

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ  
DEPARTAMENTO ACADÊMICO DE ELETRÔNICA  
ENGENHARIA INDUSTRIAL ELÉTRICA - ÊNFASE EM  
ELETRÔNICA/TELECOMUNICAÇÕES**

**ERIKSON FERRARINI  
MATEUS LUIS TONDO  
THIAGO SOUZA LACERDA**

**PLATAFORMA DIGITAL EMULADA EM TEMPO REAL**

**TRABALHO DE CONCLUSÃO DE CURSO**

**CURITIBA**

**2011**

**ERIKSON FERRARINI  
MATEUS LUIS TONDO  
THIAGO SOUZA LACERDA**

## **PLATAFORMA DIGITAL EMULADA EM TEMPO REAL**

Trabalho de Conclusão de Curso apresentado ao Departamento Acadêmico de Eletrônica, como requisito parcial para a obtenção do grau de Engenheiro no curso de Engenharia Industrial Elétrica – Ênfase em Eletrônica/Telecomunicações, da Universidade Tecnológica Federal do Paraná

Orientador: Prof. Rubens Alexandre de Faria

**CURITIBA**

**2011**

## **AGRADECIMENTO**

Os autores agradecem, primeiramente, aos pais e companheiras, que foram compreensivos com a abstenção de momentos da vida particular em prol da realização deste objetivo comum. Também pela orientação, confiança e apoio que ofereceram durante a longa jornada percorrida.

Agradecemos especialmente aos alunos Mikhail Koslowski, Carlos Alexandre Micaloski, Diego Dias dos Reis, Alexandre Noguchi, João Paulo Sant'anna Junior, Carlos Toledo, Philipe Ambrózio Dias e também aos professores Rubens Alexandre de Faria, Dario Dergint e Wolney Betiol que, de alguma maneira, contribuíram para o resultado deste projeto.

## RESUMO

FERRARINI, Erikson; TONDO, Mateus L.; LACERDA, Thiago S.. **Plataforma digital emulada em tempo real**. 2011. 117 páginas. Trabalho de Conclusão de Curso – curso de Engenharia Industrial Elétrica - ênfase Eletrônica/Telecomunicações, Universidade Tecnológica Federal do Paraná, Curitiba, 2011.

Em alinhamento com as atuais metodologias de ensino, que objetivam propiciar o contato direto dos alunos com as tecnologias demandadas pelo mercado e desejáveis nos profissionais, é de fundamental importância dispor de plataformas didáticas para a familiarização com os recursos envolvidos. O objetivo principal é desenvolver uma plataforma didática microcontrolada com arquitetura de 8 *bits* para o auxílio na disciplina de microcontroladores da UTFPR. A metodologia envolve um estudo a ser realizado acerca do mercado nacional analisando a participação das marcas, presença de distribuidores, variedade e disponibilidade de modelos, ferramentas de desenvolvimento compatíveis e recursos disponíveis no *chip*. A plataforma didática desenvolvida contempla, além da placa que contém o microcontrolador de 8 *bits*, interfaces do tipo USB (*Universal Serial Bus*) para transferência de código-fonte para a placa e JTAG (*Joint Test Action Group*) para depuração em tempo real. Os principais resultados esperados ao final do projeto são a elevação na qualidade das disciplinas, nas quais se matriculam em torno de 130 alunos a cada semestre, através do uso de recursos mais atuais, e a transmissão de conhecimentos práticos mais alinhados com a realidade do mercado de eletrônica através do uso do *kit* didático.

**Palavras Chave:** *Kit* Didático. Microcontrolador 8 *bits*. JTAG. *Debug* Tempo Real. Interface USB. Plataforma Desenvolvimento.

## ABSTRACT

FERRARINI, Erikson; TONDO, Mateus L.; LACERDA, Thiago S.. **Plataforma digital emulada em tempo real**. 2011. 117 pages. End of Course Assignment – Electronic Engineering, Technological Federal University of Parana, Curitiba, 2011;

Aligned with the current teaching methodologies, which intend to promote the direct contact of students with technologies needed by the market and desirable in professionals, it is extremely important to count on didactic kits for the students to become familiar with the resources involved. The main objective is to develop a microcontrolled 8 *bits* architecture teaching platform to assist on tuition of microcontrollers' subjects at UTFPR. The methodology employed will be based on a market analysis that is going to evaluate the market share among companies, national distributors, variety and availability of models, compatible development environments and resources embedded on chip. The developed resource will contain not only the microcontrolled board, but additionally an USB interface for the source code to be transferred to the target board and a JTAG interface for real time debugging. The expected results at the end of the project are the improvement of quality on subjects (in which around 130 are enrolled every semester) by using newer resources and also promote the transmission of practical electronics knowledge by using the evaluation kit.

**Key words:** Didactic Kit, 8 bit microcontroller, JTAG, Real Time Debug, Interface USB, Evaluation Platform.

## SUMÁRIO

<b>1. INTRODUÇÃO .....</b>	<b>14</b>
1.1. JUSTIFICATIVA.....	14
1.2. OBJETIVOS .....	15
1.2.1. Objetivo Geral .....	15
1.2.2. Objetivos Específicos .....	15
1.2.3. Diagrama.....	16
1.3. METODOLOGIA.....	16
1.4. APRESENTAÇÃO DO DOCUMENTO .....	18
<b>2. FUNDAMENTAÇÃO TEÓRICA .....</b>	<b>19</b>
2.1. MICROCONTROLADORES.....	19
2.2. JTAG .....	20
2.3. USB ( <i>UNIVERSAL SERIAL BUS</i> ) .....	22
<b>3. DESENVOLVIMENTO.....</b>	<b>24</b>
3.1. MICROCONTROLADORES.....	24
3.1.1. C8051F386 .....	24
3.1.1.1 CIP-51.....	26
3.1.1.2 Organização da memória.....	28
3.1.1.3 Interrupções .....	30
3.1.1.4 Oscilador e <i>Clock</i> .....	35
3.1.1.5 Entradas e saídas .....	36
3.1.1.6 <i>Timer</i> .....	38
3.1.2. C8051F321 .....	38
3.2. PROTOCOLO SILICON LABS C2 .....	39
3.2.1. Modelo de instruções .....	40
3.2.2. Ciclo de escrita de endereço (11b).....	42
3.2.3. Ciclo de leitura de endereço (10b) .....	42
3.2.4. Ciclo de escrita de dados (01b).....	43
3.2.5. Ciclo de Leitura de Dados .....	44
3.2.6. Considerações sobre os tempos .....	45
3.2.7. Gravação da memória FLASH através do protocolo C2 .....	46
3.3.7.1 Interface C2 (C2I) .....	47
3.3.7.2 Registrador de identificação do dispositivo (DEVICEID).....	47
3.3.7.3 Registrador identificador de versão (REVID) .....	48

3.3.7.4	Registradores para a programação da FLASH .....	49
3.3.7.5	Interface de programação da FLASH .....	49
3.3.7.6	Escrita de um bloco da memória FLASH .....	49
3.3.7.7	Leitura de um bloco da memória FLASH .....	51
3.3.7.8	Limpeza de uma página da memória FLASH .....	52
3.3.7.9	Limpeza da memória FLASH completa .....	53
3.3.	PLACAS E CIRCUITOS .....	54
<b>4.</b>	<b>TESTES E RESULTADOS .....</b>	<b>58</b>
4.1.	ALIMENTAÇÃO .....	58
4.2.	OSCILADOR, PINOS DE ENTRADA E SAÍDA E LEDS .....	59
4.3.	DEBUG ADAPTER .....	61
4.4.	TESTE DO SOFTWARE C2 .....	62
4.5.	SUGESTÕES PARA VERSÕES FUTURAS .....	63
4.6.	CONSIDERAÇÕES .....	63
<b>5.</b>	<b>CONCLUSÃO TÉCNICA .....</b>	<b>64</b>
<b>6.</b>	<b>PLANO DE NEGÓCIOS .....</b>	<b>66</b>
6.1.	SUMÁRIO EXECUTIVO .....	66
6.1.1.	Oportunidade de negócio .....	66
6.1.2.	O produto .....	66
6.1.3.	A Empresa .....	67
6.1.4.	Visão 2014 .....	67
6.1.5.	Missão .....	67
6.1.6.	Valores .....	68
6.1.7.	Descrição do negócio .....	68
6.2.	PRODUTOS E SERVIÇOS .....	69
6.2.1.	Características .....	69
6.2.2.	Produtos Similares .....	69
6.2.3.	Análise Comparativa .....	71
6.2.4.	Análise SWOT .....	71
6.2.5.	Produtos e serviços futuros .....	72
6.3.	ANÁLISE DE MERCADO .....	72
6.3.1.	Importação de microcontroladores no Brasil .....	72
6.3.2.	Segmentação de Mercado .....	76
6.3.3.	Segmento Alvo de Mercado .....	79

6.3.4.	Necessidades do Mercado.....	79
6.3.5.	Crescimento do Mercado .....	81
6.3.6.	<i>Players</i> .....	83
6.3.7.	Modelo de Distribuição.....	84
6.3.8.	Modelo de Competitividade.....	85
6.3.9.	Proposta de valor .....	85
6.4.	PLANO DE MARKETING.....	86
6.4.1.	Política de preços.....	86
6.4.2.	Estratégia de Vendas .....	86
6.5.	PLANO FINANCEIRO .....	87
6.5.1.	Investimento Inicial.....	87
6.5.2.	Receitas .....	87
6.5.3.	Custos e Despesas .....	88
6.5.4.	Ponto de Equilíbrio.....	88
6.5.5.	Demonstrativo de Resultados .....	89
6.6.	CONSIDERAÇÕES SOBRE O PLANO DE NEGÓCIOS .....	89
<b>7.</b>	<b>GESTÃO DE PROJETO .....</b>	<b>91</b>
7.1.	GERÊNCIA DO TEMPO DE PROJETO.....	91
7.1.1.	Cronograma Planejado e Custos Homem/Hora .....	91
7.1.2.	Disparidade entre o Cronograma Previsto e Executado .....	93
7.2.	GESTÃO DE CUSTOS .....	93
7.3.	GESTÃO DE RISCOS.....	98
7.4.	CONSIDERAÇÕES SOBRE GESTÃO DO PROJETO .....	99
<b>8.</b>	<b>CONCLUSÃO MERCADOLÓGICA.....</b>	<b>100</b>
	<b>BIBLIOGRAFIA.....</b>	<b>101</b>
	<b>ANEXOS .....</b>	<b>103</b>
	<b>PDETR- GUIA DO USUÁRIO.....</b>	<b>109</b>



## LISTA DE FIGURAS

Figura 1 - Diagrama .....	16
Figura 2 - Mapa de memória on-chip do C8051F38X .....	20
Figura 3 - Pinos do protocolo JTAG .....	21
Figura 4 - Características físicas do USB.....	22
Figura 5 - Família C805138x "on chip" .....	25
Figura 6 - Diagrama de Blocos do C8051F386 .....	26
Figura 7 - Diagrama de Blocos do CIP-51.....	27
Figura 8 - Mapa de memória on-chip do C8051F38.....	28
Figura 9 - Opções de osciladores.....	35
Figura 10 - Diagrama de blocos dos pinos de entrada e saída .....	37
Figura 11 - Esquemático de um pino.....	37
Figura 12 - Diagrama de blocos da interface C2 (C2I) .....	40
Figura 13 - Modelo de instruções do protocolo C2.....	40
Figura 14 - Ciclo de escrita de endereço.....	42
Figura 15 - Ciclo de leitura de endereço .....	43
Figura 16 - Ciclo de escrita de dados.....	43
Figura 17 - Ciclo de leitura de dados.....	44
Figura 18 - Diagrama de reinicialização do dispositivo alvo .....	45
Figura 19 - Arquitetura da interface de programação da FLASH via C2 .....	46
Figura 20 - Inicialização da FPI .....	49
Figura 21 - Escrita de um bloco da memória FLASH .....	50
Figura 22 - Leitura de um bloco da memória FLASH .....	51
Figura 23 - Limpeza de uma página da memória FLASH .....	53
Figura 24 - Limpeza da memória FLASH .....	54
Figura 25 - Layout da face superior da placa PDETR .....	56
Figura 26 - Layout da face inferior da placa PDETR.....	56
Figura 27 - Layout da face superior da placa PDETR fabricada .....	57
Figura 28 - Layout da face inferior da placa PDETR fabricada .....	57
Figura 29 - Validação do circuito de alimentação.....	58

Figura 30 - Tensão de saída da placa.....	59
Figura 31 - Testes de clock e pinos de saída.....	60
Figura 32 - Port1 conectado ao barramento de Leds.....	61
Figura 33 - Debug Adapter conectado ao PC.....	61
Figura 34 - Debug Adapter conectado (PC e ambiente de desenvolvimento).....	62
Figura 35 - C8051F340DK.....	70
Figura 36 - MCB950.....	70

## LISTA DE QUADROS

Quadro 1 - Ordem de prioridade de interrupções.....	31
Quadro 2 – Registrador IE .....	32
Quadro 3 – Registrador IP .....	33
Quadro 4 – Configuração de interrupções .....	34
Quadro 5 – Registrador IT01CF .....	34
Quadro 6 – Registrado CLKSEL .....	36
Quadro 7 – Distribuição dos Timers .....	38
Quadro 8 - Instruções C2 .....	41
Quadro 9 - Tempos limites .....	45
Quadro 10 - Valores de retorno.....	47
Quadro 11 - Valores de estado .....	48
Quadro 12 - Valores configuráveis para escrita na memória FLASH.....	50
Quadro 13 - Páginas da memória FLASH.....	52
Quadro 14 - Análise comparativa.....	71
Quadro 15 - Importações por estado brasileiro - 10 maiores .....	74
Quadro 16 - Importações no Brasil – Consolidado.....	75
Quadro 17 - Segmentação de clientes potenciais .....	77
Quadro 18 - Perfil dos Egressos na Educação Superior Brasil: 2000 e 2008 .....	77
Quadro 19 - Concluintes da Educação Superior em Engenharia: 1999 a 2008 .....	78
Quadro 20 - Clientes Potenciais.....	79
Quadro 21 - Faturamento Total por Área .....	82
Quadro 22 - Ranking dos maiores players do mundo .....	84
Quadro 23 - Quadro de responsabilidades e ações .....	92
Quadro 24 - Lista de riscos .....	99

## LISTA DE TABELAS

Tabela 1 - Número de instruções x Ciclos de Clock.....	27
Tabela 2 - Investimento Inicial.....	87
Tabela 3 - Unidades produzidas X Custo de Produção.....	88
Tabela 4 - Custos Fixos .....	88
Tabela 5 - Ponto de Equilíbrio.....	89
Tabela 6 - Demonstrativo de Resultados .....	89
Tabela 7 - Lista de componentes para a placa alvo .....	94
Tabela 8 - Lista de componentes para Debbuger .....	94
Tabela 9 - Lista de componentes para pesquisa.....	95
Tabela 10 - Custos com impostos .....	95
Tabela 11 - Custo de Produção para 50 PDETR .....	96
Tabela 12 - Custo para produção de 50 Debug Adapters .....	97

## LISTA DE GRÁFICOS

Gráfico 1 – Importação de Microcontroladores e Microprocessadores - Totais em USD.....	75
Gráfico 2 - Importações no Brasil consolidado .....	76
Gráfico 3 - Investimentos no setor eletroeletrônico .....	80
Gráfico 4 - Treinamento de funcionários .....	81
Gráfico 5 - Faturamento Setor Eletroeletrônico .....	82
Gráfico 6 - Número de empregos formais .....	83



## 1. INTRODUÇÃO

A crescente busca por profissionais mais qualificados prova a necessidade da constante melhoria e atualização do sistema de educação, como um todo, para que essa demanda seja suprida. Neste processo de melhoria contínua, deve haver investimentos na qualificação do corpo docente, passando pela estrutura física e também pelos recursos de ensino, tudo para que sejam garantidas as condições que potencializem o aprendizado do aluno no ambiente de ensino.

### 1.1. JUSTIFICATIVA

A etapa que consiste na formação acadêmica se torna cada vez mais importante diante do refinamento das habilidades e capacidades exigidas dos profissionais que desejam se manter no mercado de trabalho. Desta forma, faz-se necessário estar alinhado com as tendências da tecnologia constantemente e, por esse motivo, é fundamental a melhoria dos recursos utilizados para treinamentos e capacitação.

Sob esta ótica, foi percebida a demanda para atualização da plataforma didática de microcontroladores utilizada na UTFPR. A evidência mais notável surgiu da observação dos computadores pessoais disponíveis no mercado. O aspecto que chamou, de fato, a atenção foi o de que o kit didático atual, utilizado em disciplinas de microcontroladores, tem como padrão de comunicação de dados para depuração e carregamento de programas a interface RS232, estabelecida em 1969 (EIA STANDARD RS-232-C)<sup>9</sup>. Embora ainda utilizada, tal interface vem sendo substituída há algum tempo pelo protocolo de comunicação USB (*Universal Serial Bus*), em desenvolvimento desde 1994 (UNIVERSAL SERIAL BUS SPECIFICATION)<sup>10</sup>. Além disso, a interface RS232 é raramente encontrada nos computadores pessoais à venda e revela-se uma limitadora para o uso desta plataforma didática em um futuro próximo. Entre as vantagens do protocolo USB pode-se destacar, ainda, a maior velocidade de comunicação de dados, maior compatibilidade com periféricos e *softwares*.

Outros pontos de argumentação para ainda não haver um *kit* com conectividade USB não implementada eram os custos superiores de um microcontrolador com esta funcionalidade padronizada no *chip*, não mais um obstáculo graças à popularização desta interface, e a falta de recursos de depuração acessíveis (JTAG - *Joint Test ActionGroup*), comercializados no mercado por valores na faixa de alguns milhares de dólares, que são recursos contemplados pelo escopo do presente projeto.

Sendo assim, o *kit* renovado vai ao encontro das necessidades atuais tanto dos professores quanto dos 130 novos alunos que cursam as disciplinas de microcontroladores a cada semestre, no sentido de buscar uma alternativa de baixo custo para elevar a qualidade do ensino e potencializar a transmissão de novas tecnologias aos discentes.

Por estas razões, o fator inovador da plataforma didática que irá substituir a existente justifica a realização deste projeto.

## 1.2. OBJETIVOS

### 1.2.1. Objetivo Geral

Desenvolver uma plataforma didática microcontrolada com arquitetura de 8 *bits* para o auxílio na disciplina de microcontroladores da UTFPR.

### 1.2.2. Objetivos Específicos

- Desenvolver uma placa com microcontrolador de 8 *bits* programável em *assembly* no padrão CISC (*Complex Instruction Set Computer*);
- Desenvolver uma interface de comunicação USB entre o PC e a placa do microcontrolador para transferência de código-fonte;
- Desenvolver *hardware* e *firmware* de uma interface de comunicação JTAG entre o PC e a placa do microcontrolador para depuração em tempo real;



- Desenvolver um manual de instruções da plataforma que contemple as características técnicas da placa microcontrolada, o *layout* do hardware e os procedimentos básicos de inicialização e utilização.

### 1.2.3. Diagrama em blocos

O diagrama da Figura 1 apresenta a visão do geral do projeto.

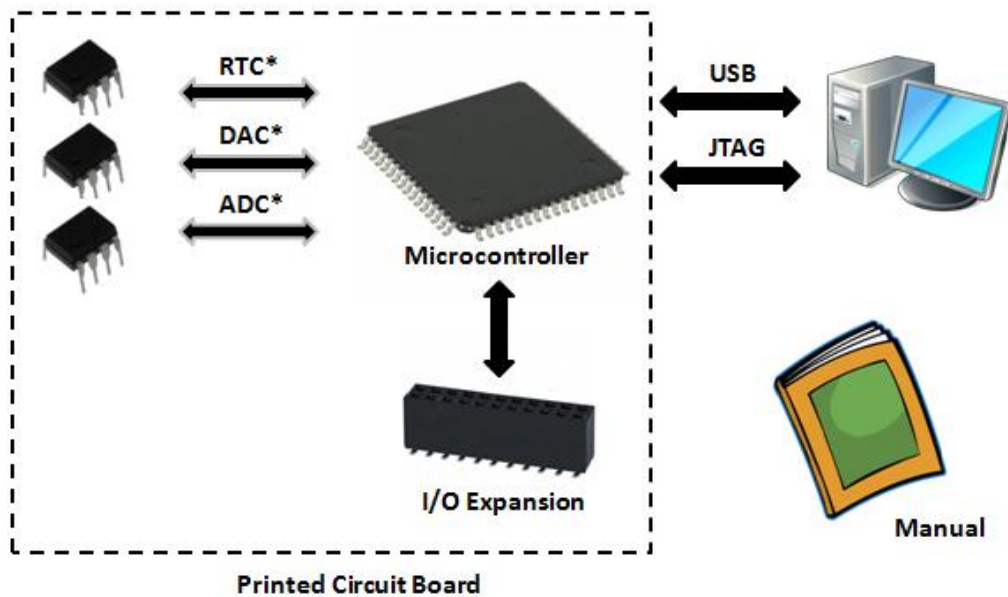


Figura 1– Diagrama  
Fonte: Autoria própria

### 1.3. METODOLOGIA

Para o desenvolvimento do projeto serão necessários alguns passos listados a seguir. Vale reforçar que dividiremos o tópico metodologia em três etapas:

#### Etapa 1

- Estudar o mercado de microcontroladores e a participação dos fabricantes no cenário nacional;
- Estudar e definir os principais canais de distribuição dos fabricantes em questão;

- Pesquisar e definir, dentre os fabricantes mais participativos, o microcontrolador que atenda as especificações técnicas de projeto;
- Estudar a estrutura (*hardware*, linguagem de programação e ambiente de desenvolvimento) do microcontrolador selecionado;

A etapa 1 é considerada um ponto crítica de alto impacto para o projeto porque o microcontrolador a ser utilizado na plataforma didática será selecionado a partir dos resultados obtidos nos estudos realizados.

### Etapa 2

- Desenvolver, implementar e testar circuitos e periféricos ligados ao funcionamento do microcontrolador (fonte, memória, *clock* e barramentos de entrada e saída);
- Pesquisar e estudar as alternativas para implementação da interface JTAG a ser utilizada para depuração em tempo real;
- Desenvolver, implementar e testar a interface JTAG;
- Integrar interfaces JTAG e USB com o microcontrolador;
- Rotear placa de circuito impresso da plataforma de desenvolvimento;

A etapa 2 é considerada um ponto crítico porque o êxito no desenvolvimento e implementação da solução proposta, com base nas tecnologias selecionadas, está diretamente ligada aos objetivos definidos no projeto.

### Etapa 3

- Comprar componentes eletrônicos para montagem da placa de circuito impresso;

- Fabricar a placa de circuito impresso da plataforma;
- Montar e testar a placa de circuito impresso;
- Desenvolver atividades para demonstração da plataforma em funcionamento;
- Redigir o manual de instruções;

#### 1.4. APRESENTAÇÃO DO DOCUMENTO

Pretende-se com esse documento descrever o processo de desenvolvimento do projeto final dos alunos Erikson Ferrarini, Mateus Luis Tondo e Thiago Souza Lacerda do curso de Engenharia Elétrica – Ênfase em Eletrônica/Telecomunicações da Universidade Tecnológica Federal do Paraná.

Primeiramente é discutida a metodologia utilizada para desenvolver o projeto. A seguir, é feita uma descrição das tecnologias utilizadas e das etapas executadas no desenvolvimento do projeto. Posteriormente, em Testes e Resultados, é apresentada a situação atual do projeto e as sugestões para as mudanças em relação ao que foi inicialmente planejado.

Na sequência, é descrito o plano de negócios, contendo informações sobre o produto, mercado, análise de marketing e questões financeiras. Por fim, apresenta-se a Gestão do Projeto e uma conclusão a respeito das etapas de desenvolvimento e do resultado final obtido. Em anexo, encontram-se o manual do produto e esquemáticos.

## 2. FUNDAMENTAÇÃO TEÓRICA

Faz-se necessário, antes de realizar a abordagem dos aspectos referentes ao desenvolvimento do projeto, uma breve fundamentação a respeito das tecnologias, dispositivos e conceitos associados. Busca-se, assim, alcançar uma contextualização adequada e facilitar a compreensão do escopo contemplado. Neste capítulo será feita uma breve descrição sobre microcontroladores 8051, protocolo JTAG e interface USB.

### 2.1. MICROCONTROLADORES

Os microcontroladores são componentes eletrônicos concebidos para realizar a função de controle em sistemas, substituindo componentes eletromecânicos usados outrora em diversos produtos utilizados no cotidiano. Desenvolvidos com base nos conceitos da eletrônica digital, estes dispositivos são marcados por duas características fundamentais: a capacidade de serem programados para operar dados sem a intervenção humana e a capacidade de armazenar e trocar dados com outros elementos de um sistema.

Na década de 80 os primeiros modelos de microcontroladores foram lançados, e exatamente no ano de 1980 a fabricante Intel lançou o 8051(MACKENZIE. I. SCOTT)<sup>13</sup>. Este foi o primeiro modelo produzido no padrão da família MCS-51, uma arquitetura desenvolvida pela Intel, com registradores e barramento de dados de 8 *bits* – e por esse motivo dito um microcontrolador de 8 *bits*. Esta linha era integrada por mais de 60.000 transistores, 4*kbytes* de memória de programa (ROM), 128 *bytes* de memória de dados (RAM), 32 pinos de entrada/saída, porta serial, interrupções, dois temporizadores de 16 *bits* e circuito gerador de *clock* integrado MCS 51 (MICROCONTROLLER FAMILY USER'S MANUAL)<sup>11</sup>, sendo estas características avançadas para um único circuito integrado naquela época. O diagrama em blocos do MCS-51 pode ser visto na Figura 2.

O funcionamento dos microcontroladores se dá pela realização de operações aritméticas (Adição, Subtração, Divisão e Multiplicação), lógicas (*AND*, *OR*, *NOT*, etc), manipulação de dados e operações de salto, todas representadas por códigos binários, chamados de grupo de instruções, operadas pela Unidade

Central de Processamento (CPU), que no caso de uma aplicação se encontram programadas internamente à memória de programa.

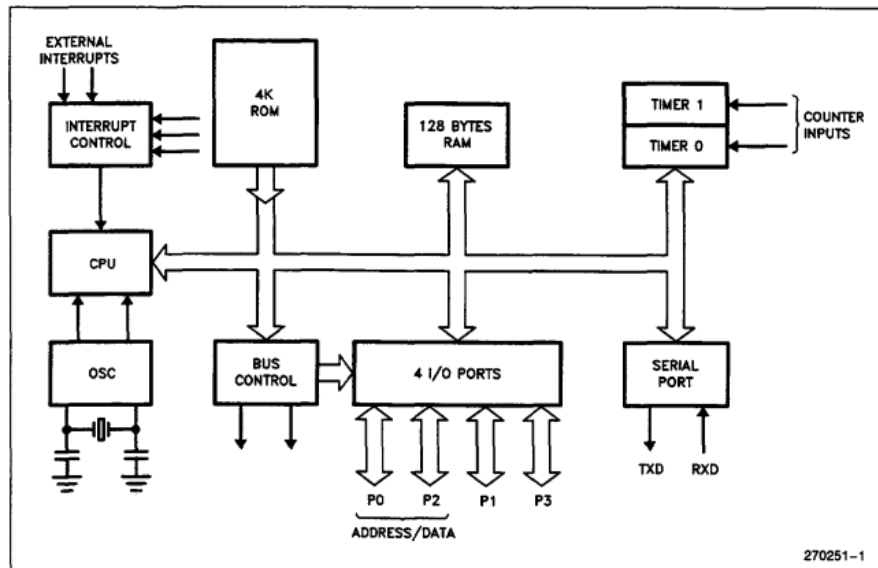


Figura 2- Mapa de memória on-chip do C8051F38X  
Fonte: (INTEL, 1994)

O grupo de instruções que leva o mesmo nome da família de *chips* MCS-51 é otimizado para trabalhar com 8 *bits* e provê uma variedade de formas de acesso à memória para facilitar as operações de *bytes* e manipulações de *bits* individualmente em sistemas de controle e lógica que demandam processamento Booleano. Através deste grupo de instruções é possível realizar a operação do núcleo do controlador e viabilizar a sua interação com os outros blocos do componente, criando uma interface com o ambiente externo à pastilha.

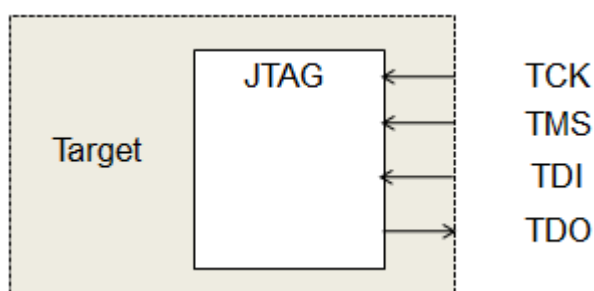
## 2.2. JTAG (*Joint Test Action Group*)

Em 1985, um grupo de empresas europeias de sistemas eletrônicos formou o *Joint European Test Action Group* (JETAG), com o objetivo de padronizar um método de testes de circuito impresso (IEEE STD 1149.1 (JTAG) TESTABILITY PRIMER)<sup>12</sup>. Em 1988, com a entrada da América do Norte, o grupo mudou para JTAG e em 1990 o IEEE padronizou o conceito e criou a norma 1149.1 conhecida como:

- *IEEE Standard Test Access Port and Boundary Scan Architecture*

Inicialmente o protocolo JTAG foi usado para testes de *chips* e placas através do método *Boundary Scan*. Hoje ele é utilizado para depuração em tempo real e gravação de código e dados em memórias FLASH, além de estar presente nos principais microprocessadores utilizados pela indústria eletrônica atual.

Como características físicas ele apresenta uma arquitetura de quatro fios. Um para *clock*, dois para envio e recebimento de dados e informações e um fio de controle (Figura 3). Um pino adicional de *reset* pode ser utilizado se necessário.



