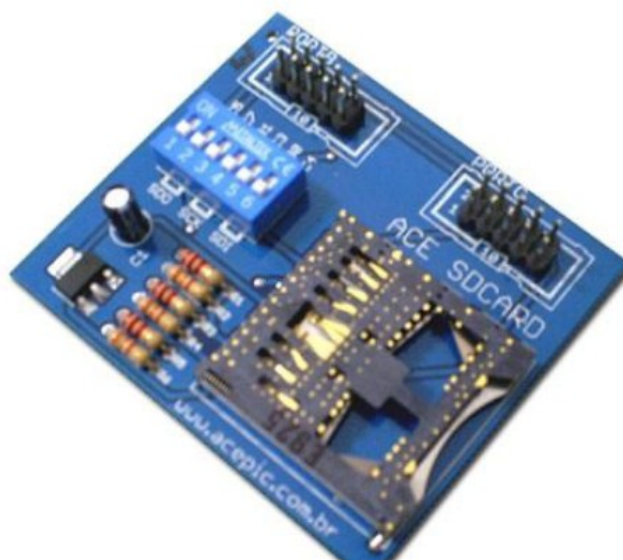


Figura 3-48 – Placa ACE SD Card

Fonte: www.acepic.com.br acessado em 14/10/2010

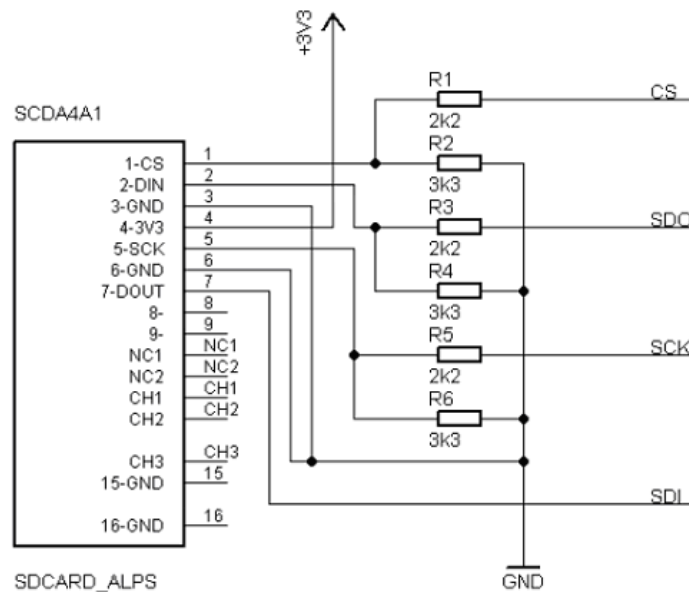


Figura 3-49 – Circuito de conversão de sinais elétricos

Fonte: www.acepic.com.br acessado em 14/10/2010

3.2.3.2. Características Gerais

As principais características desse tipo de memória são:

- Compatibilidade com protocolo de comunicação SPI (Serial Peripheral Interface)
- Tensão de alimentação de 2,7V a 3,6 V
- Frequência de clock variável: 0 a 25 MHz (padrão) ou 0 a 50 MHz (Alta performance)
- Taxa de transferência de até 25 Mbps
- Possibilidade de proteção dos dados por senha
- Chave mecânica para proteção de escrita
- Setor de gravação com tamanho variável, sendo o valor padrão de 512 bytes
- Possui alta confiabilidade dos dados gravados.

Na figura abaixo são descritos os pinos utilizados para comunicação SPI com um cartão SD:

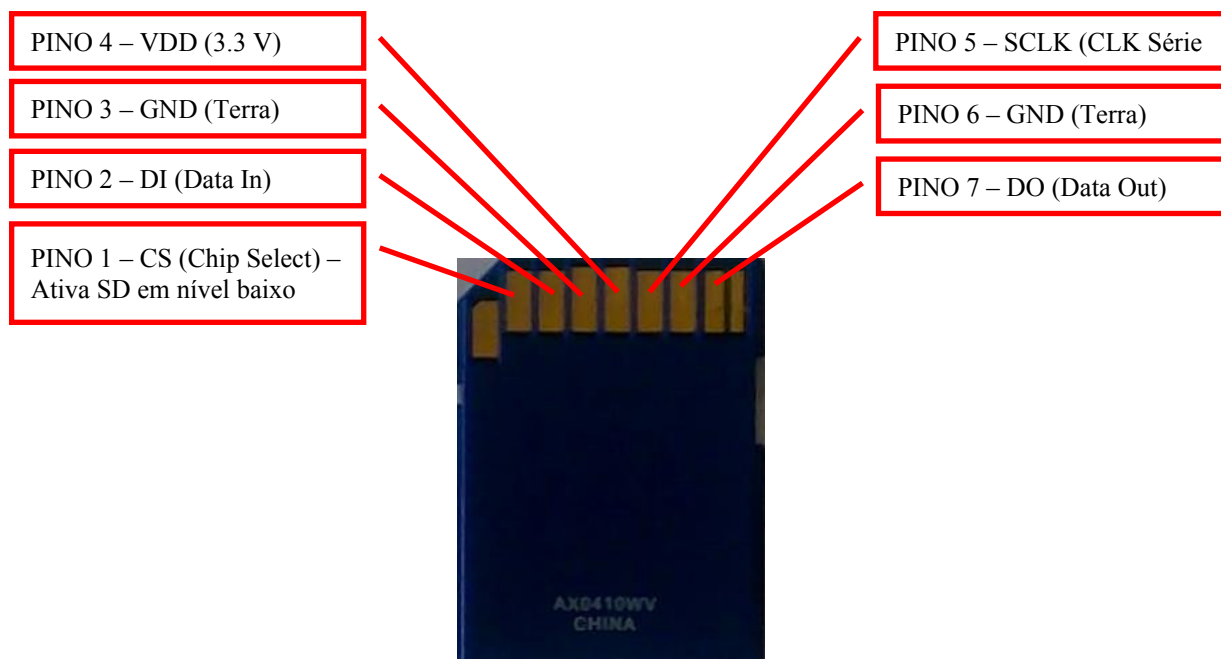


Figura 3-50 – Diagrama de pinos para comunicação SPI SD Card
Fonte: Elaborado pelo autor

3.2.3.3. Características Elétricas

Tensão de Alimentação: 2,7V a 3,6 V

Consumo para Leitura com velocidade de 25Mbps: 100mA

Consumo para Escrita com velocidade de 25Mbps: 100mA

Consumo em Modo sleep (espera): 250 μ A

3.2.3.4. Funcionamento

Conforme descrito anteriormente, para a comunicação com a memória SD Card foi utilizado o protocolo SPI (Serial Peripheral Interface), utilizando a gravação de byte a byte e não em blocos. O processo de leitura e escrita na memória é similar ao dispositivo RTC DS-1302 descrito na seção 3.2.2, porém a quantidade de bits envolvidos em cada operação é bem maior.

Assim como outros dispositivos SPI, uma memória SD Card opera a partir de bytes de comando originados a partir de um *host* (PIC 18F4520), comandos estes

utilizados para determinar o tipo de operação que será realizado na sequência. Cada comando possui sempre um registrador de resposta associado, onde o host deverá sempre realizar uma avaliação desta resposta. Abaixo um exemplo de uma simples operação de leitura de um bloco de dados de uma memória SD via SPI:

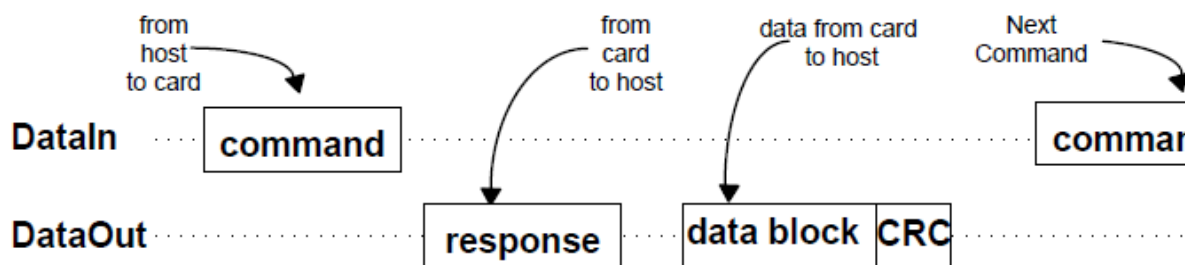


Figura 3-51 – Estrutura de comunicação SPI memória SD Card

Fonte: <http://www.sdcard.org> acessado em out/2010

A palavra de comando possui 48 bits de comprimento, totalizando 6 bytes completos de informações, sendo que a leitura sempre começa pelo bit mais significativo. A seguir tabela com um formato padrão de um comando SPI para uma memória do tipo SD Card:

Posição	47	46	[45:40]	[39:8]	[7:1]	0
Descrição	Bit início	Bit Transmissão	Código comando 0-63d	Argumento	CRC-7	Bit Fim
Qtde Bits	1	1	6	32	7	1
Valor padrão	0	1	0x00 0000 a 0x11 1111	x	x	1

Tabela 6 – Estrutura dos bytes de comando SPI para cartões SD

Fonte: Fonte: <http://www.sdcard.org> acessado em out/2010

Para os registradores de resposta são utilizadas palavras com comprimento de 8bits (1byte), denominadas de Rn, porém o registrador mais utilizado é o R1. Similar a palavra de comando, os bits começam a ser lidos sempre pelo mais significativo (MSB), que sempre deverá ser 0. Abaixo o formato padrão para o byte de resposta R1:

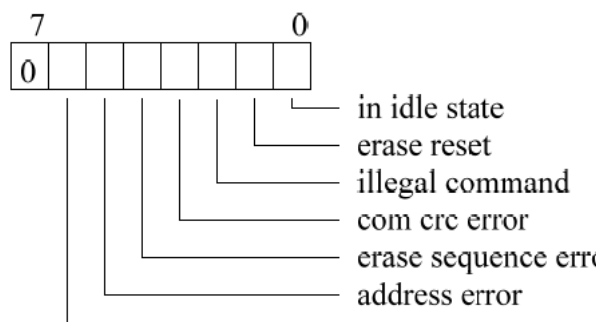


Figura 3-52 – Bits dos registradores de resposta SPI para SD Card

Fonte: <http://www.sdcard.org> acessado em out/2010

Abaixo na tabela 7, são apresentados os principais comandos SPI utilizados no projeto, bem como os registradores de resposta associados:

