

A PLATAFORMA ARDUÍNO E SUAS APLICAÇÕES

Eduardo Henrique Ferroni¹, Hugo Rodrigues Vieira¹, Jaderson Henrique Nogueira¹, Rafael Kerner Coelho dos Santos¹, Renato Martins Lemos¹, Thiago Baldin Rodrigues¹

¹Centro Universitário do Sul de Minas UNIS-MG

RESUMO

O presente artigo apresenta a plataforma Arduino, a qual consiste em um sistema que permite a automatização de processos por meio de comandos programáveis. Inicialmente é descrita a história, onde é informado o conceito do Arduino, suas características físicas e de software, os módulos capazes de ampliar sua aplicabilidade e exemplos de aplicações. Por fim, são mostrados os benefícios e a possibilidade de aplicar conhecimentos técnicos em busca da melhoria da qualidade de vida bem como a otimização de processos, seja a nível pessoal ou profissional.

Palavras-chave: Arduino, Automação, Controle, Engenharia, Programação, Tecnologia.

ABSTRACT

The present paper introduces the Arduino platform, a system that allows the automation of processes by using programmable commands. From the history of the platform, will be described the Arduino concepts, its physical and software features, the modules capable of extending its applicability and application examples. Finally, it will conclude on the benefits and the ability to apply technical knowledge, improving the quality of life and processes optimization, either in personal or professional levels.

Keywords: Arduino, Automation, Control, Engineering, Programming, Technology.

INTRODUÇÃO

Em um mundo globalizado e com a tecnologia cada vez mais democratizada e presente na vida das pessoas, o homem se torna cada vez mais dependente destes avanços tecnológicos. No entanto, esta dependência se dá sempre na condição de “usuário final”. O ser humano em geral consome a tecnologia pela compra de aparelhos já prontos e com funcionalidades específicas e, mesmo com algum conhecimento técnico, os consumidores nem sempre conseguem interagir de maneira mais aprofundada e técnica com as tecnologias.

Este artigo apresenta a plataforma Arduino, um sistema capaz de permitir a um usuário com conhecimentos técnicos interagir com o ambiente através de dispositivos e equipamentos. A plataforma Arduino conta com uma interface programável por computador, a qual permite a configuração de ações que serão realizadas por diversos dispositivos a ele conectados, seja uma tela de cristal líquido (LCD), um braço mecânico, ou até mesmo um sensor sem fio.

Para que fique claro o potencial de uso da plataforma, faz-se relevante conhecer sua história.

HISTÓRIA DO ARDUINO

Segundo RODRIGUES (2012), o Arduino foi criado pelo professor Massimo Banzi na Itália, o qual queria ensinar programação de computadores de forma que este conhecimento pudesse ser aplicado a projetos de arte, automação e robótica. A grande dificuldade encontrada por ele era justamente não encontrar no mercado placas que fossem didáticas e ao mesmo tempo poderosas e baratas.

Com o tempo, juntamente com o engenheiro espanhol David Cuartielles, Massimo decide criar sua própria placa contando com ajuda de um de seus alunos chamado David Mellis, o qual ficou responsável pela linguagem de programação do Arduino.

O hardware Arduino foi um sucesso, visto a ampla possibilidade de aplicações, como por exemplo, um leitor de temperatura e controle de velocidade por pulso de motores de corrente contínua (UNESP, 2012). Assim, o conceito do Arduino logo se espalha pelo mundo, atingindo a marca de mais de 50.000 placas vendidas até outubro de 2008, (BARROS, 2012).

A PLATAFORMA

Arduino

Para MCROBERTS (2010), Arduino é um sistema embarcado, ou seja, que pode interagir com seu ambiente por hardware e software incorporados a um dispositivo com um objetivo pré-definido. Trata-se de um projeto de código livre que pode ser “clonado” tanto em software (que utiliza linguagem de programação em C/C++) quanto em hardware.

