



**UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE TECNOLOGIA
FACULDADE DE ENGENHARIA DA COMPUTAÇÃO E
TELECOMUNICAÇÕES**

**Uso de Reconhecedor e Sintetizador de Voz Embarcados para
Controle de Equipamentos Eletrônicos**

**Belém – Brazil
Abril/2015**

Uso de Reconhecedor e Sintetizador de Voz Embarcados para Controle de Equipamentos Eletrônicos

Cassio Trindade Batista
cassio.batista.13@gmail.com
201106840003

Pedro Henrique C. F. Soares
pedrofigueiredoc@gmail.com
201106840007

Gabriel Peixoto de Carvalho
gaburiero.c@gmail.com
201106840010

Thiago Barros Coelho
tbarroscoelho@gmail.com
201106840041

Projeto apresentado à disciplina Projetos de
Hardware e Interfaceamento como requisito de
avaliação. Professores: Jeferson Leite e Adal-
bery Castro.

Belem – Brazil
Abril/2015

Sumário

1	Introdução	4
2	Objetivos	4
2.1	Reconhecimento e Síntese de Voz	4
2.2	Controle Remoto de TV e Servidor LAMP	5
3	Justificativa	6
4	Revisão Teórica	7
4.1	Produtos Relacionados	9
5	Metodologia	10
6	Orçamento	11
7	Dificuldades e Soluções	11
8	Trabalhos Futuros	12
A	Pedro: Codigos LAMP	14
B	Thiago: Codigos Arduino/C for IR Tx/Rx	14

Lista de Figuras

1	Novo van Gogh	5
2	Esquemático de um sistema ASR	7
3	Esquemático do protocolo RC-5 da Philips.	9
4	Esquemático do Cliente LaPS CSR.	10

1 Introdução

A interface homem-máquina encontra-se cada vez mais amigável. O que antes era portado somente por empresas e pessoas com poder financeiro diferenciado e acima da média, em termos de tecnologia, é hoje muito mais acessível e simples para usuários domésticos sem profundo conhecimento no assunto. Devido a abrangência de computadores pessoais e embarcados e da Internet, novas oportunidades e expectativas em termos de trabalho, estudos e até lazer são criadas a fim de melhorar ainda mais essa comunicação, de modo que a máquina se aproxime mais de ações típicas do ser humano, como pensar e falar.

Acredita-se que a síntese e o reconhecimento automático de voz (do inglês *text-to-speech* e *automatic speech recognition*, respectivamente, TTS e ASR) [1, 2] tornam a interface citada acima muito mais prática e natural, de forma que a comunicação de fato se assemelha àquela estabelecida entre duas pessoas. O ASR refere-se ao sistema que, tomando o sinal de fala digitalizado como entrada, é capaz de gerar o texto transcrito na saída. Já um sistema TTS realiza a função contrária, na qual um sinal analógico de voz é sintetizado de acordo com o texto posto na entrada. Dentre as inúmeras aplicações que utilizam os sistemas que envolvem processamento de fala, pode-se destacar a automação residencial que visa ajudar pessoas que tenham dificuldades no controle de alguns equipamentos eletrônicos, dando ênfase na acessibilidade alcançada através das tecnologias assistivas.

Tecnologia Assistiva (TA) é um campo da engenharia biomédica dedicada à aumentar a independência e mobilidade de pessoas com deficiência, englobando metodologias, práticas e serviços que objetivam promover sua autonomia, qualidade de vida e inclusão social [3, ?]. Tal tecnologia busca reduzir a necessidade vivenciada por pessoas que precisam de soluções que não as deixem à margem da utilização de dispositivos eletrônicos. Em outras palavras, para diminuir a exclusão digital imposta pela incapacidade de manipular certos dispositivos, a acessibilidade é vista como elemento fundamental para elevar a autoestima e o grau de independência dessas pessoas. Além disso, as mesmas soluções apresentadas podem ser úteis também para os não portadores de necessidades especiais, já que o controle de equipamentos se torna mais prático e confortável.

Nesse sentido, este trabalho busca preparar um servidor local portátil de reconhecimento e síntese de voz em Português Brasileiro (PT_BR) baseado no microcomputador BeagleBone Black de modo que, quando acessado pelo dispositivo que agirá como controle remoto — no caso, um smartphone com sistema operacional Android —, seja capaz de acessar as funções mais básicas de um aparelho televisivo. Vale ressaltar que todas as APIs, IDEs, *softwares*, *libraries* e pacotes utilizados para criação dos sistemas e dos recursos utilizados (com exceção do HTK, o qual será visto mais adiante) possuem licença *open source* e são encontrados disponíveis livremente na Internet.

