

UNIVERSIDADE ESTADUAL DO SUDOESTE DA BAHIA – UESB
DEPARTAMENTO DE CIÊNCIAS EXATAS – DCE
COLEGIADO DE CIÊNCIA DA COMPUTAÇÃO

JOSEMAR BRAGA LUZ

AUTOMAÇÃO RESIDENCIAL COM STARGATE-888 VIA PORTA USB

VITÓRIA DA CONQUISTA, 2013

JOSEMAR BRAGA LUZ

AUTOMAÇÃO RESIDENCIAL COM STARGATE-888 VIA PORTA USB

Monografia, apresentado ao Curso de Ciência da Computação da Universidade Estadual do Sudoeste da Bahia como requisito para obtenção do título de bacharel em Ciência da Computação, orientado pelo professor Adilson Pereira.

VITÓRIA DA CONQUISTA, 2013

JOSEMAR BRAGA LUZ

AUTOMAÇÃO RESIDENCIAL COM STARGATE-888 VIA PORTA USB

Monografia, apresentado ao Curso de Ciência da Computação da Universidade Estadual do Sudoeste da Bahia como requisito para obtenção do título de bacharel em Ciência da Computação, orientado pelo professor Adilson Pereira.

Vitória da Conquista, Março de 2013.

Prof. Adilson Pereira (Orientador)

Prof. Hélio Lopes dos Santos

Prof. Roque Mendes Prado Trindade

VITÓRIA DA CONQUISTA, 2013

RESUMO

Observando o crescente destaque de tecnologia na sociedade, este trabalho tem por objetivo mostrar a automação residencial controlada por um computador pessoal , conectado via USB, com a utilização da placa *Stargate-888*. Primeiramente, é analisada a estrutura da tecnologia USB (Universal Serial Bus). Em seguida será abordada o conceito de *Open Source Hardware*, enfatizando a placa *Stargate-888*, utilizada como base para a realização desse trabalho. Por fim, é mostrado o projeto realizado, utilizando algumas das inúmeras aplicações que podem ser desenvolvidas através dessa placa, juntamente com o software desenvolvido, *Stargate_App*.

Palavras-chave: USB, automação residencial, *Open Source hardware*, *Stargate-888* e *Stargate_App*.

ABSTRACT

Noting the growing prominence of technology in society, this paper aims to show the home automation controlled by a personal computer, connected via USB, using the Stargate-888 card. First, we analyze the structure of USB (Universal Serial Bus). Then is discussed the concept of Open Source Hardware, stressing the board Stargate-888, used as the basis for this work. Finally, shows the project carried out by using some of the many applications that can be developed through this board, along with the software developed, Stargate_App.

Keywords: USB, home automation, open source hardware, Stargate-888 and Stargate_App.

Sumário

CAPITULO 1 – INTRODUÇÃO	6
1.1 Proposta de Trabalho	7
1.2 Objetivo Geral.....	7
1.3 Objetivos específicos.....	7
1.4 Metodologia de Trabalho.....	7
1.5 Organização da monografia.....	8
CAPITULO 2 - A INTERFACE USB.....	9
2.1 Características Físicas do protocolo USB.....	9
2.2 Protocolo USB.....	12
2.3 USB 3.0.....	15
CAPITULO 3 OPEN SOURCE HARDWARE.....	16
3.1 Termos de Distribuição Open Source Hardware.....	16
3.2 Exemplos de Open Source Hardware.....	18
3.3 Placa Stargate-888.....	20
3.3.1 Microcontrolador PIC 18F4550.....	21
3.3.2 Distribuição de pinos na Stargate-888.....	23
CAPITULO 4 AUTOMAÇÃO RESIDENCIAL COM USO DA PLACA STARGATE 888	27
4.1 Descrição do projeto.....	27
4.2 Software utilizado no projeto.....	31
4.3 Considerações Finais.....	34
CAPÍTULO 5 – CONCLUSÃO.....	35
BIBLIOGRAFIAS	36

CAPITULO 1 – INTRODUÇÃO

Contextualizado por Lacombe (2004) , automação (Prudente, 2011 ; Bolzani, 2004) é a aplicação de técnicas computadorizadas ou mecânicas para diminuir o uso de mão-de-obra em qualquer processo, especialmente o uso de robôs nas linhas de produção. A automação diminui custos e aumenta a velocidade de produção.

Dentre os vários ramos da automação, como por exemplo, industrial e comercial, existe a automação residencial que visa o conforto, melhorias na comunicação e uma diminuição no consumo de energia.

O mercado oferece inúmeras tecnologias para as residências, tornando-as cada vez mais automatizadas. Consequentemente, a busca da simplicidade e facilidade de manuseio destes é uma tendência de mercado.

A automação pode ser entendida como uma tecnologia integradora de três áreas: a eletrônica, responsável pelo hardware; a mecânica, na forma de dispositivos mecânicos (atuadores) e a informática, responsável pelo software que irá controlar todo o sistema, conforme Rosário (2005) . Deste modo, um projeto nesta área exige uma grande gama de conhecimentos, por parte de um ou vários projetistas.

O protocolo USB é uma tecnologia que tornou mais simples, fácil e rápida a conexão de diversos aparelhos ao computador. Por esse motivo, a escolha de uma placa com apoio a essa tecnologia era necessária. Foi escolhida a placa *Stargate-888*, que atua com comunicação via USB, de baixo custo, utiliza o microprocessador (Pereira 2003; Moreno, Penteado e Cezar 2005) PIC 18F4550 (Miyadaira 2009; Souza 2011) e ainda, possui um bom auxílio por parte dos desenvolvedores na sua elaboração de projeto, diz Viana (2010).

Este projeto tem por objetivo principal mostrar a aplicação de sensores e atuadores no meio residencial utilizando a placa *Stargate-888* e alguns elementos da eletrônica, além do software, para ter um monitoramento a partir de um computador pessoal, destacando a origem do hardware e software, que são *open source* (“aberto a todos”)(Curtis 2009), visando apresentar novas maneiras de desenvolver a aplicação com menor custo. Para isso, foi desenvolvido uma maquete para concretização do projeto.

1.1 Proposta de Trabalho

Deseja-se unir software, hardware e ações residenciais para formação de uma automação residencial controlada pelo computador pessoal. Então necessita-se, definir o conjunto de informações sobre cada tópico para finalizar o trabalho com sucesso.

1.2 Objetivo Geral

Automatizar um ambiente simulado residencial, controlado por um computador pessoal conectado via porta USB, com utilização da placa *Stargate-888*.

1.3 Objetivos específicos

- Definir o que realizará o projeto;
- Definir os *hardwares* para o trabalho;
- Adquirir o material necessário do projeto;
- Construir o ambiente residencial;
- Implementar o *software* de comunicação com *hardware* ;
- Testar *software*;
- Executar o projeto;
- Documentar monografia.

1.4 Metodologia de Trabalho

Primeiramente, entende-se o que se quer realizar neste trabalho, definindo o material necessário e, posteriormente, aquisição deles. Então passa-se para a construção física do projeto. Concluído essa fase, tem-se toda estrutura pronta. Agora, precisa-se do *software* para interação computador com *hardware* construído. Nessa etapa é importante dizer que o projeto

desenvolvido baseia-se em uma plataforma *open source*. Logo, a placa escolhida acompanha o *software* de forma genérica para sua execução, então adapta-se o *software* de acordo o projeto. Por fim, com o projeto finalizado basta fazer a documentação, ou seja, escrita da monografia.

1.5 Organização da monografia

Esta Monografia está organizada do seguinte modo:

O capítulo 2, a interface USB, faz definição e apresentação dessa porta de conexão muito utilizada atualmente para diversos fins.

O capítulo 3, *open source hardware*, define o conceito de projeto aberto, a qual possibilita a construção do trabalho de maneira mais simples e com baixo custo, sendo essa plataforma a qual o projeto será realizado.

O capítulo 4, automação residencial com uso da placa *Stargate-888*, tem por finalidade abordar todo o projeto realizado, detalhando cada fase de suas aplicações.

O capítulo 5, fará as conclusões finais.

CAPITULO 2 - A INTERFACE USB

Tempos atrás, conectar dispositivos ao computador era uma tarefa difícil, muitas vezes realizada somente por técnicos ou usuários com experiência. Diante de vários tipos de cabos e conectores, era necessário descobrir em qual porta do computador conectar cada um dos dispositivos que iria instalar. A configuração interna ainda era pior, já que o usuário tinha que, quase sempre, configurar *jumpers*.

Diante de situações desse tipo, surgiu a necessidade de criar um padrão amigável que facilitasse a conexão de dispositivos ao computador. Segundo Messias (2007) o USB (Universal Serial Bus) surgiu um pouco antes de 1995 com o consórcio das empresas: Compaq, Hewlett-Packard, Intel, Microsoft, Philips entre outras. A 1ª versão (1.0) surgiu com velocidade de transmissão de até 1.5 Mbps, conhecida por *Low-Speed*. Logo após veio a 1.1 com velocidade se estendendo para 12 Mbps (*full-Speed*). Lançada em 2000, a versão 2.0, com velocidade que chega até 480 Mbps (*High-Speed*).

