FACULDADE DE TECNOLOGIA DE AMERICANA
Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistema

Dionis Mendanha

Cubo de LED, Montagem e Programação

Americana, SP
2015

FACULDADE DE TECNOLOGIA DE AMERICANA

Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistema

Dionis Mendanha

dionis_bmx@hotmail.com

Cubo de LED, Montagem e Programação

Trabalho monográfico, desenvolvido em cumprimento à exigência curricular do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas da Fatec Americana, sob orientação do Prof. Esp. José William Pinto Gomes.

Área de concentração: ADS

Americana, SP

2015

FICHA CATALOGRÁFICA – Biblioteca Fatec Americana - CEETEPS
Dados Internacionais de Catalogação-na-fonte

Mendanha, Dionis
M488c Cubo de LED, montagem e programação. / Dionis Mendanha. – Americana: 2015.
44f.

Monografia (Graduação em Tecnologia em Análise e Desenvolvimento de Sistemas). - - Faculdade de Tecnologia de Americana – Centro Estadual de Educação Tecnológica Paula Souza.
Orientador: Prof. Esp. José William Pinto Gomes

1.Desenvolvimento de software 2. Dispositivos móveis – aplicativos 3. Microeletrônica I. Gomes, José William Pinto II. Centro Estadual de Educação Tecnológica Paula Souza – Faculdade de Tecnologia de Americana.

CDU: 681.3.05
681.519
621.3.049.77

Dionis Mendanha

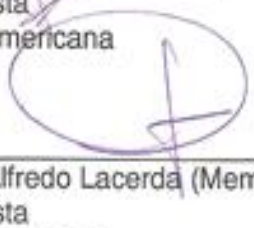
CUBO LED: MONTAGEM E PROGRAMAÇÃO

Trabalho de graduação apresentado como exigência parcial para obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas pelo CEETEPS/Faculdade de Tecnologia – Fatec/ Americana.
Área de concentração: Análise de sistemas.

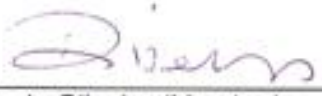
Americana, 24 de junho de 2015.

Banca Examinadora:

José William Pinto Gomes (Presidente)
Especialista
FATEC Americana



Antonio Alfredo Lacerda (Membro)
Especialista
FATEC Americana



Diógenes de Oliveira (Membro)
Mestre
FATEC Americana

AGRADECIMENTOS

Às pessoas que contribuíram para o desenvolvimento e conclusão do trabalho.

Ao professor orientador que me auxiliou a organizar as ideias para a elaboração deste trabalho.

Aos meus amigos que participaram ajudando com novas ideias e experiências com desenvolvimento de sistemas.

A minha esposa Edilaine Ferreira Andrade, pois sempre me incentivou a lutar para que esse e outros projetos fossem realizados.

DEDICATÓRIA

Dedico aos meus pais, pois sempre confiaram no meu potencial, me incentivaram a estudar, cursar uma faculdade e realizar meu sonho, e mesmo em meio às dificuldades, me ajudaram a chegar até aqui.

RESUMO

Quando falamos em trabalhar com informática, a maioria das pessoas pensam somente nos modos convencionais que temos hoje em dia, focado no desenvolvimento de software para computadores, aplicativos para celulares e desenvolvimento de aplicações web, mas a informática está sendo cada vez mais aplicada em conjunto com outras áreas a fim de facilitar os processos existentes nesses setores. Este trabalho apresenta uma outra abordagem da informática, onde foi implementada em conjunto com a eletrônica, a fim de elaborar um controle sobre um cubo de LED, sendo este inteiramente montado por componentes eletrônicos e controlado por um micro controlador chamado arduino. Para o seu desenvolvimento encontra-se descrito todos os passos necessários, desde a montagem do cubo até sua programação.

Palavras-chave: Arduino, Cubo, LED

ABSTRACT

When we talk about working with computers, most people only think in conventional ways we have today, focused on the development of computer software, mobile applications and web application development, but the computer is being increasingly applied in conjunction with other areas in order to facilitate existing processes in such sectors. This work shows another of computing approach, where it has been implemented in conjunction with electronics to develop a control over an LED cube, which is fully assembled for electronic components and controlled by a microcontroller Arduino called for development is described, all the steps required for its development , from the hub assembly to its programming.

Keywords: **Arduino, Cub, LED.**

LISTA DE ILUSTRAÇÕES

Lista de tabelas

SUMÁRIO

1 INTRODUÇÃO

Podemos definir eletrônica como um ramo da ciência onde se estuda o uso de circuitos formados por componentes elétricos e eletrônicos, seu principal objetivo é de representar, armazenar, transmitir ou processar informações além do controle destes processos (Benchimol, 1995). No início todos os problemas eram resolvidos por sistemas analógicos, onde uma quantidade é representada por um sinal elétrico proporcional ao valor da grandeza medida, por exemplo, a temperatura.

Com o passar dos anos e o avanço da tecnologia, esses problemas começaram a ser solucionados através da eletrônica digital, onde uma quantidade é representada por um arranjo de símbolos chamados dígitos utilizados em máquinas, tais como: computadores, sistemas de controle e automação, codificadores, decodificadores, entre outros. Sendo assim, pode-se afirmar que os circuitos internos dos computadores, os diversos tipos de sensores e os sistemas de telecomunicações estão dentro da área da Eletrônica.

Os componentes eletrônicos que compõem um sistema eletrônico são estruturas de um circuito que fazem parte de qualquer circuito elétrico, desde os mais simples até os mais complexos. Podemos atribuir o termo componente eletrônico a todo dispositivo que transmita corrente elétrica através de um condutor ou semicondutor.

Pode-se considerar um condutor elétrico todo material que faz com que as partículas eletrizadas se movimentem facilmente como por exemplo, os metais, mas devemos lembrar que existem condutores que se encontram nos estados líquido (soluções iônicas) e gasoso (gases ionizados). Com relação aos semicondutores podemos classificá-los como materiais que são um meio termo entre condutores e materiais isolantes sendo os mais utilizados na indústria eletrônica o Germânio, o Silício e o Diodo (Pedroso, 2008).

Um exemplo do uso de diodo é na fabricação de LED (*Light Emitting Diode*) que tem como função básica, a emissão de luz, sua utilização se torna bem viável

para substituir lâmpadas em diversos locais como painéis de avisos, semáforos e televisores, podendo até ser moldado em formas de estruturas geométricas.

Um exemplo disso é o chamado cubo de LED que combinado com um microcontrolador arduino[1] pode-se desenvolver vários tipos de animações. Este controlador é uma plataforma que possui um microprocessador de placa única com suporte de entrada/saída embutido e uma linguagem de programação padrão C/C++ (McRoberts, 2010).

Justificando a importância de estudar outras áreas onde a informática está sendo aplicada, o projeto utilizará um disfuncional biológico do olho humano conhecido como persistência da visão onde o olho armazena na retina uma quantidade de imagens exibidas rapidamente aparentando assim que somente uma imagem está sendo formada e os conceitos de programação em matriz tridimensional para que se consiga explorar todas as camadas do cubo maximizando a eficiência das animações apresentadas.

Como problema podemos citar a dificuldade em se elaborar uma estrutura correta para que os LEDs funcionem corretamente e o modo que sua programação deve ser feita para ter o máximo de eficiência. Para sanar estas questões devemos fazer as seguintes perguntas:

a) Como funciona a estrutura de um LED?
b) Como soldar os componentes?

Os objetivos específicos foram:

a) A programação utilizada pelo arduino.
b) Fazer um levantamento bibliográfico sobre a tecnologia arduino.
c) Como montar o cubo para que todos os LEDs trabalhem como um único circuito?

[1] http://www.arduino.cc - acessado em 03/04/2014

O objetivo geral do projeto é mostrar a pesquisa e o trabalho com outras áreas onde a informática pode ser aplicada saindo um pouco do que podemos classificar como “normal” sendo isto o desenvolvimento de aplicativos para celular, desenvolvimento de websites e software para gestão.

O trabalho foi estruturado em quatro capítulos, sendo o primeiro uma introdução geral do assunto, o segundo uma explicação de todo o conjunto necessário para elaborar o projeto, o terceiro apresenta um estudo de caso para comprovar que o projeto pode ser desenvolvido e o modo de fazê-lo e o quarto traz um aprofundamento geral sobre o arduino. Com base nas informações conseguidas a partir dos estudos realizados nos capítulos anteriores, o quinto capítulo se reserva às Considerações Finais.

