



Identificação de baterias através da comunicação pela linha de alimentação DC

VASCO MANUEL DIAS CORREIA

novembro de 2020

IDENTIFICAÇÃO DE BATERIAS ATRAVÉS DA COMUNICAÇÃO PELA LINHA DE ALIMENTAÇÃO DC

Vasco Manuel Dias Correia



Mestrado em Engenharia Eletrotécnica e de Computadores

Área de Especialização de Telecomunicações

Departamento de Engenharia Eletrotécnica

Instituto Superior de Engenharia do Porto

2020

Este relatório satisfaz, parcialmente, os requisitos que constam da Ficha de Disciplina de Tese/Dissertação, do 2º ano, do Mestrado em Engenharia Eletrotécnica e de Computadores

Candidato: Vasco Manuel Dias Correia, Nº 1131231, 1131231@isep.ipp.pt

Orientação científica: Lino Manuel Baptista Figueiredo, lbf@isep.ipp.pt

Orientação científica: António José Matos De Meireles, aem@isep.ipp.pt



Mestrado em Engenharia Eletrotécnica e de Computadores

Área de Especialização de Telecomunicações

Departamento de Engenharia Eletrotécnica

Instituto Superior de Engenharia do Porto

27 de novembro de 2020

Agradecimentos

Em primeiro lugar gostaria de agradecer à minha mãe e aos meus avós por todo o apoio, segurança e suporte que me deram ao longo deste percurso académico que aqui termina. Não podia pedir mais deles.

Agradeço aos meus dois orientadores pela orientação desta dissertação, ao professor Antonio Meireles e o professor Lino Figueiredo.

Gostaria igualmente, mas não menos importante à minha namorada Rita por aturar todo o meu mau humor e stress provocados pelo desenvolvimento desta dissertação, tendo sido a pessoa com quem partilho as minhas frustrações, o amor é mesmo cego. Sem ela não teria chegado ao fim.

Esta dissertação tendo sido desenvolvida durante uma viagem de trabalho prolongada a Inglaterra, tenho também de agradecer aos meus colegas de trabalho Afonso e Zeferino, pela paciência e principalmente pela ajuda nas lides domésticas e na cozinha quando eu tinha de me dedicar à dissertação, no horário pós-laboral.

Por último, mas possivelmente a mais importante tenho de agradecer ao meu amigo Rodolfo, sem o qual, esta dissertação não ficaria concluída, pois com uma situação de pandemia e com a minha estadia em Inglaterra, ele foi sempre a minha ponte a Portugal, e particularmente ao ISEP. Uma lição de amizade.

Resumo

Para uma identificação de baterias utilizadas em veículos industriais é analisada a hipótese da utilização da tecnologia PLC. Esta tecnologia permite uma fácil integração e um menor investimento, pois utiliza a linha de alimentação já existente para comunicar. Até à data, a transmissão numa linha DC nunca foi muito explorada. No entanto nos últimos anos têm surgido equipamentos que permitem estudar um sistema de comunicação utilizando esta linha.

Nesta dissertação foi feito um estudo sobre as características e protocolos PLC existentes, seguidos de uma pesquisa sobre os equipamentos existentes capazes de integrar um sistema aplicando a tecnologia PLC. Foram desenvolvidos um protótipo e um código que satisfaz as funcionalidades impostas pelo equipamento. Foram também realizados dois testes, um ao código com a análise da mensagem a enviar e outro ao *driver de comunicação* utilizado no protótipo desenvolvido.

Os resultados dos testes indicam que o código desenvolvido funciona. Contudo, o teste ao integrado não apresentou resultados conclusivos.

Palavras-Chave

PLC, AFE031, PWM, SPI, protótipo.

Abstract

To identify batteries used on industrial vehicles it's analysed the hypothesis of using the PLC technology. This technology allows an easy integration and a low investment because it uses the power line to communicate data. Despite that in the last few years new equipment has been released that allows to study the possibility of a communication system using the power line.

