

CONTROLE DE CARGA DE BATERIA COM PAINEL FOTOVOLTAICO ATRAVÉS DO MONITORAMENTO DA CORRENTE E TENSÃO EM TEMPO REAL

A. J. P. Soares , *A. C. Mateus , * M. R. Pereira , A. B. Melo

*LATUR - Laboratório de Tecnologias Urbanas e Rurais

afonso.pujoni@terra.com.br, acmateus@eletrica.ufu.br, matheu_sr_3@hotmail.com, alexentomelo@ufu.br

Universidade Federal de Uberlândia, FEELT - Faculdade de Engenharia Elétrica,

Patos de Minas – MG, Rua Major Jerônimo 566, Sala 204. www.ufu.br

Resumo – O presente trabalho tem como objetivo desenvolver um sistema para monitoramento e controle de carga de baixo custo, utilizando-se painéis solares, um sistema de aquisição de dados via *Bluetooth* e um controlador de carga e descarga. O estudo foi realizado utilizando-se um painel fotovoltaico, um controlador de carga que promove o carregamento de uma bateria conectada a uma carga, com o propósito de atender áreas rurais onde o sistema elétrico convencional ainda não está disponível. Para tal foi realizado o levantamento das curvas de tensão e corrente gerados pelo sistema, a fim de modelar o processo de geração fotovoltaica em uma determinada região.

Palavras-Chave – *Bateria; Carga; Corrente; Fotovoltaico; Iluminação; Tensão.*

BATTERY CHARGE CONTROL WITH PHOTOVOLTAIC PANEL VIA MONITORING OF CURRENT AND TENSION IN REAL TIME

Abstract - This study aims to develop a system for monitoring and cost effective load control, using solar panels, a data acquisition system via *Bluetooth* and a charge controller and unloading. The study was conducted using a solar cell panel, a charge controller which promotes charging a battery connected to a load, for the purpose of serving rural areas where conventional electrical system is not available. For this survey was carried out of the voltage and current curves generated by the system in order to model the process of photovoltaic generation in a particular region.

Keywords - *Battery; Charge; Current; Lighting; Photovoltaics; Voltage.*

I. INTRODUÇÃO

Neste projeto, realizado através de alunos do PIVIC (Programa Institucional de Iniciação Científica Voluntária), foi idealizado o desenvolvimento de um sistema de monitoramento e controle remoto, que atendesse a uma região rural que não fosse atendida por uma concessionária de energia elétrica. A fim de alcançar os objetivos propostos, foi projetado um sistema que deverá monitorar, a princípio, a variação de corrente e tensão fornecida por painéis solares. O sistema de monitoração e controle deverá realizar as medidas de tensão e corrente fornecidas pelo painel solar, assim como as médias de cargas e descargas da bateria utilizada. Para tal utilizou-se módulos de sensores; de tensão e de corrente; para que se pudesse obter a leitura exata dessas medidas.

O módulo de controle deverá alimentar uma bateria até que esta atinja a tensão de serviço, monitorando o consumo da carga. A partir do momento que o painel não conseguir fornecer mais tensão e corrente, este será desligado, permanecendo a carga alimentada somente pela bateria, conforme a Figura 1.

Foi utilizado um sensor de tensão para uso em microcontroladores de 5 V (Arduino®), que faz leituras até 25 V, baseando-se no princípio do divisor de tensão. O modelo do sensor de corrente utilizado foi o ACS712 do fabricante Allegro MicroSystems, disponível em [1], que faz leituras de até 25 A, utilizando-se do princípio do Efeito Hall, que está relacionado a uma diferença de potencial em um condutor elétrico, transversal ao fluxo de corrente e um campo magnético perpendicular à corrente.

