

**MONITORAMENTO DA DESCARGA DE BATERIA COM O USO DE  
MICROPROCESSADOR ARM**

RAFAEL LIMA PEIXOTO

PROJETO SUBMETIDO AO CORPO DOCENTE DO DEPARTAMENTO DE ENGENHARIA ELÉTRICA DA ESCOLA POLITÉCNICA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE ENGENHEIRO ELETRICISTA.

Aprovada por:

---

Prof. Luís Guilherme Barbosa Rolim, Dr.-Ing.

(Orientador)

---

Prof. Walter Issamu Suemitsu, Dr.Ing.

---

Eng. Isidoro Campos Raposo de Almeida Filho, M.Sc.

RIO DE JANEIRO, RJ - BRASIL  
FEVEREIRO DE 2012

*Aos meus familiares e amigos.*

## AGRADECIMENTOS

Agradeço primeiramente aos meus familiares: meus pais, minha avó, meus falecidos avôs, minha irmã, meus padrinhos, tios, tias e minha namorada por acreditarem em mim a todo o momento e sempre me apoiarem, e agradeço principalmente ao meu também falecido bisavô que pôde me proporcionar o estudo de qualidade que tive até o meu ingresso na UFRJ.

Ao meu orientador Luís Guilherme Barbosa Rolim por ter me ensinado o caminho e pela paciência. A todos os professores da graduação da engenharia elétrica da UFRJ por todo o conhecimento passado.

Ao Engenheiro Isidoro Campos Raposo de Almeida Filho (Recriar Tecnologias) pela grande ajuda. E a todos os funcionários da Recriar Tecnologias pelo apoio.

A todos os amigos que fiz durante o período de faculdade.

*“A alegria está na luta, na tentativa, no sofrimento envolvido. Não na vitória propriamente dita.”*

*Mahatma Gandhi*

## RESUMO

Este trabalho descreve o uso de uma metodologia baseada em cálculos de interpolação linear aplicados nas curvas de descarga de uma bateria para a estimação do tempo restante de sua energia. Esses cálculos são implementados no microprocessador ARM7 LPC2368 com o intuito de controlar a corrente de descarga e monitorar o tempo restante de uso da bateria. A descarga é controlada com o uso de um conversor eletrônico CC/CC PWM. O teste de conceito foi verificado em uma bancada experimental, na qual foram realizados testes experimentais, como descargas para diversas faixas de corrente constante. Os resultados obtidos foram satisfatórios, validando o estudo apresentado.

**PALAVRAS CHAVE:** Descarga de bateria, curvas de descarga, microprocessador ARM7 LPC2368, interpolação linear, conversor CC/CC.

## ABSTRACT

This work describes the use of a methodology based on applied calculations of linear interpolation in the curves of discharge of a battery for the estimation of the remaining time of its energy. These calculations are implemented in microprocessor ARM7 LPC2368 with the intention to control the discharge current and to monitor the remaining time of use of the battery. The discharge is controlled with the use of a DC/DC PWM electronic converter. The proof of concept was verified in a test rig, in which experimental tests were carried out, such as discharges for several constant current ranges. The obtained results were satisfactory, thus validating the study presented.

**KEYWORDS:** Battery discharge, discharge curves, microcontroller ARM7 LPC2368, linear interpolation, DC/DC converter.

## LISTA DE FIGURAS

Figura 2.1: Fotos de modelos de diferentes baterias encontradas no mercado. (a) Baterias comercial sem manutenção líquida, (b) Baterias comercial estacionárias líquidas, (c) Bateria comercial VLRA de gel, (d) Bateria comercial VRLA de AGM. ....	5
Figura 2.2: Curvas de descarga de uma bateria VRLA de 7,2 Ah. Fonte: [3]. ....	8
Figura 2.3: Curvas de descarga para diferentes valores de correntes constantes de uma bateria estacionária de 70 Ah. Fonte:[4]. ....	8
Figura 3.1: Representação de uma curva de descarga com corrente constante demonstrando o cálculo do tempo $dt$ para uma tensão $V_{bat}$ . ....	14
Figura 3.2: Gráfico de uma curva com a sua interpolação por quatro pontos e por dois pontos. ....	15
Figura 3.3: Exemplos representando curvas de descarga de uma bateria. ....	17
Figura 3.4: Gráfico apresentando pontos da curva $I \times dt$ para uma tensão fixa $V_{bat}$ . ....	18
Figura 3.5: Gráfico representando o cálculo do tempo $dt^*_1$ usando a interpolação linear. ....	19
Figura 3.6: Diagrama de bloco do microcontrolador LCP2368. Fonte: [8] ....	21
Figura 3.7: (a) Sinais de comparação para gerar o sinal PWM. (b) Sinal PWM resultante comparado com o sinal modulador. Fonte: [9]. ....	23
Figura 3.8: Onda “dente de serra” comparada com um sinal de referência e sinal PWM gera. ....	24
Figura 3.9: Circuito de um conversor <i>Buck</i> com filtro na entrada e na saída. ....	28
Figura 4.1: Diagrama de blocos da bancada de teste. ....	30
Figura 4.2: Circuito elétrico da bancada experimental. ....	32
Figura 4.3: Algoritmo implementado no ARM. ....	33
Figura 4.4: Controle do PWM: (a) Controle do sinal PWM no ARM, (b) Sinal de referência, “dente de serra” e sinal PWM gerado. ....	34
Figura 4.5: Diagrama do algoritmo implementado no ARM para cálculo do tempo e IHM. ....	35
Figura 4.6: Tela do osciloscópio durante a descarga com corrente constante de 20 A. ....	36
Figura 4.7: Curvas reais de descarga para 15, 20, 25, 27 e 30 A da bateria Optima D51. ....	37
Figura 4.8: Curvas reais de descarga com correntes 20 A, 25 A e 30 A e interpolada de 25 A. ....	38
Figura 4.9: Curvas de 27 A real e interpoladas. ....	39
Figura 1.1: Diagrama de bloco do microprocessador LCP2368. Fonte: [8] ....	44
Figura 1.2: (a) Sinais de comparação para gerar o sinal PWM. (b) Sinal PWM resultante comparado com o sinal modulador. Fonte:[9]. ....	49
Figura 1.3: Exemplo de diferentes sinais PWM gerados no ARM7 LPC2368. Fonte: [8] ....	50
Figura 1.4: Exemplo de ciclo de Timer, onde $PR = 2$ , $MRx = 6$ e a interrupção e o reset estão ativos no encontro com o comparador. ....	54

**LISTA DE TABELAS**

Tabela 3.1: Variação de $dt$ por $V_{\text{bat}}$ e $I_{\text{cte}}$ .....	18
Tabela 3.2: Pinagem do display GLCD .....	26
Tabela 1.1: Descrição dos registradores das portas GPIO com uso de acesso rápido.....	46
Tabela 1.2: Possíveis valores para os bits do registrador de divisão do <i>clock</i> dos periféricos. ....	48
Tabela 1.3: Valores usados nos comparadores do PWM, no exemplo da Figura 1.3.....	50
Tabela 1.4: Mapa de bits do registrador UnLSR .....	57
Tabela 1.5: Pinagem do display GLCD .....	58



## SUMÁRIO

1	INTRODUÇÃO .....	1
1.1	OBJETIVOS .....	1
1.2	METODOLOGIA .....	2
1.3	ORGANIZAÇÃO .....	2
2	CONCEITOS BÁSICOS .....	3
2.1	TECNOLOGIAS PARA ARMAZENAMENTO DE ENERGIA .....	3
2.1.1	TIPOS DE BATERIAS .....	3
2.1.2	DIFERENCIAIS DE APLICAÇÃO .....	6
2.1.3	CURVAS DE DESCARGA .....	7
2.2	MÉTODOS DE ESTIMAÇÃO DO TEMPO DE CARGA .....	9
2.2.1	MÉTODO LINEAR .....	10
2.2.2	METODOLOGIA “COULOMB COUNTING” .....	11
2.2.3	LEI DE PEUKERT .....	11
2.3	METODOLOGIA ALTERNATIVA PARA ESTIMAÇÃO DE DESCARGA .....	13
3	METODOLOGIA PARA ESTIMAR O TEMPO RESTANTE DE CARGA NA BATERIA .....	14
3.1	METODOLOGIA DA INTERPOLAÇÃO LINEAR .....	15
3.2	APLICAÇÃO DA INTERPOLAÇÃO NO CÁLCULO DO TEMPO RESTANTE DA BATERIA .....	17
3.3	O ARM7 LPC 2368 .....	19
3.3.1	CONFIGURAÇÃO DOS PERIFÉRICOS .....	20
3.3.2	INTERFACE HOMEM MÁQUINA - IHM .....	26
3.4	BANCADA .....	27

3.4.1 BANCADA EXPERIMENTAL PARA DESCARGA DE BATERIA COM CORRENTE CONSTANTE .....	28
4 BANCADA E RESULTADOS EXPERIMENTAIS.....	30
4.1 A BANCADA EXPERIMENTAL .....	30
4.2 CONTROLE .....	32
4.3 RESULTADOS EXPERIMENTAIS .....	36
5 CONCLUSÕES E TRABALHOS FUTUROS.....	40
5.1 CONCLUSÕES .....	40
5.2 TRABALHOS FUTUROS .....	41
6 BIBLIOGRAFIA .....	42
Apêndice.....	43
1 O ARM7 LPC2368 .....	43
1.1 CONFIGURAÇÃO DOS PERIFÉRICOS.....	43
1.1.1 ENTRADAS E SAIDAS DIGITAIS– GPIO .....	45
1.1.2 CONVERSOR ANALÓGICO-DIGITAL - ADC.....	47
1.1.3 MODULAÇÃO POR LARGURA DE PULSO - PWM.....	49
1.1.4 CONTADORES INTERNOS .....	53
1.1.5 COMUNICAÇÃO UART.....	55
1.2 INTERFACE HOMEM MÁQUINA - IHM.....	58

# 1 INTRODUÇÃO

Este trabalho aborda uma metodologia de monitoramento de descarga de bateria, com a estimação do estado de carga (SoC – *State of Charge*), utilizando um microprocessador com arquitetura ARM (*Advanced Risc Machine*).

## 1.1 OBJETIVOS

O objetivo principal deste trabalho é avaliar uma metodologia para estimar o estado de carga de uma bateria além de controlar a sua corrente de descarga. Este controle possibilita a obtenção de curvas de descarga em baterias com corrente constante.

Além disso, este trabalho tem o intuito de avaliar o funcionamento do microprocessador ARM, para uso em tempo real, eventos onde o tempo de resposta do microprocessador tem influência no resultado do experimento, e finalmente estudar a metodologia de interpolação linear e realizar um teste em uma bancada experimental para comprovar o funcionamento do conceito estudado.

A motivação para alcançar esses objetivos, é a grande aplicabilidade desse estudo, que pode ser implementado, por exemplo, em um veículo elétrico, onde a determinação do tempo restante de carga na bateria é essencial para evitar imprevistos durante uma viagem, ou em um sistema de fonte intermitente que pode haver falta de energia por um longo período de tempo.

É possível encontrar outros estudos para estimação do SoC em minutos, entretanto são usados em sistemas de pequeno porte, que utilizam baixas correntes e geralmente esses estudos são feitos para baterias de íons de lítio. Nesse trabalho o estudo feito é para bateria chumbo-ácido que são descarregadas com correntes mais elevadas, normalmente maiores que 10 A.

## 1.2 METODOLOGIA

O controle do tempo de descarga de bateria foi realizado utilizando-se um conversor CC/CC controlando a corrente de saída da bateria. Para monitorar o tempo de carga na bateria foi usado o método da interpolação linear aplicado a curvas de tensão x tempo (corrente constante) de uma bateria específica. Foi utilizado o microprocessador ARM7 LPC2368 para controlar a corrente constante de descarga independentemente da variação da tensão na bateria.

Para analisar a metodologia em estudo foram feitos testes em uma bancada experimental composta por uma bateria, uma carga fixa e um conversor CC/CC controlado pelo ARM a fim de manter a corrente de descarga desejada, obtendo-se diversas curvas tensão x tempo para utilização na interpolação de outras curvas as quais são comparadas com as curvas experimentais.

## 1.3 ORGANIZAÇÃO

No capítulo 2 são apresentados os conceitos básicos usados no estudo proposto. Os métodos para medição do tempo de energia restante da bateria são abordados, além de uma apresentação dos diferentes tipos de baterias, e curvas típicas de descarga com corrente constante.

O capítulo 3 apresenta a metodologia a ser abordada visando a estimação do tempo de energia restante, o microprocessador usado e o controle para manter a corrente de descarga da bateria constante.

No capítulo 4 são apresentados a bancada experimental usada para testes de descarga da bateria, o algoritmo implementado no microprocessador e sua configuração, além dos resultados alcançados.

O capítulo 5 apresenta a conclusão final do trabalho e as propostas de trabalho futuro.

## 2 CONCEITOS BÁSICOS

### 2.1 TECNOLOGIAS PARA ARMAZENAMENTO DE ENERGIA

Atualmente, tecnologias para armazenamento de energia têm ganhado muito espaço nas áreas de pesquisa a nível mundial. Isso ocorre não apenas pelo desenvolvimento das tecnologias tradicionais e as novas tecnologias, mas também pela necessidade de se reduzir os problemas de geração intermitente de algumas fontes de energia, como a solar e a eólica por exemplo. Outro fator importante é o benefício, em termos econômicos, para o balanço da geração e demanda, havendo a possibilidade da transferência da energia armazenada para a rede elétrica nos horários de pico de consumo.

As tecnologias tradicionais de armazenamento de energia, como as baterias, têm tido uma evolução constante, como o aumento da densidade de energia, possibilitando alavancar a sua utilização em massa, em dispositivos elétricos e eletrônicos móveis e portáteis. Com isso, novos tipos de baterias têm surgido no mercado.

#### 2.1.1 TIPOS DE BATERIAS

Segundo [1] as baterias podem ser classificadas de acordo com vários critérios, como os materiais ativos usados na sua construção, o tipo de utilização, ou de acordo com suas características internas. Na classificação de acordo com os materiais ativos, as baterias podem ser divididas em chumbo-ácido, alcalinas, íons de lítio, níquel-cádmio, hidreto metálico de níquel, níquel-zinco, entre outras.

Neste trabalho, são abordadas as baterias de chumbo-ácido tendo em vista que atualmente é o tipo mais difundido no mercado pelo seu custo e benefício. Como dito em [1] a grande inserção de baterias chumbo-ácido no mercado começou no início do século XX, com a chegada da indústria automotiva, que incentivou o crescimento na área de pesquisas e aplicações.

Tais baterias são normalmente constituídas por células eletroquímicas ligadas em série ou paralelo. Cada célula é formada por um anodo (placa de chumbo) e um catodo (placa de dióxido de chumbo) imersos em um eletrólito de ácido sulfúrico, onde ocorre a reação química. Durante o processo de descarga, o dióxido de chumbo ( $PbO_2$ ), o chumbo ( $Pb$ ) e o ácido sulfúrico ( $H_2SO_4$ ) reagem formando sulfato de chumbo ( $PbSO_4$ ) e água ( $H_2O$ ), e no processo de carga a reação é revertida. Como mostrado na equação 2.1 abaixo.



Para se determinar a energia armazenada na bateria, é importante conhecer a sua tensão (V) e a sua capacidade de descarga de corrente no tempo (Ah). Com esses dados é possível saber a quantidade de energia armazenada na bateria, a partir da equação 2.2.

$$Energia(Wh) = Tensão(V) \times Capacidade(Ah) \quad (2.2)$$

Segundo [1], as baterias são também classificadas de acordo com sua resistência interna, podendo ser normal, baixa ou muito baixa. Esses fatores influenciam na capacidade de corrente de descarga, e conseqüentemente, na energia útil de uma bateria, como foi visto na equação 2.2. Com uma resistência interna mais alta, a bateria tem uma capacidade menor de fornecer corrente do que uma bateria com uma resistência interna mais baixa.

As baterias de chumbo-ácido ainda podem ser classificadas de acordo com suas características construtivas. Elas podem ser separadas em baterias ventiladas, baterias seladas SLA (*Sealed Lead Acid*), Selada VRLA (*Valve Regulated Lead Acid*), Seladas VRLA de Gel ou AGM (*Absorbed Glass Mat*), baterias sem ou de baixa manutenção, baterias tubulares, bipolares, baterias empastadas, entre outras.

Como dito em [2], nas baterias ventiladas ocorre o livre acesso ao eletrólito, isso permite controlá-lo, mas ao mesmo tempo cria uma atmosfera ácida em torno das baterias desse tipo. Já as baterias seladas foram desenvolvidas para suprir equipamentos que não toleram ambientes com a presença de ácido sulfúrico, seja devido a vazamentos ou simplesmente devido à geração de gases que ocorre normalmente em baterias de chumbo-ácido. As baterias seladas também são conhecidas como SLA ou VRLA, dispondo de um mecanismo de recombinação de gases. O eletrólito fica imobilizado, seja na forma de gel nas

baterias conhecidas como “bateria selada de gel”, ou absorvido em uma manta de fibra de vidro nas baterias AGM.

Diferentemente das baterias VLRA, as baterias sem ou de baixa manutenção, são baterias onde o eletrólito se encontra na forma líquida e não imobilizado. Entretanto nas automotivas conhecidas como sem manutenção, não existe o acesso ao eletrólito como ocorre nas baterias ventiladas, tornando-as intermediárias entre as seladas VLRA e as ventiladas.

Como pode ser visto na Figura 2.1 os diferentes modelos comerciais de baterias apresentam um exterior parecido, mesmo tendo suas formas construtivas diferentes.



(a)



(b)



(c)



(d)

Figura 2.1: Fotos de modelos de diferentes baterias encontradas no mercado. (a) Baterias comercial sem manutenção líquida, (b) Baterias comercial estacionárias líquidas, (c) Bateria comercial VLRA de gel, (d) Bateria comercial VRLA de AGM.

Outra grande diferença na forma construtiva das baterias de chumbo-ácido está na maneira como são feitas as suas placas, podendo ser tubulares ou empastadas, bipolares ou convencionais.