2 CONCEITOS FUNDAMENTAIS

Neste capítulo serão abordados todos os conceitos básicos necessários para o entendimento deste trabalho.

2.1 ELETRÔNICA

Segundo Benchimol (1995) a evolução desta ciência foi lenta e acelerou com o passar do tempo, onde vários fatos e acontecimentos contribuíram para seu desenvolvimento, como por exemplo, a criação do dispositivo eletrônico mais antigo conhecido como célula fotovoltaica exibida na figura 1, a qual era capaz de converter luz diretamente em energia. Embora sua funcionalidade fosse meramente para curiosidade científica este fato, fez com que se alavancasse o interesse dos cientistas por este ramo.

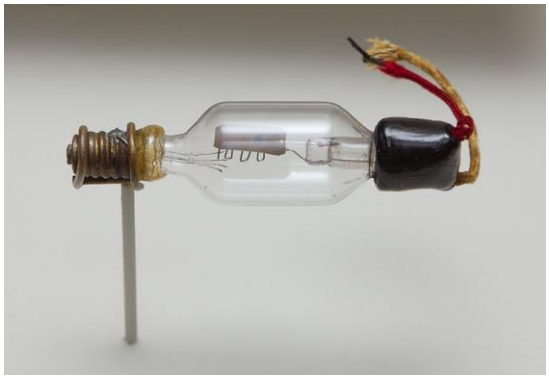


Figura 1 - Válvula termiônica amplificadora de áudio de 1906.

(Fonte: http://pt.wikipedia.org/wiki/V%C3%A1lvula_termi%C3%B4nica - acessado em 14/10/2014).

No século XX foi desenvolvido o primeiro transistor por uma empresa de telecomunicação chamada Bell Telephone, iniciando a era dos semicondutores. Logo após veio a construção do primeiro circuito integrado que era apenas uma montagem miniaturizada de componentes eletrônicos, feitos de uma só substância (germânio) assim, exponencialmente foi se desenvolvendo novas tecnologias e após os anos oitenta criou-se a Extra Larga Escala de Integração (ELSI), e com base

nesta tecnologia, desenvolveu-se os microprocessadores de alta velocidade e desempenho.

Benchimol (1995) define a eletrônica como um ramo que estuda a forma de controlar a energia elétrica, utilizando-se de circuitos formados por componentes elétricos e eletrônicos com o objetivo de representar, armazenar, transmitir ou processar informações, entre outros. Sendo assim, pode-se afirmar que os circuitos internos dos computadores, sistemas de telecomunicações, diversos tipos de sensores e transdutores estão dentro da área da eletrônica.

Esta ciência se divide em duas áreas: analógica e digital. Oioli (2001) apresenta uma boa analogia para explicar como estas duas áreas funcionam:

> “[...] o conceito das palavras Analógico e Digital, é compararmos uma rampa com uma escada. Ao analisarmos a rampa, percebemos que uma pessoa poderá ocupar cada uma das infinitas posições existentes entre o início e o fim. No caso da escada, a pessoa poderá estar em apenas um dos seus degraus. Sendo assim, podemos dizer que a rampa pode representar um sistema analógico, enquanto que a escada pode representar um sistema digital. ”

O grande avanço da eletrônica fez com que a utilização das técnicas digitais para implementar funções nos circuitos eletrônicos aumentasse isso proporcionou várias vantagens, como por exemplo, a facilidade em projetar os circuitos e armazenar as informações, com maior precisão e exatidão, além de operações programadas, entre outras.

Como o nosso mundo é quase totalmente analógico, para obter estas vantagens quando se trabalha com entradas e saídas analógicas devemos executar três passos importantes: primeiro devemos converter as entradas analógicas para o valor digital, depois realizar o processamento desta informação e por último converter a saída de volta ao formato analógico. Oioli (2001) afirma que estas conversões têm como base a técnica de numeração binária, sendo está o mais importante sistema de numeração em sistemas digitais.

A figura 2 mostra um sistema de controle de temperatura típico, onde seguindo o diagrama a temperatura analógica é medida e o valor obtido logo em seguida é convertida para digital que logo após ser processada é convertida novamente para o formato analógico, tendo no final um controlador que comanda alguma ação de ajuste na temperatura.

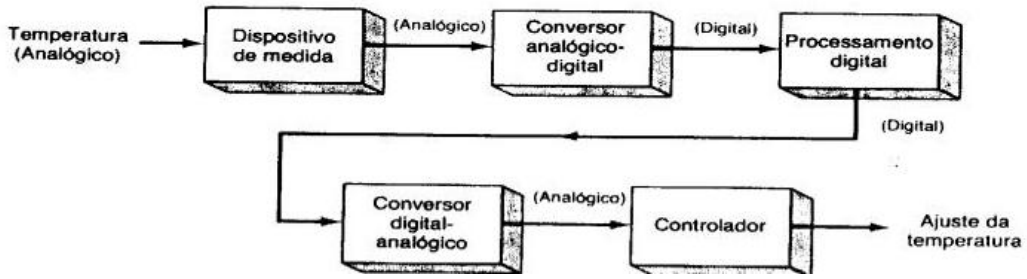


Figura 2 - Sistema de Controle de Temperatura com Conversões Analógico-Digitais.

(Fonte: http://pt.wikibooks.org/wiki/Eletr%C3%B4nica_Digital/Sistemas_anal%C3%B3gicos_e_digitais – acessado em 15/10/2014).

A eletrônica está sendo a base para a tecnologia moderna e sem ela os sistemas de controles que existem hoje não funcionariam, e já podemos vê-la sendo fundida com outras áreas como a pneumática, hidráulica e informática compondo os sistemas de analogia eletrônica para o nosso futuro.

2.2 SISTEMAS EMBARCADOS

Chase (2007) classifica um sistema como embarcado quando ele é dedicado a uma única tarefa que interage continuamente com o ambiente em sua volta por meio de atuadores (conversão digital/analógico, acionamento eletromecânico). Por exigir esta interação, o projetista deste tipo de sistema deve ter conhecimento em diversas áreas como programação, sistemas digitais, noções de controle de processos, sistemas de tempo real, tecnologia de aquisição de dados, cuidados com a estruturação do projeto e com o código produzido. A única flexibilidade desejada é que no caso de um *upgrade* o sistema possa ser reprogramado.

A denominação “sistema embarcado” (do inglês *Embedded Systems*) vem do fato destes sistemas serem desenvolvidos para possuírem uma fonte de energia fixa

(celulares, forno micro-ondas, automóveis, aparelhos de DVD, PALM's) tendo como características, a capacidade computacional e a sua independência de operação além de todos eles possuírem uma unidade de processamento e um circuito integrado, fixado a uma placa de circuito impresso, conforme alguns exemplos na figura 3.

Figura 3 - Lógica de um sistema embarcado usando um microprocessador como unidade de processamento.

(Fonte: Sistemas Embarcados, Otávio Chase, 2007).

Estes sistemas estão inseridos em milhares de dispositivos comuns que utilizamos no dia-a-dia. Cunha (2007) nos traz alguns tipos de aplicações para estes sistemas:

Aplicações Gerais: são aplicações que costumam ter uma grande interação com o usuário geralmente através de terminais de vídeo ou monitores como por exemplo, vídeo games, conversores de TV a cabo, caixas de banco, etc.

Aplicações de Controle: são aplicações que utilizam placas dedicadas e múltiplos sensores de entrada e saída, fornecendo pouca interação com o usuário, mostrando sinalizações através de LEDs. Utilizado nos motores de automóveis, processos químicos, controle de voo, etc.

Aplicações de processamento de sinais: são aplicações que envolvem um grande volume a ser processado em curto tempo, os sinais são digitalizados e processados e novamente convertido em sinais analógicos como no caso de tratamento de áudio, filtros, modens, radares, etc.

Aplicações de comunicação e redes: são aplicações de chaveamento e distribuição de informações, usadas em sistemas de telefonia, telecomunicações e internet.

Temos nestes sistemas dois modos de funcionamentos distintos que são determinantes para saber como se programar o dispositivo e como será seu comportamento na aplicação para qual foi desenhado.

