

SISTEMA DIDÁTICO DE EXPANSÃO DE E/S PARA ARDUINO, VIA SPI E COM CPLD**João Marcos de Aguiar¹, Marcos Vinicius Leal da Silva,² Fco. Édson N. de Mélo³**¹UFSC/DEEL/Campus Florianópolis/Email: joao.een@gmail.com²Instituto Federal de Santa Catarina/DAELN/Campus Florianópolis/Email: marcos.leal93@gmail.com³Instituto Federal de Santa Catarina/DAMM/Campus Florianópolis/Email: edson@fpgaparatodos.com.br

Resumo: *Numerosas aplicações de sistemas digitais demandam a expansão de dispositivos de entrada e saída (E/S) de microprocessadores ou microcontroladores. A abordagem mais trivial, usar um dispositivo com maior quantidade de periféricos nem sempre é viável, pois em algumas situações implica em aumento injustificável de custo. A interface SPI tem se tornado um dos padrões mais adotados em tais aplicações, devido à sua simplicidade e fácil implementação. Numerosos microcontroladores integram um dispositivo de comunicação SPI, e alguns circuitos integrados digitais simples podem interagir com o padrão. Dispositivos lógicos programáveis (PLDs), por outro lado, oferecem uma solução de projeto digital importante como alternativa ou complementarmente aos microcontroladores. PLDs têm a capacidade de processamento concorrente de sinais, enquanto microcontroladores são sequenciais. O artigo abordará a utilização de um CPLD (dispositivo lógico programável complexo) para fazer a expansão de pinos de E/S de um uCduíno (módulo didático de microcontrolador com arquitetura similar à do padrão Arduino), como uma interface SPI. O projeto foi desenvolvido em VHDL (uma linguagem específica para descrição de hardware), no ambiente de desenvolvimento Quartus II, da ALTERA. O objetivo deste estudo é estabelecer o domínio do tema pelo grupo de pesquisa, visando à preparação de material de apoio a outras pesquisas e didático em geral, em consonância com as necessidades do Projeto FPGA para Todos. Os resultados obtidos foram a produção de uma página explicativa divulgada na internet, a utilização desse sistema em outras pesquisas do IFSC e do próprio projeto FPGA para Todos. Ao final do desenvolvimento, os objetivos propostos inicialmente foram satisfatoriamente atingidos.*

Palavras-Chave: SPI, CPLD, Arduino, Lógica Programável, Expansão de E/S.

1 INTRODUÇÃO

Microcontroladores são dispositivos padrão na indústria para aplicação em diversas categorias de projetos na área de Eletrônica Digital, devido aos numerosos recursos integrados à maioria destes componentes, facilidade de uso e baixo custo. Entre as novas soluções propostas para o desenvolvimento com microcontroladores, destaca-se a plataforma Arduino, que une uma placa (solução de hardware) e um ambiente de desenvolvimento aberto (solução de software). O padrão Arduino é caracterizado por uma biblioteca com funções que simplificam o desenvolvimento de projetos e o seu aprendizado, tornando-o um excelente recurso didático para o ensino de tecnologia.

