

DROIDCONTROLE: SISTEMA DE AUTOMAÇÃO WIRELESS VIA BLUETOOTH USANDO A PLATAFORMA ANDROID

Francisco de Assis de Freitas Gomes¹ Alex Torquato Carneiro¹

Universidade Ibirapuera
Av. Interlagos, 1329 – São Paulo – SP
chicoassis2000@ig.com.br

Resumo

Este artigo descreve a arquitetura de um sistema de Automação o que tem como foco principal o acionamento de cargas elétricas e o monitoramento de sensores de temperatura e infravermelho, usando um Tablet como Interface de Usuário. O ambiente de desenvolvimento usa tecnologias como Programação o com a Plataforma Android, Linguagens de Programação JAVA e C e um micro controlador da família PIC, que fará parte do hardware periférico, o qual é controlado via Bluetooth.

Palavras-chave: Android, Sensores, Automação, Microcontrolador, Programação, IDE.

Abstract

This paper aims to describe the architecture of an automation system mainly designed to drive electric charges and monitor temperature and infrared sensors, using a Tablet as user interface. The development environment uses technologies like programming for the Android platform, JAVA and C Programming Language besides a PIC microcontroller, which is part of the peripheral hardware, controlled via Bluetooth.

Keywords: Android, sensors, automation, Programming.

1. Introdução

A Automação está presente no cotidiano do ser humano e traz grandes benefícios aos lares, indústrias, comércio e meio ambiente e representa um conjunto de tecnologias aplicadas a um determinado contexto, visando aumentar a produtividade, a qualidade e o conforto nos processos que envolvem situações que englobem determinadas áreas, tais como Automação Industrial, Residencial ou Comercial. Alguns exemplos que podem ser citados são: abertura e fechamento automático de portão, monitoramento de temperatura em ambientes controlados, preenchimento automático de cheques e monitoramento de gases poluentes. Esses são alguns exemplos que fazem parte do dia-a-dia do ser humano e que trazem segurança, comodidade, conforto e qualidade de vida.

Diante de todo o cenário descrito, o presente artigo aborda e demonstra conceitos tecnológicos empregados na materialização de um projeto que tem a interatividade como ponto fundamental em sua estrutura. Trata-se de um sistema que tem como finalidade fazer o acionamento de cargas elétricas, representadas por lâmpadas incandescentes e fazer a monitoração da atuação de um sensor de temperatura e de um sensor infravermelho. A interatividade é realizada através de comunicação wireless, via Bluetooth, onde um Tablet que utiliza a plataforma Android comunica-se com um hardware periférico, o qual terá como componente principal um microcontrolador da família PIC (PIC18F45K22).

A este hardware estão conectados os sensores e as cargas descritas anteriormente, os quais são acionados e monitorados. Uma parte do sistema é desenvolvida no ambiente Android, onde a base é a linguagem de programação JAVA em conjunto com a linguagem de formatação XML (eXtensible Markup Language), que é a responsável pela geração das telas que são apresentadas no Tablet e a partir destas telas o usuário tem controle de todo o sistema, acionando as lâmpadas incandescentes e lendo os alarmes gerados pelos sensores, tudo através da tela touchscreen do Tablet. Um painel elétrico montado com as lâmpadas e os sensores exibe o resultado da interação existente entre o usuário e o hardware periférico. Espera-se que este resultado esteja em conformidade com todos os requisitos apresentados pelo sistema.

Este artigo está dividido na seguinte ordem:

Seção 2 – Mostra conceitos teóricos das tecnologias envolvidas no projeto, Seção 3 – Apresenta as especificações e técnicas das tecnologias (software) e componentes (hardware) usados no projeto, Seção 4 – Mostra a arquitetura e funcionamento geral do projeto, Seção 5 – Mostra através de um painel elétrico, os resultados obtidos e Seção 6 – A conclusão do projeto é mostrada.

2. Materiais e Conceitos teóricos

Este artigo agrega tecnologias ligadas a área de programação, comunicação wireless e a componentes eletrônicos, todos interagindo para gerar como resultado um sistema de automação, o qual tem um Tablet como transmissor e receptor de comandos para um hardware periférico onde este aciona cargas elétricas e envia respostas lidas de sensores, de volta para o Tablet. Nesta seção, o conceito teórico de todas as partes que compõem em o projeto é descrito.

2.1. Sistema Operacional para Dispositivos Móveis

Dispositivos Móveis são produtos eletrônicos que integram diversas funcionalidades, tais como acesso a Internet, Multimídia (áudio, vídeo, imagem), armazenamento de arquivos, GPS, Redes Sociais, acesso a ambientes em Nuvem, Bluetooth, mensagens de texto, funções de telefonia, etc. Tudo isto integrado em apenas um dispositivo.

Para que todas essas funcionalidades sejam gerenciadas existem Sistemas Operacionais Móveis responsáveis pela funcionalidade do dispositivo. Assim como os PCs usam sistemas operacionais para gerenciar funções e recursos disponíveis, os dispositivos móveis também usam sistemas operacionais desenvolvidos especificamente para seu gerenciamento [Lee 2011].

