

GIOVANI MADALENA DEROSA

**DESENVOLVIMENTO DE UM SISTEMA PARA
ACIONAMENTO DE CARGAS E SENSOREAMENTO
DE UM AMBIENTE RESIDENCIAL COM DISPOSITIVO
ANDROID**

FLORIANÓPOLIS, 2012

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E
TECNOLOGIA DE SANTA CATARINA
DEPARTAMENTO ACADÊMICO DE ELETRÔNICA
CURSO DE PÓS-GRADUAÇÃO *LATO SENSU*
ESPECIALIZAÇÃO EM DESENVOLVIMENTO DE
PRODUTOS ELETRÔNICOS**

GIOVANI MADALENA DEROSA

**DESENVOLVIMENTO DE UM SISTEMA PARA
ACIONAMENTO DE CARGAS E SENSOREAMENTO
DE UM AMBIENTE RESIDENCIAL COM DISPOSITIVO
ANDROID**

Trabalho de Conclusão de Curso submetido ao Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina como parte dos requisitos para obtenção do Certificado de Conclusão do Curso de Pós-Graduação Lato Sensu Especialização em Desenvolvimento de Produtos Eletrônicos.

Orientador – Charles Borges de Lima, Dr. Eng.

FLORIANÓPOLIS, 2012

CDD 629.895
D437d

Derosa, Giovani Madalena
Desenvolvimento de um sistema para acionamento de cargas e
sensoreamento de um ambiente residencial com dispositivo Android
[monografia] / Giovani Madalena Derosa; orientação de Charles Borges de
Lima. – Florianópolis, 2012.

1 v. : il.

Monografia de especialização (Desenvolvimento de Produtos Eletrônicos) –
Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina.
Curso de Especialização em Desenvolvimento de Produtos Eletrônicos.

Inclui referências.

1. Sistema operacional Android. 2. Bluetooth. 3. Automação residencial. I.
Lima, Charles Borges de. II. Título.

Sistema de Bibliotecas Integradas do IFSC
Biblioteca Dr. Hercílio Luz – Campus Florianópolis
Catalogado por: Edinei Antonio Moreno CRB 14/1065
Rose Mari Lobo Goulart CRB 14/277

DESENVOLVIMENTO DE UM SISTEMA PARA ACIONAMENTO DE CARGAS E SENSOREAMENTO DE UM AMBIENTE RESIDENCIAL COM DISPOSITIVO ANDROID

GIOVANI MADALENA DEROSA

Este trabalho foi julgado adequado para obtenção do Certificado de Especialista em Desenvolvimento de Produtos Eletrônicos e aprovado na sua forma final pela banca examinadora do Curso de Pós-Graduação *Lato Sensu* – Especialização em Desenvolvimento de Produtos Eletrônicos do Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina.

Florianópolis, 13 de dezembro de 2012.

Banca Examinadora:

Charles Borges de Lima, Dr. Eng.
Orientador

André Luís Dalcastagnê, Dr. Eng.

Fernando Santana Pacheco, Dr. Eng.

RESUMO

Este trabalho consiste no desenvolvimento de um aplicativo para o sistema operacional Android, utilizado em dispositivos móveis, que remotamente, via Bluetooth, permite o acionamento de relés e o sensoramento de grandezas analógicas em um ambiente residencial. Isso se torna possível, através de uma placa microcontrolada e o resultado é visualizado na tela do dispositivo móvel. Utiliza-se a ferramenta oficial do Android com a linguagem de programação Java e por meio de uma placa de sistema microcontrolado, com um módulo de comunicação Bluetooth, os dados passam dos sensores de temperatura e umidade para o sistema. Dessa forma, é evidenciada a comodidade que o uso da automação residencial pode proporcionar.

Palavras-chave: Sistema operacional Android. Bluetooth. Automação residencial.

ABSTRACT

This work consists of an application developing for the Android operating system used in mobile devices, remotely, via Bluetooth, allows triggering of relays and sensing of analog quantities in a residential environment. This becomes possible through a plate microcontroller and the results are displayed on the screen of one mobile device. We use the tool with the official Android Java programming language and through a system board microcontroller, with a Bluetooth communication module, data flow from temperature and humidity sensors to the system. Thus, it is evident that the home automation can provide convenience use.

Keywords: Operating system Android. Bluetooth. Home automation.

Lista de Figuras

FIGURA 1: Mascote Android.....	20
FIGURA 2: Ambiente Android.....	22
FIGURA 3: Arquitetura do sistema Android.....	25
FIGURA 4: Logotipo Bluetooth.....	28
FIGURA 5: Redes Piconets e Scatternet.....	30
FIGURA 6: Pilha de Protocolos Bluetooth.....	33
FIGURA 7: Modelo de comunicação RFCOMM.....	36
FIGURA 8: Módulo Adaptador Bluetooth.....	41
FIGURA 9: Módulo de desenvolvimento ARM7(eLPC48).....	42
FIGURA 10: Circuito de gravação de programa.....	44
FIGURA 11: Circuito de acionamento dos relés.....	45
FIGURA 12: Circuito do sensor de luminosidade.....	46
FIGURA 13: Módulo sensor DHT11.....	47
FIGURA 14: Esquemático simplificado da placa de sensoreamento e acionamento.....	48
FIGURA 15: Android SDK Manager.....	49
FIGURA 16: O emulador Android para dispositivo virtual.....	51
FIGURA 17: Celular Atrix, com Sistema Operacional Android.....	52
FIGURA 18: Programa desenvolvido para o Android.....	53

Sumário

1 INTRODUÇÃO.....	16
1.1 JUSTIFICATIVA.....	17
1.2 DEFINIÇÃO DE PROBLEMA.....	17
1.3 OBJETIVOS.....	18
1.3.1 Objetivo geral.....	18
1.3.2 Objetivos específicos.....	18
2 FUNDAMENTAÇÃO TEÓRICA.....	20
2.1 ANDROID.....	20
2.1.1 A Plataforma Android.....	21
2.1.2 Características da Arquitetura Android.....	22
2.1.3 Construção do Núcleo Android.....	25
2.1.4 A Máquina Virtual Dalvik.....	26
2.1.5 Mercado Android.....	26
2.2 BLUETOOTH.....	28
2.2.1 Topologia da Rede.....	29
2.2.2 Endereços.....	31
2.2.3 Pilha de protocolos do Bluetooth.....	32
2.2.4 RFCOMM.....	35
2.2.5 Protocolo de descoberta de serviços.....	37
3 DESENVOLVIMENTO.....	40
3.1 MÓDULO DE COMUNICAÇÃO BLUETOOTH.....	40
3.2 PLACA DE ACIONAMENTO E SENSOREAMENTO.....	42
3.2.1 Programação e gravação do microcontrolador.....	43
3.2.2 Circuito de gravação do microcontrolador.....	43
3.2.3 Acionamentos de relés	44
3.2.4 Sensoreamento.....	45
3.3 FERRAMENTA DE DESENVOLVIMENTO PARA O ANDROID.....	49
3.3.1 O Emulador Android.....	50
3.4 O TELEFONE CELULAR (DISPOSITIVO MÓVEL).....	51
3.5 PROGRAMA DESENVOLVIDO PARA O ANDROID.....	52
3.5.1 Comunicação Bluetooth do celular.....	53
3.6 ALGORITMO E INTEGRAÇÃO DO SISTEMA.....	54
4 RESULTADOS.....	55
5 CONCLUSÕES	57

REFERÊNCIAS.....	58
ANEXO A – MÓDULO BLUETOOTH.....	59
ANEXO B – MÓDULO ELPC48	62
ANEXO C – ESQUEMÁTICO DO EPLC48.....	64
ANEXO D – SENSOR DHT11.....	66

1 INTRODUÇÃO

O conforto de se controlar as variáveis de um ambiente qualquer de uma casa é sinônimo de luxo e comodidade. Atualmente, alguns aparelhos oferecem essa funcionalidade a baixo custo, tais como controles remotos por infravermelho, por radiofrequência e circuitos fechados de televisão.

Além dessas tecnologias, existem várias outras soluções para a automação residencial desenvolvidas por grandes empresas. Soluções, por exemplo, que integram internet e sistemas com celulares. Os sistemas já existentes oferecem páginas da Web como interfaces de controle que podem ser acessadas de qualquer computador pessoal. Empresas, como a Scenario, fazem uso da internet para permitir acionamentos em equipamentos móveis através de um sistema operacional, no caso o iOS. Os equipamentos utilizados neste desenvolvimento são da empresa Apple (iPad, iPhone e iPod).

O desenvolvimento dos *smartphones* tem permitido uma melhor utilização dos sistemas de automação. Esses aparelhos têm recursos de maior memória e melhor poder de processamento, combinados com sistemas operacionais mais sofisticado, em comparação com os celulares comuns.

A motivação para este trabalho é integrar novas tecnologias no desenvolvimento de uma aplicação para dispositivos móveis baseados no sistema operacional Android, utilizando a comunicação Bluetooth para o controle das variáveis de um ambiente residencial. O *smartphone* se comunica com uma placa microcontrolada que faz os devidos acionamentos de relés e sensoramento de grandezas analógicas, cujas informações são apresentadas na tela do aparelho.

Dessa forma, é possível controlar luzes, condicionadores de ar, eletrodomésticos e outros equipamentos. Além disso, pode-se monitorar as informações extraídas de sensores espalhados no ambiente, como, por

exemplo, sensores de temperatura, umidade relativa, luminosidade e pressão barométrica.

1.1 JUSTIFICATIVA

Este projeto, através de um dispositivo móvel (*smartphone*) e de um protótipo de baixo custo, visa trazer o conforto e comodidade no controle de cargas e sensoramento de temperatura, umidade e luminosidade em um ambiente residencial.

1.2 DEFINIÇÃO DE PROBLEMA

O desenvolvimento dos *smartphones* e do sistema operacional Android tem possibilitado um grande avanço no controle de variáveis em ambientes residenciais. Com este trabalho, pretende-se desenvolver uma aplicação e um protótipo, o que se torna cada vez mais relevante em automação residencial, a fim de permitir o acionamento de relés para o controle de cargas, como por exemplo, ventiladores e lâmpadas, bem como a leitura de sensores de temperatura, umidade e luminosidade.

1.3 OBJETIVOS

1.3.1 Objetivo geral

O desenvolvimento de uma aplicação para dispositivos móveis com o sistema operacional Android é o objetivo geral deste trabalho. A ideia é utilizar a comunicação Bluetooth de um *smartphone* para ler e controlar as variáveis de um determinado ambiente.