2.1 Características Físicas do protocolo USB

Um dos principais motivos que levou a criação da tecnologia USB foi a necessidade de facilitar a conexão de variados dispositivos ao computador. Sendo assim, o USB dispõe das seguintes características, segundo Alecrim (2010):

- Padrão de conexão - qualquer dispositivo compatível com USB usa padrões definidos de conexão ;
- *Plug and play* – algo como plugar e usar. Alguns exigem instalação de *drivers* ou *software* específicos. No entanto, mesmo nesses casos, o sistema operacional reconhecerá a conexão do dispositivo imediatamente;
- Alimentação elétrica – Sua arquitetura dispõe de corrente elétrica (5v) que podem alimentar o dispositivo, porém é uma corrente baixa. Por conta disso, há até determinados dispositivos como celulares, mp3 *players*, que têm suas baterias carregadas via USB. Na necessidade de uma corrente maior, como exemplo

impressoras, *scanner*, utiliza-se fonte externa;

- Ampla compatibilidade – o padrão USB é compatível com diversas plataformas e sistemas operacionais. O Windows, por exemplo, o suporta desde a versão 98. Sistemas operacionais Linux e Mac também são compatíveis. Atualmente, é possível encontrar portas USB em vários outros aparelhos, como televisores, sistemas de comunicação de carros, aparelhos de som, etc;
- *Hot-swappable* - dispositivos USB podem ser conectados e desconectados a qualquer momento. Em um computador, por exemplo, não é necessário reiniciá-lo ou desligá-lo para conectar ou desconectar o dispositivo;
- Cabos de até 5 mts - os cabos USB podem ter até 5 metros de tamanho, e esse limite pode ser aumentado com uso de *hubs* ou de equipamentos capazes de repetir os sinais da comunicação. Acima de 5 mts a transmissão de dados excede o limite de 1,5 microssegundos, perdendo informações e
- Conexão de vários aparelhos ao mesmo tempo - É possível conectar até 127 dispositivos, porém há necessidade de utilizar *hubs* conectados em cascata. Os *hubs* são componentes que fornecem novos canais físicos para que possam inserir novos dispositivos, segue ilustração nº1.

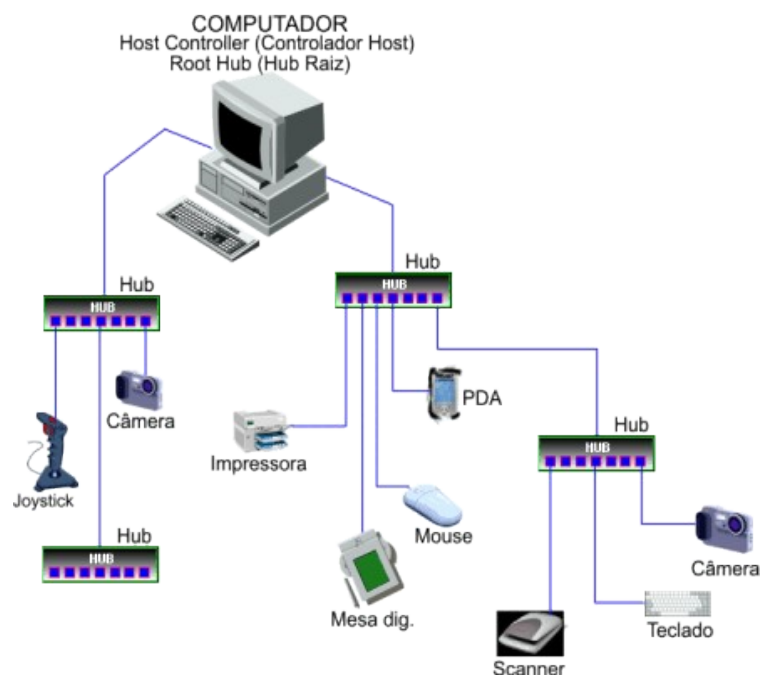


Ilustração 1: Arquitetura típica de um sistema USB com hub.

A tecnologia USB conta com vários tipos de conectores, conforme ilustração nº2, sendo o tipo A, o mais conhecido, uma vez que está presente na maioria dos computadores compatíveis com a tecnologia, além de poder ser encontrado em outros tipos de aparelhos. Uma vez que o objetivo principal do padrão USB é facilitar a conexão de variados dispositivos ao computador, geralmente os cabos desses aparelhos são do tipo A em uma ponta e de algum dos outros tipos na outra, podendo a segunda ponta ter também, algum formato proprietário, isto é, específico de um fabricante. O USB tipo B, que é o conector de formato quadrado, encontrado em impressoras e outros periféricos, além do USB mini 5P e o USB mini 4P, dois formatos menores, que são utilizados por câmeras, mp3 players, palmtops, etc.

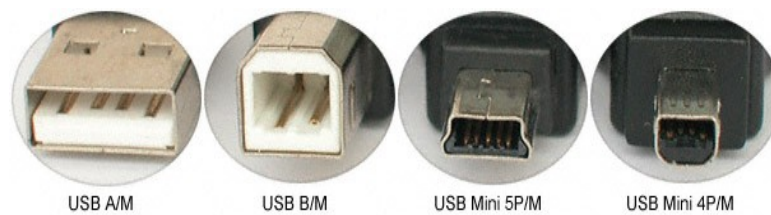


Ilustração 2: Tipos de conectores USB ("macho").

Os quatro tipos utilizam a mesma pinagem, sendo o diferencial seu formato físico. Existem ainda, alguns formatos de conectores proprietários, geralmente diferenciada destes quatro formatos. Por serem compatíveis, é relativamente fácil encontrar adaptadores diversos para permitir encaixar cabos com conectores de formatos diferentes, tais como os demonstrados na ilustração nº3.

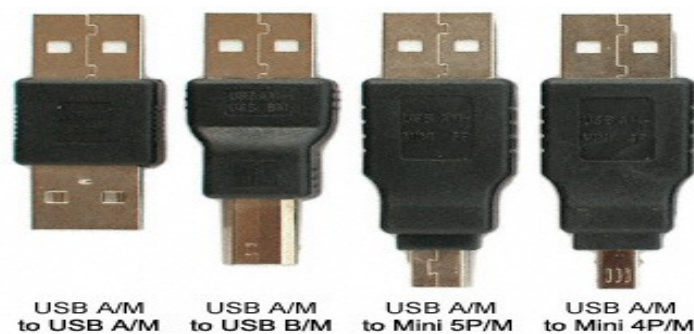


Ilustração 3: Adaptadores USB.

Um cabo USB possui dois fios de energia (+5v e o terra) e um par trançado para condução dos dados, aqui representado por D+ e D-, apresentados na ilustração nº4.

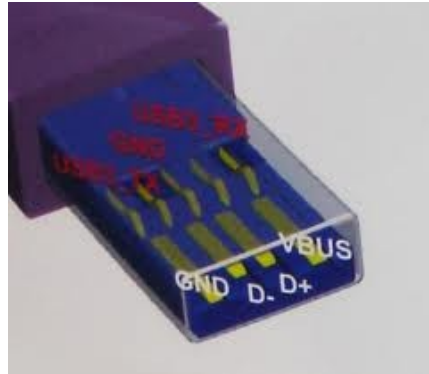


Ilustração 4: Configuração dos fios no conectos USB tipo A.

2.2 Protocolo USB

Segundo Messias (2007), protocolo é uma das partes mais importantes do funcionamento do sistema USB. É através dele que um ou mais dispositivos conversam e trocam informações com o *host*. Neste a pilha de protocolo está disponível em forma de *drivers* de arquivos .sys, .dll, .drv, .exe, etc. Já no dispositivo, o protocolo pode ser encontrado dentro de um microcontrolador específico, como por exemplo, em um PIC que disponha de uma certa quantidade de memória. Nesse caso, o protocolo é escrito com base na norma USB, em uma linguagem específica: C ou *Assembly*. Depois é compilado e gravado na memória do microcontrolador através de um gravador PIC. Esta é uma das maneiras mais complexas, trabalhosas e não muito produtiva. Ainda segundo Messias, há microcontroladores PIC como o 18F2450 e 18F4550 (este utilizado na placa *Stargate-888*) que internamente já dispõem do protocolo, tornando assim mais fácil o desenvolvimento de aplicações. Uma outra maneira, é através de um *chipset* que carrega dentro de si toda a funcionalidade do protocolo USB, livrando o microcontrolador de todo o trabalho pesado.

Diferente da porta paralela ou serial que se comunica com sinais elétricos e um simples programa, o USB, apesar de facilitar as conexões, precisam carregar o seu protocolo num *chipset* ou microcontrolador.

O protocolo USB possui vários recursos como: controle de fluxo de dados, detecção e correção de erros, detecção de conexão e remoção de dispositivos, entre outros.

A transmissão de dados é baseado no envio de pacotes. A transmissão começa quando o *host* (ou dispositivo USB) envia um pacote (*Token Packet*) descrevendo o tipo, direção de transmissão, o endereço do dispositivo USB e o referido número de *endpoint*. Logo o dispositivo USB decodifica o campo de endereço, reconhecendo que o pacote lhe é referente. A seguir, a fonte de transmissão envia um pacote de dados (*Data Packet*) ou indica que não há dados há transferir. Por fim, o destino responde com um pacote de *handshake* (*Handshake Packet*) indicando se a transferência obteve sucesso.

O USB define formatos diferentes para os três tipos diferentes de pacotes existentes: *Token*, *Data* e *Handshake*, representados na ilustração nº5 por: (a), (b) e (c), respectivamente.

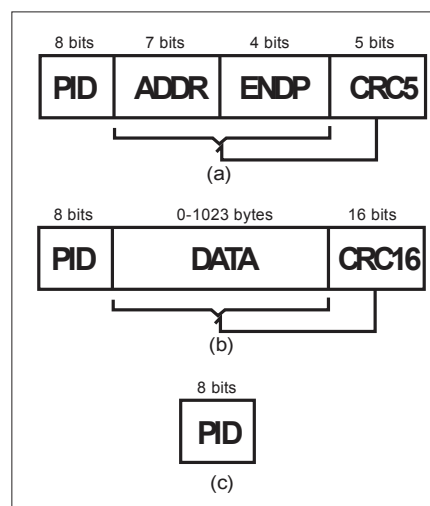


Ilustração 5: (a)Token, (b) Data e (c) Handshake

Esses pacotes possuem os seguintes campos:

- PID (*Packet Identifier*): composto de oito bits. Os quatro mais significativos identificam e descrevem o pacote e os demais são bits de verificação para prevenção de erros (*check bits*). Esses *check bits* são constituídos pelo complemento um dos quatro bits identificadores;
- ADDR (*Address*): endereço do dispositivo USB envolvido. Composto de 7 bits, limita o número de dispositivos endereçáveis em 127;

- ENDP (*endpoint*): possui quatro bits que representam o número de endpoint envolvido. Permite maior flexibilidade no endereçamento de funções que necessitem de mais de um subcanal;
- CRC (*cyclic Redundancy Checks*): bits destinados à detecção de erros na transmissão;
- DATA : bits de dados.

