



MARCEL SALVIONI DA SILVA

**REDE DE SENSORES SEM FIO DE BAIXO CUSTO PARA
MONITORAMENTO AMBIENTAL**

Campinas

2013



Universidade Estadual de Campinas
Faculdade de Engenharia Elétrica e de Computação

Marcel Salvioni da Silva

REDE DE SENSORES SEM FIO DE BAIXO CUSTO PARA MONITORAMENTO AMBIENTAL

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Faculdade de Engenharia Elétrica e de Computação da Universidade Estadual de Campinas para obtenção do título de Mestre em Engenharia Elétrica, na área de **Eletrônica, Microeletrônica e Optoeletrônica**.

Orientador: Prof. Dr. Fabiano Fruett

ESTE EXEMPLAR CORRESPONDE À VERSÃO FINAL DA DISSERTAÇÃO
DEFENDIDA PELO ALUNO MARCEL SALVIONI DA SILVA E ORIENTADO
PELO PROF. DR. FABIANO FRUETT

Campinas
2013

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca da Área de Engenharia e Arquitetura
Rose Meire da Silva - CRB 8/5974

Si38r Silva, Marcel Salvioni da, 1980-
Rede de sensores sem fio de baixo custo para monitoramento ambiental /
Marcel Salvioni da Silva. – Campinas, SP : [s.n.], 2013.

Orientador: Fabiano Fruett.
Dissertação (mestrado) – Universidade Estadual de Campinas, Faculdade de
Engenharia Elétrica e de Computação.

1. Redes de sensores sem fio. 2. Monitoramento ambiental - Sensoriamento
remoto. 3. Microcontroladores. 4. Sistemas embutidos de computador. I. Fruett,
Fabiano, 1970-. II. Universidade Estadual de Campinas. Faculdade de Engenharia
Elétrica e de Computação. III. Título.

Informações para Biblioteca Digital

Título em outro idioma: Low cost wireless sensor network for environmental monitoring

Palavras-chave em inglês:

Wireless sensor networks

Environmental monitoring - Remote sensing

Microcontrollers

Embedded systems

Área de concentração: Eletrônica, Microeletrônica e Optoeletrônica

Titulação: Mestre em Engenharia Elétrica

Banca examinadora:

Fabiano Fruett [Orientador]

Paulo Cardieri

Eddie Batista de Lima Filho

Data de defesa: 29-10-2013

Programa de Pós-Graduação: Engenharia Elétrica

COMISSÃO JULGADORA - TESE DE MESTRADO

Candidato: Marcel Salvioni da Silva

Data da Defesa: 29 de outubro de 2013

Título da Tese: "Rede de Sensores sem Fio de Baixo Custo para Monitoramento Ambiental"

Prof. Dr. Fabiano Fruett (Presidente): _____

Prof. Dr. Eddie Batista de Lima Filho: _____

Prof. Dr. Paulo Cardieri: _____

Resumo

Existem diversos sistemas e equipamentos, com finalidades específicas, envolvendo o monitoramento das mudanças de clima e de tempo. Entretanto, a maioria desses sistemas e equipamentos disponíveis no mercado possuem alto custo, o que normalmente inviabiliza sua utilização em larga escala. Além disso, a maioria dos sistemas e equipamentos para este fim são importados, o que dificulta ainda mais a sua inserção em aplicações corriqueiras. Este trabalho descreve o desenvolvimento, projeto, testes e aplicação de uma rede de sensores sem fio de baixo custo, para monitoramento em tempo real da temperatura ambiente, umidade relativa do ar e pressão atmosférica, como uma alternativa nacional para pesquisas que necessitem de equipamentos para monitoramento ambiental. A rede é composta de quatro nós sensores e um nó coordenador. Os nós sensores possuem um sensor de temperatura e umidade e um sensor de pressão barométrica – ambos com interface digital – além de um microcontrolador de baixo consumo de energia e baixo custo, e um módulo de comunicação para uma rede operando na frequência de 2,4GHz, para conexão numa rede padrão IEEE 802.15.4. O nó coordenador possui outro módulo de comunicação para a conexão na rede sem fio, um microcontrolador, um cartão de memória para armazenamento das informações de cada nó sensor e interface Ethernet para disponibilizar as medidas do sensor em uma página *web*, acessada através de um IP estático em um browser de internet comum. Testes em laboratório e em campo atestaram autonomia do nó sensor (em torno de 90 dias para uma bateria de 500mAh) e alcance (de aproximadamente 150m em visada direta) e comprovaram a conformidade dos valores medidos por cada nó, através da comparação com um sistema de referência.

Palavras-chave: *Rede de sensores sem fio, Monitoramento ambiental – Sensoriamento remoto, Microcontroladores, Sistemas embutidos de computador.*

Abstract

There are several systems and equipments with specific applications related to climate and weather change monitoring. However, most of these systems and equipments available in the market are too expensive – which usually prevents their use in large scale. Furthermore, most of them used for such purpose are imported, which further complicates their insertion in everyday applications. This work describes the development, implementation and application of a low-cost wireless sensor network for temperature, relative humidity and atmospheric pressure monitoring, in real-time as a national alternative for researches which demand equipments for environmental monitoring. The network consists of four sensor nodes and a coordinator node. Sensor nodes have a temperature and humidity sensor and a barometric sensor – both with digital interface – plus a low cost and low power microcontroller and a communication module for the wireless network operating at 2.4GHz for establishing connection to an IEEE 802.15.4 network. Coordinator node has another communication module to the wireless network connection, a microcontroller for receiving measurements, a memory card for storage of each sensor node's information, and Ethernet interface to provide sensor node's information in an embedded webpage, accessed through a static IP in a common internet browser. Tests performed in laboratories and in the field attested the node's autonomy (around 90 days for a 500mAh battery) and reach (approximately 150m line-of-sight), and proved the compliance of measurements performed by each sensor node, through comparison with a reference system.

Keywords: *Wireless sensor networks, Environmental monitoring – Remote sensing, Microcontrollers, Embedded systems.*

Sumário

LISTA DE FIGURAS.....	XIX
LISTA DE TABELAS.....	XXI
LISTA DE ABREVIATURAS E SIGLAS	XXIII
INTRODUÇÃO.....	1
1.1 APRESENTAÇÃO DO PROBLEMA.....	1
1.2 OBJETIVOS.....	4
1.2.1 Objetivo Geral	4
1.2.2 Objetivos específicos.....	5
1.3 ORGANIZAÇÃO DA DISSERTAÇÃO	5
REVISÃO DE LITERATURA	7
2.1 MEDIDAS AMBIENTAIS – BREVE HISTÓRICO	7
2.2 SENSORES.....	11
2.2.1 Sensores de Temperatura	12
2.2.2 Sensores de Umidade.....	15
2.2.3 Sensores de Pressão Atmosférica	17
2.3 MICROCONTROLADORES	18
2.3.1 Microprocessador.....	18
2.3.2 Memórias	20
2.3.3 Dispositivos de entrada e saída.....	21
2.3.4 Temporizadores / Contadores.....	25
2.3.5 Blocos Internos de Aplicações Diversas	25
2.4 COMUNICAÇÃO E REDES SEM FIO	26

2.4.1	Topologias e Roteamento de Redes Sem Fio	29
2.4.2	Tecnologia ZigBee.....	31
2.4.3	Circuitos e módulos de radiofrequência comerciais	33
2.5	MODOS DE ALIMENTAÇÃO.....	35
2.6	CONCLUSÃO DA REVISÃO DE LITERATURA.....	38
	DESENVOLVIMENTO DE HARDWARE E SOFTWARE	39
3.1	REQUISITOS DO SISTEMA	39
3.1.1	Microcontrolador	40
3.1.2	Sensor de Temperatura Ambiente e Umidade Relativa do Ar	43
3.1.3	Sensor de Pressão Atmosférica.....	44
3.1.4	Módulo de comunicação sem fio	47
3.1.5	Microcontrolador do coordenador da rede	48
3.1.6	1º protótipo do nó sensor	49
3.1.7	2º protótipo do nó sensor	50
3.1.8	3º protótipo do nó sensor	53
3.1.9	Baterias e Circuito Carregador de Baterias	54
3.2	MODO DE OPERAÇÃO DA REDE DE SENSORES.....	56
3.2.1	Operação do nó	56
3.2.2	Operação do coordenador	59
	RESULTADOS EXPERIMENTAIS	63
4.1	TESTES DOS NÓS SENSORES.....	64
4.1.1	Comunicação e alcance	64
4.1.2	Calibração	75
4.1.3	Autonomia.....	87
4.2	TESTES DO NÓ COORDENADOR.....	88
4.2.1	Recepção e Gravação das medidas	89
4.2.2	Conexão em Rede via Ethernet e Acesso Simultâneo	90
4.2.3	Gravação dos dados no Cartão de Memória.....	92
4.3	CONCLUSÕES.....	92
	CONCLUSÕES	95

5.1	CONCLUSÕES SOBRE COMUNICAÇÃO E ALCANCE.....	95
5.2	CONCLUSÕES SOBRE A AUTONOMIA.....	97
5.3	CONCLUSÕES SOBRE A CALIBRAÇÃO	97
5.4	CONCLUSÕES SOBRE O ACESSO REMOTO	98
5.5	CUSTO ESTIMADO E COMPARAÇÕES COM SISTEMAS EXISTENTES.....	100
5.6	SUGESTÕES DE TRABALHOS FUTUROS	103
	REFERÊNCIAS BIBLIOGRÁFICAS	105
	APÊNDICE A FOLHA DE DADOS DO SENSOR DE TEMPERATURA E UMIDADE - SENSIRION.....	109
A.1	PERFORMANCE DO SENSOR [53].....	109
A.1.1	Temperatura.....	109
A.1.2	Umidade relativa	110
	APÊNDICE B FOLHA DE DADOS DO SENSOR DE PRESSÃO ATMOSFÉRICA- FREESCALE.....	113
B.1	CARACTERÍSTICAS DE OPERAÇÃO [54].....	113
	APÊNDICE C MEDIDAS EM CÂMARA CLIMÁTICA	115
C.1	COMPARAÇÃO A 25°C	115
C.2	COMPARAÇÃO A 15°C	117
C.3	COMPARAÇÃO A 80%	118
C.4	COMPARAÇÃO A 40%	120
	APÊNDICE D CÓDIGOS-FONTE	123
D.1	CÓDIGO-FONTE PRINCIPAL DO NÓ SENSOR.....	123
D.2	CÓDIGO-FONTE DO NÓ COORDENADOR.....	129
D.3	CÓDIGO-FONTE DA PÁGINA WEB EMBARCADA	138

À Silvia.
A Albani e José Otávio.
A Lirna e Andrei.

Em memória de Rosita, Hildete, Augusto e José.

Agradecimentos

A Deus, Criador do Céu e da Terra, e do meio ambiente que observamos, pela vida, a brisa suave no caminho da Unicamp, a paz nas madrugadas de volta para casa e a segurança em cada viagem, seja pela terra ou pelo ar.

Ao Prof. Dr. Fabiano Fruett, pela paciência, orientação, confiança e apoio presentes por todo o tempo de projeto.

À Universidade Estadual de Campinas – UNICAMP, pela oportunidade de participar do Mestrado em Engenharia Elétrica na Faculdade de Engenharia Elétrica e Computação – FEEC.

Ao Centro de Ciência, Tecnologia e Inovação do Pólo Industrial de Manaus – CT-PIM – pela confiança, apoio logístico e financeiro, por todo o tempo de permanência em Campinas, fundamentais para a realização deste trabalho, bem como ao seu Laboratório de Confiabilidade e Caracterização de Circuitos Integrados – LCC-CI – pelo apoio técnico nos testes em laboratório.

À Silvia, minha esposa e companheira, pela paciência, confiança, compreensão, amor, carinho e incentivo, hoje e sempre.

A meus pais, irmãos, tios, cunhados, por todo o carinho e apoio sempre presentes.

À NXP, pelo apoio técnico e tecnológico.

Ao Laboratório de Sensores Microeletrônicos – LSM – pelo apoio operacional e técnico.

Ao Experimento de Larga Escala na Biosfera-Atmosfera da Amazônia – LBA – do Instituto Nacional de Pesquisas da Amazônia – INPA – por todo o apoio técnico para realização de testes em campo e em laboratório, fundamentais para embasar nossos objetivos e resultados.

Ao Departamento de Meteorologia da Universidade do Estado do Amazonas – UEA – pelo apoio nos testes em campo com balão atmosférico.

A todos meus amigos, colegas de trabalho e de laboratório, pelo carinho e apoio.

Lista de Figuras

Figura 1 – Vista da floresta a partir de uma torre instrumentada (Manaus).	2
Figura 2 – Desenhos de Leonardo Da Vinci, descrevendo o funcionamento do higrômetro.	8
Figura 3 – (a)Termoscópio de Galileu e (b)termômetros de Fahrenheit.	9
Figura 4 – Barômetro de Torricelli.	10
Figura 5 – Exemplos de conversão de energia realizada pelos sensores.	11
Figura 6 – Efeito Seebeck: uma tensão ΔV é gerada a partir de uma diferença de temperatura ΔT .	12
Figura 7 – Circuito de geração do ΔV_{BE} (a) e seção transversal de transistor com r -emissores (b).	13
Figura 8 – Circuito de geração da tensão de referência V_{ref} .	15
Figura 9 – Sensor de Umidade Capacitivo: detalhe (a) e símbolo equivalente (b).	16
Figura 10 – Tipos de medida da pressão.	17
Figura 11 – Esquemático (a) e fotografia (b) de um sensor de pressão piezoresistivo.	17
Figura 12 – Arquiteturas: Von Neumann(a) e Harvard (b).	18
Figura 13 – Exemplo de uso da MMU para controle de memórias <i>cache</i> lógicas.	21
Figura 14 – Diagrama em blocos das camadas OSI abordadas pelos padrões 802.	29
Figura 15 – Topologia em estrela.	29
Figura 16 – Topologia em árvore.	30
Figura 17 – Topologia em malha (ou <i>mesh</i>).	30
Figura 18 – Visão geral das camadas de rede da tecnologia ZigBee.	33
Figura 19 – Transceptores 802.15.4 da Freescale (a) e Micrel (b).	34
Figura 20 – Nó sensor comercial (mote) MICA.	34
Figura 21 – Módulo de Comunicação X-Bee.	35
Figura 22 – Baterias de íons de lítio (Li-Ion).	36
Figura 23 – Curva típica de tensão de saída da bateria por estado de carga.	36
Figura 24 – Painel solar.	36
Figura 25 – Exemplos de microgeradores.	37
Figura 26 – Exemplos de uso de alimentação por RF.	37
Figura 27 – Circuito Integrado do sensor SHT15 (a); detalhe do sensor de umidade (b).	44
Figura 28 – Diagrama interno e conexão do sensor de pressão atmosférica MPL115A2.	45
Figura 29 – Imagem microscópica da membrana microfabricada do sensor de pressão MPL115A2.	46
Figura 30 – Módulo de comunicação sem fio Xbee-Pro.	47
Figura 31 – Primeiro protótipo do nó sensor em placa genérica.	49
Figura 32 – Diagrama em blocos do nó sensor.	51
Figura 33 – Leiaute da placa de circuito impresso do 2º protótipo do nó sensor.	51
Figura 34 – Esquema elétrico do 2º protótipo do nó sensor.	52
Figura 35 – Detalhes da construção do 2º protótipo do nó sensor já montado.	52
Figura 36 – 3º protótipo do nó sensor (a) com detalhe da placa de sensores (b).	53
Figura 37 – Leiaute da placa principal do nó (a) e da placa de sensores (b) do 3º protótipo.	53
Figura 38 – Descrição das etapas de carga da bateria.	55
Figura 39 – Diagrama esquemático do carregador de baterias utilizando o LPC1114.	55
Figura 40 – Circuito implementado para carga da bateria utilizando a placa LPCXpresso.	56
Figura 41 – Diagrama de exemplo de comunicação entre os nós sensores e coordenador.	58
Figura 42 – Diagrama em blocos do nó coordenador da rede.	59
Figura 43 – Página <i>web</i> embarcada com as leituras e os dados dos nós disponibilizados.	61
Figura 44 – Protótipos posicionados na base (a) e no topo (b) da torre.	65
Figura 45 – Tela do navegador obtida durante os testes na torre da estrada ZF3.	65
Figura 46 – Protótipos do sensor (a) e do coordenador (b) na base da torre.	68

Figura 47 – Temperatura da base e do topo da torre medidas com o 1º protótipo durante teste na torre ZF2.	68
Figura 48 – Umidade Relativa da base e do topo da torre medidas com o 1º protótipo durante o teste na torre ZF2..	69
Figura 49 – Temperatura medida ao longo da torre ZF2 durante teste pelos sensores da torre	69
Figura 50 – Temperatura medida ao longo da torre ZF2 durante teste pelos sensores da torre a 5m (a) e 51m (b).....	70
Figura 51 – Equipe de Meteorologia da UEA preparando medidas com o balão.	71
Figura 52 – Temperatura e Pressão Barométrica medidos pelo nó sensor durante a subida do balão.	72
Figura 53 – Umidade e Pressão Barométrica medidos pelo nó sensor durante a subida do balão.	72
Figura 54 – Comparação dos valores de temperatura medidos pelo protótipo do nó sensor e pela sonda.	73
Figura 55 – Comparação dos valores de umidade medidos pelo protótipo do nó sensor e pela sonda.	73
Figura 56 – Comparação dos valores de pressão atmosférica medidos pelo protótipo e pela sonda.....	73
Figura 57: Curvas de Configuração de Temperatura e Umidade da Câmara Climática.....	76
Figura 58 – Disposição dos sensores na câmara (a) e detalhe da proximidade dos sensores (b).	76
Figura 59 – Temperatura do Protótipo e Sensor de Referência na posição 1.....	77
Figura 60 – Temperatura do Protótipo e Sensor de Referência na posição 2.....	77
Figura 61 – Temperatura do Protótipo e Sensor de Referência na posição 3.....	77
Figura 62 – Temperatura do Protótipo e Sensor de Referência na posição 4.....	78
Figura 63 – Temperatura do Protótipo e Sensor de Referência na posição 1, a 55°C.....	78
Figura 64 – Temperatura do Protótipo e Sensor de Referência na posição 2, a 55°C.....	79
Figura 65 – Temperatura do Protótipo e Sensor de Referência na posição 3, a 55°C.....	79
Figura 66 – Temperatura do Protótipo e Sensor de Referência na posição 4, a 55°C.....	79
Figura 67 – Umidade Relativa do Protótipo e Sensor de Referência na posição 1.	81
Figura 68 – Umidade Relativa do Protótipo e Sensor de Referência na posição 2.	81
Figura 69 – Umidade Relativa do Protótipo e Sensor de Referência na posição 3.	81
Figura 70 – Umidade Relativa do Protótipo e Sensor de Referência na posição 4.	82
Figura 71 – Umidade Relativa do Protótipo e Sensor de Referência na posição 1, a 65%.	82
Figura 72 – Umidade Relativa do Protótipo e Sensor de Referência na posição 2, a 65%.	83
Figura 73 – Umidade Relativa do Protótipo e Sensor de Referência na posição 3, a 65%.	83
Figura 74 – Umidade Relativa do Protótipo e Sensor de Referência na posição 4, a 65%.	83
Figura 75 – Pressão Atmosférica do Protótipo e Sensor de Referência na posição 1, em hPa.....	85
Figura 76 – Pressão Atmosférica do Protótipo e Sensor de Referência na posição 2, em hPa.....	86
Figura 77 – Pressão Atmosférica do Protótipo e Sensor de Referência na posição 3, em hPa.....	86
Figura 78 – Pressão Atmosférica do Protótipo e Sensor de Referência na posição 4, em hPa.....	86
Figura 79 – Resultados dos Testes de Autonomia.	88
Figura 80 – Tela do navegador obtida durante os primeiros testes com o nó coordenador.....	89
Figura 81 – Tela do navegador com as implementações finais da página <i>web</i>	90
Figura 82 – Exemplo de arquivo texto que armazena os dados de cada sensor.	92
Figura 83 – (a) Mote sem fio <i>Wasp mote</i> e (b) placa de sensores de gás.	101
Figura 84 – Coordenador <i>Meshlilum</i>	102
Figura 85 – Máxima Tolerância do valor de temperatura para o sensor SHT15.....	110
Figura 86 – Máxima tolerância do valor de umidade relativa a 25 C para o sensor SHT15.	111
Figura 87 – Temperatura do Protótipo e Sensor de Referência na posição 1, a 25°C.....	115
Figura 88 – Temperatura do Protótipo e Sensor de Referência na posição 2, a 25°C.....	116
Figura 89 – Temperatura do Protótipo e Sensor de Referência na posição 3, a 25°C.....	116
Figura 90 – Temperatura do Protótipo e Sensor de Referência na posição 4, a 25°C.....	116
Figura 91 – Temperatura do Protótipo e Sensor de Referência na posição 1, a 15°C.....	117
Figura 92 – Temperatura do Protótipo e Sensor de Referência na posição 2, a 15°C.....	117
Figura 93 – Temperatura do Protótipo e Sensor de Referência na posição 3, a 15°C.....	118
Figura 94 – Temperatura do Protótipo e Sensor de Referência na posição 4, a 15°C.....	118
Figura 95 – Umidade Relativa do Protótipo e Sensor de Referência na posição 1, a 80%.	119
Figura 96 – Umidade Relativa do Protótipo e Sensor de Referência na posição 2, a 80%.	119
Figura 97 – Umidade Relativa do Protótipo e Sensor de Referência na posição 3, a 80%.	119
Figura 98 – Umidade Relativa do Nó Protótipo e Sensor de Referência na posição 4, a 80%.	120
Figura 99 – Umidade Relativa do Protótipo e Sensor de Referência na posição 1, a 40%.	120
Figura 100 – Umidade Relativa do Protótipo e Sensor de Referência na posição 2, a 40%.	121
Figura 101 – Umidade Relativa do Protótipo e Sensor de Referência na posição 3, a 40%,	121
Figura 102 – Umidade Relativa do Protótipo e Sensor de Referência na posição 4, a 40%.	121

Lista de Tabelas

Tabela I – Comparação entre desempenhos da arquiteturas 8051 e Cortex-M0	20
Tabela II – Principais Faixas de frequência ISM.	27
Tabela III – Tipos de dispositivos em redes ZigBee.	32
Tabela IV – Relação entre porcentagem do nível do sinal e seu valor acima da sensibilidade (em dBm).....	47
Tabela V – Detalhamento da codificação de dados do nó.....	58
Tabela VI – Código das cores para o estado da bateria no nó.....	62
Tabela VII – Valores de temperatura e umidade da torre na ZF3.	66
Tabela VIII – Valores de referência para intervalo mínimo entre medidas.	67
Tabela IX – Comparação da Temperatura Média dos Nós Sensores e Sensores de Referência a 55°C.....	80
Tabela X – Comparação da Temperatura Média dos Protótipos e Sensores de Referência a 25°C	80
Tabela XI – Comparação da Temperatura Média dos Protótipos e Sensores de Referência a 15°C.....	80
Tabela XII – Comparação da Umidade Relativa Média dos Protótipos e Sensores de Referência a 65%	84
Tabela XIII – Comparação da Umidade Relativa Média dos Protótipos e Sensores de Referência a 80%.....	84
Tabela XIV – Comparação da Umidade Relativa Média dos Protótipos e Sensores de Referência a 40%	85
Tabela XV – Comparação da Pressão Atmosférica Média dos Protótipos e Sensores de Referência em hPa	87
Tabela XVI – Valores Comerciais para o Nó Sensor.....	100
Tabela XVII – Valores Comerciais para o Nó Coordenador.	100
Tabela XVIII – Estimativa de custo do Nó Sensor da <i>Libelium</i>	101
Tabela XIX – Estimativa de custo do Nó Coordenador da <i>Libelium</i>	102
Tabela XX – Comparação entre os valores do sistema proposto e do sistema da <i>Libelium</i>	102
Tabela XXI – Especificações de Temperatura para o sensor SHT15.....	109
Tabela XXII – Especificações de Umidade Relativa para o sensor SHT15.....	110
Tabela XXIII – Especificações de Pressão para o sensor MPL115A2.....	113
Tabela XXIV – Especificações de Temperatura para o sensor MPL115A2.	114

Lista de Abreviaturas e Siglas

LBA – *Large-Scale Biosphere-Atmosphere Experiment in Amazonia*

HTML – *Hipertext Markup Language*

SAW – *Surface-Acoustic-Wave*

RTD – *Resistance-temperature Detector*

PTAT – *Proportional To Absolute Temperature*

MEMS – *Micro-Eleto-Mechanical Systems*

CISC – *Complex Instruction Set Computer*

RISC – *Reduced Instruction Set Computer*

MIPS – *Million Instructions Per Second*

MMU – *Memory Management Unit*

ARM – *Advanced RISC Machines*

ROM – *Read-Only Memory*

EEPROM – *Electrically Erasable Read-Only Memory*

RAM – *Random Access Memory*

SRAM – *Static Random Access Memory*

DRAM – *Dynamic Random Access Memory*

EMC – *External Memory Controller*

GPIO – *Generic Purpose Input/Output*

ADC – *Analog-to-Digital Converter*

DAC – *Digital-to-Analog Converter*

UART – *Universal Asynchronous Receiver-Transmitter*

I²C – *Inter-Integrated Circuits*

SCL – *Serial Clock*

SDA – *Serial Data*

SPI – *Serial Protocol Interface*

MISO – *Master Input, Slave Output*

MOSI – *Master Output, Slave Input*

SL – *Slave Select*

CAN – *Controller Area Network*

OSI – *Open Systems Interconnection*

CDSA/CD – *Carrier Sense Multiple Access with Collision Detection*

SD/MMC – *Secure Digital/MultiMedia Card*

LAN – *Local Area Network*

WAN – *Wide Area Network*

IEEE – *Institute of Electrical and Electronics Engineers*

MAC – *Medium Access Control*

MII – *Media Independent Interface*

RMII – *Reduced Media Independent Interface*

IP – *Internet Protocol*

TCP – *Transmission Control Protocol*

UDP – *User Datagram Protocol*

TLS – *Transport Layer Security*

SSL – *Secure Sockets Layer*

HTTP – *Hyper Text Transfer Protocol*

SMTP – *Simple Mail Transfer Protocol*

FTP – *File Transfer Protocol*

USB – *Universal Serial Bus*

HID – *Human Interface Device*

UVC – *Universal Video Class*

CDC – *Communication Device Class*

MSD – *Mass Storage Device*

HCI – *Host Controller Interface*

OHCI – *Open Host Controller Interface*

UHCI – *Universal Host Controller Interface*

EHCI – *Enhanced Host Controller Interface*

LS – *Low-Speed*

FS – *Full-Speed*

HS – *High-Speed*

SP – *Super-Speed*

PWM – *Pulse Width Modulation*

PLL – *Phase Locked Loop*

RTC – *Real Time Clock*

BOD – *Brownout Detector*

PMU – *Power Management Unit*

AMPS – *Advanced Mobile Phone System*

GSM – *Groupe Special Mobile*

TDMA – *Time Division Multiple Access*

CDMA – *Code Division Multiple Access*

MBWA – *Mobile Broadband Wireless Access*

AM – *Amplitude Modulation*

FM – *Frequency Modulation*

PAL – *Phase Alternate Lock*

NTSC – *National Television Standard Committee*

SECAM – *Sequential Color and Memory System*

DAB – *Digital Audio Broadcasting*

DVB - *Digital Video Broadcasting*

DTTB – *Digital Terrestrial Television Broadcasting*

ISM – *Industrial, Scientific and Medical*

VSAT – *Very Small Aperture Terminal*

IrDA – *Infrared Data Association*

PAN – *Personal Area Network*

MAN – *Metropolitan Area Network*

RAN – *Regional Area Network*

VHF – *Very High Frequency*

UHF – *Ultra High Frequency*

LLC – *logical link control*

MANET – *Mobile Ad-Hoc Network*

AODV – *Ad-Hoc On Demand Distance Vector*

DYMO – *Dynamic MANET On Demand*

DSR – *Dynamic Source Routing*

DHT – *Distributed Hash Table*

GHT – *Geographic Hash Tables*

GPSR – *Greedy Perimeter Stateless Routing*

GRWLI – *Geographic Routing Without Location Information*

VRR – *Virtual Ring Routing*

VCP – *Virtual Cord Protocol*

LR-WPAN – *Low Rate – Wireless Personal Area Network*

AES – *Advanced Encryption Standard*

FFD – *Full Function Device*

RFD – *Reduced Function Device*

ZC – *ZigBee Coordenador*

ZR – *ZigBee Router*

ZED – *ZigBee End Device*

COTS – *Commercial-Off-The-Shelf*

API – *Application Programming Interface*

MCI – *Memory Card Interface*

FAT – *File Allocation Table*

LSM – Laboratório de Sensores Microeletrônicos

FEEC – Faculdade de Engenharia Elétrica e de Computação

UNICAMP – Universidade Estadual de Campinas

LCC-CI – Laboratório de Caracterização e Confiabilidade de Circuitos Integrados

CT-PIM – Centro de Ciência, Tecnologia e Inovação do Pólo Industrial de Manaus

SUFRAMA – Superintendência da Zona Franca de Manaus

INPA – Instituto Nacional de Pesquisas da Amazônia

EMBRAPA – Empresa Brasileira de Pesquisa Agropecuária

UEA – Universidade do Estado do Amazonas

RSSF – Rede de Sensores Sem Fio

DHCP – *Dynamic Host Configuration Protocol*

Capítulo 1

Introdução

Nos tempos atuais, o meio ambiente se tornou uma das principais preocupações da sociedade, em todos os seus níveis. Essa tendência deve-se principalmente à influência cada vez maior das mudanças climáticas na economia e no modo de vida das pessoas. Eventos como secas e enchentes se tornam cada vez mais frequentes, causando dor, morte e prejuízos materiais.

As temperaturas médias do planeta, a presença abundante de água e a pressão da atmosfera, constituída basicamente por gases como nitrogênio, oxigênio e dióxido de carbono – essenciais para a respiração e fotossíntese – são fatores chave para que haja vida e que a mesma evolua. Após a Revolução Industrial, observam-se algumas alterações no clima e na proporção da concentração desses gases na atmosfera. A partir deste momento histórico, mudanças ambientais ficaram mais acentuadas e as estações do ano já não refletem como antes seu comportamento regular [1].

1.1 Apresentação do problema

Com as mudanças de clima e de tempo, o monitoramento das variáveis de ambiente se torna uma ferramenta essencial para se compreender os efeitos das mudanças e permite a adoção de ações preditivas, adaptativas e corretivas. Para esta finalidade, são utilizadas diversas abordagens. A mais conhecida e mais ampla é o monitoramento das condições ambientais

através de imagens de satélite. Através delas, é possível realizar análise de temperatura, movimento de massas de ar, possibilidade de chuvas, focos de incêndio e queimadas. Para o monitoramento mais apurado de variáveis locais como temperatura ambiente, pressão atmosférica e umidade, contudo, as informações de satélites não são suficientes, sendo necessárias medidas a partir de sensores instalados no próprio local de interesse. Para a obtenção destas medidas, são utilizadas estações meteorológicas que, além de fornecer informações sobre as variáveis citadas, ainda permitem obter informações sobre quantidade de materiais particulados (como fuligem), dióxido e monóxido de carbono, direção e velocidade do vento, radiação solar, entre outros.



Figura 1 – Vista da floresta a partir de uma torre instrumentada (Manaus).

Outro aspecto relevante é o impacto das mudanças climáticas sobre os grandes biomas terrestres. Um desses biomas é a Floresta Amazônica. A maior floresta tropical do mundo é objeto de estudo de vários campos da ciência, incluindo-se aí a meteorologia. Sua biodiversidade e dimensão a transformam num gigantesco laboratório natural. Estudar como a floresta se comporta e “funciona” ajuda a entender como o seu clima influencia as condições meteorológicas do planeta e vice-versa. Um dos questionamentos a serem respondidos pelas pesquisas da área de meteorologia é como se comporta o dióxido de carbono na floresta e qual o seu impacto na formação de biomassa [2]. Na busca de resposta para esse e outros

questionamentos, são aplicados vários instrumentos aliados a várias técnicas e tecnologias. Um dos programas científicos que realizam estas pesquisas é o Experimento de Larga Escala na Biosfera-Atmosfera na Amazônia (*Large-Scale Biosphere-Atmosphere Experiment in Amazonia* - LBA). Para colher seus resultados, os projetos deste programa aliam imagens de satélite a medições feitas com sensores de diversos tipos para aquisição de temperatura e umidade relativa do ar, umidade e temperatura do solo, radiação incidente do Sol, radiação absorvida, velocidade e direção do vento, nível de CO₂ e pressão atmosférica, entre outros dados geralmente colhidos durante vários dias sem interrupção.

Para experimentos como esses descritos, há a necessidade de se utilizar instrumentos de precisão com tecnologia de ponta. Sem alternativas nacionais, tais projetos normalmente precisam recorrer a importação de tecnologia com alto valor agregado para conseguir obter os resultados esperados. Isso pressiona os gestores dos projetos a reservar uma importante parte do seu orçamento à importação de equipamentos. Os recursos dirigidos a estas importações acabam, de forma indireta, fortalecendo e estimulando empresas e instituições estrangeiras, sem contribuir significativamente para o desenvolvimento tecnológico brasileiro. Desta forma, este investimento poderia ser usado para alavancar o desenvolvimento tecnológico nacional na área de instrumentação meteorológica. Para disponibilizar uma alternativa tecnológica a estas e outras aplicações, foi desenvolvida uma rede de sensores sem fio de baixo custo para ser aplicada a monitoramento ambiental. Ele é construído a partir de componentes de baixo custo, com alta integração e desenvolvidos para operar com baixo consumo de energia. O sistema tem como inspiração os sistemas já empregados nas pesquisas em curso, agregando, porém, recursos como monitoramento remoto e comunicação sem fio, permitindo ao sistema ser aplicado nas pesquisas atuais e em novas pesquisas que precisam de instrumentos com tais características.

A tecnologia de redes de sensores sem fio é um campo bastante explorado pela área de instrumentação e medidas por combinar sensoriamento, computação e comunicação em um único e pequeno dispositivo [3]. Na maioria das aplicações, a capacidade de processamento dos dispositivos é baixa autonomia de funcionamento. Porém, o uso de dezenas (e até centenas) destes dispositivos integrados em rede oferece possibilidades tecnológicas realmente novas. O poder das redes reside na habilidade de agregar um grande número de pequenos sensores que

se organizam e se configuram independentemente. Os custos na instalação deste tipo de sistema tendem a ser minimizados, uma vez que estruturas como tubulações, cabos e fios não são necessárias, permitindo que uma dada área possa ser facilmente monitorada por centenas de sensores que automaticamente formam uma rede de interconexão e informam os dados sendo medidos. Um aspecto que reforça o uso das redes de sensores sem fio é o avanço no desenvolvimento de microsensores para medição de diversas grandezas (pressão, temperatura, umidade, fluxo de ar e calor, etc.) e detecção de elementos químicos (pH, CO₂, O₃, CH₃, gases tóxicos, etc). Este desenvolvimento é reflexo dos avanços nas pesquisas com novos materiais e também das técnicas e métodos para microfabricação de sensores microeletrônicos. Os sensores microeletrônicos em especial realizam medidas com exatidão com um baixo consumo de energia.

A rede sem fio proposta neste trabalho é voltada para a aquisição de temperatura ambiente, umidade relativa do ar e pressão atmosférica. Uma rede com quatro nós realizam essas medidas e as transmitem ao coordenador da rede que é responsável por receber, identificar e armazenar as medidas, referenciando-as no tempo. Ao mesmo tempo, o coordenador permite acesso via rede a uma página web contendo as últimas leituras dos quatro nós, permitindo a observação e o monitoramento do ambiente sendo monitorado e do estado da rede – com informações sobre estado da bateria, nível do sinal recebido dos nós sensores pelo coordenador e estado do relógio.

1.2 Objetivos

1.2.1 OBJETIVO GERAL

- ✓ Desenvolver uma rede de sensores sem fio para monitoramento ambiental que apresente um baixo custo ao ser comparado a sistemas comerciais existentes.

1.2.2 OBJETIVOS ESPECÍFICOS

- ✓ Desenvolver um nó da rede de sensores capaz de efetuar medidas de temperatura, pressão atmosférica e umidade relativa do ar, empregando sensores microeletrônicos, comunicação robusta e otimização de bateria;
- ✓ Desenvolver um coordenador da rede de sensores que seja preparado para se comunicar com quatro nós sensores, receber seus dados, classificá-los, gravá-los em uma memória física e disponibilizar uma página web em HTML (*HiperText Markup Language*) para visualização dos dados remotamente;
- ✓ Realizar testes e ensaios para avaliação do rendimento da rede de sensores quanto à autonomia de energia, precisão das medidas, faixa de operação e alcance da rede.

1.3 Organização da dissertação

Esta dissertação está organizada em seis capítulos, incluindo esta introdução. Os demais capítulos são descritos resumidamente a seguir:

- ✓ Capítulo 2: revisão de literatura realizada para o projeto e desenvolvimento da rede, descrevendo fundamentos técnicos e tecnológicos que compõem o sistema, os atuais sistemas de medição e um breve histórico sobre medidas ambientais.
- ✓ Capítulo 3: descrição do projeto, da arquitetura e o desenvolvimento da rede de sensores sem fio proposto.
- ✓ Capítulo 4: descrição dos testes realizados e apresentação dos resultados, conforme descrito nos objetivos do trabalho, assim como a descrição dos diversos ambientes utilizados para obtenção das medidas e caracterização do sistema.
- ✓ Capítulo 5: conclusões face aos resultados atingidos e sugestões de futuras implementações e melhorias.

Capítulo 2

Revisão de Literatura

NESTE capítulo, apresenta-se um pequeno histórico das medidas ambientais e os tipos de sensores a serem utilizados no projeto, bem como sua construção e evolução. Também se abordam os fundamentos nos quais se baseiam o projeto e o desenvolvimento do sensor.

2.1 Medidas ambientais – breve histórico

O estudo do clima, tempo e temperatura não são exclusividade da era moderna. É bem verdade que o interesse, seguido da possibilidade de se registrar os quatro parâmetros básicos da meteorologia sinótica (pressão, umidade, temperatura e ventos) estão presentes há apenas algumas centenas de anos. Isso não quer dizer que tais medidas não fossem feitas antes disso. Há registros na antiguidade de medidas de vento (através de bússola) e de índice pluviométrico – até antes de Cristo. Contudo, instrumentos capazes de incluir a meteorologia na era moderna, com uma abordagem científica, só surgiram com os avanços da época do Renascimento [4].

A umidade foi o primeiro parâmetro a ser estudado, com registros científicos. Aristóteles, ao estabelecer o ar como elemento, induziu a ciência ao erro através dos séculos, assumindo que o vapor d'água se tornava 'ar', tornando-se água posteriormente durante o ciclo hidrológico. Tal definição foi aceita através dos séculos até o surgimento da teoria 'atômica' no início do século XVII, com René Descartes. A partir desse momento o vapor d'água é tratado como uma "entidade" separada do ar. Apesar disso, as propriedades higroscópicas já eram

conhecidas havia duzentos anos, quando dois homens quase que simultaneamente relataram suas observações acerca deste princípio. O cardeal alemão (também matemático) Nicolas de Cusa, dentre suas várias contribuições para vários campos da ciência, descreveu em seu livro publicado em 1450 como ele descobriu que o peso de novelos de lã poderia aumentar com o aumento da umidade no ar. Dois anos depois, o cientista e arquiteto Leoni Alberti publicou um livro em que descrevia o fato de uma esponja seca poder ter seu peso acrescido em dias mais úmidos. Ele sugeriu que uma esponja poderia ser colocada em uma balança graduada para acompanhar as mudanças de umidade. Cinquenta anos mais tarde, Leonardo da Vinci fez dois desenhos bastante detalhados do que seriam higrômetros, baseados no conceito de Alberti e acabou por levar por muito tempo os créditos da invenção. Outras maneiras de se medir a umidade foram descobertas através dos anos. Ferdinando II, grão-duque da Toscana, descobriu ao lidar com umidade condensada na parte externa de um copo o princípio para o higrômetro de condensação. Mais tarde, Francesco Eschinardi notou o fato que um termômetro molhado esfria, o que teria levado ao surgimento do psicrômetro, que se trata de um instrumento com dois termômetros – um com bulbo seco e outro com bulbo exposto à umidade – com o qual se pode realizar medidas de índice de umidade do ar e de ponto de orvalho [4]. A figura 2 mostra um esboço de Leonardo da Vinci que demonstra o funcionamento do instrumento [5].

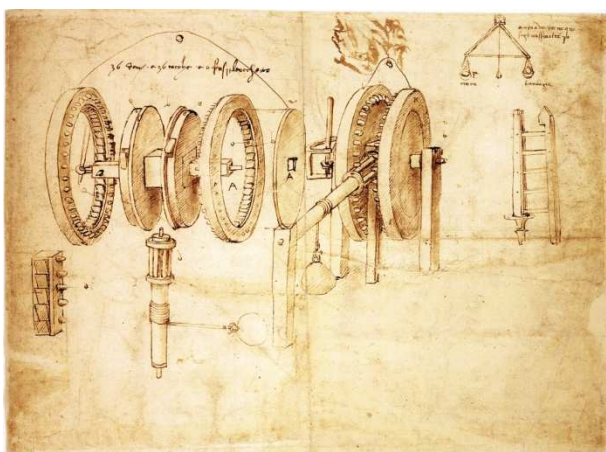


Figura 2 – Desenhos de Leonardo Da Vinci, descrevendo o funcionamento do higrômetro.

O interesse pela medida da temperatura também tem sua origem em tempos remotos. No século XVI, a tradução de um livro grego descrevia experiências de Filo de Bizâncio – datadas do século 3 AC – e de Heron de Alexandria – datadas do século 1 AC – ambos

reconhecidos por suas pesquisas na área da engenharia. Uma das experiências diz respeito ao fato da água poder ser forçada para fora de um recipiente ligeiramente aberto através da aplicação de calor. Em 1590, Galileu Galilei se mostra interessado pela experiência e constrói um 'termoscópio' baseado no princípio demonstrado. Galileu adicionou um fluido a um tubo de vidro delgado que tinha uma de suas extremidades ligada a uma taça com o mesmo fluido. Ele notou que, após o aquecimento do fluido, o mesmo subia pelo túbulo na mesma proporção da quantidade de calor aplicada. O termo 'termoscópio' foi utilizado por não se utilizar escala no tubo. O primeiro instrumento a ser chamado termômetro foi atribuído ao Duque de Toscana, financiador de Torricelli, que construiu um instrumento dotado de um tubo de vidro selado com um fluido no seu interior, e que apresentava uma escala fixa no corpo [4]. Porém, o trabalho relevante de Daniel Fahrenheit no século XVIII desenvolveu mais seriamente a construção dos termômetros. Além de ter definido a calibração deste instrumento baseada nos pontos de ebulição e evaporação da água, ele empregou o mercúrio líquido como fluido, expandindo o alcance das medidas do termômetro a toda a escala de temperaturas atmosféricas [6] [7].

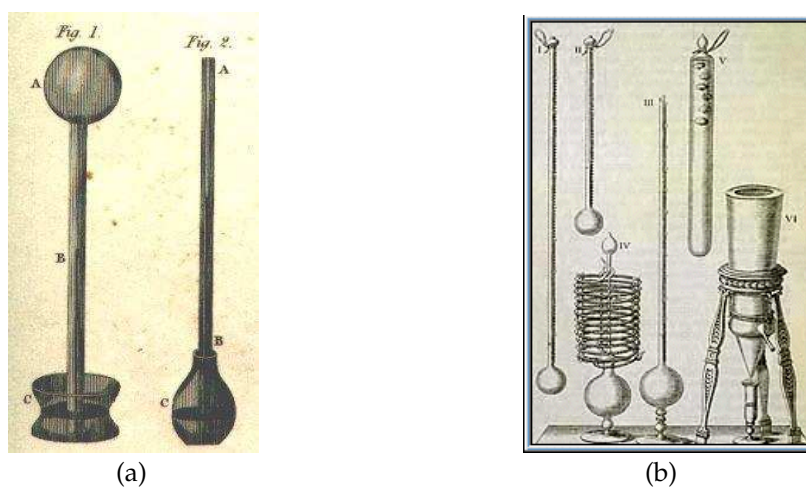


Figura 3 – (a)Termoscópio de Galileu e (b)termômetros de Fahrenheit.

Assim como nas outras medidas, o estudo da pressão atmosférica também tem suas peculiaridades. Após analisar problemas de mineiros italianos com seus sifões, Galileu observou que o vácuo poderia ser estudado nas situações onde o sifão não era eficiente (ainda não associando o fenômeno diretamente à pressão atmosférica). Galileu pediu a um de seus ex-alunos (Gasparo Berti) que realizasse um experimento para verificar a formação do vácuo nestas condições. Um tubo posto na vertical em um balde teve um barril de metal encaixado à sua

extremidade superior. Duas válvulas foram instaladas – uma no barril e outra na outra extremidade do tubo – e então se encheu ambos com água. Então, a válvula superior foi fechada e a inferior, aberta. Notou-se que uma grande quantidade de água saiu através do cano, mas alguma parte da água permanecia suspensa. Quando finalmente a válvula superior foi aberta, o ar adentrou ao barril de forma rápida, provocando um ruído alto. A partir daí, outro membro do grupo se encantou pelo efeito encontrado e continuou os experimentos – era Evangelista Torricelli. Torricelli reproduziu o experimento de Berti de maneiras diversas – uma delas em menor escala, utilizando um tubo longo de vidro, fechado numa das extremidades. Completou o tubo até a borda de mercúrio, tampou-o e o inverteu num recipiente também com mercúrio. Ao destampá-lo, o tubo também apresentou o vácuo presente na experiência do barril. No entanto, foi possível notar visualmente o vácuo formado pela pressão atmosférica exercida no tubo. Apesar de Torricelli nunca ter construído ele mesmo um barômetro realmente bem sucedido e nunca ter graduado os barômetros que experimentou, é a ele atribuída a invenção do instrumento, já que se tratou do primeiro a observar a pressão da atmosfera utilizando um aparato para isso. Blaise Pascal, em 1648, constrói a partir das experiências de Torricelli, o que seria o primeiro barômetro graduado e preciso. Ele sugere a seu colega pesquisador Florin Perier que dois instrumentos calibrados poderiam ser utilizados para se encontrar a variação da pressão com a altitude. Um deles foi posto no topo da montanha Puy-de-Dome enquanto que o outro permaneceu na base da montanha para fins de comparação, e tal diferença foi observada, no que é considerada a primeira ‘auscultação’ da atmosfera [4]. A figura 4 ilustra este instrumento [8].

