

## **1 INTRODUÇÃO:**

A automação, de uma forma geral, está diretamente ligada ao controle e à supervisão de processos mecanizados, isto é, que independem da ação direta do ser humano. Desse modo, a automação aplicada na indústria naval está presente em dois níveis: na construção e na operação do navio. Tem como objetivos principais minimizar o esforço humano, aumentar a qualidade, diminuir custos e aumentar a segurança e a comodidade.

Nas últimas décadas, o uso da tecnologia da automação e controle tem aumentado consideravelmente; e com a evolução e a popularização do computador, a manipulação e o armazenamento de dados se tornaram muito mais simples e baratos. Sistemas de aquisição de dados, baseados na tecnologia da informação, estão se popularizando cada vez mais na indústria por permitirem o controle mais eficiente de processos através do recolhimento de informações a uma maior velocidade, com elevados níveis de segurança, conforto e economia para os usuários.

O uso das tecnologias de controle e automação trouxe consigo, no âmbito da indústria naval, um aspecto importante na condução de navios, especialmente com relação à praça de máquinas. Atualmente a maioria dos navios mercantes modernos possui um nível de automação tal que eles são capazes de navegar com a máquina desguarnecida. Isso não quer dizer que um navio que possua máquina desguarnecida não necessite da ação humana. Um navio com máquina desguarnecida apenas introduz a idéia de que os oficiais de máquinas a bordo do mesmo não precisam ficar o tempo todo na praça de máquinas supervisionando os sistemas. Os alarmes são transferidos para o camarote do oficial de quarto e, caso algum alarme acione, o oficial de quarto de serviço visualiza o alarme, dirige-se até o Centro de Controle de Máquinas (CCM) e “reseta” (desliga) o alarme, procurando sanar o problema que causou o disparo do alarme.

Este nível de automação trouxe muitos benefícios para o usuário, tais como: segurança, conforto e economia. E por economia, entende-se um conjunto de ações a bordo que visa manter em níveis mais baixos possíveis os índices de consumo de energia e combustível do navio, mantendo-se a bordo um gerenciamento da eficiência de propulsão e de consumos básicos. Por isso, num navio com máquina desguarnecida, no período em que não há ninguém na praça de máquinas, geralmente à noite, o sistema de gerenciamento de energia ou otimização de carga básica determina que as luzes principais da praça de máquinas e do Centro de Controle de Máquinas permaneçam apagadas.

Mas ao mesmo tempo em que isso gera um benefício na economia, também gera um problema de segurança. Devido ao fato de a praça de máquinas e o CCM estarem somente com as luzes de emergência acesas, alguns locais pontuados ficam na penumbra. Quando acontece de o oficial de quarto dirigir-se à praça de máquinas para desligar o alarme, há a grande possibilidade de acontecer um acidente com o referido tripulante, somente pelo fato de ele se locomover pela praça de máquinas com a mesma na penumbra.

Assim, o objetivo geral deste trabalho é propor uma solução de baixo custo e desenvolver um protótipo que visa automatizar a iluminação da praça de máquinas e do Centro de Controle de Máquinas de um navio com a praça de máquinas desguarnecida. Este procedimento se dará através de um sensor de movimento ligado a um microcontrolador Arduino. Com um sensor de movimento PIR integrado com uma placa Arduino, poder-se-á detectar o movimento de uma pessoa no interior da praça de máquinas e programar o Arduino para ligar as luzes principais da praça de máquinas por um certo período de tempo, uma vez que o movimento ou a presença de alguém seja detectada.

O objetivo específico é mostrar as vantagens da utilização da automação desse protótipo para alavancar a economia, a comodidade e, principalmente, a segurança, e descrever o funcionamento da solução proposta. Destacam-se ainda como objetivos específicos, a implementação no Arduino, de forma correta, das lâmpadas que representam as luzes da praça de máquinas, ligando-as nas saídas analógicas e digitais do microcontrolador, e a programação do hardware microcontrolador Arduino, ou seja, o desenvolvimento do código de programação na linguagem própria do microcontrolador Arduino.

## 2 SENSORES

Um sensor pode ser definido como um componente eletrônico capaz de converter energia de uma forma em outra forma de energia. Os sensores tem a vantagem de captar várias formas de energia – luz, calor, movimento – e transformá-las em energia elétrica. Pode-se pensar num sensor como sendo um tipo de transdutor, que é um componente eletrônico que converte formas de energia em energia elétrica.

O sinal de saída de um sensor pode ser na forma de tensão, corrente ou carga, e pode ser descrito na forma de amplitude, frequência, fase, ou como um código digital. A este conjunto de características dá-se o nome de “formato do sinal de saída”. Assim, um sensor possui propriedades de entrada e propriedades elétricas de saída.

Os sensores são classificados como sensores ativo e passivo. A diferença entre eles é que um sensor passivo não precisa de fonte de energia adicional, gerando diretamente um sinal elétrico em resposta a um estímulo externo. A energia de entrada (estímulo externo) é convertida diretamente em sinal elétrico de saída. Por outro lado, um sensor ativo é um tipo de sensor que exige uma fonte externa para sua operação, chamado de “sinal de excitação”. Este sinal é utilizado pelo sensor para produzir o sinal de saída.

São vários os tipos de sensores existentes no mercado. Existem os sensores internos que fornecem informações sobre os parâmetros do próprio agente ou ambiente que controlam como por exemplo: posição, velocidade, aceleração, temperatura, pressão; e os sensores externos, que lidam com a observação de aspectos do mundo exterior, ou seja, ao redor do agente ou ambiente que controlam. Por exemplo: proximidade, força, visão, distância.

Um sensor analógico pode assumir qualquer valor no seu sinal de saída ao longo do tempo, desde que esteja dentro da sua área de atuação. As variáveis controladas por esse tipo de sensor são medidas por elementos sensíveis com circuitos eletrônicos não digitais, dando informações em forma de um sinal elétrico proporcional à grandeza medida. O sensor digital só pode assumir apenas dois valores no seu sinal de saída ao longo do tempo, que são interpretados como zero ou um. As grandezas físicas medidas por esse tipo de sensor logicamente não assumem tais valores, porém são assim mostrados ao sistema de controle após serem convertidos por um circuito eletrônico, dando informações em forma de pulsos elétricos.

Deve-se levar em consideração uma série de características relacionadas aos sensores na hora da seleção para uma determinada aplicação. A descrição de tais características não é foco do objeto de estudo deste trabalho, porém pode-se relacionar cada uma dessas

características que são de fundamental importância para a automação no âmbito industrial. São elas: tipos de saída, linearidade, alcance, velocidade de resposta, etc.

Figura 1 – Tipos de Sensores

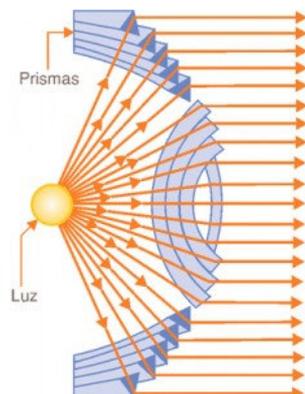


Fonte: <http://1612182.blogspot.com.br/2013/05/21-sensores-y-transductores.html>

Para a aplicação deste trabalho em questão será utilizado o sensor de movimento Passive Infrared Motion Sensor (PIR) que é um detector de movimento por infravermelho. Os sensores infravermelhos são sensores passivos que operam dentro da faixa ótica de radiação térmica e se sensibilizam ao calor irradiado entre o sensor e o objeto em movimento. O princípio da detecção de movimento por calor é baseado na teoria da emissão de radiação eletromagnética de qualquer objeto que tenha temperatura acima do zero absoluto, que corresponde a zero grau Kelvin (0 K).

Nesse sensor existe uma substância que se polariza na presença de radiação infravermelha, gerando assim uma tensão que pode ser amplificada e empregada para efeitos de controle. Desse modo, o calor do corpo de uma pessoa é suficiente para produzir uma emissão infravermelha detectável por esse tipo de sensor. Para aumentar sua sensibilidade e dirigir as ondas de infravermelho diretamente para o sensor, são usadas lentes especiais denominadas Lentes de Fresnel, que têm o padrão exibido na figura 2. ([www.sabereletronica.com.br](http://www.sabereletronica.com.br)).

