



[www.fpgaparatodos.com.br](http://www.fpgaparatodos.com.br)

# Módulo PLDProg



## Manual do Usuário

Maio, 2013

## SUMÁRIO

|   |                                                     |    |
|---|-----------------------------------------------------|----|
| 1 | SOBRE O MÓDULO PLDPROG .....                        | 3  |
| 2 | ESTRUTURA DO PLDPROG .....                          | 6  |
| 3 | LAYOUT DA PLACA.....                                | 8  |
| 4 | DESCRIÇÃO DOS CONECTORES.....                       | 10 |
| 5 | VISÃO 3D DA PLACA.....                              | 11 |
| 6 | CONFIGURAÇÃO DO PLDPROG NO COMPUTADOR .....         | 12 |
| 7 | CONFIGURANDO O PLDPROG NO SOFTWARE QUARTUS II ..... | 17 |
| 8 | PLDPROG COMO INTERFACE JTAG DE COMUNICAÇÃO .....    | 19 |
|   | ANEXOS.....                                         | 24 |
|   | ANEXO 1 - Lista de Materiais.....                   | 25 |
|   | ANEXO 2 - Código VHDL da Placa .....                | 26 |
|   | ANEXO 3 - Programação do FT245 .....                | 30 |
|   | REFERÊNCIAS .....                                   | 31 |

## 1 SOBRE O MÓDULO PLDPROG

O módulo **PLDProg** é o equipamento desenvolvido pelo **PROJETO FPGA PARA TODOS** para a programação dos seus módulos de PLD e FPGA (kit em desenvolvimento). De fato, ele é também parcialmente compatível com a USBBlaster<sup>®</sup> da Altera [6], permitindo a programação de outros kits de CPLD e FPGA baseados em componentes deste fabricante. Observamos, entretanto que é fundamental que seja verificada esta compatibilidade em caso de se querer utilizar o PLDProg em substituição a uma USBBlaster<sup>®</sup>.

A Figura 1 ilustra a conexão do PLDProg, como interface entre um microcomputador padrão PC e uma placa de FPGA ou CPLD, dotado de uma entrada JTAG para programação.

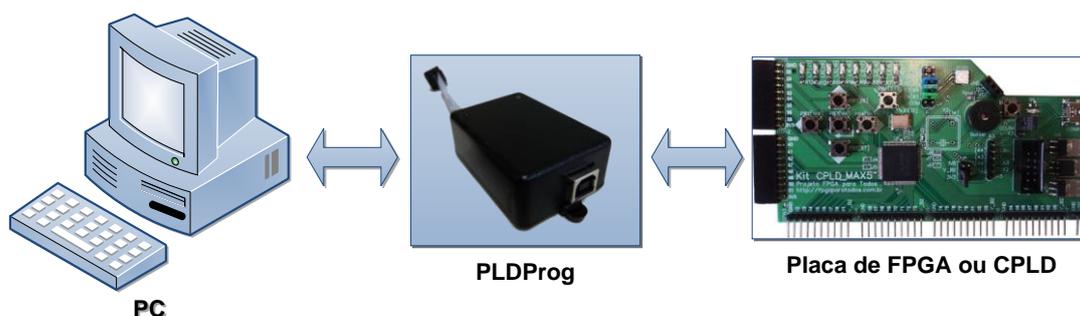


Figura 1 – Conexão da PLDProg com o PC e um kit de PLD.

O PLDProg é uma interface JTAG, e isto o torna útil para outras aplicações além da simples programação de um PLD. Interessantes aplicações, por exemplo, seriam:

- a comunicação entre um PC e um processador implementado no FPGA. A ALTERA, por exemplo, oferece uma PI (propriedade intelectual) que implementa uma UART via JTAG para o seu processador NIOS [1];
- depuração de projetos no FPGA. O *Signal Tap*, da ALTERA, é uma excelente ferramenta de depuração é oferecida pela ALTERA no Quartus II [2];
- teste de sistemas dotados de interface JTAG, através de ferramentas de “*Boundary Scan Testing*”, como se encontra descrito (por exemplo) em [3].

Este módulo foi criado a partir dos conceitos do projeto OpenJTAG [4] e do código para uma interface JTAG em VHDL de Kolja Waschk [5], com alguns ajustes para simplificação da placa e redução de custos, sem perda de suas funcionalidades. Duas versões do PLDProg foram criadas: com ou sem conversor de nível de tensão.

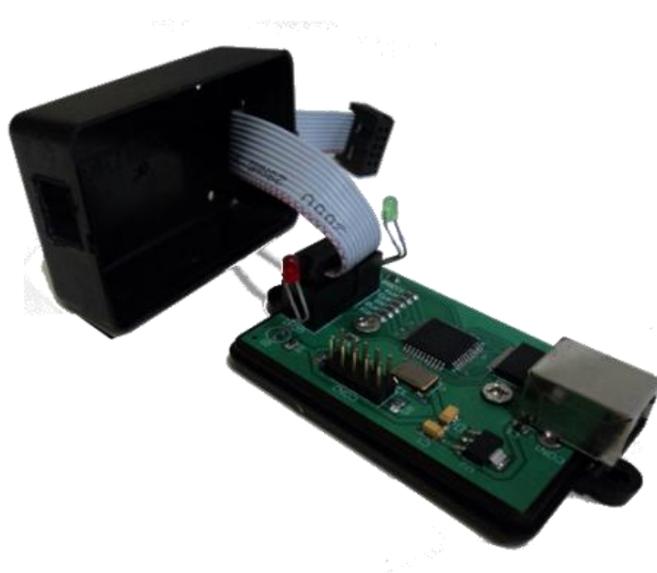
O projeto do hardware é composto principalmente de:

- Uma placa de circuito impresso (PCI) em duas camadas;
- Um CI FT245RL (conversor USB/FIFO);
- Um CI CPLD Altera EPM3064ATC44-10 TQFP44;
- Um oscilador de 24 MHz;
- Um regulador de tensão LM1117-3.3 3,3V LDO;
- Dois conversores de níveis de tensão Texas TXS0104 (na versão com conversor de nível de tensão).

O FT245 [7] é usado para converter dados da interface USB para um barramento paralelo conectado ao CPLD EPM3064 [8]. O CPLD recebe os dados e implementa o protocolo JTAG, levando os quatro principais sinais JTAG (TCK, TMS, TDI e TDO) para gravação de outros dispositivos que se comunicam neste padrão. O sinal TRST não é disponível no PLDProg.

A versão do PLDProg sem o conversor de nível de tensão permite gravar dispositivos-alvo que suportem nível de tensão de 3,3V (fornecidos internamente pelo regulador LM1117-3.3, a partir da saída de 5V da USB). Para a versão com o conversor de nível de tensão, os pinos do conector JTAG para os dispositivos-alvo passam por dois conversores de nível TXS0104 bidirecionais da Texas que permitem a conexão de dispositivos-alvo alimentado em diversas fontes de tensão como: 1,2V; 1,8V; 2,5V; 3,3V ou 5V.

O conector para o cabo JTAG é pino a pino compatível com o padrão da USB Blaster da Altera. A Figura 2 mostra o módulo PLDProg com a caixa aberta.



**Figura 2 – Visão geral do módulo PLDProg com a caixa aberta.**

## 2 ESTRUTURA DO PLDPROG

A estrutura da PLDProg como interface para CPLD, em modo JTAG, é apresentada no diagrama de blocos das Figura 3 (com conversor de nível de tensão) e Figura 4 (sem conversor de nível de tensão). As seguintes funções são executadas pelos blocos da placa:

- Conversão USB/FIFO: Este bloco se comunica com a interface USB do PC, recebendo dela os códigos para gravação no CPLD, e convertendo-os para um formato digital paralelo, que são enviados ao bloco “MÁQUINA\_JTAG”;
- O bloco “MAQUINA\_JTAG” recebe os comandos da interface USB e os converte para o protocolo JTAG, para comunicação com o “PLD alvo” (que deverá ser gravado com um projeto);
- O módulo conversor de nível permite, quando existe, que a placa de programação grave placas alvo alimentadas em diferentes níveis de tensão, como 1,8V, por exemplo;
- O conector JTAG CPLD permite a gravação inicial do CPLD do PLDProg, e que a interface JTAG do mesmo venha a ser usada em outras aplicações, em algum projeto particular.

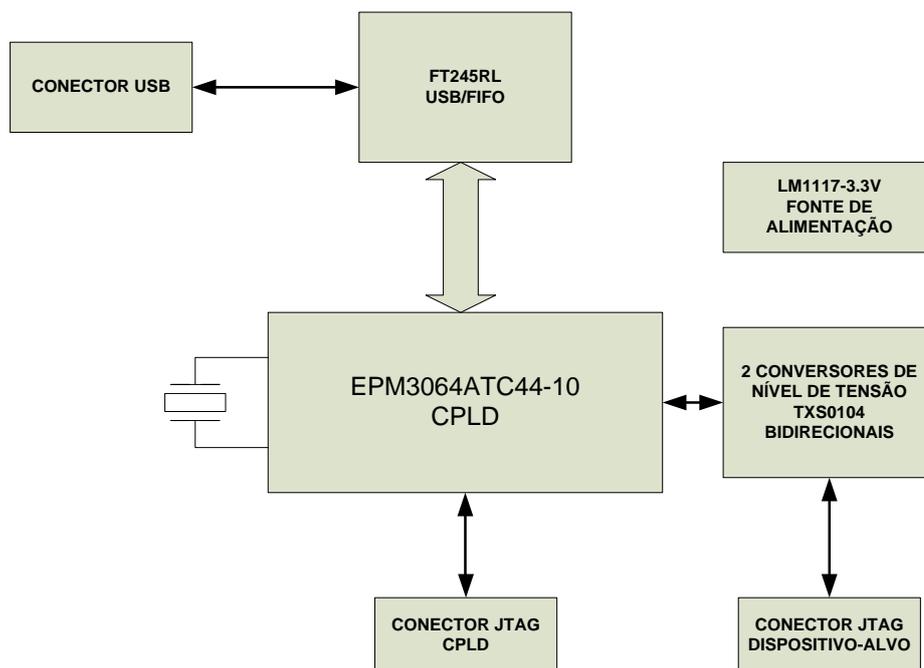


Figura 3 – PLDProg com conversor de nível de tensão.