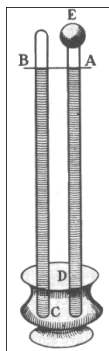


Figura 4 – Barômetro de Torricelli.

A seguir, seguem informações acerca dos sensores modernos. Apresentam-se também seu princípio de funcionamento e suas aplicações.

2.2 Sensores

Os sensores são fundamentais na medida das variáveis ambientais. Eles convertem a energia associada a um estímulo físico (ou químico) – por exemplo, calor, luz, som, pressão, campo magnético, ou um deslocamento particular – em um sinal com energia no domínio elétrico. A figura 5 a seguir ilustra os seis domínios de sinais (térmico, mecânico, magnético, radiante, químico e elétrico) com seus principais parâmetros físicos.

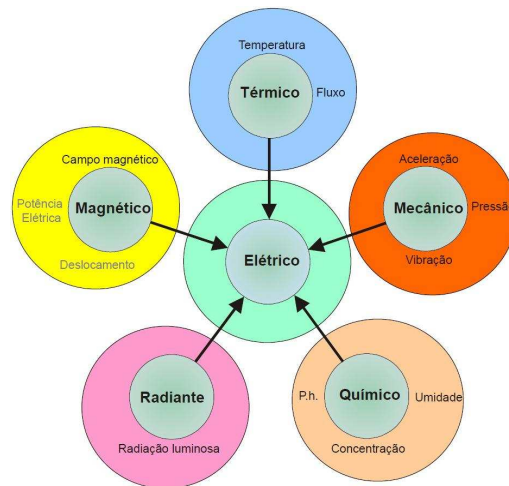


Figura 5 – Exemplos de conversão de energia realizada pelos sensores.

Sensores microeletrônicos são sensores construídos a partir do silício ou algum outro material semicondutor. Seus aspectos construtivos (dopagem, dimensão, geometria, orientação cristalográfica, etc) definem as características do sensor, listadas a seguir:

- ✓ Resolução;
- ✓ Seletividade;
- ✓ Estabilidade (a curto e longo prazos);
- ✓ Precisão e Exatidão;
- ✓ Tempo de vida;
- ✓ Tempo de recuperação;
- ✓ Velocidade de resposta;
- ✓ Histerese;
- ✓ Zona morta;
- ✓ Formato de saída;
- ✓ Impedância de entrada/saída;

Os sensores podem ser classificados como passivos ou ativos. Os sensores passivos têm, por definição, a propriedade de serem auto-suficientes, isto é, são capazes de por si mesmos gerarem um sinal elétrico equivalente à perturbação de entrada a que estão submetidos, sem necessidade de polarização. Já os sensores ativos (também chamados de sensores modulantes) necessitam de polarização para poderem realizar a conversão da energia do domínio de interesse em um sinal elétrico.

A seguir, são descritas as principais características de sensores de temperatura, umidade e pressão.

2.2.1 SENSORES DE TEMPERATURA

Os sensores de temperatura convertem energia térmica em um sinal elétrico. Entende-se por energia térmica a energia cinética interna das moléculas de um corpo ou ambiente [10].

Como exemplo de sensores de temperatura passivos, pode-se citar os termopares, que são sensores feitos a partir de ligas metálicas (como Cromel, Alumel, Constantan, Ferro, etc). Ao serem unidas, estas ligas, através do efeito Seebeck, produzem uma diferença de potencial entre si que varia conforme a temperatura à qual ambos estão submetidos.

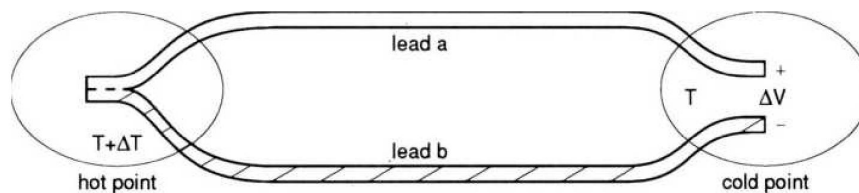


Figura 6 – Efeito Seebeck: uma tensão ΔV é gerada a partir de uma diferença de temperatura ΔT .

Dentre os sensores ativos têm-se os termo-resistores, termo-diodos e os termo-transistores. Os termo-resistores (ou RTD's – *Resistive Temperature Detectors*) são os mais utilizados por seu baixo custo e simples utilização e são construídos a partir de níquel, níquel-cobre ou platina. O termo-resistor mais conhecido é o Pt100, que apresenta uma resistência de 100Ω a 0°C .

A partir do silício, sensores são construídos para se medir a variação da temperatura com a variação da resistência (termo-resistor), ou com a variação da tensão de polarização direta

(termo-diodos) ou ainda com a variação da tensão de base-emissor em transistores bipolares (termo-transistor).

Termoresistores apresentam forte dependência do *stress* mecânico e da queda de tensão, além de uma tolerância muito ampla. A variação do valor da resistência de um resistor semicondutor pode chegar a $\pm 20\%$. A razão entre duas resistências casadas, no entanto, faz essa variação cair a valores em torno de $\pm 0.1\%$. Por esta razão, o emprego de pontes de resistores é apropriado neste caso.

Os termodiodos e termotransistores têm seu princípio de funcionamento baseado no comportamento da junção p-n do silício, sendo que o transistor é mais preciso porque o diodo apresenta não-idealidades físicas devido à sua geometria e ao seu processo de fabricação.

Os termotransistores são conhecidos como sensores de temperatura proporcional à temperatura absoluta (*Proportional To Absolute Temperature - PTAT*). A diferença ΔV_{BE} entre a base e o emissor de dois transistores é a medida de temperatura, no qual a tensão é proporcional à temperatura absoluta. A figura 7 ilustra o circuito de geração da tensão ΔV_{BE} e a construção de um transistor com multi-emissores, utilizado neste circuito [10]:

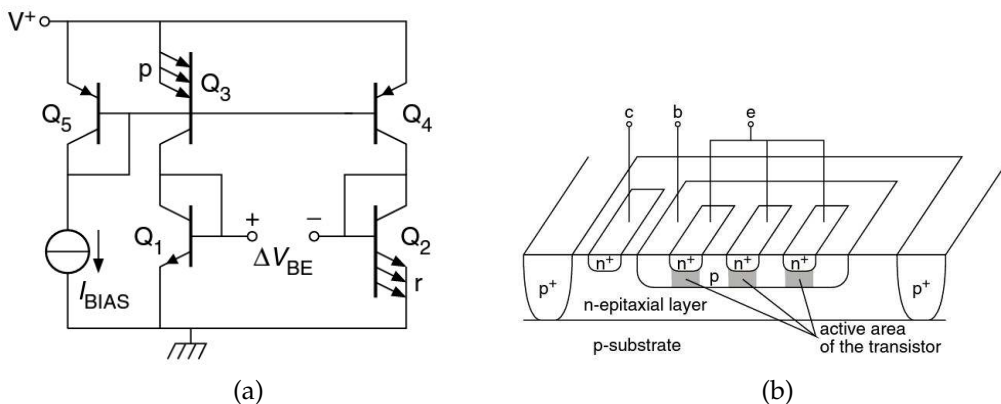


Figura 7 – Circuito de geração do ΔV_{BE} (a) e seção transversal de transistor com *r*-emissores (b).

A figura 7(a) mostra o princípio de um circuito de geração de tensão ΔV_{BE} . No circuito, todos os transistores estão submetidos a uma mesma temperatura T , o que é possível quando ele é fabricado em um único circuito integrado. Deste modo, a tensão ΔV_{BE} a partir dos transistores Q_1 e Q_2 é obtido conforme mostra a equação (2.1) [10]:

$$\Delta V_{BE} = V_{BE1} - V_{BE2} = \left(\frac{kT}{q} \right) \ln \left(\frac{I_{C1} J_{s2} A_{e2}}{I_{C2} J_{s1} A_{e1}} \right) \quad (2.1)$$

Para transistores idênticos fabricados no mesmo *chip*, pode-se afirmar que $J_{s1} = J_{s2}$. Para se conseguir uma razão de área de emissor precisa $r = A_{e2}/A_{e1}$, o transistor Q_2 é implementado como um resistor com multiemissores conectados em paralelo (vide figura 7(b)). Os transistores Q_3 , Q_4 e Q_5 operam na configuração de espelho de corrente. Em função dos terminais base-emissor interconectados e da mesma temperatura T , as densidades de corrente dos três transistores são iguais. Uma vez que o transistor Q_3 é implementado como um transistor multi-emissor – com um número p de emissores – a razão de corrente de coletor $I_{C3}/I_{C4} = I_{C1}/I_{C2}$, é mantido em um valor constante p , que não depende da temperatura ou de outros parâmetros físicos. A equação (2.1) pode ser reescrita de acordo com as características apontadas [10]:

$$\Delta V_{BE} = \left(\frac{kT}{q} \right) \ln(pr) \quad (2.2)$$

Segundo Meijer [10], em sistemas sensores de temperatura inteligentes, as tensões sensíveis a temperatura são convertidas em quantidades digitais adimensionais. Isso significa que, em algum momento na cadeia de processamento de sinais, um recurso precisa ser implementado para se calcular a razão da tensão do sensor e da tensão de referência. Tal referência pode ser gerada utilizando-se um circuito de referência *bandgap*, no qual uma tensão de compensação $V_C(T)$ é adicionada a $V_{BE}(T)$ para compensar no mínimo a dependência de temperatura de primeira ordem de $V_{BE}(T)$. Esta tensão de correção é obtida ao se amplificar a diferença PTAT ΔV_{BE} . Com isso, uma tensão de saída independente da temperatura V_{REF} pode ser obtida através da Equação (2.3) [10]:

$$V_{ref} = V_{BE}(T) + V_C(T) = V_{BE}(T) + A V_{BE}(T) = V_{BE0} \quad (2.3)$$

Por outro lado, ao se inverter a tensão $V_{BE}(T)$ e se ajustando o fator de escala A , obtém-se uma tensão $V_T(T)$ dependente da temperatura em C, F, ou outra escala. A figura 8 mostra um circuito básico para se gerar uma tensão de referência *bandgap* [10].

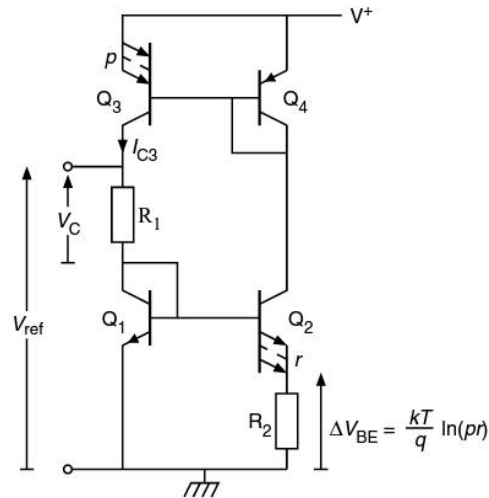


Figura 8 – Circuito de geração da tensão de referência V_{ref} .

2.2.2 SENSORES DE UMIDADE

A umidade é definida como a quantidade vapor d'água presente em um gás [11]. Os parâmetros mais importantes para se medir umidade são pressão parcial do vapor d'água, razão de mistura, umidade específica, umidade absoluta, fração molar do vapor d'água, umidade relativa e temperatura de ponto de orvalho (*dew-point*) [12]. Dentre esses, a umidade relativa – que é definida pela proporção da pressão do vapor d'água real com relação à pressão de saturação – é a mais utilizada para se medir umidade na atmosfera. O sensor de umidade fornece um sinal elétrico de saída proporcional à umidade relativa. De forma geral, sensores de umidade são mais complexos quando comparados aos sensores de temperatura.

Sensores de umidade relativa devem atender aos seguintes requisitos [12]:

- ✓ Alta sensibilidade sobre uma larga faixa de umidade;
- ✓ Resposta rápida;
- ✓ Boa reprodutibilidade e ausência de histerese;
- ✓ Robustez e vida longa;
- ✓ Resistência a contaminantes;
- ✓ Baixa deriva térmica;

- ✓ Estrutura simples e baixo custo.

Devido às propriedades térmicas da água, muitos métodos de se medir umidade são baseados em princípios térmicos. Em dois deles a umidade é expressa apenas em termos de temperatura. São eles [9]:

- a. Temperatura de Ponto de orvalho (*Dew-Point*);
- b. Diferença de Temperatura Psicrométrica;
- c. Sensores Capacitivos Microfabricados;

Neste trabalho, sensores capacitivos comerciais são empregados. Um sensor de umidade capacitivo consiste de um eletrodo principal (base), um eletrodo superior de metal poroso e uma camada dielétrica de polímero intermediária que tem o poder de adsorver moléculas de água. Tal adsorção resulta em um aumento da constante dielétrica relativa que, por sua vez, resulta em aumento da capacitância entre os eletrodos [10].

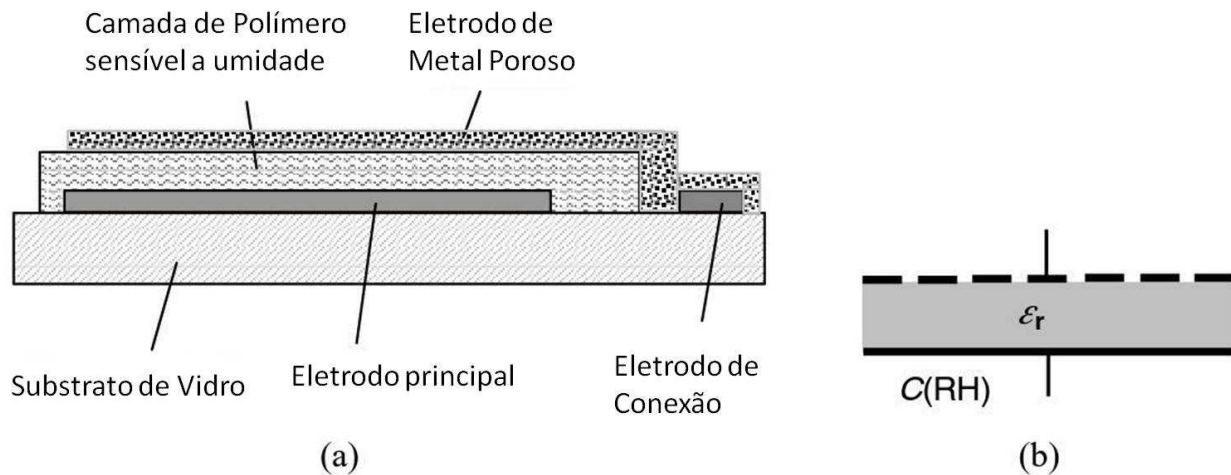


Figura 9 – Sensor de Umidade Capacitivo: detalhe (a) e símbolo equivalente (b)

2.2.3 SENSORES DE PRESSÃO ATMOSFÉRICA

A medida de pressão é sempre uma medida diferencial, feita com referência a outra conhecida ou de referência. A figura 10 ilustra algumas definições de tipos de medida de pressão [13]:

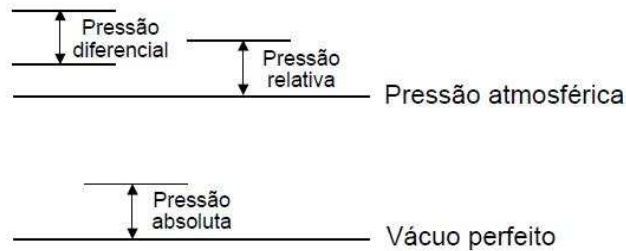


Figura 10 – Tipos de medida da pressão.

A pressão atmosférica é a que interessa a este trabalho. Portanto, trata-se da medida da variação da Pressão Absoluta causada pela atmosfera. Para converter esta pressão em sinal elétrico, utiliza-se o efeito piezoresistivo, que é a propriedade que metais e semicondutores apresentam de variação da resistividade de acordo com o *stress* mecânico aplicado. Esta propriedade está presente em metais (fator *gauge* K_g em torno de 2), e é ainda mais presente no silício (K_g maior em 2 ordens de grandeza), o que, combinada a excelentes propriedades mecânicas, o torna um ótimo material para se construir sensores de pressão com base neste princípio. Em geral, utilizam-se piezoresistores construídos sobre membranas de silício. Ao se exercer pressão sobre a membrana, é causado um *stress* mecânico em sua estrutura. Para se conseguir extrair a resistência equivalente à pressão, minimizando efeitos de temperatura, os sensores são construídos em forma de ponte, como se observa na figura 11 [14] [15].

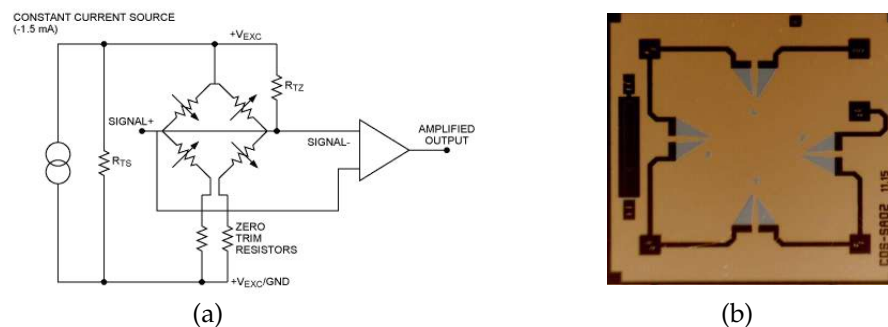


Figura 11 – Esquemático (a) e fotografia (b) de um sensor de pressão piezoresistivo

2.3 Microcontroladores

Microcontroladores são sistemas digitais integrados capazes de receber dados, interpretá-los e enviar os resultados a saídas pré-definidas, compostos dos seguintes elementos:

2.3.1 MICROPROCESSADOR

É a unidade responsável pelo processamento dos dados, gerenciamento da memória e controle dos dispositivos de entrada e saída de informações. Também conhecido como núcleo do microcontrolador (*core*), ele define a velocidade de processamento e, dependendo da arquitetura adotada, o consumo de energia e o tamanho de memória necessário para armazenar os dados e manipulá-los. Um dos pontos que definem a arquitetura do processador é quanto ao seu conjunto de instruções, que podem ser complexas (*Complex Instruction Set Computer* – CISC) – que demanda *hardware* mais complexo – ou simples (*Reduced Instruction Set Computer* – RISC) – que requer *hardware* mais simplificado.

Outra diferença entre arquiteturas se refere ao tipo de acesso à memória. A arquitetura do tipo *Von Neumann* utiliza o mesmo barramento de dados para resgatar instruções e dados da memória. Isso leva a uma limitação do número de níveis de *pipeline* e consequente limitação de execução de instruções por segundo (que possui unidade de *million instructions per second* – MIPS). A arquitetura *Harvard* possui barramentos distintos para acesso a instruções e dados na memória, permitindo *pipeline* mais extenso. Em geral, esta arquitetura utiliza uma unidade de gerenciamento de memória (*Memory Management Unit* – MMU) para controlar o acesso simultâneo às memórias. A figura 12 ilustra estas arquiteturas citadas [16]

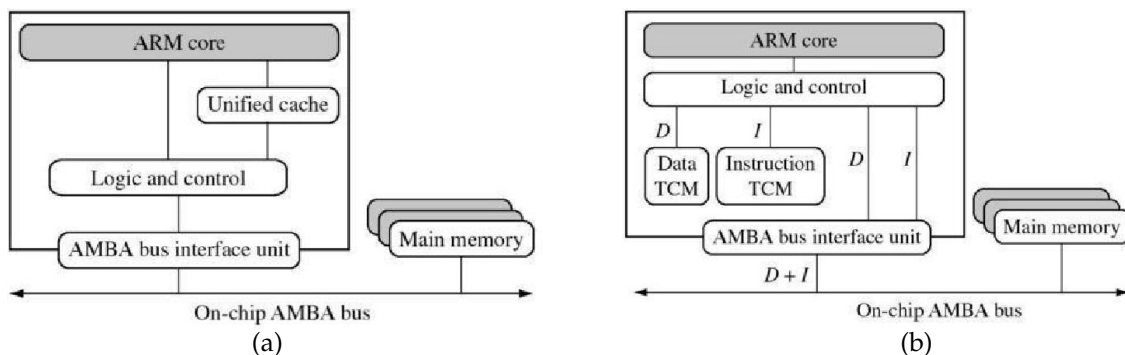


Figura 12 – Arquiteturas: Von Neumann(a) e Harvard (b)

O número de *bits* também serve para distinguir tecnologias e definir o volume de processamento dos microcontroladores. Os processadores podem ser classificados conforme o número de *bits* processados simultaneamente – 8, 16 ou 32 *bits*. Isso não significa que, p. ex., um microcontrolador de 8 bits não possa processar 32 bits com uma instrução. Com as devidas instruções isso é possível, mas tal operação leva um tempo bem maior, pois a arquitetura precisa acessar a memória mais vezes e executar um número maior de passos, enquanto que no microcontrolador de 32 bits este acesso é instantâneo e instruções específicas para este processamento podem ser utilizadas.

Uma tendência no projeto das aplicações das Redes de Sensores Sem Fio segue a idéia de permitir ao nó sensor (e a própria rede) que aprenda o comportamento no ambiente em vez de meramente realizar medidas e reportá-las sobre um único efeito de interesse [17]. Este novo tipo de desenvolvimento exige mecanismos que forneçam à rede formas mais complexas de atenção sobre o estado a ser reportado ou o contexto dos eventos. Ao mesmo tempo, tais paradigmas de aplicação também facilitam a incorporação de interações entre a rede e os eventos em andamento, fornecendo formas diferentes de dados e informações aos observadores, através do reconhecimento das preferências estabelecidas por eles. A escolha da arquitetura para estas tarefas compõe este paradigma, e a comparação entre arquiteturas é inevitável. Geralmente ela é feita entre as de 8 e 32 bits, pois se tratam da arquitetura mais popular (e mais antiga) e da arquitetura mais recente e moderna. Os processadores de 8 bits mais conhecidos são o 8051 da Intel e o PIC da Microchip, com frequências de 12MHz até o máximo de 40MHz em algumas versões. Dentre as arquiteturas de 32 bits, destacam-se a MIPS, PowerPC e a ARM, que será abordada com mais detalhes.

A arquitetura ARM (Advanced RISC Machines Ltd) existe desde 1991 com lançamento da arquitetura ARMv3 (ARM6). A partir disso, várias versões foram lançadas e licenciadas por empresas para utilização em microcontroladores de propósito geral.

Para efeito de comparação entre microcontroladores de 8 e 32 bits, foi realizada uma avaliação de desempenho das duas arquiteturas perante uma multiplicação. Os resultados podem ser vistos a seguir [18]:

Tabela I – Comparação entre desempenhos da arquiteturas 8051 e Cortex-M0

Arquitetura	8051 (8 bits)	ARM Cortex-M0
Rotina exemplo (multiplicação de 16 bits)	MOV A, XL ; 2 bytes	MUL AB; 1 byte
	MOV B, YL ; 3 bytes	ADD A, R1; 1 byte
	MUL AB; 1 byte	MOV R1, A; 1 byte
	MOV R0, A; 1 byte	MOV A, B ; 2 bytes
	MOV R1, B; 3 bytes	ADDC A, R2 ; 1 bytes
	MOV A, XL ; 2 bytes	MOV R2, A; 1 byte
	MOV B, YH ; 3 bytes	MOV A, XH ; 2 bytes
	MUL AB; 1 byte	MOV B, YH ; 3 bytes
	ADD A, R1; 1 byte	MUL AB; 1 byte
	MOV R1, A; 1 byte	ADD A, R2; 1 byte
	MOV A, B ; 2 bytes	MOV R2, A; 1 byte
	ADDC A, #0 ; 2 bytes	MOV A, B ; 2 bytes
	MOV R2, A; 1 byte	ADDC A, #0 ; 2 bytes
	MOV A, XH ; 2 bytes	MOV R3, A; 1 byte
	MOV B, YL ; 3 bytes	
		MULS r0,r1,r0
Ciclos de clock p/ execução	48	1
Tempo (ns)	4000 (@ 12 MHz)	21 (@ 48 MHz)
Tamanho de código (bytes)	48	2
Consumo [μ(A.s)]¹	7200	151,2

¹ Considerando o consumo médio de 150 μ A/MHz

Se for considerado que ambas as arquiteturas gastam a mesma energia por *clock*, pode-se concluir que o microcontrolador de 32 bits gasta 2,1% da energia gasta pelo microcontrolador de 8 bits para uma multiplicação. Para se tratarem os dados dos sensores são necessárias várias multiplicações, e quanto mais rápido for executada esta tarefa, mais rápido o microcontrolador poderá ser posto em modo de espera para economia de energia. Portanto, o uso do microcontrolador de 32 bits se mostra uma boa escolha para ser aplicado a redes de sensores sem fio.

2.3.2 MEMÓRIAS

São as unidades responsáveis por armazenar dados. As memórias utilizadas em microcontroladores podem ser classificadas em voláteis – para manipulação e tratamento dos dados – e não-voláteis – para armazenamento de dados. Como memórias não voláteis, pode-se citar as memórias ROM (*Read-Only Memory*) como as *flash* e as EEPROM's (*Electrically Erasable*

Read-Only Memory). Tais memórias são utilizadas para armazenar o *firmware* do microcontrolador e alguns dados de referência em tempo de execução que não podem ser perdidos quando o sistema é desenergizado. Como memórias voláteis, podem-se enumerar as memórias RAM (*Random Access Memory*) como as SRAM (*Static Random Access Memory*) e as DRAM (*Dynamic Random Access Memory*). As SRAM's são caracterizadas por ocuparem grande área de silício, serem de rápido acesso e consumirem pouca energia (se comparadas com as DRAM's), sendo as mais empregadas em memórias internas do microcontrolador pelas duas últimas características citadas. As DRAM's consomem mais energia, são mais lentas que as SRAM's e necessitam de um sinal de atualização (*refresh*) para que os dados gravados se mantenham gravados. Porém ocupam muito menos espaço que as memórias estáticas, sendo empregadas como memórias externas, gerenciadas por um controlador (*External Memory Controller* – EMC) ou, nos casos de vários níveis de *pipeline*, de uma unidade de gerenciamento de memória (MMU) [16].

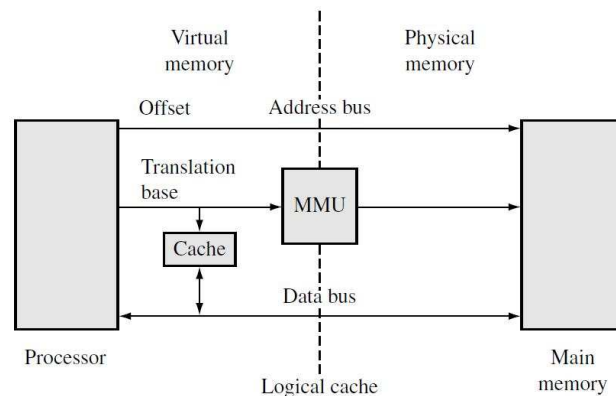


Figura 13 – Exemplo de uso da MMU para controle de memórias *cache* lógicas

2.3.3 DISPOSITIVOS DE ENTRADA E SAÍDA

São unidades dedicadas à interface entre o microcontrolador e elementos externos. Para esta tarefa, pode-se utilizar pinos de entrada e saída genéricos (*Generic Purpose Input/Output* – GPIO), blocos de comunicação serial e paralela, interfaces com memórias externas e *displays*, conversores analógico-digitais (*Analog-to-Digital Converters* – ADC's) e digital-analógicos (*Digital-to-Analog Converters* – DAC's), pinos de controle de motor, entre outros.

Dentre as interfaces de comunicação serial mais importantes, pode-se enumerar:

- ✓ UART (*Universal Asynchronous Receiver Transmitter*): largamente utilizada para comunicação entre o microcontrolador e um microcomputador (PC) ou outros sistemas, através de 2 pinos para comunicação – um para transmissão (TX) e outro para recepção (RX) dos dados. Outros pinos estão disponíveis para auxiliar na comunicação. Para transmissão dos dados da UART, é necessário escolher entre os vários padrões existentes de comunicação, dentre eles o RS-232, RS-422, RS-485, entre outros disponíveis. Para estar compatível com estes padrões, um circuito integrado adicional é utilizado para transformar os sinais da UART em sinais compatíveis eletricamente com o padrão do barramento e converter os sinais vindos do barramento para níveis aceitáveis pelo microcontrolador [19].
- ✓ I²C (*Inter-Integrated Circuits*): É uma interface de comunicação de dois fios utilizado para comunicação entre circuitos integrados, especialmente para fins de configuração. Para este modo, um dos dispositivos no barramento precisa ser definido como Mestre – responsável por gerar o sinal de *clock* de referência (*serial clock*, SCL), enquanto os demais são definidos como Escravos. Além disso, são definidos os transmissores e os receptores dos dados no barramento, que são transmitidos pelo canal de dados (*serial data*, SDA). Para acessar o escravo, o mestre deve indicar o seu endereço – único para cada dispositivo – antes do envio de qualquer comando. As velocidades de comunicação podem ser de 100kbps (Modo normal) 400kbps (Modo rápido) e 1Mbps (Modo rápido estendido);
- ✓ SPI (*Serial Protocol Interface*): Nesta interface, a comunicação é feita de modo parecido com o I²C, com um sinal de *clock* e dois sinais de dados, que são utilizados conforme os papéis de Mestre e Escravo como entrada ou saída, com os pinos MISO (*Master Input, Slave Output*) e MOSI (*Master Output, Slave Input*). Para a sincronização dos dados é usado um quarto sinal (*Slave Select – SL*), por onde se habilita o tráfego de dados entre Mestre e Escravo no barramento. Sua velocidade pode chegar a 7,5Mbps;
- ✓ CAN (*Controller Area Network*) : É uma interface usada para comunicação robusta em ambientes ruidosos e hostis, desenvolvido inicialmente para automóveis, mas

com uso também em ambientes industriais. Seu protocolo é baseado em mensagens e implementa as duas primeiras camadas do modelo OSI (*Open Systems Interconnection*) para realizar as trocas de mensagens, deixando para a aplicação a implementação das outras camadas, caso necessário. O protocolo é do tipo CSMA/CD (*Carrier Sense Multiple Access with Collision Detection*), que define várias diretivas de segurança de dados para o protocolo. A velocidade de transmissão chega a 1Mbps;

- ✓ Interface SD/MMC (*Secure Digital/MultiMedia Card*): É a interface utilizada para comunicação entre o microcontrolador e um cartão de memória SD/MMC. Para acessá-lo, é utilizado um pino para fornecer um sinal de *clock*, quatro pinos para a troca de dados e um pino exclusivo para o envio de comandos e recebimento de respostas. Enquanto o software define quais os comandos a serem enviados e se prepara para enviar os dados, os blocos de controle se responsabilizam por gerar o clock e as temporizações requeridas pelo protocolo. As velocidades de transmissão e recepção podem chegar a 30Mbps.

Além dessas interfaces citadas, têm-se ainda interfaces de comunicação serial mais velozes para aplicações distintas, como as que seguem:

- ✓ Ethernet: É a interface de rede mais utilizada em todo o mundo atualmente. Através dela é possível conectar o sistema a redes locais – Intranets, *Local Area Networks* (LANs) – e globais – Internet, *Wide Area Network* (WANs). Ela segue o padrão IEEE (*Institute of Electrical and Electronics Engineers*) 802.3, que estabelece quais os padrões elétricos e óticos utilizados, bem como a forma e controle de acesso ao meio (*Medium Access Control* – MAC) físico adotado para comunicação. A interface implementa as duas camadas mais baixas do modelo OSI – Camadas Física e de Enlace. No caso dos microcontroladores, é necessário um circuito integrado externo (PHY) que realiza a comunicação na camada física, conectando-se com o microcontrolador através de uma Interface Independente do Meio (*Media Independent Interface* – MII) ou de uma RMII (*Reduced MII*). Cabe ao software embarcado no microcontrolador montar as camadas superiores do

modelo OSI, criando a camada de rede – IP(*Internet Protocol*), IPSec – a camada de transporte – TCP (*Transmission Control Protocol*), UDP (*User Datagram Protocol*) – a camada de sessão – NetBIOS, SOCKS – a camada de Apresentação – TLS (*Transport Layer Security*), SSL (*Secure Sockets layer*) – e camada de Aplicação – HTTP (*Hyper Text Transfer Protocol*), SMTP (*Simple Mail Transfer Protocol*), FTP (*File Transfer Protocol*). Dependendo do meio utilizado, as velocidades de comunicação nesta interface variam de 10Mbps a 100Gbps [20].

- ✓ USB (*Universal Serial Bus*): É uma interface serial diferencial criada para padronizar a comunicação entre computador e dispositivos diversos (ou ainda a comunicação inter-dispositivos). O padrão se baseia na comunicação entre “*host*” e periféricos “*device*”, onde o “*host*” tem como tarefa abrigar os “*devices*” em uma de suas 127 portas, numa configuração do tipo estrela em camadas (*tiered-star*). A transferência de dados pode ocorrer da forma isócrona (ou seja, com uma taxa de transmissão fixa com o tempo), por interrupção, transferência em massa e de controle. A comunicação é baseada em *pipes* – canais lógicos – por onde o *host*, através de um *endpoint* próprio, acessa um dos *endpoints* do *device* e realiza a troca de mensagens. Em microcontroladores podem coexistir as duas interfaces, sendo que cada uma é utilizada com funções distintas. A interface USB *device* pode implementar as classes HID (*Human Interface Device*), UVC (*Universal Video Class*), CDC (*Communication Device Class*), MSD (*Mass Storage Device*), de Áudio, entre outras de aplicação específica. Já a interface USB *host* necessita implementar uma interface de controle (*Host Controller Interface*, HCI) que pode ser aberta (*Open HCI*, OHCI), universal (UHCI) e aprimorada (*Enhanced HCI*, EHCI). A velocidade de transferência pode atingir 1,5Mbps em dispositivos USB da versão 1.0 *Low-Speed*(LS) e 12Mbps da *Full-Speed*(FS) e 480Mbps em dispositivos USB da versão 2.0 *High-Speed*. Alguns microcontroladores são compatíveis com a versão 3.0 da USB (*Super Speed*) e podem atingir até 5Gbps [21].

2.3.4 TEMPORIZADORES / CONTADORES

Temporizadores e Contadores são circuitos internos ao microcontrolador capazes de contar ciclos do *clock* interno (ou externo) e torná-los disponíveis para o programa em execução ou tomar ações em um tempo ou intervalos definidos, podendo gerar interrupções programadas do sistema. Também podem ser dotados de pinos de captura que sejam capazes de monitorar transições em um pino externo, para em seguida tomar alguma ação ou gerar uma interrupção.

Alguns temporizadores têm função especial, como o temporizador de *Watchdog* e os geradores de sinal PWM (*Pulse Width Modulation*). O temporizador de *Watchdog* tem por tarefa monitorar alguma falha no processamento interno do microcontrolador e disparar um reinício do sistema (*reset*). Os geradores de PWM são temporizadores que, ao serem previamente configurados, geram sinais de controle periódicos com largura de pulso modulada, que podem ser utilizadas para controle de motor ou dimerização de lâmpadas, entre outras aplicações.

2.3.5 BLOCOS INTERNOS DE APLICAÇÕES DIVERSAS

Além dos circuitos já citados, podem-se citar outros utilizados pelo microcontrolador com diversas funções, como segue:

- ✓ PLL (*Phase-Locked Loop*): É o circuito responsável por gerar a frequência interna do microcontrolador a partir de uma frequência de referência, que pode ser fornecida por um circuito RC interno ou por um cristal externo. A frequência gerada pelo circuito PLL é conduzida à entrada do bloco de controle do sistema que fornece a frequência ao núcleo de processamento e a subdivide aos periféricos;
- ✓ Relógio de Tempo Real (*Real Time Clock – RTC*): Trata-se de um temporizador capaz de fornecer ao sistema uma base de tempo separada da utilizada pelo microcontrolador, podendo funcionar com uma alimentação exclusiva e utilizar um cristal de precisão para gerar uma referência à passagem do tempo real. Pode, dependendo do bloco, deter função de calendário, alarme e memória auxiliar para armazenamento de algum estado relevante ao sistema.

- ✓ Detector de *Brownout* (*Brownout Detector* – BOD): Este circuito é responsável por monitorar níveis de tensão de alimentação abaixo do normal e, ao detectá-los, proceder com uma ação previamente designada – como um disparo de interrupção ou *reset* do dispositivo. O termo lembra um estado intermediário entre o valor de tensão nominal de trabalho presente (*White-Out*) e sua ausência total (*Black-Out*). Em alguns microcontroladores apenas um valor é disponível para verificação, enquanto que em outros dispositivos a tensão pode ser detectada em alguns níveis, para ações distintas para cada um deles.
- ✓ Unidade de Gerenciamento de Energia (*Power Management Unit* – PMU): É o circuito que controla a distribuição de energia para os blocos do microcontrolador e gerencia os seus modos de energia. Através dele, pode-se habilitar o sistema para economizar energia e restabelecer o funcionamento pleno através de uma interrupção externa ou interna.

2.4 Comunicação e Redes Sem Fio

Juntamente com sensores e microcontroladores, os circuitos para comunicação sem fio são basilares para as Redes de Sensores Sem Fio. A maioria das aplicações nesta área são projetadas sob restrições de energia de comunicação e largura de banda [17]. Tais restrições levam aplicações a limitar seus mecanismos de aquisição de dados para tipos de dados de baixa largura de banda, ou seja, a quantidade de dados transferida (*throughput*) tende a ser pequena para não comprometer a autonomia e o processamento de dados internamente ao nó. Idealmente, um objetivo essencial a ser perseguido em projetos de Redes de Sensores Sem Fio é um equilíbrio perfeito entre o tempo de vida de cada um dos nós sensores e o *throughput* [22]. O correto dimensionamento e configuração da comunicação utilizada são fundamentais para se estabelecer um baixo consumo de energia sem comprometimento do seu desempenho.

A comunicação sem fio existe desde o final do séc XIX, com Landell de Moura e Marconi, com a demonstração da telegrafia sem fio [23]. Ao longo do séc. XX, as formas e meios de comunicação foram crescendo e mudando conforme a evolução das telecomunicações, com a

utilização de satélites, fibras óticas, cabos submarinos, *links* de microondas e diversas formas de comunicação sem fio. Dentre as várias aplicações para elas, pode-se citar como áreas principais a telefonia, os sistemas de transmissão de rádio e TV (Broadcasting), sistemas de dados e multimídia e redes de comunicação infravermelha [24].

Para se utilizar o espectro eletromagnético para fins de transmissão e comunicação, as emissoras e operadoras necessitam de concessão governamental. No entanto, existem algumas faixas de frequência nas quais podem ser estabelecidas comunicação sem que essa formalização seja necessária. Ainda assim, os dispositivos que utilizam estas faixas de trabalho precisam estar em conformidade com o padrão escolhido e, por isso, submetidos a testes eletromagnéticos que comprovem sua eficiência [25]. As faixas de frequência utilizadas são conhecidas como bandas ISM (*Industrial, Scientific and Medical*) e são reservadas internacionalmente para uso de radiocomunicação em aplicações industriais, médicas e científicas. Apesar das intenções iniciais, essas bandas começaram a ser utilizadas posteriormente para comunicação de curto alcance e de baixa potência [26] [27].

Tabela II – Principais Faixas de frequência ISM.

Frequência Central	Faixa de operação
433,92 MHz	433,05 MHz – 434,79 MHz
868 MHz	856 MHz – 868 MHz
915 MHz	902 MHz – 928 MHz
2,45 GHz	2,4 GHz – 2,5 GHz
5,8 GHz	5,725 GHz – 5,875 GHz

O IEEE, a partir de 1999, iniciou a criação do padrão 802, que são documentos estabelecendo especificações e procedimentos elaborados para garantir a confiabilidade de redes de acesso pessoal (*Personal Area Network* – PAN), local (*Local Area Network* – LAN), metropolitano (*Metropolitan Area Network* – MAN), amplo (*Wide Area Network* – WAN) e regional (*Regional Area Network* – RAN) e sua interação (quando factível). Dentro desta grande

quantidade de normas estão, por exemplo, as interfaces Ethernet, Wi-Fi, WiMAX e ZigBee. Dentre as redes sem fio padronizadas pelo padrão, existem quatro grupos principais:

- ✓ *Wireless PAN*: segue o padrão 802.15, caracterizado por comunicação em curta distância, de até 10m. Nessa categoria está o padrão Zigbee (IEEE 802.15.4) [28], Bluetooth (IEEE 802.15.1) [29] – ambos com taxas baixas – e o padrão 802.15.3 com taxas de 480Mbps – utilizando UWB (*Ultra Wide Band*) [30];
- ✓ *Wireless LAN*: segue o padrão IEEE 802.11, com distâncias de até 150m. É onde se encontra a tecnologia Wi-Fi – com transferências de 2Mbps (802.11a), 11 Mbps(802.11b), 54 Mbps (802.11g) e 108Mbps(802.11n) [31];
- ✓ *Wireless MAN*: segue o padrão IEEE 802.16, e abrange um raio de até 5km. O padrão WiMAX segue estas especificações e estabelece transferências de 35Mbps (802.16e) e 70Mbps [32];
- ✓ *Wireless WAN*: trata-se do padrão IEEE 802.20. Ele opera nos moldes da comunicação celular, como o GSM, GPRS (*General Packet Radio Service*), CDMA, 2,5G e 3G, sendo uma forma de acesso sem fio de banda larga móvel (*Mobile Broadband Wireless Access* – MBWA). Segundo o padrão, abrangem extensões de até 15km e lidam com taxas de transferência de 10kbps até 2,4Mbps [33];
- ✓ *Wireless RAN*: refere-se ao padrão IEEE 802.22. Sua operação envolve a interoperabilidade entre redes sem fio de alcances menores para fomentar a concorrência no acesso a banda larga. Ele se utiliza de canais vagos na banda de VHF(*Very High Frequency*) e UHF(*Ultra High Frequency*) – entre 54MHz e 862MHz. Seu alcance é de até 100km (com o gerenciamento apropriado do tráfego de dados pelas redes). As taxas de transferências são de 18Mbps a 24Mbps [34].

Nos termos do modelo OSI, o padrão IEEE 802 aborda a implementação da camada física (PHY) e da camada de enlace (*data link layer*), composta pelo MAC e pelo controlador de enlace lógico (*logical link control*, LLC), conforme mostrado na figura 14 [31].

- ✓ Topologia em árvore: Os nós sensores se conectam a nós roteadores, que se encarregam de efetuar as trocas de dados com o coordenador. Nos casos em que os roteadores assumem funções do coordenador (p. ex., para algum pré-processamento), a topologia é considerada como estrela estendida [35];



Figura 16 – Topologia em árvore

- ✓ Topologia em malha: em teoria, é caracterizada por ter a comunicação permitida entre todos os nós, ou seja, todos os nós se comunicam entre si, sejam eles sensores, roteadores ou coordenador. Ela estabelece a redundância no tráfego de dados e favorece segurança na comunicação (isto é, mínima chance de perda de dados). No entanto, para grandes redes, essa configuração se torna inviável por sobrecarregar cada nó, uma vez que este se torna responsável por transmitir seus dados e de todos os outros da rede. Por isso, existe a configuração de malha parcial, que permite a ligação de vários nós entre si, mas com conexão limitada aos mais próximos, simplificando as interações, porém ainda oferecendo um nível de redundância capaz de garantir a comunicação dos dados através da rede [35];



Figura 17 – Topologia em malha (ou *mesh*)

Apesar de servirem de exemplo, as topologias descritas não encerram todas as topologias disponíveis. Há variações e hibridismos possíveis que devem ser considerados ao se definir a arquitetura de redes a ser implementada.

A respeito do roteamento das redes, podem ser utilizados diversos protocolos e modos. Dentre os protocolos de roteamentos mais aplicados [36], há o AODV (*Ad-Hoc On Demand Distance Vector*), o DYMO (*Dynamic MANET On Demand*) e o DSR (*Dynamic Source Routing*). Para redes com grande quantidade de nós, são utilizadas tabelas *hash* distribuídas (*Distributed Hash Tables*, DHT), que são formas de se estabelecer identificações para cada nó a fim de situá-lo na rede e fornecer mecanismos para a melhor conexão possível entre nós. Dentre os tipos de DHT's, pode-se citar:

- ✓ DHT's baseadas na localização real: o nó é posicionado num local fixo e através disso são definidas as decisões de repasse de pacotes. Exemplos desse tipo são as GHT's (*Geographic Hash Tables*) ou a GPSR (*Greedy Perimeter Stateless Routing*) [37]. Por definição, são fortemente dependentes da localização geográfica;
- ✓ DHT's baseadas na localização virtual: bem semelhantes com as DHT's com localização real, porém o nó recebe uma posição relativa na rede, emulando uma localização geográfica. Como exemplo, tem-se o GRWLI (*Geographic Routing Without Location Information*);
- ✓ DHT's sem localização definida: Nenhuma informação de localização é utilizada. Dentre os tipos, destacam-se o VRR (*Virtual Ring Routing*) e o VCP (*Virtual Cord Protocol*). O VRR (*Virtual Ring Routing*) adota números inteiros para os nós, sem levar em conta sua localização. Já o VCP (*Virtual Cord Protocol*) usa um "cabo virtual" para posicionar os nós na rede, levando em conta a vizinhança e escolhendo a melhor rota disponível, iniciando do ponto "S" (*start*), até o ponto "E" (*end*), adotando valores reais para definir o caminho dos dados.

