DESENVOLVIMENTO DE BIBLIOTECA PARA REPRODUÇÃO SINTETIZADA DE VALORES NUMÉRICOS DECIMAIS EM LINGUA PORTUGUESA IMPLEMENTÁVEL EM SISTEMAS EMBARCADOS DE ARQUITETURA OITO BITS.

Adriano Regis1, Roberto Alexandre Dias[2], Kallin Mansur da Costa[3]
[1]IFSC / DAMM / Campus Florianópolis/Email: adriano.regis@ifsc.edu.br
[2]IFSC / DAMM / Campus Florianópolis/ Email: roberto@ifsc.edu.br
[3]IFSC / DAMM / Campus Florianópolis/ Email: kallinmansur@gmail.com

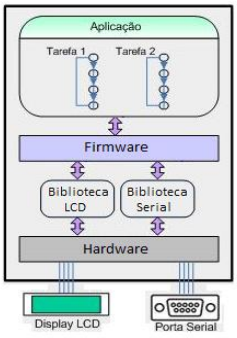
Resumo: *Esse artigo apresenta o trabalho de desenvolvimento de uma biblioteca em Linguagem C++ para sistemas embarcados que permite sintetizar frases, em língua portuguesa, com o auxilio de fonemas pré-gravados em um cartão de memória do tipo Secure Digital (SD Card), permitindo a reprodução de números decimais inteiros ou fracionários por meio da fala tomando como base uma variável de ponto flutuante de um programa principal. Serão apresentadas uma introdução aos sistemas embarcados, a teoria da reprodução de áudio em sistemas digitais, o as características da plataforma eletrônica escolhida para desenvolvimento e validação do projeto, a utilização de mostradores em projetos, a inserção da reprodução de áudio na acessibilidade, as soluções similares já desenvolvidas e suas limitações, o funcionamento da modulação por largura de pulsos, como o sinal de áudio é gerado por meio da modulação de código de pulso, a integração com cartão de memória, os resultados e testes experimentais. O sistema é demonstrado através da utilização da biblioteca em um programa gerado na plataforma Arduino, que reproduz sequencialmente valores numéricos inteiros de zero a vinte em ordem crescente. Ao final desse trabalho são apresentadas potenciais aplicações dessa biblioteca em projetos diversos substituindo os tradicionais mostradores numéricos e em soluções de tecnologia assistiva.*

Palavras-Chave: *Sintetização de voz, Sistemas embarcados, Arduino*

1 INTRODUÇÃO

Sistemas embarcados são estruturas de eletrônica digital capazes de agregar funções computacionais e de interação a um determinado produto ou equipamento por meio do processamento de informações. O barateamento de circuitos eletrônicos tem permitido a aplicação desses sistemas equipamentos e bens de consumo como eletrodomésticos, sensores, sistemas de monitoramento, entre outros. Uma arquitetura clássica de sistema embarcado pode ser visto na figura 1.

Figura 01 – Arquitetura de um sistema embarcado



Segundo Wolf (2011, p.2) os sistemas embarcados distinguem-se dos sistemas computacionais de propósitos gerais por não possuírem uniformidade já que aplicações com diferentes requisitos de desempenho, consumo de potência e área ocupada resultam em uma combinação distinta de módulos de hardware e software. Essa personalização garante o desenvolvimento de soluções otimizadas e de custo efetivo, porém a falta de padronização em nível de plataforma torna-se inconveniente nas fases de concepção e prototipagem.

A plataforma Arduino é uma proposta de sistema embarcado para prototipagem que tem sido largamente utilizada por ser amigável ao usuário iniciante e altamente configurável para projetos de usuários mais avançados, uma vez que ela permite um grau de padronização de hardware e software. O Arduino é um projeto aberto, licenciado com atribuição de compartilhamento *Share-Alike* (CREATIVECOMMONS, 2012), e foi concebido para permitir que profissionais e entusiastas de diversas áreas de conhecimento possam desenvolver protótipos funcionais. Existem atualmente quinze versões oficiais de Arduino, a tabela 1 apresenta um comparativo de características e recursos das versões mais populares.

