

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ  
BACHARELADO EM ENGENHARIA DE COMPUTAÇÃO

LEONARDO WINTER PEREIRA  
LUCAS ZIMMERMANN CORDEIRO  
LUÍS FELIPE MAZZUCHETTI ORTIZ

**DALLE PAD**

ARTIGO ACADÊMICO

**CURITIBA**

**2016**

LEONARDO WINTER PEREIRA  
LUCAS ZIMMERMANN CORDEIRO  
LUÍS FELIPE MAZZUCHETTI ORTIZ

## **DALLE PAD**

Artigo Acadêmico apresentado pelo Bacharelado em Engenharia de Computação da Universidade Tecnológica Federal do Paraná como requisito parcial para aprovação na disciplina de "Oficinas de Integração 3".

Orientador: Gustavo Benvenuto Borba  
Guilherme Alceu Schneider

**CURITIBA**

**2016**

AQUI A DEDICATÓRIA

## **AGRADECIMENTOS**

AQUI OS AGRADECIMENTOS

”A geometria é uma ciência de todas as espécies possíveis de espaços.”(Kant)

## **RESUMO**

WINTER PEREIRA, Leonardo; ZIMMERMANN CORDEIRO, Lucas; MAZZUCHETTI ORTIZ, Luís F.. DALLE PAD. 49 f. Artigo Acadêmico – Bacharelado em Engenharia de Computação, Universidade Tecnológica Federal do Paraná. Curitiba, 2016.

Resumo (Máximo de 500 palavras).

**Palavras-chave:** Arduino, Android, Projeto, Gerenciamento

## **ABSTRACT**

WINTER PEREIRA, Leonardo; ZIMMERMANN CORDEIRO, Lucas; MAZZUCHETTI ORTIZ, Luís F.. DALLE PAD. 49 f. Artigo Acadêmico – Bacharelado em Engenharia de Computação, Universidade Tecnológica Federal do Paraná. Curitiba, 2016.

Abstract text (maximum of 500 words).

**Keywords:** Arduino, Android, Project, Management

## LISTA DE FIGURAS

FIGURA 2.1 – Exemplo de transmissão das mensagens de Nota ON e OFF .....	20
FIGURA 2.2 – Comparação entre transmissão serial e paralela .....	21
FIGURA 2.3 – Conector MIDI .....	22
FIGURA 2.4 – Conectores fêmeas MIDI .....	23
FIGURA 2.5 – Exemplo de canais MIDI .....	24
FIGURA 2.6 – Byte de estado e de dados .....	25



## LISTA DE TABELAS

TABELA 2.1 – Pinos em um conector MIDI .....	22
TABELA A.1 – Notas musicais em MIDI .....	32
TABELA B.1 – Lista de Bytes de estado para a função Note Off .....	34
TABELA B.2 – Lista de Bytes de estado para a função Note On .....	34
TABELA B.3 – Lista de Bytes de estado para a função Polyphonic Aftertouch .....	35
TABELA B.4 – Lista de Bytes de estado para mudança de controle / modo de canal ...	36
TABELA B.5 – Lista de Bytes de estado para a função mudança de programa .....	36
TABELA B.6 – Lista de Bytes de estado para a função Channel Aftertouch .....	37
TABELA B.7 – Lista de Bytes de estado para a função de controle do Pitch Wheel ....	38
TABELA B.8 – Lista de Bytes de estado para as funções de sistema .....	38
TABELA B.9 – Lista de mensagens de mudança de controle / modo de canal .....	43

## LISTA DE SIGLAS

USB	Universal Serial Bus
MIDI	Musical Instrument Digital Interface
SMF	Standard MIDI Files
bps	<i>bits</i> por segundo
GND	Ground
Thru	Through
MSB	Most Significant Bit
MMA	MIDI Manufacturers Association

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>12</b>
1.1	TEMA	12
1.2	DELIMITAÇÃO DO ESTUDO	14
1.3	PROBLEMA	14
1.4	OBJETIVOS	14
1.4.1	Objetivos Gerais	14
1.4.2	Objetivos Específicos	15
1.5	JUSTIFICATIVA	15
1.6	PROCEDIMENTOS METODOLÓGICOS	15
1.7	EMBASAMENTO TEÓRICO	16
1.8	ESTRUTURA DO TRABALHO	16
1.9	BANCA EXAMINADORA	17
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b>	<b>18</b>
2.1	MIDI	18
2.1.1	O que é MIDI	18
2.1.2	O protocolo de comunicação MIDI	20
2.1.3	Hardware e conectores	22
2.1.4	Formatos de distribuição	23
2.1.5	Canais MIDI	24
2.2	MENSAGENS MIDI	25
2.2.1	Mensagens de estado e de dados	25
2.3	MICROCONTROLADORES E ARDUINO	26
2.3.1	Microcontroladores	26
2.3.2	Arduino	26
<b>3</b>	<b>DESENVOLVIMENTO</b>	<b>27</b>
3.1	HARDWARE	27
3.2	ESTAÇÃO BASE PRINCIPAL	27
3.2.1	Interface	27
3.2.2	Lógica	27
3.3	ESTAÇÃO BASE SECUNDÁRIA	27
3.3.1	Interface	27
3.3.2	Lógica	27
3.4	COMUNICAÇÃO ENTRE HARDWARE E SOFTWARE	27
3.5	PROJETO MECÂNICO - INVÓLUCRO	27
3.5.1	Programas Utilizados para a execução do Projeto	27
<b>4</b>	<b>RESULTADOS E DISCUSSÕES</b>	<b>28</b>
<b>5</b>	<b>CONSIDERAÇÕES FINAIS</b>	<b>29</b>
5.1	SUGESTÕES PARA TRABALHOS FUTUROS	29
	<b>REFERÊNCIAS</b>	<b>30</b>
	<b>Apêndice A – NOTAS MUSICAIS EM MIDI</b>	<b>32</b>
	<b>Apêndice B – ESPECIFICAÇÃO MIDI</b>	<b>33</b>

B.1 BYTES DE ESTADO .....	33
B.2 BYTES DE DADOS .....	39
<b>Anexo A – DATASHEETS .....</b>	<b>44</b>
A.1 POTENCIÔMETRO ROTATIVO .....	45
A.2 POTENCIÔMETRO LINEAR .....	47

## 1 INTRODUÇÃO

**DALLE PAD - O Gadget que te transforma em um DJ** foi desenvolvido para a disciplina de Oficina de Integração 3 (IF66J - S71), do curso de Engenharia de Computação da Universidade Tecnológica Federal do Paraná.

Como proposto pela disciplina, o sistema é composto por uma estação base principal (desenvolvida para *Desktop e notebook*) e outra secundária (desenvolvida para *mobile*), um sistema de comunicação e um sistema embarcado. Em ambas as estações base encontram-se os softwares de interface com o usuário; o sistema de comunicação é baseado em tecnologia sem fio, além de conexão USB e MIDI; o sistema embarcado consiste em um sistema microcontrolado e a estrutura física do produto é composta por um invólucro mecânico contendo todo o sistema microcontrolado e toda a estrutura física necessária para que o usuário final possa utilizar tudo o que o **DALLE PAD** permite.

Este capítulo divide-se em nove seções. Na Seção 1.1 e 1.2, apresentam-se o tema e a delimitação do projeto, respectivamente. Na Seção 1.3, enuncia-se o problema cuja solução é proposta neste trabalho. Na Seção 1.4, enunciam-se os objetivos geral e específicos. A justificativa e o procedimento metodológico adotados são descritos, respectivamente, nas seções 1.5 e 1.6. Na Seção 1.7 é apresentado o embasamento teórico, na Seção 1.8 é apresentada a estrutura do restante do trabalho e por fim, na Seção 1.9, é apresentada a banca examinadora e pessoas convidadas para a defesa do trabalho.

### 1.1 TEMA

A música é a arte de combinar sons de maneira agradável ao ouvido e a sensibilidade emocional utilizando elementos como melodia, harmonia e ritmo. Atualmente, não se conhece nenhuma civilização ou agrupamento que não possua manifestações musicais próprias. A criação, o desempenho, o significado e até mesmo a definição de música variam de acordo com a cultura e o contexto social, como composições fortemente organizadas e improvisadas.

A música expandiu-se ao longo dos anos, e atualmente encontra-se em diversas utilidades, não só como arte, mas também como a militar, educacional ou terapêutica (musicoterapia). Além disso, tem presença central em diversas atividades coletivas, como os rituais religiosos e festas. (PPD, 2011)

Os instrumentos musicais até o século XIX baseavam-se em um mesmo princípio de produção sonora, todo som era proveniente da vibração de algum material elástico (as cordas do violão e do piano, por exemplo) que gerava ondas que se propagam pelo ar até atingirem o sistema auditivo do ouvinte. Entretanto, o surgimento de novas tecnologias baseadas na eletricidade e no uso de sinais eletromagnéticos abriu a possibilidade da geração de sons artificiais, sem a utilização de instrumentos mecânicos. Embora as ondas que atingem os ouvidos possuam a mesma natureza, a sua produção é radicalmente diferente. (SANTINI, 2005)

Para alguns indivíduos, a música está extremamente ligada à sua vida, mas a dificuldade de manter um grupo musical unido é muito grande, sem contar o custo elevado de determinados instrumentos e outros equipamentos. Atualmente, pode-se contar com a Tecnologia MIDI.