2.4.2 TECNOLOGIA ZIGBEE

O protocolo ZigBee foi desenvolvido pela Aliança ZigBee – um consórcio entre empresas de tecnologia e fabricantes de semicondutores globais. Trata-se de uma tecnologia de transmissão de dados sem fio para concepção de redes de baixo custo e baixo consumo, possibilitando grande autonomia. Ela segue o padrão IEEE 802.15.4 – destinado a redes sem fio de área pessoal com baixas taxas de transmissão (*Low Rate – Wireless Personal Area Network* – LR-

WPAN) – para implementar as camadas física e de enlace, transmitindo os dados nas frequências ISM de 868 MHz (Europa), 915 MHz (EUA) e 2,45 GHz (global).

As principais características da tecnologia são:

- ✓ Baixo consumo;
- ✓ Suporte a diversas topologias de rede;
- ✓ Grande capacidade de nós na rede;
- ✓ Criptografia dos dados em AES (*Advanced Encryption Standard*), de 128 bits;

Seguindo o IEEE802.15.4, o ZigBee define seus dispositivos como de função completa (*Full Function Device* – FFD) e de função reduzida (*Reduced Function Device* – RFD), que são descritos na tabela III [38].

Tabela III – Tipos de dispositivos em redes ZigBee.

Dispositivo	Funcionalidades disponíveis	Estado do rádio em modo de espera	Fonte típica de alimentação
RFD (<i>Reduced Function device</i>)	Limitada	Desligado	Bateria
FFD (<i>Full Function device</i>)	Todas	Ligado	Rede elétrica

Os dispositivos que fazem parte da rede são:

- ✓ Coordenador (*ZigBee Coordenador* ou ZC): Dispositivo do tipo FFD responsável pela formação da rede ZigBee; dele depende a sua existência, estabelecendo o canal de comunicação e um número lógico da rede. Ao definir os parâmetros, o coordenador permite o tráfego de dados entre os elementos da rede (roteadores e dispositivos finais). O coordenador, então, participa do redirecionamento dos dados, recebendo-os ou transmitindo-os. Cada rede possui um único coordenador;
- ✓ Roteador (*ZigBee Router* ou ZR): Dispositivo do tipo FFD, responsável por manter as informações da rede (ou criar novas informações), estabelecendo a melhor rota

para o tráfego de dados dos dispositivos finais até o coordenador e vice-versa. Uma vez integrado à rede, permite a conexão entre dispositivos finais a outros roteadores ou ao coordenador.

- ✓ Dispositivo Final (*ZigBee End Device*, ou ZED): dispositivo do tipo RFD, atua como receptor ou transmissor de dados, unindo-se a um nó pai – que pode ser um roteador ou o próprio coordenador. Não é capaz de redirecionar informações como um roteador. A figura 18 ilustra a hierarquia da rede em camadas do modelo OSI [35].

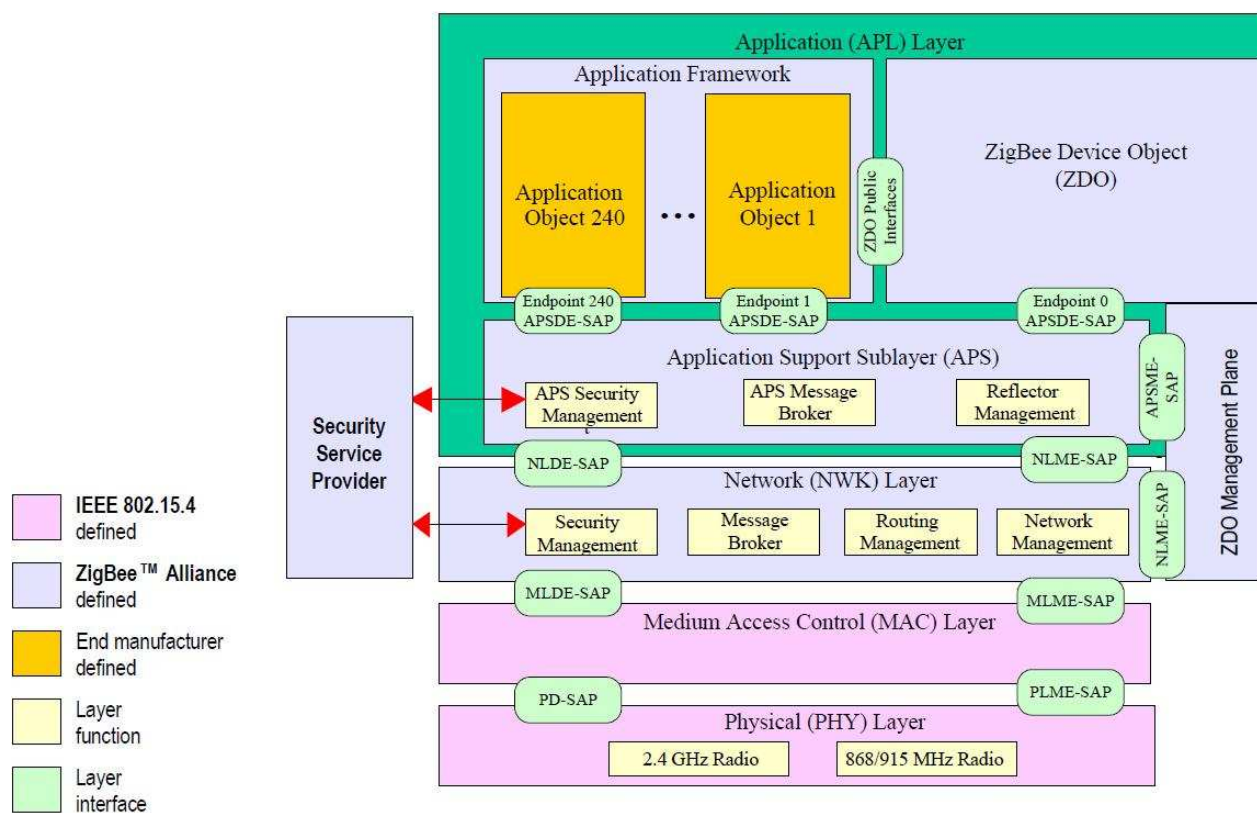
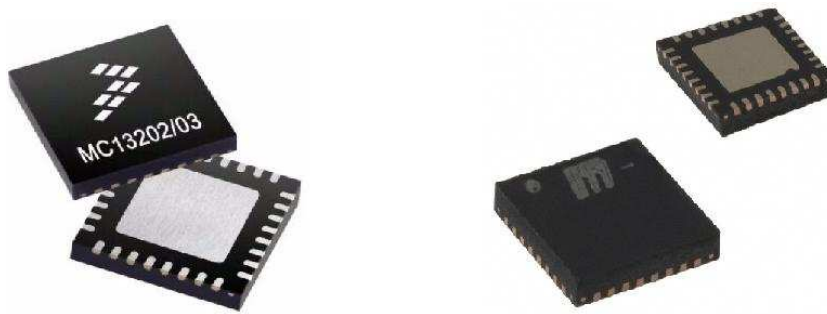


Figura 18 – Visão geral das camadas de rede da tecnologia ZigBee.

2.4.3 CIRCUITOS E MÓDULOS DE RADIOFREQUÊNCIA COMERCIAIS

Para a montagem da rede de sensores sem fio, pode-se utilizar desde transceptores até módulos completos já preparados e pré-definidos para esta aplicação. Os transceptores são circuitos integrados responsáveis por estabelecer a comunicação segundo padrões estabelecidos (como o padrão IEEE 802.15.4) a dispositivos já constituídos de um sistema microcontrolado,

interfaceados por uma comunicação serial (SPI, UART). Como exemplos, têm-se os MC1319x e MC1320x da Freescale [39] e o MICRF505 da Micrel [40].



(a) Figura 19 – Transceptores 802.15.4 da Freescale (a) e Micrel (b)

Outra opção são módulos autônomos (*motes*), capazes de implementar todas as camadas necessárias para comunicação, ficando a critério do desenvolvedor da rede integrar os sensores ou atuadores referentes àquilo que se quer monitorar ou controlar. Esses módulos são conhecidos como COTS (*Commercial-Off-The-Shelf*), também conhecidos como solução de prateleira. Um exemplo de *mote* é o MICA, desenvolvido pela Universidade de Berkeley. Sua arquitetura é baseada no microcontrolador de 8 bits ATmega 128 da Atmel e aplica o protocolo ZigBee para a criação da rede. Como a maioria dos *motes* do mercado, ele é construído de forma modular e permite a conexão de várias placas extensoras com várias funções como conexão USB (para testes locais) e sensores. Ele tem como característica de operação o uso do sistema operacional aberto *TinyOS*, desenvolvido para sistemas com recursos limitados que requerem versatilidade para integração de funções, e frequência de operação em 2,4 GHz, com versões operando em 915 MHz [41].

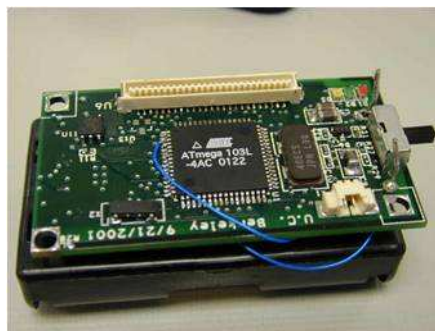


Figura 20 – Nó sensor comercial (mote) MICA

Como alternativa a estes modelos, existem soluções intermediárias que implementam o protocolo ZigBee mas são flexíveis quanto à aplicação. É o caso dos módulos X-Bee e Xbee Pro da Digi. São módulos de comunicação que implementam o protocolo ZigBee mas são construídos de forma a favorecer a integração a nós específicos. Eles operam nas faixas ISM de 915 MHz e 2,4GHz e têm a opção de utilizarem comandos de sua Interface de Programação de Aplicação (*Application Programming Interface*, API) ou de utilizarem comunicação transparente, através de uma porta UART por onde os dados são transmitidos e enviados [42].



Figura 21 – Módulo de Comunicação X-Bee.

2.5 Modos de alimentação

Para todo nó sensor é fundamental uma alimentação independente de fios, confiável e de longa duração. Ela pode ser proveniente de fontes fixas (pilhas e baterias), variáveis (painéis solares, microgeradores) e híbridas (que combinam fontes fixas e variáveis). Elas são apresentadas a seguir.

- ✓ Baterias: são módulos que fornecem corrente elétrica através de cargas acumuladas em células por reações químicas (p. ex., eletrólise) entre seus elementos. Podem ser classificadas em primárias (que não aceitam recarga e devem ser descartadas) ou recarregáveis. Os principais elementos químicos utilizados são o zinco(Zi), manganês (Mn), chumbo (Pb), níquel (Ni), mercúrio (Hg), cádmio (Cd), lítio (Li), entre outros elementos, como ligas de hidretos e polímeros. As mais populares são as de Carbono-Zinco (CZ), Manganês Alcalino (para pilhas e baterias alcalinas) e Chumbo Ácido. Algumas baterias utilizadas no mercado para uso em rede de sensores são as de Níquel-Cádmio (Ni-Cd), Níquel-Metal-Hidreto (NiMH) e íons de Lítio (Li-Ion) [43]. As baterias de íons de Lítio

tem como característica serem leves, compactas e apresentarem tensão constante durante todo o tempo de descarga.



Figura 22 – Baterias de íons de lítio (Li-Ion).

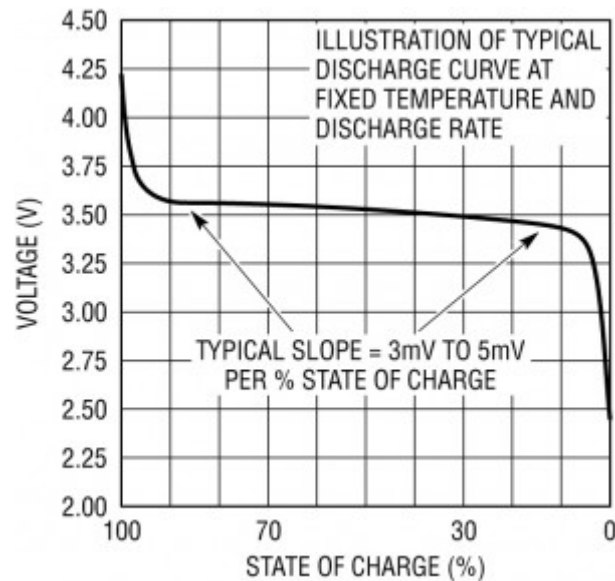


Figura 23 – Curva típica de tensão de saída da bateria por estado de carga.

- ✓ Painéis solares: São módulos dotados de painéis de silício fotovoltaicos que convertem a energia solar incidente em corrente elétrica. Apesar da baixa eficiência, esses painéis têm encontrado aplicações tanto em pequenos dispositivos como em Campos Geradores de Energia Solar (*solar farms*) [44].



Figura 24 – Pannel solar.

- ✓ Microgeradores elétricos: São pequenos geradores (microfabricados ou miniaturizados) capazes de gerar corrente elétrica a partir da energia mecânica proveniente de vibrações, oscilações, fluxos de ar, água, ou outros fluidos [45].

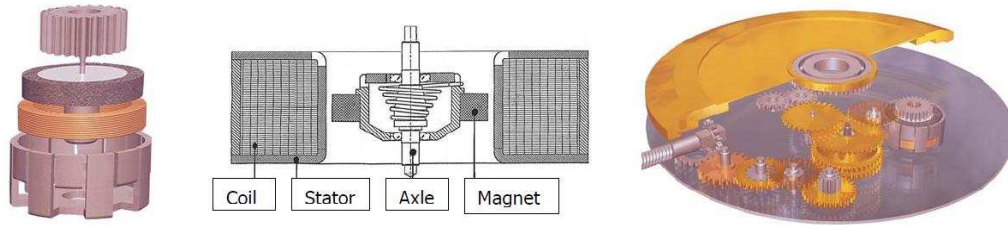


Figura 25 – Exemplos de microgeradores.

- ✓ Captura de ondas em radiofrequência (RF): além de ser utilizado para comunicação, há pesquisas para se utilizar o sinal de RF como fonte alternativa de energia para perfis de consumo baixo e ultra baixo, como é o caso dos nós das redes de sensores [46]. Apesar de promissor, é uma tecnologia que não está plenamente desenvolvida, mas pode ser uma alternativa em um futuro próximo com alguns casos comerciais conhecidos [47].



Figura 26 – Exemplos de uso de alimentação por RF.

Em se tratando de fontes variáveis de energia, o termo que se utiliza para se denominar a sua aplicação é Colheita ou Varredura de Energia (*Energy Harvesting / Scavenging*). Segundo estas técnicas, a energia só pode ser convertida ou acumulada quando disponível – daí o caráter variável [48]. Devido a essa característica, a operação das redes de sensores sem fio pode apresentar dois cenários, dependendo das aplicações, utilizando fontes híbridas [49]:

- ✓ Operação sob disponibilidade de energia: A leitura e transmissão dos dados rede só ocorre após a mínima quantidade de energia para isso for acumulada em uma

bateria de tamanho mínimo ou um capacitor, através um circuito capaz de detectar e tratar a fonte intermitente de tensão [50];

- ✓ Operação contínua: Os dados são lidos sempre que necessário utilizando a energia de uma bateria de alta capacidade, que é recarregada pela energia colhida pelas fontes alternativas;

Desta forma, é possível dar aos nós sensores uma vida longa e permite-se aprimorar as tarefas dos nós que exijam um consumo adicional de energia.

2.6 Conclusão da Revisão de Literatura

Neste capítulo foram abordados todos os itens fundamentais para a concepção de redes de sensores sem fio que estão disponíveis em livros, artigos, periódicos e *internet*. Como se trata de uma área ampla a ser coberta, o desenvolvedor de redes de sensores precisa conhecer diversas áreas de conhecimento e tecnologia que as compõe, procurando respeitar os requisitos de projeto e a aplicação à qual ele é destinado.

Capítulo 3

Desenvolvimento de Hardware e Software

APRESENTA-SE, neste capítulo, o projeto do hardware e do software do sistema que compõem a rede de sensores sem fio para monitoramento ambiental. As especificações dos principais requisitos do sistema são discutidas. Um diagrama de blocos desta rede de sensores é mostrado e na sequência detalha-se a escolha dos principais componentes do hardware. O projeto dos nós sensores da rede é apresentado. Por fim, discutem-se as principais especificações do software e seu se descreve o projeto em detalhes.

3.1 Requisitos do sistema

A rede de sensores sem fio para monitoramento ambiental será composta por dois tipos de nós: sensor e coordenador. Cada nó possui particularidades que resultam em especificações distintas.

Para o nó sensor, os requisitos necessários foram:

- Diâmetro igual ou menor a 70 mm, facilitando o manuseio e transporte;

- Autonomia de bateria que permita uma leitura por minuto por 90 dias corridos (tempo médio de manutenções previstas em campo);

- Custo inferior ao custo de um nó sensor comercial;
- Alcance mínimo de 60 metros no modo *indoor*;
- Aquisição de pressão barométrica, umidade relativa do ar e temperatura ambiente;
- Transmissão de leitura das medidas dos sensores já processadas e condicionadas;
- Peso igual ou inferior a 170g (peso médio de uma fruta [51]);

Já para o coordenador, têm-se os requisitos a seguir:

- Custo inferior ao custo de um datalogger simples;
- Fácil operação e comunicação com um computador (*plug-and-play*);
- Recepção e coleta dos dados dos nós;
- Disponibilização dos dados de forma remota;
- Conexão à rede (*Intranet* ou *Internet*) via cabo ou sem fio;

O nó sensor apresenta um desafio em relação aos requisitos. Ao mesmo tempo que precisa ser leve e ter tamanho reduzido, precisa também apresentar grande autonomia, o que leva à necessidade de uso de baterias pequenas e um sistema com baixo consumo, sem comprometer o processamento – necessário para a interpretação dos dados dos sensores – e o alcance. Para isso, foi avaliado o uso de componentes que permitissem o atendimento destes requisitos, estabelecendo um compromisso entre aqueles que são antagônicos. Para a construção do nó e do coordenador da rede, foi levantada a necessidade dos seguintes componentes: microcontrolador, sensor de temperatura, sensor de umidade relativa, sensor de pressão barométrica, módulo de comunicação e bateria, que são detalhados a seguir.

3.1.1 MICROCONTROLADOR

O nó sensor, conforme os requisitos apresentados, realiza a leitura dos sensores e o envio dessas informações sem fio. Para isto, é necessário o uso de componentes capazes de, junto com os sensores e módulos de comunicação, transferir, condicionar os dados das leituras e prepará-los para transmissão. Como foi apresentado no capítulo 2, o microcontrolador é um componente

único que abriga processamento, interface, memória e diversas funções integradas em um único circuito microeletrônico.

O microcontrolador da NXP Semiconductors – LPC1114 – foi o eleito para fazer o gerenciamento geral das tarefas do nó sensor. Trata-se de um microcontrolador com o núcleo de 32 bits da ARM mais simples (Cortex-M0). O núcleo deste microcontrolador foi desenvolvido para aplicações que necessitam de processamento em 32 bits – com suporte a multiplicação e alta velocidade – ao mesmo tempo em que sejam alimentados por baterias e que necessitem de um gerenciamento especial de energia.

Além destas características, o LPC1114 contém ainda [52]:

- ✓ 32 kB de memória Flash e 4kB de memória RAM
- ✓ Interfaces UART, SPI, I2C;
- ✓ ADC de 10 bits;
- ✓ Oscilador interno de 12MHz (permitindo *clocks* internos de até 50MHz);
- ✓ - Temporizador *Watchdog*;
- ✓ - Detector de *Brownout*;
- ✓ - 4 Temporizadores uso geral: 2 de 16 bits e 2 de 32 bits;
- ✓ - PLL;

Este componente agrega diversas funcionalidades que contribuem para um projeto compacto, pois evitam a aquisição de outros componentes, como um cristal externo. Apesar de tantos recursos, seu custo é comparado aos microcontroladores de 8 bits mais simples do mercado e assim contribui para um projeto de baixo custo. Sua alta integração e grande quantidade de blocos internos permitiu minimizar o número de componentes externos.

Sua programação pode ser realizada através de dois ambientes de desenvolvimento integrado (*Integrated Development Environment*, IDE) disponíveis: o IDE Realview desenvolvido pela da Keil™ e o IDE LPCXpresso desenvolvido pela Code_red™. O espaço em memória do firmware utilizado pode alcançar até 32kB. Para o uso direto do nó para entrada e saída dos dados, estão disponíveis a interface serial UART para comunicação com o módulo sem fio, a

interface I²C para comunicação com sensor de pressão e dois pinos de entrada e saída para comunicação com o sensor de umidade e temperatura (ambos os sensores descritos adiante neste capítulo).

Para economia de energia, o LPC1114 dispõe de três modos de operação em baixa energia. São eles:

- Modo "*Sleep*": Neste modo, o clock do microcontrolador é interrompido e a execução de instruções é suspensa. Todos os elementos internos continuam energizados e o sistema volta a funcionar depois de uma interrupção oriunda de quaisquer desses elementos. O consumo médio deste modo é de 4mA.

- Modo "*Deep Sleep*": Neste modo, o clock do microcontrolador é interrompido e a execução de instruções é suspensa. Todos os blocos analógicos são desligados, com exceção do Brownout e do Watchdog. Os estados e memória são conservados. Consumo médio deste modo é de 6uA.

- Modo "*Deep Power Down*": Neste modo todos os elementos internos e *clocks* são desligados, ficando apenas o pino de *wakeup* ligado. Apenas uma porção da memória reservada ao gerenciador de energia permanece energizada, podendo armazenar alguns estados importantes. O consumo do microcontrolador chega a 220nA neste modo de operação.

Para monitoramento e temporização, podem ser utilizados o detector de *Brownout* e o *watchdog*. O detector de *Brownout* é um componente interno ao microcontrolador que permite monitorar a tensão de alimentação em níveis. O detector de *Brownout* do LPC1114 possui até 4 níveis de alerta, assim discriminados de acordo com a folha de dados do fabricante:

- ✓ Nível 4: Sem alertas - bateria a 100% ou acima de 2,8V
- ✓ Nível 3: primeiro alerta - bateria a 80% ou entre 2,8V e 2,52V
- ✓ Nível 2: segundo alerta - bateria a 60% ou entre 2,52V e 2,22V
- ✓ Nível 1: terceiro alerta - bateria a 40% ou entre 2,22V e 1,8V
- ✓ Nível 0: quarto alerta - bateria a 20% ou abaixo de 1,65V

O *watchdog* é um temporizador originalmente empregado para monitorar eventuais travamentos dos programas em execução em microcontroladores. No caso do programa sendo

executado no nó, ele tem o papel de acordar o nó do estado de hibernação (*deep-sleep*). Isso é necessário porque no modo *deep-sleep* apenas o temporizador do *watchdog* é habilitado para funcionar. Todos os outros temporizadores são bloqueados para economizar energia enquanto nenhuma leitura ou transmissão estiver sendo efetuada. Ao se atingir o valor de contagem, o microcontrolador acorda do estado de hibernação, efetua as leituras dos sensores e os transmite, enviando os dados via UART ao módulo de comunicação. Após todas as operações, o microcontrolador volta a entrar no modo de hibernação e espera o *watchdog* disparar a próxima interrupção.

3.1.2 SENSOR DE TEMPERATURA AMBIENTE E UMIDADE RELATIVA DO AR

O sensor escolhido para medir a umidade relativa do ar foi o SHT15 da fabricante Sensirion. Este sensor possui características de precisão, tolerância e faixa de operação que se encaixam nos requisitos do nó. Além disso, ele também efetua a medida de temperatura ambiente. Esta característica de leitura integrada ajuda a minimizar o tamanho do nó, reduzindo espaço ocupado pelo circuito sensor. O condicionamento das medidas é feito através de um conversor analógico digital de até 14 bits de resolução, um circuito processador de sinais e um circuito de interface serial.

As características detalhadas do sensor são as seguintes [53]:

- Leitura de umidade relativa do ar: O elemento sensor utiliza o princípio capacitivo para realizar as medidas de umidade. O sensor é capaz de medir umidade de 0% até 100%, com um tempo de resposta de 8 segundos;

- Leitura de temperatura ambiente: o elemento sensor utiliza o efeito *bandgap* do semicondutor para realizar as medidas de temperatura ambiente. O sensor é capaz de medir temperatura de -40 °C até 123,8 °C

- Comunicação digital: o sensor dispõe de saída digital dos dados, através de dois fios pelos quais se realiza uma comunicação serial dos dados relativos às leituras de umidade e temperatura realizados pelo sensor. Através deste canal serial ainda é possível enviar comandos de reset e de configuração referentes à resolução da leitura.

A comunicação digital serial com o sensor se apresenta como uma grande vantagem quando se verifica a independência com relação ao acoplamento do sensor com o microcontrolador e assim dispensando o uso de conversores analógico-digitais externos. Uma vez que se garante que o dado vindo do sensor não necessita de nenhuma compensação referente à impedância ou resistividade das trilhas da placa de circuito, o projeto adquire versatilidade e liberdade de montagem e construção, sem a necessidade de se modificar o firmware utilizado para se realizar a medida. Em contrapartida, a comunicação utilizada pelo sensor SHT15 da Sensirion não segue nenhum padrão disponível no mercado. Esta característica demandou a necessidade de se emular em software o protocolo proprietário para comunicação bidirecional com o sensor. Por algum tempo durante o desenvolvimento do sensor foi assumido que a comunicação I²C poderia ser utilizada devido à semelhança entre protocolos. No entanto, os testes se revelaram ineficazes e a emulação, mais complexa e trabalhosa, precisou ser implementada.

A figura 27 mostra imagens de microscópio do circuito integrado do sensor SHT15.

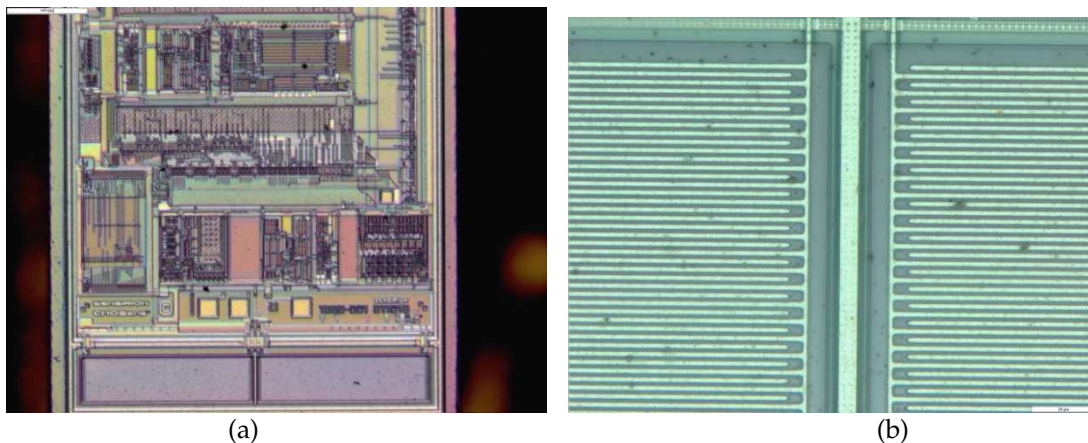


Figura 27 – Circuito Integrado do sensor SHT15 (a); detalhe do sensor de umidade (b).

3.1.3 SENSOR DE PRESSÃO ATMOSFÉRICA

O sensor de pressão atmosférica selecionado para o uso no nó foi o MPL115A2, sensor de pressão absoluta com saída digital para aplicações de baixo custo, fabricado pela Freescale. Seu diagrama interno e sugestão de conexão com o microcontrolador é mostrado na figura 28.

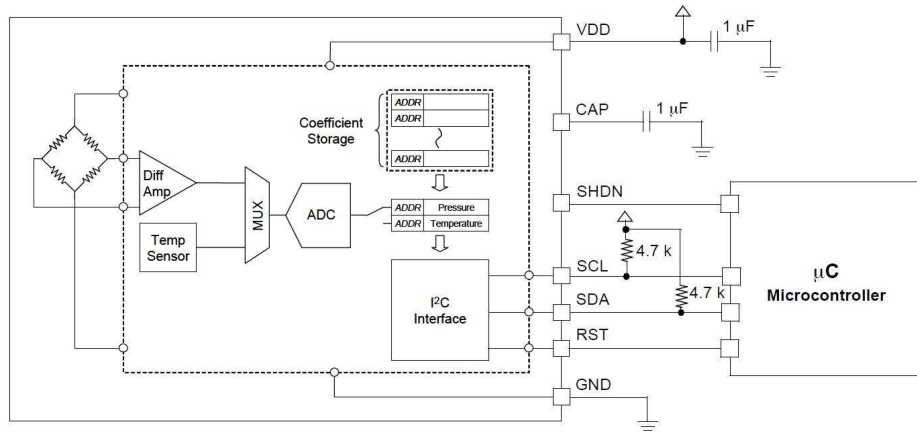


Figura 28 – Diagrama interno e conexão do sensor de pressão atmosférica MPL115A2.

As principais características deste sensor são detalhadas a seguir [54]:

- Leitura de pressão barométrica: o elemento sensor empregado é um sistema microeletromecânico (MEMS), baseado em ponte de Wheatstone resistiva sobre uma membrana de silício, construído para medida de pressão atmosférica. O sistema ainda conta com um sensor de temperatura integrado que, através de um algoritmo de compensação, permite calcular a pressão absoluta com 1kPa de precisão. Este sensor foi originalmente projetado para ser utilizado para uso em altímetros e discos rígidos;

- Comunicação digital: o sensor de pressão contém um conversor analógico-digital que, em conjunto com um circuito de condicionamento e um circuito de interface serial realiza a aquisição da informação do sensor e a disponibiliza em formato digital. A facilidade encontrada com este sensor é que ele utiliza o protocolo I²C, que é compatível com o protocolo de comunicação do microcontrolador, sem a necessidade de emulação via software. O cálculo da pressão atmosférica envolve um polinômio de duas variáveis de segunda ordem e, para calculá-lo, o sensor fornece os valores de pressão atmosférica, temperatura e um conjunto de 12 coeficientes, para se utilizar na equação de compensação (3.1), que segue:

$$P_c = a_0 + (c_{11}P_{adc} + c_{12}T_{adc})P_{adc} + (b_2 + c_{22}T_{adc})T_{adc} \quad (3.1)$$

Sendo que:

P_c – valor da pressão compensada;

P_{adc} – saída de pressão de 10 bits do ADC do MPL115A2;

T_{adc} – saída de temperatura de 10 bits do ADC do MPL115A2;

a_0 – coeficiente de offset de pressão;

b_1 – coeficiente de sensibilidade de pressão;

c_{11} – coeficiente de segunda ordem de linearidade da pressão;

c_{12} – coeficiente de sensibilidade de temperatura;

b_2 – coeficiente de primeira ordem do offset de temperatura;

c_{22} – coeficiente de segunda ordem de offset de temperatura.

Os valores dos coeficientes são transmitidos em formato digital, com dados em formato de ponto flutuante. Por isso, é preciso inicialmente preparar os valores antes de calculá-los. Depois de devidamente preparados e convertidos para valores reais, os cálculos são realizados através da equação de compensação parte por parte até que o valor seja encontrado. O fato de os cálculos serem feitos em um microcontrolador de 32 bits é de fundamental importância para que nem o tempo e nem a energia consumidos neste processo sejam preponderantes no uso da carga da bateria.

Na figura 29, uma imagem de microscópio ilustra a estrutura microfabricada do sensor responsável pela transdução da pressão.

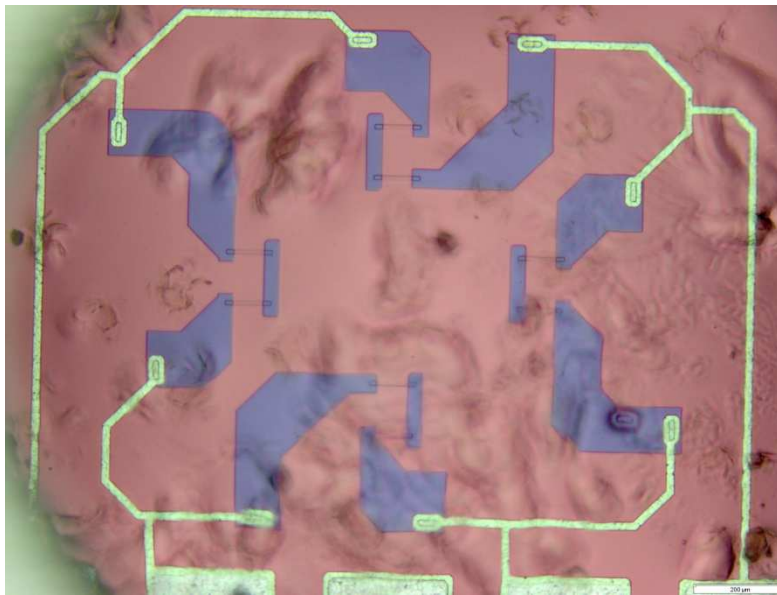


Figura 29 – Imagem microscópica da membrana microfabricada do sensor de pressão MPL115A2.

3.1.4 MÓDULO DE COMUNICAÇÃO SEM FIO

O módulo de comunicação sem fio selecionado para esta aplicação foi o Xbee-PRO da fabricante Digi. Ele opera na faixa de 2.4GHz (frequência ISM) obedecendo ao padrão IEEE 802.15.4 e ainda opera em modo *unicast* (quando um módulo se comunica com outro, formando um par) e em modo *broadcast* (quando um módulo recebe dados de todos os módulos e todos recebem dados vindos dele). Este módulo possui um microcontrolador interno e uma interface que permite receber sinais digitais e analógicos e transmiti-los a outro módulo ligado em rede. Para comunicação, ele possui interface UART para estabelecer a comunicação de modo transparente entre o nó sensor e nó coordenador [42]. Para controle de energia, o módulo possui um pino de hibernação (*sleep*), pelo qual se ativa ou desativa o modo de baixo consumo configurado no programa. Ele também possui um pino que fornece um sinal PWM, cuja largura representa a intensidade do sinal que chega ao receptor (*RX Signal Strenght Indicator, RSSI*). A porcentagem (medida pela largura do pulso) reflete o nível do sinal recebido (em dB) que esteja acima nível de sensibilidade do módulo. A tabela IV mostra a relação entre esses dois valores.



Figura 30 – Módulo de comunicação sem fio Xbee-Pro.

Tabela IV – Relação entre porcentagem do nível do sinal e seu valor acima da sensibilidade (em dBm).

Nível de sinal (%)	Nível do Sinal (dB)
0	Sinal não recebido
1 – 24	≤ 0 (nível igual ou abaixo da sensibilidade mínima)
41	10
58	20
75	30

Para a nossa aplicação, o nó utiliza a interface UART e o pino de hibernação (*sleep*), enquanto que o coordenador utiliza o pino indicador da intensidade do sinal recebido e a interface UART para receber os dados transmitidos pelos nós. O coordenador utiliza um módulo configurado em modo *broadcast* para receber os dados de todos os nós da rede. Isso não significa, contudo, que o coordenador receberá os dados de todos os nós que estejam ao alcance. Para que os nós sensores enviem os dados ao coordenador, os módulos instalados em ambos devem operar no mesmo canal de comunicação. Isso habilita o uso de até 12 redes sem fio utilizadas simultaneamente na mesma área, sem que haja interferência entre redes.

3.1.5 MICROCONTROLADOR DO COORDENADOR DA REDE

Para o coordenador, a escolha do microcontrolador levou ao uso do LPC2378, que engloba o núcleo ARM7 de 32 bits. Ele possui 512 kB de memória flash e 32 kB de memória RAM. Este componente contém os seguintes periféricos, necessários para a utilização no projeto [55]:

- ✓ Interface Ethernet;
- ✓ Interface UART;
- ✓ Relógio de tempo real(RTC);
- ✓ Interface de cartão de Memória (*Memory Card Interface*, MCI).

Com a interface Ethernet, as leituras podem ser acessadas remotamente através de uma página *web* embarcada. Um IP estático é configurado internamente e rotinas de servidor HTTP são implementadas, enquanto que um arquivo à parte armazena a página *web* em formato HTML, que é atualizada com os dados correntes conforme uma atualização (automática ou manual) é realizada. A interface serial UART é configurada para receber os dados do módulo sem fio com as informações de temperatura, umidade e pressão de todos os sensores presentes na rede. O RTC fornece a data e o horário em segundos para que o nó coordenador possa gravar as leituras juntamente com a data e hora correspondentes ao momento no qual elas foram realizadas. Finalmente, a interface MCI é utilizada para a interação do coordenador com os

dados contidos em um cartão SD ou MMC. Ela possibilita ao controlador a criação de novos arquivos, leitura de arquivos já existentes e registro dos dados referentes às leituras efetuadas pelo sensor. A versão de gerenciamento dos dados do cartão de memória está configurada para suportar tabelas de arquivos no formato FAT32 (*File Allocation Table*), que implica o uso de cartões de até 2GB de capacidade.

Estas funções encontradas no microcontrolador permitem que o controlador da rede sem fio assuma simultaneamente a função de *gateway* e de *datalogger*. Isso leva ao atendimento do requisito de baixo custo, uma vez que são substituídos dois equipamentos por um só, sem que seja utilizado um computador ou notebook em campo. Ainda que não haja interface Ethernet ou infraestrutura de rede disponível, a rede pode operar normalmente. Desse modo, O coordenador opera apenas como *datalogger*, registrando as medidas transmitidas pelos nós.

3.1.6 1º PROTÓTIPO DO NÓ SENSOR

Os primeiros protótipos do nó sensor foram construídos a partir de placas genéricas de desenvolvimento rápido contendo o microcontrolador escolhido. Nelas, os sensores e módulos de comunicação foram adicionados para que fosse possível testar o *firmware* desenvolvido, todas as comunicações estabelecidas e os elementos de *hardware* adicionais necessários para o funcionamento do circuito. A figura mostra um dos protótipos iniciais do nó sensor.

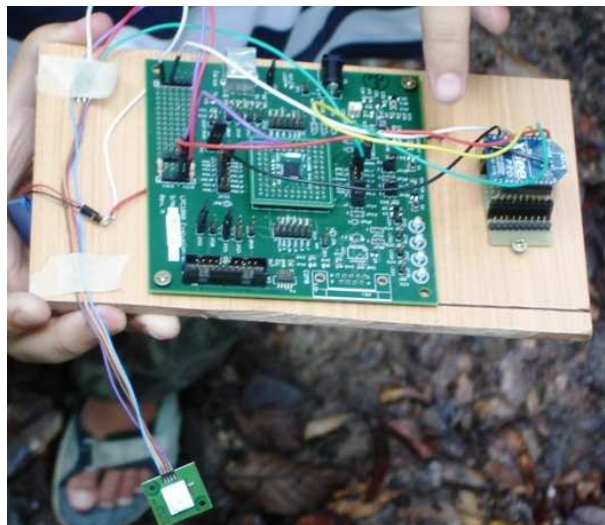


Figura 31 – Primeiro protótipo do nó sensor em placa genérica.

3.1.7 2º PROTÓTIPO DO NÓ SENSOR

A placa utilizada no nó levou em conta fatores como facilidade de manuseio e dimensão reduzida. A figura 32 mostra o leiaute da placa. Segundo essas características, a versão inicial da placa foi projetada contendo:

- ✓ Terminais de acesso: a placa possui 6 terminais de acesso, dispostos de maneira linear. Os terminais 1 e 2 permitem a alimentação da placa através de uma fonte externa. Os pinos 3 e 4 permitem o acesso aos pinos de gravação e depuração do *firmware*. Finalmente, os pinos 5 e 6 permitem a carga da bateria em separado do circuito, através da seleção da chave;
- ✓ Formato arredondado: este formato foi definido para se adequar a alguns processos de injeção plástica e também para facilitar o uso de encapsulamentos que favoreçam a aplicação em monitoramento de ambientes específicos como silos de grãos;
- ✓ Chave seletora: é responsável por posicionar a bateria para os estados de carga e de operação. O estado de carga também pode ser definido como estado de programação;
- ✓ Dois leds (Verde e Vermelho): São responsáveis por indicar transmissão das informações e por identificar o estado do nó. O led verde indica que o nó está enviando as medidas. O led Vermelho indica se o nó está em estado de programação (ligado) ou operacional (desligado). No início da operação do nó, os dois leds piscam ;
- ✓ Resistores e capacitores: São responsáveis por diversas funções no circuito, dentre elas *pull-up* para as saídas de coletor aberto, limitadores de corrente para os leds e filtros para eliminar ruídos de chaveamento;
- ✓ Conector para o módulo de comunicação: foi definido o uso de conectores para o módulo de comunicação para poder facilitar a programação do módulo fora da placa e para permitir que houvesse um espaço abaixo do módulo para abrigar a bateria.

A seguir, são apresentados o diagrama em blocos do circuito, o leiaute da placa de circuito impresso, o esquema elétrico do nó sensor e uma fotografia da placa do nó sensor montada.

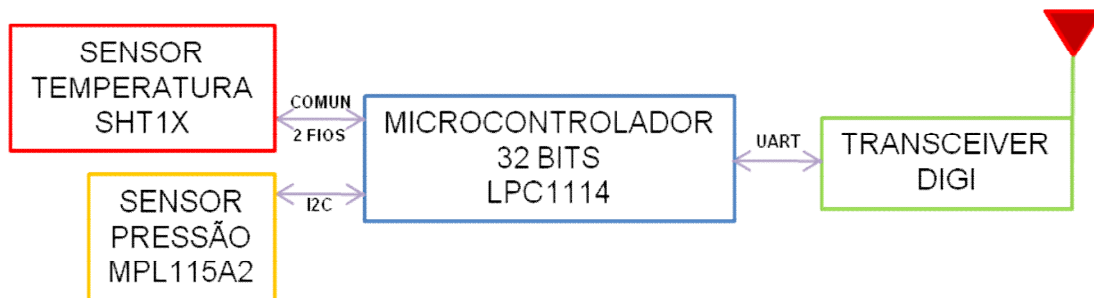


Figura 32 – Diagrama em blocos do nó sensor.

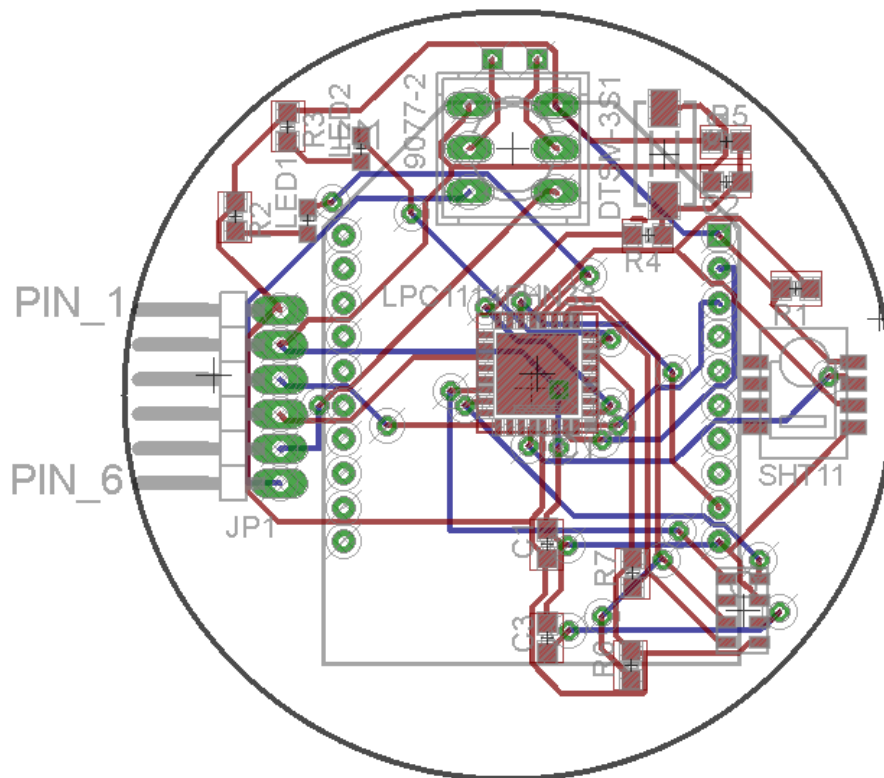


Figura 33 – Leiaute da placa de circuito impresso do 2º protótipo do nó sensor.