2.2. Linguagens de Programação

Em um computador são executados diversos tipos de processamentos, tais como cálculos matemáticos, consultas e atualizações em entidades de Banco de Dados, gerenciamento de sistemas de cadastro, entre tantos outros tipos de processamento. Dentro deste contexto, existem as Lin-

guagens de Programação, que representam regras sintáticas e semânticas dispostas em um texto contendo instruções que serão processadas pelo computador. O referido texto é chamado de código fonte e é ele o responsável pela execução de todos os processamentos citados anteriormente. Existem vários tipos de Linguagens de Programação, cada uma delas voltada para um determinado tipo de ambiente (WEB, Desktop, Plataformas Móveis, etc). As linguagens de Programação podem ser classificadas em duas formas: Alto Nível, onde o código fonte é desenvolvido para o entendimento humano e passa por uma etapa de compilação, para que seja processada pelo computador e Baixo Nível, onde o código fonte é compilado, gerando um resultado que é processado em nível de máquina [Schildt 2010].

2.3. Ambientes de Desenvolvimento Integrado (IDE)

Quando se desenvolve um sistema usando uma Linguagem de Programação, por uma questão de produtividade é necessário que se trabalhe em um ambiente que agrupe todos os recursos necessários para que se produza o projeto de forma rápida e eficaz. As IDEs, Integrated Development Environment, ou Ambiente Integrado de Desenvolvimento, são ferramentas que proporcionam para o desenvolvedor vários tipos de recursos, que estão integrados de tal forma a facilitar todo o processo de programação. Recursos como Editor de Código, Compilador, Bibliotecas, Plugins, Suporte a várias Linguagens de Programação, Modelagem de Dados, Depuração, entre tantos outros, fazem parte da IDE e são necessários para que se aplique uma técnica chamada RAD, Rapid Application Development ou Desenvolvimento Rápido de Aplicativos, que tem como foco principal dar maior produtividade ao desenvolvedor [Alves 2006].

2.4. Microcontrolador

O Microcontrolador é um circuito integrado eletrônico programável, que possui internamente hardware necessário para processamento de dados de entrada e de saída, os quais são responsáveis por acionamento de circuitos elétricos e eletrônicos externos, bem como o monitoramento de informações de entrada, como as provenientes de sensores de temperatura, de proximidade, de humidade, etc. O pro-

cesso de aumento na capacidade de integração de componentes possibilitou aos fabricantes de Microcontroladores inserirem uma quantidade imensa de diferentes hardwares no mesmo, possibilitando todos os recursos necessários ao seu funcionamento, tais como memórias flash, interface de comunicação serial (RS232), conversores analógicos digitais, temporizadores, dispositivos de entrada/saída, contadores, geradores de PWM e registradores de uso geral. O campo de aplicação dos microcontroladores é bastante extenso, indo desde simples acionamentos de circuitos eletrônicos, como o LED (Diodo Emissor de Luz), por exemplo, até sistemas de controle residencial, controle comercial e controle industrial [Nicolosi 2005] [MICROCHIP 2012].

2.5. Sensor

É um dispositivo de natureza elétrica ou eletrônica que ao receber um sinal externo responde com um sinal que ele próprio gera e que serve, ao ser captado pelo circuito de controle, para monitoramento de uma determinada situação. Os sensores são classificados de acordo com o tipo de sinal que recebem. Como exemplos, podem-se citar sensores de temperatura, que ao captar um aumento ou uma diminuição nesta grandeza, gera sinais de tensão elétrica de acordo com o valor da temperatura e, sensores infravermelhos, que ao detectar a presença de algum corpo que interrompa a radiação gera um nível de tensão que aciona um determinado hardware, como por exemplo, contagem de objetos em uma esteira industrial. Ou seja, um sensor faz a medição de um determinado sinal (grandeza física) e gera níveis de tensão elétrica analógica, no caso do sensor de temperatura e níveis de tensão elétrica digital (alta ou baixa, que podem ser representadas por 5VDC ou 0VDC, respectivamente). Os sensores possuem diversas características, as quais definem seu comportamento. São elas: Sensibilidade, Linearidade, Resposta em Frequência, etc. Um sensor analógico é conectado aos pinos Conversores A/D do microcontrolador, ao passo que um sensor digital é conectado aos pinos de I/O, digitais, do microcontrolador [Boylestad 1999].

2.6. Atuador

Representa um dispositivo eletrônico que, quando energizado, produz internamente um movimento mecânico através de seus contatos. Desta forma um atuador converte energia elétrica em energia mecânica. São várias as aplicações dos atuadores na indústria. Uma delas é quando existe a necessidade da comutação de um ponto ao outro.

Isto acontece assim que ocorre a energização elétrica do atuador. O acionamento de uma máquina é um exemplo bem claro, ocorrendo a energização da mesma assim que o atuador for, também, energizado. Ou seja, um atuador em determinadas condições dentro de um projeto elétrico é considerado uma chave comutadora eletromecânica [Capuano 1988] [Markus 2004].