Um *Token packet* pode identificar a transmissão como: transferência para *host* (IN); transferência para função (OUT); início de *frame* (SOF) ou transferência de informações de controle para o *endpoint* (SETUP). O CRC de um *token packet* possui 5 bits e atua apenas sobre os campos ADDR e ENDP, uma vez que o PID possui seu próprio sistema de prevenção contra erros. Os dados transmitidos via *data packet* devem ter um número inteiro de bytes. O CRC de um *data packet* possui 16 bits que atua sobre o campo DATA. O *handshake packet* é constituído apenas de um PID. Esse pacote pode significar que o receptor recebeu os dados livres de erros (ACK), que o receptor não pode receber os dados, que o transmissor não pode transmitir (NAK) ou que o *endpoint* está em parado (STALL).

Quando o computador é iniciado, ele designa um endereço para cada dispositivo conectado ao barramento, processo esse chamado por enumeração. Além de enumerar os dispositivos, ele encontra o tipo de transferência de dados do mesmo. Há três tipos, definidos por Aguiar (2010):

- *Interrupt* (Interrupção) - usado em dispositivos como mouse ou teclados, que enviam poucos dados;
- *Bulk* - Um dispositivo como impressoras, que recebe dados em grandes pacotes, utiliza esse modo de transferência. Um bloco é enviado à impressora (em 64 fragmentos de bytes) e verificado para garantir se está correto;
- *Isochronous* (Isócronos) - um dispositivo de transmissão contínua (como webcams) utiliza esse modo. Os dados fluem entre dispositivos e o *host* em tempo real e não há correção, nem controle de erros.

Atualmente há um quarto tipo, definidos por *Control*, usada pelo *host* para transmissão de parâmetros de controle e configuração de dispositivo.

2.3 USB 3.0

Atualmente, existe no mercado a USB 3.0, conhecida por *superspeed* USB, uma tecnologia que atende a necessidade atual de maior taxa de transmissão de dados, por exemplo, vídeos em alta definição. Para tal fim, a USB 3.0 soluciona tal problema, uma vez que possui taxa máxima de transferência de dados de 4.8 Gbps (600MB/s), ou seja, dez vezes mais rápida que a USB 2.0 (60MB/s).

A USB 3.0 possui cinco fios a mais dentro do seu cabo (totalizando nove), os quais funcionam de maneira unicamente para enviar e recebe, ao mesmo tempo, dados do computador. Esses nove fios ficam caracterizados por:

- Vbus – positivo;
- D- e D+ - par trançado transferência 2.0;
- GND – negativo;
- StdA_SSTX - e StdA_SSTX+ - *superspeed* par diferencial transmissor;
- GND_DRAIN- terra para retorno de sinal;
- StdA_SSDX - e StdA_SSDX+ - *superspeed* par diferencial receptor.

Além disso, o conector USB 3.0 funcionará normalmente na porta 2.0, haja visto que os pinos do conector anterior continua os mesmos com as mesmas funções. Observe a ilustração nº 6, de um conector USB3.0 tipo A .

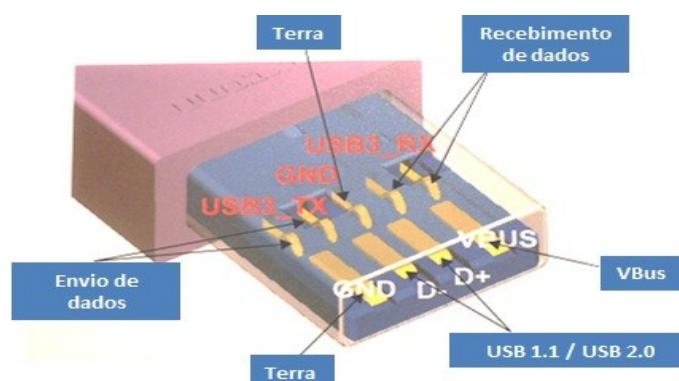


Ilustração 6: Estrutura interna de um conector 3.0 A

Agora, após conhecer um pouco mais sobre a tecnologia USB, será definido no próximo capítulo o projeto *open source hardware*, o qual o placa stargate-888, utilizada no trabalho, se caracteriza.

CAPITULO 3 OPEN SOURCE HARDWARE

Primeiramente, será definido o “*Open source hardware*”(OSHW) . Este é definido como projeto em que os criadores decidiram publicar todas as informações (diagrama esquemáticos, lista de material e circuito impresso), ou seja, a informação está aberto a todos. Ele é tratado da mesma maneira que a distribuição do software livre. Basicamente, muitos dos princípios do *open source software* são simplesmente adaptados para o equivalente em hardware. Contudo, existe a legislação de patentes que pode controlar o uso e fabricação de dispositivos físicos (TAPR *Open Hardware License*), segundo Viana (2010).

É importante observar que o hardware é diferente do software, pois utiliza-se recursos físicos para criar bens físicos. Logo, pessoas ou empresas produtoras de itens (produtos) sob licença OSHW tem o dever não implicar que tais produtos são fabricados, vendidos, garantidos ou sancionadas pelo designer original e também não fazer uso de quaisquer marcas de propriedade do designer original.

3.1 Termos de Distribuição Open Source Hardware

Os termos de distribuição de *Open Source Hardware* deve respeitar os seguintes critérios , por Andrade (2011):

1. Documentação - O hardware deve ser lançado com a documentação, incluindo arquivos do projeto, e deve permitir a modificação e distribuição dos mesmos. Onde essa documentação não for apresentada com o produto físico, deve haver um meio amplamente divulgado de obter essa documentação não mais do que um custo razoável de reprodução, de preferência, fazendo download através da Internet sem custo. A documentação deve incluir arquivos de desenho no formato preferido para fazer as alterações, por exemplo, o formato nativo de arquivo de um programa de CAD. Arquivos do projeto deliberadamente ofuscado não é permitido. Intermediárias formas análogas ao código compilado – como a arte de cobre prontos para impressão a partir de um programa de CAD – não são permitidas como substitutos. A licença pode exigir que os arquivos do projeto são fornecidos em formato totalmente documentado,

formato(s) aberto(s).

2. Âmbito – A documentação para o *hardware* deve explicar claramente qual parte do projeto, se não todo, está sob a licença;
3. Software necessário - Se o projeto requer software licenciado, incorporado ou não, para funcionar corretamente e cumprir suas funções essenciais, a licença pode exigir que uma das seguintes condições:
 - a) As interfaces estão suficientemente documentados tal que possa ser razoavelmente considerada simples para ser escrita em software de código aberto e que permita que o dispositivo funcione adequadamente e cumpra suas funções essenciais. Por exemplo, pode incluir o uso de diagramas de tempo detalhada do sinal ou pseudocódigo para ilustrar claramente a interface em funcionamento.
 - b) O software necessário é liberado sob uma licença de código aberto aprovada pela OSI.
4. Trabalhos derivados - A licença deve permitir modificações e trabalhos derivados, e deve permitir-lhes ser distribuído sob os mesmos termos da licença do trabalho original. A licença deve permitir a fabricação, venda, distribuição e utilização de produtos criados a partir dos arquivos do projeto, e de seus derivados.
5. Redistribuição gratuita - A licença não deve restringir nenhuma parte de vender ou doar a documentação do projeto. A licença não deve exigir um *royalty* ou outra taxa para tal venda. A licença não deve exigir nenhum *royalty* ou taxa relacionada com a venda de obras derivadas.
6. Atribuição - A licença pode exigir documentos derivados, e avisos de direitos autorais associados aos dispositivos, para fornecer a atribuição aos licenciados, quando a distribuição de arquivos de projeto, os produtos fabricados e / ou seus derivados. A licença pode exigir que estas informações sejam acessíveis ao utilizador final, utilizando o dispositivo normalmente, mas não deve especificar um formato específico para publicação. A licença pode exigir que trabalhos derivados tenham um nome ou versão diferentes do projeto original.
7. Sem discriminação contra pessoas ou grupos - A licença não deve discriminar nenhuma pessoa ou grupo de pessoas.

8. Sem discriminação contra campos de trabalho - A licença não deve restringir ninguém de fazer uso da obra (incluindo hardware fabricado) em um campo específico de atuação. Por exemplo, não deve restringir o hardware a ser utilizado em uma empresa, ou de ser usado na pesquisa nuclear.
9. Distribuição da Licença - Os direitos conferidos pela licença deve aplicar a todos a quem o trabalho é redistribuído, sem a necessidade de execução de uma licença adicional por aquelas pessoas.
10. Licença não deve ser específica para um produto - Os direitos conferidos pela licença não deve depender do trabalho licenciado sendo parte de um determinado produto. Se uma parte é extraída de uma obra e usado ou distribuído dentro dos termos da licença, todas as partes para quem esse trabalho é redistribuído devem ter os mesmos direitos que aqueles que são concedidas para a obra original.
11. Licença não deve restringir *hardware* ou *software* - A licença não deve colocar restrições em outros itens que são agregados com a obra licenciada, mas não derivados do mesmo. Por exemplo, a licença não deve insistir que todos os outros hardwares vendidos com o item licença ser de código aberto, nem que apenas o *software open source* ser usado externalmente ao dispositivo.
12. Licença deve ser tecnologicamente neutra - Nenhuma parte da licença poderá ser subordinada a qualquer pessoa, tecnologia, parte ou componente específico, material, ou o estilo de interface ou utilização.

