

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE CIÊNCIA DA COMPUTAÇÃO

HUGO CRISTIAN MEZARINA BELTRÁN

**Aplicação de recursos computacionais
móveis como dispositivos de interface
humana USB**

Trabalho de Graduação.

Prof. Dr. Sérgio Cechin

Porto Alegre, janeiro de 2013.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. Carlos Alexandre Netto

Vice-Reitor: Prof. Rui Vicente Oppermann

Pró-Reitora de Graduação: Prof. Sérgio Franco

Diretor do Instituto de Informática: Prof. Luís da Cunha Lamb

Coordenador do CIC: Prof. Raúl Weber

Bibliotecária-Chefe do Instituto de Informática: Beatriz Regina Bastos Haro

AGRADECIMENTOS

A minha família por seu carinho e ajuda ao longo da minha vida.

À UFRGS por ser minha alma mater, da qual me orgulho muito.

A Porto Alegre e ao Brasil por ter sido meu lar por todos estes anos e ter me oferecido novas experiências, conhecimentos, amizades e boas lembranças.

A Maria Luisa por ter ser minha inspiração e ter-me ajudado com algumas das ilustrações do presente trabalho.

SUMÁRIO

LISTA DE ABREVIATURAS E SIGLAS.....	6
LISTA DE FIGURAS	7
LISTA DE TABELAS	8
RESUMO	9
ABSTRACT.....	10
1. INTRODUÇÃO	11
1.1 Motivação.....	11
1.2 Objetivo.....	13
2. ESTADO DA ARTE.....	15
2.1 Interação Homem-Computador.....	15
2.1.1 Tecnologias de IHC.....	15
2.2 Recursos computacionais móveis para usuários finais.....	16
2.2.1 A plataforma Android.....	16
2.2.2 As capacidades dos sensores e periféricos disponíveis na plataforma Android.....	16
2.3 Universal Serial Bus	18
2.3.1 História	18
2.3.2 O conector e o cabo USB.....	19
2.3.3 Topologia.....	20
2.3.3.1 Mestre.....	21
2.3.3.2 Periférico.....	21
2.4 Bluetooth	22
2.4.1 Princípios de funcionamento.....	22
2.4.2 Interface do Controlador Mestre.....	22
2.4.3 Interação entre a pilha Bluetooth e um aplicativo Android	23
2.5 Microcontroladores.....	24
2.5.1 Novas oportunidades.....	24
2.6 Alternativas oferecidas no mercado.....	24
3. DESENVOLVIMENTO DO PROJETO.....	27
3.1 Visão geral.....	27
3.1.1 A ideia original	27
3.1.2 Os primeiros passos no percurso da pesquisa	27
3.1.3 A segunda tentativa.....	28
3.2 Estudo dos elementos envolvidos	29
3.2.1 Etapas da concepção ergonômica da interface.....	29
3.2.1.1 Análise do potencial usuário	29
3.2.1.2 Tarefas do aplicativo.....	29
3.2.1.3 Design do aplicativo.....	30
3.2.2 Dispositivo Android e conexão Bluetooth.....	30
3.2.3 Microcontrolador USB	30
3.2.4 Microcontrolador Bluetooth.....	30
3.3 Frameworks e Interfaces de comunicação	30
3.3.1 IOIOLib	30
3.3.2 Teensyduino.....	30
3.3.3 UART	30
3.4 Organização do projeto.....	31

3.5	Implementação dos blocos do projeto	32
3.5.1	Aplicativo Android	32
3.5.2	Interface Bluetooth / UART.....	33
3.5.3	Software Embarcado.....	34
3.5.4	Montagem física dos elementos de hardware.....	35
4.	TESTES	36
4.1	Conexão USB	36
4.2	Emparelhamento Bluetooth.....	38
4.3	Envio de dados UART.....	40
4.4	Produção de entradas e consumo de quadros USB HID sobre Bluetooth.....	40
4.5	Desconexão.....	41
5.	CONCLUSÃO	42
	REFERÊNCIAS	43

LISTA DE ABREVIATURAS E SIGLAS

HID	Human Interface Device
USB	Universal Serial Bus
LUFA	Lightweight USB Framework for AVRs
UART	Universal Asynchronous Receiver-Transmitter
IHC	Interação Homem-Computador
GPIO	General Purpose Input/output
SoC	System on Chip
FIFO	First in, First out
HCI	Host Controller Interface

LISTA DE FIGURAS

Figura 1.1 : Processo da convergência digital.....	14
Figura 1.2 : Convergência digital acontecida no microcomputador.....	12
Figura 1.3 : Convergência digital acontecida nos smartphones e tablets	12
Figura 1.4 : Representação do projeto.....	14
Figura 2.1 : Alguns dos conectores e receptáculos USB.....	20
Figura 2.2 : Topologia de estrela USB.....	21
Figura 2.3 : Funcionamento da pilha Bluetooth.....	23
Figura 2.4 : Windroid.....	26
Figura 3.1 : Organização do projeto.....	32
Figura 3.2 : Sistema de mensagens.....	33
Figura 3.3 : Esquemático do projeto.....	34
Figura 3.4 : Transformação de mensagens UART em quadros USB HID.....	35
Figura 3.6 : Montagem do projeto na protoboard.....	35
Figura 4.1 : Elementos utilizadas para o teste.....	36
Figura 4.2 : Conexão da placa Teensy++ 2.0.....	37
Figura 4.3 : Conexão à entrada USB do microcomputador.....	37
Figura 4.4 : Conectado à fonte e o adaptador Bluetooth V2 na placa IOIO.....	38
Figura 4.5 : Ativando o Bluetooth.....	39
Figura 4.6 : Emparelhamento Bluetooth.....	39
Figura 4.7 : Led acesso quando passam dados pelo UART.....	40
Figura 4.8 : Inicializando aplicativo na tablet.....	40
Figura 4.9 : Produção de quadros na tablet e consumo deles no microcomputador.....	41

LISTA DE TABELAS

Tabela 2.1: Linha de tempo da evolução USB	19
Tabela 2.2 Host Controller Interface	23

RESUMO

Este trabalho tem como objetivo realizar um estudo e implementação de uma interface homem-computador utilizando os recursos computacionais móveis, como os sensores dos telefones celulares e tablets, como dispositivos de interface humana USB. É apresentado um estudo das bases teóricas que sustentam ditas tecnologias e como podem ser compostas para obter uma nova classe de periférico de entrada que aproveita as características dos dispositivos móveis e os usa para conectá-los a um microcomputador como dispositivos USB nativos, via a conexão Bluetooth.

Palavras-Chave: USB, Interface Homem-Computador, Bluetooth, Dispositivos Móveis.

ABSTRACT

This document has the objective to study and implement classic human interfaces (keyboard and mouse) using mobile computer resources, such as mobile phones and tablets, as USB Human Interface Devices. It is presented a study of their theoretical basis and how they could be composed to obtain a new class of input peripherals that take advantages of the features of those mobile devices and use them to connect them to a personal computer as native USB devices, via a Bluetooth connection.

Keywords: USB, Human Interface Device, Bluetooth, Mobile Devices.

1. INTRODUÇÃO

O desenvolvimento e maturidade comercial de muitos dispositivos eletrônicos oferecem muitas vantagens. Todos estes dispositivos, como smartphones e tablets, contém vários sensores e periféricos de entrada/saída embutidos e aumentam as funcionalidades oferecidas aos usuários finais.

Mas mesmo com todos estes avanços, o microcomputador continua sendo a plataforma mais utilizada no momento de fazer trabalho produtivo sério. Isto é devido ao fato que a dupla teclado/mouse se adapta melhor às mais diversas tarefas e são quase universalmente aceitas pelos usuários em geral.

Este trabalho visa aproveitar os recursos embutidos nos smartphones e tablets para funcionarem junto com o microcomputador, como se fossem recursos nativos deste. O estudo de caso faz uma adaptação da tela sensível ao toque de um tablet, para utilizá-lo como um teclado nativo USB, conectados sem fios via Bluetooth.

1.1 Motivação

O grande avanço e massificação de muitos dispositivos eletrônicos no nosso dia-a-dia nos levam cada vez mais para um estado de convergência digital (JEON, 2010). Este fenômeno é o resultado da interação muito próxima entre as tecnologias da informação, as telecomunicações, os eletrônicos de consumo massivo, e a indústria do entretenimento.

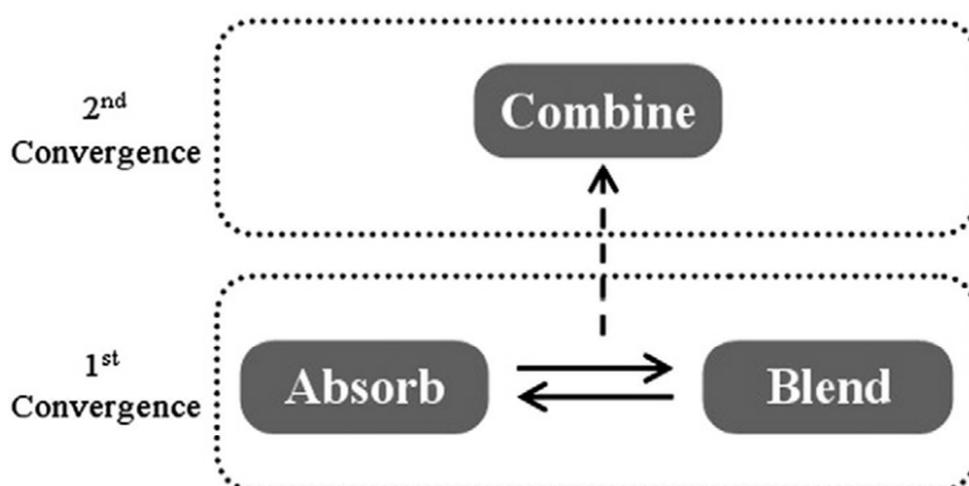


Figura 1.1 Processo da convergência digital como processo criativo na indústria de TI (JEON, 2010).

Por exemplo, há alguns anos a comunicação à distância entre pessoas era principalmente pelo telefone ou aplicativos VOIP no microcomputador. A criação de conteúdo, como a captura de imagens, era feita com uma câmera digital, a música era ouvida em um aparelho específico para esse fim e assim por diante. Mesmo que todos esses dispositivos eletrônicos fossem digitais, eles precisavam de um microcomputador para poder extrair e utilizar ditos conteúdos.

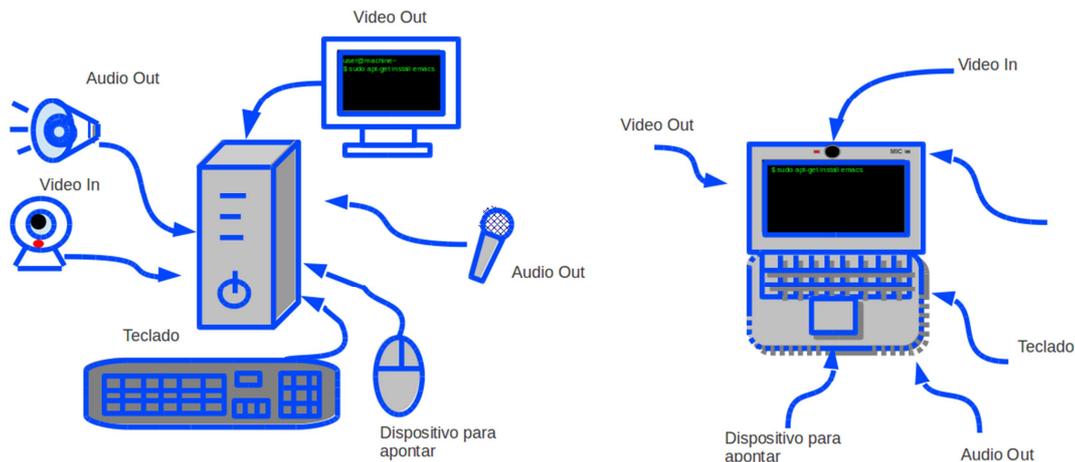


Figura 1.2: Convergência digital acontecida no microcomputador.

Atualmente estes e outros dispositivos foram absorvidos pelos telefones celulares, smartphone ou tablets devido a este processo de convergência. Este processo de convergência digital fez com que mais pessoas pudessem acessar às vantagens e benefícios de uma grande quantidade de periféricos de entrada e saída, diminuindo a complexidade e a dependência a um microcomputador que centralize todos esses recursos.



