

# DESENVOLVIMENTO DE UMA FERRAMENTA PARA AUXÍLIO AO APRENDIZADO DE ALGORITMOS E PROGRAMAÇÃO EM CURSOS DE ENGENHARIA

**Armando L. Keller** – contato@armandokeller.com

**Farlei Heinen** – farlei@unisinis.br

**Lucio R. Prade** – luciorp@unisinis.br

**Marcio R. Silva** – marcior@unisinis.br

**Rodrigo M. Figueiredo** – marquesf@unisinis.br

Universidade do Vale do Rio dos Sinos (UNISINOS)

Avenida Unisinis, 950, Cristo Rei

93022-000 – São Leopoldo - RS

***Resumo:** Este trabalho apresenta uma ferramenta desenvolvida dentro da Universidade do Vale do Rio dos Sinos (UNISINOS), que tem como objetivo tornar mais amigável o contato inicial de estudantes dos cursos de engenharia da área eletro-eletrônica, com as disciplinas de algoritmos e programação. Esta ferramenta proporciona a antecipação do contato do aluno com um dispositivo microcontrolado, que permite a manipulação de entradas e saídas de forma simples, sem a necessidade de conhecimentos específicos sobre microcontroladores. Com este contato antecipado a estas tecnologias que serão abordadas em profundidade no decorrer do seu curso de graduação, busca-se tornar o estudo de algoritmo um pouco menos abstrato e mais paupável. Ações físicas como, o acionamento de algum dispositivo, ou ações programadas para o acionamento de uma chave, por exemplo, ajudam a entender a lógica e o fluxo de execução de um programa. Além disso, também busca-se aumentar a curiosidade do aluno sobre conteúdos que serão aprendidos ao longo do curso aumentando assim o caráter multidisciplinar.*

***Palavras-chave:** Algoritmo, Microcontrolador, Programação, Python, ANSI C.*

## 1. INTRODUÇÃO

Já a alguns anos, observa-se um aumento na procura por formação na área de engenharia, um mercado ainda carente de profissionais no Brasil e que muitas vezes proporciona bons salários. Outro fenômeno que vem aumentando com esse movimento o índice de reprovação e evasão dos alunos ingressantes em cursos de engenharia. Disciplinas como cálculo, lógica de programação e algoritmos que estão alocadas no início da formação possuem índices médios em torno de 50% de reprovação [7].

Na Universidade do Vale do Rio dos Sinos, várias ações estão sendo feitas com o intuito de melhorar esse índice, uma delas, desenvolvida nos cursos de Engenharia Elétrica, Engenharia Eletrônica e Engenharia da Computação, visa diminuir a reprovação e a evasão especificamente na disciplina de Algoritmos e Linguagem de Programação C.

Por experiências prévias em sala de aula, observa-se que alunos que tiveram alguma falha em sua formação matemática no ensino médio, tendem a apresentar essa dificuldade em abstrair situações e informações, práticas estas essenciais para o entendimento de lógica de programação e o desenvolvimento de algoritmos para programação.

Conforme Piaget [8] o aprendizado por meios concretos e palpáveis se dá mais rapidamente do que aquele que usa apenas a abstração, sendo assim buscou-se introduzir ferramentas que gerassem interações físicas junto com o desenvolvimento da lógica e do algoritmo para tornar o aprendizado da disciplina mais fácil, com isso diminuído a reprovação e a evasão.

## **2. DESENVOLVIMENTO DA FERRAMENTA**

Com a atualização da disciplina de circuitos microcontrolados, novos *kits* didáticos que utilizam microcontroladores ARM foram adquiridos e os *kits* antigos que utilizavam o microcontrolador PIC ficaram em desuso.

Para dar uma continuidade de uso a estes *kits*, diminuindo assim também o descarte de lixo eletrônico, pensou-se em reconfigurar seu *hardware*, desenvolvendo uma interface amigável para ativar e desativar saídas e ler entradas, que fossem gerenciadas por software via interface USB de um microcomputador PC. Surgiu assim, um *kit* que foi rebatizado de USB/IO, que é utilizado nas disciplinas introdutórias de programação.

