



**CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA CELSO SUCKOW
DA FONSECA**

**DEPARTAMENTO DE ENGENHARIA ELÉTRICA – DEPEL
PROFESSORA: RENATA**

FALA DO TCC

**ALUNO: MANOEL FELIPE COSTA FURTADO
MATRICULA: 0620200190**

ORIENTADOR: VINICIUS COUTINHO

**RIO DE JANEIRO
DEZEMBRO DE 2015**

SUMÁRIO

Slide 1.....	1
Slide 2.....	1
Slide 3.....	2
Slide 4.....	3
Slide 5.....	3
Slide 6.....	3
Slide 7.....	5
Slide 8.....	6
Slide 9.....	7
Slide 10.....	7
Slide 11.....	7
Slide 12.....	8
Slide 13.....	8
Slide 14.....	8
Slide 15.....	8
Slide 16.....	10
Slide 17.....	12
Slide 18.....	13
Slide 19.....	13
Slide 20.....	14
Slide 21.....	15
Slide 22.....	17
Slide 23/24.....	17
Slide 25.....	17
Slide 26.....	20
Slide 27/28/29.....	20
Slide 30/31.....	23
Slide 32.....	23
Slide 33/34/35/36.....	24
Slide 37.....	26

Slide 38.....26

Slide 39.....28

Slide 46 - Apendice.....29

Slide 1

Capa da Apresentação. Formalidades.

Boa Tarde aos senhores Professores, mestres e doutores ao aceitarem a fazer parte da banca a minha defesa do meu TCC.

Meu nome é Manoel Felipe e apresentarei o meu TCC em cumprimento normativo do DEPS. Meu curso é Engenharia Elétrica com ênfase em Telecomunicações. E o meu Orientador e o Vinicius Coutinho presente.

Slide 2

Apresentação do tema do trabalho. Com relação à fala não posso falar com naturalidade porque sempre passo do tempo normativo. Então já tenho a Fala escrita no papel.

Antes fazer uma reflexão do ponto de visto do autor do trabalho:

Então o tema desse trabalho é uma tentativa de retribuir a sociedade. Conseguindo de fato estabelecer alguma telecomunicação de forma que seja de valor de uso para quem interessa.

O Nome do surgiu da motivação de conectar todo meu conhecimento adquirido durante a graduação e cursos extracurriculares num TCC. No início eu tinha proposto ao orientador usar a própria rede local(Wi-Fi, Ethernet via Cabo UTP) da residência conectada a internet. Mas sua sugestão era interessante de utilizar a rede 2G da operadora local. Trazendo a vantagem da cobertura. E também pode ser facilmente adaptável a rede 3G e LTE, devido a arquitetura de rede a ser apresentada.

A escolha do protocolo ZigBee, vai de encontro com o conceito em voga de utilizar a energia de forma sustentável.

A opção de monitorar e catalogar os eventos conectados ao PC é importante para melhorar o uso dos recursos que consomem energia, gás, água de forma sustentável. Informando um relatório do que foi gasto e a duração do uso. E de forma simplificada no Smartphone.

Slide 3

Comentar sobre o avanço da tecnologia na eletrônica(microeletrônica):

No Brasil é notório a insuficiência da produção nacional de tecnologia para ofertar ao mercado interno. E com um país subdesenvolvido apresenta atraso tecnológico nesse mundo globalizado. E hoje com a facilidade de se obter esses produtos importados com alto valor tecnológico, provoca uma concorrência com a produção nacional e torna cada vês mais barato devido a oferta internacional.

Comentar sobre a eficiência energética, segurança e facilidade:

Eficiência energética:

O principal desafio na engenharia elétrica é garantir a eficiência do uso da energia fornecida pela fonte. A microeletrônica usada no trabalho não foi desenvolvida por mim mas acredito que tende esse objetivo. A escolha do protocolo ZigBee também é nesse contexto.

Segurança:

O uso da rede de telefonia móvel no caso 2G (GSM), garante uma segurança muito eficiente. O protocolo ZigBee também tem sua segurança que pode ser implementada mas não foi implementada.

Facilidade:

O uso da tecnologia 2G (GSM, SIM CARD) com protocolo ZigBee sendo **interoperado** pelo Arduino. É o tema para defender do trabalho.

Slide 4

Explicar essa “Arquitetura de Rede”.

Essa Arquitetura de rede proposta foi elaborado por mim no software Paint do Windows depois da pesquisa de viabilidade para realizar o trabalho. Ela tem como objetivo com uma única imagem demonstrar todo o trabalho.

Garantir a Bidirecionalidade é importante e viável. Mesmo não implementado é crucial para seu funcionamento cobrindo todos os aspetos do tema do trabalho. Como a exemplo receber um SMS se uma lâmpada está acesa, e que foi catalogado esse evento pelo PC.

Falar rapidamente sobre o cada componente.

Slide 5

Essa imagem é interessante pois ela demonstra quase todas as possibilidades do que se pode monitorar. Claro que um XBee para cada sensor seria caro.

Está voltada mais para a residência. Mas pode ser adaptado a apartamentos, salas comerciais e em indústrias.

A própria instalação elétrica etc pode ser da residencial, AP etc adaptado. Fios fornecendo 5Vs, 3.3v a exemplo.

Slide 6

Nesse slide se refere aos temas para implementação da arquitetura de rede proposta.

Protocolo Zig-Bee - IEEE 802.15.4:

A tecnologia Bluetooth, foi inventada pela fabricante Ericsson, desenvolvido por Jaap Haartsen e Sven Mattisson em 1994, e foi, ratificada pelo IEEE posteriormente através do padrão 802.15.1. Um processo similar aconteceu com a tecnologia ZigBee (ZigBee Alliance) e o padrão correspondente, IEEE 802.15.4.

ZigBee Alliance visa proporcionar às camadas superiores da pilha de protocolos (a partir da camada de rede para aplicação) a interoperabilidade da rede de dados, serviços de segurança e uma gama de soluções de controle para redes sem fio residenciais e prediais. A interoperabilidade é a principal característica; ela garante que fabricantes diferentes possam se comunicar.

O nome ZigBee foi criado a partir da analogia entre o funcionamento de uma Rede em Malha, e o modo como as abelhas trabalham e se locomovem. As abelhas que vivem em colméia voam em Zig...Zag, e dessa forma, durante um voo a trabalho em busca de néctar, trocam informações com outros membros da colmeia sobre, distância, direção e localização de onde encontrar alimentos. Uma Malha ZigBee dispõe de vários caminhos possíveis entre cada nó da Rede para a passagem da informação, assim, é possível eliminar falhas se um nó estiver inoperante, simplesmente mudando o percurso da informação. É uma analogia somente parecido com um protocolo de Roteamento.

O padrão IEEE 802.15.4 atende o tema do trabalho. Em 2006 foi revisado e publicado o padrão IEEE 802.15.4

IEEE 802.15.4 (LR-WPAN - low-rate wireless personal area networks): Se destina a servir aplicações industriais (usando sensores), residenciais e médicas, com baixo consumo de energia e baixo custo, garantindo a taxa de dados necessária e possibilidade de QoS. A baixa taxa de dados permite que o LR-WPAN consuma pouca energia.

