

## **iPIC: UMA FERRAMENTA PARA AUXILIAR O APRENDIZADO DE DISPOSITIVOS PERIFÉRICOS EM MICROCONTROLADORES**

**René Pegoraro** – pegoraro@fc.unesp.br

**Humberto Ferasoli Filho** – ferasoli@fc.unesp.br

**Marco Corbucci Caldeira Caldeira** – caldeira@fc.unesp.br

Universidade Estadual Paulista, Faculdade de Ciências, Departamento de Computação

Avenida Luiz Edmundo Carrijo Coube 14-01

17033-360 – Bauru – São Paulo

**Silas F. R. Alves** – silas.alves@gmail.com

Universidade Estadual de Campinas, Faculdade de Engenharia Mecânica, Departamento de Projeto Mecânico, Laboratório de Automação Inteligente e Robótica

Rua Mendeleev, 200

13083-860 – Cidade Universitária "Zeferino Vaz" Barão Geraldo – Campinas – São Paulo

***Resumo:** Este trabalho apresenta detalhes do funcionamento e os benefícios da utilização da ferramenta iPIC como uma alternativa aos métodos de aprendizagem de microcontroladores e às ferramentas tradicionais utilizadas. O objetivo da ferramenta é facilitar o entendimento do funcionamento e da configuração dos periféricos em um microcontrolador qualquer sem depender da compreensão da linguagem de programação e do ambiente específicos usados na programação do microcontrolador.*

***Palavras-chave:** Microcontroladores, Periféricos, Interfaces*

### **1 INTRODUÇÃO**

Os microcontroladores são sistemas computacionais completos (SoC – *System on a Chip*), integrando em um único componente a unidade central de processamento (CPU), a memória e diversos periféricos programáveis que frequentemente apresentam interfaces conectáveis a dispositivos eletrônicos externos. Eles simplificam o desenvolvimento de equipamentos controlados eletronicamente com redução do número de componentes, aumento na versatilidade, barateamento dos projetos e economia de energia. Hoje, os microcontroladores equipam muitos equipamentos de uso cotidiano como TVs, telefones, geladeiras, cafeteiras, automóveis, etc.

O aumento da exigência dos consumidores por equipamentos mais versáteis e simples de serem utilizados, somada à redução dos preços dos microcontroladores, amplia a demanda por profissionais capacitados no desenvolvimento de equipamentos microcontrolados. Esta demanda se reflete na grade curricular de diversos cursos como Engenharias e Ciência da Computação, que precisam formar profissionais com conhecimentos na área de microcontroladores.

O aprendizado dos conceitos necessários à aplicação de microcontroladores ocorre através de ambientes integrados - editores, montadores, programadores, simuladores e emuladores - onde o estudante se debruça com o objetivo de desenvolver e testar seus programas. A utilização de um componente microcontrolador está sempre vinculada ao controle de dispositivos físicos com os quais ele deve interagir para realizar as tarefa

necessárias a um equipamento. Neste contexto, o desenvolvimento de um programa para um microcontrolador requer que o aluno escreva o programa em um ambiente específico fora do microcontrolador, grave este programa nele e em seguida insira o microcontrolador em um circuito com outros componentes eletrônicos para então realizar testes de funcionamento. Assim, o ensino dos componentes microcontroladores apresenta alguns desafios, pois o estudante necessita aprender um ambiente e uma linguagem de programação voltados às características específicas, para só então trabalhar com os periféricos disponíveis nestes componentes e conectá-los aos dispositivos físicos externos que serão controlados.