2.7. Antena de RF – Rádio Frequência

A RF, ou Rádio Frequência, significa uma determinada faixa de frequência que engloba os valores entre 3KHz a 300GHz, definindo então os valores compreendidos para a frequência das ondas de rádio. Um grande diferencial em relação as correntes contínua e alternada, que operam em baixas frequências, é que as ondas de rádio ionizam o ar, formando um meio físico não guiado, responsável pelo transporte das informações. As ondas de RF possuem características de frequência e comprimento de onda associados a ela. As ondas de RF precisam de um tipo dispositivo capaz de irradiá-las de um determinado ponto a outro. Neste outro ponto este mesmo tipo dispositivo é capaz de fazer a recepção das ondas de RF. Tal dispositivo é chamado Antena de RF e está conectada a um dispositivo transmissor, responsável pelo envio das informações e um dispositivo receptor, que capta as informações vindas do transmissor [Luiz 2009].

2.8. Dispositivos Móveis

No momento em que houve realmente a popularização da Internet, grandes avanços tecnológicos começaram a ser desenvolvidos e, com isto, grandes mudanças comportamentais do ser humano também aconteceram por conta deste desenvolvimento tecnológico. Estas mudanças estão associadas a forma de agir, de pensar, de se comunicar, de re-

alizar trabalhos que antes eram feitos de uma maneira não tão eficaz e também estão associadas a tantos outros fatores que mudaram consideravelmente atitudes e comportamentos, de uma forma geral. A partir deste nível de evolução, surge uma nova tecnologia de comunicação sem fio, chamada Wireless, que garante vários tipos de comunicação onde a ausência de cabeaço é o foco principal.

Com isto, foram desenvolvidos vários tipos de dispositivos móveis para interagir com esta nova tecnologia. Tais dispositivos como smartphones, tablets, entre outros, integram funcionalidades de telefonia, multimídia e localização (GPS) que permitem o seu uso dentro de um contexto de mobilidade.

3. Especificações Técnicas

Nesta seção são descritas as características elétricas dos componentes eletrônicos, bem como as versões e últimas atualizações das linguagens de programação JAVA e C e de suas respectivas IDEs utilizadas no desenvolvimento do sistema. As especificações técnicas são fundamentais para o estabelecimento de critérios aplicáveis ao funcionamento do projeto, pois elas mostram as características dos componentes, tanto de hardware quanto de software, necessárias para a construção do projeto.

3.1. Plataforma Android

O Android é um software de código aberto (open-source) criado pela Google, para ser utilizado em dispositivos móveis, como smartphones e tablets, por exemplo.

É um Sistema Operacional baseado no kernel Linux e foi criado, originalmente, pela Android Inc. até ser adquirido pela Google, em 2005. Totalmente focado para dispositivos móveis, o Android recebeu a adesão da OHA (Open Handset Alliance), que é um consórcio de empresas que se dedicam ao desenvolvimento de códigos abertos para dispositivos móveis e, além disso, oferecem o suporte necessário para manter o sucesso desta plataforma. Este Sistema Operacional teve início com a versão 1.5 (Cupcake), passando pelas versões 2.0, 2.2, 3.0, 4.0 e 4.1, esta última é a versão mais atual. Cada uma destas versões faz referência a sobremesas (Ice Cream Sandwich, Froio, Ginger Bread, Jellybeans).

Ao se criar uma aplicação para o Android, usando-se a linguagem de programação JAVA, é criada uma estrutura de pastas específicas para cada ação a ser desenvolvida.

O layout do aplicativo, em XML (eXtended Marked Language), fica alocado na pasta Layout, todas as mensagens utilizadas ficam alocadas na pasta String, os desenhos utilizados ficam alocados na pasta Drawable e assim por diante. Esta estrutura se estende com a criação de várias outras pastas, dedicadas ao armazenamento dos arquivos de programa e de referência, necessários para o desenvolvimento do aplicativo [Lee 2011].

3.2. Linguagem de Programação JAVA

Esta linguagem é estruturada com base no paradigma de Programação Orientada a Objetos (POO) e foi desenvolvida pela empresa Sun Microsystems, na década de 90.

Em contraposto a algumas linguagens de programação, JAVA é compilada e gera Bytecodes, executados por uma máquina virtual chamada JVM (JAVA Virtual Machine), que representa um sistema que carrega e executa aplicativos em JAVA, onde realiza a conversão desses Bytecodes e estes podem ser executados em qualquer ambiente de software (WINDOWS, LINUX, MAC OS) que tenham esta máquina virtual instalada.

Para que aplicações desenvolvidas em JAVA sejam executadas é necessário configurar um ambiente em que os mesmos possam ser executados. Para isto, existe a necessidade do JRE (JAVA Runtime Environment), que nada mais é que uma configuração formada pela máquina virtual e por bibliotecas e APIs necessárias a execução do aplicativo.

Esta configuração é imprescindível no computador para que aplicativos como jogos online, páginas da internet, visualizadores de imagens, entre outros, funcionem de forma adequada [Mattos 2007].

3.3. IDE Netbeans

O Netbeans é a IDE utilizada para desenvolvimento do projeto mostrado neste artigo.

É um sistema de código aberto e serve para ajudar os desenvolvedores na criação de projetos que usem diferentes tipos de tecnologias.

Neste projeto a tecnologia utilizada para desenvolvimento do lado do Servidor (Tablet) será JAVA. Com o Netbeans pode-se criar modelos de sistemas baseados em criação de design e posicionamento de componentes, para a montagem não só do layout principal, mas também dos demais layouts que fazem parte do sistema.

