



TRABALHO DE CONCLUSÃO DE CURSO

**TRANSMISSÃO DE ÁUDIO SEM FIO POR
TECNOLOGIA *BLUETOOTH***

Alex Chaves Rocha Lima
Guilherme Lima Gonçalves

Brasília, Dezembro de 2009

UNIVERSIDADE DE BRASÍLIA

FACULDADE DE TECNOLOGIA

UNIVERSIDADE DE BRASÍLIA

Faculdade de Tecnologia

TRABALHO DE CONCLUSÃO DE CURSO

TRANSMISSÃO DE ÁUDIO SEM FIO POR TECNOLOGIA *BLUETOOTH*

Alex Chaves Rocha Lima

Guilherme Lima Gonçalves

Relatório submetido como requisito parcial para obtenção
do grau de Engenheiro Eletricista

Banca Examinadora

Prof. Alexandre Romariz, UnB/ENE (Orientador)

Prof. Janaína Guimarães, UnB/ENE

Prof. Ricardo Zelenovsky, UnB/ENE

DEDICATÓRIAS

Dedico este trabalho a minha família que sempre esteve do meu lado durante toda esta trajetória. Muito obrigado por todos os momentos de compreensão e apoio. Sem vocês eu não chegaria até aqui. Aos amigos que fizeram parte desta jornada o meu muito obrigado, vocês fazem parte desta conquista.

Guilherme Gonçalves

Este trabalho é dedicado à minha família, que deu todo o suporte para que ele pudesse ser realizado e demonstrou enorme compreensão quando eu precisei me ausentar nesta reta final. Aos amigos que, além do apoio oferecido, me obrigaram a parar para relaxar algumas vezes ao longo do semestre, uma menção especial.

Alex Lima

AGRADECIMENTO

Queríamos agradecer, primeiramente, a Deus. Ao Eng. Claudio Carnevali da equipe de suporte da Eikon que sanou uma quantidade enorme de dúvidas surgidas ao longo do desenvolvimento do projeto, um muito obrigado. Gostaríamos de agradecer, também, ao Felipe, ao Cícero e ao Célio que fazem parte da equipe técnica do SG11 e que nos pouparam horas de trabalho auxiliando, sempre com boa vontade, nas tarefas de montagem final. Agradecemos ao Prof. Camargo e a todos os companheiros do LPCI que nos acolheram e compartilharam parte do seu espaço quando precisamos. Por último, um agradecimento especial a todos os professores que participaram com idéias, soluções, experiências e que entendem a dimensão que este trabalho toma em nossas vidas e são compreensivos com as conseqüentes faltas, atrasos e rendimento mediano em suas disciplinas.

RESUMO

Este trabalho apresenta um estudo sucinto sobre a tecnologia *Bluetooth* e as etapas do desenvolvimento do protótipo de um transmissor de áudio de alta qualidade sem fio utilizando um módulo *Bluemore110*.

Palavras-Chave: *Bluetooth. Sem fio. Áudio. Bluemore110.*

ABSTRACT

This work presents a concise study on Bluetooth technology and the prototype development phases of a high-quality audio wireless transmitter using a *Bluemore110* module.

Keywords: *Bluetooth. Wireless. Audio. Bluemore110.*

SUMÁRIO

1. INTRODUÇÃO	1
1.1. MOTIVAÇÃO	1
1.2. HISTÓRICO	2
2. CONSIDERAÇÕES TEÓRICAS	6
2.1. TÉCNICA DE ESPALHAMENTO ESPECTRAL	6
2.2. TÉCNICA DE MODULAÇÃO	7
2.3. SUBDIVISÃO INTERNA DE PROTOCOLOS	8
2.4. HIERARQUIA DE PEQUENAS REDES (PICONETS)	8
3. DESENVOLVIMENTO DO PROJETO	11
3.1. O MÓDULO BLUETOOTH	11
3.2. A PLACA DE DESENVOLVIMENTO	13
3.3. ATUALIZAÇÕES DE <i>FIRMWARE</i>	14
3.4. IMPLEMENTAÇÃO	16
3.5. DESENVOLVIMENTO DO CIRCUITO	17
4. RESULTADOS	22
4.1. CARACTERÍSTICAS FINAIS	22
4.2. ANÁLISE DE CUSTO	23
4.3. TESTES PARALELOS	24
4.3.1. HEADSET	Erro! Indicador não definido.
4.3.2. A2DP (<i>Advanced Audio Distribution Profile</i>)	25
4.3.3. COMUNICAÇÃO DE DADOS	25
5. CONSIDERAÇÕES FINAIS	26
REFERÊNCIAS	27

LISTA DE FIGURAS

Figura 1 – Esquema ilustrativo de <i>frequency hopping</i> . [3].....	6
Figura 2 – Esquema de modulação 2FSK. [3].....	8
Figura 3 – Funcionamento do filtro gaussiano.[3].....	8
Figura 4 – Fluxograma da transição de estados. [7].....	10
Figura 5 – O módulo <i>Bluetooth</i>	11
Figura 6 – A placa de desenvolvimento.....	13
Figura 7 – Interface do programa para atualização de <i>firmware</i>	14
Figura 8 – Circuito de acionamento de botões.....	18
Figura 9 – Circuito de acionamento de LEDs.	18
Figura 10 – Esquemático do circuito base do LM317. [10].....	19
Figura 11 – Circuito regulador de tensão.	19
Figura 12 – Circuito de testes na <i>protoboard</i>	20
Figura 13 – Circuito de transmissão.....	20
Figura 14 – Circuito de recepção.....	21
Figura 15 – Foto do protótipo das placas pronto.....	21
Figura 16 – Circuito base do LM7805. [11].....	23
Figura 17 – Bloco de alimentação inserido no circuito.....	23

LISTA DE TABELAS

Tabela 1 – Evolução das versões do <i>Bluetooth</i> . [1] [2].....	4
Tabela 2 – Classes de dispositivos <i>Bluetooth</i> . [5].....	5
Tabela 3 – Estados de operação de um dispositivo em uma <i>piconet</i> . [7].....	9
Tabela 4 – Descrição dos pinos de JF1.	12
Tabela 5 – Descrição dos pinos de JF2.	12
Tabela 6 – Listagem de <i>firmwares</i> disponibilizados pelo fabricante.....	15
Tabela 7 – Função dos pinos programáveis para o <i>firmware Wireless PTT</i>	16
Tabela 8 – Tabela de custos aproximados para o desenvolvimento do dispositivo.....	23
Tabela 9 – Entradas e saídas utilizadas no <i>firmware Wireless Headset</i> . [12].....	24

1. INTRODUÇÃO

Neste capítulo, serão apresentadas as razões que nos levaram à concepção da idéia e ao desenvolvimento deste trabalho. Adicionalmente, apresentaremos um breve histórico da tecnologia *Bluetooth* e apresentaremos uma sucinta ordem cronológica de eventos.

1.1. MOTIVAÇÃO

O objetivo deste projeto é que, ao final do desenvolvimento, tenhamos um par de dispositivos que sejam capazes de, utilizando a tecnologia *Bluetooth*, fazer a comunicação entre uma fonte, cujo sinal esteja dentro da faixa de frequências audíveis, e um receptor, que seja capaz de reproduzir tais sinais. Para exemplificar, um microfone ligado ao transmissor poderia transmitir, sem a utilização de cabos, um sinal de voz para um alto-falante que esteja conectado ao receptor. Dessa maneira, esperamos eliminar os fios que conectam qualquer dispositivo de áudio ao seu alto-falante.

Eliminar cabos e conexões é uma das maiores tendências na modernização dos equipamentos presentes em nosso cotidiano. Entre as diversas tecnologias disponíveis no mercado, que desempenham este papel, temos o *Wi-Fi*, o *Bluetooth*, o IrDA (*Infrared Data Association*) e a Radiofrequência (RF) na faixa de FM. Apesar de todas terem a capacidade de transmitir dados sem utilizar fios, cada uma delas possui características particulares e, por isso, são mais ou menos indicadas dependendo da aplicação desejada.

Encontra-se aplicação para a transmissão *wireless* de áudio em várias situações do nosso dia-a-dia. Quando pensamos em um sistema de *home-theater* que possui caixas de som espalhadas por todo um ambiente, a idéia de não ter uma quantidade enorme de cabos espalhados parece bastante conveniente. Se pensarmos em um sistema de som para a área externa de uma casa, a ausência de fios proporciona mobilidade às caixas de som. Da mesma maneira, instrumentistas poderiam se movimentar livremente no palco se não precisarem de fios para conectar seus instrumentos musicais aos alto-falantes.