Cunha (2007) nos explica estes dois modos:

> “[...] Reativo: o funcionamento se dá como resposta a eventos externos, que podem ser periódicos (caso de sistemas rotacionais ou de controles de *loop*) ou assíncronos (pressionamento de um botão por parte do usuário). Há, então, uma necessidade de entrada de dados para que aconteçam as ações de funcionamento. Geralmente não há limite de tempo para que os sinais de entrada sejam acionados, pois dependem da interação com o usuário ou com o processo ao qual é destinado. Porém, a saída, função do sinal de entrada, deve ser realizada exatamente após os sinais de entrada começarem a atuar. ”

> “[...] Controle em tempo real: existem limites de tempo para executar cada tarefa (leitura de sensor, emissão de sinais para um atuador, atualização de *display*, etc.). Por isso, nem sempre tempo real é igual ao modo mais rápido de executar uma tarefa. Estes modos de operação, por serem cíclicos, não dependem da entrada se sinais para executar as atividades, sendo capaz de tomar decisões referentes a ausência dos mesmos. ”

O controle em tempo real é divido em duas categorias conhecidas: como *Sotf Real Time* e *Hard Real Time*. Cunha (2007) explica suas diferenças:

> “[...] *Sotf Real Time* as tarefas podem ser executadas em um intervalo de tempo e não há consequência grave se este limite de tempo não for cumprido. ”

Como exemplo de *sotf real time* podemos citar um sistema bancário onde apenas uma mensagem de erro é exibida caso uma determinada tarefa não for realizada dentro do tempo determinado.

> “[...] *Hard Real Time* as tarefas devem ser executas em um tempo especifico com consequências graves se qualquer tarefa falhar. ”

Temos como exemplo de *hard real time* um sistema de controle de aviões, onde uma falha pode resultar em queda e perda de vidas.

Hoje estes sistemas se tornaram uma poderosa força predominante em vários setores da indústria eletrônica, pois torna possível a execução das tarefas de maneira simples.

2.3 PERSISTÊNCIA RETINIANA

Machado (2007) refere-se ao termo persistência retiniana como a capacidade que o olho humano tem de manter uma imagem na retina por cerca de 1/20 a 1/5 segundos. O responsável por definir esta capacidade satisfatoriamente foi Peter Mark Roget, em 1824, e mesmo este conceito sendo conhecido desde o Antigo Egito, por muito tempo foi considerado o fenômeno responsável pela síntese de movimento.

Para a percepção de luz e cor o olho humano utiliza basicamente uma lente e uma superfície fotossensível dentro de uma câmera, quando vemos um objeto sua imagem passa pela córnea, depois pela íris regulando a quantidade de luz e por

último chega ao cristalino que fica sobre a retina, e após impulsos eletroquímicos o cérebro interpreta esta imagem. A figura 4 mostra todas as etapas que este processo percorre.

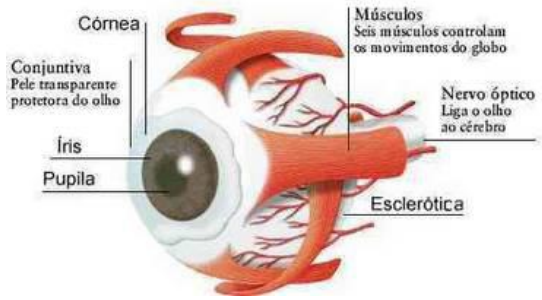


Figura 4 – Composição do olho humano.

(Fonte: https://www.passeidireto.com/arquivo/1874828/200902diufes---persistencia-retiniana).

Mesmo após o cérebro interpretar a imagem a retina continua enviando informações durante 1/10 segundos, então se uma imagem for trocada numa velocidade superior a esta, elas vão se fundir, provocando a sensação de movimento.

2.4 MICROCONTROLADORES

Segundo Chase (2007), microcontroladores são pequenos sistemas computacionais bastante poderosos que englobam em um único chip: interfaces de entrada/saída digitais e analógicas, periféricos importantes como a memória RAM, memória *FLASH*, interfaces de comunicação serial, conversores analógicos/digitais e temporizadores/controladores.

A vantagem de utilizar microcontroladores é que além de possuir os periféricos integrados em um único componente, eles são responsáveis por executar e armazenar os programas escritos para eles (*firmware*), assim como a capacidade de incorporar mais funções com a adição de periféricos, por meio de canais de comunicação como USB, porta PS/2, entre outros. Com a chegada dos

microcontroladores de 16 e 32 bits (atualmente o padrão é de 8 bits) a capacidade de gerar soluções mais complexas e com maior velocidade de processamento, irá crescer.

Os microcontroladores se diferenciam dos microprocessadores porque além de possuir os componentes lógicos e aritméticos usuais dos microprocessadores o microcontrolador, integra elementos adicionais em sua estrutura interna que já foram citados anteriormente. O projetista que desenvolve aplicações utilizando microcontroladores tem que lidar também com grandes desafios, fazendo muitas vezes todo o processo construtivo do aparelho: BIOS, *firmware* e circuitos.

2.5 A PLATAFORMA ARDUINO

O site oficial da arduino[2] classifica seu produto da seguinte maneira:

> "[...] O arduino é uma plataforma de prototipagem eletrônica *open-source* que se baseia em *hardware* e *software* flexíveis e fáceis de usar. É destinado a artistas, *designers*, *hobbistas* e qualquer pessoa interessada em criar objetos ou ambientes interativos. ".

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++. Sendo utilizado no desenvolvimento de objetos interativos, permitindo a entrada de sensores ou chaves, controlando uma variedade de luzes, motor ou saídas físicas, podendo assim manipular o *hardware* (sensores, portas, alarmes) através do *software*. A figura 5 mostra como é a placa do arduino Uno.

[2] http://playground.arduino.cc/Portugues/HomePage - acessado em 15/09/2014

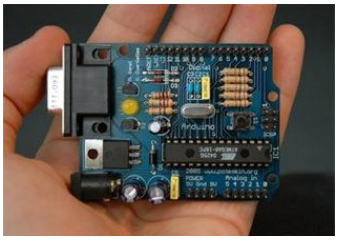


Figura 5 – Placa Arduino
(Fonte: http://www.arduinors.net/blog/arduino-placa-de-desenvolvimento-open-source/ - acessado 09/092014).

Com um kit de desenvolvimento, capaz de interpretar variáveis no ambiente transformando-as em sinais elétricos correspondentes e através de sensores ligados aos seus terminais de entrada, ele pode atuar no controle ou acionamento de algum outro elemento conectado ao terminal de saída. O arduino se torna uma ferramenta de controle de entrada e saída de dados, podendo ser acionada por um sensor. Podemos compará-lo como um computador, que tem como sensores de entrada: o mouse e o teclado e, de saída - impressoras e caixas de som.

McRoberts(2010) afirma que podemos utilizar o arduino em projetos visando a interatividade, já que ele pode ser conectado a um computador, a uma rede, ou até mesmo a Internet para recuperar e enviar dados e atuar sobre eles. Em outras palavras, ele pode enviar um conjunto de dados recebidos de alguns sensores para um site, dados estes, que poderão assim, ser exibidos na forma de um gráfico.

O arduino é mais uma das ferramentas criadas para atender a área educacional, mas devido ao seu aprimoramento e às suas derivações, acaba sendo mais utilizado para outras finalidades.

2.5.1 História do Arduino

Seu projeto teve início em 2005. Um italiano chamado Massimo Banzi queria que seus alunos de design utilizassem a eletrônica e a programação de computadores em seus projetos de interatividade robótica. O objetivo inicial era desenvolver uma ferramenta de baixo custo e de fácil manuseio, pois eram estas as maiores limitações que professores, pesquisadores e alunos encontravam para iniciar atividades relacionadas à robótica.

O site hecomecatronica (2014), exemplifica que atualmente o arduino original é fabricado pela companhia italiana Smart Projects. Porém a SparkFun Electronics também possui algumas marcas comerciais sob a mesma licença, mas há uma proteção do nome arduino, reservando o uso apenas para a fábrica oficial italiana, porém a fabricação do dispositivo, sem uso do nome oficial arduino, por outras empresas é permitido.

Por possuir licença totalmente *open-source* qualquer usuário pode fabricar seu próprio dispositivo arduino, no entanto os idealizadores do projeto possuem um serviço de venda do produto pronto, através deles próprios e também por distribuidores oficiais com pontos de venda mundiais. O núcleo da equipe arduino é composto por David Cuartielles, Gianluca Martino, Tom Igoe, David Mellis, and Massimo Banzi.

Segundo Souza (2008) o arduino possui o código aberto e existem empresas de desenvolvimento de *hardware* que optam em criar suas próprias versões como, por exemplo, a Tato responsável pelo Tatuino sendo esta 100% compatível com o arduino. Existem também o Severino, Freeduino entre outros.

3 ESTUDO DE CASO – CUBO DE LED

Este capítulo discorre sobre o desenvolvimento do cubo de LED, mostrando toda parte da sua criação até a programação utilizada.