**Figura 3 - Pinos do protocolo JTAG**  
Fonte: Autoria própria

O TCK (*Test clock*) é um pino de controle que recebe um sinal de *clock* independente do *clock* do sistema. A frequência de TCK deve ser suportada pelo circuito com o qual se deseja trocar informações. O TDI (*Test Data Input*) recebe serialmente os dados e instruções de entrada sempre na borda de subida do TCK. O TMS (*Test Mode Select*) é um pino de controle utilizado para se orientar através dos estados da máquina de estados do protocolo. O TDO (*Test Data Output*) entrega serialmente os dados de saída sempre na borda de descida do TCK. O TAP (*Test Access Port*) é uma máquina de estados finita que responde por variações nos sinais TCK e TMS e controla os estados do circuito de teste. As transições de estado ocorrem baseadas no valor de TMS durante a borda de subida do TCK, ou quando ocorrer um *reset* (O TRST é um pino de reset opcional do protocolo). O TAP deve gerar os sinais para controlar a operação dos circuitos envolvidos no teste.

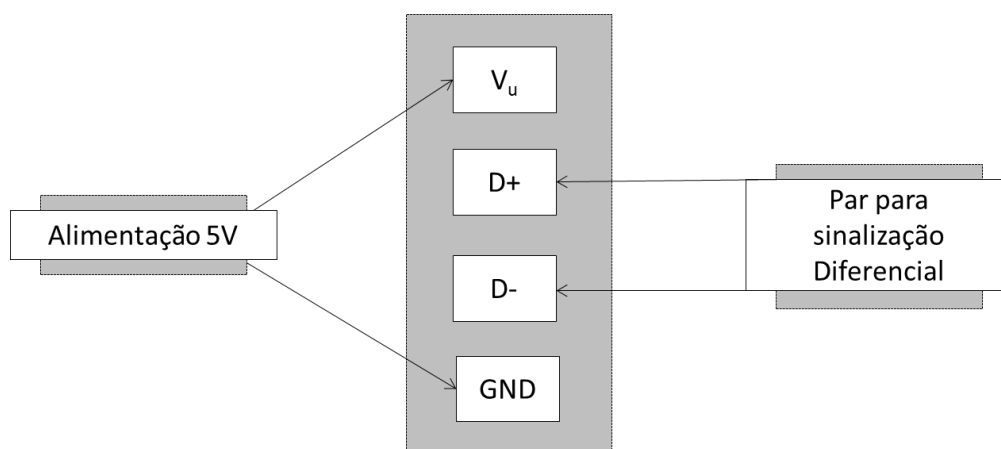
A utilização do protocolo JTAG foi indicada na proposta do PDETR, porém, durante o desenvolvimento do projeto, julgou-se necessário sua substituição. O protocolo C2 da *Silicon Labs*, similar ao IEEE 1149.1, foi adotado e será descrito neste relatório.

### 2.3. USB (*Universal Serial Bus*)

A interface USB surgiu originalmente como um esforço em conjunto de sete empresas (COMPAQ, INTEL, MICROSOFT, HP, LUCENT, DEC e PHILIPS) que tinham como objetivo padronizar uma comunicação entre o computador pessoal e seus periféricos. O trabalho de normatização e organização da documentação foi posteriormente repassado para uma organização independente, o USB-IF (*USB Implementation Forum*) (*UNIVERSAL SERIAL BUS SPECIFICATION*)<sup>10</sup>. De 1996 até 2011, foram lançadas cinco versões do protocolo USB. Em 1996 surgiu a Versão 1.0 que suportava as funções *Low Speed* (1,5 Mb/s) e *Full Speed* (12 Mb/s). Em 1998 a versão 1.1 adicionou melhorias e, principalmente, fez esclarecimentos sobre o protocolo. A função *High Speed* (480 Mb/s) foi introduzida junto com a versão 2.0 em 2000. Em 2001 surgiu a versão USB OTG que permite aos periféricos atuarem como *hosts* (mestres) e em 2008 a versão foi lançada com suporte a 4 Gb/s (*UNIVERSAL SERIAL BUS SPECIFICATION*)<sup>10</sup>.

As principais características do USB são o baixo custo, utilizar um único padrão de conectores e cabos, permitir a alimentação do periférico a partir do barramento (*Bus Powered Devices*), permitir a conexão com o sistema energizado (*Hot Pluggable*), suporte para reconhecimento automático de dispositivos (*Plug and Play*), previsão para mecanismos de detecção e correção de erros (*Handshaking*), previsão de mecanismos para conservação de energia (*Power Down*) (*UNIVERSAL SERIAL BUS SPECIFICATION*)<sup>10</sup>.

Fisicamente a conexão USB é implementada através de 4 fios (Figura 4).



**Figura 4 Características físicas do USB**

**Fonte: Autoria própria**

Quanto à comunicação entre dispositivos USB, existem três classes disponíveis: *Communication Device Class (CDC)*, *Communication Interface Class (CIC)* e *Data Interface Class (DIC)*. A *Communication Device Class* é uma definição de nível de dispositivo e é usada pelo *host* para identificar corretamente um dispositivo de comunicação que pode apresentar diferentes tipos de interface. Ela fornece uma classe de dispositivo único, porém pode haver mais de uma interface implementada (interfaces de controle, dados, áudio e armazenamento em massa). A CDC define um mecanismo de propósito geral que pode ser usado para habilitar todos os tipos de serviços de comunicação no USB. A DIC define um mecanismo de propósito geral para permitir a transferência de dados. Ela identifica as interfaces de transmissão de dados quando os dados não coincidem com a estrutura ou modelo de utilização para qualquer outro tipo de classe (UNIVERSAL SERIAL BUS SPECIFICATION)<sup>10</sup>.



### 3. DESENVOLVIMENTO

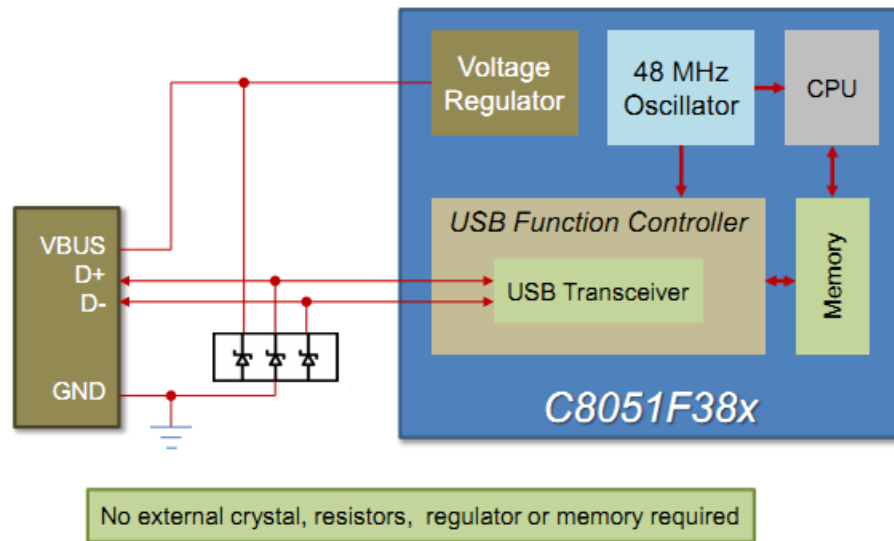
Os módulos elaborados durante o desenvolvimento do projeto serão apresentados neste capítulo. Inicialmente as características principais dos microcontroladores selecionados serão descritas com detalhes, uma vez que o conhecimento aprofundado do microcontrolador selecionado faz parte do desenvolvimento da PDETR. Na sequência será apresentado o protocolo de depuração em tempo real e suas funções utilizadas para gravar conteúdos na memória flash dos dispositivos alvos. Por fim, será apresentado o projeto das placas desenvolvidas e fabricadas.

#### 3.1. MICROCONTROLADORES

Frente ao grande número de fabricantes e famílias de controladores com arquitetura 8051, fez-se uma pesquisa de mercado com o objetivo de selecionar um modelo de controlador que atendesse, o mais próximo possível, as características propostas no projeto da PDETR. Levando em consideração as características técnicas do componente e sua representação de sua fabricante no mercado nacional, os microcontroladores selecionados para a placa didática e para o dispositivo de gravação foram, respectivamente, o C8051386 e o C8051321, ambos da fabricante *Silicon Labs*.

##### 3.1.1. C8051F386

O microcontrolador C8051F386 é um controlador de 8 *bits* da família C8051F38x desenvolvido pela *Silicon Labs*. Ele se enquadra na proposta inicial por ser um dispositivo de arquitetura 8051, com interfaces C2 e USB, que possui todas as suas atribuições e periféricos *on-chip*. Isto significa dizer que osciladores, resistores, reguladores de tensão, controlador e *transceiver* USB e memória programável estão integrados no mesmo *chip* conforme ilustra a Figura 5.



**Figura 5 - Família C805138x "on chip"**  
**Fonte: (Silicon Labs, 2011)**

As principais características do C8051F386 são (C8051F380/1/2/3/4/5/6/7. MANUAL. SILICON LABS)<sup>14</sup>:

- Arquitetura 8051;
- Programável em *assembly* por instruções CIP-51, equivalentes ao padrão MCS-51 desenvolvido pela Intel;
- Oscilador interno de frequência 48 MHz;
- Memória FLASH de 32 kbytes;
- Memória RAM 2304 bytes;
- Interface USB;
- Regulador de Tensão interno com saída 3.3 V;
- Duas interfaces SMBus/I2C (*System Management Bus/Inter-Integrated Circuit*);
- Interface SPI (*Serial Peripheral Interface*);
- Duas interfaces UART (*Universal Asynchronous Receiver/Transmitter*);
- Seis temporizadores (*Timers* de 16 bits);
- PCA (*Programmable Counter/Timer Array*) com 5 modos de captura Watchdog Timer;
- Quatro *ports* de Entrada/Saída (totalizando 40 pinos de e/s);
- Interface para memória externa;
- Dois comparadores analógicos;

- Interface de programação da memória FLASH (C2);

A Figura 6 mostra o diagrama de blocos de todos os módulos do C8051F386 e como os mesmos se relacionam.

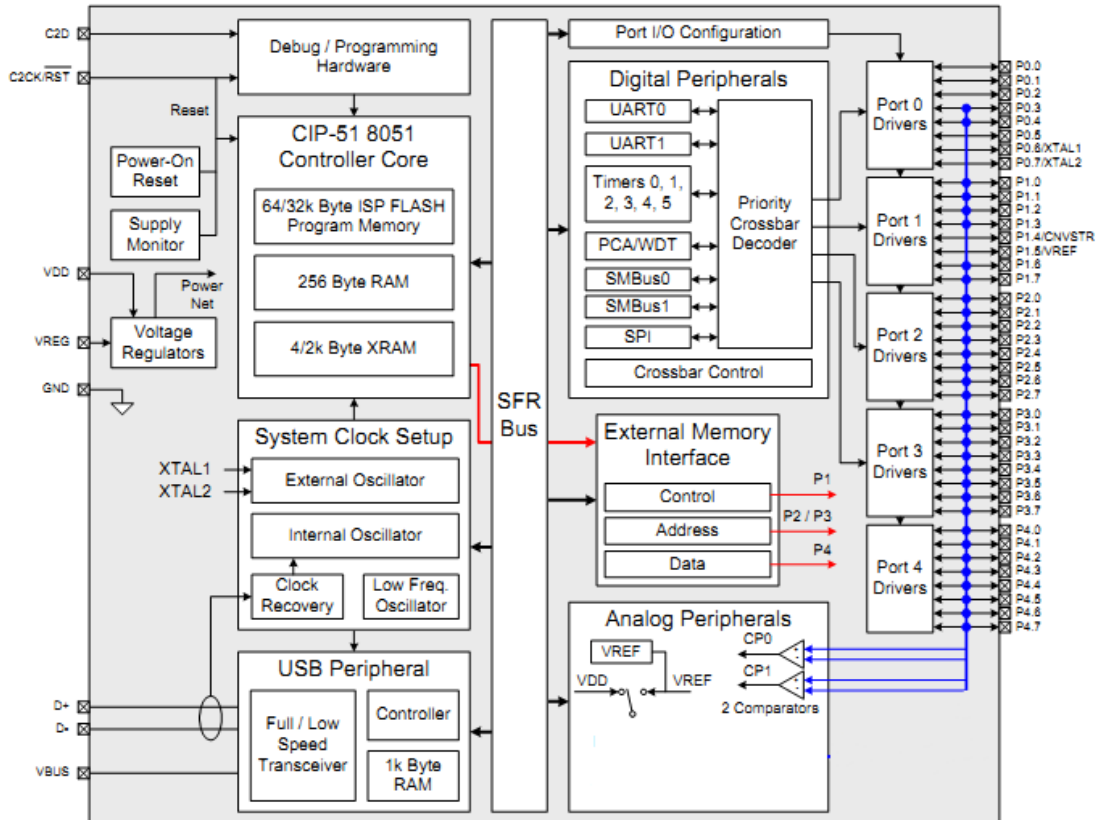
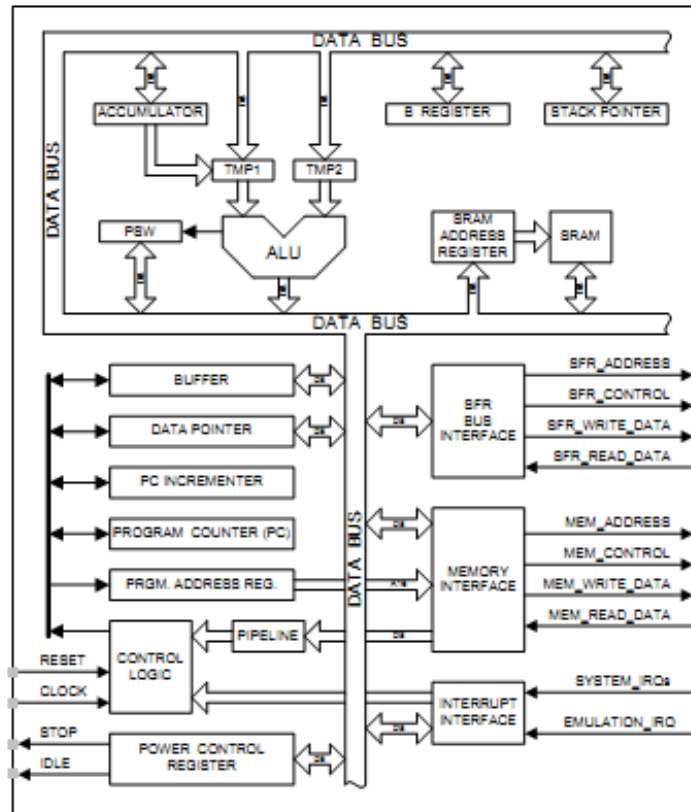


Figura 6 - Diagrama de Blocos do C8051F386  
Fonte: (Silicon Labs, 2011)

### 3.1.1.1 CIP-51

O núcleo do C8051F386 é o CIP-51, arquitetura patenteada e pertencente à *Silicon Labs* (Figura 7). Implementado em arquitetura padrão 8051, ele é totalmente compatível com o conjunto de instruções MCS-51™ e assembladores e compiladores 803x/805x (C8051F380/1/2/3/4/5/6/7. MANUAL. SILICON LABS)<sup>14</sup>. Diferente dos microcontroladores 8051 convencionais, o core CIP-51 utiliza o sistema *pipeline* para processar instruções, o que o torna mais rápido e eficiente. Um microcontrolador 8051 comum possui um *clock* de sistema de 12MHz e processa a maioria de suas instruções (exceção MUL e DIV) com 12 ou 24 ciclos de

*clock*. O núcleo CIP-51 executa 70% de suas instruções com um ou dois ciclos de *clock* em até 48MHz.



**Figura 7 - Diagrama de Blocos do CIP-51**  
Fonte: (Silicon Labs, 2011)

O CIP-51 possui 109 instruções. A Tabela 1 mostra o número de instruções que necessitam uma quantidade conhecida de ciclos de *clock* para serem executadas.

**Tabela 1 - Número de instruções x Ciclos de Clock**

Ciclos de clock	1	2	2/4	3	3/5	4	5	4/6	6	8
Nºde instruções	26	50	5	10	6	5	2	2	2	1

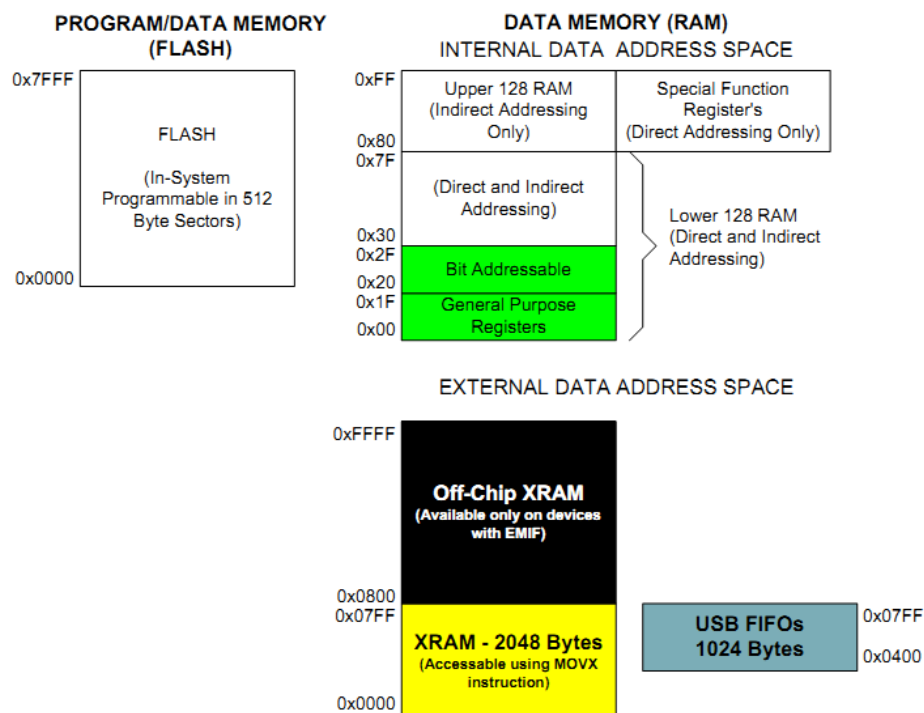
Fonte: (Silicon Labs, 2011)

O CIP-51 pode ser programado e depurado pela interface C2 da *Silicon Labs*. Esta interface, que será apresentada neste relatório, permite durante a

depuração a realização de configurações de hardware, inserir pontos de interrupção, iniciar interrupções, ler e escrever na memória, tudo sem utilizar recursos do *chip*, como o *timer* por exemplo. O *core* em questão é suportado pelo ambiente de desenvolvimento do fabricante e de terceiros.

### 3.1.1.2 Organização da memória

A organização da memória do controlador CIP-51 é semelhante a um 8051 convencional. Há dois espaços de memória diferentes: memória de programa e memória de dados. As duas compartilham o mesmo espaço de endereço, mas são acessadas através de diferentes tipos de instruções. A disposição da memória do CIP-51 é mostrada na Figura 8.



**Figura 8 - Mapa de memória on-chip do C8051F38**  
Fonte: (Silicon Labs)

### Memória de programa

O CIP-51 possui 32kbytes de memória de programa do tipo FLASH. Normalmente esta memória é apenas para leitura (*read-only*), porém o CIP51 pode escrever na memória de programa usando o *Program Store Write Enable bit* (PSCTL.0) associado à instrução MOVX. Este recurso fornece um mecanismo para a atualização do código do programa e liberação de espaço da memória de programa para utilizar como memória não volátil de armazenamento de dados.

### Memória de dados

O CIP-51 possui 256 bytes para memória RAM interna mapeados na memória de dados de 0x00 até 0xFF. Os primeiros 128 bytes da memória de dados são utilizados para registradores de propósito gerais. Os locais 0x00 até 0x1F são endereçados através de quatro banco de registradores de uso geral. Os próximos 16 bytes, 0x20 até 0x2F, podem ser tratadas como bytes ou 128 bits acessíveis de modo direto.

### Registradores de propósito geral

Os primeiros 32 bytes da memória de dados, localizados em 0x00 até 0x1F, podem ser tratados como 4 bancos de 8 registradores (R0 até R7). Esses bancos podem ser ativados um por vez através de 2 bits no registrador PSW (RS0 = PSW.3 e RS1 = PSW.4).

### Pilha

A pilha pode estar localizada em qualquer lugar da memória de dados. Sua área é indicada usando o *Stack Pointer* (SP, 0x81). O SP aponta para o último valor empilhado, assim, quando inserimos um valor na pilha, o ponteiro é incrementado na sequência. Toda vez que um *reset* acontece, o SP encontra-se no endereço 0x07. A pilha pode ter um tamanho máximo de 256 bytes.

### Registradores de funções especiais

O local 0x80 até 0xFF da memória de dados é destinado os registradores de funções especiais. Consultar a documentação do C8051F386 para obter a lista de registradores especiais.

### 3.1.1.3 Interrupções

O C8051F386 contém um sistema de interrupção que suporta múltiplas fontes de interrupção com dois níveis de prioridade. Quando o sistema recebe uma interrupção, a *flag* referente a esta fonte de interrupção vai para valor lógico '1'. A CPU gera um LCALL para um endereço pré-determinado que tratará a interrupção. O fim dessa rotina deve terminar com RETI, que retorna a execução do programa principal para a próxima instrução que seria executada se a interrupção não tivesse ocorrido. As interrupções devem ser globalmente habilitadas no *bit* EA (*Global Interrupt Enable*) do registrador IE.

#### Prioridades

Cada fonte de interrupção pode ser individualmente programada para um dos dois níveis de prioridade: alta ou baixa. A prioridade baixa é padrão. Uma rotina de tratamento de interrupção de baixa prioridade pode ser interrompida por uma interrupção de alta prioridade. Se duas interrupções definidas com o mesmo nível de prioridade acontecem simultaneamente, a ordem de atendimento de interrupção é dada segundo o Quadro 1. Cada fonte de interrupção tem um *bit* de prioridade associado nos registradores IP, EIP1 ou EIP2.

Interrupt Source	Interrupt Vector	Priority Order	Pending Flag	Bit Address?	Cleared by HW?	Enable Flag	Priority Control
Reset	0x0000	Top	None	N/A	N/A	Always Enabled	Always Highest
External Interrupt 0 (INT0)	0x0003	0	IE0 (TCON.1)	Y	Y	EX0 (IE.0)	PX0 (IP.0)
Timer 0 Overflow	0x000B	1	TF0 (TCON.5)	Y	Y	ET0 (IE.1)	PT0 (IP.1)
External Interrupt 1 (INT1)	0x0013	2	IE1 (TCON.3)	Y	Y	EX1 (IE.2)	PX1 (IP.2)
Timer 1 Overflow	0x001B	3	TF1 (TCON.7)	Y	Y	ET1 (IE.3)	PT1 (IP.3)
UART0	0x0023	4	RI0 (SCON0.0) TI0 (SCON0.1)	Y	N	ES0 (IE.4)	PS0 (IP.4)
Timer 2 Overflow	0x002B	5	TF2H (TMR2CN.7) TF2L (TMR2CN.6)	Y	N	ET2 (IE.5)	PT2 (IP.5)
SPI0	0x0033	6	SPIF (SPI0CN.7) WCOL (SPI0CN.6) MODF (SPI0CN.5) RXOVRN (SPI0CN.4)	Y	N	ESPI0 (IE.6)	PSPi0 (IP.6)
SMB0	0x003B	7	SI (SMB0CN.0)	Y	N	ESMB0 (EIE1.0)	PSMB0 (EIP1.0)
USB0	0x0043	8	Special	N	N	EUSB0 (EIE1.1)	PUSB0 (EIP1.1)
ADC0 Window Compare	0x004B	9	AD0WINT (ADC0CN.3)	Y	N	EWADC0 (EIE1.2)	PWADC0 (EIP1.2)
ADC0 Conversion Complete	0x0053	10	AD0INT (ADC0CN.5)	Y	N	EADC0 (EIE1.3)	PADC0 (EIP1.3)
Programmable Counter Array	0x005B	11	CF (PCA0CN.7) CCFn (PCA0CN.n)	Y	N	EPCA0 (EIE1.4)	PPCA0 (EIP1.4)
Comparator0	0x0063	12	CP0FIF (CPT0CN.4) CP0RIF (CPT0CN.5)	N	N	ECP0 (EIE1.5)	PCP0 (EIP1.5)
Comparator1	0x006B	13	CP1FIF (CPT1CN.4) CP1RIF (CPT1CN.5)	N	N	ECP1 (EIE1.6)	PCP1 (EIP1.6)
Timer 3 Overflow	0x0073	14	TF3H (TMR3CN.7) TF3L (TMR3CN.6)	N	N	ET3 (EIE1.7)	PT3 (EIP1.7)
VBUS Level	0x007B	15	N/A	N/A	N/A	EVBUS (EIE2.0)	PVBUS (EIP2.0)
UART1	0x0083	16	RI1 (SCON1.0) TI1 (SCON1.1)	N	N	ES1 (EIE2.1)	PS1 (EIP2.1)
Reserved	0x008B	17	N/A	N/A	N/A	N/A	N/A
SMB1	0x0093	18	SI (SMB1CN.0)	Y	N	ESMB1 (EIE2.3)	PSMB1 (EIP2.3)
Timer 4 Overflow	0x009B	19	TF4H (TMR4CN.7) TF4L (TMR4CN.6)	Y	N	ET4 (EIE2.4)	PT4 (EIP2.4)
Timer 5 Overflow	0x00A3	20	TF5H (TMR5CN.7) TF5L (TMR5CN.6)	Y	N	ET5 (EIE2.5)	PT5 (EIP2.5)

**Quadro 1 - Ordem de prioridade de interrupções**  
**Fonte: (Silicon Labs, 2011)**



### IE (Interrupt Enable)

O Quadro 2 decreve o registrador IE.

Bit	7	6	5	4	3	2	1	0
Name	EA	ESPI0	ET2	ES0	ET1	EX1	ET0	EX0
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xA8; SFR Page = All Pages; Bit-Addressable

Bit	Name	Function
7	EA	<b>Enable All Interrupts.</b> Globally enables/disables all interrupts. It overrides individual interrupt mask settings. 0: Disable all interrupt sources. 1: Enable each interrupt according to its individual mask setting.
6	ESPI0	<b>Enable Serial Peripheral Interface (SPI0) Interrupt.</b> This bit sets the masking of the SPI0 interrupts. 0: Disable all SPI0 interrupts. 1: Enable interrupt requests generated by SPI0.
5	ET2	<b>Enable Timer 2 Interrupt.</b> This bit sets the masking of the Timer 2 interrupt. 0: Disable Timer 2 interrupt. 1: Enable interrupt requests generated by the TF2L or TF2H flags.
4	ES0	<b>Enable UART0 Interrupt.</b> This bit sets the masking of the UART0 interrupt. 0: Disable UART0 interrupt. 1: Enable UART0 interrupt.
3	ET1	<b>Enable Timer 1 Interrupt.</b> This bit sets the masking of the Timer 1 interrupt. 0: Disable all Timer 1 interrupt. 1: Enable interrupt requests generated by the TF1 flag.
2	EX1	<b>Enable External Interrupt 1.</b> This bit sets the masking of External Interrupt 1. 0: Disable external interrupt 1. 1: Enable interrupt requests generated by the $\overline{\text{INT1}}$ input.
1	ET0	<b>Enable Timer 0 Interrupt.</b> This bit sets the masking of the Timer 0 interrupt. 0: Disable all Timer 0 interrupt. 1: Enable interrupt requests generated by the TF0 flag.
0	EX0	<b>Enable External Interrupt 0.</b> This bit sets the masking of External Interrupt 0. 0: Disable external interrupt 0. 1: Enable interrupt requests generated by the $\overline{\text{INT0}}$ input.

**Quadro 2 – Registrador IE**  
**Fonte: (Silicon Labs, 2011)**

### IP (Interrupt Priority)

O Quadro 3 descreve o registrador IP.

Bit	7	6	5	4	3	2	1	0
Name		PSPI0	PT2	PS0	PT1	PX1	PT0	PX0
Type	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	0	0	0	0	0	0	0

SFR Address = 0xB8; SFR Page = All Pages; Bit-Addressable

Bit	Name	Function
7	Unused	Read = 1b, Write = Don't Care.
6	PSPI0	<b>Serial Peripheral Interface (SPI0) Interrupt Priority Control.</b> This bit sets the priority of the SPI0 interrupt. 0: SPI0 interrupt set to low priority level. 1: SPI0 interrupt set to high priority level.
5	PT2	<b>Timer 2 Interrupt Priority Control.</b> This bit sets the priority of the Timer 2 interrupt. 0: Timer 2 interrupt set to low priority level. 1: Timer 2 interrupt set to high priority level.
4	PS0	<b>UART0 Interrupt Priority Control.</b> This bit sets the priority of the UART0 interrupt. 0: UART0 interrupt set to low priority level. 1: UART0 interrupt set to high priority level.
3	PT1	<b>Timer 1 Interrupt Priority Control.</b> This bit sets the priority of the Timer 1 interrupt. 0: Timer 1 interrupt set to low priority level. 1: Timer 1 interrupt set to high priority level.
2	PX1	<b>External Interrupt 1 Priority Control.</b> This bit sets the priority of the External Interrupt 1 interrupt. 0: External Interrupt 1 set to low priority level. 1: External Interrupt 1 set to high priority level.
1	PT0	<b>Timer 0 Interrupt Priority Control.</b> This bit sets the priority of the Timer 0 interrupt. 0: Timer 0 interrupt set to low priority level. 1: Timer 0 interrupt set to high priority level.
0	PX0	<b>External Interrupt 0 Priority Control.</b> This bit sets the priority of the External Interrupt 0 interrupt. 0: External Interrupt 0 set to low priority level. 1: External Interrupt 0 set to high priority level.

**Quadro 3 – Registrador IP**  
**Fonte: (Silicon Labs, 2011)**

### Interrupções Externas (INT0 E INT1)

A INT0 e INT1, fontes de interrupções externas, podem ser ativas em baixo ou alto e por borda ou nível. Os *bits* IN0PL (INT0 *polarity*) e IN1PL (INT1 *polarity*), pertencentes ao registrador IT01CFestabelecem se as interrupções serão ativas em alto ou baixo. Os *bits* IT0 e IT1, do registrador TCON, seleccionam se determinada interrupção será sensível a nível ou a borda. Ver Quadro 4.