O Arduino, assim como um CLP (controlador lógico programável), controla sistemas industriais. A plataforma tem como principal finalidade facilitar, a nível doméstico, comercial ou móvel, a automação e controle nestes ambientes.

O Arduino também pode ter suas aplicações estendidas utilizando placas que contêm outros dispositivos, as quais são facilmente conectadas a ele. Estas placas são chamadas de módulos ou *Shields* (escudos, em inglês). Tais placas podem funcionar como receptores GPS, módulos de rede ethernet ou wireless, dentre outros, (MCROBERTS, 2010).

Existem diversos modelos de placa da plataforma Arduino. A Figura 1 apresenta uma versão recente e acessível do Arduino, chamada Arduino UNO:



Figura 1 – Arduino UNO (DESTACOM, 2012)

A Figura 2, por sua vez, ilustra um modelo mais antigo da placa Arduino, chamado Duemilanove, referenciado por autores como Simon Monk e Michael McRoberts.

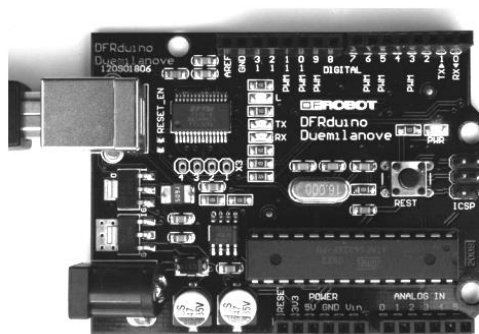


Figura 2 – Arduino Duemilanove (MONK, 2010)

Já a Figura 3 demonstra o aspecto do Arduino LilyPad, mais flexível e que possibilita a fixação em tecidos, para utilização em vestíveis.

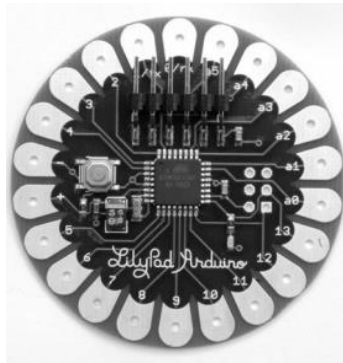


Figura 3 – Arduino LilyPad (MONK, 2010)

Hardware – Arduino Duemilanove

Para os fins aos quais o Arduino foi idealizado, deve-se fundamentalmente compreender a placa na qual a plataforma é estruturada e, principalmente, seu microcontrolador.

Segundo MONK (2010), a placa em si existe tão somente para prover o suporte estrutural para o funcionamento do microcontrolador, permitindo que seus terminais se comuniquem com outros dispositivos, além de permitir a transmissão de informação através de portas de acesso à placa.

Componentes cuja finalidade se resume à alimentação do microcontrolador, condução dos seus sinais às saídas e demais funções estritamente estruturais serão desconsiderados. Neste artigo são tratados componentes cuja compreensão é fundamental para o uso adequado do Arduino. São eles:

Alimentação:

Próximo à porta serial universal (USB) da placa existe um regulador de tensão de 5 V, de modo que variações na tensão de entrada são ajustada para o valor de tensão citado.

Normalmente utiliza-se um regulador de tensão com grande tamanho físico quando comparado aos demais componentes da placa, com o intuito de favorecer à dissipação de calor gerado na regulação da tensão, (MONK, 2010).

Conectores:

Ao observar os cantos inferior e superior da placa, os quais são mostrados na Figura 4, é possível identificar conectores do tipo "fêmea" para ligação do Arduino com outros periféricos.