3.1 O PROJETO

O projeto consiste no desenvolvimento de um cubo 8x8x8, uma matriz de LED que será controlada pelo micro controlador arduino, onde, foram aplicados conceitos de programação em matriz para acender os LEDs de várias maneiras, criando assim, diversos tipos de animações.

O processo deve ser feito com muito cuidado, pois qualquer erro acarretará no mau funcionamento dos LEDs prejudicando os efeitos. Outro cuidado que se deve ter antes de soldá-los, é testar os componentes para certificar que todos estejam funcionando corretamente.

Por se tratar de um cubo 3D a quantidade de animações que podem ser programadas são enormes, podemos escolher entre animações aleatórias, formas geométricas, palavras, entre outras. Para auxiliar no desenvolvimento destas animações podemos contar com bibliotecas do próprio arduino e algumas desenvolvidas por pessoas que já construíram o cubo e colocaram na internet para download.

Para o desenvolvimento foi utilizado os seguintes componentes:

- Capacitores TM1818, APM4953, HT1238 smd
- Resistores 471, 103, 102
- Barra terminal 90º macho

- Barra terminal 90º macho duplo
- Barra 12 vias fêmea 90º
- Jack P4
- 512 led azuis
- Placa dedicada
- Microcontrolador

3.2 MONTAGEM DO CUBO

Para a montagem do cubo é necessário ter conhecimento em eletrônica para soldar todos os componentes corretamente e fazer uma solda de qualidade.

O modo mais simples de realizar a construção do cubo é dividindo-o em camadas, cada uma irá conter 64 LEDs enfileirados horizontalmente, formando 8 linhas com 8 LEDs cada. Os terminais de catodo dos LEDs serão todos soldados uns nos outros de modo que haja somente um catodo no final. Para manter todos os LEDs alinhados foi usada uma plataforma de madeira, como molde atingindo a simetria do cubo entre cada LED, como mostra a figura 6.

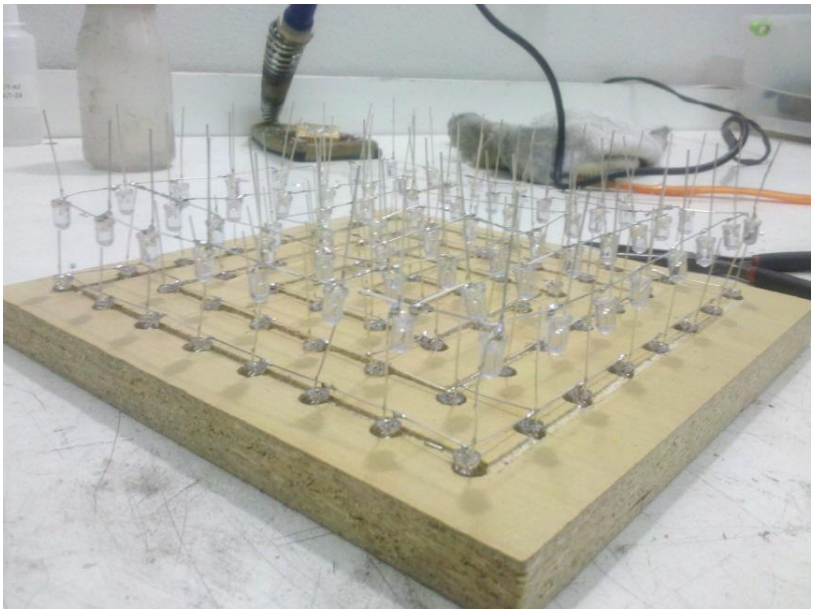


Figura 6 - Base para montar as camadas do cubo.

(Fonte: Autoria Própria).

Com as 8 camadas prontas, começa-se a erguer o cubo, onde uma camada é soldada na outra através dos terminais de anodo dos LEDs. No final deve-se ter algo parecido com a figura 7.

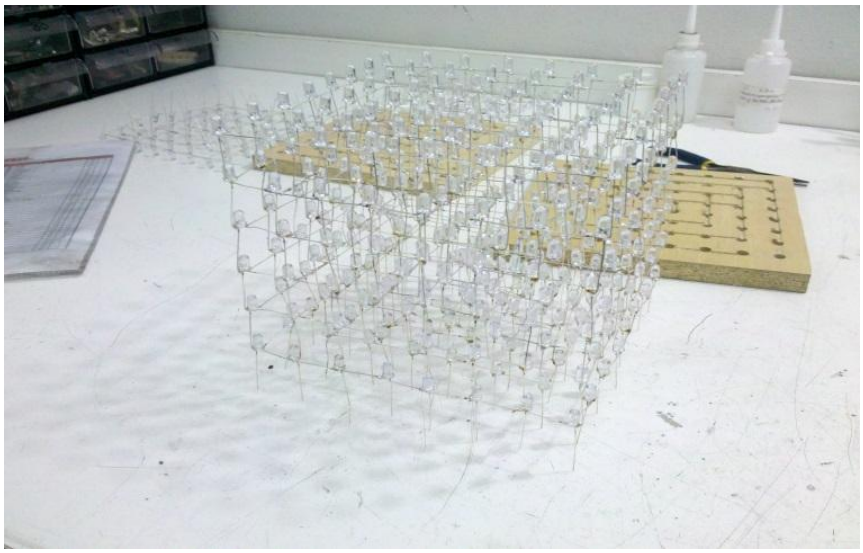


Figura 7 - Cubo com todas as camadas soldadas.

(Fonte: Autoria Própria).

Após todas as camadas estarem soldadas devemos soldar os componentes na placa, seguindo o *datasheet*. Um dos componentes usados é outro microcontrolador que fará o trabalho mais importante que será realizar a comunicação do arduino com a placa, pois a mesma foi desenvolvida para que as animações fossem passadas através de um *software* diretamente do computador. A figura 8 exemplifica como deve ficar a placa depois de finalizada.

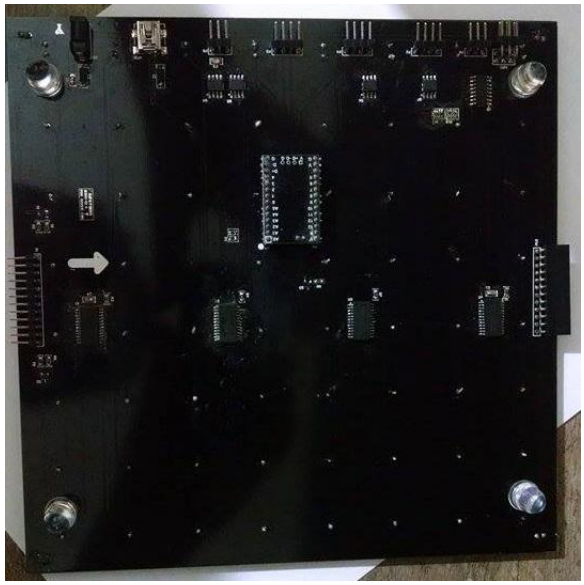


Figura 8 - Placa com os componentes soldados.

(Fonte: Autoria Própria).

O último passo é soldar as camadas de LED na placa, finalizando assim, a construção do cubo que deverá ficar como na figura 9.

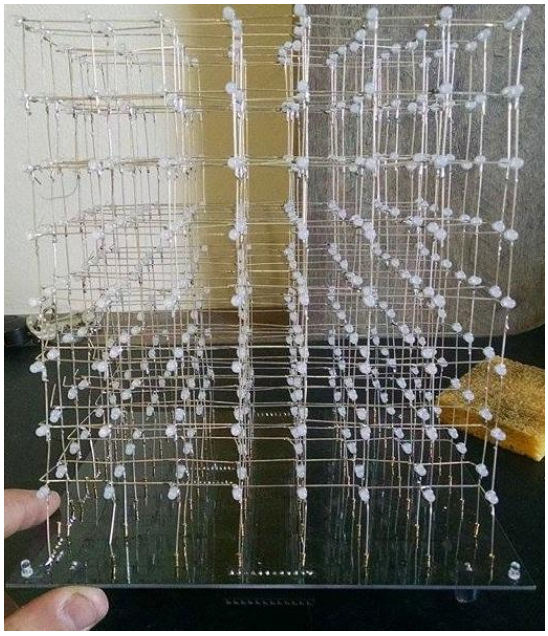


Figura 9 - Montagem do cubo completa.

(Fonte: Autoria Própria).