1.3.2 Objetivos específicos

- Desenvolver uma rotina de comunicação para um módulo Bluetooth em um sistema microcontrolado de recepção de comandos enviados por um *smartphone*.
- Desenvolver um programa para o controle e aquisição de dados no sistema operacional Android.
- Desenvolver uma rotina de comunicação e leitura de sensores de temperatura, umidade e luminosidade no sistema microcontrolado.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 ANDROID

O Android é um sistema operacional para uso em dispositivos móveis (celulares e *tablets*), desenvolvido pelo Google em conjunto com 50 empresas de tecnologia e mobilidade que formam a *Open Handset Alliance*. Diversos fabricantes de dispositivos móveis adotam o sistema Android em seus produtos, tais como: Motorola, Samsung, HTC, Sony e Ericsson, o que causou uma rápida popularização do sistema.

Cada comunidade de desenvolvedores adota um mascote, ou seja, um ícone representativo da marca. Por exemplo: o Linux adota o pinguim, o GNU tem um guinu, e o Mozilla Firefox uma raposa. Para o Android, foi escolhido um pequeno robô verde (ZECHNER, 2011), que pode ser visto na Figura 1.



FIGURA 1: Mascote Android.

Fonte: <http://developer.android.com/> Acesso em: 2 ago. 2012.

2.1.1 A Plataforma Android

O sistema operacional Android possui seu código aberto, isto é, a comunidade de *software* pode contribuir com melhorias; seu núcleo é baseado na versão 2.6 do Linux (ZECHNER, 2011).

O Android possui uma interface de usuário com inúmeros aplicativos, bibliotecas de código, suporte a multimídia e outras funcionalidades. Os componentes do sistema operacional são escritos em linguagem C ou C++ e as aplicações do usuário são normalmente construídas em Java.

Uma das características da plataforma Android é que não há diferença entre as aplicações do sistema e aquelas criadas com o SDK Android (*Standard Development Kit*), que é a ferramenta de desenvolvimento para a linguagem de programação Java. Isso significa que é possível desenvolver aplicações para explorar todos os recursos disponíveis no dispositivo (ZECHNER, 2011).

A característica mais notável do Android é possuir código fonte aberto. Portanto, novos elementos podem ser fornecidos pela comunidade global de desenvolvedores.

O ambiente do *software* Android contém um *kernel* (núcleo) Linux, a máquina virtual Dalvik que executa os códigos java da aplicação com suas bibliotecas, e o *Application Framework*, entendido como um conjunto de bibliotecas do Android. Além disso, há aplicações internas e personalizadas, ou seja, programas do sistema e de terceiros. O ambiente do Android é representado na Figura 2.



FIGURA 2: Ambiente Android.

Fonte: adaptado de <http://skogberg.eu/android/>, Acesso em: 1 nov. 2012.

2.1.2 Características da Arquitetura Android

O *software* Android possui um conjunto de funções para aplicações móveis, incluindo os componentes principais de uma plataforma de computação. Em resumo, os componentes do Android são listados a seguir (ZECHNER, 2011).

a) Núcleo do Linux, que fornece uma camada de abstração fundamental do *hardware*, bem como serviços essenciais, como o processamento, gerenciamento da memória e a gestão do sistema de arquivos. No núcleo estão os aplicativos de comunicação com o *hardware* específico (*drivers*), onde são implementados recursos como o Wi-Fi e o Bluetooth. O sistema Android foi concebido para ser flexível, com muitos componentes opcionais que dependem da disponibilidade de

hardware específico. Esses componentes incluem características como: telas sensíveis ao toque, câmeras fotográficas, receptores de GPS, acelerômetros, magnetômetros e giroscópios.

b) As bibliotecas de código, que incluem:

- A tecnologia de navegador WebKit, a mesma estrutura do código aberto do navegador Safari do Mac do iPhone Mobile. O WebKit tornou-se o padrão para a maioria das plataformas móveis.

- Suporte a banco de dados via SQLite, um banco de dados fácil de usar.

- Suporte avançado a gráficos, incluindo 2D, 3D, animação *Scalable Games Language* (SGL) e OpenGL ES.

- Áudio e suporte de mídia de vídeo OpenCORE PacketVideo e do Google Stagefright.

- *Secure Sockets Layer* (SSL), recursos do projeto Apache.

c) Uma matriz de registros que dá apoio para:

- Visualização.

- Janelas.

- Os serviços baseados em localização (GPS).

- Telefonia.

d) Android *Runtime*:

- Pacotes do núcleo Java para um ambiente de programação Java (este não é um ambiente Java ME).

- A máquina virtual Dalvik, que emprega os serviços do núcleo baseado no Linux para fornecer um ambiente para hospedar os aplicativos Android.

Na Figura 3 pode-se ver de forma mais detalhada a relação entre as camadas do sistema operacional. Na parte de baixo da figura fica o núcleo do Linux com os aplicativos de comunicação com os periféricos do dispositivo. Acima do núcleo estão as bibliotecas do sistema, onde também se situa a máquina virtual Dalvik.

No topo da Figura 3 estão os aplicativos acessados pelo usuário, ou seja, onde ocorre a instalação e desinstalação de aplicativos. O desenvolvedor de programas para Android tem acesso até a camada *Application Framework* ao criar seus programas.

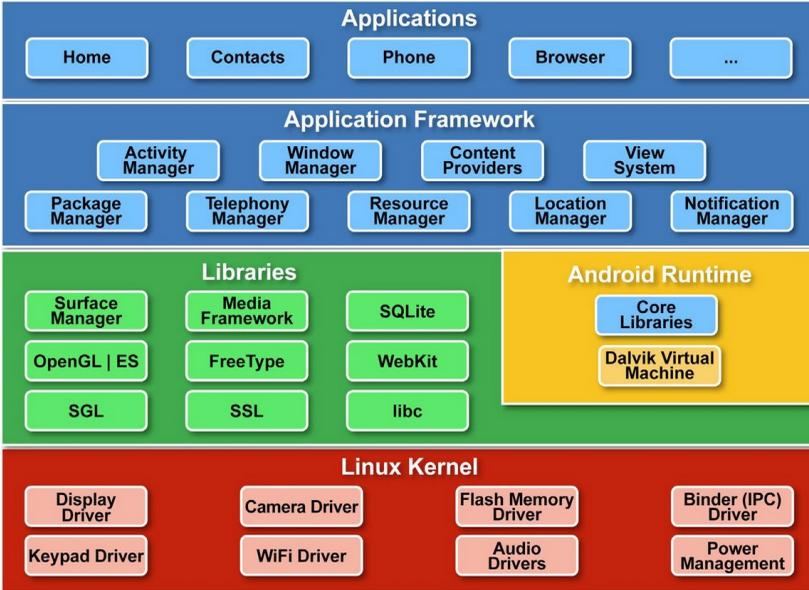


FIGURA 3: Arquitetura do sistema Android.

Fonte: <http://reply2viveksshah.blogspot.com.br/2010>, Acesso em: 7 nov. 2012.

2.1.3 Construção do Núcleo Android

O Android é construído em um núcleo Linux e com uma máquina virtual otimizada para suas aplicações Java. O componente do núcleo Linux do Android permite agilidade e portabilidade, tirando proveito do *hardware*. O ambiente Java torna o Android acessível para um grande número de programadores, porque a linguagem de programação java é muito utilizada na atualidade.

O emprego de uma plataforma cheia de recursos como o Linux e o uso de código aberto permite um desenvolvimento rápido. Tal arranjo é importante no mundo dos dispositivos móveis, onde os produtos

mudam rapidamente. Além disso, o Linux é confiável, o que é muito importante para dispositivos móveis como também fornece uma camada de abstração de *hardware*, o que facilita a programação.

Aplicações do usuário, bem como aplicações principais do Android, são escritas em Java e são compiladas em códigos de byte (bytecodes).

2.1.4 A Máquina Virtual Dalvik

Códigos de byte são interpretados em tempo de execução por um intérprete conhecido como uma máquina virtual (VM).

No Android, é utilizada a Máquina Virtual Dalvik, que foi projetada para funcionar em sistemas com baixa frequência de CPU e com pouca memória RAM. Além disso, ela é otimizada para um consumo mínimo de memória, bateria e processamento da CPU (Da Silva, 2009). Aplicações escritas em Java são compiladas em *bytecodes* Dalvik e executadas usando a Máquina Virtual Dalvik, o que permite que programas sejam distribuídos em formato binário (*bytecodes*) e possam rodar em qualquer dispositivo Android, independentemente do processador utilizado.

Do ponto de vista do desenvolvedor de aplicativos móveis, o Android é um ambiente Java, mas em tempo de execução não é estritamente uma máquina virtual Java. Isso explica as incompatibilidades entre Android e ambientes adequados de bibliotecas do Java.

2.1.5 Mercado Android

Atualmente, a grande maioria dos fabricantes de dispositivos móveis tem produtos com Android no mercado, desde modelos mais baratos até os de tecnologia de ponta em processamento.

No mercado de programas, no site Play Store do Google, pode-se encontrar programas desenvolvidos por várias empresas.

Para se tornar um desenvolvedor, é necessário comprar uma licença no valor de 25 dólares, o que permite postar programas no Play Store. Isso torna fácil a distribuição de programas. Há atualmente mais de 700 mil programas disponíveis no Play Store.

2.2 BLUETOOTH

O Bluetooth é um padrão para redes sem fio de curto alcance para comunicações de rádio (10 a 100 metros). O logotipo que identifica essa comunicação pode ser visto na Figura 4.



FIGURA 4: Logotipo Bluetooth.

Fonte: THE Bluetooth SIG Standard <http://www.bluetooth.com/> Acesso em: 4, nov. 2012.

O Bluetooth foi projetado para ter baixa potência e baixo custo. Foi desenvolvido em 1994 pela Ericsson para substituir fiações. Atualmente, está disponível em muitos tipos de dispositivos eletrônicos como: PDAs (*Personal Digital Assistant*), telefones móveis, computadores, mouses, teclados, fones de ouvido e impressoras. Em 1998, a Ericsson juntou esforços com a Intel, IBM, Nokia e Toshiba para formar o *Bluetooth Special Interest Group* (SIG). Em 1999, foram incorporados ao desenvolvimento, a 3Com, Lucent / Agere e Microsoft, fazendo com que o Bluetooth se tornasse um padrão aberto e seguro.

A primeira versão do padrão foi a 1.0 (julho de 1999), com três variações, 1.0, 1.0a e 1.0b. Em particular, as versões 1.0 e 1.0b tiveram muitos problemas, especialmente falta de interoperabilidade entre dispositivos de diferentes fabricantes. A segunda versão foi a 1.1 (fevereiro de 2001), cujas diferenças em relação à versão anterior estão relacionadas a correção de alguns problemas que surgiram durante o uso, uma otimização de desempenho, bem como a possibilidade de

canais de comunicação criptografada.