Figura 2 – Lentes de Fresnel



Fonte: [www.sabereletronica.com.br](http://www.sabereletronica.com.br)

Uma Lente de Fresnel é um tipo de lente que foi inventada pelo físico francês Augustin-Jean Fresnel, no ano de 1822. Originalmente essa lente foi criada para uso em faróis de sinalização marítima. Elas são mais finas e mais leves que as lentes convencionais permitindo a passagem de mais luz e possibilitando um desenho de construção de lentes de grande abertura e curta distância focal, fazendo com que os faróis equipados com essas lentes sejam visíveis a distâncias maiores.

Segundo a Lei de Stefan-Boltzmann todos os objetos emitem radiação térmica. A Lei de Stefan-Boltzmann estabelece que a energia total irradiada por unidade de área superficial de corpo negro na unidade de tempo (J), ou densidade do fluxo energético, ou fluxo radiante, é diretamente proporcional à quarta potência da sua temperatura termodinâmica (T). Ou seja:

$$J = \sigma T^4$$

Onde,  $\sigma$  é a constante de proporcionalidade, também chamada de constante de Stefan-Boltzmann, e possui um valor de  $5,6697 \times 10^{-8} \text{ Wm}^{-2} \text{ K}^{-4}$ .

Josef Stefan deduziu, de forma experimental, no ano de 1879, que a potência P (energia radiada por segundo) de um corpo negro é diretamente proporcional à sua temperatura T elevada à quarta potência e também diretamente proporcional à área A da superfície emissora (Lei de Stefan).

$$P = \sigma T^4 A.$$
$$\sigma = 5,6697 \times 10^{-8} \text{ Wm}^{-2} \text{ K}^{-4}$$

Em 1884 Ludwig Boltzmann deduziu teoricamente essa lei, supondo uma máquina térmica ideal, tendo como substância de trabalho a luz. Assim pode-se derivar esta lei sobre a radiação de um corpo negro:

$$j^* = \int_0^{\infty} \left( \frac{dj^*}{d\lambda} \right) d\lambda$$

Na física, um corpo negro é um corpo que absorve toda a radiação que nele incide: nenhuma luz o atravessa nem é refletida. Corpos negros produzem radiação eletromagnética, tal como a luz. Um corpo negro é um corpo que emite ou absorve radiação eletromagnética em todos os comprimentos de onda, absorvendo completamente toda radiação incidente. Nele,

a máxima radiação possível para a temperatura do corpo é emitida em todos os comprimentos de onda e em todas as direções.

Segundo Glauber Luciano Kitor,

“todo corpo emite e absorve radiação. Quando a temperatura do corpo é maior que a do ambiente onde ele está inserido, a taxa de emissão é maior que a taxa de absorção. Quando a temperatura do corpo é menor que a do ambiente onde ele se encontra, a taxa de emissão é menor que a taxa de absorção. Um corpo só não emite radiação térmica se sua temperatura for zero absoluto, ou seja, 0 K (zero kelvin)”. (www.infoescola.com).

Ainda de acordo com Glauber,

“quando um corpo se encontra à temperatura ambiente, ele é visto pela radiação que ele reflete na faixa de frequência da luz visível. Se estiver a temperaturas altíssimas, em torno de mil kelvins, ele emite luz visível própria em intensidade suficiente para ser detectada pela visão humana”.

“É definido como corpo negro todo aquele que emite um espectro de radiação universal que depende apenas de sua temperatura, não de sua composição. Este tipo de corpo absorve toda a radiação que incide sobre eles. Daí a denominação corpo negro.” (www.infoescola.com).

Assim, para que haja detecção de movimento pelo sensor PIR, é necessário que a temperatura da superfície do objeto ou do corpo a ser detectado seja diferente da temperatura dos objetos ao redor, de forma que possa existir um contraste térmico. A tensão de saída do sensor infravermelho é dada por:

$$V = G(T_b^4 - T_s^4)$$

Onde:  $T_b$  – Temperatura absoluta do objeto ou corpo a ser detectado;  $T_s$  – Temperatura absoluta da face do sensor;  $G$  – Constante.

O elemento sensor PIR é responsável por captar radiação infravermelha de médio e longo alcance dentro de uma distância espectral de aproximadamente 4 a 20  $\mu\text{m}$ , onde grande parte da energia térmica do ser humano está concentrada (para temperaturas superficiais em torno de 28 a 37°C). O material piroelétrico de que é feito gera uma carga elétrica em reação à energia térmica emanada do corpo. Essa carga elétrica se manifesta como tensão elétrica através dos eletrodos montados no lado oposto do material. Portanto, devido às suas características peculiares e por seu baixo custo, o sensor de movimento PIR é o sensor que será utilizado na confecção do protótipo deste trabalho. Ele será usado para captar o

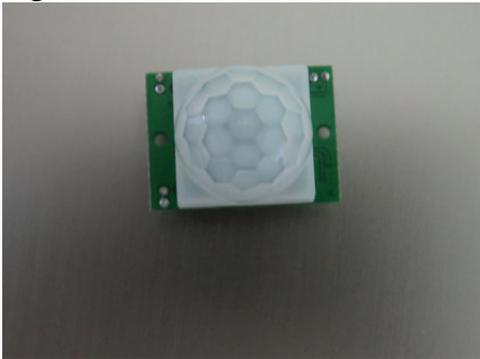
movimento de qualquer pessoa que adentrar no ambiente onde estiver instalado o sensor, de modo a gerar uma tensão elétrica em seus terminais.

Este sensor foi escolhido para esta aplicação pelo fato de ele monitorar uma área de tamanho razoável, além de ser pequeno e leve. Porém, para este trabalho, isso é tudo do que será preciso.

“Os sensores PIR são simplesmente gatilhos. Eles normalmente apenas reportam movimento, não as informações do movimento, o local desse movimento ou qualquer informação acerca da quantidade de tempo que o movimento durou” (Arduino em Ação).

O sensor utilizado no protótipo é fabricado pela Parallax e contém três pinos conectores: alimentação, terra e o pino de sinal. Sua ilustração pode ser visualizada na figura 3.

Figura 3 – Sensor de Movimento PIR



Fonte: do arquivo pessoal

### 3 MICROCONTROLADORES

Em 1969, a BUSICOM, uma empresa japonesa, solicitou à INTEL a criação de um circuito integrado para sua nova linha de calculadoras programáveis. Assim, uma equipe de engenheiros, sob a regência do engenheiro Marcian Hoff, recebeu a missão de criar calculadoras com circuitos integrados. Hoff então propôs uma idéia revolucionária: ele afirmou que seria possível construir uma máquina com um circuito integrado programável, através de programas armazenados em uma memória. Desta idéia, surgiu o primeiro microprocessador da INTEL a ser comercializado: o Intel 4004. Desde então, os microprocessadores só fizeram evoluir até os dias atuais.

Um microcontrolador é um computador miniaturizado capaz de controlar funções ou ações de outros dispositivos. É um microprocessador que pode ser programado para realizar funções específicas, embarcado em algum outro dispositivo. Isto quer dizer que os microcontroladores são dispositivos que possuem seus recursos computacionais projetados exclusivamente para controlar um outro dispositivo com um propósito especial, interagindo continuamente com o ambiente através de sensores e atuadores, gerenciado por um software programável. Um microcontrolador é um “computador-num-chip, contendo um processador, memória e periféricos de entrada/saída”. ([www.pt.wikipedia.org](http://www.pt.wikipedia.org)).

A aplicabilidade dos microcontroladores está intrinsecamente ligada à sua programação, sem a qual é impossível desenvolver projetos ou sistemas capazes de executar tarefas para uma determinada aplicação ou de exercer função de controle. Um microcontrolador sem uma programação é como um cego sem bengala; um carro sem motor. E a programação é dependente de uma linguagem em código, linguagem esta que servirá de elo de comunicação entre a máquina e os projetistas ou o programador.

A linguagem de programação é o caminho através do qual o sistema interpreta e executa uma lista de comandos que são escritos pelo programador. Estes comandos são um conjunto de palavras e símbolos, ou melhor, um conjunto de instruções que mudam o estado de uma máquina ou um sistema. Isto equivale dizer que uma máquina computadorizada só entende instruções exatas do que deverá ser executado e que deverá mudar o seu estado de ligado para desligado ou vice-versa.