É esperado do ZigBee, fornecer conectividade de baixo custo e baixo consumo de energia para equipamentos onde a vida útil da bateria é extremamente importante, e que não necessite de altas taxas de dados, como aqueles utilizados pelo Bluetooth. O ZigBee pode ser implementada em redes de malha bem maiores que o Bluetooth.

Tecnologia GSM - Comandos AT:

Os primeiros modems comerciais foram desenvolvidos pela empresa AT&T na década de sessenta. Os comandos AT's, tornou-se um padrão para a indústria (SIEMENS,2003).

Microcontrolador – Atmel:

Os microcontroladores oferecem uma tecnologia essencial que alimenta comunicações máquina-a-máquina (M2M).

Segundo Atmel oferece arquiteturas comprovadas e otimizadas em baixo consumo de energia e outras tantas características. Logo atende o tema do trabalho.

Hardware Arduino e suas bibliotecas:

Utiliza o microcontrolador ATmega2560 e com suas bibliotecas garante a interoperabilidade da arquitetura proposta.

Linguagem de Programação C/C++, Shell Script, SQL, PHP, HTML, CSS, JavaScript etc:

Esse tema é muito abrangente e não foi explorado como programado durante o trabalho. Somente a Linguagem de Programação C foi utilizada.

Explicar o porquê de tantas linguagens de programações e sua aplicabilidade:

Fiz o vestibular para o tecnólogo em Tecnologia em Sistemas para Internet. Mas como não era permitido cursar as duas larguei no segundo período e escolhi em continuar em Telecom.

Somando o pouco conhecimento que obtive mais cursos extracurriculares sobre sistemas Linux e o curso da Cisco que me ajudou na graduação em telecom., sei que é perfeitamente implementar serviços complementando o tema do trabalho. Claro que as bibliotecas para o Arduino é vital para concretizar esse meu objetivo.

Com o World Wide Web Consortium (W3C) pode observar como essas linguagens se complementam. No trabalho escrito fiz uma menção sobre plugins Cordova que está em desenvolvimento para plataforma Android etc.

Slide 7

Para mim: É essa a parte mais importante do trabalho e significativa que infelizmente pouco explorada na prática por exemplo com os cursos entre eles no CEFET-RJ e com outras Instituições de Ensino e entre elas.

Na prática a interdisciplinaridade é um esforço de superar a fragmentação do conhecimento, tornar este relacionado com a realidade e os problemas da vida moderna. Na ciência, por sua vez, os esforços estão na busca de respostas, impossíveis com os conhecimentos fragmentados de uma única área especializada.

Para viabilizar a rede proposta pode ser observado a necessidade da interdisciplinaridade dos cursos de Engenharia Industrial Elétrica com ênfase em Eletrônica, Telecomunicações, de Controle e Automação, Ciência da Computação e Tecnologia em Sistemas para Internet no CEFET/RJ.

Exemplos Reais:

1) Na UNIVERSIDADE FEDERAL DO CEARÁ - UFC. Oferta o Bacharelado em Engenharia de Teleinformática reconhecido pelo MEC.

2) criação em 1986, o Grupo de Teleinformática e Automação (GTA) vinculado a (COPPE) da UFRJ.

Slide 8

Linus Torvald: “Estou fazendo um sistema operacional gratuito (apenas um hobby, não será grande e profissional como GNU) para 386/486 AT.”

O sistema operacional GNU é um sistema de software livre completo, compatível com o Unix. GNU significa “GNU's Not Unix” (GNU Não é Unix).

A palavra “livre” em “software livre” se refere à liberdade, não ao preço. E sim as quatro liberdades específicas ao usá-lo: a liberdade de executar o programa como você desejar; a liberdade de copiá-lo e dá-lo a seus amigos e colegas; a liberdade de modificar o programa como você desejar, por ter acesso total ao código-fonte; a liberdade de distribuir versões melhoradas e, portanto, ajudar a construir a comunidade.

Por volta do início dos anos 90 Quando Linus Torvalds escreveu o Linux, ele completou a última grande lacuna. Pessoas puderam então colocar o Linux junto com o sistema GNU para compor um sistema livre completo: um sistema GNU baseado em Linux (ou sistema GNU/Linux, para simplificar).

Massimo Banzi: “Tornar acessíveis os microcontroladores com o código aberto para projetos interativos, a partir de instalações artísticas.”

O surgimento do Arduino se dá no ano de 2005, na Itália, por um professor chamado Massimo Banzi, com intuito de ensinar eletrônica e programação de computadores para os alunos de design, para uso em projeto de arte, interativos e robótica. Ele faz parte do conceito de uma plataforma eletrônica "open-source". Isso significa que qualquer pessoa pode montar seu próprio Arduino e modificá-lo, tanto o software quanto o hardware. Existem inúmeras pessoas desenvolvendo diversas aplicações programas e placas que podem ser acopladas ao Arduino.

Creative Commons tem sido abraçada por muitos criadores de conteúdos, pois permite controle sobre a maneira como sua propriedade intelectual será compartilhada publicamente. O software Arduino também é open-source.

O código fonte para o ambiente Java é liberado sob a GPL e C / C ++ bibliotecas de microcontroladores estão sob a LGPL.

A principal diferença entre a GPL e a LGPL é que esta permite também a associação com programas que não estejam sob as licenças GPL ou LGPL, incluindo Software proprietário.

Relação comum ?? Colaboração das pessoas pela internet. Globalização.

“Você não precisa mais da permissão de alguém para fazer algo incrível.” Ter acesso ao código fonte e modificar à sua vontade.

Slide 9

Uma vez exposto todo o conceito do tema do trabalho. Vou a demonstração prática.

O objetivo completo mostrar o funcionamento mostrando o protótipo na bancada.

Mas está faltando uma parte da arquitetura já apresentada....

Slide 10

Mostra o funcionando !!!! Do que consegui somente !!!!

Slide 11

Explicar a programação necessária.

Programação da comunicação entre celular(GSM-SMS) e modulo (Shield-GSM) acoplado no Arduino.

Programação da comunicação entre o Arduino e Xbee acoplado no Arduino. E entre o Xbee acoplado no Arduino e o outro Xbee receptor.

Programação da comunicação das mensagens SMS através de comandos AT integrando todo o projeto.

Programação da comunicação entre o PC e o Arduino.

Desenvolvimento de interface de usuário e o registro dos eventos através de banco de dados e HTML, utilizando serviços GNU/Linux.

Slide 12

Explicar a imagem que motivou em fazer o receptor sem microcontrolador no circuito.

Talvez seja necessário o microcontrolador mas montar ele sem é o objetivo.

O XBee receptor pode e deve ser também o transmissor garantindo a bidirecionalidade. Ele transmitirá por estímulos do circuito ligado nos pinos específicos detectando os eventos. Ex.: Mudança de tensão, sensores em geral.

...

Slide 13

Comentar portas do Microcontrolador com as portas do Arduino.