Desde seu lançamento no mercado no início da década de 1980, o protocolo MIDI tem tido um papel de grande importância na indústria da música. Entretanto, mais importante ainda é que este trouxe para músicos e entusiastas uma ferramenta que lhes permitiu preencher a lacuna antes existente. MIDI permitiu, pela primeira vez, um meio de comunicação de informações musicais de um dispositivo para outro de uma forma que foi aceito e adotado por toda uma indústria. (GUERIN, 2009)

Alguns anos mais tarde, em meados da década de 1990, alguns acreditavam que MIDI não tinha mais futuro. Estações de trabalho de áudio digital foram se tornando cada vez mais acessíveis e computadores passaram a oferecer um poder de processamento cada vez maior, que o uso de MIDI foi quase considerada uma coisa do passado, lento demais para continuar sendo utilizado. Entretanto, não foi isso o que aconteceu.

Este protocolo continua sendo amplamente utilizado na área musical, e é de interesse deste projeto compreender a teoria por trás do mesmo, não somente sua utilização no produto final, o controlador DALLÉ PAD.

Inspirado em diversos produtos já existentes, o protótipo aqui apresentado será capaz de realizar todas as principais funções de um controlador MIDI (a ser explicitado no decorrer deste documento).

## 1.2 DELIMITAÇÃO DO ESTUDO

Este trabalho busca disponibilizar as informações necessárias sobre o protocolo MIDI para os leitores interessados, incluindo sua importância, utilidade e formas de utilização, além de explicar a teoria por trás de todos os conceitos utilizados neste protótipo. Desta forma, até mesmo o leitor leigo na área de computação musical poderá acompanhar este trabalho sem maiores problemas.

Com o objetivo de desenvolver um produto acessível para o usuário final, procurou-se componentes de baixo custo e interfaces gráficas para trabalhar em conjunto com o mesmo que fossem comuns e de fácil aquisição: computadores, *notebooks*, *smartphones* e *tablets*.

## 1.3 PROBLEMA

O trabalho consiste na confecção de um controlador MIDI e sua integração à um sistema através do uso das comunicações *bluetooth*, USB e MIDI. Com o objetivo principal do trabalho sendo adquirir experiência e aprendizado quanto aos elementos envolvidos, o problema do projeto se limita à utilização do *bluetooth* e ao custo total do projeto. Ao longo do desenvolvimento do projeto, o problema que se procura resolver é: Existe como confeccionar um controlador MIDI que utilize as comunicações citadas e que possua custo acessível?

A primeira etapa do desenvolvimento do projeto se baseia na escolha dos componentes e sua montagem, procurando a solução quanto aos custos totais. Já na segunda fase, o desenvolvimento da integração com um sistema irá permitir avaliar a viabilidade do *bluetooth* no projeto.

## 1.4 OBJETIVOS

Nesta seção são apresentados os objetivos geral e específicos do trabalho, relativos ao problema anteriormente apresentado.

### 1.4.1 OBJETIVOS GERAIS

Desenvolver um controlador MIDI capaz de exercer todas as principais funções impostas a ele no meio musical, através de um dispositivo que possua sistema operacional *Android* e comunicação por *bluetooth* ou através de um computador ou *notebook* que possua o sistema operacional *Windows*, utilizando-se das comunicações USB e MIDI.

#### 1.4.2 OBJETIVOS ESPECÍFICOS

- Estudo profundo do protocolo MIDI;
- Projetar e montar o protótipo do controlador aqui proposto;
- Desenvolver um aplicativo para *Android* capaz de se comunicar com o mesmo;
- Desenvolver um software para *Windows* capaz de trabalhar com arquivos MIDI e de se comunicar com o controlador;
- Implementação do protocolo de comunicação entre as estações base e o controlador.

#### 1.5 JUSTIFICATIVA

O controlador MIDI surgiu para conectar instrumentos musicais eletrônicos, computadores e outros dispositivos relacionados através de uma simples conexão MIDI. Esse produto auxilia músicos amadores e profissionais na edição e criação de músicas, incentivando-os à utilizar sua criatividade com os novos tipos de efeitos e funcionalidades que estão contidos no software utilizado.

O desenvolvimento desse produto no Brasil é limitado e seu preço no mercado pode ser alto para usuários amadores. Ao inserirmos um novo tipo de comunicação no dispositivo, permitimos outra maneira de acessá-lo e um software para controle básico de suas funções. Ao mesmo tempo, procuramos diminuir o custo do projeto, tornando-o acessível para o mercado e, principalmente, às pessoas que desejam começar a aprender esse ramo da música.

#### 1.6 PROCEDIMENTOS METODOLÓGICOS

A natureza deste trabalho é de pesquisa aplicada, pois gerará um protótipo de aplicação prática utilizando conhecimento teórico já existente, dirigido a um problema específico. A abordagem é qualitativa, envolvendo testes do projeto em ambiente simulado.

O seu desenvolvimento é organizado em três principais estágios: pesquisa bibliográfica, implementação e experimentação. Na fase de pesquisa bibliográfica, são considerados os resultados de trabalhos passados que se mostraram relevantes para a implementação do presente projeto, desde os conceitos mais abrangentes (e.g.: programação em Android) até os mecanismos básicos (e.g.: protocolo MIDI). Será dada uma atenção fundamental ao estudo do protocolo MIDI para o desenvolvimento desse projeto.



No estágio de implementação serão realizados diagramas esquemáticos do hardware e software que serão desenvolvidos com base nos conceitos estudados na fase de pesquisa bibliográfica. Assim que os esquemáticos forem finalizados, a compra dos materiais para o hardware e o início do desenvolvimento dos softwares poderão ser iniciados. Ao obtermos todos os materiais necessários para o hardware, será possível iniciar o projeto da PCB e dos componentes necessários através do diagrama esquemático.

Após finalizada a implementação do projeto, a fase de experimentação será iniciada. Vários testes individuais com o hardware e os softwares serão realizados, assim como testes envolvendo a conexão dos dois componentes. Assim, haverá espaço para ajustes e verificação do projeto antes da montagem do produto final.

## 1.7 EMBASAMENTO TEÓRICO

Para que seja possível a execução deste projeto, diversas referências serão utilizadas.

Referente ao protocolo MIDI, serão utilizados como referencial teórico, principalmente, (ALVES, 2009), (HEWITT, 2008), (COLBECK, 2016), (GUERIN, 2009), (MC-GUIRE, 2013) e (HUBER, 2012), mas diversos outros trabalhos serão citados no decorrer deste trabalho.

Referente às estações base, os principais referenciais teóricos são, além dos já citados anteriormente, (BALLOU, 2015), (GREGOIRE, 2014), (JACKSON, 2013), além de (ANDROID, 2015).

Por último, para o sistema microcontrolado, as principais referencias são (WHEAT, 2011), (BAYLE, 2013), (GHASSAEI, 2015) e (HASS, 2013), além de (ARDUINO, 2014a), (ARDUINO, 2014b).

## 1.8 ESTRUTURA DO TRABALHO

O trabalho terá a estrutura abaixo apresentada:

- Capítulo 1 - Introdução: são apresentados o tema, as delimitações da pesquisa, o problema e a premissa, os objetivos da pesquisa, a justificativa, os procedimentos metodológicos, as indicações para o embasamento teórico e a estrutura geral do trabalho.
- Capítulo 2 - Fundamentação Teórica: são apresentados os conceitos e equipamentos necessários para a construção do Dalle Pad.

- Capítulo 3 - Desenvolvimento: é apresentado o funcionamento do Hardware e Software do Dalle Pad, bem como a comunicação entre ambas as partes.
- Capítulo 4 - Resultados e Discussões: são apresentados os resultados obtidos e discussões pertinentes.
- Capítulo 5 - Considerações Finais: serão retomadas a pergunta de pesquisa e os seus objetivos e apontado como foram solucionados, respondidos, atingidos, por meio do trabalho realizado. Além disto, serão sugeridos trabalhos futuros que poderiam ser realizados a partir do estudo realizado.

## 1.9 BANCA EXAMINADORA

Durante toda a execução deste projeto, diversos alunos e professores foram de extrema ajuda e importância.

É com grande alegria que nomeio alguns destes para participar da banca examinadora do projeto:

- Aluno(s) convidado(s):  
João Pedro Curti  
André Eleutério
- Professor orientador:  
César Manuel Vargas Benitez (DAELN)  
Rafael Barreto (DAFIS)
- Professor(a) convidado(a):  
Leyza Dorini (DAINF)  
Fábio Dorini (DAMAT)
- Professor(es) da disciplina:  
Gustavo Benvenuto Borba (DAELN)  
Guilherme Alceu Schneider (DAELN)

## 2 FUNDAMENTAÇÃO TEÓRICA

Para uma melhor compreensão e análise do funcionamento e princípios envolvidos em um controlador MIDI, é fundamental entendermos todos os conceitos nesse processo, bem como as características das tecnologias utilizadas na construção do protótipo e de ambos os aplicativos de controle. Assuntos, esses, que serão tratados no decorrer dessa seção.

### 2.1 MIDI

Uma vez que o projeto aqui proposto baseia-se, basicamente, no funcionamento deste protocolo e na troca de informações deste com outros dispositivos, é imprescindível um completo entendimento do mesmo.

Esta será a mais longa seção deste capítulo, mas também a mais importante. Além disso, aqui serão citados grandes nomes utilizados como referência para este projeto, para o caso de o leitor se interessar pelos conteúdos aqui expostos e quiser se aprofundar nos mesmos.