CMD Index	ABREV	DESCRIÇÃO	BINÁRIO	RESP
CMD0	GO_IDLE_STATE	Utilizado para RESET do SD	00 0000	R1
CMD1	SEND_OP_COND	Inicializa o SD Card	00 0001	R1
CMD8	SEND_IF_COND	Envia informações elétricas de trabalho	00 1000	R7
CMD9	SEND_CSD	Solicita informações sobre Registrador o Card Specific Data (CSD)	00 1001	R1
CMD10	SEND_CID	Solicita informações sobre o Registrador Card Identification (CID)	00 1010	R1
CMD13	SEND_STATUS	Verifica condição do Registrador de Status do SD	00 1101	R2
CMD16	SET_BLOCKLEN	Configura no SD o tamanho do bloco de dados. No caso de cartões SDHC é fixado em 512 bytes	01 0000	R1
CMD17	READ_SINGLE_BLOCK	Lê um bloco de dados com tamanho setado no CMD16	01 0001	R1
CMD24	WRITE_BLOCK	Grava um bloco de dados com tamanho setado no CMD16	01 1000	R1
CMD55	APP_CMD	Verifica se o cartão em serviço é do tipo SD ou MMC	11 0111	R1
CMD58	READ_OCR	Lê o Registrador OCR do SD card	11 1010	R3



CMD59	CRC_ON_OFF	Habilita ou desabilita o CRC-7 durante a transmissão	11 1011	R1
-------	------------	--	---------	----

Tabela 7 – Estrutura dos bytes de comando SPI para cartões SD

Fonte: Fonte: <http://www.sdcard.org> acessado em out/2010

A etapa crítica do funcionamento do SD Card é a etapa de inicialização, que precisa seguir rigorosamente a seguinte sequência de comandos descritas na figura 3-27:

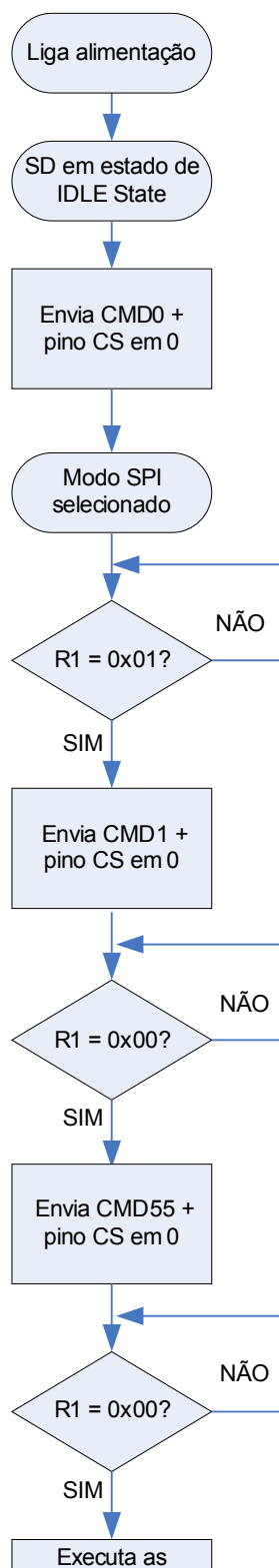


Figura 3-53– Fluxograma de inicialização memória SD Card com protocolo SPI
Fonte: <http://www.sdcard.org> acessado em out/2010

Além dos registradores de respostas, existem mais 3 registradores geralmente utilizados que são o CID, CSD e o OCR.

O CID (Card Identification) possui comprimento de 128 bits e contém dados de identificação do SD card, originalmente gravados durante o processo de fabricação do mesmo. Abaixo exemplo de uma leitura do registrador CID:

```
Manufacturer ID: 30
OEM/Application ID: SD
OEM/Application ID: SU02G
Product Revision: 80
Serial Number: 0019D401
Manufacturer Date Code: 00A7
CRC-7 Checksum: D5
```

Figura 3-54 – Exemplo de leitura registrador CID
Fonte: Elaborada pelo autor

O CSD (*Card Specific Data*) possui comprimento de 128 bits sendo um registrador das informações de configuração. Este registrador é necessário para acessar os dados do cartão. O CSD define o formato de dados, tipo de correção de erros, tempo de acesso máximo de dados, etc. Os valores do registrador CSD podem variar dependendo das especificações físicas e capacidade de memória do SD-Card. Abaixo exemplo de uma leitura do registrador CSD:

```
CSD_STRUCTURE: 00
TAAC: 5E
NSAC: 00
TRAN_SPEED: 32
CCC: 05B5
READ_BL_LEN: 0A
READ_BL_PARTIAL: 01
WRITE_BLK_MISALIGN: 00
READ_BLK_MISALIGN: 01
DSR_IMP: 00
C_SIZE: A7
VDD_R_CURR_MIN: 07
VDD_R_CURR_MAX: 07
VDD_W_CURR_MIN: 07
VDD_W_CURR_MAX: 07
C_SIZE_MULT: 07
ERASE_BLK_EN: 01
SECTOR_SIZE: 7F
WP_GRP_SIZE: 00
WP_GRP_ENABLE: 00
R2W_FACTOR: 02
WRITE_BL_LEN: 0A
WRITE_BL_PARTIAL: 00
FILE_FORMAT_GRP: 00
COPY: 00
PERM_WRITE_PROTECT: 00
TMP_WRITE_PROTECT: 00
FILE_FORMAT: 00
CRC: F1
```

Figura 3-55 – Exemplo de leitura registrador CSD
Fonte: Elaborada pelo autor

O OCR (*Operation Conditions Register*) possui 32 bits de comprimento sendo responsável por armazenar os perfis utilizados de alimentação VDD nos pinos do *SD Card*

3.2.4 Microcontrolador PIC 18F4520

Para o processamento e controle dos dispositivos desenvolvidos no projeto, foi utilizado um microcontrolador PIC da família 18F, modelo 18F4520.

O microcontrolador PIC 18F4520 é um dispositivo desenvolvido e fabricado pela *Microchip*, que possui uma CPU de alta performance com tecnologia RISC (*Reduced Instruction Set Computer*). Esta tecnologia traz uma grande vantagem em relação à velocidade de processamento, pois muitos códigos de instruções são efetuados em apenas um ciclo de máquina. Isto significa que a maioria das instruções são realizadas em 1 período de tempo de clock do cristal, multiplicados por quatro. Pois para cada quatro ciclos de frequência do cristal ressonador acoplado ao PIC, tem-se uma frequência do ciclo de máquina. A máxima frequência de *clock* para este dispositivo é de 40 MHz.

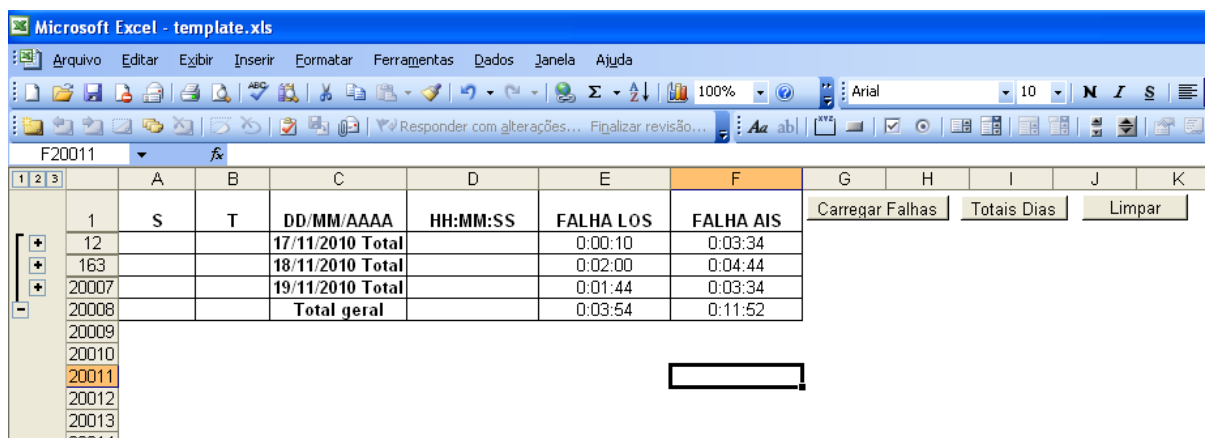
O dispositivo utilizado no projeto possui um oscilador de 8MHz portanto os ciclos de máquina oscilam em uma frequência de 2 MHz, sendo assim um período de 500ns, tempo este utilizado para a execução da maioria das instruções. Além disso, esse microcontrolador possui como vantagem de ter apenas 35 instruções (*assembly*).

O microcontrolador PIC dessa família também possui uma série de funcionalidades incorporadas na mesma estrutura física do mesmo, como uma memória de programa de 32kbytes e mais 768 bytes de memória RAM. Possui tecnologia CMOS FLASH / EEPROM, com características de baixo consumo de energia, permitindo um armazenamento não volátil de dados na EEPROM interna em tempo de execução.

O PIC 18F4520 possui 4 timers distintos, sendo 1 de 8 bits e 3 com 16 bits, tendo a possibilidade do uso de até 20 interrupções. Ele conta com 33 pinos, de I/O (entrada/ saída), configuráveis, distribuídos em 5 portas, geralmente com 8 pinos (1 byte) cada porta. Alguns desses pinos são compartilhados para uma série de outras funções como conversor A/D de 8 canais para 10 bits, 2 pinos para captura, comparação e módulo PWM. Além disso esses pinos são compartilhados também com funções específicas para a comunicação com outros dispositivos periféricos, o PIC 16F877A possui funções de Porta Serial Síncrona com SPI (Mestre) e I2C

3.3.3 Interface para visualização dos resultados

Foi desenvolvida uma planilha padrão (template) no Microsoft Excel para conversão dos arquivos “txt” extraídos durante o processo de descarga dos eventos gravados no SD. Esta planilha também realiza a consolidação dos eventos automaticamente de falhas, através de uma macro desenvolvida para este propósito. A planilha Excel possui o seguinte formato:



The screenshot shows a Microsoft Excel window titled "Microsoft Excel - template.xls". The interface includes a menu bar (Arquivo, Editar, Exibir, Inserir, Formatar, Ferramentas, Dados, Janela, Ajuda) and a toolbar. The active sheet is "F20011". The data table is structured as follows:

	A	B	C	D	E	F	G	H	I	J	K
1	S	T	DD/MM/AAAA	HH:MM:SS	FALHA LOS	FALHA AIS					
12			17/11/2010 Total		0:00:10	0:03:34					
163			18/11/2010 Total		0:02:00	0:04:44					
20007			19/11/2010 Total		0:01:44	0:03:34					
20008			Total geral		0:03:54	0:11:52					
20009											
20010											
20011											
20012											
20013											
20014											

Below the table, there are three buttons: "Carregar Falhas", "Totais Dias", and "Limpar".

Figura 3-57 – Modelo de planilha Excel desenvolvida para consolidação dos eventos

Fonte: Elaborada pelo autor

3.4. Fluxograma do software desenvolvido – Modo de Coleta

Conforme a figura 3-32 abaixo, após a inicialização do dispositivo, o mesmo sempre realiza a verificação do cartão SD inserido corretamente no *slot* destinado para ele. Caso não seja verificada a presença do SD, o dispositivo retorna ao início do programa, solicitando na tela do operador e display LCD frontal a presença do cartão SD. Isto faz com que a presença do cartão SD no dispositivo seja imprescindível.