### **2.1. Hardware**

O *kit* que foi utilizado no desenvolvimento (Figura 1), possui um microcontrolador Microchip PIC18F4550, um microcontrolador de baixo custo com suporte a entradas analógicas, entradas e saídas digitais, comunicação serial, I2C, CAN, USB entre outros recursos.

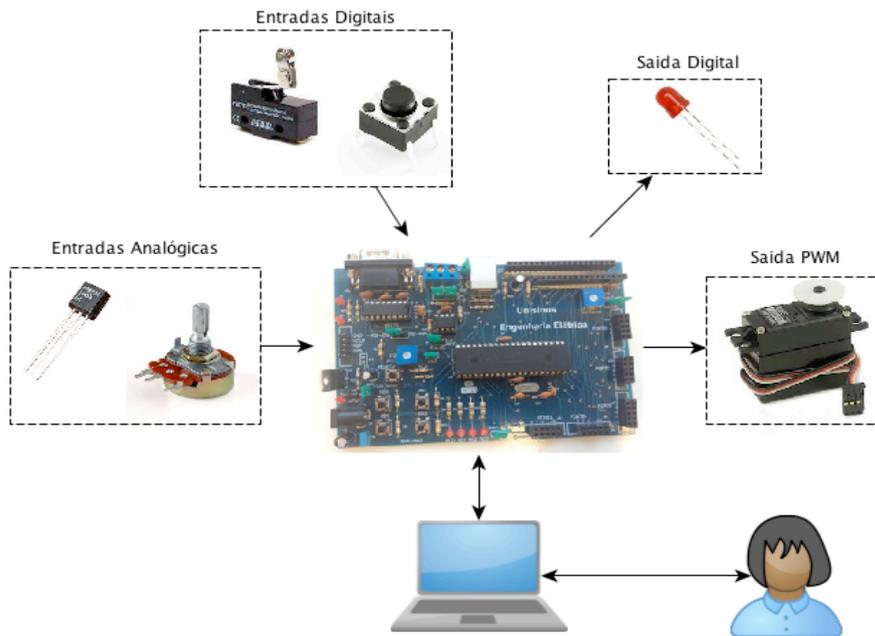


Figura 1 – Kit microcontrolado usado como base para a ferramenta.

Todos os pinos utilizáveis do microcontrolador são acessíveis por barras de pinos separados pelo *PORT* aos quais pertencem, além de alguns pinos já estarem ligados aos elementos de uso mais comum como 4 botões, 4 *LEDs*, e 1 potenciômetro, que podem ser desativados por *jumpers* disponíveis na placa. A alimentação do *kit* pode ser por uma fonte de 5V ou pela própria porta USB, isto sendo configurado por *jumpers* disponíveis na placa.

## 2.2. O *Firmware* do *Kit*

O *firmware* foi desenvolvido na linguagem de programação C baseado no exemplo de utilização da biblioteca *libusb* disponibilizado pela Microchip “*USB Device - Libusb - Generic Driver*”[3], compilado pelo compilador Microchip C18[2].

Utilizou-se uma conexão USB tipo *interrupt* para nos comunicar com o computador, estabelecendo uma comunicação Mestre-Escravo, onde o *firmware* tem o papel de escravo, ou seja, ele só opera quando solicitado pelo mestre que neste caso é um computador, como demonstrado na Figura 2.



Figura 2 – Fluxo de dados entre o PC e o Kit

Optou-se por este modelo mestre-escravo pois como os usuários da API (*Application Programming Interface*) serão na sua grande maioria iniciantes na área da programação e utilizarão uma programação sequencial, ou seja, que executará somente uma instrução por vez, ao programa mestre precisa saber qual dados receberá para tratá-lo. Ao enviar algum dado para o escravo espera-se por uma resposta, que após ser tratada é devolvida para o usuário.