Comentar que não sabia nada de como funcionava um Microcontrolador antes de começar o trabalho e agora estou começando a entender e usar.

Slide 14

Comentar sobre os Shields mostrando exemplos.

Slide 15

Comentar sobre o XBee:

Tecnologia baseada no padrão IEEE 802.15.4/ZigBee Alliance.

XBee é a marca da Digi International® Inc. (Digi®) International para uma família de módulos de rádio. Os primeiros rádios XBee foram introduzidos sob a marca MaxStream em 2005 e foram baseados no padrão 802.15.4/2003 projetado para ponto-a-ponto e comunicação estrela para taxas de transmissão “over-the-air” de 250 kb/s.

Comentar sobre testes ANATEL vs Teroria. Exemplo de equipamento para isso. Muito caro. Interessante pois dá para explorar muitos conceitos de Eng. de Telecomunicações. APENDICE.

O módulo XBee no meu entendimento é um microcontrolador com uma antena. Esse microcontrolador foi construído seguindo o protocolo IEEE 802.15.4/ZigBee Alliance e aperfeiçoado com a marca XBee. Garantindo que funcione nas características informada pelo fabricante.

Plataforma	XBee	XBee-PRO
Desempenho		
Potência de Sáida	1mW (+0 dBm) North American & International	63 mW (+18 dBm) North American 10 mW (+10 dBm) International
Cobertura – Indoor/Urban	Até 100 ft (30 m)	Até 300 ft (90 m)
Cobertura – Outdoor/RF	Até 300 ft (90 m)	Até 1 mile (1.6 km) RF LOS
Sensibilidade do receptor	-92 dBm	-100 dBm (all variants)
Taxa de dados RF	250 Kbps	250 Kbps
Frequência de Operação	2.4 GHz	2.4 GHz
Taxa de dados na interface	Até 115.2 Kbps	Até 115.2 Kbps
Característica da Rede		
Método de Acesso – Spread spectrum type	DSSS (Direct Sequence Spread Spectrum)	
Topologias de Rede	Ponto ou Malha, Estrela, Arvore (Point-to-point, point-to-multipoint, & peer-to-peer)	
Tratamento de Erros	Tentativas e Confirmações (Retries & acknowledgements_)	
Opções de Filtragem	PAN ID, Channel, and 64-bit addresses	
Quantidades dos canais	16 Channels	12 Channels
Endereçamento	65,000 endereços disponível na rede por canal	
Potencia		
Tensão de Alimentação	2.8 - 3.4 VDC Recomendação: 3.0 – 3.4 VDC	2.8 - 3.4 VDC Recomendação: 3.0 - 3.4 VDC
Corrente de transmissão	45 mA (@ 3.3 V) boost mode ; 35 mA (@ 3.3 V) normal mode	215 mA (@ 3.3 V)
Corrente de Recepção	50 mA (@ 3.3 V)	55 mA (@ 3.3 V)
Corrente no modo Sleep	<10 µA até 25° C	<10 µA até 25° C

DSSS-OQPSK

Offset Quadrature Phase Shift Keying is referred as OQPSK Direct Sequence Spread Spectrum

Fornece uma densidade espectral da potência muito baixa espalhando a potência do sinal sobre uma faixa de frequência muito larga.

O princípio das técnicas de modulação usando espalhamento espectral é aumentar a quantidade de bits utilizados para transmitir uma mesma informação, de modo espalhar o espectro de frequência do sinal. Desta forma, aumenta-se a banda de frequência na qual o sinal é transmitido.

Normalmente, utiliza-se um código de espalhamento que é multiplicado pelos bits de informação. Após ser espalhado, o sinal é modulado em uma portadora. Normalmente, são usados dois esquemas de modulação para DSSS (BPSK e OPSK)

Slide 16

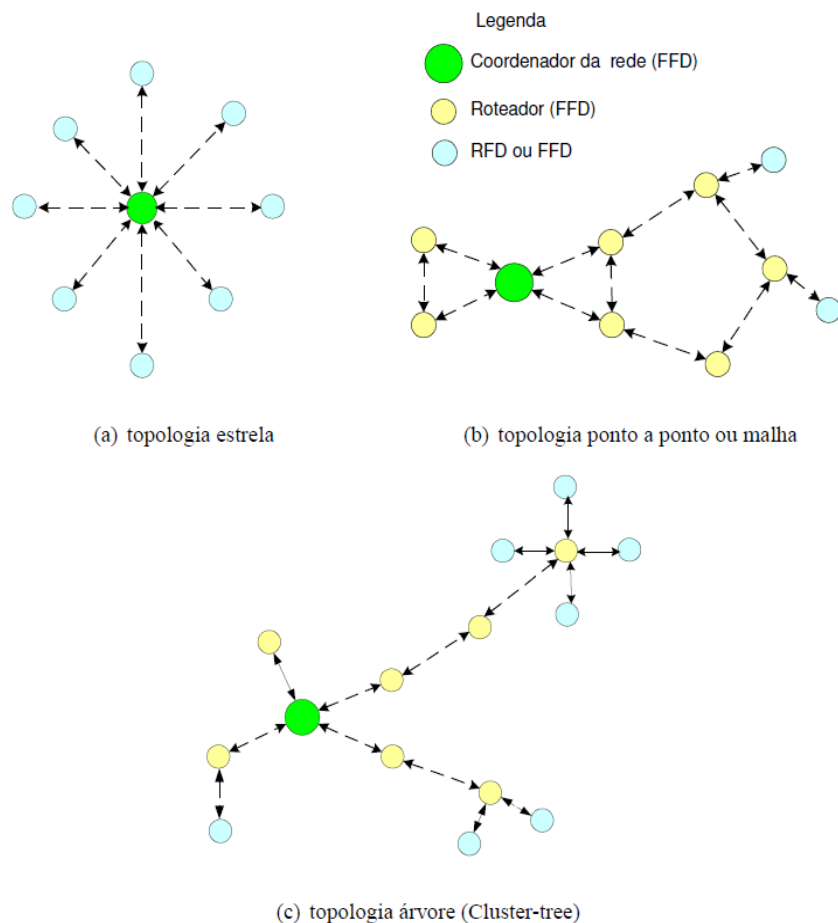
Algumas características do IEEE 802.15.4 TG4 [3]:

- a) As taxas de dados de 250 kb/s, 40 kb/s e 20 kb/s.
- b) Dois modos de endereçamento; 16-bit curto e de 64 bits IEEE.
- c) Suporte para dispositivos com latência crítica, como joysticks.
- d) Operação em topologia estrela, ponto a ponto e árvore (Star, Mesh e Cluster Tree).
- e) Método de acesso CSMA-CA ou ALOHA.
- f) Estabelecimento automático da rede pelo coordenador.
- g) O protocolo usa o handshake para melhorar a confiabilidade na transferência.
- h) O gerenciamento de energia para garantir baixo consumo de energia como a detecção de energia (ED - Energy detection (ED)) e o indicador que dá a qualidade do link (LQI(Link Quality Indication))
- i) 16 canais na banda ISM de 2,4 GHz, 10 canais em 915MHz e 1 canal na banda de 868 MHz
- j) Opcional. Alocação garantida do tempo de slots (GTSs (Guaranteed Time Slots))

Somente dois tipos de dispositivos podem participar de uma rede IEEE 802.15.4: dispositivos de – Função Completa (FFD – Full-Function) ou de Função Reduzida (RFD – Reduced-Function). O FFD é capaz de servir como um coordenador de pessoal (PAN – Personal Area Network) ou como um coordenador diferentemente do RFD, ou seja, o FFD pode assumir as funções do RFD. A rede deve incluir pelo menos um FFD, atuando com coordenador PAN. O RFD é destinado a aplicações que são extremamente simples, como acender e apagar uma lâmpada, um sensor infravermelho passivo, ou seja, ele não tem a necessidade de enviar grandes quantidades de dados e só se associa com um único FFD de cada vez, logo o RFD pode ser implementado usando o mínimo de recursos e capacidade de memória. Nesse padrão basta ter dois ou mais dispositivos de comunicação, sendo 1 deles necessariamente do tipo FFD, na mesma camada física para formar a WPAN.

