

UNIVERSIDADE FEDERAL FLUMINENSE
ESCOLA DE ENGENHARIA
MESTRADO EM ENGENHARIA DE TELECOMUNICAÇÕES

Joacir de Oliveira Silva

**Controlador de Carga Microcontrolado
para Redes em Malha**

Niterói
2012

Joacir de Oliveira Silva

**Controlador de Carga Microcontrolado
para Redes em Malha**

Dissertação apresentada ao Curso de Pós-Graduação “Stricto Sensu” em Engenharia de Telecomunicações da Universidade Federal Fluminense, como requisito parcial para obtenção do Grau de Mestre. Área de Concentração: Sistemas de Telecomunicações.

Orientador: Prof. Luiz Cláudio Schara Magalhães, PhD

Niterói
2012

**Ficha Catalográfica elaborada pela Biblioteca da Escola de Engenharia e Instituto de
Computação da UFF**

S586 Silva, Joacir de Oliveira
Controlador de carga microcontrolado para redes em malha /
Joacir de Oliveira Silva. – Niterói, RJ : [s.n.], 2012
162 f.

Dissertação (Mestrado em Engenharia de Telecomunicações) -
Universidade Federal Fluminense, 2011.
Orientador: Luiz Cláudio Schara Magalhães.

1. Energia Solar. 2. Controlador de Carga. 3. Sistema de
comunicação. 4. Rede sem Fio. 5. Rede Mesh. I. Título.

CDD 621.47

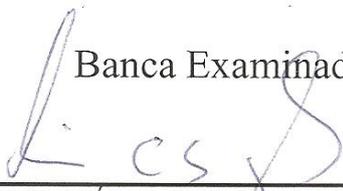
Joacir de Oliveira Silva

Controlador de Carga Microcontrolado para Redes em Malha

Dissertação apresentada ao Curso de Pós-Graduação “Stricto Sensu” em Engenharia de Telecomunicações da Universidade Federal Fluminense, como requisito parcial para obtenção do Grau de Mestre. Área de Concentração: Sistemas de Telecomunicações.

Aprovado em Junho de 2012

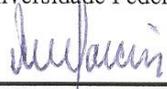
Banca Examinadora



Prof. Dr. LUIZ CLAUDIO SCHARA MAGALHÃES - Orientador
Universidade Federal Fluminense



Prof. Dr. DÉBORA CHRISTINA MUCHALUAT SAADE
Universidade Federal Fluminense



Prof. Dr. ANILTON SALLES GARCIA
Universidade Federal do Espírito Santo

Niterói
2012

‘O único homem que está isento de errar é aquele que não arrisca acertar.’

Albert Einstein

Agradecimentos

A minha esposa Silvia, a meus pais, a meus irmãos e amigos pelo apoio e tempo que lhes foi “roubado por culpa” desta dissertação.

A meu orientador, Schara, pela idealização deste trabalho e pelas valiosas contribuições.

Aos professores Célio, Débora, Schara, Carrano e Igor pelos investimentos no laboratório MidiaCom que resultam num ambiente cada vez melhor para se trabalhar.

A Diego Passos, Felipe Rolim e Rafael Ladislau pelas importantes contribuições no desenvolvimento do firmware do microcontrolador

A Rafael Valle, Alvaro Justem, Bruno Wanderley, Clayton, Marister e a todos do MidiaCom, que transformam esse espaço num ambiente extremamente agradável.

A Alan Pombo, Márcio e Luiz Carlos, colegas do Instituto de Física, pelas dicas sobre microcontroladores e na construção da placa do protótipo do controlador.

A empresa TBE (Transmissores Brasileiros de Energia) pela parceria com o Laboratório MidiaCom da Universidade Federal Fluminense que possibilitou a realização deste trabalho através do fomento à pesquisa.

RESUMO

Esta dissertação apresenta um controlador de carga inteligente para ser usado em sistemas autônomos de comunicação sem fio em malha, alimentados por energia solar fotovoltaica, denominado controlador de carga microcontrolado (CCM).

Os sistemas de controle de carga existentes atualmente têm um escopo de aplicação limitado já que na sua maioria não são programáveis. Para criar um “módulo de sobrevivência”, que não é mais do que um sistema que usa a energia disponível de modo inteligente, é apresentado nesta dissertação o desenvolvimento de um controlador de carga com as seguintes características: coleta de parâmetros elétricos do sistema fotovoltaico, atuação sobre o sistema para controlar os períodos em que o roteador fica ligado para aumentar a autonomia do sistema de baterias, acionamento de 2 cargas de forma independente e função *WatchDog Timer* que reinicia o roteador em caso de travamento.

Esta dissertação apresenta os principais componentes eletrônicos que compõem o protótipo, em especial o microcontrolador ATMEGA16. Também é apresentado o firmware desenvolvido para o microcontrolador.

Palavras chave: Energia solar, controlador de carga, Sistema de comunicação, Wireless, Redes em malha, *Mesh*.

ABSTRACT

This dissertation shows an intelligent charge controller to be used in autonomous systems for wireless communication network, powered by photovoltaic solar energy, called the charge controlled microcontroller (CCM).

The load control systems available today have a limited scope of application and most of them are not programmable. To create a "survival module", which is nothing more than a system that uses the energy available in an intelligent way, we have developed an intelligent charge controller with the following specs: collection of electrical parameters of the photovoltaic system, acting on the system to control the periods when router is connected and consequently, to increase the autonomy of the battery system, drive two loads independently and WatchDog Timer function to restart the router in case of locking.

This work presents the main electronic components that make up the prototype, especially the microcontroller ATMEGA16. Also is shown the developed firmware for the microcontroller.

Keywords: Solar energy, Charge Controller, Communication System, Wireless Mesh Networks.

LISTA DE ACRÔNIMOS

| | |
|--------|---|
| A/D | <i>Analógico/Digital</i> |
| AGM | <i>Absorbed Glass Mat</i> |
| AP | <i>Access Point</i> |
| CISC | <i>Complex Instruction Set Computer</i> |
| CMOS | <i>Complementary Metal-Oxide-Semiconductor</i> |
| CPU | <i>Central Processor Unit</i> |
| D/A | <i>Digital/Analógico</i> |
| DC | <i>Direct Current</i> |
| EEPROM | <i>Electric Erasable Programable ROM</i> |
| FV | <i>Fotovoltaico</i> |
| I/O | <i>Input/Output</i> |
| I2C | <i>Inter-Integrated Circuit</i> |
| JTAG | <i>Joint Test Action Group</i> |
| LCD | <i>Liquid Crystal Display</i> |
| MIPS | <i>Milhões de Instruções por Segundo</i> |
| PWM | <i>Pulse Width Modulation</i> |
| RAM | <i>Random Access Memory</i> |
| RF | <i>Rádio Frequência</i> |
| SPI | <i>Serial Peripheral Interface</i> |
| RAM | <i>Random Access Memory</i> |
| RISC | <i>Reduced Instruction Set Computer</i> |
| SRAM | <i>Static Random Access Memory</i> |
| TCP | <i>Transmission Control Protocol</i> |
| TCP/IP | <i>Transmission Control Protocol/Internet Protocol</i> |
| USART | <i>Universal Synchronous and Asynchronous serial Receiver and Transmitter</i> |
| UDP | <i>User Datagram Protocol</i> |
| UFF | <i>Universidade Federal Fluminense</i> |
| USB | <i>Universal Serial Bus</i> |
| VRLA | <i>Valve Regulated Lead Acid</i> |
| Wi-Fi | <i>Wireless Fidelity</i> |

LISTA DE SÍMBOLOS

| | |
|------------------|---|
| A | Ampère |
| Ah | Ampère-hora |
| a-Si | silício amorfo |
| C | Capacidade da bateria |
| C/20 | Capacidade da bateria em relação ao tempo de descarga de 20 horas |
| CdTe | <i>Cadmium Telluride</i> (Cádmio Telúrio) |
| CIGS | <i>Copper Indium Gallium Diselenide</i> |
| c-Si | silício mono-cristalino |
| H | Tempo de descarga relacionado com a especificação da bateria |
| I | Corrente de descarga |
| I_m | Corrente máxima |
| I_{mp} | Corrente de máxima potência |
| I_{sc} | Corrente de curto-circuito |
| J/Kg | Joule/Quilograma |
| mA | Miliampère |
| m-Si | Silício poli-cristalino (multi-cristalino) |
| μs | Microsegundo |
| mW | Miliwatt |
| P_{mp} | Potência no ponto de máxima potência |
| Rx | Recepção |
| Tx | Transmissão |
| V | Volt |
| Vdc | Voltagem de corrente direta |
| V_m | Voltagem máxima |
| V_{mp} | Voltagem de máxima de potência |
| V_{oc} | Voltagem de circuito aberto |
| V_p | Voltagem no ponto de máxima potência |
| W | Watt |
| W/Kg | Watt/Quilograma |
| W/m ² | Watt/metro quadrado |

ÍNDICE

| | |
|---|---------------|
| CAPÍTULO 1 – INTRODUÇÃO | 1 |
| 1.1 Motivação..... | 3 |
| 1.2 Objetivo..... | 4 |
| 1.3 Proposta..... | 5 |
| 1.4 Metodologia..... | 6 |
| 1.5 Principal Contribuição..... | 6 |
| 1.6 Organização do Texto..... | 6 |
| CAPÍTULO 2 - TRABALHOS RELACIONADOS | 8 |
| 2.1. Redes em malha (Mesh)..... | 8 |
| 2.2. Trabalhos relacionados..... | 10 |
| 2.2.1 Solar Wi-fi Grid..... | 10 |
| 2.2.2 Meraki Solar..... | 11 |
| 2.2.3 Solar Mesh..... | 11 |
| 2.2.4 SAM SOLAR (Sistema Autônomo em Malha Solar)..... | 12 |
| 2.2.5 Comparação entre as soluções pesquisadas..... | 13 |
| 2.3. Controladores de carga comerciais..... | 15 |
| 2.4. Controladores de carga comerciais..... | 16 |
| CAPÍTULO 3 - FUNDAMENTAÇÃO TEÓRICA | 19 |
| 3.1 Energia solar..... | 19 |
| 3.2 Sistemas Fotovoltaicos..... | 20 |
| 3.2.1 Sistemas Isolados..... | 20 |
| 3.2.2 Sistemas Híbridos..... | 21 |
| 3.2.3 Sistemas Interligados à rede..... | 21 |
| 3.3 Módulo Fotovoltaico..... | 21 |
| 3.3.1 Características elétricas dos módulos fotovoltaicos..... | 22 |
| 3.3.2 Fatores que afetam as características elétricas dos módulos..... | 24 |
| 3.3.3 Tipos de módulos fotovoltaicos..... | 25 |
| 3.4 Módulo de Armazenamento..... | 29 |
| 3.4.1 Tipos de Baterias..... | 30 |
| 3.4.2 Características ideais das baterias para uso em Sistemas Fotovoltaicos..... | 33 |
| 3.5 Módulo alimentado..... | 34 |
| 3.6 Módulo de controle de energia..... | 35 |

| | | |
|-------|--|----|
| 3.6.1 | Tipos de controladores de carga..... | 35 |
| 3.6.2 | Estratégias de Controle..... | 37 |
| 3.6.3 | Baterias de chumbo-ácido, NiCd e controladores em sistemas fotovoltaicos.. | 41 |
| 3.6.4 | Detalhamento das características e funções de um controlador de carga..... | 43 |
| 3.7 | Microcontroladores..... | 44 |
| 3.7.1 | Vantagens da utilização dos microcontroladores..... | 45 |
| 3.8 | Controladores de carga microcontrolados..... | 46 |
| 3.8.1 | Recursos dos controladores de carga microcontrolados..... | 46 |

CAPÍTULO 4 - PROTÓTIPO DO CONTROLADOR DE CARGA MICROCONTROLADO

| | | |
|-------|---|----|
| 4.1 | Requisitos e funcionalidades..... | 48 |
| 4.2 | Descrição geral do sistema..... | 48 |
| 4.3 | Diagrama Esquemático..... | 49 |
| 4.4 | Especificação do CCM..... | 50 |
| 4.5 | Descrição do Hardware do CCM..... | 51 |
| 4.5.1 | Bloco de alimentação..... | 52 |
| 4.5.2 | Bloco de controle..... | 53 |
| 4.5.3 | Bloco de Sensores..... | 63 |
| 4.5.4 | Bloco de Chaveamento..... | 65 |
| 4.5.5 | Bloco de Comunicação..... | 67 |
| 4.6 | Funcionalidades do CCM..... | 69 |
| 4.6.1 | Sensoriamento Remoto..... | 69 |
| 4.6.2 | Execução de Comandos Remotos..... | 70 |
| 4.6.3 | WatchDog Timer/reinicia roteador..... | 72 |
| 4.6.4 | Desligamento dos leds indicadores de status | 73 |
| 4.7 | Fluxograma..... | 74 |

CAPÍTULO 5 - TESTES e ANÁLISE DOS RESULTADOS

| | | |
|-------|---|----|
| 5.1 | Descrição do experimento..... | 75 |
| 5.2 | Sistema de coleta e armazenamento de dados..... | 78 |
| 5.3 | Testes realizados..... | 79 |
| 5.3.1 | Comunicação..... | 79 |
| 5.3.2 | Parâmetros ajustáveis..... | 82 |
| 5.3.3 | Subsistemas do controlador..... | 86 |
| 5.3.4 | Desativação dos Leds Indicadores..... | 92 |
| 5.3.5 | Auto-consumo..... | 92 |
| 5.4 | Saturação do sensor de luminosidade..... | 93 |
| 5.5 | Interface WEB de Monitoramento | 94 |
| 5.6 | Análise dos resultados..... | 96 |

| | |
|---|-----|
| CAPÍTULO 6 - CONCLUSÃO E TRABALHOS FUTUROS | 98 |
| 6.1 Conclusão..... | 98 |
| 6.2 Trabalhos Futuros..... | 99 |
| GLOSSÁRIO..... | 101 |
| REFERÊNCIAS BIBLIOGRÁFICAS..... | 103 |
| ANEXOS..... | 107 |

LISTA DE FIGURAS

| | |
|--|--|
| Figura 1.1 - Kit mesh desenvolvido para o projeto ReMoTe..... | |
| Figura 1.2 - Topologia da rede instalada pelo Projeto Remote em linha de transmissão..... | |
| Figura 1.3 - Sistema desenvolvido para a coleta dos dados..... | |
| Figura 2.1 - Modos de operação da rede sem fio 802.11..... | |
| Figura 2.2 - Rede em malha sem fio | |
| Figura 2.3 - Protótipo Wifi Grid Project (GREEN-WIFI, 2006)..... | |
| Figura 2.4 - Produto Meraki Solar (MERAKI, 2011)..... | |
| Figura 2.5 - Protótipo SolarMesh (SOLARMESH, 2002)..... | |
| Figura 2.6 - Topologia da rede | |
| Figura 2.7 - Controlador de carga desenvolvido na USP..... | |
| Figura 2.8 - Sistema de comunicação instalado na USP..... | |
| Figura 2.9 - Visualização da rede sem fio em malha solar – EPUSP..... | |
| Figura 3.1 - Sistema fotovoltaico isolado..... | |
| Figura 3.2 - Sistema fotovoltaico híbrido..... | |
| Figura 3.3 - Sistema fotovoltaico Interligados á Rede..... | |
| Figura 3.4 - Células em paralelo..... | |
| Figura 3.5 - Células em série..... | |
| Figura 3.6 - Curva característica I x V mostrando a corrente I_{sc} | |
| Figura 3.7 - A Curva típica de potência versus tensão..... | |
| Figura 3.8 - Parâmetros de potência máxima..... | |
| Figura 3.9 - Efeito causado pela variação de intensidade luminosa..... | |
| Figura 3.10 - Efeito causado pela temperatura na célula..... | |
| Figura 3.11 - Célula de silício monocristalino..... | |
| Figura 3.12 - Célula de silício policristalino..... | |
| Figura 3.13 - Célula de silício amorfo..... | |
| Figura 3.14 – Células com e sem sombras | |
| Figura 3.15 - Controlador série..... | |
| Figura 3.16 - Controlador Shunt..... | |
| Figura 3.17 - Controlador MPP..... | |
| Figura 3.18 - Tensão na bateria durante a estratégia de controle “ON-OFF” | |
| Figura 3.19 - Tensão na bateria durante a estratégia de controle “on-off” (detalhado)..... | |
| Figura 3.20 - Controle linear série: reostato..... | |
| Figura 3.21 - Controle linear paralelo: Diodo Zener..... | |
| Figura 3.22 – Estratégia de carga utilizando PWM..... | |
| Figura 3.23 - Pulsos PWM..... | |
| Figura 3.24 - Estados do carregamento a tensão constante por PWM..... | |

| | |
|---|----|
| Figura 3.25 – Display LCD..... | 47 |
| Figura 4.1 – Visão geral de uma rede mesh com sistema fotovoltaico Isolado..... | 49 |
| Figura 4.2 – Elementos do Sistema Fotovoltaico..... | 49 |
| Figura 4.3 - Diagrama esquemático do CCM..... | 50 |
| Figura 4.4 - Placa protótipo do CCM..... | 51 |
| Figura 4.5 - Circuito regulador de tensão | 52 |
| Figura 4.6 - Circuito de proteção reversão de polaridade..... | 52 |
| Figura 4.7 – Encapsulamentos do Atmega6..... | 55 |
| Figura 4.8 - PWM com período T e ciclo ativo de 0%, 25%, 50%, 75% e 100%..... | 59 |
| Figura 4.9 - Variação de um PWM em um intervalo de tempo..... | 59 |
| Figura 4.10 - Barramento I2C – TWI..... | 60 |
| Figura 4.11 - Fonte de Clock..... | 61 |
| Figura 4.12 - Circuito de RESET..... | 61 |
| Figura 4.13 - Relógio I2C DS1307..... | 62 |
| Figura 4.14 - Sensor de tensão do painel solar e bateria..... | 63 |
| Figura 4.15 – Sensor de corrente..... | 63 |
| Figura 4.16 - Sensor de temperatura..... | 64 |
| Figura 4.17 - Sensor de luminosidade (LDR)..... | 64 |
| Figura 4.18 - Chaveamento das cargas..... | 65 |
| Figura 4.19 - Chaveamento Painel Solar–Bateria..... | 66 |
| Figura 4.20 - Interface serial roteador LinkSys Wrt54G..... | 67 |
| Figura 4.21 - Conector da porta serial do CCM..... | 68 |
| Figura 4.22 - Interconexão entre roteador e CCM..... | 68 |
| Figura 4.23 - Detalhe da interface serial do Bullet2..... | 69 |
| Figura 4.24 - Leituras dos logs enviados pelo CCM..... | 70 |
| Figura 4.25 - Dados após o tratamento..... | 70 |
| Figura 4.26 - Chave para desligar Leds de sinalização..... | 73 |
| Figura 4.27 - Fluxograma do loop principal do algoritmo de carga..... | 74 |
| Figura 5.1 - Bancada montada para os testes | 75 |
| Figura 5.2 - Conversor serial RS232/TTL..... | 78 |
| Figura 5.3 - Ligação entre computador e CCM..... | 79 |
| Figura 5.4 - Diagrama do circuito montado para testes do sinal PWM..... | 79 |
| Figura 5.5 – Tela de monitoramento após o tratamento dos dados..... | 80 |
| Figura 5.6 – Circuito para os testes de monitoramento remoto..... | 80 |
| Figura 5.7 - Ativação da carga secundária..... | 81 |
| Figura 5.8 - Desligamento da carga secundária..... | 81 |
| Figura 5.9 - Mudança de modo de operação para Econômico..... | 81 |
| Figura 5.10 - Mudança de modo de operação para Padrão..... | 82 |
| Figura 5.11 – Diagrama esquemático do circuito de teste..... | 83 |
| Figura 5.12 – Detalhe da tensão $V_{mín}$ | 83 |

| | |
|--|----|
| Figura 5.13 – Detalhe da tensão de conexão da carga primária(V_C)..... | 84 |
| Figura 5.14 – Detalhe da tensão ($V_{m\acute{a}x}$)..... | 84 |
| Figura 5.15 – Diagrama do circuito montado para testes do sinal PWM..... | 86 |
| Figura 5.16 - PWM com 10% de Duty-Cycle..... | 86 |
| Figura 5.17 - PWM com 50% de Duty-Cycle..... | 87 |
| Figura 5.18 - PWM com 99% de Duty-Cycle..... | 87 |
| Figura 5.19 – Diagrama esquemático do circuito..... | 88 |
| Figura 5.20 – Curva de carga da bateria chumbo-ácido..... | 88 |
| Figura 5.21 – Curva de descarga da bateria chumbo-ácido..... | 89 |
| Figura 5.22 – Células LiFePo4 em série..... | 89 |
| Figura 5.23 – Detalhe do contador WDT..... | 91 |
| Figura 5.24 – Detalhe dos leds antes e depois da utilização da chave..... | 92 |
| Figura 5.25 - Curva de luminosidade LDR..... | 93 |
| Figura 5.26 - LDR's sem filtro e com filtro..... | 93 |
| Figura 5.27 - Curva luminosidade de um LDR com filtro..... | 94 |
| Figura 5.28 - Interface Web para visualização dos dados coletados..... | 95 |

LISTA DE TABELAS

| | |
|---|----|
| Tabela 2.1 - Comparação das soluções apresentadas..... | 14 |
| Tabela 2.2 - Resumo dos controladores de carga..... | 16 |
| Tabela 3.1 - Comparação entre células de silício..... | 27 |
| Tabela 3.2 – Redução da potencia em função da sombra..... | 28 |
| Tabela 3.3 – Estado de carga para bateria de 12 V..... | 31 |
| Tabela 3.4 – Consumo de pontos de acesso..... | 34 |
| Tabela 3.5 - Vantagens e desvantagens das estratégias de controle (IEA PVPS-1998 adaptado)..... | 41 |
| Tabela 3.6 – Características de um controlador de carga..... | 44 |
| Tabela 4.1 - Requisitos e funcionalidades do CCM..... | 48 |
| Tabela 4.2 - Especificações do CCM..... | 50 |
| Tabela 4.3 - Pontos de ajuste para baterias chumbo-ácido e LiFePO ₄ | 51 |
| Tabela 4.4 - Bytes x Operações remotas..... | 71 |
| Tabela 5.1 - Fonte de tensão/corrente ajustável..... | 76 |
| Tabela 5.2 - Bateria chumbo-ácido..... | 76 |
| Tabela 5.3 - Bateria LiFePO ₄ | 76 |
| Tabela 5.4 - Roteador LinkSys WRT54G..... | 77 |
| Tabela 5.5 - Osciloscópio analógico..... | 77 |
| Tabela 5.6 - Multímetro Digital..... | 77 |

CAPÍTULO 1 – INTRODUÇÃO

Este trabalho apresenta o desenvolvimento de um controlador de carga inteligente com características próprias para ser usado em nós de uma rede em malha alimentados por energia solar.

Os sistemas fotovoltaicos são uma alternativa para a geração de energia em locais onde a infraestrutura para provimento de energia elétrica não está instalada. Esses sistemas são compostos basicamente dos seguintes módulos: fotovoltaico (transformam luz em corrente elétrica), armazenamento (mantém o sistema quando a luz não está disponível), carga (dispositivo sendo alimentado) e controle (mantém a bateria com nível de carga dentro dos parâmetros seguros de operação).

Dos elementos que compõem o sistema fotovoltaico, o mais complexo é o controlador de carga, cuja principal função é controlar o fluxo de energia elétrica que entra e sai da bateria para evitar que esta seja sobrecarregada ou sofra descargas profundas que reduzem a sua vida útil e podem danificá-la de forma irremediável.

A maioria dos controladores de carga comerciais tem uma configuração fixa e apresentam limitações no seu uso. Como exemplo de configuração fixa, pode-se citar o modo de operação padrão. Neste modo, a carga é mantida permanentemente ligada até a tensão da bateria alcançar um limiar pré-determinado no projeto do controlador, quando ela é desligada. Isto não permite qualquer tipo de economia de energia, pois a carga fica ligada mesmo em períodos em que não está sendo utilizada.

Durante o desenvolvimento do projeto ReMoTe (Rede para Monitoramento de Linhas de Transmissão de Energia)[01], foi constatado que os controladores de carga existentes no mercado não eram capazes de atender aos requisitos das funcionalidades necessárias e baixo custo para tornar o sistema viável.

O projeto ReMoTe propõe a utilização de uma rede em malha sem fio para o controle e supervisão de linhas de transmissão de energia. Como as linhas de transmissão de energia são, em geral, instaladas em áreas carentes de infraestrutura de comunicação, a rede em malha disponibiliza um canal de comunicação que, entre outros serviços, permite oferecer conectividade a técnicos operadores que estiverem em campo executando reparos e manutenção, através de serviços de transmissão de dados, voz e vídeo; a transmissão de dados e vídeo de sensores instalados nas torres de transmissão para uma estação de controle; o uso de sensores, inclusive de robôs de inspeção, e sua conectividade a centrais de controle de forma a permitir a supervisão automatizada do sistema elétrico. Como exemplo dessa última

aplicação, câmeras de monitoramento patrimonial e de vegetação podem ser instaladas nas torres de transmissão enviando suas imagens a uma central de controle.

O Laboratório MídiaCom da Universidade Federal Fluminense (UFF) em parceria com a empresa TBE (Transmissores Brasileiros de Energia), tem trabalhado no desenvolvimento de uma solução para infraestrutura de comunicação para supervisão e controle de linhas de transmissão de energia utilizando redes em malha sem fio. Foi desenvolvida uma solução mesh de baixo custo para ser instalada no topo de torres de transmissão de energia.

Para prover conectividade ao longo de toda a linha de transmissão, foram instalados quarenta kits *mesh* alimentados por energia elétrica proveniente de um painel solar.

A figura 1.1 mostra um kit mesh sendo instalado na linha de transmissão que liga a Usina Hidrelétrica de Machadinho no Rio Grande do Sul e a Subestação de Campos Novos em Santa Catarina. Cada kit é composto de painel solar policristalino de 20W, 2 baterias de chumbo-ácido de 12V/ 7 Ah, roteador LinkSys modelo WRT54G e controlador de carga SunSaver-10.



Figura 1.1 - Kit mesh desenvolvido para o projeto ReMoTe

Uma rede experimental foi instalada na linha de transmissão que liga a subestação de Campos Novos (SC) à Usina Hidrelétrica de Machadinho (RS). Esta linha de transmissão possui mais de 40 km de extensão e são utilizados 41 pontos para conectar a rede de ponta a ponta. A Figura 1.2 ilustra a topologia da rede instalada.

A capacidade do sistema de baterias permite o funcionamento do nó *mesh* por até dois dias sem insolação. Para aumentar a autonomia, pode ser efetuado o redimensionamento do sistema trocando o painel por um modelo de maior potência e aumentando a capacidade de armazenamento, porém isto eleva de forma significativa os custos.



Figura 1.2 - Topologia da rede instalada pelo Projeto ReMoTe em linha de transmissão

No entanto, é possível aumentar a autonomia do sistema sem elevação significativa dos custos, através do desenvolvimento de técnicas de otimização que acrescentem inteligência ao sistema possibilitando economia de energia e minimizando o consumo do nó *mesh*.

Assim, este projeto foi desenvolvido como parte de um “módulo de sobrevivência” que pode configurar o sistema de acordo com as necessidades como por exemplo, aumentando a autonomia do sistema ou mantendo o sistema ligado conforme desejado.

1.1 MOTIVAÇÃO

Com a mudança do local da instalação da rede piloto do projeto ReMoTe, de Maranhão para Santa Catarina, que tem menos insolação e repetidos dias nublados, tornou-se clara a necessidade de um uso mais eficiente da energia armazenada, já que tanto os painéis e as baterias dimensionados não funcionaram a contento em Santa Catarina.

Além disso, foi detectado um problema de travamento de alguns roteadores e como o controlador de carga utilizado não possui nenhuma forma de sensoriamento do sistema, não foi possível identificar a causa do travamento.

Inicialmente, com a finalidade de medir com maior precisão e automatizar a coleta de alguns parâmetros elétricos do sistema fotovoltaico, foi desenvolvido um circuito eletrônico utilizando microcontrolador denominado Kit de Sensoriamento. Entre as medidas coletadas estão: a tensão na bateria, a corrente e a tensão no painel solar, a luminosidade e as temperaturas interna e externa à caixa hermética onde ficam armazenados roteador, bateria e controlador de carga. A figura 1.3 ilustra o sistema desenvolvido para a coleta dos dados do sistema de alimentação solar do projeto ReMoTe.

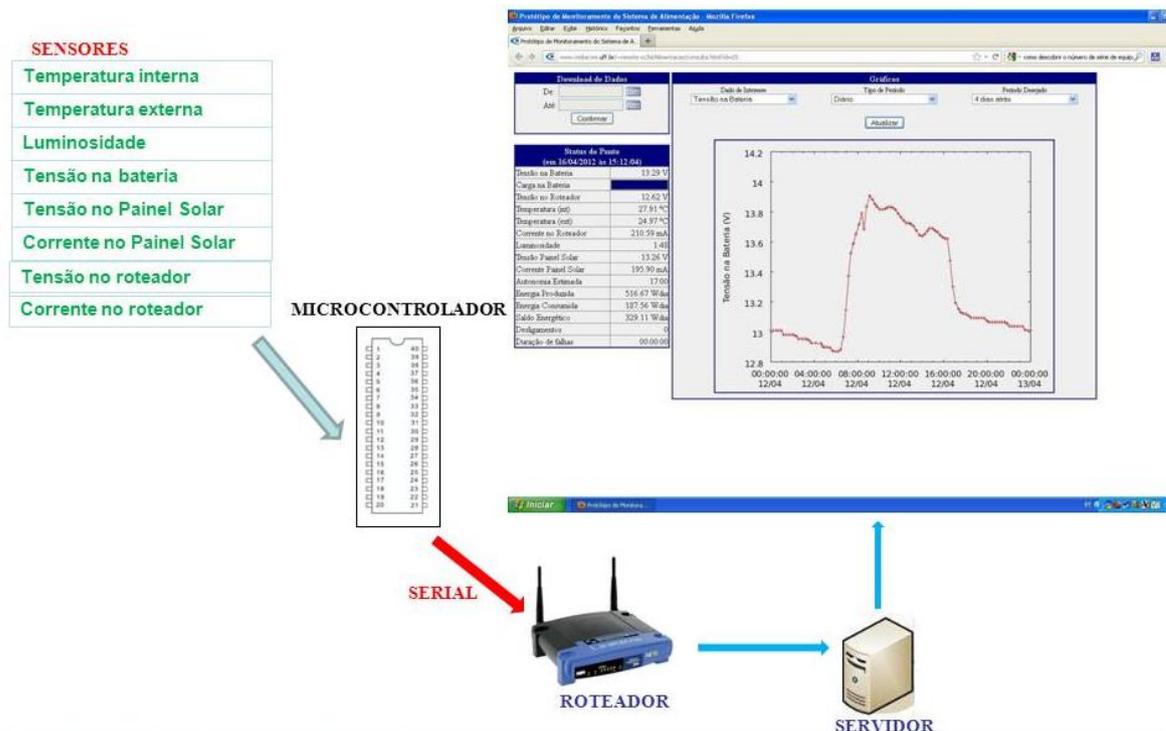


Figura 1.3 - Sistema desenvolvido para a coleta dos dados

As medidas coletadas são enviadas através da rede e armazenadas num computador central, permitindo a criação de uma série histórica que pode ser utilizada posteriormente para análise e exibição dos dados coletados. Por exemplo, a potência gerada pelo painel solar em um determinado dia, semana ou mês pode ser exibida através de um gráfico.

O controlador de carga microcontrolado surgiu como uma evolução natural do kit de sensoriamento interconectado ao roteador do nó *mesh*. Uma vez que existia um sistema que era capaz de medir os parâmetros elétricos do sistema fotovoltaico, alterá-lo para permitir seu uso como controlador de carga era o próximo passo lógico.

Nesta dissertação são descritas as funcionalidades adicionadas ao controlador de carga para, além de usar energia de forma inteligente, coletar informações e atuar sobre o sistema como por exemplo, desligando o roteador nas horas ociosas.

1.2 OBJETIVOS DA DISSERTAÇÃO

Este trabalho aborda os elementos que compõem as redes em malha alimentadas por energia solar, em especial os controladores de carga a fim de definir os requisitos e funcionalidades que estes devem possuir para o uso mais eficiente da energia do sistema.

Neste contexto, este trabalho tem por objetivo o desenvolvimento de um protótipo de controlador de carga inteligente (microcontrolado) com características próprias para ser usado

em nós de uma rede em malha alimentados por energia solar, denominado controlador de carga microcontrolado (CCM).

1.3 PROPOSTA

Além da função principal de um controlador de carga (controle do processo de carga e descarga da bateria para evitar danos à mesma), o CCM possui outras funcionalidades como coleta de informações (sensoriamento) e atuação sobre o sistema fotovoltaico. Além disso, possui um baixo custo comparado com outros controladores comerciais que apresentam algumas funcionalidades semelhantes.

O envio das medidas efetuadas pelos sensores do nó *mesh* para um servidor possibilita a coleta dos parâmetros elétricos do sistema fotovoltaico para ser visualizado em tempo real e também permite a análise posterior do desempenho do sistema.

A atuação sobre o sistema é efetuado através de comandos que são enviados pelo roteador. O CCM é programado para realizar 4 ações: desligar a carga primária por 5 minutos, ligar a carga secundária por 5 minutos, desligar a carga secundária antes dos 5 minutos e alternar os modos de operação programados do controlador. No modo padrão, o roteador é mantido ligado até a tensão na bateria atingir o valor mínimo definido no controlador, enquanto que no modo econômico o roteador é ligado apenas no período das 07:00 às 17:00, o que resulta numa considerável economia de energia. Outros modos de operação também podem ser definidos através da reprogramação do microcontrolador.

Devido aos dois circuitos de chaveamento independentes das cargas, é possível manter a carga primária (roteador) ligada enquanto a carga secundária (câmera IP, sensor ou outro dispositivo) pode ser ligada apenas quando necessário, para economizar energia.

Como os roteadores do ReMoTe ficam instalados em linha de transmissão, o seu acesso é bastante difícil. Se o roteador apresentar um estado não responsivo (travamento), será reiniciado automaticamente pelo controlador através da função *WatchDog Timer*. Este recurso se assemelha ao *WatchDog Timer* disponível em muitos microcontroladores e sistemas que exigem requisitos semelhantes de operação.

Devido aos pontos de ajustes selecionáveis, o CCM é programado para 2 tipos de baterias (Chumbo-ácido e LiFePO₄). Devido ao problema de descarga heterogênea das células de LiFePO₄ (como será visto na seção de testes), atualmente apenas são utilizadas as baterias de chumbo-ácido.

A chave de ativação dos 3 leds indicadores de status do controlador permite que os leds sejam ligados apenas quando houver necessidade, economizando energia.

Finalmente, este trabalho aborda os aspectos da implementação do hardware, as implementações testadas e a utilização futura do protótipo proposto.

1.4 METODOLOGIA

A metodologia utilizada neste trabalho baseia-se, inicialmente, numa pesquisa relacionada a projetos acadêmicos e comerciais de redes em malha alimentadas por energia solar, procurando dar maior ênfase aos controladores de carga. A partir desta pesquisa, a metodologia segue:

- Identificação das principais características e recursos de cada sistema pesquisado;
- Definição dos requisitos e especificações técnicas do controlador de carga proposto;
- Apresentação das funcionalidades da proposta a partir dos requisitos definidos;
- Definição dos dispositivos eletrônicos para executar as funcionalidades do controlador;
- Montagem do protótipo do controlador;
- Elaboração do programa do controlador utilizando a linguagem C e geração do firmware para ser gravado no microcontrolador;
- Montagem do sistema fotovoltaico em Laboratório com fonte de tensão/corrente ajustável no lugar do painel solar;
- Testar as funcionalidades do controlador;
- Avaliar os resultados dos testes.

1.5 PRINCIPAL CONTRIBUIÇÃO

A principal contribuição desta dissertação é o desenvolvimento de um controlador de carga microcontrolado (conjunto hardware e firmware) de baixo custo que além da função de controlar o processo de carga e descarga da bateria, permite o sensoriamento, atuação sobre o sistema fotovoltaico através de comandos remotos e de outras funcionalidades.

1.6 ORGANIZAÇÃO DO TEXTO

Esta dissertação está organizada da seguinte maneira: no capítulo 2 é feito um breve estudo dos trabalhos relacionados na área de sistemas fotovoltaicos utilizados para fornecer energia elétrica para pontos de acessos em redes em malha, procurando focar o trabalho no tipo de controlador de carga utilizado. São apresentados também dois controladores de carga comerciais utilizados inicialmente no projeto ReMoTe e alguns problemas encontrados durante a utilização desses que motivaram o desenvolvimento do controlador de carga microcontrolado apresentado nesta dissertação. O capítulo 3 aborda a fundamentação teórica

relacionada com sistemas fotovoltaicos: módulos fotovoltaicos, de carga, de controle de energia e tipo de bateria. Também são abordados conceitos sobre microcontroladores e controladores de carga microcontrolados.

O capítulo 4 apresenta o protótipo do controlador desenvolvido, descreve os principais componentes eletrônicos utilizados no CCM e os recursos adicionais implementados como: monitoramento remoto, execução de comandos remotos, modos de operação programados, duas saídas independentes, dois tipos de baterias e função *WatchDog Timer*. No capítulo 5 são apresentados os testes e a avaliação dos resultados. Finalmente o capítulo 6 apresenta as conclusões e as perspectivas para os trabalhos futuros no intuito de melhorar e aprimorar o protótipo desenvolvido.

Para complementar as informações contidas na dissertação, ao final da mesma é apresentado um Glossário com a definição de termos referentes as baterias.

Além disso, são apresentados os seguintes anexos: descrição funcional dos pinos do microcontrolador Atmega16, os *scripts* (WDT.sh, enviaMedida.sh e trataMedidas.sh) utilizados nos testes e o código fonte do *firmware* gravado no microcontrolador.

CAPÍTULO 2 – TRABALHOS RELACIONADOS

Este capítulo faz um breve relato de algumas redes em malha (*mesh*) que utilizam energia solar, procurando abordar as características dos controladores de carga. Além disso, são abordados: conceitos básicos sobre redes em malha, três trabalhos científicos sobre controladores de carga microcontrolados e alguns controladores de carga comerciais, incluindo os que foram testados para o projeto ReMoTe.

2.1 REDES EM MALHA (*MESH*)

As redes locais sem fio padrão são amplamente utilizadas hoje em dia graças à facilidade de instalação, flexibilidade, robustez e baixo custo dos equipamentos.

Esse tipo de rede é utilizada em locais onde se precisa de mobilidade ou onde a instalação de infraestrutura cabeada é cara, como por exemplo universidades, aeroportos, shoppings e áreas rurais. No caso das universidades, devido a maior concentração de usuários recomenda-se usar muitos pontos de acesso de baixa potência e a tecnologia wi-fi. Já nas áreas rurais, o uso de rede sem fio minimiza o custo com infraestrutura e neste caso o Wi-Max é mais recomendado devido ao maior alcance, podendo colocar poucas antenas p/ usuários bem espalhados.

Hoje em dia há disponíveis vários dispositivos para facilitar a mobilidade como laptops, tablets, pdas e telefones voip. Para estes, a rede sem fio é vital. Além disso, como os celulares estão cada vez mais inteligentes é interessante ter a alternativa de conexão wi-fi pois estas têm menor custo do que a comunicação de dados via operadora.

Os dispositivos utilizados nas redes sem fio 802.11 podem operar em dois modos. No modo infraestruturado (figura 2.1–a), a comunicação entre os dispositivos passa por uma estação base (ponto de acesso), e no modo Ad-hoc (figura 2.1–b), a comunicação é feita diretamente entre os dispositivos, isto é, não precisam da estação-base.

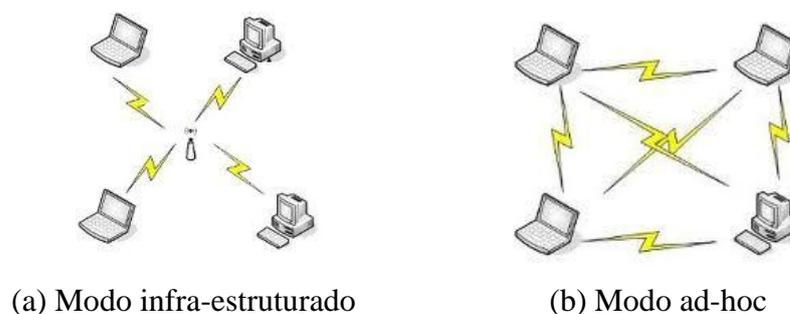


Figura 2.1 - Modos de operação da rede sem fio 802.11

O padrão 802.11 foi homologado para operar na banda ISM (Industrial, Scientific and Medical) nas faixas de 2.4 – 2.4835GHz e 5.15 – 5.825GHz com potência de até 1 watt. As redes sem fio que operam nessa banda não necessitam de licença para serem utilizadas e estão mais sujeitas a interferências provocadas por equipamentos que se utilizam dessa banda de frequência, como telefones sem fio e fornos micro-ondas.

A tecnologia das redes sem fio evoluiu para uma topologia denominada redes em malha (*mesh*) sem fio. As redes *mesh* surgiram a partir de redes Ad-hoc e possuem um backbone formado por roteadores estacionários interconectados como pode ser visto na figura 2.2.

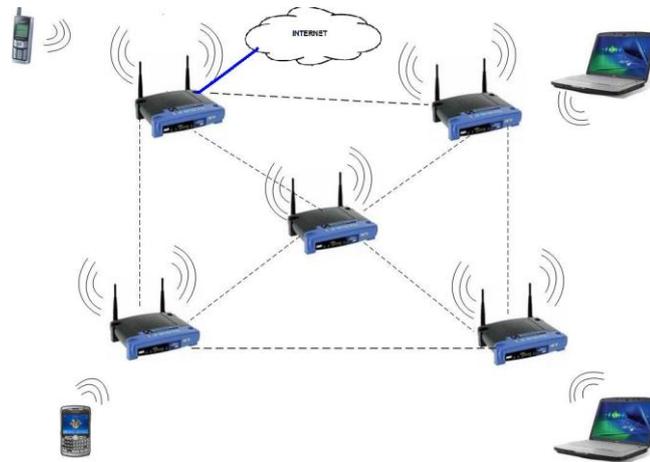


Figura 2.2 – Rede em malha sem fio

Como ilustrado na figura 2.2, dispositivos como laptops, celulares e telefones voips podem acessar a rede *mesh* para se comunicar entre si ou acessar a internet caso haja um gateway na rede. Algumas características das redes mesh são:

- Escalável, robusta e com a possibilidade de inclusão de uma série de serviços;
- Encaminhamento de pacotes por múltiplos saltos;
- Cada nó que compõe a rede pode receber, transmitir e rotear informações, além de atualizar e otimizar as rotas dinamicamente;
- Podem empregar topologia em malha completa ou parcial;
- Desempenho da rede é afetado em função da quantidade de saltos (hops) entre o nó de origem e o nó de destino;

Nos últimos anos, diversos campi de universidades e centros de pesquisa ao redor do mundo têm desenvolvido e vêm amplamente utilizando redes *mesh* como redes de acesso ao campus por usuários residentes nas suas proximidades [23]. Exemplos de projeto piloto de redes de acesso sem fio do tipo mesh são o ReMesh em Niterói/RJ [02], RoofNet no MIT

[04], Google Mesh na Califórnia [19], VMesh na Grécia [20], MeshNet na UCSB [21], Microsoft Mesh [22], entre outros.