#### 2.1.1 O QUE É MIDI

O primeiro conceito que o leitor deve entender é o que é, de fato, o protocolo MIDI, citado repetidas vezes no capítulo anterior, mas que ainda não foi explicitado. Para tanto, serão citadas três definições deste e após, será estudado o real significado das mesmas.

Adam Olson, do Conservatório Shenandoah, define MIDI como:

[...] MIDI é um protocolo de controle  
que envia dados entre vários  
instrumentos musicais e sequenciadores.

---

(MCGUIRE, 2013)

Já David Miles Huber, autor do livro *The MIDI Manual* define MIDI como:

[...] Basicamente, MIDI é uma linguagem e especificação compatível de comunicação digital que permite múltiplos hardwares e softwares se comunicarem uns com os outros através de uma rede conectada.

---

(HUBER, 2012)

Robert Guérin, autor do livro *MIDI Power!*, por sua vez, define MIDI como:

[...] MIDI permite que músicos, engenheiros de som e iluminação, entusiastas da computação, ou qualquer outra pessoa usem computadores e instrumentos musicais eletrônicos para criar, ouvir e aprender sobre música, oferecendo uma linguagem comum que é compartilhado entre dispositivos e software compatíveis.

---

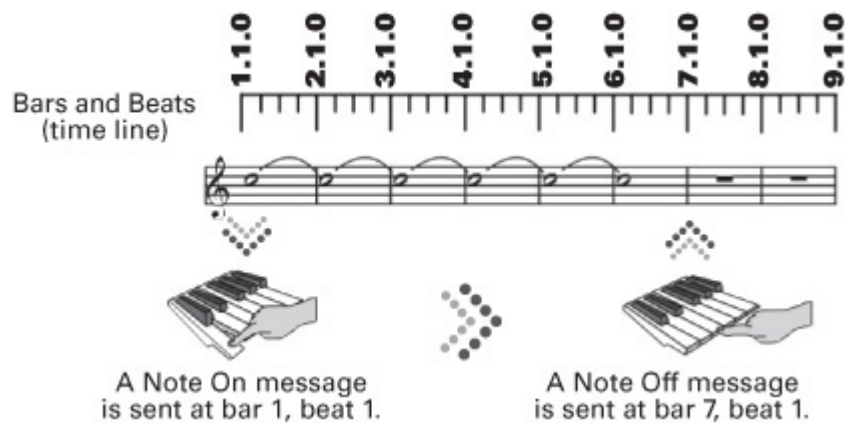
(GUERIN, 2009)

Com estas definições, pode-se perceber que MIDI é, por si só, incapaz de reproduzir som. Esta é sua maior qualidade. Seus dados podem ser gravados em um programa de software (conhecido como sequenciador) ou dispositivo de hardware, onde estes podem ser editados e transmitidos para instrumentos eletrônicos ou outros dispositivos para criar a música ou controlar qualquer parâmetro desejado. Além disso, MIDI pode ser separado em três entidades diferentes: A linguagem utilizada, também conhecida como protocolo de comunicação; a interface de hardware utilizada para transmitir e receber essas informações (tais como conectores e cabos); e os formatos de distribuição, tal como SMF (Arquivos MIDI padrão). No decorrer deste capítulo, abordaremos separadamente cada entidade, a fim de entender como elas trabalham juntas e o porquê do MIDI funcionar tão bem, oferecendo interconectividade, flexibilidade e (em termos de tamanho de arquivo) portabilidade.

Fundamentalmente, MIDI é uma linguagem de descrição de música em forma binária, em que cada palavra binária descreve um evento de uma música. Cada instrumento musical realiza um som, o qual está totalmente sobre controle do músico. O músico controla o momento em que o instrumento começará a realizar um determinado som - um tecladista pressionará uma tecla, um trompetista soprará ar e um baixista puxará uma corda. Considere esta ação como o

evento **Note ON**. A mensagem MIDI em si não contém o real som tocado pelo instrumento, mas sim a ação de tocar uma determinada nota em um determinado momento e durante um determinado período.

Quando o músico deixa de pressionar uma tecla em seu teclado ou de puxar uma corda em seu baixo, o som pára. Com MIDI, quando tal ação é realizada, uma mensagem é enviada para essa nota, para que esta deixe de realizar som. Esta mensagem é conhecida por **Note OFF**. A figura 2.1 demonstra um exemplo destes dois eventos.

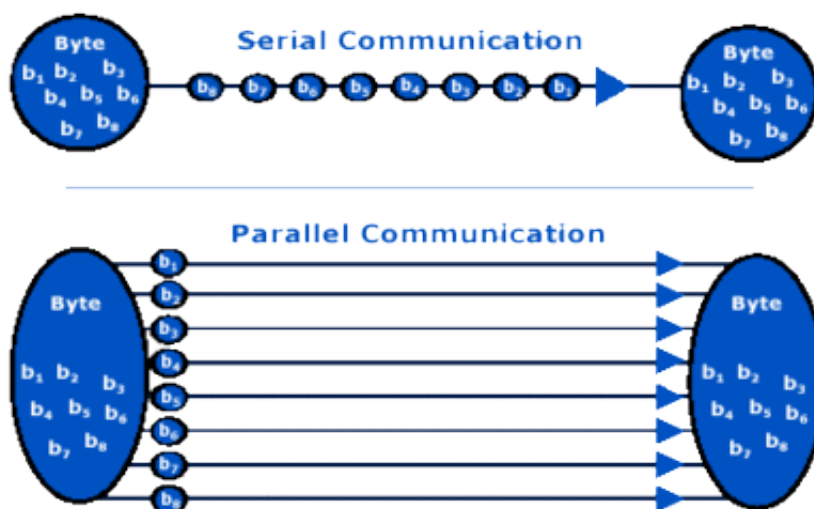


**Figura 2.1: Exemplo de transmissão das mensagens de Nota ON e OFF (GUERIN, 2009)**

A fim de identificar qual nota foi ativada / desativada no instrumento, um número é atribuído para cada nota. MIDI também lida com valores interpretativos, como por exemplo a velocidade que a nota foi pressionada e a pressão que é executada na mesma.

### 2.1.2 O PROTOCOLO DE COMUNICAÇÃO MIDI

O protocolo de comunicação MIDI é transmitido de forma serial ao invés de paralela. Em uma transmissão paralela, como o nome sugere, as informações são transmitidas simultaneamente. A quantidade de dados que podem ser transmitidos simultaneamente depende na capacidade física do fio e da velocidade com que os dispositivos podem enviar suas informações. Na figura 2.2, podemos perceber que um *byte* (oito *bits*) é transmitido simultaneamente utilizando uma transmissão paralela. A transmissão serial, por outro lado, consegue enviar apenas um *bit* após o outro.



**Figura 2.2: Comparação entre transmissão serial e paralela**

A pergunta que fica é: Por que MIDI usa transmissão serial ao invés da transmissão paralela?

A resposta para essa pergunta está no período em que o protocolo foi desenvolvido. A transmissão paralela tinha algumas desvantagens que superavam as vantagens desta sobre a transmissão serial, além de ser muito mais cara, uma vez que os fios, conectores e tomadas eram mais complexos. Já a transmissão serial, por outro lado, era muito mais simples de ser produzida, mais acessível para o público e rápida o suficiente para os fins que os fabricantes e usuários tinham em mente naquela época, para não mencionar mais confiável. Com o passar dos anos, a tecnologia avançou muito e os custos da transmissão paralela despencaram. Entretanto, para manter o MIDI compatível com todos os antigos dispositivos, sua forma de transmissão de dados permaneceu inalterada desde sua introdução no mercado.

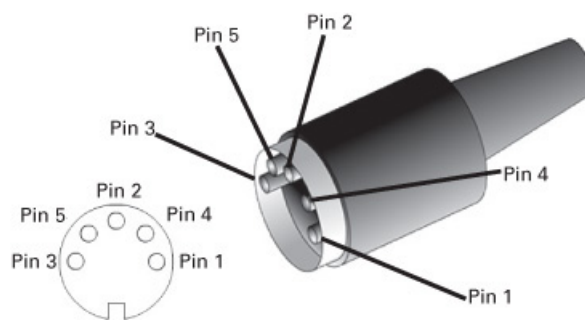
MIDI envia informação a uma taxa de 31250 bps. Essa velocidade é chamada de taxa de transmissão. Uma vez que o protocolo utiliza transmissão serial, o mesmo envia apenas um *bit* por vez. Cada *byte* em uma mensagem MIDI contém 10 *bits* de dados (8 *bits* para as informações e 2 *bits* para correção de erro). Isso significa que MIDI envia cerca de 3125 *bytes* de dados cada segundo.

Quando comparamos esse valor com a taxa de transmissão de 176400 *bytes* necessária para transmitir áudio digital (reprodução e gravação) em formato de CD de áudio, MIDI pode parecer incrivelmente devagar. Entretanto, neste último não é necessário enviar tanta informação quanto o áudio digital, sendo capaz de, teoricamente, transmitir até 500 mensagens MIDI por segundo. Na realidade, leva cerca de um milésimo de segundo para transmitir uma única nota. O limiar para distinguir os eventos de som individuais é de aproximadamente 10

milissegundos, então você apresentará dificuldades com 10 ou mais eventos simultâneos.