Uma abordagem que vem sendo frequentemente empregada para suplantar estes desafios é o oferecimento aos alunos de ambientes de desenvolvimento mais didáticos e simples de serem usados, onde os estudantes possam testar seus programas. O teste da programação de um microcontrolador pode usar diversas técnicas, podendo ser realizado em um simulador, emulador ou através da gravação no próprio chip. Um exemplo de ambiente que oferece estas técnicas é o MPLab da MICROCHIP (MICROCHIP, 2010). Em um simulador, o microcontrolador é criado virtualmente em software representando a execução do programa do usuário e o acionamento dos periféricos internos. Em poucos casos, alguns dispositivos externos são incluídos nestes simuladores (DEL RIO & ANDINA, 2002) (MARTIN-DEL-BRIO & BERNAL-RUIZ, 1999) (ROGER, 2009). Assim, usando-se um simulador, os dispositivos elétricos/eletrônicos externos serão limitados aos oferecidos pelo simulador, o que dificulta a compreensão dos aspectos práticos da utilização das interfaces e limita a criatividade do aluno. Além disso, nenhuma possibilidade de teste das conexões com os dispositivos externos reais é oferecida. Nos emuladores, um PC (Computador Pessoal) se conecta através de uma interface eletrônica e software específicos ao microcontrolador que está fisicamente instalado no circuito eletrônico do equipamento, permitindo que um programa possa ser depurado diretamente no microcontrolador. O inconveniente desta técnica é que a emulação exige que o microcontrolador tenha sido construído com circuitos específicos para esta finalidade. Uma alternativa, semelhante aos emuladores, utilizada por AL-DHAHER (AL-DHAHER, 2001), consiste em um programa monitor que é executado no microcontrolador e que recebe comandos predefinidos enviados por um PC, normalmente através da interface serial de um terminal de texto, e os executa no microcontrolador. Finalmente o estudante pode se submeter ao ciclo de programação, gravação do programa no microcontrolador e testes no circuito eletrônico real, mas, neste caso, a identificação de problemas na programação é dificultada pela falta de informações externadas pelo microcontrolador. Em qualquer uma destas técnicas o aluno precisará, antes da utilização do microcontrolador, enfrentar os desafios da compreensão do ambiente de programação, da linguagem específica utilizada e das características do microcontrolador.

Neste trabalho, é apresentada uma ferramenta que auxilia na aprendizagem e utilização dos dispositivos periféricos de um microcontrolador sem envolver a compreensão de um ambiente e linguagem específicos do microcontrolador. Nesta ferramenta, executada em um microcomputador usando uma biblioteca simples, os periféricos são programados em linguagem de alto nível padrão, permitindo que os periféricos e dispositivos reais conectados aos microcontroladores sejam manipulados diretamente pelo programa, sem a necessidade de reprogramações do microcontrolador.

## 1.1 O treinamento na programação de microcontroladores

O ensino-aprendizagem de microcontroladores no curso de Ciências da Computação da Universidade Estadual Paulista – UNESP, campus de Bauru, ocorre em uma disciplina alojada no quarto dos oito semestres do curso. Os alunos chegam à disciplina com bons conhecimentos de programação em linguagem de alto nível e de circuitos elétricos e eletrônicos, mas com pouca maturidade em linguagem de montagem. Neste contexto, os

alunos apresentam dificuldades para integrar os conceitos de linguagem de montagem e a programação dos periféricos do microcontrolador utilizando ambientes ainda desconhecidos. A proposta do iPIC traz a facilidade da utilização da linguagem de programação C padrão ANSI ou Pascal para configurar e utilizar os periféricos do microcontrolador de forma real, conectados externamente aos dispositivos, sem as limitações por um ambiente simulado. Assim, o aluno pode realizar programas em linguagem de alto nível em um PC que controlam os periféricos do microcontrolador, mesmo sem conhecer o ambiente ou a linguagem específica de programação deste.

Outro argumento importante é que grande parte do tempo de desenvolvimento da programação de um microcontrolador está na configuração dos seus periféricos, o que requer diversos ciclos de depuração e reprogramação do microcontrolador, até que todos os erros de configuração e de programação sejam resolvidos. Com o microcontrolador previamente programado com o código do iPIC, o aluno não precisa reprogramar o microcontrolador. Ele pode se dedicar a escrita do código de configuração dos periféricos na sua linguagem de alto nível e ambiente integrado preferidos até que a configuração esteja correta. Então, a partir deste estágio, apenas a programação na linguagem de programação específica do microcontrolador deve ser realizada, incorporando a configuração dos dispositivos periféricos do microcontrolador.

