

**UNIVERSIDADE FEDERAL DE MATO GROSSO
INSTITUTO DE FÍSICA
PROGRAMA DE PÓS-GRADUAÇÃO EM FÍSICA AMBIENTAL**

**IMPLEMENTAÇÃO POR SISTEMAS EMBARCADOS
DE PROTÓTIPO PARAMETRIZÁVEL PARA COLETA
DE DADOS MICROCLIMÁTICOS
GEORREFERENCIADOS**

ROBERTO BENEDITO DE OLIVEIRA PEREIRA

PROF. DR. CARLO RALPH DE MUSIS

**Cuiabá, MT
Dezembro 2013**

**UNIVERSIDADE FEDERAL DE MATO GROSSO
INSTITUTO DE FÍSICA
PROGRAMA DE PÓS-GRADUAÇÃO EM FÍSICA AMBIENTAL**

**IMPLEMENTAÇÃO POR SISTEMAS EMBARCADOS
DE PROTÓTIPO PARAMETRIZÁVEL PARA COLETA
DE DADOS MICROCLIMÁTICOS
GEORREFERENCIADOS**

ROBERTO BENEDITO DE OLIVEIRA PEREIRA

Tese apresentada ao Programa de Pós-Graduação em Física Ambiental da Universidade Federal de Mato Grosso, como parte dos requisitos para obtenção do título de Doutor em Física Ambiental.

PROF. DR. CARLO RALPH DE MUSIS

**Cuiabá, MT
Dezembro 2013**

Dados Internacionais de Catalogação na Fonte.

O48i Oliveira Pereira, Roberto Benedito de.
Implementação por sistemas embarcados de protótipo parametrizável para coleta
de dados microclimáticos georreferenciados / Roberto Benedito de Oliveira Pereira. --
2013
157 f. : il. color. ; 30 cm.

Orientador: Carlo Ralph De Musis.
Tese (doutorado) - Universidade Federal de Mato Grosso, Instituto de Física,
Programa de Pós-Graduação em Física Ambiental, Cuiabá, 2013.
Inclui bibliografia.

1. Arduino. 2. Sensores. 3. Datalogger. 4. Embarcados. I. Título.

Ficha catalográfica elaborada automaticamente de acordo com os dados fornecidos pelo(a) autor(a).

Permitida a reprodução parcial ou total, desde que citada a fonte.

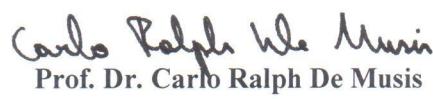
**UNIVERSIDADE FEDERAL DE MATO GROSSO
INSTITUTO DE FÍSICA
Programa de Pós-Graduação em Física Ambiental**

FOLHA DE APROVAÇÃO

**TÍTULO: IMPLEMENTAÇÃO POR SISTEMAS EMBARCADOS DE
PROTÓTIPO PARAMETRIZÁVEL PARA COLETA DE
DADOS MICROCLIMÁTICOS GEORREFERENCIADOS**

AUTOR: ROBERTO BENEDITO DE OLIVEIRA PEREIRA

Tese de Doutorado defendida e aprovada em 13 de dezembro de 2013, pela comissão julgadora:


Prof. Dr. Carlo Ralph De Musis
Orientador
Universidade de Cuiabá - UNIC


Prof. Dr. Denilton Carlos Gaio
Examinador Interno
Instituto de Física – UFMT


Prof. Dr. Josiel Maimone de Figueiredo –
Examinador Interno
Instituto de Computação – UFMT


Prof. Dr. Higo José Dalmagro
Examinador Externo
Universidade de Cuiabá - UNIC


Prof. Dr. Jó Ueyama – Examinador Externo
Instituto de Ciências Matemáticas e de Computação - USP

DEDICATÓRIA

A Deus, a meus pais, esposa, amigos e
companheiros de todas as horas.

AGRADECIMENTOS

A Deus.

Ao Prof. Dr. Carlo Ralph de Masis, pela orientação concedida, pelo incentivo, empenho, dedicação, paciência e pela amizade, fatores que contribuíram e incentivaram a produção deste material.

Ao Prof. Dr. José de Souza Nogueira (Paraná), pelo apoio, incentivo e confiança a mim depositados e por ser um grande amigo.

A meu pai, Joaquim Benedito Lopes Pereira, a minha mãe Wânia Guilherma de Oliveira Pereira, e a minha esposa, Tatiana Michelli de Lima, aos quais dedico este trabalho.

A todos os professores do Programa de Mestrado e Doutorado em Física Ambiental da Universidade Federal de Mato Grosso, onde transmitiram seus conhecimentos.

Aos técnicos, pela dedicação, ajuda e orientação na realização dos ensaios e empréstimos dos laboratórios.

A Soilce e Cesário, pela colaboração nos serviços da secretaria, o que nos auxiliou para o desenvolvimento deste trabalho.

A todos os colegas de curso que ajudaram em muito, os quais se tornaram grandes amigos.

EPÍGRAFE

*“Nossos atos e vontades refletem no
que somos hoje e no que seremos
amanhã.”*

(Roberto Benedito)

SUMÁRIO

LISTA DE FIGURAS	IV
LISTA DE TABELAS	VIII
LISTA DE ABREVIATURAS E SÍMBOLOS	IX
RESUMO	XII
ABSTRACT.....	XIII
1. INTRODUÇÃO	1
1.1. PROBLEMÁTICA.....	1
1.2. JUSTIFICATIVA.....	2
2. REVISÃO BIBLIOGRÁFICA.....	5
2.1. CLIMA URBANO	5
2.2. TRANSECTOS MÓVEIS	7
2.3. SENsoRES.....	9
2.4. SISTEMA DE AQUISIÇÃO E REGISTRO DOS DADOS	13
2.5. SISTEMAS EMBARCADOS.....	19
2.5.1. A Estrutura do sistema embarcado comparado a um computador	21
2.5.2. Princípio de funcionamento de um processador	23
2.5.3. Placas para desenvolvimento e prototipação de embarcados.....	26
2.5.3.1. <i>Arduino</i>	27
2.5.3.2. <i>Raspberry Pi – Modelo B</i>	32
2.5.3.3 <i>BeagleBone Black</i>	33
2.5.4. Comparativo entre as placas para desenvolvimento e prototipação de embarcados.....	35
3. MATERIAIS E MÉTODOS.....	36
3.1. CARACTERÍSTICAS GERAIS DO LOCAL DO EXPERIMENTO	37
3.2. APRESENTAÇÃO DO TRAJETO UTILIZADO NO TRANSECTOR	38
3.3. SELEÇÃO DE HORÁRIO E DIA TÍPICO	39
3.4. ZOILOGGER	40
3.4.1. <i>Hardware</i>	43
3.4.1.1 Levantamento de requisitos.....	43
3.4.1.2. Componentes eletrônicos selecionados	45
3.4.1.3. Placa principal – <i>Arduino</i> adaptada.....	53
3.4.1.4. Shield do <i>Cartão SD</i> e <i>Realtime</i>	55

3.4.1.5. Shield de comunicação e barramento de conexões	57
3.4.1.6. Estrutura de suporte dos sensores	58
3.4.1.7. Suporte e Abrigo do sensor de temperatura e umidade relativa.....	61
3.4.1.8. Suporte e Abrigo do GPS	63
3.4.1.9. Alimentação	65
3.4.2. <i>Software</i>	66
3.4.2.1 Levantamento de requisitos.....	66
3.4.2.2. Diagrama de Caso de Uso	68
3.4.2.3. Diagrama de Atividade	69
3.4.2.4. Estrutura do arquivo de configuração	71
3.4.2.5. Estrutura de armazenamento dos arquivos no cartão	72
3.5. CONFECÇÃO DAS PLACAS E MONTAGEM	73
3.6. CALIBRAÇÃO E ESTATÍSTICA	74
3.7. MONITORAMENTO DO SISTEMA E AFERIÇÃO DO CONSUMO DE ENERGIA	75
4. RESULTADOS E DISCUSSÃO	77
4.1. CONFECÇÃO DAS PLACAS E MONTAGEM DO EQUIPAMENTO	77
4.1.1. Da placa do <i>Arduino</i> adaptada	77
4.1.2 <i>Shield</i> do Cartão <i>SD</i> e <i>Realtime</i>	78
4.1.3 <i>Shield</i> de comunicação e barramento de conexões	78
4.1.4 Estrutura de suporte dos sensores	79
4.1.5 Placa de suporte ao sensor e seu abrigo térmico	80
4.1.6 Suporte e Abrigo do GPS	80
4.1.7 Estrutura completa montada.....	81
4.2. AVALIAÇÃO DO SISTEMA	81
4.2.1. Autonomia de energia	81
4.2.2. Recursos computacionais consumidos	83
4.2.3. Monitoramento do sistema.....	83
4.2.4 Monitoramento em tempo real	84
4.3. ESTATÍSTICA E VALIDAÇÃO DO SISTEMA POR MEIO DE UM TRANSECTO MÓVEL NO CAMPUS DA UFMT	85
4.3.1. Semivariografia	87
4.3.2. Krigeagem e Estudo microclimático do transecto	88
5. CONSIDERAÇÕES FINAIS	108
5.1. TRABALHOS FUTUROS	110
REFERÊNCIAS BIBLIOGRÁFICAS	112

BIBLIOGRAFIAS CITADAS	112
BIBLIOGRAFIAS CONSULTADAS	117
ANEXOS	118
Anexo A - Código do ZOIOLOGGER	118
Anexo B – PARTE DO Manual do protocol NMEA.....	124
Anexo C – A história do <i>Arduino</i>	134

LISTA DE FIGURAS

Figura 01 – Diagrama funcional de uma conversão n-bit por aproximação sucessiva.	16
.....
Figura 02 – Diagrama Funcional de uma conversão n-bit de um conversor flash A/D.....	18
.....
Figura 03 – As camadas de <i>Software</i>	22
.....
Figura 04 – Arquitetura de um computador genérico.....	24
.....
Figura 05 – Arquitetura de um micro controlador genérico.	25
.....
Figura 06 – Placa do <i>Arduino</i>	28
.....
Figura 07 – Diagrama de bloco do AVR ATmega328.	30
.....
Figura 08 – Placa do Raspberry Pi – Modelo B.....	33
.....
Figura 09 – Placa do BeagleBoard Black.	34
.....
Figura 10 – Universidade Federal de Mato Grosso - campus Cuiabá.	38
.....
Figura 11 – Trajeto do transecto móvel na UFMT.	39
.....
Figura 12 – Diagrama esquemático do Zoiologger.....	41
.....
Figura 13 - Arquitetura de comunicação do Zoiologger.....	42
.....
Figura 14 – Microcontrolador ATmega328P-PU – encapsulamento DIP	45
.....
Figura 15 – Sensor de temperatura e umidade relativa AM2302 / DHT22	46
.....
Figura 16 – GPS - SKM53.....	48
.....
Figura 17 – RealTime - DS1307.....	50
.....
Figura 18 – Regulador de tensão LM7805 e 78L33.	51
.....
Figura 19 – Diagrama de utilização padrão dos reguladores de tensão.	51
.....
Figura 20 – Módulo de comunicação <i>bluetooth</i> JY-MCU.....	52
.....
Figura 21 – Diagrama elétrico da placa do <i>Arduino</i> adaptada.....	53
.....
Figura 22 – <i>Layout</i> da placa do <i>Arduino</i> adaptada.....	54
.....
Figura 23 – Diagrama elétrico da <i>Shield</i> do Cartão <i>SD</i> , <i>Realtime</i> e Divisor de tensão para medir a tensão de alimentação.	55
.....
Figura 24 – <i>Shield</i> do cartão <i>SD</i> , <i>Realtime</i> e o divisor de tensão.	56
.....
Figura 25 – Diagrama elétrico da <i>shield</i> de conexão <i>nrf</i> e <i>bluetooth</i>	57
.....
Figura 26 – <i>Layout</i> da <i>shield</i> de conexão <i>nrf</i> e <i>bluetooth</i>	58
.....

Figura 27 – Visão lateral da estrutura de suporte dos sensores.....	59
Figura 28 – Visão superior da estrutura que auxilia na estabilidade e condução do suporte.....	60
Figura 29 – Mochila utilizada como abrigo para os equipamentos e suporte para a estrutura de canos.....	61
Figura 30 – Diagrama elétrico do suporte do sensor DHT22	62
Figura 31 – <i>Layout</i> do suporte do sensor DHT22	63
Figura 32 – Diagrama elétrico do suporte do GPS.	64
Figura 33 – Layout do suporte do GPS.....	65
Figura 34 – A bateria e a conexão da bateria por meio do conector P4.....	66
Figura 35 – Diagrama de Caso de Uso do Zoiologger segundo a notação UML da Astah Community	69
Figura 36 – Diagrama de atividade do Zoiologger segundo a notação UML da Astah Community.	70
Figura 37 – Exemplo do arquivo de configuração config.txt	71
Figura 38 – Medidor de Estresse Térmico TGD-300	74
Figura 39 – Estação micro meteorológica TP1080WC	75
Figura 40 – Placa do Arduino adaptado.....	77
Figura 41 – <i>Shield</i> do cartão SD, <i>Realtime</i> e divisor de tensão.	78
Figura 42 – <i>Shield</i> de conexão nrf e bluetooth.	79
Figura 43 – Estrutura de suporte: (a) visão do suporte de estabilização, (b) visão lateral da estrutura de suporte dos sensores montada.	79
Figura 44 – Suporte do sensor: (a) Placa de suporte ao sensor de temperatura e umidade relativa do ar, (b) Abrigo térmico.	80
Figura 45 – Suporte ao módulo do GPS: (a) Placa de suporte, (b) Abrigo do GPS .	80
Figura 46 – Estrutura de suporte aos sensores e o equipamento de coleta de dados	81
Figura 47 – Aferição do consumo de energia sem o bluetooth: (a) Sistema em modo de espera, (b) Sistema gravando dados no cartão SD.	82
Figura 48 – Aferição do consumo de energia com bluetooth: (a) Sistema em modo de espera, (b) Sistema gravando dados no cartão SD e transmitindo pelo bluetooth	82
Figura 49 – Mensagem do sistema informando a ausência do cartão de memória...	84

Figura 50 – Mensagem do sistema informando a ausência do arquivo de configuração.....	84
Figura 51 – Tela do <i>software Blueterm</i> monitorando uma coleta de dados.....	85
Figura 52 – Trajeto do transecto móvel na UFMT	86
Figura 53 – Semivariograma isotrópico da temperatura do ar.....	87
Figura 54 – Semivariograma isotrópico da umidade relativa do ar	87
Figura 55 – Mapa de isolinhas da temperatura e umidade relativa do ar nos trajetos.	
.....	88
Figura 56 – Foto do início do primeiro trecho: (a) Visão inicial do percurso, (b) Visão do bosque ao lado do ponto inicial.....	91
Figura 57 – Visão do céu no ponto inicial.	92
Figura 58 – Foto do início do segundo trecho: (a) Portal de entrada do Zoológico, (b) Pista de caminhada do Zoológico.	92
Figura 59 – Visão de parte do céu no segundo trecho	93
Figura 60 – Foto do início do terceiro trecho: (a) Pista que contorna a oca do índio, (b) Visão da pista atrás do R.U.....	93
Figura 61 – Visão parcial da cobertura na parte de trás do restaurante universitário.	
.....	94
Figura 62 – Foto do quarto trecho: (a) Visão parcial do bosque na frente do ginásio, (b) Visão dos fundos do ginásio indo em direção ao ponto de ônibus. 94	
Figura 63 – Fotos da cobertura de parte do quarto trecho.	95
Figura 64 – Foto da parte inicial do quinto trecho: (a) Bosque atrás do ponto de ônibus do Ginásio, (b) Campo de Futebol fechado para reforma.....	95
Figura 65 – Visão de parte do céu no quinto trecho.	96
Figura 66 – Foto da parte inicial do sexto trecho: (a) Pista de paralelepípedos ao lado da quadra de esportes, (b) Bosque na frente da guarita e ao lado da pista.....	96
Figura 67 – Visão de parte do céu no sexto trecho.	97
Figura 68 – Bosque de caminhada da UFMT: (a) Trecho inicial do bosque com uma lacuna Arbórea, (b) Trecho do meio ao fim do trecho de caminhada. .	97
Figura 69 – Visão de parte do céu no sétimo trecho.....	98

Figura 70 – Trecho de caminhada na avenida Fernando Corrêa da Costa: (a) Trecho inicial no final do bosque (b) Parte do meio do oitavo trecho de caminhada.....	98
Figura 71 – Visão de parte do céu no oitavo trecho.....	99
Figura 72 – Trecho da avenida de acesso a UFMT: (a) Calçada inicial da avenida de acesso à UFMT, (b) Trecho do meio da pista de caminhada.....	99
Figura 73 – Visão de parte do céu no nono trecho.....	100
Figura 74 – Avenida principal de acesso à UFMT após a guarita: (a) Calçada inicial da avenida de acesso à UFMT, (b) Bosque na entrada da UFMT.....	100
Figura 75 – Visão de parte do céu no décimo trecho.....	101
Figura 76 – Avenida principal da UFMT: (a) Calçada do inicial do trajeto, (b) Trecho intermediário do percurso.....	101
Figura 77 – Visão de parte do céu no décimo primeiro trecho.....	102
Figura 78 – Avenida principal da UFMT: (a) Calçada do inicial do trajeto, (b) Trecho intermediário do percurso próximo ao Ginásio.....	102
Figura 79 – Visão de parte do céu no décimo segundo trecho.....	103
Figura 80 – Avenida principal da UFMT: (a) Calçada do inicial do trajeto, (b) Trecho intermediário do percurso na frente do Restaurante Universitário e do estacionamento.....	103
Figura 81 – Visão de parte do céu no décimo terceiro trecho.....	104
Figura 82 – Avenida principal da UFMT: (a) Calçada do inicial do trajeto, (b) Trecho na frente do Restaurante Universitário e do estacionamento.	104
Figura 83 – Visão de parte do céu no décimo quarto trecho.....	105
Figura 84 – Trecho final do percurso: (a) Visão da rotatória em frente ao Bloco da Física Ambiental, (b) Trecho do percurso final.....	105
Figura 85 – Visão de parte do céu no décimo quarto trecho.....	106

LISTA DE TABELAS

Tabela 01 – Comparativo entre as placas de prototipação de embarcados	35
Tabela 02 – Requisitos funcionais de <i>hardware</i>	44
Tabela 03 – Características do ATmega328.	46
Tabela 04 – Características do sensor de temperatura e umidade relativa do ar - AM2303 / DHT22.	47
Tabela 05 – GPS SKM53 UART.	48
Tabela 06 – Parâmetros do Real Time DS1307.	50
Tabela 07 - Características dos reguladores de tensão LM7805 e 78L03	51
Tabela 08 – Características do módulo JY-MCU.	52
Tabela 09 – Requisitos funcionais do <i>software</i>	67
Tabela 10 – Requisitos não funcionais do <i>software</i>	68
Tabela 11 – Lista de comandos suportados no arquivo config.txt.....	72
Tabela 12 - Média das temperaturas e umidade relativa do ar nos trechos.....	107

LISTA DE ABREVIATURAS E SÍMBOLOS

μ V	micro-volts
A/D	Conversor Analógico-Digital
ARM	<i>Advanced RISC Machine</i> (Máquina Risc Avançada)
BPS	<i>Bits Per Seconds</i> (Bits Por Segundo)
CAD	<i>Computer Aided Design</i> (Desenho Assistido por Computador)
CD	<i>Compact Disc</i> (Disco Compacto)
CPU	<i>Central Processing Unit</i> (Unidade Central de Processamento)
CSV	<i>Comma-separated values</i> (Valores Separados por Vírgula)
DVD	<i>Digital Versatile Disc</i> (Disco Digital Versátil)
DHT	<i>Digital-output relative Humidity & Temperature sensor</i> (Sensor com saída Digital de Temperatura e Umidade relativa do ar)
EEPROM	<i>Electrically-Erasable Programmable Read-Only Memory</i> (Memória somente de Leitura Programável Eletronicamente)
FPGA	<i>Field Programmable Gate Array</i> (Arranjo de Portas Programáveis em Campo)
GMT	<i>Greenwich Mean Time</i> (Hora Média de Greenwich)
GPIO	General Purpose Input/Output (Portas Programáveis de Entrada e Saída)
GPL	<i>General Public License</i> (Licença Pública Geral)
GPS	<i>Global Positioning System</i> (Sistema de Posicionamento Global)
GPU	<i>Graphics Processing Unit</i> (Unidade de Processamento Gráfico)
HDMI	<i>High-Definition Multimedia Interface</i> (Interface Multimídia de Alta-Definição)
I/O	<i>Input/Output</i> (Entrada/Saída)
I2C	<i>Inter-Integrated Circuit</i> (Circuito Inter-Integrado)
IDE	<i>Integrated Development Environment</i> (Ambiente Integrado de Desenvolvimento)
IDII	<i>Interaction Design Institute Ivrea</i> (Instituto de Design Interativo de Ivrea)
LCD	<i>Liquid Crystal Display</i> (Display de Cristal Líquido)

LSB	Bit Menos Significativo
mA	mili-Amperes
MHz	mega-hertz
MIT	<i>Massachusetts Institute of Technology</i> (Instituto de Tecnologia de Massachusetts)
MP3	MPEG-1/2 Audio Layer 3
MSB	Bit Mais Significativo
mV	mili-Volts
NRF	<i>Nordic Radio Frequency</i> (Radio Frequência da Nordic)
Opcodes	<i>Operation Codes</i> (Código de Operação)
OTP	<i>One Time Programmable</i> (Uma Vez Programável)
PCB	<i>Printed Circuit Board</i> (Placa de Circuito Impresso)
PDIP	<i>Dual In-Line Package</i> (Encapsulamento Duplo em Linha)
PWM	<i>Pulse-Width Modulation</i> (Modulação por Largura de Pulso)
QFN	<i>Quad Flat No lead</i> (Encapsulamento Quadrado Plano sem terminais projetados)
RAM	<i>Random Access Memory</i> (Memória de Acesso Aleatório)
RISC	<i>Reduced Instruction Set Computer</i> (Computador com um Conjunto Reduzido de Instruções)
ROM	<i>Read-only Memory</i> (Memória Somente Leitura)
RTC	<i>Real Time Clock</i> (Relógio de Tempo Real)
SAD-69	<i>South American Datum</i> (Datum da América do Sul)
SD	<i>Secure Digital</i> (Segurança Digital)
SPI	<i>Serial Peripheral Interface</i> (Interface de Periférico Serial)
SoC	<i>System-on-a-chip</i> (Sistema em um Chip)
SRAM	<i>Static Random Access Memory</i> (Memória Estática de Acesso Aleatório)
S. O.	Sistema Operacional
TQFP	<i>Thin Quad Flat Package</i> (Encapsulamento Quadrado Fino)
TTL	<i>Transistor-Transistor Logic</i> (Lógica Transistor-Transistor)
UART	<i>Universal Asynchronous Receiver/Transmitter</i> (Recebimento/Envio Assíncrono Universal)
UFMT	Universidade Federal do Mato Grosso

USB	<i>Universal Serial Bus</i> (Barramento Serial Universal)
UML	<i>Unified Modeling Language</i> (Linguagem de Modelagem não proprietária)
UTM	<i>Universal Transverse Mercator</i> (Universal Transversa de Mercator)
WMO	<i>World Meteorological Organization</i> (Organização Meteorológica Mundial)

RESUMO

PEREIRA, R. B. O. IMPLEMENTAÇÃO POR SISTEMAS EMBARCADOS DE PROTÓTIPO PARAMETRIZÁVEL PARA COLETA DE DADOS MICROCLIMÁTICOS GEORREFERENCIADOS. Tese (Doutorado em Física Ambiental) - Instituto de Física. Universidade Federal de Mato Grosso. Cuiabá: UFMT, 2013. 157 f.

Esta pesquisa investiga a utilização de sistema embarcado para coleta e armazenamento de dados microclimáticos. O objetivo geral deste trabalho foi o desenvolvimento de um conjunto de *hardware* e *software* para a coleta, monitoramento e armazenamento de transectos micrometeorológicos. O lócus da pesquisa foi na Universidade Federal de Mato Grosso (UFMT). No que diz respeito à delimitação de espaço/tempo, esta pesquisa ocorreu em março de 2013. No teste de usabilidade do sistema de aquisição de dados realizou-se um estudo de caso na área de conforto ambiental na pista de passeio da UFMT e, para isso, foi realizada uma calibração do equipamento por meio de uma análise de variância pareada e depois se coletou a temperatura e umidade relativa do ar em três alturas, e as coordenadas geográficas nesse percurso. Após a coleta e a análise dos dados, realizaram-se as devidas correções nos valores obtidos por cada sensor e em seguida, gerou-se um semivariograma isotrópico da temperatura e da umidade relativa do ar, ambos resultaram em um modelo exponencial com coeficiente de determinação de 76.6% ($Co = 0.2220$; $Co + C = 1.1360$; $Ao = 612.00$; $r^2 = 0.766$; $RSS = 0.133$) e 78.4% ($Co = 1.2900$; $Co + C = 5.6180$; $Ao = 347.00$; $r^2 = 0.784$; $RSS = 2.25$) respectivamente. Gerou-se também um mapa de isolinhas por meio da krigagem dos dados. Durante o teste de autonomia de energia, o kit montado sem o módulo *bluetooth*, obteve-se uma média de consumo em modo espera de 57.45 mA e em modo gravação de 74.87 mA, com o módulo *bluetooth* os valores foram respectivamente 81.27 mA e 106.98 mA. O peso total do equipamento ficou em aproximadamente 2.1 Kg. A utilização do cartão de memória para configurar o equipamento e para recuperar os dados armazenados mostrou-se eficiente. Com base nos resultados pode-se confirmar a eficiência do equipamento, da metodologia utilizada e do padrão *Arduino* para a elaboração de protótipos.

Palavras chave: *Arduino*; Sensores; *Datalogger*; Embarcados.

ABSTRACT

PEREIRA, R. B. O. IMPLEMENTATION IN CONFIGURABLE EMBEDDED SYSTEMS PROTOTYPE FOR DATA COLLECTION MICROCLIMATIC GEOREFERENCED. Thesis (Doctor in Physics and Environment) - Institute of Physics, Federal University of Mato Grosso. Cuiabá: UFMT, 2013. 157 P.

This research investigates the use of embedded systems for collecting and storing data microclimate. The aim of this study was to develop a set of hardware and software for the collection, storage and monitoring transects micrometeorologics. The locus of the research was the Federal University of Mato Grosso (UFMT). Regarding the delimitation of space/time, this research took place in March 2013. In usability testing of the data acquisition system performed a case study in the area of environmental comfort on the trail walking UFMT for this, we performed a calibration of the equipment through a paired analysis of variance and then we collected the temperature and relative humidity in three heights, and the geographic coordinates that route. After collecting and analyzing data , there were the necessary corrections in the values obtained by each sensor and then generated a semivariogram isotropic temperature and relative humidity , both resulted in an exponential model with a coefficient of determination 76.6 % ($Co = 0.2220$, $C = 1.1360 + Co$, $Ao = 612.00$; $r^2 = 0.766$; $RSS = 0.133$) and 78.4 % ($Co = 1.2900$, $C = 5.6180 + Co$, $Ao = 347.00$; $r^2 = 0.784$; $RSS = 2.25$), respectively. Generated was also a contoured map by kriging data. During testing of autonomous power, the kit assembled without the bluetooth module, we obtained an average consumption in the standby mode is 57.45 mA and 74.87 mA for recording, with bluetooth module values were respectively 81.27 mA and 106.98 mA. The use of the memory card to set up the equipment and to retrieve stored data proved efficient. Based on the results, we confirm the efficiency of the equipment, the methodology used and the standard Arduino for prototyping.

Keywords: Arduino; Sensors; Datalogger; Embedded.

1. INTRODUÇÃO

1.1. PROBLEMÁTICA

A coleta de dados microclimáticos se tornou um dos recursos utilizados por pesquisadores para tentar entender fenômenos e eventos atmosféricos, a fim de validar suas teses ou hipóteses.

Atualmente, para a realização da coleta de dados dessas informações, utilizam-se equipamentos importados das mais diversas marcas e modelos, o que resulta, na maioria das vezes, no pagamento de royalties e na utilização apenas de sensores e peças exclusivas daquela marca. Restringindo, assim, a utilização de sensores de outros fabricantes.

Além disso, para sua compra, torna-se necessário passar por todo um processo de importação e, no caso de empresas públicas, por processo de licitação. Todas essas etapas envolvem diversos setores, pessoas e instituições, o que, de certa forma, acaba refletindo diretamente sobre o seu custo, deixando, assim, o pesquisador sem uma alternativa de baixo custo.

Devido a essa demanda por um equipamento de baixo custo, que registre e armazene dados, necessário principalmente para pesquisadores, torna-se fundamental a confecção de um equipamento capaz de realizar essa função, mas que tenha como característica ser de fácil construção, com baixo custo de manutenção e personalizável, para atender aos requisitos de aquisição. Então, por não se ter essa alternativa, faz-se necessário refletir de que forma torna-se possível ter um equipamento que seja de baixo custo, de baixo consumo energético, portátil, robusto e principalmente de configuração flexível.

Norteado por essas necessidades elaborou-se uma proposta metodológica com a utilização de sistemas embarcados para a coleta de dados. Para tanto, fez-se necessário refletir até que ponto a utilização dos sistemas embarcados, com seu baixo

custo, mas também com limitada capacidade de processamento, atenderia de forma satisfatória a solução deste problema? Quais as necessidades de armazenamento? O quanto simples seria a configuração deste sistema? E, por fim, como seria o procedimento de recuperação dessas informações armazenadas?

Estas perguntas remontam a um cenário em que um pesquisador tem, não somente a necessidade de um sistema robusto, eficiente, eficaz e portátil, mas também, uma necessidade de se ter um equipamento de fácil configuração e acessível para realizar seus experimentos.

Todos os pontos abordados definem uma problemática que envolve tecnologias emergentes, conceitos e princípios da ciência da computação e sistemas de código fonte aberto. Assim, este trabalho tenta estabelecer uma proposta referenciada em arquitetura aberta, livre, passível de modificações, tanto em sua estrutura, quanto em sua programação, vislumbrando minimizar ou mesmo resolver os problemas expostos.

1.2. JUSTIFICATIVA

Com a evolução tecnológica nas últimas décadas, uma arquitetura computacional tem se destacado na área da automação de uma forma geral: a arquitetura de sistemas embarcados.

Essa arquitetura pode ser utilizada para diversas finalidades, dentre elas a coleta e armazenamento de dados, tendo uma boa relação de custo-benefício, principalmente por ser, de *hardware* aberto e de *software de código fonte livre*.

Quando se é remetido a um conjunto de *software de código fonte livre*, isso permite que o desenvolvedor do sistema possa alterar os códigos ou bibliotecas já existentes para adequar as funcionalidades desse sistema a um objetivo específico.

Apropriando-se desses conceitos, ao reutilizar essa estrutura, adequando-a com o intuito de atender as demandas relatadas a partir de um conjunto de códigos que, presume-se, bem depurados, diminui-se relativamente o tempo de sua implementação.

Assim, a utilização de uma arquitetura aberta de *hardware* e *software* chamada *Arduino*¹ se justifica por possibilitar o acesso às suas funções, arquitetura e especificações, possibilitando adequá-las à metodologia proposta na problemática, de forma sistemática e simplificada.

Portanto, o trabalho se justifica pela importância que esse equipamento e sistema trarão à comunidade científica, à comunidade aberta do *Arduino* e na valorização do *software* livre. Bem como, na proposta de um equipamento genérico que auxiliará na condução de pesquisas a um custo mais baixo, por ser livre e aberto ao uso e desenvolvimento pela comunidade científica e educacional em geral.

Nesse sentido, o objetivo geral deste trabalho foi o de desenvolvimento de um conjunto de *hardware* e *software* para a coleta, monitoramento e armazenamento de transectos micrometeorológicos. Para conseguir tal objetivo, foram propostos os seguintes objetivos específicos:

- a) Desenvolver um conjunto de placas no padrão *Arduino* para a coleta, monitoramento e armazenamento de dados;
- b) Desenvolver um *software* (*firmware*) para *Arduino* de gerência da práxis de registro parametrizada por arquivo texto disposto em cartão SD;
- c) Avaliar e dimensionar o sistema quanto à sua autonomia operacional por meio de um sensor de corrente, bem como registrar a voltagem fornecida pela bateria;
- d) Testar e validar a usabilidade do sistema de aquisição de dados com um estudo de caso na área de conforto ambiental na pista de passeio da UFMT;
- e) Validar o método de monitoramento da coleta dos dados por meio de sinais sonoros e da comunicação sem fio por *bluetooth*².

A partir dos objetivos definidos, o texto desse trabalho foi organizado em 5 capítulos. Início, com uma introdução e justificativa desta pesquisa. O Capítulo 2 apresenta a revisão bibliográfica básica que alicerçará todo o trabalho, focando na definição e diferenciação do clima urbano, elucidando o conceito do transecto móvel, bem como os principais autores que a utilizaram, definindo o princípio de funcionamento dos sensores, bem como o método e forma aos quais esses dados são

¹ A arquitetura do *Arduino* será detalhada no tópico 2.6.

² É um padrão de comunicação sem fio utilizado para interconectar dispositivos de computação e acessórios, utilizando ondas de rádio de curto alcance, baixa potência e baixo custo (TANENBAUM, 2011).

armazenados e contextualizando a trajetória dos sistemas embarcados com a do *Arduino* como projeto de *hardware open-source*.

O roteiro metodológico deste trabalho é apresentado no Capítulo 3, no qual foram definidas as características do local do experimento para a validação do protótipo, bem como as estruturas necessárias para a construção do sistema, os componentes utilizados e sua calibração.

No Capítulo 4, são apresentados os resultados obtidos a partir dos referentes delineados no Capítulo 3, e no Capítulo 5, encontram-se as conclusões deste estudo e as sugestões para trabalhos futuros.

2. REVISÃO BIBLIOGRÁFICA

“A imaginação é mais importante que a ciência, porque a ciência é limitada, ao passo que a imaginação abrange o mundo inteiro.”

(ALBERT EINSTEIN)

Este capítulo aborda os principais conceitos necessários para a construção de um sistema automatizado de coleta e armazenamento de dados a ser utilizado em um transecto móvel, organizado em 6 subcapítulos, começando pela descrição do clima urbano, e foca no entendimento de como esse fenômeno é formado dentro da cidade. Depois aborda, no segundo subcapítulo, o transecto móvel, o que vem a ser esse método e quais as principais pesquisas que utilizou. O terceiro subcapítulo aborda o conceito de sensor, seus tipos, suas características e a suas formas típicas de leitura. O quarto subcapítulo é voltado ao sistema de aquisição de dados, chamado de *datalogger*, evidenciando o processo de conversão dos sinais fornecidos pelos sensores em um valor digital. E, por fim, o quinto subcapítulo diz respeito a sistemas embarcados, abordando as suas definições, conceituações, diferenças, e realiza uma abordagem comparativa sobre as principais placas para prototipação disponíveis no mercado.

2.1. CLIMA URBANO

O clima pode ser entendido como o resultado de diversos elementos atmosféricos (sol, ventos, massas de terra e água, topografia, vegetação, solo e outros) que ocorrem na atmosfera da terra, e é caracterizado de acordo com alguns elementos, como a temperatura do ar, a umidade do ar, os movimentos das massas de ar e a precipitação (ROMERO, 2001).

O processo de urbanização, que se intensificou pelo mundo desde o fim da Revolução Industrial, devido ao avanço tecnológico e ao mundo cada vez mais globalizado, resultou nas formações de grandes centros e dos conglomerados urbanos, fator este que afetou a prevalência climática.

No Brasil, o crescimento das cidades teve como principal responsável o êxodo rural, que ocorreu principalmente movido pelas oportunidades de trabalhos, pelo acesso à educação e à saúde, em contraponto as limitações do campo (UGEDA JÚNIOR e AMORIM, 2012).

Dessa forma, o crescimento demográfico em áreas urbanas se agravou, sobretudo, pela falta de planejamento, resultando em diversos problemas ambientais, como a modificação das condições iniciais do clima por meio da alteração do balanço energético no sistema superfície-atmosfera, a alteração na cobertura natural do solo, o incremento da massa edificada, resultando na criação de um clima peculiar no local, chamado de “clima urbano” (BARBOSA e VECCHIA, 2009).

Em consonância, Souza e Mattos (1997) descrevem o clima urbano como fortemente relacionado à ocupação antrópica do meio, possuindo fenômenos muitas vezes causados pela própria urbanização, constituindo assim uma situação climática específica, denominada clima urbano. Caracteriza-se, normalmente, pelo aumento da temperatura do ar ou da superfície de um local em relação ao seu ambiente rural circunvizinho (OKE e VOOGT, 2003). O incremento desses valores se deve às condições peculiares do meio ambiente urbano, como pela sua rugosidade, orientação, permeabilidade, ocupação do solo e, até mesmo, pelas propriedades físicas dos materiais constituintes.

Segundo Campos Neto (2007), o clima urbano pode ser analisado de forma mais ampla, usando variáveis climáticas e as alterações feitas no meio ambiente pelo homem. Essa análise, proposta por Maitelli (1994), ressalta que o processo de urbanização afeta a qualidade do ar pela poluição, diminuindo a transparência do ar e interferindo diretamente na transferência da radiação, tendo, como resultado, núcleos de condensação do ar, aumentando a precipitação.

Além disso, os materiais de construção, as edificações e o asfalto das ruas aumentam o armazenamento de energia e a impermeabilização do solo, o que pode

resultar em enchentes e na respectiva formação de um microclima local (CAMPOS NETO, 2007).

Em uma visão mais ampla do clima urbano, Monteiro e Mendonça (2003) enfatizam o clima urbano como um sistema dinâmico e adaptativo, revelando a essência de um fenômeno de complexidade evidenciado por Vianello e Alves (2002). Estes relatam que as principais dificuldades surgem quando se tenta considerar as interações ocorridas na natureza, desde que essas interações criem muitos mecanismos de realimentação, que agem amplificando ou amortecendo pequenas perturbações iniciais.

A importância de se estudar o clima urbano (CAMPOS NETO, 2007) como um sistema complexo de causas e efeitos foi observada em pesquisas anteriores, com a constante preocupação de que o meio disposto nas cidades não acarrete insalubridade no convívio da sociedade.

Esses fatores relatam a preocupação que se tem com o clima urbano e a sua importância, principalmente para o processo de planejamento de uma cidade, no controle da poluição do ar e no balanço de energia.

2.2. TRANSECTOS MÓVEIS

O transecto pode ser definido como uma faixa amostral de uma comunidade, com comprimento e largura variáveis, definidos de acordo com a necessidade do experimento e a critério do pesquisador (GARCIA e LOBO-FARIA, 2005).

O método de transecto móvel pode ser utilizado nos estudos de climatologia urbana para tentar verificar diferenças nas variáveis climáticas, como a variação de temperatura, umidade relativa do ar e outros. Ele foi utilizado pela primeira vez na Áustria em 1927.

No Brasil, foi utilizado pela primeira vez em 1982, por Hasenack (DA MARTHA et al., 2011), sendo uma maneira eficiente e de baixo custo utilizada para estudar as variações térmicas em uma cidade.

A aplicação desse método é feita normalmente com o auxílio de automóveis ou a pé, por meio de aparelhos portáteis que realizam as medições pontuais, possibilitando assim, percorrer uma extensa área urbana (CAMPOS NETO, 2007).

Nessa mesma linha, Fialho (2003) utilizou o método para estudar a relação entre os microclimas e o uso do solo na Ilha do Governador e conseguiu inferir que as áreas de maior gradiente térmico horizontal ao longo do transecto oscilaram nas partes extremas do dia, durante a manhã (8 às 10 horas) e no período do entardecer até o anoitecer (17 às 20 horas).

