

DADOS GERAIS

Nomes dos Estudantes: Lucas Correa da Silva e Raphael Peroza

Curso Técnico: Técnico em Manutenção e Suporte em Informática

Professor Orientador: Uberdam Junior Cavaletti

Início: 13/08/2012 Término: 30/11/2012

Carga Horária: 100 horas

Problema**INTRODUÇÃO**

As casas inteligentes constituem já algum tempo como sendo objeto de desejo de muitas pessoas de todas as partes do mundo, mas de certa forma estão fora do alcance da maior parte da população pelo seu alto custo de implementação. A automação ganha cada vez mais força no Brasil, trazendo mais conforto, segurança e economia aos usuários.

Com a automação, você programa sua casa para agir de acordo com suas necessidades, sendo possível controlar iluminação, portas, portões, janelas, alarmes, som entre muitos outros equipamentos elétricos e eletrônicos.

Através da automação temos a possibilidade de adaptar uma casa para melhor acomodar seu usuário, podendo ser realizado projetos para atender as necessidades de cada pessoa, como por exemplo, auxiliar uma pessoa idosa ou com necessidades especiais.

Uma central de automação pode ser muito útil para realizar tarefas repetitivas, como ligar e desligar o filtro da piscina, fechar persianas e janelas, entre muitas outras possibilidades.

Na questão de segurança é muito útil para agilizar processos do cotidiano, como o simples fato de sair de casa e não precisar levar as chaves consigo, pois o acionamento das travas pode ser feito utilizando um aparelho de biometria, o qual pode armazenar todas as informações de cada usuário cadastrado, fazendo assim com que o imóvel saiba quem está em casa, adaptando o ambiente para aquela pessoa como, por exemplo, ligar a iluminação desejada ao chegar em casa como também regular a temperatura. Podemos ainda armazenar uma condição para cada dedo da pessoa, o que seria muito útil para definirmos mais de uma

Problema

programação para cada usuário.

Com um circuito de monitoramento remoto e alarme, a central avisa em tempo real o que está acontecendo em sua casa através de mensagens SMS, podendo ainda ver as imagens ao vivo pela internet de qualquer parte do mundo.

Um projeto de automação vai além da eficiência residencial, podendo ser direcionada para o ambiente corporativo, pois está diretamente relacionada a questões de energia gasta de forma eficaz, tornando-se extremamente importante para o meio ambiente.

Em poucas palavras a automação pode ser definida como o uso da tecnologia para facilitar algumas tarefas do dia a dia. Usando da utilização de sensores temos um leque muito grande de opções para automação sendo possível automatizar todo equipamento ou dispositivo que possuam um comando de funcionamento sendo ele elétrico ou eletrônico.

Este projeto teve o objetivo de desenvolver uma pequena automação residencial de baixo custo utilizando um servidor, o micro controlador arduíno e uma página web desenvolvida em PHP com a finalidade de controlar dispositivos usando-se da internet, podendo ser feita de qualquer lugar do planeta, algo muito útil para os dias de hoje.

PROBLEMA

Desenvolver uma solução para automação residencial de baixo custo onde seja possível controlar vários equipamentos elétricos de longa distância utilizando a internet como meio de comunicação

JUSTIFICATIVA

Com o aumento da renda dos brasileiros, é cada vez mais comum que as pessoas invistam em tecnologia para satisfazer os prazeres que até então eram considerados de difícil acesso pelo valor a ser gasto. Investir em automação ainda é um luxo que poucas pessoas se interessam pelo fato de ser um investimento de grandes valores. Pensando nisso juntamos um pouco de conhecimento de programação e alguns componentes eletrônicos encontrados facilmente na internet para desenvolver uma automação eficiente e de baixo custo visando atender um público que quer ter o poder da automação com um investimento muito baixo.

OBJETIVO GERAL

Problema

Atender um público que gosta de tecnologia e conforto buscando soluções de fácil implementação com um valor muito abaixo aos praticados no mercado.

OBEJETIVOS ESPECÍFICOS

Desenvolver uma solução baseada em programas e componentes de licença gratuita para atuarem em conjunto a fim de desenvolverem uma função preestabelecida.

Solução proposta

Usar a placa Arduino para servir como uma central controladora de reles com comandos vindos da internet para realizar tarefas a longa distância, como acender e apagar lâmpadas, ligar e desligar aparelhos eletrônicos abrir e fechar portões eletrônicos.

Embasamento teórico para a resposta.

O projeto foi tema da apresentação do SENAI Casa Aberta de Xanxerê na edição de 2012, desenvolvida com o objetivo de incentivar a busca pelo conhecimento da programação de forma descomplicada e divertida que se levada a sério pode ser transformada em grandes projetos.

Para um melhor entendimento, esse projeto será apresentado em três partes que terão seu conteúdo detalhado no decorrer do projeto sendo que a primeira trata de uma

Embasamento teórico para a resposta.

apresentação da placa microcontroladora e como foi realizada sua programação, em seguida veremos um pouco sobre a linguagem utilizada para desenvolver a interface de controle utilizada para enviar informações via WEB para o microcontrolador, tornando assim possível a realização de tarefas a longas distâncias, e por fim uma breve descrição sobre o servidor usado para dar o suporte necessário para alcançarmos nosso objetivo.

Na imagem 1 veremos uma ilustração das possíveis possibilidades de automação residencial.

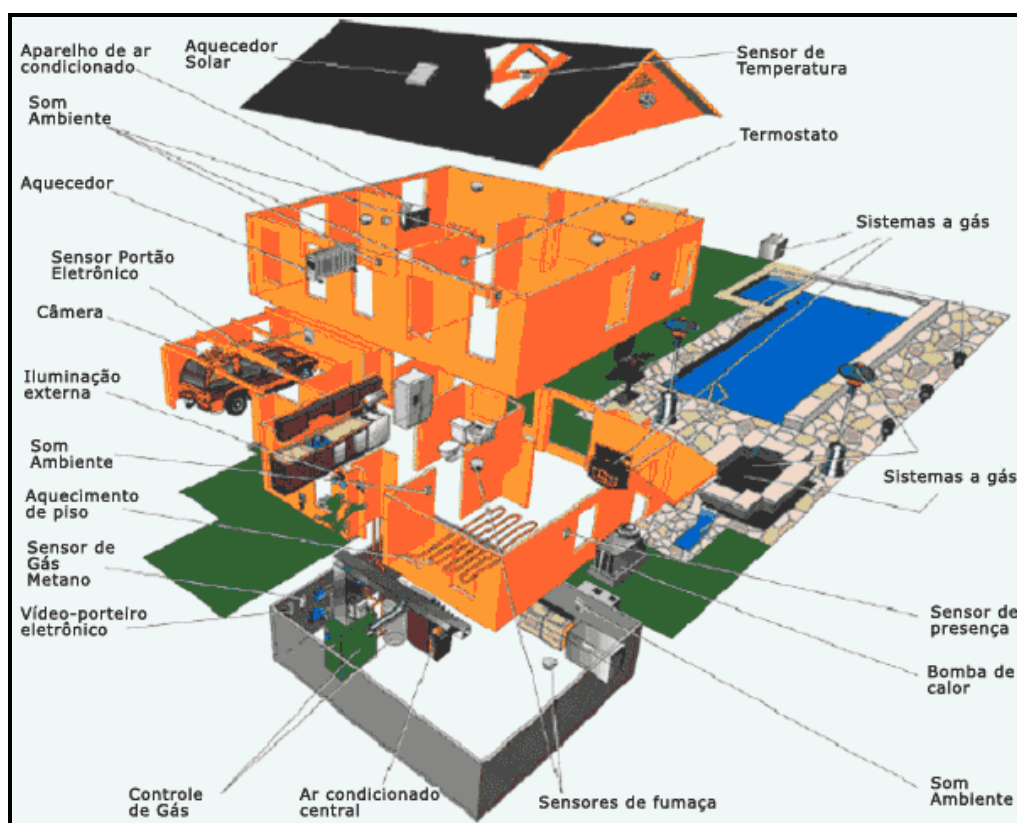


Figura 1 - Automação Residencial

OMICROCONTROLADOR ARDUINO

O Arduino surgiu em 2005, na Itália, com um professor chamado Massimo Banzi, que queria ensinar eletrônica e programação de computadores a seus alunos de design, para que eles usassem em seus projetos de arte, interatividade e robótica. Porém, ensinar eletrônica e programação para pessoas que não são da área não era uma tarefa simples, e outra

Embasamento teórico para a resposta.

dificuldade era a inexistência de placas poderosas e baratas no mercado.