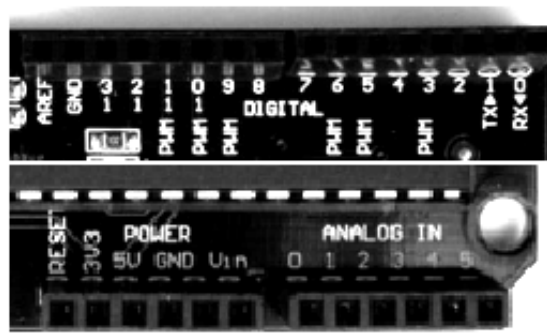


Figura 4 – Conectores - Arduino Duemilanove (MONK, 2010)

A seguir são descritas as funções de cada conector:

- RESET: Quando energizado em 5 V, reinicializa o microcontrolador.
- GND: Conector com tensão 0 V (aterramento), usado como referência de tensão nula para os demais conectores.
- 3V3 (3,3 V), 5 V, V_{in} : Conectores de tensão, utilizados para levar alimentação aos módulos conectados à placa.
- Entradas Analógicas (de 0 a 5): Conectores de entrada que recebem sinais analógicos ligados a cada um deles. A corrente que circulará por eles é pequena, uma vez que a resistência interna destes circuitos é grande, resultando em informações que poderão ser utilizadas de acordo com o projeto no qual o Arduino for aplicado.
- Conexões Digitais (0 a 13, GND e AREF): As conexões digitais 0 a 13 são conexões de entrada e saída. Especificamente as conexões 0 e 1 possuem a indicação RX e TX respectivamente, que são conexões seriais de recepção e transmissão de dados.

Em geral, todas as conexões digitais numeradas trabalham com valores absolutos de tensão, de 0 V ou 5 V. A conexão digital GND possui a função de aterramento (referência para 0 V), e a conexão AREF pode ser utilizada para medição precisa de tensões ligadas às entradas analógicas, usando como base a tensão de referência, que neste caso é a tensão de alimentação do próprio Arduino (para a maioria dos modelos de placas, é 5 V).

Ainda sobre as conexões digitais, as conexões 3, 5, 6, 9, 10 e 11 são sinalizadas com as letras PWM e podem prover uma tensão de saída variável, diferente dos valores absolutos 0 V e 5 V das demais conexões digitais.

Microcontrolador: Segundo MCROBERTS (2010), a essência do Arduino é um microcontrolador AVR da ATMEL, sendo usado nos modelos mais recentes o ATMEGA328P-PU da referenciada fabricante (Figura 5).

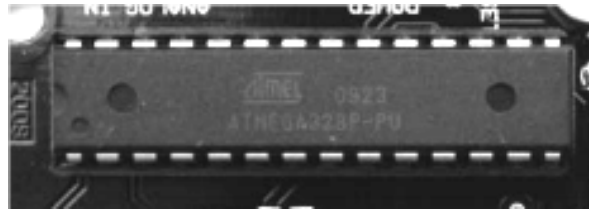


Figura 5 - Microcontrolador ATMEGA328P-PU (MONK, 2010)

Os circuitos, layout e componentes da placa são projetados de modo que facilite o acesso de dispositivos externos aos terminais deste microcontrolador permitindo seu uso. O Microcontrolador é fixado na placa através de um soquete DIL (dual in-line), facilitando sua remoção e substituição.

MONK (2010) considera este microcontrolador o coração do Arduino por controlar todo o funcionamento e uso do dispositivo. É possível programá-lo de modo a determinar todas as tensões de entrada, saída, e a lógica dos sinais gerados em cada um de seus terminais, oportunizando o uso para quaisquer fins aos quais é direcionado.

A Figura 6 mostra o diagrama de blocos do microcontrolador ATMEGA328, bem como todos os seus componentes.

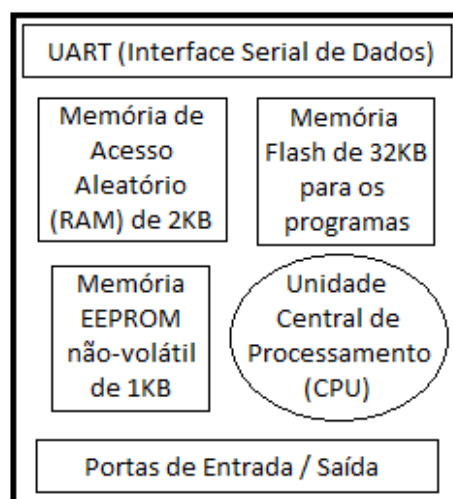


Figura 6 - Diagrama de blocos do ATMEGA328 (MONK, 2010).