Coordenador ZigBee	Roteador ZigBee	Dispositivo Final ZigBee	Função na Camada de Rede
X			Estabelecer uma nova rede ZigBee
X	X		Conceder endereço lógico de rede
X	X		Permitir que dispositivos entrem ou saiam da rede
X	X		Manter lista de vizinhos e rotas
X	X		Rotear pacotes da camada de rede
X	X	X	Transferir pacotes da camada de rede

Tipos de dispositivos	Funcionalidade disponíveis no protocolo	Fonte de alimentação típica	Configuração típica do receptor
FFD – Full Function Device	A maioria ou todas	Principal	Ligado quando em espera
RFD – Reduced Function Device	Limitada	Bateria	Desligado quando em espera



Slide 17

Comentar sobre os pinos. É um microcontrolador né !!! Parecido com o Sim900 também.

Pino	Nome	Direção	Descrição
1	VCC	-	Alimentação (3,3v)
2	DOUT	Saída	Saída de dados UART
3	DIN	Entrada	Entrada de dados UART
4	DO8	Ambos	Saída Digital 8
5	RESET	Entrada	Reset do Módulo
6	PWM0/RSSI	Ambos	Saída PWM0 ou indicador de força do sinal RSSI
7	PWM1	Ambos	Saída PWM1
8	Reservado	-	Não Conectado
9	SLEEP/18	Ambos	Pino de controle do modo Sleep ou Entrada Digital 8
10	GND	-	Terra
11	AD4/DIO4	Ambos	Entrada analógica 4 ou E/S Digital 4
12	CTS/DIO	Ambos	Sinal de controle CTS(Clear-to-send) ou E/S Digital 7
13	ON/Sleep	Saída	Indicador de estado do módulo
14	VREF	Entrada	Voltagem de referência para as entradas A/D
15	AD5/DIO5	Ambos	Entrada analógica 5 ou E/S Digital 5
16	RTS/AD6/DIO6	Ambos	Sinal de controle RTS(Request-to-send) ou Entrada analógica 6 ou E/S Digital 6
17	AD3/DIO3	Ambos	Entrada analógica 3 ou E/S Digital 3
18	AD2/DIO2	Ambos	Entrada analógica 2 ou E/S Digital 2
19	AD1/DIO1	Ambos	Entrada analógica 1 ou E/S Digital 1
20	AD0/DIO0	Ambos	Entrada analógica 0 ou E/S Digital 0

Slide 18

Falar da comunicação UART-to-SPI e da importância de um outro Microcontrolador como o Arduino Uno com o XBee RXTX

UART, for Universal Asynchronous Receiver Transmitter

Serial Peripheral Interface (SPI)

MISO (Master In Slave Out) - The Slave line for sending data to the master,

MOSI (Master Out Slave In) - The Master line for sending data to the peripherals,

SCK (Serial Clock) - The clock pulses which synchronize data transmission generated by the master and one line specific for every device:

SS (Slave Select) - the pin on each device that the master can use to enable and disable specific devices.

Slide 19

A Segunda Geração da rede de telefonia móvel no Brasil foi marcada pela disputa entre a tecnologia CDMA e GSM. E foi um marco no que se refere a mudança da rede analógica para a digital.

Sigla para Global System for Mobile Communications (algo como “Sistema Global para Comunicações Móveis”), o GSM é uma tecnologia desenvolvida e amplamente utilizada na Europa, e que não tardou para chegar em países de outros continentes, inclusive ao Brasil.

Uma característica bastante interessante da tecnologia GSM é o uso de um dispositivo chamado SIM (Subscriber Identity Module). Conhecido no Brasil simplesmente como “chip” ou como “cartão SIM”, este minúsculo dispositivo armazena informações referentes à linha telefônica e ao usuário, como número, operadora, lista de contatos, entre outros.

Redes GSM podem operar em várias frequências. Quad-Band (850/900/1800/1900MHz).

Outra vantagem do GSM é que, por esta ser uma tecnologia utilizada em diversos países, torna-se mais fácil a ativação do recurso de roaming.

O uso dos cartões SIM também ajuda a evitar um problema recorrente em dispositivos

móveis CDMA: a clonagem do aparelho

Tecnologias 2,5G

No início só permitia comunicação por voz. No entanto, como você sabe, a telefonia móvel também possibilita a troca de mensagens de textos curtas via SMS (Short Message Service) e acesso à internet.

Sigla para General Packet Radio Service (algo como “Padrão de Transmissão de Rádio por Pacote”), o GRPS tem como base a tecnologia GSM a mais usada. Seu foco, é claro, é a transferência de dados, mais precisamente, a comunicação com a internet, dada a sua compatibilidade com o protocolo IP. Esta atividade pode ser realizada de maneira simultânea às chamadas de voz.

Slide 20

Apresentar o módulo suas características. Falar também que existem módulos para 3G e LTE.

O módulo SIM900 é fabricado pela SIMCOM (SHANGHAI SIMCOM LTD.) e distribuído no Brasil pela ME Componentes (Importadora de Componentes e Equipamentos).

Principais Características

- Banda: Quad-Band GSM/GPRS 850/900/1800/1900 MHz
- Encapsulamento: SMT (surfaced mount technology).
- TCP/IP: Pilha TCP/IP já embutida.
- Antena: Externa.
- Aplicações: Rastreadores Veiculares, Rastreadores Pessoais, M2M, entre outras.

O SIM900 é um completo módulo GSM e por isso, requer um chip (cartão) GSM para que possa transmitir/receber voz, dados e SMS.

Slide 21

Diagrama lógico Shild SIM900

Canal em banda básica aceita frequências próximas ao zero. Uma definição equivalente é que um canal em banda básica aceita uma faixa de frequências que é da mesma ordem de grandeza da sua frequência central (a média entre a maior e a menor frequência que passam pelo canal). Assim, um modem discado, que transmite aproximadamente entre 100Hz e 3.400Hz, é um canal em banda básica. Por outro lado, o sistema celular, que transmite faixas de frequência de cerca de 30kHz a uma frequência de 900MHz, é conhecido como um canal em banda passante.