### 2.1.3 HARDWARE E CONECTORES

Informações MIDI são transmitidas através de fios e conectores, e seus cabos podem apresentar até 15 metros de comprimento. Entretanto, por se tratar de um cabo serial, todos os dados são transmitidos através de um único fio principal, e grandes distâncias podem inferir em uma degradação do sinal, ocasionando em possíveis perdas de dados ou mensagens impossíveis de serem lidas.



**Figura 2.3: Conector MIDI**

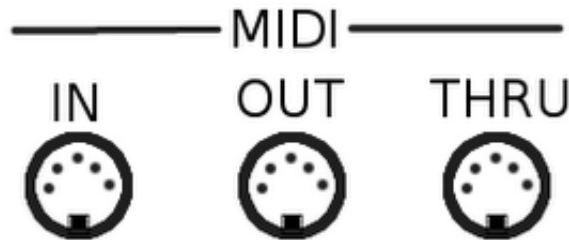
O conector MIDI, semelhante ao apresentado na figura 2.3, apresenta 5 pinos distintos. Entretanto, como já mencionado, MIDI envia informações utilizando um protocolo de transmissão serial. Desta forma, apenas um pino é realmente utilizado para o envio de informações. A tabela 2.1 descreve cada pino do conector e sua respectiva função.

Pino	Descrição
1	Não é utilizado. Na maioria dos cabos MIDI, este pino não está conectado a nenhum fio.
2	Este é utilizado para proteção elétrica ( GND). Esta proteção impede que a transmissão apresente sinais elétricos indesejados.
3	Como o Pino 1, este também não é utilizado, e na maioria dos cabos MIDI, também não está conectado a nenhum fio.
4	Este é o único receptor de dados MIDI. As informações através deste cabo fluem em uma única direção.
5	Este é o único transmissor de dados MIDI e, assim como no pino 4, as informações também fluem unidirecionalmente.

**Tabela 2.1: Pinos em um conector MIDI**

Além do conector, outro hardware importante, que vale a pena ser ressaltado, é a versão fêmea do cabo MIDI. Controladores MIDI, como o desenvolvido neste projeto, apresentam

normalmente 2 ou 3 destes conectores, rotulados como **In**, **Out** e **Thru** (veja a figura 2.4). O conjunto destes conectores é chamado de **porta MIDI** e cada uma delas será explicado com mais detalhes a seguir.



**Figura 2.4: Conectores fêmeas MIDI**

- **MIDI IN:** Através deste conector são recebidas as mensagens MIDI de um determinado dispositivo ou software MIDI. Estas mensagens podem então ser processadas e, caso desejado, enviadas para outro dispositivo através do conector MIDI Thru.
- **MIDI OUT:** Através deste conector são transmitidas as mensagens MIDI que são geradas em um software ou dispositivo MIDI. Esta saída não envia informação de áudio, mas sim simples mensagens MIDI que são interpretadas por um conector MIDI In de um outro dispositivo. Tais mensagens são códigos digitais que representam o que e como músicas e eventos são tocados em um determinado instrumento.
- **MIDI THRU:** envia uma réplica do sinal recebido na porta In para o componente ligado a ele, seja em cadeia ou somente os dois, desde que sejam componentes MIDI.

#### 2.1.4 FORMATOS DE DISTRIBUIÇÃO

Como já dito no início deste capítulo, existe um formato de arquivo padrão para MIDI, o SMF. Este tipo de arquivo armazena todas as informações necessárias para reproduzir todos os parâmetros suportados pelo protocolo. Além disso, ele também adiciona o que chamamos de *time stamp* em cada evento, para que o software MIDI, ou até mesmo o usuário final, saiba o momento correto de realizar cada um destes.

MIDI tem a vantagem de ser compacto, uma vez que o som tocado por determinado instrumento não é gravado, mas sim apenas as informações de cada evento. Devido a este fato, arquivos MIDI apresentam um tamanho cerca de 300 vezes menor do que um arquivo de



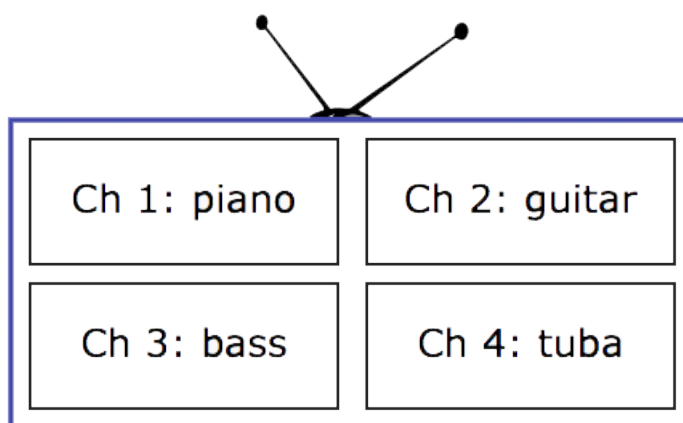
música atual (MP3). Entretanto, esse valor pode ser muito maior se compararmos com formatos descompactados de músicas.

O fato de MIDI não gravar som, entretanto, pode também ser uma desvantagem. Quando o som de um determinado instrumento é gravado, o usuário tem total controle sobre o resultado final, enquanto com MIDI, este mesmo resultado depende do dispositivo ou software que será utilizado para reproduzir este mesmo som.

### 2.1.5 CANAIS MIDI

Quando trabalhamos com música, muitas vezes desejamos que diversos instrumentos participem da mesma composição. Com MIDI, podemos incluir instruções para cada um deles através dos chamados **canais MIDI**.

Pense em um canal MIDI como um canal de televisão (veja a figura 2.5). Um determinado instrumento conecta-se apenas a um único canal, assim como um usuário o faz em sua televisão. Isso não significa que os outros canais não estão disponíveis, mas sim que o conteúdo disponível (no nosso caso, a mensagem MIDI) foi filtrado para processar apenas o relevante para aquele canal.



**Figura 2.5: Exemplo de canais MIDI (GIBSON, 2013)**

Cada porta MIDI suporta até 16 canais simultâneos. Em outras palavras, um usuário pode escolher um determinado canal para transmitir as mensagens MIDI ou possuir até 16 instrumentos conectados simultaneamente, cada um realizando diferentes eventos.

## 2.2 MENSAGENS MIDI

Agora que já explicamos o básico sobre o protocolo MIDI, é necessário estudarmos como podemos transmitir informações através de dispositivos. Essas informações são as chamadas **mensagens MIDI**.

O conteúdo de uma mensagem MIDI é bastante simples. Todas elas apresentam um componente chamado de *status byte*. Este componente é acompanhado por um valor que é definido por um ou mais *bytes* de dados.

Durante essa seção, os seguintes tópicos serão abordados:

- A estrutura de uma mensagem MIDI.
- Diferenças entre os tipos de mensagens MIDI.
- O que essas mensagens contêm e o que seus valores representam.
- Como MIDI atribui nomes e números de notas para cada evento.
- Como as mensagens MIDI transmitem e armazenam informações sobre a performance, e então as reproduzem no tempo correto.

### 2.2.1 MENSAGENS DE ESTADO E DE DADOS

Como já visto na seção anterior, cada *byte* em uma mensagem MIDI contém 10 *bits*. Destes, 8 são utilizados para a transmissão de informações. Desta forma, cada mensagem pode conter um valor entre 0 e 255. Tais mensagens são divididas em duas categorias: *Estado* e *Dados* (veja a figura 2.6).

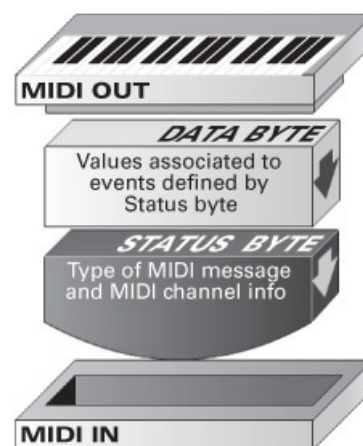


Figura 2.6: Byte de estado e de dados (GUERIN, 2009)

- A porção referente ao *byte* de estado serve para identificar o tipo de informação que está sendo enviada. Ela informa ao dispositivo receptor a qual canal MIDI determinado evento pertence e o que este evento é. Um evento pode ser, por exemplo, um Note On ou um Note Off. Outros eventos serão explicados no decorrer do projeto.
- A porção referente ao *byte* de dados informa ao dispositivo receptor o valor que está associado ao evento determinado na outra porção da mensagem MIDI. Por exemplo, se o usuário decidir tocar um *Dó* mediano com uma força também mediana, o *byte* de estado irá enviar um evento "Note On", enquanto o *byte* de dados irá passar o valor correspondente à essa nota e à velocidade que esta foi tocada. No nosso exemplo, a nota C4 tem o valor correspondente a 60 (veja a tabela A.1) e a velocidade de aproximadamente 64.

Os valores utilizados pelo *byte* de estado variam entre 128 e 255 (1000 0000 a 1111 1111 em números binários), enquanto para o *byte* de dados estes variam entre 0 a 127 (0000 0000 a 0111 1111 em números binários). É fácil perceber que, para um dispositivo MIDI reconhecer qual o tipo de mensagem que se está sendo transmitida, basta verificar o MSB (*Bit* mais significativo, do inglês). Se este valor for 1, trata-se de um *byte* de estado. Caso contrário, de um *byte* de dados. Simples assim.