Foi pensando nisso que Massimo e David Cuartielles decidiram criar sua placa própria, com a ajuda do aluno de Massimo, David Mellis, que ficou responsável por criar a linguagem de programação do Arduino. Várias pessoas conseguiram utilizar o Arduino e fazer coisas incríveis, surgindo assim essa febre mundial da eletrônica.

O QUE É UM ARDUINO?

Arduino é uma placa de código aberto usada para controle de entrada de dados, como sensores, e saída de dados, como motores e leds, com um regulador de tensão de 5 V, botão de reset, plugue de alimentação, pinos conectores, e alguns LEDs para facilitar a verificação do funcionamento. A porta USB já fornece alimentação enquanto estiver conectado ao computador, e a tensão de alimentação quando desconectado pode variar de 7 V a 12 V, graças ao regulador presente na placa.

No Arduino, informações são transmitidas de um computador para a placa através de Bluetooth, wireless, USB, infravermelho, etc. Essas informações devem ser traduzidas utilizando a linguagem Wiring baseada em C/C++. Abaixo veremos alguns modelos disponíveis do micro controlador arduino.

Embasamento teórico para a resposta.

Modelo	Descrição e tipo de conexão ao hospedeiro	Controlador
Serial Arduino	Serial DB9 para programação	ATmega8
Arduino Extreme	USB para programação	ATmega8
Arduino Mini	Versão em miniatura do Arduino utilizando montagem superficial	ATmega168
Arduino Nano	Versão menor que o <i>Arduino Mini</i> , energizado por USB e conectado por montagem superficial	ATmega168/328
LilyPad Arduino	Projeto minimalista para aplicações portáteis, utilizando montagem superficial	ATmega168
Arduino NG	USB para programação	ATmega8
Arduino NG plus	USB para programação	ATmega168
Arduino BT	Interface Bluetooth para comunicação	ATmega168
Arduino Diecimila	Interface USB	Atmega168 em um pacote <i>DIL28</i>
Arduino Duemilanove	<i>Duemilanove</i> significa "2009" em italiano. É energizado via USB/DC, com alternância automática	Atmega168 (Atmega328 para a versão mais nova)
Arduino Mega	Montagem superficial	ATmega1280 para E/S adicionais e memória
Arduino Uno	Utiliza Atmega8U2 para driver conversor Serial-USB, ao invés do FTDI. "Uno" significa "Um" em italiano. Esta versão é considerada a versão 1.0 do projeto, sendo que as placas que sucederem a esta serão referenciadas.	ATmega328

Figura 2 - Modelos Arduino

Nessa imagem 3 podemos identificar os principais componentes do arduino tendo uma visão simplificada sobre as partes que formam o micro controlador.

Embasamento teórico para a resposta.

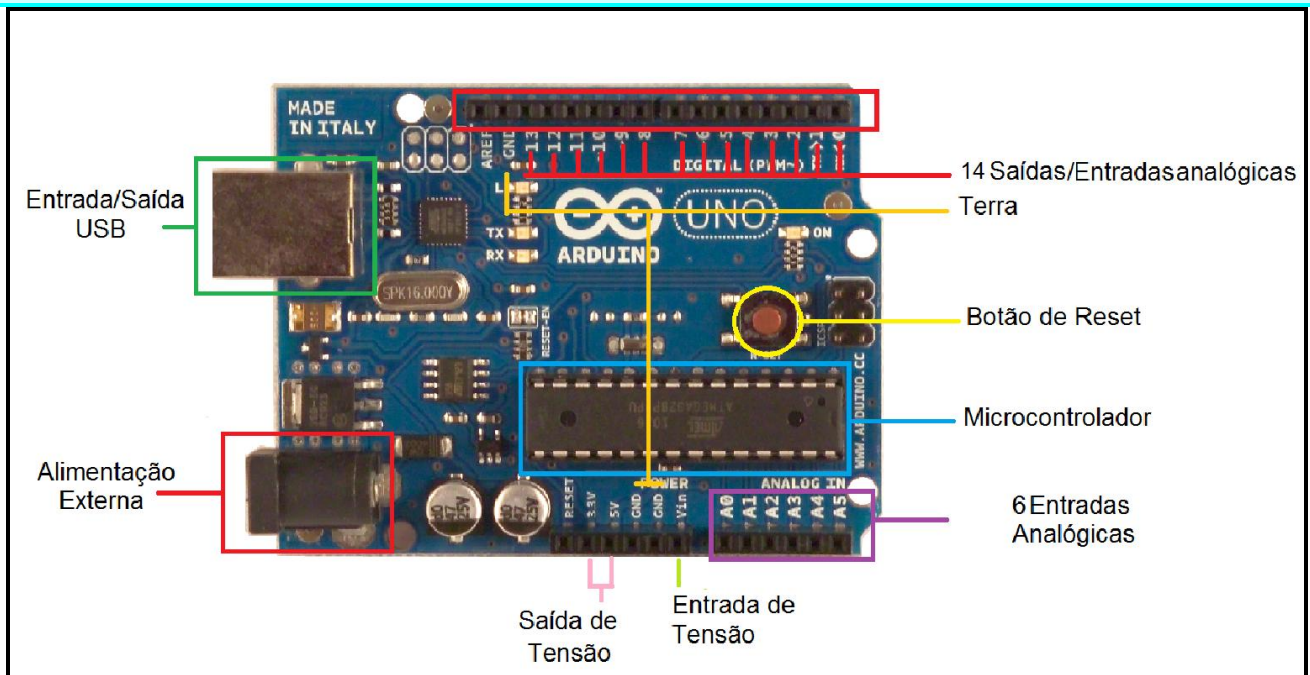


Figura 3 - Componentes

Arduino pode monitorar o ambiente através da recepção de entrada a partir de uma variedade de sensores e pode controlar vários circuitos como luzes, motores e outros atuadores. Projetos arduino podem ser stand-alone ou podem comunicar com software rodando em um computador.

As placas podem ser construídas à mão ou compradas prontas, o software pode ser baixado gratuitamente. Os projetos de hardware de referência estão disponíveis sob uma licença de código aberto, você é livre para adaptá-los às suas necessidades.

Para que possamos ter acesso ao micro controlador é necessário realizar o download da IDE, que nada mais é do que o software de programação desenvolvido em JAVA utilizado para desenvolver e gravar toda a programação que será executada pelo hardware.

O software está disponível para os três principais sistemas operacionais, sendo eles Mac, Linux e Windows. Depois de realizarmos o seu download, faz-se necessário à instalação do drive da placa do micro controlador arduino que assim como o software pode ser baixado diretamente da página oficial do dispositivo em www.arduino.cc.

Na imagem 4 podemos observar como é apresentada a interface de programação do micro controlador arduino.

Embasamento teórico para a resposta.