Tabela 01 – Tabela comparativa entre placas de Arduino oficiais

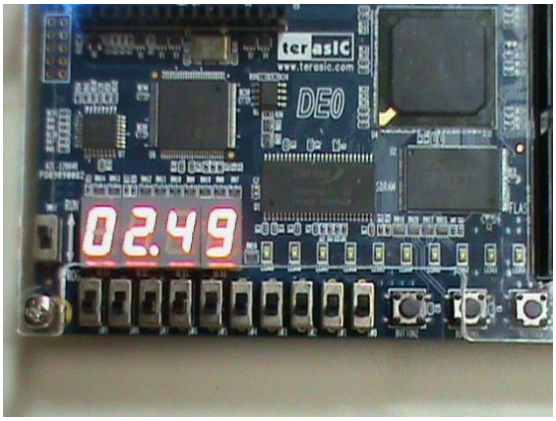
Placa	**Processador**	**Pinos de I/O**	**Saídas PWM**	**Entradas Analógicas**
Duemilanove	Atmega328P	14	6	6
Mega	Atmega2560	54	14	16
Uno	Atmega328P	14	6	6
Leonardo	Atmega32u4	14	6	12
Lily Pad	Atmega168V	14	6	6
Diecimila	Atmega168V	14	6	6

Fonte: www.arduino.cc

A filosofia de desenvolvimento com Arduino permite que sejam integradas soluções externas de hardware de maneira relativamente rápida, devido à disponibilização funções em software para tratamento de dispositivos sob a forma de bibliotecas que, umas vez utilizadas, permitem um alto grau de abstração para o programador do sistema embarcado (ARDUINO, 2012). O Ambiente de Desenvolvimento Integrado-IDE do Arduino já disponibiliza bibliotecas nativas para integração das principais soluções de hardware de mercado.

Um dos dispositivos de hardware mais utilizado em projetos de sistemas embarcados são os mostradores numéricos, pois permitem que valores processados ou adquiridos (como valores de sensores ou contagens) sejam exibidos para o usuário. Os mostradores mais utilizados são do tipo LCD e Led de sete segmentos, um exemplo de mostrador numérico integrado a um protótipo pode ser visto na figura 2.

Figura 02 –Mostrador do tipo sete segmentos



Fonte: Projeto FPGA para todos (fpgaparatodos.com.br)

No entanto, a disponibilização de informações sob a forma visual nem sempre é adequada para projetos que exigem cuidados com a luminosidade do ambiente ou atenção do usuário em condições de operação, além de limitar (ou impedir) a utilização por pessoas com deficiência visual.

Atualmente observa-se uma crescente a oferta de produtos com tecnologia embarcada onde a disponibilização de informações faz-se também por voz sintetizada. Essa tecnologia já pode ser encontrada em veículos, caixas de autoatendimento, entre outros. No entanto, a reprodução de áudio ainda é uma funcionalidade que exige alto poder de processamento e memória, e portando de alto custo. Devido a inexistência de uma solução simples para disponibilização de informações numéricas faladas em plataforma embarcada de 8 bits, esse artigo descreve o desenvolvimento de uma biblioteca que transforma valores digitais de ponto flutuante em fonemas numéricos decimais em língua portuguesa, modulados em PWM e disponível por pinos de I/O.

2 METODOLOGIA

Durante o projeto preliminar de pesquisa *"Plataforma de Automática de Sondagem em Altitude, para uso em monitoramento atmosférico"*, do Campus Florianópolis do IFSC, observou-se a necessidade de desenvolver uma forma para transmissão de dados entre uma sonda de altitude (balão meteorológico) e estações terrestres que permita a massificação de pontos de acompanhamento.

Várias soluções foram propostas para a modulação dos dados, porém as características do projeto apontaram que a forma mais adequada seria através de fonia por permitir que o acompanhamento das sondagens seja realizado por radioescutas e radioamadores.

Uma vez que a plataforma em questão foi prototipada em Arduino, iniciou-se a busca por uma solução já desenvolvida por terceiros resultando em dois projetos denominadas PCM (SMITH, 2012) e SDplayWAV (LADYADA, 2012), porém ambas as soluções possuem características restritivas ao projeto.

A solução de Smith realiza a modulação de áudio inserido em um vetor localizado no código do programa. Considerando o limite de tamanho desse tipo de memória, essa solução permite uma reprodução de no máximo 4 segundos (para os Arduinos Uno e Duemilanove).

Já a biblioteca SDplayWAV permite a reprodução de arquivos armazenados em um cartão de memória externa do tipo SD, porém o áudio é gerado por um conversor D/A externo ao Arduino (disponível sob a forma de um *Shield* comercializado pela Empresa Adafruit). Por limitações de peso e volume da sonda, a utilização de um hardware externo é indesejável.

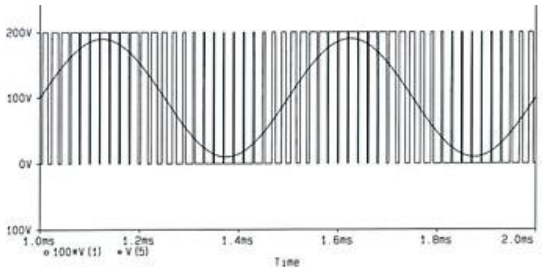
Tomando como base as características das duas soluções encontradas iniciou-se o desenvolvimento de uma nova biblioteca, cujas características são apresentadas a seguir.