Segundo [2], no caso das baterias tubulares o material ativo das placas positivas fica em tubos porosos feitos de um material resistente ao ácido sulfúrico, o que aumenta significativamente o tempo de vida útil das baterias, cerca de 5 a 10 vezes, se comparadas com as empastadas, onde grades retangulares de uma liga de chumbo recebem no seu processo de produção uma mistura de sulfatos básicos e óxidos de chumbo.

A diferença entre as bipolares e as convencionais, está no tipo de placa utilizada. Nas bipolares as placas são formadas de maneira que um lado da placa é o eletrodo positivo de uma célula e o outro lado é o eletrodo negativo da próxima célula, reduzindo assim a resistência interna da bateria e o peso, pois este procedimento elimina partes do barramento que existe em baterias convencionais para fazer a conexão entre as diferentes células. Segundo [2], há um aumento na densidade de energia da bateria de 35 Wh/Kg para até 65-70 Wh/Kg.

Além das classificações segundo a forma construtiva, as baterias de chumbo-ácido também são separadas de acordo com sua utilização, podendo ser estacionárias, tracionárias, de arranque (automotivas) ou para usos especiais.

### 2.1.2 DIFERENCIAIS DE APLICAÇÃO

Como foram apresentadas anteriormente, as baterias de chumbo-ácido podem ser subdivididas em várias categorias, variando seu tipo construtivo. Por conseguinte, variam também suas propriedades de armazenamento de carga, de capacidade de corrente de descarga e sua vida útil. Com essa variedade nos tipos construtivos e nas suas propriedades, resulta certa variedade nas suas aplicações, podendo cada uma ter um tipo de uso específico.

Por exemplo, nas baterias estacionárias, também conhecidas como de ciclo profundo, as placas do catodo e do anodo são mais grossas do que nas baterias de arranque (automotivas), pois as baterias estacionárias operam mais na região de flutuação, apenas acontecendo cargas e descargas elevadas esporadicamente. Por isso elas são mais recomendadas para o uso em *no breaks*, fontes alternativas de energia, etc., pois suportam um longo período de carga e descarga.



As baterias de arranque (automotivas) têm suas placas mais finas e maior superfície por elemento, pois são feitas para fornecerem uma grande corrente de descarga em pouco tempo, como é preciso para o acionamento de um motor de arranque, e depois não é mais necessário o fornecimento grande de corrente. Essas baterias não têm um bom funcionamento quando usadas com uma corrente de descarga baixa por um longo tempo. Esse tipo de bateria é mais usado para o acionamento de motores, como por exemplo, em um carro.

Com características mistas entre as baterias estacionárias e as de arranque, estão as baterias tracionárias, as quais suportam longos períodos de descarga com uma grande corrente, essas baterias são as mais usadas em empilhadeiras elétricas e carros de golf.

Também existem baterias especiais, que têm suas características construtivas de acordo com o seu uso, como é o caso das baterias de submarinos.

Pode ser visto que as baterias de chumbo-ácido têm grande diversidade em seu uso e estão presentes em muitas aplicações, seja para armazenar a energia gerada em uma fonte de geração intermitente ou para dar partida no motor de um carro. Com essa diversidade, seu comportamento durante a descarga varia para diferentes modelos de bateria, como pode ser notado nas curvas de descarga (entende-se por curva de descarga a curva tensão x tempo para uma determinada corrente de descarga constante).

### 2.1.3 CURVAS DE DESCARGA

Curvas de descarga de baterias são funções não lineares de modelagem matemática complexa. Esses gráficos apresentam a variação da tensão da bateria em função do tempo, para uma determinada corrente constante de descarga.

Com a variação dessa corrente, são obtidas diferentes curvas de descarga. É de se esperar que quanto maior a corrente drenada da bateria, maior é o decremento de sua tensão, diminuindo o tempo restante de fornecimento de energia. Como é mostrado na Figura 2.2 e na Figura 2.3, mesmo para baterias com diferentes tipos construtivos, suas curvas de descarga apresentam um mesmo comportamento geral, i.e. uma parte mais linear e outra fortemente não linear.

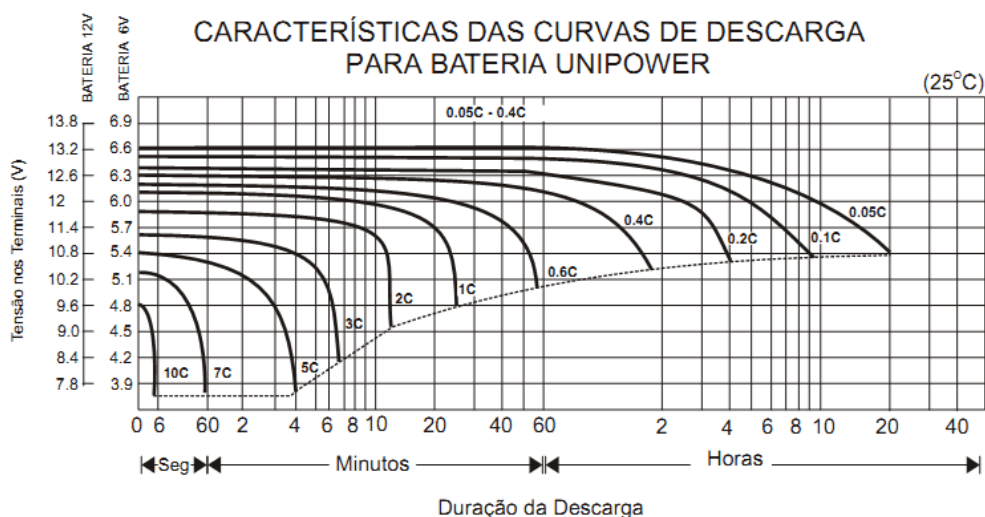


Figura 2.2: Curvas de descarga de uma bateria VRLA de 7,2 Ah. Fonte: [3].

Nos gráficos apresentados na Figura 2.2, o “C” indica a capacidade nominal da bateria em 20 horas. Ou seja, para saber o valor da corrente basta substituir o “C” pelo valor da capacidade da bateria que nesse caso é de 7,2 Ah. Por exemplo, na curva de 0,05C a bateria está sendo descarregada com uma corrente constante de 0,35 A. As curvas da Figura 2.2 mostram um comportamento aparentemente mais linear no começo da descarga devido à escala usada para o tempo. Na Figura 2.3 a seguir são apresentadas curvas de descarga com um maior detalhamento.

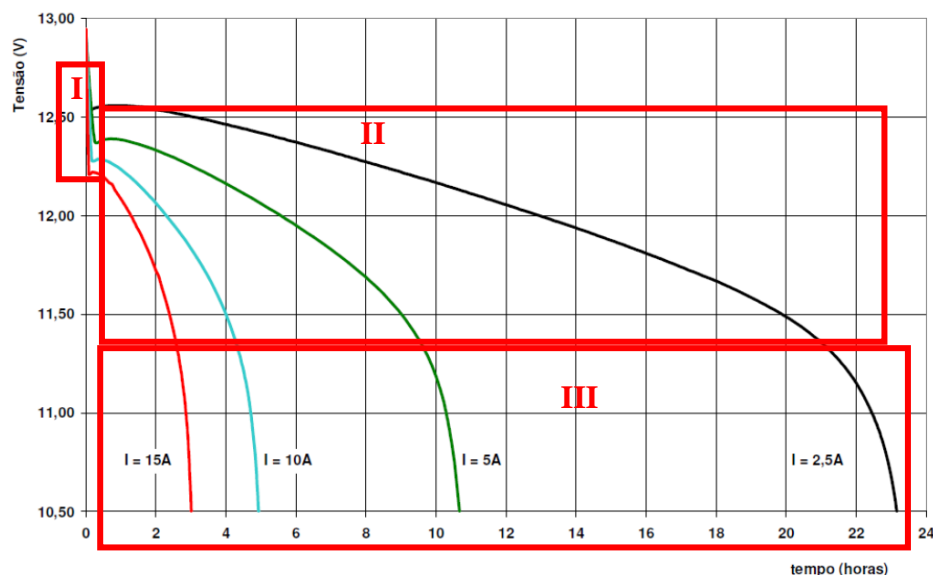


Figura 2.3: Curvas de descarga para diferentes valores de correntes constantes de uma bateria estacionária de 70 Ah. Fonte:[4].

Na Figura 2.3, são apresentadas curvas de descarga de uma bateria estacionária de 70 Ah. São apresentadas curvas de descarga para correntes de 2,5 A, 5 A, 10 A e 15 A. Nessas curvas é observável a redução do tempo total de descarga de acordo com o aumento da corrente, por exemplo, a descarga com a corrente de 2,5 A tem uma duração total de 23 h e a descarga à 15 A demora apenas 3 h. É também notado que o comportamento das baterias durante sua descarga pode ser dividido três fases: início (I), meio (II) e fim (III). Esse comportamento é destacado pelos retângulos vermelhos.

O momento inicial da descarga é caracterizado como o intervalo de tempo imediatamente após a carga ser ligada, quando ocorre uma rápida queda da tensão devido à resistência interna da bateria, isso não ocorreria se a bateria fosse ideal, sem resistência interna. Em seguida a bateria passa para o momento II, onde ela tem uma queda de tensão quase linear. Nesse momento a bateria sofre uma lenta queda de tensão, ou seja, a bateria tem uma pequena redução da tensão durante um longo período de tempo. O final é o momento onde a bateria está quase totalmente descarregada, neste momento sua tensão tem uma redução acentuada em um curto espaço de tempo. Na Figura 2.2 é notável que na curva de 2,5 A, a parte final (III) da bateria tem uma queda de tensão de 1 V em apenas 3 h, se comparado com o meio (II) da curva, onde ocorre uma queda de também 1 V em 20 h, a queda de tensão no final é muito mais rápida.

Devido a esse comportamento não linear durante todo o processo de descarga, a modelagem matemática da bateria para a estimação do seu estado de carga não é trivial, sendo analisada a seguir.

## 2.2 MÉTODOS DE ESTIMAÇÃO DO TEMPO DE CARGA

Um dos grandes problemas para a estimação do tempo de descarga de baterias é fazer a modelagem dos efeitos não-lineares que acontecem durante a descarga da bateria. Segundo [5] alguns dos efeitos não-lineares que ocorrem durante a descarga da bateria são os “efeitos de recuperação”, “nível de *cutoff*” e “efeito da taxa de capacidade”. Ressalta-se que tais efeitos ocorrem em todos os tipos de baterias. No entanto, dependendo do tipo de bateria, estes efeitos têm maiores ou menores impactos na sua capacidade.

O efeito de recuperação da bateria é a reorganização dos elétrons uniformemente no eletrólito durante o período em que a corrente de descarga é reduzida, esse período também é conhecido como período de relaxação. Com essa reorganização dos elétrons a capacidade efetiva da bateria é aumentada, antes da bateria chegar ao nível de *cutoff*, quando a bateria está totalmente descarregada. Para ocorrer o efeito de recuperação não é necessário que a carga seja desligada, basta que a corrente de descarga seja reduzida significativamente por um determinado intervalo de tempo.

Já o nível de *cutoff* segundo [5] pode ser definido como o valor limite de carga (capacidade) em que a bateria consegue fornecer uma tensão constante. Quando a bateria atinge o valor de *cutoff*, ela não está descarregada por completo, mas está incapaz de fornecer carga ao sistema por não ocorrerem mais as reações eletroquímicas.

A taxa de capacidade de uma bateria depende de como está sua corrente de descarga e a sua capacidade atual. Como dito em [5] a capacidade da bateria é reduzida para uma alta corrente de descarga, pois não há tempo suficiente para a reorganização dos elétrons no eletrólito, não ocorrendo o efeito de recuperação. Para aumentar a capacidade da bateria, é melhor variar a corrente de descarga, passando por períodos com uma corrente mais baixa ou até mesmo sem corrente, dando tempo assim para os elétrons se reorganizarem no eletrólito. Sendo que essa capacidade nunca será maior, nem igual à capacidade inicial.

Devido aos efeitos apresentados acima, a modelagem matemática da descarga de uma bateria torna-se complexa, o que dificulta o cálculo do tempo restante de energia da bateria. A seguir serão apresentados métodos para se fazer essa estimativa.

### 2.2.1 MÉTODO LINEAR

Esse método é o mais simples para determinação do tempo de carga de uma bateria. Ele é considerado o modelo analítico mais simples pelo fato de considerar a bateria como um dispositivo linear de corrente, fazendo-se valer a equação 2.3.

$$C = C_i - I \times t_d \quad (2.3)$$

Onde  $C$  é a capacidade de carga atual,  $C_i$  a capacidade de carga inicial,  $I$  é a corrente de descarga e  $t_d$  o tempo de descarga. Mas como dito anteriormente as baterias não se comportam de forma linear, e isso torna esse método muito impreciso, e ainda seria necessário manter a corrente de descarga constante para realizar a aproximação.

### 2.2.2 METODOLOGIA “COULOMB COUNTING”

A metodologia conhecida como “Coulomb Counting” calcula o estado de carga de uma bateria usando a integral no tempo da corrente de descarga  $i_{bat}$  segundo a equação 2.3 apresenta a seguir.

$$SoC(t) = 100 \times \left( \frac{Q_c - \int_0^t i_{bat}(\tau) d\tau}{Q_c} \right) \quad (2.3)$$

Onde  $SoC$  é o estado de carga atual,  $Q_c$  representa o estado de carga inicial da bateria. Em longo prazo este método sofre devido à falta de um ponto de referência: portanto, o SoC deve ser re-calibrados, redefinindo o SoC para 100% quando um carregador determina que a bateria esteja completamente carregada.

### 2.2.3 LEI DE PEUKERT

Pela Lei de Peukert a descarga de uma bateria é influenciada por vários fatores, como o valor da corrente elétrica, a duração da corrente, a tensão nos terminais e a temperatura. Segundo [6] a equação de Peukert descreve um processo de descarga de bateria chumbo-ácido que relaciona a corrente  $I$ , corrente total de descarga da bateria em certa temperatura com o tempo  $t_{dc}$ , tempo total de descarga da bateria.

$$I^n \times t_{dc} = k \quad (2.4)$$

Onde  $n$  e  $k$  são constantes determinadas, realizando duas descargas completas com valores distintos de correntes de descarga. A temperatura em que são realizadas essas descargas para a determinação de  $n$  e  $k$ , é chamada de temperatura de referência,  $T_r$ .

Com base na equação de Peukert, equação 2.4, pode-se determinar a equação da capacidade da bateria em uma determinada temperatura de referência, como mostrado a seguir:

$$C_{NTR}(I_N) = kI_N^{(1-n)} \quad (2.5)$$

Sendo  $C_{NTR}$ , a capacidade nominal da bateria na temperatura de referência,  $T_r$ , e  $I_N$  é a corrente nominal em que a descarga foi efetuada. A equação que modela a capacidade da bateria em relação à temperatura é dada pela equação 2.6 a seguir como descrita em [6].

$$C_{RTR}(I) = \frac{C_{RT}}{1 + \delta(T - T_R)} \quad (2.6)$$

Sendo  $C_{RT}$  a capacidade da bateria relativa na temperatura  $T$ ,  $C_{RTR}$  é capacidade relativa da bateria na temperatura de referência e  $\delta$  é a variação da capacidade por °C em porcentagem. Com a equação 2.6 fica possível obter a equação que descreve a capacidade relativa da bateria para qualquer temperatura  $T$ .

$$C_{RT} = C_{RTR}(I) \times 1 + \delta(T - T_R) \quad (2.7)$$

E para obter a capacidade nominal da bateria em qualquer temperatura  $T$ , basta substituir  $I$  por  $I_N$  na equação 2.7. Com a equação que define  $C_{RT}$  fica possível obter o estado de carga da bateria na condição de descarga, pela equação 2.8 abaixo, que relaciona  $Q_{AT}(t)$ , a carga remanescente da bateria, em Ah, no instante  $t$ , com a capacidade relativa da bateria [6].

$$EC(t) = \frac{Q_{AT}(t)}{C_{RT}(I)} \quad (2.8)$$

Para se determinar o estado de carga atual em cada instante pode-se usar a relação

$$EC(t) = EC(t_{-1}) - \frac{Q_{out}}{C_{RT}(I)} \quad (2.9)$$

Onde  $Q_{out}$  é a carga fornecida pela bateria em um intervalo de tempo  $\Delta t$  em meio a duas medições de corrente de descarga, e  $EC(t-1)$  é o estado de carga o instante anterior. A autonomia remanescente da bateria pode ser calculada a partir do valor de  $EC(t)$ , usando a expressão 2.10:

$$Q_{AT}(t) = EC(t) \times C_{RT}(I) \quad (2.10)$$

### 2.3 METODOLOGIA ALTERNATIVA PARA ESTIMAÇÃO DE DESCARGA

A metodologia para estimação do tempo de descarga que será utilizada neste projeto baseia-se no cálculo por interpolação linear, a partir de valores tabelados para tensão, corrente e os tempos de descarga correspondentes. A partir das curvas de descarga, da medição de corrente e tensão atual na bateria, torna-se possível calcular uma aproximação do tempo restante de uso da bateria.

O grande diferencial dessa metodologia em relação às apresentadas é o fato de que usando as curvas de descarga, a resposta é o tempo em minutos, e não uma proporção do SoC inicial em porcentagem, além disso, não precisa de um estado inicial de referência como ocorre nas outras metodologias.

### 3 METODOLOGIA PARA ESTIMAR O TEMPO RESTANTE DE CARGA NA BATERIA

As curvas de descarga apresentam a variação da tensão ao longo do tempo para uma determinada corrente constante. A partir da medição da tensão e da curva da corrente de descarga, é possível estimar o tempo restante de uso de uma bateria ( $dt$ ).

Esse intervalo de tempo,  $dt$ , é a diferença entre o tempo total de descarga da bateria e o tempo decorrido, com uma determinada corrente, conforme ilustrado na Figura 3.1.