2 Objetivos

O objetivo principal consiste em criar um protótipo portátil, baseado em um microcomputador embarcado, que seja capaz de controlar um aparelho de televisão através do envio remoto de sinais. O sistema será configurado como um servidor que disponibiliza um serviço genérico de reconhecimento de fala, de modo que o aparelho de TV mencionado possa ser remotamente controlado através da voz do usuário; e um serviço de síntese de fala, provendo *feedback* das ações de acordo com o entendimento do sistema de ASR. Além disso, a informação a ser enviada para a TV deve ser armazenada em um banco de dados, também configurado no mesmo servidor.

2.1 Reconhecimento e Síntese de Voz

Para que o reconhecimento automático de voz seja possível, o *software* Julius deverá ser instalado no servidor. Julius é um software capaz de processar e decodificar áudio em aproximadamente tempo real para tarefas de ditado de até 60 mil palavras. Este também será o principal programa do sistema, o qual receberá em seu código nativo todos os outros módulos.

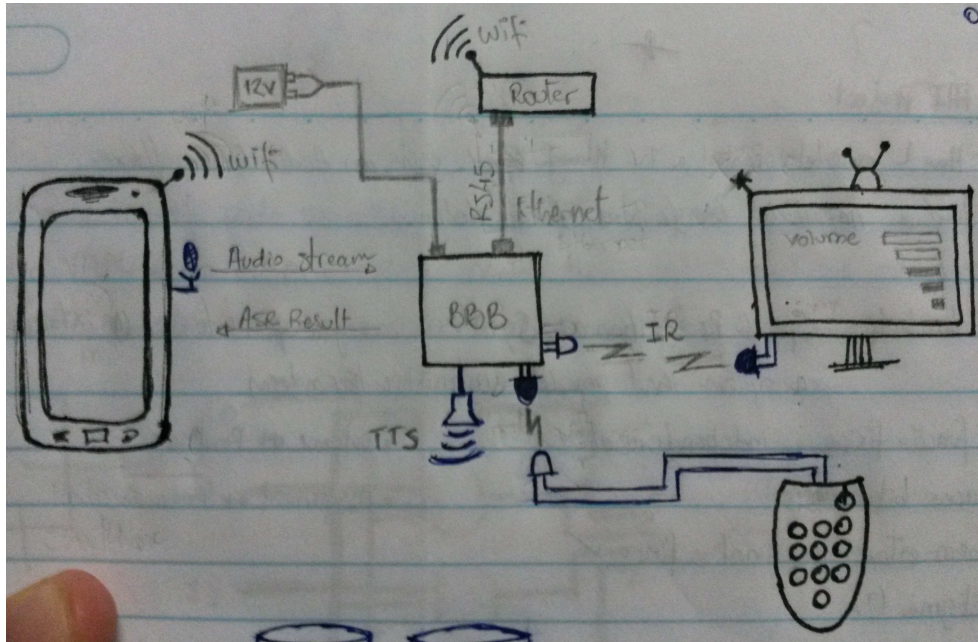


Figura 1: Novo van Gogh

Para que o Julius possa realizar o reconhecimento em Português Brasileiro, serão necessários basicamente dois recursos: um modelo acústico e um dicionário fonético. Modelos acústicos genéricos para PT_BR podem ser encontrados na página do Grupo FalaBrasil [4], bem como o software que cria o dicionário fonético (conversor grafema-fonema ou G2P) [5]. Entretanto, embora a taxa de acerto dos modelos seja satisfatória, é possível melhorá-la através da criação ou treino de modelos específicos para a aplicação.

O processo de treino poderá ser realizado pelo software HTK (*Hidden Markov Models Toolkit*), o qual é capaz de extrair segmentos de fala de um arquivo de áudio e assinalar uma referência à ele. Tal referência é retirada do dicionário fonético, previamente criado com o software G2P.

O software eSpeak é a principal referência em síntese de voz em ambientes Linux. Graças à disponibilização de uma API no site oficial do desenvolvedor, o sistema, além de conseguir “ouvir e entender”, também será capaz de “falar”. O download dos modelos para PT_BR é feito juntamente com o das bibliotecas necessárias. Como a BBB não possui saída de áudio nativa, um auto-falante USB será utilizado.

2.2 Controle Remoto de TV e Servidor LAMP

Os aparelhos televisivos atuais, assim como a grande maioria dos dispositivos eletrônicos domésticos, possuem a tecnologia de controle remoto baseada em luz infravermelha. Pode-se observar que na extremidade superior dos controles remotos, há pelo menos um led infravermelho (IR Led) capaz de emitir luz e, dessa forma, transmitir uma informação binária para o circuito localizado na parte frontal da TV. Esse circuito possui um sensor infravermelho (IR sensor), o qual, atuando como o receptor da comunicação, é capaz de receber os bits transmitidos e repassá-los para o processador do circuito, o qual executará a tarefa relacionada à decodificação dos bits (desligar a TV, mostrar o menu, etc).