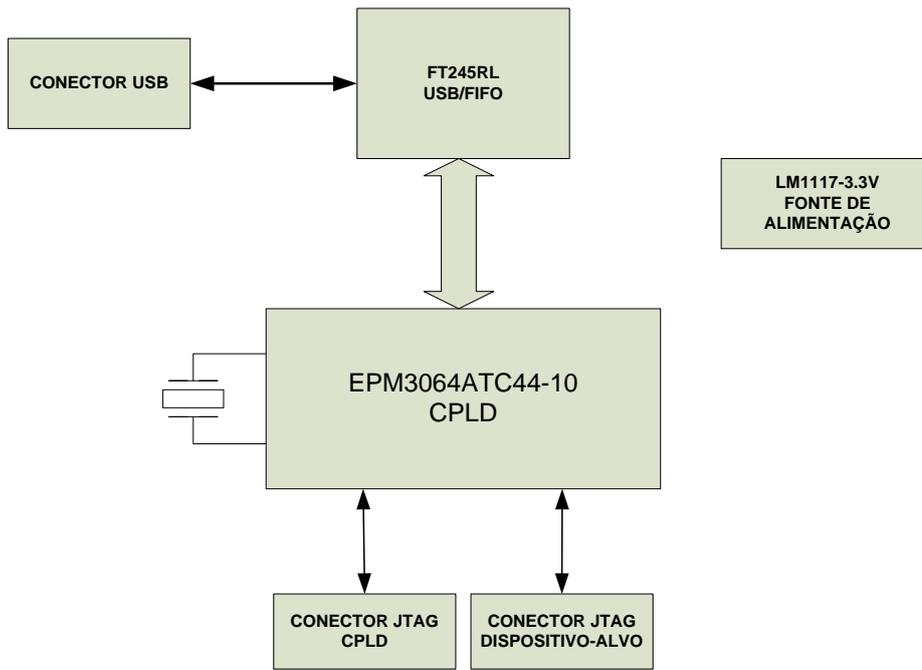


Figura 4 – PLDProg sem conversor de nível de tensão.

### 3 LAYOUT DA PLACA

Com conversor de nível de tensão (Figura 5):

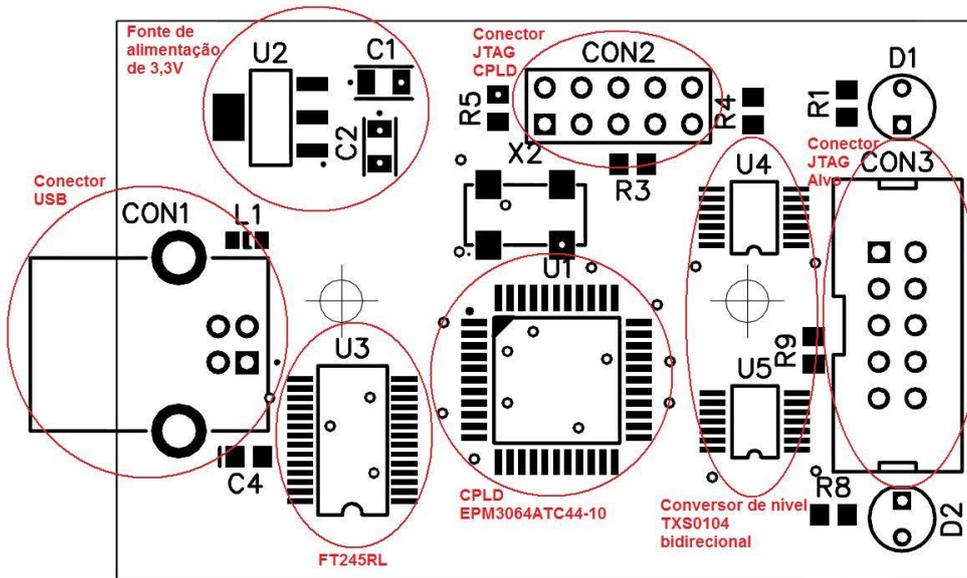


Figura 5 – PLDProg com conversor de nível de tensão.

Sem conversor de nível de tensão (Figura 6):

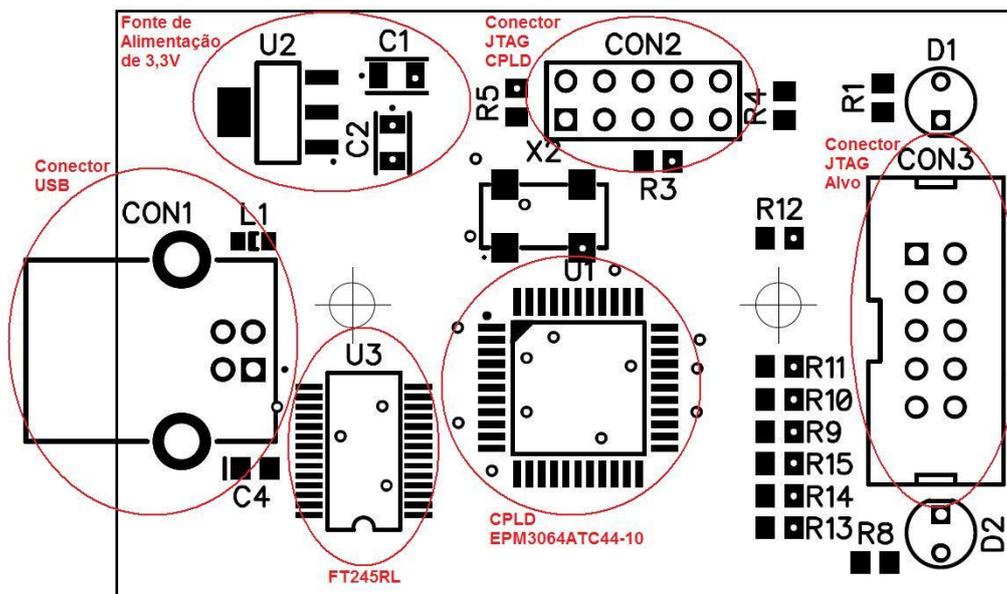


Figura 6 - PLDProg sem conversor de nível de tensão.

O tamanho da PCB é 59 milímetros x 39 milímetros (2,32" x 1,54") e a placa montada se encaixa dentro da caixa da Patola ([HTTP://www.patola.com.br/](http://www.patola.com.br/)) do modelo PBT-064, ligeiramente modificada por alguns recortes.

## 4 DESCRIÇÃO DOS CONECTORES

- a) **CON1** – Conector USB: Usado para conectar o módulo de gravação JTAG ao computador.
- b) **CON2** – Conector JTAG CPLD: Este conector é usado apenas para programar o CPLD do módulo, utilizando outro módulo de gravação JTAG, tais como, o USB *Blaster* ou o Byte *Blaster* da Altera.

| Pinos | Função         |
|-------|----------------|
| 1     | TCK/DCLK       |
| 2     | GND            |
| 3     | TDO/CONF_DON   |
| 4     | +VCCIO (+3,3V) |
| 5     | TMS/nCONFIG    |
| 6     | NC             |
| 7     | NC             |
| 8     | NC             |
| 9     | TDI/ASDI       |
| 10    | GND            |

- c) **CON3** – Conector JTAG Alvo: Este é o conector principal, usado para se comunicar com o dispositivo-alvo.

| Pinos | Função                                                                                                                                                                              |
|-------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1     | TCK/DCLK                                                                                                                                                                            |
| 2     | GND                                                                                                                                                                                 |
| 3     | TDO/CONF_DON                                                                                                                                                                        |
| 4     | +VCC<br>Módulo com conversor de nível de tensão:<br>Provido pelo dispositivo-alvo para alimentar as<br>E/S dos TXS0104.<br>Módulo sem conversor de nível de tensão:<br>Sem conexão. |
| 5     | TMS/nCONFIG                                                                                                                                                                         |
| 6     | nCE                                                                                                                                                                                 |
| 7     | DATAOUT                                                                                                                                                                             |
| 8     | nCS                                                                                                                                                                                 |
| 9     | TDI/ASDI                                                                                                                                                                            |
| 10    | GND                                                                                                                                                                                 |

O conector para o cabo JTAG tem distribuição de pinos e de sinais compatível com o padrão da USB Blaster da Altera (para a interface JTAG).

## 5 VISÃO 3D DA PLACA

As imagens 3D da placa foram extraídas do software de criação de esquemáticos e *layouts* de placas de circuito impresso.

Com conversor de nível de tensão (Figura 7):

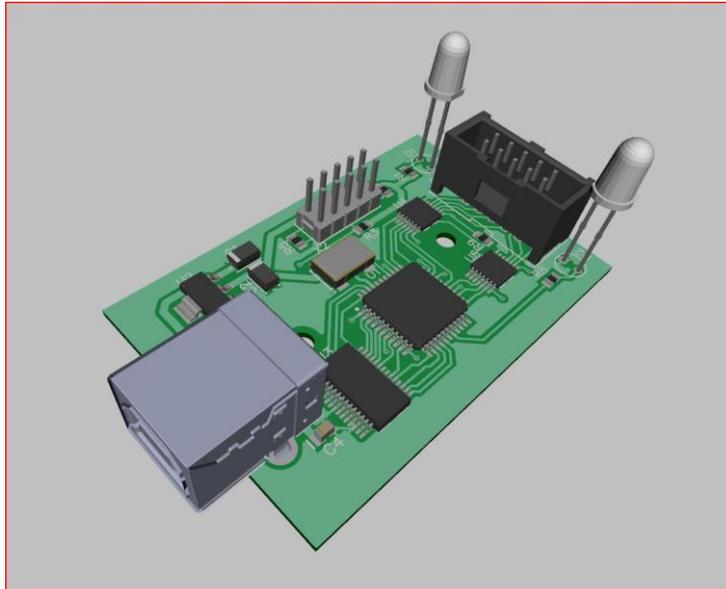


Figura 7 – Visão 3D do PLDProg com conversor de nível de tensão.

Sem conversor de nível de tensão (Figura 8).