Devido ao sucesso e utilização do padrão desenvolvido, em junho de 2002, o IEEE (*Institute of Electrical and Electronic Engineers*) produziu sua própria versão do padrão chamado 802.15 ou IEEE 802.15.1 WPANs (*Wireless Personal Area Network*). Em novembro de 2003, lançou a versão 1.2 compatível com 1.1 e acrescentou melhorias para reduzir a interferência em áreas populosas por meio de seleção adaptativa de sub-banda, para maior velocidade e qualidade da transmissão e, em caso de perda de dados, retransmissão. A maior diferença ocorreu com a versão 2.0 do Bluetooth, que permite transmissões 3 a 10 vezes mais rápidas que as versões anteriores, cobrindo até 100 metros com menor consumo de energia.

O protocolo Bluetooth funciona na frequência ISM (*Industrial, Scientific and Medical*) de 2,45 GHz. Para minimizar a interferência, o protocolo divide a banda em 79 canais e alterna entre eles 1600 vezes por segundo (por um mecanismo chamado de frequência de salto ou FH) (KOBAYASHI, 2004).

A versões 1.1 e 1.2 do Bluetooth gerenciam taxas de transferências de até 723,1 kbit/s. A versão 2.0 executa um modo de alta velocidade que permite até 2,1 Mbit/s. Essa modalidade aumenta o consumo de energia. A nova versão utiliza sinais mais curtos, capaz de reduzir pela metade a potência necessária pelo Bluetooth 1.2.

2.2.1 Topologia da Rede

Dois ou mais dispositivos Bluetooth podem se conectar e compartilhar um canal de rádio, formando uma pequena rede sem fio chamada Piconet que tem uma capacidade total de até 1 Mbit/segundo.

Segundo The Bluetooth SIG Standard, 2012, nesta configuração, um dispositivo de Bluetooth atua como mestre, controlando o tráfego da Piconet, enquanto todos os outros dispositivos se comportam como escravos. O mestre é definido como aquele que inicia o processo de

conexão para estabelecer a Piconet e é único dentro da rede. O mestre estabelece o acesso à rede de escravos por meio de um mecanismo de pesquisa, ou seja, todo o tráfego deve passar por ele.

Dentro de uma Piconet pode haver até 7 escravos ativos, mas ela ainda pode ter até 255 dispositivos adicionais, que devem estar no estado inativo, ou seja, não pode transmitir, mas manter sincronização com o mestre, e podem ser conectados a qualquer momento (STALLINGS, 2002).

Duas Piconets espacialmente sobrepostas podem ser interligadas através de um dispositivo Bluetooth para formar uma Scatternet. Os escravos podem pertencer a Piconets diferentes, utilizando multiplexação por divisão de tempo, enquanto o mestre de uma Piconet pode ser escravo em outra Piconet. Possíveis topologias de rede Bluetooth podem ser vistas na Figura 5.

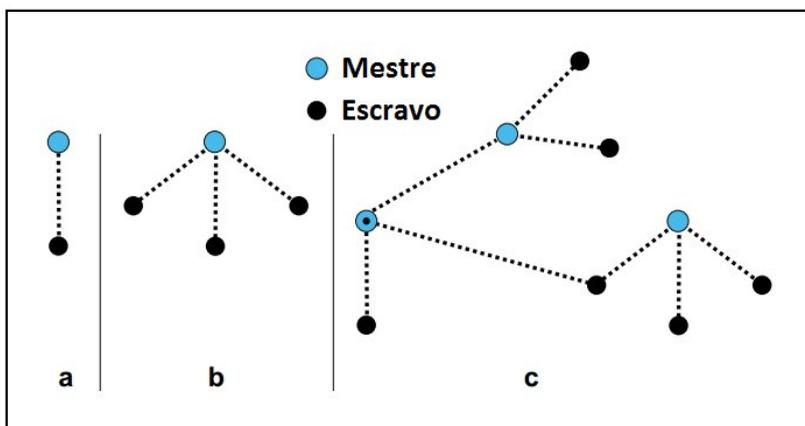


FIGURA 5: Redes Piconets e Scatternet: a) Ponto a ponto, mestre escravo; b) Rede Piconet; c) Rede Scatternet, formada por três Piconets.

Fonte: Adaptado de Specification of the Bluetooth System 2001.

As Piconets interconectadas na Scatternet formam a estrutura da Rede Móvel (MANET), que permite que dispositivos que não se comunicam diretamente, ou que estão fora do alcance para troca de dados, se comuniquem através de várias etapas da Scatternet. As implementações atuais do Bluetooth são principalmente por ponto simples ou ponto a ponto.

2.2.2 Endereços

A cada chip Bluetooth é atribuído um endereço único de 48 bits, muito similar ao endereço MAC Ethernet. Esses endereços são geridos pela mesma entidade, a Autoridade de Registro do IEEE. Esses endereços são colocados no dispositivo durante a fabricação e permanecem estáticos durante o tempo de vida do chip.

Um dispositivo Bluetooth pode enviar a seguinte informação:

- DeviceName.
- DeviceClass.
- Lista de serviços.
- Informações técnicas, tais como: funcionalidade, fabricante e especificação Bluetooth.

O DeviceName é a lista de nomes de dispositivos ao alcance.

O DeviceClass refere-se às classes de potência. Dispositivos Bluetooth são divididos em três categorias distintas, cada uma delas correspondendo a uma potência de transmissão de rádio:

- Classe 1: dispositivos de longo alcance (100 metros), com uma potência máxima de 20 dBm (100 mW).

- Classe 2: dispositivos de médio alcance (10 metros), com uma potência máxima de 4 dBm (2,5 mW).

- Classe 3: para curto alcance (1 metros), com uma potência de saída de 0 dBm (1 mW).

Na classificação estabelecida com referência aos dispositivos da classe 1, a distância percorrida é inteiramente teórica, dado que a aplicação é influenciada pelo *hardware* empregado, e pelos obstáculos e, em menor medida, pelo ruído gerado por outras fontes de sinais de rádio, tais como: as redes celulares ou Wi-fi. Além disso, cada classe determina o consumo de energia do dispositivo. Atualmente, a maioria dos dispositivos pertence as classes 2 e 3.

Qualquer dispositivo pode enviar um pedido para procurar outros dispositivos ao alcance, pode ser configurado para conectar qualquer dispositivo e se deve ou não responder a esses pedidos. Se o dispositivo solicitante já conhece o endereço, ele pode responder imediatamente, muitas vezes, porém, os endereços são substituídos por nomes atribuídos pelos usuários ou a partir do fabricante, como a marca e o modelo. Isto porque seria muito complexo visualizar números de 48 bits, do mesmo modo que os endereços IP. É preferível mostrar o nome do usuário para a identificação.

2.2.3 Pilha de protocolos do Bluetooth

A tecnologia Bluetooth foi desenvolvida através de uma pilha de protocolos. É essa pilha de protocolos que permite a localização, conexão e troca de dados entre os dispositivos.

A especificação da pilha de protocolos do Bluetooth é apresentada na Figura 6, sendo dividida em três grupos ou camadas lógicas:

1. Grupo de Protocolo de Transporte.
2. Grupo de Protocolos de Middleware.
3. Grupo de Aplicativos.

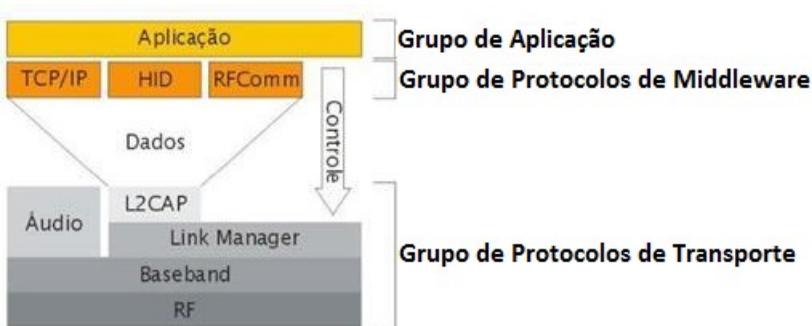


FIGURA 6: Pilha de Protocolos Bluetooth.

Fonte: http://www.gta.ufrj.br/grad/09_1/versao-final/rssf/padroes_ieee.html.

Acesso em: 4 nov. 2012.

O Grupo de Protocolo de Transporte permite que os dispositivos Bluetooth possam descobrir outros dispositivos e gerenciar as conexões física e lógica. A utilização da palavra "transporte" não indica uma ligação com a camada de transporte no modelo de referência OSI (*Open Systems Interconnectio*) de redes de computadores. Em vez disso, corresponde à camada de ligação física.

Níveis de Rádio, *Baseband*, *Link Manager*, *Logical Link Control* e *Adaptação* (L2CAP) e *Host Controller Interface* (HCI) são agrupados no Grupo de Protocolo de Transporte. Esses grupos suportam transmissões síncrona e assíncrona. Todos os protocolos nesse grupo são necessários para a comunicação entre dispositivos Bluetooth.

O Grupo de Protocolo Middleware utiliza os protocolos da camada de transporte para dar suporte à camada de aplicação, permitindo que as aplicações utilizem um nível de abstração elevado em relação aos detalhes da comunicação (Miller e Bisdikian, 2000).

Os protocolos desenvolvidos pela The Bluetooth SIG Standard incluem:

- Emulador de porta serial (RFCOMM), que permite que aplicações operem com o protocolo serial padrão, por exemplo, o RS232.
- Protocolo baseado em pacotes *Telephony Control Signaling* (TCS), para lidar com operações de telefonia.
- *Service Discovery Protocol* (SDP), que permite que um dispositivo possa obter informações sobre os serviços disponíveis em outros dispositivos.

O Grupo de Aplicativos inclui aplicações que utilizam conexões Bluetooth. Detalhadamente os níveis do grupo de transporte são:

- Nível de Rádio: refere-se principalmente ao modo físico de transmissão. O módulo de rádio de um dispositivo Bluetooth é responsável pela modulação e demodulação de dados nos sinais de RF para a transmissão sem fio.
- Nível Baseband: define como um dispositivo Bluetooth se conecta a outros dispositivos. Também a esse nível são definidos os tipos de pacotes, procedimentos de processamento e estratégias para a detecção e correção de erros, mudanças de transmissão do sinal de criptografia e retransmissão de pacotes.
- Nível de Link Manager: implementa o *Link Manager Protocol* (LMP), que gerencia as propriedades da interface de rádio entre os dispositivos e serve para traduzir os comandos do HCI em operações ao nível baseband. O LMP gera o Piconet (link set-up), cuida da troca mestre-escravo se necessária, gerencia a alocação de largura de banda e reserva de recursos para o tráfego, executa a autenticação usando um sistema de negociação, cuida do pareamento entre os dispositivos, a

criptografia de dados e gerencia o consumo.

- Nível HCI: define uma interface padrão para os aplicativos acessarem do nível mais alto para os mais baixos. É uma interface comum entre *hardware* e *software*. A finalidade é permitir a interoperabilidade entre os dispositivos e o uso de protocolos e aplicações a um nível superior.