Souza e Batista (2005) utilizaram esse método para a confecção de perfis térmicos a fim de verificar a influência da resolução espacial na determinação das ilhas urbanas de calor em São Paulo. Conseguiram inferir que a resolução espacial é um fator de extrema importância na determinação das ilhas de calor urbanas.

Esse mesmo método também foi utilizado Campos Neto (2007) para realizar um estudo bioclimático no campus da Universidade Federal de Mato Grosso, e para isso utilizou 16 pontos fixos distribuídos, a fim de detectar a presença de ilhas de calor.

Da Martha et al. (2011) utilizaram esse método em Pelotas-RS, para avaliar o impacto que a Avenida Fernando Osório tinha sobre as variáveis de temperatura e umidade do ar e propuseram um projeto de requalificação urbana, envolvendo uma série de mudanças na via.

Para analisar o microclima do bairro Morada da Serra em Cuiabá-MT, Vasconcelos e Zamparoni (2011) aplicaram a metodologia do transecto móvel, coletando dados referentes à temperatura do ar e à nebulosidade às 20h. Conseguiram identificar a influência das edificações sobre o microclima local, propondo assim um planejamento do espaço urbano que apresenta soluções para se viver bem num ambiente saudável.

Barros (2012), com o objetivo de avaliar a extensão da influência das áreas verdes e sua fragmentação, quantificada pela dimensão fractal, sobre o ambiente térmico de um lugar, em um estudo conduzido em 5 bairros da região Sudeste da cidade de Cuiabá-MT, utilizou esse método para percorrer os bairros selecionados.

Dessa forma, o método de transecto móvel consiste no delineamento de uma área a ser estudada, na definição dos horários da coleta e no conjunto de sensores de variáveis climáticas ligados a um equipamento, que irá coletar e armazenar esses dados. Estes pontos são abordados nos próximos capítulos.

Assim, a técnica do transecto móvel, conforme aplicada pelos autores citados acima, é recomendada quando se tem a necessidade de se obter um conjunto de valores, como temperatura e umidade relativa do ar, de áreas relativamente grandes, como bairros e parques.

2.3. SENSORES

Um ponto importante a ser ressaltado nesse trabalho é o princípio de funcionamento dos sensores analógicos e digitais, que podem ser comparados aos cinco sentidos humanos, que são: a visão, a audição, o tato, o paladar e o olfato.

Esses sentidos são as referências das sensações que os humanos possuem, auxiliando a distinguir características de um objeto ou local, como por exemplo, se o objeto está quente ou frio, se está dia ou noite, a distância aproximada entre dois pontos. Dessa forma, esses sentidos nos proporcionam meios para capturar informações.

Analogamente à termocepção³ (percepção da temperatura), que é um subgrupo do tato⁴, existe o sensor de temperatura, que possibilita a mensuração da temperatura de um determinado corpo, funcionando muito semelhante ao sensor humano.

Os sentidos humanos podem ser classificados ou categorizados como dispositivos de entrada de informações que nos permitem analisar e interpretar o ambiente e, de certa forma, interagir com o mesmo (SANDHU, 2009).

Já os dispositivos eletrônicos com essa capacidade são chamados de sensores (WILSON, 2004), sendo eles conversores de fenômenos físicos em sinais normalmente elétricos, classificados como uma interface entre os dispositivos e os sentidos humanos. Ressalta-se que, entre esses, nem todos têm um correspondente funcional que capture tal sentido, como por exemplo, o prazer (WILSON, 2004). O

³ Refere-se à percepção das alterações térmicas do ambiente, geralmente incluído no sentido do tato. A sensibilidade térmica aumenta conforme o organismo se adapta a uma temperatura, sendo assim, um organismo adaptado ao calor é mais sensível ao frio e adaptado ao frio é mais sensível ao calor, tanto a curto prazo quanto a longo prazo, por predisposições filogenéticas (da espécie) ou ontogenéticas (do organismo durante a vida) (NERY et al., 2000).

⁴ O tato é um dos cinco sentidos clássicos propostos por Aristóteles, porém os especialistas o dividem em quatro outros sentidos: sistema somatosensorial (identificação de texturas), propriocepção ou cinestesia (reconhecimento da localização espacial do corpo), termocepção (percepção da temperatura) e nocicepção (percepção da dor).

inverso também é válido, principalmente quando se pensa sobre algo que os sentidos humanos não podem detectar ou detectam com imprecisão, como por exemplo, a luz infravermelha, que não é percebida pelos olhos sem um recurso específico (SANDHU, 2009).

Geralmente isso se refere a sinais que, para os seres humanos, são fracos ou difíceis de serem diferenciados no ambiente, de tal forma que, para que sejam detectados, precisam ser filtrados, amplificados e convertidos em outros sinais, como, por exemplo, um sinal luminoso ou sonoro, indicando sua presença (FRADEN, 2010).

Esses sinais obtidos pelos sensores podem ser medidos pelas suas propriedades elétricas, como a resistência, voltagem, capacidade e frequência (SANDHU, 2009), e podem ser qualificados, ou classificados quanto às suas características de funcionamento, como a função de transferência, a sensibilidade, a faixa de valores lidos, a acurácia ou a exatidão, a precisão, o ruído, a resolução e a largura de banda (WILSON, 2004).

Uma das vantagens de se substituir o sensor humano por um eletrônico é, justamente, pelo fato de o segundo ter um valor agregado menor, poder ser dimensionado quanto à sua precisão, acurácia, assim como quanto à fadiga ou estresse. Essa característica é muito ressaltada quando é necessário monitorar e registrar informações por um longo período de tempo, com várias repetições e em ambiente inadequado à permanência humana (SANDHU, 2009).

A relação funcional entre o sinal de entrada física e sinal de saída elétrico no sensor é chamada de função de transferência, é uma relação representada por curvas de ajuste, normalmente obtidas no processo de calibração do instrumento (WILSON, 2004; FRADEN, 2010).

Um fator diretamente relacionado à função de transferência é a sensibilidade do sensor que, no caso, relaciona o sinal de entrada física com o de saída elétrica. A sensibilidade pode ser entendida como o nível de reação de um instrumento ou sensor a quaisquer impulsos externos (WILSON, 2004).

Exemplo disso são sensores com alta sensibilidade, como os termistores, que podem alterar a sua resistência em 5% para cada °C, representando uma variação de aproximadamente 52 mV por °C, enquanto termopares possuem uma baixa taxa de

variação, normalmente em torno de $5\mu\text{V}$ por $^{\circ}\text{C}$. Logo, para realizar uma leitura adequada de um termopar, necessita-se de um conversor analógico digital (A/D) com resolução de no mínimo $5\mu\text{V}$ ou o uso de um amplificador.

Assim a sensibilidade pode ser definida como a quantidade de mudança no sinal de saída com relação à variação da entrada mensurada pelo sensor (PARK e MACKAY, 2003). Dessa forma, todos esses sensores possuem uma faixa de valores correspondente ao intervalo, espaço ou extensão de valores aceitos durante as conversões dos sinais de entrada físicos em sinais elétricos (WILSON, 2004), podendo ser considerada como a medida da menor quantidade mensurável por um instrumento ou, apenas, como o limite de detecção de um aparelho (SILVA, 2006).

No entanto, cada sensor tem um valor para a acurácia ou exatidão, que corresponde ao grau de concordância entre o valor obtido em uma medição e o valor verdadeiro, representando a exatidão de um valor lido, correspondendo assim à capacidade que um sensor tem para ler valores próximos ao valor real (FRADEN, 2010).

Dessa forma, a acurácia pode ser entendida como a exatidão do valor, em que um sensor descreve o quanto próximo o valor obtido está do real, modelando assim o erro que se pode esperar de um sensor (PARK e MACKAY, 2003), enquanto que a precisão é associada à repetitividade.

A precisão tem, como estatística usual, o desvio padrão de uma série de repetições, sendo uma característica que indica o grau de variação do resultado de uma medição, se diferenciando da exatidão, que se refere à conformidade com o valor real; assim, a exatidão é a distância estimada entre a medida e um valor verdadeiro, que normalmente é expressa como desvio ou desvio percentual (SILVA, 2006).

Um fator impactante na coleta e na própria transmissão dos sinais dos sensores é um fenômeno físico, chamado de ruído, que pode ser qualquer interferência que possa prejudicar a comunicação ou a leitura de uma informação, sendo ela interna, produzida pelo próprio conjunto de componentes, ou externa, causada por outros agentes (WILSON, 2004).

O ruído pode ser entendido como a presença de um sinal elétrico indesejado em um circuito, que pode vir a interferir no funcionamento do equipamento ou na

transmissão de um sinal, principalmente quando, em uma transmissão, a magnitude relativa do ruído é da mesma ordem do sinal a ser propagado naquele meio, comprometendo assim a integridade do sinal transmitido (PARK e MACKAY, 2003).

Um dos métodos propostos por Park e Mackay (2003) é a utilização de cabos blindados e a blindagem do sistema de aquisição de dados, a fim de minimizar as possíveis fontes dos ruídos, que podem ser classificadas como: o ruído gerado por fonte de energia; o gerado no canal de acoplamento e o do próprio circuito de recebimento.

Torna-se válido ressaltar a dificuldade ou a impossibilidade de se retirar todo o ruído de um sinal, contudo torna-se possível diminuí-lo por meio de técnicas, como o acoplamento condutivo, capacitivo ou indutivo (PARK e MACKAY, 2003).

Além do tratamento do ruído, outro fator impactante na qualidade do sensor é a resolução, que pode ser definida pela capacidade que ele tem em responder a pequenos estímulos externos (FRADEN, 2010), ou pode ser entendido como a medida da fineza de detalhe revelado pelo instrumento da medida (SILVA, 2006).

A largura de banda é a relação entre as frequências máxima e mínima num canal de transmissão, utilizada como um fator determinante para indicar a capacidade de transmissão de dados (WILSON, 2004), podendo ser entendida como uma propriedade física do meio de transmissão, que depende, por exemplo, da construção, da espessura e do comprimento do meio de transmissão (TANENBAUM, 2011).

Com base nesses conceitos apresentados sobre as características de um sensor, torna-se possível especificar qual tem melhor desempenho e qual está mais apto a ser utilizado em um cenário de estudo.

Normalmente, um sensor não funciona por si próprio, ele tende a ser parte de um sistema maior que, geralmente, incorpora componentes adicionais, como: condicionadores de sinais, processadores, memória e gravadores de dados (FRADEN, 2010).

2.4. SISTEMA DE AQUISIÇÃO E REGISTRO DOS DADOS

Tão importante quanto registrar um conjunto de informações é a etapa da aquisição dos dados, que pode ser entendida como um processo que irá transformar um fenômeno físico do mundo real, capturado por um sensor na forma de sinais elétricos, em um formato digital, para que depois possa ser armazenado, processado e interpretado (PARK e MACKAY, 2003).

Durante esse processo, os dados convertidos serão registrados em uma memória não volátil do equipamento, para depois serem recuperados. O equipamento com essa finalidade de conversão de sinal e armazenamento é chamado de *datalogger*, podendo ser concebido como um dispositivo coletor, conversor e registrador de dados.

Sua aplicação é recomendada quando os registros desses valores demandam tempo ou são relativamente difíceis de conseguir manualmente, como por exemplo, temperaturas ao longo do dia ou ano, temperatura de caldeiras, ou o monitoramento da tensão elétrica de uma determinada rede (BADIN JR., 2009).

De certa forma, na maioria das aplicações, quando se remete aos sistemas de aquisição de dados, eles não somente adquirem e armazenam as informações, mas também agem sobre eles, de forma a tentar controlar a sua integridade ou conteúdo (PARK e MACKAY, 2003). Isso é importante para a detecção de falha de funcionamento e, até mesmo, a interação com outros equipamentos.

Os elementos básicos de um *datalogger* são: o sensor, o meio de conexão do sensor com o conversor de sinal, o conversor de sinal, o *hardware* e o *software* de aquisição de dados.

Quanto aos sensores, conforme apresentado anteriormente, eles são a interface entre o mundo real e o mundo digital, sendo um meio de conversão de sinal para que o sistema de aquisição de dados o converta em dados digitais (PARK e MACKAY, 2003).

O meio de conexão de um sensor pode ser físico, por meio de cabos, que devem ser de preferência blindados, ou por conexão sem fio, que tem por finalidade fazer a ponte entre o sensor e o sistema de aquisição de dados.

O sinal elétrico gerado por esse sensor pode ser do tipo digital ou analógico, quando for digital, em essência, ele precisa apenas ser capturado; quando analógico, torna-se necessário ter um conversor analógico/digital (A/D).

Os conversores A/D transformam dados analógicos, normalmente variações de voltagem, em um valor equivalente no formato digital, compatível com o dispositivo de processamento de dados (FRADEN, 2010). Tendo ele, o conversor, como principais tarefas a filtragem, a amplificação, a linearização, a isolação e a excitação (PARK e MACKAY, 2003).

A finalidade do filtro de sinal é filtrar os ruídos, tentando manter a integridade do sinal capturado, para que não interfira no valor a ser convertido, sendo esse filtro de suma importância quando se tem sensores que geram baixa variação de tensão, como por exemplo, um termopar que tem seus valores variando na escala de mV.

Dessa forma, quando o ruído é da mesma ordem de magnitude ou maior que o sinal gerado, o ruído deve ser previamente filtrado, visto que, muitas vezes, os conversores A/D possuem filtro passa-baixo projetado para eliminar os ruídos de alta frequência (PARK e MACKAY, 2003).

Após a filtragem do sinal de entrada, dependendo do sensor que o gera, torna-se necessário amplificar o sinal, tendo por finalidade aumentar a resolução, tentando equiparar a variação do sinal com a escala que o conversor A/D consegue detectar, possibilitando assim uma conversão adequada (WILSON, 2004). Logo, essa filtragem tem por finalidade amplificar os sinais de pequena intensidade para que o conversor A/D detecte as suas variações.

A linearização (PARK e MACKAY, 2003) é um método utilizado para manter a proporção entre o valor do sinal de entrada e o valor convertido; faz-se

necessária porque alguns componentes, como um termopar, apresentam uma relação não linear quanto ao sinal de entrada, fator esse que pode dificultar a conversão do sinal.

O conversor de sinal A/D pode ser utilizado para proporcionar o isolamento dos sinais gerados pelo sensor e o computador ou microcontrolador. Isso ocorre pelo fato da origem do sinal ser de uma propriedade ou grandeza que, se fosse imputada direta no equipamento, poderia queimá-lo. Assim, o isolamento protege o equipamento de danos, principalmente quando os níveis de tensão são elevados (PARK e MACKAY, 2003).

Outra função do A/D é a excitação, que pode ser utilizada para ativar os sensores e obter os dados gerados por eles. Esse método pode ser utilizado em sensores que requerem uma voltagem externa ou sinais de excitação externa, como: medidores de tensão de termistores (PARK e MACKAY, 2003).

A conversão do sinal pode ser por meio de várias técnicas e, dentre elas, as principais são a conversão por aproximações sucessivas e a conversão por flash (PARK e MACKAY, 2003).

As aproximações sucessivas tornaram-se o método mais comumente utilizado para realizar a conversão em um A/D (WILSON, 2004), porque possuem altas taxas de amostragem e de resolução, e principalmente por possuir uma boa relação custo benefício.

Este método tem como principal vantagem ter um tempo fixo de conversão proporcional ao número de bits, n , em sua saída digital (PARK e MACKAY, 2003). Se o período de aproximação é T , então uma conversão de n -bits terá um tempo de conversão de cerca de nT . Dessa forma, cada bit a mais dobra a precisão da conversão, mas aumenta o tempo de conversão. O diagrama funcional desse método é apresentado na Figura 01.

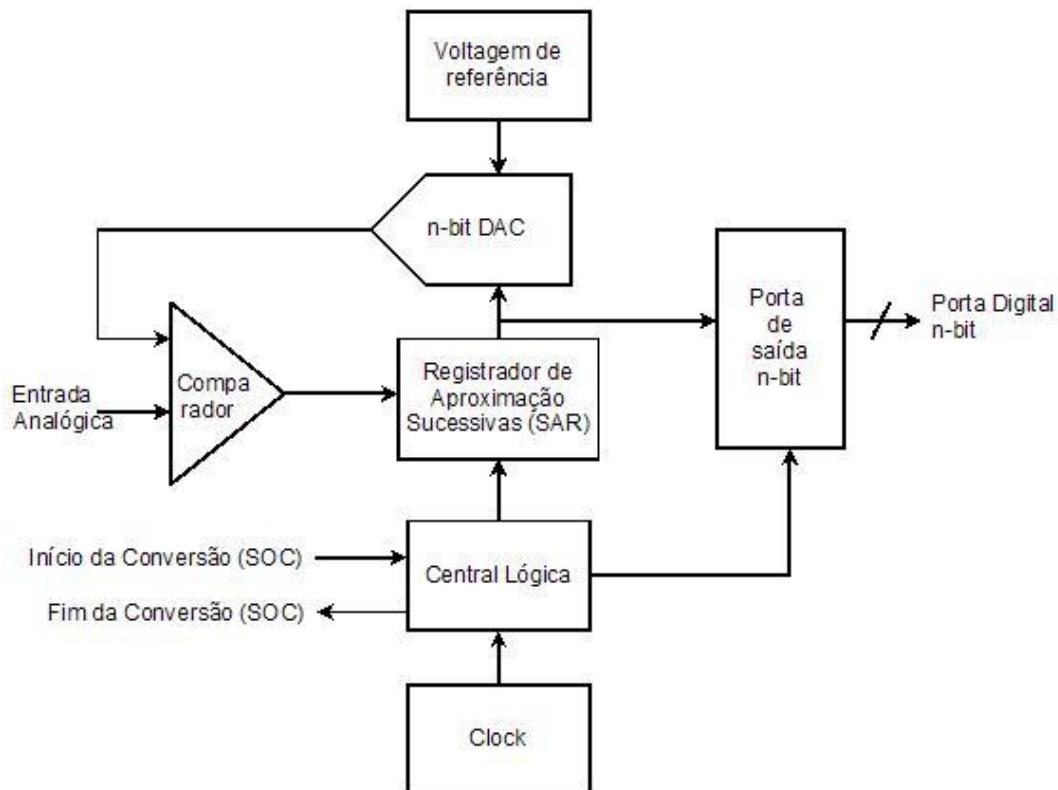


Figura 01 – Diagrama funcional de uma conversão n-bit por aproximação sucessiva.

Fonte: Adaptado de CATSOULIS (2009).

A técnica utilizada gera cada bit do código de saída sequencialmente, começando com o bit mais significativo (MSB) para o menos significativo. A operação se torna semelhante a uma busca binária e se baseia em comparações sucessivamente mais estreitas entre o sinal de entrada analógica e saída de um conversor A/D.

O processo de conversão se inicia definindo a entrada digital para o conversor, de modo que a sua tensão de saída analógica seja a metade da voltagem da escala total do dispositivo e, então, um comparador passa a ser utilizado para comparar a saída analógica com a entrada do sinal analógico a ser medido.

Se o sinal de entrada analógico for maior, o bit mais significativo (MSB) do conversor será ajustado para o nível lógico 1. Então o conversor configura o ajustado para 3/4 de tensão da escala completa. Se o sinal de entrada analógico for menor, o MSB do conversor será apagado e a entrada lógico 0 e o próximo bit mais

significativo da entrada serão ajustados para o lógico 1, definindo a saída analógica do conversor para 1/4 da escala.

Assim, cada passo divide a fração restante da faixa de entrada em sua metade e, em seguida, compara-se novamente com o sinal de entrada analógico. Esse processo será repetido até que todos os n -bits da conversão sejam determinados. Considera-se de suma importância que o sinal de entrada analógico não se altere durante o processo de transformação, o que justificaria a necessidade de circuitos de amostragem e retenção.

Já o método de conversão *Flash* (PARK e MACKAY, 2003) é o mais rápido, operando a velocidade de MHz. Esse tipo de dispositivo passa a ser usado quando se necessita de conversões muito rápidas, mas com uma baixa resolução, como, por exemplo, 8 bits.

A Figura 02 apresenta o diagrama funcional desse método de conversão Flash para n bits. O seu princípio de funcionamento é comparar a tensão do sinal de entrada com a tensão de referência e, assim, determinar a posição na série de resistências que corresponderia ao código de saída do conversor. Esse método ocorre mais rápido porque todos os bits do código de saída são encontrados simultaneamente, independente da quantidade de bits de resolução.

No entanto, quanto maior a resolução desse conversor, maior o número de comparadores necessários para realizar a conversão, de tal forma que a cada bit a mais de resolução dobraria a quantidade de comparadores, o que, proporcionalmente, tende a aumentar o tamanho do conversor e logo o seu custo.

Logo, a utilização desse tipo de conversor é mais recomendada quando se tem placas especializadas para essa finalidade, como por exemplo, os osciloscópios digitais de tempo real, as aplicações com processamento digital de sinal e outras aplicações de alta frequência.

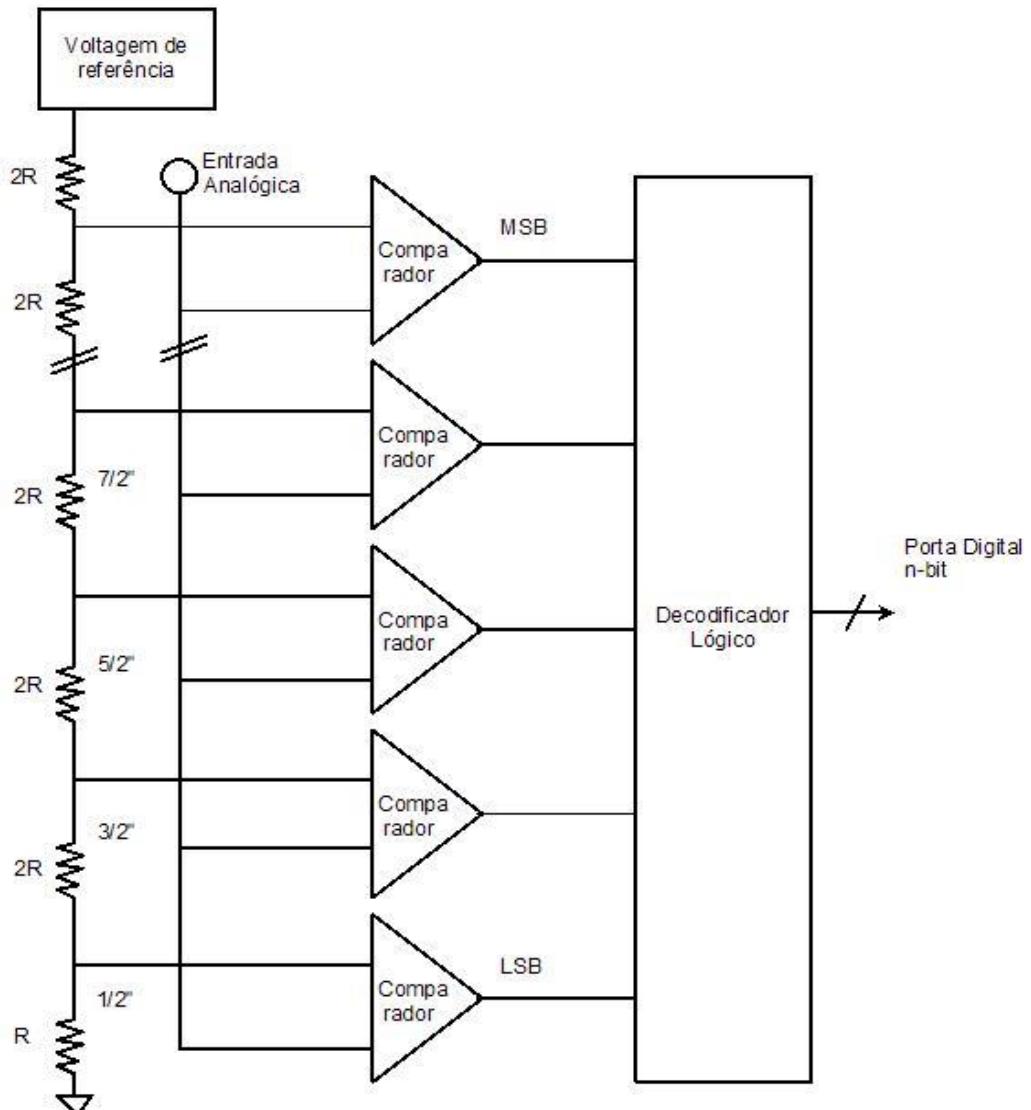


Figura 02 – Diagrama Funcional de uma conversão n-bit de um conversor flash A/D.

Fonte: Adaptado de CATSOULIS (2009).

Dessa forma, independente da técnica de conversão, caso o conversor tenha 10 bits de resolução e trabalhe com 5 Volts, isso indica que ele possui uma faixa de conversão de 2^{10} , o que representa 1024 unidades, de tal forma que, no caso de chegar 2.5 volts, ele irá retornar o valor do meio da escala. Essa relação pode ser estabelecida pela Equação 1:

$$\text{Unidade} = \frac{\text{Voltagem}}{\text{Resolução}} \quad [1]$$

Logo, o valor obtido nesse cenário teria uma sensibilidade de 0,0048828125 volts por unidade. Caso o valor lido tenha uma variação menor que essa sensibilidade, o conversor não conseguirá registrar tal variação, truncando⁵ o valor até que se obtenha uma variação que chegue ao próximo valor escalar.

No caso de o valor ser fornecido digitalmente, o *datalogger* terá que interpretar esses valores para conseguir então recuperar ou converter o conjunto de dados lido em um valor conhecido.

Todos os valores obtidos, tanto analogicamente quanto digitalmente, tendem a ser salvos (BADIN JR., 2009) em um dispositivo de armazenamento, que pode ser desde um cartão de memória ou pen-drive até discos rígidos.

Para controlar e comandar todos esses processos, tanto da coleta dos dados por meio de um conversor A/D quanto pela aquisição digital dos dados e seus respectivos armazenamentos, o *datalogger* possui um componente principal, o microcontrolador ou microprocessador, que será tratado no próximo tópico.

2.5. SISTEMAS EMBARCADOS

Os sistemas computacionais têm em sua essência três grandes classes de computadores, conforme definido por Delgado e Ribeiro (2010), que são: grandes computadores, computadores pessoais e computadores embarcados.

A classe de computadores de grande porte é formada por supercomputadores e servidores. A classe de computadores pessoais é formada por estações de trabalho (*workstation*), computador de mesa (*desktop*), computador portátil (*laptop* e *notebook*), computador de rede (*netbook*) e computador de mão (*palmtop*). A classe de embarcados é formada por microcontroladores embarcados.

Os computadores embarcados são muito mais numerosos do que os sistemas de desktop, mas muito menos óbvios. Na verdade, podem ter uns 30 ou mais, escondidos dentro de suas TVs, aparelhos de DVD, controles remotos, máquinas de lavar, celulares, condicionadores de ar, consoles de jogos, fogões, brinquedos e uma série de outros dispositivos.

Assim, tanto um computador *Desktop* quanto um sistema embarcado possuem um processador, memória e dispositivos de entrada e saída. Afinal a sua diferença reside em sua utilização, e isso reflete no projeto do sistema e no seu *software* (SALCIC e RADOJEVIC, 2011).

No entanto, os computadores de mesa podem executar uma variedade de aplicações, com recursos de sistema orquestrados por um sistema operacional. Ao executar aplicativos diferentes, a funcionalidade do computador é alterada, de tal forma que em um momento pode-se utilizá-lo como um processador de texto e, logo em seguida, como um tocador de música MP3 ou, até mesmo, ser um agente que conecte a uma base de dados, ficando a critério do usuário qual aplicativo ele quer carregar e em qual momento. Sistemas *desktop*, portanto podem ser definidos como sistemas de uso geral, uma vez que possuem processadores com esta característica de uso geral.

O sistema embarcado é normalmente dedicado a uma tarefa específica (CATSOULIS, 2009), como em uma central de injeção eletrônica de um automóvel, cuja finalidade é calcular a melhor relação da mistura do combustível ao ar, melhorando o desempenho e a eficiência do carro.

Dessa forma, a vantagem de se utilizar um microprocessador embutido sobre um conjunto de componentes eletrônicos dedicados está na sua funcionalidade provida pelo *software* que fica embutido nele e não, pura e exclusivamente, pelo seu *hardware*. Isso faz com que seja relativamente mais fácil projetar um sistema embarcado para determinados cenários, do que utilizar apenas circuitos elétricos (SALCIC e RADOJEVIC, 2011).

Assim pode-se entender que sistema embarcado é uma combinação de *hardware* e *software* projetado para realizar uma função dedicada e, em muitos casos, fazem parte de um sistema ou produto maior (STALLINGS, 2010), podendo ser definido como todo dispositivo computacional (computador) inserido em outro equipamento, o qual o controlará, mas, normalmente, o usuário desconhece isso (SOARES, 2008).

2.5.1. A Estrutura do sistema embarcado comparado a um computador

Em essência, um computador é uma máquina projetada para processar diversos programas, armazenar e recuperar dados. Os dados podem ser números em uma planilha eletrônica, caracteres de texto em um documento, pontos de cor em uma imagem, formas de onda de som, ou o estado de algum sistema, como um ar condicionado ou um CD player.

Todos os dados são armazenados no computador como números binários, que são manipulados por meio de operações, de tal forma que, para exibir uma imagem na tela, seja necessário mover uma matriz com esses números para a memória de vídeo, sendo que cada número, nessa situação, representaria um pixel de cor.

Tudo o que um computador faz, desde uma simples navegação na web até a sua respectiva impressão, consiste em mover e processar números. A eletrônica de um computador não é nada mais do que um sistema projetado para manter, movimentar e alterar esses números (CATSOULIS, 2009).

Para isso, um computador é composto por muitas partes, tanto de *hardware* como de *software*. O “cérebro” do computador é o processador, sendo ele o *hardware* que executa os programas. De forma complementar, ele possui um conjunto de memórias, que são usadas para armazenar programas, bem como armazenar os dados que são gerados e ou manipulados por ele. Além disso, ele possui também dispositivos de armazenamento de dados, que são responsáveis por armazenar os dados gerados por um processamento.

O *software* controla as operações e funcionalidades de um computador. Existe um conjunto de "camadas" de *software* em um computador (Figura 03). Tipicamente uma camada interage apenas com a camada imediatamente acima ou abaixo da mesma.

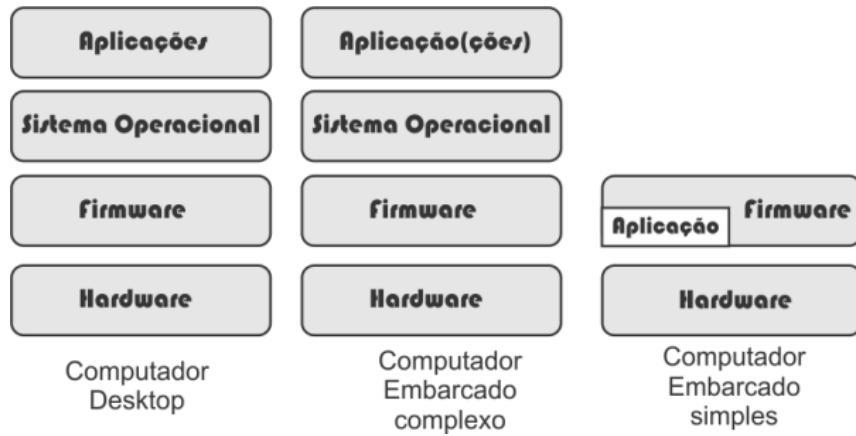


Figura 03 – As camadas de *Software*.

Fonte: Adaptado de CATSOULIS (2009).

No nível mais baixo dessa estrutura de camadas de *software* existe um conjunto de programas que são executados pelo processador quando o computador é ligado. Estes programas inicializam outros subsistemas de *hardware* para um estado conhecido e configuram o computador para o seu correto funcionamento. Esses *softwares* quando armazenados permanentemente em uma memória do computador são chamados de *firmware* (CATSOULIS, 2009).

O *bootloader* está localizado no *firmware*, sendo ele um programa especial, executado pelo processador, que tem por finalidade carregar o Sistema Operacional e colocá-lo na memória para que o processador possa executá-lo. O *bootloader* está presente tanto em computadores *desktop* quanto em alguns computadores embarcados.

Acima do *firmware* (Figura 03), o Sistema Operacional (S.O.) controla todas as operações do computador, organizando o uso da memória e controlando os dispositivos, como o teclado, o mouse, telas e disco. Ele também fornece uma interface para o usuário, permitindo-lhe executar aplicativos e acessar arquivos no disco (TANENBAUM, 2010).

No entanto, nem todos os sistemas embarcados usam, ou mesmo precisam de um Sistema Operacional. Muitas vezes, um sistema embarcado simplesmente executa um conjunto de código dedicado à sua tarefa, e a presença de um Sistema Operacional pode ser considerada, nessa situação, um exagero (SALCIC e RADOJEVIC, 2011).

Em outros casos, como por exemplo, na utilização de um sistema embarcado para servir como um roteador de rede, o Sistema Operacional oferece um conjunto de *software* que vai simplificar o processo de desenvolvimento. No entanto, se ele é realmente necessário, vai depender da finalidade do computador embarcado e, em menor grau, da preferência do design (CATSOULIS, 2009).

No nível mais alto da camada, o *software* de aplicação se constitui nos programas que proporcionarão a funcionalidade do computador. Tudo abaixo dessa camada pode ser considerado como *software* do sistema.

Para sistemas embarcados, o limite entre a aplicação e o *software* do sistema é muitas vezes tênue. Isso reflete diretamente sobre o princípio de os sistemas embarcados serem projetados para atingir o seu objetivo de forma direta e simples (CATSOULIS, 2009).

2.5.2. Princípio de funcionamento de um processador

O processador é um dispositivo eletrônico capaz de manipular dados (informações) de uma forma específica, por meio de um conjunto de instruções armazenado na sua memória.

O processamento dessas instruções (*opcodes*) consiste basicamente em duas etapas: a primeira é a leitura das instruções que estão na memória, uma por vez; e a segunda, a execução de cada uma delas (STALLINGS, 2010).

Para o processador, as instruções são um conjunto de bits que, uma vez trazidos para os registradores internos, decodificados por uma unidade de controle e executados, utilizando geralmente as unidades funcionais da Unidade Lógica e Aritmética, geram resultados. Essas instruções podem ativar pequenos segmentos da CPU, fazendo-a realizar operações algébricas, como a adição de dois valores.

Uma sequência de instruções é um programa de código de máquina. Cada tipo de processador pode ter uma arquitetura de conjunto de instruções diferente, o que significa que a funcionalidade das instruções e os padrões de bits que os ativam variam. Essas instruções de processador são muitas vezes simples, como “somar dois números” ou “chamar uma função” (CATSOULIS, 2009).

Assim, cada processador ou microcontrolador possui um conjunto de instruções e padrões que podem variar de acordo com a sua arquitetura, representando que nem sempre um *software* desenvolvido para um, servirá para o outro, e da mesma forma, os requisitos e instruções de um, não serão necessariamente compatíveis com a de outros.

Isso ressalta a importância de se conhecer a estrutura e a forma de trabalho de cada microcontrolador, para que se possa dar uma sobrevida ao sistema desenvolvido nele e, principalmente, tentar usá-lo em sua totalidade.

Assim, a forma como um computador será usado, quais as tarefas que ele deve realizar, e como os seres humanos interagem com ele e outros sistemas, determina a funcionalidade da máquina e, portanto, sua arquitetura, memória e I/O.

Um computador de mesa, representado pelo diagrama da Figura 04, possui uma memória principal para armazenar o sistema operacional, aplicativos e dados, e uma interface para dispositivos de armazenamento em massa. Tendo ele uma variedade de dispositivos de entrada de dados (teclado, mouse e áudio), de saída (interface de exibição e áudio) e de conectividade (redes e periféricos).

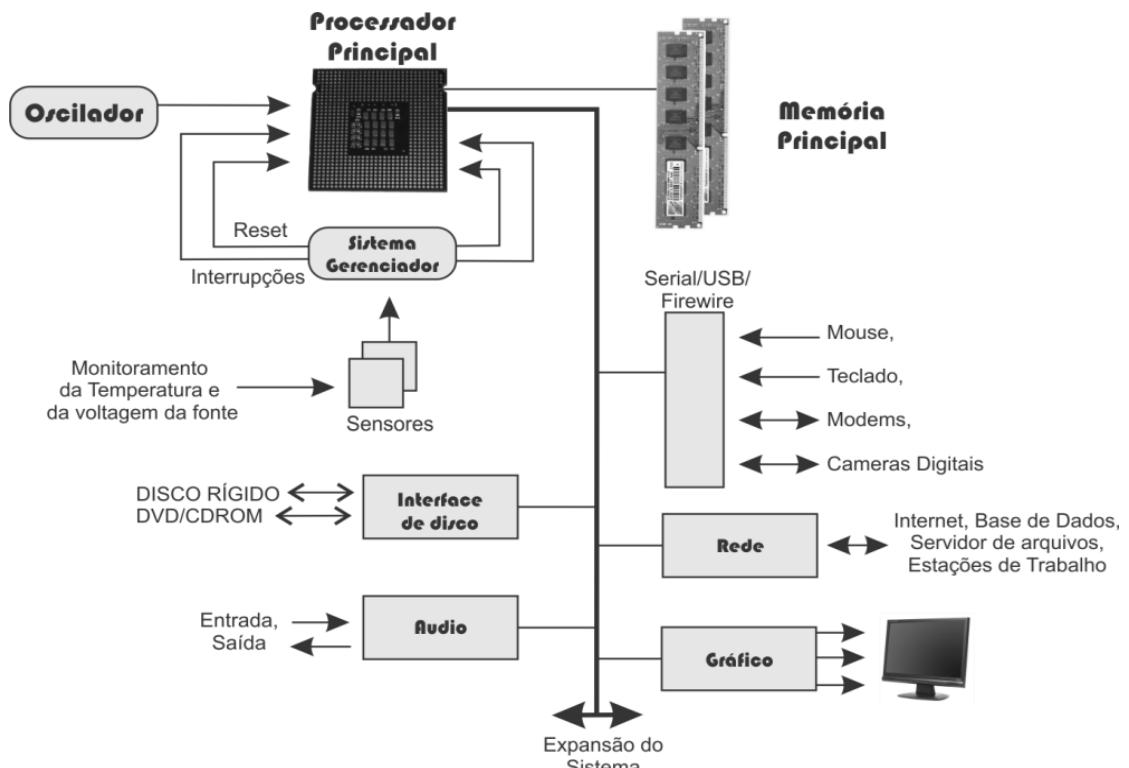


Figura 04 – Arquitetura de um computador genérico.
Fonte: Adaptado de CATSOULIS (2009).

Os computadores de grande porte embutidos podem também tomar a mesma estrutura apresentada na Figura 04. Assim, em termos de *hardware*, muitos sistemas de alto desempenho embarcado não são muito diferentes, no tocante a sua estrutura, de um computador convencional (SALCIC e RADOJEVIC, 2011).

Já sistemas embarcados menores, que usam microcontroladores, têm como principal vantagem a de que esse processador incorpora muito das funcionalidades do computador em um único chip (Figura 05). Ele possui no mínimo, uma CPU, uma pequena quantidade de memória interna (ROM e/ou RAM) e um conjunto de portas de I/O, que são utilizadas para programar subsistemas do micro controlador. Esses subsistemas fornecem funcionalidades adicionais e são comuns em muitos processadores.