2.2 TRABALHOS RELACIONADOS

Nesta seção são apresentadas algumas redes em malha alimentadas por energia solar.

2.2.1 SOLAR WI-FI GRID

O projeto Solar Wi-Fi Grid (Figura 2.3) desenvolvido pela organização sem fins lucrativos Green Wi-Fi [03] é um sistema de comunicação sem fio padrão 802.11 b/g alimentado por energia solar.

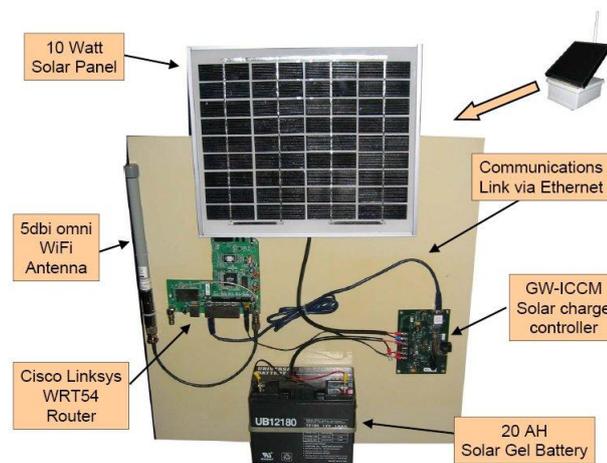


Figura 2.3 - Protótipo Wifi Grid Project (GREEN-WIFI, 2006)

Conforme mostrado na figura 2.3, o protótipo é formado de painel solar de 10 W, bateria chumbo-ácido de Gel com capacidade nominal de 20Ah, antena omnidirecional de 5dBi e controlador de carga microprocessado interconectado através de uma interface Ethernet ao roteador LinkSys modelo WRT54 que possui sistema operacional Linux embarcado com protocolo de roteamento desenvolvido pelos alunos do MIT Media Lab no projeto Roofnet[04].

As vantagens deste sistema são: monitoramento do painel solar, status da bateria e a potência usada. Este sistema também controla a carga disponível na bateria e pode colocar o roteador em um modo que consuma menos energia quando a carga na bateria cai abaixo de um certo nível. Pode até mesmo controlar o período em que o roteador fica ligado, o que aumenta a autonomia das baterias.

2.2.2 MERAKI SOLAR

A empresa Meraki [05] comercializa uma solução (figura 2.4) para comunicação sem fio Padrão 802.11 b/g alimentado por energia solar .



Figura 2.4 - Produto Meraki Solar (MERAKEI, 2011)

De acordo com as especificações, o sistema utiliza painel solar de 20 ou 40 W, antena omnidirecional de 2 dBi, bateria de Fosfato de Íons de Lítio com capacidade não divulgada e módulo de comunicação operando com protocolo de roteamento embarcado desenvolvido e licenciado pela organização. O controlador de carga não é divulgado.

2.2.3 SOLAR MESH

O projeto SolarMESH [06] foi desenvolvido pela Universidade McMaster no Canadá com o objetivo de interligar os prédios da universidade por uma rede comunicação sem fio padrão 802.11 em malha e disponibilizar cobertura Wi-Fi no campus.

A figura 2.5 mostra a primeira versão do protótipo desenvolvido com painel solar de silício policristalino e antena omnidirecional com ganho de 8 dBi. Os módulos de comunicação, de armazenamento e o controlador de carga não são divulgados.



Figura 2.5 - Protótipo SolarMesh (SOLARMESH, 2003)

Em 2003, quatro protótipos do projeto SolarMESH foram instalados em prédios da Universidade McMaster. A Figura 2.6 apresenta o plano final da rede sem fio com um total de seis pontos de acesso distantes em média de 200 m.



Figura 2.6 - Topologia da rede

2.2.4 SAM SOLAR

Pesquisadores da Universidade de São Paulo (USP) desenvolveram o sistema de comunicação sem fio em malha alimentado por energia solar denominado SAM SOLAR [07]. Também foi desenvolvido um controlador de carga com componentes discretos visando o mínimo custo e a possibilidade de correções durante os testes. Na figura 2.7 é possível ver o controlador e as baterias no interior da caixa hermética.



Figura 2.7 – Controlador de carga desenvolvido na USP

Além do controlador de carga, o sistema possui duas baterias de chumbo-ácido seladas de 6V/12AH, painel solar de 21 W, roteador Meraki mini 802.11b/g e antena omnidirecional

de 2 dBi. Segundo os desenvolvedores, a principal vantagem e inovação do controlador de carga desenvolvido é garantir a operação do módulo de comunicação no mínimo de 8 horas por dia, com mínima insolação e baterias com níveis mínimos de carga limitado em 5 Vdc sem comprometer a vida útil destas.

A figura 2.8 ilustra um dos nós da rede em malha instalado em poste de iluminação na Escola Politécnica da USP.



Figura 2.8 – Sistema de comunicação instalado na USP

A Figura 2.9 apresenta a arquitetura da rede sem fio e o local da instalação dos nós mesh que permitem a transmissão do sinal em um raio de 300 m.

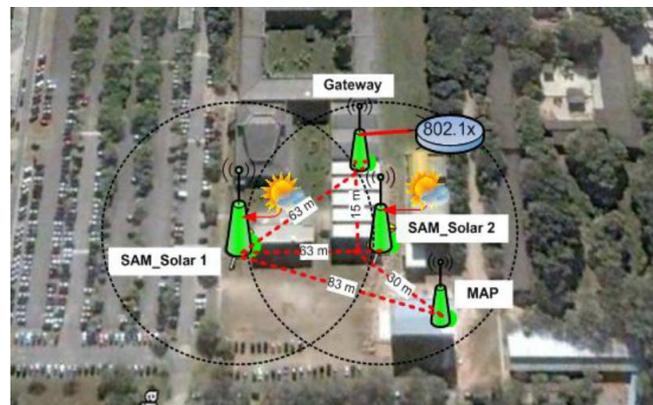


Figura 2.9 - Visualização da rede sem fio em malha solar – EPUSP

2.2.5 COMPARAÇÃO DOS TRABALHOS RELACIONADOS

Este capítulo apresentou alguns sistemas fotovoltaicos utilizados para prover energia elétrica para redes *mesh*. A tabela 2.1 compara os elementos dos sistemas abordados neste capítulo.

| | SOLAR Wi-FI GRID | SOLARMESH | MERAKI | SAM SOLAR |
|--|-------------------------|-------------------|---------------|-------------------------|
| CONTROLADOR DE CARGA | Desenvolvimento próprio | Não divulgado | Não divulgado | Desenvolvimento próprio |
| PAINEL SOLAR | Silício Poli 10 W | Silício Poli 21 W | 20W ou 40 W | Silício Mono 21 W |
| BATERIA | Chumbo-Ácido (12V/20Ah) | Não divulgado | Íons de Lítio | Chumbo-ácido (6V/24 Ah) |
| ROTEADOR | LinkSys Wrt54 | Não divulgado | Meraki | Meraki mini |
| ANTENA | Omni 5 dBi | Omni 8 dBi | Omni 2 dBi | Omni 2 dBi |
| SENSORIAMENTO | Sim | Não | Não | Não |
| ATUAÇÃO SOBRE O SISTEMA P/ECONOMIA DE ENERGIA | Sim | Não | Não | Não |
| COMUNICAÇÃO ENTRE ROTEADOR E CONTROLADOR DE CARGA | Interface Ethernet | Não divulgado | Não divulgado | Não possui |

Tabela 2.1 – Comparação das soluções apresentadas.

Pelo exibido na tabela 2.1, algumas soluções não informam especificações importantes como controlador de carga, tipo e capacidade da bateria. Também é possível verificar que a única solução que apresenta opção de sensoriamento e atuação sobre o sistema é o Solar Wi-fi Grid.

2.3 TRABALHOS CIENTÍFICOS

Nesta seção serão apresentados três trabalhos científicos de controladores de carga que utilizam microcontroladores.

Em [41] é apresentado o protótipo de um controlador de carga microcontrolado projetado com o objetivo de prolongar a vida útil da bateria. Além de ser versátil, pode possibilitar mais recursos que os controladores convencionais (não microcontrolado), como funções de monitoramento e controle do sistema. Um transistor Mosfet é utilizado como chave nos estágios de carga e descarga do controlador e um diodo Schottky (diodo de Bloqueio) é usado para impedir o fluxo de corrente para o painel solar quando a tensão neste é menor que a tensão na bateria. Leds são utilizados para indicar o status do sistema. O controlador proposto utiliza a estratégia PWM para controlar a fluxo de corrente que entra na bateria. Esta estratégia permite além de inúmeras vantagens, um processo de carga mais eficiente e mais rápido. Também é apresentado o algoritmo de controle de tensão e os pontos de operação do controlador como por exemplo, os valores de tensão máximo e mínimo que a bateria deve atingir sem sofrer danos a mesma. Os autores citam que o uso do microcontrolador permite que recursos mais sofisticados sejam implementados e que reduz o consumo de energia no processo de carregamento da bateria. Também citam a possibilidade

de uma maior flexibilidade no projeto do controlador através da forma mais fácil de ajustar os valores dos pontos de corte em função do tipo de bateria utilizada e da facilidade de modificar o algoritmo de carregamento da bateria apenas alterando alguns parâmetros do programa instalado no microcontrolador. Além disso, os autores trabalham na implementação de novas funcionalidades no controlador, como: uso de rastreamento do ponto de máxima potência para atingir uma forma eficaz para carregar baterias mais rapidamente, técnica de compensação de temperatura para alcançar um alto estado de carga e uso de display LCD¹ para exibir os valores de tensão da bateria, corrente e potência.

Em [42] é proposto um controlador de carga inteligente utilizando microcontrolador também com o objetivo de prolongar a vida útil da bateria. São apresentadas as vantagens obtidas com o uso do microcontrolador como: técnica PWM de carregamento da bateria, recurso de Watchdog Timer (permite uma maior confiabilidade pois qualquer mau funcionamento do programa faz com que o microcontrolador seja resetado), redução do consumo de energia pela utilização dos modos de sleep disponíveis nos microcontroladores (todas as funções são suspensas, com exceção do oscilador e temporizadores) e impossibilidade de copiar o programa gravado no microcontrolador (vantagem adicional para o fabricante). Este controlador inclui um processo de compensação de temperatura controlado por software. As portas I/O digitais aumentam a capacidade do controlador, como por exemplo: o rastreamento do ponto de máximo potência; o controle da resistência interna da bateria pode ser usado como um indicador de outras variáveis de sistema, tais como a idade da bateria, a capacidade de armazenamento e o estado de carga; e a monitoração do estado da carga através da integração da corrente em relação ao tempo pode permitir um meio de prever teoricamente a tensão de saída da bateria. Se a tensão real cai significativamente abaixo deste valor previsto, é possível gerar um alarme precoce e com isso a bateria pode ser substituída. Os testes realizados demonstraram que o protótipo do controlador de carga microcontrolado é versátil e possibilita mais funções como monitoração e controle do sistema. Também é discutido um algoritmo de controle de carga fotovoltaico.

Em [43] é apresentada uma técnica alternativa baseada em lógica Fuzzy para o carregamento de bateria chumbo-ácido em sistemas fotovoltaicos. Para isso, é utilizado um controlador de carga microcontrolador com o qual a arquitetura digital é implementada. O controlador Fuzzy proposto lê a tensão da bateria e controla a quantidade de corrente que flui do módulo fotovoltaico.

¹ LCD: Liquid Crystal Display: = display de cristal líquido

Isto é feito com a utilização de um conversor de energia Buck tipo DC / DC. O controlador gera um sinal PWM para fornecer o controle de chaveamento do MOSFET no circuito conversor Buck. É utilizado o algoritmo FZ-(PI + DP), uma melhoria do método de carregamento por tensão constante combinado com o método de carregamento por corrente constante. O algoritmo de carregamento permite carregar a bateria de chumbo-ácido com uma elevada corrente inicial do módulo fotovoltaico e, como a tensão de carga atingir o valor pré-definido, a corrente de carga diminui de acordo com as regras fuzzy para permitir que a tensão da bateria atinja uma carga completa sem sobrecarga. Um módulo de LCD pode ser adicionado ao microcontrolador para visualização da corrente de carga e de tensão e um conversor serial TTL/RS232 pode ser utilizado para monitorar o controlador através de um computador.

Esses trabalhos científicos mostram as várias vantagens e recursos extras que podem ser implementados nos controladores de carga quando são utilizados microcontroladores.

2.4 CONTROLADORES DE CARGA COMERCIAIS

Foram realizadas pesquisas sobre alguns controladores de carga comerciais e a tabela 2.2 resume as características dos mesmos.

| Fabricante / Modelo | Lead CCD1212 | MorningStar SunSaver-10 | SunLaber Power SLC-10 | MorningStar TriStar-45 |
|------------------------------|-------------------------|------------------------------------|----------------------------------|-----------------------------------|
| Configuração | série | série | série | série |
| Método de Carga | ON-OFF | PWM | PWM | PWM |
| Interface comunicação | Não possui | Não possui | Não possui | RS-232 |
| Tensão do sistema | 12 V | 12 V | 12 V | 12 V |
| Tensão desconexão da carga | 10,5 V | 11.5 V | 10,8 ou 11,0 V | * |
| Tensão de reconexão da carga | 11.5 V | 12.6 V | 12.5 V | * |
| Tensão máxima carga bateria | 14,4 V | 14.4 V | 14,5 | * |
| Corrente máxima de saída | 12 A | 10 A | 10 A | 10 A |
| Temperatura operação | - | -40°C a +85°C | -25°C a +65°C | -40°C a +60°C |
| Sensoriamento | Não | Não | Não | SIM |
| Atuação sobre o sistema | Não | Não | Não | SIM |
| Interface de comunicação | Não | Não | Não | RS-232 |
| Custo | R\$ 150,00 ¹ | R\$ 438,00 ¹ | R\$ 159,00 ¹ | R\$ 998,00 ¹ |

Tabela 2.2 – Resumo dos controladores de carga (*não informado)

¹ – Fonte: www.solarbrasil.com.br

Como pode ser visto na tabela 2.2, dos controladores comerciais pesquisados, o único que permite sensoriamento e atuação é o TriStar-45. Porém, este dispositivo possui a limitação de operar apenas em um dos seguintes modos de cada vez: carregamento de bateria, controle da carga conectada ao sistema para evitar a descarga profunda da bateria ou controle do carregamento da bateria a partir de outras fontes de energia elétrica. Dois ou mais dispositivos devem ser usados para prover múltiplas funções. Além dessas limitações, este controlador possui um custo elevado (R\$ 998,00), inviabilizando o seu uso.

Os controladores CCD1212 e SunSaver-10 foram testados no laboratório MídiaCom e a seguir serão apresentados os problemas encontrados.

O modelo nacional CCD1212 foi o primeiro controlador de carga utilizado e durante os testes foi encontrado um problema de travamento do roteador quando a tensão da bateria atingia o valor definido no manual do controlador como ponto de desligamento da carga (10,5 V). Mesmo após a tensão na bateria ter atingido o nível pleno de carga, o roteador continuava travado.

Para verificar o nível de sensibilidade do controlador em função das variações de tensão/corrente da bateria e se influenciava no problema de travamento, foi montada no laboratório MídiaCom (www.midiacom.uff.br) uma bancada de testes composta de fonte de tensão/corrente ajustável, multímetro, controlador de carga e roteador.

Inicialmente, a fonte foi ajustada para 12 V e interconectada ao controlador de carga substituindo a bateria. A tensão da fonte foi sendo reduzida gradativamente numa razão de 0,2V em cada passo. Quando a tensão da fonte atingiu o limiar de 10,5 V, o LED verde responsável por indicar o status de bateria carregada foi desligado, entretanto constatou-se que o roteador continuava ligado, porém travado. O roteador só foi desligado quando a tensão atingiu 9,8 V, indicando valor diferente do especificado no manual. Neste ponto, o LED vermelho que indica carga desligada foi ligado.

Em seguida, foi realizado teste inverso. Aumentou-se a tensão da fonte de forma gradativa numa razão de 0,2V em cada passo. Foi constatado que a partir de 10,5V o roteador ligava parcialmente, entretanto, não havia corrente suficiente para ligá-lo completamente. Isto resultou no travamento do roteador, e mesmo quando a tensão da fonte alcançava valores equivalentes a da bateria plenamente carregada, o travamento continuou.

Segundo Coutinho [10], o roteador WRT54G (LinkSys) precisa de aproximadamente 160mA para ligar, mas o seu funcionamento em *standby* necessita de pelo menos 220mA. O roteador liga parcialmente quando há corrente suficiente para realizar tal fato, entretanto, como não há corrente suficiente para o restabelecimento de todas as funções do roteador,

ocorre o travamento. A corrente necessária para o funcionamento ideal do roteador somente é obtida com uma tensão de aproximadamente 11,4V, porém uma vez que ocorre o travamento, ele persiste.

Os ensaios mostram que os pontos de ajustes da tensão de desconexão da carga em função da baixa tensão na bateria de alguns controladores apresentam valores abaixo da profundidade de descarga máxima [27]. Outros controladores até mesmo permitem descargas excessivas pois seus pontos de ajustes estão abaixo da tensão final mínima da bateria.

Devido ao problema de travamento descrito, o controlador CCD-1212 foi substituído pelo controlador de carga importado SunSaver-10. Foram repetidos os mesmos testes realizados com o controlador da seção anterior e o roteador não apresentou problemas de travamento. Em função disso, este controlador foi integrado ao kit de alimentação solar do projeto ReMoTe.

Apesar deste controlador funcionar bem no laboratório MídiaCom, o problema de travamento foi constatado em campo. Este problema indicou a necessidade de outra funcionalidade ao controlador de carga microcontrolado proposto neste trabalho para reiniciar o roteador em caso de travamento.

No esforço de pesquisa desta dissertação, a única solução que apresenta algumas funcionalidades semelhantes à proposta nesta dissertação é o Solar Wi-Fi Grid. Como visto na seção 2.1, esta solução permite o sensoriamento (do painel solar, do status da bateria e da potência usada) e atuação no sistema para economia de energia. Não foi possível obter o custo dessa solução.

O controlador de carga microcontrolado proposto nesta dissertação possui além de sensoriamento e atuação no sistema, outras funcionalidades e recursos que são apresentados no capítulo 4.

No próximo capítulo são apresentados conceitos básicos dos sistemas fotovoltaicos, microcontroladores e controladores de carga microcontrolados.

CAPÍTULO 3 – FUNDAMENTAÇÃO TEÓRICA / SISTEMAS FOTOVOLTAICOS

Para entender o funcionamento de um controlador de carga, é preciso compreender o que é um sistema geral de alimentação por energia solar. Neste capítulo são apresentados os conceitos básicos de um sistema alimentado por energia solar, ou seja, os seus componentes como módulos fotovoltaicos, módulos armazenadores, módulos alimentados, módulos controladores de energia. Será abordado cada um dos dispositivos eletrônicos que compõem este sistema. Além disso, são abordados conceitos básicos sobre microcontroladores, controladores de carga microcontrolados/microprocessados.

3.1 ENERGIA SOLAR

O aproveitamento da energia gerada pelo Sol, inesgotável na escala terrestre de tempo, tanto como fonte de calor quanto de luz, é hoje, sem sombra de dúvidas, uma das alternativas energéticas mais promissoras para que possam ser enfrentados os desafios do novo milênio [11]. Quando se fala em energia, deve-se lembrar que o Sol é responsável pela origem de praticamente todas as outras fontes de energia. Em outras palavras, as fontes de energia são, em última instância, derivadas, em sua maioria, da energia do Sol. É a partir da energia do Sol que se dá a evaporação, origem do ciclo das águas, que possibilita o represamento e a conseqüente geração de eletricidade (hidroeletricidade). A radiação solar também induz a circulação atmosférica em larga escala, causando os ventos. Petróleo, carvão e gás natural foram gerados a partir de resíduos de plantas e animais que, originalmente, obtiveram a energia necessária ao seu desenvolvimento da radiação solar. As reações químicas às quais a matéria orgânica foi submetida, a altas temperaturas e pressões, por longos períodos de tempo, também utilizaram o Sol como fonte de energia. É também por causa da energia do Sol que a matéria orgânica, como a cana-de-açúcar, é capaz de se desenvolver, fazer fotossíntese para, posteriormente, ser transformada em combustível nas usinas.

A Terra recebe anualmente $1,5 \times 10^{18}$ kWh de energia solar, o que corresponde a 10.000 vezes o consumo mundial de energia neste período [12].

A luz solar é uma radiação eletromagnética que se propaga a uma velocidade de 300.000 km/s. Além de ser responsável pela manutenção da vida na Terra, a radiação solar constitui-se numa importante fonte energética, havendo um enorme potencial de utilização por meio de sistemas de captação e conversão em outra forma de energia como térmica, elétrica e mecânica.

Devido à intensidade de energia solar que incide sobre a terra depender da inclinação do sol em relação ao horizonte, para capturar o máximo de energia solar ao longo do ano, um sistema de captação solar deve ser colocado inclinado com relação à horizontal de um ângulo próximo ao da latitude onde está instalado. No caso do Hemisfério Sul, o sistema de captação deve também estar orientado para o Norte Geográfico.

3.2 SISTEMAS FOTOVOLTAICOS

Os sistemas fotovoltaicos são cada vez mais utilizados em áreas remotas desprovidas de rede elétrica convencional devido às suas diversas vantagens como modularidade, baixos custos em manutenção e longa vida útil.

Um sistema fotovoltaico pode ser classificado em três categorias distintas: sistemas isolados, híbridos e conectados à rede. Estes serão descritos a seguir. Dependendo do sistema, podem possuir os módulos: fotovoltaico, controle de potência, armazenamento de energia e alimentado.

3.2.1 SISTEMAS ISOLADOS

Um sistema isolado não está conectado a rede elétrica convencional como ilustrado na figura 3.1.

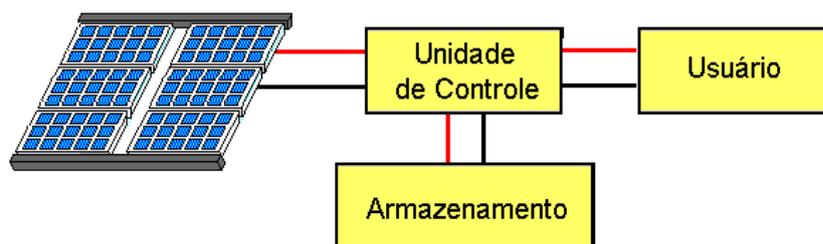


Figura 3.1- Sistema fotovoltaico isolado (CRESESB, 2004)

Este sistema normalmente utiliza algum tipo de sistema de armazenamento de energia, como, por exemplo, um banco de baterias, para ser capaz de prover energia na ausência de insolação. Para o controle de energia, deve ser utilizado um controlador de carga, que tem por função principal evitar a descarga profunda ou sobrecarga da bateria, evitando danos à mesma. Caso seja necessário alimentar dispositivos elétricos com energia elétrica alternada é necessário o uso de um inversor, que transforma a corrente contínua proveniente dos sistemas fotovoltaicos em corrente alternada.

3.2.2 SISTEMAS HÍBRIDOS

Os sistemas híbridos (figura 3.2) são compostos de mais de uma fonte de geração de energia, como módulos fotovoltaicos, turbinas eólicas, geradores a diesel, etc.

Esse tipo de sistema não é conectado a rede elétrica convencional, e geralmente são empregados em sistemas de médio e grande porte com capacidade para atender um número bem maior de usuários do que os sistemas isolados vistos anteriormente.

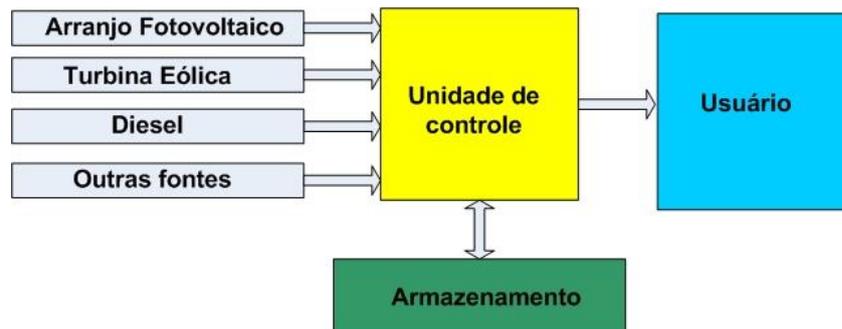


Figura 3.2 - Sistema fotovoltaico híbrido

3.2.3 SISTEMAS INTERLIGADOS À REDE

Nos sistemas interligados à rede (figura 3.3), toda a energia elétrica gerada pelos painéis fotovoltaicos é entregue à rede elétrica após passar por inversores. Além disso, não são utilizados módulos de armazenamento e nem de controle de energia, pois a energia pode ser suprida pela rede na ausência de insolação.

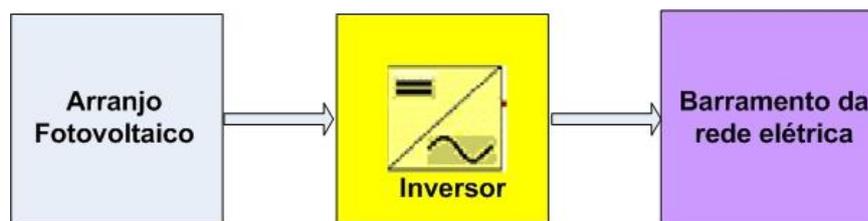


Figura 3.3 - Sistema Fotovoltaico Interligados à Rede

3.3 MÓDULO FOTOVOLTAICO

Um módulo fotovoltaico é formado por um conjunto de células fotovoltaicas. Cada célula fornece uma potência elétrica em torno de 1 a 3 Watts (corrente máxima por volta de 3A e tensão muito baixa entre 0,7V a 1V) e são agrupadas em série e/ou paralelo para atingir

os valores requeridos de tensão e potência. A figura 3.4 mostra a ligação em paralelo de células.

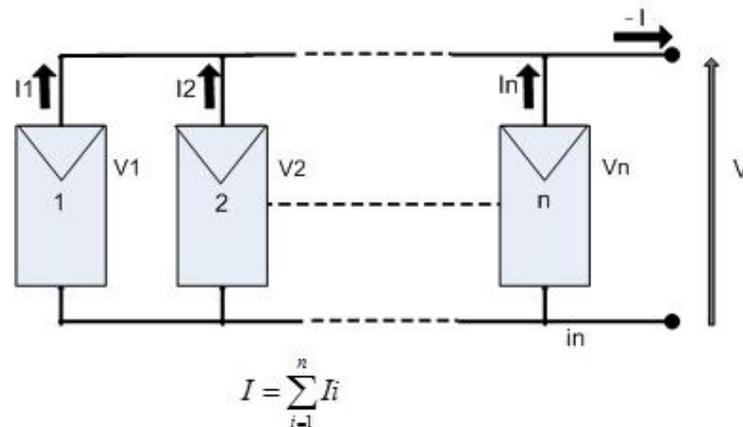


Figura 3.4 - Células em paralelo

Para serem utilizadas em sistemas fotovoltaicos com a finalidade de carregar baterias, por exemplo, a ligação mais comum das células é em série (Figura 3.5). Desta forma, a soma das tensões de cada célula resultará na tensão nominal do módulo.

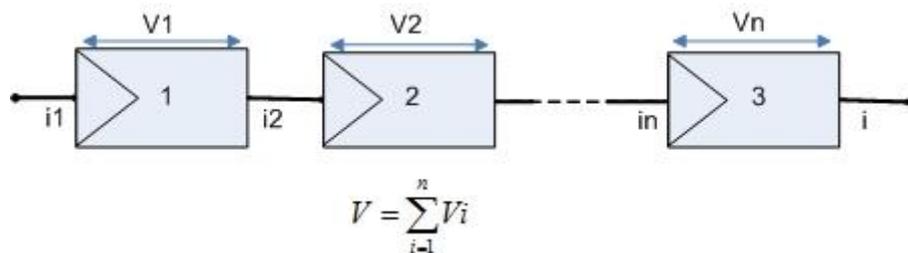


Figura 3.5 - Células em série

Uma bateria de tensão nominal de 12 V precisa ser carregada com uma tensão em torno de 14 V para alcançar pleno estado de carga. Devido ao efeito da temperatura e às perdas que ocorrem nos cabos e nos diodos de bloqueio, os módulos fotovoltaicos devem gerar uma tensão de aproximadamente 16V [11].

3.3.1 CARACTERÍSTICAS ELÉTRICAS DOS MÓDULOS FOTOVOLTAICOS

Uma das principais características elétricas dos módulos fotovoltaicos é a potência de pico, especificada em W_p^1 (Watt pico). Além disso, existem outras características elétricas como:

- **Voltagem de Circuito Aberto (Voc)** – é a tensão do módulo fotovoltaico quando o mesmo não está conectado a nenhuma carga, ou seja, quando não há corrente fluindo;
- **Corrente de Curto Circuito (Isc)** – é a corrente máxima possível de um módulo fotovoltaico que flui livremente em um circuito externo sem resistência;

A figura 3.6 apresenta a curva característica I x V com a corrente Isc x a tensão Voc.

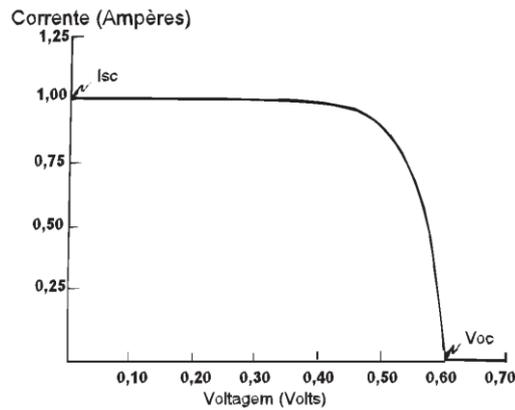


Figura 3.6 - Curva característica I x V mostrando a corrente Isc (CRESESB, 2004)

- **Potência Máxima (Pm)** - é a potencia máxima gerada por um módulo fotovoltaico;
- **Voltagem de Potência Máxima (Vmp)** – é a tensão máxima gerada por um módulo fotovoltaico;
- **Corrente de Potência Máxima (Imp)** - é a corrente máxima gerada por um módulo fotovoltaico);

As figuras 3.7 e 3.8 apresentam respectivamente, a curva típica de potência versus tensão e os parâmetros de potência máxima.

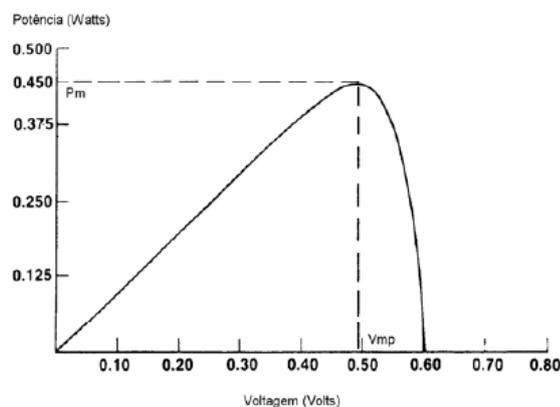


Figura 3.7 - Curva típica de potência versus tensão (CRESESB, 2004)

¹ A especificação da potência do módulo fotovoltaico é dada em Wp (Watt pico), associada às condições padrão de testes (STC - *Standard Test Conditions*): radiação solar de 1 kW/m^2 , temperatura de célula 25°C e Massa de Ar 1,5.

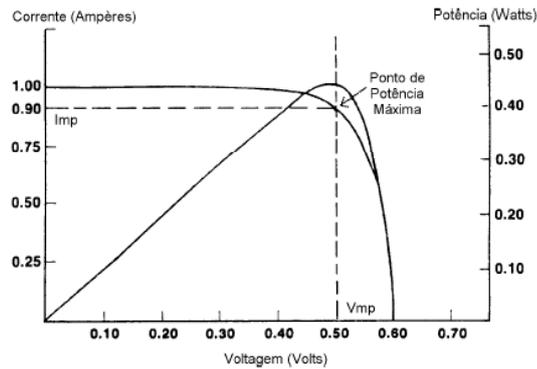


Figura 3.8 - Parâmetros de potência máxima - (CRESESB, 2004)

A condição padrão para se obter as curvas características dos módulos é definida para radiação de 1000W/m^2 (radiação recebida na superfície da Terra em dia claro, ao meio dia), e temperatura de 25°C na célula.

3.3.2 FATORES QUE AFETAM AS CARACTERÍSTICAS ELÉTRICAS DOS MÓDULOS

A intensidade luminosa e a temperatura das células influenciam diretamente nas características elétricas de um módulo fotovoltaico.

O aumento da intensidade luminosa produz de forma linear o aumento na corrente gerada nos módulos, conforme pode ser visto na figura 3.9.

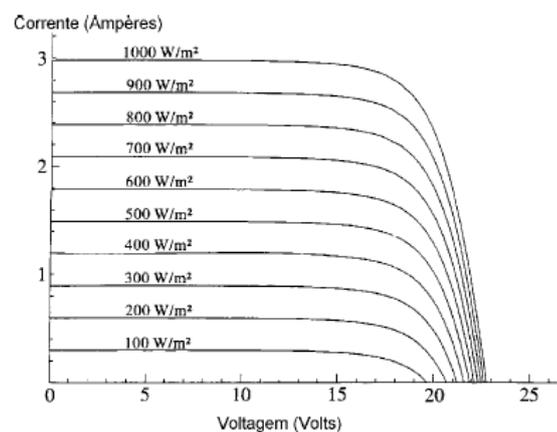


Figura 3.9 - Efeito causado pela variação de intensidade luminosa. (CRESESB, 2004)

Por outro lado, o aumento da temperatura na célula diminui a eficiência do módulo, fazendo com que fique abaixo dos pontos de operação da potência máxima gerada. A figura 3.10 ilustra esse efeito.

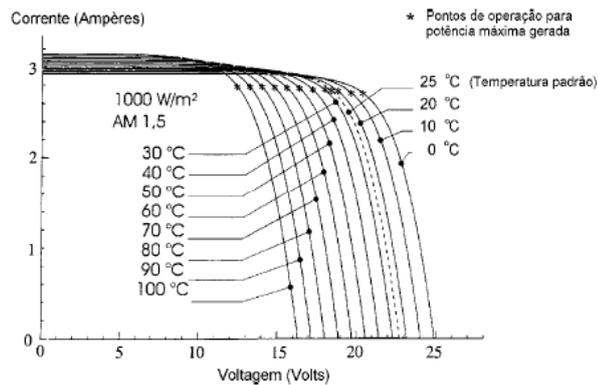


Figura 3.10 - Efeito causado pela temperatura na célula. (CRESESB, 2004)

3.3.3 TIPOS DE MÓDULOS FOTOVOLTAICOS

Os módulos fotovoltaicos são classificados em função dos tipos de células que são construídos. Nesta seção são abordadas as células que utilizam o silício em sua composição. Também são produzidos células de outros materiais, como *Cadmium Telluride* (CdTe) e *Copper Indium Gallium Diselenide* (CIGS).

TIPOS DE CÉLULAS DE SILÍCIO

As células fotovoltaicas são fabricadas na sua grande maioria usando o silício (Si), podendo ser constituída de cristais monocristalinos, policristalino ou de silício amorfo.

SILÍCIO MONOCRISTALINO (c-Si)

Esse material é basicamente o mesmo utilizado na fabricação de circuitos integrados para microeletrônica. As células de silício monocristalino (figura 3.11) são formadas em fatias de um único grande cristal, previamente crescido e enfatiado.

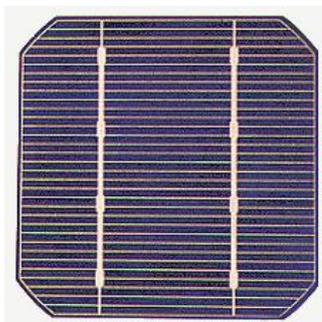


Figura 3.11 – Módulo de silício monocristalino

A grande experiência na sua fabricação e pureza do material, garantem alta confiabilidade do produto e altas eficiências. Enquanto o limite teórico de conversão da luz solar em energia elétrica, para esta tecnologia é de 27%, valores nas faixas de 12 a 16% são encontrados em produtos comerciais [11].

Dentre as células fotovoltaicas que utilizam o silício como material base, as monocristalinas são, em geral, as que apresentam as maiores eficiências na conversão da luz solar em energia elétrica, chegando a valores nas faixas de 12 a 16%. Porém, Devido às quantidades de material utilizado e à energia envolvida na sua fabricação, essa tecnologia apresenta sérias barreiras para redução de custos, mesmo em grandes escalas de produção.

SILÍCIO POLICRISTALINO (m-Si)

As células de silício policristalino (figura 3.12) são mais baratas que as de silício monocristalino por exigirem um processo de preparação das células menos rigoroso. A eficiência, no entanto, cai um pouco em comparação as células de silício monocristalino.

Basicamente, as técnicas de fabricação de células policristalinas são as mesmas na fabricação das células monocristalinas, porém com menores rigores de controle.

Ao longo dos anos, o processo de fabricação tem alcançado eficiência máxima de 12,5% em escalas industriais.

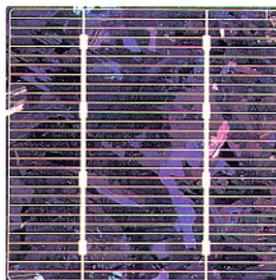


Figura 3.12 - Módulo de silício policristalino

SILÍCIO AMORFO (a-Si)

Uma célula de silício amorfo (Figura 3.13) difere das demais estruturas cristalinas (mono ou poli) por apresentar alto grau de desordem na estrutura dos átomos. A utilização de silício amorfo para uso em fotocélulas tem mostrado grandes vantagens tanto nas propriedades elétricas quanto no processo de fabricação. Além disso, o silício amorfo vem se mostrando uma tecnologia adequada para sistemas fotovoltaicos de baixo custo devido às seguintes vantagens: processo de fabricação relativamente simples e barato; possibilidade de fabricação de células com grandes áreas e baixo consumo de energia na produção.



Figura 3.13 – Módulo de silício amorfo (GREENPRO, 2004)

Devido às células de película fina que compõem os módulos de silício amorfo serem longas e estreitas fitas, sofrem menos os efeitos de sombreamento do que os módulos de célula cristalina. Como exemplo, uma folha de árvore pode cobrir completamente uma célula cristalina, acarretando uma diminuição significativa da eficiência do módulo, pois a célula encoberta se comporta como uma resistência. No caso de um módulo de silício amorfo, essa mesma folha cobrirá parcialmente várias células, porém apenas uma pequena área de cada uma destas células efetivamente será sombreada e a potência apenas será reduzida de forma proporcional à área sombreada. Essa característica de resistência às sombras parciais faz com que os módulos de silício amorfo sejam recomendados para locais onde não se possa evitar a incidência de sombra.

Mesmo apresentando um custo reduzido na produção, o uso de silício amorfo apresenta duas desvantagens: a primeira é a baixa eficiência de conversão comparada às células mono e policristalinas de silício; em segundo, as células são afetadas por um processo de degradação logo nos primeiros meses de operação, reduzindo assim a eficiência ao longo da vida útil. Taxa máxima de eficiência na conversão: de 8%.

A tabela 3.1 mostra a comparação entre os tipos de células de silício utilizadas para a produção de módulos fotovoltaicos.

| | Monocristalino | Policristalino | Amorfo |
|-------------------|--------------------------|---------------------------|-------------------------------------|
| Eficiência Máxima | 14 % | 13 % | Até 8 % |
| Vantagem | Mais eficientes | Mais baratas que as mono | Mais resistente às sombras parciais |
| Desvantagem | Altos custos na produção | Menos eficientes que mono | Baixa eficiência |

Tabela 3.1 - Comparação entre células de silício

O custo dos módulos fotovoltaicos no mercado internacional normalmente é estimado em dólar por cada Watt de pico (U\$/Wp). Em pesquisas feitas em Maio/2011 num site internacional, um módulo de 50 Wp apresentou um valor médio de U\$ 100, resultando num

custo de 2 dólares por cada Wp (2 U\$/Wp) ou cerca de 3,4 R\$/Wp. No Brasil, o preço médio está em torno de R\$ 13,6 R\$/Wp. (Fonte: www.solarbrasil.com.br/Julho de 2011).

SOMBRAS PARCIAIS

Com o objetivo de verificar a eficiência do painel solar de silício policristalino (projeto ReMoTe) quando no mesmo incidiam sombras, foram realizados vários testes gerando sombras em algumas células como pode ser observado na figura 3.14 - (a), (b) e (c).

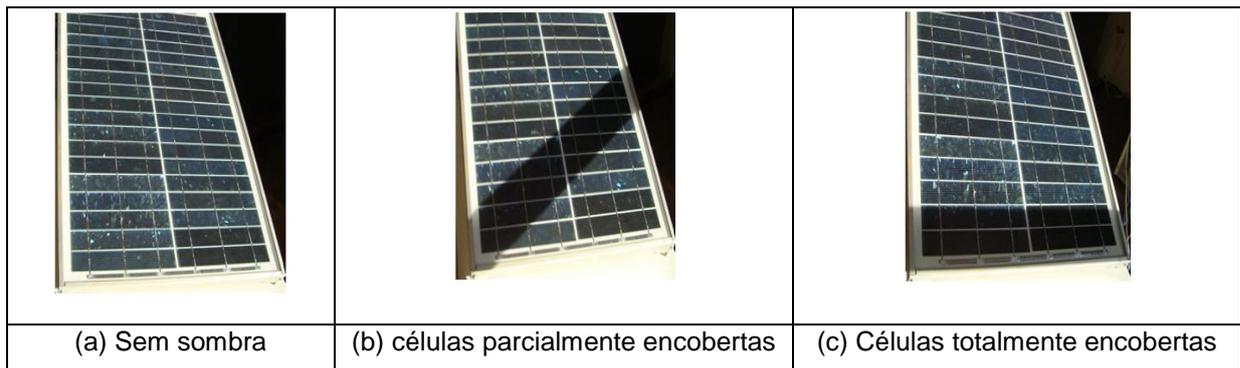


Figura 3.14 – Células sem e com sombras

Com o uso do Kit de Sensoriamento, foi possível medir a corrente e a tensão gerada do painel solar, o que permitiu calcular a potência instantânea. A tabela 3.2 ilustra os valores obtidos durante os testes em cada um dos casos.

| | Sem sombra | Células parcialmente encobertas | Células totalmente encobertas |
|----------------------|------------|---------------------------------|-------------------------------|
| Corrente (mA) | 315 | 120 | 50 |
| Tensão (V) | 12,49 | 12,39 | 12,44 |
| Pot. Instantânea (W) | 3,93 | 1,48 | 0,62 |

Tabela 3.2 – Redução da potencia em função da sombra

Como pode ser visto pela tabela 3.2, as sombras produzem uma redução significativa na eficiência do painel. No caso em que algumas células são parcialmente encobertas, a potência cai para 1,48 W, o que equivale a uma redução de 60 %.