Figura 1.3: Convergência digital acontecida nos smartphones e tablets.

Mas mesmo assim, considerando o exposto acima, nenhum smartphone ou tablet atualmente pode substituir totalmente a um microcomputador. A grande quantidade de software e hardware disponível para esta plataforma faz com que seja a ferramenta de trabalho profissional mais importante. Para trabalhos profissionais, comerciais ou que de geração de renda, o microcomputador ainda é o mais indicado devido às suas funcionalidades. Os smartphones ou tablets, já há alguns anos no mercado, auxiliam em muitas das tarefas do dia a dia, mas ainda não suprem todas as necessidades da grande gama de profissionais existente. Por exemplo, um programador não pode escrever o código que quiser num tablet. Assim também um engenheiro não pode utilizar uma

ferramenta completa de CAD para fazer todos seus projetos, tornando ainda o uso do computador pessoal indispensável.

O processo de convergência foi tão importante nos smartphones e tablets que eles atualmente fazem muitas coisas que os usuários finais desejam (acessar a internet, compartilhar fotos, jogar vídeo games, etc.), mas estes dispositivos móveis não interagem diretamente com outros dispositivos da sua mesma classe ou com outros dispositivos de classe diferente, como no caso de um microcomputador. Se algum tipo de interação acontece entre estes dispositivos, geralmente é com a intermediação da internet ou de um ponto de acesso à internet e em alguns casos minoritários via Bluetooth ou uma tecnologia semelhante como o Near Field Communication (NFC).

Este trabalho oferece uma alternativa ao duplo processo de convergência digital, aproveitando os diversos sensores e dispositivos de entrada que já estão incorporados aos smartphones e tablets e aproveitá-los fora deles como dispositivos de entrada/saída em um computador.

1.2 Objetivo

Ao longo deste trabalho serão detalhadas as técnicas de software e os meios de hardware para desenvolver um dispositivo, que ao ser conectado a uma porta USB de um computador qualquer, seja reconhecido como um dispositivo USB nativo. Este dispositivo irá funcionar sem a necessidade de instalar drivers específicos, e assim pode trabalhar virtualmente com qualquer sistema operacional. No caso do estudo deste trabalho, um tablet ou smartphone irá receber as entradas do usuário final em um teclado exibidos na tela sensível ao toque. Essas entradas irão ser repassadas ao dispositivo conectado ao computador através de uma conexão Bluetooth.

A solução proposta (representada na figura 1.4) possui duas placas microcontroladoras e um aplicativo Android que irão implementar os protocolos USB e Bluetooth de maneira que:

- Um dos microcontroladores funcione como um dispositivo USB HID.
- Outro microcontrolador consiga se comunicar com o primeiro microcontrolador, faça a conexão Bluetooth com o smartphone e tablet.
- Um aplicativo desenvolvido para o smartphone ou tablet Android que consiga interagir com o microcontrolador acima (o qual faz conexão Bluetooth).

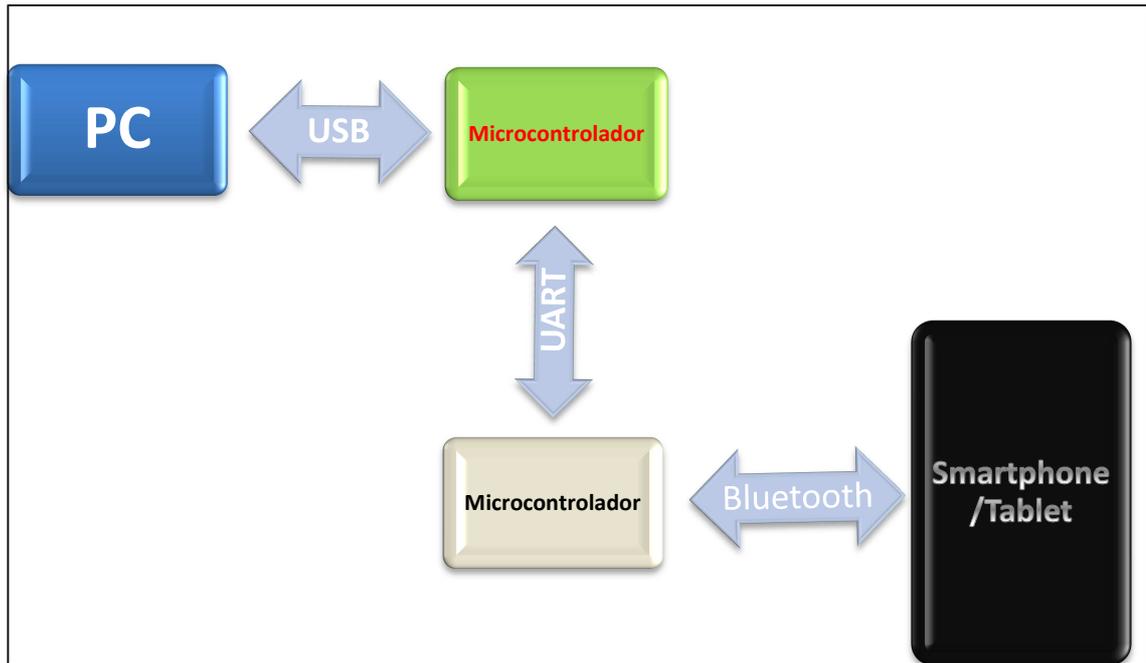


Figura 1.4: Representação do projeto

2. ESTADO DA ARTE

Nesta parte do trabalho serão apresentadas as tecnologias e conhecimentos atuais disponíveis para implementação da proposta do presente trabalho.

2.1 Interação Homem-Computador

A utilização de computadores tem requerido que um tipo de interface com as pessoas que os operam. Isto é devido ao seguinte fato: “As mais sofisticadas máquinas são inúteis, a menos que possam ser usadas propriamente por pessoas.” (KARRAY, 2008, p.138).

O fato mencionado acima delimita os termos de funcionalidade e usabilidade que tem que se levar em conta no projeto de uma interface homem-computador. Entende-se que a funcionalidade de um sistema está determinada por um conjunto de ações e serviços que fornece a seus usuários. Mas estas ações e serviços somente se traduzem em benefícios ao usuário quando podem ser usados eficientemente (KARRAY, 2008). Desta última ideia de eficiência, pode-se derivar o significado do termo usabilidade: o grau de flexibilidade, eficiência e adaptabilidade de alguma funcionalidade específica para um determinado objetivo de um tipo de usuário específico.

2.1.1 Tecnologias de IHC

As tecnologias atuais levam em consideração os aspectos físicos, cognitivos e afetivos das pessoas. No nível físico determinam-se os mecanismos de interação entre as máquinas e as pessoas, enquanto que no nível cognitivo é considerado aquilo que o usuário consegue compreender e interpretar do sistema com o qual ele está interagindo. No nível afetivo, avalia-se quanto o usuário é afetado positivamente ou negativamente quanto às impressões sobre o sistema. Isto melhora a probabilidade do usuário usar o sistema novamente.

A maioria das conquistas da IHC é na área do nível físico, que corresponde ao nível no qual a proposta do presente trabalho atinge. Neste nível, as tecnologias atuais baseiam-se nos sentidos da visão, audição e tato (três dos cinco sentidos do ser humano) (TE’ENI, 2007). No caso do estudo deste trabalho os sentidos predominantes usados pelos usuários são o tato e a visão, devido que a interação é feita através de um teclado e mouse.

Nesse sentido, os tradicionais métodos de entrada (mouse e teclado) são expandidos com o uso de outras tecnologias, como, por exemplo: redes, sensores e animações, que podem ser categorizadas em três grupos: dispositivos de vestir (wearable devices), dispositivos sem fio e dispositivos virtuais (KARRAY, 2008). A proposta apresentada neste trabalho encaixa-se nas duas últimas categorias, porque o dispositivo irá utilizar

uma conexão sem fio Bluetooth e os recursos que ele oferece ao usuário final são representações de dispositivos de entrada reais.

Enquanto avanços recentes na área de IHC pode-se mencionar as interfaces inteligentes e adaptativas e a computação ubíqua. Estas novidades envolvem a mistura e composição de todos os níveis de atividade do usuário (Físico, cognitivo e afetivo). No caso da computação ubíqua, no qual existem para cada ser humano, muitos computadores das mais diversas formas e tamanhos (SCHULMEISTER, 2008). Isto se encaixa com a Terceira Onda da Computação, que veio depois da era dos mainframes (primeira onda: para um computador, muitos usuários) e a era dos computadores pessoais (segunda onda: um computador para cada usuário).

Atualmente começa a surgir a terceira onda, com sua primeira etapa bem definida e com um mercado já conquistado, as telecomunicações pessoais por meio dos smartphones e tablets. Como foi visto na introdução deste trabalho, os computadores ainda são as ferramentas mais úteis e produtivas no momento do trabalho real. A proposta deste trabalho encaixa-se muito bem no momento atual, aproveita os abundantes recursos computacionais dos smartphones e tablets, que pertencem ao início da terceira onda da computação, e os explora para o uso nos microcomputadores, que pertencem ao fim da segunda onda computacional.

2.2 Recursos computacionais móveis para usuários finais

2.2.1 A plataforma Android

O Android é um sistema operacional e uma plataforma de software para aplicações móveis. Ele contém um conjunto básico de aplicações para tarefas cotidianas, como navegação web, leitura de e-mail, leitura de e-books, entre outras. O Android é uma plataforma de desenvolvimento móvel emergente e foi concebido com o único propósito de incentivar um mercado livre, nos quais os usuários queiram consumir e os desenvolvedores de software produzir.

A plataforma Android é projetada para ser mais tolerante a falhas do que muitos de seus antecessores. O aparelho executa um sistema operacional Linux em que os aplicativos do Android são executados em um ambiente seguro. Cada processo é executado em sua própria máquina virtual do Android e, portanto eles são muito menos propensos a levar o telefone a um estado de *deadlock*. Também diminuiu os casos de falha geral do dispositivo (também conhecidos como "Bricking"), tornando-o inutilizável.

O sistema operacional Linux (kernel 2.6) lida com serviços essenciais do sistema e funciona como uma camada de abstração de hardware (HAL) entre o hardware físico do aparelho e a camada de software Android. Algumas das principais funções do kernel incluem: execução de permissões de aplicativos e segurança gerenciamento de memória de baixo nível.

2.2.2 As capacidades dos sensores e periféricos disponíveis na plataforma Android

A plataforma Android pode acessar uma grande variedade de sensores e periféricos embutidos. Segundo (MILLETE, 2012), um sensor é um dispositivo com a capacidade

de fazer medições sobre o estado e comportamento do aparelho mestre e seu ambiente externo.

Estas capacidades de medir e “sentir” são produto do hardware disponível no aparelho Android e das composições que possam ser feitas dessas medições. Por exemplo, os sensores de uma tela capacitiva podem ser desativados quando o sensor de proximidade identificar que o smartphone está próximo ao rosto para atender uma ligação.

Os sensores disponíveis para a programação na plataforma Android podem ter a seguinte classificação, segundo (MILLETE, 2012):

- Sensores de Localização: determinam a localização de um dispositivo utilizando uma variedade de sensores como o GPS.
- Sensores físicos: Detectam propriedades específicas do dispositivo como a orientação, aceleração e rotação. Também propriedades ambientais, como luz, campos magnéticos, e pressão barométrica.
- Scanner de NFC: Detecta etiquetas (tags) do protocolo *Near Field Communication* (NFC) para interagir com outros aparelhos NFC.
- Câmera: Coleta de imagens visuais.
- Microfone: Captura áudio.
- Reconhecimento de voz: converte áudio para texto usando uma combinação de áudio gravado a partir do microfone e algoritmos de reconhecimento.
- Sensores externos: Qualquer sensor conectado usando o mecanismo Android Open Accessory (AOA).

2.3 Universal Serial Bus

O protocolo de comunicação utilizado entre o PC e sistema proposto neste trabalho é o Universal Serial Bus (USB). Todas as seções subsequentes desta etapa do trabalho foram extraídas de (ALEXSON, 2005).

