

EDSON RODRIGO SCHLOSSER

**IMPLEMENTAÇÃO DE UM PWM DIGITAL DE ALTA
RESOLUÇÃO E FREQUÊNCIA UTILIZANDO CIRCUITO DE ATRASO**

Trabalho de conclusão de curso apresentado como parte das atividades para obtenção do título de Engenheiro Eletricista, do curso de Engenharia Elétrica do Campus Alegrete da Fundação Universidade Federal do Pampa.

Orientador: Alessandro G. Girardi


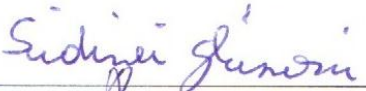

Co-Orientador: Marcos V. T. Heckler

**Alegrete
2011**

Autoria: Edson Rodrigo Schlosser

Título: Implementação de um PWM Digital de Alta Resolução e Frequência utilizando Circuito de Atraso

Trabalho de conclusão de curso apresentado como parte das atividades para obtenção do título de Engenheiro Eletricista, do curso de Engenharia Elétrica do Campus Alegrete da Fundação Universidade Federal do Pampa.

Os componentes da banca de avaliação, abaixo listados, consideram este trabalho aprovado.				
	Nome	Titulação	Assinatura	Instituição
1	Alessandro Gonçalves Girardi	Doutor		Universidade Federal do Pampa
2	Sidinei Ghissoni	Mestre		Universidade Federal do Pampa
3	Marcos Vinicio Thomas Heckler	Doutor		Universidade Federal do Pampa

Data da aprovação: 20 de Junho de 2011.

“Dedico este trabalho à minha família, em especial aos meus pais, Maria e Adil Schlosser e às minhas irmãs, Carla e Carmem pelo apoio e carinho”.

AGRADECIMENTOS

Primeiramente agradeço a Deus pela vida e sabedoria me concedida.

Agradeço ao CNPEM, em especial ao Grupo de Eletrônica de Potência do Laboratório Nacional de Luz Síncrotron que me selecionou para o 20º Programa Bolsa de Verão. Durante o programa tive a oportunidade de desenvolver este projeto envolvendo engenharia de aceleradores, onde levarei a experiência adquirida para toda minha vida pessoal e profissional.

Agradeço aos profissionais do Grupo de Eletrônica de Potência do Laboratório Nacional de Luz Síncrotron, em especial ao Eng. Cleber Rodrigues, Eng. Luís Henrique de Oliveira e ao Eng. Ângelo Romeu da Silva pelo projeto proposto e pelos conhecimentos passados a mim.

Agradeço aos professores e aos amigos da Universidade Federal do Pampa.

*“A mente que se abre a uma nova idéia jamais
voltará ao seu tamanho original”*

Albert Einstein

RESUMO

O presente trabalho apresenta o projeto de um PWM digital de alta resolução e frequência tendo como base um circuito de atraso, que fará parte de um sistema de regulação digital das fontes de corrente do Laboratório Nacional de Luz Síncrotron. As fontes de corrente são responsáveis por alimentarem diversos equipamentos que compõem o anel de armazenamento de elétrons, tais como dipolos, quadruplos, sextupolos e bobinas corretoras de órbita.

Os sistemas de regulação analógica utilizados tradicionalmente nos sistemas de regulação estão susceptíveis a ruídos, bem como expostos a variação da temperatura ambiente, responsáveis pela alteração de grandezas elétricas que causam perda de precisão do sistema. Os sistemas de regulação digital, como o proposto neste trabalho, apresentam algumas vantagens sobre os sistemas analógicos, tais como versatilidade, facilidade de ajustar parâmetros, entre outros.

Neste projeto foi utilizado um microcontrolador que possui internamente um módulo de PWM com 10 bits de resolução e frequência de chaveamento em dezenas de kHz que, após passar por um circuito de atraso, pode atingir 18 bits de resolução, permanecendo a frequência de chaveamento constante.

Para testes, foi desenvolvida uma interface computacional em Matlab, de modo a facilitar a comunicação do usuário com o hardware. Utilizou-se a comunicação serial, conforme padrão RS232, que envia informações do software para o microcontrolador. Este recebe os dados, processa e, posteriormente atua sobre uma chave através da modulação PWM. O projeto foi prototipado e implementado sobre um IGBT de forma a alimentar uma carga do tipo RL. Com um DCCT foram realizadas medidas da variação de corrente em função da variação do tempo em que a chave permanecia ligada, a fim de se comprovar a resolução do sistema. Os resultados obtidos são apresentados e analisados, validando o projeto proposto.

Palavras-chave: Fonte de Corrente, Regulação Digital, PWM e Circuito de Atraso.

ABSTRACT

This paper presents the design of a high resolution and high frequency digital PWM based on a delay circuit, which will be part of a digital control system that could replace in future the analog control system presently used in the power supplies of the National Laboratory of Synchrotron Light. These power supplies are responsible for driving many equipment of the electron storage ring, such as bending magnets, quadrupoles, sextupoles and steering magnets.

Analog control systems are susceptible to noise and exposed to ambient temperature variations. Both are responsible for the change of electrical parameters that cause loss of accuracy. Digital control systems have some advantages over analog systems, such as versatility, ease of adjusting parameters, among others.

In this project we used a microcontroller that has an internal PWM module with 10-bit resolution and switching frequency in the tens of kHz. After passing through a delay circuit, it can achieve 18-bits of resolution and the frequency remains constant.

In order to test the control system, we developed a computer interface in Matlab to facilitate communication between the user and the hardware. We used standard RS232 as serial communications. The software sends information to the microcontroller, receives data, process it, and then acts on a switch by PWM. The design was prototyped and implemented on an IGBT in order to drive an RL load. With a DCCT we measured the variation of current as a function of the variation in the time that the switch remained on, in order to prove the system resolution. The results are presented and analyzed, validating the proposed design.

Keywords: Power Supplies, Digital Control, PWM and Circuit Delay.

LISTA DE ILUSTRAÇÕES

FIGURA 1- Foto panorâmica do anel de armazenamento de elétrons do LNLS.....	15
FIGURA 2 - Sistema de regulação analógica utilizado atualmente no LNLS.	16
FIGURA 3 – Esquema de um sistema de regulação digital de uma fonte de corrente.	17
FIGURA 4 – Esquema do sistema de regulação digital projeto.....	18
FIGURA 5 - Atraso inserido ao sinal PWM.	19
FIGURA 6 - Inserção de atrasos entre dois passos do PWM.....	19
FIGURA 7 - Características da modulação PWM.	20
FIGURA 8 - PWM obtido a partir de um circuito analógico.	21
FIGURA 9 - PWM obtido a partir de um circuito digital.	22
FIGURA 10 - Comunicação paralela de 8 bits entre um emissor e um receptor.	23
FIGURA 11 - Comunicação serial entre um emissor e um receptor.....	24
FIGURA 12 - Pacote de dados com magnitude de 10bits enviados no padrão RS232.....	26
FIGURA 13 – Pinagem do circuito de atraso DS1020.....	26
FIGURA 14 - Diagrama esquemático interno do DS1020.....	28
FIGURA 15 – Definição do efeito jitter em um sinal de clock.....	29
FIGURA 16 - Interface do sistema de comunicação desenvolvida em Matlab.....	31
FIGURA 17 - Funcionamento da interface de comunicação entre o computador e o microcontrolador do sistema de controle digital.	32
FIGURA 18 - Conversor MAX232.....	32
FIGURA 19 – Esquema de ligação do conversor MAX232.	33
FIGURA 20 – Aplicação do conversor MAX232 entre hardware e computador.	33
FIGURA 21 – Pinagem do microcontrolador PIC 16F877.....	34
FIGURA 22 - Arquitetura interna ao microcontrolador PIC 16F877.	35
FIGURA 23 - Módulo de transmissão de dados serialmente do microcontrolador.	37
FIGURA 24 - Módulo de recepção de dados serialmente do microcontrolador.	39
FIGURA 25 - Módulo PWM do microcontrolador PIC16F877.	41
FIGURA 26 - Funcionamento do módulo PWM do microcontrolador PIC16F877.....	42
FIGURA 27 - Aproximação linear dos valores de atraso.....	46
FIGURA 28 - Porta Lógica OU.....	47
FIGURA 29 – Diagrama de funcionamento do microcontrolador.....	48

FIGURA 30 – Detalhamento da etapa de recebimento das informações pelo microcontrolador	49
FIGURA 31 – Detalhamento da etapa de verificação dos dados recebidos.....	50
FIGURA 32 – Detalhamento da etapa da aproximação linear realizada no pacote de atraso..	51
FIGURA 33 – Detalhamento da etapa de atualização do PWM.	52
FIGURA 34 - Diagrama esquemático do circuito proposto.....	54
FIGURA 35 – Imagens do protótipo desenvolvido.....	53
FIGURA 36 – Sinal de saída para alguns valores de duty cycle atribuídos.....	55
FIGURA 37 – Sinal de saída obtido a partir de 10 bits de resolução do PWM.	56
FIGURA 38 - Sinal de saída obtido a partir de 18 bits de resolução do PWM.....	57
FIGURA 39 – Saída do PWM com passos de $0,25\eta_s$	57
FIGURA 40 – Imagens do driver, IGBT e Carga RL (Bobina Corretora de Órbita e Resistências) utilizados para teste.	58
FIGURA 41 - Variação da corrente em função da variação do duty cycle.	60
FIGURA 42 - Detalhes da variação de corrente em função da variação do duty cycle.	61
FIGURA 43 – Detalhes da corrente de saída sem monotonicidade.	62
FIGURA 44 - Corrente de saída com monotonicidade após correção.	63

LISTA DE TABELAS

TABELA 1 - Família DS1020.....	27
TABELA 2 - Registrador TRISC com a configuração utilizada.....	36
TABELA 3 - Registrador TRISD com a configuração utilizada.....	36
TABELA 4 - Registrador TRISE com a configuração utilizada.....	36
TABELA 5 - Registrador TXSTA com a configuração utilizada.....	40
TABELA 6 - Registrador RCSTA com a configuração utilizada.....	40
TABELA 7 - Registrador CCPR1L.....	41
TABELA 8 - Registrador T2CON.....	44
TABELA 9 - Registrador CCP1CON.....	42
TABELA 10 - Incrementação do PWM.....	43
TABELA 11 - Atraso porta lógica OU.....	47
TABELA 12 - Pinos porta lógica OU.....	47
TABELA 13 - Registrador PIR1.....	48

LISTA DE ABREVIATURAS E SIGLAS

PWM - Pulse Width Modulation

LNLS - Laboratório Nacional de Luz Síncrotron

TTL - Transistor-Transistor Logic

CMOS - Complementary Metal Oxide Semiconductor

DCCT - DC Current Transformer

FPGA - Field Programmable Gate Array

DSP - Digital Signal Processor

IGBT - Insulated Gate Bipolar Transistor

MOSFET - Metal Oxide Semiconductor Field Effect Transistor

DAC - Digital to Analogic Converter

ADC - Analogic to Digital Converter

MSb - Most Significant bit

LSb - Least Significant bit

USART - Universal Synchronous Asynchronous Receiver Transmitter

PLL - Phase Locked Loop

RL - Resistor-Indutor

SUMÁRIO

Errata	2
Agradecimentos	5
RESUMO	7
Abstract.....	8
Lista de ilustrações	9
Lista de tabelas	11
Lista de abreviaturas e siglas	12
Sumário.....	13
INTRODUÇÃO.....	15
1. REVISÃO VISANDO APLICAÇÃO EM PROJETO	20
1.1 Modulação PWM.....	20
1.1.1 PWM Analógico	21
1.1.2 PWM Digital	22
1.2 Modos de Comunicação.....	23
1.2.1 Comunicação Paralela	23
1.2.2 Comunicação Serial.....	24
1.2.2.1 Transmissão Síncrona	24
1.2.2.2 Transmissão Assíncrona.....	25
1.3 Circuito de Atraso	26
1.4 Microcontroladores	28
1.4.1 Efeito Jitter	29
2. IMPLEMENTAÇÃO DE UM PWM DIGITAL DE ALTA RESOLUÇÃO	30
2.1 Interface Computacional e Comunicação de Dados	30

2.2	Microcontrolador	34
2.2.1	Configurações das Portas de Entrada e Saída do Microcontrolador	36
2.2.2	Configuração da Comunicação Serial	37
2.2.3	Configuração do Módulo PWM	41
2.1	Circuito de Atraso	44
2.2	Porta Lógica OU	46
2.3	Algoritmo do Microcontrolador.....	47
2.4	Diagrama Esquemático e Protótipo	53
3.	TESTES E RESULTADOS	55
3.1	Análises e Teste Prático	58
3.2	Resultados Obtidos por Medição	59
	CONSIDERAÇÕES FINAIS	64
	REFERÊNCIAS BIBLIOGRÁFICAS	66
	APÊNDICE	68
	Registrador TXSTA	68
	Registrador RCSTA	69
	Registrador PIR1	70
	Registrador T2CON.....	70
	ANEXOS	71
	Código Assembly.....	71

INTRODUÇÃO

O Laboratório Nacional de Luz Síncrotron (LNLS) construiu e opera um anel de armazenamento de elétrons de 1,37GeV, única fonte de Luz Síncrotron da América Latina. Este anel é constituído por uma rede magnética que faz com que elétrons descrevam uma órbita fechada, atingindo uma velocidade muito próxima à velocidade da luz. Estes elétrons emitem uma luz síncrotron ao sofrerem um desvio em sua trajetória, luz esta cujo espectro vai do infravermelho ao raio-x, e que é utilizada em estudos de diversos tipos de materiais. O anel de armazenamento de elétrons pode ser visualizado na Fig. 1.



FIGURA 1 – Foto panorâmica do anel de armazenamento de elétrons do LNLS.

Os vários tipos de magnetos (dipolos, quadrupolos, sextupolos e bobinas corretoras de órbita) que mantêm esta rede magnética são alimentados por fontes de corrente cuja regulação, atualmente, baseia-se em circuitos analógicos. A utilização de um sistema de regulação digital através de microcontroladores, DSPs e FPGAs apresentam várias vantagens sobre o sistema de regulação analógica, tais como versatilidade, facilidade de ajustar parâmetros e padronização. No entanto, circuitos de regulação digital que combinem alta resolução e velocidade, ambas necessárias às fontes do LNLS, são extremamente complexos e sofisticados.

As fontes de corrente que alimentam os eletroímãs devem ter grande precisão e resolução, sendo a primeira relacionada à eficácia do equipamento sobre o feixe de elétrons que circula no anel, enquanto que a segunda indica a menor variação de corrente que consegue ser

regulada. Quanto menor a variação de corrente, menor será a variação do campo magnético produzido pela circulação da corrente no eletroímã, o que corresponde a melhores resultados.

Nos sistemas atuais do LNLS um computador central é responsável por enviar um sinal digital para um circuito analógico através de uma comunicação serial ou ethernet, sendo assim necessária a inserção de um conversor digital/analógico (DAC) [1]. O sinal de saída do DAC varia de uma tensão $-V$ a $+V$, que corresponde ao sinal de referência. Este é somado com o valor medido (invertido), resultando em uma subtração dos sinais, sendo a diferença entre os valores conhecido como erro. Este sinal de erro recebe um ganho e passa por um controlador proporcional-integral (PI) e um condicionamento de sinal, tendo uma saída de 0 a 5 V. Este sinal entra no PWM analógico, que produz um pulso de largura correspondente à tensão de entrada, atuando assim sobre os drivers que realizam um chaveamento unipolar [2] do conversor CC/CC utilizado na fonte de corrente. Na saída do conversor é ligada a carga de saída, que apresenta um comportamento RL. Em grande parte das fontes é inserido um shunt em série com a carga, a fim de realizar medidas corrente/tensão. Em outros modelos são usados DCCTs para fazer esta medida. Um exemplo de fonte de corrente usada no LNLS é o modelo PSA, no qual os valores medidos de corrente são bidirecionais, ou seja, fornecem uma tensão equivalente $-V$ a $+V$. O valor de tensão medido passa por um amplificador isolador, pois não existe um ponto de referência para a medida realizada. O sinal de saída então é invertido e posteriormente somado com o valor de referência. O sentido da corrente depende unicamente de quais chaves do conversor full-bridge [2] foram acionadas pelos drivers. O sistema atual de regulação de uma fonte de corrente pode ser visualizado na Fig.2.

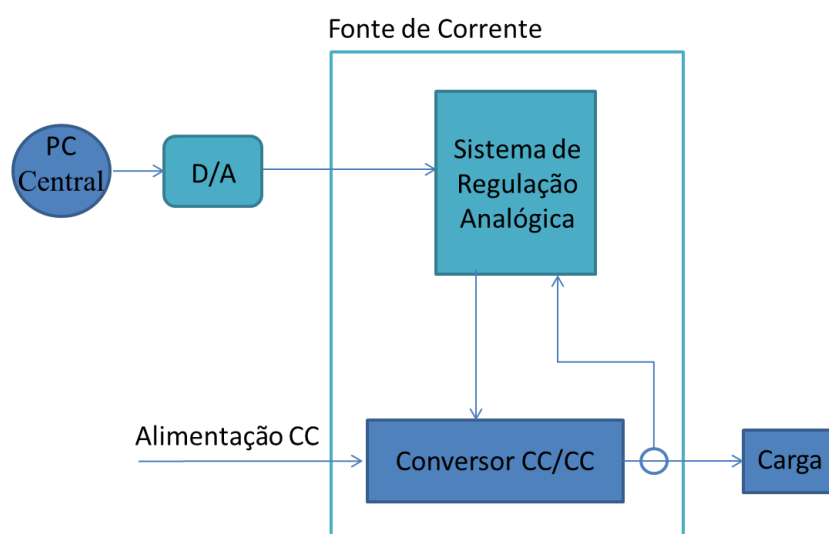


FIGURA 2 – Sistema de regulação analógica utilizado atualmente no LNLS.

O sistema de regulação que é atualmente utilizado nas fontes de corrente apresenta uma resolução de 16 bits, alcançando uma frequência de chaveamento de dezenas de kHz.

O problema encontrado nos sistemas atuais é que os sinais analógicos destes circuitos sofrem interferências externas, tais como ruídos e alteração de parâmetros a partir da variação de temperatura. Deste modo, algumas grandezas elétricas são alteradas, ocasionando perda de precisão do sistema de regulação da fonte de corrente.

A fim de substituir estes circuitos analógicos, foi proposto um sistema digital de modo a solucionar os problemas hoje existentes, conforme mencionado anteriormente. A implementação de um sistema de regulação digital apresenta um problema na relação entre resolução e frequência, pois quando a resolução é aumentada, a frequência de chaveamento diminui e vice-versa. Tendo em vista a substituição dos circuitos analógicos de regulação, necessita-se de um circuito digital que apresente resultados satisfatórios em relação ao sistema atual, não apenas em termos de precisão, mas sim em resolução e frequência.

O sinal PWM gerado deve apresentar alta resolução, além de apresentar uma elevada frequência de chaveamento, reduzindo assim o ruído audível ocasionado pelo chaveamento e também os valores de capacitores e indutores do circuito, tornando os circuitos menores, com um menor peso e custo de implementação. Desta forma, propõe-se um projeto que tenha uma entrada digital e que aumente a resolução do PWM sem alterar a frequência de chaveamento, tendo como base um circuito de atraso. O esquema de um sistema de regulação digital é visualizado na Fig. 3.