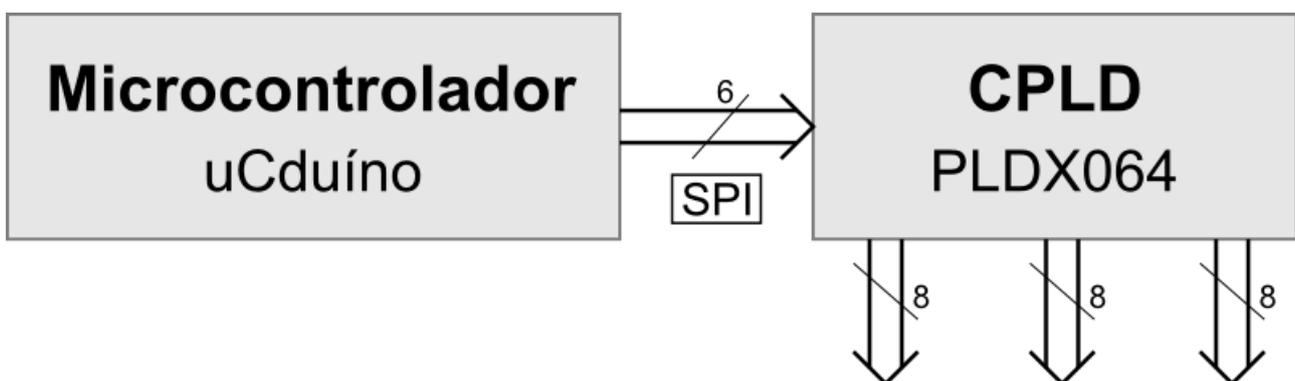
Por outro lado, dispositivos de lógica programável (PLDs, sigla em inglês) são também uma tecnologia largamente adotada em novos projetos. PLDs têm a capacidade de executar uma grande quantidade de processos lógicos em modo concorrente (simultaneamente), enquanto microcontroladores executam operações sequencialmente, uma instrução após outra. Ainda, PLDs existem em versões com grande quantidade de pinos, com maiores ou menores quantidades de recursos de lógica, permitindo o

desenvolvimento de projetos bastante complexos. Um CPLD (Complex Programmable Logic Device) é um PLD que se destaca por ter um alto desempenho, baixos custo e consumo de energia.

A integração entre sistemas microcontrolados e de lógica programável pode, por vezes, ser a solução adequada para um projeto. Um problema comumente encontrado é a necessidade de expansão dos recursos de entrada e saída (E/S) de um microcontrolador. Shulte (2011, p.1) descreve o interfaceamento com um microcontrolador, para expansão das suas capacidades de E/S, como uma das cinco principais aplicações práticas de um CPLD.

Como um recurso didático, e para aplicação em outras pesquisas, o projeto FPGA para Todos, uma iniciativa do CNPq e do IFSC que tem por objetivo a disseminação da tecnologia de lógica programável (MELO, 2011, p.5), estudou e implementou sistemas integrando um módulo uCduíno, kit desenvolvido pelo professor Édson Melo, baseado na Arduino Duemilanove (MELO, 2011, p.6), e um módulo de CPLD, desenvolvido pelo próprio projeto (MELO, 2011, p.4). Na aplicação descrita neste artigo, o kit CPLD atua como uma expansão dos pinos de saída do microcontrolador, gerando 24 sinais de saída a partir de 6 linhas de comunicação entre os módulos, como ilustra a Figura 01. A interface de comunicação utilizada entre os kits foi a SPI, que se caracteriza por ser um protocolo simples de ser implementado.

Figura 01: Expansão dos pinos de saída usando o protocolo SPI



Fonte: Projeto FPGA para Todos (fpgaparatodos.com.br)

O desenvolvimento desse projeto visa a demonstrar uma aplicação da lógica programável, na integração de sistemas microcontrolados, explorando as suas possibilidades e promovendo a iniciação tecnológica dos bolsistas envolvidos.

2 METODOLOGIA

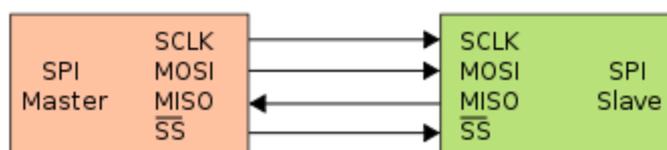
O sistema foi desenvolvido em quatro etapas: Inicialmente, analisou-se o padrão de comunicação SPI e decidiu-se pelo modo de operação a ser utilizado. Em seguida, estudaram-se as funções disponíveis na biblioteca do Arduino para o uso da comunicação SPI. A implementação do protocolo SPI de comunicação no CPLD compatível com o Arduino foi o terceiro passo. Finalmente, integrou-se o microcontrolador com o CPLD, compondo o sistema completo.

