

UNIVERSIDADE FEDERAL DE VIÇOSA  
CENTRO DE CIÊNCIAS EXATAS E TECNOLÓGICAS  
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

WALTER MAGNO VALENTE

**DATALOGGER PARA AQUISIÇÃO DE DADOS DE SISTEMAS  
ELÉTRICOS**

VIÇOSA  
2016

WALTER MAGNO VALENTE

**DATALOGGER PARA AQUISIÇÃO DE DADOS DE SISTEMAS  
ELÉTRICOS**

Monografia apresentada ao Departamento de Engenharia Elétrica do Centro de Ciências Exatas e Tecnológicas da Universidade Federal de Viçosa, para a obtenção dos créditos da disciplina ELT 490 – Monografia e Seminário e cumprimento do requisito parcial para obtenção do grau de Bacharel em Engenharia Elétrica.

Orientador: Prof. DS. Denilson Eduardo Rodrigues.

VIÇOSA  
2016



**WALTER MAGNO VALENTE**

**DATALOGGER PARA AQUISIÇÃO DE DADOS DE SISTEMAS  
ELÉTRICOS**

Monografia apresentada ao Departamento de Engenharia Elétrica do Centro de Ciências Exatas e Tecnológicas da Universidade Federal de Viçosa, para a obtenção dos créditos da disciplina ELT 490 – Monografia e Seminário e cumprimento do requisito parcial para obtenção do grau de Bacharel em Engenharia Elétrica.

Aprovada em 03 de Junho de 2016.

**COMISSÃO EXAMINADORA**

---

**Prof. DS. Denilson Eduardo Rodrigues - Orientador**  
**Universidade Federal de Viçosa**

---

**Prof. DS. Heverton Augusto Pereira - Membro**  
**Universidade Federal de Viçosa**

---

**Eng. Igor Rodrigues de Oliveira - Membro**  
**Universidade Federal de Viçosa**

*“Daria tudo que sei pela metade do que ignoro”.*

*(René Descartes)*

*Dedico este trabalho a todos que me ajudaram a chegar onde estou.*

## *Agradecimentos*

Agradeço primeiramente a minha mãe, por ter me apoiado sempre, e ter me suportado tanto tempo estudando engenharia.

Agradeço também Eduardo Raymundo por ter me ensinado o que é engenharia na prática, e isto foi um grande diferencial na minha formação como engenheiro eletricitista.

Agradeço a todos os meus amigos que desde o início desta caminhada, me apoiaram e me zuaram, me fez rir e chorar. Clair (patrícia), Cabeça (mestre), Zé (pequeno), Salsa, Josy (dinossaura), Aroaldo (baiano), Thaís (que de tão chata nem apelido tem) e Nissim (vinicera).

Agradeço também a todos os moradores do 1712 (só mora jamanta naquela casa), que me acolheram em sua humilde residência. Markiimm (não vai formar por causa de Física 3), Melãozim, Taruíra (que se formou para fazer trabalhos filantrópicos).

A todos vocês muito obrigado por me ajudar, e a fazer parte desta história.

## ***Resumo***

Neste trabalho foi desenvolvido um protótipo para monitorar tensão e corrente de um sistema de geração de energia. A importância deste projeto é dar suporte a pesquisadores e técnico para projetar ou propor melhorias em sistemas já existentes. Desta forma o equipamento concebido lê e armazena as informações em um cartão SD juntamente com a data e hora do momento da leitura. Posteriormente com os dados coletados e um computador, podem-se realizar estudos variados. Este equipamento, também conhecido como *datalogger* de energia, utiliza um microcontrolador e módulos periféricos facilmente encontrados no mercado a preços acessíveis. Este *datalogger* tem erro menor que 5% nas medidas, podendo ser usados para levantamentos preliminares. Possui autonomia de energia de 20 dias e autonomia de memória de dados de 144 anos.

## *Abstract*

In this work was developed a voltage and current monitor, for a power generation system. This project importance is to support researchers and technicians to design and propose improvements in existing systems. The equipment designed reads and stores the information in an SD card with the date and time of the readings. Later with the collected data and a computer various studies can be performed. This equipment, also known as a power datalogger, uses a microcontroller and peripheral modules easily found in the market at affordable prices. This datalogger has error smaller than 5% in the measures and may be used for preliminary surveys. Has 20 days energy independence and autonomy of 144 years data memory.

# Sumário

1	Introdução.....	14
1.1	Objetivo .....	15
2	Revisão Bibliográfica.....	16
2.1	Instrumentação.....	16
2.2	<i>Datalogger</i> .....	16
2.3	Microcontrolador .....	16
2.3.1	Conversor ADC .....	19
2.3.2	Comunicação I2C .....	19
2.3.3	Comunicação SPI .....	21
2.3.4	Contador WDT e Modo SLEEP .....	22
2.4	Relógio RTC DS1307.....	23
2.5	Sensor de Corrente Elétrica ASC712 .....	24
2.6	Cartão <i>Secure Digital</i> .....	25
2.7	Display LCD.....	27
2.8	Bateria Chumbo-Ácido.....	27
3	Materiais e Métodos .....	29
3.1	Montagem.....	29
3.2	Algoritmo .....	33
4	Resultados e Discussões.....	36
4.1	Arquivo txt.....	36
4.2	Precisão das Medidas .....	37
4.3	Autonomia .....	39
5	Conclusões.....	42
6	Trabalhos Futuros.....	44
	Referências Bibliográficas .....	45

## *Lista de Figuras*

Figura 1 - Arquitetura <i>Harvad</i> . Barramento de dados e barramento de instrução. ....	18
Figura 2 - Microcontrolador 18F2550 do MICROCHIP .....	19
Figura 3 - Pinos do microcontrolador.....	19
Figura 4 - Esquema elétrico interno do conversor ADC por amostragem. ....	20
Figura 5 - Montagem de vários equipamentos em rede I2C.....	21
Figura 6 - Comunicação I2C. ....	21
Figura 7 - Montagem de vários equipamentos em barramento SPI. ....	23
Figura 8 - Esquema elétrico e ligação ao microcontrolador.....	24
Figura 9 - Registradores do RTC.....	25
Figura 10 - Módulo sensor de corrente ASC712.....	25
Figura 11 - Esquema elétrico do módulo sensor de corrente. ....	26
Figura 12 - Cartão SD.....	26
Figura 13 - Esquema elétrico do módulo SD. ....	27
Figura 14 - Módulo <i>display</i> LCD. ....	28
Figura 15 - Bateria chumbo ácido 12 V 7 Ah.....	28
Figura 16 - Módulo RTC.....	31
Figura 17 - Regulador de tensão 78l05 de baixo consumo.....	31
Figura 18 - Esquema elétrico da fonte do <i>datalogger</i> . ....	31
Figura 19 - Diagrama das conexões feitas entre o <i>nobreak</i> e a fonte do <i>datalogger</i> .....	32
Figura 20 - Esquema elétrico parcial do <i>datalogger</i> . Microcontrolador 18F2550 e <i>display</i> LCD.....	33
Figura 21 - Periféricos necessários ao funcionamento do <i>datalogger</i> .....	34
Figura 22 - Fluxograma da rotina do modo de configuração do sistema e modo leitura. ....	35
Figura 23 - Fluxograma que exemplifica a rotina que reduz o nível de ruído dos sinais lidos.	37
Figura 24 - Arquivo gerado pelo <i>datalogger</i> aberto no bloco de notas.....	38

Figura 25 - Gráfico gerado no MATLAB a partir dos dados lidos do arquivo gerado pelo *datalogger* ..... 39

Figura 26 - Circuito divisor de tensão para testes de precisão ..... 40

## *Lista de Tabelas*

Tabela 1 – Teste do medidor de tensão. ....	40
Tabela 2 - Teste do sensor de corrente. ....	41
Tabela 3 – Consumo de corrente dos periféricos. ....	42

# ***1 Introdução***

Com o aumento da consciência ambiental, surgiu a expressão sustentabilidade, que significa consumir sem degradar o ambiente em que vivemos [1]. A energia elétrica ainda hoje produzida degrada consideravelmente o meio ambiente. Um bom exemplo disso são as termelétricas a combustíveis fósseis que emitem gás carbônico ( $\text{CO}_2$ ) no ar, contribuindo para o efeito estufa. Até mesmo as hidrelétricas que eram consideradas produtoras de energia limpa, poluem o ar com gás metano ( $\text{CH}_4$ ) proveniente da decomposição das matérias orgânicas que morrem ao se alagar uma área para a construção da barragem [2].

Neste contexto tem-se aumentado também a implantação de usinas de energia que utilizam fontes alternativas menos poluentes, como por exemplo, usinas eólicas e fotovoltaicas. Em um país grande como o Brasil, têm-se diversas condições ambientais favoráveis à implantação destas usinas, da mesma forma que existem regiões que não são propícias à implantação de certas usinas. Numa mesma região podem existir várias outras condições que afetam a viabilidade técnico-econômica dessas usinas, pois as condições ambientais não se repetem [3].

A cada dia que passa, a energia elétrica fica mais cara, seja pelo aumento de tributos ou pela escassez de água, isto acaba viabilizando a implantação de usinas em regiões onde antes eram considerados inviáveis [3, 4]. Daí surge a necessidade de novos estudos de viabilidade técnica e econômica.

A resolução nº 482/2012 da Agência Nacional de Energia Elétrica (ANEEL) estabelece as regras para conexão de mini e microgeração distribuída no sistema elétrico e o sistema de compensação de energia elétrica. A resolução nº 687/2015 modifica alguns pontos da resolução nº 482/2012 e os Módulos 1 e 3 dos Procedimentos de Distribuição – PRODIST [5, 6]. Tais resoluções são boas oportunidades para a implantação de células fotovoltaicas e aerogeradores em residências, com potencial para abastecer a residência e outros empreendimentos mesmo que distante geograficamente.