Algumas destas aplicações são, hoje, supridas por dispositivos que trabalham com Radiofrequência em FM. Trata-se de um circuito muito simples com baixo consumo de energia e uma enorme gama de aplicações. Entretanto, é um sistema que está extremamente sujeito a interferências a julgar pela quantidade de dispositivos que operam nessa faixa de frequência. Dito isso, e levando em consideração o desejo de inovar, optamos por utilizar outra tecnologia entre aquelas citadas e dar a ela esta nova aplicação.

Além do inegável apelo comercial do *Bluetooth*, que está presente em uma enorme parcela dos dispositivos mais modernos, a escolha de tal tecnologia para o desenvolvimento deste projeto foi fruto de uma comparação das vantagens e desvantagens de cada uma das tecnologias citadas anteriormente. Quando comparado ao *Bluetooth*, o *Wi-Fi* possui um custo de implementação cerca de três vezes maior, consome cerca de cinco vezes mais e o projeto envolvendo esta tecnologia é muito mais complexo. O IrDA, para a transmissão de dados, depende do alinhamento de um feixe cônico cujo ângulo de abertura tem cerca de 30° e possui um alcance máximo de 1 metro enquanto o *Bluetooth* conta com uma transmissão onidirecional com alcance aproximado de 100 metros nos dispositivos mais potentes. [2]

Em última análise, o *Bluetooth* é uma tecnologia em alta que desperta curiosidade nas pessoas e sempre nos chamou a atenção. Além disso, atende bem ao propósito do nosso projeto e possibilita o desenvolvimento de dispositivos robustos e capazes de transmitir áudio de ótima qualidade. O baixo consumo de energia e o alcance elevado irão conferir ao dispositivo uma grande mobilidade.

1.2. HISTÓRICO

Em 1998, já iniciando a era das redes sem fio, surgiu a necessidade de se desenvolver uma maneira rápida e eficiente de conexão para curtas distâncias. Isso porque a interconexão entre os aparelhos pessoais que surgiam diariamente tornava-se cada vez mais complicada. Os aparelhos mais antigos utilizavam cabos RS232 e cada fabricante utilizava um padrão na configuração dos pinos, o que tornava necessário um cabo para cada aparelho. A idéia era criar uma conexão rápida e segura sem precisar de uma estrutura de provedor. Para resolver este problema, a Ericsson (maior fabricante de celulares naquela época), IBM, Intel, Nokia e Toshiba se uniram para formar o *Bluetooth Special Industry Group* (SIG). Este grupo tinha a missão de desenvolver e promover, em todo o mundo, esta nova tecnologia de comunicação sem fio. Escolheu-se, na época, a banda de 2,4 GHz ISM (*Industrial, Scientific, Medical*) para operar, uma vez que, por acordos internacionais, não era preciso licença. [7]

O nome surgiu de uma analogia ao sobrenome do rei dinamarquês Harald Blatand que, em inglês, significa Bluetooth e ficou conhecido por unir tribos norueguesas, suecas e dinamarquesas durante o Século X. Como o objetivo do *Bluetooth* era interligar equipamentos pessoais e novas tecnologias, foi feita esta referência. O nome era para ser provisório, porém teve uma grande aceitação e foi mantido. O logotipo nasceu da união de duas letras características do alfabeto dinamarquês do século X, Hagall e Berkanan correspondentes às letras H e B no alfabeto latino. [5]

Para acelerar a divulgação da nova tecnologia no mercado, o SIG disponibilizou a especificação *Bluetooth*, sem o pagamento de *royalties*, para empresas que fizessem parte de um grupo de interesse. Mas estas empresas só poderiam utilizar o *Bluetooth* com o objetivo de divulgá-lo no mercado. Logo que anunciou esta proposta, o SIG foi procurado por diversas empresas. Em poucos meses, atingiu a marca de 400 integrantes. E no primeiro semestre de 1999, foi disponibilizada a primeira versão da especificação para os integrantes do grupo. O público em geral só teve acesso mais tarde. [7]

Todos os produtos criados por estas empresas tinham que passar por um teste de qualidade. Estes testes eram feitos pelo BQP (*Bluetooth Qualification Program*) e neles verificava-se itens como conexão e transmissão, entre outros. Em março de 1999, o grupo de trabalho IEEE 802.15 foi criado para definir padrões de redes pessoais de comunicação sem fio. O *Bluetooth* SIG apresentou a proposta da recém criada tecnologia e foi aceito como o padrão 802.15.1 para as WPANs (*Wireless Personal Area Networks*). Em dezembro de 1999, as empresas 3Com Corporation, Lucent (substituída pela Agere em 2001), Microsoft e Motorola entraram para o SIG. [7]

Após 11 anos do lançamento da tecnologia *Bluetooth*, várias versões já foram desenvolvidas. A cada nova versão, são aprimorados itens como segurança e velocidade de transmissão, facilidade de conexão, economia de energia, alcance e muitos outros. Assim como as versões foram se atualizando, novos aparelhos foram sendo equipados com a tecnologia ao longo dos anos. Hoje em dia, não se imagina um equipamento pessoal sem a presença da tecnologia *Bluetooth*. Além disso, a cada dia que passa, os usuários exigem uma interação entre os dispositivos com mais qualidade. A última versão, lançada em abril deste ano, pode fornecer taxas de transmissão de até 24 Mbps e tem um baixíssimo consumo de energia. [2] Com essa nova conexão é possível transmitir vídeos de um telefone celular para a televisão e interligar computadores com muita eficiência e rapidez. A tabela 1 mostra a evolução das versões e ainda o ano em que a tecnologia *Bluetooth* foi inserida em alguns dispositivos.

Tabela 1 – Evolução das versões do *Bluetooth*. [1] [2]

Ano	Versões – Principais características	Novos produtos
1999	Bluetooth 1.0 – Apresentava problemas de conexão entre os dispositivos.	Computadores, Telefones Celular.
2001	Bluetooth 1.1 - Representa o estabelecimento do <i>Bluetooth</i> como um padrão IEE 802.15. Muitos problemas encontrados na versão anterior foram solucionados.	Impressora, <i>Laptop</i> , <i>Mouse</i> , <i>Headset</i> , <i>Kits</i> para receber chamadas no carro.
2003	Bluetooth 1.2 - Conexões mais rápidas, proteção contra interferências mais eficiente.	MP3 <i>player</i> , Equipamentos médicos, GPS, Câmera Digital.
2004	Bluetooth 2.0 - Diminuição do consumo de energia, aumento na velocidade de transmissão de dados para 3 Mbps, correção às falhas existentes na versão anterior e melhor comunicação entre os dispositivos.	Fone Estéreo, Óculos, Relógio, Rádio-relógio.
2007	Bluetooth 2.1 - Permite uma melhor seleção dos dispositivos antes de estabelecer uma conexão, melhorias nos procedimentos de segurança e melhor gerenciamento do consumo de energia.	Televisão, Porta-retrato.
2009	Bluetooth 3.0 - Tem como principal atrativo as altas taxas de velocidade de transmissão de dados. Dispositivos compatíveis podem atingir a taxa de 24 Mbps. Controle eficiente do consumo de energia.	Eletrrodomésticos.

Como observado na tabela, qualquer tipo de equipamento pode e está recebendo a tecnologia *Bluetooth*. Não é por acaso que, atualmente, já existem mais de dois bilhões de *chips* embutidos em equipamentos pelo mundo. Um ponto chave no desenvolvimento de novas versões é a compatibilidade com as versões anteriores. A conexão entre dois dispositivos com versões diferentes ocorre normalmente, porém fica limitada às características daquela mais antiga. O *Bluetooth* SIG disponibiliza em seu *site* uma pesquisa de compatibilidade entre dispositivos. O usuário coloca a marca ou modelo do seu equipamento no campo de pesquisa e o banco de dados retorna a lista de tipos de dispositivo com os quais seu equipamento pode se conectar e seus respectivos fabricantes.