Já no caso em que algumas células são totalmente encobertas, há uma redução de cerca 84,5 % da potência instantânea gerada pelo painel solar.

Os testes demonstraram quedas significativas na eficiência do painel (superiores a 80%) mesmo com sombras parciais (da ordem de 10% da superfície do painel). Isso resulta em problemas de autonomia em sistemas fotovoltaicos onde há incidência de sombras.

A correção deste problema passa pela mudança dos pontos de instalação do painel solar ou pela substituição dos painéis por modelos de silício amorfo, resistentes às sombras parciais.

3.4 MÓDULO DE ARMAZENAMENTO

Nesta seção são abordadas as principais características de alguns tipos de baterias, principalmente as que são utilizadas nos módulos de armazenamento dos sistemas fotovoltaicos.

As baterias armazenam a energia elétrica gerada pelo módulo fotovoltaico para que a energia armazenada esteja disponível ao módulo de carga quando necessário.

A bateria é um dispositivo que converte energia química em energia elétrica produzindo corrente contínua (DC).

A unidade mais simples de operação de uma bateria é chamada de “célula eletroquímica” ou, simplesmente, “célula”. Uma bateria pode ser composta de apenas uma célula ou do arranjo elétrico de diversas células [11].

As baterias podem ser classificadas em função do tipo de célula de que são compostas: primárias e secundárias. As baterias primárias transformam energia química em energia elétrica de forma irreversível, ou seja, não são recarregáveis e sua energia não pode ser restabelecida. Já as baterias secundárias são recarregáveis. Com o auxílio de uma fonte externa é possível recuperar a composição química inicial e deixá-la pronta para um novo ciclo de operação um número limitado de vezes.

As baterias secundárias podem ser classificadas de acordo com a aplicação como automotivas, tração, estacionárias e fotovoltaicas. As Automotivas são projetadas fundamentalmente para descargas rápidas com elevadas taxas de corrente e com reduzidas profundidades de descarga. Essa condição é típica na partida de motores de automóveis. As baterias de Tração são indicadas para alimentar equipamentos móveis elétricos como por exemplo, empilhadeiras, e são projetadas para operar em regime de ciclos diários profundos com taxa de descarga moderada (C/6). As Estacionárias são direcionadas tipicamente para aplicações em que as baterias permanecem em flutuação e são solicitadas ocasionalmente para ciclos de carga/descarga, como em sistemas de back-up. As fotovoltaicas são projetadas para ciclos diários rasos com taxas de descarga reduzidas e devem suportar descargas profundas esporádicas devido à possível ausência de geração de energia solar pelo módulo fotovoltaico em dias nublados.

Quanto à forma de confinamento do eletrólito, as baterias recarregáveis também podem ser diferenciadas em “abertas” ou “seladas”. Enquanto que as abertas (ventiladas) necessitam de verificação periódica do nível do eletrólito líquido e livre (não confinado), as baterias seladas possuem o eletrólito confinado no separador ou sob a forma de GEL e não precisam de manutenção periódica para repor as características químicas.

3.4.1 TIPOS DE BATERIAS

O universo das baterias pode ser dividido segundo a tecnologia de construção e diversas possibilidades de composição dos materiais das células. Nesta seção são abordadas as baterias chumbo-ácido, níquel-cádmio e íons de lítio.

BATERIAS DE CHUMBO-ÁCIDO

As baterias de chumbo-ácido são as mais utilizadas em sistemas fotovoltaicos devido à disponibilidade e boa relação custo-benefício, entre outros fatores. Respondem pela quase totalidade dos sistemas fotovoltaicos já instalados e tem-se dado preferência às abertas para sistemas de grande capacidade e às seladas para sistemas de pequena capacidade [11].

Existem três tipos de tecnologias utilizadas nas baterias chumbo-ácido: as Fluidas usam eletrólito fluido, as de Gel possuem o eletrólito com a consistência de Gel e as AGM utilizam entrelaçado de fibra de vidro no lugar do Gel.

As baterias fluídas utilizam placas (eletrodos) de dióxido de chumbo (PbO_2) e chumbo metálico (Pb). Essas placas são imersas em uma solução diluída de ácido sulfúrico (H_2SO_4) que se comporta como o eletrólito. A diferença de materiais origina uma diferença de potencial entre os eletrodos. Para aplicações domésticas de sistemas fotovoltaicos, destacam-se as baterias fluidas chamadas “Deep-cycle”, que suportam ciclos acentuados de descarga sem sofrerem danos.

Dentre as baterias chumbo-ácido seladas (que não precisam de manutenção periódica para repor as características químicas) estão a de GEL e AGM. As baterias de Gel contêm um aditivo de sílica que envolve o eletrólito. Essas baterias usam a tecnologia VRLA (*Valve Regulated Lead Acid Battery*): um mecanismo de válvula de regulação que permite o escape dos gases, hidrogênio e oxigênio, durante o processo de carga que regula a pressão interna. As baterias de Gel possuem as seguintes características: número muito maior de ciclos de carga e descarga, suporta temperaturas mais baixas e não suporta taxa de carga muito elevada (máximo de $C/20$).

Já as baterias AGM (*Absorbed Glass Mat*), são o último passo na evolução das baterias chumbo-ácido, recarregáveis e seladas. Em vez de usarem Gel, usam fibra de vidro para envolver o eletrólito, o que contribui para que sejam as mais resistentes aos impactos. Possui as seguintes características: ótima potência específica (W/Kg), energia específica (J/Kg) menor do que das baterias chumbo-ácido tradicionais, resistência interna muito baixa, suporta temperaturas mais altas e autodescarga mais lenta.

ESTADO DA CARGA DAS BATERIAS CHUMBO-ÁCIDO

O estado de carga da bateria é fundamental para o bom desempenho e operação do sistema que está sendo alimentado. A melhor forma de saber o estado da carga da bateria é medir a densidade do eletrólito, porém, para baterias seladas, esse método não é possível, de maneira que neste caso deve-se utilizar um medidor de voltagem para determinar, aproximadamente, o estado de carga da bateria. A tabela 3.3 mostra o estado de carga relacionada com os valores típicos de voltagem para uma bateria com tensão nominal de 12V. Esses valores podem variar em função das cargas e descargas e do fabricante da bateria.

| Estado da carga (%) | Tensão (V) |
|---------------------|------------|
| 100 | 12,70 |
| 90 | 12,50 |
| 80 | 12,42 |
| 70 | 12,32 |
| 60 | 12,20 |
| 50 | 12,06 |
| 40 | 11,90 |
| 30 | 11,75 |
| 20 | 11,58 |
| 10 | 11,31 |
| 0 | 10,50 |

Tabela 3.3 – Estado de carga para bateria de 12V ²

²Fonte: <http://www.universitybattery.com>

Quanto menor a taxa de descarga de uma bateria, maior é a sua vida útil. Por isso, recomenda-se que a mesma trabalhe numa faixa acima de 40% de seu estado de carga. Se utilizada frequentemente na faixa abaixo de 30% a vida útil da bateria será reduzida drasticamente.

BATERIAS DE NÍQUEL-CÁDMIO

O primeiro tipo de pilha ou bateria recarregável portátil desenvolvida foi a bateria de níquel-cádmio (NiCd). Apesar de serem geralmente mais baratas, essas baterias têm as seguintes desvantagens em relação as tecnologias mais modernas: menor tempo de vida útil, menor capacidade de carga, são muito nocivas ao meio ambiente devido ao cádmio e sofrem do chamado efeito memória³.

Esse tipo de bateria é considerado antigo, sendo substituído por outros tipos de bateria como as de hidreto metálico de níquel (NiMH) e de íons de lítio (Li-Ion).

BATERIAS DE ÍONS DE LÍTIO

São nomeadas de acordo com o material utilizado no catodo. A maioria das baterias de íons de lítio usada em produtos eletrônicos são as baterias de lítio de óxido de cobalto (LiCoO₂). Outras variedades de baterias de íons de Lítio são: óxido de lítio-manganês (LiMn₂O₄) e óxido de lítio-níquel (LiNiO₂). As baterias de íons de Lítio têm várias vantagens em relação a outros tipos de baterias: armazenam o dobro da energia de outras baterias (como as de níquel-cádmio), não possuem efeito memória, não é necessário carregá-las durante um período de tempo elevado da primeira vez que são utilizadas, o que acontece com outras baterias. As desvantagens são: preço mais elevado, sofrem um processo de envelhecimento (perdem a sua capacidade de armazenamento de energia mesmo que não estejam sendo utilizadas) e são muito sensíveis a temperatura elevada.

Quando submetidas à sobrecarga, as baterias de lítio geram gases que podem causar a ruptura da capa da bateria. Infelizmente o lítio se oxida rapidamente com enorme geração de calor o que pode fazer a bateria explodir. Há relatos de bateria de celular que explodiram após sofrerem este problema [35]. Isto pode ser evitado através de um circuito que limita a carga e descarga das células de lítio.

³ Efeito memória - processo no qual a bateria não consegue se carregar totalmente após a primeira vez ter sido carregada parcialmente

BATERIAS LiFePO_4 (Lítio Ferro Fosfato)

Em 1996, a Universidade do Texas (e outros colaboradores) descobriram o fosfato como material para ser usado como catodo em baterias de lítio [13].

A bateria LiFePO_4 (Lítio Ferro Fosfato) usa uma química derivada do íon de lítio e compartilha de muitas vantagens e desvantagens de outros tipos de baterias de íons de lítio. Devido ao material do catodo em escala nanométrica de fosfato, as baterias de LiFePO_4 têm um bom desempenho eletroquímico com baixa resistência.

De acordo com os desenvolvedores, as principais vantagens deste tipo de baterias são:

- Boa estabilidade térmica e química;
- Suporta altas temperaturas;
- Ciclo de vida útil maior;
- Custo mais baixo na produção (em relação às baterias de íon de lítio);
- Não poluente;
- Maior potencia de pico ou corrente.

Algumas utilizações destas baterias são: Veículos elétricos (bicicletas, motos e carros), Programa *One Laptop per Child* (OLPC) [14] e no *Killacycle* [15], a motocicleta mais rápida do mundo.

3.4.2 CARACTERÍSTICAS IDEAIS DAS BATERIAS PARA USO EM SISTEMAS FOTOVOLTAICOS

A operação de uma bateria usada em um sistema fotovoltaico deve atender a dois tipos de ciclos [11]:

- Ciclos rasos a cada dia;
- Ciclos profundos por vários dias com tempo nublado ou semanas, durante o inverno.

Os ciclos profundos ocorrem quando a bateria não consegue ser recarregada suficientemente para repor a quantidade de energia consumida durante todo o dia. Em função disso, o estado de carga depois de cada ciclo diário é ligeiramente reduzido e, se isto ocorrer por vários dias, levará a um ciclo profundo. Quando o nível de insolação melhora, há um carregamento extra, aumentando o estado de carga depois de cada ciclo diário.

As características mencionadas a seguir devem ser observadas para que as baterias tenham um bom desempenho quando instaladas em um sistema solar fotovoltaico:

- Elevada vida cíclica para descargas profundas;
- Necessidade de pouca ou nenhuma manutenção;

- Elevada eficiência de carregamento;
- Baixa taxa de auto-descarga;
- Confiabilidade;
- Mínima mudança no desempenho quando trabalhando fora da faixa de temperatura de operação.

Pelas características das baterias abordadas nesta seção, as baterias chumbo-ácido AGM do tipo seladas e “Deep-Cycle” são as mais recomendadas para serem utilizadas em sistemas fotovoltaicos. O fato de não necessitarem de manutenção periódica, a capacidade de suportarem muitos ciclos frequentes de carga e descarga, o menor custo e maior oferta no mercado são as características que justificam o seu uso.

As baterias de íons de lítio do tipo LiFePO_4 também são usadas em sistemas fotovoltaicos, porém com menor frequência em função de seu custo elevado.

3.5 MÓDULO ALIMENTADO

O módulo alimentado é a carga (dispositivo/equipamento) que consome a energia elétrica armazenada nas baterias e gerada pelo módulo fotovoltaico.

O consumo do módulo alimentado e o tempo de autonomia do sistema em dias de pouca insolação determinam o dimensionamento do sistema fotovoltaico (módulo fotovoltaico, armazenamento, controle de energia). A potência do módulo fotovoltaico, a tensão e a capacidade nominal da bateria, a tensão de operação e a corrente máxima de entrada e saída do módulo de controle são especificados em função do módulo alimentado.

A Tabela 3.4 apresenta um comparativo de potência (W) e consumo diário (Wh/dia), de quatro pontos de acesso testados no laboratório MídiaCom da Universidade Federal Fluminense no projeto ReMesh.

| Roteador | Consumo Corrente(A) | Tensão (V) | Potencia (W) | Wh/dia |
|----------------|---------------------|------------|--------------|--------|
| LinkSys WRT54G | 0.32 | 12 | 3.6 | 86.4 |
| Meraki Mini V2 | 0.4 | 7.5 | 3.0 | 72 |
| Bullet 2 | 0.22 | 12 | 2.6 | 62.4 |
| Fonera | 0.3 | 12 | 3.6 | 86.4 |

Tabela 3.4 - Consumo de alguns pontos de acesso

3.6 MÓDULO DE CONTROLE DE ENERGIA

O módulo de controle de energia, também chamado de controlador de carga, é o elemento mais complexo dos sistemas fotovoltaicos. Sua função central é controlar o processo de carga e descarga da bateria para evitar que esta sofra sobrecarga ou descarga profunda, que podem trazer danos irreversíveis à mesma.

Existem vários tipos de controladores de carga disponíveis. Alguns determinam o estado de carga da bateria integrando a corrente que está entrando ou saindo, ao longo do tempo. Outros indicam o estado de carga da bateria através da densidade⁴ do eletrólito [11]. O tipo mais comum estima o estado de carga medindo a tensão nos seus terminais. Neste caso, a tensão da bateria é permanentemente monitorada para que seja mantida entre os valores mínimo e máximo definidos no projeto do controlador. Quando a tensão máxima na bateria é alcançada, o controlador interrompe o fluxo de corrente elétrica gerada pelo módulo fotovoltaico. Por outro lado, o controlador desliga o módulo alimentado quando a tensão na bateria atinge o valor mínimo.

3.6.1 TIPOS DE CONTROLADORES DE CARGA

Além de se diferenciarem pela forma de como estimam o estado da carga na bateria como visto anteriormente, os controladores são diferenciados pelos dois tipos de configurações de construção física: série ou paralelo (shunt) e pelas estratégias ou algoritmos de controle de carga: “ON/OFF” ou TENSÃO CONSTANTE (CV).

CONTROLADORES EM SÉRIE

Nos controladores em série (figura 3.15), a chave que controla o fluxo de corrente gerada pelo módulo fotovoltaico (painel solar) fica em série com o módulo alimentado(carga).

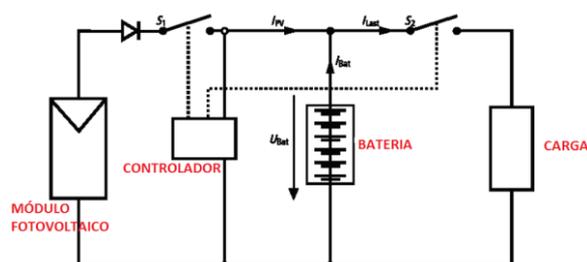


Figura 3.15 - Controlador série (GREENPRO, 2004)

⁴ Densidade - é a medida do peso específico do eletrólito comparado com o peso específico da água.

CONTROLADORES EM PARALELO

Nos controladores em paralelo (também chamados de *Shunt*), a chave fica em paralelo com o módulo fotovoltaico como ilustrado na Figura 3.16. Quando a bateria atinge o valor máximo de carga, a chave é fechada e como o módulo fotovoltaico continua a gerar energia elétrica, a corrente é curto-circuitada e passa a ser dissipada por um elemento do controlador de energia, denominado *Shunt*. Dessa maneira, o módulo fotovoltaico passa a operar em corrente de curto-circuito e a energia não aproveitada é dissipada em forma de calor.

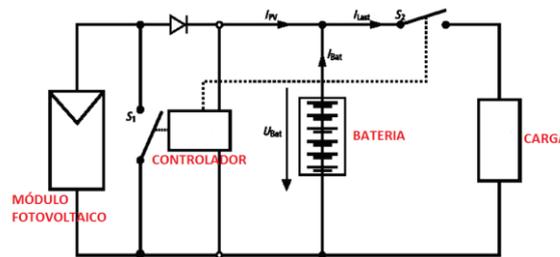


Figura 3.16 - Controlador Shunt (GREENPRO, 2004)

CONTROLADOR DE CARGA MPP (PONTO DE MÁXIMA POTÊNCIA)

O controlador de carga MPP (Ponto de Potência Máxima) é a tecnologia de controle dos sistemas fotovoltaicos mais moderna [11]. O ponto de máxima potência de um módulo fotovoltaico é ilustrada na curva da figura 3.8.

Quando os controladores série ou paralelo são utilizados, o módulo fotovoltaico não atinge o Ponto de Máxima Potência (MPP). Isto ocorre em função da tensão do módulo de armazenamento determinar o ponto de operação do sistema fotovoltaico, resultando assim, numa perda de 10% a 40% de energia.

Essa perda de energia pode ser evitada, utilizando um controlador de carga MPP que é responsável por avaliar a curva de operação do módulo fotovoltaico e determinar o ponto de máxima potência. Esse ajuste é efetuado por um conversor DC/DC, conforme ilustrado na Figura 3.17.

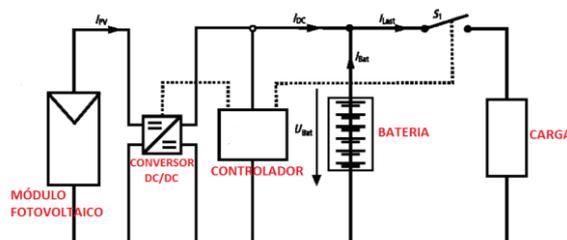


Figura 3.17 - Controlador MPP - (GREENPRO, 2004)

A eficiência do conversor DC/DC varia entre 90% a 96%. Por razões de eficiência, a utilização de um controlador MPP apenas faz sentido para módulos fotovoltaicos com potências superiores a 200 W. Para módulos fotovoltaicos de baixa potência, as perdas de conversão são geralmente maiores do que os ganhos obtidos com o controlador MPP. Devido à maior complexidade do circuito eletrônico, o preço do controlador de carga MPP é superior aos que não utilizam esta tecnologia de controle. Por razões econômicas, os controladores de carga MPP são usados com maior frequência em sistemas fotovoltaicos superiores a 500 W.

3.6.2 ESTRATÉGIAS DE CONTROLE

As estratégias de controle de carregamento de baterias mais comuns são “ON-OFF” e “TENSÃO CONSTANTE”.

ON-OFF

É a estratégia mais simples e mais barata. Utiliza chaves (Mosfet ou relé eletromecânico) para interromper o fluxo de corrente oriunda do módulo fotovoltaico para o acumulador quando este atinge o valor máximo ($V_{máx}$).

A partir desse momento, a bateria só voltará a ser carregada quando atingir o valor VRR. Esta estratégia de controle é ilustrada nas figuras 3.18 e 3.19.

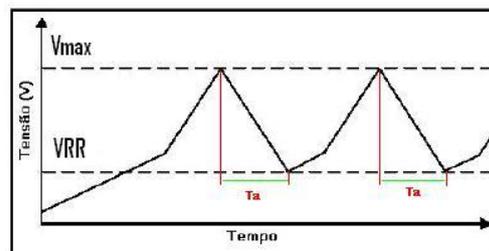


Figura 3.18 – Tensão na bateria durante a estratégia de controle “ON-OFF”

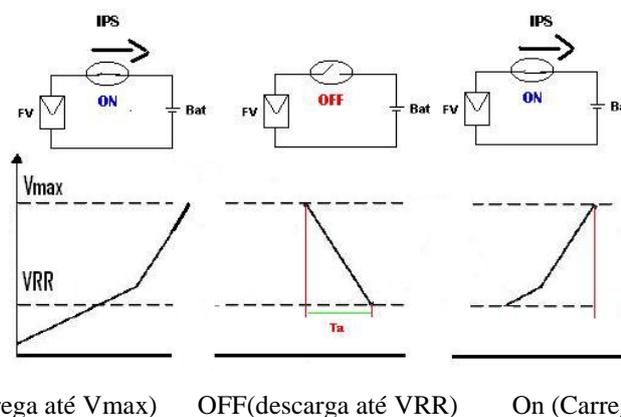


Figura 3.19 – Tensão na bateria durante a estratégia de controle “On-Off” (detalhado)

Como é possível observar pela figura 3.19, o acumulador não é carregado mesmo que o módulo fotovoltaico esteja gerando energia (intervalo T_a). Ainda nestes mesmos intervalos de tempo, a bateria estará sendo descarregada pois estará conectada ao módulo alimentado.

Dessa forma, nesses intervalos de tempo, o módulo de controle será totalmente ineficiente, pois estará desperdiçando parte da energia solar disponível durante o dia e ao mesmo tempo o módulo alimentado estará consumindo energia da bateria.

Essa estratégia apresenta dificuldade para carregar completamente a bateria, pois reduz a energia disponibilizada pelo módulo fotovoltaico de 10% a 15 % em comparação a um sistema sem controle [12].

TENSÃO CONSTANTE

Nesse tipo de estratégia, o controlador regula a corrente que carrega a bateria com o objetivo de manter a tensão na bateria no valor constante igual a tensão máxima ($V_{m\acute{a}x}$) suportada pela mesma. Existem dois tipos de estratégia de controle a tensão constante: controle linear e a PWM (Modulação por Largura de Pulso).

CONTROLE LINEAR

A estratégia de controle por tensão constante por controle linear pode ser efetuado em dois métodos: série ou paralelo.

No método série (figura 3.20), para manter a tensão constante na bateria, um reostato é colocado em série para dissipar a energia proveniente do módulo fotovoltaico quando a tensão na bateria atingir o valor máximo de carga ($V_{m\acute{a}x}$).

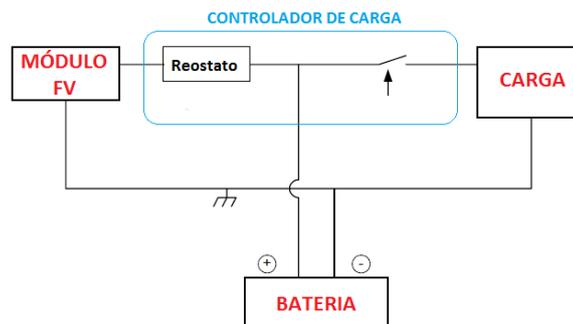


Figura 3.20 – Controle Linear Série: Reostato (IEA PVPS T3-05, 1998 / Versão modificada)

No método paralelo (figura 3.21), um diodo zener com tensão reversa igual à tensão $V_{m\acute{a}x}$ é colocado em paralelo. Quando a tensão na bateria atinge o valor máximo ($V_{m\acute{a}x}$), o

diodo Zener começa a conduzir e drena a corrente do módulo fotovoltaico. Isto mantém a tensão constante da bateria.

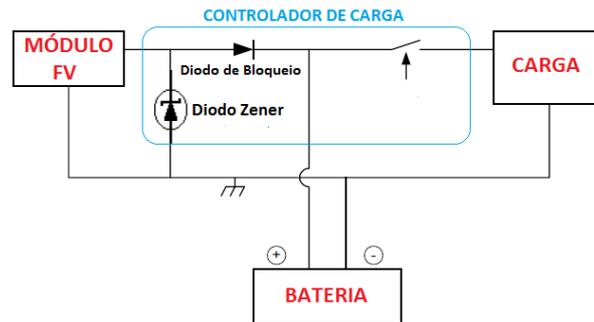


Figura 3.21 – Controle Linear Paralelo: Diodo Zener(IEA PVPST3-05,1998 / Modificada)

MODULAÇÃO POR LARGURA DE PULSOS(PWM)

Outro tipo de estratégia de controle a tensão constante (que pode ser implementada tanto em configuração série como em paralela) utiliza a técnica PWM - *Pulse Width Modulation*. Nesse tipo de estratégia, o controlador utiliza chaves de estado sólido (Transistor Mosfet) para aplicar pulsos de corrente em altas frequências (em torno de 300 Hz) para manter a tensão constante na bateria após a tensão na mesma atingir o valor máximo[16]. A figura 3.22 mostra um controlador com a técnica PWM em configuração série.

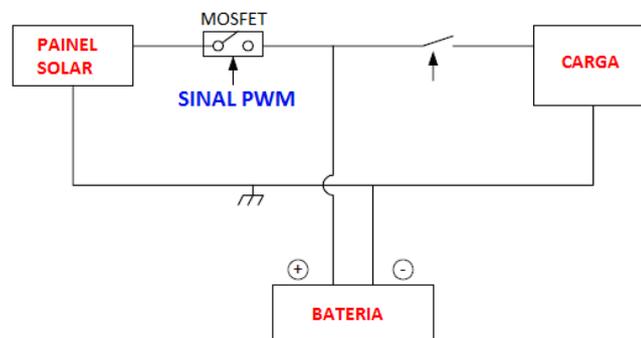


Figura 3.22 – Estratégia de carga PWM (IEA PVPS T3-05, 1998 / Versão modificada)

Apesar do período definido, esses pulsos de corrente possuem uma variação na largura (Duty-Cycle), resultando na variação do ciclo de operação como visto no gráfico 3.23.

Para carregar a bateria, uma forma seria utilizar dois estágios de carga. No estado “CHARGING”, utiliza-se um sinal PWM com uma máxima largura de DUTY-CYCLE para chavear o Mosfet fazendo com que toda a corrente do painel solar seja usada para carregar a bateria.

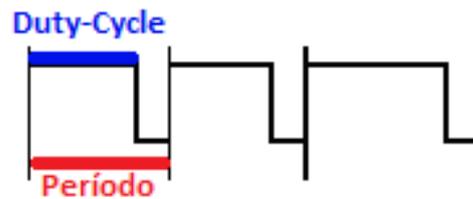


Figura 3.23 - Pulso PWM

Para evitar a sobrecarga da bateria quando a tensão da mesma alcançar o valor máximo suportado ($V_{m\acute{a}x}$), inicia-se um segundo estágio de carga chamado “FLOAT”, onde pulsos de corrente com largura menor e variável são aplicados a bateria. Este processo faz com que a bateria continue recebendo pulsos de corrente, porém em menores intervalos de tempo, o que resulta em continuar carregando a bateria com um fluxo bem menor de corrente o suficiente apenas para mantê-la com uma tensão constante em $V_{m\acute{a}x}$. A figura 3.24 ilustra dois estados do carregamento de bateria utilizando a técnica PWM.

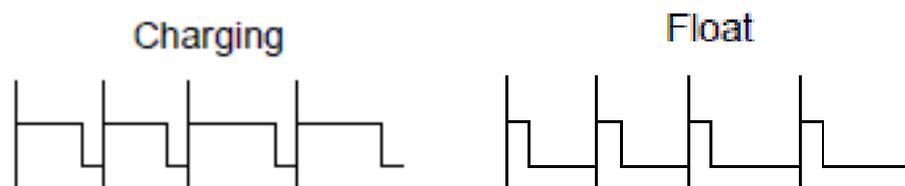


Figura 3.24 - Estados do carregamento a tensão constante por PWM

Apesar da complexidade maior exigida nesse tipo de estratégia, ela consegue uma redução na dissipação de potência comparada com a estratégia de Tensão Constante com sistema de controle linear. Outras vantagens da técnica PWM são: evita os efeitos da gaseificação, estratificação e o aquecimento interno da bateria; processo de carga rápido e eficaz; aumenta o ciclo de vida das baterias; recupera a capacidade das baterias em processo de sulfatação; reduz o envelhecimento e a formação de gases; equalização das células; prevenção contra perda de capacidade (pulsos minimizam o processo de corrosão); elevação constante (em média de 90%) do estado da carga do acumulador e aumento da energia gerada pelo módulo fotovoltaico em cerca de 20 a 30%.

A tabela 3.5 resume as vantagens e desvantagens das estratégias de controle dos controladores de carga citadas nesta seção.

| Configuração | Estratégia de controle | Vantagens | Desvantagens |
|-------------------|--------------------------------------|---|---|
| Série | On-Off | - Pouca dissipação de potência - Simples, barato e confiável. | -Dificuldade para carregar completamente a bateria |
| Série | Tensão constante | - Corrente de carga suave | -Requer dissipação de potência -Queda de tensão pelo controlador |
| Paralela | On-Off | - Simples barato, robusto. - Menor queda de tensão pelo controlador | -Significativa dissipação de potência no elemento de chaveamento; -Diodo de bloqueio; -Pode causar pontos quentes em arrays de altas tensões -Dificuldade para carregar completamente a bateria em altas correntes |
| Paralela | Tensão Constante com controle linear | - Corrente de carga suave -Menor queda de tensão através do controlador | -Significativa dissipação de potência no chaveamento -Diodo de bloqueio -Pode causar pontos quentes em arrays de altas tensões |
| Série ou Paralela | Tensão Constante a PWM | - Corrente de carga suave - Baixa dissipação de potência - Menor dissipação de potência que outros métodos a Tensão Constante | - Geralmente mais complexo que controladores On/Off - Às vezes causa interferência eletromagnética em equipamento próximo |

Tabela 3.5–Vantagens e desvantagens das estratégias de controle (IEA PVPS-1998 adaptado)

3.6.3 Baterias de chumbo-ácido, NiCd e controladores em sistemas fotovoltaicos

Nesta seção são apresentadas algumas pesquisas relacionadas a elementos utilizados em sistemas fotovoltaicos como: baterias chumbo-ácido de GEL, AGM e fluidas, estratégias de controle de controladores de carga e algoritmos de controle de carga para baterias de NiMh utilizadas com painel solar.

O comportamento de baterias de chumbo-ácido de GEL e AGM reguladas à válvula (VRLA) em aplicações de energia solar com diferentes estratégias de controle foram investigadas na pesquisa realizada [39] em vários sistemas fotovoltaicos. Cada sistema foi dividido em cerca de 26 grupos (com mais de 200 módulos solares) e cada grupo tinha o mesmo tipo de bateria, a mesma carga e o mesmo gerador solar. A única diferença estava na estratégia de carga/descarga. A maioria dos testes foram realizados no campo para serem tão próximos possíveis da aplicação real. No entanto, foram completados por alguns testes de laboratório com parâmetros e condições bem definidos. Esta foi a primeira vez que, para

aplicações de energia solar, a influência da estratégia de carga/descarga sobre a vida útil e o desempenho de baterias do tipo VRLA foi investigada sistematicamente em campo. Depois de 2 anos (1998 a 2000) de testes, os resultados indicaram que existiam muitas diferenças na capacidade disponível e também desvios nas capacidades restantes da carga da bateria, indicando que as estratégias de carga utilizadas na época não conseguiam carregar as baterias completamente. No entanto, as baterias teriam sua capacidade total após uma carga intensiva. Isto significa que existe um problema claro de “undercharging” com os regimes de carga para a aplicação de energia solar. Para superar este problema, as estratégias de carga e descarga necessitariam ser melhoradas.

No artigo “*Evaluation of the batteries and charge controllers in small stand-alone photovoltaic systems*”[44] são relatados os resultados de testes realizados com o objetivo de avaliar o comportamento das baterias de chumbo-ácido fluídas e controladores de carga utilizados em sistemas fotovoltaicos de pequeno porte. Os sistemas foram fortemente instrumentados para ajudar a determinar quais parâmetros afetam o estado de carga, durabilidade e perda de água das baterias e a confiabilidade do sistema. Foram feitos diferentes testes de longa duração (cerca de 2 anos) com baterias e controladores de carga em quatorze sistemas fotovoltaicos. Os pesquisadores acreditavam que sob condições apropriadas, as baterias de chumbo-ácido fluidas destes sistemas fotovoltaicos poderiam alcançar ciclos de vida comparáveis às expectativas do fabricante para aplicações de baterias convencionais. Dentre os vários testes realizados, podem ser destacados com relação ao controlador que: (1) o estado de carga da bateria está mais correlacionada com o valor da tensão em que o controlador reconecta o módulo fotovoltaico à bateria para reiniciar o processo de carregamento com corrente máxima do que com a tensão máxima atingida pela bateria em qualquer ciclo; (2) para os sistemas de fotovoltaicos que passam uma quantidade significativa de tempo numa situação de déficit de energia, aumentando o valor do ponto de corte de baixa tensão (que desconecta o dispositivo usado como carga), melhora significativamente a vida útil da bateria, porém diminui o tempo de disponibilidade de energia da bateria; e (3) os controladores de carga que operam em alta frequência ou num modo de limitação de corrente pode gerar calor internamente durante o ciclo de regulação e, em alguns casos, isto pode elevar a temperatura do controlador de carga 30°C acima da temperatura ambiente. Portanto, as sondas de compensação de temperatura externa colocadas em contato diretamente com as baterias devem ser utilizadas, evitando as sondas de temperatura de compensação localizadas dentro do controlador. Para a perda de água das baterias testadas conclui-se que: (1) a perda de água está diretamente relacionada com o estado da carga da

bateria. Em geral, quanto maior for o estado de carga, maior é perda de água. Isto deve ser previsto no projeto do sistema ou planos de manutenção; (2) a perda de água da bateria variou por um fator de 2 a 3 do verão para o inverno. Isto pode ser explicado pela reação de gaseificação de sobre-tensão e variações nas temperaturas da bateria nestas estações; e (3) a compensação de temperatura dos pontos de ajuste do controlador pode reduzir o tamanho do ciclo de perda de água verão-a-inverno, porém não reduziu a perda de água global do sistema para os sistemas testados. Deve notar-se contudo, que a compensação de temperatura também é esperado para reduzir a corrosão de grade em tempo quente e ajudar a manter um estado de carga elevado em tempo frio.

Já no artigo “*Solar Battery Chargers for NiMH Batteries*” [45] são apresentados novos algoritmos de controle de carga adequados para carregar baterias NiMH com painéis solares portáteis. Segundo os autores, os novos algoritmos são robustos, confiáveis e superam as falhas dos controladores de carga para baterias NiMH que ocorrem quando há mudanças nas condições ambientais. Enquanto um algoritmo baseia-se na tensão da bateria, um segundo algoritmo baseia-se no derivado de temperatura. A característica comum para os dois algoritmos é que eles incluem mecanismos especializados de reset para eliminar o falso indicativo de final de carga que ocorre devido à mudança de condições de iluminação, oscilação na temperatura ou outras mudanças no ambiente. Qualquer um destes dois algoritmos podem ser utilizados separadamente, mas podem também ser combinados para melhorar a robustez do sistema. Para aumentar a velocidade do carregamento na bateria, é implementado um rastreador de ponto de máxima potência máxima no controlador proposto. A utilização de um microcontrolador no controlador de carga proposto ao invés de um circuito integrado dedicado implica no aumento do custo resultante, porém pode ser compensada pela adição de recursos extra, como por exemplo o controle de MPPT citado. Embora o carregador proposto seja projetado para baterias de NiMH, também pode ser usado diretamente para baterias de NiCd ajustando os limites dos pontos de operação.

3.6.4 DETALHAMENTO DAS CARACTERÍSTICAS E FUNÇÕES DO CONTROLADOR

O mínimo necessário para se especificar um controlador são os valores de corrente máxima (maior que a máxima corrente de curto-circuito do módulo fotovoltaico) e a tensão nominal de operação do sistema [11]. Outras características são apresentadas na tabela 3.6.

| | |
|---|---|
| Tipo de configuração | Série ou Paralelo |
| Estratégia de controle | “ON-OFF”, Tensão Constante ou MPP. |
| Tensão Nominal | 12 V, 24V ou 48V. |
| Corrente máxima da carga | A corrente máxima que a carga vai consumir deve ser levada em consideração no projeto do controlador de carga. |
| Número de estágios de carga | 1 estágio, 2 estágios ou mais estágios. |
| Pontos de ajuste | V_{máx} – Tensão máxima de carga. Quando este valor é atingido, o controlador de carga interrompe o processo de carga para evitar a sobrecarga da bateria. V_R – Tensão mínima para que o controlador continue o processo de carga da bateria (para estratégia ON-OFF). V_{mín} – Tensão mínima de desconexão. Quando este valor é atingido, o controlador de carga desconecta a carga que está sendo alimentada para evitar a descarga profunda da bateria. VC - Tensão mínima de conexão. Quando este valor é atingido, o controlador de carga conecta a carga que está sendo alimentada. |
| Tipo de bateria | Os pontos de ajustes do controlador são definidos em função do tipo de bateria que será utilizada no controlador. |
| Compensação térmica | As características de carregamento das baterias mudam com a variação da temperatura [11]. O aumento da temperatura provoca o aumento da reação eletro-química na bateria gerando sobrecarga. Para evitar isto, a tensão de carga deve ser reduzida. Por outro lado, quando a temperatura diminui, a tensão de carga deve ser aumentada para evitar que a bateria sofra uma carga parcial. A compensação térmica otimiza a vida útil da bateria e faz-se mais necessária quando a faixa de temperatura de operação das baterias excede 5°C a temperatura ambiente padrão de 25°C. A compensação pode não ser necessária no caso de intervalos breves de variação de temperatura. Para uma bateria chumbo-ácido de 12 Volts de tensão nominal recomenda-se o fator de compensação de aproximadamente -30mV/°C. Como exemplo, para uma bateria instalada num local onde a temperatura permaneça por longo tempo em 35 °C, o que resulta numa variação de 10°C, a tensão de carga deve ser reduzida de 0,3 Volts. |
| Proteção contra inversão de polaridade | circuito eletrônico utilizado para evitar danos aos componentes do controlador de carga caso haja inversão na polaridade da bateria ou painel solar. |
| Indicadores de Status | Alguns dispositivos luminosos ou sonoros podem ser utilizados para indicar alguns status do controlador. Como por exemplo, um led ou buzzer pode indicar quando a bateria está em baixo nível de carga. |
| Controle de mais de uma carga | O CC pode ser projetado para operar mais de uma carga. Essas cargas são chamadas de principal e secundária, que podem ser acionadas de forma totalmente independente. |
| Desvio da energia do painel | Quando a bateria atinge a plena carga, o controlador interrompe (método “On-Off”) ou reduz (método Tensão Constante) o fluxo de corrente para a bateria. Neste caso, o excesso de energia (que seria perdido) do painel pode ser utilizado para ligar cargas secundárias não essenciais como por exemplo, ventilador de resfriamento, sensores diversos ou outros dispositivos. |
| Proteção contra corrente reversa | Para impedir o fluxo de corrente da bateria para o módulo fotovoltaico durante a noite, quando a tensão de circuito aberto do arranjo é inferior à tensão da bateria. |
| Baixo consumo interno | |

Tabela 3.6 – Características de um controlador de carga

3.7 – MICROCONTROLADORES

O microcontrolador é o agrupamento de vários componentes em um sistema microprocessado [17]. Basicamente o microcontrolador é um microprocessador com memória RAM e de programa, temporizadores e circuitos de clock embutidos. O único componente externo que pode ser necessário é um cristal para determinar a frequência de trabalho.

Um microprocessador de uso geral integra vários elementos adicionais em sua estrutura interna como: gerador interno independente de clock; memória SRAM, EEPROM e FLASH; conversores A/D, D/A; vários temporizadores/contadores; comparadores analógicos; PWM; diferentes tipos de interface de comunicação, incluindo USB, USART, I2C, SPI,

JTAG; relógios de tempo real; circuitos para gerenciamento de energia no chip; circuitos para controle de reset, alguns tipos de sensores; interface para LCD; e outras funcionalidades de acordo com o fabricante.

A grande vantagem dos microcontroladores com relação aos processadores é que eles podem ser programados para funções específicas e já incorporam a maior parte do hardware necessário para sistemas embarcados.

Os microcontroladores são embarcados no interior de algum outro dispositivo (geralmente um produto comercializado) para que possam controlar as funções ou ações específicas do produto. Dessa forma, também são denominados controlador embutido.

Apesar de operarem a uma frequência muito baixa (clock de alguns Megahertz) se comparados com os poderosos microprocessadores atuais, os microcontroladores são adequados para diversas aplicações que não exigem muito processamento.

Com relação ao consumo de energia, em geral é relativamente pequeno, normalmente na casa dos miliWatts. Além disso, os modos de *sleep* permitem que alguns elementos internos do microcontrolador sejam desligados reduzindo ainda mais o consumo. Isto torna o microcontrolador uma solução ideal para aplicações onde a exigência por baixo consumo de energia é um fator importante no projeto, como sistemas alimentados por baterias.

3.7.1 VANTAGENS DA UTILIZAÇÃO DOS MICROCONTROLADORES

A utilização dos microcontroladores possibilita uma série de vantagens, como:

- **Custo reduzido:** geralmente dispensam diversos outros componentes;
- **Tamanho reduzido:** como microcontroladores possuem vários circuitos dentro de um único chip, isso permite a confecção de placas com pouquíssimos componentes de apoio.
- **Facilidade de desenvolvimento:** apesar de ser necessário certo conhecimento de lógica digital e de programação, desenvolver um projeto utilizando microcontrolador é em geral mais rápido e fácil.
- **Facilidade de manutenção:** como os circuitos microcontrolados são menores que os circuitos discretos e utilizam menos componentes, sua manutenção se torna mais ágil.
- **Facilidade de modificação:** por se tratar de um software interno a modificação de características do projeto é muito mais rápida.
- **Baixo consumo:** para utilização em sistemas onde se deseja o máximo de economia de energia, seu uso é recomendado pois possui baixo consumo de energia.

3.8 CONTROLADORES DE CARGA MICROCONTROLADOS

Os controladores de carga microcontrolados possibilitam inúmeras vantagens e recursos adicionais como citado na seção 2.3.

A utilização de microcontroladores ou microprocessadores em controladores de carga faz com que esses sejam mais inteligentes devido aos algoritmos de regulação que podem ser implementados [18].

O monitoramento dos parâmetros elétricos do sistema como tensão e corrente pode ser feito de forma mais precisa graças aos conversores analógico-digital integrados nos microcontroladores.

Existem no mercado controladores de carga que informam a geração e o consumo do sistema fotovoltaico em Ampere-hora (Ah) numericamente em um display LCD, o que permite ao usuário ter uma visibilidade maior do que está acontecendo no sistema.

Atualmente existem controladores que permitem que o usuário tenha acesso às condições operacionais do sistema e da bateria através de interface de comunicação serial RS232⁵.

O ponto de máxima potência (MPP) pode ser monitorado através de um dispositivo eletrônico e estratégias como “Perturbe e Observe” (P&O) e “Incremento da condutância” podem ser utilizadas para deslocar o ponto de operação do módulo fotovoltaico, atingir a máxima potência e com isso aumentar a eficácia do módulo [18].