O problema da implantação destas fontes alternativas, é que em cada região existe uma particularidade que impede que as fontes de energia trabalhem com a potência máxima. No caso de aerogeradores esses problemas podem ser ventos fracos, turbulência em decorrência de obstáculos próximos, etc. No caso de painéis fotovoltaicos, os problemas podem ser

sombra em uma dada hora do dia ou período do ano, baixa incidência de raios solares, etc. Estes problemas impedem a utilização de dados recolhidos de outras regiões, mesmo que próximas, para fazer estudos de potencial energético [3, 7].

Daí surge à necessidade de estudo de potenciais energéticos pontuais. Com o aumento dos incentivos da utilização de fontes alternativas, cresce também a demanda por profissionais e equipamentos capazes de atender as necessidades do mercado [8].

## **1.1 Objetivo**

Este trabalho foi desenvolvido para alcançar vários objetivos, dentre eles objetivos gerais e outros específicos. Os objetivos específicos são:

- Elaborar um protótipo para quantificar a tensão e corrente que um sistema elétrico está fornecendo ou consumindo em um dado instante e registra-los;
- Elaborar um projeto confiável, com erros de medidas menores que 5%.
- Desenvolver o protótipo com baixo custo.
- Desenvolver o protótipo com autonomia superior a duração máxima de interrupção contínua (DMIC), que é de 2,69 horas nesta região.

Os objetivos gerais são:

- Aperfeiçoar mais o conhecimento sobre eletrônica e programação de microcontroladores.

## **2 Revisão Bibliográfica**

### **2.1 Instrumentação**

Pode-se definir instrumentação como a ciência que aplica e desenvolve técnicas para adequação de instrumentos de medição, transmissão, indicação, registro e controle de variáveis físicas em equipamentos nos processos industriais. Ou seja, conjunto de técnicas para o desenvolvimento e construção de equipamentos para medição de grandezas físicas [9].

Neste trabalho, a instrumentação é à base do processo de monitoramento, pois o protótipo construído irá medir e registrar tensão e corrente elétrica [10]. Para isso utilizam-se sensores e técnicas específicas para cada fim.

### **2.2 Datalogger**

*Dataloggers* são equipamentos capazes de medir uma grandeza física (ou várias), e gravá-las em uma memória com a data e a hora em que as medidas foram feitas [9].

Os *dataloggers* geralmente funcionam a bateria e podem ser utilizados para os mais diversos fins, como por exemplo, colher dados de temperatura, umidade relativa do ar, velocidade e direção do vento, dentre outros. Por exemplo, um *datalogger* que colhe dados sobre as condições do ambiente, pode ser usado para os cálculos de previsão do tempo.

### **2.3 Microcontrolador**

Hoje em dia é fácil encontrar circuitos eletrônicos que possuem certa “inteligência”, brinquedos, eletrodomésticos, etc. Esta “inteligência” se deve a circuitos capazes de executar funções lógicas. Os microcontroladores são circuitos integrados capazes de executar alguma tarefa lógica programada previamente [11]. Existem vários tipos de microcontroladores com as mais diversas especialidades e funcionalidade, variando apenas as quantidades de entradas/saídas (IOs), periféricos específicos como conversores ADC, interrupções,

quantidade de temporizadores, velocidade de *clock*, etc. Dentre os microcontroladores mais utilizados, encontra-se a família PIC (*Peripheral Interface Controller*) do fabricante MICROCHIP, que é muito utilizada devido seu baixo custo e versatilidade.

A família PIC possui microcontroladores de 12, 14 e 32 *bits*. A arquitetura *Harvard* separa o barramento de dados do barramento de instrução, como pode ser observado destacado na Figura 1. Este tipo de arquitetura garante ótimo desempenho em tarefas que dependem apenas de lógicas. Por possuir o barramento de instrução separado, as instruções ocupam apenas uma palavra de 16 bits, podendo assim, executar a grande maioria das instruções em um único ciclo de máquina. Esta família de microcontroladores é chamada também de RISC (*Reduced Instruction Set Computer*), que significa que eles possuem um *set* de instrução reduzido, mais precisamente 35 instruções, o que facilita a programação de rotinas [12].

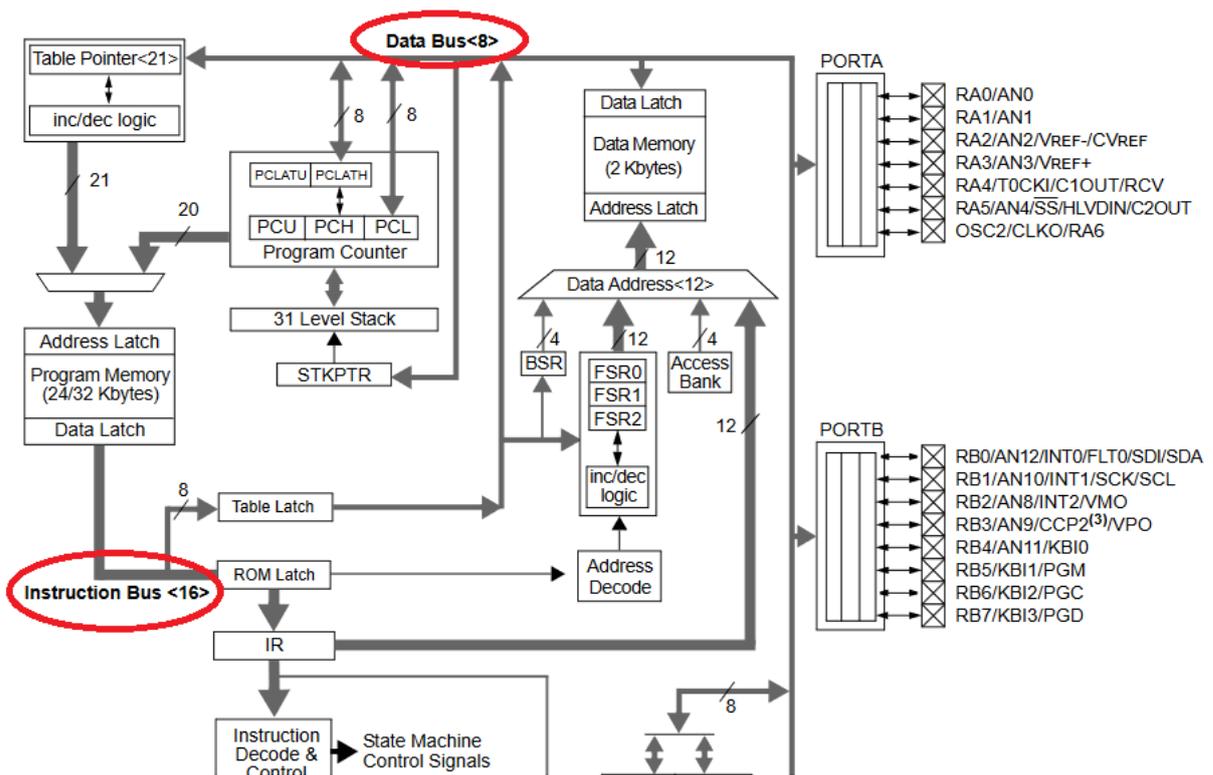


Figura 1 - Arquitetura *Harvard*. Barramento de dados e barramento de instrução [13].

O microcontrolador escolhido para integrar este projeto foi o PIC18F2550, que pode ser visto na Figura 2. Os microcontroladores da família PIC têm custo acessível e são facilmente encontrados no mercado.



Figura 2 - Microcontrolador 18F2550 do MICROCHIP [14].

Na Figura 3 pode ser visto as funções de cada pino do microcontrolador PIC18F2550.

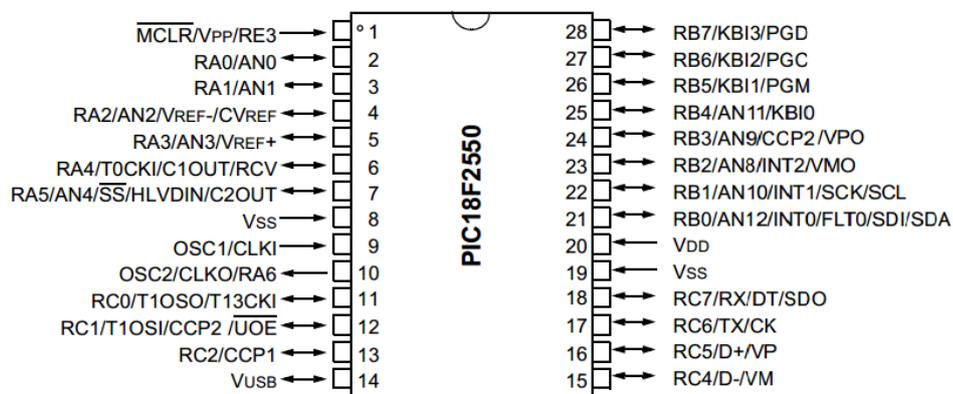


Figura 3 - Pinos do microcontrolador [13].

As principais características do microcontrolador 18F2550 são [13]:

- Memória de programa (FLASH) 32 Kb;
- Memória de dados (SRAM) 2048 bytes;
- Memória de dados (EEPROM) 256 bytes;
- 24 portas I/O;
- Conversor ADC de 10 bits;
- 19 fontes de interrupção;
- 4 temporizadores (TIMERS);
- Comunicação SPI, I2C, USB e USART;
- Oscilador externo: cristal ou RC, e oscilador interno RC;

- Temporizador de segurança WDT;
- Controladores “Power-on”.
- Gravação in-circuit.

A opção de gravação *in-circuit* é muito útil na construção de protótipos, pois facilita a modificação e testes do *firmware*.

### 2.3.1 Conversor ADC

O conversor analógico/digital (A/D) efetua a conversão de um sinal analógico para a sua representação digital de 10 *bits*. O PIC18F2550 possui treze entradas analógicas (AN0, AN1, AN2, AN3, AN4, AN5, AN6, AN7, AN8, AN9, AN10, AN11 e AN12) que são multiplexadas para um circuito de amostragem e retenção conforme ilustrado na Figura 4. A saída do circuito de amostragem e retenção é conectada à entrada do conversor ADC de 10 *bits*.