Recebe e envia Banda Base voz \longleftrightarrow analógica \longleftrightarrow digital \longleftrightarrow Banda Passante.

Ficha Técnica	
GPRS multi-slot	Class 10 (Default) 8 (option)
GPRS mobile station	Class B
Controle	Via comandos AT (GSM 07.07, 07.05, mais os comandos AT da SIMCom)
Potência na transmissão	Class 4 (2W) at GSM 850 and EGSM 900. Class1 (1W) at DCS 1800 and PCS 1900
Nível	LVTTL
Tensão de Entrada	3.2V ~ 4.8V
Medidas	24mm x 24mm x 3mm
Pinagem	68 pinos
Economia da Potencia	Típico consumo de potencia na modo sleep é 1mA (BS-PA-MFRMS=9)
SIM application toolkit	GSM 11.14 Release 99
Real Time Clock	Support RTC
Banda das Frequências	SIM900 Quad-band: GSM 850, EGSM 900, DCS 1800, PCS 1900. SIM 900 pode procurar as 4 bandas de frequências automaticamente. As bandas de frequência também pode ser definida pelo comando AT "AT+CBAND".
SIM interface	Support SIM card: 1.8v , 3V
Antena Externa	Antenna Pad
Temperatura de Operação	Normal Operation: -30 ⁰ C ~ +80 ⁰ C.
Data GPRS	GPRS data Downlink transfer: Max 85.6 kbps. GPRS data Uplink transfer: Max 42.8 kbps. Coding Scheme: CS-1, CS-2, CS-3 and CS-4
SMS	MT, MP, CB, Text and PDU Mode. SMS Storage: SIM card
Serial port	Full moden interface with status and control lines, unbalanced, asynchronous. 1200bps to 115200bps. Can be user for AT commands or data stream. Support RTS;CTS hardware handshale and software ON/OFF flow control. Multiplex ability accordin to GSM 07.10 Mult

Slide 22

Tipo	Simbolo	Descrição
Pinagem entre Arduino e o SIM900	D0	Pino a comunicação, pode ser selecionado para RX ou TX
	D1	Pino a comunicação, pode ser selecionado para RX ou TX
	D2	Pino a comunicação, pode ser selecionado para RX ou TX
	D3	Pino a comunicação, pode ser selecionado para RX ou TX
	D4	Conecta o SIM900 UART Bus RI
	D5	Pino para controlar o Reset SIM900
	D6	Pino para controlar o Power on/off SIM900
DEBUG	GND	Pino para controlar o Power Ground
	PWR	Pino para controlar o Power Supply
	RX	SIM900 Debug Port Recevie. For debugging and uploading firmware
	TX	SIM900 Debug Port Transmit. For debugging and uploading firmware

Slide 23/24

Comentar que eu usei o software TinyCad para tentar desenhar o protótipo do circuito. E comentar.

Slide 25

A software/bibliotecas é muito importante pois sem ele a inoperabilidade não completa.

No nível de software, as bibliotecas do Arduino (Comunidade) permite que qualquer outra linguagem de programação com capacidade de se comunicar em série nativamente, para não precisar de intermediário, exemplos Python, Processing, C/C++, Linux TTY etc.

O ambiente de desenvolvimento integrado Arduino (IDE) é uma aplicação multiplataforma escrita em Java, derivada dos projetos Processing e Wiring.

Um programa ou código escrito para Arduino é chamado de "Sketch".

Os plugins Cordova para Android e etc é um ótimo exemplo da integração das bibliotecas. Cria toda uma ponte para proporcionar a interoperabilidade das tecnologias atuais, com os dispositivos móveis (smartphones).

Essa interoperabilidade é caracterizada desde a comunicação serial até esses aplicativos híbridos para os dispositivos móveis, através desses plugins.

Tipo de bibliotecas	Bibliotecas	Descrição
Standard Libraries	EEPROM	reading and writing to "permanent" storage
	Ethernet	for connecting to the internet using the Arduino Ethernet Shield
	Firmata	for communicating with applications on the computer using a standard serial protocol.
	GSM	for connecting to a GSM/GRPS network with the GSM shield.
	LiquidCrystal	for controlling liquid crystal displays (LCDs)
	SD	for reading and writing SD cards
	Servo	for controlling servo motors
	SPI	for communicating with devices using the Serial Peripheral Interface (SPI) Bus
	SoftwareSerial	for serial communication on any digital pins. Version 1.0 and later of Arduino incorporate Mikal Hart's NewSoftSeriallibrary as SoftwareSerial.
	Stepper	for controlling stepper motors
	TFT	for drawing text , images, and shapes on the Arduino TFT screen
	WiFi	for connecting to the internet using the Arduino WiFi shield
	Wire	Two Wire Interface (TWI/I2C) for sending and receiving data over a net of devices or sensors.
Due Only Libraries	Audio	Play audio files from a SD card.
	Scheduler	Manage multiple nonblocking tasks.
	USBHost	Communicate with USB peripherals like mice and keyboards.
Esplora Only Library	Esplora	this library enable you to easily access to various sensors and actuators mounted on the Esplora board.
Arduino Robot Library	Robot	this library enables easy access to the functions of the Arduino Robot
Arduino Yún Bridge Library	Bridge Library	Enables communication between the Linux processor and the Arduino on the Yún.
USB Libraries (Leonardo, Micro, Due, and Esplora)	Keyboard	Send keystrokes to an attached computer.
	Mouse	Control cursor movement on a connected computer.

Tipo de bibliotecas	Bibliotecas	Descrição
Contributed Libraries	If you're using one of these libraries, you need to install it first. See these instructions for details on installation. There's also a tutorial on writing your own libraries.	
Communication (networking and protocols)	Messenger	for processing text based messages from the computer
	NewSoftSerial	an improved version of the SoftwareSerial library
	OneWire	control devices (from Dallas Semiconductor) that use the One Wire protocol.
	PS2Keyboard	read characters from a PS2 keyboard.
	Simple Message System	send messages between Arduino and the computer
	SSerial2Mobile	send text messages or emails using a cell phone (via AT commands over software serial)
	Webduino	extensible web server library (for use with the Arduino Ethernet Shield)
	X10	Sending X10 signals over AC power lines
	XBee	for communicating with XBees in API mode
	SerialControl	Remote control other Arduinos over a serial connection
Sensing	Capacitive Sensing	turn two or more pins into capacitive sensors
	Debounce	for reading noisy digital inputs (e.g. from buttons)
Displays and LEDs	GFX	base class with standard graphics routines (by Adafruit Industries)
	GLCD	graphics routines for LCD based on the KS0108 or equivalent chipset.
	Improved LCD library	fixes LCD initialization bugs in official Arduino LCD library
	LedControl	for controlling LED matrices or seven segment displays with a MAX7221 or MAX7219.
	LedControl	an alternative to the Matrix library for driving multiple LEDs with Maxim chips.
	LedDisplay	control of a HCMS 29xx scrolling LED display.
	Matrix	Basic LED Matrix display manipulation library
	PCD8544	for the LCD controller on Nokia 55100 like displays (by Adafruit Industries)
	Sprite	Basic image sprite manipulation library for use in animations with an LED matrix
	ST7735	for the LCD controller on a 1.8", 128x160 TFT screen (by Adafruit Industries)
Audio and Waveforms	FFT	frequency analysis of audio or other analog signals
	Tone	generate audio frequency square waves in the background on any microcontroller pin
Motors and PWM	TLC5940	16 channel 12 bit PWM controller.
Timing	DateTime	a library for keeping track of the current date and time in software.
	Metro	help you time actions at regular intervals
	MsTimer2	uses the timer 2 interrupt to trigger an action every N milliseconds.
Utilities	PString	a lightweight class for printing to buffers
	Streaming	a method to simplify print statements

Slide 26

Comentar os principais que existe na parte escrita as configurações e código bem detalhado.