Gabriel Andrade afirma que

“existem dois tipos de linguagens de programação: as de baixo nível e as de alto nível. Os computadores interpretam tudo como números em base binária, ou seja, só entendem zero e um. As linguagens de baixo nível são interpretadas diretamente pelo computador, tendo um resultado rápido,

porém é muito difícil e incômodo se trabalhar com elas. Exemplos de linguagens de baixo nível são a linguagem binária e a linguagem Assembly”. (www.infoescola.com).

Ainda de acordo com Andrade,

“as linguagens de alto nível são mais fáceis de se trabalhar e de entender, as ações são representadas por palavras de ordem (faça, imprima), geralmente em inglês; foram feitos assim para facilitar a memorização e a lógica. Elas não são interpretadas diretamente pelo computador, sendo necessário traduzí-las para linguagem binária utilizando-se de um programa chamado compilador” (www.infoescola.com).

A linguagem de programação C é um exemplo de linguagem de alto nível que pode ser utilizada na programação de microcontroladores, sendo considerada uma linguagem bastante poderosa e versátil, celebrada por ter um equilíbrio entre estruturação e eficiência. A única desvantagem da linguagem C está no fato de ela consumir muita memória. Fábio Pereira afirma que a maioria dos microcontroladores disponíveis no mercado atualmente conta com compiladores de linguagem C para o desenvolvimento de software.

### **3.1 Microcontrolador Arduino**

Em 2005, na cidade de Ivrea, a cidade da “batalha das laranjas”, na Itália, um grupo de alunos da Interaction Design Institute costumava sair para beber em um bar local. Estes alunos eram estudantes de arte deste Instituto cujo professor, Massimo Banzi, procurava uma solução que fosse barata para tornar mais fácil o aprendizado de seus estudantes de design para trabalharem com tecnologia. O professor Banzi discutiu, então, o problema com David Cuartielles, um pesquisador da Universidade de Malmö, na Suécia, que estava de visita no Instituto de Arte. Na época, os produtos existentes no mercado eram caros, difíceis de manusear e, portanto, inviáveis para o intento deles. Assim, Banzi e Cuartielles entraram num consenso e decidiram desenvolver um microcontrolador que pudesse ser utilizado pelos estudantes em seus projetos de arte e design.

David Cuartielles fez o desenho da placa e um aluno de Massimo, David Mellis, programou o software para executar a placa. Havia um engenheiro que trabalhara no Instituto, Gianluca Martino, que, contratado por Massimo Banzi, produziu um lote inicial de duzentas placas. Ora, todo objeto que nasce e que é projetado deve ser batizado com um nome. Como os alunos de Banzi e os membros do corpo docente do Instituto gostavam muito do bar que eles costumavam frequentar, eles resolveram batizar a nova placa com o nome deste referido bar: Caffeteria Arduino. Assim nasceu o microcontrolador Arduino.

As duzentas placas iniciais foram rapidamente vendidas e mais placas foram produzidas. Sua popularidade rapidamente se expandiu, e designers e artistas de outras áreas resolveram experimentar e utilizar o Arduino também em seus projetos. Outros profissionais de outros ramos perceberam que o Arduino era de fácil utilização e barato, e decidiram utilizá-lo também em seus projetos. As vendas atingiram um número tão expressivo que alcançaram mercados de outros países. O projeto original foi melhorado e novas versões foram produzidas. Atualmente existem várias versões de placas Arduino.

Mas afinal, o que é exatamente um Arduino? Segundo a Wikipedia,

“Aduino é um microcontrolador de placa única e um conjunto de software para programá-lo. O hardware consiste em um projeto simples de hardware livre para o controlador, com um processador Atmel AVR e suporte embutido de entrada/saída. O software consiste de uma linguagem padrão e do bootloader que roda na placa”. ([www.pt.wikipedia.org](http://www.pt.wikipedia.org).)

O conceito é relativamente novo. O Arduino é uma plataforma de computação física ou embarcada, um pequeno computador que possui um software programável para processar entradas e saídas entre o microcontrolador e os componentes externos conectados a ele, ou seja, o sistema interage com o ambiente por meio do hardware e do software.

A placa Arduino é composta de um microprocessador de 8 bits Atmel AVR, um cristal ou oscilador e um regulador linear de 5V. Além disso, há um botão de reset e uma série de outras funcionalidades instaladas em seu bojo, cuja explicação foge do contexto deste tópico. As placas possuem pinos de entrada e saída a fim de que se possa conectá-los a outros circuitos e dispositivos. O hardware e o software do Arduino são de fonte aberta, podendo ser utilizado livremente por qualquer pessoa ou empresa. Por isso, há muitas placas- clones baseadas no Arduino disponíveis no mercado.

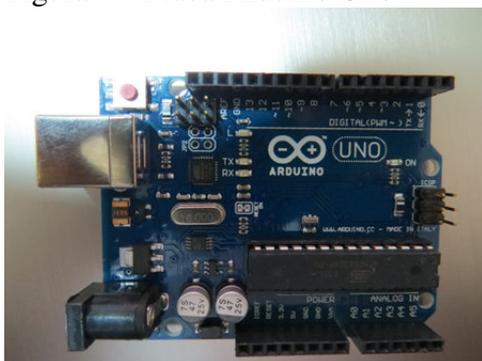
Um oscilador de cristal é um circuito eletrônico que utiliza a ressonância de um cristal em vibração de um material piezoelétrico, para criar um sinal elétrico com uma frequência bastante precisa. Esta frequência é comumente usada para medir precisamente o tempo, tais como em relógios de quartzo, bem como para estabilizar frequências de transmissores de rádio. O cristal piezoelétrico mais utilizado é o quartzo.

Um regulador de tensão é um dispositivo, geralmente formado por semicondutores, que tem por finalidade a manutenção da tensão de saída de um circuito elétrico. Sua função principal é manter a tensão produzida pelo gerador ou alternador dentro dos limites exigidos pela bateria ou pelo sistema elétrico que está alimentando e para tanto, é necessário que a

tensão de entrada seja superior à tensão de saída. O regulador de tensão mantém a tensão de saída constante, estabilizada, mesmo havendo variações na tensão de entrada ou na corrente de saída.

Até o momento da confecção deste trabalho existe uma série de versões do Arduino em comercialização. O Arduino Uno é a mais versátil e a mais popular dentre as versões existentes. Esta versão do Arduino anunciou sua chegada ao mercado no dia 25 de setembro de 2010. Possui um chip padrão de 28 pinos ligado a um soquete de circuito integrado compatível com as versões anteriores, incluindo a Duemilanove, a versão anterior mais popular de Arduinos. Trabalha com memória de 32kb, tensão de operação de 5 V, corrente máxima para as portas de entrada/saída de 40 mA, 14 portas digitais, 6 portas PWM, 6 portas analógicas, alimentação por USB e conector para fonte externa de 7 a 12 Vdc. A principal diferença entre o Arduino Uno e seus predecessores é a exclusão do chip FTDI, que servia como condutor da porta USB para serial, e a inclusão de um microcontrolador programado ATmega8U2 como um conversor USB para serial. Quando comparada à sua predecessora Duemilanove, em termos de custos, a placa Uno sai mais em conta pois o chip Atmega é muito mais barato que o chip FTDI. Mais ainda, o chip Atmega pode ser reprogramado, permitindo que o chip USB tenha seu firmware atualizado, de modo que o Arduino possa ser visualizado pelo PC como outro dispositivo USB, tal como mouse, teclado, joystick, etc.

