UNIVERSIDADE ESTADUAL PAULISTA "JÚLIO DE MESQUITA FILHO"
CAMPUS DE ILHA SOLTEIRA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

IMPLEMENTAÇÃO DE UM MÓDULO DE INTERFACE PARA TRANSDUTORES INTELIGENTES (STIM), IEEE 1451.2, UTILIZANDO UM MICROCONTROLADOR.

Eduardo Marques Martins

Dissertação de Mestrado submetida à Faculdade de Engenharia de Ilha Solteira da Universidade Estadual Paulista "Julio de Mesquita Filho", Departamento de Engenharia Elétrica, como parte dos requisitos necessários para obtenção do título de Mestre em Engenharia Elétrica.

Orientador: Prof. Dr. Aparecido Augusto de Carvalho

Co-Orientador: Prof. Dr. Alexandre César Rodrigues da Silva

ILHA SOLTEIRA – SP, MAIO DE 2005

RESUMO

O objetivo deste trabalho é a implementação de um STIM (Módulo de Interface para Transdutores Inteligentes) de uma forma funcional e econômica, seguindo as características do padrão de interfaceamento IEEE 1451.2. Para esta finalidade é escolhido um microcontrolador MC68HC908QY4, de baixo custo, da Motorola. O STIM implementado possui 2 canais de entradas analógicas de 8 bits e 2 canais de saídas digitais, sendo que nas entradas analógicas são conectados sensores de temperatura e nas saídas digitais dois ventiladores. Para testar o STIM utiliza-se um NCAP (Processador de Aplicação com Capacidade de Operar em Rede) IEEE 1451.1, baseado em PC e em FPGA de uso geral, desenvolvido pelo Grupo de Instrumentação Distribuída da UNESP- Ilha Solteira, sendo parte do *software* desenvolvida através da tecnologia Java. Em um exemplo de aplicação considera-se uma câmara fria na qual a temperatura não pode exceder um certo valor visando a conservação do produto. Considera-se um sensor de temperatura 1 próximo a um ventilador 1 e um sensor de temperatura 2 próximo a um ventilador 2. Quando o sensor de temperatura 1 indica temperatura elevada o ventilador 1 é acionado forçando uma queda de temperatura na região, o mesmo ocorre com o conjunto 2. Uma vantagem deste sistema é que o operador pode, além de deixar o sistema da câmara funcionando, ler os dados do sensor de forma remota e interligá-lo com outros, através da rede, conseguindo uma visão global de todos os processos importantes que ocorrem em um processo. Outra vantagem está relacionada com a característica "*plug and play*": desacopla-se um STIM, desenvolvido pelo Grupo de Instrumentação Distribuída, do NCAP e conecta-se o STIM implementado neste trabalho, observando-se a operabilidade do sistema. O campo de aplicações do sistema descrito é muito vasto. Neste aspecto o fato de se conseguir realizar uma implementação 1451.2 de baixo custo apresenta-se como uma contribuição para esta área de pesquisa.

Palavras chave: Padrão IEEE 1451, interfaceamento, STIM, microcontrolador.

ABSTRACT

The aim of this work is the implementation of a functional and economical STIM (Smart Transducer Interface Module) following the characteristics of the IEEE 1451.2 Interfacing Standard. In order to achieve this goal, a low cost Motorola microcontroller is chosen. The implemented STIM has two 8-bit analogical input channels and two digital output channels. Two temperature sensors are connected in the analogical inputs and two fans are connected in the digital outputs. To test the STIM a NCAP (Network Capable Application Processor) IEEE 1451.1 is used, based on PC and in FPGA of general use, developed by UNESP-Ilha Solteira Distributed Instrumentation Group, and the software is developed in Java programming language. As an example of application, a cold chamber in which the temperature cannot exceed a certain value, in order to conserve a product, is chosen. A temperature sensor 1 is considered close to a fan 1 and a temperature sensor 2 close to a fan 2. When the temperature sensor 1 indicates high temperature, the fan 1 operate, forcing a temperature decrease in the surrounding area. The same happens with group 2. An advantage of this system is the fact that the operator is able to make the system in the chamber work and read remotely the sensor data. Furthermore it is possible to connect other sensors through the network in order to get a global view of all important processes that happen inside a company. Another advantage is the “plug and play” characteristic of the system. The STIM developed in this work replaced a STIM constructed with FPGA devices without any problem. The application field of the implemented STIM is large. In this sense the implementation of a low cost 1451.2 is a contribution to this research´s field.

KeyWords: Standard IEEE 1451, interface, STIM, microcontroller.

LISTA DE FIGURAS

Figura 2.1 – Rede de transdutores..... 6
Figura 2.2 – Diagrama de blocos de um transdutor conectado a uma rede..... 7
Figura 2.3 – Barramento de sensores..... 8
Figura 3.1 – Concepção básica de um sensor inteligente..... 10
Figura 3.2 – Exemplo de diferentes formas de implementar um módulo STIM..... 11
Figura 3.3 – Estrutura de um endereço completo..... 13
Figura 3.4 – Função de disparo (IEEE 1451.2)..... 14
Figura 3.5 – Registrador de estado..... 16
Figura 3.6 – Interface TII entre STIM e NCAP..... 18
Figura 3.7 – TII: Protocolo geral de transferência de dados..... 22
Figura 3.8 – Protocolo que implementa a função de disparo / diagrama de tempos..... 23
Figura 3.9 – Formas de onda dos protocolos de transferência de dados..... 25
Figura 4.1 – Foto do Microcontrolador MC68HC908QY4..... 29
Figura 4.2 – Pinagem do Microcontrolador MC68HC908QY4..... 31
Figura 4.3 – Foto da placa de desenvolvimento M68EVB908Q®..... 32
Figura 5.1 – Protocolo 1451.2 entre o gerenciador de protocolos e STIM..... 34
Figura 5.2 – Diagrama da aplicação..... 34
Figura 5.3 – Diagrama de blocos: STIM-NCAP-REDE..... 35
Figura 5.4 – Fluxograma de inicialização e reconhecimento do STIM..... 40
Figura 5.5 – Fluxograma de habilitação e leitura da função e canal..... 41
Figura 5.6 – Fluxograma de desvio para a função requerida..... 41
Figura 5.7 – Fluxograma de leitura de dados..... 42
Figura 5.8 – Fluxograma de leitura de dados(TEDS)..... 43
Figura 5.9 – Fluxograma de escrita de dados..... 44
Figura 6.1 – Sinal de Disparo(NTRIG) e de Reconhecimento(NACK)..... 45
Figura 6.2 – Sinal de Habilitação(NIOE) e de Reconhecimento(NACK)..... 46
Figura 6.3 – Sinal de sincronismo(DCLK) e de Limite de pacote de dados(NACK)..... 47
Figura 6.4 – Sinal DIN e de Limite de pacote de dados(NACK)..... 47
Figura 6.5 – Sinal DOUT e NACK para entrada analógica em 0V..... 48
Figura 6.6 – Sinal DOUT e NACK para entrada analógica em 1,53V..... 48
Figura 6.7 – Sinal NACK e DOUT para leitura de TEDS..... 49
Figura 6.8 – Sinal NACK e DOUT para leitura de TEDS..... 50
Figura 6.9 – Sinal NACK e Saída Digital 2..... 51
Figura 6.10 – Montagem realizada para simular STIM..... 51
Figura 6.11 – Microcontrolador, gerenciador de protocolos e placas de sensor e atuador..... 52

LISTA DE TABELAS

Tabela 3.1 - Definição das linhas da TII – IEEE 1451.2..... 19
Tabela 3.2 - Exibe a atribuição de pinos e cores, recomendada pelo padrão..... 20
Tabela 5.1 – Tabela de descrição de sinais da Interface Independente para Transdutores(TII)..... 36
Tabela 5.2 – Tabela de descrição das entradas analógicas..... 36
Tabela 5.3 – Tabela de descrição das saídas digitais..... 36
Tabela 5.4 – Tabela de descrição das funções adicionais..... 37
Tabela 5.5 – Taxa de divisão do relógio ADC 39
Tabela A1: Endereços de função, globais..... 58
Tabela A2: Endereços de função (canais 1 até 255)..... 58
Tabela B1: Representação de Unidades IEEE 1451.2:..... 60
Tabela C1 – Exemplo TEDS..... 62

LISTA DE ABREVIATURAS E SIGLAS

FPGA	*Field-Programmable Gate Array* (Matriz de Portas Programável em Campo)
I/O (E/S)	*Input/Output* (Entrada/Saída)
ISO	*International Standards Organization* (Organização Internacional de Padronização)
NCAP	*Network Capable Application Processor* (Processador de Aplicação com Capacidade de Operar em Rede
NIST	*National Institute of Standards and Technology* (Instituto Americano de Padrões e Tecnologia)
STIM	*Smart Tranducer Interface Module* (Módulo de Interface para Transdutores Inteligentes)
TCP/IP	*Transmission Control Protocol/Internet Protocol* (Protocolo de Controle de Transmissão/Protocolo de Internet)
TEDS	*Transducer Electronic Data Sheet* (Especificações de Transdutor em em Formato Eletrônico)
TII	*Transducer Independent Interface* (Interface Independente de Transdutores)
VHDL	*Very High Speed Integrated Circuit - Hardware Description Language* Linguagem de Descrição de Hardware para Circuitos Integrados de Muito Elevada Velocidade)

SUMÁRIO

1 - INTRODUÇÃO GERAL..... 1
1.1- Introdução..... 1
1.2 - Breve Histórico do Padrão IEEE 1451..... 2
1.3 – Contextualização do Projeto..... 3
1.3.1 - Objetivo da Dissertação..... 3
1.3.2 - Motivação e Justificativa..... 4
1.3.3 – Trabalhos Relevantes na Área..... 4
1.3.4 – Organização do texto..... 5

2 – REDES DE TRANSDUTORES..... 6
2.1 - Características das Redes de Transdutores..... 6
2.2 – Definição de Sensor e Atuador..... 7
2.3 - Transdutor conectado em Rede..... 7
2.4 – Tecnologias para Redes de Transdutores..... 8

3- A INTERFACE PADRÃO 1451..... 9
3.1 – Objetivos do Padrão IEEE 1451..... 9
3.2 - Padrão de Interfaceamento IEEE 1451.2..... 9
3.2.1-Definição de Sensor Inteligente:..... 9
3.3 - Módulo de Interface para Transdutores Inteligentes (STIM)..... 10
3.4- Especificação Funcional do Transdutor Inteligente:..... 12
3.4.1. Funções:..... 12
3.4.2. Endereçamento:..... 13
3.4.3. Disparo:..... 14
3.4.4. Transporte de Dados:..... 15
3.4.5. Controle:..... 16
3.4.6. Estado:..... 16
3.4.7. Interrupções:..... 17
3.5. Interface Independente para Transdutores (TII):..... 18
3.5.1. Aspectos Físicos e Elétricos da Interface TII:..... 18
3.5.2. Protocolos de Comunicação:..... 21
3.6. Formatos TEDS:..... 26
3.6.1. Especificação dos Formatos TEDS:..... 26

4- MICROCONTROLADOR DA FAMÍLIA MC68HC908QT/QY® 29
4.1 - Pinagem..... 31
4.2 - Memória..... 31
4.3 - Placa de desenvolvimento M68EVB908Q®..... 31

5- IMPLEMENTAÇÃO DO STIM UTILIZANDO MICROCONTROLADOR DA MOTOROLA..... 33
5.1 - NCAP utilizado para simulação do STIM..... 33
5.2 - Metodologia..... 34
5.3 – Diagrama de blocos: STIM – NCAP – REDE..... 35
5.4 - Interface Padrão TII..... 35
5.5 - Entradas Analógicas..... 36
5.6 – Saídas Digitais..... 36
5.7 – Funções..... 37
5.8 - Configurações do Microcontrolador..... 37
5.9 - Fluxograma..... 39
5.9.1- Fluxograma que implementa o STIM..... 40
5.9.2- Código fonte, em Assembly, que implementa o STIM..... 44

6 - RESULTADOS..... 45
6.1 - Resultados Experimentais..... 45
6.2 - Montagem no Laboratório..... 51
6.3 - Característica "*Plug and Play*"..... 52

7 - CONCLUSÕES GERAIS..... 53
7.1 – Conclusões..... 53
7.2 – Sugestões de Trabalhos Futuros..... 54

REFERÊNCIAS..... 55

Apêndice A - Endereços de Funções mais Utilizados..... 58

Apêndice B - Tipos de Dados IEEE 1451.2..... 59

Apêndice C - Exemplo TEDS: 1canal transdutor, sensor de pressão..... 62

1 – INTRODUÇÃO GERAL

1.1- Introdução

Os sistemas de instrumentação distribuída são compostos por transdutores (sensores e atuadores) interconectados em rede. Quando esta é implementada adequadamente, os transdutores podem operar de forma integrada, melhorando o desempenho de um sistema de instrumentação. É possível, então, a monitoração remota das variáveis de um processo e o compartilhamento de informações através desta rede.

O interfaceamento de um transdutor com uma rede é o processo de conexão tanto elétrica quanto física do transdutor com um nó da rede. Durante vários anos, os engenheiros projetistas de instrumentação tiveram que empregar muito tempo e esforço para projetar interfaces de transdutores com redes de campo. Devido a ausência de interfaces padronizadas, a utilização de diferentes redes e mesmo mudanças nas aplicações exigiam que a interface tivesse de ser reprojetada.

Evidentemente, o esforço de reprojeto é um ponto negativo, porque influencia diretamente no custo da aplicação. O problema torna-se mais sério quando se utiliza tecnologias proprietárias, isto é, quando utilizados dispositivos e técnicas que pertencem apenas a um fabricante, fazendo com que o usuário fique vinculado a esse fornecedor.

O mercado dos barramentos de campo caracterizou-se sempre por uma acentuada fragmentação. Houve, e continua havendo, uma grande quantidade de diferentes tecnologias associadas, sendo que, na maioria as interfaces, os protocolos de comunicação são proprietários. Isto se transformou em uma barreira a ser superada pelos projetistas e fabricantes de transdutores, pois há necessidade de seus produtos serem suficientemente flexíveis para operarem com diferentes redes de campo e para suportar a grande variedade de protocolos de comunicação [1].

Existe atualmente uma grande variedade de produtos inteligentes com capacidade de facilitar o fluxo de informações através de uma rede. É o caso dos transdutores inteligentes, que possuem várias vantagens em relação aos convencionais, mas que não são aproveitados com toda sua potencialidade. A falta de interfaces padronizadas entre transdutores inteligentes e redes é um dos fatores responsáveis por este problema.

1.2 - Breve Histórico do Padrão IEEE 1451

Em 1993, o Comitê Técnico em Tecnologia de Sensores (TC-9 - *Technical Committee on Sensor Technology*) da Sociedade de Instrumentação e Medidas do IEEE (IEEE *Instrumentation and Measurement Society*) começou a trabalhar na definição de um padrão de interfaceamento para conexão de transdutores inteligentes em rede.

Em parceria com o Instituto Americano de Padrões e Tecnologia (NIST - *National Institute of Standards and Technology*), o IEEE iniciou uma série de *workshops* sobre padronização de interfaces, que deram origem aos grupos de trabalho que formularam as especificações da família de normas 1451 [2].

Em março de 1994 foi realizado o primeiro *workshop* do NIST/IEEE sobre interfaces para transdutores inteligentes. Em setembro, do mesmo ano, ocorreu o segundo w*orkshop*, no qual foi apresentado o conceito de especificações de transdutores em formato eletrônico.

Esses eventos motivaram a formação de dois grupos de trabalho denominados IEEE P1451.1 (em fase de projeto) e IEEE 1451.2, respectivamente. No terceiro *workshop*, realizado em maio de 1995, foram demonstrados os primeiros conceitos de interoperabilidade e *plug and play*[1] em redes de transdutores. Em setembro de 1996 foram criados os grupos de trabalho IEEE P1451.3 e IEEE P1451.4 .

Em 1997 foi aprovado o padrão IEEE Std. 1451.2 (*Transducer to Microprocessor Communication Protocols and Transducers Electronic Data Sheet (TEDS) Formats*) [3]. Esta norma foi criada com o objetivo de padronizar a interface entre transdutores e processadores de rede, definindo, para tal fim, o Módulo de Interface para Transdutores Inteligentes (STIM - *Smart Transducer Interface Module*), a Interface Independente de Transdutores (TII - *Transducer Independent Interface*) e as especificações de transdutores para serem gravadas em formato eletrônico (TEDS – *Transducer Electronic Data Sheet*).

O STIM deve conter um processador para implementar as funcionalidades descritas no padrão 1451.2, uma memória de programa para armazenar as estruturas TEDS, circuitos de condicionamento e conversão de sinais, e os transdutores propriamente ditos.

Em 1999 foi aprovado o padrão IEEE Std. 1451.1 (*Network Capable Application Processor (NCAP) Information Model for Smart Transducers*) [4]. O padrão IEEE 1451.1 foi criado com o objetivo de padronizar o *software* da interface entre processadores de aplicação para operar em rede (NCAPs) e redes de controle. Isto possibilitou a independência entre um

[1]*Plug and play*: termo que faz referência ao processo de conexão e posterior funcionamento de um dispositivo em um determinado sistema, sem a necessidade de reconfigurações.

nó de rede e o protocolo de comunicação da rede. A padronização é atingida através do desenvolvimento de um conjunto comum de informações ou modelo objeto, incluindo blocos de transdutores, blocos de funções e blocos de rede. Este modelo de informação é baseado na programação orientada a objetos.

Em 2003 foi aprovada a diretriz IEEE Std. 1451.3 (*Digital Communication and Transducer Electronic Data Sheet (TEDS) Formats for Distributed Multidrop Systems*) [5]. Nesta norma é especificada a forma de comunicação para um arranjo de vários transdutores distribuídos, cujas informações precisam ser lidas de forma sincronizada através de um barramento conectado ao NCAP.

Em 2004 foi aprovado o padrão IEEE Std. 1451.4 (*Mixed-mode Communication Protocols and Transducer Electronic Data Sheet (TEDS) Formats*) [6]. Nesta norma especifica-se como um sinal analógico, associado a um transdutor, e um sinal digital podem ser disponibilizados através da mesma interface com o NCAP. Este fato é importante para aplicações que precisam de elevadas taxas de aquisição de dados.

Atualmente há um grupo de trabalho desenvolvendo a especificação IEEE P1451.5 (*Wireless Communication Protocols and Transducer Electronic Data Sheets*) visando sua futura aprovação. O grupo de trabalho tem como objetivo a criação de uma especificação para direcionar as aplicações do padrão IEEE 1451 em ambientes de rede sem fio [7].

Há também uma proposta recente denominada de P1451.0, cujo objetivo é a criação de uma diretriz que sirva de base para todas as outras especificações da família 1451, por exemplo, a unificação das TEDS das especificações dos componentes [8], [9].

Outra proposta lançada em 2004 foi a IEEE P1451.6 (*A High-speed CAN-open-based Transducer Network Interface for Intrinsically Safe and Non-intrinsically Safe Applications*) [9]. Neste caso propõe-se o uso de uma rede de alta velocidade baseada no sistema *CAN-open*[2], com diversos módulos contendo transdutores, e a definição de uma camada de segurança no modelo de comunicação.

1.3 – Contextualização do Projeto

1.3.1 - Objetivo da Dissertação

O objetivo deste trabalho foi a implementação de um STIM, com dois canais analógicos de entrada e duas saídas digitais, utilizando um microcontrolador de baixo custo, de acordo com as especificações do padrão IEEE 1451.2.

[2]*CAN-open*: protocolo de comunicação aberto para aplicações industriais baseado em CAN (*Controller Area Network*) e introduzido pelo grupo CiA (*CAN in Automation*).

1.3.2 - Motivação e Justificativa

Recentemente o Grupo de Instrumentação Distribuída da UNESP-Campus de Ilha Solteira implementou um nó de rede completo constituído por um NCAP e por um STIM, em conformidade com o padrão IEEE 1451. O NCAP foi desenvolvido com base nos recursos de um PC (*Personal Computer*) convencional e de um sistema embarcado denominado de gerenciador de protocolo, construído com dispositivos FPGA (*Field Programmable Gate Array*), e conectado à porta paralela do PC. Parte do *software* do NCAP foi escrito em linguagem Java. O STIM também foi implementado com utilização de tecnologia FPGA [10-13].

Como seqüência natural do trabalho considerou-se oportuno a implementação de um STIM com microcontrolador de baixo custo, visando domínio mais amplo da tecnologia de desenvolvimento de STIMs e se constatar a característica *plug and play* de trandutores inteligentes.

Vale ressaltar a importância do desenvolvimento de sistemas baseados no padrão IEEE 1451, levando-se em conta, que o mercado de transdutores deverá movimentar U$S 43 bilhões até o ano de 2008 [14].

1.3.3 – Trabalhos Relevantes na Área

Um trabalho pioneiro na área de transdutores inteligentes em rede foi realizado, em 1995, por um grupo da HP (*Hewlett Packard*), que desenvolveu o protótipo de um sensor inteligente conectado em rede [15]. O trabalho teve como finalidade apresentar as idéias básicas sobre padronização de interfaces.

Em 1998 foi desenvolvido um circuito integrado CMOS que implementava um STIM constituído por um microcontrolador de 8 bits, conversores A/D, e D/A [16].

Em 2000 foi descrito um módulo STIM, que utilizava um sistema embarcado fornecido pela empresa *Analog Devices*, e que continha um núcleo processador baseado no microcontrolador 8051 [17].

Ainda em 2000 foi implementado um nó IEEE 1451 completo para funcionar com uma rede CAN *(Controller Area Network*), utilizando microcontroladores comerciais [18].

Em 2002 foi desenvolvido um STIM de dois canais, utilizando tecnologia de lógica programável FPGA, com aproximadamente 800.000 portas típicas [19].

Em 2003 foi implementado um STIM com um microcontrolador PIC16F877, da Microchip [20].

Em 2004 foi desenvolvido um STIM utilizando dispositivos FPGA´s de baixo custo, com aproximadamente 100.000 portas típicas [11].

1.3.4 – Organização do restante do texto

A dissertação foi dividida em sete capítulos, além dos apêndices e referências.

No Capítulo 2 são apresentados os conceitos fundamentais relacionados às redes de transdutores

No Capítulo 3 é apresentada a interface padrão IEEE 1451, abordando-se em detalhes, a norma IEEE 1451.2, que é a base do presente trabalho de dissertação.

O Capítulo 4 descreve o microcontrolador MC68HC908QY4® da Motorola, que foi utilizado na implementação do STIM.

No Capítulo 5 são apresentados os materiais e métodos empregados para implementar o STIM.

No Capítulo 6 são descritos os resultados e, finalmente, no Capítulo 7, apresentam-se as conclusões deste trabalho.

2 – REDES DE TRANSDUTORES

No final da década de 1970 e início da década de 1980, as empresas utilizavam computadores de grande porte, oferecendo terminais nos quais os funcionários podiam se conectar para obter informações sobre toda a planta do sistema. Esses sistemas possuíam um custo muito elevado, tornando inviável a manutenção e a flexibilidade na transmissão dos dados.

O Computador Pessoal (PC) da IBM, contribuíu para o surgimento das redes de computadores e para o grande desenvolvimento de *software* para aplicações em controle e supervisão. No transcorrer da década de 1980 começaram as aplicações de microprocessadores em ambientes industriais e surgiram os primeiros sensores inteligentes.