2.1 Modulação em PCM

O método utilizado para modulação de áudio nesse projeto é conhecido como Pulse Code Modulation-PCM , uma vez que a forma de onda natural do som pode ser modulada por largura de pulso (PWM).

A figura 3 apresenta o sinal PWM resultante da senóide cujo sinal discretizado está representado.

Figura 03 –Sinais no domínio do tempo - PCM e Senóide discretizada



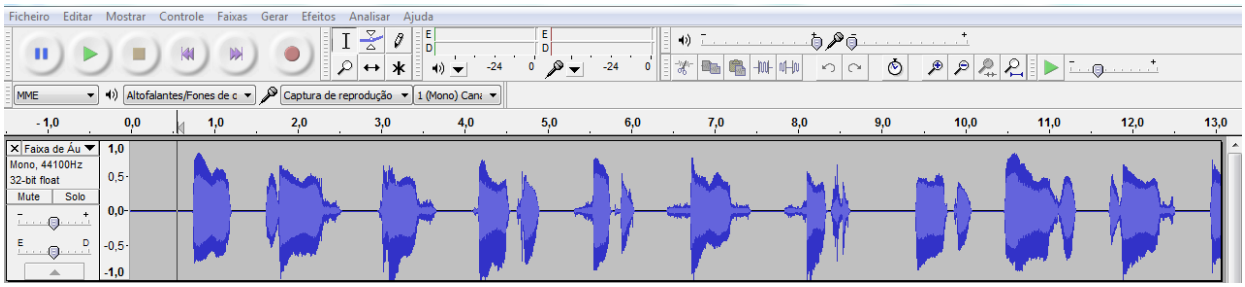
Fonte: Adam Audio

A fidelidade do sinal digitalizado em relação a uma senóide natural (analógica) é determinado principalmente pela taxa de amostragem e resolução em bits do conversor A/D no sistema de aquisição, e a qualidade da reprodução em PCM desse sinal digitalizado é determinado pela frequência de amostragem do PWM e a resolução da razão cíclica (TROFINO, 2012, p.17).

2.2 Geração de fonemas em PCM

A biblioteca acessa fonemas gravados em formato PCM armazenados em um cartão de memória do tipo SD. A gravação desses fonemas foi feita de por meio do programa Audacity (AUDACITY, 2012), um software livre que permite a captura, edição e exportação de áudio para diferentes formatos. Foram gravados sons equivalentes aos números, gerados pelo sistema Google Translate (GOOGLE, 2012), que por sua vez é baseado no programa GNU eSpeak (ESPEAK, 2012). A figura 4 apresenta a tela principal do Audacity com os fonemas numéricos adquiridos e dispostos no domínio do tempo.

Figura 04 –Tela de aquisição dos fonemas das unidades decimais



Fonte: Software Audacity

A possibilidade de visualização gráfica do áudio no Audacity permite que se editem adequadamente os fonemas, tornando-se precisa a divisão dos arquivos que posteriormente foram exportados em formato “RAW (header-less) Unsigned 8bit PCM”.

2.3 Fonemas decimais

A pronúncia de números decimais na língua portuguesa difere da representação escrita posicional, pois sua pronuncia está relacionada com as ordens numéricas adjacentes. Isso torna a separação dos fonemas mais complexa do que o desmembramento em milhares, centenas, dezenas e unidades conforme apresentado na tabela 2.

Tabela 02 – Comparação entre as representações simbólica e fonética dos números decimais

Representação simbólica	Representação fonética
0	zero
3	três
13	treze
20	vinte
23	vinte e três
100	cem
103	cento e três

Para o desenvolvimento da biblioteca criou-se arquivos equivalentes a todos os fonemas individuais em língua portuguesa, sendo esses: *zero, um, dois, três, quatro, cinco, seis, sete, oito, nove, dez, onze, doze, treze, quatorze, quinze, dezesseis, dezessete, dezoito, dezenove, vinte, vinte e, trinta, trinta e, quarenta, quarenta e, cinquenta, cinquenta e, sessenta, sessenta e, setenta, setenta e, oitenta, oitenta e, noventa, noventa e, cem, cento e, quinhentos, quinhentos e, seiscentos, seiscentos e, setecentos, setecentos e, oitocentos, oitocentos e novecentos, novecentos e, mil, mil e, vírgula.*

O algoritmo elaborado efetua o desmembramento da variável ponto flutuante e, com base nas regras fonéticas, monta uma pronuncia única correspondente ao número. Assim a variável 4513,8 resulta na reprodução sequencial dos fonemas: “quatro”, “mil”, “quinhentos e”, “treze”, “vírgula” e “oito”, por exemplo.