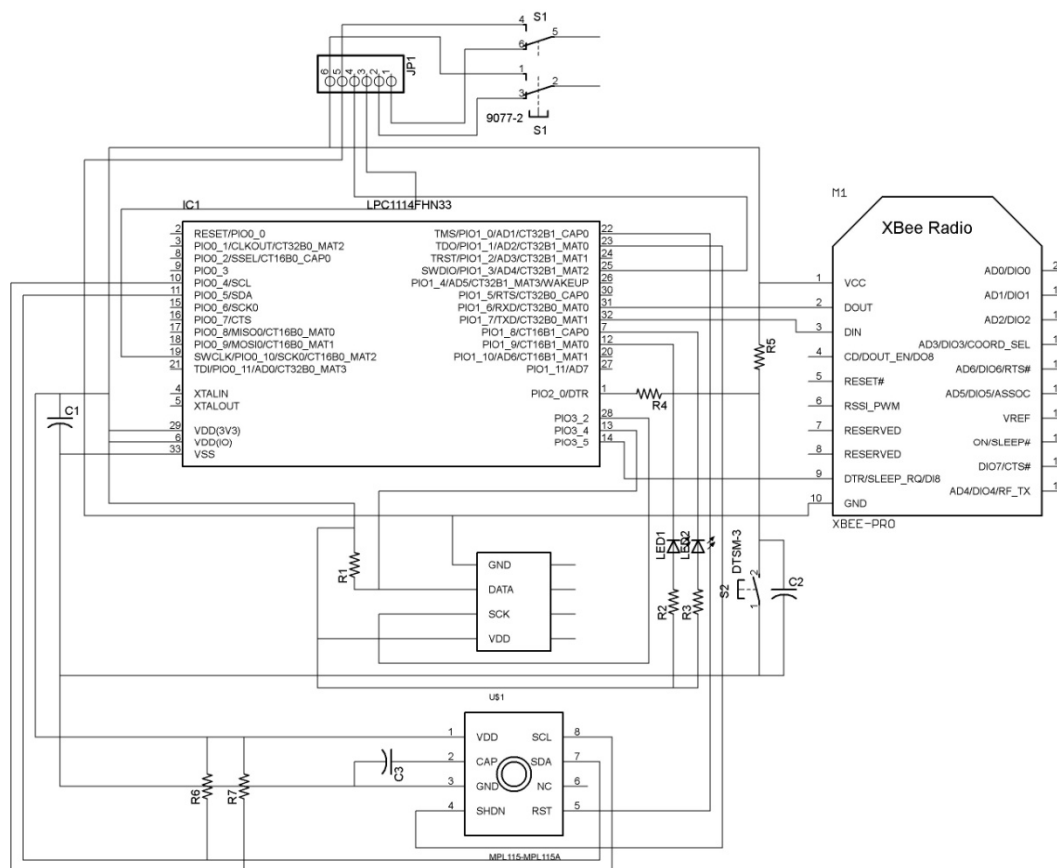


Figura 34 – Esquema elétrico do 2º protótipo do nó sensor.

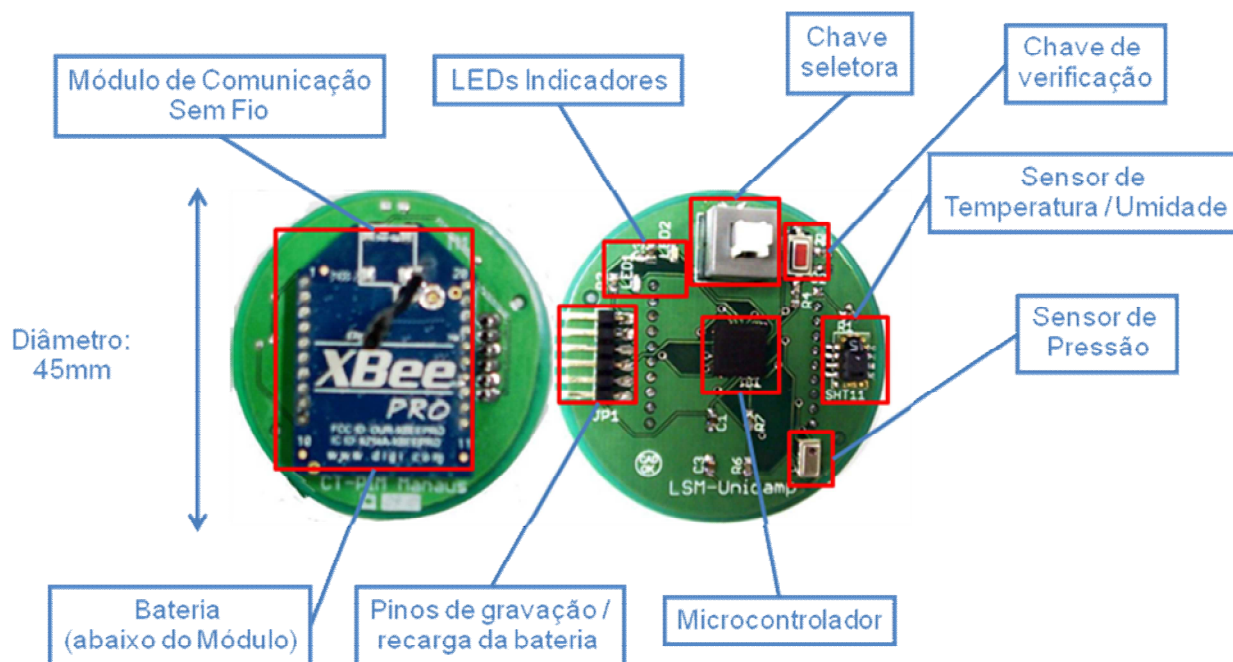


Figura 35 – Detalhes da construção do 2º protótipo do nó sensor já montado.

3.1.8 3º PROTÓTIPO DO NÓ SENSOR

O segundo protótipo do nó sensor apresentou vantagens em seu tamanho e funcionalidade. Porém, sua resposta à mudança de temperatura e umidade se mostrou lenta quando comparada a uma referência (vide Capítulo 4). Por isso, foi construído uma nova versão do nó sensor, separando o sensor em uma placa própria e isolando o sensor da placa através de dois cortes laterais, reduzindo assim a condutividade térmica da placa para o sensor. Os circuitos e conexões se mantiveram os mesmos, o que possibilitou a manutenção da versão de firmware idêntica à utilizada na versão inicial. Também foi utilizado um disco de nylon na cobertura da placa de sensores para reduzir o efeito da incidência direta no Sol sobre a medida do sensor, que está alojado abaixo da placa de circuito impresso, como mostrado na figura 36b.

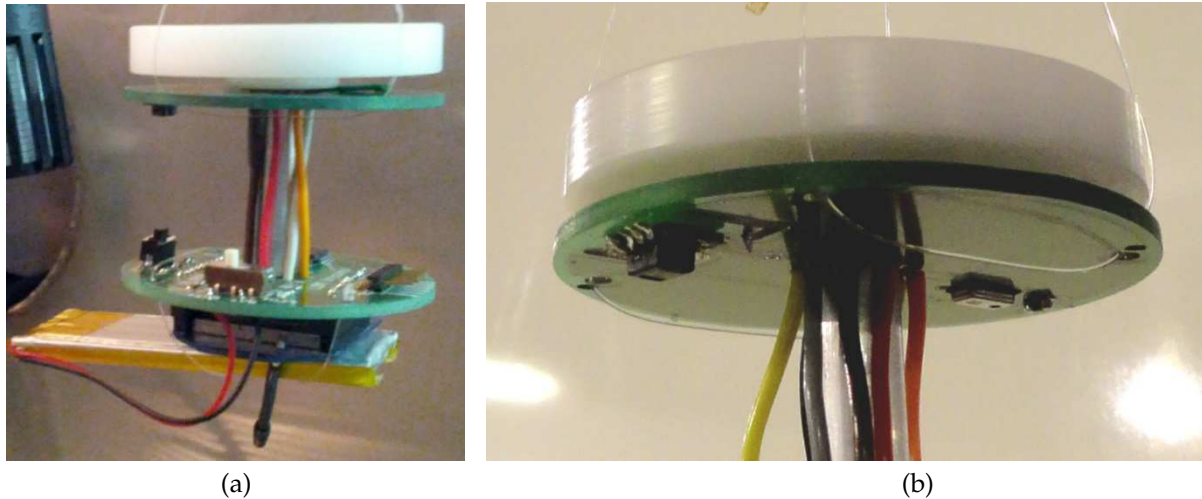


Figura 36 – 3º protótipo do nó sensor (a) com detalhe da placa de sensores (b).

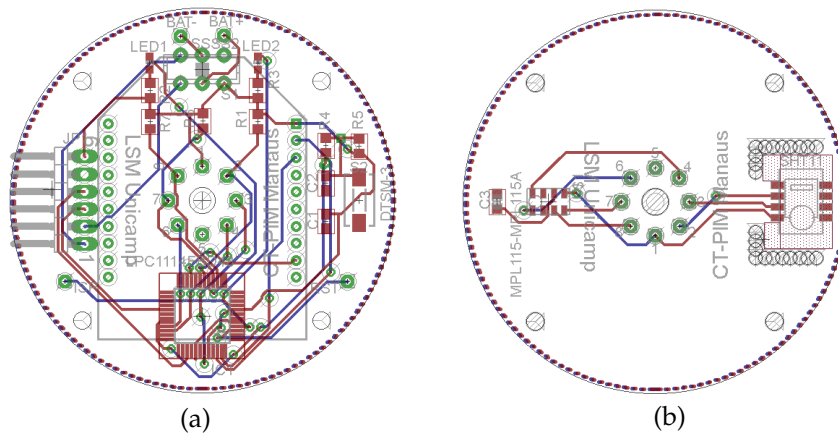


Figura 37 – Leiaute da placa principal do nó (a) e da placa de sensores (b) do 3º protótipo.

3.1.9 BATERIAS E CIRCUITO CARREGADOR DE BATERIAS

As baterias utilizadas no projeto são de íons de Lítio Polímero. Elas foram adaptadas sem dificuldades ao nó sensor, sendo que para cada nó foi utilizada uma bateria com capacidade diferente para poder comparar a autonomia conseguida com cada uma. Porém, para poder utilizá-las corretamente, é preciso se certificar que elas foram carregadas corretamente e no seu limite máximo. Não foi encontrado, contudo, um carregador comercial específico para estas baterias de íons de lítio entre 120 mAh e 700 mAh. Por isso, foi necessário construir um circuito à parte para carregar as baterias e certificar que os nós poderiam funcionar a plena carga.

O carregador proposto seguiu as diretrizes das notas de aplicação da NXP AN10338, que descreve um carregador de bateria de íon de lítio utilizando um microcontrolador. O carregador construído utiliza um microcontrolador de 32 bits que alimenta um conversor Buck com um sinal PWM e recebe via ADC os valores de corrente através da leitura da tensão de um resistor em série com a bateria. Durante o processo, é necessário seguir três etapas de carga, em sequência [56]:

- ✓ Etapa de pré-carga: uma corrente de 65 mA é aplicada à bateria até que ela alcance o valor de tensão mínima de 3 V;
- ✓ Etapa de Carga a Corrente Constante: A corrente da bateria é limitada a 350 mA , enquanto que a tensão da bateria aumenta até o valor de 4,23 V, aproximadamente;
- ✓ Etapa de Carga a Tensão Constante: A tensão é estabilizada em 4,23 V, enquanto a corrente decresce até o valor mínimo de 30 mA, finalizando a carga total da bateria;

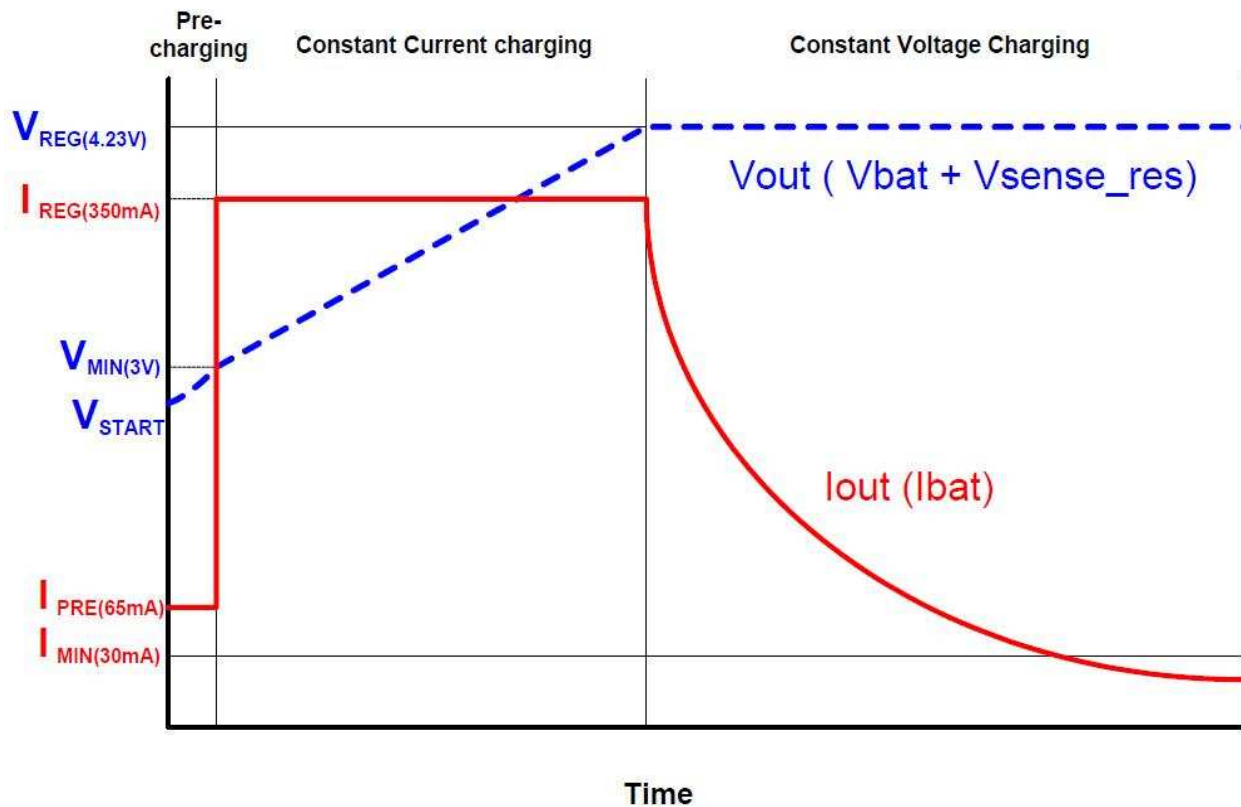


Figura 38 – Descrição das etapas de carga da bateria.

As figuras 39 e 40 mostram, respectivamente, o esquemático e o circuito implementado utilizando uma placa da LPCXpresso contendo o microcontrolador LPC1114.

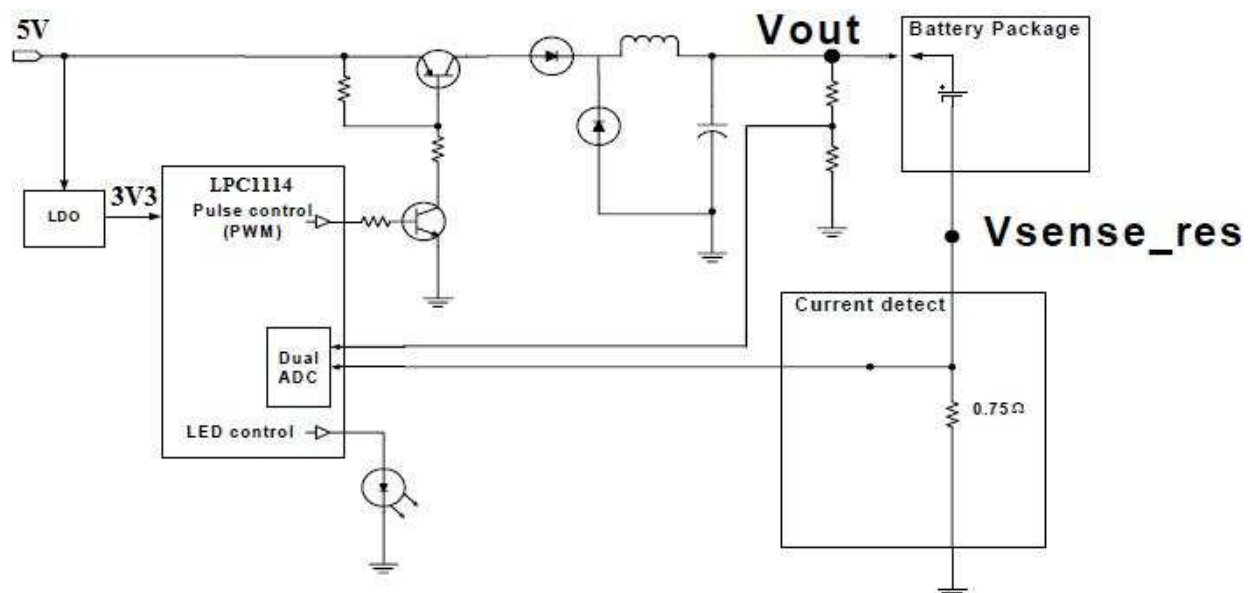


Figura 39 – Diagrama esquemático do carregador de baterias utilizando o LPC1114.

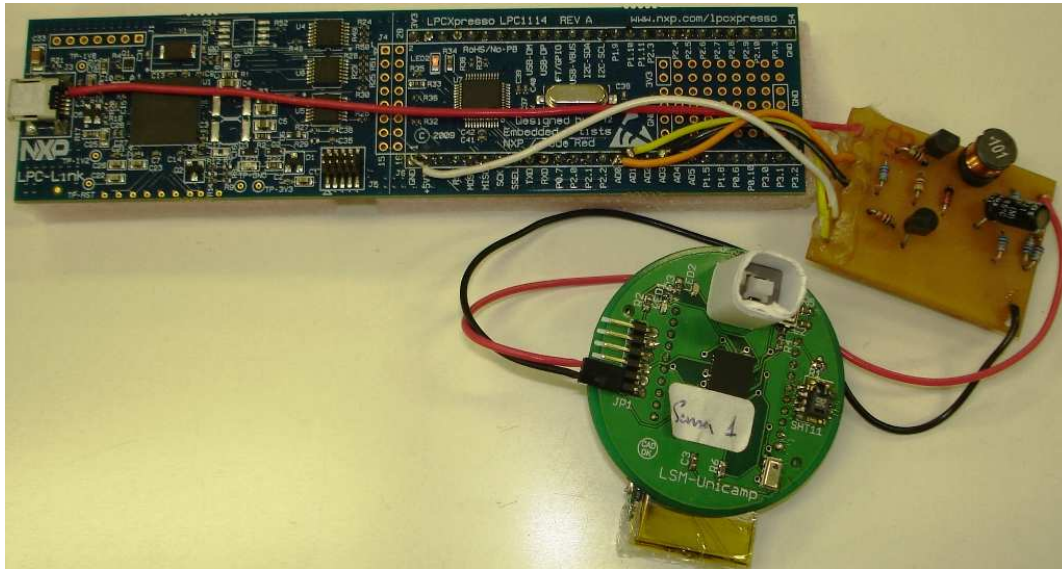


Figura 40 – Circuito implementado para carga da bateria utilizando a placa LPCXpresso.

O *firmware* implementado, além do controle e monitoramento da tensão da bateria, indica, através de um *led*, qual o estado do sistema, isto é, o estado de espera por uma bateria ser conectada (*led* piscando com frequência alta), estado de carga (*led* piscando com frequência baixa), ou estado de carga concluída (*led* apagado).

3.2 Modo de operação da Rede de Sensores

Para o usuário final da rede de sensores, as tarefas de preparação são mínimas. Antes de se ligar a placa do coordenador da rede, é necessário inserir o cartão de memória no conector correspondente. Após isso, pode-se alimentar o circuito e iniciar o processo de ativação dos nós. O estado de programação no nó equivale ao nó desligado e ao se comutar para o estado de operação, a bateria fornece energia ao sistema, que passa a realizar as leituras dos sensores e sua transmissão de modo temporizado. A temporização do nó pode variar de acordo com a necessidade da aplicação.

3.2.1 OPERAÇÃO DO NÓ

O *firmware* embarcado no nó realiza, no início da operação, todas as preparações necessárias para o ciclo de leituras e gravação, a saber:

- ✓ Preparação da interface UART;
- ✓ Preparação da interface I2C;
- ✓ Preparação dos pinos de entrada e saída;
- ✓ Preparação e configuração do sensor de temperatura e umidade;
- ✓ Preparação e configuração do sensor de pressão barométrica;
- ✓ Configuração dos temporizadores;
- ✓ Preparação do monitoramento da bateria;
- ✓ Início do modo de baixa energia.

Após iniciar o modo de baixa energia, o microcontrolador aguarda o temporizador do módulo *Watchdog* atingir seu valor configurado para então iniciar as leituras. É feito primeiramente a leitura de temperatura e umidade do sensor de modo bruto e em seguida se realizam os cálculos para transformar os dados em temperatura ambiente e umidade relativa, ambos com resolução em centésimos e armazenados em duas variáveis distintas em forma de *string*, isto é, já preparada para transmissão através da porta serial em caracteres ASCII. Logo após é feita a aquisição dos valores de pressão e temperatura do sensor de pressão atmosférica, bem como dos valores dos coeficientes para compensação da pressão. Após o cálculo da compensação, é realizado o cálculo para obtenção do valor da pressão em kPa com a compensação da temperatura e o valor obtido é guardado em uma outra variável em formato *string*. O valor do estado da bateria é definido pelos valores de 1 a 4, em ordem crescente, que apontam de 25 a 100% de energia na bateria.

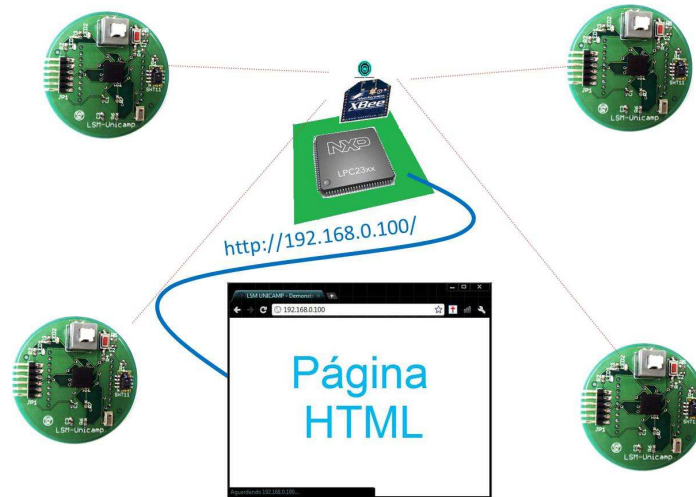


Figura 41 – Diagrama de exemplo de comunicação entre os nós sensores e coordenador.

Já com as variáveis medidas e tratadas, o microcontrolador está habilitado a transmiti-las ao módulo de comunicação que por sua vez os transmite ao coordenador. Como a rede está em formato estrela (com o coordenador operando em modo broadcast), o nó deve identificar cada informação como sendo sua própria, através de um rótulo (*tag*), a fim de evitar que, ao chegar ao coordenador, as informações sejam mal interpretadas. Além disso, um rótulo também se faz necessário para identificar qual medida (temperatura, umidade e pressão) está sendo enviada. Com isso, estabelece-se uma codificação de transmissão que permite identificar cada informação que está sendo enviada, como se discrimina a seguir na tabela V.

Tabela V – Detalhamento da codificação de dados do nó.

TA24.72		
Dado recebido: A primeira letra se refere ao tipo de dado recebido, onde:	Identificação do sensor: A segunda letra indica de que sensor está vindo a informação:	Valor correspondente ao tipo de informação recebida. Neste caso, sabe-se que a temperatura do sensor 1 é de 24.72 °C. O único valor inteiro recebido é o estado da bateria, que varia entre 0 e 4 (vide tabela V)
T : temperatura	A : dados do sensor 1	
H : umidade	B : dados do sensor 2	
P : Pressão Atmosférica	C : dados do sensor 3	
B : Estado da Bateria	D : dados do sensor 4	

Após a transmissão dos dados, o nó retorna ao estado de espera em baixa energia e aguarda para recomençar o processo.

3.2.2 OPERAÇÃO DO COORDENADOR

O firmware embarcado no coordenador realiza as seguintes tarefas iniciais:

- ✓ Preparação da interface UART;
- ✓ Preparação do relógio (caso esteja inválido);
- ✓ Preparação da interface de acesso ao cartão SD (inicialização e criação/abertura dos arquivos);
- ✓ Preparação dos temporizadores;
- ✓ Preparação do acesso aos dados de montagem da página em formato HTML (protocolo TCP/IP).

O microcontrolador realiza troca de dados com três elementos:

- ✓ Módulo de comunicação sem fio: através dele são recebidos os dados dos nós sensores;
- ✓ PHY de ethernet para acesso à rede: o PHY é responsável por receber e transmitir os dados de acordo com o padrão da interface Ethernet;
- ✓ Cartão de memória: Os dados são gravados e registrados no momento da sua chegada.

O diagrama em blocos da figura 42 ilustra os elementos do nó coordenador.

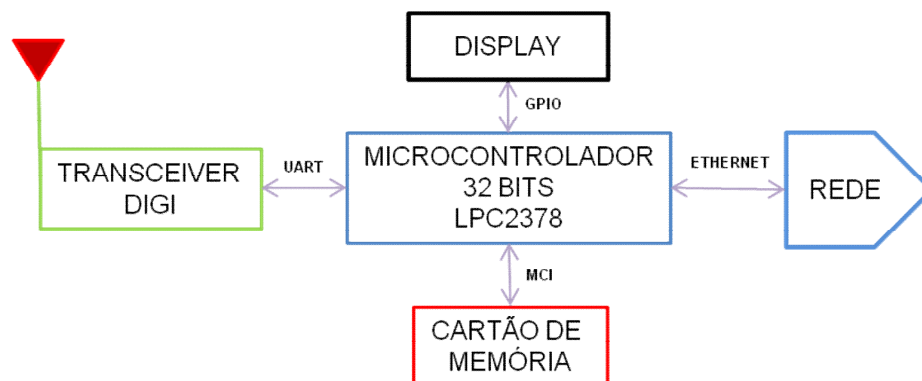


Figura 42 – Diagrama em blocos do nó coordenador da rede.

Quatro dos três temporizadores disponíveis no microcontrolador são utilizados. O temporizador 3 é o mais prioritário e é responsável por verificar a cada 1 ms se há dados sendo requisitados pela rede LAN. Em caso positivo, todos os dados disponíveis são disponibilizados em formato HTML para a posterior leitura em um browser. O temporizador 2 verifica a cada meio segundo se há dados a serem classificados vindos dos nós. Para cada informação recebida, há o registro no cartão de memória da data e da hora em que os dados chegaram e seu armazenamento para eventual requisição da rede LAN. Os dados anteriores são substituídos pelos mais recentes e os dados sendo visualizados através da LAN são os últimos a serem disponibilizados, ficando a cargo do cartão de memória registrar o histórico das medidas. O temporizador 1 é o menos prioritário e é responsável por medir qual o nível de sinal com o qual cada nó está estabelecendo comunicação com o coordenador. Ele é ativado assim que o microcontrolador recebe algum dado da porta serial e é ajustado para monitorar qual o valor do ciclo de trabalho do PWM do sinal gerado pelo módulo de comunicação.

Assim que chegam os dados referentes a um sensor, a interface UART os recebe e os armazena em um buffer. Imediatamente, o microcontrolador inicia o cálculo da intensidade do sinal referente aos dados recebidos, fazendo o tratamento do sinal PWM proveniente do pino de RSSI. Neste momento, no entanto, não se sabe a quem pertencem aqueles dados. Porém, uma vez que eles possuem o ID referente ao sensor de onde partiram, eles podem ser identificados durante a classificação. Os valores de níveis de sinal obtidos são armazenados em um buffer secundário na mesma ordem de chegada dos dados, garantindo que ao serem classificados, os níveis de sinal sejam atribuídos e armazenados corretamente aos respectivos sensores.

Uma vez que as medidas já foram recebidas, classificadas e armazenadas, o sistema está apto a fornecer os dados via rede local LAN se assim for requisitado. O usuário pode acessar o coordenador através de um browser, utilizando como endereço o IP configurado no *firmware*. No caso de acesso via Internet, é necessário que o IP configurado seja disponibilizado através de um Serviço de Nome Dinâmico (*Domain Name Service* – DNS). Na requisição dos dados, uma página em formato HTML é disponibilizada com os últimos dados recebidos pelo coordenador. Os dados são disponibilizados por ordem de sensor (1 a 4) com a seguinte sequência:

- ✓ Temperatura (em °C);

- ✓ Umidade (em %);
- ✓ Pressão (em kPa);
- ✓ Estado da bateria (em cores);
- ✓ Nível de sinal (em %, conforme sinal do pino de RSSI);
- ✓ Data e hora do último dado recebido (no formato DD/MM/AAAA HH:MM:SS).

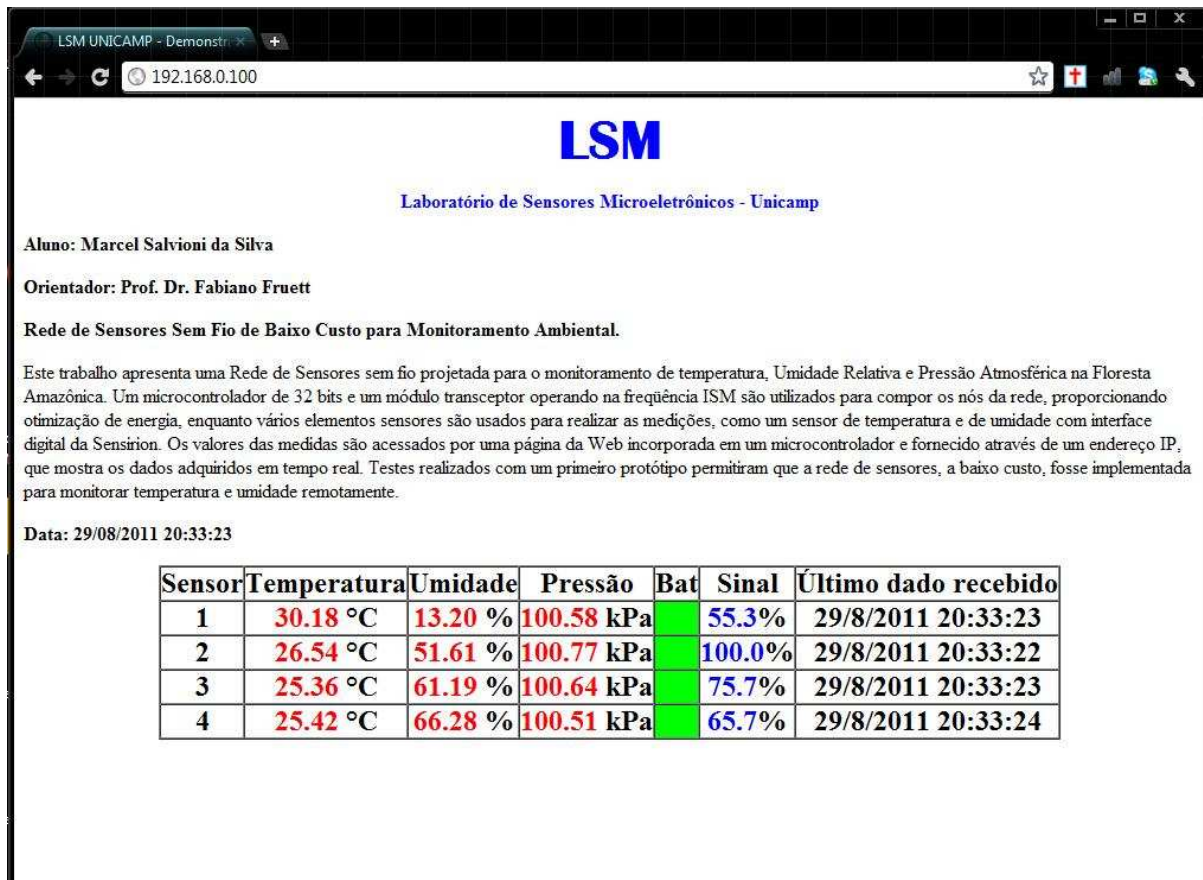


Figura 43 – Página *web* embarcada com as leituras e os dados dos nós disponibilizados.

As cores do estado de bateria foram selecionadas para fornecer uma informação visual, já que a informação não é fornecida em volts. As cores são discriminadas na tabela VI:

Tabela VI – Código das cores para o estado da bateria no nó.

Cor presente	Estado da Bateria (%)	Voltagem estimada (V)
Verde	100	$V_{bat} > 2,9$
Amarelo	75	$2,8 < V_{bat} < 2,9$
Vermelho	50	$2,52 < V_{bat} < 2,66$
Laranja	25	$2,22 < V_{bat} < 2,35$
Preto	0	$1,65 < V_{bat} < 1,8$

Encerra-se aqui o capítulo que descreve o desenvolvimento da rede de sensores sem fio proposta, sua construção e seu modo de operação. Nos apêndices B e C encontram-se informações referentes aos sensores.

Capítulo 4

Resultados Experimentais

Neste capítulo estão descritos todos os resultados dos experimentos realizados para o desenvolvimento da Rede de Sensores Sem Fio. Embora o nó sensor e o nó coordenador tenham sido desenvolvidos em conjunto, eles serão abordados neste capítulo de forma separada. O sistema foi testado em cinco locais distintos, sendo:

- ✓ Laboratório de Sensores Microeletrônicos (LSM): localizado na Faculdade de Engenharia Elétrica e de Computação (FEEC) da Universidade Estadual de Campinas (UNICAMP);
- ✓ Laboratório de Caracterização e Confiabilidade de Circuitos Integrados (LCC-CI): um dos laboratórios do Centro de Ciência, Tecnologia e Inovação do Pólo Industrial de Manaus (CT-PIM);
- ✓ ZF2: Torre de 51 metros de aquisição de medidas ambientais do INPA (Instituto Nacional de Pesquisas da Amazônia), situado no Km 43 da BR-174 do Ramal do Distrito Agropecuário da Superintendência da Zona Franca de Manaus (SUFRAMA);

- ✓ ZF3: Ramal do Distrito Agropecuário da SUFRAMA com Torre de 43 metros de aquisição de medidas ambientais do INPA situado no Km 78 da BR-174;
- ✓ Campo Experimental da EMBRAPA (Empresa Brasileira de Pesquisa Agropecuária): Situado no Centro de Pesquisas da Embrapa, com lançamento de radiossondas e a elevação controlada de balões Atmosférico e acompanhamento do departamento de Meteorologia da UEA (Universidade do Estado do Amazonas);

4.1 Testes dos nós sensores

Os primeiros testes com os nós sensores foram realizados utilizando os protótipos montados sobre a placa LPCXpresso (contendo o microcontrolador LPC1114, na versão de 44 pinos) e sobre a placa de validação NXP(contendo o mesmo microcontrolador na versão de 33 pinos).

Os módulos X-Bee Pro foram configurados da seguinte forma:

- ✓ Taxa de transmissão: 115200 bps;
- ✓ Nível de potência de transmissão (*power level*, PL): 0 (10 dBm);
- ✓ Canal utilizado: 10;
- ✓ Versão de Firmware: 10E8

A taxa de transmissão é considerada alta e foi definida deste modo para que o tempo gasto na transmissão fosse o mínimo possível, minimizando o consumo de energia. O nível de potência também foi configurado para se reduzir o consumo de bateria.

4.1.1 COMUNICAÇÃO E ALCANCE

Os testes de comunicação foram realizados para verificar se o módulo de comunicação da Digi apresentava as características de conexão descritas na sua folha de dados. Outro objetivo foi verificar quais as implicações do uso desses módulos comunicando em um ambiente como a floresta, com mata densa e solo úmido.

4.1.1.1 Teste na torre ZF3

O primeiro teste de comunicação foi realizado na torre de 43 metros localizada na estrada ZF3. Um dos sensores foi posicionado na base da torre e o outro no topo da torre, conforme mostrado na figura 43. Ambos tiveram o sensor da Sensirion configurados para converter as grandezas (temperatura e umidade) em palavras digitais com 14 bits e 12 bits, respectivamente.

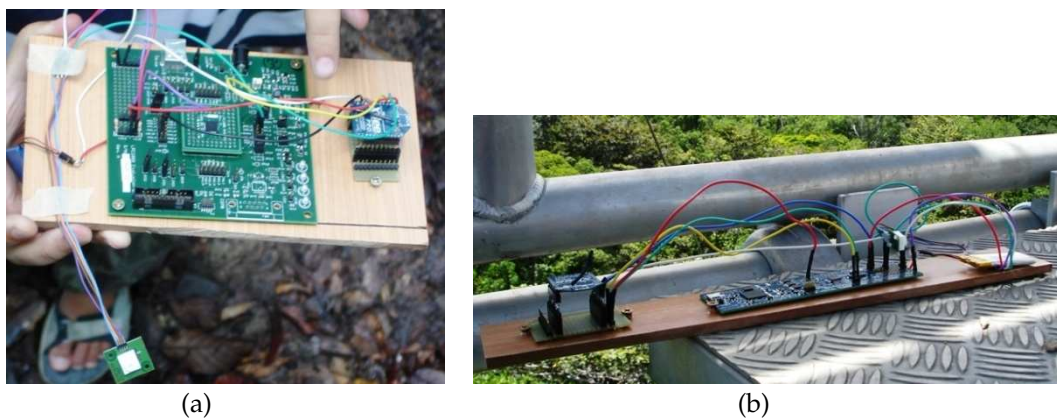


Figura 44 – Protótipos posicionados na base (a) e no topo (b) da torre.

Os dados transmitidos ao coordenador foram visualizados em tempo real através da página *web* mostrada na figura 44.

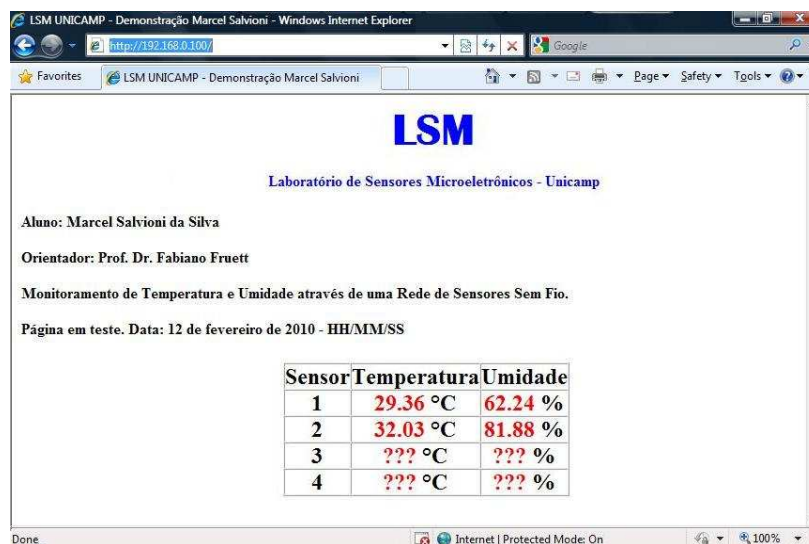


Figura 45 – Tela do navegador obtida durante os testes na torre da estrada ZF3.

Apesar de se tratar do primeiro protótipo utilizando placas genéricas, neste teste já foi possível identificar as diferenças de temperatura e umidade da base e do topo da torre. Nesta etapa corrigiu-se também um problema na aquisição dos dados por parte do nó coordenador

que invertia os valores de umidade lidos pelos sensores. Ao serem comparadas entre si, observou-se uma coerência com os valores típicos. Na comparação dos valores obtidos pelos nós com os medidos pelos sensores já instalados na torre, observou-se que os valores de temperatura medidos pelo nó estavam entre 1,5°C a 1,7°C acima dos registrados pelos sensores da torre e os valores de umidade medidos pelos nós estavam de 4,5% a 8% abaixo dos registrados pelos sensores da torre, nas mesmas posições de medida, conforme mostrado na tabela VII:

Tabela VII – Valores de temperatura e umidade da torre na ZF3.

Grandeza	Valor Medido pelo protótipo	Valor medido pelos sensores da torre
Temperatura no topo (°C)	32,03	30,45
Umidade no topo (%)	62,24	67,1
Temperatura na base (°C)	29,36	27,67
Umidade na base(%)	81,88	89,6

Os valores obtidos fora da tolerância do sensor foram atribuídos a alguns fatores internos e externos, como :

- ✓ Radiação térmica: O calor acumulado pela estrutura metálica da torre acabou por submeter o sensor a uma condição fora do normal para aquele ambiente, especialmente no topo da torre onde a incidência de Sol sobre a estrutura metálica é mais intensa;
- ✓ Radiação solar: Dependendo da posição do protótipo (também no topo da torre), havia incidência de luz solar diretamente sobre o sensor, o que influenciou as medidas de temperatura e umidade;
- ✓ Alta taxa de amostragem: O sensor foi configurado inadvertidamente para uma amostragem de 1 medida de temperatura e umidade a cada segundo com resolução máxima possível para as medidas (14 bits para temperatura e 12 bits para umidade relativa). Apesar de responder os comandos a tempo para cada medida, as leituras são poluídas com o processo de espera e auto-aquecimento do sensor para medida

de umidade. Após estudo, verificou-se que o sensor não deve estar ativo – realizando medidas – mais do que 10% do tempo, para que o processo de auto-aquecimento não represente mais que 0,1 °C no valor lido de temperatura. Para conferir qual o tempo mínimo de intervalo entre as leituras, devem-se utilizar os valores de tempo de ativação do sensor para cada leitura que variam conforme a resolução que se utiliza para se realizar a medida. Os valores dos intervalos e as respectivas resoluções estão na tabela VIII [53]:

Tabela VIII – Valores de referência para intervalo mínimo entre medidas.

Resolução (bits)	Tempo de espera para leitura (ms)	Intervalo mínimo entre leituras (ms)
8	20	200
12	80	800
14	320	3200

Com os testes realizados na Torre do ramal ZF3, foi possível identificar as principais mudanças a serem inseridas no *firmware* para leituras de umidade e temperatura. Além disso, foi observado o impacto da falta de um encapsulamento para se medir as variáveis ambientais com maior exatidão.

4.1.1.2 Teste na torre ZF2

O teste seguinte de comunicação em campo realizado na torre de 51 metros localizada no km 34 na estrada ZF2. Para este teste foram utilizados os mesmos protótipos com as mesmas configurações para o módulo de comunicação e para o sensor temperatura e umidade. O intervalo entre as medidas, porém, foi alterado para 10s, e o sensor da base da torre foi posicionado a 10 m da placa de aquisição. As leituras realizadas pelos sensores foram armazenadas através da placa adaptadora USB-Serial, enquanto que o coordenador recebeu os dados e os exibiu através da tela mostrada na figura 46.

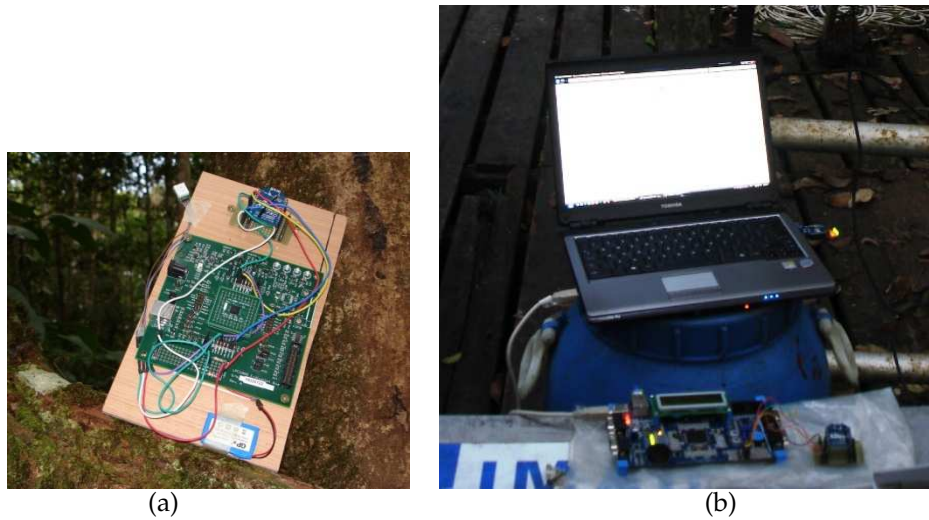


Figura 46 – Protótipos do sensor (a) e do coordenador (b) na base da torre.

As medidas armazenadas foram armazenadas em um arquivo de log que posteriormente preencheram uma planilha eletrônica. Através da planilha, foram obtidos os seguintes gráficos mostrados na figura 47 e 48.

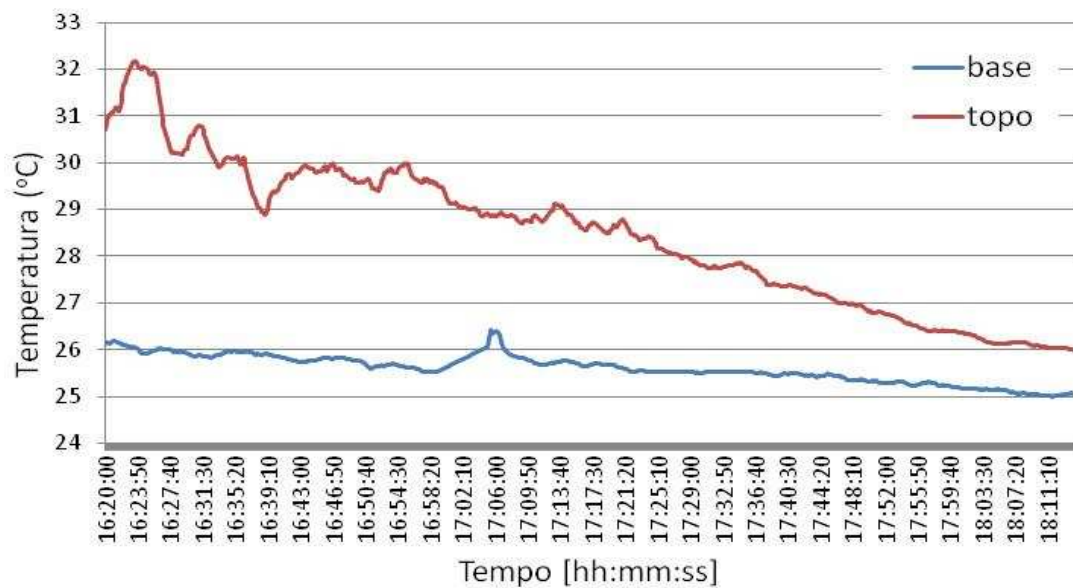


Figura 47 – Temperatura da base e do topo da torre medidas com o 1º protótipo durante teste na torre ZF2.