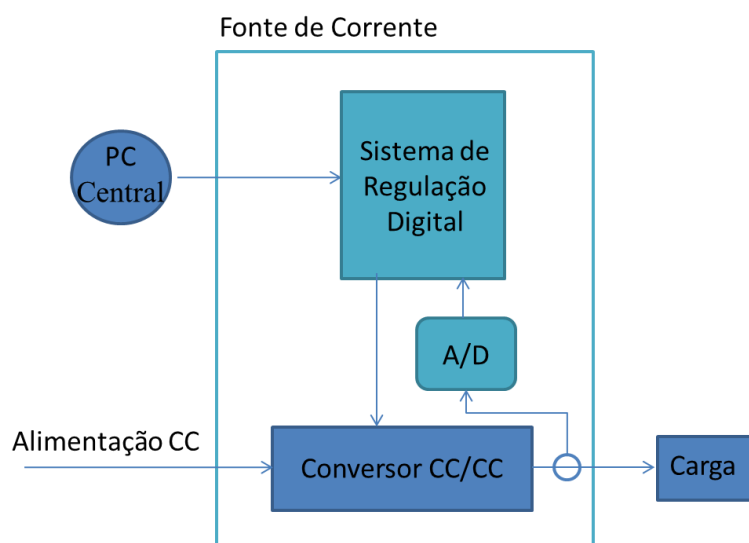


FIGURA 3 – Esquema de um sistema de regulação digital de uma fonte de corrente.

O objetivo deste projeto é utilizar um PWM digital de baixa resolução que, através de um circuito de atraso controlado digitalmente, consiga aumentar a sua resolução sem alterar a frequência de operação. A ideia é alcançar uma resolução de até 18 bits através de um microcontrolador que possui um módulo PWM com resolução máxima de 10 bits e com uma frequência de chaveamento de dezenas de kHz. Através de uma interface computacional o usuário envia pacotes de dados ao microcontrolador correspondendo a uma magnitude total de 18 bits. Estes são recebidos e processados, gerando um sinal PWM com 10 bits de resolução. Os 8 bits restantes são enviados de maneira paralela pelo microcontrolador a um circuito de atraso e estes bits impõem um atraso sobre o sinal PWM gerado. Em seguida, é realizada a lógica OU entre o sinal PWM gerado com o sinal PWM atrasado, obtendo assim o PWM final. A Fig. 4 ilustra o sistema digital implementado a fim de aumentar a resolução do PWM.

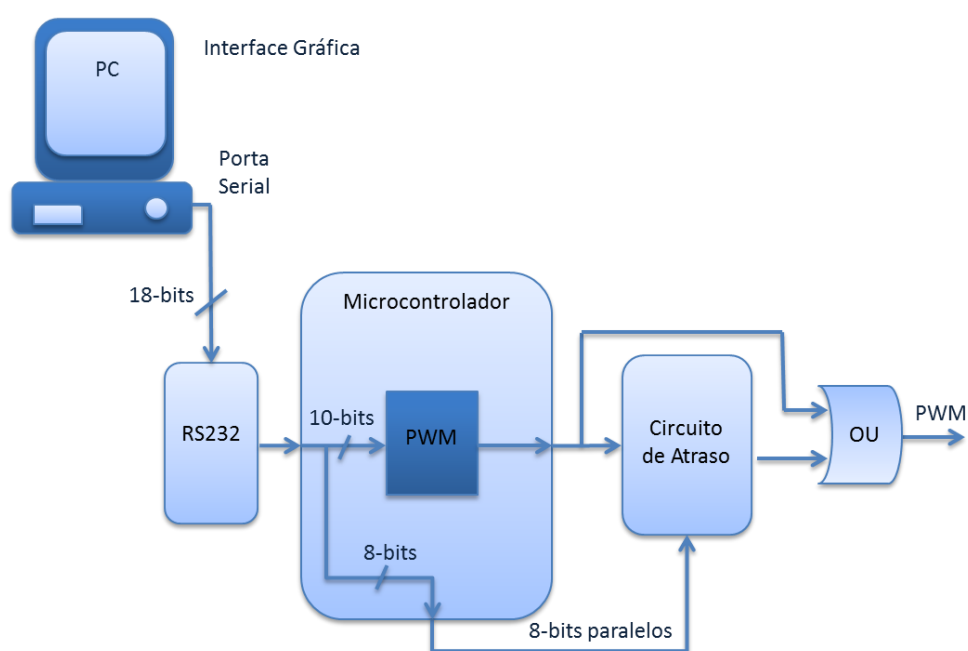


FIGURA 4 – Esquema do sistema de regulação digital projetado.

O pacote enviado pelo usuário através da interface tem uma magnitude de 18 bits, destes, 10 bits são utilizados pelo microcontrolador para gerar o sinal PWM e 8 bits para determinar o atraso que é imposto ao sinal gerado. Com o pacote de 18 bits enviados pode-se obter:

- 10 bits do microcontrolador - até 1023 diferentes valores de largura de pulso;
- 8 bits do circuito de atraso - até 255 diferentes valores de atraso (Δ atraso).

Através dos 18 bits pode-se atingir até 262.144 valores de largura de pulso. O atraso inserido ao sinal PWM pode ser visualizado na Fig. 5.

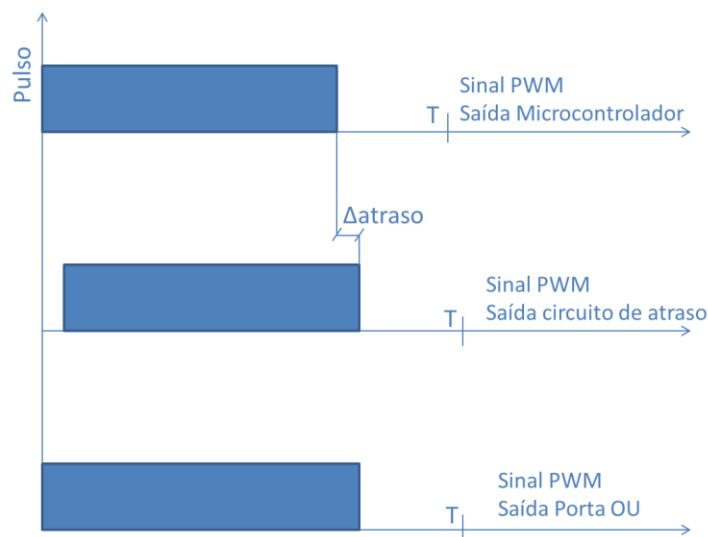


FIGURA 5 – Atraso inserido ao sinal PWM.

A ideia do projeto é inserir os 255 valores possíveis de atraso entre dois passos do PWM gerado pelo microcontrolador, obtendo assim 262.144 valores de saída. O passo do PWM está relacionado ao clock externo utilizado no microcontrolador. Já o passo de atraso incrementado depende do circuito de atraso que é adotado no projeto. A Fig. 6 ilustra a ideia do projeto.

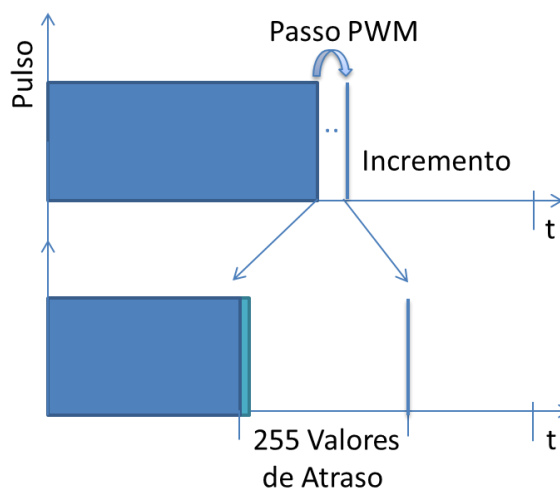


FIGURA 6 – Inserção de atrasos entre dois passos do PWM.

1. REVISÃO VISANDO APLICAÇÃO EM PROJETO

Nos dias atuais, com o avanço da tecnologia na área de semicondutores e circuitos integrados, os circuitos analógicos vêm perdendo espaço e, ao mesmo tempo, estão sendo substituídos por circuitos integrados digitais que além de reduzir o tamanho do circuito, apresentam menor consumo de potência, maior confiabilidade e versatilidade se comparado aos analógicos. Os sistemas digitais são capazes de armazenar e processar informações em alta velocidade e em forma discreta, medidas em bits [3]. Um exemplo a ser citado na área de eletrônica de potência é a substituição de circuitos analógicos que geram um sinal PWM por microcontroladores e similares que desempenham esta mesma função.

1.1 Modulação PWM

A modulação por largura de pulso (PWM – Pulse Width Modulation) tem grandes aplicações em equipamentos eletrônicos, como em fontes de alimentação, inversores de frequência, entre outros. Este tipo de modulação apresenta algumas características que podem ser destacadas, tais como frequência constante e largura de pulso variável [4]. A Fig. 7 ilustra as características desta modulação.

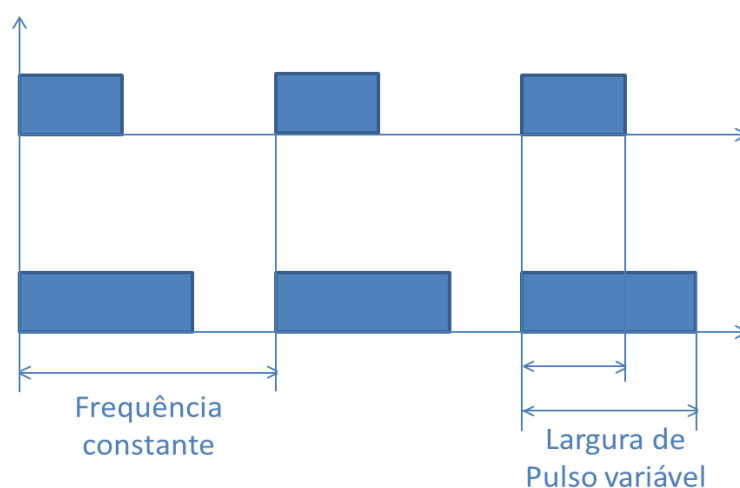


FIGURA 7 – Características da modulação PWM.

Esta modulação é utilizada em aplicações em que se deseja ter um controle de tensão ou corrente entregue a uma determinada carga. O sinal PWM é utilizado para acionar chaves, geralmente MOSFETs, IGBTs ou similares, pois os mesmos são capazes de chavear elevadas tensões ou correntes, em frequências consideráveis. A razão entre a largura de pulso e o período da onda recebe o nome de duty cycle, e sua variação faz com que os valores de tensão ou corrente média e/ou rms entregue à carga varie. A resolução obtida através desta modulação é dada a partir do número de incrementações do duty cycle dentro de um determinado período, ou seja, quanto maior o número de casos possíveis de duty cycle, maior será a resolução de saída. Conforme citado anteriormente, um sinal PWM pode ser obtido através de um circuito analógico ou por um sistema digital. Serão apresentadas as peculiaridades e modo de funcionamento para cada um dos tipos citados.

1.1.1 PWM Analógico

Um circuito analógico que gera um sinal PWM tem como base entradas analógicas. Um circuito comparador tem a finalidade de comparar o sinal de uma onda portadora com um sinal modulante, conforme Fig. 8.

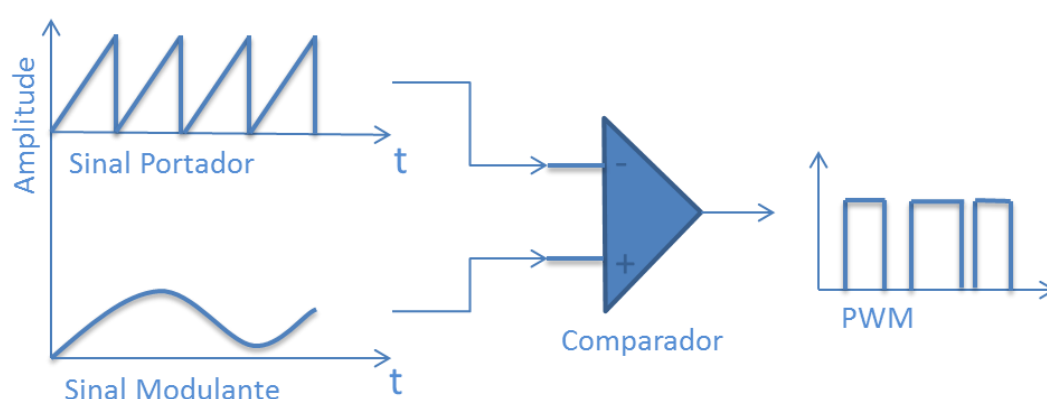


FIGURA 8 – PWM obtido a partir de um circuito analógico.

A onda portadora, geralmente dente de serra, é utilizada para aplicar um determinado período ao sinal PWM de saída, ou seja, quem determina a frequência de saída é a onda por-

tadora aplicada ao comparador [4]. O sinal modulante é o sinal analógico de entrada do circuito, responsável por determinar a largura de pulso. Neste tipo de circuito tem-se uma saída em nível alto enquanto o nível de tensão do sinal modulante for superior ao do sinal portador.

1.1.2 PWM Digital

Um sinal PWM obtido a partir de um circuito digital tem como base microcontroladores, DSPs ou FPGAs e utilizam a mesma técnica analógica, porém os dados comparados são digitais. A frequência de saída do PWM destes dispositivos está relacionada ao oscilador utilizado, bem como a base de tempo ajustada, pois quanto maior for a base de tempo, menor será a frequência de chaveamento e maior será a resolução. O valor ajustado corresponde à largura de pulso do PWM. A Fig. 9 ilustra o funcionamento.

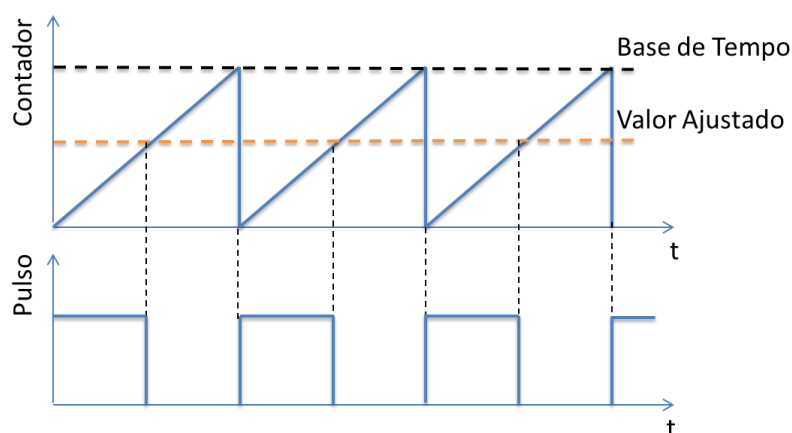


FIGURA 9 – PWM obtido a partir de um circuito digital.

Na utilização de microcontroladores, um registrador contador é incrementado a cada ciclo de clock, aumentando seu valor de maneira linear. A base de tempo é ajustada a partir de outro registrador, que é responsável por determinar a frequência do sinal PWM de saída. A largura de pulso é determinada pelo valor ajustado, pois quando o valor ajustado for igual ao valor do contador o pulso vai para nível baixo, ficando deste modo até o contador obter o mesmo valor da base de tempo. Neste ponto, o contador reinicia a contagem, gerando um pulso em nível alto até o contador se igualar ao valor ajustado novamente [5].

1.2 Modos de Comunicação

A transmissão de dados surgiu a partir da necessidade de comunicar diversos equipamentos interligados a uma rede de comunicação. As formas mais básicas de transmissão de dados entre dois dispositivos são a paralela e a serial. Existem algumas características que as distingue, que é a forma com que os dados são enviados e recebidos.

1.2.1 Comunicação Paralela

No modo de transmissão paralela, vários bits são enviados simultaneamente a cada ciclo de clock, como o próprio nome fala, de forma paralela.

A quantidade de barramentos utilizadas nesta comunicação está diretamente relacionada ao número de bits que se deseja transmitir ao mesmo tempo. A Fig 10 ilustra uma transmissão de dados em 8 bits.

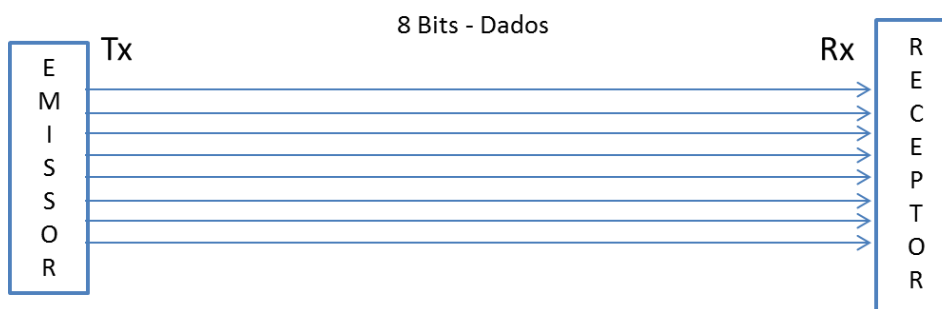


FIGURA 10 – Comunicação paralela de 8 bits entre um emissor e um receptor.

A principal vantagem da comunicação paralela é a velocidade na transmissão dos dados. Porém, existe uma grande desvantagem deste tipo de comunicação, que é a grande quantidade de barramentos de interligação e necessidade de hardware adicional para recebimento dos dados. Este tipo de comunicação é utilizada principalmente em aplicações nas quais existem pequenas distâncias entre os elementos interligados, evitando elevado custo de implementação [6].

1.2.2 Comunicação Serial

No modo de transmissão serial os bytes são enviados bit a bit por um mesmo barramento que interliga o transmissor ao receptor. O padrão RS232 utiliza este tipo de comunicação, e os sinais elétricos na porta serial que correspondem a um dado enviado possuem níveis de tensão entre 15VDC e -15VDC, sendo que de -15VDC a -3VDC corresponde ao nível lógico '1' e de +3VDC a +15VDC corresponde ao nível lógico '0'. A comunicação serial pode ser visualizada na Fig. 11.

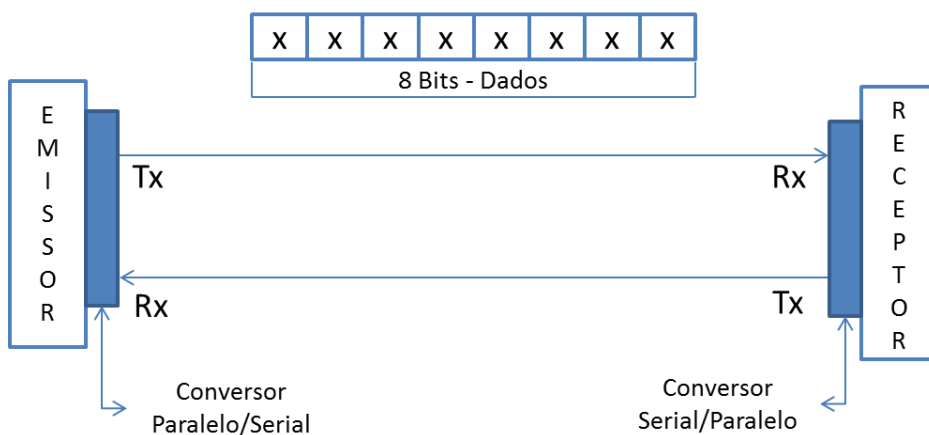


FIGURA 11 – Comunicação serial entre um emissor e um receptor.