3.8.1 RECURSOS DOS CONTROLADORES DE CARGA MICROCONTROLADOS

Devido aos vários elementos integrados no microcontrolador, os controladores de carga que utilizam este dispositivo podem ter diversos recursos como:

- **Pontos de ajustes programáveis:** o microcontrolador pode ser programado com diferentes pontos de ajustes. Isto permite a utilização de mais de um tipo de bateria.
- **Monitoramento remoto:** A utilização de interface de comunicação presente em determinados microcontroladores (Serial ou USB) ou a integração de um circuito com interface Ethernet ao microcontrolador permite o monitoramento de parâmetros elétricos do sistema fotovoltaico como: tensão da bateria, tensão e corrente no painel solar.

⁵ RS232: padrão para troca serial de dados binários entre um DTE (terminal de dados, de Data Terminal equipment) e um DCE (comunicador de dados, de Data Communication equipment).

- **Gerenciamento remoto:** Outro recurso que pode ser implementado através da interface de comunicação é o gerenciamento remoto. Desta forma, o controlador de carga é programado para executar comandos enviados pelo dispositivo ao qual está conectado. Como exemplo, um comando pode fazer com que a carga seja desligada por um determinado período de tempo, com o objetivo de economizar energia armazenada nas baterias.
- **Modos de operação programados:** Os controladores de carga comerciais geralmente só suportam um tipo de modo de operação, por exemplo, manter a carga ligada de forma ininterrupta até a bateria descarregar até um determinado valor pré-estabelecido. O controlador de carga microcontrolado pode ser programado com vários modos de operação, como por exemplo, manter a carga ligada durante apenas algumas horas do dia.
- **Indicadores de status por display LCD:** Para a indicação do status do controlador e visualização dos parâmetros elétricos do sistema, um display LCD (Figura 3.25) pode ser ligado ao microcontrolador.



Figura 3.25 – Display LCD

CAPÍTULO 4 – PROTÓTIPO DO CONTROLADOR DE CARGA MICROCONTROLADO

Este capítulo descreve o protótipo do controlador de carga microcontrolado (CCM) para redes em malha alimentadas por energia solar. São apresentados os requisitos e as funcionalidades do protótipo desenvolvido neste trabalho. Também são descritos os principais componentes eletrônicos utilizados na montagem do protótipo.

4.1 – REQUISITOS E FUNCIONALIDADES

O controlador de carga apresentado neste trabalho deve atender aos seguintes requisitos: coletar informações, permitir atuação sobre o sistema, aceitar modos de economia de energia e ser flexível. A tabela 4.1 apresenta as funcionalidades existentes no CCM em função dos requisitos e na seção 4.4 são descritas essas funcionalidades.

| Requisitos | Funcionalidades |
|-------------------------|--|
| Coleta de informações | - Monitoramento dos parâmetros elétricos do kit de alimentação solar. |
| Atuação sobre o sistema | - Execução de comandos remotos. |
| Economia de energia | - 2 modos de funcionamento: padrão e econômico; - Desligar cargas primária e secundária; - Desligar leds indicadores de status. |
| Flexibilidade | - Possibilidade de selecionar mais de um tipo de bateria; - Utilização de duas cargas independentes(primária e secundária); - Função WatchDog Timer(reinicializar roteador); |

Tabela 4.1 – Requisitos e funcionalidades do CCM

4.2 - DESCRIÇÃO GERAL DO SISTEMA

A figura 4.1 apresenta a visão geral de uma rede *mesh* onde os nós integram um sistema fotovoltaico. O roteador é interconectado ao CCM através da interface serial (figura 4.2). Isto permite que os dados dos sensores utilizados no controlador cheguem ao servidor após passarem pelo gateway da rede. Dessa forma, o supervisor da rede *mesh* monitora os parâmetros elétricos do sistema fotovoltaico.

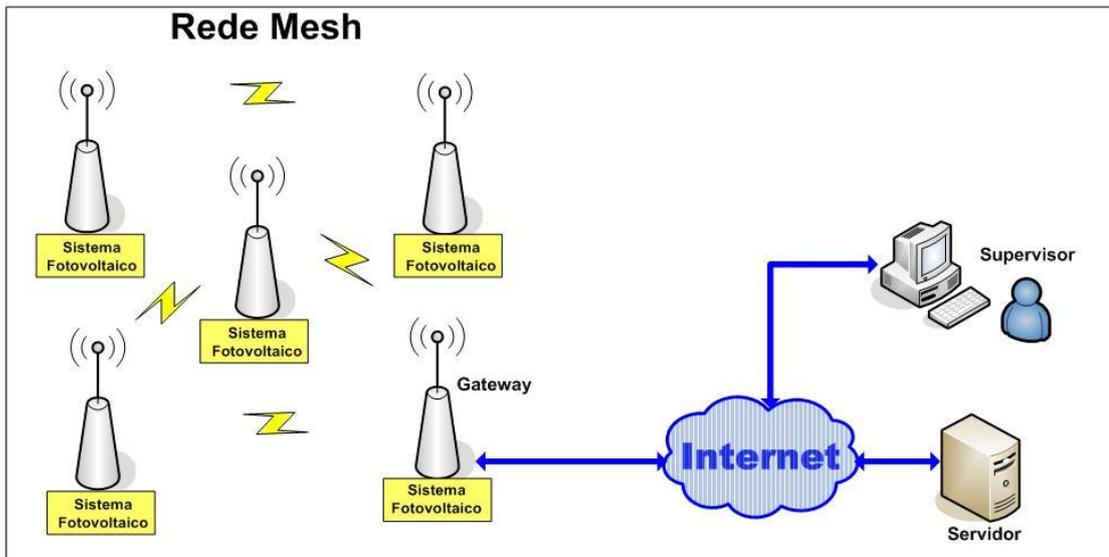


Figura 4.1 – Visão geral de uma rede *mesh* com sistema fotovoltaico Isolado

Além disso, o supervisor pode atuar remotamente sobre o CCM. Isso pode ser feito através de bytes enviados pelo roteador. Para que o CCM execute um comando remoto, primeiro é preciso se conectar ao roteador e depois enviar o byte relacionado ao comando que se deseja executar através da porta serial. Na seção 4.4 são apresentados os comandos que podem ser executados remotamente.

A figura 4.2 ilustra os elementos que compõem o nó *mesh*: painel solar policristalino de 40 W, 3 baterias seladas de chumbo-ácido de 12V/7Ah em paralelo (resultando em 21 Ah de capacidade nominal), roteador LinkSys WRT54G e protótipo do CCM.

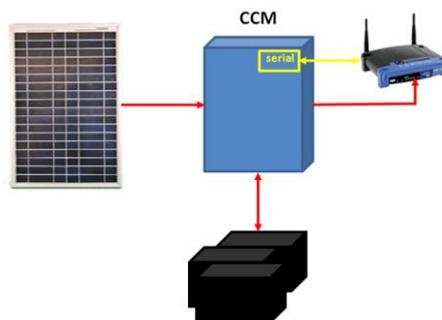


Figura 4.2 – Elementos do sistema fotovoltaico

4.3 DIAGRAMA ESQUEMÁTICO DO CCM

A figura 4.3 apresenta o diagrama esquemático do CCM. Como pode ser visto, o controlador é formado basicamente por sensores (V_{bat} = tensão na bateria, V_{ps} = tensão no painel solar, I_{ps} = Corrente no painel solar, Temp = temperatura e Lum = luminosidade), 3 chaves (ativação das cargas primária/secundária e chaveamento painel solar – bateria), relógio

I2C e comunicação serial com roteador. Os elementos que compõem o CCM são descritos na seção 4.5.

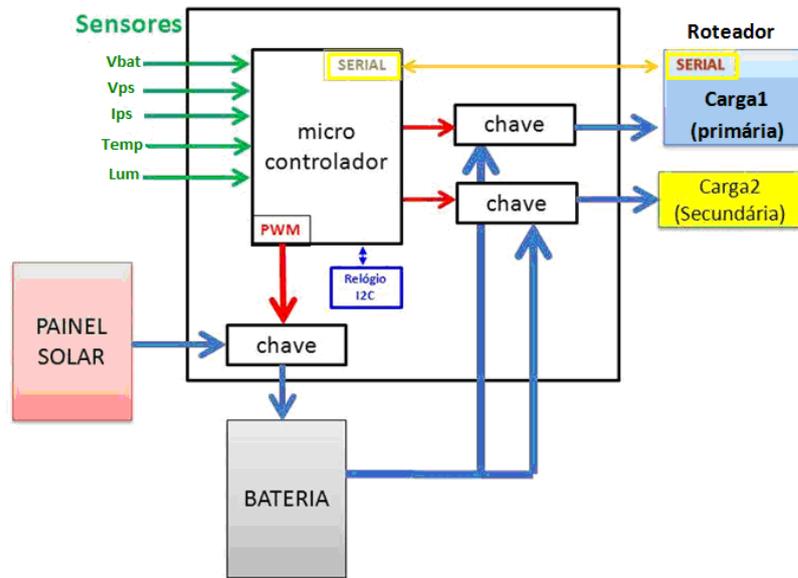


Figura 4.3 – Diagrama Esquemático do CCM

4.4 ESPECIFICAÇÃO DO CCM

A tabela 4.2 apresenta as especificações do CCM com as respectivas características.

| | |
|--|--|
| Tipo de configuração | Série. Evita que a corrente não utilizada seja dissipada em forma de calor. |
| Estratégia de controle | Tensão Constante a PWM. Como visto na seção 3.6.2.2.2, além de ser uma técnica mais eficiente no carregamento de baterias, possui inúmeras vantagens em comparação com outras estratégias de controle. |
| Estágios de carga | BULK e FLOAT. No estágio BULK, é utilizado um sinal PWM com máxima largura de DUTY-CYCLE para que toda corrente do painel solar seja usada para carregar a bateria. Após a tensão na bateria chegar ao limiar de valor máximo ($V_{m\acute{a}x}$), inicia-se o estágio FLOAT para que a bateria receba um fluxo bem menor de corrente o suficiente apenas para mantê-la na tensão constante em $V_{m\acute{a}x}$. |
| Especificações nominais | A tensão nominal é de 12 V. A corrente máxima de saída é 3 A. |
| Proteção contra reversão de polaridade da bateria | Para evitar danos ao controlador caso haja reversão de polaridade na ligação da bateria. |
| Número de Saídas | Duas, sendo uma para carga primária (roteador) e uma para carga secundária. |
| Tipo de Bateria configurável | O CCM é programado para 2 tipos de baterias (Chumbo-ácido e $LiFePo_4$), selecionada através de uma chave (figura 4.4) de 2 posições. |
| Pontos de ajustes selecionáveis | De acordo com o tipo de bateria selecionada, os pontos de ajustes são selecionados automaticamente pelo CCM e a tabela 4.3 ilustra os valores para cada tipo de bateria. |
| Modo de Operação selecionável | O CCM é programado para funcionar em dois modos de operação: Modo Padrão: Encontrado normalmente nos controladores: a carga só é desligada quando a bateria alcança a tensão mínima. Este modo não permite economia de energia, pois a carga fica ligada independente de estar sendo utilizada. Modo Econômico: O roteador é ligado apenas no horário das 07:00 às 17:00 h. Este modo é econômico, pois mantém o roteador desligado nos períodos em que a rede <i>Mesh</i> normalmente não é utilizada, como nos períodos noturnos. Independente do modo de operação, o CCM desligará a carga quando a tensão da bateria alcançar a tensão V_{min} . Os modos de operação podem ser alternados de duas maneiras: (1) através de uma chave (figura 4.4) montada na placa do CCM e (2) através da execução de comandos remotos (seção 4.6.2). |

Tabela 4.2 - Especificações do CCM

| Pontos de ajuste | Chumbo-ácido | LiFePO ₄ |
|------------------|--------------|---------------------|
| V _{max} | 14,4 V | 14,2 V |
| V _{min} | 11,5 V | 11,4 V |
| VC | 12,6 V | 12,6 V |

Tabela 4.3 – Pontos de ajuste para baterias chumbo-ácido e LiFePO₄

4.5 DESCRIÇÃO DO HARDWARE DO CCM

A figura 4.4 ilustra a placa protótipo do CCM com a indicação dos elementos que a compõe.

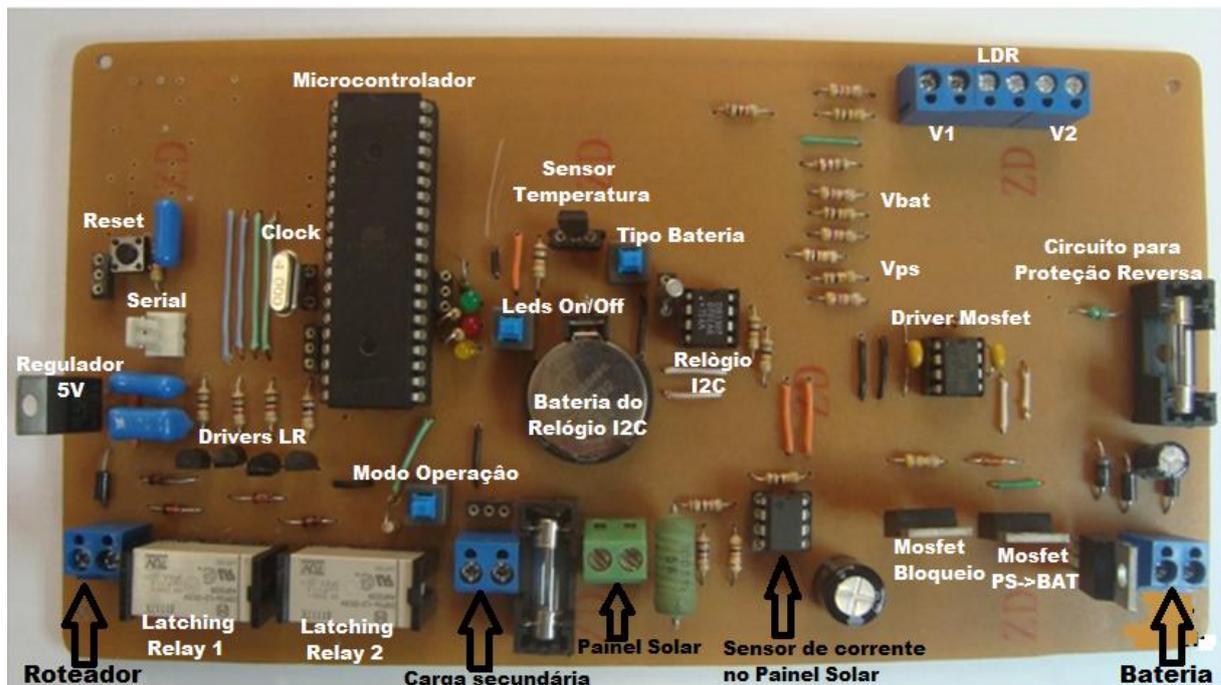


Figura 4.4 – Placa protótipo do CCM

O CCM é dividido em 4 blocos: alimentação, controle, sensores e chaveamento. Cada um desses blocos é apresentado nas próximas seções com o respectivo esquema elétrico. São abordadas algumas características dos principais componentes/dispositivos eletrônicos utilizados no controlador, porém para obter dados completos do componente deve ser consultado o *Data Sheet* do mesmo.

4.5.1 BLOCO DE ALIMENTAÇÃO

REGULADOR DE TENSÃO LM7805

O principal elemento do bloco de alimentação (figura 4.5) é o regulador de tensão LM7805 [28]. Permite uma faixa de tensão na entrada de 7 a 25 V, fornecendo uma tensão de saída de 5V e corrente de até 1 A (desde que seja utilizado um dissipador adequado). A tensão de saída é responsável pela alimentação de vários dispositivos: microcontrolador, sensor de temperatura, amplificador operacional (sensor de corrente), relógio I2C, Drivers do Mosfet e do Latching Relay. O bloco regulador de tensão possui capacitores para filtragem e desacoplamento.

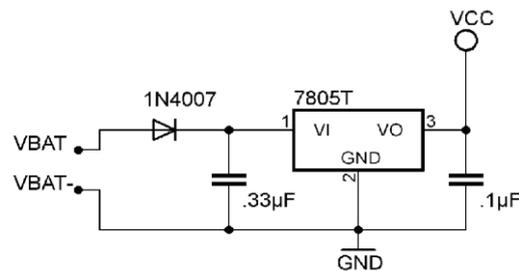


Figura 4.5 – Circuito regulador de tensão

PROTEÇÃO CONTRA REVERSÃO DE POLARIDADE DA BATERIA

O CCM possui um circuito de proteção (figura 4.6) contra reversão de polaridade da bateria composto de fusível (F_p) e diodo (D_p). Se a bateria for ligada com a polaridade invertida, o fusível é interrompido e a bateria é desconectada do circuito, evitando danos e possível queima de componentes eletrônicos do controlador.

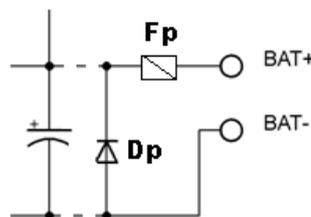


Figura 4.6 – Circuito de proteção contra reversão de polaridade

4.5.2 BLOCO DE CONTROLE

O bloco de controle é o mais complexo, pois é nele que se faz todo o monitoramento da tensão da bateria e do painel solar, comparação com os pontos de ajuste e onde são tomadas decisões para controlar os circuitos de chaveamentos.

• MICROCONTROLADOR ATMEGA16

O principal elemento do bloco de controle é o microcontrolador Atmega16 desenvolvido pela Atmel (www.atmel.com). O Atmega16 é um microcontrolador CMOS¹ de 8 bits com baixo consumo baseado na arquitetura AVR RISC². Executando um poderoso jogo de instruções de um ciclo, este microcontrolador consegue uma capacidade de processamento cerca de um 1 MIPS³ por MHz, proporcionando grande eficiência na comunicação de dados, permitindo ao projetista de um sistema otimizar o consumo graças à grande velocidade de processamento [29].

Os microcontroladores de tecnologia AVR tem 32 registradores de 8 bits de uso geral e qualquer um deles pode conter endereços ou dados. Os 32 registradores estão conectados diretamente à unidade lógica aritmética (ALU), permitindo que dois registradores independentes sejam acessados mediante uma instrução simples executada em um só ciclo de instrução.

A arquitetura resultante é muito eficiente permitindo uma capacidade de processamento aproximadamente 10 vezes superior aos microcontroladores CISC⁴ convencionais.

A arquitetura RISC (Reduced Instruction Set Computer) tem operações primitivas mais básicas do que a arquitetura CISC, facilitando a criação de sistemas mais rápidos no número de instruções executadas por segundo, já que cada instrução é, em média, mais simples do que da arquitetura CISC. Exatamente por isso, a utilização da arquitetura RISC é uma grande tendência da indústria de microcontroladores: uma vez que há um pequeno número de instruções a serem manipuladas. Os benefícios em se utilizar uma arquitetura RISC são chips menores, número menor de pinos, e considerável redução do consumo de energia por parte do processador.

¹ CMOS: Complementary Metal-Oxide-Semiconductor

² RISC: Reduced Instruction Set Computer (Computadores com um Conjunto Reduzido de instruções)

³ Milhões de Instruções por Segundo, é a medida de frequência de execução de uma dada instrução em uma máquina específica

⁴ CISC: Complex Instruction Set Computer (Computadores com um Conjunto Complexo de Instruções), não é de interesse ser aqui abordada, uma vez que foge ao escopo deste projeto

As características básicas do Atmega16 são:

- Arquitetura RISC avançada;
- 131 instruções, as mais simples são executadas em um só ciclo de clock;
- 32 x 8 registradores de trabalho de uso geral;
- Até 16 MIPS em 16 MHz;

Memória de dados e programa não volátil:

- 16K bytes de memória FLASH;
- 512 Bytes de memória EEPROM;
- 1K Bytes de memória SRAM;
- Interface de testes e programação JTAG;
- Programação de Flash, EEPROM, fusíveis e Bits de bloqueio pela interface JTAG;

Recursos de periféricos:

- 2 temporizadores/Contador de 8 bits;
- 1 Temporizador/Contador de 16 bits;
- 4 canais PWM;
- 8 canais conversor analógico-digital de 10 bits;
- Watchdog Timer programável com oscilador On-Chip;
- Comparador Analógico On-Chip;
- Contador em Tempo Real com Oscilador separado;
- Gravação através da interface serial USART;
- Interface serial orientada a byte de dois fios;
- Suporte a barramento I2C;
- Interface SPI;

Recursos especiais:

- 32 linhas de entrada/saída programáveis;
- Fontes de interrupção internas e externas;

Tensões de operação:

- Atmega16L: 2,7 – 5,5 V
- Atmega16: 4,5 – 5,5 V

Clocks:

- 0 – 8 MHz p/ Atmega16L;
- 0 – 16 MHz p/ Atmega16;

Seis modos de SLEEP:

Idle, ADC Noise Reduction, Power Save, Power Down, Standby e Extended Standby.

Consumo de energia @ 1 MHz, 3V, e 25 ° C para ATMEGA16L

Ativo: 1,1 mA

Modo Idle: 0,35 mA

Modo de baixo consumo: < 1 mA

A figura 4.7 mostra as duas formas de encapsulamento em que o Atmega16 é disponibilizado: PDIP (*Plastic Dual Inline Package*) e TQFP/MLF (*Thin profile plastic Quad Flat Package*). A descrição funcional de cada um dos pinos pode é apresentada no ANEXO I.

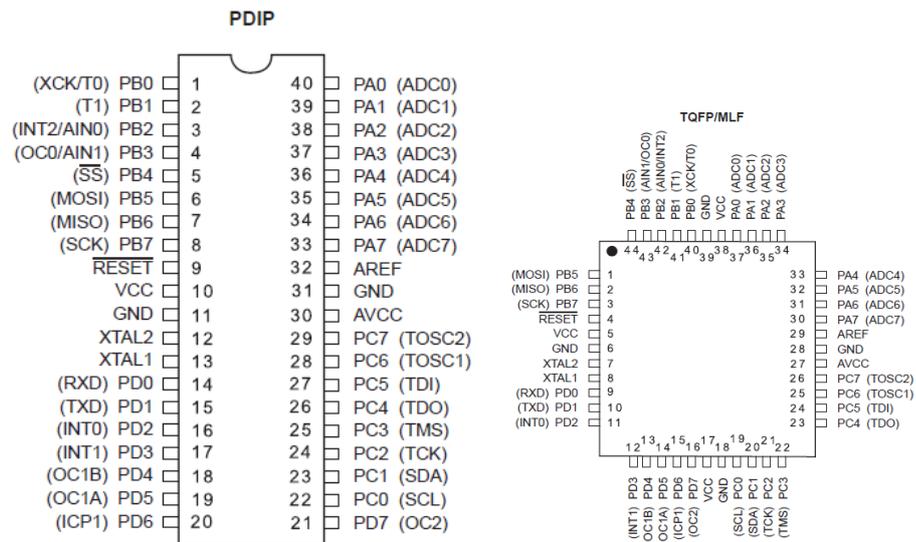


Figura 4.7 – Encapsulamentos do Atmega16

Alguns recursos do microcontrolador Atmega16 serão descritos com mais detalhes a seguir.

- INTERRUPÇÕES

As interrupções são modificações assíncronas ocorridas no fluxo de controle e são muito importantes porque permitem simplificar o código e tornar mais eficiente o desempenho do processamento. O exemplo clássico é o teclado de computador: o processador não fica monitorando se alguma tecla foi pressionada, quando isso ocorre, o teclado gera um pedido de interrupção e a CPU interrompe o que está fazendo para executar a tarefa necessária. Assim, não há desperdício de tempo nem de processamento[17].

De modo geral, existem três fontes de interrupção: a interrupção por software (instrução), a pedida por periférico externo e a interrupção pedida por periférico interno (Timer/Contador, porta serial, dentre outros). O Atmega16 possui 13 interrupções diversas e todas são “vetoradas”, ou seja, cada interrupção tem uma posição fixa na memória de programa. Há registradores de I/O específicos para controlar o nível, a prioridade e a habilitação de cada interrupção.

- CONVERSOR ANALÓGICO-DIGITAL

O Atmega16 possui um conversor A/D que emprega o processo de aproximações sucessivas para converter um sinal analógico em digital. Suas principais características são:

- 10 bits de resolução (1024 pontos).
- Precisão de $\pm 2 \text{ LSB}^5$.

- Tempo de conversão de 13-260 μ s.
- Até 15 kSPS na resolução máxima.
- Faixa de tensão de entrada de 0 até VCC (A_{REF}).
- Tensão de referência selecionável de 2,56 V.
- Modo de conversão simples ou contínua.
- Interrupção ao término da conversão.
- Eliminador de ruído.

O A/D produz um resultado de 10 bits e o resultado para uma conversão é dado por: $AD = (V_{in} \times 1024) / V_{REF}$, onde V_{in} é a tensão para conversão na entrada no pino e V_{REF} é a tensão de referência selecionada (A_{VCC} , tensão interna referência de 2,56V ou tensão externa de referência).

O valor mínimo representado digitalmente é 0 V (GND) e o valor máximo corresponde à tensão selecionada do pino A_{REF} menos 1 LSB ou a tensão do pino A_{VCC} ou a tensão interna de referência de 2,56V.

Para o CCM, a tensão selecionada como V_{REF} é A_{VCC} (5 volts). Desta forma, o resultado da conversão é $AD = (V_{in} \times 204,6)/V$. O resultado dessa equação é um valor entre 0 e 1023. Como exemplo, uma tensão (V_{in}) de 5V na entrada de uma porta analógica resultará como conversão o valor 1023. Com o objetivo de converter os dados para o valor real entendido pelo usuário, foi desenvolvido um *script* para tratamento dos dados que é explicado com mais detalhes na seção 4.4.3.5.

- USART

A *Universal Synchronous and Asynchronous serial Receiver and Transmitter* (USART) é uma interface de comunicação serial muito flexível que permite a comunicação com outros dispositivos microcontrolados e microprocessados e está programada para modo de transmissão/recepção simultânea (*Full-duplex*⁶).

A USART presente na arquitetura AVR é muito semelhante às encontradas em equipamentos convencionais, como em microcomputadores PCs, por exemplo.

⁵ LSB: Bits menos significativos

⁶ Full-Duplex ou simplesmente duplex, consiste num modo pelo qual os sistemas podem transmitir e receber dados simultaneamente

Capaz de receber e enviar bytes completos, com o stop bit programável, permite a otimização da transmissão serial, o que possibilita taxas de comunicação que vão de 2400 a 115200 bauds⁷ (faixa da norma RS232C). Suas principais características são:

- Operação *Full Duplex* (independentes registradores de recepção e transmissão).
- Operação Síncrona ou Assíncrona.
- Operação Síncrona com clock mestre ou escravo.
- Gerador de taxa de transmissão de alta resolução.
- Suporta frames seriais com 5, 6, 7, 8 ou 9 bits de dados e 1 ou 2 bits de parada.
- Gerador de paridade par ou ímpar e conferência de paridade por hardware.
- Detecção de colisão de dados e erros de frames.
- Filtro para ruído incluindo falso bit de início e filtro digital passa-baixas.
- Três fontes separadas de interrupção (TX completo, RX completo e esvaziamento do Registrador de dados TX).
- Modo de comunicação multi-processado.
- Modo de comunicação assíncrono com velocidade duplicável.

- WATCHDOG TIMER

É um recurso encontrado nos microcontroladores que evita o travamento do programa em aplicações onde este travamento pode ser crítico. O *WatchDog Timer* funciona como um temporizador independente e resseta o microcontrolador se não houver uma instrução WDR (*WatchDog Reset*) antes que o período de *Reset* expire. Oito diferentes períodos do ciclo de clock podem ser selecionados para determinar o período de *Reset*. O *WatchDog Timer* é acionado a partir de um oscilador *on-chip* separado com uma frequência de 1 MHz.

- SPI

A Interface Serial Periférica (SPI – *Serial Peripheral Interface*) permite grande velocidade na transferência síncrona de dados entre o microcontrolador e dispositivos periféricos ou entre outros microcontroladores AVR's. A SPI do ATmega16 inclui as seguintes características:

- Transferência Síncrona de Dados, Full-duplex;
- Operações em modo mestre ou escravo;
- Possibilidade de transmitir primeiro o LSB ou o MSB;

⁷ bauds (unidade de medidas) - número de variações do sinal por segundo
- Frequência máxima de 5 Mbit/s;

- Quatro velocidades de transmissão selecionáveis;
- Flag para a sinalização de fim de transmissão;
- Flag de proteção para a colisão de escrita.;

A SPI também é empregada para programar a memória *flash* e ler ou escrever na EEPROM de dispositivos AVR. A chamada programação ISP (*In-System Programming*) utiliza a interface SPI para permitir a gravação do microcontrolador sem retirá-lo do circuito de trabalho (programação em campo). Isto é muito importante principalmente para aplicação com componentes SMD⁸, que podem sofrer danos ao serem retirados da placa em que são soldados. Os pinos empregados na programação *In-System* do microcontroladores AVR são MOSI, MISO, SCK, RESET, VCC e GND. A descrição funcional desses pinos é apresentada no Anexo 1.

- TEMPORIZADORES/CONTADORES

O ATMEGA16 possui 3 temporizadores/contadores de uso geral com interrupções definidas, sendo dois de 8 bits (*Timer0*) e um de 16 bits (*Timer1*).

O segundo (*Timer 1*) conta com um modo de operação para geração independente de PWM (*Pulse Width Modulation*), que será usado para gerar o sinal utilizado na técnica de carga da bateria do CCM. Também possui outros modos, como o *Capture* (a partir de um sinal externo, pára o funcionamento do timer gerando uma interrupção) e o *Compare* (também a partir de um sinal externo, compara o valor capturado no timer com um, de dois possíveis, pré-determinados, gerando assim a respectiva interrupção).

- SINAL PWM

O microcontrolador Atmega16 pode gerar um sinal PWM, que é baseado no conceito de valor médio de uma forma de onda periódica [17]. Digitalmente, o valor médio de uma forma de onda é controlado pelo tempo em que o sinal fica em nível lógico alto durante um determinado intervalo de tempo. No PWM esse tempo é chamado de ciclo ativo (*Duty Cycle*).

A figura 4.8 apresenta formas de onda PWM em que, a largura do pulso (*Duty Cycle*) é variável de 0 até 100%, com incremento de 25%.

⁸ SMD: *Surface Mount Device* (Dispositivos Montados em Superfície)

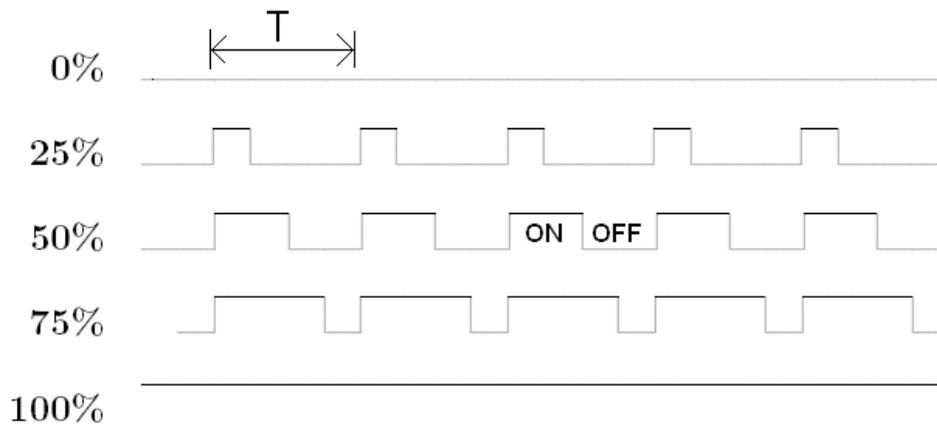


Figura 4.8 – PWM com período T e ciclo ativo de 0%, 25%, 50%, 75% e 100%.

Fonte: *Os Poderosos ucontroladores AVR*

O cálculo do valor médio de um sinal digital é dado por:

$$\text{Valor Médio} = (\text{amplitude máxima} / \text{período}) * \text{Tempo ativo no período}$$

Assim, se o sinal digital tem variação de 0 a 5 V, um ciclo ativo de 50% corresponde a um valor médio de 2,5 V, enquanto um ciclo ativo de 75% corresponderia a 3,75 V. Ao se alterar o ciclo útil do PWM, altera-se o seu valor médio. A Fig. 4.9 ilustra a variação de um PWM digital de 0 a 100% do seu ciclo útil com o passar do tempo.

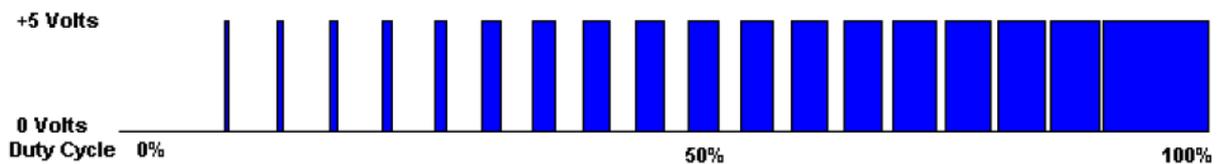


Figura 4.9 – Variação de um PWM em um intervalo de tempo.

Fonte: *Os Poderosos ucontroladores AVR*

A resolução do PWM em um microcontrolador é dada em bits e indica quantos diferentes ciclos ativos podem ser gerados, por exemplo, um PWM de 10 bits pode gerar 1024 larguras diferentes de pulsos.

- TWI (TWO WIRE SERIAL INTERFACE) – I2C

O I2C foi desenvolvido pela Philips e é baseado no protocolo de comunicação entre mestre/escravo empregando duas vias de comunicação [17]. No Atmega foi denominado TWI e algumas de suas características são:

- interface de comunicação simples, mas poderosa e flexível com barramento a dois fios (clock e dado).
- Operação como mestre ou escravo. Tanto Mestre quanto escravo podem transmitir ou receber dados.
- Endereçamento de 7 bits (permitindo até 128 diferentes dispositivos no mesmo barramento).
- Máxima transferência de dados de 400 kHz.

A diferença entre o I2C e o TWI é que o primeiro permite endereçamento também de 10 bits (virtualmente até 1024 diferentes dispositivos no mesmo barramento) enquanto que o segundo só permite o endereçamento de 7 bits. Nos pinos SDA (entrada/saída de dados) e SCL (clock) são necessários resistores de pull-up.

Cada dispositivo ligado ao barramento é reconhecido por um único endereço e pode operar como transmissor ou receptor de dados. A Fig. 4.10 ilustra o barramento I2C com vários dispositivos.

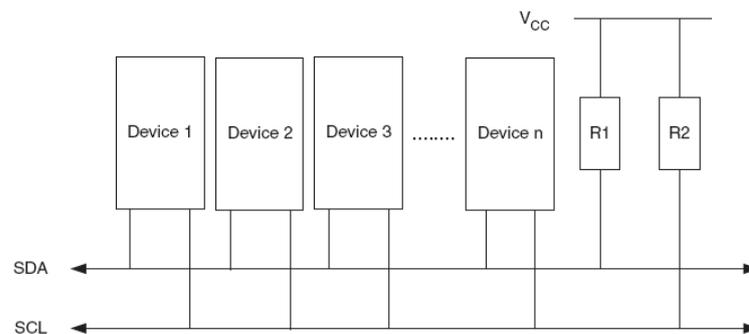


Figura 4.10 – Barramento I2C – TWI.

O dispositivo mestre inicia e termina uma transmissão, gerando o sinal de clock SCL e endereçando o dispositivo escravo.

- MODOS DE SLEEP

Quando se deseja economizar energia das baterias utilizadas na alimentação de microcontrolador, pode ser utilizado o modo *sleep* que habilita o desligamento de módulos não utilizados pelo μ controlador. O Atmega16 possui 6 modos de *sleep* [29]:

- **Idle**: este modo para a CPU, embora permita que outros dispositivos (SRAM, USART, conversores A/D, interface I2C, os Timer/Counter, a porta SPI e sistema de interrupções) continuem funcionando.

- **Power-down:** guarda o conteúdo dos registradores, porém detém o oscilador, desabilitando todas as funções do chip até que se produza uma interrupção externa ou um reset por hardware.
- **Power-save:** neste modo o temporizador assíncrono continua em execução, permitindo ao usuário manter uma base do temporizador enquanto o resto do dispositivo está “dormindo”.
- **ADC Noise Reduction:** para a CPU e todos os módulos de I/O, exceto o Temporizador assíncrono e conversor analógico-digital para minimizar o ruído durante conversões ADC.
- **Standby:** o oscilador de cristal/ressonador continua ligado enquanto o resto do dispositivo está dormindo. Isto permite uma rápida inicialização combinado com baixo consumo de energia.
- **Standby Extended:** o oscilador principal e o temporizador assíncrono continuam ligados, enquanto o resto do dispositivo está dormindo.

• FONTE DE CLOCK

O CCM conta com uma fonte de clock (figura 4.11) para suprir o microcontrolador e também para servir como referência para a geração de taxa de baud rate para a porta de comunicação serial, composta de cristal de 4 MHz e dois capacitores cerâmicos de 22pF. O microcontrolador Atmega16 pode utilizar clock de até 16 Mhz, porém foi usado um cristal de 4 MHz (valor suficiente para operação da porta serial).

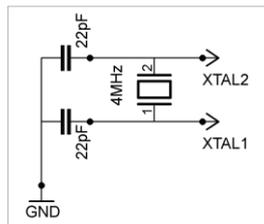


Figura 4.11 – Fonte de Clock

• RESET

O circuito de reset (figura 4.12) reinicializa o microcontrolador quando um nível baixo é detectado no pino 9 (RESET) durante mais de dois ciclos de clock.

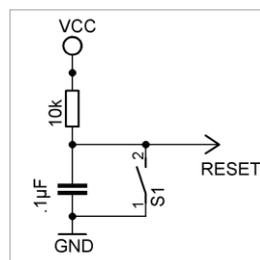


Figura 4.12 – Circuito de RESET

• RELÓGIO I2C

O CCM é programado para ligar/desligar a carga em um determinado horário com o objetivo principal de aumentar a autonomia do sistema de baterias. Para auxiliar nesta funcionalidade, foi instalado o relógio em tempo real DS1307 [30] com barramento I2C. Um cristal de 32.768 KHZ é conectado aos terminais 1 e 2 para geração de clock do relógio e precisão de 1 segundo. Além disso, possui registradores para segundo, minuto, hora, dia da semana, mês e ano com compensação para ano bissexto válida até 2100. Uma entrada para alimentação auxiliar permite que uma bateria do tipo CR2032 seja utilizada para manter o relógio atualizado quando a tensão da alimentação principal (Pino 8) falhar. A figura 4.13 mostra o circuito elétrico do relógio.

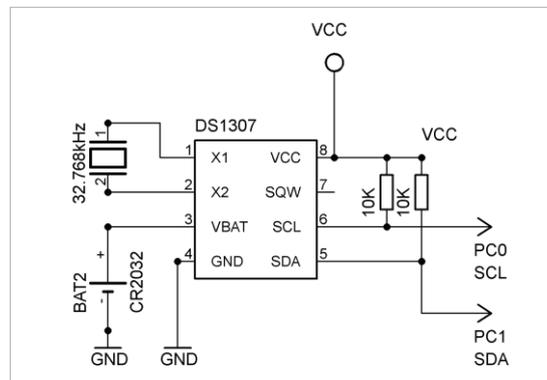


Figura 4.13 – Relógio I2C DS1307

PROGRAMANDO O ATMEGA16

Para programar os microcontroladores AVR (como o Atmega16) é necessário o programa AVR Studio obtido gratuitamente do sítio da ATMEL (www.atmel.com). Para programação em C no AVR Studio, pode-se utilizar o programa WinAVR (<http://winavr.sourceforge.net/>), também gratuito. Para a gravação dos microcontroladores deve-se adquirir hardware específico, para tal, existem esquemas disponíveis na internet [17].

Durante todo este trabalho foi utilizado o kit de desenvolvimento STK-500 da Atmel.

Os microcontroladores AVR são apoiados por um completo conjunto de programas e ferramentas de desenvolvimento de sistemas incluindo: compiladores C, assemblers, simuladores, emuladores de circuitos e kits de avaliação.

4.5.3 BLOCO DE SENSORES

Este bloco é composto pelos sensores cujas saídas são utilizadas pelos CCM nas tomadas de decisão do bloco de controle. Os sensores utilizados são: de tensão do painel solar e da bateria, de corrente do painel solar, de temperatura e de luminosidade.

- **TENSÃO DO PAINEL SOLAR E DA BATERIA**

As portas analógicas do microcontrolador permitem tensões máximas de 5 Volts. Para medir tensões acima deste valor, divisores de tensão com resistores (figura 4.14) foram instalados na entrada das portas analógicas programadas para a medição da tensão do painel solar que chega a 21 V em circuito aberto (V_{oc}) e da tensão da bateria que chega até 14,4 V quando plenamente carregadas.

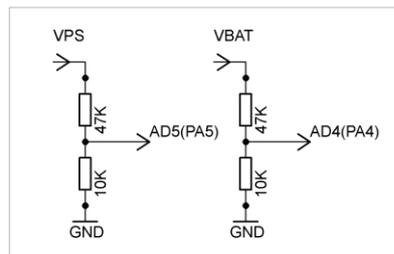


Figura 4.14 – Sensor de tensão do painel solar e da bateria

- **CORRENTE DO PAINEL SOLAR**

Para medir a corrente gerada pelo painel solar foi montado um circuito com o amplificador operacional CA3140 [33] e resistores que definem o ganho do amplificador. Para o resistor *Shunt*, é utilizado um resistor de 0.1 ohms para minimizar a dissipação de potência. Uma função importante do amplificador operacional é isolar o microcontrolador do circuito de medida de corrente. O circuito da figura 4.15 permite a medição de corrente de até 3 A.

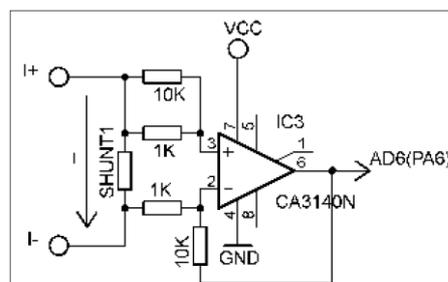


Figura 4.15 – Sensor de corrente

• SENSOR DE TEMPERATURA

A temperatura no interior da caixa hermética é monitorada pelo sensor analógico de temperatura LM35[31] da *National Semiconductor* (www.national.com). Algumas características básicas do sensor da figura 4.16 são:

- Calibrado diretamente em ° Celsius (graus centígrados);
- Faixa de temperatura: de +2 °C a 150°C com alimentação de 4 a 20 volts e de –55°C a 150°C com alimentação simétrica;
- Apresenta uma tensão linear proporcional à temperatura de 10.0 mV para cada °C;.
- Precisão de $\frac{1}{4}$ °C até $\frac{3}{4}$ °C dentro da faixa de temperatura de –55°C a 150°C;

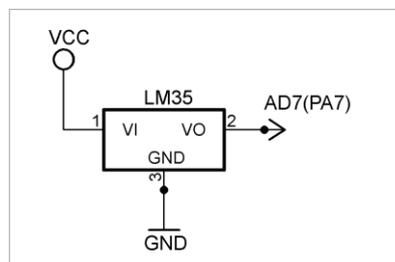


Figura 4.16 – Sensor de temperatura

• SENSOR DE LUMINOSIDADE (LDR)

Com o objetivo de medir a curva de luminosidade no local de instalação do painel solar, foi utilizado um circuito (figura 4.17) com LDR⁹ de 5mm [34]. O LDR (figura 4.17) é um resistor cuja resistência varia em função da luminosidade que incide sobre o mesmo.