Conforme citado a placa possui um microcontrolador que será usado como intermediário para a comunicação do arduino com a placa. Para isso foi utilizado uma das entradas P2 disponíveis na placa, cada pino desta entrada corresponde a uma entrada do arduino sendo exatamente nesta ordem: GND, RX, TX e 5V. Após a conexão destas entradas o projeto deverá estar como mostrado na figura 10.

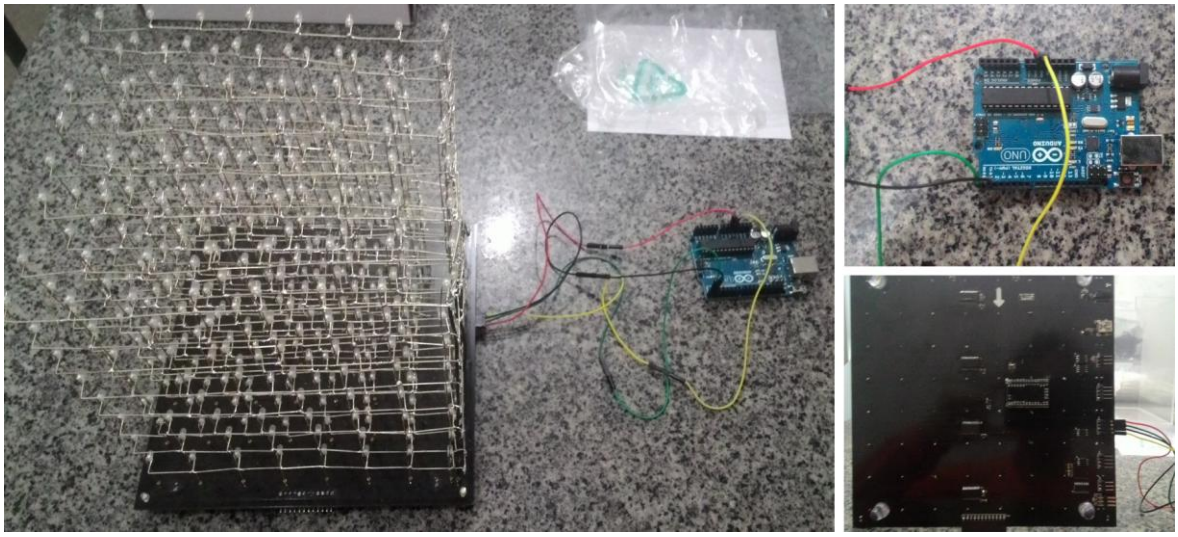


Figura 10 - Ligação do Arduino com a Placa.
(Fonte: Autoria Própria).

Na figura 10 vemos como as conexões devem ser feitas. Cada fio está ligado a um pino e a sua respectiva entrada no arduino, seguindo a ordem citada anteriormente:

- Fio vermelho conectado ao GND;
- Fio verde conectado a entrada RX;
- Fio preto conectado a entrada TX;
- Fio amarelo conectado a entrada 5V;

Após finalizar toda a parte de montagem do cubo é interessante que se faça uma caixa de acrílico para guardá-lo, já que sua estrutura é bem frágil. Abaixo haverá uma explicação mais detalhada de cada entrada e de como devemos utilizá-las. Depois desse procedimento, basta conectar o USB do arduino no computador e começar a programar seu cubo.

3.3 PROGRAMANDO O CUBO

Como citado anteriormente ao se programar no ambiente do arduino utilizamos suas funções principais o `setup()` onde colocamos as funções das animações e o `loop()` onde declaramos as funções para que elas se repitam fazendo assim que o arduino exiba nossas animações.

Ao se pensar nos tipos e quantidades de animações que podemos criar podemos ir bem longe, já que tudo irá depender da sua imaginação para criar animações aleatórias e/ou conhecimento para trabalhar com desenhos 3D.

As animações foram desenvolvidas para fins educativos, procuramos focar em animações aleatórias. Utilizando a biblioteca `Controller_A3D8_Basic` em conjunto com a placa em que o cubo foi montado facilita o desenvolvimento, pois ela já possui alguns efeitos que podemos utilizar como base e implementar novas animações.

Para utilizar a biblioteca citada anteriormente devemos inclui-la no projeto, para isso usamos a *tag* include desta maneira: `#include "Controller_A3D8_Basic.h".` Logo abaixo declaramos as variáveis de ambiente necessárias para utilizar a biblioteca e devemos instanciar um objeto para usar suas funções ficando desta maneira `Controller_A3D8_Basic cube (Serial)`.

Ao se programar utilizando o conceito de orientação a objetos dentro da função `setup ()` temos que informar para a biblioteca todos os parâmetros necessários para que ela funcione corretamente. Esses, parâmetros mudam, de acordo com a biblioteca que você escolher para trabalhar.

Depois de informar todos os parâmetros essenciais podemos começar a desenvolver as animações, que podem ser feitas em blocos de funções as quais devem ser sempre iniciadas pelo tipo `void` e um identificador como por exemplo: `void animationBlockScan () {}`, `void animationRiseZ () {}` e dentro de cada bloco desenvolvemos a lógica necessária para que as animações aconteçam como exemplificado logo abaixo:

```
void animationFacetScan()
   {
     static byte value = 0x01;
     cube.sendGlobal(value);
     value <<= 1;
     if (value == 0x00)
          value = 0x01;
   }
```

Finalmente, após todas as animações desejadas estarem criadas devemos chamá-las dentro da função `loop ()`, que fará o trabalho de repetir todas as animações enquanto o cubo estiver ligado.

4 CUBO DE LED UTILIZANDO ARDUINO

Este capítulo aborda a parte técnica do arduino. Uno junto com uma explicação básica de como programar utilizando a linguagem *Wiring*.

4.1 ESPECIFICAÇÕES TÉCNICAS DO ARDUINO UNO

Na tabela 1 temos a descrição técnica da placa arduino uno:

Chip Controlador	ATmega328
Tensão de operação	5V
Tensão Entrada	7-12V
Limite de tensão	6-20V
Pinos Digitais de E/S	14 (06 ajustáveis em PWM)
Entradas analógicas	6 pinos
DC Corrente E/S	40 mA
DC Corrente 3.3V	50 mA
Memória Flash	32 KB (0.5 KB reservado)
SRAM	2 KB
EEPROM	1 KB
Clock	16MHz
Dimensões	68x55x10mm
Peso	26g
Furos	4
Conectores	USB e Power Jack

Tabela 1 – Especificações Técnicas Arduino Uno.

(Fonte: http://arduino.cc/en/Main/arduinoBoardUno - acessado em 11/09/2014)

Na figura 11 mostra a interface do arduino UNO, que mesmo sendo simples é muito poderosa.

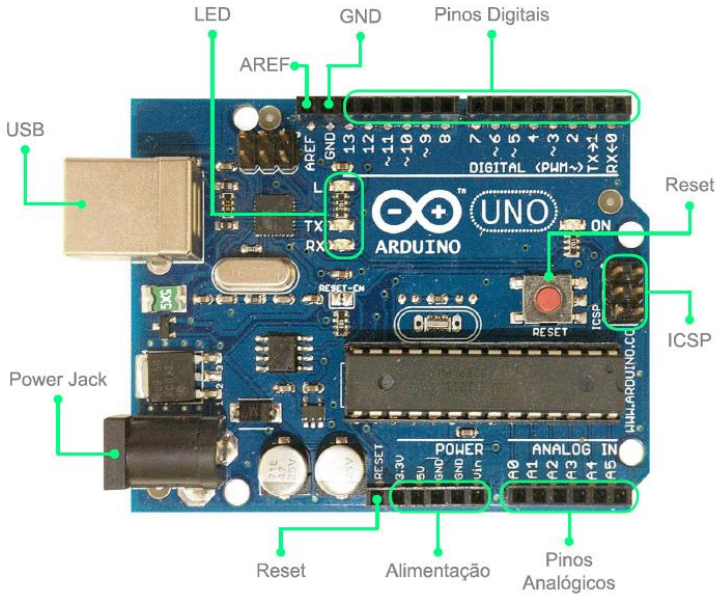


Figura 11 – Componentes da placa Arduino UNO.

(Fonte: http://www.hecomecatronica.com.br/index.php/downloads/manuais-e-documentacao/22-arduino-uno-guia-do-usuarios - acessado 28/09/2014).