A vantagem da transmissão serial em relação à paralela está na economia de hardware para recebimento de dados, além da redução do número de barramentos que interligam os elementos envolvidos na comunicação. A transmissão serial ocorre em duas principais formas: síncrona e assíncrona [7].

1.2.2.1 Transmissão Síncrona

Neste tipo de transmissão os bits são enviados bit a bit, porém sem nenhum bit indicativo que determina o início e o fim do byte enviado. Os bytes enviados são agrupados em fra-

mes que podem conter inúmeros bytes, sendo de responsabilidade do receptor separar os diferentes bytes recebidos através de um decodificador.

A vantagem da transmissão síncrona em relação à assíncrona é a velocidade na transmissão de dados, porém esta necessita de um sinal de clock comum para sincronizar a transmissão e recepção dos dados entre os dispositivos.

1.2.2.2 Transmissão Assíncrona

Neste tipo de transmissão os bits são enviados bit a bit, porém com bits extras indicando o início e o fim do byte enviado. Toda comunicação assíncrona depende de alguns fatores, tais como: velocidade, número de bits, verificação e bits indicativos.

A velocidade de comunicação é medida em bits por segundo (bps) e os equipamentos interligados na comunicação devem ser configurados para operar na mesma velocidade. Esta taxa de velocidade é chamada de baud rate e pode ser configurada em múltiplos de 300, valores que ficam entre a faixa de 300 bps a 115.000 bps.

O número de bits define quantos bits compõe o byte a ser enviado, sendo que na maioria das comunicações são utilizados todos os possíveis, ou seja, 8 bits.

O checksum é um método utilizado para verificar se houve algum erro durante a transmissão e recepção dos dados. Este método realiza a soma dos pacotes recebidos, que posteriormente é subtraído pelo pacote de checksum transmitido (contendo a soma dos pacotes enviados). Caso o valor resultante seja zero, os pacotes enviados foram recebidos corretamente, caso contrário houve um erro de transmissão, sendo necessário o reenvio dos dados.

Além do valor transmitido contendo geralmente 8 bits, o transmissor envia 2 bits indicativos, um no início do dado enviado e outro no final.

O primeiro é chamado de start bit e serve para informar ao receptor que será dado início à transmissão de um pacote de dados. Já o último bit é chamado de stop bit e é utilizado para informar ao receptor o final do dado enviado serialmente. Um pacote completo de dados enviado no padrão RS232 pode ser visualizado na Fig. 12.

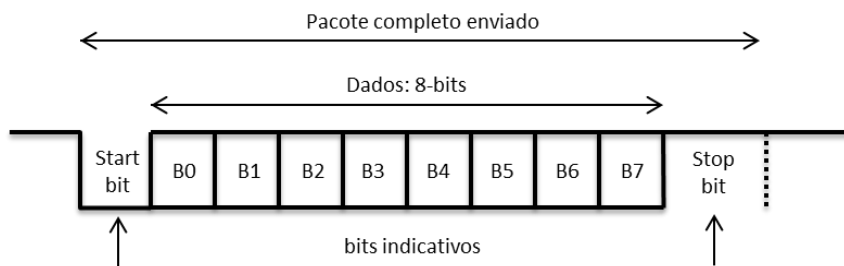


FIGURA 12 – Pacote de dados com magnitude de 10bits enviados no padrão RS232.

1.3 Circuito de Atraso

Um circuito de atraso é um circuito que produz determinado atraso sobre um sinal de entrada. Um circuito integrado que produz este atraso é o DS1020, que é construído com a tecnologia CMOS. Este circuito de silício é compatível com tecnologia CMOS e TTL e é utilizado em aplicações em que se deseja obter um sinal de saída com determinado atraso em relação ao de entrada, sendo o usuário capaz de escolher o respectivo valor de atraso [8]. A pinagem do circuito integrado pode ser visualizada na Fig. 13.

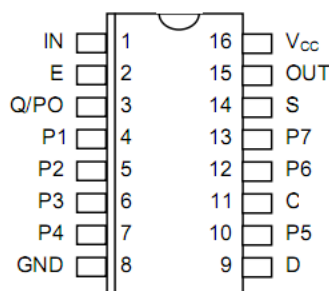


FIGURA 13 – Pinagem do circuito de atraso DS1020 [8].

O sinal de entrada é ligado no pino IN, sendo o atraso determinado a partir de 8 bits enviados de forma paralela ou serial. Desta forma, pode-se alcançar até 255 valores de atraso, cujo valor é determinado pelo usuário, obtendo o sinal atrasado através da porta OUT.

A família DS1020 possui diversos tipos de circuitos de atraso, com diferentes valores de atrasos em suas incrementações. Estes diferentes valores podem ser visualizados na Tabela 1.

TABELA 1 – Família DS1020 [8]

	Mín. Atraso(η s)					Máx. Atraso(η s)			Porta Paralela	Signifi- cativos
Valor Binário atraso	0	0	0	0	0	..	1	1	P7	MSB
	0	0	0	0	0	..	1	1	P6	
	0	0	0	0	0	..	1	1	P5	
	0	0	0	0	0	..	1	1	P4	
	0	0	0	0	0	..	1	1	P3	
	0	0	0	0	1	..	1	1	P2	
	0	0	1	1	0	..	1	1	P1	
	0	1	0	1	0	..	0	1	P0	LSB
DS1020-15	10	10,15	10,30	10,45	10,6	..	48,1	48,25		
DS1020-25	10	10,25	10,5	10,75	11	..	73,5	73,75		
DS1020-50	10	10,5	11	11,5	12	..	137	137,5		
DS1020-100	10	11	12	13	14	..	264	265		
DS1020-200	10	12	14	16	18	..	518	520		

Estes circuitos integrados possuem um atraso inicial que é determinado a partir do atraso das portas lógicas internas ao DS1020, ou seja, independentemente do valor setado de atraso, o sinal de saída terá um valor de 10η s além do valor de atraso atribuído. O valor do atraso total pode ser obtido a partir da Eq. 1 e Eq. 2.

$$A_{total} = A_{inicial} + A_{atraso} \quad (1)$$

Onde,

$$A_{atraso} = A_{passo} * x \quad (2)$$

Sendo, o atraso total do circuito de atraso (A_{total}), o atraso inicial devido às portas lógicas ($A_{inicial}$), o atraso obtido a partir de 8 bits (A_{atraso}), o valor de 8 bits recebidos paralelamente no circuito de atraso (x) e o atraso do passo (A_{passo}).

Para configurar o atraso de forma paralela, o valor que determina o atraso aplicado sobre o sinal de entrada é dado através das portas P7 a P0. Na atribuição dos valores de atraso deve-se tomar cuidado com o MSb, no caso atribuído a porta P7. Para que haja um funcionamento neste modo, deve-se colocar o valor '1' lógico no pino S.

No modo serial, o valor que determina o atraso aplicado sobre o sinal de entrada é dado de forma serial na porta D, cujo valor de 8 bits atribuirá o valor de atraso. Caso o sistema opere neste modo, é necessário enviar um sinal de clock para o dispositivo, a fim de sincronizar os diferentes módulos que são interligados ao circuito de atraso. Além disso, o circuito exige um sinal na porta E para leitura de dados.

Para que haja um funcionamento neste modo, deve-se colocar o valor '0' lógico no pino S. Os pinos que não são utilizados em ambos os modos devem ser ligados ao GND ou VDD, pois os mesmos não podem ficar em alta impedância. Um diagrama esquemático interno que representa melhor o DS1020 pode ser visualizado na Fig. 14.

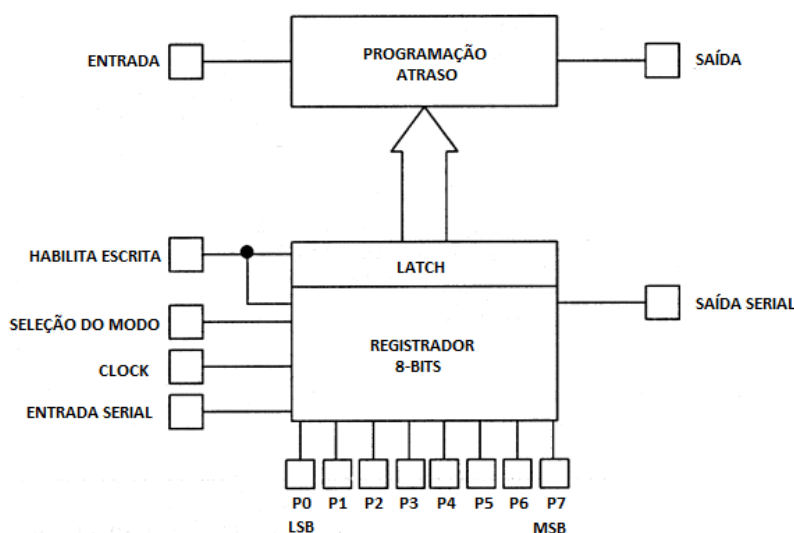


FIGURA 14 – Diagrama esquemático interno do DS1020 [8].

1.4 Microcontroladores

São dispositivos que executam determinadas tarefas que são definidas por um programa armazenado em sua memória interna. Os microcontroladores são compostos de processador, memória e periféricos de entrada e saída, além de poder integrar funções adicionais em sua estrutura interna, como, por exemplo, memórias, ADC, DAC, USART, PLL e PWM. Porém, estas funções não são encontradas em todos os modelos de microcontrolador. Os microcontroladores encontrados atualmente no mercado trabalham em uma frequência de clock de até

80MHz. Seu consumo é consideravelmente pequeno, o que possibilita a aplicação em projetos de dispositivos embarcados de pequeno e médio porte, pois é o fator fundamental quando se fala em alimentação limitada, podendo ser empregado na indústria no controle e automação de processos.

1.4.1 Efeito Jitter

A maioria dos microcontroladores possui um circuito PLL. Estes são utilizados para atingir maiores frequências de clock interno a partir de um clock externo inferior. O uso deste circuito permite melhorar o desempenho, reduzindo o ruído. Um ponto interessante que deve ser analisado na utilização destes circuitos PLL dos microcontroladores é que eles criam um nível pequeno, mas mensurável, de deslocamentos em seu período, chamado de jitter [9].

O valor do jitter é dado geralmente pelo fabricante do microcontrolador e através destas informações pode-se calcular o deslocamento máximo no clock dado pela Eq. 3.

$$\Delta clock = Jitter * \frac{1}{F_{osc}} \quad (3)$$

Para este projeto, o deslocamento total no clock deve ser menor que o passo do circuito de atraso visto anteriormente, assim garantindo um PWM com incremento de seu duty cycle para todos os valores possíveis dentro de um determinado período quando conectado a um circuito de atraso. O efeito jitter pode ser visualizado na Fig.15.

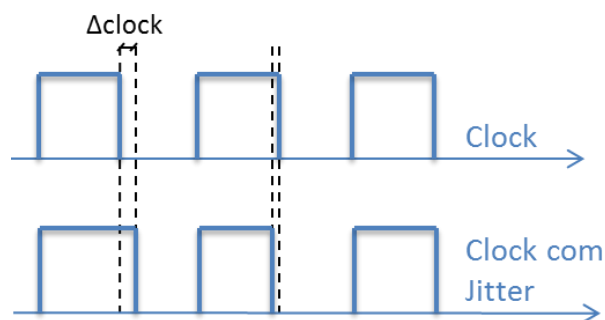


FIGURA 15 – Definição do efeito jitter em um sinal de clock.

2. IMPLEMENTAÇÃO DE UM PWM DIGITAL DE ALTA RESOLUÇÃO

Conforme visto na Fig. 4, a ideia do projeto é aumentar a resolução de um PWM sem alterar a frequência de chaveamento, aplicando um circuito de atraso. A fim de se realizar testes do projeto, foi desenvolvida uma interface de comunicação entre o usuário e o hardware. Os dados são enviados do computador para um microcontrolador através de uma determinada comunicação. Os dados são recebidos e processados, gerando um PWM, sendo posteriormente imposto um atraso no sinal gerado através de um circuito de atraso. Os sinais PWM gerados e atrasados passam por uma lógica OU, obtendo na saída o PWM final com alta resolução e frequência.

2.1 Interface Computacional e Comunicação de Dados

Tendo em vista a necessidade de transmitir dados do computador para o microcontrolador, foi desenvolvido um programa em Matlab para realizar esta comunicação.

O Matlab [10] foi criado em 1970 e é reconhecido mundialmente até hoje como uma das melhores ferramentas para o processamento matemático, pois é voltado às áreas de engenharia, física, estatística, economia, entre outras. Pode-se dizer que este software tem amplas aplicações, como por exemplo, em processamento de sinais e imagem, aquisição de dados, além de permitir a construção de interfaces gráficas, o que o torna muito versátil.

A partir deste software foi desenvolvida uma interface gráfica com o objetivo de realizar testes com o sistema digital implementado, facilitando a comunicação entre usuário e hardware. Foi utilizada a comunicação serial neste projeto, visando reduzir o tamanho do hardware que seria necessário para o recebimento de 18 bits de maneira paralela.

A interface possibilita entrar com 2 pacotes de dados, um deles com magnitude de 8 bits, contendo os bits mais significativos do PWM, e o outro com 10 bits, sendo os 2 bits menos significativos do PWM, mais 8 bits do circuito de atraso. A comunicação serial opera com transmissão de pacotes cuja magnitude é igual a 8 bits. Desta forma, o valor de 18 bits fornecidos pelo usuário deve ser quebrado em 3 pacotes para que posteriormente seja transmitido.

O primeiro pacote de 8 bits é responsável pelo ajuste dos bits mais significativos do PWM, o segundo pacote pelo ajuste dos bits menos significativos da PWM e o terceiro pacote responsável pelo atraso do sinal PWM, obtendo a seguinte representação:

B17	B16	B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
8- bits MSB PWM								2-bits LSB PWM		8-bits circuito de atraso							

A interface computacional desenvolvida pode ser visualizada na Fig. 16.

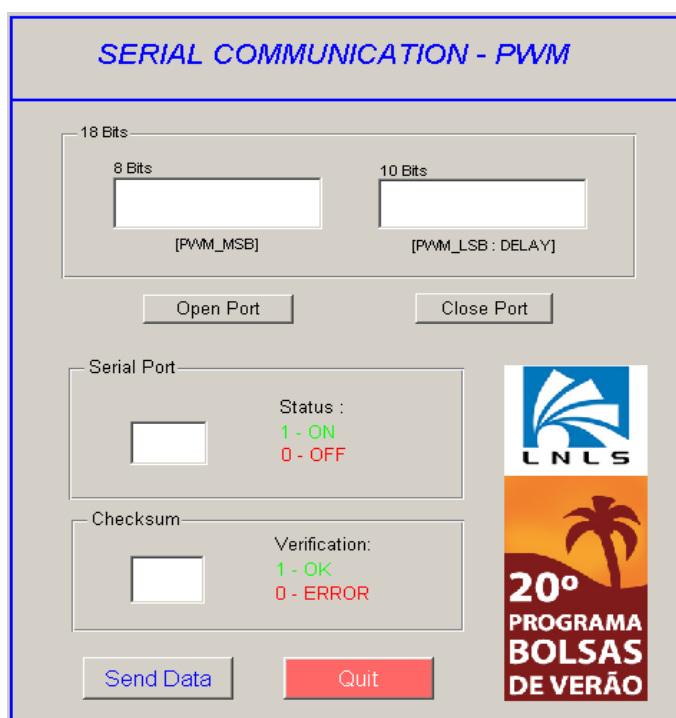


FIGURA 16 – Interface do sistema de comunicação desenvolvida em Matlab.

A fim de verificar se os dados foram recebidos pelo microcontrolador de maneira correta, é realizada a soma dos pacotes transmitidos pela interface. A soma dos 3 pacotes resultaria em um quarto pacote com magnitude superior a 8 bits, porém os bits que ultrapassam o valor de 8 bits são ignorados de modo que a interface envie somente 1 byte para verificação.

Após a transmissão, o microcontrolador calcula o checksum e retorna um pacote referente ao recebimento de dados. Caso o valor retornado pelo microcontrolador seja '0' então ocorreu um erro no recebimento de algum pacote. Desta forma, o checksum apresentado para o usuário estará errado, sendo que o mesmo deverá reenviar os dados. Caso o valor seja '255'

o valor do checksum estará correto, e o usuário não precisará enviar um mesmo valor novamente. O fluxograma da Fig. 17 ilustra o funcionamento da interface.

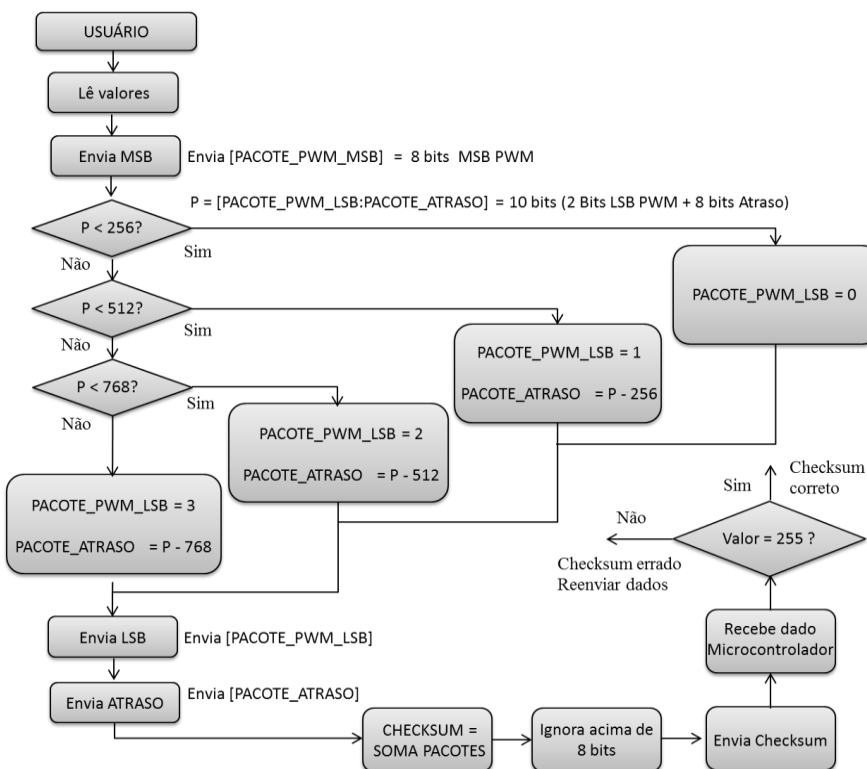


FIGURA 17 – Funcionamento da interface de comunicação entre o computador e o microcontrolador do sistema de controle digital.

Como este projeto utiliza a comunicação serial conforme padrão RS232 é necessário inserir no circuito de comunicação entre o computador e microcontrolador um conversor MAX232 [11]. Este circuito converte o sinal padrão RS232 para um sinal em nível TTL que varia de 0 a 5 volts. Este circuito integrado é fabricado pela Texas Instruments e sua pinagem pode ser visualizada na Fig. 18.