IT0	IN0PL	INT0 Interrupt	IT1	IN1PL	INT1 Interrupt
1	0	Active low, edge sensitive	1	0	Active low, edge sensitive
1	1	Active high, edge sensitive	1	1	Active high, edge sensitive
0	0	Active low, level sensitive	0	0	Active low, level sensitive
0	1	Active high, level sensitive	0	1	Active high, level sensitive

**Quadro 4 – Configuração de interrupções**  
**Fonte: (Silicon Labs 2011)**

A INT0 e INT1 são definidas em qual pino e *port* serão acionadas no registrador IT01CF como mostra o Quadro 5.

Bit	7	6	5	4	3	2	1	0
Name	IN1PL	IN1SL[2:0]			IN0PL	IN0SL[2:0]		
Type	R/W	R/W			R/W	R/W		
Reset	0	0	0	0	0	0	0	1

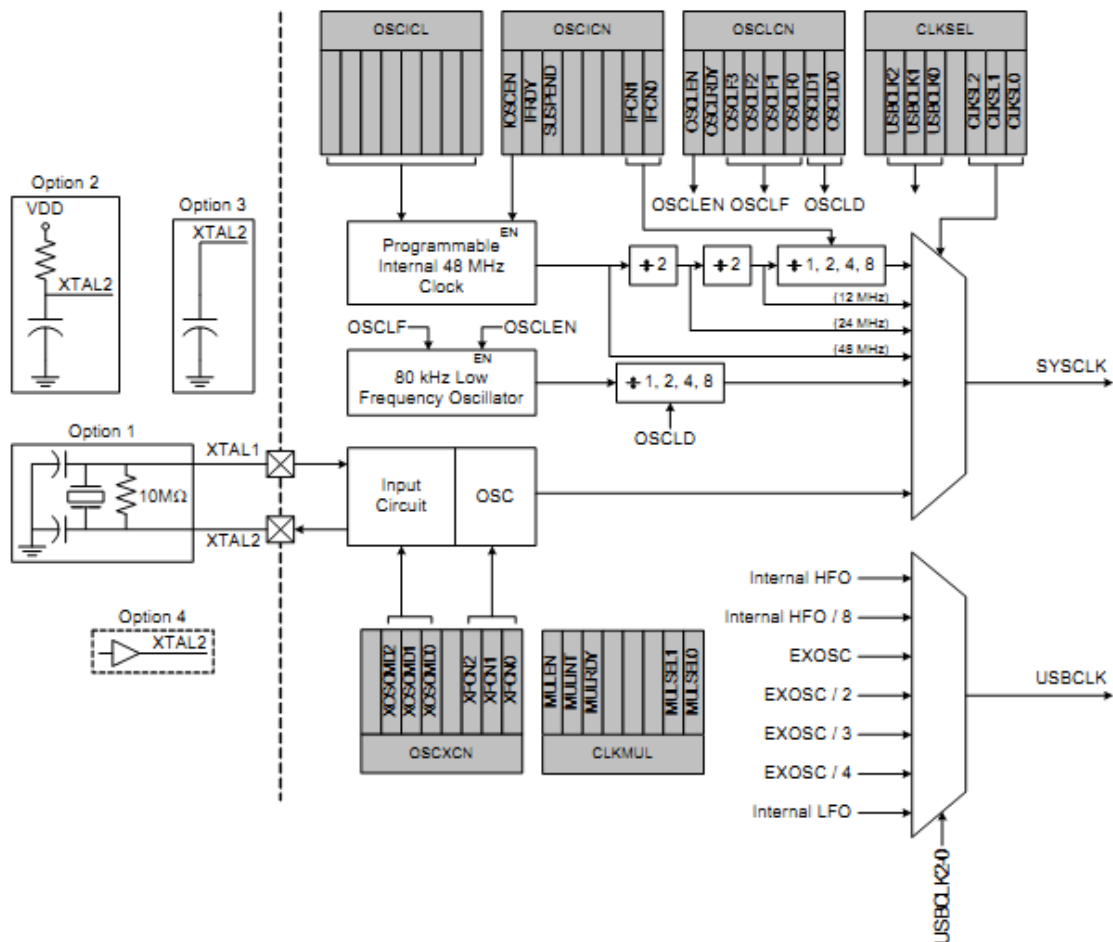
SFR Address = 0xE4; SFR Page = 0

Bit	Name	Function
7	IN1PL	<b>INT1 Polarity.</b> 0: $\overline{\text{INT1}}$ input is active low. 1: INT1 input is active high.
6:4	IN1SL[2:0]	<b>INT1 Port Pin Selection Bits.</b> These bits select which Port pin is assigned to $\overline{\text{INT1}}$ . Note that this pin assignment is independent of the Crossbar; $\overline{\text{INT1}}$ will monitor the assigned Port pin without disturbing the peripheral that has been assigned the Port pin via the Crossbar. The Crossbar will not assign the Port pin to a peripheral if it is configured to skip the selected pin. 000: Select P0.0 001: Select P0.1 010: Select P0.2 011: Select P0.3 100: Select P0.4 101: Select P0.5 110: Select P0.6 111: Select P0.7
3	IN0PL	<b>INT0 Polarity.</b> 0: $\overline{\text{INT0}}$ input is active low. 1: INT0 input is active high.
2:0	IN0SL[2:0]	<b>INT0 Port Pin Selection Bits.</b> These bits select which Port pin is assigned to $\overline{\text{INT0}}$ . Note that this pin assignment is independent of the Crossbar; $\overline{\text{INT0}}$ will monitor the assigned Port pin without disturbing the peripheral that has been assigned the Port pin via the Crossbar. The Crossbar will not assign the Port pin to a peripheral if it is configured to skip the selected pin. 000: Select P0.0 001: Select P0.1 010: Select P0.2 011: Select P0.3 100: Select P0.4 101: Select P0.5 110: Select P0.6 111: Select P0.7

**Quadro 5– Registrador IT01CF**  
**Fonte: (Silicon Labs, 2011)**

### 3.1.1.4 Oscilador e Clock

O C8051F386 possui um oscilador interno programável de alta frequência, um oscilador interno programável de baixa frequência e um circuito para oscilador externo. O oscilador interno de alta frequência pode ser ativado e ajustado nos registradores OSCICN e OSCICL como mostra a Figura 9. O oscilador interno de baixa frequência pode ser ativado e ajustado utilizando o registrador OSCLCN. O *clock* da interface USB (USBCLK) também pode ser derivado dos osciladores internos ou externo.



**Figura 9- Opções de osciladores**  
**Fonte: (Silicon Labs, 2011)**

A fonte de oscilação é definida programando o registrador CLKSEL.

CLKSEL.

O Quadro 6 decreve o registrador CLKSEL.

Bit	7	6	5	4	3	2	1	0
Name		USBCLK[2:0]			OUTCLK	CLKSL[2:0]		
Type	R	R/W			R/W	R/W		
Reset	0	0	0	0	0	0	0	0

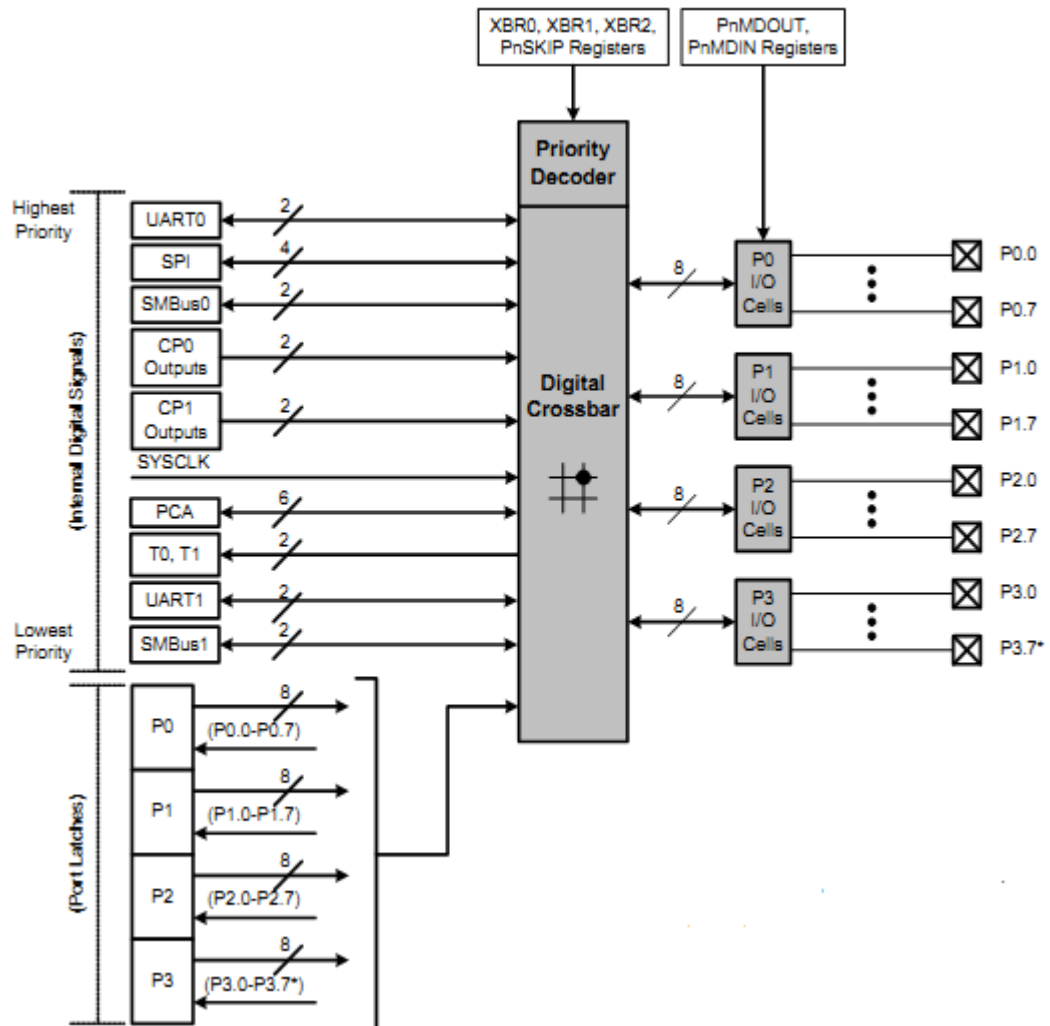
SFR Address = 0xA9; SFR Page = All Pages

Bit	Name	Function
7	Unused	Read = 0b; Write = don't care
6:4	USBCLK[2:0]	<b>USB Clock Source Select Bits.</b> 000: USBCLK derived from the Internal High-Frequency Oscillator. 001: USBCLK derived from the Internal High-Frequency Oscillator / 8. 010: USBCLK derived from the External Oscillator. 011: USBCLK derived from the External Oscillator/2. 100: USBCLK derived from the External Oscillator/3. 101: USBCLK derived from the External Oscillator/4. 110: USBCLK derived from the Internal Low-Frequency Oscillator. 111: Reserved.
3	OUTCLK	<b>Crossbar Clock Out Select.</b> If the $\overline{\text{SYSCLK}}$ signal is enabled on the Crossbar, this bit selects between outputting $\overline{\text{SYSCLK}}$ and $\overline{\text{SYSCLK}}$ synchronized with the Port I/O pins. 0: Enabling the Crossbar $\overline{\text{SYSCLK}}$ signal outputs $\overline{\text{SYSCLK}}$ . 1: Enabling the Crossbar $\overline{\text{SYSCLK}}$ signal outputs $\overline{\text{SYSCLK}}$ synchronized with the Port I/O.
2:0	CLKSL[2:0]	<b>System Clock Source Select Bits.</b> 000: $\overline{\text{SYSCLK}}$ derived from the Internal High-Frequency Oscillator and scaled per the IFCN bits in register OSCICN. 001: $\overline{\text{SYSCLK}}$ derived from the External Oscillator circuit. 010: $\overline{\text{SYSCLK}}$ derived from the Internal High-Frequency Oscillator / 2. 011: $\overline{\text{SYSCLK}}$ derived from the Internal High-Frequency Oscillator. 100: $\overline{\text{SYSCLK}}$ derived from the Internal Low-Frequency Oscillator and scaled per the OSCLD bits in register OSCLCN. 101-111: Reserved.

**Quadro 6 – Registrado CLKSEL**  
**Fonte: (Silicon Labs, 2011)**

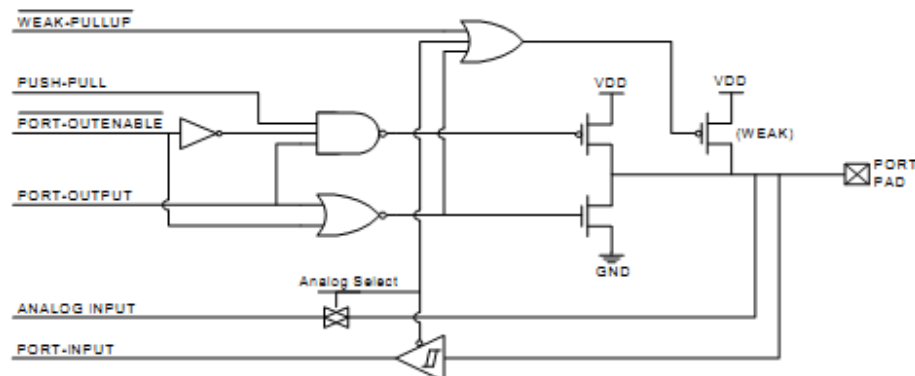
### 3.1.1.5 Entradas e saídas

O C8051F386 possui 40 pinos de entradas e saídas que estão distribuídos conforme a Figura 10. Cada pino pode ser definido como de uso geral ou entrada analógica. Essa flexibilidade proporciona ao programador ter total controle sobre todos os pinos do microcontrolador, ficando seus projetos limitados somente ao número de pinos disponíveis. Essa flexibilidade de programação de recursos é obtida através de um decodificador denominado *Digital Crossbar*.



**Figura 10 - Diagrama de blocos dos pinos de entrada e saída**  
**Fonte: (Silicon Labs)**

A Figura 11 mostra o diagrama esquemático das possíveis configurações de um pino de entrada e saída.



**Figura 11 - Esquemático de um pino**  
**Fonte: (Silicon Labs)**

### 3.1.1.6 Timer

O C8051F386 possui 6 contadores (*timers/counters*). Dois de 16 *bits* de uso geral e 4 de recarga automática para uso com a SMBus ou para uso geral com configuração e modos de funcionamento iguais ao 8051 da Intel. Estes *timers* podem ser utilizados para contagem de tempo, interrupções ou contagem de eventos externos.

Timer 0 and Timer 1 Modes:	Timer 2, 3, 4, and 5 Modes:
13-bit counter/timer	16-bit timer with auto-reload
16-bit counter/timer	
8-bit counter/timer with auto-reload	Two 8-bit timers with auto-reload
Two 8-bit counter/timers (Timer 0 only)	

**Quadro 7– Distribuição dos Timers**  
**Fonte: (Silicon Labs)**

O *clock* para os *timers* são definidos nos registradores CKCON e CKCON1. Os *timers* são controlados programando o registrador TCON e definidos seus modos de operação no TMOD.

### 3.1.2. C8051F321

O microcontrolador C8051F321 é um controlador de 16 *bits* da família C8051F32x desenvolvido pela *Silicon Labs*. Sua arquitetura (registradores, interrupções e *timer*) é semelhante a do C8051F31, com algumas diferenças pontuais que serão listadas abaixo.

- Oscilador interno de 25 MHz;
- Memória FLASH de 16 *kbytes*;
- Uma interface SMBus/I2C;
- Uma interface UART;
- Quatro *timers* (16 *bits*);
- Dois *ports* de Entrada/Saída (21 pinos – Tensão máxima 5V);
- Sensor de temperatura;

- Conversor Digital Analógico de 10 *bits*.

### 3.2. PROTOCOLO SILICON LABS C2

O protocolo *Silicon Labs 2-wire*, ou C2, é um protocolo de comunicação serial a dois fios desenvolvido de maneira a permitir a programação em seu próprio sistema (*in-system*), depuração (*debugging*) e varredura periférica (*boundary scan*) utilizando-se de baixo número de conexões físicas para dispositivos *Silicon Labs*. Esse modelo de comunicação envolve um dispositivo mestre (*master*), responsável pelas atividades de programação, teste e depuração em um dispositivo alvo, ou escravo, cuja interface será submetida a tais ações.

Os dois pinos físicos envolvidos são o de dados (C2D) e o *clock* (C2CK), entretanto a característica mais interessante é que o protocolo facilita o esquema de compartilhamento de pinos, visto que tais ligações não são exclusivas à interface C2 (C2I), ao contrário dos pinos JTAG. Quando o dispositivo não está em uma comunicação com o mestre, ou seja, quando está ocioso, ambos os pinos ficam disponíveis para utilização do usuário na aplicação desenvolvida.

Todos os ciclos de comunicação são iniciados com um comando *START* no pino C2CK, o que permite que sejam enviados dados para a interface através de C2D e a configuração do uso da interface seja iniciada, e encerrados por um comando *STOP*, também em C2CK. No caso, C2CK é compartilhado com a funcionalidade de *reset* em nível baixo (*/RST*). Cabe ressaltar que para a inicialização da interface C2 o pino C2D deve ser alocado antes do comando *START* para tal comunicação e o mesmo deixa, desta forma, de desempenhar as funções para a qual havia sido configurado pelo usuário. Ao final da comunicação o dispositivo mestre disponibiliza novamente o pino C2D, permitindo que o mesmo volte a trabalhar como outrora configurado pelo usuário.

A interface C2 é similar à interface JTAG, porém, com os três sinais de dados (TDI, TDO e TMS) mapeados no pino bidirecional C2D. O sentido do fluxo, no caso de C2D, é determinado pelas instruções definidas no escopo do protocolo de forma que nunca haja conflitos entre dispositivo mestre e escravos.

Ainda, esta interface de dois fios foi concebida para que o acesso às funcionalidades de programação e depuração sejam feitas apenas por um registrador de endereço e um grupo de registradores de dados, como mostrado pelo diagrama de blocos da interface C2 na Figura 12. O valor do registrador de



endereço, no caso, define qual registrador de dados será acessado nos ciclos de escrita e leitura, de maneira similar ao registrador de instruções JTAG, o que é possível por meio de um único *shift-register* que trabalha como um conversor paralelo-serial-paralelo comum para a interface.

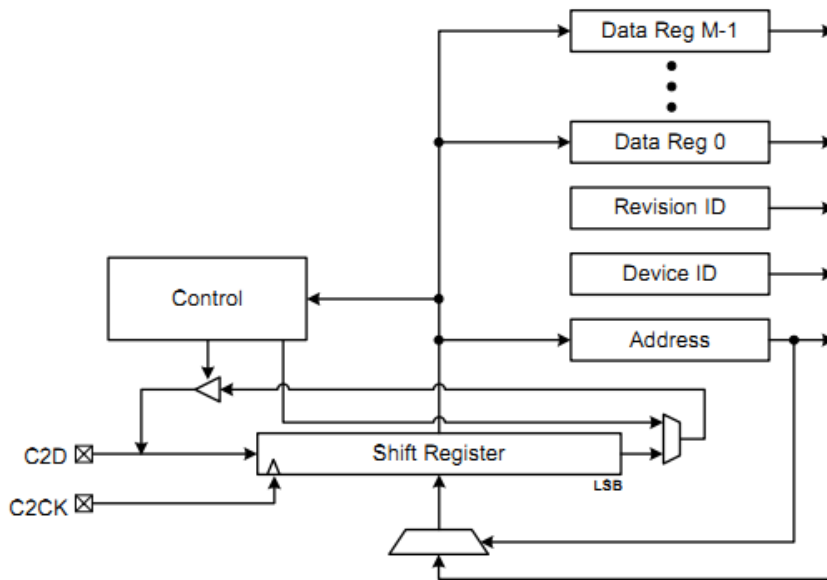


Figura 12- Diagrama de blocos da interface C2 (C2I)  
Fonte: (Silicon Labs, 2011).

### 3.2.1. Modelo de instruções

Um dispositivo mestre C2 acessa um escravo por meio de um grupo de quatro modelos básicos de instruções: Leitura de Endereço (*Address Write*), Escrita de Endereço (*Address Read*), Escrita de Dados (*Data Write*) e Leitura de Dados (*Data Read*). Tais modelos são resumidos na Figura 13 abaixo:

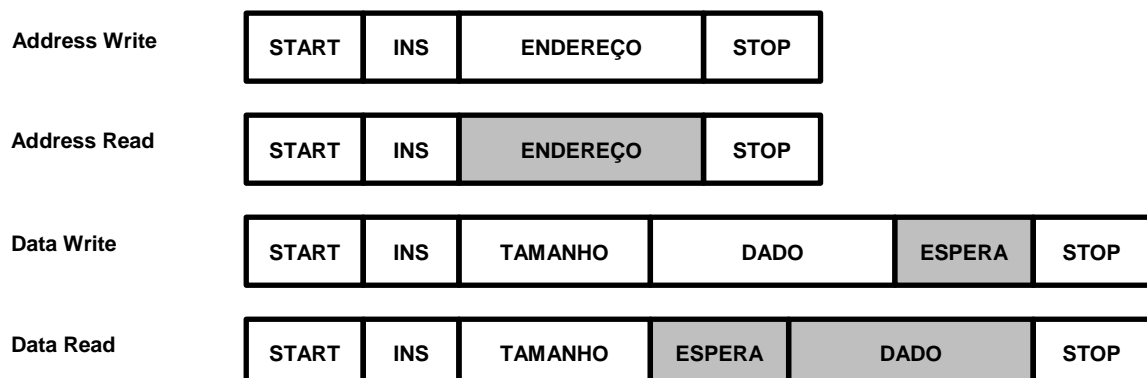


Figura 13 - Modelo de instruções do protocolo C2

**Fonte: (adaptado de Silicon Labs, 2011)**

As etapas sombreadas indicam que o sinal C2 é controlado pelo dispositivo escravo.

**START:** É o comando que inicializa um ciclo C2 de instrução. Apenas o dispositivo mestre pode gerar essa condição ao desabilitar o pino C2D e chavear o pino C2CK com um pulso de valor lógico '0' para '1'.

**INS:** Os quatro valores possíveis para determinar a instrução a ser realizada pelo dispositivo mestre no escravo são descritas no Quadro 8.

Instrução	Código INS
Leitura de Dados ( <i>Data Read</i> )	00 b
Leitura de Endereço ( <i>Address Write</i> )	10 b
Escrita de Dados ( <i>Data Write</i> )	01 b
Escrita de Endereço ( <i>Address Read</i> )	11 b

**Quadro 8 - Instruções C2**  
**Fonte: (Adaptado de Silicon Labs, 2011)**

**TAMANHO:** O campo relativo ao tamanho é um código de 2 *bits* que indica o comprimento do bloco DADO em quantidade de *bytes* a ser escrita ou lida durante um ciclo de acesso aos dados. O número de *bytes* é dado pela relação abaixo:

$$\text{DADO (em bytes)} = \text{TAMANHO} + 1$$

Por exemplo, quando o valor enviado em TAMANHO durante um ciclo de leitura de dados for 01b, o resultado será a leitura de 2 *bytes* de dados.

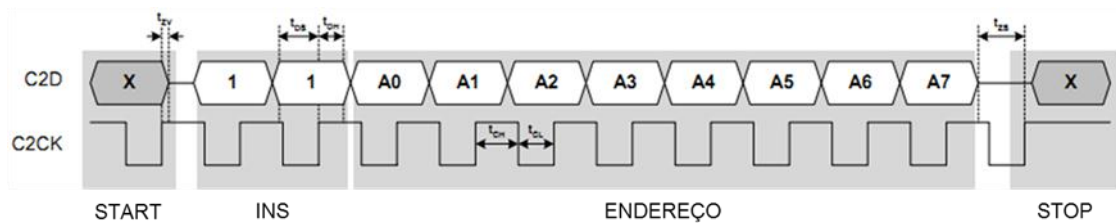
**ENDEREÇO:** Este bloco é utilizado para o envio e recepção de endereços e seu comprimento deve ser o mesmo do registrador de endereço no dispositivo escravo, 8 *bits* no caso do C8051F386.

**DADO:** Este bloco só aparece nos ciclos de escrita e leitura de dados e seu comprimento é determinado pelo bloco TAMANHO.

**ESPERA:** O bloco ESPERA é apenas utilizado pelo dispositivo escravo durante ciclos de escrita ou leitura de dados. Ele é composto por uma sequência de zero ou mais *bits* '0' encerrados por um único *bit* '1', indicando a disponibilidade para iniciar uma leitura ou o fim de uma escrita de dados.

### 3.2.2. Ciclo de escrita de endereço (11b)

O ciclo de Escrita de Endereço, ou *Address Write*, realiza o carregamento do registrador de endereço no dispositivo escravo. A sequência é novamente exemplificada pela Figura 14.

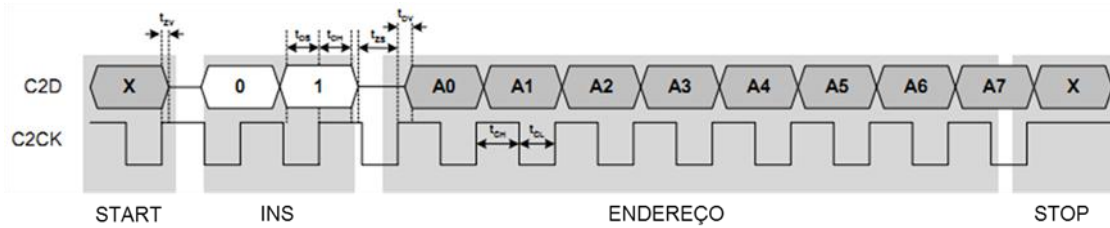


**Figura 14 - Ciclo de escrita de endereço**  
**Fonte: (Adaptado de Silicon Labs, 2011)**

O comprimento do bloco ENDEREÇO deve ser sempre igual ao comprimento do registrador de endereço do dispositivo alvo. Após uma reinicialização (RESET) o mesmo tem como padrão o valor 0x00h, correspondendo ao endereço do registrador *Device ID* que é responsável pela identificação do componente, no caso do C8051F386 este valor é 0x28h. Cabe lembrar que a transmissão é realizada enviando os *bits* menos significativos primeiro (*LSB first*).

### 3.2.3. Ciclo de leitura de endereço (10b)

O ciclo de Leitura de Endereço retorna informações de estado ou conteúdos do registrador de endereços em questão do dispositivo alvo. Essa instrução é geralmente utilizada para realizar um acesso ao estado do registrador, muito embora isto dependa da função de cada registrador de endereço dos dispositivos alvo. Novamente a sequência do comando é exibida na Figura 15

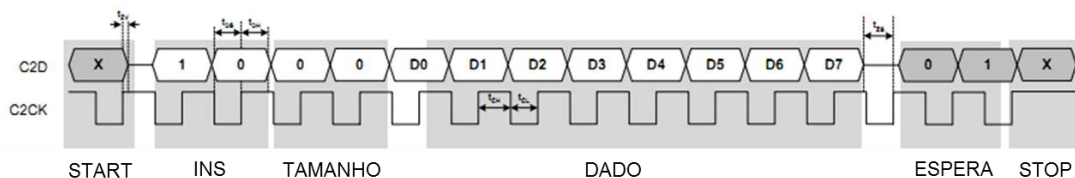


**Figura 15 - Ciclo de leitura de endereço**  
**Fonte: (Adaptado de Silicon Labs, 2011)**

Novamente, cabe lembrar que o comprimento do campo endereço deve ser igual ao registrador de endereço do dispositivo alvo e que os dados são transmitidos iniciando pelos *bits* menos significativos (*LSB first*).

#### 3.2.4. Ciclo de escrita de dados (01b)

O ciclo de Escrita de Dados escreve um valor em um registrador selecionado por um ciclo imediatamente anterior de Escrita de Endereço. A sequência do comando é descrito pela Figura 16.



**Figura 16 - Ciclo de escrita de dados**  
**Fonte: (Adaptado de Silicon Labs, 2011)**

Como já mencionado, o bloco TAMANHO é composto por 2 *bits* que especificam o comprimento do campo DADO a ser escrito, em seguida, pela equação:

$$\text{DADO (em Bytes)} = \text{TAMANHO} + 1$$

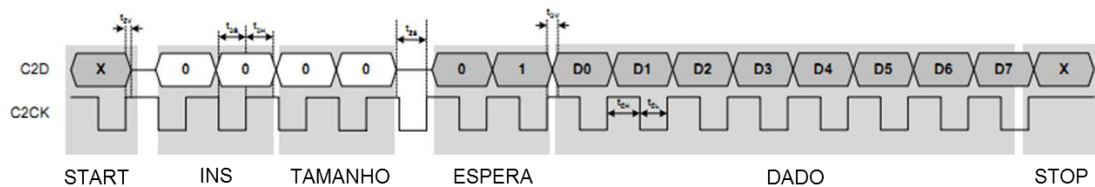
Assim sendo, o campo DADO deve compreender um múltiplo de 8 *bits*. Entretanto, a informação a ser transmitida no bloco DADO não precisa ser necessariamente do mesmo tamanho, podendo ter um comprimento menor do que o especificado pela equação acima. É possível, por exemplo, especificar um comprimento de 16 *bits* para o bloco com a intenção de enviar apenas dez *bits* de

informação. Novamente, os dados são transmitidos iniciando-se pelo bit menos significativo (*LSB first*).

O comprimento do bloco ESPERA é controlado pelo dispositivo escravo, sendo transmitida uma sequência indeterminada de pulsos de nível lógico '0' no pino C2D, até que a escrita seja completada, seguida de um pulso '1' que o encerra

### 3.2.5. Ciclo de Leitura de Dados (00b)

O ciclo de Leitura de Dados lê o conteúdo de um registrador selecionado durante um ciclo de Leitura de Endereço imediatamente anterior e a sequência de comandos para isto é descrita na Figura 17.