Internamente o microcontrolador possui uma memória EEPROM (memória de somente-leitura programável e eletricamente apagável), utilizada para a gravação de dados como instruções de programas, sem o risco de que estes dados sejam perdidos no caso de reinicialização (através da

função RESET descrita anteriormente, ou por meio do botão de mesmo nome, localizado na placa), ou falha de alimentação.

Além disso, o microcontrolador também apresenta uma memória RAM (memória de acesso aleatório) que registra as informações geradas durante o uso do dispositivo e que são eliminadas sempre que há a reinicialização ou desligamento do equipamento.

Oscilador: Outro componente fundamental para o funcionamento do Arduino localiza-se próximo ao botão RESET. Trata-se de um oscilador formado por um cristal de quartzo estruturalmente posicionado entre dois eletrodos que, ao emitirem impulsos elétricos alternados a uma determinada frequência, provocam a vibração (oscilação) do cristal, e esta oscilação produz um campo elétrico senoidal que controla a impedância entre os eletrodos, (MATOS e TAVARES, 1998).

O oscilador do Arduino é capaz de vibrar a aproximadamente 16 milhões de vezes por segundo (16 MHz), determinando a quantidade de operações realizadas neste intervalo de tempo (MONK, 2010).

Interface USB: A placa apresenta uma interface USB, através da qual se pode conectar a placa a um computador de modo a possibilitar sua programação e a troca de informações.

Para possibilitar esta comunicação, a placa conta com um chip ATMEL ATMEGA8U2 ou ATMEGA16U2 (cuja diferença se resume no tamanho do pacote de dados enviados e no endereçamento de bytes (ATMEL, 2012), a fim de converter os sinais emitidos pelo dispositivo conectado a esta porta, para níveis ideais para comunicação com o Arduino).

MCROBERT (2010) cita ainda a possibilidade de se reprogramar o firmware deste chip de modo a fazer com que o computador identifique a placa Arduino como um outro periférico (cita-se como exemplo um mouse, ou joystick), ampliando ainda mais os potenciais de uso da plataforma.

Software

As operações realizadas pelo Arduino se resumem em essência a sinais elétricos. A lógica por trás dos impulsos e sinais transmitidos e interpretados pela plataforma dependem de programas inseridos em seu microcontrolador.

De acordo com MARGOLIS (2011), comandos e programas são escritos no computador utilizando um ambiente de desenvolvimento integrado (IDE), que permite escrever e alterar códigos e

convertê-los em instruções que o Arduino consiga compreender e executar, além de permitir a sua gravação no microprocessador.

Em outras palavras, tudo o que será realizado pelo Arduino, seja através do próprio microcontrolador ou de periféricos a ele conectados, deverá ser programado em scripts inseridos por intermédio da IDE, onde são definidas as orientações necessárias para cada execução.

A interface da IDE do Arduino é intuitiva. No entanto, como depende da programação de comandos e execuções a mesma exige do usuário certos conhecimentos de algoritmos e linguagem de programação, em geral, C ou C++. A Figura 6 mostra a interface inicial da IDE do Arduino.