Para a comunicação foi utilizado um módulo *Bluetooth*® modelo HC-06 do fabricante Guangzhou HC, cujo *datasheet* está disponível em [2], que é usado para comunicação *wireless* entre o microcontrolador Arduino® e algum dispositivo que utilize esse padrão, como um telefone celular ou *tablet*. Os dados podem ser lidos através de um aplicativo instalado no dispositivo, que recebe as leituras enviadas pelo módulo. O módulo tem alcance de até 10 metros com o dispositivo que irá receber os dados.

Foi criada uma interface na qual podem ser monitoradas em tempo real as leituras dos dados de tensão e corrente. Para tal foi utilizado o *software MATLAB*®, do desenvolvedor *MathWorks* [3], escrevendo-se um código específico, onde são capturados os valores lidos pelos sensores, gerando-se gráficos de tensão *versus* tempo e corrente *versus* tempo.



XIV CEEL - ISSN 2178-8308
03 a 07 de Outubro de 2016
Universidade Federal de Uberlândia - UFU
Uberlândia - Minas Gerais - Brasil

II. PROCEDIMENTO EXPERIMENTAL

A fim de desenvolver o sistema de monitoramento para realizar as leituras de tensão e corrente geradas pelo painel fotovoltaico, conectou-se os condutores de saída deste à entrada do sensor de tensão e a saída do sensor de tensão à entrada analógica do CAD (Conversor Analógico Digital) do Arduino®, dessa forma os valores apresentados no monitor serial serão as leituras convertidas para o formato digital de 0 a 255 (2^8 , 8 bits).

O sensor de corrente foi ligado em série com a carga (lâmpada de 12 W), a fim de se determinar a corrente fornecida pelo painel em função da insolação, que é variável com a hora do dia. O valor máximo registrado utilizando-se um miliamperímetro foi de 0,365 A.

A Figura 1 apresenta o esquema de ligação do circuito montado. Neste processo se tem o painel solar ligado a um controlador responsável pela carga e descarga da bateria e pelo controle da lâmpada.

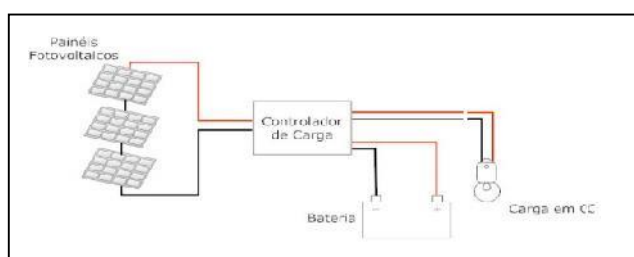


Fig. 1. Esquema ligação do Controlador de Carga.

A comunicação utilizada baseou-se na arquitetura WPAN (Wireless Personal Area Network) através da tecnologia IEEE 802.15, também conhecida como *Bluetooth*®. Para tanto foi utilizada uma biblioteca compatível com o Arduino® para permitir a transferência dos dados dos sensores para o sistema de monitoramento.

O código abaixo, escrito na linguagem C, na IDE (Integrated Development Environment) do Arduino®, mostra a forma com que o microcontrolador realiza as medições de tensão e corrente. Para o envio de dados via *Bluetooth* também se utilizou uma biblioteca específica (DHT.h). O desenvolvimento do código foi baseado em [4 e 7].

```
#include "DHT.h" //biblioteca para uso do Bluetooth
#define dht_pin1 A0 //Pino DATA do Sensor ligado
na porta Analógica A0
#define dht_pin2 A4 //Pino DATA do Sensor ligado
na porta Analógica A4

#define DHTTYPE DHT11

DHT dht(dht_pin1, DHTTYPE);
DHT Dht(dht_pin2, DHTTYPE);

int valor1; // declaração da variável inteira valor1.
const int sensorPin = A4; // declaração da variável inteira
igualando a entrada do sensor.

float sensorValue_aux = 0; // declaração da variável do tipo
```