A figura 11 detalha todas as partes em que se divide a placa do arduino UNO, mostrado todos os conectores que a ela possui. O site Erusbot (2012) explica que eles estão organizando-os da seguinte forma:

- 14 pinos de entra e saída digital (pinos 0-13): esses pinos podem ser utilizados como entradas ou saídas digitais de acordo com a necessidade do projeto e conforme foi definido no sketch criado na IDE.
- Pinos de entradas analógicas (pinos A0 – A5): esses pinos são dedicados a receber valores analógicos. O valor a ser lido deve estar na faixa de 0 a 5 V onde serão convertidos para valores entre 0 e 1023.
- Pinos de saídas analógicas (pinos 3, 5, 6, 9, 10 e 11): são pinos digitais que podem ser programados para ser utilizados como saídas analógicas, utilizando modulação PWM.

4.2 POWER JACK - FONTE DE ALIMENTAÇÃO

A principal função da fonte de alimentação é energizar a CPU e os demais componentes da placa para que estes funcionem corretamente. Esta energia deve conter uma tensão mínima de 7 volts e máxima de 35, tendo também uma corrente de 300mA (Site Erusbot, 2012).

4.3 NÚCLEO DA CPU

O microcontrolador pode ser comparado a um computador completo por possuir uma CPU, com memória RAM, memória ROM, uma unidade de processamento e os dispositivos de entrada e saída.

Os desenvolvedores do arduino optaram por utilizar os microcontroladores da empresa ATMEL[3] mais especificamente o Atmega. Podemos encontrar placas com diversos modelos sendo mais comuns os que utilizam ATMEGA8, ATMEGA16, ATMEGA328P estes se diferenciando pela quantidade de memória e configurações dos módulos de entrada e saída (Site Erusbot, 2012).

[3] http://www.atmel.com/pt/br/ - acessado em 29/09/2014

4.4 ENTRADAS E SAÍDAS

Para tratar deste tópico pode-se basear no chip mais básico o ATMEGA8 exibido na figura 11. Ele possui um total de 28 pinos para conexões, ficando 14 de cada lado é através deles que acessamos as funções do microcontrolador.

Figura 12 – Microcontrolador ATMega8.

(Fonte: http://cal-eng.com/?wpsc-product=atmega8_8mhz – acessado 02/10/2014).

Os pinos que encontramos no arduino são divididos da seguinte maneira:

- 14 pinos digitais de entrada ou saída (programáveis)
- Pinos de entrada analógica ou entrada/saída digital (programáveis)
- Pinos de alimentação (GND, 5V, referência analógica)
- 1 pino de reset
- 2 pinos para conectar o cristal oscilador

Podemos destacar os 20 pinos que ficam disponíveis para o usuário utilizar (14 digitais e 6 analógicos todos programáveis) já que é através destes que o arduino se conecta com a eletrônica externa. Como estes pinos possuem mais de uma função o programador deve escolher o que o pino irá fazer quando escrevem um programa para sua placa. Eles ficam expostos ao usuário através de conectores fêmeas onde se encaixam os conectores para construir o circuito externo da placa arduino (Site Erusbot, 2012).

4.5 ENTRADAS DIGITAIS

Quando programamos um pino para trabalhar como uma entrada digital, o comando que utilizamos efetua a leitura da tensão que está sendo aplicada e após a execução deste comando sabemos se ele está no estado “alto” ou “baixo“. A função de entrada digital somente defini os valores 0 e 1 não podemos saber quanta tensão está sendo aplicada no pino, para isso usamos entrada analógicas (JACEE, 2012).

4.6 ENTRADAS ANALÓGICAS

Temos 6 entradas, pinos deste tipo disponíveis, na placa arduino é nestas entradas que utilizamos sensores para converter grandezas físicas em um valor de tensão, ao contrário de uma entrada digital, que nos informa somente se a tensão existe ou não, medindo a quantidade de tensão que está sendo aplicada (Site Erusbot, 2012).

4.7 SAÍDAS DIGITAIS

Conseguimos programar o arduino para que ele tenha até 20 saídas (pinos) digitais, mas podemos programar uma única saída para controlar um bloco de pinos. É por meio destas saídas conseguimos acender LEDs, ligar um motor entre outras coisas (Site Erusbot, 2012).

4.8 PINOS ESPECIAIS

O site Erusbot (2012), apresenta que encontramos no arduino alguns pinos com características especiais, onde as funções que eles irão executar são feitas através da programação. São eles:

PVM: é considerado uma saída analógica, mas na verdade trata-se de uma saída digital gerando sinais alternados (0 e 1) e quando está ligado (1) é utilizado para controlar ou gerar tensões controladas pelo programa. **Pinos 3, 5, 6, 9, 10 e 11**.

Portal Serial *Usart*: estes pinos são responsáveis pela comunicação serial assíncrona. Assim podemos conectar um dispositivo *bluetooth* e nos comunicarmos com o arduino através deles. **Pinos 0 (RX recebe dados) e pino 1 (TX envia dados).**

Comparadores analógicos: usados para comparar tensões sem precisar de um programa para obtê-las já que o próprio hardware se encarrega desta tarefa. **Pinos 6 e 7**.

Interrupção Externa: podem ser programados para avisar sobre uma mudança que deve ocorrer no programa, utilizando-se de algum componente externo (botões). **Pinos 2 e 3**.

Porta SPI: estes pinos são responsáveis pela comunicação serial síncrona, sendo bem mais rápidos que os *Usart*. Nela podemos conectar cartões de memórias, dentre outras coisas. **Pinos 10, 11, 12 e 13.**

4.9 FIRMWARE

O firmware nada mais é do que um software localizado dentro da memória do microcontrolador. Sendo uma combinação de uma memória ROM usada para leitura e um programa que fica gravado nesta memória (Site Erusbot, 2012).

4.10 SOFTWARE

Neste capítulo iremos abordar o ambiente de desenvolvimento do arduino, mostrando algumas funcionalidades e explicando sua interface.

O site Erusbot (2012) explica que o ambiente de desenvolvimento do arduino é um compilador que usa uma interface gráfica do Java. Este software basicamente é um programa que utiliza um IDE simples de utilizar e entender suas bibliotecas que são facilmente encontradas. Sua principal função é o desenvolvimento do software e enviá-lo a placa para que ele possa ser executado.

A linguagem de programação utilizada no arduino e modelada a partir da linguagem *Wiring* e quando pressionamos o botão de upload da IDE o código escrito é traduzido para linguagem C e transmitido para o compilador que novamente traduz os comandos para uma linguagem que é compreendida pelo microcontrolador.

A figura 13 mostra o ambiente de programação da placa arduino que tem as seguintes opções:

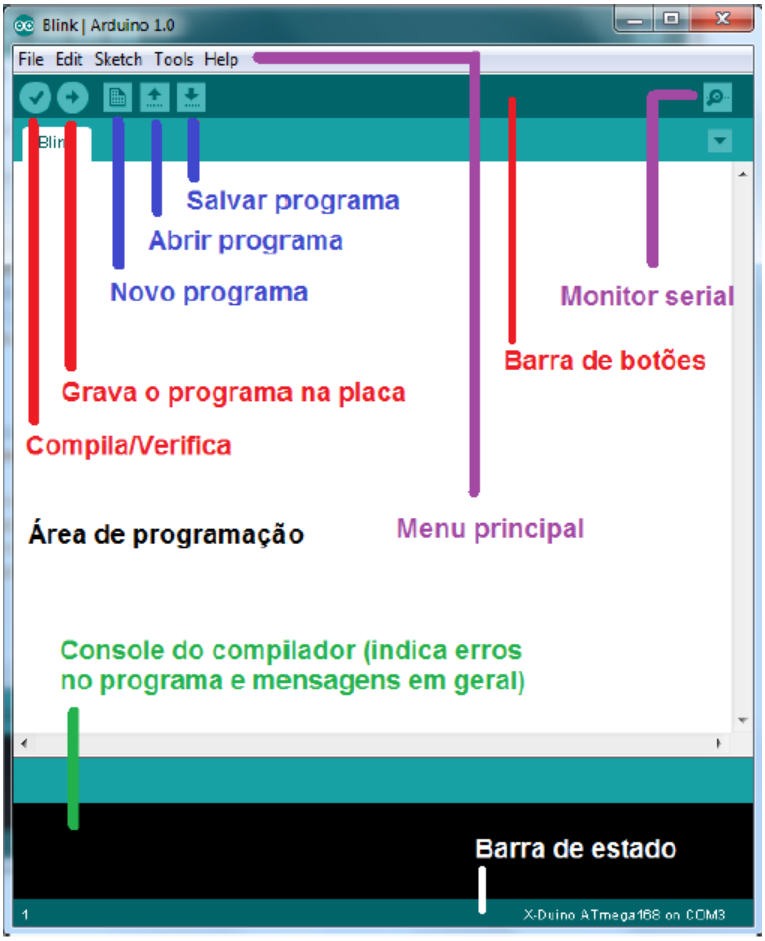


Figura 13 – Ambiente de Programação placa Arduino.