Nesse sentido, um módulo receptor será adicionado à BeagleBone para que esta seja capaz de “hackear”, através de um sensor infravermelho, informações dos controles remotos, dado que o acesso aos *datasheets* de diversos aparelhos não é trivial. O sistema de gerenciamento de banco de dados MySQL será instalado e configurado no servidor para armazenar o código, o qual é relacionado a ações como “aumentar volume” ou “desligar” da TV principal e dos demais aparelhos que vierem a ser cadastrados no banco. Toda a informação de saída a ser enviada para a TV através dos IR leds deve ser oriunda do banco de dados.

3 Justificativa

No censo realizado em 2010 pelo Instituto Brasileiro de Geografia e Estatística (IBGE), aproximadamente 23,9% dos brasileiros declaram ter alguma deficiência [6]. Esse número é realmente impressionante, pois revela que cerca de um quarto de uma população de 190 milhões de habitantes é portadora de necessidades especiais. Segundo os dados, 6,9% (13,3 mi) dos brasileiros apresentam algum grau de deficiência motora, enquanto 18,8% (35,7 mi) afirmam serem cegas ou terem alguma dificuldade para enxergar.

Em [?], uma pesquisa foi realizada com brasileiros portadores de necessidades especiais para se determinar quais as características que esses cidadãos gostariam de adicionar, se pudessem, em controles remotos convencionais, visando minimizar suas dificuldades em utilizá-los. A sugestão de um *design* mais limpo foi unânime, enquanto os deficientes visuais, em particular, apontaram a necessidade de algum tipo de *feedback* quando os botões fossem pressionados e a implementação de alguma referência reconhecível pelo tato nos botões. Já os deficientes motores desejam um dispositivo menor, que não escorregasse facilmente das mãos, contendo um ângulo de controle mais abrangente e com botões maiores para aqueles que não conseguem ter controle absoluto sobre as mãos e/ou dedos. O estudo também mostrou que as referências sentidas pelo tato não foram incorporadas no design a fim de se manter o baixo custo.

Essa pesquisa tem como finalidade apresentar uma solução para diminuir a exclusão digital vivenciada especialmente por pessoas com necessidade motora ou visual, as quais estão à margem da utilização de dispositivos eletrônicos por conta da ausência de soluções que os adaptem às suas necessidades. A tecnologia de reconhecimento de fala torna acessível a utilização de qualquer dispositivo eletrônico por usuários incapazes de realizar movimentos específicos com membros superiores, como segurar um controle físico e apertar botões ou digitar, por exemplo. Além disso, os portadores de necessidades visuais também poderão ser ajudados, já que nem todos os controles possuem referências hápticas. A síntese de fala também se torna muito importante no contexto da dificuldade visual, já que, nesse sentido, um *feedback* por fala será muito mais útil do que via texto. Em [?] é sugerido que as interfaces multimodais (métodos alternativos de controle como voz, escrita, gestos, etc) associadas ao controle de equipamentos provêm uma experiência muito mais prazerosa e prática ao usuário do que os métodos convencionais (botões), apesar de diminuir a usabilidade e aumentar o tempo de execução da tarefa.

O Ato de Americanos com Deficiência (ADA) [?] é um documento que regula os direitos dos cidadãos com deficiência nos EUA, além de prover a base legal dos fundos públicos para compra dos recursos que estes necessitam. Algumas categorias de TA foram criadas com base nas diretrizes gerais da ADA, das quais podemos ressaltar três como justificativa do trabalho:

1. Recursos de acessibilidade ao computador

- Equipamentos de entrada e saída (síntese de voz, Braille), auxílios alternativos de acesso (ponteiros de cabeça, de luz), teclados modificados, softwares especiais (reconhecimento de voz, etc.), que permitem as pessoas com deficiência a usarem o computador.

2. Sistemas de controle de ambiente

- Sistemas eletrônicos que permitem as pessoas com limitações moto-locomotoras controlar remotamente aparelhos eletro-eletrônicos, sistemas de segurança, entre outros, localizados em seu quarto, sala, escritório, casa e arredores.

3. Auxílios para cegos ou com visão subnormal

- Auxílios para grupos específicos que inclui lupas e lentes, Braille para equipamentos com síntese de voz, grandes telas de impressão, sistema de TV com aumento para leitura de documentos, publicações etc.