2.3.1 História

Este protocolo foi criado para simplificar a diversidade de conectores que os microcomputadores precisavam para acessar os diversos periféricos de entrada/saída. Um grupo de fabricantes (Intel, Compaq, NEC e a Microsoft) trabalhou em conjunto para projetar um barramento com largura de banda suficiente para conectar diversos dispositivos sem ter que reestruturar o hardware, nem carregar necessariamente um software específico para cada novo dispositivo. Este projeto também procurava minimizar os custos dos dispositivos existentes nessa data e no futuro (ALEXSON, 2005).

Para atingir os objetivos, se estabeleceram os seguintes critérios:

- Exigir o cumprimento do *standard*.
- Permitir a detecção automática do disposto, sem necessidade de reiniciar o computador (Hot Plug&Play), além de fornecer a energia necessária ao dispositivo conectado.
- Cabeamento simples e universal, implementado de modo simples com um conector que não pode ser colocado de maneira incorreta.
- De baixo custo, mas com alta vazão.
- Diversidade de interconexões para atender dispositivos de orientados a caractere (teclados, mouse) e a bloco (vídeo, áudio, etc.).
- Variedade de tipo de transferências: Modo síncrono para áudio e vídeo, e assíncrono para mensagens.
- Arquitetura aberta para novas classes de periféricos e facilidade para conexão de dispositivos concentradores (HUB) para formar topologias de estrela e cascatas.

O protocolo USB não atribui funções específicas às linhas de sinais, nem pressupõe qual interface irá ser utilizada. No USB, não tem cabos dedicados, mas tem classes lógicas definidas que indicam os requisitos e o protocolo de comunicação a ser utilizado segundo o dispositivo a ser tratado.

O protocolo USB amadureceu muito até chegar nos padrões atuais. Ele atingiu amplamente os seus objetivos e atualmente é o padrão mais utilizado na sua categoria. A seguir se mostra sua evolução no tempo na tabela 2.1.

Tabela 2.1: Linha de tempo da evolução USB

1995	1996	1998	2000	2012
USB 0.9	USB 1.0	USB 1.1	USB 2.0	USB 3.0
Primeiro rascunho	Estabelece os tipos de conexão: Low speed (1,5Mbps) e Full Speed (12 Mbps)	Vira o standard mínimo para cumprir com o protocolo USB	Extensão da norma compatível com as anteriores e aumenta um novo tipo de conexão: High Speed (480 Mbps)	Mantém a retro-compatibilidade acrescentando a velocidade até 4.8 Gbps

Fonte: ALEXSON, 2005. p.13.

2.3.2 O conector e o cabo USB

A especificação USB define muitas formas standard de conectores, cada uma para diversas finalidades com pode-se ver na figura 2.2. Mas todas essas formas têm em comuns quatro cabos. Os sinais de USB trafegam em um par de cabos trançados que se chamam de D+ e D-, cuja impedância característica é de 90 ohms $\pm 15\%$. Ambos os cabos trançados usam sinalização diferencial em full duplex, o que ajuda a minimizar os efeitos do ruído eletromagnético que acontece em cabos compridos.

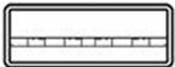
Type	Port Image	Connector Image
Type A	4.5mm x 12.0mm 	
Type B	7.3mm x 8.5mm 	
Mini-A	3.0mm x 6.8mm 	
Mini-B	3.0mm x 6.8mm 	

Figura 2.1: Alguns dos conectores e receptáculos padrão USB (INTEL CORPORATION, 2012).

Na proposta apresentada neste trabalho utiliza-se os conectores tipo A no lado do microcomputador e mini B, no lado do microcontrolador que irá implementar o dispositivo de interface humana.

2.3.3 Topologia

Um dispositivo USB está modelado da forma hospede-mestre, se adaptando às requisições do microcomputador, quem é o mestre deste barramento e o controla. Um dispositivo USB sabe como reagir nas petições do microcomputador, habilitando-se um protocolo para que o microcomputador o identifique e se comunique com ele.

Assim é possível utilizar uma topologia de estrela, e se for necessário, cascadeá-la com *hub*'s USB. No centro da estrela acha-se o hub do mestre e nas pontas da estrela pode-se engatar os periféricos USB, como se pode observar na figura 2.2.

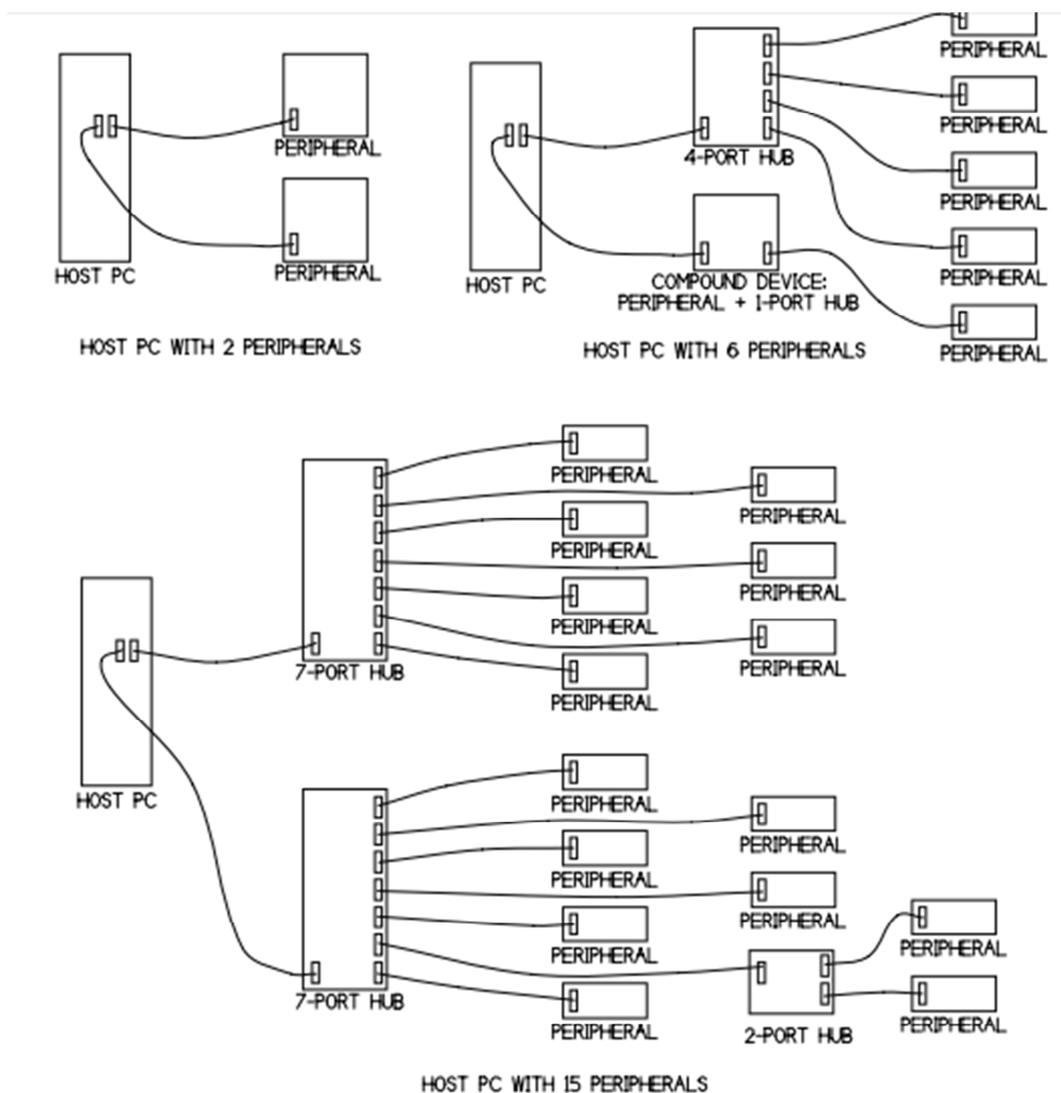


Figura 2.2: Topologia de estrela USB, cascadeada com Hubs. (AXELSON, 2005).

Como já foi dito, este protocolo trabalha fortemente acima da estrutura hospede-mestre, dividindo as tarefas entre eles.

2.3.3.1 Mestre

O mestre é o mestre do barramento, e tem que conhecer quem está ligado no barramento e quais são as características dos dispositivos conectados. O mestre tem que garantir que todos os dispositivos possam enviar e receber os dados que precisam para sua comunicação. Tudo isto é implementado com hardware e drivers no sistema operacional, fazendo com que os aplicativos que rodam acima dele não precisem conhecer os detalhes do protocolo USB.

A seguir se apresenta uma lista com as principais tarefas do mestre:

- Detectar o dispositivo
- Gerenciar o fluxo de dados
- Comprovar erros
- Fornecer energia
- Troca de dados com os periféricos

No nível de mestre, a proposta apresentada neste trabalho, aproveita-se o fato que todos os sistemas operacionais atuais mais populares (Windows, OS X, Linux, BSD etc.) têm já implementados os drivers da classe *Human Interface Device* e estão prontos para uso *Plug & Play*. Isto faz com que esta proposta seja virtualmente universal e evita a reescrita de drivers específicos para cada um de esses sistemas operacionais.

2.3.3.2 Periférico

Na maioria das vezes o periférico vai ficar aguardando ordens do mestre, para depois respondê-las. Somente nos instantes iniciais depois de engatar o periférico na porta USB, o periférico toma iniciativa e informa ao mestre para ser enumerado dentro da lista de recursos disponíveis para o sistema operacional.

A seguir são mostradas as funções principais do chip que é conectado ao periférico hóspede.

- Detectar comunicações direcionadas ao chip
- Responder solicitações padrão
- Comprovar erros
- Gestão de energia
- Troca de dados com o mestre

A proposta apresentada neste trabalho tira proveito destas características do hóspede, pois é neste nível que o microcontrolador, que irá ser conectado ao microcomputador, se encarregará de identificar como um dispositivo de interface humana (como teclado e mouse), e receberá do microcontrolador que faz a conexão bluetooth os caracteres ou coordenadas (segundo o caso) mandadas pelas entradas do usuário no tablet ou celular.

2.4 Bluetooth

Bluetooth é um padrão de tecnologia sem fio para troca de dados em distâncias curtas (usando transmissões de ondas curtas de rádio na banda ISM de 2400-2480 MHz) de dispositivos fixos e móveis, criando redes de área pessoal (PANs) com altos níveis de segurança. Foi criado em 1994 para se uma alternativa sem fio para cabos RS-232 (Huang, 2007).

Bluetooth é um padrão projetado para baixo consumo de energia e permitir muita mobilidade a periféricos de pequeno porte, devido a que não precisa ficar alinhado ao transmissor.

2.4.1 Princípios de funcionamento

Os serviços oferecidos pela pilha Bluetooth são semelhantes aos usados protocolo IP, como é sugerido por (HUANG, 2007). O processo inicial de escolha de um dispositivo de destino marca a principal diferença entre a programação Bluetooth e a programação sobre Internet.

Na figura 2.3 temos a comparação lado a lado da diferentes camadas dos protocolos TCP/IP e Bluetooth. Claramente pode-se observar no a grande similitude entre essas pilhas, tanto nas conexões de saída como as de entrada.

