



CONSTRUINDO VEÍCULO TELEOPERADO COM ARDUINO PARA AUXÍLIO NO ENSINO DE SISTEMAS EMBARCADOS E ROBOTICA MÓVEL

Jorge Fredericson de Macedo Costa da Silva – jf.engtelecom@gmail.com

Raif Carneiro Gomes – raif.carneiro@gmail.com

Adson Oliveira França Nascimento – adson.ofn@gmail.com

Daniel Matias Silva dos Santos – danieljava6@gmail.com

Katielle Dantas Oliveira – katielledantas@gmail.com

José Wally Mendonça Menezes – wally@ifce.edu.br

Instituto Federal de Educação, Ciência e Tecnologia do Ceará, IFCE.

Endereço Av. Treze de Maio, 2081. Benfica

CEP 60040-531 – Fortaleza – Ceará

Resumo: *O trabalho apresenta o desenvolvimento de um veículo terrestre teleoperado para explorar os conceitos presentes nas áreas de Sistemas Embarcados e Robótica Móvel. Esse trabalho foi desenvolvido com base na Plataforma de Desenvolvimento Plataforma de Desenvolvimento “Open Source”, Arduino, que além de ser voltada para os iniciantes nas áreas citadas tem, comparativamente a outras plataformas de desenvolvimento, tem um baixo custo. Outro fator para o uso do Arduino é a facilidade de conexão com periféricos por meio dos Shields, placas de circuito impresso desenvolvidos para se conectar fisicamente com o Arduino. Ao final do trabalho, verifica-se uma redução no nível de abstração de alguns conceitos expostos em sala/laboratório pelos professores e o consequente maior aprendizado dos discentes. Como um fator positivo, podemos justificar o fato de que o Arduino, que funciona como mediador entre a teoria e a prática, tem como filosofia básica o princípio de facilitar à eletrônica e torná-la mais atraente aos que querem ingressar na área computacional embarcada.*

Palavras-chave: *Arduino, Sistemas Embarcados, Ensino, Robótica Móvel e Eletrônica.*

1. INTRODUÇÃO

Nos últimos anos, a área de Sistemas Embarcados tem avançado de forma muito dinâmica. Esse dinamismo dá-se por diversos aspectos, ora pela convergência das redes de telecomunicações, ora pela convergência digital, ora pela evolução da microeletrônica e, por conseguinte, dos produtos advindos desse meio (sensores e atuadores), além da evolução das linguagens de programação e da própria forma de programar esses sistemas.

Paralelo a isso tem-se o desenvolvimento da Robótica, que faz uso dos sistemas

Realização:



Organização:



**O ENGENHEIRO
PROFESSOR E O
DESAFIO DE EDUCAR**



computacionais embarcados para controlar máquinas e processos industriais. Partindo para a Robótica Móvel, um dos focos desse trabalho, tais sistemas tornam-se ainda mais dependentes desse tipo de tecnologia. Isso ocorre porque eles são responsáveis tanto pela comunicação e locomoção quanto pelo tratamento que será dado ao ambiente no qual o Protótipo Móvel está inserido.

Diante do exposto, o estudante ou profissional dessas áreas necessita de um treinamento objetivo e dinâmico para que seja capaz de discernir e aplicar os conceitos corretos, evitando ao máximo falhas e prejuízos na execução de projetos. É nesse contexto que o Arduíno entra em destaque, posto que ele atende satisfatoriamente a essas necessidades.

2. SISTEMAS EMBARCADOS, ROBÓTICA E FERRAMENTAS DE ENSINO

O treinamento dos alunos/profissionais no ensino de tarefas que exigem o conhecimento de ferramentas de programação, sobretudo para sistemas embarcados, tem suas dificuldades desde a aquisição das ferramentas de trabalho até o aprendizado para uso destas ferramentas. Não obstante a essa causa, encontra-se na literatura vários trabalhos cujo o foco é o desenvolvimento de ferramentas ou metodologias para auxílio no desenvolvimento do conhecimento nessa área.