A decisão de criar um servidor próprio de síntese e reconhecimento de voz baseia-se principalmente na possibilidade de usufruir de tais recursos de forma offline, ou seja, sem a necessidade de conexão com a

Internet. O fato de as comunicações ocorrerem em LAN diminui muito o tempo de espera do usuário, já que o áudio não precisa acessar servidores de voz distantes. Ainda sobre o sistema ASR, pode-se também citar a vantagem de limitar o vocabulário de palavras utilizados através da implementação de uma gramática, já que serviços online de reconhecimento (como o disponibilizado pelo Google, por exemplo), trabalham com a inteira modelagem das palavras do idioma, impossibilitando a criação de um contexto específico para a aplicação.

4 Revisão Teórica

Como o iOS e o Android foram lançados, respectivamente, em 2007 e 2008, e a ascensão dos smartphones é relativamente recente, a ideia de adicionar a eles a acessibilidade no controle de equipamentos eletrônicos somente começou a revelar resultados concretos a partir de 2010. Em [?], o decodificador PocketSphinx foi embarcado em um smartphone Android para que este pudesse controlar aparelhos domésticos através da interface de voz. O resultado era enviado para uma SparkFun IOIO Board, a qual, estando fisicamente conectada ao controle da TV, disparava um determinado *trigger*. A justificativa do trabalho era ajudar pessoas afetadas com tetraplegia a serem mais independentes; os testes avaliaram o desempenho de decodificadores embarcados e distribuídos.

Cassio: Add another reference here

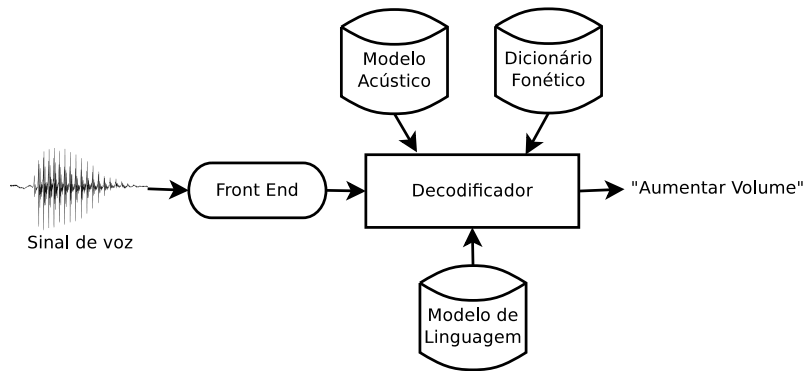


Figura 2: Esquemático de um sistema ASR

Os fundamentos do reconhecimento automático de voz, assim como os da síntese de voz, são descritos com bastante detalhes em [2]. A arquitetura mais geral e aceita na literatura é mostrada na Figura 2. Vale salientar que, ao invés do uso de modelos mais gerais, que descrevem a maior parte de uma linguagem, serão utilizadas gramáticas livres de contexto, as quais limitam o vocabulário utilizado a apenas um conjunto de sentenças possíveis, escolhidas pelo desenvolvedor do sistema. A construção do dicionário fonético para PT_BR dar-se-á através do software descrito em [5]; o tutorial para o treino do modelo acústico encontra-se disponível na página do projeto Voxforge [7], bem como capítulo 3 do livro do HTK [8, p. 22–42]; a gramática reconhecida pelo Julius é criada manualmente de acordo com o descrito na página oficial [9]. Instruções de configuração e utilização do Julius encontram-se na documentação oficial [?].

Como saída analógica do sistema TTS, a voz sintetizada deve ser reproduzida por um dispositivo externo à BeagleBone, já que esta não possui auto-falantes próprios. Como visto em [?], o dispositivo primário de saída de áudio da BeagleBone é o HDMI, o qual pode ser desabilitado mediante modificações em parâmetros do kernel. Feito isso, o USB, que é o dispositivo secundário, se torna o principal, fazendo com que a solução mais simples seja plugar um auto-falante (*speaker*) na porta USB. Na página oficial do eSpeak, um arquivo de cabeçalho (*header*) permite a utilização de uma API em C/C++, a qual facilita o acesso aos módulos do software que permitem que a BeagleBone “fale” [10].

A escolha da plataforma foi fundamental para a esquematização do projeto. Arduino, Raspberry Pi e

BeagleBone Black foram as três principais opções a serem escolhidas. Diversos tutoriais de comparação entre as plataformas foram consultados e estão disponíveis na Internet [11, 12, 13]. O Arduino, apesar de ser uma ferramenta flexível e com grande capacidade de interfaceamento com uma vasta quantidade de dispositivos, é uma plataforma simples, recomendada para projetos de menor porte. O microcontrolador, que pode ser programado em C, torna-se muito limitado quando o projeto requer um servidor estável e relativamente potente; O Raspberry Pi, por ser bastante completo, já se enquadra no conceito de mini computador. Todo o seu armazenamento é fornecido por um cartão SD, além de ser possível conectá-lo à Internet através de um conector Ethernet. Sendo necessário a instalação de um sistema operacional, o Raspberry Pi ainda possui interface de saída HDMI, tornando-se muito útil para aplicações gráficas.