Atualmente os conceitos associados às redes de computadores são utilizados pelos projetistas da área de instrumentação para construir sistemas de instrumentação distribuída.

Uma rede de transdutores é um sistema que interliga diferentes tipos de sensores e atuadores com sistemas baseados em PC, com a finalidade de atuar e realizar monitoramento de um determinado processo. A Figura 2.1 apresenta, esquematicamente, uma rede de sensores.

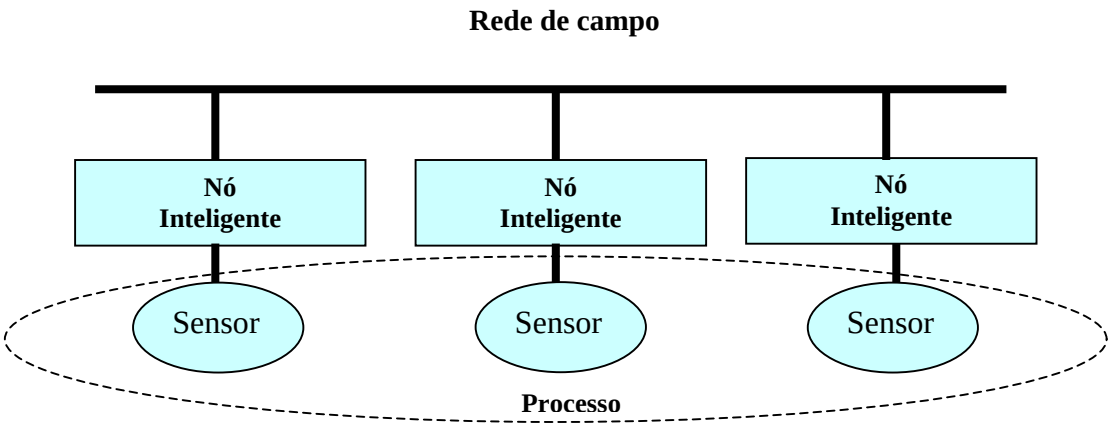


Figura 2.1 – Rede de sensores

2.1 - Características das Redes de Transdutores

As redes de transdutores apresentam as seguintes características:

a) Possibilidade de compartilhar dados de vários transdutores;

b) Facilidade de manutenção;

c) Possibilidade de monitoramento de um processo produtivo completo;

d) Possibilidade de criar sistemas distribuídos;

e) Possibilidade de acesso e monitoramento remoto de um processo.

2.2 – Definição de Sensor e Atuador

Neste trabalho será utilizada a definição proposta pelo IEEE, definindo sensor como sendo um transdutor que converte um parâmetro físico, biológico ou químico em um sinal elétrico e, atuador como sendo um transdutor que aceita um sinal elétrico para convertê-lo em uma ação física [3], [4].

2.3 - Transdutor Conectado em Rede

O diagrama de blocos de um dispositivo transdutor conectado em rede é mostrado na Figura 2.2. É possível observar que uma grandeza pertencente ao mundo físico é detectada pelo transdutor (sensor), e este fornece um sinal elétrico que precisa ser condicionado, ou seja, amplificado, filtrado e calibrado. Na seqüência há um dispositivo conversor que transforma o sinal elétrico do domínio analógico para o digital, a fim de que possa ser processado e posteriormente transmitido para a rede. No caso do transdutor ser um atuador, o sentido inverso é percorrido pelo sinal.

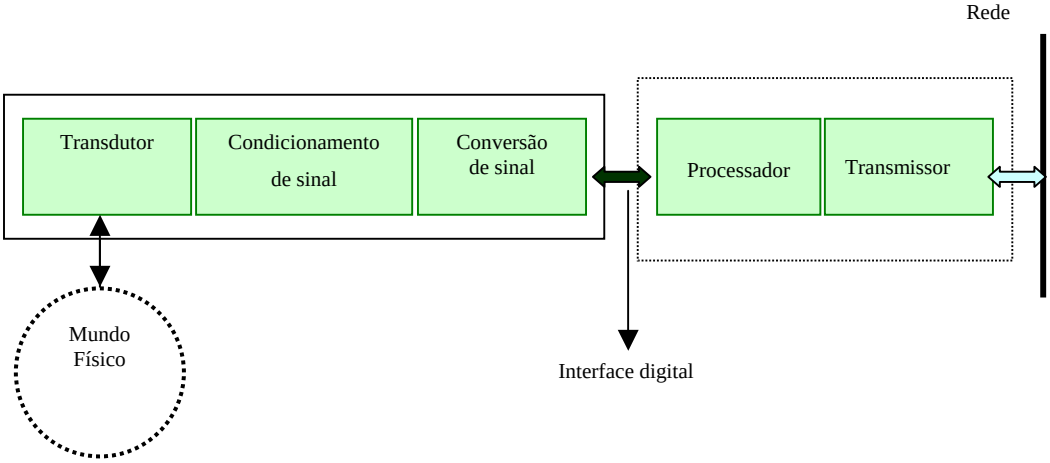


Figura 2.2 – Diagrama de blocos de um transdutor conectado a uma rede

O fato de conectar transdutores em rede, como mostrado na Figura 2.2, possibilita a redução considerável de custos devido a cabeamento, além dos dados serem transmitidos com maior confiabilidade, gerando assim, flexibilidade no sistema de instrumentação.

2.4 – Tecnologias para Redes de Transdutores

Atualmente existem várias tecnologias que podem ser usadas em redes de transdutores, entretanto, muitas delas são de natureza proprietária [1]. Estas tecnologias podem ser usadas em diferentes níveis de uma rede, gerando assim a seguinte classificação geral:

a) Barramentos de sensores: são sistemas básicos de interconexão entre transdutores e computadores ou microcontroladores. Os dados, em formato digital, circulam por um barramento bidirecional. Este tipo de barramento, mostrado esquematicamente na Figura 2.3, constitui a base das outras tecnologias.

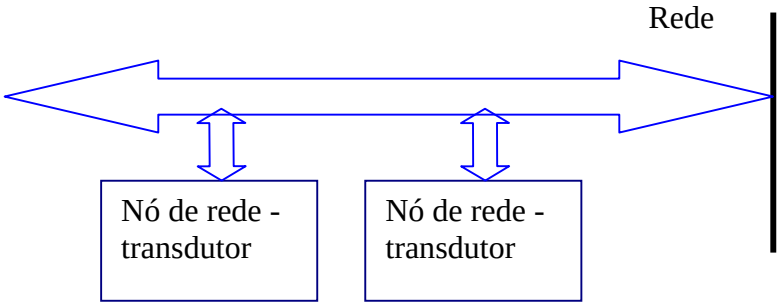


Figura 2.3 – Barramento de sensores

b) Redes de controle: são barramentos de alto nível que executam algum tipo de processamento de dados em cada nó de rede antes de serem passados para o computador ou processador correspondente.

c) Barramentos de campo: são barramentos bidirecionais de dados mais sofisticados e funcionam com um determinado protocolo de comunicação.

De maneira geral, as tecnologias citadas serão chamadas de redes de campo.

3- A INTERFACE PADRÃO 1451

3.1 – Objetivos do Interfaceamento Padrão 1451

A interface padrão IEEE 1451 é resultado do esforço iniciado pelo Comitê Técnico em Tecnologia de Sensores (TC-9 - *Committee on Sensor Technology*) da Sociedade de Instrumentação e Medidas do IEEE (*IEEE Instrumentation and Measurement Society*) e pelo Instituto Americano de Padrões e Tecnologia (NIST - *National Institute of Standards and Technology*). O TC-9 e o NIST iniciaram, na década de 1990, uma série de *workshops* sobre padronização de interfaces que criaram os grupos de trabalho 1451 [2].

O padrão IEEE 1451 define um conjunto comum de interfaces cujas características e objetivos são [13]:

a) O padrão é um conjunto de especificações que tem como objetivo simplificar a conectividade de transdutores;

b) Foi desenvolvido como um conjunto de especificações de domínio aberto de acordo com o consenso da indústria;

c) Seus propósitos são:

- Fornecer um modelo geral de um dispositivo transdutor, quanto a dados, controle, configuração e calibração;
- Fornecer um conjunto comum de interfaces para conectar transdutores com redes de campo.

O IEEE 1451 não é uma outra rede de campo, mas um padrão aberto que pode ser utilizado com diferentes redes. A família IEEE 1451 divide-se em seis componentes. Atualmente têm-se quatro padrões aprovados (IEEE 1451.1, IEEE 1451.2, IEEE 1451.3 e IEEE 1451.4) e três especificações em desenvolvimento (IEEE P1451.0, IEEE P1451.5 e IEEE P1451.6).

3.2 - Padrão de Interfaceamento IEEE 1451.2

3.2.1-Definição de Sensor Inteligente:

Em termos gerais, um sensor inteligente é um dispositivo integrado no qual coexistem o sensor propriamente dito, os circuitos de condicionamento de sinal e um dispositivo contendo um processador e uma memória não volátil, a fim de introduzir a inteligência necessária. O transdutor inteligente pode ser definido como o dispositivo com capacidade de processamento

local, habilitado para tomar decisões baseando-se no sinal de entrada, e que pode enviar ou receber dados em formato digital, facilitando as atividades em sistemas distribuídos [11].

Na Figura 3.1 apresenta-se esquematicamente o modelo elementar de um sensor inteligente.

Figura 3.1. Concepção básica de um sensor inteligente.

No seu caminho, o sinal de entrada atravessa vários blocos. A maioria dos sinais que provêm de diferentes sensores são analógicos em essência e, portanto, precisam de um pré-processamento, isto é, condicionamento de sinal e conversão analógica-digital. Na seqüência, tem-se o microcontrolador encarregado do processamento de dados digitais.

O sensor da Figura 3.1 poderia ser também um atuador, nesse caso a informação percorrerá a seqüência em sentido contrário. Assim sendo, um sinal digital entra no processador, logo depois o sinal passa pelo pós-processador e finalmente o sinal elétrico entra no atuador. O atuador irá executar algum tipo de ação física baseando-se nessa informação.

3.3 - Módulo de Interface para Transdutores Inteligentes (STIM)

O módulo STIM é composto por um ou mais transdutores, circuitos de condicionamento de sinal, conversores A/D e D/A necessários para realizar o interfaceamento dos transdutores com o microcontrolador local, um dispositivo de memória não volátil para armazenar os formatos TEDS, acessível através do microcontrolador e a lógica necessária para implementar a interface.

Deve-se notar, que no contexto do padrão, os transdutores consideram-se como sendo parte do STIM. De fato para fornecer a auto-identificação de cada transdutor, eles devem permanecer inseparáveis do STIM durante seu funcionamento normal [3].

Cada transdutor que faz parte do STIM é chamado de canal transdutor. Cada canal é considerado inteligente devido aos três fatos mencionados a seguir:

- ✓ Um canal transdutor é descrito através das especificações armazenadas em formato eletrônico em um dispositivo de memória não volátil (TEDS);
- ✓ O controle e os dados associados com cada canal são digitais;

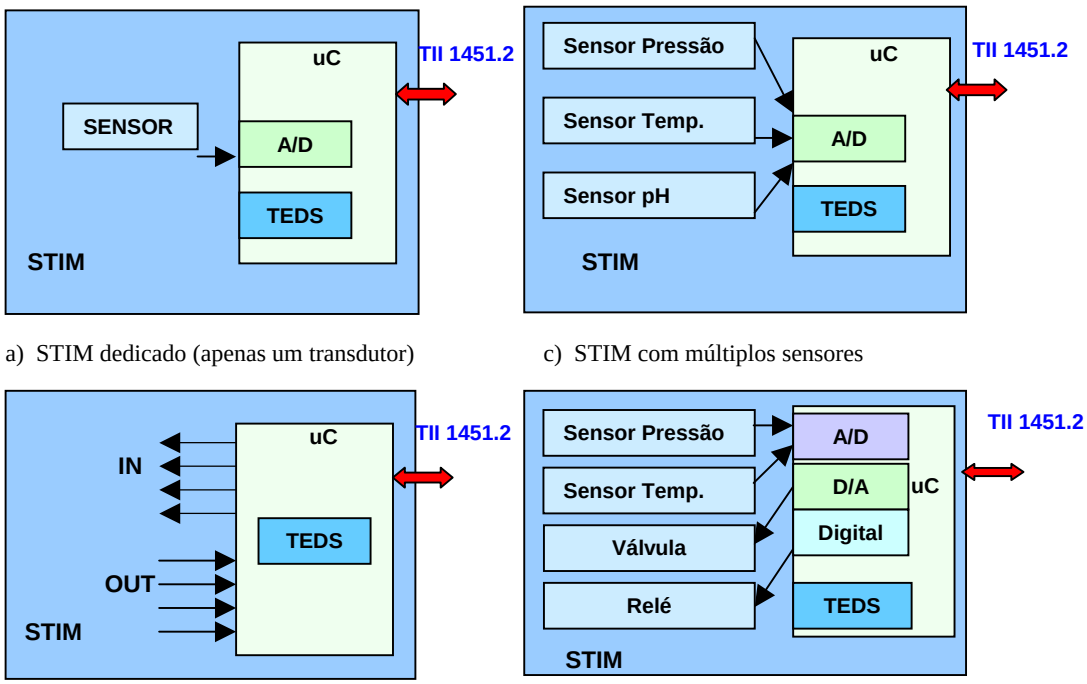
- ✓ São fornecidas funções de controle, estado e disparo para suportar a funcionalidade própria de cada canal.

O STIM é controlado pelo módulo NCAP via interface TII. Deste modo, quando conectado a uma rede através do NCAP, o conjunto é "visto" pela rede como um nó de inteligência distribuída. Este fato é fundamental porque permite o funcionamento de transdutores inteligentes em sistemas de medição e controle distribuídos.

Apenas a interface entre STIM e NCAP é padronizada pelo IEEE. As outras interfaces, por exemplo, entre transdutor e circuitos de condicionamento ou entre circuitos de condicionamento e conversor, ficam ao critério do projetista. Deste modo, através da implementação livre destas interfaces, pretende-se continuar introduzindo a necessária diferenciação de produtos no que diz respeito ao desempenho, qualidade, apresentação e custo.

A lógica do Padrão 1451.2 pode ser implementada de diferentes maneiras, como por exemplo, com microcontrolador de baixo custo, FPGA ou ASIC.

A implementação do módulo STIM é bastante flexível e sua concepção pode adotar diversas formas. A fim de esclarecer um pouco mais esta idéia, na Figura 3.2 apresenta-se algumas possibilidades de implementação com um microcontrolador de baixo custo.



a) STIM dedicado (apenas um transdutor)

c) STIM com múltiplos sensores

b) STIM com 8 canais digitais I/O

d) STIM com sensores e atuadores

Figura 3.2. Exemplo de diferentes formas de implementar um módulo STIM.

A Figura 3.2a mostra um STIM com um único canal transdutor, no caso, um sensor analógico. Essa aplicação, que contém apenas um transdutor, é chamada de STIM dedicado e pode ser utilizada quando se precisa um módulo específico para detectar as condições de um determinado ambiente através de um sensor, ou então, executar alguma ação específica através de um atuador.

O exemplo da Figura 3.2b mostra um módulo STIM com entradas e saídas digitais. O formato TEDS neste caso permite realizar a descrição de um STIM de 8 canais, ou em forma alternativa, de um STIM de dois canais, um de entrada e outro de saída e cada um deles composto por quatro canais transdutores. O caso da Figura 3.2c apresenta um módulo STIM com múltiplos sensores analógicos. Essa configuração pode ser utilizada, por exemplo, para detectar as condições de um líquido que faz parte de um determinado processo. Finalmente, o ultimo exemplo da Figura 3.2d ilustra um STIM com sensores e atuadores. Este caso, é conveniente quando é necessário que um atuador execute alguma ação baseada em algum algoritmo de controle. O algoritmo pode ser executado no STIM ou no NCAP.

Estes exemplos ilustram a grande flexibilidade para projetar sistemas baseados em transdutores inteligentes através do padrão 1451.2.

3.4- Especificação Funcional do Transdutor Inteligente

O padrão apresenta um conjunto de diretrizes que especificam os tipos de canais transdutores que podem ser implementados no STIM e as funções requeridas para permitir que os canais possam operar em conjunto.

3.4.1. Funções

Cada STIM deverá implementar as seguintes funções:

- Endereçamento;
- Transporte de dados;
- Meta-TEDS;
- Estado global;
- Controle global;
- Disparo;
- Capacidade *hot-swap*;
- Interrupção.

Cada canal transdutor de um STIM deverá implementar as seguintes funções:

- Canal-TEDS;
- Dados do transdutor;
- Estado;
- Controle.

3.4.2. Endereçamento

A função de endereçamento é utilizada em conjunto com a função de transporte de dados via interface TII. A estrutura de endereçamento padronizada faz uso de um conjunto composto por dois *bytes*, chamado de endereço completo. Um endereço completo especifica se o dado deve começar a ser lido ou gravado, com que tipo de função e através de que canal do STIM. A estrutura de um endereço completo divide-se em um *byte* que representa o endereço de função (*byte* mais significativo) e um *byte* que representa o endereço de canal (*byte* menos significativo). A estrutura de um endereço completo é apresentada na Figura 3.3.

A cada transdutor no STIM deverá ser atribuído um número de canal entre 1 e 255. Para saber a quantidade de canais que possui um STIM em particular, pode-se acessar o campo "número de canais implementados" pertencente ao bloco de dados Meta-TEDS. O canal zero, denominado de *CHANNEL_ZERO* no padrão, tem um significado especial. Quando uma função é endereçada para o canal zero, a função refere-se ao STIM como sendo uma coleção de todos os canais. Assim, uma função terá uma interpretação global quando for endereçada para o canal zero e uma interpretação especifica para cada canal quando for endereçada para os canais 1 até 255.

Endereço de função Byte mais significativo								**Endereço de canal** Byte menos significativo							
r/w	Código de função							Número de canal							
msb							lsb	msb							lsb

r =leitura, w= escrita

Figura 3.3. Estrutura de um endereço completo.

Na cláusula [4.4.3.2] do padrão, descrevem-se os endereços e as funções correspondentes. Para exemplificar, o endereço 128 utiliza-se para ler dados de transdutores em forma global quando o endereço for aplicado ao canal zero, e para ler dados de um transdutor específico quando for aplicado a um canal entre 1 e 255. No Apêndice A encontram-se os endereços de função mais utilizados.

3.4.3. Disparo

A função de disparo deve ser suportada por uma das linhas da interface TII. Esta função estabelece o momento em que deve ser enviado um comando determinado desde o NCAP para o STIM (sinal de disparo) e o momento em que o STIM deve indicar que a ação ocorreu (reconhecimento do disparo).

Este tipo de ação fará, por exemplo, que um sensor amostre um dado novo. A resposta de um STIM para um evento de disparo é ilustrada na Figura 3.4.

O primeiro estágio do ciclo de disparo é a inicialização do sistema, logo depois, o STIM ficará esperando um evento de disparo.

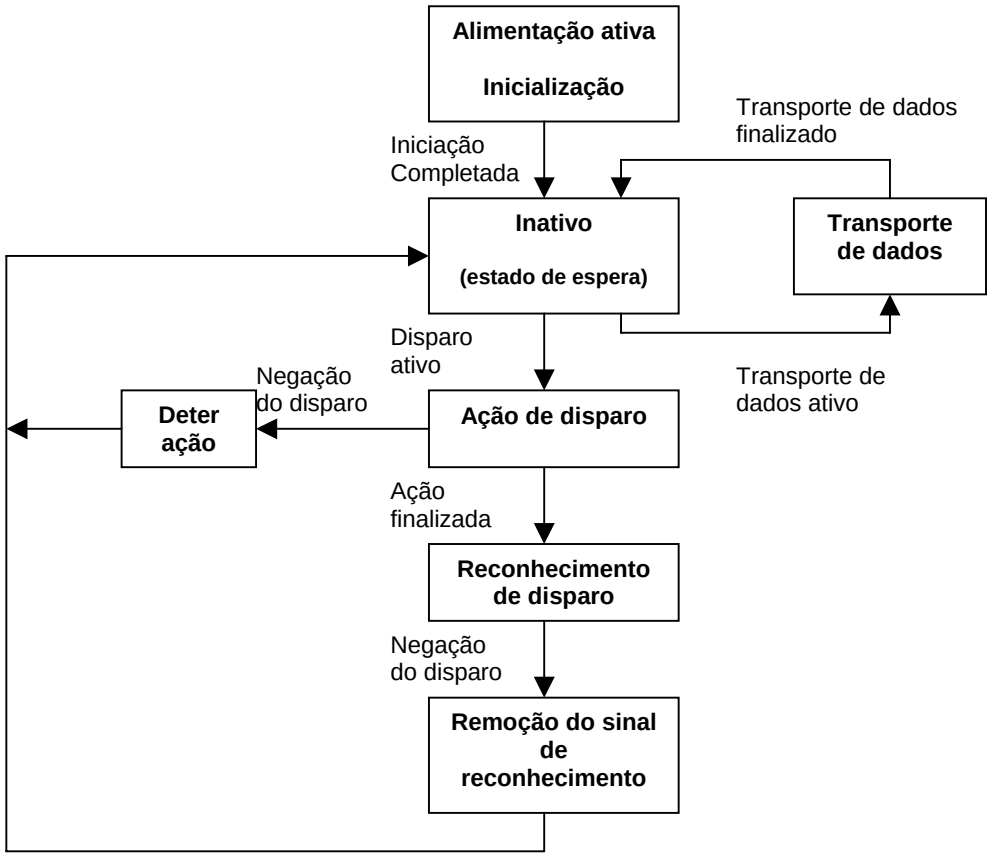


Figura 3.4. Função de disparo sobre a TII.

Se o processo de disparo for detido, será abortada toda ação que estiver acontecendo no STIM. Assim sendo, o resultado de um canal transdutor ou o estado de um atuador, será indefinido. Se nenhuma ação de negação acontecer, o STIM reconhecerá o sinal de disparo e ficará esperando até o NCAP mudar o nível lógico do sinal. Uma vez feito o reconhecimento,

o STIM removerá esse sinal e o ciclo evoluirá para o estado de espera. O STIM permanecerá nesse estado até completar o transporte de dados (escrita ou leitura de dados de transdutores, estado, controle ou campos de dados TEDS). O disparo permanecerá inativo até o transporte de dados ter finalizado.

3.4.4. Transporte de Dados

O transporte de dados é suportado pelo conjunto de linhas de sinal da interface padronizada TII. Deste modo, é possível identificar quando um sinal de transporte de dados está ativo e, além disso, utilizá-lo para delimitar as estruturas ou quadros de leitura e escrita.

O transporte de dados interage com a função de disparo e deve estar inativo antes de ativar o sinal de disparo. Quando a função de transporte de dados passa para o estado ativo, o NCAP envia para o STIM um endereço completo especificando o tipo de operação (escrita ou leitura) a ser desenvolvido, qual a função e qual o canal envolvido no processo.

a) Dados de Transdutores

A função de transporte de dados mais comumente utilizada é a relacionada com os transdutores. Utiliza-se para ler dados produzidos por sensores ou escrever dados para serem usados por atuadores. A leitura de dados de um transdutor, após abortar o ciclo de disparo, produzirá resultados imprevisíveis.

b) Transporte de Dados Relacionado com o Canal Zero

A função de transporte de dados utiliza-se também para ler especificações de transdutores gravadas nos blocos TEDS, para ler informações de estado e escrever comandos de controle. As funções relacionadas com o canal zero, *write global transducer data* (escrever dados globais para transdutores) e *read global transducer data* (ler dados globais de transdutores), gravam estruturas de dados ou lêem estruturas de dados de todos os canais implementados, começando pelo canal número 1.