Os módulos *Bluetooth* ainda são divididos em classes de acordo com a potência de transmissão (Tabela 2). Isto acontece para que nenhum dispositivo conte com um *chip* que ofereça uma capacidade maior que aquela necessária. Dessa maneira, os equipamentos que utilizam baixa potência, não precisam de um módulo com alta capacidade que tornaria o produto muito mais caro. Por exemplo, um fone de ouvido *Bluetooth* para celular não precisa ter 100 metros de alcance. [1]

Tabela 2 – Classes de dispositivos *Bluetooth*. [5]

Classe	Potência máxima permitida [mW]	Alcance aproximado [m]
1	100	100
2	2,5	10
3	1	1

Apesar de sua curta existência, o *Bluetooth* faz parte da maioria dos aparelhos pessoais mais modernos. No contexto da progressiva eliminação dos fios para comunicação, essa é uma tecnologia que ainda tem muito a se desenvolver e, com certeza, estará presente no dia-a-dia das pessoas por um bom tempo.

2. CONSIDERAÇÕES TEÓRICAS

A seguir, serão apresentados alguns conceitos teóricos que julgamos importantes para a compreensão da tecnologia. A maior parte destes conceitos e teorias dará suporte ao entendimento de algumas passagens anteriores e outras que estão por vir.

2.1. TÉCNICA DE ESPALHAMENTO ESPECTRAL

Parte das vantagens do *Bluetooth* sobre as demais tecnologias sem fio é a sua imunidade a ruídos, a imunidade a distorções multipercurso, a possibilidade de utilização da mesma faixa de frequências por vários usuários e a criptografia presente em sua transmissão. Todas estas vantagens podem ser creditadas à técnica de espalhamento espectral FHSS (*Frequency Hopping Spread Spectrum*), que é utilizada pela tecnologia. [3]

Esta técnica consiste em variar rapidamente, de maneira pseudoaleatória, a frequência de transmissão. Cada frequência é utilizada por um curto período de tempo (*dwell time*) que pode ser ajustado e não deve ser superior a 400 ms. No *Bluetooth*, o *dwell time* é de 625 μ s, ou seja, 1600 mudanças de frequência por segundo. [7]

Estas mudanças de frequência são determinadas por um gerador de números pseudoaleatórios, que deve estar presente em cada um dos dispositivos, onde cada número representa uma frequência de transmissão como pode ser visto na figura 1.

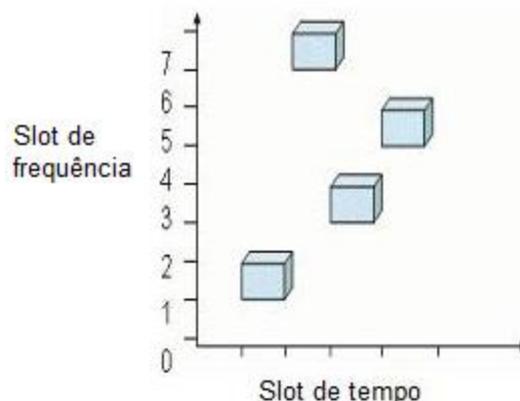


Figura 1 – Esquema ilustrativo de *frequency hopping*. [3]

Daí vem a necessidade dos dispositivos *Bluetooth* serem sincronizados antes da transmissão. Uma vez definida a semente que origina a sequência pseudoaleatória, todos os dispositivos que participarão da transmissão de dados devem conhecê-la. A

comunicação só acontece se os dispositivos envolvidos estiverem mudando de frequência simultaneamente e de forma sincronizada.

Esta particularidade do FHSS permite que vários dispositivos *Bluetooth* utilizem a mesma banda de frequências. Basta, para isso, que a sequência de mudanças seja diferente para cada par que esteja se comunicando.

O Bluetooth, de acordo com uma resolução da FCC (Federal Communications Commission), órgão de regulamentação das telecomunicações nos EUA, utiliza a faixa ISM do espectro. Esta se encontra em 2,4 GHz e está dividida em 79 canais de 1 MHz. Essa resolução determina ainda que o *Bluetooth* utilize, no mínimo, 75 dos 79 canais disponíveis para o FHSS e não permaneça por mais de 400 ms em uma mesma frequência para evitar interferência com outros dispositivos que trabalham na banda ISM. Em geral, os dispositivos Bluetooth utilizam todos os 79 canais disponíveis e a frequência deles pode ser determinada de acordo com a expressão da equação 1. [7]

$$f = (2402 + k) \text{ MHz}, \quad k = 0, 1, \dots, 78.$$

Equação 1 - Frequência dos canais disponíveis para comunicação. [6]

À parte a capacidade de reutilização da banda de frequências, podemos perceber que esta técnica dá certa segurança criptográfica à transmissão, que só pode ser interceptada caso se conheça a semente que gerou a sequência pseudoaleatória.

2.2. TÉCNICA DE MODULAÇÃO

A modulação FSK (*Frequency Shift Keying*) consiste na atribuição de diferentes frequências para a portadora de acordo com o bit a ser transmitido. Na modulação binária 2FSK, a cada bit (0 e 1) é atribuída uma frequência. No entanto, pode-se utilizar uma frequência para cada grupo de bits. No 4FSK, temos uma frequência atribuída para cada conjunto de 2 bits (00, 01, 10 e 11) totalizando quatro frequências diferentes e possibilitando uma taxa de transmissão maior, mas, em contra partida, provoca um aumento na largura de banda. [3]

A variação brusca de frequência faz com que este tipo de modulação apresente grandes bandas. A figura 2 ilustra essa variação na frequência do 2FSK.

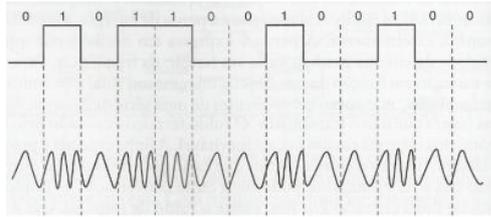


Figura 2 – Esquema de modulação 2FSK. [3]

Para reduzir a largura de banda, o *Bluetooth* utiliza o GFSK (*Gaussian Frequency Shift Keying*) que corresponde à modulação FSK com a introdução de um filtro gaussiano. Tal filtro funciona como um formatador de pulsos, suavizando a transição entre os valores de pico como pode ser observado na figura 3. Tendo reduzido as variações bruscas de frequência, reduz-se significativamente a largura de banda. [3]



Figura 3 – Funcionamento do filtro gaussiano.[3]

O FHSS utiliza a técnica de modulação GFSK, sendo que o 2GFSK permite uma taxa de transmissão de 1 Mbps e o 4GFSK, 2 Mbps.

2.3. SUBDIVISÃO INTERNA DE PROTOCOLOS

A pilha de protocolos do sistema Bluetooth é dividida em duas camadas básicas. Na primeira camada, está o grupo de protocolos de transporte e, na segunda, o grupo denominado *middleware*. [6]

O grupo de transporte engloba protocolos desenvolvidos especificamente para o Bluetooth que cuidam de todas as etapas de comunicação entre dois dispositivos. Na camada *middleware*, existem tanto protocolos exclusivos quanto protocolos adaptados de outras tecnologias. Tais protocolos são usados de maneira seletiva para permitir diversas aplicações, sejam elas recentes ou mais antigas. [6]

2.4. HIERARQUIA DE PEQUENAS REDES (*PICONETS*)

Uma vez estabelecida a conexão, os dispositivos assumem funções diferentes na rede. Cada terminal poderá desempenhar o papel de mestre ou escravo. O mestre é aquele que iniciou a comunicação e gerencia toda a troca de informações. O escravo é o dispositivo

que foi encontrado pelo mestre e está subordinado a ele. A comunicação pode acontecer de forma simultânea (*Full-Duplex*), mas o escravo depende de uma autorização do mestre para enviar dados. À rede formada por um mestre e um escravo, damos o nome de *point-to-point* ou ponto a ponto. Esta é a rede formada por dois telefones celular à qual estamos habituados quando trocamos um arquivo por *Bluetooth*. [7]

Quando dois ou mais escravos estão conectados a um único mestre, forma-se um tipo de rede denominado *piconet*. Ao dispositivo mestre, podem estar conectados até sete escravos ativos. Os escravos só se comunicam entre si com autorização do mestre. Qualquer dispositivo pode desempenhar função de mestre ou escravo dependendo da configuração da rede em que está inserido. [7]

Pode-se exceder número sete de escravos conectados ao mestre em uma *piconet*, mas os excedentes devem estar inativos (*parked*). Cabe ao mestre gerenciar, de maneira conveniente, os dispositivos escravos ativando e desativando-os de modo que estejam ativos no máximo sete de cada vez. Veremos mais adiante que o mestre poderá determinar outros estados além de *parked* aos escravos. [7]