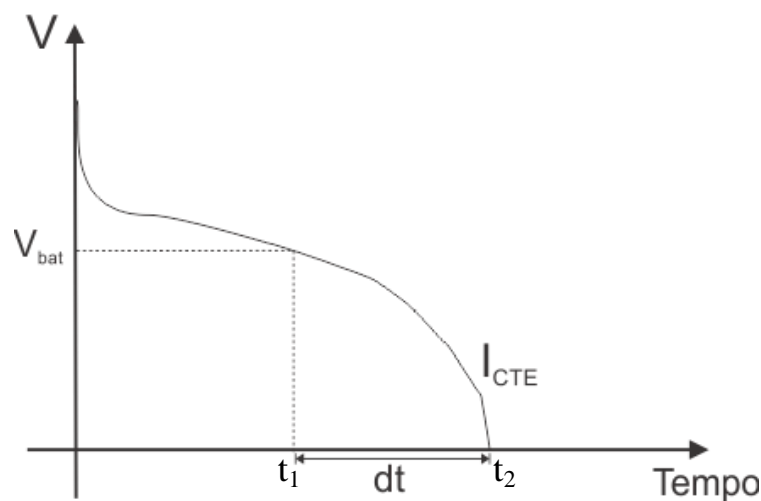


Figura 3.1: Representação de uma curva de descarga com corrente constante demonstrando o cálculo do tempo  $dt$  para uma tensão  $V_{bat}$ .

O gráfico na Figura 3.1, ilustra o formato de uma curva de descarga. Sabendo-se o valor da tensão atual na bateria ( $V_{bat}$ ) e o valor da corrente de descarga ( $I_{cte}$ ) é possível descobrir o valor do tempo  $t_1$ , tempo decorrido para a tensão chegar a  $V_{bat}$ , e do tempo  $t_2$ , tempo que a bateria leva para se descarregar completamente com a corrente  $I_{cte}$ . Conhecendo os tempos  $t_1$  e  $t_2$ , o cálculo de  $dt$ , tempo restante de carga na bateria, torna-se fácil, subtraindo  $t_2$  de  $t_1$ .

Entretanto para obter o tempo restante de carga na bateria durante uma operação normal, onde a corrente não se mantém constante, ou seja, varia com o tempo, faz-se necessário alternar as curvas de descarga para diferentes valores de corrente. Para isto é preciso ter os valores de  $V_{bat}$  e  $dt$  tabelados para distintas correntes de descarga. Para realizar o cálculo de  $dt$  a partir de valores medidos de tensão e corrente que não estejam tabelados, será utilizado o método matemático de interpolação linear, conforme descrito a seguir.



### 3.1 METODOLOGIA DA INTERPOLAÇÃO LINEAR

Segundo [7], interpolação é um método matemático onde é possível obter a aproximação de uma função a partir de um conjunto de pontos dessa função previamente conhecidos. Com essa metodologia torna-se possível construir uma função aproximada para dar continuidade a pontos obtidos através de um experimento, por exemplo. Outra funcionalidade da interpolação é fazer a aproximação de funções complexas usando funções mais simples. Esse é o caso da interpolação linear onde uma curva de modelagem matemática mais complexa é aproximada por funções de retas.

A aproximação de uma curva por interpolação linear consiste em achar a função da reta que liga dois pontos consecutivos da curva original. Com essa função da reta é possível obter valores aproximados para qualquer ponto entre os pontos que determinaram a reta. A Figura 3.2 ilustra duas interpolações lineares para uma mesma curva, uma usando dois pontos da curva e outra usando quatro pontos.

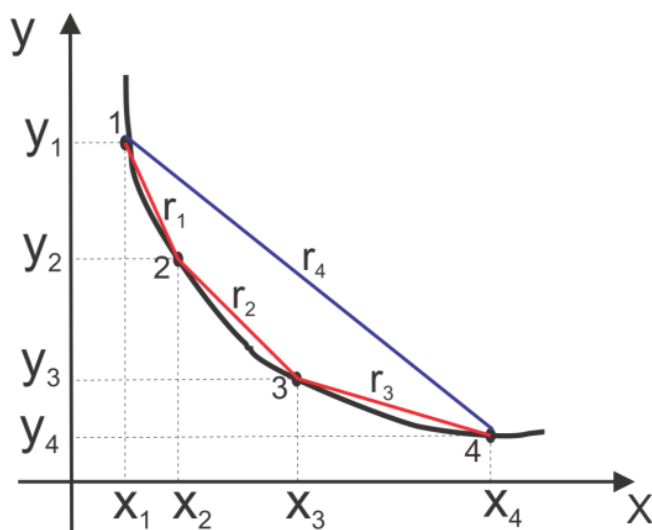


Figura 3.2: Gráfico de uma curva com a sua interpolação por quatro pontos e por dois pontos.

Analisando a Figura 3.2 é possível obter os pontos 1  $(x_1, y_1)$ , 2  $(x_2, y_2)$ , 3  $(x_3, y_3)$  e 4  $(x_4, y_4)$  da curva em preto, a partir desses pontos é possível traçar as retas  $r_1$ ,  $r_2$ ,  $r_3$  e  $r_4$ . Ao aplicar a interpolação linear entre os pontos 1 e 4, é traçada a reta  $r_4$ . Pode se notar que essa reta não gera uma aproximação fiel a curva real, o que pode causar resultados errôneos. Entretanto, com as retas  $r_1$ ,  $r_2$  e  $r_3$  é alcançada uma melhor aproximação da curva original. Essas retas podem ser obtidas através da interpolação linear entre os pontos 1, 2, 3 e 4.

Assim, sabendo o valor de dois pontos é possível obter a função da reta que passa por eles. Conhecendo essa função é possível obter qualquer ponto que faça parte dessa reta. A função de uma reta é representada por uma equação de primeiro grau

$$f(x) = y = ax + b \quad (3.1)$$

A partir de valores conhecidos de dois pares  $(x, y)$  torna-se possível calcular os valores de  $a$  e  $b$ , como mostrado nas equações seguir.

$$a = \frac{y_2 - y_1}{x_2 - x_1} \quad (3.2)$$

$$b = y_1 - \frac{y_2 - y_1}{x_2 - x_1} x_1 \quad (3.3)$$

Obtidos os valores de  $a$  e  $b$ , em função de  $x_1$ ,  $y_1$ ,  $x_2$  e  $y_2$ , é possível calcular o valor de  $y'$  em função do valor de  $x'$  compreendido entre  $x_1$  e  $x_2$ , pela equação 3.4 a seguir

$$y' = \frac{(y_2 - y_1)}{(x_2 - x_1)} x' + y_1 - \frac{(y_2 - y_1)}{(x_2 - x_1)} x_1 \quad (3.4)$$

Reorganizando temos

$$y' = y_1 + (x' - x_1) \frac{(y_2 - y_1)}{(x_2 - x_1)} \quad (3.5)$$

Com a equação 3.5 podem ser calculados valores aproximados para pontos da curva original. A interpolação linear faz a aproximação da curva utilizando dois pontos da curva original, ligando-os com retas, por isso o maior número de pontos da curva original possibilita

uma aproximação mais fidedigna, como pôde ser visto na Figura 3.2. Esta metodologia tem desvantagem quando não há pontos suficientes para aproximar a curva, como exemplificado pela reta  $r_4$  na Figura 3.2.

### 3.2 APLICAÇÃO DA INTERPOLAÇÃO NO CÁLCULO DO TEMPO RESTANTE DA BATERIA

Como visto no item 2.1.3, as curvas de descarga seguem um padrão, sempre diminuindo o tempo conforme a corrente vai aumentando, como ilustrado na Figura 3.3 a seguir.

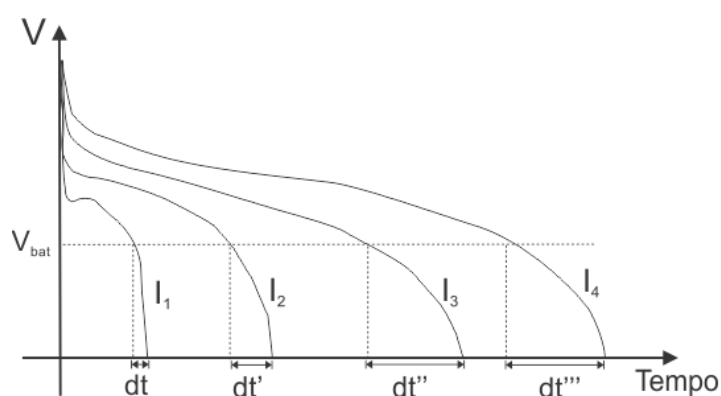


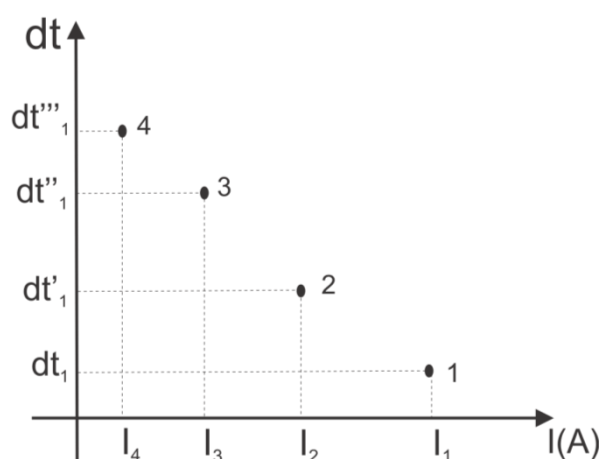
Figura 3.3: Exemplos representando curvas de descarga de uma bateria.

Na Figura 3.3 nota-se que para um mesmo valor de tensão  $V_{bat}$  é possível obter diferentes valores de tempo  $dt$  para diferentes valores de corrente  $I$ . Variando o valor de  $V_{bat}$ , são obtidos novos valores de  $dt$ . Esses valores são dispostos em uma tabela, como apresentado a seguir.

Tabela 3.1: Variação de  $dt$  por  $V_{\text{bat}}$  e  $I_{\text{cte}}$ .

$V_{\text{bat}}$	$I_1$	$I_2$	$I_3$	$I_4$	...
$V_0$	$dt_0$	$dt'_0$	$dt''_0$	$dt'''_0$	...
$V_1$	$dt_1$	$dt'_1$	$dt''_1$	$dt'''_1$	...
$V_2$	$dt_2$	$dt'_2$	$dt''_2$	$dt'''_2$	...
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

Ao observar a Tabela 3.1, é notado um retângulo destacando uma linha. Com os dados dessa linha, ou de qualquer outra, é possível montar um gráfico de corrente ( $I$ ) por variação de tempo ( $dt$ ),  $I \times dt$ , para um determinado valor fixo de tensão  $V$ . O gráfico montado a partir dos dados da linha  $V_1$  fica como o ilustrado pela Figura 3.4.

Figura 3.4: Gráfico apresentando pontos da curva  $I \times dt$  para uma tensão fixa  $V_{\text{bat}}$ .

O gráfico da Figura 3.4 apresenta a variação de  $dt$  de acordo com a corrente  $I$  para um valor de tensão  $V_1$  constante. Esses valores foram extraídos da linha em destaque na Tabela 3.1. É possível dizer que  $dt$  e  $I$  são inversamente proporcionais. Como o problema da metodologia estudada está no cálculo de  $dt$  para as correntes não tabeladas, utiliza-se os dados do gráfico da Figura 3.4, extraídos da Tabela 3.1, para possibilitar a interpolação linear entre os pontos 1 ( $I_1, dt_1$ ), 2 ( $I_2, dt'_1$ ), 3 ( $I_3, dt''_1$ ) e 4 ( $I_4, dt'''_1$ ). Realizando esse cálculo, pode-se

determinar os valores aproximados de  $dt$  para quaisquer valores de correntes não tabelados, entre os valores de  $I_1$  a  $I_4$ , como é apresentado na Figura 3.5.

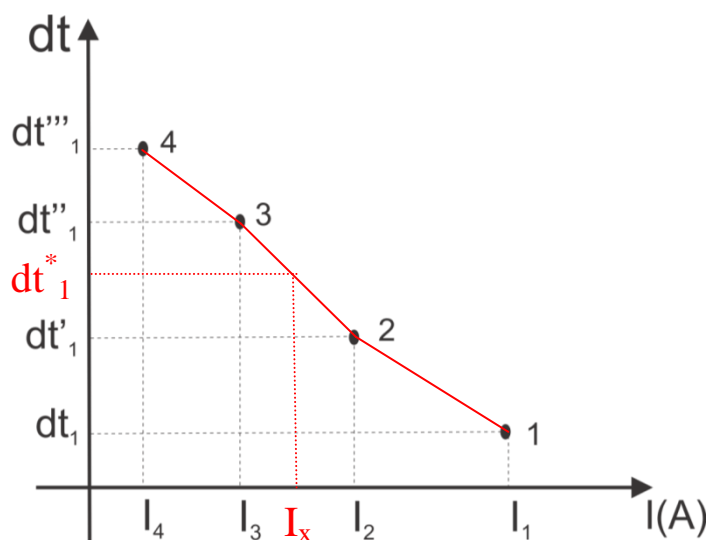


Figura 3.5: Gráfico representando o cálculo do tempo  $dt^*_1$  usando a interpolação linear.

A Figura 3.5 representa a interpolação linear entre os pontos 1, 2, 3 e 4. Com isso são determinadas as funções das retas que ligam esses pontos. Sabendo a função da reta que passa pelos pontos 2 e 3 é possível calcular o valor de  $dt^*_1$ , tempo restante de carga na bateria com a tensão  $V_1$ , para um determinado valor de  $I_x$ .

Para realizar os cálculos apresentados acima e estimar o tempo restante de carga na bateria, foi usado um microprocessador com a arquitetura ARM (*detalhado no Apêndice 1*).

### 3.3 O ARM7 LPC 2368

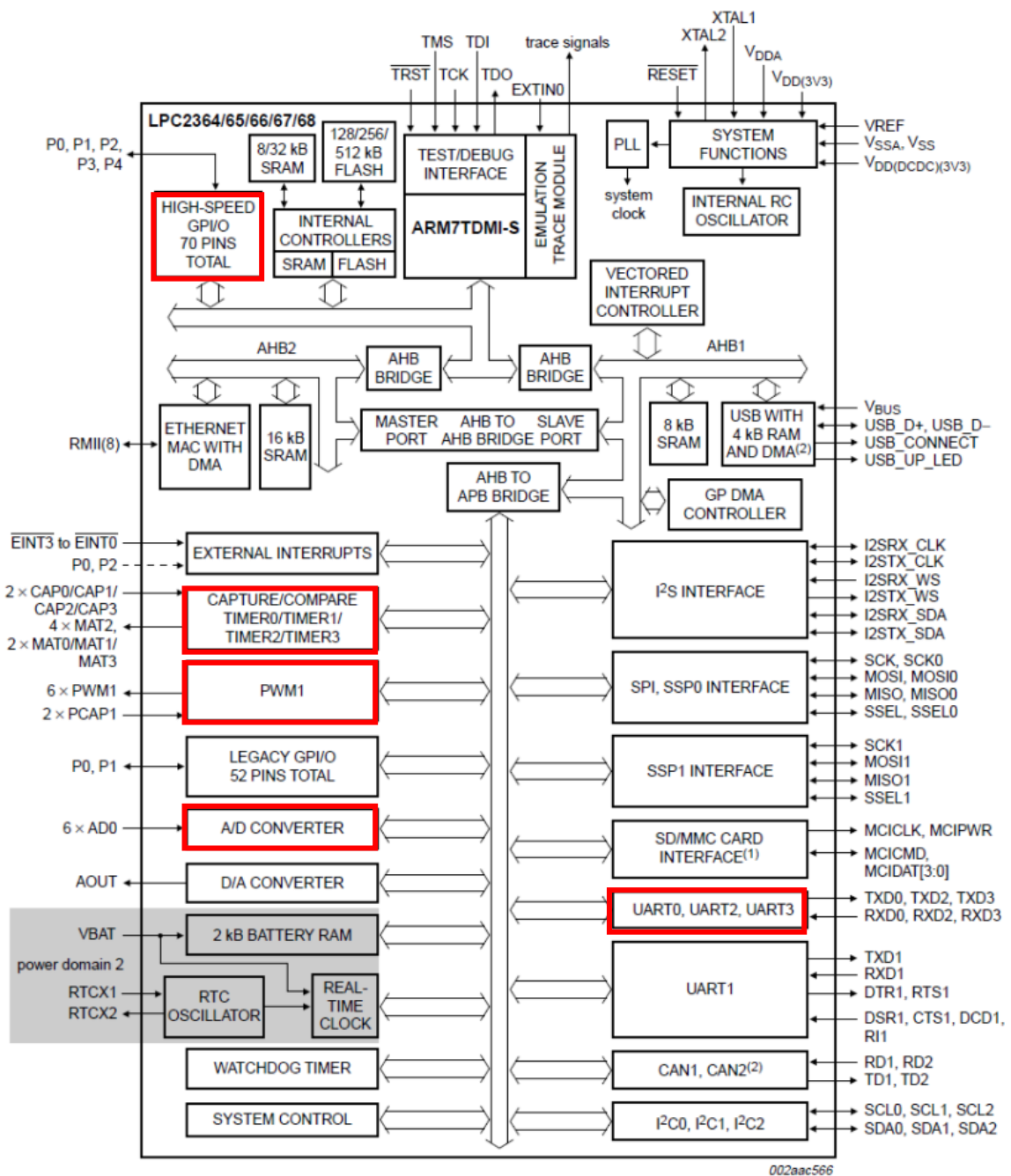
O ARM7 (*Advanced RISC Machine*) é um microprocessador do tipo RISC (*Reduced Instruction Set Computer*). A simplicidade na sua arquitetura é seu grande destaque, facilitando o entendimento e desenvolvimento de algoritmos. Neste trabalho, foi utilizado o microcontrolador LPC2368, um NXP da série LPC23XX. Segundo [8] ele tem um microprocessador ARM7 funcionando em até 72 MHz. E dispõe de até 512 kB de programa

de memória flash no chip com ISP (*In-System Programming*) e IAP (*In-Application Programming*), e até 64 kB de SRAM no *local bus* do ARM para ter mais velocidade no acesso à CPU.