Slide 27/28/29

Comentar sobre o código

A cada loop é executado o comando AT [mySerial.println("AT+CMGF=1");] para forçar o modo texto e pede-se para mostrar os dados recebidos (no caso proveniente do SMS [mostraDadosSerial();]). Pede-se para esperar 1 segundo [delay(1000);].

Em seguida, define-se uma condição do tipo While [while (mySerial.available(>0))]. Essa condição vai verificar se a entrada no RX recebeu um valor maior que zero; no caso, quando chega um SMS com certeza é maior que zero. Para que isso funcione é chamada a rotina [void mostraDadosSerial()], que determina que quando a função definida pela biblioteca [mySerial.available()!=0] for diferente de 0, ou seja, trafega algum dado, se executa-se a função [Serial.write(mySerial.read());].

A função [Serial.write(mySerial.read());] executa o seguinte procedimento: se existe algum dado trafegando essa função vai ler de forma serial todos os dados com a função [mySerial.read()]; Logo, os dados estarão já lidos em [mySerial.read()]. Essa leitura será passada para a variável [char inChar] de forma dinâmica, ou seja, a cada loop dentro da condição While irá ler dado por dado (nesse caso irá ler os dados do SMS recebido). A cada Loop dentro do While será detectado se houve algum dado e será armazenado na variável local [char inChar]. Na ausência de dados o void loop() ficará em loop a cada 1 segundo.

Um passo importante e necessário é definir uma variável inChar do tipo Char porque a cada loop dentro da condição [while (mySerial.available(>0))], em cada leitura, [mySerial.read();], vai transformar o byte em caractere.

No momento em que a variável do tipo String linhaAtual começa a receber caractere por caractere a cada loop da condição do While através da variável inChar. Então esses caracteres é inserida numa outra variável sms do tipo String (linha 57), com isso a variável linhaAtual se torna uma variável como ponte entre o byte que a variável inChar, recebe e transforma em caractere, para a variável sms.

Quando houver quebra de linha para evitar problema do reconhecimento do caractere na variável sms do tipo String existe a condição do tipo IF, como mostrado no trecho do código abaixo.

Toda vez que existir um quebra de linha a variável linhaAtual irá ser limpa. Pois a cada byte lido pode haver uma quebra de linha, pois como explico anteriormente a função [mySerial.read();] está sendo escrita devido a função [Serial.write(mySerial.read());], isso é importante para que a Sketch cumpra sua função, pois as variáveis linhaAtual e inChar vai servir para descriptografar a chave contida no texto do SMS, explicado a diante para interpretar a mensagem do SMS. Antes sobre essa questão da quebra de linha é profundado

A criptografia na Sketch se caracteriza por dois caracteres especiais distintos definidos segundo o critério abaixo:

- 1) Os dois caracteres escolhidos não podem estar presentes no cabeçalho do SMS.
- 2) O carácter escolhido não pode estar presente no texto do SMS.

Esses caracteres determinam o início e o fim do texto.

Na Sketch do trabalho foi escolhido o caracter "@" que determinará o início da leitura do texto do SMS e o caracter "<" que determinará o final da leitura do texto do SMS.

A condição if, na linha 48, verificará se a função [string.endsWith] é verdadeira. Essa função verificará a string linhaAtual se termina com o caractere "@". Sendo verdadeira a variável lendoSMS do tipo boolean será verdadeira e fará que a variável sms seja limpa, segundo a condição [if (linhaAtual.endsWith("@"))]. Isso significa que a exemplo, uma mensagem de texto enviada por SMS "@Liga LED<", acionará o preenchimento da variável SMS. Ou seja, todos os caracteres que a variável linhaAtual recebe antes do caractere "@" é ignorado.

A condição if, na linha 55, verificará se a variável lendoSMS do tipo boolean é verdadeira, ou seja, é será incrementado o carácter na variável SMS do tipo string. Enquanto a variável lendoSMS for verdadeira e a condição [if (inChar != '<')] fará que o caractere armazenada na variável inChar, do tipo Char, coloque o valor do caractere na variável SMS que foi limpa como explicado no paragrafo acima. Dessa forma todo o texto depois do surgimento do caractere "@" e incluindo ele será armazenada na string SMS. Quando surgir o caractere "<" entrará na condição contrária de [if (inChar != '<')], logo a estrutura do eles, linhas 59 á 79, entra em execução.

Dentro do bloco, linhas 59 á 79, o estado variável lendoSMS muda para falsa [lendoSMS = false;] e escreve a variável sms na serial pela função [Serial.println(sms);], escreve só para monitorar pelo serial monitor figura 5.1.

Quando a string SMS receber todos os caracteres de forma atender uma das condições, linhas: 63, 68 e 73. A Sketch (código) terá realizado seu objetivo que será mudar o estado lógico da porta do Arduino, somente quando solicitado por SMS pelo usuário.

Condições:

- a) `[if(sms=="@Liga LED" && smsAnterior!=sms)],`
- b) `[if(sms=="@Desliga LED" && smsAnterior!=sms)],`
- c) `[if(sms == smsAnterior)]`

Antes de explicar as condições, é fundamental para entendimento que a forma que a Sketch (código) foi desenvolvida, tem como objetivo de ser a forma mais simples para acionar com segurança. Então foi criado a variável `smsAnterior` do tipo string para evitar mensagem SMS repetida. O desenvolvimento do código para o trabalho foi para demonstrar o protótipo na defesa do trabalho, mostrado na sessão 4. Como não foi obtido sucesso com em operar o modo API com XBee dificultando completar objetivo do trabalho descrito em 1.3.

A ideia das 3 condições é acionar os LEDs tornando o estado lógico da porta digital 30 do Arduino Mega 2560 representado pela variável `led`. Para evitar acionamento desnecessário foi criada a variável `smsAnterior`, do tipo string, para fazer um AND lógico com a variável `sms`. Esse AND lógico é representado pelos caracteres `&&`. No final do else a variável `smsAnterior` recebe o texto da variável `sms`, para que no próximo loop o AND lógico funcione. Quando a variável `sms` é igual a variável `smsAnterior` executa-se o comando AT `["AT+CMGD=1,4"]` pela função `[mySerial.println("AT+CMGD=1,4");]` e em seguida a variável `sms` é limpa.