float igual a zero.

```
float sensorValue = 0; // declaração da variável do tipo float
igual a zero.
```

```
float currentValue = 0; // declaração da variável do tipo float
igual a zero.
```

```
float voltsporUnidade = 0.0048828125; // declaração da
variável do tipo float que é igual a proporcionalidade da
entrada do sinal pela unidade correspondente.
```

```
void setup() {
```

```
Serial.begin(115200); // Baud Rate.
}
```

```
void loop() {
```

```
for (int i = 500; i > 0; i--) { // criação do loop ou laço
```

```
sensorValue_aux = (analogRead(sensorPin) - 511); // lê o
sensor na pino analógico A4 e ajusta o valor lido já que a
saída do sensor é VCC/2 para corrente = 0.
```

```
sensorValue += pow(sensorValue_aux, 2); // soma os
quadrados das leituras no laço.
```

```
}
```

```
sensorValue = (sqrt(sensorValue / 500)) * voltsporUnidade;
// finaliza o calculo da média quadrática e ajusta o valor
lido para volts.
```

```
currentValue = (sensorValue / 185) * 1000; // calcula a
corrente considerando a sensibilidade do sensor (185 mV
por A).
```

```
sensorValue = 0;
```

```
delay(1000); // tempo de atraso de 1000 ms.
```

```
valor1 = analogRead(A0); // entrada analógica A0 inserida
na variável valor1
```

```
float tensao = ((valor1 * 0.00489) * 5.07); // cálculo para o
valor real de tensão
```

```
Serial.print(currentValue); // comando para a exibição da
leitura no monitor serial.
```

```
Serial.print(tensao); // comando para a exibição da
leitura no monitor serial
```

```
delay(100); // tempo de atraso de 100 ms.
```

```
}
```

O código utilizado para a criação da interface no MATLAB® para receber os dados em tempo real é detalhado abaixo:

```
clc;
```

```
clear all;
```

```
comport = serial('COM3', 'BaudRate', 115200); % setup
comport fopen(comport); % Open comport
```

```
x=0;
```

```
while(x<inf)
```

```

x=x+1;

y1(x)=fscanf(comport, '%f'); %#ok<SAGROW> % receive
ADC1 y2(x)=fscanf(comport, '%f'); %#ok<SAGROW> %
receive ADC2 drawnow;

subplot(2,1,1);
plot(y1,'r--','linewidth',3)% axis ([ 0 inf 0 0.400]);
xlabel('Tempo (s)');
ylabel('Corrente (A)');
grid on;

hold on;

subplot(2,1,2);
plot(y2,'b--','linewidth',4)
title('Valores em Tempo Real');
xlabel('Tempo(s)');
ylabel('Tensao (V)');

grid on;
pause(0.1);
end

fclose(comport); % Close comport
delete(comport); % Clearcomport

```

III. CIRCUITO

Baseado nas informações obtidas em [5 e 6] foi montado o circuito apresentado na Figura 2, onde é mostrado o circuito responsável pela ligação do circuito de controle de carga e descarga da bateria, assim como o do desligamento do painel, quando este não consegue fornecer carga ao sistema.

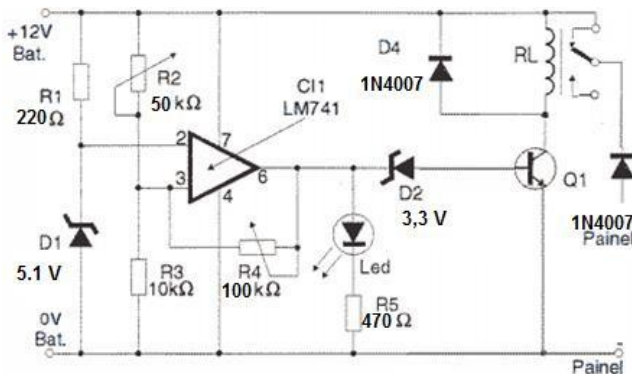


Fig. 2. Circuito montado para a experimentação.

IV. FUNCIONAMENTO DO CIRCUITO:

Conforme mostrado na Figura 2, o resistor R1 polariza o diodo zener D1 através de R1, o que mantém a tensão em seu catodo em 5,1 V, conseqüentemente a entrada inversora (pino 2) também apresenta esse valor. O CI1 (LM741) funciona como um comparador de tensão: Quando a tensão na entrada inversora (pino 2) é maior que a da entrada não-inversora (pino 3), a saída (pino 6) apresenta um nível baixo de tensão (aproximadamente 1,5 V), deixando o LED apagado e, devido ao diodo zener (D2) ligado na base

do transistor Q1, o relé não é energizado, pois não circula corrente de base e o transistor fica em corte (não conduzindo corrente no coletor).

Supondo-se que a bateria esteja inicialmente descarregada (abaixo de 7,0 V), o divisor de tensão formado por R2 e R3 é ajustado para que, no pino 3, tenhamos um valor de tensão abaixo de 5,1 V. Uma vez que, com o relé não energizado, temos o painel solar em paralelo com a bateria através do contato NF (normalmente fechado) do mesmo e, em situação de incidência de radiação solar sobre o painel, a bateria começa a ser carregada, aumentando a tensão em seus terminais. Quando a carga da bateria atinge um valor que torna a tensão no pino 3 do LM741 maior que 5,1 V, ocorre a mudança de estado da saída (pino 6) do mesmo (indo para mais de 4,0 V), que acende o LED e provoca uma corrente de base no transistor, levando-o à situação de saturação. Dessa forma o relé RL é fechado e o painel solar é desconectado da bateria.

À medida que a bateria se descarrega devido à carga (lâmpada), a situação inicial volta a ocorrer, abrindo o relé e voltando a conectar o painel à bateria, repetindo o processo de carga. O diodo D4 tem a função de proteger o transistor Q1 e é conhecido como “diodo de roda livre”, eliminando a reação à variação do fluxo magnético causada pela bobina do relé ao ser ligada e desligada. O painel é conectado ao relé através de um diodo 1N4007 a fim de impedir que a corrente flua da bateria para o mesmo, na falta de incidência de radiação solar.

V. RESULTADOS

Após ser executado o código na IDE do Arduino®, foram observados os valores em tempo real de tensão e corrente em um aplicativo, conforme apresentado na Figura 3.

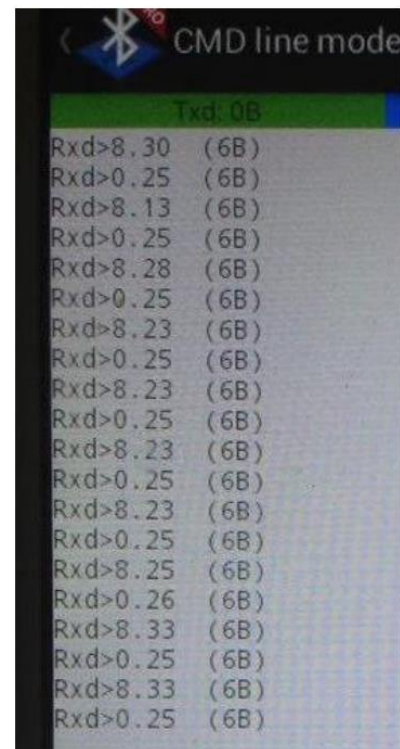


Fig. 3. Aplicativo recebendo os dados de tensão e corrente em tempo real através do módulo Bluetooth.

REFERÊNCIAS

Ao ser executado o código no MATLAB®, são apresentadas as leituras dos dados em tempo real e gerados gráficos de corrente versus tempo e de tensão versus tempo, conforme apresenta a Figura 4.

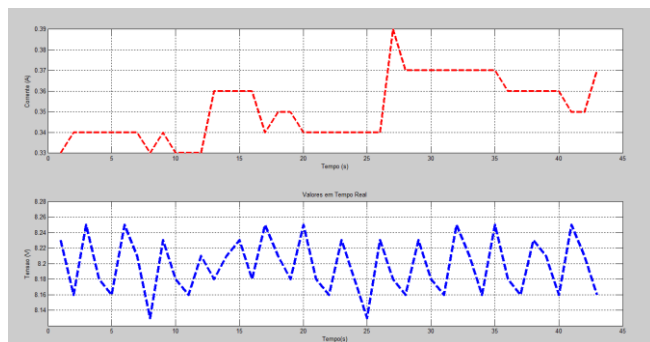


Fig. 4. Gráficos gerados pelo MATLAB

A interface desenvolvida mostra de que forma será o monitoramento do sistema fotovoltaico. Pode-se, através dos dados adquiridos, compreender o que ocorre no sistema, em diversas situações, verificando se o painel consegue fornecer carga para a bateria e energia para a lâmpada.

Observou-se que os valores de tensão e corrente, conforme apresentados na Figura 2, apresentaram valores de 8.23 V e 0.30 A, respectivamente. Foi constatado que à medida que a bateria vai sendo carregada, o valor da corrente vai diminuindo gradativamente.

Através do uso de um cronômetro e multímetros, observou-se que, para a carga utilizada, o tempo de descarga da bateria foi de aproximadamente 60 minutos e o tempo de carga foi de aproximadamente 1h30min.

VI. CONCLUSÕES

O projeto se mostrou eficiente e de baixo custo. Após a realização de vários testes em diferentes datas e horas do dia, foi possível se testar diversas situações.

O desenvolvimento da interface gráfica possibilitou uma análise mais precisa do comportamento do circuito e do painel em diversas situações. Com base nesses dados foi possível compreender os processos de carga e descarga da bateria, além de otimizar o processo de coleta de dados permitindo o conhecimento do funcionamento do painel fotovoltaico em função da radiação solar incidida na região onde está instalado.

O processo demonstrou ser viável para a aplicação em localidades nas quais não existe rede elétrica convencional, na iluminação de espaços públicos e de residências, sendo que, para tanto, bastaria se redimensionar a capacidade de geração do painel a partir da demanda de carga solicitada, utilizando um número maior de painéis para a geração de energia e de baterias, para acumular a mesma.

Com esta interface pode se analisar a capacidade de geração de energia de fontes renováveis, em uma determinada localidade, pelo levantamento de dados que o sistema fornece, através do monitoramento da corrente e tensão.

Como propostas para o futuro estão o aperfeiçoamento desta plataforma, criando um sistema supervisório mais robusto, no qual se possa ter uma interface mais amigável e que forneça ao usuário, além de informações como corrente e tensão, alarmes de defeitos no circuito, temperatura ambiente, umidade, potência fornecida e insolação, com o objetivo de se otimizar o processo.

[1] Allegro MicroSystems. Disponível em: <http://www.allegromicro.com/>. Acesso em: 09 de Setembro de 2016

[2] Olimex. Disponível em: <https://www.olimex.com/Products/Components/RF/BLUETOOTH-SERIAL-HC-06/resources/hc06.pdf>. Acesso em: 09 de Setembro de 2016

[3] MathWorks. Disponível em: <http://www.mathworks.com/>. Acesso em: 09 de Setembro de 2016

[4] Matsumoto, Élia Yathie, Matlab R2013a - *Teoria e Programação* - Guia Prático - 1ª Ed.

[5] BOYLESTAD, R. L., NASHELSKY, I., *Dispositivos eletrônicos e teoria de circuitos*. 11ª Edição. São Paulo: Pearson Prentice Hall, 2013.

[6] MALVINO, A. P., *Eletrônica: volume 2*. 7ª Edição. São Paulo: Makron Books, 2007.

[7] OXER J., BLEMINGS H., *Practical Arduino, USA*: APRESS, 2009.