

CAPÍTULO 6

Conclusões

6.1 Considerações Finais

A integração de diferentes aproveitamentos energéticos renováveis perfaz de uma maneira mais confiável o suprimento energético de uma determinada carga. A pouca disponibilidade de uma das fontes de geração, pode ser suplementada pela outra. Em caso de falta mútua, deve-se suprir emergencialmente através de banco de baterias. Contudo, modificações, problemas esporádicos e de mau funcionamento tornam os sistemas de geração alternativa suscetíveis às mais diversas falhas possíveis.

De fato, o equipamento construído pode servir como uma ferramenta modular, portátil, eficaz e segura em pequenos aproveitamentos eólicos e fotovoltaicos para geração distribuída. Por ser modular, o protótipo desenvolvido é facilmente adaptado à nova configuração de carga e/ou sistemas de geração. Evidencia-se eficaz para pequenas potências o controle pela carga, aliado à busca do máximo aproveitamento de potência disponível.

Sendo solução para geração descentralizada e autônoma, através de fontes renováveis, a integração de fontes contribui pelo fato de causar impacto ambiental, limitando-se ao sombreamento da área onde estão instalados os painéis solares, efeito estroboscópico, sonoro ou outro inconveniente do sistema eólico.

A possibilidade de acréscimo de uma ou mais fontes ao sistema é plenamente viável, bastando para isso a definição dos parâmetros individuais das fontes, a implementação do software, a calibração dos novos sensores e o dimensionamento adequado dos conversores de potência.

6.2 Principais contribuições

Os resultados do presente trabalho mostraram que é plenamente possível integrar diferentes fontes alternativas de energia. Dentre as contribuições desta dissertação, podemos citar:

- O estabelecimento de uma plataforma expansível para estabelecimento e avaliação de diferentes métodos de controle, integração e configurações de fontes alternativas de energia.
- Integração de dois ou mais sistemas de geração através de um controle simples, respeitando as características individuais das fontes.
- Baixo custo, elevado conteúdo tecnológico, robustez e versatilidade.

6.3 Sugestões para continuação deste trabalho

- Implementação de um sistema que utilize células de combustível ou mais outras fontes de geração que pode ser usado para estudos de operação, pré-operação, otimização, qualidade de energia e outros.
 - Promover a integração do módulo com um conversor CC-CA para injeção na rede.
 - Inserir um sistema de predição da disponibilidade energética com antecedência de alguns dias, alimentado por informações meteorológicas.
 - Inserir um conversor abaixador-elevador não isolado para estabilizar a tensão no barramento da carga principal.

6.4 Resumo do Capítulo

Neste capítulo, apresentamos as razões de ter sido realizado este trabalho, suas principais contribuições e também apresentamos sugestões para a continuação do mesmo.

BIBLIOGRAFIA

Introdução

Para o desenvolvimento deste trabalho, foi fundamental a revisão bibliográfica para que pudéssemos estabelecer referências e utilizar como base os estudos feitos anteriormente sobre o assunto a ser trabalhado. Por isso citamos a seguir os trabalhos e autores consultados.

[1] – PORTOLANN, C.A., Farret, F.A., and Machado, R.Q., **“Electrical Regulation by the Load of the Electrical Variables and Speed of Induction Micro Turbogenerators”**. IV Congresso Brasileiro de Eletrônica de Potência (“COBEP 97”), Belo Horizonte pp 125-129, 1997.

[2] – FARRET, F. A. ,et al, **Comparison of the Hill Climbing Control Methods Used in Micro Powerplants**. IV INDUSCON Procedures, Porto Alegre, Nov/2000.

[3] –SADHU, D.P., **Estudos Sobre Energia Eólica**. Dissertação (Mestrado em Engenharia Elétrica) Departamento de Engenharia Mecânica, UFRGS, 1983.

[4] – RICH, E., **Inteligência Artificial**. Ed. McGraw-Hill, São Paulo, 1998.

[5] – RIGHI, L. A. , **Associação de Microgeradores Assíncronos Sem Controle de Velocidade para Geração de Energia Elétrica a Partir de Fontes Renováveis**. Dissertação (Mestrado em Engenharia Elétrica) UFSM, Santa Maria, 1998, p.63 a 83.

[6] – BUEHRING, I.K.. et al, **Control Policies for Wind-energy Conversion Systems**. IEEE. Procedures. Vol 128, nº 5. Sep/1981

[7]– FITSCHER, L.L., **Direcionamento Ativo de Turbinas Eólicas de Eixo Horizontal sem Utilização de Sensores**. Dissertação (Mestrado em Engenharia Elétrica). UFSM, Santa Maria, 2001.

- [8] – SCHUCH, L., **Sistema CA/CC com um Conversor PWM Bidirecional para Interfaceamento entre o Barramento CC e o Banco de Baterias**. Dissertação (Mestrado em Engenharia Elétrica) UFSM, Santa Maria. 2001.
- [9] – FISCHER W., **Stationary Lead-Acid Batteries an Introductory Handbook**. Hoppecke Batterien, 1994.
- [10] – KIEHNE, H.A., **Battery Technology Handbook**, Dekker, 1998.
- [11] – GRAY, A. Wallace, G.A., **Eletrotécnica – Princípios e Aplicações**. Ed. Rio de Janeiro: Livros Técnicos e Científicos. 1978.
- [12] – RYNKIEWICKZ, R., **Discharge and Charge Modeling of Lead-Acid Batteries**. IEEE Applied Power Electronics Conference.
- [13] – SIEMENS. **Electrical Engineering Handbook**, Berlim, 1976.
- [14] – VUTETAKIS, D. D., BATTELLE, h.Wu., **The Effect of Charge Rate and Depth of Discharge on The Cycle Life of Sealed Lead-Acid Aircraft Batteries**, IEEE 35th International Power Sources Symposium, P.103-105, 1992.
- [15] – MCANDREWS, J.M. Jones, R.H., **A Valve Regulated Lead-Acid Battery Management System (VMS)**. IEEE International Telecommunications Energy Conference, P.507-513, 1996.
- [16] – UNITRODE. **Products & Applications Handbook**. 1995-96.
- [17] – FARRET, F. A., BUENO, J. **Controle Eletrônico de Velocidade e Tensão para Microgeradores Assíncronos usando Microprocessadores**. Anais do Congresso Brasileiro de Eletrônica de Potência – UFSC, 1991.
- [18] – MELLO Jr. C. D., **Geração Assíncrona Conectada à Rede Pública com Controle Adaptativo pela Carga**. Dissertação (Mestrado em Engenharia Elétrica) UFSM-CEEMA, Santa Maria, 1994.