Com a chave frontal de seleção do dispositivo selecionada para a opção de coleta, o dispositivo trabalha para realizar uma espécie de “escuta” no meio de transmissão, realizando ciclos para análise do meio conforme a figura 3-32. Caso não seja detectada nenhuma falha no meio, o dispositivo retorna para uma nova análise. Caso seja detectada alguma anomalia no meio, é executada na sequência



uma função para análise do tipo de falha, que basicamente analisa o estado do pino da saída do monoestável retriggerável (pino RC0) do microcontrolador.

Na sequência é realizada uma função para a leitura do RTC, com o objetivo de se obter todos os dados relativos à data/hora da falha ocorrida. Após é realizada a gravação dos 8 bytes no cartão SD (2 bytes de caracterização da falha + 6 bytes de informações de tempo)

Após a gravação dos 8 bytes de informações no cartão SD, o dispositivo entra em um estado de espera até que a linha normalize, sendo que são realizado laços contínuos até o timer1 detectar novamente a presença de pulsos na linha. Para esta detecção de normalização da linha, são realizados os mesmos passos para detecção de alguma anomalia. A partir deste instante é completado um ciclo de trabalho do dispositivo.

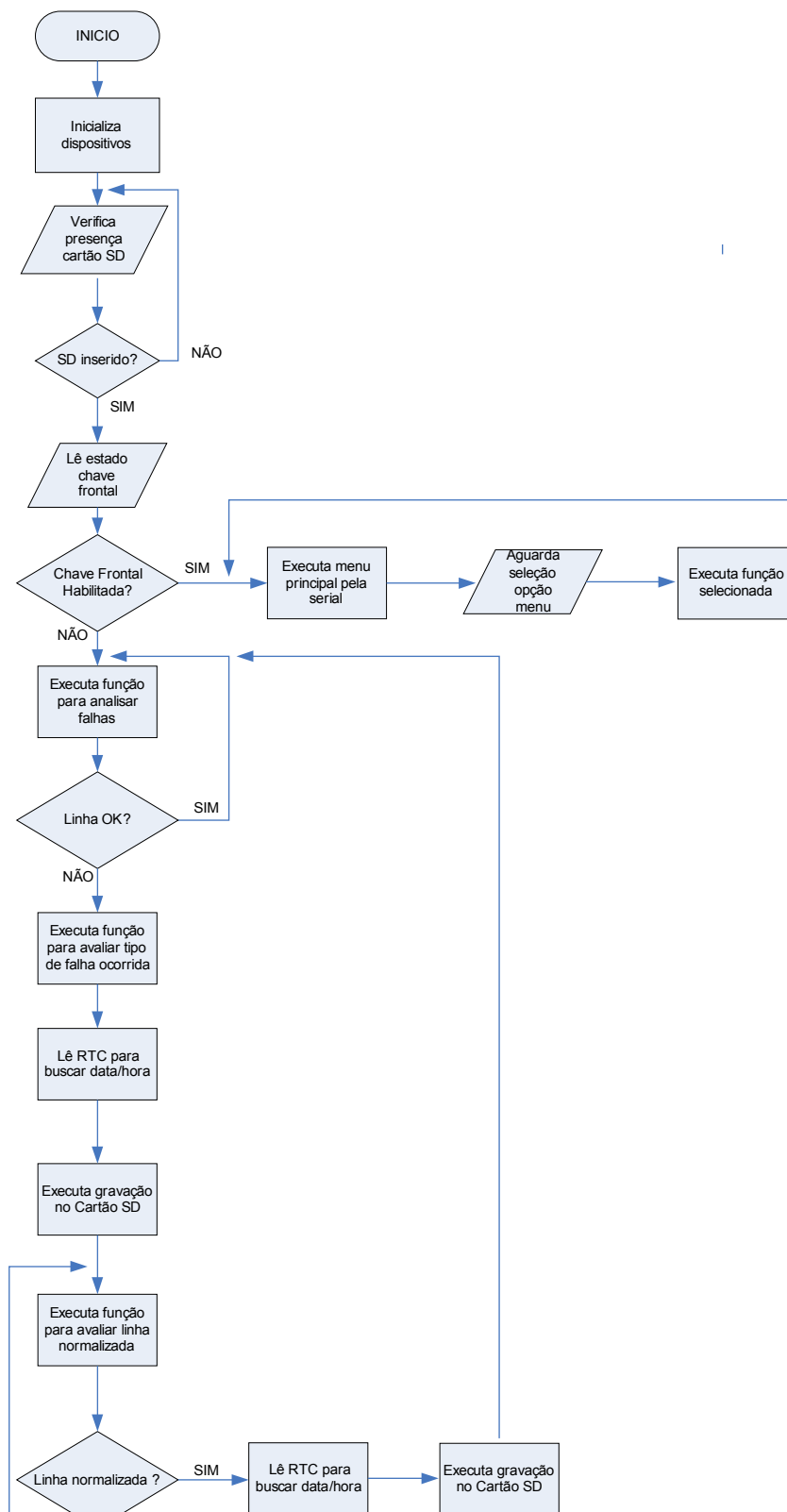


Figura 3-58 – Fluxograma de software desenvolvido para modo coleta

Fonte: Elaborado pelo autor

3.5. Fluxograma do software desenvolvido – Modo de manutenção via porta serial

A figura 3-33 abaixo descreve o modo de funcionamento para o caso de seleção da chave frontal para o modo de manutenção, onde através da conexão do dispositivo na porta serial RS-232, são geradas opções de seleção através do teclado do PC, além disso, todas as respostas são visualizadas no terminal. As seguintes opções são exibidas para a escolha pelo operador:

- Opção 1 – Possibilidade de leitura e escrita manualmente no cartão SD, informando apenas o endereço em hexa desejado, e em caso de escrita devem-se serem fornecidos os dados a serem gravados no respectivo endereço.
- Opção 2 – Executa a leitura e mostra no terminal todos os endereços do SD até a última posição em que se teve gravação.
- Opção 3 – Executa uma função para a limpeza do cartão SD, ou seja, são gravados bytes 0 em todas as posições da memória.
- Opção 4 – Executa uma função para a descarga do cartão SD. A diferença desta opção em relação à leitura é que os dados exibidos na tela, já ficam em um formato bruto, próprios para a exportação dos dados em MS Excel.
- Opção 5 – Executa uma função para a contagem de quantas posições estão ocupadas com gravação no cartão SD. Retorna com número int32
- Opção 6 – Exibe na tela as configurações do atuais do registrador CID do cartão SD.
- Opção 7 – Exibe na tela as configurações do atuais do registrador CSD do cartão SD.
- Opção 8 – Executa uma função que possibilita o ajuste de data/hora do RTC através do teclado do PC conectado via serial
- Opção 9 – Exibe dados de versão do software e autor.

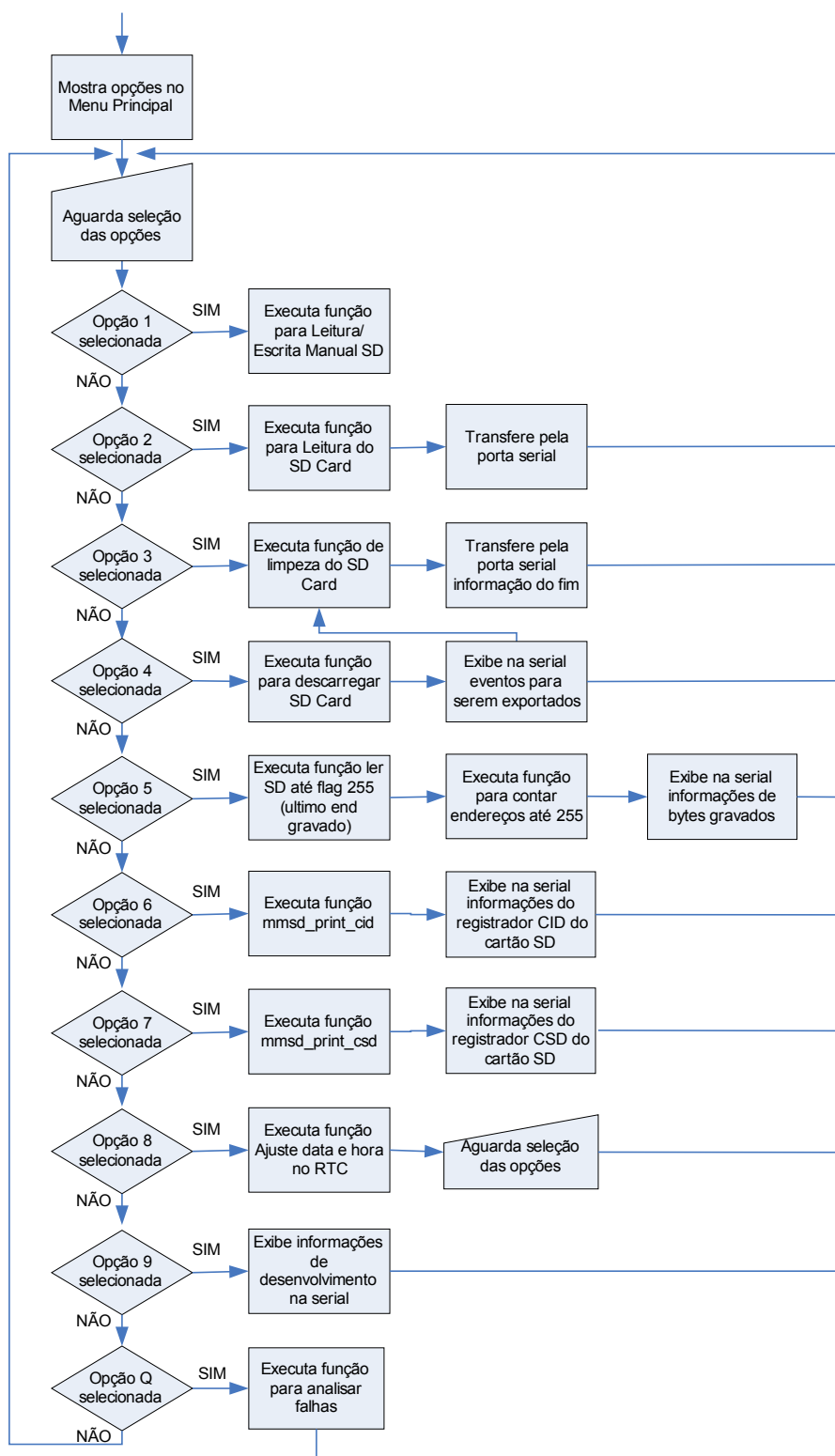


Figura 3-59 – Fluxograma de software desenvolvido para modo manutenção

Fonte: Elaborado pelo autor

4. APRESENTAÇÃO E DISCUSSÃO DOS RESULTADOS

4.1. Testes Preliminares

4.1.1 Simulações da Capacidade de Memória SD

Foram efetuados diversos testes quanto a capacidade máxima de armazenamento no cartão SD, porém na prática com o dispositivo desenvolvido é muito difícil preencher toda a capacidade real de armazenamento do SD Card de 2Gb que é de 1.966.866.432 bytes no total, pois o mesmo possui na verdade 3.842.048 setores de 512bytes cada, totalizando 1.967.128.576 bytes. Esta análise foi realizada pelo software free *Roadkil's Sector Editor v1.4*

