

Serviço Nacional de Aprendizagem Comercial
Faculdade Senac Porto Alegre
Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas

ALEXANDRE LEHMANN HOLZHEY

RELATÓRIO FINAL DE PROJETO

VOO SEGURO
Estação Climática Privada utilizando Arduino

Porto Alegre
2014

ALEXANDRE LEHMANN HOLZHEY

RELATÓRIO FINAL DE PROJETO

VOO SEGURO

Estação Climática Privada utilizando Arduino

Relatório Final de Projeto, apresentado como requisito parcial à obtenção do grau de Tecnólogo em Análise e Desenvolvimento de Sistemas, pela Faculdade Senac Porto Alegre.

Orientador: Prof. Me. Rafael Guterres Jeffman

Porto Alegre
2014

AGRADECIMENTOS

A meu pai, por ter me incentivado no despertar do pequeno cientista que tenho dentro de mim.

A minha mãe pelo carinho sempre presente.

A Suélen Fernanda Bavaresco, por ter me apoiado e incentivado a buscar pelos meus ideais e sonhos e a acreditar em meu potencial.

Ao meu orientador, Prof. Me. Rafael Guterres Jeffman, pela sua capacidade técnica e didática que muito me auxiliou neste trabalho.

Ao coordenador do curso de Análise e Desenvolvimento de Sistemas, Prof. Me. Guilherme Bertoni Machado, pela ajuda e apoio durante a etapa de decisão do tema deste trabalho.

A todos os professores do Senac, pois cada um do seu jeito e em sua disciplina colaborou para o conhecimento que tenho hoje.

RESUMO

A preparação para o voo é um procedimento fundamental na aviação que visa avaliar a viabilidade da operação aeronáutica, bem como promover a segurança do voo. Neste procedimento, as informações meteorológicas são cuidadosamente analisadas, principalmente aquelas que se referem aos locais de decolagem e pouso do avião. No Brasil estas informações são fornecidas pela Rede de Meteorologia do Comando da Aeronáutica. Pequenos aeródromos e escolas de instrução de voo desprovidos desta informação, utilizam dados de locais distantes e que não correspondem a situação climática local. Através do uso de hardware de baixo custo, é possível criar uma estação meteorológica automática capaz de fornecer os dados necessários aos locais citados, proporcionando a informação localizada para os pequenos aeroclubes e escolas de aviação. A solução neste projeto irá auxiliar o ensino nas escolas de aviação, assim como aumentar a segurança de voo nesses locais. A informação gerada será disponibilizada localmente em um *display* LCD e remotamente através da Internet, disponibilizando os dados para pilotos em locais distantes e colaborando com a comunidade de meteorologistas que utilizam este tipo de informação.

Palavras-chave: Arduino. Sistemas Embarcados. Segurança de voo.

LISTA DE ILUSTRAÇÕES

Figura 1 - Arduino Uno R2.....	14
Figura 2 - Bosch BMP085.....	16
Figura 3 - Aosong AM2302.....	16
Figura 4 - Rohm BH1750FVI.....	17
Figura 5 - Sensor de velocidade do vento.....	18
Figura 6 - Módulo de comunicação GSM.....	18
Figura 7 - Módulo de comunicação I2C para LCD.....	19
Figura 8 - Relógio DS1307.....	19
Figura 9 - Arquitetura I2C.....	20
Figura 10 - Funcionamento da solução.....	21
Figura 11 - Desenvolvimento prototipado.....	27
Figura 12 - Caso de uso - uso local.....	33
Figura 13 - Caso de uso - uso remoto.....	33
Figura 14 - Caso de uso - configuração.....	34
Figura 15 - Diagrama esquemático.....	34
Figura 16 - Implementação da tarefa relativa ao vento.....	41
Figura 17 - Tabela de declinação do sol e equação do tempo.....	43
Figura 18 - Fórmula para estimar a altitude do sol em graus.....	43
Figura 19 - Tabela de conversão de altitude para massa de ar.....	44
Figura 20 - Curvas de exemplo do resultado dos cálculos de luminosidade.....	44
Figura 21 - Fórmula do ponto de orvalho.....	45
Figura 22 - Implementação da comunicação.....	46
Figura 23 - hardware Voo Seguro.....	47
Figura 24 - Tela de inicialização.....	47
Figura 25 - Tela de verificação.....	48
Figura 26 - Tela de monitoramento.....	48
Figura 27 - Tela de configuração do dia da data atual.....	49
Figura 28 - Tela de configuração do mês da data atual.....	49
Figura 29 - Tela de configuração do ano da data atual.....	49
Figura 30 - Tela de configuração da hora da data atual.....	50
Figura 31 - Tela de configuração do minuto da data atual.....	50
Figura 32 - Tela de configuração do código ICAO.....	50
Figura 33 - Tela de configuração da latitude.....	51
Figura 34 - Tela de configuração da longitude.....	51
Figura 35 - Consulta remota.....	52

LISTA DE SIGLAS

ANAC	Agência Nacional de Aviação Civil
API	<i>Application Program Interface</i>
APN	<i>Access Point Name</i>
CMA	Centro Meteorológico de Aeródromo
DECEA	Departamento de Controle do Espaço Aéreo
EEPROM	<i>Electrically Erasable Programmable Read-Only Memory</i>
EMA	Estação Meteorológica de Altitude
EMS	Estação Meteorológica de Superfície
ERM	Estação de Radar Meteorológico
FTP	<i>File Transfer Protocol</i>
GPRS	<i>General Packet Radio Service</i>
GSM	<i>Global System for Mobile Communications</i>
HTTP	<i>Hypertext Transfer Protocol</i>
I2C	<i>Inter-Integrated Circuit</i>
ICAO	<i>International Civil Aviation Organization</i>
IDE	<i>Integrated Development Environment</i>
LED	<i>Light Emitter Diode</i>
METAR	<i>Meteorological Aerodrome Report</i>
OMM	Organização Meteorológica Mundial
PWS	<i>Personal Weather Station</i>
RBHA	Regulamento Brasileiro de Homologação Aeronáutica
REDEMET	Rede de Meteorologia do Comando da Aeronáutica
REM	Rede de Estações Meteorológicas
SCL	<i>Serial Clock</i>
SDA	<i>Serial Data Line</i>
SRAM	<i>Static Random Access Memory</i>
URL	<i>Uniform Resource Locator</i>
USB	<i>Universal Serial Bus</i>
UTC	<i>Universal Time Coordinated</i>

LISTA DE SÍMBOLOS

KB	Unidade que representa 1024 <i>bytes</i>
Mhz	Milhões de Hertz
cm	Centímetro
°	Graus
°C	Graus Celsius
hPa	Hectopascais
Lux	Unidade de iluminamento
KT	Knots, velocidade do vento em nós
pF	Picofarad
bps	Bits por segundo

SUMÁRIO

1.	APRESENTAÇÃO GERAL DO PROJETO	9
2.	DEFINIÇÃO DO PROBLEMA.....	10
2.1.	METEOROLOGIA	10
2.2.	EXATIDÃO DA INFORMAÇÃO	10
2.3.	ESTAÇÃO METEOROLÓGICA	11
3.	OBJETIVOS	13
3.1.	OBJETIVO GERAL.....	13
3.2.	OBJETIVOS ESPECÍFICOS	13
4.	ANÁLISE DE TECNOLOGIAS/FERRAMENTAS	14
4.1.	ARDUINO	14
4.2.	INTERFACE DE PROGRAMA DE APLICATIVO WUNDERGROUND	15
4.3.	SENSOR DE TEMPERATURA E PRESSÃO ATMOSFÉRICA.....	15
4.4.	SENSOR DE UMIDADE RELATIVA DO AR.....	16
4.5.	SENSOR DE LUMINOSIDADE.....	17
4.6.	SENSOR DE VELOCIDADE DO VENTO	17
4.7.	COMPONENTE GSM EFCOM PRO V1.0.....	18
4.8.	CONVERSOR I2C.....	18
4.9.	RELÓGIO DE TEMPO REAL	19
4.10.	PROTOCOLO I2C	19
5.	DESCRIÇÃO DA SOLUÇÃO	21
5.1.	MÓDULO DE COLETA E PROCESSAMENTO	22
5.2.	O FORMATO METAR	22
5.3.	ADEQUAÇÃO DOS DADOS.....	24
5.4.	WUNDERGROUND API.....	25
5.5.	EXIBIÇÃO LOCAL.....	26
6.	ABORDAGEM DE DESENVOLVIMENTO	27
6.1.	CICLOS DE DESENVOLVIMENTO	28
6.1.1.	Ciclo de desenvolvimento 1.....	28
6.1.2.	Ciclo de desenvolvimento 2.....	28
6.1.3.	Ciclo de desenvolvimento 3.....	28
6.1.4.	Ciclo de desenvolvimento 4.....	28
6.1.5.	Ciclo de desenvolvimento 5.....	28
6.1.6.	Ciclo de desenvolvimento 6.....	29
6.1.7.	Ciclo de desenvolvimento 7.....	29
6.1.8.	Ciclo de desenvolvimento 8.....	29
6.1.9.	Ciclo de desenvolvimento 9.....	29
6.1.10.	Ciclo de desenvolvimento 10.....	29
6.1.11.	Ciclo de desenvolvimento 11.....	29
6.1.12.	Ciclo de desenvolvimento 12.....	29
6.1.13.	Ciclo de desenvolvimento 13.....	29
6.1.14.	Ciclo de desenvolvimento 14.....	30
7.	ARQUITETURA DO SISTEMA	31
7.1.	REQUISITOS.....	31
7.1.1.	Captura das condições climáticas	31

7.1.2.	Processamento dos dados coletados.....	31
7.1.3.	Exibição e transmissão dos dados processados.....	32
7.2.	CASOS DE USO.....	32
7.2.1.	Uso local	32
7.2.2.	Uso remoto	33
7.2.3.	Configuração.....	34
7.3.	HARDWARE	34
7.4.	FIRMWARE.....	35
7.4.1.	Bibliotecas utilizadas.....	35
7.4.2.	Textos estáticos	36
7.4.3.	Inicialização das bibliotecas	37
7.4.4.	Temporizador de tarefas	37
7.4.5.	Variáveis globais	37
7.4.6.	Inicialização	37
7.4.7.	Funções primárias e de apoio.....	39
7.4.8.	Laço principal.....	39
7.5.	TRATAMENTO DOS DADOS.....	39
7.5.1.	Velocidade do vento	40
7.5.2.	Rajadas do vento	41
7.5.3.	Cobertura de nuvens	42
7.5.4.	Temperatura de ponto de orvalho.....	44
7.6.	COMUNICAÇÃO.....	45
8.	FUNCIONAMENTO DO SISTEMA.....	47
8.1.	INICIALIZAÇÃO	47
8.2.	MONITORAMENTO	48
8.3.	CONFIGURAÇÃO	48
8.4.	CONSULTA REMOTA.....	51
9.	VALIDAÇÃO.....	53
9.1.	ESTRATÉGIA.....	53
9.2.	CONSOLIDAÇÃO DOS DADOS COLETADOS	54
9.3.	CUSTOS.....	55
10.	CONSIDERAÇÕES.....	57
	REFERÊNCIAS.....	59
	COMPONENTES REUTILIZADOS	62
	APÊNDICE A - MODELO DE PLANILHA DE VALIDAÇÃO	63
	APÊNDICE B - PLANILHA DE VALIDAÇÃO: PERÍODO 1	64
	APÊNDICE C - PLANILHA DE VALIDAÇÃO: PERÍODO 2	66
	APÊNDICE D - PLANILHA DE VALIDAÇÃO: PERÍODO 3	68
	ANEXO A - ORÇAMENTOS DE ANEMÔMETROS.....	70

1. APRESENTAÇÃO GERAL DO PROJETO

De acordo com a Agência Nacional de Aviação Civil (ANAC), o reconhecimento e gerenciamento de ameaças e erros é premissa para a concessão de licença de piloto de avião, sendo as condições climáticas um dos fatores que devem ser avaliados antes e durante cada voo, afirma a ANAC (2013). Segundo dados da Boeing (2012), dos acidentes fatais com aeronaves ocorridos entre 2002 e 2011, 16% ocorreram na fase de decolagem e subida inicial, enquanto 37% manifestaram-se nas fases de aproximação final e pouso, o que revela a importância quanto a preparação relativa a estas fases do voo. O Departamento de Controle do Espaço Aéreo (DECEA), órgão que planeja, gerencia e controla as atividades relacionadas ao controle do espaço aéreo brasileiro, salienta a importância da informação meteorológica nas fases do voo mencionadas anteriormente, sendo que todo o piloto de avião deve buscar as informações climáticas relativas ao voo antes dele ser iniciado (DECEA, 2013).

Apesar da existência de uma Rede de Estações Meteorológicas (REM), que disponibiliza as condições climáticas dos aeródromos confeccionadas pelos Centros Meteorológicos de Aeródromo (CMA), os pequenos aeródromos que usualmente abrigam escolas de aviação não estão interligados a esta rede, não dispondo deste tipo de informação. Para a geração de seus produtos e serviços, os CMAs utilizam dados obtidos através de Estações Meteorológicas de Superfície (EMS), de Estações Meteorológicas de Altitude (EMA) e de Estações de Radar Meteorológico (ERM), o qual compõem um sistema de complexos equipamentos, com custo muito elevado para sua adoção por pequenos aeródromos. Por exemplo, em 2008 o aeroporto José Coletto no Paraná foi beneficiado com os equipamentos e serviços necessários para constar na REM, com um custo total estimado em US\$ 300.000,00 (trezentos mil dólares) (GENTE DE OPINIÃO, 2008). O custo aproximado para uma estação homologada pela *Federal Aviation Association* (FAA), autônoma e mais simples que uma EMS completa, é, nos Estados Unidos da América, de US\$ 75.000,00 (setenta e cinco mil dólares) (AIRPORT IMPROVEMENT, 2011).

Tal condição não contribui para a preparação de voo efetuada por alunos em formação nestas instituições, os quais baseiam-se em informações aproximadas oriundas de aeródromos localizados muitas vezes a dezenas ou até centenas de quilômetros de distância. A carência de uma rede de informações meteorológicas específica para os aeroclubes e escolas de aviação é notória e instiga o aluno a buscar por estas informações em alternativas como o *Weather Underground* (WUNDERGROUND), que possui dados meteorológicos de estações particulares muitas vezes próximas aos locais desejados pelos pilotos em formação (WUNDERGROUND, 2014).

Este projeto especifica a construção de uma estação meteorológica de baixo custo que disponibiliza os dados climáticos coletados em aeródromo não coberto pela rede REM, ou seja, que não possuem um CMA. Esta ferramenta permitirá ao piloto aluno um planejamento melhor de seu voo bem como uma atualização das condições climáticas no regresso ao local de partida. Visando utilizar recursos já existentes, os dados serão disponibilizados também através do WUNDERGROUND, contribuindo com a comunidade de meteorologistas profissionais e amadores que utilizam este recurso para busca de informações, bem como propiciar uma forma remota de consulta as condições meteorológicas.

2.DEFINIÇÃO DO PROBLEMA

O problema abordado é a falta de informações meteorológicas, no formato adequado para o âmbito aeronáutico, nos aeródromos que não são atendidos por um CMA. Este formato é específico para o meio aeronáutico e propicia um resumo objetivo das condições meteorológicas atuais para o local avaliado. Estas informações são relevantes para o piloto, pois permitem avaliar a viabilidade ou não do voo, conforme critérios adotados pelo piloto e/ou empresa para a qual trabalha, bem como de acordo com os parâmetros operacionais da aeronave. A localização destas informações é importante, visto que o uso de dados gerados por um CMA distante pode acarretar em uma avaliação incorreta devido a fatores que promovam alterações no clima, como o relevo. O custo de uma estação oficial, desconsiderando-se a complexidade burocrática envolvida, é elevado e somente uma solução de baixo custo pode viabilizar atender esta demanda, existente em aeroclubes e escolas de aviação no Brasil.

2.1.METEOROLOGIA

A meteorologia é uma das disciplinas que compõe a formação teórica para o curso de piloto privado de avião, conforme determinação do Comando da Aeronáutica (2004). Nela são abordados vários conceitos relativos ao comportamento da atmosfera e seu impacto no avião, assim como as formas de prevenção contra situações de risco através do planejamento de voo, onde se obtém previamente as condições climáticas referentes à origem, ao destino, à rota e ao local de pouso alternativo no caso do destino ser impraticável. O comandante L. S. Pinto afirma que “as condições de tempo no aeroporto de partida, de destino e nas alternativas, bem como as respectivas previsões, devem ser cuidadosamente analisadas” (PINTO, 2003, pg. 79).

O Comando da Aeronáutica através da Rede de Meteorologia do Comando da Aeronáutica (REDEMET) disponibiliza ampla gama de informações exclusivamente relacionadas à meteorologia associada ao planejamento e execução do voo. São disponibilizados dados atuais e previsões através de vários produtos, e alguns referem-se as condições meteorológicas dos aeroportos, ou seja, são precisamente localizadas. Destaca-se dentre estes produtos o *Meteorological Aerodrome Report* (METAR), que é a condição climática no local descrita de forma técnica, porém objetiva, disponibilizada por um CMA e que compreende os parâmetros relevantes para a correta avaliação das condições e regras de voo. Entretanto, a cobertura deste produto abrange os aeródromos federais e militares, não contemplando os mais de 1.800 aeródromos particulares (ANAC, 2013), estando inclusos neste grupo a grande maioria dos aeroclubes e escolas de aviação civil e/ou agrícola.

2.2.EXATIDÃO DA INFORMAÇÃO

Durante a formação nos cursos de piloto privado e comercial, temos presentes os voos de instrução local e de navegação. No primeiro, o aluno adquire instrução de voo em área local, geralmente próxima ao aeródromo da escola. Na navegação ocorre um voo que parte da escola e atinge um ou mais destinos, retornando à escola em seu término. Em ambos os casos, faz-se necessário obter o METAR correspondente a cada aeródromo envolvido, na maioria dos casos um aeródromo particular. Diante da falta deste produto para este tipo de local, devido a ausência de um CMA, os pilotos alunos buscam dados de aeródromos federais e/ou

militares próximos do local desejado. O procedimento é válido porém impreciso, pois de acordo com ASSIREU (2103) o relevo é fator relevante na determinação das correntes de vento. Em um avião, o deslocamento de ar que passa pela asa resulta no fenômeno da sustentação, o que permite que o avião efetue o seu voo. Este fenômeno é influenciado por diversos fatores, dentre eles o ângulo de ataque, que é a direção pela qual o deslocamento de ar incide na asa do avião. SAINTIVE (2010) explica que a variação do ângulo de ataque além do especificado pelo fabricante pode ocasionar o "estol", que é a perda da sustentação e conseqüente queda do avião. Esta variação pode ocorrer devido a rajadas de vento moderadas e com grande variação de direção, demonstrando a importância da informação meteorológica, incluindo dados do vento, para os pilotos avaliarem as condições de voo.

Entretanto, é evidente a carência de informações meteorológicas referentes aos aeródromos privados. Apesar de não ser um item previsto pelo Regulamento Brasileiro de Homologação Aeronáutica (RBHA) nº141 para ser considerado e implementado pelas escolas de aviação, de acordo com o DECEA (2013) o METAR é um produto imprescindível para a realização ou não de um voo.

Devido a esta deficiência, alguns aeroclubes adotam a exibição de METAR dos aeródromos próximos em seus *websites*, como por exemplo a Escola de Aviação de Bacacheri¹, proporcionando tal informação de forma rápida e prática. Porém, permanece imprecisa a informação, pois os dados ainda referem-se a aeroportos próximos, fornecendo um panorama da situação climática. A existência de uma informação localizada, contendo parâmetros básicos como temperatura, umidade, dados do vento e cobertura de nuvens iria contribuir para a segurança e a sua respectiva padronização na aviação regional brasileira, principalmente no tangente a formação de pilotos. O DECEA relata que o tempo de manobra das aeronaves, durante as fases de decolagem e pouso, visando corrigir a atuação de uma cortante de vento² é curto, tornando o efeito deste tipo de fenômeno "particularmente perigoso em baixos níveis" (MATSCHINSKE, 2008 págs. 1 e 2).