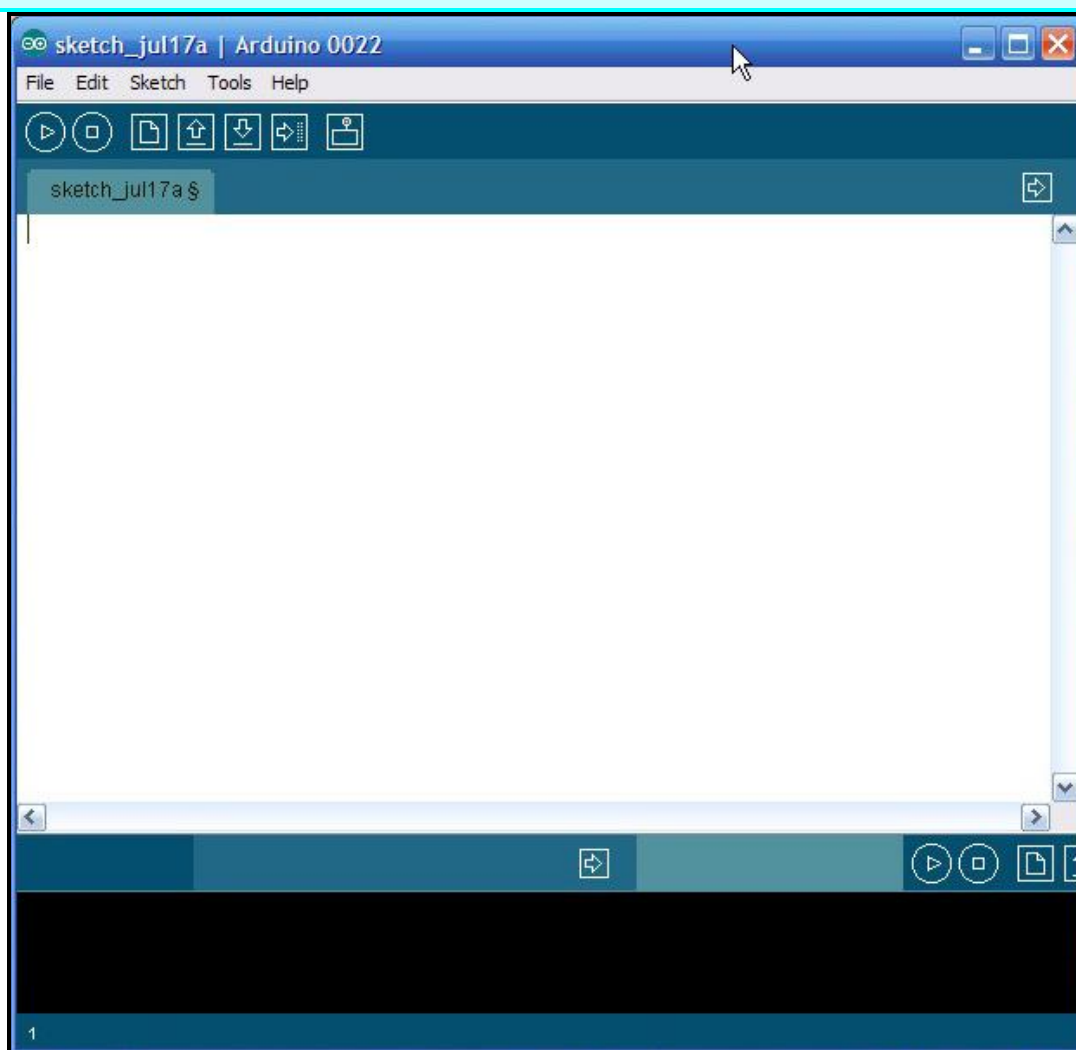


Figura 4 - IDE Arduino

Sua interface é bastante simplificada, trazendo consigo uma vasta biblioteca de programações já pré-configuradas, o que facilita muito o desenvolvimento de projetos. Por se tratar de um produto open source, conta com uma comunidade que cresce diariamente em todo o mundo, facilitando a troca de ideias e experiências entre os interessados pela arte da programação.

No site também é possível encontrar toda a documentação necessária para aprender a começar no mundo da programação.

PRINCIPAIS COMANDOS DISPONÍVEIS ATRAVÉS DOS BOTÕES:

Embasamento teórico para a resposta.

Verify: Verifica se o código contém erros.

Upload: Compila o código e faz o upload para o Arduino.

New: Cria um novo sketch.

Open: Exibe um menu com todas as sketches no seu sketchbook. Ao clicar em uma, a mesma será aberta na janela atual.

Save: Salva sua sketch.

Serial Monitor: Abre o monitor serial.

Comandos adicionais são encontrados através dos menus: File Edit, Sketch, Tools, Help. As funções disponíveis pelos menus, File, Edit e Help são semelhantes a outros programas bem conhecidos e, por isso, não iremos detalhá-las aqui.

menu Sketch

Verify/Compile - Verifica se seu código tem erros

Import Library - Adiciona bibliotecas ao seu programa

Show sketchfolder - Abre a pasta onde o programa está salvo

Add File... - Adiciona um arquivo fonte ao programa. O novo arquivo aparece em uma nova aba

menu Tools

Auto format - Formata o código para uma melhor leitura, alinhando as chaves e indentando seu conteúdo.

Board - Seleciona o kit de desenvolvimento onde se deseja realizar o projeto.

Serial Port - Mostra todas as portas seriais que o computador possui.

Burn Bootloader - Permite gravar um bootloader no kit de desenvolvimento do Arduino

CONCEITOS E SINTAXE DA LINGUAGEM DE PROGRAMAÇÃO

A plataforma de implementação dos programas em Arduino é baseada nas linguagens C/C++, preservando sua sintaxe na declaração de variáveis, na utilização de operadores, na manipulação de vetores, na conservação de estruturas, bem como é uma linguagem sensível a maiúsculas e minúsculas.

Contudo, ao invés de uma função `main()`, o Arduino necessita de duas funções elementares: `setup()` e `loop()`.

SETUP E LOOP

Pode-se dizer que todo código criado para o Arduino deve obrigatoriamente possuir duas funções para que o programa funcione corretamente: a função `setup()` e a função `loop()`. Essas duas funções não utilizam parâmetros de entrada e são declaradas como void. Não é necessário invocar a função `setup()` ou a função `loop()`. Ao compilar um programa para o Arduino, o compilador irá, automaticamente, inserir uma função `main` que invocará ambas as funções.

setup()

A função `setup` é utilizada para inicializar variáveis, configurar o modo dos pinos e incluir bibliotecas. Esta função é executada automaticamente uma única vez, assim que o kit Arduino é ligado ou resetado.

SINTAXE:

Embasamento teórico para a resposta.

```
void setup()  
{  
.  
:  
}
```

loop()

A função loop faz exatamente o que seu nome sugere: entra em looping (executa sempre o mesmo bloco de código), permitindo ao seu programa executar as operações que estão dentro desta função. A função loop() deve ser declarada após a função setup().

SINTAXE:

```
void loop()  
{  
.  
:  
}
```

HIGH E LOW

Quando estamos lendo ou escrevendo em um pino digital há apenas dois valores que um pino pode ter: HIGH (alto) e LOW (baixo).

HIGH

O significado de HIGH (em referência a um pino) pode variar um pouco dependendo se este pino é uma entrada (INPUT) ou saída (OUTPUT). Quando um pino é configurado como INPUT com a função pinMode, e lido com a função digitalRead, o micro controlador considera como HIGH se a voltagem for de 3 Volts ou mais. Um pino também pode ser configurado como

Embasamento teórico para a resposta.

um INPUT com o pinMode, e posteriormente receber um HIGH com um digitalWrite, isto vai “levantar” o resistor interno de 20 KOhms que vai manter a leitura do pino como HIGH a não ser que ela seja alterada para LOW por um circuito externo. Quando um pino é configurado como OUTPUT com o pinMode, e marcado como HIGH com o digitalWrite, ele está a 5 Volts. Neste estado ele pode enviar corrente para, por exemplo, acender um LED que está conectado com um resistor em série ao terra, ou a outro pino configurado como OUTPUT e marcado como LOW.

LOW

O significado de LOW também pode variar dependendo do pino ser marcado como INPUT ou OUTPUT. Quando um pino é configurado como INPUT com a função pinMode, e lido com a função digitalRead, o micro controlador considera como LOW se a voltagem for de 2 Volts ou menos. Quando um pino é configurado como OUTPUT com a função pinMode, e marcado como LOW com a função digitalWrite, ele está a 0 Volts. Neste estado ele pode “drenar” corrente para, por exemplo, acender um LED que está conectado com um resistor em série ao +5 Volts, ou a outro pino configurado como OUTPUT e marcado como HIGH.

INPUT E OUTPUT

Pinos digitais podem ser tanto de INPUT como de OUTPUT. Mudar um pino de INPUT para OUTPUT com pinMode() muda drasticamente o seu comportamento elétrico.