Existem várias iniciativas *Open Source Hardware*, dentre elas destacam-se a Aurora 224, a *Fusebox*, a Arduino, a *Beagleboard* e a *Stargate-888*, esta utilizada nesse projeto.

3.2 Exemplos de Open Source Hardware

Como 1º exemplo, a Aurora 224, *open source mixer* de áudio, vídeo e mix de cores. É um mixer multicanal alimentado por USB, possui: 2 canais de *faders*, 1 *crossfader* a/b, 8 botões retroiluminados, 24 controles de efeito, controla áudio e luz ambiente e compatível com software MIDI, segundo Torrone (2008).

Um outro exemplo, o *Fusebox*, console de videogame open source. Ele é projetado para pessoas que sabem um pouco de programação para expandir, desenhar e criar seus próprios jogos e demos. É um núcleo completo que roda em background e faz todo o vídeo e processamento de áudio para que o código fique limpo e fácil de entender, também segundo Torrone.

Temos também, o Arduíno, uma plataforma open source que permite que computadores pessoais interajam com mundo físico. Arduíno pode perceber o ambiente por receber a entrada de uma variedade de sensores e pode afetar seus arredores por luzes de controle, motores e outros atuadores. Baseado em microcontrolador e ambiente de programação própria. Este possui características similares ao *Stargate-888*, que será apresentado mais adiante.

Como um 4º exemplo, o *Beagleboard*, CPU compacta de baixo custo. *BeagleBoard* é uma pequena placa que equivale a um computador completo. Ele tem entrada USB, entrada de cartões de memória (tipo de câmera digital), entrada e saída de áudio, saída de vídeo e tudo que um computador precisa ter para funcionar. As especificações oficiais, segundo Vegano (2008), são:

- 600 MHz superscalar ARM Cortex A8 processor;
- HD Vídeo (720p) capable C64x DSP core;
- 128MB RAM;
- 256MB Flash;
- I2C, I2S, SPI, MMC/SD (via expansion connector);
- DVI-D;
- JTAG;
- S-Video;
- SD/MMC+;
- Stereo Out;
- Stereo In;
- USB 2.0 HS OTG;
- *RS-232 Serial*;

Por ultimo exemplo, tem-se a versão *stargate* 864, projetada para aplicações 3D, jogos de corrida e simuladores de vôo, que demandam uma grande quantidade de botões e eixos analógicos. Essa placa tem sob característica receptora, somente, diferenciando da *stargate* 888 que é atuadora/receptora.

Segue a ilustração nº 7 com a imagem de cada uma dos exemplo citados acima.

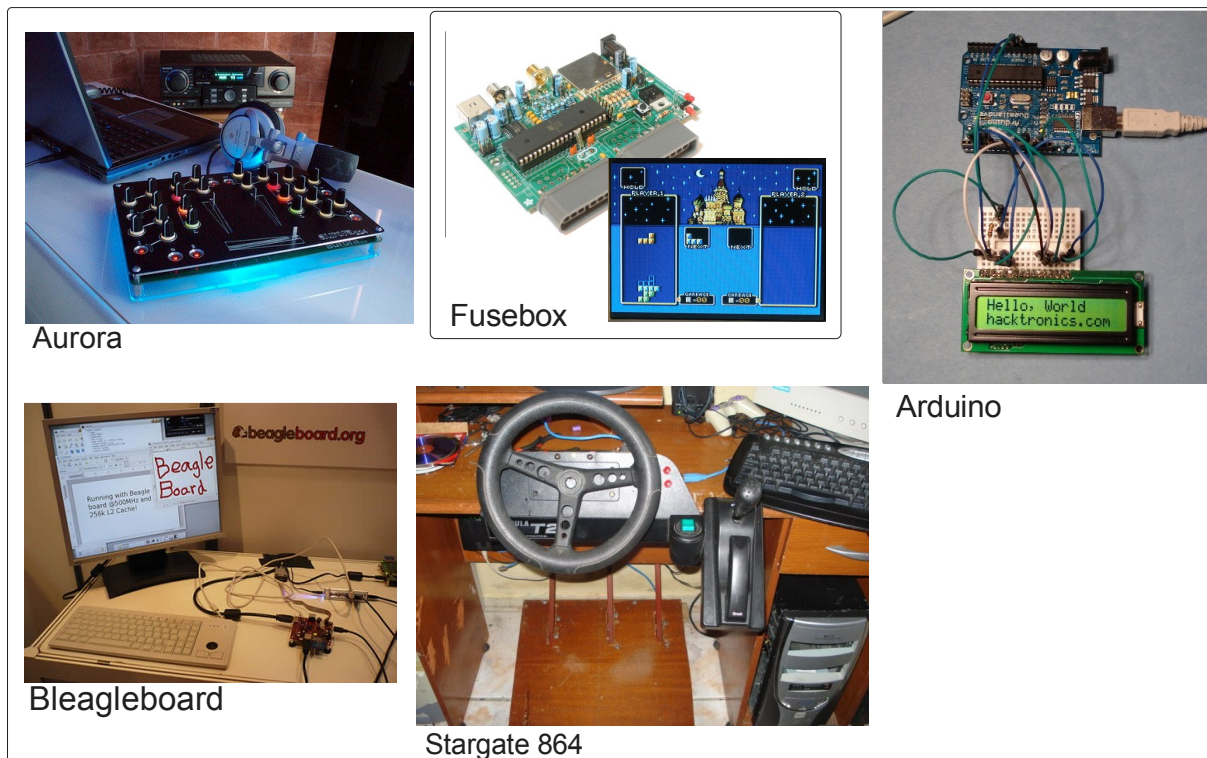


Ilustração 7: Exemplos de projetos Open source Hardware

Esses são alguns exemplos das aplicações *open source hardware* que existem no mercado e mostram os diferentes tipos de utilizações dos dispositivos.

3.3 Placa Stargate-888

Segundo Viana (2010), “A plataforma Stargate é uma placa eletrônica, baseada em um microcontrolador da PIC18F4550 (Miyadaira 2009), programado para facilitar o acesso a componentes de hardware e software do seu PC”. A depender do modelo, pode atuar como: receptora, modelo o qual a placa só recebe informações dos componentes a ela conectada, ou seja, ela somente recebe dados que são convertidos a ação ou valores no software utilizado;

ela, também, pode ser receptora/atuadora, caracterizado por receber informações e enviar impulsos elétricos que poderão ser convertidos a uma ação. Sua utilização permite a integração de dispositivos externos ao computador através da porta USB. Dentre os modelos de placa *Stargate*, será utilizado o modelo *Stargate-888*, conforme ilustração nº8, por ter configuração receptora/atuadora.

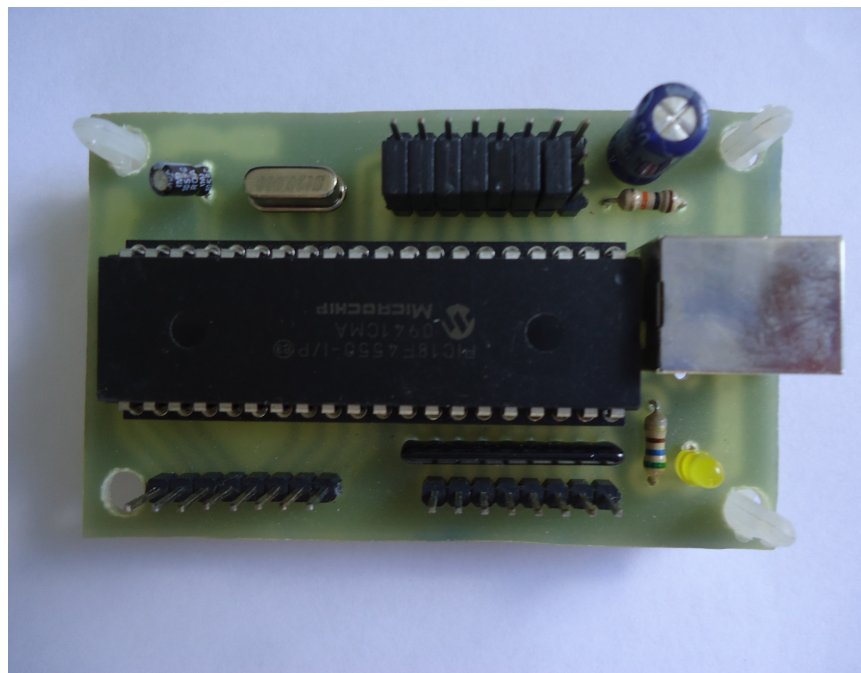


Ilustração 8: Imagem da placa Stargate 888

3.3.1 Microcontrolador PIC 18F4550

Como disse Viana (2010), essa placa é baseada em um microcontrolador PIC18F4550. E se faz entender as características gerais desse microcontrolador e sua configuração de pinagem na placa, o que será abordado neste capítulo.

De acordo com dados do fabricante (Microchip) podem-se resumir as unidades internas desse microcontrolador nos tópicos abaixo:

- Memória FLASH de 32 Kbytes para armazenamento de programa;
- Memória SRAM de 2 Kbytes para armazenamento de dados;

- Memória EEPROM de dados : 256 bytes;
- 35 portas configuráveis como entradas ou saídas;
- 13 portas configuráveis como canais de entradas analógicas;
- Módulo CPP (Capture / Compare / PWM);
- Porta paralela de 8 bits (SPP – *Streaming Parallelal Port*);
- 4 temporizadores de 8 e 16 bits;
- *Watchdog Times*;
- Frequência de operação de até 48 Mhz;
- Múltiplas fontes de interrupção (20);
- 2 comparadores;
- Periféricos avançados de comunicação. Porta de comunicação serial, porta de comunicação USB 2.0;
- Arquitetura Harvard, tecnologia RISC com um conjunto de 75 instruções e
- Pilha de 31 níveis.

Na ilustração nº9, segue a distribuição geral das pinagens do PIC18F4550.