A título de ilustração, considere-se o seguinte exemplo: uma operação de leitura que utiliza a função *read global transducer data*, executa-se em um STIM que possui sensores nos canais 1,2 e 4, e um atuador no canal 3.

O dado deverá incluir: N1 *bytes* para o canal 1, N2 *bytes* para o canal 2, N3 *bytes* para o canal 3 e N4 para o canal 4, sendo N1,N2,N3 e N4 os tamanhos dos modelos de dados para cada canal, como especificado no bloco de dados Canal-TEDS.

3.4.5. Controle

A função de controle permite enviar comandos para o STIM de forma global, ou então, para cada canal em particular. Esses comandos irão afetar o estado de funcionamento do STIM. Para escrever um comando de controle utiliza-se o endereço de função *write channel control command* (escrever comando de controle para um canal) caso for endereçado para um canal entre 1 e 255, ou *write global control comand* (escrever comando de controle global) caso for endereçado para o canal zero. Os comandos de controle estão compostos por dois *bytes*, um *byte* correspondente ao endereço de função e outro que indica qual o comando a ser enviado. O padrão define os comandos para o canal zero e para os canais restantes. Por exemplo, o valor "1" indica *reset do STIM* quando o comando é endereçado para o canal zero e *reset de canal* quando é endereçado para um canal entre 1 e 255.

3.4.6. Estado

A função de estado permite ao NCAP determinar o estado do STIM de forma global ou para cada canal individual. A fim de implementar esta função utilizam-se registradores de estado padrão e auxiliares. Cada *bit* em uma posição específica de um registrador de estado representa a ausência ou presença de uma condição particular. A presença de uma condição indica-se com um bit em "1" na posição apropriada. A Figura 3.5 ilustra a idéia de forma esquemática.

A função de estado também é utilizada em conjunto com as máscaras de interrupção e as interrupções para indicar que o STIM está solicitando algum tipo de serviço. Os estados possíveis tanto padrões quanto auxiliares especificam-se no padrão.

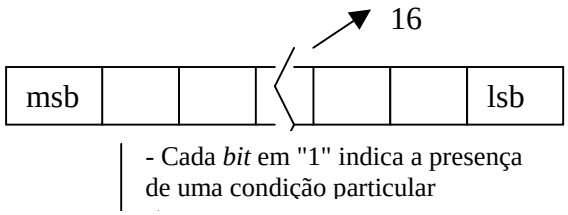


Figura 3.5. Registrador de estado.

3.4.7. Interrupções

A requisição de interrupção se faz através de uma linha de entrada de uma Unidade Microprocessadora (MPU). A requisição de interrupção é utilizada por dispositivos de E/S para interromper a execução do programa em curso e fazer a MPU passar para um programa especial, denominado rotina de serviço de interrupção. Quando a entrada é ativada, a MPU executa esse programa especial, que normalmente envolve atender ao dispositivo que provocou a interrupção. Completada a execução, a MPU retoma o programa em que estava trabalhando ao ser interrompida.

Existem situações nas quais responder a uma requisição de interrupção não é conveniente para a MPU. O que acontece quando um dispositivo de E/S interrompe a MPU enquanto ela está executando uma parte de um programa que requer processamento contínuo? Neste caso, a ocorrência de uma interrupção pode resultar indesejável; por esse motivo, todos os microprocessadores dispõem de algum recurso para inibir a operação de interrupção, sob controle do programa.

A ocorrência da interrupção será ignorada pela MCU quando o *flag* ou sinalizador de inibição de interrupções estiver ativo em uma posição de um registrador de máscaras de interrupção. Este tipo de interrupção denomina-se mascarável. Um outro tipo de interrupção é a não mascarável, neste caso, a MPU sempre lhe dá atenção à requisição independentemente do estado apresentado pelo sinalizador. Em outras palavras, seu efeito não pode ser inibido pelo programador.

O STIM deve conter dois registradores de máscaras de interrupção, um deles padrão e outro auxiliar. Ambos os registradores devem possuir um tamanho de dois *bytes*. No STIM estabelecem-se operações lógicas do tipo *and bit* a *bit* entre as posições dos registradores padrão e auxiliar de máscaras de interrupção e os registros padrão e auxiliar de estado. Desta maneira, quando um *bit* estiver em "1" em qualquer uma das posições em ambos os registradores de interrupção e a posição correspondente no registrador de estado também estiver em "1", produz-se uma requisição de interrupção que pode ser global ou para um canal em particular dependendo da posição do *bit* de requisição de serviço, no registrador padrão de estados.

A interface digital TII possui uma linha dedicada para permitir ao STIM realizar uma requisição de interrupção ao NCAP. O sinal de interrupção será ativado se a posição correspondente a *service request bit* no registrador padrão de estados do STIM estiver em "1".

3.5. Interface Independente para Transdutores (TII)

A interface TII entre STIM e NCAP, uma outra diretriz que faz parte do padrão, especifica as características elétricas e físicas da interface, protocolos de comunicação e sincronização.

3.5.1. Aspectos Físicos e Elétricos da Interface TII

A interface digital independente para transdutores é composta por dez pinos: oito pinos suportam a funcionalidade do padrão e os dois restantes fornecem a alimentação. A Figura 3.6 mostra a interface e suas linhas em forma esquemática.

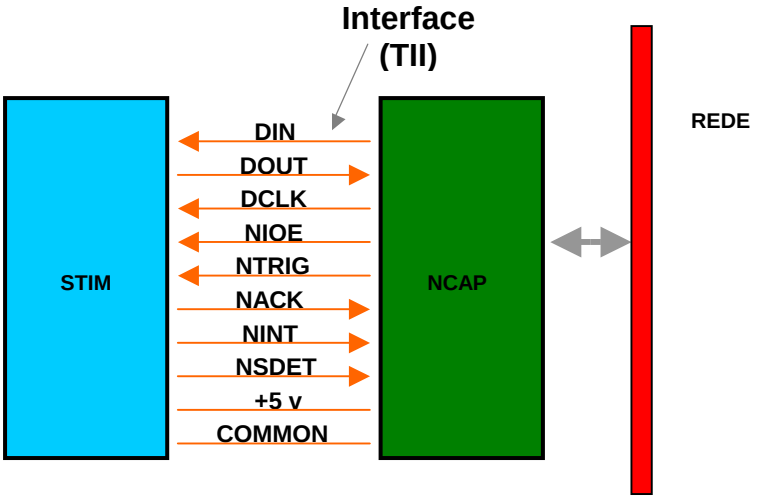


Figura 3.6. Interface TII entre STIM e NCAP.

A comunicação através da interface é do tipo serial síncrona e concebida em base ao protocolo *Serial Peripheral Interface* (SPI™).

Na tabela 3.1, estão definidas as linhas da interface, indicando-se o tipo de lógica utilizada e qual o módulo que opera cada linha.

Em seguida a Tabela 3.2 exibe a atribuição de pinos e cores recomendadas pelo Padrão 1451.2.

Tabela 3.1 - Definição das linhas da TII – IEEE 1451.2

Grupo	Linha	Abreviação	Lógica	Operada pelo:	Descrição
Dados	DATA_OUT	DOUT	Positiva	STIM	Transporte de dados desde o STIM para o NCAP
	DATA_IN	DIN	Positiva	NCAP	Endereçamento e transporte de dados desde o NCAP para o STIM
	DATA_CLOCK	DCLK	Borda ascendente	NCAP	A cada evento de clock guardam-se os dados associados a DIN ou DOUT
	N_IO_ENABLE	NIOE	Ativa Baixa	NCAP	Sinaliza se o transporte de dados está ativo e delimita a estrutura de dados
Disparo	N_TRIGGER	NTRIG	Flanco Descendente	NCAP	Executa funções de Disparo
Suporte	POWER	POWER	-	NCAP	Alimentação 5 v
	COMMON	COMMON	-	NCAP	Referência
	N_ACKNOWLEDGE	NACK	Flanco Descendente	STIM	Serve para 2 funções: 1.Reconhecimento de disparo 2.Reconhecimento de transporte de dados (Limitador de 8 *bits*)
	N_STIM_DETECT	NSDET	Ativa Baixa	STIM	Usada pelo NCAP para detectar a presença do STIM
Interrupção	N_IO_INTERRUPT	NINT	Flanco Descendente	STIM	Usada pelo STIM para solicitar um serviço ao NCAP

Tabela 3.2 - Exibe a atribuição de pinos e cores, recomendada pelo padrão.

Pino	Sinal	Cor	Direção para o NCAP	Direção para o STIM
1	DCLK	Marrom	OUT	IN
2	DIN	Vermelho	OUT	IN
3	DOUT	Laranja	IN	OUT
4	NACK	Amarelo	IN	OUT
5	COMMON	Verde	POWER	POWER
6	NIOE	Azul	OUT	IN
7	NINT	Violeta	IN	OUT
8	NTRIG	Cinza	OUT	IN
9	POWER	Branco	POWER	POWER
10	NSDET	Preto	IN	OUT

Para definir as entradas e saídas utiliza-se o STIM como referência, isto é, a linha DIN é a entrada de dados no STIM e DOUT é a saída de dados no mesmo módulo. A mesma convenção utiliza-se para as outras linhas. O DCLK é o relógio, sendo que essa linha é operada pelo NCAP. A linha NIOE faz referência à habilitação de entrada / saída e também é controlada pelo NCAP. Através do NTRIG o NCAP efetua um sinal de disparo e por meio do NACK o STIM envia um sinal de reconhecimento. A linha NINT é utilizada pelo STIM para efetuar uma interrupção e NSDET é utilizada para detectar a presença do STIM. Finalmente é fornecida a alimentação necessária de 5 (ou 3) volts e o pino de referência (COMMON) [3].

O módulo NCAP é quem controla todas as comunicações. Nele se geram todas as trocas de informações. A única linha que pode ser operada diretamente pelo STIM é a linha NINT. Embora apresentadas no padrão como operadas pelo STIM, as linhas DOUT, NSDET e NACK não são totalmente controladas pelo STIM. NSDET é um sinal passivo que se conecta a terra. A linha NACK só é ativada em resposta a uma ação do NCAP.

Ainda, no caso da linha NINT, que pode ser chamada de "linha de requisição de serviços", as comunicações resultantes estão sob controle do módulo NCAP.

Quanto ao conector físico para implementar a interface (se bem que o grupo de trabalho do IEEE 1451.2 tentou padronizar um tipo de conector) foi reconhecido que os conectores são muito dependentes da aplicação para poder estabelecer um consenso geral.

É provável que vários setores da indústria desenvolvam algum tipo de conector padronizado, ao menos, para as aplicações primárias.

Um par de conectores já foi utilizado com sucesso. Conectores com 10 terminais (5x2) e conector para fita plana têm sido utilizados para demonstrações em ambientes pouco agressivos. Outro conector, tal como o DB de 15 pinos, usado normalmente em sistemas informatizados, provavelmente seja mais apropriado para aplicações industriais.

Outro aspecto importante é a forma em que é alimentado o módulo STIM. O NCAP deve fornecer alimentação, via linha POWER da interface TII. O consumo do STIM não deve superar os 75 mA para o valor de tensão especificado.

Se requerido, o STIM pode utilizar uma fonte de alimentação separada, mas, a alimentação dos circuitos de controle da interface (microcontrolador, FPGA, ou ASIC), só pode ser fornecida pelo NCAP utilizando as linhas POWER e COMMON.

3.5.2. Protocolos de Comunicação

Os protocolos descrevem a implementação da função de disparo (*trigger*) e a forma em que se realiza o transporte de dados utilizando a interface TII. Os protocolos de níveis mais elevados são a estrutura ou quadro de leitura, quadro de escrita e disparo.

O termo quadro ou estrutura (*frame*) faz referência à seqüência de *bytes* que é transmitida de uma maneira bem definida. Um *byte* transferido é uma seqüência de 8 *bits*, onde cada *bit* é transferido seguindo o protocolo de transferência de *bits.*

A gravação, ou operação de escrita é realizada quando o NCAP envia informações em formato de quadro através da linha DIN; essas informações são caracterizadas por um *byte* indicando um endereço de função e um *byte* indicando o endereço de canal correspondente. Também é utilizado o protocolo de escrita quando o NCAP envia informações necessárias, por exemplo, para serem utilizadas por um atuador.

A leitura é realizada quando o STIM envia informações em formato de quadro através da linha DOUT da interface. Estas informações são dados associados com um sensor e que são enviadas para o NCAP e podem ser lidas através da rede.

A seqüência de leitura/escrita entre o NCAP e o STIM é ilustrada na figura 3.7.

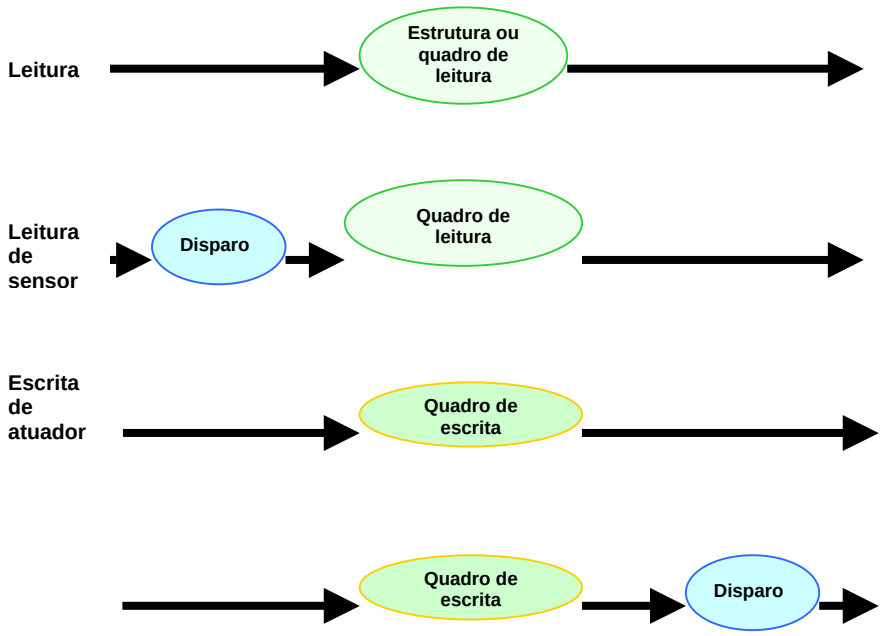


Figura 3.7. TII: Protocolo geral de transferência de dados.

A seguir são apresentados os protocolos de comunicação especificados no padrão IEEE 1451.2.

a) DISPARO

A função de disparo é utilizada antes de ler dados de um sensor ou depois de escrever dados para serem usados por um atuador. A seqüência de operações é a seguinte:

a) o NCAP espera durante um intervalo de tempo igual a *Channel Write Setup Time* (tws), gravado no bloco de dados Canal-TEDS;

b) o NCAP ativa a linha NTRIG;

c) o STIM ativa a linha NACK;

d) o NCAP nega a NTRIG;

e) o STIM nega a NACK;

f) o NCAP espera durante um intervalo de tempo igual a *Channel Read Setup Time* (trs), gravado no bloco de dados Canal-TEDS.

A Figura 3.8, ilustra a seqüência de eventos e mostra alguns dos tempos mais importantes especificados no padrão. Por simplicidade, as transições correspondentes do diagrama de tempos têm sido consideradas abruptas, embora, na realidade correspondam a uma rampa ascendente ou descendente com uma determinada especificação de tempo.

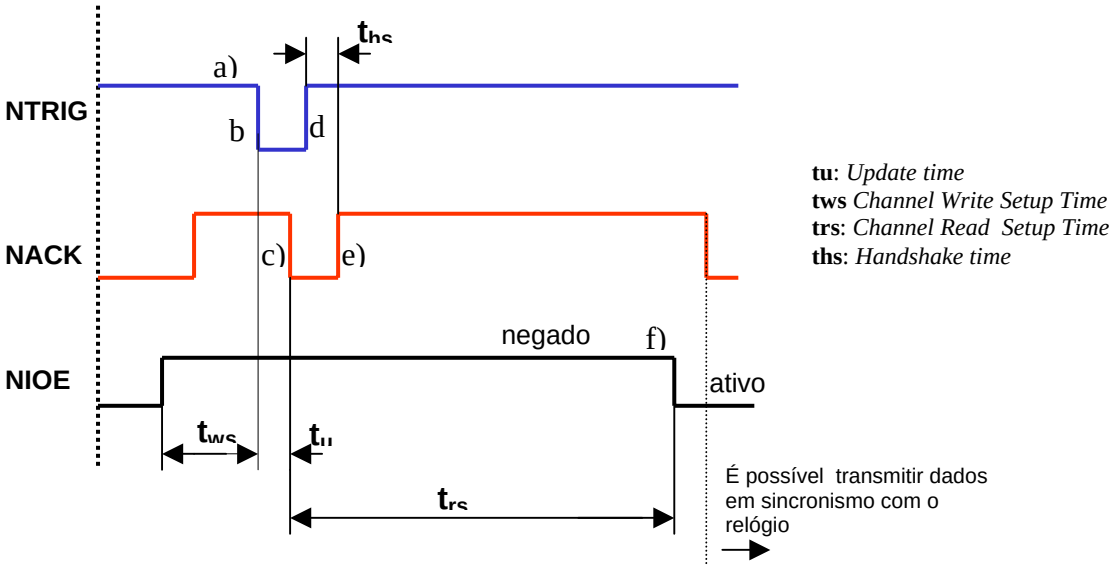


Figura 3.8. Protocolo que implementa a função de disparo / diagrama de tempos.

Após um evento de disparo, e com o sinal NIOE ativo, assim que ocorre uma transição na linha NACK, é possível a transferência de *bits*. A partir desse momento, e em sincronismo com o sinal de relógio (DCLK), é possível realizar operações de escrita e de leitura utilizando-se as linhas DIN e DOUT da interface (protocolos de transferência de quadro de leitura e escrita).

b) TRANSFERÊNCIA DE *BITS*

A transferência de dados entre o NCAP e o STIM, se realiza em forma serial via linha DIN, e via DOUT desde o STIM para o NCAP. A transferência de dados é controlada pela linha DCLK (relógio) através da seguinte seqüência:

a) Evento na linha DCLK (borda de subida);

b) Na primeira borda de subida do DCLK, o primeiro *bit* é enviado pelo emissor (NCAP via DIN, ou STIM via DOUT);

c) Na primeira borda de descida, o *bit* é armazenado receptor (NCAP em DIN ou STIM em DOUT);

d) Os *bits* seguintes são transmitidos repetindo-se os passos b) e c).

A comunicação é do tipo *half duplex*, isto é, pode-se estabelecer comunicação entre STIM e NCAP, ou vice-versa, mas não ao mesmo tempo. Assim, quando for estabelecida a comunicação entre o NCAP e o STIM, a linha DOUT é ignorada pelo NCAP, e vice-versa.

c) TRANSFERÊNCIA DE *BYTES*

A transferência entre os módulos deve ser feita em grupos de 8 *bits* (1 *byte*) utilizando-se o protocolo de transferência de *bits*. O padrão especifica o protocolo de transferência de *bytes* desde o NCAP para o STIM (*byte write transfer*) e desde o STIM para o NCAP (*byte read transfer).*

Transferência de *byte* de escrita (*byte write transfer*):

O NCAP começará a transferir um *byte* de escrita só depois de observar uma transição na linha NACK. O módulo STIM estabelecerá uma transição na linha NACK quando for apropriadamente entregue o *byte* prévio e o NCAP estiver pronto para proceder.

Transferência de *byte* de leitura (*byte read transfer*):

O NCAP começará a transferir um *byte* de leitura só depois de observar uma transição na linha NACK. O módulo STIM estabelecerá uma transição na linha NACK quando for apropriadamente entregue o *byte* prévio e o NCAP estiver pronto para operar.

d) PROTOCOLO DE TRANSFERÊNCIA DE QUADRO DE LEITURA

a) o NCAP ativa a linha NIOE;

b) o NCAP espera o momento em que o STIM ativar a linha NACK;

c) o NCAP "escreve" o endereço funcional utilizando o protocolo *byte write transfer*;

d) o NCAP "escreve" o endereço do canal correspondente utilizando o protocolo *byte write transfer;*

e) o NCAP lê os dados (*bytes*) utilizando o protocolo *byte read transfer* desde o *byte* mais significativo até o *byte* menos significativo;

f) o NCAP nega a linha NIOE;

g) o STIM nega a linha NACK.

e) PROTOCOLO DE TRANSFERÊNCIA DE QUADRO DE ESCRITA

a) o NCAP ativa a linha NIOE;

b) o NCAP espera o momento em que o STIM ativar a linha NACK;

c) o NCAP "escreve" o endereço funcional utilizando o protocolo *byte write transfer*;

d) o NCAP "escreve" o endereço do canal correspondente utilizando o protocolo *byte write transfer;*

e) o NCAP escreve os dados (*bytes*) utilizando o protocolo *byte write transfer* desde o *byte* mais significativo até o *byte* menos significativo;

f) o NCAP nega a linha NIOE;

g) o STIM nega a linha NACK.

Na Figura 3.9 ilustra-se os protocolos de transferência de quadro de leitura e escrita [3].

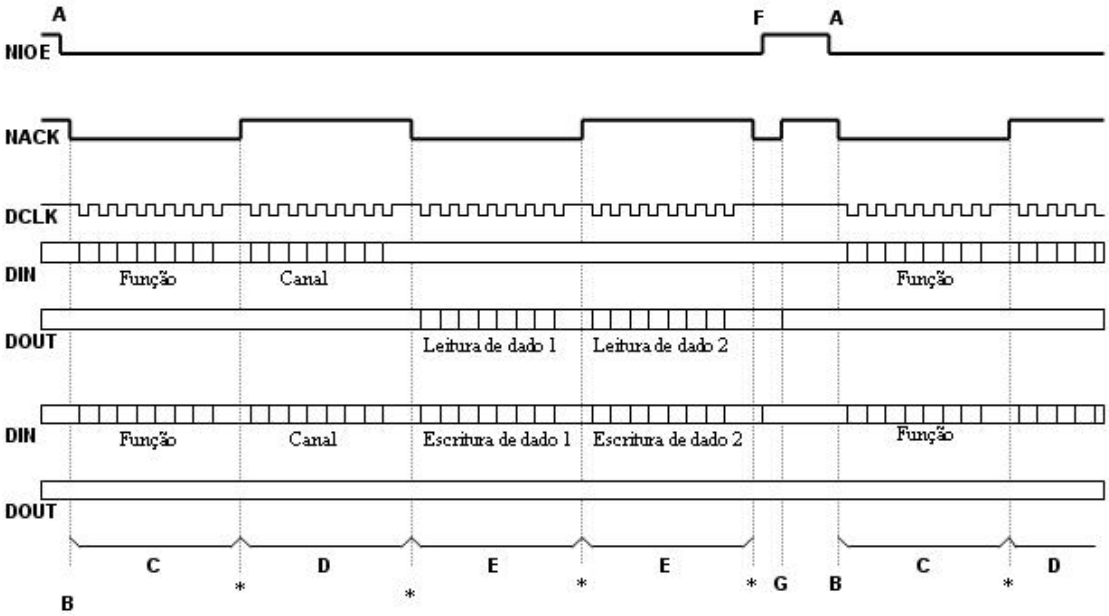


Figura 3.9 – Formas de onda dos protocolos de transferência de dados

3.6. Formatos TEDS

O aspecto mais interessante do padrão IEEE 1451.2, diz respeito à utilização dos formatos TEDS, que são as especificações do transdutor em formato eletrônico. Quando se faz necessário implementar algum tipo de sensor para uma aplicação determinada, é desejável e necessário conhecer as especificações do sensor. É assim que normalmente recorre-se à folha de dados do dispositivo (*data sheet*). Na folha de dados encontram-se diversas informações que caracterizam o sensor, quanto a seu desempenho e identificação.