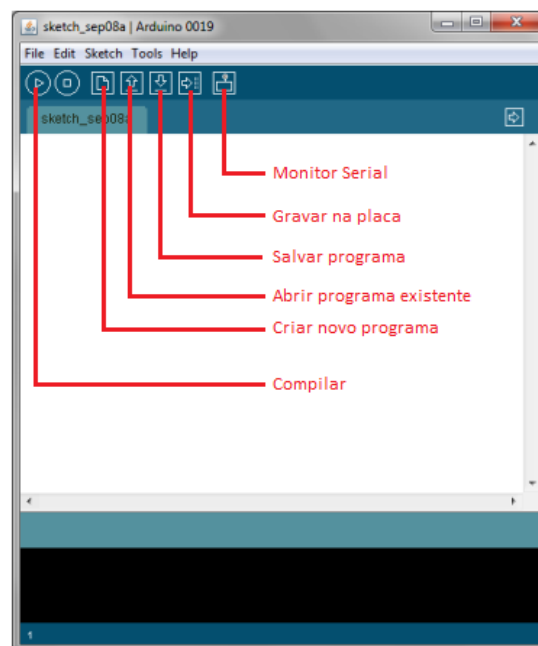


Figura 6 - Tela principal da IDE do Arduino (MARGOLIS, 2011)

Através da IDE é possível projetar, alterar, compilar e gravar programas que comandam o Arduino e qualquer Shield ou módulos a ele conectados.

A IDE do Arduino traz diversos projetos exemplo prontos para serem executados. O programador pode utilizar estes projetos, bem como alterá-los ou usá-los como base para a criação de novos programas. A Figura 7 mostra os caminhos do menu suspenso da interface da IDE que permite a escolha de programas exemplo.

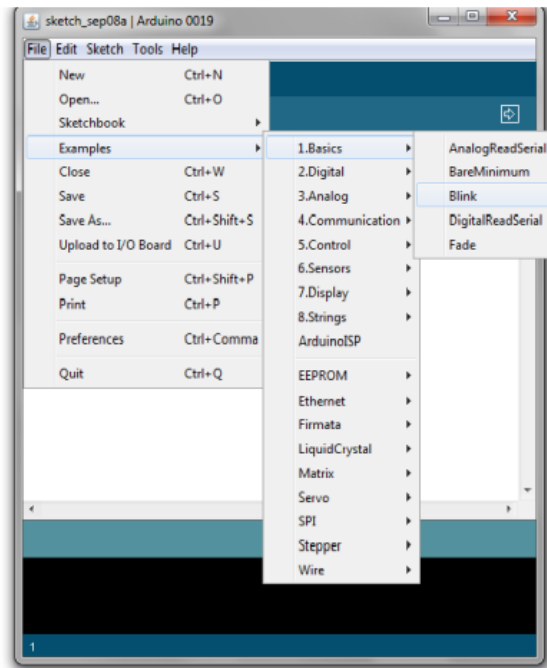


Figura 7 - Selecionando programas-exemplo (MARGOLIS, 2011)

A seguir é transcrito o código do programa de exemplo do tipo Blink, cujo objetivo é fazer com que um diodo emissor de luz (LED) emita luz durante um segundo, e permaneça apagado por mais um segundo, repetindo este ciclo até que a placa seja desligada:

```
int led = 13;

void setup() {
  pinMode(led, OUTPUT);
}

void loop() {
  digitalWrite(led, HIGH);
  delay(1000);
  digitalWrite(led, LOW);
  delay(1000);
}
```

A explicação do código acima é realizada da seguinte maneira: é feita a declaração da variável “led”, que recebe o valor 13, correspondendo ao número do terminal que comandará o LED. Conforme descrito anteriormente, este é um terminal de entrada/saída digital. Logo em seguida, é parametrizada a configuração da rotina geral através do comando void setup(), que é executada

uma única vez quando o sistema é ligado. Neste comando, determina-se que o terminal digital declarado anteriormente será utilizado como saída, através do comando `pinMode(led, OUTPUT)`. A seguir, configura-se uma rotina redundante (loop), que se repetirá até que o dispositivo seja desligado. O comando `digitalWrite(led, HIGH)` indica que a placa fará com que o terminal 13 eleve a sua tensão ao valor “alto” (que neste caso, é 5 V e representa o acionamento do LED). Por fim, o comando `delay(1000)` faz com que o LED permaneça ligado por 1000 milissegundos, para então executar o próximo comando, `digitalWrite(led, LOW)`, reduzindo a tensão do terminal 13 V a 0 V, fazendo com que o LED deixe de emitir luz, e assim permanecendo por mais 1000 milissegundos em função do novo comando `delay(1000)`. Neste momento, toda a rotina reinicia, repetindo os comandos até a interrupção da alimentação da placa.