Os primeiros passos foram dados com Seymour Papert e seu conceito de construcionismo. Seymour Papert desenvolveu uma nova abordagem para “pensar, aprender e resolver problemas” geométricos. Esta abordagem, que ele chamou de *geometria da tartaruga*, se utiliza de ferramentas computacionais. Da mesma maneira que as abordagens lógica de Euclides e algébrica de Descartes, a abordagem computacional de Papert é uma ferramenta muito poderosa para “pensar, aprender e resolver problemas”, não só de geometria, mas, também, de outras áreas do conhecimento (GUEDES, 1998).

A criação de novas formas de inovar no ensino da programação não se limitou à linguagem LOGO, ou linguagem da tartaruga, e com o passar do tempo tais ferramentas ganharam uma nova maneira de interação com os usuários. Um exemplo dessas novas ferramentas é o *Scratch*. O *Scratch* é um programa que permite ensinar conceitos de programação para crianças e permite associar esse conhecimento à criação de projetos multidisciplinares (SCRATCHBRASIL, 2012). A *Scratchboard* ou *PicoBorad*, uma maneira para você fazer projectos *Scratch* responder a coisas que acontecem no mundo fora de seu computador (MIT, 2012).

Uma maneira de tornar o uso do *Scratch* mais acessível é interação do *Scratch* com o Arduíno, feita através do *software Scratch for Arduíno*, ou S4A. O S4A interage com Arduíno por envio e recepção de estado de atuadores e sensores a cada 75 milissegundo, largura de pulso. Esta troca de informações é feita usando o protocolo, mesmo presente na *Picoboard*, e tem que haver um programa específico (chamado de firmware) na placa (CITILAB, 2012).

Um outro fator importante para quem deseja desenvolver projetos na área de sistemas embarcados é o uso de um ambiente de programação e compilador. O ambiente open-source *Arduino* torna fácil escrever o código e enviá-lo à placa de i / o. Ele roda em Windows, Mac OS X e Linux. O ambiente é escrito em Java e baseado em Processing, avr-gcc e outros softwares de código aberto (ARDUINO, 2011).

A robótica entra nesse contexto de uma forma intrigante, porque no início de seu desenvolvimento ela fazia uso desde sistemas mecânicos até os sistemas computacionais. Essa evolução de como fazer um robô tem se confrontado atualmente seja vista com um outro enfoque, os robôs autônomos, mas como ensinar aos alunos como construir esses protótipos?

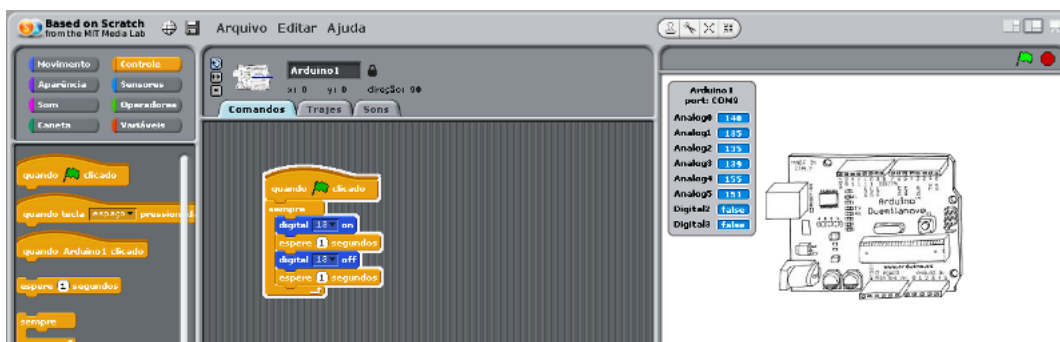


Figura 1 – Exemplo de um Pisca Led em *Scratch for Arduino*.

A robótica educacional, atualmente, é servida por vários produtos de acordo com a faixa etária e do contexto pedagógico que se deseja trabalhar. Existem brinquedos pedagógicos com eletrônica de controle, kits educacionais com foco em alunos do ensino fundamental e ensino médio. E há conteúdo didático e competições utilizando kits de montagem robótica e até robôs móveis inteligentes de pequeno porte para o nível técnico e de graduação, que também podem ser aplicados em pesquisas por alunos de pós-graduação (XBOT, 2012).