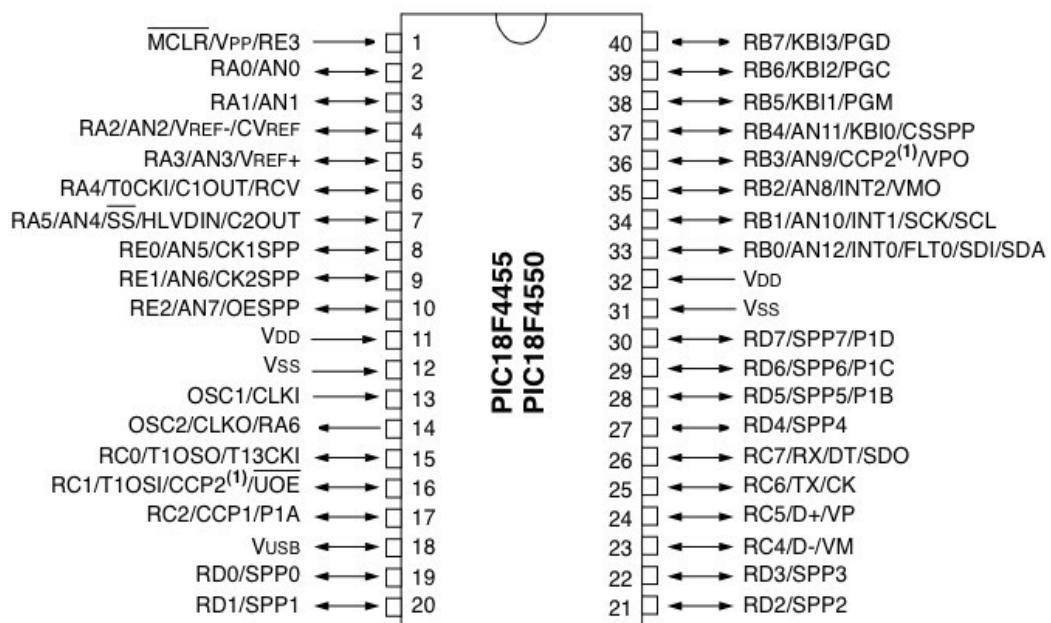


Ilustração 9: Distribuição dos pinos do microcontrolador PIC18F4550

De acordo as pinagens representadas na imagem anterior, será explicada os principais pinos:

- RA, RB, RC, RD e RE - representa 35 portas de entrada e saída;
- AN - 13 canais de entradas analógicas;
- VDD e VSS - pinos de alimentação: positivo(+5v) e negativo, respectivamente. OSC - pino de entrada para o oscilador externo;
- OSC – pino de entrada para oscilador externo;
- RX e TX - portas de comunicação serial;
- D+ e D- - portas de comunicação USB.

Observa-se que os pinos podem ocupar mais de uma função, como é visto, por exemplo, no caso de coincidência da porta RB0 e entrada analógica AN12. Por isso, a cada forma de utilização do microcontrolador, os valores máximo de portas e canais serão alterados, assim como a configuração de pinagem.

3.3.2 Distribuição de pinos na Stargate-888

Será detalhada a pinagem utilizada na modelo *Stargate-888*, como também, o que corresponde cada pino, conforme a ilustração nº10.

Analisando a imagem, parte esquerda encontra-se três colunas de pinos: o positivo (5v), o negativo (terra) e as entradas analógicas, respectivamente. Cada linha representa uma entrada analógica, nomeadas por: X, Y, Z, Rx, Ry, Rz, SL e Di, sendo o positivo e negativo comum a todas.

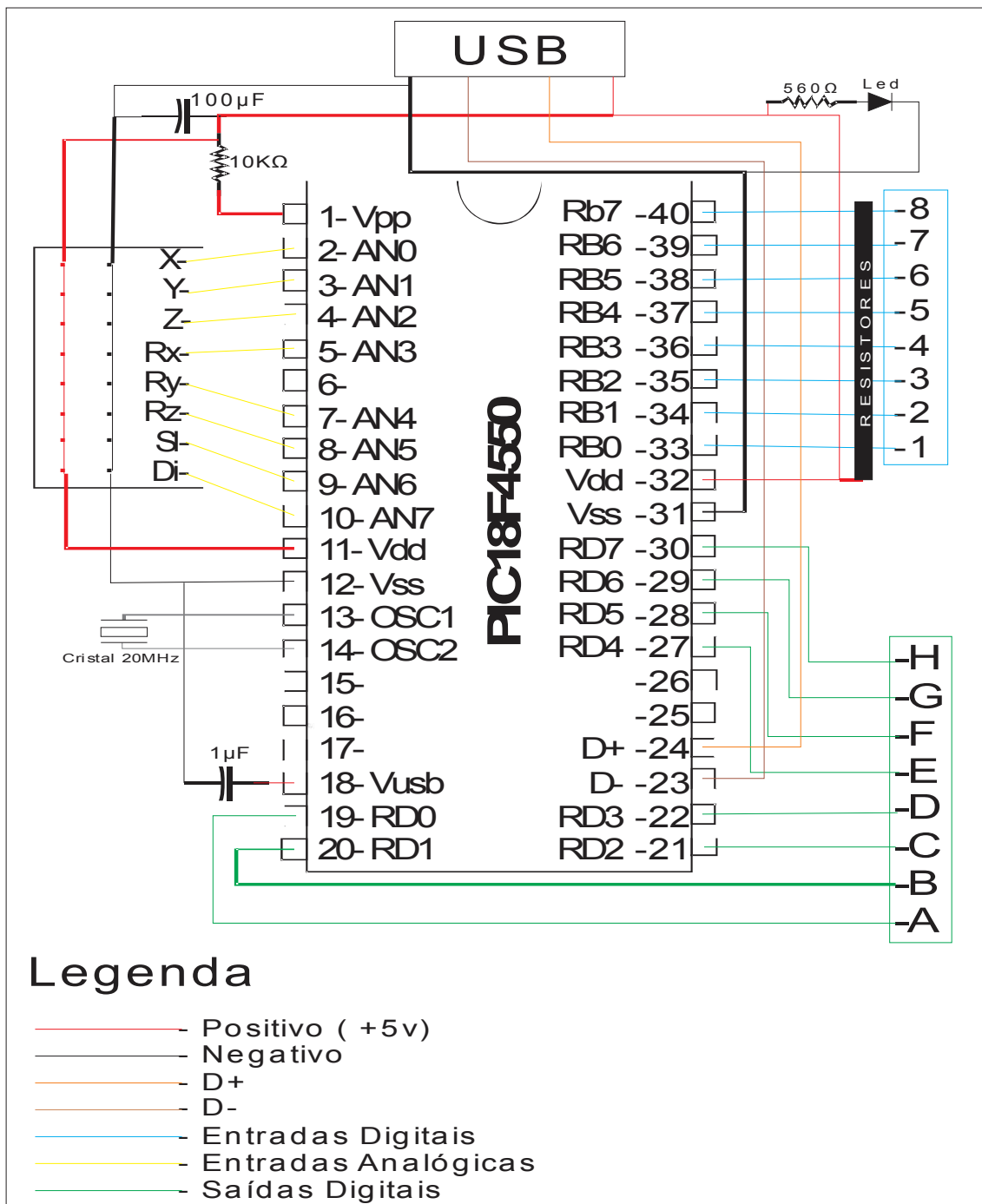


Ilustração 10: Diagrama Esquemático da placa Stargate 888

Nas entradas analógicas podem-se conectar até 8 potenciômetros (deslizantes ou rotativos). Na ilustração nº11, está representada o modo de conexão de um potenciômetro, onde o fio amarelo deve ser conectada a uma das entradas analógicas da placa, e as

extremidades do potenciômetro ligados ao positivo e negativo, fio vermelho e preto, respectivamente. Para maior desempenho, o fabricante recomenda o uso de potenciômetro de 10K Ohm. Este componente eletrônico possui resistência elétrica ajustável, ou seja, retorna valores flutuantes da sua aplicação, no projeto final tem-se uma aplicação desse componente.

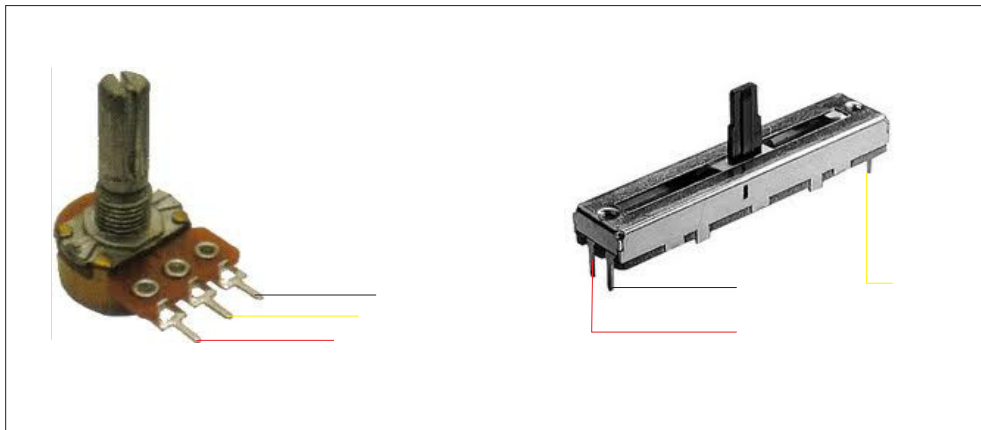


Ilustração 11: Exemplo de potenciômetro : rotativo e deslizante, respectivamente.

Outros sensores analógicos como: temperatura (ex: LM35), efeito hall, podem ser conectados de forma semelhante ao potenciômetro.

Quando não utilizadas todas as entradas analógicas, recomenda-se aterrar as inutilizadas para evitar captura de sinais desses eixos, prejudicando o mapeamento.

As entradas digitais (localizadas na parte direita superior da placa) são representadas pelos números de 1 a 8, cujas instalações de botões comuns (dois pinos), devem ter um fio em um pino enumerado e o outro no aterramento da placa.