Após a seleção ou criação do programa a ser gravado no Arduino, deve-se verificar a consistência do programa e a capacidade de converter as instruções para a plataforma através da função “compilar”, e logo após é feita a gravação do programa. Tão logo o programa é gravado, ele já passa a funcionar enquanto o Arduino se mantiver alimentado (a não ser que programe-se o acionamento do programa através de chave).

No entanto, como dito anteriormente, o Arduino em si trabalha com a transmissão, recepção e leitura de sinais elétricos. As aplicações reais da plataforma dependem de outros dispositivos conectados à placa do Arduino através das conexões analógicas e seriais: os Shields.

SHIELDS: AMPLIANDO FUNCIONALIDADES

Embora a plataforma Arduino possua um número extremamente amplo de aplicações, nem sempre é necessária a programação e a criação integral dos circuitos necessários para cada um dos projetos. Segundo MONK (2010), a própria existência do Arduino é uma maneira de facilitar a criação de projetos. Para fins de facilitar ainda mais este processo, o Arduino conta com a possibilidade de se integrar outras placas ou dispositivos cuja programação é feita em microcontrolador independente daquele localizado na placa do Arduino. Estes dispositivos são chamados Shields ou Módulos.

Cita-se como exemplo, a aplicação do Arduino para o controle das informações exibidas em um display LCD, que é detalhada a seguir. O LCD utilizado é um módulo que possui microcontrolador próprio necessário para seu funcionamento, reduzindo assim o volume de trabalho e a quantidade de terminais do Arduino necessários para seu acionamento, (MONK, 2010)

Como outros exemplos, MONK (2010) cita módulos utilizados para a comunicação via Ethernet (cabo de 8 vias utilizado para comunicação em rede física de computadores), comunicação Wireless (rede sem fio), Bluetooth (comunicação sem fio mais utilizada por dispositivos móveis, como celulares ou tablets), GPS (mapeamento e localização), e outras.

Para utilização de Shields no Arduino, é necessária a inclusão de bibliotecas de comandos no código programado através da IDE. Estas bibliotecas tornam o Arduino capaz de reconhecer e interpretar comandos que são específicos para o módulo utilizado.

Existem vários Shields disponíveis para aplicação no Arduino. A seguir, são descritos alguns módulos para aplicação em trabalhos acadêmicos e profissionais:

Módulo LCD Keypad

Inicialmente, apresenta-se o módulo LCD Keypad, demonstrado na Figura 8.



Figura 8 – Módulo LCD Keypad (RAHMAT, 2012)

Trata-se de um módulo compatível com o Arduino que permite ao usuário exibir qualquer texto e inserir dados através de um teclado. Este módulo inclui uma tela de cristal líquido (LCD) de 2x16 pixels com luz de fundo azul e permite exibir 1602 caracteres na cor branca, além de possuir seis botões, que são o de reset, seleção e os direcionais para cima, baixo, direita e esquerda (RAHMAT, 2012).

Os terminais digitais 4, 5, 6, 7, 8, 9 e 10 são usados para interface com o LCD e o terminal analógico 0 é usado para a leitura dos botões.

Os valores de entrada das teclas são lidos através de um divisor de tensão de 5 estágios.

Além disso, este módulo possui um potenciômetro que permite controlar o contraste do LCD.

Módulo Wifi

O Módulo Wifi (Figura 9) permite uma placa Arduino se conectar a internet utilizando a especificação para comunicação sem fio (Wifi) 802.11.