## 2 INTERFACE DE ACIONAMENTO

A ferramenta é composta por software e hardware. O software cria uma camada de abstração ao usuário, que utilizando linguagem de alto nível em um PC tem a impressão de programar diretamente os periféricos internos do microcontrolador. O hardware é implementado por um sistema de comunicação serial que conecta o PC do usuário ao microcontrolador objeto dos estudos. Nesta comunicação, os endereços a serem modificados, bem como os dados que serão escritos nestes endereços, são enviados ao microcontrolador, permitindo que seus periféricos sejam configurados pelo programa do usuário. Dessa forma, um programa executado no PC pode controlar os dispositivos eletrônicos conectados ao microcontrolador.

Esta ferramenta pode ser utilizada em qualquer Sistema Operacional (SO) e linguagem de programação, desde que estes apresentem suporte à comunicação serial.

### 2.1 Hardware

O hardware conecta o PC ao microcontrolador usando um cabo serial, um adaptador RS-232/TTL que compatibiliza os sinais da RS-232 aos níveis de tensão utilizados pelo microcontrolador e, na indisponibilidade da interface RS-232 no PC, um conversor USB/RS-232 pode ser utilizado.

A escolha da interface RS-232 (muitas vezes denominada por serial) traz características importantes à esta ferramenta. Do ponto de vista do microcontrolador, a interface serial pode ser implementada facilmente por software em qualquer microcontrolador. Ela necessita de apenas dois pinos de entrada e saída genéricos o que compromete pouco as interfaces e periféricos a serem programados. Além disso, devido a sua simplicidade, a serial não requer que o microcontrolador possua um periférico embarcado específico para a comunicação serial. Assim, até mesmo a uma interface serial do microcontrolador pode ter ser configurada através da ferramenta iPIC, pois esta não depende da serial que pode estar disponível no microcontrolador. No PC, muitos chipsets modernos de fabricantes como a Intel e da AMD continuam a incorporar a RS-232, mas seus conectores têm desaparecido das placas mães e consequentemente dos gabinetes dos PCs atuais. Entretanto, nos casos onde a interface serial

não pode ser usada diretamente, a ferramenta iPIC funciona normalmente através de um conversor USB/RS-232.

A partir da possibilidade da utilização de uma interface USB com um conversor USB/RS-232, surge sempre o questionamento: por que não usar simplesmente a USB que está disponível em todos os PC modernos? Quando comparamos as duas interfaces para esta aplicação a serial traz algumas vantagens. A serial usada para a transmissão de poucos bytes é mais rápida que a USB no modo HID (*Human Interface Device*) e CDC (*Communications Device Class*), que seriam os indicados para a implementação desta ferramenta. Além disso, o suporte em hardware para a USB não está disponível em todos os microcontroladores e sua implementação é inviável apenas por software.

Os dados transferidos da RS-232 do PC ao microcontrolador utilizam lógica negativa com sinais variando de -3 a -24V no nível 1 e de 3 a 24V no nível 0, enquanto os portos em modo de entrada do microcontrolador (PIC16F873) utilizado na construção do iPIC, entende como nível 0 valores entre  $V_{SS}$  e  $0,15 \cdot V_{DD}$ , e como nível 1 valores de 2V a  $V_{DD}$ , sendo  $V_{SS} = 0$  e  $V_{DD} = 5V$ . Assim, para que esses sinais possam ser compatibilizados, alguns componentes precisam ser utilizados. No sentido contrário, do microcontrolador para a serial, felizmente, apesar de fora do padrão da RS-232, as RS-232 que equipam os PCs e os adaptadores USB/RS-232 usados onde a ferramenta foi testada aceitam como entrada níveis de tensão TTL, que são gerados pelo microcontrolador usado, sem a necessidade de componentes para compatibilização de tensão. A Figura 1 apresenta o circuito utilizado.