Deste modo, disponibilizam-se informações relevantes tais como características de entrada/saída, valores máximos e mínimos, resolução, especificações de erro e outras informações tais como, fornecedor, número de série e versão. Se essas informações pudessem estar disponíveis em algum formato digital, obedecendo a uma forma padronizada e ficassem disponíveis para serem lidas através da rede, seria resolvido um conjunto de problemas associados ao interfaceamento de transdutores com redes de controle e barramentos de campo. Seriam facilitados as tarefas de manutenção e o trabalho com sistemas que utilizam diversas entradas/saídas. A idéia de "dados em formato eletrônico" baseia-se nesses aspectos. Deste modo, tentando implementar as idéias citadas acima, e visando resolver os problemas de interfaceamento entre transdutores e rede, é que foi criado o conceito TEDS do padrão IEEE 1451.2.

As TEDS são as especificações do transdutor escritas em formato eletrônico e armazenadas em um elemento de memória não volátil . Quando for feita uma solicitação por parte do NCAP, o microcontrolador do STIM acessa essas informações e as disponibiliza para a rede via interface padronizada (TII).

A inclusão dos formatos TEDS fornece o conceito de auto-identificação, uma ferramenta de importância fundamental para implementar o modo *plug and play*. Esta, juntamente com a utilização de uma interface padronizada para estabelecer a comunicação com o nó de rede, baseado em microprocessador, é a grande contribuição do padrão IEEE 1451.2 na área de sistemas de instrumentação eletrônica.

3.6.1. Especificação dos Formatos TEDS

O padrão possui uma cláusula específica que define os formatos TEDS. A especificação estabelece os formatos lógicos e o conteúdo das TEDS. Não se faz qualquer restrição quanto à forma física das TEDS. Porém, é necessário que se mantenham fisicamente associadas aos transdutores que descrevem, durante o funcionamento normal do STIM conectado ao NCAP,

e também, quando o STIM é retirado do NCAP para propósitos como transporte ou calibração.

O padrão define oito formatos TEDS diferentes. Dois desses formatos são obrigatórios e os seis restantes opcionais.

A seguir apresenta-se uma breve descrição dos oito formato TEDS especificados no padrão :

- **Bloco de dados Meta-TEDS**

Contém dados que descrevem o módulo STIM de forma global, daí o nome Meta. Meta é um prefixo de origem grega que significa aquele que pertence a uma entidade global ou que possui alguma coisa em comum ou em combinação com todos os membros que conformam o conjunto. Assim sendo, o Meta-TEDS inclui níveis de revisão, o identificador único do STIM, número de canais implementados, especificações de tempo, e outras informações considerando o STIM como um conjunto.

- **Bloco de dados Canal - TEDS**

Define o modelo funcional de transdutor, limites máximos e mínimos, unidades físicas do Sistema Internacional de Unidades, restrições de tempo e outras informações necessárias para descrever cada canal transdutor. Se for implementado mais do que um canal transdutor, deve-se repetir a estrutura para cada canal.

- **Bloco de dados Calibração –TEDS**

Opcional. Se vai ser realizado algum tipo de calibração, então este campo de dados deve ser incluído a fim de definir os coeficientes que serão utilizados na correção. O bloco define também as últimas data e hora de calibração e os intervalos de calibração requeridos por cada canal.

- **Bloco de dados Identificação Meta - TEDS**

Opcional. Este bloco fornece informações em formato de dados do tipo *string*, tais como nome do fabricante, número de modelo, número de série, códigos e descrição do produto.

- **Bloco de dados TEDS - Identificação de Canal**

Opcional. Este campo de dados inclui informações como as citadas no bloco anterior, mas, para cada canal em particular. Isto é útil quando se utilizam transdutores de diversos fabricantes.

- **Bloco de dados TEDS - Identificação de Calibração**

Opcional. Fornece informação legível pelo usuário e que é relevante para a calibração de cada canal. Esta estrutura deve-se repetir por cada canal implementado.

- **Bloco de dados TEDS - Aplicações Específicas do Usuário Final**

Opcional. A sua finalidade é fornecer uma forma de armazenar dados do tipo *string*, que não têm sido levados em conta nos outros blocos de dados. Por exemplo, a descrição da alocação do STIM no sistema.

- **Bloco de dados TEDS - Extensões Genéricas**

Opcional. Este bloco é fornecido para permitir extensões futuras em conformidade com o padrão e os setores industriais.

No Apêndice B são mostrados os tipos de dados IEEE 1451.2, contendo especificações de transdutores, representações de unidades e a maneira de expressar o tamanho de um bloco TEDS.

Um exemplo de TEDS é ilustrado no Apêndice C.

No Capítulo 4 aborda-se as principais características do microcontrolador MC68HC908QY4® da Motorola, que foi utilizado na implementação do STIM.

4- MICROCONTROLADOR DA FAMÍLIA MC68HC908QT/QY®

Os microcontroladores da família MC68HC908QT/QY® têm como características básicas: baixo custo, alto desempenho e baixa pinagem (8 ou 16 pinos). Todos os membros dessa família utilizam a unidade central de processamento CPU08, desenvolvida para arquitetura HC08, e estão disponíveis em uma variedade de módulos, tamanhos, tipos de memória e tipos de encapsulamento [21].

O dispositivo utilizado neste trabalho foi o MC68HC908QY4® que possui 4096 *bytes* de memória *FLASH*, conversor A/D de 4 canais de 8 *bits* e possui 16 pinos, conforme mostrado na Figura 4.1.

Figura 4.1 – Foto do Microcontrolador MC68HC908QY4®

As principais características deste componente são:

- Núcleo da CPU de alto desempenho HC08;
- Tensão de operação (VDD) de 5 V ou 3 V;
- 8 MHz @ 5 V (4 MHz @ 3 V) para operações do barramento interno;
- Oscilador interno ajustável;
    - 3,2 MHz para operações do barramento interno;
    - Capacidade de ajuste com registrador de 8 *bits;*
    - Precisão de ±25% sem utilizar ajuste;
    - Precisão de ±5% com o ajuste;
- Capacidade de acordar automaticamente da condição *STOP;*
- Programação *in-system* da memória *FLASH;*

- Segurança da memória *FLASH;*
- Memória *FLASH* programável na própria aplicação (com geração interna de tensões de apagamento/programação) com capacidade de 1,5 ou 4 Kbytes;
- 128 *bytes* de memória RAM;
- Temporizador com 2 canais de 16 *bits* (TIM);
- 4 canais de conversor A/D de 8 *bits;*
    - 13 linhas de entrada/saída (I/O) bidirecionais e mais 1 entrada;
    - 6 compartilhadas com função de interrupção do teclado e ADC;
    - 2 compartilhadas com os canais do temporizador;
    - 1 compartilhada com interrupção externa (IRQ);
    - 8 linhas extras no empacotamento de 16 pinos;
    - Capacidade de alta corrente dreno/fonte em todos os pinos;
    - *Pull-ups* selecionáveis individualmente em todos os pinos;
    - Habilidade de *tri-state* em todos os pinos;
- 6 *bits* de interrupção de teclado com característica de “despertar”;
- Inibição por baixa tensão (*Low-Voltage Inhibit-LVI*);
    - Ponto de disparo selecionável por software no registrador CONFIG;
- Sistema de proteção;
    - Computador operando adequadamente (Computer Operating Properly-COP);
    - Detecção de baixa tensão com *reset*;
    - Detecção de opcode ilegal com *reset*;
    - Detecção de endereço ilegal com *reset*;
- Interrupção externa assíncrona com *pull-up* interno (/IRQ) compartilhada com pino de I/O de propósito geral;
- *Reset* na energização (POR);
- *Pull-ups* internos no /IRQ e /RST para redução de componentes externos;
- Registradores de I/O mapeados em memória;
- Modos de redução de consumo-*STOP* e *WAIT;*

4.1 - Pinagem

Os componentes da família MC68HC908QT/QY® estão disponíveis em 8 e 16 pinos sendo que o utilizado neste trabalho está ilustrado na Figura 4.2 [22].

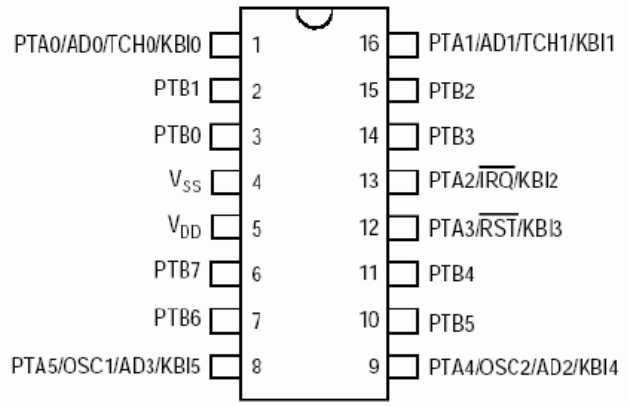


Figura 4.2 – Pinagem do Microcontrolador MC68HC908QY4

4.2 - Memória

Os componentes da família MC68HC908QY® possuem os seguintes tipos de memória implementados:

- 4096 *bytes* de memória *FLASH*;
- 128 *bytes* de memória RAM para dados;
- 48 *bytes* para vetores de interrupção programáveis e armazenados na *FLASH*;
- 416 *bytes* de memória ROM para monitor;
- 1536 *bytes* de rotinas para apagamento e programação da memória *FLASH*, na ROM.

4.3 - Placa de desenvolvimento M68EVB908Q®

A placa de desenvolvimento M68EVB908Q® é fornecida como parte do *Kit* de desenvolvimento e emulação para projetistas que utilizam unidades microcontroladoras da família MC68HC908QT/Y® [23].

A placa de desenvolvimento funciona com um PC (Computador Pessoal) para desenvolver aplicações, simular, emular e programar as *MCUs FLASH* Motorola.

O *software CodeWarrior*, da *Metrowerks*, o *software* ICS08, da *P&E Microcomputer Systems*, e outros *softwares* de terceiros, compatíveis com a interface MON08, podem ser

usados como *software* de controle no PC para desenvolver, depurar e gravar programas na *FLASH* na placa de desenvolvimento M68EVB908Q®.

A placa de desenvolvimento juntamente com o *software* de controle, permite edição, compilação, simulação *in-circuit*, emulação *in-circuit* e gravação da memória *FLASH*. Suas características incluem:

- Um ambiente integrado de desenvolvimento;
- Programação em linguagem C ou *Assembly*;
- Gravação da memória *FLASH*;
- Emulação em tempo-real incluindo carregar código em RAM, execução em tempo-real em RAM e *FLASH*, um *breakpoint* no modo de Emulação e múltiplos *breakpoints* no modo Simulação;
- A placa de desenvolvimento suporta somente encapsulamento DIP;
- Área para protótipos de aplicações do usuário;
- Conector para acesso aos pinos da MCU.

A placa de desenvolvimento é ilustrada na Figura 4.3.

Figura 4.3 – Foto da placa de desenvolvimento M68EVB908Q®.

5- IMPLEMENTAÇÃO DO STIM UTILIZANDO MICROCONTROLADOR DA MOTOROLA

5.1 - NCAP utilizado para emulação do STIM

O NCAP utilizado para emular o STIM foi desenvolvido pelo Grupo de Instrumentação Distribuída da UNESP-Ilha Solteira.

O *hardware* do NCAP foi desenvolvido com base nos recursos de um PC convencional e em um sistema embarcado denominado gerenciador de protocolo, implementado com dispositivos FPGA e conectado à porta paralela do PC. O *software* foi desenvolvido em Java [10],[25].

O gerenciador de protocolo é responsável por produzir um disparo via sinal NTRIG da TII, pertencente ao padrão IEEE 1451.2, para estabelecer o protocolo de comunicação com o STIM.

Os códigos enviados através da porta paralela do PC, são colocados nos barramentos de função, canal e escrita de acordo com a entrada de controle correspondente e enviados de forma sincronizada, para o STIM, por meio do sinal DIN da TII, que no caso do gerenciador de protocolo, constitui uma saída. O STIM envia os valores, através de DOUT e o gerenciador os coloca nos barramentos SAIDAPARALELA1[7..0] e SAIDAPARALELA2[7..0], do módulo gerenciador de protocolo, para que ocorra o processamento interno no NCAP.

O gerenciador de protocolo é um sistema embarcado que foi implementado com uma FPGA EPF10K20RC240-4 de 20.000 portas típicas, sendo que foram utilizados aproximadamente 400 elementos lógicos de um total de 1152 disponíveis e 33 entradas e saídas de um total de 189 disponíveis [13]. Pode-se observar, na Figura 5.1, o protocolo de comunicação implementado pelo gerenciador de protocolo.

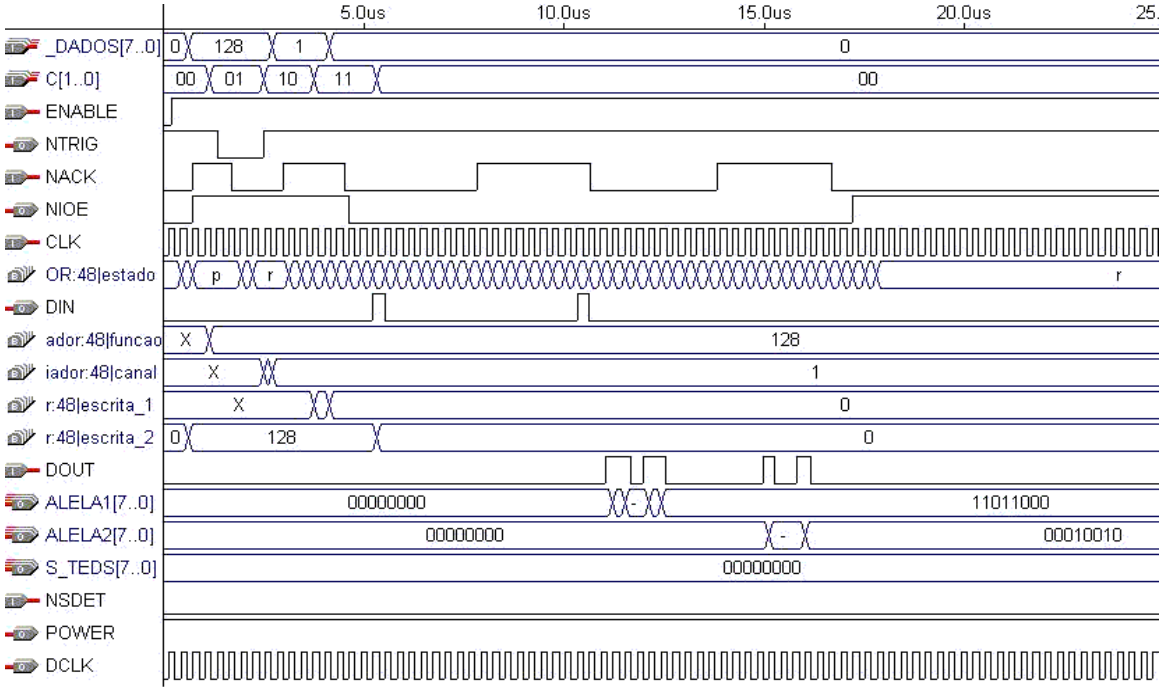


Figura 5.1– Protocolo 1451.2 entre o gerenciador de protocolo e o STIM

5.2 - Metodologia

O STIM é conectado ao gerenciador de protocolo através do conector da TII. O gerenciador é conectado à porta paralela do NCAP-PC que age como servidor em um ambiente de comunicação cliente-servidor. Através da rede, diferentes aplicativos clientes podem acessar as informações dos canais implementados no STIM. A Figura 5.2 mostra o diagrama da aplicação.

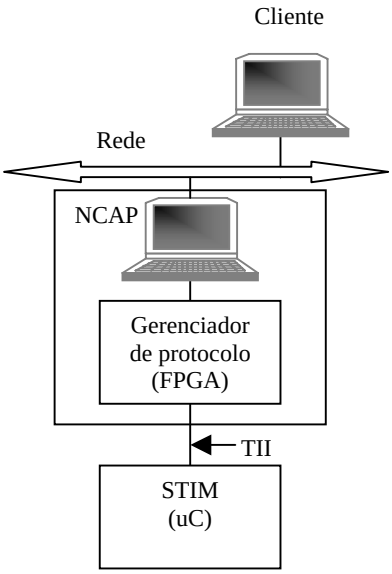


Figura 5.2 – Diagrama da aplicação

O STIM desenvolvido no presente trabalho foi implementado com o microcontrolador apresentado no capítulo 4.

5.3 – Diagrama de blocos: STIM – NCAP – REDE

Na Figura 5.3 são ilustrados o diagrama de blocos do STIM implementado, a relação entre os pinos do microcontrolador e os sinais da TII, as entradas analógicas utilizadas, as saídas digitais e as funções de *reset* e interrupção. O fluxograma geral do STIM é ilustrado na seção 5.9.

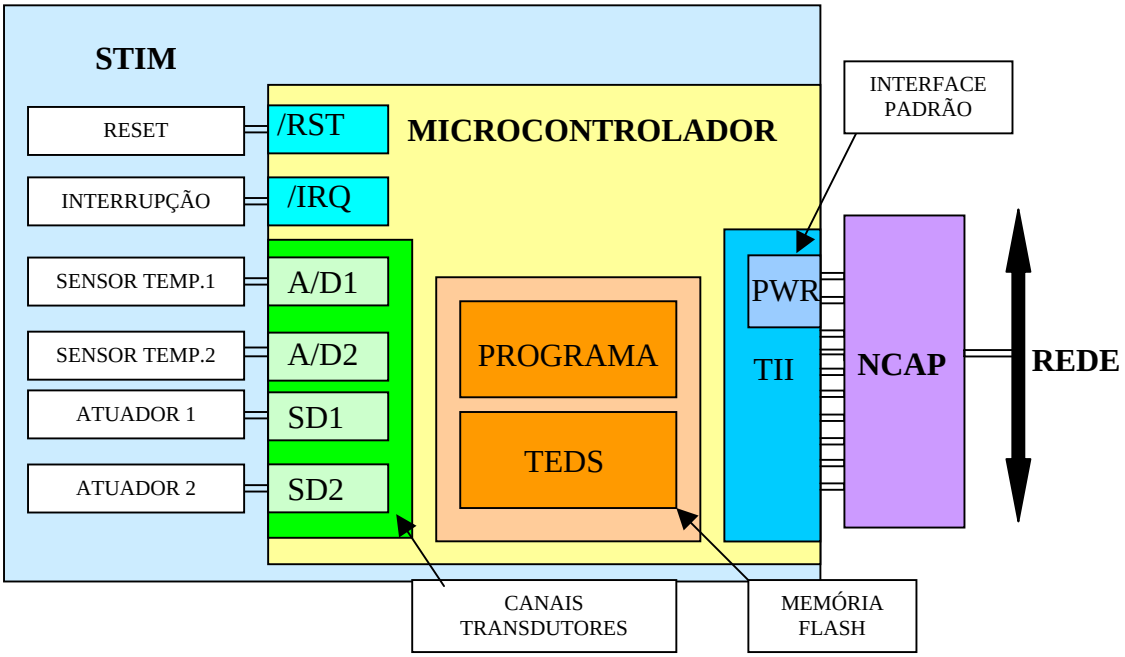


Figura 5.3 – Diagrama de blocos do STIM-NCAP-REDE

5.4 - Interface Padrão TII

Na Tabela 5.1 são mostrados os sinais da TII com os respectivos pinos do microcontrolador e a identificação de entrada ou saída de dados.

Tabela 5.1 – Descrição de sinais da Interface Independente para Transdutores(TII)

SINAIS	PINOS MICROCONT.	PINAGEM	I / O
DIN	PTB0	3	I
DCLK	PTB1	2	I
NIOE	PTB2	15	I
NTRIG	PTB3	14	I
+3/5V	VSS	1	I
GND	VDD	16	I
DOUT	PTB4	11	O
NACK	PTB5	10	O
NINT	PTB6	7	O
NSDET	PTB7	6	O

5.5 - Entradas Analógicas

As entradas analógicas do microcontrolador utilizadas estão descritas na Tabela 5.2.

Tabela 5.2 – Descrição das entradas analógicas

SINAIS	PINOS MICROCONT.	N^{o} do PINO	I/O
ENT. ANALOG.1	PTA0	13	I
ENT. ANALOG.2	PTA1	12	I

5.6 – Saídas Digitais

Na tabela 5.3 é feita a descrição das saídas digitais implementadas.

Tabela 5.3 – Tabela de descrição das saídas digitais

SINAIS	PINOS MICROCONT.	N^{o} do PINO	I/O
SD1	PTA4	5	O
SD2	PTA5	4	O

5.7 – Funções

As funções de Reset e interrupção são descritas na Tabela 5.4.

Tabela 5.4 – Tabela de descrição das funções adicionais

SINAIS	PINOS MICROCONT.	Nº do PINO	I/O
RESET	/RST	8	I
INTERRUPÇAO	/IRQ	9	I

5.8 - Configurações do Microcontrolador

Para que o microcontrolador funcione de acordo com o desejado é necessário o ajuste de alguns registradores, descritos abaixo:

a) Registrador CONFIG2

IRQPUD	IRQEN	R	OSCOPT1	OSCOPT0	R	R	RSTEN
0	1	0	0	0	0	0	1

Desta maneira o microcontrolador está com o *pull-up* interno conectado, com a função de requisição ativada no pino IRQ, oscilador do tipo interno e com a função *reset* habilitada no pino RST.

b) Registrador CONFIG1

COPRS	LVISTOP	LVIRSTD	LVIPWRD	LVI5OR3	SSREC	STOP	COPD
0	0	0	1	0	0	0	1

Com esta configuração *reset* de ciclo longo no modo *STOP*, desabilita Interrupção de baixa tensão durante o modo *STOP*, habilita o módulo RESET de inibição de baixa tensão, módulo de energia LVI desabilitado, LVI operando em Modo 3V, Modo *STOP* recuperado após 4096 ciclos de BUSCLKX4, Instrução *STOP* tratada como código Ilegal, Módulo COP desabilitado.

c) Registrador DDRA (Registros de direção de dados do Bit A)

Este registrador determina se os pinos, do Porto A do microcontrolador serão definidos como entrada (bit = 0) ou como saída(bit = 1).

R	R	DDRA5	DDRA4	DDRA3	DDRA2	DDRA1	DDRA0
0	0	1	1	0	0	0	0

Neste caso considera-se o porto A com quatro entradas e duas saídas.

d) Registrador DDRB (Registros de direção de dados do Bit B)

Este registrador determina se os pinos do Porto B do microcontrolador serão definidos como entrada (bit = 0) ou como saída (bit = 1).