- Nível L2CAP: fornece a interface entre os protocolos da camada superior e protocolos da camada de transporte. O L2CAP suporta a multiplexação de vários protocolos de alto nível, tais como RFCOMM e SDP. Isso permite que uma multiplicidade de protocolos possam partilhar a interface. O L2CAP é responsável também, pela segmentação de pacotes e por monitorar e manter o nível de qualidade e conjunto do serviço. Acima do L2CAP, os níveis restantes da pilha de protocolo Bluetooth não são ordenados de maneira linear.

2.2.4 RFCOMM

O protocolo RFCOMM é um protocolo de transporte que fornece simulação de portas serial sobre o protocolo L2CAP proporcionando um formato comum para consolidar os vários formatos em um único padrão de comunicação serial. Muitas vezes referido como um protocolo de substituição de cabos, sendo específico para comunicação serial, tem a vantagem de permitir que os desenvolvedores de *software* possam usar aplicativos que usam portas seriais. Essa virtualização traz consigo as operações da porta COM, como fragmentação, multiplexação e outras características típicas da comunicação serial. O nome do protocolo é derivado de RF orientada a emulação de porta serial.

O RFCOMM é baseado no padrão ETSI TS 10.07, embora apenas parte deste padrão seja utilizado, existindo algumas adaptações nas especificações do protocolo RFCOMM. Esse protocolo pode suportar até 30 conexões simultâneas entre dois dispositivos, ou seja, em

um computador poderia se ter até 256 portas COM, e neste caso apenas 30.

Na Figura 7 é apresentada comunicação entre dois dispositivos, onde o protocolo de transporte RFCOMM dá suporte ao protocolo L2CAP.

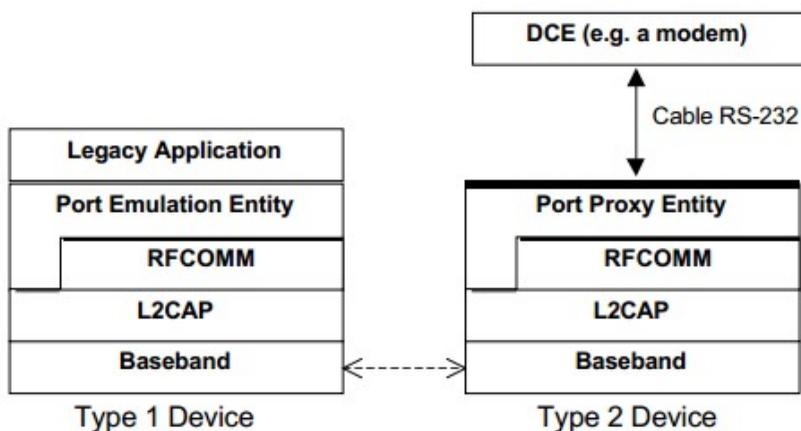


FIGURA 7: Modelo de comunicação RFCOMM.

Fonte: Specification of the Bluetooth System 2001.

O protocolo RFCOMM fornece aproximadamente os mesmos serviços e confiabilidade que o TCP. Em geral, as aplicações que utilizam TCP referem-se ao modo ponto a ponto sobre o qual podem enviar um fluxo de dados de maneira confiável. O RFCOMM têm aproximadamente as mesmas características principais do protocolo TCP (KANSAL, 2002).

Além disso, os aplicativos que usam como protocolo de transporte L2CAP devem aderir a uma determinada unidade de transferência máxima (*Maximum Transmission Unit* o MTU), uma conexão RFCOMM consiste de um fluxo de dados, o que simplifica muito a programação.

A maior diferença entre o TCP e o RFCOMM, do ponto de vista

do desenvolvedor, é a escolha do número de portas: TCP suporta até 65535 portas abertas em uma mesma máquina, ao passo que o RFCOMM permite apenas 30. Isso tem um impacto significativo sobre como escolher quais portas usar para aplicativos de servidor. O fato de o protocolo de transporte ter sido projetado com poucas portas disponíveis, significa que não se pode escolher arbitrariamente as portas no projeto. A consequência disso é que há maior possibilidade de colisão entre os números de porta com os aplicativos de servidor. Nesses casos, o desenvolvedor deve usar um recurso do Bluetooth, que é o protocolo de descoberta de serviços SDP (Service Discovery Protocol).

2.2.5 Protocolo de descoberta de serviços

O protocolo de descoberta de serviços, SDP (Service Discovery Protocol), permite que dispositivos e aplicativos possam procurar os serviços oferecidos por outros dispositivos e aplicações. Um serviço é uma entidade capaz de fornecer informações, realizar uma ação ou controlar um recurso para outras entidades. Um serviço pode ser implementado em *hardware* ou *software*, ou uma combinação dos dois (KANSAL, 2002).

O SDP é a parte da pilha de protocolo Bluetooth que aborda o problema de descoberta do serviço em um ambiente altamente otimizado para a natureza dinâmica da comunicação Bluetooth. O SDP está preocupado essencialmente em descobrir os serviços disponíveis a partir ou através de dispositivos Bluetooth, não define, porém, os métodos para acessar os serviços. Uma vez que o serviço foi encontrado pelo SDP, ele pode ser utilizado por outros protocolos definidos pela especificação Bluetooth; o modo de acesso depende do serviço em particular.

O mecanismo de descoberta de serviço, permite que um cliente possa descobrir a existência e os atributos dos serviços prestados pelas aplicações servidores. A interação acontece entre um cliente e um servidor SDP em uma ou mais operações, cada uma consistindo de uma

solicitação e resposta.

Cada pedido ou resposta consiste, por sua vez, de um único PDU (*Protocol Data Unit*). Um dispositivo Bluetooth pode atuar tanto como cliente quanto como servidor SDP. Se não é uma prestação de serviços, não há necessidade para o servidor e cada dispositivo pode ter apenas um servidor e um cliente.

3 DESENVOLVIMENTO

Este trabalho é composto por duas partes: a placa microcontrolada de sensoramento e acionamento com comunicação Bluetooth e o aplicativo com comunicação Bluetooth para aparelhos com sistema operacional Android.

3.1 MÓDULO DE COMUNICAÇÃO BLUETOOTH

Existem vários módulos para a implementação do protocolo serial padrão através da comunicação Bluetooth. Neste trabalho, foi utilizado o modelo JY-MCU produzido pela empresa Jiayuan Electronic (Figura 8), o qual opera na classe 2, pode ser alimentado por 3 a 5 volts, com um consumo de 40 mA no modo ativo e 1mA no modo de espera (*standby*). Possui 6 pinos, com alimentação nos pinos 2 e 3, e comunicação serial nos pinos 4 e 5, TX e RX, respectivamente. Possui um pino de *status* (pino 6), que é utilizado para indicar se o dispositivo está ou não conectado a outro, formando uma rede Piconet. O pino 1 controla o modo *standby*. No Anexo A, pode ser visto o manual do módulo.

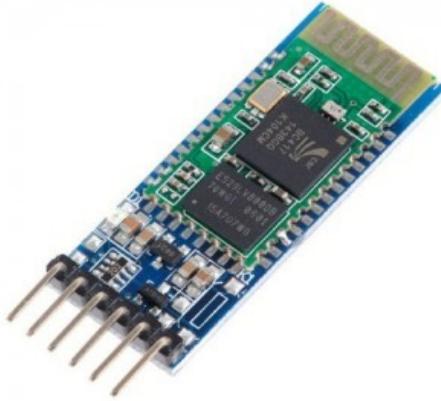


FIGURA 8: Módulo Adaptador Bluetooth.

Fonte: <http://shop34694757.taobao.com/>. Acesso em: 30 ago. 2012.

No módulo da figura 8 há também um led de status, piscando quando não está se comunicando e permanecendo ligado quando está fazendo parte de uma rede Piconet.

Através de comandos seriais, é possível configurar vários parâmetros de operação do módulo, tais como: velocidade de transmissão, nome do dispositivo e senha de conexão. Também é possível a leitura dos parâmetros configurados.

Neste trabalho foi utilizado uma taxa de comunicação de 9600 bps, com 8 bits de dados, 1 bit de parada, sem bit de paridade, usando somente os pinos TX e RX, ligados, respectivamente, aos pinos RX e TX do microcontrolador.

Ao ser alimentado, o módulo fica pronto para se conectar a uma rede Piconet. É neste momento que o aplicativo Android ativa o Bluetooth do aparelho do telefone celular, permitindo o pareamento com o dispositivo externo.

3.2 PLACA DE ACIONAMENTO E SENSOREAMENTO

Os comandos enviados pelo telefone celular são recebidos pelo módulo Bluetooth, que os envia ao microcontrolador da placa de acionamento e sensoreamento. O microcontrolador processará os dados recebidos, podendo acionar relés, além de fazer a leitura de sensores, enviando os resultados para o sistema Android no telefone celular.

Foi usado um microcontrolador ARM7, para a placa de sensoreamento e acionamentos, com a capacidade para execução das tarefas pretendidas, bem como permitindo futuras expansões.

Para o desenvolvimento do *hardware* foi empregado o módulo de desenvolvimento eLPC48 produzido pela empresa eSysTech (ver Figura 9). O módulo utiliza um ARM7 LPC2103 o qual possui 32 kB de *flash*, 8 kB de memória RAM, 32 pinos de I/O, ADC de 10 bits com 8 entradas. Também apresenta circuitos de alimentação de 1,8 V e 3,3 V, sendo alimentado com 5 V. Além disso, possui um circuito de *reset* e cristal de 32,768 KHz para o RTC (*Real Time Clock*). O Manual do eLPC48 pode ser visto no Anexo B e seu esquemático, no anexo C.

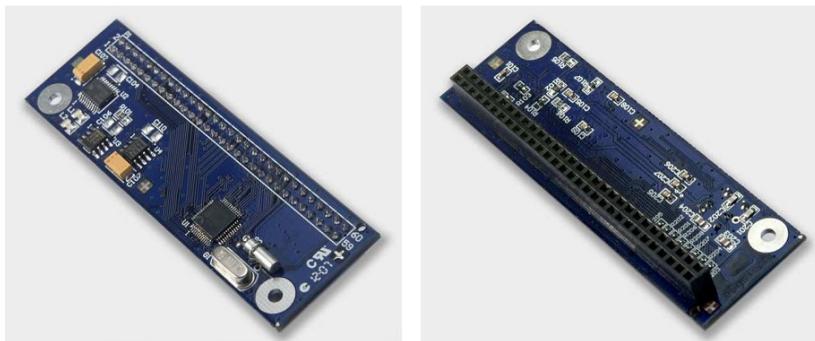


FIGURA 9: Módulo de desenvolvimento ARM7(eLPC48).

Fonte: eSysTech, 2009.

No módulo eLPC48, os pinos do microcontrolador e os demais relacionados a sua alimentação estão disponíveis através de um conector de 60 pinos.