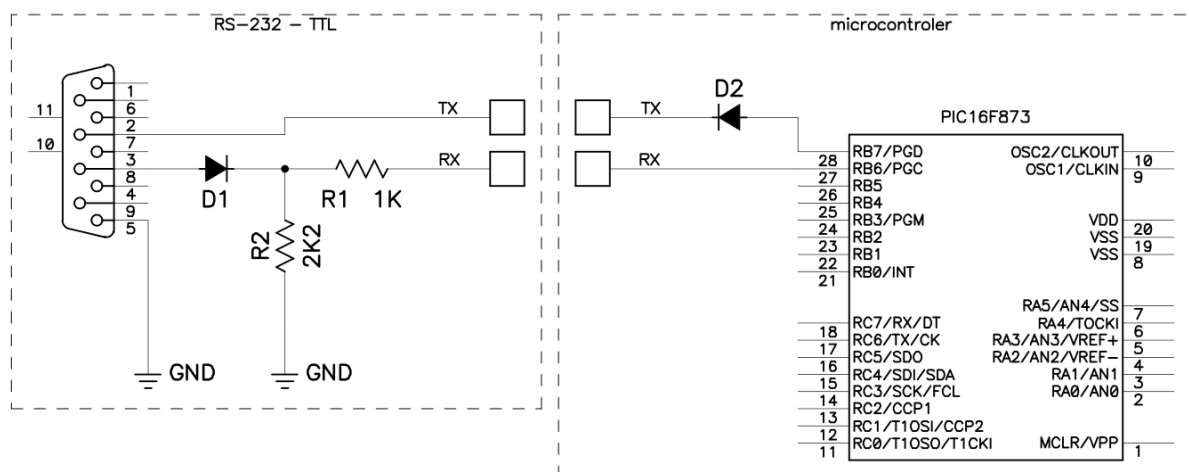


Figura 1 – Circuito de conversão RS-232 - PIC

Observe que no circuito da Figura 1 não existe a inversão da lógica que permanece negativa. A inversão é realizada pelo software que implementa a transferência serial dos dados no microcontrolador.

Para que vários microcontroladores possam ser ensaiados em sistemas com interfaces SPI, I2C, ou outra qualquer que permita a conexão de dois ou mais microcontroladores, o hardware descrito permite a conexão em uma única serial de até 16 microcontroladores. Portanto, vários módulos de microcontroladores, como os mostrados à direita da Figura 1, podem ser conectados em paralelo a um único módulo RS-232-TTL, também apresentado da Figura 1 ao lado esquerdo.

## 2.2 Software

O software é dividido em duas partes: software embarcado e API (*Application Programming Interface*). O software embarcado é embutido no microcontrolador e é responsável pelo recebimento dos comandos via serial e pelo o acionamento dos periféricos

internos do microcontrolador, com um retorno opcional de dados pela serial. Este software pode ser entendido de forma análoga a um serviço de uma SOA (*Service Oriented Architecture*), sendo este responsável por esperar requisições de leitura ou escrita provenientes do PC. Por outro lado, a API é executada no PC e se apresenta na forma de uma biblioteca de funções que enviam requisições pela serial ao microcontrolador, abstraindo o processo de comunicação. A API pode ser vista como o cliente do serviço prestado pelo microcontrolador. O objetivo desta biblioteca é simplificar a utilização da ferramenta em linguagem de alto nível.

Dois tipos de operações são servidas pelo microcontrolador, uma operação de leitura e uma de escrita, que são aplicadas aos registradores do microcontrolador. A leitura consulta o valor armazenado em um registrador ou posição de memória e o devolve para o PC. A escrita atribui um valor em um registrador ou posição de memória. Estas operações de leitura e escrita, representadas na Figura 2, são utilizadas em linguagem de programação de alto nível através das funções *consulta* e *atribui*, respectivamente. Quando uma destas duas funções é chamada no PC, um pacote de dados é enviado para ao microcontrolador, que o processa e devolve um pacote de resposta contendo uma confirmação ou um dado. O processamento no microcontrolador faz acesso através de endereçamento indireto ao registrador/memória, o que pode limitar a utilização desta ferramenta em microcontroladores muito simples sem este modo de endereçamento.

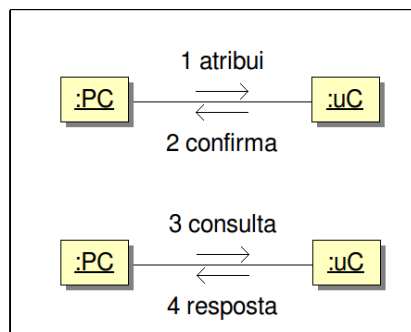


Figura 2 – Operações realizadas entre o PC e o microcontrolador (uC)

A função *atribui* tem dois parâmetros de entrada: *numReg* e *dado*. O parâmetro *numReg* contém o endereço do registrador que será alterado, enquanto o parâmetro *dado* carrega o valor que será atribuído para tal endereço. Quando esta função é chamada, um pacote de atribuição (Figura 3) é enviado pela serial, o microcontrolador recebe o pacote, verifica se ele está correto e realiza sua função, atribuindo o dado ao registrador indicado por *numReg*. Em seguida o microcontrolador envia um pacote (Figura 4) de resposta confirmando o recebimento da atribuição e sua realização. O byte 2 de todos os pacotes indica o tipo do pacote (4 bits menos significativos) e o processador recebendo acesso (4 bits mais significativos).