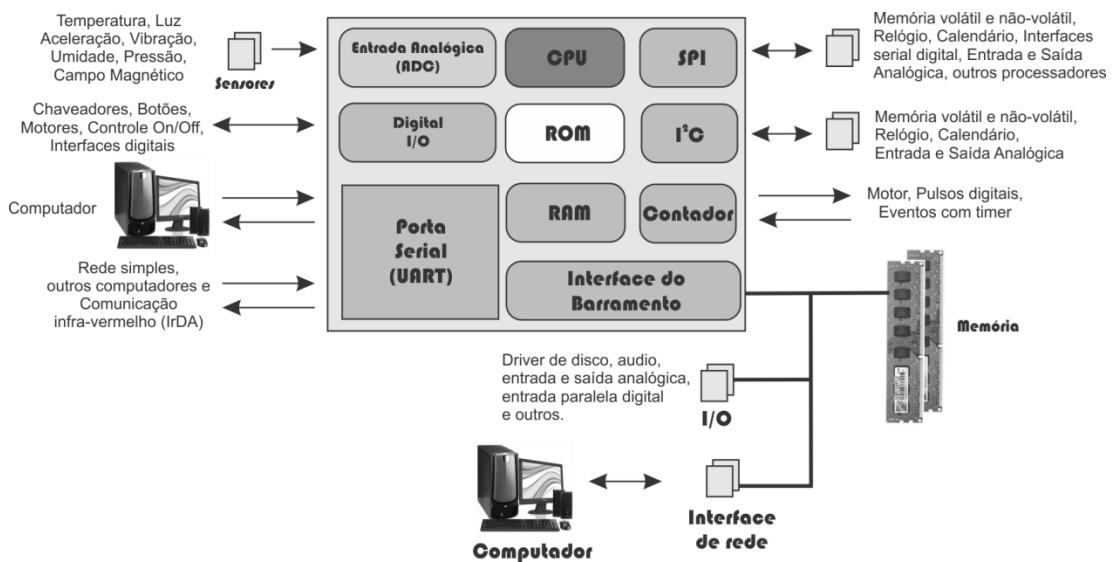


Figura 05 – Arquitetura de um micro controlador genérico.
Fonte: CATSOULIS (2009).

Os recursos mais comuns de entrada e saída de dados, na arquitetura embarcada, são as portas digitais de I/O ou GPIO. Essas portas podem ser configuradas pelo *software* como de entrada ou de saída. Caso seja definido como porta digital de entradas, ela pode ser usada para ler o estado de interruptores ou botões, ou para ler o estado digital de um dispositivo. Caso a porta digital seja configurada como sendo de saída, ela poderá ser utilizada para comunicar com outros

dispositivos externos, mudando para isso o seu estado, de ligados para desligados e vice-versa (SALCIC e RADOJEVIC, 2011).

Além das portas digitais, alguns microcontroladores também possuem portas analógicas, interfaces de rede, portas USB (CATSOULIS, 2009), soquetes para cartão de memória e até mesmo uma unidade de processamento gráfico com saída HDMI para ligar em uma televisão ou monitor.

Por meio das portas analógicas torna-se possível obter valores de sensores analógicos. Dessa maneira, um computador embarcado pode medir, por exemplo, os níveis de luz, pressão, vibração, temperatura ou aceleração, ar ou água, umidade, ou campo magnético.

As portas seriais permitem que o computador embarcado conecte-se a um computador, um modem, ou ainda, outro sistema embarcado. Outras formas de interface serial, como SPI e I2C, fornecem uma maneira de expandir as funcionalidades do microcontrolador. Elas permitem que por meio dessas interfaces, sejam conectados, por exemplo, chips de memória externa, relógio, sensores com interfaces digitais, chips de áudio e até mesmo outros processadores.

Como pôde ser visto, os computadores embarcados fornecem uma gama de conexões e recursos, que podem ser utilizados tanto para a captura de dados, quanto para a interação ou monitoramento de determinados cenários.

Essas conexões nos permitem conectar diversos tipos de sensores, como por exemplo, de temperatura, de umidade relativa do ar, de gás, de luz infravermelha, GPS entre outros, tornando-o um componente de suma importância para este trabalho.

2.5.3. Placas para desenvolvimento e prototipação de embarcados

Nesta seção são abordadas as principais placas de desenvolvimento e prototipação de sistemas embarcados, focando as versões com maior apelo comercial e com maior conjunto de acessórios.

2.5.3.1. *Arduino*

O *Arduino* é uma placa com um microcontrolador de baixo custo que foi idealizada para o uso leigo (KUSHNER, 2011) e, por meio dele, torna-se possível conectar a quase todos os tipos de sensores, luzes, motores e outros aparelhos, sendo ele projetado como uma tecnologia lúdica para ensinar os conceitos básicos de eletrônica e de programação. Sendo possível, também, construir sistemas interativos e, em seguida, compartilhar seu projeto com o mundo, colocando-o na rede e contribuindo com a comunidade aberta do *Arduino*.

Esse conjunto de ferramentas disponibilizado pelo sistema *Arduino* possibilita detectar e controlar diversos tipos de equipamentos do mundo físico, sendo ele uma plataforma de desenvolvimento de *hardware open-source*, baseado em uma placa microcontrolada e em um ambiente de desenvolvimento para escrever código para essa estrutura (ARDUINO, 2013). Ele é uma fusão de três elementos críticos: *hardware*, *software*, e da comunidade (OXER e BLEMMINGS, 2009).

Monk (2010) complementa que ele é uma pequena placa microcontrolada com uma porta USB usada para conectar ao computador e uma série de portas que podem ser conectadas a outros dispositivos eletrônicos, tais como motores, relês, sensores de luz, diodos de laser, alto-falantes, microfones. Eles podem ser alimentados pela conexão USB do computador ou a partir de uma bateria de 9V.

O *Arduino* é registrado sob a licença *Creative Commons Attribution Share - A like license*⁶ que permite a cópia, distribuição, transmissão e adaptação (OXER e BLEMMINGS, 2009), e com o projeto do *hardware* fornecido em formato CAD do *Eagle*⁷, possibilitando que todas as atualizações e melhorias possam ser estudadas e aprimoradas por seus usuários.

⁶ Disponível em: <<http://creativecommons.org/licenses/by-sa/3.0/>>.

⁷ Disponível em: <<http://arduino.cc/en/uploads/Main/arduino-diecimila-reference-design.zip>>.

O *software* de desenvolvimento do *Arduino* também é *open-source*, codificado em Java e liberado sob a licença GPL. A linguagem de desenvolvimento desse ambiente é C/C++ e ela possui um conjunto de bibliotecas já implementadas com diversos tipos de rotinas otimizadas para essa estrutura e toda ela também é fornecida sob a mesma licença (ARDUINO, 2013).

Assim é possível entender que a estrutura *Arduino* envolve não somente um *hardware open-source*, mas também um conjunto de *software* e bibliotecas *open-source*, e que ambas podem ser alteradas e adaptadas conforme a necessidade.

O fato do *hardware* do *Arduino* possuir um conjunto de soquetes ou pinos, representado na Figura 06, permite que se desenvolvam placas adicionais, chamadas de *Shields*, expandindo assim as funcionalidades dessa arquitetura, como por exemplo, adicionando receptores GPS, telas de LCD, conexão de rede *Ethernet* e drive para cartão de memória do tipo SD⁸.

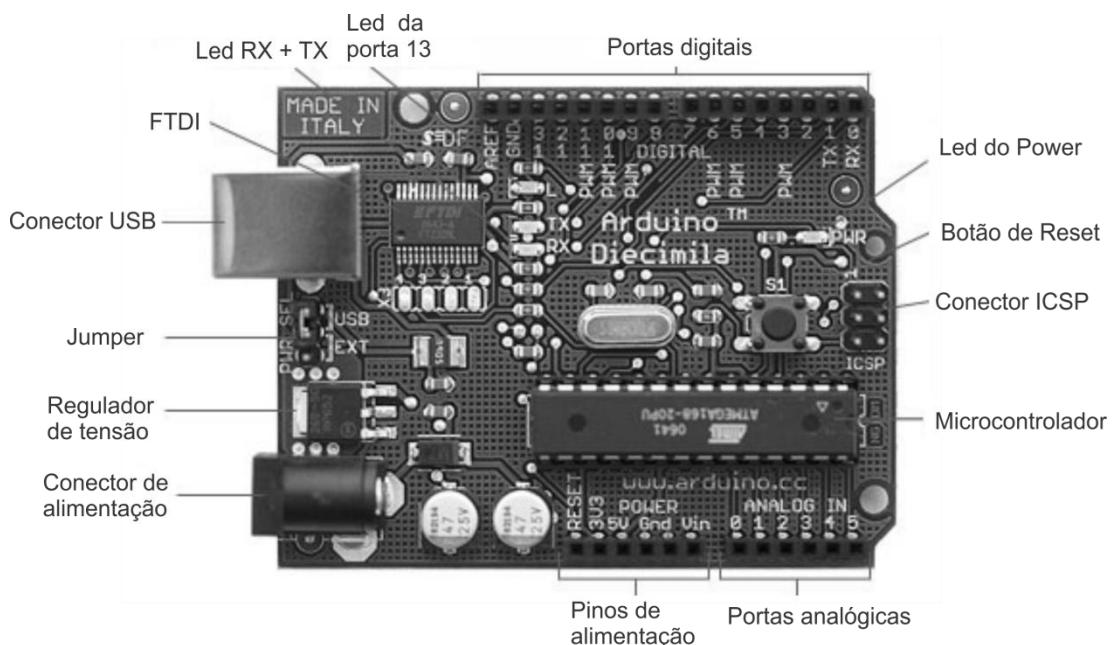


Figura 06 – Placa do Arduino.

Fonte: Adaptado de ARDUINO (2013).

⁸ *Secure Digital Card* – São cartões de memória que evoluíram da tecnologia MultiMediaCard (MMC) sendo adicionado a eles o recurso de criptografia e gestão de direitos digitais. Atualmente essas memórias são usadas em celulares, *smartphones* e câmeras fotográficas digitais.

A proposta do *Arduino* (Anexo C) é propiciar um ambiente onde qualquer pessoa interessada possa criar e programar suas ideias, além de permitir que a mesma compartilhe com a comunidade, de preferência, ideias que gastam relativamente pouco, se comparado a outras arquiteturas (WARREN, ADAMS e MOLLE, 2010).

Nesse sentido, essa arquitetura tem um forte apelo educacional, possibilitando, por meio dela, o estudo de princípios físicos, como a eletricidade, a resistência e outros fenômenos; isso se torna possível, porque foi projetado para ser usado por pessoas que não tenham uma formação técnica, incluindo um *software* que contém um conjunto de códigos de exemplo para demonstrar como utilizar seus recursos (MARGOLIS, 2011).

Para isso, o *Arduino* possui um conjunto de portas digitais e analógicas (Figura 06) que permite ao desenvolvedor, conectar fios e ou *shield*, para ligar componente externo, possui também um regulador de tensão, para que, ao se conectar uma bateria externa ao conector de alimentação, ele mantém a tensão de alimentação em 5 volts.

O *firmware* gerado pela *IDE* do *Arduino* é enviado através da porta USB, utilizando um conversor de sinal USB em serial. Os ledes de RX e TX, indicados na Figura 06, representam a utilização da comunicação por esse meio.

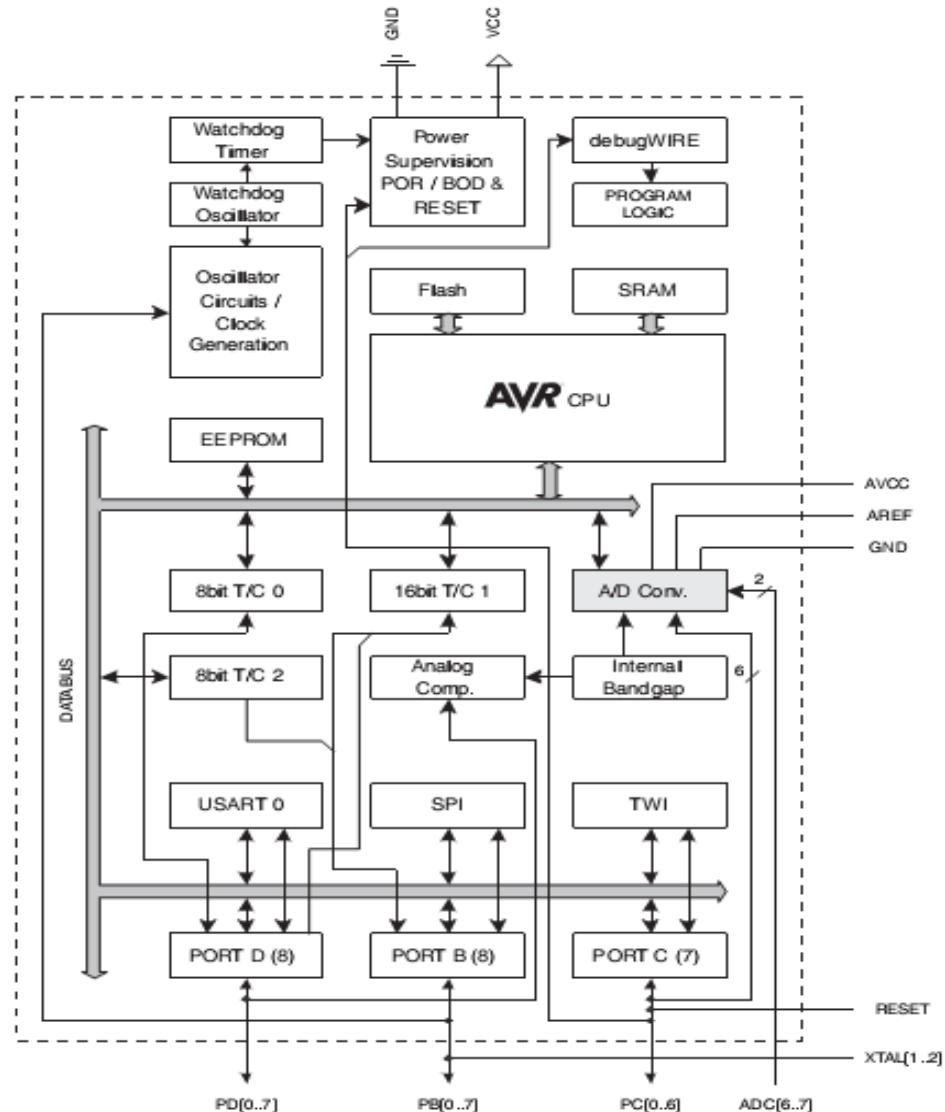


Figura 07 – Diagrama de bloco do AVR ATmega328.

Fonte: Atmel (2009).

Quanto ao *hardware* do *Arduino* versão UNO (ARDUINO, 2013), ele possui um micro controlador ATmega328 (Figura 08), fabricado pela Atmel, sendo ele um AVR de 8 bits, trabalhando a uma velocidade de 16 MHz, podendo trabalhar com um cristal oscilador de até 20 MHz, o que resultaria em 20MIPS, entretanto é necessário reconfigurar o *bootload* para adequá-lo a essas configurações.

Este microcontrolador possui 32KB de memória flash para programação com garantia para 10.000 ciclos de leitura e escrita, 2KB de memória SRAM interna, 1K de EEPROM com garantia para 100.000 ciclos de leitura e escrita.

Quanto aos seus periféricos (Figura 08), possuem as seguintes características: Dois 8-bit Timer / Contadores com *Prescaler* separado e com modo de comparação;

Um 16-bit Timer / Contador com *Prescaler* separado, *Compare Mode*, e *Capture modo*; Contador de tempo real com oscilador separado; Seis PWM Canais; 8-channel 10-bit ADC com os encapsulamento TQFP e QFN / MLF ou 6-channel 10-bit ADC com PDIP; USART Serial Programável; Interface serial SPI Master / Slave; Interface serial I2C; Watchdog com Timer programável com oscilador separado On-chip; Pinos de Interrupções e *Wake-up*; Seis modos de *sleep*: Espera, Redução ADC ruído, de economia de energia, *Power-down*, espera e espera estendida (ATMEL, 2009).

Além disso, ele possui 14 portas digitais ou GPIO, das quais 6 delas podem ser usadas como saídas PWM e mais 6 portas analógicas com resolução de 10 bits. Possui também uma porta UART no padrão TTL de 5V, duas portas para interrupção e suporte ao protocolo de comunicação SPI e I2C (ARDUINO, 2013).

Esses recursos possibilitam ao desenvolvedor um conjunto de ferramentas, tanto de *hardware* quanto de *software*, para o desenvolvimento de soluções, a fim de resolver ou contornar as limitações dos equipamentos existentes no mercado; isso é percebido na narrativa de Loup et al. (2011), quando desenvolveram um projeto utilizando sistemas embarcados para o monitoramento de servidores em uma sala cofre, consistindo em uma estrutura que mensurava a temperatura da sala e enviava mensagem, via *bluetooth*, para desligar os servidores quando a temperatura da sala ultrapassasse o nível de segurança.

Já Souza et al. (2011) apresentam o *Arduino* como uma opção de baixo custo para a aquisição de dados por meio de transdutores para análise das variáveis físicas e seu respectivo uso em sala de aula.

Enquanto Cavalcante et al. (2011) utilizaram o *Arduino* como uma interface alternativa, na aquisição e automação de dados em atividades experimentais de física, via porta USB do computador, a fim de estudar o processo de carga e descarga de um capacitor.

Zachariadou et al. (2012) utilizaram o *Arduino* para criar um laboratório de baixo custo para os alunos de graduação de engenharia elétrica e física, tendo como objetivo principal proporcionar aos alunos a instrumentação básica para trabalharem os conceitos de semicondutores e para a elaboração de experimentos físicos.

Kristinsson et al. (2012) verificaram a utilização do *Arduino* como um gerador de números aleatórios verdadeiros, lendo os ruídos gerados pelas portas analógicas que não possuíam nada conectado a elas.

Kornuta et al. (2013) desenvolvem um sistema com *Arduino* para controlar o ciclo de abertura e fechamento de uma bomba peristáltica digital, ativadas por meio de comandos seriais, a fim de criar diferentes taxas de fluxo, gerando uma vasta variedade de formas de onda de fluxo no experimento.

Como ótimo exemplo, Faugel e Bobkov (2013) utilizaram o *Arduino* para manter um conjunto de *hardware* de sintetização direta de sinal com controle de fase em funcionamento, substituindo o microcontrolador antigo pelo utilizado no *Arduino*, com as devidas adaptações na estrutura do *hardware*.

Conforme os autores citados acima, é possível ver o quanto flexível e robusto é o *Arduino*, podendo ser utilizado desde um controlador de temperatura de uma sala cofre, até um sistema de controle de bomba peristáltica.

Assim, a estrutura de *hardware* e *software* do *Arduino*, com base nos artigos referendados acima, fornece um conjunto básico de requisitos necessários para que ele trabalhe como um *datalogger*, necessitando apenas fazer as devidas modificações ou ampliações em sua estrutura.

2.5.3.2. Raspberry Pi – Modelo B

A ideia de um computador barato e pequeno surgiu em 2006, quando Eben Upton, Rob Mullis, Jack Lang e Alan Mycroft, do laboratório de informática da Universidade de Cambridge, identificaram um declínio ao decorrer dos anos, do conhecimento dos alunos de ciência da computação na área de arquitetura de computadores (RASPBERRY PI, 2013).

Ele foi desenvolvido no Reino Unido pela Fundação Raspberry Pi que integrou todo o *hardware* em uma única placa, tendo como principal finalidade estimular o ensino de ciência da computação nas escolas.



Figura 08 – Placa do Raspberry Pi – Modelo B.

Fonte: RASPBERRY PI (2013).

O *Raspberry Pi* (Figura 08) possui um sistema “tudo em um” (SoC) que inclui um processador Arm de 700Mhz, 512 MB de memória RAM, no modelo B, e um processador gráfico (GPU) VideoCore IV com conexão HDMI e RCA. O sistema de armazenamento é por meio de um cartão *SD* ou pendrive conectado à porta USB (RASPBERRY PI, 2013).

As vantagens do *Raspberry Pi* sobre o *Arduino* é o fato dele ter mais memória RAM e velocidade de processamento, além de permitir a instalação de um sistema operacional, bem como dar acesso a diversos periféricos por meio da porta USB, como por exemplo, um modem 3G. Contudo, a desvantagem sobre o *Arduino* é o consumo de energia que nos modelos A e B do Raspberry Pi está em 300 mA e 700 mA, respectivamente, contra 16 mA.

2.5.3.3 BeagleBone Black

O *BeagleBone Black* foi desenvolvido pela Fundação *BeagleBoard.org* com o objetivo de disseminar o conceito do desenvolvimento por meio do *software* e *hardware* aberto. O seu desenvolvimento se deu em parceria com funcionários da Texas Instruments, a fim de criar um sistema de prototipação que fosse poderoso e aberto (BEAGLEBOARD.ORG, 2013).

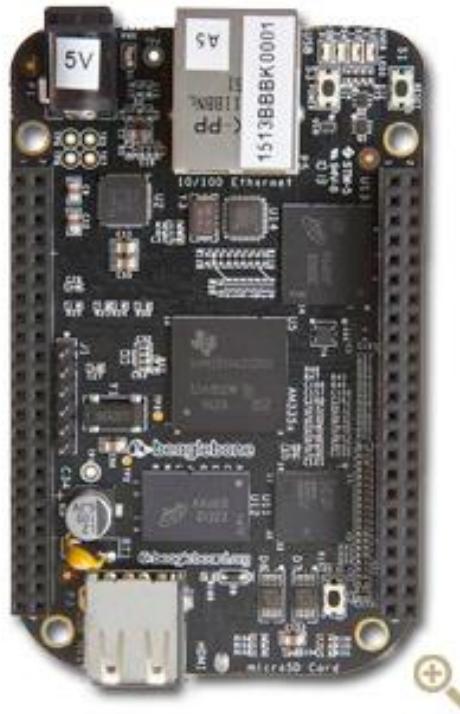


Figura 09 – Placa do BeagleBoard Black.

Fonte: BEAGLEBOARD.ORG (2013).

O *BeagleBoard Black* (Figura 09) possui um processador Sitara AM335x ARM® Cortex-A8 da Texas Instruments com clock de 1Ghz, com 512 MB de memória RAM DDR3 e um processador gráfico (GPU) PowerVR SGX530 com conexão microHDMI.

Além de ele vir pré-instalado com a distribuição *Linux Ångström*, um diferencial desse sistema é o suporte a *IDE Cloud9*⁹, que é um ambiente de desenvolvimento *on-line* para aplicativos Javascript baseados em Node.js ((BEAGLEBOARD.ORG, 2013).

Uma vantagem do *BeagleBoard Black* sobre o *Raspberry pi* é a *IDE Cloud9*. Já sobre o *Arduino*, é relativamente a mesma do *Raspberry pi*, além do já citado acima. Contudo, a desvantagem sobre o *Arduino* é a mesma do *Raspberry Pi*, o consumo de energia, que neste modelo fica entorno de 600 mA, contra 16 mA.

⁹ Disponível em: <<https://c9.io/>>.

2.5.4. Comparativo entre as placas para desenvolvimento e prototipação de embarcados

Para comparar os recursos providos pelas placas de desenvolvimento e de prototipação de sistemas embarcados, elaborou-se a Tabela 01, contendo os principais parâmetros de comparação, como: a arquitetura, a quantidade de memória RAM e a programável disponível, bem com o consumo de energia.

Tabela 01 – Comparativo entre as placas de prototipação de embarcados

	<i>Arduino</i>	<i>Raspberry Pi – B</i>	<i>BeagleBone Black</i>
Arquitetura	AVR – 8 bits	ARM 11	ARM Cortex A8
Memória RAM	2 KB	512 MB	512 MB DDR3
Memória Flash Programável	32 KB	-	2GB
Velocidade	16 MHz	700 Mhz	1 GHz
Portas Digitais	14	8	65
Resolução A/D	10 bits	-	12 bits
Portas A/D	6	0	7
Consumo de energia (mA)	30 mA	700 mA	600 mA
Preço em dólar ¹⁰	\$12	\$35	\$55

Elab.: PEREIRA, R. B. de O. (2013).

¹⁰ Data de referência dos preços: 29 de outubro de 2013, site pesquisado: <<http://www.ebay.com>>.

3. MATERIAIS E MÉTODOS

“Os que se encantam com a prática sem a ciência são como os timoneiros que entram no navio sem timão nem bússola, nunca tendo certeza do seu destino.”

(LEONARDO DA VINCI)

A seguir será apresentado como este trabalho foi desenvolvido, descrevendo os métodos, técnicas, materiais e instrumentos desenvolvidos para realizar a pesquisa de campo.

Para isso, foi planejado um roteiro metodológico para a realização do estudo, sendo dividido em 8 etapas.

A primeira foi composta pela definição das características gerais do local do experimento. A segunda define a seleção do horário e do dia típico. Na terceira foi apresentada e definida a área experimental, composta pelos seus respectivos trajetos. A quarta foi focada nos equipamentos, contendo seus esquemas elétricos e diagramas. A quinta, no processo de fabricação desses equipamentos. A sexta, na calibração do equipamento construído com outros *datalogger* certificados. A sétima, sobre os métodos estatísticos utilizados para validar os dados coletados pelo equipamento no estudo de caso e, na oitava e última etapa, foram definidas as formas utilizadas para monitorar o sistema, bem como mensurar o seu consumo.

3.1. CARACTERÍSTICAS GERAIS DO LOCAL DO EXPERIMENTO

Cuiabá, a capital de Mato Grosso, está localizada no Centro da América do Sul, nas coordenadas geográficas centrais de 15° 35' 56" latitude Sul e 56° 06' 01" longitude Oeste (GOOGLE EARTH, 2013).

Essa cidade possui um comportamento térmico característico. Segundo os dados do Instituto Nacional de Meteorologia (INMET), durante o período de 1970 a 1989, teve temperaturas médias superiores a 28°C, no período entre 21 de agosto a 07 de abril (CAMPELO JÚNIOR et al., 1991).

Em Cuiabá está sediada a Universidade Federal de Mato Grosso, fundada em 10 de dezembro de 1970, pela Lei nº 5.647¹¹, a partir da fusão da Faculdade de Direito de Cuiabá, que foi criada em 1952, e do Instituto de Ciências e Letras de Cuiabá (GIANEZINI, 2009).

O campus Cuiabá, representado na Figura 10 pela área demarcada em verde, possui um perímetro de 4.370 metros e uma área total de 764.510 metros quadrados. Situado na Avenida Fernando Corrêa da Costa, nº 2.367, Bairro Boa Esperança, Cuiabá-MT, possui suas coordenadas geográficas centrais de 15° 36' 32" latitude Sul e 56° 03' 55" longitude Oeste e está a 176 metros acima do nível do mar (GOOGLE EARTH, 2013).

A escolha, para a realização do experimento no campus da UFMT, se deu pela praticidade para a realização dos testes com o equipamento, e essa facilidade se constrói pela premissa de ser o mesmo local em que se trabalha.

Dessa forma, o equipamento poderia ser colocado à prova e, no caso de falhas ou necessidades de ajustes, isso poderia ser analisado e efetivado, dependendo de seu grau de complexidade, no próprio local, o que em teoria, agilizaria o processo de adaptação e melhoria do equipamento.

¹¹ Disponível em: <<http://www2.camara.leg.br/legin/fed/lei/1970-1979/lei-5647-10-dezembro-1970-358725-normaatualizada-pl.html>>.



Figura 10 – Universidade Federal de Mato Grosso - campus Cuiabá.

Fonte: GOOGLE EARTH (2013) - adaptado.

3.2. APRESENTAÇÃO DO TRAJETO UTILIZADO NO TRANSECTO

A área para a realização do transecto móvel foi definida inicialmente com base na pista de caminhada, que tem o seu ponto Zero na Avenida Fernando Corrêa da Costa e terminando entre o Zoológico e a Faculdade de Ciências Médicas.

Essa pista de caminhada foi criada com o objetivo de proporcionar aos moradores da cidade um percurso alternativo para a prática de atividade física, com o intuito de promover um passeio agradável, mantendo contato com a natureza e percorrendo boa parte do zoológico.

Devido às obras para a Copa do Mundo de 2014, em que Cuiabá é uma das cidades sede, o trajeto inicial teve que ser alterado para a adequação e revitalização do campo de futebol e das quadras poliesportivas.

O novo percurso, representado na Figura 11, tem o seu início no final da pista de caminhada, localizada ao lado do Biotério da UFMT, e segue em direção à Avenida Fernando Corrêa da Costa, contornando as obras de adequação e revitalização, e então, retorna pela pista central em direção ao Instituto de Computação, resultando em um trajeto de aproximadamente 3.519 metros.

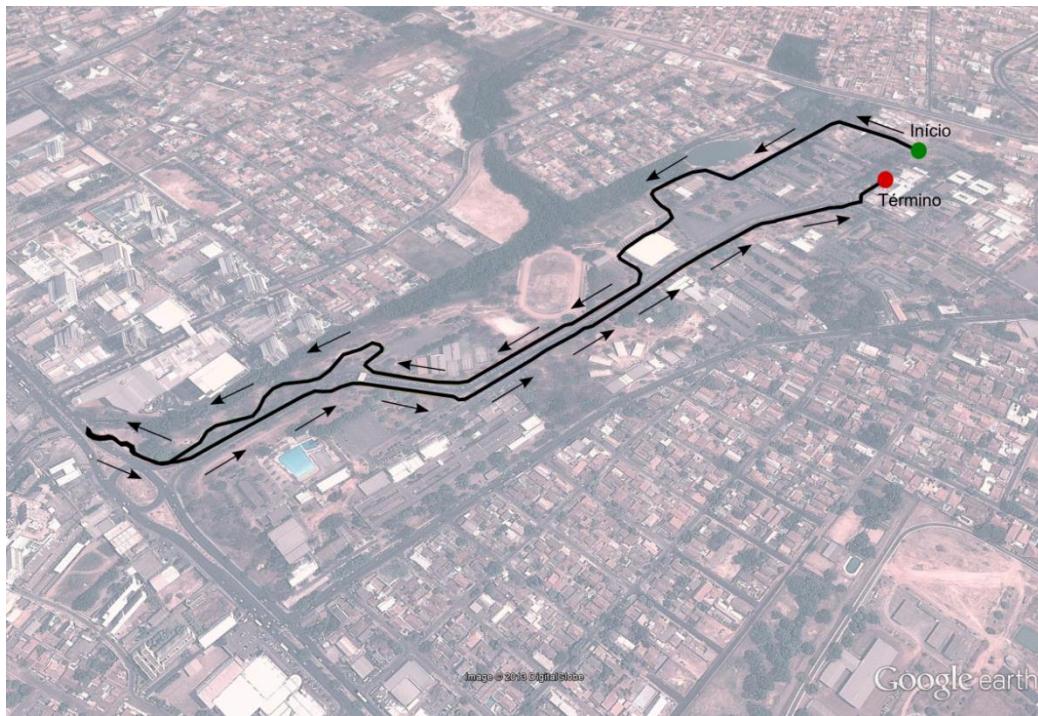


Figura 11 – Trajeto do transepto móvel na UFMT.

Fonte: Google Earth (2013) - adaptado.

O trecho apresentado foi analisado conforme as suas características morfológicas, incluindo o seu entorno, tipo de pavimento, distância e coordenadas, tendo sua caracterização apresentada no capítulo de resultados.

A escolha deste trajeto se deu pela diversidade de ambientes existentes na UFMT, podendo ser considerado bom para o teste do equipamento, pois, neste trajeto presume-se que haverá diferença de temperatura e umidade relativa do ar, devido às variedades de entorno e pavimentos.

3.3. SELEÇÃO DE HORÁRIO E DIA TÍPICO

A seleção do dia típico leva em consideração as condições meteorológicas: céu claro, sem chuva e com ventos reduzidos. Em Cuiabá, essas são as condições típicas da cidade, normalmente com temperatura média superior a 28°C; segundo Campelo Júnior et al. (1991), sobre a característica do clima em Cuiabá: “[...] a ocorrência de períodos de curta duração com temperatura baixa alternando-se a temperaturas moderadas ou elevadas [...]”.

A WMO (*World Meteorological Organization*)¹² recomenda que as observações meteorológicas de um dia típico devam acontecer às 00h, 06h, 12h e 18h GMT (*Greenwich Meridian Time*) (BARROS, 2012).

Para a validação do equipamento foi realizada uma coleta de dados no dia 19 de Abril de 2013, seguindo o trajeto apresentado na Figura 11. Este dia foi considerado típico, por ter céu claro, sem chuva e com ventos reduzidos.

3.4. ZOIOLOGGER – SISTEMA DE COLETA E ARMAZENAMENTO DE DADOS

Os fatores preponderantes para a escolha da arquitetura a ser usada neste experimento foram: consumo de energia, bibliotecas do *software* e *hardware* disponíveis e quantidade de placas acessórias disponíveis no mercado.

Quanto ao consumo de energia, conforme apresentado na Tabela 01, o padrão *Arduino* leva vantagem, não somente pelo consumo de 30 mA, mas também pelo encapsulamento ser do tipo DIP, o que facilita a construção artesanal de placas modificadas.

No tocante a bibliotecas para desenvolvimento e prototipação de placas acessórias, o *Arduino* possui tantas quanto os outros fabricantes. Já quanto a placas acessórias disponíveis no mercado, o *Arduino* leva uma relativa vantagem sobre os concorrentes, contudo, é importante ressaltar que os outros fabricantes possuem placas adaptadoras às *shields* do *Arduino*.

O principal fator levado em consideração para a escolha do *Arduino* para este trabalho foi o consumo de energia e a possibilidade de confecção de uma placa otimizada, a fim de minimizar ainda mais o seu consumo de energia. Dessa forma, a arquitetura e o padrão de desenvolvimento, selecionados para esse projeto, são as do *Arduino*.

O sistema de coleta e armazenamento de dados será chamado de Zoiologger, e será constituído por um conjunto de placas segundo o padrão *Arduino*, podendo, ou não, ser usado em conjunto com outros periféricos, desde que se respeite a pinagem, a tensão de alimentação e os critérios de *software*.

¹² Disponível em: <http://www.wmo.int/pages/index_en.html>.

A concepção geral do projeto desse sistema é apresentada na Figura 12, que tem como base a estrutura de placas do Zoiologger e os dispositivos e sensores que podem ser conectados à placa principal do *Arduino* através de *Shields*.

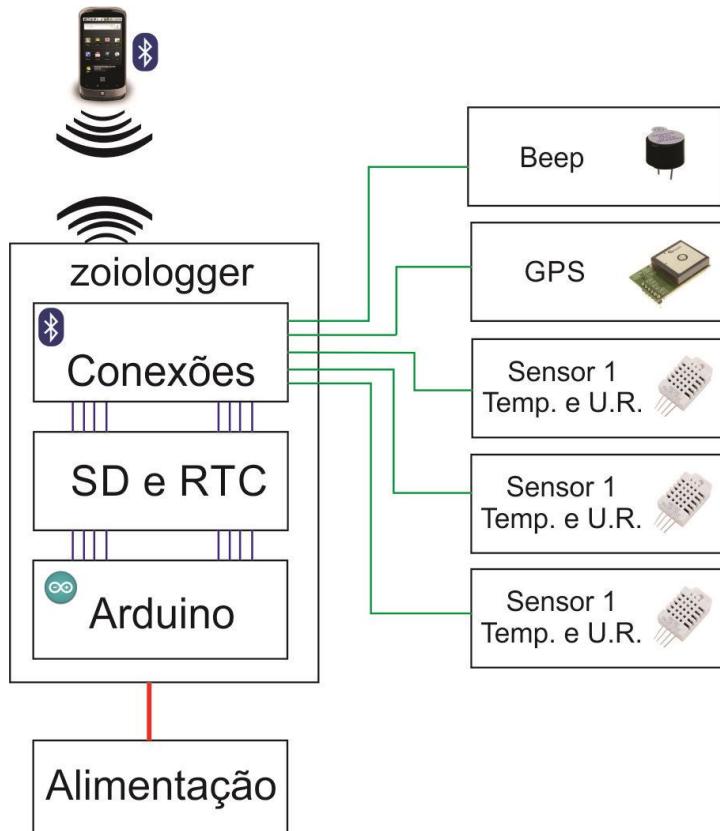


Figura 12 – Diagrama esquemático do Zoiologger.
Elab.: PEREIRA, R. B. de O. (2013).

A primeira placa do *Arduino* adaptada terá o sistema básico de alimentação, contando com fusível para proteção de sobrecarga e seus respectivos reguladores de tensão; para a alimentação desses circuitos, essa placa terá duas formas de alimentação: uma por meio de bateria externa e a outra por meio de uma bateria de 9 V que será colocada dentro do equipamento.

A segunda placa terá o sistema de relógio com bateria independente para manter a data e hora no caso de falta de energia, bem como um meio de armazenamento de dados. A terceira placa terá os soquetes para conexão dos sensores e dos respectivos módulos.

Com base nessa definição, pensou-se em uma organização de comunicação levando em conta os protocolos disponíveis no mercado, bem como a disponibilidade de tais circuitos, com a qual se chegou ao resultado apresentado na Figura 13.

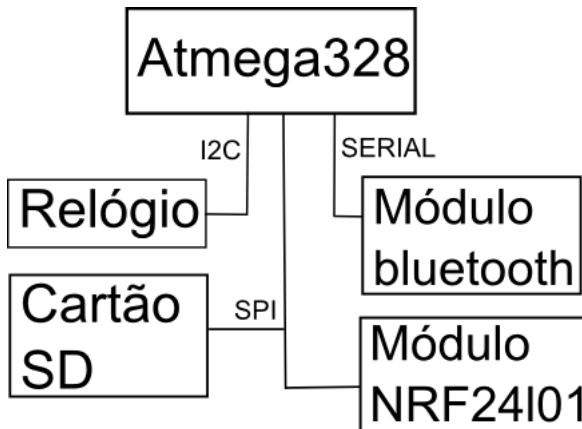


Figura 13 - Arquitetura de comunicação do Zoiologger.
Elab.: PEREIRA, R. B. de O. (2013).

Com base nessa arquitetura proposta, o meio de armazenamento dos dados coletados pelo equipamento será o cartão *SD*, pelos fatos de utilizar o barramento de comunicação *SPI* além de ser de fácil aquisição no mercado e por ter um bom custo x benefício.

O sistema de obtenção de data e hora para auxiliar no controle da sincronização do tempo de coleta será um *realtime* externo que utilizará o barramento de comunicação *I2C*, com o objetivo de não ocupar mais portas digitais disponíveis aos sensores.

Para o monitoramento, foi pensado em um módulo *bluetooth* pelo motivo de consumir menos energia que os módulos com tecnologia *WiFi* e *3G*, contudo, pelo fato de a comunicação ser serial, é possível adaptar esses módulos ao sistema, porém com a penalidade de aumentar consideravelmente o consumo de energia, fator esse que deve ser levado em consideração quando se trabalha com dispositivos móveis durante longo períodos de tempo.

Caso haja a necessidades futuras de comunicação com outros módulos, foi pensado na utilização de módulos transceptores *NRF24L01*, em virtude de consumirem relativamente pouca energia e de terem um custo x benefício relativamente alto, se comparado às tecnologias *WiFi* e *ZigBee*.

A escolha do sensor de temperatura se deu pelos critérios de consumo de energia, resolução, acurácia e principalmente pelo custo x benefício. Da mesma forma, o sistema de posicionamento global foi selecionado com base na acurácia, sensibilidade, possibilidade de ajuste do DATUM e também pelo custo x benefício.

Todas as comunicações com as *shields* são feitas pelos soquetes disponíveis no *Arduino*, respeitando seus barramentos e protocolo de comunicação, conforme o apresentado na Figura 13.

Dessa forma, a concepção desse modelo, apresentada na Figura 12, bem como os componentes selecionados, será abordada nas seções subsequentes.

3.4.1. *Hardware do Zoiologger*

A partir dessa seção apresenta-se todo o processo de análise e fabricação do hardware utilizado neste trabalho, detalhando-se as necessidades iniciais até a própria confecção dos equipamentos.

Para a elaboração dos diagramas elétricos e dos diagramas das placas foi utilizado o *software Eagle PCB*¹³ (CADSOFT, 2013), que é gratuito para projetos de placas com dimensões de até 9 cm por 10 cm, versão 5 da *Cadsoft* para *Linux*, em conjunto com as bibliotecas da *Sparkfun*¹⁴ (SPARKFUN, 2013) do *Arduino* para o *Eagle PCB*.

3.4.1.1 *Levantamento de requisitos*

Para a realização do levantamento de requisitos de *hardware*, realizou-se uma investigação sobre as demandas da práxis associada à execução de transepto móvel, considerando a possibilidade de parametrização das informações, resultando na seguinte lista de requisitos funcionais (Tabela 02).