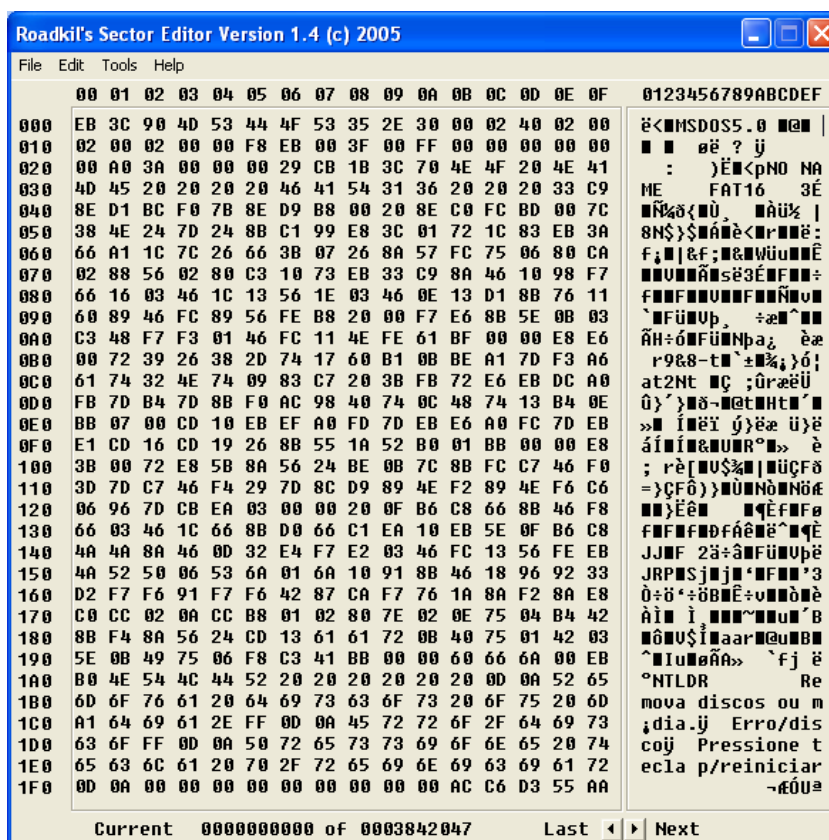


Figura 4-60– Tela do software para análise de setores memória SD Card

Fonte: Elaborado pelo autor

Esta diferença de 262.144 (2^{18}) bytes estão armazenados os registradores de instruções, bem como configurações do cartão SD.

Considerando que o dispositivo grava no máximo 1 evento de 8 bytes a cada segundo, seriam necessários:

$$Tempo = \frac{1.966.866.432 \text{ bytes}}{8 \text{ bytes}} = 245.858.304 \text{ seg}$$

O que equivale a aproximadamente 7,8 anos com 1 evento por segundo neste período.

4.1.2 Simulações de falhas de energia durante etapa de Gravação

Foram efetuadas simulações de falha de energia durante o funcionamento do dispositivo e o mesmo se tornou bem eficiente, pois o software do microcontrolador durante a inicialização sempre procura o último byte gravado, que é a flag 255 no último endereço. Portanto o sistema começa a gravar na memória sempre após o último endereço (flag 255).

O problema detectado é quando a etapa de gravação é interrompida no meio da palavra de 8 bytes, portanto a flag 255 nunca é localizada, fazendo o sistema de contagens sempre apresentar um erro. Mas na prática esta falha é difícil de ocorrer pois a etapa de gravação ocorre de byte a byte e não em blocos, porém este tempo de gravação de 8 bytes é muito pequeno se comparado ao restante do processo.

Exemplo abaixo:

```
Endereco: 00000000, Valor: 1
Endereco: 00000001, Valor: 2
Endereco: 00000002, Valor: 13
Endereco: 00000003, Valor: 11
Endereco: 00000004, Valor: 10
Endereco: 00000005, Valor: 18
Endereco: 00000006, Valor: 30
Endereco: 00000007, Valor: 41
Endereco: 00000008, Valor: 255
Gravacao concluida. ----->Palavra 1 OK
Endereco: 00000008, Valor: 0
Endereco: 00000009, Valor: 0
Endereco: 0000000A, Valor: 13
Endereco: 0000000B, Valor: 11
Endereco: 0000000C, Valor: 10
Endereco: 0000000D, Valor: 18
Endereco: 0000000E, Valor: 34
Endereco: 0000000F, Valor: 57
Endereco: 00000010, Valor: 255
Gravacao concluida. ----->Palavra 2 OK
Endereco: 00000010, Valor: 1
Endereco: 00000011, Valor: 2
Endereco: 00000012, Valor: 13
Endereco: 00000013, Valor: 11
Endereco: 00000014, Valor: 10
Endereco: 00000015, Valor: 18
FALHA DE ENERGIA ----->Palavra 3 não OK – A flag 255 desaparece
Gravação não concluída!!!!!!
```

4.1.3 Monitoração da precisão do relógio fornecido pelo RTC

Foi detectado um atraso no relógio do RTC de cerca de 1 segundo a cada 24 horas, isto se deve a falta de qualidade e precisão no cristal oscilador acoplado ao RTC de 32.768kHz.

4.1.4 Testes de compatibilidade com outros tipo de cartões SD

Foram efetuados testes em outros tipos de cartões SD e de menor capacidade, e todos foram compatíveis, inclusive cartões MMC de 512 Mbytes de capacidade. A limitação ocorreu para dispositivos com capacidade superior a 2Gbytes, que devido a possibilidade da escrita / leitura em blocos maiores que 512 bytes, não funcionaram corretamente.

4.1.5 Testes no Osciloscópio

Foram realizados simulações e medidas no osciloscópio com o objetivo de serem avaliados os sinais nas diversas etapas do dispositivo eletrônico.

Para a situação de tráfego normal, na entrada do dispositivo é obtido o seguinte sinal (com configuração do osciloscópio para 1V/div – Eixo Y e 500ns/div- Eixo X)

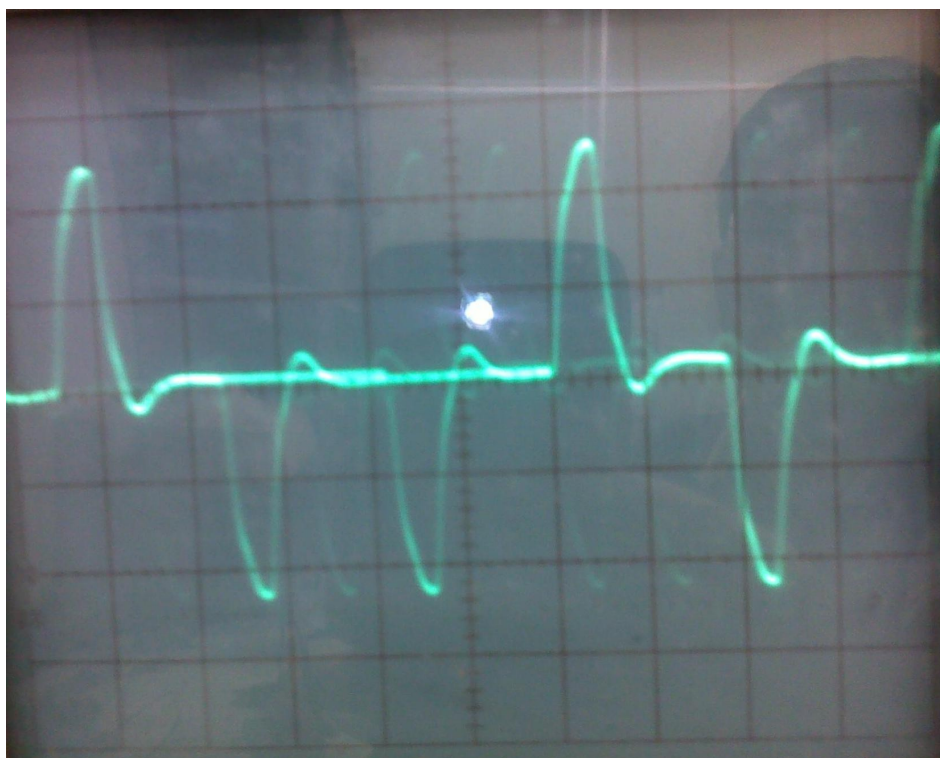


Figura 4-61 – Tela do osciloscópio para tráfego normal

Fonte: Elaborado pelo autor

Através do sinal da figura 4-2 podem-se constatar as características elétricas em um sinal trafegando com base na recomendação G.703, onde o valor de pico do sinal, tem-se + ou - 2,4 V aproximadamente, e a largura de cada pulso em torno de 244ns.

Na figura 4-3 abaixo é apresentado o sinal medido na saída do receiver diferencial SN75176, com um tráfego normal na interface, obteve-se a seguinte forma de onda (com configuração do osciloscópio para 2V/div – Eixo Y e 200ns/div- Eixo X)

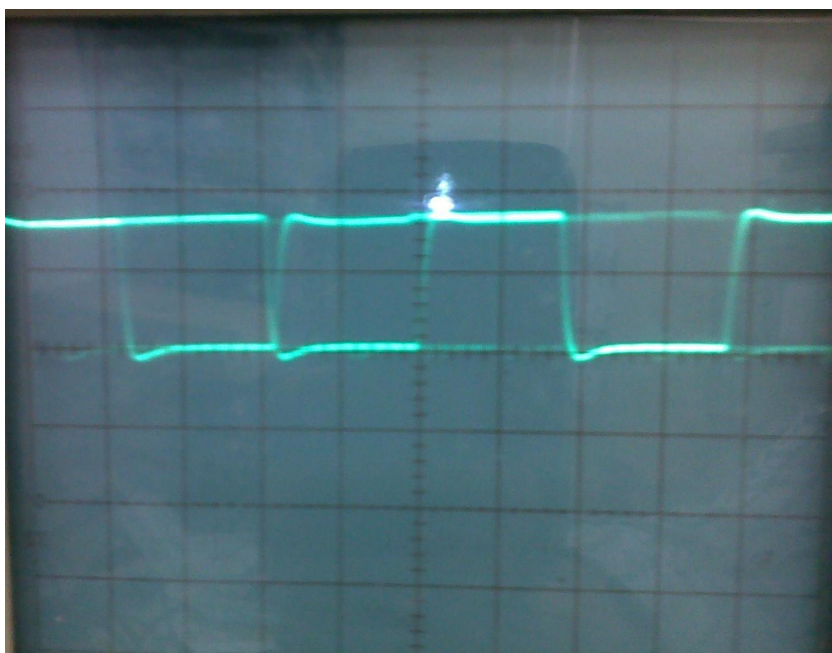


Figura 4-62 – Tela do osciloscópio na saída do receiver diferencial

Fonte: Elaborado pelo autor

Na figura 4-4 abaixo é apresentado o sinal medido na saída do circuito monoestável retriggerável, com um tráfego normal na interface, obteve-se a seguinte forma de onda (com configuração do osciloscópio para 2V/div – Eixo Y e 200ns/div- Eixo X)