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
F0	Ender. do Proc.	0	Endereço do Registrador	Dado
				Byte Verificação

Figura 3-Pacote de atribuição enviado do PC ao microcontrolador.

Byte 1	Byte 2	Byte 3	Byte 4
F0	Ender. do Proc.	3	Código do Erro
			Byte Verificação

Figura 4-Pacote de confirmação enviado do microcontrolador ao PC.

A função consulta tem apenas o parâmetro *numReg*, que indica qual o registrador será lido no microcontrolador. Quando esta função envia um pacote de consulta (Figura 5) pela serial, o microcontrolador recebe este pacote, verifica se ele está correto e consulta o registrador endereçado por *numReg*. Se a leitura ocorrer com sucesso um pacote com o dado lido (Figura 6) é enviado do microcontrolador ao PC, por outro lado, caso ocorra algum erro, um pacote de confirmação (Figura 4) com o código do erro é enviado.

Byte 1	Byte 2	Byte 3	Byte 4
F0	Ender. do Proc.	1	Endereço do Registrador
			Byte Verificação

Figura 5-Pacote de consulta enviado do PC ao microcontrolador.

Byte 1	Byte 2	Byte 3	Byte 4
F0	Ender. do Proc.	2	Dado
			Byte Verificação

Figura 6-Pacote de resposta enviado do microcontrolador ao PC.

As funções responsáveis pela leitura e escrita, bem como as mensagens resultantes delas trocadas entre os PC e o microcontrolador, podem ser observadas na Figura 7.

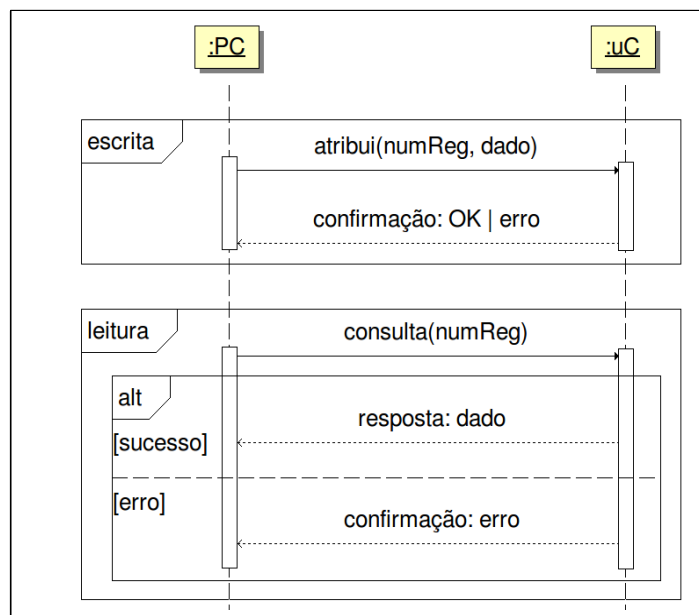


Figura 7-Mensagens trocadas entre o PC e o microcontrolador (uC) nas operações de leitura/escrita.

O tempo calculado para a troca de mensagens de leitura ou escrita, ignorando a latência do SO usado, é menor que  $800\mu\text{s}$  para transferir 9 bytes com a serial configurada a 115200bps utilizando a interface serial nativa do PC.

Para que os alunos possam conhecer os periféricos uma série de experimentos é realizada através da ferramenta iPIC.