DDRB7	DDRB6	DDRB5	DDRB4	DDRB3	DDRB2	DDRB1	DDRB0
1	1	1	1	0	0	0	0

O porto B é utilizado para implementar a Interface Independente para Transdutores (TII) que é de fundamental importância na implementação de um STIM que possui quatro sinais de entrada e quatro sinais de saída.

e) Registrador ADSCR(Registrador de controle e estado do conversor A/D)

COCO	AIEN	ADC0	CH4	CH3	CH2	CH1	CH0
0	0	1	0	0	0	0	0

O Modulo ADC possui 4 canais com entradas multiplexadas, resolução de 8 *bits*, conversão simples ou contínua, seleção de *clock* e *bit* de conversão completa ou conversão parada por interrupção.

A seleção de canais será necessária para escolher o sensor no qual se deseja fazer a leitura, no momento da conversão.

Neste trabalho foram considerados 2 canais analógicos situados nos canais 0 e 1.

f) Registrador ADICLK (Registrador de entrada de relógio A/D)

ADIV2	ADIV1	ADIV0	-	-	-	-	-
0	0	0	0	0	0	0	0

Para fazer a escolha da taxa de *clock* do conversor A/D é utilizada a Tabela 5.5.

Tabela 5.5 – Taxa de divisão do relógio ADC

ADIV2	ADIV1	ADIV0	Taxa de clock ADC
0	0	0	Clock Barramento/1
0	0	1	Clock Barramento/2
0	1	0	Clock Barramento/4
0	1	1	Clock Barramento/8
1	X	X	Clock Barramento/16

5.9 - Fluxograma

O fluxograma que implementa o STIM é dividido em várias subrotinas.

a) Subrotina INICIO

Faz o protocolo de comunicação, lê o canal e a função do NCAP e desvia para a subrotina referente à função escolhida.

b) Subrotina RD

Subrotina que faz a leitura de dados dos dois canais transdutores se o canal recebido pelo NCAP for zero, ou faz a leitura do canal se este for o canal 1 ou 2.

c) Subrotina RT

Subrotina que lê os dados do transdutor que estão gravadas na memória do STIM.

Se o canal escolhido pelo NCAP foi o zero, serão lidos os dados dos dois transdutores. Caso o canal escolhido for o 1,2,3 ou 4, apenas o dado do canal escolhido será transferido para o NCAP.

d) Subrotina ENV

Envia os dados bit a bit para o NCAP via DOUT. Para que ocorra sincronismo, cada bit é enviado durante um período de DCLK.

e) Subrotina LEITURADIN

Recebe os dados bit a bit do NCAP via DIN. Para que ocorra sincronismo, cada bit é recebido durante um período de DCLK.

5.9.1- Fluxograma que implementa o STIM

a) Inicialização e reconhecimento do STIM

Na Figura 5.4 observa-se como é realizada a atribuição inicial dos pinos NINT, NSDET e NACK e do valor zero no endereço de memória da função e do canal. Também é realizado o reconhecimento do disparo (NTRIG) vindo do NCAP. A descrição DLY2DCLK é a chamada de uma subrotina de atraso (*Delay*) de 2 ciclos de DCLK.

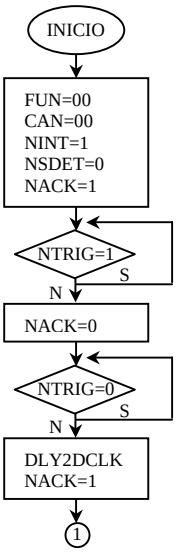


Figura 5.4 – Fluxograma de inicialização e reconhecimento do STIM.

b) Habilitação do STIM e leitura da função e do canal

Pode-se observar na Figura 5.5, após o disparo, o STIM deve esperar um sinal de habilitação (NIOE) para começar a receber dados. Recebido este sinal o STIM recebe através de DIN os valores da função e do canal e os armazena em um endereço de memória. A subrotina LEITURADIN faz a leitura de 8 *bits* e a subrotina DLY4DCLK realiza um atraso de 4 ciclos de DCLK.

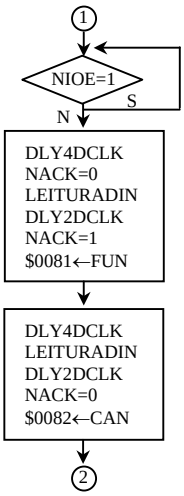


Figura 5.5 – Fluxograma de habilitação e leitura da função e canal.

c) Desvio para a função escolhida

A Figura 5.6 mostra a comparação do valor da função requerida que está na memória com as funções implementadas. Se a função for implementada ocorre o desvio para a subrotina, se não for o STIM é inicializado.

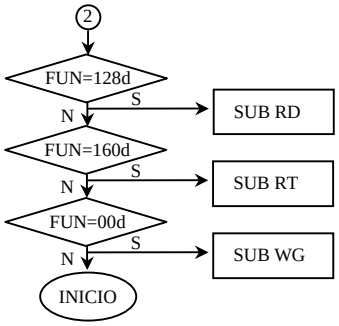


Figura 5.6 – Fluxograma de desvio para a função requerida

d) Subrotina leitura de dados (RD)

Nesta subrotina é comparado o canal escolhido pelo NCAP e realizado o desvio para o canal escolhido. Caso o canal seja o zero é realizada a leitura global, isto é, dos canais 1 e 2. Se for escolhido o canal 1 ou 2 o STIM faz a leitura do conversor A/D do canal escolhido, coloca o dado em uma memória auxiliar e envia este dado para o NCAP através de DOUT. O diagrama correspondente a este fluxograma é mostrado na Figura 5.7.

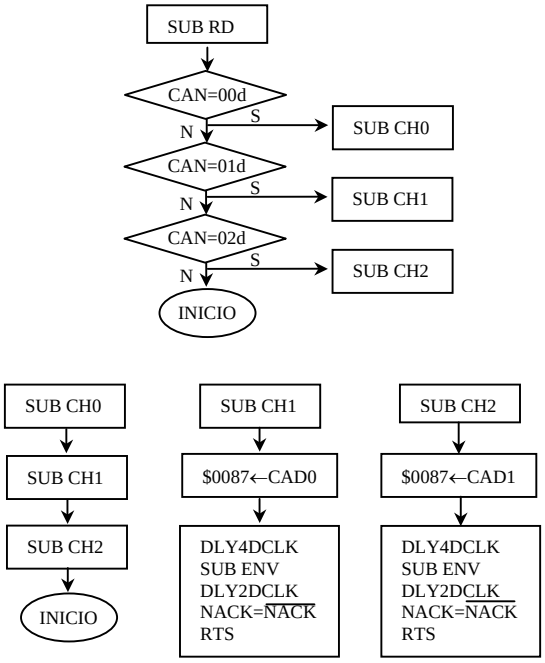


Figura 5.7 – Fluxograma de leitura de dados

e) Subrotina leitura de dados (RT)

Nesta subrotina ocorre a leitura de dados (TEDS) contidos na memória dos transdutores implementados. Como neste trabalho foram implementados dois canais para sensores e dois para atuadores é necessário ter 5 canais. Se o canal zero foi escolhido, serão enviados dados globais do STIM. Se o canal escolhido for outro, serão enviados dados específicos do canal. A Figura 5.8 ilustra este processo.

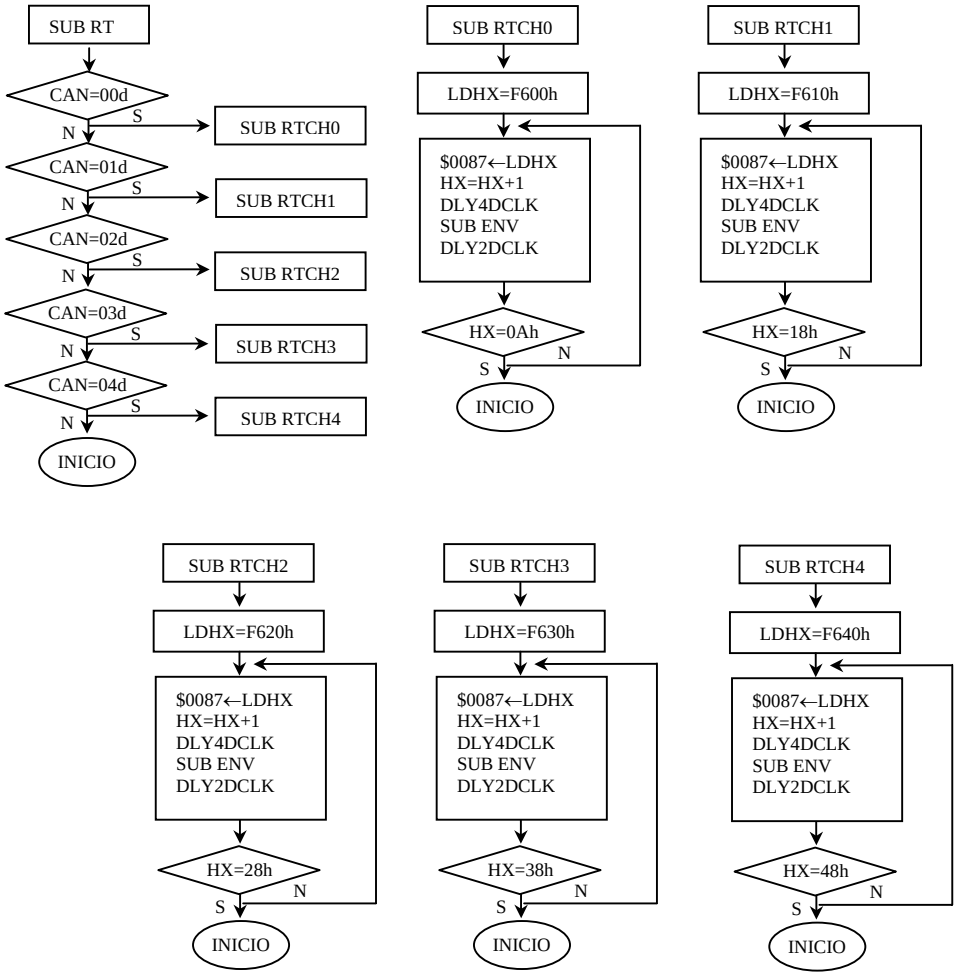


Figura 5.8 – Fluxograma de leitura de dados (TEDS).

f) Subrotina escrita de dados (WG)

Nesta subrotina é comparado o canal escolhido pelo NCAP e realizado o desvio para o canal escolhido. Caso o canal seja o zero é realizada a escrita global, isto é, dos canais 1 e 2. Se o canal escolhido for o 1 ou 2 o STIM faz a leitura do conversor A/D do canal escolhido, e atua através das saídas digitais (ventiladores) dependendo do valor lido no conversor. Na Figura 5.9 pode-se observar o fluxograma correspondente.

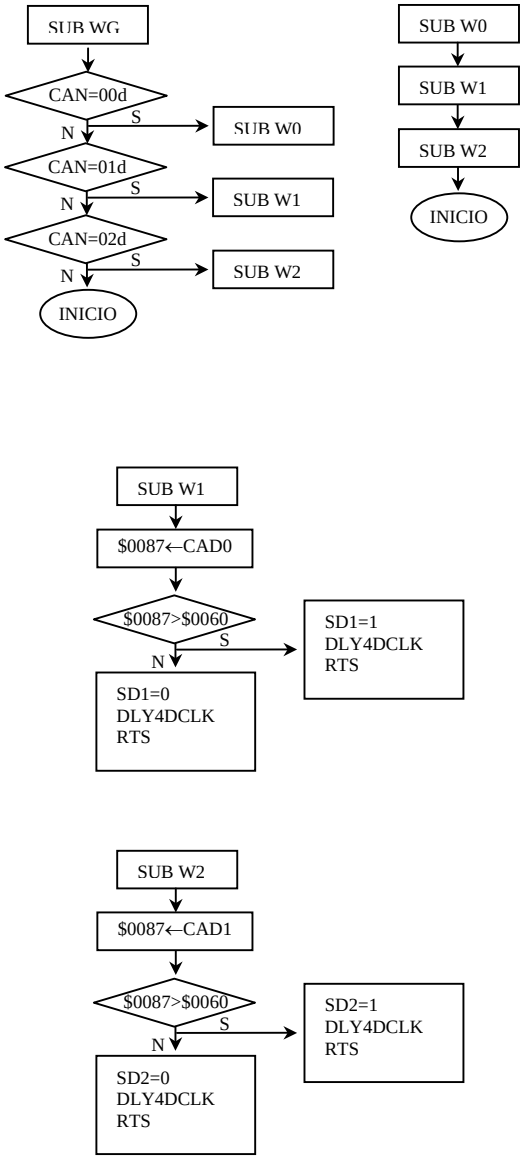


Figura 5.9 – Fluxograma de escrita de dados

5.9.2- Código fonte, em Assembly, que implementa o STIM

O código fonte foi desenvolvido em linguagem Assembly, e encontra-se em disquete anexo.

6 - RESULTADOS

6.1 - Resultados Experimentais

Visando a utilização de STIMs em ambientes industriais, decidiu-se simular uma câmara fria, na qual a temperatura não deve atingir valores elevados, com a finalidade de conservar os produtos. Para isso foram conectados dois sensores de temperatura, em duas entradas analógicas do microcontrolador, possibilitando assim a monitoração da temperatura em duas regiões distintas. Para fazer o controle da temperatura, foram implementados dois atuadores, que podem ser ventiladores ou compressores, que são capazes de baixar a temperatura ambiente quando ligados, através de dois sinais de saídas digitais do microcontrolador, que ligam e desligam de acordo com a temperatura lida na região.

Para verificar o funcionamento do circuito foi utilizado um osciloscópio digital (Minipa, modelo MO1150D) com a escala de tensão em 3 V/div e a escala de tempo de 5 s/div. O sinal de DCLK foi gerado por um gerador de ondas quadráticas com período de aproximadamente 1 segundo.

a) Resposta ao sinal de reconhecimento

Primeiramente testou-se a resposta do microcontrolador através do sinal de reconhecimento (NACK) a um disparo (NTRIG), enviado pelo NCAP, obtendo a resposta mostrada na Figura 6.1.

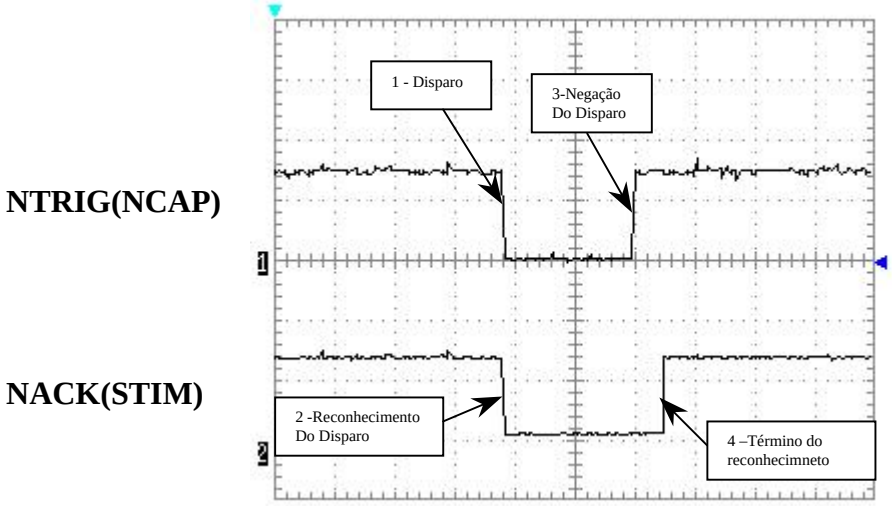


Figura 6.1 - Sinal de Disparo (NTRIG) e de Reconhecimento(NACK)

Nota-se que quando o sinal NTRIG vai para nível lógico “0” o sinal NACK também vai para nível lógico “0”, notificando que foi reconhecido o disparo. Assim, o STIM fica aguardando o sinal de NTRIG voltar para nível lógico “1”, conforme especificado na norma IEEE 1451.2 . Decorrido certo tempo o sinal de NTRIG volta para nível lógico “1” e após 2 períodos de DCLK o sinal NACK volta para o nível lógico “1”. Nesta situação, o STIM fica aguardando um sinal de habilitação por parte do NCAP, através do sinal NIOE, para poder transferir dados via TII.

b) Resposta ao sinal de habilitação

Realizado o reconhecimento, o STIM deve esperar um sinal de habilitação do NCAP (NIOE) indicando que está pronto para enviar/receber dados, conforme mostrado na Figura 6.2.

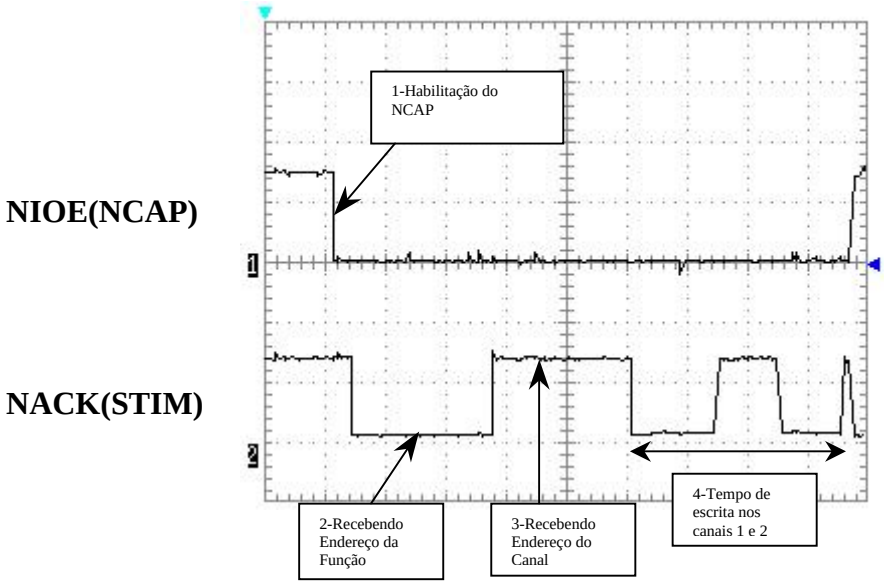


Figura 6.2 - Sinal de Habilitação(NIOE) e de Reconhecimento(NACK)

Após a habilitação (NIOE=0), o STIM produz uma mudança de estado no sinal NACK, pois uma vez realizado o reconhecimento, o sinal NACK é utilizado pelo STIM para limitar pacotes de 8 bits. Este fato é ilustrado na Figura 6.2. Desta maneira, o NCAP envia dados para o STIM, via DIN, e o STIM para o NCAP, via DOUT. Primeiramente, o NCAP envia, via DIN, o endereço de função, com o sinal NACK em nível “0”. Na seqüência, o NCAP envia o endereço de canal com o NACK em nível “1” e, finalmente, podem ser realizadas operações de escrita, usando a linha DIN, ou operações de leitura, usando o sinal DOUT.

c) Sincronismo do sinal NACK

Para que ocorra uma comunicação confiável entre o NCAP e o STIM é necessário existir um sinal de referência que é o sinal DCLK fornecido pelo NCAP. Quando o sinal NACK muda de estado é necessário ter um tempo de *hold off*, conforme especificado no padrão IEEE

1451.2. Neste trabalho, o tempo de *hold off* equivale a um período de DCLK antes de enviar/receber o primeiro *bit* . Este mesmo tempo é utilizado depois que o oitavo *bit* foi enviado. Sendo assim, para enviar/receber oito *bits* existem 10 períodos de DCLK em cada mudança de NACK. Este sincronismo é mostrado na Figura 6.3.

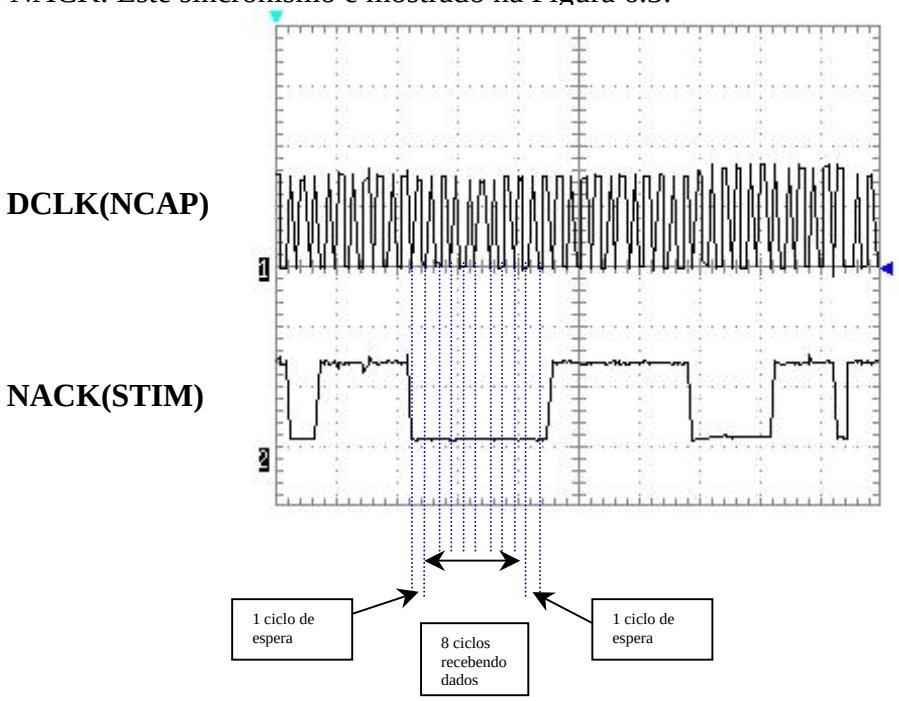


Figura 6.3 - Sinal de sincronismo(DCLK) e de Limite de pacote de dados(NACK)

d) Recebimento de dados

A partir da forma de onda mostrada na Figura 6.4, é possível observar que o NCAP está enviando 1 *byte* referente ao valor da função 128d(10000000b) e outro *byte* ao canal 1d(00000001b). Esta função está associada à leitura de dados do transdutor conectado ao canal 1. Após ter realizado a leitura do canal, o STIM começa a enviar dados via DOUT.

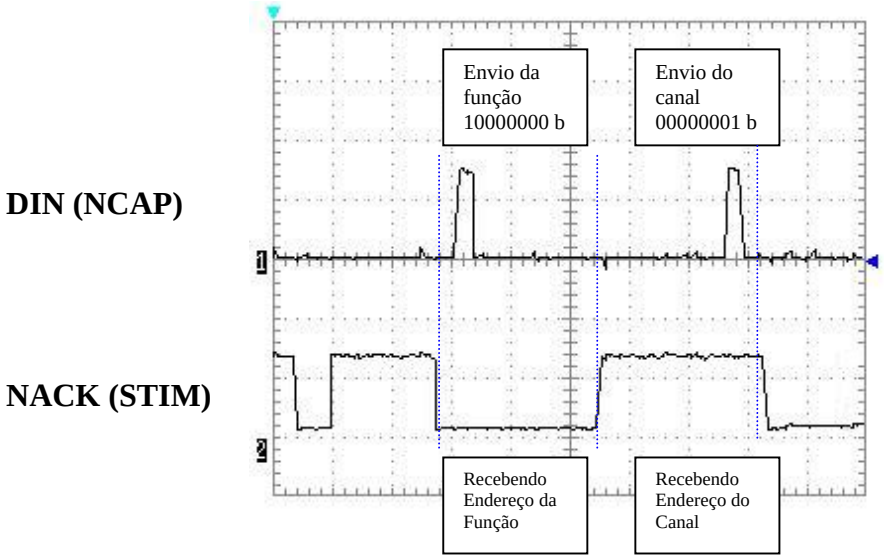


Figura 6.4 - Sinal DIN e de Limite de pacote de dados(NACK)

e) Envio de dados: função 128, canal 1

Para ler o valor de temperatura do sensor conectado no canal 1, o NCAP envia um *byte* (128d) referente à operação de leitura de dados e outro *byte* (01d) que identifica o canal. O NCAP recebe como resposta um *byte* (resposta da conversão do conversor A/D de 8 *bits*) que é enviado do STIM para NCAP, via DOUT.