É um ambiente de desenvolvimento integrado que é utilizado em qualquer Sistema Operacional (LINUX, WINDOWS, MAC, SOLARIS) e possui desenvolvedores de todas as partes do mundo colaborando e documentando todas as modificações necessárias para que as melhorias nesta IDE estejam sempre ao alcance de outros desenvolvedores. Esta IDE proporciona suporte para a criação de sistemas que utilizam interfaces gráficas e também para criação de aplicações voltadas ao ambiente WEB e ao ambiente de Tablets e Smartphones. Os dados referentes a versão utilizada para o projeto mostrado neste artigo são mostrados a seguir. Versão do Produto: NetBeans IDE 7.0 (Build 201107282000); Versão do Produto: Netbeans IDE 7.0 (Build 201107282000); JAVA 1.7.0.05 ; Java HotSpot 64 Bit Server VM 23.1 b03; Sistema Windows 7 versão 6.1 executando em amd64 Cp1252.

3.4. Linguagem de Programação C

Esta linguagem foi desenvolvida entre os anos de 1969 e 1973, nos laboratórios da AT&T Bell. Foi criada, originalmente, para o implementar o desenvolvimento do sistema operacional UNIX. Este sistema foi o primeiro a ser implementado em uma linguagem de programação que não fosse o Assembly. Em 1978, os criadores da Linguagem C, Dennis Ritchie e Brian Kernighan publicaram o livro The C Programming Language, que durante algum tempo, serviu como especificação, não formal, para esta linguagem. Durante a década de 70, o C substituiu o Basic, que era a linguagem mais utilizada para programação de computadores. Ao longo do tempo, as implementações realizadas nesta linguagem foram aproximando-a um pouco mais dos recursos de hardware do computador, pois fornecia acesso a memória, através de estruturas de Ponteiros e acesso de baixo nível através da inclusão de código Assembly dentro de programas escritos em C [Schildt 2010].

Este artigo mostra um projeto de automação, onde existe a comunicação entre um Tablet e um hardware exter-

no, e este tem como seu principal componente o Micro-controlador PIC18F45K22, programado em Linguagem C. Este programa carregado no microcontrolador é responsável por toda a troca de informações com o Tablet e realizará todos os processos de acionamento de cargas elétricas, quanto aquisição de dados dos sensores de temperatura e infravermelho.

3.5. IDE Mikro C Compiler

O Compilador MikroC for PIC é responsável por todo o desenvolvimento e gravação do programa Cliente no microcontrolador usado neste projeto: PIC18F45K22. Possui uma IDE bastante intuitiva, rápida em seus processamentos para geração do código a ser gravado no microcontrolador e bastante rica em recursos oferecidos aos desenvolvedores de programas em Linguagem C, para este tipo de componente eletrônico. Micro-controladores com barramentos de 8 e 16 bits podem ser programados através desta IDE e a mesma possui uma série de bibliotecas disponíveis para uso no desenvolvimento de 6 projetos das mais diversas áreas, como programação wireless (wi-fi e bluetooth) e USB, por exemplo.

Este compilador é disponibilizado pela empresa Mikroelektronika, responsável pela distribuição deste software em suas versões para 8 e 16 bits, dependendo do microcontrolador da família PIC a ser utilizado. O site oficial da empresa é <http://www.mikroe.com/>.

Provê diversos benefícios ao desenvolvedor, principalmente no tocante a facilidade de elaboração do programa e no desempenho do código gerado, pois consegue gerar o código compilado com tamanho reduzido. Devido a existência de diversas bibliotecas integradas, facilita a rapidez de desenvolvimento. Tais bibliotecas agregam funções matemáticas, funções para manipulação de displays gráficos, funções de comunicação serial, etc. A versão 4.6 deste compilador foi usada no desenvolvimento do programa gravado no PIC-18F45K22 [MIKROELEKTRONIKA 2012].

3.6. Microcontrolador PIC18F45K22

É um componente eletrônico com características bem definidas quando se trata de projetos que envolvam automação industrial ou comercial. Possui áreas de memória que armazenam os programas desenvolvidos, após o processo de compilação e grava-

ção. Existem vários hardwares integrados a este componente, para que o mesmo execute funções de acordo com a sua programação, que pode ser feita em Assembly ou uma linguagem de alto nível, como a Linguagem C. Alguns destes hardwares estão descritos abaixo: Timers, Conversores A/D (Analogico/Digital), Watch Dog, Portas de I/O, Memória de Programa, Comunicação Serial, PWM, entre outros. Um fato interessante neste componente é a sua arquitetura RISC (Reduced Instruction Set Computer), a qual, como diz o próprio nome, utiliza um conjunto de instruções reduzido e trabalha com dois barramentos independentes para o tratamento de dados e programa, diferentemente da arquitetura usada na Máquina de Von Newman, que processava todas as informações (escrita e leitura) em apenas um barramento. Percebe-se que a arquitetura RISC é mais rápida.

Algumas de suas especificações técnicas importantes são mostradas adiante.