O LDR apresenta resistência de 6 KOhm quando exposto a 10 Lux¹⁰ e 20 MegaOhm na ausência de luz (*darkness*).



Figura 4.17 – Sensor de luminosidade LDR

Fez-se necessário o uso de filtro para adequar o ponto de operação do LDR, já que este saturava rapidamente. Isto será abordado melhor na seção de testes do próximo capítulo.

⁹ LDR (Light Dependent Resistor)

¹⁰ Lux: unidade de medida de iluminamento. Corresponde a incidência perpendicular de 1 Lúmen (unidade de medida de fluxo luminoso) em uma superfície de um metro quadrado. Fonte: Wikipédia.

SCRIPT DE TRATAMENTO DOS DADOS

Para se obter os valores reais de tensão e corrente do painel solar, tensão da bateria, temperatura interna à caixa hermética e luminosidade, foi desenvolvido um script denominado trataMedidas.sh (Anexo 4) que trata os dados enviados pelo CCM. Esse *script* utiliza a linguagem de programação para processamento de dados proveniente de texto denominada AWK [26]. Cada tipo de medida efetuada pelos sensores possui um fator de conversão correspondente.

4.5.4 BLOCO DE CHAVEAMENTO

O CCM é composto de um bloco de chaveamento com 3 estágios: um para o chaveamento painel solar x bateria e dois estágios responsáveis por ligar/desligar as cargas primária e secundária.

- **CHAVEAMENTO DAS CARGAS PRIMÁRIA E SECUNDÁRIA**

O CCM possui dois circuitos independentes (figura 4.18) para o acionamento das cargas primária e secundária.

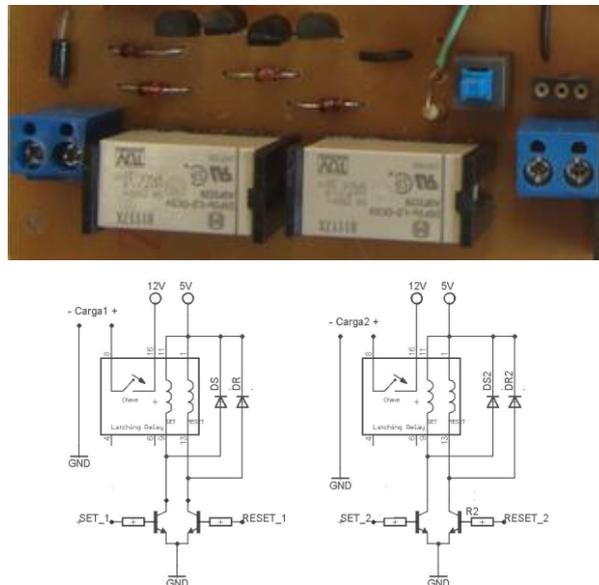


Figura 4.18 – Chaveamento das cargas

O principal elemento deste bloco é o relé mecânico Latching Relay modelo DSP2A-L2-DC5V [25] da Panasonic. Este dispositivo possui duas bobinas, sendo uma para ligar (*set*) e outra para desligar (*reset*) a carga conectada ao mesmo. Diferentemente dos outros relés

circuito formado entre D_3 e C_4 gera a tensão de gate para chavear o Mosfet Q2. Os outros elementos do circuito possuem as seguintes funções: **C2** - Filtro de entrada que suaviza os pulsos de corrente do painel solar; **Q1** - Bloqueia o retorno de corrente da bateria para o painel à noite. Isto poderia ser feito por um diodo de bloqueio, porém devido à queda de tensão produzida pelos diodos, os Mosfets são mais eficientes. Também pode ser utilizado um *Latching Relay* como elemento de bloqueio; e **Indutor L** - suaviza o chaveamento de corrente e junto com C_3 suavizam a tensão de saída.

4.5.5 BLOCO DE COMUNICAÇÃO

Este bloco é formado pela interface USART do microcontrolador Atmega16 e a interface serial do roteador WRT54G da LinkSys. Optou-se por utilizar esta interface devido ao baixo custo da implementação e menor complexidade comparado com outras interfaces como Ethernet ou Wi-fi que deveriam ser adicionadas ao circuito do CCM uma vez que o atmega16 não possui nenhuma destas interfaces.

Além disso, a porta serial consome menos energia, o que é um fator importante quando se utiliza sistemas fotovoltaicos e se deseja economizar energia.

• INTERFACE SERIAL ROTEADOR

A placa-mãe do roteador Wrt54G disponibiliza os sinais referentes à interface serial no conector JP1. Para ter acesso a esses sinais, foi necessário soldar um conector. Na figura 4.20 é possível ver a localização da interface serial e a síntese da conexão dos sinais da USART.

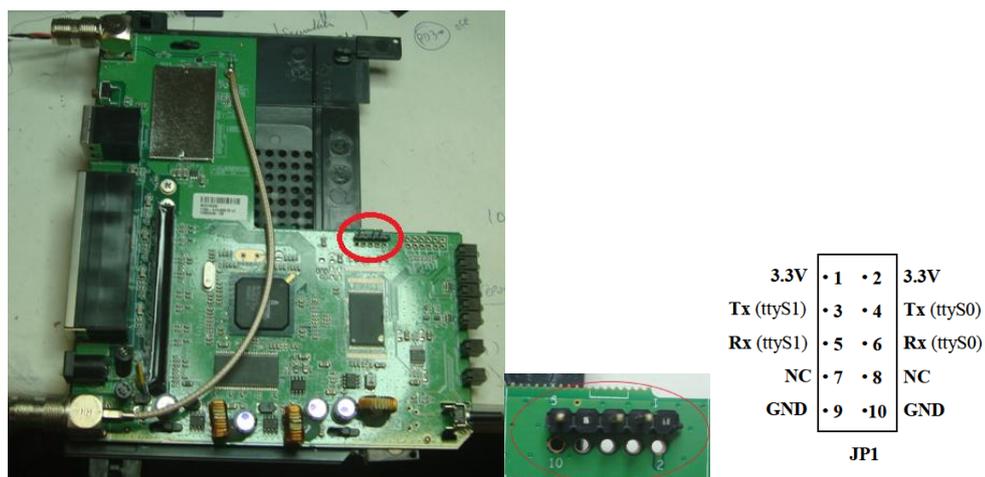


Figura 4.20 – Interface serial roteador LinkSys *Wrt54G*

No detalhe, é possível ver duas interfaces de comunicação serial. A primeira (pinos 3, 5 e 9) é utilizada para a comunicação com o CCM. A segunda interface (pinos 4, 6 e 10) pode ser

utilizada para a visualização do boot do roteador e para acessar o sistema operacional, porém foge do escopo deste trabalho e não será abordada.

Dependendo da versão do sistema operacional OpenWrt, as interfaces podem ser acessadas como `/dev/tts/0`, `/dev/tts/1`, `/dev/ttyS0` ou `/dev/ttyS1`. Na versão do OpenWrt instalada no roteador em testes (Backfire), a interface serial é acessada como `/dev/tts/1`.

- **INTERFACE SERIAL DO MICROCONTROLADOR**

Os pinos 15 e 16 do microcontrolador Atmega16 são utilizados, respectivamente, para RX e TX da interface serial USART. Para permitir acesso, foi montado na placa do protótipo do CCM um conector como pode ser visto na figura 4.21. A figura 4.22 mostra a conexão das interfaces seriais do roteador e do CCM.



Figura 4.21 – Conector da porta serial do CCM

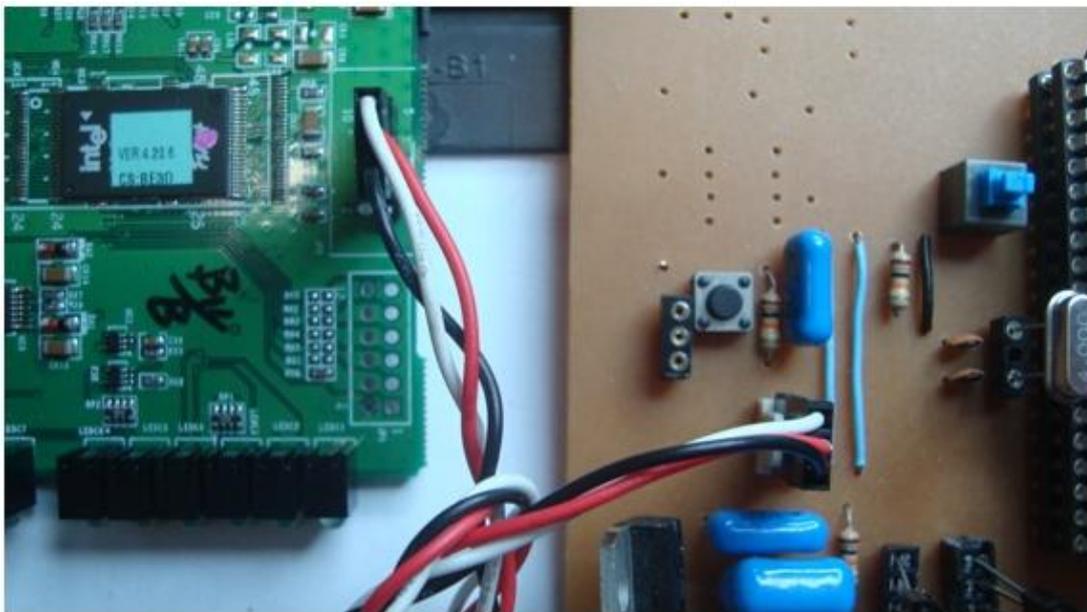


Figura 4.22 – Interconexão serial entre roteador e CCM

Durante o desenvolvimento deste trabalho, também foi testada a interface serial do roteador Bullet2 fabricado pela Ubiquiti. A figura 4.23 apresenta o conector da serial integrada no Bullet2.



Figura 4.23 – Detalhe da interface serial do Bullet2

4.6 - FUNCIONALIDADES DO CCM

A utilização do microcontrolador permitiu a implementação dos seguintes recursos: monitoramento remoto, execução de comandos remotos (controle das cargas primária e secundária e alteração de modos de operação) e *WatchDog Timer*.

4.6.1 SENSORIAMENTO REMOTO

O CCM é programado para enviar dados sensoriais e outras informações para o roteador possibilitando que os parâmetros elétricos do sistema fotovoltaico sejam monitorados em tempo real. Através da execução do comando `cat /dev/tts/1` no roteador é possível verificar os dados enviados pelo controlador pela porta serial e na figura 4.24 são exibidas as entradas de um arquivo de logs criado a partir desses dados.

| Dados do Relógio I2C | ESTÁGIO DE CARGA | V1 | V2 | L _{mppt} | V3 | Vbat | Vps | ips | Temp | Carga2 | WDT | Byte |
|-----------------------------|---------------------------|----|----|-------------------|----|------|-----|-----|------|--------|-----|------|
| SEXTA, 27/04/12 - 11:36:13: | BULK: BAT:CHUMBO MODO:24H | 0 | 0 | 696 | 0 | 422 | 461 | 88 | 43 | OFF | 0 | X |
| SEXTA, 27/04/12 - 11:36:15: | BULK: BAT:CHUMBO MODO:24H | 0 | 0 | 696 | 0 | 422 | 461 | 88 | 43 | OFF | 0 | X |
| SEXTA, 27/04/12 - 11:36:18: | BULK: BAT:CHUMBO MODO:24H | 0 | 0 | 688 | 0 | 420 | 461 | 88 | 43 | OFF | 0 | X |

Figura 4.24 – Leituras dos logs enviados pelo CCM

Pelo que pode ser visto na figura 4.24, os seguintes dados são enviados pelo CCM:

- **Dados do relógio I2C:** dia da semana, data e hora atual do relógio;
- **Estágio de carga:** indica o atual estágio de carga da bateria (*BULK* ou *FLOAT*);
- **Tipo de bateria:** Chumbo-ácido ou LiFePO₄ ;
- **Modo de operação:** o controlador pode está no modo econômico (07-17) ou padrão(24h);
- **Sensores analógicos:** Luminosidade, tensão na bateria, tensão e corrente no painel e temperatura. Devido a resolução de 10 bits das portas conversoras analógico–digital do microcontrolador Atmega16, os valores variam de 0 a 1023.
- **Estado da carga2:** indica se a carga secundária está ligada(ON) ou desligada(OFF);
- **WTD:** informa o número de reinicializações do roteador pela função WTD. Será explicado na seção 4.4.2.4;
- **ByteRec:** indica o byte recebido pelo CCM. Será explicado na seção 4.4.2.4.

No roteador, o script `enviaMedida.sh` (anexo 3) envia os dados recebidos do CCM para um servidor de arquivos. No servidor, o script `trataMedidas.sh` (anexo 4) faz o tratamento dos dados gerando uma nova saída que é entendida pelo usuários. A figura 4.25 ilustra os dados após o tratamento.

| | | | | | |
|-----------------------------|----------------|-------------|------------|-------------|---------------|
| SEXTA, 27/04/12 - 11:36:13: | BULK: | BAT:CHUMBO | MODO:24H | Lum= 3.41 X | Vbat= 12.31 V |
| Vps= 13.34 V | lps= 602.77 mA | Temp= 21 °C | Load2= OFF | WTD= 0 | ByteRec= X |
| SEXTA, 27/04/12 - 11:36:15: | BULK: | BAT:CHUMBO | MODO:24H | Lum= 3.41 X | Vbat= 12.31 V |
| Vps= 13.34 V | lps= 602.77 mA | Temp= 21 °C | Load2= OFF | WTD= 0 | ByteRec= X |
| SEXTA, 27/04/12 - 11:36:18: | BULK: | BAT:CHUMBO | MODO:24H | Lum= 3.37 X | Vbat= 12.26 V |
| Vps= 13.34 V | lps= 602.77 mA | Temp= 21 °C | Load2= OFF | WTD= 0 | ByteRec= X |

Figura 4.25 – Dados após a tratamento

4.6.2 EXECUÇÃO DE COMANDOS REMOTOS

O CCM é programado para executar 4 comandos remotos enviados pelo roteador: mudar o modo de operação do controlador, desligar carga primária por 5 minutos, ligar carga secundária também por 5 minutos e desligar carga secundária imediatamente antes dos 5 minutos.

As operações remotas são executadas a partir de bytes enviados pelo roteador. Para executar um comando remoto, primeiro é preciso se conectar via `ssh`¹¹ [37] ao roteador. Depois envia-se o byte específico através da porta serial utilizando o comando `echo`. Como por exemplo, `echo P > /dev/tts/1`.

A tabela 4.4 ilustra a relação entre os bytes e as operações executadas.

| Byte enviado | Operação executada pelo CCM |
|---------------------|---|
| 1 | Desliga carga primária por 5 minutos |
| 2 | Liga carga secundária por 5 minutos |
| 3 | Desliga carga secundária |
| P | Seleciona Modo de Operação do controlador para Padrão (24h) |
| E | Seleciona Modo de Operação do controlador para Econômico (07-17) |
| O | Byte de sinalização (reseta função WatchDog Timer) |
| X | Não executa nenhum comando |

Tabela 4.4 - Bytes x Operações remotas

Além dos bytes para execução de comandos (“1”, “2”, “3”, “P” e “E”), outros dois bytes são enviados para o CCM:

O – enviado pelo script WDT.sh (Anexo 2). Funciona como um protocolo de *Keep Alive*, em que o roteador envia um byte para o controlador anunciando que está ligado. Será descrito com mais detalhes na seção 4.5.3.

X – byte padrão enviado pelo controlador que está programado para reenviar para o roteador cada byte recebido. Uma vez que os dados são armazenados no arquivo de logs, é possível analisar este arquivo para verificar se há algum problema relacionado com os bytes recebidos.

O microcontrolador pode ser programado para aceitar vários períodos distintos, caso seja necessário.

¹¹ SSH(Security Shell): Protocolo de rede que permite a conexão entre computadores de forma segura, pois criptografa os dados.

- **Comando 1: MODOS DE OPERAÇÃO PROGRAMADOS**

Conforme visto na seção 4.3, o modo de operação pode ser alterado por uma chave. Outra forma de alternar o modo de operação é através de um comando enviado pelo roteador. Isso permite que o usuário alterne entre os modos Padrão (24h) e Econômico (07:00-17:00).

Para alternar para o modo Padrão, é necessário que seja enviado o byte “P” para o controlador. Por outro lado, para alternar para o modo Econômico, envia-se o byte E.

Outros modos de operação diferentes dos citados anteriormente podem ser utilizados em função da necessidade do usuário, bastando para isso reprogramar o microcontrolador.

Como exemplo, os roteadores em modo de operação Econômico poderiam ser ligados numa hora específica da noite para operações de Backup ou atualização. Isso exige o sincronismo entre todos os CCM dos nós *mesh* para que sejam ligados no mesmo período.

- **Comando 2: DESLIGAR CARGA PRIMÁRIA**

Para desligar a carga primária por 5 minutos, basta enviar o byte 1 para o CCM. Após os 5 minutos, a carga primária é religada automaticamente.

- **Comando 3: LIGAR CARGA SECUNDÁRIA**

Por padrão, a carga secundária é mantida desligada. Para ligar a carga secundária por 5 minutos, basta enviar o byte 2 para o CCM. Após os 5 minutos, a carga secundária é desligada automaticamente.

- **Comando 4: DESLIGAR CARGA SECUNDÁRIA**

Se a carga secundária estiver ligada e o CCM receber o byte 3, a mesma é desligada antes de completar os 5 minutos.

4.6.3 - WATCHDOG TIMER / REINICIAR ROTEADOR

Para assegurar o funcionamento do roteador mesmo se um evento inesperado o torna não responsivo (travado), o CCM foi programado para reiniciar o roteador caso não receba uma sinalização do mesmo após um determinado tempo. Esse procedimento se assemelha ao recurso *WatchDog Timer* presente em muitos microcontroladores. A idéia central é obrigar

que o roteador envie mensagens específicas em períodos regulares de tempo para o CCM. Caso a mensagem não chegue, acredita-se que o roteador esteja inoperante e ele será reiniciado. Para que a supervisão da rede *mesh* seja alertada e tome uma ação corretiva, o contador WTD (que indica o número de vezes que o roteador foi reiniciado) é incrementado.

O script *WDT.sh* (Anexo 2), executado permanentemente no roteador, envia o byte “O” para o CCM a cada minuto. O roteador é reiniciado quando o CCM fica por mais de 2 minutos sem receber o byte “O”.

4.6.4 DESLIGAMENTO DOS LEDS INDICADORES DE STATUS

O CCM possui três LEDs: um para indicar que a bateria está conectada ao controlador (Bat_ON), o segundo para indicar que o painel solar está carregando a bateria (PWM_ON) e um terceiro para indicar que a carga primária está desconectada por baixa tensão da bateria (LOAD1_OFF). O CCM geralmente é instalado em locais remotos onde o usuário geralmente não costuma freqüentar e por isso normalmente não precisa monitorar os leds indicadores. Para deixar os leds desligados e só ligá-los quando houver necessidade, uma chave (figura 4.26) foi instalada na placa do CCM.

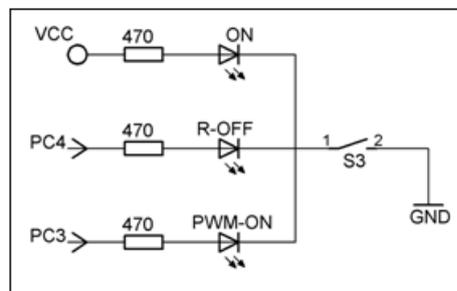


Figura 4.26 - Chave para desligar Leds de sinalização

No caso de manutenção, o operador pode alterar a posição da chave para acionar os leds e assim permitir a leitura do estado do sistema. Dessa forma, os leds podem indicar se a carga está desligada pelo baixo nível de carga na bateria e/ou se o painel solar esta carregando a bateria. A instalação dessa chave diminui o auto-consumo do CCM, pois os leds ficarão desligados praticamente 100% do tempo.

4.7 – FLUXOGRAMA

O programa do CCM foi desenvolvido na linguagem C utilizando o programa AVR STUDIO. O fluxograma da figura 4.27 representa o loop principal do algoritmo de carga. As rotinas ilustradas neste fluxo podem ser encontradas no arquivo charger.c. Todos os arquivos fonte utilizados no programa do CCM são apresentados no Anexo 5.

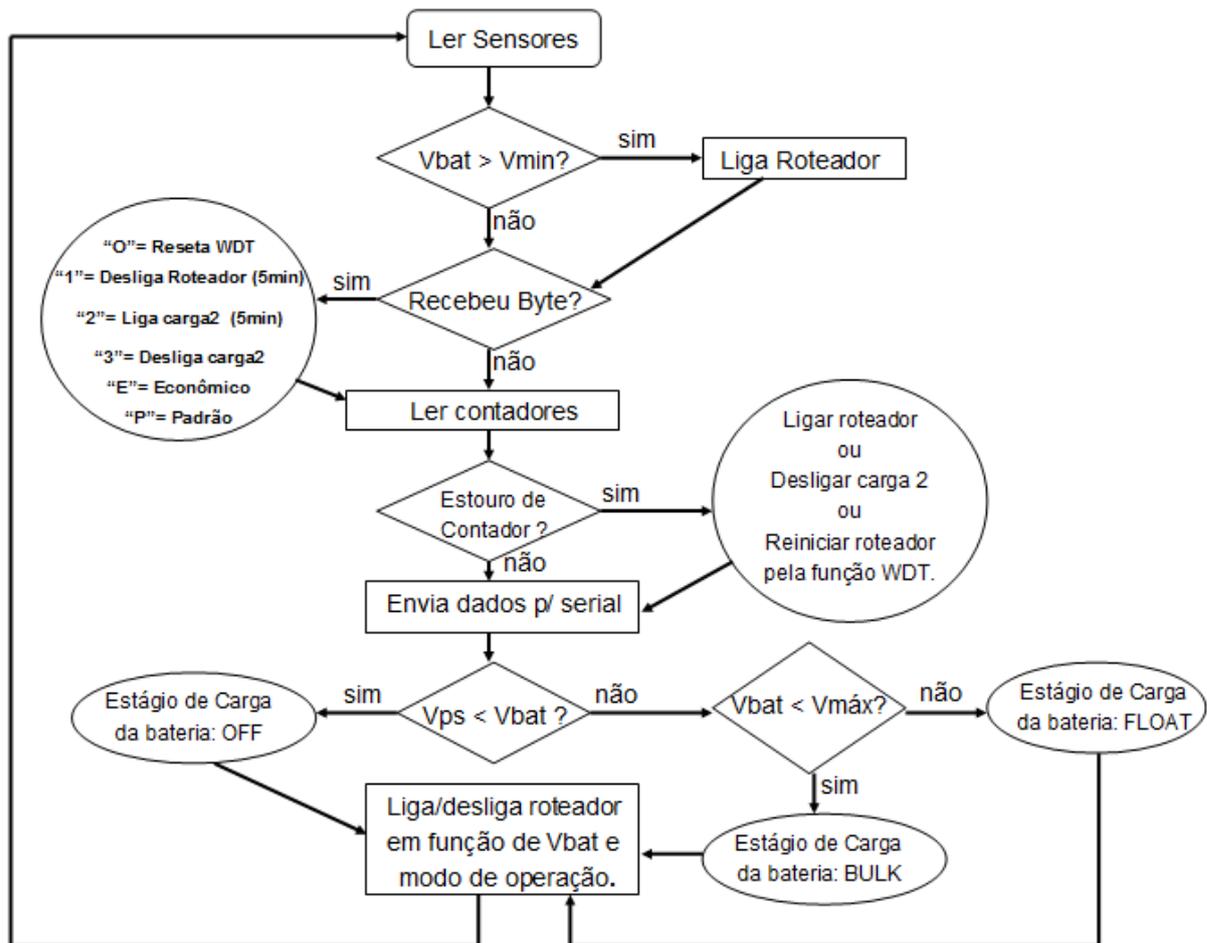


Figura 4.27 - Fluxograma do loop principal do algoritmo de carga

Este capítulo apresentou o protótipo do CCM e seus recursos. No próximo capítulo são descritos todos os testes efetuados e também é apresentada a análise dos resultados.

CAPÍTULO 5 – TESTES E ANÁLISE DOS RESULTADOS

Neste capítulo, é descrita a validação do controlador de carga medindo diferentes aspectos de sua operação como a comunicação, os parâmetros ajustáveis e o funcionamento dos sub-sistemas.

Do aspecto da comunicação são apresentados os testes de sensoriamento remoto, ou seja, se os dados enviados pelo controlador estão sendo recebidos pelo servidor e se é possível atuar sobre o controlador através da execução dos comandos remotos (desligar carga primária, ligar/desligar carga secundária e alterar modos de operação). Dos subsistemas do controlador pode-se destacar: geração do sinal PWM, função de carga/descarga completa de baterias chumbo-ácido e LiFePO₄, função *WatchDog Timer*/reiniciar roteador, sensor de luminosidade e desligamento dos leds indicadores de status. Finalmente são apresentados os testes dos parâmetros ajustáveis para verificar se os pontos de ajustes ($V_{\text{máx}}$, $V_{\text{mín}}$ e V_C) do controlador estão corretos.

5.1 – DESCRIÇÃO DO EXPERIMENTO

Para a realização dos testes, foi montada uma bancada (figura 5.1) no Laboratório de Inclusão Digital (LID) da Universidade Federal Fluminense com os seguintes dispositivos/equipamentos: computador pessoal (PC) com sistema operacional Linux, fonte de tensão/corrente ajustável, conversor serial R232/TTL, protótipo do CCM, baterias de chumbo-ácido e LiFePO₄, roteador modelo WRT54G da LinkSys, osciloscópio e multímetro.



Figura 5.1 - Bancada montada para os testes

A seguir são descritas as principais características de cada elemento utilizado na bancada.

- **Fonte de tensão/corrente regulada** - A fonte de tensão/corrente ajustável utilizada nos testes é a MPL1303/MINIPA.

Fonte: Site Minipa

| | | |
|---|-----------------------|-------------|
|  | Tensão de saída | 0 a 30 V DC |
| | Resolução de Tensão | 20mV |
| | Corrente de saída | 0 a 3 A DC |
| | Resolução de Corrente | 30mA |

Tabela 5.1 - Características básicas da fonte de tensão/corrente ajustável

- **Bateria chumbo-ácido** - A bateria chumbo-ácido selada utilizada é a UPC1270 fabricada pela UNIPOWER.

Fonte: Manual Unipower

| | | |
|---|--------------------------|---------------|
|  | Tensão Nominal | 12 V |
| | Capacidade Nominal (C) | 7 AH |
| | Tensão máxima | 14,4 V |
| | Tensão mínima (de corte) | 10,5 V |
| | Corrente máxima de carga | C/3 (2,3 AH) |
| | | |

Tabela 5.2 - Características básicas da bateria

- **Bateria LiFePO₄** - Para os testes com a bateria LiFePO₄ são utilizadas 4 células modelo 18650 da fabricante a123systems.

Fonte: Manual a123systems / Modelo 18650

| | | | |
|---|--------------------------|----------|--------------------|
|  | | 1 célula | 4 células em série |
| | Tensão Nominal | 3,0 V | 12 V |
| | Capacidade Nominal | 1350 mAh | 1350 mAh |
| | Tensão máxima de carga | 3,6 V | 14,4 V |
| | Corrente máxima de carga | C | C |
| | Tensão mínima | 2,8 V | 11,2 V |

Tabela 5.3 - Características básicas

- **Roteador** - O elemento utilizado como carga nos testes é o roteador WRT54G da Linksys que utiliza como firmware uma personalização do sistema operacional para sistemas embarcados OpenWrt [36]. O OpenWRT é uma distribuição Linux bastante reduzida, desenvolvida especialmente para roteadores sem fio. Esta distribuição, que já foi portada para dezenas de plataformas, conta com uma comunidade bastante ativa de desenvolvedores e usuários. **Fonte: Site Linksys**

| | | |
|---|---------------------------------|--------------------|
|  | Tensão de operação | 12 V |
| | Consumo de corrente(máximo) | 0,32 A a 12 V |
| | Potência de transmissão nominal | 255 mW (31.75 dBm) |
| | Padrão IEEE 802.11 | b/g |
| | Antena Omni direcional | 2 dbi |
| | | |
| | | |

Tabela 5.4 - Características básica do Roteador WRT54G

- **Osciloscópio** - Para a visualização do sinal PWM gerado pelo microcontrolador é utilizado o osciloscópio analógico MO-1225/Minipa. **Fonte: Site MINIPA**

| | | |
|---|--------------------------|---------------------|
|  | Frequência | 20 Mhz |
| | Canais | 2 (duplo traço) |
| | Sensibilidade | 1 mV/ DIV |
| | Varredura | 20ns/DIV ~ 0.2s/DIV |
| | Máxima Tensão de Entrada | 400V (DC+Pico AC). |
| | | |

Tabela 5.5 - Osciloscópio analógico

- **Multímetro** - Para efetuar as medidas de tensão e corrente é utilizado o multímetro digital ET-2053/Minipa. **Fonte: Site MINIPA**

| | | |
|---|-------------------------|---------------------------------------|
|  | Display | LCD 3 ¾ dígitos (4000 Contagens). |
| | Taxa de amostragem | 3 vezes/seg. |
| | Indicação de polaridade | indicação de polaridade negativa "-". |
| | Mudança de faixa | Automática. |
| | Indicação de sobrefaixa | "OL" é mostrado. |
| | | |

Tabela 5.6 - Multímetro Digital

- **Conversor Serial RS232/TTL**

Um computador com sistema operacional Linux é utilizado na realização dos testes iniciais das interfaces seriais. Devido ao fato da interface serial do computador (padrão RS232) possuir níveis de tensão diferentes do CCM e do roteador (padrão TTL¹), um conversor (figura 5.2) de nível serial RS232 p/ TTL foi desenvolvido para converter os níveis das interfaces.

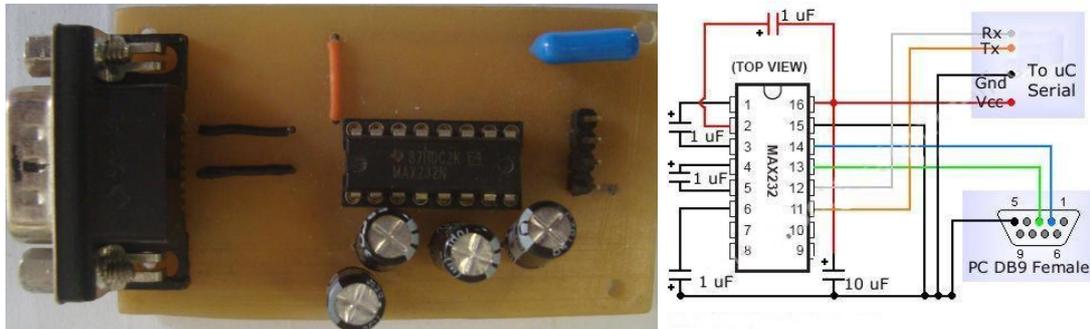


Figura 5.2 - Conversor serial RS232/TTL. Fonte:

O elemento central do conversor é o CI MAX232 da Maxim [24]. Este CI é um *driver/receiver dual* que inclui um gerador de voltagem capacitivo para prover os níveis de sinais compatíveis com o padrão EIA-232.

Cada receptor converte as entradas EIA-232 (+/-12V) para níveis de 5V compatíveis com os níveis TTL/CMOS.

Cada *driver* converte as entradas com níveis TTL/CMOS para níveis simétricos de +/- 12V compatíveis com o padrão EIA-232.

Devido à necessidade de poucos componentes auxiliares (apenas 4 capacitores de 1uF), o CI MAX232 é muito utilizado na conversão de níveis TTL/RS232.

5.2 – SISTEMA DE COLETA E ARMAZENAMENTO DE DADOS

Um computador com sistema operacional Linux é utilizado para armazenar os dados dos sensores enviados pelo controlador durante os testes. A figura 5.3 mostra a interconexão do conversor serial RS232/TTL entre o computador e o CCM.

¹ TTL: família lógica (Transistor-Transistor Logic)

² CMOS: família lógica (Complementary Metal-Oxid Semiconductor)

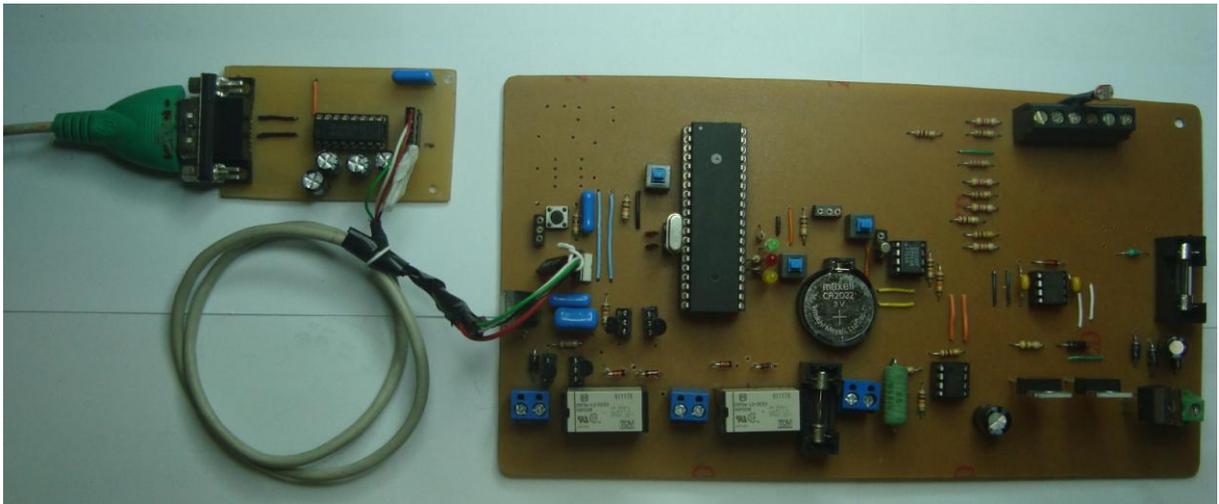


Figura 5.3 – Ligação serial entre computador e CCM

5.3 - TESTES REALIZADOS

Nesta seção são descritos os testes realizados no CCM medindo os seguintes de sua operação: a comunicação, funcionamento dos sub-sistemas e os parâmetros ajustáveis.

5.3.1 – COMUNICAÇÃO

- **SENSORIAMENTO REMOTO**

O circuito da figura 5.4 é montado para a realização dos testes de sensoriameto remoto.

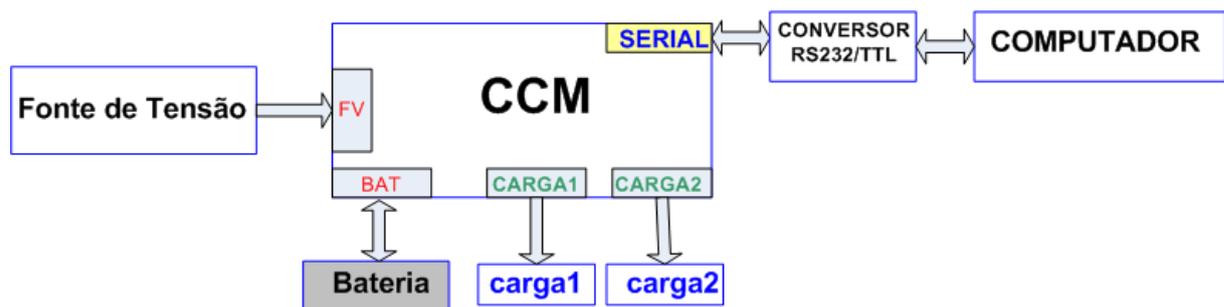


Figura 5.4 – Circuito para monitoramento remoto

O CCM envia os dados sensoriais e outras informações pela interface serial. Para verificar se os dados estavam sendo enviados corretamente pelo controlador, o script “trataMedidas.sh” (anexo 4) trata os dados recebidos. A figura 5.5 mostra os dados após o tratamento realizado pelo script em execução no computador.

| | | | | | |
|-----------------------------|----------------|-------------|-----------|-------------|---------------|
| SEXTA, 27/04/12 - 13:45:32: | BULK: | BAT:CHUMBO | MODO:24H | Lum= 3.41 X | Vbat= 12.64 V |
| Vps= 13.95 V | lps= 786.72 mA | Temp= 22 °C | Load2= ON | WTD= 1 | ByteRec= X |
| SEXTA, 27/04/12 - 13:45:34: | BULK: | BAT:CHUMBO | MODO:24H | Lum= 3.37 X | Vbat= 12.64 V |
| Vps= 13.95 V | lps= 786.72 mA | Temp= 22 °C | Load2= ON | WTD= 1 | ByteRec= X |
| SEXTA, 27/04/12 - 13:45:37: | BULK: | BAT:CHUMBO | MODO:24H | Lum= 3.41 X | Vbat= 12.64 V |
| Vps= 13.95 V | lps= 786.72 mA | Temp= 22 °C | Load2= ON | WTD= 1 | ByteRec= X |

Figura 5.5 – Tela de monitoramento após o tratamento dos dados

Analisando os dados da figura 5.5, é possível verificar que a carga secundária foi ligada no horário indicado (Load2=ON) e a carga primária é reinicializada uma vez (WTD=1).

- EXECUÇÃO DE COMANDOS REMOTOS

Conforme visto na seção 4.5, o CCM é programado para executar 4 comandos remotos apartir de bytes enviados pelo roteador: desligar carga primária por 5 minutos, ligar carga secundária por 5 minutos, desligar carga secundária imediatamente e mudar o modo de operação. O circuito da figura 5.6 foi montado para a realização dos testes apresentados nessa seção.

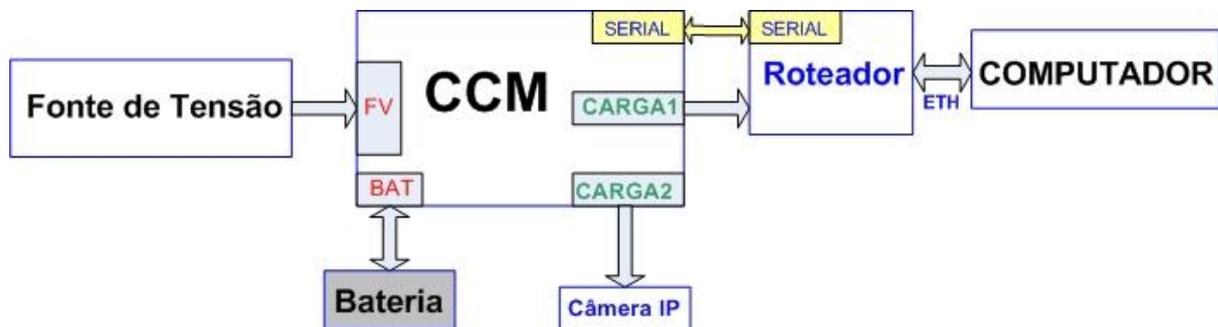


Figura 5.6 – Circuito para os testes de monitoramento remoto

• Teste 1: DESLIGAR CARGA PRIMÀRIA

Para executar este teste, o roteador é acessado via ssh e o byte 1 enviado para o controlador pela interface serial utilizando o comando `echo "1" > /dev/tts/1`. O roteador foi desligado e o led vermelho que indica carga primária desconectada ligou. Após cerca de 5 minutos, o roteador foi ligado e o led vermelho desligou. Como a conexão ssh com o roteador foi perdida quando o mesmo desligou, foi necessário estabelecer uma nova conexão ssh para continuar os testes. Não foi possível registrar o desligamento do roteador pelo arquivo de logs pois é o roteador que salva os dados dos sensores no servidor.