2.3. ESTAÇÃO METEOROLÓGICA

Para solucionar este problema, é necessário que os aeródromos possuam algum tipo de equipamento que mensure os dados meteorológicos e disponibilize-os para o grupo de alunos e usuários da escola de aviação e/ou aeroclube. Alguns locais possuem uma estação meteorológica de pequeno porte, que efetua várias medições inclusive da direção e velocidade do vento, mas que não exibe estas informações no formato METAR e também não transmite elas para visualização remota.

O objetivo deste projeto é viabilizar, com baixo custo, a coleta de informações climáticas, disponibilizando-as no formato METAR para os usuários de pequenos aeródromos desprovidos destas informações. Desta forma a carência de informações evidenciada anteriormente pode ser reduzida, melhorando a segurança de voo e auxiliando no ensino da leitura de formatos de dado padrão nas escolas de aviação.

Existem produtos semelhantes ao proposto por este projeto, como o USB *Weather Board* SEN-10586 da SPARKFUN (2013), no entanto, não foram encontradas soluções que

¹ Website da escola disponível em: <http://www.aerocon.com.br>. Acesso em 08 set. 2013.

² Variação na direção e/ou na velocidade do vento.

forneçam as informações meteorológicas no formato METAR, sendo esta uma das características que diferencia a solução construída das demais existentes no mercado.

3.OBJETIVOS

Os objetivos deste trabalho são divididos em objetivo geral e objetivos específicos.

3.1.OBJETIVO GERAL

O objetivo principal deste projeto é desenvolver um protótipo composto por *hardware* e *software* para a coleta de dados meteorológicos disponibilizando-os localmente, através de um painel interligado ao dispositivo e de forma remota através da *internet*. Pretende-se com esse dispositivo, atender a demanda pela informação meteorológica de acordo com o padrão aeronáutico, através de um módulo com baixo custo e autônomo no seu funcionamento, propiciando um aprendizado adequado para as escolas de aviação e também disponibilizando informações localizadas para os usuários de aeroclubes beneficiados com a solução.

3.2.OBJETIVOS ESPECÍFICOS

Os itens a seguir compreendem os objetivos específicos:

- a) construção de dispositivo de coleta de dados climáticos, obtidos através da leitura de sensores para temperatura, umidade relativa do ar, pressão atmosférica, luminosidade e velocidade do vento;
- b) produção da estimativa de dados que serão deduzidos com base na leitura dos sensores: cobertura de nuvens e variação da velocidade do vento;
- c) implementação da transformação dos dados meteorológicos, coletados e deduzidos, para o formato METAR;
- d) disponibilização dos dados obtidos localmente no padrão METAR e na *internet* através do WUNDERGROUND.

4. ANÁLISE DE TECNOLOGIAS/FERRAMENTAS

O dispositivo desenvolvido contempla a utilização das tecnologias e ferramentas enumeradas a seguir.

4.1. ARDUINO

O Arduino é uma plataforma de prototipação eletrônica *Open Source*, baseada em hardware e software flexível e de simples utilização. Conforme o *website* oficial do ARDUINO (2013), ele é baseado no uso de um microcontrolador o qual é programado em linguagem própria baseada no *framework* Wiring³, com sintaxe semelhante a linguagem C e possuindo um *Integrated Development Environment* (IDE) nativo para o desenvolvimento. A figura 1 ilustra um dos modelos de prototipação disponibilizados por esta plataforma.

Figura 1 - arduino Uno R2.



Fonte: http://arduino.cc/en/uploads/Main/ArduinoUno_r2_front.jpg. Acesso em: 08 set. 2013.

Baseado em microcontroladores da Atmel Corporation, o Arduino Uno é um modelo muito popular e de baixo custo. É composto por um microcontrolador ATmega328, um conversor *Universal Serial Bus* (USB) para serial ATmega16U2 e outros componentes. Dotado de 14 portas de entrada e saída digitais, 6 portas de entradas analógicas, capacidade de comunicação serial direta ou através da porta USB, 32 KB de memória *flash* utilizada para o armazenamento do programa, 2 KB de memória do tipo *Static Random Access Memory* (SRAM) utilizada pela execução do programa, 1 KB de memória do tipo *Electrically Erasable Programmable Read-Only Memory* (EEPROM) utilizada para armazenamento permanente de dados dinâmicos e *clock* de operação de 16 Mhz.

O Arduino suporta a criação de bibliotecas através de uma sintaxe que permite declarar métodos e atributos, privados ou públicos, facilitando a modularização de funcionalidades e sendo amplamente utilizada na expansão e na padronização do suporte da plataforma para diversos componentes e sensores disponíveis no mercado. Métodos e atributos privados são aqueles utilizados somente pela biblioteca enquanto os métodos e

³ *Framework Open Source* de desenvolvimento para microcontroladores. *Website*: <http://wiring.org.co>

atributos públicos são acessíveis por quaisquer código que utilize a biblioteca. Essa estrutura assemelha-se a programação orientada a objetos.

O Arduino possui uma quantidade suficiente de entradas e saídas, tanto digitais quanto analógicas, para efetuar a comunicação com os todos os sensores, botões e display utilizados neste projeto. Seu poder computacional e memória disponível também atendem as necessidades deste projeto.

O Arduino Uno foi escolhido por possuir um hardware compatível com as necessidades deste projeto, disponibilizando, em número suficiente, portas analógicas e digitais necessárias para a comunicação com os sensores. O baixo custo deste componente e a sua disponibilidade no mercado brasileiro, sendo disponibilizada uma IDE nativa e gratuita, também foram pontos importantes na escolha deste componente.

4.2.INTERFACE DE PROGRAMA DE APLICATIVO WUNDERGROUND

O WUNDERGROUND é uma empresa que disponibiliza um serviço de coleta e divulgação de dados de estações meteorológicas privadas através da Internet, sendo a primeira no gênero. Fazendo parte das empresas do grupo *Weather Channel*⁴, os serviços prestados pela empresa evoluíram ao ponto de fornecer previsões com base nos dados coletados por estações heterogêneas espalhadas pelo mundo.

As estações meteorológicas privadas, chamadas *Personal Weather Station* (PWS), geralmente são equipamentos utilizados em casas, fazendas e outras habitações sem a necessidade de precisão de uma estação meteorológica utilizada, por exemplo, em grandes aeroportos. O módulo desenvolvido neste projeto trata-se de um modelo de PWS customizado para atender as necessidades do âmbito aeronáutico que foram evidenciadas.

Os dados do módulo de coleta são enviados para o WUNDERGROUND através de uma *Application Program Interface* (API), utilizando um protocolo específico para as PWS, que é baseada no método GET do *Hypertext Transfer Protocol* (HTTP). Todas as APIs são disponibilizadas para uso de forma gratuita, visto a intenção colaborativa do WUNDERGROUND.

Os fatores determinantes na escolha deste serviço para a publicação dos dados da estação meteorológica na internet são a gratuidade do serviço para estações pessoais (PWS), a boa documentação da API e a simplicidade da API que permite sua utilização em conjunto com o Arduino.

4.3.SENSOR DE TEMPERATURA E PRESSÃO ATMOSFÉRICA

Fabricado pela Bosch Sensortec, o sensor BMP085 fornece a pressão atmosférica e a temperatura do ar. Conforme o *datasheet* da BOSCH (2008), este sensor possui uma margem de erro de 1 hPa para mais ou para menos em relação a pressão atmosférica obtida e de 0,5 graus Celsius para mais ou para menos em relação a temperatura do ar mensurada. A sigla hPa significa hectopascal e é uma unidade de medida da pressão atmosférica do ar.

⁴ Rede de televisão dedicada a meteorologia, hoje disponível em diversos meios inclusive na Internet através do endereço <http://www.weather.com>

Exibido na figura 2, este sensor trabalha com o protocolo de comunicação I2C⁵, que permite a transferência de dados utilizando-se somente dois fios.

Este sensor foi adotado por ser fabricado por uma empresa conhecida no mercado, mas principalmente pelo fato de utilizar o protocolo I2C, cujas vantagens são abordadas posteriormente.

Figura 2 - sensor Bosch BMP085.



Fonte: Bosch Sensortec.

4.4.SENSOR DE UMIDADE RELATIVA DO AR

O sensor AM2302 fornece a umidade relativa do ar e a temperatura do ar, sendo fabricado pela Aosong Electronics Company. AOSONG ELECTRONICS (2012) relata que a margem de erro obtida na leitura da umidade relativa é de 2% para mais ou para menos e na leitura da temperatura do ar é de 0,5 graus Celsius para mais ou para menos.

O sensor, exibido na figura 3, utiliza um protocolo próprio de comunicação que utiliza apenas um fio, porém através deste fio pode-se efetuar a comunicação somente com um sensor, ao contrário do barramento I2C que suporta vários dispositivos limitados somente pela distância em que se encontram. Devido a esta característica, bem como pelo fato deste padrão de comunicação ser mais lento em relação ao I2C, optou-se por obter a temperatura do ar do sensor BMP085.

Este sensor foi adotado por ser o único encontrado no mercado brasileiro que opera com o padrão I2C, tornando o AM2302 uma das únicas opções de baixo custo viáveis para este projeto.

Figura 3 - sensor de temperatura e umidade relativa do ar AM2302



Fonte: Aosong Electronics.

⁵ Inter-integrated circuit: barramento inventado pela Philips para comunicação entre dispositivos.

4.5.SENSOR DE LUMINOSIDADE

Sensor modelo BH1750FVI fabricado pela ROHM Company, exibido na figura 4, utiliza o padrão I2C de comunicação e fornece a luminosidade em Lux (ROHM COMPANY, 2011). É através dele que é computada a cobertura de nuvens, através do relação entre a luminosidade obtida, a hora local e os horários de nascer e por do sol para este local, gerados a partir da posição geográfica configurada pelo usuário.

Este sensor foi adotado por trabalhar com o padrão I2C e também por fornecer a luminosidade em Lux. Outros sensores, de custo muito inferior em relação a este, fornecem a luminosidade de forma analógica, que é muito mais difícil de ser tratada pois seria necessário implementar uma interrupção de software adicional dedicada a leitura deste tipo de sinal, assim como o tratamento de sua amplitude. O fornecimento da luminosidade em Lux também será importante para a análise e implementação do tratamento da cobertura de nuvens, a qual utiliza esta unidade de medida.

Figura 4 - rohm BH1750FVI.



Fonte: ROHM Company.

4.6.SENSOR DE VELOCIDADE DO VENTO

A figura 5 ilustra o sensor que mede a velocidade do vento ou anemômetro. Trata-se de um conjunto rotativo de conchas que mensura a velocidade do vento que passa por elas, independentemente da direção (DAHLBERG, 2006). Sua rotação é obtida através de um sensor de efeito *hall*, o qual transforma o movimento em pulsos elétricos analisados pelo algoritmo implementado e executado no Arduino. O sensor de efeito *hall* é composto por uma chave de contato elétrico, que é acionada mediante a presença de uma fonte magnética que atrai os contatos desta chave, que encontram-se naturalmente separados, ocasionando a passagem da corrente elétrica.

O anemômetro adquirido foi o único encontrado no mercado brasileiro que possuía um custo viável para a proposta do projeto. Diversos orçamentos foram efetuados e os valores de anemômetros profissionais são impraticáveis para o propósito deste projeto (acima de R\$ 10.000,00). Soluções alternativas tais como a construção de um anemômetro baseado em ultra-som foram consideradas para o projeto, e trariam diversos benefícios mas devido ao tempo necessário para a sua implementação foram descartadas para este projeto, mas são alternativas interessantes para implementações futuras.

Figura 5 - sensor de velocidade do vento.



Fonte: Gerais Sinalizadores.

4.7.COMPONENTE GSM EFCOM PRO V1.0

Fabricado pela Elec Freaks e contendo o circuito integrado SIM900 fabricado pela SimCom, trata-se de um dispositivo para comunicação utilizando telefonia móvel no padrão *Global System for Mobile Communications* (GSM). Este dispositivo possui uma interface serial e sua utilização é efetuada através de comandos AT (SIMCOM WIRELESS SOLUTIONS LTD, 2009). A figura 6 demonstra o componente, montado com a antena para recepção e transmissão GSM.

O módulo proposto precisava de uma forma de comunicação, sendo escolhido o padrão GSM por ser totalmente independente da infraestrutura existente nos locais de instalação, garantindo assim o isolamento do dispositivo em relação a outros equipamentos. Adicionalmente, com o uso desta tecnologia é possível efetuar a instalação do módulo remotamente, em local próximo a pista de pouso e decolagem, por exemplo.

Figura 6 - módulo de comunicação GSM.



Fonte: Elec Freaks.

4.8.CONVERSOR I2C

O módulo LCD1602 é um adaptador que permite a conexão de um *display* LCD padrão Hitachi em um barramento de comunicação I2C (SCHENZHEN EONE ELECTRONICS, 2013). O padrão Hitachi é amplamente utilizado em *displays* LCD,

inclusive pelo display que foi selecionado para este projeto. O módulo I2C ilustrado na figura 7, permite economia significativa no uso de portas digitais no Arduino, propiciando uma expansão de suas funcionalidades se necessário.

Este componente foi adquirido após a definição pelo uso do padrão I2C no projeto, e como o *display* LCD já havia sido adquirido, foi necessária a aquisição deste componente para otimizar o uso das portas digitais do Arduino.

Figura 7 - módulo de comunicação I2C para LCD.



Fonte: UC Tronics.

4.9.RELÓGIO DE TEMPO REAL

O módulo DS1307 é um componente que permite armazenar e obter a data e a hora através do protocolo I2C, mesmo que o dispositivo esteja desligado (MAXIM INTEGRATED, 2008). Exibido pela figura 8, este componente possui uma bateria que atua enquanto o dispositivo estiver desligado, mantendo a contagem da data e hora durante este período. O formato METAR exige a exibição da data e hora sendo este componente imprescindível para o projeto. O Arduino possui um relógio interno, porém ele não armazena a informação ao ser desligado, o que forçaria o usuário a configurar a data e hora em toda ocasião de reinício do sistema, por exemplo, no desligamento por queda de energia.

Figura 8 - relógio DS1307.



Fonte: Comunica Ltd.

4.10.PROTOCOLO I2C

Durante a etapa de pesquisa, definição e estudo dos componentes e sensores que foram utilizados neste projeto, foi constatado que o protocolo I2C é suportado pela maioria dos

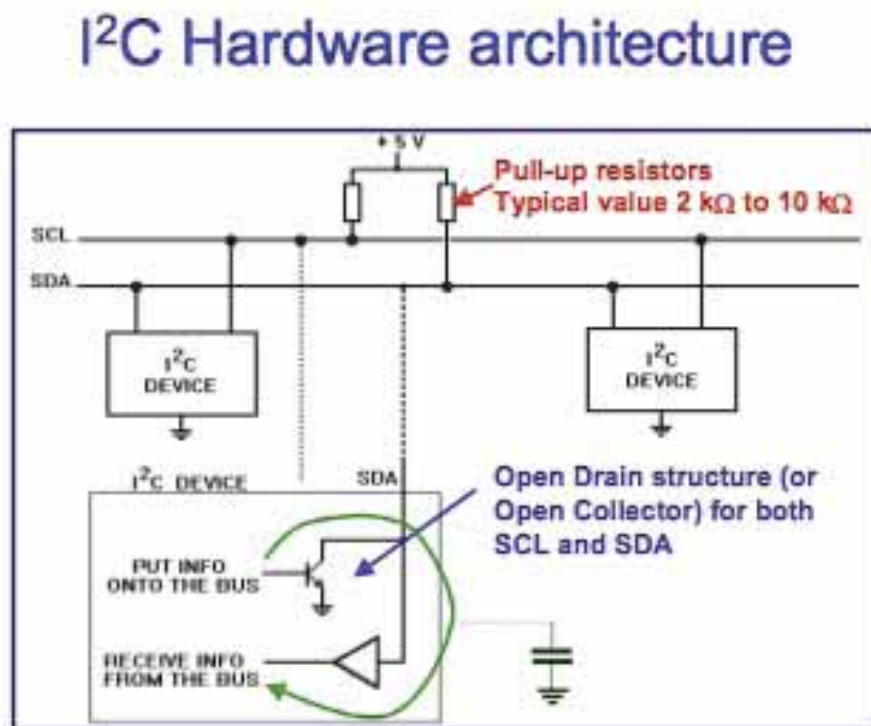
elementos selecionados. Pesquisando o protocolo, ficou claro que sua adoção traria benefícios ao projeto.

O protocolo I2C permite a comunicação com vários componentes através de um único barramento de três fios (NXP SEMICONDUCTORS, 2012). Originalmente elaborado pela Philips, este barramento viabiliza uma comunicação serial bi-direcional, de 8 bits, que suporta desde 100 Kbps no modo padrão até 3,4 Mbps no modo de alta velocidade, possuindo limitação apenas na capacitância total do barramento, que é de 400 pF. Ou seja, muitos componentes e/ou um barramento com uma quantidade elevada de fios pode comprometer a comunicação. No módulo desenvolvido neste projeto, a quantidade de componentes, condutores e fios é reduzida e não impacta nesta limitação.

O uso do protocolo I2C reduziu a quantidade de portas digitais necessárias. Sem o uso deste protocolo, seriam necessárias uma ou mais portas digitais para cada sensor. Com a adoção do I2C foram utilizadas apenas duas portas para este barramento, permitindo a comunicação com quase a totalidade dos componentes deste projeto.

A figura 9 ilustra o funcionamento básico do barramento I2C, onde existem duas linhas de comunicação e um terceiro fio que é o aterramento do circuito. Uma das linhas é a *Serial Clock* (SCL) e sua função é estabelecer a velocidade do barramento. A outra linha é a *Serial Data* (SDA) pela qual os dados são transmitidos. Basicamente, o protocolo I2C se baseia no conceito de requisição e resposta, onde um dispositivo do barramento solicita que outro retorne dados, sendo que este processo de requisição e transmissão ocorre de forma síncrona na linha SDA controlada pela linha SCL.

Figura 9 - arquitetura I2C.



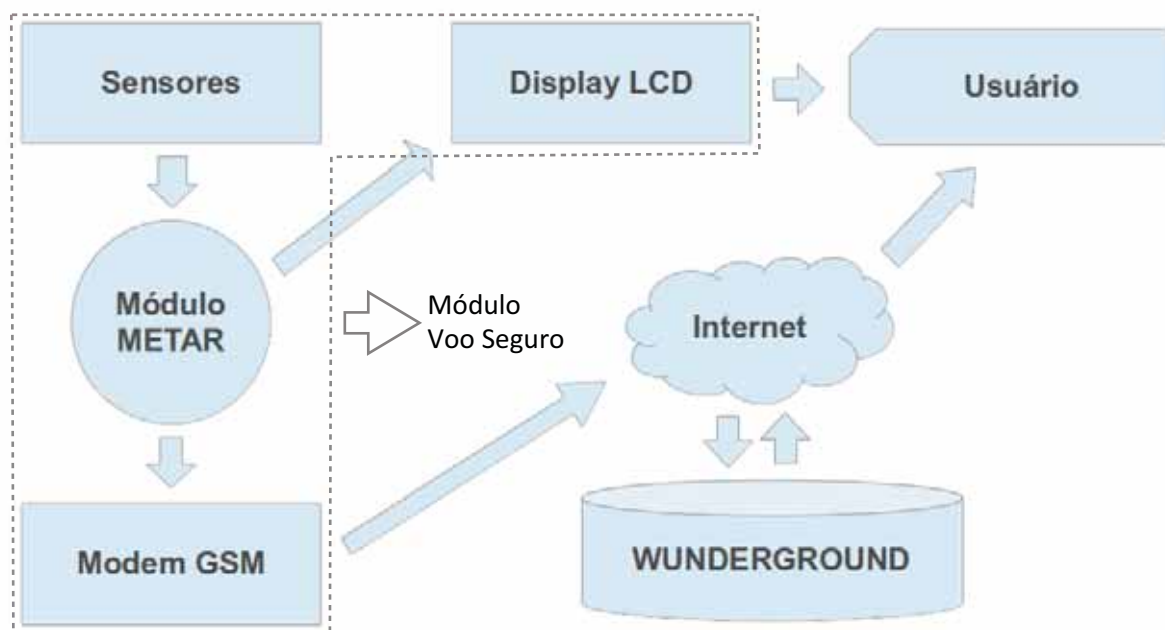
5.DESCRICÃO DA SOLUÇÃO

A solução proposta por este projeto compreende a coleta de informações meteorológicas e a disponibilização destas em um painel no equipamento e remotamente através da *internet*, utilizando o site WUNDERGROUND.