- [19] – RASHID, M., **Power Electronics**, Livro, Ed. Prentice Hall, International Editions.
- [20] – FUCKS, M. G., **Eficiência e Confiabilidade de Sistemas Industriais e de Distribuição usando Baterias e/ou Geradores à Diesel com Vistas à Otimização Discreta**. Dissertação, (Mestrado em Engenharia Elétrica) UFSM - 1998.
- [21] – BRAGA, N. C., **Células à Combustível – Um Novo Conceito na Produção de Energia**, Saber Eletrônica, ano 33, nº 295, Out/1997.
- [22] – NUNES, M. V., BEZERRA, U. V., **Proposta de Regulação de Tensão para um Gerador de Indução com Rotor em Gaiola de Esquilo Acoplado a um Sistema de Captação de Energia Eólica**; Anais do III Congresso Latino-americano de Geração e Transmissão de Energia Elétrica, Campos do Jordão, São Paulo, Brasil, Novembro de 1997.
- [23] – VALE, S. B. do, U. H. BEZERRA, J. T. Pinho e E. J. da S. Pereira, **Estudo da Inserção de Célula a Combustível Integrada a Sistema Híbrido de Geração de Eletricidade Isolado**". 2º WICaC - Workshop Internacional de Células a Combustível, Unicamp - Campinas, out. 2004.
- [24] – HUYNCH, P., CHO, B., H., **Design and analysis of a microprocessor-controlled-peak-power-tracking system**. IEEE Transactions on Aerospace and Electronic Systems, Vol. 32, No. 1 January 1996 (182-190).
- [24] –PATEL, M. **Wind and Solar Power Systems**, CRC Press, 1999.
- [25] – BURTON, T., et al. **Wind energy handbook**, Wiley, 2001.
- [26] – LARMINIE, J., DICKS, A. **Fuel cell systems explained, 2nd ed.**, Wiley, 2003.
- [27] – BURTON, T., et al. **Wind energy handbook**, Wiley, 2001.

- [28] – BORBELY, A., KREIDER, J.M. **Distributed generation the power paradigm for the new millennium**, CRC Press, 2001.
- [29] – ZUMERCHIK, J. **Macmillan encyclopaedia of energy**, Gale group, 2001.
- [30] – HORDESKI, M. **New technologies for energy efficiency** Fairmont Press 2003.
- [31] – HUA, C., SHEN, C. **Study of maximum power tracking techniques and control of DC/DCconverters for photovoltaic power system** IEEE Power Electronics Specialists Conference, 1998 (86-93).
- [32] – CHIANG, M., HUA, C., LIN, J. **Direct power control for distributed PV power system** IEEE Power Conversion Conference, 2002 (311-315).
- [33] – MASOUM, M., DEHBONEI, H., FUCHS. **Theoretical and experimental analyses of photovoltaic systems with voltageand current-based maximum power-point tracking** IEEE Transactions on Energy Conversion, 2002 (514- 522).
- [34] – KROUTROPOULIS, E. KALITZAKIS, K. VOULGARIS, N. **Development of a microcontroller-based, photovoltaic maximum power point tracking control system** IEEE Transactions on Power Electronics, 2001 (46-54).
- [35]– KROUTROPOULIS, E. KALITZAKIS, K. **Design of a maximum power tracking system for wind-energy-conversion applications** IEEE Transactions on Industrial Electronics, 2006 (46-54).
- [36] – ITO, R., MATSUKI, R., TANI, T., YACHI, T. **Evaluation of performance of MPPT equipment in photovoltaic system** INTELEC 2003 (256–260).
- [37] – FEMIA, N., PETRONE, G., SPAGNUOLO, G., VITELLI, M. **Optimization of perturb and observe maximum power point tracking method** IEEE Transactions on 2005 (963–973)
- [38] – JUNG, Y., SO, J., YU, G., CHOI, J. **Improved perturbation and observation method (IP&O) of MPPT control for photovoltaic power systems** IEEE Photovoltaic Specialists Conference, 2005. (1788–1791).

APÊNDICE I

SOFTWARE DE CONTROLE DO SISTEMA

Este apêndice mostra o programa desenvolvido C para: 1) controle baseado no algoritmo Hill Climbing para obter máxima potência individual de cada fonte primária; 2) controlar o estado de carga e descarga das baterias (níveis de tensão e corrente); 3) regulação da tensão na carga principal.

1) Controle das fontes

```

////////////////////////////////////
//
////////////////////////////////////

#include <18f452.h>
#define ADC=10
#include <Delay(Clock=1000000)>
#include <RS232(Baud=9600,Xmit=PIN_C6,Rcv=PIN_C7,brgh1ok)>

#include <lcdd.c>

#define HS,NOWDT,NOPROTECT,NOLVP

//-----
// Faz 32 leituras e retorna o valor da media do canal a/d em questão
//-----

float media_ad(byte canal) {

    int i,amostras = 10;
    float media = 0, temp_media = 0;

```

```

set_adc_channel(canal);
delay_us(10);
for (i=1; i<=amostras; ++i)
{
    media += read_adc();
    delay_us(10);
}
return(media/amostras);
}
//-----
// ajusta o ciclo de trabalho do PWM, através da função set_pwm1_ciclo, e é
// chamada com uma percentagem do ciclo e a frequência máxima do pwm
//-----
void pwm1_ciclo100(long pwm_percent1, long pwm_maximum1) {

    set_pwm1_ciclo((pwm_maximum1/100)*pwm_percent1);    // PWM #1 ciclo
cycle
}

void pwm2_ciclo100(long pwm_percent2, long pwm_maximum2) {

    set_pwm2_ciclo((pwm_maximum2/100)*pwm_percent2);    // PWM #1 ciclo
cycle
}

//-----
// principal
//-----
main()
{
    CONST float CARREGADA = 14.4;                // inicializa constantes and
variaveis

```

```

CONST BYTE BARRAMENTO = 12;
CONST long NDELTA= -1, NPDELTA = -1;
CONST byte AMP_CANAL = 0, BAMP_CANAL = 3, BVOLT_CANAL = 2,
VOLT_CANAL = 1;
float corrente, tensao, potencia=0.0, potencia_pv = 0.0,
old_potencia=0.0,old_potencia_pv = 0.0;
float batamps, batvolts, batpotencia;
float temp_potencia, potencia100, amp100, volt100;
long delta =1, pdelta = 1, pwm_max1, pwm_max2, pwm100, pwm200;
long boost = 0, boost_counter;
byte led_count = 0;

// hardware
output_low(PIN_C3); // desliga fets, desabilitando
setup_port_a(ALL_ANALOG); // canais A/D
setup_adc(adc_clock_div_32); // seta A/D clock
setup_timer_2(T2_DIV_BY_16, 49 , 2); // pwm, @ 49 50kHz
setup_ccp1(CCP_PWM); // Configura CCP1 como PWM
setup_ccp2(CCP_PWM);
pwm_max1 = 100;
pwm_max2 = 100; // maximo valor para PWM baseado na
frequencia
pwm100 = 50;
pwm200 = 50; // seta pwm 50% para comeco de
conversa
pwm1_ciclo100(pwm100, pwm_max1); // seta pwm ciclo cycle em pwm100
pwm2_ciclo100(pwm200, pwm_max2);
output_high(PIN_C3); // seta enable pin FET driver
output_high(PIN_C0); // liga led
delay_ms(1000); // espera tudo se acomodar..
lcd_init();
while(TRUE) { // laco perpetuo
// verifica maxima tensao na bateria,