Em uma *piconet*, o mestre determina o padrão de mudança das frequências e o tempo de duração de cada frequência. É dele a função de gerenciar o envio e o recebimento de informações para cada escravo e determinar em qual modo de operação ele deve estar de acordo com a tabela 3. [7]

Tabela 3 – Estados de operação de um dispositivo em uma *piconet*. [7]

Estado	Característica
<i>Standby</i>	Dispositivo está ligado, porém não está conectado em uma <i>piconet</i>
<i>Inquiry</i>	Quando o dispositivo está localizando outro aparelho com que ele possa se conectar.
<i>Page</i>	Estado que o mestre de uma <i>piconet</i> se encontra. Fica enviando mensagens com pedidos de conexão para outros dispositivos.
<i>Page Scan</i>	Estado que o escravo se encontra no momento que recebe um convite para se conectar a uma <i>piconet</i> .
<i>Connected</i>	É o estado do novo dispositivo quando a conexão é estabelecida com o mestre. A partir desse momento ele recebe um endereço ativo.
<i>Transmit</i>	Estado que o escravo entra quando recebe a autorização do mestre para transmitir. Após a transmissão volta para o <i>Connected</i> ou finaliza a conexão.
<i>Sniff</i>	Estado de economia de energia em que o dispositivo está “dormindo” por um pré-determinado número de <i>slots</i> . Ele sai deste modo no <i>slot</i> determinado pelo mestre, transmite informações e volta para o mesmo estado.
<i>Hold</i>	É o estado que o dispositivo não está ativo por um determinado período de tempo, consome pouca energia e não pode transmitir.
<i>Park</i>	Quando o dispositivo não tem informações para receber nem para enviar o

mestre o coloca nesse estado. O endereço ativo é perdido e fornecido pelo mestre a outro dispositivo.

O diagrama de blocos da figura 4 mostra como podem ser feitas as transições entre os estados descritos acima.

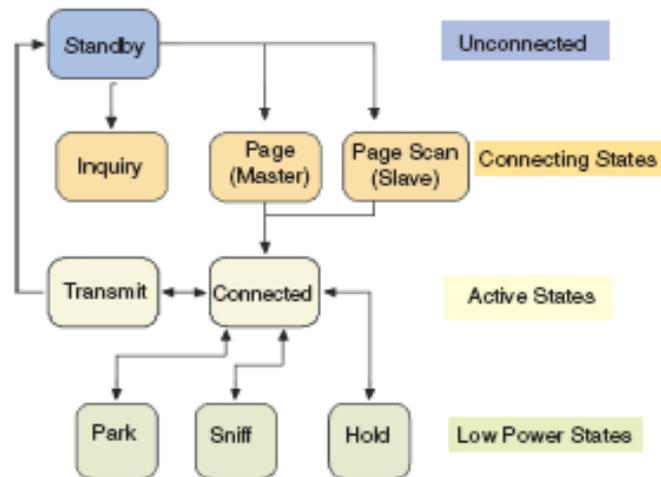


Figura 4 – Fluxograma da transição de estados. [7]

Existe a possibilidade de fazer ainda a conexão entre duas ou mais *piconets*. A essa estrutura, damos o nome de *scatternet*. Para realizar a interconexão entre duas *piconets*, um dispositivo funciona como ponte (*bridge*). Este dispositivo pode ser escravo em ambas as *piconets* ou mestre em uma das duas. [7]

A combinação de várias pequenas redes pode gerar grandes teias com topologias complexas. De sorte que várias aplicações envolvendo grandes redes ainda estão por vir.

3. DESENVOLVIMENTO DO PROJETO

Para a realização do projeto, foi adquirido um *kit* Eval-Bluemore110 do fabricante italiano Eikon (www.eikonsite.it). O kit é composto por uma placa de desenvolvimento e um módulo Bluemore110. Era necessário ainda um módulo adicional que nos foi gentilmente cedido pelo fabricante como um projeto de apoio à pesquisa não-comercial.

3.1. O MÓDULO BLUETOOTH

O Bluemore110 (figura 5) é um dispositivo de classe 2 equipado com a versão 1.2 de especificações do *Bluetooth*. Desta maneira, a potência máxima do dispositivo é de 2.5 mW (4 dBm), o alcance nominal chega a aproximadamente 20 metros em campo aberto e a taxa de transmissão é de aproximadamente 1 Mbps.

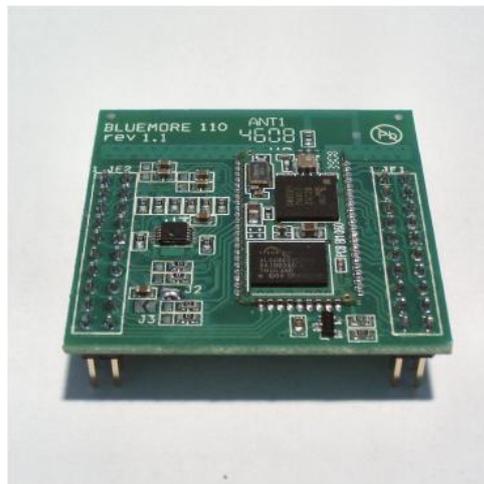


Figura 5 – O módulo *Bluetooth*.

O módulo tem aproximadamente 40x41 mm, um regulador interno de tensão que permite alimentação com 5 ou 3.3 V, uma antena do tipo “F” invertido com suporte para antena externa e possui 40 conectores divididos em 2 *pinstrips* de 20 pinos cada.

O módulo possui pinos com função fixa e pinos programáveis. A tabela 4 corresponde ao *pinstrip* da direita (JF1). Nele temos os pinos de alimentação, os pinos de comunicação via porta serial e as entradas e saídas programáveis.

Tabela 4 – Descrição dos pinos de JF1.

Pino	Nome	Entrada/Saída	Descrição
1	GND	Entrada	Terra.
2	RES	Entrada	<i>Reset</i> , ativo baixo.
3	+5V	Entrada	Fonte de alimentação de 5 V.
5	+3.3V	Saída	Saída do regulador de tensão interno 3.3 V. Corrente máxima de 500 mA.
7	MISO	SPI (<i>Serial Port Interface</i>)	Para atualização de <i>firmware</i> .
9	CSB		
11	CLK		
13	MOSI		
Demais Pinos	PIO_#	Entrada/Saída	Entradas e saídas digitais programáveis.

No *pinstrip* da esquerda (JF2), estão as entradas e saídas de áudio e os pinos para comunicação UART (*Universal Asynchronous Receiver/Transmitter*) como mostra a tabela 5. O dispositivo conta com entradas e saídas de áudio diferenciais para utilização de um amplificador externo e, ainda, saídas e entradas pré-amplificadas.

Tabela 5 – Descrição dos pinos de JF2.

Pino	Nome	Entrada/Saída	Descrição
1	Out_L+	Saída	Saída de áudio diferencial
2	Sx_Out	Saída	Saída de áudio 1 Vpp
3	Out_L-	Saída	Saída de áudio diferencial
4	GND	Entrada	Terra do áudio
5	Out_R-	Saída	Saída de áudio diferencial
6	Dx_Out	Saída	Saída de áudio 1 Vpp
7	Out_R+	Saída	Saída de áudio diferencial
8	Sx_In2	Entrada	Entrada de áudio 1 Vpp
9	In_L+	Entrada	Entrada de áudio diferencial
10	Sx_In1	Entrada	Entrada de áudio 0.4 Vpp
11	In_L-	Entrada	Entrada de áudio diferencial
12	GND	Entrada	Terra do áudio
13	In_R-	Entrada	Entrada de áudio diferencial
14	Dx_In1	Entrada	Entrada de áudio 0.4 Vpp
15	In_R+	Entrada	Entrada de áudio diferencial
16	Dx_In2	Entrada	Entrada de áudio 1 Vpp
17	UART_Tx	Saída	Sinal Tx UART
18	UART_Rts	Saída	Sinal Rts UART
19	UART_Rx	Entrada	Sinal Rx UART
20	UART_Cts	Entrada	Sinal Cts UART

3.2. A PLACA DE DESENVOLVIMENTO

A placa de desenvolvimento possui toda uma estrutura pronta para a realização de testes e configuração do módulo a ela conectado como mostra a figura 6. Ela possui uma entrada e uma saída de áudio, 12 entradas/saídas que podem ser configuradas como botão (entrada) ou LED (saída), uma entrada para alimentação em 5 V e interfaces do tipo RS232 e porta serial para comunicação com computador.