No painel do equipamento, existe um display LCD com 4 linhas de 20 caracteres cada, no qual são exibidas, no formato METAR, as seguintes informações: código ICAO do aeródromo, data e hora no fuso horário de referência *Universal Time Coordinated* (UTC), cobertura de nuvens, velocidade do vento, temperatura do ar, temperatura do ponto de orvalho e pressão atmosférica do ar. Junto ao *display*, existem dois botões para configuração e operação do módulo, nomeados de "Config" e "Ação".

Após configurado, o módulo é capaz de funcionar de forma autônoma, propiciando as informações coletadas ininterruptamente, enquanto estiver sendo energizado através da fonte de alimentação. Ainda que ocorra a falta de energia elétrica, por exemplo, o módulo recupera os dados configurados e retoma suas atividades automaticamente. Serão gerados alertas quando for necessária a intervenção do usuário, porém as informações disponíveis e que não possuam estado de erro reportado pelos sensores continuarão a ser exibidas. A figura 10 ilustra o funcionamento da solução.

Figura 10 - funcionamento da solução



Fonte: Próprio autor.

O dispositivo Voo Seguro é composto pelo módulo de coleta (sensores) e processamento (módulo METAR) de informações do meio ambiente, modem GSM para a comunicação e um *display* LCD para interface com o usuário. O módulo de coleta é implementado utilizando um conjunto de sensores e o Arduino, contendo a implementação de um *firmware*, que é o elemento programável que interage com todos os demais componentes. O módulo de coleta obtém os dados dos sensores, efetua o processamento destes dados e exibe o resultado no *display* LCD, efetuando a transmissão deles através do modem GSM para o site WUNDERGROUND. O usuário pode efetuar a consulta destes dados localmente

através do *display* LCD ou remotamente através da internet, consultando o site WUNDERGROUND.

5.1.MÓDULO DE COLETA E PROCESSAMENTO

O módulo de coleta e processamento é o principal componente do projeto, responsável por coletar, processar e disponibilizar as informações meteorológicas conforme proposto, permitindo ao usuário configurar os parâmetros necessários. É composto por um Arduino Uno R2, sensores, componentes, display de LCD, dois botões de comando e fonte de alimentação. Contém toda a lógica necessária, tratando-se efetivamente do elemento programável.

Os sensores irão fornecer informações meteorológicas que serão coletadas e processadas pelo Arduino. Já os componentes são elementos com funcionalidades específicas e operam de forma independente do Arduino. O relógio RTC armazena a data e a hora de forma ininterrupta e o componente GSM permite a comunicação utilizando a rede de telefonia móvel.

O *firmware* existente no Arduino é dividido em duas partes, que podemos chamar de inicialização e processamento. Na inicialização, todos os sensores e componentes são inicializados, além de serem configuradas todas as variáveis de controle do módulo e também as que armazenam as informações climáticas. No processamento, o módulo permanece num laço permanente e infinito, efetuando tarefas que são escalonadas por tempo e que são interrompidas somente durante a configuração do módulo e durante a transmissão dos dados. A leitura do anemômetro funciona de forma diferenciada, possuindo uma interrupção de hardware que desvia o fluxo de processamento sempre que uma volta deste sensor for percebida pelo Arduino, viabilizando o tratamento deste sinal.

Sua programação é efetuada através de uma *Integrated Development Environment* (IDE) própria e após programado pode ser operado normalmente, desligado e ligado novamente sem necessidade de nova programação, visto que a mesma fica retida no microcontrolador. Esta implementação trata-se de um *firmware*, utilizado em sistemas embarcados e que funciona especificamente no hardware para o qual foi codificado, admitindo atualizações conforme necessário.

A configuração é efetuada através de botões de comando, que permitem ao usuário selecionar o item que deseja configurar e alterar o item selecionado. Devem ser configurados o código ICAO da localidade, a data e hora UTC e a latitude e longitude. Estes dados são armazenados na memória interna do Arduino, sendo recuperados automaticamente em caso de falta de energia, tornando o módulo totalmente autônomo após sua configuração inicial.

5.2.O FORMATO METAR

Este projeto tem como principal diferencial a transformação de dados meteorológicos para o formato METAR. No mercado, encontram-se com facilidade módulos para obtenção e transmissão de dados climáticos obtidos através de sensores. O USB *Weather Board* SEN-10586 da SPARKFUN (2013) conta inclusive com alguns dos sensores utilizados neste projeto incorporados na placa do módulo, porém não efetua nenhuma transformação dos dados para o formato METAR e não possibilita esta funcionalidade nativamente, necessitando de um hardware adicional, como por exemplo um Arduino.

SONNEMAKER (2009) descreve o METAR como uma informação meteorológica obtida de forma rotineira e periódica, conforme os padrões internacionais da Organização Meteorológica Mundial (OMM). Esta informação é específica para fins aeronáuticos, visto que as informações nela contida são orientadas para que o piloto possa mensurar a situação climática de superfície, avaliando as condições de decolagem e/ou pouso em determinado aeródromo. Conforme detalhado anteriormente, esta informação é fornecida oficialmente no Brasil pela REDEMET para cada aeródromo que contenha um CMA.

Este formato pode ser dividido nos elementos local, dia e hora no fuso horário UTC, dados do vento, temperatura do ar e do ponto de orvalho, visibilidade, alcance visual, tempo presente da pista, nebulosidade e pressão ao nível do mar para ajuste de altímetro. Tais elementos são construídos utilizando-se códigos textuais, compostos de letras e números, que irão representar as informações climáticas.

Baseado em entrevista com o coordenador dos instrutores do Aeroclube de Eldorado do Sul, as informações a seguir são relevantes para a análise das condições meteorológicas para o voo, as quais são exibidas pelo módulo construído:

- a) local: É composto por quatro caracteres alfabéticos que formam o identificador de localidade no padrão ICAO, evidenciando onde as informações foram obtidas;
- b) dia e hora: Formado pelo dia, hora e minutos todos representados por dois algarismos decimais, preenchidos com zero a esquerda quando necessário. Deve ser computada a hora UTC e para isto o módulo permite a sua parametrização em tela de configuração;
- c) dados do vento: É composto pela velocidade do vento em nós, sendo dois algarismos decimais finalizados pelo código KT, que é a abreviatura para a velocidade do vento em nós. O vento informado é a média dos dez minutos precedentes a observação. Sempre que a velocidade do vento for menor que um nó a informação resultante para os dados do vento é o texto 00000KT. O vento de rajada é observado durante 10 minutos antecedentes ao da obtenção do METAR, sendo que caso haja variação maior ou igual a dez nós na velocidade obtida esta variação é informada adicionando-se a letra G logo após a velocidade do vento e antes do código KT, sucedida pela velocidade de rajada de vento mensurada neste intervalo;
- d) temperatura do ar e do ponto de orvalho: São apresentados em graus Celsius inteiros, sem arredondamentos. Apresenta-se primeiro a temperatura do ar e após a temperatura do ponto de orvalho, ambas separadas por uma barra. Caso alguma das temperaturas seja negativa, a mesma é precedida pela letra M. A temperatura do ponto de orvalho é aquela em que o ar, sob pressão e umidade constantes, se condensa (SONNEMAKER, 2009). Em termos práticos, esta informação é importante pois pode-se deduzir a altitude mínima para a presença de nuvens a partir dela, permitindo ao piloto deduzir o teto de voo para os aeródromos local e remotos;
- e) nebulosidade: O formato METAR prevê quatro tipos de nebulosidade: poucas nuvens, esparso, nublado e encoberto, os quais são representados pelos códigos FEW, SCT, BKN e OVC, respectivamente. Estes tipos de nebulosidade estão associados a cobertura de nuvens existente no céu. Adicionalmente, o formato prevê a informação da altura das camadas de nuvens, partindo da mais baixa até a mais alta e informado em centenas de pés até a altitude de 10.000 pés. Usualmente esta

informação é obtida para várias camadas de nuvens através de um balão meteorológico, sendo que neste projeto é informada apenas a cobertura de nuvens em uma camada, sem informar a altitude. Este projeto propõe exibir esta informação estimada com base na luz do sol, sendo que durante os períodos noturnos ou quando não for possível estimar a presença de nuvens, tal informação não será exibida;

- f) pressão ao nível do mar para ajuste de altímetro: É informada em valores inteiros com quatro algarismos e precedidos pela letra Q, preenchidos com zero a esquerda quando necessário.

Como exemplo de uma observação no formato METAR, obtida para o Aeroporto Internacional Salgado Filho em Porto Alegre (cujo código ICAO é SBPA), as 12:00 UTC do dia 19, no qual foi observado um vento de 20 nós e com rajadas de 32 nós, sendo mensurada uma pressão de 1003 hPa, uma temperatura de 17 graus Celsius com ponto de orvalho em 15 graus Celsius e cobertura de nuvens equivalente a 5/8 do céu, teremos um METAR da seguinte forma: METAR SBPA 191200 20G32KT 17/15 BKN Q1003. Neste exemplo, encontramos as observações citadas anteriormente:

- a) SBPA: sigla ICAO do Aeroporto Internacional Salgado Filho;
- b) 191200: observação efetuada no dia 19 às 12:00 UTC;
- c) 20G32KT: vento de 20 nós com rajadas de 32 nós;
- d) 17/15: temperatura do ar de 17 graus Celsius e temperatura do ponto de orvalho de 15 graus Celsius;
- e) BKN: presença de nuvens configurando um céu nublado, ou seja, 5/8 do céu coberto por nuvens;
- f) Q1003: pressão atmosférica do ar de 1003hPa.

Adicionalmente, para melhorar a aderência ao padrão, o METAR gerado contém no final o texto RMK A01, indicando que a estação que gerou a informação é automática e não possui sensor de precipitação. Também é inserido imediatamente após o código ICAO da localidade a sigla AUTO, que indica que a estação não possui um observador e funciona de forma autônoma.

O METAR deve ser emitido a cada 60 minutos, conforme a norma (WORLD METEOROLOGICAL ORGANIZATION, 1995). Entretanto, para este projeto foi adotado o tempo de 10 minutos, visando manter os dados atualizados com maior frequência, tendo em vista que serão divulgados localmente. Desta forma colabora-se com a segurança de voo, através de dados atualizados com no máximo 10 minutos de defasagem. Nas estações integrantes da REDEMET, um METAR especial é emitido quando condições significativas de clima são percebidas, tais como rajadas de vento forte ou temporais. O tempo utilizado no projeto de 10 minutos visa atender este tipo de reporte especial, sendo que a implementação de um METAR especial pode ser efetuada como melhoria deste projeto.

5.3.ADEQUAÇÃO DOS DADOS

Os sensores são os componentes que irão transformar a situação climática em dados que serão interpretados pelo módulo. Alguns dados podem ser consumidos diretamente, sem a necessidade de interpretação adicional, enquanto outros precisam ser interpretados e relacionados com outros parâmetros.

A rajada de vento deve ser estimada com base em um histórico de leituras. Quando for percebida uma alteração conforme a especificação do formato METAR, esta deve ser considerada como rajada. O módulo trabalha com este conceito de histórico e efetua a comparação das leituras e indicação de rajada conforme o formato METAR.

A temperatura do ponto de orvalho é estimada com base na temperatura do ar e na umidade relativa do ar. A estimativa é realizada utilizando-se uma fórmula matemática, explicada posteriormente neste documento.

A cobertura de nuvens é estimada com base na luminosidade do sol que é percebida pelo sensor de luminosidade. De acordo com o tempo no qual o sensor percebe uma obstrução é deduzida a cobertura de nuvens. Para isto, o módulo deduz a luminosidade estimada para a posição do módulo com base em cálculos matemáticas parametrizados com a data e hora atuais, latitude e longitude do módulo, de onde se estima a luminosidade esperada de acordo com a posição do sol, informação esta que será comparada com o valor medido pelo sensor.

5.4.WUNDERGROUND API

Os dados coletados pelo módulo são enviados ao site WUNDERGROUND através de API específica para as PWS. Para cada estação baseada no módulo atual, é necessária a criação de uma estação correspondente no site WUNDERGROUND, resultando em um identificador da estação e uma senha associados. A implementação do módulo conta com estes dados de acesso configurados no código, para simplificar a interface com o usuário, uma vez que a interação com o dispositivo é realizada utilizando-se apenas duas teclas.

Para o envio, é efetuada uma requisição HTTP com método GET direcionado para a URL de atualização para estações PWS, utilizando os dados de identificação citados anteriormente, um parâmetro *action* com a *string* *updateraw* e também os dados meteorológicos, sendo um parâmetro para cada dado. O WUNDERGROUND suporta vários parâmetros climáticos, porém para o propósito deste projeto são enviados os seguintes parâmetros:

- a) *dateutc*: data e hora no fuso horário UTC no seguinte formato: YYYY-MM-DD HH:MM:SS;
- b) *windspeedmph*: velocidade instantânea do vento em milhas por hora;
- c) *windgustmph*: velocidade da rajada do vento em milhas por hora;
- d) *humidity*: umidade relativa do ar em percentual;
- e) *dewptf*: temperatura do ponto de orvalho em Fahrenheit;
- f) *tempf*: temperatura ambiente externa em Fahrenheit;
- g) *baromin*: pressão atmosférica em polegadas de mercúrio;
- h) *clouds*: cobertura de nuvens em formato METAR sucinto, da seguinte forma: OVC para céu encoberto, BKN para céu nublado, SCT para céu parcialmente nublado, FEW para céu com poucas nuvens e SKC para céu livre de nuvens.

O WUNDERGROUND especifica que não é necessário o envio de todos os parâmetros, justamente por lidar com estações de tipos variados. Para este projeto são enviados em todas as conexões todos os parâmetros listados acima, exceto o item relativo a cobertura de nuvens durante o período noturno somente, tendo em vista que a forma pela qual será deduzida esta informação baseia-se na luz diurna.

Após o envio dos dados, é obtido um retorno que pode ser uma das duas possibilidades a seguir:

- a) "*success*": para os casos em que o envio ocorreu normalmente;
- b) qualquer mensagem diferente de "*success*" indica uma condição problemática que não será tratada pelo módulo, podendo ser desde a senha informada de forma incorreta até a falta de disponibilidade dos servidores. Neste cenário, devido a ampla gama de possibilidades, o módulo é incapaz de determinar precisamente o problema, pois o retorno da API é textual e semântico, não sendo especificado pelo WUNDERGROUND.

Conforme a especificação da API, os parâmetros mandatórios para todo e qualquer envio de dados são: identificador, senha, *action* e *dateutc*. A API também especifica o uso de atualizações RapidFire, no qual os dados são atualizados a cada 2,5 segundos. Não sendo o intuito deste projeto, este tipo de envio não será abordado.

Periodicamente a cada 30 minutos o módulo desenvolvido inicia uma tarefa para transmitir os dados para o WUNDERGROUND. Seu funcionamento é semelhante a exibição local, utilizando-se dos dados coletados e armazenados em memória, com as adequações no formato que forem necessárias. Porém, nesta funcionalidade aplica-se o tratamento de falhas de transmissão, efetuando novas tentativas em intervalo mais curto até que seja efetuada a transmissão com sucesso. No caso de falha no envio, é notificado no *display* local o alerta de que o módulo encontra-se *offline*, para efeitos de diagnóstico.

5.5.EXIBIÇÃO LOCAL

O módulo efetua a adequação dos dados lidos pelos sensores para o formato METAR. Foi utilizado um *display* de 4 linhas com 20 caracteres cada, permitindo um máximo de 80 caracteres visíveis simultaneamente. Esta interface também conta com telas para configuração do módulo.

6.ABORDAGEM DE DESENVOLVIMENTO

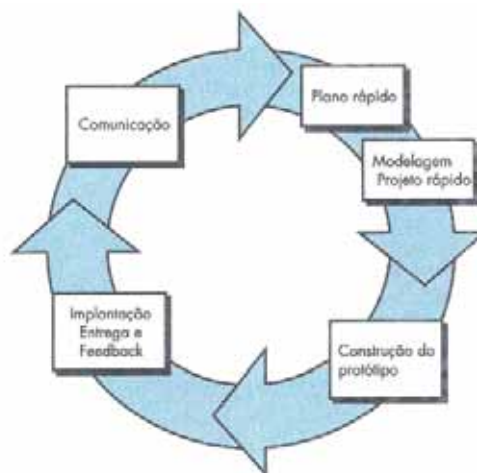
Para a construção e implementação do projeto, foi necessário considerar os seguintes itens:

- a) compreensão da meteorologia envolvida no projeto;
- b) seleção e estudo das especificações dos componentes eletrônicos necessários para a implementação do dispositivo;
- c) projeto da arquitetura detalhada do dispositivo de acordo com as especificações dos sensores e com foco no escalonamento de tarefas.

A quantidade de fatores que precisaram ser compreendidos para a definição do que foi implementado foi elevada e complexa. A definição dos sensores estabeleceu parte da implementação do módulo de coleta. CARR e VERNER (2007) relatam que uma abordagem de desenvolvimento baseada em protótipo é adequada para projetos com esta característica de indefinição inicial, no qual um protótipo é elaborado e então revisto para que sejam feitas as devidas correções, seguindo para a geração de um novo protótipo e assim sucessivamente até que o resultado final seja atingido.

O método de desenvolvimento utilizado foi baseado em prototipação com desenvolvimento incremental, onde uma parte do sistema é implementada de forma a permitir uma avaliação inicial visando coletar melhorias, porém com uma arquitetura que permita que o desenvolvimento seja continuado agregando código adicional que se encaixa com o que já foi codificado. A figura 11 ilustra o fluxo do processo de prototipação, no qual é efetuado um plano rápido do que será desenvolvido no protótipo, seguido de uma modelagem do que foi definido no plano. Após, é efetuada a construção do protótipo com posterior entrega do mesmo, obtendo o retorno do cliente ou usuário. Repete-se então este conjunto de etapas para cada protótipo a ser elaborado, até que todos os requisitos sejam atendidos. Isto permite uma construção em etapas, com entregáveis parciais funcionais, viabilizando modificações no decorrer dos ciclos, através do retorno e comunicação com o cliente ou usuário.

Figura 11 - desenvolvimento prototipado.



Fonte: <http://www.galeote.com.br/blog/wp-content/uploads/2012/06/image5.png>. Acesso em: 08 set. 2013.

Os artefatos utilizados adotam a especificação UML e compreendem diagramas de estado e diagramas de atividade, exceto no caso dos diagramas esquemáticos os quais utilizam notação padrão para circuitos eletrônicos.

6.1.CICLOS DE DESENVOLVIMENTO

A organização do desenvolvimento foi baseada em ciclos de desenvolvimento prototipados, sendo que para cada ciclo uma pequena parte do todo foi especificada e/ou implementada, viabilizando alterações de tecnologia e/ou implementação se necessário.

6.1.1.Ciclo de desenvolvimento 1

Compreendeu a etapa inicial do projeto, com atividades de pesquisa e aquisição de componentes e sensores. Também foi definida a arquitetura inicial do projeto a ser implementada. Este ciclo teve como produto uma arquitetura inicial do projeto e a definição dos sensores a serem utilizados.

6.1.2.Ciclo de desenvolvimento 2

Neste ciclo foi implementada o escalonamento de tarefas, utilizando a biblioteca Metro. O produto foi um protótipo inicial com um escalonador de tarefas, com testes efetuados através da porta serial do Arduino, tendo em vista a falta do *display* LCD nesta etapa.