Figura 9 – Módulo WiFi (GRIMMETT, 2014)

Baseia-se no sistema de empacotamento de dados de rede wireless 802.11 b/g do chip HDG104. Trabalha através do chip ATmega 32UC3 com um protocolo de controle de transmissão (TCP/IP), que se baseia em um modelo de referência de quatro camadas, ou seja, níveis de processamento de dados em uma rede, (Microsoft, 2014).

Há também na placa do Shield um soquete para cartão SD que pode ser usado para armazenar arquivos a serem utilizados na rede.

Módulo Xbee

Por fim, apresentamos o módulo Xbee (Figura 10). Shield que permite a comunicação sem fio entre Arduinos utilizando o Zigbee, um padrão global de comunicação sem-fio entre dispositivos eletrônicos com ênfase na baixa potencia de transmissão e taxa de transferência de dados, (ARDUINO, 2014)



Figura 10 – Módulo Xbee (FALUDI, 2011)

Seu alcance é de aproximadamente 30 metros em locais fechados ou 90 metros em locais abertos (com visada) podendo ser usado em substituição A uma interface USB, permitindo sua configuração através de modo de comando.

No Xbee existem dois jumpers para determinar como a comunicação serial do Shield será feita com microcontrolador (ATmega 8 ou ATmega 16 e o chip USB na placa Arduino). Esta seleção pode ser feita através de três terminais rotulados como Xbee/USB.

Para a comunicação adequada entre dois módulos Xbee, estes devem fazer parte de uma mesma rede através de seus parâmetros de protocolo de conexão (IP), devem estar no mesmo canal configurado no parâmetro Channel (CH), além de terem um mesmo endereço de destino configurado.

APLICAÇÕES PARA O ARDUINO

A forma de utilização dos usuários do Arduino depende muito mais da criatividade do usuário do que limitações técnicas ou de conhecimento em eletrônica ou em programação.

De acordo com MARGOLIS (2011), o Arduino permite a construção de objetos que respondem a sinais luminosos, controlem intensidade de luzes, emitam ruídos, executem movimentos, entre outros.

Suas aplicações são as mais diversas, que vão da automação industrial no acionamento de motores à diversão, passando pela automação residencial até a melhoria de vida e conforto.

A seguir, são citados três exemplos de aplicação do Arduino, tanto para níveis pessoais quanto profissionais:

Exemplo 1: Comunicação Wireless. (MARGOLIS, 2011).

A plataforma Arduino é capaz de ampliar significativamente as maneiras de utilização de diversas tecnologias e recursos. No entanto, para que isso ocorra normalmente é necessária a existência de fios que conduzem os sinais elétricos.

Através da utilização do Shield Xbee, é possível realizar a comunicação entre o Arduino e outros dispositivos remotamente. Existem vários modelos de módulos Xbee no mercado, e em geral sua aquisição e aplicação em projetos possui baixo custo, permitindo o uso doméstico em experimentos acadêmicos.

Com isso, é possível instalar um sensor de temperatura, ou um receptor qualquer de sinais em um local e, remotamente, receber as informações destes dispositivos no computador, ou outro dispositivo.

Exemplo 2: Display LCD (MARGOLIS, 2011).

Essa aplicação é bastante útil em ambientes onde há a necessidade de comunicação visual, como em recepção de empresas ou demais instituições. Nessa aplicação o Arduino é capaz de comunicar com uma tela LCD, enviando uma mensagem que será exibida de acordo com a informação programada via código.

Aqui é fundamental que se compreenda a estrutura do módulo LCD, uma vez que a informação será escrita em um display que possui números particulares de linhas e colunas. O código deverá ser escrito observando as especificações técnicas do Shield.

Exemplo 3: Sensor de Temperatura (MCROBERTS, 2010).

Através do Arduino, é possível a leitura de temperatura e exibição através da própria interface IDE da plataforma. Para isso, utiliza-se um sensor de temperatura instalado no local onde a leitura será feita, e através de configurações realizadas por meio de programação a placa recebe informações do sensor e as traduz em números, configurados através da IDE de modo a corresponderem a valores de graus, seja em Celsius, Kelvin ou Fahrenheit. Neste exemplo, ainda é possível combinar os exemplos dados anteriormente, resultando na exibição remota da temperatura em um display LCD que se comunica via wireless com o Arduino.