3.2.1 Programação e gravação do microcontrolador

Para o desenvolvimento do programa, que é executado na placa de recepção e acionamento, foi empregada a linguagem C, usando o programa Keil, com seu devido compilador para o LPC2103. Para a gravação do programa na memória flash do microcontrolador foi usado o programa FlashMagic (do fabricante NXP).

3.2.2 Circuito de gravação do microcontrolador

O modo de gravação escolhido foi o modo ISP (*In-System Programming*) que utiliza os pinos TX0, RX0 e P0.14 do LPC2103, cuja gravação é gerenciada por um programa de *bootloader* nativo do microcontrolador.

O programa FlashMagic utiliza a comunicação serial padrão para a gravação. Isso é feito com o emprego de um cabo conversor serial/USB, que após instalado, cria uma porta COM virtual no computador.

Na Figura 10 é apresentada a conexão do microcontrolador, ao conversor serial/USB. Os sinais DTR e RTS são utilizados para o controle da gravação e os sinais RX e TX são responsáveis pela comunicação.

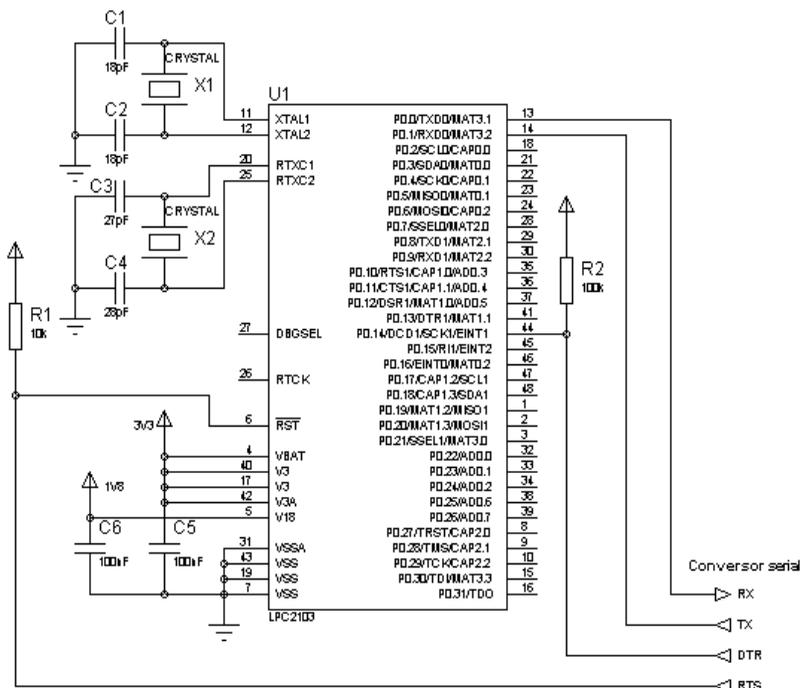


FIGURA 10: Circuito de gravação de programa.

3.2.3 Acionamentos de relés

O acionamento dos relés é realizado através de chaves transistorizadas ligadas a pinos do microcontrolador, como apresentado na Figura 11.

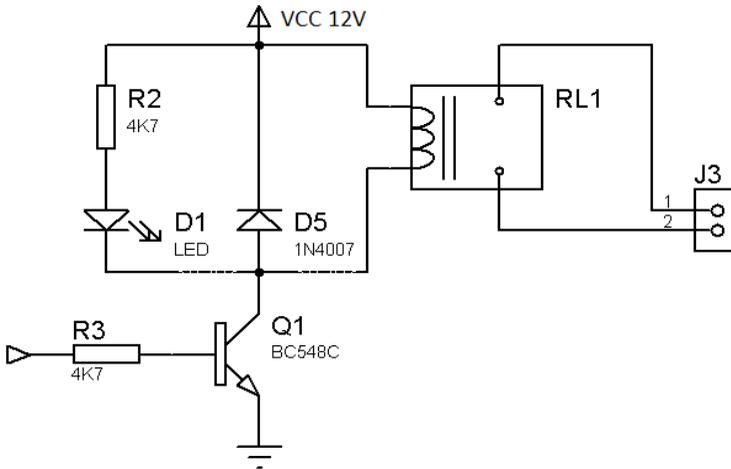


FIGURA 11: Circuito de acionamento dos relés.

3.2.4 Sensoamento

Para a demonstração da capacidade de sensoamento do sistema, foi utilizado um sensor de temperatura/umidade e um sensor de luminosidade.

O sensor de luminosidade empregado é um LDR (*Light Dependent Resistor*), cuja resistência aumenta quando luminosidade é reduzida. Para que a sua variação de resistência seja convertida em tensão, foi utilizado um resistor de 10 KOhms em série com o LDR (ver Figura 12). Dessa forma, a variação de luminosidade pode ser medida pelo conversor analógico/digital do microcontrolador.

A unidade usada para medir a intensidade luminosa é o Lux. Entretanto, neste trabalho foi empregado o valor em percentual calculado da conversão analógica-digital com o circuito, usando leituras de sol intenso para calcular 100% de luminosidade, e o sensor coberto em um ambiente escuro para calcular o valor de 0% , evitando assim, a aferição de escala, o que está fora do escopo deste trabalho. O objetivo

do uso do LDR é apenas mostrar a possibilidade de uso de mais sensores pelo sistema.

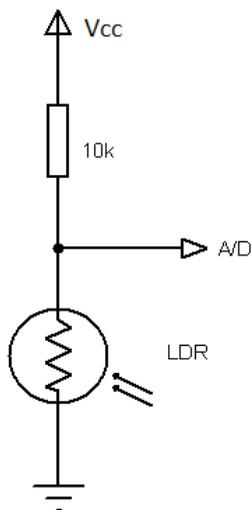


FIGURA 12: Circuito do sensor de luminosidade.

Para a medição de umidade relativa do ar e da temperatura foi usado o sensor DHT11. Esse sensor usa o protocolo 1-wire, medindo temperaturas de 0 a 50 graus Celsius com precisão de ± 2 graus, e a umidade relativa do ar de 20 a 95%, com precisão de $\pm 5\%$.

A medida de umidade relativa do ar é mostrada no dispositivo Android em valor percentual, e significa o quanto de água na forma de vapor existe na atmosfera no momento em relação ao máximo que poderia existir na mesma temperatura. Em outras palavras, pode-se dizer que a umidade relativa do ar é a relação entre a quantidade de água existente no ar (umidade absoluta) e a quantidade máxima que poderia haver na mesma temperatura (ponto de saturação) (CEPAGRI, 2012).

O DHT11 possui 4 pinos, mas somente 3 pinos são usados: os pinos de alimentação (1 e 4), e o pino da comunicação 1-wire (2) (Ver Figura 13). Maiores informações podem ser vista no Anexo D.



FIGURA 13: Módulo sensor DHT11.

O circuito simplificado entre o módulo eLPC48 e os circuitos envolvidos pode ser visto na Figura 14.

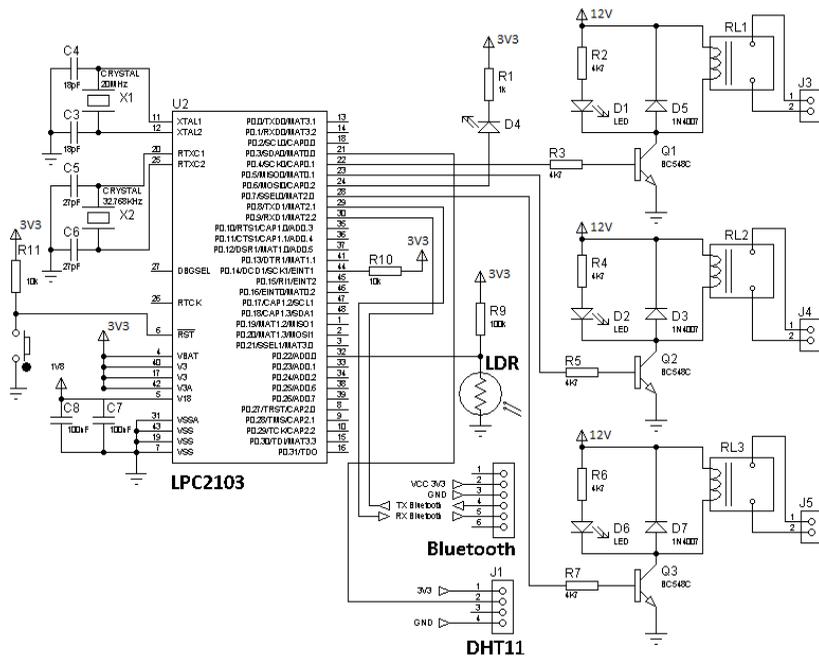


FIGURA 14: Esquemático simplificado da placa de sensores e acionamento.

Foram utilizados barras de pinos para a conexão com o módulo Bluetooth e com o DHT11, o LDR foi ligado no pino 32, (A/D0), o sensor DHT11 ao pino 21, e os relés, ligados aos pinos 22 (P0.4), 23 (P0.5) e 28 (P0.7).

A comunicação serial entre o microcontrolador e o módulo Bluetooth foi configurada para uma taxa de comunicação de 9600 bps, 8 bits de dados, 1 bit de parada. Essa taxa é suficiente para a transmissão dos dados de luminosidade, temperatura, umidade e dos comandos para o acionamento dos relés.

3.3 FERRAMENTA DE DESENVOLVIMENTO PARA O ANDROID

Para o desenvolvimento de programas para a plataforma Android são necessárias algumas ferramentas de programação: é preciso instalar a IDE (*Ambiente Integrado de Desenvolvimento*) do Eclipse e o SDK (*Standard Development Kit*) do Android.

Após a instalação do Eclipse e do SDK é necessária a atualização deste, através do *SDK Manager*, como pode ser visto na Figura 15. A atualização permite a escolha dos pacotes de *software* (bibliotecas de funções) que se deseja utilizar.

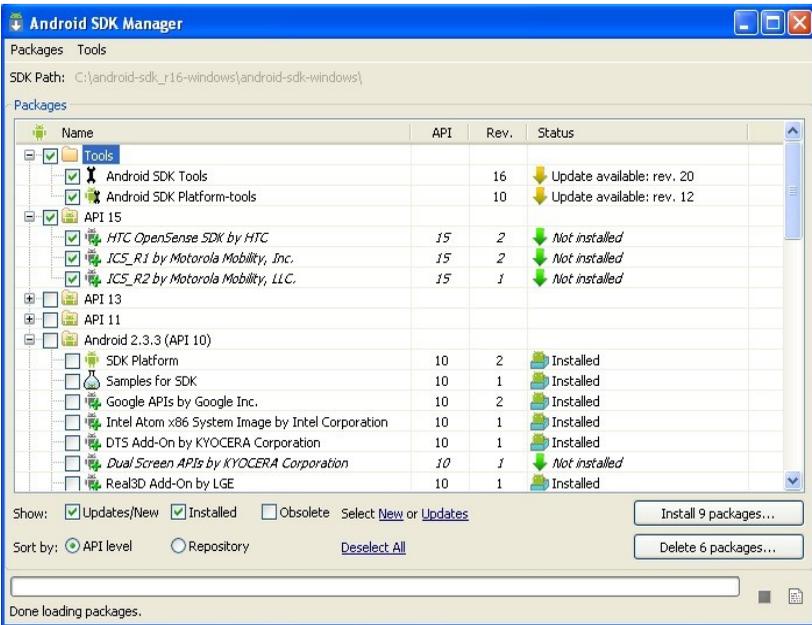


FIGURA 15: Android SDK Manager.