**Figura 17 - Ciclo de leitura de dados**  
**Fonte: (Adaptado de Silicon Labs, 2011)**

Como mencionado no Ciclo de Escrita de Dados, o bloco TAMANHO especifica o comprimento do bloco DADO a ser recebido e deve, também, ser um valor múltiplo de 8 *bits*. Não é necessário, como no caso anterior, que o comprimento do registrador seja exatamente igual ao especificado em TAMANHO, podendo ocorrer o recebimento de um número de *bits* inferior ao reservado para a recepção de acordo com o que for determinado para cada registrador do dispositivo alvo.

O recebimento também é feito iniciando-se pelo *bit* menos significativo (*LSB first*), só tendo início, entretanto, após o envio de um pulso de nível lógico '1' antecedido por uma sequência indeterminada de pulsos '0' que indicam a disponibilidade dos dados a serem enviados pelo dispositivo alvo.

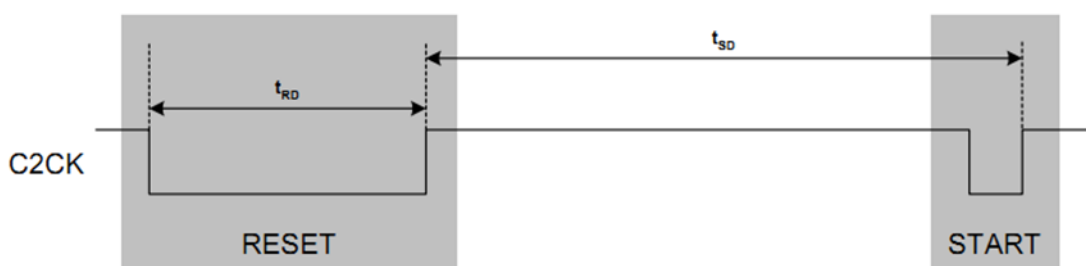
### 3.2.6. Considerações sobre os tempos

Como visto nas figuras descritivas para cada ciclo de instrução, o dispositivo alvo amostra os dados recebidos e disponibiliza os dados a serem enviados sempre nas bordas de subida, ou positivas, do pino C2CK. Em seguida, no Quadro 9, os parâmetros de tempo limites mínimos e máximos são listados.

Parâmetro	Descrição	Mínimo	Máximo
$t_{RD}$	Duração de C2CK para reinicialização do dispositivo alvo	20 $\mu$ s	-
$t_{SD}$	Atraso de START após reinicialização do dispositivo alvo	2 $\mu$ s	-
$t_{CL}$	C2CK em nível lógico '0' para transmissão de bit	20 ns	5000 ns
$t_{CH}$	C2CK em nível lógico '1'	20 ns	-
$t_{DS}$	Disponibilidade de C2D	10 ns	-
$t_{DH}$	Duração de C2D	10 ns	-
$t_{ZS}$	Disponibilidade de C2D em nível lógico 'Z'	0ns	-
$t_{DV}$	Validade de C2D	-	20 ns
$t_{ZV}$	Validade de C2d em nível lógico 'Z'	-	20 ns

**Quadro 9 - Tempos limites**  
**Fonte: (Adaptado de Silicon Labs, 2011)**

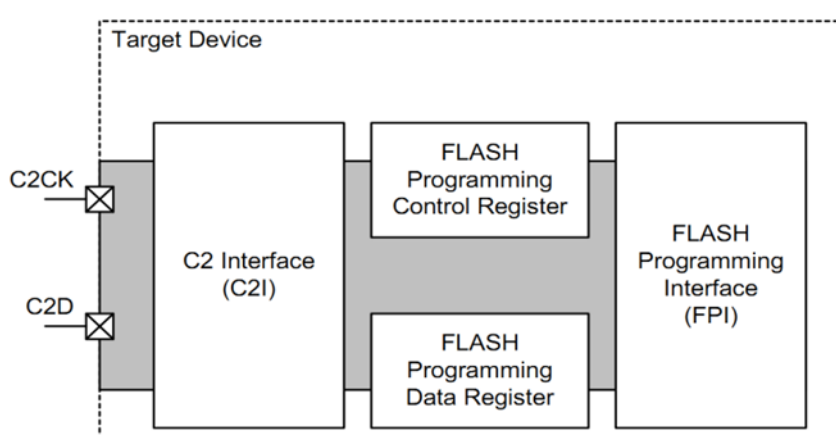
Durante as instruções do protocolo C2 o pino C2CK não deve ser mantido em nível lógico baixo por um tempo superior a  $t_{CL}$ . Esta condição permite que o dispositivo seja reinicializado (RESET) através da permanência de C2CK em nível lógico '0' por um tempo de  $t_{RD}$  (Figura 18). Então, o bloco de START da primeira instrução C2 deve iniciar em um intervalo de no mínimo  $t_{SD}$  após C2CK ter nível lógico alto após a reinicialização.



**Figura 18 - Diagrama de reinicialização do dispositivo alvo**  
**Fonte: (Silicon Labs, 2011)**

### 3.2.7. Gravação da memória FLASH através do protocolo C2

Os microcontroladores da série C8051F3XX são dotados de uma interface para programação da memória FLASH (FPI), a qual é acessada através da interface (C2I) utilizando-se de registradores padronizados para tal tarefa como demonstrado na Figura 19.



**Figura 19 - Arquitetura da interface de programação da FLASH via C2**  
 Fonte: (Adaptado de Silicon Labs, 2011)

As informações para realização deste acesso podem ser divididas em três seções:

1. Interface C2 (C2I)
  - a. Os pinos de dados (C2D) e de *clock* (C2CK);
  - b. Os registradores de endereço C2 e seus comandos de acesso de leitura (*Address Read*) e escrita (*Address Write*);
  - c. Os registradores para acesso aos dados e seus respectivos comandos (*Data Write e Data Read*);
  - d. Os registradores relacionados ao dispositivo (*Device ID*) e versão (*Revision ID*).
  
2. Registradores para a programação da FLASH
  - a. Registrador de controle da programação da FLASH (FPCTL);
  - b. Registrador de dados da FLASH (FPDAT).
  
3. Interface de programação da FLASH (FPI), que inclui comandos e procedimentos
  - a. Leitura (*FLASH Block Read*);
  - b. Escrita (*FLASH Block Write*);
  - c. Limpeza de página da FLASH (*FLASH Page Erase*);

d. Limpeza da FLASH completa (FLASH *User Space Erase*).

A interface C2 (C2I) é composta de um grupo de registradores de endereços, através dos quais se pode acessar até 256 registradores de dados, cada um com 1 *byte* de comprimento. Isto é possível através da camada física do protocolo C2.

### 3.3.7.1 Interface C2 (C2I)

O registrado de endereço (ADDRESS) serve para dois propósitos na interação com a memória FLASH:

1. O ADDRESS seleciona qual registrador de dados C2 será acessado durante a os ciclos de leitura e escrita.
2. Durante os ciclos de leitura adquire informações sobre o *status* da FPI.

Os valores de 8 *bits* retornados nos ciclos de leitura são utilizados como uma verificação entre a tarefa de programação e a FPI, como descrito no Quadro 10.

Bit	Descrição
[7..2]	Sem uso
1	InBusy: é alterado para '1' pela interface C2 quando se inicia a escrita em FPDAT. A FPI altera-o para '0' quando encerra a escrita.
0	OutReady: é alterado para '1' quando a FPI disponibiliza dados em FPDAT.

**Quadro 10 - Valores de retorno**  
**Fonte: (Adaptado de Silicon Labs, 2011)**

Dessa forma deve-se verificar o valor Ocupado (*InBusy*) ao final de cada ciclo de escrita e o valor Disponível (*OutReady*) antes de cada ciclo de leitura.

### 3.3.7.2 Registrador de identificação do dispositivo (DEVICEID)

Esse registrador de 8 *bits*, disponível apenas para leitura, identifica qual o dispositivo sendo gravado no momento. O endereço C2 associado ao registrador DEVICEID tem valor 0x00. No caso do dispositivo utilizado no projeto, o C8051F386, o DEVICEID retorna o valor 0x28h para identificação do mesmo.



### 3.3.7.3 Registrador identificador de versão (REVID)

O registrador REVID tem endereço C2 associado de valor 0x01, também disponível apenas para leitura e composto de 8 *bits*.

### 3.3.7.4 Registradores para a programação da FLASH

A comunicação entre a interface C2 e a FPI é viabilizada através de dois registradores já mencionados, FPCTL, ao qual o endereço associado é 0x02, e FPDAT, cujo endereço é 0xB4.

**FPCTL:** O registrador FPCTL tem a função de habilitar a programação da FLASH, para tanto é necessária a escrita sequencial dos códigos, após um comando *reset*–reinicialização – do dispositivo, abaixo na ordem:

- a. 0x02
- b. 0x01

Uma vez que os valores acima são escritos no registrador FPCTL, o dispositivo a ser programado fica suspenso até que um comando de reinicialização seja recebido.

**FPDAT:** O registrador de dados da FLASH é utilizado para toda a transferência de informações entre as interfaces C2I e a FPI, o que inclui:

- a. Envio de comandos da C2I para a FPI;
- b. Recepção de informações de estado enviadas da FPI para a C2I;
- c. Endereços da FLASH vindas da C2I para a FPI;
- d. Dados tanto da C2I para a FPI quanto no sentido inverso.

Os valores de estado retornados através da FPI para a interface C2 são exibidos no Quadro 11.

Valor	Estado
0x00h	Comando inválido
0x02h	Comando falhou
0x0Dh	Comando OK

**Quadro 11 - Valores de estado**

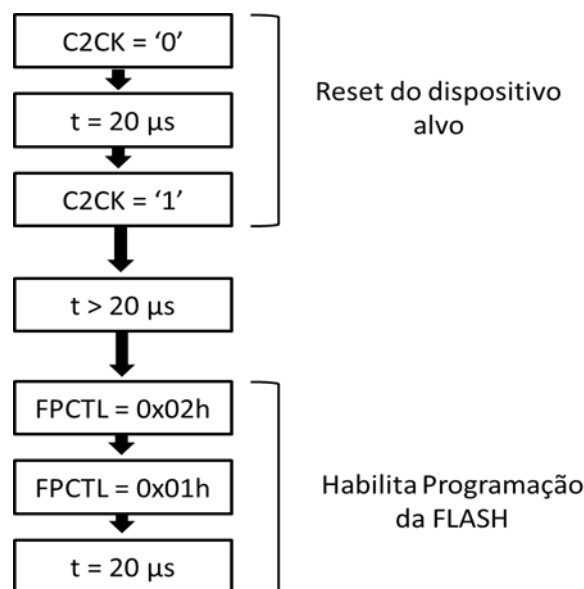
### 3.3.7.5 Interface de programação da FLASH

A FPI realiza um conjunto de quatro comandos, cada um executando uma série de escritas e leituras através do registrador FPDAT:

- a. Escrita em um bloco da memória FLASH – 0x07;
- b. Leitura de um bloco da memória FLASH – 0x06;
- c. Limpeza de uma página da memória FLASH – 0x08;
- d. Limpeza da memória FLASH completa – 0x03.

Entretanto, antes do envio de qualquer um destes comandos é necessária a inicialização da FPI através da sequência padronizada (Figura 20):

1. Reset do dispositivo a ser gravado, descrita na figura a seguir;
2. Atraso de no mínimo  $2\mu s$  antes do próximo comando;
3. Escrita do valor 0x02, seguido do valor 0x01, no registrador FPCTL;
4. Atraso de no mínimo  $20\ ms$  antes do próximo comando.



**Figura 20- Inicialização da FPI**  
 Fonte: (Adaptado de Silicon Labs, 2011)

### 3.3.7.6 Escrita de um bloco da memória FLASH

Todas as escritas na memória FLASH do microcontrolador alvo (*target device*) a ser programado são realizadas através do comando FLASH *Write Block*, já mencionado anteriormente. A quantidade de dados, em *bytes*, a serem escritos pode ser definida pelo usuário em um intervalo variando de 1 até 256 *bytes*.

Os valores configuráveis em 8 *bits* que representam o comprimento em *bytes* do bloco a ser escrito seguem a lógica demonstrada no Quadro 12.

Comprimento do Bloco (em <i>bytes</i> )	Valor associado
$1 \leq \text{Comprimento} \leq 255$	Igual ao Comprimento
Comprimento = 256	0x00

Quadro 12- Valores configuráveis para escrita na memória FLASH

O diagrama da figura 21 exemplifica como é realizado o procedimento de escrita, entretanto é presumido que a inicialização da FPI tenha sido realizada antes do início do procedimento abaixo.

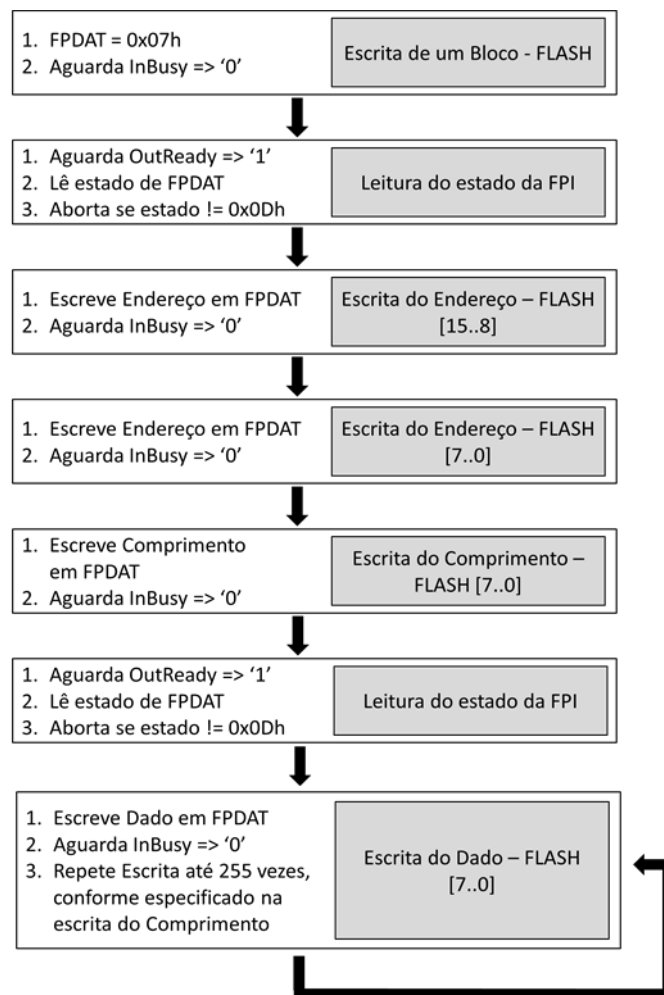
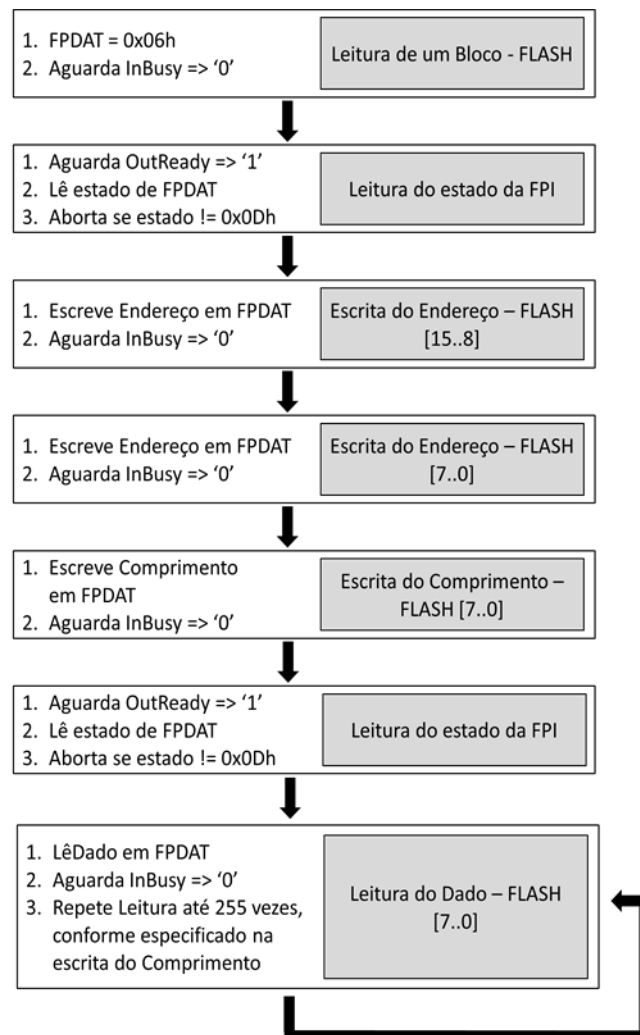


Figura 21 - Escrita de um bloco da memória FLASH  
Fonte: (Adaptado de Silicon Labs, 2011)

### 3.3.7.7 Leitura de um bloco da memória FLASH

Semelhante ao comando de escrita, a leitura de blocos da FLASH pode ser feita em números que também variam no intervalo de 1 a 256 *bytes*. Inclusive, os valores associados ao comprimento do bloco a ser lido seguem a lógica do quadro apresentado na seção anterior.

A diferença, entretanto, é que isto é feito através do comando FLASH *Block Read*, próprio para a leitura e que também parte do pressuposto que a inicialização da FPI tenha sido corretamente realizada.



**Figura 22 - Leitura de um bloco da memória FLASH**  
 Fonte: (Adaptado de Silicon Labs, 2011)

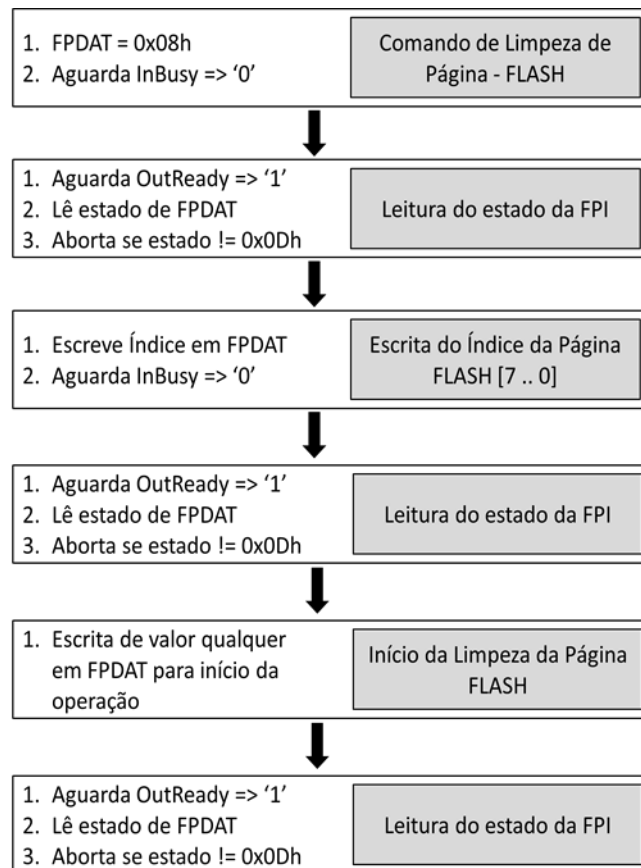
### 3.3.7.8 Limpeza de uma página da memória FLASH

A limpeza de uma página da memória FLASH (Figura 23) pode ser realizada, por meio do comando FLASH *Page Erase*, em blocos de 512 *bytes*. Cada página indexada se refere a um intervalo da memória conforme apresentado, para o caso de um dispositivo dotado de memória FLASH com capacidade de 64 *kbytes*, no Quadro 13.

Índice da Página	Intervalo correspondente na FLASH
0	0x0000 – 0x01FF
1	0x0200 – 0x03FF
2	0x0400 – 0x05FF
3	0x0600 – 0x07FF
4	0x0800 – 0x09FF
5	0x0A00 – 0x0BFF
6	0x0C00 – 0x0DFF
7	0x0E00 – 0x0FFF
8	0x1000 – 0x11FF
9	0x1200 – 0x13FF
10	0x1400 – 0x15FF
11	0x1600 – 0x17FF
12	0x1800 – 0x19FF
13	0x1A00 – 0x1BFF
14	0x1C00 – 0x1DFF
...	...
126 (Reservada)	FC00 – FDFF
127 (Reservada)	FE00 – FFFF

**Quadro 13 - Páginas da memória FLASH**  
**Fonte: (Adaptado de Silicon Labs, 2011)**

Sequência para apagar páginas da memória FLASH:



**Figura 23 - Limpeza de uma página da memória FLASH**  
 Fonte: (Adaptado de Silicon Labs, 2011)

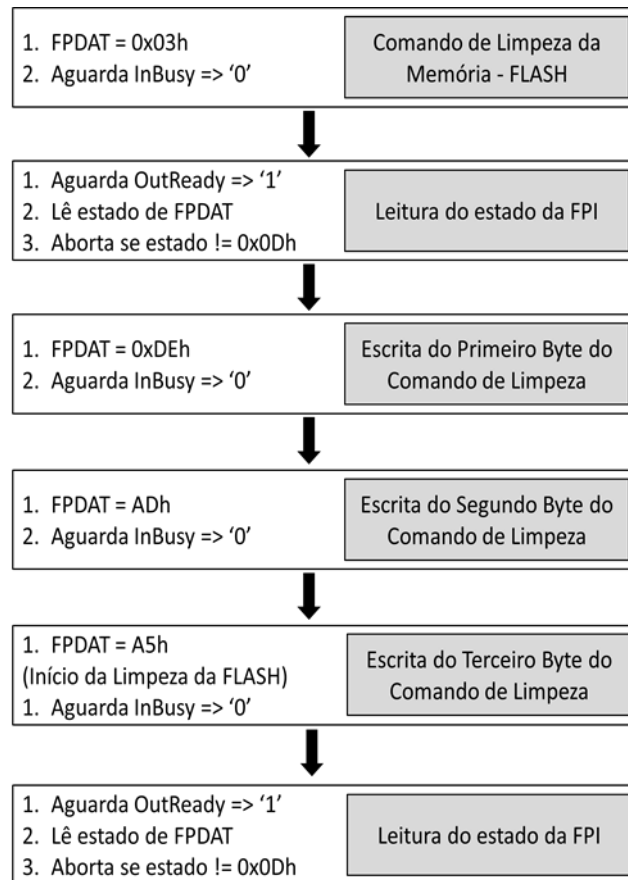
### 3.3.7.9 Limpeza da memória FLASH completa

Para apagar todo o conteúdo da memória FLASH do dispositivo a ser programado, por meio do comando FLASH *Device Erase*, uma série de 3bytes deve ser escrita na FPI. Sem fazê-lo, o comando não é habilitado e a limpeza não é iniciada.

Assim sendo, os *bytes* a serem enviados são a sequência:

- 0xDE;
- 0xAD;
- 0xA5.

Como em todos os casos anteriores, assume-se que a inicialização da FPI tenha sido corretamente realizada. Assim, o procedimento de deleção do conteúdo da memória é o descrito no diagrama da Figura 24:



**Figura 24 - Limpeza da memória FLASH**  
**Fonte: (Adaptado de Silicon Labs, 2011)**

### 3.3. Placas e Circuitos

O desenvolvimento de *hardware* é definitivamente uma das etapas mais críticas do projeto. Isto porque, apenas após a fabricação, montagem e primeira energização da placa é que se pode confirmar se o seu funcionamento está ou não como esperado.

Em função do fato de terem sido escolhidos dois *chips* diferentes para a placa alvo durante a etapa de definição do projeto, isto tendo em vista que há dois protocolos – JTAG e C2 – utilizados pelo fabricante para a realização da gravação e depuração de códigos, optou-se pela fabricação de apenas uma placa contendo distintas regiões para acomodar cada uma das pastilhas. Assim, foi possível deixar a decisão de escolha do protocolo para um momento posterior e condicioná-la à

experiência prática de desenvolvimento, significativamente mais flexível do que a alteração, eventualmente necessária, da placa de circuito impresso.

Para a elaboração do *layout* da placa - que contém em sua área os três núcleos microcontroladores 8051 - foi utilizado o *software Altium Designer*, famoso por ser um dos mais completos programas para a atividade de projetos de hardware. Ainda que contando com bibliotecas bastante sofisticadas, pelo fato de ter seu foco voltado a desenvolvimento de dispositivos de lógica reconfigurável, foi necessário empregar horas do projeto para o desenho de esquemáticos e *layouts* de vários dos componentes ativos de hardware, tais como os microcontroladores, *buffers* e transistores. Os esquemáticos referentes às placas definitivas para o gravador/depurador e placa alvo C8051F326 (C2) estão localizadas no anexo A.

Cada uma das quatro regiões distintas da placa, isoladas a partir linhas verticais ou horizontais de cor verde, representam uma sub-placa com funcionalidade diferenciada. Na figura 25, posicionada à esquerda na vertical, pode-se visualizar a placa com funcionalidade de gravador e depurador na qual será montado o controlador C8051F321. Ao centro na porção superior verifica-se a placa alvo equipada com o controlador C8051F018 (opção para comunicação JTAG) e abaixo desta a placa alvo com o controlador C8051F386 (opção para comunicação C2). Na região inferior à direita localiza-se um barramento com 8 *leds*, ali posicionada com propósito de testes e servir, eventualmente, como ferramenta de desenvolvimento associado a um dos *ports* de qualquer uma das placas alvo.

Uma prática comum dos desenvolvedores de placas é o posicionamento de extensos planos de terra (GND). Na figura 26 isto fica bastante evidente, e cumpre o propósito de reduzir, tanto quanto for possível, as interferências decorrentes de diversas linhas de sinalização que se encontram próximas umas das outras.

Concluída a etapa de projeto de *hardware* e realizada a conferência de todas as conexões, comparando-as com os manuais dos fabricantes de cada componente, foram gerados os arquivos de fabricação no *software Altium* chamados *Gerber Files* e *NC Drills Files*, que serviram de referência para a manufatura das placas de circuito impressa em fibra FR-4, conforme exibido nas figuras 27 e 28.