Utilizou-se estruturas de dados para organizar as informações transferidas por vetores de *bytes*, sendo estas separadas por informações e *flags* de comando, descartando assim necessidade da utilização de um protocolo mais complexo.

### 2.3. A API de Desenvolvimento

Foi desenvolvida uma API de uso simples para que as entradas e saídas do *kit* possam ser manipuladas pela porta USB de qualquer computador com sistema operacional *Linux*. A API do Kit está disponível em *Python* e ANSI C, linguagens de programação adotadas pela facilidade de aprendizado e utilização futura das mesmas, sendo os comandos da API nas duas linguagens exatamente iguais, para diminuir a curva de aprendizagem. Estas funções que serão inseridas no fluxo do algoritmo desenvolvido podem ser vistas na Tabela 1.

Função	Utilização
iniciar()	Realiza as verificações iniciais, como se o <i>kit</i> esta conectado e disponível para uso e estabelece o inicio da comunicação.
abrir_pwm()	Habilita o periférico e a saída de PWM ( <i>Pulse Width Modulation</i> ) na frequência desejada.
setar_pwm_dc()	Ajusta o <i>Duty Cycle</i> do PWM.
fechar_pwm()	Desliga o periférico de PWM.
abrir_adc()	Habilita o periférico conversor ADC ( <i>Analog Digital Converter</i> ) com a taxa de aquisição desejada.
ler_adc()	Retorna o valor da entrada analógica desejada.
fechar_adc()	Desliga o periférico ADC.
setar_saida()	Seta a saída digital desejada para o nível escolhido, sendo estes ligado ou desligado.
ler_entrada()	Retorna se existe um na entrada digital escolhida.
finalizar()	Desaloca a memória utilizada pela API, e encerra a conexão.

Tabela 1 - Funções disponíveis na API desenvolvida, que manipulam entradas e saídas do *kit* de maneira transparente para o usuário.

Estas 10 funções são responsáveis pelo interfaceamento entre *hardware* e *software* desenvolvido pelo do aluno, sem que este precise ter conhecimento de como funciona uma

comunicação USB ou serial, reduzindo a complexidade do sistema e tornando assim a atividade mais agradável para o aluno, e instigando a curiosidade de como este interfaceamento funciona para as disciplinas futuras.

A API foi inicialmente desenvolvida utilizando a linguagem de programação ANSI C por esta linguagem ser utilizada na disciplina de algoritmos. Em seguida a biblioteca foi portada para a linguagem *Python*[6], para que fosse mais natural ainda esta comunicação entre o *software* e o *hardware*, por esta última ser uma linguagem interpretada e também por possuir um console que permite manipular o sistema interativamente. O aluno pode controlar o *hardware* diretamente do terminal, enquanto está desenvolvendo o seu programa. Além disso, *Python* é uma linguagem de fácil aprendizagem e com uma gama imensa de bibliotecas, as quais por exemplo podem ser utilizadas para facilitar a plotagem de gráficos referentes a dados adquiridos pelo *hardware*.

A portabilidade de C para *Python* foi simples, uma vez que a API havia sido compilada como biblioteca compartilhada (*Shared Library*), assim utilizando a biblioteca *ctypes*, nativa do *Python*, podemos importar uma biblioteca compartilhada diretamente para o *Python*, sem precisar realizar alterações no código.

### 3. ESTUDO DE CASO

Para a avaliação dos efeitos do ensino utilizando o *kit*, foram escolhidas algumas turmas da disciplina de Algoritmos e Programação em linguagem C, dos cursos de Engenharia Elétrica, e Engenharia Eletrônica da Unisinos.

No curso de Engenharia Elétrica, as aulas foram ministradas sem utilizar o *kit*, utilizando exemplos clássicos que normalmente são referentes a tratamento de texto, ou resolução de cálculos matemáticos utilizando o console. Já no curso de Engenharia Eletrônica, as aulas foram ministradas com o *kit*, com exemplos que continham elementos externos ao console, tendo um grande apelo visual.