3. METODOLOGIA

A metodologia do trabalho está segmentada da seguinte forma: Plataforma de Desenvolvimento Arduino, conceitos explorados nas áreas de Sistemas Embarcados e de Robótica e Robo Móvel controlado pelo *Arduino*.

3.1. Plataforma de Desenvolvimento Arduino

O Arduino é uma plataforma livre, *Open Source*, para desenvolvimento de circuitos computacionais embarcados destinado a pessoas com pouca experiência em eletrônica e em desenvolvimento de softwares. Valendo-se dessa filosofia base é possível inserir os estudantes desde os seus primeiros contatos com a eletrônica e com a programação nesse contexto e, assim, dar maior objetividade e dinamismo ao longo do processo de aprendizagem. Além disso, o Arduino é usado tanto em programas de ensino quanto no mercado em várias partes do mundo em diversas áreas (MARGOLIS, 2011).

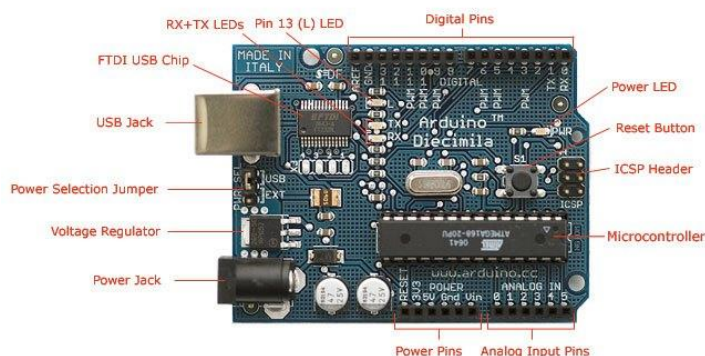


Figura 2 – *Hardware* Arduino com Atmega328.



A Plataforma de Desenvolvimento *Arduíno*, Figura 2, faz uso do microcontrolador *Atmega328*, que dá suporte aos recursos físicos e computacionais presentes na placa de desenvolvimento. Os recursos físicos são: portas de entradas e saídas digitais e analógicas, comunicação USB e ICSP (), suporte de alimentação por fonte externa (até 12V em Corrente Contínua), *shields* - placas que podem ser conectadas em cima do placa *Arduíno* estendendo as suas capacidades, onde os *shields* alternativos seguem a mesma filosofia que o kit de ferramentas originais: eles são fáceis de montar e barato de produzir - e LEDs associado ao pino digital 13 e aos pinos de comunicação serial, UART (*Universal Asynchronous Receiver Transmitter*). Já os suportes computacionais são: interfaces de comunicação serial UART, I²C (*Inter-Integrated Circuit*) e SPI (*Serial Peripheral Interface*), timer *Watchdog*, seis canais de PWM, um circuito *Real Time Clock* (RTC), dois timers contadores de 8 bits e um de 16 bits, conversores analógicos-digitais com 10 bits de precisão e pino de *sleep* ou *wake-up*.

Do ponto de vista do software, o *Arduíno* é multiplataforma de fácil programação na implementação de projetos devido à sua *Integrate Development Environment* (IDE). Essa IDE, Figura 3, é baseada nos projetos *Open Source Processing* e *Wiring*, com isso, ela visa facilitar o trabalho de quem não tem familiaridade com desenvolvimento de *software*. Ainda possui editor de código com recursos de realce de sintaxe, parênteses correspondentes e indentação automática, sendo capaz de compilar e carregar programas para a placa, portanto não há a necessidade de editar *Makefiles* ou rodar programas em ambientes de linha de comando. Por fim, a biblioteca *Wiring* dá a capacidade de programar em C/C++, logo permite criar com facilidade muitas operações de entrada e saída, tendo que definir apenas duas funções no pedido para fazer um programa funcional, as funções “*setup(void)*” e “*loop(void)*” (NOBLE, 2009).

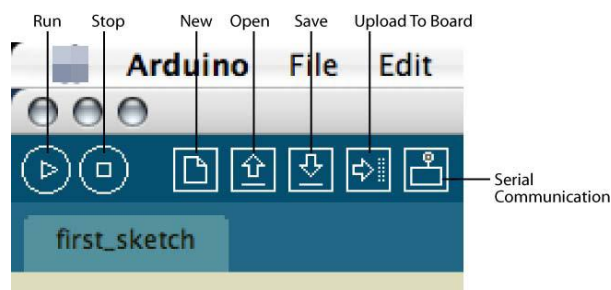


Figura 3 – IDE do *Arduíno*.