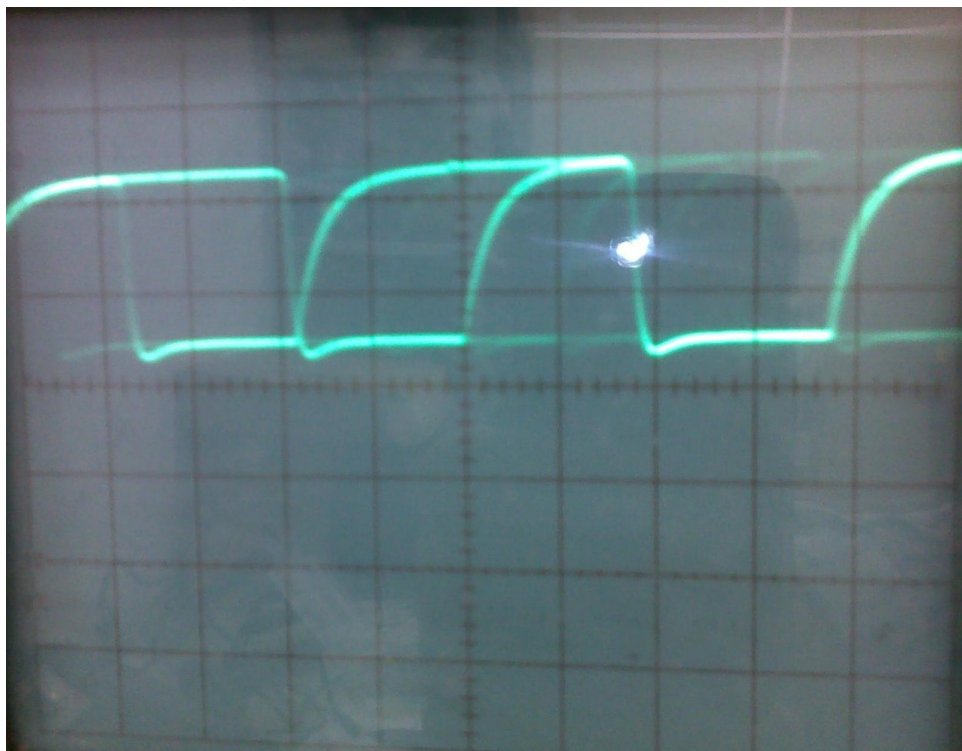


Figura 4-63 – Tela do osciloscópio na saída do monoestável retriggerável

Fonte: Elaborado pelo autor

Nota-se uma flutuação de aproximadamente 1 V neste sinal, isto se deve ao resistor de pull-up de $1k\Omega$, inserido na saída do circuito, que tem por finalidade ultrapassar os 4 V, não deixando dúvidas assim para a porta C0 do microcontrolador PIC 18F4520 a presença do sinal nível alto ou nível baixo presente oferecendo uma melhor performance para detecção de pulsos.

4.2. Metodologia dos Testes Integrados no Sistema Real

4.2.1 Testes Central PABX Fabricante Alcatel-Lucent

Para testes com uma central telefônica de fabricante Alcatel-Lucent em operação, foi utilizado um canal E1 de serviço, conectado a uma operadora X de telecomunicações. Para esta simulação não foi possível simular a falha AIS, pois a mesma deixaria indisponível grande parte de um sistema de transmissão ATM (155Mbps).

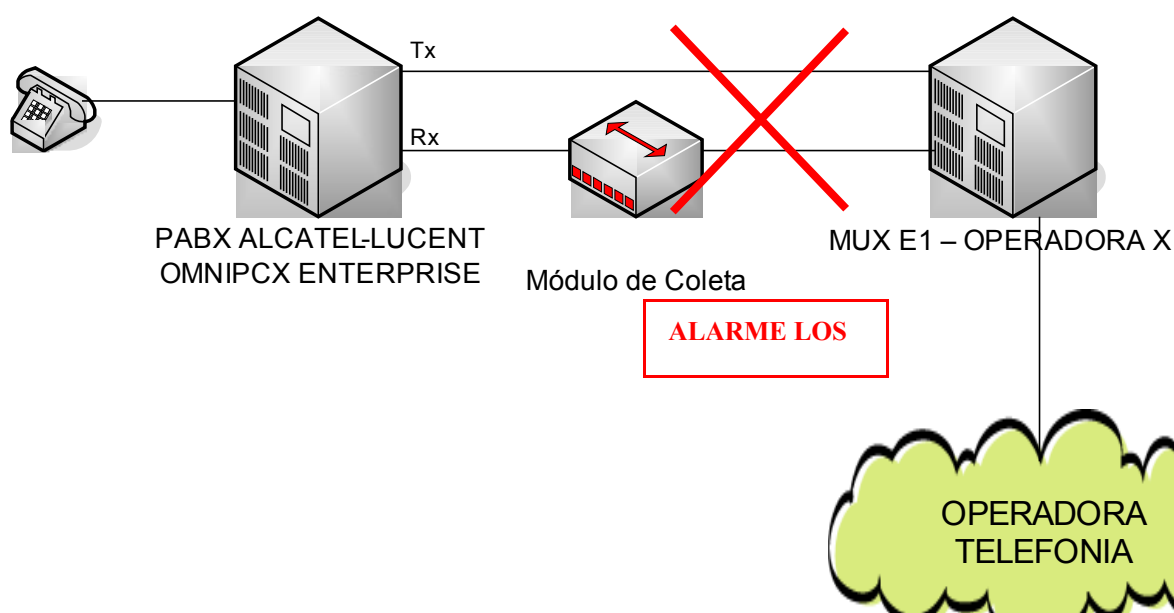


Figura 4-64– Topologia de testes com PABX fabricante Alcatel-Lucent

Fonte: Elaborado pelo autor

4.2.2 Testes Central PABX Fabricante Siemens

Para testes com uma central telefônica em funcionamento, foi utilizado um canal E1 conectado entre duas placas de E1, controladas por uma única central telefônica Siemens. Para esta simulação foi possível simular todos os tipos de falhas que o dispositivo está programado para detectar, pois o sistema utilizado encontra-se fora de operação. Estas falhas foram simuladas durante um período de 72 horas para a construção de uma maior base de dados, e assim obter a geração de dados históricos.

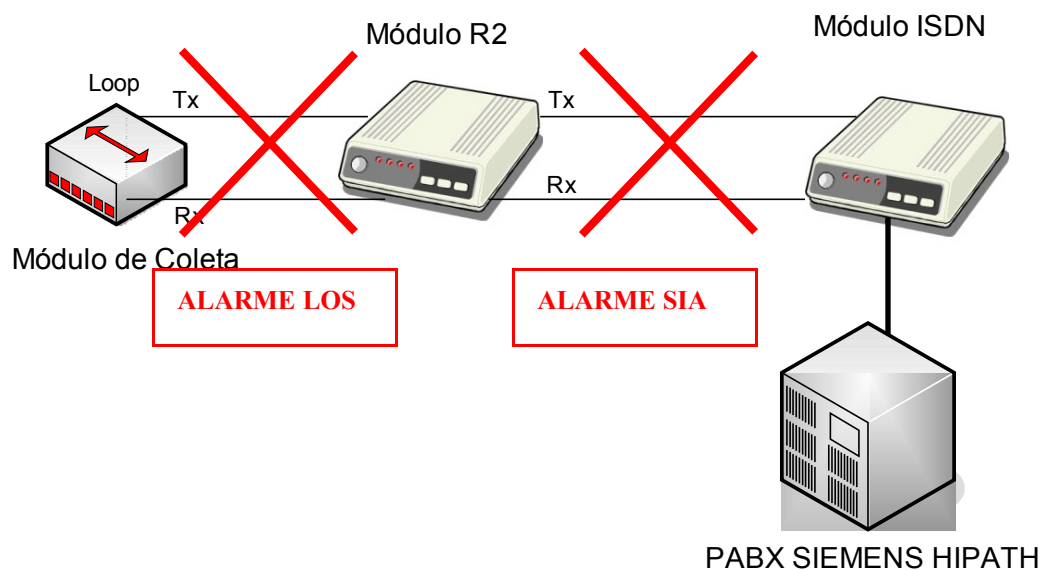


Figura 4-65 – Topologia de testes com PABX fabricante Siemens

Fonte: Elaborado pelo autor

4.3. *Comparações dos Resultados das Simulações, Modelos e o Sistema Real*

1.1.1. **Testes Central PABX Fabricante Alcatel-Lucent**



Figura 4-66 – Fotografia do dispositivo instalado em ambiente de produção

Fonte: Elaborado pelo autor

Na figura 4-8 abaixo é mostrada a tela de descarga dos dados gravados na memória SD Card destacando um alarme do tipo LOS detectado:

4	Total de Bytes Gravados no SD= 32
STATUS 1 TIPO 2 DETECTADO 13/11/2010 18:54:12	
STATUS 0 TIPO 0 DETECTADO 13/11/2010 18:56:56	
STATUS 1 TIPO 2 DETECTADO 13/11/2010 18:56:57	
STATUS 0 TIPO 0 DETECTADO 13/11/2010 18:57: 0	

Figura 4-67 – Tela com dados gravados no cartão SD para falha LOS Detectada

Fonte: Elaborado pelo autor

Na figura 4-9 abaixo é mostrada a tela de gerenciamento PABX Alcatel-Lucent destacando um alarme do tipo LOS detectado, com 1 segundo de diferença em relação ao dispositivo:

```

13/11/10 18:54:04 000020M 034/26/0/002 =3:2050=UA Terminal 2 Loss
13/11/10 18:54:08 000020M 016/03/0/004 =5:2053=Terminal 4 in service
13/11/10 18:54:13 000020M 036/08/0/000 =4:2112=SIA64 alarm on MIC/SYNC access
13/11/10 18:54:13 000020M 036/08/0/000 =4:2112=NOC alarm on MIC/SYNC access
13/11/10 18:54:13 000020M 036/08/0/000 =4:2112=LOS2M alarm on MIC/SYNC access
13/11/10 18:54:18 000020M 016/03/0/002 =5:2053=Terminal 2 in service
13/11/10 18:54:23 000020M 016/03/0/004 =3:2050=UA Terminal 4 Loss
13/11/10 18:54:23 000020M 036/08/0/000 =2:2101=MIC/SYNC access still in state of
alarm
13/11/10 18:54:28 000020M 016/03/0/002 =3:2050=UA Terminal 2 Loss
13/11/10 18:54:31 000020M 016/03/0/001 =5:2053=Terminal 1 in service
13/11/10 18:54:37 000020M 012/02/0/005 =3:2050=UA Terminal 5 Loss
13/11/10 18:54:49 000020M 016/03/0/001 =3:2050=UA Terminal 1 Loss
13/11/10 18:57:10 000020M 036/08/0/000 =5:2102=MIC/SYNC access back to normal