```

```

    if (batvolts > CARREGADA) { // se batvolts > carregado, bateria
carregada
        delta = NDELTA; // inicia vazao de corrente ateh < carregada
        output_high(PIN_C5); // liga led indicando maxima carga na
bateria
    } // inicia HCC para procura do pico de
potencia

    else if (old_potencia > potencia) { // else, se bateria nao carregada,
muda o valor de
        delta = -delta; // delta para mudar o pwm, aumento ou
diminuicao, para maximizar potencia
        output_low(PIN_C5); // desliga led se bateria nao esta com tensao
limite
    }
    old_potencia = potencia; // carrega old_potencia com potencia atuais
para proxima interacao
    pwm100 += delta; // adiciona delta para mudar o ciclo de
trabalho para o algoritmo PPT
    if (pwm100 > 100) pwm100 = 100; // verifica limites do ciclo de
trabalho do pwm e ajusta entre 0-100
    else if (pwm100 < 0) pwm100 = 0;
    pwm1_ciclo100(pwm100, pwm_max1); // chama funcao para ajustar o pwm
no pic
    if (tensao > BARRAMENTO) { // se batvolts > carregado, bateria
carregada
        pdelta = NPDELTA; // inicia vazao de corrente ateh < carregada
        output_high(PIN_C5); // liga led indicando maxima carga na
bateria
    } // inicia HCC para procura do pico de
potencia

    else if (old_potencia_pv > potencia_pv) { // else, se bateria nao
carregada, muda o valor de
        pdelta = -pdelta; // delta para mudar o pwm, aumento ou
diminuicao, para maximizar potencia

```

```

    output_low(PIN_C5);           // desliga led se bateria não esta com tensão
limite
}
    old_potencia_pv = potencia_pv;           // carrega old_potencia com potencia
atuais para proxima interacao
    pwm200 += pdelta;           // adiciona delta para mudar o ciclo
de trabalho para o algoritmo PPT
    if (pwm200 > 100) pwm200 = 100;           // verifica limites do ciclo de
trabalho do pwm e ajusta entre 0-100
    else if (pwm200 < 0) pwm200 = 0;
    pwm2_ciclo100(pwm200, pwm_max2);           // chama função para ajustar o pwm
no pic

    corrente = media_ad(AMP_CANAL)/20.885; // le corrente e tensão do painel solar
para ir para a bateria
    tensao = media_ad(VOLT_CANAL)/8.2887; // usa a função media_ad, para
retornar a media
    batamps = media_ad(BAMP_CANAL)/31.3428; //
    batvolts = media_ad(BVOLT_CANAL)/20.4831;//

    potencia = corrente * tensao;           // potencia solares com tensao e corrente de
entrada
    batpotencia = batamps * batvolts;           // bateria tb

    pwm1_ciclo100(100, pwm_max1);           // ajusta o ciclo de trabalho do PWM
ciclo para 100% para dar o valor da conexao direta
    delay_ms(20);           // entre o painel solar e a bateria
    amp100 = media_ad(AMP_CANAL)/30.885; // lê amperes solares e tensão
utilizando as constantes de medida
    volt100 = media_ad(VOLT_CANAL)/18.2887;
    if (amp100 < 0.05) {           // enquanto PWM ajustado para 100% ,
verifica se ha sol

```

```

        output_high(PIN_C4);           // se corrente < .05 desliga o PPT e liga led
sinalizador
        output_low(PIN_C3);           // desabilita chaves
        delay_ms(10);                 // atraso para desligar
    }                                  // aqui o sistema desliga se corrente solar
insuficiente
    }

                                //envia todos valores pela porta serial
printf(" %lu %lu %6.2f %6.2f %6.2f %6.2f %6.2f %6.2f\r\n",
pwm100, pwm200 , corrente, tensao, potencia, batamps, batvolts, batpotencia);

output_bit(PIN_C0, (++led_count & 0x01)); //pisca led para dizer q o software
esta rodando
delay_ms(50);                       //espera um pouco para a proxima interacao
lcd_gotoxy(1,1);
printf(LCD_PUTC,"V%3.2fi%3.2fc%lu",corrente, tensao,pwm100);
lcd_gotoxy(1,2);
printf(LCD_PUTC,"VB%3.2fiB%3.2fc%li",batvolts,batamps,pwm200);
} //final formal do while perpetuo
} //final do main.

```

2) Controle da carga e descarga das baterias e da tensão do barramento

```

////////////////////////////////////
//
//
////////////////////////////////////

```

```
#include <18f452.h>
```

```
#DEVICE ADC=10

#use Delay(Clock=10000000)

#use RS232(Baud=9600,Xmit=PIN_C6,Rcv=PIN_C7)//,brgh1ok)

#include <lcdd.c>

#fuses HS,NOWDT,NOPROTECT,NOLVP

//CANAIS DO A/D

#define AMP_BAT    0
#define VOLTS_BAT  1
#define AMP_BAR    2
#define VOLTS_BAR  3

//PINAGENS DIVERSAS
#define ATIVIDADE  PIN_C0
#define BAR_OK     PIN_B0
#define HAB_BAT    PIN_B1
#define DES_BAT    PIN_B2
#define BAT_OK     PIN_B3
#define BAT_BXA    PIN_B4
#define BAT_ALT    PIN_B5
#define PV_OK      PIN_B6
#define WES_OK     PIN_B7
//#define

//CONSTANTES

#define CARREGADA  14.4
```

```

#define ABAIXO 11.5
#define BARRAMENTO 12.5
#define N_DELTA_BAT -1
#define N_DELTA_BAR -1

////////////////////////////////////
///                                     ///
///      1 -> BAT                       ///
///      pwm                             ///
///      2 -> CARGA                       ///
///                                     ///
////////////////////////////////////

//-----
// Faz 10 leituras e retorna o valor da media do canal a/d em questão
//-----

float media_ad(byte canal)
{

    int i,amostras = 10;
    float media = 0, temp_media = 0;

    set_adc_channel(canal);
    delay_us(10);
    for (i=1; i<=amostras; ++i)
    {
        media += read_adc();
        delay_us(50);
    }
    return(media/amostras);
}

//-----