- **Teste 2: LIGAR CARGA SECUNDÁRIA**

A carga secundária é, por padrão, mantida desligada. Para ligá-la, foi enviado o byte 2. observou-se que a câmera ip utilizada como carga secundária ligou e a figura 5.7 mostra o arquivo de logs com o status da carga secundária (Load2 = ON) após o recebimento do byte. Após cerca de 5 minutos a câmera IP foi desligada e o status da carga secundária mudou (Load2 = OFF).

| | | | | | |
|-----------------------------|----------------|-------------|------------|-------------|---------------|
| SEXTA, 27/04/12 - 09:12:33: | BULK: | BAT:CHUMBO | MODO:24H | Lum= 3.37 X | Vbat= 12.64 V |
| Vps= 13.88 V | ips= 960.15 mA | Temp= 21 °C | Load2= OFF | WTD= 0 | ByteRec= X |
| SEXTA, 27/04/12 - 09:12:36: | BULK: | BAT:CHUMBO | MODO:24H | Lum= 3.37 X | Vbat= 12.64 V |
| Vps= 13.88 V | ips= 960.15 mA | Temp= 21 °C | Load2= ON | WTD= 0 | ByteRec= 2 |
| SEXTA, 27/04/12 - 09:12:38: | BULK: | BAT:CHUMBO | MODO:24H | Lum= 3.37 X | Vbat= 12.64 V |
| Vps= 13.88 V | ips= 960.15 mA | Temp= 21 °C | Load2= ON | WTD= 0 | ByteRec= X |
| SEXTA, 27/04/12 - 09:12:41: | BULK: | BAT:CHUMBO | MODO:24H | Lum= 3.37 X | Vbat= 12.64 V |
| Vps= 13.88 V | ips= 960.15 mA | Temp= 21 °C | Load2= ON | WTD= 0 | ByteRec= X |
| ... | | | | | |
| SEXTA, 27/04/12 - 09:17:21: | BULK: | BAT:CHUMBO | MODO:24H | Lum= 3.37 X | Vbat= 12.64 V |
| Vps= 13.88 V | ips= 960.15 mA | Temp= 21 °C | Load2= ON | WTD= 0 | ByteRec= X |
| SEXTA, 27/04/12 - 09:17:23: | BULK: | BAT:CHUMBO | MODO:24H | Lum= 3.37 X | Vbat= 12.64 V |
| Vps= 13.88 V | ips= 960.15 mA | Temp= 21 °C | Load2= OFF | WTD= 0 | ByteRec= X |

Figura 5.7 - Ativação da carga secundária

- **Teste 3: DESLIGAR CARGA SECUNDÁRIA**

Para este teste, a câmera IP foi ligada com o envio do byte 2. Logo em seguida, enviou-se o byte 3 e a câmera IP foi desligada. A figura 5.8 ilustra o arquivo de logs com o registro do desligamento da carga secundária antes dos 5 minutos.

| | | | | | |
|-----------------------------|----------------|-------------|------------|-------------|---------------|
| SEXTA, 27/04/12 - 09:18:07: | BULK: | BAT:CHUMBO | MODO:24H | Lum= 3.37 X | Vbat= 12.72 V |
| Vps= 13.88 V | ips= 960.15 mA | Temp= 21 °C | Load2= OFF | WTD= 0 | ByteRec= X |
| SEXTA, 27/04/12 - 09:18:09: | BULK: | BAT:CHUMBO | MODO:24H | Lum= 3.37 X | Vbat= 12.72 V |
| Vps= 13.88 V | ips= 960.15 mA | Temp= 21 °C | Load2= ON | WTD= 0 | ByteRec= 2 |
| SEXTA, 27/04/12 - 09:18:12: | BULK: | BAT:CHUMBO | MODO:24H | Lum= 3.37 X | Vbat= 12.72 V |
| Vps= 13.88 V | ips= 960.15 mA | Temp= 21 °C | Load2= ON | WTD= 0 | ByteRec= X |
| SEXTA, 27/04/12 - 09:18:14: | BULK: | BAT:CHUMBO | MODO:24H | Lum= 3.37 X | Vbat= 12.72 V |
| Vps= 13.88 V | ips= 960.15 mA | Temp= 21 °C | Load2= OFF | WTD= 0 | ByteRec= 3 |

Figura 5.8 - Desligamento da carga secundária

- **Teste 4: ALTERAÇÃO DO MODO DE OPERAÇÃO**

Como o modo estava em Padrão, o byte “E” foi enviado para o controlador. A figura 5.9 ilustra o arquivo de logs após o envio do byte “E”. Como pode ser visto, após o recebimento do byte, o modo é alterado de Padrão(24h) p/ Econômico(07-17).

| | | | | | |
|-----------------------------|----------------|-------------|------------|-------------|---------------|
| SEXTA, 27/04/12 - 09:25:36: | BULK: | BAT:CHUMBO | MODO:24H | Lum= 3.41 X | Vbat= 13.25 V |
| Vps= 14.35 V | ips= 960.15 mA | Temp= 21 °C | Load2= OFF | WTD= 0 | ByteRec= X |
| SEXTA, 27/04/12 - 09:25:38: | BULK: | BAT:CHUMBO | MODO:07-17 | Lum= 3.41 X | Vbat= 13.25 V |
| Vps= 14.35 V | ips= 960.15 mA | Temp= 21 °C | Load2= OFF | WTD= 0 | ByteRec= E |
| SEXTA, 27/04/12 - 09:25:40: | BULK: | BAT:CHUMBO | MODO:07-17 | Lum= 3.41 X | Vbat= 13.25 V |
| Vps= 14.35 V | ips= 960.15 mA | Temp= 21 °C | Load2= OFF | WTD= 0 | ByteRec= X |

Figura 5.9 - Mudança de modo de operação para Econômico

O retorno ao modo Padrão (24h) foi efetuado com o envio do byte P. A figura 5.10 ilustra o arquivo de logs com a mudança do modo de operação de Econômico para Padrão.

| | | | | | |
|-----------------------------|----------------|-------------|------------|-------------|---------------|
| SEXTA, 27/04/12 - 09:25:44: | BULK: | BAT:CHUMBO | MODO:07-17 | Lum= 3.41 X | Vbat= 13.25 V |
| Vps= 14.35 V | Ips= 960.15 mA | Temp= 21 °C | Load2= OFF | WTD= 0 | ByteRec= X |
| SEXTA, 27/04/12 - 09:25:46: | BULK: | BAT:CHUMBO | MODO:24H | Lum= 3.41 X | Vbat= 13.25 V |
| Vps= 14.35 V | Ips= 960.15 mA | Temp= 21 °C | Load2= OFF | WTD= 0 | ByteRec= P ← |
| SEXTA, 27/04/12 - 09:25:48: | BULK: | BAT:CHUMBO | MODO:24H | Lum= 3.41 X | Vbat= 13.25 V |
| Vps= 14.35 V | Ips= 960.15 mA | Temp= 21 °C | Load2= OFF | WTD= 0 | ByteRec= X |

Figura 5.10 - Mudança de modo de operação para Padrão

O modo de operação econômico pode ser observado pois o roteador é desligado após as 17:00 hs e religado após as 07:00 hs. Além disso, o led vermelho fica ligado no horário entre 17:00 e 07:00. Nesse mesmo período, o servidor deixa de receber dados enviados pelo roteador e o arquivo de logs deixa de ser atualizado, ou seja, não apresenta nenhum registro no horário entre 17:00 e 07:00.

5.3.2 – PARÂMETROS AJUSTÁVEIS

Os parâmetros ajustáveis são essenciais para o bom desempenho da bateria, evitando sobrecargas ou descargas profundas. Dessa forma, um dos testes mais importantes é verificar o funcionamento do CCM em relação aos pontos de ajuste definidos no firmware gravado no microcontrolador.

Nesta seção, são descritos os testes realizados para os dois tipos de baterias utilizadas: chumbo-ácido e LiFePO₄.

- **BATERIA 1: CHUMBO-ÁCIDO**

Com a chave de tipo de bateria posicionada na posição referente a chumbo-ácido, uma bateria selada de chumbo-ácido de 12V/7AH é conectada ao controlador. Para este tipo de bateria, espera-se que os pontos de corte sejam $V_{Mín} = 11,5V$, $V_c = 12,5V$ e $V_{Máx} = 14,4V$.

DESCRIÇÃO DOS TESTES

Estes testes são realizados para adequar o ponto de operação ao circuito real construído, uma vez que no projeto são esperadas determinadas características que ao serem implementadas com componentes reais apresentam desvios.

A figura 5.11 apresenta um diagrama esquemático do circuito de teste montado para a realização dos testes dos pontos de ajuste ($V_{Mín}$, V_c e $V_{Máx}$). Como pode ser visto, a fonte de tensão é conectada nos conectores referentes à bateria. Dessa forma, a variação da tensão da fonte representa a variação da tensão da bateria permitindo assim que os pontos de ajustes possam ser testados.

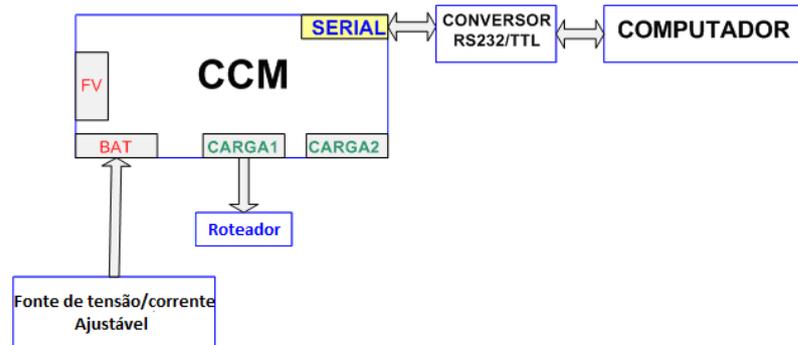


Figura 5.11 – Diagrama esquemático do circuito de teste

- **1º TESTE: TENSÃO DE DESCONEXÃO DA CARGA PRIMÁRIA ($V_{MÍN}$)**

Inicialmente, a fonte de tensão é ajustada para 12V e 1A. Em seguida, a tensão é reduzida gradativamente numa razão de 0,1 V em cada passo. A figura 5.12 mostra algumas entradas do arquivo de log em que podem ser visualizados alguns valores da tensão da bateria até o valor $V_{MÍN}$ ser alcançado.

| | | | | | |
|-----------------------------|-----------|-------------|------------|-------------|---------------|
| SEXTA, 27/04/12 - 14:02:01: | OFF: | BAT:CHUMBO | MODO:24H | Lum= 3.41 X | Vbat= 11.73 V |
| Vps= 0 V | lps= 0 mA | Temp= 21 °C | Load2= OFF | WTD= 2 | ByteRec= X |
| SEXTA, 27/04/12 - 14:02:03: | OFF: | BAT:CHUMBO | MODO:24H | Lum= 3.41 X | Vbat= 11.73 V |
| Vps= 0 V | lps= 0 mA | Temp= 21 °C | Load2= OFF | WTD= 2 | ByteRec= X |
| SEXTA, 27/04/12 - 14:02:06: | OFF: | BAT:CHUMBO | MODO:24H | Lum= 3.41 X | Vbat= 11.62 V |
| Vps= 0 V | lps= 0 mA | Temp= 21 °C | Load2= OFF | WTD= 2 | ByteRec= X |

Figura 5.12 – Detalhe da tensão $V_{MÍN}$

Nota-se que quando a tensão na fonte atinge o limiar de 11,62 V, a carga primária é desconectada e o roteador desliga. Além disso, como o CCM é programado para ligar o led vermelho (sinalização de status) quando a carga primária é desconectada foi possível observar o acionamento deste led.

Assim, para que o sistema funcionasse de acordo com as especificações, o controlador foi reprogramado para adequá-lo ao circuito construído de forma que o valor de corte seja 11.5V. Este ajuste tem que ser feito para cada unidade construída.

- **2º TESTE: TENSÃO DE CONEXÃO DA CARGA PRIMÁRIA (V_C)**

Usando o mesmo circuito da figura 5.13, a tensão da fonte é aumentada gradativamente numa razão de 0,1 V em cada passo. Nota-se que quando a tensão na fonte atinge o limiar de 12,67 V, a carga primária é conectada e o roteador liga. Além disso, o led vermelho é desligado. A figura 5.13 mostra algumas entradas do arquivo de log em que podem ser visualizados alguns valores da tensão da bateria até o valor V_C ser alcançado e o roteador ser ligado.

| | | | | | |
|-----------------------------|-----------|-------------|------------|-------------|---------------|
| SEXTA, 27/04/12 - 14:25:17: | OFF: | BAT:CHUMBO | MODO:24H | Lum= 3.41 X | Vbat= 12.41 V |
| Vps= 0 V | lps= 0 mA | Temp= 21 °C | Load2= OFF | WTD= 2 | ByteRec= X |
| SEXTA, 27/04/12 - 14:25:10: | OFF: | BAT:CHUMBO | MODO:24H | Lum= 3.41 X | Vbat= 12.53 V |
| Vps= 0 V | lps= 0 mA | Temp= 21 °C | Load2= OFF | WTD= 2 | ByteRec= X |
| SEXTA, 27/04/12 - 14:25:22: | OFF: | BAT:CHUMBO | MODO:24H | Lum= 3.41 X | Vbat= 12.67 V |
| Vps= 0 V | lps= 0 mA | Temp= 21 °C | Load2= OFF | WTD= 2 | ByteRec= X |

Figura 5.13 – Detalhe da tensão de conexão da carga primária (V_C)

Da mesma forma que no teste anterior, o controlador foi reprogramado para adequá-lo ao circuito construído de forma que o ponto de operação fosse ajustado e a tensão de conexão da carga ajustada para 12,5 V.

- **3º TESTE: TENSÃO MÁXIMA DA BATERIA ($V_{m\acute{a}x}$)**

Para evitar que a bateria sofra uma sobrecarga no processo de carregamento, o controlador deve interromper o fluxo de corrente do painel quando a tensão na bateria atingir o valor máximo ($V_{m\acute{a}x}$) para o qual ela foi projetada. Segundo o fabricante da bateria chumbo-ácido utilizada nos testes, a tensão máxima recomendada é 14.4 V.

Com o objetivo de verificar se a bateria seria carregada até o valor $V_{m\acute{a}x}$, o controlador foi reprogramado para desconsiderar a tensão no painel solar, uma vez que o processo de carga só deve ocorrer quando a tensão no painel for maior que a tensão na bateria.

Usando o mesmo circuito da figura 5.11, a tensão da fonte é aumentada gradativamente numa razão de 0,1 V em cada passo até o estado de carga mudar de BULK para FLOAT. Notamos pela figura 5.14 que a mudança de estado ocorreu quando a tensão na bateria atingiu o valor de 14,21 V.

| | | | | | |
|----------------------------|-----------|-------------|------------|-------------|---------------|
| SEXTA, 27/04/12- 14:45:03: | BULK: | BAT:CHUMBO | MODO:24H | Lum= 3.41 X | Vbat= 14.21 V |
| Vps= 0 V | lps= 0 mA | Temp= 22 °C | Load2= OFF | WTD= 3 | ByteRec= X |
| SEXTA, 27/04/12- 14:45:05: | FLOAT: | BAT:CHUMBO | MODO:24H | Lum= 3.41 X | Vbat= 13.67 V |
| Vps= 0 V | lps= 0 mA | Temp= 22 °C | Load2= OFF | WTD= 3 | ByteRec= X |

Figura 5.14 – Detalhe da tensão ($V_{m\acute{a}x}$)

Da mesma forma que os 2 testes anteriores, o controlador foi reprogramado para corrigir o ponto de operação e a tensão máxima de carga ajustada para 14,4 V.

- **BATERIA 2: LiFePO₄**

Com o objetivo de adequar o ponto de operação ao circuito real construído em função dos desvios apresentados por componentes reais, são efetuados testes semelhantes aos das baterias chumbo-ácido com o mesmo circuito da figura 5.11.

Segundo o fabricante da bateria LiFePO₄ utilizada nos testes, as tensões mínima e máxima recomendadas para cada célula são, respectivamente, 2.8V e 3.6V. Como são colocadas em série, as tensões mínima e máxima do pack com 4 células são, respectivamente, 11.2V e 14.4 V. Para evitar descarga profunda e sobrecarga durante os testes, as tensões mínima e máxima programadas no microcontrolador são, respectivamente, 11.4V e 14.2V.

DESCRIÇÃO DOS TESTES

Inicialmente a chave de seleção do tipo de bateria é colocada na posição para LiFePO₄. Para verificar os pontos de ajustes da bateria LiFePO₄ ($V_{\text{mín}}=11,4\text{V}$, $V_C=12,5\text{V}$ e $V_{\text{máx}}=14,2\text{V}$), são realizados testes semelhantes aos da bateria chumbo-ácido, ou seja, a fonte é interconectada no CCM no lugar da bateria LiFePO₄.

- **1º TESTE: TENSÃO DE DESCONEXÃO DA CARGA ($V_{\text{MÍN}}$)**

Após iniciar os testes, é possível notar que o desligamento da carga primária ocorreu quando a tensão atingiu o valor de 11.55 V. Assim, para que o sistema funcionasse de acordo com as especificações das baterias LiFePO₄, o controlador foi reprogramado para adequá-lo ao circuito construído de forma que o valor de corte seja 11.4 V. Também é possível verificar o desligamento da carga através da ativação do led de sinalização.

- **2º TESTE: TENSÃO DE CONEXÃO DA CARGA (V_C)**

Note que a tensão de reconexão da carga primária ocorreu quando a tensão atingiu o valor de 12.5 V. Isto era esperado porque o ajuste já feito no teste anterior para as baterias chumbo-ácido garantiu que a tensão de conexão estivesse correta. Isso se deve ao fato de tanto as baterias-chumbo ácido como as LiFePO₄ estarem programadas com V_C de 12.5V. Também é possível verificar a conexão da carga primária através do desligamento do led de sinalização.

- **3º TESTE: TENSÃO MÁXIMA DA BATERIA ($V_{Máx}$)**

Constatou-se que a mudança de estado de carga de BULK para FLOAT ocorreu quando a tensão atingiu o valor de 14.1V. Assim, para que o sistema funcionasse de acordo com as especificações das baterias LiFePO₄, o controlador foi reprogramado para adequá-lo ao circuito construído de forma que o valor de tensão máxima fosse ajustado para 14.2 V.

5.3.3 – SUBSISTEMAS DO CONTROLADOR

- **GERAÇÃO DO SINAL PWM**

Para visualizar o sinal PWM gerado pelo microcontrolador, montou-se o circuito da figura 5.15. O microcontrolador é programado para gerar um sinal PWM numa frequência de 330 Hz (período de 3 milissegundos) com as larguras de pulso (*Duty Cycle*) em 10%, 50% e 99%. Após ajustar o osciloscópio nas escalas apropriadas e ligar o CCM, as formas de onda correspondentes aos *Duty-Cycle* programados são visualizadas na tela do osciloscópio conforme pode ser visto nas figuras de 5.16 a 5.18.

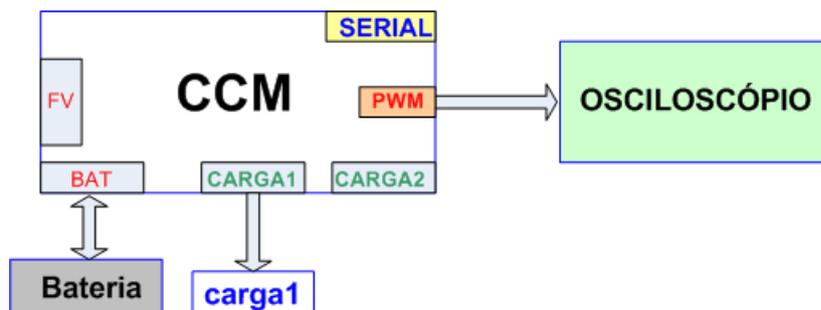


Figura 5.15 – Diagrama do circuito montado para testes do sinal PWM

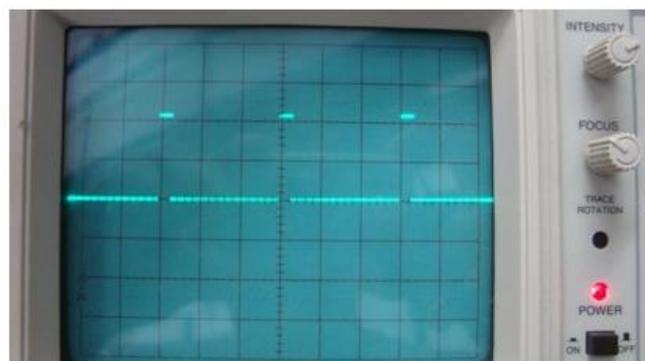


Figura 5.16 – PWM com 10% de *Duty-Cycle*

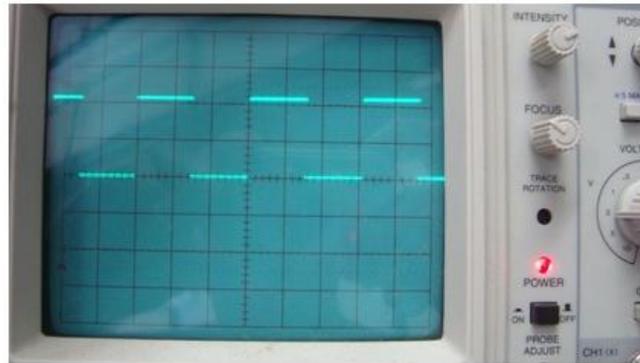


Figura 5.17 – PWM com 50% de *Duty-Cycle*

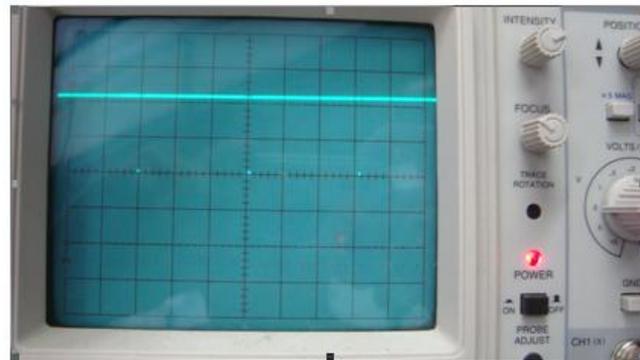


Figura 5.18 – PWM com 99% de *Duty-Cycle*

- **CARGA/DESCARGA COMPLETA**

Nesta seção são descritos os testes feitos com o objetivo de verificar o tempo de carga e descarga das baterias chumbo-ácido e LiFePO_4 .

- **BATERIA1: CHUMBO-ÁCIDO**

- **Teste1: CARGA COMPLETA**

A figura 5.19 apresenta o diagrama do circuito montado para a realização dos testes de carga completa. Com a fonte desligada, uma bateria de 12V/7Ah é conectada ao controlador e descarregada até a tensão em seus terminais alcançar o valor $V_{mín}$ 11,5 V.

Um multímetro é conectado aos terminais da bateria para se ter uma segunda maneira de mensurar a tensão na bateria durante todo o carregamento da mesma. Após ajustar a fonte 16V/1A, a mesma é ligada e inicia-se o processo de carregamento da bateria. É possível verificar a corrente da fonte (pelo log I_{PS} da tela de monitoramento) e o aumento gradativo da tensão da bateria. Ao analisar o arquivo de log armazenado no computador, a bateria alcançou

a carga completa (14.4 V) em torno de 6 horas. O figura 5.20 mostra a curva de carga da bateria.

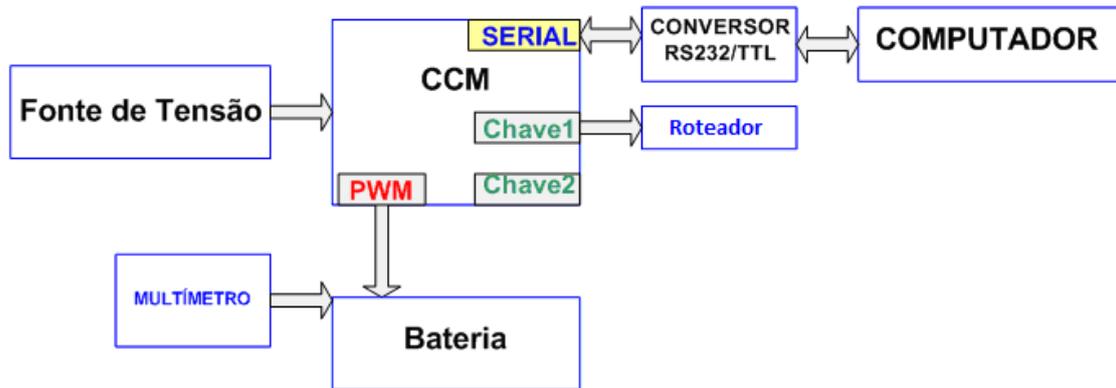


Figura 5.19 – Diagrama esquemático do circuito

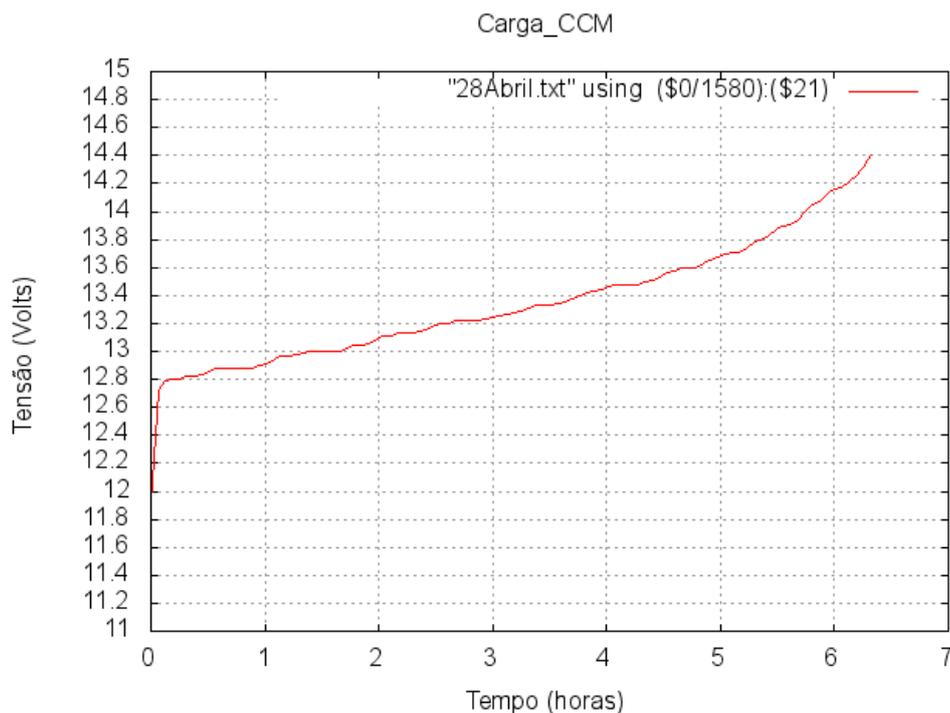


Figura 5.20 – Curva de carga da bateria chumbo-ácido

- **Teste2: DESCARGA COMPLETA**

Após a bateria ser carregada (conforme descrito na seção anterior), o processo de descarga é iniciado com o desligamento da fonte de tensão/corrente. A figura 5.21 mostra a curva de descarga plotada a partir do arquivo de logs armazenado no computador. Como pode ser visto, o tempo de descarga é de aproximadamente 16 horas.

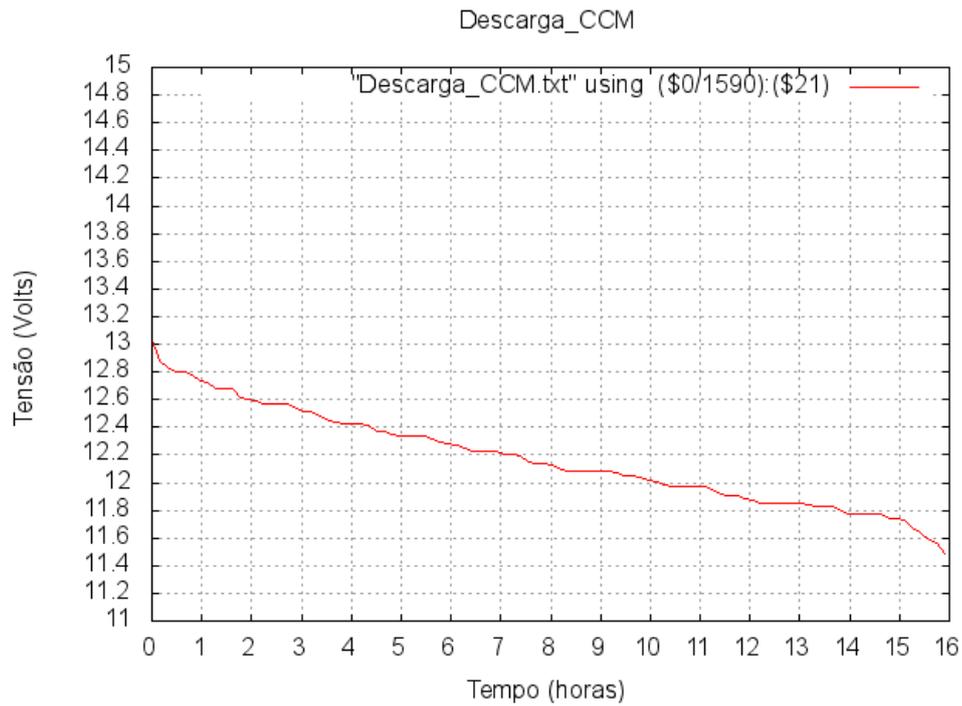


Figura 5.21 – Curva de descarga da bateria chumbo-ácido

- BATERIA 2: LiFePO_4

• Teste1: CARGA COMPLETA

Para que obtivessem o mesmo potencial após o carregamento, cada uma das células de LiFePO_4 é carregada individualmente até o valor máximo recomendado pelo fabricante. Dessa forma, o microcontrolador precisou ser reprogramado com a tensão máxima de carga de 3,6 V e a fonte de tensão é ajustada para 4,4V e 1A. Depois de carregadas, as células são colocadas em série e montadas num PACK, conforme pode ser visto na figura 5.22. A associação série das 4 células LiFePO_4 resulta numa tensão nominal de 12 V e capacidade nominal de 1350 mAh.



Figura 5.22 – Células LiFePO_4 em série

- **Teste2: DESCARGA COMPLETA**

Após reprogramar o microcontrolador com os pontos de ajustes ($V_{\text{mín}} = 11,4\text{V}$, $V_c = 12,5\text{ V}$ e $V_{\text{máx}} = 14,4\text{ V}$) e conectar o *Pack* das células no controlador, iniciou-se o processo de descarga. Pode-se observar que a carga primária é desligada cerca de 1 hora após o início da descarga.

Ao medir a tensão total do conjunto de células e de cada célula individualmente, os seguintes valores foram encontrados: $V_{c_1} = 3,09\text{V}$, $V_{c_2} = 2,60\text{V}$, $V_{c_3} = 2,93\text{V}$, $V_{c_4} = 2,66\text{V}$ e $V_{\text{total}} = 11,28\text{V}$. Este resultado mostrou que apesar da tensão total ($11,28\text{V}$) ser menor que o $V_{\text{mín}}$ programado ($11,4\text{ V}$), o que determinou o desligamento da carga primária, as células apresentaram níveis bens distintos de descarga. As células (C_1 e C_3) não estavam com o valor mínimo de tensão ($2,8\text{ V}$) recomendado pelo fabricante e ainda poderiam fornecer energia à carga sem sofrerem danos. Por outro lado, as tensões das células C_2 e C_4 atingiram valores inferiores a $2,8\text{ V}$, que podem trazer danos às células.

Para evitar que as células de LiFePO_4 sejam descarregadas ou carregadas de forma não homogênea e sofram danos irreversíveis, podem ser utilizados 3 procedimentos: equalização das células, monitoramento da pior célula e aquisição de células com mesmas características.

No processo de Equalização das células, a tensão em cada célula é monitorada constantemente. Quando apresentam desvios de tensão considerados significativos, as células são colocadas em paralelo durante alguns instantes para que obtenham o mesmo nível de carga e são recolocadas em série logo em seguida. Este procedimento exige um hardware adicional e deve ser levado em consideração.

No monitoramento da pior célula também é necessário que a tensão em cada célula seja monitorada constantemente para não ficar abaixo da tensão mínima ou acima da tensão máxima recomendada pelo fabricante. Quando a tensão da pior célula se aproxima da tensão mínima, a carga primária é desconectada para evitar que a célula seja danificada. Por outro lado, quando a tensão da pior célula se aproxima da tensão máxima, o estágio de carga deve ser alterado para evitar a sobrecarga da célula. Este procedimento também exige um hardware adicional, apesar de menor complexidade que o processo de Equalização das células citado anteriormente.

Outra solução é adquirir células com as mesmas características para que sejam descarregadas/carregadas de forma homogênea. As células utilizadas nestes testes foi adquirida de forma avulsa e isso não garante que tenham as mesmas características, apesar de terem a mesma especificação nominal.

Na patente “*Method and System For Charge Equalization of Lithium-Ion Batteries*” [40] é apresentado um sistema de carga e equalização das células de uma bateria de íons de lítio. Cada célula contém um controlador de carga individual e o conjunto de células (em série) possui além de um controlador de carga central, um controlador de descarga da bateria. Cada controlador tem um circuito sensor de tensão. Quando a tensão máxima de carga é atingida em cada célula, o carregamento termina ou pode ser reduzido para manter a tensão de carga constante. Cada controlador de carga da célula pode terminar independente do outro controlador de carga para evitar sobrecarga de uma das células que pode carregar mais rápido que outras. Por outro lado, a descarga é iniciada após se detectar o limiar máximo de carga predeterminado em cada célula. Um dos objetivos da invenção é possibilitar um controle mais eficiente da equalização das cargas nas células, eliminando a necessidade de separar o circuito de equalização de cada célula.

- **FUNÇÃO WATCHDOG TIMER / REINICIAR ROTEADOR**

Como visto na seção 4.5, o CCM é programado para reiniciar o roteador através da função *WatchDog Timer* caso algum problema inesperado trave o roteador. Podem ser realizados dois testes para verificar o funcionamento dessa função.

O primeiro teste consiste em encerrar o script *WDT.sh*, parando com isso o envio do byte “O” para o CCM. Depois de cerca de 2 minutos, o roteador foi desligado e logo em seguida religado. Também foi observado que o led vermelho ligou indicando que a carga primária foi desconectada e logo em seguida foi desligado, indicando que a carga primária foi ligada.

A figura 5.23 ilustra o arquivo de logs armazenados no computador onde é possível verificar a alteração do contador WTD indicando o número de vezes que o roteador foi reiniciado pela função *WatchDog Timer*.

| | | | | | |
|-----------------------------|----------------|-------------|------------|-------------|---------------|
| SEXTA, 27/04/12 - 10:22:52: | BULK: | BAT:CHUMBO | MODO:24H | Lum= 3.41 X | Vbat= 13.32 V |
| Vps= 14.35 V | Ips= 960.15 mA | Temp= 21 °C | Load2= OFF | WTD= 0 | ByteRec= X |
| SEXTA, 27/04/12 - 10:22:54: | BULK: | BAT:CHUMBO | MODO:24H | Lum= 3.41 X | Vbat= 13.32 V |
| Vps= 14.35 V | Ips= 960.15 mA | Temp= 21 °C | Load2= OFF | WTD= 1 | ByteRec= P |
| SEXTA, 27/04/12 - 10:22:56: | BULK: | BAT:CHUMBO | MODO:24H | Lum= 3.41 X | Vbat= 13.32 V |
| Vps= 14.35 V | Ips= 960.15 mA | Temp= 21 °C | Load2= OFF | WTD= 1 | ByteRec= X |

Figura 5.23 – Detalhe do contador WDT

O contador **WDT** pode ser utilizado, por exemplo, para que o administrador da rede *mesh* monitore o número de vezes que o roteador é reiniciado e busque uma solução corretiva para o problema.

O segundo teste consiste em desconectar o cabo serial conectado entre CCM e roteador. Isso fez com que o controlador deixasse de receber o byte de sinalização e também resultou na reiniciação do roteador.

5.3.4 – DESATIVAÇÃO DOS LEDS INDICADORES

Para testar o desligamento dos 3 leds indicadores de status, a chave correspondente a essa função foi utilizada e a figura 5.24 mostra os leds de sinalização antes e depois da utilização da chave.



Figura 5.24 – Detalhe dos leds antes e depois da utilização da chave

5.3.5 – AUTO-CONSUMO DO CCM

Pelos testes efetuados o CCM consome cerca de 40 mA e os leds consomem juntos cerca de 12 mA. Dessa forma, a utilização da chave para desativar os leds reduz em até 30% o auto-consumo do controlador. Além disso, o uso dos Latching Relay no circuito de chaveamento das cargas primárias e secundárias contribui na redução do consumo do controlador.

5.4 – SATURAÇÃO DO SENSOR DE LUMINOSIDADE LDR

Durante os testes com o LDR, é constatada uma saturação das medidas efetuadas no LDR durante 9 e 15 horas de um dia ensolarado com céu claro e sem nuvens, conforme pode ser observado pela figura 5.25.

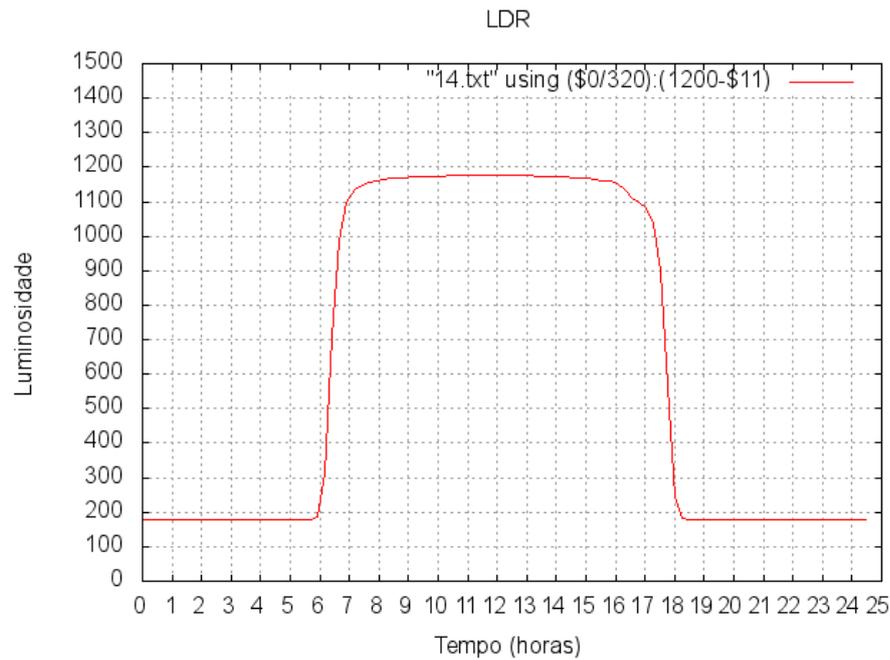


Figura 5.25 - Curva de luminosidade LDR

Para minimizar a saturação do LDR, filme fotográfico de 35 mm (em vários níveis de transparência) é utilizado como filtro. A figura 5.26 ilustra o circuito montado com 2 LDR's com e sem filtro.



Figura 5.26 - LDR's sem filtro e com filtro

Este circuito é instalado junto ao painel solar no topo do prédio de Engenharia da Universidade Federal Fluminense. Após coletar os dados do LDR com filtro durante 24 horas de um dia ensolarado de céu claro e sem nuvens, é traçada a curva de luminosidade apresentada na figura 5.27.

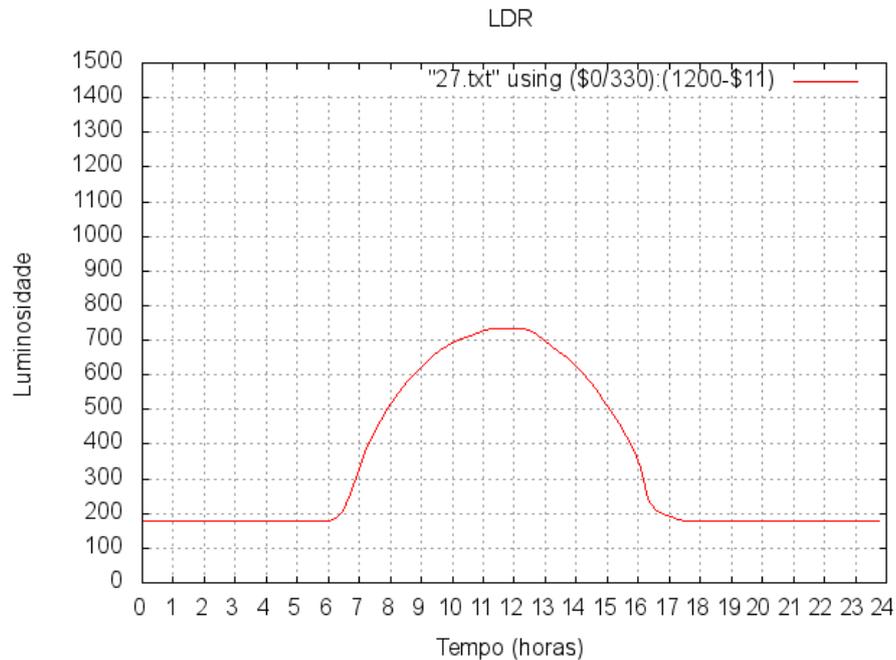


Figura 5.27 - Curva luminosidade de um LDR com filtro

Como pode ser observado, o uso do filtro reduziu bastante a saturação sofrida pelo LDR (comparada com a curva da figura 5.25). A luminosidade máxima medida pelo sensor foi obtida por volta das 12:00 (meio-dia), como era de se esperar.

5.5 - INTERFACE WEB PARA MONITORAMENTO

O projeto ReMesh [02] também utiliza energia solar para alimentar os nós da rede mesh instalada no campus da Praia Vermelha da UFF. Com o objetivo de analisar de forma mais precisa os parâmetros elétricos coletados pelo Kit de Sensoriamento instalado em um dos nós da rede, é utilizada a interface WEB [38] (figura 5.28) desenvolvida inicialmente para o Projeto ReMoTe.

Os dados coletados são armazenados num servidor e a interface WEB desenvolvida permite uma melhor visualização desses dados.

Como pode ser visto na figura 5.28, os dados dos sensores são exibidos em tempo real no lado esquerdo da figura. Para uma melhor análise dos dados, também são exibidos gráficos de tensão da bateria, luminosidade, tensão e corrente do painel solar e temperatura de um determinado dia, semana ou mês.

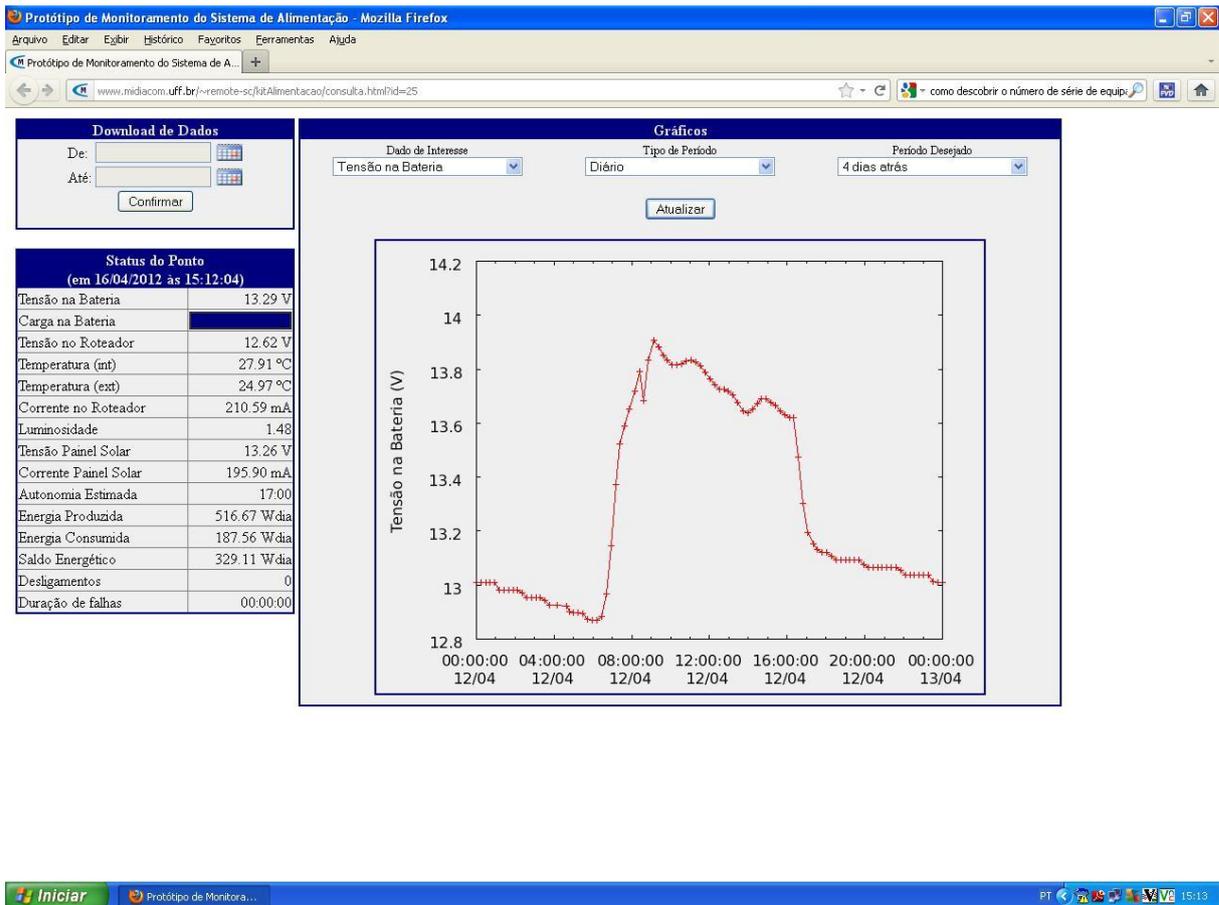


Figura 5.28 - Interface Web para visualização dos dados coletados

Isso permite que sejam feitas várias análises, como por exemplo o comportamento da bateria. Caso esta esteja sendo descarregada muito rapidamente mesmo após um dia ensolarado em que a energia gerada pelo painel solar normalmente é suficiente para carregá-la, isto pode ajudar a identificar um possível problema na bateria.