2.1 Protocolo SPI de Comunicação

O protocolo SPI define um enlace de comunicação síncrona *full-duplex*, em que um dispositivo mestre comanda o processo de comunicação com um ou mais dispositivos escravos. A Figura apresenta uma configuração bastante simples de comunicação SPI entre dois dispositivos, destacando os sinais envolvidos (WIKIPEDIA, 2012):

- SCLK: Sinal de clock da comunicação, emitido pelo mestre;
- MOSI (*Master Output, Slave Input*): Sinal para envio de dados do mestre para o escravo;
- MISO (*Master Input, Slave Output*): Sinal para envio de dados do escravo para o mestre;
- SS\ (*Slave Select*): Sinal de habilitação do escravo.

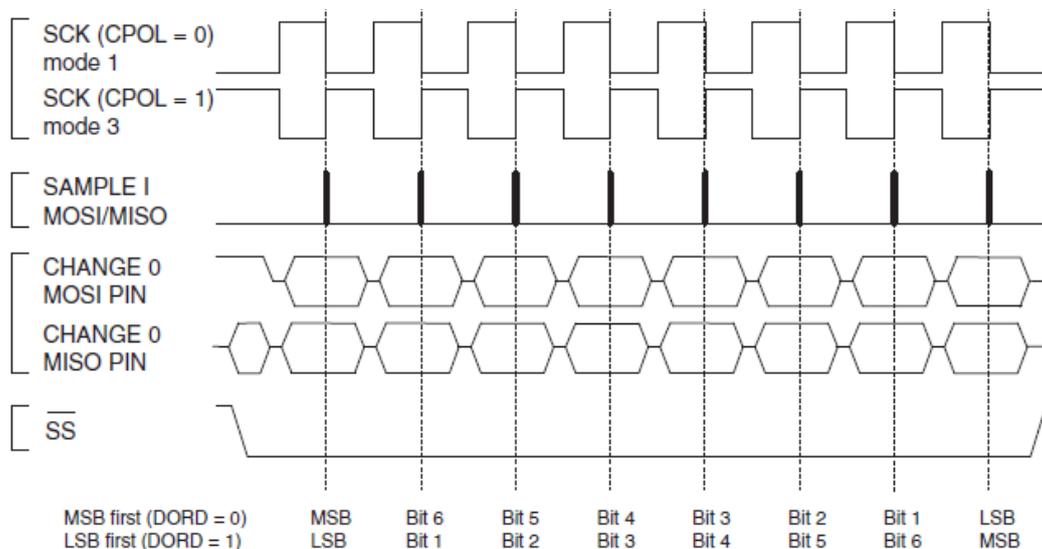
Figura 02: Interface de comunicação SPI



Fonte: Wikipedia(WIKIPEDIA,2012)

O padrão SPI define o comportamento das linhas de modo a garantir a comunicação de dados, em quatro possíveis modos de operação. A figura abaixo mostra as formas de onda para os modos de operação 3 e 1 (ATMEL, 2012). O modo escolhido para a transmissão foi o 3, com a transmissão dos bits da palavra em ordem crescente (iniciando-a pelo bit menos significativo).

Figura 03: Formas de onda da comunicação SPI.



Fonte: Folha de dados do microcontrolador ATmega328 (ATMEL, 2012)

2.2 Padrão SPI no Arduino

O Arduino provê uma biblioteca que simplifica o desenvolvimento de aplicações envolvendo a comunicação SPI. Para a configuração do modo de operação, são disponíveis as funções *SPI.setBitOrder()*, *SPI.setClockDivider()* e *SPI.setDataMode()*. Para início e fim de comunicação, utilizam-se as funções *SPI.begin()* e *SPI.end()*. Finalmente, a transmissão e recepção (simultâneas) de dados é comandada pela função *SPI.transfer()* (ARDUINO, 2012).