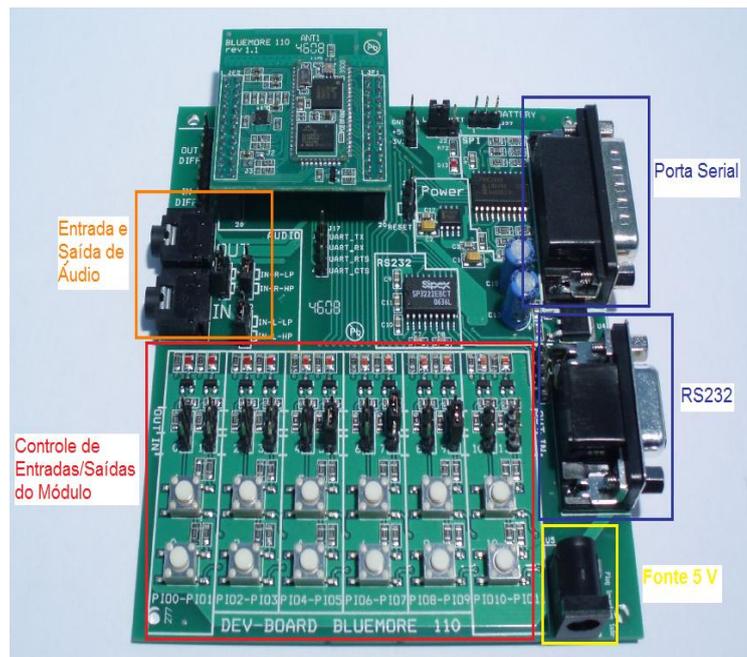


Figura 6 – A placa de desenvolvimento.

Com esta placa temos mais liberdade para realizar testes e aprender a utilizar novas funções. Como os circuitos já estão prontos, não dependemos de uma *protoboard* e elementos externos para testar seu funcionamento. A única operação que devemos executar é a troca do *firmware* e a escolher saídas, entradas e pinos utilizados.

Na placa de desenvolvimento temos a opção de habilitar ou desabilitar entradas e saídas, podendo montar vários tipos de circuitos. Cada pino pode ser configurado como entrada ou saída e esta opção é feita com um *jump* que fecha o circuito escolhido. Se o pino não for utilizado, este *jump* não é colocado e ambos os circuitos ficam abertos.

3.3. ATUALIZAÇÕES DE *FIRMWARE*

Para que o módulo funcionasse conforme desejado, foi necessária a atualização do *firmware*. Utilizamos, para este fim, o *software* Blue Flash, cedido pelo fabricante.

Com o módulo acoplado à placa de desenvolvimento, a atualização do *firmware* funciona de maneira simples e direta. Basta, para isso, executar o *software* com a placa de desenvolvimento conectada ao computador via porta paralela. Uma vez estabelecida a conexão, escolhe-se o arquivo referente ao *firmware* desejado e faz-se a atualização. A figura 7 mostra a interface bem simples do programa com suas funções.

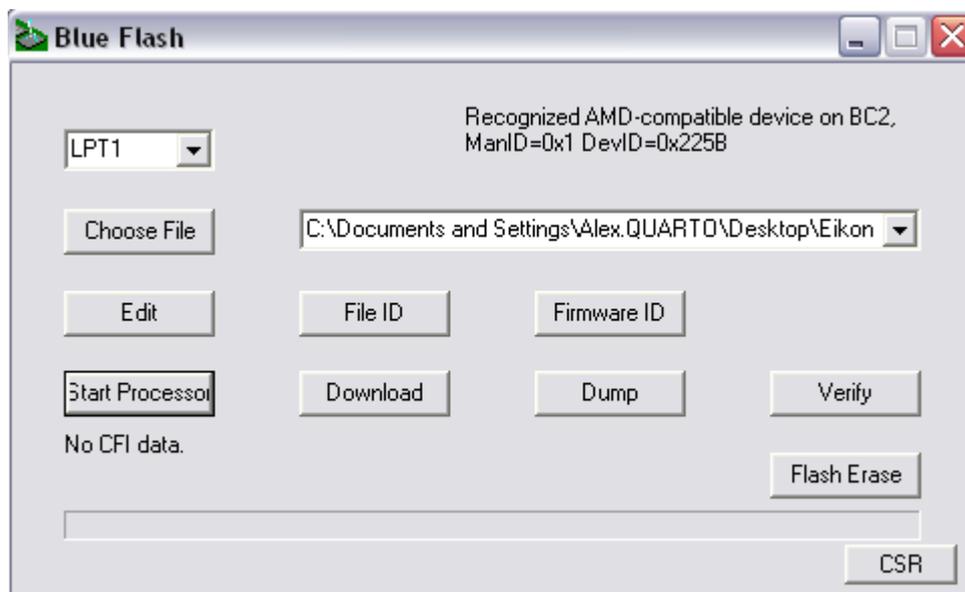


Figura 7 – Interface do programa para atualização de *firmware*.

Apesar de o módulo ser programável, a ferramenta utilizada para o desenvolvimento de novos *firmwares* é bem específica. Segundo o fabricante, teríamos que usar o *software* Blue Lab desenvolvido pela CSR (www.csr.com) que custa aproximadamente 3 mil Euros. Ainda assim, o engenheiro da Eikon Claudio Carnevali, que nos deu o suporte, informou que a equipe de desenvolvimento demorou cerca de oito meses para produzir resultados significativos. Conseqüentemente, optamos por utilizar um dos *firmwares* que o fabricante coloca à nossa disposição.

O fabricante disponibiliza quatro versões pré-definidas de *firmware* que conferem ao módulo uma aplicação particular cada um. Os *firmwares* disponíveis para o Bluemore110 estão listados na tabela 6.

Tabela 6 – Listagem de *firmwares* disponibilizados pelo fabricante.

Firmware	Aplicação
<i>Wireless Headset</i>	Conecta-se a qualquer aparelho que transmita, via <i>Bluetooth</i> , um sinal de áudio tais como celulares e computadores.
<i>Serial Port Profile</i>	Suprime o uso de um cabo serial para conexão entre dois dispositivos tornando-a <i>wireless</i> .
<i>Wireless PTT</i>	Permite o desenvolvimento de um dispositivo <i>push to talk</i> para comunicação sem fios usando <i>Bluetooth</i> .
<i>Wireless A2DP</i>	Quando conectado a uma fonte de áudio, transforma-o em um <i>gateway Bluetooth</i> cujo sinal pode ser captado por dispositivos próprios para este tipo de conexão tais como fones de ouvido igualmente equipados com a tecnologia.

Para o desenvolvimento do dispositivo, escolhemos utilizar o *firmware Wireless PTT* que pode desempenhar a função que queremos e possui um esquema de pareamento simples e confiável.

Utilizando este *firmware*, os módulos se conectam automaticamente ao ligar, uma vez que já tenham sido pareados antes. Ao se conectarem, começam imediatamente a transmitir áudio de alta qualidade. Ele dá ao módulo controle digital de volume, função *push to talk* e saídas para controlar elementos externos como amplificadores e rádios que façam transmissão do tipo *broadcast*.

Parte destes atributos não será aproveitada porque nosso projeto não requer a função *push to talk* e nem o controle de elementos externos.

3.4. IMPLEMENTAÇÃO

Após a atualização do *firmware*, os pinos programáveis, citados na tabela 4, assumem funções específicas de acordo com a tabela 7 abaixo.

Tabela 7 – Função dos pinos programáveis para o *firmware Wireless PTT*.

Pino	Nome	Entrada/Saída	Descrição
4	PIO_8_Out1	Saída	Saída genérica
6	PIO_9_In1	Entrada	Entrada genérica
8	PIO_10_LED1	Saída	LED 1
10	PIO_11_LED2	Saída	LED 2
12	PIO_0_PTT	Entrada	Push To Talk
14	PIO_1_Ampli_En	Saída	Habilita Amplificador
15	PIO_4_Pair	Entrada	Entra em modo de pareamento
16	PIO_2_V+	Entrada	Volume +
17	PIO_5_Squelch	Entrada	Silenciar
18	PIO_3_V-	Entrada	Volume -
19	PIO_6_Call	Entrada	Chamar
20	PIO_7_Vsend	Saída	Habilita a Transmissão

O par genérico de entrada/saída pode funcionar para o controle de qualquer função externa desejada. Quando ativamos a entrada do dispositivo A, a saída do dispositivo B é ativada com atraso médio de 50 ms. Entretanto, estes pinos estarão desativados em nosso circuito por não terem utilidade.