O campo “*carga na bateria*” indica qual o estado da carga da bateria em função da tensão na mesma. Como ilustrado na tabela 2.4, se a tensão na bateria for 12,42V a bateria terá 80% de sua carga máxima, ou seja, terá sido descarregada cerca de 20%.

Também pode ser traçada a curva de luminosidade do local onde o sistema fotovoltaico foi instalado no período de por exemplo, um mês.

A autonomia do sistema de bateria pode estimar o tempo mínimo que o roteador ficará ligado caso haja problemas na geração de energia pelo painel.

A energia produzida pelo painel solar e a energia consumida pela carga durante o dia também são indicadas, assim como o saldo energético (diferença entre as energias produzida e consumida).

A interface indica quantas vezes o roteador foi desligado (campo “*Desligamentos*”) devido a tensão na bateria ter alcançado o valor $V_{\text{mín}}$. Além disso, o campo “*Duração de falhas*” indica o tempo que roteador ficou desligado.

Além de enviar os dados para serem armazenados no servidor, outra opção seria armazená-los no próprio roteador. Este procedimento poderia evitar a perda de dados caso algum problema impossibilitasse o salvamento no servidor de arquivos, como falta de conectividade na rede *mesh* ou falha no próprio servidor de arquivos. A grande limitação de salvar os dados no roteador ocorre devido à pouca quantidade de espaço de memória disponível (poucos MegaBytes) comparada com o espaço disponível (vários GigaBytes) em disco rígido do computador.

5.6 – ANÁLISE DOS RESULTADOS

Nesta seção é apresentada a análise dos resultados dos testes realizados no protótipo do controlador de carga.

- **Pontos de Ajuste**

Devido aos componentes eletrônicos do mesmo tipo não apresentarem exatamente as mesmas características elétricas, os testes da seção 5.3.3 constataram a necessidade de reprogramar o microcontrolador para corrigir os pontos de ajuste $V_{\text{máx}}$, V_c e $V_{\text{mín}}$ para cada novo controlador montado. Como exemplo, podem ser citados os resistores que apresentam tolerância de até 20% nos seus valores nominais, o que faz com que as medidas nas saídas dos divisores de tensão utilizados na entrada de cada porta conversora analógico-digital não sejam coincidentes, mesmo usando resistores com o mesmo valor nominal. Para evitar a necessidade da reprogramação do microcontrolador sempre que for montado um novo controlador, é preciso que sejam utilizados componentes eletrônicos com as mesmas características elétricas. Como exemplo, nos circuitos divisores de tensão devem ser utilizados resistores com tolerância mais baixa possível. Além disso, o uso de trimpots (resistores ajustáveis) permitem ajustes finos que evitam a necessidade da reprogramação do microcontrolador.

- **Script de conversão**

Em função das características elétricas de alguns componentes eletrônicos não serem iguais, como a tolerância dos resistores explicada anteriormente, o script de conversão precisa ter alguns fatores de conversão ajustados, em especial os que fazem referência aos sensores de tensão e corrente. Apesar de ser o mesmo script, os fatores de conversão podem

necessitar de alguma correção para cada novo CCM montado. Da mesma forma que nos pontos de ajuste, a utilização de componentes com características semelhantes, resistores com tolerância mais baixa e o uso de trimpots evitam a necessidade de alteração nos scripts de conversão.

- **Células LiFePO₄**

A constatação de que as células de LiFePO₄ se descarregam ou carregam em níveis diferentes quando associadas em série, exige uma preocupação adicional para evitar que as mesmas sofram danos irreversíveis. Conforme explicado na seção 5.3.3.2.6, para evitar que as células de LiFePO₄ sejam descarregadas ou carregadas de forma não homogênea podem ser utilizados 3 procedimentos: equalização das células, monitoramento da pior célula e aquisição de células com mesmas características.

Os procedimentos de equalização das células e monitoramento da pior célula exigem hardware adicional com maior complexidade eletrônica. Por isso, é importante que seja feito um levantamento mais detalhado da relação custo-benefício do uso deste tipo de célula em função dos circuitos adicionais necessários.

- **Auto-consumo do CCM**

Os testes de auto-consumo do controlador demonstraram que a desativação dos leds proporciona uma redução de até 30% do consumo de energia e deve ser considerado no momento da montagem do CCM. Os Latching Relay utilizados para acionar as chaves primária e secundária também contribuem para a redução do auto-consumo.

- **Sensor de Luminosidade**

A utilização de filtro no LDR reduziu de forma significativa a saturação no sensor de luminosidade. Para traçar a curva de luminosidade do local onde o painel solar é instalado deve ser usado o filtro.

CAPÍTULO 6 – CONCLUSÃO E TRABALHOS FUTUROS

6.1 Conclusão

A principal contribuição deste trabalho é o desenvolvimento de um controlador de carga inteligente com funções adicionais como coleta informações (sensoriamento) e atuação sobre o sistema fotovoltaico utilizado no projeto ReMoTe.

Buscou-se sempre o desenvolvimento de um dispositivo de baixo custo comparado com os controladores de cargas comerciais e com recursos/funcionalidades não disponíveis nestes. O custo aproximado do CCM está em torno de R\$ 170,00.

A utilização de um microcontrolador como coração de um controlador de carga possibilitou a implementação de funcionalidades importantes devido aos vários elementos encontrados num único chip (memórias, contadores, temporizadores, portas de I/O, portas conversoras analógico-digital, canal PWM e interface de comunicação serial, entre outros).

As funcionalidades implementadas e testadas no CCM são:

- Coleta (sensoriamento) dos parâmetros elétricos do sistema fotovoltaico: tensão na bateria, tensão e corrente no painel solar, temperatura e luminosidade, tipo de bateria, estágio de carga da bateria, estado da carga secundária e nº de reiniciações do roteador pela função WDT ;

- Atuação sobre o sistema fotovoltaico: Alteração do modo de operação do controlador (24h ou das 07:00 às 17:00, aumentando a autonomia do sistema de baterias), Acionamento das cargas (primária e secundária) de forma totalmente independente.

- Uso da função de *WatchDog Timer* para reiniciar o roteador caso ocorra o travamento do mesmo;

- Pontos de ajuste selecionáveis podem permitir a utilização de dois tipos de bateria (chumbo-ácido e LiFePO₄). As baterias chumbo-ácido já podem ser utilizadas. Para utilizar as baterias de LiFePO₄ deverão ser realizados testes mais extensivos e deverá ser solucionado o problema da descarga/carga não homogênea das células.

O presente trabalho permitiu que os pesquisadores do Laboratório Midiacom da Universidade Federal Fluminense obtivessem uma experiência importante em dimensionamento e instalação de dispositivos utilizados em sistemas fotovoltaicos (painel solar, bateria e controlador de carga).

A coleta dos dados dos sensores enviados pelo kit de sensoriamento durante os últimos 12 meses (Fevereiro/2011 a Fevereiro /2012) foi utilizada para criar uma série histórica que será muito importante na análise dos parâmetros elétricos do sistema fotovoltaico.

Os dados da energia produzida pelo painel solar nas 4 estações do ano estão sendo analisados para verificar por exemplo, se o sistema está subdimensionado ou superdimensionado.

As pesquisas e o uso dos microcontroladores possibilitaram a descoberta de novas aplicações para esses dispositivos, principalmente na área de sensoriamento remoto.

Este trabalho nos motivou a continuar pesquisando novos dispositivos e recursos para serem integrados ao CCM com o objetivo de aumentar a eficiência energética do sistema de energia solar das redes em malha instaladas no Campus da Engenharia da Universidade Federal Fluminense.

Como os microcontroladores são flexíveis e possuem vários elementos integrados, acreditamos que as funcionalidades do protótipo do CCM e seus dispositivos eletrônicos, podem e devem ser expandidos no futuro para uma maior eficiência e para adaptá-lo de acordo com as necessidades que se queira suprir.

6.2 Trabalhos futuros

Serão montados 3 controladores para serem instalados na rede *mesh* externa da Uff e serão realizados testes mais extensivos em condições de temperatura não controladas durante um maior período de tempo. Isso servirá para verificar a robustez e a eficiência do protótipo do controlador de carga inteligente apresentado neste trabalho. Esses testes também poderão servir para resolver possíveis problemas que ocorram no hardware e bugs no firmware gravado no microcontrolador,

Além disso, serão feitos:

- Testes mais extensivos com as baterias LiFePO₄ e o estudo da viabilidade da implementação de circuitos adicionais para o monitoramento individual de cada célula;
- Pesquisas para reduzir o auto-consumo através, por exemplo, da utilização de dispositivos eletrônicos mais econômicos principalmente no circuito de chaveamento entre painel solar e bateria. Além disso, serão testados os vários modos de Sleep disponíveis no microcontrolador com o mesmo objetivo de economia energética;
- Adicionar uma funcionalidade à Interface Web de Monitoramento que permita que os comandos remotos sejam executados de forma mais prática. Dessa forma, para enviar um comando remoto para o CCM será necessário apenas clicar num ícone na interface;
- Implementação de um recurso para que a reprogramação do *firmware* seja feita de forma remota;

- Análise dos dados coletados com o objetivo de verificar a variação de temperatura no interior da caixa hermética e se há necessidade de efetuar “compensação térmica” durante a carga das baterias;
- Diminuição das dimensões físicas do controlador de carga;
- Como o hardware do CCM é uma plataforma que permite a criação de outros dispositivos, serão estudados outros usos para o hardware desenvolvido como: o sensoriamento remoto do carro do projeto Mini-Baja, o módulo de controle de acesso as portas do laboratório e a reprogramação do CCM para dar suporte ao “módulo de sobrevivência”.

GLOSSÁRIO

Termos referentes as baterias

Autodescarga - processo espontâneo no qual todas as baterias descarregam gradualmente mesmo não estando em uso, através de processos químicos internos.

Capacidade Nominal - indica a capacidade que a bateria tem de fornecer a amperagem especificada durante uma hora, ou ainda, fornecer uma parte da amperagem em mais horas. A capacidade nominal de uma bateria é dimensionada em Ampere-hora (Ah). Teoricamente, uma bateria de 7Ah tem capacidade para fornecer 7A durante uma hora, 1A durante 7 horas ou 0,35A durante 20 horas. No entanto, a real capacidade da bateria depende das condições de descarga da qual ela é submetida, como intensidade da corrente, tempo em que essa corrente é mantida, voltagem admissível nos terminais da bateria e temperatura. Por exemplo, a energia suprida de uma bateria descarregada a uma temperatura de 40° será diferente de uma bateria descarregada a 25°C.

Ciclo - Considera-se um ciclo de carga e descarga de uma bateria quando esta inicialmente está a 100% da sua capacidade, é submetida a uma determinada profundidade de descarga e após isto volta a ser recarregada a 100% da sua capacidade.

Estado de carga (SOC = State of Charge) - Capacidade disponível em uma bateria ou célula expressa como porcentagem da capacidade nominal. Por exemplo, se 25 Ah foram retirados de uma bateria de capacidade nominal de 100 Ah completamente carregada, o novo estado de carga é 75 %.

Estratificação - Divisão do eletrólito em camadas de diferentes densidades, o que reduz a capacidade da bateria e provoca a corrosão da parte inferior dos eletrodos.

Gaseificação - Geração de gás em um ou mais eletrodos de uma célula, resultante de ação local (autodescarga) ou da eletrólise da água no eletrólito durante o processo de carga. Está relacionada com a sobrecarga e aumenta para elevadas temperaturas de operação. Reduz a eficiência da bateria.

Máxima corrente de recarga - Cada tipo de bateria suporta uma taxa máxima de recarga, ou seja, uma quantidade máxima de corrente que pode suportar no seu processo de recarga. Se este valor for maior que o sugerido pelo fabricante, a bateria pode sofrer mudanças irreversíveis na sua composição química, danificando-a. Por exemplo, as baterias de chumbo-ácido suportam uma taxa de recarga C/3. Isto indica que, por exemplo, uma bateria com capacidade nominal de 7Ah poderia ser carregada no máximo por uma corrente em torno de 2,3 A.

Máxima taxa de descarga - Porcentagem da capacidade da bateria. Indica quanto ela pode se descarregar sem sofrer danos.

Números de Ciclos de vida útil - Número total de ciclos que uma bateria pode ter antes de ficar inoperante. Se uma bateria sofre um grande número de ciclos profundos de carga e descarga, ela terá menor vida útil. A vida útil da bateria é inversamente proporcional a profundidade de descarga sofrida por esta. Se a bateria for submetida a sucessivos ciclos de

descarga profunda de 30%, o tempo de vida útil ficará em torno de 1200 ciclos (> 3 anos). Por outro lado, se for submetida a profundidade de descarga de 50%, a vida útil da bateria será em torno de 600 ciclos (< 2 anos). Na prática, o “fim da vida” é geralmente considerado quando a bateria, estando totalmente carregada, pode fornecer apenas 80% da sua capacidade nominal.

Profundidade de descarga - A profundidade de descarga indica, em termos percentuais, quanto da capacidade nominal da bateria foi retirada a partir do estado de plena carga. Por exemplo, a remoção de 25 Ah de uma bateria de capacidade nominal de 100 Ah resulta em uma profundidade de descarga de 25 %.

Sobrecarga - Ocorre quando uma bateria é submetida à uma taxa de recarga mais alta que ela pode aceitar, ou quando ela já está carregada e ainda continua recebendo carga. O processo de sobrecarga traz danos à bateria, como perda de água do eletrólito e corrosão das grades.

Sulfatação - Se a bateria não for suficientemente recarregada depois da descarga, ocorre a intensificação da formação de cristais de sulfato e não poderão ser convertidos de novo em chumbo ou em óxido de chumbo. Esse fenômeno faz com que, no decorrer do período de operação, a massa ativa diminua e conseqüentemente a capacidade de descarga reduz-se. A parte inferior da célula é afetada de um modo particularmente negativo pela sulfatação, uma vez que nesta zona raramente se dá uma carga total.

REFERÊNCIAS BIBLIOGRÁFICAS

- [01] ReMoTe, **Rede de Monitoramento de linhas de transmissão de energia, Grupo de Pesquisas em Rede Mesh**, Universidade Federal Fluminense, <http://mesh.ic.uff.br>, acesso em Março de 2011.
- [02] ReMesh, **Grupo de Pesquisas em Rede Mesh**, Universidade Federal Fluminense, <http://mesh.ic.uff.br>, acesso em Março de 2011.
- [03] GREEN-WIFI Organization. **Solar Wi-fi Grid Project**. Protótipo disponível em: <http://www.green-wifi.org/projects/gw/Green-WiFi-1-Pager.pdf>. Acesso em Maio 2011.
- [04] BICKET, J. AGUAYO, D. BISWAS, S. **Architecture and Evaluation of an Unplanned 802.11b Mesh Network**, SIGMOBILE, 31-42 p. 2005.
- [05] MERAKI Company. **Meraki Solar**. Disponível em <http://meraki.com/press-releases/2008/12/04/meraki-announces-availability-of-innovative-solar-powered-wifi-device/>. Acesso em 10 de agosto de 2011.
- [06] **SolarMesh**. Disponível em: <http://owl.eng.mcmaster.ca/~todd/SolarMESH/>. Acesso em 8 de Setembro de 2011.
- [07] ALONSO, R. F. “**Sistema autônomo de comunicação sem fio em malha alimentado por energia solar fotovoltaica**”. Dissertação de Mestrado em Engenharia Elétrica, USP, São Paulo, 2009.
- [08] Sunlab Power. **Controlador de Carga Sunlab-SLC**. http://www.sunlab.com.br/manuais/SLC_Manual%20de%20Produto_rev3_2010.pdf. Acesso em 02 setembro de 2011.
- [09] MorningStar Corporation. “**TriStar Controller**”. Janeiro 2009. Disponível em <<http://www.morningstarcorp.com/en/tristar>>. Acesso em 05 Outubro de 2011.
- [10] **COUTINHO, A.A.C.** “Módulo de Sobrevivência para nós de redes mesh sem fio com múltiplos saltos”. Niterói, Dissertação de Mestrado em Engenharia de Telecomunicações, Universidade Federal Fluminense, Rio de Janeiro, 2008.
- [11] CRESESB/CEPEL. Grupo de Trabalho de Energia Solar. **Manual de Engenharia para Sistemas Fotovoltaicos**, 2004. Rio de Janeiro.
- [12] GREENPRO, **Energia Fotovoltaica: Manual sobre Tecnologias, Projeto e Instalação**. 2004. 368 p.
- [13] WIKIPEDIA. **Lithium iron phosphate battery**. Disponível em:<http://en.wikipedia.org/wiki/Lithium_iron_phosphate_battery>. Acesso em: 30 de Outubro de 2011.
- [14] **OLPC Project**. <http://one.laptop.org/about/hardware>. Acesso em: Setembro de 2010.
- [15] **Killacyle Project**. <http://www.killacyle.com/about/>. Acesso em: 05 de Outubro de 2010.

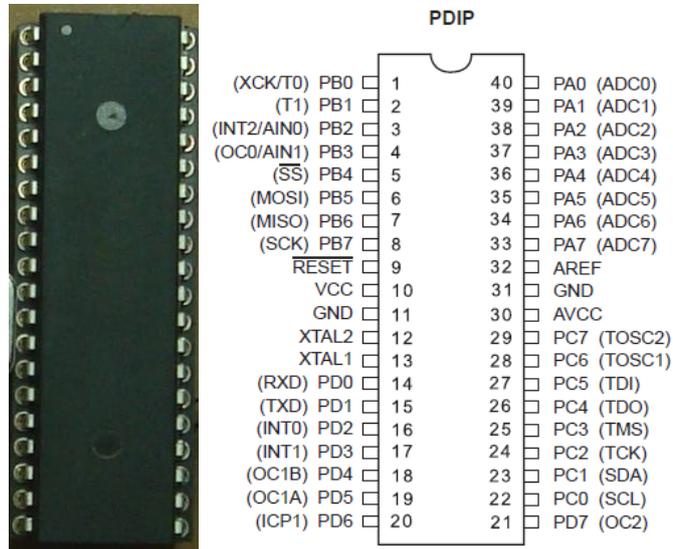
-
- [16] INTERNACIONAL ENERGY AGENCY (IEA) – **Recommended Practices for charge controllers: Use of PV System in Stand-Alone and Island Application**. PVPS, agosto 1998, 68 P. Relatório Técnico IEA PVPS T3-05.
- [17] BORGES, C. **Os poderosos microcontroladores AVR** – APOSTILA - Instituto Federal de Educação, Ciência, e Tecnologia de Santa Catarina. Departamento de Eletrônica, 2009.
- [18] AKKAYA, R. e KULAKSIZ, A. “**A microcontroller-based stand-alone photovoltaic power system for residential appliances**” *Applied Energy*, 2004, vol. 78, p. 419-431. Número 4.
Disponível <http://www.sciencedirect.com/science/article/pii/S0306261903001934>
Acesso em 03 de Maio de 2012.
- [19] **Google Mesh Network** (2007), URL: <http://www.google.com>
- [20] **VMESH Project** (2003), URL: <http://vmesh.inf.uth.gr/>. Acesso em: 03 de Outubro de 2011.
- [21] **MESHNET**. URL: <http://moment.cs.ucsb.edu/meshnet/>. Acesso em 16 Outubro de 2011.
- [22] Draves, Padhye, Zill (2004), “**Routing in Multi-radio, Multi-hop Wireless Mesh Networks**”, ACM MobiCom, Philadelphia, PA, Setembro. Disponível em: <http://research.microsoft.com/mesh/>.
- [23] **ABELÉM, A.; ALBUQUERQUE, C.; SAADE, D.; AGUIAR, E.; DUARTE, J.; DA FONSECA, J.; MAGALHÃES, L.**, “Redes Mesh: Mobilidade, Qualidade de Serviço e Comunicação em Grupo”. SBRC2007”, Belém.
- [24] MAXIM. +5-Powered, **Multichanel RS232 Drivers/Receivers**. Novembro de 1997. Disponível em <<http://www.maxim-ic.com/>>. Acesso em 03 de novembro de 2010.
- [25] Disponível em <http://pewa.panasonic.com/assets/pcsd/catalog/dsp-catalog.pdf> acessado em 10 Julho de 2011.
- [26] AWK, “The GNU Awk User’s Guide”. Disponível em <http://www.gnu.org/software/gawk/manual/gawk.html>, acesso em Junho de 2010.
- [27] MONTEIRO, L. G. “**Estratégia de controle de carga e descarga em sistemas fotovoltaicos domiciliares**”. Dissertação de Mestrado em Energia, Universidade de São Paulo, São Paulo, 2005.
- [28] ST Microeletronics. **LM7805 Voltage Regulator**. Janeiro 2001. Disponível em <http://www.datasheetcatalog.org/datasheet/fairchild/LM7805.pdf>. Acesso em 13 de Janeiro 2009.
- [29] ATMEL. **AVR Microcontroller with 16K Bytes In-System Programmable Flash (ATMEGA16)**”, Atmel Corporation. Outubro de 2002. Disponível em <<http://www.atmel.com/atmel/acrobat/doc2466.pdf>>. Acesso em 30 de setembro. 2009.

-
- [30] DALLAS Semiconductor. **DS1307 – I²C Real-Time Clock**. Setembro de 2001. Disponível em <http://datasheets.maxim-ic.com/en/ds/DS1307.pdf> . Acesso em 06 Agosto 2009.
- [31] NATIONAL Semiconductor. **LM35-Precision Centigrade Temperature Sensors**. Novembro 2000. Disponível em <http://www.national.com/ds/LM/LM35.pdf> Acesso em Março 2010.
- [32] INTERNATIONAL Rectifier. **IR2104 - Half-Bridge Driver**. Fevereiro 2004. Disponível em <http://www.irf.com/product-info/datasheets/data/ir2104.pdf>. Acesso em Julho 2011.
- [33] INTERSIL. **OPERATIONAL AMPLIFIER CA3140** . Julho 2005. Disponível em <http://www.intersil.com/data/fn/fn957.pdf> Acesso em março de 2010.
- [34] SUNROM Technologies. **LDR**. Junho 2008. Disponível em <http://www.sunrom.com/files/3190-datasheet.pdf>. Acesso em Janeiro 2009.
- [35] <http://www.automotivebusiness.com.br/noticia/13711> Acesso em 19 Abril 2012.
- [36] <http://openwrt.org>. Acesso em Maio de 2011.
- [37] ssh. <http://pt.wikipedia.org/wiki/ssh>. Acesso em 20 de Maio de 2012.
- [38] Diego Passos. Interface Web do kit de Sensoriamento. Disponível em <http://www.midiacom.uff.br/~remote-sc/kitAlimentacao/consulta.html?id=25> Acesso em 16 de Abril de 2012.
- [39] WAGNER, R. e SAUER, R. “**Charge strategies for valve-regulated lead/acid batteries in solar power applications**”. Solar Energy Vol. 61, p. 225–230, 1997 Disponível: <http://www.sciencedirect.com/science/article/pii/S0378775300006145> Acesso em 02 de Junho de 2012
- [40] (CHOY, W.W. 2001), “**Method and System For Charge Equalization of Lithium-Ion Batteries**”. Disponível em <http://www.google.com.br/patents?hl=pt-BR&lr=&vid=USPAT6377024&id=81kLAAAEB&oi=fnd&dq=Method+and+System+For+Charge+Equalization+of+Lithium-Ion+Batteries&printsec=abstract>. Acesso em 08 Maio de 2012.
- [41] DAKKAK, M. e HASAN, A. “**A charge Controller Based on Microcontroller In Stand-alone Photovoltaic Systems**”. Energy Procedia 19 (2012) 87- 90. Disponível <http://www.sciencedirect.com/science/article/pii/S1876610212009575> Acesso em 02 de Junho de 2012.
- [42] MASHELENI, H. e CARELSE, X. F. “**Microcontroller-based charge controller for stand-alone photovoltaic systems.**” Solar Energy Vol. 61, p. 225–230, 1997. Disponível <http://www.sciencedirect.com/science/article/pii/S0038092X97000558> Acesso em 04 de Junho de 2012
- [43] DAOUD, A. e MIDOUN, A. “**Fuzzy Control of a Lead Acid Battery Charger**”. J. Electrical Systems, p. 52-59, 2005. Disponível em: <http://en.scientificcommons.org/24718666>. Acesso em 03 de Junho de 2012

- [44] WOODWORTH, J. THOMAS, M. STEVENS, W. HARRINGTON, S. DUNLOP, J. **“Evaluation of the batteries and charge controllers in small stand-alone photovoltaic systems.”**
Disponível em: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=520115.
Acesso em 23 de Junho de 2012
- [45] BOICO, F. LEHMAN, B. e SHUJAE, K. **“Solar Battery Chargers for NiMH Batteries”** Disponível em:
http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4300859
Acesso em 04 de Junho de 2012

ANEXOS

ANEXO 1- Descrição Funcional dos pinos do Atmega16



VCC: Alimentação;

GND: Terra;

RESET: Entrada de Reset. Um nível de tensão baixo neste pino por um período maior que uma determinada largura de pulso produzirá um reset, mesmo que o clock não esteja rodando. Pulsos menores não são garantia para geração de reset;

XTAL1: Entrada para o amplificador inversor do oscilador e entrada para o circuito operacional interno de clock;

XTAL2: Saída do amplificador inversor do oscilador;

AVCC: Pino para a tensão de alimentação do conversor AD. Deve ser externamente conectado ao VCC, mesmo se o A/D não estiver sendo utilizado. Se o A/D for usado deve ser empregado um filtro passa-baixas entre este pino e o VCC.

AREF: Pino para a tensão de referência analógica do conversor AD.

Port A (PA7..PA0): A Port C é uma porta bi-direcional de I/O de 8 bits com resistores internos de pull-up (selecionável para cada bit). Os buffers de saída possuem características simétricas com alta capacidade de fornecer e receber corrente. Como entradas, os pinos que forem externamente colocados em zero fornecerão corrente se os resistores de pull-up

estiverem habilitados. Os pinos ficarão em tri-state quando uma condição de reset estiver ativa, mesmo que o clock não esteja rodando.

Outras funcionalidades da Port A:

Também servem como entradas analógicas do conversor Analógico/Digital (ADCx).

Port B (PB7..PB0): A Port B é uma porta bi-direcional de I/O de 8 bits com resistores internos de pull-up (selecionável para cada bit). Os buffers de saída possuem características simétricas com alta capacidade de fornecer e receber corrente. Como entradas, os pinos que forem externamente colocados em zero fornecerão corrente se os resistores de pull-up estiverem habilitados. Os pinos ficarão em tri-state quando uma condição de reset estiver ativa, mesmo que o clock não esteja rodando.

Outras funcionalidades da Port B:

XCK (USART External Clock Input/Output), **T0** (Entrada externa para o Timer/Counter0), **T1** (Entrada externa para o Timer/Counter1), **INT2**: Interrupção externa número 2, ou Bit de controle para o Timer/Counter1; **AIN0** (Analog Comparator Positive Input), **AIN1** (Analog Comparator Negative Input). Interface SPI (**SCK** - Master Clock Input, **MISO** - Master Input/Slave Output, **MOSI** - Master Output/Slave Input, **SS** - Master Slave Select).

Port C (PC7..PC0): A Port B é uma porta bi-direcional de I/O de 8 bits com resistores internos de pull-up (selecionável para cada bit). Os buffers de saída possuem características simétricas com alta capacidade de fornecer e receber corrente. Como entradas, os pinos que forem externamente colocados em zero fornecerão corrente se os resistores de pull-up estiverem habilitados.

Outras funcionalidades da Port C:

Também serve para a comunicação por dois fios - I2C (**SCL** -Two-wire Serial Bus Clock Line, **SDA** - Two-wire Serial Bus Data Input/Output Line). TOSC2, TOSC1, TDI, TDO, TMS e TCK.

Port D (PD7..PD0): A Port D é uma porta bi-direcional de I/O de 8 bits com resistores internos de pull-up (selecionável para cada bit). Os buffers de saída possuem características simétricas com alta capacidade de fornecer e receber corrente.

¹ Um resistor de pull-up representa um processo pelo qual é garantido que um certo ponto num circuito lógico ficará num nível lógico fixo, não flutuando aleatoriamente. Neste caso, é garantido o nível 1.

² Exatamente como pull up, apenas aqui o nível é 0.

Como entradas, os pinos que forem externamente colocados em zero fornecerão corrente se os resistores de pull-up estiverem habilitados. Os pinos ficarão em tri-state quando uma condição de reset estiver ativa, mesmo que o clock não esteja rodando.

Outras funcionalidades da Port D:

RXD : Receptor da porta serial assíncrona;

TXD : Transmissor da porta serial assíncrona;

INT0: Interrupção externa número 0, ou Bit de controle para o Timer/Counter0;

INT1: Interrupção externa número 1, ou Bit de controle para o Timer/Counter1;

OC1B(Timer/Counter1 Output Compare Match B Output),

OC1A (Timer/Counter1 Output Compare Match A Output),

ICP1 (Timer/Counter1 Input Capture Pin).

As descrições dos pinos foram baseadas no Datasheet ATMEGA16 [29].

ANEXO 2 – Script WDT.sh // Roteador envia byte a cada 1 minuto para o CCM através da
// interface serial indicando que está ativo.

```
#!/bin/ash
while [ 1 ]
do
    echo "O" > /dev/tts/1
    sleep 60 # em segundos
done
```

ANEXO 3 – Script EnviaMedida.sh // Roteador salva no servidor de arquivos a cada 10 segundos os dados dos sensores recebidos do CCM.

```
#!/bin/ash
```

```
INTERVALO=10                # tempo em segundos
USUARIO=remote-sc
SERVIDOR="200.20.10.91 -p 2222" # IP do servidor de arquivos
LOG=alimentacaoid25P40w3bat.txt
PORTA=/dev/tts/1            # porta serial do roteador
/wwww/setserial 4800 $PORTA # configuração da taxa de tx da porta serial

cat $PORTA | while [ 1 ]
do
    read NOVA_MEDIDA
    /usr/bin/ssh -i /etc/dropbear/dropbear_rsa_host_key $SERVIDOR -I $USUARIO "echo
    \"\$(date) * $NOVA_MEDIDA\" >> $LOG"
    sleep $INTERVALO
done
```

ANEXO 4 – Script TrataMedidas.sh // Dados dos sensores salvos no servidor são tratados.

```
#!/bin/bash

PORTA=/dev/tts/1          # porta serial do roteador

stty -F $PORTA ispeed 4800  # configuração da porta serial do roteador em 4800 bps
stty -F $PORTA ospeed 4800  # configuração da porta serial do roteador em 4800 bps

cat $PORTA | awk -W interactive '

BEGIN{
    # Constantes de conversão
    n2v = 5.01/1024;        # Fator comum = Vcc / 1023 (Nº de níveis p/ 10 bits de resolução)
    step1 = n2v * 0 ;      # Porta não usada
    step2 = n2v * 6.05;    # Sensor de Tensão 1 (Não usado)
    step3 = n2v * 1.0;     # Sensor de Luminosidade
    step4 = n2v * 5.95;    # Sensor de Tensão 2 (Não usado)
    step5 = n2v * 5.95;    # Tensão na Bateria
    step6 = n2v * 5.90;    # Tensão no Painel Solar
    step7 = n2v * 1400 ;   # Corrente no Painel Solar
    step8 = n2v * 100;     # Temperatura
    lin = 1;
}

{
    if (lin) {

printf "%s %s %s %s %s %s %s Lum= %.2f X  Vbat= %.2f V  Vps= %.2f V  Ips= %.2f mA
Temp= %d °C  Load2= %s  WTD= %d  ByteRec= %c \n",
$1, $2, $3, $4, $5, $6, $7, $10*step3, $12*step5, $13*step6, $14*step7, $15*step8, $16, $17, $18;
lin = 0;
    }
    else
        lin = 1;
}'
```

ANEXO 5 – FIRMWARE

LISTAGEM DO CÓDIGO FONTE

Anexo 5.1 CHARGER.C // Arquivo principal do CCM: faz leituras dos dados sensoriais, gerencia o processo de carga e descarga da bateria, desliga/liga cargas primária e secundária, alterna entre modos de operação econômico e padrão, reinicializa roteador pela função Wacthdog Timer, envia dados pela serial e recebe dados/comandos do roteador pela serial.

Universidade Federal Fluminense

Curso: Mestrado em Comunicação de Dados e Multimídia

Aluno: JOACIR DE OLIVEIRA SILVA

Orientador: Luiz Cláudio Schara Magalhães

Tese: Controlador de carga microcontrolado para redes em malha (*mesh*) sem fio.

Código: charger.c

* \autores: Joacir Silva joacir@midia.com.uff.br, Rafael Ladislau <rafa.ladis@gmail.com> e Felipe Rolim

* <ferolim@ic.uff.br>

*/

```
#include "charger.h"
```

```
#include "timers.h"
```

```
#include "DIG.h"
```

```
#include "sensors.h"
```

```
#include <avr/io.h>
```

```
#include <compat/deprecated.h>
```

```
#include <inttypes.h>
```

```
#include <string.h>
```

```
#include <stdio.h>
```

```
#include "helper.h"
```

```
#define OFF (0)
```

```
#define BULK (1)
```

```
#define FLOAT (3)
```

```
#define LEAD_ACID_BAT_MIN (391) // 11.5 V (391) p/ Bateria chumbo-ácido
```

```
#define LEAD_ACID_BAT_MAX (492) // 14.4 V (492) p/ Bateria chumbo-ácido
```

```
//#define LEAD_ACID_BAT_VRR (472) // 13.8 V (472) p/ chumbo ácida
```

```
#define LIFEP04_BAT_MIN (384) // 11.4 V(384) p/ baterias LiFePo4
```

```
#define LIFEP04_BAT_MAX (480) // 14.2 V(480) p/ baterias LiFePo4
```

```
//#define LIFEP04_BAT_VRR (472) // 13.8 V(472) p/ baterias LiFePo4 Valor para se manter a bateria quando em Float
```

```
#define PeriodoPWM (3000) // 3 mSseg -> frequencia = 333 Hz
```

```
#define MEAN_VALUE (8)
```

```
#define START_BATTERY_VOLTAGE (423) // 12.5 V (423) com CARGA off p/ chumbo ácida e 12.5 V (423) p/ baterias LIFEP04
```

```
unsigned char nEstado = 0;
```

```
unsigned char nRouterOn = 0;
```

```
unsigned char nU = 1;
```

```
unsigned char nLoad1On = 1;
```

```
unsigned char nLoad2On = 0;
```

```
unsigned char nWTD = 0;
```

```
int BAT_MIN;
```

```
int BAT_MAX;
```

```
int BAT_VRR;
```

```
int counterWTD = 0;
```

```
int counterLoad1 = 0;
```

```
int counterLoad2 = 0;
```

```
unsigned char battery_type;
```

```

void check_battery_type(void){ // Checa o tipo de bateria
    battery_type = DIG_read(DIG_PORT_BATTERY_TYPE);

    if(battery_type == BATTERY_LEAD_ACID) {
        BAT_MIN = LEAD_ACID_BAT_MIN;
        BAT_MAX = LEAD_ACID_BAT_MAX;
        // BAT_VRR = LEAD_ACID_BAT_VRR;
    } else {
        BAT_MIN = LIFEP04_BAT_MIN;
        BAT_MAX = LIFEP04_BAT_MAX;
        // BAT_VRR = LIFEP04_BAT_VRR;
    }
}

unsigned char mode_operation;

void check_mode_operation(void){ // Checa o modo de operação: Padrão (24 horas) ou Econômico (07-17).
    mode_operation = DIG_read(DIG_PORT_MODE_OPERATION);
}

struct portValues
{
    unsigned short i2;
    unsigned short v1;
    unsigned short lum;
    unsigned short v3;
    unsigned short vbat;
    unsigned short vps;
    unsigned short ips;
    unsigned short temp;
};

void disable_pwm(void){ // Desabilita o gerador de sinal PWM
    DIG_write(29, 0);
}

void Off() // Desliga estágio de carga quando tensão no painel solar é menor que tensão na bateria
{
    if(nEstado != OFF)
    {
        nEstado = OFF;
        DIG_write(DIG_PORT_CHARGING, 0); // Desliga Led Amarelo indicador de carga PWM
        disable_pwm();
        DIG_write(DIG_PORT_ENABLE_DRIVER_MOSFET, 0); //Desabilitar DRIVER MOSFET I2104
    }
}

void Bulk() // Estágio de carga em que a máxima corrente do painel é aplicada a bateria
{
    if(nEstado != BULK)
    {
        nEstado = BULK;
        DIG_write(DIG_PORT_CHARGING, 1); // Seta Led Amarelo indicador de carga PWM
        DIG_write(DIG_PORT_ENABLE_DRIVER_MOSFET, 1); // Habilita DRIVER MOSFET IR2104
        timers_setup_pwm(PeriodoPWM,990); // Setado em 99% para ativar ir2104
    }
}

void Float(struct portValues* pValues) // Estágio de carga em que apenas uma parcela da corrente do painel é
// aplicada a bateria pois a mesma está carregada
{
    if(nEstado != FLOAT)

```

```

    {
        nEstado = FLOAT;
        timers_setup_pwm(PeriodoPWM, 100);
        DIG_write(DIG_PORT_CHARGING, 1); // Seta Led Amarelo indicador de carga PWM
        DIG_write(DIG_PORT_ENABLE_DRIVER_MOSFET, 1); // Habilita DRIVER MOSFET IR2104
    }
}

void ClearStructPortValues(struct portValues* pValues) //
{
    memset(pValues, 0, sizeof(struct portValues));
}

char digitToChr(int digit) //
{
    return((char) (digit + 48));
}

void readPortValues(struct portValues* pValues) // Leitura de portas AD
{
    ClearStructPortValues(pValues);

    for(int nInd = 0; nInd < MEAN_VALUE; nInd++)
    {
        (*pValues).i2 += ADC_read(0);
        (*pValues).v1 += ADC_read(1);
        (*pValues).lum += ADC_read(2);
        (*pValues).v3 += ADC_read(3);
        (*pValues).vbat += ADC_read(4);
        (*pValues).vps += ADC_read(5);
        (*pValues).ips += ADC_read(6);
        (*pValues).temp += ADC_read(7);
    }
    (*pValues).i2 /= MEAN_VALUE;
    (*pValues).v1 /= MEAN_VALUE;
    (*pValues).lum /= MEAN_VALUE;
    (*pValues).v3 /= MEAN_VALUE;
    (*pValues).vbat /= MEAN_VALUE;
    (*pValues).vps /= MEAN_VALUE;
    (*pValues).ips /= MEAN_VALUE;
    (*pValues).temp /= MEAN_VALUE;
}

char * ucharToStr(unsigned int ch, int * len) {
    int i = 0;
    static char string[12];
    char temp;

    while(ch >= 10) {
        string[i++] = digitToChr(ch % 10);
        ch = ch / 10;
    }
    string[i++] = digitToChr(ch % 10);
    * len = i;

    for (i = 0; i < (* len)/2; i++) {
        temp = string[i];
        string[i] = string[(* len) - i - 1];
        string[(* len) - i - 1] = temp;
    }
    string[(* len)++] = '\t';
    string[* len] = 0;
}

```

```

        return(string);
    }

void printEstado();

void printValue(unsigned short uValue);

void printValues(struct portValues* pValues, sensors_i2c_clock_datetime * datetime ) // Envia os dados
// pela porta serial
{
    char dataStr[60];

    sprintf(dataStr, "%s, %02hhu/%02hhu/%02hhu - %02hhu:%02hhu:%02hhu: ",
Encontra_Dia_Semana(datetime), datetime->day, datetime->month, datetime->year, datetime->hour, datetime->minute, datetime->second);

    USART_block_write((unsigned char *) dataStr, strlen(dataStr));
    USART_block_write((unsigned char *) "\t", 1);
    printEstado();
    USART_block_write((unsigned char *) (battery_type == BATTERY_LIFEPO4 ? "BAT:LIFEPO \t" :
"BAT:CHUMBO \t"), 12);
    USART_block_write((unsigned char *) (mode_operation == MODE_07_17 ? "MODO:07-17 \t" :
"MODO:24H \t"), 12);
    printValue((*pValues).i2);
    USART_block_write((unsigned char *) "\t", 1);
    printValue((*pValues).v1);
    USART_block_write((unsigned char *) "\t", 1);
    printValue((*pValues).lum);
    USART_block_write((unsigned char *) "\t", 1);
    printValue((*pValues).v3);
    USART_block_write((unsigned char *) "\t", 1);
    printValue((*pValues).vbat);
    USART_block_write((unsigned char *) "\t", 1);
    printValue((*pValues).vps);
    USART_block_write((unsigned char *) "\t", 1);
    printValue((*pValues).ips);
    USART_block_write((unsigned char *) "\t", 1);
    printValue((*pValues).temp);
    USART_block_write((unsigned char *) "\t", 1);
    USART_block_write((unsigned char *) (nLoad2On == 0 ? "OFF \t" : "ON \t"), 4);
    USART_block_write((unsigned char *) "\t", 1);
}

void printValue(unsigned short uValue)
{
    int nLenTemp;
    char* stringTemp;
    stringTemp = ucharToStr(uValue, &nLenTemp);
    USART_block_write((unsigned char *) stringTemp, nLenTemp);
    free(stringTemp);
}

void printEstado() // Envia pela serial o estado de carga do controlador
{
    switch(nEstado)
    {
        case OFF : USART_block_write((unsigned char *)"OFF: ", 7); break;
        case BULK : USART_block_write((unsigned char *)"BULK: ", 7); break;
        case FLOAT : USART_block_write((unsigned char *)"FLOAT: ", 7); break;
    }
}

```

```

    }
}

void AutomaticRouterPowerControl(struct portValues* pValues, sensors_i2c_clock_datetime *datetime)
{
    // Liga ou desliga roteador em função da tensão da bateria e dos modos de operação Padrão e Economico

    if(nRouterOn)    // Desliga roteador se Vbat < 11.5 V ou se modo estiver setado para economico
    {
        {
            if ((pValues->vbat < BAT_MIN) || (mode_operation == MODE_07_17 && (datetime->hour < 7 ||
datetime->hour >= 17)))
            {
                nRouterOn = 0;
                DIG_TURN_OFF_ROUTER();
                nWTD = 0;
            }
        }

        else if(!nRouterOn && pValues->vbat > START_BATTERY_VOLTAGE && (mode_operation ==
MODE_24HORAS || (mode_operation == MODE_07_17 && (datetime->hour >= 7 && datetime->hour <
17))))
            // Liga roteador se Vbat > 12.6 V ou se modo estiver setado em padrão
            {
                nRouterOn = 1;
                DIG_TURN_ON_ROUTER();
                nWTD = 1;
            }
    }
}

void InitialBatteryCheck()    // Checa estado inicial da bateria e liga roteador se Vbat > Vmín
{
    struct portValues values;
    readPortValues(&values);

    if(values.vbat > BAT_MIN)
    {
        DIG_TURN_ON_ROUTER();
        nRouterOn = 1;
        nWTD = 1;
    }
}

void data_handler(char received_byte)    // Executa comandos remotos em função do byte recebido
{
    switch(received_byte)
    {
        case '1': // Desliga Carga Primária por 5 minutos
        {
            DIG_TURN_OFF_ROUTER();
            counterLoad1 = 0 ;
            nRouterOn = 1;
            nWTD = 0;
            nLoad1On = 0;
        };break;
        case '2':    // Liga Carga Secundária por 5 minutos
        {
            DIG_TURN_ON_SECONDARY_LOAD();
            counterLoad2 = 0;
            nLoad2On = 1;
        };break;

        case '3': // Desliga Carga Secundária
        {