Figura 4 – Placa Arduino Uno

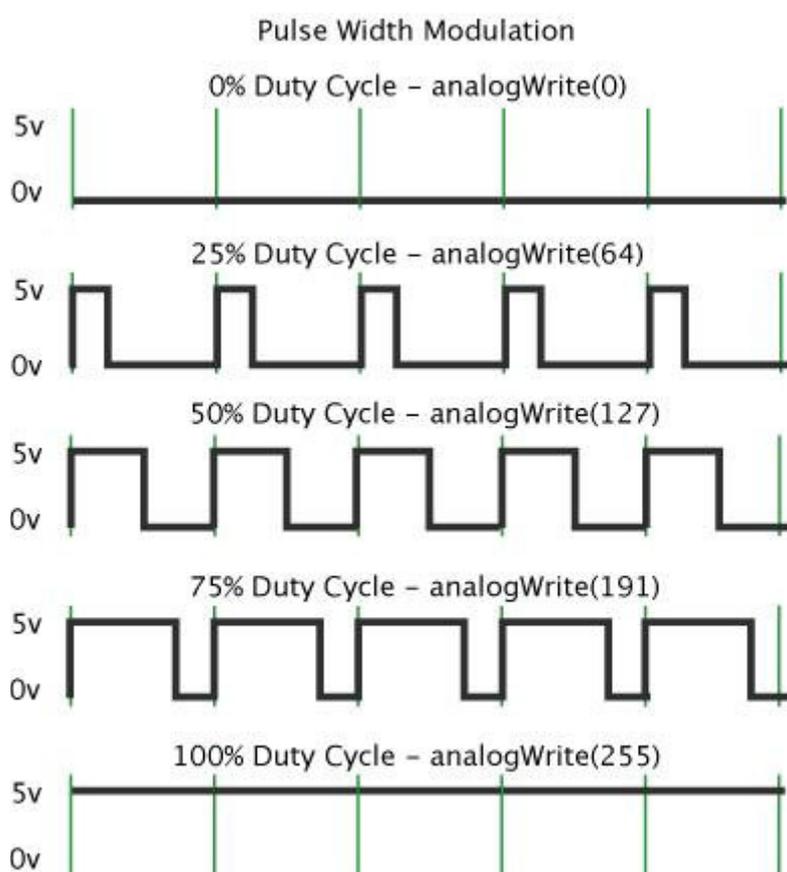


Fonte: do arquivo pessoal

A porta Pulse Width Modulation (PWM), ou modulação de largura de pulso, é uma técnica que fornece um sinal analógico através de meios digitais. Num gráfico tensão x tempo, a forma de onda do sinal digital expressa-se sob a forma de uma onda quadrada que alterna seu estado de nível lógico em nível alto e nível baixo, representado pelo sistema binário ligado/desligado, zero/um. A razão entre o período de pico e o período total da onda é chamado de Duty Cycle. Pode-se, então, entender que para que exista uma onda quadrada real

(que possui picos e vales iguais) é necessário que o Duty Cycle seja de 50%, ou seja, 50% de pico e 50% de vale. O controle digital cria ondas quadradas com pulsos de largura variável, podendo assim dar uma tensão média à carga. No Arduino Uno, as portas digitais que possuem modulação de largura de pulso, PWM, são as portas numeradas 3, 5, 6, 9, 10 e 11. Essas portas são identificadas na placa pelo símbolo “~” na direção de cada uma das portas. Como foi citado, a tensão de saída de uma porta digital da placa Arduino Uno é de 5V. Assim, esse mesmo valor de tensão pode ser obtido da porta digital ao enviá-lo a uma saída analógica de valor 255, ou seja, duty cycle de 100%, conforme mostrado na figura 5.

Figura 5 – Gráfico tensão x tempo mostrando a saída da porta PWM



Fonte: <http://arduino.cc/en/Tutorial/PWM>

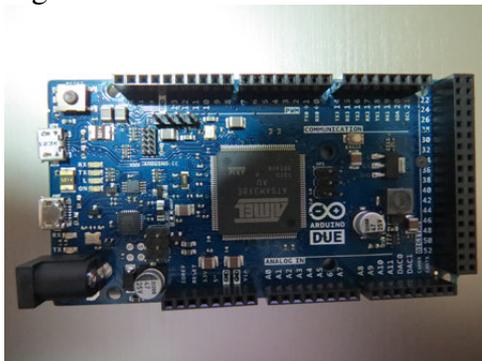
Além das versões já citadas, ainda existem outras versões de placas Arduino. O Arduino Mega2560 utiliza o microprocessador ATmega2560, tem 54 portas digitais, 15 portas PWM, 16 portas analógicas, 256Kb de memória, tensão de operação de 5V, corrente

máxima das portas de entrada/saída de 40 mA e alimentação por USB, contendo conector para alimentação externa de 7 a 12Vdc.

O Arduino Leonardo é uma versão do Arduino que emprega o microprocessador ATmega32U4, usufrui de 20 portas digitais, 7 portas PWM, 12 portas analógicas, 32 Kb de memória, tensão de operação de 5V, corrente máxima para as portas de entrada/saída de 40 mA, conexão microUSB e conexão para alimentação externa de 7 a 12 Vdc.

O Arduino DUO dispõe do microprocessador AT91SAM3X8E, contendo 54 portas digitais, 12 portas PWM, 12 portas analógicas, 512Kb de memória, tensão de operação de 3.3V, corrente máxima das portas de entrada/saída de 130 mA, possuindo conexão microUSB e conexão para alimentação externa de 7 a 12 Vdc.

Figura 6 – Placa Arduino Duo



Fonte: do arquivo pessoal

Arduino ADK tem em sua placa o microprocessador ATmega2560, 54 portas digitais, 15 portas PWM, 16 portas analógicas, memória de 256Kb, tensão de operação de 5V, portas de entrada/saída com corrente máxima de 40 mA, conexão USB e conexão para alimentação externa de 7 a 12 Vdc.

O Arduino Nano possui duas versões: a versão 2.x utiliza o microprocessador ATmega168 e a versão 3.x usa o microprocessador ATmega328; tem 14 portas digitais, 6 portas PWM, 8 portas analógicas, memória de 16Kb (para a versão 2.x) e de 32Kb (na versão 3.x), tensão de operação de 5V, corrente máxima das portas de entrada/saída 40 mA; não possui conector para alimentação externa, mas tem conexão USB Mini-B; alimentação externa de 7 a 12 Vdc.

Arduino Pro Mini: Microprocessador ATmega168, 14 portas digitais, 6 portas PWM, 8 portas analógicas, 16Kb de memória, tensão de operação de 3.3V ou 5V, dependendo do modelo; também não possui conector para alimentação externa; utiliza conexão serial com

módulo USB externo; alimentação externa de 3.3 a 12Vdc para o modelo de tensão de operação de 3.3V, e de 5 a 12 Vdc para o modelo com tensão de operação de 5V.

Como já citado, o Arduino é um hardware de código aberto (open-source). Isto levou à geração de um número de cópias de Arduino com vários fabricantes clones usando a especificação original. O nome Arduino é o único item que não pode ser copiado por ser uma marca registrada. Mas isso não impede que exista um certo número de clones do Arduino. São eles: Seeduino, Seeduino Film, BoArduino, Sippino, Freeduino, Illuminato, Metaboard, Wiseduino, Brasuino, entre outros.

Para a programação do Arduino utiliza-se o Integrated Development Environment (IDE), ambiente de desenvolvimento integrado, no qual é escrito o código na linguagem compreendida pelo Arduino. O IDE é o software que fornece as ferramentas necessárias para programar o Arduino, incluindo exemplos de programas ou sketches que mostram como conectá-lo e comunicar-se com alguns dispositivos e sensores.

O Arduino usa sua própria linguagem de programação que é basicamente um dialeto da linguagem C. Ela utiliza um conjunto de funções das linguagens C e C++ em seu código de programação. É possível usar um ambiente diferente para programar o Arduino, bem como também é possível programar uma placa Arduino sem utilizar o software próprio do Arduino.

Shields são placas de circuito contendo outros dispositivos que são plugadas no topo da placa Arduino estendendo suas funcionalidades. Existem diferentes tipos de shields. Os shields que adicionam novas funcionalidades à placa Arduino exigem bibliotecas especializadas, normalmente fornecidas e mantidas pelos desenvolvedores dos shields.

Em referência às bibliotecas, Martin Evans destaca que

“as bibliotecas ajudam a linguagem a ocultar muito de sua complexidade e simplificar muitas tarefas comuns, tais como definir os pinos digitais como entrada ou saída, ler valores analógicos, controlar servomotores ou ligar e desligar motores DC. Muitas das bibliotecas essenciais utilizadas são baseadas em uma biblioteca chamada Wiring, que foi desenvolvida por Hernando Barragan”. (Arduino em Ação).

Assim, o Arduino, como um projeto que aglomera software e hardware para uma plataforma de fácil interação para prototipagem de projetos iterativos, utilizando um microprocessador, é uma ferramenta bastante apropriada, de código aberto e de fácil utilização, capaz de processar sinais analógicos e digitais, tornando possível a integração com dispositivos eletrônicos, através de suas entradas e saídas e interagindo direta ou indiretamente com o ambiente.