Nas Figuras 6.5 e 6.6 ilustra-se o sinal DOUT para 2 valores de tensões diferentes no canal analógico 1, quando foi requerida pelo NCAP uma função de leitura de dados.

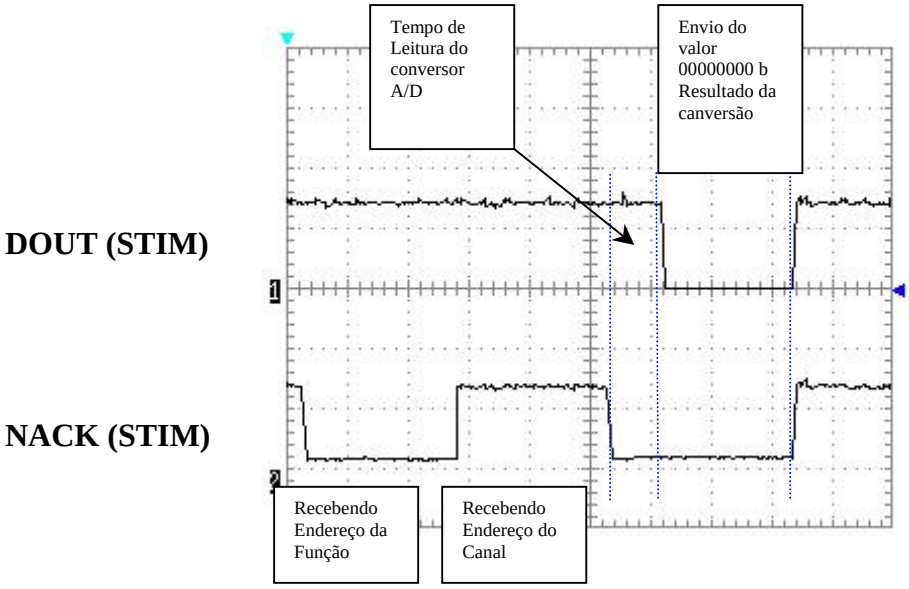


Figura 6.5 - Sinal DOUT e NACK para entrada analógica em 0V

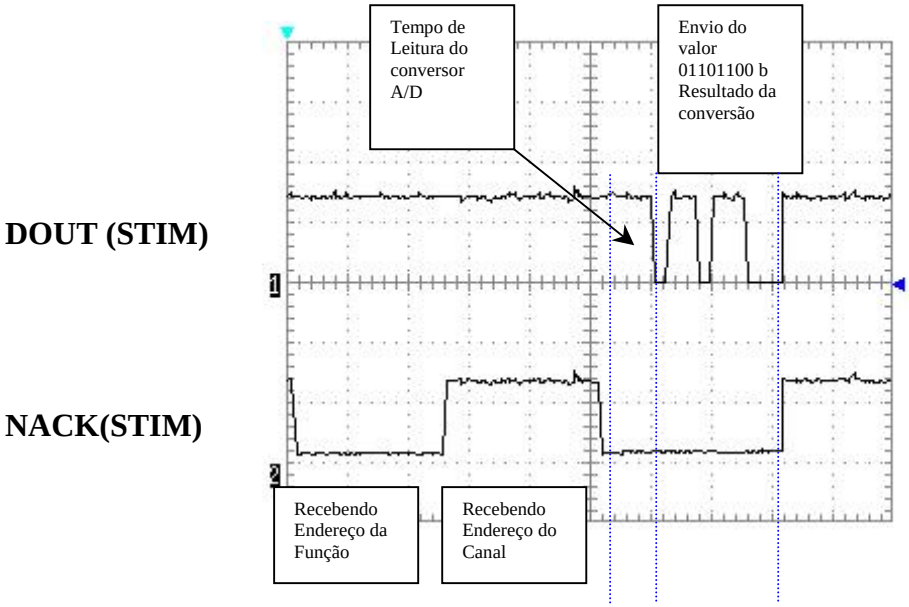


Figura 6.6 - Sinal DOUT e NACK para entrada analógica em 1,25V

Foi utilizado o sensor de temperatura LM35, de uso geral, com resolução de 10 mV/ºC e utilizado na faixa de 0 a 60 ºC. O conversor A/D é alimentado com 3 V, que corresponde a uma temperatura de 60ºC.

Para um valor de tensão de 0 V no canal analógico 1 obteve-se a resposta indicada na figura 6.5a, conforme era esperado (temperatura de 0 ºC).

Para um valor de tensão de 1,25 V no canal analógico 1, obteve-se a resposta indicada na figura 6.5b (01101100b) que em decimal corresponde a 108d e a uma temperatura de 25 ºC.

f) Envio de dados: função 160, canal 0 – Dado enviado AAh

Conforme especificado na seção 3.6, os formatos TEDS são gravados em intervalos de memórias e possuem a função de se auto-identificar em uma rede de dados, através de informações técnicas referentes a cada canal transdutor do microcontrolador.

Neste trabalho não se preocupou em definir os TEDS de maneira completa, mas foram deixados intervalos de memórias definidos para que os dados possam ser acrescentados futuramente.

O *software* utilizado para gravar os dados no microcontrolador não é uma versão completa, o que limitou o programa no momento do acréscimo dos canais transdutores.

Nas Figuras 6.7 e 6.8 ilustra-se o envio de vários pacotes de 8 *bits* quando foi solicitada a leitura de TEDS no canal 0 sendo que, com a finalidade de testar a transferência dos TEDS, o dado aleatório enviado (10101010b) foi escrito em diferentes enderecos de memória .

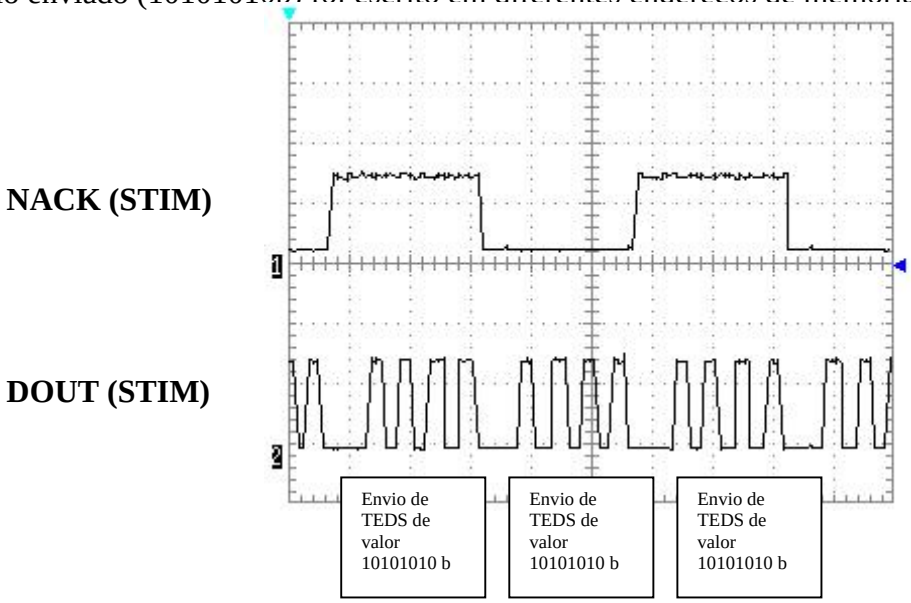


Figura 6.7- Sinal NACK e DOUT para leitura de TEDS.

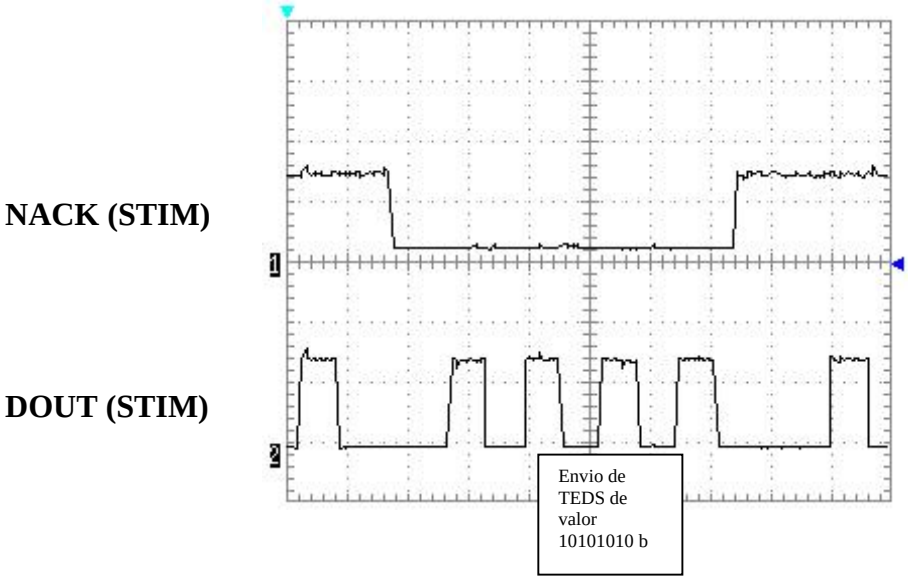


Figura 6.8- Sinal NACK e DOUT para leitura de TEDS(detalhado).

g) Escrita de dados: Função 0, canal 2

Na escrita de dados (Função = 00d), se o canal escolhido for o 00d o STIM faz a leitura do sensor de temperatura do canal 1 e aciona/inibe a atuação do ventilador 1. Terminado este processo (conjunto 1), o mesmo ocorre com o conjunto 2 (sensor de temperatura do canal 2 e ventilador 2). Caso o canal escolhido for o 01d somente será realizada a escrita no conjunto 1 e se o canal escolhido for o 02d será realizada a escrita do conjunto 2. O processo de atuação dos atuadores é ordenada pelo STIM quando a função é requisitada pelo NCAP através da função de escrita (Função = 00d). Na programação foi adotado o valor de referência de 60h que corresponde a 96d e equivale a 22,5 ºC. Se a temperatura estiver acima desta temperatura o atuador correspondente ao canal será setado. Caso estiver abaixo da temperatura, este continua em nível lógico “0”.

Na Figura 6.9, ilustra-se a atuação da saída Digital 2, a qual aciona o ventilador 2 por alguns segundos (definido via programação), devido ao sinal da entrada analógica 2 estar com um valor acima do valor estipulado, via programação. Nota-se que a atuação ocorreu após o NCAP requisitar a função de escrita no canal 2.

Uma outra lógica de funcionamento poderia ser utilizada como, por exemplo, a permanência da saída até a próxima requisição do NCAP.

É importante ressaltar que a lógica de funcionamento do processo não é definida pelo Padrão IEEE 1451.2, mas sim pelo projetista.

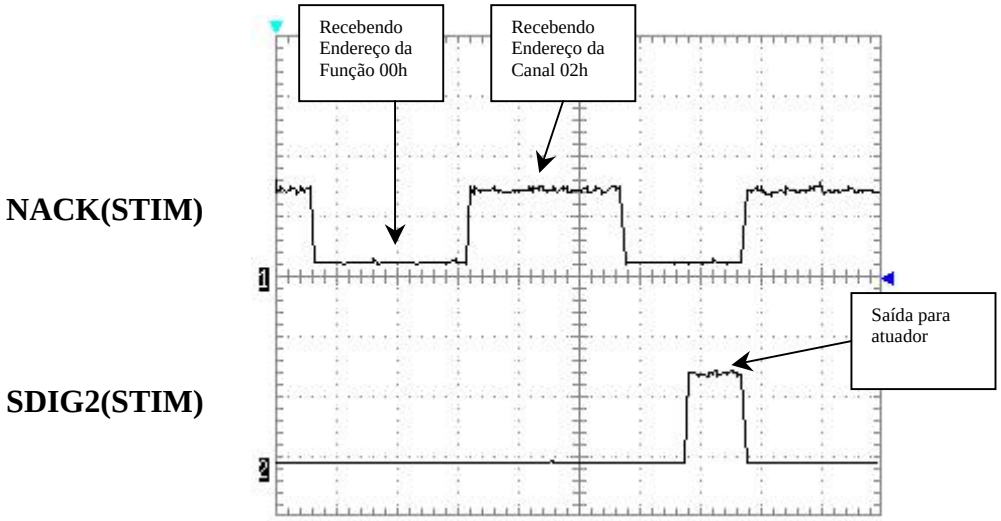


Figura 6.9 - Sinal NACK e Saída Digital 2.

6.2 - Montagem no Laboratório

A bancada de laboratório, com os materiais utilizados para fazer a implementação, é ilustrada na Figura 6.10. O computador localizado à esquerda é o servidor que, conectado ao FPGA de uso geral (gerenciador de protocolos), forma o NCAP. O computador do lado direito é o cliente, o qual se comunica com o servidor através da rede. O osciloscópio utilizado é digital, modelo MO1150D, da MINIPA.

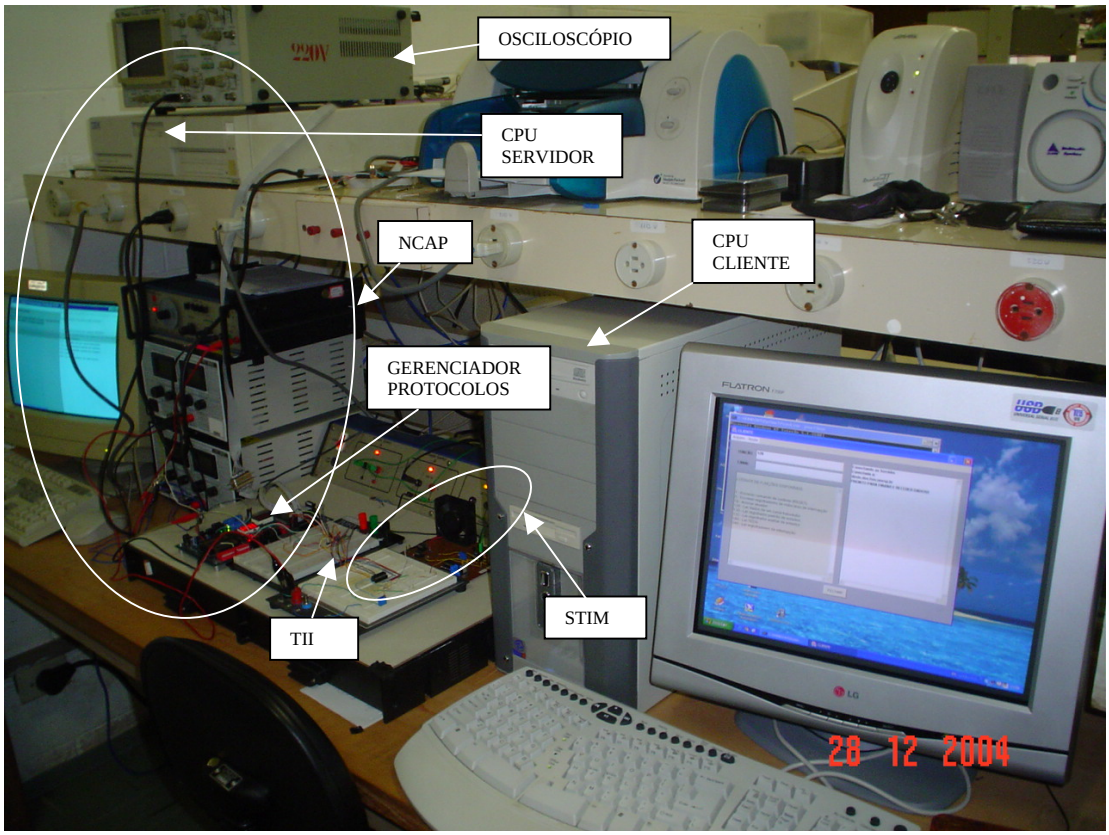


Figura 6.10 – Montagem realizada para simular STIM

A Figura 6.11 mostra o STIM, com um canal transdutor conectado ao sensor de temperatura LM35 e seus circuitos associados, um canal transdutor conectado ao ventilador e o gerenciador de protocolo, que faz parte do NCAP. Ambos os módulos estão vinculados através da TII.

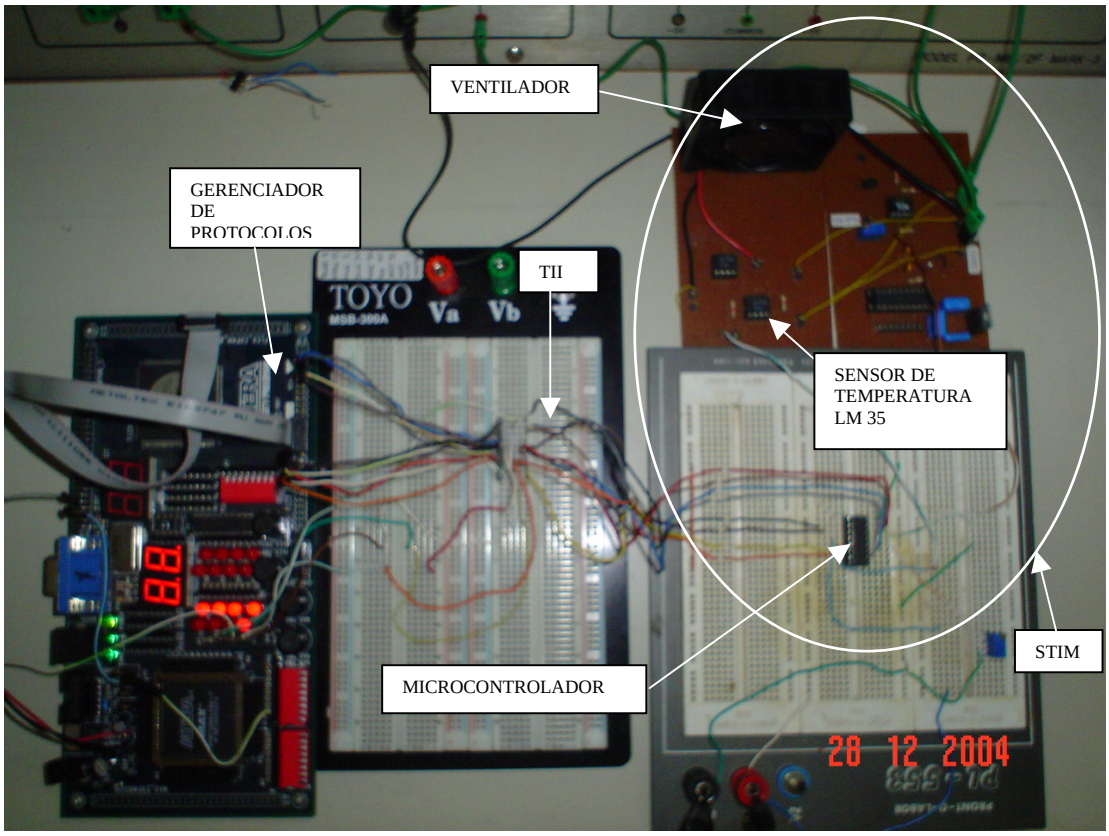


Figura 6.11 – Microcontrolador, gerenciador de protocolos e placas de sensor e atuador

6.3 – Característica "*Plug and Play*"

A característica "*plug and play*" foi avaliada quando se retirou um STIM construído com tecnologia FPGA, pelo Grupo de Instrumentação Distribuída, e instalado o STIM desenvolvido neste trabalho.

Constatou-se que o sistema identificou a mudança e se adequou à alteração realizada, mantendo o sistema em funcionamento.

7 - CONCLUSÕES GERAIS

7.1 – Conclusões

Neste trabalho foi descrita a implementação de um STIM com base na interface padrão IEEE 1451.2. O STIM foi implementado com um microcontrolador de uso geral MC68HC908QY4® da empresa Motorola, contendo quatro canais transdutores.

As principais vantagens deste microcontrolador são o baixo custo e o baixo consumo de energia, que é de aproximadamente 300 mW. Os microcontroladores de 16 pinos da Motorola tem um custo médio de aproximadammente US$1,36, o que garante um baixo custo de manutenção caso seja necessária a troca do componente. O usuário que utilizar um microcontrolador com um grande número de canais deverá utilizar o maior números de canais possíveis e, no caso de uma troca de componente, este terá um custo maior. Neste sentido, o microcontrolador usado neste trabalho permite implementar vários STIMs de poucos canais, o que pode ser muito útil, por exemplo, para montar uma pequena rede de automação predial ou residencial. O STIM de pequeno porte pode ser empregado na grande maioria dos processos industriais que necessitam de poucos canais sensores e atuadores.

Em contraste com aplicações já desenvolvidas, usando microcontroladores mais sofisticados quanto ao número de pinos e capacidade, ASICs, FPGAs e soluções comerciais, o emprego do microcontrolador MC68HC908QY4 é uma alternativa interessante para construir redes de transdutores com vários STIMs de poucos canais transdutores, devido a seu baixo custo [16], [17], [18], [20].

De acordo com os resultados obtidos, o projeto apresenta-se viável para a implementação de STIMs de pequeno porte, em conformidade com o padrão IEEE 1451.2, utilizando microcontrolador de uso geral. Em relação aos canais transdutores implementados, o STIM contém 2 sensores de temperatura e 2 atuadores (ventiladores), sendo que o ventilador 1 é ligado quando o sensor de temperatura 1 excede um valor de referência especificado, o mesmo ocorrendo com o conjunto 2.

A funcionalidade do STIM foi comprovada através de emulação realizada com um NCAP IEEE 1451.1, baseado em PC e em FPGA de uso geral, sendo que parte de *software* foi desenvolvida em Java. O monitoramento dos sinais, na interface TII, foi feito por meio de um osciloscópio digital.

Os TEDS foram emulados com valores aleatórios e em apenas 1 de um total de 5 blocos TEDS, devido a restrição da utilização da memória do microcontrolador (a utilização de uma versão demonstrativa do *software* limita a memória de programação em 1 *Kbyte* em um total de 4 *Kbyte*). Com a aquisição de uma licença do software a alocação dos canais restantes pode ser realizada.

A memória *flash* alocada do microcontrolador foi de ¼ do valor disponível o que mostra a viabilidade da utilização deste dispositivo.

Foi constatada a característica "*plug and play*" dos transdutores, retirando um STIM confeccionado com dispositivos FPGA modelo ACEX EP1K50TC144-3 e introduzindo o STIM implementado neste trabalho, verificando que o sistema continuou em funcionamento, sem a necessidade de qualquer reconfiguração.

Com base na experiência do trabalho, este tipo de STIM poderia ser utilizado industrialmente para fazer monitoramento de variáveis analógicas como temperatura, pressão, umidade, luminosidade, vazão, etc, além do controle de variáveis digitais, como por exemplo, o acionamento de sistemas de alarme, refrigeração e motores em geral.

A implementação das diretrizes do padrão IEEE 1451 poderá ser relevante para vários setores industriais, pois através da sua implementação, é possível obter importantes benefícios como redução de custos de sistemas devido à redução de cabeamento, flexibilidade para transmitir informações através de redes de comunicação e maior facilidade no que diz respeito a escalabilidade dos sistemas, sem efetuar grandes configurações.