¹³ Disponível em: <<http://www.cadsoftusa.com/download-eagle/?language=en>>.

¹⁴ Disponível em: <<https://www.sparkfun.com/static/eagle/>>.

Tabela 02 – Requisitos funcionais de hardware.

[RF_001] Sistema micro controlado simplificado
O equipamento deve ter uma estrutura de <i>hardware</i> com a quantidade mínima de componentes necessários para o seu funcionamento afim de minimizar o consumo de energia.
[RF_002] Sistema de proteção de curto e sobrecarga
O equipamento deve estar dotado de componentes que controlem o consumo de energia, prevenindo-o contra sobrecarga e queima de circuitos.
[RF_003] Sistema de obtenção de data e hora
O equipamento deve possuir um relógio de tempo real para auxiliar na obtenção da data e horário.
[RF_004] Armazenamento de dados removível
O equipamento deve ter uma estrutura, por meio de cartões de memória, que possibilite a sua remoção.
[RF_005] Sistema de monitoramento da alimentação
Por meio de um <i>hardware</i> específico, o <i>datalogger</i> deve ser capaz de monitorar a tensão do sistema de alimentação.
[RF_006] Conexão dos sensores ao equipamento de coleta
O equipamento deve conter um conjunto de bornes para facilitar a fixação dos fios em seu circuito.
[RF_007] Sistema de conexão sem fio
O equipamento deve prover acesso aos dados através de conexões sem fio, utilizando preferencialmente o barramento serial e/ou barramento SPI, habilitando conexão via <i>bluetooth</i> ou com o transceptor NRF24L01 ¹⁵
[RF_008] Abrigo e suporte para o sensor de temperatura e umidade
Para uma melhor fixação e suporte, o equipamento deve prover de uma estrutura que auxilie na fixação do sensor de temperatura e umidade relativa do ar, modelo DHT-22, para que não sofra insolação direta e o proteja de obstáculos.
[RF_009] Abrigo e suporte para o GPS
Visando uma melhor fixação e suporte ao GPS, o equipamento deve prover uma estrutura que auxilie na fixação do GPS, bem como fornecer um barramento de conexão ao módulo principal.
[RF_010] Estrutura geral de suporte
Para a melhor fixação de todos os sensores, bem como a acomodação da fiação que irá conectar os sensores ao equipamento, deve ser produzida uma estrutura geral de suporte para a fixação dos mesmos.

Elab.: PEREIRA, R. B. de O. (2013).

¹⁵ Tecnologia de comunicação sem fio de baixo consumo desenvolvida pela Nordic.

3.4.1.2. Componentes eletrônicos selecionados

Para a construção do equipamento, em consonância com os requisitos funcionais, foram selecionados os seguintes componentes.

3.4.1.2.1 Microcontrolador ATmega328

O ATmega328 (Figura 14) é um microcontrolador AVR de 8 bits, de arquitetura RISC, com um total de 131 diferentes tipos de instruções em seu conjunto. A maioria delas utiliza apenas um ciclo de clock para execução. O microcontrolador também possui 32 registradores de 8 bits de uso geral e pode ser comercializado com os seguintes encapsulamentos: DIP de 28 pinos, TQFP de 32 contatos, QFN de 28 e 32 contatos (ATMEL, 2009).

Para este trabalho foi escolhido o encapsulamento DIP pela facilidade que ele proporciona ao desenvolvimento de circuitos de forma manual.

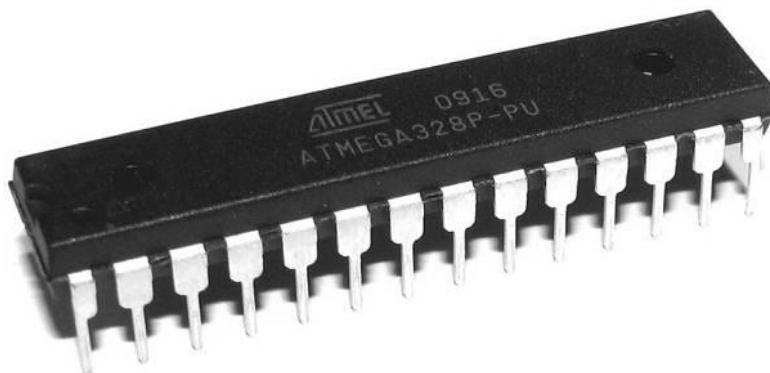


Figura 14 – Microcontrolador ATmega328P-PU – encapsulamento DIP
Fonte: ATMEL (2009).

A Tabela 03 apresenta as características elétricas do microcontrolador Atmega328.

Tabela 03 – Características do ATmega328.

Característica	Valor típico
Voltagem de operação	1.8 a 5.5 V, padrão 5 V DC
Velocidade	1 – 20 MHz @ 1.8 a 5.5 V
Pinos digitais de I/O	14 (e 6 com saída PWM)
Pinos de Entrada Analógica	6
Corrente DC por pino	40 mA
Memória Flash	32KB (0.5KB usado para o <i>bootloader</i>)
Preço no Ebay ¹⁶	R\$ 5,00

Fonte: ATMEAL (2009).

3.4.1.2.2. Sensor de temperatura e umidade – AM2302/DHT22

O DHT22 (Figura 15), também conhecido como AM2302 ou DHT03, é um sensor digital de temperatura e umidade relativa do ar, encapsulado em uma estrutura de plástico branco, tendo ele um baixo consumo energético (Tabela 04).



Figura 15 – Sensor de temperatura e umidade relativa AM2302 / DHT22
Fonte: AOSONG (2013).

Todos os seus sensores são conectados a um microcontrolador de 8 bits, cada sensor é calibrado em uma câmara de precisão e o coeficiente de calibração é salvo

¹⁶ Data de referência dos preços: 29 de outubro de 2013, site pesquisado: <<http://www.ebay.com>>.

em uma memória *EPROM OTP* do sensor (AOSONG, 2013). Além disso, ele possui protocolo de comunicação próprio de um único barramento.

Os sensores utilizados neste trabalho foram calibrados também com outros equipamentos, conteúdo este que será abordado na seção 3.6.

Tabela 04 – Características do sensor de temperatura e umidade relativa do ar - AM2303 / DHT22.

Característica	Valor típico
Tensão de alimentação	3.3 a 6 V - DC
Sinal de Saída	Sinal digital em um barramento
Sensores	DS18B20 para temperatura e um Polímero Capacitivo para umidade
Faixa de leitura	Umidade: 0 – 100% RH Temperatura: -40 ~ 125 °C
Acurácia	Umidade: +- 2% Temperatura: +-0.2 °C
Resolução	Umidade: 0.1% RH Temperatura: 0.1 °C
Período de coleta	2s (média)
Consumo de energia	Mínimo: 1.3 mA Média: 1.5 mA Máximo: 2.1 mA
Preço no Ebay ¹⁷	R\$ 12,00

Fonte: AOSONG (2013).

3.4.1.2.3 GPS – SKM53

O SKM53 (Figura 16) é um sensor GPS com antena embutida, baseado no chip de alto desempenho da MediaTek 3327 com uma acurácia de 3 metros (Tabela

¹⁷ Data de referência dos preços: 29 de outubro de 2013, site pesquisado: <<http://www.ebay.com>>.

05), que pode ser usado em área de *canyons* urbanos e em meios ambientes com folhagem densa.



Figura 16 – GPS - SKM53.

Fonte: SKYLAB (2010).

Tabela 05 – GPS SKM53 UART.

Parâmetro	Especificação
Tipo de recebimento	Banda L1, C/A code, 22 Tracking / 66 Acquisition-channel
Sensibilidade	Tracking: -165dBm Acquisition: -148dBm
Acurácia	Posição: 3.0 m 3D RMS Velocidade: 0.1 m/s Tempo (PPS): 60ns RMS
Tempo de aquisição	Início a frio: 36s Início a morno: 33s Início a quente: 1s
Tensão de alimentação	5 V – DC
Consumo de energia	55mA
Taxa de atualização padrão	1Hz
Preço no Ebay ¹⁸	R\$ 90,00

Fonte: SKYLAB (2010).

¹⁸ Data de referência dos preços: 29 de outubro de 2013, site pesquisado: <<http://www.ebay.com>>.

Ele possui uma porta de comunicação serial *UART full duplex*, com velocidade padrão de 9600bps e trabalha com o protocolo NMEA-0183¹⁹ (SKYLAB, 2010).

O sistema de referência padrão utilizado pelo GPS é o WGS-1984. Ele permite que se ajuste o DATUM²⁰ com base em uma tabela interna com 222 opções (Anexo B). O SAD-69, que foi utilizado como referência é a opção 196.

Para realizar esse procedimento, foi enviado o comando \$PMTK330,196*20 pela porta serial do GPS, entretanto, toda vez que o GPS for inicializado é necessário reenviar esse comando (Anexo A) para reconfigurá-lo.

3.4.1.2.4 Relógio de Tempo Real – DS1307

O DS1307 (Figura 17) é um relógio de tempo real – RTC com baixo consumo de energia (Tabela 06), com 56 bytes de NV SRAM.

Esse RTC permite lidar com a variação da quantidade de dias nos diferentes meses do ano, bem como com anos bissextos. Pode operar com padrão de 12 ou 24 horas. O seu encapsulamento pode ser no padrão DIP ou SO, todos de 8 pinos (MAXIM, 2012).

Este componente utiliza o protocolo de comunicação I2C, sob um barramento serial multi-mestre baseado em duas vias, onde uma é utilizada para envio da temporização (SCL) e a outra para envio dos dados seriais (SDA). Este barramento/protocolo foi desenvolvido pela Philips e tem por finalidade interligar circuitos integrados, e por isso o nome do barramento (Inter Integrated Circuit). Esse barramento no *Arduino* é disponibilizado nos conectores A4 e A5.

Para este trabalho foi escolhido o encapsulamento DIP, pela facilidade que ele proporciona ao desenvolvimento manual dos circuitos.

¹⁹ Disponível em: <<http://www.gpsinformation.org/dale/nmea.htm>>.

²⁰ DATUM é um sistema de referência utilizado para a correlação dos resultados de um levantamento. Existem dois tipos de datums: o vertical e o horizontal. O datum vertical é uma superfície de nível utilizada no referenciamento das altitudes tomadas sobre a superfície terrestre. O datum horizontal, por sua vez, é utilizado no referenciamento das posições tomadas sobre a superfície terrestre. Este último é definido: pelas coordenadas geográficas de um ponto inicial, pela direção da linha entre este ponto inicial e um segundo ponto especificado, e pelas duas dimensões (a e b) que definem o elipsoide utilizado para representação da superfície terrestre.



Figura 17 – RealTime - DS1307.

Fonte: MAXIM (2012).

Tabela 06 – Parâmetros do Real Time DS1307.

Parâmetro	Especificação
Interface de comunicação	I2C
Tensão de alimentação	5 V - DC
Consumo de energia	500 nA
Encapsulamento	DIP ou SO
Suporte a bateria	Sim
Preço no Ebay ²¹	R\$ 1,20

Fonte: MAXIM (2012).

3.4.1.2.5. Reguladores de tensão LM7805 e 78L33

O LM7805 e 78L33 (Figura 18) são reguladores de tensão utilizados para a alimentação de todo o circuito eletrônico. Esses componentes possuem proteção contra sobrecarga e desligam quando estão superaquecidos.

Em seu princípio de funcionamento, eles não necessitam de outros componentes para funcionar, entretanto, recomenda-se a utilização de capacitores na entrada e na saída da alimentação do regulador para ajudar na estabilização da tensão, como ilustra a Figura 19.

As características dos reguladores LM7805 (FAIRCHILD SEMICONDUCTOR, 2001) e 78L33 (STMicroelectronics, 2003) são apresentadas na Tabela 07.

²¹ Data de referência dos preços: 29 de outubro de 2013, site pesquisado: <<http://www.ebay.com>>.



Figura 18 – Reguladores de tensão LM7805 e 78L33.
Fonte: FAIRCHILD SEMICONDUCTOR (2001).

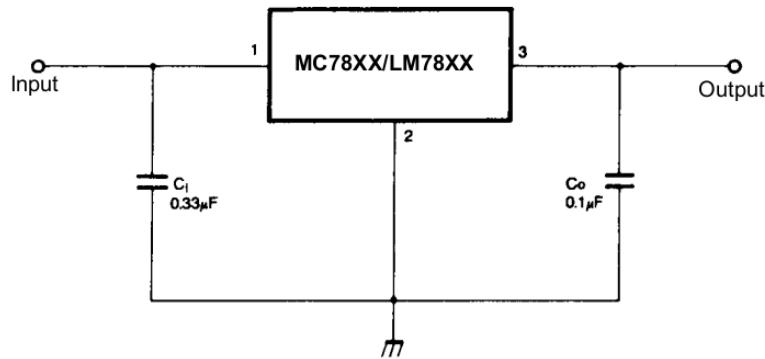


Figura 19 – Diagrama de utilização padrão dos reguladores de tensão.
Fonte: FAIRCHILD SEMICONDUCTOR (2001).

Tabela 07 - Características dos reguladores de tensão LM7805 e 78L03

Parâmetros	Especificação	
	LM7805	78L33
Tensão de entrada	7 V a 35 V - DC	5 V a 9V - DC
Corrente de saída	1 A	100 mA
Tensão de saída	Mínimo: 4.8 V Típico: 5.0 V Máximo: 5.2 V	Mínimo: 3.168 V Típico: 3.3 V Máximo: 3.432 V
Queda de tensão	2 V	1.7 V
Encapsulamento	TO-220 ou D-PAK	TO-92, SO-8 e SOT-92
Preço no Ebay ²²	R\$ 0,50	R\$ 0,50

Fonte: FAIRCHILD SEMICONDUCTOR (2001).

²² Data de referência dos preços: 29 de outubro de 2013, site pesquisado: <<http://www.ebay.com>>.

3.4.1.2.6. Módulo de comunicação *bluetooth* JY-MCU

O JY-MCU versão 1.02 (Figura 20) é um módulo que utiliza um circuito integrado BC417 para converter uma comunicação serial em *bluetooth* (E-GIZMO MECHATRONIX CENTRAL, 2011). Em sua configuração padrão, a velocidade de comunicação serial é 9600 bps e suas características são apresentadas na Tabela 08.

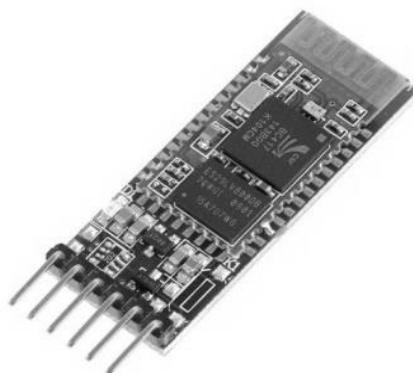


Figura 20 – Módulo de comunicação *bluetooth* JY-MCU.
Fonte: E-GIZMO MECHATRONIX CENTRAL (2011).

Tabela 08 – Características do módulo JY-MCU.

Parâmetro	Especificação
Interface de comunicação	Serial
Versão do <i>Bluetooth</i>	2.0
Encapsulamento do Módulo	DIP
Taxa de Transmissão	1200 – 1382400 bps
Tensão de alimentação	5 V - DC
Consumo	40 mA
Preço no Ebay ²³	R\$ 12,00

Fonte: E-GIZMO MECHATRONIX CENTRAL (2011).

²³ Data de referência dos preços: 29 de outubro de 2013, site pesquisado: <<http://www.ebay.com>>.

3.4.1.3. Placa principal – Arduino adaptada

Em consonância com os requisitos funcionais de *hardware* números 1 e 2, que focam em uma estrutura básica do microcontrolador, simplificada com o objetivo de minimizar o consumo de energia e, ao mesmo tempo, proteção de curto e sobrecarga, foi desenvolvida uma placa do *Arduino* modificada, com seu diagrama elétrico apresentado na Figura 21.

Essa estrutura consiste de duas formas de alimentação: uma interna, por bateria de 9 V e outra externa, que pode vir de um transformador ou bateria, respeitando o limite de 12 V para não superaquecer o regulador de tensão.

O fusível foi utilizado como um protetor de curto e sobrecarga, sendo ele um componente suplementar aos reguladores de tensão que têm também essa função. A porta Aref foi isolada e seu barramento foi conectado após o fusível, para fazer uma ponte ao divisor de tensão que estará na placa da *Shield* do cartão.

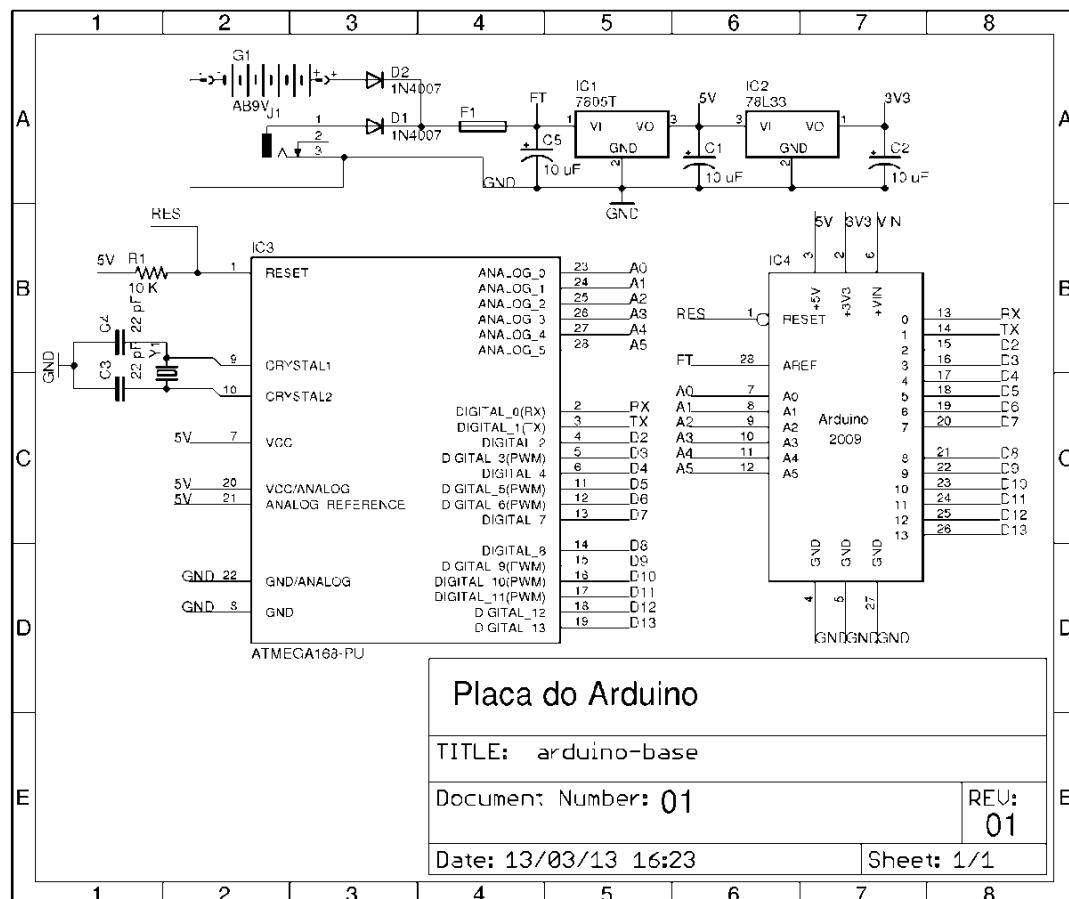


Figura 21 – Diagrama elétrico da placa do *Arduino* adaptada.

Figura 21 Diagrama eláctico
Elab.: PEREIRA, R. B. de O. (2013)

Com base no diagrama elétrico apresentado na Figura 21, elaborou-se o *layout* da placa (Figura 22), com as dimensões de 7,46 cm de altura e 5,66 cm de largura, ficando em consonância com o padrão *Arduino*.

Com o objetivo de minimizar os ruídos, as trilhas foram trabalhadas em ângulos inferiores a 50° e também foi desenvolvida uma malha negativa no circuito para funcionar como uma blindagem do sistema.

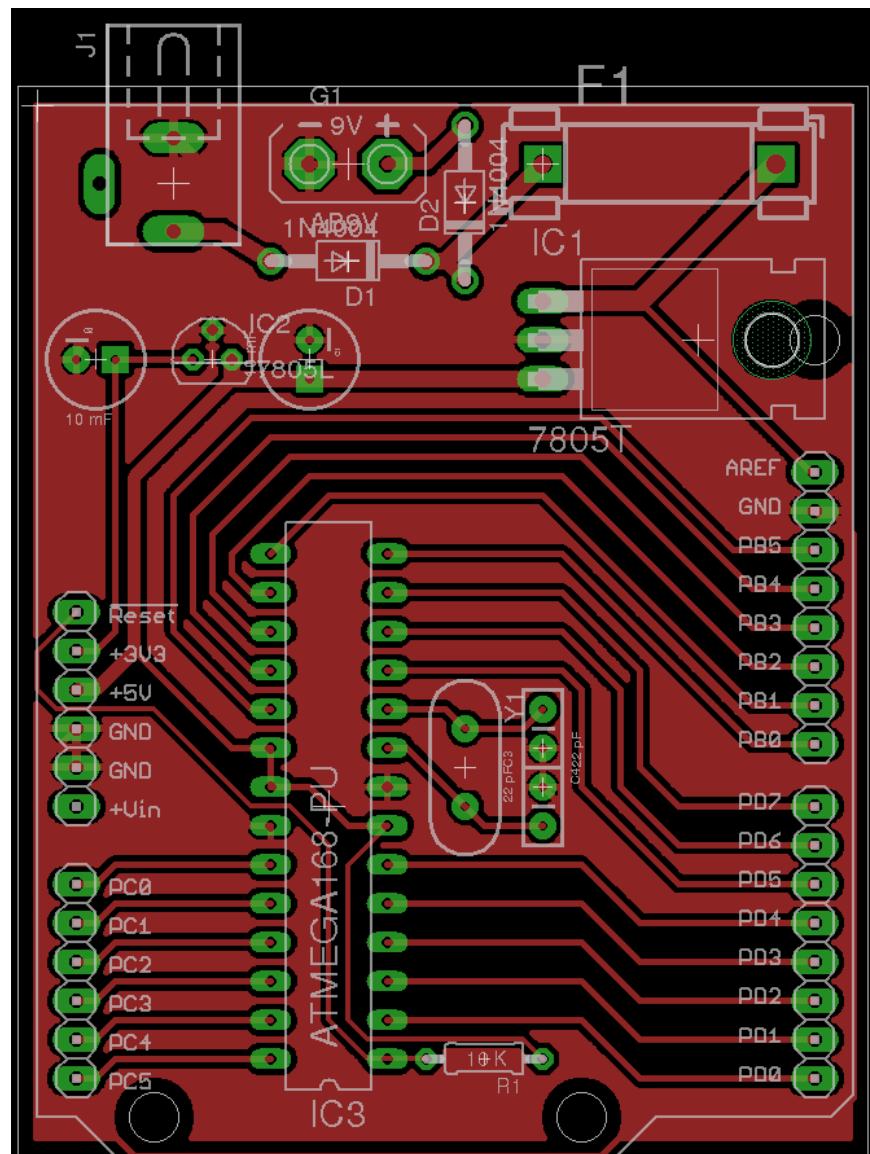


Figura 22 – Layout da placa do Arduino adaptada.
Elab.: PEREIRA, R. B. de O. (2013).

3.4.1.4. Shield do Cartão SD e Realtime

Em consonância com os requisitos funcionais de *hardware* números 3, 4 e 5 que focam em uma estrutura de suporte ao Relógio DS1307, ao cartão de memória SD e a um divisor de tensão para que possa mediar a tensão de alimentação, ajustando a escala para a tolerância do microcontrolador, desenvolveu-se uma *Shield* nos padrões do *Arduino*, com seu diagrama elétrico apresentado na Figura 23.

A estrutura consiste em três divisores de tensão ligados ao barramento SPI para adequar a tensão de 5 V para 3.3 V que é a tensão de trabalho do cartão. O relógio foi ligado ao barramento de comunicação I2C do microcontrolador, bem como adicionado um soquete para conectar uma pilha modelo CR2032 e um divisor de tensão ligado da porta Aref à porta analógica A3.

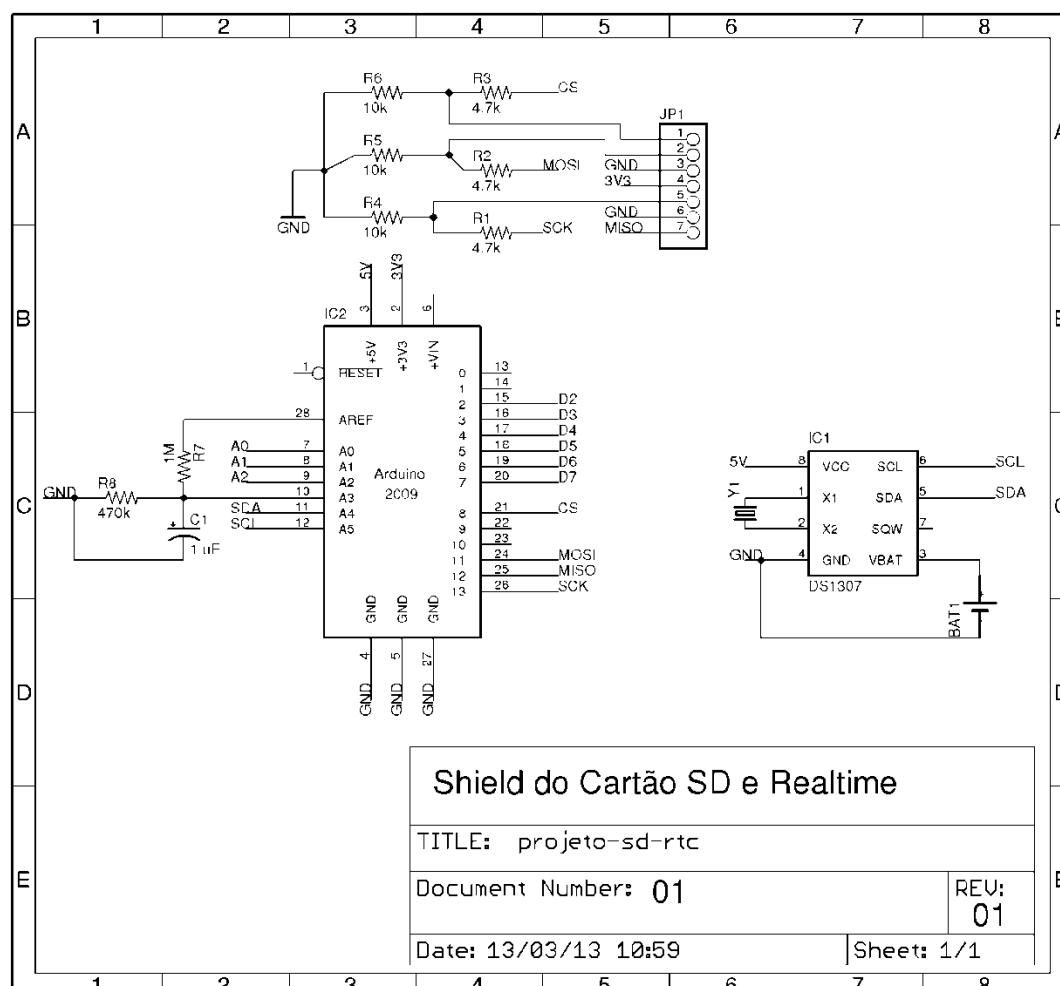


Figura 23 – Diagrama elétrico da Shield do Cartão SD, Realtime e Divisor de tensão para medir a tensão de alimentação.
Elab.: PEREIRA, R. B. de O. (2013).

Com base no diagrama elétrico apresentado na Figura 23, elaborou-se o *layout* da placa (Figura 24), com as dimensões de 7,46 cm de altura e 5,66 cm de largura.

Da mesma forma que na placa clone do *Arduino*, foi desenvolvida uma malha negativa na placa para a blindagem do sistema com objetivo de minimizar as interferências.

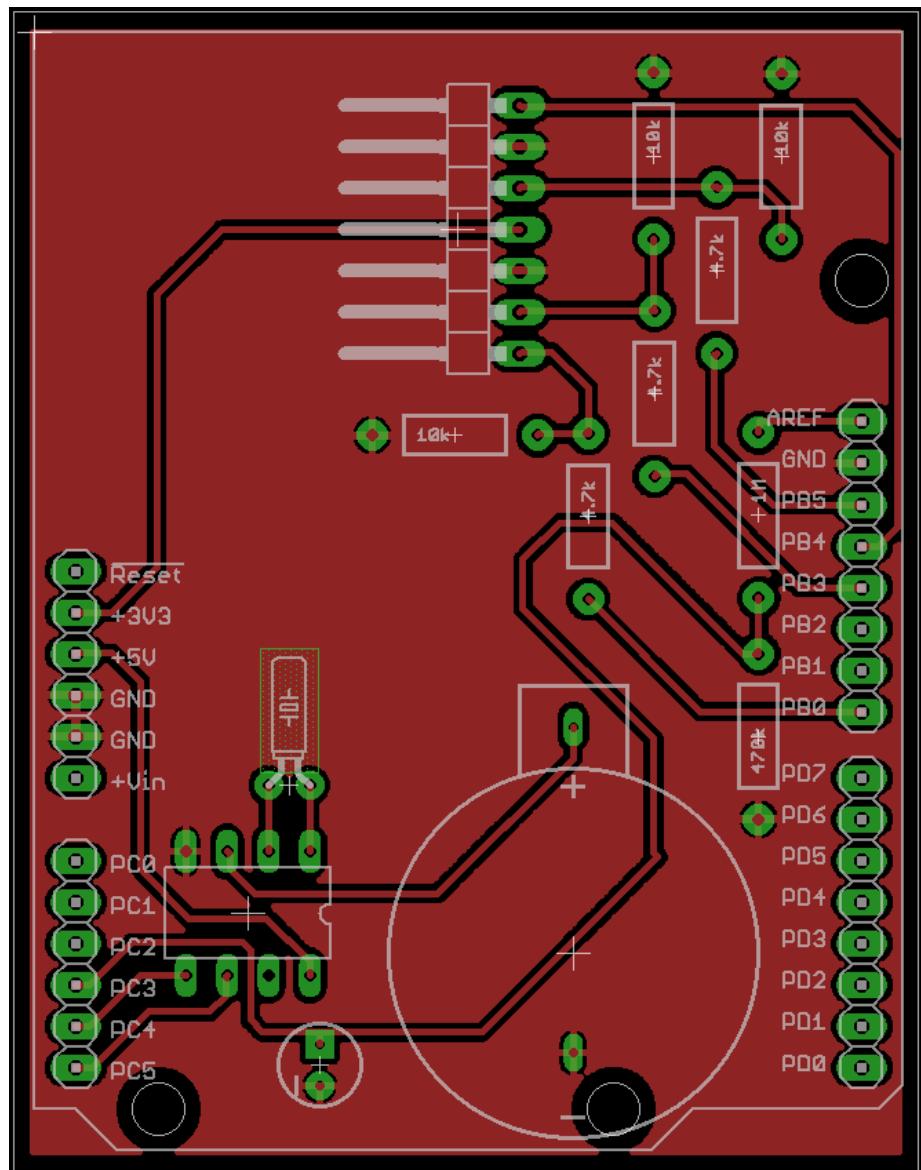


Figura 24 – Shield do cartão SD, Realtime e o divisor de tensão.
Elab.: PEREIRA, R. B. de O. (2013).

3.4.1.5. Shield de comunicação e barramento de conexões

Em harmonia com os requisitos funcionais de hardware números 6 e 7 que focam na conexão de sensores e na comunicação sem fio, desenvolveu-se uma *shield* no padrão *Arduíno*, apresentada na Figura 25. Além de possuir um sistema de conexão dos sensores e comunicação dos transmissores sem fio *bluetooth* e *NRF24L01*.

A estrutura consiste em dois conjuntos de soquetes, o primeiro para ligar o *bluetooth* à porta serial do microcontrolador e a outra para interligar o barramento SPI ao soquete do *NRF24L01*. Os slots de conexões dos sensores são compostos por bornes tipo Kre de 2 e 3 terminais.

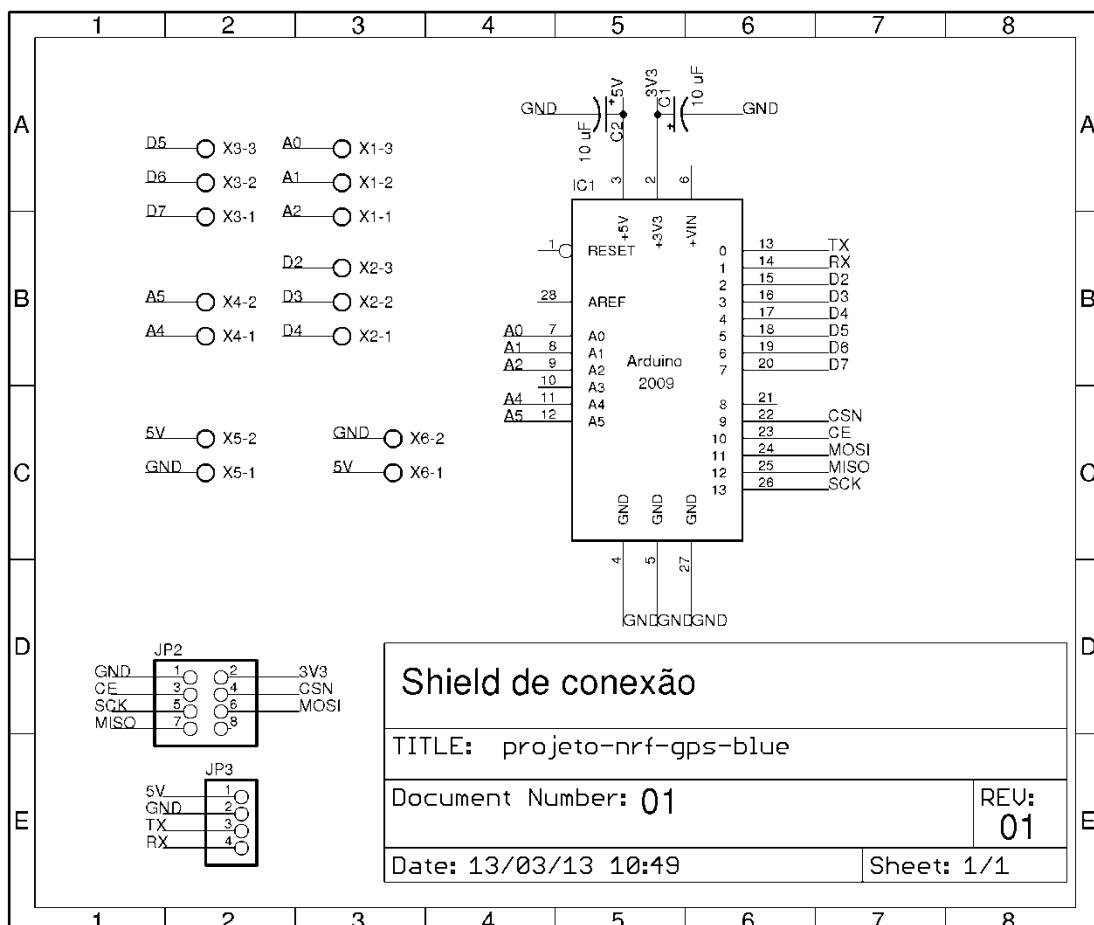


Figura 25 – Diagrama elétrico da shield de conexão nrf e bluetooth.

Elab.: PEREIRA, R. B. de O. (2013).

Com base no diagrama elétrico apresentado na Figura 25, elaborou-se o *layout* da placa, apresentado na Figura 26, com as dimensões de 7,46 cm de altura e 5,66 cm de largura.

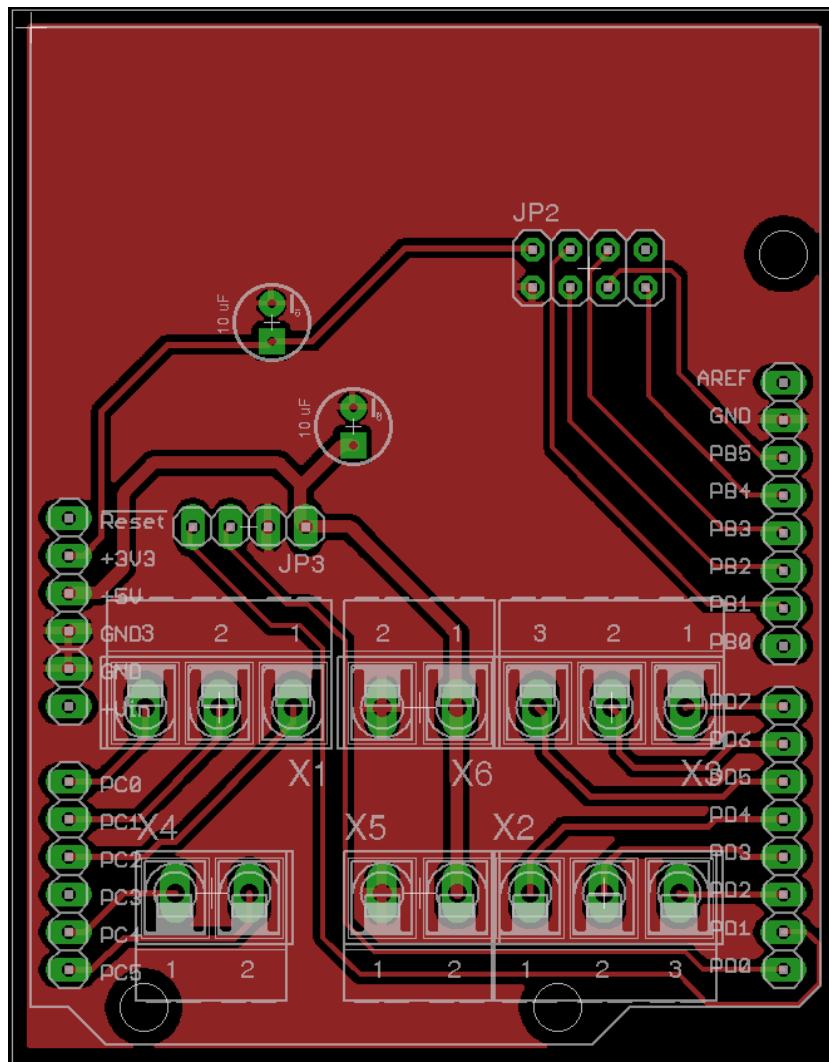


Figura 26 – Layout da shield de conexão nrf e bluetooth.
Elab.: PEREIRA, R. B. de O. (2013).

3.4.1.6. Estrutura de suporte dos sensores

Em consonância com o requisito funcional de *hardware* número 10 para melhor fixação de todos os sensores, bem como a acomodação da fiação que irá conectar os sensores ao equipamento, produziu-se uma estrutura geral que dê suporte a fixação dos mesmos.

Diante do exposto, a construção do protótipo incluiu um conjunto de materiais que fosse relativamente fácil encontrar no mercado local. Dentre os materiais pesquisados, os canos de PVC de 50 mm mostraram-se os mais adequados para atender este requisito. Colaboram para isso o fato da largura interna permitir que facilmente se realizasse a passagem dos fios que conectam aos sensores, da cor do tubo ser branca, absorvendo menos calor, e pela espessura da parede do cano, que possibilita certo grau de rigidez e por possuir um conjunto de conexões como: curvas e luvas, que possibilitam moldá-lo da forma necessária a esse requisito.

A criação da estrutura de suporte ocorreu por dois segmentos distintos: o primeiro segmento é composto por uma estrutura de suporte (Figura 27) e o segundo, por uma estrutura que auxilia na estabilidade e condução do suporte (Figura 28).