Neste trabalho foram instalados todos os pacotes da API (*Application Programming Interface*) versão 10, que é usada para desenvolver programas em plataformas com a versão do Android 2.3.6 ou superior.

Também através do *SDK Manager* foram instalados o pacote *Android SDK Tools*, que incluem o conjunto completo de ferramentas de desenvolvimento e depuração, e o *Android Platform-tools*, que adiciona as ferramentas da plataforma Android, a fim de apoiar recursos mais recentes.

Para que o Eclipse seja capaz de compilar os programas para Android, deve ser instalado seu Plugin ADT (*Adroid Development Tools*). Da mesma forma que os programas anteriores, essa ferramenta é gratuita e pode ser encontrada na internet (para maiores detalhes, consultar <http://developer.android.com>).

3.3.1 O Emulador Android

Com a instalação das ferramentas de desenvolvimento pode-se criar um emulador de celular para o teste dos programas desenvolvidos. O emulador permite analisar o comportamento do programa sem a dependência do *hardware*, com as características do *hardware* para o qual está sendo desenvolvido o programa. Isso é feito com a ajuda do AVD (*Android Virtual Device*) que é instalado com o SDK Android. Na Figura 16 pode ser visto o AVD sendo executado e as suas janelas de configuração.

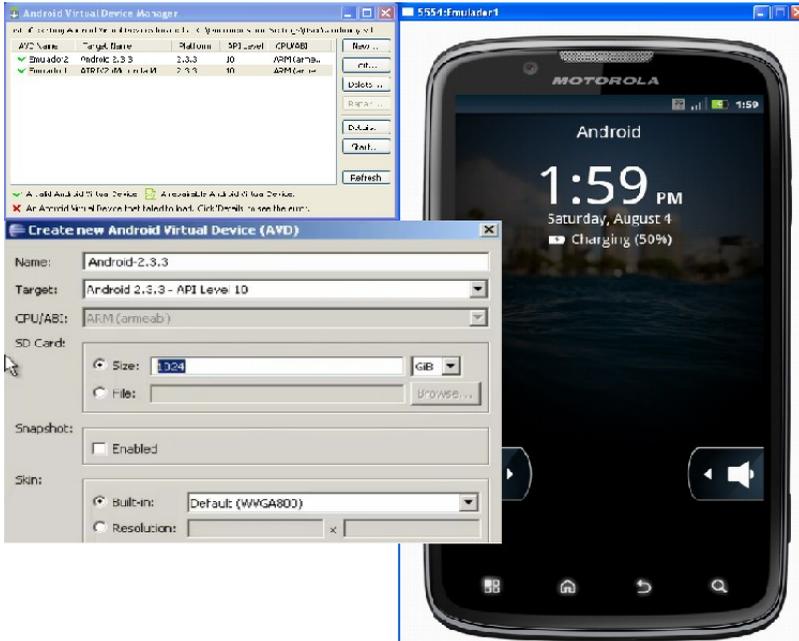


FIGURA 16: O emulador Android para dispositivo virtual.

O emulador Android para dispositivo virtual foi usado neste trabalho apenas para validações relacionadas à interface gráfica, pois não é possível emular aplicações que necessitem do dispositivo Bluetooth.

3.4 O TELEFONE CELULAR (DISPOSITIVO MÓVEL)

Este trabalho foi desenvolvido com o uso de um telefone celular com comunicação Bluetooth e com sistema operacional Android. Assim, este trabalho é compatível com a maioria dos modelos de celulares Android. Foi utilizado o modelo Atrix da Motorola (ver Figura 17), com boa capacidade de memória e processamento, permitindo o

desenvolvimento de aplicativos mais elaborados. Esse celular possui 1 GB de memória RAM, 10 GB de memória interna e processador *dual-core* com velocidade de 1GHz.



FIGURA 17: Celular Atrix, com Sistema Operacional Android.

Fonte: http://www.motorola.com.br/consumers/MOTOROLA-ATRIX/72556,pt_BR,pd.html. Acesso em: ago 2012.

3.5 PROGRAMA DESENVOLVIDO PARA O ANDROID

O programa feito para o celular Android apresenta uma tela com botões que enviam comandos a relés ligados ao *hardware* de acionamento e três campos para os valores de luminosidade, temperatura e umidade, recebidos. Existe também uma área de texto usada para demonstrar a variação de temperatura, mostrando as temperaturas máxima, média e mínima. As leituras dos sensores são atualizadas a cada 3 segundos, e é usado as ultimas 40 leituras para o cálculo de média, ou seja, é considerado dois minutos, assim como máximo e mínimo (ver Figura 18).

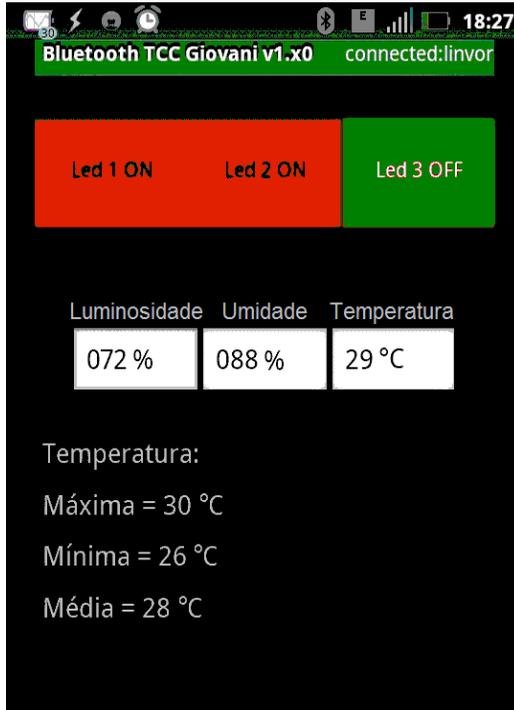


FIGURA 18: Programa desenvolvido para o Android.

3.5.1 Comunicação Bluetooth do celular

Através do Bluetooth do celular, o programa irá procurar por dispositivos ao alcance e apresentar ao usuário a lista de dispositivos encontrados, indicando ao usuário que selecione um dos dispositivos para conexão.

Essa rotina é ativada pelo usuário, através do botão de menu do sistema Android. Após o usuário identificar e selecionar o módulo Bluetooth da placa de sensoramento e acionamento, o sistema pede uma senha para se conectar a ele. No caso do módulo usado, é a senha é

“1234” e o nome é “linvor” (esses parâmetros podem ser alterados na configuração do módulo). Uma vez reconhecida a senha, o chamado pareamento, esse processo não é mais necessário, e a conexão se dá automaticamente ao se selecionar o módulo.

3.6 ALGORITMO E INTEGRAÇÃO DO SISTEMA

Quando o programa desenvolvido para o Android se conecta com o módulo Bluetooth, é enviada uma primeira mensagem para a placa de sensoramento pedindo o estado atual dos relés, ou seja, se estão acionados ou não. Após receber a resposta, o programa do celular indica o estado dos relés mudando a cor dos botões, bem como o texto contido neles, ficando com a cor verde e com o texto “Led ON” para o relé desligado, e com a cor vermelha e texto “Led OFF”, para o relé ligado.

Após a primeira comunicação, a placa de acionamento e sensoramento irá fazer a leitura de seus sensores e enviar os dados sequencialmente, com um intervalo de 3 segundos. Esses dados são identificados pelo celular e mostrados na tela, sendo atualizados a cada nova mensagem.

Com um toque no botão de comando na tela do celular é enviada uma mensagem à placa de controle que faz o devido acionamento e envia uma resposta indicando que o acionamento foi realizado. Dessa forma, o programa no celular pode indicar que a tarefa foi concluído com sucesso.

4 RESULTADOS

O programa desenvolvido para o celular Android ocupa 29 kB de memória, e embora tenha sido desenvolvido com a API 15 para a versão 2.3.6 do Android ou superior, foi testado também em outros dois modelos com a versão 2.1 apresentando perfeito funcionamento. No entanto, em um deles, por apresentar uma tela de menores dimensões, parte das informações ficaram ocultas, mas o fato não afetou o pleno funcionamento do programa.

Em testes efetuados, os comandos enviados por Bluetooth atingiram um alcance de aproximadamente 9 metros em um ambiente interno com obstáculos e cerca de 25 metros em ambientes de campo aberto.

O firmware desenvolvido para o microcontrolador ARM7 ocupou apenas 15 kB de uma capacidade de memória de até 32 kB.

Os custos do protótipo resultaram em um valor aproximado de R\$ 150,00 (U\$ 80,00), que corresponde aos módulos do ARM no valor de R\$ 49,00 e do Bluetooth, de R\$ 57,00; além do sensor DHT11 de R\$ 18,00. O restante do custo é referente aos demais componentes utilizados na montagem do protótipo (placa).

A alimentação da placa é de 12 V e sua corrente é de aproximadamente 75 mA sem acionamento dos relés. Cada um dos 3 relés que compõem a placa, quando acionados, têm corrente de 80 mA, fazendo com que o sistema, durante seu funcionamento pleno, venha a consumir 315 mA. Em caso de falta de energia na rede elétrica, poderia se fazer a alimentação através de uma bateria em um ponto do circuito após o regulador de tensão. Isto se daria com uma tensão de somente 5 V, já que os relés perderiam sua utilidade sem a tensão de operação de 12 V. Entretanto, o restante da placa funcionaria perfeitamente.

Na Figura 19 pode ser visto o hardware do protótipo montado em uma placa universal de 10x10 centímetros. Na lateral, à direita, está o módulo eLPC48 e ao centro o módulo bluetooth. Observam-se, ainda, os

três relés e seus bornes para a ligação de 220 V. Na parte inferior da figura encontra-se o sensor DTH11.