Digamos que um dispositivo MIDI receba três valores: 175, 43 e 125. Ele saberá que 175 é um *byte* de estado, enquanto os outros dois valores são os *bytes* de dados que estão acompanhando o *byte* de estado. Caso este dispositivo receba outros valores, digamos: 200, 220 e 90. Ele saberá que os dois primeiros valores são *bytes* de estados, enquanto o último é um *byte* de dados que acompanha apenas o último *byte* de estado, não ambos.

Mensagens MIDI são normalmente representadas em um dos três mais comuns formatos digitais: decimais; binários; e hexadecimais. Explicar a conversão entre estes formatos (e os formatos em si) jazem fora do escopo deste projeto, mas podem ser consultados no livro *MIDI Power!* (GUERIN, 2009).

## 2.3 MICROCONTROLADORES E ARDUINO

### 2.3.1 MICROCONTROLADORES

### 2.3.2 ARDUINO

### **3 DESENVOLVIMENTO**

#### **3.1 HARDWARE**

#### **3.2 ESTAÇÃO BASE PRINCIPAL**

##### **3.2.1 INTERFACE**

##### **3.2.2 LÓGICA**

#### **3.3 ESTAÇÃO BASE SECUNDÁRIA**

##### **3.3.1 INTERFACE**

##### **3.3.2 LÓGICA**

#### **3.4 COMUNICAÇÃO ENTRE HARDWARE E SOFTWARE**

Nesta seção discutiremos como foi realizada a comunicação entre o Hardware e o Software do Dalle Pad.

#### **3.5 PROJETO MECÂNICO - INVÓLUCRO**

##### **3.5.1 PROGRAMAS UTILIZADOS PARA A EXECUÇÃO DO PROJETO**

## **4 RESULTADOS E DISCUSSÕES**

## **5 CONSIDERAÇÕES FINAIS**

### **5.1 SUGESTÕES PARA TRABALHOS FUTUROS**

## REFERÊNCIAS

- ALVES, L. **Fazendo Música no Computador**. 368: Editora Campus, 2009.
- ANDROID. **Android Reference**. 2015. Disponível em: <<http://developer.android.com/>>. Acesso em: 19 de dezembro de 2015.
- ARDUINO. **Arduino Core functions and Libraries**. 2014. Disponível em: <[http://en.wikipedia.org/wiki/persistence\\_of\\_vision](http://en.wikipedia.org/wiki/persistence_of_vision)>. Acesso em: 02 de janeiro de 2014.
- ARDUINO. **Arduino Reference**. 2014. Disponível em: <<http://arduino.cc/en/Reference/HomePage>>. Acesso em: 02 de janeiro de 2014.
- BALLOU, G. **Handbook for Sound Engineers, 5th Edition**. 368: Focal Press, 2015.
- BAYLE, J. C **Programming for Arduino**. 368: Packt Publishing, 2013.
- COLBECK, J. **MIDI Inside and Out**. 368: musicPRO guides, 2016.
- GHASSAEI, A. **Send and Receive MIDI with Arduino**. 2015. Disponível em: <<http://www.instructables.com/id/Send-and-Receive-MIDI-with-Arduino/?ALLSTEPS>>. Acesso em: 19 de dezembro de 2015.
- GIBSON, J. **Introduction to MIDI and Computer Music: The MIDI Standard**. 2013. Disponível em: <<http://www.iu.edu/emusic/361/midi.htm>>. Acesso em: 04 de janeiro de 2016.
- GREGOIRE, M. **Professional C++, 3rd Edition**. 368: John Wiley and Sons, Inc., 2014.
- GUERIN, R. **Midi Power!** 368: Course Technology PTR, 2009.
- HASS, J. **Indiana University - Introduction to Computer Music**. 2013. Disponível em: <[http://www.indiana.edu/emusic/etext/MIDI/chapter3\\_MIDI.shtml](http://www.indiana.edu/emusic/etext/MIDI/chapter3_MIDI.shtml)>. Acesso em: 19 de dezembro de 2015.
- HEWITT, M. **Music Theory for Computer Musicians**. 368: Course Technology, 2008.
- HUBER, D. M. **The MIDI Manual, 3rd Edition**. 368: Focal Press, 2012.
- JACKSON, W. **Learn Android App Development**. 368: Focal Press, 2013.
- MCGUIRE, S. **Modern MIDI**. 368: Focal Press, 2013.
- PPD. **Projetos Pedagógicos Dinâmicos - Música**. 2011. Disponível em: <<http://www.projetospedagogicosdynamics.com/musica.htm>>. Acesso em: 28 de dezembro de 2015.
- SANTINI, R. M. **Admirável Chip Novo: A música na era da internet**. Pág. 27 a 29: Rio de Janeiro: E-Papers Serviços Editoriais, 2005.
- WHEAT, D. **Arduino Internals**. 368: Apress, 2011.

WIKIPEDIA. **Oitava**. 2016. Disponível em: <<https://pt.wikipedia.org/wiki/Oitava>>. Acesso em: 04 de janeiro de 2016.



## APÊNDICE A – NOTAS MUSICAIS EM MIDI

O estudo da música em si está fora do escopo deste projeto. Entretanto, é imprescindível entender o que é uma oitava. Segue uma breve definição:

Uma oitava é o intervalo entre uma nota musical e outra com a metade ou o dobro de sua frequência.

---

(WIKIPEDIA, 2016)

Dizer que uma nota está uma oitava acima significa dizer que a nota é a mesma, porém ela está em uma região mais aguda do instrumento.

Oitava	Número referente a nota											
	C	C#	D	D#	E	F	F#	G	G#	A	A#	B
-1	0	1	2	3	4	5	6	7	8	9	10	11
0	12	13	14	15	16	17	18	19	20	21	22	23
1	24	25	26	27	28	29	30	31	32	33	34	35
2	36	37	38	39	40	41	42	43	44	45	46	47
3	48	49	50	51	52	53	54	55	56	57	58	59
4	60	61	62	63	64	65	66	67	68	69	70	71
5	72	73	74	75	76	77	78	79	80	81	82	83
6	84	85	86	87	88	89	90	91	92	93	94	95
7	96	97	98	99	100	101	102	103	104	105	106	107
8	108	109	110	111	112	113	114	115	116	117	118	119
9	120	121	122	123	124	125	126	127				

**Tabela A.1: Notas musicais em MIDI**

## APÊNDICE B – ESPECIFICAÇÃO MIDI

As seguintes tabelas representam a especificação MIDI, atualizada em 1995 pela MMA (Associação dos Fabricantes de MIDI, em inglês), que define o protocolo MIDI.

### B.1 BYTES DE ESTADO

Nas tabelas a seguir, separadas entre os possíveis estados de uma mensagem MIDI, serão citados todos os possíveis valores de *bytes* de estado e o que cada um deles representa.

Byte de estado				Byte de dados	
1° Byte Binário	1° Byte Hexa	1° Byte Decimal	Função Note OFF	2° Byte	3° Byte
10000000	80	128	Chan 1	Número da nota (0-127)	Velocidade da nota (0-127)
10000001	81	129	Chan 2	”	”
10000010	82	130	Chan 3	”	”
10000011	83	131	Chan 4	”	”
10000100	84	132	Chan 5	”	”
10000101	85	133	Chan 6	”	”
10000110	86	134	Chan 7	”	”
10000111	87	135	Chan 8	”	”
10001000	88	136	Chan 9	”	”
10001001	89	137	Chan 10	”	”
10001010	8A	138	Chan 11	”	”
10001011	8B	139	Chan 12	”	”
10001100	8C	140	Chan 13	”	”
10001101	8D	141	Chan 14	”	”
10001110	8E	142	Chan 15	”	”
continua na próxima página					

<i>continuação da página anterior</i>					
Byte de estado				Byte de dados	
1° Byte Binário	1° Byte Hexa	1° Byte Decimal	Função Note OFF	2° Byte	3° Byte
10001111	8F	143	Chan 16	”	”

Tabela B.1: Lista de Bytes de estado para a função Note Off

Byte de estado				Byte de dados	
1° Byte Binário	1° Byte Hexa	1° Byte Decimal	Função Note ON	2° Byte	3° Byte
10010000	90	144	Chan 1	Número da nota (0-127)	Velocidade da nota (0-127)
10010001	91	145	Chan 2	”	”
10010010	92	146	Chan 3	”	”
10010011	93	147	Chan 4	”	”
10010100	94	148	Chan 5	”	”
10010101	95	149	Chan 6	”	”
10010110	96	150	Chan 7	”	”
10010111	97	151	Chan 8	”	”
10011000	98	152	Chan 9	”	”
10011001	99	153	Chan 10	”	”
10011010	9A	154	Chan 11	”	”
10011011	9B	155	Chan 12	”	”
10011100	9C	156	Chan 13	”	”
10011101	9D	157	Chan 14	”	”
10011110	9E	158	Chan 15	”	”
10011111	9F	159	Chan 16	”	”

Tabela B.2: Lista de Bytes de estado para a função Note On

Byte de estado				Byte de dados	
1° Byte Binário	1° Byte Hexa	1° Byte Decimal	Função Polyphonic Aftertouch	2° Byte	3° Byte
10100000	A0	160	Chan 1	Número da nota (0-127)	valor do aftertouch (0-127)
10100001	A1	161	Chan 2	”	”
10100010	A2	162	Chan 3	”	”
10100011	A3	163	Chan 4	”	”
<i>continua na próxima página</i>					