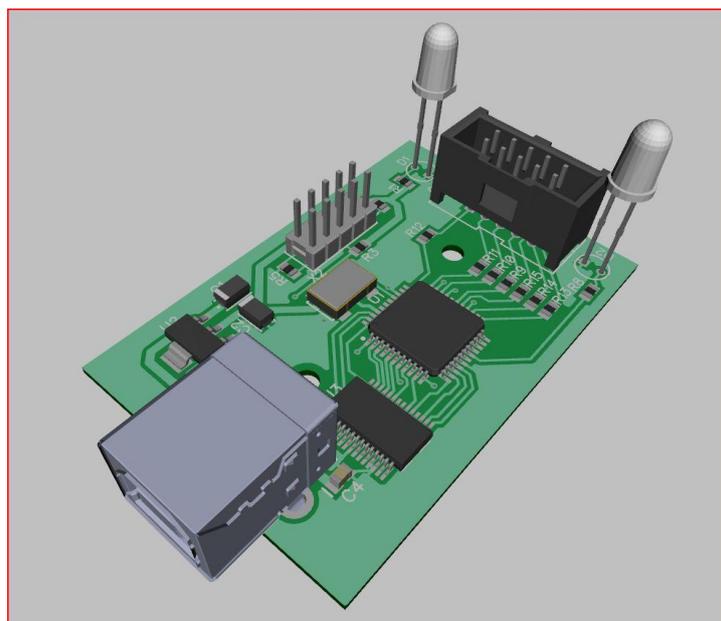
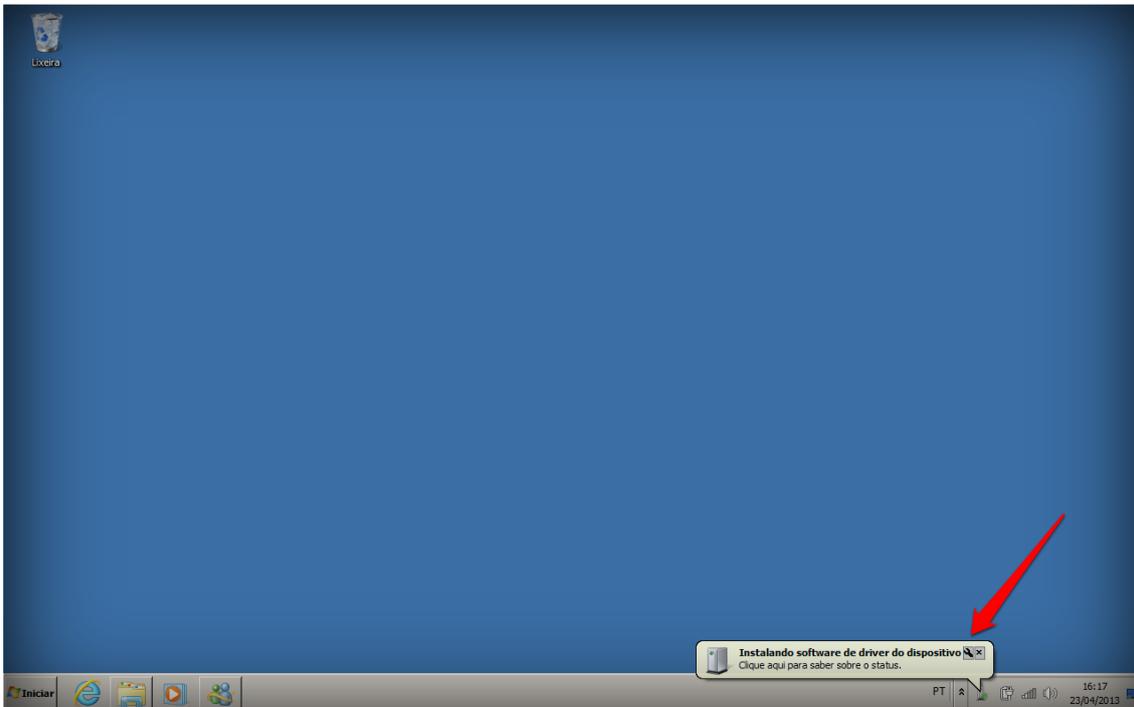


Figura 8 - Visão 3D do PLDProg sem conversor de nível de tensão.

## 6 CONFIGURAÇÃO DO PLDPROG NO COMPUTADOR

Ao conectar pela primeira vez o PLDProg ao seu computador, o dispositivo será reconhecido mas seu driver não será instalado, conforme mostra a Figura 9.



**Figura 9 – Reconhecendo o dispositivo no computador.**

Ao seguir o procedimento de instalação do driver do PLDProg certifique-se que você tenha feito a instalação correta de uma versão atualizada do Quartus II. O driver a ser instalado é o mesmo utilizado para a USB Blaster da Altera.

Para instalar o driver, deve-se clicar no ícone “Computador” e, com o botão direito do mouse e clicar em “Propriedades” (Figura 10):

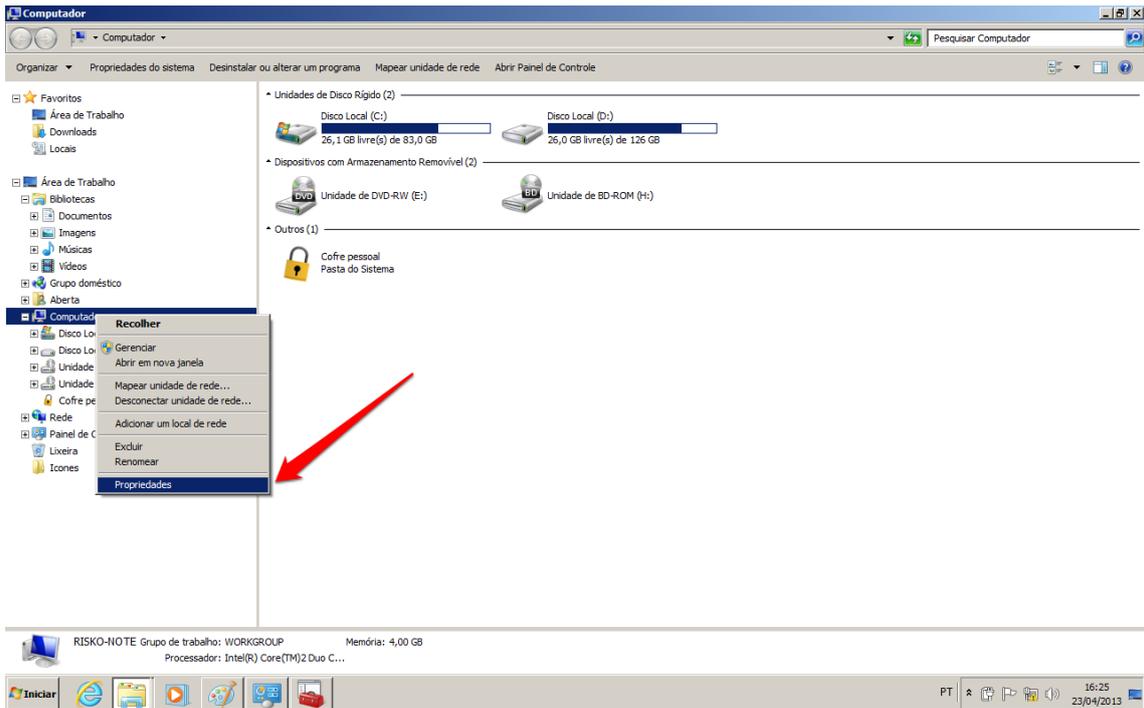


Figura 10 – Instalando o driver do PLDProg (Procedimento 1).

Em seguida se abrirá a janela “Sistema”. Nela deve-se clicar em “Gerenciador de Dispositivos” (Figura 11).

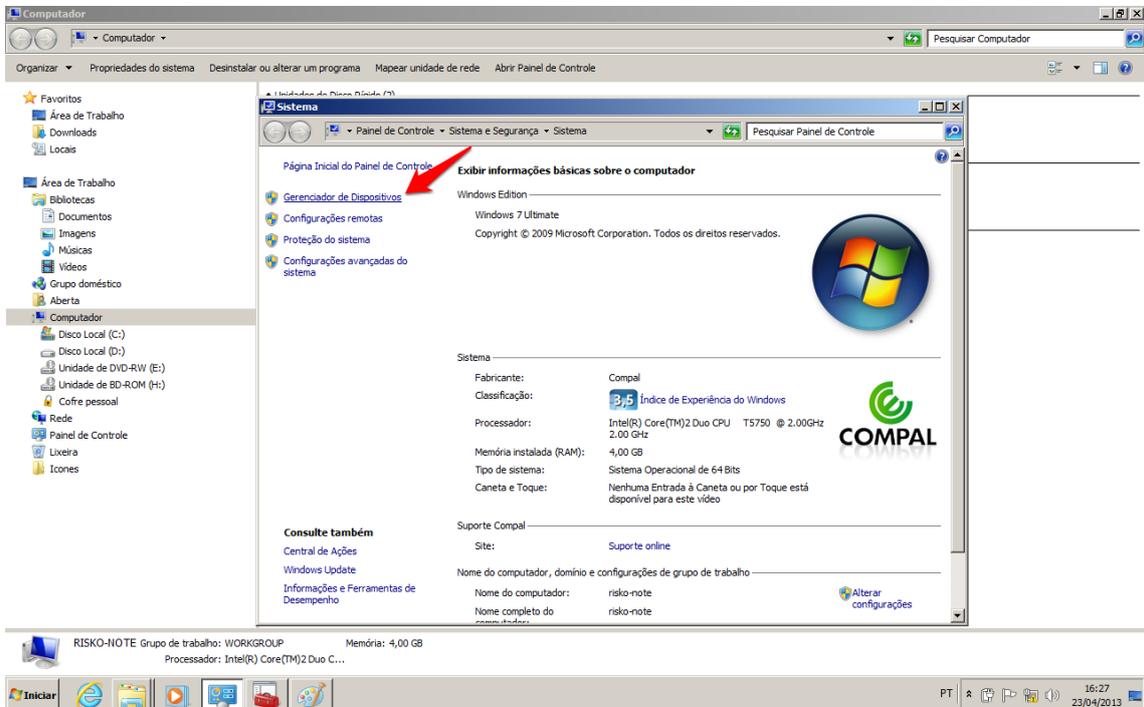
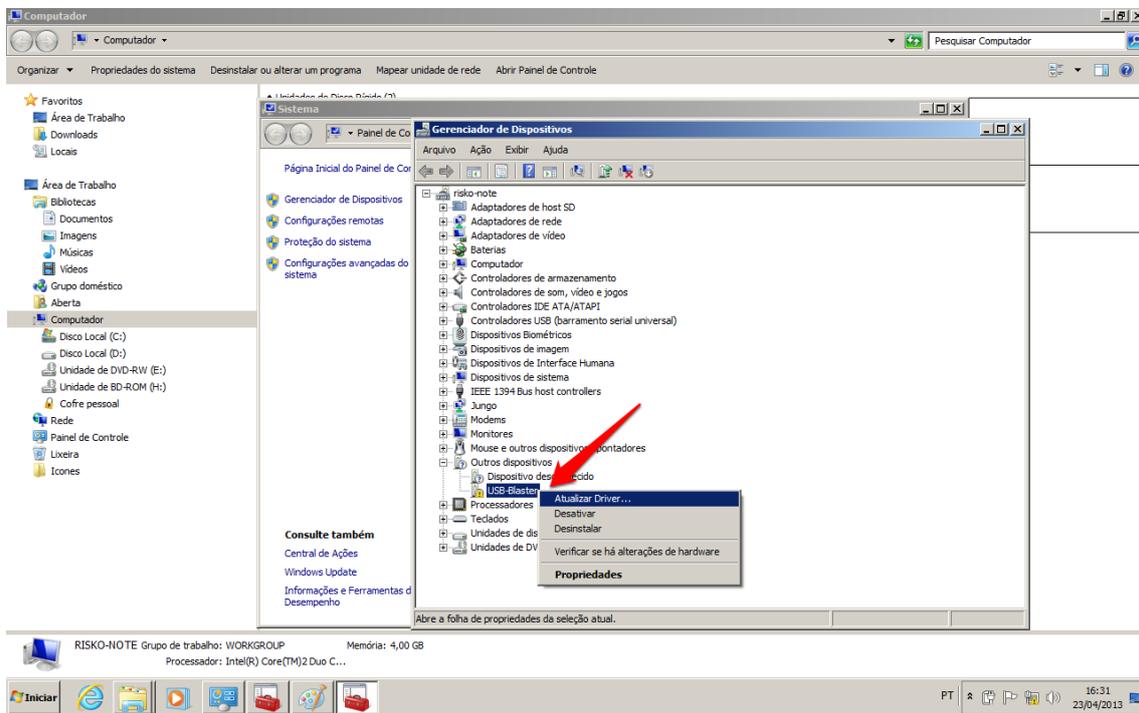