Levando em conta que existem pessoas auditivas, visuais e cinestésicas, ao ministrar uma disciplina que aborda algo completamente novo é importante que todos estes tipos de pessoas sejam atendidas durante uma aula, espera-se que com este *kit*, o processo de construção do conhecimento seja mais fácil e rápido.

Tendo acesso as entradas e saídas do *hardware* por uma API simples, os alunos podem desenvolver projetos mais interessantes onde já serão apresentados a elementos como chaves, sensores, LEDs, *buzzers* e motores que são vistos em disciplinas futuras do curso.

Durante quatro semestres consecutivos o curso de Engenharia Eletrônica utilizou o *kit* integrado com *hardwares* especialmente desenvolvidos pelos professores para o processo de aprendizagem. O objetivo dos alunos foi criar um *software* de controle simples para esse *hardware*. Para manter o foco dos alunos no aprendizado de Algoritmos e Linguagem C, o controle do *hardware* foi simplificado, utilizando somente algumas entradas e saídas digitais.

O primeiro *hardware* foi uma plataforma de transporte automatizada (figura 4a), com a capacidade de selecionar o destino dos materiais de acordo com o tamanho de cada um deles. Os alunos deveriam desenvolver um programa em linguagem C, que se comunicava com o *kit* e controlava a plataforma. Era possível acionar a plataforma nas duas direções, acionar cada

motor que selecionava os materiais, obter leituras de fim de curso e de qual material estava sendo transportado (tudo através de sinais digitais simples).

O segundo *hardware* desenvolvido foi um sistema para estande de tiro (figura 4b) utilizando sensores piezo e um rifle de brinquedo (Nerf). O objetivo dos alunos era desenvolver um jogo em linguagem C que controlava o estande usando o *kit* e informava o jogador sobre os pontos e os tempos obtidos. O *hardware* fornecia a informação de qual alvo foi atingido e era possível acionar leds em cada alvo e emitir um aviso sonoro. Novamente o *hardware* foi desenvolvido com uma interface simples, puramente digital, para que os alunos se preocupassem mais com o desenvolvimento dos algoritmos em linguagem C, e menos com o processo de interfaceamento.

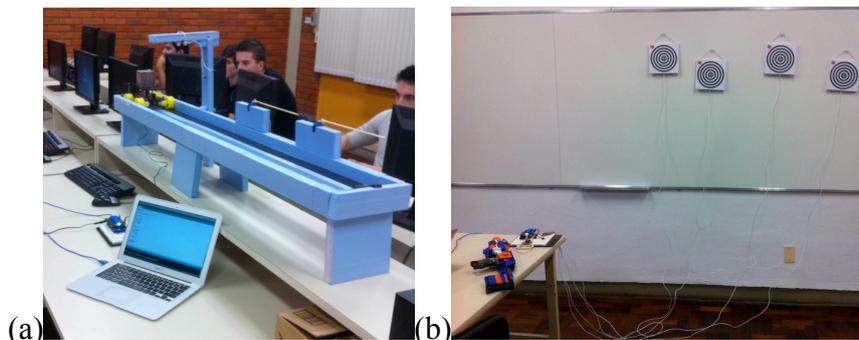


Figura 3 - (a) plataforma de transporte; (b) estande de tiro

#### 4. RESULTADOS OBTIDOS

Após um semestre de acompanhamento das turmas que utilizaram a ferramenta em sala de aula, observou-se um desempenho geral da turma acima dos resultados de semestres anteriores, 90% dos alunos tiveram aproveitamento superior 8,0. Ainda em relatos espontâneos durante as aulas, um grande grupo afirmou estar empolgado com o transcorrer de seu curso e muito ansiosos com as próximas disciplinas específicas, onde poderão conseguir mais subsídios de eletrônica, microprocessadores, acionamentos, sistemas embarcados e desenvolver projetos por conta própria.