### 3 EXPERIMENTOS

Os experimentos propostos aos alunos do conjunto de disciplinas de hardware e de introdução à programação vão desde experimentos simples como o acionamento de LEDs e a leitura de teclas até o controle de dispositivos mais complexos. Para esta finalidade, são utilizadas algumas plataformas didáticas multidisciplinares criadas para ensinar aos alunos conceitos de Algoritmos, Controle, Software Básico, Eletrônica e Eletricidade. Anteriormente ao desenvolvimento da iPIC, estas plataformas eram controladas pela interface paralela dos PCs (ALVES et al., 2011) (ALVES & FERASOLI, 2008). Agora, estas plataformas também integram o treinamento da programação dos periféricos do microcontrolador. Apresentamos a seguir três experimentos que exigem dos alunos o aprendizado de periféricos do microcontrolador: um elevador, um pequeno robô e um gerenciador de sistema de reservação de água.

#### 3.1 Elevador

O Elevador simula um elevador real, com seis andares. Este elevador possui: um botão de chamada para o elevador e um LED indicando o atendimento da chamada em cada andar; um display de 7 segmentos para indicar o andar atual da cabine; um sensor de presença da cabine em cada andar; um painel para a escolha do andar desejado dentro da cabine, contando com um botão de chamada e um LED de indicação para cada andar.

Sua estrutura foi construída a partir do conjunto mecânico de uma impressora a jato de tinta. Seu circuito eletrônico foi elaborado com componentes discretos e Circuitos Integrados TTL – estudados nas disciplinas de Circuitos Digitais e Laboratório de Circuitos Digitais.

A implementação do Software de controle é a proposta do Elevador. Os alunos devem programar o Elevador para que ele funcione como um elevador real, atendendo as chamadas dos passageiros e tratando os problemas de lógica que podem ocorrer.

O iPIC é utilizado aqui para que os alunos leiam os estados dos botões e façam o acionamento do motor de passo, dos LEDs e do indicador de 7 segmentos. Para esta finalidade, o PIC16F873 deve ter os pinos de entrada/saída programados, fazer as leituras dos estados e acionar os pinos correspondentes às ações desejadas..

#### 3.2 Robô Michelangelo

O Michelangelo é um robô móvel desenvolvido para fins didáticos capaz de identificar marcações na superfície por onde ele trafega e se deslocar sobre uma superfície plana. Dois motores movimentam o robô em uma configuração de acionamento diferencial (*differential-drive*). Diversas atividades são propostas para o aluno, normalmente relacionadas a navegação.

A estrutura mecânica é composta por dois servo motores de miniaturas radio controladas modificados para rodarem livremente. O acionamento de cada motor ocorre através de uma ponte-H.

O robô é dotado de quatro sensores ópticos, em linha na parte frontal, que permitem a detecção de faixas pretas no solo. Com estes sensores, o Michelangelo pode ser usado para seguir uma linha preta desenhada no solo, procurar saídas de uma região delimitada por uma cor preta, explorar labirintos, etc.

Aqui as velocidades dos motores devem ser programadas através do periférico PWM do microcontrolador para que o robô faça acelerações e desacelerações suaves na sua velocidade de deslocamento. Os alunos devem programar os periféricos de PWM e os pinos de entrada do microcontrolador que lerão os sensores ópticos, para depois fazerem os acionamentos necessários para realizar uma tarefa específica com o robô.

### 3.3 Sistema de Reservação de Água

Este experimento é composto por duas caixas de acrílico em níveis diferentes, de forma que a caixa mais alta fornece água deslocada pela gravidade para a caixa mais baixa. A caixa mais alta recebe água da caixa mais baixa através de uma bomba. Para que o experimento fique mais interessante, existe um percurso de 5 metros de dutos que a água precisa percorrer para chegar à caixa superior. Uma boia conectada a um potenciômetro mede o nível da caixa mais alta.

O objetivo deste experimento é manter a caixa superior sempre com água, ligando a bomba o menor número de vezes possível.

Para que o nível de água possa ser aferido pelo programa do aluno, este deve configurar um conversor AD (Analogico para Digital) do microcontrolador. Durante o processo de controle, o valor analógico correspondente ao nível da bomba deve ser monitorado. Quando este nível baixar de um limite mínimo, a bomba deve ser acionada e permanecer ligada até que um nível máximo de água na caixa superior seja alcançado.