```

```

        DIG_TURN_OFF_SECUNDARY_LOAD();
        counterLoad2 = 0;
        nLoad2On = 0;
    };break;

    case 'O': //Reinicializa contador da função WTD
    {
        counterWTD = 0;
    };break;

    case 'E': // Alterna para Modo de operação Econômico (07:00-17:00)
    {
        mode_operation = (1);
    };break;

    case 'P': // Alterna para Modo de operação Padrão (24 horas)
    {
        mode_operation = (0);
        InitialBatteryCheck();
    };break;

    default:
    {
    }
}
}

void charge(void) // Loop Principal
{
    struct portValues values;
    sensors_i2c_clock_datetime datetime;
    uint8_t received_byte;
    DIG_TURN_OFF_SECUNDARY_LOAD();
    InitialBatteryCheck();
    nEstado = OFF;
    int NumRebootWTD = 0;

    while(1)
    {
        received_byte = 'X'; // Byte Padrão enviado pela serial

        if(USART_new_data_for_reading())
        {
            received_byte = USART_read();
            data_handler(received_byte);
        }

        if(!nWTD)
        {
            counterWTD = 0;
        }
        counterWTD++;
        counterLoad1++;
        counterLoad2++;

        if((nWTD)&&(counterWTD > 50)) // Reinicializa roteador pela função WTD
        {
            DIG_TURN_OFF_ROUTER();
            _delay_ms(1000);
            DIG_TURN_ON_ROUTER();
            nRouterOn = 1;
            nWTD = 1;
        }
    }
}

```

```

        counterWTD = 0;
        NumRebootWTD = NumRebootWTD + 1 ;
    }

if(!nWTD)&&(counterLoad1>60)&&!nLoad1On) // Liga roteador após 5 minutos que é desligado
{
    DIG_TURN_ON_ROUTER();
    nLoad1On = 1 ;
    nWTD = 1;
}

if((nLoad2On)&&(counterLoad2 > 70) ) // Desliga carga secundária após 5 minutos
{
    DIG_TURN_OFF_SECUNDARY_LOAD();
    nLoad2On = 0;
}

readPortValues(&values);
sensors_read_i2c_clock(&datetime);
printValues(&values, &datetime);

char counter3String[50];
sprintf(counter3String, "%d", NumRebootWTD);
USART_block_write((unsigned char *) counter3String, (unsigned char)strlen(counter3String));
USART_block_write((unsigned char *) "\t", 1);
USART_write(received_byte);

if((values.vps) < values.vbat)
{
    Off();
}
else
    if(((nEstado != FLOAT) && (values.vbat < BAT_MAX)) || (values.vbat < BAT_VRR))
        {
            Bulk();
        }

    else
        if (values.vbat >= BAT_MAX)
            {
                Float(&values);
            }

AutomaticRouterPowerControl(&values, &datetime);
USART_block_write((unsigned char *) "\r\n", 2);
_delay_ms(1000);

}
}

```

Anexo 5.2 CHARGER.H // Declaração e definição das funções do controlador.

```
* \author Joacir Silva <joacir@midia.com.uff.br>
```

```
* \author Rafael Ladislau <ladislau@ic.uff.br>
```

```

#ifndef __CHARGER_H__
#define __CHARGER_H__
#include <avr/interrupt.h>
#include <avr/io.h>
#include "USART.h"
#include "defs.h"
#include <util/delay.h>

```

```
#include "ADC.h"
#include "timers.h"
#include <stdlib.h>
#include <string.h>

#define BATTERY_LEAD_ACID (0) // Constante que identifica bateria Chumbo-ácido
#define BATTERY_LIFEPO4 (1) // Constante que identifica bateria LiFePo4
#define MODE_24HORAS (0) // Constante que identifica modo de operação Padrão (24h)
#define MODE_07_17 (1) // Constante que identifica modo de operação Econômico (07-17)

extern unsigned char mode_operation; // Variável que define modo de operação (Padrão ou Econômico)
extern unsigned char battery_type; // Variável que define o tipo de bateria (Chumbo-ácido ou LiFePo4)

void disable_pwm(void); // Desabilita gerador de sinal PWM
void charge(void); // Função principal
void check_battery_type(void); // Checa qual bateria foi selecionada (Chumbo-ácido ou LiFePo4)
void check_mode_operation(void); // Checa o modo de operação do controlador (Padrão ou Econômico)

#endif // __CHARGER_H__
```

Anexo 5.3 DIG.h // Declaração e definição de funções de manipulação das portas digitais do Atmega16.

```
* \author Diego Passos <dpassos@ic.uff.br>
```

```
#ifndef __DIG_H__
#define __DIG_H__
```

```
/*
```

O ID de uma porta digital é formada por 2 partes, sendo uma parte para o port (PORTA, PORTB, PORTC, PORTD) e um índice (0 a 7). São necessários 2 bits para o tipo de port e 3 bits para o índice. O sexto bit será usado para o tipo de operação (escrita ou leitura). O sétimo bit será usado para o valor a ser escrito (se for aplicado).

```
/*
```

```
typedef enum {
```

```
    DIG_PORTA,
    DIG_PORTB,
    DIG_PORTC,
    DIG_PORTD
```

```
} t_dig_ports;
```

```
// Retorna a porta (PORTA, PORTB, PORTC ou PORTD)
#define DIG_GET_PORT(x)          ((x >> 3) & 0x03)
```

```
// Retorna o índice da porta (0 a 7)
#define DIG_GET_PORT_INDEX(x)    (x & 0x07)
```

```
// Porta digital de saída usada para ligar a carga primária (roteador)
#define DIG_PORT_ROUTER_POWER_ON    (8) // PB0
```

```
// Porta digital de saída usada para desligar a carga primária (roteador)
#define DIG_PORT_ROUTER_POWER_OFF    (9) // PB1
```

```
// Porta digital de saída usada para ligar a carga secundária
#define DIG_PORT_SECUNDARY_LOAD_POWER_ON    (10) // PB2
```

```
// Porta digital de saída usada para desligar a carga secundária
#define DIG_PORT_SECUNDARY_LOAD_POWER_OFF    (11) // PB3
```

```
// Porta digital de entrada usada para definir o modo de operação do controlador:
```

```
// 0 p/ modo padrão (24 h)
```

```
// 1 p/ modo economico (07-17)
```

```
#define DIG_PORT_MODE_OPERATION    (18) // PC2
```

```
// Porta digital de entrada usada para definir o tipo de bateria:
```

```
// 0 p/ bateria LIFEP04
```

```
// 1 p/ bateria Chumbo-ácido
```

```
#define DIG_PORT_BATTERY_TYPE    (31) // PD7
```

```
// Porta digital de saída utilizada para ligar LED que indica que bateria está sendo carregada
```

```
#define DIG_PORT_CHARGING    (19) // PC3
```

```
// Porta digital de saída utilizada para ligar LED que indica que carga primária (roteador) desligada
```

```
#define DIG_PORT_ROUTER_LED    (20) // PC4
```

```
// Porta digital de saída utilizada para habilitar o Driver_Mosfet IR2104
```

```
#define DIG_PORT_ENABLE_DRIVER_MOSFET    (30) // PD6
```

```
// Liga a carga primária e desliga LED indicador de carga primária desligada
```

```
#define DIG_TURN_ON_ROUTER ( )
```

```
{ DIG_write(DIG_PORT_ROUTER_POWER_ON, 1);
  _delay_ms(100); \
  DIG_write(DIG_PORT_ROUTER_POWER_ON, 0);
  DIG_write(DIG_PORT_ROUTER_LED, 0);
}

// Desliga a carga primária e liga LED indicador de carga primária desligada
#define DIG_TURN_OFF_ROUTER ( )
  { DIG_write(DIG_PORT_ROUTER_POWER_OFF, 1);
    _delay_ms(100); \
    DIG_write(DIG_PORT_ROUTER_POWER_OFF, 0);
    DIG_write(DIG_PORT_ROUTER_LED, 1);
  }

// Liga a carga secundária
#define DIG_TURN_ON_SECONDARY_LOAD( )
  { DIG_write(DIG_PORT_SECONDARY_LOAD_POWER_ON, 1);
    _delay_ms(100);
    DIG_write(DIG_PORT_SECONDARY_LOAD_POWER_ON, 0);
  }

// Desliga a carga secundária
#define DIG_TURN_OFF_SECONDARY_LOAD( )
  { DIG_write(DIG_PORT_SECONDARY_LOAD_POWER_OFF, 1);
    _delay_ms(100);
    DIG_write(DIG_PORT_SECONDARY_LOAD_POWER_OFF, 0);
  }

// Leitura da porta digital especificada
unsigned char DIG_read(unsigned char address);

// Escrita do bit especificado na porta digital especificada
void DIG_write(unsigned char address, unsigned char bit);

#endif
```

Anexo 5.4 DIG.c // Implementa as funções de manipulação das portas digitais.

```
* \author Diego Passos <dpassos@ic.uff.br>
```

```
#include <avr/io.h>
```

```
#include "DIG.h"
```

```
unsigned char DIG_read(unsigned char address) { // Leitura da porta digital especificada
    t_dig_ports port;
    unsigned char pin_index;
    unsigned char reading = 0xFF;
    port = DIG_GET_PORT(address);
    pin_index = DIG_GET_PORT_INDEX(address);

    switch(port) {
        case DIG_PORTA:
            DDRA &= ~(1 << pin_index);
            reading = (PINA & (1 << pin_index)) >> pin_index;
            break ;
        case DIG_PORTB:
            DDRB &= ~(1 << pin_index);
            reading = (PINB & (1 << pin_index)) >> pin_index;
            break ;
        case DIG_PORTC:
            DDRC &= ~(1 << pin_index);
            reading = (PINC & (1 << pin_index)) >> pin_index;
            break ;
        case DIG_PORTD:
            DDRD &= ~(1 << pin_index);
            reading = (PIND & (1 << pin_index)) >> pin_index;
            break ;
    }
    return(reading);
}
```

```
void DIG_write(unsigned char address, unsigned char bit){ // Escrita do bit especificado na porta digital
    t_dig_ports port;
    unsigned char pin_index;
    port = DIG_GET_PORT(address);
    pin_index = DIG_GET_PORT_INDEX(address);

    switch(port) {
        case DIG_PORTA:
            DDRA |= (1 << pin_index);
            PORTA = (PORTA & ~(1 << pin_index)) | ((bit & 0x01) << pin_index);
            break ;
        case DIG_PORTB:
            DDRB |= (1 << pin_index);
            PORTB = (PORTB & ~(1 << pin_index)) | ((bit & 0x01) << pin_index);
            break ;
        case DIG_PORTC:
            DDRC |= (1 << pin_index);
            PORTC = (PORTC & ~(1 << pin_index)) | ((bit & 0x01) << pin_index);
            break ;
        case DIG_PORTD:
            DDRD |= (1 << pin_index);
            PORTD = (PORTD & ~(1 << pin_index)) | ((bit & 0x01) << pin_index);
            break ;
    }
}
```

Anexo 5.5 USART.h // Declaração e definição de funções de manipulação da porta serial (USART).

```

* \author Diego Passos <dpassos@ic.uff.br>

#ifndef __USART_H__
#define __USART_H__

#include <avr/io.h>

// função que inicializa a USART
void USART_init(void);

// Escreve uma byte através da USART
void USART_write(unsigned char msg);

// Lê um byte através da USART
unsigned char USART_read(void);

// Envia uma sequencia de bytes através da USART
void USART_block_write(unsigned char * msg, unsigned char msg_size);

// Lê uma sequencia de bytes da USART. Onde msg é o buffer que armazena os bytes para serem recebidos e
msg_size é o número de bytes a ser lido. Esta função é bloqueante.
void USART_block_read(unsigned char * msg, unsigned char msg_size);

// Escreve uma variável do tipo genérico para a porta serial
#define USART_type_write(x)    USART_block_write((unsigned char *) (& (x)), sizeof(x))

// Escreve uma variável do tipo short para a porta serial
#define USART_short_write(x)  USART_type_write(x)

// Escreve uma variável do tipo int para a porta serial
#define USART_int_write(x)    USART_type_write(x)

// Escreve uma variável do tipo float para a porta serial
#define USART_float_write(x)  USART_type_write(x)

// Lê uma variável do tipo genérico para a porta serial
#define USART_type_read(x)    USART_block_read((unsigned char *) (& (x)), sizeof(x))

// Lê uma variável do tipo short para a porta serial
#define USART_short_read(x)   USART_type_read(x)

// Lê uma variável do tipo int para a porta serial
#define USART_int_read(x)     USART_type_read(x)

// Lê uma variável do tipo float para a porta serial
#define USART_float_read(x)   USART_type_read(x)

// Verifica se tem dado pronto para ser lido na porta serial
#define USART_new_data_for_reading() (UCSRA & (1 << RXC))

#endif

```

Anexo 5.6 USART.c // Implementa as rotinas de manipulação da porta serial (USART).

```

* \author Diego Passos <dpassos@ic.uff.br>

#include "USART.h"
#include "defs.h"

// Esta implementação define os seguintes parâmetros para a comunicação serial :
* - Baud Rate: 4800;
* - Data bits: 8;
* - Parity: none;
* - Stop bits: 1; and
* - Flow control: none.

// Testa se a USART é disponível
#define USART_IS_AVAILABLE(UCSRA & (1 << UDRE))

// Checa se o dado recebido está pronto
#define USART_IS_DATA_READY (UCSRA & (1 << RXC))

// Constantes para configurar o BAUDRATE desejado, de acordo com o clock da CPU.

#ifndef F_CPU
#error "Macro F_CPU must be defined as the CPU clock in Hz."
#else
#define BAUD_PRESCALE (((F_CPU / (USART_BAUDRATE * 16UL))) - 1)
#endif

void USART_init(void) { // Inicializa a USART
    UBRRL = BAUD_PRESCALE;
    UBRRH = (BAUD_PRESCALE >> 8);
    UCSRB = (1 << RXEN) | (1 << TXEN);
}

void USART_write(unsigned char msg) { // Função de escrita de um byte
    while(!USART_IS_AVAILABLE);
    UDR = msg;
}

unsigned char USART_read(void) { // Função de leitura de um byte
    while(!USART_IS_DATA_READY);
    return(UDR);
}

void USART_block_write(unsigned char * msg, unsigned char msg_size) { // Escreve múltiplos bytes
    while(msg_size--) {
        USART_write(* (msg++));
    }
}

void USART_block_read(unsigned char * msg, unsigned char msg_size) { // Ler múltiplos bytes
    while(msg_size--) {
        *(msg++) = USART_read();
    }
}

```

Anexo 5.7 ADC.h // Declaração e definição das funções de manipulação da portas analógicas.

```
* \author Diego Passos <dpassos@ic.uff.br>
```

```
#ifndef __ADC_H__  
#define __ADC_H__
```

```
// Número de portas A/D disponíveis no microcontrolador Atmega16  
#define ADC_NUM_PORTS 8
```

```
// Inicialização do hardware ADC  
void ADC_init(void);
```

```
// Faz a conversão da porta especificada e retorna o valor entre 0 e 1023 (10 bits de resolução).  
unsigned short ADC_read(unsigned char port);
```

```
// Faz a conversão de várias portas A/D  
void ADC_block_read(unsigned short * adc_output);
```

```
#endif
```

Anexo 5.8 ADC.c // Implementa as rotinas para manipulação das portas analógicas.

```
* \author Diego Passos <dpassos@ic.uff.br>

#include <avr/io.h>
#include "ADC.h"
#include "defs.h"

// Máxima frequência de operação do ADC. Pelo datasheet, o ADC não pode operar acima de 200 Khz e a faixa
de frequência ótima é [50:200] (KHz).

#define ADC_MAX_FREQUENCY      400000UL

void ADC_init(void) { // Inicialização do hardware ADC

    unsigned char prescaler, division_factor;

#ifdef F_CPU
    #error "Macro F_CPU must be defined as the CPU clock in Hz."
#else
    #if (F_CPU > ADC_MAX_FREQUENCY * 128)
        #warning "CPU frequency is too high. ADC will operate above its optimal frequency."
        prescaler = 0x07;
    #else

        prescaler = 0;
        division_factor = (F_CPU / ADC_MAX_FREQUENCY);
        while(division_factor) {
            division_factor >>= 1;
            prescaler++;
        }
    #endif
#endif
    ADCSRA = prescaler;
}

unsigned short ADC_read(unsigned char port) { // Faz a conversão da porta especificada e retorna valor
// entre 0 e 1023 (10 bits de resolução).

    unsigned short result;
    ADMUX = port;
    ADCSRA |= (1 << ADEN);
    ADCSRA |= (1 << ADSC);
    while(ADCSRA & (1 << ADSC));
    result = ADCL + (((unsigned short) ADCH) << 8);
    return(result);
}

void ADC_block_read(unsigned short * adc_output) { // Faz a conversão de várias portas A/D
    unsigned char port_index;
    for (port_index = 0; port_index < ADC_NUM_PORTS; port_index++) {
        adc_output[port_index] = ADC_read(port_index);
    }
}
```

Anexo 5.9 I2C.h // Declaração das funções do protocolo de comunicação I2C do μ controlador Atmega16.

```
* \author Diego Passos <dpassos@ic.uff.br>
```

```
#ifndef __I2C_H__
#define __I2C_H__
#define I2C_HIGH_ABSTRACTION_LAYER

// Inicializa o hardware I2C do microcontrolador e seta a frequência do barramento em 100 Khz
void I2C_init(void);

// Envia um sinal de START/RESTART para o controle do barramento de comunicação
void I2C_start(void);

// Envia um sinal de STOP para liberar o controle do barramento de comunicação
void I2C_stop(void);

// Ler um byte do barramento I2C
unsigned char I2C_read(unsigned char ack);

// Escreve um byte no barramento I2C
unsigned char I2C_write(unsigned char msg);

// Funções que permite uma forma mais fácil para uso de dispositivos I2C. Estas funções podem ser
// habilitadas/desabilitadas durante a compilação, através da definição I2C_HIGH_ABSTRACTION_LAYER.

#ifdef I2C_HIGH_ABSTRACTION_LAYER

// Escreve uma sequência de bytes para um dispositivo I2C escravo
unsigned char I2C_block_write(unsigned char addr, unsigned char * msg, unsigned char msg_size);

// Ler uma sequência de bytes de um dispositivo I2C escravo
unsigned char I2C_block_read(unsigned char addr, unsigned char * msg, unsigned char msg_size);

#endif

// Define o endereço default para o relógio I2C DS1307
#define I2C_CLOCK_ADDRESS 0xD0

#endif
```

Anexo 5.10 I2C.c // Implementa o protocolo de comunicação I2C do μ controlador Atmega16.

```

* \author Diego Passos <dpassos@ic.uff.br>

#include <avr/io.h>      // For definitions of ports, registers and bits.
#include <util/twi.h>    // For definitions of error codes and status of TWI.

#include "I2C.h"
#include "defs.h" // Definições comuns (como o clock da CPU).

// Checa se o bit TWINT está setado e se está pronto para o próximo comando
#define TWI_IS_AVAILABLE (TWCR & (1 << TWINT))

void I2C_init(void) { // Inicializa o hardware I2C e seta a frequência do barramento em 100 KHz

    TWSR = 0;

    #ifndef F_CPU
    #error "Macro F_CPU must be defined as the CPU clock in Hz."
    #else

    #if F_CPU < 3600000UL
    #warning "CPU frequency is too low. It must be, at least, 3.6MHz."
    #endif
    TWBR = (F_CPU / 100000UL - 16) / 2;
    #endif
}

void I2C_start(void) { // Envia um sinal de START/RESTART para o controle do barramento de comunicação

    unsigned char status;
    while(1) {
        TWCR = (1 << TWINT) | (1 << TWSTA) | (1 << TWEN);
        while(!TWI_IS_AVAILABLE);
        status = TW_STATUS;
        if (status == TW_REP_START || status == TW_START) return ;
    }
}

void I2C_stop(void) { // Envia um sinal de STOP para liberar o controle do barramento de comunicação
    while(!TWI_IS_AVAILABLE);
    TWCR = (1 << TWINT) | (1 << TWSTO) | (1 << TWEN);
}

unsigned char I2C_read(unsigned char ack) { // Ler um byte do barramento I2C
    unsigned char msg;
    while(!TWI_IS_AVAILABLE);

    if (ack) TWCR = (1 << TWINT) | (1 << TWEN) | (1 << TWEA);
    else TWCR = (1 << TWINT) | (1 << TWEN);

    while(!TWI_IS_AVAILABLE);

    msg = TWDR;

    return(msg);
}

unsigned char I2C_write(unsigned char msg) { // Escreve um byte no barramento I2C

```

```

    unsigned char status;
    while(!TWI_IS_AVAILABLE);
    TWDR = msg;
    TWCR = (1 << TWINT) | (1 << TWEN);

    while(!TWI_IS_AVAILABLE);
    status = TW_STATUS;
    if (status == TW_MT_DATA_NACK || status == TW_MT_SLA_NACK) return(0);
    if (status == TW_MT_DATA_ACK || status == TW_MT_SLA_ACK) return(1);
    return(-1);
}

#ifdef I2C_HIGH_ABSTRACTION_LAYER

unsigned char I2C_block_write(unsigned char addr, unsigned char * msg, unsigned char msg_size) {

    I2C_start();

    if (I2C_write(addr | TW_WRITE) < 1) {
        return(-1);
    }

    while(msg_size--) {

        if (I2C_write(* (msg++)) == -1) {
            return(-1);
        }
    }

    I2C_stop();
    return(0);
}

unsigned char I2C_block_read(unsigned char addr, unsigned char * msg, unsigned char msg_size) {

    I2C_start();

    if (I2C_write(addr | TW_READ) < 1) {
        return(-1);
    }

    while(msg_size--) {

        * (msg++) = I2C_read(msg_size);
    }

    I2C_stop();

    /

    return(0);
}

#endif

```

Anexo 5.11 sensors.h // Declaração da estrutura e funções de manipulação de escrita/leitura de dados de dispositivos I2C. Este arquivo contém as funções para manipulação do relógio I2C DS1307 utilizado no controlador. Caso sejam utilizados outros dispositivos I2C, suas funções podem ser declaradas aqui.

* \author Diego Passos <dpassos@ic.uff.br>

```
#ifndef __SENSORS_H__
#define __SENSORS_H__
```

```
#include<inttypes.h>
```

```
// Estrutura de dados para o retorno das leituras do relógio I2C.
// Os valores podem ser em formato normal ou em formato BCD.
```

```
typedef struct {
    uint8_t second;
    uint8_t minute;
    uint8_t hour;
    uint8_t day_of_week;
    uint8_t day;
    uint8_t month;
    uint8_t year;
} __attribute__((__packed__)) sensors_i2c_clock_datetime;
```

```
// Leitura do relógio i2c. Os valores setados no buffer estão em formato normal (não BCD)
void sensors_read_i2c_clock(sensors_i2c_clock_datetime *buffer);
```

```
// Escrita do relógio i2c. Os valores setados no buffer deverão estar em formato BCD.
void sensors_set_i2c_clock(sensors_i2c_clock_datetime *buffer);
```

```
#endif
```

Anexo 5.12 sensors.c // Implementa as funções de manipulação de leitura/escrita de dispositivos I2C.
 // Este arquivo contem as funções para manipulação do relógio I2C DS1307
 // utilizado no controlador. Caso sejam utilizados outros dispositivos I2C,
 // suas funções de manipulação podem ser declaradas aqui.

```
* \author Diego Passos <dpassos@ic.uff.br>
```

```
#include "sensors.h"
#include "I2C.h"
#include "DIG.h"
```

```
#include "USART.h"
#include "helper.h"
#include <alloca.h>
```

```
void sensors_read_i2c_clock(sensors_i2c_clock_datetime *buffer) { // Lê dados do relógio I2C DS1307
```

```
    unsigned char byte = 0;
    unsigned char * msg_buffer;
    msg_buffer = (unsigned char *) alloca(7);
```

```
    I2C_block_write(I2C_CLOCK_ADDRESS, &byte, sizeof(byte));
    I2C_block_read(I2C_CLOCK_ADDRESS, msg_buffer, 7);
```

```
    buffer->year = BCD2Byte( msg_buffer[6] );
    buffer->month = BCD2Byte( msg_buffer[5] );
    buffer->day = BCD2Byte( msg_buffer[4] );
    buffer->hour = BCD2Byte( msg_buffer[2] );
    buffer->minute = BCD2Byte( msg_buffer[1] );
    buffer->second = BCD2Byte( msg_buffer[0] );
    buffer->day_of_week = BCD2Byte( msg_buffer[3] );
```

```
}
```

```
void sensors_set_i2c_clock(sensors_i2c_clock_datetime *buffer) { // Escreve dados do relógio I2C DS1307
```

```
    unsigned char send_buffer[8];
```

```
    send_buffer[0] = 0;
    send_buffer[1] = buffer->second;
    send_buffer[2] = buffer->minute;
    send_buffer[3] = buffer->hour;
    send_buffer[4] = buffer->day_of_week;
    send_buffer[5] = buffer->day;
    send_buffer[6] = buffer->month;
    send_buffer[7] = buffer->year;
```

```
    I2C_block_write(I2C_CLOCK_ADDRESS, (unsigned char *) send_buffer, sizeof(send_buffer));
```

```
}
```

Anexo 5.13 helper.h // Declaração de funções com utilizadas diversas.

```
* \author Rafael Ladislau <ladislau@ic.uff.br>
```

```
#ifndef __HELPER_H__  
#define __HELPER_H__
```

```
#include <inttypes.h>  
#include "sensors.h"
```

```
// Converte um byte BCD para byte normal  
uint8_t BCD2Byte(uint8_t bcd);
```

```
// Converte um byte normal para um byte BCD  
uint8_t Byte2BCD(uint8_t _byte);
```

```
// Compara dados. Se dado1 é menor que dado2, retorna -1. Se dado1 é igual a dado2, retorna 0.  
uint8_t compare_dates(sensors_i2c_clock_datetime * date1, sensors_i2c_clock_datetime * date2);
```

```
// Encontra dia da semana  
char * Encontra_Dia_Semana(sensors_i2c_clock_datetime * datetime);
```

```
#endif // __HELPER_H__
```

Anexo 5.15 defs.h // Definição de constantes de configuração do microcontrolador Atmega16.
// Basicamente contem informação específica do hardware, como clock da CPU e
// BaudRate da USART

* \author Diego Passos <dpassos@ic.uff.br>

```
#ifndef __DEFS_H__  
#define __DEFS_H__
```

// Clock da CPU

```
#ifndef F_CPU  
#define F_CPU 4000000UL // Cristal de 4 megahertz  
#endif
```

// Baud rate da USART configurada em 4800 bps

```
#ifndef USART_BAUDRATE  
#define USART_BAUDRATE 4800  
#endif
```

```
#endif
```

Anexo 5.16 timers.h // Define constantes e funções de manipulação de temporizadores disponíveis
// no microcontrolador Atmega16.

* \author Diego Passos <dpassos@ic.uff.br>

#ifndef __TIMERS_H__
#define __TIMERS_H__

// Precisão de cada temporizador em termos de número de ciclos.
// Tamanho do contador de cada temporizador (registrador).

// Valor máximo do contador do temporizador 0 (timer0)
Define TIMER0_PRECISION 256

// Valor máximo do contador do temporizador 1A (timer1A)
Define TIMER1A_PRECISION 65536

// Valor máximo do contador do temporizador 1B (timer1B)
Define TIMER1B_PRECISION 65536

// Valor máximo do contador do temporizador 2 (timer2)
Define TIMER2_PRECISION 256

// Apesar de ser usado o timer1 para implementar PWM, no modo que foi usado, sua precisão é de 8 bits.
// Mesmo no modo mais preciso, so é possível utilizar PWM com 10 bits de precisão.
// Valor máximo do contador do PWM (timer1)
Define PWM_PRECISION 256

// Comprimento (em ticks) do temporizador emulado por software

// Comprimento do temporizador (em ticks) do timer1A.
Define TIMER0_SOFTWARE_TICK 194

// Comprimento do temporizador (em ticks) do timer1A.
Define TIMER1A_SOFTWARE_TICK 194

// Comprimento do temporizador (em ticks) do timer1B.
Define TIMER1B_SOFTWARE_TICK 194

// Comprimento do temporizador (em ticks) do timer2.
Define TIMER2_SOFTWARE_TICK 194

// Declaração de funções

//Configura o temporizador 0 (timer0)
void timers_setup_timer0 vazio (unsigned long microseconds, void (* callback) (void));

// Configura o temporizador 1A (timer1A).
// Os temporizadores 1A e 1B são acoplados e podem ser usados juntos.
// Porém quando o timer1 é usado para PWM, não se deve utilizar Timer1A ou timer1B.
void timers_setup_timer1A (unsigned long microseconds, void (* callback) (void));

// Configura o temporizador 1B (timer1B).
void timers_setup_timer1B (unsigned long microseconds, void (* callback) (void));

// Configura o temporizador 2 (timer2).
void timers_setup_timer2 vazio (unsigned long microseconds, void (* callback) (void));

```
// Função para criação de um sinal PWM na porta D5. Período da onda em milisegundos e o percentual do duty-
// cycle multiplicado por 10 para permitir maior precisão.
void timers_setup_pwm (unsigned short pwm_period, pwm_duty unsigned short);

// Função que alterna estado (ativado/desativado) do temporizador0 (timer0)
void timers_toggle_timer0 (void);

// Função que alterna estado (ativado/desativado) do temporizador1 (timer1)
// Uma vez que os temporizadores 1A e 1B são acoplados, esta função desativa ambos os temporizadores.
void timers_toggle_timer1 (void);

// Função que alterna estado (ativado/desativado) do temporizador2 (timer2)
void timers_toggle_timer2 (void);

// Função que alterna estado (ativado/desativado) do gerador de forma de onda PWM (que usa timer1).
# Define timers_toggle_pwm timers_toggle_timer1

# Endif
```

Anexo 5.17 timers.c // Implementa as rotinas de utilização dos temporizadores disponíveis no Atmega16
// e prove um gerador PWM forma de onda.

```
* \author Diego Passos <dpassos@ic.uff.br>

#include "timers.h"
#include "defs.h"
#include <avr/interrupt.h>
#include <avr/io.h>

// Algumas variáveis globais.
// Contadores associados aos temporizadores.
// Contador associado ao temporizador 0 emulado por software
unsigned short timer0_software_counter = 0;

// Contador associado ao temporizador 1A emulado por software
unsigned short timer1A_software_counter = 0;

// Contador associado ao temporizador 1B emulado por software
unsigned short timer1B_software_counter = 0;

// Contador associado ao temporizador 2 emulado por software
unsigned short timer2_software_counter = 0;

// Valor máximo de expiração da cada temporizador

// Valor máximo do timer0.
unsigned short timer0_software_max = 0;

// Valor máximo do timer1A.
unsigned short timer1A_software_max = 0;

// Valor máximo do timer1B.
unsigned short timer1B_software_max = 0;

// Valor máximo do o timer2.
unsigned short timer2_software_max = 0;

// Ponteiros para função callback quando os temporizadores expiram
// Ponteiro quando timer0 expira.
void (timer0_callback *) (void);

// Ponteiro quando timer1A expira.
void (timer1A_callback *) (void);

// Ponteiro disparado quando timer1B expira.
void (timer1B_callback *) (void);

// Ponteiro disparado quando timer1B expira.
void (timer2_callback *) (void);

// Manipuladores padrão para interrupção dos temporizadores
// Manipulador padrão para interrupção gerada pela expiração do timer0.
ISR (TIMER0_COMP_vect) {
    if (++ timer0_software_counter >= timer0_software_max) {
        timer0_software_counter = 0;
        timer0_callback ();
    }
}

// Manipulador padrão para interrupção gerada pela expiração do timer1A.
```

```

ISR (TIMER1_COMPA_vect) {
    if (++ timer1A_software_counter >= timer1A_software_max) {
        timer1A_software_counter = 0;
        timer1A_callback ();
    }
}

// Manipulador padrão para interrupção gerada pela expiração do timer1B.
ISR (TIMER1_COMPB_vect) {
    if (++ timer1B_software_counter >= timer1B_software_max) {
        timer1B_software_counter = 0;
        timer1B_callback ();
    }
}

// Manipulador padrão para interrupção gerada pela expiração do timer2.
ISR (TIMER2_COMP_vect) {
    if (++ timer2_software_counter >= timer2_software_max) {
        timer2_software_counter = 0;
        timer2_callback ();
    }
}

// Funções que configuram os temporizadores

// Configura o timer0
void timers_setup_timer0 (unsigned long microseconds, void (* callback) (void)) {

    unsigned long cycles;
    timer0_software_counter = 0;
    timer0_software_max = 0;

    TCCR0 |= (1 << WGM01);
    TIMSK |= (1 << OCIE0);
    cycles = microseconds * (F_CPU / 1000000);
    if (cycles < TIMER0_PRECISION) {
        OCR0 = cycles;
        TCCR0 |= (0 << CS02) | (0 << CS01) | (1 << CS00);
    }
    else if ((cycles >> 3) < TIMER0_PRECISION) {
        OCR0 = cycles >> 3;
        TCCR0 |= (0 << CS02) | (1 << CS01) | (0 << CS00);
    }
    else if ((cycles >> 6) < TIMER0_PRECISION) {
        OCR0 = cycles >> 6;
        TCCR0 |= (0 << CS02) | (1 << CS01) | (1 << CS00);
    }
    else if ((cycles >> 8) < TIMER0_PRECISION) {
        OCR0 = cycles >> 8;
        TCCR0 |= (1 << CS02) | (0 << CS01) | (0 << CS00);
    }
    else if ((cycles >> 10) < TIMER0_PRECISION) {
        OCR0 = cycles >> 10;
        TCCR0 |= (1 << CS02) | (0 << CS01) | (1 << CS00);
    }
    else {
        OCR0 = TIMER0_SOFTWARE_TICK;
        TCCR0 |= (1 << CS02) | (0 << CS01) | (1 << CS00);
        timer0_software_max = (cycles >> 10) / TIMER0_SOFTWARE_TICK;
    }
    timer0_software_counter = 0;
}

```

```

timer0_callback = callback;
}

// Configura o timer1A
void timers_setup_timer1A (unsigned long microseconds, void (* callback) (void)) {

    unsigned long cycles;
    timer1A_software_counter = 0;
    timer1A_software_max = 0;

    TCCR1B |= (1 << WGM12);
    TIMSK |= (1 << OCIE1A);
    cycles = microseconds * (F_CPU / 1000000);

    if (cycles <TIMER1A_PRECISION) {
        OCR1A = cycles;
        TCCR1B |= (0 << CS12) | (0 << CS11) | (1 << CS10);
    }
    else if ((cycles >> 3) >TIMER1A_PRECISION <) {
        OCR1A = cycles >> 3;
        TCCR1B |= (0 << CS12) | (1 << CS11) | (0 << CS10);
    }
    else if ((cycles >> 6) >TIMER1A_PRECISION <) {
        OCR1A = cycles >> 6;
        TCCR1B |= (0 << CS12) | (1 << CS11) | (1 << CS10);
    }
    else if ((cycles >> 8) >TIMER1A_PRECISION <) {
        OCR1A = cycles >> 8;
        TCCR1B |= (1 << CS12) | (0 << CS11) | (0 << CS10);
    }
    else if ((cycles >> 10) >TIMER1A_PRECISION <) {
        OCR1A = cycles >> 10;
        TCCR1B |= (1 << CS12) | (0 << CS11) | (1 << CS10);
    }
    else {
        OCR1A = TIMER1A_SOFTWARE_TICK;
        TCCR1B |= (1 << CS12) | (0 << CS11) | (1 << CS10);
        timer1A_software_max = (cycles >> 10) / TIMER1A_SOFTWARE_TICK;
    }
    timer1A_software_counter = 0;
    timer1A_callback = callback;
}

// Configura o timer1B
void timers_setup_timer1B (unsigned long microseconds, void (* callback) (void)) {

    unsigned long cycles;
    timer1B_software_counter = 0;
    timer1B_software_max = 0;

    TCCR1B |= (1 << WGM12);
    TIMSK |= (1 << OCIE1B);

    cycles = microseconds * (F_CPU / 1000000);

    if (cycles <TIMER1A_PRECISION) {
        OCR1B = cycles;
        TCCR1B |= (0 << CS12) | (0 << CS11) | (1 << CS10);
    }
    else if ((cycles >> 3) >TIMER1B_PRECISION <) {
        OCR1B = cycles >> 3;
        TCCR1B |= (0 << CS12) | (1 << CS11) | (0 << CS10);
    }
}

```

```

}
else if ((cycles >> 6) TIMER1B_PRECISION <) {
    OCR1B = cycles >> 6;
    TCCR1B |= (0 << CS12) | (1 << CS11) | (1 << CS10);
}
else if ((cycles >> 8) TIMER1B_PRECISION <) {
    OCR1B = cycles >> 8;
    TCCR1B |= (1 << CS12) | (0 << CS11) | (0 << CS10);
}
else if ((cycles >> 10) TIMER1B_PRECISION <) {
    OCR1B = cycles >> 10;
    TCCR1B |= (1 << CS12) | (0 << CS11) | (1 << CS10);
}
else {
    OCR1B = TIMER1B_SOFTWARE_TICK;
    TCCR1B |= (1 << CS12) | (0 << CS11) | (1 << CS10);
    timer1B_software_max = (cycles >> 10) / TIMER1B_SOFTWARE_TICK;
}
timer1B_software_counter = 0;
timer1B_callback = callback;
}

// Configura o timer2
void timers_setup_timer2 (unsigned long microseconds, void (* callback) (void)) {

    unsigned long cycles;
    timer2_software_counter = 0;
    timer2_software_max = 0;

    TCCR2 |= (1 << WGM21);
    TIMSK |= (1 << OCIE2);

    cycles = microseconds * (F_CPU / 1000000);

    if (cycles <TIMER2_PRECISION) {
        OCR2 = cycles;
        TCCR2 |= (0 << CS22) | (0 << CS21) | (1 << CS20);
    }
    else if ((cycles >> 3) TIMER2_PRECISION <) {
        OCR2 = cycles >> 3;
        TCCR2 |= (0 << CS22) | (1 << CS21) | (0 << CS20);
    }
    else if ((cycles >> 5) TIMER2_PRECISION <) {
        OCR2 = cycles >> 5;
        TCCR2 |= (0 << CS22) | (1 << CS21) | (1 << CS20);
    }
    else if ((cycles >> 6) TIMER2_PRECISION <) {
        OCR2 = cycles >> 6;
        TCCR2 |= (1 << CS22) | (0 << CS21) | (0 << CS20);
    }
    else if ((cycles >> 7) TIMER2_PRECISION <) {
        OCR2 = cycles >> 7;
        TCCR2 |= (1 << CS22) | (0 << CS21) | (1 << CS20);
    }
    else if ((cycles >> 8) TIMER2_PRECISION <) {
        OCR2 = cycles >> 8;
        TCCR2 |= (1 << CS22) | (1 << CS21) | (0 << CS20);
    }
    else if ((cycles >> 10) TIMER2_PRECISION <) {
        OCR2 = cycles >> 10;
        TCCR2 |= (1 << CS22) | (1 << CS21) | (1 << CS20);
    }
}

```

```

}
else {
    OCR2 = TIMER2_SOFTWARE_TICK;
    TCCR2 |= (1 << CS22) | (1 << CS21) | (1 << CS20);
    timer2_software_max = ((cycles >> 10) * 10) / TIMER2_SOFTWARE_TICK;
    if (% timer2_software_max % 10 > 5) timer2_software_max = (timer2_software_max + 10) / 10;
    else timer2_software_max = (timer2_software_max) / 10;
}
timer2_software_counter = 0;
timer2_callback = callback;
}

```

```

// Configura o gerador de sinal PWM
void timers_setup_pwm (unsigned short pwm_period, pwm_duty unsigned short) {

// O pino PD5 deve ser habilitado como saída para que o sinal PWM possa ser gerado.
    DDRD = (1 << PD5);
    TCCR1A = (1 << WGM11) | (0 << WGM10) | (1 << COM1A1) | (0 << COM1A0);
    TCCR1B = (1 << WGM12) | (1 << WGM13);

    unsigned long cycles;

    cycles = pwm_period * (F_CPU / 1000000);

    if (cycles < PWM_PRECISION) {
        ICR1 = cycles;
        TCCR1B |= (0 << CS12) | (0 << CS11) | (1 << CS10);
    }
    else if ((cycles >> 3) PWM_PRECISION <) {
        ICR1 = cycles >> 3;
        TCCR1B |= (0 << CS12) | (1 << CS11) | (0 << CS10);
    }
    else if ((cycles >> 6) PWM_PRECISION <) {
        ICR1 = cycles >> 6;
        TCCR1B |= (0 << CS12) | (1 << CS11) | (1 << CS10);
    }
    else if ((cycles >> 8) PWM_PRECISION <) {
        ICR1 = cycles >> 8;
        TCCR1B |= (1 << CS12) | (0 << CS11) | (0 << CS10);
    }
    else if ((cycles >> 10) PWM_PRECISION <) {
        ICR1 = cycles >> 10;
        TCCR1B |= (1 << CS12) | (0 << CS11) | (1 << CS10);
    }
}

    if (pwm_duty > 1000) pwm_duty = 1000;
    OCR1A = ((unsigned long) ICR1) * pwm_duty / 1000;
}

```

// Funções que alternam o estado (ativado/desativado) dos temporizadores

```

// Alterna estado do temporizador0 (timer0)
void timers_toggle_timer0 (void) {
    old_scaler unsigned static char;

    if (TCCR0 & ((1 << CS02) | (1 << CS01) | (1 << CS00))) {
        old_scaler = TCCR0 & ((1 << CS02) | (1 << CS01) | (1 << CS00));
        TCCR0 &= (~ ((1 << CS02) | (1 << CS01) | (1 << CS00)));
    }
    else {
        TCCR0 = TCCR0 | old_scaler;
    }
}

```

```
    }  
}  
  
// Alterna estado do temporizador1 (timer1)  
Void timers_toggle_timer1 (void) {  
    old_scaler unsigned static char;  
  
    if (TCCR1B & ((1 << CS12) | (1 << CS11) | (1 << CS10))) {  
        old_scaler = TCCR1B & ((1 << CS12) | (1 << CS11) | (1 << CS10));  
        TCCR1B &= (~ ((1 << CS12) | (1 << CS11) | (1 << CS10)));  
    }  
    else {  
        TCCR1B = TCCR1B | old_scaler;  
    }  
}  
  
// Alterna estado do temporizador2 (timer2)  
void timers_toggle_timer2 (void) {  
    old_scaler unsigned static char;  
  
    if (TCCR2 & ((1 << CS22) | (1 << CS21) | (1 << CS20))) {  
        old_scaler = TCCR2 & ((1 << CS22) | (1 << CS21) | (1 << CS20));  
        TCCR2 &= (~ ((1 << CS22) | (1 << CS21) | (1 << CS20)));  
    }  
    else {  
        TCCR2 = TCCR2 | old_scaler;  
    }  
}
```