O Arduino possui fonte de alimentação externa AC/DC ou pode ser alimentado pela própria conexão USB. Qualquer uma delas é automaticamente selecionada pelo Arduino quando em uso. E ainda existe uma proteção extra, um polifusível, que evita que a porta USB do PC seja danificada por um curto-circuito ou sobrecorrente.

Para o escopo desta aplicação, a versão escolhida foi o Arduino Uno, cujos dados técnicos já foram expressamente detalhados. Baseou-se essa escolha no fato de ser o Arduino Uno um circuito integrado simples, barato e de fácil manuseio para este protótipo, além de ser compacto e ser compatível com vários outros dispositivos, em especial, com o sensor de movimento PIR.

## 4 MODELO PROPOSTO

O sistema proposto para a arquitetura do protótipo inicia-se com o estudo da viabilidade da aplicação da placa Arduino para o processamento e análise de sinais em bancada eletrônica, simulando um sistema de controle e monitoramento direcionado para o cenário de automação da iluminação de uma praça de máquinas desguarnecida de um navio mercante, porém com a possibilidade de aplicação para outros objetivos.

### 4.1 Análise de hardware

Inicialmente o protótipo seria composto por um microcontrolador Arduino Duo, porém esta placa não foi implementada por ter uma tensão de operação de 3.3 V, deixando o sistema instável. Preferiu-se, então, a implementação do microcontrolador Arduino Uno que possui uma tensão de operação de 5 V. Qualquer outra versão do Arduino que tivesse este valor de tensão poderia ser utilizado, porém, com a premissa de que todos os dados técnicos referentes à placa escolhida devem ser previamente analisados.

O microcontrolador ATmega328 do Arduino Uno permite a comunicação serial no padrão Universal Asynchronous Receiver Transmitter (UART) nos pinos 0 RX e 1 TX, isto é, pega bytes de dados e os transmite em bits individuais sequencialmente, trabalhando com níveis de tensão Transistor to Transistor Logic (TTL), ou seja, 5 V. TTL é uma classe de circuitos digitais construídos de transistores de junção bipolar e resistores. É chamado lógica transistor-transistor por possuir ambas as funções de porta lógica e de amplificação pelos transistores. A principal característica deste circuito é que ele utiliza sinais de 5 V para níveis lógicos altos.

O Arduino Uno utiliza o Atmega8U2 como um chip programado para converter a porta USB em porta serial, encaminhando essa comunicação de dados seriais entre o Arduino e o computador. Quando os dados estão sendo transferidos, com a porta USB e a placa Arduino conectadas, os LEDs RX e TX na placa piscam, mas isto não ocorre quando se utiliza a comunicação através dos pinos 0 e 1 da placa.

Qualquer um dos 14 pinos digitais da placa Arduino Uno pode ser utilizado como entrada ou saída, podendo cada um deles receber ou fornecer uma corrente máxima de 40 mA. Cada pino ainda possui um resistor pull-up interno de 20 a 50 K $\Omega$ , que é desconectado por padrão. Há um LED de indicação do pino 13 que se acende quando este pino está com valor high (ligado) e se apaga quando o pino tem valor low (desligado).

A entrada USB pode ser usada não só como entrada de comunicação, mas também para fornecer alimentação à placa Arduino Uno. O conector VIN é a entrada de alimentação da placa quando se deseja utilizar uma fonte de alimentação externa ao invés da entrada USB. O pino 5V da placa Arduino Uno é uma fonte de alimentação regulada usada tanto para o microcontrolador quanto para outros componentes conectados na placa. O pino 3V3 é uma fonte de 3.3 V gerada pelo regulador embarcado e a corrente máxima suportada para este pino é de 50 mA. Os pinos GND são pinos terra. Os pinos 2 e 3 são interruptores da placa Arduino Uno que podem ser configurados para disparar uma interrupção de acordo com alguma variação sensível pelo circuito. AREF é a voltagem de referência para as entradas analógicas da placa e RESET envia o valor low para a placa a fim de resetar o microcontrolador. O pino RESET também é utilizado para adicionar um botão de reset caso se deseje montar shields sobre a placa original. O reset do Arduino Uno também pode ser feito via software, através de uma linha de comando do programa que esteja rodando para o funcionamento de algum aplicativo que o Arduino esteja controlando. Isto só é possível porque uma das linhas de controle de hardware do chip está conectada ao reset do ATmega328 através de um capacitor de 100 nF (cem nanofarads). Quando esta linha é resetada, isto é, ativo baixo, o sinal cai por tempo suficiente para resetar o chip.

Nenhum driver externo é necessário para instalação da placa Arduino no computador. Entretanto, para o sistema operacional Windows, é necessário instalar um arquivo .inf (arquivo necessário para setar parâmetros e armazenar informações) para o correto funcionamento da placa. Este arquivo é instalado sem dificuldades e sem ônus. É possível a instalação também em outros sistemas operacionais. Para o escopo deste trabalho a plataforma utilizada foi o sistema operacional Windows. O software do Arduino inclui um monitor serial que permite que dados textuais sejam enviados e recebidos pela placa.

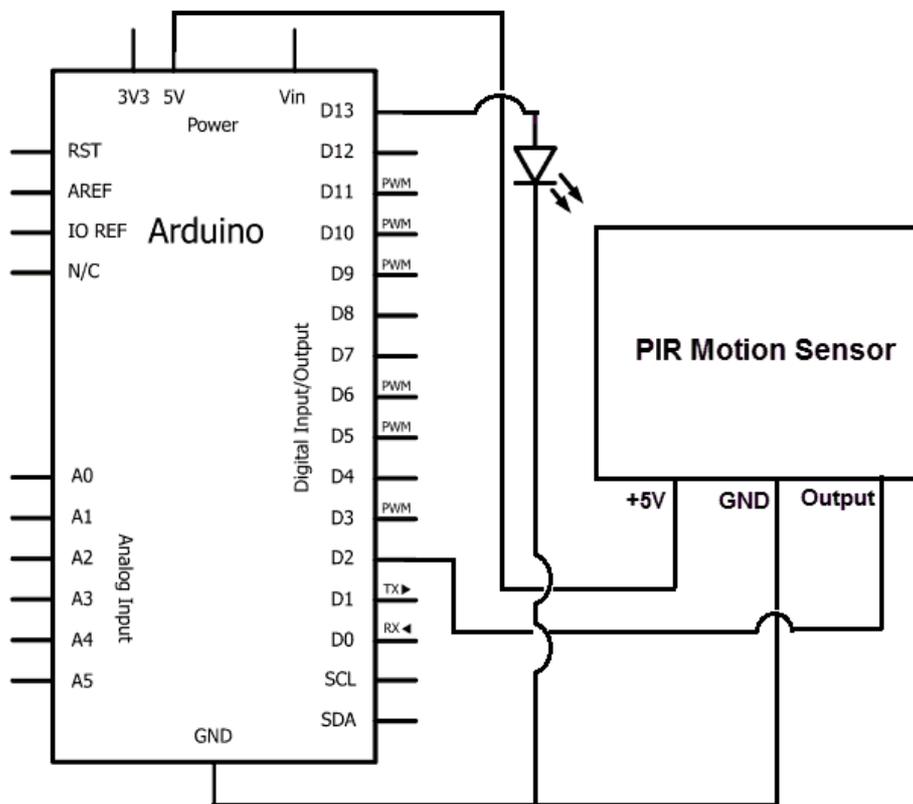
A função do microcontrolador Arduino neste protótipo consiste em interpretar as funções recebidas via processing e acionar a saída digital correta, permitindo ligar ou desligar o led correspondente que representa as luzes da praça de máquinas e do centro de controle de máquinas do navio.

O sensor de movimento PIR utilizado neste trabalho é um módulo altamente integrado, popular, barato, simples e fácil de operar, usado para detecção de entrada ou presença. Possui dois potenciômetros ajustáveis no módulo que podem ser usados para alterar a sensibilidade e a duração do sinal de disparo. Possui temperatura de trabalho de -20 a 80°C. Tem tensão de entrada de 4 a 20 V e corrente de trabalho de 50 µA. Quando o sensor está ocioso, possui sinal de saída de 0,3 V, e de 5 V quando detecta movimento. Possui um ângulo

de visão de 110 graus e distância máxima de 7 metros, suficiente para o propósito deste ensaio.