Após quatro semestres de acompanhamento, foi possível observar uma melhora qualitativa no desempenho dos alunos e também pode-se comprovar através de dados obtidos no sistema de gerenciamento de alunos da universidade [7] a diminuição de reprovações e evasão se comparado com a turma que não utiliza a ferramenta em sala de aula. Esse comparativo pode ser visto na Tabela 2.

	Turmas sem uso da ferramenta	Turmas com o uso da ferramenta
<b>Reprovação Média</b>	23,8%	11,3%
<b>Evasão Média</b>	12,7%	4,8%

Tabela 2 – Comparativo de evasão e reprovação nos dois cenários.

## 5. CONCLUSÃO

Este estudo ainda está em fase inicial, apesar dos resultados positivos o acompanhamento desta mesma turma deverá prosseguir nos próximos semestres onde disciplinas que utilizam algoritmos e programação como base serão cursadas. Nestas disciplinas sequenciais como Estrutura de Dados e Circuitos Microprocessados, atualmente observa-se uma grande deficiência dos alunos nos conhecimentos básicos de lógica de programação e algoritmos. Assim, espera-se monitorar os impactos que a introdução deste dispositivo para o auxílio ao ensino de algoritmos e programação possa causar nestas disciplinas posteriores.

## 6. TRABALHOS FUTUROS

A partir da observação dos primeiros resultados favoráveis já demonstrados na seção de conclusão do trabalho, pretende-se realizar evoluções no *hardware* utilizado, com a criação de placas intercambiáveis para expansão das funcionalidades, tais como matriz de LEDs para desenvolvimento de trabalhos com interfaces informativas, circuitos de potência para o acionamento e controle de pequenos motores que poderão ser utilizados em automações simples, circuitos com sensores de corrente e tensão para a medição destas grandezas em experimentos realizados em disciplinas do curso, entre outros. Pretendendo assim aumentar o caráter multidisciplinar durante a formação do aluno no curso de engenharia.

### *Referencia bibliográfica*

- [1] Datasheet do kit disponível em [moodle.unisinos.br](http://moodle.unisinos.br)
- [2] [http://ww1.microchip.com/downloads/en/devicedoc/mplab\\_c18\\_libraries\\_51297f.pdf](http://ww1.microchip.com/downloads/en/devicedoc/mplab_c18_libraries_51297f.pdf)
- [3] <http://libusb.sourceforge.net/api-1.0/index.html>
- [4] <http://www.beyondlogic.org/usbnutshell/usb1.shtml>
- [5] <http://docs.python.org/library/ctypes.html>
- [6] Página Oficial da linguagem Python <http://www.python.org/> acessada em junho de 2014.
- [7] Sistema RM da Unisinos, dados acadêmicos
- [8] PIAGET, Jean. *Biologia e Conhecimento*. 2<sup>a</sup> Ed. Vozes : Petrópolis, 1996

## **DEVELOPMENT OF A TOOL TO AID THE LEARNING OF ALGORITHMS AND PROGRAMMING IN ENGINEERING COURSES**

**Abstract:** *This work describes an educational tool developed at Universidade do Vale do Rio dos Sinos (UNISINOS). The aim of this tool is to make the first programming experience*

*easier for freshman students. With this tool, the student can interact with electronic devices with no extra knowledge required, making it possible for the student to focus on the programming task. The use of electronic devices to interact makes the study of programming less tedious for the student and should make him/her more interested in the subject. Physical interaction with devices other than the computer like changing the state of a LED or a motor may help the understanding of programming logic and program flow by increasing the student's interest on the subject. As the student gets more familiarized with these electronic devices, it will be easier for him/her to get more interested on future subjects to be studied in his/her undergrad course.*

**Key-words:** *Microcontroller, Programming, Python, ANSI C, USB*