```

Figura 4-68– Tela com dados gravados PABX para falha LOS Detectada

Fonte: Elaborado pelo autor

Na figura 4-10 abaixo é mostrada a tela de descarga dos dados gravados na memória SD Card destacando um alarme do tipo LOS normalizado:

4	Total de Bytes Gravados no SD= 32
STATUS 1 TIPO 2 DETECTADO 13/11/2010 18:54:12	
STATUS 0 TIPO 0 DETECTADO 13/11/2010 18:56:56	
STATUS 1 TIPO 2 DETECTADO 13/11/2010 18:56:57	
STATUS 0 TIPO 0 DETECTADO 13/11/2010 18:57: 0	

Figura 4-69 – Tela com dados gravados SD Card para falha LOS normalizada

Fonte: Elaborado pelo autor

Na figura 4-11 abaixo é mostrada a tela de gerenciamento PABX Alcatel-Lucent destacando um alarme do tipo LOS normalizado, com 10 segundos de diferença em relação ao dispositivo desenvolvido:

```
13/11/10 18:54:04 000020M 034/26/0/002 =3:2050=UA Terminal 2 Loss
13/11/10 18:54:08 000020M 016/03/0/004 =5:2053=Terminal 4 in service
13/11/10 18:54:13 000020M 036/08/0/000 =4:2112=SLA64 alarm on MIC/SYNC access
13/11/10 18:54:13 000020M 036/08/0/000 =4:2112=NOS alarm on MIC/SYNC access
13/11/10 18:54:13 000020M 036/08/0/000 =4:2112=LOS2M alarm on MIC/SYNC access
13/11/10 18:54:16 000020M 016/03/0/002 =5:2053=Terminal 2 in service
13/11/10 18:54:23 000020M 016/03/0/004 =3:2050=UA Terminal 4 Loss
13/11/10 18:54:23 000020M 036/08/0/000 =2:2101=MIC/SYNC access still in state of
alarm
13/11/10 18:54:28 000020M 016/03/0/002 =3:2050=UA Terminal 2 Loss
13/11/10 18:54:31 000020M 016/03/0/001 =5:2053=Terminal 1 in service
13/11/10 18:54:37 000020M 012/02/0/005 =3:2050=UA Terminal 5 Loss
13/11/10 18:54:49 000020M 016/03/0/001 =3:2050=UA Terminal 1 Loss
13/11/10 18:57:10 000020M 036/08/0/000 =5:2102=MIC/SYNC access back to normal
```

Figura 4-70 – Tela com dados gravados no PABX para falha LOS normalizada

Fonte: Elaborado pelo autor

4.3.2 Testes Central PABX Fabricante Siemens



Figura 4-71 – Fotografia do dispositivo instalado em ambiente de produção

Fonte: Elaborado pelo autor

Como foram simuladas falhas durante um período de 72 horas, pode-se extrair relatórios gráficos com a quantidade de falhas ocorridas, a partir da macro em Excel desenvolvida previamente:

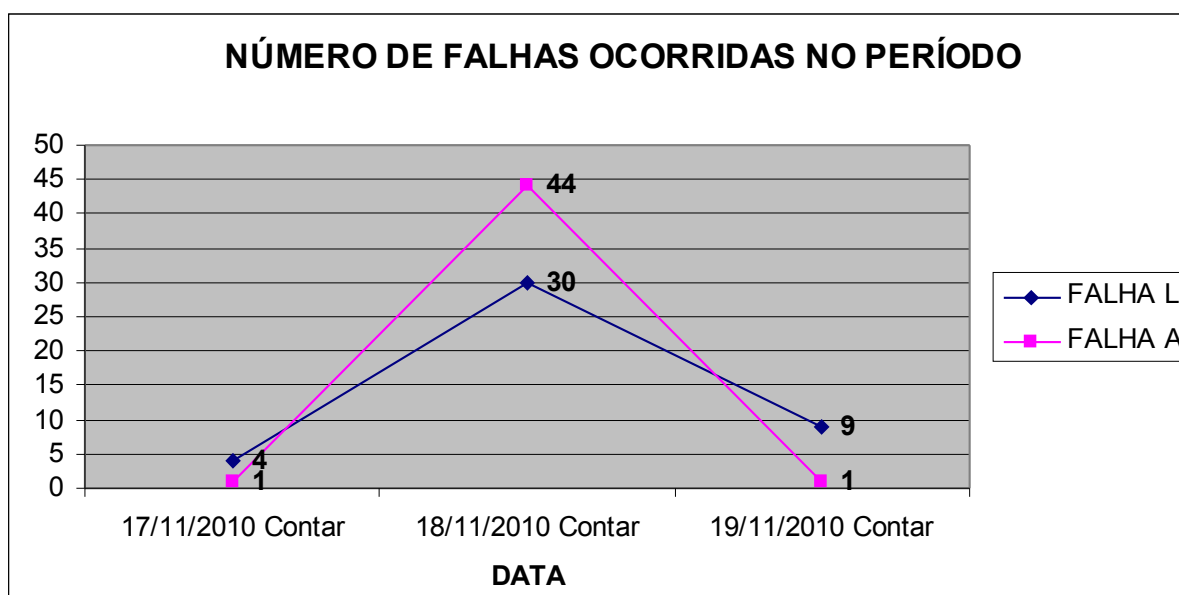


Figura 4-72 – Gráfico quantitativo de falhas avaliadas por 72 horas

Fonte: Elaborado pelo autor

Além dos indicadores do número de falhas ocorridas no período, como o dispositivo fornece informações detalhadas de tempo, foi possível a extração de indicadores de tempo de indisponibilidade por falha, conforme a figura 4-14 abaixo:

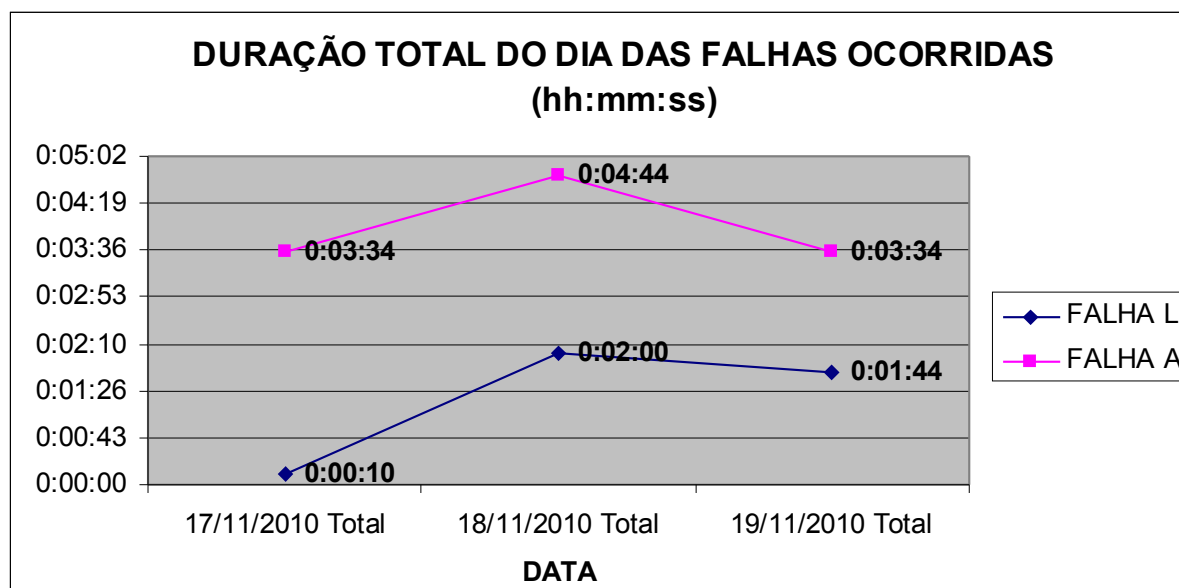


Figura 4-73 – Gráfico qualitativo das falhas avaliadas por 72 horas

Fonte: Elaborado pelo autor

5. CONSIDERAÇÕES FINAIS

5.1. *Problemas Encontrados*

Os problemas encontrados no projeto foram em sua maioria sempre relacionados ao hardware do protótipo, ou mesmo adequação da tensão de funcionamento da memória SD Card, pois a mesma opera com alimentação de 3.3 VCC e o microcontrolador e demais dispositivos com 5 VCC. Este problema de adequação somente foi possível com a utilização das portas C do microcontrolador PIC. Isto devido as mesmas serem dotadas de circuitos baseados em topologia do tipo *Schmidt Trigger*, fazendo com que menores amplitudes de sinal possam já ser interpretadas como sinal de nível lógico alto na entrada do microcontrolador PIC.

Houveram problemas de ruídos e interferências na linha de dados da memória SD Card, que foi solucionado com o uso de cabos do tipo flat cable de melhor qualidade construtiva bem como a montagem do dispositivo em caixa metálica com a carcaça aterrada.

5.2. *Avaliação dos Objetivos Propostos*

Praticamente todos os objetivos do presente trabalho foram alcançados. Foi removido somente do escopo inicial do projeto o desenvolvimento de uma interface específica para efetuar a descarga dos eventos via serial, visto a mesma precisar ser desenvolvida em plataforma Windows. Além disso, existem vários softwares emuladores seriais do tipo *Hyperterminal* do Windows, que realizam a mesma tarefa, que facilmente podem ser configurados para geração dos arquivos no formato txt e possivelmente geração de relatórios em formato MS Excel.

5.3. *Avaliação dos Índices de Qualidade*

O sistema projetado e desenvolvido mostrou uma exatidão em quesitos de tempo. Conforme constatado nos testes realizados na seção 4.3.1, os atrasos para a detecção de falha ou normalização foram inferiores aos equipamentos PABX existentes. Isto se deve ao fato do processamento desses dispositivos ser compartilhado por uma infinidade de outras tarefas, visto que o módulo de coleta desenvolvido possui processamento exclusivo a detecção arquivamento de falhas.

5.4. Avaliação do Cronograma Previsto e Executado

Todos os atrasos decorrentes no projeto foram devidos a falhas de hardwares relativas a memória SD Card não previstas durante a fase de planejamento, visto ser uma tecnologia nova, com pouca literatura disponível e poucos recursos na internet.

5.5. Conclusões

Os resultados mostraram que o método utilizado para a detecção de falhas foi muito confiável e eficiente, mesmo quando o link E1 apresenta ocupação de vários canais de voz simultâneos. O método utilizado para a gravação dos bytes na memória *SD Card* não foi eficiente no caso de haver alguma falta de energia que ocorra durante o processo de gravação, ocasionando uma perda na ordem adequada dos dados gravados. Este problema ocorre porque o dispositivo não consegue gravar a flag 255 no último byte da palavra de informações. Este problema pode ser evitado facilmente com a utilização de alguma bateria externa ao dispositivo.

Também foi verificado neste projeto que a manipulação de um dispositivo de tecnologia de ponta, como o cartão SD, requer uma preocupação extra desde a confecção correta da placa de circuito, até no manuseio correto dos componentes eletrônicos utilizados.