```

```

// ajusta o ciclo de trabalho do PWM, atraves da funcao set_pwm1_ciclo, e eh
// chamada com uma percentagem do cciclo e a frequencia maxima do pwm-
//
//-----

void pwm1_ciclo_bat(long pwm_percent1, long pwm_maximum1)
{
    set_pwm1_ciclo((pwm_maximum1/100)*pwm_percent1);    // PWM #1 ciclo
cycle
}

void pwm2_ciclo_carga(long pwm_percent2, long pwm_maximum2)
{
    set_pwm2_ciclo((pwm_maximum2/100)*pwm_percent2);    // PWM #2 ciclo
cycle
}

//-----
// principal
//-----

main()
{
    float corrente_bar, tensao_bar;
    float corrente_bat, tensao_bat;
    float temp_potencia_wes, potencia100, amp100, volt100;
    signed char delta_bat =1, delta_bar = 1;
    long pwm_max_carga, pwm_max_bat, pwm_bat, pwm_carga;
    byte led_count = 0;

                                // hardware

    output_low(PIN_C3);           // desliga fets, desabilitando
    setup_port_a(ALL_ANALOG);    // canais A/D

```

```

setup_adc(adc_clock_div_32);           // seta A/D clock
setup_timer_2(T2_DIV_BY_1, 49 , 2);    // pwm, @ 49 50kHz
setup_ccp1(CCP_PWM);                   // Configura CCP1 como PWM
setup_ccp2(CCP_PWM);
pwm_max_bat = 200;
pwm_max_carga = 200;                   // maximo valor para PWM baseado na
frequencia
pwm_bat = 50;
pwm_carga = 50;                        // seta pwm 50% para comeco de
conversa
pwm1_ciclo_bat(pwm_bat, pwm_max_bat);  // seta pwm ciclo cycle em
pwm_wes
pwm2_ciclo_carga(pwm_carga, pwm_max_carga);
output_high(PIN_C3);                   // seta enable pin FET driver
output_high(ATIVIDADE);                // liga led
delay_ms(1000);                        // espera tudo se acomodar..
lcd_init();
while(TRUE)
{
    // laco perpetuo
    // verifica maxima tensao na bateria,

    tensao_bar=media_ad(VOLTS_BAR);     //Mede tensao do barramento
    tensao_bat=media_ad(VOLTS_BAT);     //mede tensao na bateria

    if(tensao_bat > ABAIXO && tensao_bar > BARRAMENTO)
    {

        delta_bat=delta_bat;
        delta_bar=delta_bar;
        output_high(BAT_OK);

    }

    if (tensao_bar < BARRAMENTO) //Verifica barramento, se menor decrementa
    razao ciclica

```

```

{
    //para subir tensao no barramento

    delta_bar = N_DELTA_BAR;           // delta para mudar o pwm, aumento
ou diminuicao, para maximizar potencia
    output_high(HAB_BAT);              // solicita acao da bateria se ela
comportar

}

if (tensao_bat > CARREGADA)
{
    // se batvolts > carregado, bateria carregada -protecao sobrecarga

    delta_bar = N_DELTA_BAR;           // inicia vazao de corrente ateh <
carregada
    output_high(BAT_ALT);              // Sinalizacao de bateria acima do
adequado

}

// if (tensao_bat >

// else
if (tensao_bar > BARRAMENTO)
{
    // else, se bateria nao carregada, muda o valor de

    delta_bar = DELTA_BAR;             // delta para mudar o pwm, aumento
ou diminuicao, para maximizar potencia
    output_high(BAR_OK);               // liga led de barramento acima do
estipulado.

}

```

```

// old_potencia_wes = potencia_wes;           // carrega old_potencia com
potencia atuais para proxima interacao
    pwm_bat += delta_bat;                       // adiciona delta para mudar o ciclo
de trabalho para o algoritmo HCC
    if (pwm_bat > 100)
        pwm_bat = 100;       // verifica limites do ciclo de trabalho do pwm e ajusta
entre 0-100
    else if (pwm_bat < 0) pwm_bat = 0;

    pwm1_ciclo_bat(pwm_bat, pwm_max_bat);       // chama funcao para ajustar
o pwm no pic
//inicia proxima interacao

    pwm_carga += delta_bar;                     // adiciona delta para mudar o ciclo
de trabalho para o algoritmo HCC
    if (pwm_carga > 100) pwm_carga = 100;       // verifica limites do ciclo de
trabalho do pwm e ajusta entre 0-100
    else if (pwm_carga < 0) pwm_carga = 0;
    pwm2_ciclo_carga(pwm_carga, pwm_max_carga); // chama funcao para
ajustar o pwm no pic

    if (tensao_bat < 5 && tensao_bar < 9 )
    {
        // enquanto PWM ajustado, verifica disponibilidade energetica
        output_high(PIN_C4);                   // se corrente < .05 desliga o hcc e liga led
sinalizador
        output_low(PIN_C3);                     // desabilita chaves
        delay_ms(10);                           // atraso para desligar
    }
    // }                                         // aqui o sistema desliga se condicoes insuficientes

// }

//envia todos valores pela porta serial

*/
printf(" %lu %6.2f %lu %6.2f\r\n",pwm_bat, tensao_bat, pwm_carga, tensao_bar);

```

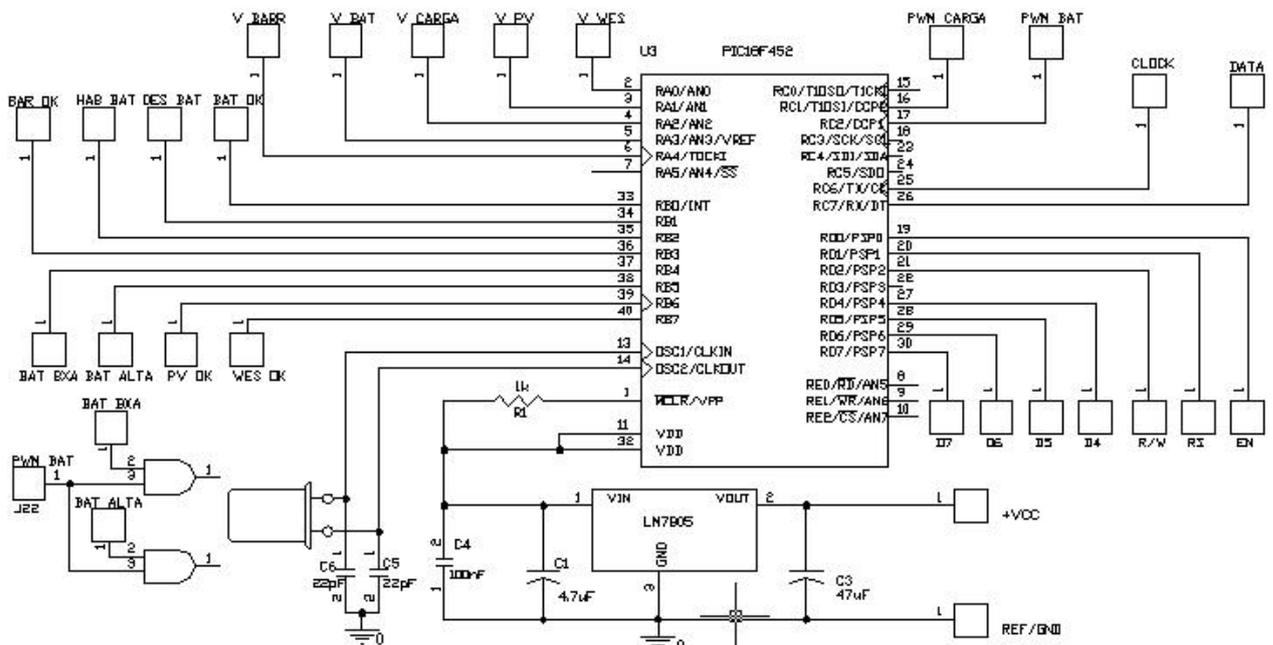
```
output_bit(ATIVIDADE, (++led_count & 0x01));    //pisca led para dizer q o
software esta rodando
delay_ms(50);                                  //espera um pouco para a proxima interacao
lcd_gotoxy(1,1);
printf(LCD_PUTC,"Vbat%3.2f c%lu",tensao_bat,pwm_bat);
lcd_gotoxy(1,2);
printf(LCD_PUTC,"Vcarga%3.2f c%li",tensao_bar,pwm_carga);
}    //final formal do while perpetuo
}    //final do main...fim de papo.
```