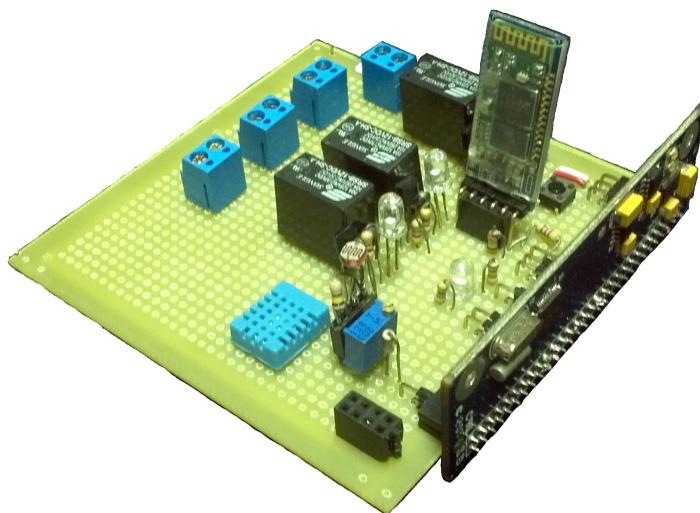


FIGURA 19: Placa de sensoriamento e acionamento.

5 CONCLUSÕES

Os objetivos propostos neste trabalho foram alcançados, dentre os quais destacam-se: o desenvolvimento de um programa para o sistema operacional Android, visando o sensoreamento de grandezas físicas; o controle de variáveis através de um telefone celular, e o desenvolvimento de um protótipo para a conexão de sensores e de relés. A integração entre as partes do projeto foi satisfatória; foram controlados 3 relés e realizada a leitura de temperatura, umidade e luminosidade.

A maior dificuldade encontrada foi o desenvolvimento do aplicativo para o sistema operacional Android. Fato devido ao pouco conhecimento da linguagem de programação Java. O ponto mais difícil foi a implementação das funções de conexão e transmissão para a comunicação Bluetooth.

Uma melhoria que poderia ser feita em um trabalho futuro, seria a previsão do tempo de curto termo (poucas horas adiante) usando os valores fornecidos pelos sensores de temperatura, umidade e adicionando-se, ainda, um sensor de pressão barométrica. Com isso, seria possível prever o momento de uma chuva com alguns minutos de antecedência. Outra possível melhoria seria o aperfeiçoamento da interface gráfica, com a possibilidade do usuário alterar os nomes na tela do celular que descrevem cada acionamento. Dessa forma, as informações seriam apresentadas de forma personalizada.

Em resumo, um protótipo foi desenvolvido validando as ideias de um controle e sensoreamento com um dispositivo móvel. Ainda, é necessário dar uma forma mais amigável ao sistema para que possa ser utilizado em um controle específico de um ambiente residencial ou comercial.

REFERÊNCIAS

CEPAGRI BRASIL. Universidade de Campinas. Centro de Pesquisas Meteorológicas e Climáticas Aplicadas a Agricultura. Disponível em: <<http://www.cpa.unicamp.br/artigos-especiais/umidade-do-ar-saude-no-inverno.html>>. Acesso em: 22 nov. 2012.

DA SILVA, Michel Lourenço. Android Para Desenvolvedores. 1º Edição. Rio de Janeiro, RJ: Editora Brasport, 2009.

GEOFF LAWDAY, “Implementing Bluetooth in an Embedded Environment”, Techonline, Gennaio 2007. Disponível em: <http://techonline.com/article/192200576> Acesso em: 20 de setembro 2012.

MILLER, B; BISDIKIAN, C. Bluetooth Revealed, 1º Edição. Englewood Cliffs, NJ: Prentice Hall, 2000.

SIG (Special Interest Group) Specification of the Bluetooth System 2001.

STALLINGS, William. Wireless Communications and Networking, 1º Edição. New Jersey: Prentice Hall, 2002.

The Bluetooth SIG Standard. Documentação oficial do Bluetooth. Disponível em: <http://www.bluetooth.com/>. Acesso em: 25 de agosto de 2012.

KANSAL, Aman. Bluetooth Primer. Los Angeles: Red-M. 2002.

KOBAYASHI, C. Y. A Tecnologia Bluetooth e Aplicações. Monografia - Universidade de São Paulo. São Paulo. 2004.

ZECHNER, Mario. Beginning Android Games. New York. Spring Verlag. 2011.

ANEXO A – MÓDULO BLUETOOTH



BLUETOOTH SERIAL CONVERTER UART INTERFACE 9600BPS USER'S GUIDE

Chapter1. Bluetooth Serial Converter UART Interface

9600bps

Key Features

- ✧ Bluetooth Spec v2.0+EDR Compliant
- ✧ Enhanced Data Rate (EDR) compliant with V2.0.E.2 of specification for both 2Mbps and 3Mbps modulation modes
- ✧ Class 2 Type Output Power
- ✧ Full Speed Bluetooth Operation with Full Piconet Support
- ✧ Scatternet Support
- ✧ 3.3V Operation
- ✧ Minimum External Components
- ✧ UART Interface
- ✧ Support for 8Mbit External Flash Onboard
- ✧ Support for 802.11Co-Existence
- ✧ RoHS Compliant



Product Description

GL-6B is a Class 2 Bluetooth module using BlueCore4-AudioROM chipset from leading Bluetooth chipset supplier Cambridge Silicon Radio.

This converter is used to convert a UART with 9600bps, one start bit, 8 data bits, one stop bit, no parity bit format to Bluetooth UART protocol. It can not talk to a same device but only to PC Bluetooth dongle. That is, it will create communication between an embedded microcontroller and a PC, but not between 2 embedded devices.

For the password, normally it doesn't need a password, when PC could find the converter, just double click it and the connection will be created. If it requests you to input password all the same, it shall be 0000 or 1234.

Applications

- ✧ Bluetooth Carkit
- ✧ PCs
- ✧ Computer Accessories

Bluetooth Serial Converter UART Interface 9600bps

◇ Access Points

Specifications

Operating Frequency Band	2.4GHz-2.48GHz unlicensed ISM band
Bluetooth Specification	V2.0+EDR
Output Power Class	Class2
Operating Voltage	3.3V
Host Interface	UART
Dimension	26.9mm(L)×13mm(W)×2.2mm(H)

* Specification are subject to change without prior notice

Electronics Characteristics

Absolute Maximum Ratings		
Rating	Min	Max
Storage Temperature	-40°C	+150°C
Supply Voltage	-0.4V	5.6V
Other Terminal Voltage	VSS-0.4V	VDD+0.4V

Recommended Operating Conditions		
Operating Conditions	Min	Max
Operating Temperature Range	-40°C	+150°C
Guaranteed RF Performance Range*	-40°C	+150°C
Supply Voltage	2.2V	4.2V

* Typical figures are given for RF performance between -40°C and +105°C.

Power Consumption

Operation Mode	Connection Type	UART Rate (Kbps)	Average	Unit
Page Scan	-	115.2	0.42	mA
ACL No Traffic	Master	115.2	4.60	mA
ACL with File Transfer	Master	115.2	10.3	mA
ACL 1.28s Sniff	Master	38.4	0.37	mA
ACL 1.28s Sniff	Slave	38.4	0.42	mA
Standby Host Connection	-	38.4	40	µA

* Low power mode on the linear regulator is entered and exited automatically when the chip enters/leaves Deep Sleep mode. For more information about the electrical characteristics of the linear regulator, see section 4 in this document.

ANEXO B – MÓDULO ELPC48

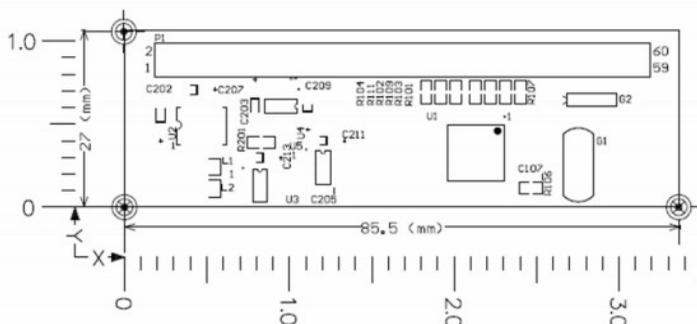


2. CARACTERÍSTICAS TÉCNICAS

- *Header* de 60 pinos (2 linhas de 30 pinos com passo de 0.1") para a conexão com a placa base
- Suporte aos processadores NXP LPC21xx com encapsulamento LQFP48
- Alimentação externa de 5 V +/- 1 V. Reguladores de tensão estão incluídos no SOM.
- Regulador de tensão LDO para o processador e dispositivos externos com capacidade de fornecimento de até 300 mA (100 mA para o processador e 200 mA para circuitos externos).
- Cristais para o processador (até 20 MHz) e RTC (32,768 kHz).
- Alguns processadores possuem pinos de I/O configuráveis como saídas apenas no modo dreno aberto. Nestes casos, a placa base deve prover um resistor de *pull-up* para o correto funcionamento do pino. Consulte o *data sheet* do processador utilizado.
- Compatível com a placa de desenvolvimento eLPC-Main 2122.

2.1 Dimensões mecânicas

O SOM eLPC48 tem dimensões de 27 x 85.5 mm, conforme figura abaixo:



O conector P1 é um header fêmea de 2 x 30 pinos montado na parte inferior da placa. Requer uma barra de pinos de 2 x 30 com passo de 0.1" na placa base para conexão.

A altura total (incluindo o SOM e o conector P1) é de aproximadamente 10 mm.

2.2 Consumo e fornecimento de corrente

O consumo máximo do processador é de 100 mA em 3,3 V. O regulador de tensão LDO utilizado na eLPC48 tem sua capacidade de corrente limitada pela dissipação térmica. Para o tipo de dissipador utilizado, pode operar com correntes de até 300 mA, ou seja, 100 mA para o processador e 200 mA ficam disponíveis para circuitos conectados externamente.

3. PRINCIPAIS COMPONENTES

Os principais componentes da eLPC48 são:

Identificador	Descrição
U1	Processador ARM
G1, G2	Cristais do processador e do RTC
P1	Header de 60 pinos com 0,1" para a conexão com a placa base
U2, U4	Regulador LDO 3V3
U3	Regulador LDO 1V8
L1	Led indicador de reset (ativado em alta intensidade durante o reset; ativado em baixa intensidade durante a operação normal do processador)
L2	Led indicador de alimentação da placa

Os processadores suportados estão listados a seguir. **Importante:** módulos SOM baseados em alguns destes processadores só são produzidos mediante encomenda. Consulte a eSysTech sobre a disponibilidade.