O conjunto desses fatores, plataforma livre e objetividade na hora de programar, dão suporte a uma série de experimentos e opiniões variadas, além de uma gama de trabalhos tanto na Internet quanto em livros didáticos. A grande quantidade de materiais disponíveis permite a troca de ideias em diversas áreas do conhecimento no mundo dos sistemas embarcados, robótica, técnicas de programação, eletrônica, dentre outras.

3.2. Conceitos Explorados em Sistemas Embarcados e Robótica

Esse estudo de caso foi feito com base nas ementas das disciplinas de Sistemas Embarcados e Robótica dos cursos Engenharia de Instituições de Ensino conceituadas e que encontram-se no protótipo montado ao final deste trabalho. A partir disso, elenca-se alguns termos recorrentes ao longo do desenvolvimento de Sistemas Embarcados em geral, sobretudo quando aplicamos esses sistemas a atividades de robôs fixos ou móveis. Esses



conceitos podem ser divididos entre implementação via *software* e montagem de circuitos e interfaces com o *hardware* de controle.

Os conceitos com implementação a nível de *software* são: controle de motores DC por Modulação por Largura de Pulso, PWM (*Pulse Width Modulation*), comunicação por meios sem fio *Zigbee* e *Bluetooth* através de interfaces seriais, além do infravermelho, e controle de sensores e atuadores. Já os conceitos de montagem física são: controle da rotação dos motores por ponte H e circuito regulador de tensão.

3.2.1. Modulação por Largura de Pulso (PWM)

A Modulação por Largura de Pulso é uma técnica usada para controlar motores de corrente contínua, DC. Ela trabalha com sinais pulsados ao longo do tempo, controlando o tempo do sinal em nível lógico alto para que o seu valor médio ao final do período seja proporcional a frequência de rotação dos motores DC. No *Arduíno*, essa técnica pode ser feita de duas formas, com o uso da função “`analogWrite(int valor)`” – Figura 4 – ou adicionando atrasos entre sequências de comandos nos pinos digitais – forma usada neste trabalho.

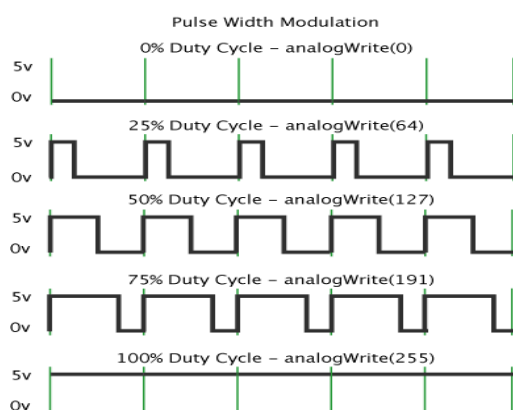


Figura 4 – PWM no Arduíno.

3.2.2. Comunicação Serial UART e Infravermelho (IR)

A comunicação do protótipo é feita com base na interface de comunicação serial UART e por meio de infravermelho. A comunicação UART consistem no envio e recepção dos dados através dos pinos “0” (Rx) e “1” (Tx) do *Arduíno*, sendo que cada um desses pinos tem um led associado para indicar recepção ou transmissão de dados oriundos dessa forma de comunicação. Os pinos de UART do *Arduíno* são usados para interfaciá-lo com os módulos *Zigbee X-Bee Pro S2B* e *Bluetooth* usados no projeto. O *Arduíno* faz a leitura e dá o devido tratamento a esses dados vindos da UART por meio das funções “`Serial.read(void)`” e “`Serial.available(void)`”.



Figura 5 – Módulos *Bluetooth* e *Zigbee* usados para Comunicação.