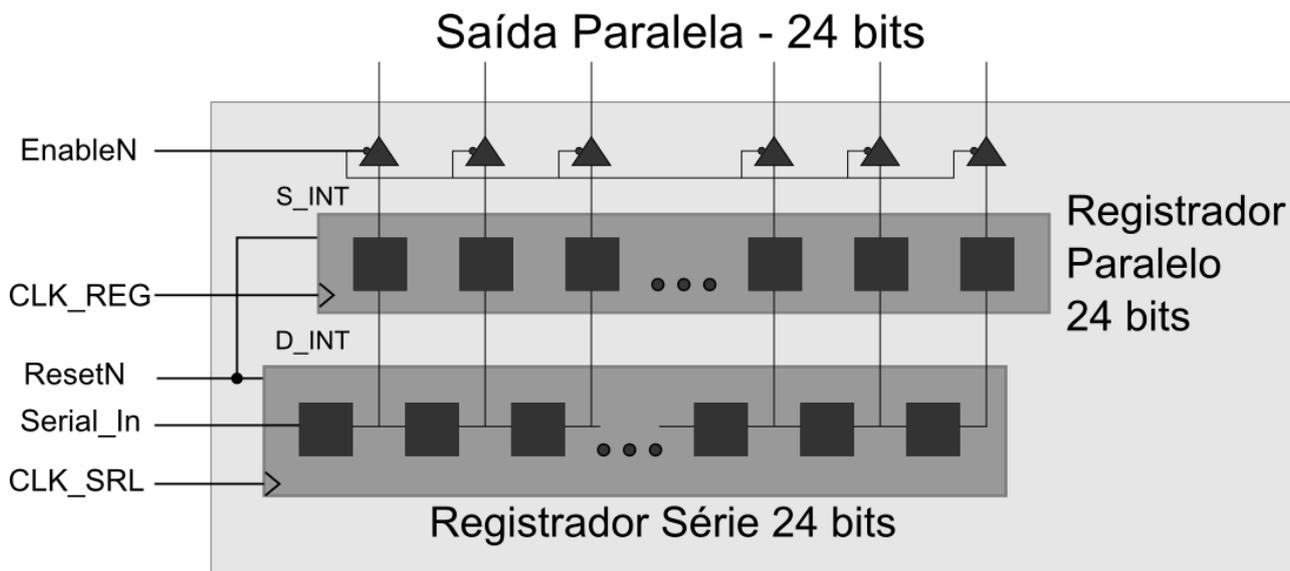
Para aplicação neste projeto, decidiu-se por utilizar velocidade de comunicação de 4 Mbits/s, executando a leitura no momento da subida do clock e o envio na descida do clock (modo 3) e os dados são enviados do LSB (“least significant bit”, ou seja, bit menos significativo) para o MSB (“most significant bit”, ou bit mais significativo).

2.3 Implementação de Recepção SPI em CPLD

O expensor foi desenvolvido como atividade do projeto FPGA para Todos, utilizando o kit de CPLD como base. O CPLD presente no kit é um EPM7064SLC44-10 da família MAX7000S, da ALTERA®. Além do CPLD, o kit também possui geradores de *clock* de alta e baixa frequências e 32 pinos de E/S.

A descrição do expansor foi feita no Quartus® II em linguagem VHDL (*VHSIC Hardware Description Language*), de forma que o circuito implementado funcionasse como um registrador de deslocamento de 24 bits. Além do registrador de deslocamento, o sistema também possui um reset e suas saídas podem ser habilitadas ou não. A Figura 04 apresenta um diagrama de blocos do sistema implementado.

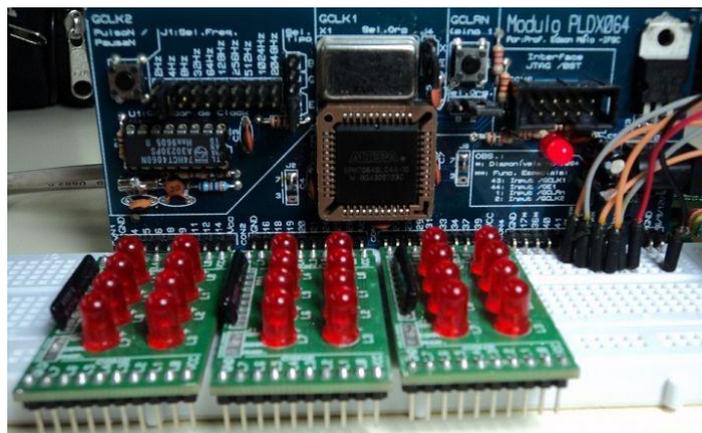
Figura 04 – Sistema implementado no CPLD



Fonte – Projeto FPGA para todos (fpgaparatodos.com.br)