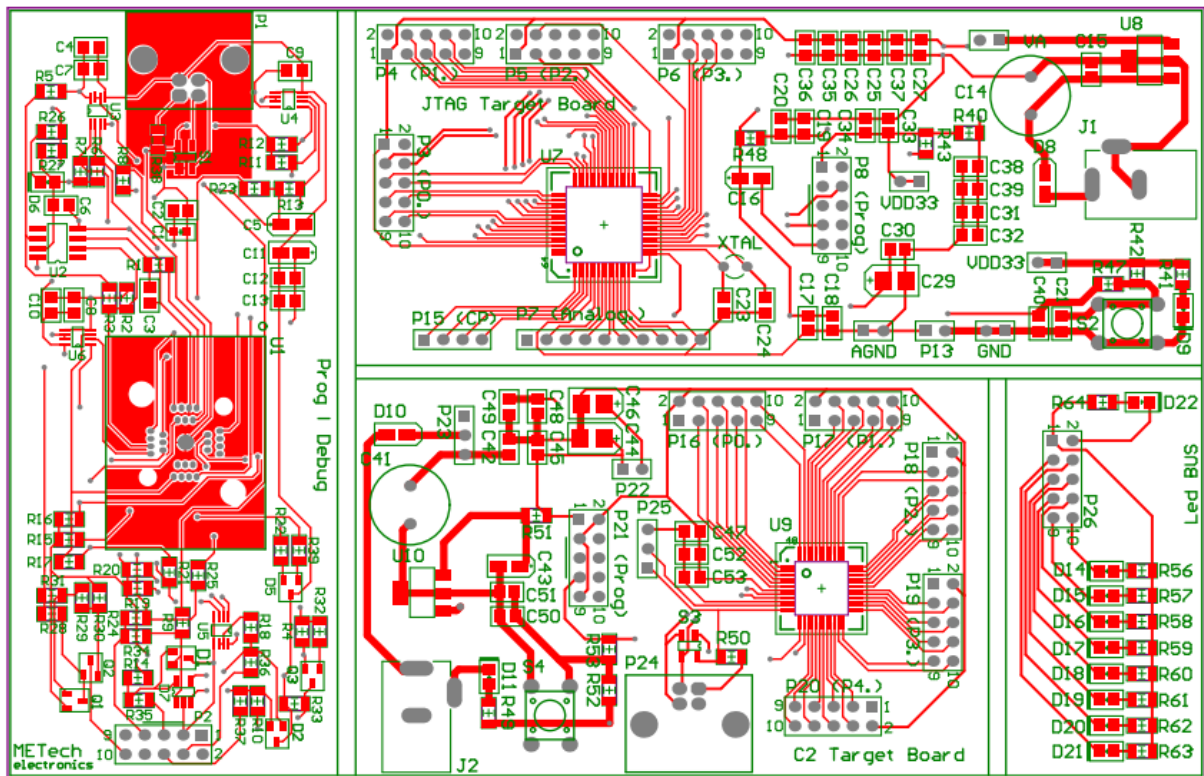


Figura 25 - Layout da face superior da placa PDETR

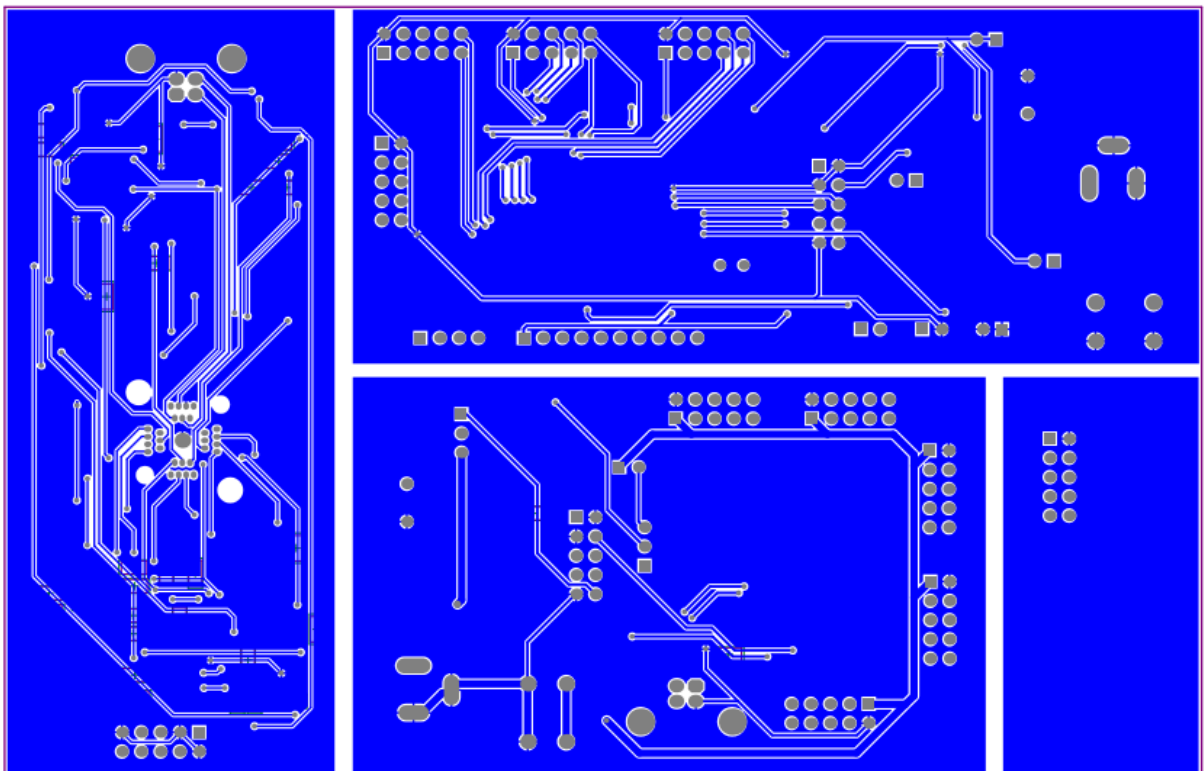


Figura 26 - Layout da face inferior da placa PDETR

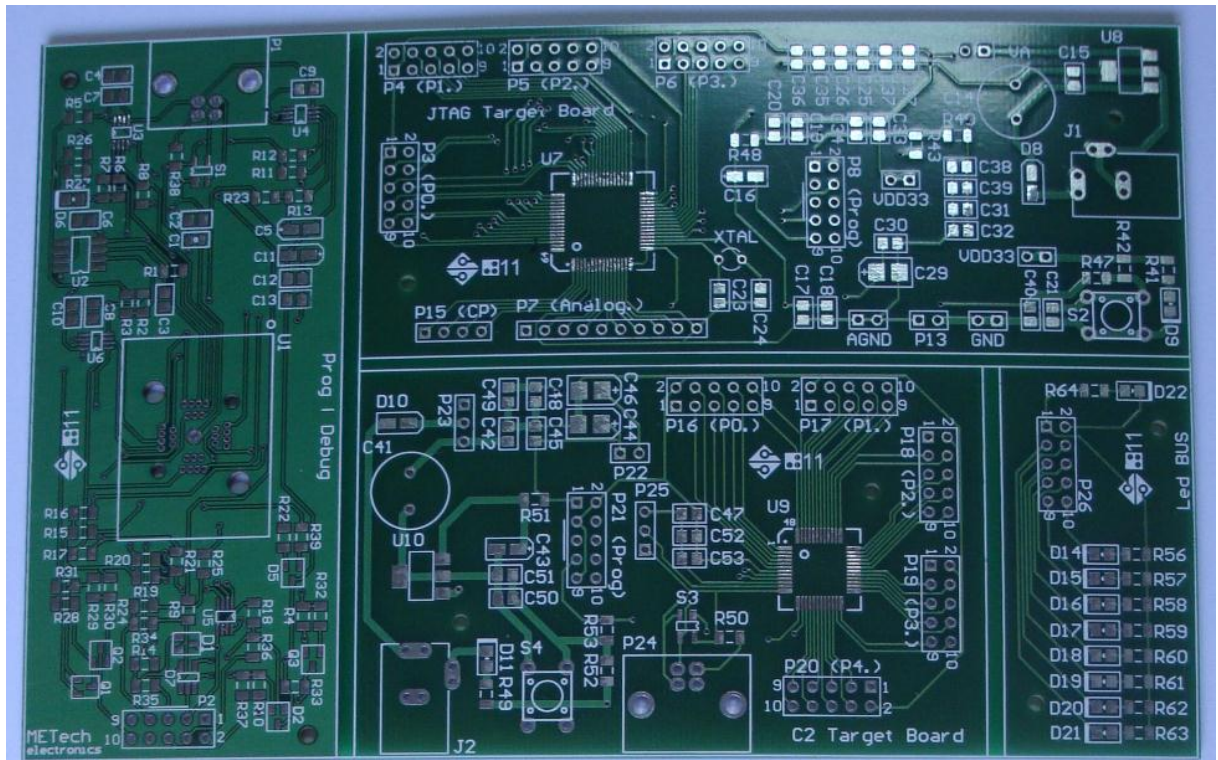


Figura 27 - Layout da face superior da placa PDETR fabricada

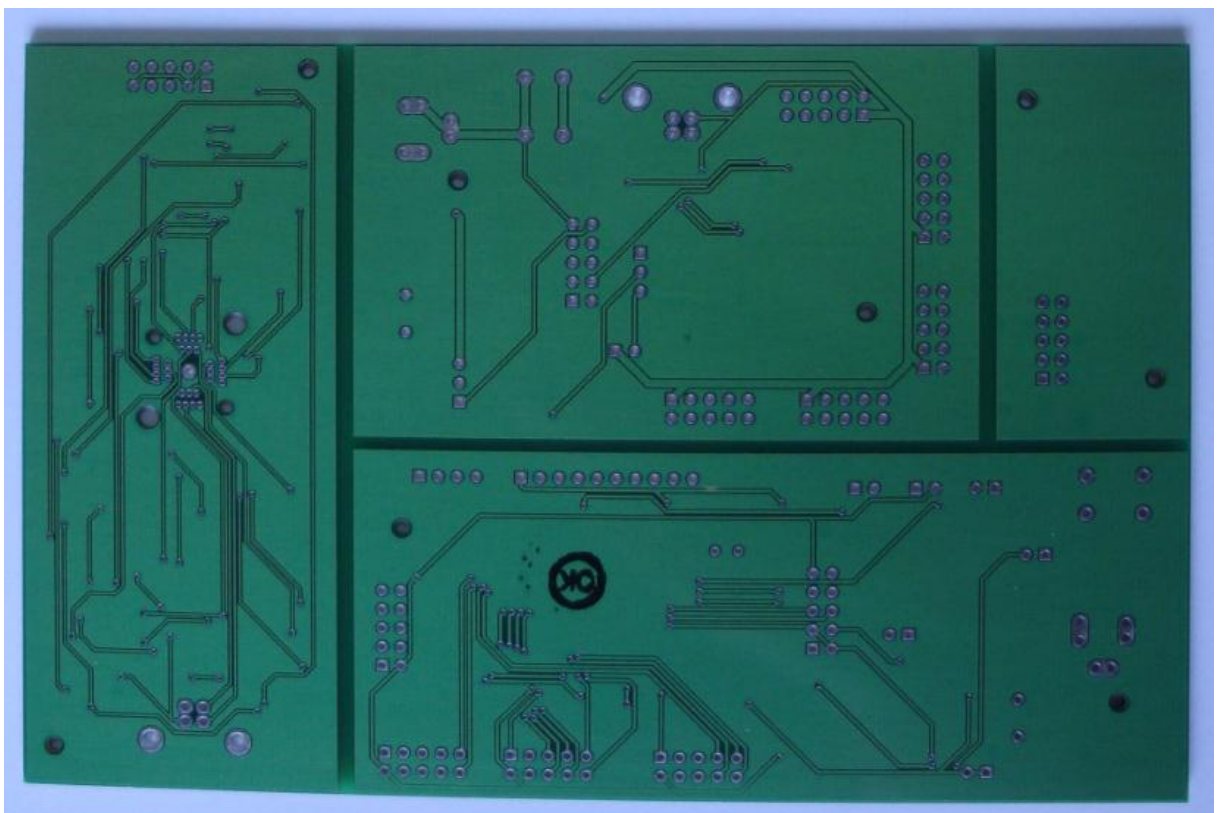


Figura 28 - Layout da face inferior da placa PDETR fabricada



## 4. TESTES E RESULTADOS

O principal resultado do projeto PDETR foi uma placa de circuito impresso dividida em três módulos: a placa do microcontrolador C8051F386 depurável via interface C2, a placa do microcontrolador C8051F018 depurável via interface JTAG e a placa do *Debug Adapter* (C8051F321). O presente capítulo apresenta os testes que validam o funcionamento e o êxito do projeto.

### 4.1. ALIMENTAÇÃO

O funcionamento do circuito de alimentação foi validado com o auxílio do osciloscópio, verificando as tensões de alimentação dos componentes, dos microcontrolados e as tensões nos pinos de  $V_{DD}$  da placa. Na figura 29, no canto inferior esquerdo, o LED D11 indica que a PDETR esta ligada.

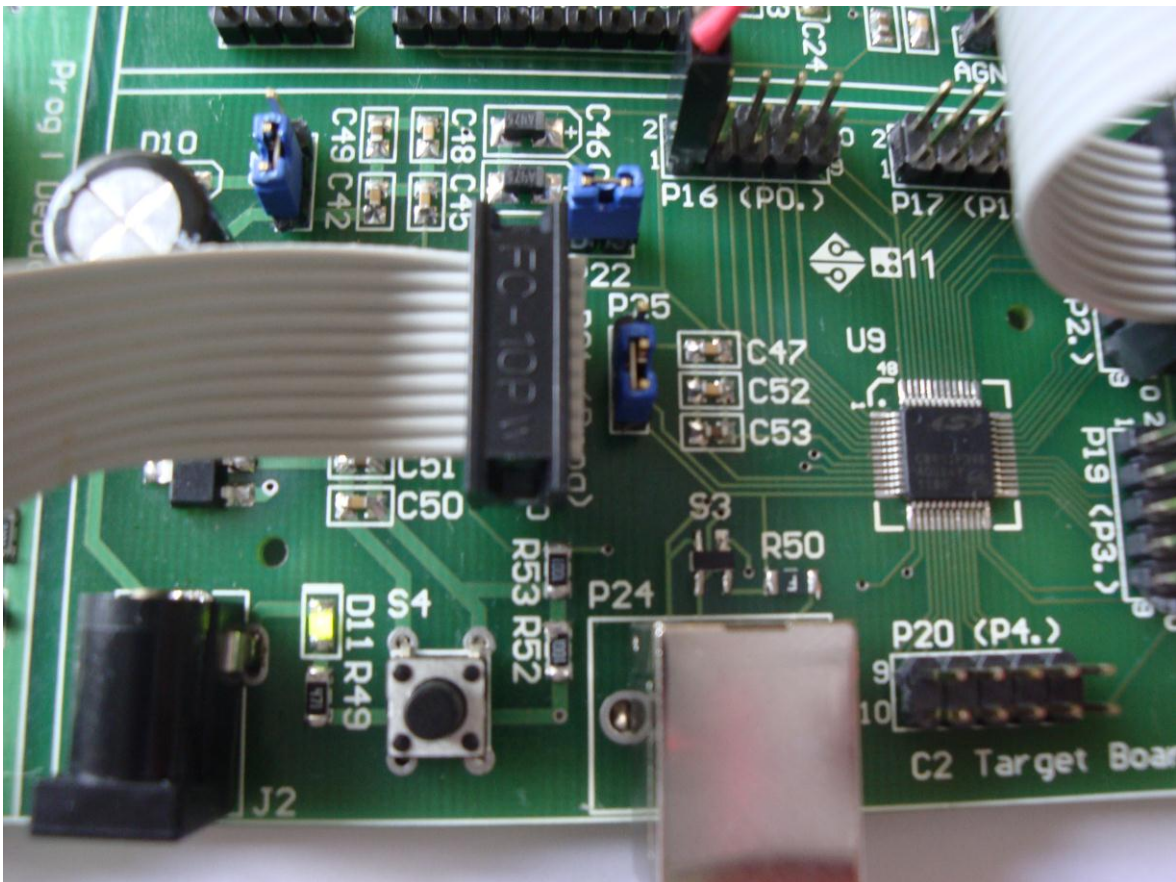


Figura 29 - Validação do circuito de alimentação

A tensão de saída foi projetada para ser 3.3V. A figura 30 mostra que a tensão de saída do pino 1.1 ( $V_{DD}$ ) está em 3.4 volts. Um erro de 3%, que pode ser tolerável para os fins didáticos para os quais a PDETR foi desenvolvida.

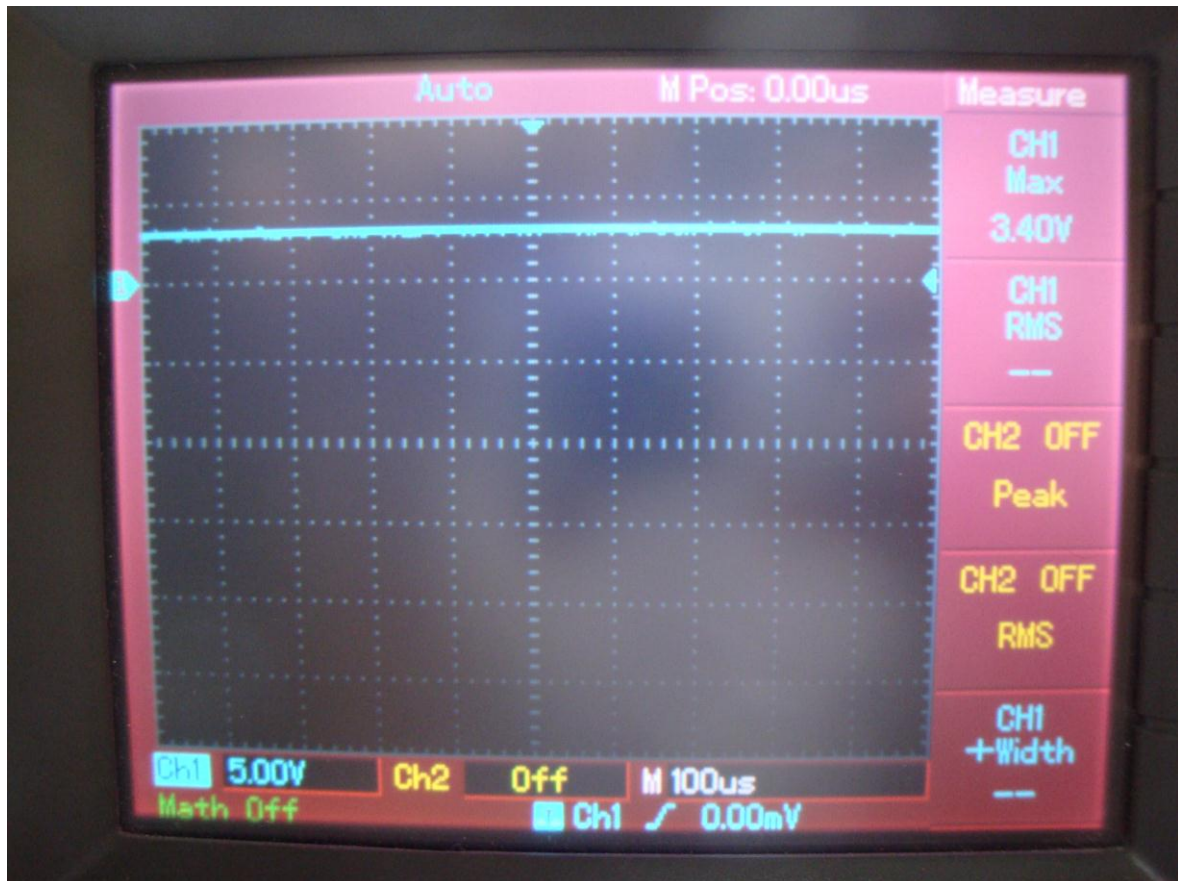


Figura 30 - Tensão de saída da placa

#### 4.2. OSCILADOR, PINOS DE ENTRADA E SAÍDA E LEDS.

Para testar o funcionamento do oscilador interno do microcontrolador, os *ports* e o barramento de *leds*, foi utilizado um código fonte com base em referências dos manuais do fabricante do C8051F386 e do C8051F321. O *timer2* foi ajustado para auto recarga com frequência de 5kHz. A figura 31 mostra o sinal gerado no *port1*. A frequência medida foi de 5.03kHz. O pequeno erro de frequência é justificado pelo fato de o código de teste ter sido desenvolvido em linguagem de programação C, o

que impossibilita um controle total de quantos ciclos de *clock* será executado por instrução.

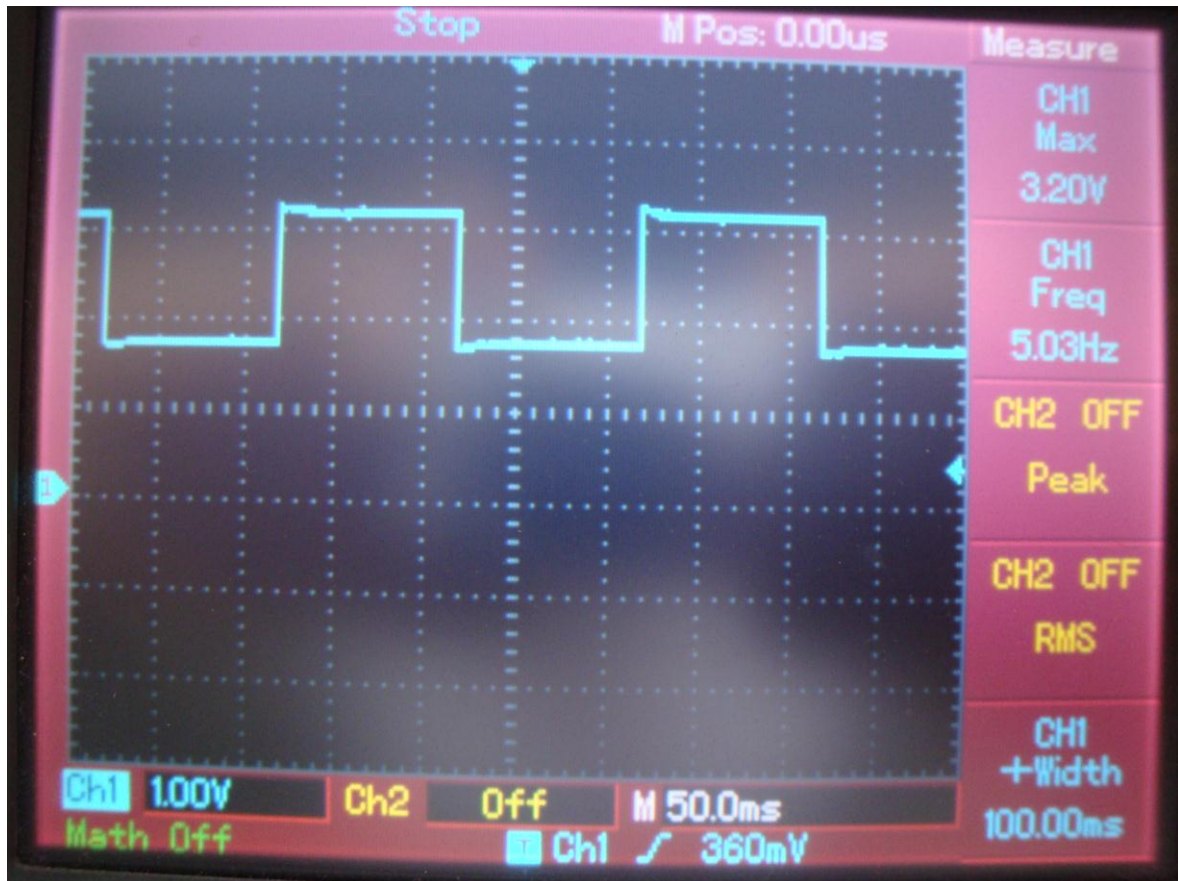


Figura 31 - Testes de clock e pinos de saída

O funcionamento dos pinos de saída e do barramento de *leds* pode ser verificado na figura 32. O port1 foi conectado ao barramento de *leds* e o LED D19, polarizado a cada “estouro” de *timer*, indica que os *ports* estão funcionando corretamente.



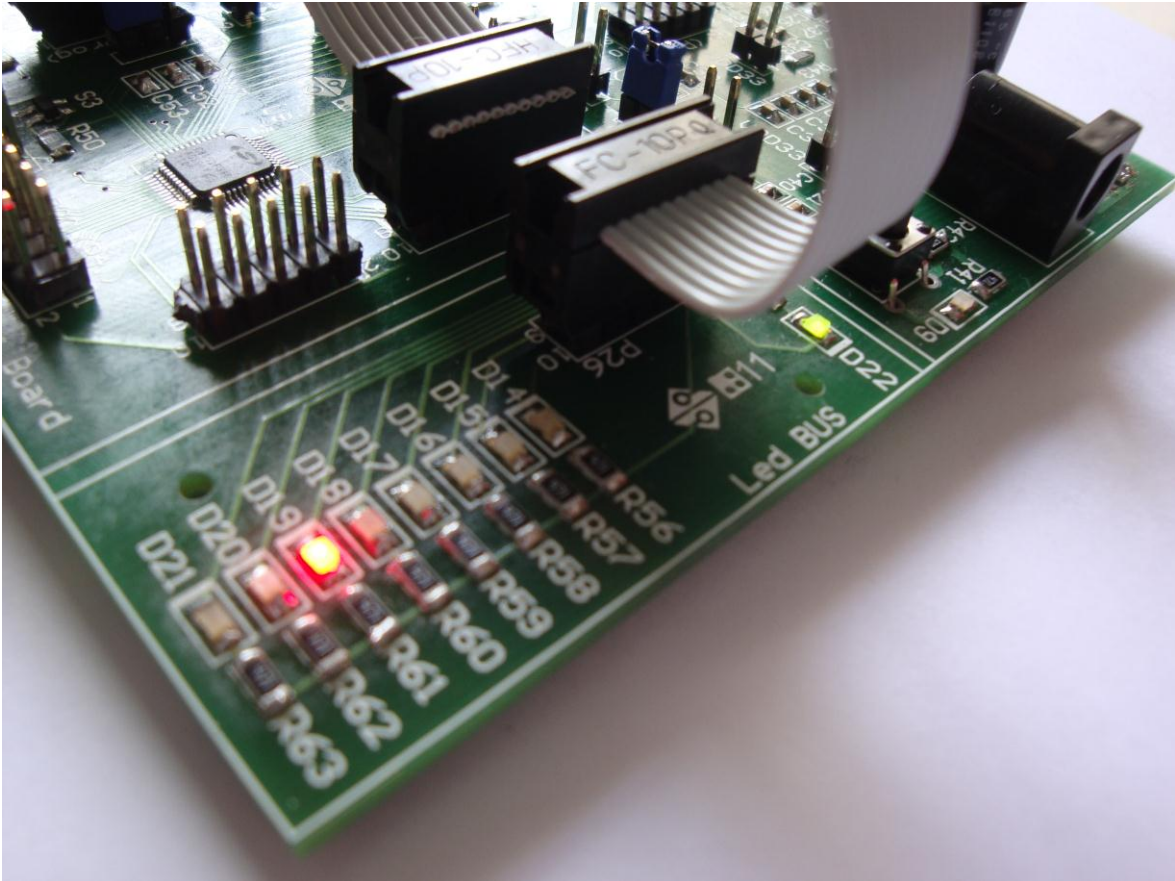


Figura 32 - Port1 conectado ao barramento de Leds

#### 4.3. DEBUG ADAPTER

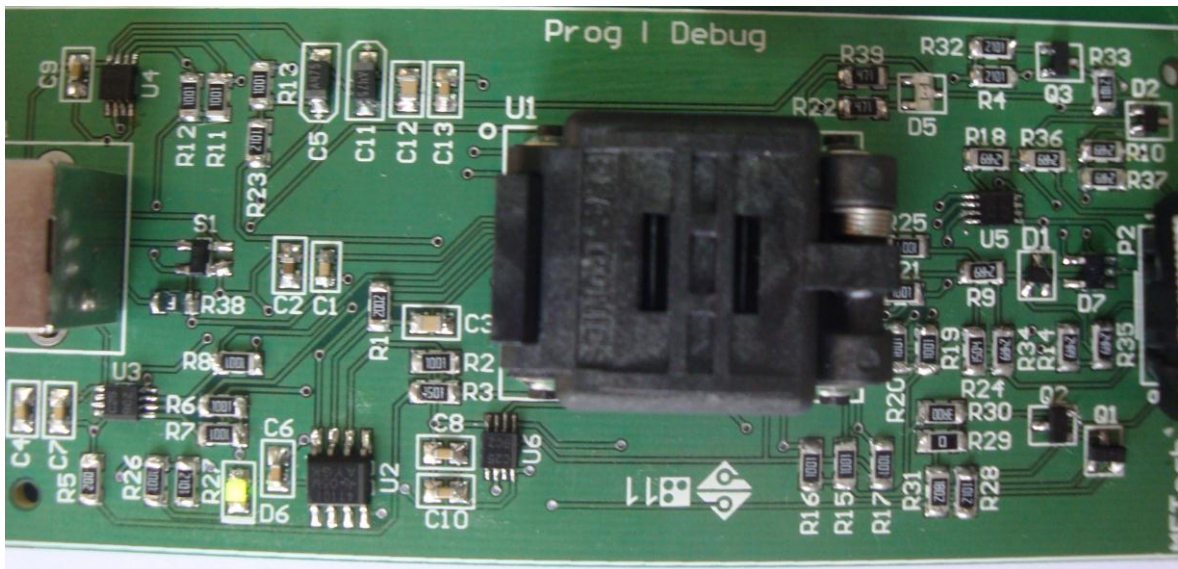


Figura 33 - Debug Adapter conectado ao PC

O funcionamento do *Debug Adapter* é validado pelo êxito dos testes anteriores. O download do código fonte, do teste do oscilador, do PC para o C8051F386 foi realizado pelo *Debug Adapter*. A figura 33 mostra o *Debug Adapter* alimentado, LED D6 polarizado, porém desconectado do ambiente de desenvolvimento. O LED D5 polarizado indica que o *Debug Adapter* está conectado ao ambiente de desenvolvimento (figura 34).

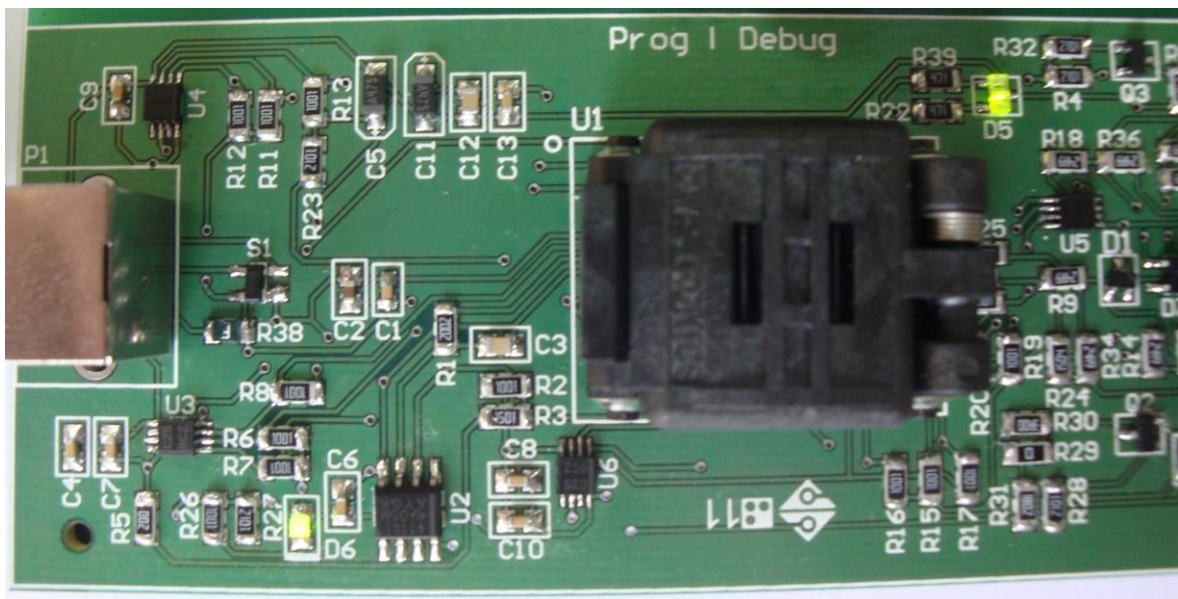


Figura 34 - Debug Adapter conectado (PC e ambiente de desenvolvimento)

#### 4.4. TESTE DO SOFTWARE C2

O protocolo C2 utilizado para comunicação do C8051F321, *Debug Adapter*, com o C8051F386, placa *target*, foi implementado em linguagem c com auxílio das notas de aplicação disponibilizadas pela *Silicon Labs*. O código foi validado escrevendo comandos na FLASH do microcontrolador C8051F386 através de comandos C2 enviados do C8051F321. Com auxílio do IDE da *Silicon Labs*, o código transferido para o C8051F386 foi extraído de sua memória FLASH e comparado com o código original. O resultado foi que o código extraído era 100% igual ao código enviado, comprovando o funcionamento da implementação do protocolo C2 para gravação de memória FLASH.

#### 4.5. SUGESTÕES PARA VERSÕES FUTURAS

- Retirar a placa com núcleo controlador 8051 de protocolo JTAG, microcontrolador C8051F018, do *layout* definitivo da PDETR.
- Realinhar horizontalmente a placa do *Debug Adapter*.
- Tornar mais informativo o *layout* da serigrafia da placa, nomeando, por exemplo, os pinos referentes à  $V_{DD}$  e GND.
- Acrescentar *Push Buttons* configuráveis para aplicações do usuário.
- Estudar alternativas de IDE com protocolo aberto para implementação de *firmware* da depuração em tempo real.
- Simplificar o *Hardware* do *Debug Adapter*.

#### 4.6. CONSIDERAÇÕES

Os testes apontaram o perfeito funcionamento da Placa PDETR. Vale ressaltar que os testes da placa do microcontrolador C8051F018 não foram apresentados, pois o durante o desenvolvimento do projeto optou-se por utilizar o protocolo C2.