## 4 RESULTADOS OBTIDOS

Esta ferramenta vem sendo utilizada nos últimos dois anos (duas turmas) nas disciplinas Laboratório de Circuitos Digitais e Algoritmos I e II que antecedem o ensino-aprendizagem do microcontrolador. Não há, portanto, resultados quantitativos estatisticamente confiáveis do sucesso desta ferramenta, uma vez que as duas turmas somam aproximadamente 60 alunos. Notamos, porém, que os alunos estão mais confiantes sobre o funcionamento e a programação dos periféricos e que grande parte da preocupação dos alunos com os periféricos não aparece durante o aprendizado da programação do microcontrolador em linguagem de montagem. Mesmo a confusão apresentada por alguns alunos que consideravam que a compreensão dos periféricos era parte da linguagem de montagem foi menos percebida pelo professor.

## 5 CONSIDERAÇÕES FINAIS

Este trabalho apresentou a ferramenta iPIC de apoio ao aprendizado da configuração de periféricos de microcontroladores. Apesar de não estar limitada a um microcontrolador específico, utilizou-se um microcontrolador PIC16F873 conectado a um PC através de uma interface serial e uma biblioteca de rotinas que possibilita aos alunos a programação direta dos periféricos do microcontrolador em alto nível no seu ambiente e SO preferidos, sem regravações do programa na flash.

Algumas das vantagens oferecidas pela ferramenta iPIC são:

- Programação dos periféricos em linguagem de alto nível padrão ANSI;
- Testes de ambientes reais;
- Permite que alguns periféricos sejam entendidos e utilizados sem conhecer as linguagens de programação específicas para o microcontrolador.

A ferramenta iPIC contrapõe a simplicidade da ideia com vasta gama de aplicações. Além dos experimentos apresentados, que são base para a compreensão de alguns periféricos e sua futura programação em linguagem específica no microcontrolador, fica evidente que esta ferramenta tem potencial para ser usada mesmo em aplicações onde o PC permanece como sendo o local da execução da lógica do controle e o acionamento através dos periféricos do microcontrolador.



**REFERÊNCIAS BIBLIOGRÁFICAS**

ALVES, S. F. R. ; FERASOLI FILHO, H. . Bancada Experimental Robótica para o Ensino de Computação. **Anais**: XX Congresso de Iniciação Científica, São José dos Campos – SP, 2008.

ALVES, S. F. R., FERASOLI FILHO, H., PEGORARO, R., CALDEIRA, M. A. C., ROSÁRIO, J. M., YONEZAWA, W. M. Educational Environment for Robotic Applications in Engineering. In: Eurobot Conference 2011 - 4th International Conference on Research and Education in Robotics, 2011, Praga - República Tcheca. **Anais**: Research and Education in Robotics - EUROBOT 2011. Berlin - Alemanha: Springer, 2011. v. 161. p.17–28.

AL-DHAHER, A. H. G. Integrating hardware and software for the development of microcontroller-based systems. **Microprocessors and Microsystems**, v. 25, n. 7, p. 317–328, 2001.

DEL RIO, A.; ANDINA, J. J. R. UVI51: A Simulation Tool for Teaching/Learning the 8051 Microcontroller. **Anais**: Frontiers in Education Conference, 2000. FIE 2000. 30th Annual. v. 2, p.F4E, 2002.

MARTIN-DEL-BRIO, B.; BERNAL-RUIZ, C. A Software Tool for Teaching Microcontroller System Principles. *International journal of electrical engineering education*, v. 36, p. 279–286, 1999.

MICROCHIP. **MPLAB Integrated Development Environment**. Microchip, 2010.

ROGER, J. **EdSim51's guide to the 8051 : core of the popular 51 series of 8-bit microcontrollers**. CreateSpace, 2009.

**iPIC: A TOOL TO ASSIST THE LEARNING OF MICROCONTROLLER PERIPHERAL DEVICES**

***Abstract:** This paper shows the operational details and benefits of using the iPIC tool as an alternative to traditional tools and learning methods for teaching the configuration and usage of the PIC microcontroller embedded peripheral devices.*

*The purpose of this tool is to ease the learning of the microcontroller peripherals' functionalities and configuration without specific knowledge about the programming language and development environment.*

**Key-words:** *Microcontrollers, Peripherals, Interfaces.*