As saídas digitais (localizadas na parte direita inferior da placa) estão representadas pelas letras de A a H. Suas conexões devem ter o fio receptor positivo conectado em algum dos pinos representados por letras e o outro no terra. As saídas digitais atuam com valores de potencial 0v ou 5v, aproximadamente.

Resumindo, o modelo *Stargate-888*, escolhido para realizar o projeto, caracterizado por funções atuadoras/receptoras, apresenta a seguinte configuração:

- 8 Entradas Analógicas, com precisão de 10-bit (1024 níveis) e filtro FIR

linear de

4-amostras resultando numa leitura estável de 8-bit @1ms;

- 8 Entradas Digitais, para botões comuns;
- 8 Saídas Digitais, que atuam com sinais 0 ou 1 (0 ou 5v);
- Dimensões da placa: 75x45 mm e
- Interface: USB 2.0 Hi-speed (12Mbit/s) .

Observe a ilustração nº 12, o layout da *Stargate-888* com representação de suas conexões, contendo o conector USB (tipo B fêmea) na parte superior, o microcontrolador PIC18F4550 no centro, além de seus componentes eletrônicos.

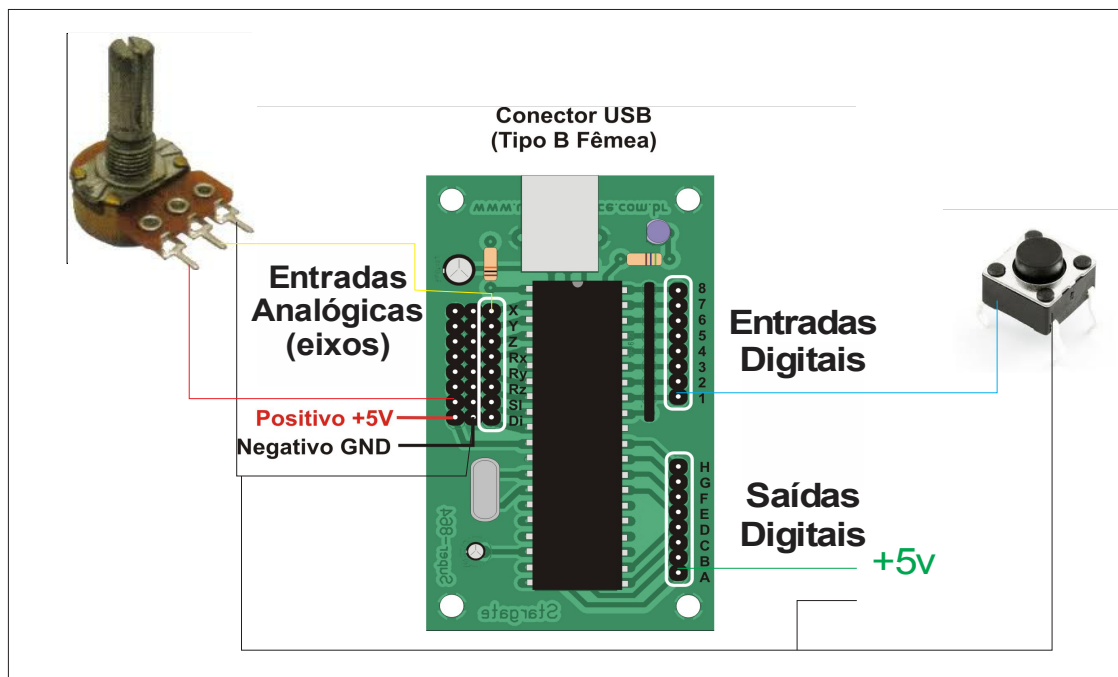


Ilustração 12: Layout da placa Stargate-888

Este é o modelo que utilizaremos para desenvolver a aplicação proposta nesse projeto. Por se tratar de uma placa com entradas analógicas e digitais, e saídas digitais, torna-se a mais adequada para uso no trabalho. Para isso foi desenvolvido um projeto que possui a utilização de cada tipo de entrada e saída composto na placa, que serão apresentados no próximo capítulo.

CAPITULO 4 AUTOMAÇÃO RESIDENCIAL COM USO DA PLACA STARGATE 888

Imagine você em sua casa, de frente ao computador, fazendo um monitoramento dela. Por exemplo, através de sensores saber a temperatura fora e dentro da casa, saber a quantidade de água que existe nos seus reservatórios (um em cima da casa e outro no térreo, por alguma necessidade você tem um reservatório no térreo, pois a água chega primeiro nele, ou que receba a água da chuva). Por outro lado verificar se as portas de entradas e saídas da residencia estão abertas e/ou fechadas. E além disso tudo, poder interagir com o meio, ou seja, abrir porta de entrada ou garagem, também fechar garagem, ligar várias lâmpadas da casa, em uma necessidade acionar uma bomba d`agua que transfira água do reservatório térreo para o principal.

Todo esse cenário seria possível produzir com uso da placa *stargate-888*, com uso dessa simples placa e alguns conhecimento eletrônicos. Como já foram definidas as características da *Stargate-888*, para um segundo estágio, tem-se a sua aplicação, objetivo principal do trabalho. Para melhor entendimento do projeto, foi produzido uma maquete para uma visualização real e concreta da utilização da placa para automação residencial controlada a partir de um computador pessoal. Esta possui algumas de muitas outras aplicações possíveis de realizar com ela. A seguir detalharemos o projeto construído para fim de demonstração do trabalho.

4.1 Descrição do projeto

Primeiramente, será apresentado o diagrama esquemático, assim como detalhamento de cada função dos componentes do projeto.

A placa fornece até 8 entradas analógicas, 8 entradas digitais e 8 saídas digitais. Para realização do projeto utilizamos apenas uma entrada analógica, uma entrada digital e três saídas digitais, poderíamos utilizar todos, mas a aquisição de mais material exigiria um gasto maior.

A maquete possui a utilização de um potenciômetro, com um exemplo de sua utilização, na entrada analógica; uma chave push-button, representando a entrada digital e três dispositivos (ventoinha, led e neon), representando as saídas digitais. Observe a ilustração nº13 e posteriormente o detalhamento de suas funcionalidades.

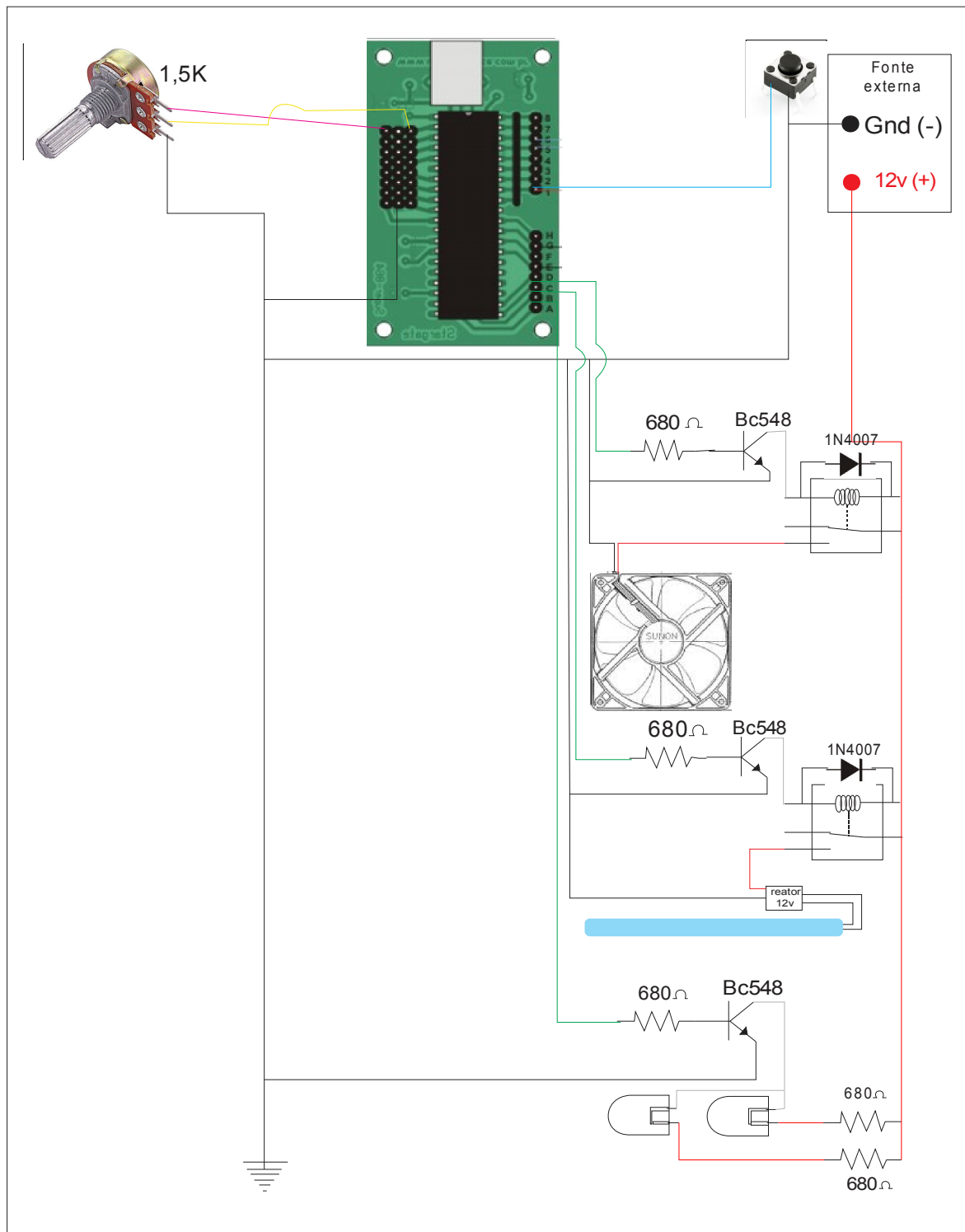


Ilustração 13: Diagrama esquemático do projeto

No esquema cada ligação é representada por cores diversas, sendo estas:

- Preto - negativo (terra) do projeto, junção do negativo da placa e fonte externa;
- Vermelho - Positivo, +12v (fonte externa);
- Rosa - positivo, +5v (USB);
- Amarelo – Valor elétrico retornado pelo potenciômetro;
- Azul - Fio que quando acionado retorna a placa um terra (negativo), habilitando a entrada digital;
- Verde - Sinal positivo que atua quando a placa ativar a saída, seu valor vai de 0 para +5v.