Tem-se ainda o uso do infravermelho para controle da locomoção do Veículo Teleoperado. Ele foi o primeiro método sem fio testado para verificar a locomoção, porque ele é de simples implementação e possui um custo acessível, além disso, a presença desse receptor de infravermelho podemos controlar o protótipo a partir de qualquer controle remoto que faz uso de um emissor de infravermelho. Esse controle via infravermelho é feito com base numa biblioteca, com código fonte aberto, disponível na comunidade que faz uso do *Arduíno*, (ARDUINO, 2011), responsável pela conversão do sinal óptico do emissor IR em sinal elétrico pulsado que é “compendido” pelo microcontrolador do *Arduíno*. Segue o esquema de conversão na Figura 6.

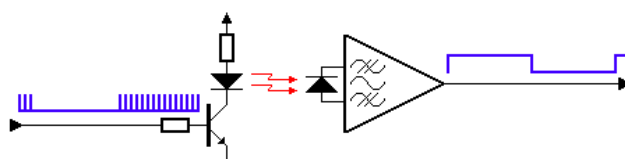


Figura 6 – Funcionamento do Receptor de Infravermelho.

3.2.3. Controle de Sensores e Atuadores

O papel dos sensores e dos atuadores tanto em Sistemas Embarcados quanto na Robótica são de suma importância, pois é a partir desse par que o sistema pauta o seu funcionamento. A utilização desse par junto com o *Arduíno* é de fácil entendimento e montagem, posto que a filosofia inicial do *Arduíno* é tornar isto “amigável” aos iniciantes na área. Em geral, os sensores são ligados nos conversores analógico-digital (ADC), portanto necessitam de um fator para calibração e ter seus dados lidos corretamente. Já os atuadores, em geral, não ligados nas portas digitais não necessitando, assim, de conversões ou calibrações contínuas ao longo do tempo de funcionamento.

O *Arduíno* possui seis conversores ADC com resolução de dez bits, logo ele associará os níveis de tensão de entrada nesse ADCs entre os valores de 0 até 1023. Essa associação de valores é feita com base num fator de conversão é dado por uma regra de três simples. No *Arduíno*, os sensores são tratados pela função “`analogRead(int ADC)`” e os atuadores pelas funções “`digitalRead(int Pin)`” e “`digitalWrite(int Pin)`”, mostados na Figura 7.

```
sketch_jun10a | Arduino 0022
File Edit Sketch Tools Help
sketch_jun10a $
const int ledPin = 13; // LED conectado ao pino digital 13.
const int sensorPin = 0; // sensor conectado na entrada analogica 0.

void setup()
{
  pinMode(ledPin, OUTPUT); // LED como saída do sistema.
  Serial.begin(9600); //Comunicação Serial com 9600 bps de baudrate.
}

void loop()
{
  int rate = analogRead(sensorPin); // Lê o valor da entrada analogica
  Serial.println(rate);
  rate = map(rate, 200,800,minDuration, maxDuration); // fator de conversão
  digitalWrite(ledPin, HIGH); // LED on
  delay(rate); // espera dependendo no nivel de tensao do sensor
  digitalWrite(ledPin, LOW); // LED off
  delay(rate);
}
```

Figura 7 – Uso de Sensores e Atuadores no *Arduíno*.



3.2.4. Ponte H

A Ponte H é um esquema elétrico que permite o controle de motores DC pelos microcontroladores. Ela é necessária porque os microcontroladores em seus pinos de saídas digitais não conseguem fornecer a corrente elétrica necessária para o acionamento desses motores, além disso, ela ajuda na inversão do sentido de rotação dos motores por meio do chaveamento de seus transistores, Figura 8, entretanto no presente trabalho o grupo optou por usar um circuito integrado, L293D, no qual há duas pontes encapsuladas no mesmo *chip*.

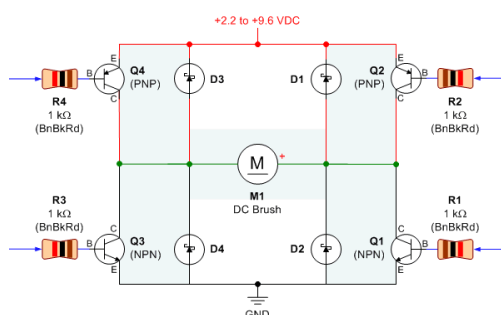


Figura 8 – Exemplo de uma Ponte H.