Tabela 1: Comparação entre as três principais plataformas

	Arduino UNO	BeagleBone Black	Raspberry Pi
Chip	-	TI AM3359	BCM2835 SoC full HD
CPU	20 MHz ATmega328	1 GHz ARM Cortex-A8	700 MHz ARM1176JZ-F
GPU	-	PowerVR SGX530	Dual Core VideoCore IV
Armazenamento	2 kB SRAM	512 MB DDR3	512 MB SDRAM
Flash	32 kB	2 GB eMMC, MicroSD	SD, MMC, SDIO card slot
GPIO	14	65	8
Video	-	mini HDMI	HDMI
OS	-	Linux	Linux
Amperagem (mA)	42	210-460	150-350
Voltagem (V)	7-12	5	5
USB	-	1 Host, 1 Mini Client	2 Hosts, 1 Micro Power
Ethernet	-	1 10/100 Mbps	1 10/100 Mbps
Preço	5 conto	300 conto	200 conto

A BeagleBone é comparável ao Raspberry Pi. Entretanto, por ter mais pinos (GPIO) e um processador mais poderoso, a BeagleBone é uma escolha óbvia para projetos mais elaborados. Além de possuir diversas opções de conexão, a BeagleBone une a flexibilidade de interfaceamento do Arduino com a capacidade de processamento rápido do Raspberry Pi. Apesar da desvantagem no preço, não restaram muitas dúvidas no momento da escolha dessa plataforma para o projeto. Uma comparação entre os principais parâmetros dos três equipamentos é dada na Tabela 1.

↓ Thiago: LIRC ↓ Gabriel: IR theory ↓ References + Figures + Tables ↓

O funcionamento de controles remotos, com ênfase nos baseados em luz infravermelha para televisores, é explicado de forma clara e detalhada em diversos tutoriais para “curiosos” disponíveis na Internet, como os da revista Mundo Estranho [14] e do blog *How Stuff Works?* da UOL [15]. A maioria dos aparelhos eletrônicos atualmente recebe informação através de sensores infravermelhos localizados em painéis frontais. Um grande problema é a interferência que surge com a vasta transferência de informação via IR. Devido à isso, a comunicação entre o controle remoto e a televisão ocorre em 4 passos: um comando *start* inicia a transferência, seguido dos bits do comando específico (como aumentar o volume, por exemplo) e do endereço do aparelho. Por fim, um comando de *stop* encerra o envio de bits. Dessa forma, a chance de a informação ser reconhecida por mais de um aparelho é baixa (salvo o caso de serem dois equipamentos do mesmo tipo e da mesma empresa).

Uma das grandes dificuldades relacionadas à comunicação IR para controle de equipamentos é que cada empresa praticamente segue o seu próprio padrão de transmissão de informação: o número, a ordem e o significado dos bits é variado, a frequência dos pulsos oscila entre 32 kHz e 40 kHz, chegando a 50 kHz em determinados aparelhos mais modernos. Além disso, a disponibilização de documentação pelos fabricantes é muito rara.

Neste trabalho, será abordado o protocolo RC-5, desenvolvido pela Philips. O protocolo teve sua última documentação liberada em 1992, quando não existiam home theaters, DVDs e outra vasta gama de dis-

positivos eletrônicos modernos remotamente controlados. MAS EU ESPERO QUE FUNCIONE NESSA TV DE JESUS.