APÊNDICE II

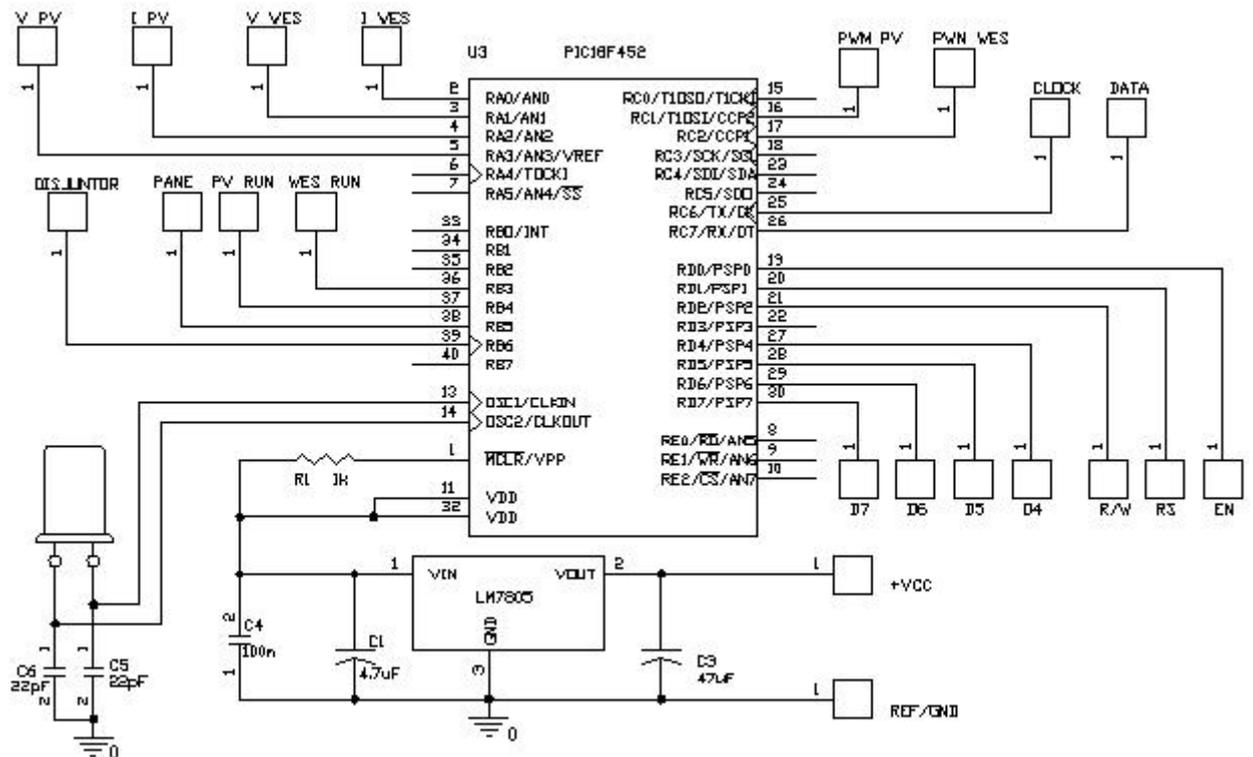
CONTROLES DAS BATERIAS, CARGA PRINCIPAL E FONTES

Neste apêndice está o diagrama de blocos do módulo de controle da carga e descarga das baterias e da tensão na carga principal.

1) Controles da carga e descarga das baterias e da regulação de tensão na carga principal.



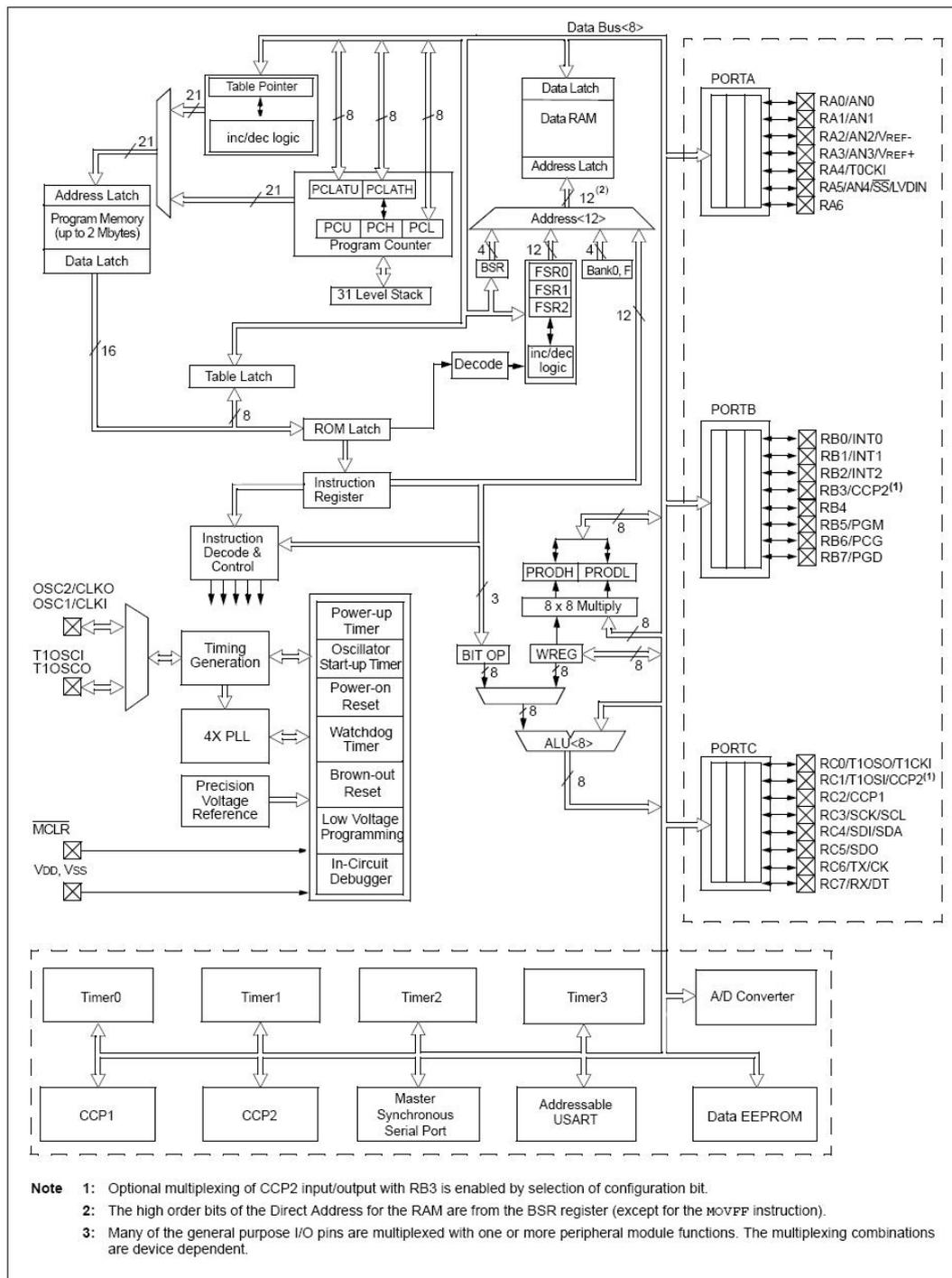
2) Diagrama de blocos do circuito do módulo de controle das fontes baseado no HCC de forma a obter individualmente a máxima potência de cada fonte.