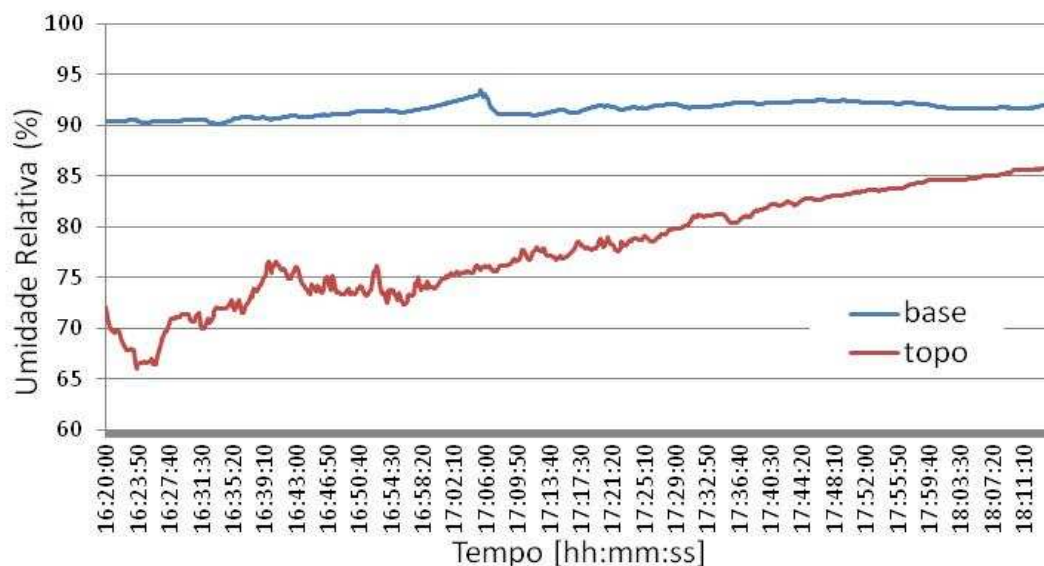


Figura 48 – Umidade Relativa da base e do topo da torre medidas com o 1º protótipo durante o teste na torre ZF2.

A figura 49 mostra os valores medidos pelos sensores de temperatura da torre no mesmo período em que os nós de teste realizaram as medidas. Foi realizada 1 medida a cada 30 minutos, em seis pontos distintos da torre.

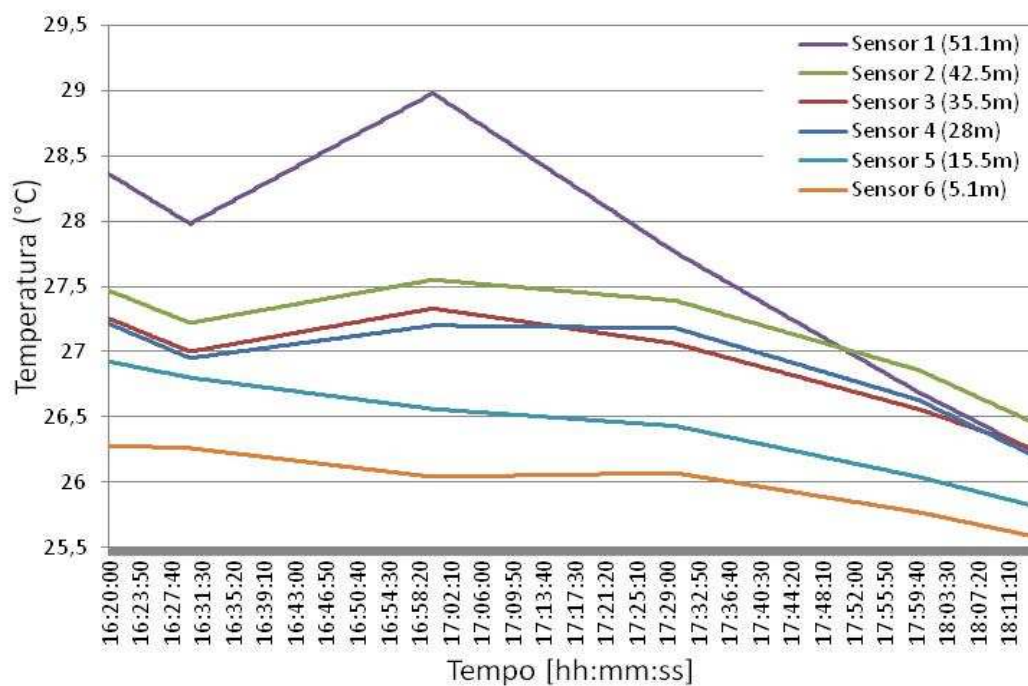


Figura 49 – Temperatura medida ao longo da torre ZF2 durante teste pelos sensores da torre

Tomando as leituras realizadas pelos protótipos de nós sensores e pelos sensores da própria torre, pode-se compará-los ao longo do tempo de medidas e confrontá-los, como é mostrado no gráfico da figura 50.

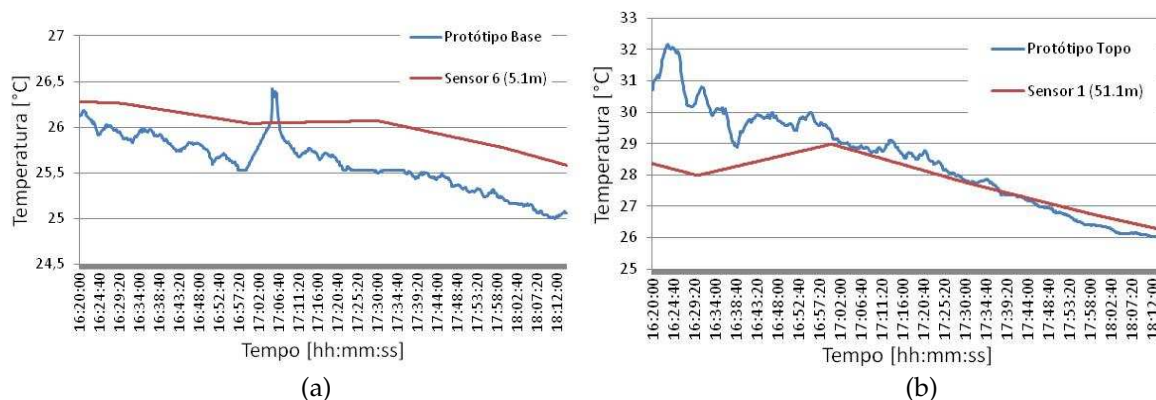


Figura 50 – Temperatura medida ao longo da torre ZF2 durante teste pelos sensores da torre a 5m (a) e 51m (b).

Ao analisar as medidas realizadas durante o período de aproximadamente duas horas (compreendido entre 16:20 e 18:12), foi possível certificar que os valores de temperatura medidos com os protótipos estavam coerentes com os medidos pelos sensores já instalados na torre, respeitadas as tolerâncias e o fato dos sensores não estarem exatamente na mesma posição. Alguns efeitos de radiação térmica e solar foram observados dentre os valores medidos – como o período compreendido entre 16:20 e 17:00 no topo da torre – mas não alteraram demasiadamente o resultado. Também foi observada que em posições diferentes os sensores mantiveram a comunicação sem perda de pacotes durante todo o tempo dos testes.

4.1.1.3 Teste #3 com balão atmosférico

Já com os sensores montados sobre a placa desenvolvida especialmente para o 2º protótipo do nó sensor, realizou-se um teste de comunicação através da sua fixação em um balão atmosférico utilizado para realizar medidas de perfil altimétrico de temperatura, umidade relativa e pressão atmosférica, como parte das pesquisas realizadas pelo Departamento de Meteorologia da Universidade Estadual do Amazonas (UEA). O teste foi realizado no Campo Experimental da Empresa Brasileira de Pesquisa Agropecuária (EMBRAPA), situado no km 29

da rodovia AM-010. Neste experimento, um balão inflado com hélio sobe a uma velocidade de 1 m/s aproximadamente, com uma sonda sem fio equipada com sensores de temperatura, umidade relativa, pressão atmosférica, direção e velocidade do vento, até uma altura máxima de aproximadamente 1000 m, transmitindo seus dados a um receptor específico em FM na faixa de 28 MHz.



Figura 51 – Equipe de Meteorologia da UEA preparando medidas com o balão.

Para este teste, apenas um sensor foi utilizado, configurado para realizar medidas de temperatura, umidade e pressão, em intervalos de 1s. Para isso, funções de leitura do sensor de umidade e temperatura a baixas taxas de amostragem foram definidas no *firmware*. As resoluções finais para cada variável ambiental foram de 12 e 8 bits para temperatura e umidade, respectivamente.

Para este teste, em vez de se utilizar a placa adaptadora USB-serial para se salvar os dados, foi utilizado o próprio coordenador da rede, que incluiu a função de relógio e de gravação das leituras em um cartão de memória. Além disso, foi incorporada a informação sobre a bateria do sensor em código de cores, visualizada durante o acesso à página web embarcada.

As figuras 51 e 52 mostram o gráfico das leituras de umidade e temperatura em função do tempo e comparadas com a pressão atmosférica cujo valor varia com a altitude e serve como indicativo de altura do balão.

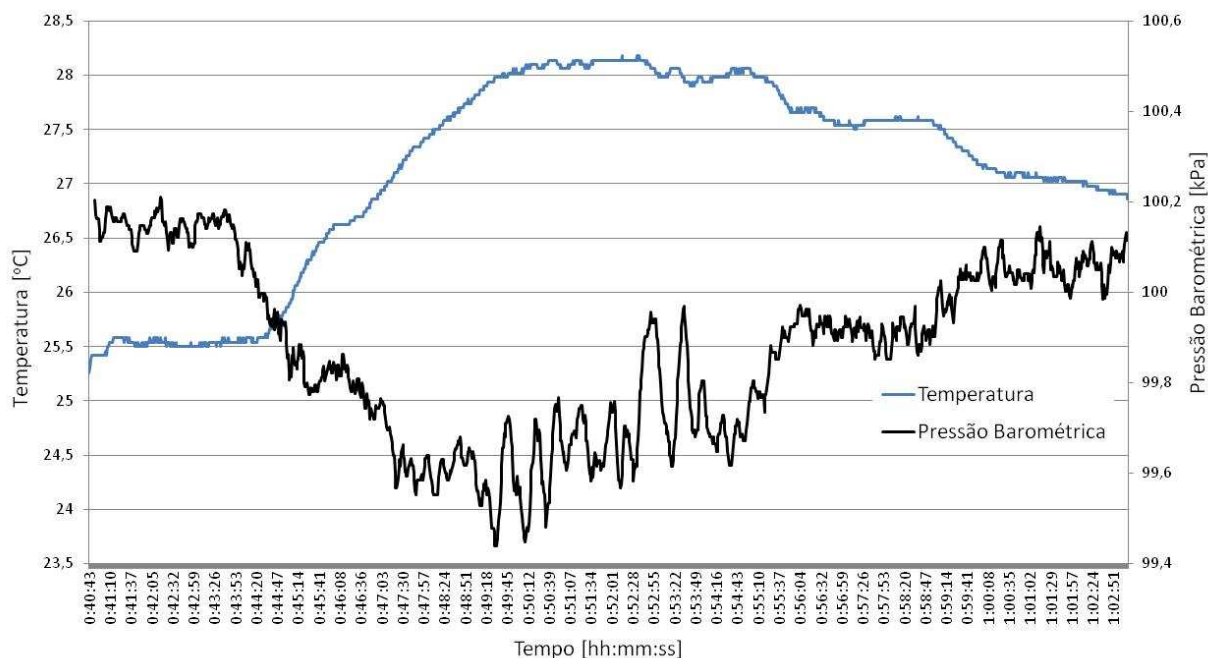


Figura 52 – Temperatura e Pressão Barométrica medidos pelo nó sensor durante a subida do balão.

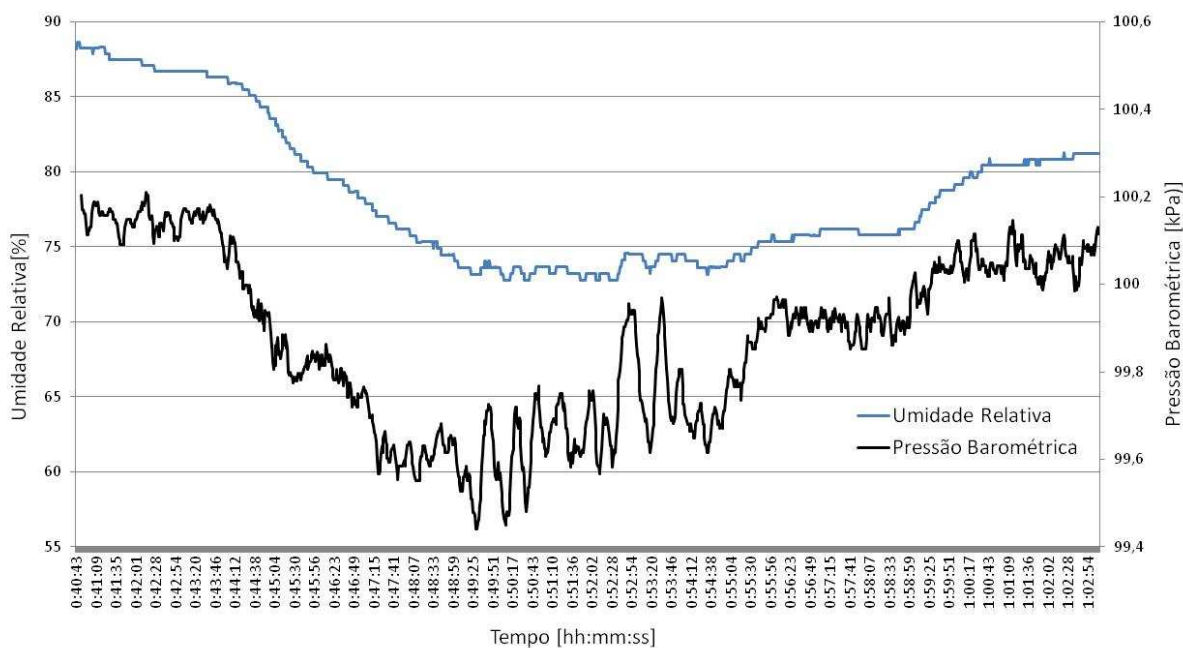


Figura 53 – Umidade e Pressão Barométrica medidos pelo nó sensor durante a subida do balão.

Comparando os valores medidos através do protótipo em desenvolvimento com os valores da sonda utilizada pelo grupo de pesquisa, têm-se os seguintes resultados:

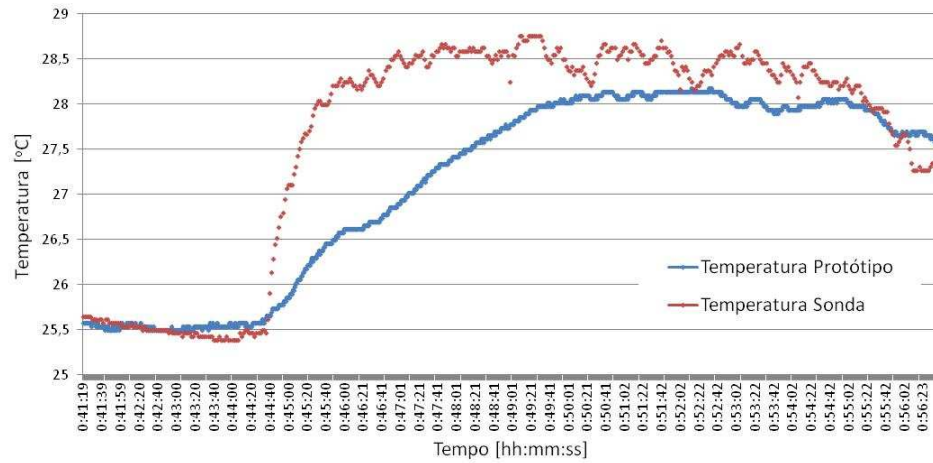


Figura 54 – Comparação dos valores de temperatura medidos pelo protótipo do nó sensor e pela sonda.

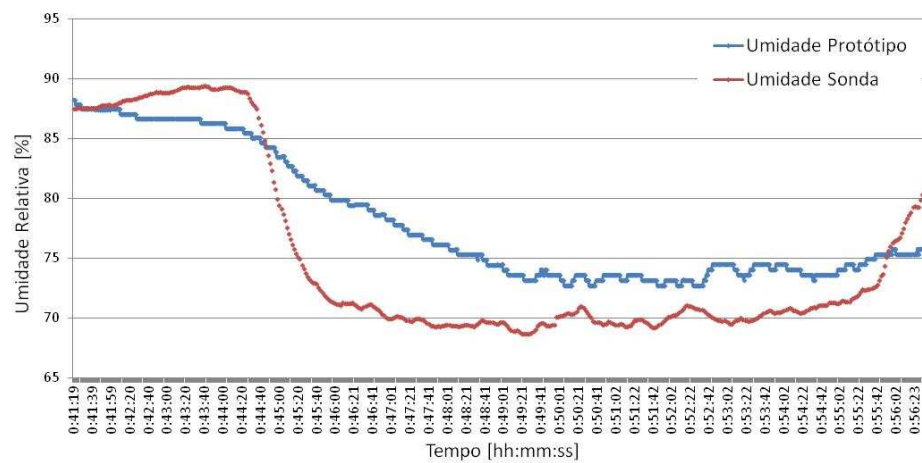


Figura 55 – Comparação dos valores de umidade medidos pelo protótipo do nó sensor e pela sonda.

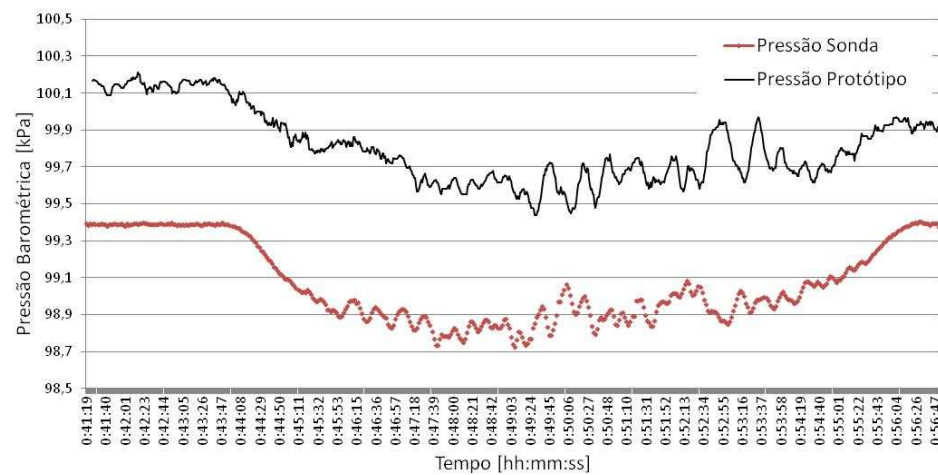


Figura 56 – Comparação dos valores de pressão atmosférica medidos pelo protótipo e pela sonda.

A análise dos dados levou às seguintes avaliações:

- ✓ Pressão Atmosférica: A leitura de pressão atmosférica, que deveria apresentar um decréscimo linear com a elevação do balão, mostrou que em pouco tempo a pressão tendeu a se estabilizar em torno de um valor. Isso foi resultado da ação do vento no corpo do balão – especialmente após o balão atingir a altura da copa das árvores – que o fez deslocar lateralmente. Esse comportamento impediu que a pressão medida pelo sensor fosse tomada como parâmetro de altura e pudesse ser utilizada para determinar a distância entre o coordenador e o sensor. O sensor, contudo, não deixou de enviar corretamente os dados das leituras e, através do cálculo da hipotenusa do triângulo formado pela altura estimada do balão com a distância horizontal provocada pelo vento, a distância foi estimada em 120m. Não foi possível avançar mais com o sensor pois, com os ventos fortes, havia a preocupação com a integridade dos instrumentos, uma vez que o cabo de nylon que sustentava o balão e ele próprio poderiam se enroscar em alguma árvore;
- ✓ Condução de calor através da placa: com os testes à noite, foi possível analisar as leituras de temperatura com o nó sensor montado na placa, sem a influência da radiação solar ou térmica. Uma vez que o sensor de temperatura e umidade é construído sobre a placa sem nenhum tipo de isolamento, o sensor é passível de ter suas leituras “contaminadas” pelo calor vindo dos outros componentes através da placa, sendo similar ao efeito produzido pelo auto-aquecimento. Neste teste, mesmo sob condições de alta taxa de leitura, não foi observada a elevação se comparados os valores de pressão obtidos através do sensor com o valor da sonda.

Apesar de, neste teste, serem comparados valores de medidas, isto apenas corrobora o resultado principal – a confirmação da comunicação da rede a 120 m. O objetivo original do teste (obtenção do perfil altimétrico) não pôde ser alcançado devido aos ventos fortes e inesperados, conforme descrito anteriormente. Porém, os resultados secundários do teste foram considerados satisfatórios, pois confirmaram o comportamento esperado. Observou-se também que as medidas de temperatura e umidade do 2º protótipo apresentam uma inércia maior quando

comparadas com as medidas dos sensores da sonda. A grande inércia térmica é, portanto, uma deficiência do 2º protótipo.

4.1.2 CALIBRAÇÃO

Os sensores utilizados não necessitam de calibração. Todos os sensores saem calibrados de fábrica e os valores de calibração para cada sensor estão armazenados em uma memória OTP (*One-Time Programmable*) interna e não são acessíveis por comandos externos. Porém, uma vez que são necessários alguns ajustes nos valores obtidos dos sensores em tempo de execução, foi necessário comparar o valor das medidas realizadas pelos sensores com o valor real das medidas, de forma que se garantisse que as medidas não estão sendo distorcidas pelo processamento interno do sistema. Os testes foram realizados em uma câmara climática da fabricante Teeney, modelo T10RC contido no Laboratório de Caracterização e Confiabilidade de Circuitos Integrados (LCC-CI) do CT-PIM. Neste teste foram utilizados quatro nós protótipos baseados na 3ª versão do projeto.

A rede de Sensores foi submetida a testes de desempenho em laboratório. Para isso, foi utilizado um sistema cedido pelo projeto LBA-INPA, utilizado pelo projeto para realizar as medidas ambientais. O sistema é composto de quatro sensores de temperatura e umidade da fabricante Vaisala, um sensor de Pressão Barométrica e um datalogger da fabricante Campbell. Estes componentes do sistema são os mesmos utilizados em campo nas torres instrumentadas. Estes sensores foram posicionados dentro de uma câmara climática T10RC da fabricante Tenney. A vantagem de se testar os sistemas em uma câmara climática é o controle das variáveis ambientais em níveis conhecidos (podendo-se extrapolar os valores obtidos no meio ambiente) e a ausência de problemas externos – como precipitações atmosféricas e incidência direta da luz solar. O teste foi realizado medindo o ambiente dentro de uma câmara climática configurada com os seguintes valores de temperatura e umidade, representadas no gráfico da figura 57.

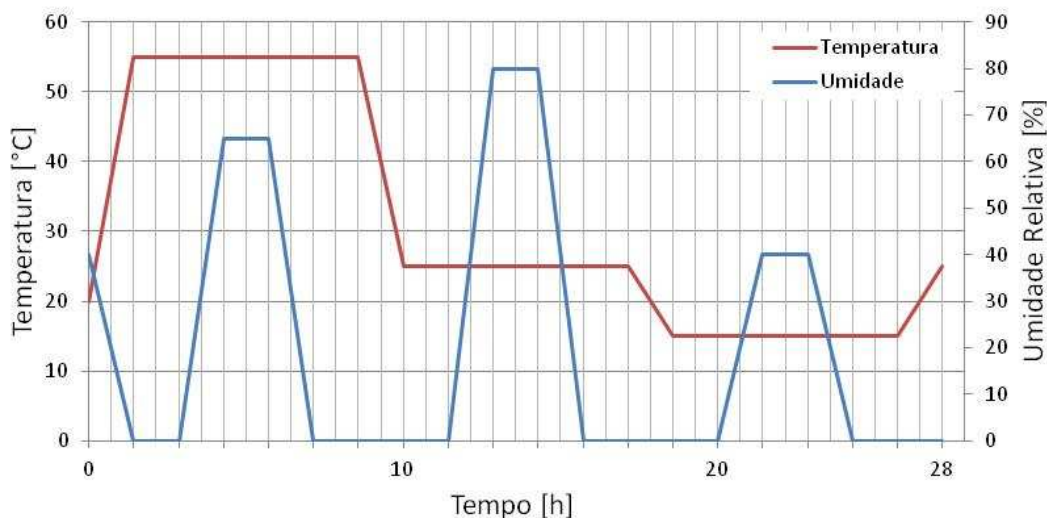


Figura 57: Curvas de Configuração de Temperatura e Umidade da Câmara Climática.

As curvas de temperatura e umidade foram configuradas de modo que se pudesse evitar a condensação do vapor dentro da câmara e danificasse, especialmente, os circuitos expostos do protótipo dos nós. Foram realizados três ciclos de 28 horas de testes com os quatro nós. Os sensores de referência e os nós da rede foram posicionados o mais próximo possível uns dos outros, de modo que cada nó fosse comparado com uma referência própria. A disposição dos sensores dentro da câmara climática é mostrada na figura 58, onde na figura 58a vê-se os sensores, da esquerda para a direita, na posição 4, 3, 1 e 2.



Figura 58 – Disposição dos sensores na câmara (a) e detalhe da proximidade dos sensores (b).

Seguem os gráficos para cada grandeza medida, para cada sensor:

4.1.2.1 Comparação de Valores Medidos de Temperatura

Seguem os valores de temperatura totais de todos os nós e sensores de referência:

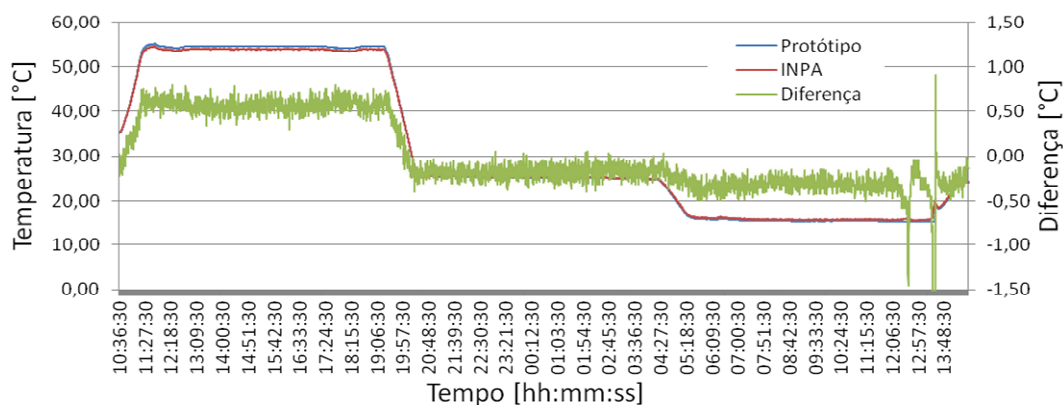


Figura 59 – Temperatura do Protótipo e Sensor de Referência na posição 1.

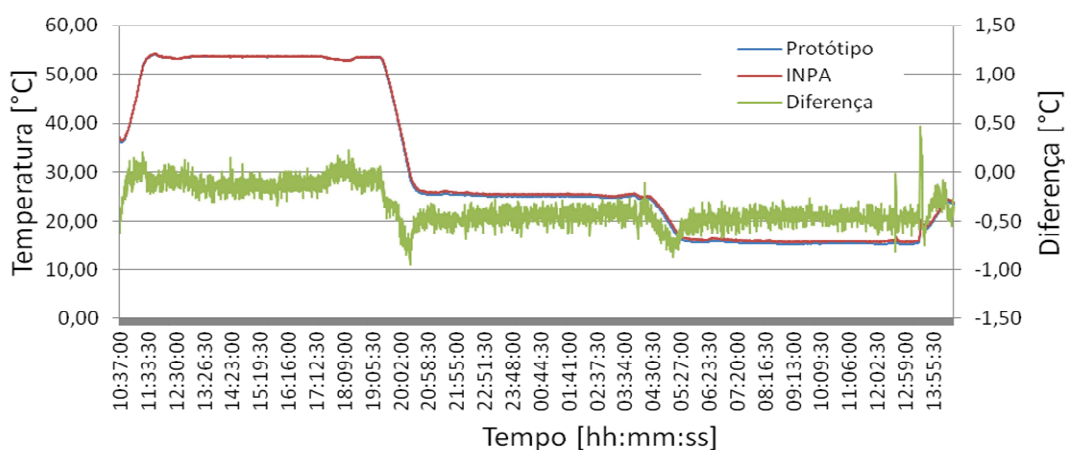


Figura 60 – Temperatura do Protótipo e Sensor de Referência na posição 2.

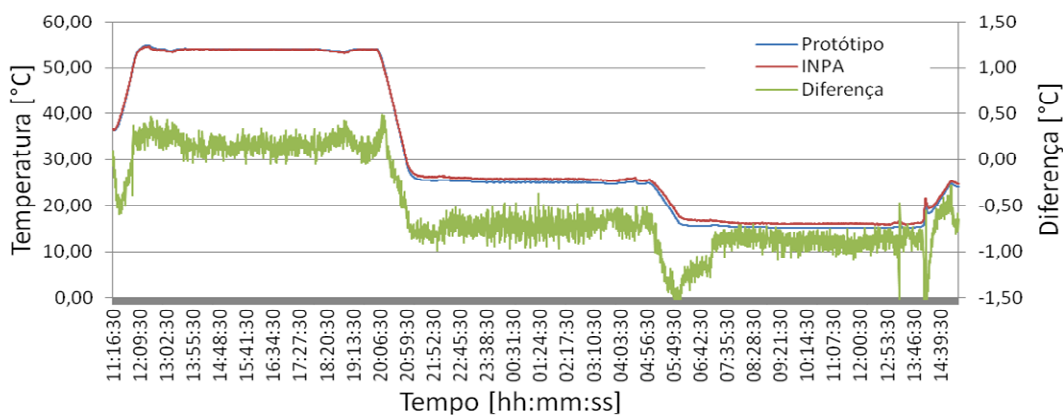


Figura 61 – Temperatura do Protótipo e Sensor de Referência na posição 3.

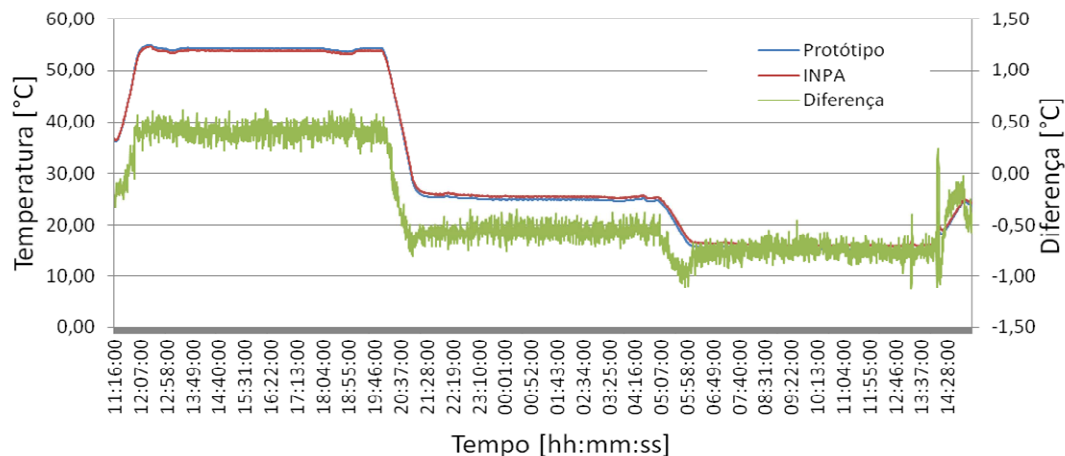


Figura 62 – Temperatura do Protótipo e Sensor de Referência na posição 4.

Os valores de temperatura foram selecionados, separando-os nos valores encontrados nos degraus (mais estáveis), sendo eles 55°C, 25°C e 15°C, mostrados a seguir:

Comparação a 55°C

Seguem os gráficos de temperatura dos sensores a 55°C, comparados com o sensor de referência do INPA:

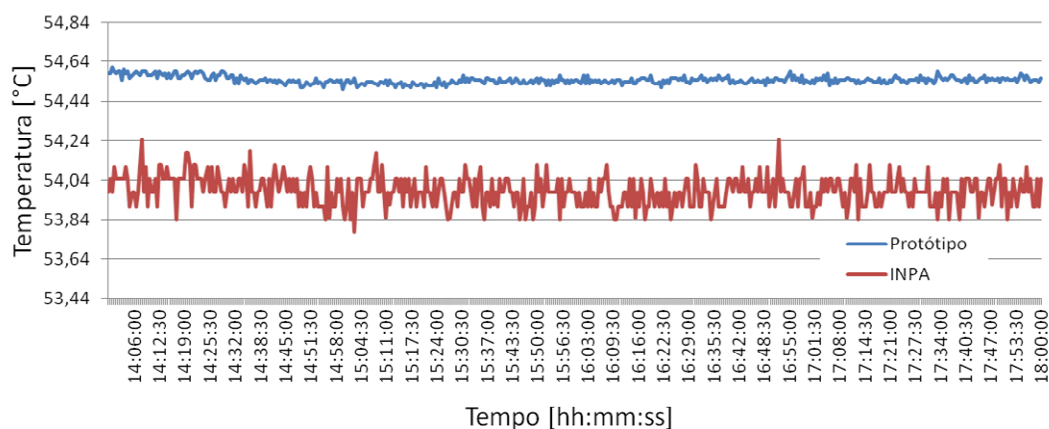


Figura 63 – Temperatura do Protótipo e Sensor de Referência na posição 1, a 55°C.

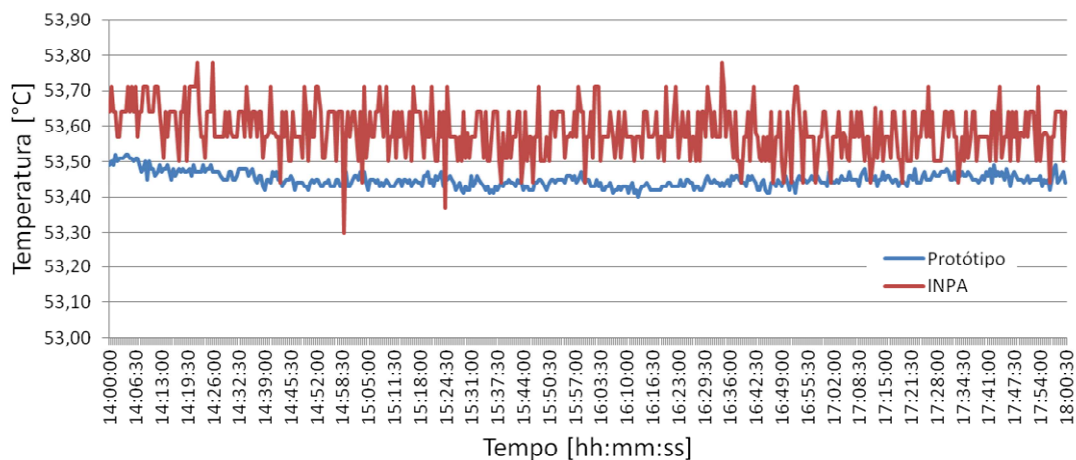


Figura 64 – Temperatura do Protótipo e Sensor de Referência na posição 2, a 55°C.

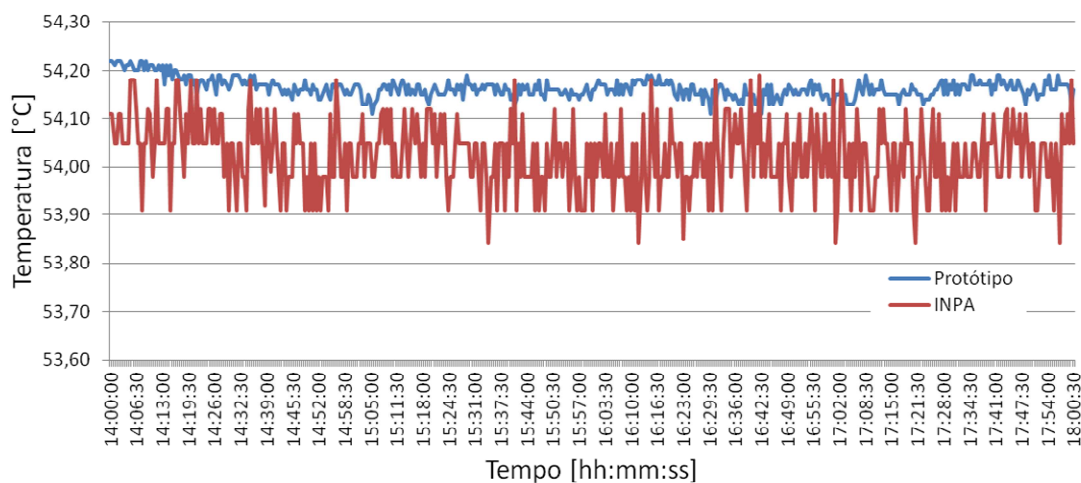


Figura 65 – Temperatura do Protótipo e Sensor de Referência na posição 3, a 55°C.

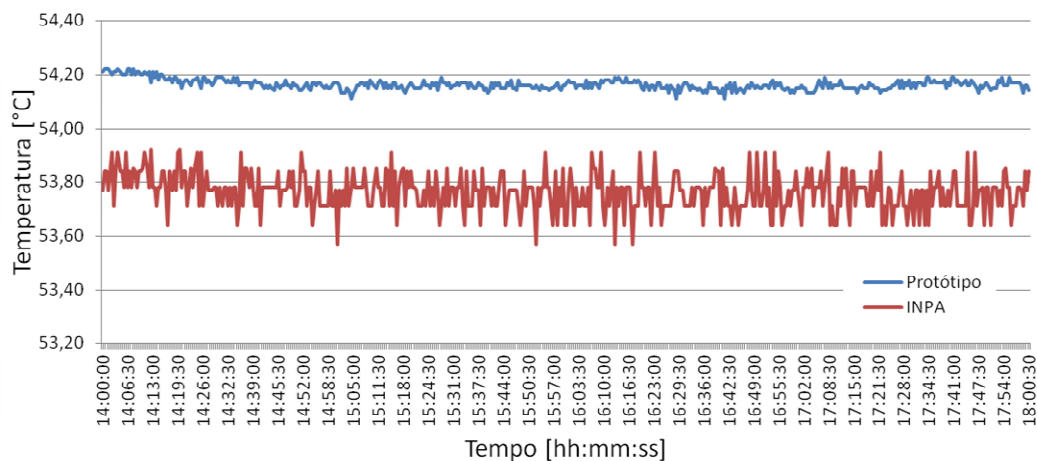


Figura 66 – Temperatura do Protótipo e Sensor de Referência na posição 4, a 55°C.

A tabela IX resume os valores obtidos utilizando sua média, desvio padrão e diferença entre os valores do protótipo e da referência INPA.

Tabela IX – Comparação da Temperatura Média dos Nós Sensores e Sensores de Referência a 55°C

	Protótipo (°C)	Desvio Padrão Protótipo (°C)	Referência (°C)	Desvio Padrão Referência(°C)	Diferença (°C)
Posição 1	54,55	0,018	53,99	0,075	0,56
Posição 2	53,45	0,020	53,59	0,070	-0,14
Posição 3	54,16	0,019	54,03	0,072	0,14
Posição 4	54,16	0,019	53,76	0,070	0,40

O mesmo procedimento de comparação foi realizado para as temperaturas de 25°C e 15°C. Os gráficos de comparação de cada nó sensor do protótipo com o sensor de referência do INPA estão no apêndice C. As tabelas X e XI resumem os valores medidos e indicam seu desvio padrão e diferença.

Tabela X – Comparação da Temperatura Média dos Protótipos e Sensores de Referência a 25°C

	Protótipo (°C)	Desvio Padrão Protótipo (°C)	Referência (°C)	Desvio Padrão Referência(°C)	Diferença (°C)
Posição 1	25,25	0,045	25,43	0,072	-0,18
Posição 2	25,03	0,061	25,48	0,082	-0,45
Posição 3	25,02	0,063	25,75	0,101	-0,73
Posição 4	25,02	0,062	25,58	0,076	-0,56

Tabela XI – Comparação da Temperatura Média dos Protótipos e Sensores de Referência a 15°C

	Protótipo (°C)	Desvio Padrão Protótipo (°C)	Referência (°C)	Desvio Padrão Referência(°C)	Diferença (°C)
Posição 1	15,37	0,024	15,68	0,072	-0,31
Posição 2	15,37	0,028	15,83	0,068	-0,45
Posição 3	15,19	0,068	16,08	0,083	-0,89
Posição 4	15,14	0,031	15,89	0,073	-0,74

4.1.2.2 Comparação de Valores Medidos de Umidade Relativa

Seguem os valores de umidade relativa totais de todos os nós e sensores de referência:

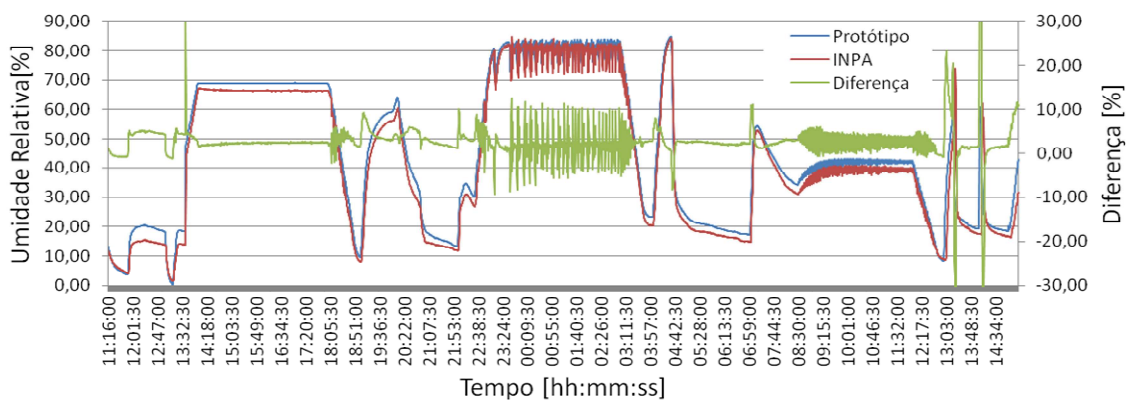


Figura 67 – Umidade Relativa do Protótipo e Sensor de Referência na posição 1.

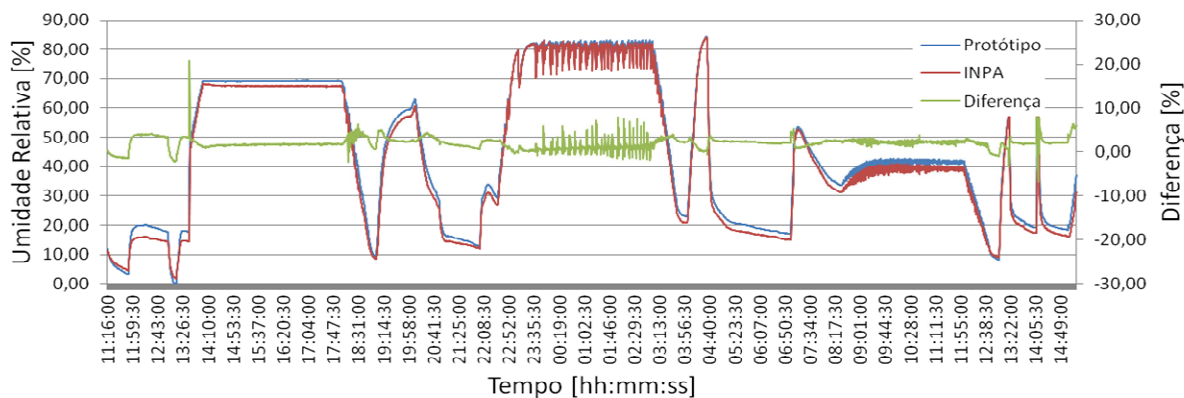


Figura 68 – Umidade Relativa do Protótipo e Sensor de Referência na posição 2.