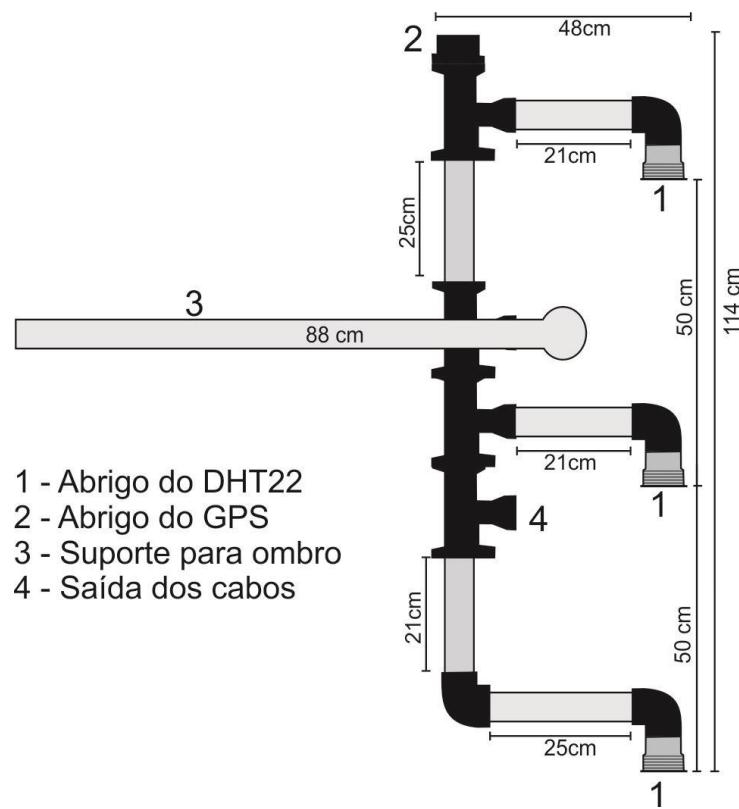


Figura 27 – Visão lateral da estrutura de suporte dos sensores.
Elab.: PEREIRA, R. B. de O. (2013).

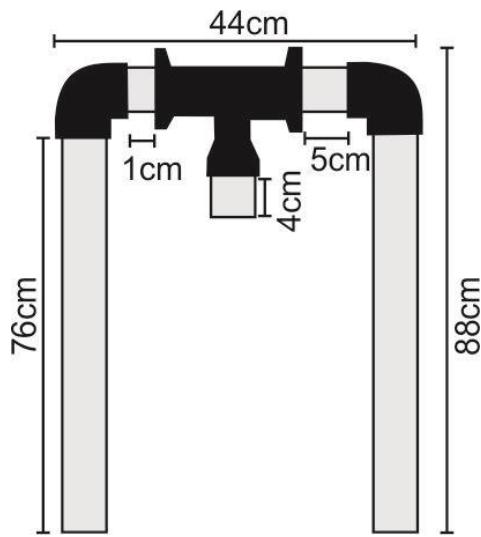


Figura 28 – Visão superior da estrutura que auxilia na estabilidade e condução do suporte.
Elab.: PEREIRA, R. B. de O. (2013).

No primeiro segmento foi utilizada uma estrutura principal de cano, com 4 conectores do tipo T, sendo 2 deles utilizados para a saída de conexão ao sensor, o outro para a saída dos cabos que irão se conectar ao Zoiologger, e o último que é utilizado para conectar a estrutura auxiliar. A distância entre cada saída de sensor é de 50 cm, e a altura da estrutura é de 1,14 m.

O segundo segmento, referente à estrutura auxiliar, para a sua construção utilizou-se uma conexão do tipo T como divisor para as duas curvas que interligaram o suporte aos dois canos de 76 cm que foram colocados no ombro para proporcionar suporte e alinhamento ao cano.

A fim de auxiliar na utilização dessa estrutura, pensou-se em uma mochila que desse suporte à fixação do cano e para que se pudesse utilizá-la como abrigo para o sistema de coleta de dados e para a bateria; como resultado, encontrou-se a mochila apresentada na Figura 29.



Figura 29 – Mochila utilizada como abrigo para os equipamentos e suporte para a estrutura de canos.

Fonte: PEREIRA, R. B. de O. (2013).

3.4.1.7. Suporte e Abrigo do sensor de temperatura e umidade relativa

Em consonância com o requisito funcional de *hardware* número 8 para melhor fixação e suporte, construiu-se uma placa para auxiliar na fixação do sensor de temperatura e umidade DHT-22, de tal forma que proteja o sensor da insolação e de obstáculos. O diagrama elétrico da placa de suporte é apresentado na Figura 30.

Essa estrutura de suporte consiste em uma placa circular, com diâmetro de 49 mm aproximadamente, com quatro furos no centro para a fixação do sensor, e a ela foi adicionado um resistor *pull-up* para aumentar a estabilidade do sinal e um capacitor para reforçar a estabilidade do fornecimento de energia. Para minimizar o ruído foi utilizado cabo blindado e malha de terra comum.

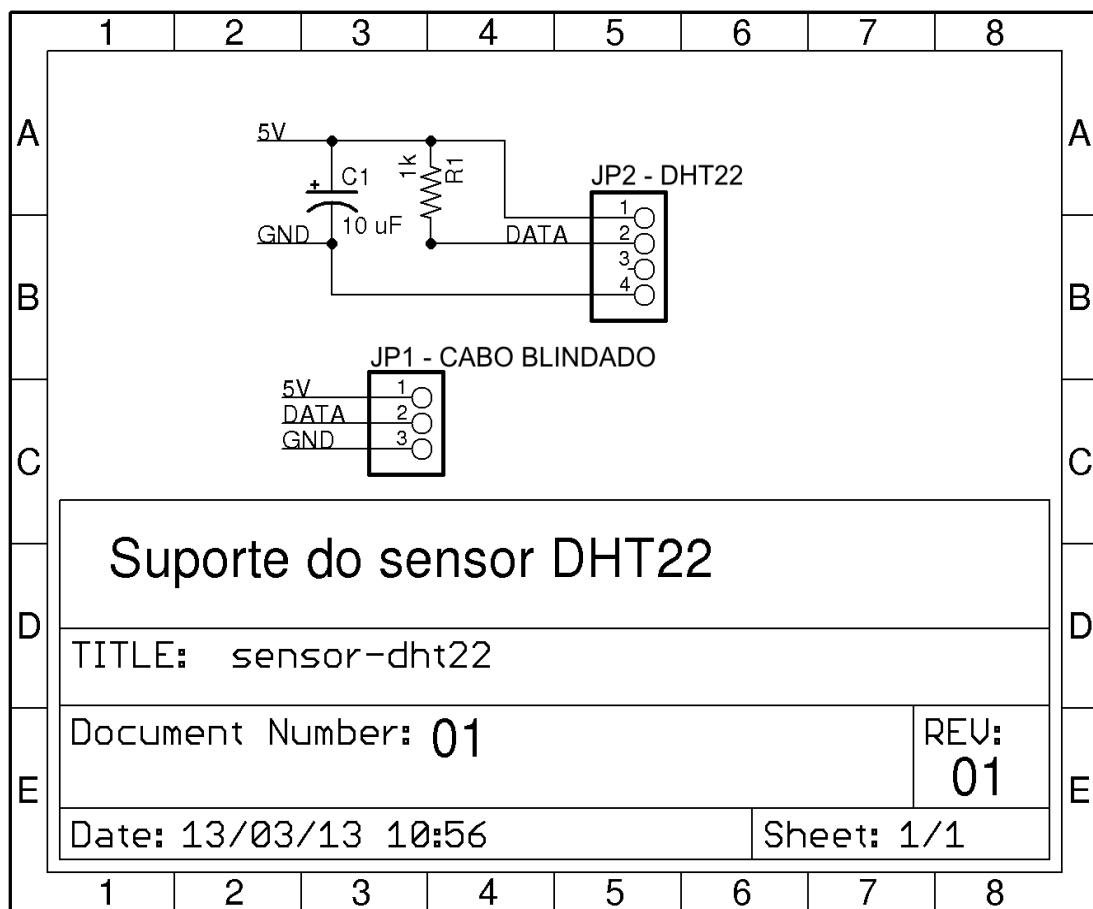


Figura 30 – Diagrama elétrico do suporte do sensor DHT22

Fonte: PEREIRA, R. B. de O. (2013).

Com base no diagrama elétrico apresentado na Figura 30, elaborou-se o *layout* da placa, apresentado na Figura 31, com as dimensões de 7,46 cm de altura e 5,66 cm de largura.

O conector demarcado na Figura 30 como JP1 é referente ao local aonde o cabo blindado que vem do sistema de registro e armazenamento de dados Zoiologger, sendo ele em três vias, uma responsável pela alimentação de 5 V, pelo Ground e pelo barramento de dados.

O sistema de alimentação possui um capacitor (C1) para minimizar os ruídos no sistema de alimentação, no tocando ao barramento de dados tem-se um resistor em modo *pull-up* (R1) a fim de proporcionar maior estabilidade na transmissão dos dados.

O conector JP2 representado na Figura 30 é o local onde é soldado o sensor de temperatura e umidade relativa do ar DHT-22.

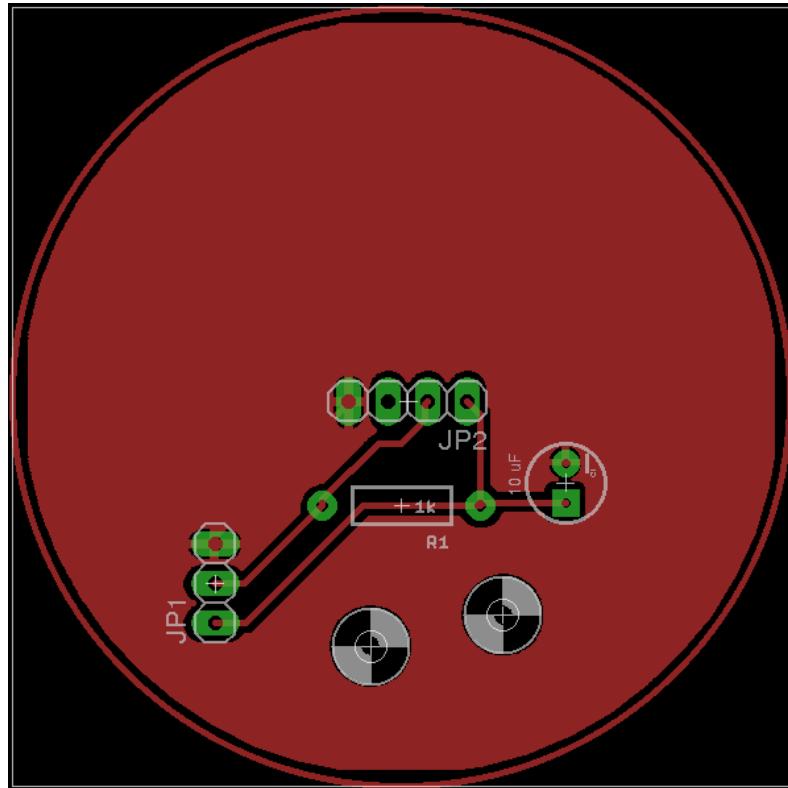


Figura 31 – Layout do suporte do sensor DHT22.

Elab.: PEREIRA, R. B. de O. (2013).

3.4.1.8. Suporte e Abrigo do GPS

Com base no requisito funcional de *hardware* número 9, visando uma melhor fixação e suporte ao GPS, o equipamento deve prover uma estrutura que auxilie na fixação do GPS, bem como forneça um barramento de conexão ao módulo principal. Para esse fim foi construída uma placa cujo diagrama elétrico é apresentado na Figura 32.

Essa estrutura de suporte consiste em uma placa circular, com diâmetro de 49 mm aproximadamente, com um soquete de 6 contatos, com distância de 2 mm entre cada pino, mais dois capacitores e duas resistências *pull-up* para aumentar a

estabilidade do sinal serial de RX e TX, e mais um cabo blindado e malha de terra comum para minimizar ruído.

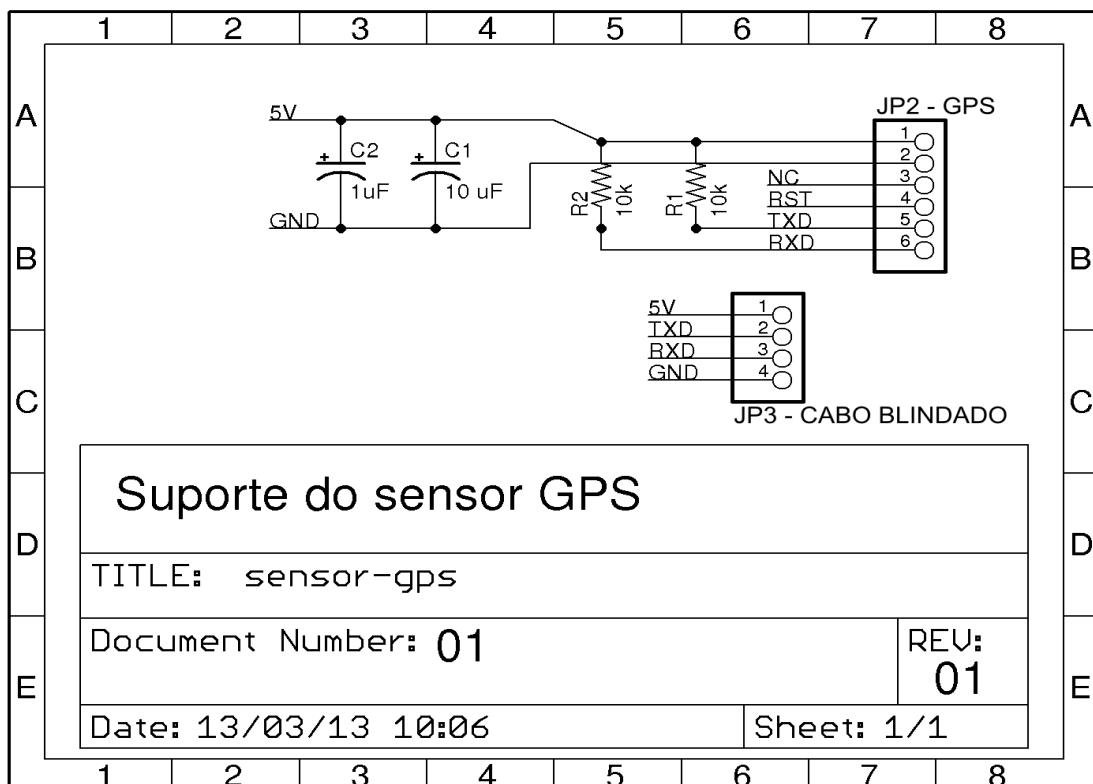


Figura 32 – Diagrama elétrico do suporte do GPS.

Fonte: PEREIRA, R. B. de O. (2013).

O conector JP3 tem por finalidade conectar o cabo duplamente blindado a placa, sendo composto por quatro vias, duas responsáveis pela alimentação do circuito com 5 V e o terra, e os outros dois responsáveis pela transmissão e recepção dos dados por meio do protocolo serial de comunicação (RX e TX).

Os capacitores C1 e C2, apresentados na Figura 32, tem por finalidade filtrar e estabilizar a alimentação vinda do capo duplamente blindado. Para aumentar a estabilidade da comunicação serial, foram ligados dois resistores em modo *pull-up*, um no RX e outro no TX, a fim de aumentar a estabilidade do barramento.

O conector JP2 é utilizado para unir os contatos do GPS ao barramento de comunicação e a alimentação.

Com base no diagrama elétrico apresentado na Figura 32, elaborou-se o *layout* da placa, apresentado na Figura 33, com um diâmetro de 50 mm.

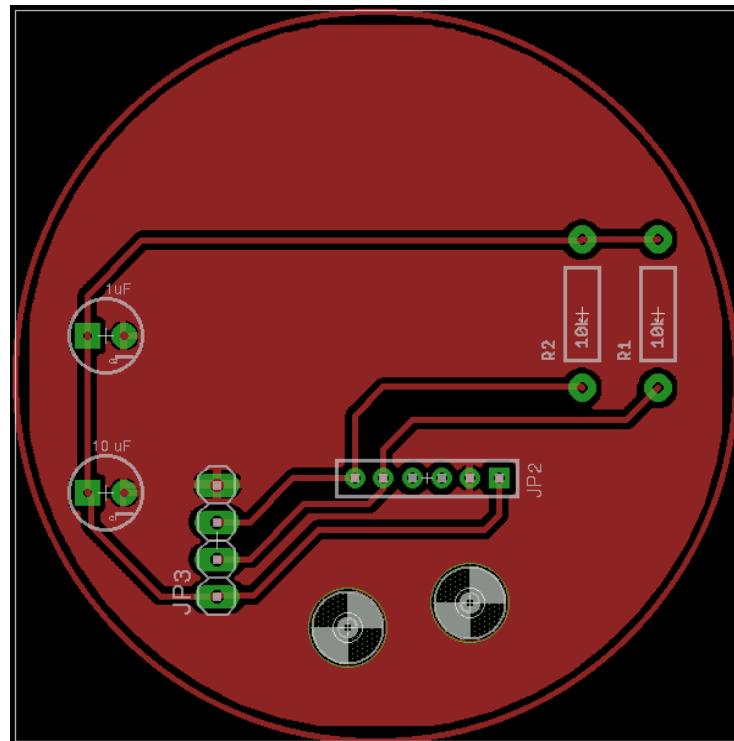


Figura 33 – Layout do suporte do GPS.
Elab.: PEREIRA, R. B. de O. (2013).

3.4.1.9. Alimentação

A alimentação do circuito foi feita por meio de uma bateria selada de alarme residencial da UNIPOWER de 12 V e 5 Ah, com composição à base de Chumbo-Ácida regulada por válvula – VRLA, com um limite máximo de corrente de 1,5 A e uma tensão de flutuação de 13,5 V.

Para a conexão da bateria com o circuito foi feito um cabo com um conector *plug* modelo P4 do tipo macho (Figura 34).



Figura 34 – A bateria e a conexão da bateria por meio do conector P4.
Fonte: PEREIRA, R. B. de O. (2013).

3.4.2. Software do Zoiologger

Esta subseção consiste em todo o processo de análise e fabricação do *firmware* utilizado no microcontrolador, relatando desde as necessidades iniciais, sua respectiva documentação e sua implementação.

Para a elaboração dos diagramas foi utilizada a linguagem de modelagem UML, gerada por meio do *software Astah Community* (novo *jude*) da *Astah* (ASTAH, 2013), versão 6.7.0 para *Linux* e a implementação do código foi pela *IDE* do *Arduino* (ARDUINO.CC, 2013).

3.4.2.1 Levantamento de requisitos

Para a realização do levantamento de requisitos de *software*, realizou-se uma investigação sobre soluções atuais no mercado, resultando na seguinte lista de requisitos funcionais (Tabela 09) e não funcionais (Tabela 10).

Tabela 09 – Requisitos funcionais do *software*.

[RF_001] Deve ler um arquivo texto com os parâmetros de configuração

O sistema deve estar apto a ler um arquivo no cartão de memória *SD* que conterá os parâmetros de configuração do *datalogger*. Esse arquivo deve conter as seguintes características:

1. O arquivo deve se chamar “config.txt”;
2. Os tokens²⁴ do arquivo são identificados pelos caracteres “@” e “#”;
3. Possuir codificação ASCII.

[RF_002] Deve gravar os arquivos no padrão CSV

O sistema deve gravar os dados coletados no padrão CSV, separado por ponto e vírgula contendo o seguinte padrão:

1. O nome do arquivo deve seguir a máscara MMDDHH.csv , onde MM é o mês com dois dígitos, o DD é o dia com dois dígitos e HH a hora com dois dígitos;

[RF_003] Deve possibilitar a especificação da frequência de amostragem

O sistema deve permitir a especificação do tempo de leitura dos dados, expresso em segundos.

[RF_004] Deve possibilitar a coleta de dados georreferenciados

O sistema deve ler os dados recebidos do GPS no padrão NMEA e gravá-los no cartão de memória.

[RF_005] O sistema deve sincronizar a data e hora com a do GPS

O sistema deve ler os dados recebidos do GPS no padrão NMEA e sincronizá-lo com a do *Real Time*.

[RF_006] Deve dar suporte ao sensor DHT22

O sistema deve ler o sensor digital de temperatura e umidade relativa do ar - DHT22.

[RF_007] Deve dar suporte a leitura de sensores que enviam sinais analógicos

O sistema deve estar apto a ler valores fornecidos por sensores analógicos por meio do seu conversor analógico/digital.

(continua)

²⁴ é um segmento de texto ou símbolo que pode ser manipulado por um analisador sintático, que fornece um significado ao texto; em outras palavras, é um conjunto de caracteres (de um alfabeto, por exemplo) com um significado coletivo.

(continuação)

[RF_008] Deve dar suporte a mensuração da tensão da bateria

O sistema, por meio de um divisor de tensão, deve estar apto a ler a voltagem da bateria.

[RF_009] Deve dar suporte a mensuração do consumo do equipamento

O sistema, por meio de um sensor de corrente, deve estar apto a ler o valor fornecido pelo sensor e armazená-lo

[RF_010] Deve sinalizar ao usuário sobre o status de armazenamento da leitura

A cada processo de coleta, o sistema deve gerar um aviso sonoro para indicar a finalização do processo. Deve haver diferentes avisos para sucesso e falha.

[RF_011] Deve sinalizar ao usuário que houve falha na leitura de sensores.

Na presença de um *beep*, a cada falha de leitura de um sensor ele deverá gerar um *beep* indicando a falha na leitura.

[RF_012] Monitoramento dos dados sem fio

Todos os dados coletados, deverão ser transmitidos através do padrão *bluetooth*.

Elab.: PEREIRA, R. B. de O. (2013).

Tabela 10 – Requisitos não funcionais do software

[RNF_001] Arquivos com os dados devem ser aberto em qualquer editor de texto ou planilha eletrônica

Para permitir a portabilidade de programas os arquivos devem ser salvos em um formato texto padrão como o CSV.

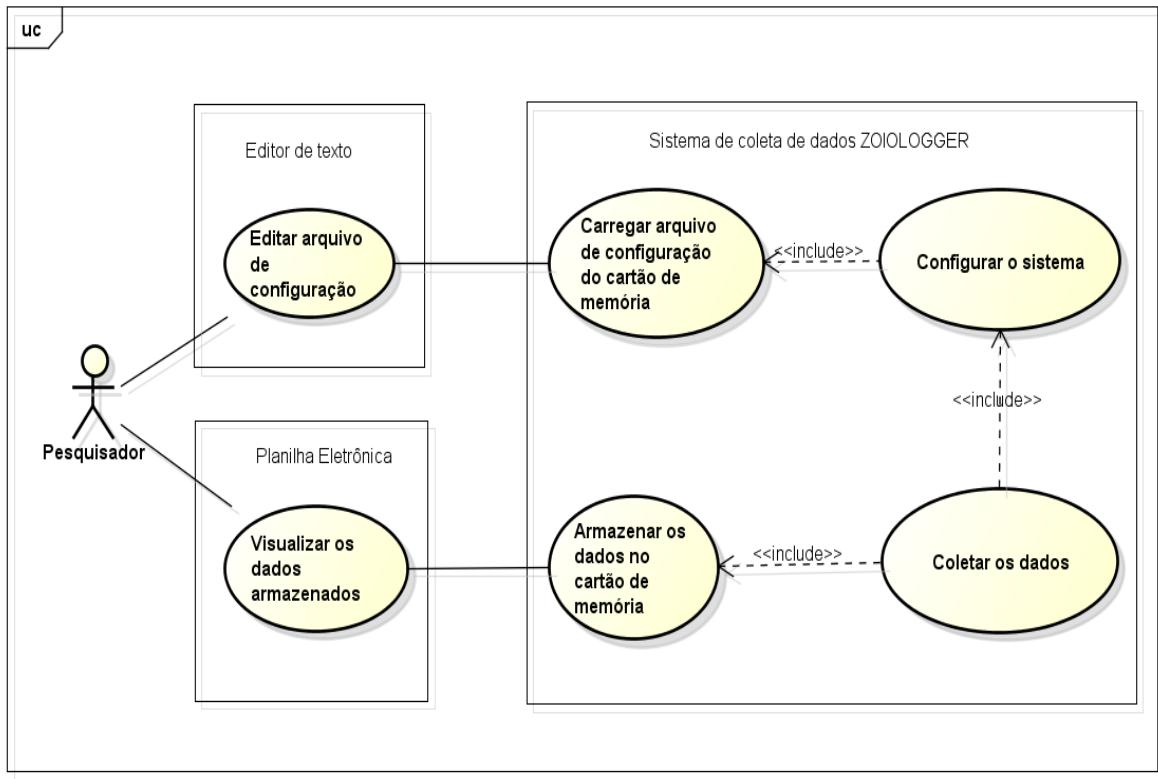
[RNF_002] Facilidade de uso.

Os usuários deverão operar o sistema após algumas horas de treinamento.

Elab.: PEREIRA, R. B. de O. (2013).

3.4.2.2. Diagrama de Caso de Uso

Com base na análise de requisitos de *software* apresentada, foi gerado o diagrama de caso de uso apresentado na Figura 35.



powered by Astah

Figura 35 – Diagrama de Caso de Uso do Zoiologger segundo a notação UML da

Astah Community

Elab.: PEREIRA, R. B. de O. (2013).

3.4.2.3. Diagrama de Atividade

Com base no diagrama de casos de uso (Figura 35) e no levantamento de requisitos de *software* (Tabelas 09 e 10), foi desenvolvido um Diagrama de Atividade apresentado na Figura 36, descrevendo o princípio de funcionamento do sistema em uma visão macro.

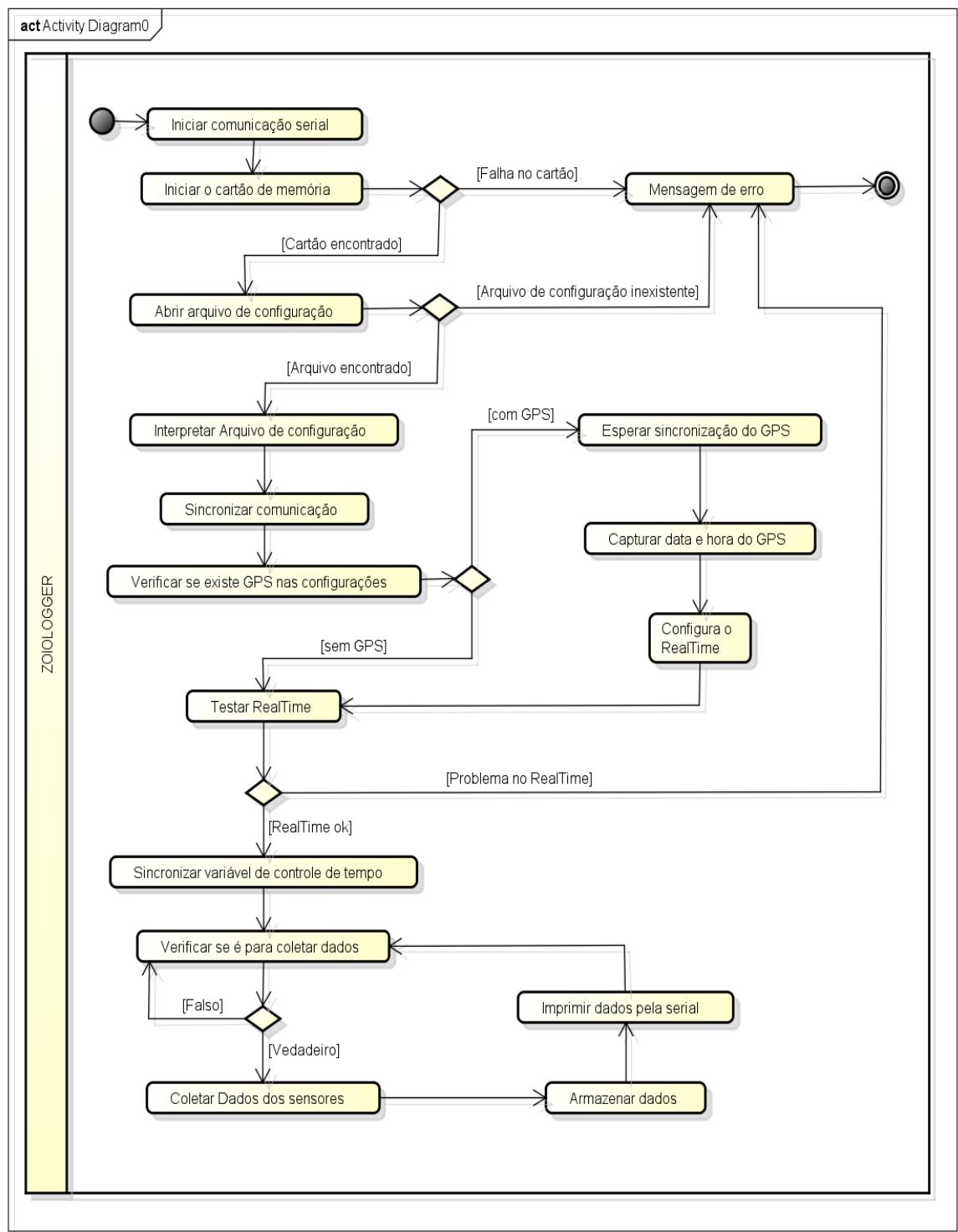


Figura 36 – Diagrama de atividade do Zoiologger segundo a notação UML da Astah Community.

Elab.: PEREIRA, R. B. de O. (2013).

3.4.2.4. Estrutura do arquivo de configuração

Em consonância com os requisitos funcionais do *software*, o sistema deve estar apto a ler um arquivo no cartão de memória *SD* que conterá os parâmetros de configuração do *datalogger*. Esse arquivo é do tipo texto, que pode ser editado por qualquer programa que dê suporte a esse formato.

O arquivo utiliza três *tokens* o “#”, o “@” e o “=”. O primeiro é utilizado para informar ao sistema que a linha é um comentário, o segundo é utilizado para indicar o início de um comando e o terceiro indica o início do parâmetro, conforme apresentado na Figura 37.

```
# Arquivo de configuração
#
# Data 01/03/2012

# Som de Alerta ( 0 - Desativado / 3 ou 5 ou 6 - Porta digital );
@BEEP=5;

#
# Parâmetros Gerais
#

# Taxa de amostra geral em Segundos
@AMOSTRA=5

#
# Sensor
#
# Defini a lista de sensores
# Parâmetro 1 = tipo do sensor (dht22,analo,gps)
# Parâmetro 2 = Porta
@SENSOR=divte,3
@SENSOR=analo,1
@SENSOR=dht22,2
@SENSOR=dht22,3
@SENSOR=dht22,4
@SENSOR=gpskm,6
```

Figura 37 – Exemplo do arquivo de configuração config.txt
Elab.: PEREIRA, R. B. de O. (2013).

Os comandos suportados por esta versão são apresentados na Tabela 11. É válido ressaltar que, segundo os comandos suportados, quando se define um tipo de sensor como “divte” ou “analo” as portas disponíveis para uso serão apenas as analógicas. Para os tipos “dht22” ou “gpskm” as portas disponíveis serão apenas as digitais. Dessa forma, os comandos podem estar associados ao mesmo número de

porta, mas não a mesma porta física, ficando um referente ao conjunto analógico e o outro ao conjunto de portas digitais, conforme a descrição do comando sensor acima.

Tabela 11 – Lista de comandos suportados no arquivo config.txt

Comandos	Descrição
@BEEP	Indica em qual porta PWD se instalara o <i>BEEP</i> . Sintaxe: @BEEP=<porta-pwm> Opções possíveis: 0 – Desativado 3, 5 ou 6 – Portas PWD
@AMOSTRA	Indica o tempo em que os dados serão coletados. Sintaxe: @AMOSTRA=<tempo-em-segundos> OBS: deve-se respeitar o tempo mínimo de amostra de cada sensor.
@SENSOR	Indica a presença de um sensor em uma porta. Sintaxe: @SENSOR=<tipo-do-sensor>,<porta> Tipos de sensores: divte – Divisor de tensão, porta analógica 3. analo – Indica a leitura de uma porta analógica dht22 – Indica a leitura de um sensor DHT22 em uma porta digital gpskm – Indica a leitura de um GPS, porta padrão 5 e 6.

Elab.: PEREIRA, R. B. de O. (2013).

3.4.2.5. Estrutura de armazenamento dos arquivos no cartão

Atendendo aos requisitos funcionais do *software*, os nomes dos arquivos estão estruturados sob a máscara “MMDDHH”.CSV, onde MM representa o mês atual com dois dígitos, o DD é o dia com dois dígitos e HH a hora atual com dois dígitos.

Para o sistema atual, a data e a hora são capturados do GPS durante a inicialização do sistema, conforme o apresentado no diagrama de atividade na Figura 36.

Assim, toda vez que o sistema detectar que é momento de realizar uma leitura dos sensores, ele capturará as informações do relógio *Realtime* pelo barramento I2C e então verificará se o arquivo com esse nome existe. Caso exista, anexará novos dados ao final do arquivo. Caso contrário, criará um novo arquivo e inserirá os dados da coleta nele.

O formato escolhido para a gravação dos arquivos foi CSV separado por ponto-e-vírgula, porque a maioria dos editores e planilhas eletrônicas consegue lidar com este formato, o que dá liberdade ao usuário de qual programa usar.

3.5. MÉTODO UTILIZADO PARA A CONFECÇÃO DAS PLACAS E MONTAGEM

Com base nas dimensões e *layout* de cada placa projetada neste trabalho, foram cortadas placas de fenolite de face simples, respeitando as dimensões e formatos de cada projeto.

O método utilizado para transferir o *layout* foi por termotransferência, o qual se imprimiu por meio de uma impressora a laser, os diagramas em papel do tipo coche.

Após essa etapa, foi efetuada a limpeza das placas de fenolite com água, sabão e lã de aço, e seca com papel toalha. Em seguida foram colocados os diagramas sobre a face cobreada da placa e aplicadas fitas adesivas para que o diagrama não se movesse.

Para a realização do processo de transferência térmica foi utilizado um ferro de passar roupa, que ficou sobre o papel coche durante 5 minutos. Após isso, foi feita a retirada do papel coche e a limpeza do excesso de toner com água.

Para a corrosão da placa demarcada com o toner da termotransferência foi utilizado o percloro de ferro e um pote plástico, no qual a placa ficou submersa por aproximadamente 15 minutos.

Após a corrosão as placas foram limpas com água, sabão e lã de aço e secas com papel toalha a fim de retirar todo toner remanescente. Para a confecção dos

furos, utilizou-se uma furadeira de bancada adaptada, utilizando brocas de 1,0 e 0,8 mm.

A soldagem dos componentes selecionados foi feita por meio de uma estação de solda modelo *Ya Xun 878 D+* com soldas de estanho da marca *Best* com diâmetro de 1,0 e 0,5 mm.

3.6. CALIBRAÇÃO E ESTATÍSTICA

Para avaliar os sensores quanto à sua precisão e acurácia realizou-se a calibração dos sensores de temperatura e umidade relativa do ar com um medidor de estresse térmico TGD-300 (Figura 38) certificado e com uma micro estação meteorológica TP1080WC (Figura 39). A realização da calibração foi feita em uma sala fechada, com controle de iluminação e de acesso de terceiros por meio de uma fechadura mecânica.

Após a coleta e a análise dos dados foram realizadas as devidas correções nos valores obtidos por sensor.



Figura 38 – Medidor de Estresse Térmico TGD-300
Fonte: PEREIRA, R. B. de O. (2013).



Figura 39 – Estação micro meteorológica TP1080WC
Fonte: PEREIRA, R. B. de O. (2013).

Para a calibração e validação dos dados obtidos e armazenados pelo Zoiologger realizou-se uma regressão linear simples e a análise de variância pareada com o suporte do software *IBM SPSS Statistics* da *IBM* versão 21 (IBM, 2012). Para o processamento dos semivariogramas e krigagens utilizou-se o *GS+ Geostatistics for Environmental Sciences* da *Gamma Design Software*, versão 9.0 (GAMMA DESIGN, 2013) e para a plotagem dos mapas utilizou-se o software *Surfer*, versão 10.1.561 da *Golden software* (GOLDEN, 2011).

3.7. MONITORAMENTO DO SISTEMA E AFERIÇÃO DO CONSUMO DE ENERGIA

Para o monitoramento do sistema foi utilizado um celular com o sistema operacional *Android* em sua versão 2.3 por meio do software *Blueterm*²⁵ (PYMASDE.ES, 2013) versão 1.1, disponível gratuitamente na *PlayStore*.

O *Blueterm* é um emulator de terminal VT-100 para comunicação via serial usando um adaptador *bluetooth*. Ele pode ser usado, também, para registrar as informações oriundas dessa comunicação, funcionando como um segundo *logger* do

²⁵ Disponível em: <https://play.google.com/store/apps/details?id=es.pymasde.blueterm&hl=pt_BR>.

sistema. Faz-se necessário ressaltar que para o monitoramento do sistema pode-se utilizar qualquer monitor serial disponível no mercado.

Para o monitoramento, planejou-se um conjunto de testes com base nos requisitos funcionais apresentados e estando eles em conformidade com o diagrama de atividade apresentado na Figura 36.

A bateria de testes consistiu nas seguintes etapas:

- a) Testar o sistema sem o cartão de memória;
- b) Testar o sistema com o cartão de memória, mas sem arquivo de configuração;
- c) Validar o monitoramento do sistema por meio do *bluetooth*.

Para a estimativa do consumo de energia do sistema utilizou-se um multímetro com a função de mensuração da amperagem da marca *Minipa* modelo *APPA ET-2609 True RMS Multimeter*. A coleta dessas informações se deu de duas formas: a primeira, sem a utilização do *bluetooth* e a segunda, com a utilização desse equipamento.

Em cada etapa foi analisada a carga do sistema em modo de descanso e no modo de gravação e transmissão dos dados.

4. RESULTADOS E DISCUSSÃO

"Encontrar erros em programas é duas vezes mais difícil que escrever o código original. Portanto, se você escrever o código da maneira mais esperta que puder então, por definição, você não será esperto o suficiente para encontrar os erros."
(BRIAN W. KERNIGHAN)

4.1. CONFECÇÃO DAS PLACAS E MONTAGEM DO EQUIPAMENTO

4.1.1. Da placa do *Arduino* adaptada

O resultado da confecção da placa do *Arduino* adaptada é apresentado na Figura 40. O custo aproximado dessa placa foi de R\$ 13,70.

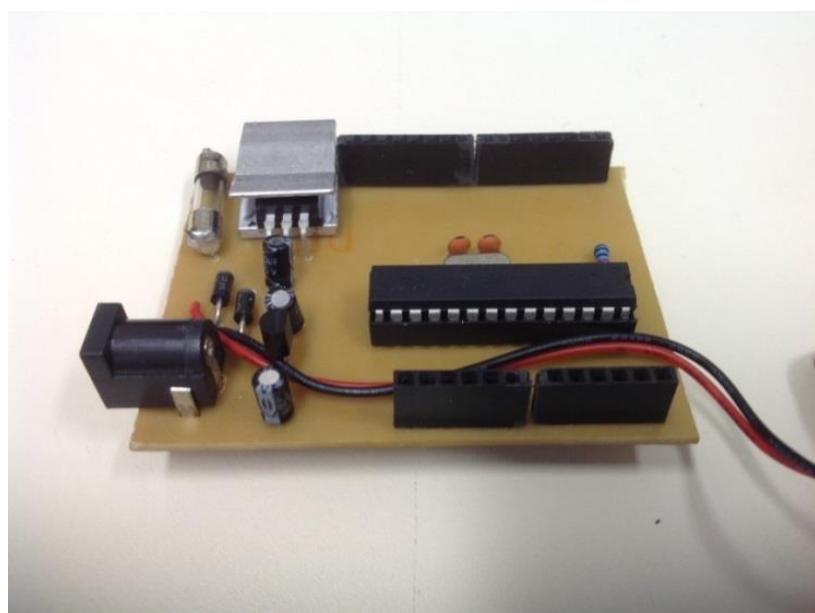


Figura 40 – Placa do *Arduino* adaptado

Fonte: PEREIRA, R. B. de O. (2013).