INPUT

Os pinos do Arduino (Atmega) configurados como INPUT com a função pinMode() estão em um estado de alta impedância. Pinos de entrada são válidos para ler um sensor, mas não para energizar um LED.

OUTPUT

Pinos configurados como OUTPUT com a função pinMode() estão em um estado de baixa impedância. Isto significa que eles podem fornecer grandes quantidades de corrente para

Embasamento teórico para a resposta.

outros circuitos. Os pinos do Atmega podem fornecer corrente positiva ou drenar corrente negativa até 40 mA (milliamperes) 22 mA para outros dispositivos ou circuitos. Isto faz com que eles sejam úteis para energizar um LED mais funcionais para a leitura de sensores. Pinos configurados como OUTPUT também podem ser danificados ou destruídos por curto-circuitos com o terra ou com outros pontos de 5 Volts. A quantidade de corrente fornecida por um pino do Atmega também não é suficiente para ativar muitos relês e motores e, neste caso, algum circuito de interface será necessário.

FUNÇÕES

Funções de Entrada e Saída Digital
`pinMode()` Configura o pino especificado para que se comporte ou como uma entrada ou uma saída. Deve-se informar o número do pino que se deseja configurar e em seguida, se o pino será determinado como uma entrada (INPUT) ou uma saída (OUTPUT).

SINTAXE:

`pinMode(pino, modo);`

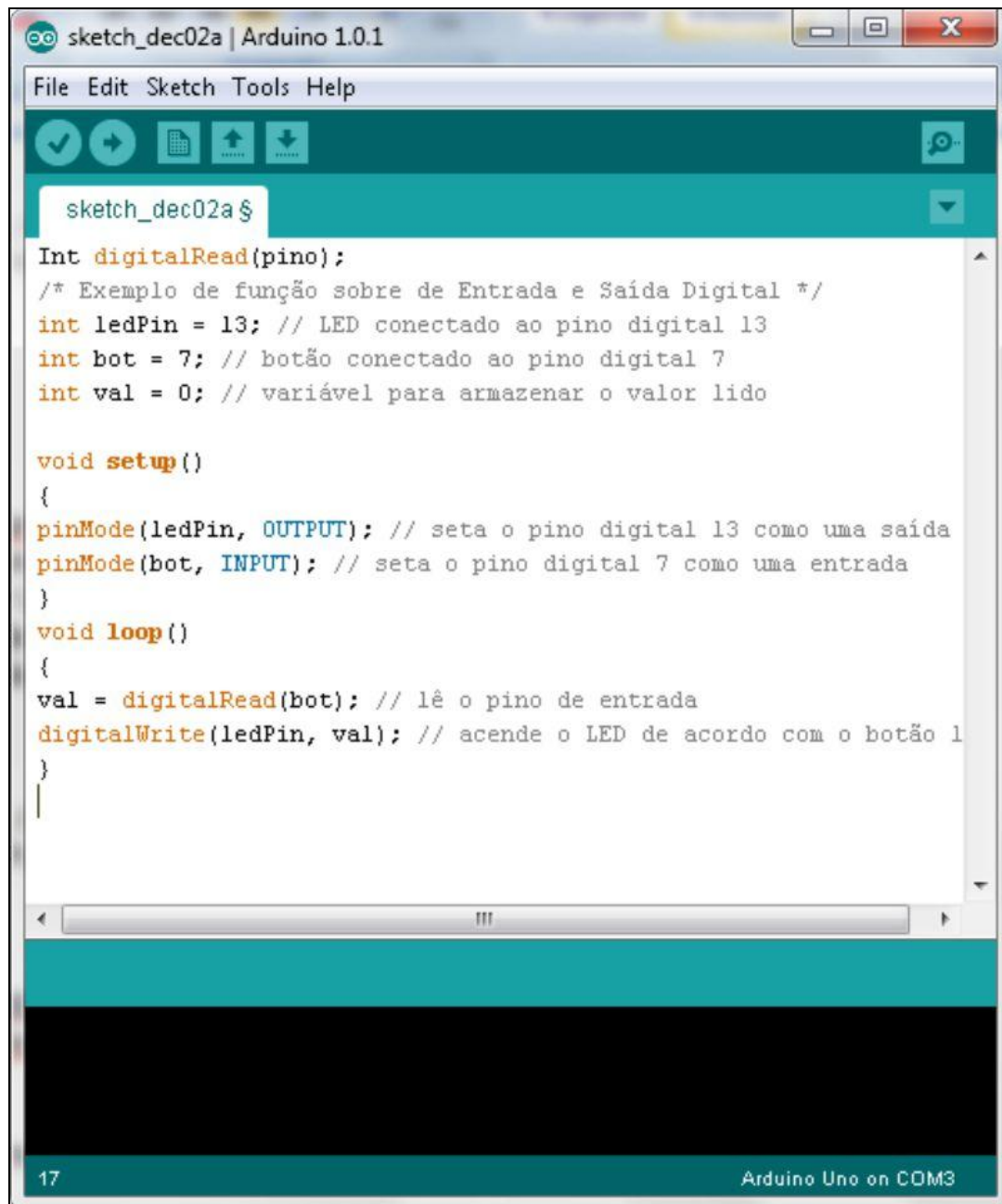
`digitalWrite()` Escreve um valor HIGH ou LOW em um pino digital. Se o pino foi configurado como uma saída, sua voltagem será determinada ao valor correspondente: 5V para HIGH e 0V para LOW. Se o pino está configurado como uma entrada, HIGH levantará o resistor interno de 20KOhms e LOW rebaixará o resistor.

SINTAXE:

`digitalWrite(pino, valor);`

`digitalRead()` Lê o valor de um pino digital especificado e retorna um valor HIGH ou LOW.

SINTAXE:



```
sketch_dec02a | Arduino 1.0.1
File Edit Sketch Tools Help
[Icons]
sketch_dec02a $
Int digitalRead(pino);
/* Exemplo de função sobre de Entrada e Saída Digital */
int ledPin = 13; // LED conectado ao pino digital 13
int bot = 7; // botão conectado ao pino digital 7
int val = 0; // variável para armazenar o valor lido

void setup()
{
  pinMode(ledPin, OUTPUT); // seta o pino digital 13 como uma saída
  pinMode(bot, INPUT); // seta o pino digital 7 como uma entrada
}
void loop()
{
  val = digitalRead(bot); // lê o pino de entrada
  digitalWrite(ledPin, val); // acende o LED de acordo com o botão lido
}
|
17 Arduino Uno on COM3
```

Figura 5 - anexo 1

Essa função transfere para o pino 13, o valor lido no pino 7 que é uma entrada.

FUNÇÕES DE ENTRADA E SAÍDA ANALÓGICA

Embasamento teórico para a resposta.

`analogWrite()` - PWM Pulse Width Modulation ou Modulação por Largura de Pulso (MLP) é um método para obter resultados analógicos com meios digitais.

Essa função basicamente escreve um sinal analógico. Ela pode ser usada para acender um LED variando o seu brilho, ou girar um motor com velocidade variável. Depois de realizar um `analogWrite()`, o pino gera uma onda quadrada estável com o ciclo de rendimento especificado até que um `analogWrite()`, um `digitalRead()` ou um `digitalWrite()` seja usado no mesmo pino. Em kits Arduino com o chip ATmega168, esta função está disponível nos pinos 3,5,6,9,10 e 11.

Kits Arduino mais antigos com um ATmega8 suportam o `analogWrite()` apenas nos pinos 9,10 e 11. As saídas PWM geradas pelos pinos 5 e 6 terão rendimento de ciclo acima do esperado. Isto se deve às interações com as funções `millis()` e `delay()`, que compartilham o mesmo temporizador interno usado para gerar as saídas PWM. Para usar esta função deve-se informar o pino ao qual deseja escrever e em seguida informar um valor entre 0 (pino sempre desligado) e 255 (pino sempre ligado).