Na parte superior esquerda está representado o potenciômetro de 1,5K Ω (o ideal é o de 10K Ω) com três fios: preto, amarelo e vermelho, representando , negativo, entrada analógica e positivo (5v), respectivamente. O potenciômetro foi instalado em uma suposta caixa d'água, como uma suposta bóia que faz ele girar de acordo nível da água, onde à medida que a água do recipiente altera o potenciômetro retorna à placa um valor de acordo a sua posição, valores de -127 a 127 , que o programa recebe e converte, através de cálculos, em litros. Com esse recurso a pessoa pode manter constante verificação da quantidade de água no seu reservatório e em caso de o nível está baixo, por diversos motivos, pode-se tomar providências antes que acabe a água.

Analisando esse exemplo, o potenciômetro rotativo não vai girar seu cursor 100%, então procura-se descobrir seu mínimo e máximo e altera a conversão do software para obter uma relação coerente no painel de comunicação do projeto.

Um outro componente que, também, poder ser utilizado nessas entradas é o sensor de temperatura (ex: LM35), que lhe retornará após conversão a temperatura do ambiente, objeto a qual foi aplicado.

Na parte superior direita encontramos uma chave push-button, que atua como dispositivo de fechamento de contato, no projeto ele atua como dispositivo que indica se a porta está aberta ou fechada. É acionado quando o negativo da placa se conecta a uma das entradas. Essas entradas digitais possibilita fazer um monitoramento de portas ou portões, por exemplo, com finalidade de saber se estão abertas ou fechadas.

Por último, no inferior direito da imagem, encontra as saídas digitais. Essas saídas

quando acionados pelo programa eleva sua tensão de 0v para 5v, porém sua corrente é muito baixa, impossível de acionar um dispositivo de corrente maior. Para isso precisamos de um esquema eletrônico com simples componentes da eletrônica para conseguirmos acionar aparelhos que exigem maior corrente ou tensão. Então necessita-se: resistores, transistor, relés, diodo e fonte externa.

Para melhor entendimento, foi ampliado o circuito eletrônico que possibilita a utilização de dispositivos de corrente e voltagem maior, observe a ilustração nº14. É necessário uma fonte externa com voltagem em 12v (DC) e uma corrente superior a 2A, compatível com os dispositivos utilizados, mas poderia ser outro valores inclusive corrente alternada AC (através de circuitos eletrônicos).

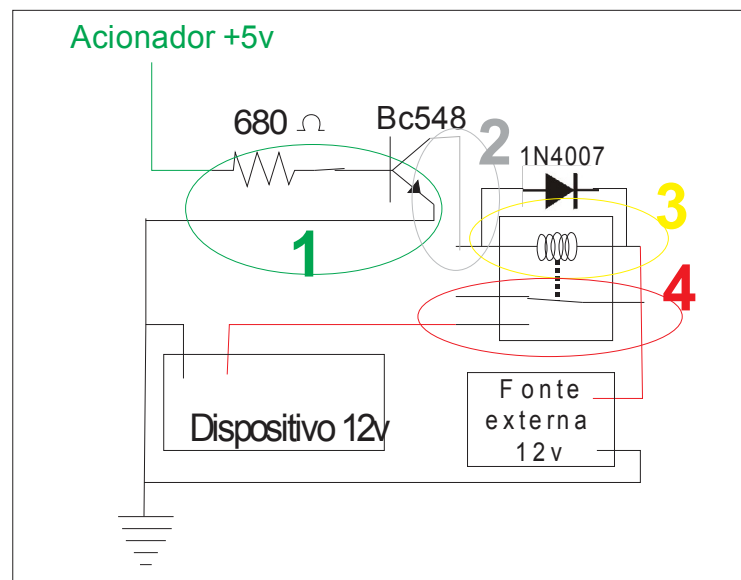


Ilustração 14: Esquema eletrônico do projeto

Analisando o sistema, primeiro tem-se que se entender o funcionamento de um transistor, que é um componente com três terminais: coletor (conectado ao fio cinza), emissor (negativo do sistema) e base (ligado ao resistor 680 Ω), onde a corrente flui entre os dois primeiro, controlado pelo sinal elétrico do último terminal, base. Quando alguém aciona a porta da saída digital, ela manda um sinal +5v que ativa o sistema representado pela fase 1. Ativado essa fase o transistor libera a passagem da corrente negativa para o terminal coletor.

O transistor Bc548 é um transistor popular e baixa potência que permite uma corrente máxima de 0,1A e voltagem máxima de 30v (suficientes para o projeto). Para sua ativação com a base +5v, necessita de uma resistor de 680 Ω .

Na fase 2, após ativação da fase 1, é habilitado uma corrente negativa da fonte externa que então ativa o relé (componente eletromecânico capaz de controlar circuitos externos de grandes corrente a partir de pequenas corrente ou tensões).

Na fase 3, com o positivo da fonte (+12v) e recebendo da fase anterior o sinal negativo da fonte, o relé “arma” o circuito 4. Nesse setor é observado a representação de um diodo 1N4007, este tem função de inibir que a corrente +12v passe para o coletor de transistor.

Na última fase, após relé acionado, a corrente +12v passa para o terminal conectado ao dispositivo, ativando-o.

No projeto, esses dispositivos são representados pela ventoinha e pelo neon, por necessitar potências maiores.

No caso de led, não necessita de um relé por ele ter baixa potência. Não é necessário um diodo, pois ele próprio é um diodo impossibilitando o retorno de uma corrente da fonte externa.

4.2 Software utilizado no projeto

Partindo para o software do projeto, denominado por open source, ou seja, possui o código fonte aberto, disponível. Logo o desenvolvedor da placa *Stargate-888*, disponibiliza a software para funcionamento da placa, deixando o direito do usuário modificar, adaptar para sua utilização (não comercial).

O programa é codificado na linguagem C++, utilizando o compilador Visual C++ 2010 express, disponível gratuitamente. Precisa-se também da instalação do Windows Driver kit (WDK), pois o projeto utiliza alguns de seus diretórios. Após instalação do Visual C++ 2010 express e do WDK, configura-se o projeto *Stargate_app* com inclusão de alguns diretórios do WDK, possibilitando compilar e criar o executável do programa.

Como o projeto a princípio é genérico, faz necessário a modificação para adequação ao projeto, então fez-se a alteração do código fonte nos itens de utilização e os demais permaneceram inalterado (no caso do projeto, mas poderia ter sido ocultado). Observe a ilustração nº 15 que é interface de comunicação do projeto.

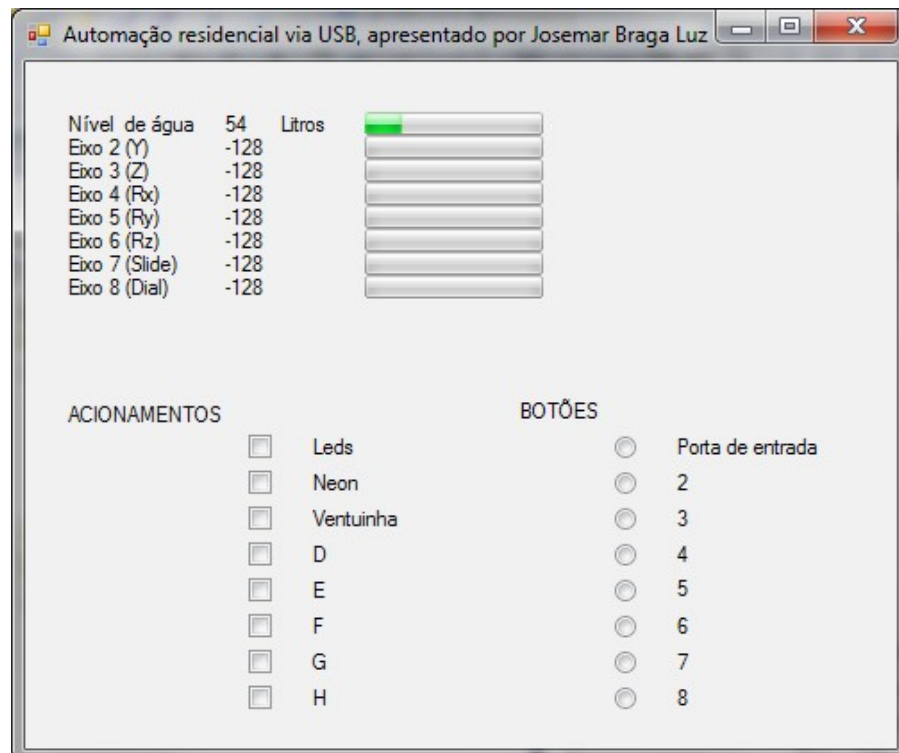


Ilustração 15: Interface de comunicação do projeto

De acordo com a interface mostrada, podem observar, na parte superior estão representados as entradas analógicas. No projeto, representado por “Nível de água”, esta entrada possui um potenciômetro que atua similar a uma boia comum em caixas d’água, retornando um valor entre -127 e 127 que automaticamente é convertido em um valor em litros (na imagem acima 54 Litros), dependendo da caixa d’água o valor altera. Logo exibirá a litragem aproximada juntamente com preenchimento gráfico ao lado. As demais saídas podem ser utilizadas de diversas formas, por não está sendo utilizadas elas estão aterradas exibindo o valor -128.