3.2.5. Circuito Regulador de Tensão

O regulador de tensão é um componente que mantém a tensão de saída constante, independente da variação de entrada e do consumo de saída, dentro de certos limites (OLIVEIRA, 2006). Geralmente, os circuitos reguladores usados são encapsulados em circuitos integrados. O esquema de ligação desses circuitos é mostrado na Figura 9.

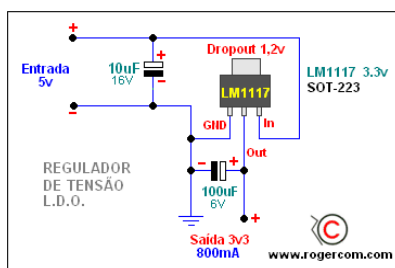


Figura 9 – Circuito Regulador de Tensão Positiva com LM117.

3.3. Robô Móvel Controlado pelo Arduino

A execução do projeto pautou-se na aquisição de modelo já pronto de baixo custo. Depois disso, foi removido a parte de circuito presente neste modelo e no lugar dela foi posta uma nova placa desenvolvida pelos proponentes deste trabalho. Essa nova placa de controle passou por duas etapas: a parte de montagem física da placa e a programação, a construção, do algoritmo de controle. Ao longo do desenvolvimento do trabalho foram explorados os conceitos descritos nos tópicos e subtópicos do item 3. Tais conceitos são de grande relevância no desenvolvimento das disciplinas alvo desse estudo, os Sistemas Embarcados e a Robótica, posto que esses conceitos tem bastante destaque para os aluno e profissionais.



O projeto foi feito para explorar conceitos e produzir conhecimento, todavia pode facilmente ser adaptado a uma série de áreas acadêmicas e comerciais. Na esfera acadêmica, o protótipo está sendo integrado com um trabalho sobre rede de sensores sem fio (RSSF), onde ele será responsável por ser coletor de dados advindos dos nós dessa RSSF e na esfera comercial, foi desenvolvido um Plano de Negócios junto a incubadora de empresas da Rede de Incubadoras do Ceará (RIC) com foco na produção de robô autônomo de propósitos gerais (RIC, 2011).

Por fim, esse tipo de trabalho mostra que é viável formar provas de conceitos, minimizando as dúvidas dos alunos, com um custo acessível. Essas dúvidas surgem devido ao nível de abstração exigido pelos conceitos computacionais aliados a natureza física presente na montagem dos circuitos eletrônicos, portanto é importante pensar nas duas vias do conhecimento, os docentes e discentes, quando visamos uma melhoria na qualidade do ensino tecnológico.

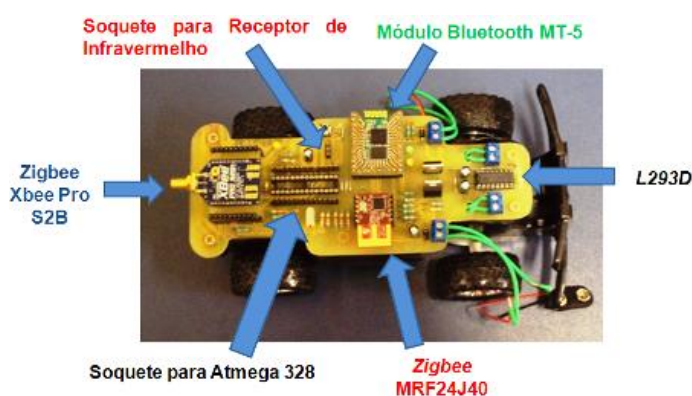


Figura 10 – Modelo montado ao final do Trabalho.

4. ANÁLISE DOS RESULTADOS

O trabalho foi realizado inicialmente com quatro alunos do quarto semestre do curso superior em Engenharia de Telecomunicações do IFCE, alcançando o objetivo traçado. Como resultado imediato do experimento, houve o incentivo em expor o projeto aos outros alunos das engenharias e de outros cursos, além da troca de experiências ocorridas em eventos no IFCE e outras instituições de ensino superior. Num segundo momento, foi acompanhando dois alunos do segundo semestre e notou-se que eles obtiveram êxito mais rapidamente devido à integração com os membros mais antigos e logo foram deslocado para execução de um projeto na área de Sistemas Embarcados.