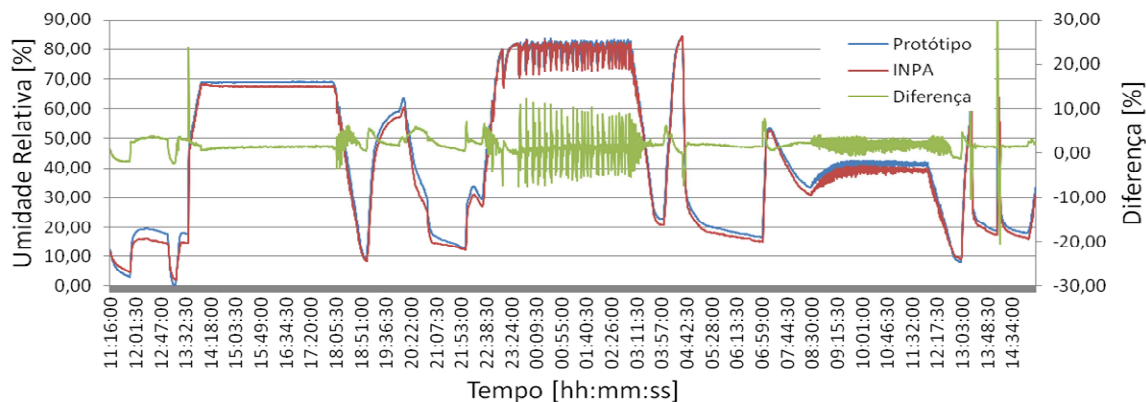


Figura 69 – Umidade Relativa do Protótipo e Sensor de Referência na posição 3.

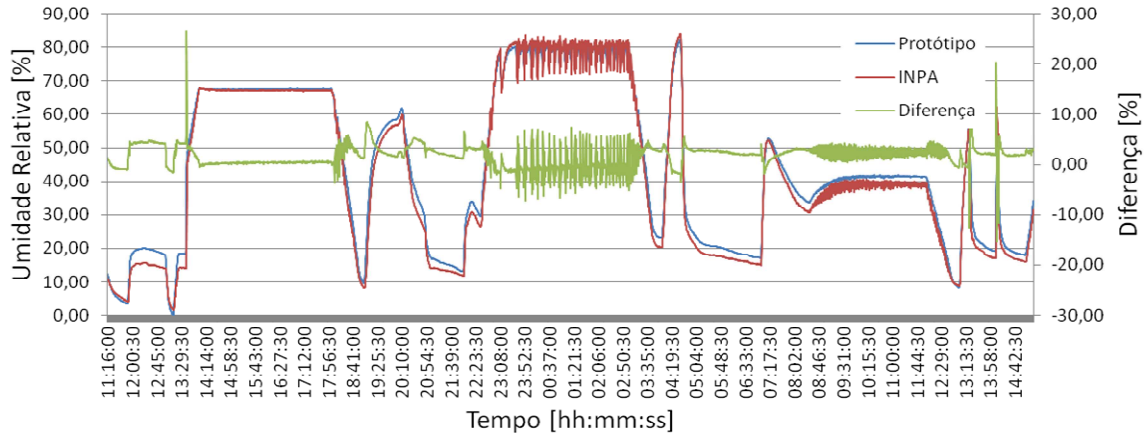


Figura 70 – Umidade Relativa do Protótipo e Sensor de Referência na posição 4.

Os valores de umidade relativa foram selecionados, separando-os nos valores encontrados nos degraus (mais estáveis) de temperatura e umidade, sendo estes 65%, 80% e 40%, mostrados a seguir:

Comparação a 65%

Seguem os gráficos de umidade relativa dos sensores a 65% medidos a temperatura de 55°C, comparados com o sensor de referência do INPA:

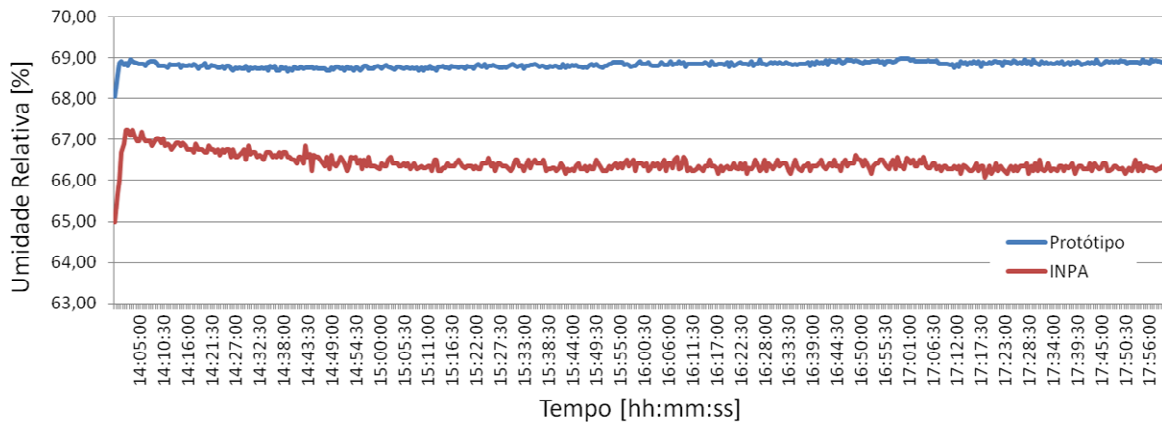


Figura 71 – Umidade Relativa do Protótipo e Sensor de Referência na posição 1, a 65%.

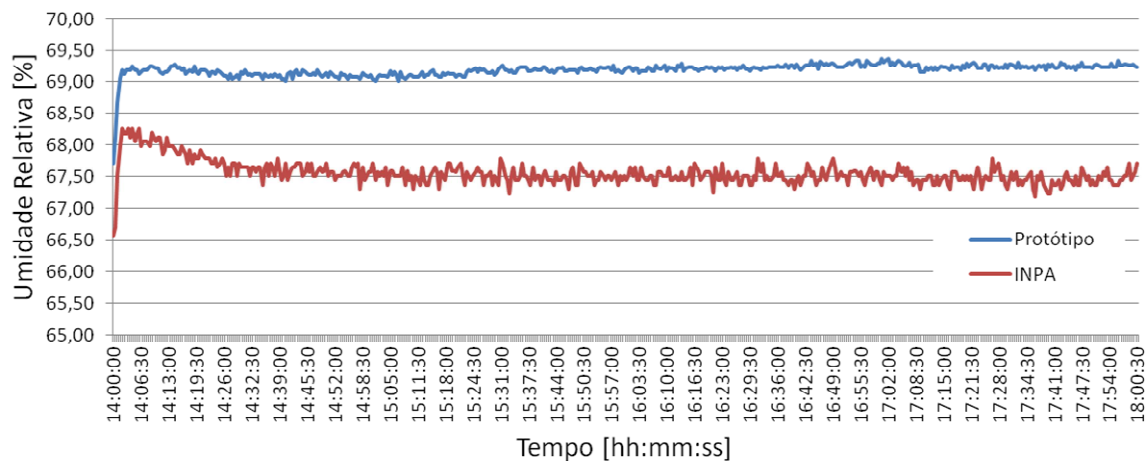


Figura 72 – Umidade Relativa do Protótipo e Sensor de Referência na posição 2, a 65%.

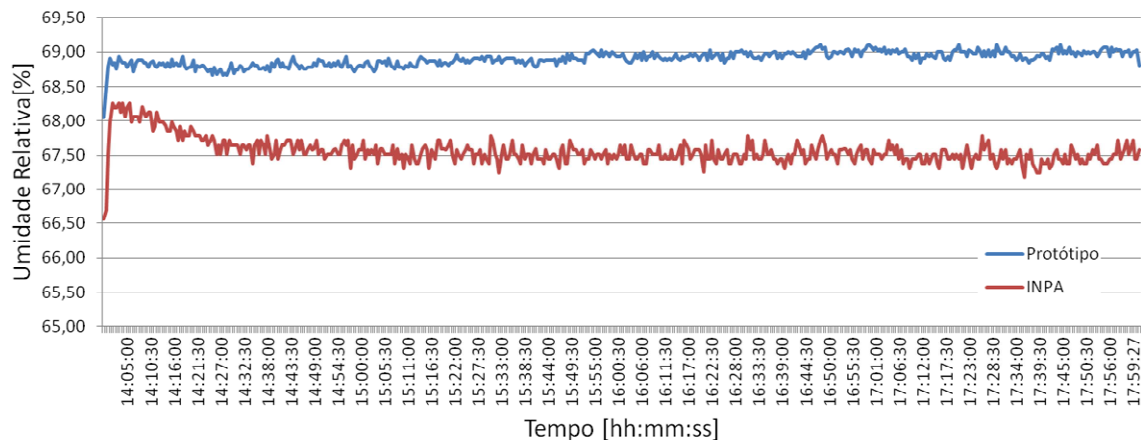


Figura 73 – Umidade Relativa do Protótipo e Sensor de Referência na posição 3, a 65%.

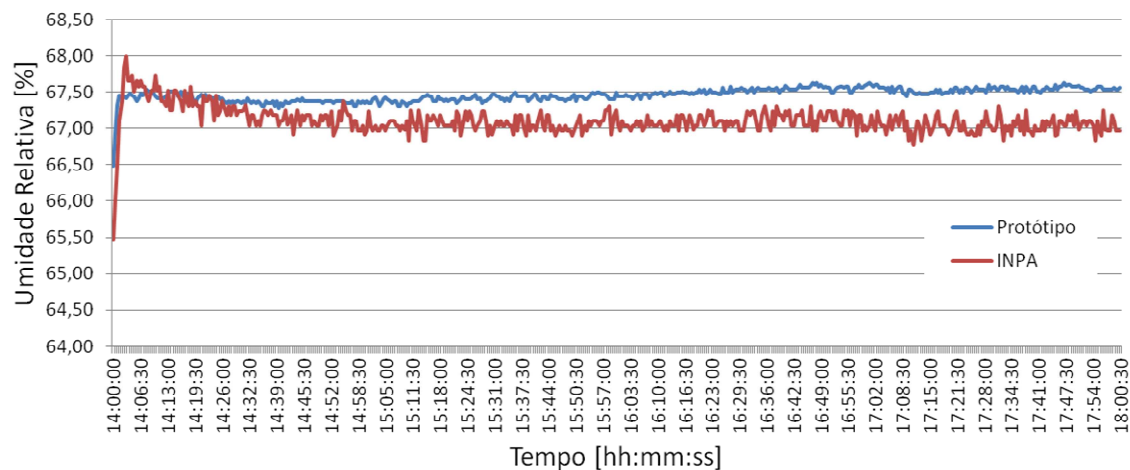


Figura 74 – Umidade Relativa do Protótipo e Sensor de Referência na posição 4, a 65%.

Calculando-se a média dos valores encontrados durante a sessão com 65%, a seguinte tabela foi montada para comparação dos valores obtidos:

Tabela XII – Comparação da Umidade Relativa Média dos Protótipos e Sensores de Referência a 65%

	Protótipo (%RH)	Desvio Padrão Protótipo (%RH)	Referência (%RH)	Desvio Padrão Referência (%RH)	Diferença (%RH)
Posição 1	68,82	0,078	66,43	0,213	2,40
Posição 2	69,19	0,111	67,56	0,188	1,63
Posição 3	68,90	0,101	67,56	0,187	1,34
Posição 4	67,46	0,094	67,11	0,188	0,35

Os gráficos de comparação das demais medidas de umidade de cada nó sensor do protótipo com o sensor de referência do INPA estão no apêndice C. As tabelas XIII e XIV resumem os valores medidos e indicam seu desvio padrão e diferença. As medidas de umidade em torno de 80% foram realizadas com a temperatura de 25°C, e as medidas de umidade em torno de 40% foram realizadas com a temperatura de 15°C. Devido às tentativas da câmara de estabilizar o valor de umidade nos valores configurados, as medidas variaram consideravelmente em torno do nível médio, o que afetou os valores de desvio padrão.

Tabela XIII – Comparação da Umidade Relativa Média dos Protótipos e Sensores de Referência a 80%

	Protótipo (%RH)	Desvio Padrão Protótipo (%RH)	Referência (%RH)	Desvio Padrão Referência (%RH)	Diferença (%RH)
Posição 1	80,77	2,787	79,03	3,432	1,74
Posição 2	80,12	2,882	67,56	3,174	0,95
Posição 3	80,07	2,916	79,17	3,174	0,90
Posição 4	78,28	2,156	78,68	3,444	-0,40

Tabela XIV – Comparação da Umidade Relativa Média dos Protótipos e Sensores de Referência a 40%

	Protótipo (%RH)	Desvio Padrão Protótipo (%RH)	Referência (%RH)	Desvio Padrão Referência (%RH)	Diferença (%RH)
Sensor 1	40,54	2,426	37,64	2,662	2,89
Sensor 2	40,02	2,493	37,65	2,650	2,37
Sensor 3	39,84	2,461	37,65	2,650	2,19
Sensor 4	39,92	2,358	37,34	2,673	2,58

4.1.2.3 Comparação de Valores Medidos de Pressão Atmosférica

Uma vez que não se pôde controlar a pressão dentro da câmara, a pressão atmosférica serviu como parâmetro de comparação. O sensor de referência foi posicionado no fundo da câmara, e assumiu-se que a variação da pressão interna entre os nós sensores e o sensor de referência é irrelevante. Seguem os valores de pressão atmosférica totais de todos os nós e sensores de referência:

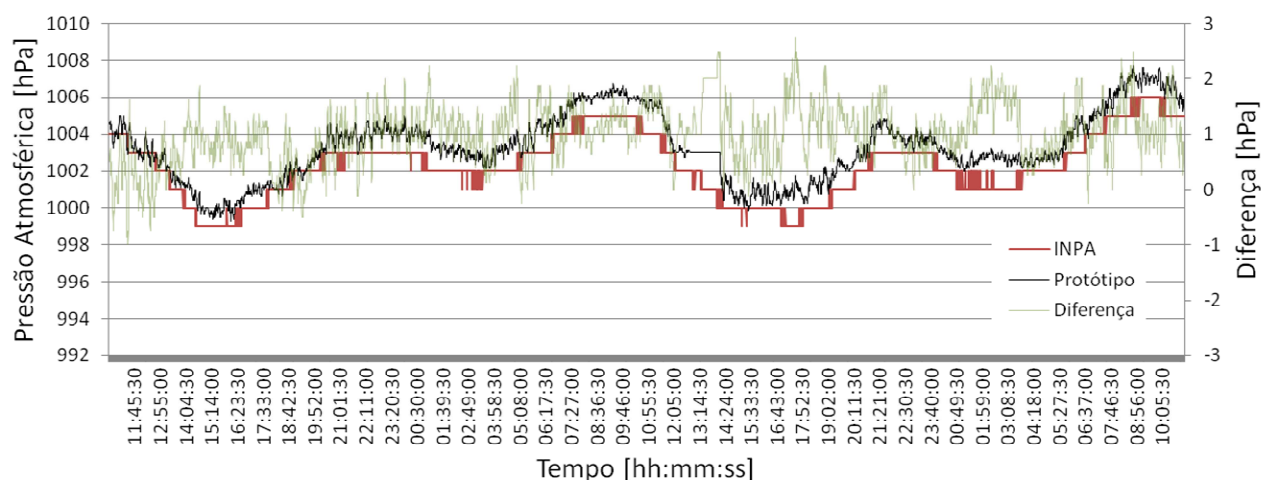


Figura 75 – Pressão Atmosférica do Protótipo e Sensor de Referência na posição 1, em hPa.

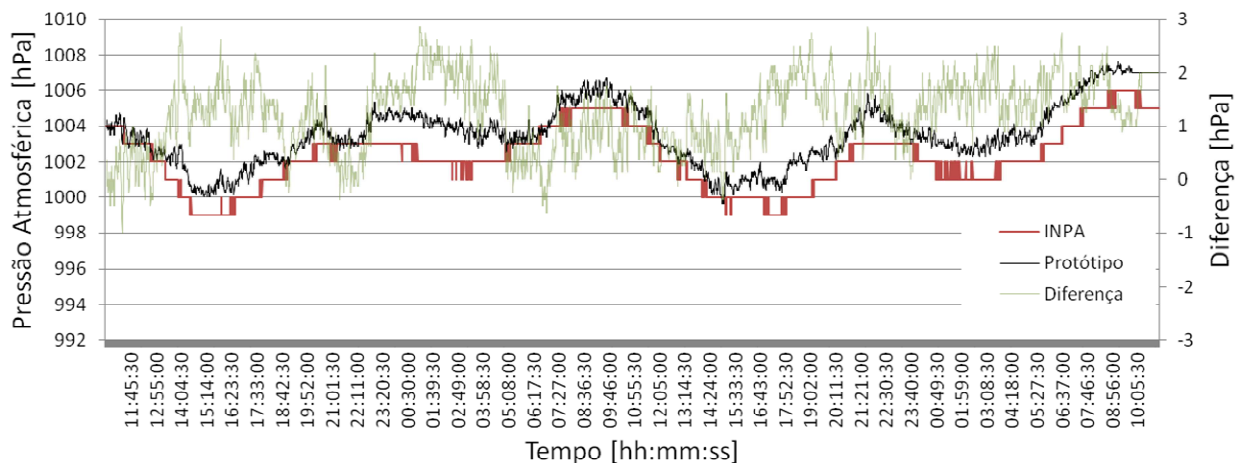


Figura 76 – Pressão Atmosférica do Protótipo e Sensor de Referência na posição 2, em hPa.

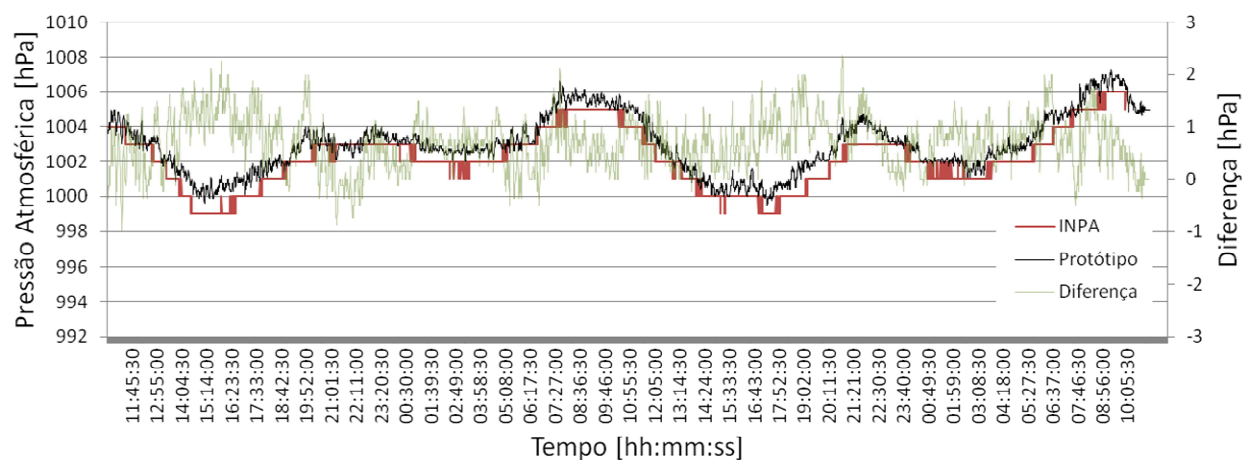


Figura 77 – Pressão Atmosférica do Protótipo e Sensor de Referência na posição 3, em hPa.

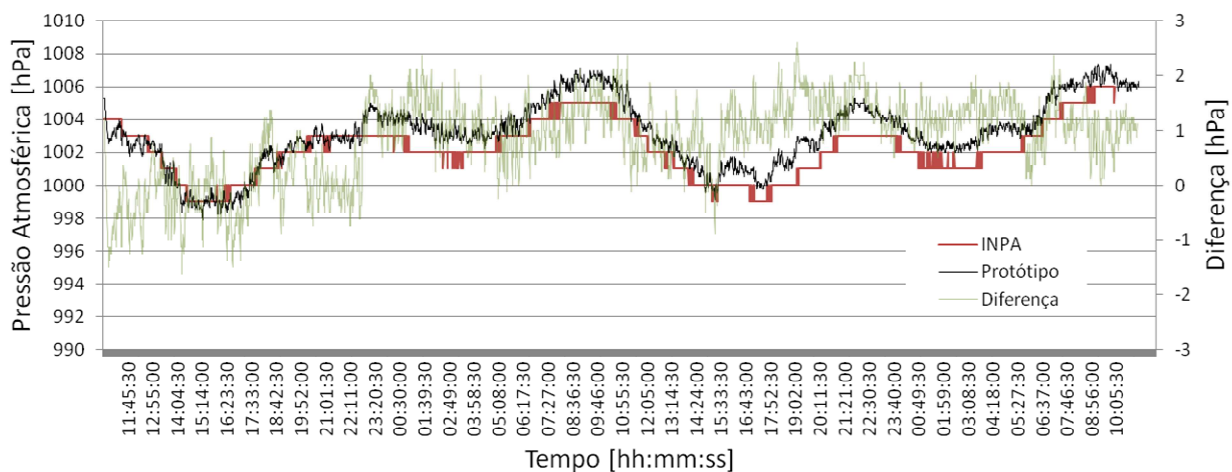


Figura 78 – Pressão Atmosférica do Protótipo e Sensor de Referência na posição 4, em hPa.

Foram calculadas as médias dos valores medidos durante o teste. Os valores médios encontrados são listados na Tabela XV, em hPa. Os valores medidos pelos protótipos foram convertidas de kPa para hPa para poderem ser comparadas com a referência, que registrou a pressão atmosférica em hPa por padrão.

Tabela XV – Comparação da Pressão Atmosférica Média dos Protótipos e Sensores de Referência em hPa

	Protótipo (hPa)	Desvio Padrão Protótipo (hPa)	Referência (hPa)	Desvio Padrão Referência (hPa)	Diferença (hPa)
Posição 1	1003,29	1,946	1002,28	1,733	1,01
Posição 2	1003,52	1,873			1,24
Posição 3	1002,98	1,789			0,70
Posição 4	1003,13	2,146			0,85

4.1.3 AUTONOMIA

Para testar a autonomia da bateria, uma estrutura de testes foi montada para se monitorar a carga de uma bateria de 135mAh, medindo-se a corrente e a tensão da bateria desde a primeira medida até a queda total da carga da bateria. Para isso, foi utilizado um multímetro Keithley com interface serial e um sistema microcontrolado composto de um cartão de memória para comunicação com o multímetro, de forma que fosse possível medir os valores de tensão e corrente da bateria com o nó operando normalmente. A bateria de 135mAh foi escolhida por ser aquela com menor autonomia disponível, permitindo um teste completo de carga e descarga no menor intervalo possível de tempo. O módulo Digi foi utilizado no modo de baixíssima energia (*hibernate*) e com nível de potência de transmissão mínimo. Após 26 dias de operação do nó, os valores de tensão e corrente da bateria variam conforme o gráfico na figura 79 a seguir.

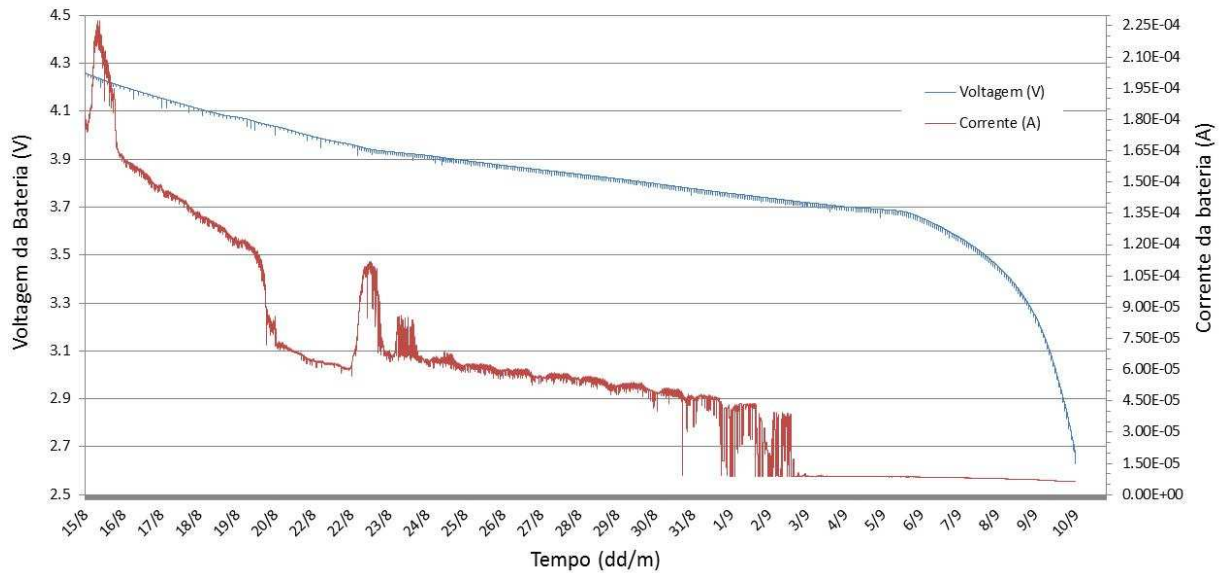


Figura 79 – Resultados dos Testes de Autonomia.

De acordo com este teste, a autonomia do nó sensor é de aproximadamente 428 horas para cada 100mAh de carga nominal de bateria. Como exemplo, para uma autonomia de 90 dias, é necessária uma bateria com carga nominal de pelo menos 504mAh.

Pela análise da curva de corrente, pôde-se observar que a corrente do nó sensor assume valores máximos de 225 μ A, estabilizando em valores abaixo de 9 μ A somente após a tensão da bateria chegar a 3,72V. Esse comportamento da corrente é atribuído ao consumo do módulo de comunicação de 2.4GHz da Digi operando no modo de baixíssima energia. O módulo trabalha muito bem nos valores de tensão próximos à nominal (3,7V), mas possui um comportamento irregular em valores superiores de tensão. No modo de baixa energia (*doze*), o consumo de energia possui um comportamento mais regular, porém com consumo maior (mínimo de 40 μ A). Um teste realizado neste modo de operação resultou num período de atividade do nó de apenas 10 dias.

4.2 Testes do Nó Coordenador

Assim como os nós sensores, o nó coordenador da Rede de Sensores Sem Fio foi desenvolvido por partes, adquirindo novas funções e melhorando principalmente o

aproveitamento da recepção dos dados transmitidos pelos nós ao longo do desenvolvimento da rede.

Para abordar o teste destes aprimoramentos, as características do nó coordenador foram divididas em duas partes: recepção/gravação das medidas e conexão à rede / acesso simultâneo ao sistema

4.2.1 RECEPÇÃO E GRAVAÇÃO DAS MEDIDAS

Na primeira fase do projeto, apenas a parte de recepção e disponibilidade dos dados via rede foi abordada. A recepção dos dados era feita a partir da leitura do *buffer* em intervalos pré-definidos de tempo. A figura 80 mostra a tela gerada pelo nó coordenador, mostrando as medidas realizadas por dois nós sensores.



Sensor	Temperatura	Umidade
1	29.36 °C	62.24 %
2	32.03 °C	81.88 %
3	???	???
4	???	???

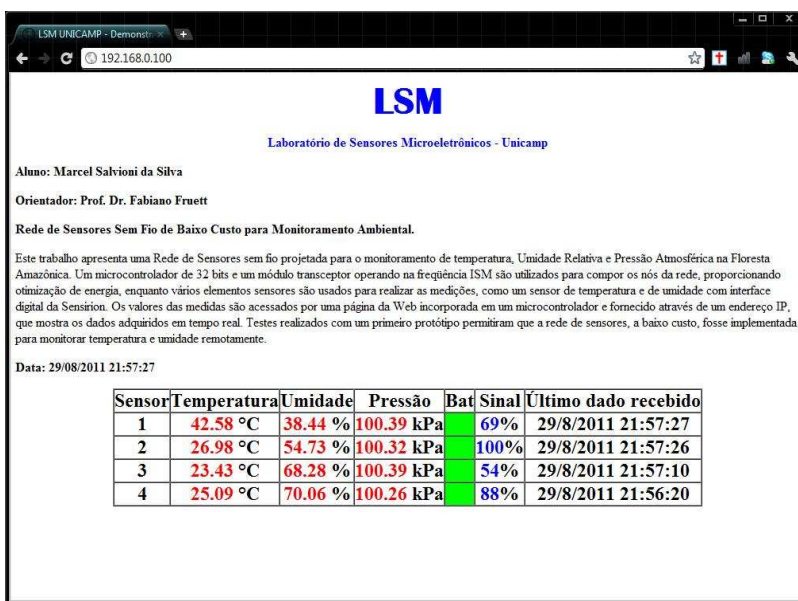
Figura 80 – Tela do navegador obtida durante os primeiros testes com o nó coordenador.

Nesta implementação, eliminou-se o problema da perda de pacotes. A solução encontrada foi fundamental para os desenvolvimentos seguintes. Posteriormente, foi desenvolvida a gravação das medidas em um cartão de memória e a utilização do relógio interno para marcar o momento em que cada medida é recebida de cada nó sensor. Os testes confirmaram a geração e a correta gravação dos arquivos no cartão de memória e as marcações de data e hora para cada uma das medidas para cada sensor.

O valor correspondente ao nível de sinal recebido quando das leituras dos sensores – última mudança de *firmware* - foi testada ainda em laboratório. O teste envolveu os quatro nós

em pontos diferentes do laboratório, nos quais se pôde verificar um valor diferente do valor máximo de recepção. Os sinais gerados foram verificados com um osciloscópio, observando a largura de pulso relativa ao nível de sinal que alimenta o microcontrolador. Após a análise do sinal e do valor convertido pelo *firmware*, foi observada uma diferença de aproximadamente 1% do valor convertido para o valor medido no osciloscópio, que se manteve constante entre os níveis 35% a 99% da potência do sinal. Essa diferença é atribuída a um truncamento do valor obtido pela rotina de tratamento do sinal de entrada e é considerado aceitável para um valor estimado de potência de sinal.

Por fim, a página *web* onde as informações e medidas de cada nó sensor são exibidas é mostrada na figura 81. A seção 3.2.2 desta dissertação discorre acerca dos detalhes de implementação.



LSM UNICAMP - Demonstration

192.168.0.100

LSM

Laboratório de Sensores Microeletrônicos - Unicamp

Aluno: Marcel Salvioni da Silva

Orientador: Prof. Dr. Fabiano Fruett

Rede de Sensores Sem Fio de Baixo Custo para Monitoramento Ambiental.

Este trabalho apresenta uma Rede de Sensores sem fio projetada para o monitoramento de temperatura, Umidade Relativa e Pressão Atmosférica na Floresta Amazônica. Um microcontrolador de 32 bits e um módulo transceptor operando na frequência ISM são utilizados para compor os nós da rede, proporcionando otimização de energia, enquanto vários elementos sensores são usados para realizar as medições, como um sensor de temperatura e de umidade com interface digital da Sensirion. Os valores das medidas são acessados por uma página da Web incorporada em um microcontrolador e fornecido através de um endereço IP, que mostra os dados adquiridos em tempo real. Testes realizados com um primeiro protótipo permitiram que a rede de sensores, a baixo custo, fosse implementada para monitorar temperatura e umidade remotamente.

Data: 29/08/2011 21:57:27

Sensor	Temperatura	Umidade	Pressão	Bat	Sinal	Último dado recebido
1	42.58 °C	38.44 %	100.39 kPa		69%	29/8/2011 21:57:27
2	26.98 °C	54.73 %	100.32 kPa		100%	29/8/2011 21:57:26
3	23.43 °C	68.28 %	100.39 kPa		54%	29/8/2011 21:57:10
4	25.09 °C	70.06 %	100.26 kPa		88%	29/8/2011 21:56:20

Figura 81 – Tela do navegador com as implementações finais da página *web*.

4.2.2 CONEXÃO EM REDE VIA ETHERNET E ACESSO SIMULTÂNEO

O principal objetivo dos testes de conexão da rede via Ethernet foi verificar os seguintes pontos:

- ✓ A transmissão dos dados da página HTML através da interface Ethernet sem travamentos ou falhas;

- ✓ As leituras dos sensores apresentadas na página de acordo com o medido pelos nós sensores;
- ✓ O impacto da transmissão dos dados via interface Ethernet no funcionamento geral do coordenador (recepção das medidas, gravação no cartão, medida do nível de sinal).

A placa do coordenador não responde a comandos *ping* através da rede, uma vez que o seu *firmware* foi desenvolvido de modo simplificado. Por isso, o teste de conexão do coordenador à rede foi realizado acessando o IP estático através de um navegador web (*browser*).

Nas primeiras versões do *firmware* – com a obtenção dos dados dos sensores em intervalos regulares – foi observada a perda momentânea de medidas de alguns dos sensores. Isso ocorria se sua recepção fosse feita logo após o envio da página *web* para o usuário, mascarando a atualização dos valores medidos. Outro problema encontrado foi o estouro de memória causado pelo mesmo problema que impedia que os dados da página fossem transmitidos. Com a mudança na forma de receber as medidas dos sensores, a página *web* foi atualizada sempre com o valor mais recente de leitura dos sensores, sem interrupções.

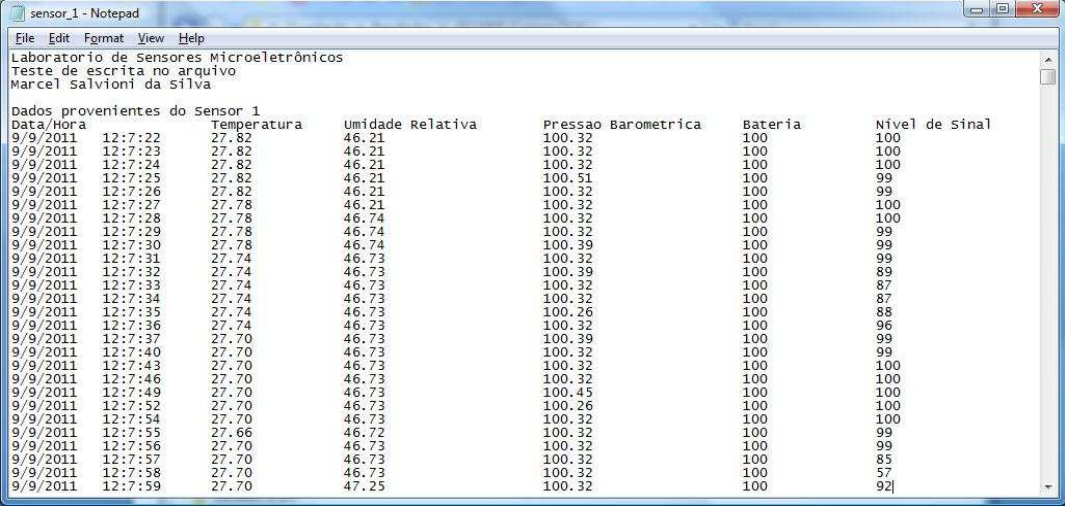
Com a implementação da gravação dos dados e dos valores de data e hora da aquisição dos sensores, o microcontrolador passou a processar uma quantidade consideravelmente maior de informação se comparado com as versões anteriores. Ao que foi observado, esta adição não influenciou na transmissão e montagem da página no navegador do usuário. Porém, ao serem adicionados os valores de bateria e de nível de sinal, alguns valores apresentaram um erro de formatação na página. Como todos os valores são tratados através do mesmo mecanismo (inserção dinâmica), foi necessário o aumento do *buffer* dedicado para a transmissão dos dados da página via Ethernet. Apesar do valor acima do estipulado inicialmente, não foram encontrados novos problemas após a mudança.

Outro teste realizado foi o acesso de múltiplos usuários à página. O teste consistiu em configurar cinco *browsers* em computadores diferentes para simular um acesso simultâneo ao sistema e verificar sua robustez. Um dos acessos foi realizado através de um *smartphone* funcionando com sistema Android para verificar a compatibilidade dos dados vindos do

coordenador em HTML com o navegador deste sistema operacional embarcado. Os testes com múltiplos usuários foram bem sucedidos, sendo observada apenas uma diferença de tempo adicional em torno de um segundo enquanto a página era atualizada. O teste com o *smartphone* também gerou bons resultados, o que era esperado pela estrutura simples e genérica adotada para se construir a página.

4.2.3 GRAVAÇÃO DOS DADOS NO CARTÃO DE MEMÓRIA

Os valores das medidas dos sensores são gravados no cartão de memória SD/MMC assim que são recebidos. Um cabeçalho seguido de uma formatação foram adotados para organizar as medidas dentro de um arquivo de texto simples, permitindo utilizá-las posteriormente em uma planilha genérica para gerar os gráficos e proceder com as análises necessárias. Cada arquivo recebe o nome do sensor correspondente. A figura 82 mostra um exemplo de arquivo utilizado para o armazenamento das medidas realizadas pelo sensor 1.



sensor_1 - Notepad

File Edit Format View Help

Laboratório de Sensores Microeletrônicos
 Teste de escrita no arquivo
 Marcel salvioni da Silva

Dados provenientes do Sensor 1

Data/Hora	Temperatura	Umidade Relativa	Pressao Barometrica	Bateria	Nível de Sinal
9/9/2011 12:7:22	27.82	46.21	100.32	100	100
9/9/2011 12:7:23	27.82	46.21	100.32	100	100
9/9/2011 12:7:24	27.82	46.21	100.32	100	100
9/9/2011 12:7:25	27.82	46.21	100.32	100	99
9/9/2011 12:7:26	27.82	46.21	100.32	100	99
9/9/2011 12:7:27	27.78	46.21	100.32	100	100
9/9/2011 12:7:28	27.78	46.74	100.32	100	100
9/9/2011 12:7:29	27.78	46.74	100.32	100	99
9/9/2011 12:7:30	27.78	46.74	100.39	100	99
9/9/2011 12:7:31	27.74	46.73	100.32	100	99
9/9/2011 12:7:32	27.74	46.73	100.39	100	89
9/9/2011 12:7:33	27.74	46.73	100.32	100	87
9/9/2011 12:7:34	27.74	46.73	100.32	100	87
9/9/2011 12:7:35	27.74	46.73	100.26	100	88
9/9/2011 12:7:36	27.74	46.73	100.32	100	96
9/9/2011 12:7:37	27.70	46.73	100.39	100	99
9/9/2011 12:7:40	27.70	46.73	100.32	100	99
9/9/2011 12:7:43	27.70	46.73	100.32	100	100
9/9/2011 12:7:46	27.70	46.73	100.32	100	100
9/9/2011 12:7:49	27.70	46.73	100.45	100	100
9/9/2011 12:7:52	27.70	46.73	100.26	100	100
9/9/2011 12:7:54	27.70	46.73	100.32	100	100
9/9/2011 12:7:55	27.66	46.72	100.32	100	99
9/9/2011 12:7:56	27.70	46.73	100.32	100	99
9/9/2011 12:7:57	27.70	46.73	100.32	100	85
9/9/2011 12:7:58	27.70	46.73	100.32	100	57
9/9/2011 12:7:59	27.70	47.25	100.32	100	92

Figura 82 – Exemplo de arquivo texto que armazena os dados de cada sensor.

4.3 Conclusões

Desde o início do desenvolvimento da Rede de Sensores Sem Fio, testes prévios foram úteis para se observar comportamentos esperados – e às vezes inesperados, corrigir erros pontuais e implementar novas funções de acordo com o proposto no início do projeto. Pôde-se comprovar com as medidas e ensaios realizados que o sistema construído atendeu aos requisitos

de alcance de comunicação e de função, necessitando de uma revisão de projeto de circuito para atender o requisito de autonomia, uma vez que o *firmware* desenvolvido e os testes com as modificações ainda na versão inicial contemplaram esta característica.

Capítulo 5

Conclusões

Este trabalho apresenta o projeto de um sistema embarcado usando rede de sensores sem fio para aplicações que envolvam monitoramento ambiental, com uma abordagem de baixo custo para medidas de temperatura ambiente, umidade relativa do ar e pressão barométrica. Neste projeto, foram pesquisadas as principais tecnologias de comunicação sem fio, sensores comerciais para medidas das variáveis desejadas e microcontroladores capazes de assumir as tarefas de aquisição das medidas e transmissão com a melhor relação custo-benefício. As medidas realizadas pelos nós sensores podem ser acessadas através de uma página *web* embarcada no nó coordenador, sendo armazenadas continuamente em um cartão de memória. Os testes foram realizados em laboratório e em campo, e as conclusões acerca das características levantadas são apresentadas a seguir.

5.1 Conclusões sobre comunicação e alcance

Os testes realizados para se verificar o alcance de comunicação dos nós sensores confirmaram a capacidade da rede de sensores, usando o módulo ZigBee operando a 2,4GHz, de

ser aplicada nos cenários de monitoramento ambiental apresentados neste trabalho. Tanto os testes realizados nas torres instrumentadas quanto no balão atmosférico foram concluídos com sucesso.

Um fator observado durante os testes foi a atenuação dos sinais transmitidos pelos nós sensores durante testes de comunicação com árvores e floresta entre os nós sensores e o nó coordenador. Dependendo da densidade de vegetação entre os módulos de comunicação, a distância máxima pode ser drasticamente reduzida. Para a solução deste problema, podem ser feitas as seguintes mudanças no sistema:

- ✓ Alteração da potência emitida pelo nó sensor: atualmente a potência emitida é a menor possível, favorecendo menor consumo. Uma potência mais alta habilita a transmissão em uma distância maior, porém aumenta o consumo de potência e reduz a autonomia dos nós;
- ✓ Reposicionamento do coordenador da rede: Dependendo da possibilidade, pode ser conveniente reposicionar o nó coordenador para favorecer a transmissão de um dos nós, mantendo os demais ao alcance;
- ✓ Adoção de módulos ZigBee operando a 900MHz: A Digi possui outros módulos de comunicação que operam na frequência ISM a 900MHz. Eles também permitem a comunicação em rede, mas trabalham com um consumo mais alto de energia, o que compromete a autonomia geral da rede. Uma vez adotado um módulo novo, todos os outros devem ser substituídos.

Os testes com o balão atmosférico mostraram que, sem obstáculos (aproximadamente 200m), o sistema é bastante eficiente e opera a longo alcance, permitindo o levantamento do perfil altimétrico de temperatura, umidade e pressão atmosférica com um ou vários sensores acoplados ao balão, sem comprometer a altura alcançada por não serem necessários cabos para a transmissão ou alimentação.

5.2 Conclusões sobre a autonomia

A partir dos testes realizados em laboratório e dos valores de tensão e corrente obtidos, pode-se concluir que o nó sensor possui autonomia suficiente para ser utilizado, com uma frequência de intervenções para carga de bateria compatível com a frequência média de intervenções de outros equipamentos. Deste modo, a agenda de manutenção do sistema proposto pode ser integrada à de outros sistemas que estejam sendo utilizados no mesmo ambiente sendo monitorado. A partir de cálculos obtidos a partir dos testes realizados, foi estimado que, para uma autonomia de 90 dias, seria necessária uma bateria com capacidade de 504mAh. Uma vez que a capacidade máxima de bateria, determinada inicialmente para a mesma autonomia, foi de 700mAh, pode-se dizer que o desempenho do nó sensor com relação a consumo de energia ficou acima da expectativa, isto é, a bateria definida para o projeto supera a autonomia desejada.

A autonomia poderia ser ainda maior do que a obtida, caso o módulo de comunicação da Digi obedecesse a um consumo de energia menor e mais regular no modo de baixíssima energia – cujo comportamento não pode ser configurado. Ainda assim, o seu funcionamento não comprometeu a autonomia total do nó sensor.

Não foram realizados testes de autonomia no nó coordenador da rede, pois se espera que o mesmo seja conectado a uma infraestrutura mínima de energia, como a rede elétrica, geradores ou banco de baterias.

5.3 Conclusões sobre a calibração

Tanto os testes realizados em campo quanto os testes na câmara climática mostraram que a resolução e a faixa de valores das medidas dos sensores satisfazem os requisitos das aplicações voltadas para medidas ambientais. Durante os testes na câmara climática, foram observadas algumas diferenças nos valores medidos entre o sensor de referência e o sensor de temperatura do nó. As maiores diferenças foram observadas em temperaturas próximas a 85 °C que se encontram na faixa de tolerância documentada pelo fabricante. Uma vez que as temperaturas

máximas em campo não atingem mais que 50°C, pode-se concluir que para as faixas de interesse de temperatura, o sensor opera dentro dos valores esperados.

Mesmo obedecendo a todas as orientações quanto à configuração do sensor de umidade e temperatura ambiente e à interpretação das medidas coletadas, os requisitos mecânicos (encapsulamento) e de montagem são fundamentais para melhorar a exatidão das medidas.

O encapsulamento do sensor não foi foco do trabalho. No entanto, ele se mostra igualmente importante para o circuito dos nós sensores.

5.4 Conclusões sobre o acesso remoto

Para a aplicação de rede de sensores sem fio proposta, o acesso das medidas realizadas através da página *web* embarcada mostrou-se suficiente para apresentar o estado do sistema de forma remota. Esta página apresenta as medidas de cada sensor, além das informações de operação (bateria, nível de sinal e tempo de chegada das informações). Além disso, comporta um breve texto resumindo a aplicação da rede. O leiaute proposto para a página *web* pode ser modificado livremente, desde que não exceda o tamanho de 7kB e seja desenvolvido em linguagem HTML. O tempo de *download* dos arquivos de tamanho acima do recomendado influencia no funcionamento do nó coordenador, que pode enviar os dados da página *web* parcialmente ao destinatário, comprometendo a visualização dos dados no *browser*.