É importante ressaltar que todas as características deste componente devem ser rigorosamente obedecidas, durante a fase de desenvolvimento do hardware, pois o correto funcionamento deste componente depende disto. Os dados a seguir foram extraídos do datasheet do PIC18F45K22, o qual pode ser acessado através do site do fabricante deste componente, para as devidas orientações para desenvolvimento do hardware (ww1.microchip.com/downloads/em/DeviceDoc/41412F.pdf). Vide a seguir: - Resolução do Conversor A/D de 10 bits, Circuito Oscilador Interno de 16MHz, tensão de operação entre 2.3V a 5.5V, suporte a comunicação RS232, memória Flash de 32KB, até 35 pinos de I/O, encapsulamento 40-Pin PDIP, 14 canais A/D (AN0...AN13), três Controladores de Interrupção externa (IE0, IE1, IE2), etc [MICROCHIP 2012].

3.7. Sensor de Temperatura LM35

Fabricado pela National Semiconductor, uma das grandes características deste sensor de temperatura é a sua precisão em fornecer valores de temperatura, em graus Celsius, através de seu pino de saída. Faz parte do hardware periférico deste projeto, onde atua na captação de temperaturas e as exibe no Tablet.

Este componente não necessita de qualquer tipo de calibração para gerar valores de temperatura de forma bastante precisa.

Em sua forma mais utilizada em projetos encontra-se no encapsulamento plástico do tipo TO-92, conforme mos-

trado nas figura 1b (<http://singularidadegenerica.blogspot.com.br/2011/05/sistema-avancado-de-controle.html> (Último acesso: 17/11/2012) e na figura 1a, o desenho de sua pinagem (http://arduino-ce.blogspot.com.br/2011/01/01_archive.html, Último acesso: 17/11/2012). O site do fabricante do Sensor LM35 (www.national.com) mostra o datasheet deste componente. Algumas especificações técnicas foram extraídas deste datasheet e são mostradas abaixo:

Faixa de operação: entre -55 graus Celsius e 150 graus Celsius, corrente elétrica de dreno menor que $60\mu\text{A}$, fornece 10mV/ grau Celsius em sua saída (V_{out}), tensão de operação entre 4V e 30V (neste projeto é 5V , etc).

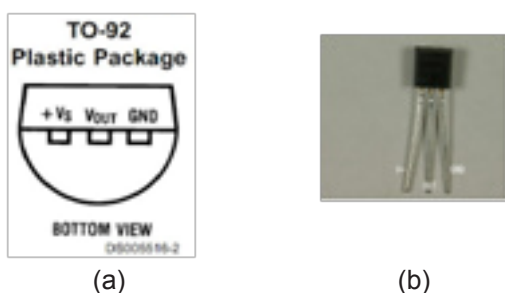


Figura 1. Sensor de Temperatura LM35

3.8. Sensor Infravermelho

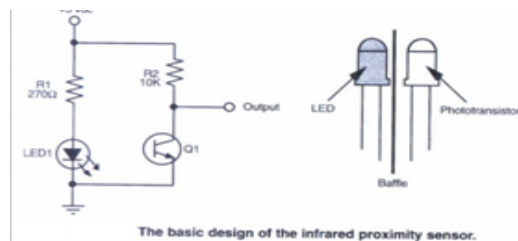
Estes sensores fazem parte do hardware periférico e têm como finalidade sinalizar, através do Tablet, caso alguma interrupção no feixe infravermelho ocorra.

O circuito elétrico mostrado na figura 2a, extraída do link <http://www.eletronica.com/sensor-de-proximidade-bem-simples/> (Último acesso: 17/11/2012), tem o sensor TIL32 (LED) transmitindo luz ininterruptamente, enquanto o TIL78 (Q1) representa o receptor, captando a luz que está sendo emitida.

Cada sensor tem suas características próprias. A figura 2b foi extraída do link a seguir: <http://newportcom.com.br/catalogsearch/result/index/?dir=asc&mode=list&order=relevance&q=diodo> (Último acesso: 17/11/2012).



Figura 2. Sensor Infravermelho



(a)

Figura 2. Sensor Infravermelho

A seguir, algumas especificações técnicas para os sensores TIL32 e TIL78. O ponto "Output" da figura acima deverá ser conectado a um dos pinos configurados como saída digital, do microcontrolador. Especificações:- TIL32: Fabricante: Texas Instruments (www.ti.com/), potência de saída = 0.5mW , corrente elétrica de operação = 20mA , encapsulamento TO-18.

- TIL78: Fabricante: Texas Instruments (www.ti.com/), tensão de operação = 5V , corrente elétrica quando iluminado = 1mA , corrente elétrica sem iluminação = 25nA , encapsulamento TO-18.

Vale ressaltar que o circuito do sensor Infravermelho é representado pelo Diodo emissor de luz Infravermelho (TIL32) e pelo fototransistor (TIL78), responsável pela captação das irradiações transmitidas pelo Diodo.

3.9. Relé Metaltex ML2R C-5V

Este projeto apresenta uma etapa de acionamento de cargas elétricas, representadas por lâmpadas incandescentes, onde o acionamento das mesmas é realizado através de contatos de relés eletromecânicos. O papel principal deste componente é servir como atuador, pois vai acionar cargas que consomem um valor um pouco mais alto de corrente elétrica, valor este que não pode ser suprido pelo microcontrolador. O relé usado no projeto é fabricado pela empresa Metaltex (site: <http://www.metaltex.com.br/index.asp>) e possui características mecânicas e elétricas ideais para sua utilização.