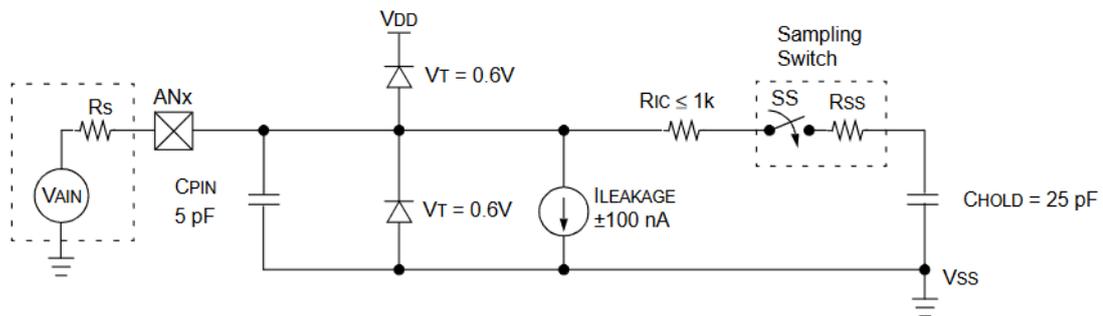


Figura 4 - Esquema elétrico interno do conversor ADC por amostragem [13].

O conversor ADC gera um resultado binário através de um processo de aproximação sucessiva e armazena o resultado em um registrador de 10 *bits*. A tensão de referência utilizada pelo conversor pode ser selecionada por *software*, entre a tensão de alimentação ou a tensão aplicada ao pino “V<sub>ref</sub>” [15].

### 2.3.2 Comunicação I2C

A comunicação síncrona *Inter-Integrated Circuit* (I2C) é um barramento serial desenvolvido pela Philips que é usado para conectar periféricos de baixa velocidade a um

circuito integrado principal (mestre). A comunicação I2C usa apenas duas linhas bidirecionais, a linha serial de dados (SDA) e linha serial de *clock* (SCL) com resistores *pull-up*. Existe somente um barramento de dados, logo não é possível enviar e receber dados ao mesmo tempo, este tipo de comunicação é chamado de *Half-duplex* [16].

Com o barramento I2C é possível conectar vários equipamentos, um deles é denominado mestre e os outros escravos, como mostra a Figura 5. O dispositivo mestre é sempre o que controla a linha de *clock* (SCL), os escravos são os dispositivos que respondem ao mestre. O dispositivo escravo nunca inicia uma transmissão se que o mestre tenha requisitado.

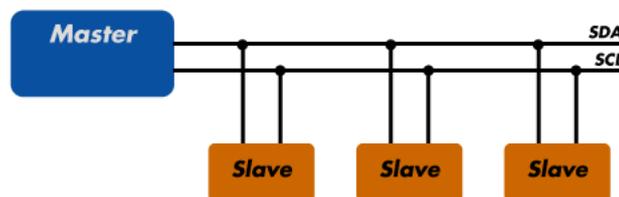


Figura 5 - Montagem de vários equipamentos em rede I2C.

Cada dispositivo escravo tem um endereço que o identifica dentre uma rede de dispositivos escravos, estes endereços não podem se repetir. Os endereços têm 7 bits, o que limita o número de dispositivos conectados na rede I2C em 127 dispositivos. A transmissão é dividida em *strings* de 8 bits cada, o último bit da *string* endereço é o que informa ao escravo se será uma operação de leitura ou de escrita, se o último bit for '0' o dispositivo mestre quer ler o dispositivo, se o bit for '1' o mestre quer escrever no dispositivo. A Figura 6 exemplifica a transmissão de dados entre os dispositivos [17].

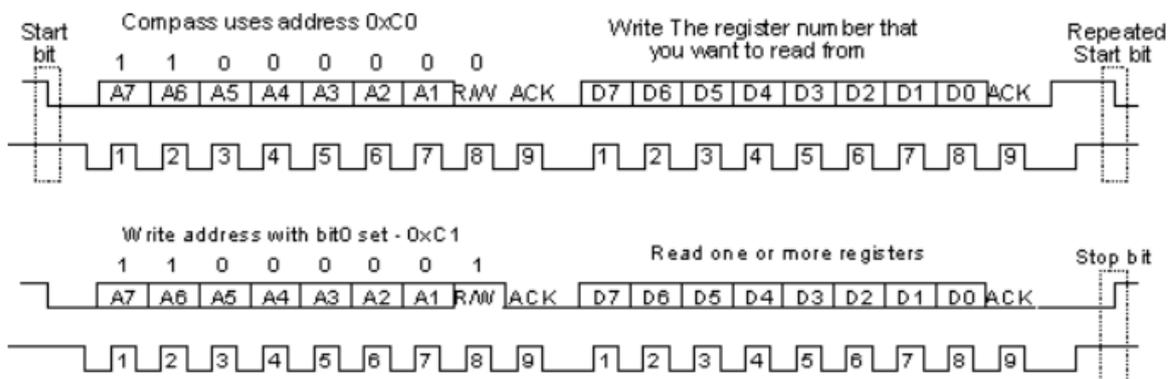


Figura 6 - Comunicação I2C [17].

Como mostra a figura anterior, primeiro é feito um sinal de *start* para iniciar a transmissão, depois é transmitido o endereço do dispositivo que se quer obter a informação, seguido do *bit* que diz se a operação será leitura ou escrita, depois um sinal de reconhecimento (ACK). Caso a transmissão seja uma operação de leitura, a próxima *string* é o endereço do registrador que o dispositivo mestre quer ler, se a operação for escrita, a próxima *string* é o endereço do registrador onde será gravado o dado que o mestre irá transmitir. Tanto na operação de leitura ou escrita, sempre é enviado o sinal de reconhecimento ACK depois de cada *string* de 8 *bits*. Posteriormente se a operação for leitura, o dispositivo escravo transmite o dado daquele registrador, se a operação for escrita, o dispositivo escravo grava o dado transmitido naquele registrador. Quando esta operação chega ao fim, é transmitido um sinal de *stop* para encerrar a transmissão [17].

### 2.3.3 Comunicação SPI

A *Serial Peripheral Interface* (SPI) é um protocolo de dados seriais síncronos, utilizado em circuitos eletrônicos para comunicação entre circuito integrado mestre e circuitos integrados escravos. O protocolo de comunicação SPI tem certas semelhança e também diferenças do protocolo de comunicação I2C. O protocolo SPI trabalha a 4 fio, a saber [16]:

- MISO (*Master IN Slave OUT*) - Dados do escravo para o mestre;
- MOSI (*Master OUT Slave IN*) - Dados do mestre para o escravo;
- SCK (*Serial Clock*) - *Clock* de sincronização para transmissão de dados entre o mestre e o escravo;
- SS (*Slave Select*) - Seleciona qual escravo receberá os dados.

O protocolo SPI aceita somente um mestre, e não existem endereços para os escravos, a seleção do dispositivo no qual o mestre vai iniciar uma transmissão é feita através do pino de seleção, como mostra a Figura 7. Nota-se então que o mestre deve ter um pino de seleção exclusivo para cada dispositivo escravo, o que faz com que o mestre necessite de vários pinos para controlar uma rede de comunicação SPI [18].

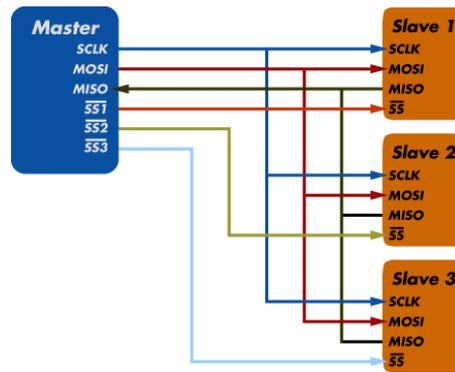


Figura 7 - Montagem de vários equipamentos em barramento SPI.

A comunicação SPI tem um pino de *clock* controlado pelo mestre, e dois pinos de transmissão de dados, um usado pelo mestre para transmitir ao escravo e outro usado pelo escravo para transmitir ao mestre. Nesta comunicação pode acontecer dos dispositivos enviar e receber mensagem ao mesmo tempo, este tipo de comunicação se chama *full-duplex*. A comunicação SPI pode transmitir as mensagens em uma taxa mais alta que a comunicação I2C, esta taxa pode chegar a 3 MB/s [18].

### 2.3.4 Contador WDT e Modo SLEEP

O temporizador de segurança *Watch Dog Timer* (WDT) é um recurso interessante para evitar que o microcontrolador trave por alguma pane qualquer no *software* ou *hardware*. Ele funciona como um contador independente que *reseta* a CPU do microcontrolador sempre que a contagem chega ao final (*overflow*). Para que o *reset* do WDT não prejudique o desenvolvimento normal do programa que está sendo executado, o programador tem que acrescentar em determinadas partes do programa uma instrução para *resetar* a contagem do WDT para que esta não chegue até o final da contagem [12]. Este contador pode contar de 18 ms a 2,304 segundos, dependendo da configuração.

O modo SLEEP é um estado do microcontrolador muito utilizado para economizar energia quando este se encontra ocioso. O microcontrolador quando entra neste estado, desliga o oscilador e com ele todo o processamento. As portas continuam da mesma forma, por isso é necessário antes de coloca-lo em modo SLEEP verificar todas as portas a fim de não consumir energia desnecessariamente [19].

Para fazer o microcontrolador voltar ao estado normal pode-se proceder de três formas, dentre elas:

- Colocar em estado baixo ( 0 ) o pino MCLR;
- Interrupção externa no pino RB0/INT ou mudança de estado no PORT B;
- Estouro (*overflow*) do WDT.

## 2.4 Relógio RTC DS1307

O relógio *Real Time Clock* (RTC) DS1307 é um dispositivo capaz de marcar a hora e a data com uma boa precisão. A programação e leitura da data e da hora são feitas através do protocolo de comunicação I2C. Através de algoritmos específicos pode-se fazer a leitura de segundos, minutos, horas, dia da semana, dia do mês e ano. Este relógio RTC pode operar em dois modos: relógio de 12 horas ou de 24 horas. Este relógio possui dois pinos específicos para ligar uma bateria externa de 3 V, a fim de alimentar o oscilador interno e manter os registros de data e hora mesmo quando não há energia.