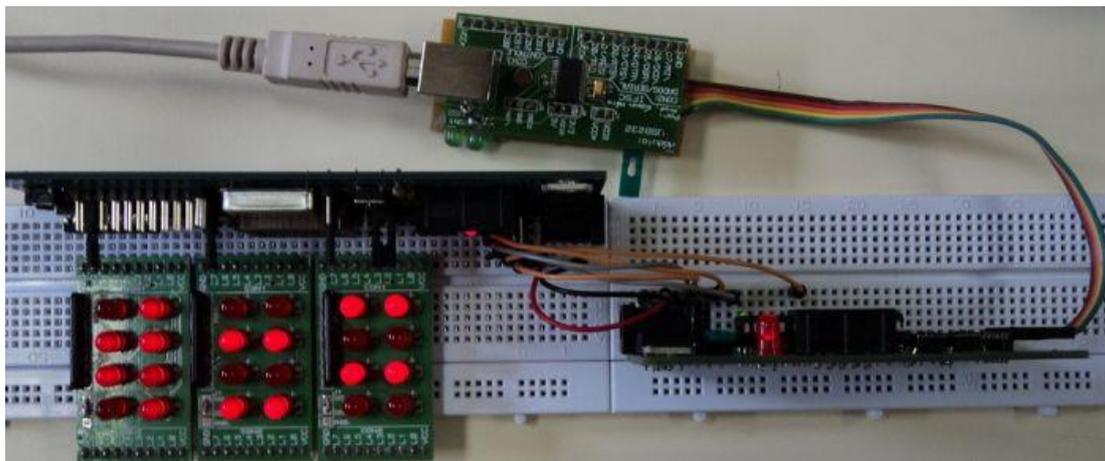
2.4 Integração e Teste do Sistema

Para exemplificar o uso do expansor, utilizaram-se três placas com oito LEDs cada como saídas do kit CPLD, representando uma expansão de seis interfaces (pinos da comunicação SPI e pinos de habilitação) para 24. O CPLD recebe dados do uCduíno e os apresenta nos LEDs. A Figura 05 mostra o kit CPLD montado com as placas de LEDs e os fios conectando o kit CPLD ao kit uCduíno.

Figura 05 – Kit CPLD com as placas de LEDs e uCduíno conectados.

Fonte – site FPGA para todos (fpgaparatodos.com.br)

A Figura 06 apresenta a montagem, com a interligação dos kits de CPLD e uCduíno. Na fotografia, o sistema é alimentado via cabo USB, a partir de um microcomputador PC.

Figura 06 - Sistema montado com o kit CPLD e o kit uCduíno

Fonte – site FPGA para todos (<http://fpgaparatodos.com.br>)