<i>continuação da página anterior</i>					
Byte de estado				Byte de dados	
1° Byte Binário	1° Byte Hexa	1° Byte Decimal	Função Polyphonic Aftertouch	2° Byte	3° Byte
10100100	A4	164	Chan 5	”	”
10100101	A5	165	Chan 6	”	”
10100110	A6	166	Chan 7	”	”
10100111	A7	167	Chan 8	”	”
10101000	A8	168	Chan 9	”	”
10101001	A9	169	Chan 10	”	”
10101010	AA	170	Chan 11	”	”
10101011	AB	171	Chan 12	”	”
10101100	AC	172	Chan 13	”	”
10101101	AD	173	Chan 14	”	”
10101110	AE	174	Chan 15	”	”
10101111	AF	175	Chan 16	”	”

**Tabela B.3: Lista de Bytes de estado para a função Polyphonic Aftertouch**

Byte de estado				Byte de dados	
1° Byte Binário	1° Byte Hexa	1° Byte Decimal	Função de controle Modo de Canal	2° Byte	3° Byte
10110000	B0	176	Chan 1	Veja tabela B.9 (0-127)	Veja B.9 (0-127)
10110001	B1	177	Chan 2	”	”
10110010	B2	178	Chan 3	”	”
10110011	B3	179	Chan 4	”	”
10110100	B4	180	Chan 5	”	”
10110101	B5	181	Chan 6	”	”
10110110	B6	182	Chan 7	”	”
10110111	B7	183	Chan 8	”	”
10111000	B8	184	Chan 9	”	”
10111001	B9	185	Chan 10	”	”
10111010	BA	186	Chan 11	”	”
10111011	BB	187	Chan 12	”	”
10111100	BC	188	Chan 13	”	”
<i>continua na próxima página</i>					

<i>continuação da página anterior</i>					
Byte de estado				Byte de dados	
1º Byte Binário	1º Byte Hexa	1º Byte Decimal	Função de controle Modo de Canal	2º Byte	3º Byte
10111101	BD	189	Chan 14	”	”
10111110	BE	190	Chan 15	”	”
10111111	BF	191	Chan 16	”	”

**Tabela B.4: Lista de Bytes de estado para mudança de controle / modo de canal**

Byte de estado				Byte de dados	
1º Byte Binário	1º Byte Hexa	1º Byte Decimal	Função Mudança de Programa	2º Byte	3º Byte
11000000	C0	192	Chan 1	Programa (0-127)	Nulo
11000001	C1	193	Chan 2	”	”
11000010	C2	194	Chan 3	”	”
11000011	C3	195	Chan 4	”	”
11000100	C4	196	Chan 5	”	”
11000101	C5	197	Chan 6	”	”
11000110	C6	198	Chan 7	”	”
11000111	C7	199	Chan 8	”	”
11001000	C8	200	Chan 9	”	”
11001001	C9	201	Chan 10	”	”
11001010	CA	202	Chan 11	”	”
11001011	CB	203	Chan 12	”	”
11001100	CC	204	Chan 13	”	”
11001101	CD	205	Chan 14	”	”
11001110	CE	206	Chan 15	”	”
11001111	CF	207	Chan 16	”	”

**Tabela B.5: Lista de Bytes de estado para a função mudança de programa**

Byte de estado				Byte de dados	
1º Byte Binário	1º Byte Hexa	1º Byte Decimal	Função Channel Aftertouch	2º Byte	3º Byte
11010000	D0	208	Chan 1	Valor do aftertouch (0-127)	Nulo
11010000	D1	209	Chan 2	”	”
<i>continua na próxima página</i>					

<i>continuação da página anterior</i>					
Byte de estado				Byte de dados	
1° Byte Binário	1° Byte Hexa	1° Byte Decimal	Função Channel Aftertouch	2° Byte	3° Byte
11010000	D2	210	Chan 3	”	”
11010000	D3	211	Chan 4	”	”
11010000	D4	212	Chan 5	”	”
11010000	D5	213	Chan 6	”	”
11010000	D6	214	Chan 7	”	”
11010000	D7	215	Chan 8	”	”
11010000	D8	216	Chan 9	”	”
11010000	D9	217	Chan 10	”	”
11010000	DA	218	Chan 11	”	”
11010000	DB	219	Chan 12	”	”
11010000	DC	220	Chan 13	”	”
11010000	DD	221	Chan 14	”	”
11010000	DE	222	Chan 15	”	”
11010000	DF	223	Chan 16	”	”

**Tabela B.6: Lista de Bytes de estado para a função Channel Aftertouch**

Byte de estado				Byte de dados	
1° Byte Binário	1° Byte Hexa	1° Byte Decimal	Função Pitch Wheel	2° Byte	3° Byte
11100000	E0	224	Chan 1	LSB Pitch Wheel (0-127)	MSB Pitch Wheel (0-127)
11100001	E1	225	Chan 2	”	”
11100010	E2	226	Chan 3	”	”
11100011	E3	227	Chan 4	”	”
11100100	E4	228	Chan 5	”	”
11100101	E5	229	Chan 6	”	”
11100110	E6	230	Chan 7	”	”
11100111	E7	231	Chan 8	”	”
11101000	E8	232	Chan 9	”	”
11101001	E9	233	Chan 10	”	”
11101010	EA	234	Chan 11	”	”
11101011	EB	235	Chan 12	”	”
<i>continua na próxima página</i>					

continuação da página anterior					
Byte de estado				Byte de dados	
1° Byte Binário	1° Byte Hexa	1° Byte Decimal	Função Pitch Wheel	2° Byte	3° Byte
11101100	EC	236	Chan 13	”	”
11101101	ED	237	Chan 14	”	”
11101110	EE	239	Chan 15	”	”
11101111	EF	239	Chan 16	”	”

**Tabela B.7: Lista de Bytes de estado para a função de controle do Pitch Wheel**

Byte de estado				Byte de dados	
1° Byte Binário	1° Byte Hexa	1° Byte Decimal	Função Mensagem	2° Byte	3° Byte
11110000	F0	240	Exclusivo do sistema	Mensagem SysEx Atual	Mensagem SysEx Atual
11110001	F1	241	Código de tempo MIDI	Mensagem tempo Atual	Mensagem tempo Atual
11110010	F2	242	Ponteiro de Posição da música	LSB	MSB
11110011	F3	243	Seleção da música (música #)	Valores (0-127)	Nulo
11110100	F4	244	Indefinido	N/A	N/A
11110101	F5	245	Indefinido	N/A	N/A
11110110	F6	246	Pedido de Afinação	Nulo	Nulo
11110111	F7	247	Final da SysEx (EOX)	”	”
11111000	F8	248	Cronômetro	”	”
11111001	F9	249	Indefinido	”	”
11111010	FA	250	Iniciar	”	”
11111011	FB	251	Continuar	”	”
11111100	FC	252	Pausar	”	”
11111101	FD	253	Indefinido	”	”
11111110	FE	254	Sensoriamento ativo	”	”
11111111	FF	255	Reset do Sistema	”	”

**Tabela B.8: Lista de Bytes de estado para as funções de sistema**

## B.2 BYTES DE DADOS

A seguinte tabela é uma referência de todas os possíveis mensagens de controle e modos de canais, além de citar o que cada uma delas representa.

2° Byte Binário	2° Byte Hexa	2° Byte Decimal	Função	3° Byte Valor	3° Byte Uso
00000000	00	0	Chan 1	0-127	LSB
00000001	01	1	Chan 2	0-127	LSB
00000010	02	2	Chan 3	0-127	LSB
00000011	03	3	Chan 4	0-127	LSB
00000100	04	4	Chan 5	0-127	LSB
00000101	05	5	Chan 6	0-127	LSB
00000110	06	6	Chan 7	0-127	LSB
00000111	07	7	Chan 8	0-127	LSB
00001000	08	8	Chan 9	0-127	LSB
00001001	09	9	Chan 10	0-127	LSB
00001010	0A	10	Chan 11	0-127	LSB
00001011	0B	11	Chan 12	0-127	LSB
00001100	0C	12	Chan 13	0-127	LSB
00001101	0D	13	Chan 14	0-127	LSB
00001110	0E	14	Chan 15	0-127	LSB
00001111	0F	15	Chan 16	0-127	LSB
00010000	10	16	Chan 1	0-127	LSB
00010001	11	17	Chan 2	0-127	LSB
00010010	12	18	Chan 3	0-127	LSB
00010011	13	19	Chan 4	0-127	LSB
00010100	14	20	Chan 5	0-127	LSB
00010101	15	21	Chan 6	0-127	LSB
00010110	16	22	Chan 7	0-127	LSB
00010111	17	23	Chan 8	0-127	LSB
00011000	18	24	Chan 9	0-127	LSB
00011001	19	25	Chan 10	0-127	LSB
00011010	1A	26	Chan 11	0-127	LSB
<i>continua na próxima página</i>					