Na Figura 8 tem-se o circuito do módulo DS1307 [20].

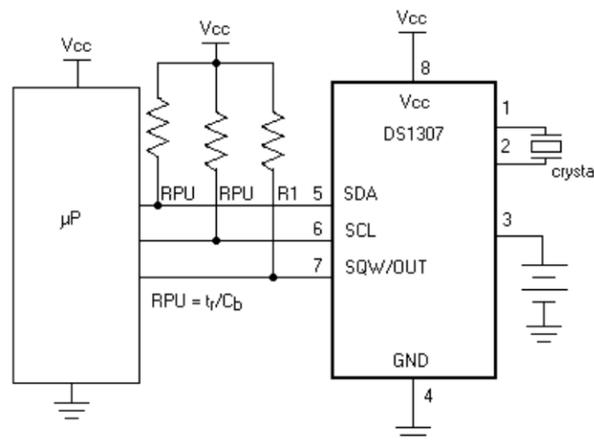


Figura 8 - Esquema elétrico e ligação ao microcontrolador [20].

As informações de data e hora são obtidas a partir das leituras de alguns registradores específicos do relógio. Na Figura 9 pode-se observar o endereço de cada item que o relógio/calendário manipula, e as faixas de valores.



O sensor de corrente ASC712-20A mede corrente de -20 A até +20 A, e retorna um valor de tensão de 0 a 5 V proporcional a corrente lida, então a resolução deste sensor é de 100 mV/A [22]. Na folha de dados do circuito integrado ASC712 fornecido pelo fabricante, têm-se vários circuitos de exemplos e aplicações. Na Figura 11, pode-se ver o circuito elétrico do módulo utilizado.

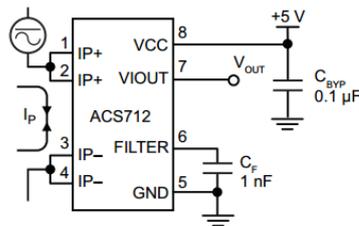


Figura 11 - Esquema elétrico do módulo sensor de corrente [22].

## 2.6 Cartão Secure Digital

O cartão de memória *Secure Digital* (SD) é um pequeno cartão de memória *flash*, para armazenamento de dados. Eles são de pequenas dimensões, econômicos no consumo de energia, têm alta capacidade de armazenamento e são capazes de armazenar dados por um longo período de tempo [23]. Pode-se observar seu aspecto na Figura 6.



Figura 12 - Cartão SD.

Os cartões SD funcionam em tensão de 3,3 V, por isso se faz necessário uma interface de acoplamento entre o microcontrolador e o cartão. O módulo do cartão SD, já vem com o circuito de interface que é simplesmente um divisor de tensão resistivo. O circuito do módulo pode ser visto na Figura 13.

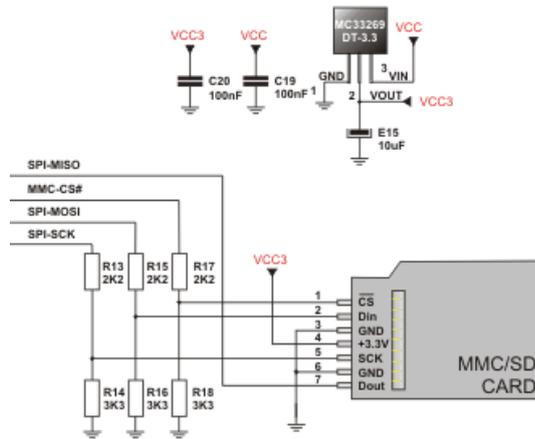


Figura 13 - Esquema elétrico do módulo SD [24].

A comunicação entre o módulo e o microcontrolador é feita através do protocolo de comunicação SPI. Os pinos responsáveis pela comunicação entre o microcontrolador e o cartão SD são [25]:

- SCK (pino 5): Barramento de *clock*, necessário para sincronizar a comunicação entre microcontrolador e cartão SD;
- CS (pino 1): Usado para selecionar o equipamento SPI, caso haja vários conectados no barramento de comunicação;
- D<sub>in</sub> (pino 2): Linha usada para entrada de dados no cartão SD;
- D<sub>out</sub> (pino 7): Linha usada para enviar dados ao microcontrolador. Nota-se que este pino é conectado diretamente no pino do microcontrolador, isto só é possível se o pino utilizado não é *Schmitt-trigger*.

Para o correto funcionamento da comunicação, o cartão SD tem que estar formatado no sistema de arquivos FAT16 (*File Allocation Table*), isso porque a biblioteca utilizada para a manipulação dos dados no cartão SD, só tem suporte para este tipo de formatação. O FAT16 tem um inconveniente, mesmo um cartão com capacidade superior a 2 GB de memória, só é reconhecido como tendo 2 GB, o restante não é utilizado. Isto porque a formatação FAT16 possuem endereços de apenas 16 *bits*, totalizando 65536 endereços de memória, tais endereços são chamados de *clusters*, que são os locais da memória onde são gravadas as informações. Esses *clusters* podem possuir de 512 *bytes* a 32 kB dependendo da capacidade total do cartão SD [24].

## 2.7 Display LCD

Para facilitar a interação entre homem e máquina é necessário algum tipo de interface que faça essa comunicação. Foi utilizado neste projeto um *display* LCD (*liquid crystal display*) com 16 colunas e 2 linhas. O *display* LCD pode ser visto na Figura 14.



Figura 14 - Módulo *display* LCD.

O baixo custo e a simplicidade de uso são grandes vantagens deste periférico. Como são muito utilizados, praticamente todos os compiladores têm bibliotecas específicas para este tipo de *display*, o que poupa muito tempo na programação.

## 2.8 Bateria Chumbo-Ácido

A bateria de chumbo ácido, mostrada na Figura 15, é a mais conhecida e difundida no mercado. É utilizada para os mais diversos fins, como, por exemplo, em veículos de combustão interna para alimentar o motor de arranque e os sistemas eletrônicos, em sistemas onde não pode haver interrupção de energia, também conhecida como *nobreak*, em iluminação de emergência, dentre outros [26].

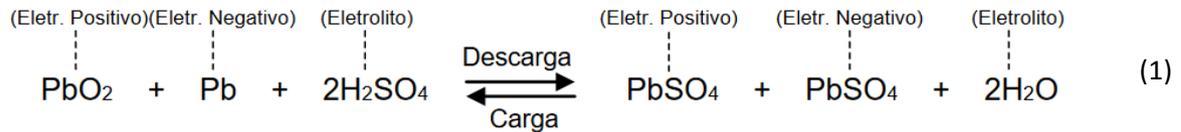


Figura 15 - Bateria chumbo ácido 12 V 7 Ah.

Estas baterias são compostas por células ligadas em série com tensão nominal de 2,1 V por célula. Em baterias de 12 V são necessárias 6 células. Também chamada de acumuladores, estas baterias usam ácido sulfúrico ( $H_2SO_4$ ) dissolvido em água a uma proporção de 1,280 g/ml para carga total e 1,1 g/ml para carga mínima ou sem carga. A placa

positiva é composta por chumbo e é recoberta ou impregnada de dióxido de chumbo (PbO<sub>2</sub>), e a placa negativa é formada por chumbo esponjoso [27].

A reação química responsável pela geração de energia na bateria chumbo-ácido pode ser vista na Equação 1:



Nota-se pela equação química (1) que quando se utiliza a bateria, há formação de moléculas de água, muitas vezes na forma gasosa. A água evaporando diminui o nível de eletrólito dentro da bateria, tornando necessário completar o nível da bateria de tempos em tempos.

Deste problema surgiram as baterias do tipo Valve Regulated Lead-Acid (VRLA), que são unidades seladas de chumbo-ácido com melhores capacidades e desempenho. Tais baterias usam gel como eletrólito e possuem uma válvula que regula automaticamente a emissão de gases de hidrogênio e a pressão interna da bateria [28].

## 3 *Materiais e Métodos*

Neste tópico são abordados os componentes do sistema e seus respectivos funcionamentos.

### 3.1 *Montagem*

O projeto foi feito em partes modulares para facilitar o intercâmbio de peças e possíveis alterações que viessem a ocorrer por incompatibilidade, dificuldades de aquisição ou de adequação dos projetos.

Para fazer a escolha do microcontrolador a ser utilizado, foram levados em consideração os periféricos que serão utilizados neste projeto. O periférico determinante para tal escolha foi o módulo cartão SD, que “pede” uma memória de *buffer* para escrita e leitura de no mínimo 1MB, que somente é encontrada a partir da família 18F. O microcontrolador deverá ter também os recursos SLEEP e WDT, com o intuito de economizar energia.

A opção de gravação *in-circuit* é muito útil em protótipos, pois facilita a modificação e testes do *firmware*. Serão utilizados os periféricos de comunicação síncrona SPI para a comunicação com o módulo de cartão SD, também será utilizada a comunicação I2C para a comunicação com o relógio/calendário DS1307. Serão utilizados também dois canais de conversores A/D para as leituras de corrente e de tensão. Duas entradas lógicas, onde serão alocados os botões para configuração, e pinos para conexão do *display* LCD. Então o modelo escolhido que contem os recursos necessários é o PIC 18F2550.

O módulo relógio RTC foi montado como sugerido no *datasheet* do fabricante e o cristal utilizado foi de 32768 Hz. Na Figura 16 encontra-se o módulo de relógio/calendário já montado com bateria CR2032 e cristal oscilador.



Figura 16 - Módulo RTC.

Neste projeto, todos os componentes foram escolhidos levando em consideração o consumo de energia. Como se trata de um equipamento que irá funcionar aproximadamente 1 ano sem a intervenção humana, foi necessário calcular e escolher cuidadosamente os componentes do sistema, para não haver gastos desnecessários de energia no equipamento.