### 3.3.1 CONFIGURAÇÃO DOS PERIFÉRICOS

O microprocessador ARM7 LPC2368 conta com vários periféricos úteis para o projeto de monitoramento da descarga de bateria. A seguir serão apresentados os periféricos usados, e suas possíveis configurações. Na Figura 3.6 mostrada a seguir pode ser visto o diagrama de blocos do microprocessador, onde é possível ver todos os periféricos disponíveis no LPC2368, entretanto nem todos serão descritos neste capítulo. Como foi dito anteriormente, apenas os periféricos usados para o projeto receberão um destaque especial.

Os periféricos destacados em vermelho na Figura 3.6 são apresentados nos próximos tópicos, e detalhados no Apêndice 1. Eles são as entradas e saídas digitais GPIO's (*In, Out*), os conversores analógicos digitais ADC's (*Analog Digital Converter*), os contadores internos, o modulador por largura de pulso PWM e a interface de comunicação serial assíncrona UART (*Universal Asynchronous Receiver Transmitter*).



- (1) LPC2367/68 only.  
 (2) LPC2364/66/68 only.

Figura 3.6: Diagrama de bloco do microcontrolador LPC2368. Fonte: [8]

### 3.3.1.1 ENTRADAS E SAIDAS DIGITAIS– GPIO

As GPIO's são as entradas e saídas digitais do microprocessador, elas são as responsáveis por fazer a transmissão de dados entre dispositivos de hardware. Nessas entradas e saídas são reconhecidos apenas os níveis lógicos “1” ou “0”. Na entrada, o software reconhece quando é aplicada uma tensão acima de 3 V em um pino do hardware, com um bit de valor igual a “1”, e quando se tem aplicada uma tensão baixa, ou seja, com um valor abaixo dos 3V o sinal reconhecido é 0. Quando é usado como saída os sinais têm o mesmo significado, mas dessa vez é o software quem fornece os sinais, aplicando ou não uma tensão em um pino do hardware.

Essas portas podem ter várias funções, como por exemplo, controlar leds, acionar relés, detectar sinais de entrada, enviar dados para outros periféricos, etc. No projeto as entradas e saídas de dados digitais foram usadas para realizar a comunicação com o display gráfico. Entretanto para a medição de tensão e corrente não é possível utilizar as GPIO, sendo necessária a utilização dos conversores analógico-digital.

### 3.3.1.2 CONVERSOR ANALÓGICO-DIGITAL - ADC

Muitas vezes, não precisamos apenas saber o estado dos sinais digitais, como 0 ou 1, com ou sem tensão, mas também precisamos saber a intensidade desses sinais, quando se deseja fazer alguma medição. Para isso existem os conversores de sinais analógicos para digitais, os ADC's (para simplificação serão chamados de AD's) que fazem a conversão do sinal analógico, um valor de tensão, para um sinal digital, uma combinação de bits.

No caso do LPC2368 essa conversão é feita para um valor de 10 bits, ou seja, para uma tensão de 0 V, o AD produz como resultado o valor 0, e a máxima tensão na entrada do AD é interpretada como 1023 (máximo valor de 10 bits:  $2^{10}-1 = 1023$ ). A variação da tensão medida na entrada é de 0 a 3 V, e o tempo que o microprocessador leva para realizar a conversão, varia em torno de 2,4 $\mu$ s. O conversor AD foi usado para a medição da tensão na



bateria,  $V_{bat}$  e das correntes, de referência  $I_{ref}$  e da bateria  $I_{bat}$ . As correntes  $I_{ref}$  e  $I_{bat}$  além de atualizarem o display, são usadas no controle da largura de pulso do sinal PWM.

### 3.3.1.3 MODULAÇÃO POR LARGURA DE PULSO - PWM

Outro periférico de grande importância é o gerador de sinal modulado por largura de pulso, o PWM, esse é um tipo de sinal onde a amplitude e a frequência se mantêm constante e o que varia é a largura do pulso de acordo com o valor do sinal de referência. O sinal PWM é resultado de uma comparação entre uma onda triangular ou “dente de serra” com um sinal de referência, como pode ser visto na Figura 1.2.

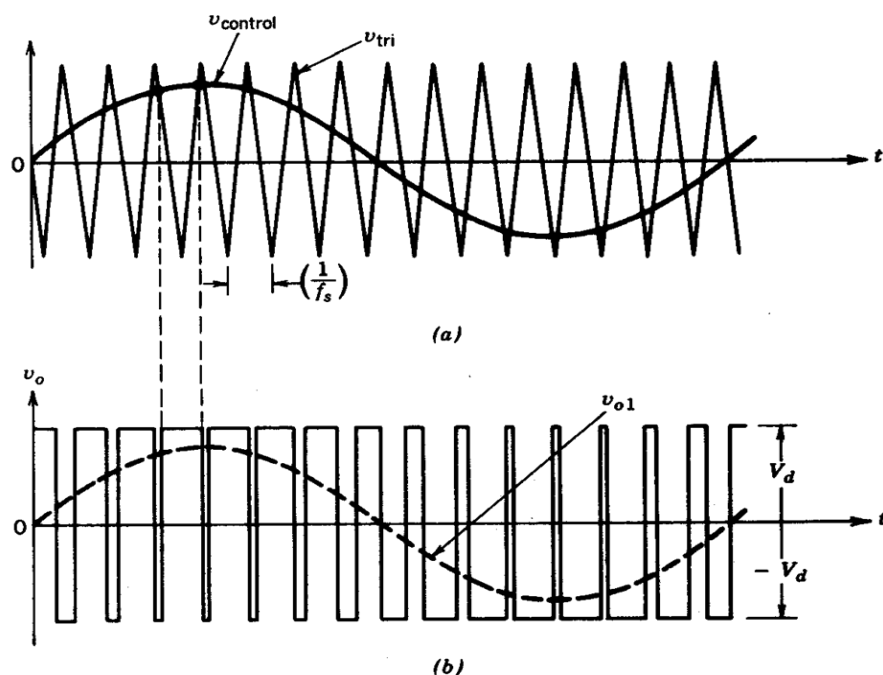


Figura 3.7: (a) Sinais de comparação para gerar o sinal PWM. (b) Sinal PWM resultante comparado com o sinal modulador. Fonte: [9].

O sinal PWM apresentado na Figura 1.2(a) é um exemplo para modulação de valor de tensão onde o sinal de referência é a onda senoidal  $v_{control}$  que é comparada com uma onda triangular  $v_{tri}$ . Pode-se notar na Figura 3.7(b) que a largura do pulso varia de acordo com o valor instantâneo da onda senoidal: na parte mais alta da onda, o pulso tem uma largura de

quase 100% do período, e nas partes mais baixas da senóide, a largura do pulso é quase 0% do período. No ARM a modulação por largura de pulso funciona como exemplificada na Figura 3.8.

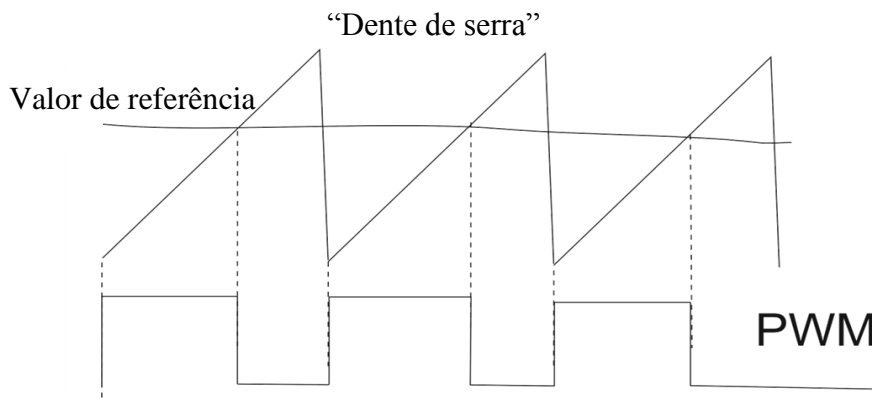


Figura 3.8: Onda “dente de serra” comparada com um sinal de referência e sinal PWM gera.

Na Figura 3.8 o sinal do PWM começa quando a contagem da onda “dente de serra” é iniciada. A largura do pulso é definida pelo sinal de referência. O sinal do PWM se mantém com valor lógico alto (1) enquanto o valor da “dente de serra” é menor que o valor do sinal de referência, quando esses valores se igualam, o sinal PWM muda para o valor lógico baixo (0), e se mantém assim até a o período se reiniciar, ou seja, quando a “dente de serra” chega ao seu valor máximo. Esse modo de operação do PWM é chamado de *single edge*. Nesse microprocessador existem dois modos de operação do sinal PWM, o *single edge* e o *double edge*, sendo esse segundo e a diferença entre eles melhor explicada no ApêndiceErro! **Fonte de referência não encontrada.** 1.

#### 3.3.1.4 CONTADORES INTERNOS

O microprocessador ARM7 LPC2368 conta com quatro contadores internos, onde é possível criar interrupções controladas. Esses contadores podem ser usados para contar ciclos de *clock*, e realizar alguma ação após um determinado numero de ciclos, definido no

comparador. Com essa configuração são chamados de *TIMER* ou podem capturar eventos externos e assim são chamados de *COUNTER*.

No *timer* a interrupção é definida pelo número do ciclo de *clock*, quando esse valor é alcançado acontece uma interrupção no programa, que foi usada para atualizar o display, já que não era necessário atualizar toda vez que o programa roda.

### 3.3.1.5 COMUNICAÇÃO UART

A UART é um periférico do microprocessador, que converte dados de comunicação serial para a forma de bits em paralelo e vice-versa. Na comunicação serial os dados são enviados para entre o receptor e o transmissor em sequência, bit a bit, bastando apenas um par de fios para realizar a comunicação. Já na comunicação paralela, os dados são enviados byte a byte, ou em quantidade maior de bits de uma só vez, mas para isso é necessário uma quantidade maior de fios, do que na comunicação serial.

Antes de cada grupo de bits de dados enviados, é primeiro enviado o bit de partida, que é um bit de valor lógico 0. Ao final é enviado o bit de parada, com valor lógico 1, e antes do bit de parada pode ser enviado também o bit de paridade. O tamanho do pacote de dados que é enviado pode ser programado via software, variando de 5 a 8 bits.

Algumas UART contam ainda com um buffer de memória FIFO (*First In, First Out*) antes do seu receptor, aonde certa quantidade de dados vai ficando armazenada para evitar perda de dados, quando ocorre uma transmissão com quantidade elevada de dados. No LPC2368 a UART é de 16 Bytes. A UART foi usada para realizar a comunicação com o computador e isso possibilitou a calibração dos ADs.

### 3.3.2 INTERFACE HOMEM MÁQUINA - IHM

A interface entre o usuário e o microprocessador pode ser feita de várias maneiras, uma delas é o uso de um display onde podem ser expostas as informações do sistema em tempo real, como tensão, corrente e tempo de bateria.

Neste trabalho, foi utilizado um display tipo GLCD (*Graphic Liquid Crystal Display*) controlado pelo microprocessador T6963 da Toshiba. A comunicação entre o display e o microprocessador ARM é feita pelas portas IO (*In Out*). Para utilizar o display são necessárias oito portas do tipo IO para a transferência de dados e seis IO para controle e programação do display, ou seja, são necessárias 14 IO. Como apresentado no item 4.1.1, no ARM estão disponíveis 70 IO, viabilizando a comunicação da IHM com o microprocessador.

Na Tabela 1.5 é apresentado um resumo da pinagem do display e suas funções.

Tabela 3.2: Pinagem do display GLCD

Nº	Nome	Função
1	FGND ( <i>Frame Ground</i> )	Terra do display (aterramento da carcaça).
2	V <sub>ss</sub>	Terra para o circuito lógico (referência de tensão zero).
3	V <sub>DD</sub>	Entrada de alimentação para o circuito lógico.
4	V <sub>0</sub>	Entrada de alimentação para o circuito do display.
5	WR ( <i>Write</i> )	Para mandar um byte para o display.
6	RD ( <i>Read</i> )	Para ler um byte do display.
7	CE ( <i>Chip Enable</i> )	Habilita o chip.
8	C/D ( <i>Comand/Data</i> )	Seleciona se o byte enviado para o display vai ser um comando ou um dado.
9	V <sub>EE</sub>	Entrada de alimentação de tensão negativa
10	RES ( <i>Reset</i> )	Reinicializa o display
11	DB0 ( <i>Data Bus Line 0</i> )	Pino de dado que recebe o bit menos significativo (LSB)
12	DB1 ( <i>Data Bus Line 1</i> )	Pino de dado intermediário
13	DB2 ( <i>Data Bus Line 2</i> )	Pino de dado intermediário
14	DB3 ( <i>Data Bus Line 3</i> )	Pino de dado intermediário

<b>15</b>	DB4 ( <i>Data Bus Line 4</i> )	Pino de dado intermediário
<b>16</b>	DB5 ( <i>Data Bus Line 5</i> )	Pino de dado intermediário
<b>17</b>	DB6 ( <i>Data Bus Line 6</i> )	Pino de dado intermediário
<b>18</b>	DB7 ( <i>Data Bus Line 7</i> )	Pino de dado que recebe o bit mais significativo (MSB)
<b>19</b>	FS ( <i>Font Select</i> )	Seleciona a fonte do display entre 6x8 e 8x8
<b>20</b>	N.C.	Não conectado

Para realizar a comunicação do display com o ARM, foram definidas algumas máscaras para as GPIO com os nomes dos pinos em que elas estão conectadas no display. Isso facilitou a comunicação do ARM com o display e possibilitou o monitoramento da bateria com informações da corrente de saída, do tempo restante de carga e da tensão atual.

Com esse microcontrolador foi montada uma bancada para realizar o experimento, e testar a metodologia.

### 3.4 BANCADA

O passo mais importante para determinar o cálculo do tempo restante de carga em uma bateria, usando as curvas de descarga e a interpolação linear é montar a Tabela 3.1 com valores reais de um modelo específico de bateria. Para montar essa tabela é preciso obter os valores de variação da tensão em função do tempo para no mínimo dois valores corrente constante. Entretanto não é fácil encontrar na literatura a quantidade necessária de dados de um mesmo modelo de bateria para montar essa tabela. Por isso foi primeiro montada uma bancada para coletar esses dados.

### 3.4.1 BANCADA EXPERIMENTAL PARA DESCARGA DE BATERIA COM CORRENTE CONSTANTE

Não seria possível descarregar a bateria com corrente constante usando simplesmente uma carga invariável. Durante o processo de descarga, pela lei de Ohm, a corrente varia com o decaimento da tensão. Assim, caso a bateria seja ligada diretamente a uma carga constante, a sua corrente de descarga irá variar, tornando-se necessário empregar um controle para realizar tal procedimento.

Para controlar a corrente a fim de mantê-la constante, foi utilizado um conversor CC/CC (corrente contínua – corrente contínua) do tipo *Buck*. O conversor CC/CC do tipo *Buck*, é um dispositivo chaveado que reduz a tensão na carga em relação à tensão da fonte, por isso ele também é conhecido como abaixador. Nesse tipo de conversor não há a inversão da polaridade, e o fluxo de energia é controlado por uma chave em série com a carga. A Figura 3.9 representa um conversor *Buck* com filtros na entrada e na saída.

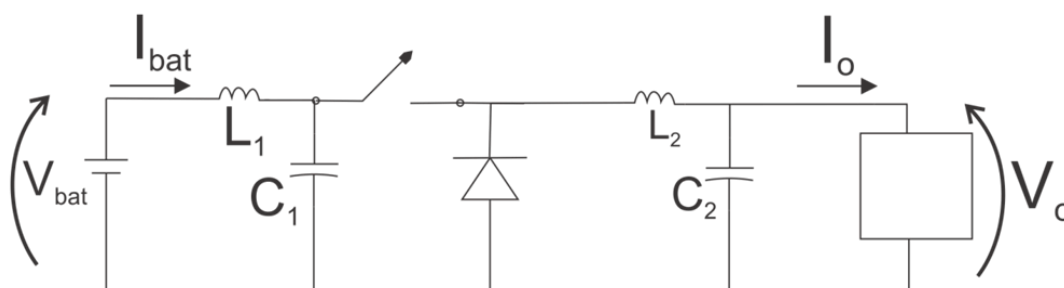


Figura 3.9: Circuito de um conversor *Buck* com filtro na entrada e na saída.

A tensão  $V_{bat}$  na Figura 3.9 representa a tensão de entrada no *Buck*, ou seja, a tensão a bateria. A tensão  $V_o$  é a tensão na carga. Segundo [9] a relação entre essas tensões é dada por

$$D = \frac{V_o}{V_{bat}} \quad (3.6)$$

Onde  $D$  é o ciclo de trabalho (*duty cycle*) do chaveamento e varia de 0 a 1, tornando assim a tensão de saída  $V_o$  uma proporção da tensão de entrada  $V_{bat}$ . Esse ciclo de trabalho é determinado pelo sinal PWM, que controla o chaveamento. O indutor  $L_2$  na saída gera o

alisamento da corrente fornecida à carga, enquanto o capacitor  $C_2$  funciona como um filtro para a tensão. O conversor *Buck* pode funcionar de dois modos diferentes dependendo dos valores definidos para o filtro LC. Existe o modo de operação descontinuo e o modo contínuo. No funcionamento descontinuo a corrente na carga chega a zero quando a chave está aberta. Quando o filtro é dimensionado para o conversor operar em modo contínuo, essa variável não chega a zero, apenas ocorre um pequeno *ripple*. Como o objetivo era de controlar a corrente de descarga da bateria a fim de manter ela constante, o filtro da entrada foi dimensionado para operação em modo contínuo. Ainda observando a Figura 3.9 é notado um filtro  $L_1C_1$  na entrada do conversor. Esse filtro tem o objetivo de manter em modo de condução contínua a corrente de entrada, da bateria.