4.1.2 *Shield* do Cartão SD e *Realtime*

O resultado da confecção desta *shield* é apresentado na Figura 41. O adaptador do cartão SD foi utilizado como soquete do cartão mini-SD, isso facilitou o processo de soldagem. O custo aproximado dessa *shield* já com o cartão de 2GB foi de R\$ 37,20.

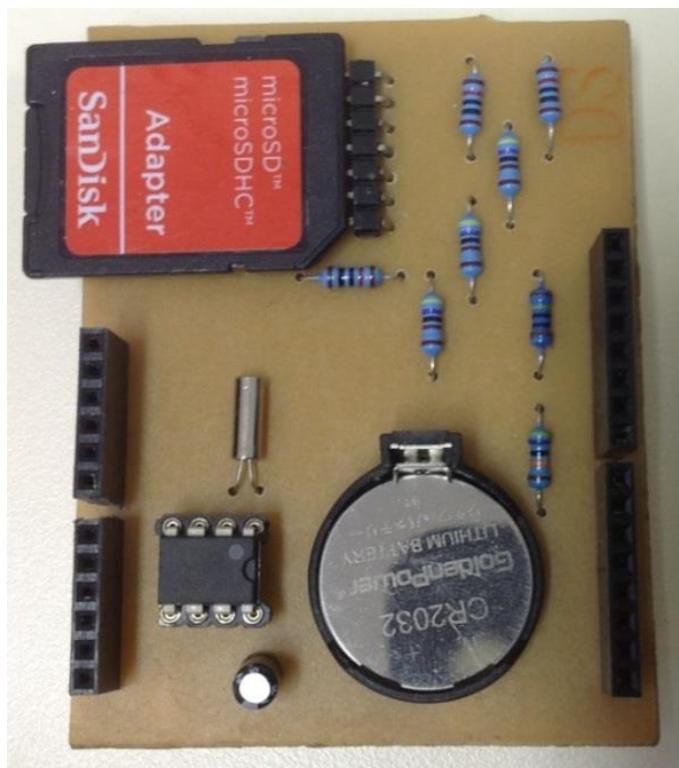


Figura 41 – *Shield* do cartão SD, *Realtime* e divisor de tensão.

Fonte: PEREIRA, R. B. de O. (2013).

4.1.3 *Shield* de comunicação e barramento de conexões

O resultado da confecção da placa é apresentado na Figura 42, com todos os conectores bornes do tipo Kre soldados e preparados para a utilização. O custo aproximado foi de R\$ 16,00.

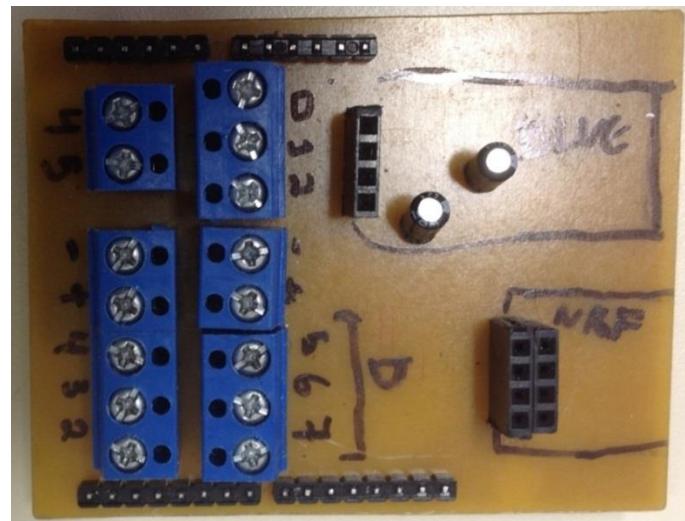


Figura 42 – Shield de conexão nrf e bluetooth.

Fonte: PEREIRA, R. B. de O. (2013).

4.1.4 Estrutura de suporte dos sensores

A estrutura de suporte dos sensores é apresentada na Figura 43, com todos os componentes soldados, montados e preparados para a utilização, tendo sido o seu custo aproximado de R\$ 42,00.

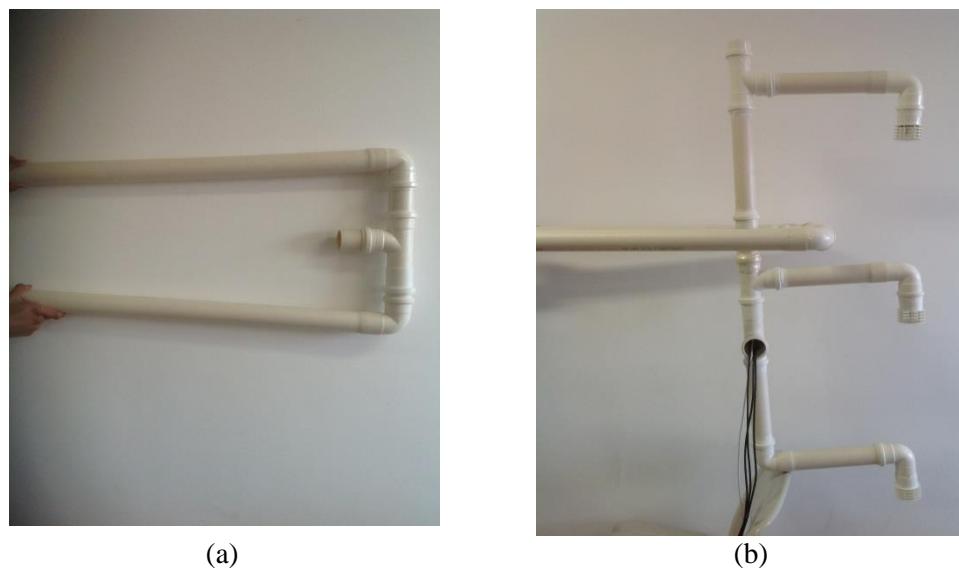


Figura 43 – Estrutura de suporte: (a) visão do suporte de estabilização, (b) visão lateral da estrutura de suporte dos sensores montada.

Fonte: PEREIRA, R. B. de O. (2013).

4.1.5 Placa de suporte ao sensor e seu abrigo térmico

A placa de suporte ao sensor e seu respectivo abrigo térmico são apresentados na Figura 44, com todos os componentes soldados e preparados para a utilização. Para o suporte do sensor foi utilizado um suspiro de esgoto de 50 mm, por possuir acesso à ventilação externa (Figura 44 b). O custo aproximado foi de R\$ 21,50.

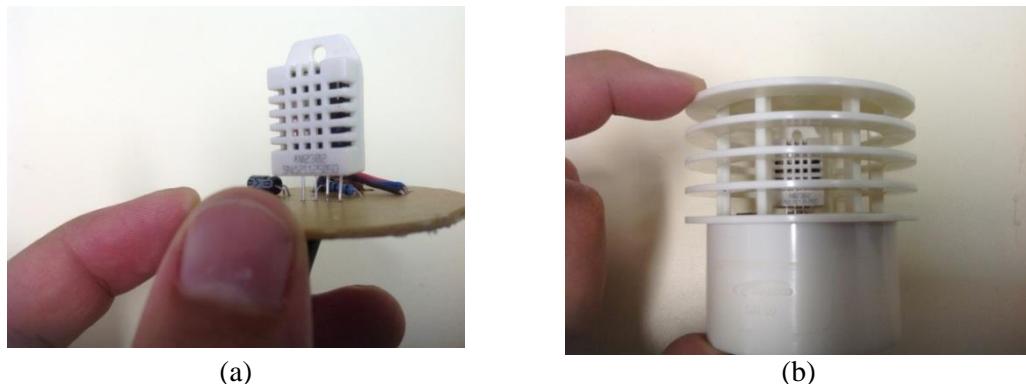


Figura 44 – Suporte do sensor: (a) Placa de suporte ao sensor de temperatura e umidade relativa do ar, (b) Abrigo térmico.

Fonte: PEREIRA, R. B. de O. (2013).

4.1.6 Suporte e Abrigo do GPS

A placa de suporte do GPS é apresentado na Figura 45a, com todos os componentes soldados e preparado para a utilização. Para o suporte do sensor foi utilizado um tampão de esgoto de 50 mm. Para não interferir no sinal do GPS foi feito um corte quadrado do tamanho da antena no tampão de 50 mm e a folga entre eles vedada com cola quente (Figura 45b). O custo aproximado foi de R\$ 101,50.



Figura 45 – Suporte ao módulo do GPS: (a) Placa de suporte, (b) Abrigo do GPS.

Fonte: PEREIRA, R. B. de O. (2013).

4.1.7 Estrutura completa montada

O resultado da montagem da estrutura, da conexão dos sensores e da instalação do sistema é apresentado na Figura 46, com todos os componentes soldados, montados e prontos para a sua utilização. O custo aproximado foi de R\$ 310,90.

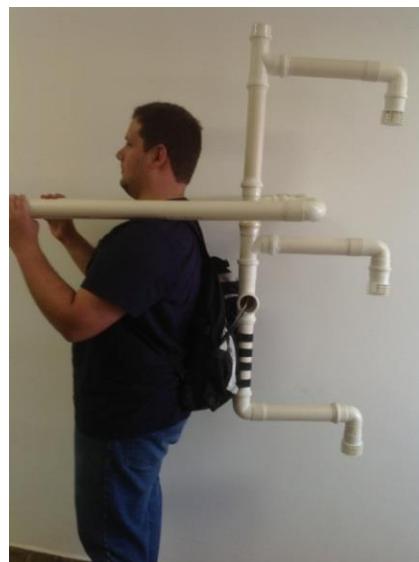


Figura 46 – Estrutura de suporte aos sensores e o equipamento de coleta de dados.
Fonte: PEREIRA, R. B. de O. (2013).

4.2. AVALIAÇÃO DO SISTEMA

Para a validação do sistema verificou-se o consumo de energia, para que se pudesse ter uma noção da demanda energética do sistema, bem como o monitoramento de possíveis erros por *bluetooth* e com os dados obtidos do transecto móvel foram confeccionados semivariogramas e krigeagens para a temperatura e umidade relativa do ar.

4.2.1. Autonomia de energia

Durante o teste de autonomia de energia, o módulo do *Arduino* adaptado sozinho consumiu 16 mA, já o kit montado sem o módulo *bluetooth* obteve uma

média de consumo em modo espera (Figura 47 a) de 57.45 mA e em modo gravação dos dados no cartão de memória SD (Figura 47 b) de 74.87 mA.

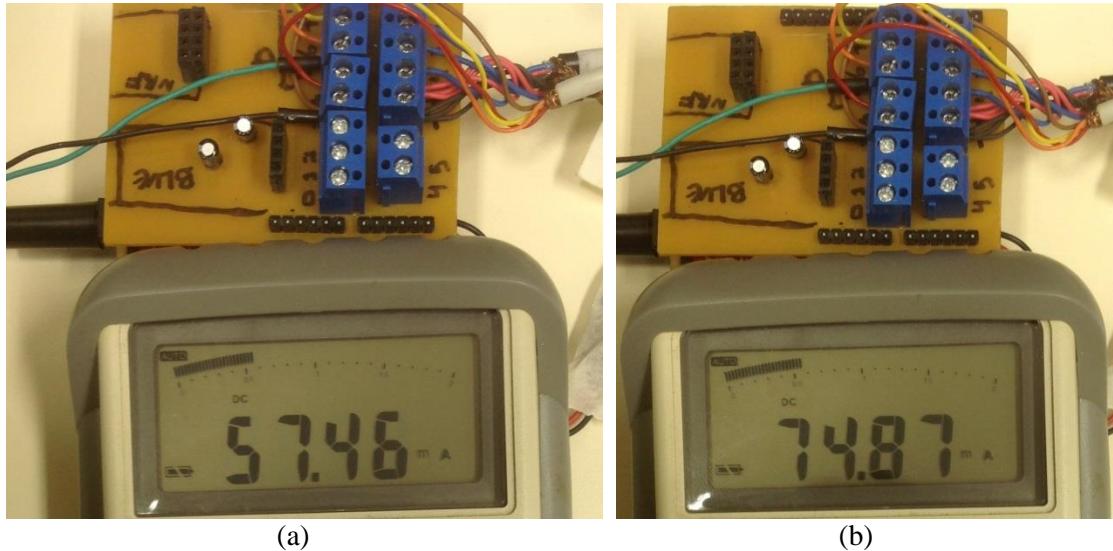


Figura 47 – Aferição do consumo de energia sem o *bluetooth*: (a) Sistema em modo de espera, (b) Sistema gravando dados no cartão *SD*.
fonte: PEREIRA, R. B. de O. (2013).

Nos testes com módulo *bluetooth* obteve-se uma média de consumo em modo espera (Figura 48a) de 81.27 mA e em modo gravação dos dados no cartão de memória SD (Figura 48b) de 106.98 mA.

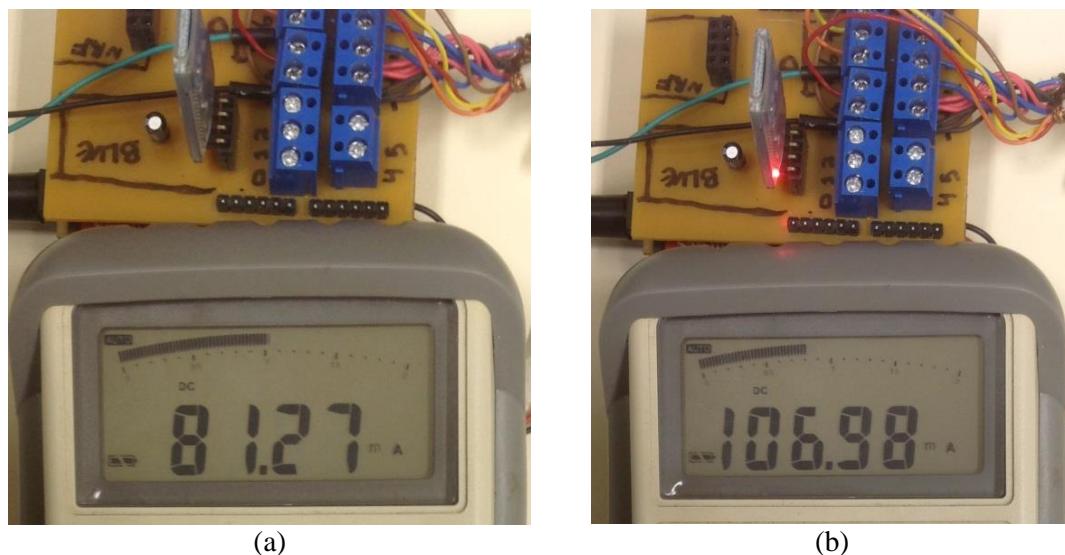


Figura 48 – Aferição do consumo de energia com *bluetooth*: (a) Sistema em modo de espera, (b) Sistema gravando dados no cartão *SD* e transmitindo pelo *bluetooth*
Fonte: PEREIRA, R. B. de O. (2013).

4.2.2. Recursos computacionais consumidos

Devido ao fato de o sistema ter nove portas disponíveis para as conexões dos sensores, sendo elas seis digitais e três analógicas, e tendo o seu espaço alocadas em memória, independente do uso ou não, gerou um consumo total de 1.868 bytes de memória RAM, deixando disponíveis 180 bytes.

Essa postura foi tomada para evitar alocações dinâmicas na memória e não correr o risco de subscrever a memória inicial do sistema, resultando em seu travamento, isso foi feito pelo fato de o *Arduino* não possuir nenhum gerenciador de memória como em um sistema operacional.

Quanto ao espaço de memória flash programável, o sistema ficou em 29.152 bytes, restando disponíveis aproximadamente 3KB. Uma técnica utilizada para minimizar o consumo da memória programável foi transferir as mensagens de informação do sistema para a memória EEPROM do microcontrolador.

O armazenamento dos dados coletados, utilizando todas as portas digitais e analógicas, consumiu aproximadamente 90 KB de memória por hora, o que representa uma autonomia, com um cartão de 2GB, de aproximadamente 2.300 horas de coleta ininterrupta.

Quanto à sobrecarga de processamento do microcontrolador, o mesmo fica grande parte do tempo ocioso, visto que foi adotada a rotina de leitura a cada três segundos. A comunicação com os sensores de temperatura e umidade relativa do ar é relativamente rápida, durando em torno de 80 ms a leitura de cada sensor, a do GPS se dá entorno dos 200 ms, a leitura do relógio em torno de 50 ms e o processo de escrita no cartão em 220 ms, o que totaliza aproximadamente 700 ms.

4.2.3. Monitoramento do sistema

Levando em consideração os testes dispostos na seção 3.8, o primeiro deles é referente à ausência ou um problema no cartão de memória e o seu resultado é apresentado na Figura 49.



Figura 49 – Mensagem do sistema informando a ausência do cartão de memória.

Fonte: PEREIRA, R. B. de O. (2013).

Para o teste de ausência do arquivo de configuração do *datalogger* no cartão de memória SD a mensagem do erro do sistema é apresentada na Figura 50.

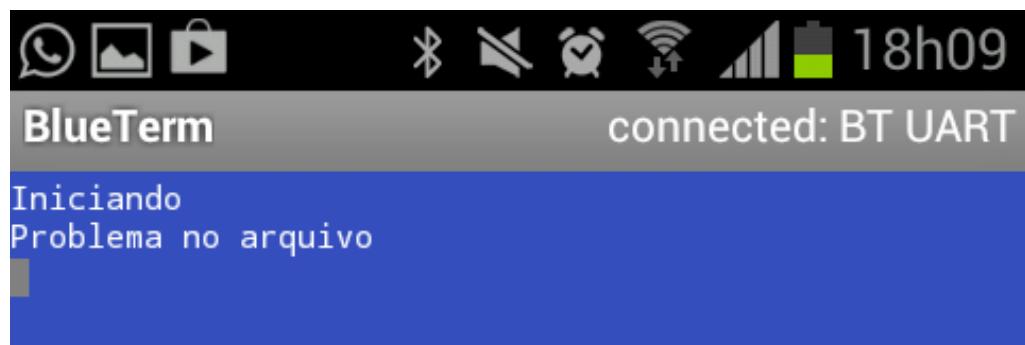


Figura 50 – Mensagem do sistema informando a ausência do arquivo de configuração

Fonte: PEREIRA, R. B. de O. (2013).

4.2.4 Monitoramento em tempo real

O monitoramento em tempo real do sistema por meio do *software Blueterm* durante a coleta de dados é apresentado na Figura 51, contendo nela as informações dos três sensores de temperatura e umidade relativa do ar, a tensão fornecida pela bateria, a leitura do sensor de corrente ligado na porta analógica 1 e as informações do GPS.

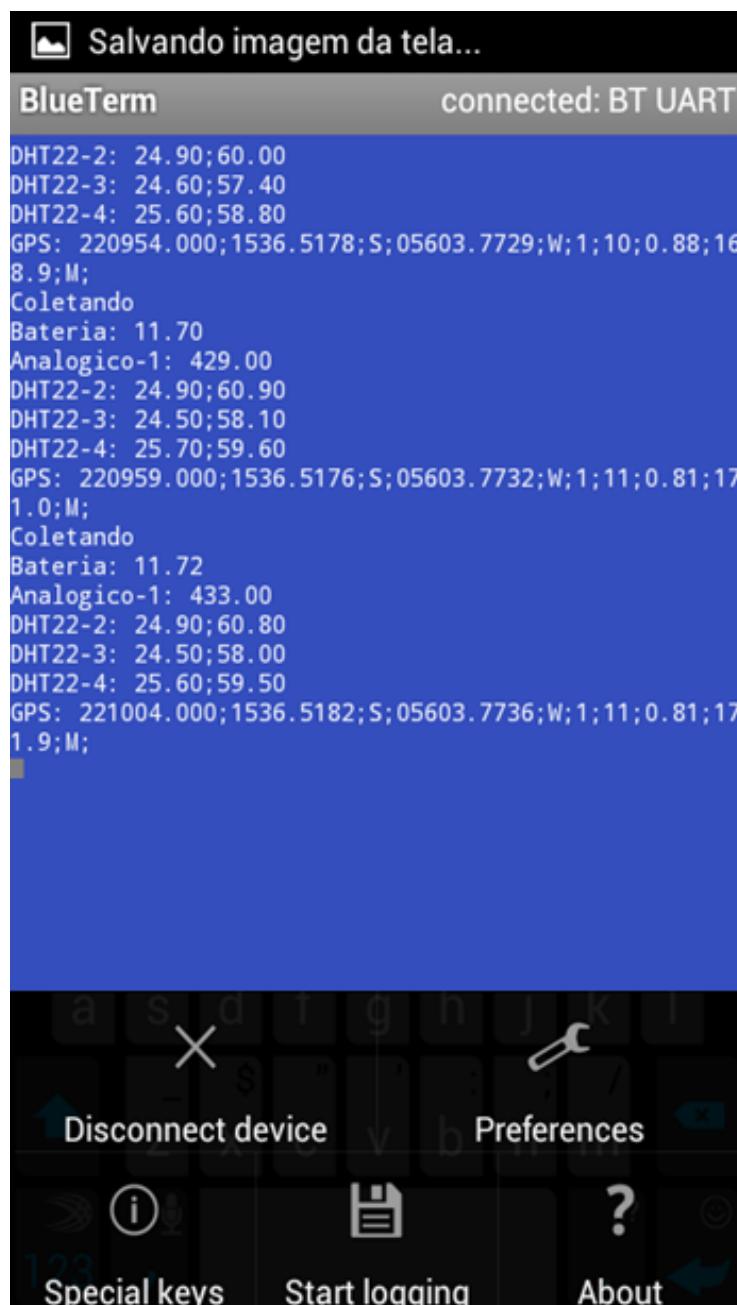


Figura 51 – Tela do software *Blueterm* monitorando uma coleta de dados.

Fonte: PEREIRA, R. B. de O. (2013).

4.3. ESTATÍSTICA E VALIDAÇÃO DO SISTEMA POR MEIO DE UM TRANSECTOR MÓVEL NO CAMPUS DA UFMT

A validação do equipamento foi realizada por meio de um transector móvel no dia 19 de Abril de 2013, seguindo o trajeto apresentado na Figura 11, a fim de verificar a funcionalidade do equipamento, a sensibilidade dos sensores e tentar

detectar possíveis gradientes de temperatura. Este dia foi considerado típico, com céu claro, sem chuva e com ventos reduzidos.

Para uma análise detalhada, o trajeto do transecto móvel foi dividido em 15 trechos para representar o comportamento do entorno (Figura 52) e com base nesses trechos, realizou-se as divisões no mapa de isolinhas resultante da krigagem dos dados. Tal mapa é apresentado na Figura 55, demarcado por uma linha tracejada, contornado inferiormente por uma área indicando a posição e a qual trecho ela se refere.

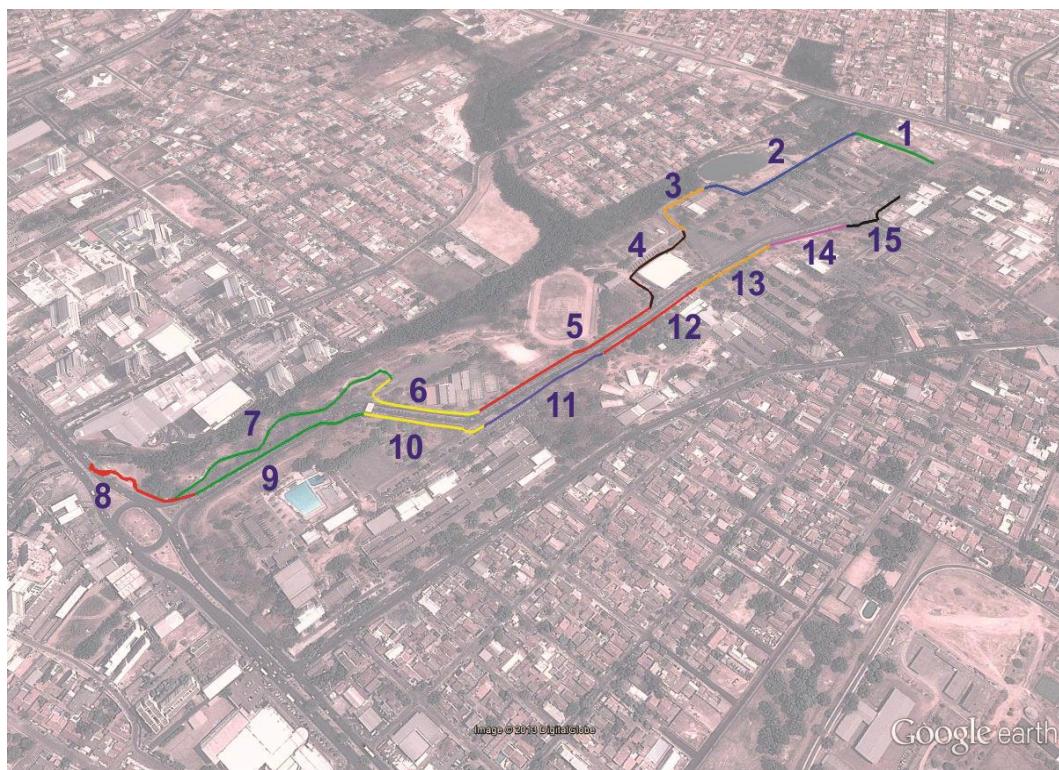


Figura 52 – Trajeto do transecto móvel na UFMT.

Fonte: PEREIRA, R. B. de O. (2013) - adaptado.

A análise de variância pareada foi altamente significativa ($p<1\%$), demandando, para um dos sensores de temperatura e umidade relativa, o ajuste de suas leituras por regressão linear simples ($R^2 = 0,978$). Nos procedimentos estatísticos adotados, os pressupostos de homocedasticidade e aderência dos resíduos a Normal foram avaliados como não significativos ($p>5\%$) pelos testes de Levene e Kolmogorov-Smirnov, respectivamente.

4.3.1. Semivariografia

O semivariograma isotrópico da temperatura do ar pode ser observado na Figura 53, sendo ele um modelo exponencial com coeficiente de determinação de 76.6% ($Co = 0.2220$; $Co + C = 1.1360$; $Ao = 612.00$; $r^2 = 0.766$; $RSS = 0.133$), e o semivariograma isotrópico da umidade relativa do ar pode ser observado na Figura 54, sendo ele um modelo exponencial com coeficiente de determinação de 78.4% ($Co = 1.2900$; $Co + C = 5.6180$; $Ao = 347.00$; $r^2 = 0.784$; $RSS = 2.25$).

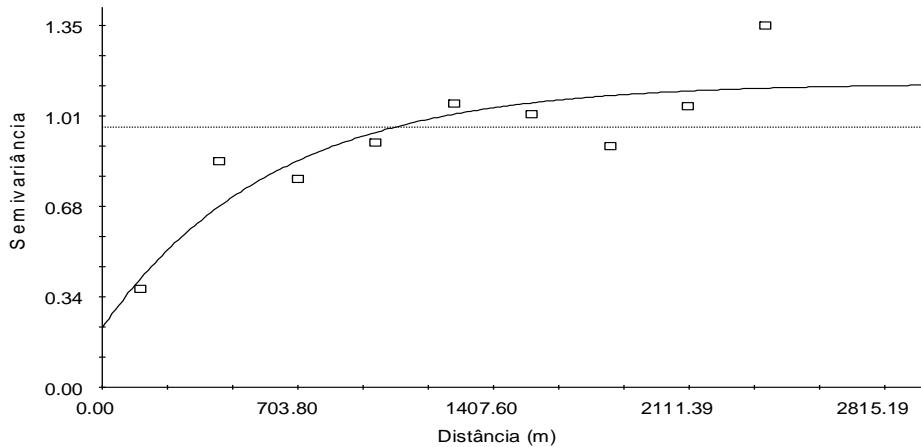


Figura 53 – Semivariograma isotrópico da temperatura do ar.
Elab.: PEREIRA, R. B. de O. (2013).

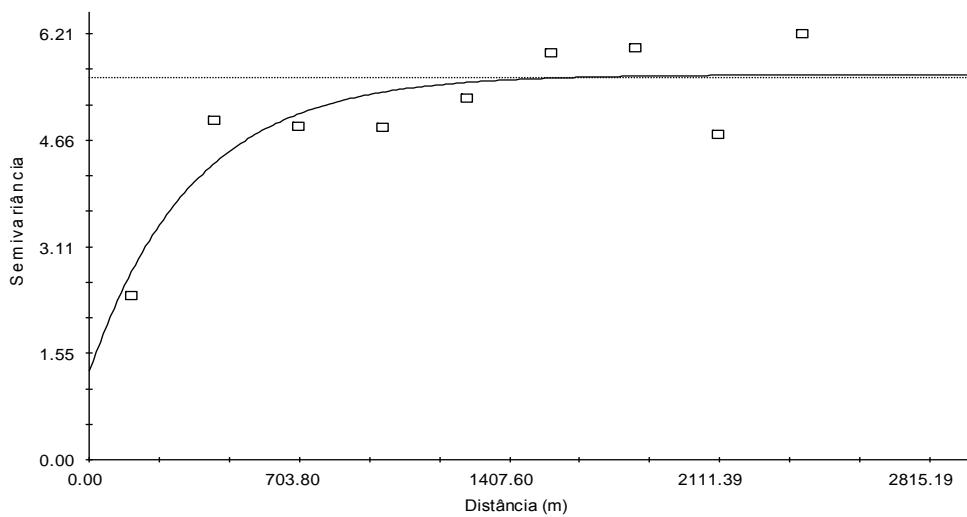
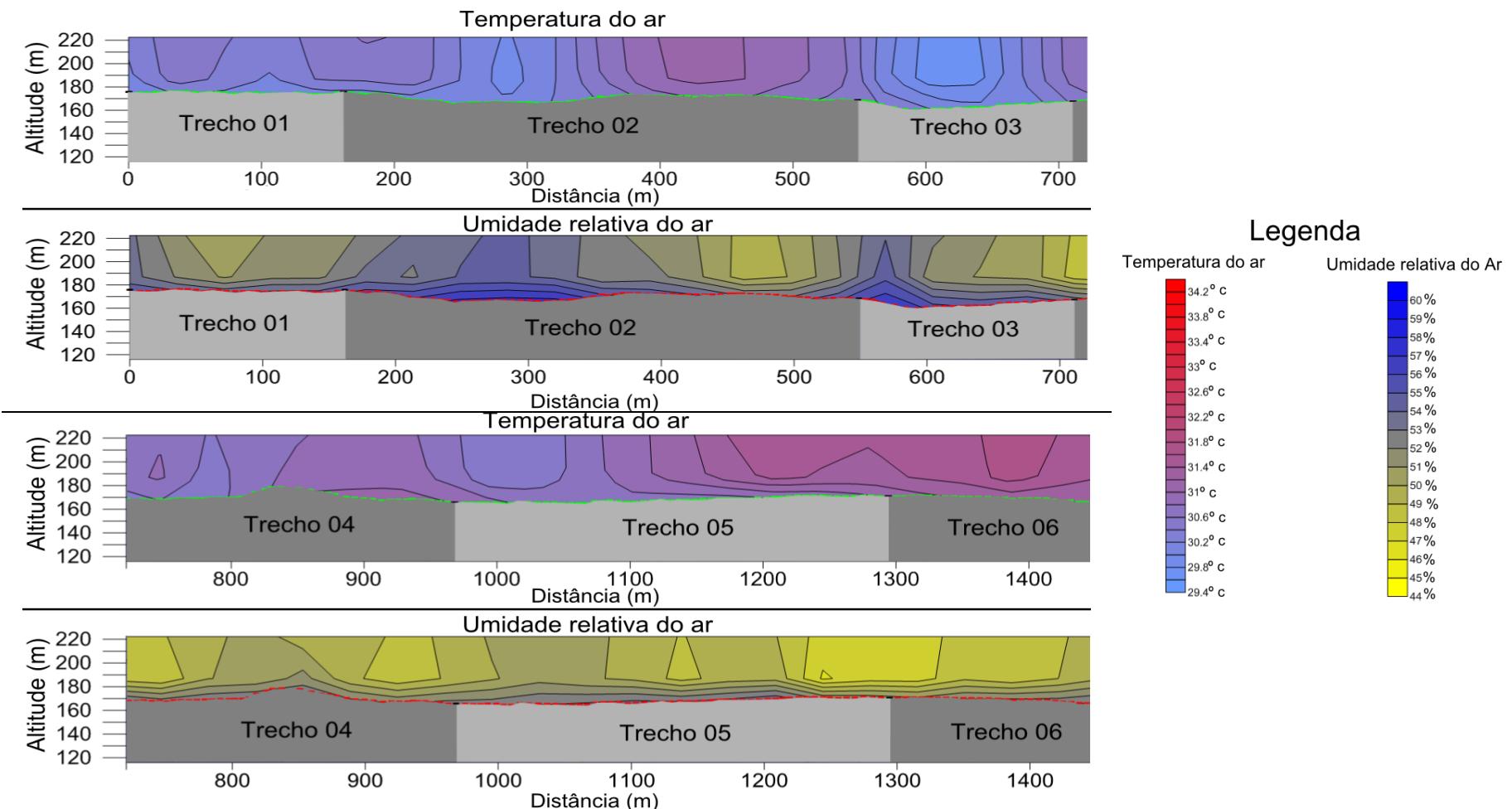
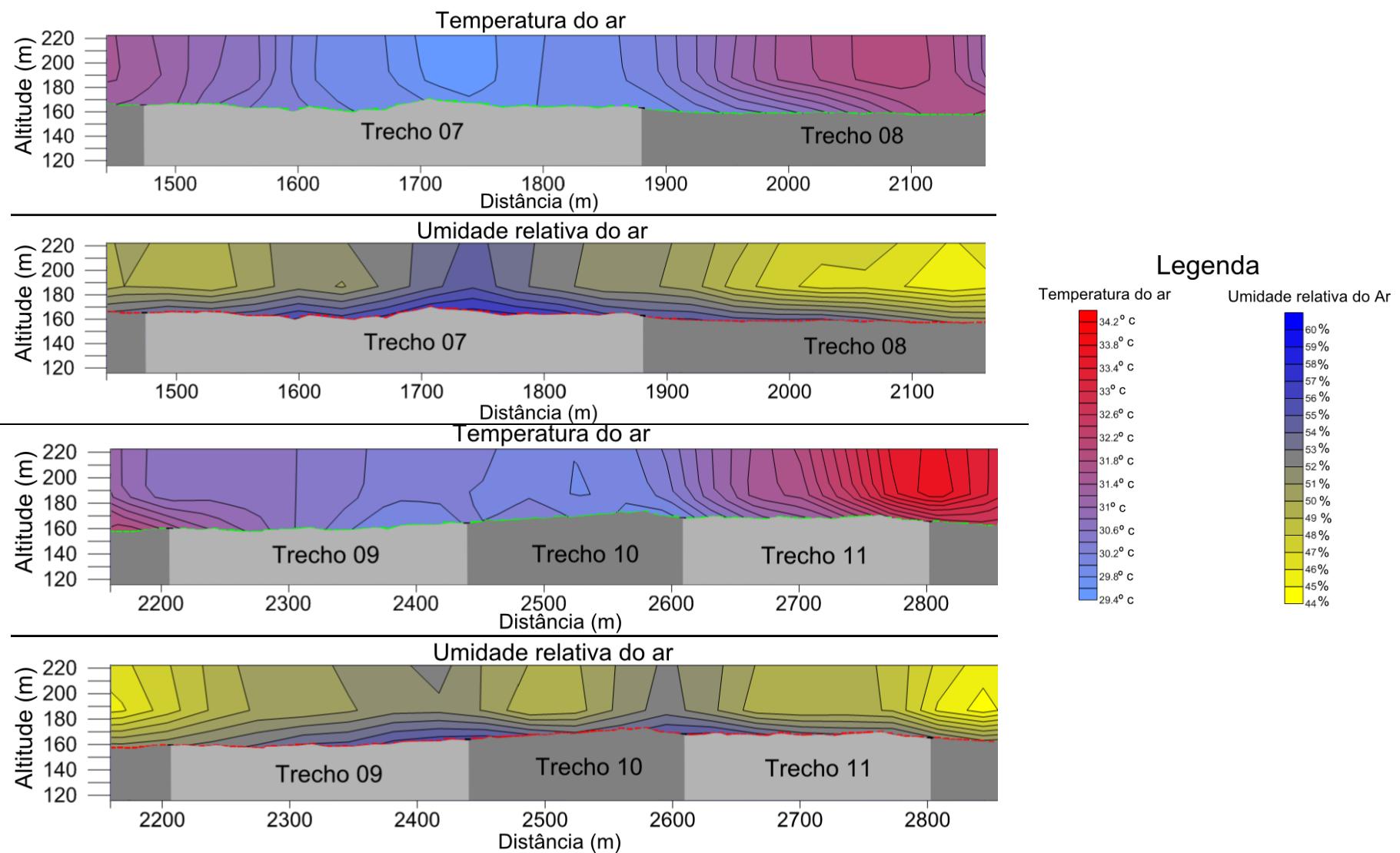


Figura 54 – Semivariograma isotrópico da umidade relativa do ar.
Elab.: PEREIRA, R. B. de O. (2013).

4.3.2. Krigagem e Estudo microclimático do transecto





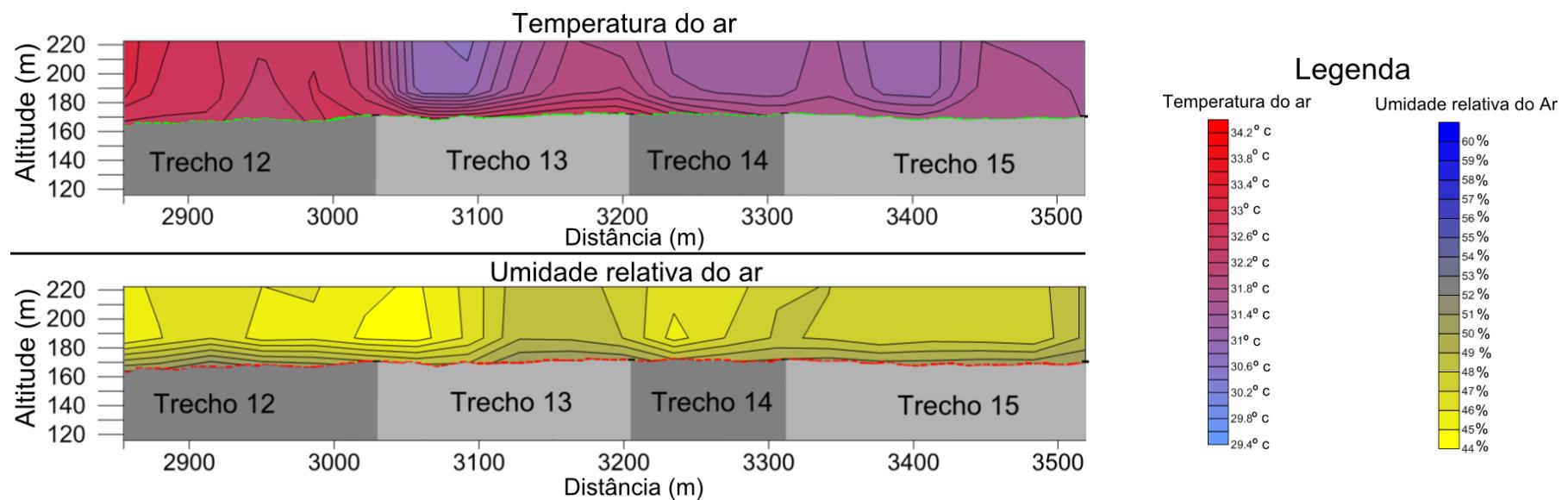


Figura 55 – Mapa de isolinhas da temperatura e umidade relativa do ar nos trajetos.

Elab.: PEREIRA, R. B. de O. (2013).