2.4 Reprodução de áudio por PWM

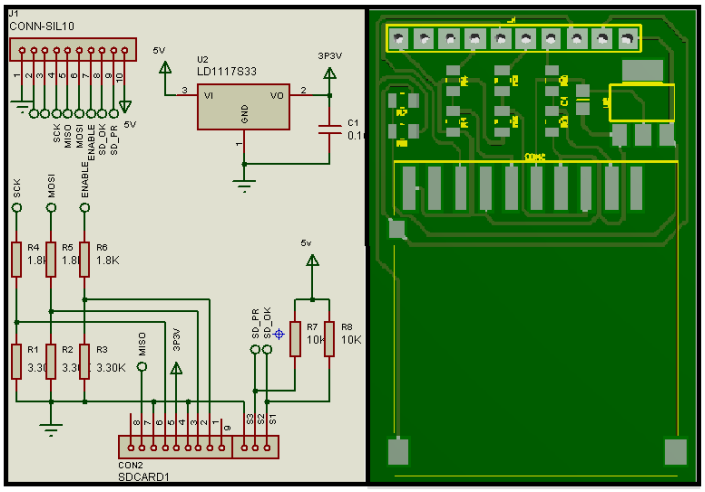
Os arquivos em formato PCM possuem os fonemas representados por uma sequencia de words cujos valores correspondem à magnitude do sinal de cada ponto da taxa de amostragem. No sistema desenvolvido esses words possuem em oito bits de resolução (256 níveis) e enviados ao gerador de PWM.

Cabe ressaltar que a frequência PWM padrão do Arduino é de 500 Hz, muito abaixo dos 3400Hz necessários para transmissão de voz humana (CLARA, 1999, p.2). A biblioteca exigiu o emprego de temporizadores extras de 16bits para utilização de um PWM de alta velocidade, capaz de gerar o áudio em uma faixa de 8kHz.

2.4 Reprodução de áudio por PWM

Diferentemente do problema encontrado na solução de Smith, o armazenamento dos dados em cartão de memória externo não impõe limitação ao tamanho dos dados e, consequentemente, dos fonemas. Cartões de memória do tipo SD são amplamente utilizados em dispositivos eletrônicos e, por isso, seu custo é muito inferior aos circuitos integrados de memória (aproximadamente um centavo de real por megabyte, conforme valores correntes em agosto de 2012). Além disso, a opção pelo formato SD deu-se devido à interface física desse dispositivo ser facilmente adaptável ao protocolo SPI já disponível no Arduino, necessitando um circuito simples para adequação dos níveis de tensão. O esquemático elaborado para esse propósito juntamente com *layout* pode ser observado na figura 5.

Figura 05 –Esquemático e *layout* da interface SD para Arduíno



Como forma de facilitar a atualização e inclusão de fonemas, os dados da memória estão dispostos em arquivos organizados no formato FAT16, utilizando-se a biblioteca SdFatLib desenvolvida por William (GREIMAN, 2012).

3 RESULTADOS E DISCUSSÃO

Como forma de validar o desenvolvimento e a funcionalidade da biblioteca desenvolvida (denominada de SDrawPCM) foi proposto um programa que reproduza sequencialmente os valores numéricos de 0 a 20 de forma crescente, a figura 6 apresenta a tela da IDE do Arduino com esse programa redigido.

Figura 06 –Código de um contador decimal com base na biblioteca RawPCM

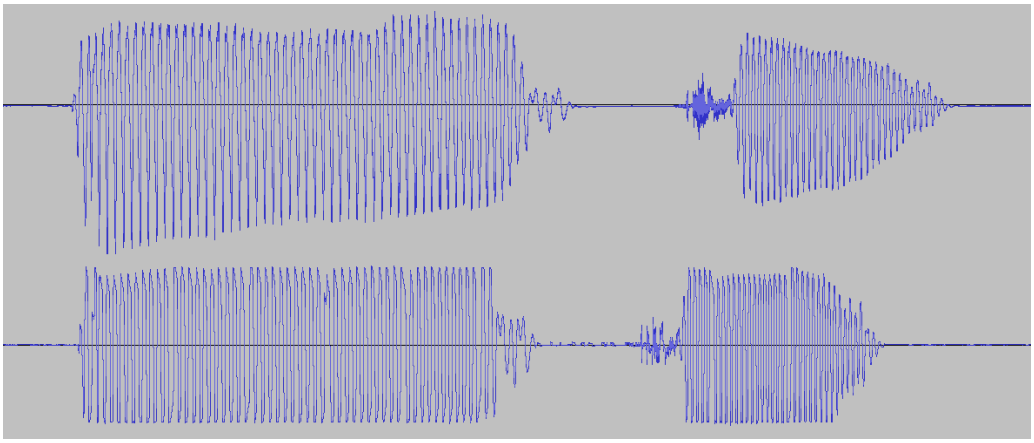
É possível observar no código que, depois de declaradas a biblioteca e a classe, a reprodução do número contido em contagem é feito apenas através da instrução “Wave.Number(Contagem,0)”. O parâmetro “0” corresponde ao número de casas pronunciadas após a vírgula (em que se tratando de uma contagem de inteiros, despreza-se).