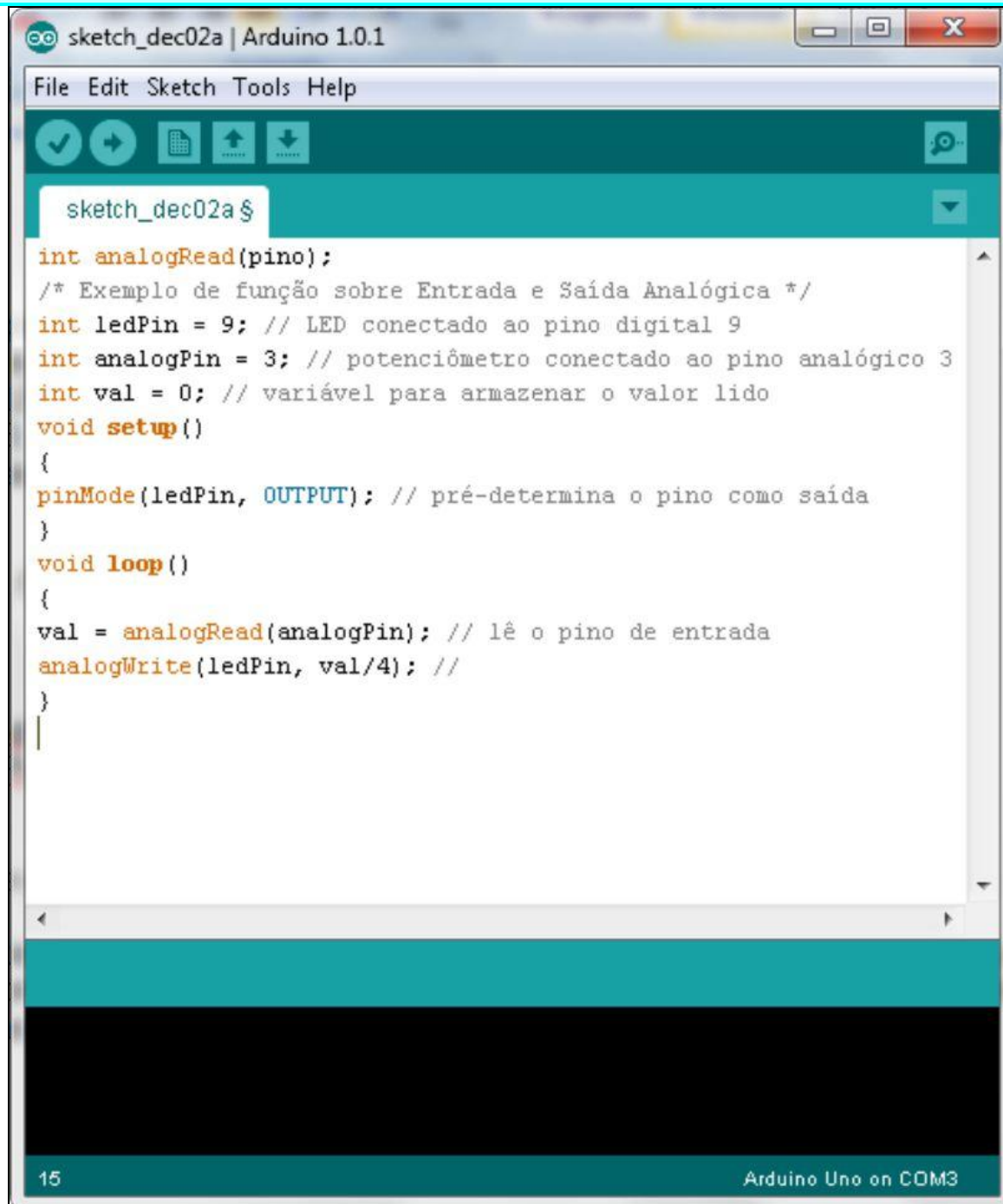
SINTAXE:

`analogWrite(pino, valor);`

`analogRead()` Lê o valor de um pino analógico especificado. O kit Arduino contém um conversor analógico-digital de 10 bits com 6 canais. Com isto ele pode mapear voltagens de entrada entre 0 e 5 Volts para valores inteiros entre 0 e 1023. Isto permite uma resolução entre leituras de 5 Volts / 1024 unidades ou 0,0049 Volts (4.9 mV) por unidade.

SINTAXE:

Embasamento teórico para a resposta.



```
sketch_dec02a $
int analogRead(pino);
/* Exemplo de função sobre Entrada e Saída Analógica */
int ledPin = 9; // LED conectado ao pino digital 9
int analogPin = 3; // potenciômetro conectado ao pino analógico 3
int val = 0; // variável para armazenar o valor lido
void setup()
{
  pinMode(ledPin, OUTPUT); // pré-determina o pino como saída
}
void loop()
{
  val = analogRead(analogPin); // lê o pino de entrada
  analogWrite(ledPin, val/4); //
}
```

15 Arduino Uno on COM3

Figura 6 - anexo 2

Torna o brilho de um LED proporcional ao valor lido em um potenciômetro.

Tempo millis() Retorna o número de milissegundos desde que o kit Arduino começou a executar o programa. Este número extrapolará (voltará a zero) depois de aproximadamente 50 dias.

SINTAXE

unsigned long tempo;

void loop

{

.

:

tempo = millis()

}

delay() Suspende a execução do programa pelo tempo (em milissegundos)

especificado. Em um segundo há 1.000 milissegundos.

Sintaxe:

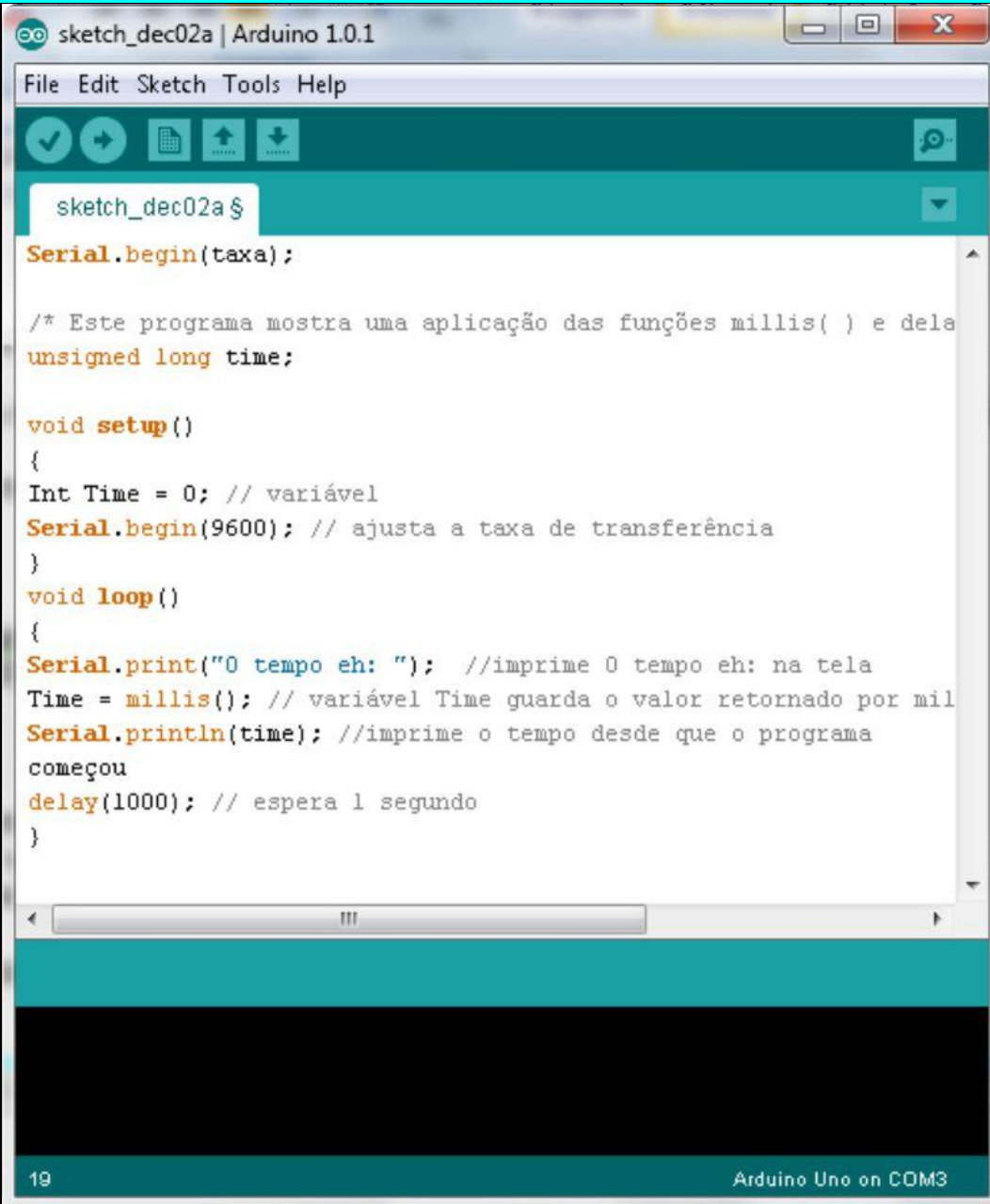
delay(tempo);