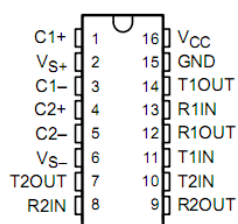


FIGURA 18 – Conversor MAX232 [11].

A forma de ligação deste circuito pode ser visualizada na Fig. 19.

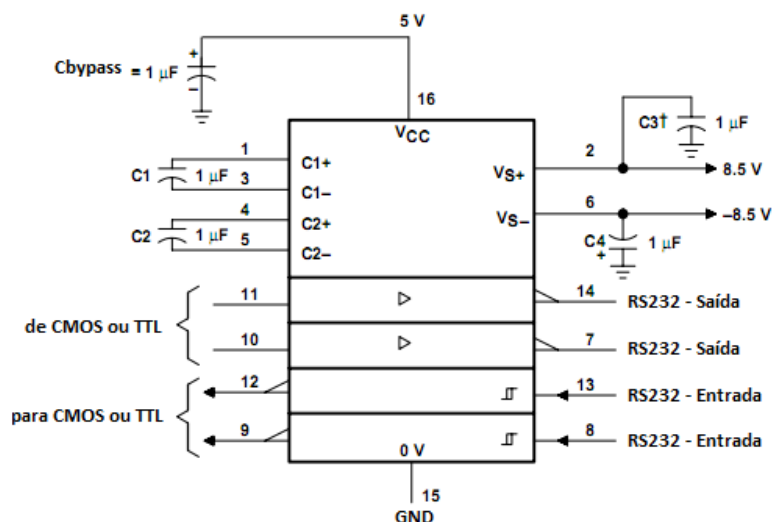


FIGURA 19 – Esquema de ligação do conversor MAX232 [11].

O conversor MAX232 é ligado ao conector DB9 do computador. O conector possui nove pinos, destes, 3 foram utilizados. O pino 2 é responsável por receber os dados enviados do hardware para o computador, enquanto que o pino 3 envia informações do computador para o hardware. O pino 5 é utilizado para deixar ambos elementos envolvidos na comunicação em uma mesma referência.

A Fig. 20 ilustra a forma de ligação do conversor MAX232 com o microcontrolador e a porta serial do computador.

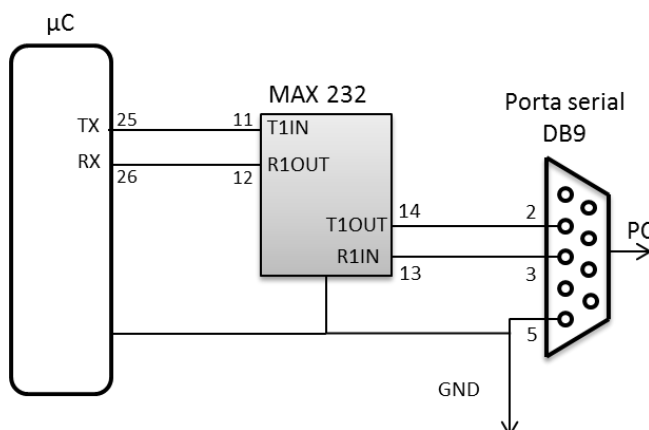


FIGURA 20 – Aplicação do conversor MAX232 entre hardware e computador.

2.2 Microcontrolador

Pode-se dizer que o microcontrolador é a parte principal do projeto, pois através do mesmo é realizado o recebimento e processamento dos dados enviados a partir da interface computacional desenvolvida em Matlab.

Para este projeto foi utilizado um microcontrolador PIC16F877[12], fabricado pela Microchip. O microcontrolador trabalha em uma frequência máxima de 20MHz, apresentando um tempo por ciclo de instrução de 200ns.

Este microcontrolador possui um módulo USART, fator muito importante para esta aplicação, pois a comunicação entre software e hardware é realizada serialmente. Além disso, este microcontrolador possui dois canais PWM, ambos com uma resolução máxima de 10 bits. O mesmo possui diversas portas paralelas de 8 bits. A porta D será utilizada para enviar o valor de 8 bits ao circuito de atraso.

A pinagem do microcontrolador PIC16F877 utilizado neste projeto pode ser visualizada na Fig. 21.

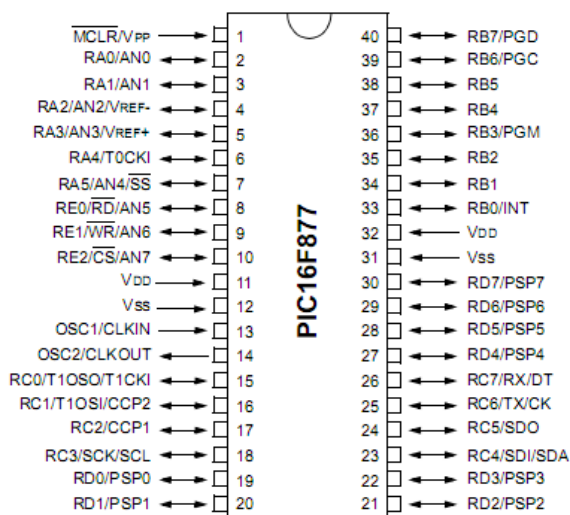


FIGURA 21 – Pinagem do microcontrolador PIC 16F877.

A arquitetura interna ao microcontrolador PIC16F877 pode ser visualizada na Fig.22.

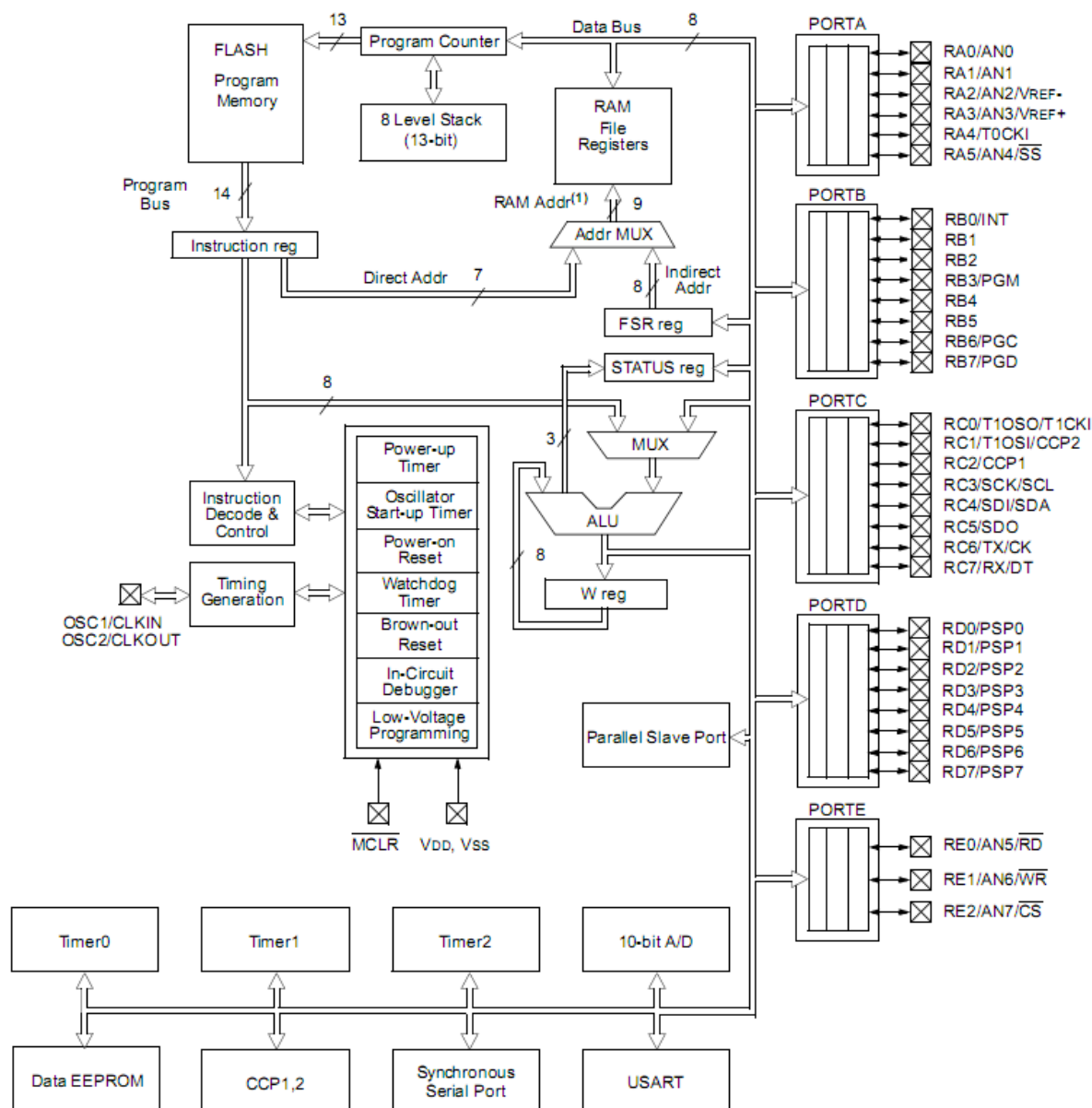


FIGURA 22 – Arquitetura interna ao microcontrolador PIC 16F877.

Para esta aplicação foram utilizadas as seguintes portas deste microcontrolador:

RC2 - Saída PWM (sinal PWM gerado pelo microcontrolador);

RC6 - Saída de dados serial (envio de dados para a interface);

RC7 – Entrada de dados serial (recebimento de dados da interface);

RD7 a RD0 – Saída de dados paralelamente (circuito de atraso);

RE0 – Flag de saída (sistema ligado).

Para que o circuito funcione de maneira adequada devem-se realizar as configurações das portas e dos registradores do microcontrolador. A próxima seção apresenta de maneira sucinta estas configurações.

2.2.1 Configurações das Portas de Entrada e Saída do Microcontrolador

Em projetos que envolvem a aplicação de microcontroladores devem-se realizar as configurações das portas de maneira a indicar se a porta irá operar como entrada ou saída. Cada porta possui um registrador que configura a direção dos seus pinos.

Caso seja atribuído o valor '1' à posição do registrador, equivalente àquele pino, então o mesmo será entrada, caso contrário (valor '0') será saída. Assim, as configurações dos registradores das respectivas portas podem ser visualizadas na tabela 2, 3 e 4.

TABELA 2 – Registrador TRISC com a configuração utilizada

	RC7	RC6	RC5	RC4	RC3	RC2	RC1	RC0
Posição	7	6	5	4	3	2	1	0
I/O	1	0	x	x	x	0	x	x

TABELA 3 – Registrador TRISD com a configuração utilizada

	RD7	RD6	RD5	RD4	RD3	RD2	RD1	RD0
Posição	7	6	5	4	3	2	1	0
I/O	0	0	0	0	0	0	0	0

TABELA 4 – Registrador TRISE com a configuração utilizada

	RE7	RE6	RE5	RE4	RE3	RE2	RE1	RE0
Posição	7	6	5	4	3	2	1	0
I/O	x	x	x	x	x	x	x	0

As instruções abaixo são utilizadas para a configuração.

```
=====
MOVLW    B'10111011'
MOVLW    TRISC
MOVLW    B'00000000'
```

```

MOVWF    TRISD
MOVLW    B'00000000'
MOVWF    TRISE

```

=====

2.2.2 Configuração da Comunicação Serial

Para a comunicação serial entre a interface e microcontrolador deve-se configurar os registradores envolvidos na transmissão e recepção dos dados, pois o microcontrolador apresenta dois modos de transmissão, que são:

Síncrono: depende de um sinal de clock para sincronizar a comunicação entre os elementos envolvidos;

Assíncrono: não depende de sinal de clock, porém deve-se atribuir um baud rate a comunicação.

Neste projeto utilizou-se o modo de comunicação assíncrona, com um baud rate de 9.600bps. O módulo de transmissão de dados do microcontrolador pode ser visualizado na Fig. 23.

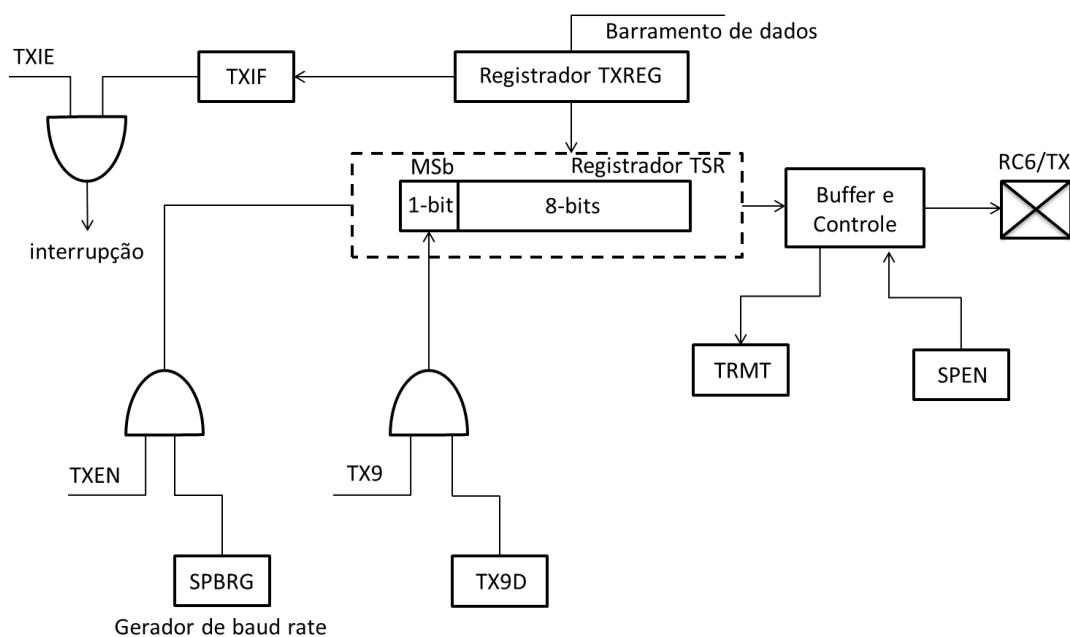


FIGURA 23 – Módulo de transmissão de dados serialmente do microcontrolador.

Analisando o módulo de envio de dados pode-se dizer que os dados contendo 8 bits a serem transmitidos devem ser armazenados em TXREG para que posteriormente sejam transmitidos. Além disso, o módulo permite a transmissão de um bit adicional, assim o pacote transmitido pode apresentar uma magnitude de 9 bits. O TRMT é utilizado para indicar a situação do registrador da transmissão. Deve-se indicar ao módulo se o mesmo vai operar na forma síncrona ou assíncrona através do SYNC do registrador. Caso seja atribuído o valor '1' a esta posição o mesmo atuará no modo síncrono, caso contrário o mesmo funcionará no modo assíncrono. Para habilitar o módulo de transmissão deve-se colocar o TXEN em '1', além disso, deve-se configurar o gerador de baud rate através do SPBRG.

Para este módulo existem duas opções de baud rate, o mesmo pode operar no modo 'baixo' ou 'alto'. Deve-se calcular o baud rate para ambos os modos de forma a utilizar o que apresente um menor erro relacionado à sua precisão.

Cálculo de SPBRG (seleção do baud rate):

Para baud rate baixo (BRGH = 0) o cálculo é realizado pela Eq. 4.

$$SPBRG = \frac{F_{osc}}{64 * BR_{projeto}} - 1 \quad (4)$$

$BR_{projeto} = 9.600\text{bps}$;

$F_{osc} = 20\text{MHz}$.

Desta forma se tem SPBRG igual a 31,552 ~ 31.

Posteriormente deve ser calculado o valor de baud rate real através da Eq. 4, levando-se em consideração a um SPBRG de 31.

Assim o baud rate real (BR_{real}) é igual a 9.765,625 bps.

O erro em percentagem é calculado através da Eq.5.

$$Erro = \frac{(BR_{real} - BR_{projeto})}{BR_{projeto}} * 100\% \quad (5)$$

O erro obtido é igual a **1,725%**.

Para baud rate alto (BRGH = 1) o cálculo é realizado pela Eq. 6.

$$SPBRG = \frac{F_{osc}}{16 * BR_{projeto}} - 1 \quad (6)$$

$BR_{projeto} = 9.600\text{bps}$;

$F_{osc} = 20\text{MHz}$.

Desta forma se tem SPBRG igual a 129,2 ~ 129.

Posteriormente deve ser calculado o valor de baud rate real através da Eq. 6, levando-se em consideração a um SPBRG de 129.

Assim o baud rate (BR_{real}) é igual a 9.615,38 bps.

Pela Eq. 5 chega-se a um erro igual a **0,16%**.

O cálculo que apresentou menor erro foi o baud rate alto, então se deve colocar o valor '1' na respectiva posição do registrador.

O módulo de recepção de dados do microcontrolador pode ser visualizado na Fig. 24.

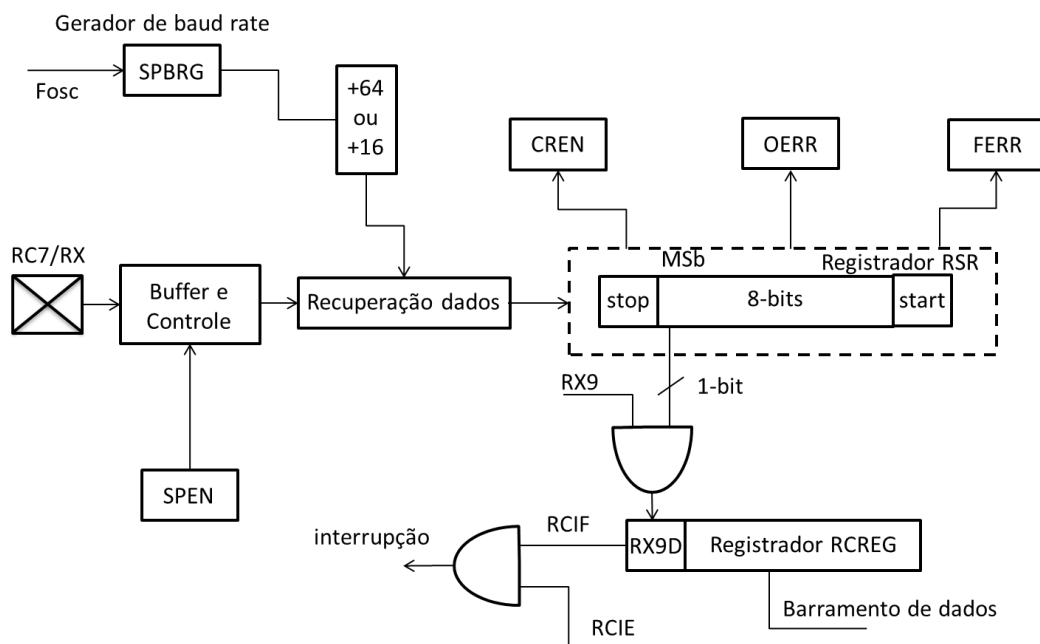


FIGURA 24 – Módulo de recepção de dados serialmente do microcontrolador.

Analisando o módulo de envio de dados pode-se dizer que os dados recebidos são armazenados em RREG para que posteriormente o mesmo seja movido para outra variável declarada no projeto.