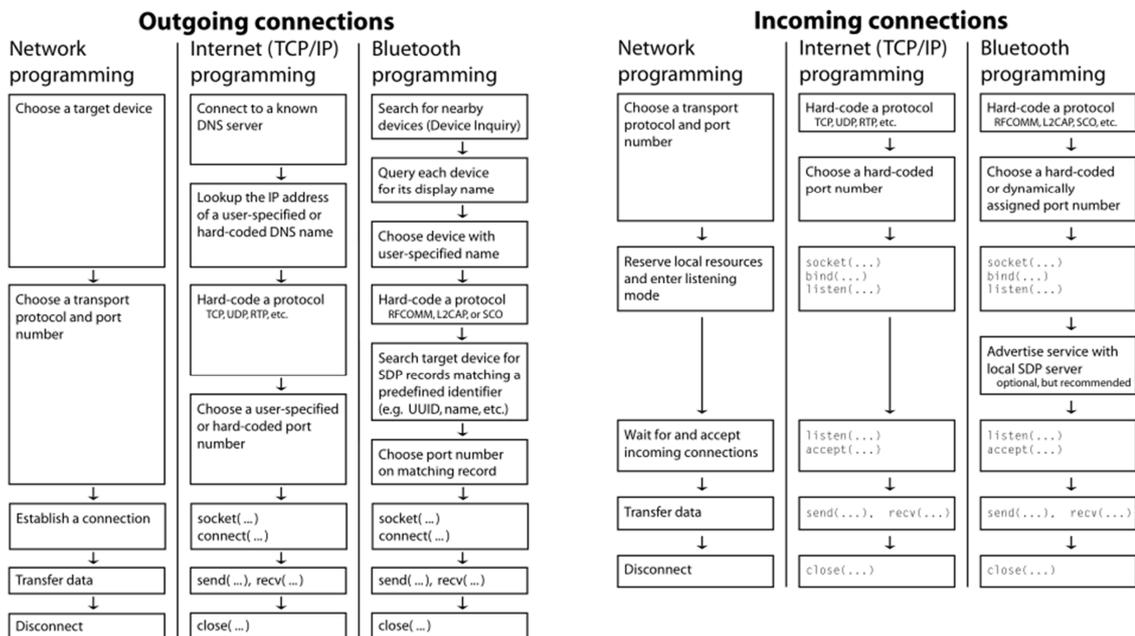


Figura 2.3: Funcionamento da pilha Bluetooth comparado com a pilha TCP/IP (HUANG, 2007).

2.4.2 Interface do Controlador Mestre

Nesta parte do trabalho se detalha como o computador mestre (Host) interage e se comunica com um adaptador Bluetooth local. Esta interface se chama de Host Controller Interface (HCI) e encapsula toda a comunicação entre o Host e o adaptador

(HUANG, 2007). Estes pacotes encapsulados existem em quatro formas diferentes, como visto na tabela a seguir.

Tabela 2.2 : Host Controller Interface

Tipos de pacotes do Host Controller Interface				
Nome do tipo	Command packet	Event packet	ACL data packet	Synchronous data packet
Funções	Controla o adaptador desde o computador mestre.	Avisa ao mestre que algum evento de seu interesse tem acontecido no lado do adaptador Bluetooth.	Serve de transporte para encapsular dados destinados por um dispositivo Bluetooth remoto	Serve de transporte para encapsular pacotes iniciais no sentido Host – adaptador Bluetooth.
Usos	Inicia eventos como pesquisa de dispositivos, conexão a dispositivos remotos, ajusta parâmetros de conexão, etc.	Durante o funcionamento normal de algum serviço Bluetooth.	Encapsulamento para outras camadas de mais alto nível.	Início de comunicações Host-Adaptador.

Fonte: HUANG, 2007. p. 33-34

Estes tipos de quadros são utilizados em operações Bluetooth de baixo-nível e frequentemente usados para acessar a camada comunicação HCI Bluetooth. Mas muitos ambientes de programação escondem esta camada, para diminuir a complexidade dela por meio de interfaces que são mais simples e seguras de usar.

Esse é o caso que acontece na proposta deste trabalho. A camada de comunicação HCI Bluetooth é implementada pela biblioteca que trabalha com Android e placa que irá fornecer a conexão Bluetooth. Esta biblioteca é a IOIOLib (BEN-TSVI, 2012) e implementa o protocolo HCI entre o microcontrolador embarcado na placa IOIO Board e um dongle Bluetooth V2 genérico, que assume as responsabilidades de adaptador Bluetooth local.

2.4.3 Interação entre a pilha Bluetooth e um aplicativo Android

As capacidades Bluetooth fazem parte da plataforma Android desde a versão 2.0 do SDK. Quando o hardware Bluetooth está presente, as aplicações Android podem (SHANE, 2011):

- Procurar e descobrir dispositivos Bluetooth e interagir com o adaptador Bluetooth.
- Estabelecer conexões RFCOMM de transferências de/para dispositivos, através de streams de dados.

- Manter conexões ponto-a-ponto e multiponto com dispositivos Bluetooth e gerenciar múltiplas conexões.

As APIs Bluetooth fazem parte do pacote de `android.bluetooth`. Para usar bluetooth deve-se ter permissão de uso dos serviços Bluetooth. Da mesma forma, os aplicativos do Android devem ter a permissão `android.permission.BLUETOOTH_ADMIN` para administrar hardware Bluetooth e serviços relacionados, incluindo tarefas ativar ou desativar o hardware e executando verificações de descoberta.

As APIs Bluetooth são divididas em várias classes úteis listadas a seguir:

- `BluetoothAdapter` representa o hardware de rádio Bluetooth no dispositivo local.
- `BluetoothDevice` representa um dispositivo Bluetooth remoto.
- `BluetoothServerSocket` é usado para abrir um socket para escutar as conexões de entrada e fornece um objeto `BluetoothSocket` quando uma conexão é feita.
- `BluetoothSocket` é usado pelo cliente para estabelecer uma ligação a um dispositivo. Depois que o dispositivo remoto é ligado, objeto `BluetoothSocket` é utilizado por ambas as partes para manter a conexão, recuperará-la em caso de quedas e controlar o fluxos de saída.

2.5 Microcontroladores

2.5.1 Novas oportunidades

Atualmente os microcontroladores têm evoluído muito e cada vez ficam mais simples de programar e com mais funcionalidades disponíveis. Iniciativas com o Firmata (STEINER, 2009) visam utilizar microcontroladores como extensões funcionais dos computadores pessoais. O Firmata é um protocolo de comunicação entre um microcontrolador e software de nível usuário rodando em um computador pessoal.

Outra alternativa é o Hiduino (DIAKOPOULOS, 2010) que é muito similar às ideias do presente trabalho. O Hiduino é um firmware para construir dispositivos USB MIDI, sem empregar drivers específicos.

Das duas ideias exploradas nesta parte do trabalho pode-se afirmar que a estratégia do protocolo Firmata é similar à estratégia empregada na parte da conexão Bluetooth entre dispositivo Android e o controlador que repassará os comandos ao computador mestre. Também se pode afirmar que a estratégia do Hiduino é similar a ideia de construir um dispositivo que seja nativo USB e que não precise de drivers específicos.

2.6 Alternativas oferecidas no mercado

Como visto anteriormente, os computadores pessoais, os smartphones e tablets não operam de forma integrada. Cada um desses dispositivos tenta suprir todas as necessidades dos usuários, sem colaboração direta entre eles.

Atualmente existem alguns empreendimentos para utilizar as características e sensores dos celulares como se fossem dispositivos independentes no computador.

O mais antigo e comum destes empreendimentos são os inúmeros aplicativos de controle remoto. Mas todos eles apresentam duas grandes limitações. Primeiro, eles precisam de algum software ou driver adicional a ser instalado no computador mestre. Segundo, os tempos de resposta não são ótimos, devido a que a maioria deles utiliza a sua conexão a internet para se conectar ao computador mestre.

Mais recentemente, existem alguns protótipos e produtos para consumidores finais que visam compor as funcionalidades presentes em tablets, laptops e smartphones.

Uma dessas tentativas já no mercado, procura juntar as capacidades do tablet ou celular a um microcomputador com um cabo USB. Este é o caso do Windroid Android Linker (U2U, 2012). Para implementar essas características o Windroid funciona sem drivers a serem instalados no computador pessoal. Mas o funcionamento é totalmente dual à proposta apresentada neste trabalho. O Windroid não aproveita os recursos computacionais embutidos no aparelho Android. O que o Windroid faz é usar o teclado e mouse de um computador pessoal como se fosse um dispositivos de entrada padrão do dispositivo Android.

O caso de uso típico do Windroid é seguinte:

O usuário tem a sua disposição um computador pessoal e um tablet. Ele conecta o tablet ao computador pessoal pelo cabo Windroid. Depois, pode utilizar o teclado e mouse do computador pessoal no tablet.



Figura 2.4: Windroid, uma aproximação ao trabalho conjunto entre o microcomputador e os smartphones/tablets (U2U, 2012).

Uma limitação desta proposta é que se o usuário quer escrever muito, por exemplo, e tem na frente um microcomputador e um tablet, faz muito mais sentido utilizar o microcomputador diretamente do que utilizar o tablet.

Um característica interessante, mas que não é ressaltada pelo fabricante do Windroid é que dá para compartilhar o clipboard e fazer transferências de dados. Mas para isto é necessário instalar um software no microcomputador mestre.

3. DESENVOLVIMENTO DO PROJETO

3.1 Visão geral

Nesta etapa do trabalho será descrito o caminho percorrido desde a ideia original, onde começou todo o esforço de desenvolvimento e as diversas opções avaliadas durante o percurso da pesquisa. Também se irão descrever as correspondentes opções que foram descartadas e aquelas que progrediram até chegar ao estágio final da proposta deste trabalho.

3.1.1 A ideia original

Esta ideia tem várias origens e resultou do processo dual de convergência digital, já discutido no capítulo introdutório deste trabalho. O processo de convergência dual consiste em virar o sentido da convergência e convertê-lo em uma divergência. Ou seja, o processo de absorção e mistura que aconteceu com muitos dos periféricos dos microcomputadores e que, atualmente, estão totalmente embutidos para trabalhar de forma integrada no smartphone ou no tablet, vão poder ser utilizados como se fossem dispositivos independentes e nativos do microcomputador.

Para isso, pensou-se em uma estrutura estrela, com um hub no centro dessa estrela e os recursos computacionais dos smartphones/tablets nas pontas. Assim estes recursos seriam vistos como se fossem dispositivos independentes e nativos no PC. Este dispositivo, quando conectado no computador, iria ser reconhecido como um Hub USB e poderia registrar cada um dos recursos computacionais oferecidos pelo smartphone ou tablet como se fosse um dispositivo USB tradicional e independente.

3.1.2 Os primeiros passos no percurso da pesquisa

Para poder fazer isto foram empreendidas a seguintes possibilidades de implementação. Começou-se pelo estudo de uma placa da Texas Instruments chamada Stellaris EKI-LM3S6965. Ela possui um microcontrolador da família ARM Cortex-M3 e conta com uma interfase de rede Ethernet, tela OLED, um leitor de cartões MicroSD, interfase USB serial, buzzer e muitos GPIO.

Esta placa foi selecionada como primeiro passo devido ao grande prestígio da marca Texas Instruments e ao suporte do framework Stellaris. O Stellaris é uma iniciativa desta marca para facilitar a implementação de software embarcado sobre estes microcontroladores. Este framework tenta mapear cada uma das funcionalidades mencionadas anteriormente como se fossem funções prontas para serem chamadas com parâmetros adequadas para cada uma das finalidades.

Depois do primeiro contato, esta alternativa parecia muito boa, pois o Stellaris também oferece uma biblioteca para implementar o protocolo USB. Mas tinha duas

grandes deficiências: a falta de recursos para implementar um USB Hub acima do protocolo USB oferecido pelo fabricante e também a inexistência de alguma pilha bluetooth compatível.

3.1.3 A segunda tentativa

Como a principal desvantagem dos outros circuitos era a falta de recursos para implementação de um Hub USB, buscou-se um *System on Chip* (SoC) que já tenha o recurso necessário. Foi analisado o Cypress CY7C65642-28LTXC, e se observou que ele fornece quatro portas USB. As opções de comunicação oferecidas por ele são: o SPI, I2C e GPIO.

Esta era a única opção disponível para implementar um Hub USB de maneira efetiva, mas apresentava muitos desafios. Primeiro, devido a que ideia original era fazer com que o dispositivo projetado se comportasse como um dispositivo Hub, mas de maneira virtual, todas as quatro portas tinham que ser redirecionadas para outro chip, que pudesse simular as entradas correspondentes de cada dispositivo ou sensor proveniente do smartphone ou tablet. Ou seja, seria necessário fazer um driver completo para cada um e achar outro SoC capaz de dar conta das requisições que o microcomputador mestre fizesse. Para obter um SoC com essas capacidades, se analisou um SoC recente de produção massiva, o Raspberry Pi.

O Raspberry Pi é um microcomputador completo com todos os recursos para ser usado como um computador pessoal. Foi lançado pela Raspberry Pi Foundation em fevereiro de 2012, com um preço de USD \$25. Ele pode ser utilizado com vários sistemas operacionais (Linux, RISC OS, FreeBSD, Plan 9, etc). O interessante do Raspberry é que tem oito GPIO, UART, I²C bus e SPI para comunicação de periféricos de baixo nível. Mas naquela data existia implementado nessa plataforma a pilha USB Host, ou seja, este dispositivo nesse momento somente poderia rodar como mestre para assim poder receber dispositivos USB, mas não poderia ser utilizado como um dispositivo USB hóspede, ou seja, com Raspberry Pi não era possível controlar as entradas redirecionadas que viessem do Cypress.