ANEXO I

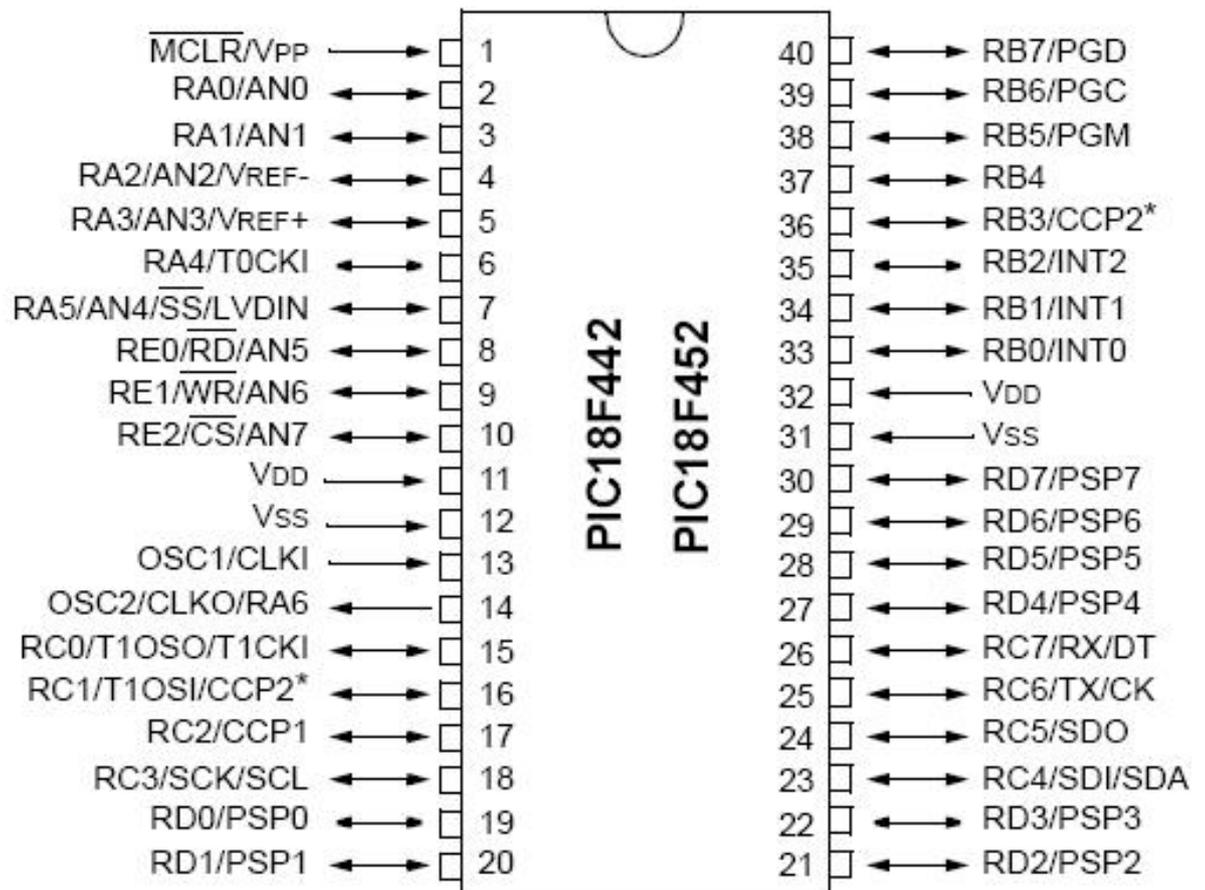
DIAGRAMA DE BLOCOS DO PIC18F452

Este Anexo mostra o diagrama de blocos do PIC18F452 (Microchip) usado para todo o controle do sistema integrado de energia.