A figura 3 (<http://www.metaltex.com.br/downloads/ML.pdf>) mostra o formato físico deste componente. Algumas de suas principais especificações técnicas, extraídas do catálogo geral da Metaltex, são mostradas a seguir:

- Tensão Nominal de operação = 5VCC, corrente elétrica de comutação (máxima) = 2A, máxima tensão contínua = 10VCC, tensão elétrica de operação (bobina) = 3.75VCC, tensão elétrica de desoperação (bobina) = 0.5VCC, consumo nominal (bobina) = 200mW, resistência da bobina = 125 Ohms.

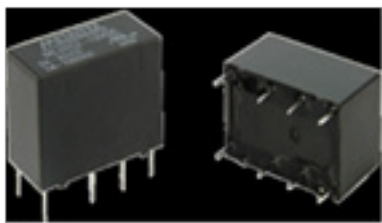


Figura 3. Relé MLR2 C-5V

3.10. Módulo EasyBluetooth

Este módulo consiste de um hardware composto por alguns componentes eletrônicos (resistores, capacitores e transistores) e tem como componente principal uma antena transmissora/receptora.

Este módulo foi desenvolvido pela empresa Mikroelektronika (Site: www.mikroe.com/) e a mesma disponibiliza o manual com as especificações técnicas deste módulo através do link <http://www.mikroe.com/add-on-boards/communication/> (Último acesso: 18/11/2012).

A figura 4, extraída do link anterior, mostra a estrutura física do módulo EasyBluetooth.

Este módulo está preparado para ser conectado as linhas de comunicação serial RS232 do microcontrolador, encaminhando os dados que capta, via bluetooth para estas linhas seriais (tx/rx).

É conectado ao Port C do microcontrolador PIC-18F45K22 e a tensão de alimentação para funcionamento do módulo pode ser configurada em 3.3V ou 5V, através de straps. É um módulo Bluetooth Classe 2 (alcance de 10m, potência máxima permitida 2.5 mW, 4 dBm).

Configurações adicionais devem ser feitas no Módulo EasyBluetooth, para que o mesmo funcione adequadamente. Uma chave principal composta por várias micro-chaves deve estar configurada conforme o manual deste módulo [MIKROELEKTRONIKA 2012].

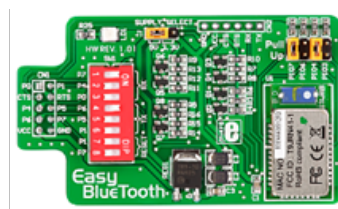


Figura 4. Módulo EasyBluetooth

3.11. API de Comunicação Bluetooth

A transmissão de dados a curta distância é uma das características do Bluetooth. Esta tecnologia é uma alternativa quando se pretende transmitir e receber dados a distâncias que estão no intervalo de até 10m (Classe 2), normalmente. O estabelecimento de uma comunicação Bluetooth envolve uma série de configurações necessárias para que isto ocorra e a API Bluetooth é a responsável por tudo isto.

Ela encontra-se no pacote `android.bluetooth` e tem neste pacote suas principais classes, que são: `BluetoothAdapter`, `BluetoothDevice` e `BluetoothSocket`.

A classe `BluetoothAdapter` é responsável por todo o processo de conexão e localização de aparelhos que estejam no raio de alcance. A classe `BluetoothDevice` representa o dispositivo Bluetooth e é através dela que obtemos o tipo do dispositivo (Smartphone ou Tablet ou Fone de Ouvido), o nome amigável do aparelho, o MAC Address, etc. E, por fim, a classe `BluetoothSocket`, que aguarda que uma conexão seja estabelecida e a partir deste momento uma instância desta classe é criada, retornando os objetos `InputStream` e `OutputStream`, responsáveis pela leitura e escrita de informações, respectivamente, pelo canal formado durante o estabelecimento desta conexão.

Dentro de uma aplicação, existem permissões que são definidas para que as mesmas possam utilizar os recursos de Bluetooth. Estas permissões são declaradas em um arquivo chamado `AndroidManifest.xml` e possuem o seguinte formato: `android.permission`.

`BLUETOOTH` e `android.permission.BLUETOOTH_ADMIN`. Para que a aplicação seja conectada ao hardware externo se faz necessário que tanto o lado Cliente quanto o lado Servidor estejam devidamente configurados, pois cada lado terá um comportamento diferente para o estabelecimento da conexão. O Cliente e o Servidor (hardware externo) têm um canal de comunicação estabelecido quan-

do estão parelhados, previamente. A partir deste momento, a troca de informações entre os dispositivos Cliente e Servidor está pronta para ser executada [SIG 2012].

4. Arquitetura proposta e funcionamento do projeto

Esta seção descreve o diagrama em blocos do projeto, bem como o detalhamento de cada parte que compõe este diagrama. As várias etapas são descritas e, além dos softwares envolvidos, o circuito elétrico (hardware) necessário ao funcionamento geral de toda a arquitetura.

Tendo em vista que este projeto trata-se de um misto de desenvolvimento de hardware e software, as considerações iniciais feitas, principalmente na Seção 3 (Especificações Técnicas), foram fundamentais para o entendimento de toda a estrutura.