3.1.4 A escolha final

Devido ao problema exposto no paragrafo anterior, manter a ideia de um Hub Virtual ia trazer muitos mais problemas do que soluções. Então continuando a pesquisa, foi estudado um trabalho apresentado no congresso Black Hat USA '10 onde se apresenta o “USB - HID, The Hacking Interface Design”(HUSHING, 2010) utilizando uma placa com o SoC que pode ser reprogramada a vontade para poder-se identificar como um dispositivo USB da classe HID. Neste trabalho foi enfatizado o seu uso para fins maliciosos, mas, para os objetivos do presente trabalho, poderia se aplicar como um dispositivo que iria ser conectado no computador mestre.

Esta nova abordagem é mais vantajosa do que o SoC da Texas Instruments, devido a que pode o dispositivo de hardware que usaram pode se identificar no computador mestre como um dispositivo USB Composite, ou seja, que podia atuar como mais de 1 dispositivo USB, por exemplo, podia atuar como teclado, mouse e joystick ao mesmo tempo. Esta placa é oferecida pela PJRC e leva o nome de Teensy.

Enquanto isso, o dispositivo final precisava ainda componentes para a conexão sem fios com o smartphone ou tablet. Como sugerido pelo (MILLETTE, 2012) se analisou

muitas das plataformas disponíveis para fazer acessórios para Android. Entre eles, o mais destacado foi o microcontrolador da Google, o Accessory Development Kit (ADK), porém devido ao seu preço e quantidade limitada no mercado teve que ser descartado.

Dentre as alternativas mencionadas em (MILLETTE, 2012), se estudou uma derivativa do ADK que utiliza o mesmo chip PIC que tinha o kit da Google. Esta alternativa se chama de Sparkfun IOIO Board. O grande diferencial era que este chip já vinha com o USB Host, possibilitando ser ligado diretamente ao celular ou tablet Android como se estivesse sendo ligado diretamente no computador pessoal. Nessa mesma direção, encontrou-se que nas últimas atualizações no desenvolvimento da IOIO Board, o USB Host que acompanhava o chip podia ser utilizado diretamente com um adaptador USB para Bluetooth genérico (MONK, 2012). Com esta descoberta, podia-se fechar toda a ideia do presente projeto.

3.2 Estudo dos elementos envolvidos

3.2.1 Etapas da concepção ergonômica da interface

Para atingir um bom nível de ergonomia nesta proposta de trabalho, identificaram-se os conceitos mais utilizados em ergonomia de interfaces. Estes conceitos são: a correta caracterização do usuário e a definição das tarefas a serem cumpridas pela proposta

3.2.1.1 Análise do potencial usuário

Os usuários potenciais são pessoas que possuem, no mínimo, um smartphone e/ou tablet e também tem acesso a um ou vários computadores, que não necessariamente são administrados pelo usuário.

Este potencial usuário utiliza muito frequentemente o seu smartphone e/ou tablet, mas normalmente o seu trabalho principal (gerador de renda) é na frente de um microcomputador.

Este potencial usuário tem a necessidade de controlar estes microcomputadores a certa distância, além do alcance tradicional dos cabos do mouse e os teclados convencionais. Por exemplo, no momento de ministrar uma palestra.

Também este usuário potencial está acostumado com a interface WIMP (Windows, Icons, Menus e Pointer) e também tem conhecimentos de como usar dispositivos USB (como pendrives, mouses, etc.).

3.2.1.2 Tarefas do aplicativo

O aplicativo pode-se instalar diretamente no tablet ou smartphone.

Depois de instalado o aplicativo está pronto para ser usado, enquanto o transmissor USB estiver conectado no computador.

Caso seja a primeira vez que o transmissor é utilizado no tablet e/ou smartphone, é preciso fazer o processo de pareamento bluetooth (pairing), com uma senha predefinida. Depois disso, basta apenas iniciar a aplicação no smartphone para poder utilizar o teclado e mouse simples desenhado na tela como se fossem dispositivos USB HID.

3.2.1.3 Design do aplicativo

O aplicativo está projetado para capturar as entradas do usuário com o uso de botões virtuais na superfície da tela.

3.2.2 Dispositivo Android e conexão Bluetooth

3.2.3 Microcontrolador USB

Como visto anteriormente, foi escolhida a plataforma USB Teensy++ 2.0, fabricado pela PJRC, essa plataforma vem equipada com o microcontrolador Atmel AVR AT90USB1286. Resumidamente, esta placa é um socket com conector USB Mini-A Standard, um botão de Reset e um LED para o microcontrolador da Atmel (PJRC, 2012).

3.2.4 Microcontrolador Bluetooth

Para fazer a comunicação entre o smartphone e o tablet, foi escolhido o IOIO Board, fabricado pela Sparkfun. Esta placa contém o microcontrolador PIC com um receptáculo USB Host, um botão programável, e muitos pinos de entrada/saída de propósito geral (MONK, 2012). Para comunicar-se com o smartphone ou celular com a placa IOIO, é utilizado um adaptador genérico USB para o enlace Bluetooth.

3.3 Frameworks e Interfaces de comunicação

3.3.1 IOIOLib

Esta biblioteca Android permite controlar o IOIO Board desde o aplicativo no smartphone ou celular (BEN-TSVI, 2012). As características e dispositivos embarcados nesta placa podem ser acessados pelas interfaces Java contidas numa biblioteca que faz parte do projeto IOIO.

O aplicativo Android é compilado junto com a biblioteca IOIOLib, que pode ser usado com as plataformas Android a partir da versão 1.5 em diante.

Esta biblioteca fornece as funções de conexão e desconexão, atende os eventos do aplicativo e implementa as interfaces com os periféricos contidos no microcontrolador PIC, como por exemplo o UART, SPI, leds, etc.

3.3.2 Teensyduino

Trata-se de um complemento de software (add-on) para o Arduino IDE. Este complemento possibilita a compatibilidade entre muitos sketches e bibliotecas próprias do projeto Arduino (PJRC, 2012).

3.3.3 UART

A interface de comunicação selecionada para a presente proposta deste trabalho foi o Universal Asynchronous Receiver Transmitter (UART). O UART é uma interface serial de comunicação e serve como base para outros protocolos como o MIDI e RS-232.

Esta interface foi escolhida devido a que está presente nos microcontroladores dos dois circuitos e oferece suficientes capacidades para atingir o objetivo deste trabalho. Considerando que o teclado é um dispositivo cujo driver é orientado a caracteres, a interface UART pode dar conta da transmissão e recepção dos sinais que transportam as

entradas do usuário. Então é possível implementar estes serviços sobre o UART, devido a sua natureza de enviar bit a bit, bytes como unidade básica de transmissão.

O UART delimita o início e fim dos bytes que envia com um Start Bit e um Stop Bit, respectivamente. Opcionalmente, se podem utilizar um bit de paridade, mas na proposta deste trabalho não será utilizado.

Acima desta interface foi montado um sistema de troca de mensagens que identifica o destinatário do serviço que irá receber as entradas geradas a partir do dispositivo móvel.

3.4 Organização do projeto

Como já foi mencionada na introdução e objetivos deste trabalho, a proposta que se apresenta pode ser organizada em blocos definidos assim:

- Um aplicativo Android que recebe as entradas do usuário em um teclado virtual os envia para a placa IOIO através do Bluetooth. Este aplicativo controla o fluxo dessas entradas, por um mecanismo de mensagens que identificam o serviço a ser ativado no protocolo USB HID.
- Uma interface UART de comunicação entre a placa IOIO e a placa Teensy++ 2.0.
- Um software embarcado na placa Teensy++ 2.0 para recepção das mensagens recebidas pela interface UART e a sua transformação em quadros próprios do protocolo USB HID.

Estes blocos podem ser entendidos melhor quando sobrepostos acima das unidades de hardware das quais este trabalho vai dispor, como mostrado na figura 3.1. A figura mostra três blocos diferentes, o bloco com o Aplicativo Android recebe as entradas do usuário e as prepara para serem despachadas como mensagens via a interface UART. O bloco com a interface Bluetooth / UART faz a conexão entre o protocolo Bluetooth e a interface UART, para repassar as mensagens ao seguinte microcontrolador. O seguinte bloco corresponde ao Software embarcado no microcontrolador que irá conectado ao computador mestre via USB. Este bloco vai transformar as mensagens que chegaram do via UART em quadros USB HID.

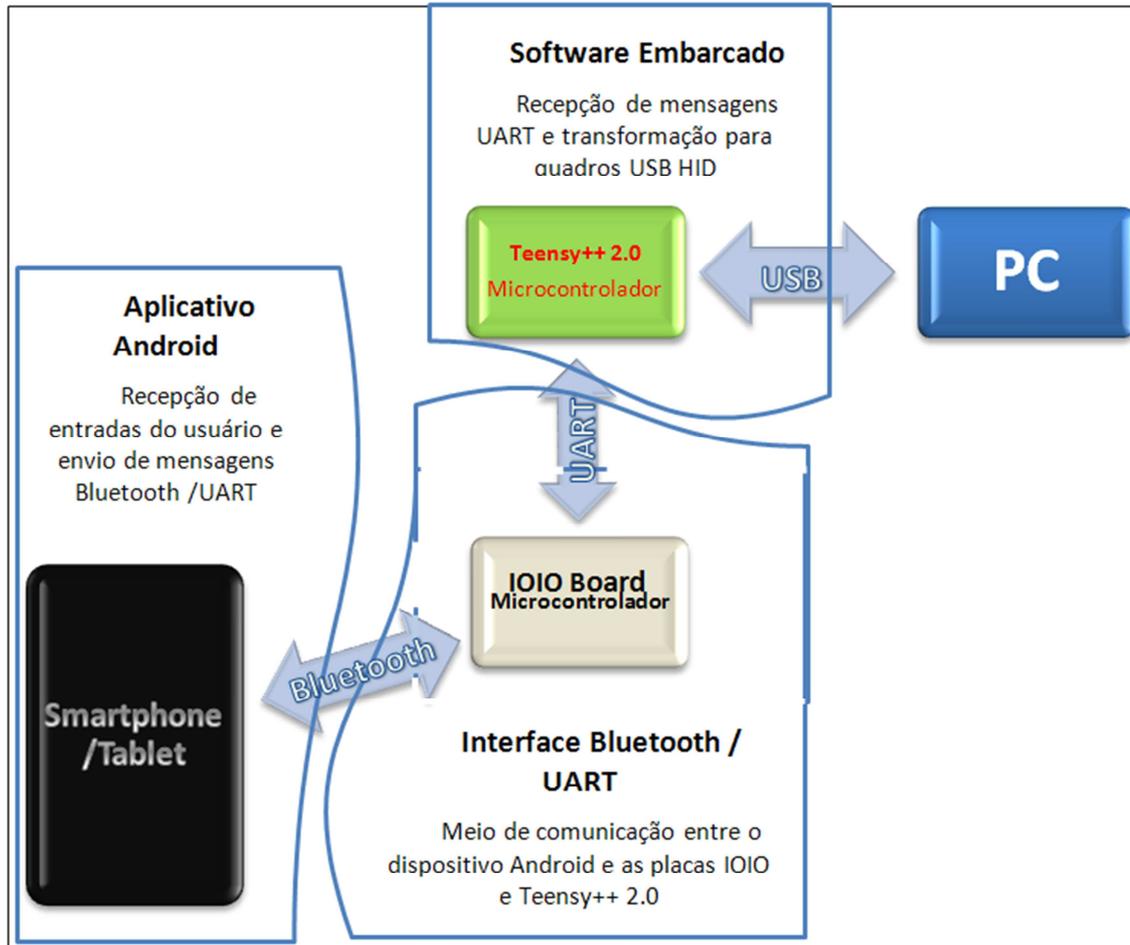


Figura 3.1: Organização do projeto em blocos

3.5 Implementação dos blocos do projeto

3.5.1 Aplicativo Android

No caso do primeiro bloco, correspondente ao aplicativo Android, foram seguidas as indicações dadas em (SHANE, 2011) sobre o uso de métodos de entrada textual, para poder aproveitar o teclado virtual fornecido pelo sistema operacional Android. O teclado virtual corresponde ao tipo de entrada *text* e *textCapCharacters*, que pertencem à classe `android.view.inputmethod.InputMethodManager`.