O módulo permite a recepção de um bit adicional, assim o pacote recebido pode apresentar uma magnitude de 9 bits. O SREN é utilizado para indicar se a recepção será unitária, caso o valor atribuído seja '1' o microcontrolador receberá apenas 1 pacote, ou seja, a recepção será unitária, caso contrário, o módulo sempre estará recebendo e atualizando os dados recebidos.

Deve-se também configurar a posição CREN, indicando que a recepção dos dados será contínua. Caso seja atribuído o valor '1' a esta posição, a mesma estará configurada para sempre receber novos dados pela serial. A posição FERR, OERR indicam erro no stop bit e na leitura dos dados.

As configurações dos registradores envolvidos na transmissão e recepção são visualizadas na Tabela 5 e Tabela 6.

TABELA 5 – Registrador TXSTA com a configuração utilizada

	CSRC	TX9	TXEN	SYNC	-	BRGH	TRMT	TX9D
Posição	7	6	5	4	3	2	1	0
Valor	x	0	1	0	x	1	x	x

TABELA 6 – Registrador RCSTA com a configuração utilizada

	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D
Posição	7	6	5	4	3	2	1	0
Valor	1	0	0	1	x	x	x	x

As instruções abaixo são utilizadas para a configuração do módulo de transmissão e recepção dos dados.

```

=====
MOVLW .129
MOVWF SPBRG
MOVLW B'00100100'
MOVWF TXSTA
MOVLW B'10010000'
MOVWF RCSTA
=====

```


2.2.3 Configuração do Módulo PWM

O microcontrolador 16F877 possui dois módulos PWM que devem ser configurados de forma a atender as necessidades de projeto. O módulo PWM do microcontrolador pode ser visualizado na Fig. 25.

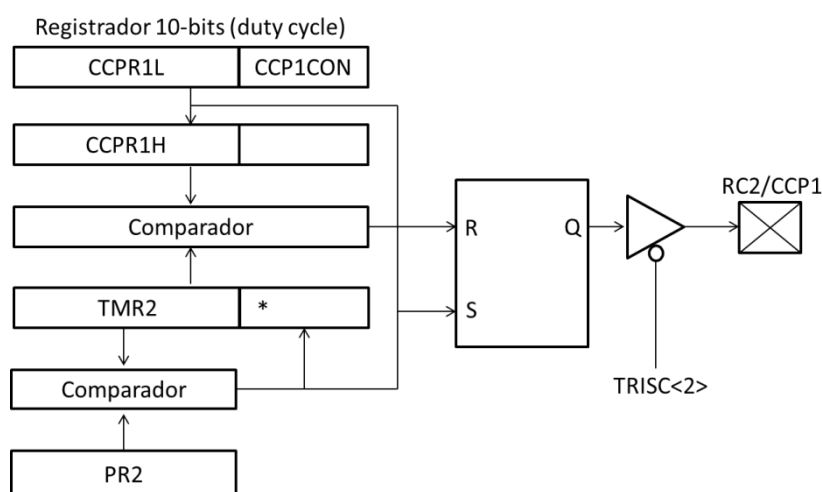


FIGURA 25 – Módulo PWM do microcontrolador PIC16F877.

Analisando o módulo PWM pode-se dizer que o mesmo possui um PWM com resolução máxima de 10 bits, cujo valor é dividido em dois registradores. Os 8 bits mais significativos são armazenados no registrador CCPR1L <7:0>, enquanto que os 2 menos significativos são armazenados em CCP1CON <5:4>. Este valor de 10 bits é o valor responsável por ajustar o duty cycle.

Assim, obtem-se a seguinte representação para os bits do PWM:

$$\langle \text{CCPR1L}[7:0] : \text{CCP1CON}[5:4] \rangle$$

Estes registradores podem ser visualizados na Tabela 7 e 8.

TABELA 7 – Registrador CCPR1L

CCPxB9	CCPxB8	CCPxB7	CCPxB6	CCPxB5	CCPxB4	CCPxB3	CCPxB2
7	6	5	4	3	2	1	0

O registrador PR2 determina o período do sinal PWM de saída. Já o TMR2 é utilizado como contador. O RC2/CCP1 é o pino de saída que entrega a largura de pulso ao sistema. Através do TRISC < 2> é ativada a saída PWM do microcontrolador.

O funcionamento do módulo PWM segue as explicações vistas na seção 1.1.2. Segundo explicações, um módulo de PWM digital deve possuir um contador que é incrementado a cada ciclo de clock, uma base de tempo e um valor ajustado, funções estas que são realizadas pelo TMR2, PR2, CCP1L e CCP1CON. O funcionamento do módulo PWM pode ser visualizado na Fig. 26.

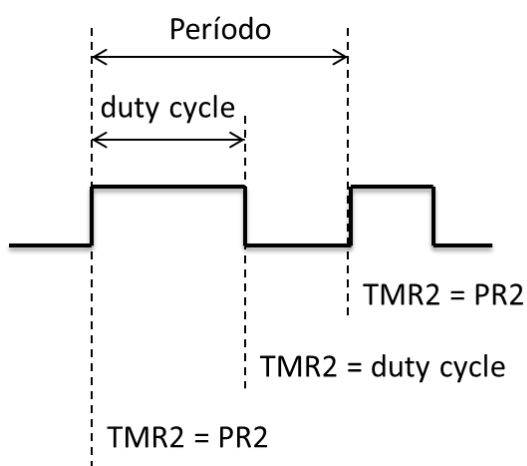


FIGURA 26 – Funcionamento do módulo PWM do microcontrolador PIC16F877.

Analisando a Fig. 26 pode-se dizer que se tem um pulso de saída em nível alto enquanto o valor do duty cycle (valor ajustado no CCP1L e CCP1CON) for menor que o valor do contador (TMR2). Quando os valores se igualarem, o pulso vai para nível baixo até que o contador TMR2 seja reinicializado. A reinicialização ocorre quando TMR2 for igual à base de tempo ajustada (PR2). Quando o período for reinicializado, o pulso volta para o nível alto até que o valor ajustado do duty cycle se iguale ao contador novamente.

Para que o módulo PWM funcione é necessário configurar o registrador CCP1CON. Este registrador é visualizado na Tabela 8.

TABELA 8 – Registrador CCP1CON

-	-	CCPxX	CCPxY	CCP1M3	CCP1M2	CCP1M1	CCP1M0
7	6	5	4	3	2	1	0

As posições CCP1CON <3:0> são utilizadas para selecionar o modo CCP1, podendo funcionar como Compare/Capture/PWM. Para funcionar no modo PWM, deve-se atribuir '1111' as posições CCP1CON <3:0> do registrador.

A resolução máxima (10 bits) do PWM deste microcontrolador é obtida movendo-se o valor 255 ao registrador PR2, valor máximo possível a ser ajustado em 8 bits.

Quando este microcontrolador funciona com a resolução máxima, ou seja, sem perda de resolução, a frequência máxima obtida é 19,531kHz, valor obtido através da Eq. 7.

$$T_{pwm} = [PR2 + 1] * 4 * \frac{1}{F_{osc}} * Prescale \quad (7)$$

Onde a frequência do oscilador (F_{osc}) é 20MHz, a base de tempo ajustada (PR2) é 255 e o Prescale é 1:1.

O período do PWM é 51.2µs, onde a frequência é obtida aplicando o inverso do período, desta forma chega-se a 19,531kHz. O passo no período do PWM é dado pela Eq.8.

$$T = \frac{1}{F_{osc}} \quad (8)$$

Neste projeto adotou-se a frequência máxima do microcontrolador (20MHz), e através da Eq. 8 pode-se dizer que cada valor incrementado nos bits do PWM se tem um passo de 50ns. Caso fosse trabalhado somente com os bits mais significativos do PWM do microcontrolador, se teria um passo de 200ns, porém este valor é dividido por 4 quando se trabalha com os 2 menos significativos. Na Tabela 9 pode ser visualizado o tempo incrementado a partir da variação dos bits do PWM.

TABELA 9 – Incrementação do PWM

[CCPR1L] : [CCP1CON]	Passo
[00000000] : [01]	50ns
[00000000] : [10]	100ns
[00000000] : [11]	150ns
[00000001] : [00]	200ns

Para que o contador do PWM funcione é necessário ativar o TMR2 através da posição TMR2ON, bem como se deve ajustar o seu prescale, ambas as configurações são realizadas no registrador T2CON, visto na Tabela 10.

TABELA 10 – Registrador T2CON

-	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0
7	6	5	4	3	2	1	0

As instruções abaixo são utilizadas para a configuração do módulo PWM do microcontrolador.

```

=====
MOVLW B'00001111'
MOVWF CCP1CON
MOVLW .255
MOVWF PR2

MOVLW B'00000100'

MOVWF T2CON
=====

```

2.1 Circuito de Atraso

Conforme mencionado anteriormente, cada passo do PWM possui 50ns, então se realizou a análise dos diferentes tipos de circuito de atraso da família DS1020, adotando assim para este projeto o circuito DS1020-25, pois o mesmo possui um valor de atraso máximo muito próximo ao valor do passo do PWM gerado pelo microcontrolador.

Analisando a Tabela 1, pode-se dizer que o circuito DS1020-25 possui um atraso máximo de 73,75ns, obtido a partir da Eq.1.

O circuito de atraso DS1020-25 possui um passo de atraso de 0,25ns, ou seja, a cada valor incrementado no circuito de atraso é somado 0,25ns. O valor de atraso pode ser variado em 255 passos, desta forma pode-se obter um atraso total de 63,75ns.

O incremento dos bits menos significativos do PWM gera um avanço de 50ns, desta forma deve-se realizar um condicionamento do valor de atraso através de uma aproximação linear de modo a obter um valor máximo de atraso de 50ns, valor este obtido a partir de um valor de entrada igual a '255'.

Assim, o valor inserido ao circuito de atraso que faz com que o sinal obtenha um atraso de 50ns é '200', podendo ser aproximado pela Eq. 9.

$$\text{Atraso} = a * x + b \quad (9)$$

Onde 'x' é valor de entrada (0 a 255 em decimal) e 'a' e 'b' as variáveis da equação da reta obtidas pelo ponto máximo e mínimo.

O ponto mínimo da reta é obtido quando o valor zero for enviado ao circuito de atraso, valor obtido pela Eq. 9.

$$0 = a * 0 + b$$

$$b = 0$$

Como visto anteriormente, um atraso de 50ns é obtido quando um valor de '200' é atribuído ao circuito de atraso, desta forma, quando a interface computacional enviar o valor '255' este deve ser aproximado para '200'. Substituindo o valor de entrada e saída aproximada na Eq. 9 obtem-se,

$$200 = a * 255 + b$$

$$b = 0$$

$$a = \frac{200}{255}$$

Substituindo os valores das variáveis encontradas a partir do ponto máximo e mínimo se obtêm a equação que representa a aproximação linear realizada do circuito de atraso. A Eq.10 representa esta aproximação.

$$y = \frac{200}{255} * x \quad (10)$$

Sendo ‘x’ o pacote de atraso enviado pela interface. O condicionamento do atraso em função do valor de entrada pode ser visualizado na Fig. 27.

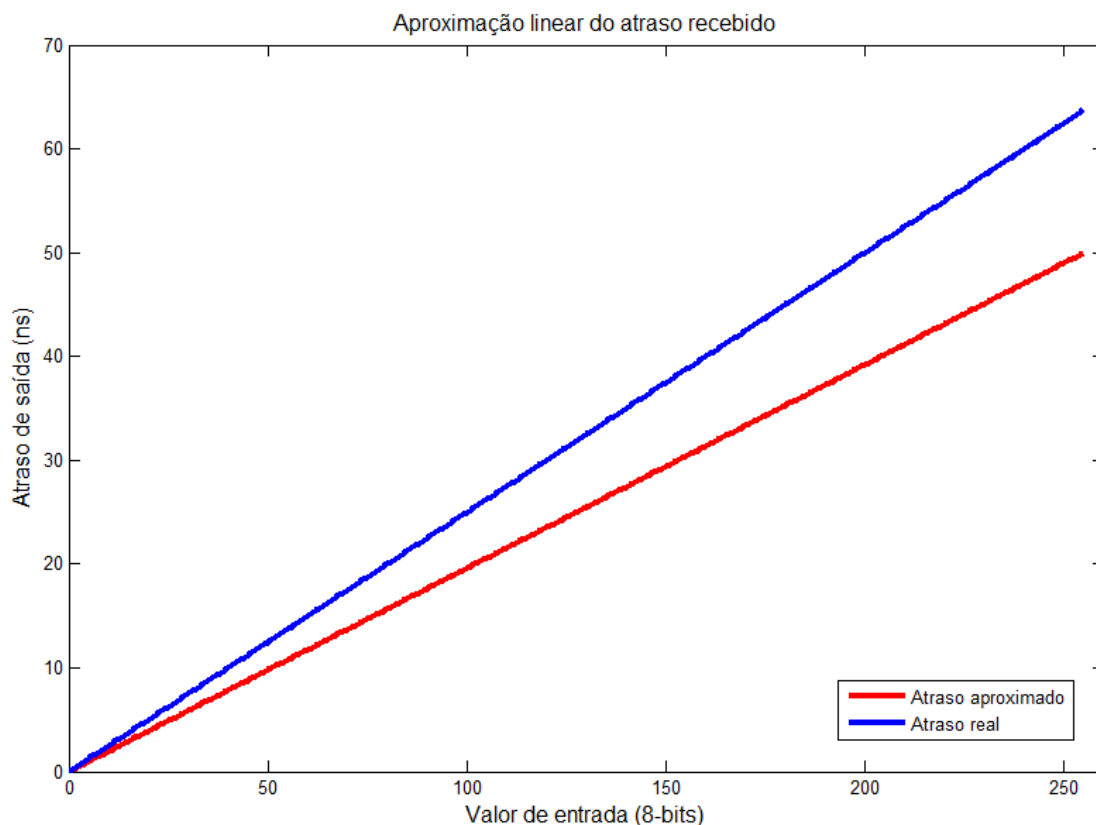


FIGURA 27 – Aproximação linear dos valores de atraso.

Analisando a Fig. 27 pode-se dizer que os 255 valores de atraso produzem um atraso máximo de 50ns através da aproximação realizada. Esta aproximação foi realizada internamente ao microcontrolador.

2.2 Porta Lógica OU

Após obter o sinal PWM do microcontrolador, e o sinal PWM atrasado, ambos entram em uma porta lógica OU, sendo que o valor de saída depende unicamente dos valores de entrada. Para este projeto necessitava-se de uma porta lógica rápida, então se utilizou o circuito integrado 74HC32 [13]. Este circuito integrado pode ser visualizado na Fig. 28 (A), bem como seu atraso visto na Tabela 11.



(A) - Configurações de pinos

(B) - Símbolo Lógico

FIGURA 28 – Porta Lógica OU [13].

TABELA 11 – Atraso porta lógica OU [13]

<i>Símbolo</i>	<i>Parâmetro</i>	<i>Condições</i>	<i>Tipo: HC</i>	<i>Unidade</i>
tPHL/ tPLH	Atraso de propagação nA or nB p/ nY	CL = 15 pF VCC =5V	6	ns

Este circuito integrado possui 4 portas lógicas OU conforme visto na Fig. 28 (B), porém foi utilizado neste projeto somente uma delas. O sinal PWM do microcontrolador é colocado na entrada 4A e o sinal PWM atrasado é colocado na entrada 4B, obtendo na saída 4Y o PWM que passará pelo driver (elevando o sinal de 5V para 12V), fazendo o acionamento da chave. A tabela 12 apresenta os pinos de entrada, saída e de alimentação deste circuito.

TABELA 12 – Pinos porta lógica OU [13]

<i>Número dos pinos</i>	<i>Símbolo</i>	<i>Nome e função</i>
1, 4, 9, 12	1A a 4A	Entradas
2, 5, 10, 13	1B a 4B	Entradas
3, 6, 8, 11	1Y a 4Y	Saídas
7	GND	0V
14	VCC	5V

2.3 Algoritmo do Microcontrolador

A programação do microcontrolador PIC 16F877 foi desenvolvida em linguagem assembly e seu funcionamento depende das configurações que foram vistas anteriormente. O funcionamento do microcontrolador é dividido em diferentes etapas que podem ser visualizadas na Fig. 29.

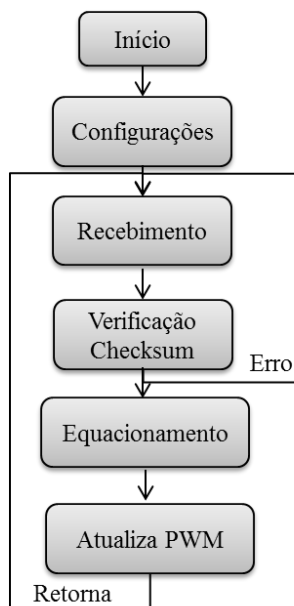


FIGURA 29 – Diagrama de funcionamento do microcontrolador.

Inicialmente é realizada a configuração das portas, definindo as mesmas como entrada ou saída, além disso, configura o módulo PWM e USART do microcontrolador.

Após realizar suas configurações iniciais o programa entra em um loop aguardando recebimento de algum dado em seu pino RC2 (RX). Para esta aplicação deve-se destacar um registrador conhecido como PIR1, visto na Tabela 13.

TABELA 13 – Registrador PIR1

	PSPIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF
Posição	7	6	5	4	3	2	1	0

Neste projeto foi utilizado a posição 5 (RCIF) deste registrador, a qual é responsável por indicar se algum dado foi recebido. O RCIF apresenta o status do buffer de recepção USART, esta posição do registrador vai para '1' lógico quando algum dado é recebido. O valor recebido pelo microcontrolador fica armazenado em outro registrador conhecido como RCREG, sendo necessário atribuir este valor a uma variável declarada no projeto. Após o valor recebido ser movido de RCREG, a posição RCIF do registrador é limpa automaticamente e o programa entra em um próximo loop, aguardando recebimento de um segundo pacote. Estes sinais podem ser visualizados na Fig. 24.

Na Fig. 30 pode-se visualizar a utilização de 4 loops internos ao bloco de recebimento, os quais aguardam o recebimento dos 4 pacotes de dados, o primeiro com os 8 bits mais significativos do PWM, o segundo com os 2 bits menos significativos do PWM, o terceiro com os 8 bits de atraso e um quarto com 8 bits contendo o checksum.

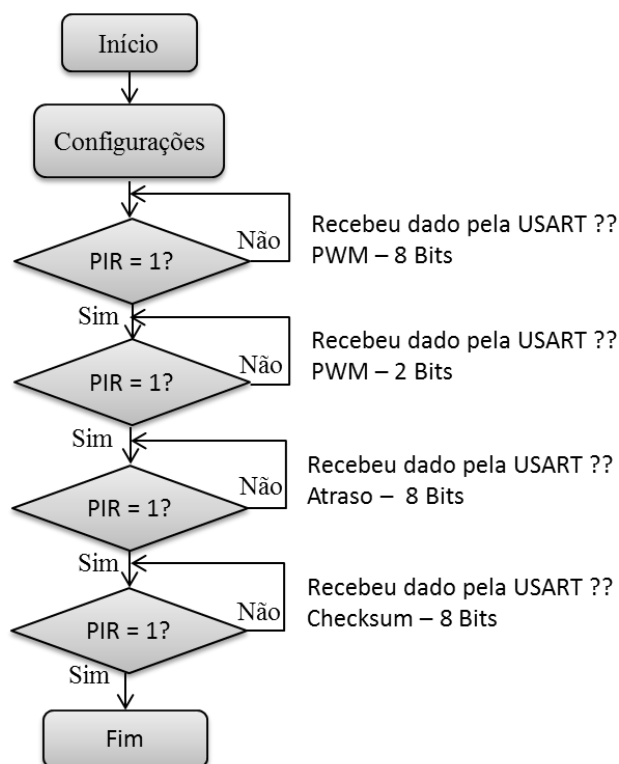


FIGURA 30 – Detalhamento da etapa de recebimento das informações pelo microcontrolador.