Os trechos apresentados e numerados na Figura 52 foram analisados conforme as suas características morfológicas, o seu entorno, tipo de pavimento, distância, coordenadas geográficas e seus respectivos comportamentos da temperatura e umidade relativa do ar, com base na Figura 55 apresentado nas páginas 86 a 88.

O primeiro trecho, demarcado pela linha verde e marcado com o número 1 na Figura 52, intitulado percurso inicial, tem aproximadamente 159 metros de comprimento, iniciando no término da pista de caminhada (Figura 56a) indo em direção ao zoológico, ficando ao lado do biotério da UFMT. A pista é de pavimento rígido (calçada de concreto), com um tamanho de aproximadamente 80 cm, contornado em seu lado direito por uma cerca de metal e em seu outro lado, por um bosque (Figura 56b) separado por pavimento flexível do tipo PMF (Pré-misturado a frio). O percurso tem as coordenadas iniciais em $15^{\circ}36'26.36"S$, $56^{\circ} 3'43.61"O$ e término em $15^{\circ}36'23.17"S$, $56^{\circ} 3'47.33"O$.



Figura 56 – Foto do início do primeiro trecho: (a) Visão inicial do percurso, (b) Visão do bosque ao lado do ponto inicial.
Fonte: PEREIRA, R. B. de O. (2013).

Esse trajeto está localizado em uma área que pode ser considerada aberta, possuindo muito acesso à ventilação e quase sem obstrução à dissipação de energia disponível no pavimento para o céu (Figura 57), além de propiciar insolação quase que total durante o fotoperíodo.

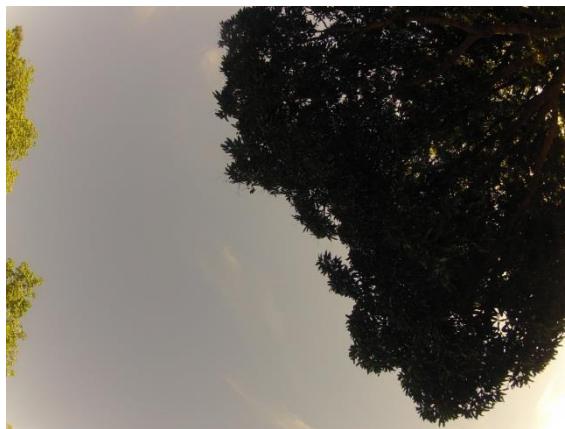


Figura 57 – Visão do céu no ponto inicial.

Fonte: PEREIRA, R. B. de O. (2013).

O segundo trecho, demarcado pela linha azul e marcado com o número 2 na Figura 52, contorna a área de passeio do Zoológico, tem um total aproximado de 388 metros, iniciando na porta de entrada do Zoológico (Figura 58a) indo em direção ao Restaurante Universitário, inserida em um pavimento rígido (calçada de concreto), junto a uma cerca de metal disposta ao seu lado direito, e em seu outro lado uma bosque composta por grama (Figura 58b). O percurso tem o seu início nas coordenadas $15^{\circ}36'23.17"S$, $56^{\circ} 3'47.33"O$ e término nas coordenadas $15^{\circ}36'28.37"S$, $56^{\circ} 3'57.90"O$.



(a)



(b)

Figura 58 – Foto do início do segundo trecho: (a) Portal de entrada do Zoológico, (b) Pista de caminhada do Zoológico.

Fonte: PEREIRA, R. B. de O. (2013).

Esse trajeto está localizado em uma área que pode ser considerada parcialmente aberta, permitindo o acesso à ventilação e quase sem obstrução à

dissipação da energia disponível no pavimento (Figura 59), além de propiciar a insolação quase total durante o fotoperíodo.



Figura 59 – Visão de parte do céu no segundo trecho

Fonte: PEREIRA, R. B. de O. (2013).

O terceiro trecho, demarcado pela linha laranja e marcada com o número 3 na Figura 52, tem um total de aproximadamente 178 metros, iniciando ao lado do Restaurante Universitário, entre a oca do índio e a lagoa do zoológico (Figura 60a), contornando o restaurante universitário pelos fundos e saindo na frente do ginásio de esportes. Todo o percurso é composto por pavimento flexível do tipo pré-misturado a frio (Figura 60b), tendo o seu início nas coordenadas 15°36'23.17"S, 56° 3'47.33"O e término nas coordenadas 15°36'28.37"S, 56° 3'57.90"O.



(a)



(b)

Figura 60 – Foto do início do terceiro trecho: (a) Pista que contorna a oca do índio, (b) Visão da pista atrás do R.U.

Fonte: PEREIRA, R. B. de O. (2013).

Esse trajeto está localizado ao redor de uma área de mata ciliar, parcialmente coberta por uma estrutura de metal (Figura 61), tendo do seu lado esquerdo o prédio

do Restaurante Universitário, possui acesso parcial à ventilação e obstruções à dissipação de energia do pavimento (Figura 61), resultando na obstrução a insolação durante o fotoperíodo.



Figura 61 – Visão parcial da cobertura na parte de trás do restaurante universitário.
Fonte: PEREIRA, R. B. de O. (2013).

O quarto trecho, referente ao percurso do ginásio de esportes, demarcado pela linha marrom e numerado de 4 na Figura 52, tem um total aproximado de 247 metros, iniciando na frente do Ginásio de Esporte e ao lado do Restaurante Universitário, indo em direção à avenida principal da UFMT. O percurso inicial deste trecho é composto por pavimento rígido (calçada de concreto), com tamanho aproximado de 70 cm, cercado por um pequeno bosque com árvores lenhosas, e parte atravessando uma estrutura de concreto armado (Figura 62a e 62b), tendo o seu início nas coordenadas 15°36'32.74"S, 56° 3'59.62"O e término nas coordenadas 15°36'38.72"S, 56° 4'2.08"O.



(a)



(b)

Figura 62 – Foto do quarto trecho: (a) Visão parcial do bosque na frente do ginásio, (b) Visão dos fundos do ginásio indo em direção ao ponto de ônibus.
Fonte: PEREIRA, R. B. de O. (2013).

Esse trajeto está em trecho de área construída, ao lado de um campo de futebol, e parcialmente coberto (Figura 63), tendo restrições à ventilação e obstruções à dissipação de energia do pavimento, resultando também em obstruções à insolação durante o fotoperíodo.



Figura 63 – Fotos da cobertura de parte do quarto trecho.
Fonte: PEREIRA, R. B. de O. (2013).

O quinto trecho, às margens da avenida principal da UFMT, passando em frente ao campo de futebol, demarcado pela linha vermelha e numerado de 5 na Figura 52, tem um total aproximado de 328 metros, iniciando na saída lateral do ginásio, indo em direção à quadra da UFMT. O percurso inicial deste trecho é composto por pavimento rígido (calçada de concreto), circundado por um pequeno bosque (Figura 64a), a maior parte do trecho é composta por paralelepípedo (Figura 64b), cercado ao lado direito por tapumes de zinco e ao lado esquerdo composto por pavimento flexível do tipo pré-misturado a frio; o trecho tem início nas coordenadas $15^{\circ}36'38.56"S, 56^{\circ} 4'1.99"O$ e término em $15^{\circ}36'45.41"S, 56^{\circ} 4'10.52"O$.



Figura 64 – Foto da parte inicial do quinto trecho: (a) Bosque atrás do ponto de ônibus do Ginásio, (b) Campo de Futebol fechado para reforma.
Fonte: PEREIRA, R. B. de O. (2013).

Esse trajeto está em trecho de avenida, ao lado de um campo de futebol (Figura 64b), possui acesso parcialmente alto de ventilação e dissipação de energia do pavimento, recebendo insolação durante o fotoperíodo (Figura 65).



Figura 65 – Visão de parte do céu no quinto trecho.

Fonte: PEREIRA, R. B. de O. (2013).

O sexto trecho, às margens da avenida principal da UFMT, vai em direção à primeira guarita, próximo à Avenida Fernando Corrêa da Costa, demarcado pela linha amarela e numerada de 6 na Figura 52, tem um total aproximado de 197 metros, seu trecho é composto por paralelepípedo (Figura 66a), cercado ao lado direito por alguns tapumes de alumínio (Figura 66b), e ao lado esquerdo composto por pavimento flexível do tipo pré-misturado a frio; o trecho tem início nas coordenadas $15^{\circ}36'45.41"S$, $56^{\circ} 4'10.52"O$ e término em $15^{\circ}36'43.44"S$, $56^{\circ} 4'14.86"O$.



Figura 66 – Foto da parte inicial do sexto trecho: (a) Pista de paralelepípedos ao lado da quadra de esportes, (b) Bosque na frente da guarita e ao lado da pista.
Fonte: PEREIRA, R. B. de O. (2013).

O trajeto está em trecho de avenida, ao lado da quadra de esportes da UFMT. Possui acesso à ventilação e dissipação de energia do pavimento, recebendo insolação durante o fotoperíodo (Figura 67).

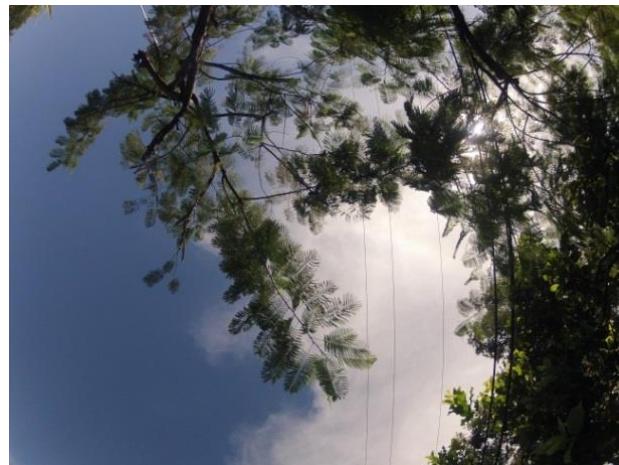


Figura 67 – Visão de parte do céu no sexto trecho.
Fonte: PEREIRA, R. B. de O. (2013).

O sétimo trecho, situado no bosque de caminhada da UFMT, demarcado pela linha verde e numerada de 7 na Figura 52, tem um total de aproximadamente 380 metros, iniciando na saída da guarita e indo em direção à Avenida Fernando Corrêa da Costa. O percurso deste trecho é composto por pavimento rígido (calçada de concreto), circundado por espécies de vegetação lenhosas (Figura 68a) de grande porte, além de uma vegetação nativa rasteira (Figura 68b). O trecho tem início nas coordenadas $15^{\circ}36'43.44"S$, $56^{\circ} 4'14.86"O$ e término em $15^{\circ}36'49.85"S$, $56^{\circ} 4'23.33"O$.



Figura 68 – Bosque de caminhada da UFMT: (a) Trecho inicial do bosque com uma lacuna Arbórea, (b) Trecho do meio ao fim do trecho de caminhada.
Fonte: PEREIRA, R. B. de O. (2013).

O local possui acesso parcial a ventilação e a dissipação de energia do pavimento, recebendo insolação em parte do fotoperíodo (Figura 69).



Figura 69 – Visão de parte do céu no sétimo trecho.

Fonte: PEREIRA, R. B. de O. (2013).

O oitavo trecho, às margens da avenida Fernando Corrêa da Costa, demarcado pela linha vermelha e numerada de 8 na Figura 52, tem um total aproximado de 332 metros, iniciando no final do bosque de caminhada (Figura 70a) indo até a ponte sobre o Córrego do Barbado e retornando ao acesso da UFMT. Este trecho é composto por pavimento rígido (Figura 70b), tendo o início nas coordenadas $15^{\circ}36'49.85"S, 56^{\circ} 4'23.33"O$ e término em $15^{\circ}36'49.62"S, 56^{\circ} 4'22.50"O$.



Figura 70 – Trecho de caminhada na avenida Fernando Corrêa da Costa: (a) Trecho inicial no final do bosque (b) Parte do meio do oitavo trecho de caminhada.
Fonte: PEREIRA, R. B. de O. (2013).

Esse trajeto possui acesso total à ventilação e à dissipação de energia do pavimento, recebendo insolação em todo o fotoperíodo (Figura 71).



Figura 71 – Visão de parte do céu no oitavo trecho.

Fonte: PEREIRA, R. B. de O. (2013).

O nono trecho, às margens da avenida principal de acesso à UFMT, demarcado pela linha verde e numerada de 9 na Figura 52, tem um total aproximado de 244 metros, iniciando no acesso à UFMT (Figura 72a) até a guarita. O percurso deste trecho é composto por paralelepípedo (Figura 72b), contornado por espécies de vegetação lenhosa de grande porte, além de uma vegetação nativa rasteira, tendo o início nas coordenadas $15^{\circ}36'49.62"S, 56^{\circ} 4'22.50"O$ e término em $15^{\circ}36'45.50"S, 56^{\circ} 4'15.82"O$.



Figura 72 – Trecho da avenida de acesso a UFMT: (a) Calçada inicial da avenida de acesso à UFMT, (b) Trecho do meio da pista de caminhada.

Fonte: PEREIRA, R. B. de O. (2013).

O trajeto está localizado em uma área que pode ser considerada parcialmente aberta, possuindo acesso parcial à ventilação e com obstrução à dissipação de energia disponível no pavimento para o céu (Figura 73), além de propiciar uma insolação parcial durante o fotoperíodo.

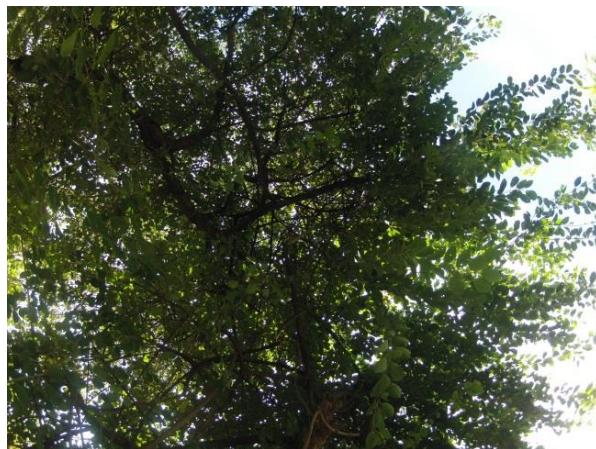


Figura 73 – Visão de parte do céu no nono trecho.

Fonte: PEREIRA, R. B. de O. (2013).

O décimo trecho, demarcado pela linha amarela e numerada de 10 na Figura 52, tem um total de aproximadamente 180 metros, correspondendo ao trecho de entrada após a primeira guarita da UFMT, tendo o início nas coordenadas $15^{\circ}36'45.50"S$, $56^{\circ} 4'15.82"O$ e término em $15^{\circ}36'46.22"S$, $56^{\circ} 4'10.26"O$. Todo o percurso é composto por paralelepípedo, contornado por um bosque com vegetação lenhosa de grande porte (Figura 74b), além de uma vegetação nativa rasteira e, ao seu lado direito, um pavimento flexível do tipo pré-misturado a frio (Figura 74a).



Figura 74 – Avenida principal de acesso à UFMT após a guarita: (a) Calçada inicial da avenida de acesso à UFMT, (b) Bosque na entrada da UFMT.

Fonte: PEREIRA, R. B. de O. (2013).

Esse trajeto está localizado em uma área que pode ser considerada fechada, possuindo restrições à ventilação e com obstrução à dissipação de energia disponível no pavimento (Figura 75), além de não propiciar uma insolação direta durante o fotoperíodo.

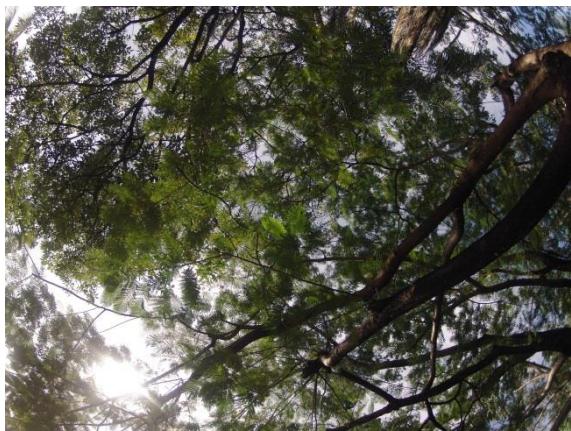


Figura 75 – Visão de parte do céu no décimo trecho.

Fonte: PEREIRA, R. B. de O. (2013).

O décimo primeiro trecho fica na avenida principal da UFMT, demarcado pela linha azul e numerada de 11 na Figura 52, tem um total aproximado de 219 metros, correspondendo ao trecho que vai da frente das quadras de esporte ao campo de futebol da UFMT, tendo o início nas coordenadas $15^{\circ}36'46.22"S$, $56^{\circ}4'10.26"O$ e término em $15^{\circ}36'41.74"S$, $56^{\circ}4'4.70"O$. Todo o percurso é composto por paralelepípedo, ao seu lado esquerdo, um pavimento flexível do tipo pré-misturado a frio (Figura 76a) e ao lado direito, contornado por um bosque com vegetação lenhosa de grande porte (Figura 76a e 76b), além de uma vegetação nativa rasteira.



(a)



(b)

Figura 76 – Avenida principal da UFMT: (a) Calçada do inicial do trajeto, (b) Trecho intermediário do percurso.

Fonte: PEREIRA, R. B. de O. (2013).

Esse trajeto está localizado em uma área que pode ser considerada aberta, não possuindo restrições à ventilação e com obstrução parcial da dissipação da energia disponível no pavimento (Figura 77), além de propiciar uma insolação parcial direta durante o fotoperíodo.



Figura 77 – Visão de parte do céu no décimo primeiro trecho.
Fonte: PEREIRA, R. B. de O. (2013).

O décimo segundo trecho, também na avenida principal da UFMT, demarcado pela linha vermelha e numerada de 12 na Figura 52, tem um total aproximado de 203 metros, correspondendo ao trecho que vai da frente do campo de futebol da UFMT ao Ginásio, tendo o início nas coordenadas $15^{\circ}36'41.64"S$, $56^{\circ}4'4.57"O$ e término em $15^{\circ}36'37.49"S$, $56^{\circ}3'59.40"O$. Todo o percurso é composto por paralelepípedo, ao seu lado esquerdo, um pavimento flexível do tipo pré-misturado a frio (Figura 78a) e ao lado direito, um bosque com vegetação lenhosa de grande porte (Figura 78b) e uma vegetação nativa rasteira.

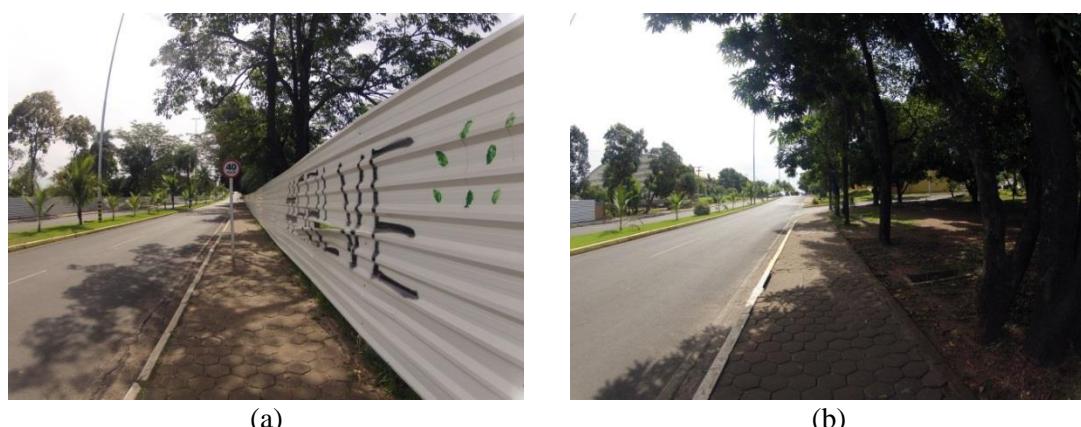


Figura 78 – Avenida principal da UFMT: (a) Calçada do inicial do trajeto, (b) Trecho intermediário do percurso próximo ao Ginásio.
Fonte: PEREIRA, R. B. de O. (2013).

O trajeto está localizado em uma área que pode ser considerada aberta, possuindo restrições à ventilação principalmente em sua parte inicial e com obstrução parcial da dissipação da energia disponível no pavimento (Figura 79), além de propiciar uma insolação parcial direta durante o fotoperíodo.



Figura 79 – Visão de parte do céu no décimo segundo trecho.
Fonte: PEREIRA, R. B. de O. (2013).

O décimo terceiro trecho na avenida principal da UFMT, demarcado pela linha laranja e numerada de 13 na Figura 52, tem um total aproximado de 170 metros, correspondendo ao trecho que vai do Ginásio e passa pela frente do Restaurante Universitário, tendo o início nas coordenadas $15^{\circ}36'37.49"S$, $56^{\circ}3'59.40"O$ e término em $15^{\circ}36'34.01"S$ $56^{\circ}3'55.00"O$. Todo o percurso é composto por paralelepípedo, ao seu lado esquerdo, um pavimento flexível do tipo pré-misturado a frio, comportando um grande estacionamento (Figura 80b) e, ao lado direito, um bosque com vegetação lenhosa de grande porte (Figura 80a), uma vegetação nativa rasteira e prédios de salas de aula.



Figura 80 – Avenida principal da UFMT: (a) Calçada do inicial do trajeto, (b) Trecho intermediário do percurso na frente do Restaurante Universitário e do estacionamento.
Fonte: PEREIRA, R. B. de O. (2013).

Esse trajeto está localizado em uma área aberta, não possuindo restrições à ventilação e sem obstrução à dissipação da energia disponível no pavimento (Figura 81), além de propiciar uma insolação direta durante o fotoperíodo.



Figura 81 – Visão de parte do céu no décimo terceiro trecho.
Fonte: PEREIRA, R. B. de O. (2013).

O décimo quarto trecho na avenida principal da UFMT, demarcado pela linha rosa e numerada de 14 na Figura 52, tem um total aproximado de 145 metros, correspondendo ao trecho que vai do Restaurante Universitário até a rotatória em frente ao bloco da Física Ambiental, tendo início nas coordenadas $15^{\circ}36'34.01"S$, $56^{\circ} 3'55.00"O$ e término em $15^{\circ}36'32.35"S$, $56^{\circ} 3'50.39"O$. O percurso é composto por pavimento rígido (calçada de concreto), ao lado direito, um bosque com vegetação lenhosa de grande porte (Figura 82a), uma vegetação nativa rasteira e prédios de salas de aula e, ao seu lado esquerdo, um pavimento flexível do tipo pré-misturado a frio, comportando um grande estacionamento (Figura 82b).



Figura 82 – Avenida principal da UFMT: (a) Calçada do inicial do trajeto, (b) Trecho na frente do Restaurante Universitário e do estacionamento.
Fonte: PEREIRA, R. B. de O. (2013).

Esse trajeto está localizado em uma área aberta, não possuindo restrições à ventilação e sem obstrução à dissipação da energia disponível no pavimento (Figura 83), além de propiciar uma insolação direta durante o fotoperíodo.



Figura 83 – Visão de parte do céu no décimo quarto trecho.

Fonte: PEREIRA, R. B. de O. (2013).

O décimo quinto trecho, parte final do percurso, na avenida principal da UFMT, demarcado pela linha preta e numerada de 15 na Figura 52, tem um total aproximado de 149 metros; tem o seu trecho da rotatória em frente ao bloco da Física Ambiental até a frente do bloco do Instituto de Computação; o seu início tem as coordenadas $15^{\circ}36'32.35"S$, $56^{\circ} 3'50.39"O$ e o término em $15^{\circ}36'29.71"S$, $56^{\circ} 3'46.64"O$. O percurso é composto por pavimento rígido (Figura 84b) e pavimento flexível do tipo pré-misturado a frio, ao lado direito, prédios de salas de aula (Figura 84a) e, ao seu lado esquerdo, um pequeno bosque com salas de aula.



Figura 84 – Trecho final do percurso: (a) Visão da rotatória em frente ao Bloco da Física Ambiental, (b) Trecho do percurso final.

Fonte: PEREIRA, R. B. de O. (2013).

Esse trajeto está localizado em uma área aberta, não possuindo restrições a ventilação e sem obstrução à dissipação da energia disponível no pavimento (Figura 85), além de propiciar uma insolação direta durante o fotoperíodo.



Figura 85 – Visão de parte do céu no décimo quarto trecho.
Fonte: PEREIRA, R. B. de O. (2013).

Dessa forma, é possível perceber, que a partir do meio do trecho 02, há um gradiente da temperatura e umidade relativa do ar, isso provavelmente deve ter ocorrido pela presença de construção e obras no entorno da pista.

Já no trecho 03, que contorna o Restaurante Universitário pelos fundos, percebe-se uma pequena diminuição na temperatura, isso provavelmente se deve à presença de uma pequena mata que fica entre o restaurante e o Córrego do Barbado.

A partir do trecho 4 até o 6, percebe-se um incremento na temperatura, isso pode ser o resultado das construções, como a do ginásio, bem como as reformas para melhoria de infraestrutura visando a Copa do Mundo em 2014, onde foi colocada uma fileira de zinco na avenida para fechar o local, fator esse que, somado ao próprio asfalto, deve ter resultado nesse aumento de temperatura.

No trecho 9, onde se localiza a via de retorno à universidade, se percebe um decremento na temperatura e um aumento na umidade relativa do ar, o que pode ser resultado da influência do bosque sobre a avenida, o que se estende até boa parte do trecho 10, pelo fato desse trecho ainda ser circundado por um bosque.

Já no trecho 11, que retorna à avenida fechada pelas placas de zinco, percebe-se um aumento gradativo da temperatura e uma queda significativa da umidade, tendo o seu ápice entre o trecho 11 e 12.

Entre esses trechos, que estão entre o ginásio de esportes e o pátio de estacionamento na frente do Restaurante Universitário, que é uma área relativamente grande de asfalto e poucas árvores nas imediações, percebe-se um incremento na temperatura e um decréscimo na umidade relativa do ar, o que pode ter se originado dos elementos citados.

Já nos trechos 13 a 15, percebe-se uma diminuição da temperatura e a umidade relativa do ar teve o mesmo comportamento do trecho 12; isso deve ter ocorrido pelo aumento da vegetação nativa e da diminuição da área asfaltada, se comparada ao trecho 12.

A média das variações, tanto de temperatura e umidade relativa do ar é demonstrada na Tabela 12.

Tabela 12 - Média das temperaturas e umidade relativa do ar nos trechos.

Trecho	Temperatura do ar (°C)	Umidade relativa do ar (%)
1	30,23	54,88
2	30,62	55,34
3	30,42	54,71
4	31,15	52,08
5	31,06	52,50
6	31,50	52,05
7	30,32	55,01
8	31,30	51,59
9	30,78	52,95
10	30,32	53,88
11	31,57	53,20
12	33,03	49,80
13	32,97	49,55
14	32,43	49,49
15	32,09	49,67

Elab.: PEREIRA, R. B. de O. (2013).

5. CONSIDERAÇÕES FINAIS

A utilização do método de termotransferência para a confecção das trilhas na placa de fenolite se mostrou eficiente, pois com aproximadamente 5 minutos, o processo estava concluído, sendo um fator importante para a prototipagem de várias versões.

A utilização do padrão *Arduino* para o desenvolvimento das placas se mostrou eficiente, pois durante os testes foi possível alternar as *shield* para verificá-las quanto a maus contatos, soldas frias, interferências e outras adversidades que podem ocorrer durante a confecção.

Tanto o GPS quanto o relógio de tempo real cumpriram com suas funções, sendo estáveis e com uma boa relação custo-benefício.

O sensor de temperatura e umidade relativa do ar DHT22 se mostrou eficiente, mesmo sendo um sensor de baixo custo, porém, devido à utilização de capacitores de origem duvidosa nos testes iniciais, o sensor apresentou instabilidade, pois, ao invés de os capacitores funcionarem como filtros para o desacoplamento da fonte, funcionaram como ressonadores, imputando ruído à fonte de alimentação e aos sinais digitais transmitidos, sendo resolvido com a troca dos capacitores.

A utilização do cartão SD para o armazenamento dos dados coletados e do arquivo de configuração pode ser considerada satisfatória, pois com um cartão de 2 GB e adotando que a cada hora serão gastos 90 KB, tem-se capacidade de gravar aproximadamente 2300 horas de dados, o que pode ser considerado, pela relação custo-benefício do cartão, um resultado bom.

A utilização da técnica de monitoramento por *bluetooth* se mostrou eficiente, pois com a utilização do *buzzer* em ambiente com uma considerável perturbação sonora, como em uma avenida, logo fica quase impossível escutar o sinal devido ao barulho externo, fator esse que inexiste na utilização desse método de conexão, pois

com ela é possível obter informações mais detalhadas sobre o real estado do *datalogger*.

Os cabos utilizados para a transmissão dos dados também se mostraram eficientes, pois além de diminuírem a interferência externa por meio de suas blindagens, eles facilitavam a sua introdução na estrutura de suporte, devido à presença de uma camada externa de plástico que os deixava mais rígidos, facilitando a sua inserção na estrutura.

A utilização da estrutura em canos de PVC de 50 mm foi positiva, pois, além de ser de fácil aquisição no mercado local, possibilitou um conjunto de adequações e adaptações de forma simples que, com outros materiais, podem não trazer a mesma vantagem. Além disso, existem vários tipos de acessórios disponíveis no mercado, como: curvas, luvas, conexões, fator esse que favorece a utilização desse material.

A autonomia do sistema pode ser considerada satisfatória, pois, utilizando o mesmo com todos os recursos, obteve-se um consumo máximo de aproximadamente 107 mA e, levando em consideração a bateria utilizada neste trabalho, a autonomia dele seria de aproximadamente 40 horas de coleta contínua.

A utilização do arquivo de configuração pode ser considerada satisfatória, pois retira a necessidade de um programa específico de um fornecedor para a sua configuração, resultando, assim, em uma maior praticidade para a configuração do equipamento.

O método utilizado para armazenar os dados, bem como o método de recuperação desses arquivos, em consonância com o parágrafo anterior, também podem ser considerados satisfatórios, pois da mesma forma que não é atrelada a um programa a configuração do sistema, a recuperação dos arquivos também não é, o que resulta, para o usuário final, a possibilidade de utilizar seu programa predileto, diminuindo assim a resistência na utilização do equipamento e facilitando a vida do usuário final.

A utilização das ferramentas do *Arduino* para a codificação do sistema também pode ser considerada satisfatória, pois, além de já possuir um conjunto de bibliotecas nativas, é possível encontrar um grande acervo de bibliotecas de código aberto na Internet, além da vasta documentação disponível.

A utilização do *software Blueterm* se mostrou eficiente, pois, além de ser gratuito e apresentar os dados recebidos pelo dispositivo, é possível utilizá-lo como um segundo *logger*, fator esse que dá mais segurança ao usuário durante coletas críticas e difíceis de serem repetidas.

A partir dos resultados das análises estatísticas realizadas no estudo de caso, pode-se concluir que o equipamento e a metodologia utilizada são eficientes. A estrutura de *hardware* e *software* livre pôde ser avaliada por meio deles. O equipamento se mostrou robusto no que tange ao armazenamento dos dados.

A utilização do conjunto como um todo pode ser considerada satisfatória, pois o peso do equipamento ficou em aproximadamente 2,1 Kg, o que, nos dias atuais, representa o peso de um notebook comum.

O custo total do experimento ficou em R\$ 310,90, incluindo todos os sensores, suportes, canos, mídias de armazenamento e bateria, resultado esse que pode ser classificado como bom.

No tocante ao consumo de energia, a confecção da placa do *Arduino* adaptada, bem como todos os módulos, pode ser considerado bom, principalmente por proporcionar uma autonomia relativamente alta ao usuário do sistema.

Dessa forma, é possível concluir que a utilização da tecnologia do *Arduino* para o desenvolvimento de *datalogger* é satisfatória, bem como esta tese pode contribuir com o programa, proporcionando um salto qualitativo no que tange à possibilidade de obtenção de séries microclimáticas com baixa quantidade de falhas com o *Arduino*.

5.1. TRABALHOS FUTUROS

Ao longo da elaboração desta tese, novas linhas de pesquisas foram emergindo, porém não foi possível incluí-las neste trabalho, mas podem se constituir matéria para trabalhos futuros; dentre elas é possível elencar:

1. Avaliar a troca da tecnologia AVR do *Arduino* para ARM ou FPGA;
2. Implementar uma rede de comunicação de dados por NRF, para que se possa aumentar a dinamicidade e a capacidade da coleta de dados;

3. Melhorar a estrutura de suporte dos sensores, bem como do abrigo térmico do sensor de temperatura e umidade do ar, para que eles possam ser utilizados como estação fixa;
4. Utilizar o sistema de conversão DC-DC *buck-booster*, para aumentar a eficiência energética do sistema, diminuindo assim a perda de energia por dissipação de calor do regulador de tensão.

REFERÊNCIAS BIBLIOGRÁFICAS

BIBLIOGRAFIAS CITADAS

AOSONG, E. **Digital-output relative humidity & temperature sensor/module AM2303**. Guangzhou, China: Aosong, 2013. 7 f.

ARDUINO. **Frequently Asked Questions**. [on line]. Disponível em: <<http://arduino.cc/en/Main/FAQ>>. Acessado em: Março de 2013.

ARDUINO.CC. **Arduino IDE versão 1.05**. [on line]. Disponível em: <<http://arduino.cc/en/Main/Software>>. Acessado em: Abril de 2013.

ASTAH. **Astah Community versão 6.7.0 para Linux**. [on line]. Disponível em: <<http://astah.net/download>>. Acessado em: Abril de 2013.

ATMEL. **8-bit AVR ATmega328 Microcontroller with 4/8/16/32K Bytes In-System Programmable Flash**. San Jose, CA, USA: Atmel Corporation, 2009. Disponível em: <<http://www.atmel.com/Images/doc8161.pdf>>. Acessado em: Março de 2013.

BADIN JR., H. **Datalogger**. São Paulo: Saber Eletrônica, 2009. 11 f.

BARBOSA, R. V. R. , VECCHIA, F.A.S. Estudos de ilha de calor urbana por meio de imagens do LANDSAT 7 ETM+: Estudo de Caso em São Carlos (SP). **Revista Minerva -Pesquisa e Tecnologia**, v. 6, n. 3, set./dez. 2009. 6 f.

BARROS, M. P. **Dimensão fractal e ilhas de calor urbanas: uma abordagem sistemica sobre as implicações entre a fragmentação das áreas verdes e o ambiente térmico do espaço urbano**. Tese (Doutorado em Física Ambiental) – Programa de Pós-Graduação em Física Ambiental. Universidade Federal de Mato Grosso. Cuiabá: UFMT, 2012. 171 f.

BEAGLEBOARD.ORG. **About BeagleBoard.org**. 2013. Disponível em: <<http://beagleboard.org/about>>. Acessado em: Outubro de 2013.

CADSOFT. **Eagle PCB versão 5 para Linux**. [on line]. Disponível em: <<http://www.cadsoftusa.com/download-eagle/?language=en>>. Acessado em: Abril de 2013.

CAMPELO JÚNIOR, J. H.; CASEIRO, F. T.; PRIANTE FILHO, N.; BELLO, G. C. C.; MAITELLI, G. T.; ZAMPARONI, C. A. G. P. Caracterização Macroclimática de Cuiabá. In: ENCONTRO NACIONAL DE ESTUDOS SOBRE O MEIO AMBIENTE, 3, *Anais...* p. 545-552, Londrina, PR: FUEL, 1991. 1991.

CAMPOS NETO, A. A. **Estudo bioclimático no campus da Universidade Federal de Mato Grosso.** Dissertação (Mestrado em Climatologia) – Programa de Pós-Graduação em Física e Meio Ambiente. Instituto de Ciências Exatas e da Terra. Universidade Federal de Mato Grosso. Cuiabá: UFMT, 2007. 177 f.

CATSOULIS, J. **Designing Embedded Hardware.** Sebastopol, CA, USA: O'Reilly Media, 2009.

CAVALCANTE, M. A.; TAVOLARO, C. R. C.; MOLISANI, E. Física com Arduino para iniciantes. **Revista Brasileira de Ensino de Física**, v. 33, n. 4, São Paulo, out./dez. 2011. 9 f.

DA MARTHA, E. G. M.; FELIX, L. S.; COLLISCOHNN, E. Variação da temperatura, umidade relativa do ar e vento no eixo viário norte da cidade de Pelotas, RS – Levantamento Episódico no verão de 2011. **XX Congresso de Iniciação Científica**, UFPEL, Pelotas-RS, 2011.

DELGADO, J.; RIBEIRO, C. **Arquitectura de computadores.** 4. ed. Lisboa, Portugal: Editora Lidel, 2010. 758 f.

E-GIZMO MECHATRONIX CENTRAL. **Bluetooth Modules Wireless UART Cable Replacement Hardware Manual & AT Commands Reference Manual Rev. 1r0.** Estados Unidos da América: Rasmicro, 2011. Disponível em: <<http://www.rasmicro.com/Bluetooth/EGBT-045MS-046S%20Bluetooth%20Module%20Manual%20rev%201r0.pdf>>. Acessado em: Março de 2013.

FAIRCHILD SEMICONDUCTOR. **MC78XX/LM78XX/MC78XXA 3-Terminal 1A Positive Voltage Regulator.** San Jose, CA, EUA: FCS, 2001. 29 f.

FAUGEL, H.; BOBKOV, V. Open source hard-and software: Using Arduino boards to keep old hardware running. **Fusion Engineering and Design**, Los Angeles, CA, EUA: Elsevier, 2013. 4 f.

FIALHO, E. S. **Os microclimas e o uso do solo na Ilha do Governador.** X SIMPÓSIO BRASILEIRO DE GEOGRAFIA FÍSICA APLICADA, Número especial de 2003 – ISSN 1415 – 1543, Revista do Departamento de Geografia (Geo UERJ), Rio de Janeiro, 2003. P 163.

FRADEN, J. **Handbook of Modern Sensors: Physics, Designs, and Applications.** 4. ed. New York, NY, USA: Springer, 2010. 680 f.

FRAMPTON, A.; PAINTER, S. L.; DESTOUNI, G. Permafrost degradation and subsurface-flow changes caused by surface warming trends. **Hydrogeology Journal**, V. 21, Issue 1, p 271-280, Springer, Berlin/Heidelberg-DE, 2013.

GAMMA DESIGN. **GS+ Geostatistics for Environmental Sciences versão 9 de demonstração.** [on line]. Disponível em: <<http://www.gammadesign.com/default.aspx>>. Acessado em: Abril de 2013.

GARCIA, P. O.; LOBO-FARIA, P. C. **Metodologias para Levantamentos da Biodiversidade Brasileira.** Palestra sobre metodologia. Universidade Federal de Juiz de Fora. Juiz de Fora: UFJF, 2005.

GIANEZINI, Quelen. **O processo de expansão do ensino superior em Mato Grosso.** Dissertação (Mestrado em Sociologia) - Programa de Pós-Graduação em Sociologia. Instituto de Filosofia e Ciências Humanas. Universidade Federal do Rio Grande do Sul. Porto Alegre: UFRS, 2009. 238 f.

GOLDEN. **Surfer versão 10.1.561 para Windows.** Edição de 32 bits. DVD-ROM, 2011.

GOOGLE. **Google Earth versão 7.1 Pro gratuito por 7 dias.** [on line]. Disponível em: <<http://www.google.com.br/earth/download/gep/agree.html>>. Acessado em: Abril de 2013.

IBM. **IBM SPSS Statistics da IBM versão 21.** Edição de 32 bits. DVD-ROM, 2012.

KORNUTA, J. A.; NIPPER, M. E.; DIXON J. B. Low-cost microcontroller platform for studying lymphatic biomechanics in vitro. **Journal of Biomechanics**, p. 183-187, Los Angeles, CA, EUA: Elsevier, 2013.

KRISTINSSON, B.; VIGFÚSSON, Ý. Ardrand: The Arduino as a Hardware Random-Number Generator. **arXiv**, v. 1, Cornell University Library, Ithaca NY, USA, 2012. 20 f.