6.1.3.Ciclo de desenvolvimento 3

Foi implementado a leitura dos sensores que não necessitam de tratamento dos dados, gerando como produto um protótipo capaz de ler estes sensores utilizando o escalonador do protótipo gerado no ciclo de desenvolvimento 2, utilizando a saída via porta serial para testes e depuração.

6.1.4.Ciclo de desenvolvimento 4

Foram codificadas a estimativa da temperatura de ponto de orvalho e a renderização do formato METAR, gerando um protótipo capaz de enviar via comunicação serial um METAR parcial, contendo os dados sensores de temperatura do ar, da pressão atmosférica do ar e da estimativa da temperatura de ponto de orvalho.

6.1.5.Ciclo de desenvolvimento 5

Efetuada a implementação do controlador do *display* LCD, através do padrão de comunicação I2C, viabilizando um protótipo com saída do METAR no LCD.

6.1.6.Ciclo de desenvolvimento 6

Nesta etapa foram efetuados testes do conjunto de implementações geradas pelos ciclos anteriores, assim como foram efetuadas correções e melhorias nestas implementações. Este ciclo gerou um protótipo cuja implementação foi qualificada.

6.1.7.Ciclo de desenvolvimento 7

Efetuada pesquisa, estudo e implementação da leitura dos sensores relativos a velocidade do vento, com estudo da meteorologia associada, proporcionando um protótipo com a leitura destes dados.

6.1.8.Ciclo de desenvolvimento 8

Implementação da estimativa das rajadas de vento e velocidade média do vento, conforme proposto neste projeto. O produto deste ciclo foi um protótipo capaz de fornecer os dados completos relativos ao vento.

6.1.9.Ciclo de desenvolvimento 9

Efetuada qualificação parcial do projeto, excetuando-se a transmissão dos dados. Isto gerou um protótipo confiável para prosseguimento do projeto.

6.1.10.Ciclo de desenvolvimento 10

Neste ciclo foi efetuado estudo e análise da especificação da API do WUNDERGROUND e do uso de HTTP com o Arduino. Este estudo forneceu a base para a implementação destes requisitos nos ciclos posteriores.

6.1.11.Ciclo de desenvolvimento 11

Implementado a formatação dos dados climáticos para o padrão especificado pela API do WUNDERGROUND. O produto foi um protótipo pronto para comunicar-se com o WUNDERGROUND, necessitando ainda a implementação da camada de comunicação.

6.1.12.Ciclo de desenvolvimento 12

Efetuada a codificação do componente GSM, utilizando o protocolo HTTP para comunicação com o WUNDERGROUND, gerando um protótipo capaz de comunicar-se com este site.

6.1.13.Ciclo de desenvolvimento 13

Compreendeu testes com a solução completa, seguida por correções gerando um protótipo final consolidado.

6.1.14.Ciclo de desenvolvimento 14

Neste ciclo, foi efetuada a validação da solução conforme especificado neste documento, com acertos finais de códigos e parâmetros, gerando o protótipo final deste projeto.

7.ARQUITETURA DO SISTEMA

O projeto é composto de um dispositivo de *hardware* e o *firmware* que controla as respostas do dispositivo. Para a implementação do projeto, o mesmo foi dividido em três módulos: coleta de dados, exibição local e transmissão. O hardware é composto pelo Arduino, os sensores e os componentes. Já o *software* trata da implementação de um *firmware*, cuja finalidade é transformar os dados sensoriais em informações climáticas a serem exibidas localmente e transmitidas para ambiente remoto via *internet*.

7.1.REQUISITOS

Para cumprir os objetivos deste projeto, foram evidenciados três requisitos primários que são relacionados a seguir.

7.1.1.Captura das condições climáticas

Os dados meteorológicos necessários para a geração do METAR proposto devem ser coletados pelo módulo. Para cumprir este requisito, foi necessário implementar a comunicação com os sensores correspondentes. A temperatura do ar, a temperatura do ponto de orvalho, a velocidade do vento, a velocidade da rajada de vento, a cobertura de nuvens e a pressão atmosférica do ar tiveram ser lidas dos sensores, e os valores retornados foram processados pelo módulo. Esta tarefa foi concluída com o auxílio de bibliotecas padronizadas de apoio, tais como a biblioteca *Wire*, que viabiliza a interação com componentes através do padrão I2C. Este padrão foi estudado e avaliado antes de ser determinado como vantagem para o cumprimento do requisito, sendo que logo nos ciclos iniciais foi estabelecida a sua prioridade quanto a aquisição dos sensores a serem implementados, devido a otimização de conexões eletrônicas proporcionadas por este padrão. Outra biblioteca utilizada foi a *BMP085DRIVER*, que apesar de não ser fornecida pelo fabricante do sensor foi implementada pela comunidade *open source* do Arduino e disponibilizada através de repositório de códigos para utilização e colaboração. O uso deste tipo de biblioteca permitiu implementar os sensores com relativa facilidade, viabilizando a dedicação a implementação do processamento dos dados coletados.

Após a coleta, alguns dados precisaram ser transformados para que a respectiva informação meteorológica fosse gerada. Os dados dos sensores de velocidade do vento, de luminosidade e da umidade relativa do ar precisaram desta transformação, cujo detalhamento é abordado posteriormente neste documento.

Outra informação necessária para a produção do METAR é a data e hora UTC, que deve ser configurada pelo usuário antes de ser armazenada e consumida. Outras configurações, necessárias, são por exemplo o código ICAO do aeródromo e a posição GPS do mesmo.

7.1.2.Processamento dos dados coletados

Os dados climáticos são coletados dos sensores periodicamente, através de tarefas atribuídas com o uso da biblioteca *Metro*. Após coletados, a velocidade do vento e a luminosidade necessitam de tratamento para que sejam transformadas em informações

climáticas adicionais. A variação da velocidade do vento após processada resulta na velocidade da rajada de vento, que é o vento máximo obtido em determinado intervalo de tempo, ainda que seja momentâneo. A variação da luminosidade, associada aos horários de nascer e pôr do sol, resulta na estimativa de cobertura de nuvens. Para cada caso, um algoritmo particular foi elaborado, visando gerar a informação necessária.

7.1.3.Exibição e transmissão dos dados processados

Com os dados coletados e processados, foi necessário uma rotina de conversão destes em um vetor de caracteres contendo os mesmos no formato METAR. Esta rotina basicamente converte os tipos de dados e monta uma linha de texto concatenando-os de acordo com o formato METAR. Também foi processada a exibição condicional de informações meteorológicas, como por exemplo as rajadas de vento, que quando menores que determinado valor devem ter sua exibição em formato diferenciado.

Após coletados os dados e gerada a informação METAR, este produto deve ser transmitido através da Internet para o servidor remoto, viabilizando a consulta da informação por um usuário distante, munido apenas de uma conexão com a rede mundial de computadores. Para esta conexão foi utilizado um módulo GSM para o Arduino, cuja interface é serial.

A International Telecommunication Union (ITU) é uma agência especializada em tecnologias para informação e comunicação. O padrão de comandos utilizados para comunicação serial, também conhecido como conjunto de comandos Hayes, foi recomendado por esta agência (ITU, 2003) e são adotados pelo componente GSM adquirido. Tais comandos foram implementados no Arduino, assim como um fluxo de verificação de conectividade que detecta se o componente de comunicação não possuir registro na rede GSM, reportando ao usuário esta situação na tela do dispositivo.

Estas tarefas exigiram aprofundamento da arquitetura do Arduino, tendo em vista a memória SRAM de apenas 2.048 *bytes*, a forma pela qual o compilador trata os textos estáticos e também os problemas gerados pelo uso inicial da classe *String*. Tais dificuldades serão expostas no detalhamento da implementação.

7.2.CASOS DE USO

Baseado nos requisitos evidenciados, foi efetuada uma análise das funcionalidades que deveriam ser implementadas. Esta análise visou cumprir com os requisitos e também verificar se havia alguma característica quanto ao funcionamento do módulo que não foi evidenciada. Tal verificação proporcionou uma idéia mais detalhada das características que o módulo precisava possuir, em relação a sua interação com o usuário.

7.2.1.Uso local

O usuário fisicamente próximo ao módulo irá efetuar uma leitura direta da tela LCD presente no dispositivo. Na figura 12, foram evidenciados os tipos de usuário: alunos de escolas de aviação, instrutores de escolas de aviação, pilotos privados, pilotos comerciais e pessoal operacional.

Figura 12 - caso de uso - uso local



Fonte: Próprio autor

Neste caso de uso, será efetuada uma simples consulta ao METAR informado pelo dispositivo, sendo suficiente para o usuário a exibição permanente desta informação na tela, visto que estes usuários tem o hábito de anotar o METAR para consulta posterior antes da decolagem e durante o voo quando no regresso ao mesmo local.

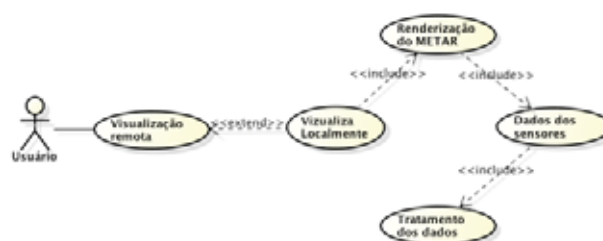
A visualização local depende que um METAR seja montado, com base nos dados dos sensores. A obtenção dos dados dos sensores, por sua vez, exige que alguns deles sejam tratados antes de serem retornados.

7.2.2. Uso remoto

Os usuários deste cenário são os mesmos do caso de uso local, porém restritos a conexão através da Internet. As informações são obtidas através do site WUNDERGROUND, efetuando-se a consulta pelo código da estação previamente obtido ou então buscando a mesma através da localidade. Este caso de uso foi documentado pois é viabilizado por implementação no módulo, porém a interface com o usuário é implementada pelo WUNDERGROUND e não pode ser alterada. Como melhoria futura, pode ser elaborado um site que consulte os dados da PWS através da mesma API, disponibilizando eles no formato METAR.

A figura 13 demonstra este caso de uso, que basicamente estende as funcionalidades do caso de uso para consulta local. Esta extensão ocorre através da implementação da comunicação do módulo com o WUNDERGROUND. Esta análise proporcionou uma visão da organização que o código poderia possuir, onde uma implementação poderia ser expandida e desta forma atender ao caso de uso.

Figura 13 - caso de uso - uso remoto



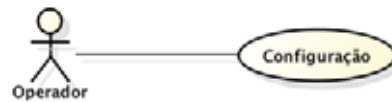
Fonte: Próprio autor

7.2.3. Configuração

Este caso de uso tem como usuário o operador do dispositivo. Não há restrição de acesso a configuração do equipamento, como por exemplo com o uso de senhas, devido ao reduzido número de botões implementados. Esta característica é facilmente sanada com o uso de restrições físicas quanto ao acesso não autorizado, através do uso de chaves para acesso aos botões de configuração.

Neste procedimento, o usuário irá configurar o módulo com os parâmetros: data e hora UTC, código ICAO, latitude e longitude do aeródromo onde o equipamento está instalado. A figura 14 ilustra a análise deste caso.

Figura 14 - caso de uso - configuração

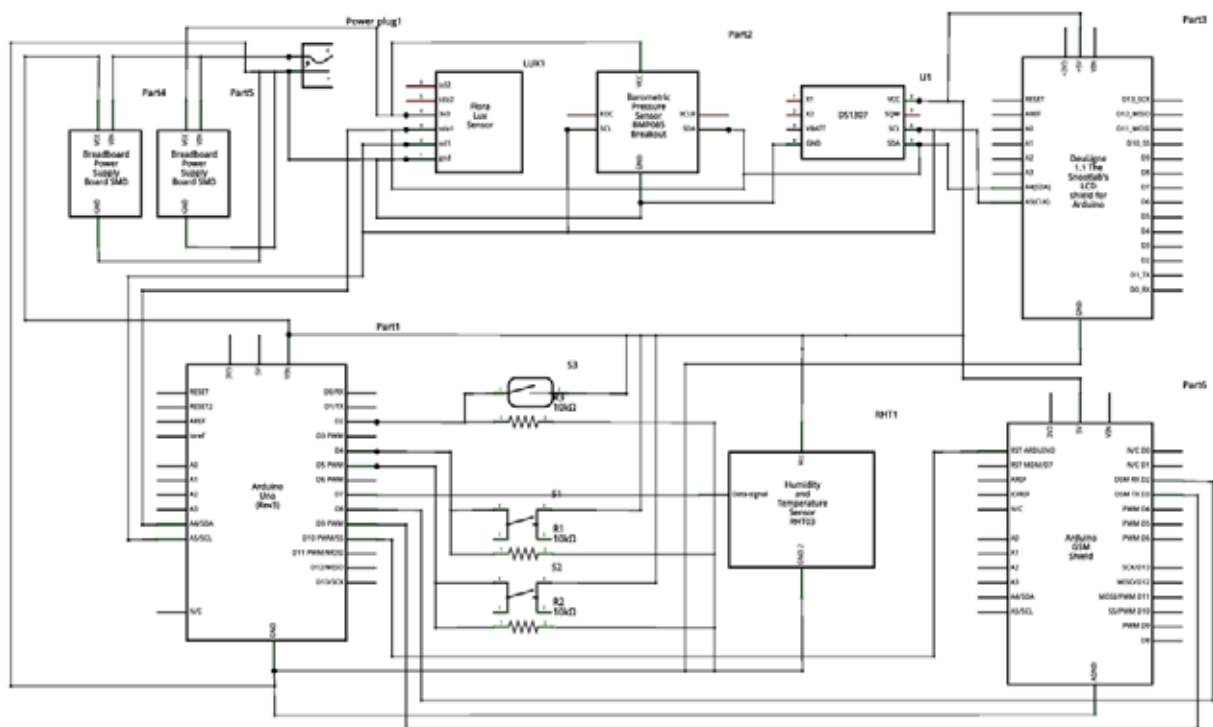


Fonte: Próprio autor

7.3. HARDWARE

É composto pelo Arduino, sensores, componentes eletrônicos e display LCD. A figura 12 ilustra o diagrama esquemático deste projeto.

Figura 15 - diagrama esquemático.



Fonte: próprio autor.

O protótipo final foi construído em uma *proto-board*, que é uma placa específica para prototipação eletrônica. Esta placa permite inúmeras alterações no diagrama esquemático conforme ele evolui durante a implementação e escolha dos sensores, sendo que foi totalmente adequada ao modelo de desenvolvimento por ciclos compostos por protótipos. O produto final deste projeto compreende um módulo totalmente funcional e apto para ser tornado um produto para comercialização.

7.4.FIRMWARE

Contém a implementação de toda a lógica de funcionamento do módulo de coleta, processamento, exibição e transmissão das condições meteorológicas utilizando o formato METAR. A linguagem utilizada foi o C no padrão Arduino, o qual adota um padrão codificação que permite a criação de bibliotecas semelhante a programação orientada a objetos. Este padrão é amplamente utilizado pelos desenvolvedores do Arduino e todas as bibliotecas utilizadas para a implementação fizeram uso deste benefício, tornando o código legível e permitindo múltiplas instâncias de uma mesma biblioteca, replicando suas funcionalidades sem necessidade de adequação do código.

Outra padronização do Arduino refere-se a segmentação do código fonte, que deve possuir de forma separada as implementações de inicialização e execução das rotinas que serão executadas. Tratando-se de um componente eletrônico, sua configuração deve ser estabelecida durante a inicialização, não sendo mais alterada. Já as rotinas executadas serão processadas de forma ininterrupta pelo Arduino, em uma repetição inifinta até que sua alimentação seja eliminada ou um pedido de reinicialização seja efetuado. No Arduino, a reinicialização ocorre a pedido do usuário através de um botão físico com esta finalidade.

O código fonte do módulo foi segmentado sequencialmente de forma explicitada pelos itens descritos a seguir. A figura 16 ilustra o diagrama de componentes

7.4.1.Bibliotecas utilizadas

A biblioteca SoftwareSerial foi utilizada para viabilizar a comunicação serial com o componente responsável pela conexão com a Internet e pelo protocolo HTTP utilizado para transmitir os dados. Seria possível obter o mesmo resultado utilizando-se a porta serial implementada pelo *hardware* do Arduino, porém seria eliminado o uso desta porta para a depuração da execução do código durante os ciclos de desenvolvimento. Tal depuração é fundamental para o processo de desenvolvimento, pois o Arduino trata-se de um componente eletrônico avulso ao computador no qual o código que será executado está sendo desenvolvido, e portanto não possui nativamente uma ferramenta para depuração, delegando ao desenvolvedor a forma que isto será efetuado. A saída de mensagens e valores através da porta serial conectada ao computador é comumente utilizada e proporciona parcialmente esta funcionalidade. Como melhoria e considerando que a implementação do módulo está finalizada, pode-se eliminar o uso de tal biblioteca.

As bibliotecas BMP085 e DHT foram utilizadas para comunicação com os sensores de temperatura e pressão e de umidade relativa do ar, respectivamente. A primeira utiliza a biblioteca Wire, que implementa o protocolo I2C e viabiliza a comunicação com dispositivos que adotem este protocolo, consumindo apenas duas portas digitais do Arduino para diversos

dispositivos. A segunda possui um protocolo próprio, definido pelo fabricante do sensor, o qual demanda uma porta digital isolada no Arduino para comunicação.

A biblioteca RTCLib foi utilizada para gravar e ler a data e a hora em componente possuir de um relógio de tempo real, alimentado por bateria e memória próprias, capaz de armazenar tais informações sem durante o tempo que o módulo estiver desligado.

Como evolução da intenção inicial de utilizar apenas um *display* como tela do módulo, tendo em vista a adoção do protocolo I2C como prioritário para a comunicação com os componentes do módulo, foi adotada a biblioteca LiquidCrystal_I2C. Esta biblioteca compreende as mesmas funcionalidades da biblioteca padrão para *displays* LCD do Arduino porém permite o uso do protocolo I2C para a comunicação com um ou mais *displays*.

A implementação de tarefas periódicas determinou o uso da biblioteca Metro, que possui uma implementação simples porém conservadora aos princípios do Arduino, tais como não alterar o valor do contador de tempo. Sempre buscando utilizar implementações que permitam o reuso, foi adotada esta biblioteca para o escalonamento de tarefas.

A biblioteca EEPROM foi utilizada para ler e gravar os dados de configuração na memória EEPROM do Arduino, visando recuperá-las sempre que o módulo for ligado.

A biblioteca Math foi utilizada tendo em vista funções para o cálculo dos horários de nascer e pôr do sol em determinada latitude, as quais possuem funções matemáticas complexas para implementação no Arduino, tais como seno e cosseno.

7.4.2. Textos estáticos

O uso de textos estáticos no Arduino exigiu aprofundamento no estudo da forma como o compilador trata deste tipo de dado. Nos ciclos de desenvolvimento iniciais do projeto, tais textos foram implementados utilizando-se código C, utilizando os textos como parâmetros. Adicionalmente, os textos que não eram estáticos também foram codificados da forma mais simples, utilizando a classe *String* do Arduino sempre que necessário. Tal classe, implementa o uso de vetores de caracteres visando o uso facilitado e efetuando diversas conversões de tipo se necessário.

Entretanto, com o avanço do projeto, foram identificados problemas referentes ao uso indiscriminado da classe *String*. A facilidade proporcionada por ela tinha como ponto fraco o elevado consumo de memória SRAM, que é a memória de 2KB disponível para o Arduino para armazenamento de dados não estáticos, tais como variáveis e a pilha. Os sintomas percebidos foram a perda de dados e reinicialização do dispositivo de forma abrupta e sempre quando era utilizada a classe *String*. A solução, parcial, encontrada foi refatorar o código convertendo o uso da classe *String* para vetores de caracteres utilizados na linguagem C.