Para poder simular corretamente o comportamento de dispositivos diferentes e independentes entre si, optou-se por um modelo produtor-consumidor com um sistema de mensagens. Neste caso, o produtor é cada um dos dispositivos a serem simulados. Os produtos são eventos gerados pelas entradas dos usuários que se transmitirão como mensagens. O consumidor será a thread quem irá empacotar e enviar ditos eventos pelo Bluetooth com destino às portas UART de transmissão na placa IOIO.

Por exemplo, a cada vez que acontece um evento relacionado ao teclado, o produtor gera uma mensagem que indica o destinatário e o caractere a ser enviado. Este quadro é colocado em um buffer protegido por um mecanismo que oferece exclusão mútua.

A proteção fornecida a este buffer é um monitor Java, quem vai evitar que o produtor coloque uma nova mensagem quando o buffer estiver cheio, e que o consumidor retire uma mensagem, quando o buffer estiver vazio.

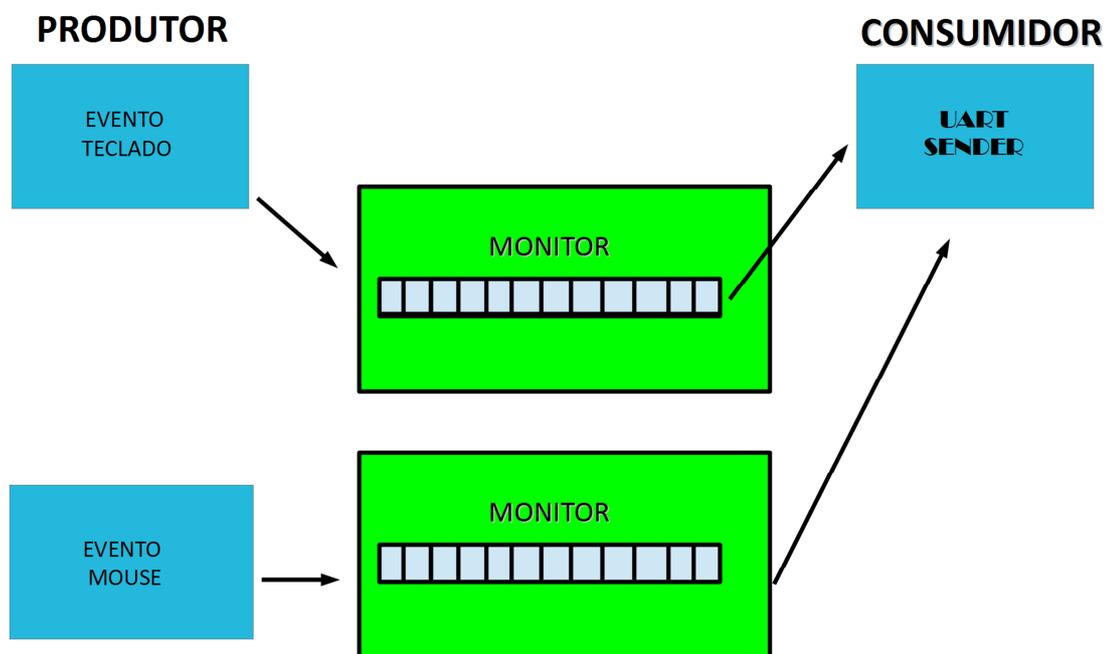


Figura 3.2: Sistema de mensagens, implementado com buffers FIFO protegidos com monitores.

O buffer tem uma política First In-First Out, que define uma ordem de sequência para os caracteres que são inseridos pelo usuário e que finalmente irão ser transmitidos pela interface serial UART.

O modelo produtor consumidor oferece certa garantia de não ter perdas no caso de acontecer múltiplas entradas concorrentes do usuário, devido à existência do buffer. Também tem que se considerar que a velocidade das entradas do usuário é muitas ordens de grandeza inferior à capacidade destes microcontroladores para processá-las.

3.5.2 Interface Bluetooth / UART

Este bloco foi implementado na sua parte lógica graças à biblioteca IOIOLib. Ela vai se encarregar de fazer o emparelhamento, a início-fim da comunicação, e de transmitir as mensagens por UART até os pinos 13 e 14 da placa IOIO. Tudo isto fica dentro do aplicativo Android e irá controlar o comportamento da placa IOIO, após o emparelhamento Bluetooth.

No pino 14 foi redirecionado o transmissor Tx da interface UART, enquanto que o receptor Rx, foi redirecionado para o pino 13. Nesta conexão elétrica, teve que se fazer um ajuste da voltagem utilizando uma resistência pull-up. Isto é devido ao fato que o a voltagem nativa na placa IOIO é de 3,3 volts, enquanto que o receptor na placa Teensy++ 2.0 funciona a 5 volts. Estas conexões podem se observar no seguinte esquemático.

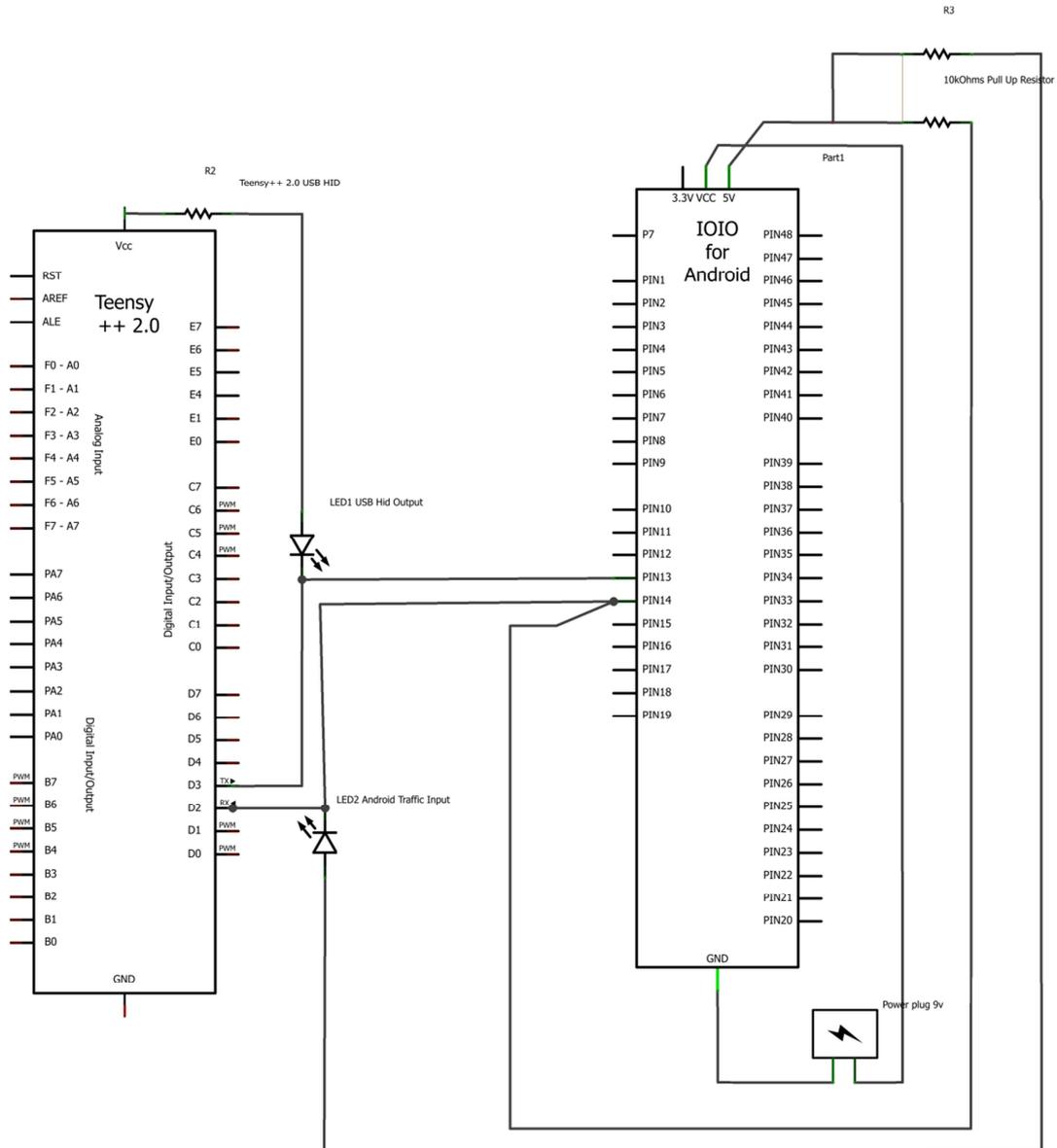


Figura 3.3: Esquemático do projeto

3.5.3 Software Embarcado

O software embarcado na placa Teensy++ 2.0 vai receber as mensagens vindas da interface UART. Estas mensagens serão recebidas no pino D2, que vai ser o receptor Rx na interface UART.

Devido às medidas tomadas no bloco anterior (placa Teensy++ 2.0), não é preciso fazer conferência alguma sobre a ordem de chegada das mensagens.

Quando chegarem as mensagens pelo UART, o software embarcado na placa Teensy++ 2.0 vai se encarregar de converter essas mensagens em quadros USB HID utilizando a Lightweight USB Framework for AVR's (LUFA).

A transformação da mensagem produzida no smartphone ou tablet para o quadro final USB HID está descrita na imagem a seguir.

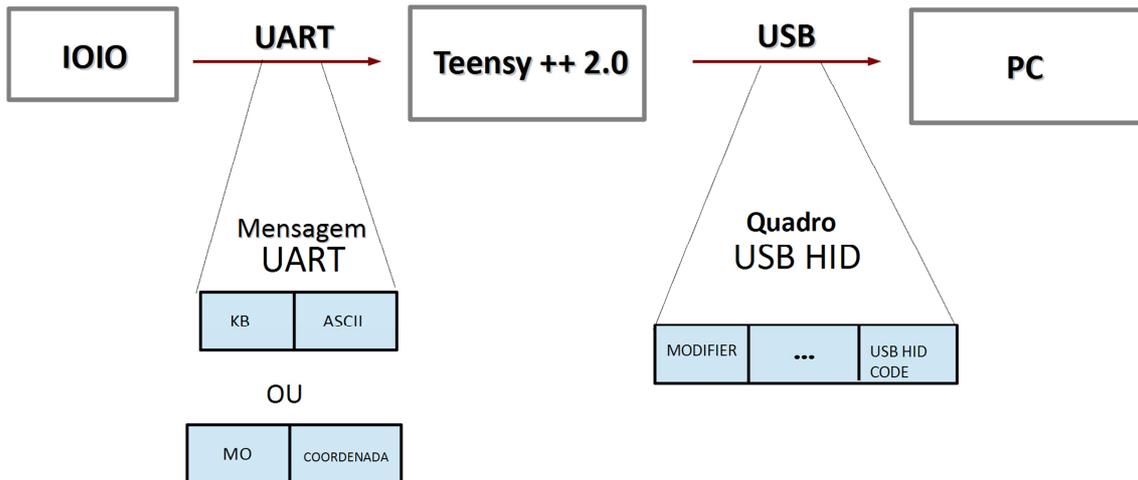


Figura 3.4: Transformação de mensagens UART em quadros USB HID.

3.5.4 Montagem física dos elementos de hardware.

A seguir se mostra como foi feita a montagem dos elementos de hardware e as conexões entre eles para a comunicação. Para maior clareza, a IOIO Board foi deslocada da sua posição final, para mostrar corretamente as suas conexões.

Na figura 3.5, se mostra como forem feitas as conexões físicas entre as placas Teensy++ 2.0 e IOIO Board. No lado esquerdo encontra-se a placa Teensy++ 2.0 e suas conexões elétricas. No lado superior direito está o IOIO Board, seus cabos de alimentação e outras conexões elétricas. As conexões da interface UART estão em verde e azul. Verde para as conexões com origem na placa Teensy, e azul para as conexões com origem da placa IOIO. Os cabos vermelhos e laranjas são cabos de força para VCC de 5 volts. O cabo laranja, também é de 5 volts, e de uso exclusivo para a resistência pull-up que irá aumentar a voltagem das interface UART da placa IOIO (cabo azul). Esta resistência vai aumentar de 3,3 volts nos quais trabalha a IOIO até 5 volts nos quais trabalha a placa Teensy++ 2.0.