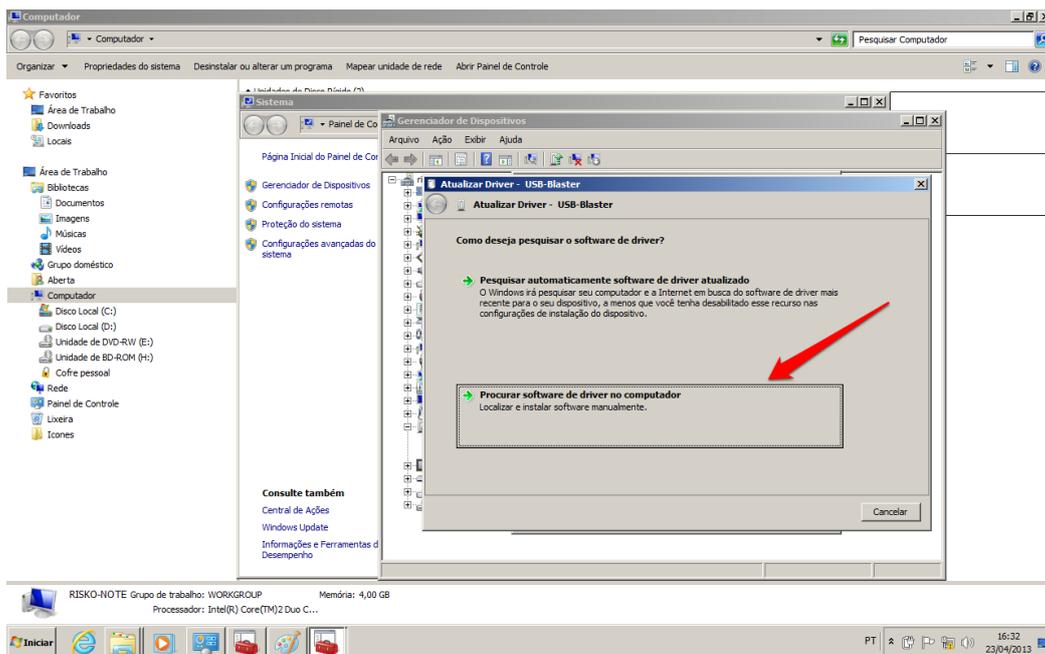
Figura 11 - Instalando o driver do PLDProg (Procedimento 2).

Na janela do “Gerenciador de Dispositivo”, vá em “Outros dispositivos”, selecione o dispositivo “USB-Blaster”, clique com o botão direito do mouse em cima dele e clique em “Atualizar Driver...” (Figura 12).



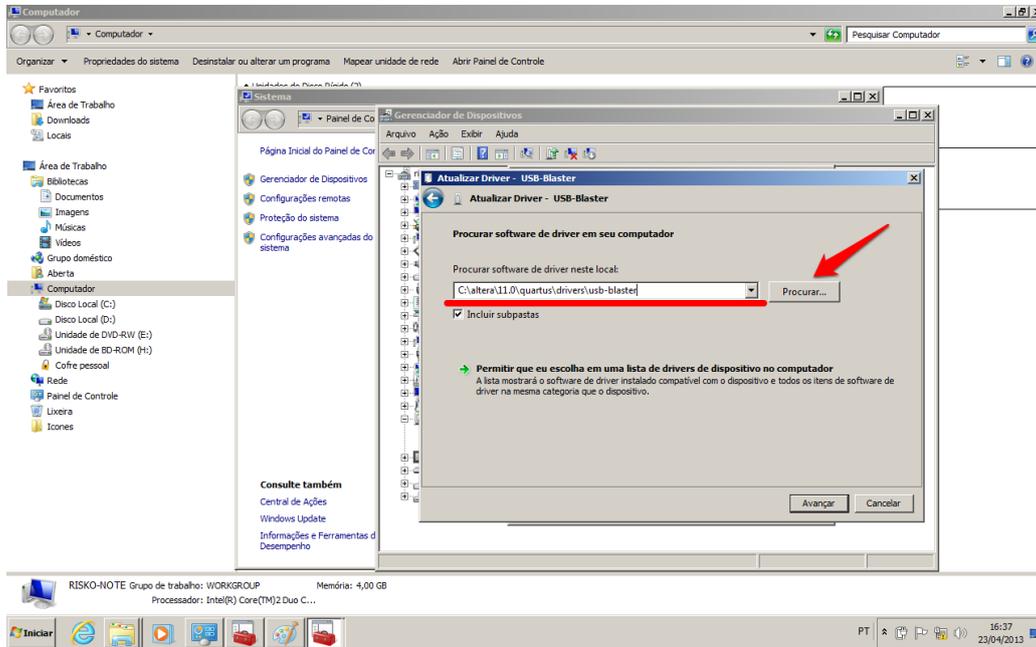
**Figura 12 - Instalando o driver do PLDProg (Procedimento 3).**

Na Janela a seguir, clique em “Procurar software de driver no computador” (Figura 13).



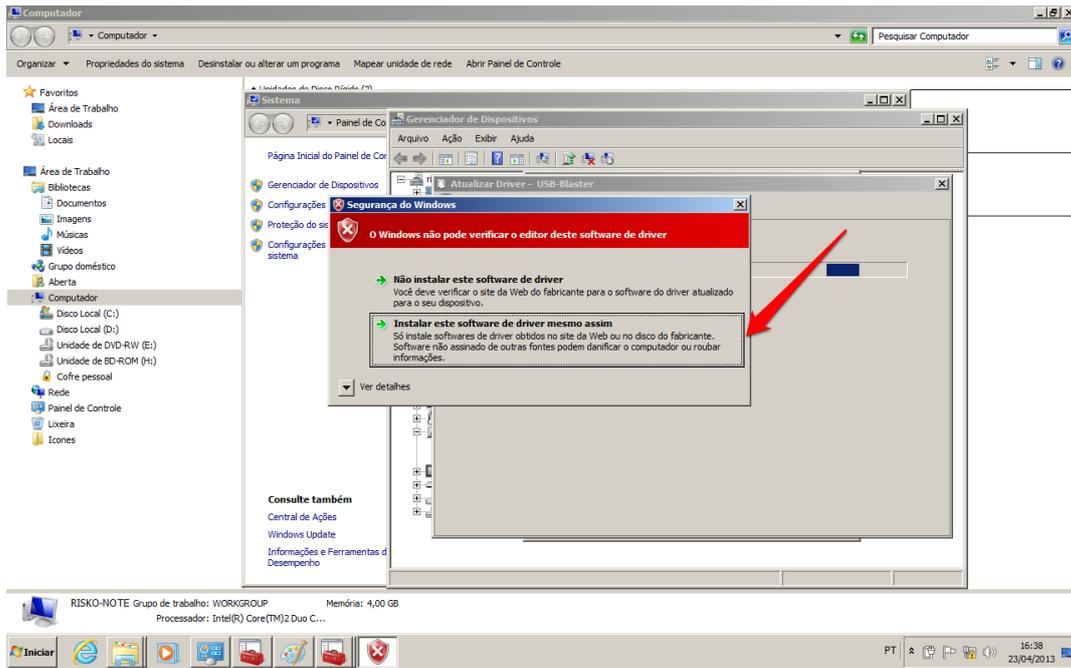
**Figura 13 - Instalando o driver do PLDProg (Procedimento 4).**

Na próxima etapa, busque a pasta da instalação do Quartus em seu computador clicando em “Procurar” e selecione o caminho como é mostrado na Figura XX. O caminho deve ser algo como “C:\altera\11.0\quartus\drivers\usb-blaster”. Em seguida, clique em avançar (Figura 14).



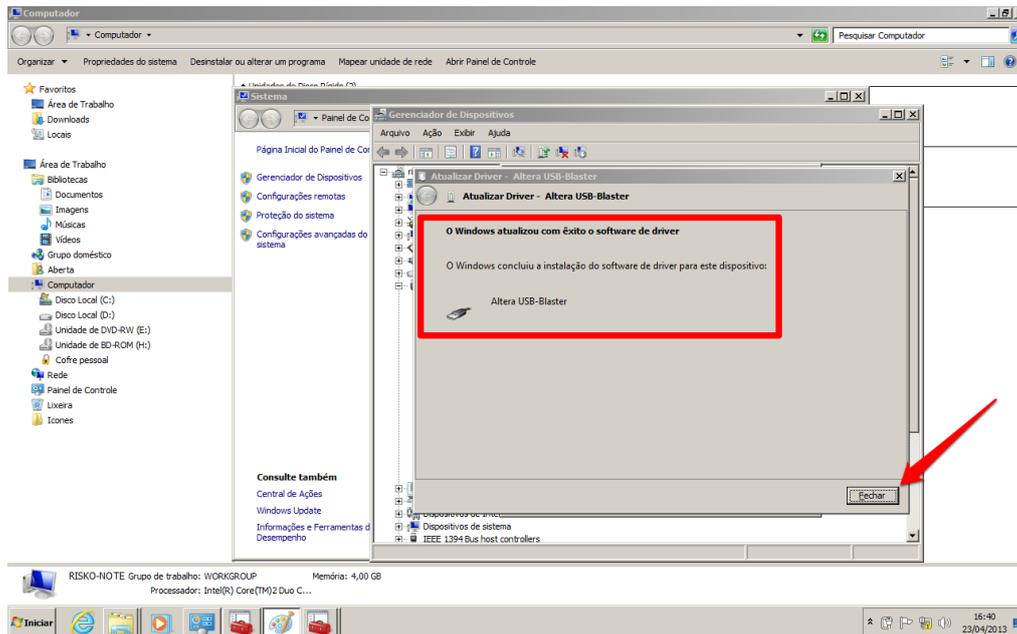
**Figura 14 - Instalando o driver do PLDProg (Procedimento 5).**

Após isso, o driver começará a ser instalado. Se aparecer uma janela de Segurança do Windows, clique em “Instalar este software de driver mesmo assim” (Figura 15).



**Figura 15 - Instalando o driver do PLDProg (Procedimento 6).**

Finalmente, aguarde a conclusão da instalação do driver e clique em Fechar (Figura 16).