Porém, nos ciclos finais de desenvolvimento onde existia texto em grande quantidade, como por exemplo os comandos AT enviados ao componente GSM, novamente foram percebidas perda de dados e reinicializações cuja origem não era a implementação. Foram necessários estudos adicionais relativos ao uso de *strings* no Arduino, os quais revelaram uma peculiaridade do compilador: toda e qualquer *string* declarada de forma estática no código era armazenada na memória SRAM do Arduino. Este fato associado ao uso intenso de *strings* tornou evidente que a memória SRAM do Arduino era muito pequena na forma em que eram implementadas as *strings* estáticas.

Para solucionar o problema, adotou-se o uso da diretiva de compilador *PROGMEM*. Tal diretiva informava ao compilador que o conteúdo estático declarado deveria ser

armazenado na memória *flash* do Arduino, a qual possui 32KB. Adicionalmente, foi implementada uma função de apoio para recuperar as *strings* declaradas desta forma, impedindo que o compilador gerasse código que armazenasse tais *strings* na memória SRAM, armazenando elas temporariamente nesta memória apenas durante a execução do código que as utilizasse, liberando a mesma em seguida.

Diante do exposto acima, esta seção do código contém todas as 46 constantes strings usadas desde a exibição de mensagens ao usuário até a comunicação com o módulo GSM. Com esta prática, a memória EEPROM utilizada no protótipo final do módulo foi de 28.767 bytes, o que corresponde a 87% da memória total disponível para código e dados estáticos.

7.4.3.Inicialização das bibliotecas

Nesta etapa são inicializadas as bibliotecas que serão utilizadas. Tais declarações equivalem a criação de instâncias de objetos, onde é efetuada uma chamada ao construtor da biblioteca com parâmetros para a sua inicialização.

As bibliotecas RTC e BMP085 são inicializadas sem parâmetros, tendo em vista que ambas utilizam o protocolo I2C e ambas acessam componentes cujos endereços no barramento I2C são fixos. A biblioteca DHT é inicializada informando a porta digital onde o componente está conectado e também o tipo de sensor utilizado, neste caso um DHT22. A biblioteca LiquidCrystal_I2C é inicializada com o endereço do *display* no barramento I2C e também a quantidade de linhas e colunas suportadas pelo mesmo. Finalmente, a biblioteca SoftwareSerial é inicializada com as portas digitais utilizadas para os barramentos de envio e recebimento de dados via padrão serial.

7.4.4.Temporizador de tarefas

Nesta seção de código foram declaradas as tarefas escalonadas através de temporizadores implementados com o uso da biblioteca Metro. São inicializados contadores para as tarefas: leitura de sensores, renderização e exibição do METAR no *display* local, publicação na Internet, tratamento da leitura do anemômetro, reinicialização das leituras de velocidade média e rajadas de vento, leitura da luminosidade e cálculo da cobertura de nuvens.

7.4.5.Variáveis globais

Aqui são declaradas 28 variáveis de uso global. Foi adotado este padrão para as variáveis de uso comum visando minimizar o uso da memória SRAM, o qual ocorre sempre que uma variável é enviada para uma função através da pilha. Devido aos problemas ocorridos com *strings* explicados anteriormente, foi priorizada a minimização de uso da SRAM sempre que possível.

7.4.6.Inicialização

Neste trecho foi implementada a inicialização do Arduino. Nesta etapa são configuradas todas as portas digitais e analógicas que foram utilizadas no módulo, assim como demais configurações tais como a interrupção que trata a leitura do anemômetro.

A primeira configuração é do display, limpando o seu conteúdo e exibindo a mensagem de versão e inicialização do módulo.

Na sequência, é configurada a porta digital utilizada para enviar um pulso de três segundos ao componente GSM, o qual aciona a energização do mesmo. Após configurada a porta, é chamada a rotina que efetua o procedimento descrito, ligando o componente GSM. São efetuadas três tentativas de ligar o componente, marcando o mesmo como inoperante após estas tentativas e sendo exibido no *display* a mensagem *offline*. Basicamente, a rotina envia o pulso e em seguida envia o comando "AT" via comunicação serial, o qual em caso de sucesso deve retornar um "OK". A rotina aguarda até dois segundos pelo retorno positivo, efetuando nova tentativa caso o componente GSM não responda.

Após a inicialização do modem GSM, são configuradas as portas digitais referentes aos botões de ação e ao *Light Emitting Diode* (LED) indicador de atividade do anemômetro. O LED permite ao instalador verificar se o anemômetro está corretamente conectado, emitindo um pulso luminoso a cada volta do anemômetro. Já a configuração dos botões indica ao Arduino que as portas são de entrada, procedimento necessário visto que cada porta pode assumir somente um estado: entrada ou saída.

Neste momento são chamadas funções para inicialização das bibliotecas de comunicação via barramento I2C e de leitura do sensor de luminosidade. Em seguida, também é efetuada a inicialização da biblioteca que trata do relógio de tempo real, porém verificando se o mesmo contém uma data e hora válidas, inserindo a data e hora da compilação do código caso contrário. Isto visa inicializar o relógio de tempo real com uma data e hora válida e próxima do instante atual.

O próximo passo é a leitura do código ICAO armazenado na EEPROM do Arduino. São lidos quatro *bytes* sucessivos da memória EEPROM e armazenados em um vetor de caracteres. Este vetor possui o valor inicial "SBXX" o qual será substituído pelos caracteres lidos por este procedimento, salvo se a memória contiver caracteres inválidos, que estejam além dos limites do padrão ASCII. Neste caso, será mantido o caractere existente no vetor.

Outro valor armazenado na memória EEPROM e lido nesta etapa são as coordenadas GPS configuradas pelo usuário. Análogo ao procedimento anterior, os valores padrão para ambas latitude e longitude são zero, salvo se a memória contiver um valor válido, que correspondem ao intervalo de -90 até +90 para a latitude e -180 até 180 para a longitude. As coordenadas são armazenadas no formato de ponto flutuante em tempo de execução, porém na memória EEPROM são armazenadas em graus, minutos e segundos. É efetuada uma conversão entre os formatos, tanto para salvar quanto para recuperar as coordenadas da memória EEPROM.

Neste momento são inicializadas as bibliotecas dos sensores de umidade relativa do ar e de pressão atmosférica e temperatura do ar. A seguir, são inicializadas as variáveis globais que armazenam os dados de controle do anemômetro e sua interrupção, assim como é inicializada a interrupção de *hardware* que é executada a cada revolução do anemômetro. O fluxo completo de funcionamento do tratamento deste sensor será detalhado posteriormente.

Finalizando, são inicializadas as variáveis globais para tratamento da cobertura de nuvens e é exibido o padrão de testes no *display* LCD por dois segundos. Após este tempo, o procedimento de inicialização é finalizado e o controle do programa transferido ao *loop* principal.

7.4.7. Funções primárias e de apoio

Neste bloco de código foram implementadas funções que executam tarefas primárias, acionadas pelo código do bloco principal de *loop*, assim como funções de apoio tais como conversões de formatos e formatação.

Dentre elas, podemos destacar as funções de envio dos comandos AT para o componente GSM, a rotina que liga o componente GSM, as funções que retornam os dados lidos dos sensores conectados ao módulo, a rotina de publicação dos dados na Internet, a rotina que trata da interrupção referente a revolução do anemômetro, as rotinas que montam e exibem o METAR na tela LCD, as funções de transformação de coordenadas decimais para angular e as rotinas de formatação de números.

7.4.8. Laço principal

Trata-se do bloco *loop* padrão do Arduino, o qual contém o código que será repetido infinitamente, conforme explicado anteriormente. Foi implementado neste trecho de código o tratamento das tarefas escalonadas, assim como o bloco que permite a configuração do módulo.

As tarefas escalonadas são ativadas conforme a sinalização da biblioteca Metro. É efetuada uma chamada ao método *check*, que retorna se a atividade teve seu tempo de execução expirado ou não. Se expirado, são efetuadas as tarefas pertinentes e o contador do tempo de execução é reinicializado através de chamada ao método *reset*. Tal implementação não garante que as tarefas serão executadas impreterivelmente na periodicidade estabelecida, podendo haver margens de tempo superiores. Devido a isto, todas as tarefas que estimam dados através de cálculos derivados do tempo decorrido foram implementadas com base no contador de tempo interno do Arduino, que retorna um inteiro longo sem sinal cuja contagem se inicia no momento que o Arduino é energizado e só termina quando ele é desligado.

As tarefas escalonadas basicamente se dividem em: leitura dos sensores, tratamento dos dados, exibição local e transmissão para a Internet. Além delas, temos a interface de configuração que interrompe todas as tarefas, quando acionada, até o término da configuração exceto o tratamento da interrupção do anemômetro, que permanece ativa de forma ininterrupta.

A configuração foi implementada seguindo a sequência de telas. São exibidos os dados relativos a tela atual e processado a interpretação dos botões, efetuando o incremento do parâmetro selecionado ou o avanço para a próxima tela ou término da configuração, conforme o caso.

7.5. TRATAMENTO DOS DADOS

Das informações coletadas pelo módulo através dos sensores, quatro necessitam de tratamento para que possam ser utilizadas: a umidade relativa do ar, a velocidade do vento, as rajadas de vento e a cobertura de nuvens. Para todas não existe uma leitura direta mas sim a interpretação de dados que possibilitam estimar os dados necessários.

7.5.1. Velocidade do vento

O anemômetro é um sensor que mede a velocidade do vento, sendo que foi utilizado o anemômetro do tipo concha, possuindo três conchas. Este tipo foi selecionado devido ao fato de perceber o vento oriundo de qualquer direção e pela disponibilidade e preço no mercado. Foram efetuados orçamentos e pesquisas na Internet, obtendo-se resultados demasiadamente onerosos ou impraticáveis dentro do tempo disponível para aquisição e implementação.

Determinado o tipo de sensor e após adquirido o mesmo, foi necessário calibrá-lo pois o fabricante não atendeu a este requisito de forma satisfatória, utilizando um velocímetro de bicicleta como medidor da velocidade do vento, o qual estava desconfigurado para o sensor adquirido. Foram efetuadas novas buscas, tanto através da Internet quanto localmente em Porto Alegre, visando encontrar uma forma de calibrar o sensor. Em São Paulo foi localizada uma empresa que efetua a calibragem se o sensor é enviado por Correios, ao custo de mais de trezentos reais. Em Porto Alegre, ambas as faculdades PUC e UFRGS não retornaram os pedidos de uso dos respectivos túneis de vento que ambas possuem, com a finalidade de calibrar o anemômetro. Com o custo elevado da calibragem remota, foi efetuado este procedimento utilizando um anemômetro portátil, calibrado de fábrica, que foi validado em relação as velocidades de vento indicadas pelo Aeroporto Internacional Salgado Filho (SBPA). Após verificado que o anemômetro portátil indicava corretamente a velocidade do vento, o mesmo foi utilizado na calibragem do sensor adquirido, através de velocidades diferentes de vento em bancada, permitindo analisar o comportamento do sensor em diferentes situações. Foi constatado que o sensor adquirido responde bem à mudança de velocidade do vento, demonstrando uma margem de erro de 1 a 2 nós em relação ao anemômetro aferido. Adicionalmente, o sensor adquirido é pouco sensível a ventos fracos, onde a margem de erro se manifesta com maior magnitude, ainda que dentro do intervalo de 1 a 2 nós.

O anemômetro adquirido possui um sensor de rotação, que gera um pulso a cada volta do seu eixo, impulsionado pelas conchas presas a ele. A transformação deste tipo de informação na velocidade do vento em nós foi efetuada através da contagem dos pulsos e posterior cálculo da velocidade do vento.

Os pulsos gerados pelo anemômetro ocasionam uma interrupção de *hardware* no Arduino, o qual imediatamente desvia a execução do código para a rotina que trata desta interrupção, retornando a execução normal após o término. A rotina de interrupção precisa executar da forma mais rápida possível, visando diminuir o impacto na execução do sistema. A rotina de interrupção é configurada para ser acionada sempre que ocorrer uma alteração de estado na porta digital onde o sinal do anemômetro é conectado. Desta forma, a interrupção é gerada sempre que o valor da porta for alterado, por mais breve que isto ocorra.

A rotina implementada conta com uma variável sinalizadora que indica o último estado de leitura do anemômetro: pulso baixo ou pulso alto. Os termos baixo e alto referem-se a voltagem percebida pela porta digital do Arduino, correspondendo as voltagens de zero volts e cinco volts, respectivamente. Com este controle, a rotina somente irá tratar das alterações de estado, desconsiderando as interrupções que ocorram através de ruído eletrônico, as quais ocasionam uma oscilação de pulso muito rápida, que pode ser percebido pelo Arduino como mudança de estado. Ao ocorrer uma troca de estado, este é atribuído a variável de estado e um contador é configurado para o instante atual adicionado de uma constante, que varia conforme o estado é alto ou baixo. Desta forma temos uma variável que controla o estado e outra que

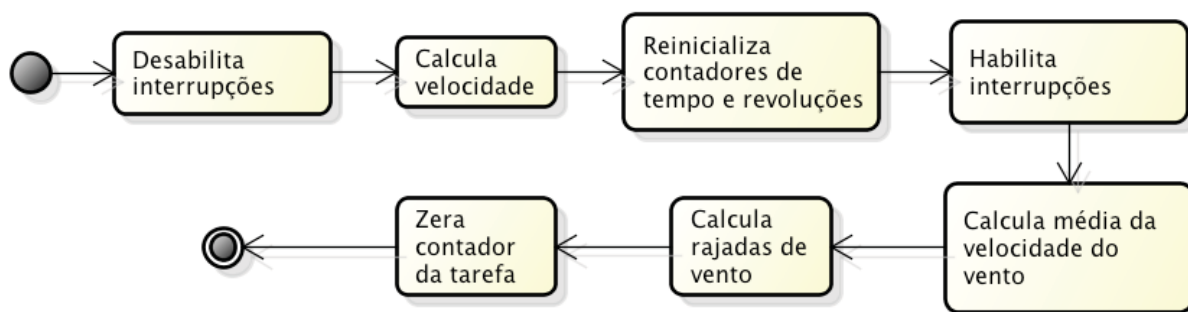
exige por quanto tempo o estado deve existir para que seja efetivamente considerado. Os tempos exigidos para valores altos são menores que os tempos para os valores baixos, tendo em vista que em uma revolução de 360 graus apenas em 10 graus o sinal assume o estado alto.

Para cada estado alto o LED indicador é acionado e o contador de revoluções é incrementado. Com este procedimento finalizou-se o tratamento das interrupções, cujo produto é a quantidade de voltas que o sensor efetuou.

A transformação deste dado na velocidade do vento é efetuada através de uma tarefa do escalonador do *loop* principal. Nesta tarefa, é considerado a quantidade de voltas mensuradas pela interrupção em relação ao tempo transcorrido durante o período de escalonamento da tarefa. Com isto, gera-se o valor de rotações por minuto geradas pelo anemômetro.

A partir desta informação e com base na calibragem efetuada, pode-se obter a velocidade instantânea do vento em nós multiplicando-se as rotações por minuto pela constante gerada através do processo de calibragem. Esta constante transforma as revoluções do anemômetro em velocidade do vento, de forma linear. Durante a calibragem, não foi percebida uma variação considerável entre as rotações baixas e elevadas do anemômetro, salvo nas rotações mais baixas conforme salientado anteriormente. Nelas, o conjunto físico de conchas, hastes e rolamentos não proporcionou revoluções suaves devido ao fraco vento e a rigidez do conjunto, assim como o seu peso e inércia. A figura 16 ilustra o diagrama de atividades desta implementação.

Figura 16 - implementação da tarefa relativa ao vento



Fonte: Próprio autor

A informação gerada pela tarefa descrita é a estimativa da velocidade instantânea do vento, que corresponde a velocidade do vento mensurada pelo anemômetro no intervalo de tempo. Isto ocorre a cada cinco segundos, tempo estabelecido através de testes que demonstraram uma uniformidade maior dos dados, principalmente em condições de vento fraco. A velocidade do vento informada no METAR gerado pelo módulo é calculada com base na média das velocidades instantâneas medidas em um intervalo de um minuto, amortizando oscilações muito bruscas ou a falta de captação de ventos fracos.

7.5.2. Rajadas do vento

Para o cômputo das rajadas de vento foi implementado um algoritmo simples, que atribui a velocidade do vento instantâneo sempre que ela for maior que o valor armazenado em outra variável. Este processamento é reinicializado, ou seja, a variável que contém a rajada

de vento é inicializada com o valor zero a cada dez minutos, a fim de atender ao padrão do formato METAR que determina a exibição de rajadas ocorridas em até dez minutos subsequentes ao instante atual.

7.5.3. Cobertura de nuvens

A cobertura de nuvens é estimada com base na leitura do sensor de luminosidade, que retorna um valor correspondente em Lux, que é uma métrica para medir a quantidade de luz percebida em determinado local. Apenas a leitura da luminosidade percebida pelo sensor não fornece dados suficientes para determinar se o céu tem presença de nuvens ou não. Foi necessário pesquisar um cálculo que fornecesse um referencial, no qual fosse possível determinar um valor lux presumido para um dia e horário determinado pelo relógio do módulo.

Com base neste valor presumido, o algoritmo consegue estimar se existe ou não presença de algum tipo de obstrução entre o sensor e a luz do sol. Por este motivo, o sensor deve estar em local em que não possua fonte de sombra ao seu redor, visto que a mesma poderia acarretar em uma estimativa ineficiente da cobertura de nuvens.

Ao perceber a presença de obstáculo que esteja causando uma luminosidade menor que a esperada, o módulo passa a contar o intervalo de tempo em que este fenômeno ocorre. Ao término da obstrução, o módulo verifica o intervalo transcorrido e deduz a presença de nuvens da seguinte forma:

- a) se o intervalo foi de até 5 segundos, é considerado que ocorreu uma obstrução não natural, como por exemplo devido a passagem de um avião, e é considerado céu limpo;
- b) se o intervalo foi entre 6 e 20 segundos, é considerado a presença de poucas nuvens, sendo indicado no METAR do módulo com o texto FEW;
- c) se o intervalo foi entre 21 e 120 segundos, é considerado céu parcialmente nublado, sendo indicado no METAR do módulo com o texto SCT;
- d) se o intervalo foi entre 121 e 480 segundos, é considerado céu nublado, sendo indicado no METAR do módulo com o texto BKN;
- e) se o intervalo foi maior que 480 segundo, é considerado céu encoberto, sendo indicado no METAR do módulo com o texto OVC.

Os tempos utilizados para considerar os estados de cobertura do céu foram estimados com base em observações práticas, sendo que não podem ser considerados de precisão tendo em vista que fatores como a velocidade do vento pode alterar significativamente tais tempos.

A análise de cobertura do céu só é efetuada no intervalo de tempo compreendido entre o nascer e o por do sol, ignorando completamente o sensor de luminosidade fora deste intervalo e não reportando dados de cobertura do céu no LCD do módulo durante este intervalo, devido ao fato do sensor fazer a estimativa utilizando a luminosidade provida pelo sol.

Para viabilizar esta estimativa, foi necessário calcular a luminosidade prevista em determinado dia e horário do ano. Para isto, foi deduzido a altitude do sol em graus (ELVEGARD, 1940) a qual permite posteriormente deduzir a massa de ar existente entre o sol e a superfície do local parametrizado e finalmente estimar a luminosidade do céu limpo e do céu encoberto.

Elvegard explica em seu artigo fórmulas matemáticas que viabilizam este cálculo. Basicamente, é necessário a latitude do local estipulado, a declinação do sol no dia desejado, a

correção de minutos de acordo com a equação do tempo que também está associada a declinação do sol e a diferença do horário desejado em relação ao meio dia verdadeiro, que é calculado utilizando-se a correção de minutos. A figura 17 demonstra a tabela de declinação do sol e equação do tempo utilizada.