Os valores dos 4 pacotes foram gravados em diferentes variáveis declaradas, conforme visto abaixo:

Os 8 bits mais significativos foram movidos para *PACOTE_PWM_MSB*;

Os 2 bits menos significativos foram movidos para *PACOTE_PWM_LSB*;

Os 8 bits do atraso foram movidos para *PACOTE_ATRASO*;

Os 8 bits do checksum são movidos para *PACOTE_CHEKSUM*.

Após receber os 4 pacotes o programa entra no bloco de verificação do checksum que pode ser visualizado na Fig. 31.

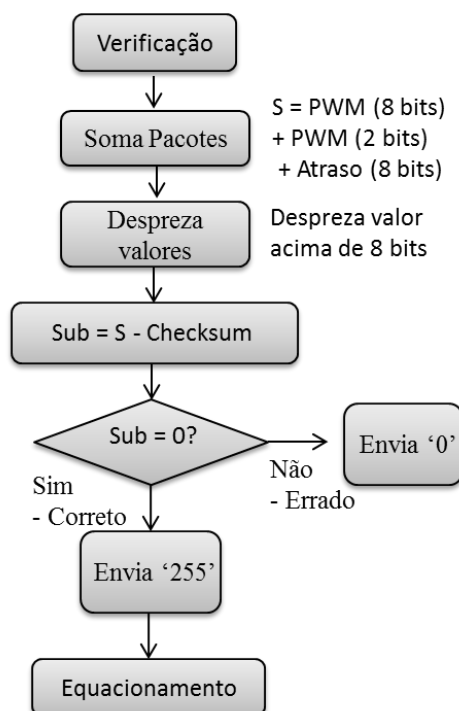


FIGURA 31 – Detalhamento da etapa de verificação dos dados recebidos.

Os 3 pacotes recebidos (*PACOTE_PWM_MSB*, *PACOTE_PWM_LSB* e *PACOTE_ATRASO*) são somados e considerados somente os 8 bits menos significativos desta soma. Este valor é subtraído pelo pacote *PACOTE_CHEKSUM* e posteriormente é realizado um teste sobre o resultado, verificando se o mesmo é zero.

Caso o resultado da subtração seja zero, então o microcontrolador envia um pacote com valor '255' à interface indicando que o valor recebido está correto, pois o valor do *PACOTE_CHEKSUM* é igual aos 8 bits menos significativos da soma dos pacotes de dados recebidos. Caso o resultado seja diferente de zero, então o microcontrolador envia um pacote com valor '0' a interface, indicando assim que o valor recebido está incorreto, sendo assim necessário reenviar os dados ao microcontrolador.

Após o checksum ser realizado com sucesso, é realizada a aproximação linear do pacote de 8 bits enviados pela interface computacional, de forma a produzir um atraso máximo de 50ns. A etapa de equacionamento pode ser visualizada na Fig. 32.

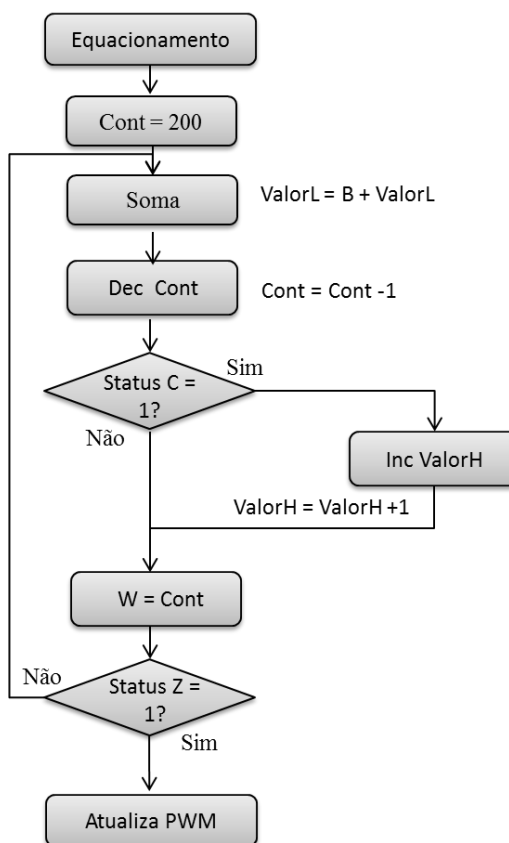


FIGURA 32 – Detalhamento da etapa da aproximação linear realizada no pacote de atraso.

O bloco de equacionamento apresentado na Fig.32 funciona da seguinte maneira:

O valor '200' da Eq. 10 é movido para a variável 'cont' e posteriormente o valor de entrada 'x' (0 a 255) é somado com o valor da variável 'ValorL', decrementando de 'cont', ou seja, a multiplicação é realizada a partir de sucessivas somas. O valor somado é testado no registrador STATUS (posição C), que é uma posição do registrador responsável por indicar se houve overflow do valor somado. Caso o valor seja superior a 8 bits, é incrementado um outro registrador (ValorH). Após o teste no registrador STATUS, é realizado outro teste de verificação do valor de 'cont' no registrador STATUS (posição Z). A posição Z indica se o resultado da última operação foi zero. Caso o valor de 'cont' seja igual a zero, então o programa sai deste loop. Caso contrário, o mesmo permanece somando até que o valor de 'cont' seja zero. Quando a operação de multiplicação terminar, é desprezado o valor contido em 'ValorL'. Desta forma, se estará dividindo a equação por 255. O 'ValorH' é o valor resultante, que posteriormente é movido para a variável *PACOTE_ATRASO_EQUACIONADO*, responsável por atribuir o atraso condicionado ao circuito de atraso. Após obter o valor do atraso aproximado,

o programa do microcontrolador entra na etapa que atualiza o valor do PWM. Este bloco pode ser visualizado na Fig.33.

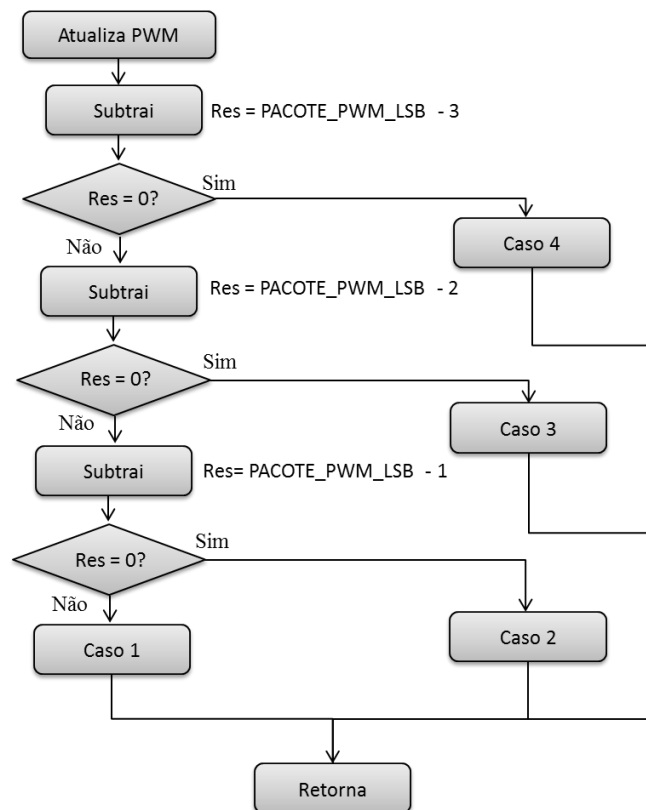


FIGURA 33 – Detalhamento da etapa de atualização do PWM.

A etapa de atualização do PWM tem o objetivo de definir a largura de pulso de saída do modulador tendo como base o valor contido nos registradores (CCPR1L e CCP1CON) que é atribuído pela interface computacional.

Como visto, o sinal gerado com 10 bits de resolução é dado a partir dos 8 bits contidos em CCPR1L e 2 bits do CCP1CON. Nesta etapa são realizados testes sobre o valor dos bits do registrador CCP1CON. Desta forma, além de atribuir o valor ao registrador CCPR1L, este testa as 4 possibilidades dos bits contidos em CCP1CON.

Assim, nesta etapa o microcontrolador atualiza os valores dos 8 bits mais significativos e os 2 menos significativos do módulo PWM, além de inserir o atraso aproximado linearmente de forma paralela através do circuito de atraso.

2.4 Diagrama Esquemático e Protótipo

Através da ferramenta ISIS do Proteus foi realizado o diagrama esquemático a fim de apresentar a interligação dos diferentes componentes que fazem parte do circuito montado em placa de circuito impressa.

A fim de reduzir as interferências e más conexões encontradas na montagem em matriz de contato o projeto foi prototipado em placa e este pode ser visualizado na Fig. 34.

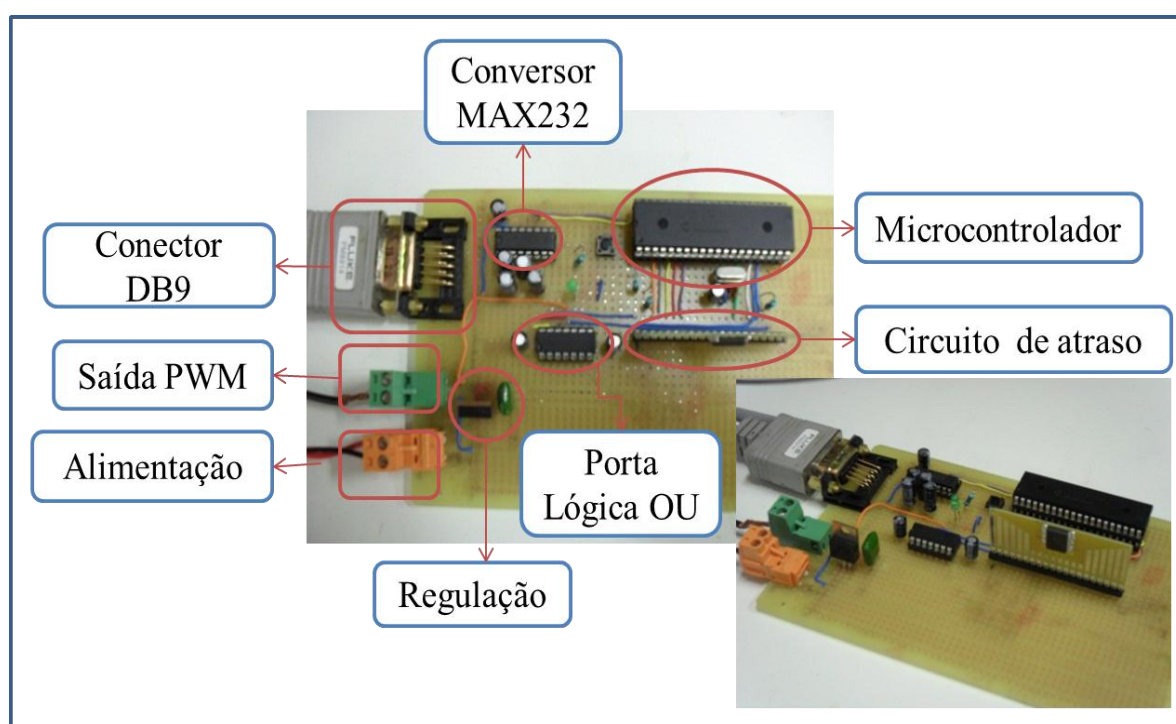


FIGURA 34 – Imagens do protótipo desenvolvido.

Na Fig. 35 podem ser visualizados os diferentes componentes utilizados no projeto, tais como microcontrolador PIC16F877, circuito de atraso DS1020-25, porta lógica 74HC32, conversor MAX232, os bornes de alimentação e saída do PWM e conector DB09 (serial).

No circuito implementado utilizou-se um regulador de tensão LM7805, a fim de fornecer uma alimentação em 5 V. Na saída do regulador e sobre os terminais de alimentação dos componentes envolvidos foram inseridos capacitores do tipo eletrolítico e cerâmico, eliminando assim os ruídos.

3. TESTES E RESULTADOS

Foram realizados testes do circuito PWM a fim de validar o funcionamento do mesmo em placa. Fazendo uso de um osciloscópio puderam-se capturar alguns valores de largura de pulso, os quais correspondem à variação dos dados enviados pela interface gráfica, conforme a Fig. 36.

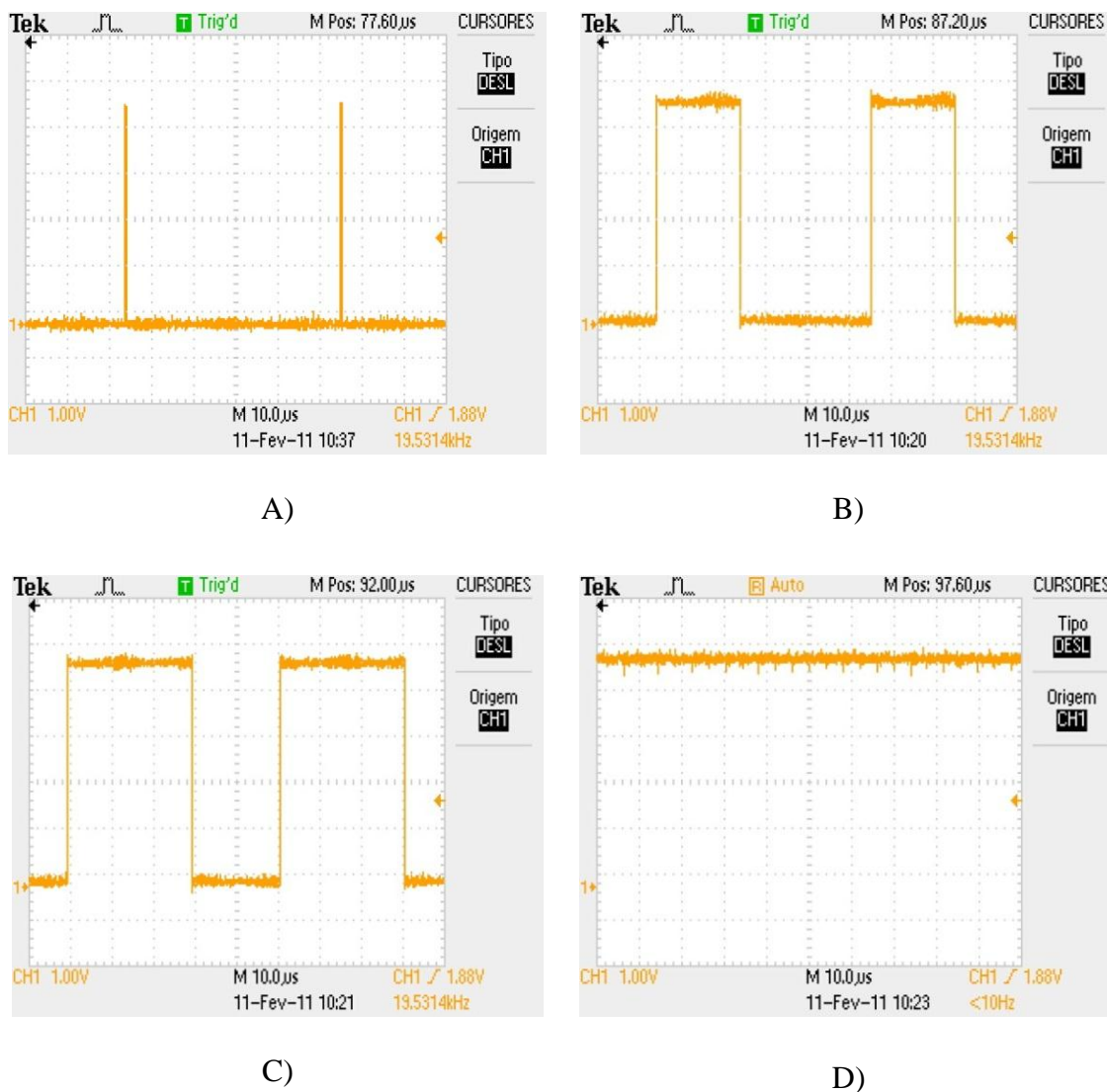


FIGURA 36 – Sinal de saída para alguns valores de duty cycle atribuídos

A) 2%, B) 35%, C) 65%, D) 100% do ciclo ativo.

Conforme visto no decorrer do trabalho, o microcontrolador PIC16F877 possui um PWM com resolução máxima de 10 bits e frequência de chaveamento para a máxima resolução de 19,531 kHz, onde cada valor incrementado no PWM corresponde a um passo de 50ns. Com o osciloscópio em modo de persistência infinita pode-se capturar a borda de descida de alguns passos do PWM gerado pelo microcontrolador, os quais podem ser visualizados na Fig.37.



FIGURA 37 – Sinal de saída obtido a partir de 10 bits de resolução do PWM.

Com a inserção do circuito de atraso, o PWM permaneceu com a mesma frequência de chaveamento, porém passou a ter uma resolução de até 18 bits. Cada passo do PWM correspondia a um acréscimo de 50ns. Porém, com o circuito de atraso passou-se a ter um acréscimo de 0,25ns, que corresponde a um valor 200 vezes menor que o passo antigo. Com o osciloscópio em modo de persistência infinita pode-se capturar a borda de descida do PWM com 18 bits de resolução, conforme visualizado na Fig.38.

Analisando a Fig.38 percebe-se que existe um espaço inicial entre o sinal PWM gerado pelo microcontrolador e o sinal PWM de saída. Isto ocorre devido à latência do circuito de atraso, somado ao atraso da porta lógica OU. O atraso total chega a aproximadamente 16ns. Este atraso inicial influencia apenas nos casos extremos de duty cycle (valor máximo e mínimo). Para o valor mínimo, o primeiro sinal PWM é dado depois de 66ns, resultado da soma dos 50ns do PWM e, 16ns do circuito de atraso e porta lógica. Para o valor máximo, não se teria problema com este microcontrolador, pois o sinal PWM do microcontrolador nunca chega a 100% do ciclo de trabalho. Assim, os valores que ultrapassariam acabam completando o

espaço que faltaria para chegar a 100%. Nas fontes de corrente nunca se trabalha nos casos extremos, ficando limitado entre aproximadamente 10% a 90% do ciclo de trabalho.