4.1. Arquitetura do projeto

A base deste projeto é o Sistema Operacional para Dispositivos Móveis, o Android, atuando em conjunto com a tecnologia Bluetooth para acionamento de um Hardware Periférico composto por cargas elétricas, representadas por lâmpadas incandescentes e também o monitoramento de temperatura e de invasão de perímetro, representados pelos sensores de temperatura LM35 e infravermelho TIL32/TIL78, respectivamente.

O usuário tem em mãos um Tablet que comanda todo o acionamento das cargas externas e também recebe informações provenientes dos referidos sensores. A figura 5 exemplifica o que está descrito.

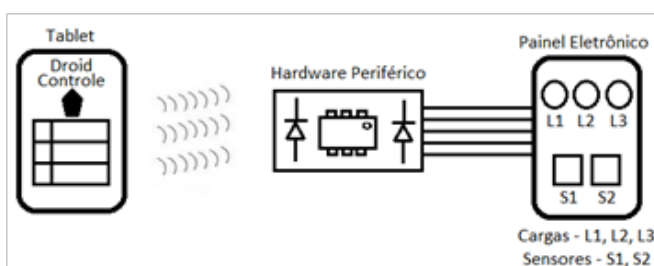


Figura 5. Diagrama em Blocos

O Hardware Periférico tem como seu principal componente o microcontrolador PIC18F45K22, o qual é programado em Linguagem C e através desta progra-

mação torna-se parte fundamental em todo processo de acionamento e monitoramento, pois possui internamente recursos de hardware, já descritos anteriormente, que o permite utilizar seus recursos de hardware internos, necessários ao funcionamento do projeto.

Os protocolos responsáveis por esta interatividade estão definidos dentro do microcontrolador, devidamente desenvolvido em linguagem C e dentro do aplicativo desenvolvido em JAVA, que está instalado no Tablet.

O projeto eletrônico completo do Hardware Periférico é mostrado na figura 6, a seguir. Como pode-se perceber, os dois sensores (Temperatura e Infravermelho) estão conectados aos pinos referentes ao Conversor A/D (AN0 e AN1) do microcontrolador PIC18F45K22. As Cargas elétricas, representadas por L1, L2 e L3, têm seus respectivos circuitos conectados as saídas do Port B (RB5, RB6 e RB7).

O responsável pela troca de informações entre o Tablet e o Hardware Periférico é o Módulo Bluetooth, conectado ao canal de comunicação Serial (RS232, pinos TX e RX). Este módulo captura tudo que é enviado pelo Tablet e envia para o canal serial, onde este interpreta os dados e faz os devidos acionamentos ou monitoramentos [Miyadaira 2009].

4.2. Aplicativo do projeto

Toda estrutura de hardware mostrada anteriormente tem como suporte para seu correto funcionamento, linguagens de programação que tornam possível toda a troca de informação para que os acionamentos e monitoramentos sejam executados de forma correta.

O aplicativo desenvolvido para rodar no Tablet, o DroidControle, foi elaborado na plataforma Android, usando linguagem JAVA no ambiente de desenvolvimento Netbeans. Um aplicativo Android, ao ser criado, gera uma estrutura de pastas onde são armazenados determinados arquivos responsáveis pelo funcionamento deste aplicativo. Como exemplos, a pasta "Drawable", que armazena todas as figuras usadas dentro da aplicação e a pasta "resource", que armazena os arquivos XML (Extended Marked Language, ou seja,

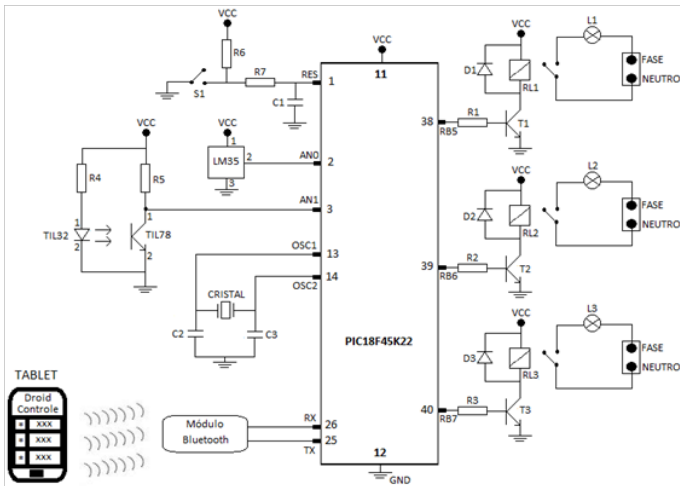


Figura 6. Hardware Periférico

Linguagem de Marcação Extendida), responsáveis pela elaboração das telas (layouts) usadas no aplicativo. O Hardware Periférico tem como seu componente principal o micro-controlador PIC18F45K22, o qual é programado em linguagem C, usando para isto o ambiente de programação do Compilador MikroC Pro for PIC, que é o compilador responsável por toda a elaboração e gravação do programa no microcontrolador. O programa desenvolvido em C é o responsável por receber as solicitações de acionamento, vindas do aplicativo DroidControle e enviar ao Tablet as leituras captadas pelos sensores de temperatura e infravermelho.