As limitações do acesso remoto são as seguintes:

- ✓ Restrição ao acesso aos arquivos gravados no cartão de memória: nesta versão de *firmware* para o coordenador, a leitura ou alteração dos arquivos presentes no cartão de memória SD de forma remota não é suportada. A decisão de não se disponibilizar este recurso foi tomada para não se permitir que houvesse uma interferência no andamento do processo de aquisição das medidas durante sua execução;
- ✓ O uso de um IP estático para conexão com a rede: O *firmware* atual não prevê o uso de IP dinâmico através de DHCP (*Dynamic Host Configuration Protocol*). Isso obriga o *firmware* a ser configurado previamente com um IP válido para uma

dada rede. Ao mesmo tempo, a necessidade de configuração garante que o IP seja conhecido desde o início – o que não é garantido com o uso do DHCP – e o coordenador tenha um único IP para o acesso remoto;

- ✓ O uso da linguagem HTML para a confecção da página traz a vantagem de tornar a página portátil e de fácil manutenção. Por outro lado, a limitação de recursos deixa a página muito simples, sem a possibilidade do uso de um gráfico ou de recursos visuais adicionais. Para o uso em questão, a página em formato HTML se mostrou satisfatório. O uso de recursos avançados, contudo, requer a adoção de outro microcontrolador compatível.

Apesar das limitações apresentadas, o acesso remoto aos nós permitiu verificar seu funcionamento e integração com a rede de sensores sem fio.

Problemas com *Jitter* não foram observados durante os testes e ensaios, por sempre serem adotados intervalos entre as leituras bem mais longos que o tempo de transmissão. Algumas perdas pontuais de pacotes foram detectadas. Essas perdas são atribuídas ao uso de um oscilador RC interno com precisão de 1% que define também o erro do *clock* do bloco da UART, responsável por transmitir os dados.

Os dados de estado da bateria e de nível de sinal são exibidos corretamente. Porém, a solução proposta inicialmente para se acompanhar o nível de bateria apresenta problemas. O nível de bateria mostra o valor de bateria correspondente ao esperado. Contudo, os valores monitorados se encontram na parte final da curva de descarga da bateria. Isso faz com que haja uma detecção tardia de descarga da bateria, deixando o usuário sem tempo hábil para realizar a manutenção. Para auxiliar este monitoramento, o nível de sinal recebido pelo nó pode ser utilizado, relacionando o nível do sinal recebido com bateria a plena carga (recepção inicial) com valor obtido no momento da análise. Porém, este critério também apresenta falhas, uma vez que o valor de intensidade do sinal recebido pode sofrer alterações devido a obstáculos entre os nós, mudanças no meio (chuva, nevoeiro, fumaça), entre outros. Por conta disso, sugere-se que, em versões futuras, o nó sensor faça a medida direta da bateria – através do uso de componentes adicionais – e tal valor seja enviado ao coordenador da rede, juntamente com as outras medidas realizadas pelo nó.

5.5 Custo estimado e comparações com sistemas existentes

Para justificar o argumento de que o sistema é de baixo custo, foram levantados os custos dos componentes dos nós sensores e do coordenador, assim como o custo total de um sistema comercial equivalente para a mesma aplicação. Para esta demonstração, os itens para montagem dos circuitos foram cotados com preços internacionais (sem adição de impostos) e a baixa quantidade (menor que 10 unidades), como é mostrado nas tabelas XVI e XVII:

Tabela XVI – Valores Comerciais para o Nó

Sensor.	
Componentes	Custo (US\$)
Microcontrolador	1,75
Sensor de Temperatura e Umidade	33,07
Sensor de Pressão	2,21
Módulo de Comunicação	32,00
Bateria	8,00
Placa de Circuito Impresso	15,20
Componentes Diversos (Passivos, Conectores)	5,00
Total (1 sensor)	97,23

Tabela XVII – Valores Comerciais para o Nó

Coordenador.	
Componentes	Custo (US\$)
Microcontrolador	8,75
PHY para Ethernet	5,79
LCD	8,59
Módulo de Comunicação	32,00
Componentes Diversos (Passivos, Conectores)	20,00
Placa de Circuito Impresso	20,25
Total (1 coordenador)	100,38

No levantamento de custo deste protótipo não foram incluídos mão-de-obra de montagem e nem lucro. Por outro lado, o custo de cada componente depende da quantidade adquirida. Esta negociação entre custo final e quantidade produzida, incluindo lucro (se houver) está além dos objetivos deste trabalho.

O valor estimado para o custo deste protótipo é utilizado como parâmetro de comparação com um sistema equivalente fornecido pela empresa *Libelium*. Trata-se dos nós *Waspnote* e do coordenador *Meshlium*. O sistema disponibilizado, no entanto, é mais genérico e

feito para ser utilizado em várias aplicações e ambientes. Para isso foram escolhidas, dentre as opções disponíveis, aquelas que definem um sistema mais próximo ao proposto neste trabalho.

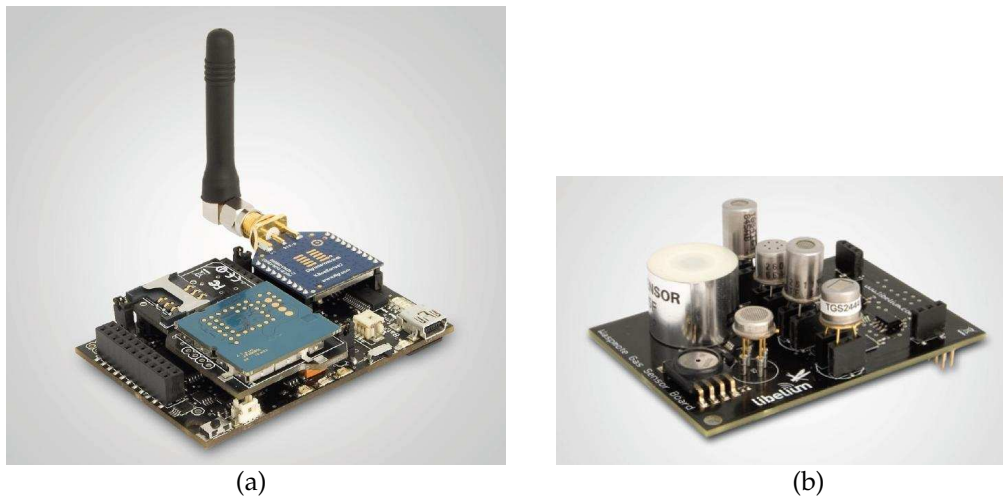


Figura 83 – (a) Mote sem fio *WaspMote* e (b) placa de sensores de gás.

Para um nó *Libelium* com as mesmas funções do nó proposto, é necessário definir o tipo de comunicação do mote (*WaspMote*), qual a placa de sensores adequada para a aplicação e quais os sensores a serem adotados no monitoramento. Para cada nó, foram definidos os itens enumerados na tabela XVIII [57]:

Tabela XVIII – Estimativa de custo do Nó Sensor da *Libelium*.

Componente necessário	Componentes disponíveis	Custo (US\$)
Mote	<i>WaspMote 802.15.4 PRO</i>	201
Placa de Sensores	<i>Gases Board</i>	154
Sensores	<i>Temperature</i>	4
	<i>Humidity</i>	27
	<i>Pressure</i>	35
Bateria	<i>Li-Ion Rechargeable 1150mAh</i>	25
Total (1 nó)		446

Para a definição do coordenador, foi escolhido o módulo *Meshlium* com suporte às redes 802.15.4 e Ethernet. A opção disponível ainda conta com conexão Wi-Fi:



Figura 84 – Coordenador *Meshlilum*.

Tabela XIX – Estimativa de custo do Nó Coordenador da Libelium.

Componente necessário	Componentes disponíveis	Custo (US\$)
Coordenador	<i>Meshlilum 802.15.4 PRO AP</i>	900
Total (1 coordenador)		900

Para um sistema funcional (com quatro nós e o coordenador) os custos foram comparados e os valores encontrados se encontram na tabela XX:

Tabela XX – Comparação entre os valores do sistema proposto e do sistema da Libelium.

Sistema	Custo de 4 nós sensores (US\$)	Custo do Coordenador (US\$)	Custo Total
Proposto	388,92	100,38	489,30
<i>Libelium</i>	1784	900	2684

O custo do sistema proposto é de aproximadamente 20% da solução comercial avaliada. Nesta, nós e coordenador foram concebidos para serem versáteis e genéricos, podendo ser

utilizados em várias aplicações. Estes nós, por exemplo, têm com itens opcionais receber diversas placas de sensores (para várias finalidades), armazenar os próprios dados em memórias SD próprias e se comunicarem através de outras interfaces como *Wi-Fi*, *Bluetooth* e GPRS. O coordenador, por sua vez, também tem como adotar outras interfaces e possui uma interface web gráfica mais aprimorada e com mais opções de personalização.

Além do custo mais alto o sistema comercial precisa ser programado e configurado de acordo com os sensores e módulos de comunicação adotados, através de uma API fornecida pelo fabricante, aumentando o tempo e o custo de desenvolvimento. O sistema proposto neste trabalho não necessita de nenhuma configuração adicional.

5.6 Sugestões de trabalhos futuros

A seguir sugerimos algumas ações e trabalhos futuros que podem melhorar e ampliar as aplicações da rede de sensores sem fio propostas nesta dissertação:

- ✓ Desenvolvimento de um encapsulamento para os nós: Para a plena utilização dos nós, é importante que os nós sejam encapsulados de forma adequada. Para isso, tornam-se necessários pesquisas e ensaios para se desenvolver um invólucro que proteja os sensores das intempéries e ao mesmo tempo não interfira na medida. O desafio consiste em equilibrar uma exposição máxima do sensor – requerida para leitura de temperatura ambiente – com uma exposição mais moderada – necessária para a leitura de umidade. O encapsulamento deve ainda proteger o circuito do nó sensor para evitar problemas com a condensação do vapor d'água;
- ✓ Topologia aprimorada de rede: A topologia da rede em estrela pode ser inadequada para algumas aplicações avançadas da rede de sensores – sobretudo no caso do uso de mais nós e cobertura de áreas com densa vegetação maiores que 100 metros de raio. Para isso, sugere-se utilizar outros módulos de comunicação que permitam a configuração do nó sensor como dispositivo final e como roteador. Isso exige um novo levantamento da energia gasta por estes nós

para se estimar qual a autonomia de funcionamento de um nó, bem como de toda a rede;

- ✓ Customização do Transceptor 802.15.4: Para o uso proposto, o módulo de comunicação adotado está sendo subutilizado, já que muitas outras funções disponíveis por ele não são empregadas. Este módulo foi escolhido por ser uma solução comercial conhecida, confiável e com custos compatíveis com o projeto. Porém, um transceptor 802.15.4 personalizado e integrado ao nó diminuiria em boa parte os custos por nó construído. Para isso, propõe-se que, em um projeto futuro, um transceptor simplificado seja utilizado para se estabelecer a comunicação entre os nós e o coordenador da rede;
- ✓ Integração de outros sensores aos nós: o microcontrolador adotado e seus recursos permite o uso de sensores de temperatura, umidade e pressão diferentes dos propostos por este trabalho, bem como a adoção de sensores para outras medidas (como CO₂, O₃, raios, etc). Para esta utilização, o *firmware* dos nós deve ser adequado aos novos sensores utilizados e o coordenador deve ser preparado para receber os dados das novas medidas.
- ✓ Colheita de energia (*energy scavenging*): Sendo o baixo consumo de energia uma das características do nó sensor, propõe-se que, para novas versões de *hardware*, sejam utilizados circuitos de colheita de energia para se prolongar ainda mais a autonomia do nó sensor, ou até mesmo seu uso contínuo, sem necessidade de carga da bateria. Para isso, pode-se utilizar alguma das alternativas estudadas neste trabalho (painéis solares, microgeradores, RF) ou ainda outros não abordados na revisão de literatura.

Capítulo 6

Referências Bibliográficas

- [1] José de Fátima da Silva. *El-Niño: O fenômeno climático do século*. Brasília. Thesaurus. 2000.
- [2] A. C. de Araújo, A. D. Nobre, B. Krujti, J. A. Elbers, R. Dallarosa, P. Stefani, C. von Randow, A. Manzi, A. D. Culf, H. C. Gash, R. Valentini e P. Kabat. *Comparative measurements of carbon dioxide fluxes on two nearby towers in a central Amazonian rainforest: the Manaus LBA Site*. Journal of Geophysical Research, 2006.
- [3] Jason L. Hill. *System Architecture for Wireless Sensor Networks*. Ph. D. Thesis. University of California, Berkeley, 2003
- [4] John F. Weaver. *History of Thunderstorm Forecasting: Part II: The Instruments*. Site de internet, 1982.
<http://www.stormtrack.org/archive/0522.htm> (Acessado em 21/08/2013, 16:46).
- [5] Frank Zöllner e Johannes Nattan. *Leonardo da Vinci: The Complete Painting & Drawings*. Cologne, Taschen, 2003.
- [6] David Brewster. *Thermometer* The Edinburgh Encyclopaedia: Vol XVIII, Plate DXXIV, University of California: Los Angeles, p 603. 1832.
- [7] U. Grigull. *Fahrenheit. a Pioneer of Exact Thermometer*. Proceeding of the Heat Transfer, San Francisco, pp 9-18, 1966.
- [8] William F. Magie. *A Source Book in Physics: Letter to Michelangelo Ricci concerning the Barometer (1644)*. McGraw-Hill, 1935.
- [9] Simon M Sze. *Semiconductor Sensors*. John Wiley & Sons, 1994

- [10] Gerard C. M. Meijer. *Smart Sensor Systems*. John Wiley & Sons, 2008
- [11] Ramón Pallàs-Areny e John Webster. *Sensors and Signal Conditioning*. J Wiley, 1991.
- [12] W Göpel, J. Hesse, J. N. Zemel. *Sensors A Comprehensive Survey: Volume 3 – Chemical and Biochemical Sensors: Part II*. VCH. 1992.
- [13] Vitor Garcia. *Sensor de Pressão Microeletrônico Baseado no Efeito PiezoMOS*. Dissertação de Mestrado. Universidade Estadual de Campinas, 2006.
- [14] Maxim Integrated Products. AN871 – *Desmistifying Piezoreistive Pressure Sensors*. Notas de Aplicação, 2001.
- [15] CiS Forschungsinstitut für Mikrosensorik und Photovoltaik. *Silizium-Drucksensoren mit piezoresistiver Brücke*. Datasheet, 2010.
- [16] Andrew N. Sloss, Dominic Symes e Chris Wright. *ARM Systems Developer's Guide*. Elsevier, 2004.
- [17] Ian Downes, Leili Rad e Hamid Aghajam. *Development of a mote for Wireless Sensor Networks*. Int. Proc. of Cognitive Systems and Interactive Sensors (COGIS). Paris, 2006
- [18] NXP Semiconductors. *Get Better Code Density than 8/16 bit MCUs – NXP LPC1100 Cortex M0*. Notas de Aplicação, 2009
- [19] NXP Semiconductors. *LPC2378 User's Manual*. Manual do Usuário, 2010.
- [20] Sergio Scaglia. *The Embedded Internet*. Addison-Wesley. 2007.
- [21] USB Implementers Forum, Inc. *USB Specifications*. Página de Internet. www.usb.org (Acessado em 21/08/2013 – 16:46).
- [22] Rogério Sustiano e Carlos A. dos Reis Filho. *Barrier Synchronization Simulator for Wireless Sensor Networks*. Int. Proc. Caribbean Conference on Devices Circuits and Systems. Mexico, 2008
- [23] David Tse e Pramod Visvanath. *Fundamentals of Wireless Communication*. Cambridge University Press. 2005
- [24] Jean-Paul Linnartz. *Wireless Communication*. Página de Internet, 2006. <http://www.sps.ele.tue.nl/members/J.P.Linnartz/web/reference/index.htm> (Acessado em 21/08/2013 - 16:50).
- [25] Texas Instruments. *ISM-Band and Short Range Device Regulatory Compliance Overview*. Notas de Aplicação, 2005.
- [26] International Telecommunications Union. *Available frequency bands for ISM applications*. Página de Internet. <http://www.itu.int/ITU-R/terrestrial/faq/index.html#g013> (acessado em 21/08/2013 - 16:51).
- [27] Federal Communications Commission. *License-Exempt Wireless Applications for Public Safety*. Página de Internet. <http://www.fcc.gov/help/public-safety-tech-topic-10-license-exempt-wireless-applications-public-safety> (acessado em 21/08/2013 - 16:52)

- [28] The Institute of Electrical and Electronics Engineers, Inc. *Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs)*. IEEE Standard for Information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Specific Requirements. USA, 2006.
- [29] The Institute of Electrical and Electronics Engineers, Inc. *Part 15.1: Wireless medium access control (MAC) and physical layer (PHY) specifications for wireless personal area networks (WPANs)*. IEEE Standard for Information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Specific Requirements. USA, 2005.
- [30] The Institute of Electrical and Electronics Engineers, Inc. *Part 15.3: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for High Rate Wireless Personal Area Networks (WPANs)*. IEEE Standard for Information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Specific Requirements. USA, 2005.
- [31] The Institute of Electrical and Electronics Engineers, Inc. *Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*. IEEE Standard for Information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Specific Requirements. USA, 2007.
- [32] The Institute of Electrical and Electronics Engineers, Inc. *Part 16: Air Interface for Broadband Wireless Access Systems*. IEEE Standard for Local and metropolitan area networks. USA, 2009.
- [33] The Institute of Electrical and Electronics Engineers, Inc. *Part 20: Air Interface for Mobile Broadband Wireless Access Systems Supporting Vehicular Mobility – Physical and Media Access Control Layer Specification*. IEEE Standard for Local and metropolitan area networks. USA, 2008.
- [34] The Institute of Electrical and Electronics Engineers, Inc. *Part 22: Cognitive Wireless RAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Policies and Procedures for Operation in the TV Bands*. IEEE Standard for Information Technology – Telecommunications and information exchange between systems Wireless Regional Area Networks (WRAN) – Specific requirements. USA, 2011.
- [35] ZigBee Standards Organization. *ZigBee Specification*. The ZigBee Alliance, 2008
- [36] Abdalkarim Awad, Lei Shi, Reinhard German e Falko Dressler. *Advantages of Virtual Addressing for Efficient and Failure Tolerant Routing in Sensor Networks*. 6th Int. Conf. on Wireless On-Demand Network Systems and Services. University of Erlanger 2009
- [37] Brad Karp e H. T. Kung. *GPSR: Greedy Perimeter Stateless Routing for Wireless Networks*. Proceedings of the 6th Annual ACM/IEEE International Conference on Mobile Computing and Networking. Boston, 2000
- [38] Vika Controls. *Protocol ZigBee*. Notas de Aplicação. São Paulo, 2008.

- [39] Freescale Semiconductors. *MC13202 2.4 GHz ISM Band Low Power Transceptor*. Folha de dados. USA, 2008
- [40] Micrel Inc. *MICRF505 850MHz and 950MHz ISM Band Transceptor*. Folha de Dados USA, 2006.
- [41] Memsic Corporation. *Wireless Modules*. Site de internet. <http://www.memsic.com/wireless-sensor-networks/> (acessado em 21/08/2013 - 16:58)
- [42] Digi International. *XBee® ZigBee®/Mesh RF Modules*. Folha de Dados. USA, 2008.
- [43] Battery University. *Learn about batteries*. Página de Internet <http://batteryuniversity.com/learn/> (acessado em 21/08/2013 - 16:59).
- [44] Renewable Energy UK. *Solar Articles*. Página de Internet. <http://www.reuk.co.uk/solar.htm> (acessado em 21/08/2013 - 17:00).
- [45] Kinetron The Micro Energy Company. *Microgenerator Technology*. Página de Internet. <http://www.kinetron.nl/> (acessado em 21/08/2013 - 17:00).
- [46] Bing Jiang, Joshua Smith, Matthai Philipose, Sumit Roy, Kishore Sundara Rajan, Alexander Mamishev. *Energy Scavenging for Inductively Coupled Passive RFID Systems*. IEEE Transactions on Instrumentations and Measurements. Vol 56, No 1, 2007.
- [47] RF Wireless Sensors Blog. *Powercast demonstrates wireless charging of credit card size smart cards*. Página de Internet. <http://www.rfwirelessensors.com/> (acessado em 21/08/2013 - 17:02).
- [48] Gruia Calinescu, Ophir Frieder e Peng-Jun Wan. *Handbook on Theoretical Algorithmic Aspects of Sensor, Ad Hoc Wireless, and Peer-to-Peer Networks – Chapter 10 A Survey of Algorithms for Power Assignment in Wireless Ad Hoc Networks*. Taylor & Francis Group, LLC, 2006
- [49] Brett Warneke. *Handbook of Sensor Networks: Compact Wireless and Wired Sensing Devices – Chapter 05: Miniaturizing Sensor Network with MEMS*. CRC Press LLC, 2005
- [50] Ramon Casanella, Óscar Casas, Marco Ferrari, Vittorio Ferrari e Ramon Pallàs-Areny. *Synchronous Demodulator for Autonomous Sensors*. IEEE Transactions on Instrumentations and Measurements. Vol 56, No 4, 2007.
- [51] Luciana Mendonça, Alzira Conceição, Juliana Piedade, Vania Carvalho, Vanessa Theodoro. *Caracterização da Composição Química e do Rendimento dos Resíduos Industriais do Limão Tahiti (Citrus latifolia Tanaka)*. Ciência e Tecnologia dos Alimentos. Campinas, 2006
- [52] NXP Semiconductors. *LPC111x/LPC11Cxx User manual*. Manual de Usuário. USA, 2011
- [53] Sensirion AG. *Datasheet SHT1x – Humidity and Temperature Sensors*. Folha de Dados. Suíça, 2010
- [54] Freescale Semiconductor. *MPL115A2 Miniature I²C Digital Barometer*. Folha de Dados. USA, 2010.
- [55] NXP Semiconductors. *LPC23xx User manual*. Manual de Usuário. USA, 2009
- [56] NXP Semiconductors. *Li-Ion battery charger with P89LPC916*. Notas de Aplicação, 2007
- [57] Libellium. *Libellium Product Catalogue*. Catálogo de Produtos, 2011

Apêndice A Folha de dados do

Sensor de Temperatura e Umidade -

Sensirion

A.1 Performance do Sensor [53]

A.1.1 TEMPERATURA

Tabela XXI – Especificações de Temperatura para o sensor SHT15.

Parâmetro	Condição	mín	tip	máx	unidades
Resolução		0,04	0,01	0,01	°C
		12	14	14	Bits
Precisão SHT15	típica		±0,3		°C
	máxima	Ver fig 85			
Repetibilidade			±0.1		°C
Faixa de Operação		-40		123,8	°C
		-40		254,9	°F
Histerese			±1		°C
Tempo de resposta	$\tau(63\%)$	5		30	s
Desvio a longo prazo			<0,4		°C/ano

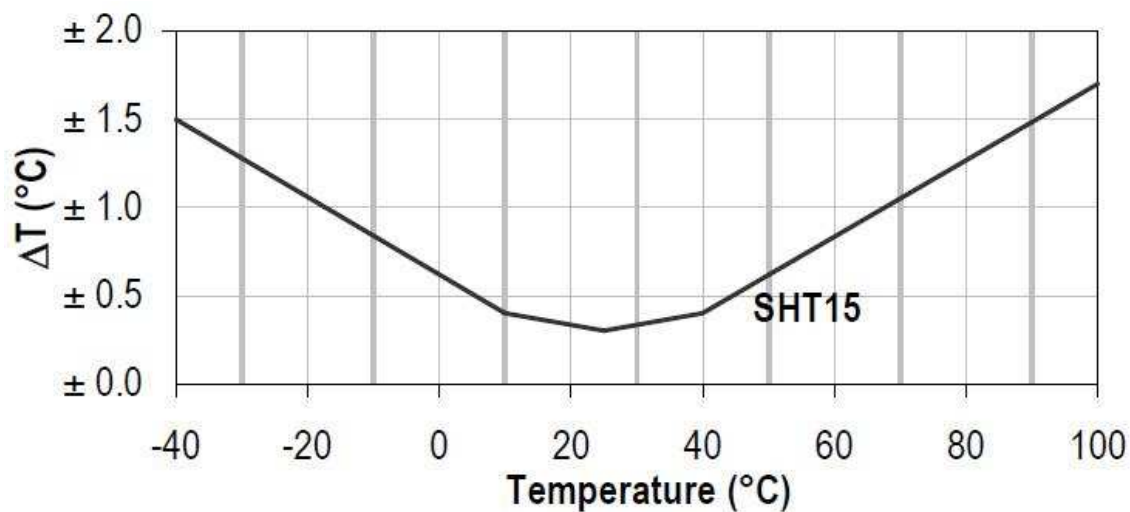


Figura 85 – Máxima Tolerância do valor de temperatura para o sensor SHT15.

A.1.2 UMIDADE RELATIVA

Tabela XXII – Especificações de Umidade Relativa para o sensor SHT15.

Parâmetro	Condição	mín	tip	máx	unidades
Resolução		0,4	0,05	0,05	%Umid. Rel.
		8	12	12	Bits
Precisão SHT15	típica		±2,0		%Umid. Rel.
	máxima	Ver fig 85			
Repetibilidade			±0.1		%Umid. Rel.
Histerese			±1		%Umid. Rel.
Não-linearidade	linearizado		<<1		%Umid. Rel.
Tempo de resposta	τ(63%)		8		s
Faixa de Operação		0		100	%
Desvio a longo prazo	Normal		<0,5		%Umid. Rel./ano

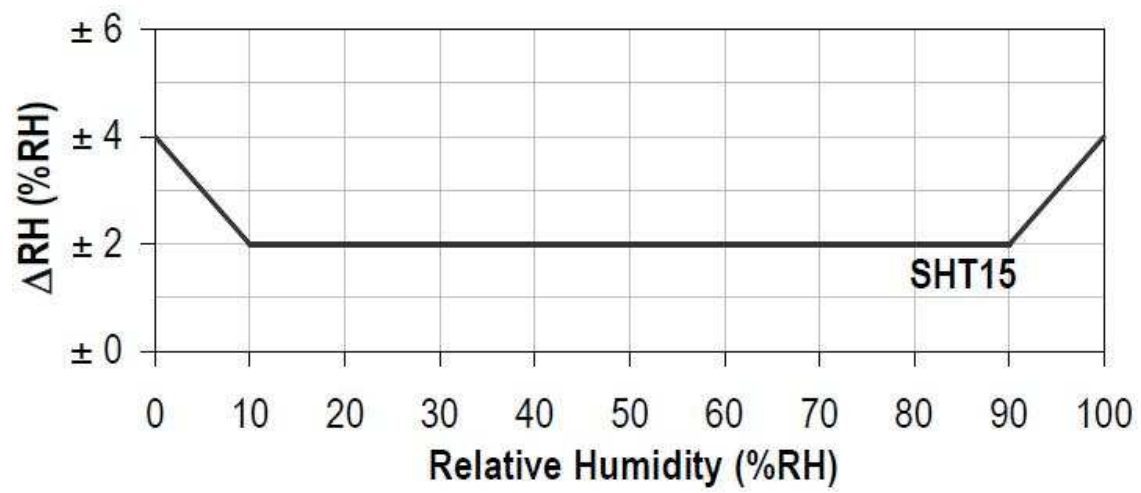


Figura 86 – Máxima tolerância do valor de umidade relativa a 25 C para o sensor SHT15.

Apêndice B Folha de dados do

Sensor de Pressão Atmosférica-

Freescle

B.1 Características de Operação [54]

Tabela XXIII – Especificações de Pressão para o sensor MPL115A2.

Parâmetros	Condições	Mín	Típ	Máx	Unidades
Faixa de operação		50		115	kPa
Resolução			0,15		kPa
Precisão	-20 °C a 85 °C		±1		kPa
Tempo de conversão (Início de Medição de Pressão)	Tempo entre comando de início de conversão e dados disponíveis no Registro de Pressão		0,6	0,7	ms

Tabela XXIV – Especificações de Temperatura para o sensor MPL115A2.

Parâmetros	Condições	Mín	Típ	Máx	Unidades
Faixa de operação		-40		105	°C
Resolução	ADC de temperatura é 472 a 25 °C		-5,35		contagens/°C
Tempo de conversão (Início de Medição de Temperatura)	Tempo entre comando de início de conversão e dados disponíveis no Registro de Temperatura		0,6	0,7	ms
Tempo de conversão (Início de Medição de Pressão e Temperatura)	Tempo entre comando de início de conversão e dados disponíveis nos Registros de Pressão e Temperatura		0,8	1	ms

Apêndice C Medidas em Câmara

Climática

C.1 Comparação a 25°C

Seguem os gráficos de temperatura dos sensores a 25°C, comparados com o sensor de referência do INPA:

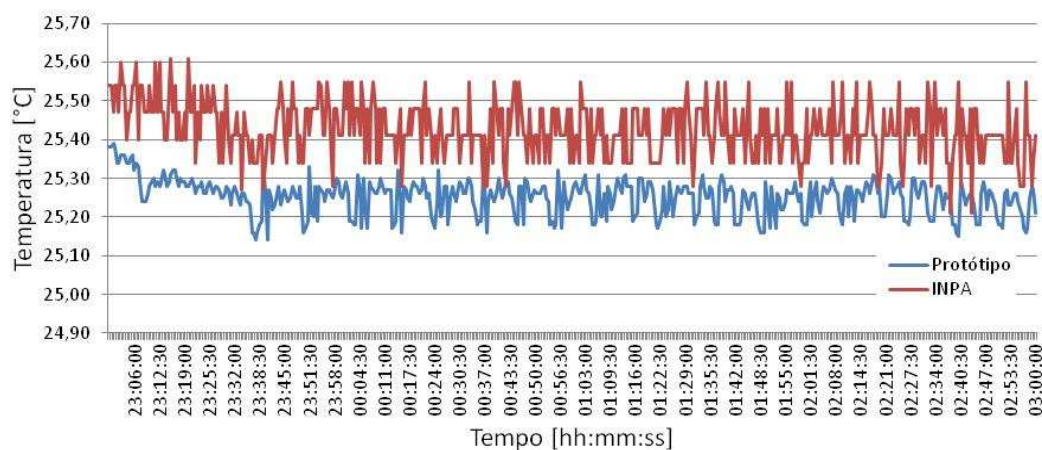


Figura 87 – Temperatura do Protótipo e Sensor de Referência na posição 1, a 25°C.

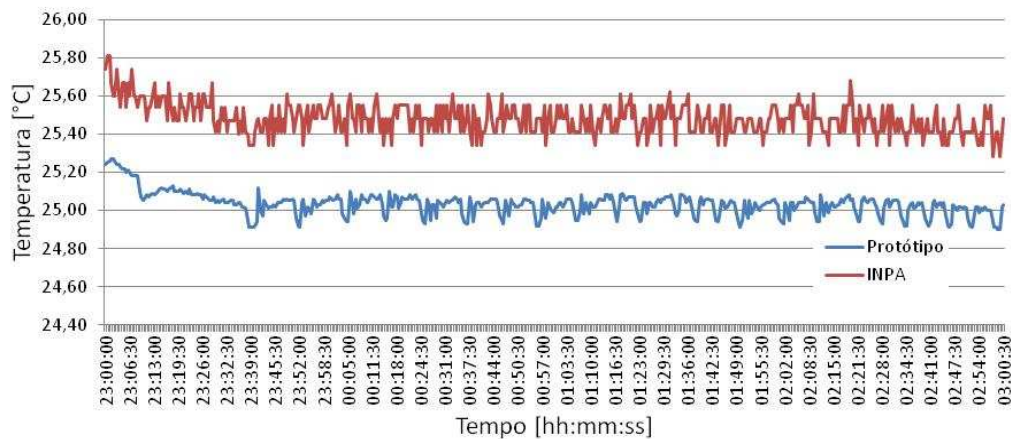


Figura 88 – Temperatura do Protótipo e Sensor de Referência na posição 2, a 25°C.

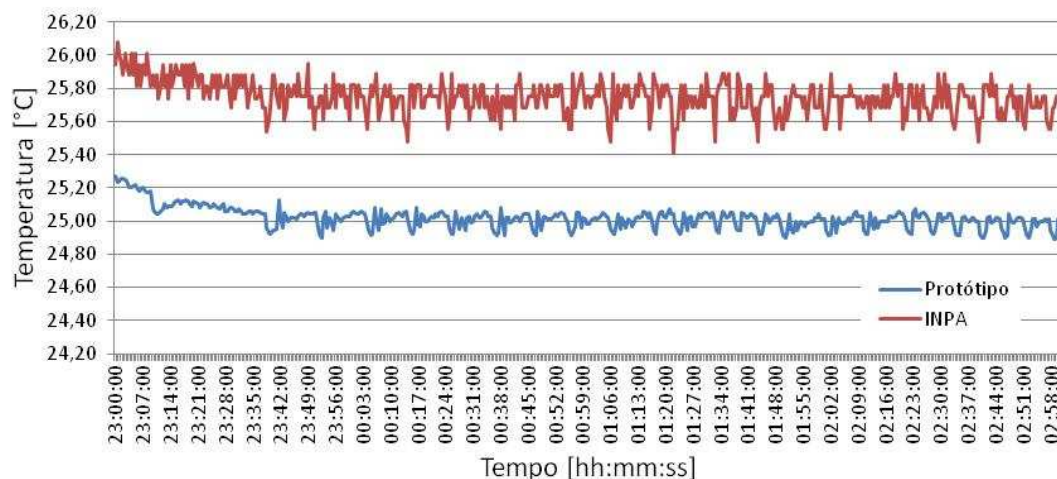


Figura 89 – Temperatura do Protótipo e Sensor de Referência na posição 3, a 25°C.

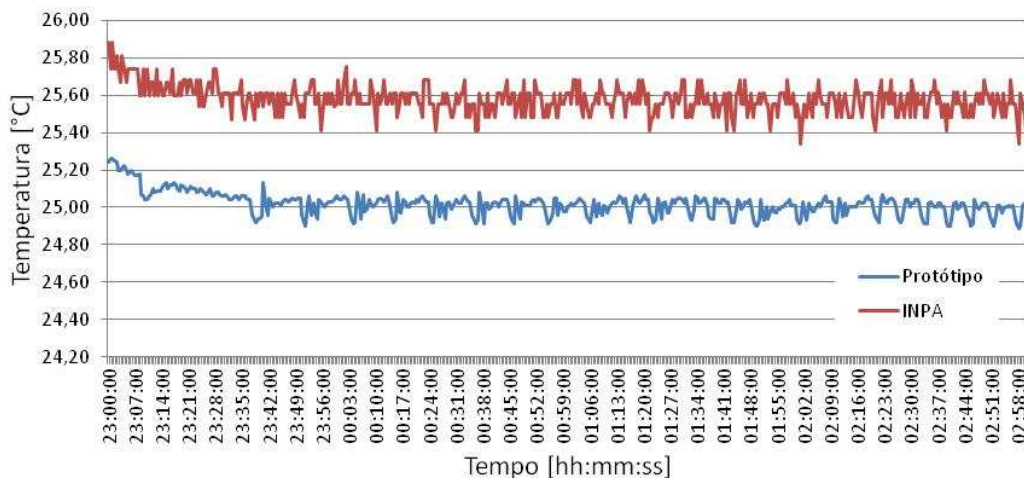


Figura 90 – Temperatura do Protótipo e Sensor de Referência na posição 4, a 25°C.

C.2 Comparação a 15°C

Seguem os gráficos de temperatura dos sensores a 15°C, comparados com o sensor de referência do INPA:

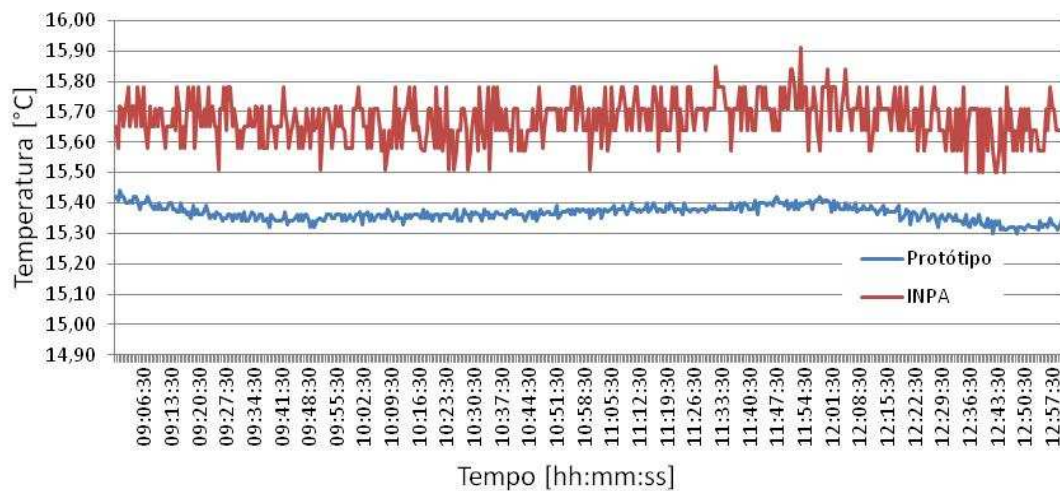


Figura 91 – Temperatura do Protótipo e Sensor de Referência na posição 1, a 15°C.

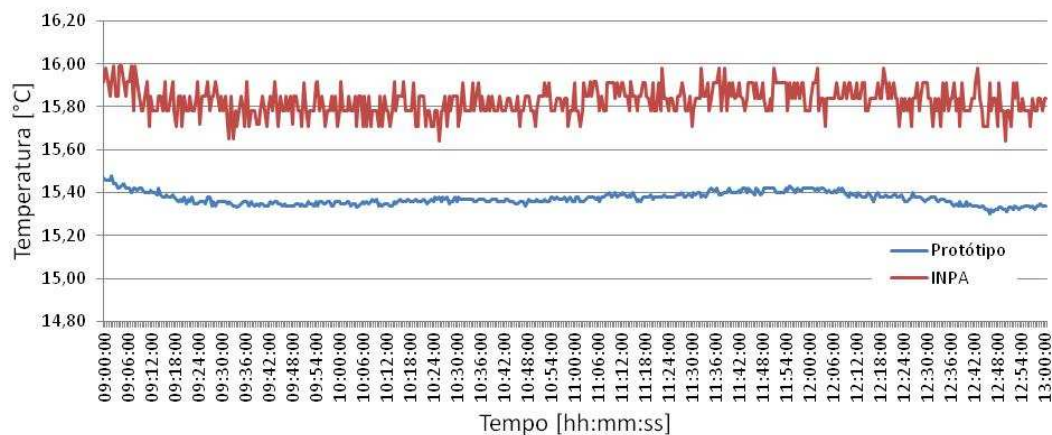


Figura 92 – Temperatura do Protótipo e Sensor de Referência na posição 2, a 15°C.

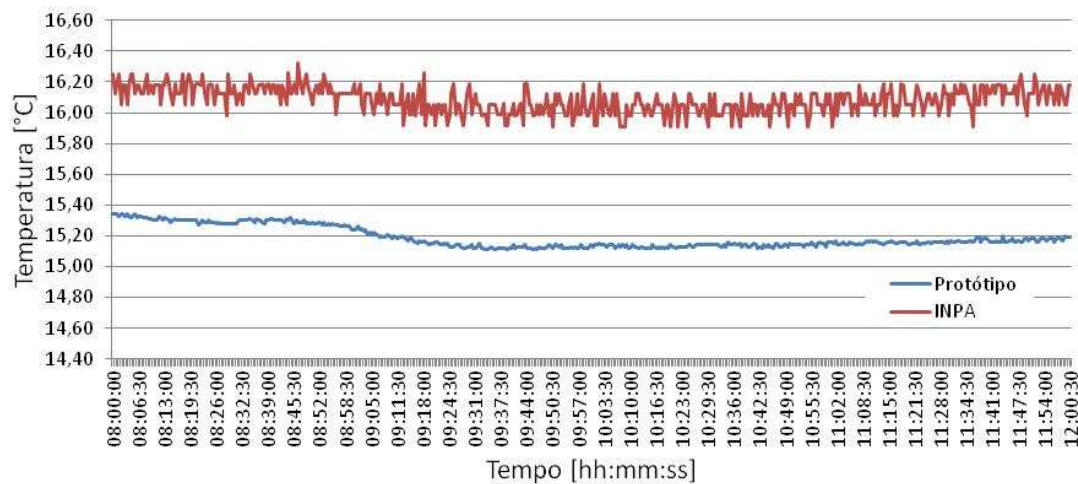


Figura 93 – Temperatura do Protótipo e Sensor de Referência na posição 3, a 15°C.

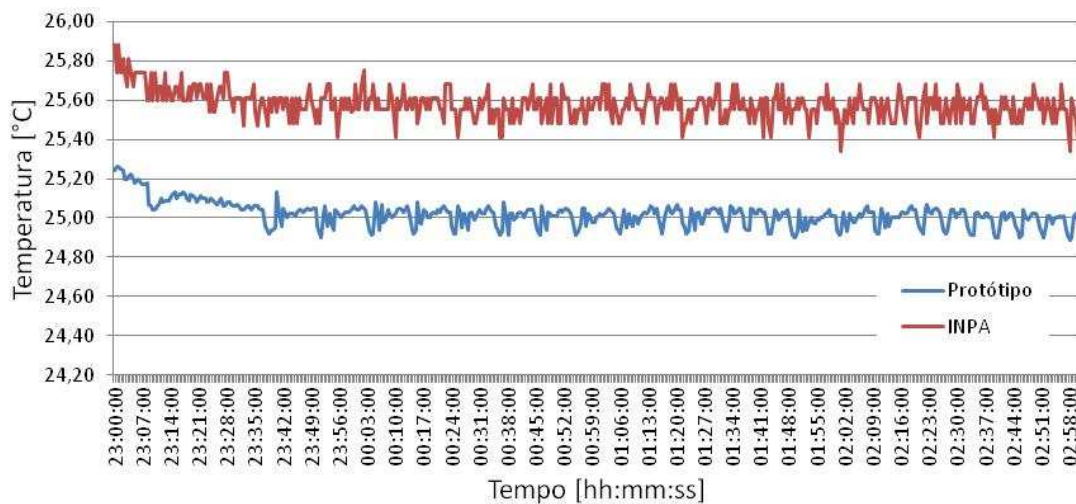


Figura 94 – Temperatura do Protótipo e Sensor de Referência na posição 4, a 15°C.

C.3 Comparação a 80%

Seguem os gráficos de temperatura dos sensores a 80% medidos a temperatura de 25°C, comparados com o sensor de referência do INPA:

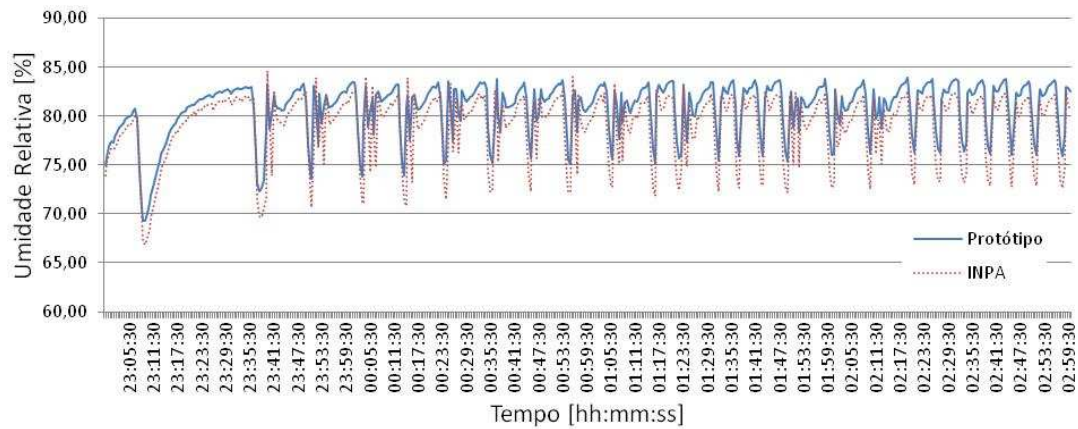


Figura 95 – Umidade Relativa do Protótipo e Sensor de Referência na posição 1, a 80%.

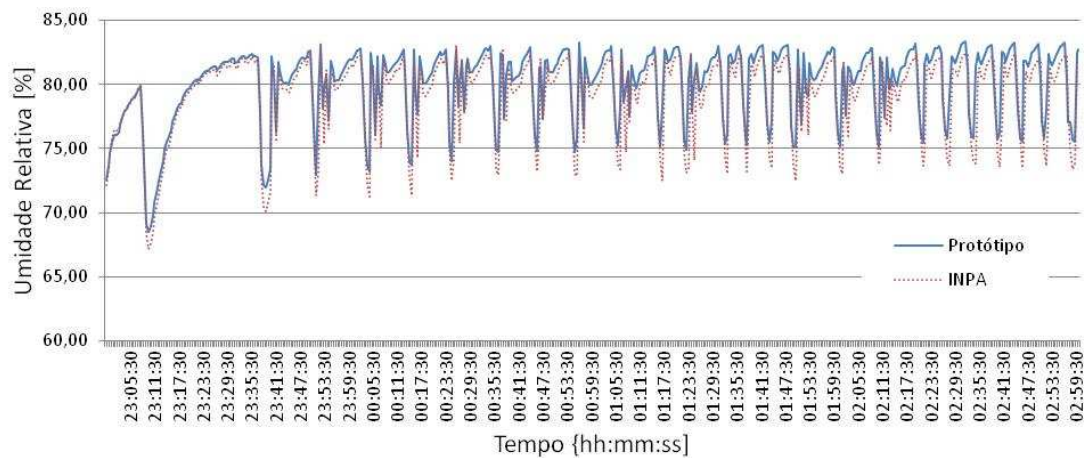


Figura 96 – Umidade Relativa do Protótipo e Sensor de Referência na posição 2, a 80%.