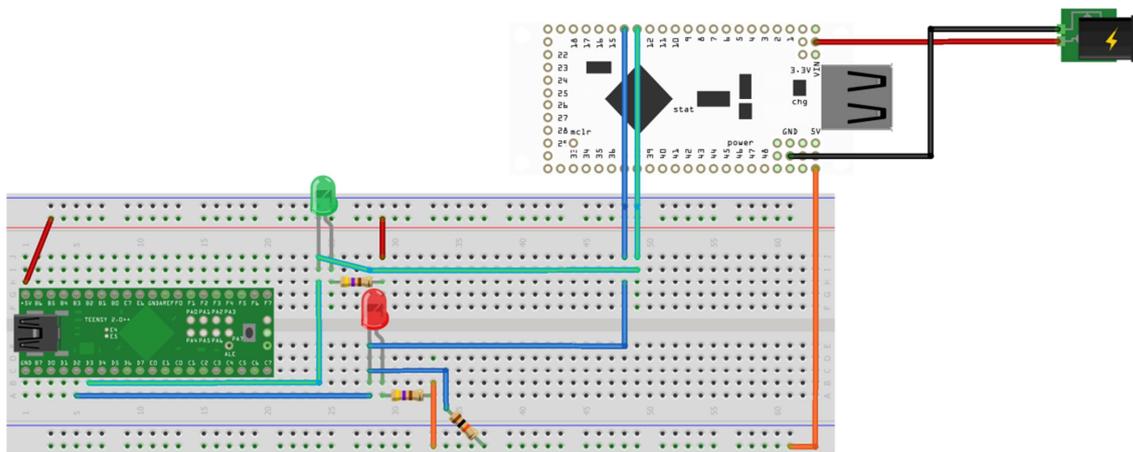


Figura 3.5: Montagem do projeto na protoboard.

4. TESTES

Os testes a seguir correspondem a cada um dos blocos da organização descrita para este projeto. Estes testes são acumulativos, ou seja, que tem que ser feitos um após o outro. Para estes testes se utilizam:

- O Teensy++ 2.0 e IOIO Board montados como na figura 3.5.
- Um cabo USB de 5 Pinos com o plug Mini-B
- Um adaptador de Bluetooth V2 (USB).
- Uma fonte de 9V, 1.7 A.
- Um tablet Android 2.3.1
- Um microcomputador com Windows XP.



Figura 4.1 : Elementos utilizadas para o teste.

4.1 Conexão USB

O teste de conexão USB tem por objetivo provar o funcionamento, conexão e reconhecimento do dispositivo como um periférico da classe USB HID.

Este teste é simples: somente basta espetar o cabo USB de 5 Pinos com o plug Mini-B no receptáculo correspondente à placa Teensy++ 2.0.

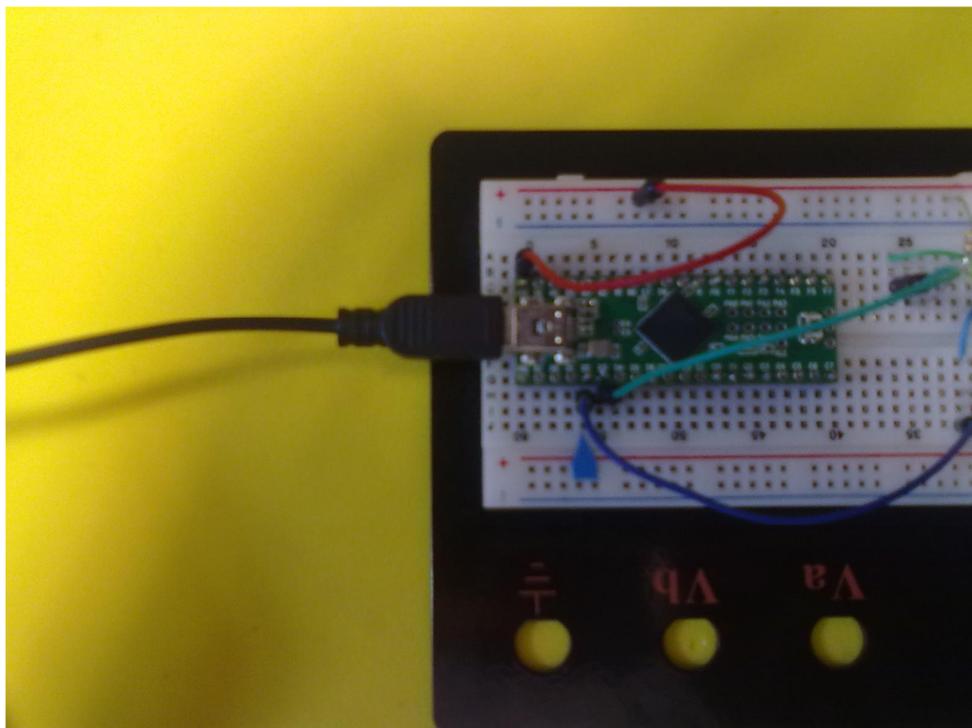


Figura 4.2 : Conexão da placa Teensy++ 2.0

A outra ponta do cabo precisa ser conectada em alguma das entradas USB disponível do microcomputador que ira ser controlado. Isto faz com que o dispositivo espetado se identifique e peça para o sistema operacional instalar os drivers correspondentes a sua classe.

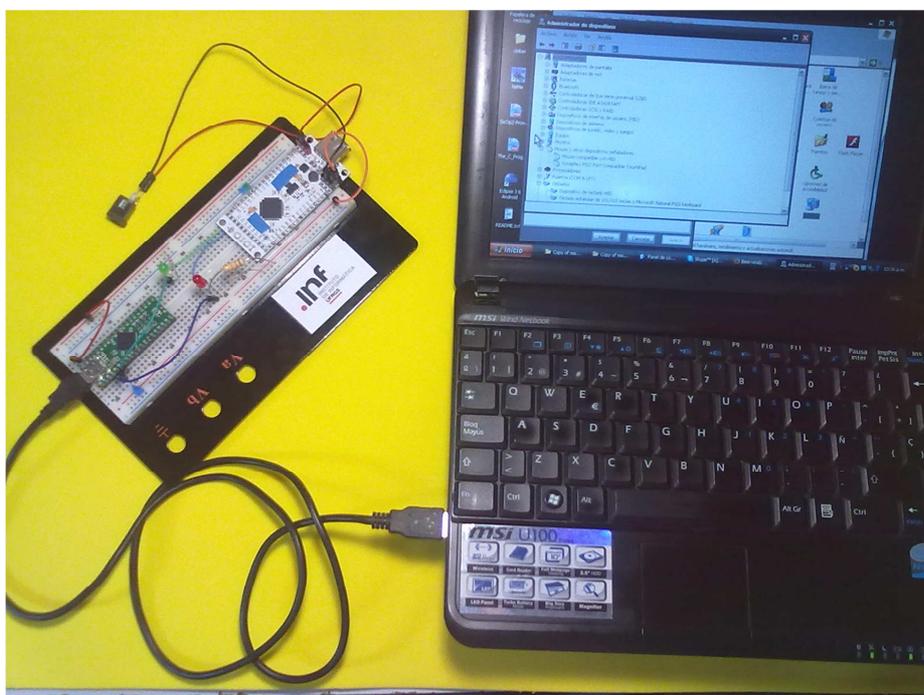


Figura 4.3 : Conexão à entrada USB do microcomputador

Para demonstrar a correta instalação e conexão, pode ser observado seu estado no Administrador de Hardware.

4.2 Emparelhamento Bluetooth

Para fazer o emparelhamento Bluetooth, primeiro tem que ligar à fonte de 9 v. o conector que vai para a placa IOIO. Depois tem que se espetar o adaptador Bluetooth V2 no receptáculo USB correspondente à placa IOIO.

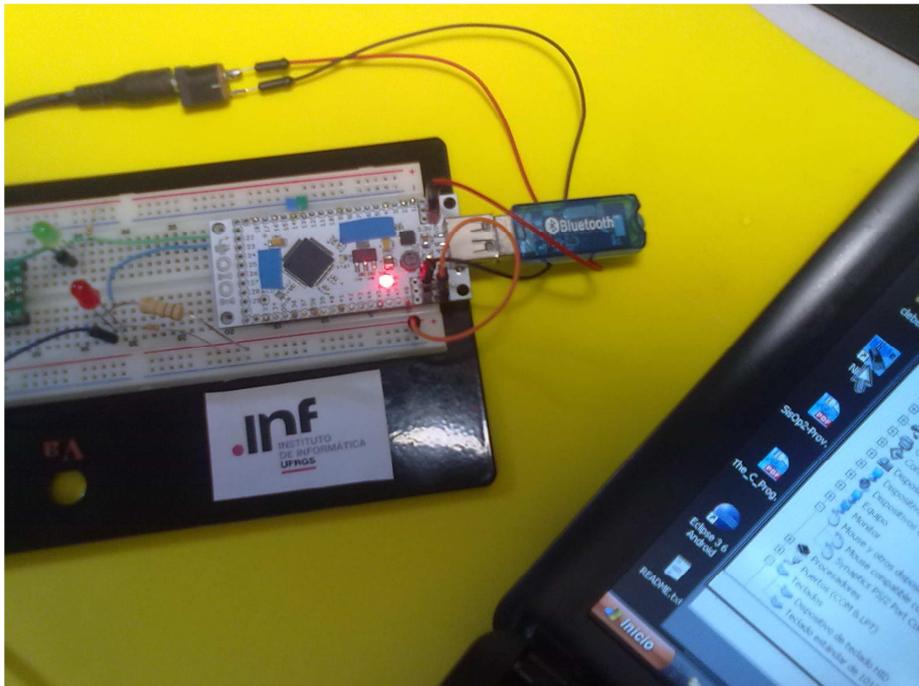


Figura 4.4: Conectado à fonte de 9 volts e depois o adaptador Bluetooth V2 na placa IOIO.

Já no tablet Android tem que se ativar o Bluetooth indo para: Configurações > Redes sem fio e outras > Conf. de Bluetooth. Tem que aparecer um *check mark* verde na *checkbox* que diz “Bluetooth”. Dai tem que apertar o botão “Procurar dispositivos”



Figura 4.5: Ativando o Bluetooth

Depois disso aparece visível um dispositivo chamado de IOIO (xx:xx), aonde xx:xx são os quatro últimos dígitos do endereço da placa IOIO. Cada placa IOIO tem um endereço diferente, dependendo do adaptador Bluetooth ao qual esteja ligado. Seleciona o dispositivo IOIO que aparece na tela para começar o emparelhamento. Imediatamente vai ser pedido um código PIN, que por default é 4545. O emparelhamento Bluetooth está pronto.

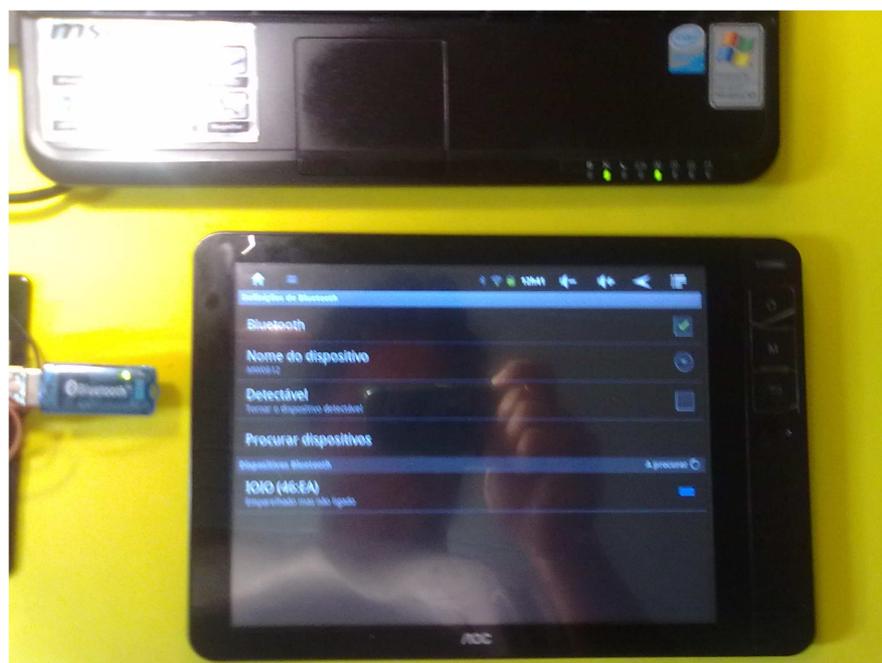


Figura 4.6: Emparelhamento Bluetooth

4.3 Envio de dados UART

Para fazer o teste de envio de mensagens pela interface UART entre as placas IOIO e Teensy++ 2.0 se utilizam a seguinte abordagem. Colocou-se um LED para cada um dos fios que comunicam os transmissores (Tx) e receptores (Rx). O LED vermelho indica que está trafegando dados com sentido do tablet até o microcomputador. O LED verde é indica o tráfego de controle que vai desde o dispositivo USB até o tablet.