Por fim, o projeto está no processo de acompanhamento de sua terceira turma composta por seis alunos de semestres diversos do curso de Engenharia de Telecomunicações, mantendo o foco na melhoria da formação dos alunos ao longo do período de graduação.

5. CONCLUSÃO

Ao final das etapas deste trabalho vislumbra-se a necessidade dos alunos por em prática a teoria vista em sala não só pela questão do mercado de trabalho, mas sim pela necessidade de ver que seu conhecimento gerar algo produtivo e, principalmente, quando esta produção é inovadora. Tal fato ganha relevância a medida que surgem no mercado várias soluções de



baixo custo e com grande potencial para ensinar e executar essas ações práticas ao longo da formação dos futuros engenheiros e cientistas.

REFERÊNCIAS

Livros:

MARGOLIS, Michael. Arduino Cookbook. 1. ed. USA: O' Reilly 2011. 632p, il.

NOBLE, Joshua. Programming Interactivity. 1. ed. USA: O' Reilly 2009. 713p, il.

OLIVEIRA, A. S; Andrade, F. S. Sistemas Embarcados Hardware e Firmware na Prática. 1. ed. São Paulo: Erika, 2006. 315p, il.

ORDONEZ, E. D. M; PENTEADO, C. G; SILVA, A. C. R. Microcontroladores e FPGAs Aplicações em Automação. 1. ed. São Paulo: Novatec, 2005. 378p, il.

Monografias, dissertações e teses:

GUEDES, R. B. M. UNIVERSIDADE FEDERAL DE PERNAMBUCO, Centro de Ciências Exatas e da Natureza. INTELIGÊNCIA COMPUTACIONAL: Métodos Procedimentais para Pensar, Aprender e Resolver Problemas, 1998. 202p, il. Dissertação (Mestrado).

Internet:

Arduíno. **Arduíno Software**. Disponível em:
<<http://arduino.cc/hu/Main/Software>> Acesso em: 03 mar. 2011.

Arduíno. **IR Library**. Disponível em:
<<http://www.arduino.cc/cgi-bin/yabb2/YaBB.pl?num=1295618674>> Acesso em: 10 fev. 2011.

Citilab. **Scratch for Arduino**. Disponível em:
<<http://seaside.citilab.eu/scratch/arduino>> Acesso em: 25 mai. 2012.

Massachusetts Institute of Technology (MIT). **Scratch Sensor Board**. Disponível em:
<http://info.scratch.mit.edu/Support/Sensor_Boards> Acesso em: 02 jun. 2012.

Rede de Incubadoras do Ceará (RIC). **Desafio de Plano de Negócios**. Disponível em:
<<http://www.ric.org.br/noticias/44-resultado-1o-etapa-do-concurso-desafio-de-plano-de-negocios-2011.html>> Acesso em: 20 nov. 2011.

Scratch Brasil. **O que é o Scratch?** Disponível em:
<<http://scratchbrasil.wordpress.com/>> Acesso em: 08 jun. 2012.

Xbot. **Apostila de Robótica**. Disponível em:
<http://www.xbot.com.br/downloads/apostila_Robotica_1-0.pdf> Acesso em: 22 nov. 2012.



BUILDING TELEOPERATED VEHICLE WITH ARDUÍNO TO HELP IN TEACHING EMBEDDED SYSTEMS AND MOBILE ROBOTICS

Abstract: *This paper presents the development of a teleoperated land vehicle to explore the concepts in the areas of Embedded Systems and Mobile Robotics. This work was developed based on Developer Platform Development Platform "Open Source", Arduino, which besides being aimed at beginners in the areas cited have compared to other development platforms, has a low cost. Another factor to use the Arduino is the ease of connecting peripherals through Shields, printed circuit boards designed to physically connect to the Arduino. At the end of the work, there is a reduction in the level of abstraction of concepts exposed in the classroom / laboratory teachers and the consequent higher learning of students. As a positive factor, we can justify the fact that the Arduino, which acts as a mediator between theory and practice basic philosophy is the principle of facilitating the electronics and make it more attractive to those who want to join in the embedded computing.*

Keywords: *Arduino, Embedded Systems, Teaching, Mobile Robotics & Electronics.*