A fonte de alimentação foi concebida utilizando um LM78L05 para alimentar o microcontrolador e os demais periféricos. O regulador LM78L05 tem a característica de baixo consumo de potência, diferentemente dos reguladores de tensão da família LM78XX, que independente da carga, consomem aproximadamente 5 mA. A família LM78LXX tem seu consumo variando com a impedância da carga [29]. Neste projeto isto é muito útil, pois nos intervalos de tempo que o microcontrolador não irá executar tarefa alguma, ele estará em modo SLEEP para economizar energia, assim o regulador de tensão também irá reduzir o consumo automaticamente. O aspecto do regulador pode ser visto na Figura 17.



Figura 17 - Regulador de tensão 78L05 de baixo consumo.

O projeto elétrico da fonte de alimentação do *datalogger* pode ser vista na Figura 18.

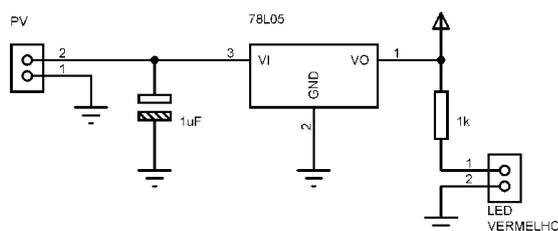


Figura 18 - Esquema elétrico da fonte do *datalogger*.

Um *datalogger* para o monitoramento de energia também precisa estar em funcionamento durante a falta de energia, pois em muitos casos é preciso saber quanto tempo o sistema elétrico ficou sem energia. Pensando nisso foi projetado um simples *nobreak* para alimentar o *datalogger* em casos de falta de energia.

Para o projeto elétrico do *nobreak*, foram levados em consideração a capacidade da bateria e a profundidade de descarga, para o caso de se ter uma falta de energia muito longa. O preço e disponibilidade no mercado também foram levados em consideração, e a escolha feita foi à bateria de chumbo-ácido de 12 V e 7 Ah. Este tipo de bateria é facilmente encontrada no mercado por ser muito utilizada em *nobreak* de computadores pessoais. Este tipo de bateria pode ser recarregada utilizando diversos algoritmos, como por exemplo, carga rápida, carga lenta, tensão constante, corrente constante, etc. Os carregadores de bateria mais eficientes levam em consideração a temperatura das células da bateria e muda as estratégias de carregamento, dependendo do estágio da carga [30]. Como não é objetivo deste trabalho o tipo de recarga da bateria, foi usada a técnica de tensão constante. Esta técnica é a mais simples e mais utilizadas em sistemas que tem problemas apenas com pequenas interrupções de energia [26]. O *nobreak* deste projeto consiste em um circuito regulador de tensão, que regula a tensão para a tensão de flutuação da bateria, o que torna o *nobreak* muito simples. O desenho esquemático do funcionamento do *nobreak* e da fonte de alimentação do circuito do *datalogger* pode ser estudado na Figura 19.

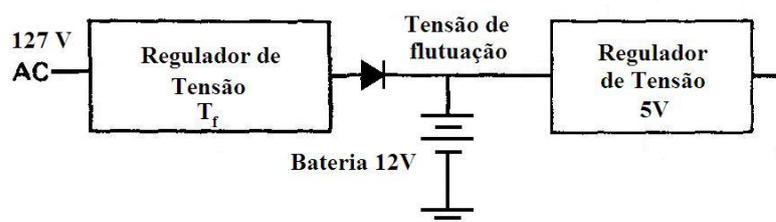


Figura 19 - Diagrama das conexões feitas entre o *nobreak* e a fonte do *datalogger*.

A técnica de tensão constante consiste em um regulador de tensão regulado para fornecer a tensão de flutuação da bateria. Esta tensão de flutuação é fornecida pelo fabricante e é geralmente 15% superior a tensão nominal da bateria, neste caso 13,8 V. Quando a bateria atinge o valor da tensão de flutuação, ela entra em equilíbrio com a fonte e mantém apenas uma pequena corrente de manutenção, que é referente às fugas internas da bateria [31].

Foi feita a montagem em *protoboard* para simular as partes do projeto separadamente e identificar possíveis erros de montagem e programação. Na Figura 20 encontra-se o microcontrolador PIC18F2550 com as conexões dos seus pinos. Podem-se observar também as ligações necessárias para o funcionamento do LCD.

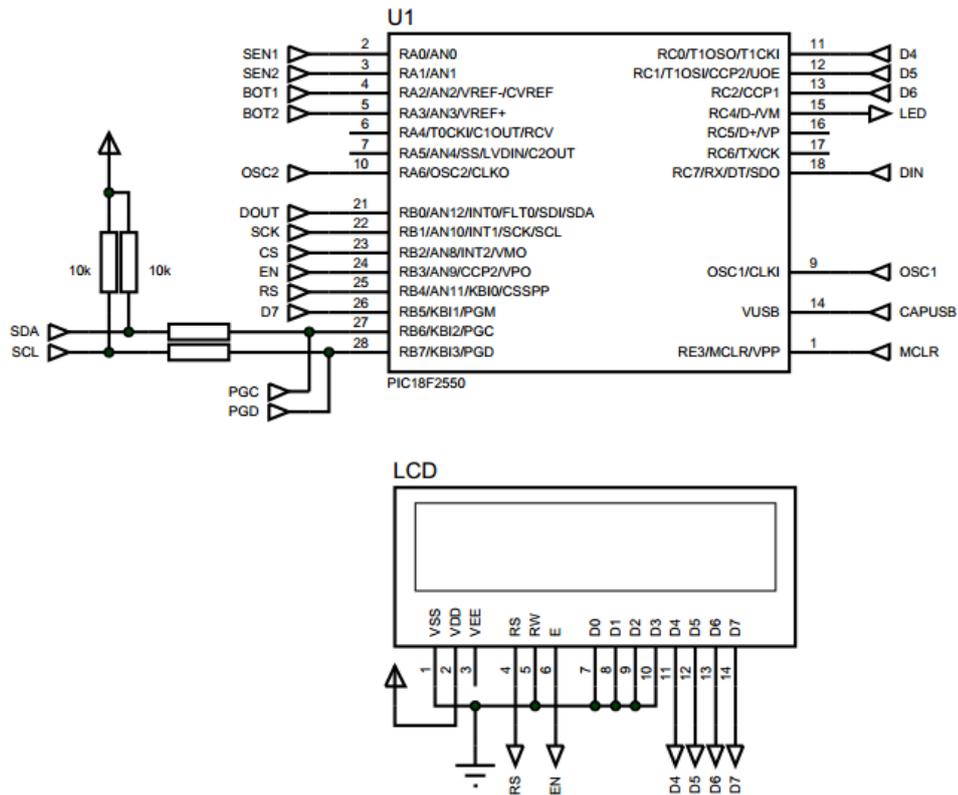


Figura 20 - Esquema elétrico parcial do *datalogger*. Microcontrolador 18F2550 e Display LCD.

Certos pinos do microcontrolador são dedicados a funções específicas, como por exemplo, os pinos de comunicação SPI, I2C, oscilador, etc. Se for utilizado uma destas funções, é obrigatório a utilização dos pinos dedicados a esta função. Isto foi um problema neste projeto, pois foram utilizados os recursos de comunicação I2C e SPI, que dividem os mesmos pinos, que neste microcontrolador são os pinos 21 e 22. Para contornar o problema, foi feita uma biblioteca a parte no *firmware* do microcontrolador para simular o periférico I2C em outros pinos do microcontrolador. Isto custou mais tempo de processamento e espaço na memória do microcontrolador. Os pinos escolhidos para a nova comunicação I2C foram os pinos 27 e 28, que neste microcontrolador são utilizados para a gravação do *firmware in-circuit*. Neste caso não houve problema em dividir os pinos, pois foi feito um circuito de desacoplamento entre esses periféricos. O módulo de gravação *in-circuit*, é desabilitado quando o *datalogger* está em funcionamento.

Cada periférico do projeto foi cuidadosamente alocado nos pinos específicos do microcontrolador. Os sensores de tensão e corrente foram alocados nos pinos onde existem os canais de conversão A/D. Na Figura 21 podem ser observados outros periféricos mais simples que são necessários ao projeto. Nesta figura pode-se observar as ligações do cristal oscilador, conectores do sensor de corrente e do cartão SD, dois botões, tensão de 5 V aplicado no pino MCLR através de uma resistência de 10 k $\Omega$  e um capacitor também obrigatório para o periférico USB. O periférico USB do microcontrolador não será utilizado neste projeto.

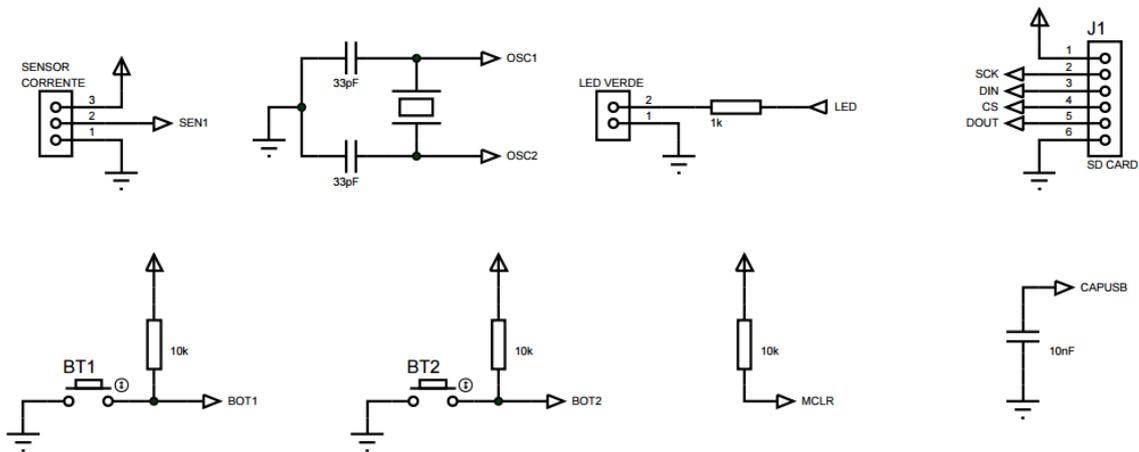


Figura 21 - Periféricos necessários ao funcionamento do *datalogger*.