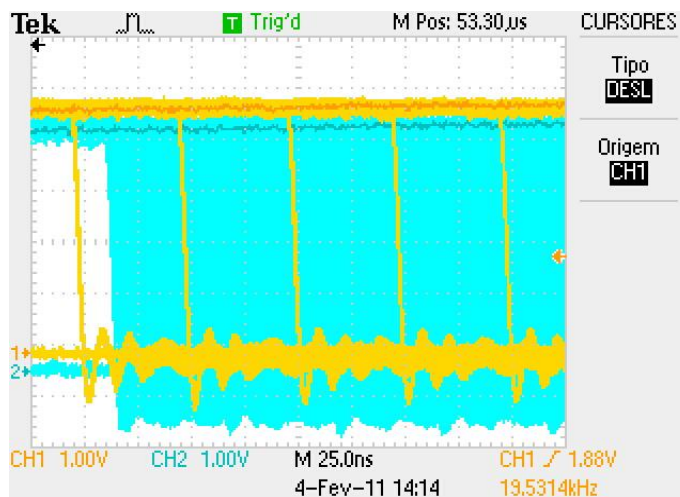


FIGURA 38 – Sinal de saída obtido a partir de 18 bits de resolução do PWM.

Na Fig.38 existem 200 valores discretos de duty cycle (pela aproximação linear) em um intervalo de $50\mu\text{s}$. Porém, os passos de $0,25\mu\text{s}$ não podem ser visualizados, pois o equipamento apresenta uma escala mínima horizontal de $2,5\mu\text{s}$. A Fig. 39 mostra os valores discretos do PWM para uma escala mínima.

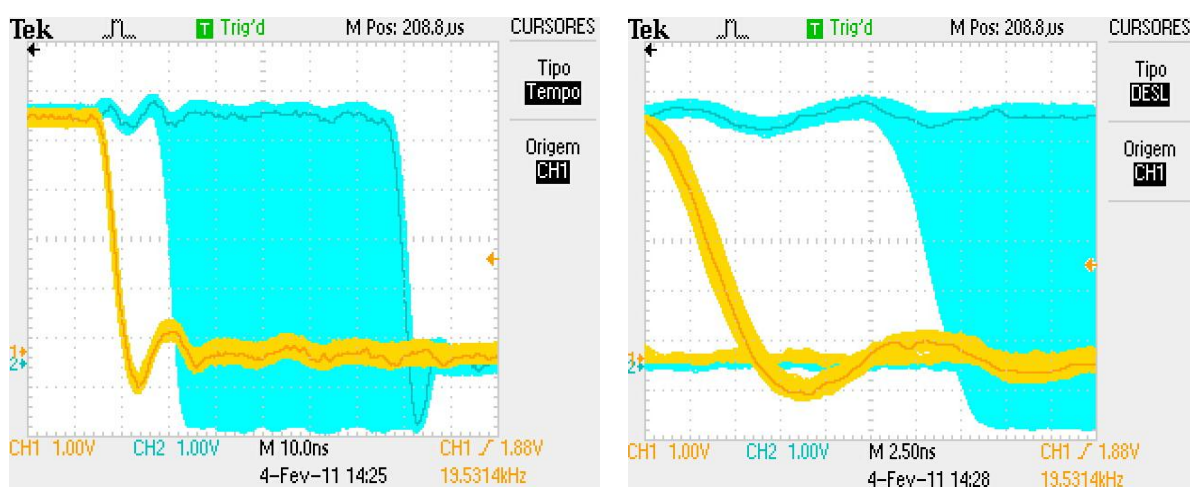


FIGURA 39 – Sinal de saída do PWM com passos de $0,25\mu\text{s}$.

3.1 Análises e Teste Prático

Os microcontroladores encontrados hoje no mercado tem um funcionamento em até 80MHz. Sabendo-se que o incremento do PWM está relacionado a esta frequência, então se pode dizer que estes microcontroladores trabalhando com uma resolução máxima de 18 bits estariam chaveando em aproximadamente 306 Hz, frequência consideravelmente muito baixa. Este valor é obtido calculando o período para uma frequência de 80 MHz multiplicada pelos 18 bits de resolução (2^{18}), correspondendo assim o período total de saída. Aplicando o inverso ao período total de chaveamento se chegaria à frequência de aproximadamente 306 Hz.

Para que um microcontrolador apresente passos de 0,25ns, seria necessário um microcontrolador operando em 4 GHz, frequência inviável nos dias de hoje.

A fim de se comprovar a resolução do sistema, foram realizados testes práticos do PWM de 18 bits desenvolvido. Os testes foram realizados sobre um IGBT que controla a tensão/corrente entregue a uma carga RL, conforme Fig. 40. Vale destacar que os testes práticos foram realizados em malha aberta, e para ciclo de trabalho de aproximadamente 10 a 15%, pois o aumento da corrente provocaria um aquecimento da carga RL e, conseqüentemente, na mudança de impedância, ocasionando falta de monotonicidade da corrente de saída.

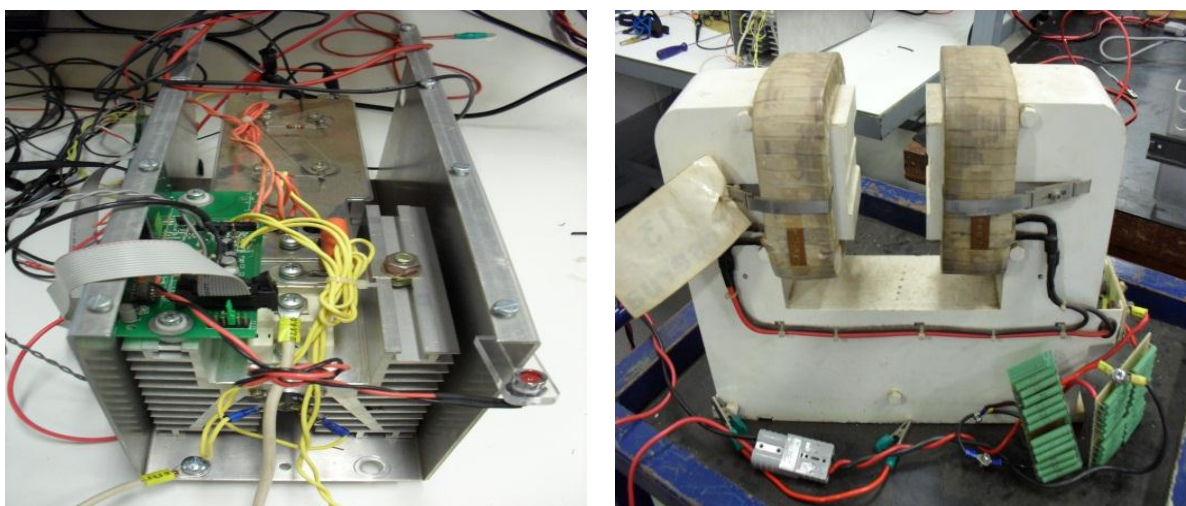


FIGURA 40 – Imagens do driver, IGBT e carga RL (Bobina Corretora de Órbita e Resistências) utilizados para teste.

3.2 Resultados Obtidos por Medição

Para análise do comportamento da corrente em função da variação do PWM foram realizadas medições, podendo-se assim determinar a corrente a cada passo do valor do PWM atribuído.

Considerando que para um PWM máximo se tem uma tensão entregue à carga de 30 V, cuja carga apresenta uma impedância constante de $0,79\Omega$, tem-se uma corrente máxima de 38A a partir da Eq.11. O valor da impedância foi definido através do cálculo de uma corrente e tensão medida sobre a carga.

$$I_{max} = \frac{V_{max}}{Z} \quad (11)$$

Sendo a corrente máxima de carga (I_{max}), a impedância de carga (Z) e tensão máxima de alimentação (V_{max}). Cada incremento do PWM deve apresentar uma variação de aproximadamente $185\mu A$, obtida a partir da Eq. 12.

$$\Delta i = \frac{I_{max}}{Resolução} \quad (12)$$

Sendo a corrente máxima da carga (I_{max}) e a resolução (obtida a partir dos 1023 passos do PWM (microcontrolador), multiplicado pelos 200 valores de atraso entre 2 passos do PWM (microcontrolador), obtendo assim 204.600 valores).

As medidas de corrente sobre a carga foram obtidas através de um DCCT, que realiza medida isolada da corrente. O DCCT possui uma relação de medida de 20A/10V. Um multímetro de 6,5 dígitos de precisão modelo 34970A da HP foi utilizado para realizar a medição da tensão de saída do DCCT.

O valor do duty cycle foi atualizado a cada 3 segundos pela interface computacional, sendo realizada medidas com o multímetro a cada 0,5 segundos, o que representa 6 medidas por largura de pulso ajustada.

A corrente de saída em função da variação do duty cycle pode ser visualizada na Fig. 41. Pode-se perceber que houve uma linearidade da corrente em função da variação do duty cycle. Aplicando um zoom em determinada região da reta pode-se verificar as medidas realizadas, bem como a variação de corrente em função da variação do duty cycle. A Fig. 42 ilustra os detalhes da medição.

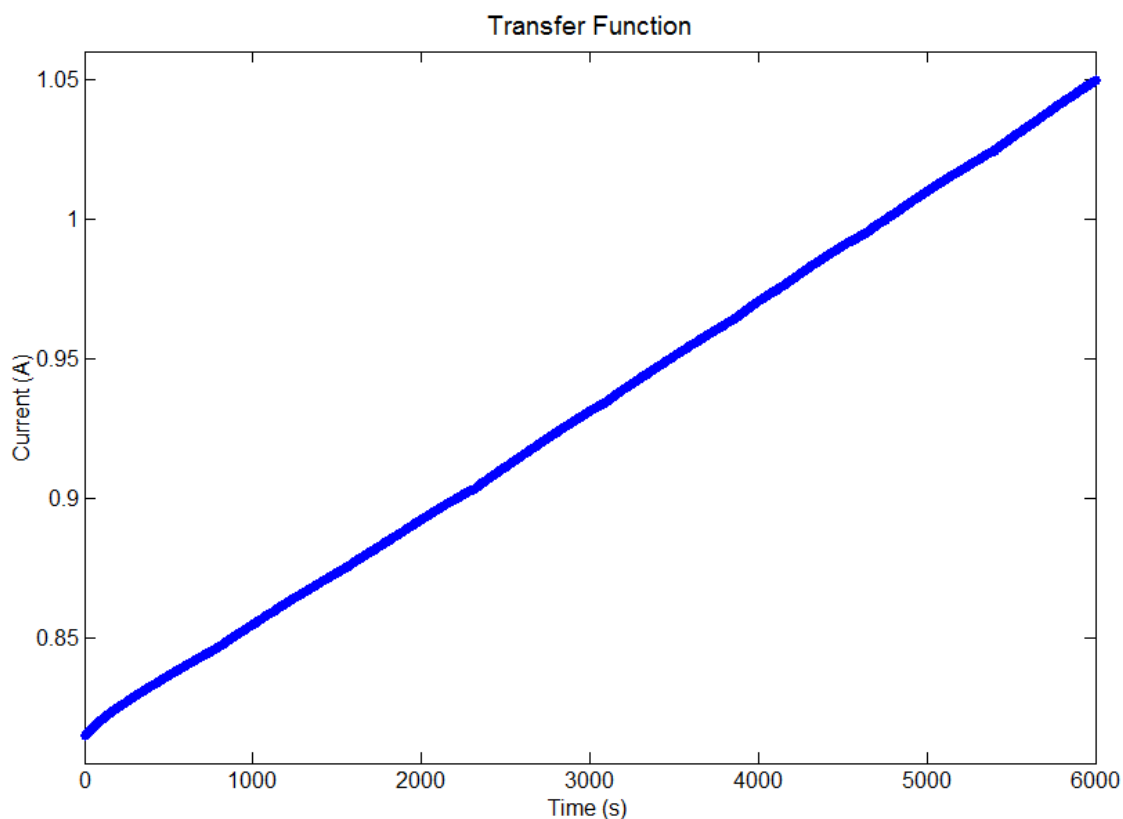


FIGURA 41 – Variação da corrente em função da variação do duty cycle.

Pode-se perceber que entre 0.8235A e 0.8245A obteve-se aproximadamente 6 passos completos de corrente. Assim, cada passo representa um acréscimo de aproximadamente 167 μ A, valor muito próximo ao calculado.

Verifica-se que se repete a leitura da corrente de tempos em tempos. Isto ocorre devido à aproximação linear realizada no projeto, ou seja, a largura de pulso não é alterada para alguns valores de atrasos atribuídos pela interface computacional.

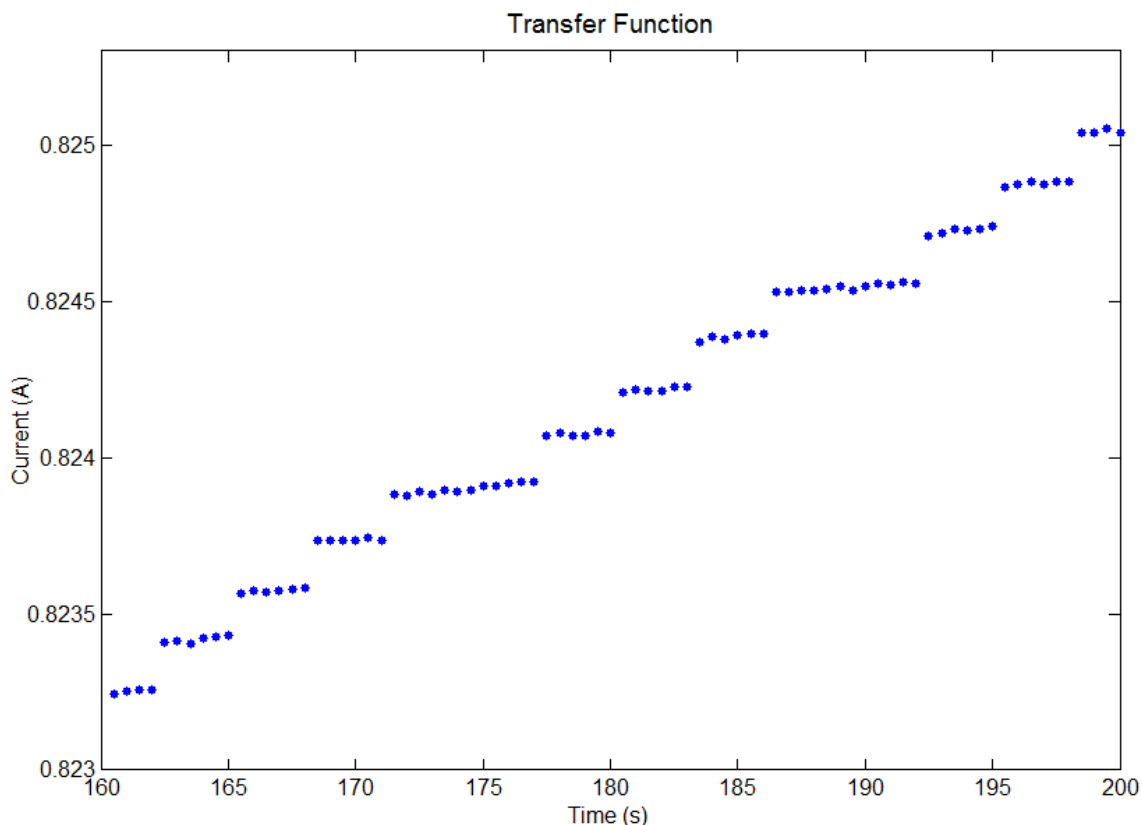


FIGURA 42 – Detalhes da variação de corrente em função da variação do duty cycle.

Outro aspecto interessante de se analisar é o comportamento da corrente de saída quando ocorre o incremento do sinal PWM do microcontrolador. Com esta análise é possível identificar se os 200 valores de atraso aproximados linearmente ficaram exatamente entre dois intervalos de 50ns do PWM, sendo assim possível verificar se a corrente de saída mantém sua monotonicidade.

Conforme visto, o circuito de atraso possui um atraso total de $63,75\text{ns}$, que é aproximado de forma linear, obtendo uma saída máxima de atraso de 50ns . Assim todos os casos possíveis de atraso são inseridos entre dois intervalos.

Na prática observou-se que para alguns valores de incremento dos bits menos significativos do PWM ocorria falta de monotonicidade, ou seja, em certos momentos a corrente decrescia, conforme mostrado na Fig. 43.

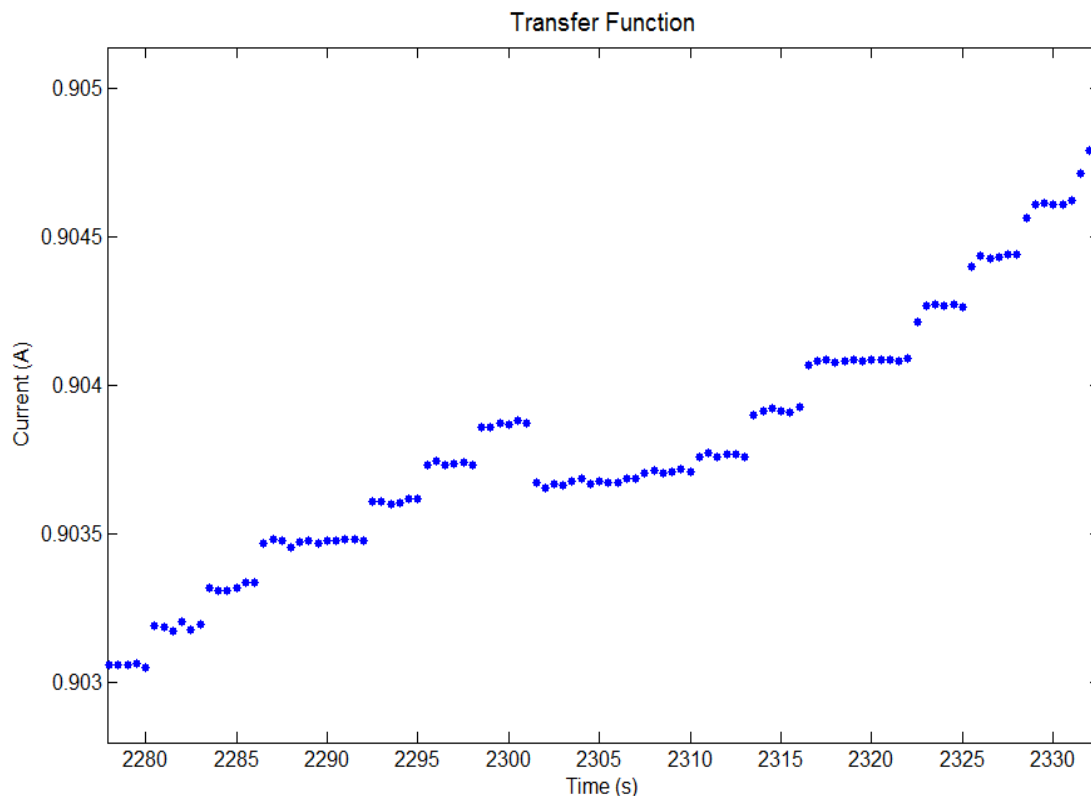


FIGURA 43 – Detalhes da corrente de saída sem monotonicidade.

O problema encontrado era causado pelos passos dos dois bits menos significativos do PWM, pois em algumas combinações destes o PWM aplicava um passo inferior aos 50ns esperados. Apesar da diferença pequena, para esta aplicação o erro inserido foi considerável. Quando o passo do PWM é menor, a corrente de saída decai e posteriormente, começa a subir novamente. Para solucionar este problema foi necessário realizar uma correção através da programação do microcontrolador.

Foi atribuído para alguns valores de alteração dos bits menos significativos um tempo maior de atraso, de forma a manter repetitividade e monotonicidade da corrente de saída. Após a correção foi analisada novamente a corrente através de medidas e o resultados obtidos podem ser observados na Fig. 44.