O **LED 1** indica que o dispositivo está ligado mas não está conectado quando está piscando em velocidade normal e indica condição de pareamento quando pisca mais rapidamente. Ele se apaga quando o dispositivo está conectado.

O **LED 2** fica piscando quando um terminal está conectado ao outro e apagado caso contrário.

Ao se ativar o **PTT** no dispositivo A, o **Vsend** é ativado no dispositivo B com o mesmo atraso padrão de 50 ms. O sinal **Vsend** é, geralmente, ligado a um rádio externo que permite a transmissão em *broadcast*. Como, em nossa aplicação, não há conexões com elementos externos, deixaremos esta função fora do circuito final.

O **Ampli_En** é o sinal de saída que responde às entradas **Call** e **Squelch** também com o atraso padrão. O sinal **Ampli_En** habilita um amplificador externo que pode ser utilizado no terminal de recepção. As entradas **Call** e **Squelch** no dispositivo A permitem que o amplificador acoplado ao dispositivo B seja habilitado somente quando A quiser transmitir. Isto evita que o receptor em B fique escutando ruído durante o período que não

houver transmissão. No nosso caso, como sempre há transmissão, este controle não faz sentido.

O sinal **Pair** é aquele que deve estar ativo ao se ligar o circuito para que o módulo entre em estado de pareamento. Este sinal será ativado por um botão nos circuitos transmissor e receptor. Este botão deve estar pressionado no momento em que o circuito é energizado.

Por último, temos os controles digitais **V+** e **V-** que permitem ao receptor controlar o volume do áudio que está recebendo. Apesar de contar com um dispositivo capaz de realizar uma comunicação bidirecional, nosso projeto necessitava apenas de uma comunicação unidirecional e, por esse motivo, o controle de volume estará presente apenas no circuito receptor.

Os sinais de áudio que os nossos dispositivos vão transmitir ou receber estão ligados aos pinos do *pinstrip* JF2, cujas funções estão colocadas na tabela 5. Sem tempo hábil para realizar os testes que viabilizariam a utilização das entradas e saídas diferenciais do módulo juntamente com um amplificador externo, optamos por utilizar as entradas e saídas pré-amplificadas disponíveis.

Sendo assim, nossa saída de áudio estéreo, que se dará através de um conector P2 de 3,5 mm, terá seus terminais esquerdo, comum e direito ligados, respectivamente, aos pinos 2, 4 e 6. A entrada, que também é feita com conector P2, será ligada aos pinos 8 (esquerda), 12 (comum) e 16 (direita). Depois de uma sucessão de testes, escolhemos este grupo de entradas que nos dá sinais de 1 Vpp e apresentaram menos ruído quando comparado às saídas de 0,4 Vpp.

3.5. DESENVOLVIMENTO DO CIRCUITO

Tendo em mente todas as funções a serem utilizadas, precisávamos montar um circuito que nos desse a alimentação para o módulo, a possibilidade de controlar entradas e monitorar saídas. Para controlar as entradas, são necessários apenas botões normalmente abertos e, para monitorar as saídas, LEDs.

Com a ajuda do fabricante, a realização de alguns testes e consultando o *datasheet*, foi possível determinar alguns blocos de circuito tais como o bloco de acionamento de botões (figura 8) e o bloco de acionamento de LEDs (figura 9).

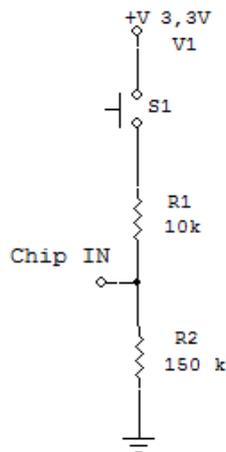


Figura 8 – Circuito de acionamento de botões.

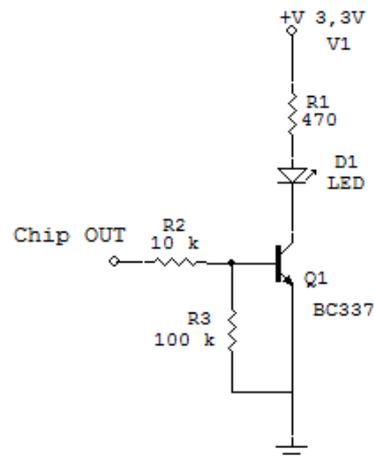


Figura 9 – Circuito de acionamento de LEDs.

No primeiro circuito temos um botão normalmente aberto que, quando apertado, envia um sinal para uma das entradas do chip. Este circuito está ligado a funções como aumento e redução do volume e pareamento dos dispositivos. Já o segundo circuito recebe um sinal do chip e, dependendo da amplitude desse sinal, aciona o transistor que está funcionando como chave e acende o LED.

O transistor do circuito da figura 9 é um TBJ do tipo NPN que funciona como uma chave normalmente aberta. Precisávamos de uma chave que se fecha quando recebe o nível lógico 1, que tem amplitude de 3,3 V, da saída do chip. Após alguns testes, optamos pelo transistor BC337, cujo chaveamento acontece quando a tensão base-emissor supera 1 V.

Chegada a hora de cuidar da alimentação do módulo, enfrentamos certa dificuldade. Para dar mais mobilidade ao dispositivo, optamos por acrescentar a possibilidade de alimentar o circuito utilizando uma bateria. Uma opção mais prática que aquela que vínhamos usando feita por uma fonte DC de 5 V ligada à rede. Entretanto, a bateria de 5 V é pouco usual e difícil de se achar.

Após conversar com alguns profissionais da área e fazer algumas pesquisas, encontramos um regulador de tensão de três terminais que é largamente utilizado para este fim. Este dispositivo permite que a tensão de saída seja ajustada combinando dois resistores como na figura 10 e ela pode variar de 1,2 a 37 V. A tensão de entrada deve ser, no mínimo, 2,5 V maior que a de entrada e a corrente máxima fornecida é de 1,5 A. A potência dissipada no caso extremo é 20 W, porém, no circuito que estamos trabalhando, as correntes são baixas e não será necessário o uso de dissipadores. [10]

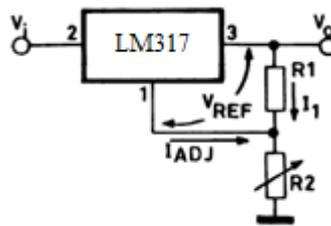


Figura 10 – Esquemático do circuito base do LM317. [10]

O LM317 tem uma tensão de referência interna de 1,25 V entre a saída e o terminal de ajuste. Geralmente fixamos o valor do resistor R1 e, dependendo da tensão desejada na saída, calculamos o valor do resistor R2. Para definirmos este valor, utilizamos a equação 2.

$$V_O = V_{REF} \left(1 + \frac{R_2}{R_1} \right) + I_{ADJ} \times R_2$$

Equação 2 – Tensão de saída do LM317. [10]

Como a corrente I_{ADJ} é praticamente constante e tem o valor máximo de 100 μ A, podemos desprezar esse segundo termo da equação. Assim, para obtermos uma tensão de saída de 5 V, utilizando uma bateria de 9 V e fixando o valor de R1 em 270 Ω , devemos utilizar um resistor R2 de 820 Ω . A tensão de saída será aproximadamente 5,05 V.

No nosso circuito, acrescentamos três capacitores que irão atuar como filtros. O circuito do regulador de tensão é mostrado na figura 11.

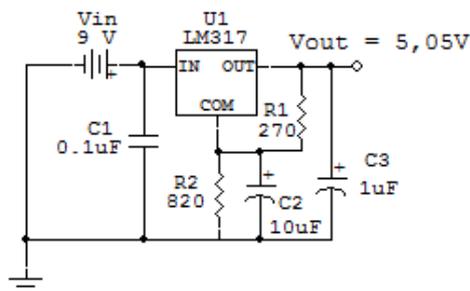


Figura 11 – Circuito regulador de tensão.

Num primeiro momento, utilizamos uma *protoboard* para montar e testar os blocos de circuito projetados. Posteriormente, montamos o que viria a ser o circuito completo ainda na *protoboard* (figura 12) e realizamos diversos testes, inclusive com outros *firmwares*.