7.2 – Sugestões de Trabalhos Futuros

Como sugestões de trabalhos futuros sugerem-se os seguintes:

- Implementação de novas funções;
- Implementação de campos TEDS opcionais definidos no padrão IEEE 1451.2 e gravá-los na memória;
- Exploração e possível implementação das diretrizes do padrão IEEE 1451.3, utilizando microcontrolador.

REFERÊNCIAS

[1] LEE, K. Sensor networking and interface standardization. In: INSTRUMENTATION AND MEASUREMENT TECHNOLOGY CONFERENCE - IMTC, 18, 2001, Budapest. **Proceedings of the...**, Budapest: IEEE, 2001, v.1, p. 147 – 152.

[2] NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY. IEEE 1451 website. Disponível em: <**http://ieee1451.nist.gov**>. Acesso em: 2004.

[3] INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS, Instrumentation and Measurement Society. **IEEE Standard for a Smart Transducer Interface for Sensors and Actuators – Transducer to Microprocessor Communication Protocols and Transducer Data Sheet (TEDS) Formats, (Std. 1451.2)**. Standards Board. NY:IEEE, 1997. 114 p.

[4] INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS, Instrumentation and Measurement Society. **IEEE Standard for a Smart Transducer Interface for Sensor and Actuators - Network Capable Application Processor (NCAP) Information Model (Std. 1451.1).** Standards Board. NY: IEEE, 1999. 341 p.

[5] INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS, Instrumentation and Measurement Society. **IEEE Standard for a Smart Transducer Interface for Sensor and Actuators - Digital Communication and Transducer Electronic Data Sheet (TEDS) Formats for Distributed Multidrop Systems, (Std. 1451.3)**. StandardsBoard. NY: IEEE, 2003. 185p.

[6] INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS. IEEE Standards website. Disponível em: <**http://standards.ieee.org/announcements/pr_14514.html**>. Acesso em: 2004

[7] INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS P1451.5 PROJECT. IEEE P1451.5 website. Draft Standard for a Smart Transducer Interface for Sensor and Actuators - Wireless Communication Protocols and Transducer Electronic Data Sheets. Disponível em: <**http://grouper.ieee.org/groups/1451/5/** >. Acesso em 2003.

[8] JOHNSON, R. N. Proposed IEEE standard P1451.0: defining the core features of smart sensors to facilitate broader adoption. Papers in PDF format, Telemonitor Inc., June 2003.
Disponível em: <**www.telemonitor.com/doc/dot2.pdf**>. Acesso em: 2003.

[9] SENSORSPORTAL. Sensorsportal website. Standardization. Disponível em: <**http://www.sensorsportal.com/HTML/standard.htm**>. Acesso em: 2004.

[10] ROSSI, S. R.; BATISTA, E. A.; CARVALHO, A. A.; SILVA, A. C. R. IEEE 1451 node development for connecting transducers to networks. In: CONFERÊNCIA INTERNACIONAL DE APLICAÇÕES INDUSTRIAIS - INDUSCON, 6, 2004. **Proceedings of the…**, Joinville, p.361-366.

[11] ROSSI, S. R.; BATISTA, E. A.; CARVALHO, A. A.; SILVA, A. C. R. Utilização da tecnologia de lógica programável na implementação de um módulo de interface para transdutores inteligentes (STIM) IEEE 1451.2. In: CONGRESSO BRASILEIRO DE AUTOMÁTICA - CBA, 15, 2004. **Proceedings of the...**, Gramado, Código: 630.

[12] BATISTA, E.A. **Emprego da Tecnologia Java para Implementar a Parte Lógica de um Processador de Aplicação com Capacidade de Operar em Rede de Comunicação (NCAP), em Conformidade com o Padrão IEEE 1451**. Dissertação (mestrado). UNIVERSIDADE ESTADUAL PAULISTA, FACULDADE DE ENGENHARIA, FEIS-UNESP. Ilha Solteira: [s.n], 2004.140 p.

[13] ROSSI, S.R. **Implementação de um Nó IEEE 1451, Baseado em Ferramentas Abertas e Padronizadas, para Aplicações em Ambientes de Instrumentação Distribuída.** Tese (Doutorado). UNIVERSIDADE ESTADUAL PAULISTA, FACULDADE DE ENGENHARIA, FE/IS-UNESP. Ilha Solteira: [s.n], 2005.213 p.

[14] LEE, K. IEEE 1451: A standard in support of smart transducer networking. In: INSTRUMENTATION AND MEASUREMENT TECHNOLOGY CONFERENCE-IMTC, 17, 2000, Baltimore. **Proceedings of the...**, Baltimore: IEEE, 2000, v.2, p. 525 – 528.

[15] EDISON, J.; WOODS, S.P. A research prototype of a networked smart sensor system. Papers in PDF format, Measurement Systems Department-Instruments and Photonics Laboratory (HP), August 1995. Disponível em: <**www.hpl.hp.com/techreports/95/HPL-95-91.pdf**>. Acesso em: 2004.

[16] CUMMINS, T.; BRANNICK, D.; BYRNE, E.; O´MARA, B.; STAPLETON, H.; CLEARY, J.; O'RIORDAN, J.; LYNCH, D.; NOONAN, L.; DEMPSEY, D. An IEEE 1451 standard transducer interface chip. In: INTERNATIONAL SOLID-STATE CIRCUITS CONFERENCE - ISSCC, 45, 1998. **Conference....**IEEE, 1998. p.276 - 277.

[17] CONWAY, P.; HEFFERNAN, D.; O'MARA, B.; BURTON, P.; MIAO, T. IEEE 1451.2: an interpretation and example implementation. In: INSTRUMENTATION AND MEASUREMENT TECHNOLOGY CONFERENCE - IMTC, 17, 2000, Baltimore. **Proceedings of the...**, Baltimore: IEEE, 2000, v.2, p. 535 - 540.

[18] CAMARA, L.; RUIZ, O.; SAMITIER, J. Complete IEEE-1451 node, STIM and NCAP, implemented for a CAN network. In: INSTRUMENTATION AND MEASUREMENT TECHNOLOGY CONFERENCE, IMTC, 17, 2000, Baltimore. **Proceedings of the...**, Baltimore: IEEE, 2000. v. 2, p.541- 545.

[19] CASTRO A.; RIESGO, T.; de la TORRE, E.; TORROJA, Y.; UCEDA, J. Custom hardware IEEE 1451.2 implementation for smart transducers. In: INDUSTRIAL

ELECTRONICS CONFERENCE - IECON, 28, 2002. **Proceedings of the...**, IEEE, 2002, v.4, p.2752-2757.

[20] RODRIGUES VIEGAS V.M. **Projeto e Implementação de um Sistema de Sensores Inteligentes Baseado na Norma IEEE 1451.** Dissertação (mestrado). UNIVERSIDADE TÉCNICA DE LISBOA – INSTITUTO SUPERIOR TÉCNICO. Lisboa: [s.n], 2003.223 p.

[21] NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY. IEEE 1451 website. Disponível em: <**http://ieee1451.nist.gov**>. Acesso em: 2004.

[22] Motorola. Treinamento de Microcontroladores Baseado na Família MC68HC908QT/Y. Motorola. Databook, 94 p.

[23] Motorola. Família MC68HC908QY/QT. Motorola. Data Sheet, Rev 0.1, 12/2002. 240 p.

[24] Motorola. Microcontroladores HC08, Curso Básico – Família MC68HC98QT/QY. Motorola. Tutorial. 2003. 93 p.

[25] BATISTA, E. A.; ROSSI, S. R.; SILVA, A. C. R.; CARVALHO, A. A.; KITANO, C. Implementation of a Java Language Program for Developing the Entire Network Capable Application Processor (NCAP) software. In: CONFERÊNCIA INTERNACIONAL DE APLICAÇÕES INDUSTRIAIS - INDUSCON, 6, 2004. **Proceedings of the...**, Joinville, p.382-387.

Apêndice A

Endereços de Funções mais Utilizados:

Tabela A1: Endereços de função, globais.

Adress	Function
0	Write global transducer data
1	Write global control command
3	Write triggered channel adress
5	Write global standard interrupt mask
96	Write global End-User's Application -Specific TEDS
128	Read global transducer data
130	Read global standard status
132	Read global auxiliary status
160	Reda Meta-TEDS
161	Read Meta-Identification TEDS
224	Read global End-User's Application -Specific TEDS

Tabela A2: Endereços de função (canais 1 até 255).

Adress	Function
0	Write channel transducer data
1	Write channel control command
5	Write channel standard interrupt mask
64	Write Calibration TEDS
65	Write Calibration Identification TEDS
96	Write End-User's Application -Specific TEDS
128	Read channel transducer data
130	Read channel standard status
132	Reda channel auxiliary status
160	Read Channel TEDS
161	Read Channel Identification TEDS
192	Read Calibration TEDS
192	Read Calibration Identification TEDS
224	Read End-User's Application -Specific TEDS

Apêndice B

Tipos de Dados IEEE 1451.2:

U8, U16, U32: *Unsigned integer*. Valores inteiros de tamanho 8, 16 e 32 bits respectivamente.

F32: *Float.* Número de ponto flutuante, precisão simples (IEEE Std 754-1985).

F64: *Double.* Número de ponto flutuante, precisão dupla (IEEE Std 754-1985).

STRING: Arranjo de *bytes* de caracteres.

UNITS: Representação descrita na tabela C1.

Letras Associadas:

C: Utiliza-se para contagem.

Exemplo: Expressar a quantidade de *bits* significativos necessários para caracterizar um dado associado a um canal, como especificado no campo 16 do bloco de dados Canal-TEDS. Se é utilizado um conversor de 8 *bits*, o valor neste campo será 8.

L: Utiliza-se para caracterizar tamanho.

Exemplo: Expressar o tamanho em *bytes* de um bloco TEDS completo.

E: Enumeração. Indica como codificar a informação que pode ser interpretada pelo usuário, de uma maneira conveniente para ser armazenada em forma eletrônica.

Exemplo: Enumeração dos tipos de canais pertencente ao campo oito do bloco de dados Canal-TEDS.

Valor	Significado
0	Sensor
1	Atuador
	Outros

Tabela B1: Representação de Unidades IEEE 1451.2:

Field#	Description	# bytes
1	ENUMERATION 0: Unit is described by the product of SI base units raised to the powers recorded in field 2 trough 10.	1
	1: Unit is U/U, where U is described by the product SI base units raised to the powers recorded in fields 2 trough 10.	
	2: Unit is $\log_{10}$ (U), where U is described by the product SI base units raised to the powers recorded in fields 2 trough 10.	
	3: Unit is $\log_{10}$ (U/U), where U is described by the product SI base units raised to the powers recorded in fields 2 trough 10.	
	4: The associated quantity is digital data (e.g a bit vector) and has no unit. Fields 2-10 shall be set to 128.	
	5: The associated physical quantity is represented by values on an arbitrary scale (e.g., hardness). Fields 2-10 are reserved, and shall be set to 128.ç	
	6-255: Reserved	
2	(2* <exponent of radians>) + 128	1
3	(2* <exponent of steradians>) + 128	1
4	(2* <exponent of meters>) + 128	1
5	(2* <exponent of kilograms>) + 128	1
6	(2* <exponent of seconds>) + 128	1
7	(2* <exponent of amperes>) + 128	1
8	(2* <exponent of kelvins>) + 128	1
9	(2* <exponent of moles>) + 128	1
10	(2* <exponent of candelas>) + 128	1

Exemplos:

- Aceleração em metros*segundos^{-2}

Campo TEDS: 0,128,128,130,128,124,128,128,128,128

Enumeração :1, unidade SI: metro
[2*(exponente de metros)] +128 =
(2*1)+128=130

Enumeração :-2, unidade SI: segundo
[2*(exponente de segundos)] +128 = (2*(-2))+128=124

Representação lógica: 0,0,0,1,0,-2,0,0,0,0

- Adimensional metro/metro,

Enumeração: 1, unidade SI: metro

Campo TEDS: 0,128,128,130,128,128,128,128,128,128

Representação lógica: 1,0,0,1,0,0,0,0,0,0

Apêndice C

Exemplo TEDS: 1canal transdutor, sensor de pressão:

O exemplo a seguir é adaptado da referência [16].

Características:

- Material cerâmico.
- Saída analógica 0 - 5 V, correspondentes a 0 - 20.648.190 Pa de pressão de entrada.
- Tempo de resposta: 10 ms, não requer um tempo de *warm-up* apreciável.
- Não linearidade máxima: 0.56% de *Vsupply*.
- Utiliza-se um conversor A/D de 12 bits cujo ciclo de conversão é de 75 μs.
- O STIM utiliza um microcontrolador PIC de 8 bits, com uma memória EEPROM de 4K.

Tabela C1 – Exemplo TEDS

Field nº	META-TEDS Description	Type	Nº of bytes
TEDS version constant related data sub-block			
1	Meta-TEDS Lenght	U32L	48
2	IEEE 1451 Standards Family Working Group Number	U8E	2
3	TEDS Version Number	U8E	1
Identification related data sub-block			
4	Globally Unique Identifier	UUID	-
Data structure related data sub-block			
5	CHANNEL_ZERO Industry Calibration TEDS Extension Key	U8E	0 (N0NE)
6	CHANNEL_ZERO Industry Nonvolatile Data Field Extension Key	U8E	0 (N0NE)
7	CHANNEL_ZERO Industry TEDS Extension Key	U8E	0 (N0NE)
8	CHANNEL_ZERO End-User's Application-Specific TEDS Key	U8E	0 (N0NE)
9	Number of implemented channels	U8C	1
10	Worst-Case Channel Data Model Lenght	U8C	2
11	Worst-Case Channel Data Repetitions	U16C	1
12	CHANNEL_ZERO writable TEDS Lenght	U32C	0

Timing related data sub-block			
13	Worst-Case Channel Update Time (t_{wu})	F32	2.00E-05
14	Global Write Setup Time (t_{gws})	F32	0
15	Global Read Setup Time (t_{grs})	F32	8.00E-05
16	Worst-case Channel Sampling Period (t_{wsp})	F32	2.00E-04
17	Worst-case Channel Warm-Up Time	F32	1
18	Command Response Time	F32	5.00E-04
19	STIM Handshake Timing (t_{hs})	F32	-
20	End-of-Frame Detection Latency (t_{lat})	F32	-
21	TEDS Hold-Off Time (t_{th})	F32	-
22	Operational Hold-Off Time (t_{oh})	F32	-
23	Maximum Data Rate	U32C	-
Channel grouping related data sub-block			
24	Channel Groupings Data Sub-Block Lenght	U16L	0
25	Number of Channel Groupings = G	U8C	-
Fields 26-28 are repeated G times, once for each group			
26	Group Type	U8E	-
27	Number of Group Members = N	U8C	-
28	Member Channel Numbers List = M(N)	Array of U8E	-
Data integrity data sub-block			
29	Checksum for Meta-TEDS	U16C	62856
Field nº	**CANAL-TEDS Description**	**Type**	**Nº of bytes**
Data structure related data sub-block			
1	Channel TEDS Lenght	U32L	80
2	Calibration Key	U8E	1 (FIXED)
3	Channel Industry Calibration TEDS Extension Key	U8E	0 (NONE)
4	Channel Industry Nonvolatile Data Fields Extension Key	U8E	0 (NONE)
5	Channel Industry TEDS Extension Key	U8E	0 (NONE)
6	Channel End-User's Application-Specific TEDS Key	U8E	0 (NONE)
7	Channel Writable TEDS Lenght	U32C	0
Transducer related data sub-block			
8	Channel Type Key	U8E	0 (SENSOR)
9	Physical Units	UNITS	Pa (0,128,128, 126,130,124, 128,128,128, 128)
10	Lower Range limit	F32	0
11	Upper Range Limit	F32	20648190
12	Worst-Case Uncertainty	F32	206842
13	Self-Test Key	U8E	0 (NONE)
Data converter related data sub-block			
14	Channel Data Model	U8E	0 (N BYTE)
15	Channel Data Model Lenght	U8C	2
16	Channel Model Significant Bits	U16C	12
17	Channel Data Repetitions	U16C	1

18	Series Origin	F32	0
19	Series Increment	F32	0
20	Series Unit	UNITS	-
Timing related data sub-block			
21	Channel Update Time (t_u)	F32	2.00E-05
22	Channel Write Setup Time (t_{ws})	F32	0
23	Channel Read Setup Time (t_{rs})	F32	8.00E-05
24	Channel Sampling Period (t_{sp})	F32	2.00E-04
25	Channel Warm-Up Time	F32	1
26	Channel Aggregated Hold-Off-Time (t_{ch})	F32	-
27	Timing Correction	F32	0
28	Trigger Accuracy	F32	5.00E-06
Event sequence options field			
29	Event Sequence Options	U8E	0
Data Integrity data sub-block			
30	Checksum for Channel TEDS	U16C	59968