In this dissertation a study has been made about the PLC characteristics and the existing protocols, followed by a research about the equipment's on the market capable of offering a solution for this goal. A prototype and the respective code compatible with the equipment chosen. A couple of tests has been made, one testing the code related to message sent and another to the prototype.

The test made show that the code that has been developed works, sending the intended message. The test made to the prototype does not show good results.

Keywords

PLC, AFE031, PWM, SPI, prototype

Índice

AGRADECIMENTOS	I
RESUMO	III
ABSTRACT	V
ÍNDICE	VII
ÍNDICE DE FIGURAS	IX
ÍNDICE DE TABELAS	XI
ACRÓNIMOS	XIII
1. INTRODUÇÃO	1
1.1. CONTEXTUALIZAÇÃO	2
1.2. OBJETIVOS.....	2
1.3. CALENDARIZAÇÃO	3
1.4. ORGANIZAÇÃO DO RELATÓRIO	3
2. COMUNICAÇÃO PELA LINHA DE ALIMENTAÇÃO DC	5
2.1. CARACTERÍSTICAS DO CANAL EM PLC	6
2.2. PLC NA INDÚSTRIA	7
2.3. COMUNICAÇÃO.....	10
2.4. MODULAÇÃO OFDM.....	13
2.5. PROTOCOLOS PLC	22
3. PROTÓTIPO	34
3.1. AFE031	35
3.2. DESENVOLVIMENTO	38
3.3. ATMEGA 328P	45
4. IMPLEMENTAÇÃO	47
4.1. COMUNICAÇÃO.....	47
5. TESTES E RESULTADOS	65
5.1. TESTE MICROCONTROLADOR – ANALISADOR LÓGICO	65
5.2. TESTE AFE031	69

6. CONCLUSÕES	73
REFERÊNCIAS DOCUMENTAIS.....	75
ANEXO A. – ESQUEMA ELÉTRICO DO PROTÓTIPO	77
ANEXO B. – CÓDIGO DESENVOLVIDO.....	79

Índice de Figuras

Figura 1	Diagrama de integração do ruído no sistema	7
Figura 2	Comunicação SPI[5]	12
Figura 3	Formas de onda de uma ligação SPI[5]	13
Figura 4	Espectro de frequências OFDM[6].....	14
Figura 5	Largura de banda utilizada por OFDM e FDM[6]	15
Figura 6	Exemplo de ortogonalidade no domínio do tempo.....	16
Figura 7	Subportadoras piloto sobre o campo das frequências (<i>frequency</i>)[8].....	17
Figura 8	Diagrama de blocos de transmissão OFDM[6]	18
Figura 9	Diagrama explicativo da ISI, devido ao multipercurso[6].	19
Figura 10	Tempo de guarda e prefixo cíclico[8]	20
Figura 11	BFSK no domínio do tempo[9]	20
Figura 12	Diagrama de transmissor simplificado FSK[9]	21
Figura 13	Diagrama de transmissão FSK[9].....	22
Figura 14	Modelo OSI[12]	23
Figura 15	Pilha de protocolos norma PRIME[1]	23
Figura 16	Processamento da PPDU[11]	24
Figura 17	Codificador convolucional[11].....	25
Figura 18	Formato da trama[11].....	26
Figura 19	Pilha de protocolos G3-PLC[15].....	28
Figura 20	Processamento de transmissão G3-PLC[14]	29
Figura 21	Estrutura típica da trama G3-PLC[14]	31
Figura 22	Estrutura típica da trama G3-PLC[14]	31
Figura 23	Diagrama de blocos do sistema de comunicação PLC.....	34
Figura 24	Modulo de avaliação BOSTXL-AFE031-DF1.....	36
Figura 25	Diagrama do sistema PLC[17]	36
Figura 26	Exemplo de vários módulos acoplados à linha DC[9]	37
Figura 27	Esquema elétrico do protótipo.....	38
Figura 28	Circuito abaixador de tensão	39
Figura 29	Diagrama de blocos funcional do AFE031[18].....	39
Figura 30	Ligações ao amplificador de tensão	40