O sensor de movimento PIR tem dois slots de sensores sensíveis ao infravermelho conectados a um amplificador diferencial para que os sinais indesejáveis sejam anulados e apenas entradas válidas sejam detectadas. Assim, quando o sensor PIR estiver ocioso, ambos os slots detectam a mesma quantidade de infravermelho, ou seja, o sensor PIR fica calibrando com a quantidade de luz infravermelha irradiada a partir do ambiente em que ele está instalado (paredes ou ao ar livre). Quando um corpo quente como um ser humano se locomove na frente do seu raio de visão, ele intercepta essa alteração de calor (luz infravermelha) no primeiro slot do sensor PIR, que faz com que seja registrado um pulso positivo diferencial entre os dois slots. Quando o corpo quente sai da área de detecção, acontece o inverso, o segundo slot do sensor gera um pulso diferencial negativo. Estas alterações de pulsos detectadas promovem a atuação do sensor de movimento PIR. Pela figura 7, pode-se visualizar o diagrama esquemático para o circuito deste estudo.

Figura 7 – Diagrama esquemático do circuito com Arduino e sensor PIR



Fonte: [www.makezine.com](http://www.makezine.com)

O ideal seria que este circuito fosse montado em uma placa de circuito impresso, porém, para fins de experimento, foi utilizado uma protoboard básica. Devido à simplicidade deste circuito, também poder-se-ia fazer a ligação direta dos componentes, nada impedindo que, futuramente, este mesmo circuito fosse atualizado, desenvolvido e expandido. A ligação dos componentes foi realizada do seguinte modo: conexão terra do sensor no pino GND da placa Arduino Uno; conexão de alimentação de entrada 5V do sensor no pino 5V da placa; saída do sensor no pino digital 2 da placa; anodo do diodo (led) no pino digital 13 do Arduino; catodo do diodo (led) no terra GND. Esta é a ligação mais básica, suficiente para iniciar a experiência.

## 4.2 Análise de software

O Arduino Uno pode ser programado com o software Arduino. O microcontrolador ATmega328 já vem pré-gravado com um bootloader em sua memória. O bootloader, ou gerenciador de boot, é um programa simples que tem a função de acessar e carregar o sistema operacional na memória para assumir o controle do equipamento. O bootloader do Arduino permite enviar novos programas sem o uso de um programador de hardware externo. A comunicação é feita utilizando-se o protocolo original do Arduino, o STK500. Também há a possibilidade de se programar o ATmega328 através do In-Circuit Serial Programming (ICSP) header, que é uma forma de se programar o Arduino diretamente no circuito.

Neste projeto, a programação será feita via software, no ambiente de desenvolvimento do Arduino, que utiliza, como já foi citado, uma vertente das linguagens C e C++. A placa Arduino tem como função principal coletar os dados analógicos e digitais, e convertê-los em sinais digitais que serão enviados para o computador via porta USB. O código instalado e executado corretamente efetuará a leitura dos dados nas entradas, iniciará a conexão serial com o computador e executará cada linha de programação interpretada pelo processing e modificará ou não o modo de saída dos pinos em que estão conectadas as lâmpadas em que se deseja atuar.

O código de programação, também chamado de sketch, consiste em duas funções básicas: *setup* e *loop*. A função *setup* é executada somente uma vez no início do programa. Esta função é responsável por iniciar e definir a taxa de comunicação serial entre o computador e a porta USB. É também nesta função que serão emitidas instruções gerais para preparar o programa antes que o loop principal seja executado. A função *loop* será executada num processo cíclico durante a atividade do programa, fazendo leituras periódicas dos pinos

da placa Arduino, realizando chamadas de funções, fazendo operações matemáticas e lógicas, entre outras.

Para o intento deste projeto, o código de programação deverá ser escrito de forma que o Arduino Uno reconheça os sinais do sensor de movimento PIR instalado no ambiente. Para isso, é necessário primeiramente, na função setup, definir os modos dos pinos do Arduino Uno que serão usados na conexão com o sensor e também dos pinos que controlarão as luzes da praça de máquinas. O código de programação escrito na função loop executará os sinais analógicos e digitais para controlar o processo.

Figura 8 – Código de programação

```
////////////////////////////////////
// INICIALIZAÇÃO DAS VARIÁVEIS //
////////////////////////////////////
int pinoSensor = 2;
int pinoLed = 13;

//////////
// SETUP //
//////////
void setup()
{ Serial.begin(9600);

  pinMode(pinoLed,OUTPUT);

  pinMode(pinoSensor,INPUT); }

//////////
// LOOP //
//////////
void loop()
{
  if (digitalRead (pinoSensor) == HIGH)
  {
    digitalWrite (pinoLed, HIGH);
    Serial.println("Movimento detectado pelo Sensor PIR");
  }
  if (digitalRead (pinoSensor) == LOW)
  {
    digitalWrite (pinoLed, LOW);
    Serial.println("Nenhum movimento detectado pelo Sensor PIR");
  }
  delay(1000); }
}
```

Fonte: do arquivo pessoal

A figura 8 apresenta na íntegra todo o código da programação deste protótipo. As linhas onde estão escritos os símbolos “//” não são executadas. São apenas comentários relativos ao programa que o mesmo ignora como instrução. Porém, é possível notar claramente as funções setup e loop. Pela descrição do código, logo no início, no cabeçalho, declara-se as variáveis que serão utilizadas durante toda a execução do programa. Pode-se dar qualquer nome à variável, mas, para melhor visualização e compreensão do código, normalmente pretende-se que o nome da variável corresponda ao elemento ao qual ela está relacionada. No início do programa, nas duas primeiras linhas, são declaradas as variáveis como `int nome da variável`, um sinal de igual e um número inteiro. Isto significa, pela análise do código, que a variável referente ao pino digital D13, onde será conectado o led, receberá o número inteiro 13, e a variável referente ao pino digital D2, onde será efetuada a leitura do sensor, receberá o número inteiro 2.

Na função setup inicializa-se a velocidade da comunicação via porta serial, setando-a para 9600 bps (bits por segundo). Em seguida, nas duas últimas linhas da função, são configurados os pinos 13 como saída digital (output) e 2, que irá ler o sensor, como entrada digital (input).

Na função loop são efetuados dois testes de verificação: primeiro, quando o nível na saída digital do sensor for alto (movimento detectado), o Arduino Uno irá ligar o led e imprimir na tela do serial monitor uma mensagem informando que houve movimento; depois, o segundo teste irá verificar que, quando o nível na saída digital do sensor for baixo (sensor ocioso), o Arduino Uno irá desligar o led e imprimir na tela do serial monitor uma mensagem informando que não houve movimento.

A última linha de comando manda o Arduino Uno aguardar por 1000 ms (mil milissegundos), isto é, por 1 segundo, antes de recomeçar a leitura das linhas de código do programa da função loop.

## 5 REALIZAÇÃO DOS TESTES

Após ter sido feita a ligação dos componentes e o projeto das linhas de programação do Arduino Uno, efetuou-se o teste. Como foi mencionado anteriormente, primeiro foi feito o teste com o Arduino Duo. Observou-se que o sistema teve um comportamento instável. Foi efetuada, então, a troca do Arduino Duo pelo Arduino Uno, tornando assim, o sistema perfeitamente estável e operante. Verificou-se o funcionamento do protótipo com os componentes sendo ligados através da protoboard e depois fazendo-se a ligação direta.

Após o carregamento do programa, efetuou-se na íntegra os testes relativos ao sistema. O sistema ficou operacional, em observação, durante um período de tempo de aproximadamente 12 horas. Sempre que o sensor de movimento PIR detectava a presença de uma pessoa no ambiente, o led acendia. E quando o sensor ficava ocioso, isto é, quando o sensor não detectava movimento algum, o led se apagava. Assim, observou-se a correta calibração e comunicação do sensor de movimento PIR e o correto funcionamento dos comandos do programa do Arduino Uno. Para corroborar a veracidade da funcionalidade do sistema, pode-se verificar as mensagens enviadas para o serial monitor disponível no software de compilação do Arduino.

Os testes elaborados nesta configuração apresentaram bons resultados. Os comportamentos dos sinais analógicos gerados pelo sensor de movimento PIR interagiram convenientemente com o Arduino Uno. Os testes foram realizados com leds simulando as luzes da praça de máquinas do navio. Entretanto, estes leds, conectados diretamente na placa Arduino Uno, são alimentados com uma tensão máxima de 12 Vdc. Deve-se ter em mente que a tensão de trabalho tanto das luzes da praça de máquinas quanto do centro de controle de máquinas é de 110 ou 220 Vac. Portanto, algum arranjo de equipamentos conversores de sinais digitais e analógicos comercializados a baixo custo e de alta simplicidade teve que ser efetuado no circuito para o ajuste do sistema.