**Figura 16 - Instalando o driver do PLDProg (Procedimento 7).**

Agora você já pode utilizar o PLDProg para gravar seus projetos no software Quartus II.

## 7 CONFIGURANDO O PLDPROG NO SOFTWARE QUARTUS II

Após conectar o PLDProg ao seu computador, siga as instruções abaixo para configurá-lo no software Quartus II:

- 1) Inicie o software Quartus II.
- 2) Abra o programador do Quartus II em menu “Ferramentas/Programmer”.
- 3) Clique em “Hardware Setup”, como mostra a Figura 17.

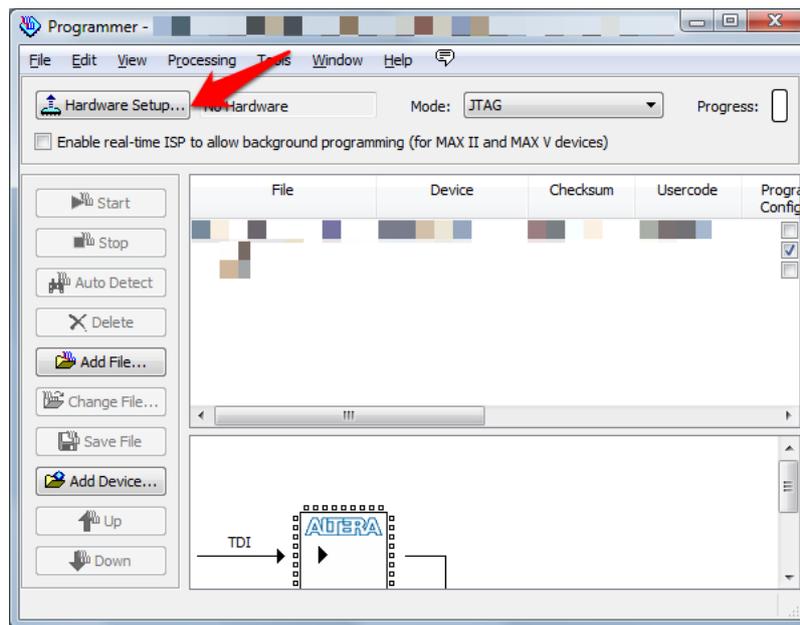
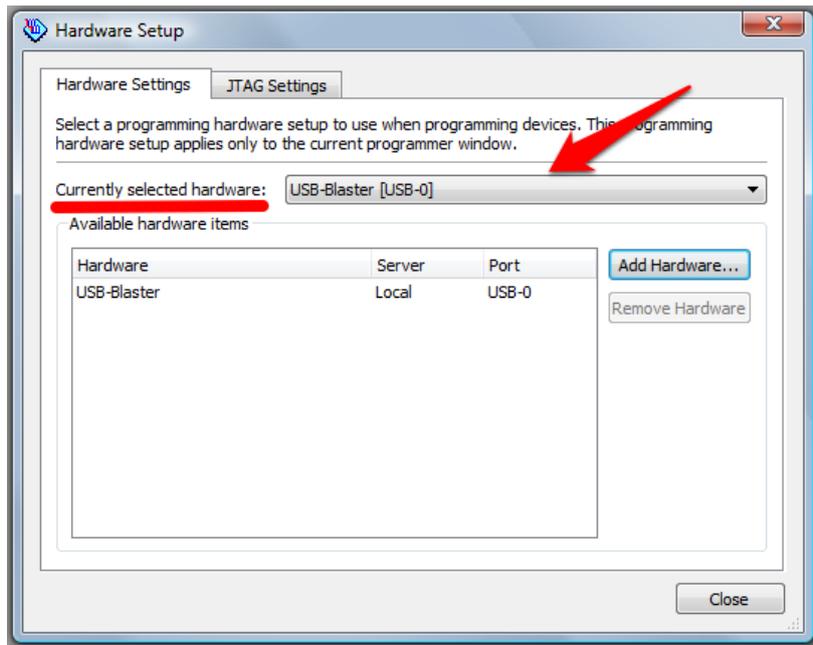


Figura 17 – Janela Programmer.

- 4) Na aba “Hardware Settings” em “Hardware Setup”, selecione em “Currently selected hardware” o item “USB-Blaster [USB-0]”, como mostra a Figura 18. Após isso, feche a janela.



**Figura 18 – Janela Hardware Setup.**

Agora é possível programar o dispositivo-alvo com os projetos desenvolvidos dentro do Quartus II. O PLDProg aceita apenas o modo JTAG para gravação. Este modo já vem configurado por padrão na janela Programmer, mostrado na Figura 17.

OBS.: Se o item USB-Blaster [USB-0] não estiver na lista, verifique se o hardware do PLDProg esteja conectado corretamente na USB do computador ou verifique a instalação correta do driver.

## 8 PLDPROG COMO INTERFACE JTAG DE COMUNICAÇÃO

O PLDProg, como interface de comunicação JTAG, apresenta-se útil em diversas aplicações de verificação e monitoração de dispositivos como PLDs e microcontroladores, ou de placas que integrem uma interface JTAG.

Mas a ALTERA disponibiliza também um excelente recurso de comunicação em sistemas baseados em seu processador NIOS II: o uso da interface JTAG para comunicação de dados, como uma UART (de “Universal Asynchronous Receiver Transmitter”), entre um computador PC e um sistema embarcado em FPGA.

Como uma demonstração simples das possibilidades de aplicação da interface JTAG em um sistema baseado em NIOS II, a Figura 19 mostra a imagem de uma placa de FPGA com um CI ALTERA Cyclone IV, interfaceada com um PC através de uma PLDProg, e conectada a um conjunto de periféricos básicos do **Projeto FPGA para Todos**. O objetivo do sistema, um exemplo didático relativamente simples, é monitorar o estado dos botões e chaves e controlar o estado dos leds na tela do computador, em uma janela de console simples.

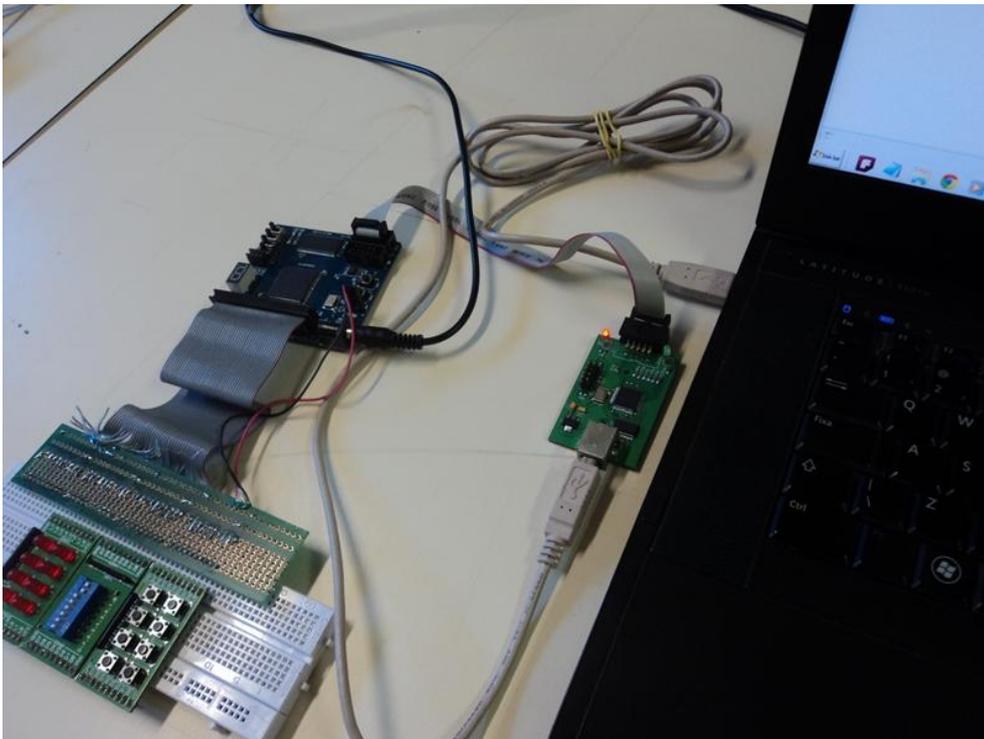


Figura 19 – Foto do PLDProg funcionando como interface de comunicação com o FPGA.

Apresentamos aqui, a título de introdução, alguns passos do processo de desenvolvimento desta aplicação. Maiores detalhes poderão ser encontrados no portal FPGA Para Todos.

A definição do sistema embarcado o FPGA começa, no Quartus II, com a execução da ferramenta QSys, na qual os blocos do sistema são selecionados e devidamente interconectados. A Figura 20 mostra uma captura de tela do programa, com o destaque da interface JTAG UART para comunicação. O sistema deve prever ainda duas portas de entrada de oito bits (para chaves e botões) e uma porta de saída de oito bits (para leds). O sistema é então gerado no QSys, incluindo um símbolo (arquivo “.bsf”).