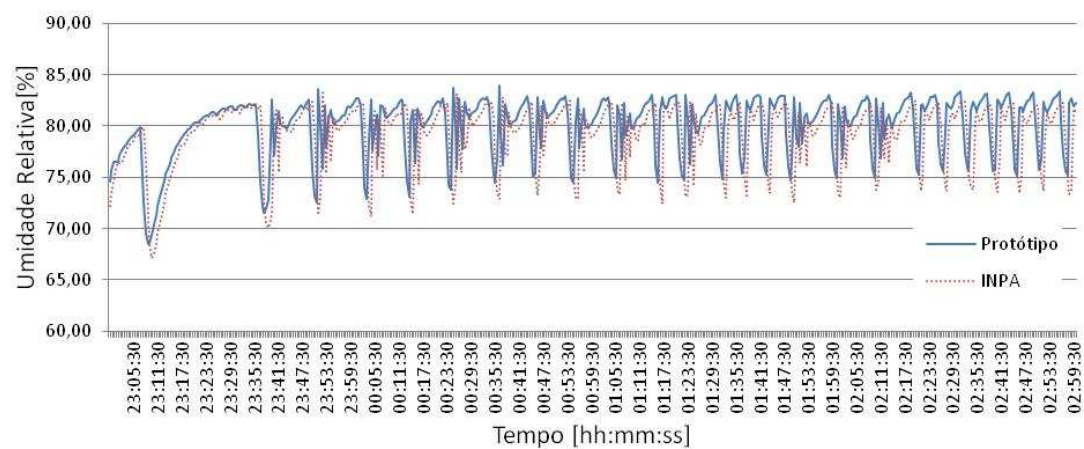


Figura 97 – Umidade Relativa do Protótipo e Sensor de Referência na posição 3, a 80%.

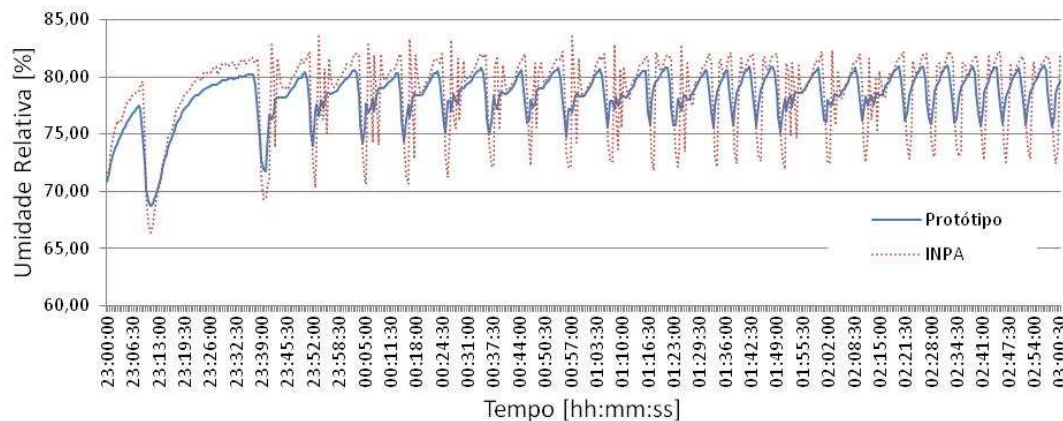


Figura 98 – Umidade Relativa do Nó Protótipo e Sensor de Referência na posição 4, a 80%.

C.4 Comparação a 40%

Seguem os gráficos de temperatura dos sensores a 40% medidos a temperatura de 15°C, comparados com o sensor de referência do INPA:

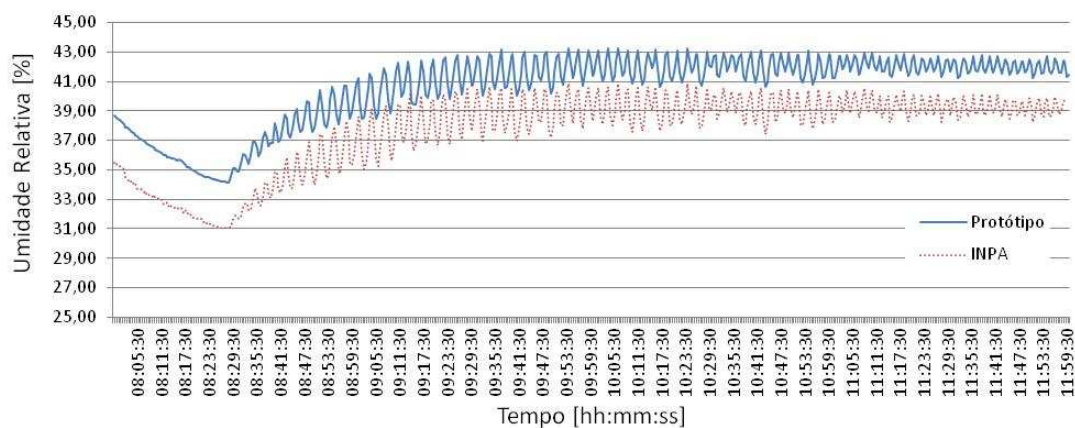


Figura 99 – Umidade Relativa do Protótipo e Sensor de Referência na posição 1, a 40%.

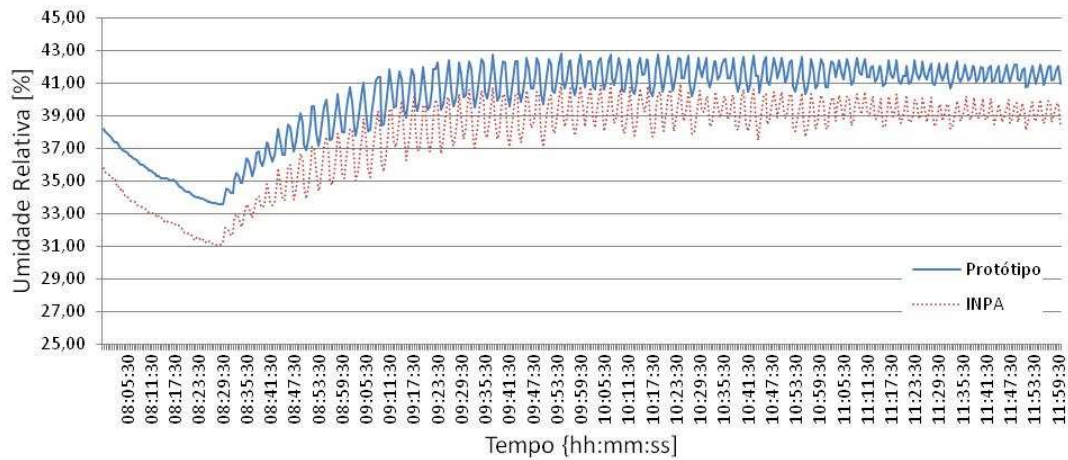


Figura 100 – Umidade Relativa do Protótipo e Sensor de Referência na posição 2, a 40%.

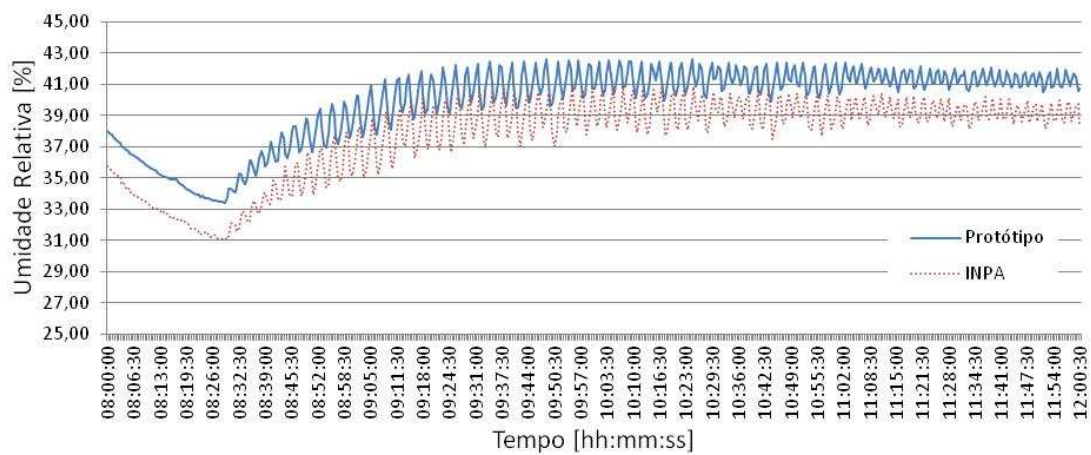


Figura 101 – Umidade Relativa do Protótipo e Sensor de Referência na posição 3, a 40%.

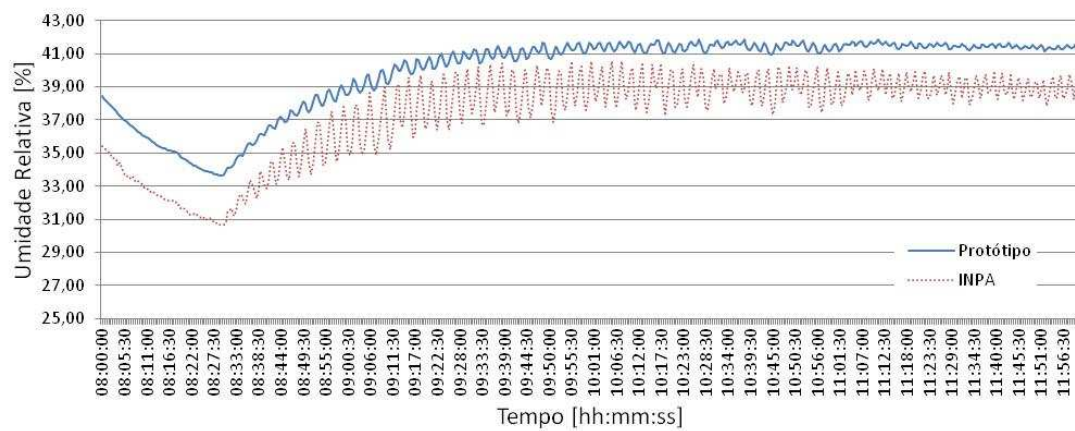


Figura 102 – Umidade Relativa do Protótipo e Sensor de Referência na posição 4, a 40%.

Apêndice D Códigos-fonte

D.1 Código-fonte principal do nó sensor

```
/* *****  
** Wireless Sensor Network – Sensor node code source  
**  
** This project configures the microcontroller to be used within the sensor node.  
** It configures the watchdog oscillator at lowest frequency to clock the 16-bit Timer 0  
** in run and deep sleep modes. The device generates a repetitive self wake-up from deep-sleep mode and  
** toggles a port pin when in run mode.  
**  
** Watchdog Osc configured at slowest freq (FREQSEL = 0x0; DIVSEL = 0x1F): 9600 Hz  
**  
** The wake-up time can be measured by using the following pins:  
** Trigger pin (PIO0_8/CT16B0_MAT0)  
** Timer Output to get the device out of the deep-sleep mode.  
**  
** Pin 2_0 is used to flash program  
**  
** The wakeup time is the difference between the rising edge of the trigger pin and  
** the first falling edge of the Strobe pin.  
**  
***** */  
  
#ifdef __USE_CMSIS  
#include "LPC11xx.h"  
#endif  
  
#include "sth1x.h"  
#include "uart.h"  
#include "mpl115a2.h"  
#include "timer16_tk.h"  
#include "defines.h"  
  
/* ENABLE_CLKOUT = 0      PIO0_1 is configured as a GPIO  
   ENABLE_CLKOUT = 1      PIO0_1 outputs CLKOUT */  
#define ENABLE_CLKOUT    0
```

```

/* DEEP_SLEEP_MODE = 0: Device will go into Sleep Mode
   DEEP_SLEEP_MODE = 1: Device will go into Deep Sleep Mode */
#define DEEP_SLEEP_MODE 1

/* ENABLE_TIMER specifies the self wake up feature */
#define ENABLE_TIMER 1

void wait(uint32_t time);
void config_ios(void);

value humi_val,temp_val;
unsigned char error,checksum;

unsigned char Error[12] = {"error!\r\n0"};
unsigned char cTemp[13] = {0};
unsigned char cHumi[13] = {0};
unsigned char cPres[13] = {0};

int16_t dPressure = 0;
float f_Pressure = 0;

unsigned short BattStatus = FULL_BAT;
unsigned char cBattStatus[4] = {0};

void BOD_IRQHandler(void)
{
    switch (BattStatus)
    {
        case FULL_BAT:
        {
            BattStatus = THIRD_BAT;
            SYSCON->BODCTRL = 2 << 2; // VDD < 2.52V
            break;
        }
        case THIRD_BAT:
        {
            BattStatus = HALF_BAT;
            SYSCON->BODCTRL = 1 << 2; // VDD < 2.22V
            break;
        }
        case HALF_BAT:
        {
            BattStatus = CRIT_BAT;
            SYSCON->BODCTRL = 0; // VDD < 1.65V
            break;
        }
        case CRIT_BAT:
        {
            BattStatus = NO_BAT;
            break;
        }
    }
}

```

```

    default:
    {
        BattStatus = FULL_BAT;
        SYSCON->BODCTRL = 3 << 2; // VDD < 2.80V
        break;
    }
}

void WAKEUP_IRQHandler(void)
{
    Clear_MAT0(0);

    /* Clear the pending interrupt of the start logic */
    SYSCON->STARTRSRP0Clr |= (1<<(8));
    return;
}

int main(void)
{
    int i;
    SystemInit();

    SYSCON->IRCCTRL = TRIM_IRC; /* Trimming Sensor 1 */

    SYSCON->SYSAHBCLKCTRL = /* Enable SYSCLK Peripherals */
        (1<<0) //SYS Clock
        | (1<<1) //ROM
        | (1<<2) //RAM
        | (1<<3) //FLASHREG
        | (1<<4) //FLASHARRAY
        | (1<<6) //GPIO
        | (1<<16) //IOCON
        ;

    /* Configure Wakeup I/O */
    /* Specify the start logic to allow the chip to be waken up */
    SYSCON->STARTAPRP0 |= (1<<(8)); // Rising edge
    SYSCON->STARTRSRP0Clr |= (1<<(8)); // Clear pending bit
    SYSCON->STARTERP0 |= (1<<(8)); // Enable Start Logic

    /* Specify output pin to toggle an LED */
    GPIO1->DIR |= (1<<8) | (1<<9);
    GPIO0->DIR |= (1<<11);
    GPIO1->DATA |= (0x300);

    #if ENABLE_CLKOUT
    /* Output the Clk onto the CLKOUT Pin PIO0_1 to monitor the freq on a scope */
    IOCONFIG->PIO0_1 = 1;
    /* Select the MAIN clock as the clock out selection since it's driving the core */
    SYSCON->CLKOUTCLKSEL = 3;
    #endif
}

```

```

/* Set CLKOUTDIV to 6 */
SYSCON->CLKOUTCLKDIV = 1; // CLKOUT Divider = 10
/* Enable CLKOUT */
SYSCON->CLKOUTCLKUEN = 0;
SYSCON->CLKOUTCLKUEN = 1;
while (!(SYSCON->CLKOUTCLKUEN & 0x01));
#endif /* ENABLE_CLKOUT */

if (~GPIO0->DATA & 0x800)
{
    GPIO1->DATA &= ~(0x200);
    while(1); // Prepare HW to a new FW
}

/* Brown out configuration */
SYSCON->BODCTRL = 3 << 2; // VDD < 2.80V
NVIC_EnableIRQ(BOD_IRQn);

#if ENABLE_TIMER
    /* Initialize 16-bit Timer 0 */
    Init_Timer(0, 30, TIME_BASE); // 30s
#endif /* ENABLE_TIMER */

/* Turn on the watchdog oscillator */
SYSCON->PDRUNCFG &= ~(1<<6);

/* Run watchdog oscillator */
SYSCON->WDTOSCCTRL = (0x1<<5) | 0x1F; // SLOWEST WDT FREQ

/* Switch MAINCLKSEL to Watchdog Oscillator */
SYSCON->MAINCLKSEL = 2;
SYSCON->MAINCLKUEN = 0;
SYSCON->MAINCLKUEN = 1;
while (!(SYSCON->MAINCLKUEN & 0x01));
SYSCON->SYSAHBCLKDIV = 1;

/* Make sure that only the watchdog oscillator & Flash are running now */
SYSCON->PDRUNCFG = ~( (1<<0) | (1<<1) |
    (1<<2) | (1<<3) | (1<<5) |
    (1<<6) | (1<<7) |
    (1<<9) | (1<<12));

#if ENABLE_TIMER
    /* Initialize 16-bit Timer 0 */
    Enable_Timer(0);
#endif /* ENABLE_TIMER */

#if DEEP_SLEEP_MODE
    /* Clear the Deep Sleep Flag */
    PMU->PCON |= (1<<8);

```

```

/* Specify peripherals to be powered up again when returning from deep sleep mode */
SYSCON->PDAWAKECFG = SYSCON->PDRUNCFG;

/* Configure PDSLEEPCFG */
SYSCON->PDSLEEPCFG = 0x000018BF;

/* Specify Deep Sleep mode before entering mode */
SCB->SCR |= (1<<2); //Set SLEEPDEEP bit
#endif /* DEEP_SLEEP_MODE */

/* Reconfigure the IOs */
config_ios();

UARTInit(115200);
GPIO0->DATA |= (1<<11);
GPIO3->DATA |= (1<<5);
GPIO3->DATA |= (1<<4);
GPIO1->DATA |= (0x300);
GPIO1->DATA &= ~(3<<0);
mpl115a2_init_comm();

while(1)
{
    /* Deinit UART */
    SYSCON->SYSAHBCLKCTRL &= ~(1<<12);
    SYSCON->PCLKCTRL &= ~(1<<1); /* Enable the peripheral clock */
    SYSCON->UARTCLKDIV = 0x0; /* divided by 1 */

    SYSCON->STARTSRP0Clr |= (1<<(8));
    NVIC_ClearPendingIRQ(WAKEUP8_IRQn);
    NVIC_EnableIRQ(WAKEUP8_IRQn);
    /* Go to Sleep */
    __WFI();

    GPIO1->DATA |= (3<<0);
    for(i = 0; i < 0x30; ++i);
#ifdef ENABLE_TIMER
    /* Initialize 16-bit Timer 0 */
    Disable_Timer(0);
#endif /* ENABLE_TIMER */
    SYSCON->MAINCLKSEL = 3;
    SYSCON->MAINCLKUEN = 0;
    SYSCON->MAINCLKUEN = 1;
    while (!(SYSCON->MAINCLKUEN & 0x01));
    SYSCON->SYSAHBCLKDIV = 4;

    /* Reinit UART */
    SYSCON->SYSAHBCLKCTRL |= (1<<12);
    SYSCON->PCLKCTRL |= (1<<1); /* Enable the peripheral clock */
    SYSCON->UARTCLKDIV = 0x1; /* divided by 1 */

```

```

error=0;
s_connectionreset();
s_transstart();

error+=s_measure((unsigned char*) &temp_val.i,&checksum,TEMP); //measure temperature
error+=s_measure((unsigned char*) &humi_val.i,&checksum,HUMI); //measure humidity
dPressure = Compensate_Pressure();
GPIO1->DATA &= ~(3<<0);
temp_val.f=temp_val.i;           //converts integer to float
humi_val.f = humi_val.i;         //converts integer to float
calc_sth11(&humi_val.f,&temp_val.f); //calculate humidity, temperature, high resolution
sprintf(cTemp, "T%c%2.2f ", SENSOR_ID, temp_val.f);
sprintf(cHumi, "H%c%2.2f ", SENSOR_ID, humi_val.f);
f_Pressure = (dPressure*(65.0/ 1023.0))+50;
GPIO1->DATA &= ~(3<<0);
sprintf(cPres, "P%c%3.2f ", SENSOR_ID, f_Pressure);
sprintf(cBattStatus, "B%c%d ", SENSOR_ID, BattStatus);

if(error!=0)
{
    GPIO3->DATA &= ~(1<<5);
    for(i = 0; i < 0x3000; ++i);
    s_connectionreset();           //in case of an error: connection reset
    UARTSend(Error, 12);
    GPIO3->DATA |= (1<<5);
}
else
{
    GPIO1->DATA &= ~(1<<8);
    GPIO3->DATA &= ~(1<<5);
    for(i = 0; i < 0x4400; ++i);    //delay for the comm module
    UARTSend(cTemp, 8);
    UARTSend(cHumi, 8);
    UARTSend(cPres, 9);
    UARTSend(cBattStatus, 4);
    for(i = 0; i < 0x4400; ++i);    //delay for the comm module
    GPIO3->DATA |= (1<<5);
    GPIO1->DATA |= (1<<8);
}
SYSCON->MAINCLKSEL = 2;
SYSCON->MAINCLKUEN = 0;
SYSCON->MAINCLKUEN = 1;
while (!(SYSCON->MAINCLKUEN & 0x01));
SYSCON->SYSAHBCLKDIV = 1;
#ifdef ENABLE_TIMER
    /* Initialize 16-bit Timer 0 */
    Enable_Timer(0);
#endif /* ENABLE_TIMER */

```



```

    BattStatus = FULL_BAT;
    SYSCON->BODCTRL = 3 << 2; // VDD < 2.80V
}
}

void wait(uint32_t time){
    while(time--);
}

```

D.2 Código-fonte do nó coordenador

```

/*****
** Wireless Sensor Network – Coordinator node code source
**
** This project configures the microcontroller to be used within the Coordinator node.
** It configures the Ethernet to show a webpage when accessed through the IP (current is 192.168.0.100) .
** The SD card stores all data from all four nodes
** Timer 1 is configured to get a PWM signal from the comm module with signal strength information
** Based on Easyweb project from Andreas Dannenberg
*****/

#include <stdio.h>
#include <LPC23xx.h> // Keil: Register definition file for LPC2378
#include <string.h>

#define extern // Keil: Line added for modular project management

#include "type.h"
#include "easyweb.h"
#include "EMAC.h" // Keil: *.c->*.h // ethernet packet driver
#include "tcpip.h" // Keil: *.c->*.h // easyWEB TCP/IP stack
#include "rtc.h"
#include "LCD.h"
#include "irq.h"
#include "webpage.h" // website for our HTTP server (HTML)
#include "header.h"
#include "sdmmc.h"

extern void init_serial(void);
extern int getkey(void);

void ShowTime (void);
void PrintTime (void);
unsigned char cStamped = 0, cLevel = 0;
RTCTime local_time, current_time;
unsigned int temp[4] = {767}, str_enable = 18;
unsigned int sig_wr = 0, sig_rd = 0;
float strength = 100.0;
unsigned int pwm_count = 767;
unsigned char NewKey[116] = {0}, isLocked = 0;
volatile unsigned int k = 0, print = 0;

unsigned char * GetBattCode ( unsigned int Code )
{
    switch (Code)
    {

```

```

    case '4': return (unsigned char *)cGREEN;
    case '3': return (unsigned char *)cYELLOW;
    case '2': return (unsigned char *)cRED;
    case '1': return (unsigned char *)cORANGE;
    case '0': return (unsigned char *)cBLACK;
    default : return (unsigned char *)cGREEN;
  }
}

__irq void T1_IRQHandler (void)
{
  T1IR    = 0x1;
  isLocked = 1;
  temp[sig_wr] = T1CR0;
  isLocked = 0;

  ++sig_wr;
  if (4 <= sig_wr) sig_wr = 0;

  VICVectAddr = 0;          /* Acknowledge Interrupt      */
}

__irq void T3_IRQHandler (void)
{
  T3IR    = 0x1;          /* Clear interrupt flag      */
  ++str_enable;
  if (!(SocketStatus & SOCK_ACTIVE)) TCPPassiveOpen(); // listen for incoming TCP-connection
  DoNetworkStuff();      // handle network and easyWEB-stack
  HTTPServer();
  VICVectAddr = 0;          /* Acknowledge Interrupt      */
}

__irq void UART_IRQHandler (void)
{
  unsigned int escape_cnt = 0;
  FIO2SET = 0x00000080;
  while ((U1IIR & 0x04))
  {
    NewKey[k] = U1RBR;
    ++k;
  }
  FIO2CLR = 0x00000080;

  if (20 <= str_enable)
  {
    str_enable = 0;
    // Check if signal level is 100%
    while (FIO1PIN & 0x00040000) if (++escape_cnt >= 190) break;

    T1TCR = 2;
    // Waits for the positive edge
    while ((~FIO1PIN & 0x00040000));
    T1TCR = 1;
  }
  VICVectAddr = 0;          /* Acknowledge Interrupt      */
}

__irq void T2_IRQHandler (void)
{
  volatile unsigned int n = 0;
  ++print;
  T2IR = 0x1;

  for(n = 0; n < 116; ++n )
  {
    /* Dados vindos do Sensor 1 */
    if ((NewKey + (n+1)) == 'A' && !(cStamped & 0x1))
    {

```

```

    InsertStamp('A');
    cStamped |= 0x1;
    if (!(cLevel & 0x1))
    {
        if (!isLocked) pwm_count = temp[sig_rd];
        else pwm_count = 767;
        strength = ((pwm_count/767.0)*100.0);
        sprintf(XA, "%3.0f", strength);
        cLevel |= 0x1;
        ++sig_rd;
    }
}
if ((* (NewKey + n) == 'T') && (* (NewKey + (n+1)) == 'A'))
{
    FIO2SET = 0x00000001;
    memcpy(TA, NewKey+(n+2), 5);
    InsertTemp('A');
}
if ((* (NewKey + n) == 'H') && (* (NewKey + (n+1)) == 'A'))
{
    FIO2SET = 0x00000001;
    memcpy(HA, NewKey+(n+2), 5);
    InsertHumi('A');
}
if ((* (NewKey + n) == 'P') && (* (NewKey + (n+1)) == 'A'))
{
    FIO2SET = 0x00000001;
    memcpy(PA, NewKey+(n+2), 6);
    InsertPres('A');
}
if ((* (NewKey + n) == 'B') && (* (NewKey + (n+1)) == 'A'))
{
    FIO2SET = 0x00000001;
    memcpy (BA, (void *)GetBattCode(* (NewKey+(n+2))), 7);
    InsertBatt('A', (unsigned char *) (NewKey+(n+2)));
    InsertStrength('A');
}

/* Dados vindos do Sensor 2 */
if (( (* (NewKey + n) == 'T') || (* (NewKey + n) == 'H')
    || (* (NewKey + n) == 'P') || (* (NewKey + n) == 'B')) && (* (NewKey + (n+1)) == 'B') && !(cStamped & 0x2))
{
    InsertStamp('B');
    cStamped |= 0x2;
    if (!(cLevel & 0x2))
    {
        if (!isLocked) pwm_count = temp[sig_rd];
        else pwm_count = 767;
        strength = ((pwm_count/767.0)*100.0);
        sprintf(XB, "%3.0f", strength);
        cLevel |= 0x2;
        ++sig_rd;
    }
}
if ((* (NewKey + n) == 'T') && (* (NewKey + (n+1)) == 'B'))
{
    FIO2SET = 0x00000002;
    memcpy(TB, NewKey+(n+2), 5);
    InsertTemp('B');
}
if ((* (NewKey + n) == 'H') && (* (NewKey + (n+1)) == 'B'))
{
    FIO2SET = 0x00000002;
    memcpy(HB, NewKey+(n+2), 5);
    InsertHumi('B');
}

```

```

}
if ((* (NewKey + n) == 'P') && (* (NewKey + (n+1)) == 'B'))
{
    FIO2SET = 0x00000002;
    memcpy(PB, NewKey+(n+2), 6);
    InsertPres('B');
}
if ((* (NewKey + n) == 'B') && (* (NewKey + (n+1)) == 'B'))
{
    FIO2SET = 0x00000002;
    memcpy (BB, (void *)GetBattCode(* (NewKey+(n+2))), 7);
    InsertBatt('B', (unsigned char *) (NewKey+(n+2)));
    InsertStrength('B');
}

/* Dados vindos do Sensor 3 */
if ((* (NewKey + (n+1)) == 'C') && !(cStamped & 0x4))
{
    InsertStamp('C');
    cStamped |= 0x4;
    if (!(cLevel & 0x4))
    {
        if (!isLocked) pwm_count = temp[sig_rd];
        else pwm_count = 767;
        strength = ((pwm_count / 767.0) * 100.0);
        sprintf(XC, "%3.0f", strength);
        cLevel |= 0x4;
        ++sig_rd;
    }
}
if ((* (NewKey + n) == 'T') && (* (NewKey + (n+1)) == 'C'))
{
    FIO2SET = 0x00000004;
    memcpy(TC, NewKey+(n+2), 5);
    InsertTemp('C');
}
if ((* (NewKey + n) == 'H') && (* (NewKey + (n+1)) == 'C'))
{
    FIO2SET = 0x00000004;
    memcpy(HC, NewKey+(n+2), 5);
    InsertHumi('C');
}
if ((* (NewKey + n) == 'P') && (* (NewKey + (n+1)) == 'C'))
{
    FIO2SET = 0x00000004;
    memcpy(PC, NewKey+(n+2), 6);
    InsertPres('C');
}
if ((* (NewKey + n) == 'B') && (* (NewKey + (n+1)) == 'C'))
{
    FIO2SET = 0x00000004;
    memcpy (BC, (void *)GetBattCode(* (NewKey+(n+2))), 7);
    InsertBatt('C', (unsigned char *) (NewKey+(n+2)));
    InsertStrength('C');
}

/* Dados vindos do Sensor 4 */
if ((* (NewKey + (n+1)) == 'D') && !(cStamped & 0x8))
{
    InsertStamp('D');
    cStamped |= 0x8;
    if (!(cLevel & 0x8))
    {
        if (!isLocked) pwm_count = temp[sig_rd];
        else pwm_count = 767;
    }
}

```

```

    strength = ((pwm_count/767.0)*100.0);
    sprintf(XD, "%3.0f", strength);
    cLevel |= 0x8;
    ++sig_rd;
}
}
if ((* (NewKey + n) == 'T') && (* (NewKey + (n+1)) == 'D'))
{
    FIO2SET = 0x00000008;
    memcpy(TD, NewKey+(n+2), 5);
    InsertTemp('D');
}
if ((* (NewKey + n) == 'H') && (* (NewKey + (n+1)) == 'D'))
{
    FIO2SET = 0x00000008;
    memcpy(HD, NewKey+(n+2), 5);
    InsertHumi('D');
}
if ((* (NewKey + n) == 'P') && (* (NewKey + (n+1)) == 'D'))
{
    FIO2SET = 0x00000008;
    memcpy(PD, NewKey+(n+2), 6);
    InsertPres('D');
}
if ((* (NewKey + n) == 'B') && (* (NewKey + (n+1)) == 'D'))
{
    FIO2SET = 0x00000008;
    memcpy (BD, (void *)GetBattCode(* (NewKey+(n+2))), 7);
    InsertBatt('D', (unsigned char *) (NewKey+(n+2)));
    InsertStrength('D');
}
}

k = 0;
memset(NewKey, '?', 116);
cStamped = 0;
cLevel = 0;
sig_wr = 0;
sig_rd = 0;
VICVectAddr = 0;          /* Acknowledge Interrupt      */
if (0 == print%2)
{
    ShowTime();
}
if (~FIO2PIN & 0x400)
{
    RTCGetTime( &current_time);
    local_time = current_time;
    ++local_time.RTC_Sec;
    RTCSetTime( local_time );
}

}

//void main(void)
int main(void)
{
    SCS |= 0x1;
    RTCInit();

    local_time.RTC_Sec = 0;
    local_time.RTC_Min = 30;
    local_time.RTC_Hour = 23;
    local_time.RTC_Mday = 25;
    local_time.RTC_Wday = 6;

```

```

local_time.RTC_Yday = 237;
local_time.RTC_Mon = 8;
local_time.RTC_Year = 2012;
RTCGetTime( &current_time );
if (2012 != current_time.RTC_Year)
    RTCSetTime( local_time );           /* Set local time */
TCPLowLevelInit();
init_serial();

/* Enable and setup timer interrupt, start timer */
T1MR0 = 767;           /* 64usec = 768-1 at 12.0 MHz */
T1CCR = 2;             /* Interrupt and Falling Edge detection on CR0 */
T1MCR = 5;             /* Interrupt and Stop on MR0 */

/* Enable and setup timer interrupt, start timer */
PCONP |= (1 << 22);
T2MR0 = 11999;         /* 1msec = 12000-1 at 12.0 MHz */
T2MCR = 3;             /* Interrupt and Reset on MR0 */
T2PR = 499;

/* Enable and setup timer interrupt, start timer */
PCONP |= (1 << 23);
T3MR0 = 11999;         /* 1msec = 12000-1 at 12.0 MHz */
T3MCR = 3;             /* Interrupt and Reset on MR0 */

install_irq( TIMER1_INT, (void *)T1_IRQHandler, HIGHEST_PRIORITY + 2 );
install_irq( TIMER2_INT, (void *)T2_IRQHandler, HIGHEST_PRIORITY + 3 );
install_irq( TIMER3_INT, (void *)T3_IRQHandler, HIGHEST_PRIORITY );
install_irq( UART1_INT, (void *)UART_IRQHandler, HIGHEST_PRIORITY + 1 );

HTTPStatus = 0;           // clear HTTP-server's flag register

TCPLocalPort = TCP_PORT_HTTP; // set port we want to listen to

lcd_init();
lcd_clear();
set_cursor (0, 0);
lcd_print (" LSM Unicamp ");
set_cursor (0, 1);
lcd_print (" Sensor Network ");

RTCStart();

PrepareDisk();
PINSEL3 |= 0x00000030;
PINSEL4 &= ~0x000000FF;

FIO2DIR |= 0x000000FF;
FIO2CLR = 0x000000FF;
FIO2MASK = 0;
//T0TCR = 1;
T3TCR = 1;           /* Timer0 Enable */
T2TCR = 1;           /* Timer0 Enable */

while (1)           // repeat forever
{
}

// This function implements a very simple dynamic HTTP-server.
// It waits until connected, then sends a HTTP-header and the
// HTML-code stored in memory. Before sending, it replaces
// some special strings with dynamic values.
// NOTE: For strings crossing page boundaries, replacing will
// not work. In this case, simply add some extra lines
// (e.g. CR and LFs) to the HTML-code.

```

```

void HTTPServer(void)
{
    if (SocketStatus & SOCK_CONNECTED)        // check if somebody has connected to our TCP
    {
        if (SocketStatus & SOCK_DATA_AVAILABLE) // check if remote TCP sent data
            TCPReleaseRxBuffer();              // and throw it away

        if (SocketStatus & SOCK_TX_BUF_RELEASED) // check if buffer is free for TX
        {
            FIO2SET = 0x00000040;
            if (!(HTTPStatus & HTTP_SEND_PAGE)) // init byte-counter and pointer to webside
            {
                // if called the 1st time
                HTTPBytesToSend = sizeof(WebSide) - 1; // get HTML length, ignore trailing zero
                PWebSide = (unsigned char *)WebSide; // pointer to HTML-code
            }
            if (HTTPBytesToSend > MAX_TCP_TX_DATA_SIZE) // transmit a segment of MAX_SIZE
            {
                if (!(HTTPStatus & HTTP_SEND_PAGE)) // 1st time, include HTTP-header
                {
                    memcpy(TCP_TX_BUF, GetResponse, sizeof(GetResponse) - 1);
                    memcpy(TCP_TX_BUF + sizeof(GetResponse) - 1, PWebSide, MAX_TCP_TX_DATA_SIZE - sizeof(GetResponse) + 1);
                    HTTPBytesToSend -= MAX_TCP_TX_DATA_SIZE - sizeof(GetResponse) + 1;
                    PWebSide += MAX_TCP_TX_DATA_SIZE - sizeof(GetResponse) + 1;
                }
                else
                {
                    memcpy(TCP_TX_BUF, PWebSide, MAX_TCP_TX_DATA_SIZE);
                    HTTPBytesToSend -= MAX_TCP_TX_DATA_SIZE;
                    PWebSide += MAX_TCP_TX_DATA_SIZE;
                }
                TCPTxDataCount = MAX_TCP_TX_DATA_SIZE; // bytes to xfer
                PrintTime();
                InsertDynamicValues(); // exchange some strings...
                TCPTransmitTxBuffer(); // xfer buffer
            }
            else if (HTTPBytesToSend) // transmit leftover bytes
            {
                memcpy(TCP_TX_BUF, PWebSide, HTTPBytesToSend);
                TCPTxDataCount = HTTPBytesToSend; // bytes to xfer
                PrintTime();
                InsertDynamicValues(); // exchange some strings...
                TCPTransmitTxBuffer(); // send last segment
                TCPClose(); // and close connection
                HTTPBytesToSend = 0; // all data sent
            }

            HTTPStatus |= HTTP_SEND_PAGE; // ok, 1st loop executed
            FIO2CLR = 0x00000040;
        }
    }
    else
        HTTPStatus &= ~HTTP_SEND_PAGE; // reset help-flag if not connected
}

// searches the TX-buffer for special strings and replaces them
// with dynamic values (AD-converter results)

void InsertDynamicValues(void)
{
    unsigned char *Key;
    unsigned int i = 0;

    FIO2CLR = 0x0000003F;
    if (TCPTxDataCount < 4) return; // there can't be any special string

```

```

Key = TCP_TX_BUF;

for (i = 0; i < (TCPTxDataCount - 4); ++i)
{
    if (*Key == 'S')
    {
        if (*(Key + 1) == 'H')
        {
            if (*(Key + 2) == 'T')
            {
                if (*(Key + 3) == '1')
                {
                    switch (*(Key + 4))
                    {
                        // Temperature Sensor 1
                        case 't' : memcpy(Key, TA, 5); break;
                        // Humidity Sensor 1
                        case 'h' : memcpy(Key, HA, 5); break;
                        // Pressure Sensor 1
                        case 'p' : memcpy(Key, PA, 6); break;
                        // Stamp Sensor 1
                        case 's' : memcpy(Key, cSA, 21); break;
                    }
                }
            }
            else if (*(Key + 3) == '2')
            {
                switch (*(Key + 4))
                {
                    // Temperature Sensor 2
                    case 't' : memcpy(Key, TB, 5); break;
                    // Humidity Sensor 2
                    case 'h' : memcpy(Key, HB, 5); break;
                    // Pressure Sensor 2
                    case 'p' : memcpy(Key, PB, 6); break;
                    // Stamp Sensor 2
                    case 's' : memcpy(Key, cSB, 21); break;
                }
            }
            else if (*(Key + 3) == '3')
            {
                switch (*(Key + 4))
                {
                    // Temperature Sensor 3
                    case 't' : memcpy(Key, TC, 5); break;
                    // Humidity Sensor 3
                    case 'h' : memcpy(Key, HC, 5); break;
                    // Pressure Sensor 3
                    case 'p' : memcpy(Key, PC, 6); break;
                    // Stamp Sensor 3
                    case 's' : memcpy(Key, cSC, 21); break;
                }
            }
            else if (*(Key + 3) == '4')
            {
                switch (*(Key + 4))
                {
                    // Temperature Sensor 4
                    case 't' : memcpy(Key, TD, 5); break;
                    // Humidity Sensor 4
                    case 'h' : memcpy(Key, HD, 5); break;
                    // Pressure Sensor 4
                    case 'p' : memcpy(Key, PD, 6); break;
                    // Stamp Sensor 4
                    case 's' : memcpy(Key, cSD, 21); break;
                }
            }
        }
    }
    if (*Key == 'B')
    {
        switch (*(Key + 1))
        {
            // Battery Sensor 1
            case '1' : memcpy(Key, BA, 8); break;
            // Battery Sensor 2
            case '2' : memcpy(Key, BB, 8); break;
            // Battery Sensor 3
            case '3' : memcpy(Key, BC, 8); break;
            // Battery Sensor 4
            case '4' : memcpy(Key, BD, 8); break;
        }
    }
}

```



```

}
if (*Key == 'X')
if (* (Key + 1) == 'S')
switch (* (Key + 2))
{
// Signal Strength Sensor 1
case '1' : memcpy(Key, XA, 3); break;
// Signal Strength Sensor 2
case '2' : memcpy(Key, XB, 3); break;
// Signal Strength Sensor 3
case '3' : memcpy(Key, XC, 3); break;
// Signal Strength Sensor 4
case '4' : memcpy(Key, XD, 3); break;
}
++Key;
}
}

void PrintTime(void)
{
unsigned char *Key;
unsigned int i = 0;

unsigned char chHour[3] = {'1','5',0},
chMinute[3] = {'0','0',0},
chSecond[3] = {'0','0',0},
chDay[3] = {'0','0',0},
chMonth[3] = {'0','0',0},
chYear[5] = {'0','0','0','0',0};

RTCGetsTime( &current_time );

//Charing Day
chDay[0] = 0x30 + current_time.RTC_Mday/ 10;
chDay[1] = 0x30 + current_time.RTC_Mday%10;
//Charing Month
chMonth[0] = 0x30 + current_time.RTC_Mon/ 10;
chMonth[1] = 0x30 + current_time.RTC_Mon%10;
//Charing Second
chYear[0] = 0x30 + current_time.RTC_Year/ 1000;
chYear[1] = 0x30 + ( current_time.RTC_Year%1000)/ 100;
chYear[2] = 0x30 + ( current_time.RTC_Year%1000)%100/ 10;
chYear[3] = 0x30 + ( current_time.RTC_Year%1000)%100%10;
//Charing Hour
chHour[0] = 0x30 + current_time.RTC_Hour/ 10;
chHour[1] = 0x30 + current_time.RTC_Hour%10;
//Charing Minute
chMinute[0] = 0x30 + current_time.RTC_Min/ 10;
chMinute[1] = 0x30 + current_time.RTC_Min%10;
//Charing Second
chSecond[0] = 0x30 + current_time.RTC_Sec/ 10;
chSecond[1] = 0x30 + current_time.RTC_Sec%10;

if (TCPTxDataCount < 4) return; // there can't be any special string

Key = TCP_TX_BUF;

for (i = 0; i < (TCPTxDataCount - 4); ++i)
{
if ( (*Key == 'D') && (* (Key + 1) == 'D') ) memcpy(Key, chDay, 2);

if ( (*Key == 'E') && (* (Key + 1) == 'E') ) memcpy(Key, chMonth, 2);

if ( (*Key == 'Y') && (* (Key + 1) == 'Y') && (* (Key + 2) == 'Y') && (* (Key + 3) == 'Y') ) memcpy(Key, chYear, 4);

if ( (*Key == 'H') && (* (Key + 1) == 'H') ) memcpy(Key, chHour, 2);

```

```

if ( (*Key == 'M') && (* (Key + 1) == 'M') ) memcpy(Key, chMinute, 2);

if ( (*Key == 'S') && (* (Key + 1) == 'S') ) memcpy(Key, chSecond, 2);

++Key;
}
}

```

D.3 Código-fonte da página *web* embarcada

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML><HEAD><TITLE> LSM UNICAMP - Demonstração Marcel Salvioni </TITLE>
<META content="text/html; charset=windows-1252" http-equiv=Content-Type>
<META content=30 http-equiv=refresh>
<META name=GENERATOR content="MSHTML 8.00.6001.18865"></HEAD>
<BODY background="" bgColor=#ffffff text=#000000><B><FONT color=#0000ff size=7
face="Britannic Bold">
<CENTER>LSM</CENTER></FONT></B>
<P></P>
<P><B><FONT color=#0000ff>
<CENTER>Laboratório de Sensores Microeletrônicos - Unicamp</FONT></CENTER>
<P></P></B>
<P><B>Aluno: Marcel Salvioni da Silva</B></P>
<P><B>Orientador: Prof. Dr. Fabiano Fruett</B></P>
<P><B>Rede de Sensores Sem Fio de Baixo Custo para Monitoramento Ambiental.</B></P>
<P>Este trabalho apresenta uma Rede de Sensores sem fio projetada para o monitoramento de temperatura, Umidade Relativa
e Pressão Atmosférica na Floresta Amazônica. Um microcontrolador de 32 bits e um módulo transceptor operando na frequência
ISM são utilizados para compor os nós da rede, proporcionando otimização de energia, enquanto vários elementos sensores são usados
para realizar as medições, como um sensor de temperatura e de umidade com interface digital da Sensirion e um sensor de pressão atmosférica
com interface I2C da Freescale. Os valores das medidas são acessados por uma página da Web incorporada em um microcontrolador e fornecido
através de um endereço IP, que mostra os dados aquisitados em tempo real.</P>
<P><B>Data: DD/EE/YYYY HH:MM:SS</B></P><B>
<P>
<TABLE border=1 cellSpacing=0 cellPadding=0 align=center>
<TBODY>
<TR>
<TD>
<P align=center><STRONG><FONT size=5>Sensor</FONT></STRONG></P></TD>
<TD>
<P align=center><STRONG><FONT size=5>Temperatura</FONT></STRONG></P></TD>
<TD>
<P align=center><STRONG><FONT size=5>Umidade</FONT></STRONG></P></TD>
<TD>
<P align=center><STRONG><FONT size=5>Pressão</FONT></STRONG></P></TD>
<TD>
<P align=center><STRONG><FONT size=5>Bat</FONT></STRONG></P></TD>
<TD>
<P align=center><STRONG><FONT size=5>Sinal</FONT></STRONG></P></TD>
<TD>
<P align=center><STRONG><FONT size=5>Último dado recebido</FONT></STRONG></P></TD></TR>
<TR>
<TD>
<P align=center><STRONG><FONT size=5>1</FONT></STRONG></P></TD>
<TD>
<P align=center><STRONG><FONT size=5><FONT color=#ff0000>SHT1t</FONT> °C</FONT></STRONG></P></TD>
<TD>
<P align=center><STRONG><FONT size=5><FONT color=#ff0000>SHT1h</FONT> %</FONT></STRONG></P></TD>
<TD>
<P align=center><STRONG><FONT size=5><FONT color=#ff0000>SHT1pr</FONT> kPa</FONT></STRONG></P></TD>
<TD BGColor=B1>
<P align=center>&nbsp;</P></TD>
<TD>

```

[illegible]