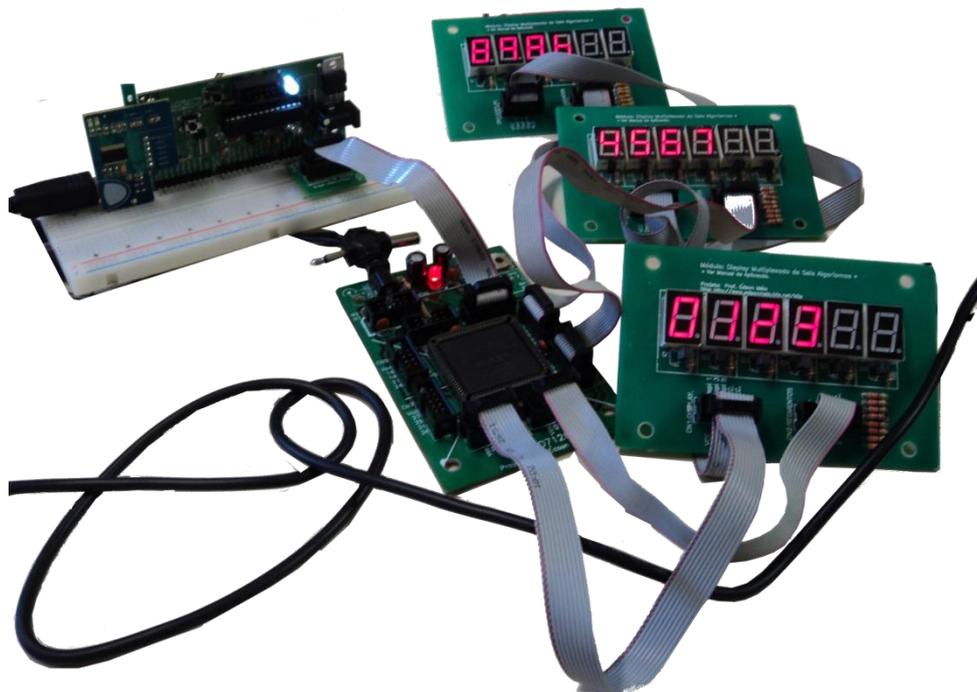
3 RESULTADOS E DISCUSSÃO

Em uma análise simples da temporização do sistema, verifica-se que a comunicação SPI no padrão Arduino, para o modo de operação programado e para o clock de 16 MHz do uCduíno, sendo a comunicação realizada a 4MHz, implica em um intervalo de aproximadamente 2 μ s para transmissão de um byte. Considerando o tempo de execução de instruções adicionais no programa, verificou-se um tempo médio de 8 μ s por byte, adequado para muitas aplicações práticas.

O desenvolvimento do sistema de expansão de E/S teve como um primeiro resultado a produção de uma página explicativa (SILVA, 2012), publicada no site fpgaparatodos.com.br, demonstrando passo a passo como implementar o sistema, além de elucidar, para os bolsistas envolvidos, conceitos de lógica programável, VHDL e aplicação de CPLDs.

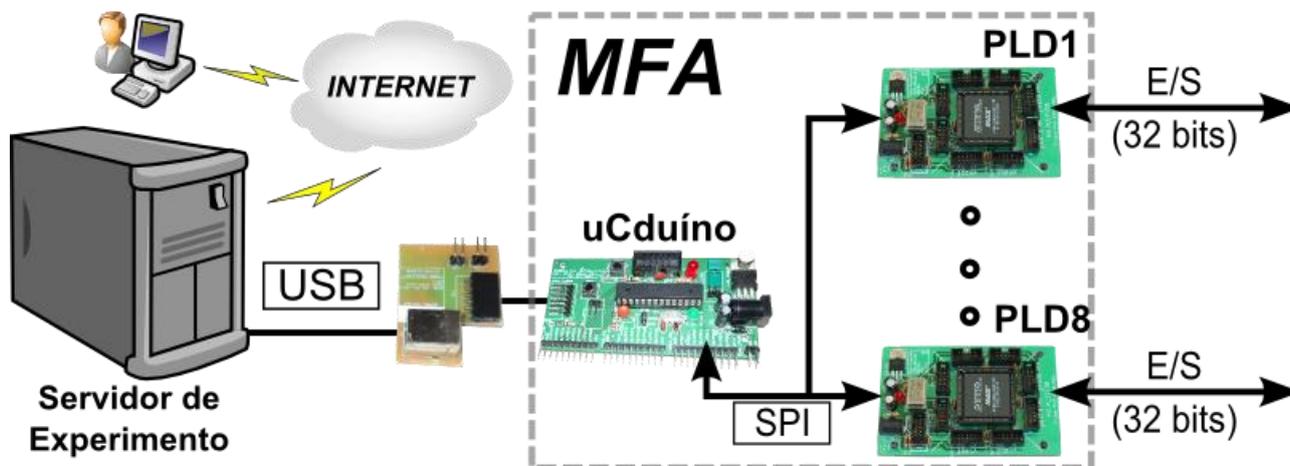
Além da página, um grupo de alunos e professores do curso de Automobilística do IF-SC (Campus Florianópolis) está utilizando o sistema em uma pesquisa na qual se exige o acionamento de diversos mostradores numéricos. A Figura 07 mostra um bloco do seu projeto, com os displays e o uCduíno. Nesta aplicação, o CPLD recebe do microcontrolador do uCduino, via SPI, a cada 2 ms, uma palavra de 32 bits que comanda os três mostradores. O tempo para a transmissão do dado é fator crítico aqui, pois o envio dos dados não deve comprometer os processos de aquisição de dados e cálculos matemáticos a serem executados no sistema. Aqui, o tempo verificado para o envio dos dados é de aproximadamente 32 μ s, compatível com as restrições do projeto.