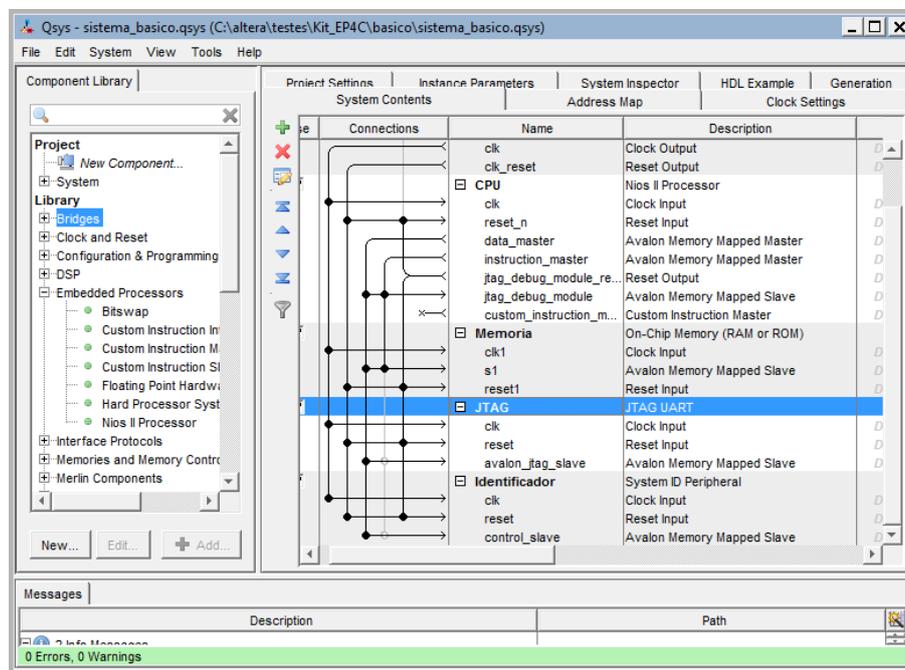
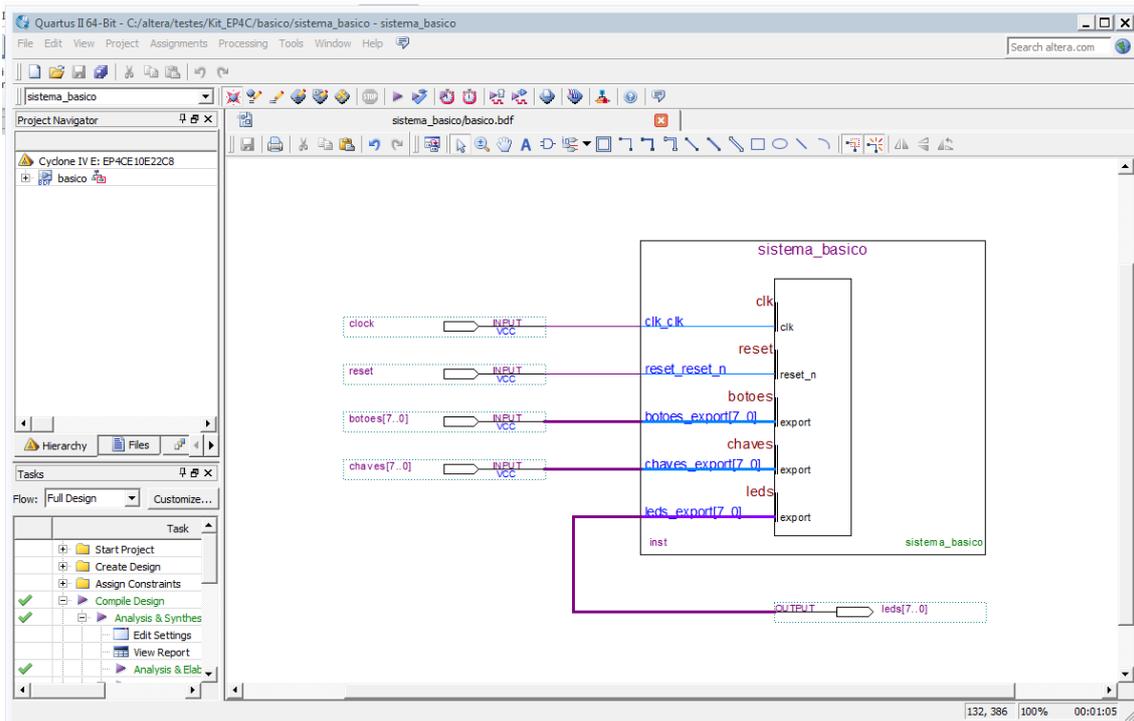


Figura 20 – Projeto básico de um processador NIOS II.

De volta ao ambiente do Quartus II, pode-se agora desenhar o diagrama esquemático com o símbolo do sistema projetado no Qsys e os pinos de saída e de entrada necessários, como mostrado na Figura 21. Nenhum bloco adicional será necessário neste projeto simples.

Concluído o diagrama esquemático, e após a atribuição de pinos para o sistema, de acordo com a estrutura a ser montada, processa-se uma compilação completa do projeto, e a gravação na placa de FPGA, com o uso do **PLDProg**.

O conjunto de “hardware” do sistema está concluído, e agora deve-se desenvolver o programa a ser gravado nele.



**Figura 21 – Diagrama esquemático do sistema completo.**

Para o desenvolvimento do conjunto de “software” do sistema, executa-se a ferramenta “Nios II SBT for Eclipse”. Um novo projeto deve ser construído, juntamente com um pacote de suporte ao sistema (BSP, de “Board Support Package”) previamente construído no QSys. Aqui, elaboramos um projeto simples, que a cada segundo deverá incrementar uma contagem, exibindo-a na tela do PC (o “console” do nosso projeto). Também, o estado das chaves e dos leds deverá ser informado. A listagem do código-fonte do projeto é mostrada a seguir.

```

/*
 *
 *  TESTE_BASICO
 *
 *  Exemplo simples de projeto no ambiente NIOS_II.
 *
 *  Este sistema fica, a cada 1s, lendo duas portas
 *  simples de entrada de oito bits, e escrevendo uma
 *  contagem de oito bits em uma porta de saída. O
 *  valor da contagem e as leituras das portas são
 *  exibidas no console (PC), enviadas via interface
 *  JTAG UART.
 *
 *  Por: Prof. Edson Melo, em maio de 2013.
 *
 */

#include "sys/alt_stdio.h"
#include "system.h"
#include "altera_avalon_pio_regs.h"

```

```

#include "unistd.h"
#include "teste_basico.h"

int contador = 0;

int main()
{
    alt_putstr("Projeto TESTE_BASICO\n");

    /* Event loop never exits. */
    while (1) {
        alt_printf("\n\tContagem agora: %xH\n", contador);
        saida (LEDS, contador);

        alt_printf ("\nEstado das chaves: %x", entrada(CHAVES) );
        alt_printf  ("\nEstado dos botoes %x", entrada(BOTOES) );

        // Prepara a proxima iteracao...
        usleep (1000000);
        alt_putstr("\n\n");
        contador++;

    }

    return 0;
}

```

As figuras seguintes mostram, respectivamente, a tela do ambiente de desenvolvimento do NIOS II durante o projeto, e uma captura da janela de console durante a execução do programa.

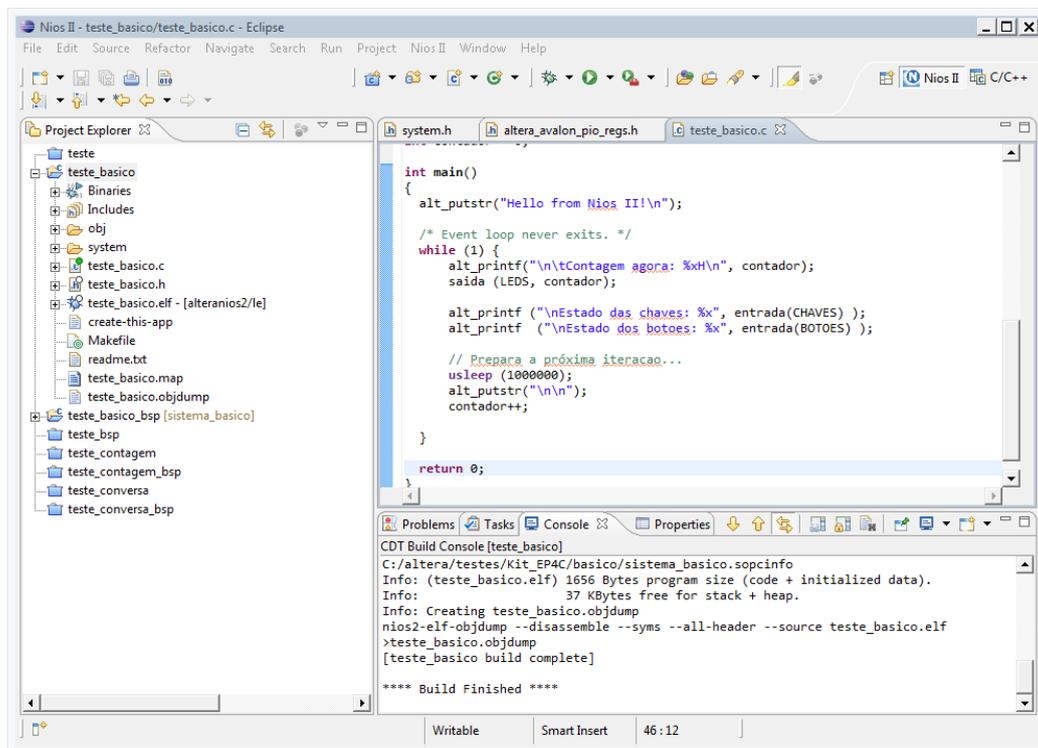


Figura 22 – Projeto simples “TESTE\_BASICO”.

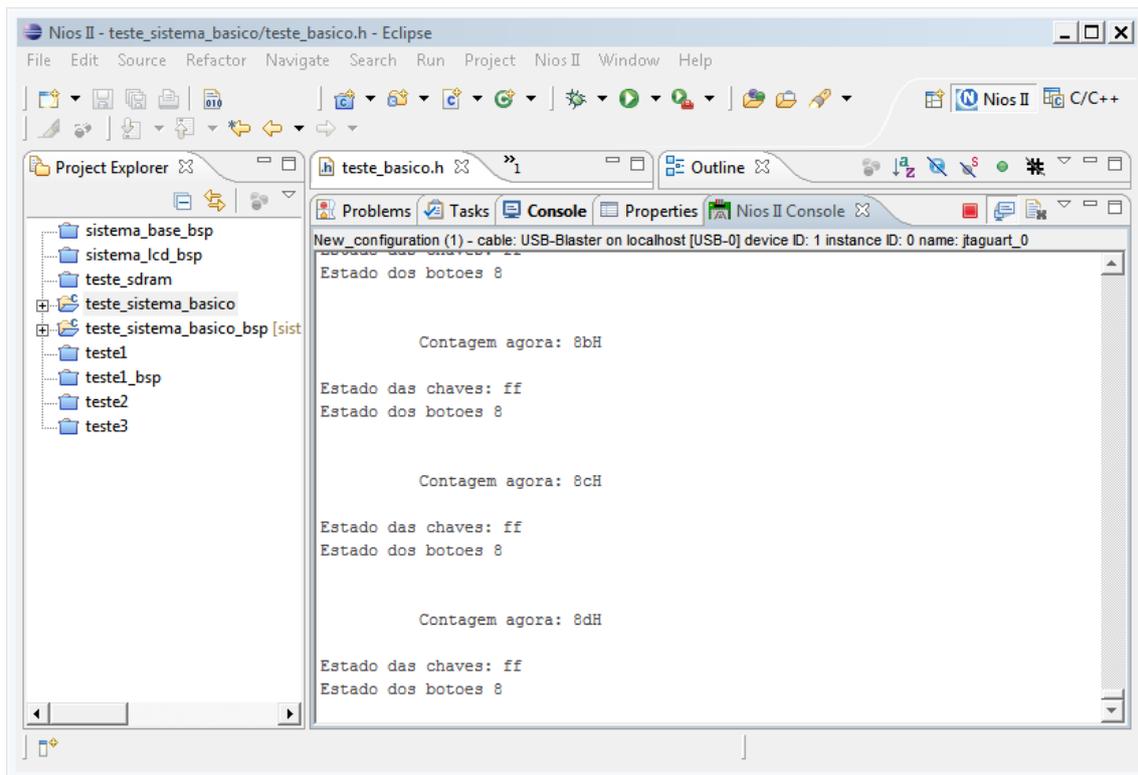


Figura 21 – Tela de execução do programa.