Figura 17 - tabela de declinação do sol e equação do tempo

Date	Declination	Equation of Time	Date	Declination	Equation of Time
January 1	-23°.1	+3 ^m	July 9	+22°.5	+5 ^m
10	-22.1	+7	19	+21.0	+6
20	-20.3	+11	29	+18.9	+6
30	-17.8	+13	August 8	+16.4	+6
February 9	-14.9	+14	18	+13.3	+4
19	-11.5	+14	28	+10.0	+1
March 1	-7.8	+13	September 7	+6.3	-2
11	-4.0	+10	17	+2.5	-5
21	0.0	+7	27	-1.4	-9
31	+3.9	+4	October 7	-5.2	-12
April 10	+7.7	+2	17	-9.0	-14
20	+11.3	-1	27	-12.6	-16
30	+14.6	-3	November 6	-15.8	-16
May 10	+17.5	-4	16	-18.6	-15
20	+19.8	-4	26	-20.8	-13
30	+21.7	-3	December 6	-22.4	-9
June 9	+22.9	-1	16	-23.3	-5
22	+23.5	+2	23	-23.5	-1
30	+23.2	+3	30	-23.2	+2

Fonte: Illuminating Engineering, Abril de 1940. Pag. 338.

Para este cálculo, faz-se necessário converter o horário armazenado no módulo em horário local, com correção do fuso horário. A figura 18 ilustra a fórmula matemática utilizada para determinar a altitude do sol no horizonte em graus:

Figura 18 - fórmula para estimar a altitude do sol em graus

$$\sin b = \sin \varphi \sin \delta + \cos \varphi \cos \delta \cos 15 t$$

Fonte: Illuminating Engineering, Abril de 1940. Pag. 337.

Com base na altitude do sol, é possível determinar a massa de ar existente entre o sol e o sensor de luminosidade, através da tabela ilustrada pela figura 19.

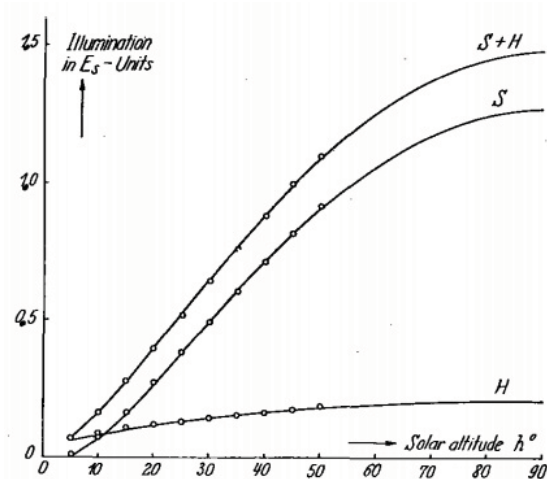
Figura 19 - tabela de conversão de altitude para massa de ar

Solar Altitude	0°	1°	2°	3°	4°	5°	6°	7°	8°	9°
0°	—	26.96	19.79	15.36	12.44	10.40	8.90	7.77	6.88	6.18
10	5.60	5.12	4.72	4.37	4.08	3.82	3.59	3.39	3.21	3.05
20	2.90	2.77	2.65	2.55	2.45	2.36	2.27	2.20	2.12	2.06
30	2.00	1.94	1.88	1.83	1.78	1.74	1.70	1.66	1.62	1.59
40	1.55	1.52	1.49	1.46	1.44	1.41	1.39	1.37	1.34	1.32
50	1.30	1.28	1.27	1.25	1.24	1.22	1.20	1.19	1.18	1.17
60	1.15	1.14	1.13	1.12	1.11	1.10	1.09	1.09	1.08	1.07
70	1.06	1.06	1.05	1.05	1.04	1.04	1.03	1.03	1.02	1.02
80	1.02	1.01	1.01	1.01	1.01	1.00	1.00	1.00	1.00	1.00
90	1.00									

Fonte: Illuminating Engineering, Abril de 1940. Pag. 337.

A partir da massa de ar, é possível estimar a quantidade de luz em Lux, tanto para a condição de céu claro quanto para céu coberto por nuvem, permitindo ao módulo detectar a presença de nuvens e processar a estimativa de cobertura de nuvens com base no algoritmo explicado anteriormente. A figura 20 demonstra o resultado final dos cálculos, onde se obtém de forma separada a luminosidade obtida do sol e a obtida da radiação do céu.

Figura 20 - curvas de exemplo do resultado dos cálculos de luminosidade



Fonte: Illuminating Engineering, Abril de 1940. Pag. 337.

7.5.4. Temperatura de ponto de orvalho

A estimativa da temperatura de ponto de orvalho, sem possuir um instrumento ou sensor que permita mensurar a mesma diretamente, exige complexos cálculos matemáticos (LAWRENCE, 2005). Porém, através de uma fórmula simples pode-se estimar esta

temperatura com precisão de um grau Celsius para mais ou para menos, em ambientes cuja umidade do ar seja superior a 50% (LAWRENCE, 2005). Esta fórmula, exibida na figura 21, foi escolhida devido a sua simplicidade de implementação e resultados obtidos, os quais se enquadram com os propósitos deste projeto.

Figura 21 - fórmula do ponto de orvalho

$$\text{TempOrvalho} = \text{Temperatura} - ((100 - \text{PercentualUmidade}) / 5)$$

Fonte: Próprio autor

7.6.COMUNICAÇÃO

Foi implementada o envio das informações meteorológicas coletadas e processadas pelo módulo para o site WUNDERGROUND, através de uma requisição GET do protocolo HTTP, conforme as especificações do WUNDERGROUND já explicitadas anteriormente. Para isto, foi utilizado o componente GSM que opera através de comandos no padrão AT emitidos através de comunicação serial.

Inicialmente, durante a inicialização do módulo, o componente GSM é ligado e a comunicação serial com o mesmo é estabelecida com uma velocidade de 19.200 bps. O componente GSM só será utilizado novamente quando a tarefa de publicação for acionada. Neste momento, será verificado se o componente GSM foi ligado com sucesso e se o mesmo apresenta registro na rede GSM, através do comando AT CREG. Se o componente não foi ligado com sucesso ou não está registrado na rede GSM, a tarefa é cancelada e a mensagem de dispositivo *offline* é exibida ao usuário na tela LCD local. Caso o componente GSM esteja corretamente ligado e registrado, é inicializado o procedimento de transmissão dos dados climáticos e é exibido ao usuário a mensagem "Enviando".

É efetuada então uma conexão interna no componente GSM, requerendo o módulo GPRS do mesmo através do comando AT CGATT. Este módulo é necessário para o trâmite de informações através de uma conexão de dados GSM. Em seguida, é configurado um perfil dentro do componente para seleção do tipo de dados e do APN que será utilizado, sendo escolhido o primeiro que estiver disponível. Para a operadora utilizada nos testes, a configuração do APN foi automática e instantaneamente assumida pelo componente GSM. Não foram efetuados testes com as demais operadoras de telefonia móvel. Em seguida, é inicializado um serviço HTTP no próprio componente GSM, que suporta não só o protocolo HTTP como também o FTP e TELNET⁶.

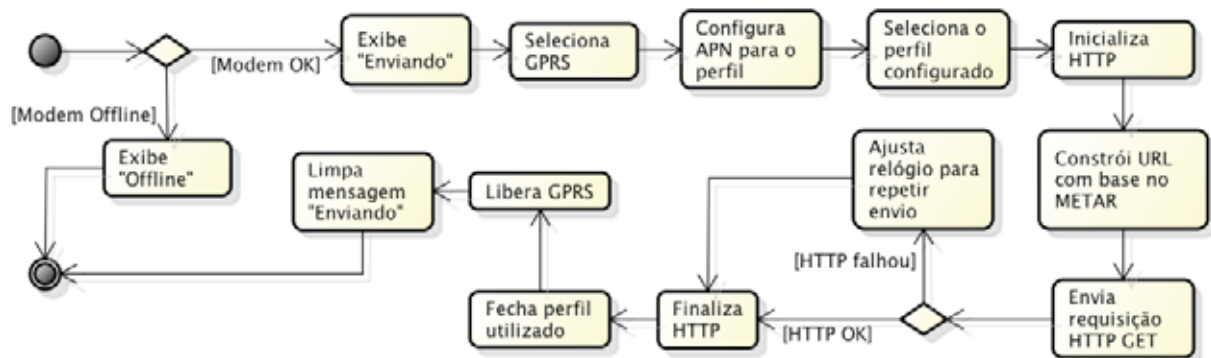
Após a inicialização da conexão HTTP, é efetuada a construção da URL que irá ser utilizada para a requisição HTTP, o que envolve a conversão de diversas informações em um vetor de caracteres adequado. A conversão ocorre de forma análoga a confecção do METAR, porém mais simples pois não está restrita as regras impostas por este formato. Entretanto, algumas conversões adicionais foram necessárias para adequar-se ao formato do WUNDERGROUND, como por exemplo a temperatura em fahrenheit, a pressão atmosférica em polegadas de mercúrio e as velocidades do vento em milhas por hora. Esta conversões foram resolvidas com simples fórmulas matemáticas.

⁶ Protocolo de transmissão em rede.

Após criada a URL, é enviada a requisição HTTP GET e processado o retorno. Caso seja diferente da mensagem *success*, será programado novo envio em três minutos. A especificação da API do WUNDERGROUND não permite uma implementação mais aprimorada, devido ao fato de não gerar um retorno padronizado em casos de falha.

Finalizada a transmissão ou falha desta, o processo inverso é executado em relação ao componente GSM, finalizando o serviço HTTP, fechando o perfil de conexão que foi aberto e desconectando o módulo interno de GPRS. A figura 22 mostra o diagrama de atividades da implementação da comunicação.

Figura 22 - implementação da comunicação

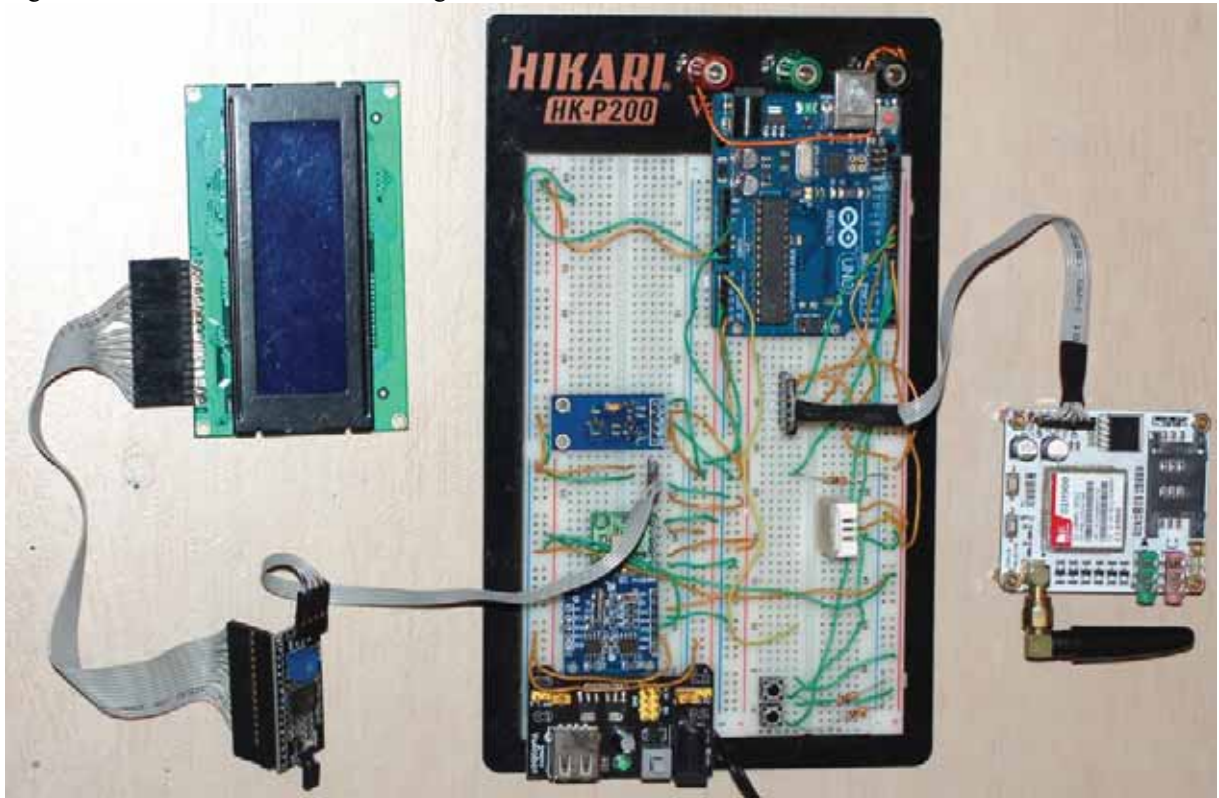


Fonte: Próprio autor

8.FUNCIONAMENTO DO SISTEMA

A figura 23 exibe o protótipo de *hardware* construído neste projeto. Será necessária a produtificação deste protótipo para a instalação em campo.

Figura 23 - hardware do módulo Voo Seguro



Fonte: próprio autor.

A implementação contém as seguintes funcionalidades: inicialização, monitoramento e configuração.

8.1.INICIALIZAÇÃO

A primeira tela exibida pelo módulo, conforme figura 23, contém uma mensagem de inicialização, visando indicar que o módulo está ligado e iniciando suas atividades.

Figura 24 - tela de inicialização.



Fonte: próprio autor.

Nesta etapa o componente GSM é acionado e verificado. Caso ocorra alguma falha neste processo, a mensagem de inicialização irá persistir por um tempo maior que o normal,

pois o módulo tenta acionar o componente GSM com mais de uma tentativa. Após a inicialização do módulo, é exibido um padrão no *display* para verificação de seu funcionamento durante três segundos, conforme a figura 24.

Figura 25 - tela de verificação.



Fonte: próprio autor.

8.2.MONITORAMENTO

É a tela principal do módulo, exibida na figura 25, onde é visualizada a informação METAR obtida pelo módulo. A informação é atualizada automaticamente, sem intervenção do usuário.

Figura 26 - tela de monitoramento.



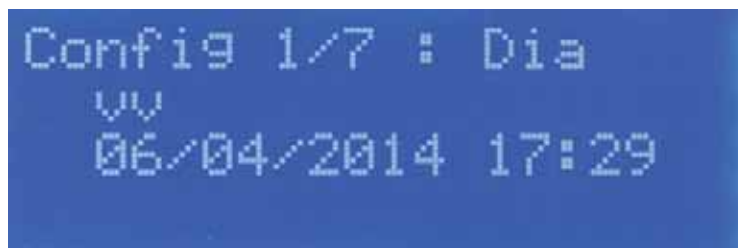
Fonte: próprio autor.

8.3.CONFIGURAÇÃO

Através do botão "Config" o usuário acessa as telas de configuração do módulo. Ao pressionar sucessivamente este botão, o usuário alterna entre os parâmetros e telas existentes. Durante a configuração são exibidos para referência a tela atual, o total de telas e a informação que será modificada, indicada pelo caractere "V". Ao final, o usuário retorna para o modo de monitoramento. Nas telas de configuração, ao pressionar o botão "Ação" a informação selecionada é alterada, sendo o comportamento relativo a informação que é modificada. Por exemplo: ao pressionar o botão "Ação" na tela de ajuste do dia, este é incrementado em um até atingir o valor máximo, quando então é retornado a 1 e reiniciado o incremento.

Na tela exibida pela figura 26, o usuário efetua a configuração do dia da data atual, atual. Através de consecutivos acionamentos do botão de "Ação", o dia é incrementado até que o dia desejado seja exibido. Ao ultrapassar o dia máximo permitido para o mês configurado, a contagem do parâmetro reinicia no valor um.

Figura 27 - tela de configuração do dia da data atual.



Fonte: próprio autor.

Na tela exibida pela figura 27, o usuário efetua a configuração do mês da data atual. Através de consecutivos acionamentos do botão "Ação", o parâmetro é incrementado até que o valor desejado seja exibido e configurado. Ao ultrapassar o décimo segundo mês, o parâmetro retorna para o valor um.

Figura 28 - tela de configuração do mês da data atual.



Fonte: próprio autor.

Na tela exibida pela figura 28, o usuário efetua a configuração do ano da data atual. Através de consecutivos acionamentos do botão "Ação", o parâmetro é incrementado até que o valor desejado seja exibido e configurado. Ao ultrapassar o valor 2030 a contagem reinicia no valor 2014. Anos posteriores a 2030 serão suportados mediante atualizações do firmware.

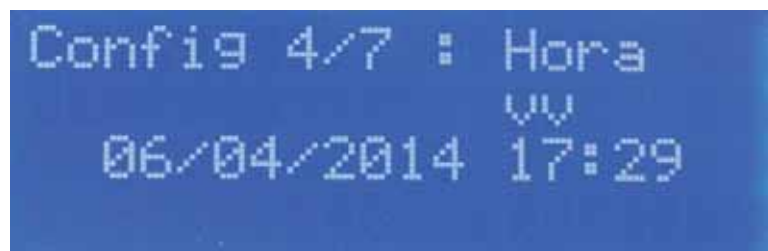
Figura 29 - tela de configuração do ano da data atual.



Fonte: próprio autor.

Na tela exibida pela figura 29, o usuário efetua a configuração da hora da data atual. Através de consecutivos acionamentos do botão "Ação", o parâmetro é incrementado até que o valor desejado seja exibido e configurado. Ao ultrapassar a vigésima terceira hora, o parâmetro retorna para o valor zero.

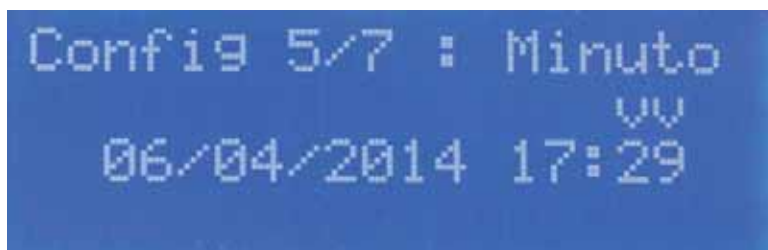
Figura 30 - tela de configuração da hora da data atual.



Fonte: próprio autor.

Na tela exibida pela figura 30, o usuário efetua a configuração do minuto da hora da data atual. Através de consecutivos acionamentos do botão "Ação", o parâmetro é incrementado até que o valor desejado seja exibido e configurado. Ao ultrapassar o valor cinquenta e nove, o parâmetro retorna para o valor zero.

Figura 31 - tela de configuração do minuto da data atual.



Fonte: próprio autor.

Na tela exibida pela figura 31, o usuário efetua a configuração do código ICAO do aeródromo a ser monitorado. Através de consecutivos acionamentos do botão "Ação", a letra selecionada é incrementada de forma alfabética, retornando ao início quando atingir a última letra. Ao pressionar o botão "Config", o usuário seleciona a próxima letra de maneira sequencial até que a última letra seja configurada, passando então para o próximo parâmetro.

Figura 32 - tela de configuração do código ICAO.



Fonte: próprio autor.

Na tela exibida pela figura 32, o usuário efetua a configuração da latitude em que o equipamento está instalado, utilizada no cálculo de cobertura de nuvens. O usuário incrementa os dois parâmetros existentes de forma sequencial, configurando a latitude em graus, seguida

pelos minutos. Os parâmetros estão limitados pelos seus valores máximos, reiniciando a contagem quando este valor máximo for ultrapassado.

Figura 33 - tela de configuração da latitude.



Fonte: próprio autor.

Na tela exibida pela figura 33, o usuário efetua a configuração da longitude em que o equipamento está instalado, utilizada no cálculo de cobertura de nuvens. O usuário incrementa os dois parâmetros existentes de forma sequencial, configurando a longitude em graus, seguida pelos minutos. Os parâmetros estão limitados pelos seus valores máximos, reiniciando a contagem quando este valor máximo for ultrapassado.

Figura 34 - tela de configuração da longitude.