(Fonte: http://www.inf.ufes.br/~erus/arquivos/ERUS_minicurso%20arduino.pdf – acessado 12/10/2014).

Para começar a desenvolver seu programa, basta acessar o site oficial do arduino (www.arduino.cc) e baixar a IDE de acordo com seu sistema operacional. Ao conectar o arduino ao computador pela primeira vez será atribuída uma porta serial COM para comunicação e selecionar qual modelo de placa estaremos usando.

4.11 ESTRUTURA DO PROGRAMA

Segundo o site Erusbot (2012), quando se codifica um programa para o arduino devemos nos atentar a duas funções principais o `setup()` e o `loop()`

A função `setup()` serve para inicializar a placa e o programa, ela é executada uma única vez quando a placa é ligada ou resetada, informando quais componentes do arduino serão utilizados pelo programa.

A função `loop()` pode ser comparada com a função `main` da linguagem C, pois é dentro dela que colocamos a lógica que o programa deve ficar executando indefinidamente até que a placa seja desligada ou o botão de *reset* seja pressionado. Podemos ver um exemplo destas funções na figura 14.

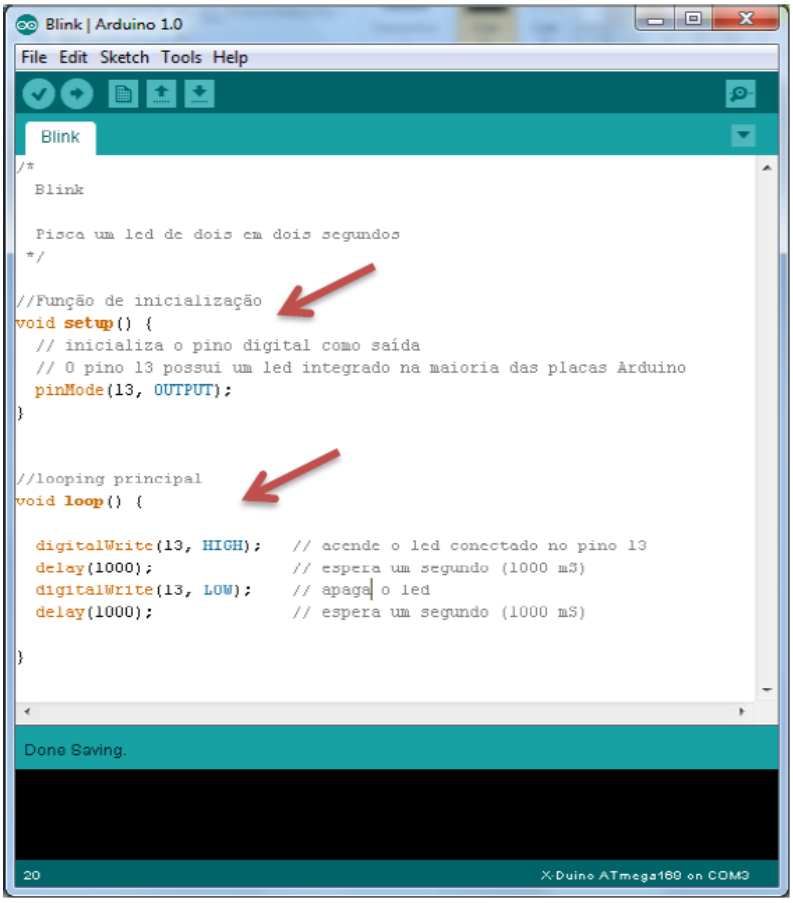


Figura 14 – Estrutura de um programa em Arduino Arduino.

(Fonte: http://www.inf.ufes.br/~erus/arquivos/ERUS_minicurso%20arduino.pdf – acessado 12/10/2014).

Encontramos na figura 14 a estrutura de um programa arduino onde está sendo utilizadas as funções citadas anteriormente. A função `setup()` está definindo o pino digital 13 como saída através do comando `pinMode(13, OUTPUT)` na função `loop()` temos os comandos que farão o controle deste pino, o comando `digitalWrite` envia uma tensão *HIGH*, equivalente a 5 volts, fazendo com que o LED acenda o comando `delay` é simplesmente o tempo que o programa deve

aguardar para executar a próxima instrução que é outro comando *digitalWrite* enviando uma tensão *LOW*, equivalente 0 volts, fazendo com que o LED apague. Como a função *loop()* funciona indefinidamente o LED ficará piscando (Site Erusbot, 2012).

4.12 PRINCIPAIS COMANDOS PARA O ARDUINO

A linguagem de programação utilizada no arduino é derivada da linguagem C, então sua estrutura de controle (*if*, *else*, *while*, *for* ...), seus elementos de sintaxe (#define,#include), operadores aritméticos (+, -, *, /) e operadores de lógicos (==, !=, <, > ...) são utilizados nos programas para arduino.

Abaixo encontramos algumas das principais funções para programar em arduino:

- *pinMode (pin, mode):* Configura o pino especificado para que se comporte como entrada ou saída, sendo *Pin* = número do pino e *mode* = *INPUT* ou *OUTPUT*.

- *digitalWrite (pin,value):* escreve um valor *HIGH* ou *LOW* em um pino digital. Se o pino foi configurado como saída sua voltagem será determinada ao valor correspondente: 5V para *HIGH* e 0V para *LOW*. Se o pino estiver configurado como entrada escrever um *HIGH* levantará o resistor interno de 20kΩ. Escrever um *LOW* rebaixará o resistor. Obviamente *pin* = número do pino e *value* = *HIGH* ou *LOW*.

- *digitalRead (pin):* Lê o valor de um pino digital especificado, *HIGH* ou *LOW*. *Pin* = número do pino. Retorna *HIGH* ou *LOW*.

- *analogRead (pin):* Lê o valor de um pino analógico especificado. Pode mapear voltagens entre 0 a 5v, sendo 4,9mV por unidade.
- *analogWrite (pin, value):* Escreve um valor analógico (onda PWM, explicaremos mais abaixo). Pode ser utilizada para acender um LED variando o brilho ou girar um motor a velocidade variável. Após realizar essa função o pino vai gerar uma onda quadrada estável com ciclo de rendimento especificado até que o próximo *analogWrite()* seja realizado (ou que seja realizado um *digitalRead()* ou *digitalWrite()* no mesmo pino). Podemos encontrar este e mais comandos no guia de referência do arduino http://arduino.cc/en/Reference/HomePage.

4.13 PROGRAMANDO EM ARDUINO PASSO A PASSO

Para explicar como se programa em arduino sem utilizar uma biblioteca, será utilizado como exemplo algo bem simples, como acender e apagar dois LEDs, sendo este um exemplo básico, mas que com tempo e empenho pode-se se criar animações incríveis.

Para começar, devemos primeiro criar uma variável chamada *ledPin* que receberá o número do pino onde o LED está conectado. Neste exemplo o pino nove e logo abaixo mostraremos mais uma variável com o nome de *numero* que irá funcionar como um controlador das ações de acender e apagar o LED. Neste exemplo, para controlar as ações devemos iniciar a variável com um número negativo qualquer. Enquanto esta variável não for maior que zero, nada será executado.

Após criarmos e iniciarmos nossas variáveis, temos que criar a função *setup()* e especificar como iremos trabalhar, ou seja, falaremos que nossa variável *ledPin* é uma variável saída através do comando *pinMode* e como estamos

usando serial, iniciaremos especificando sua velocidade, sendo que esta deve estar configurada no serial monitor.

Com a função `setup()` finalizada devemos criar agora a função `loop()` onde desenvolveremos a lógica para que o LED acenda e apague. Esta lógica iniciará com uma condição na qual verificamos se o valor do serial é maior que zero e caso a condição seja verdadeira passaremos este valor para a variável `numero` usando o comando `Serial.read`. Caso você queira verificar qual o valor que o serial está recebendo, deve utilizar o comando `Serial.print` passando como parâmetro a variável `numero` já que ela armazena o valor do serial.