## **ANEXOS**

## ANEXO 1 - Lista de Materiais

| COMP.     | VALOR/<br>TIPO                     | DESCRIÇÃO                                                                                                                                                                                                                                                                                                                    | PLACA PLDPROG<br>COM TRADUTOR DE<br>TENSÃO |     |                   | PLACA PLDPROG<br>SEM TRADUTOR DE TENSÃO |     |                   | COTAÇÃO                                 |                      |       |
|-----------|------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------|-----|-------------------|-----------------------------------------|-----|-------------------|-----------------------------------------|----------------------|-------|
|           |                                    |                                                                                                                                                                                                                                                                                                                              | REF.<br>PLACA                              | QTD | PREÇO             | REF.<br>PLACA                           | QTD | PREÇO             | ONDE<br>COMPRAR                         | PREÇO<br>UNIT.       |       |
| Capacitor | 10uF/SMD<br>3216                   | CAPACITOR TANTALO 10UF 10V SMD                                                                                                                                                                                                                                                                                               | C1, C2                                     | 2   | R\$ 1,42          | C1, C2                                  | 2   | R\$ 1,42          | <a href="#">B45196H2106K109</a>         | R\$ 0,71             |       |
| Capacitor | 100nF/SMD<br>0805                  | CAPACIT CERAM MULT 100NF 50V S<br>CAPACITOR CERAMIC 0.1UF 50V X7R 0805;<br>Dielectric Characteristic:X7R; Capacitance:0.1uF;<br>Capacitance Tolerance:± 20%; Voltage<br>Rating:50V; Capacitor Case Style:0805; Capacitor<br>Mounting:SMD (KEMET) C0805C104M5RAC-TU                                                           | C3, C5-<br>C16                             | 13  | R\$ 0,65          | C3, C5-<br>C12                          | 9   | R\$ 0,45          | <a href="#">34C4152</a>                 | R\$ 0,05             |       |
| Capacitor | 10nF/SMD<br>0805                   | CAPACITOR CER. MULTICAMADA 10NF 50V<br>SMD<br>CAPACITOR 0805 10NF 50V (Multicomp)<br>U0805R103KCT                                                                                                                                                                                                                            | C4                                         | 1   | R\$ 0,04          | C4                                      | 1   | R\$ 0,04          | <a href="#">9406352</a>                 | R\$ 0,04             |       |
| CI        | MAX3378EEU<br>D+                   | CI LOGICO 14P TSSOP<br>TRANSLATOR LOW VOLT LEVEL QUAD 3378                                                                                                                                                                                                                                                                   | U4, U5                                     | 2   | R\$ 26,74         | ND                                      | 0   | R\$ 0,00          | <a href="#">MAX3378EEU<br/>D+</a>       | R\$ 13,37            |       |
| CI        | EPM3064ATC4<br>4-10N/TQFP44        | CPLD MAX7000S EPM3064ATC4-10N<br>CI SMD CPLD EEPROM, 64MACROCELLS,<br>TQFP44<br>IC PLD EEPROM 64 MACROCELL 10NS<br>TQFP44; CPLD Type:EEPROM; No. of<br>Macrocells:64; No. of I/O's:34; Series:MAX 3000A;<br>Propagation Delay:10ns; Global Clock Setup<br>Time:2.8ns; Frequency:222.2MHz; Supply Voltage<br>Range:3V to 3.6V | U1                                         | 1   | R\$ 17,56         | U1                                      | 1   | R\$ 17,56         | <a href="#">EPM3064ATC4<br/>4-10N</a>   | R\$ 17,56            |       |
| CI        | LM1117IMP-<br>3.3_NOPB/SO<br>T-223 | CI REGULADOR TENSÃO 4P SOT-223<br>IC LDO VOLT REG 3.3V 0.8A SOT-223-4; Primary<br>Input Voltage:5V; Output Voltage Fixed:3.3V;<br>Dropout Voltage Vdo:1.2V; No. of Pins:4; Output<br>Current:800mA; Operating Temperature Range:-<br>40°C to +125°C; MSL:MSL 1 - Unlimited (National<br>Semiconductor)                       | U2                                         | 1   | R\$ 2,39          | U2                                      | 1   | R\$ 2,39          | <a href="#">LM1117IMP-<br/>3.3/NOBP</a> | R\$ 2,39             |       |
| CI        | FT245RL/SSO<br>P-28                | CI SMD 28P SSOP<br>IC USB FIFO INTERFACE SSOP-28; Data<br>Rate:1Mbps; Supply Voltage Range:1.8V to 5.25V<br>3.3V to 5.25V; Operating Temperature Range:-<br>40°C to +85°C; Digital IC Case Style:SSOP; No. of<br>Pins:28; Filter Terminals:SMD; IC Generic<br>Number:245                                                     | U3                                         | 1   | R\$ 14,77         | U3                                      | 1   | R\$ 14,77         | <a href="#">91K9920</a>                 | R\$ 14,77            |       |
| Cristal   | 24MHz/SMD                          | CRISTAL OSCILADOR SMD 24MHZ<br>CRYSTAL OSCILLATOR SMD 24MHZ                                                                                                                                                                                                                                                                  | X2                                         | 1   | R\$ 6,68          | X2                                      | 1   | R\$ 6,68          | <a href="#">LF<br/>SPX0018541</a>       | R\$ 6,68             |       |
| Conector  | 4 VIAS/USB                         | CONECTOR USB 4 VIAS<br>USB CONNECTOR RECEPTACLE 4POS THD;<br>Connector Type:USB; Gender:Receptacle; No. of<br>Contacts:4; Contact Termination:Through Hole<br>Right Angle; Connector Mounting:PCB; Contact<br>Plating:Gold; Contact Material:Copper Alloy FCI<br>61729-0010BLF                                               | CON1                                       | 1   | R\$ 3,53          | CON1                                    | 1   | R\$ 3,53          | <a href="#">62K6482</a>                 | R\$ 3,53             |       |
| Conector  | 2X5/IDC                            | Conector IDC 2X5 Macho 90 graus                                                                                                                                                                                                                                                                                              | CON2                                       | 1   | R\$ 0,40          | CON2                                    | 1   | R\$ 0,40          | -                                       | R\$ 0,40             |       |
| Header    | 2X5/HDR                            | Barra pinos para os conectores JTAG                                                                                                                                                                                                                                                                                          | CON3                                       | 1   | R\$ 0,50          | CON3                                    | 1   | R\$ 0,50          | -                                       | R\$ 0,50             |       |
| Indutor   | 330OHM/FERR<br>ITE SMD 0805        | INDUTOR FERRITE 330OHM 0.09OHM FERRITE<br>BEAD 0805 CASE 330OHM                                                                                                                                                                                                                                                              | L1                                         | 1   | R\$ 0,20          | L1                                      | 1   | R\$ 0,20          | <a href="#">BLM21PG331S<br/>N1D</a>     | R\$ 0,20             |       |
| LED       | 3mm/PTH                            | LED 3MM VERM 50MCD<br>LED 3MM RED (KINGBRIGHT)                                                                                                                                                                                                                                                                               | D1                                         | 1   | R\$ 0,27          | D1                                      | 1   | R\$ 0,27          | <a href="#">L-934ID</a>                 | R\$ 0,27             |       |
| LED       | 3mm/PTH                            | LED VERDE LENTE DIFUSA 3MM 32MCD<br>LED 3MM GREEN (KINGBRIGHT)                                                                                                                                                                                                                                                               | D2                                         | 1   | R\$ 0,23          | D2                                      | 1   | R\$ 0,23          | <a href="#">L-934GD</a>                 | R\$ 0,23             |       |
| Resistor  | 0R0/SMD 0805                       | RESISTOR FILME ESP 0R0 100VDC 0.1W<br>RESISTOR 0805 0R0 0.1W (MULTICOMP)                                                                                                                                                                                                                                                     | R6, R7                                     | 2   | R\$ 0,02          | R6, R7                                  | 2   | R\$ 0,02          | <a href="#">MC 0.1W 0805<br/>0R</a>     | R\$ 0,01             |       |
| Resistor  | 220R/SMD<br>0805                   | RESISTOR FILME ESP 220R 100V 0 SMD<br>RESISTOR 0805 220R                                                                                                                                                                                                                                                                     | R1, R8                                     | 2   | R\$ 0,02          | R1, R8                                  | 2   | R\$ 0,02          | <a href="#">9332804</a>                 | R\$ 0,01             |       |
| Resistor  | 10K/SMD 0805                       | RESISTOR FILME ESP 10K 100VDC 0.1W SMD<br>RESISTOR 0805 10K 0.1W 1%                                                                                                                                                                                                                                                          | R2-R5                                      | 4   | R\$ 0,04          | R2-R5,<br>R16-R22                       | 11  | R\$ 0,11          | <a href="#">9332391</a>                 | R\$ 0,01             |       |
| Resistor  | 100R/SMD<br>0805                   | RESISTOR FILME ESP 100R 100V<br>RESISTOR 0805 100R 0.1W 1%                                                                                                                                                                                                                                                                   | ND                                         | 0   | R\$ 0,00          | R9-R15                                  | 7   | R\$ 0,07          | <a href="#">9332375</a>                 | R\$ 0,01             |       |
| Resistor  | 2k2/SMD 0805                       | RESISTOR FILME ESP 2K2 100V 0.1W SMD<br>RESISTOR 0805 2K2 0.1W 1%                                                                                                                                                                                                                                                            | R9                                         | 1   | R\$ 0,01          | ND                                      | 0   | R\$ 0,00          | <a href="#">9332612</a>                 | R\$ 0,01             |       |
|           |                                    |                                                                                                                                                                                                                                                                                                                              |                                            |     | PREÇO<br>ESTIMADO | R\$ 75,47                               |     | PREÇO<br>ESTIMADO | R\$ 48,66                               | PREÇO UNIT.<br>TOTAL | 60,75 |

## ANEXO 2 - Código VHDL da Placa

O projeto do CPLD da placa PLDProg é de autoria de Kolja Waschk, que o disponibilizou como “open source”, e pode ser encontrado na internet, na página de endereço: [http://ixo-jtag.sourceforge.net/#usb\\_jtag](http://ixo-jtag.sourceforge.net/#usb_jtag).