Figura 31	Ligações em modo PWM[18]	40
Figura 32	Condensadores externos do Rx Filter	41
Figura 33	Filtro passa banda à entrada do RX PGA1	41
Figura 34	Circuito de ligações aos condensadores externos	42
Figura 35	Pontos de conexão para a comunicação com o AFE031	42
Figura 36	Parte do circuito de saída de acoplamento à linha	43
Figura 37	Placa em PCB do protótipo	43
Figura 38	Face superior a desenhar na placa PCB	44
Figura 39	Face inferior a desenhar na placa PCB	44
Figura 40	Protótipo montado	45
Figura 41	Diagrama de blocos da interface entre o controlador e o AFE031	47
Figura 42	Fluxograma da função principal	48
Figura 43	Fluxograma da configuração do AFE031	49
Figura 44	Fluxograma da configuração do microcontrolador	50
Figura 45	Configuração dos pinos no Atmel 328P – Pinos destinados à comunicação SPI	51
Figura 46	Configuração dos pinos no AFE031 – Pinos destinados à comunicação SPI	51
Figura 47	Configuração dos pinos do Atmel 328P – Pinos destinados ao sinal PWM	55
Figura 48	Fluxograma das configurações do Timer 0	60
Figura 49	Esquema de ligações para teste do PWM com analisador logico	65
Figura 50	Circuito com analisador logico	66
Figura 51	Período de 1 bit	67
Figura 52	Mensagem enviada por PWM-Leitura com “marcadores”	68
Figura 53	Mensagem enviada por PWM – Leitura do sinal PWM	68
Figura 54	Esquema de ligações do teste do AFE031	70
Figura 55	Montagem do circuito de teste do AFE031	70
Figura 56	Leitura do sinal à saída do AFE031	71
Figura 57	Leitura do sinal à saída do AFE031	72

Índice de Tabelas

Tabela 1	Calendarização do projeto.....	3
Tabela 2	Características dos protocolos de redes industriais[4].....	8
Tabela 3	Especificações do IEEE P.1901 e IEEE P.1901.2[4]	10
Tabela 4	Aplicações que incorporam OFDM[6].....	15
Tabela 5	Parâmetros do cabeçalho[11]	26
Tabela 6	Parâmetros PRIME PHY[11]	27
Tabela 7	Taxas de transmissão e tamanho de pacotes para modulação e codificação[11]	27
Tabela 8	Parâmetros G3-PLC[1].....	32
Tabela 9	Taxas de transmissão para diferentes modulações[1]	32
Tabela 10	Parâmetros G3-PLC e PRIME[16].....	33
Tabela 11	Funcionalidade CPOL	52
Tabela 12	Funcionalidade CPHA.....	52
Tabela 13	Relação entre SCK e a frequência de oscilador.....	53
Tabela 14	Modo de comparação de saída Timer1.....	56
Tabela 15	Descrição de bit do modo de geração de onda do Timer1	57
Tabela 16	Descrição de bit para escolha de <i>Clock</i> Timer1	58
Tabela 17	Modo de comparação de saída Timer0.....	61
Tabela 18	Descrição de bit do modo de geração de onda do Timer0	61
Tabela 19	Descrição de bit para escolha de <i>Clock</i> Timer0	62

Acrónimos

AC	–	Alternate Current
ACK	–	Acknowledge
ADC	–	Analog-to-Digital Conversion
AFE	–	Analog Front-End
ARIB	–	Association of Radio Industries and Businesses
BER	–	Bit Error Ratio
CDMA	–	Code Division Multiple Acces
CENELEC	–	Comité Européen de Normalisation Électrotechnique
CRC	–	Cyclic Redundancy Check
CS	–	Convergence Sublayer
DC	–	Direct Current
DBPSK	–	Differential Phase Shift Keying
DQPSK	–	Differential Quadrature Phase Shift Keying
D8PSK	–	Differential 8-Phase Shift Keying
EMF	–	Eletromotive Force
FCC	–	Federal Communications Comission
FEC	–	Forward Error Correction
FCH	–	Frame Control Header