## 5. CONCLUSÃO TÉCNICA

Desenvolver uma plataforma didática microcontrolada com arquitetura de 8 *bits*, para o auxílio na disciplina de microcontroladores da UTFPR, foi o objetivo principal do projeto PDETR. Tecnicamente especificou-se que o resultado deveria contemplar uma placa programável em *assembly*, com interface USB para transferência de código e possibilidade de depuração em tempo real.

Antes de iniciar o desenvolvimento da plataforma, uma pesquisa sobre microcontroladores que possuíssem as características desejáveis foi realizada. O resultado mostrou que existe um número restrito de dispositivos 8051 com interface de depuração como esperado. Verificou-se que dois fabricantes possuíam dispositivos que atendiam a necessidade do projeto: *Infineon* e *Silicon Labs*. A família C8051F3XX da *Silicon Labs* foi a selecionada, em função da gama de opções de microcontroladores, periféricos agregados, documentação existente e disponibilizada, além de oferecer, adicionalmente, um protocolo próprio – chamado C2 - de depuração em tempo real.

Com relação ao desenvolvimento, uma mudança de escopo de projeto foi realizada, o protocolo JTAG para gravação e depuração em tempo real de código fonte foi substituído pelo protocolo C2 da *Silicon Labs*. Esta decisão foi tomada tendo como base a pesquisa inicial que apontou que os microcontroladores com JTAG custam em média 150% a mais do que os dotados de interface de depuração com o protocolo C2. Adicionalmente, durante a fase de pesquisa foi observado, tanto no site do fabricante quanto em páginas de distribuidores, a existência de um *chip* exatamente igual ao do dispositivo de gravação/depuração comercializado com o *firmware* gravado de fábrica.

Após esgotar todos os contatos comerciais disponíveis, nacional e internacionalmente, o resultado foi negativo, com todas as partes negando o pedido de comercialização do componente. Se a aquisição deste chip fosse concretizada e viabilizada em escala para suprir a demanda existente na universidade, o projeto teria se resumido a uma carga muito inferior de trabalho e alocação de recursos e cumpriria, ainda assim, os objetivos de projeto, visto que este protocolo alternativo possui a mesma capacidade do protocolo JTAG, além de possibilitar uma economia

de pinos de I/O e de ser financeiramente mais viável. Sua única desvantagem decorre do fato de ser um protocolo exclusivo do fabricante.

Frente a essa situação, com todo o hardware desenvolvido e preparado para receber esse chip gravado de fábrica, surgiu o desafio de desenvolver completamente o protocolo de gravação/depuração. Tal objetivo não pode ser alcançado principalmente pelo fato de o trecho de comunicação entre a IDE e o controlador C8051F321 não dispor de documentação aberta, protegendo, desta maneira, a propriedade intelectual da marca nesta tecnologia. A alternativa para validar o *hardware* desenvolvido foi efetuar a retirada de uma pastilha gravada do *Debug Adapter* do fabricante, fato que se concretizou com sucesso.

Mesmo com mudanças no escopo, existe viabilidade na solução encontrada, uma vez que os avanços foram consideráveis e, tendo em vista que a exploração dessas tecnologias para a arquitetura 8051 está sendo pioneira no âmbito da UTFPR. Sendo assim, os resultados atingidos são expressivos mesmo não tendo sido plenos em todas as frentes propostas inicialmente.

Além disso, a documentação e consolidação do conhecimento ficam como legado e ponto de partida para futuras melhorias e desenvolvimento de uma nova versão.

## 6. PLANO DE NEGÓCIOS

Este plano de negócios tem como objetivo a análise da viabilidade mercadológica e financeira do negócio que compreende a implantação da Plataforma Digital Emulada em Tempo Real, PDETR, no mercado de *kits* didáticos. Esse estudo será fundamentado baseando-se em análises de mercado, projeções de vendas e previsões de gastos. Com isso foi possível estimar o desempenho econômico desse negócio, o qual se mostrou lucrativo a partir do terceiro ano de operações.

### 6.1. SUMÁRIO EXECUTIVO

#### 6.1.1. Oportunidade de negócio

No momento da disputa por uma colocação no mercado de trabalho, cada vez mais competitivo e exigente, os alunos de engenharia de formação mais teórica se encontram em desvantagem quando comparados aos que durante os anos de graduação tiveram, além da teoria, maior contato com a prática.

Observando ainda, a crescente demanda por profissionais técnicos cada vez mais capacitados para o desenvolvimento de novas soluções tecnológicas nos leva a conclusão que este perfil de formação é uma variável chave para a empregabilidade após a graduação, sobretudo nas oportunidades de emprego iniciais da carreira profissional.

Em meio a este cenário, nota-se a carência de plataformas didáticas adequadas disponíveis para a imersão dos acadêmicos nesse universo. Considerando a escassez de meios facilitados para a aquisição de recursos, além do alto custo e a burocracia incorrida na importação de plataformas didáticas, foi identificada a oportunidade de atuar neste mercado.

#### 6.1.2. O produto

A Plataforma Didática Emulada em Tempo Real – PDETR pretende oferecer uma solução baseada na arquitetura de microcontroladores 8051 para os primeiros passos de alunos que ainda não possuem experiência com estes dispositivos,

disponibilizando um recurso de fácil acesso, com preço convidativo, dotado de recursos tecnológicos alinhados com as demandas do mercado e acompanhado de um ambiente de desenvolvimento de fácil adaptação.

Através da escolha dos microcontroladores 8051, famosos pela simplicidade dentre as arquiteturas de 8 *bits*, a curva de aprendizado é acelerada e permite aos acadêmicos uma transição mais suave, em um momento posterior, para microcontroladores e microprocessadores de maior complexidade. Tal facilidade não é apenas desejável, mas fundamental nas etapas de ambientação dos alunos a novas tecnologias, potencializando a absorção de conhecimentos e contribuindo para a formação tecnológica com foco na prática.

### 6.1.3. A Empresa

A METech foi fundada em 2011 pelos alunos do curso de Engenharia Elétrica da UTFPR, Mateus, Erikson e Thiago e tem como objetivo desenvolver *kits* didáticos para o aprendizado, pesquisa e desenvolvimento de microcontroladores.

Recentemente identificamos uma nova carência de tecnologia em *kits* de desenvolvimento na Universidade Tecnológica Federal do Paraná, e com isso apresentamos o atual plano de negócios para a PDETR, Plataforma Digital Emulada em Tempo Real, a qual vai ao encontro das reais necessidades dos alunos e utiliza o que há de mais novo em tecnologia e inovação disponíveis no mercado.

### 6.1.4. Visão 2014

Até o final de 2014, ser referência no mercado de plataformas didáticas microcontroladas e fazer com que nossos produtos cheguem a todas as instituições de ensino do Brasil, garantindo, desta maneira, uma formação mais completa aos acadêmicos.

### 6.1.5. Missão

Fornecer um produto com tecnologia de ponta, de fácil acesso e utilização para os alunos de engenharia.

#### 6.1.6. Valores

- Qualidade – Nossos produtos visam à solução de propostas requeridas pelos seus usuários, e por isso, a qualidade é fundamental.
- Inovação – Em nossos produtos, a matéria prima e tecnologias empregadas são de ponta, com isso garantimos a total integração do usuário com o que há de melhor no mercado.
- Comprometimento – Nossa equipe trabalha com prazos, objetivos e metas. Tudo para garantir a solução das necessidades dos clientes no tempo em que eles necessitam.

#### 6.1.7. Descrição do negócio

O produto PDETR é inserido no mercado com o intuito de ocupar uma lacuna existente, que consiste na carência de plataformas didáticas atualizadas amigáveis para usuários iniciantes. Com isso, o objetivo é que seja associada ao produto a ideia de que este é definitivamente o melhor recurso para as pessoas que desejam adquirir os conhecimentos pertinentes à eletrônica microcontrolada e não possuem nenhuma experiência no assunto. As principais vantagens associadas serão o preço reduzido e a disponibilidade no mercado nacional, possibilitando, assim, fácil acesso ao produto.

Atualmente o desenvolvimento visa à portabilidade a ser alcançada por meio da utilização do *kit* didático em qualquer computador com interface USB, possibilitando a programação do *hardware* e depuração de códigos em tempo real, sendo essa uma funcionalidade destacada em relação às alternativas disponíveis no mercado.

Atingiremos nossos clientes finais com uma abordagem direcionada, inicialmente, ao nicho composto por estudantes de engenharia de universidades públicas, mais especificamente o sub-segmento de alunos de engenharia eletrônica. Com isso buscamos uma expansão das vendas, para futuramente atender a outras instituições de ensino técnico, superior e de especialização, centros de tecnologia e empresas prestadoras de serviços.

## 6.2. PRODUTOS E SERVIÇOS

### 6.2.1. Características

A concepção da plataforma PDETR foi direcionada para a solução de um problema existente no âmbito da UTFPR – Campus Curitiba. Atualmente, nas disciplinas de microcontroladores lecionadas pelo Departamento Acadêmico de Eletrônica - DAELN, o *kit* didático P51 adotado como padrão para o acompanhamento e desenvolvimento das aulas das práticas possui limitações e tecnologia antiga.

Alguns dos problemas detectados no recurso em uso são:

- Necessidade de computador com porta serial (RS-232) para a utilização do *kit* didático, ou seja, o que limita o uso em laptops e, conseqüentemente, reduz a praticidade;
- Limitações para a depuração de códigos em tempo real em função de o microcontrolador utilizado no referido *kit* não ter originalmente tal funcionalidade disponível;
- Escassez de periféricos intrínsecos, como, por exemplo, interface SPI (*Serial Peripheral Interface*), conversores Analógico-Digital e Digital-Analógico, interface USB (*Universal Serial Bus*), etc.

Em vista de tais problemas, o produto proposto pretende atuar na melhoria dos seguintes pontos: substituindo a interface de uso do *kit* de RS-232 para USB, oferecendo um ambiente de desenvolvimento completo, porém de fácil utilização, equipando a nova placa com um microcontrolador 8051 mais sofisticado e com maior número de periféricos internos, de forma a permitir que o aluno desenvolva experimentos completos utilizando apenas o *kit*.

### 6.2.2. Produtos Similares

A PDETR encontrará, no mercado de componentes e *kits* eletrônicos, outras plataformas digitais semelhantes a ela. Algumas diferenças com relação aos

componentes, tais como, modelos de microcontroladores, capacidade de processamento, memórias, periféricos, etc., refletem diretamente no desempenho, na aplicabilidade e facilidade de uso do *kit*.

Atualmente, identificamos como principais concorrentes os *kits* MCB950, da fabricante e fornecedora de ambientes de desenvolvimento KEIL,8051F340 da *Silicon Labs* e a P51 desenvolvida pela UTFPR e atualizada pela última vez em 2006.



**Figura 35- C8051F340DK**  
Fonte:(Silicon Labs, 2011)



**Figura 36 - MCB950**  
Fonte: ( KEIL, 2011)

Tais *kits* são fabricados fora do país, com exceção do *kit* P51, por grandes empresas do ramo de eletrônica, e importados por distribuidores especializados. O principal objetivo da METech, para atingir o mercado brasileiro, é trabalhar com um preço reduzido, atuando no mercado de varejo e oferecendo uma solução integrada ao recomendar também o ambiente de desenvolvimento e documentação completa da solução voltada aos usuários iniciantes.

### 6.2.3. Análise Comparativa

	<b>Preço</b>	<b>Gravador</b>	<b>Conexão (PC)</b>	<b>Depuração</b>
PDETR	R\$ 189,00	JTAG Integrado	USB	Tempo Real
P51	R\$ 80,00	Não tem	RS-232	Simula Tempo Real
C8051F340DK	R\$ 200	Debug Adapter	USB	Tempo Real
MCB950	US\$ 129,00	Não tem	USB	Tempo Real

**Quadro 14 - Análise comparativa**

Como pode ser visto no quadro 14, os grandes diferenciais da PDETR em relação aos principais concorrentes são o seu preço e o gravador integrado a placa, podendo ser conectado ao PC via USB.

### 6.2.4. Análise SWOT

#### Pontos Fortes

- Preço acessível
- Conectividade via interface USB
- Gravador/Depurador embutido na placa
- Fácil utilização

#### Pontos Fracos

- Necessidade de importar a grande maioria dos componentes
- Montagem e Reposição de componentes complexa

#### Oportunidades Externas

- Diferencial tecnológico frente aos concorrentes
- Mercado consumidor potencial em crescimento

#### Ameaças Externas

- Aumento da cotação do dólar



- Entrada de empresas renomadas para disputa do nicho

#### 6.2.5. Produtos e serviços futuros

A PDETR acompanhará o desenvolvimento do mercado de componentes e as tendências da indústria, visando sempre estar atualizada com o que há de mais novo e melhor no segmento. Outras plataformas, semelhantes a esta, porém com propósitos e configurações diferentes serão desenvolvidas no futuro. *Kits* com microcontroladores de 16 *bits* e 32 *bits*, por exemplo. Pretendemos com isso, estabelecer uma ampla família de produtos, sempre proporcionando a solução ideal para o consumidor.

#### 6.3. Análise de Mercado

A análise de mercado tem a importante função de possibilitar o conhecimento do contexto atual, em meio ao qual o produto será lançado. Com base nas informações geradas será possível elaborar um parecer acerca do potencial do produto, de potenciais clientes, de fornecedores capazes de suprir a lista de componentes necessários para o desenvolvimento, de prováveis concorrentes que competirão pela mesma parcela do mercado e por patentes, direitos autorais ou marcas já existentes. Ainda, ao confrontar esses fatores pode-se avaliar a viabilidade do projeto.

##### 6.3.1. Importação de microcontroladores no Brasil

Mais importante do que elaborar o projeto de um produto a ser desenvolvido, é ter em mãos ferramentas ou dados que possibilitem avaliar o potencial de aceitação para tal produto no mercado destinado. Neste contexto, foi realizada uma pesquisa no portal Mercosur Online– uma página de acesso restrito a assinantes que disponibiliza informações das aduanas de países da América Latina em suas negociações de importação e exportação realizadas com países de todas as regiões do globo – para que, com base em dados concretos, pudesse ser realizada uma análise real da situação do mercado de microcontroladores no Brasil.

As pesquisas são realizadas a partir do código NCM (Nomenclatura Comum do Mercosul), o qual é dado em função de uma sequência de classificações que descrevem o produto em questão. Neste caso, especificamente, utilizou-se o código NCM 854231 - PROCESSADORES E CONTROLADORES, COMBINACIONAIS COM MEMÓRIAS, CONVERSORES, ETC.; que descreve de forma mais aproximada os circuitos microcontroladores em questão no projeto para a extração de dados do portal no dia 23 de maio de 2011.

Dependendo do país utilizado como parâmetro para pesquisa na base de dados, diferentes tipos de informações podem estar disponíveis. A aduana brasileira, por exemplo, disponibiliza informações como: Volume de Importações (em US\$), Quantidades Importadas (Unidades), País de Origem, Local de desembarque no Brasil, etc. para um intervalo de tempo que vai desde 2007 até o primeiro trimestre de 2011.

Primeiramente são apresentadas as quantidades (número de unidades) e valores (em dólares) para as importações de controladores e processadores realizadas apenas no primeiro trimestre do ano atual. Este quadro elenca os dez estados brasileiros que mais investiram recursos realizando importações dos referidos componentes, e é seguido de um gráfico que clarifica a representatividade de cada estado importador em relação ao país como um todo.

### Brasil – Importação

(1-Jan-2011 até 31-Mar-2011) NCM: 854231

Componentes – Microcontroladores e Microprocessadores eletrônicos

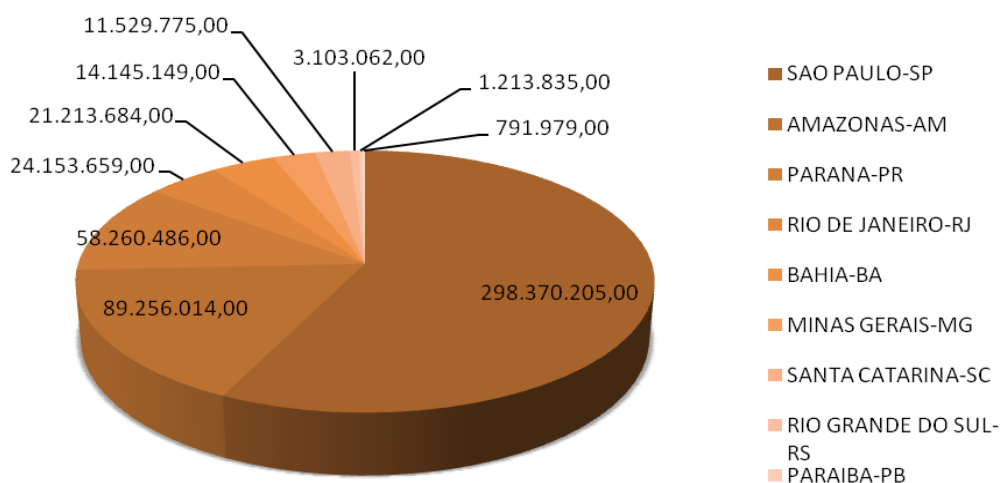
Estado Importador	Unidades	Valor FOB US\$	Valor FOB US\$
São Paulo-SP	75,962,282.00	298,370,205.00	57.03%
Amazonas-AM	45,173,549.00	89,256,014.00	17.06%
Paraná-PR	32,266,980.00	58,260,486.00	11.14%
Rio de Janeiro-RJ	296,287.00	24,153,659.00	4.62%
Bahia-BA	695,158.00	21,213,684.00	4.05%
Minas Gerais-MG	8,784,528.00	14,145,149.00	2.70%
Santa Catarina-SC	3,969,836.00	11,529,775.00	2.20%

Rio Grande do Sul-RS		2,203,737.00	3,103,062.00	0.59%
Paraíba-PB		19,073.00	1,213,835.00	0.23%
Pernambuco-PE		647,007.00	791,979.00	0.15%
Total TOP 10	(Absoluto)	170,018,437.00	522,037,848.00	99.78%
Total	(Absoluto)	170,793,125.00	523,182,402.00	100.00%
Total TOP 10	(%)	99.55%	99.78%	99.78%

**Quadro 15 - Importações por estado brasileiro - 10 maiores**  
**Fonte:(Mercosuronline,2011)**

Fica evidente que apenas dez estados da federação concentram praticamente 100% das importações realizadas por todo o país. O estado do Paraná, de interesse no âmbito do produto que está sendo desenvolvido, é listado na terceira posição entre os maiores importadores de controladores e processadores. Isto induz à conclusão de que existe um mercado representativo neste estado, cuja atividade envolve o uso destes elementos eletrônicos e, portanto, há uma tendência de haver maior demanda por profissionais que estejam capacitados a trabalhar com os mesmos.

### Valor FOB US\$ por Estado – TOP10 - Jan-Mar/2011



**Gráfico 1 – Importação de Microcontroladores e Microprocessadores - Totais em USD**  
**Fonte:(MERCOSURONLINE, 2011)**

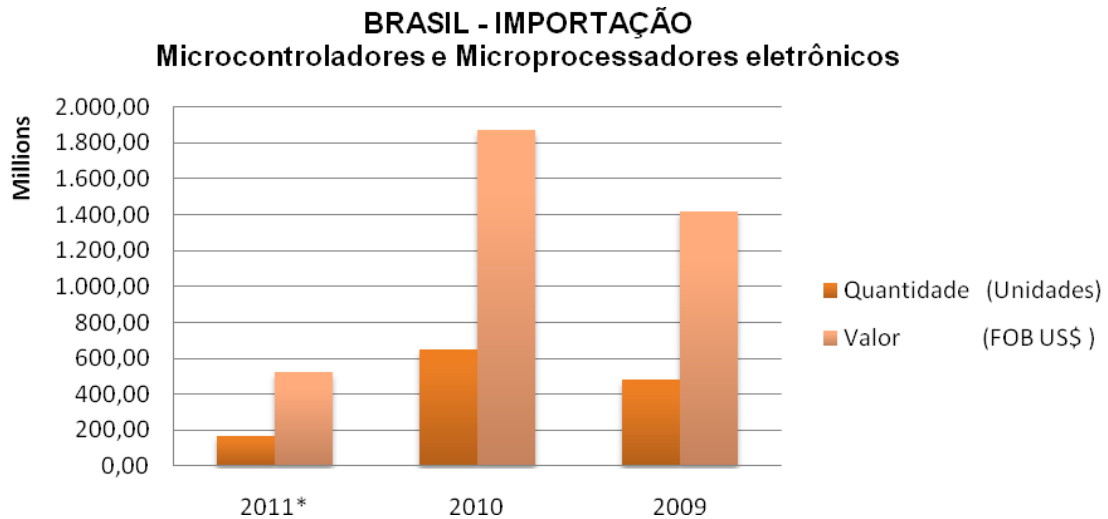
Logo abaixo, é disponibilizado um quadro que revela a evolução das quantidades e valores importados desde o ano de 2009 até o primeiro trimestre do ano atual. Estes dados são referentes ao Brasil e não discriminam quais os valores e quantidades para cada estado individualmente.

Brasil – Importação			
(1-Jan-2009 até 31-Mar-2011) NCM: 854231			
Componentes – Microcontroladores e Microprocessadores eletrônicos			
Ano	2011 (Janeiro a Março)	2010	2009
Quantidade (Unidades)	170,793,125.00	650,955,157.00	481,222,444.00
Valor (FOB US\$)	523,182,402.00	1,874,891,191.00	1,417,545,681.00

**Quadro 16 - Importações no Brasil – Consolidado**  
**Fonte:(MERCOSURONLINE, 2011)**

Nota-se que tanto a quantidade importada bem como o valor despedido para realizar tal atividade têm evoluído a cada ano, sendo possível projetar um crescimento de 200 milhões de unidades ao final de 2011 em relação a 2009, caso a regularidade do primeiro trimestre seja mantida nos restantes. Isto representa uma

expansão de aproximadamente 30% na quantidade de componentes importados em um curtíssimo período de tempo.



**Gráfico 2 - Importação de Microcontroladores e Microprocessadores**  
**Fonte: (MERCOSURONLINE, 2011)**

### 6.3.2. Segmentação de Mercado

A estratégia adotada para segmentar o mercado é fundamentada no modelo de preferências conglomeradas, que compreende o mercado como um universo no qual há vários agrupamentos definidos a partir das preferências dos consumidores que o integram.

Sendo assim, em meio ao universo de clientes que potencialmente adquirem plataformas didáticas eletrônicas, os dois sub-segmentos que temos o potencial para atingir de forma mais abrangente são definidos abaixo, a partir de variáveis demográficas de renda, educação e ocupação e também por variáveis comportamentais de ocasião e benefícios.

Segmento	Renda	Educação	Ocupação	Ocasão	Benefícios
A	< R\$ 1.000	Eng. Eletrônica – Universidade Pública	Estudante e Estagiário	Especial - Matriculado em disciplina de microcontroladores ou em disciplinas que exijam projetos de aprovação	Produto Completo e Preço Acessível
B	entre R\$ 1.000 e R\$ 2.000	Eng. Elétrica, Eng. de Controle e Automação, Eng. de Computação, Eng. Mecatrônica, etc.	Estudante e Estagiário	Especial – Em busca de recursos para desenvolver trabalho de graduação ou disciplinas que exijam trabalho de aprovação	Produto de Qualidade, Fácil manuseio e Rápida adaptação
C	> R\$ 2.000	Engenharias relacionadas ou Cursos técnicos	Estudante, Pesquisador ou Hobbista	Normais – Para desenvolvimento de projetos particulares ou capacitação	Preço Acessível, Produto de Qualidade e Fácil Manuseio

**Quadro 17 - Segmentação de clientes potenciais**

Baseado na segmentação mostrada acima, fomos à busca dos números que nos mostrassem a realidade da educação superior no Brasil. De acordo com o MEC, Inep e o Censo da Educação Superior, o número de formandos em Engenharia caiu quando comparados os anos de 2000 e 2008.

Áreas	2000	2008
Total	100,0%	100,0%
Educação	25,9%	21,1%
Humanidades e Artes	3,2%	3,6%
Ciências Sociais e Direito	26,6%	27,3%
Economia e Administração	13,2%	13,7%
Ciências e Matemática	6,2%	5,9%
Ciência da computação	2,0%	1,8%
Engenharia	5,6%	5,1%
Arquitetura e urbanismo	1,2%	0,8%
Agricultura e Veterinária	2,1%	2,0%
Saúde e Bem-Estar Social	13,0%	16,0%
Serviços	1,0%	2,6%
Total absoluto de egressos	352.305	800.318

**Quadro 18 - Perfil dos Egressos na Educação Superior Brasil: 2000 e 2008**

Fonte: (MEC, Inep, Censo da Educação Superior, 2008)

Obviamente que, dos 5,1% do total de egressos em 2008, uma parcela bem menor será de nosso interesse, esta correspondente a formandos nas áreas de eletricidade, eletrônica e automação. Ainda de acordo com MEC, Inep e o Censo de

Educação Superior, no ano de 2008, do total de 47.098 alunos formados em Engenharia, 11,7% são formados nas áreas de eletricidade e energia e 10,2% nas áreas de eletrônica e automação.

Área	1999	2001	2002	2003	2004	2005	2006	2007	2008
Eletricidade e energia	16,5%	16,6%	17,0%	16,1%	14,8%	14,4%	13,9%	12,3%	11,7%
Eletrônica e automação	5,2%	6,2%	8,5%	8,8%	11,6%	11,8%	12,7%	11,0%	10,2%
Eng. civil e de construção	29,5%	26,6%	25,5%	24,5%	22,3%	20,3%	18,0%	14,7%	14,9%
Engenharia (cursos gerais)	18,8%	21,5%	19,6%	21,7%	22,2%	26,8%	29,1%	31,9%	32,4%
Eng. mecânica e metalurgia <sup>1</sup>	19,8%	19,4%	19,2%	19,3%	16,8%	15,2%	15,3%	14,7%	13,9%
Mineração e extração	0,5%	0,7%	0,6%	0,4%	0,5%	0,5%	1,1%	6,0%	6,4%
Processamento de alimentos	1,9%	3,0%	3,4%	3,3%	4,8%	4,7%	4,2%	4,2%	4,7%
Química e processos	7,3%	5,5%	5,4%	5,2%	6,2%	5,7%	5,0%	4,7%	4,8%
Outros cursos	0,4%	0,4%	0,8%	0,7%	0,7%	0,7%	0,7%	0,5%	1,0%
Total Engenharia (nº. abs.)	18.671	25.310	28.024	30.456	33.148	36.918	41.491	47.016	47.098

**Quadro 19 - Concluintes da Educação Superior em Engenharia: 1999 a 2008**  
**Fonte: (MEC, Inep, Censo da Educação Superior, 2008)**

Outro dado importante para a classificação do nosso segmento e público alvo é a quantidade de formandos em Engenharia em universidades públicas (Federais, Estaduais e Municipais) e privadas. No ano de 2008, o percentual de concluintes em instituições públicas foi de 41,4%, contra 58,6% formados em universidades privadas. FONTE: MEC, Inep e Censo da Educação Superior.

Concluimos com esta análise, que o total de clientes potenciais somente no âmbito universitário, ou seja, sem considerar clientes que buscam nosso produto para treinamento de funcionários, hobbistas, entre outras atividades, poderá representar um total de vendas de mais de 1000 unidades da PDETR, considerando uma absorção de 1% desse potencial total.

Clientes Potenciais			Ano 2008	Total (%)
Segmento A	Eletrônica e Automação	Instituições Públicas	2281	22%
Segmento B		Instituições Particulares	3229	31%
Segmento C	Eletricidade e Energia	Instituições Públicas	1989	19%
Segmento D		Instituições Particulares	2815	27%
<b>TOTAL</b>			<b>10314</b>	<b>100%</b>

**Quadro 20 - Clientes Potenciais**

Embora esses números sejam de 2008, ano da última atualização encontrada no site das fontes geradoras dos relatórios, acreditamos que caso ocorram variações nos números de 2010, essas devem ser pequenas e poderão ser desprezadas.

### 6.3.3. Segmento Alvo de Mercado

O segmento A, definido anteriormente, terá maior enfoque em todas as ações direcionadas que a nossa empresa adotará. Isto se dá pelo fato de o produto contemplar funcionalidades implementadas para este nicho de clientes em especial e , mais do que isso, ter sido pensado a partir de necessidades dos mesmos que não vinham sendo atendidas pelos fabricantes.

Ainda que represente um volume menor de clientes, acreditamos que o percentual de clientes que estará disposto a pagar pela solução é relativamente maior em relação ao total do que no segmento B. Portanto, o índice de absorção do produto neste caso tende a ser superior.