O *Buck* operando em regime permanente e em condução contínua os filtros  $L_1C_1$  e  $L_2C_2$  não afetam a relação entre  $V_o$  e  $V_{bat}$ , mantendo válida a equação 3.6. Considerando as perdas no semicondutor e nos filtros desprezíveis, é mantida a relação da conservação de energia

$$P_{in} = P_{out} \quad (3.7)$$

Assim, também é mantida a relação

$$\bar{V}_{bat} \bar{I}_{bat} = \bar{V}_o \bar{I}_o \quad (3.8)$$

Logo se tem

$$\frac{\bar{V}_{bat}}{\bar{V}_o} = \frac{\bar{I}_o}{\bar{I}_{bat}} = \frac{1}{D} \quad (3.9)$$

Como o objetivo é controlar a corrente  $I_{bat}$  da bateria

$$\bar{I}_{bat} = \bar{I}_o D = \frac{\bar{V}_o}{R} D = \frac{\bar{V}_{bat}}{R} D^2 \quad (3.10)$$

A equação 3.10 apresenta a relação entre a variável controlada  $I_{bat}$ , a tensão de entrada  $V_{bat}$ , que é a tensão nos terminais da bateria, e o ciclo de trabalho  $D$ . Dessa maneira torna-se possível controlar a corrente na entrada do conversor. Após controlar a descarga e obter os dados necessários para o cálculo do tempo remanescente de carga na bateria, esse conversor foi usado como uma carga variável a fim de e verificar o cálculo do tempo. No capítulo 4 é explicado o controle e são apresentados os resultados experimentais.

## 4 BANCADA E RESULTADOS EXPERIMENTAIS

Para o uso da metodologia do cálculo do tempo restante de carga na bateria baseado nas curvas de descarga, foi feito uma programa em linguagem C para a configuração dos periféricos do ARM e para realizar os cálculos de interpolação das curvas de descarga de bateria.

Foi montada uma bancada experimental para testar o cálculo do tempo restante de carga na bateria usando o mesmo conversor CC/CC, apresentado no item 3.4, controlado por corrente, para poder ter a corrente de saída da bateria controlada. Mas dessa vez a corrente de referência foi controlada por um potenciômetro e podia ser variada pelo usuário. Desta forma pode-se acompanhar a resposta do tempo calculado, e saber o tempo restante de carga na bateria de acordo com a corrente definida e para a carga.

### 4.1 A BANCADA EXPERIMENTAL

Para colher os dados de tensão e tempo para correntes constantes e para testar a metodologia de calculo de tempo, foi montada uma bancada experimental conforme esquematizado no digrama de blocos da Figura 4.1.

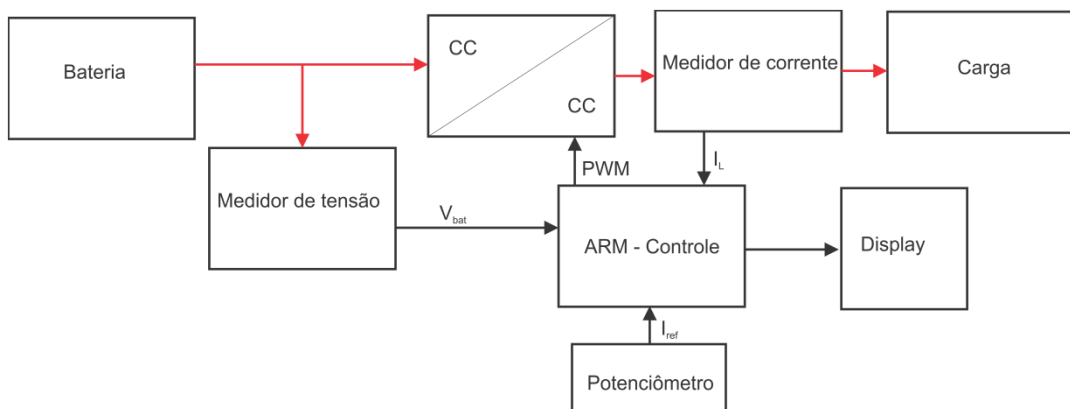


Figura 4.1: Diagrama de blocos da bancada de teste.



No diagrama de blocos da Figura 4.1 está ilustrada a bancada que foi montada para realizar os testes, tanto de descarga como para o monitoramento. As setas em vermelho indicam as ligações elétricas e as setas pretas representam os sinais de medição e controle.

A bateria utilizada foi uma Optima D51 de 38 Ah, essa é uma bateria estacionária selada de gel. Ela foi conectada ao conversor CC/CC e este à carga, um reostato com resistência em  $0,3 \Omega$ . Para o controle a bateria foi conectada ao sensor de tensão, e há um sensor de corrente *Hall*. Era medida a corrente na entrada do conversor, ou seja, a corrente na saída da bateria, esse valor de corrente é uma das entradas na placa do microprocessador, em conjunto com a corrente de referência, valor definido por um potenciômetro, definia o ciclo de trabalho do sinal PWM. O sinal de PWM gerado controla o chaveamento do conversor CC/CC. Os valores medidos são exibidos em um display gráfico.

O material listado a seguir foi utilizado na montagem da bancada:

- Bateria estacionária Optima D51 de 38 Ah;
- Medidor de tensão;
- Conversor CC/CC tipo *Buck* com filtro LC na entrada;
- Medidor de corrente: sensor de corrente *Hall* da LEM HAS50;
- Microprocessador ARM7 LPC2368;
- Display GLCD da Toshiba T6369;
- Potenciômetro;
- Carga: reostato ajustado para  $0,3\Omega$ ;

O valor da resistência no reostato foi ajustado para  $0,3 \Omega$  a fim de o controle conseguir manter uma descarga de 30 A mesmo com a bateria em uma tensão de 10 V. A seguir na Figura 4.2 é ilustrado o circuito elétrico da bancada experimental.

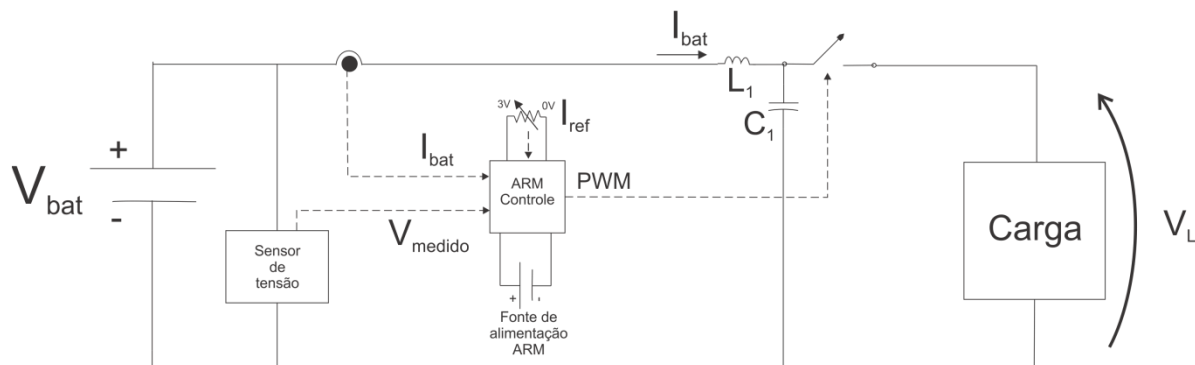


Figura 4.2: Circuito elétrico da bancada experimental.

Como pode ser notado na Figura 4.2 o conversor CC/CC usado foi um *Buck* modificado, com apenas o filtro  $L_1C_1$  na entrada. Só foi colocado o filtro na entrada, pois o que era importante manter em condução constante, não deixando ficar pulsado, era  $I_{bat}$ . Foi usado um indutor  $L_1$  de 0,3 mH e um capacitor  $C_1$  de 2,2 mF, isso gerou um frequência de corte de 196 Hz, dessa maneira, duas décadas acima a frequência de chaveamento fica em 10 kHz. Entretanto em 10 kHz estava ocorrendo uma ressonância que aumentava o *ripple* da corrente. Isso foi solucionado quando foi usada uma frequência de chaveamento de 6,55 kHz.

Na placa de controle, onde foi soldado o microcontrolador, tem uma fonte de alimentação externa. O potenciômetro varia o seu sinal de 0 a 3 V, que é o range de leitura de tensão na entrada do AD, e com isso gera o sinal da corrente de referência. O controle implementado é mais bem detalhado a seguir.

## 4.2 CONTROLE

Os periféricos do microprocessador usados para o controle e o monitoramento da descarga da bateria foram:

- ADC
- PWM
- GPIO
- *TIMER*

O ADC foi configurado para ler os dados recebidos do sensor de corrente, do potenciômetro de referência e do sensor de tensão da bateria. Como é preciso ler três dados do sistema, foram habilitadas três entradas ADC.

Na Figura 4.3 está o diagrama do algoritmo implementado no ARM. É possível observar, como os periféricos são usados. Na parte superior da figura é esquematizado o funcionamento do controle do PWM e a parte inferior representa os cálculos para determinação do tempo restante de carga e a comunicação com o display.

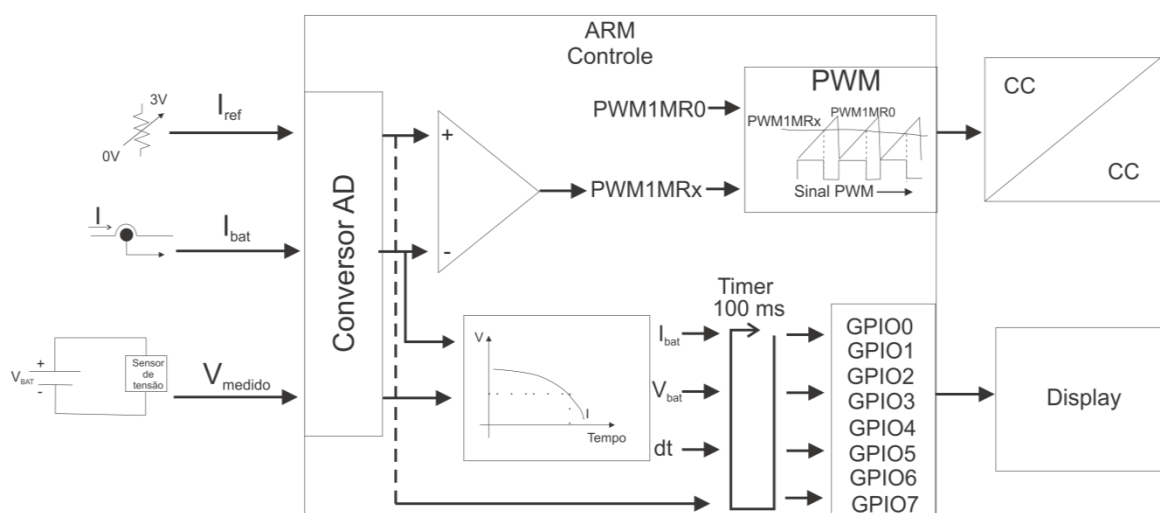


Figura 4.3: Algoritmo implementado no ARM.

Observando a Figura 4.3 nota-se que no ARM são usadas três entradas AD, sendo que duas entradas representam valores de corrente e uma representa valor de tensão. Como conversor AD faz a conversão do sinal analógico, de 0 a 3 V, para um sinal digital, de 0 a 1023, os valores na entrada do programa são os sinais digitais. Para facilitar os cálculos e plotar no display, os sinais digitais são convertidos para valores de corrente e tensão, usando o sistema de base, como apresentado nas equações 4.1 e 4.2 abaixo.

$$V = \frac{AD_{tensão} \times V_{max}}{1023} \quad (4.1)$$

$$I = \frac{AD_{corrente} \times I_{max}}{1023} \quad (4.2)$$

$AD_{\text{tensão}}$  representa o resultado da conversão do AD usado na medição da tensão e  $AD_{\text{corrente}}$  os resultados dos ADs usados para as medições de corrente. Na Figura 4.3 os valores de  $V_{\text{bat}}$  e de  $I_{\text{bat}}$  são usados para calcular o tempo restante de carga na bateria usando a metodologia da interpolação linear e os valores de  $I_{\text{bat}}$  e  $I_{\text{ref}}$  são usados para controlar a largura do pulso do sinal de PWM. Depois de calculado o valor de  $dt$ , os valores de  $I_{\text{ref}}$ ,  $I_{\text{bat}}$  e  $V_{\text{bat}}$  são atualizados no display.

O PWM foi configurado para controlar o chaveamento do conversor CC/CC. Para isso foi criada a função `PWM_init`. Essa função tem como parâmetro de entrada o valor desejado para frequência de chaveamento. Nessa função é criada uma variável chamada `cont_PWM`, representada pela equação 4.3

$$\text{cont\_PWM} = \frac{\text{Clock}}{\text{Frequência}} \quad (4.3)$$

A variável *clock*, é o valor da frequência de *clock* em que o microprocessador está operando e a variável *Frequência*, é o parâmetro de entrada da função, ou seja, a frequência de chaveamento.

Os valores de  $I_{\text{bat}}$  e  $I_{\text{ref}}$  são usados no controle da largura de pulso do sinal PWM. Primeiro é calculado o  $I_{\text{ctrl}}$ , o valor de corrente que será usado no comparador, como é ilustrado na Figura 4.4.

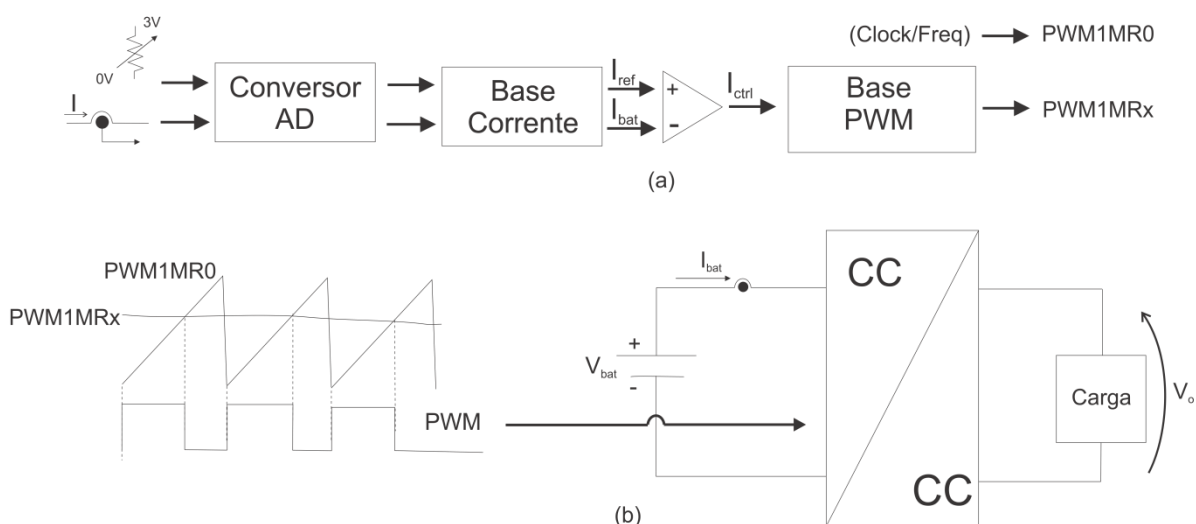


Figura 4.4: Controle do PWM: (a) Controle do sinal PWM no ARM, (b) Sinal de referência, “dente de serra” e sinal PWM gerado.

Na Figura 4.4(a) o valor de  $I_{ctrl}$  passa por uma mudança de base antes de atualizar o registrador PWM1MRx do PWM. Essa mudança de base é feita para transformar o valor de corrente em um valor entre 0 e o valor PWM1MR0 que recebeu o valor da variável  $count\_PWM$ . Depois de feita essa transformação, o valor do registrador PWM1MRx é atualizado e então comparado com o valor do PWM1MR0, que representa a onda dente de serra. Isso pode ser visto na Figura 4.4(b), onde também está ilustrado o sinal do PWM gerado que vai controlar o chaveamento do conversor CC/CC.

Na Figura 4.5 é apresentado o diagrama blocos da parte usada para o monitoramento e o IHM, onde é possível controlar a corrente  $I_{ref}$  e monitorar a tensão, as correntes, e o tempo.

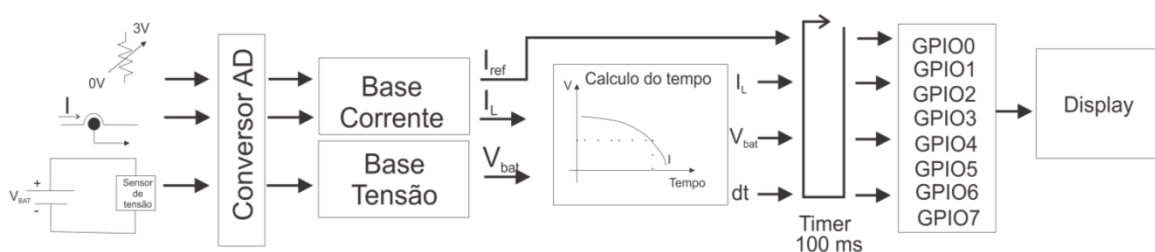


Figura 4.5: Diagrama do algoritmo implementado no ARM para calculo do tempo e IHM.

Como pode ser visto na Figura 4.5, os valores gerados pelos ADs primeiro passam pela conversão que foi representada nas equações 4.1 e 4.2. Depois de convertidos os valores  $I_{bat}$  e  $V_{bat}$ , que representam respectivamente a corrente de descarga e a tensão na bateria, são usados para o cálculo do tempo restante de energia. O cálculo do tempo é feito usando a metodologia das curvas de descarga e interpolação linear. Depois de realizados os cálculos, os valores de  $I_{bat}$ ,  $V_{bat}$ ,  $dt$  e  $I_{ref}$  são exibidos no display.

O contador interno foi configurado como temporizador (*Timer*), para gerar uma interrupção a cada 100 ms. Nessa interrupção ocorre a transferência dos dados entre o ARM e o display. Ou seja, os dados mostrados no display são atualizados a cada 100 ms.