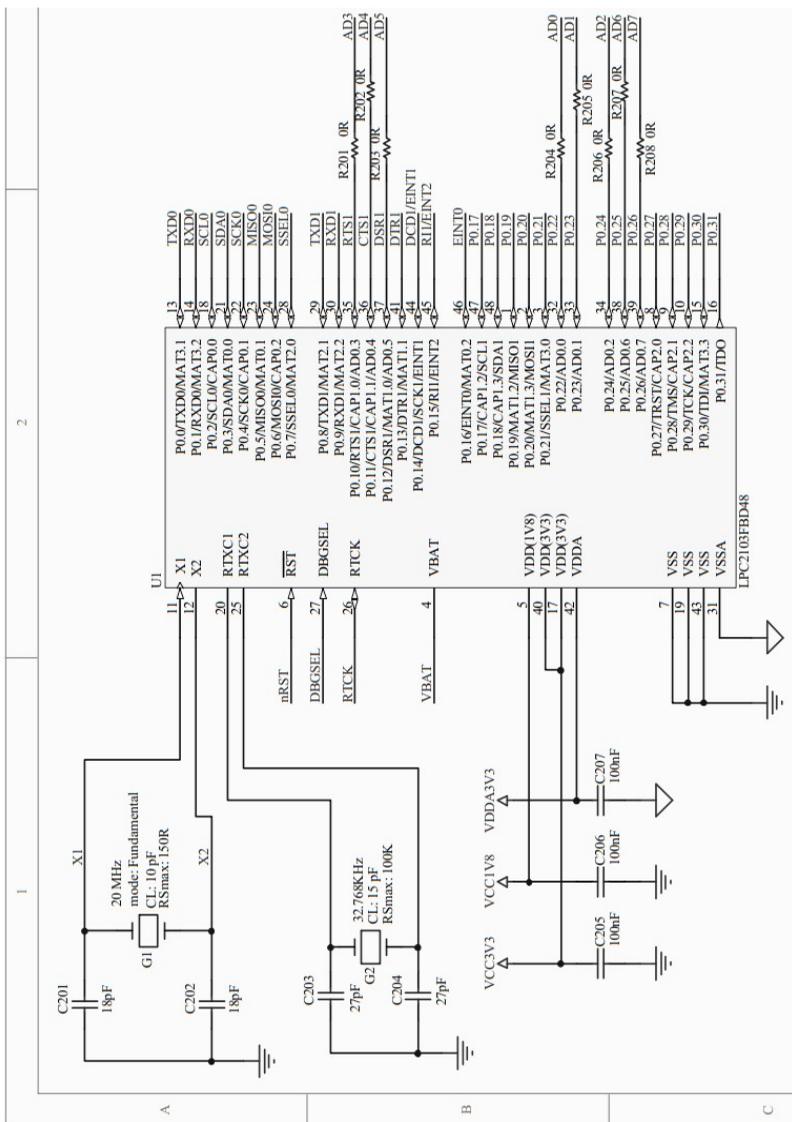
Processador	Memória		Timer/Counter			PWM		I/O		Interface Serial				A/D	Ext. Interrupts	Freq. Max (MHz)
	Flash	RAM	32-bit Timers	Capt.	Match	32-bit Timers	Ch.	Pins	UART	I2C	SPI	CAN	ch/b			
LPC2106	128K	64K	4	7	7	1	6	32	2	1	1	-	-	3	60	
LPC2105	128K	32K	4	7	7	1	6	32	2	1	1	-	-	3	60	
LPC2104	128K	16K	4	7	7	1	6	32	2	1	1	-	-	3	60	
LPC2103	32K	8K	6	8	8	1	14	32	2	2	2	-	8/out	3	70	
LPC2102	16K	4K	6	8	8	1	14	32	2	2	2	-	8/out	3	70	
LPC2101	8K	2K	6	8	8	1	14	32	2	2	2	-	8/out	3	70	

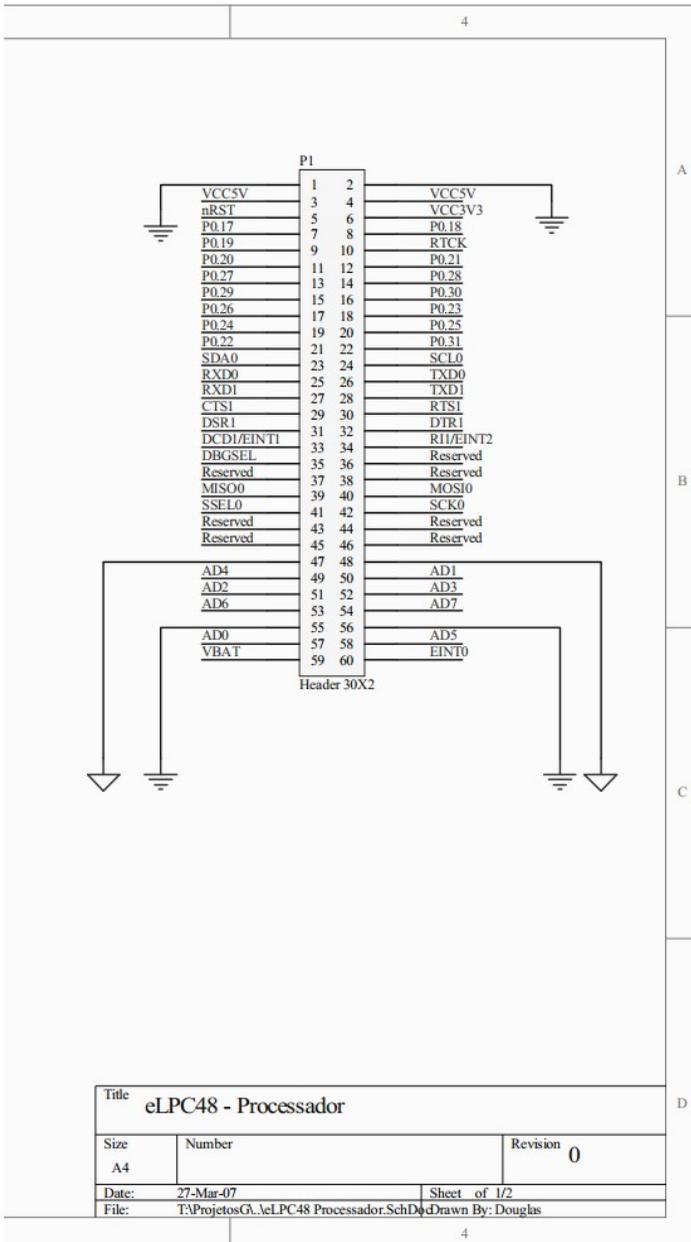
Importante: Nota sobre o RTC:

Para o processador LPC2103 com marcação NXP (existe outra versão com marcação Philips) o consumo do pino VBat é maior que 1 mA. Neste caso a bateria de lítio (CR2032) da placa eLPC2122Main seria consumida em aproximadamente 10 dias.

A tensão em VBat é necessária para o funcionamento do processador. Por conta disso, foram curto-circuitados os sinais VDD1V8 e VBat (pinos 4 e 5 do processador) de modo a garantir a polarização do pino VBat. Nesse caso, para que o horário seja mantido, é necessário manter o SOM alimentado com 5V. O esquemático Rev0p reflete essa alteração.

ANEXO C – ESQUEMÁTICO DO EPLC48





ANEXO D – SENSOR DHT11

Digital-output relative humidity & temperature sensor/module – DHT11



Resistive-type humidity and temperature module/sensor

1. Feature & Application:

- * Full range temperature compensated
- * Relative humidity and temperature measurement
- * Calibrated digital signal
- * Outstanding long-term stability
- * Extra components not needed
- * Long transmission distance
- * Low power consumption
- * 4 pins packaged and fully interchangeable

2. Description:

DHT11 output calibrated digital signal. It utilizes exclusive digital-signal-collecting-technique and humidity sensing technology, assuring its reliability and stability. Its sensing elements is connected with 8-bit single-chip computer.

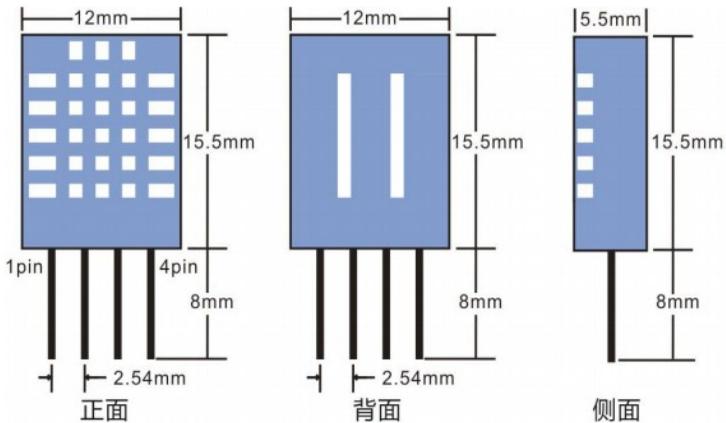
Every sensor of this model is temperature compensated and calibrated in accurate calibration chamber and the calibration-coefficient is saved in OTP memory.

Small size & low consumption & long transmission distance(20m) enable DHT11 to be suited in all kinds of harsh application occasions. Single-row packaged with four pins, making the connection very convenient.

3. Technical Specification:

Model	DHT11		
Power supply	3-5.5V DC		
Output signal	digital signal via single-bus		
Sensing element	Polymer resistor		
Measuring range	humidity 20-90%RH; temperature 0-50 Celsius		
Accuracy	humidity +4%RH (Max +-5%RH); temperature +-2.0Celsius		
Resolution sensitivity	or	humidity 1%RH;	temperature 0.1Celsius
Repeatability		humidity +-1%RH;	temperature +-1Celsius
Humidity hysteresis	+-1%RH		
Long-term Stability	+-0.5%RH/year		
Sensing period	Average: 2s		
Interchangeability	fully interchangeable		
Dimensions	size 12*15.5*5.5mm		

4. Dimensions: (unit----mm)

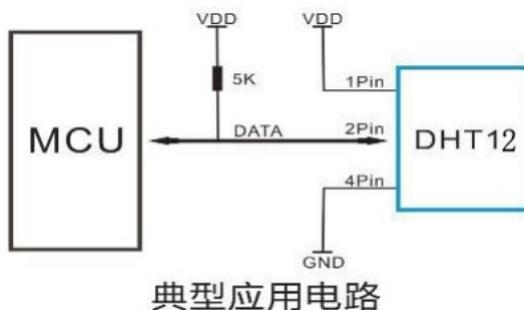


Front view

Back view

Side view

5. Typical application



3Pin=NULL, MCU=Microcomputer or single-chip computer

6. Operating specifications:

(1) Power and Pins

Power's voltage should be 3-5.5V DC. When power is supplied to sensor, don't send any instruction to the sensor within one second to pass unstable status. One capacitor valued 100nF can be added between VDD and GND for power filtering.

(2) Communication and signal

Single-bus data is used for communication between MCU and DHT11.

7. Electrical Characteristics:

Item	Condition	Min	Typical	Max	Unit
Power supply	DC	3	5	5.5	V
Current supply	Measuring	0.5		2.5	mA
	Stand-by	100	Null	150	uA
	Average	0.2	Null	1	mA