;PROGRAMA EM ASSEMBLY DO MICROCONTROLADOR

```
*************************************************************
* This stationery is meant to serve as the framework for a   *
* user application. For a more comprehensive program that    *
* demonstrates the more advanced functionality of this       *
* processor, please see the demonstration applications       *
* located in the examples subdirectory of the                *
* Metrowerks Codewarrior for the HC08 Program directory      *
*************************************************************

                                        XDEF Entry,main,t_isr,trim_val


                    Include 'qtqy_registers.inc'    ; For the QT1, QT2, QT4, QY1, QY2, QY4

t25ms           EQU             15624               ;Constante definida 15624

        org $FFC0                                   ;Inicio da Flash

trim_val:  DC.B $FF                                  ; here we set the FLASH trim to a default value.
                                                    ; DO NOT change this value, as the trim will not be
                                                    ; automatically callibrated by the programming interface if
                                                    ; this value is anything other than $FF

DEFAULT_RAM                                         SECTION SHORT

; Place RAM Variables here

DEFAULT_ROM                                         SECTION


*************************************************************
* Init_Timer - Turns on timer 1 Channel 1 for an Output      *
*           Compare in approximately 2ms. The timer       *
*           interrupt service routine continually sets    *
*           the next interrupt to occur 2ms later.        *
*************************************************************
Init_Timer:
     mov   #$36,TSC      ; Timer 1 - Cleared + Stopped.
                         ;    Clicks once every 64 BUS Cycles
                         ;    200t Clicks ~ 2ms

     mov   #$0, TCH1H   ; Set Output Compare to happen 77T clicks
     mov   #$4D,TCH1L   ; after we start the timer. (~2ms). The
                         ; timer interrupt will set OC for another ~2ms.

     mov   #$54,TSC1    ; Timer 1 Channel 1
                        ; Set for Output Compare operation.

     mov   #$06,TSC     ; Start the timer
     rts
*************************************************************
* Main_Init - This is the point where code starts executing  *
*           after a RESET.                                  *
```

```
*************************************************************
Entry:
main:
      lda $FFC0            ; load the TRIM value stored in FLASH
      cmp #$FF
      beq no_trim
      sta osctrim            ; if the TRIM value has been calibrated and
                             ; stored previously,use this stored value.
no_trim:
      rsp
      clra
      clrx
      bsr Init_Timer
      cli                  ; Allow interrupts to happen

main_loop


                LDA          #01000001B    ;Carrega acumulador
                STA          CONFIG2       ;Pullup conectado, Habilita interrup,
                                           ;oscilador INterno, reset ativo no pino /RST


                LDA          #00010001B    ;Reset com ciclo longo quando no Modo STOP
                                           ;LVI disabilitado durante Modo STOP
                                           ;Reset do modulo LVI habilitado
                                           ;Modulo de energia LVI disabilitado
                                           ;LVI opera em 3V
                                           ;Modo STOP após 4096 ciclos de BUSCLKX4
                                           ;Instrução de STOP tratada como um codigo ilegal
                                           ;Módulo COP desabilitado
                STA          CONFIG1

                LDA          #$FF
                STA          PORTA
                LDA          #$FF
                STA          PORTB
                LDA          #$30    ;DEFINE PORTO A COMO ENTRADAS ANALOGICAS E
                             SAIDAS EM PTA4 e PTA5
                STA          DDRA
                LDA          #$F0    ;DEFINE PORTO B COM 4 ENTRADAS E 4 SAIDAS ;
                STA          DDRB
                LDA          #$FF    ;INICIALIZAÇÃO DO CONVERSOR A/D
                STA          ADSCR
                LDA          #$00
                STA          ADICLK


;ALIMENTAÇÃO
;VDD -P01- 3 OU 5V           (BRANCO)
;VSS -P16- GND               (VERDE)

;CONFIGURAÇÃO DA PINAGEM DO MICROCONTROLADOR
;
;PTA0 -P13- ENTRADA ANALÓGICA 1
;PTA1 -P12- ENTRADA ANALÓGICA 2
```

```
;PTA2 -P09- /INTERRUPÇÃO
;PTA3 -P08- /RESET
;PTA4 -P05- SAIDA DIGITAL 1
;PTA5 -P04- SAIDA DIGITAL 2


;TII - INTERFACE INDEPENDENTE PARA TRANSDUTORES
;ENTRADAS
;PTB0 -P15- DIN        (VERMELHO)
;PTB1 -P14- DCLK       (MARROM)
;PTB2 -P11- NIOE       (AZUL)
;PTB3 -P10- NTRIG      (CINZA)

;SAIDAS
;PTB4 -P07- DOUT       (LARANJA)
;PTB5 -P06- NACK       (AMARELO)
;PTB6 -P03- NINT       (VIOLETA)
;PTB7 -P02- NSDET      (PRETO)

INICIO:

;DEFINIÇÃO DE VALORES NA MEMÓRIA RAM
            LDA         #$00
            STA         $0080       ;VALOR AUXILIAR
            LDA         #$00
            STA         $0081       ;ENDEREÇO DA FUNÇÃO
            LDA         #$00
            STA         $0082       ;ENDEREÇO DO CANAL
            LDA         #$80
            STA         $0083       ;FUNÇÃO LEITURA DADOS(128d)
            LDA         #$00
            STA         $0084       ;0 - FUNÇÃO ESCRITA(0d)
            LDA         #$01
            STA         $0085       ;1
            LDA         #$02
            STA         $0086       ;2
            LDA         #$00
            STA         $0087       ;ENDEREÇO DO DADO A SER ENVIADO
            LDA         #$A0
            STA         $0088       ;FUNÇÃO LEITURA TEDS(160d)
            LDA         #$0A
            STA         $0089
            LDA         #$30        ;VALOR QUE SE DEVE COMPARAR PARA SAIDA
            STA         $0090
            LDA         #$03
            STA         $0091       ;3
            LDA         #$04
            STA         $0092       ;4
;EXEMPLO ALEATÓRIO DE TEDS CANAL ZERO
            LDA         #$AA
            STA         $00A0
            LDA         #$AA
            STA         $00A1
            LDA         #$AA
            STA         $00A2
            LDA         #$AA
```

```
                STA     $00A3
                LDA     #$AA
                STA     $00A4
                LDA     #$AA
                STA     $00A5
                LDA     #$AA
                STA     $00A6
                LDA     #$AA
                STA     $00A7
                LDA     #$AA
                STA     $00A8
                LDA     #$AA
                STA     $00A9

;INICIALIZAÇÃO DO STIM
                BSET    6,PORTB         ;NINT=1
                BCLR    7,PORTB         ;NSDET=0
                BSET    5,PORTB         ;NACK=1
                BCLR    4,PORTA         ;SD1=0
                BCLR    5,PORTA         ;SD2=0

;RECONHECIMENTO DO STIM
LOOP0:          BRSET 3,PORTB,LOOP0     ;SE NTRIG=1 VAI PARA LOOP0
                BCLR    5,PORTB         ;NACK=0
LOOP1:          BRCLR 3,PORTB,LOOP1     ;SE NTRIG=0 VAI PARA LOOP1
                JSR     SUB1PULSOS
                BSET    5,PORTB         ;NACK=1

;HABILITAÇÃO DO STIM PELO NCAP
LOOP2:          BRSET 2,PORTB,LOOP2     ;SE NIOE=1 VAI PARA LOOP2
                JSR     SUB2PULSOS

;LEITURA DA FUNÇÃO
                BCLR    5,PORTB         ;NACK=0 - RECONHECIMENTO SINAL HABILITAÇAO
                JSR     SUB1PULSOS
                JSR             LEITURADIN
                JSR             SUB1PULSOS
                BSET    5,PORTB         ;NACK=1 - RECONHECIMENTO LIMITE PACOTE
                LDA             $0080   ;MEMÓRIA AUXILIAR
                STA             $0081   ;ENDEREÇO FUNÇÃO

;LEITURA DO CANAL
                JSR             SUB1PULSOS
                LDA             #$00
                STA             $0080
                JSR             LEITURADIN
                JSR             SUB1PULSOS
                BCLR    5,PORTB         ;NACK=0 - RECONHECIMENTO LIMITE PACOTE
                LDA             $0080   ;MEMÓRIA AUXILIAR
                STA             $0082   ;ENDEREÇO CANAL

;FUNÇÕES
                LDA             $0081   ;CARREGA A COM VALOR FUNÇÃO
                CMP             $0083
                BEQ             RD      ;SE IGUAL AO CONTEUDO
                                        ;DA MEMORIA $0083 DESVIA PARA RD
```

```
            LDA         $0081       ;CARREGA A COM VALOR FUNÇÃO
                        CMP         $0088       ;SE IGUAL AO CONTEUDO
            BEQ         RT1         ;DA MEMORIA $0088 DESVIA PARA RT
            LDA         $0081       ;CARREGA A COM VALOR FUNÇÃO
            CMP         $0084       ;SE IGUAL AO CONTEUDO
            BEQ         WG          ;DA MEMORIA $0084 DESVIA PARA WG
            JSR         INICIO

INICIO1:    JSR INICIO
RT1:        JSR RT

WG:
            LDA         $0082       ;VALOR CANAL
            CMP         $0084       ;SE FOR 0 VAI PARA CHO
            BEQ         W0
            LDA         $0082       ;VALOR CANAL
            CMP         $0085       ;SE FOR 1 VAI PARA CH1
            BEQ         W1
            LDA         $0082       ;VALOR CANAL
            CMP         $0086       ;SE FOR 2 VAI PARA CH2
            BEQ         W2
            Jsr         INICIO

W0:                                 ;ESCREVE NAS SAIDAS DIGITAIS 1 E 2
            JSR         W1
            JSR         W2
            JSR         INICIO
W1:                                 ;ESCRITA DA DADOS DA SAIDA DIGITAL 1
            LDA         #$20        ;LEITURA DE DADOS DO CANAL 1 SENSOR DE
TEMPERATURA
            STA         ADSCR       ;HABILITA CH1 DO MICRO PTA0
            JSR         Dly_125ms   ;SALTA PARA SUBROTINA Dly_125ms

LOOP3:
            LDA         ADSCR       ;CARREGA ACUMULADOR COM REG ADSCR
            AND         #$80
            BEQ         LOOP3       ;DESVIA SE CONVERSÃO A/D IMCOMPLETA
            LDA         ADR         ;CARREGA A COM RESULTADO DA CONVERSÃO
            CMP         $0090       ;Compara A com memória
            BHI         LIGA        ;Desvia se conteudo de A for maior que o
                                            ;conteudo da memória
            BCLR   4,PORTA          ;ZERA PINO 4 DA PORTA A
            JSR         SUB2PULSOS  ;DELAY DE 2 PULSOS DE DCLK
            JSR         SUB2PULSOS  ;DELAY DE 2 PULSOS DE DCLK
            RTS
LIGA:                               ;LIGA VENTILADOR
            BSET   4,PORTA          ;Manda 1 pra PTA4
            JSR         SUB2PULSOS  ;DELAY DE 2 PULSOS DE DCLK
            JSR         SUB2PULSOS  ;DELAY DE 2 PULSOS DE DCLK
            RTS

W2:                                 ;ESCRITA DA DADOS DA SAIDA DIGITAL 2
            LDA         #$21        ;LEITURA DE DADOS DO CANAL 1 SENSOR DE
TEMPERATURA
            STA         ADSCR       ;HABILITA CH1 DO MICRO PTA0
```

```
            JSR         Dly_125ms     ;SALTA PARA SUBROTINA Dly_125ms

LOOP3X:
            LDA         ADSCR         ;CARREGA ACUMULADOR COM REG ADSCR
            AND         #$80
            BEQ         LOOP3X        ;DESVIA SE CONVERSÃO A/D IMCOMPLETA
            LDA         ADR           ;CARREGA A COM RESULTADO DA CONVERSÃO
            CMP         $0090         ;Compara A com memória
            BHI         LIGA2         ;Desvia se conteudo de A for maior que o
                                      ;conteudo da memória
                                      ;DESLIGA VENTILADOR
            BCLR  5,PORTA             ;ZERA PINO 5 DA PORTA A
            JSR         SUB2PULSOS    ;DELAY DE 4 PULSOS DE DCLK
            JSR         SUB2PULSOS    ;DELAY DE 4 PULSOS DE DCLK
            RTS
LIGA2:                                ;LIGA VENTILADOR
            BSET  5,PORTA             ;Manda 1 pra PTA4
            JSR         SUB2PULSOS    ;DELAY DE 4 PULSOS DE DCLK
            JSR         SUB2PULSOS    ;DELAY DE 4 PULSOS DE DCLK
            RTS

;LEITURA DE DADOS (CANAIS ANALÓGICOS 1 E 2)
RD:         LDA         $0082         ;VALOR CANAL
            CMP         $0084
            BEQ         CH0           ;SE FOR 0 VAI PARA CHO

            LDA         $0082         ;VALOR CANAL
            CMP         $0085         ;SE FOR 1 VAI PARA CH1
            BEQ         CH1

            LDA         $0082         ;VALOR CANAL
            CMP         $0086         ;SE FOR 2 VAI PARA CH2
            BEQ         CH2
            JMP         INICIO

CH0:                                  ;ENVIA DADOS DOS CANAIS 1 E 2
            JSR         CH1
            JSR         CH2
            JSR         INICIO

CH1:                                  ;ENVIA DADOS DO CANAL 1
            LDA         #$20
            STA         ADSCR         ;HABILITA CH1 DO MICRO PTA0
            JSR         Dly_125ms     ;SALTA PARA SUBROTINA Dly_125ms

            JSR         SUB2PULSOS
LOOP4:
            LDA         ADSCR         ;CARREGA ACUMULADOR COM REG ADSCR
            AND         #$80
            BEQ         LOOP4         ;DESVIA SE CONVERSÃO A/D IMCOMPLETA

            LDA         ADR           ;CARREGA A COM RESULTADO DA CONVERSÃO
            STA         $0087         ;GRAVA RESULTADO DA CONVERSÃO NA MEMÓRIA
            JSR   SUB1PULSOS
            JSR         ENV           ;ENVIA O RESULTADO PARA NCAP
            JSR   SUB1PULSOS
```

```
;INVERÇÃO DO ESTADO DA SAÍDA NACK
            BRCLR 5,PORTB,NACK1
            BCLR  5,PORTB
            RTS
NACK1:      BSET  5,PORTB
            RTS

CH2:                                    ;ENVIA DADOS DO CANAL 1
            LDA         #$21
            STA         ADSCR           ;HABILITA CH2 DO MICRO PTA1
            JSR         Dly_125ms       ;SALTA PARA SUBROTINA Dly_125ms

LOOP5:
            LDA         ADSCR           ;CARREGA ACUMULADOR COM REG ADSCR
            AND         #$80
            BEQ         LOOP5           ;DESVIA SE CONVERSÃO A/D IMCOMPLETA

            LDA         ADR             ;CARREGA A COM RESULTADO DA CONVERSÃO
            STA         $0087           ;GRAVA RESULTADO DA CONVERSÃO NA MEMÓRIA
            JSR     SUB1PULSOS
            JSR         ENV             ;ENVIA O RESULTADO PARA NCAP
            JSR     SUB1PULSOS
;INVERÇÃO DO ESTADO DA SAÍDA NACK
            BRCLR 5,PORTB,NACK11
            BCLR  5,PORTB
            RTS
NACK11:     BSET  5,PORTB
            RTS

;FUNÇÃO LEITURA DOS TEDS MUDAR O COMANDO CBEQ POR CMP
RT:         LDA         $0082           ;ENDEREÇO DO CANAL
            CBEQ  $0084,RTCH0           ;SE FOR 0 VAI PARA CHO
            LDA         $0082           ;ENDEREÇO DO CANAL
            CBEQ  $0085,RTCH1           ;SE FOR 1 VAI PARA CH1
            LDA         $0082           ;ENDEREÇO DO CANAL
            CBEQ  $0086,RTCH2           ;SE FOR 2 VAI PARA CH2
            LDA         $0082           ;ENDEREÇO DO CANAL
            CBEQ  $0091,RTCH3           ;SE FOR 3 VAI PARA CH3
            LDA         $0082           ;ENDEREÇO DO CANAL
            CBEQ  $0092,RTCH4           ;SE FOR 4 VAI PARA CH4
            JSR         INICIO

RTCH0:                                  ;SUPONDO QUE TEDS DO CANAL 0 ESTÃO
        LOCALIZADOS NAS
            LDHX  #$00A0        ;F600;MEMÓRIAS DE 00A0 A 00AA (10 BYTES)
CONT:  MOV  X+,$0087        ;ENDEREÇO INICIAL DE DADOS DO CANAL 0(10 BYTES)
            JSR     SUB1PULSOS
            JSR         ENV
;INVERÇÃO DO ESTADO DA SAÍDA NACK
            BRCLR 5,PORTB,NACK2
            BCLR  5,PORTB
            JMP         CONT5
NACK2:      BSET  5,PORTB
CONT5:      JSR     SUB1PULSOS
            CBEQX       #$AA,DESV       ;A;ENDEREÇO FINAL DA LEITURA DE TEDS
            JSR     CONT
```

```
RTCH1:                                          ;SUPONDO QUE TEDS DO CANAL 1 ESTÃO
LOCALIZADOS NAS
                LDHX   #$F610          ;MEMÓRIAS DE F610 A F617 (8 BYTES)
CONT1:          MOV    X+,$0087        ;ENDEREÇO INICIAL DE DADOS DO CANAL 1(8 BYTES)
                JSR    SUB1PULSOS
                JSR            ENV
;INVERÇÃO DO ESTADO DA SAÍDA NACK
                BRCLR 5,PORTB,NACK3
                BCLR   5,PORTB
                JMP            CONT6
NACK3:          BSET   5,PORTB
CONT6:          JSR    SUB1PULSOS
                CBEQX          #$18,DESV      ;ENDEREÇO FINAL DA LEITURA DE TEDS DO CANAL
1
                JSR    CONT1

RTCH2:          ;SUPONDO QUE TEDS DO CANAL 2 ESTÃO LOCALIZADOS NAS
                LDHX   #$F620          ;MEMÓRIAS DE F620 A F627 (8 BYTES)
CONT2:          MOV    X+,$0087        ;ENDEREÇO INICIAL DE DADOS DO CANAL 2(8 BYTES)
                JSR    SUB2PULSOS
                JSR            ENV
;INVERÇÃO DO ESTADO DA SAÍDA NACK
                BRCLR 5,PORTB,NACK4
                BCLR   5,PORTB
                JMP            CONT7
NACK4:          BSET   5,PORTB
CONT7:          JSR    SUB1PULSOS
                CBEQX          #$28,DESV;ENDEREÇO FINAL DA LEITURA DE TEDS DO CANAL 2
                JSR    CONT2
DESV:  JSR      INICIO

RTCH3:                         ;SUPONDO QUE TEDS DO CANAL 3 ESTÃO LOCALIZADOS NAS
                LDHX   #$F630          ;MEMÓRIAS DE F630 A F637 (8 BYTES)
CONT3:          MOV    X+,$0087        ;ENDEREÇO INICIAL DE DADOS DO CANAL 3(8 BYTES)
                JSR    SUB2PULSOS
                JSR            ENV
;INVERÇÃO DO ESTADO DA SAÍDA NACK
                BRCLR 5,PORTB,NACK5
                BCLR   5,PORTB
                JMP            CONT8
NACK5:          BSET   5,PORTB
CONT8:          JSR    SUB1PULSOS
                CBEQX          #$38,DESV ;ENDEREÇO FINAL DA LEITURA DE TEDS DO CANAL 3
                JSR    CONT3

RTCH4:                         ;SUPONDO QUE TEDS DO CANAL 4 ESTÃO LOCALIZADOS NAS
                LDHX   #$F640          ;MEMÓRIAS DE F640 A F647 (8 BYTES)
CONT4:          MOV    X+,$0087        ;ENDEREÇO INICIAL DE DADOS DO CANAL 4(8 BYTES)
                JSR    SUB1PULSOS
                JSR            ENV
;INVERÇÃO DO ESTADO DA SAÍDA NACK
                BRCLR 5,PORTB,NACK6
                BCLR   5,PORTB
                JMP            CONT9
NACK6:          BSET   5,PORTB
CONT9:          JSR    SUB1PULSOS
```

```
                CBEQX       #$48,DESV;ENDEREÇO FINAL DA LEITURA DE TEDS DO CANAL 4
                JSR     CONT4

;SUBROTINA DE ENVIO DE DADOS
ENV:

LOOP6:          BRCLR 1,PORTB,LOOP6             ;SE DCLK=0 VAI PARA LOOP6
                LDA         $0087   ;ENDEREÇO DO DADO A SER ENVIADO
                JSR         DESR7
                AND         #$01
                CBEQ  $0085,DOUTSET11 ;DOUTCLR
                BCLR  4,PORTB
                JMP         OUT1
DOUTSET11:BSET  4,PORTB
OUT1:
LOOP7:BRSET 1,PORTB,LOOP7                       ;SE DCLK=1 VAI PARA LOOP7

LOOP8:          BRCLR 1,PORTB,LOOP8             ;SE DCLK=0 VAI PARA LOOP8
                LDA         $0087
                JSR         DESR6
                AND         #$01
                CBEQ  $0085,DOUTSET12
                BCLR  4,PORTB
                JMP         OUT2
DOUTSET12:BSET  4,PORTB
OUT2:

LOOP9:BRSET 1,PORTB,LOOP9                       ;SE DCLK=1 VAI PARA LOOP9

LOOP10:         BRCLR 1,PORTB,LOOP10            ;SE DCLK=0 VAI PARA LOOP10
                LDA         $0087
                JSR         DESR5
                AND         #$01
                CBEQ  $0085,DOUTSET13
                BCLR  4,PORTB
                JMP         OUT3
DOUTSET13:BSET  4,PORTB
OUT3:
LOOP11:BRSET    1,PORTB,LOOP11                  ;SE DCLK=1 VAI PARA LOOP11

LOOP12:         BRCLR 1,PORTB,LOOP12            ;SE DCLK=0 VAI PARA LOOP12
                LDA         $0087
                JSR         DESR4
                AND         #$01
                CBEQ  $0085,DOUTSET14
                BCLR  4,PORTB
                JMP         OUT4
DOUTSET14:BSET  4,PORTB
OUT4:
LOOP13:BRSET    1,PORTB,LOOP13                  ;SE DCLK=1 VAI PARA LOOP13

LOOP14:         BRCLR 1,PORTB,LOOP14            ;SE DCLK=0 VAI PARA LOOP14
                LDA         $0087
                JSR         DESR3
                AND         #$01
                CBEQ  $0085,DOUTSET15
```

```
                BCLR   4,PORTB
                JMP            OUT5
DOUTSET15:BSET  4,PORTB
OUT5:
LOOP15:BRSET    1,PORTB,LOOP15          ;SE DCLK=1 VAI PARA LOOP15

LOOP16:         BRCLR 1,PORTB,LOOP16    ;SE DCLK=0 VAI PARA LOOP16
                LDA            $0087
                JSR            DESR2
                AND            #$01
                CBEQ   $0085,DOUTSET16
                BCLR   4,PORTB
                JMP            OUT6
DOUTSET16:BSET  4,PORTB
OUT6:
LOOP17:         BRSET 1,PORTB,LOOP17    ;SE DCLK=1 VAI PARA LOOP17

LOOP18:         BRCLR 1,PORTB,LOOP18    ;SE DCLK=0 VAI PARA LOOP18
                LDA            $0087
                LSRA
                AND            #$01
                CBEQ   $0085,DOUTSET17
                BCLR   4,PORTB
                JMP            OUT7
DOUTSET17:BSET  4,PORTB
OUT7:
LOOP19: BRSET   1,PORTB,LOOP19          ;SE DCLK=1 VAI PARA LOOP19

LOOP20: BRCLR   1,PORTB,LOOP20          ;SE DCLK=0 VAI PARA LOOP20
                LDA            $0087
                AND            #$01
                CBEQ   $0085,DOUTSET18
                BCLR   4,PORTB
                JMP            OUT8
DOUTSET18:BSET  4,PORTB
OUT8:
LOOP21: BRSET   1,PORTB,LOOP21          ;SE DCLK=1 VAI PARA LOOP21
                RTS

;LEITURA DO PINO DIN
LEITURADIN:
LOOP22:         BRCLR 1,PORTB,LOOP22    ;SE DCLK=0 VAI PARA LOOP22
                LDA    PORTB
                AND            #$01    ;LÊ DIN
                JSR            DESL7
                ORA            $0080
                STA            $0080
LOOP23:         BRSET 1,PORTB,LOOP23    ;SE DCLK=1 VAI PARA LOOP23

LOOP24:         BRCLR 1,PORTB,LOOP24    ;SE DCLK=0 VAI PARA LOOP24
                LDA    PORTB
                AND            #$01    ;LÊ DIN
                JSR            DESL6
                ORA            $0080
                STA            $0080
LOOP25:         BRSET 1,PORTB,LOOP25    ;SE DCLK=1 VAI PARA LOOP25
```

```
LOOP26:     BRCLR 1,PORTB,LOOP26        ;SE DCLK=0 VAI PARA LOOP26
            LDA   PORTB
            AND         #$01    ;LÊ DIN
            JSR         DESL5
            ORA         $0080
            STA         $0080
LOOP27:     BRSET 1,PORTB,LOOP27        ;SE DCLK=1 VAI PARA LOOP27

LOOP28:     BRCLR 1,PORTB,LOOP28        ;SE DCLK=0 VAI PARA LOOP28
            LDA   PORTB
            AND         #$01    ;LÊ DIN
            JSR         DESL4
            ORA         $0080
            STA         $0080
LOOP29:     BRSET 1,PORTB,LOOP29        ;SE DCLK=1 VAI PARA LOOP29


LOOP30:     BRCLR 1,PORTB,LOOP30        ;SE DCLK=0 VAI PARA LOOP30
            LDA   PORTB
            AND         #$01    ;LÊ DIN
            JSR         DESL3
            ORA         $0080
            STA         $0080
LOOP31:     BRSET 1,PORTB,LOOP31        ;SE DCLK=1 VAI PARA LOOP31

LOOP32:     BRCLR 1,PORTB,LOOP32        ;SE DCLK=0 VAI PARA LOOP32
            LDA   PORTB
            AND         #$01    ;LÊ DIN
            JSR         DESL2
            ORA         $0080
            STA         $0080
LOOP33:     BRSET 1,PORTB,LOOP33        ;SE DCLK=1 VAI PARA LOOP33

LOOP34:     BRCLR 1,PORTB,LOOP34        ;SE DCLK=0 VAI PARA LOOP34
            LDA   PORTB
            AND         #$01    ;LÊ DIN
            LSLA
            ORA         $0080
            STA         $0080
LOOP35:     BRSET 1,PORTB,LOOP35        ;SE DCLK=1 VAI PARA LOOP35

LOOP36:     BRCLR 1,PORTB,LOOP36        ;SE DCLK=0 VAI PARA LOOP36
            LDA   PORTB
            AND         #$01    ;LÊ DIN
            ORA         $0080
            STA         $0080
LOOP37: BRSET       1,PORTB,LOOP37      ;SE DCLK=1 VAI PARA LOOP37
            RTS

SUB2PULSOS:
LOOP193:    BRCLR 1,PORTB,LOOP193
LOOP194:    BRSET 1,PORTB,LOOP194
LOOP195:    BRCLR 1,PORTB,LOOP195
LOOP196:    BRSET 1,PORTB,LOOP196
                  RTS
```

```
SUB1PULSOS:
LOOP1931:   BRCLR 1,PORTB,LOOP1931
LOOP1941:   BRSET 1,PORTB,LOOP1941
            RTS

Dly_125ms:  LDA       #5        ;Carrega A com 5 decimal
Loop3A:     LDHX      #t25ms ;#t25ms = 15624, inicio programa
Loop4A:     AIX       #-1       ;Adiciona valor imediato ao reg de indice HX
            CPHX      #0        ;Compara reg de indice com memória???? ou zero
            BNE       Loop4A;Desvia para Loop4 se diferente
            DECA                ;Decrementa A
            BNE       Loop3A;Desvia para Loop3 se diferente
            RTS                 ;Retorno de Subrotina

DESL7:      LSLA                ;7 DESVIOS PARA ESQUERDA
            LSLA
            LSLA
            LSLA
            LSLA
            LSLA
            LSLA
            RTS

DESL6:      LSLA                ;6 DESVIOS PARA ESQUERDA
            LSLA
            LSLA
            LSLA
            LSLA
            LSLA
            RTS

DESL5:      LSLA                ;5 DESVIOS PARA ESQUERDA
            LSLA
            LSLA
            LSLA
            LSLA
            RTS

DESL4:      LSLA                ;4 DESVIOS PARA ESQUERDA
            LSLA
            LSLA
            LSLA
            RTS

DESL3:      LSLA                ;3 DESVIOS PARA ESQUERDA
            LSLA
            LSLA
            RTS

DESL2:      LSLA                ;2 DESVIOS PARA ESQUERDA
            LSLA
            RTS
DESR7:      LSRA                ;7 DESVIOS PARA DIREITA
            LSRA
            LSRA
```

```
                LSRA
                LSRA
                LSRA
                LSRA
                RTS

DESR6:          LSRA        ;6 DESVIOS PARA DIREITA
                LSRA
                LSRA
                LSRA
                LSRA
                LSRA
                RTS

DESR5:          LSRA        ;5 DESVIOS PARA DIREITA
                LSRA
                LSRA
                LSRA
                LSRA
                RTS

DESR4:          LSRA        ;4 DESVIOS PARA DIREITA
                LSRA
                LSRA
                LSRA
                RTS

DESR3:          LSRA        ;3 DESVIOS PARA DIREITA
                LSRA
                LSRA
                RTS

DESR2:          LSRA        ;2 DESVIOS PARA DIREITA
                LSRA
                RTS

*****************************************************************
* t_isr - Timer Interrupt Service Routine.                    *
*          after a RESET.                                     *
*****************************************************************
t_isr:
    pshh
    lda   TSC1
    and   #$7f
    sta   TSC1          ; Clear O.C. Flag
    ldhx  TCH1h
    aix   #$4d          ; Setup another interrupt in ~2ms
    sthx  TCH1h
    pulh

done_tisr:
    rti
END
```