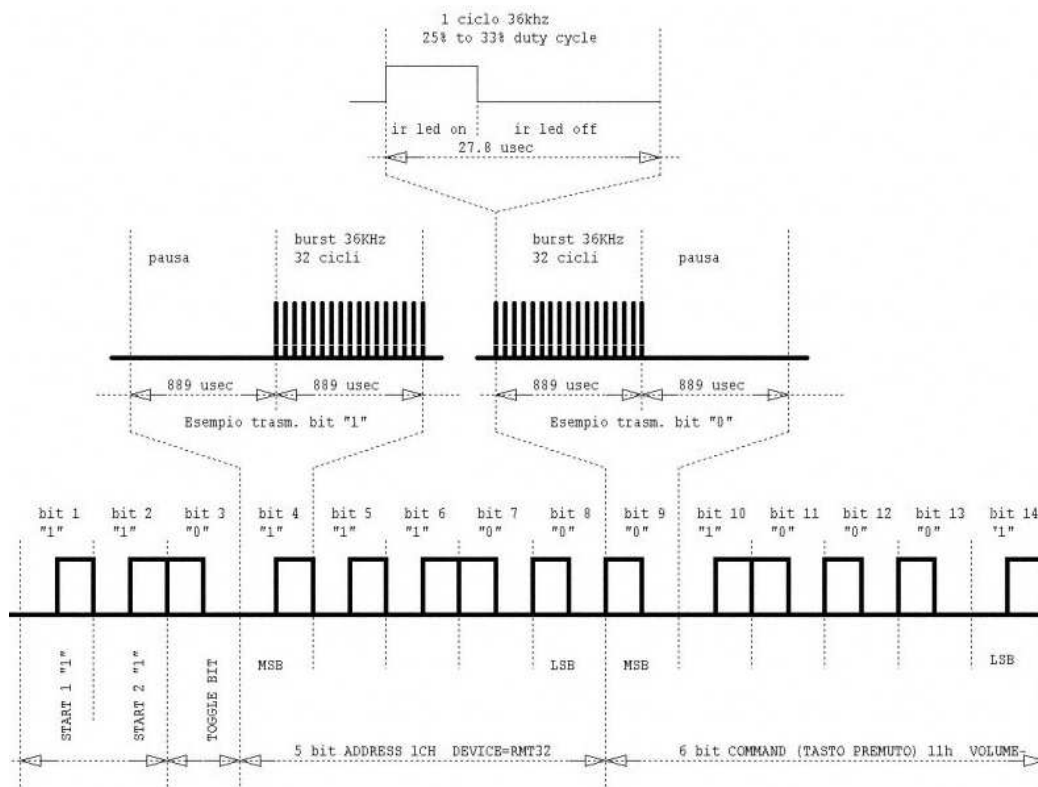


Figura 3: Esquemático do protocolo RC-5 da Philips.

4.1 Produtos Relacionados

Existem alguns produtos disponíveis no mercado com a finalidade de tornar o controle de equipamentos eletrônicos mais prático. Um deles é o *Harmony Smart Control*, o qual possui em seu “pacote” um aplicativo para iOS e Android (porém sem versão para tablets; somente smartphones), um hub e um controle remoto genérico. Segundo o review da cnet, vale a pena pagar US\$ 130 por todas as funcionalidades que o sistema apresenta, como usar uma conexão RF entre o smartphone/controlador com o hub (que, infelizmente, ainda não conseguiu se livrar do tão antiquado IR), o que faz com que o usuário não precise apontar o controle para o dispositivo que precisa controlar. Todavia, o hub precisa estar numa boa posição para conseguir emitir a informação de forma clara para o aparelho desejado.

IRdroid é outro aplicativo que permite o controle de aparelhos televisivos com o celular. Como o próprio nome sugere, funciona apenas em dispositivos Android, desde que seu módulo de hardware esteja acoplado na saída de audio jack do smartphone. Versões mais recentes já possuem o hardware que pode ser acessado via bluetooth, custando US\$ 60 o mais caro. A grande vantagem é que o IRdroid, além de ser baseado na biblioteca LIRC, a qual possui uma vasta quantidade de equipamentos em seu banco de dados, possui código livre e disponível.

Nenhuma das soluções possui suporte à reconhecimento e síntese de voz offline em PT_BR. CHUPA!!!

5 Metodologia

O servidor, por ser o elemento chave na consolidação do projeto, deve ser o módulo a ser prioritariamente configurado, a fim de ser preparado para atender às devidas requisições, bem como executar qualquer tipo de aplicação solicitada. Sendo assim, a instalação do sistema operacional Ubuntu foi tomada como primeiro passo. As dependências a serem instaladas são listadas na Lista 1.

Listing 1: Pre-instalação de dependências no servidor

```
# general dependences
build-essential alsa-tools sox

# eSpeak dependences
libespeak-dev libportaudio2 libportaudio-dev

# Julius dependences
libasound2 libasound2-dev

# LAMP dependences
apache2 libapache2-mod-fastcgi # apache server
mysql-server libapache2-mod-auth-mysql php5-mysql # MySQL
libmysqlclient-dev # C
phpmyadmin # (opcional?)
php5 libapache2-mod-php5 php5-mcrypt # PHP (opcional?)
libmysqlcppconn7 libmysqlcppconn-dev # C++ (opcional?)
```

É importante ressaltar que os sistemas operacionais embarcados são simplificações de sistemas operacionais mais robustos, tendo a maior parte das suas funcionalidades reduzidas ou simplificadas para se adequar à uma plataforma de menor porte. Por isso, a preparação deve ocorrer a partir dos pacotes mais básicos, como o GCC, por exemplo. Outros pacotes devem ser instalados de forma gradual, tais quais os requeridos pelo Julius, eSpeak e os necessários para a implementação do servidor LAMP em C.