Esse comando AT `["AT+CMGD=1,4"]` deleta todos os SMSs da memória disponível para armazenar os SMS's no SIM900, tanto mensagens lidas como não lidas. Esse comando AT está disponível no documento SIM900 AT Commands Manual_V1.07 [19] na Página 99 em AT Commands According to GSM07.05.

Slide 30/31

Para configurar o XBee, existe o software do fabricante, o XCTU. Entretanto, optou-se por fazer configuração por um outro software terminal serial. O software Tera Term.

Existem alternativas para configurar esse tipo de comunicação e pode ser usado em conjunto com o modo API, que é a principal modo de comunicação.

O módulo XBee Série 1 suporta um simples recurso chamado I/O line passing, que permite usar qualquer um dos 8 pinos (DIO 0 até 7) para detectar mudanças do estado lógico de Alta(~5v) para Baixa (~3v) do módulo XBee em um outro pino correspondente no módulo do XBee receptor. Basicamente espelhará o estado lógico, como criação de um “fio virtual” livre de qualquer apoio de microcontrolador externo ou de análise de série.

Slide 32

API operation

API (Application Programming Interface) Operation is an alternative to the default Transparent Operation. The frame-based API extends the level to which a host application can interact with the networking capabilities of the module.

When in API mode, all data entering and leaving the module is contained in frames that define operations or events within the module.

Transmit data frames (received through the DI pin (pin 3)) include:

- RF Transmit data frame
- Command frame (equivalent to AT commands)

Receive Data frames (sent out the DO pin (pin 2)) include:

- RF-received data frame
- Command response
- Event notifications such as reset, associate, disassociate, and so forth

The API provides alternative means of configuring modules and routing data at the host application layer. A host application can send data frames to the module that contain address and

payload information instead of using command mode to modify addresses. The module will send data frames

to the application containing status packets; as well as source, RSSI and payload information from received data packets.

The API operation option facilitates many operations such as the examples cited below:

- Transmitting data to multiple destinations without entering Command Mode
- Receive success/failure status of each transmitted RF packet
- Identify the source address of each received packet

To implement API operations, refer to API Operation for the

Slide 33/34/35/36

Segue algumas descrições dos principais parâmetros modificados da configuração padrão, como mostrado na tabela 6.1. Por padrão existem mais de 60 parâmetros.

1) Parâmetro: Número do Canal [CH]

Pode variar entre 0x0B – 0x1A. [16]

2) Parâmetro: Personal Area Network – [ID]

Pode variar entre 0x0 – 0xFFFF. [16]

3) Parâmetro: Destination Address High and Low Address [DH / DL]

Pode variar entre 0x0 – 0xFFFFFFFF. [16]

Esses parâmetros têm a função de endereçar o endereço de destino. Os 32 bits dos 64 bits superiores para o DH e os 32 bits inferiores para o DL do endereço de destino de 64 bits. Define a DH registrar a zero e DL menos de 0xFFFF para transmitir usando um endereço de 16 bits. 0x0000000000000000FFFF é o endereço de broadcast para o PAN. [16]

4) Parâmetro: Source Address [MY]

Pode variar entre 0x0 – 0xFFFF. [16]

Configura o endereço de origem de 16 bits. Define-se 0xFFFF para desativar a recepção de pacotes com endereços de 16 bits. Endereço de origem de 64 bits é o número de série e está sempre ativada. [16]

5) Parâmetro: Note Identifier [NI]

Pode variar entre 0 – 20 caracteres ASCII. [16]

Tem como função de identificar por String o módulo.

6) Parâmetro: Sleep Mode [SM]

Opções: No Sleep [0], Pin Hibernate [1], Pin Doze [2], Reserved [3], Cyclic Sleep Remote [4], Cyclic Sleep Remote w/ pin wakeup [5], *Sleep Coordinator-See descripton [6]. [16]

Pino Hibernação é reduz a energia economizando, Pino Doze fornece o wake up(despertador) mais rápido, Ciclo sono remoto com ou sem o pino wake up (despertador). Definição Coordenador sono é definido pelo parâmetro SM compatível com a versão 106 parâmetros SM 106. [16]

7) Parâmetro: Interface Data Rate [BD]

Opções: 1200 [0], 2400 [1], 4800 [2], 9600 [3], 19200 [4], 38400 [5], 57600 [6], 115200 [7]. [16]

Define a taxa de transmissão interface serial para a comunicação entre os modems seriais. [16]

8) Parâmetro: Modo API [AP]

I/O de dados é enviado para fora do UART usando um quadro(frame) de API. Todos os outros dados podem ser enviados e recebidos através da Operação transparente ou enquadramento(framing) API se o modo de API está habilitado (AP> 0). [16]

Operações API suportar dois RX (recepção) identificadores de quadros para dados de I/O (definir o endereço de 16 bits para 0xFFFFE e o módulo fará endereçamento de 64 bits). [16]

9) Parâmetro: DIOX Configuration [DX]

Para configurar as opções para os pinos, tabela 6.2, do módulo XBee.

As opções: conversor analógico para digital, entrada e saída digital.

Funções dos pinos e seus números PIN associado e comandos. [16]

AD = conversor analógico-digital, DIO = Digital Input/Output

Funções dos pinos não aplicáveis na presente seção são indicadas dentro de parênteses.

Slide 37

Comentar as dificuldades encontradas

Slide 38

Automação Industrial e Residencial - OS DEZ MAIORES DESAFIOS DA AUTOMAÇÃO INDUSTRIAL AS PERSPECTIVAS PARA O FUTURO

1) Formação Técnica de Profissionais e Educação da Sociedade quanto à Evolução Tecnológica Proporcionada pela Automação. [30]

Esse primeiro desafio traduz a dificuldade de no Brasil se conseguir projetar um sistema útil para sociedade brasileira, tanto pela dificuldade técnica, quanto pelo baixo desenvolvimento tecnológico e baixíssima educação da sociedade para usufruir dos sistemas.

2) Segurança e Confiabilidade em Sistemas Críticos. [30]

Esse desafio técnico influencia muito no desenvolvimento de qualquer projeto, pois aumentar a segurança inclui uma complexidade para garantir a confiabilidade no futuro sistema e principalmente nos cenários críticos que possa ocorrer que pode inviabilizar o projeto. A todo momento pode elevar o custo e forçar a revisão do planejamento.

3) Otimização de Informações, no Sentido de Fornecer uma Interface Homem-Máquina Apropriada. [30]

É um desafio que pode se tornar mais difícil devido à primeira situação; aliás, todos os desafios estão interligados e o mais importante é o primeiro desafio citado. O conhecimento técnico das linguagens HTML, PHP, CSS, JavaScript, AJAX, etc. ajudam a amenizar, melhorando a interface Homem-Máquina e favorecendo a interoperabilidade das interfaces.

4) Reconhecimento de Padrões. [30]

Esse desafio é o que garante a interoperabilidade das interfaces, a exemplo desse trabalho, que só foi viável pois foram relacionados os padrões para tal.