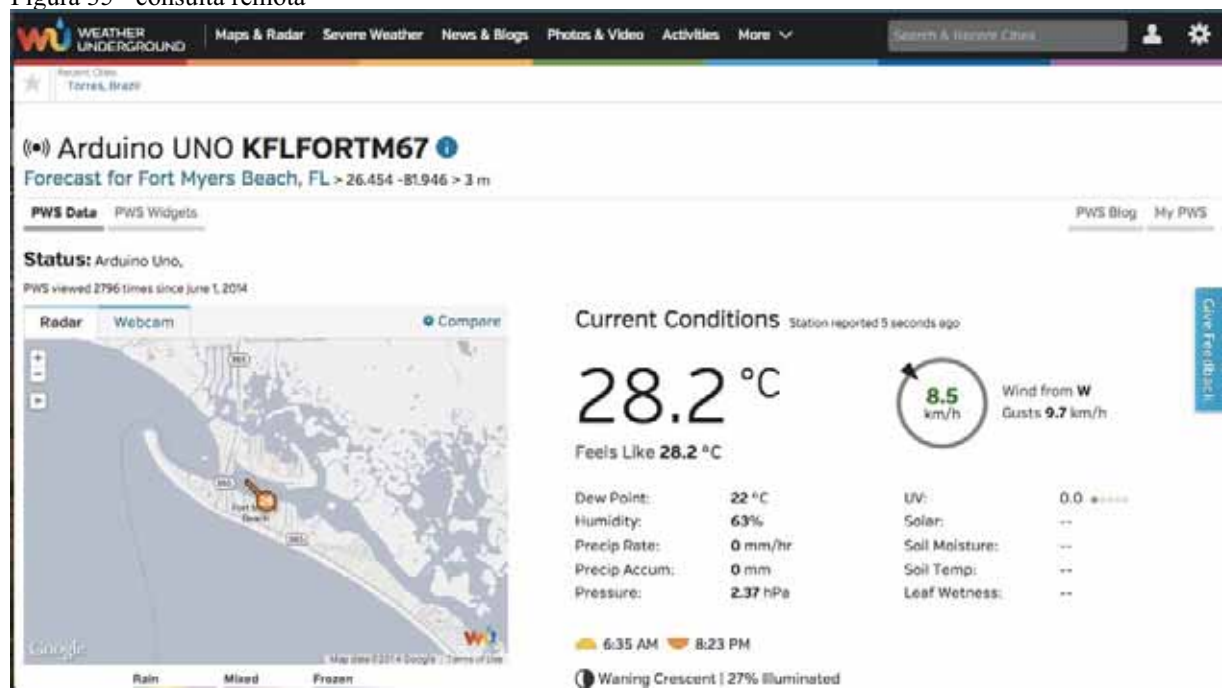
Fonte: próprio autor.

8.4. CONSULTA REMOTA

O usuário pode efetuar uma consulta remota aos dados do módulo através do site WUNDERGROUND, fornecendo o código da estação previamente obtido. Um exemplo de consulta pode ser visualizado na figura 35.

Os dados exibidos pelo WUNDERGROUND são os dados previamente transmitidos pelo dispositivo, não sendo realizada uma consulta remota diretamente ao dispositivo.

Figura 35 - consulta remota



Fonte: próprio autor.

9.VALIDAÇÃO

9.1.ESTRATÉGIA

As informações geradas pelo módulo através da leitura de seus sensores foram comparadas com as informações de um METAR oficial, sendo escolhido para esta finalidade o Aeroporto Internacional Salgado Filho em Porto Alegre. Somente as informações pertinentes a implementação deste módulo foram comparadas, desconsiderando-se as demais informações não implementadas, tais como a altitude das nuvens.

O equipamento foi instalado na cabeceira da pista 11 do referido aeroporto durante os períodos de teste, contemplando as premissas a seguir:

- a) a validação compreendeu três períodos de teste;
- b) um período de teste ocorreu em uma data única, não ocorrendo mais de um período de teste no mesmo dia;
- c) um período de teste compreendeu pelo menos quatro horas de teste, sendo que um período não sobrepôs o horário já validado por outro período de testes;
- d) um período de teste iniciou em hora pontual, com os minutos zerados, e finalizou da mesma forma. Sendo assim, a montagem do equipamento iniciou vinte minutos antes da referida hora pontual e sua respectiva finalização finalizou em até dez minutos após;
- e) todas as datas e horas utilizadas para configuração do módulo e aferição do mesmo durante os testes deveriam estar sincronizadas com a data e hora UTC informada pelo METAR do aeródromo.

Para cada período de teste, foram adotados os seguintes procedimentos:

- a) montagem do equipamento no mesmo local para todos os testes;
- b) anotação da data e hora em planilha de período de teste para validação do módulo;
- c) configuração do módulo no primeiro teste;
- d) aguardada até a hora pontual para início do teste;
- e) iniciado o teste, anotada a informação METAR do aeroporto e do módulo na planilha;
- f) a cada dez minutos contados a partir do início do teste, anotada a informação METAR do módulo na planilha;
- g) a cada sessenta minutos contados a partir do início do teste, anotada a informação METAR do aeroporto na planilha;
- h) as anotações de ambos os METAR prosseguiram, até o término do teste que ocorreu impreterivelmente no intervalo onde foi anotada a informação do aeroporto;
- i) finalizado o teste, foi desmontado o equipamento e anotadas as observações diversas na planilha, quando aplicável.

Após a conclusão dos três testes, os dados obtidos foram comparados. Para os intervalos de sessenta minutos, deveria existir uma precisão na informação da seguinte forma:

- a) data e hora: não foram admitidas variações, a leitura da data e hora foram todas precisas;
- b) velocidade do vento: foi admitida uma variação máxima de cinco nós na velocidade, para mais ou para menos, em ambas as medidas constantes e rajadas. Esta margem foi estabelecida conforme entrevistas informais com instrutores do Aeroclube de Eldorado do Sul, os quais relataram que uma variação de cinco nós não compromete a avaliação do piloto em relação a análise do vento, salvo quando a velocidade indicada estiver

no valor limítrofe permitido para determinado avião. Nestes casos, o voo sequer é praticado devido a interdição das operações aeronáuticas, segundo estes instrutores;

c) pressão atmosférica: foi admitida uma variação máxima de um hectopascal, para mais ou para menos, conforme as especificações do *datasheet* da BOSCH (2008), as quais prevêm uma margem de erro desta magnitude;

d) temperatura do ar: foi admitida uma variação máxima de dois graus, para mais ou para menos, conforme as especificações do *datasheet* da BOSCH (2008), as quais prevêm uma margem de erro desta magnitude;

e) temperatura do ponto de orvalho: foi admitida uma variação máxima de três graus, para mais ou para menos. De acordo com AOSONG ELECTRONICS (2012), a leitura da umidade relativa do ar possui uma margem de erro que resulta em uma variação de até 1% para mais ou para menos. A temperatura do ponto de orvalho, que é derivada da leitura da temperatura do ar associada a umidade relativa, possui uma margem de erro no mínimo equivalente a margem de erro da temperatura do ar obtida, que é de dois graus Celsius para mais ou para menos. Foi adicionado um grau a esta margem tendo em vista a margem de 1% do sensor de umidade, visto que o valor fracionário resultante pode gerar margens maiores que dois graus Celsius, porém menores que três graus Celsius, para mais ou para menos;

f) cobertura de nuvens: foi efetuada uma comparação. Quando o aeroporto indicou presença de nuvens o módulo também deveria indicar esta informação, porém não na mesma precisão que o aeroporto fornece, apenas indicando a presença de nuvens e desconsiderando o tipo de nuvem. Não é possível comparar a metodologia utilizada no aeroporto com a que foi implementada neste projeto. Entretanto, a indicação de nuvens é informação que colabora na segurança de voo, caso esteja similar a informação gerada a equipamentos de maior porte e complexidade.

Estes requerimento atendem a demanda dos pequenos aeródromos e escolas de aviação, assim como são compatíveis com os sensores e algoritmos utilizados na implementação do módulo.

Também foi validada a transmissão dos dados, comparando as anotações dos períodos de validação com os dados que foram transmitidos. Os dados anotados deveriam ser idênticos aos registrados em todas as transmissões recebidas. Deveriam ser percebidas pelo menos 75% transmissões efetuadas com sucesso, salvo aquelas ocorridas devido a problemas na rede móvel utilizada.

9.2. CONSOLIDAÇÃO DOS DADOS COLETADOS

Os dados foram coletados conforme modelo de planilha de validação de acordo com o apêndice A, sendo que os resultados estão contidos nas planilhas existentes nos apêndices B, C e D.

Os resultados obtidos através das planilhas de validação evidenciam que o módulo desenvolvido atende aos parâmetros especificados pela estratégia de validação, sem ressalvas. Não foi observada nenhuma variação maior que a especificada, o módulo funcionou de forma ininterrupta sem quaisquer anomalia no seu funcionamento e transmitiu corretamente todos os METARs em intervalos de trinta minutos conforme especificado.

As condições climáticas durante os períodos de validação foram estáveis, sem grandes variações de vento e/ou temporais presentes. Uma validação mais abrangente poderia ser feita durante um período em que o clima estivesse bastante instável, viabilizando aferir o

comportamento dos sensores nesta condição. Entretanto, para os propósitos deste projeto e com os resultados obtidos, é possível afirmar que o módulo atende aos requisitos do projeto, propiciando dados localizados o formato necessário.

As validações também permitiram perceber aspectos adicionais, tais como a presença de nevoeiros. Este tipo de fenômeno não é percebido pelo módulo, salvo se o nevoeiro for muito intenso a ponto de afetar a luminosidade prevista para o horário em que estiver ocorrendo. Isto ocorre pois os nevoeiros de baixa intensidade funcionam como um elemento que dispersa a luz do sol, irradiando os feixes solares tornando a luminosidade percebida pelo sensor maior do que a luz do sol atingindo o mesmo sensor de forma oblíqua, por exemplo. A implementação da detecção deste tipo de fenômeno gera a necessidade de pesquisas adicionais, envolvendo a utilização de sensores adicionais. Os equipamentos oficiais da REDEMET utilizam sensores baseados em laser, com custo aproximado de 19.500,00 dólares australianos⁷, para detectar tanto o teto de nuvens quanto a presença de nevoeiros (COMANDO DA AERONÁUTICA, 2011).

9.3.CUSTOS

Um dos diferenciais deste projeto é o custo baixo para a construção do módulo de coleta. A listagem a seguir compreende todos os componentes adquiridos para este projeto e o respectivo custo total por item:

- a) 01 *proto-board* para prototipação Hikari HK-P200: R\$ 68,00;
- b) 01 Arduino Uno R3: R\$ 68,00;
- c) 02 metros de fio para *proto-board*: R\$ 3,00;
- d) 05 botões tipo *push-button*: R\$ 3,00;
- e) 03 resistores de 10K Ω : R\$ 0,50;
- f) 01 Bosch BMP085 : R\$ 58,00;
- g) 01 Aosong AM2302: R\$ 36,00;
- h) 01 LCD Redlion LMC-SS4A20 : R\$ 40,00;
- i) 01 BH1750FVI : R\$ 25,00;
- j) 01 LCD1602 : R\$ 8,00;
- k) 01 RTC DS1307 : R\$ 16,00;
- l) 01 3.3V / 5V Power Supply : R\$ 7,00;
- m) 01 anemômetro : R\$ 230,00;
- n) 01 módulo GSM EF COM PRO : R\$ 315,00.

As ferramentas necessárias para a montagem do protótipo não foram consideradas como custo com o projeto, tais como ferro de soldar, solda, alicates e demais ferramentas. Os preços são aqueles praticados na data da compra dos componentes. O custo total para o protótipo atual é de R\$ 877,50.

Como comparativo, um anemômetro ultrassônico para uso profissional, fornecido pela empresa Intercâmbio Eletro Mecânico de Porto Alegre, foi cotado entre R\$ 11.484,00 e R\$ 23.837,40 conforme anexo A. O custo com uma estação que forneça uma METAR e que seja homologada oscila entre US\$ 75.000,00 (setenta e cinco mil dólares) (AIRPORT IMPROVEMENT, 2011) e US\$ 300.000,00 (trezentos mil dólares) (GENTE DE OPINIÃO, 2008).

⁷ Valor obtido em http://www.defence.gov.au/teamaustralia/8200-CHS_ceilometer.htm. Acesso em 29 jun. 2014.

Ficou constatado que o módulo apresenta um custo muito baixo em relação aos equipamentos existentes no mercado. Apesar de não possuir a qualidade, certificações e homologações que os equipamentos profissionais possuem, a validação evidenciou que o módulo fornece dados corretos e com pouca margem de erro.

15. CONSIDERAÇÕES

O propósito deste projeto foi atendido, através da construção e implementação do módulo de coleta e os respectivos sensores e componentes. Os requisitos foram atendidos através da validação, que comprovou a eficácia do funcionamento do módulo e evidenciou que o mesmo está pronto para se tornar um produto. Como melhorias, podemos destacar alguns aspectos que podem contribuir para um produto melhor, tanto quanto a interatividade com o usuário com também em relação a parte técnica e a segurança de voo que o módulo proporciona.

O nevoeiro, que é um fenômeno de restrição visual, é fator relevante para a decisão do piloto em relação a segurança de voo. Este tipo de fenômeno não foi implementado nesta versão do módulo, mas pode ser considerado em versões futuras. Para sua detecção serão necessários sensores adicionais e pesquisa dos sensores existentes e do fenômeno.

Também pode ser utilizando um *display* gráfico, colorido e com maior resolução, o que permitirá uma melhor visualização pelo usuário. Este *display* também pode conter resposta ao toque na tela, eliminando os botões existentes atualmente e melhorando significativamente a interação com o usuário.

Inicialmente planejado, a leitura da direção do vento foi descartada devido a ausência de sensor de baixo custo no mercado. Foi considerada uma solução artesanal, a qual foi desconsiderada devido a criticidade do projeto em relação a componentes que não tenham sido previamente testados e homologados. Um aprimoramento quanto a leitura dos dados do vento, assim como a implementação da leitura da direção do vento, poderiam ser efetivada através do uso de um anemômetro baseado em ultrassom, que possui melhor resposta a rajadas de vento e é muito mais sensível que o anemômetro utilizado neste projeto.

As melhorias sugeridas até aqui, bem como as que seguem a seguir, ocasionam a necessidade de um poder computacional maior. O uso de um sistema com maior poder de processamento como Arduino Mega⁸, Raspberry Pi⁹ ou Beagle Bone Black¹⁰, permitiria que o módulo absorvesse todas as funcionalidades adicionais propostas, com um custo relativamente baixo.

O módulo poderia possuir outras formas de transmissão dos dados, funcionando como contingência inclusive. A implementação de um componente *ethernet* permitira ao módulo enviar os dados através de rede local, que muitos aeródromos possuem atualmente. Como o fator distância é presente nestes locais, onde o módulo certamente precisaria estar posicionado longe da área administrativa, onde usualmente encontramos a rede local se existente, também pode ser considerado o uso de componentes *wifi*, viabilizando a conexão através de rede sem fio.

Para a sincronização da data e hora, poderia ser utilizado o protocolo *Network Time Protocol* (NTP), que permite o sincronismo destes parâmetros entre diversos elementos distribuídos. Para esta funcionalidade, poderia ser utilizado um servidor próprio ao invés do WUNDERGROUND, o qual poderia possuir funcionalidades adicionais, tais como:

- a) centralização da comunicação;
- b) controle de erros reportados pelos módulos, acionando suporte;

⁸ <http://arduino.cc/en/Main/arduinoBoardMega>

⁹ <http://www.raspberrypi.org/product/model-b/>

¹⁰ <http://beagleboard.org/Products/BeagleBone+Black>

- c) configuração do módulo através de interface *web*;
- d) consulta do METAR através de API, viabilizando a implementação de aplicativos móveis;
- e) histórico de METAR;
- f) consulta de METAR de estações interligadas;
- g) algoritmo de análise meteorológica.

Também pode ser inserido um componente para uso de cartão de memória no módulo, de forma que ele armazene um histórico de METAR, viabilizando ao usuário a consulta a este histórico.

REFERÊNCIAS

- AGÊNCIA NACIONAL DE AVIAÇÃO CIVIL. **ANAC RBAC nº 61 EMENDA nº 01**: 2013: Licenças, habilitações e certificados para pilotos. Brasília, 2013.
- AGÊNCIA NACIONAL DE AVIAÇÃO CIVIL. Relação de aeródromos privados. 2013. Disponível em: <http://www2.anac.gov.br/arquivos/pdf/aerodromos/aerodromosPrivados.xls>. Acesso em 03 set. 2013.
- AIRPORT IMPROVEMENT. *Aviation magazine*. Setembro de 2011. Disponível em: <http://www.airportimprovement.com/content/story.php?article=00299>>. Acesso em 21 jun. 2014.
- AOSONG ELECTRONICS. **AM2302 Product Manual**. China, 2012.
- ARDUINO. Site oficial. 2013. Disponível em: <http://www.arduino.cc>>. Acesso em: 03 set. 2013.
- ASSIREU, Arcilan Trevenzoli et al. **Aplicação do Operador de Fragmentação Assimétrica (FA) para estimativas da rugosidade superficial a partir de mapas de elevação**. 2013. Disponível em: <http://www.dsr.inpe.br/sbsr2013/files/p1198.pdf>>. Acesso em: 31 ago. 2013.
- BOEING COMMERCIAL AIRPLANES. Aviation Safety. **Relatório**. Statistical Summary of Commercial Jet Airplanes Accidents - Worldwide Operations 1959 - 2011. Washington, 2012.
- BOSCH. **Datasheet: BST-BMP085-DS000-03**. Estados Unidos, 2008.
- CARR, Mahil. VERNER, June. **Prototyping and Software Development Approaches**. 2007. Disponível em: http://http://rguerrero334.blogspot.es/img/Prototyping_and_Software_Development_Approaches.pdf>. Acesso em 02 set. 2013.
- CELENZA, John J.; CARVER, R. W. A community supported, Internet-based surface meteorological network, the Wunderground experience. In: 25th Conference on International Interactive Information and Processing Systems. 2009.
- COMANDO DA AERONÁUTICA. **MCA 58-3**: 2004: Manual do curso piloto privado - avião. Brasília, 2004.
- COMANDO DA AERONÁUTICA. **MCA 101-1**: 2011: Instalações de estações meteorológicas de superfície e de altitude. Brasília, 2011.
- DAHLBERG, J. A.; PEDERSEN, T. F.; BUSCHE, Peter. **ACCUWIND - Methods for Classification of Cup Anemometers**. Riso National Laboratory. Dinamarca, 2006.

DEPARTAMENTO DE CONTROLE DO ESPAÇO AÉREO. **Meteorologia Aeronáutica**. Rio de Janeiro, 2012. Disponível em: <<http://www.decea.gov.br/espaco-aereo/meteorologia-aeronautica/>>. Acesso em: 31 ago. 2013.

ELVEGARD, E.; SJOSTEDT, G. **The Calculation of Illumination from Sun and Sky**. Illuminating Engineering Society. New York, 1940.

GENTE DE OPINIÃO. Site oficial. Porto Velho, 2008. Disponível em: <<http://www.gentedeopinioao.com.br/lerConteudo.php?news=31084>>. Acesso em: 30 mai. 2014.

INTERNATIONAL TELECOMMUNICATION UNION. **V.250**. Series V: Data Communication Over The Telephone Network - Control Procedures. Suíça, 2003. Disponível em <http://www.itu.int/rec/dologin_pub.asp?lang=e&id=T-REC-V.250-200307-I!!PDF-E&type=items>. Acesso em: 21 mai. 2014.

LAWRENCE, Mark G.; **The Relationship between Relative Humidity and the Dewpoint Temperatur in Moist Air: A Simple Conversion and Applications**. American Meteorological Society. Boston, 2005.

MATSCHINSKE, Martim Roberto. FREITAS, José Carlos de. **Windshear: Versão 2008. Departamento de Controle do Espaço Aéreo**. Rio de Janeiro, 2008.

MAXIM INTEGRATED. **DS1307 Datasheet**. San Jose, 2008.