5. Resultados

Nesta etapa é mostrado o resultado obtido a partir de toda a estrutura criada na seção anterior.

Um cenário contendo todas as telas principais do software de controle e monitoramento e como é realizado todo o processo de interação Tablet e hardware periférico. A figura 7 mostra a tela inicial do aplicativo Droid Controle, onde são exibidos 3 (três) botões, com funcionalidades distintas (Automação, Relatório e Sobre). As telas a seguir fazem parte do aplicativo principal (DroidControle), instalado no Tablet.

Pressionando o botão "Automação", a tela correspondente a este botão é aberta e exibe uma série de botões, responsáveis pelo acionamento das cargas elétricas

(L1, L2 e L3) e pelo monitoramento dos sensores (Temperatura e Infravermelho), conforme mostra a figura 8a.

Um painel eletrônico exibe a montagem de um hardware formado por 3 (três) lâmpadas incandescentes, 2 sensores (Temperatura e Infravermelho), um módulo de controle (PIC18F45K22) e um módulo Bluetooth.

Todos estes componentes dão sustentação ao cenário apresentado até o momento, pois os resultados deste projeto são apresentados através da atuação dos mesmos. Pressionando o botão "Relatório", é exibida uma tela contendo uma listagem com informações (listview) de data e hora das ocorrências de cada evento referente ao monitoramento dos sensores de Temperatura e infravermelho,



Figura 7. Tela Principal do DROIDCONTROLE

conforme mostra a figura 8b. O armazenamento e consequente exibição dos eventos gerados pelos sensores só é possível devido ao fato da utilização do Banco de Dados nativo do Android, o SQLite. Por último, ao pressionar o botão "Sobre", uma tela com informações gerais sobre o aplicativo é exibida, conforme mostra a figura 8c.



(a)

Figura 8. Telas Secundárias do DROIDCONTROLE



(c)

Figura 8. Telas Secundárias do DROIDCONTROLE



(b)

Figura 8. Telas Secundárias do DROIDCONTROLE

6. Conclusão

O uso da tecnologia Bluetooth vem ganhando espaço a cada dia, pois é responsável por trazer ao ser humano benefícios que vão desde sistemas residenciais e industriais automatizados, até o monitoramento e aquisição de dados proveniente de sensores. A união das tecnologias Android e Bluetooth permite que tudo isto seja possível. A capacitação e a sensibilidade dos desenvolvedores devem estar sempre a postos para permitir que os usuários em geral usufruam de todos os benefícios que um projeto deste tipo possa trazer. Este artigo demonstra a possibilidade de automação e aquisição de dados em diversos processos, sejam residenciais ou industriais. A expectativa para este projeto é, futuramente, implementar um sistema de aquisição onde todos os dados e eventos de acionamento de cargas e as informações captadas pelos sensores sejam armazenados em um banco de dados nativo da própria plataforma Android, o SQLite. Estas informações armazenadas irão produzir relatórios para análise e tomada de decisão.

Portanto, o desenvolvimento de projetos de automação e aquisição de dados através de sistemas wireless via Bluetooth, permite que seja possível a interatividade de

diversos dispositivos como Tablets, Smartphones com hardwares periféricos, o que representa uma forte tendência no desenvolvimento de projetos com este tipo de tecnologia sem fio. Este projeto agrega grande contribuição na área de automação, pois mostra que é possível o desenvolvimento deste tipo de arquitetura, onde diversas tecnologias estão envolvidas.

7. Referências Bibliográficas

Alves, W. P. (2006). JAVA 2 Programação Multiplataforma, volume 1. Editora ÉRICA, 1st edition.

Boylestad, R. L. (1999). Dispositivos Eletrônicos e Teoria de Circuitos, volume 1. Editora LTC – Livros Técnicos e Científicos, 6th edition.

Capuano, F. G. (1988). Laboratório de Eletricidade e Eletrônica, volume 1. Editora ÉRICA, 1st edition.

Lee, W.-M. (2011). Introdução ao Desenvolvimento de Aplicativos para o ANDROID, volume 1. Editora Ciência Moderna, 1st edition.

Luiz, A. M. (2009). Eletromagnetismo, Teoria e Problemas Resolvidos, volume 3. Editora Livraria da Física, 1st edition.

Markus, O. (2004). Circuitos Elétricos em Corrente Contínua e Corrente Alternada, volume 1. Editora ÉRICA, 4th edition.

Mattos, E. C. T. (2007). Programação de Softwares em Java, volume 1. Digerati Books, 1st edition.

MICROCHIP (Acessado em 2012). Site oficial microchip. <http://www.microchip.com/>. MIKROELEKTRONIKA, I. (Acessado em 2012). Site oficial mikroelektronika. www.mikroe.com/.

Miyadaira, A. N. (2009). Microcontroladores PIC18, Aprenda e Programe em Linguagem C, volume 1. Editora ÉRICA, 1st edition.

Nicolosi, D. E. C. (2005). Microcontrolador 8051 com Linguagem C – Prático e Didático. Editora ÉRICA, 1st edition.

Schildt, H. (2010). C – Completo e Total, volume 1. Person Education, 3rd edition. SIG, B. (Acessado em 2012). Site oficial do bluetooth. www.bluetooth.org/.