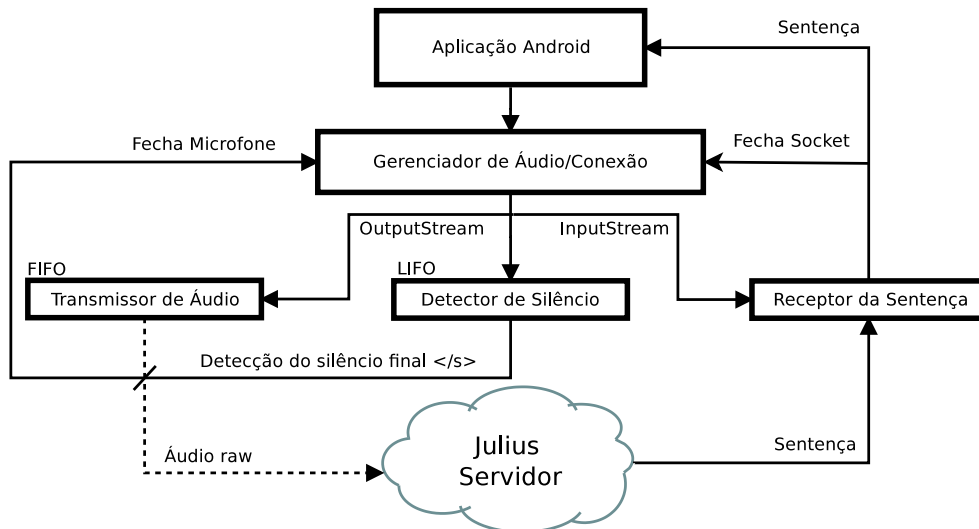


Figura 4: Esquemático do Cliente LaPS CSR.

Em [?], o Julius foi configurado para funcionar em modo servidor através da opção nativa “-adinnnet” (A/D *Input from Network*, conversão A/D com entrada pela rede). Isso permite que o Julius receba

amostras de áudio via *streaming* através de uma comunicação com um cliente genérico via *socket*. O código foi alterado para que o resultado gerado pelo Julius, também conhecido como sentença, seja retornado ao cliente através desse mesmo *socket*. Além disso, uma aplicação foi construída sobre a plataforma Android 2.3 exclusivamente para se comunicar com o servidor. Basicamente, as amostras de áudio obtidas pelo microfone do aparelho são enviadas, enquanto são paralelamente analisadas a fim de se detectar o silêncio do fim da fala do usuário. Feito isso, o aplicativo apenas aguarda a sentença a ser enviada pelo servidor.

A construção do dicionário fonético para o PT_BR se dá por meio do *software* `lapsg2p`, o qual recebe uma lista de palavras como entrada e gera suas transcrições fonéticas, conforme visto na lista abaixo, à direita. Já a gramática é utilizada para restringir o vocabulário, de modo a gerar somente uma das sentenças listadas, como mostrado na lista abaixo, à esquerda. A construção da gramática no formato do Julius utiliza diretamente o dicionário fonético em seu escopo.

<s> aumentar volume </s>	aumentar	a u ~ m e ~ t a X
<s> diminuir volume </s>	diminuir	dZ i ~ m i ~ n u j X
<s> canal mais </s>	volume	v o l u ~ m i
<s> canal menos </s>	canal	k a n a w
5 <s> ligar televisão </s>	5 mais	m a j s
<s> desligar televisão </s>	menos	m e ~ n u s
<s> cadastrar controle </s>	televisão	t e l e v i z a ~ w ~
<s> selecionar controle </s>	...	

A proposta do trabalho é adicionar funcionalidades ao código do Julius, permitindo a produção de voz sintetizada através da incorporação da API do eSpeak e a transmissão de informação para a TV através de um led IR conectado a um GPIO. Um sensor IR ficará encarregado de receber informações de diferentes controles remotos para que sejam guardadas como registros no banco de dados.

Toda a organização do trabalho será feita através da plataforma Trello [?].

Predro: Detalhes da modelagem do DB

6 Orçamento

Produto	USD (U\$)	BRL (R\$)	IOF (R\$)	Total (R\$)
BBB				
Smartphone				
IR Led				
IR Sensor				
USB Speaker 8 Ω				
Total				500 conto

7 Dificuldades e Soluções

- There's no sound output on BBB. The default audio device, as mentioned in some forums on the web, is through the HDMI connector. The same forum has said that we can disable the HDMI as audio device by changing some kernel parameters, then the second device would become the main one. In this case, the USB would be the principal sound card. The easiest way, then, would be attach an USB speaker to the connector, because the TTS would automatically output the result to that port. My point about the USB speaker:

- It must be low power consuming

- Its size must be limited. If the project was to build a thermistor plugged on the wall, we could not put a big speaker there. The circuit must be smaller than the BBB itself.