<i>continuação da página anterior</i>					
<b>2° Byte Binário</b>	<b>2° Byte Hexa</b>	<b>2° Byte Decimal</b>	<b>Função</b>	<b>3° Byte Valor</b>	<b>3° Byte Uso</b>
00011011	1B	27	Chan 12	0-127	LSB
00011100	1C	28	Chan 13	0-127	LSB
00011101	1D	29	Chan 14	0-127	LSB
00011110	1E	30	Chan 15	0-127	LSB
00011111	1F	31	Chan 16	0-127	LSB
00100000	20	32	Chan 1	0-127	LSB
00100001	21	33	Chan 2	0-127	LSB
00100010	22	34	Chan 3	0-127	LSB
00100011	23	35	Chan 4	0-127	LSB
00100100	24	36	Chan 5	0-127	LSB
00100101	25	37	Chan 6	0-127	LSB
00100110	26	38	Chan 7	0-127	LSB
00100111	27	39	Chan 8	0-127	LSB
00101000	28	40	Chan 9	0-127	LSB
00101001	29	41	Chan 10	0-127	LSB
00101010	2A	42	Chan 11	0-127	LSB
00101011	2B	43	Chan 12	0-127	LSB
00101100	2C	44	Chan 13	0-127	LSB
00101101	2D	45	Chan 14	0-127	LSB
00101110	2E	46	Chan 15	0-127	LSB
00101111	2F	47	Chan 16	0-127	LSB
00110000	30	48	Chan 1	0-127	LSB
00110001	31	49	Chan 2	0-127	LSB
00110010	32	50	Chan 3	0-127	LSB
00110011	33	51	Chan 4	0-127	LSB
00110100	34	52	Chan 5	0-127	LSB
00110101	35	53	Chan 6	0-127	LSB
00110110	36	54	Chan 7	0-127	LSB
00110111	37	55	Chan 8	0-127	LSB
00111000	38	56	Chan 9	0-127	LSB
00111001	39	57	Chan 10	0-127	LSB
<i>continua na próxima página</i>					

<i>continuação da página anterior</i>					
<b>2° Byte Binário</b>	<b>2° Byte Hexa</b>	<b>2° Byte Decimal</b>	<b>Função</b>	<b>3° Byte Valor</b>	<b>3° Byte Uso</b>
00111010	3A	58	Chan 11	0-127	LSB
00111011	3B	59	Chan 12	0-127	LSB
00111100	3C	60	Chan 13	0-127	LSB
00111101	3D	61	Chan 14	0-127	LSB
00111110	3E	62	Chan 15	0-127	LSB
00111111	3F	63	Chan 16	0-127	LSB
01000000	40	64	Chan 1	0-127	LSB
01000001	41	65	Chan 2	< 63 = Off	≥ 64 = On
01000010	42	66	Chan 3	< 63 = Off	≥ 64 = On
01000011	43	67	Chan 4	< 63 = Off	≥ 64 = On
01000100	44	68	Chan 5	< 63 = Off	≥ 64 = On
01000101	45	69	Chan 6	< 63 = Off	≥ 64 = On
01000110	46	70	Chan 7	0-127	LSB
01000111	47	71	Chan 8	0-127	LSB
01001000	48	72	Chan 9	0-127	LSB
01001001	49	73	Chan 10	0-127	LSB
01001010	4A	74	Chan 11	0-127	LSB
01001011	4B	75	Chan 12	0-127	LSB
01001100	4C	76	Chan 13	0-127	LSB
01001101	4D	77	Chan 14	0-127	LSB
01001110	4E	78	Chan 15	0-127	LSB
01001111	4F	79	Chan 16	0-127	LSB
01010000	50	80	Chan 1	0-127	LSB
01010001	51	81	Chan 2	0-127	LSB
01010010	52	82	Chan 3	0-127	LSB
01010011	53	83	Chan 4	0-127	LSB
01010100	54	84	Chan 5	0-127	Nota de origem
01010101	55	85	Chan 6	0-127	LSB
01010110	56	86	Chan 7	0-127	LSB
01010111	57	87	Chan 8	0-127	LSB
01011000	58	88	Chan 9	0-127	LSB
<i>continua na próxima página</i>					

<i>continuação da página anterior</i>					
<b>2° Byte Binário</b>	<b>2° Byte Hexa</b>	<b>2° Byte Decimal</b>	<b>Função</b>	<b>3° Byte Valor</b>	<b>3° Byte Uso</b>
01011001	59	89	Chan 10	0-127	LSB
01011010	5A	90	Chan 11	0-127	LSB
01011011	5B	91	Chan 12	0-127	LSB
01011100	5C	92	Chan 13	0-127	LSB
01011101	5D	93	Chan 14	0-127	LSB
01011110	5E	94	Chan 15	0-127	LSB
01011111	5F	95	Chan 16	0-127	LSB
01100000	60	96	Chan 1	N/A	N/A
01100001	61	97	Chan 2	N/A	N/A
01100010	62	98	Chan 3	0-127	LSB
01100011	63	99	Chan 4	0-127	MSB
01100100	64	100	Chan 5	0-127	LSB
01100101	65	101	Chan 6	0-127	MSB
01100110	66	102	Chan 7	N/A	N/A
01100111	67	103	Chan 8	N/A	N/A
01101000	68	104	Chan 9	N/A	N/A
01101001	69	105	Chan 10	N/A	N/A
01101010	6A	106	Chan 11	N/A	N/A
01101011	6B	107	Chan 12	N/A	N/A
01101100	6C	108	Chan 13	N/A	N/A
01101101	6D	109	Chan 14	N/A	N/A
01101110	6E	110	Chan 15	N/A	N/A
01101111	6F	111	Chan 16	N/A	N/A
01110000	70	112	Chan 1	N/A	N/A
01110001	71	113	Chan 2	N/A	N/A
01110010	72	114	Chan 3	N/A	N/A
01110011	73	115	Chan 4	N/A	N/A
01110100	74	116	Chan 5	N/A	N/A
01110101	75	117	Chan 6	N/A	N/A
01110110	76	118	Chan 7	N/A	N/A
01110111	77	119	Chan 8	N/A	N/A
<i>continua na próxima página</i>					

<i>continuação da página anterior</i>					
<b>2° Byte Binário</b>	<b>2° Byte Hexa</b>	<b>2° Byte Decimal</b>	<b>Função</b>	<b>3° Byte Valor</b>	<b>3° Byte Uso</b>
01111000	78	120	Chan 9	0	0
01111001	79	121	Chan 10	0	0
01111010	7A	122	Chan 11	0 = Off	127 = On
01111011	7B	123	Chan 12	0	0
01111100	7C	124	Chan 13	0	0
01111101	7D	125	Chan 14	0	0
01111110	7E	126	Chan 15	N/A	N/A
01111111	7F	127	Chan 16	0	0

**Tabela B.9: Lista de mensagens de mudança de controle / modo de canal**

## **ANEXO A – DATASHEETS**

Este capítulo compreende todos os *datasheets* utilizados durante o desenvolver do projeto.

É importante ressaltar que os componentes desenvolvidos pela própria equipe, sistemas eletrônicos e códigos estão todos relatados no capítulo anterior.

## A.1 POTENCIÔMETRO ROTATIVO



### Features

- Low torque
- Carbon element
- Plain or knurled shaft option
- Metal bushing
- Metal shaft
- Rear solder lugs
- Audio or linear taper options
- Variety of resistance values
- RoHS compliant\*



## PDB241-GTR Series - 24 mm Guitar Potentiometer

### Electrical Characteristics

Taper.....Audio, linear  
Standard Resistance .....10K to 1M ohms  
Standard Resistance Tolerance..... $\pm 20\%$

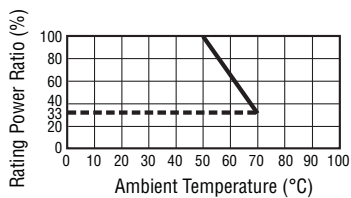
### Environmental Characteristics

Operating Temperature ..... $-10^{\circ}\text{C}$  to  $+70^{\circ}\text{C}$   
Power Rating .....0.25 watt  
Maximum Operating Voltage .....250 V  
Rotational Noise .....150 mV max.

### Mechanical Characteristics

Mechanical Angle ..... $300^{\circ} \pm 5^{\circ}$   
Rotational Torque .....10 to 45 g-cm  
Stop Strength.....8 kg-cm min.  
Rotational Life.....15,000 cycles min.  
Soldering Condition  
Manual Soldering ..... $300^{\circ}\text{C}$  within 3 seconds  
Wave Soldering ..... $260^{\circ}\text{C}$  within 3 seconds  
Hardware .....Two flat washers and two mounting nuts supplied per potentiometer

### Derating Curve

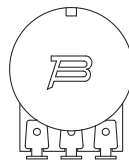
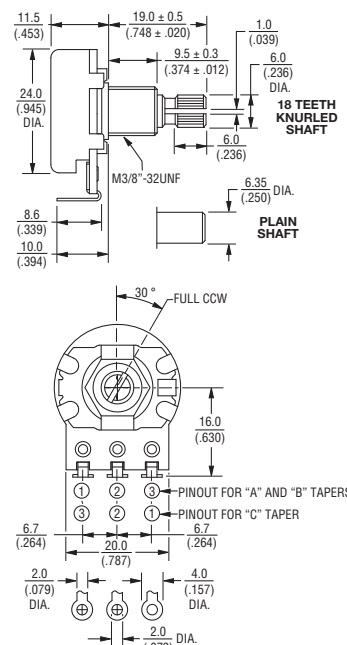


### Standard Resistance Table

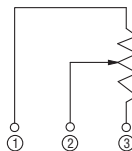
Resistance (Ohms)	Resistance Code
10,000	103
25,000	253
50,000	503
100,000	104
250,000	254
300,000	304
500,000	504
1,000,000	105

### Product Dimensions

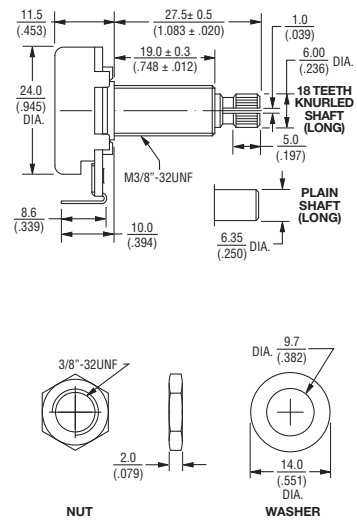
#### PDB241-GTR01



### Circuit



#### PDB241-GTR03



TOLERANCES:  
UNDER 10.0 =  $\pm 0.3$  10.0 - 100 =  $\pm 0.5$   
(0.394) (0.012) (0.394 - 3.937) (0.020)

DIMENSIONS: MM (INCHES)

### How To Order

#### PDB24 1 - GTR 01 - 504 A2

Model \_\_\_\_\_  
Number of Units \_\_\_\_\_  
1 = Single  
Guitar Pot Designator \_\_\_\_\_  
Configuration \_\_\_\_\_  
1 = Knurled Shaft / Solder Lugs  
2 = Plain Shaft / Solder Lugs  
3 = Knurled Shaft (Long) / Solder Lugs  
4 = Plain Shaft (Long) / Solder Lugs  
Resistance Code (See Table) \_\_\_\_\_  
Resistance Taper (See Taper Charts) \_\_\_\_\_

\*RoHS Directive 2002/95/EC Jan. 27, 2003 including annex and RoHS Recast 2011/65/EU June 8, 2011.