O próximo passo é verificar se a variável `numero` é maior que zero e dentro desta verificação faremos mais duas, uma para saber se o valor `numero` é igual a 1, sendo verdadeira esta condição, mandaremos um sinal para o LED acender com o comando `digitalWrite()` passando como parâmetro o pino que o LED se encontra e o valor *HIGH*. A outra verificação a ser feita é para sabermos se o valor de `numero` é igual a 0 e caso seja verdadeiro, mandaremos um sinal para o LED apagar utilizando o mesmo comando `digitalWrite()` passando seus parâmetros mas ao invés de utilizar o valor *HIGH* usaremos o valor *LOW* que faz com que o LED apague. Após estas verificações, devemos utilizar o comando `delay` para que a ação de acender e apagar tenha um intervalo. Este intervalo fica a seu critério neste exemplo, utilizamos um intervalo de 2 segundos.

Ao final da função `loop()` voltamos a variável `numero` para seu valor original no caso -1. Para realizar o teste usaremos a tela de monitor serial, onde mandaremos os valores para que o arduino interprete e execute a animação. A figura 15 mostra como o código ficará no final.

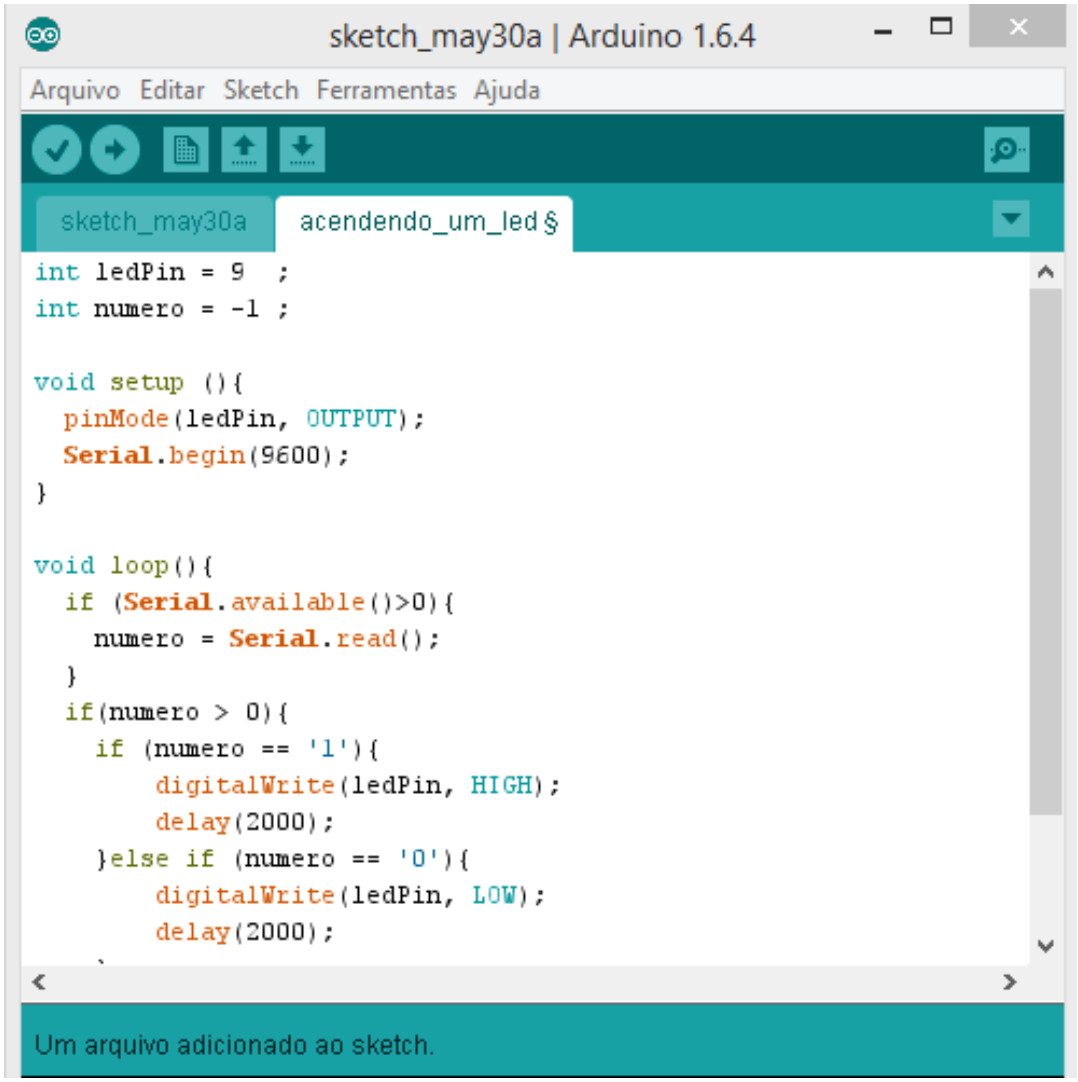


Figura 15 – Acendendo um LED utilizando Serial.

(Fonte: Autoria própria)

5 CONSIDERAÇÕES FINAIS

Quando se refere a desenvolvimento de sistemas, a maioria das pessoas imaginam algo relacionado a computadores, internet e celulares pois desconhecem o quanto a informática está se integrando com outras áreas, fazendo, assim, com que novas tecnologias surjam todos os dias, visando sempre facilitar o trabalho das pessoas.

O arduino, apesar de não ser tão recente, ainda é pouco conhecido pelas pessoas do nosso meio. Assim, o desafio de mostrar e aprender sobre algo diferente e sair das escolhas convencionais da nossa área nos levaram a escolha do tema.

Este trabalho mostrou que quando saímos do convencional encontramos uma quantidade enorme de áreas que podemos trabalhar junto com a informática e os resultados obtidos são surpreendentes. Poderíamos citar vários projetos em que o arduino se enquadra, no entanto, nos limitaremos ao que foi proposto no trabalho como exemplo de sucesso.

Com relação aos objetivos citados na introdução, algumas dificuldades foram encontradas:

- A falta de conhecimento em eletrônica e em solda.
- Dificuldade em encontrar material em português sobre o assunto.
- Falta de habilidade e equipamentos para trabalhar com soldas para componentes pequenos

Mas com empenho, dedicação, ajuda do orientador e amigos, tudo foi se resolvendo tornando o projeto prazeroso e atrativo, atingindo todos os objetivos e expectativas.

A conclusão deste trabalho é que o arduino se tornou uma poderosa ferramenta na criação de sistemas embarcados, dos simples ao mais complexo, pois

com ele consegue-se um controle preciso de suas funções dos sistemas e confiança em seus resultados.

A pesquisa realizada mostra uma visão geral do que é o arduino e como ele trabalha, sendo útil para as pessoas que estiverem iniciando nessa área, já que sua didática e o exemplo utilizado não são de grandes dificuldades e aborda todo o conhecimento básico necessário para seguir em frente com projetos maiores.

Para aqueles que aceitam desafios, é possível a continuidade deste trabalho visando o desenvolvimento de animações em 3D, criando um sistema mais interessante.

Como trabalho futuro iremos aprofundar em conhecimentos referente a eletrônica, a fim de sanar as dificuldades encontradas durante o desenvolvimento do estudo de caso, que posteriormente facilitará na criação de outros projetos utilizando o arduino, já que ainda há muito a ser explorado.

REFERÊNCIAS

BENCHIMOL, Augusto. Uma Breve História da Eletrônica. 1ª Edição. Rio de Janeiro: Interciência, 1995. 167.

MCROBERTS, Michael. Arduino Básico.1ª Edição. São Paulo: Novatec, 2011. 456.

OIOLI, Frederico de Campos, Elementos Básicos da Eletrônica Digital. Santo Andre: ETE Júlio de Mesquita, 1997. 90 f. Apostila.

CHASE, Octavio. Sistemas Embarcados. Sbajovem. Disponível em www.lyfreitas.com.br/ant/pdf/Embarcados.pdf. Acesso em 15 de out de 2014

CUNHA, A.F. O que são sistemas embarcados?. Saber Eletrônica, v. 414, p. 39-43, 2007.

SOUZA, Fábio. Arduino – Primeiros Passos. Disponível em http://www.embarcados.com.br/arduino-primeiros-passos. Acessado em 10/02/2015

HECOMECATRONICA. Arduino Uno Guia do Usuário. DIsponivel em http://www.hecomecatronica.com.br/artigos%20e%20Documentos/Artigos%20Tecnicos/Documentos/01%20arduino%20uno%20-%20guia%20do%20usuario.pdf. Acessado em 13/02/2015

Erusbot. Minicurso Arduino. Disponível em http://www.erusbot.com.br/arquivos/materiais/Minicurso%20Arduino%20JACEE%202012.pdf. Acessado em 15/02/2015

MACHADO, Arlindo. Prê-cinemas & Pós-cinemas, Editora Papirus, Campinas, 4ª Edição, 2007.