### 3.2 Algoritmo

O *firmware* gravado no microcontrolador foi compilado a partir do código escrito em linguagem C.

O algoritmo implementado no *datalogger* segue o raciocínio dos fluxogramas apresentados na Figura 22. Quando o *datalogger* é energizado, o microcontrolador irá configurar as entradas analógicas, iniciar as comunicações SPI e I2C, desativar portas não utilizadas e iniciar o arquivo de texto do cartão SD. A princípio o *datalogger* inicia no modo de configuração, no qual é possível alterar o intervalo de tempo entre duas leituras consecutivas. O modo de configuração é identificado pelo LED vermelho aceso e o LED verde apagado. No LCD estará escrito INTERVALO na primeira linha e o número do intervalo em minutos na segunda linha. Para alterar o intervalo de tempo, pressiona-se o botão “Temp”, e o valor do intervalo será ser incrementado. O valor deste intervalo de tempo varia de 1 a 20 minutos com intervalos de 2 minutos cada. Na rotina principal deste modo, o

microcontrolador entra em um *loop* infinito fazendo a verificação dos botões e da variável “Temp”. Quando a variável “Temp” é igual a 10, o próximo incremento ela assumirá o valor 1, fechando o ciclo e retornando ao início.

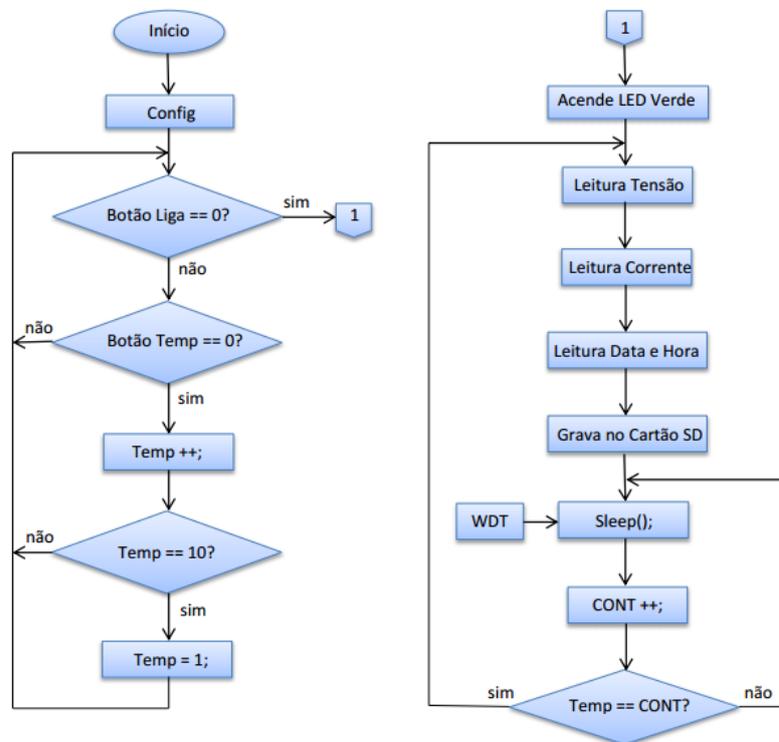


Figura 22 - Fluxograma da rotina do modo de configuração do sistema e modo leitura.

Ao pressionar o botão LIGAR, o *datalogger* entra no modo de leitura de dados. O modo leitura de dados é identificado pelos LEDs vermelho e verde acesos. No modo de leitura o microcontrolador faz as leituras de tensão e corrente, leitura da data e hora e faz a gravação dos dados no cartão SD.

Para economizar energia o microcontrolador entra no modo SLEEP, neste ponto a rotina entra em um *loop* finito que incrementa a variável “CONT” até que ela fique igual à variável “Temp”, então o *loop* é terminado e a rotina volta a fazer o ciclo de leituras e retornar ao *loop* de SLEEP.

Quando o microcontrolador entra no modo SLEEP ele desabilita grande parte dos periféricos para economizar energia, para fazer o microcontrolador voltar a executar a rotina é necessário “acorda-lo”. Existem várias formas de se fazer isto, neste projeto foi utilizado o recurso de *watchdog-timer* ou simplesmente WDT. Este recurso consiste em um contador pré-configurado que dá um RESET na CPU caso a contagem chegue ao fim. Este contador é

independente do clock da CPU. Este recurso é muito útil quando a CPU do microcontrolador trava, quando a contagem do WDT chega ao fim ele reseta a CPU. Para que este recurso não interfira no funcionamento correto da rotina do programa que está sendo executada, deve-se resetar a contagem do contador WDT em ponto específicos na rotina do programa. Quando o microcontrolador está em modo SLEEP, o contador WDT “acorda” a CPU e a rotina volta a ser executada de onde parou. O contador WDT conta apenas até 2,8 segundos, então se faz necessário acrescentar uma variável para incrementar sempre que a CPU “acorda”, se a variável CONT não for igual a variável ‘Temp’, então a CPU volta para o modo SLEEP, senão a CPU volta e refaz as leituras de tensão e corrente fechando assim o ciclo.

A Figura 23 mostra o fluxograma que exemplifica a rotina que faz as leituras das portas analógicas. Quando é feita uma medição em um sinal qualquer, este está sujeito a interferências que causa oscilações aleatórias na informação. Neste trabalho foram feitas a leitura de dois sinais analógicos, o sinal de tensão e de corrente elétrica. Estes sinais contem ruído branco, que pode ser amenizado tirando a média aritmética do sinal lido [32], isto só é possível porque os ruídos tem média igual à zero. A rotina apresentada faz 100 leituras do canal analógico num intervalo de tempo de 100 milissegundos e calcula a média aritmética, e retorna o resultado para ser gravado no cartão SD.

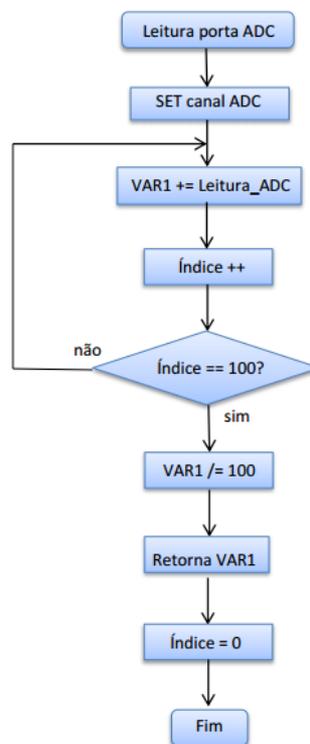
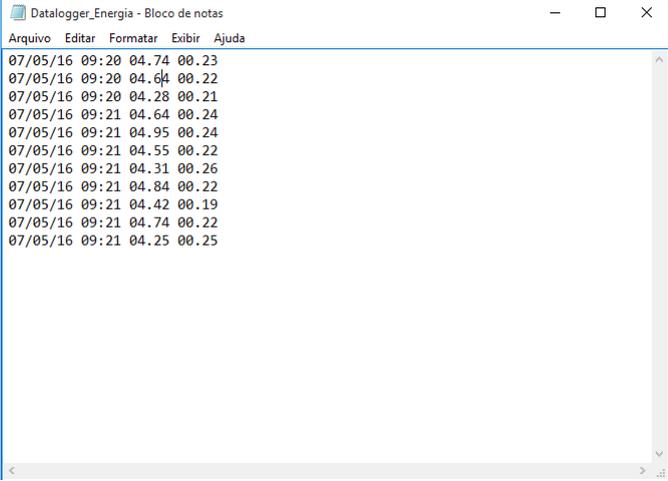


Figura 23 - Fluxograma que exemplifica a rotina que reduz o nível de ruído dos sinais lidos.

## 4 Resultados e Discussões

### 4.1 Arquivo txt

O arquivo gerado pelo *datalogger* tem informações de data, hora, tensão e corrente, todas as informações separadas por espaço. O dia, mês e ano da data é separada por “ / ”, a hora é separada dos minutos através do caractere “ : ”, a tensão é dada em Volts e tem duas casas decimais de precisão, a corrente é dada em Amperes e também tem duas casas decimais de precisão. Na Figura 24 mostra o arquivo gerado pelo *datalogger*, aberto usando um simples leitor de texto.



```
Datalogger_Energia - Bloco de notas
Arquivo Editar Formatar Exibir Ajuda
07/05/16 09:20 04.74 00.23
07/05/16 09:20 04.64 00.22
07/05/16 09:20 04.28 00.21
07/05/16 09:21 04.64 00.24
07/05/16 09:21 04.95 00.24
07/05/16 09:21 04.55 00.22
07/05/16 09:21 04.31 00.26
07/05/16 09:21 04.84 00.22
07/05/16 09:21 04.42 00.19
07/05/16 09:21 04.74 00.22
07/05/16 09:21 04.25 00.25
```

Figura 24 - Arquivo gerado pelo *datalogger* aberto no bloco de notas.

Com o arquivo “.txt” em mãos, pode-se usar vários programas que lê este arquivo como, por exemplo, Excel ou Matlab. Pode-se também programar *software* específico em linguagens específicas como, por exemplo, C/C++, que tem bibliotecas específicas para o tratamento destes tipos de dados, ou outros tipos de linguagem.

Para o teste do *datalogger* foi feito um teste preliminar utilizando uma placa fotovoltaica de 5,5 V. Adicionou-se um diodo comum 1n4048 em série com a placa fotovoltaica para abaixar a tensão em 0,7 V, e assim medi-la diretamente na entrada da porta A/D do microcontrolador. Para o teste da medição durante o dia não foi utilizado o sensor de

corrente. A Figura 25 mostra o exemplo de gráfico gerado a partir do arquivo gerado pelo *datalogger* no referido teste.