COMUNICAÇÃO SERIAL

Serial.begin() Ajusta a taxa de transferência em bits por segundo para uma transmissão de dados pelo padrão serial. Para comunicação com um computador use uma destas taxas: 300, 1200, 2400, 4800, 9600, 14400, 19200, 28800, 57600, 115200. Pode-se, entretanto, especificar outras velocidades por exemplo, para comunicação através dos pinos 0 e 1 com um componente que requer uma taxa específica.

Sintaxe:

Embasamento teórico para a resposta.



```
sketch_dec02a | Arduino 1.0.1
File Edit Sketch Tools Help
[Icons]
sketch_dec02a $
Serial.begin(taxa);

/* Este programa mostra uma aplicação das funções millis( ) e dela
unsigned long time;

void setup()
{
  Int Time = 0; // variável
  Serial.begin(9600); // ajusta a taxa de transferência
}

void loop()
{
  Serial.print("O tempo eh: "); //imprime O tempo eh: na tela
  Time = millis(); // variável Time guarda o valor retornado por mil
  Serial.println(time); //imprime o tempo desde que o programa
  começou
  delay(1000); // espera 1 segundo
}
```

19 Arduino Uno on COM3

Figura 7 - anexo 3

POR QUE USAR ARDUINO?

Baixo custo - Uma pessoa pode comprar um Arduino pagando em torno de R\$ 65,00.

Software para várias plataformas - Microsoft Windows, Mac OS X e Linux.

Embasamento teórico para a resposta.

Linguagem simples - Os desenvolvedores do Arduino tentam manter sua linguagem fácil de usar para iniciantes, mas flexível o bastante para usuários avançados.

Software livre - O Arduino é completamente um software livre. Se quiser construir seu próprio software ou modificar um, você é livre para isso. Além disso, o Web site oficial do Arduino contém um wiki extensivo no qual amostras de código e exemplos são compartilhados livremente. Existe uma comunidade ativa para usuários, por isso, existe uma quantidade enorme pessoas que podem te ajudar.

O CÓDIGO UTILIZADO

A seguir veremos o código utilizado para a realização da automação. Ele encontra-se de forma reduzida para que possamos entender a lógica utilizada, sendo que para termos todas as funções é só multiplicar o código adicionando as funções desejadas, uma vez que a principal função é receber as informações da página web através da porta USB e determinar se a porta do microcontrolador irá definir as portas como HIGH (ligada) ou LOW (desligada) como vimos no detalhamento do conceitos e sintaxe da Linguagem de Programação.

Embasamento teórico para a resposta.



```
sketch_dec02a | Arduino 1.0.1
File Edit Sketch Tools Help

sketch_dec02a $

int ledPin11 = 11; // declarando a variável

void setup() {

    Serial.begin(9600);

    pinMode(ledPin11, OUTPUT);

}

void loop()

{
    char character; //variável do tipo char chamada character

    character = Serial.read(); // função que lê os valores da porta s

    if(character == 'a') // se o valor da variável character for igual

    {
        digitalWrite(ledPin11, HIGH); // função para ligar o pino 11
    }

    else

        if(character == 'b') //
        {
            digitalWrite(ledPin11, LOW); // função que irá desligar
        }
}
```

9 Arduino Uno on COM3

Figura 8 - anexo 4

Embasamento teórico para a resposta.

Usar a placa arduino para servir como uma central controladora de reles com comandos vindos da internet para realizar tarefas a longa distância, como acender e apagar lâmpadas, ligar e desligar aparelhos eletrônicos abrir e fechar portões eletrônicos.

SOBRE O SERVIDOR

Nessa aplicação foi utilizado o XAMPP que é um pacote que consiste de servidor Web Apache, um banco de dados MySQL, e interpretadores para linguagens Perl e PHP. Seu nome vem de, X (para qualquer sistema operacional), A (Apache), M (MySQL), P (PHP) e P (Perl). XAMPP é independente de plataforma e está licenciado sob a GNU GPL. Existem versões para Linux, Windows, Mac e Solaris.

Uma vantagem do XAMPP é muito simples e rápido de ser instalado onde pode-se montar em seu ambiente de desenvolvimento da máquina de qualquer aplicação Web que usa PHP o que é fundamental para o funcionamento do projeto, já que desenvolvemos a parte de interface com esta linguagem.

PHP

O PHP sucede de um produto mais antigo, chamado PHP/FI que foi criado por Rasmus Lerdorf em 1995. Ele nomeou esta série de script de 'Personal Home Page Tools'. Como mais funcionalidades foram requeridas, Rasmus escreveu uma implementação C muito maior, que era capaz de comunicar-se com base de dados, e possibilitava à usuários desenvolver simples aplicativos dinâmicos para Web. Rasmus resolveu disponibilizar o código fonte do PHP/FI para que todos pudessem ver, e também usá-lo, bem como fixar bugs e melhorar o código.

PHP/FI, que significa Personal Home Page / Forms Interpreter, incluía algumas funcionalidades básicas do PHP que nós conhecemos hoje. Ele usava variáveis no estilo Perl, interpretação automática de variáveis vindas de formulário e sintaxe embutida no HTML. A sua própria sintaxe era similar a do Perl, porém muito mais limitada, simples, e um pouco

Embasamento teórico para a resposta.

inconsistente.

Em 1997, PHP/FI 2.0, a segunda versão da implementação C, obteve milhares de usuários ao redor do mundo (estimado), com aproximadamente 50,000 domínios reportando que tinha PHP/FI 2.0 instalado, agarrinando 1% dos domínios da Internet. Enquanto isto havia milhares de pessoas contribuindo com pequenos códigos para o projeto, e ainda assim

O PHP/FI 2.0 foi oficialmente lançado somente em Novembro de 1997, após perder a maior parte de sua vida em versões betas. Ele foi rapidamente substituído pelos alphas do PHP 3.0.

Sua versão atual é a 5.04. O principal objetivo de PHP5 foi melhorar os mecanismos de POO para solucionar as carências das versões anteriores. Um passo necessário para conseguir que PHP seja uma linguagem apta para todo tipo de aplicações e meios, inclusive os mais exigentes.

CÓDIGO PHP

Embasamento teórico para a resposta.