O dispositivo desenvolvido mostrou-se eficaz para o que foi projetado, pois as memórias do tipo *SD Card* possuem grande confiabilidade na segurança e integridade dos dados armazenados, sendo projetadas para serem robustas fisicamente, sendo que dificilmente as mesmas possam apresentar setores de memória corrompidos.



5.6. Sugestões para Trabalhos Futuros

- Arquivamento dos dados gravados em formato FAT 16 do Windows, com este tipo de gravação é possível os dados serem extraídos do *SD Card* diretamente de um PC com *Windows* instalado.
- Detecção de outros tipos de falhas em links com interface G703
- Detecção de falhas em links baseados em tecnologia TCP-IP como exemplo recomendações G.729 ou H.323.
- Gravação/leitura em memórias SD com capacidades superiores a 2Gb

6. REFERÊNCIAS

FOROUZAN, Behrouz A. – Comunicação de dados e redes de computadores – 4ª. Ed. – São Paulo:McGraw-Hill,2008

PINES, José; BARRADAS, Ovidio César Machado - Sistemas Multiplex - 2a. Ed. – Rio de Janeiro: Livros Técnicos e Científicos Editora, 1978

LATHI,P.B.. Sistemas de Comunicação 1a. Ed - Rio de Janeiro: Editora Guanabara, 1987

KRAUS, John D. – Eletromagnetismo – 2ª. Ed. – Rio de Janeiro: Editora Guanabara,1986

ROCHOL, Jürgen - Redes de Computadores – 1ª.Ed - Porto Alegre: Universidade Federal do Rio Grande do Sul. Instituto de Informática, 1999.

ITU-T (“Telecommunication Standardization Sector of ITU”). “G.703 - Physical/ electrical characteristics of hierarchical digital interfaces”. Genebra, 2001.

Disponível em: < <http://www.itu.int/ITU-T/publications/recs.html>> Acesso em: 10/10/10

ITU-T (“Telecommunication Standardization Sector of ITU”). “G.775 - Loss of Signal (LOS), Alarm Indication Signal (AIS) and Remote Defect Indication (RDI) defect detection and clearance criteria for PDH signals”. Genebra, 1998. Disponível em: < <http://www.itu.int/ITU-T/publications/recs.html>> Acesso em: 10/10/10

Technical Committee SD Card Association “SD Specifications - Part 1 -Physical Layer - Simplified Specification - Version 3.01”. San Ramon, 2010. Disponível em: <<http://www.sdcard.org>> Acesso em: 10/2010

Technical Committee SD Card Association “SD Specifications Part A2 SD Host Controller Simplified Specification Version 2.00”. San Ramon, 2007. Disponível em: <<http://www.sdcard.org>> Acesso em: 10/2010



OBRAS CONSULTADAS

SILVA, Renato A – Programando microcontroladores PIC: Linguagem C - 1a. Ed. – São Paulo:Ensino Profissional Editora, 2006

PEREIRA, Fábio – Microcontroladores PIC:Programação em C – 7ª. Ed. – São Paulo: Editora Érica, 2007

IBRAHIM, Dogan- Advanced PIC microcontroller projects in C: from USB to RTOS with the PIC18F series – 1a Ed. – USA:Elsevier,2008

COCIAN, Luis Fernando Espinosa – Engenharia de processamento digital II – 1a Ed – Canoas: Ed. ULBRA,2006