Na parte inferior esquerda estão representados os acionamentos, no projeto configurado por leds, neon e ventoinha. Com um simples clique em suas caixas, a placa aciona a saída ativando uma corrente positiva (+5v) transformada por meios eletrônicos já

apresentados anteriormente(diagrama esquemático do projeto) em um sinal +12v, capaz de ligar qualquer um dos exemplos atribuídos ao projeto. Para desligar, basta outro clique na caixa. Os demais acionamentos (D,E,F,G e H), não foram utilizadas.

Por fim, na parte inferior direita estão as entradas digitais. No projeto tem a demonstração de uma porta de entrada que possui uma chave *push-button*, que simula a porta aberta ou fechada, ilustrando um ponto verde na caixa representante da porta.

Segue adiante a imagem do projeto em funcionamento. Observe a ilustração nº16 o projeto físico com janela de comando.

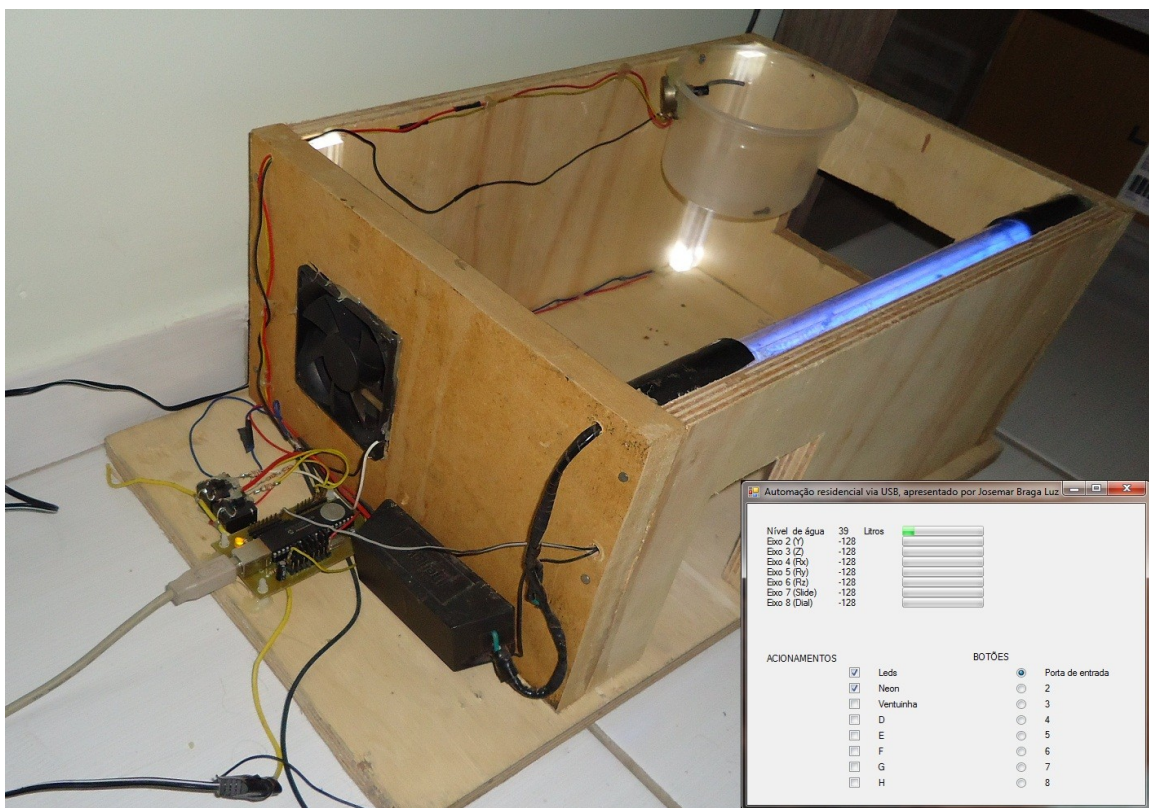


Ilustração 16: Projeto físico com janela de comando.

Acima pode-se ver pela janela de comando, que o nível da caixa de água apresenta 39 litros, os leds (duas iluminações brancas na imagem) e neon (iluminação azul) estão acionadas, ventoinha desligada e chave da porta, também acionada. Esta tem sua localização representado na ilustração nº17.



Ilustração 17: Localização da chave push button (entrada digital)

4.3 Considerações Finais

Chega-se aqui com projeto concretizado com sucesso, funcionando, mas para chegar a tal fim passou-se por vários testes, erros e acertos.

Um dos primeiros teste realizado, foi a interação da placa com o software, após toda a configuração do programa com seus diretórios externos (WDK), funcionou bem, mas por algum motivo, descuido, o programa alterou essa configuração, então veio a primeira dificuldade, pois agora o programa apresentava erro na compilação. Depois de alguns dias testando identificou-se uma alteração no caminho dos diretórios externos, apenas um número causou esse impasse (onde estava 1 era para esta 0). Pronto agora pode-se fazer alteração e criar o executável desejado para o projeto.

A cada alteração do projeto era feito um teste, apesar de programa esta pronto (genérico) necessitou um entendimento de vários trecho do software, para sua alteração e compilação correta até chegar na forma desejada.

Outra fase, foi o teste das saídas e entradas da placa, fase que com auxílio do multímetro passou-se com tranquilidade, teve um pouco de ansiedade com medo de algum erro e queimar da placa, mas deu certo.

A construção da maquete, que na verdade foi um dos primeiros passos, foi a parte mais simples, só teve dificuldade na busca do material.

Por fim, com tudo realizado fica a gratificação de projeto concretizado.

CAPÍTULO 5 – CONCLUSÃO

Este trabalho teve por objetivo a automação residencial com uso da placa *stargate 888* via porta USB. Para chegar ao ponto de concretizar o objetivos várias etapas foram realizadas e conquistadas.

Na primeira fase foi feita uma descrição do que o projeto iria executar, enumerado os objetivos e as atividades planejadas para implantação do projeto.

Na segunda fase buscou-se as informações e material necessários para implantação, assim como a viabilidade. Nessa fase fez estudo da tecnologia USB, *Open Source hardware*, *stargate 888* e circuitos eletrônicos como também leitura de materiais e projetos similares a esse.

A terceira fase caracterizou pela elaboração do projeto como todo. A fase mais demorada por tratar da construção física juntamente com software, onde as maiores dificuldade apareceram e a busca por conhecimento através da internet e professores teve um papel muito importante para realização do projeto. Dividindo essa fase tem-se: a construção da maquete com todos os dispositivos a serem mostrados, a elaboração dos circuitos eletrônicos necessários para funcionalidade e, por fim, a adaptação do software para corresponder ao apresentado na maquete.

Na quarta fase, após conseguir realizar o projeto, foi caracterizada pela documentação total do trabalho, uma escrita que buscou a forma mais clara e objetiva de descrever todo o conhecimento adquirido.

A maioria dos objetivos, mostrados na primeira seção deste documento, foram alcançados. Esse projeto poderá posteriormente atender maiores perspectivas e funcionalidades.

Como trabalhos futuros, pode-se:

- Aplicar mais dispositivos as suas saídas e entradas.
- Implementar o software conectando-o a rede, possibilitando controle do projeto de qualquer lugar.
- Fazer junção com sistema de monitoramento via vídeos.
- Aplicar tecnologia bluetooth.

BIBLIOGRAFIAS

- Revista Lacombe, 2004.
- PRUDENTE, Francesco, Automação Predial e residencial, 2011.
- BOLZANI, Caio Augustus M., Residências Inteligentes - Um Curso De Domotica, Redes Domesticas e Automacao Residencial, 2004.
- ROSÁRIO, J.M., Princípios da Mecatrônica, 2005.
- PEREIRA, Fábio, Microcontroladores PIC: Programação em C, 2003.
- MORENO, Edward David. PENTEADO, Cesar Giacomini e CESAR, Alexandre, MICROCONTROLADORES E FPGAS: APLICAÇÕES EM AUTOMAÇÃO, 2005.
- MIYADAIRA, Alberto Noboru, Microcontroladores PIC18: Aprenda e Programe em linguagem C, 2009.
- SOUZA, Vitor Amadeu, Programação em C para PIC18F4550, 2011.
- VIANA, Pablo. www.usbinterface.com.br, 2010.
- CURTIS, Keith Cary, "After the Software Wars", 2009.
- MESSIAS, A .F . Edifícios “inteligentes”: a domótica aplicada a realidade brasileira. 2007.
- ALECRIM, Emerson. Tecnologia USB (Universal Serial Bus). Infowester, 31 mar. 2010. Disponível em: <<http://www.infowester.com/usb.php>>. Acesso em: 12 nov. 2010;
- AGUIAR, Gustavo. USB (*Universal Serial Bus* - Barramento Serial Universal),2007. Disponível em: <<http://www.portalgeobrasil.org/colab/artigos/usb.pdf>>. Acesso em: 21 nov.2010.
- ANDRADE, Daniel. Disponível em: <www.danielandrade.net/2011/02/10/open-hardware-definition-1-0/>.2011;
- TORRONE, Phillip. Hardware de Código Aberto 2008. Disponível em: <<http://blog.makezine.com/2008/11/25/draft-open-source-hardwa/>>. 2008.
- VEGANO. BeagleBoard – o menor computador do mundo. Disponível em: <<http://www.pensandonisso.com/page/49/>>.2008.

Informações complementares

USB:

- www.usb.org.
- Jan Axelson, "USB Complete", www.lvr.com.
- TAN, W. M., Developing USB PC Peripherals, 1ª edição, Annabooks, 1997, ISBN 0-929392-38-8;
- Everything USB... We Mean Everything! - www.everythingusb.com;

Stargate-888:

- www.usbinterface.com.br.

Open Source Hardware:

- www.openhardware.de/

Arduino:

- www.arduino.cc;

Microchip:

- www.microchip.com/