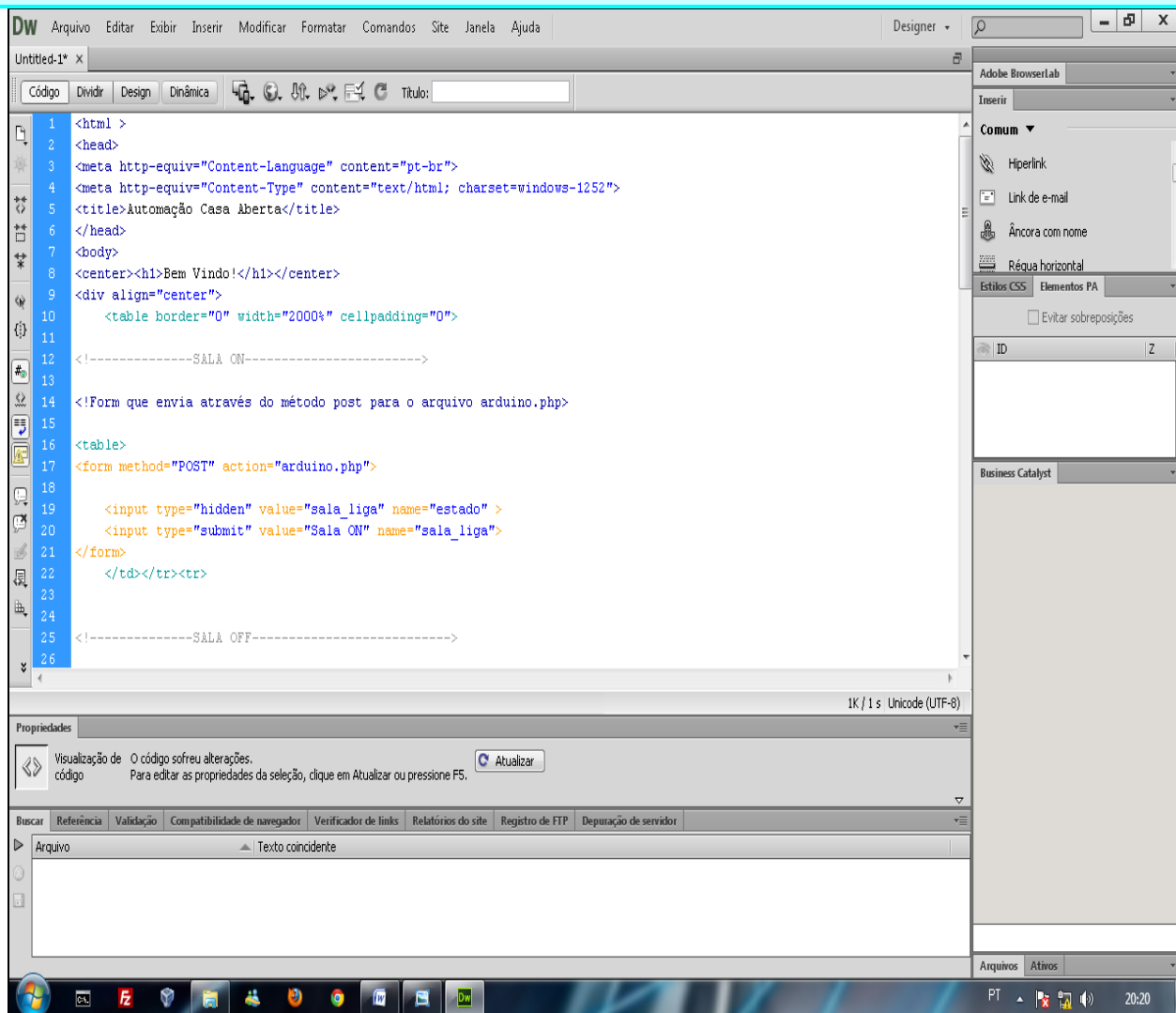


Figura 9 - código php – anexo 4

ANEXO 1

```
Int digitalRead(pino);
```

```
/* Exemplo de função sobre de Entrada e Saída Digital */
```

```
int ledPin = 13; // LED conectado ao pino digital 13
```

```
int bot = 7; // botão conectado ao pino digital 7
```

```
int val = 0; // variável para armazenar o valor lido
```

```
void setup()
```

```
{
```

```
pinMode(ledPin, OUTPUT); // seta o pino digital 13 como uma saída
```

```
pinMode(bot, INPUT); // seta o pino digital 7 como uma entrada
```

Embasamento teórico para a resposta.

```
}  
void loop()  
{  
  val = digitalRead(bot); // lê o pino de entrada  
  digitalWrite(ledPin, val); // acende o LED de acordo com o botão 13  
}
```

ANEXO 2

```
int analogRead(pino);  
/* Exemplo de função sobre Entrada e Saída Analógica */  
int ledPin = 9; // LED conectado ao pino digital 9  
int analogPin = 3; // potenciômetro conectado ao pino analógico 3  
int val = 0; // variável para armazenar o valor lido  
void setup()  
{  
  pinMode(ledPin, OUTPUT); // pré-determina o pino como saída  
}  
void loop()  
{  
  val = analogRead(analogPin); // lê o pino de entrada  
  analogWrite(ledPin, val/4); //  
}
```

ANEXO 3

```
Serial.begin(taxa);  
  
/* Este programa mostra uma aplicação das funções millis( ) e delay( ) e Comunicação  
Serial */  
unsigned long time;  
  
void setup()
```

Embasamento teórico para a resposta.

```
{
  Int Time = 0; // variável
  Serial.begin(9600); // ajusta a taxa de transferência
}
void loop()
{
  Serial.print("O tempo eh: "); //imprime O tempo eh: na tela
  Time = millis(); // variável Time guarda o valor retornado por millis()
  Serial.println(time); //imprime o tempo desde que o programa
  começou
  delay(1000); // espera 1 segundo
}
```

ANEXO 4

```
<html >
<head>
<meta http-equiv="Content-Language" content="pt-br">
<meta http-equiv="Content-Type" content="text/html; charset=windows-
1252">
<title>Automação Casa Aberta</title>
</head>
<body>
<center><h1>Bem Vindo!</h1></center>
<div align="center">
  <table border="0" width="2000%" cellpadding="0">

    <!--SALA ON-->

    <!--Form que envia através do método post para o arquivo arduino.php>

  <table>
```

Embasamento teórico para a resposta.

```
<form method="POST" action="arduino.php">
```

```
    <input type="hidden" value="sala_liga" name="estado" >
```

```
    <input type="submit" value="Sala ON" name="sala_liga">
```

```
</form>
```

```
</td></tr><tr>
```

<!--SALA OFF-->

<!--Form que envia através do método post para o arquivo arduino.php-->

```
<form method="POST" action="arduino.php">
```

```
    <input type="hidden" value="sala_desliga" name="estado" >
```

```
    <input type="submit" value="Sala OFF" name="sala_desliga">
```

```
</form>
```

```
</tr><tr><td>
```

```
</table>
```

```
<?php
```

```
    $port = fopen('COM3', 'w');
```

```
    // armazena o valor de retorno da função fopen() na variável $post
```

```
    // fopen() recebe como parâmetro o caminho da porta e o parâmetro para  
    escrever "w" (write) na COM 3.
```

Embasamento teórico para a resposta.

//Se o valor da variável "estado" for igual a "sala_liga", então escreve na porta serial o valor "a"

```
if ($_POST['estado']=='sala_liga')  
  
{  
  
    echo "<img src='imagens/lampada.jpg'/>";  
    //Imprime a imagem de lampada ligada na tela  
  
    fwrite($port, 'a');  
    //função para escrever na porta serial o caracter "a"  
  
}
```

//Se o valor da variável "estado" for igual a "sala_liga", então escreve na porta serial o valor "b"

```
if ($_POST['estado']=='sala_desliga')  
{  
  
    echo "<img src='imagens/lampadadesligada.jpg'/>";  
    //Imprime uma imagem de lâmpada desligada na tela  
  
    //função para escrever na porta serial o caracter "b"  
    fwrite($port, 'b');  
  
}
```