### 6.3.4. Necessidades do Mercado

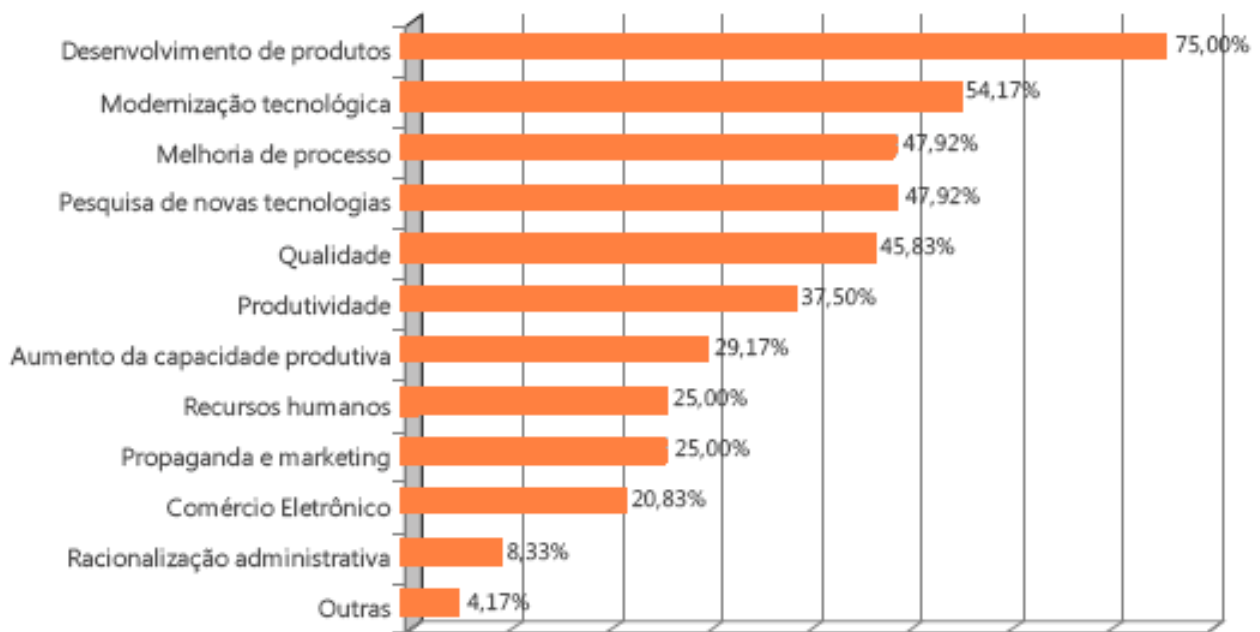
As necessidades do mercado de componentes eletrônicos crescem à medida que novas tecnologias são desenvolvidas e empregadas no nosso dia a dia. As ferramentas e recursos para que essas novas tecnologias sejam estudadas e continuem com o rápido crescimento deverão ser criadas e atualizadas na mesma velocidade.

Com isso, as reais necessidades que o segmento apresenta, são as constantes atualizações e, talvez o mais importante para o nosso segmento alvo, o



preço com que esses produtos são oferecidos no mercado brasileiro, principalmente para estudantes.

De acordo com dados fornecidos pela ABINEE (Associação Brasileira da Indústria Elétrica e Eletrônica), os investimentos do setor eletroeletrônico no Paraná se destinam a várias áreas. O “Desenvolvimento de Produtos” é responsável por 75% do total de investimentos feitos no setor. Em segundo lugar ficou a “Modernização Tecnológica”, com 54,17%, seguidos de “Melhoria no Processo” e “Pesquisa de Novas Tecnologias”, ambas com 47,92%.



**Gráfico 3 - Investimentos no setor eletroeletrônico**  
**Fonte: (ABINEE,2010)**

Embora ainda na lista tenhamos um bom percentual de investimentos feitos em “Qualidade”, “Produtividade”, “Aumento da Capacidade Produtiva”, “Recursos Humanos”, etc., notamos que os 4 primeiros itens do gráfico 3, estão relacionados com pesquisa e desenvolvimento, mostrando que a busca por novas tecnologias, treinamentos e mão de obra capacitada serão fundamentais na complementação de toda essa cadeia de investimentos.



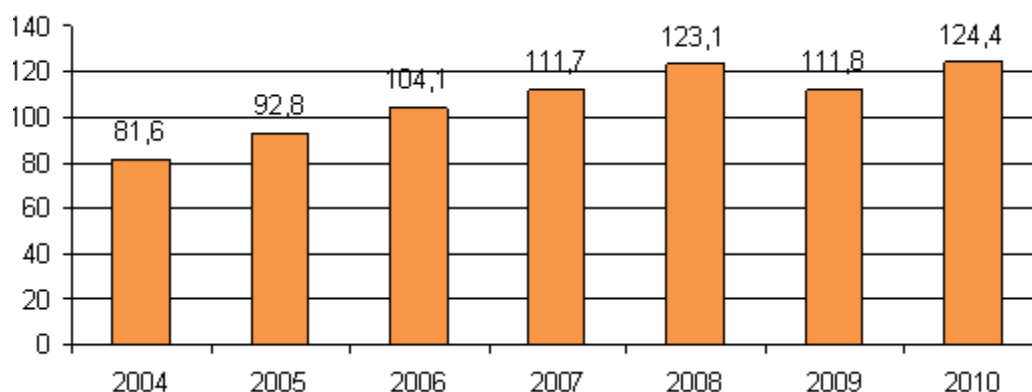
**Gráfico 4 - Treinamento de funcionários**

Fonte: (ABINEE,2010)

Além do constante investimento em tecnologia, dados fornecidos também pela ABINEE, regional do Paraná, mostram que o método mais utilizado para absorver a essa modernização tecnológica nas empresas é o treinamento de funcionários. Um total de 72,92% das empresas preferem treinar seus funcionários a ter que contratar funcionários já treinados, opção essa com 25%, conforme gráfico 4.

#### 6.3.5. Crescimento do Mercado

Segundo dados da ABINEE, em 2010 o faturamento do setor eletroeletrônico atingiu R\$ 124 bilhões, número esse que representa um crescimento de 11% em relação a 2009 (R\$ 112 bilhões). Esse crescimento é considerado abaixo da expectativa, e isso se deve a constante desvalorização do Dólar em relação ao Real, que atingiu 9%, também comparando o mesmo período de 2010 e 2011. Na comparação com 2010, período pré-crise, esse crescimento foi apenas de 1%, R\$ 123 bilhões.



**Gráfico 5 - Faturamento Setor Eletroeletrônico**  
**Fonte: (ABINEE, 2010)**

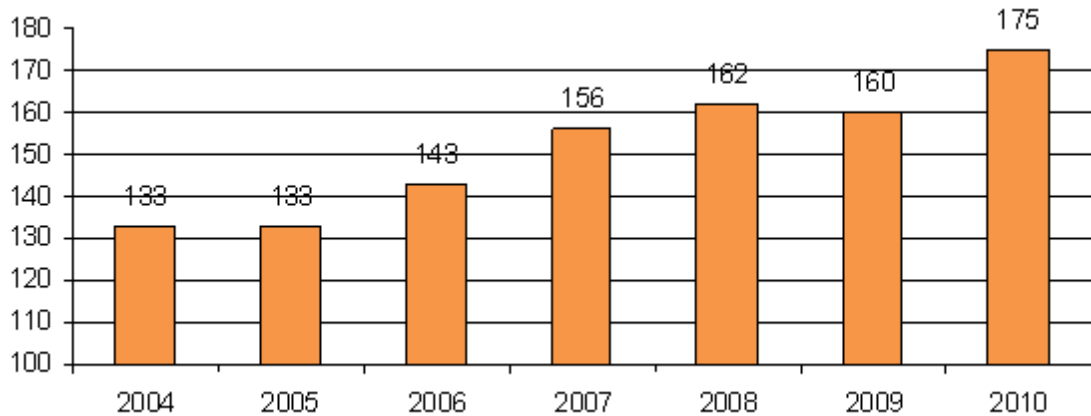
No gráfico abaixo, também elaborado pela ABINEE, observamos os números mostrados acima, porém agora segmentados por setor. Todos os segmentos apresentam um notável crescimento quando comparados os anos de 2009 e 2010. O único setor que apresentou queda nesse período foi o de telecomunicações, com 19%.

Faturamento Total por Área (R\$ milhões a preços correntes)	2008	2009	2010	2010 X 2009
Automação Industrial	3.446	2.943	3.237	10%
Componentes Elétricos e Eletrônicos	9.500	8.263	9.502	15%
Equipamentos Industriais	18.369	15.003	18.754	25%
GTD *	11.919	10.604	12.089	14%
Informática	35.278	35.278	39.864	13%
Material Elétrico de Instalação	8.323	7.954	8.909	12%
Telecomunicações	21.546	18.367	16.714	-9%
Utilidades Domésticas Eletroeletrônicas	14.710	13.427	15.307	14%
<b>Total</b>	<b>123.092</b>	<b>111.839</b>	<b>124.376</b>	<b>11%</b>

\*GTD - Geração, Transmissão e Distribuição de Energia Elétrica

**Quadro 21- Faturamento Total por Área**  
**Fonte: (ABINEE,2010)**

O crescimento do setor eletroeletrônico também pode ser constatado pelo número de empregos formais que foram gerados. Ainda segundo a ABINEE, no ano de 2010, 174,7 mil empregos formais foram gerados contra 159,8 mil do ano de 2009. Devido à crise global que se iniciou no final de 2008, observamos uma leve queda nessa número, porém que foi rapidamente retomado no ano de 2010.














**Gráfico 6 - Número de empregos formais**  
**Fonte: (ABINEE, 2010)**

#### 6.3.6. *Players*

As maiores empresas do segmento de componentes eletrônicos são estrangeiras, característica essa que nos favorece no mercado nacional devido aos preços dos produtos importados desses concorrentes. Embora esse fator atue a nosso favor, também possuímos como concorrentes empresas que atuam no mercado nacional, e também contam com preços acessíveis e características que conquistam seus consumidores. Praticamente todas elas possuem em seus catálogos de produtos *kits* de desenvolvimento e os mesmos são tão equivalentes quanto aos nossos produtos.

Dentre os principais *players* no mundo, veja abaixo o *ranking* dos maiores fabricantes de semicondutores:

Rank 2010	Rank 2009	Company	Country of origin	Revenue (million \$ USD)	2010/2009 changes	Market share
1	1	<a href="#">Intel Corporation</a>	 USA	40 394	+24.3%	13.2%
2	2	<a href="#">Samsung Electronics</a>	 South Korea	27 834	+60.8%	9.3%
3	3	<a href="#">Toshiba Semiconductor</a>	 Japan	13 010	+26.8%	4.3%
4	4	<a href="#">Texas Instruments</a>	 USA	12 944	+34.1%	4.3%
5	9	<a href="#">RenesasElectronics</a>	 Japan	11 840	+129.8%	3.9%
6	7	<a href="#">Hynix</a>	 South Korea	10 577	+69.3%	3.5%
7	5	<a href="#">STMicroelectronics</a>	 France  Italy	10 290	+20.9%	3.4%
8	13	<a href="#">Micron Technology</a>	 USA	8 853	+106.2%	2.9%
9	6	<a href="#">Qualcomm</a>	 USA	7 200	+12.3%	2.4%
10	14	<a href="#">Broadcom</a>	 USA	6 506	+52.1%	2.1%

**Quadro 22 - Ranking dos maiores players do mundo**  
**Fonte: (ISUPPLI,2010)**

O que buscamos para se tornar um grande *player* no mercado brasileiro além de um preço atrativo é estar sempre a par das novas tendências e com isso não deixar que os grandes *players* se distanciem de nós em termos de tecnologia.

### 6.3.7. Modelo de Distribuição

Inicialmente os *kits* didáticos serão repassados para lojas de varejo, desde que estudos comprovem que esse meio de distribuição é viável, sempre objetivando o preço final atraente para o consumidor. Lembrando que o preço do *kit* é o nosso principal argumento de vendas. Obviamente poderemos abrir as vendas diretamente para o consumidor final, quando se trata de um pedido de grande quantidade.

Para modelos de negociação com universidades, iremos fornecer os *kits* desmontados, conseguindo dessa forma um melhor preço para os estudantes e fazendo com que os próprios estudantes estudem previamente o *kit*, seus principais componentes e características. Esse modelo é de grande valia, pois os alunos participam da montagem do *kit* de forma prática, desenvolvem habilidades e se aproximam ainda mais da realidade.

### 6.3.8. Modelo de Competitividade

Como já mencionado anteriormente, os produtos oferecidos pelo mercado de *kits* de desenvolvimento apresentam características muito semelhantes. Variáveis de desempenho, armazenamento e funcionalidades são as alternativas que os fabricantes buscam aperfeiçoamentos, e com isso procuram seus nichos e ampliar suas participações no setor.

Como vimos no tópico *Players*, os nossos principais concorrentes são fabricantes que possuem suas atividades fora do país. Com isso os seus kits de desenvolvimento são importados para o Brasil e como consequência disso, os preços acabam subindo. Baseado nisso, acreditamos que o nosso principal fator competitividade será o preço da nossa plataforma didática, quando comparada com os produtos importados.

### 6.3.9. Proposta de valor

Como já mencionado anteriormente, o mercado alvo é composto pelo conglomerado de alunos em processo de capacitação técnica, no princípio mantendo o foco em alunos de engenharia eletrônica da UTFPR, que apresentam a necessidade de adquirir uma plataforma didática a um preço acessível.

Inicialmente, por se tratar de uma abordagem a alunos de engenharia eletrônica da UTFPR, o perfil definido para o consumidor almejado é o estudante inscrito em disciplinas de microcontroladores ou em outras cadeiras que dependam de uma plataforma microcontrolada para a execução de aulas práticas ou desenvolvimento de projetos eletrônicos.

Este aluno, normalmente, carrega consigo um computador portátil (*laptop* ou *netbook*) com portas USB e necessita de uma solução flexível para otimizar a sua rotina atarefada, um produto que possa ser utilizado de forma integrada ao seu computador em qualquer local que possua uma tomada da rede elétrica com saída 110V ou 220V e o possibilite desenvolver suas tarefas de maneira rápida, sem a necessidade de instrumentos de bancada como analisador lógico ou osciloscópio.

Além de estudante de engenharia, o aluno ainda acumula responsabilidades como estagiário na universidade ou em uma companhia do ramo de tecnologia, de onde é proveniente o capital para investir no produto. Portanto, analisando essa

perspectiva o produto deverá ter um preço de R\$ 189,00, para que o estudante possa arcar com o mesmo utilizando uma parcela inferior à metade de sua remuneração mensal.

Este investimento é feito vislumbrando a aplicação do produto em mais de uma disciplina do curso e potencialmente em seu trabalho de conclusão de curso devido à familiaridade adquirida com a ferramenta de desenvolvimento graças à intensa utilização. Adicionalmente, a compra seria bem avaliada pela perceptível facilidade de configuração e, ao mesmo tempo, robustez e aplicabilidade do *kit* para as mais diversas necessidades, tanto na vida acadêmica quanto para eventuais hobbies que envolvam a eletrônica digital.

#### 6.4. Plano de Marketing

##### 6.4.1. Política de preços

Devido ao grande número de concorrentes com produtos similares ao nosso, a política de preços é um ponto que consideramos como ponto estratégico do nosso negócio.

Na análise financeira de viabilidade, foi prevista a seguinte composição de preço para a venda ao consumidor final.

- Plataforma Digital Emulada em Tempo Real: R\$ 189,00

Obviamente, este valor poderá ser alterado conforme o número de equipamentos comprados e o prazo com que nosso produto é difundido no mercado.

##### 6.4.2. Estratégia de Vendas

A configuração técnica do produto potencializa a sua aplicação para pesquisa e desenvolvimento, aplicação direta em disciplinas para a formação acadêmica e uma excelente oportunidade para hobbistas que procuram criar e desenvolver produtos nas horas vagas.

Nossa estratégia de vendas não vai além da satisfação dos primeiros usuários do nosso produto, aliado a um preço bastante convidativo quando

comparado com os produtos disponíveis no mercado, sejam estes importados ou não.

## 6.5. Plano Financeiro

Nesta seção serão apresentados os investimentos iniciais, custos e despesas relativos ao negócio em seus primeiros quatro anos. Em seguida é feita a Demonstração de Resultados do Exercício para os mesmos quatro anos, mostrando de que forma os gastos impactam sobre o resultado final. É utilizado o lucro líquido como indicador financeiro, de modo a tornar mais clara a evolução do negócio ao longo dos anos considerados.

### 6.5.1. Investimento Inicial

O investimento total inicial é de aproximadamente R\$ 132 mil, valor este que leva em consideração somente as instalações da empresa, bem como todo o suporte jurídico e contábil para a abertura da mesma. Esse investimento inicial será disponibilizado pelos três sócios fundadores da METech, Mateus, Erikson e Thiago.

**Tabela 2 - Investimento Inicial**

<b>Start-up: jurídica / contábil</b>	<b>1.450,00</b>
<b>Start-up: Escritório</b>	<b>42.300,00</b>
<b>Start-up: Tecnologia</b>	<b>82.000,00</b>
<b>Start-up: Marketing</b>	<b>6.300,00</b>
<b>TOTAL</b>	<b>132.050,00</b>

### 6.5.2. Receitas

A receita provém da venda das plataformas digitais cobrada por unidades. Como mencionado anteriormente, a mesma terá um custo inicial de R\$ 189,00. Neste plano financeiro projetamos vender 800 unidades no primeiro ano, conforme tabela abaixo.



Tabela 3 - Unidades produzidas X Custo de Produção

ANO 1		ANO 2		ANO 3		ANO 4	
Semestre 1		Semestre 3		Semestre 5		Semestre 7	
Unidades	Total	Unidades	Total	Unidades	Total	Unidades	Total
400	75.600,00	500	94.500,00	600	113.400,00	550	103.950,00
Semestre 2		Semestre 4		Semestre 6		Semestre 8	
400	75.600,00	500	94.500,00	500	94.500,00	550	103.950,00
<b>151.200,00</b>		<b>189.000,00</b>		<b>207.900,00</b>		<b>207.900,00</b>	

### 6.5.3. Custos e Despesas

Os custos referentes aos três primeiros anos do negócio foram levantados com base nos gastos com matéria-prima, mão-de-obra e outros gastos diretamente relacionados à produção ou não.

Tabela 4 - Custos Fixos

Descrição	Ano 1	Ano 2	Ano 3	Ano 4
Investimento Inicial	132.050,00	-	-	-
Folha Pagamento/Secretária	12.000,00	13.200,00	14.520,00	15.972,00
Condomínio	4.800,00	4.800,00	6.000,00	6.000,00
Telefone	2.400,00	3.000,00	3.000,00	3.600,00
Luz	2.880,00	3.600,00	3.600,00	3.840,00
Material de escritório: papel, cartucho, pasta	3.600,00	3.600,00	4.800,00	4.800,00
Material de limpeza	480,00	600,00	600,00	600,00
Motoboy	408,00	600,00	600,00	600,00
<b>Total</b>	<b>26.568,00</b>	<b>29.400,00</b>	<b>33.120,00</b>	<b>35.412,00</b>

### 6.5.4. Ponto de Equilíbrio

O ponto de equilíbrio para o primeiro ano, ou seja, a quantidade de PDETR que a METech terá que vender para que seus custos fixos mais variáveis sejam cobertos, sem lucro e também sem prejuízo, é de 206 unidades vendidas. As mesmas projeções foram feitas para os próximos anos, haja vista que os custos fixos mudam no decorrer dos anos, e são apresentados abaixo.

Tabela 5 - Ponto de Equilíbrio

Ano	Ano 1	Ano 2	Ano 3	Ano 4
Custos Fixos	26.568,00	29.400,00	33.120,00	35.412,00
Custos de fabricação da PDETR	60,00	60,00	60,00	60,00
Preço Final	189,00	189,00	189,00	189,00
Ponto de Equilíbrio	206	230	257	275

#### 6.5.5. Demonstrativo de Resultados

O Demonstrativo de Resultado do Exercício dos quatro primeiros anos demonstra que desde o primeiro ano o negócio gera lucro líquido, porém devido aos gastos com investimento iniciais o retorno do montante investido só retorna a partir do primeiro semestre do terceiro ano de operações.

Tabela 6 - Demonstrativo de Resultados

Item	1º Sem.	2º Sem	3º Sem	4º Sem	5º Sem	6º Sem	7º Sem	8º Sem
Receita Bruta	75.600	75.600	94.500	94.500	113.400	94.500	103.950	103.950
Impostos Sobre Vendas	-11.340	-11.340	-14.175	-14.175	-17.010	-14.175	-15.593	-15.593
Receita Líquida	64.260	64.260	80.325	80.325	96.390	80.325	88.358	88.358
Custos Fabricação	-30.000	-30.000	-30.000	-30.000	-30.000	-30.000	-24.000	-36.000
Lucro Bruto	34.260	34.260	50.325	50.325	66.390	50.325	64.358	52.358
Despesas Fixas	-13.284	-13.284	-14.700	-14.700	-16.560	-16.560	-17.706	-17.706
Lucro Operacional	20.976	20.976	35.625	35.625	49.830	33.765	46.652	34.652
Imposto de Renda	-3.146	-3.146	-5.344	-5.344	-7.475	-5.065	-6.998	-5.198
Saldo Anterior	-132.050	-114.220	-96.391	-66.110	-35.828	6.527	35.227	74.881
<b>Lucro Líquido</b>	<b>-114.220</b>	<b>-96.391</b>	<b>-66.110</b>	<b>-35.828</b>	<b>6.527</b>	<b>35.227</b>	<b>74.881</b>	<b>104.335</b>

#### 6.6. CONSIDERAÇÕES SOBRE O PLANO DE NEGÓCIOS

O presente plano de negócios, com essencial conclusão no demonstrativo de resultado de exercício, demonstra que o estudo em questão é viável, porém o retorno acontece somente no 3º ano. É importante lembrar também que todo o aporte inicial é feito pelos sócios da empresa, não dependendo assim de empréstimos de terceiros no primeiro momento.

Embora em números apresentados sejam satisfatórios, o retorno esperado só será possível se todas as metas de vendas foram cumpridas, metas essas que são consideradas ousadas para uma empresa que está começando no mercado de

componentes e kits eletrônicos. Outras possibilidades e metas de vendas, menores do que esta, foram estudadas e o resultado obtido é apenas de um retorno do investimento mais tardio do que o atual apresentado neste plano de negócios.

Outro fator importante que impulsiona os números é o fato dessa empresa ser completamente gerida e estruturada apenas pelos seus 3 sócios. Há estudos para que no futuro esse panorama mude, e contratemos funcionários para a realização do trabalho operacional, e com isso os proprietários da empresa possam se dedicar a pesquisas e desenvolvimento de novos produtos e parceiros.

## **7. GESTÃO DE PROJETO**

### **7.1. GERÊNCIA DO TEMPO DE PROJETO**

Demonstraremos neste capítulo o cronograma das atividades previstas no início desse trabalho de conclusão de curso, o que realmente foi executado e os motivos que impactaram diretamente no fator tempo.

#### **7.1.1. Cronograma Planejado e Custos Homem/Hora**

No quadro 23 foram previstas as atividades e responsabilidades dos 3 integrantes do grupo para o decorrer do projeto. Embora essas atividades não tiveram seus tempos cronometrados durante suas execuções, elas serviram para que ao final de cada etapa, uma breve reflexão sobre o andamento do projeto pudesse ser feita e com isso definições de prioridade e direcionamento de ações viessem a ser o primeiro passo antes de seguirmos com o andamento do projeto.

Tarefa	Carga Individual (h)			Carga Total (h)
	Erikson	Mateus	Thiago	
<b>Projeto</b>				
Definir o projeto	8	8	8	24
<b>Mercado</b>				
Estudar mercado brasileiro	12	16	12	40
Estudar canais de distribuição	1	1	6	8
Definir microcontrolador	25	25	25	75
<b>Microcontrolador</b>				
Estudar a estrutura	18	18	18	54
Montar protótipo	20	15	15	50
Realizar testes de funcionamento	40	40	40	120
<b>JTAG</b>				
Pesquisar alternativas	35	35	35	105
Desenvolver interface	50	50	50	150
Montar e testar JTAG	35	30	30	95
<b>Placa de Circuito Impresso (PCI)</b>				
Rotear e Fabricar PCI	12	30	12	54
Elencar e cotar componentes	8	8	5	21
Montar PCI	8	8	10	26
Testar PCI	20	20	20	60
<b>Integração</b>				
Integrar JTAG e PCI	30	30	30	90
Testar JTAG e Ambiente de Desenvolvimento	50	50	50	150
Desenvolver demonstrações	20	20	20	60
Redigir manual	30	30	30	90
<b>Fechamento</b>				
Apresentar à banca	0,5	0,5	0,5	1,5
<b>Carga Horária Total (h)</b>				
	422,5	434,5	416,5	1273,5

Quadro 23 - Quadro de responsabilidades e ações

### 7.1.2. Disparidade entre o Cronograma Previsto e Executado

No geral, o total de horas efetivas para o desenvolvimento e conclusão do projeto ficou de acordo com o planejado, porém algumas atividades demandaram mais tempo que outras devido suas complexidades e/ou a não familiaridade do grupo com tecnologias empregadas na PDETR. Outras atividades dependiam de fatores externos, como o prazo de entrega de componentes, por exemplo. Também tivemos a efetivação da função de estagiário para a de analista de dois dos integrantes da equipe, reduzindo ainda mais o tempo disponível para o desenvolvimento do projeto.

Como todo o projeto envolvendo custos, prazos e tecnologia, contratemplos e dificuldades técnicas surgiram, porém cabe destacar o excelente planejamento e gestão de projetos que foi realizado pelo grupo logo no início das atividades. Esse fator permitiu com que dificuldades fossem contornadas e atrasos significativos fossem evitados.

## 7.2. GESTÃO DE CUSTOS

Demonstraremos neste capítulo todos os custos envolvidos no decorrer do projeto. É importante ressaltar que constam nessa gestão de custos, kits e componentes adquiridos para pesquisa e desenvolvimento, ou seja, materiais que nos proporcionaram o conhecimento para que decisões fossem tomadas e com isso chegássemos a uma plataforma ideal, porém muitos desses custos não fazem parte diretamente do projeto. Na tabela 7 estão demonstrados os custos para produção da versão final da PDETR. Vale comentar também que as importações foram feitas em dois fornecedores, DigiKey e Mouser.

Tabela 7 - Lista de componentes para a placa alvo

<b>Componentes para a placa alvo</b>				
<b>Item</b>	<b>Part Number</b>	<b>Quant.</b>	<b>Preço Unit. (USD)</b>	<b>Total (USD)</b>
RES 470 R	311-470ERCT-ND	50	0,0162	0,81
RES 4,75K	311-4.75KFRCT-ND	20	0,03	0,6
RES 10K	311-10.0KFRCT-ND	10	0,03	0,3
RES 1K	311-1.00KFRCT-ND	50	0,0228	1,14
CAP 0.1uF Ceramic X7R	478-1239-1-ND	100	0,0111	1,11
CAP 1.0uF Ceramic X7R	587-1242-1-ND	50	0,0636	3,18
CAP 4.7uF Tantalum	478-3887-1-ND	10	0,305	3,05
CAP 330uF Electrolytic	493-2082-1-ND	5	0,49	2,45
CAP 15uF Tantalum	478-3026-1-ND	5	0,45	2,25
Led GREEN	720-LGR971-KN-1	10	0,09	0,9
Led RED	720-LSR976-NR-1	25	0,102	2,55
IC REG LM2937	619-601-00513	2	1,24	2,48
DIODE SCHOTTKY 0.5A 20V	833-MBR0520-TP	10	0,17	1,7
RS-232 Interface IC	1016-1087-5-ND	3	2,01	6,03
DIODE ARRAY 3	576-SP0503BAHTG	6	0,8	4,8
IC 8051 MCU F386	336-2030-ND	3	3,99	11,97
<b>TOTAL</b>				<b>45,32</b>

Tabela 8 - Lista de componentes para Debugger

<b>Componentes para o debbuger</b>				
<b>Item</b>	<b>Part Number</b>	<b>Quant.</b>	<b>Preço Unit. (USD)</b>	<b>Total (USD)</b>
RES 2K	311-2.10KFRCT-ND	20	0,03	0,6
RES 470 R	311-470ERCT-ND	50	0,0162	0,81
RES 20K	311-20.0KFRCT-ND	10	0,03	0,3
RES 20 R	311-20ERCT-ND	10	0,021	0,21
RES 1.5K	311-1.50KFRCT-ND	10	0,03	0,3
RES 1K	311-1.00KFRCT-ND	50	0,0228	1,14
RES 18K	311-18.0KFRCT-ND	10	0,03	0,3
RES 3 R	311-3.00FRCT-ND	10	0,03	0,3
RES 1M	311-1.00MFRCT-ND	10	0,03	0,3
RES 1,5M	311-1.50MFRCT-ND	10	0,03	0,3
CAP 0.01uF Ceramic X7R	478-1383-1-ND	10	0,029	0,29
CAP 0.1uF Ceramic X7R	478-1239-1-ND	100	0,0111	1,11
CAP 1.0uF Ceramic X7R	587-1242-1-ND	50	0,0636	3,18
CAP 4.7uF Tantalum	478-3887-1-ND	10	0,305	3,05
LED RED/GREEN BI-COLOR	604-AM23ESGC	10	0,13	1,3
DIODE ARRAY 4CH	576-SP0504BAHTG	6	0,93	5,58
IC OPAMP 2.7MHZ	595-TLV4110ID	5	3,01	15,05
IC 8051 MCU F321	634-C8051F321-GM	3	5,72	17,16
MOSFET P-CH	NDS352APCT-ND	10	0,485	4,85
Transistor PNP	512-MMBT2907AD87Z	10	0,04	0,4
<b>TOTAL</b>				<b>56,53</b>

Além desses custos relacionados nas Tabelas 5 e 6, os itens relacionados abaixo, na Tabela 7, tiveram importante papel no estudo e entendimento das tecnologias envolvidas em todo o projeto.

**Tabela 9 - Lista de componentes para pesquisa**

<b>Componentes utilizados em pesquisa</b>				
<b>Item</b>	<b>Part Number</b>	<b>Quant.</b>	<b>Preço Unit. (USD)</b>	<b>Total (USD)</b>
C8051F380 Kit	634-C8051F380DK	1	98,8	98,8
C8051F005 Board	634-C8051F005TB	1	35	35
ADAPTER PROGRAM TOOLSTICK F321	634-TOOLSTICK321PP	1	69	69
BUFF TRI-ST	595-74LVC2G125DCTRE	12	0,48	5,76
DIODE SCHOTTKY	621-SDMG0340LA-F	10	0,36	3,6
PTC RESTTBLE 1.0A 6V CHIP 1206	576-1206L050YR	5	0,27	1,35
SWITCH TACT 6MM SPST-NO	611-PTS645SL70LFS	30	0,09	2,7
IC REG TEXAS	296-13424-1-ND	4	0,6	2,4
Potentiometer 10K	652-3302W-3-103E	5	0,48	2,4
CAP 330uF Electrolytic	647-UVR1E331MPD1TD	5	0,06	0,3
CAP 4.7uF Tantalum	647-F931A475KAA	10	0,15	1,5
CAP 15uF Tantalum	80-T491A156K004	10	0,16	1,6
IC 8051 MCU F018	336-1197-ND	3	10,67	32,01
11,0592 MHz Crystal	535-10217-1-ND	3	0,41	1,23
RES 0 R	311-0.0ERCT-ND	10	0,021	0,21
RES 2 R	311-2.00FRCT-ND	10	0,03	0,3
RES 24.9 R	311-24.9FRCT-ND	50	0,0228	1,14
RES 100K	311-100KFRCT-ND	10	0,03	0,3
CAP 33pF Ceramic X7R	709-1174-1-ND	10	0,044	0,44
IC REG TEXAS	296-13424-1-ND	6	0,66	3,96
TOOLSTICK DEBUG ADAPTER	336-1347-ND	1	9,5	9,5
DAUGHTER CARD C8051F321	336-1480-ND	1	10,5	10,5
DAUGHTER CARD C8051F381	336-2020-ND	1	8,75	8,75
<b>TOTAL</b>				<b>292,75</b>

Conforme listado nas Tabelas 5, 6 e 7, o total gasto com componentes eletrônicos foi de US\$ 394,60, ou US\$ 537,77 com o frete incluso. Além do frete, a Tabela 8 demonstra custos com impostos e trâmites necessários no Brasil.