MOYSÉS, Gerson Luís Russo; MOORI, Roberto Giro. **Coleta de dados para a pesquisa acadêmica: Um estudo sobre a elaboração, a validação e a aplicação eletrônica de questionário**. XVII Encontro Nacional de Engenharia de Produção. 2007. Disponível em <http://www.unisc.br/portal/upload/com_arquivo/coleta_de_dados_para_a_pesquisa_academica.pdf>. Acesso em 08 set. 2013.

NXP SEMICONDUCTORS. **UM10204 - I2C bus specification and user manual**. Revisão 5. Estados Unidos, 2012.

PINTO, Lili Souza; PINTO, José Geraldo Souza. **Piloto de Jato**. Rio de Janeiro: Ao Livro Técnico, 2003.

ROHM COMPANY. **BH1750FVI Technical Note**. No. 11046EDT01. Japão, 2011.

SAINTIVE, Newton Soler. **Teoria de Voo: Introdução à aerodinâmica**. Rio de Janeiro: ASA, 2010.

SHCWABER, Ken. SUTHERLAND, Jeff. **Um guia definitivo ara o Scrum: As regras do jogo**. 2011. Disponível em: <<https://www.scrum.org/Scrum-Guides>>. Acesso em 02 set. 2013.

SHENZHEN EONE ELECTRONICS. **1602A-1 LCD Module Specification**. Versão 1.2. China, 2013.

SIMCOM WIRELESS SOLUTIONS LTD. **SIM900 Hardware Design**. Version 1.01. Estados Unidos, 2009.

SONNEMAKER, João Baptista. **Meteorologia**. 30ª Edição. São Paulo: ASA, 2009.

SPARKFUN ELECTRONICS. Site Oficial. 2013. Disponível em: <<https://www.sparkfun.com>>. Acesso em: 13 nov. 2013.

WORLD METEOROLOGICAL ORGANIZATION. **WMO No. 306 - Manual on Codes**. Volume I.1 (Annex II on WMO Technical Regulations). Part A - Alphanumeric Codes. 1995 Edition. Geneva, 1995.

WUNDERGROUND. Site oficial. 2014. Disponível em: <<http://www.wunderground.com>>. Acesso em: 03 jan. 2014.

COMPONENTES REUTILIZADOS

Time Library¹¹

Biblioteca para o Arduino que implementa um relógio, capaz de armazenar data e hora com precisão de segundos, sem a necessidade de componente externo. Adicionalmente, ela também permite recuperar a hora e data atual com base em servidores *Network Time Protocol*.

BMP085DRIVER¹²

Biblioteca para o Arduino que implementa a comunicação com o sensor BMP085.

DHTLIB¹³

Biblioteca para o Arduino que implementa a comunicação com o sensor padrão DHT22.

Wire Library¹⁴

Biblioteca para o Arduino que viabiliza a comunicação com dispositivos no padrão I2C, sendo necessária para a comunicação com o sensor de pressão atmosférica e temperatura do ar e o sensor de luminosidade.

LiquidCrystal Library¹⁵

Biblioteca para o Arduino que padroniza o acesso a *displays* LCD, permitindo o uso facilitado deste tipo de recurso de forma configurável, suportando uma ampla gama de dispositivos baseados no padrão Hitachi e também flexibilizando este componente, permitindo a troca por outro semelhante sem mudanças de impacto na implementação do módulo.

Metro Library¹⁶

Biblioteca que viabiliza o controle de estados de elementos sob forma temporizada, permitindo o controle de eventos de forma individualizada.

¹¹ <http://playground.arduino.cc/Code/time>

¹² <https://code.google.com/p/bmp085driver/>

¹³ <http://playground.arduino.cc/Main/DHTLib>

¹⁴ <http://www.arduino.cc/en/Reference/Wire>

¹⁵ <http://arduino.cc/en/Reference/LiquidCrystal>

¹⁶ <http://playground.arduino.cc/code/metro>

APÊNDICE A - MODELO DE PLANILHA DE VALIDAÇÃO

VOO SEGURO - PLANILHA DE PERÍODO DE VALIDAÇÃO

Executor : (nome da pessoa que efetuou o período de validação)
 Período Nº : (número do período de validação)
 Data : (data da execução do período de validação)
 Hora Inicial : (hora do início do período de validação)
 Hora Término : (hora do término do período de validação)

Tempo	METAR Módulo	METAR SBPA
00:00		
00:10		
00:20		
00:30		
00:40		
00:50		
01:00		
01:10		
01:20		
01:30		
01:40		
01:50		
02:00		
02:10		
02:20		
02:30		
02:40		
02:50		
03:00		
03:10		
03:20		
03:30		
03:40		
03:50		
04:00		

Observações: (anotações diversas pertinentes aos dados obtidos durante o período)

APÊNDICE B - PLANILHA DE VALIDAÇÃO: PERÍODO 1

VOO SEGURO - PLANILHA DE PERÍODO DE VALIDAÇÃO

Executor : Alexandre Lehmann Holzhey Período Nº : 01 de 03
 Data : 25/05/2014 Hora Inicial : 12:00 UTC Hora Término : 16:00 UTC

Tempo	METAR Módulo	METAR SBPA
00:00	METAR SBPA AUTO 251200Z 02KT 10/10 Q1020 RMK A01	METAR SBPA 251200Z 07004KT 0800 R11/0800 R29///// FG SCT002 09/09 Q1020
00:10	METAR SBPA AUTO 251210Z 01KT 10/10 Q1020 RMK A01	
00:20	METAR SBPA AUTO 251220Z 02KT 11/11 Q1020 RMK A01	
00:30	METAR SBPA AUTO 251230Z 00000KT 10/10 Q1020 RMK A01	
00:40	METAR SBPA AUTO 251240Z 02KT 11/10 Q1020 RMK A01	
00:50	METAR SBPA AUTO 251250Z 02KT 11/10 Q1020 RMK A01	
01:00	METAR SBPA AUTO 251300Z 02KT 11/10 Q1020 RMK A01	METAR SBPA 251300Z 04003KT 1800 R29///// R11/1800 BR BKN004 12/10 Q1021
01:10	METAR SBPA AUTO 251310Z 02KT 11/09 Q1020 RMK A01	
01:20	METAR SBPA AUTO 251320Z 01KT 11/10 Q1021 RMK A01	
01:30	METAR SBPA AUTO 251330Z 01KT 12/09 Q1021 RMK A01	
01:40	METAR SBPA AUTO 251340Z 01KT 12/09 Q1021 RMK A01	
01:50	METAR SBPA AUTO 251350Z 01KT 12/09 Q1021 RMK A01	
02:00	METAR SBPA AUTO 251400Z 02KT 12/10 Q1021 RMK A01	METAR SBPA 251400Z 26002KT 3000 HZ BKN004 SCT021 14/10 Q1021
02:10	METAR SBPA AUTO 251410Z 03KT 13/10 Q1021 RMK A01	
02:20	METAR SBPA AUTO 251420Z 02KT 13/09 Q1021 RMK A01	
02:30	METAR SBPA AUTO 251430Z 01KT 13/09 Q1021 RMK A01	
02:40	METAR SBPA AUTO 251440Z 02KT 14/10 Q1020 RMK A01	
02:50	METAR SBPA AUTO 251450Z 00000KT 14/10 Q1020 RMK A01	

Tempo	METAR Módulo	METAR SBPA
03:00	METAR SBPA AUTO 251500Z 02KT 14/10 Q1020 RMK A01	METAR SBPA 251500Z 27002KT 9999 FEW004 BKN025 15/09 Q1020
03:10	METAR SBPA AUTO 251510Z 02KT 14/09 Q1020 RMK A01	
03:20	METAR SBPA AUTO 251520Z 03KT 14/09 Q1020 RMK A01	
03:30	METAR SBPA AUTO 251530Z 02KT 14/10 Q1020 RMK A01	
03:40	METAR SBPA AUTO 251540Z 03KT 15/09 Q1020 RMK A01	
03:50	METAR SBPA AUTO 251550Z 02KT 15/09 Q1020 RMK A01	
04:00	METAR SBPA AUTO 251600Z 03KT 15/10 Q1020 RMK A01	METAR SBPA 251600Z 09004KT 9999 FEW030 15/09 Q1019

Observações: condições meteorológicas conferiram clima com restrição visual através de nevoeiro e chuvisco, sendo que o módulo não acusou presença de nuvens. Isto é correto, mas é válido considerar este tipo de restrição como melhoria, visto que o nevoeiro é fator importante na avaliação para pouso e/ou decolagem, ainda que seja complexa a implementação de sensor que perceba este tipo de fenômeno. Demais parâmetros apresentaram vento fraco e estável, temperatura baixa e em crescimento, sendo que a temperatura de ponto de orvalho estava igual a temperatura do ar no início, diminuindo em seguida conforme a visibilidade foi aumentando.

APÊNDICE C - PLANILHA DE VALIDAÇÃO: PERÍODO 2

VOO SEGURO - PLANILHA DE PERÍODO DE VALIDAÇÃO

Executor : Alexandre Lehmann Holzhey Período Nº : 02 de 03
 Data : 29/05/2014 Hora Inicial : 16:00 UTC Hora Término : 20:00 UTC

Tempo	METAR Módulo	METAR SBPA
00:00	METAR SBPA AUTO 291600Z 02KT 21/14 Q1021 RMK A01	METAR SBPA 291600Z 07004KT 9999 FEW025 20/13 Q1021
00:10	METAR SBPA AUTO 291610Z 02KT 21/14 Q1021 RMK A01	
00:20	METAR SBPA AUTO 291620Z 05KT 21/13 Q1020 RMK A01	
00:30	METAR SBPA AUTO 291630Z 06KT 21/13 Q1020 RMK A01	
00:40	METAR SBPA AUTO 291640Z 06KT 21/13 Q1019 RMK A01	
00:50	METAR SBPA AUTO 291650Z 08KT 21/13 Q1019 RMK A01	
01:00	METAR SBPA AUTO 291700Z 10KT 21/14 Q1019 RMK A01	METAR SBPA 291700Z 11011KT CAVOK 21/13 Q1019
01:10	METAR SBPA AUTO 291710Z 09KT 21/14 Q1019 RMK A01	
01:20	METAR SBPA AUTO 291720Z 10KT 21/13 Q1019 RMK A01	
01:30	METAR SBPA AUTO 291730Z 10KT 21/14 Q1019 RMK A01	
01:40	METAR SBPA AUTO 291740Z 12KT 21/14 Q1019 RMK A01	
01:50	METAR SBPA AUTO 291750Z 09KT 21/13 Q1019 RMK A01	
02:00	METAR SBPA AUTO 291800Z 08KT 21/13 Q1019 RMK A01	METAR SBPA 291800Z 11009KT CAVOK 22/13 Q1019
02:10	METAR SBPA AUTO 291810Z 09KT 21/13 Q1019 RMK A01	
02:20	METAR SBPA AUTO 291820Z 12KT 21/14 Q1019 RMK A01	
02:30	METAR SBPA AUTO 291830Z 12KT 21/14 Q1019 RMK A01	
02:40	METAR SBPA AUTO 291840Z 08KT 21/13 Q1019 RMK A01	
02:50	METAR SBPA AUTO 291850Z 08KT 21/14 Q1019 RMK A01	

Tempo	METAR Módulo	METAR SBPA
03:00	METAR SBPA AUTO 291900Z 10KT 21/14 Q1019 RMK A01	METAR SBPA 291900Z 12010KT CAVOK 22/13 Q1019
03:10	METAR SBPA AUTO 291910Z 09KT 21/13 Q1019 RMK A01	
03:20	METAR SBPA AUTO 291920Z 10KT 21/13 Q1019 RMK A01	
03:30	METAR SBPA AUTO 291930Z 09KT 21/13 Q1019 RMK A01	
03:40	METAR SBPA AUTO 291940Z 08KT 21/14 Q1019 RMK A01	
03:50	METAR SBPA AUTO 291950Z 08KT 21/13 Q1019 RMK A01	
04:00	METAR SBPA AUTO 292000Z 08KT 21/12 Q1019 RMK A01	METAR SBPA 292000Z 10010KT CAVOK 20/13 Q1019

Observações: Condições climáticas favoráveis, com céu aberto, temperatura amena e vento de fraco a moderado em alguns momentos, sem rajadas. Módulo não indicou presença de nuvens na primeira amostra, o que foi considerado pois no momento de início dos testes não haviam mais nuvens (informação corroborada pelo METAR da hora seguinte), sendo que o módulo não teve tempo para mensurar e estimar a cobertura pois não estava acionado nos dez minutos antecedentes.

APÊNDICE D - PLANILHA DE VALIDAÇÃO: PERÍODO 3

VOO SEGURO - PLANILHA DE PERÍODO DE VALIDAÇÃO

Executor : Alexandre Lehmann Holzhey Período Nº : 03 de 03
 Data : 07/06/2014 Hora Inicial : 08:00 UTC Hora Término : 12:00 UTC

Tempo	METAR Módulo	METAR SBPA
00:00	METAR SBPA AUTO 070800Z 10KT 17/15 Q1008 RMK A01	METAR SBPA 070800Z 11011KT 9999 SCT027 OVC080 17/16 Q1008
00:10	METAR SBPA AUTO 070810Z 11KT 17/15 Q1008 RMK A01	
00:20	METAR SBPA AUTO 070820Z 11KT 18/15 Q1008 RMK A01	
00:30	METAR SBPA AUTO 070830Z 12KT 17/15 Q1008 RMK A01	
00:40	METAR SBPA AUTO 070840Z 11KT 17/15 Q1008 RMK A01	
00:50	METAR SBPA AUTO 070850Z 11KT 17/15 Q1008 RMK A01	
01:00	METAR SBPA AUTO 070900Z 11KT 17/15 Q1008 RMK A01	METAR SBPA 070900Z 11011KT 9999 SCT027 BKN080 17/16 Q1008
01:10	METAR SBPA AUTO 070910Z 11KT 17/15 Q1008 RMK A01	
01:20	METAR SBPA AUTO 070920Z 12KT 18/15 Q1008 RMK A01	
01:30	METAR SBPA AUTO 070930Z 12KT 18/16 Q1008 RMK A01	
01:40	METAR SBPA AUTO 070940Z 14KT 18/16 Q1008 RMK A01	
01:50	METAR SBPA AUTO 070950Z 13KT 18/16 Q1008 RMK A01	
02:00	METAR SBPA AUTO 071000Z 14KT 18/16 Q1008 RMK A01	METAR SBPA 071000Z 11015KT 8000 SCT010 BKN060 18/16 Q1008
02:10	METAR SBPA AUTO 071010Z 13KT 18/16 Q1008 RMK A01	
02:20	METAR SBPA AUTO 071020Z 14KT 18/17 OVC Q1008 RMK A01	
02:30	METAR SBPA AUTO 071030Z 15KT 18/17 OVC Q1008 RMK A01	
02:40	METAR SBPA AUTO 071040Z 16KT 18/17 OVC Q1008 RMK A01	
02:50	METAR SBPA AUTO 071050Z 14KT 18/17 OVC Q1008 RMK A01	

Tempo	METAR Módulo	METAR SBPA
03:00	METAR SBPA AUTO 071100Z 14KT 18/17 OVC Q1008 RMK A01	METAR SBPA 071100Z 11015KT 8000 BKN010 BKN050 18/17 Q1007 WS R11
03:10	METAR SBPA AUTO 071110Z 16KT 19/17 OVC Q1008 RMK A01	
03:20	METAR SBPA AUTO 071120Z 14KT 19/18 OVC Q1008 RMK A01	
03:30	METAR SBPA AUTO 071130Z 15KT 19/18 OVC Q1008 RMK A01	
03:40	METAR SBPA AUTO 071140Z 17KT 19/17 OVC Q1008 RMK A01	
03:50	METAR SBPA AUTO 071150Z 15KT 19/18 OVC Q1008 RMK A01	
04:00	METAR SBPA AUTO 071200Z 16KT 19/19 OVC Q1008 RMK A01	METAR SBPA 071200Z 11017KT 8000 BKN010 BKN052 19/18 Q1007 WS R11

Observações: período efetivado em condições climáticas com existência de nuvens e iniciado em período noturno. Indicações da presença de nuvens pelo módulo condizentes com sua implementação, indicando a presença destas após o horário do nascer do sol. Cobertura de nuvens indicada pelo módulo foi de tempo encoberto, enquanto o aeroporto indicava tempo inicialmente encoberto e nublado e posteriormente somente nublado, porém em duas camadas. Indicação do módulo foi dentro dos parâmetros de validação, sendo que para ser melhor é necessário uma evolução dos sensores utilizados, conforme exposto na seção de melhorias deste documento. Aeroporto Internacional Salgado Filho também acusou *wind shear*, a qual não foi reportada pelo módulo estando também de acordo com a especificação.

ANEXO A - ORÇAMENTOS DE ANEMÔMETROS

Resposta de e-Mail enviado em 17/10/2013 para Luciana Oliveira <lucianaoliveira@hobeco.net>, solicitando cotação de anemômetro.

Prezado Alexandre,

Agradecemos o seu contato e a oportunidade de contribuir com o seu projeto.

O sensor WS425 está custando aproximadamente R\$ 27.500,00 e nosso prazo de entrega é de 120 dias.

Este prazo é longo, pois não trabalhamos com estoque e inclui algum eventual problema de operação padrão da Receita Federal no momento de desembaraço do equipamento.

Possuímos um outro sensor modelo WMT52 que mais barato que o WS425. Este sensor está aproximadamente R\$ 11.700,00.

Estou te enviando o folheto técnico de ambos os sensores para que você possa compará-los.

Caso seja necessário a cotação formal do equipamento, me avise.

Atenciosamente,

Luciana Oliveira

Luciana Oliveira
Meteorologista
Hobeco Sudamericana
Tel.: 21 2518-2237
Email: lucianaoliveira@hobeco.net
Site: www.hobeco.net

----- Original Message -----

From: [Alexandre Lehmann Holzhey](mailto:Alexandre.Lehmann.Holzhey)
To: lucianaoliveira@hobeco.net
Sent: Thursday, October 17, 2013 2:14 PM
Subject: Cotação

Luciana,

Estou pesquisando e cotando preços de anemômetros para uso em um projeto pessoal com Arduino. Necessito somente do sensor, não sendo necessário estação de leitura. Verifiquei que vocês possuem alguns modelos, inclusive com leitura ultrasônica. Gostaria de verificar com você a disponibilidade e preço deste sensor. O Sensor Ultra-sônico WS425, por exemplo, qual o preço dele para uma unidade e o tempo para entrega?

Grato,

Alexandre.

Resposta de e-Mail enviado em 30/10/2013 para Priscila Moraes <priscila@iem.com.br>, solicitando cotação de anemômetro.

	INTERCÂMBIO ELETRO MECÂNICO	Energias Renováveis: Medições Eólicas MEASNET Energia Solar Fotovoltaica. F: +55 (051) 3343-4455 / Fx: 3343-4230 E-mail: orcamentos@iem.com.br www.iem.com.br	AV. AMAZONAS, 800 SÃO GERALDO CEP 90240-541 PORTO ALEGRE Rio Grande do Sul, BRASIL
---	--	---	---

Oferta N°: **201310301418**

À:

Aos cuidados de: Alexandre Holzhey

saleiro@gmail.com

Porto Alegre, quarta-feira, 30 de outubro de 2013

Prezados senhores,

Abaixo apresentamos proposta referente ao fornecimento de equipamentos.

Desde já agradecemos o contato e permanecemos à disposição para quaisquer dúvidas e/ou esclarecimentos.

1. Itens Importados para Medição Eólica

Item	Cód.	Descrição do Produto	Valor Unit. (R\$)	Qnt.	IPI		Valor (R\$)	Valor (R\$)
					Aliq.	Valor (R\$)		
1	0	Anemômetro Ultrassônico Gill P6022NA 0	10.937,14	1	5 %	546,86		11.484,00
2	1242350	Anemômetro Ultrassônico 2D / S82100H NCM 90158090	14.754,86	1	5 %	737,74		15.492,60
3	34622	Anemômetro ultrassônico 3D / S83100H NCM 90158090	22.702,29	1	5 %	1.135,11		23.837,40

Itens importados - Posto Porto Alegre

0,00