3.1 Análise de desempenho áudio

Esse programa foi gravado em uma placa Arduino modelo Duemilanove, a fim de analisar-se a qualidade do áudio. A reprodução do som é obtida no pino 3 do Arduino, bastando apenas a ligação desse à um circuito amplificador ou diretamente a um alto falante de 8 ohms com um capacitor de acoplamento em série de 100nF.

A figura 7 apresenta as formas de onda do número “oito” gerada pelo eSpeak (acima) e o sinal gerado no Arduino e capturado no PC (abaixo).

Figura 07 – Pronuncia do numeral 8 em eSpeak (acima) e no arduino (abaixo)



A composição dos dois sinais é semelhante, apresentando apenas uma diferença de velocidade de reprodução de cerca de dez por cento. Encontra-se disponível no endereço http://n-1.nersd.ifsc.edu.br/?p=79 um vídeo demonstrando esse sistema em funcionamento.

4 CONSIDERAÇÕES FINAIS

Através dos resultados apresentados observa-se que o desempenho da biblioteca desenvolvida atende as necessidades de reprodução em voz sintetizada de forma compreensível. O exemplo apresentado nesse trabalho demonstra que a utilização da biblioteca é suficientemente simples para ser utilizada por usuários de todos os níveis de domínio de software e hardware, permitindo que sejam elaborados facilmente novos projetos, bem como rápidas adaptações em projetos já disponíveis e que utilizam mostradores visuais.

A priorização de soluções em software e integração com dispositivo de memória SD torna essa abordagem de baixíssimo custo de implementação que, aliado à rápida integração e prototipação, permite uma redução de custos drástica no desenvolvimento de tecnologias assistivas para pessoas com deficiência visual.

AGRADECIMENTOS

A equipe executora desse projeto agradece em especial ao Instituto Federal de Santa Catarina, cuja infraestrutura disponibilizada viabilizou todo o desenvolvimento.

REFERÊNCIAS

WOLF, Marilyn. **Computers As Components: Principles of Embedded Computing System Design.** Elsevier, 2012.

CREATIVE COMMONS. **Creative Commons – Attribution-ShareAlike 3.0 Unported – CC BY-SA 3.0.** Disponível em < http://www.creativecommons.org/licenses/by-sa/3.0/ >. Acesso em 30 de agosto de 2012.

ARDUINO Team. **Arduino – HomePage.** Disponível em < http://www.arduino.cc >. Acesso em 30 de agosto de 2012.

SMITH, Michael. **Arduino Playground - PCMAudio.** Disponível em < http://www.arduino.cc/playground/Code/PCMAudio >. Acesso em 30 de agosto de 2012.

LADYADA. **Audio Shield for Arduino.** Disponível em < http://www.ladyada.net/make/waveshield/index.html >. Acesso em 30 de agosto de 2012.

TROFINO, Alexandre. **Sistemas Lineares.** Santa Catarina, 1999. (Apostila).

AUDACITY Team. **Audacity: Editor e gravador de áudio livre.** Disponível em < http://audacity.sourceforge.net >. Acesso em 30 de agosto de 2012.

GOOGLE Team. **Google Tradutor.** Disponível em < http://translate.google.com.br >. Acesso em 30 de agosto de 2012.

DUDDINGTON, Jonathan. **eSpeak: Speech Synthesizer.** Disponível em < http://espeak.sourceforge.net >. Acesso em 30 de agosto de 2012.

CLARA, Ana. **Teleprocessamento.** 1999. (Apostila).

GREIMAN, Bill. **Sdfatlib – A FAT16/FA32 Arduino library for SD/SDHC cards.** Disponível em < http://code.google.com/p/sdfatlib/ >. Acesso em 30 de agosto de 2012.