APÊNDICE A – CÓDIGO FONTE EM LINGUAGEM C

```
#include <18f4520.h>
#fuses HS,NOLVP,NOWDT,NOPROTECT
#use delay(clock=8000000)
#use rs232(baud=9600, UART1)
#include <stdlib.h> // for atoi32
#use fast_io(c)
#define MMCSD_PIN_SCL    PIN_C3 //Saída para SD
#define MMCSD_PIN_SDI    PIN_C4 //Entrada vinda do SD
#define MMCSD_PIN_SDO    PIN_C5 //Saída para SD
#define MMCSD_PIN_SELECT PIN_C2 //Saída para SD
#include "mmc_sd_1.c" //Biblioteca modificada para func. SD Card
#include <input.c> //Biblioteca para comunicação via serial (comandos)
#include <DS1302.c> //Biblioteca para funcionamento RTC - DS 1302
#include <lcd_4B.c> //Biblioteca para LCD 2 linhas x 16 colunas

//Variáveis Globais
int valor,cmd,i,cont,status,falha;
BYTE day,mth,year,dow,hour,min,sec;
int32 endereco,endereconew=0x0000,contador=0;

void escreve_mem_manual() // Função utilizada para leitura/ escrita manual no
SD Card
{
    BYTE value, cmd;
    int32 address;
    do {
        do {
            printf("\r\n");
            printf("\r\nEscrever ou Ler no cartão: ");
            cmd=getc();
            cmd=toupper(cmd);
            putc(cmd);
        } while ( (cmd!='E') && (cmd!='L') );

        printf("\r\nLocal: ");
        address = gethex();
        address = (address<<8)+gethex();
        if(cmd=='L')
        {
            mmc_sd_read_byte(address, &value);
            printf("\r\nValor: %X\r\n", value);
        }
        if(cmd=='E') {
            printf("\r\nNovo valor: ");
            value = gethex();
            printf("\r\n");
            mmc_sd_write_byte(address, value);
            mmc_sd_flush_buffer();
        }
    } while (1);
}
```

```
    }  
    } while (TRUE);  
}  
  
void set_clock() // Função utilizada para setar o horário correto no RTC  
{  
    day=13;  
    mth=11;  
    year=10;  
    dow=0;  
    hour=18;  
    min=52;  
    rtc_set_datetime(day,mth,year,dow,hour,min);  
}  
  
/*  
int posicao;  
int vetor [5];  
{  
    printf("\r\nDigite dia,mes,ano,hh,mm");  
    for (posicao=0;posicao<=4;posicao++)  
    {  
        printf("\r\nDigite:");  
        vetor[posicao]=getc();  
        putc(vetor[posicao]);  
        vetor [0]=day;  
        vetor [1]=mth;  
        vetor [2]=year;  
        vetor [3]=hour;  
        vetor [4]=min;  
    }  
*/  
  
void carrega_dados() //Função utilizada para carregador data/hora  
{  
    rtc_get_date( day, mth, year, dow);  
    rtc_get_time( hour, min, sec );  
    rm_bcd(sec); //Função para converter BYTE para BCD  
    rm_bcd(min); //Função para converter BYTE para BCD  
    rm_bcd(hour); //Função para converter BYTE para BCD  
    rm_bcd(year); //Função para converter BYTE para BCD  
    rm_bcd(mth); //Função para converter BYTE para BCD  
    rm_bcd(day); //Função para converter BYTE para BCD  
}  
  
void escreve_SD() //Grava a palavra de 8 bytes  
(status,falha,dia,mes,ano,hora,min,seg) no SD, deslocando a flag "255" para 8  
posições de memória adiante  
  
{  
    int posicao=0,flag=255;  
    int vetor [9];  
    {  
        vetor [0]=status;
```

```
vetor [1]=falha;
vetor [2]=day;
vetor [3]=mth;
vetor [4]=year;
vetor [5]=hour;
vetor [6]=min;
vetor [7]=sec;
vetor [8]=flag;

for (endereco=endereconew;endereco<=endereconew+0x0008;endereco++)
{
    carrega_dados(); // Busca informações do RTC data e hora
    printf("\r\nEndereco: %8X, Valor: %u",endereco, vetor[posicao]);
    valor=vetor[posicao];
    mmcsd_write_byte(endereco, valor);
    delay_us(1);
    posicao++;
}
mmcsd_flush_buffer();
}

void leitura_SD_cont() //Leitura inicial do SD para calcular a qtde de bytes já
gravados e na próxima não gravar por cima
{
    endereco=0x0000;
    contador=0;
    do
    {
        mmcsd_read_byte(endereco, &valor); //Função para ler byte no endereço e
retorna no ponteiro valor
        contador++; //A cada leitura, incrementa a variável global contador
        endereco++; //A cada leitura, incrementa a variável global endereco (int32)
    }while(valor!=255); // Realiza a leitura até que o ponteiro valor seja 255
(flag)
        endereconew=(contador-1); // Faz com que o novo endereco a ser gravado
seja igual ao último salvo menos 1 (sobrescrevendo assim a flag 255)
        printf("\r\nTotal de Bytes Gravados no SD= %lu \r\n",(contador-1));
//Após uma leitura completa informa qtde bytes já gravados

}

void descarrega_SD() //Descarrega SD em formato de relatórios e após limpa toda a
memória
{

    endereconew=0x0000;
    do{
        int posicao=0;
        int vetor [8];

        for (endereco=endereconew;endereco<=endereconew+0x0007;endereco++)
        {
            mmcsd_read_byte(endereco, &valor);
            vetor[posicao]=valor;
```

```
        posicao++;
        delay_ms(1);
        contador--;
    }

    printf("\r\nSTATUS %u TIPO %u DETECTADO %2u/%2u/20%2u %2u:
%2u:%2u \n\r",vetor[0],vetor[1],vetor[2],vetor[3],vetor[4],vetor[5],vetor[6],vetor[7]);
    endereconew=endereco;
    }while (contador!=1); // Vai decrementado a variável contador até que ela
seja 1, ai pára;

    }

void analisa_falha_tipo()
{
    cont=0;
    set_timer1(1);
    setup_timer_1(T1_EXTERNAL);
    delay_ms(20);
    cont=get_timer1();
    setup_timer_1(T1_DISABLED);

    if(cont<=10)
    {
        if (!input(PIN_C0))
        {
            falha=1; ///Alarme SIA detectado
            printf(lcd_escreve,"\f## ALARME AIS ## ");

        }
        else
        {
            falha=2; ///Alarme LOS detectado
            printf(lcd_escreve,"\f## ALARME LOS ## ");
        }
    }
    else
    {
        falha=0; //Linha OK sem falhas
    }

}

void analisa_falha()
{
    do
    {
        do
        {
            printf(lcd_escreve,"\f INTERFACE OK");
            carrega_dados();
```

```
printf(lcd_escreve, "\n%2u/%2u/%u  %2u:
%2u", day, mth, year, hour, min);
    delay_ms(700);
    analisa_falha_tipo();
    }while(cont >= 10);

    status = 1;
    escreve_SD();
    endereconew = endereco;
    endereconew--;
    printf("\r\n Gravacao concluida.");

    do
    {
        carrega_dados();
        analisa_falha_tipo();
        printf(lcd_escreve, "\n%2u/%2u/%u  %2u:
%2u", day, mth, year, hour, min);
        delay_ms(300);
        printf(lcd_escreve, "\f");
        delay_ms(300);
        }while (cont <= 10);

        status = 0;
        falha = 0;
        carrega_dados();
        escreve_SD();
        endereconew = endereco;
        endereconew--;
        printf("\r\n Gravacao concluida.");

    }while(TRUE);

}

void leitura_SD() // Lê a palavra de 8 bytes (status, falha, dia, mes, ano, hora, min, seg)
e imprime na tela até chegar na flag de último byte gravado
{

    endereco = 0x0000;

    do{
        mmc_sd_read_byte(endereco, &valor);
        printf("\r\nEndereco %8X, Valor: %u", endereco, valor);
        delay_us(1);
        endereco++;
        }while (valor != 255);

    }

void limpa_SD() // Zera todas as posições de memória até a posição que foi gravada
{

    endereco = 0x0000;
```

```
do{

    mmcsd_write_byte(endereco,0);
    endereco++;
    mmcsd_read_byte(endereco, &valor);
}while (valor!=255);

mmcsd_write_byte(endereco,0);
delay_us(0.1);

mmcsd_write_byte(0x0000,255);
mmcsd_flush_buffer(); // Função utilizada para descarregar o buffer de
gravação SD. Como está setado para 512 bytes, sempre precisa descarrega
delay_us(0.1);
printf("\r\nTodos enderecos apagados.");
leitura_SD_cont();
}

void verifica_presenca_SD() // Função criada para não deixar o sistema operar sem
cartão. A função de inicialização do CCS as vezes com ruído no pino RC4 nao
funciona corretamente
{
if (!input(PIN_C4))
do{
printf("SD Card nao encontrado\r\n");
printf("Insira um SD valido e reinicie o sistema\r\n");
printf(lcd_escreve,"\f SD INVALIDO");
delay_ms(5000);

}while (TRUE);
}

void main(void)
{
    lcd_ini();
    printf(lcd_escreve,"\fINICIALIZANDO..");
    delay_ms(1000);
    printf("Iniciando SD...\r\n");

    if (mmcsd_init()) // Função que envia o CMD0 para o cartão e aguarda 100
pulsos de clock até sair da condição de IDLE_STATE
    {
        printf("Cartao SD nao inicializado!!!!\r\n");
        printf(lcd_escreve,"\f SD INVALIDO");
        while(TRUE);
    }

    verifica_presenca_SD();

    printf("Cartao SD Inicializado com sucesso!!!!\r\n");
    delay_ms(1000);

    rtc_init(); // Função que inicializa o RTC DS 1302 para receber ou enviar
comandos via SPI
```

```
printf("RTC DS 1302 inicializado com sucesso!!!!\r\n");
delay_ms(1000);
printf("Display inicializado com sucesso!!!!\r\n");
delay_ms(2000);
printf(lcd_escreve, "\f LENDO SD CARD...");
leitura_SD_cont(); //

if(!input(PIN_D0)) // Validação para verificar se chave frontal (manut/coleta)
{
    analisa_falha();
}

for(i=0;i<5;i++) //Loop para mostrar no display que a opção manutenção foi
selecionada.
{
    printf(lcd_escreve, "\f #MANUTENCAO# ");
    delay_ms(500);
    printf(lcd_escreve, "\f");
    delay_ms(500);
}

do{

    do{
        //Menu Principal - quando é selecionada opção manutenção na Chave
        //frontal. Deverá estar conectado um cabo serial no PC para interação
        do {
            printf(lcd_escreve, "\f MENU PRINCIPAL");
            printf("\r\n");
            printf("\r\n##### COLETOR DE FALHAS EM LINKS E1
#####");
            printf("\r\n");
            printf("\r\nDigite uma das opcoes abaixo:");
            printf("\r\n");
            printf("\r\n 1. 'Para escrever/ler manualmente no SD'");
            printf("\r\n");
            printf("\r\n 2. 'Para leitura do SD Card'");
            printf("\r\n");
            printf("\r\n 3. 'Para clear (limpeza) do SD Card'");
            printf("\r\n");
            printf("\r\n 4. 'Para descarregar e limpar SD Card'");
            printf("\r\n");
            printf("\r\n 5. 'Para consultar o status da memoria do SD Card'");
            printf("\r\n");
            printf("\r\n 6. 'Para consultar o registrador CID do SD Card'");
            printf("\r\n");
            printf("\r\n 7. 'Para consultar o registrador CSD do SD Card'");
            printf("\r\n");
            printf("\r\n 8. 'Ajustar a data e hora do sistema'");
            printf("\r\n");
            printf("\r\n 9. 'Sobre'");
            printf("\r\n");
            printf("\r\n Q. 'Para retornar ao modo de coleta'");
            printf("\r\n");
```

```
cmd=getc();
cmd=toupper(cmd);
putc(cmd);
} while ( (cmd!='1')&&(cmd!='2')&&(cmd!='3')&&(cmd!='4')&&(cmd!='5')
&&(cmd!='6')&&(cmd!='7')&&(cmd!='8')&&(cmd!='9')&&(cmd!='Q'));

if(cmd=='1')
{
    escreve_mem_manual();
}

if(cmd=='2') //Lê todos os endereços do SD Card até o endereço que contiver
a flag 255
{
    delay_ms(500);
    printf("\r\nLendo dados do cartao...");
    printf(lcd_escreve,"\fLENDO SD CARD...");
    leitura_SD();
    printf("\r\nLeitura concluida.");
    printf(lcd_escreve,"\fLEITURA OK!!");
}

if(cmd=='3') //Limpa toda a memória (coloca 0 em todos endereços até a flag
255)
{
    printf("\r\nApagando toda memoria do SD...");
    printf(lcd_escreve,"\f APAGANDO SD...");
    limpa_SD();
    printf(lcd_escreve,"\f SD LIMPO!");
}

if(cmd=='4')

{
    leitura_SD_cont();
    if(contador>1)
    {
        printf(lcd_escreve,"\fDESCARREGANDO...");
        delay_ms(500);
        descarrega_SD();
        printf(lcd_escreve,"\f SD DESCARREG!!");
        delay_ms(2000);
        printf(lcd_escreve,"\f LIMPANDO SD...");
        limpa_SD();
        printf(lcd_escreve,"\f SD LIMPO!");
    }
    else
    {
        printf("\r\n Sem eventos para descarregar");
    }
}

}
```

```
if(cmd=='5') // Função que retorna o número de bytes salvos no cartão SD
{
    leitura_SD_cont();
}

if(cmd=='6') //Função para consultar o registrador CID do SD Card
{
    do
    {
        mmcsd_print_cid();
    }while(FALSE);
}

if(cmd=='7') //Função que retorna o registrador CSD do SD Card
{
    do
    {
        mmcsd_print_csd();
    }while(FALSE);
}

if(cmd=='8')
{
    set_clock(); //Função para acertar a hora do RTC Manualmente
}

if(cmd=='9')
{
    printf("\r\n");
    printf("\r\nDesenvolvido por Andre Oliveira - Out/2010");
    printf("\r\n");
    printf("\r\nTrabalho de Conclusao de Curso Engenharia Eletrica");
    printf("\r\n");
    printf("\r\nULBRA - Universidade Luterana do Brasil - Canoas - RS");
    printf("\r\n");
}

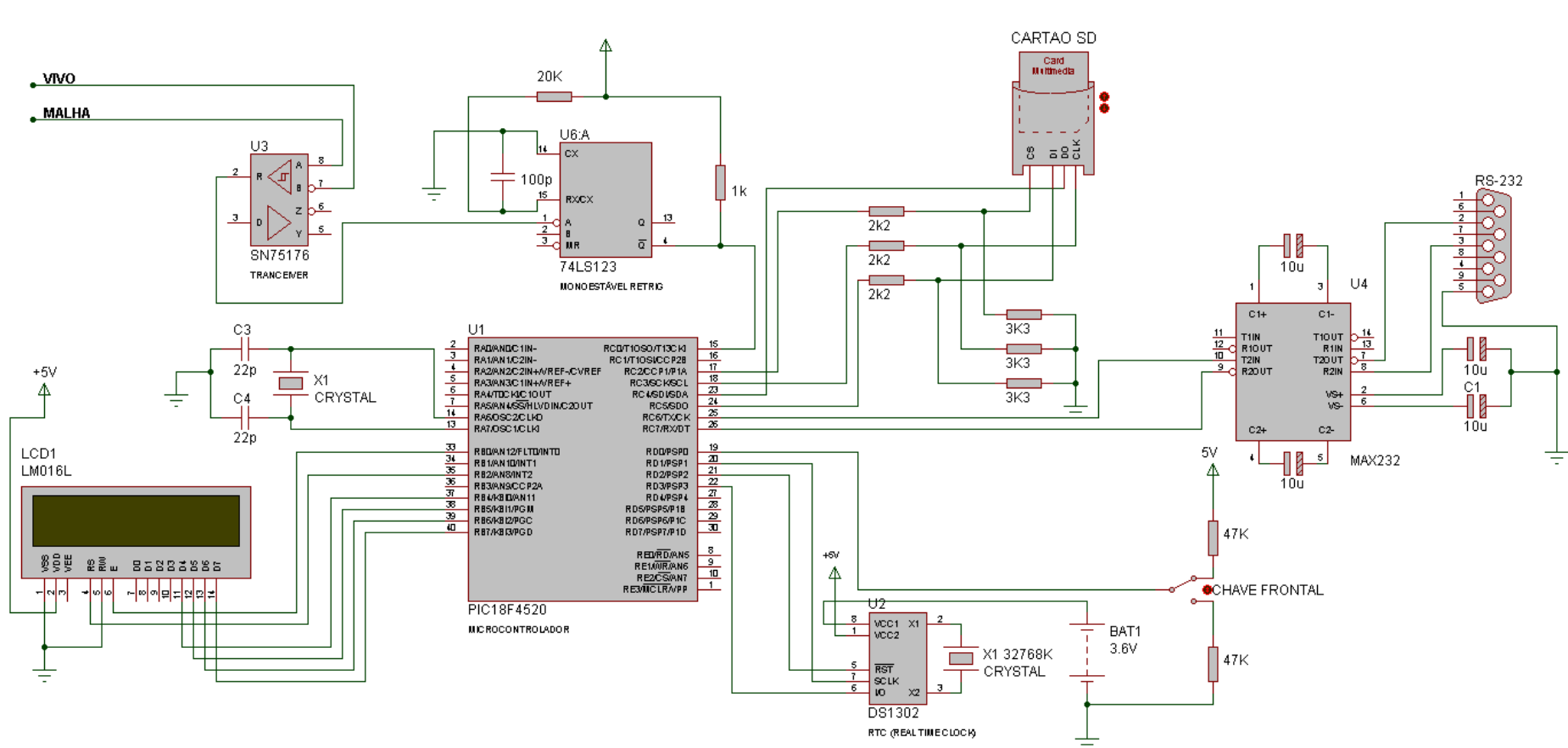
if(cmd=='Q')
{
    analisa_falha(); //Opção utilizada para entrar no modo normal de coleta.
}

cmd=getc();
cmd=toupper(cmd);
putc(cmd);

}while (cmd=='q');
}while(TRUE);
}
```



APÊNDICE B – DIAGRAMA ESQUEMÁTICO DO DISPOSITIVO



APÊNDICE C – FOTO DO PROTÓTIPO DESENVOLVIDO