ANEXO II

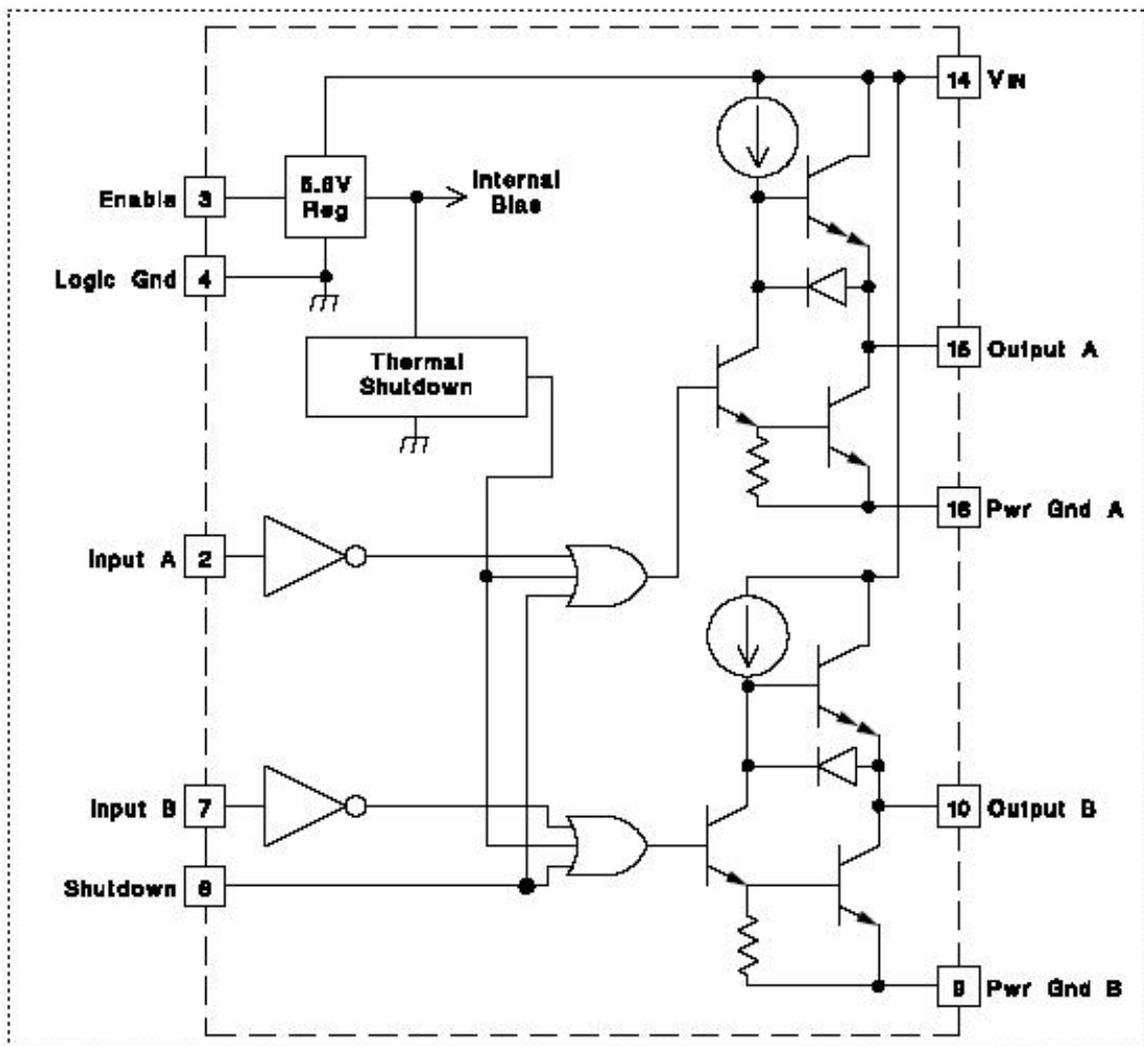
PINAGEM DO PIC 18F452



ANEXO III

“DRIVER” DE POTÊNCIA UC3708

Este Anexo o diagrama de blocos do “driver” de potência UC3708 usado para acionamento dos MOSFET de potência IRFZ42 usados tanto nas fontes de corrente como nas cargas.



ANEXO IV

TABELAS DE CARACTERÍSTICAS DO PIC18F452

Este Anexo lista as tabelas das principais características e pinagem do microcontrolador PIC18F452

Features	PIC18F452
Operating Frequency	DC - 40 MHz
Program Memory (Bytes)	32K
Program Memory (Instructions)	16384
Data Memory (Bytes)	1536
Data EEPROM Memory (Bytes)	256
Interrupt Sources	18
I/O Ports	Ports A, B, C, D, E
Timers	4
Capture/Compare/PWM Modules	2
Serial Communications	MSSP, Addressable USART
Parallel Communications	PSP
10-bit Analog-to-Digital Module	8 input channels
RESETS (and Delays)	POR, BOR, RESET Instruction, Stack Full, Stack Underflow (PWRT, OST)
Programmable Low Voltage Detect	Yes
Programmable Brown-out Reset	Yes
Instruction Set	75 Instructions
Packages	40-pin DIP 44-pin PLCC 44-pin TQFP

Tabela 4.1 – Principais Características do PIC 18F452

Pin Name	Pin Number		Pin Type	Buffer Type	Description
	DIP	SOIC			
RB0/INT0 RB0 INT0	21	21	I/O I	TTL ST	PORTB is a bi-directional I/O port. PORTB can be software programmed for internal weak pull-ups on all inputs. Digital I/O. External Interrupt 0.
RB1/INT1 RB1 INT1	22	22	I/O I	TTL ST	External Interrupt 1.
RB2/INT2 RB2 INT2	23	23	I/O I	TTL ST	Digital I/O. External Interrupt 2.
RB3/CCP2 RB3 CCP2	24	24	I/O I/O	TTL ST	Digital I/O. Capture2 input, Compare2 output, PWM2 output.
RB4	25	25	I/O	TTL	Digital I/O. Interrupt-on-change pin.
RB5/PGM RB5 PGM	26	26	I/O I/O	TTL ST	Digital I/O. Interrupt-on-change pin. Low Voltage ICSP programming enable pin.
RB6/PGC RB6 PGC	27	27	I/O I/O	TTL ST	Digital I/O. Interrupt-on-change pin. In-Circuit Debugger and ICSP programming clock pin.
RB7/PGD RB7 PGD	28	28	I/O I/O	TTL ST	Digital I/O. Interrupt-on-change pin. In-Circuit Debugger and ICSP programming data pin.

Legend: TTL = TTL compatible input

ST = Schmitt Trigger input with CMOS levels

O = Output

OD = Open Drain (no P diode to V_{DD})

CMOS = CMOS compatible input or output

I = Input

P = Power

Tabela 4.2(b) – Descrição dos pinos do PIC 18F452

Pin Name	Pin Number		Pin Type	Buffer Type	Description
	DIP	SOIC			
RC0/T1OSO/T1CKI	11	11			PORTC is a bi-directional I/O port.
RC0			I/O	ST	Digital I/O.
T1OSO			O	—	Timer1 oscillator output.
T1CKI			I	ST	Timer1/Timer3 external clock input.
RC1/T1OSI/CCP2	12	12			
RC1			I/O	ST	Digital I/O.
T1OSI			I	CMOS	Timer1 oscillator input.
CCP2			I/O	ST	Capture2 input, Compare2 output, PWM2 output.
RC2/CCP1	13	13			
RC2			I/O	ST	Digital I/O.
CCP1			I/O	ST	Capture1 input/Compare1 output/PWM1 output.
RC3/SCK/SCL	14	14			
RC3			I/O	ST	Digital I/O.
SCK			I/O	ST	Synchronous serial clock input/output for SPI mode.
SCL			I/O	ST	Synchronous serial clock input/output for I ² C mode
RC4/SDI/SDA	15	15			
RC4			I/O	ST	Digital I/O.
SDI			I	ST	SPI Data In.
SDA			I/O	ST	I ² C Data I/O.
RC5/SDO	16	16			
RC5			I/O	ST	Digital I/O.
SDO			O	—	SPI Data Out.
RC6/TX/CK	17	17			
RC6			I/O	ST	Digital I/O.
TX			O	—	USART Asynchronous Transmit.
CK			I/O	ST	USART Synchronous Clock (see related RX/DT).
RC7/RX/DT	18	18			
RC7			I/O	ST	Digital I/O.
RX			I	ST	USART Asynchronous Receive.
DT			I/O	ST	USART Synchronous Data (see related TX/CK).
Vss	8, 19	8, 19	P	—	Ground reference for logic and I/O pins.
VDD	20	20	P	—	Positive supply for logic and I/O pins.