Figura 07: Conjunto de uCduíno, Módulos de CPLD e mostradores.



Fonte: Projeto FPGA para Todos.

Mais uma aplicação da expansão de E/S do uCduino com CPLD está prevista no projeto de um Laboratório Remoto para experimentação com CPLD, atualmente em desenvolvimento no projeto FPGA para Todos. Neste caso, o microcontrolador se comunica com diversos CPLDs através da interface SPI, como mostrado no diagrama de blocos parcial seguinte.

Figura 08: Diagrama de Blocos do Laboratório Remoto.

Fonte: Projeto FPGA para Todos.

O sistema será utilizado como interface de comunicação entre o kit arduino ligado ao computador e os kits que estarão sendo utilizados. Diferentemente da aplicação anterior, o tempo despendido na comunicação entre o uCduíno e o CPLD não será crucial, pois o atraso relacionado a conexão de internet faz com que esse tempo seja irrelevante.

4 CONSIDERAÇÕES FINAIS

Através dos resultados, podemos observar que a aplicação desenvolvida é, de fato, muito útil, sendo utilizada em diversos outros projetos. Vale também destacar que este estudo visou a divulgar a utilização de lógica programável no âmbito institucional, sendo utilizado pelo curso de automobilística do próprio IFSC, bem como em âmbito geral, através da página explicativa divulgada na internet. Além disso, o sistema também se mostrou útil no desenvolvimento de outra atividade do Projeto FPGA para Todos, o de um laboratório remoto.

O conceito estudado aqui tratou apenas o CPLD fazendo a expansão das saídas. Um próximo projeto poderia ser o desenvolvimento de um sistema para a expansão da entrada e posteriormente a integração dos dois sistemas, de forma criar um componente completo.

AGRADECIMENTOS

Os autores agradecem ao CNPq, financiador do Projeto FPGA para Todos, e ao IFSC, instituição que abriga o projeto.

REFERÊNCIAS

ARDUINO. **SPI Library**. Disponível em: <<http://arduino.cc/en/Reference/SPI>>. Acesso em : 22 de Agosto de 2012.

ATMEL. **Atmel 8-bit Microcontroller with 4/8/16/32KBytes In-System Programmable Flash**. Disponível em: <<http://www.atmel.com/Images/doc8271.pdf> > Acesso em: 22 de Agosto de 2012.

MÉLO, Francisco Édson Nogueira, et al. **Do Scratch ao Arduíno: Uma proposta de ensino de programação para um primeiro semestre de um curso superior de tecnologia**. In: XXXIX Congresso Brasileiro de Educação em Engenharia, 2011, Blumenau, SC.

MÉLO, Francisco Édson Nogueira, et al. **FPGA para Todos: Um projeto para a disseminação da tecnologia de lógica programável**. In: XXXIX Congresso Brasileiro de Educação em Engenharia, 2011, Blumenau, SC.

SHULTE, Thomas. **Implementing the Top Five Control-Path Applications with Low-Cost, Low-Power CPLDs**. Altera Corporation. Disponível em: <http://www.altera.com/literature/wp/wp-01146-control-path-apps.pdf?GSA_pos=1&WT.oss_r=1&WT.oss=Implementing%20de%20Top%20Five > Acesso em: 10 de abril de 2012.

SILVA, Marcos Vinícius Leal. **Registrador Série-Paralelo (24 bits)**. Disponível em: <<http://fpgaparatodos.com.br/exemplos-de-aplicacao/exemplos-didaticos/22-em-construcao-multiplica-pinos-serie-paralelo-24bits.html>>. Acesso em: 29 de agosto de 2012.

WIKIPEDIA. **Serial Peripheral Interface Bus**. Disponível em: <http://en.wikipedia.org/wiki/Serial_Peripheral_Interface_Bus>. Acesso em: 10 de Abril de 2012.