### 4.3 RESULTADOS EXPERIMENTAIS

Primeiro foram realizadas as descargas completas com corrente constante. Para realizar essas descargas, as baterias foram carregadas passando 6 horas carregando com uma tensão de 14,8 V e corrente limitada em 5 A, e depois que a corrente ficou abaixo de 1 A a bateria ficou por mais 2 horas carregando na tensão de 13,8 V.

Com as baterias carregadas foram realizadas as descargas utilizando o conversor CC/CC. A corrente de referência  $I_{ref}$  foi definida previamente no programa de controle de ARM, ou seja, não foi utilizado o potenciômetro. O controle PWM implementado no ARM regulou a corrente de descarga para o valor de  $I_{ref}$  definido no programa. Isso pode ser visto a Figura 4.6.

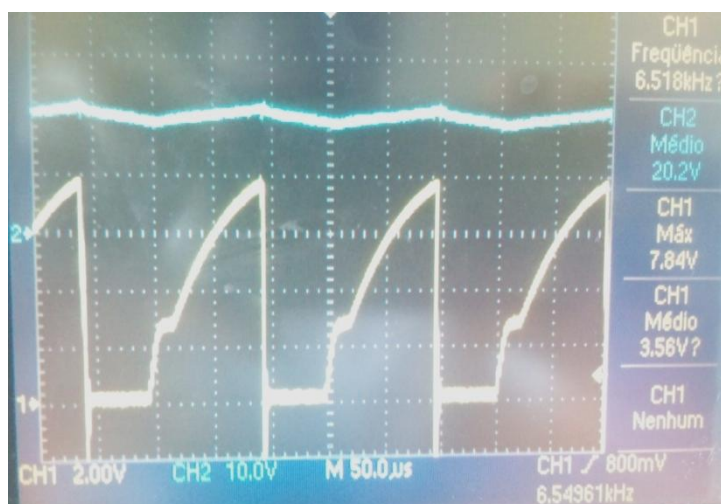


Figura 4.6: Tela do osciloscópio durante a descarga com corrente constante de 20 A.

Na Figura 4.6 o canal 1, em amarelo, é o pulso nas chaves e o canal 2, sinal azul, está medindo a corrente saindo da bateria, ou seja, a corrente de descarga. Como pode ser observado a corrente se mantém em condução contínua, não chegando a zero. Os dados obtidos para gerar a curva real, foram usados no programa para o cálculo do tempo restante da bateria.

Nos testes foram definidas correntes de 15 A, 20 A, 25 A, 27 A e 30 A. Foi conectado aos terminais da bateria um multímetro digital, ligado via serial com um

computador onde foi gravado o tempo para cada queda de 0,1 V na tensão da bateria. Com esses dados medidos foi possível traçar as curvas de descarga reais da bateria.

Na Figura 4.7 a seguir, são apresentados os resultados experimentais. As curvas da tensão da bateria em função do tempo para uma corrente de descarga constante são esboçadas graficamente e depois são usadas no programa para fazer a estimativa do tempo restante de energia na bateria.

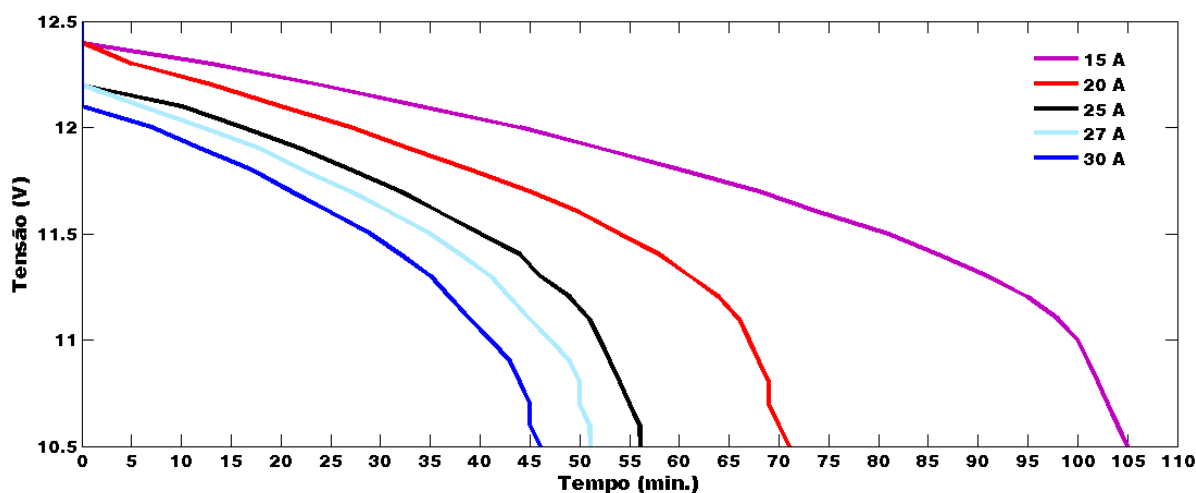


Figura 4.7: Curvas reais de descarga para 15, 20, 25, 27 e 30 A da bateria Optima D51.

Na Figura 4.7 pode ser verificado que o tempo de carga na bateria não aumenta linearmente com a redução da corrente. O tempo que a bateria leva para se descarregar com a corrente de 30 A é de 46 minutos, diminuindo a corrente para 25 A esse tempo é acrescido de 10 minutos, ou seja, o tempo total é de 56 min., reduzindo mais 5 A, em 20 A, o tempo aumenta em 25 min., indo para 71 min. o tempo total de carga com 15 A cresce 59 min. ficando em 105min.

Ainda é possível observar que as curvas seguiram o padrão das curvas de descarga, como foi descrito nesse trabalho. Elas têm um começo mais linear e o final ocorre uma queda de tensão mais rápida. Assim essas curvas geradas foram validadas e seus dados de origem, os pontos de tensão por tempo para cada valor de corrente, foram usados para o teste da estimativa do tempo restante de carga na bateria.

Com os pontos das curvas de 20 A e 30 A apresentados na Figura 4.7 foi possível realizar uma interpolação e gerar uma curva para a corrente de 25 A interpolada para estimar o tempo e compará-la com a curva de 25 A real da bateria, como pode ser visto na Figura 4.8.

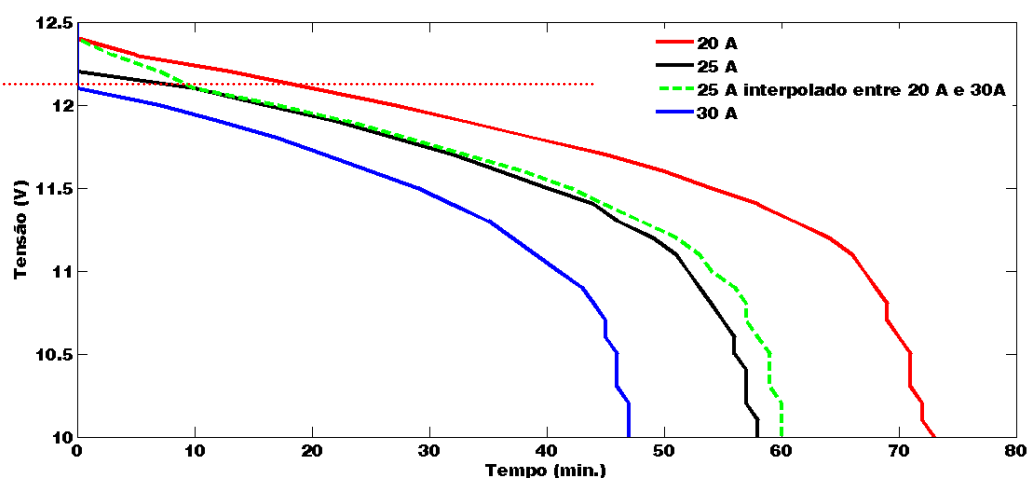


Figura 4.8: Curvas reais de descarga com correntes 20 A, 25 A e 30 A e interpolada de 25 A.

Na Figura 4.8 é possível comparar a curva real de 25 A, curva preta, e a curva de 25 A interpolada usando os dados das curvas de 20 A e 30 A, curva verde pontilhada. A curva original tem uma duração total de 58 minutos, já a curva interpolada tem uma duração de 60 minutos, gerando uma diferença de 2 minutos em relação ao tempo total de descarga real, ou seja, um erro de 3%. No começo da curva também possível notar uma diferença entre a curva real de 25 A e a interpolada. Essa diferença ocorre devido a queda de tensão inicial da descarrega de 30 A. Como pode ser observado, abaixo da linha vermelha pontilhada, quando passa a queda de tensão inicial da corrente de 30 A, a curva de 25 A interpolada segue a curva de 25 A real.

Depois de gerados os dados da curva de 25 A a partir da interpolação entre os dados de 20 A e de 30 A, para comprovar que a quantidade de dados, e a aproximação entre eles, influenciam no resultado da interpolação, foram realizadas interpolações para obter os dados da curva de 27 A. Os dados obtidos com essas interpolações foram plotados e comparados com a curva de 27 A real, como está ilustrado na Figura 4.9.



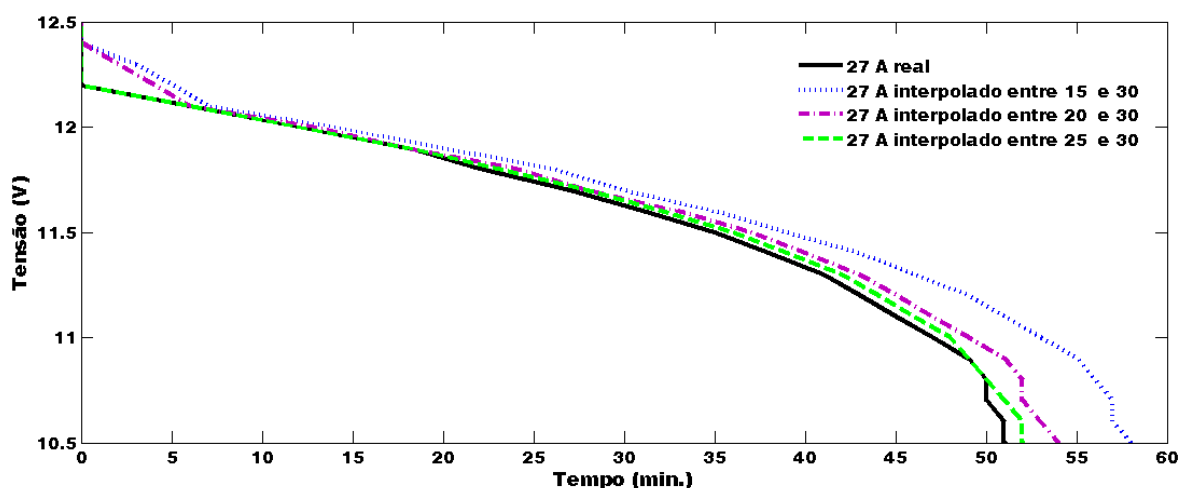


Figura 4.9: Curvas de 27 A real e interpoladas.

Na Figura 4.9 é possível ver as interpolações realizadas para estimar o tempo restante de carga na bateria com uma descarga de 27 A. Utilizando os dados das curvas de 25 A e 30 A, foram gerados os dados que originaram a curva verde tracejada. Essa curva conseguiu representar fielmente a curva real de 27 A, com um erro de apenas 1 minuto, ou seja, 2%. A interpolação usando as informações das curvas de 20 A e 30 A, foi possível construir a curva ponto-traço roxa. Essa curva apresenta um erro maior, de 3 minutos, 6%, em relação ao tempo total da curva real. E a partir dos pontos das curvas de 15 A e 30 A, a curva azul pontilhada foi gerada. Essa apresenta um erro ainda maior, de 15 %, 7 minutos.

É possível notar o aumento do erro com o aumento da diferença entre os valores de correntes usadas. Esses erros ocorrem devido ao tempo não crescer linearmente com a redução da corrente, como foi visto na Figura 4.7.

Para as aplicações desse estudo, um erro de 1 minuto entre o tempo calculado e o tempo real de uma descarga, é aceitável, validando essa metodologia. Entretanto esse erro pequeno só existe quando a diferença entre os dados iniciais é pequena. Por exemplo, quando são usados dados das curvas com um diferença de 5 A o erro é menor do que quando são usados dados de curvas com diferença de 10 A ou 15 A.

## 5 CONCLUSÕES E TRABALHOS FUTUROS

Nesse capítulo são apresentadas as conclusões relativas à bateria, ao uso da interpolação para o cálculo do tempo restante da bateria e como foi o uso do ARM para realizar os controles. Além das conclusões são apresentadas as propostas para trabalhos futuros.

### 5.1 CONCLUSÕES

Foi possível observar a diversidade das baterias de chumbo-ácido e suas aplicações que têm uma grande variedade de acordo com seu tipo construtivo. As baterias estacionárias suportam ciclos profundos de descarga, podendo fornecer energia por um longo período de tempo dependendo da sua corrente de descarga.

A partir das curvas de descarga reais de uma bateria é possível calcular o tempo restante de carga nela. Usando a metodologia de aproximação matemática da interpolação linear é possível realizar o cálculo de tempo para valores de correntes que têm suas curvas desconhecidas. A interpolação linear é uma metodologia de aproximação matemática satisfatória para curvas complexas, entretanto ela pode causar resultados errôneos quando não se tem dados suficientes para realizar essa aproximação.

O controle do conversor CC/CC pela corrente se mostrou suficiente permitindo realizar as descargas da bateria com corrente constante.

Com esse trabalho foi possível mostrar que o ARM7, o processador usado no microcontrolador LPC2368, é um bom processador para o uso em sistemas com resposta em tempo real, como por exemplo, um controle por PWM, devido à sua frequência de *clock* de até 72 MHz.

Além disso, é um dispositivo de fácil programação e configuração, sendo que apenas com a leitura de seu manual [8] foi possível programá-lo para realizar os cálculos necessários para o projeto.

## 5.2 TRABALHOS FUTUROS

Uma proposta para trabalhos futuros é testar as outras metodologias para estimação do SoC de baterias com o uso do ARM. Outra proposta é o uso do controle apresentado para gerar curvas de descarga de outros tipos de baterias.

Outro estudo que pode ser feito, é determinar um ganho que possa reduzir o erro da metodologia apresentada quando se tem dados iniciais de curvas distantes.

## 6 BIBLIOGRAFIA

- [1] Marcos Wilson Pereira Chagas, "Novas Tecnologias para Avaliação de Baterias," IEP/LACTEC, Curitiba, Tese de Mestrado 2007.
- [2] "Tecnologias aplicadas de armazenamento de energia," LACTEC, Curitiba, 2008.
- [3] Departamento Técnico da Unicoba, "Manual Técnico UNIPOWER Série UP Bateria Chumbo-Ácida Selada Regulada por Válvula," São Paulo, Manual Técnico 2003.
- [4] Jonhson Controls, "Manual Técnico Bateria estacionária FREEDOM," Sorocaba, Manual Técnico Julho 25, 2008.
- [5] Keila Kleveston Scheneider, Paulo Sausen, and Airam Sausen, "Análise comparativa de modelos analíticos para predição do tempo de vida de baterias em dispositivos móveis," UNIJUÍ, Ijuí, Tese de Mestrado 2010.
- [6] F. L. Grilo, O. P. Dias, I. M. Teixeira, and J. P. Teixeira, "Avaliação em Tempo Real da Autonomia da Baterias de Ácido-Chumbo," in *Terceiras Jornadas de Engenharia de Electrónica e Telecomunicações e de Computadores - JECT'05*, Lisboa, 2005.
- [7] Claudio Scherer, *Métodos computacionais da física*, 1st ed. São Paulo: Livraria da Física, 2005.
- [8] NXP, "UM10211 LPC23XX User Manual," User Manual 2009.
- [9] Ned Mohan, Tore Undeland, and William Robbins, *Power Electronics Converters, Applications and Design*, 2nd ed. New York: John Wiley & Sons. INC, 1995.

# Apêndice

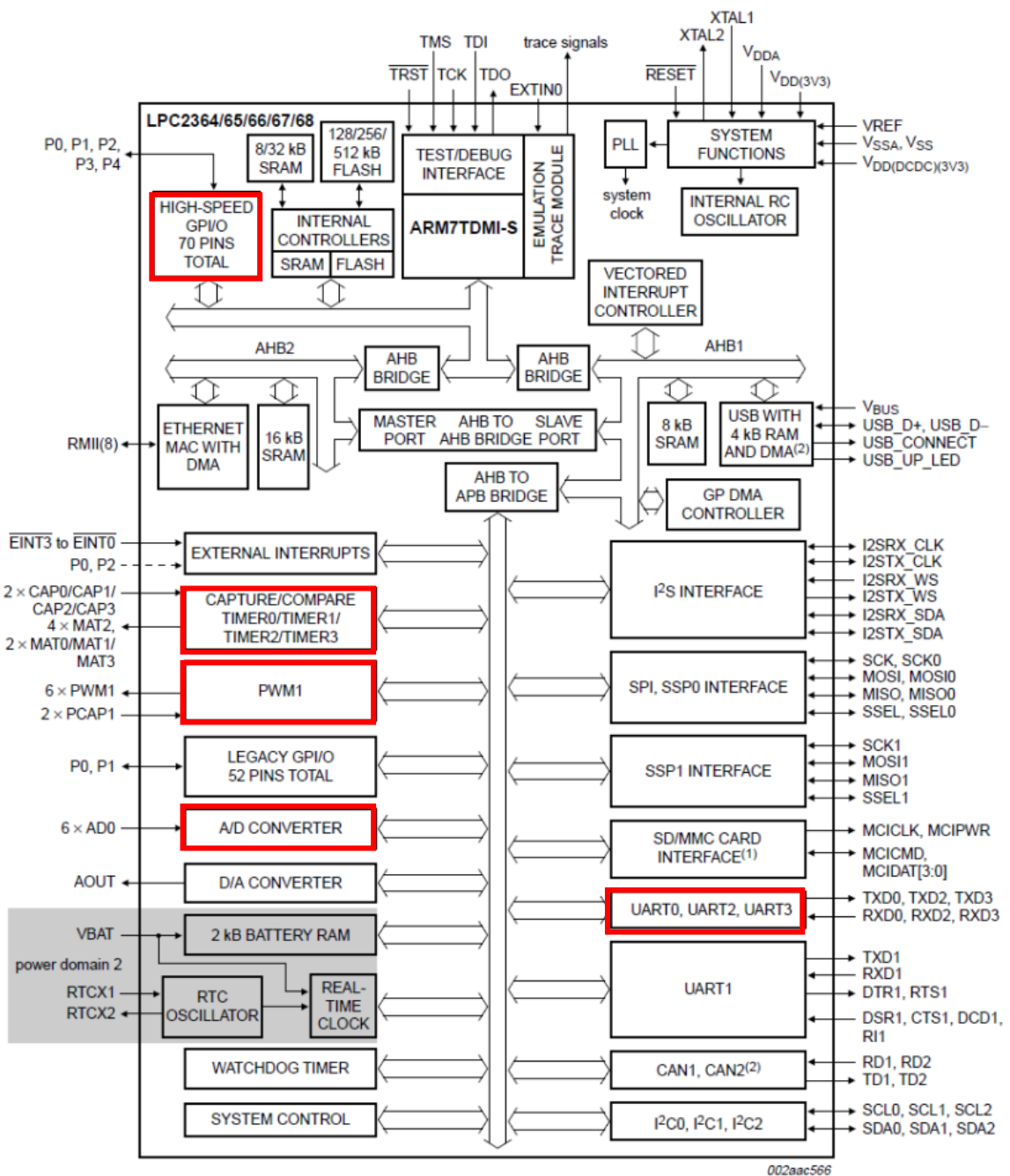
## 1 O ARM7 LPC2368

O ARM7 (*Advanced RISC Machine*) é um microprocessador do tipo RISC (*Reduced Instruction Set Computer*). A simplicidade na sua arquitetura é seu grande destaque, facilitando o entendimento e desenvolvimento de algoritmos. Neste trabalho, foi utilizado o microcontrolador LPC2368, um NXP da série LPC23XX. Segundo [8] ele tem um microprocessador ARM7 funcionando em até 72 MHz. E dispõe de até 512 kB de programa de memória flash no chip com ISP (*In-System Programming*) e IAP (*In-Application Programming*), e até 64 kB de SRAM no *local bus* do ARM para ter mais velocidade no acesso à CPU.