KUSHNER, D. **The Making of Arduino:** How five friends engineered a small circuit board that's taking the DIY world by storm. 26 Oct. 2011. [on line]. Disponível em <<http://spectrum.ieee.org/geek-life/hands-on/the-making-of-arduino/0>>. Acessado em: Abril de 2013.

LOUP, T. O.; TORRES, M.; MILIAN, F. M.; AMBRÓSIO, P. E. Bluetooth Embedded System for Room-Safe Temperatura Monitoring. **IEEE Latin America Transactions**, v. 9, n. 6, s/l, outubro 2011.

MAITELLI, G. T. **Uma abordagem Tridimensional do clima urbano em área Tropical Continental: o exemplo de Cuiabá/MT.** Tese (Doutorado em Climatologia) – Departamento de Geografia - Faculdade de Filosofia, Letras e

Ciências Humanas. Universidade de São Paulo. São Paulo: USP/FFLCH, 1994. 204 f.

MARGOLIS, M. **Arduino Cookbook.** Sebastopol, CA, USA: O'Reilly Media,, 2011. 658 f.

MAXIM. **DS1307 64 x 8, Serial, I2C Real-Time Clock.** San Jose, CA, USA: Maxim Integrated, 2012. 14 f.

MESCHIATTI, M. C.; FONTOLAN, M. R.; PENEREIRO, J. C.; FERREIRA, D. H. L. Caracterização Estatística de Tendências em Séries Anuais de Dados Hidro-Climáticos no Estado de São Paulo. **Revista Geografia Acadêmica**, v. 6, n. 1, UFG-Universidade Federal de Goiás, Goiânia, 2012. 13 f.

MONK, S. **30 Arduino Projects for the Evil Genius.** New York, NY, USA: McGraw Hill, 2010. 208 f

MONTEIRO, C. A. de; F. MENDONÇA, F. **Clima Urbano.** 1. ed. São Paulo: Contexto, 2003. 192 p.

NERY, C. A. S.; ALLOZA, J. F. M.; BATISTA, F. Sensibilidade térmica cutânea: estudo comparativo entre dois métodos de mensuração e proposta de simplificação. **Revista Brasileira de Ortopedia**, São Paulo, 2000.

OKE, T. R.; VOOGT, J. A. Thermal remote sensing of urban climates. **Remote Sensing Environment**, v. 86, p. 370-384, Los Angeles, CA, USA: Elsevier, 2003.

OXER, J.; BLEMINGS, H. **Practical Arduino:** Coll Projects for Open Source Hardware. Berkeley, CA, USA: Apress, 2009. 445 f.

PARK, J; MACKAY, S. **Data acquisition for Instrumentation and Control System.** Los Angeles, CA, USA: Elsevier, 2003. 425 f.

PYMASDE.ES. **BlueTerm versão 1.1 gratuito para Android.** [on line]. Disponível em: <https://play.google.com/store/apps/details?id=es.pymasde.blueterm&hl=pt_BR>. Acessado em: Abril de 2013.

RASPBERRY PI. **About us.** [on line]. 2013. Disponível em: <<http://www.raspberrypi.org/about>>. Acessado em: Outubro de 2013.

ROMERO, M. A. B. **Arquitetura Bioclimática do Espaço Público.** 1. ed. Brasília: Editora Universidade de Brasília, 2001.

SALCIC, Z.; RADOJEVIC, I. **Embedded Systems Design Based on Formal Models of Computation.** New York, NY, USA: Springer, 2011. 200 f.

SANDHU, H. S. **Making PIC microcontroller instruments and controlles.** New York, NY, USA: McGraw-Hill, 2009. 350 f.

SILVA, J. H. D. **Precisão, Exatidão, e a Terminologia das Medições.** Bauru: Unesp, 2006. 10 f.

SIQUEIRA, E. M. **História de Mato Grosso:** da ancestralidade aos dias atuais. Cuiabá: Entrelinhas, 2002. 272 f.

SKYLAB. **GPS Module DataSheet:** Ultra High Sensitivity and The Smart Antenna GPS Module. Shenzhen, China: Skylab M&C Technology Co., 2010. 10 f.

SOARES, M. J. Hoje tudo é sistema embarcado. Será? **Revista Saber Eletrônica,** São Paulo, 2008. 2 f.

SOUZA, L. C. L.; MATTOS, A. Ilha de calor e geometria urbana na cidade de São Carlos/SP. In: IV Encontro Nacional de Conforto no Ambiente Construído - ENCAC, **Anais...** p. 97-101, FAUFBA/LACAM/ANTAC, Salvador, BA, 1997.

SOUZA, D. O. de; BATISTA, G. M. de M.; **Análise da influência da resolução espacial na determinação das ilhas urbanas em São Paulo, por meio dos sensores ASTER e MODIS.** Anais XII Simpósio Brasileiro de Sensoriamento Remoto, Goiânia, Brasil, INPE ,16-21 abril 2005, p. 4525-4530

SOUZA, A. R. de; PAIXÃO, A. C. ; UZÊDA, D. D.; DIAS, M. A.; DUARTE, S.; AMORIM, H. S. de; **A placa Arduino: uma opção de baixo custo para experiências de física assistidas pelo PC.** Revista Brasileira de Ensino de Física, v. 33, n. 1, 2011. 5f.

SPARKFUN. **SparkFun Eagle Libraries versão 6 free.** [on line]. Disponível em: <<https://www.sparkfun.com/static/eagle/>>. Acessado em: Abril de 2013.

STALLINGS, W. **Arquitetura e Organização de computadores.** 8. ed. São Paulo: Editora Pearson, 2010. 624 f.

STMicroelectronics. **L78L00 Series Positive Voltage Regulators.** Genebra, Suíça: STMicroelectronics N.V., 2003. 22 f.

TANENBAUM, A. S. **Redes de Computadores.** 5. ed. São Paulo: Editora Pearson, 2011. 582 f.

TANENBAUM, A. S. **Sistemas Operacionais Modernos.** 3. ed. São Paulo: Editora Pearson, 2010. 653 f.

UGEDA JÚNIOR, J. C.; AMORIM, M. C. C. T. Estudo do clima urbano de Jales-SP através do transecto móvel. **Revista Geonorte**, Edição Especial 2, v. 2, n. 5, p. 365-377, UFAM, Manaus-AM, 2012

VASCONCELOS, L. C. S.; ZAMPARONI, C. A. G. P. Os Efeitos da Urbanização no Microclima no Bairro Morada da Serra, Cuiabá-MT. **Revista Ra'e Ga Ga – O Espaço Geográfico em Análise,** [1516-4136], v. 23, 573 p. UFPR, Curitiba, 2011.

VIANELLO, R. Leite; ALVES, A. Rainier. **Meteorologia Básica e Aplicada.** 1. ed. Viçosa-MG: Editora UFV, 2002. 449 p.

WARREN, J.; ADAMS, J.; MOLLE, H. **Arduino robotics.** Apress, 2010, 446 p.

WILSON, J. S. **Sensor Technology Handbook.** Burlington, MA, USA: Elsevier, 2004. 703 f.

ZACHARIADOU, K.; YIASEMIDES, K.; TROUGKAKOS. N. A low-cost computer-controlled Arduino-based educational laboratory system for teaching the fundamentals of photovoltaic cells. **European Journal of Physics**, v. 33, IOP Publishing, Bristol, UK, 2012.

BIBLIOGRAFIAS CONSULTADAS

FOWLER, M.; SCOTT, K. **UML Distilled, Applying the Standard Object Modeling Language.** Boston, USA: Addison Wesley, 1997.

JUNIOR, Osmar de Oliveira Braz. **Apostila de Engenharia de Software 2.** Tubarão-SC: UNISUL, 1998. Disponível em: <<http://tec1.unisul.rct-sc.br/osmarjr/>>. Acessado em: Novembro de 2012.

PAGE-JONES, M. **Gerenciamento de Projetos.** New York, NY, USA: McGraw Hill, 1990.

ANEXOS

ANEXO A - CÓDIGO DO ZOIOLOGGER

```
#include <dht22.h>
#include <Wire.h>
#include "RTClib.h"
#include <SD.h>
#include <SoftwareSerial.h>

#define maximovalor -9999
#define minimovalor 9999
#define Reset_AVR() wdt_enable(WDTO_30MS); while(1) {}

dht22 DHT;
RTC_DS1307 RTC;

SoftwareSerial gps = SoftwareSerial(7,6);

unsigned int amostra=0,media=0;
byte dump=0,beep=0,auxint=0,auxint2=0,auxint3=0,listasensor[9][2];
long tempoamostra=0,tempoatual=0;
double auxdouble=0;
DateTime relogio;
char buffer[32],buffer2[12],auxchar;
File arquivo;

void tocar(){
  tone(beep,3000);
  delay(150);
  noTone(beep);
  delay(80);
}

int freeRam() {
  extern int __heap_start, *__brkval;
  int v;
  return (int) &v - (__brkval == 0 ? (int) &__heap_start : (int) __brkval);
}

char lergps(){
  while(1) if(gps.available()) return gps.read();
}

char erro(){
  while(1){
    if (beep!=0) {
      tocar();
    }
    delay(100);
  }
}

void setup(){
  Serial.begin(9600);
  Serial.println(F("Iniciando"));
}
```

```

if (!SD.begin(8)) {
    Serial.println(F("Problema no cartao"));
    erro();
}
// Iniciando SD para pegar arquivo de configuracao
arquivo = SD.open("config.txt",FILE_READ);
if (!arquivo) {
    Serial.println(F("Problema no arquivo"));
    erro();
}
auxint=0;
auxint2=0;
auxint3=0;
// Processando arquivo
while (arquivo.available()) {
    auxchar=arquivo.read();
    if(auxchar==35) while(arquivo.read()!=10);
    if(auxchar==64){
        // Localizado inicio de comando
        auxint=0;
        auxint2=0;
        while((auxchar=arquivo.read())!=10){
            if(auxchar==61){
                buffer[auxint]='\0';
                auxint=0;
                auxint2=1;
            }
            else {
                if(auxint2==0){
                    buffer[auxint]=(char) auxchar;
                    auxint++;
                }
                else{
                    buffer2[auxint]=(char) auxchar;
                    auxint++;
                }
            }
        }
        buffer2[auxint]='\0';
        // Iniciando interpretacao dos dados
        if (strcmp(buffer, "AMOSTRA")==0){
            amostra=atoi(buffer2);
        }
        else if (strcmp(buffer, "BEEP")==0){
            beep=atoi(buffer2);
            pinMode(beep,OUTPUT);
        }
        else if (strcmp(buffer, "SENSOR")==0){
            //processar os tipos de sensores
            for(auxint=0;auxint<5;auxint++) buffer[auxint]=buffer2[auxint];
            buffer[auxint]='\0';
            buffer2[0]=buffer2[6];
            buffer2[1]='\0';
            if (strcmp(buffer, "analo")==0){
                listasensor[auxint3][0]=1;
                listasensor[auxint3][1]=atoi(buffer2);
                auxint3++;
            }
        }
    }
}

```

```

else if(strcmp(buffer, "dht22")==0){
    listasensor[auxint3][0]=2;
    listasensor[auxint3][1]=atoi(buffer2);
    auxint3++;
}
else if(strcmp(buffer, "gpskm")==0){
    listasensor[auxint3][0]=4;
    //listasensor[auxint3][1]=atoi(buffer2);
    gps.begin(9600);
    auxint3++;
}
else if(strcmp(buffer, "divte")==0){
    listasensor[auxint3][0]=5;
    listasensor[auxint3][1]=3;
    auxint3++;
}
}
}

arquivo.close();
// Iniciando dispositivos
Wire.begin();
RTC.begin();

auxint2=0;
for(auxint=0;auxint<9;auxint++) if(listasensor[auxint][0]==4) { auxint2=auxint; }// Procurando GPS
if (auxint2>0){ //GPS Encontrado
    gps.flush();
    auxint=0;
    auxint2=0;
    auxint3=0;
    auxdouble=0;
    Serial.println(F("Configurando GPS"));
    while(auxint3==0){
        auxchar=lergps();
        if(auxchar=='$'){
            buffer[0]=lergps();
            buffer[1]=lergps();
            buffer[2]=lergps();
            buffer[3]=lergps();
            buffer[4]=lergps();
            buffer[5]='\0';
            auxchar=lergps();
            if((strcmp(buffer,"GPGGA")==0) && auxint2==0){
                // $GPGGA,083559.00,3723.2475,N,12158.3416,W,1,07,1.0,9.0,M,,M,,0000*18
                auxint=0;
                while((auxchar=lergps())!=','){ }
                while((auxchar=lergps())!=','){
                    buffer[auxint]=auxchar;
                    auxint++;
                }
                buffer[auxint]='\0';
                auxdouble=atof(buffer);
                Serial.print(F("."));
                if(auxdouble!=8960){
                    auxint2=1;
                    gps.println(F("$PMTK330,196*20"));
                    gps.println(F("$PMTK314,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0*28"));
                }
            }
        }
    }
}

```

```

        }
        if (strcmp(buffer,"GPZDA")==0){ Serial.print(F("+")); }
        if (auxint2==1){ Serial.print(F("/")); }
        if ((strcmp(buffer,"GPZDA")==0) && auxint2==1){
            // $GPZDA,082710.00,04,07,2002,00,00*60
            //      01234567890123456789012345
            auxint=0;
            Serial.println(F("Sincronizando data"));
            while((auxchar=lergps())!='*'){
                Serial.print(F("+"));
                buffer[auxint]=auxchar;
                auxint++;
            }
            buffer2[0]=buffer[0];
            buffer2[1]=buffer[1];
            buffer2[2]='\0';
            relogio.hh=atoi(buffer2);
            buffer2[0]=buffer[2];
            buffer2[1]=buffer[3];
            buffer2[2]='\0';
            relogio.mm=atoi(buffer2);
            buffer2[0]=buffer[4];
            buffer2[1]=buffer[5];
            buffer2[2]='\0';
            relogio.ss=atoi(buffer2);
            buffer2[0]=buffer[11];
            buffer2[1]=buffer[12];
            buffer2[2]='\0';
            relogio.d=atoi(buffer2);
            buffer2[0]=buffer[14];
            buffer2[1]=buffer[15];
            buffer2[2]='\0';
            relogio.m=atoi(buffer2);
            buffer2[0]=buffer[17];
            buffer2[1]=buffer[18];
            buffer2[2]=buffer[19];
            buffer2[3]=buffer[20];
            buffer2[4]='\0';
            relogio.yOff=atoi(buffer2);
            relogio.yOff-=2000;
            RTC.adjust(relogio);
            sprintf(buffer,"%04d-%02d-%02d %02d:%02d:%02d", relogio.year(), relogio.month(),
            relogio.day(),relogio.hour(),relogio.minute(),relogio.second() );
            auxint3=1;
        }
    }
}
if (! RTC.isrunning()) {
    Serial.println(F("Problema com o relogio"));
    erro();
}
if (beep!=0) {
    delay(500);
    tocar();
}
tempoatual = (millis()/1000);
tempoamostra = tempoatual;

```

```

}

void loop(){
    tempoatual=(millis()/1000);
    if((tempoatual-tempoamostra)>=amostra){
        relogio = RTC.now();
        sprintf(buffer, "%02d%02d%02d.csv", relogio.month(), relogio.day(), relogio.hour());
        arquivo = SD.open(buffer,FILE_WRITE);
        sprintf(buffer,"%04d-%02d-%02d %02d:%02d:%02d;", relogio.year(), relogio.month(),
relogio.day(),relogio.hour(),relogio.minute(),relogio.second() );
        arquivo.print(buffer);
        Serial.println(F("Coletando"));
        for(auxint=0;auxint<9;auxint++){
            switch(listasensor[auxint][0]){
                case 1:
                    auxdouble=analogRead(listasensor[auxint][1]);
                    arquivo.print(auxdouble);
                    arquivo.print(";");
                    Serial.print(F("Analogico-"));
                    Serial.print(auxint);
                    Serial.print(F(": "));
                    Serial.println(auxdouble);
                    break;
                case 2:
                    auxint3=0;
                    while(auxint3<=2){
                        delay(450);
                        auxint2=DHT.read(listasensor[auxint][1]);
                        if (auxint2==0) auxint3=4;
                        else {
                            //delay(450);
                            auxint3++;
                        }
                    }
                    Serial.print(F("DHT22-"));
                    Serial.print(auxint);
                    Serial.print(F(": "));
                    if(auxint2==0){
                        arquivo.print(DHT.temperature);
                        arquivo.print(";");
                        arquivo.print(DHT.humidity);
                        arquivo.print(";");
                        Serial.print(DHT.temperature);
                        Serial.print(";");
                        Serial.println(DHT.humidity);
                    }
                    else {
                        Serial.print(F("Erro no sensor - "));
                        Serial.print(auxint2);
                        Serial.print(" - ");
                        Serial.print(DHT.temperature);
                        Serial.print(";");
                        Serial.println(DHT.humidity);
                        if (auxint2==1) arquivo.print("ww;ww");
                        else if (auxint2==2) arquivo.print("yy;yy;");
                        else arquivo.print("xx;xx;");
                        if (beep!=0) for (auxint3=0;auxint3<auxint;auxint3++){ tocar(); }
                    }
                }
            }
        }
    }
}

```

```

break;
case 4:
  Serial.print(F("GPS: "));
  auxint2=1;
  while(auxint2){
    gps.flush();
    auxchar=lergps();
    if(auxchar=='$'){
      buffer[0]=lergps();
      buffer[1]=lergps();
      buffer[2]=lergps();
      buffer[3]=lergps();
      buffer[4]=lergps();
      buffer[5]="'0';
      auxchar=lergps();
      if(strcmp(buffer,"GPGGA")==0){
        auxint3=0;
        // $GPGGA,083559.00,3723.2475,N, 12158.3416,W, 1,07,1.0,9.0,M, ,M, ,0000*18
        while(auxint3<=9){
          auxchar=lergps();
          if(auxchar==','){ arquivo.print(','); Serial.print(','); auxint3++; }
          else { arquivo.print(auxchar); Serial.print(auxchar); }
        }
        auxint2=0;
      }
    }
  }
  Serial.println("");
  break;
case 5:
  auxdouble=analogRead(listasensor[auxint][1]);
  auxdouble=( 5 * auxdouble ) / 326;
  arquivo.print(auxdouble);
  arquivo.print(",");
  Serial.print(F("Bateria: "));
  Serial.println(auxdouble);
  break;
}
arquivo.println();
arquivo.close();
tempoamostra=tempoatual;
if (beep!=0) {
  tocar();
}
//Serial.println(freeRam()-202);
}
}

```

ANEXO B – PARTE DO MANUAL DO PROTOCOL NMEA

*Leadtek NMEA Protocol Manual
For MTK*

**Version History**

Date	Version	Author	Comments
01/06/2009	1.0	Lunglin Shr	Initial version



1.4.18 Packet Type: 513 PMTK_DT_SBAS_ENABLED

Packet Meaning:

Enable to search a SBAS satellite or not.

DataField:

Enabled: Enable or disable

'0' = Disable

'1' = Enable

Example:

\$PMTK513,1*28<CR><LF>

1.4.19 Packet Type: 514 PMTK_DT_NMEA_OUTPUT

Packet Meaning:

NMEA sentence output frequency setting

DataField:

There are totally 19 data fields that present output frequencies for the 19 supported NMEA sentences individually.

Please refer to PMTK_API_SET_NMEA_OUTPUT for the Supported NMEA Sentences and Frequency Setting.

Example:

\$PMTK514,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0*2E<CR><LF>

1.4.20 Packet Type: 530 PMTK_DT_DATUM

Packet Meaning:

Current datum used.

DataField:

PMTK530,Datum

Datum:

0: WGS84

1: TOKYO-M

2: TOKYO-A

Example:

\$PMTK530,0*28<CR><LF>



◆ Appendix A: Datum List

No	Datum	Region
0	WGS1984	International
1	Tokyo	Japan
2	Tokyo	Mean For Japan, South Korea, Okinawa
3	User Setting	User Setting
4	Adindan	Burkina Faso
5	Adindan	Cameroon
6	Adindan	Ethiopia
7	Adindan	Mali
8	Adindan	Mean For Ethiopia, Sudan
9	Adindan	Senegal
10	Adindan	Sudan
11	Afgooye	Somalia
12	Ain El Abd1970	Bahrain
13	Ain El Abd1970	Saudi Arabia
14	American Samoa1962	American Samoa Islands
15	Anna 1 Astro1965	Cocos Island
16	Antigua Island Astro1943	Antigua (Leeward Islands)
17	Arc1950	Botswana
18	Arc1950	Burundi
19	Arc1950	Lesotho
20	Arc1950	Malawi
21	Arc1950	Mean For Botswana, Lesotho, Malawi, Swaziland, Zaire, Zambia, Zimbabwe
22	Arc1950	Swaziland
23	Arc1950	Zaire
24	Arc1950	Zambia
25	Arc1950	Zimbabwe
26	Arc1960	Mean For Kenya Tanzania
27	Arc1960	Kenya
28	Arc1960	Tamzamia
29	Ascension Island1958	Ascension Island
30	Astro Beacon E 1945	Iwo Jima
31	Astro Dos 71/4	St Helena Island
32	Astro Tern Island (FRIG) 1961	Tern Island
33	Astronomical Station 1952	Marcus Island
34	Australian Geodetic 1966	Australia, Tasmania



No	Datum	Region
35	Australian Geodetic 1984	Australia, Tasmania
36	Ayabelle Lighthouse	Djibouti
37	Bellevue (IGN)	Efate and Erromango Islands
38	Bermuda 1957	Bermuda
39	Bissau	Guinea-Bissau
40	Bogota Observatory	Colombia
41	Bukit Rimpah	Indonesia(Bangka and Belitung Ids)
42	Camp Area Astro	Antarctica(McMurdi Camp Area)
43	Campo Inchauspe	Argentina
44	Canton Astro1966	Phoenix Island
45	Cape	South Africa
46	Cape Canaveral	Bahamas, Florida
47	Carthage	Tunisia
48	Chatham Island Astro1971	New Zealand(Chatham Island)
49	Chua Astro	Paraguay
50	Corrego Alegre	Brazil
51	Dabola	Guinea
52	Deception Island	Deception Island, Antarctica
53	Djakarta (Batavia)	Indonesia(Sumatra)
54	Dos 1968	New Georgia Islands (Gizo Island)
55	Easter Island 1967	Easter Island
56	Estonia Coordinate System1937	Estonia
57	European 1950	Cyprus
58	European 1950	Egypt
59	European 1950	England, Channel Islands, Scotland, Shetland Islands
60	European 1950	England, Ireland, Scotland, Shetland Islands
61	European 1950	Finland, Norway
62	European 1950	Greece
63	European 1950	Iran
64	European 1950	Italy (Sardinia)
65	European 1950	Italy (Sicily)
66	European 1950	Malta



No	Datum	Region
67	European 1950	Mean For Austria, Belgium, Denmark, Finland, France, Germany, Gibraltar, Greece, Italy, Luxembourg, Netherlands, Norway, Portugal, Spain, Sweden, Switzerland
68	European 1950	Mean For Austria, Denmark, France, Germany, Netherland , Switzerland
69	European 1950	Mean For Iraq, Israel, Jordan, Lebanon, Kuwait, Saudi, Arabia, Syria
70	European 1950	Portugal, Spain
71	European 1950	Tunisia,
72	European 1979	Mean For Austria, Finland ,Netherlands , Norway, Spain, Sweden, Switzerland
73	Fort Thomas 1955	Nevis St Kitts (Leeward Islands)
74	Gan 1970	Republic Of Maldives
75	Geodetic Dataum 1970	New Zealand
76	Graciosa Base SW1948	Azores (Faial, Graciosa, Pico, Sao, Jorge, Terceria)
77	Guam1963	Guam
78	Gunung Segara	Indonesia (Kalimantan)
79	Gux 1 Astro	Guadalcanal Island
80	Herat North	Afghanistan
81	Hermannskogel Datum	Croatia-Serbia, Bosnia-Herzegovina
82	Hjorsey 1955	Iceland
83	Hongkong 1963	Hongkong
84	Hu Tzu Shan	Taiwan
85	Indian	Bangladesh
86	Indian	India,Nepal
87	Indian	Pakistan
88	Indian 1954	Thailand
89	Indian 1960	Vietnam (Con Son Island)
90	Indian 1960	Vietnam (Near 16 deg N)
91	Indian 1975	Thailand
92	Indonesian 1974	Indonesian
93	Ireland 1965	Ireland
94	ISTS 061 Astro 1968	South Georgia Islands



No	Datum	Region
95	ISTS 073 Astro 1969	Diego Garcia
96	Johnston Island 1961	Johnston Island
97	Kandawala	Sri Lanka
98	Kerguelen Island 1949	Kerguelen Island
99	Kertau 1948	West Malaysia and Singapore
100	Kusaie Astro 1951	Caroline Islands
101	Korean Geodetic System	South Korea
102	LC5 Astro 1961	Cayman Brac Island
103	Leigon	Ghana
104	Liberia 1964	Liberia
105	Luzon	Philippines (Excluding Mindanao)
106	Luzon	Philippines (Mindanao)
107	M'Poraloko	Gabon
108	Mahe 1971	Mahe Island
109	Massawa	Ethiopia (Eritrea)
110	Merchich	Morocco
111	Midway Astro 1961	Midway Islands
112	Minna	Cameroon
113	Minna	Nigeria
114	Montserrat Island Astro 1958	Montserrat (Leeward Island)
115	Nahrwan	Oman (Masirah Island)
116	Nahrwan	Saudi Arabia
117	Nahrwan	United Arab Emirates
118	Naparima BWI	Trinidad and Tobago
119	North American 1927	Alaska (Excluding Aleutian Ids)
120	North American 1927	Alaska (Aleutian Ids East of 180 degW)
121	North American 1927	Alaska (Aleutian Ids West of 180 degW)
122	North American 1927	Bahamas (Except San Salvador Islands)
123	North American 1927	Bahamas (San Salvador Islands)
124	North American 1927	Canada (Alberta, British Columbia)
125	North American 1927	Canada (Manitoba, Ontario)
126	North American 1927	Canada (New Brunswick, Newfoundland, Nova Scotia, Quebec)
127	North American 1927	Canada (Northwest Territories, Saskatchewan)
128	North American 1927	Canada (Yukon)



No	Datum	Region
129	North American 1927	Canal Zone
130	North American 1927	Cuba
131	North American 1927	Greenland (Hayes Peninsula)
132	North American 1927	Mean For Antigua, Barbados, Barbuda, Caicos Islands, Cuba, Dominican, Grand Cayman, Jamaica, Turks Islands
133	North American 1927	Mean For Belize, Costa Rica, El Salvador, Guatemala, Honduras, Nicaragua
134	North American 1927	Mean For Canada
135	North American 1927	Mean For Conus
136	North American 1927	Mean For Conus (East of Mississippi, River Including Louisiana, Missouri, Minnesota)
137	North American 1927	Mean For Conus (West of Mississippi, Rive Excluding Louisiana, Minnesota, Missouri)
138	North American 1927	Mexico
139	North American 1983	Alaska (Excluding Aleutian Ids)
140	North American 1983	Aleutian Ids
141	North American 1983	Canada
142	North American 1983	Conus
143	North American 1983	Hawaii
144	North American 1983	Mexico, Central America
145	North Sahara 1959	Algeria
146	Observatorio Meteorologico 1939	Azores (Corvo and Flores Islands)
147	Old Egyptian 1907	Egypt
148	Old Hawaiian	Hawaii
149	Old Hawaiian	Kauai
150	Old Hawaiian	Maui
151	Old Hawaiian	Mean For Hawaii, Kauai, Maui, Oahu
152	Old Hawaiian	Oahu
153	Oman	Oman
154	Ordnance Survey Great Britain 1936	England



No	Datum	Region
155	Ordnance Survey Great Britian 1936	England, Isle of Man, Wales
156	Ordnance Survey Great Britian 1936	Mean For England ,Isle of Man, Scotland, Shetland Island,Wales
157	Ordnance Survey Great Britian 1936	Scotland, Shetland Islands
158	Ordnance Survey Great Britian 1936	Wales
159	Pico de las Nieves	Canary Islands
160	Pitcairn Astro 1967	Pitcairn Island
161	Point 58	Mean For Burkina Faso and Niger
162	Pointe Noire 1948	Congo
163	Porto Santo 1936	Porto Santo, Maderia Islands
164	Provisional South American 1956	Bolivia
165	Provisional South American 1956	Chile (Northern Near 19 deg S)
166	Provisional South American 1956	Chile (Southern Near 43 deg S)
167	Provisional South American 1956	Colombia
168	Provisional South American 1956	Ecuador
169	Provisional South American 1956	Guyana
170	Provisional South American 1956	Mean For Bolivia Chile,Colombia, Ecuador, Guyana, Peru,Venezuela
171	Provisional South American 1956	Peru
172	Provisional South American 1956	Venezuela
173	Provisional South Chilean 1963	Chile (Near 53 deg S) (Hito XVIII)
174	Puerto Rico	Puerto Rico, Virgin Islands
175	Pulkovo 1942	Russia
176	Qatar National	Qatar
177	Qornoq	Greenland (South)



No	Datum	Region
178	Reunion	Mascarene Island
179	Rome 1940	Italy (Sardinia)
180	S-42 (Pulkovo 1942)	Hungary
181	S-42 (Pulkovo 1942)	Poland
182	S-42 (Pulkovo 1942)	Czechoslovakia
183	S-42 (Pulkovo 1942)	Lativa
184	S-42 (Pulkovo 1942)	Kazakhstan
185	S-42 (Pulkovo 1942)	Albania
186	S-42 (Pulkovo 1942)	Romania
187	S-JTSK	Czechoslovakia (Prior 1 Jan1993)
188	Santo (Dos) 1965	Espirito Santo Island
189	Sao Braz	Azores (Sao Miguel, Santa Maria Ids)
190	Sapper Hill 1943	East Falkland Island
191	Schwarzeck	Namibia
192	Selvagem Grande 1938	Salvage Islands
193	Sierra Leone 1960	Sierra Leone
194	South American 1969	Argentina
195	South American 1969	Bolivia
196	South American 1969	Brazil
197	South American 1969	Chile
198	South American 1969	Colombia
199	South American 1969	Ecuador
200	South American 1969	Ecuador(Baltra, Galapagos)
201	South American 1969	Guyana
202	South American 1969	Mean For Argentina, Bolivia, Brazil, Chile, Colombia, Ecuador, Guyana, Paraguay, Peru, Trinidad and Tobago, Venezuela
203	South American 1969	Paraguay
204	South American 1969	Peru
205	South American 1969	Trinidad and Tobago
206	South American 1969	Venezuela
207	South Asia	Singapore
208	Tananarive Observatory 1925	Madagascar
209	Timbalai 1948	Brunei, E Malaysia (Sabah Sarawak)
210	Tokyo	Japan



No	Datum	Region
211	Tokyo	Mean For Japan, South Korea, Okinawa
212	Tokyo	Okinawa
213	Tokyo	South Korea
214	Tristan Astro 1968	Tristam Da Cunha
215	Viti Levu 1916	Fiji (Viti Levu Island)
216	Voirol 1960	Algeria
217	Wake Island Astro 1952	Wake Atoll
218	Wake-Eniwetok 1960	Marshall Islands
219	WGS 1972	Global Definition
220	WGS 1984	Global Definition
221	Yacare	Uruguay
222	Zanderij	Suriname

ANEXO C – A HISTÓRIA DO ARDUINO

Na cidade de Ivrea, que fica às margens do rio Dora Baltea na cidade de Turim, ao Norte da Itália, famosa por seus reis azarões. Em 1002, o Rei Arduin foi o governante do país, apenas para ser destronado pelo rei Henrique II, da Alemanha, dois anos depois. Hoje, o Bar *di Re Arduino* se tornou-se um pub em uma rua de paralelepípedos na cidade, homenageando sua memória, e foi aí que um novo rei improvável nasceu (KUSHNER, 2013).

O bar era uma cisterna de Massimo Banzi, o co-fundador italiano do projeto de eletrônica que deu o nome de *Arduino* em homenagem ao lugar.

Lançado em 2005, como uma ferramenta para os estudantes de Banzi no *Interaction Design Institute Ivrea* (IDII), o *Arduino* gerou uma revolução na eletrônica internacional com o conceito faça-você-mesmo. Com isso, pode-se comprar uma placa *Arduino* ou construir o seu próprio a partir do zero: todos os esquemas de *hardware* e código-fonte estão disponíveis gratuitamente sob licenças públicas. Como resultado, o *Arduino* se tornou não apenas um, mas o movimento de *hardware open-source* mais influente do seu tempo (KUSHNER, 2013).

A placa é, atualmente, usada para criar equipamentos para artistas, amadores, estudantes e qualquer pessoa com um sonho de umas engenhocas. Até 2012, mais de 250.000 placas *Arduino* foram vendidas em todo o mundo, e isso não inclui os projetos clones. "Isso permitiu que as pessoas fizessem coisas que não teriam feito de outra forma", diz David A. Mellis, que era um estudante na *Interaction Design Institute Ivrea* antes de prosseguir o trabalho de pós-graduação no *MIT Media Lab*. e hoje é um dos desenvolvedores do *software* principal do *Arduino*.

Mas *Arduino* não é apenas um projeto *open-source* que visa tornar a tecnologia mais acessível, é também uma empresa *start-up*, dirigida por Banzi e um grupo de amigos, e está enfrentando um desafio: como sobreviver ao sucesso e crescer. "Precisamos fazer o próximo salto", diz Banzi, "e se tornar uma empresa estabelecida".

O *Arduino* assumiu um novo desafio: como ensinar os alunos a criar projetos eletrônicos de forma rápida. Em 2002, Banzi foi aprovado como professor associado no *Interaction Design Institute Ivrea* para promover novas maneiras de fazer um

projeto interativo. Mas, com um orçamento limitado e pouco tempo excedente para as aulas, suas possibilidades eram poucas (KUSHNER, 2013).

No início, Banzi contou com o *BASIC Stamp*²⁶, sendo ele um micro controlador criado pela empresa californiana *Parallax*, o qual tinha utilizado durante cerca de uma década, embutindo nele o essencial: uma fonte de alimentação, um micro controlador, uma memória e portas de entrada/saída para conectar novos *hardwares*.

Mas o *BASIC Stamp*, segundo Banzi, tinha dois problemas: não tem poder de processamento suficiente para alguns dos projetos que seus alunos tinham em mente e era um pouco caro. Ele também precisava de algo que pudesse ser executado em computadores Macintosh, que eram onipresentes entre os designers do *Interaction Design Institute Ivrea*.

Para tentar resolver tal necessidade, Banzi lembrou que tinha um colega do MIT que havia desenvolvido uma linguagem de programação mais amigável (*designer-friendly*), chamada de *Processing*²⁷, que ganhou popularidade, permitindo que programadores inexperientes criassem programas complexos e com um melhor visual. Uma das razões para o seu sucesso foi um ambiente de desenvolvimento (IDE) fácil de usar.

Banzi perguntou se eles poderiam criar um *software* semelhante para um microcontrolador, e então, um estudante do programa, Hernando Barragán, deu os primeiros passos nessa direção. Ele desenvolveu uma plataforma de prototipagem chamada de *Wiring*²⁸, que incluiu uma IDE amigável e uma placa de circuito impressa pronta para usar (Figura 01). Era o começo de um projeto promissor que continua até hoje, entretanto, Banzi já estava pensando em um projeto maior, ele queria fazer uma plataforma que era ainda mais simples, mais barata e mais fácil de usar (KUSHNER, 2013).

²⁶ Disponível em: <<http://www.parallax.com/tabid/295/default.aspx>>.

²⁷ Disponível em: <<http://www.processing.org/>>.

²⁸ Disponível em: <<http://wiring.org.co/>>.

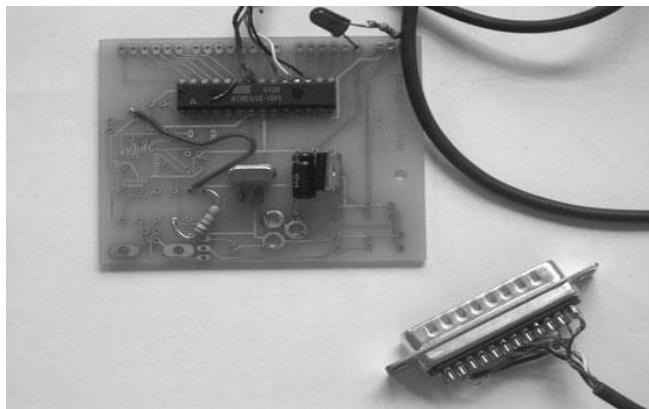


Figura 01 – Primeiro protótipo do Arduino

Fonte: BANZI (2013).

Banzi e seus colaboradores eram adeptos ao *software open-source*. Como o objetivo foi criar uma plataforma de rápido e fácil acesso, eles perceberam que seria melhor abrir o projeto para o maior número possível de pessoas, ao invés de mantê-lo fechado. Outro fator que contribuiu para essa decisão foi que, depois de operar por cinco anos, IDII estava ficando sem fundos e iria encerrar suas atividades. Com isso, os Membros do corpo docente temiam que seus projetos não fossem sobreviver ou seriam desviados e, então, eles tornaram o projeto *open-source* (KUSHNER, 2013).

O modelo *open-source* há tempos vem sendo usado para estimular a inovação de *software*, mas não de *hardware*. Para fazê-lo funcionar, eles tiveram que encontrar uma solução de licenciamento apropriado, que poderia ser aplicada a placa. Depois de algumas investigações, eles perceberam que se eles simplesmente olhassem para o seu projeto de forma diferente, eles poderiam usar uma licença da *Creative Commons*²⁹, que é uma organização sem fins lucrativos, que possibilita o compartilhamento e uso da criatividade e do conhecimento por meio de instrumentos legais livres, sendo ela uma alternativa à licença provida pela *copyright*³⁰.

Dessa forma, o *hardware* desenvolvido por Banzi não se enquadrava nesses critérios e foi aí que ele pensou na seguinte saída, “Você poderia pensar em um *hardware* como parte de uma cultura que você deseja compartilhar com outras pessoas” (KUSHNER, 2013), e então conseguiu um argumento plausível para justificar o registro do equipamento.

²⁹ Disponível em: <<http://creativecommons.org/>>.

³⁰ Disponível em: <<http://copyright.br.com/>>.

Após o registro, o grupo teve como objetivo e meta o preço final do *Arduino* para o estudante, que teria que ser na faixa dos US\$ 30,00: “Tinha que ser o equivalente a sair para jantar em uma pizzaria”, disse Banzi (KUSHNER, 2013). Eles também queriam fazê-lo peculiar, algo que iria se destacar e ser bacana para estudantes, curiosos e pesquisadores.

O produto criado pela equipe consistia em peças baratas que poderiam ser facilmente encontradas se os usuários quisessem construir suas próprias placas, como o micro controlador ATmega328. Mas, uma decisão importante era garantir que seria essencialmente *plug-and-play*, algo que alguém poderia tirar de uma caixa, ligar a um computador e usá-lo imediatamente. “A filosofia por trás do *Arduino* é que, se você quer aprender eletrônica, você deverá ser capaz de aprender sozinho em um dia, em vez de começar a aprender álgebra”, disse outro membro da equipe, o engenheiro de telecomunicações David Cuartielles (KUSHNER, 2013).

A equipe colocou essa filosofia em prática. Eles entregaram 300 placas de circuito impresso em branco para os alunos da IDII com uma diretriz básica: procure as instruções de montagem, construa sua própria placa e use-a para alguma coisa. Um dos primeiros projetos foi um despertador caseiro que era preso no teto por um cabo. Sempre que se apertasse o botão de soneca, o relógio subiria mais alto no ar, até que você tivesse que levantar.

Com a popularização dessas histórias, o primeiro cliente foi um amigo de Banzi, que fez o primeiro pedido, e assim começou o projeto; mas uma coisa importante estava faltando, um nome para a sua invenção. Em uma noite de bebedeira no pub, eles tiveram a ideia de colocar o nome do projeto como “*Arduino*”, assim como o bar do rei (KUSHNER, 2013).