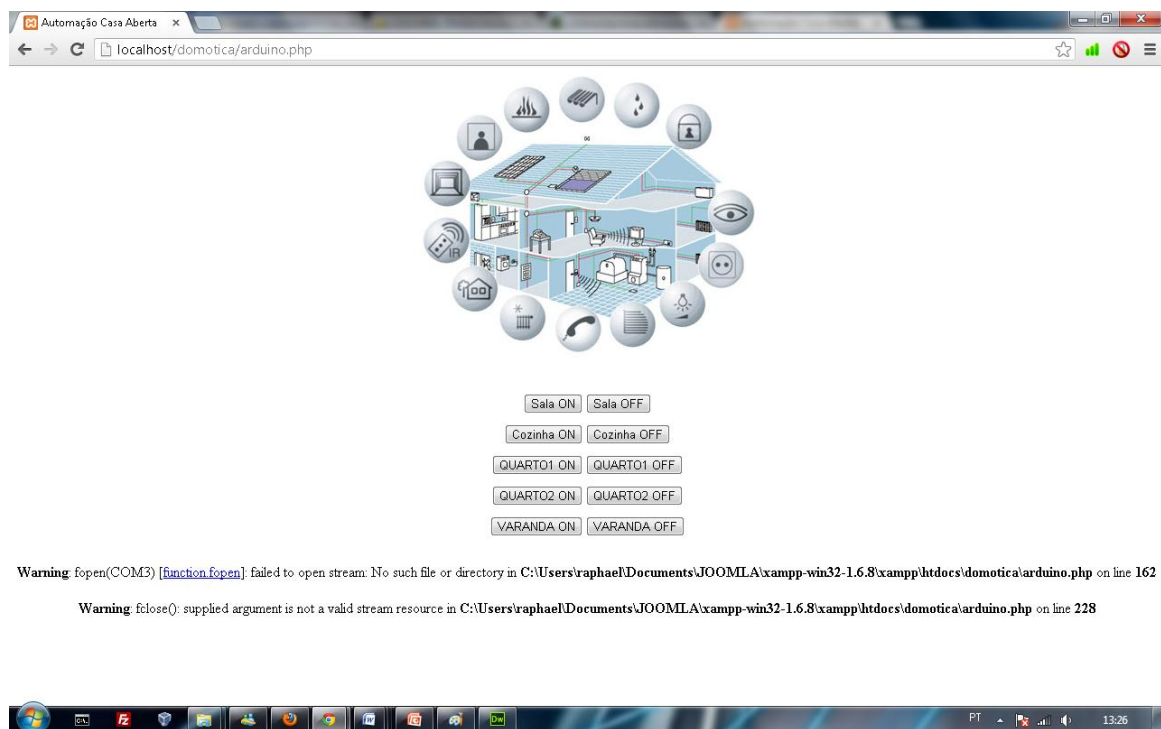
```
fclose($port);  
//Fecha a porta serial  
?>
```

Embasamento teórico para a resposta.

```
</div>
</body>
</html>
```

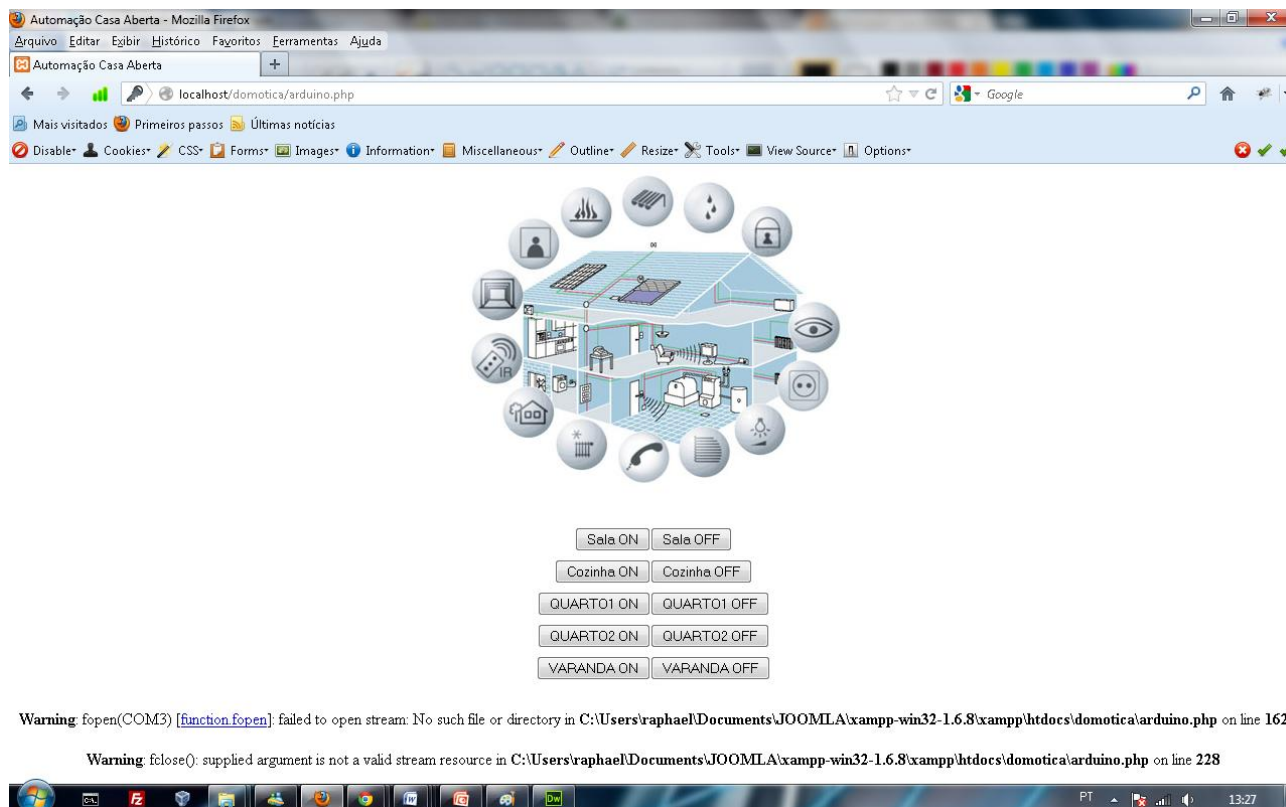
Páginas em Funcionamento

Google Chrome



Embasamento teórico para a resposta.

Mozilla Firefox



Conclusão

É possível fazer o monitoramento e o controle de uma residência via WEB, possibilitando um controle amplo através de um computador, celular, etc. A residência a qual terá o monitoramento de vera esta com todo um circuito desenvolvido propriamente para esta função, como janelas motorizadas persianas da mesma maneira, lâmpadas interligadas a um sistema de rele, podemos dizer que a residência devera passar por uma total modificação.

Em questão conforto será um ponto alto, pois o usuário poderá fazer todas as modificações de acordo com suas necessidades ou vontade, exemplo o usuário pode acende a lâmpada do quarto estando na sala ele poderá abrir ou fechar porta e portões de qualquer

Conclusão

lugar, tudo isso com apenas alguns toques.

Outro ponto a ser colocado é a segurança, pois todo esse projeto tem um lado totalmente voltado a segurança do usuário, através de sensores e câmeras de segurança que ao detectarem um movimento suspeito mandaram um aviso ao usuário através de sons, ou até mesmo mensagens, fazendo com que o usuário fique por dentro de tudo que está acontecendo em sua propriedade.

Na casa terá um servidor ligado a internet que será o gerenciador de tudo o que está ligado a casa e ao sistema este servidor deverá estar ligado a todo tempo para as funções funcionarem corretamente. Mas o sistema também disponibilizará de uma função caso aconteça algum erro com o sistema ou falta de AC, para que o usuário não perca o controle de sua residência, o sistema poderá ter vários usuários gravadores e com permissões diferentes para cada um deles, como um usuário com controle total sobre a residência outro apenas com permissão para ligar as lâmpadas, assim definindo o poder de controle de cada um, tudo isso para o conforto e comodidade do usuário.

Referências

DISPONIVEL EM: <http://www.controlled.com.br/site/servicos/43-automacao.html>

Acesso em: 05/09/2012

DISPONIVEL EM: <http://www.ppgia.pucpr.br/~laplima/ensino/pfec/concluidos/2010/autores.pdf>

Acesso em: 16/10/2012

DISPONIVEL EM: http://destacom.ufms.br/mediawiki/images/9/9f/Arduino_Destacom.pdf

Referências

Acesso em: 16/10/2012

DISPONIVE EM: <http://www.fisicaresolvida.com.br/2012/09>

Acesso em: 20/10/2012

DISPONIVEL EM: http://destacom.ufms.br/mediawiki/images/9/9f/Arduino_Destacom.pdf

Acesso em: 20/10/2012

DISPONIVEL EM: http://php.net/manual/pt_BR/history.php.php

Acesso em: 25/10/2012

DISPONIVEL EM: <http://arduino.cc/en/Reference/HomePage>

Acesso em: 25/10/2012

Assinaturas
dos estudantes:

Assinatura do
Orientador:

Data: