Posicionamento Painel Fotovoltaico

Autores:

Sérgio Santos Nº 1020881

1020881@isep.ipp.pt

LABSIS 2021

0:00 / 19:15

• Introdução

Início

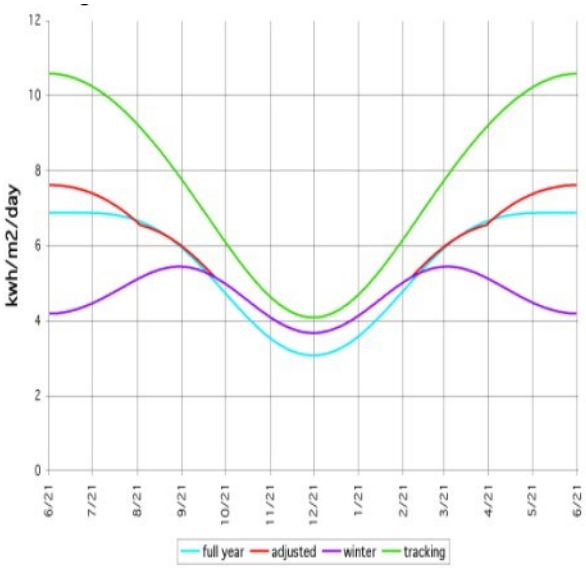
Resumo

O projeto que se pretende fazer é um sistema de posicionamento de painel fotovoltaico para obter maior rendimento de produção de energia, já que é conhecido que se pode tirar proveito até 40%. O objetivo é desenvolver um meio de controlo tendo em mente ser um sistema Stand Alone de tamanho considerável, o trabalho não vai ser concentrado ao redor do painel fotovoltaico em si e suas características ou modos de funcionamento, mas apenas o sistema de **Sun Track** , também supondo que tem baterias de carga com o intuito que o sistema possa ser autónomo.

O projeto é apenas académico e de simulação em escala pequena ou de bancada.

Apresentação

Os painéis fotovoltaicos tem vários parâmetros que determinam sua eficiência, a temperatura de funcionamento, otimização da carga, sua orientação e até limpeza são alguns pormenores, este estado de arte apenas é uma tentativa de melhor garantir tirar maior proveito da energia coletada pelo meio de orientação.



Neste gráfico podemos ver um exemplo da diferença entre o rendimento de sistemas sem controlo de orientação, orientação parcial e total.

Resposta a inclinação

Mais energia pode ser coletada ao fim do dia se o Painel fotovoltaico tiver instalado um sistema de orientação solar, através de um atuador mecânico.

Existe dois tipos de sistemas de orientação solar:[1]

1. **Sistema de um eixo, este segue o posicionamento do sol durante o dia de Este a Oeste.**

2. **Sistema de dois eixos, o mesmo que de um eixo mais a orientação Norte e Sul que tem em consideração a influencia de inclinação provocada pelas estações do ano.**

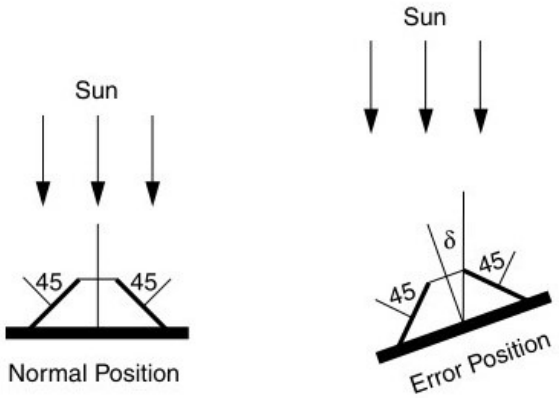
Estado da Arte

O projeto em causa pertence a disciplina de Laboratório de Sistemas que da continuidade as disciplinas de Sistemas Digitais [9], sendo o foco processadores, micro-controladores e Fpga´s (Field programmable arrays). Um Micro-controlador da **ATmel** é o que vai ser usado como o cérebro do sistema, o tipo de sistema de orientação vai ser de um eixo, um servo motor para simulação de posicionamento e dois sensores LDR (Light dependent resitor) ligados a uma ponte wheatstone e amplificador de instrumentação para ajuste fino. Um RTC (Real Time Clock) como ferramenta de posicionamento principal, e disponível um override manual, sendo seus dados visíveis num LCD (Liquid crystal display).

O motor a ser usado se for numa aplicação real seria de corrente continua de preferência com íman permanente [2] , com uma caixa redutora e sem fim para que quando estiver em estado de paragem poder se desligar sem perder sua posição, já que se sabe que a melhor maneira de poupar energia é não a gastar. Como o objetivo é obter o maior rendimento possível pretende-se não haver desperdício, se houve-se meio de não usar energia para seu posicionamento seria o ideal.
No mundo das energias renováveis só é justificado sua implementação se for em grande escala pois seus rendimentos são baixos e intermitentes, e melhor ainda se os preços forem atrativos, a não ser que haja algum avanço tecnológico de relevo.

O sistema de posicionamento pode aumentar a energia acumulada até 40 % durante o ano comparado com os sistemas fixos, durante o dia o painel acompanha a posição dos sol de Este para Oeste e durante a noite regressa a posição Este para o dia seguinte, os sistemas antigos tinham uma bateria para esta operação depois do pôr do sol, os novos modelos já não utilizam bateria mas usam a luz fraca do por do sol para regressar a origem. [1]

Um dos métodos de orientação solar é usar duas células PV em serie com polarização oposta em ângulo de 45 ° como demonstrado na figura:



Principio de Funcionamento Sensor [1]

Assim o atuador mecânico (motor) recebe o diferencial da corrente fornecida, estando assim sempre orientado a fonte de luz. A corrente no motor é dada pela expressão $I_m = I_1 - I_2 = 2 I_o δ Sin(45 °) = √2 I_o δ se δ é em radianos. [1]

No Projeto não vai ser usado este método mas um análogo utilizando dois sensores LDR também em serie e a 45 ° um do outro dando uma saída proporcional ao desfasamento da fonte de luz apenas para ajuste fino, já que o posicionamento vai ser sincronizado pelo RTC (relógio), já existe algoritmos criados usando os parâmetros de localização e tempo que nos fornece os dados do local exato da posição do sol, sendo desnecessário a utilização de sensores, aqui este projeto é flexível pois já íntegra um relógio RTC (Real time clock) com calendário de atualização automática.

O objectivo deste trabalho é executar um sistema de orientação solar fiavel utilizando um **RTC** e sensor **LDR** , com sistema interactivo através de **display** e **keypad** na qual tem integrado um override de positionamento manual para caso de manutenção, também com ligação ao **PC** através da porta **USB** para troubleshooting indicado a leitura da hora e posição do sensor a pedido, podia também integrar um control total pelo PC através de comandos criados. Um servo motor vai ser usado como simulação do sistema para comprovar sua orientação e circuito do sensor, na qual em practica seria aplicado outro tipo de motor, mas para propositos académicos é o mais practico.

Quanto a melhorias, isto como é um sistema programado a imaginação é o horizonte pois com facilidade pode-se acrescentar funcionalidades ou alterações para preencher o pretendido e desejado.

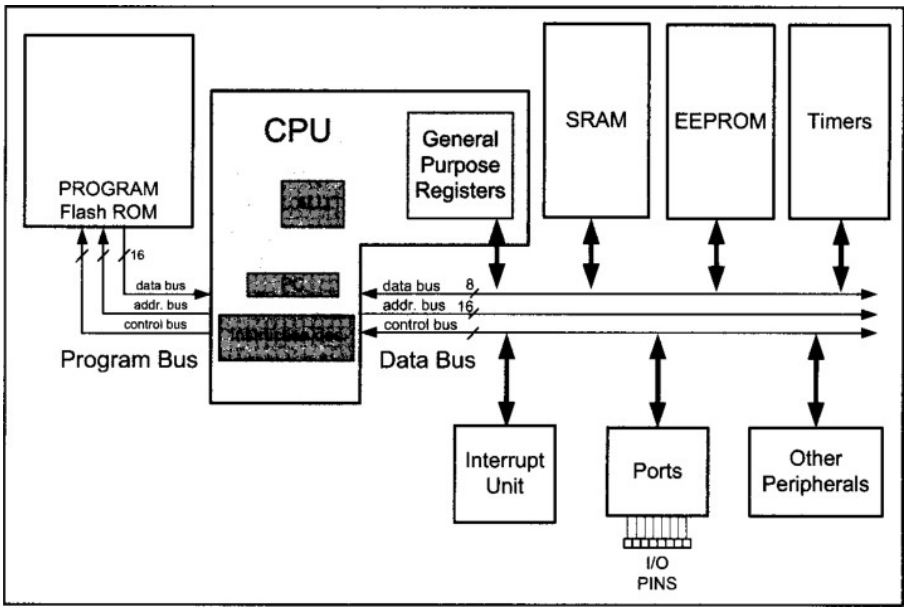
• Arquitetura

Início

Tipo de Microcontrolador

Os microcontroldores da **Atmel** de 8 e 32 bits são baseados na arquitetura avançada de **Harvard** na qual esta concebido para baixos consumos e performance.
Este tipo de arquitetura tem dois busses (barramentos) um dedicado a leitura das instruções a executar e outra para escrita e leitura de data (informação ou dados), isto assegura que uma nova instrução pode ser executada em cada ciclo de relógio, na qual elimina estados de espera quando não ha instruções prontas a executar.
Nos microcontroladores da AVR os barramentos estão configurados de forma a dar prioridade ao barramento das instruções do CPU acesso a memoria flash enquanto o barramento da CPU de dados tem prioridade de acesso a **SRAM** (Static Random Access Memory).

O espaço de memoria de dados é dividida em três, os **GPR** (General Purpose Registers) as **SFRs** (Special Function Registers) ou memoria de I/O e a data SRAM.
Os microcontroladores da AVR utiliza uma arquitetura de instruçõeses **RISC** (Reduced Instruction Set Computer ou Reduced COMPLEXITY Instruction Set Computer) na qual reduz a complexidade dos circuitos na codificaçãoo de cada instrução.
Dai que os microcontroladores que se baseiam nestes tipos de arquitetura são sinonimo de código reduzido, alta performance e baixo consumo energético.



Harvard Architecture

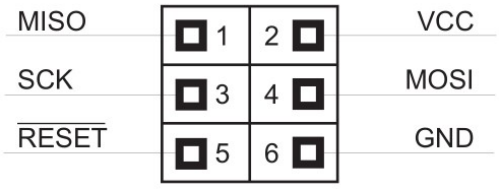
Diagrama de blocos do projecto

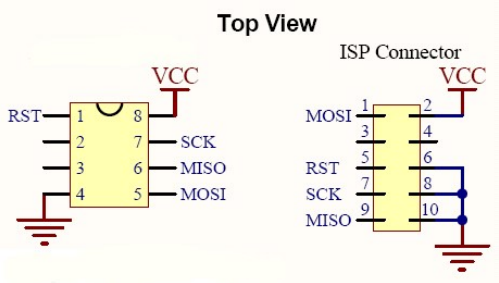
- # Hardware

Início

Ferramentas

Neste projecto foi utilizado um microcontrolador da Atmel especicifamente o Atmega 128 [9] [12]. Foi usado o programador Atmel Ice também do mesmo fabricante, na qual veio a ser muito útil especialmente em combinação com o Atmel Studio 7 , este tem disponivel programação via **ISP** e **JTAG** , ao lado esta o kit de programação e as fichas de programação indicando as connecções.
Os parametros do MCU são:
Device signature = 0x1E9702; LOW Fuse = 0xBF; HIGH Fuse = 0xC7; EXTENDED Fuse = 0xFF e os LOCK BITS off, ou seja, 0xFF.





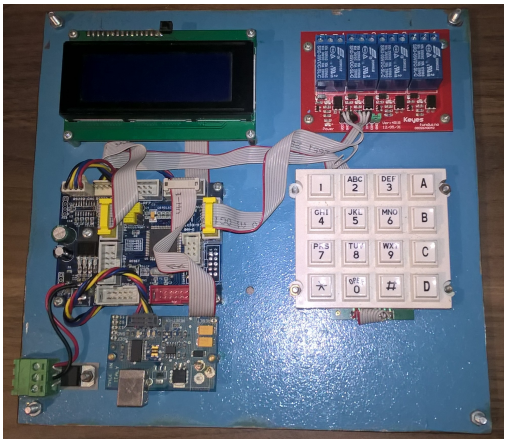
Equipamento

- ET-BASE AVR ATmega128
  Mainboard com MCU Atmega 128 da ETT.
- LDR - Light Controlled Resistor
  Dark resistance 1M.
  40k at 10 lux.
  Max voltage 100V.
  Max power 80mW.
- INA128PA - Instrumentation Amplifier
  Banda de transmissão: 1.3MHz
  Montagem: THT
  Número de canais: 1
  Carcaça: DIP8
  Rapidez de subida de tensão: 4V / µ s
  Temperatura de trabalho: -40...85°C
  Entradas de tensão instável: 0.025mV
  Tensão de trabalho: 2.25...18V
- LCD 20x4 Blue NHD-0420DZ-NSW-BBW
  4x20 Characters HD44780 compatible

https://www.ptrobotics.com/
https://www.futurlec.com/index.shtml

- Model Craft RS-2 \- Servo Motor
  Control System:
  +Pulse Width Control 1500 µ sec
  Operating Voltage:
  4.8-6.0 Volts
  Operating Speed (4.8V):
  0.19sec/60° at no load
  Stall Torque (4.8V):
  42 oz/in (3.0 kg/cm)
- PCF8563 RTC Board
  Real-time clock/calendar.
  Battery on board.
  I2C communication.
- KEYPAD4X4W
  16 Button Keypad. switch

Foi criado um kit de desenvolvimento para facilitar sua programação e simulação, na qual seus perifericos são todos ligados por fichas IDC Socket atraves de flat-cable tornado-se num sistema de plug and play . A lista de material esta acima representado.



esquemas

- **Software**

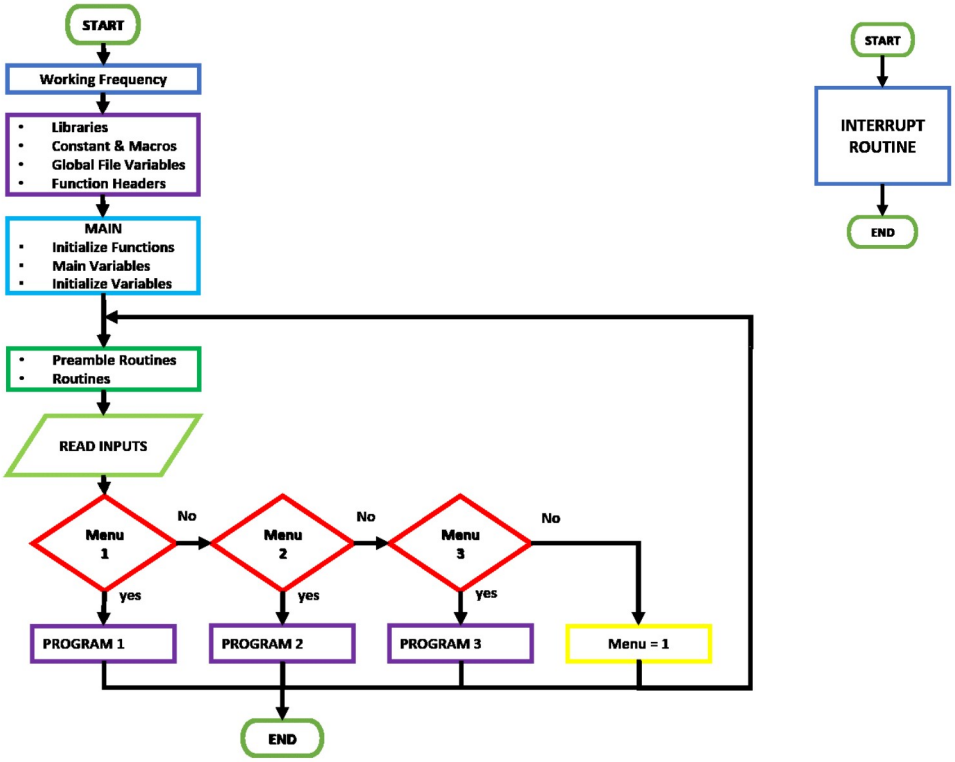
Início

Programa

O programa utilizado para executar este projecto foi o Atmel Studio 7 (Version: 7.0.2397), 2019 Microship Technology.

Já nos é conhecido que a marca **Atmel** foi comprada pela companhia que representa os PICS ou seja a Microchip sendo que seu **IDE** também já supporta seus microcontroladores e programadores, o **MPLAB XC** é agora uma boa alternativa.

Fluxograma



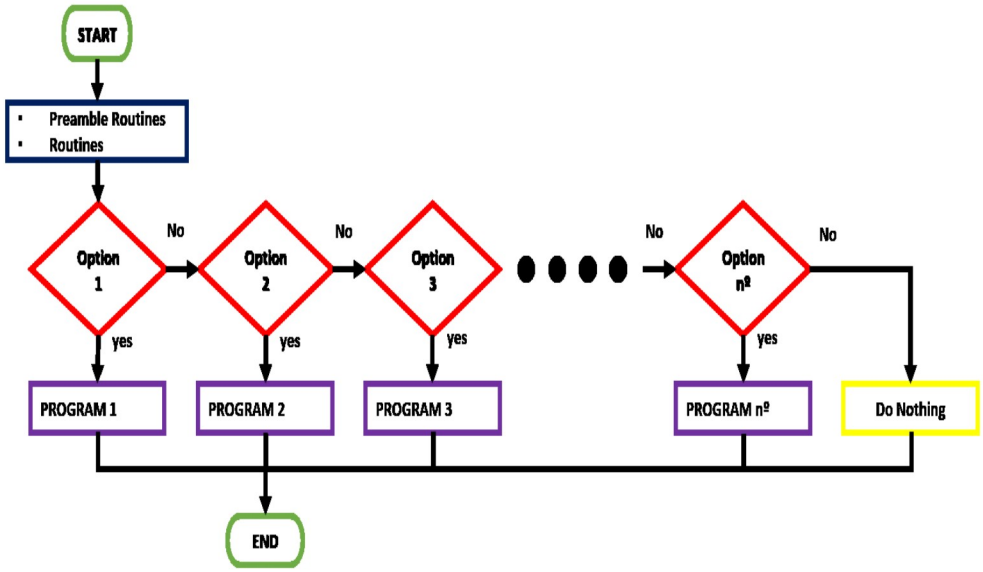
Acima demonstra a metodolgia usada para execução do Programa Principal, as subrotinas, isto é os programas 1 até 3 obedecem a mesma estrutura, que esta indicado abaixo, em que o bloco READ INPUTS é partilhado com o Programa Principal. Também de notar que o bloco **READ INPUTS** inplica sinais de entrada na qual pode vir de qualquer origem, isto é também pode ser leituras de saidas, uart, i2c, etc. Outro pormenor é que só são aceites leituras diferentes das lidas anteriormente, assim não existe redundância e o programa flui de forma **"One Shot"** podendo este entrar e sair pelo sistema sem qualquer interferência se o sinal não for reconhecido, mas só uma única vez.

```
void PORTINIT();
/****MAIN****/
int main(void)
{
    PORTINIT(); // Inic Ports
    /***INICIALIZE OBJECTS***/
    function= FUNCenable(); // Function Library
    LCD0 lcd0 = LCD0enable(&DDRA,&PINA,&PORTA); // LCD Display 4X20
    KEYPAD keypad = KEYPADenable(&DDRE,&PINE,&PORTE); // Keyboard
    ANALOG analog = ANALOGenable(1, 128, 1, 0); // Channel 0 for Position
    TIMER_COUNTER0 timer0 = TIMER_COUNTER0enable(2,2); // 1Hz to HC595
    TIMER_COUNTER1 timer1 = TIMER_COUNTER1enable(9,0); // PWM Positioning
    rtc = PCF8563RTCenable(16); // RTC with I2C
    shift = HC595enable(&DDRG,&PORTG,2,0,1);
    uart = UART1enable(103,8,1,NONE);//UART 103 para 9600, 68 para 14400
    /******/
    char Menu='1'; // Main menu selector
    uint16_t adcvalue; // analog reading
    char str[6]="0"; // analog vector
    int16_t mvalue=90; // manual position reading
    int16_t m_value; // manual positioning
    char mstr[6]="90"; // manual position vector
    char tstr[6]; // time vector
    char cal='0'; // Sub Menu for setting up date and time
    uint16_t set;
    char uartmessage[64];
    ptr=str;
    uint16_t positionhour=12;
```

```
//TODO:: Please write your application code
while(TRUE){
    /***PREAMBLE***/
    lcd0.reboot();
    keypad.read();
    uartreceive=uart.read();
    /***Reading input***/
    lcd0.gotoxy(3,13);
    lcd0.putch(':');
    lcd0.string_size(keypad.get().printstring,6);
    /***ENTRY END***/
    switch(Menu){
        /***MENU 1***/
        case '1': // Main Program Menu ···
        /***MENU 2***/
        case '2': // Manual position override ···
        /***MENU 3***/
        case '3': //Set Time and Date···
        default:
            Menu='1';
            break;
    };
}
```

```
ISR(TIMER0_COMP_vect) // 1Hz and usart Tx
{
    uint8_t Sreg;
    Sreg=SREG;
    SREG&=~(1<<7);
    if(count>59){ //59 -> 1Hz
        increment++;
        if((increment & 0x0F) < 8){
            shift.bit(0);
            shift.out();
        }else{
            shift.bit(1);
            shift.out();
        }
        count=0;
    }else
        count++;
    SREG=Sreg;
}
/***EOF***/
```

Assim temos um sistema escalavel (**"recursion"**) e de fácil manuseamento.



A programação foi feita em linguagem C [3] aplicado nos integrados da AVR [8].
Anexado [referências] tem links para literatura acerca destes assuntos.

• Resultados

Início

evidencias, imagens, fotografias, videos

• Conclusões

Início

O sistema de posicionamento fotovoltaico tem diversas soluções de diferentes Empresas que

apostam nas energias renováveis, diferentes métodos na parte mecânica e elétrica, como esta disciplina pertence na área digital, considero importante a manipulação dos datasheets dos componentes e o código, manipular o hardware unicamente por programação através de uma camada de interface [API] acho de maior importância pois cria uma abstração dos problemas que podemos enfrentar e os resolver seguindo uma metodologia sintática, sendo possível o manipular para fazer o que é requerido e desejado. Através do código podemos alterar, modificar e adaptar qualquer projeto ao mundo exterior usando uma linguagem de nível alto ou médio com facilidade.

Dai este projeto pode ser atualizado a qualquer altura acrescentando funcionalidades e dispositivos que possam beneficiar sua performance, caso necessário.

Este projeto esta online no link descrito, https://github.com/sergio1020881/LABSIS20202021/tree/main/Relatorio_2 , tudo que é necessário para o desenvolver e executar esta ao dispor, o programa usado foi o **Atmel Studio 7** basta fazer download, compilar e programar o chip [ATmega128], e depois ligar os periféricos respeitando as ligações.

Dado este trabalho como concluido, posso dizer que satisfiz os objectivos pretendidos, existe sempre a possibilidade de alterações, mas considero que esta funcional e practico, como um exercicio académico, para aplicar na vida real necessita apenas de pequenas alterações, como utilizar um motor diferent para supportar a carga precisa.

Neste projecto não inclui a parte mecânica toda, além da restante instalação do sistema electrico que um painel fotovoltaico necessita.

- **Test Code** Início

Page 1 Page 2 Page 3 Page 4 Page 5 Page 6 Page 7 Page 8 Page 9

- **Referências** Início

LIVROS

[1] Mukund R. Patel — *Wind and Solar Power Systems Design, Analysis and Operation Second Edition* , Taylor & Francis Group LLC, 2006.

[2] Stephen W Fardo, Dale R. Patrick — *Electrical Power Systems Technology, Third Edition* , CRC Press, 2008.

[3] Brian W. Kernighan, Dennis M. Ritchie — *The C Programming Language, Second Edition* , Prentice Hall, Software Serries.

[4] Stephen W. Fardo, Dale R. Patrik — *Electrical Power Systems Technology, Third Edition* , The Fairmont Press, 2009.

[5] George Chryssis — *High-Frequency switching Power Supplies: Theory and Design, Second Edition* , McGraw Hill Book, 1989.

[6] Rudolf F. Graf, William Sheets — *Encyclopedia of ELECTRONIC CIRCUITS, Volume 4* , TAB BOOKS (McGraw Hill), 1992.

[7] Hank Zumbahlen — *Linear Circuit Design Handbook* , Analod Devices, 2008.

[8] Steven F, Barret — *Embedded Systems Design with the Atmel AVR Microcontroller* , Morgan Claypool Publishers, 2009.

Manuais

[9] Sistemas Digitais 2, ISEP; Tranparências de Sistemas Digitais 2, 2008/2009.

WebSites

[10] science.nasa.gov; https://science.nasa.gov/science-news/science-at-nasa/2002/solarcells

[11] app.diagrams.net; https://app.diagrams.net/

Datasheets

[12] Datasheet Atmega 128; Micro-controlador Atmega128

[13] Datasheet PCF8563; RTC (Real Time Clock)

[14] Datasheet INA128; Amplificador de Instrumentação

Início