Maiores dificuldades não tiveram destaque durante estes testes. Até mesmo a escolha de um equipamento que resolvesse o problema de ajuste do sistema não foi de todo complexa. Após várias considerações, resolveu-se utilizar relé que não onerasse muito o sistema proposto neste estudo. Atualmente existem basicamente dois tipos de relés: os relés eletromecânicos e os relés de estado sólido.

Os relés eletromecânicos são formados por uma bobina onde, em torno dela, é gerado um campo eletromagnético quando o relé é energizado, e um conjunto de contatos que controla o circuito externo. Este campo eletromagnético gera uma força capaz de movimentar

a armadura fixa, que é um conjunto mecânico com contatos móveis, alterando seu estado: se o relé é normalmente aberto, muda seu estado para fechado; se o relé é normalmente fechado, muda seu estado para aberto. Eles são vedados pelo invólucro que é a carcaça do relé e que serve também de proteção para todo o seu conjunto. São também capazes de controlar sinais DC através de tensão AC e vice-versa. O relé eletromecânico tem a vantagem de isolar completamente o circuito de controle do circuito controlado, conduzir corrente nos dois sentidos, além de ter baixo custo e de realizar funções de comutação. A principal desvantagem está na velocidade de operação, que é lenta; ruídos e arcos voltaicos são produzidos ao se acionar os contatos.

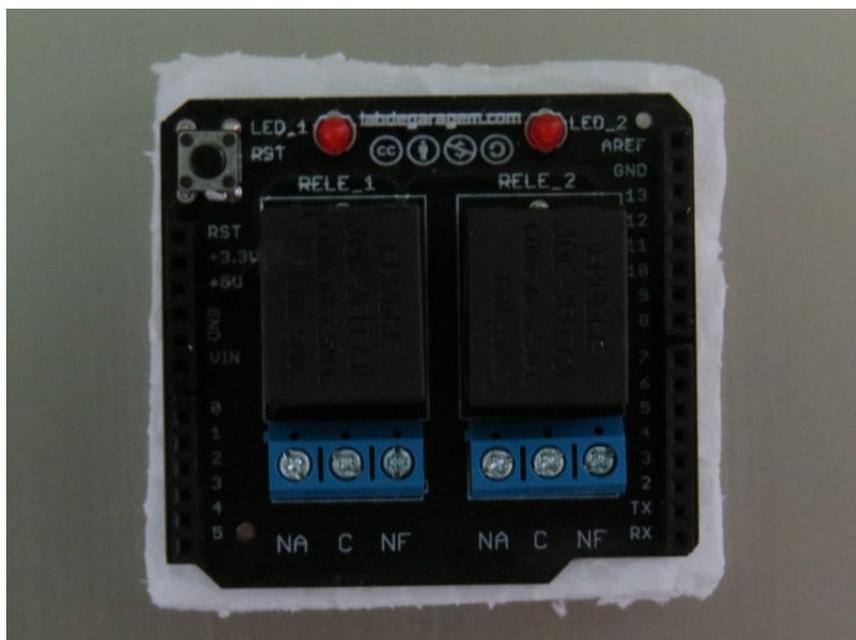
O relé de estado sólido, ou Solid State Relay (SSR), é um dispositivo semicondutor capaz de desempenhar as mesmas funções dos relés eletromecânicos, porém com um sistema de funcionamento completamente diferente. Ele não possui partes mecânicas. É um dispositivo totalmente eletrônico que, para desenvolver suas funções de isolamento e comutação, usufrui das propriedades elétricas, eletrônicas, óticas dos materiais semicondutores e dos componentes elétricos que o compõem. É capaz de comutar circuitos de potências elevadas a partir de sinais de pequenas intensidades. Este tipo de relé é também chamado de Photo MOS, formado basicamente por um LED emissor de infravermelho e um sistema foto-sensor que controla um ou mais transistores de efeito de campo de potência. A principal vantagem é que ele não possui partes móveis, sendo, portanto, silencioso e mais confiável. A desvantagem é que os dispositivos semicondutores utilizados no controle nem sempre apresentam uma resistência suficientemente baixa para o sinal, como seria o desejado. Além disso, o relé SSR possui um circuito de saída mais sensível que os relés eletromecânicos convencionais, maior aquecimento, maior custo e maior capacitância no circuito de saída.

Como já foi mencionado, shields são uma maneira de expandir o Arduino, adicionando funcionalidades à placa. Existem várias classes de shields. Repetindo: os shields do Arduino são placas de hardware plugáveis que se conectam a uma placa Arduino, estendendo suas funções. Muitos desses shields podem ser adquiridos comercialmente, montados e testados, ou podem ser projetados pelo próprio usuário.

Relé shield é uma placa bastante versátil para o Arduino, pois permite acionar cargas em outras tensões de operação. Funciona como um interruptor eletrônico, onde pode-se ligar e desligar a carga pelo Arduino. Por estas características, o relé shield foi o relé escolhido como o elemento possível de resolver o ajuste do sistema automático de iluminação da praça de máquinas do navio mercante desenvolvido por este estudo.

O relé shield escolhido para este experimento é um módulo construído com dois relés miniaturas de potência com um contato reversível, com capacidade de acionar cargas de até 7A/220V. Foi concebido para se integrar ao Arduino sem a necessidade de conexões com jumpers, somente sendo necessário conectar ou encaixar o shield direto no Arduino e ligar os equipamentos a serem controlados nos conectores instalados no shield. O relé shield é controlado diretamente pelo Arduino através de dois pinos I/O (input/output), pinos digitais 7 e 8, e há em seu circuito dois LEDs que indicam o estado dos relés, porém, os pinos digitais 7 e 8 não poderão ser utilizados por outro periférico. A alimentação do shield é feita através de 5Vdc oriundos da placa Arduino. Este relé shield tem baixo custo e é compatível com o Arduino Uno e o Duemilanove.

Figura 9 – Relé shield

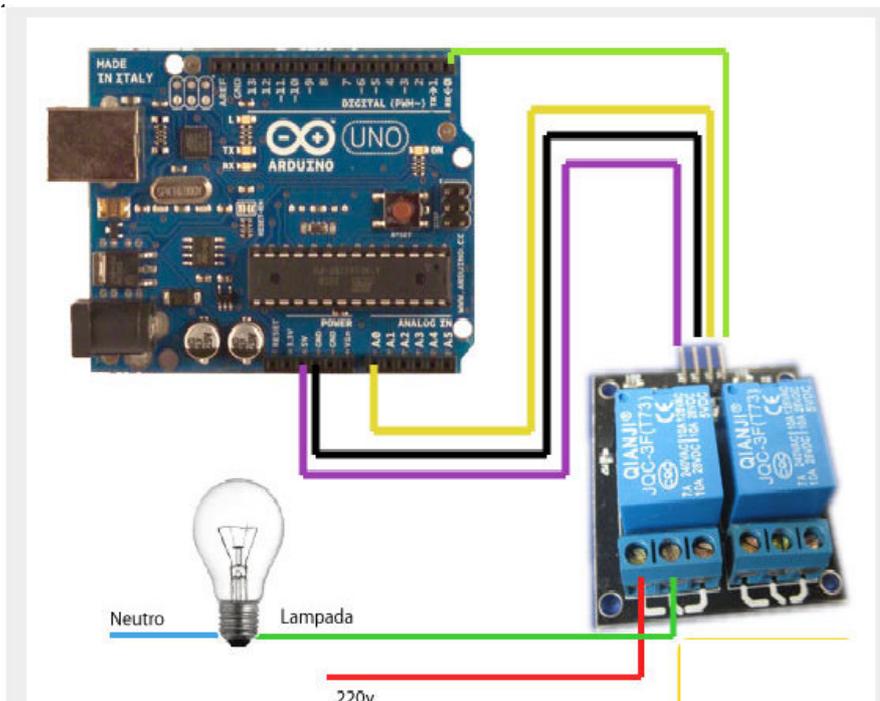


Fonte: do arquivo pessoal

Para que o Arduino possa operar corretamente o relé e, conseqüentemente, as luzes nas quais o relé estará atuando, faz-se necessário mudar o layout do esquema elétrico e os códigos de programação do Arduino Uno. Mais uma vez o teste será feito em um laboratório de eletrônica em escala reduzida de forma a simular a praça de máquinas e o centro de controle de máquinas do navio mercante. As luzes que o Arduino controlará estarão ligadas à rede de 110 Vac, lembrando que a tensão de trabalho do navio poderá ser de 110 ou 220 Vac.