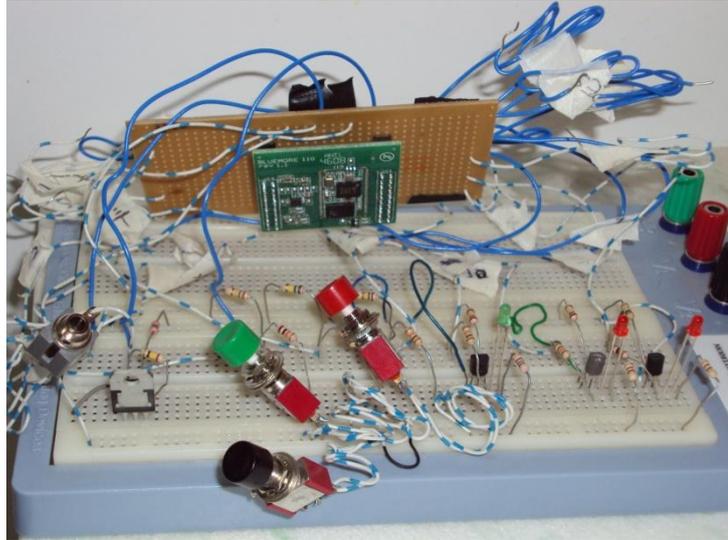


Figura 12 – Circuito de testes na *protoboard*.

Uma vez realizados todos os testes, esquematizamos o leiaute do circuito final no CircuitMaker. A seguir, mostramos o circuito de transmissão na figura 13 e o circuito receptor na figura 14.

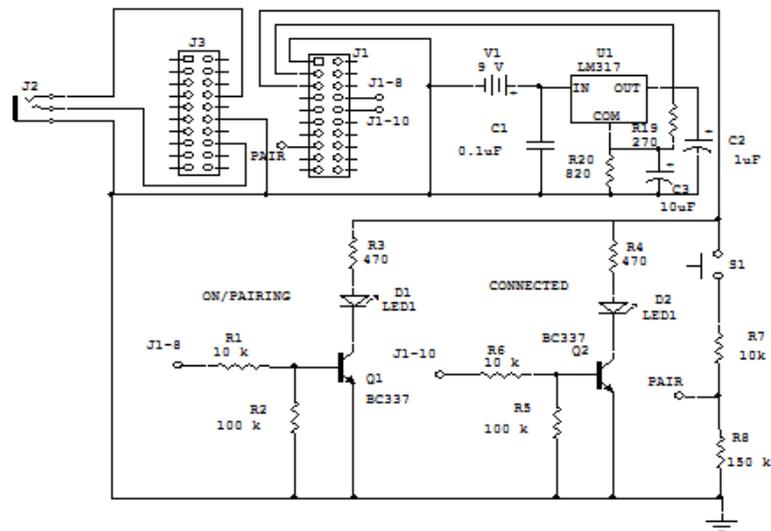


Figura 13 – Circuito de transmissão.

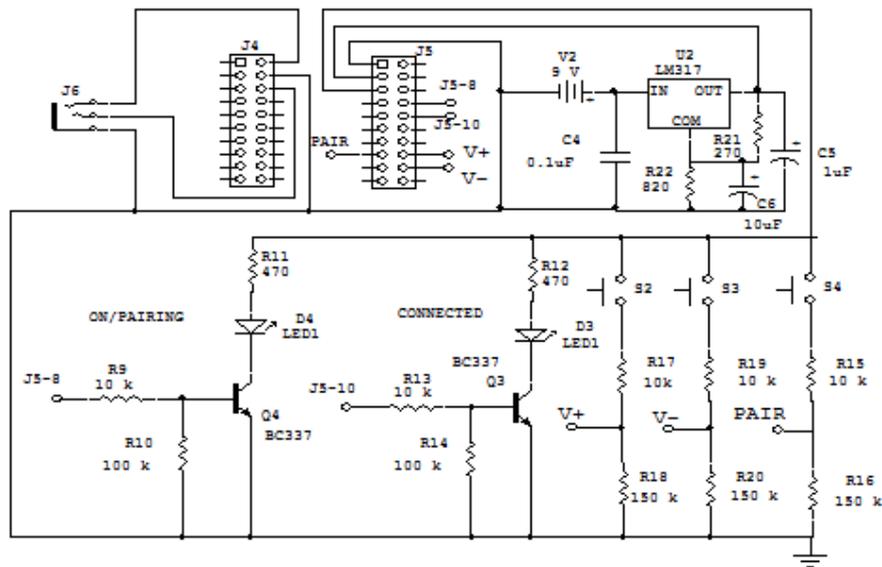


Figura 14 – Circuito de recepção.

Por último, julgamos interessante desenvolver um protótipo e, para isso, confeccionamos os circuitos das figuras 13 e 14 em placas de circuito impresso. O leiaute para impressão do circuito foi desenvolvido no programa Trax Maker. Para transferir o desenho para a placa de cobre, era necessário que tivéssemos o circuito impresso em papel especial numa impressora a laser. A transferência foi feita por um profissional que utiliza uma prensa de estampar camisetas para este trabalho.

De posse da placa com o circuito impresso nela, o trabalho de corrosão, perfuração e solda foi realizado com a ajuda dos técnicos da Central Técnica dos Laboratórios no SG-11. O resultado é mostrado na figura 15.



Figura 15 – Foto do protótipo das placas pronto.

4. RESULTADOS

Agora, serão apresentadas as características do nosso dispositivo, assim como alguns problemas encontrados com o dispositivo já pronto. Adicionalmente, alguns testes que foram realizados e sugestões de continuação para o projeto.

4.1. CARACTERÍSTICAS FINAIS

Quando os dispositivos finalmente ficaram prontos, pudemos analisar os resultados obtidos. Animadoramente, o alcance medido em campo aberto foi de 50 metros e superou bastante as nossas expectativas e, inclusive, os valores nominais. Entretanto, a transmissão ficava muito sensível quando os dispositivos estavam separados pelos 50 metros e até uma pessoa atravessando o caminho entre os dois interrompia a transmissão. Se os dispositivos estão separados por mais de 20 metros, qualquer movimento brusco de um terminal provoca uma breve interrupção da transmissão.

Em ambientes fechados, o alcance é de aproximadamente 10 metros quando temos uma parede de tijolos separando os dispositivos. Com mais paredes, o alcance piora. Combinados os dois resultados, sugerimos uma distância máxima de 20 metros a céu aberto e, para aplicações internas, o ideal é que os dispositivos se encontrem no mesmo cômodo.

Durante os testes, encontramos uma série de problemas com os conectores P2 de 3,5 mm. Substituímos vários deles e todos eram bem frágeis e apresentavam mau contato. Fomos a várias lojas especializadas e só encontramos os conectores defeituosos do mesmo fabricante. Mas o conector é de fácil reposição e ainda estamos buscando um mais robusto.

Durante algumas medições que estavam sendo feitas no protótipo, percebemos que a tensão de alimentação estava em torno de 5,5 V e o dispositivo apresentava um pequeno aquecimento que não era esperado. Para não sobrecarregar o módulo, decidimos alterar o circuito de alimentação e empregamos o regulador LM7805.

Esta família de reguladores apresenta uma tensão de 5 V fixa na saída independente de estar conectado a uma malha de resistores. A única condição é que a tensão de entrada seja 2 V acima da tensão de saída e não ultrapasse o limite de 20 V. Como usamos baterias de 9 V, o regulador fornece 5 V exatos. A corrente máxima fornecida é de 1A e a potência dissipada em casos extremos é 15W. [11] A figura 16 abaixo mostra o circuito base.

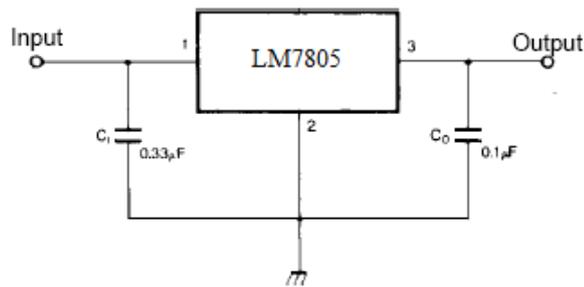


Figura 16 – Circuito base do LM7805. [11]

Os capacitores utilizados têm a mesma função que tinham no circuito de alimentação anterior. Com este novo regulador podemos utilizar diversos tipos de baterias e pilhas com diferentes voltagens. Continuamos com a bateria de 9 V porque já a estávamos utilizando, mas poderíamos optar por baterias menores. O novo circuito de alimentação é mostrado na figura 17 e, se comparado ao circuito da figura 11, percebe-se que não é difícil fazer a adaptação de um para o outro.