Comparando o sistema digital proposto com 18 bits de resolução em relação aos sistemas analógicos de 16 bits pode-se verificar uma melhoria significativa. Para uma corrente máxima de 38 A, com os sistemas analógicos se obtém passos de corrente de $579\ \mu\text{A}$, enquanto que para o sistema proposto consegue-se atingir passos de $185\ \mu\text{A}$, que corresponde a um valor 3,2 vezes menor.

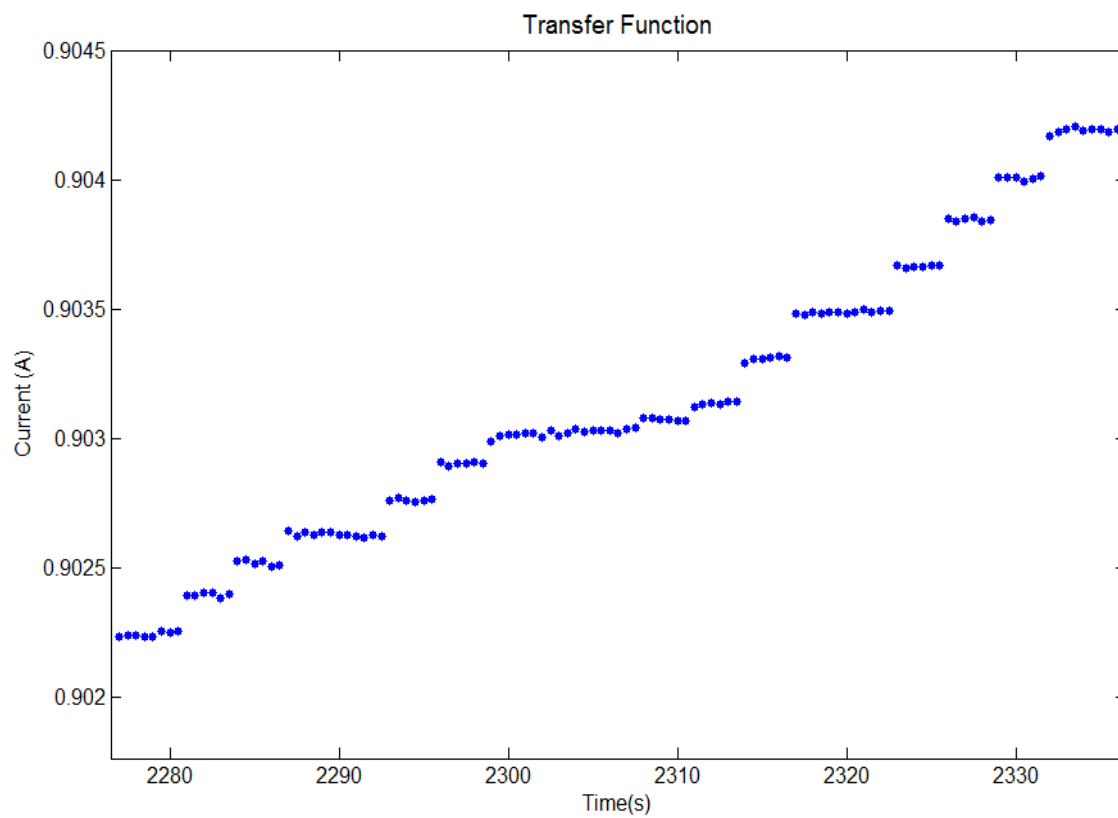


FIGURA 44 – Corrente de saída com monotonicidade após correção.

CONSIDERAÇÕES FINAIS

Neste projeto foi apresentada a implementação de um PWM digital de alta resolução e frequência tendo como base um circuito de atraso. Este projeto fará parte de um sistema de regulação digital que poderá substituir os sistemas de regulação analógicos encontrados atualmente nas fontes de corrente do Laboratório Nacional de Luz Síncrotron. As fontes de corrente são responsáveis por alimentarem diversos magnetos que compõem o anel de armazenamento de elétrons.

As variações de temperatura ambiente, bem como os ruídos fazem com que o sistema de regulação analógico perca precisão, o que causa diretamente perda de precisão dos magnetos que atuam sobre o feixe de elétrons que circulam no anel com uma velocidade muito próxima a da luz. Já um sistema digital pode solucionar estes problemas, pois os sinais que estes sistemas trabalham são discretos. Porém um sistema digital apresenta um problema na relação entre resolução e frequência, pois quando a resolução é aumentada, a frequência de chaveamento diminui.

Então este projeto visou aumentar a resolução de um modulador PWM sem alterar a frequência de chaveamento das chaves que compõem o conversor da fonte de corrente através da aplicação de um circuito de atraso.

Utilizado um microcontrolador PIC16F877 que possui um módulo de PWM com resolução máxima de 10 bits, trabalhando em 20MHz, este apresentava passos em seu PWM de 50ns. Aplicando um circuito de atraso DS1020-25 pode-se inserir diferentes valores de atrasos entre dois passos do PWM do microcontrolador, estes variados em passos de 0,25ns. A ideia do projeto foi inserir 200 valores de atraso entre dois passos do PWM gerado pelo microcontrolador.

Com este sistema digital pode-se atingir até 18 bits de resolução permanecendo a frequência de chaveamento constante em 19,531 kHz.

A fim de se realizar testes com o sistema digital implementado foi desenvolvida uma interface computacional em Matlab, o que permitia enviar informações para o hardware através da comunicação serial, segundo o padrão RS232. Foi inserido um conversor MAX232 entre o

computador e o hardware, este convertia os níveis de tensão do sinal padrão RS232 para o nível TTL, que tem um sinal de saída entre 0 a 5V.

O projeto foi prototipado em placa a fim de reduzir os ruídos e más conexões encontradas na montagem em matriz de contato. Após prototipado, o projeto foi interligado a um driver que, posteriormente, acionou uma chave do tipo IGBT que alimentava uma carga RL. Os dados enviados serialmente a partir de um programa desenvolvido em Matlab incrementava o valor do duty cycle a cada 3 segundos. Assim, pôde-se verificar a variação da corrente sobre a carga de saída com a resolução desejada.

Fazendo uso de um DCCT pode-se realizar medida de corrente/tensão. Este realizava medidas a cada 0,5 segundos, correspondendo a 6 medidas por valor de duty cycle. Os dados medidos foram plotados, sendo assim possível observar o comportamento da corrente de saída em função da variação do duty cycle. Analisando as curvas plotadas, pôde-se observar que a corrente de saída apresentou monotonicidade durante todo o tempo de medição, após a correção devida à baixa precisão do PWM do microcontrolador.

Após as medições serem realizadas e analisadas, o projeto foi validado, apresentando melhoria significativa sobre os sistemas atuais de 16 bits, onde se pode variar a corrente em passos 3,2 vezes menor que o passo encontrado nos sistemas atuais, correspondendo uma resolução satisfatória.

O projeto deverá no futuro ser testado em um sistema de regulação digital, que atuará como o sistema de controle de uma fonte de corrente.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] SEVERO, Lucas; RODRIGUES, Cleber; FURLAN, Rodrigo. **Implementação de um Sistema Digital de Comunicação para Controle de uma Fonte de Corrente**. 19º Programa Bolsa de Verão no Laboratório Nacional de Luz Síncrotron.
- [2] RODRIGUES, Cleber; SILVA, Ângelo. **A Bipolar DC Power Supply Using Full-Bridge Topology as Active Shunt for Magnets**, in Proc. Of COBEP, 2009.
- [3] YOUNG, Paul. **Técnicas de Comunicação Eletrônica**. Editora: Pearson, 5º edição, São Paulo.
- [4] POMILIO, J. A. **Técnicas de Modulação em Fontes Chaveadas**. Capítulo 3. Universidade Estadual de Campinas – UNICAMP.
- [5] NEVES, Raphael; MARQUES, Jeferson; MONTAGNER, Vinícius; GIRARDI, Alessandro. **Projeto e prototipação de um gerador de PWM digital configurável em linguagem de descrição de hardware**. Centro de Tecnologia de Alegrete, I Sipe, 2007.
- [6] FOROUZAN, Behrouz. **Comunicação de Dados e Redes de Computadores**. Editora: Mc Graz Hill, 4º edição, São Paulo.
- [7] CANZIAN, Edmur. **MINICURSO - Comunicação Serial - RS232**. CNZ Engenharia e Informática Ltda, 2006.
- [8] DALLAS Semiconductor. **DS1020 (Programmable 8-Bit - Silicon Delay Line)**, disponível em:
<<http://pdf1.alldatasheet.com/datasheet-pdf/view/89485/DALLAS/DS1020.html>>, acesso em janeiro de 2011.
- [9] SINHA, Priyabrata. **PLL Jitter and its Effects on Technology Communications**. Microchip Technology.
- [10] MATLAB. **The Language Of Technical Computing**, disponível em:
<<http://www.mathworks.com/products/matlab/>>, acesso em janeiro de 2011.
- [11] TEXAS INSTRUMENTS. **Conversor MAX232 (Dual EIA-232)**, disponível em:

<http://www.datasheetcatalog.org/datasheet/texasinstruments/max232.pdf>, acesso em janeiro de 2011.

[12] MICROCHIP. **Microcontrolador PIC16F877**, disponível em:

<http://www.datasheetcatalog.org/datasheets2/36/3629224_1.pdf>, acesso em janeiro de 2011.

[13] PHILIPS. **Porta Lógica 74HC/HCT32 (Quad 2-input OR gate)**, disponível em:

<http://www.datasheetcatalog.org/datasheet/philips/74HC_HCT32_CNV_2.pdf>, acesso em janeiro de 2011.

[14] MEDEIROS, Júlio Cesar de Oliveira. **Princípios de Telecomunicações: teoria e prática**. Editora Érica Ltda, 2^o edição, São Paulo.

[15] QIU, Yang; LI, Jian; XU, Ming; HA, Dong; LEE, Fred. **Proposed DPWM Scheme with Improved Resolution for Switching Power Converters**. Center for Power Electronics Systems, Virginia Polytechnic Institute and State University, Blacksburg, USA, Applied Power Electronics Conference, APEC 2007 - Twenty Second Annual IEEE.

[16] DANCY, Abram; CHANDRAKASAN, Anantha. **A Reconfigurable Dual Output Low Power Digital PWM Power Converter**. Department of EECS, Massachusetts Institute of Technology, Cambridge, Low Power Electronics and Design, 1998. Proceedings. 1998 International Symposium on.

[17] François Bouvet, Synchrotron Soleil. **Design of the DC and AC Magnet Power Supplies for the Soleil Synchrotron Radiation Source**. Power Electronics and Applications, 2007 European Conference on.

APÊNDICE

Visando facilitar o entendimento das diversas configurações realizadas em alguns registradores do microcontrolador, é feita uma apresentação detalhada destes.

Registrador TXSTA:

CSRC – Seleção entre Master/Slave “configuração em modo síncrono”:

0 – Slave;

1 – Master.

TX9 – Habilitação de 9 bits para a transmissão:

0 – 8 bits;

1 – 9 bits.

TXEN – Habilitação de transmissão:

0 – Desabilitada;

1 – Habilitada.

SYNC – Seleção entre modo assíncrono e síncrono:

0 – Assíncrono;

1 – Síncrono.

BRGH – Seleção para Baud Rate:

0 – Baud Rate Baixo;

1 – Baud Rate Alto.

Observação: deve-se calcular para ambos os modos, verificando qual deles apresenta menor erro.

TRMT – Situação do registrador de transmissão

0 – Cheio;

1 – Vazio.

TX9D – bit pode ser usado como paridade ou endereçamento

Registrador RCSTA:

SPEN – Habilitação de USART:

0 – Desabilitada;

1 – Habilitada.

RX9 – Habilitação de 9 bits para a recepção:

0 – 8 bits;

1 – 9 bits.

SREN – Recepção unitária:

0 – Desabilitada;

1 – Habilitada.

CREN – Habilitação de recepção contínua:

0 – Desabilitada;

1 – Habilitada.

ADDEN – Habilitação de endereçamento:

0 – Desativado;

1 – Habilitado.

FERR – Erro de Stop Bit “Configuração em modo Assíncrono”:

0 – Não ocorreu erro, Stop Bit = 1;

1 – Ocorreu erro, Stop Bit = 0.

OERR – Erros de muitos bytes recebidos sem leitura:

0 – Não houve problemas;

1 – Estouro do Limite de 3 Bytes recebidos.

***RX9D** – bit pode ser usado como paridade ou endereçamento*

Registrador PIR1:

***PSPIF** – Indica interrupção da porta paralela;*

***ADIF** – Indica se a conversão A/D foi concluída;*

***RCIF** – Indica se o buffer de recepção USART está cheio, ou vazio;*

***TXIF** – Indica se o buffer de transmissão USART está cheio, ou vazio.*

***SSPIF** – Indica interrupção da porta serial SSP;*

***CCPIIF** – Indica interrupção no módulo CCP1;*

***TMR2IF** – Indica interrupção do Timer2;*

***TMR1IF** – Indica interrupção de estouro do Timer1.*

Registrador T2CON:

TOUTPS3 à TOUTPS0 estão relacionados ao ajuste do postscale, onde se atribuiu valor de ‘0000’ que corresponde a um postscale de relação 1:1.

TRM2ON – Ativa habilitação do TMR2:

1 – Ligado;

0 – Desligado.

T2CKPS1 e T2CKPS0 estão relacionados ao ajuste do prescale, onde se atribui o valor de ‘00’ que corresponde a um prescale de relação 1:1.

ANEXOS

O algoritmo do microcontrolador foi desenvolvido em linguagem assembly, nele é apresentada a configuração dos diversos tipos de registradores utilizados no projeto, além de todo o funcionamento.

Código Assembly

```
=====
===== CONFIGURAÇÃO DO MICROCONTROLADOR =====
=====

__CONFIG _CP_OFF & _CPD_OFF & _DEBUG_OFF & _LVP_OFF & _WRT_ENABLE_OFF &
_BODEN_OFF & _PWRTE_ON & _WDT_OFF & _HS_OSC
#include <P16F877.INC>          ; MICROCONTROLADOR UTILIZADO

ERRORLEVEL -302
ERRORLEVEL -305
ERRORLEVEL -306

CBLOCK      0X20
PACOTE_CHECKSUM
PACOTE_PWM_MSB
PACOTE_PWM_LSB
PACOTE_ATRASO
TESTE
BYTE
VALOR_L
VALOR_H
PACOTE_ATRASO_EQUACIONADO
ENDC
#define BANK1 BSF    STATUS,RP0
#define BANK0 BCF    STATUS,RP0
#define PARALELA    PORTD
#define LIGADO      PORTE,0
ORG    0X0000
GOTO CONFIGURACAO
CONFIGURACAO
CLRF    PORTA
```

```

CLRF PORTB
CLRF PORTC
CLRF PORTD
CLRF PORTE
BANK1
    MOVLW    B'11111111'
    MOVWF    TRISA
    MOVLW    B'11111111'
    MOVWF    TRISB
    MOVLW    B'10111011'
    MOVWF    TRISC
    MOVLW    B'00000000'
    MOVWF    TRISD
    MOVLW    B'00000000'
    MOVWF    TRISE
    MOVLW    .255
    MOVWF    PR2
    MOVLW    B'00100100'
    MOVWF    TXSTA
    MOVLW    .129
    MOVWF    SPBRG

```

```

BANK0
    MOVLW    B'10010000'
    MOVWF    RCSTA
    MOVLW    .0
    MOVWF    CCPR1L
    MOVLW    B'00001111'
    MOVWF    CCP1CON
    MOVLW    B'00000000'
    MOVWF    CCP2CON
    MOVLW    B'00000100'
    MOVWF    T2CON
    BSF     LIGADO

```

LOOP

```

=====
===== RECEBIMENTO DE DADOS PELA SERIAL =====
=====

```

```

    BTFS    PIR1,RCIF    ; VERIFICA BUFFER DE RECEPÇÃO '0' VAZIO
    GOTO    $-1
    MOVF    RCREG,W

```



```

MOVWF      PACOTE_PWM_MSB
BTFSS PIR1,RCIF      ; VERIFICA BUFFER DE RECEPÇÃO '0' VAZIO
GOTO  $-1
MOVF  RCREG,W
MOVWF      PACOTE_PWM_LSB
BTFSS PIR1,RCIF      ; VERIFICA BUFFER DE RECEPÇÃO '0' VAZIO
GOTO  $-1
MOVF  RCREG,W
MOVWF      PACOTE_ATRASO
BTFSS PIR1,RCIF      ; VERIFICA BUFFER DEV RECEPÇÃO '0' VAZIO
GOTO  $-1
MOVF  RCREG,W
MOVWF      PACOTE_CHECKSUM

```

```

=====
===== CHECKSUM =====
=====

```

```

MOVFW PACOTE_PWM_MSB
ADDWF PACOTE_PWM_LSB,W
ADDWF PACOTE_ATRASO,W
MOVWF TESTE          ; IGNORA OS VALORES ACIMA DE 8 BITS
MOVFW TESTE
SUBWF PACOTE_CHECKSUM,W
ADDLW .0
BTFSS STATUS,Z      ;RESULTADO DA OPERAÇÃO = 0 ??
GOTO  ERRO          ;NÃO
                        ;SIM
MOVLW .255          ;ENVIA '255' P/ A INTERFACE, CORRETO !!
CALL TRANSMITE_TX
CALL EQUACIONAMENTO

```

```

=====
===== ATUALIZANDO PWM =====
=====

```

```

MOVFW PACOTE_PWM_LSB
SUBLW .3
BTFSC STATUS,Z      ;RESULTADO DA OPERAÇÃO = 0 ??
GOTO PASSO_150NS    ;SIM
                        ;NÃO
MOVFW PACOTE_PWM_LSB

```

```

SUBLW .2
BTFSZ STATUS,Z           ;RESULTADO DA OPERAÇÃO = 0 ??
GOTO PASSO_100NS        ;SIM
                          ;NÃO

MOVFW PACOTE_PWM_LSB
SUBLW .1
BTFSZ STATUS,Z           ;RESULTADO DA OPERAÇÃO = 0 ??
GOTO PASSO_50NS         ;SIM
                          ;NÃO

MOVFW PACOTE_PWM_LSB
SUBLW .0
BTFSZ STATUS,Z           ;RESULTADO DA OPERAÇÃO = 0 ??
GOTO PASSO_PWM          ;SIM
                          ;NÃO

```

PASSO_PWM

```

BCF  CCP1CON,5
BCF  CCP1CON,4
MOVFW PACOTE_PWM_MSB
MOVWF CCPR1L
MOVFW PACOTE_ATRASO_EQUACIONADO
MOVWF PARALELA
GOTO LOOP

```

PASSO_50NS

```

BCF  CCP1CON,5
BSF  CCP1CON,4
MOVFW PACOTE_PWM_MSB
MOVWF CCPR1L
MOVFW PACOTE_ATRASO_EQUACIONADO
MOVWF PARALELA
GOTO LOOP

```

PASSO_100NS

```

BSF  CCP1CON,5
BCF  CCP1CON,4
MOVFW PACOTE_PWM_MSB
MOVWF CCPR1L
MOVFW PACOTE_ATRASO_EQUACIONADO
MOVWF PARALELA
GOTO LOOP

```

PASSO_150NS

```

BSF  CCP1CON,5

```