O layout do esquema elétrico com o relé shield instalado na placa Arduino Uno e na rede elétrica ficou com a configuração mostrada na figura 10.

Figura 10 – Esquema elétrico com Arduino Uno e relé shield



Fonte: [www.vod.com.ng](http://www.vod.com.ng)

Não haverá mudança quanto à ligação do sensor de movimento PIR na placa Arduino Uno. Apenas foi adicionado o relé shield à placa de forma que, quando o sensor detectar movimento, o Arduino Uno acione o relé que, por sua vez, acionará a lâmpada respectiva à qual ele estará ligado.

Quanto ao código de programação, também não haverá grandes mudanças. A introdução do relé no circuito simplesmente faz com que seja necessário configurar um pino digital de saída para o relé. Isto é feito, primeiramente, inicializando-se uma variável, atribuindo a ela o número referente ao pino da placa Arduino Uno ao qual o relé estará conectado. Em seguida, adiciona-se uma linha de comando na função setup configurando o mesmo pino como output, isto é, como pino digital de saída. A programação na função loop permanecerá com a mesma estrutura, tendo-se o devido cuidado de efetuar a troca das variáveis nas respectivas linhas de códigos, isto é, nas linhas onde o programa irá executar a leitura do sinal do sensor e onde ele irá proceder ao acionamento do relé: quando o sensor, pela captação termal do corpo humano, causar um sinal de tensão no pino do Arduino, este irá ocasionar um sinal de tensão no pino em que está conectado o relé que, por sua vez, ligará a luz.

Com as modificações devidamente executadas, deu-se continuidade aos testes em bancadas no laboratório. Após terem sido feitas as corretas ligações da placa Arduino Uno, do

relé shield, da instalação do sensor de movimento PIR no ambiente selecionado e a conexão via USB do Arduino Uno ao computador, foi carregado o programa e colocado o sistema para operar. Por um período de 24 horas o sistema permaneceu operando e funcionou a contento. A lâmpada, ligada à rede de 110 V, acendia-se sempre que o sensor de movimento PIR detectava a presença ou o movimento de alguém no ambiente em que ele foi instalado; e durante um curto espaço de tempo, apagava-se quando o ambiente permanecia sem movimento humano. Foi necessário efetuar um ajuste no potenciômetro de tempo e de sensibilidade do sensor PIR, bem como ajustar o tempo de delay na linha de código do programa do Arduino Uno. Todos os componentes funcionaram de forma correta tanto na parte de controle quanto na parte física. Assim, o modo de funcionamento do protótipo apresentou resultados conforme esperado.

## 6 CONSIDERAÇÕES FINAIS

O trabalho desenvolvido teve o objetivo de idealizar um protótipo de automação para controle da iluminação de uma praça de máquinas desguarnecida de um navio mercante, com microcontrolador Arduino, a fim de dar mais segurança aos oficiais de máquinas que porventura estejam de serviço e precisem entrar no CCM para resetar alarme ou realizar qualquer outra atividade. Com a praça de máquinas desguarnecida, permanentemente às escuras (ainda que não totalmente apagada, mas com as luzes de emergência acesas), o oficial teria que andar pela praça de máquinas na penumbra desde o seu acesso até chegar ao CCM para acender as luzes (pois o quadro elétrico de iluminação encontra-se exatamente no CCM). Com a instalação deste sistema proposto, seria evitado que o mesmo tripulante se locomovesse pela praça de máquinas caminhando às escuras, na penumbra, prevenindo-o de um possível acidente.

Este projeto possibilita a automatização e o controle da iluminação de uma praça de máquinas desguarnecida de forma que, uma vez que um tripulante adentre ao recinto de acesso à praça de máquinas, o sistema possa atuar e acender todas as luzes da praça de máquinas e do centro de controle de máquinas, sem nenhuma ação humana.

O sistema é composto de um microcontrolador Arduino Uno, um sensor de movimento PIR, um relé shield e um computador para rodar o programa do Arduino. No lugar do relé shield poderia ser utilizado um relé SSR, porém teria que ser feita, previamente, uma análise do circuito e uma análise custo-benefício, e desde que sejam feitas as alterações corretamente no código de programação. De acordo com a viabilidade e com a aplicabilidade do sistema, outros sensores e outros tipos de relés poderiam ser empregados. O sensor deve ser instalado no lado de dentro com a face voltada para a porta de acesso à praça de máquinas, tantos quantos forem os acessos, um em cada porta de entrada. A placa Arduino Uno e o relé shield podem ser montados juntos em uma placa de circuito impresso, abrigado em um invólucro de acrílico ou outro material compatível, e todo o conjunto instalado no console do CCM. Outra alternativa seria instalar o Arduino Uno no computador do console do CCM e o relé montado no quadro de distribuição de iluminação da praça de máquinas. Durante o horário de expediente, quando as luzes precisam estar acesas durante todo o tempo, há a possibilidade de se deixar o sistema offline, instalando-se um interruptor ou via software.

No decorrer desta pesquisa, observou-se que os sistemas de controle e automação estão cada vez mais presentes na versão do mundo atual, nas residências, principalmente nas indústrias, no cotidiano da humanidade, porém há uma gama muito grande de infinitas

possibilidades que o uso da automação ainda pode oferecer. A utilização da placa microprocessada Arduino respondeu positivamente aos questionamentos de sua aplicação para o intento deste trabalho, certificando que através da solução proposta é possível conciliar tecnologia, economia e segurança.

Tendo a certeza do funcionamento do sistema, pode-se aumentar sua área de atuação para outros dispositivos. Pode-se realizar melhorias no sistema. Pode-se aumentar o número de ambientes a ser controlado, e também outros setores de iluminação. É possível ser proposta a implementação do sistema com mais funcionalidades de forma que o Arduino, por meio de shields, seja capaz de controlar e atuar na automação de outros equipamentos. O sistema se mostrou bastante flexível, permitindo adequar o projeto de acordo com a necessidade de cada aplicação, de forma simples e eficiente.

A grande vantagem da utilização da placa Arduino é a facilidade de uso e uma série de dispositivos que podem ser acrescentados, por ser um microcontrolador acessível, por conter um sistema de comunicação serial que pode ser conectado com qualquer computador que tenha conexão USB e por ser compatível com vários softwares, possibilitando, assim, o desenvolvimento de uma solução de baixo custo.

Após os estudos realizados nesta pesquisa, é possível observar que a proposta deste protótipo pode ser uma alternativa válida e satisfatória para a questão do binômio economia/segurança, que é o que se busca com afã nos dias atuais.

## REFERÊNCIAS BIBLIOGRÁFICAS

EVAN, Martin; NOBLE, Joshua; HOCHENBAUM, Jordan. **Arduino em ação**. Editora Novatec, São Paulo, 2013.

FRADEN, Jacob. **Handbook of modern sensors: physics, design and applications**. Fourth edition. Springer, California, 2010.

MCROBERTS, Michael. **Arduino básico**. Editora Novatec, São Paulo, 2011.

PEREIRA, Fábio. **Microcontroladores PIC: Programação em C**. Primeira edição. Editora Érica, 2003.

NETO, Reinaldo Oliveira. **Automação** de iluminação residencial utilizando microcontrolador Arduino e tablet ipad via wi-fi. Brasília, 2011.

SANTOS, Nuno Pessanha. **Arduino: introdução e recursos avançados**. Escola Naval. Rio de Janeiro, 2009.

CORPO Negro. Disponível em:  
<[www.sabereletronica.com.br](http://www.sabereletronica.com.br)>. Acesso em: jun. 2014

LEI de Stefan-Boltzmann. Disponível em:  
<[www.pt.wikipedia.org](http://www.pt.wikipedia.org)>. Acesso em: jun. 2014

LENTEs de Fresnel. Disponível em:  
<[www.infoescola.com](http://www.infoescola.com)>. Acesso em: jun. 2014