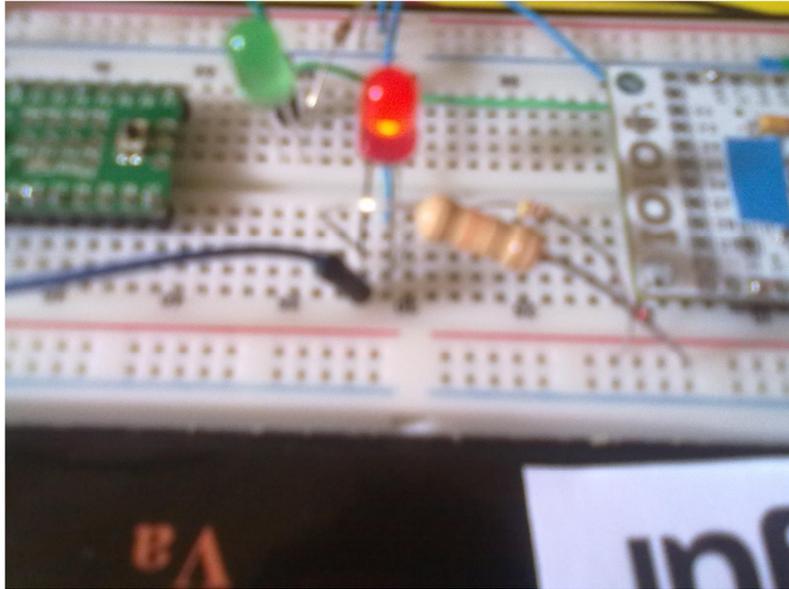


Figura 4.7: Led acesso quando passam dados pelo UART.

4.4 Produção de entradas e consumo de quadros USB HID sobre Bluetooth

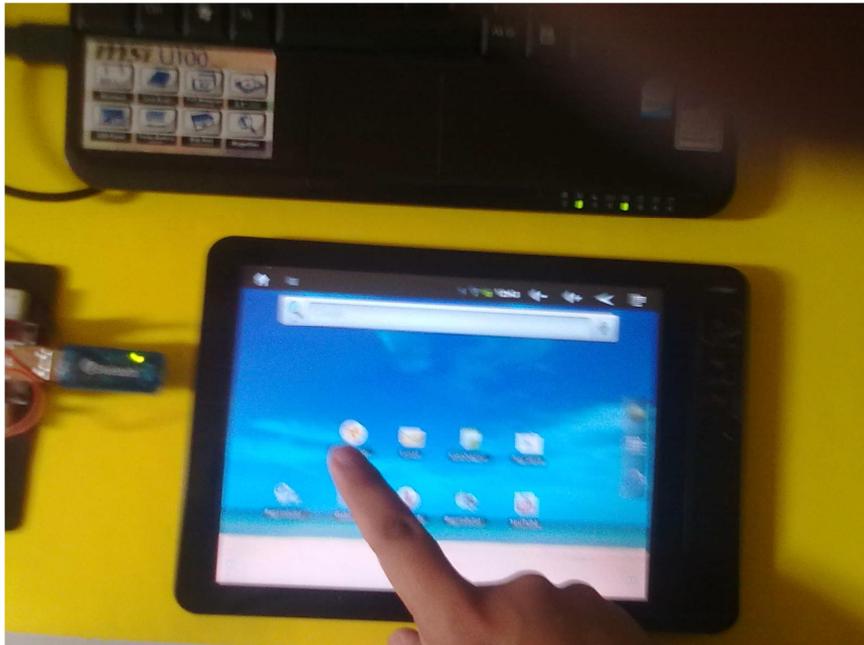


Figura 4.8: Inicializando aplicativo na tablet.

Este teste é o que fornece resultados mais concretos: é onde se utiliza o teclado virtual na tela do tablet Android. Basta inicializar o aplicativo fazendo um toque na tela para começar a usar.

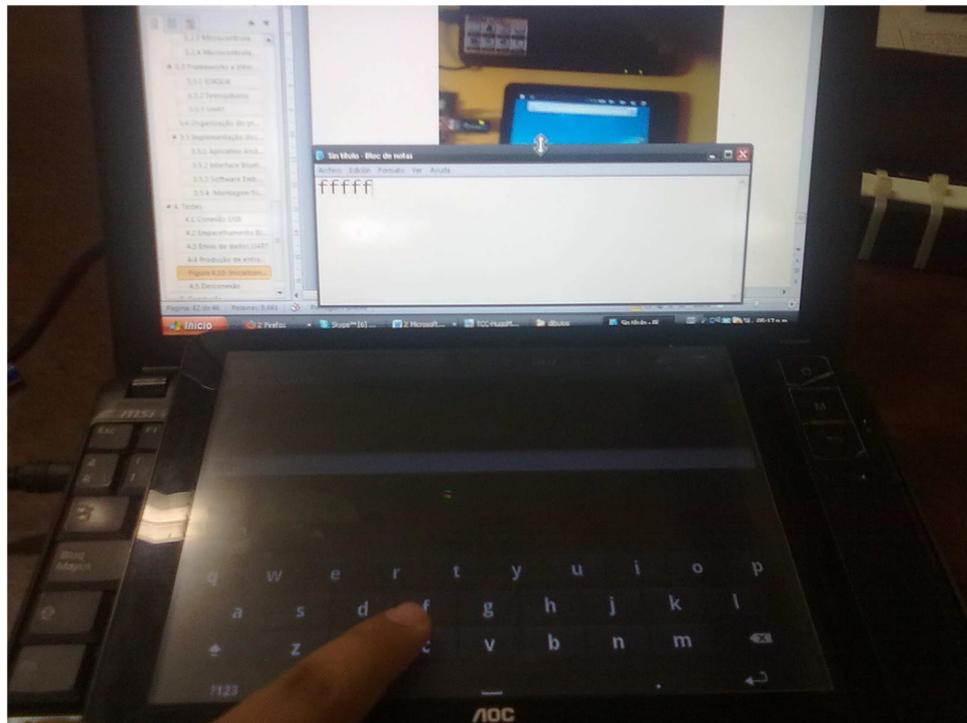


Figura 4.9: Produção de quadros na tablet e consumo deles no microcomputador.

4.5 Desconexão

Caso alguma das peças for desconectada (o cabo USB no microcomputador ou o dongle bluetooth), uma mensagem de aviso de desconexão é mostrada. Depois de fazer a reconexão, o serviço pode ser inicializado novamente.

5. CONCLUSÃO

Os muitos recursos computacionais que se tem hoje em dia podem ser melhor aproveitados quando trabalharem juntos, aplicando o melhor das características de cada plataforma.

Neste trabalho se demonstrou que esses recursos computacionais abundantes podem ser utilizados de uma maneira mais direta, transparente e universal; utilizando componentes e recursos já disponíveis para muitos usuários.

Os conceitos teóricos aplicados são simples e elegantes, mas podem ser reaproveitados para usar ainda mais recursos computacionais dos smartphones e tablets como se fossem dispositivos nativos USB.

Pode-se oferecer novas funcionalidades e novos níveis de usabilidade para muitos usuários, diminuindo a complexidade e níveis indireção para atividades do dia a dia. Estas novas funcionalidades podem ser acrescentadas acessando a mais sensores internos dos smartphones e tablets, como o microfone, sensores de luz, acelerômetros, câmeras, GPS, etc. No caso de dispositivos Android fica simples acessar esses sensores, pois este sistema operacional garante que todos os recursos de hardware podem ser acessados sem restrições desde a camada Android, sem necessidade de baixar até a camada Linux.

A usabilidade fica garantida não somente pelo amplo suporte e popularidade do protocolo USB e Bluetooth, mas também pela possível composição de mais recursos computacionais móveis. Com estes, se podem criar novos periféricos de entrada para o computador pessoal, e evoluir as interfaces unimodais dos microcomputadores atuais (que oferecem somente uma maneira de interação) para interfaces multimodais, que oferecem múltiplos modos de interação.

O grande desafio de aproveitar os diversos sensores disponíveis para Android é fazer que o produto final da composição das funcionalidades desses sensores, produza uma experiência natural e simples para o usuário.

REFERÊNCIAS

- AXELSON, Jan. **USB Complete: Everything you need to develop custom USB peripherals**. 3rd Edition. Lakeview Research LLC, 2005.
- MONK, S. **Making Android Accessories with IOIO**. 1st Edition. O'Reilly. 2012.
- SHANE, C.; DARCEY, L. **Android wireless application development**. 2nd Edition. Addison-Wesley. 2011.
- ABLESON, C. et al. **Android in Action**. 3rd Edition. Manning Publications Co, 2012.
- TE'ENI, J. Introduction. In: TE'ENI, J. et al. **Human Computer Interaction**. 1st Edition. John Wiley & Sons, Inc. Hoboken, 2007. p. 2- 9.
- MILETTE. G. Open Android Accessory. In. MILETTE. G; STROUD, A. **Professional Android Sensor Programming**. 1st Edition. John Wiley & Sons, Inc. Indianapolis, 2012 p.189- 216.
- HUANG, A. Introduction. In. HUANG, A; RUDOLPH, L. **Bluetooth Essentials for Programmers**. 1st Edition. Cambridge University Press, New York, 2007. p. 1-40.
- DIAKOPOULOS, D.; KAPUR, K. Hiduino: A firmware for building driverless USB-MIDI devices using the Arduino microcontroller. 2010. 4 f. Research paper – California Institute of Art. Valencia.
- SCHULMEISTER G. The Third Wave of Computing. 2008. 14 f. Research paper - Reykjavik University, Reykjavík.
- JEON, H. J.; KIM, M. S., SOHN, S. Y. Conjoint and WTP analyses of future mobile phones for digital convergence. **Technological Forecasting and Social Change**, v. 77, n. 3, P. 457-465, mar. 2010.

KARRAY, F. et al. Human-Computer Interaction: Overview on State of the Art. **International Journal on Smart Sensing and Intelligent Systems**. v. 1, n. 1. P 137-159, mar. 2008

STEINER, H. Firmata: Towards making microcontrollers act like extensions of the computer. **NIME Proceedings 2009**. v.1, n.1, p,125-130, jun. 2009.

INTEL CORPORATION. Desktop Boards — Troubleshooting computer accessory issues. Disponível em: < <http://www.intel.com/support/motherboards/desktop/sb/CS-023466.htm>> Acesso em: set. 2012.

RUSHING, R. USB - HID, The Hacking Interface Design. Disponível em < <https://media.blackhat.com/bh-us-10/presentations/Rushing/BlackHat-USA-2010-Rushing-USB-HID-slides.pdf> > Acesso em: out. 2012

PJRC.COM. Teensy USB Development Board. Disponível em < <http://www.pjrc.com/teensy/index.html> > Acesso em: nov. 2012

PJRC.COM. Teensyduino - Add-on for Arduino IDE to use Teensy USB development board. Disponível em < <http://www.pjrc.com/teensy/teensyduino.html> > Acesso em: nov. 2012

BEN-TSVI, Y. IOIOLib Basics. Disponível em < <https://github.com/ytai/ioio/wiki/IOIOLib-Basics> > Acesso em. dec. 2012

U2U.COM. U2U Smart KM Link. Disponível em < http://www.u2u.com.tw/U2U_web_EN/Smart-KM-Link_1.html > Acesso em nov. 2012.