**Tabela 10 - Custos com impostos**

<b>Pago ao FedEx (R\$)</b>	
Imp. Importação (60%)	449,82
ICMS/GNRE (18%)	263,3
Desembaraço	49,02
Reembolso Infraero	2,46
<b>Total</b>	<b>764,6</b>



Um último custo agregado ao projeto foi o de confecção da placa de circuito impresso. A mesma foi orçada em R\$ 285,00 (4 unidades), com dimensões de 157x101x1,6 (mm), adicionado de R\$128,00 referente a fotopltagem. Esse orçamento foi passado pela Stick Circuitos Impressos LTDA, localizada em Belo Horizonte, Minas Gerais.

Apresentamos na Tabela 11 e 12, o custo para a fabricação de 50 PDETR, custo esse que ficou bem abaixo do demonstrado nesse capítulo devido as grandes quantidades de componentes comprados.

**Tabela 11 - Custo de Produção para 50 PDETR**

C8051F380DK (F386)		QTD. Na placa	Qtd. p/ compra	Preço unid	Preço (USD)
Resistores	470	3	100	0,0123	2,46
	4.75k	2	100	0,0174	1,74
	10k	3	100	0,0174	3,48
	1k	2	100	0,0174	1,74
Capacitores	0.1uF	14	500	0,0442	44,2
	1.0uF	4	50	0,0636	12,72
	4.7uF	3	100	0,266	53,2
	330uF	1	10	0,391	19,55
	15uF	1	10	0,392	19,6
Led	Green	2	100	0,0748	7,48
	Red	1	100	0,087	8,7
Regulador	LM2937 - 3.3	1	25	1,5904	79,52
Diodo	MBR0520	1	10	0,3	15
Transceiver	SP3223ECY	1	25	1,62	81
Diodo	DIODE SP0503BAHTG	1	10	0,809	40,45
uC	C8051F380	1	25	3,8412	192,06
<b>Total</b>					<b>582,9</b>

Tabela 12- Custo para produção de 50 Debug Adapters

Debbug Adapter		QTD. Na placa	Qtd. p/ compra	Preco unid	Preço (USD)
Resistores	2k	6	100	0,0174	5,22
	470	1	50	0,0162	0,81
	20k	1	50	0,0228	1,14
	20	1	50	0,0162	0,81
	1.5k	1	50	0,0228	1,14
	1k	2	100	0,0174	1,74
	18k	1	50	0,0228	1,14
	3	1	50	0,0228	1,14
	1M	1	50	0,0228	1,14
	1.5M	1	50	0,0228	1,14
Capacitor	0.01uF	2	100	0,0169	1,69
	0.1uF	4	100	0,0111	2,22
	1uF	1	50	0,0636	3,18
	4.7uF	2	100	0,266	26,6
	Bicolor	1	100	0,2853	28,53
Diodo	SPO503BAHT	1	10	0,809	40,45
Amp. Op.	TLV4110	1	25	2,5652	128,26
uC	C8051F321	1	25	5,1104	255,52
FET	NDS352AP	1	10	0,485	24,25
Transistor	MMBT2907A	2	100	0,09	9
<b>Total</b>					<b>535,12</b>

Concluimos com esse demonstrativo, que o custo de produção de para 50 unidades da PDETR, ficaria composto da seguinte maneira:

R\$ 21,92\* (US\$ 11,66) - Componentes para a placa principal;

R\$ 20,11\* (US\$ 10,70) - Componentes para o *Debug Adapter*;

R\$ 25,00 - Placa circuito impresso;

R\$ 15,00 - Cabos USB do tipo A-B;

R\$ 15,00 - Fonte de tensão 127/220V - 12V

O custo unitário de cada Plataforma Didática Emulada em Tempo Real ficaria em R\$ 97,03, custo esse muito abaixo das opções de kits disponíveis no mercado de *kits* eletrônicos. Já quando comparado com a versão da P51 que a UTFPR utiliza atualmente, o custo da PDETR é superior, porém se justifica devido ao comparativo entre ambas as placas. O custo hoje da P51 está em torno de 80 reais.

### 7.3. GESTÃO DE RISCOS

A análise de riscos é importante para prever as ocorrências que podem afetar negativamente o andamento do projeto. No contexto do desenvolvimento da Plataforma Digital Emulada em Tempo Real, um dos riscos mais críticos que consideramos foi com relação à escolha do microcontrolador que atendesse todas as especificações do projeto. Contudo, esse item é considerado como o de maior sucesso. Além de encontrar o microcontrolador ideal para o nosso projeto, esse microcontrolador possui muitos recursos na mesma pastilha, fato esse que diminuiu o tempo de pesquisa e desenvolvimento do projeto.

Com relação ao segundo item do Quadro 24, quando especificamos o ambiente de desenvolvimento como sendo o do próprio fabricante do microcontrolador, ou seja, a IDE (*Integrated Development Environment*) da Silicon Labs, nos surpreendemos com o sigilo aplicado a toda e qualquer informação relativo ao protocolo de comunicação entre a IDE e o dispositivo de gravação. Tal situação impactou no desenvolvimento ou na reengenharia do código fonte para o controlador C8051F321.

Também considerado na gestão de riscos, o tempo para a entrega dos kits e componentes eletrônicos que adquirimos para estudos, testes e desenvolvimento, foi um item que gerou atrasos no andamento do projeto. Os correios permaneceram em greve por 28 dias consecutivos. Como nossas encomendas dependiam diretamente dos serviços dos Correios, algumas etapas do projeto ficaram paradas, sofrendo um pequeno atraso.

Embora tenhamos encontrados canais de distribuição nacional para os materiais necessários, o representante não demonstrou interesse em nos fornecer pequenas quantidades dos *kits* e componentes, além é claro de termos que arcar com custos de frete e impostos. Com esse cenário, optamos por adquirir nosso material em fornecedores fora do Brasil. Embora os preços e prazos para entrega forem bem convidativos, fomos taxados por altos impostos de importação e frete, custo esse que atingiu quase 80% do valor total da compra.

Problemas de montagem de *hardware*, Aumento da cotação do dólar e Erros no projeto de *hardware* não apresentaram erros significativos.

Grau	Risco	Efeito	Probabilidade ( P )	Impacto ( I )	Ação
ALTO	Não encontrar controlador 8 <i>bits</i> com os requisitos	Redirecionar para outra arquitetura, possivelmente 16 <i>bits</i>	0,50	5	Conviver: Redefinir escopo do projeto para minimizar impacto nos resultados
ALTO	Falhas na integração dos blocos	Atraso nos testes	0,60	5	Reduzir: Buscar soluções consolidadas
MÉDIO	Atraso na entrega dos componentes	Atraso na montagem da placa	0,40	4	Reduzir: Buscar alternativas no mercado local
MÉDIO	Não encontrar um canal de distribuição nacional	Dificuldade na confecção de um novo lote	0,25	5	Reduzir: Investir em uma margem de segurança na compra do estoque
MÉDIO	Problemas de montagem de hardware	Nova montagem	0,50	4	Eliminar: Estabelecendo revisões do projeto
BAIXO	Aumento da cotação do dólar	Aumento no custo de insumos	0,25	3	Conviver: Comprar componentes com antecedência
BAIXO	Erros no projeto de hardware	Retrabalho	0,20	4	Eliminar: Realizar simulações antes da fabricação

Quadro 24 - Lista de riscos

#### 7.4. CONSIDERAÇÕES SOBRE GESTÃO DO PROJETO

Inicialmente a maior dificuldade foi tomar as decisões e rumos do projeto, uma vez que a equipe é responsável por explorar alternativas de maneira a atingir os objetivos propostos. Além do caráter científico da pesquisa desenvolvida, forçadamente tivemos que adotar ferramentas de gestão, de forma que gerenciasse o conhecimento, e ao mesmo tempo torna-se eficiente o emprego dos recursos disponíveis.

Notavelmente a gestão de projeto proporcionou ao grupo um melhor andamento das atividades, acompanhamento em tempo real do desenvolvimento realizado pelos membros, estimativas e o estudo da viabilidade do produto frente ao mercado e seus concorrentes.

## **8. CONCLUSÃO MERCADOLÓGICA**

Administrados os riscos do projeto, o que gerou como resultado um protótipo com funcionalidade bastante próxima do esperado, consideramos que a gestão do projeto foi ingrediente fundamental para o sucesso técnico.

Observando ainda a conjuntura do mercado que se mostra favorável nos próximos anos ao ramo da eletrônica e, somando a isso, o fator inovador associado ao produto PDETR, pode-se acreditar que o mesmo terá espaço significativo dentre as opções disponíveis para o público alvo.

Sendo assim, a METech aposta em sua capacidade técnica e comercial para realizar a introdução deste produto, que se pretende que seja o primeiro de tantos, no mercado nacional e buscará, com isso, atingir as expectativas do consumidor e gerar retorno aos investimentos de seus sócios.

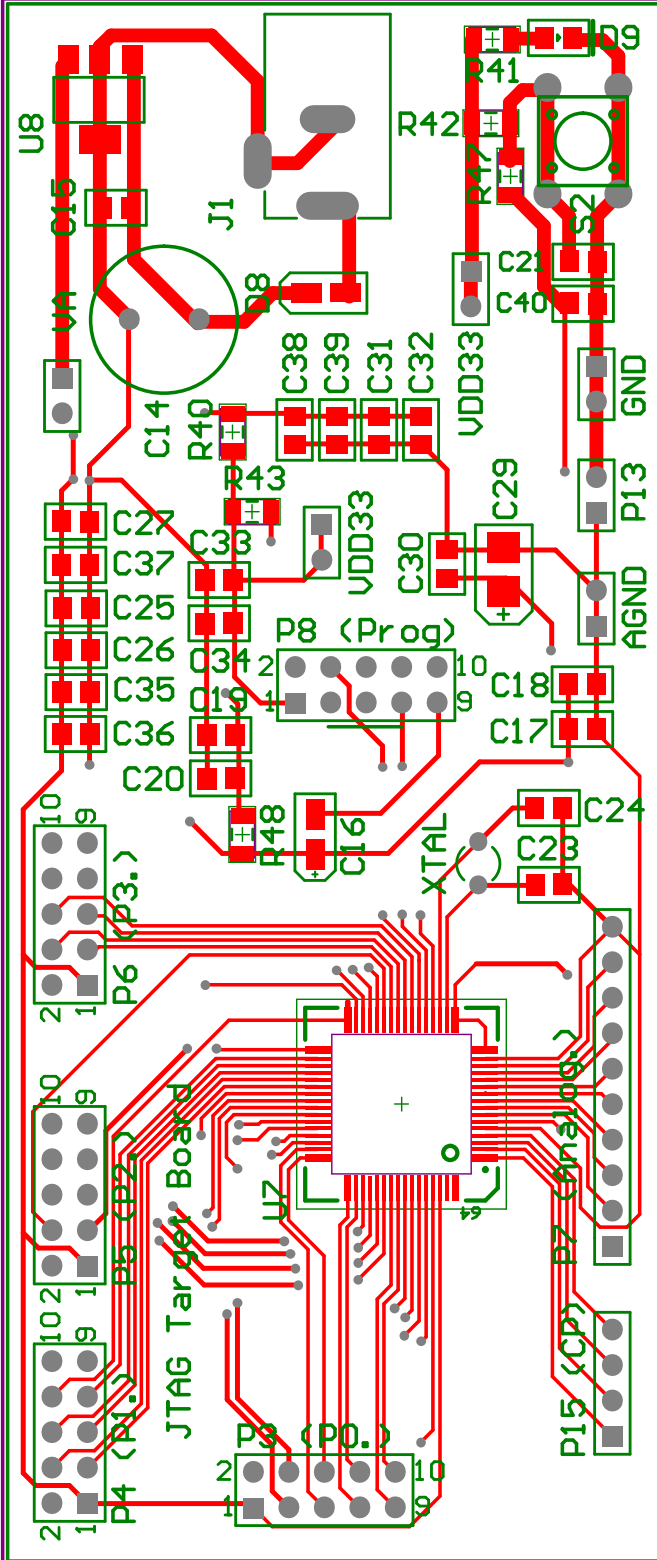
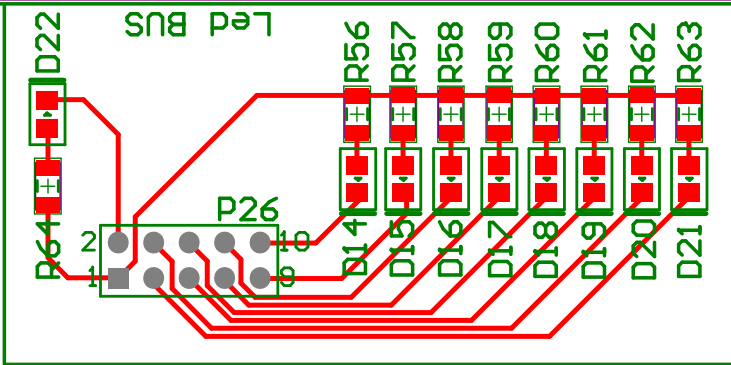
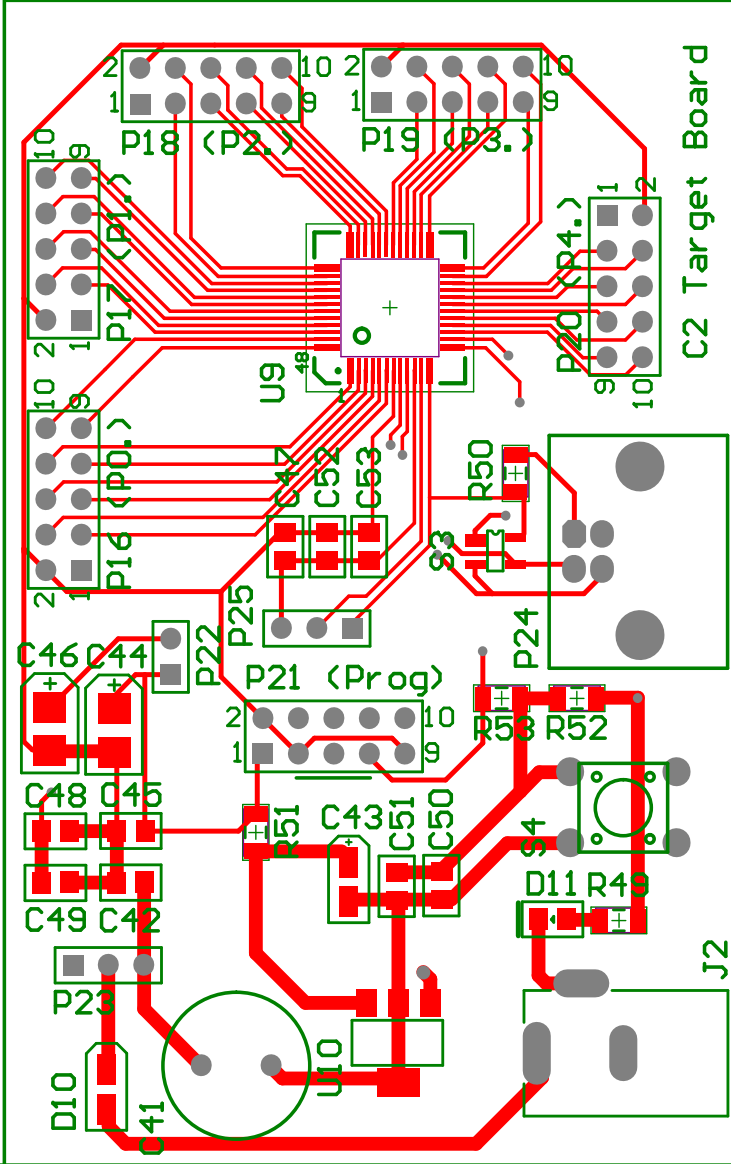
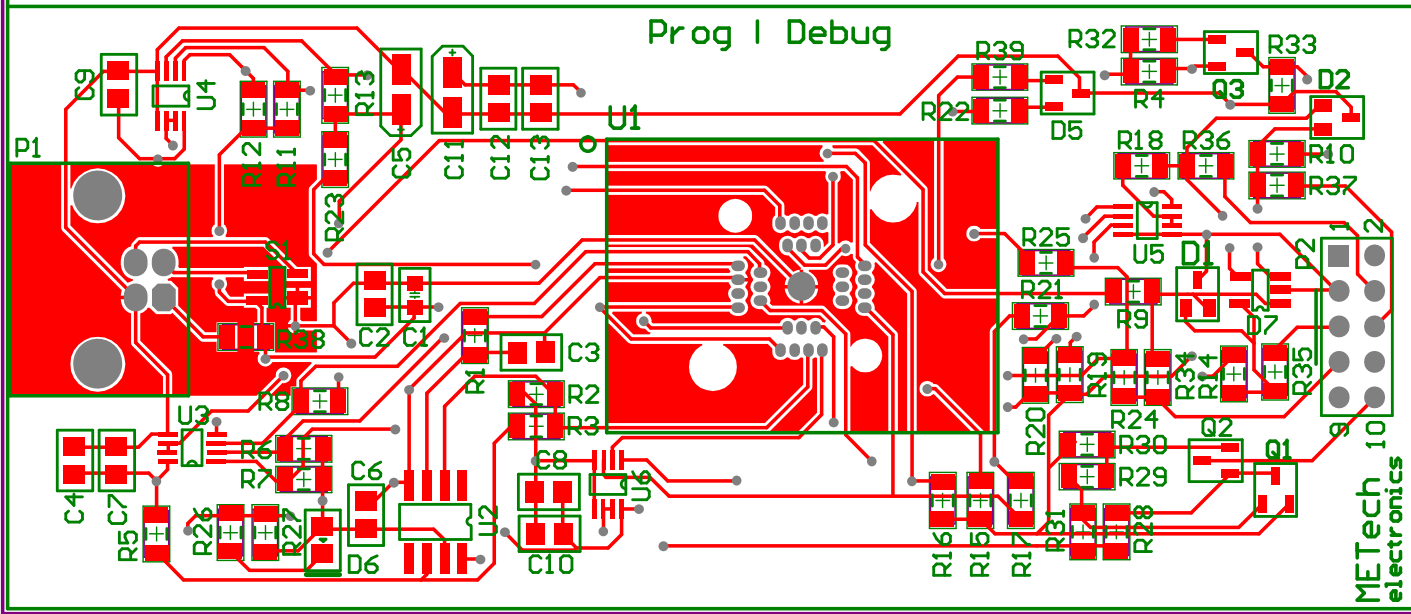
## REFERÊNCIAS

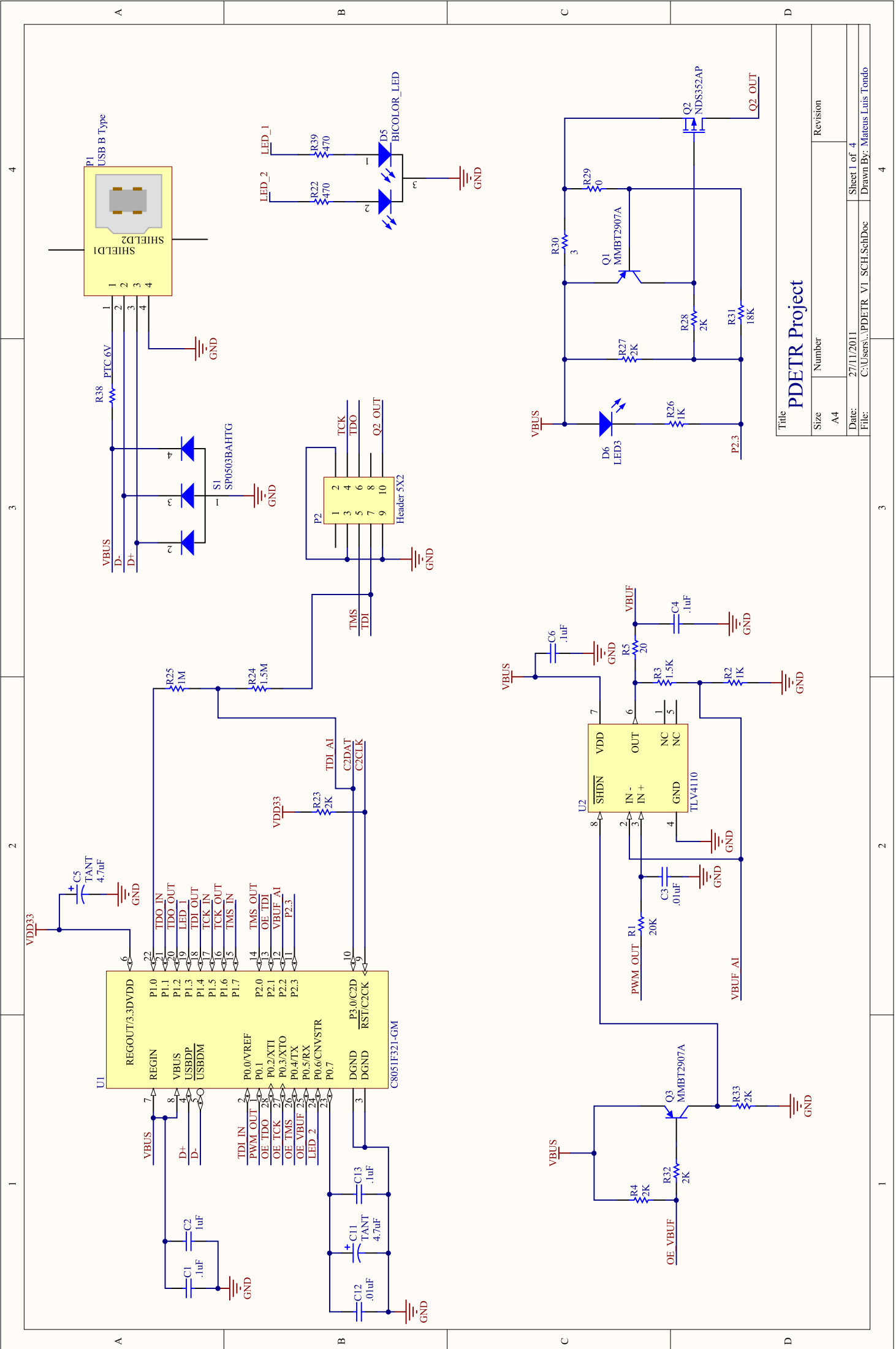
1. ZELENOVSKY, RICARDO; MENDONÇA, ALEXANDRE. *Microcontroladores - Programação e Projeto com a Família 8051*. Ed. MZ Editora. 2005.
2. MENDONÇA, ALEXANDRE; ZELENOVSKY, RICARDO. *PC: Um Guia Prático de Hardware e Interfaceamento*. Ed. MZ Editora. SP, 2002.
3. PORTALMercosur Online – Información Aduanas – Comércio Exterior.  
Disponível em: <<http://www.mercosuronline.com>>. Acesso em: 01/06/2011.
4. PIN SHARING techniques for the C2 interface. Application note. Silicon Labs.  
Disponível em:  
<<http://www.silabs.com/Support%20Documents/TechnicalDocs/an124.pdf>>  
Acesso em: 26/11/2011
5. FLASH Programming via the C2 interface. Notas de aplicação. Silicon Labs.  
Disponível em:  
<<http://www.silabs.com/Support%20Documents/TechnicalDocs/an127.pdf>>  
Acesso em: 26/11/2011
6. PROGRAMMING flash through the JTAG interface. Notas de aplicação. Silicon Labs.  
Disponível em:  
<<http://www.silabs.com/Support%20Documents/TechnicalDocs/an105.pdf>>  
Acesso em: 26/11/2011
7. C8051F320/1. Manual. Silicon Labs.  
Disponível em:  
<<http://www.silabs.com/Support%20Documents/TechnicalDocs/C8051F32x.pdf>>  
Acesso em: 26/11/2011
8. C2 Interface  
Disponível em:  
<<http://www.silabs.com/Support%20Documents/TechnicalDocs/C2spec.pdf>>  
Acesso em: 26/11/2011

9. EIA standard RS-232-C: Interface between Data Terminal Equipment and Data Communication Equipment Employing Serial Binary Data Interchange. Washington: Electronic Industries Association. Engineering Dept. 1969.
  
10. UNIVERSAL Serial Bus Specification. Manual.  
Disponível em:  
<<http://www.usb.org/developers/docs/>>  
Acesso em: 26/11/2011
  
11. MCS 51 Microcontroller Family User's Manual, February 1994, Publication number 121517, Intel Corporation
  
12. IEEE Std 1149.1(JTAG) Testability Primer. Texas Instruments
  
13. MACKENZIE, I.Scott. *The 8051 Microcontroller*. Editora Prentice-Hall.1995
  
14. C8051F380/1/2/3/4/5/6/7. Manual. Silicon Labs.  
Disponível em:  
<<http://www.silabs.com/Support%20Documents/TechnicalDocs/C8051F38x.pdf>>  
Acesso em: 26/11/2011

**ANEXOS**







Title  
**PDETR Project**

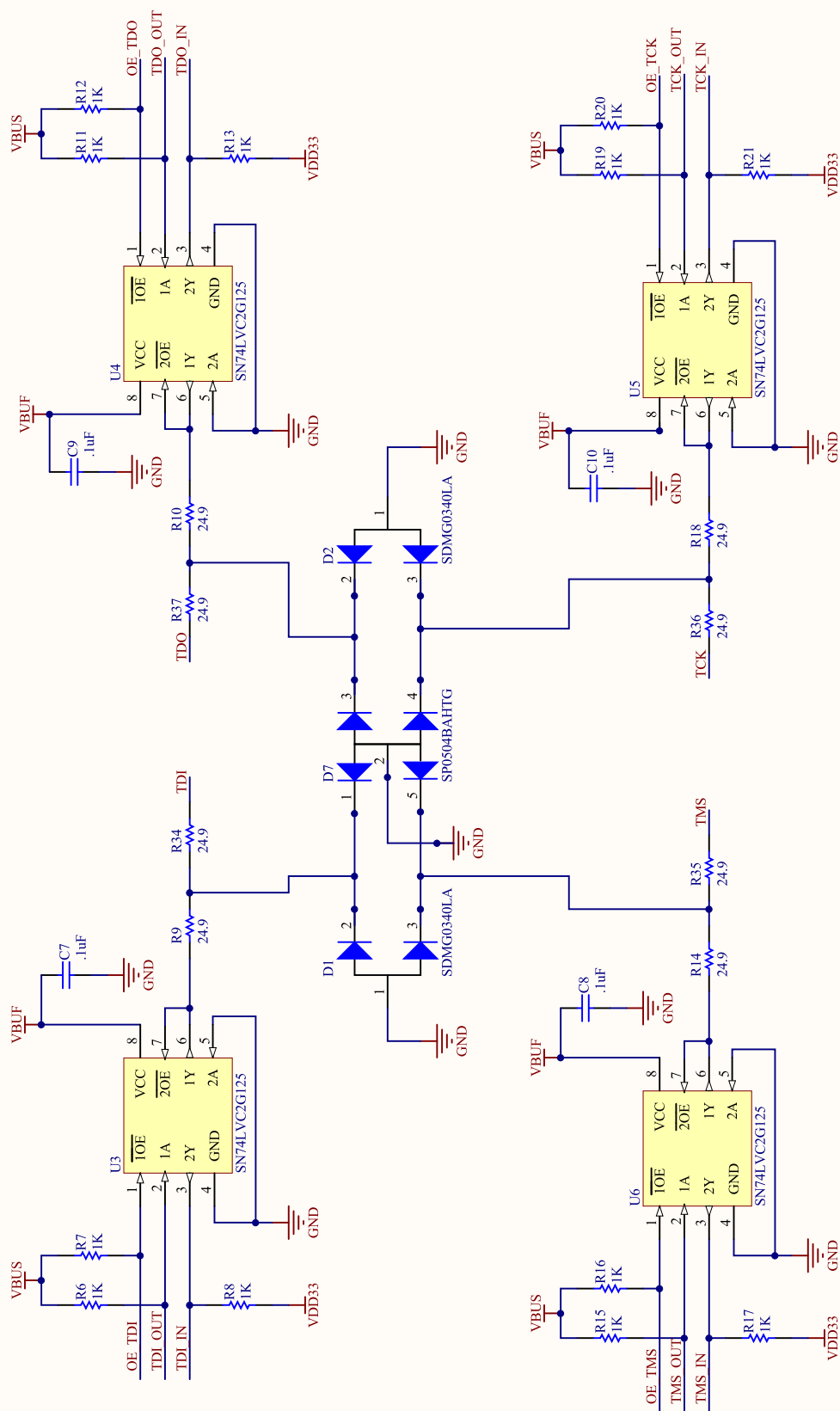
Size	Number	Revision
A4		
Date:	27/11/2011	Sheet 1 of 4
File:	C:\Users\... \PDETR_V1_SCH.SchDoc	Drawn By: Mateus Luis Tondo

A

B

C

D



Title  
**PDETR Project**

Size	Number	Revision
A4		
Date:	27/11/2011	Sheet 2 of 4
File:	C:\Users\... \PDETR_V1_SCH2.SchDoc	Drawn By: Mateus Luis Tonido

A

B

C

D