### 1.1 CONFIGURAÇÃO DOS PERIFÉRICOS

O microprocessador ARM7 LPC2368 conta com vários periféricos úteis para o projeto de monitoramento da descarga de bateria. A seguir serão apresentados os periféricos usados, e suas possíveis configurações. Na Figura 1.1 mostrada a seguir pode ser visto o diagrama de blocos do microprocessador, onde é possível ver todos os periféricos disponíveis no LPC2368, entretanto nem todos serão descritos neste capítulo. Como foi dito anteriormente, apenas os periféricos usados para o projeto receberão um destaque especial.

Os periféricos destacados em vermelho na Figura 1.1 serão apresentados nos próximos tópicos, eles são as entradas e saídas digitais GPIO's (*In, Out*), os conversores analógicos digitais ADC's (*Analog Digital Converter*), os contadores internos, o modulador por largura de pulso PWM e a interface de comunicação serial assíncrona UART (*Universal Asynchronous Receiver Transmitter*).



- (1) LPC2367/68 only.
- (2) LPC2364/66/68 only.

Figura 1.1: Diagrama de bloco do microprocessador LPC2368. Fonte: [8]

### 1.1.1 ENTRADAS E SAIDAS DIGITAIS– GPIO

As GPIO's são as entradas e saídas digitais do microprocessador, elas são as responsáveis por fazer a transmissão de dados entre dispositivos de hardware. Nessas entradas e saídas são reconhecidos apenas os níveis lógicos “1” ou “0”. Na entrada, o software reconhece quando é aplicada uma tensão acima de 3 V em um pino do hardware, com um bit de valor igual a “1”, e quando se tem aplicada uma tensão baixa, ou seja, com um valor abaixo dos 3V o sinal reconhecido é 0. Quando é usado como saída os sinais têm o mesmo significado, mas dessa vez é o software quem fornece os sinais, aplicando ou não uma tensão em um pino do hardware.

Essas portas podem ter várias funções, como por exemplo, controlar leds, acionar relés, detectar sinais de entrada, enviar dados para outros periféricos, etc.

No microprocessador LPC2368 estão disponíveis 70 pinos de entrada e saída, na realidade seriam 160 pinos, P0[0:31], P1[0:31], P2[0:31], P3[0:31] e P4[0:31] todos divididos com 32 portas GPIO, mas nem todas essas portas estão disponíveis para o usuário, restando 70 pinos, como foi mostrado na Figura 1.1. Esses 70 pinos estão espalhados entres os 32 pinos das cinco portas (P0, P1, P2, P3, P4), e dessas cinco portas, apenas as portas P0 e P1 são acessíveis através de grupos de registro, podendo ter recursos mais aprimorados e acesso acelerado ou como portas de acesso *legacy*. Já as portas P2, P3 e P4 só podem ser acessadas como portas aceleradas. As portas de acesso acelerado têm seus registradores realocados para o *local bus* do ARM tornando o tempo da I\O o mais rápido possível. Já as portas *legacy*, são as portas de acesso normal, ou portas lentas. Essas portas têm como vantagem compatibilidade com códigos de uma família anterior de microprocessadores.

As portas P0 e P2 geram interrupções para cada um de seus pinos, que funcionam durante o modo de baixa energia, pois elas são assíncronas e não dependem do *clock*, por isso elas também funcionam como despertador para o sistema, quando esse está funcionando em modo de baixa energia.

Para configurar as GPIO, são necessários os seguintes passos:

- Configurar o *clock*, no registrador PCLKSEL1;
- Selecionar os pinos usados, usando o registrador PINSEL0 até o PINSEL10 e os modos de operação pelos registradores PINMODE0 até PINMODE10;

- Se for usar a função de despertador das portas 0 e 2, essa configuração é feita usando o registrador INTWAKE;
- E para as interrupções, é usado o registrador IO0/2IntEnR ou o registrador IO0/2IntEnF.

Depois de configuradas as GPIO, elas têm várias facilidades para serem usadas, como por exemplo, várias mascaras de acesso, podendo ser acessadas como “*word*” de 32 bits, “*half-word*” de 16 bit ou por byte, 8 bits. Essas máscaras facilitam o acesso aos pinos das portas GPIO, e trabalhando-se com elas, evita-se que o estado de outros pinos seja mudado indevidamente.

Outra facilidade das GPIO no LPC2368 é a forma como seus registradores são divididos e apresentados. Na Tabela 1.1 é apresentado o mapa dos registradores de acesso acelerado das GPIO, pois esses registradores valem para todas as portas GPIO.

Tabela 1.1: Descrição dos registradores das portas GPIO com uso de acesso rápido

Registrador	Descrição
<b>FODIR</b>	Controla a direção individualmente das GPIO rápidas, quando 1 o pino está como saída, quando 0 está como entrada.
<b>FIOMASK</b>	Máscara de acesso rápido aos pinos, nela é possível escrever, atribuir valor, limpar e ler os sinais das portas. Ela altera ou retorna os bits ativados com zero.
<b>FIOPIN</b>	Altera o estado dos pinos para 1 ou 0, e também pode ser lido para saber qual o estado atual dos pinos.
<b>FIOSET</b>	Também controla o estado dos pinos, quando 1, transforma o sinal no pino em 1, quando 0 não tem efeito nenhum no sinal do pino.



**FIOCLR**

Tem a função parecida com a do FIOSET, mas nesse caso quando 1, transforma o sinal do pino em 0, e quando 0 não tem efeito no sinal do pino.

### 1.1.2 CONVERSOR ANALÓGICO-DIGITAL - ADC

Muitas vezes, não precisamos apenas saber o estado dos sinais digitais, como 0 ou 1, com ou sem tensão, mas também precisamos saber a intensidade desses sinais, quando se deseja fazer alguma medição. Para isso existem os conversores de sinais analógicos para digitais, os ADC's (para simplificação serão chamados de AD's) que fazem a conversão do sinal analógico, um valor de tensão, para um sinal digital, uma combinação de bits.

No caso do LPC2368 essa conversão é feita para um valor de 10bits, ou seja, para uma tensão de 0 V, o AD produz como resultado o valor 0, e a máxima tensão na entrada do AD é interpretada como 1023 (máximo valor de 10 bits:  $2^{10}-1 = 1023$ ). A variação da tensão medida na entrada é de 0 a 3 V, e o tempo que o microprocessador leva para realizar a conversão, varia em torno de 2,4 $\mu$ s.

O LPC2368 conta com seis canais de conversão AD e para ativá-los basta seguir os passos abaixo:

- Ligar os bits PCADC no registrador PCONP;
- Configurar o *clock* nos bits PCLK\_ADC do registrador PCLK\_SEL0. Para dividir o *clock* para o ADC, só alterar os bits CLKDIV;
- Selecionar os pinos do ADC e o modo de operação dos pinos nos registradores PINSELn e PINMODEn;

Para fazer os AD funcionarem, é preciso ativar o bit 12, PCAD, do registrador PCONP, mas esse bit funciona junto com o bit 21, PDN, do registrador AD0CR, que é o registrador de controle dos AD. Antes de limpar o bit PCAD, é necessário primeiro limpar o bit PDN, e antes de ativar o PDN, deve-se ativar primeiro o PCAD. O bit PDN é que define se o conversor AD está operando, ou em modo de energia baixa.

Depois de ligado o conversor, é possível fazer as outras configurações para o conversor funcionar, uma delas é selecionar a frequência do *clock*, o registrador PCLK\_SELO nos bits 24 e 25, PCLK\_ADC, os possíveis valores que esse bit pode receber são mostrados na Tabela 1.2:

Tabela 1.2: Possíveis valores para os bits do registrador de divisão do *clock* dos periféricos.

Valores dos bits	Função
<b>00</b>	<i>Clock/4</i>
<b>01</b>	<i>Clock</i>
<b>10</b>	<i>Clock/2</i>
<b>11</b>	<i>Clock/8</i>

Outra configuração necessária é selecionar os pinos que serão usados como entrada para o sinal analógico, em um dos registradores PINSEL. Os AD's 0, 1, 2 e 3 são selecionados no registrador PINSEL1 e os AD's 4 e 5 são selecionados no registrador PINSEL3.

O registrador AD0CR, é o registrador de controle dos AD's, nos seus bits é definido o início, e qual das seis entradas de AD será lida para a próxima conversão. Além disso, é também nos bits do registrador AD0CR que é definida a frequência de *clock* independente do AD que ocorrerá a conversão.

Após dado o início à conversão, é o registrador AD0GDR (GDR - *Global Data Register*) de 32 bits, que contém as informações da última conversão que foi realizada, entre elas estão:

- o resultado da conversão,  $V/V_{Ref}$  em 10 bits;
- o bit *DONE*, indicando que a conversão foi terminada;
- o bit *OVERRUN*, onde é possível saber se houve conversão não realizada. .

### 1.1.3 MODULAÇÃO POR LARGURA DE PULSO - PWM

Outro periférico de grande importância é o gerador de sinal modulado por largura de pulso, o PWM, esse é um tipo de sinal onde a amplitude e a frequência se mantêm constante e o que varia é a largura do pulso de acordo com o valor do sinal de referência. O sinal PWM é resultado de uma comparação entre uma onda triangular ou “dente de serra” com um sinal de referência, como pode ser visto na Figura 1.2.

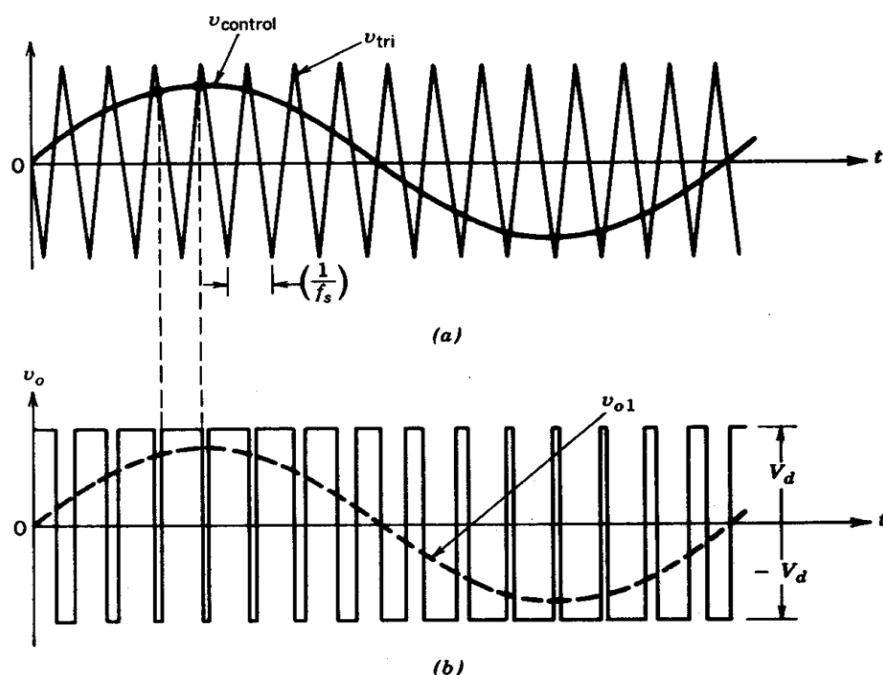


Figura 1.2: (a) Sinais de comparação para gerar o sinal PWM. (b) Sinal PWM resultante comparado com o sinal modulador. Fonte:[9].

O sinal PWM apresentado na Figura 1.2(a) é um exemplo para modulação de valor de tensão onde o sinal de referência é a onda senoidal  $v_{control}$  que é comparada com uma onda triangular  $v_{tri}$ . Pode-se notar na Figura 1.2(b) que a largura do pulso varia de acordo com o valor instantâneo da onda senoidal: na parte mais alta da onda, o pulso tem uma largura de quase 100% do período, e nas partes mais baixas da senóide, a largura do pulso é quase 0% do período.

No ARM existem seis pinos de saída para sinal PWM, o período e a largura do pulso podem ser qualquer número de contagem do timer, permitindo uma grande flexibilidade de

resolução para o sinal. Além disso, ele conta com sete comparadores, sendo o comparador 0 o comparador usado para reiniciar o contador, os outros seis, são os comparadores usados para gerar os sinais de saída em cada pino de PWM.

Esses sete comparadores permitem a geração de seis sinais do tipo *single edge*. Os sinais *single edge* iniciam o seu pulso no comparador 0, quando o contador é reiniciado, e terminam o pulso quando ocorre o encontro do valor da contagem com o valor definido no seu respectivo comparador. Existe também a possibilidade da geração de 3 sinais *double edge*, onde quem define o começo do pulso é o comparador anterior, por exemplo, se o pino 3 for usado como *double edge* o início do seu pulso será definido pelo comparador 2, e o final pelo comparador 3.

O funcionamento do PWM pode ser exemplificado pela Figura 1.3 junto com a Tabela 1.3.

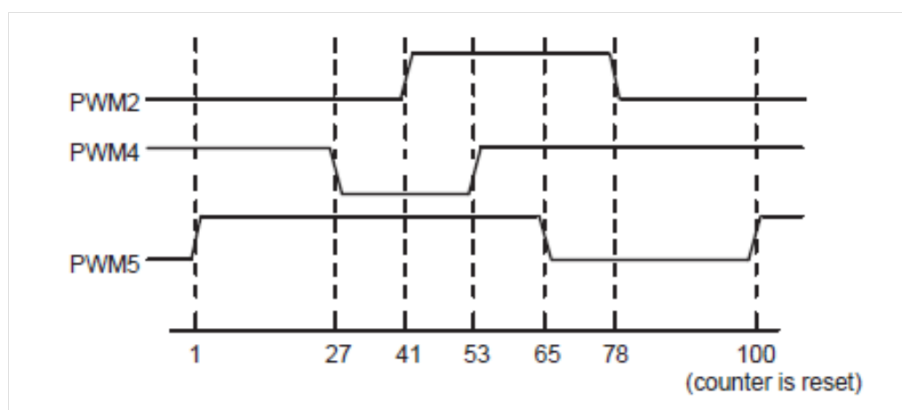


Figura 1.3: Exemplo de diferentes sinais PWM gerados no ARM7 LPC2368. Fonte: [8]

Tabela 1.3: Valores usados nos comparadores do PWM, no exemplo da Figura 1.3.

Comparador	Valor (em ciclos de <i>clock</i> )
<b>0</b>	100
<b>1</b>	41
<b>2</b>	78
<b>3</b>	53
<b>4</b>	27
<b>5</b>	65

Os sinais de PWM apresentados na Figura 1.3 seguem a seguinte configuração para os comparadores, e pinos de saída.

Foram escolhidos como pinos de saída, os pinos PWM2, PWM4 e PWM5, sendo que os pinos PWM2 e PWM4, foram configurados para sinais *double edge* e o pino PWM5 como *single edge*. Na Tabela 1.3 são apresentados os valores definidos para cada comparador. Como pode se visto o valor do comparador 0 do exemplo é definido com o valor 100, ou seja, quando a contagem chegar a 100 o contador é reiniciado e os pinos com os sinais definidos como *single edge* têm os seus pulsos iniciados. No exemplo só o PWM5 é definido como *single edge* e o valor de seu comparador é definido como 65, que é o valor onde o seu pulso termina como pode ser visto na Figura 1.3.

Ainda observando a Figura 1.3 e a Tabela 1.3 pode ser visto que os PWM2 e PWM4 foram definidos como *double edge*, logo os sinais de seus pulsos começam quando a contagem chega nos valores dos comparadores 1 e 3 e termina quando a contagem chega nos valores dos comparadores 2 e 4 respectivamente.

Para poder configurar e fazer funcionar o PWM, basta seguir os seguintes passos:

- Ligar o PWM no registrador PCONP selecionando o bit PCPWM1;
- Selecionar o *clock* do periférico nos bits PCLK\_PWM do registrador PCLK\_SEL0;
- Selecionar os pinos do PWM e seu modo de operação nos registradores PINSELn e PINMODEn;
- Para ativar a interrupção por comparação, usar o registrador PWM1MCR e para gerar a interrupção por captura de eventos, usar o comparador PWM1CCR.