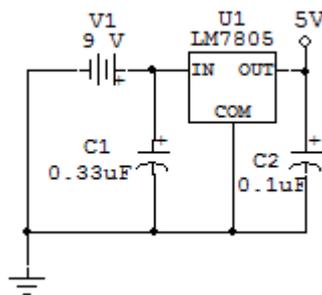


Figura 17 – Bloco de alimentação inserido no circuito.

4.2. ANÁLISE DE CUSTO

Após a conclusão do projeto, pudemos organizar uma planilha final de custos (Tabela 8) e fazer uma estimativa do custo que teria a produção de cada dispositivo.

Tabela 8 – Tabela de custos aproximados para o desenvolvimento do dispositivo.

Material	Preço
2 chips <i>Bluetooth</i>	€ 100,00 ≈ R\$ 260,00
Componentes (transistores, resistores, capacitores, chaves, etc.)	R\$ 50,00
Material para o circuito impresso (placa e fio de solda)	R\$ 10,00
Total	R\$ 320,00

O custo de cada dispositivo ficou em torno de R\$ 160,00. Se tivéssemos interesse em comercializar este produto, poderíamos reduzir significativamente este valor. Quando se compra os *chips* em lotes de grandes quantidades, o valor unitário cai consideravelmente. Outros fabricantes oferecem módulos *Bluetooth* semelhantes e com preços competitivos. Entretanto, esses fabricantes não disponibilizam a placa de desenvolvimento e, na maioria dos casos, a mudança de *firmware* não é uma opção. Por isso, escolhemos os módulos fabricados pela Eikon, que nos ofereceram mais alternativas e tornaram o projeto mais didático. Além disso, a placa de desenvolvimento ainda poderá ser aproveitada no desenvolvimento de outros projetos.

4.3. TESTES PARALELOS

Ao longo do projeto, alguns testes, paralelos ao desenvolvimento do circuito principal, foram feitos para estudar outras aplicações e gerar possíveis melhorias no sistema. Todavia, nem todos os testes que gostaríamos puderam ser realizados devido à falta de tempo ou alguma outra inconveniência. A seguir fazemos considerações sobre os outros *firmwares* do fabricante, testes que foram realizados e outros que são sugestão para complemento do trabalho.

4.3.1. FONE DE OUVIDO

Utilizando a placa de desenvolvimento, testamos as funções do *firmware Wireless Headset*. Ele funciona como os aparelhos disponíveis no mercado. As entradas e saídas foram selecionadas de acordo com a pinagem da tabela 9. Lembrando que as entradas tem botões que são operados pelo usuário. E, na saída, LEDs que indicam o funcionamento do módulo.

Tabela 9 – Entradas e saídas utilizadas no *firmware Wireless Headset*. [12]

Nome	IN/OUT	Descrição	Pinagem (Pio)
MFB (Botão Multifuncional)	IN	Liga e Desliga o equipamento, assim como é usado para iniciar o pareamento.	5
V+	IN	Aumenta o volume.	4
V-	IN	Diminui o volume.	3
LED 1	OUT	Quando aceso o dispositivo está pareado	0
LED 2	OUT	Quando aceso o dispositivo está ligado.	7

A conexão do módulo, funcionando como *headset*, com os demais aparelhos é protegida por uma senha (PIN). A senha padrão é 1234 e existe um software onde ela pode ser modificada. Neste mesmo software podemos alterar outras configurações como o nome que identifica o módulo na rede. Entretanto, estas configurações não são importantes nesta análise.

4.3.2. A2DP (*Advanced Audio Distribution Profile*)

O nosso dispositivo não é capaz de se conectar a outros aparelhos que tenham a tecnologia *Bluetooth*. Se o circuito receptor fosse capaz de reproduzir o sinal de áudio de um celular tal qual o circuito do fone de ouvido, o dispositivo teria mais funcionalidades.

O *firmware* A2DP dá ao módulo a função de *gateway* de áudio. É possível que este circuito seja capaz de se conectar ao circuito do fone de ouvido e, dessa maneira, teríamos um par de dispositivos preparados para se conectar entre si e a outros aparelhos. Todavia, este *firmware* está em fase de aprimoramento e apresenta muitos erros. O fabricante ainda não pode disponibilizar uma versão definitiva.

4.3.3. COMUNICAÇÃO DE DADOS

Outra aplicação possível para os módulos *Bluetooth* é a eliminação de cabos seriais. Desta maneira, poderíamos conectar *laptops* e computadores a impressoras, aparelhos de fax e outros periféricos que utilizam tal porta. Este teste está um pouco distante dos nossos objetivos e foi preterido em função do tempo disponível.

5. CONSIDERAÇÕES FINAIS

Ao final do trabalho, conseguimos atingir com sucesso a meta traçada. Desenvolvemos dois circuitos que transmitem áudio com boa qualidade, gastam pouca energia e fornecem mobilidade aos usuários. Os dispositivos são de fácil manuseio e podem ser operados por qualquer pessoa. As dificuldades encontradas durante o projeto foram superadas com sucesso. Os dispositivos serão de grande utilidade em nosso dia-a-dia dispensando a inconveniência do uso de cabos.

Durante a execução do projeto, várias idéias surgiram e poderiam ser implementadas com os chips *Bluetooth*. Entre elas, podemos citar o desenvolvimento de outros dispositivos transmissores de sinais de áudio, como o que foi construído, e todos iriam se conectar simultaneamente com um único dispositivo de recepção formando uma *piconet*. Assim, poderíamos conectar os instrumentos de toda uma banda à mesa de som por meio de dispositivos *Bluetooth*. Este caminho é muito interessante e desperta a nossa curiosidade. A didática da placa de desenvolvimento nos permitiu fazer testes, experiências, chegar a vários resultados e idéias para trabalhos futuros.

Por fim, depois dos cinco anos de graduação, percebemos o desenvolvimento da nossa capacidade de desenvolver projetos, proporcionar melhorias e solucionar os problemas que, invariavelmente, aparecem. Hoje temos certeza de que cada aula de cada disciplina foi de extrema importância na nossa formação e contribuiu de alguma maneira em nosso projeto.

REFERÊNCIAS

- [1] ALECRIM, E. *Tecnologia Bluetooth*. Disponível em: <www.infowester.com/bluetooth.php>. Último acesso em: 10/12/2009.
- [2] Bluetooth SIG. *Specification of the Bluetooth System*. Disponível em <www.bluetooth.com>. Último acesso em: 10/12/2009.
- [3] MALBURG, M. *Modulação*, Grupo de Teleinformática e Automação, UFRJ. Disponível em: <www.gta.ufrj.br/>. Último acesso em: 10/12/2009.
- [4] TUDE, E. *Bluetooth*. Disponível em: <www.teleco.com.br>. Último acesso em: 10/12/2009.
- [5] Wikipedia. *Bluetooth*. Disponível em: <<http://en.wikipedia.org/wiki/Bluetooth>>. Último acesso em: 10/12/2009.
- [6] BISDIKIAN, C. *An Overview of the Bluetooth Wireless Technology*. IEEE Communications Magazine, Dezembro de 2001.
- [7] MCDERMOTT-WELLS, P. *What is Bluetooth?* IEEE Potentials, Dezembro de 2004.
- [8] FLOROS, A.; TATLAS, N.; MOURJOPOULOS, J. *A High-Quality Digital Audio Bluetooth Platform*. IEEE Transactions on Consumer Electronics, Agosto de 2006.
- [9] PHILIPS. BC337 Specification Datasheet. Abril de 1999.
- [10] ST MICROELECTRONICS. LM317 Specification Datasheet. Setembro de 2004.
- [11] FAIRCHILD. LM7805 Specification Datasheet. Julho de 2001.
- [12] EIKON. Eval-Bluemore Specification Datasheet. Janeiro de 2009.
- [13] EIKON. Bluemore110 Specification Datasheet. Janeiro de 2009.
- [14] PROTEL INTERNATIONAL LIMITED. *CircuitMaker 2000*.
- [15] PROTEL INTERNATIONAL LIMITED. *TraxMaker*.