Legend: TTL = TTL compatible input

ST = Schmitt Trigger input with CMOS levels

O = Output

OD = Open Drain (no P diode to VDD)

CMOS = CMOS compatible input or output

I = Input

P = Power

Tabela 4.2(c) – Descrição dos pinos do PIC 18F452

Pin Name	Pin Number			Pin Type	Buffer Type	Description
	DIP	PLCC	TQFP			
RB0/INT0 RB0 INT0	33	36	8	I/O I	TTL ST	PORTB is a bi-directional I/O port. PORTB can be software programmed for internal weak pull-ups on all inputs. Digital I/O. External Interrupt 0.
RB1/INT1 RB1 INT1	34	37	9	I/O I	TTL ST	External Interrupt 1.
RB2/INT2 RB2 INT2	35	38	10	I/O I	TTL ST	Digital I/O. External Interrupt 2.
RB3/CCP2 RB3 CCP2	36	39	11	I/O I/O	TTL ST	Digital I/O. Capture2 input, Compare2 output, PWM2 output.
RB4	37	41	14	I/O	TTL	Digital I/O. Interrupt-on-change pin.
RB5/PGM RB5 PGM	38	42	15	I/O I/O	TTL ST	Digital I/O. Interrupt-on-change pin. Low Voltage ICSP programming enable pin.
RB6/PGC RB6 PGC	39	43	16	I/O I/O	TTL ST	Digital I/O. Interrupt-on-change pin. In-Circuit Debugger and ICSP programming clock pin.
RB7/PGD RB7 PGD	40	44	17	I/O I/O	TTL ST	Digital I/O. Interrupt-on-change pin. In-Circuit Debugger and ICSP programming data pin.

Legend: TTL = TTL compatible input

ST = Schmitt Trigger input with CMOS levels

O = Output

OD = Open Drain (no P diode to V_{DD})

CMOS = CMOS compatible input or output

I = Input

P = Power

Tabela 4.2(e) – Descrição dos pinos do PIC 18F452

Pin Name	Pin Number		Pin Type	Buffer Type	Description
	DIP	SOIC			
RC0/T1OSO/T1CKI	11	11			PORTC is a bi-directional I/O port.
RC0			I/O	ST	Digital I/O.
T1OSO			O	—	Timer1 oscillator output.
T1CKI			I	ST	Timer1/Timer3 external clock input.
RC1/T1OSI/CCP2	12	12			
RC1			I/O	ST	Digital I/O.
T1OSI			I	CMOS	Timer1 oscillator input.
CCP2			I/O	ST	Capture2 input, Compare2 output, PWM2 output.
RC2/CCP1	13	13			
RC2			I/O	ST	Digital I/O.
CCP1			I/O	ST	Capture1 input/Compare1 output/PWM1 output.
RC3/SCK/SCL	14	14			
RC3			I/O	ST	Digital I/O.
SCK			I/O	ST	Synchronous serial clock input/output for SPI mode.
SCL			I/O	ST	Synchronous serial clock input/output for I ² C mode.
RC4/SDI/SDA	15	15			
RC4			I/O	ST	Digital I/O.
SDI			I	ST	SPI Data In.
SDA			I/O	ST	I ² C Data I/O.
RC5/SDO	16	16			
RC5			I/O	ST	Digital I/O.
SDO			O	—	SPI Data Out.
RC8/TX/CK	17	17			
RC8			I/O	ST	Digital I/O.
TX			O	—	USART Asynchronous Transmit.
CK			I/O	ST	USART Synchronous Clock (see related RX/DT).
RC7/RX/DT	18	18			
RC7			I/O	ST	Digital I/O.
RX			I	ST	USART Asynchronous Receive.
DT			I/O	ST	USART Synchronous Data (see related TX/CK).
VSS	8, 19	8, 19	P	—	Ground reference for logic and I/O pins.
VDD	20	20	P	—	Positive supply for logic and I/O pins.

Legend: TTL = TTL compatible input

ST = Schmitt Trigger input with CMOS levels

O = Output

OD = Open Drain (no P diode to VDD)

CMOS = CMOS compatible input or output

I = Input

P = Power

Tabela 4.2(f) – Descrição dos pinos do PIC 18F452

Pin Name	Pin Number			Pin Type	Buffer Type	Description
	DIP	PLCC	TQFP			
RD0/PSP0	19	21	38	I/O	ST TTL	PORTD is a bi-directional I/O port, or a Parallel Slave Port (PSP) for interfacing to a microprocessor port. These pins have TTL input buffers when PSP module is enabled. Digital I/O. Parallel Slave Port Data.
RD1/PSP1	20	22	39	I/O	ST TTL	
RD2/PSP2	21	23	40	I/O	ST TTL	
RD3/PSP3	22	24	41	I/O	ST TTL	
RD4/PSP4	27	30	2	I/O	ST TTL	
RD5/PSP5	28	31	3	I/O	ST TTL	
RD6/PSP6	29	32	4	I/O	ST TTL	
RD7/PSP7	30	33	5	I/O	ST TTL	
RE0/ $\overline{\text{RD}}$ /AN5 RE0 RD	8	9	25	I/O	ST TTL	PORTE is a bi-directional I/O port. Digital I/O. Read control for parallel slave port (see also $\overline{\text{WR}}$ and $\overline{\text{CS}}$ pins). Analog input 5.
AN5					Analog	
RE1/ $\overline{\text{WR}}$ /AN6 RE1 WR	9	10	26	I/O	ST TTL	
AN6					Analog	Digital I/O. Write control for parallel slave port (see $\overline{\text{CS}}$ and $\overline{\text{RD}}$ pins). Analog input 6.
RE2/ $\overline{\text{CS}}$ /AN7 RE2 CS	10	11	27	I/O	ST TTL	Digital I/O. Chip Select control for parallel slave port (see related $\overline{\text{RD}}$ and $\overline{\text{WR}}$). Analog input 7.
AN7					Analog	
Vss	12, 31	13, 34	6, 29	P	—	Ground reference for logic and I/O pins.
VDD	11, 32	12, 35	7, 28	P	—	Positive supply for logic and I/O pins.

Legend: TTL = TTL compatible input

ST = Schmitt Trigger input with CMOS levels

O = Output

OD = Open Drain (no P diode to VDD)

CMOS = CMOS compatible input or output

I = Input

P = Power

Tabela 4.2(g) – Descrição dos pinos do PIC 18F452