A listagem do código é apresentada abaixo.

```
-----
-- Serial/Parallel converter, interfacing JTAG chain with FTDI FT245BM
-----
-- Copyright (C) 2005-2007 Kolja Waschk, ixo.de
-----
-- This code is part of usbjtag. usbjtag is free software; you can redistribute
-- it and/or modify it under the terms of the GNU General Public License as
-- published by the Free Software Foundation; either version 2 of the License,
-- or (at your option) any later version. usbjtag is distributed in the hope
-- that it will be useful, but WITHOUT ANY WARRANTY; without even the implied
-- warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
-- GNU General Public License for more details. You should have received a
-- copy of the GNU General Public License along with this program in the file
-- COPYING; if not, write to the Free Software Foundation, Inc., 51 Franklin
-- St, Fifth Floor, Boston, MA 02110-1301 USA
-----

LIBRARY ieee;
USE ieee.std_logic_1164.all;
USE ieee.std_logic_unsigned.all;

ENTITY jtag_logic IS
  PORT
  (
    CLK : IN STD_LOGIC;           -- external 24/25 MHz oscillator
    nRXF : IN STD_LOGIC;          -- FT245BM nRXF
    nTXE : IN STD_LOGIC;          -- FT245BM nTXE
    B_TDO : IN STD_LOGIC;         -- JTAG input: TDO, AS/PS input: CONF_DONE
    B_ASDO : IN STD_LOGIC;        -- AS input: DATAOUT, PS input: nSTATUS
    B_TCK : BUFFER STD_LOGIC;     -- JTAG output: TCK to chain, AS/PS DCLK
    B_TMS : BUFFER STD_LOGIC;    -- JTAG output: TMS to chain, AS/PS nCONFIG
    B_NCE : BUFFER STD_LOGIC;    -- AS output: nCE
    B_NCS : BUFFER STD_LOGIC;    -- AS output: nCS
    B_TDI : BUFFER STD_LOGIC;    -- JTAG output: TDI to chain, AS: ASDI, PS: DATA0
    B_OE : BUFFER STD_LOGIC;     -- LED output/output driver enable
    nRD : OUT STD_LOGIC;         -- FT245BM nRD
    WR : OUT STD_LOGIC;          -- FT245BM WR
    D : INOUT STD_LOGIC_VECTOR(7 downto 0) -- FT245BM D[7..0]
  );
END jtag_logic;
ARCHITECTURE spec OF jtag_logic IS

  -- There are exactly 16 states. If this is encoded using 4 bits, there will
  -- be no unknown/undefined state. The host will send us 64 times "0" to move
  -- the state machine to a known state. We don't need a power-on reset.

  TYPE states IS
  (
    wait_for_nRXF_low,
    set_nRD_low,
    keep_nRD_low,
    latch_data_from_host,
    set_nRD_high,
    bits_set_pins_from_data,
    bytes_set_bitcount,
    bytes_get_tdo_set_tdi,
    bytes_clock_high_and_shift,
    bytes_keep_clock_high,
    bytes_clock_finish,
    wait_for_nTXE_low,
    set_WR_high,
    output_enable,
    set_WR_low,
  );

```

```

        output_disable
    );

    ATTRIBUTE ENUM_ENCODING: STRING;
    ATTRIBUTE ENUM_ENCODING OF states: TYPE IS
        "0000 0001 0010 0011 0100 0101 0110 0111 1000 1001 1010 1011 1100 1101 1110 1111";

    SIGNAL carry: STD_LOGIC;
    SIGNAL do_output: STD_LOGIC;
    SIGNAL iosifter: STD_LOGIC_VECTOR(7 DOWNTO 0);
    SIGNAL bitcount: STD_LOGIC_VECTOR(8 DOWNTO 0);
    SIGNAL state, next_state: states;

BEGIN
    sm: PROCESS(nRXF, nTXE, state, bitcount, iosifter, do_output)

    BEGIN
        CASE state IS

            -- ===== INPUT

            WHEN wait_for_nRXF_low =>
                IF nRXF='0' THEN
                    next_state <= set_nRD_low;
                ELSE
                    next_state <= wait_for_nRXF_low;
                END IF;

            WHEN set_nRD_low =>
                next_state <= keep_nRD_low;

            WHEN keep_nRD_low =>
                next_state <= latch_data_from_host;

            WHEN latch_data_from_host =>
                next_state <= set_nRD_high;

            WHEN set_nRD_high =>
                IF NOT (bitcount(8 DOWNTO 3) = "000000") THEN
                    next_state <= bytes_get_tdo_set_tdi;
                ELSIF iosifter(7) = '1' THEN
                    next_state <= bytes_set_bitcount;
                ELSE
                    next_state <= bits_set_pins_from_data;
                END IF;

            WHEN bytes_set_bitcount =>
                next_state <= wait_for_nRXF_low;

            -- ===== BIT BANGING

            WHEN bits_set_pins_from_data =>
                IF iosifter(6) = '0' THEN
                    next_state <= wait_for_nRXF_low; -- read next byte from host
                ELSE
                    next_state <= wait_for_nTXE_low; -- output byte to host
                END IF;

            -- ===== BYTE OUTPUT (SHIFT OUT 8 BITS)

            WHEN bytes_get_tdo_set_tdi =>
                next_state <= bytes_clock_high_and_shift;

            WHEN bytes_clock_high_and_shift =>
                next_state <= bytes_keep_clock_high;

            WHEN bytes_keep_clock_high =>
                next_state <= bytes_clock_finish;

            WHEN bytes_clock_finish =>
                IF NOT (bitcount(2 DOWNTO 0) = "111") THEN
                    next_state <= bytes_get_tdo_set_tdi; -- clock next bit
                ELSIF do_output = '1' THEN
                    next_state <= wait_for_nTXE_low; -- output byte to host
                ELSE
                    next_state <= wait_for_nRXF_low; -- read next byte from host
            END IF;
        END CASE;
    END PROCESS;
END;

```

```

        END IF;

-- ===== OUTPUT BYTE TO HOST

WHEN wait_for_nTXE_low =>
    IF nTXE = '0' THEN
        next_state <= set_WR_high;
    ELSE
        next_state <= wait_for_nTXE_low;
    END IF;

WHEN set_WR_high =>
    next_state <= output_enable;

WHEN output_enable =>
    next_state <= set_WR_low;

WHEN set_WR_low =>
    next_state <= output_disable;

WHEN output_disable =>
    next_state <= wait_for_nRXF_low; -- read next byte from host

WHEN OTHERS =>
    next_state <= wait_for_nRXF_low;

    END CASE;
END PROCESS sm;

out_sm: PROCESS(CLK)

BEGIN
    IF CLK = '1' AND CLK'event THEN

        IF state = set_nRD_low OR state = keep_nRD_low OR state =
latch_data_from_host THEN
            nRD <= '0';
        ELSE
            nRD <= '1';
        END IF;

        IF state = latch_data_from_host THEN
            iosifter(7 DOWNTO 0) <= D;
        END IF;

        IF state = set_WR_high OR state = output_enable THEN
            WR <= '1';
        ELSE
            WR <= '0';
        END IF;

        IF state = output_enable OR state = set_WR_low THEN
            D <= iosifter(7 DOWNTO 0);
        ELSE
            D <= "ZZZZZZZZ";
        END IF;

        IF state = bits_set_pins_from_data THEN
            B_TCK <= iosifter(0);
            B_TMS <= iosifter(1);
            B_NCE <= iosifter(2);
            B_NCS <= iosifter(3);
            B_TDI <= iosifter(4);
            B_OE <= iosifter(5);
            iosifter <= "000000" & B_ASDO & B_TDO;
        END IF;

        IF state = bytes_set_bitcount THEN
            bitcount <= iosifter(5 DOWNTO 0) & "111";
            do_output <= iosifter(6);
        END IF;

        IF state = bytes_get_tdo_set_tdi THEN
            IF B_NCS = '1' THEN
                carry <= B_TDO; -- JTAG mode (nCS=1)
            ELSE
                carry <= B_ASDO; -- Active Serial mode (nCS=0)
            END IF;
        END IF;
    END IF;
END PROCESS out_sm;

```

```
        END IF;
        B_TDI <= iosifter(0);
        bitcount <= bitcount - 1;
    END IF;

    IF state = bytes_clock_high_and_shift OR state = bytes_keep_clock_high THEN
        B_TCK <= '1';
    END IF;

    IF state = bytes_clock_high_and_shift THEN
        iosifter <= carry & iosifter(7 DOWNT0 1);
    END IF;

    IF state = bytes_clock_finish THEN
        B_TCK <= '0';
    END IF;

    state <= next_state;

    END IF;
END PROCESS out_sm;

END spec;
```

### **ANEXO 3 - Programação do FT245**

De modo a ser reconhecida como uma USBBlaster<sup>®</sup> pelo sistema operacional do computador, o CI FT245 da placa deve ter a sua EEPROM gravada com os mesmos códigos de identificação (VendorID e ProductID) do cabo original da ALTERA. Este processo é executado facilmente com o aplicativo MProg, ou o FTProg, da FTDI ([www.ftdichip.com](http://www.ftdichip.com)). Para simplificar a tarefa, pode-se obter do repositório do portal fpgaparatodos.com.br um arquivo pronto de configuração da EEPROM, bem como orientação para o processo de programação.

## REFERÊNCIAS

[1] ALTERA. **Embedded Peripherals IP – User Guide**. Capítulo 6 - JTAG UART Core. 6-1 a 6-11 p. Disponível em: <[http://www.altera.com/literature/ug/ug\\_embedded\\_ip.pdf](http://www.altera.com/literature/ug/ug_embedded_ip.pdf)>. Acesso em: 29 de maio de 2013.

[2] ALTERA. **Design Debugging Using the SignalTap II Logic Analyzer**. Disponível em: <[http://www.altera.com/literature/hb/qts/qts\\_qii53009.pdf](http://www.altera.com/literature/hb/qts/qts_qii53009.pdf)>. Acesso em: 29 de maio de 2013.

[3] **.Boundary Scan, JTAG, IEEE 1149.1 Tutorial**. Disponível em: <[http://www.radio-electronics.com/info/t\\_and\\_m/boundaryscan/jtag-ieee-1149-basics-tutorial.php](http://www.radio-electronics.com/info/t_and_m/boundaryscan/jtag-ieee-1149-basics-tutorial.php)>. Acesso em: 29 de maio de 2013.

[4] **OpenJtag Project**. Disponível em: <<http://www.openjtag.org/>>. Acesso em: 29 de maio de 2013.

[5] **USB JTAG Adapter**. Disponível em: <[http://ixo-jtag.sourceforge.net/#usb\\_jtag](http://ixo-jtag.sourceforge.net/#usb_jtag)>. Acesso em: 29 de maio de 2013.

[6] ALTERA. **USB-Blaster Download Cable – User Guide**. Disponível em: <[http://www.altera.com/literature/ug/ug\\_usb\\_blstr.pdf](http://www.altera.com/literature/ug/ug_usb_blstr.pdf)>. Acesso em: 29 de maio de 2013.

[7] FTDI. **FT245R - USB FIFO IC**. Disponível em: <<http://www.ftdichip.com/Products/ICs/FT245R.htm>>. Acesso em: 29 de maio de 2013.

[8] ALTERA. **Low-Cost MAX 3000A CPLDs**. Disponível em: <<http://www.altera.com/devices/cpld/max3k/m3k-index.html>>. Acesso em: 29 de maio de 2013.