CONCLUSÃO

Conectando motores, sensores, displays, antenas, dispositivos de entrada e outros periféricos ao Arduino, as possibilidades de aplicação da plataforma são inúmeras. Ele não apenas permite que qualquer pessoa crie novos dispositivos, mas também os torna acessíveis a todos, independente da vontade ou dos planos comerciais das grandes empresas.

O Arduino possui uma aplicabilidade imensa quando se pensa no potencial de automatização de processos desta plataforma, seja em uma esfera particular, acadêmica ou comercial, conseguindo beneficiar e otimizar o desenvolvimento das mais variadas atividades em diversos níveis.

Embora exija algum conhecimento técnico, toda a literatura disponível sobre a plataforma é amplamente pedagógica, uma vez que o próprio conceito e a ideia original do Arduino se fundamentam no compartilhamento do conhecimento.

A plataforma é de uso livre e não comercial, e todos os usuários a utilizam com a tendência de compartilhar os resultados das pesquisas, contribuindo sempre para uma maior divulgação do Arduino.

São grandes os potenciais de aplicação da plataforma. Na residência, na escola, ou no mercado, o Arduino representa mais uma maneira de se utilizar a tecnologia em benefício de seus usuários e da comunidade para a qual seus projetos são idealizados.

REFERÊNCIAS

ARDUINO. Xbee Shield. Disponível em <http://arduino.cc/en/Main/ArduinoXbeeShield>. Acesso em 06/09/2014.

ATMEL Corporation. 8-bit AVR Microcontroller with 8/16/32Kbytes of ISP Flash and USB Controller datasheet. San Jose. 2012.

BARROS, Wagner B. Sistema de Automação Veicular com Arduino e Andróid. 2012. 58f. Trabalho de Conclusão de Curso (Tecnólogo em Sistemas para Internet) - Centro Universitário Adventista de São Paulo, São Paulo, 2012.

FALUDI, Robert. Building Wireless Sensor Networks. O'Reilly. Sebastopol: 2011.

GRIMMETT, Richard. Arduino Robotic Projects. Packt Publishing Ltd. Birmingham: 2014.

MARGOLIS, Michael. Arduino Cookbook - Recipes to Begin, Expand and Enhance Your Projects. O'Reilly. Sebastopol: 2011.

MCROBERTS, Michael. Beginning Arduino. Apress. Nova Iorque: 2010.

MICROSOFT Corporation. O Modelo TCP/IP. Disponível em [http://technet.microsoft.com/pt-br/library/cc786900\(v=ws.10\).aspx](http://technet.microsoft.com/pt-br/library/cc786900(v=ws.10).aspx). Acesso em 30/08/2014.

MONK, Simon. 30 Arduino Projects for the Evil Genius. McGraw-Hill. Nova Iorque: 2010.

RAHMAT, Razzis B. Temperature Alert System in Refrigerator Using Short Message Service (SMS). Universiti Teknologi Malaysia. Kuala Lumpur, 2012.

RODRIGUES, L et al. Introdução ao Arduino. Fundação Universidade Federal de Mato Grosso do Sul - UFMS. Campo Grande: 2012.

TAVARES, João M.; MATOS, João N. Osciladores para Rádio Frequência de Elevada Estabilidade. Revista do DETUA - Vol. 2, nº 2, pág. 169. Lisboa, 1998.

UNESP - UNIVERSIDADE ESTADUAL PAULISTA JULIO DE MESQUITA FILHO. Introdução à Plataforma de Desenvolvimento Arduino. Disponível em:
<http://www2.feg.unesp.br/Home/PaginasPessoais/ProfMarceloWendling/apresentacao-arduino.pdf> – Acesso em 16/08/2014.