It would be nice if we could build an amplifier and put the synthesized audio through a BBB GPIO, but...

- I don't know how to manipulate the TTS output to put in that pin
 - Even If I could do that, It's hard to play a sound with PWM. A chunk of audio, size dependent of the sampling rate, should be put in that GPIO in a slot of time. And I think it's not as simple as this description.
 - The arduino codes I've found that play sound through a pin have the craziest codes I've ever seen. Almost everything is manually done.
 - I think the output is gonna be terrible and polyphonic
 - The PCM2707 DAC is around 10 USD and will also need an amplifier circuit with LM386
- BBB não tem conexão wifi e é difícil atualizar o kernel pra receber um shield/case. Além do mais, tal shield/case seria plugado na porta USB, a qual já estaria sendo usada pelo auto-falante. Portanto, a solução mais fácil foi conectar um roteador wifi à porta Ethernet da BBB através de um cabo RJ45.
 - O Angstrom é PODRE. (Gabriel)
 - O código do Arduino que hackeia as paradas é PODRE. (Thiago)
 - Acessar MySQL from C é PODRE (Pedro)

8 Trabalhos Futuros

De acordo com o relato disponível em [?]: “Dado que o meu home theater é modesto, ele requer que eu consiga manejar APENAS 6 controles remotos para a simples tarefa de assistir a um filme”. Seria maravilhoso se houvesse um controle remoto universal que permitisse acesso à TODOS os aparelhos do ambiente residencial, mesmo os que estão em cômodos diferentes do que eu estou agora. Seria mais maravilhoso que esse controle estivesse sempre com você. E que pudesse usá-lo mesmo quando estivesse fora de casa. E que fosse acessível por uma tecnologia hands free. Compre já o seu!

- Expandir para vários aparelhos, tornando a beagle beagle um servidor centralizado no ambiente doméstico
- Em cada compartimento onde houvesse um aparelho eletrônico a ser controlado, haveria um microcontrolador (a ser avaliado, preferencialmente mais barato que o arduino) capaz de controlar determinado(s) aparelhos
- A beagle beagle e todos os outros microcontroladores estariam conectados à mesma rede LAN. Somente a beagle beagle precisaria estar conectada à internet, de modo que não houvesse limitação de distância para a conexão com o smartphone.

Referências

- [1] P. Taylor, *Text-To-Speech Synthesis*. Cambridge University Press, 2009.
- [2] X. Huang, A. Acero, and H. Hon, *Spoken Language Processing*. Prentice-Hall, 2001.
- [3] *Comitê de Ajudas Técnicas. Tecnologia Assistiva*. Brasília, Brasil: Subsecretaria Nacional de Promoção dos Direitos da Pessoa com Deficiência., 2009.

- [4] “FalaBrasil: Reconhecimento de Voz para o Português Brasileiro,” Visitado em Julho, 2014. <http://www.laps.ufpa.br/falabrasil/>.
- [5] A. Siravenha, N. Neto, V. Macedo, and A. Klautau, “Uso de regras fonológicas com determinação de vogal tônica para conversão grafema-fone em Português Brasileiro,” *7th International Information and Telecommunication Technologies Symposium*, 2008.
- [6] “Perfis do censo demográfico. Visitado em Janeiro,” 2010. www.ibge.gov.br/.
- [7] “Tutorial: Create Acoustic Model,” Visitado em Abril, 2015. <http://www.voxforge.org/>.
- [8] S. e. Young, *The HTK Book*. Microsoft Corporation, Version 3.0, 2000.
- [9] “Open-Source Large Vocabulary CSR Engine Julius,” Visitado em Julho, 2014. http://julius.sourceforge.jp/en_index.php.
- [10] “eSpeak text to speech,” Visitado em Julho, 2014. <http://espeak.sourceforge.net/>.
- [11] “Raspberry Pi or Beaglebone Black,” Visitado em Abril, 2015. <http://michaelhleonard.com/>.
- [12] “Arduino vs. Raspberry Pi vs BeagleBone,” Visitado em Abril, 2015. <http://randomnerdtutorials.com/>.
- [13] “Arduino Uno vs BeagleBone vs Raspberry Pi,” Visitado em Abril, 2015. <http://makezine.com/>.
- [14] “Como funciona o controle remoto?,” Visitado em Abril, 2015. <http://mundoestranho.abril.com.br/>.
- [15] “Como funcionam os controles remotos,” Visitado em Abril, 2015. <http://tecnologia.hsw.uol.com.br/>.

A Pedro: Codigos LAMP

B Thiago: Codigos Arduino/C for IR Tx/Rx