5) Identificação de Falhas em Sistemas de Automação. [30]

Faz parte do escopo de qualquer trabalho prever as falhas. Entretanto, sempre temos como desafio prever os possíveis falhas que desconhecemos, pois estas costumam aparecer durante a implementação do projeto e depois no funcionamento prático.

6) Comunicação Segura entre Dispositivos Heterogêneos. [30]

Esse desafio é o que melhor representa este trabalho, que se tornou viável graças às bibliotecas e ao hardware do Arduino, que abrem uma gama de possibilidades de desenvolvimento de inúmeros projetos e garante confidencialidade, autenticação, autorização e integridade entre Dispositivos Heterogêneos.

7) Sistemas de Automação Residencial. [30]

Sobre esse desafio será transcrito o trecho do artigo mencionado anteriormente.

Ao se falar em automação residencial, vários são os indicadores de muitas perspectivas de crescimento. O conceito de casa/escritório automatizado evoluiu bastante e hoje já não é mais visto com uma noção futurística e sim como uma opção de conforto, bem-estar e qualidade de vida ao permitir o controle da residência remotamente, economia de tempo e esforço com tarefas repetitivas e interação com usuários à distância. A própria evolução tecnológica contribuiu muito para que ideias que antes eram vistas como impraticáveis pudessem ser implementadas e hoje se vê muitos projetos ousados de automação residencial integrando vários dispositivos como computadores, celulares e a TV digital. Os avanços tecnológicos na área de microeletrônica baratearam os custos dos dispositivos, tornando fácil a disponibilidade de processadores, microcontroladores de tamanho reduzido, baixo consumo, bem como barramentos eficientes para melhoria de comunicação. A evolução da automação residencial passa atualmente por um período de migração de toda a tecnologia desenvolvida ao longo das décadas em automação predial e industrial, adequando-se à infraestrutura de uma residência. Dentre as funcionalidades a serem desenvolvidas e implementadas pelos sistemas de automação residencial podem-se destacar (ARAÚJO E PEREIRA, 2003) o gerenciamento, monitoramento e otimização do consumo de energia; conforto térmico através do controle HVAC (Heating, Ventilation and Air Conditioning); controle de iluminação, controle de acesso, monitoramento e segurança física de dispositivos; dispositivos de manutenção inteligente, entre outros. [30]

8) Gerência de Informações de Tempo Real. [30]

É um desafio relacionado à otimização da informação e inclui, também, a linguagem para banco de dados (MySQL). Na opinião do autor é o melhor exemplo, onde a automação residencial e industrial se complementam.

O desenvolvimento da automação ligado à evolução de tecnologias tem, na comunicação industrial, importante ligação com o crescimento da tecnologia da informação, principalmente na

relação entre os níveis gerencial e operacional de um ambiente de Automação Industrial. A TI (tecnologia da informação), que alterou bastante as hierarquias e estruturas no ambiente de escritório, passa agora a influenciar também o ambiente industrial, nos mais diversos níveis, desde indústrias de processos, prédios e sistemas logísticos, entrando até no nível de automação residencial. A gerência de informações industriais surgiu na medida em que, com o amadurecimento das tecnologias dos equipamentos de campo inteligentes, uma grande quantidade de informações possibilitadas por eles pode agora ser disponibilizada para outras aplicações em outros níveis de tecnologias, sendo possível organizá-las, por exemplo, em formato conveniente para análise no nível de gerência de processos industriais. Os modernos sistemas de gerência de informações web permitem facilitar a análise e interpretação dos dados obtidos pela comunicação com outras camadas do sistema de automação industrial. Implementar tais ferramentas, em ambiente industrial, em tempo real, é um grande desafio para a automação. [30]

9) Aplicações na Área de Medicina. [30]

Nesse desafio o autor discorda, pois acredita que não só na área de medicina mas em todas outras também enfrentamos diversos obstáculos no Brasil, como nos meios de transporte urbano em geral, indústria de eletroeletrônica, eletromecânica, componentes robotizados, etc. O que falta, como dito no artigo, em todas as áreas é integração das universidades e as empresas para produção de tecnologia nacional, assim como com empresas multinacionais no sentido de produção de artigos e pesquisas, diminuindo os obstáculos entre os mundos acadêmico e industrial.

10) Impactos Sociais e Ambientes Gerados pela Automação. [30]

O desafio de mensurar os impactos da tecnologia na sociedade ainda é muito recente. Inúmeros trabalhos estão sendo desenvolvidos nesse sentido, pois desde a década de 80 do século XX é notória a mudança que a sociedade vem sofrendo como consequência do fenômeno da globalização.

Slide 39

O projeto foi bem-sucedido no sentido de se desenvolver um protótipo plenamente funcional, demonstrando sua aplicabilidade a uma situação prática. Foi possível apresentar todos os conceitos envolvidos na proposta do tema do trabalho através de uma versão simplificada do sistema, no qual uma carga – no caso, uma lâmpada – liga e desliga por comandos SMS, responsáveis por acionar o relé ao qual ela está conectada, a qualquer distância dentro da área de cobertura 2G da operadora.

A arquitetura de rede proposta foi plenamente implementada, à exceção da comunicação entre o Arduino e PC. Uma vez estabelecida essa comunicação, torna-se possível verificar os dados das variáveis e criar variáveis adicionais, como contadores, etc. Nesse sentido, é possível, sim, pensar em criar uma gerência para verificar ao menos os eventos. Esse tipo de análise de viabilidade foi pesquisada e estudada antes da realização do presente trabalho, e entendemos ser este um objeto de estudo interessante para trabalhos futuros. Para implementar essa gerência, o requisito é realizar a comunicação do XBee receptor para o XBee transmissor, ou seja, a comunicação deve ser bidirecional. No protótipo que apresentamos, a comunicação se dá de forma unidirecional. Para isso somente no modo API.

Slide 46 - Apendice

Como no Brasil “não se faz Chip”, a única forma de tentar reproduzir o protocolo é um circuito ligado ao algum microcontrolador. Se fabricasse Chips com qualidade, poderia programar em FPGA e construir um microcontrolador. É uma alternativa viável, eu só vejo no momento essa forma. E mais daria uma flexibilidade que iria se destacar nos microcontroladores hoje no mercado porque se o projeto for muito bom implicaria em programações mais sofisticadas, algoritmos muito oportuno para resolver problemas que hoje pode estar presente. Porque uma vez que o microcontrolador é construído pode haver limitações desse hardware que dificultaria programar.

No CEFET-RJ na graduação em engenharia elétrica é usado o FPGA da Altera para aprender a VHDL, uma vez programado e simulado o futuro microcontrolador no FPGA pode pensar em fabricar esse microcontrolador com as especificações do projeto em FPGA. Garantindo assim uma comunicação M2M estável e segura.

Na prática os fabricantes de microcontroladores disponibilizam inúmeros modelos com características “um pouco distintas” para poder adequar aos projetos nos engenheiros. E é exatamente na minha visão o que acontece nesse trabalho. O Arduino Mega 2560 tem um microcontrolador com suas bibliotecas para adequar a todas as eletrônicas externas. Mas não é perfeito pois esse microcontrolador pode ter característica que impeçam programar alguns algoritmo.