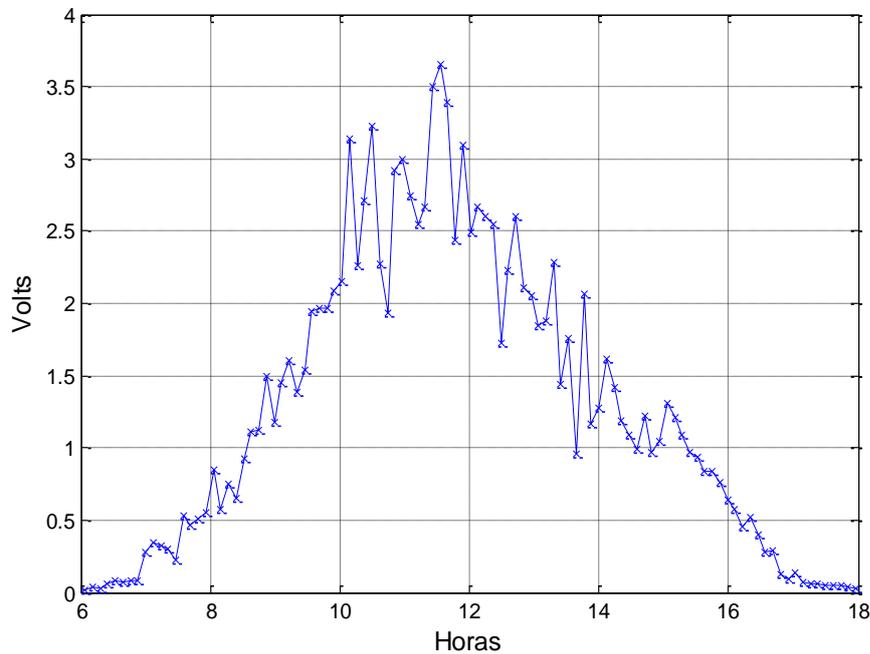


Figura 25 - Gráfico gerado no MATLAB a partir dos dados lidos do arquivo gerado pelo *datalogger*.

No Eixo das abscissas encontram-se as horas do dia e no eixo das ordenadas encontram-se as tensões que a placa fotovoltaica estava gerando. Para gerar o gráfico da Figura 25 foi utilizado ciclos de leitura de 7 minutos.

## 4.2 Precisão das Medidas

O conversor ADC do microcontrolador foi configurado para trabalhar com 10 *bits*, com as referências internas de tensão (tensão de alimentação 0 – 5 V), ou seja, GND e +Vcc. O cálculo da resolução do conversor é feito dividindo a tensão máxima de referência pela quantidade de *bits* que tem o conversor, neste caso  $5/2^{10}$ . Logo este conversor possui 4,88 mV de precisão. Isto significa que para cada *bit* incrementado no conversor, um valor igual a 4,88 mV é acrescido no valor final, isto significa que o valor total lido é sempre um múltiplo de 4,88 mV.

Para o teste da precisão do canal analógico e do algoritmo que usa média aritmética dos valores coletados para reduzir o ruído, foi medido a tensão em um simples divisor de tensão que é alimentado por uma bateria de grande capacidade. A bateria é necessária para

reduzir ao máximo os ruídos externos (menor flutuação). É necessário que a bateria forneça uma corrente constante durante o teste de precisão, isto se consegue com uma bateria de alta capacidade e um circuito de alta impedância, para que este não drene correntes significativas da bateria a ponto da tensão variar. Os resistores utilizados no divisor de tensão são de valores altos de resistência e de precisão 0,5%.

Um multímetro ICEL MD-6110 foi utilizado para verificar as tensões e correntes medidas. No momento do teste a tensão da bateria era de 12,787 V. As resistências utilizadas foram de 44 k $\Omega$ , 5,17 k $\Omega$ , 5,32k $\Omega$ , 5,65 k $\Omega$  e 5,79k $\Omega$ , todas as resistências foram verificadas com o multímetro. Primeiro foi feito um divisor com apenas duas resistências, como mostra a Figura 26, posteriormente foi-se substituindo a resistência R2 por outras resistências de valores diferentes.

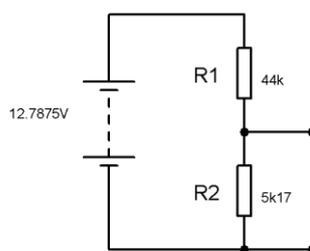


Figura 26 - Circuito divisor de tensão para testes de precisão.

As medidas de tensão foram feitas utilizando o microcontrolador e o multímetro para compará-las. As medidas foram feitas e anotadas na Tabela 1.

Tabela 1 – Teste do medidor de tensão.

Resistência R2	Tensão (V)		Erro (%)
	Multímetro	Datalogger	
5,17 k $\Omega$	1,34	1,4	4,48
10,49 k $\Omega$	2,46	4,3	6,5
16,14 k $\Omega$	3,43	3,3	3,8
21,93 k $\Omega$	4,25	4,1	3,53

Note pela Tabela 1 que se for desprezado todas as perdas e calculadas as tensões do divisor e for feita uma comparação com o multímetro, o multímetro tem erro igual a zero até duas casas decimais. A tensão medida com o protótipo acumula erros, tais erros podem ser pela resolução do conversor aliados com erros de imperícias na montagem, erros sistemáticos, etc.

O teste de precisão do sensor de corrente foi feito utilizando uma resistência de alta potência ligada em uma fonte de corrente cc variável. Um multímetro ICEL MD-6110 foi conectado em série com a resistência para fazer a comparação com o sensor de corrente ASC712. Foi-se variando a corrente no circuito e anotando na Tabela 2 para fazer a comparação entre as leituras de corrente.

Tabela 2 – Teste do sensor de corrente.

Corrente (A)		Erro (%)
Multímetro	Datalogger	
0	0	0
1,51	1,51	0
2,63	2,58	1,9
3,67	3,61	1,63
4,51	4,39	2,66
5,92	5,81	1,85
7,12	6,98	1,96
8,4	8,25	1,78

Nota-se pela observação da Tabela 2 que os erros de leitura são menores que 3%. O módulo de sensor de corrente ASC712 suporta até 20 A, logo sua resolução é de 100 mV/A. A resolução do conversor ADC do microcontrolador é de 4,88 mV, portanto os dois resultam em uma resolução de 48,8 mV. Este valor de resolução introduz erros nas medidas de corrente de baixos valores, para este protótipo que pode suportar até 20 A, a resolução não interferiu muito nas medidas. O erro encontrado neste experimento também pode ser atribuído a má calibração do instrumento usado para a comparação (multímetro ICEL MD-6110), imperícia ao manipular os instrumentos, etc.

### 4.3 Autonomia

O consumo de energia do *datalogger* foi medido com o intuito de calcular a autonomia da bateria do *nobreak*. Primeiro foi feita as medições das correntes de cada módulo separadamente e exposta na Tabela 3. A montagem elétrica do *datalogger* foi concebida sempre levando em consideração o menor consumo de energia possível, por isso optou-se em utilizar conectores “*minimodul*” na placa do *datalogger*, e barra de pinos nos módulo para facilitar a inserção e a remoção dos módulos. Esta medida foi necessária para desconectar os módulos que não estão sendo utilizados, e assim economizar energia. O módulo LCD é um bom exemplo de periférico que só é utilizado para a configuração inicial do sistema, depois de

iniciado o ciclo de leituras, não é necessário mais, então se faz necessário desconectá-lo do sistema para reduzir o consumo de energia.

Tabela 3 – Consumo de corrente dos periféricos.

Periféricos	Consumo (mA)	
	Modo Config.	Modo Leitura
Sensor corrente ASC712	12,4	12,4
Cartão SD	1,4	1,4
LED do LCD	14,9	-
LCD	0,93	-
Relógio RTC	0,15	0,15
Gravador ICSP	47,5	-
Microcontrolador	3,82	0,050
<b>Total</b>	<b>81,1</b>	<b>14</b>

Como exposto na Tabela 3, com todos os periféricos conectados o consumo é de 81,1 mA, isso porque o *display* e o gravador de microcontrolador ICSP consomem muita energia do sistema. Quando iniciado o modo leitura, são desconectados os módulos LCD e gravador ICSP e o microcontrolador fica a maior parte do tempo no modo SLEEP, com essas medidas tomadas, o consumo cai para 14 mA.

Para o cálculo da autonomia da bateria deve-se ater a capacidade da bateria e ao consumo do circuito. Utilizou-se uma bateria de 12V e 7Ah para este protótipo. Para o cálculo do tempo de descarga da bateria, considerou-se primeiro que a descarga da bateria é linear, como o objetivo é simplesmente fazer uma estimativa, esta aproximação é suficiente. A bateria de 7 Ah demora 1 hora para se descarregar quando está fornecendo uma corrente de 7 A, utilizando regra de três inversa, é então calculado o tempo para que a bateria se descarregue fornecendo uma corrente de 14 mA. O tempo calculado é de aproximadamente 20 dias.

A ANEEL possui índices que medem a qualidade e o fornecimento de energia, para que as concessionárias possam padronizar o fornecimento e garantir energia de qualidade para os seus clientes. Um dos indicadores de qualidade de fornecimento é o DMIC, que significa Duração Máxima de Interrupção Contínua. Este indicador mostra o tempo máximo que a concessionária restabeleceu a energia em uma interrupção. Na cidade de Viçosa - MG onde o fornecimento de energia é de responsabilidade da CEMIG, o DMIC foi de 2,69 horas no ano de 2015 [33].

Para cada ciclo de leitura dos sensores, data e hora, uma linha é escrita no cartão SD. O arquivo gerado é gravado no formato “.txt”. Cada linha tem 26 caracteres, com as

informações do dia, mês, ano, hora, minuto, tensão e corrente. Estas informações tem o seguinte formato “dd/mm/aa hh:nn tt.tt cc.cc” no arquivo. Para cada linha gravada, têm-se mais dois caracteres que na verdade são comandos ASCII para o programa que lê o arquivo executar, são eles “*carriage return*” e “*line feed*”, que juntos executa o comando “enter”, para que a próxima informação seja escrita na linha e baixo. Então cada linha gravada acrescenta 28 *bytes* no tamanho do arquivo. Como dito, para que o cartão SD funcione com as bibliotecas nativas do compilador, o cartão deve estar formatado no sistema de arquivo FAT16, o que faz sua capacidade máxima ser de 2 GB de memória.