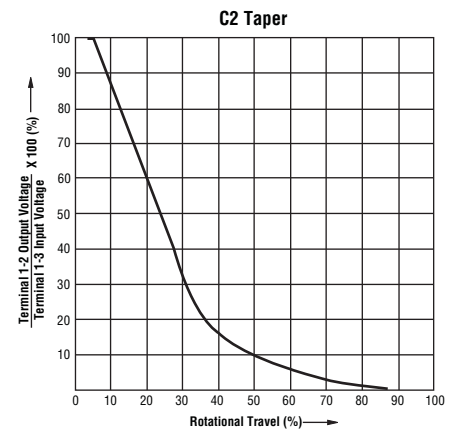
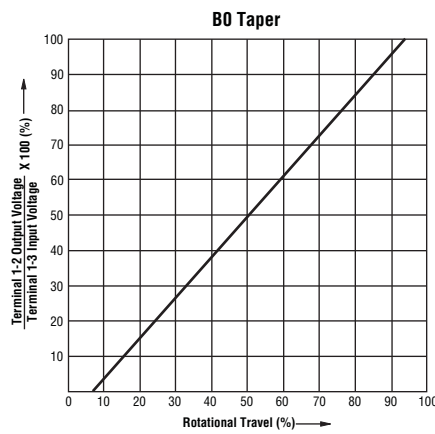
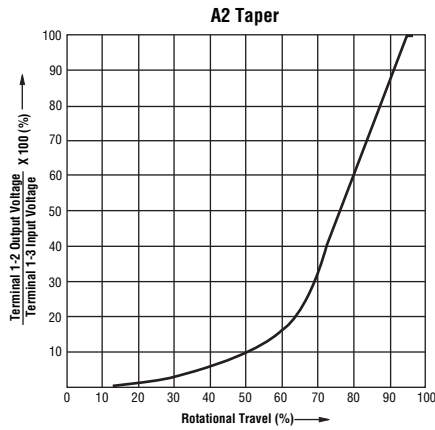
Specifications are subject to change without notice.

The device characteristics and parameters in this data sheet can and do vary in different applications and actual device performance may vary over time. Users should verify actual device performance in their specific applications.

# PDB241-GTR Series - 24 mm Guitar Potentiometer

**BOURNS®**

## Taper Charts



Other tapers available.

REV. 11/13

Specifications are subject to change without notice.  
The device characteristics and parameters in this data sheet can and do vary in different applications and actual device performance may vary over time.  
Users should verify actual device performance in their specific applications.

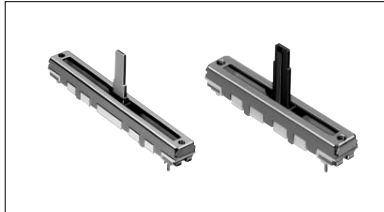
## A.2 POTENCIÔMETRO LINEAR

### Slide Potentiometer Standard Type ( Super Slide™ ) RS 1 Series



Solderable with auto dipping and available in a wide variety.

Rotary  
Potentiometers  
Slide  
Potentiometers



#### Typical Specifications

Items	Specifications
Total resistance tolerance	± 20%
Maximum operating voltage	50V AC, 10V DC
Operating force	0.3 to 2.5N
Operating life	15,000cycles
Operating temperature range	- 25 to + 70

#### Product Line

##### Insulated lever

General-use

Mixer

Number of resistor elements	Travel (mm)	Lever types	Length of lever (mm)	Total resistance (k )	Resistance taper	Detent	Mounting plate	Minimum packing unit( pcs. )	Products No.	Drawing No.	
Single-unit	30	J-1	5	10	15A	Without	Without	1,000	RS301111J00P	1	
							With		RS301111AJ01R	2	
			50	1B	Without		RS301111J00R		1		
				15A	With		RS301111J00W		2		
		10	A	1B	Without		RS301111AJ01K		3		
			C		Without		RS301111A01G		4		
			45		B		Without		RS301111C01L	5	
							With		RS301111AC00V	6	
	Dual-unit	30	C							RS451111ABA02	7
			J-1	5	3B					RS30112AC00J	8
							RS30112AJA02				

##### Metal lever

Number of resistor elements	Travel (mm)	Lever types	Length of lever (mm)	Total resistance (k )	Resistance taper	Detent	Mounting plate	Minimum packing unit( pcs. )	Products No.	Drawing No.
Single-unit	15	9-1	10	10	15A	Without	With	1,000	RS15111A900B	9
	30		15	20	1B		Without		RS30111A9012	10
		4					6		With	RS3011114A02
	6	9-1	10	10	1B				Without	RS30111A602N
	6						With		RS45111A900F	13
	45	6	15	10	15A		With		RS4511119A04	14
	9-1								10	15
	6	15	10	20	15A		With			
	6								15	10
	60	4	20	15	20		15A			
45	6								15	20
60		6	15	20	20		15A			

#### Note

Products other than those listed in above products are also available. Please contact us for details.

Refer to **P.408** for product varieties.  
Refer to **P.408** for configuration details.



**Slide Potentiometer**  
**Standard Type ( Super Slide™ ) RS 1 Series**

## ■ Dimensions

Unit:mm

No.	Style	PC board mounting hole dimensions
17		
18		
19		
20		

### Rotary Potentiometers

## Slide Potentiometers

General-use






Mixer

The length of the operating part ( $L_1$ ) can be customized. Refer to **P.408** for details.

## ALPS

## Slide Potentiometers

### List of Varieties

Type		Super Slide™		Slim Slide™ (Slim4)				Slim Slide™ (Low-Profile)	
		RS	11	RS		H1		RS	M11
photo									
Travel( mm )		15( RS15111 ) 20( RS20111 ) 30( RS30111 ) 45( RS45111 ) 60( RS6011Y )	20( RS20112 ) 30( RS30112 ) 45( RS45112 ) 60( RS60112 )	15( RS15H111 ) 15( RS15H113 ) 20( RS20H111 ) 20( RS20H113 ) 30( RS30H111 )	15( RS15H121 ) 20( RS20H123 ) 30( RS30H121 )	15( RS15H11A ) 20( RS20H11A ) 20( RS20H11C ) 30( RS30H11A )	15( RS15H12A ) 20( RS20H12A ) 30( RS30H12A )	15( RS15M111 ) 20( RS20M111 ) 20( RS20M113 )	15( RS15M11A ) 20( RS20M11A )
Number of resistor elements		Single-unit	Dual-unit	Single-unit	Dual-unit	Single-unit	Dual-unit	Single-unit	
Direction of lever		Vertical				Horizontal		Vertical	Horizontal
Electrical Performance	Total resistance( k )	10, 20, 50, 100, 200		5, 10, 20, 50, 100, 200, 250					
	Resistance taper	15A, 1B, 4B, 10A, 3B, 15C		15A, 1B, 3B					
Mechanical Performance	Operating force	0.3 to 2.5N		0.2 to 1.1N				0.15 to 0.8N	
	Stopper strength	50N		30N					
	Lever push-pull strength			30N				20N	
	Lever wobble( mm ) Both sides	$\frac{2( 2 \times L )}{20}$		1.6 max.					
	Detent slip-out force	Operating force×( 0.2 to 2N )		Operating force×( 0.05 to 0.8N )				With detent feeling	
	Lever deviation( mm ) One side	0.5 max.							
Solder heat resistance	Manual soldering	350 max. 3s max.							
	Dip soldering	260 max. 5s max.							
Environmental Performance	Cold	- 30 ± 2 for 96h							
	Dry heat	70 ± 2 for 96h						70 ± 2 for 240h	
	Damp heat	40 ± 2 , 90 to 95%RH for 96h							
Page		402		410				416	

Rotary Potentiometers  
Slide Potentiometers

General-use  
Mixer

Slide Potentiometers Cautions .....420, 421

#### Notes

1. "L" in the "Lever Wobble" column of the above table indicates the length of lever.
2. Products other than those listed in the above list are also available. Please contact us for details.
3. The resistance taper specification in ( ) are individually defined.
4. 「RS」 indicates travel.