Como todos os outros periféricos, uma das primeiras coisas a se fazer para usar o sinal de PWM, é selecionar os pinos que serão usados como saída de sinal PWM no registrador PINSELn ( $0 \leq n \leq 10$ ). Porém não é preciso ativar o PWM no registrador PCONP, pois os bits que ligam o PWM, já começam ligados quando o microprocessador é reiniciado.

Depois pode também ser configurada a divisão do *clock* para o PWM, alterando os valores dos bits PCLK\_PWM no registrador PCLK\_SEL0 de acordo com a Tabela 1.2. Antes de configurar o PWM, é preciso colocar 1 no segundo bit do registrador PWM1TCR (TCR -

*Timer Control Register*), esse bit é o *Counter Reset*. Esse registrador que é usado para controlar o modo de operação do contador de tempo do PWM.

O registrador PWM1PR (PR - *Prescale Register*), é onde é definido o valor para a contagem básica para o PWM. De acordo com os ciclos de *clock*, o contador de tempo (TC - *Timer Counter*) é incrementado a cada valor de PR+1, por exemplo, para PR = 0, a cada 1 ciclo de *clock* o contador de tempo do PWM é incrementado de 1, ou seja, o contador de tempo do PWM andaria junto com o *clock* definido, e se fosse definido um PR = 9, a cada 10 ciclos de *clock* o contador do PWM seria incrementado de 1, ou seja, o seu contador seria 1/10 do *clock*. Esse modo de operação do PWM é configurado no registrador PWM1CTCR (CTCR - *Count Control Register*), nos bits 1 e 0. O PWM vai funcionar como descrito acima quando estiver o modo TIMER, pois ele ainda tem mais três modos COUNTER onde o contador de tempo é controlado por eventos externos, e não pelo número de ciclos de *clock* definido no PR.

A configuração dos pinos de saída para *single edge* ou *double edge*, como foi explicado anteriormente, é feita no registrador PWM1PCR (PCR - *Control Registers*). Além de configurar para *single* ou *double edge*, é nesse registrador que são habilitados os canais de saída do PWM.

Depois de configurados os canais do PWM, e habilitados para saída de sinal, os valores dos comparadores são definidos nos registradores PWM1MRx (MR - *Match Register* e "x" - número do comparador de 1 a 6) e no registrador PWM1MCR (MCR - *Match Control Register*) é onde pode ser definido se haverá uma interrupção, se o contador de tempo vai ser reiniciado ou parado, quando o comparador chegar ao valor configurado no MRn.

Caso o valor do comparador precise ser atualizado após o PWM ser configurado e o programa começar a rodar, é preciso habilitar o registrador PWM1LER (LER - *Latch Enable Register*). Nesse registrador é possível habilitar qual comparador do PWM se deseja alterar durante o programa.

Feitas todas as configurações do PWM, basta ativá-lo e iniciar seu contador, mudando o valor dos bits 0, 1 e 3 no PWM1TCR.

#### 1.1.4 CONTADORES INTERNOS

O microprocessador ARM7 LPC2368 conta com quatro contadores internos, onde é possível criar interrupções controladas. Esses contadores podem ser usados para contar ciclos de *clock*, e realizar alguma ação após um determinado número de ciclos, definido no comparador. Com essa configuração são chamados de *TIMER* ou podem capturar eventos externos e assim são chamados de *COUNTER*.

Para configurar os contadores internos, segue-se o mesmo padrão dos outros periféricos:

- Ligar o Timer no registrador PCONP selecionando o bit PCTIMn (n pode ser 0, 1, 2 ou 3);
- Selecionar o *clock* do periférico nos bits PCLK\_TIMERn (n = 0 ou 1) no registrador PCLK\_SEL0 e os bits PCLK\_TIMERn (n = 2 ou 3) no registrador PCLK\_SEL1;
- Selecionar os pinos do TIMER e seu modo de operação nos registradores PINSELn e PINMODEn;
- Para ativar a interrupção, usar o registrador TIMERnMCR (n pode ser 0, 1, 2 ou 3) e para gerar a interrupção por captura de eventos, usar o comparador TIMERnCCR (mesmos valores e n).

Dos quatro *Timers*, o *Timer 0* só conta com dois comparadores, o *Timer 1* tem três comparadores e os *Timers 2 e 3* têm quatro comparadores. Os comparadores são de 32 bits e permitem uma operação contínua, podendo gerar uma interrupção quando ocorrer igualdade entre o valor do comparador e o valor do contador, ele também pode parar ou reiniciar a contagem e em cada um deles gerar uma interrupção no programa.

Para habilitar ou reiniciar os contadores basta configurar o registrador TnTCR (TCR – *Timer Control Register*, n é o número do Timer, 0, 1, 2 ou 3). O bit 0 do TnTCR habilita ou desabilita o contador, e o bit 1 reinicia o contador. Para poder definir o valor do PR (*prescale*) do contador é necessário primeiro desabilitá-lo. Após definir o PR, o contador pode ser novamente habilitado.

Os contadores seguem o mesmo tipo de funcionamento do PWM descrito no item 1.1.3 deste capítulo. Ou seja, é preciso definir um valor para o PR de cada Timer, e o TC é incrementado a cada PR+1 ciclos de *clock*, assim como no PWM, e o valor de PC (*Prescale Counter*) segue a contagem do ciclo de *clock* até o valor definido no PR, e quando alcança esse valor ele é restado. Também é preciso definir os valores dos seus registradores de comparação MR, sendo que o contador tem no máximo quatro MR e não seis como no PWM. Os valores dos MR são definidos nos registradores TnMRx (x é o número do comparador, que varia de 0 à 3). A função dos MR, é definida no registrador TnMCR (MCR – *Match Control Register*), as possíveis funções são as que foram descritas no final do parágrafo anterior.

Para gerar uma interrupção no programa usando o contador, é necessário configurar o registrador TnIR (IR – *Interrupt Register*), nele é definido quais comparadores vão gerar a interrupção.

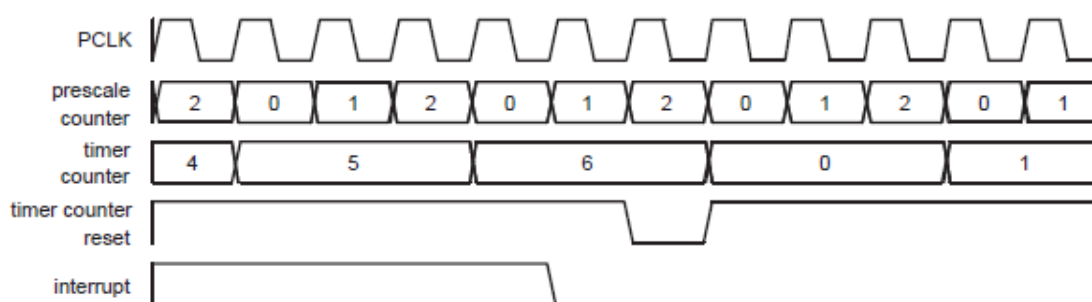


Figura 1.4: Exemplo de ciclo de Timer, onde PR = 2, MRx = 6 e a interrupção e o reset estão ativos no encontro com o comparador.

O exemplo da Figura 1.4, mostra o valor de PR = 2, logo PC conta até 2 e é reiniciado, e quando o PC é reiniciado é o mesmo momento em que começa o PR+1, ou seja o 3º ciclo de *clock*, então o TC é incrementado. Como o comparador MRx foi definido com o valor 6, e para o encontro, foi definido no TnMCR, para ocorrer uma interrupção e reiniciar o contador, logo quando TC chega ao valor 6, ocorre a interrupção, e nesse momento quando termina o ciclo do PR, o TC é reiniciado.



### 1.1.5 COMUNICAÇÃO UART

A UART é um periférico do microprocessador, que converte dados de comunicação serial para a forma de bits em paralelo e vice-versa. Na comunicação serial os dados são enviados para entre o receptor e o transmissor em sequência, bit a bit, bastando apenas um par de fios para realizar a comunicação. Já na comunicação paralela, os dados são enviados byte a byte, ou em quantidade maior de bits de uma só vez, mas para isso é necessário uma quantidade maior de fios, do que na comunicação serial.

Antes de cada grupo de bits de dados enviados, é primeiro enviado o bit de partida, que é um bit de valor lógico 0. Ao final é enviado o bit de parada, com valor lógico 1, e antes do bit de parada pode ser enviado também o bit de paridade. O tamanho do pacote de dados que é enviado pode ser programado via software, variando de 5 a 8 bits.

Algumas UART contam ainda com um buffer de memória FIFO (*First In, First Out*) antes do seu receptor, onde certa quantidade de dados vai ficando armazenada para evitar perda de dados, quando ocorre uma transmissão com quantidade elevada de dados.

No LPC2368 existem quatro UART, que contam com uma FIFO de 16 bytes, e também têm o controle de fluxo da sua taxa de transferência ajustável por software ou hardware. Sendo as UART, 0, 2 e 3 iguais, a UART1 é a única diferente pois conta com uma interface para modem.

A seguir é apresentada a sequência de inicialização da UART:

- Ligar a UART no registrador PCONP selecionado o bit PCUARTn ( n = 0, 2 ou 3);
- Selecionar o *clock* do periférico no registrador PCLK\_SEL0 nos bits PCLK\_UART0 e no registrador PCLK\_SEL1 nos bits PCLK\_UART2 e PCLK\_UART3;
- Configurar a taxa de transferência da UART, no registrador UnLCR (n = 0, 2 ou 3) ativar o bit DLAB. Isso libera o acesso aos registradores DLL e DLM, para poder configurar a taxa de transferência.
- Para ativar a memória FIFO, ativar o bit 0 do registrador U0FCR;
- Selecionar os pinos da UART e seu modo de operação nos registradores PINSELn e PINMODEn;

- Para ativar a interrupção, primeiro deve-se colocar 0 no bit DLAB no registrador UnLCR ( $n = 0, 2$  ou  $3$ ). Isso permite o acesso ao registrador UnIER ( $n = 0, 2$  ou  $3$ ).

A UART0 é ligada no bit 3, a UART1 no bit 4, a UART2 no bit 24 e a UART3, no bit 5 do registrador PCONP, sendo que as UART 0 e 1, não precisam ser ligadas por software, pois, como o PWM, já estão ligadas após o microprocessador ser iniciado. Já as UART 2 e 3, necessitam ser ligadas nesse registrador.

Para as UART também pode ser definido um divisor de *clock* nos registradores PCLK\_SEL0 e PCLK\_SEL1. Nos bits 2 e 3 e nos bits 4 e 5 do registrador PCLK\_SEL0 são definidos os divisores de *clock* para as UART 0 e 1, respectivamente, e no registrador PCLK\_SEL1 nos bits 12 e 13 e bits 14 e 15 para as UART 2 e 3. Os possíveis valores para esses bits e suas configurações são as mesmas apresentadas na Tabela 1.2.

Como já foi mostrado na configuração de outros periféricos, é no registrador PINSELn que são definidas as funções dos pinos do microprocessador, e para as UART existem diversas opções de pinos para serem usados como Tx (*Transmit*) e Rx (*Receive*), que podem ser configurados em diferentes registradores PINSELn.

As FIFO são habilitadas e reiniciadas no registrador UnFCR ( $n = 0, 1, 2$  ou  $3$ ; FCR – *FIFO Control Register*), é também nesse registrador que o nível do *trigger* do Rx é definido. No registrador UnLCR (LCR – *Line Control Register*) é onde pode ser definido o tamanho do pacote de dados que será enviado via UART, que como dito anteriormente pode variar de 5 a 8 bits, quantos bits de parada serão enviados, se será ou não enviado o bit de paridade, o tipo de paridade, o controle de parada, que força a UART a não transmitir enquanto esse bit estiver ativo e o bit DLAB (*Divisor Latch Access Bit*).

Quando o bit DLAB está ativo (definido com valor 1), é permitida a configuração da taxa de transferência de dados (*baud rate*) das UART, mudando os valores dos registradores UnDLL (DLL – *Divisor Latch LSB Register*) e UnDLM (DLM – *Divisor Latch MSB Register*) segundo a equação (6.1).

$$UARTn_{\text{boudrate}} = \frac{pclk}{16 \times (256 \times UnDLM + UnDLL)} \quad (6.1)$$

Depois de definidos os valores para DLL e DLM, para uma determinada taxa de transferência, o bit DLAB deve ser desativado (definindo valor 0), para tornar possível o acesso ao registrador UnRBR (RBR – *Receive Buffer Register*), registrador onde fica o pacote de dados mais antigo recebido na Rx FIFO. Como a FIFO do LPC2368 tem 16 bytes, e os dados só podem ser enviados em pacotes de no máximo 8 bits, ela consegue armazenar mais de um pacote de dados.

Outro motivo importante para zerar o bit DLAB após a configuração do valor da taxa de transferência, é para também poder transmitir um pacote de dados pelo registrador UnTHR (THR – *Transmit Hold Register*). O registrador UnTHR contém os dados mais recentes adicionados na Tx FIFO, para enviar.

Outro registrador de grande importância para o uso das UART é o UnLSR (LSR – *Line Status Register*), que contém as informações dos status do transmissor e do receptor. Esse registrador pode ser apenas lido pelo programa, e seus bits contêm as informações mostradas na Tabela 1.4.

Tabela 1.4: Mapa de bits do registrador UnLSR

Bit	Nome	Função
0	RDR ( <i>Received Data Ready</i> )	Serve para avisar quando tem algum dado válido não lido na UnRBR
1	OE ( <i>Overrun Error</i> )	Avisa quando ocorre um erro de <i>overrun</i> , ou seja, no registrador UnRSR já tem algum pacote de dados montado e a UnRBR FIFO ainda está cheia; quando isso acontece, o dado na UnRSR é perdido.
2	PE ( <i>Parity Error</i> )	Está ativo quando o bit de paridade recebe um dado no estado errado.
3	FE ( <i>Framing Error</i> )	Ocorre quando o bit de parada está definido como 0.
4	BI ( <i>Break Interrupt</i> )	
5	THRE ( <i>Transmitter Holding Register Empty</i> )	Avisa quando a UnTHR está vazia.
6	TEMT ( <i>Transmitter Empty</i> )	Avisa quando a UnTHR e a UnTSR estão vazias.
7	RXFE ( <i>Error in Rx FIFO</i> )	Avisa quando algum dado na UnRBR chega com qualquer um dos erros de Rx, como o FE, PE ou para o caso de BI.

Com as informações dos bits 1 e 5 conforme descrito na Tabela 1.4, fica possível realizar a transmissão de dados evitando que ocorram erros, apenas aguardando o transmissor livre com a informação do THRE . A chegada de um dado válido pode ser verificada checando o bit RDR.

## 1.2 INTERFACE HOMEM MÁQUINA - IHM

A interface entre o usuário e o microprocessador pode ser feita de várias maneiras, uma delas é o uso de um display onde podem ser expostas as informações do sistema em tempo real, como tensão, corrente e tempo de bateria.

Neste trabalho, foi utilizado um display tipo GLCD (*Graphic Liquid Crystal Display*) controlado pelo microprocessador T6963 da Toshiba. A comunicação entre o display e o microprocessador ARM é feita pelas portas IO (*In Out*). Para utilizar o display são necessárias oito portas do tipo IO para a transferência de dados e seis IO para controle e programação do display, ou seja, são necessárias 14 IO. Como apresentado no item 4.1.1, no ARM estão disponíveis 70 IO, viabilizando a comunicação da IHM com o microprocessador.

Na Tabela 1.5 é apresentado um resumo da pinagem do display e suas funções.

Tabela 1.5: Pinagem do display GLCD

Nº	Nome	Função
1	FGND ( <i>Frame Ground</i> )	Terra do display (aterramento da carcaça).
2	V <sub>SS</sub>	Terra para o circuito lógico (referência de tensão zero).
3	V <sub>DD</sub>	Entrada de alimentação para o circuito lógico.
4	V <sub>0</sub>	Entrada de alimentação para o circuito do display.
5	WR ( <i>Write</i> )	Para mandar um byte para o display.
6	RD ( <i>Read</i> )	Para ler um byte do display.
7	CE ( <i>Chip Enable</i> )	Habilita o chip.
8	C/D ( <i>Comand/Data</i> )	Seleciona se o byte enviado para o display vai ser um

		comando ou um dado.
<b>9</b>	$V_{EE}$	Entrada de alimentação de tensão negativa
<b>10</b>	RES ( <i>Reset</i> )	Reinicializa o display
<b>11</b>	DB0 ( <i>Data Bus Line 0</i> )	Pino de dado que recebe o bit menos significativo (LSB)
<b>12</b>	DB1 ( <i>Data Bus Line 1</i> )	Pino de dado intermediário
<b>13</b>	DB2 ( <i>Data Bus Line 2</i> )	Pino de dado intermediário
<b>14</b>	DB3 ( <i>Data Bus Line 3</i> )	Pino de dado intermediário
<b>15</b>	DB4 ( <i>Data Bus Line 4</i> )	Pino de dado intermediário
<b>16</b>	DB5 ( <i>Data Bus Line 5</i> )	Pino de dado intermediário
<b>17</b>	DB6 ( <i>Data Bus Line 6</i> )	Pino de dado intermediário
<b>18</b>	DB7 ( <i>Data Bus Line 7</i> )	Pino de dado que recebe o bit mais significativo (MSB)
<b>19</b>	FS ( <i>Font Select</i> )	Seleciona a fonte do display entre 6x8 e 8x8
<b>20</b>	N.C.	Não conectado

Para realizar a comunicação do display com o ARM, foram definidas algumas máscaras para as GPIO com os nomes dos pinos em que elas estão conectadas no display. Isso facilitou a comunicação do ARM com o display.

Com isso foi possível fazer o controle e ao mesmo tempo monitorar a corrente de saída, o tempo restante de carga para aquela corrente e a tensão atual na bateria.