Para o cálculo da autonomia da capacidade do cartão SD, foi levado em consideração a capacidade total do cartão SD e o menor tempo para gravar duas informações consecutivas. O menor tempo entre duas leituras consecutivas que será configurado no *datalogger* é de 1 minuto, então é acrescido 28 bytes por minuto no cartão SD. 2 GB de memória tem 2126512128 bytes, então são necessários 75946861,714 minutos para esgotar a memória do cartão, ou seja, seriam necessários 144,5 anos aproximadamente para encher a memória do cartão SD.

## 5 Conclusões

O protótipo construído lê e grava com sucesso os valores de tensão e corrente, juntamente com a data e a hora. Quando a aplicação é complexa, é exigido um *firmware* capaz de alterar os parâmetros de leitura para adequar corretamente o sistema, como por exemplo, as casas decimais dos valores lidos, de corrente e tensão. Seria necessário também acrescentar no *firmware* uma rotina de interrupção externa para desligar o modo de leitura e voltar ao modo configuração, e aumentar as opções de leitura de sensores. Os estudos que se podem realizar com os dados gravados são os mais variados, como, por exemplo, estudo do potencial solar/eólico de uma região. Sabendo que a tensão gerada por uma placa fotovoltaica é diretamente proporcional à radiação solar, se for desprezado o efeito da temperatura, pode-se ter estimativas do potencial solar. Da mesma forma a potência produzida por um gerador eólico é proporcional à velocidade do vento, se for desprezadas as perdas intermediárias, como temperatura do estator, atritos, entre outros. Pode-se usar o presente protótipo para estimar a potência consumida por cargas monofásicas. Todos os estudos que necessitem apenas de tensão, corrente e tempo, podem ser feitos utilizando os dados coletados com este *datalogger*. O *software* que será utilizado para ler o arquivo gerado que deverá conter algoritmos capazes de gerar estudos mais completos.

Houve certa dificuldade em se trabalhar com o sensor de efeito *Hall* para medir pequenas correntes, como o módulo do sensor é um produto pronto, não é possível alterar a faixa de leitura de corrente. Isto faz com que a resolução do ASC712 seja sempre fixa, tornando as medidas imprecisas para baixos valores de correntes.

Constatou-se que o consumo de corrente do módulo sensor de corrente é alto comparado com os outros periféricos. Este problema pode ser contornado utilizando um transistor na entrada da alimentação do módulo. O transistor seria montado na configuração de emissor comum e seria usado simulando uma chave para ligar e desligar a alimentação do módulo. O transistor seria polarizado através de um pino de saída do microcontrolador, que ao entrar no modo SLEEP desligaria o módulo de leitura de corrente. Isso faria o *datalogger* ainda mais econômico e consequentemente aumentaria a autonomia da bateria. Se for considerado que a energia onde se encontra instalado o *datalogger* é estável, com valor de

DMIC baixo durante o ano, então é possível reduzir a capacidade da bateria para baratear os custos e melhorar o tamanho e peso do equipamento. Com a bateria utilizada neste projeto, conseguiu-se autonomia de 20 dias, um valor muito alto comparado com 2,69 horas de tempo máximo de interrupção encontrado nesta região.

Com a finalidade de fazer medições durante 1 ano em sistemas fotovoltaicos, o cartão SD de 2 GB é mais que suficiente para a aplicação.

## 6 *Trabalhos Futuros*

O *datalogger* somente é capaz de recolher os dados de um sistema. Os estudos envolvendo os dados coletados são feitas por softwares específicos. A próxima etapa deste trabalho é criar um *software* capaz de ler, separar e gerar todos os estudos possíveis. O *software* poderá gerar gráficos de tensão, corrente e potência. Potência média diária, mensal e anual, com opção de escolher os dias da semana ou semanas do mês etc.

À medida que o sistema aumenta se faz necessário monitorar mais de um sistema, pode-se então aumentar apenas os módulos de sensores de tensão e corrente, e assim monitorar vários sistemas utilizando apenas um *datalogger*. Estes tipos de *dataloggers* se chamam *datalogger* multicanais.

O módulo sensor de corrente é capaz de medir correntes positivas e negativas, e assim podendo ser usado para leituras de corrente alternada. Com algumas poucas modificações no *firmware*, é possível produzir um analisador de qualidade de energia. O *hardware* de aquisição de tensão e corrente é o mesmo, bastando algumas poucas alterações. Acrescentando mais dois módulos sensores de corrente, pode-se fazer um analisador de qualidade de energia trifásico.

Neste trabalho utilizou-se para gravar os dados medidos o cartão SD, esta escolha foi feita para obter baixo consumo e assim permitir que o *datalogger* opere por grandes períodos de tempo, e também operar em locais remotos onde não existam infraestruturas necessárias para manter o *datalogger* funcionando. Quando se tratar de locais onde existam redes de computadores, pode-se trocar o cartão SD por módulos de comunicação *wifi* ou ETHERNET (utilizando protocolo TCP/IP). Pode-se também utilizar, quando existe disponibilidade de computadores próximos, cabos USB ou serial (RS-232). Com estes recursos também é possível fazer o monitoramento em tempo real.

## ***Referências Bibliográficas***

- [01] Instituto de Pesquisas Econômicas Aplicadas (IPEA). Sustentabilidade Ambiental no Brasil: Biodiversidade, economia e bem-estar humano. Brasília - 2010.
- [02] Santos, Marco A.; Inventário de Emissões de gases de efeito estufa derivados de hidrelétricas. Tese de Doutorado em Geografia - Universidade Federal do Rio de Janeiro. Rio de Janeiro, RJ – 2000.
- [03] Pinho, João T.; Galdino, Marco A.; Manual de Engenharia para sistemas fotovoltaicos. CEPREL - CRESSEB. Rio de Janeiro RJ - 2014
- [04] Departamento intersindical de estatística e estudo socioeconômico (DIEESE). Comportamento das tarifas de energia elétrica no Brasil. Brasília - 2015.
- [05] Resolução Normativa Nº 482, de 17 de abril de 2012. Agência Nacional de Energia Elétrica.
- [06] Resolução Normativa Nº 687, de 24 de novembro de 2015. Agência Nacional de Energia Elétrica.
- [07] Amarante, Odilon A. C.; Atlas do Potencial Eólico Brasileiro. Brasília - 2001.
- [08] Agência Nacional de Energia Elétrica (ANEEL). Atlas energia elétrica do Brasil. 2º Ed. Brasília - 2005.
- [09] Ribeiro, Marco A.; Instrumentação. 8º Ed. Salvador – 2002.
- [10] Gonçalves, Marcelo G.; Monitoramento e Controle de Processos: Programa de Qualificação de operadores. Brasília - 2012.
- [11] Nicolosi, Denys E. C.; Microcontroladores 8051. 1º Ed. Érica – 2000.
- [12] Pereira, Fábio; Microcontroladores PIC: Programação em C. 1º Ed. Ed Érica - 2006.
- [13] Datasheet 18F2550. Disponível em <www.microchip.com> acessado em 21 de abril de 2016.
- [14] MICROCHIP. Disponível em <www.microchip.com> acessado em 21 de abril de 2016.
- [15] Penido, Édilus de C. C., Trindade, Ronaldo S.; Microcontroladores. IFET - MG
- [16] Adriano, Jose D.; XM118: Microcontroladores PIC18. Exsto Tecnologia. Santa Rita do Sapucaí – MG.
- [17] UM10204 I2C-bus specification and user manual. NXP Semiconductors N.V. - 2014
- [18] Miyadairo, Alberto N.; Microcontroladores PIC18: Aprenda e Programe em Linguagem C. Érica 1º Ed. São Paulo – 2009.
- [19] Oliveira, Eider L. de, Fideles, Éderson L.: Microcontrolador PIC16F87X. Universidade Católica de Goiás – Goiânia – 2010.
- [20] Datasheet DS1307. Disponível em <www.maximintegrated.com> acessado em 21 de abril de 2016.
- [21] Ramsden, Edward. Hall-effect sensor: Theory and application. 2º Ed. Newnes-Burlington-USA
- [22] Datasheet ASC712. Disponível em < www.allegromicro.com> acessado em: 15 de dezembro de 2015.
- [23] SD Association. Disponível em < www.sdcard.org > acessado em: 14 de março de 2016.
- [24] MikroElektronika. Disponível em < www.mikroe.com > acessado em: 23 de abril de 2016.
- [25] Ziller, Roberto M.; Microprocessadores: Conceitos importantes. 2º ed. Florianópolis - 2000.
- [26] Lafuente, Cesar O.; Carregador de Baterias Monofásico para Aplicação em Veículos Elétricos. Dissertação de mestrado. Universidade Federal do Ceará. Fortaleza – 2011.
- [27] Junior, Rubens N. de F., Zarpelon, Lia M. C., Serna, Marilene M.; Baterias Recarregáveis: Introdução aos Materiais e Cálculos. ArtLiber 1º Ed. São Paulo - 2014
- [28] Rede de Tecnologia Avançada RTA. Manual Técnico: Baterias Seladas Estacionárias Chumbo-Ácido Reguladas por válvula (VRLA).
- [29] Datasheet LM78L05. Disponível em < www.ti.com> acessado em: 30 de janeiro de 2016.
- [30] Trento, Arielton; Feldens, Alexandre T.; Carregador de Baterias Tipo Chumbo Ácido com PIC16F876A. S. José dos Pinhais – PR 2001.
- [31] Manual Bateria Selada VRLA. Disponível em < www.unipower.com.br> acessado em: 30 de janeiro de 2016.
- [32] Haykin, S., Veen, B. V. (2001). Sinais e Sistemas. Bookman - 2002.
- [33] ANEEL. Divulgação dos limites dos indicadores DIC,FIC,DMIC e DICRI. Disponível em <http://www2.aneel.gov.br/aplicacoes/srd/indqual/default.cfm> acessado em: 12 de maio de 2015.