FDM	– Frequency Division Multiplexing
FDMA	– Frequency Division Multiple Acces
FSK	– Frequency Shift Keying
GPIO	– General Purpose Input/Output
IED	– Integrated Development Enviromnent
IPv6	– Internet Protocol versão 6
MAC	– Medium Access Control
MISO	– Master-in Slave-out
MOSI	– Master-out Slave-in
MPDU	– MAC Protocol Data Units
MSDU	– MAC Service Data Units
NACK	– Not Acknowledge
OSI	– Open Systems Interconnection
PA	– Power Amplifier
PHY	– Physical Layer
PLC	– Power Line Communication
PPDU	– Physical Protocol Data Unit
PRIME	– Power Line Intelligent Metering Evolution
PWM	– Pulse Width Modulation
QAM	– Quadriture Amplitude Modulation
RC	– Repitition Coder

- RS – Reed-Solomon
- SCKL – Serial Clock
- S-FSK – Spread Frequency Shift Keying
- SNR – Signal Noise Ratio
- SPI – Serial Peripheral Interface
- SS – Slave Select
- TDMA – Time Division Multiple Acces
- TP – Test Point

1. INTRODUÇÃO

A comunicação de informação é uma das áreas tecnológicas mais importantes para a sociedade atual e, pela competitividade e valor que acrescenta aos serviços e indústrias existentes, a melhoria é uma necessidade constante.

O aumento do número de transmissões de dados e das velocidades de transmissão provocaram um crescimento exponencial da área nos últimos anos.

Em diversos setores, a inclusão dos sistemas de comunicação torna-se indispensável, mas também dispendiosa. Por essa razão, um dos focos dos investigadores na área das telecomunicações tem sido a redução dos custos associados à implementação e a melhoria da qualidade do serviço.

Neste âmbito, a comunicação através da linha de alimentação – *Power Line Communication* (PLC) – destaca-se como uma das tecnologias mais promissoras. Esta comunicação consiste no aproveitamento da cablagem existente da alimentação de energia elétrica para a comunicação entre dispositivos, reduzindo assim o investimento necessário.

Um exemplo de aplicação do PLC é a monitorização e controlo. Na indústria existe uma grande exigência de eficiência e reaproveitamento de espaço, tornando o PLC uma tecnologia interessante de explorar, pois não é necessário instalar um sistema sem fios ou

cablagem para a comunicação. Uma destas aplicações industriais é a gestão e monitorização de baterias de veículos industriais.

No entanto os maiores desafios que esta tecnologia enfrenta são garantir qualidade de comunicação e assegurar que a qualidade do fornecimento de eletricidade não é afetada no processo. Estes desafios agigantam-se quando a comunicação é feita em ambientes hostis com uma grande contribuição para o ruído do sinal.

1.1. CONTEXTUALIZAÇÃO

Esta dissertação surge da necessidade de um sistema de identificação e monitorização de baterias utilizadas em veículos industriais. Através do controlo e monitorização, é necessária uma redução no consumo energético e uma melhoria logística na utilização das baterias. Em último caso, é pretendido total disponibilidade das baterias devidamente carregadas para o funcionamento normal dos veículos. Desta forma, é possível prevenir perturbações na indústria.

A seleção desta tecnologia foi motivada pela sua fácil integração e pelo facto de se apresentar como uma opção inovadora no que diz respeito à transmissão de dados através de uma linha de alimentação em corrente contínua (DC – *direct current*).

1.2. OBJETIVOS

Neste projeto foi estudado, dimensionado e desenvolvido um sistema que identifica e recolhe dados de baterias utilizando a tecnologia de comunicação PLC numa linha de alimentação DC. A sua finalidade é a identificação de diferentes baterias durante o processo de carregamento, através do respetivo número identificador, possibilitando igualmente a recolha de dados relativos aos tempos de cargas e descargas e também ao seu tempo de utilização. Para o envio dos dados, é necessário integrar equipamento capaz de enviar um sinal pela linha de alimentação.

A linha de alimentação das baterias, é uma linha de transmissão em corrente contínua. No entanto para transmitir dados o sinal não poderá ser contínuo, mas sim um sinal modulado, injetado na linha como um sinal alternado de alta frequência e baixa amplitude.

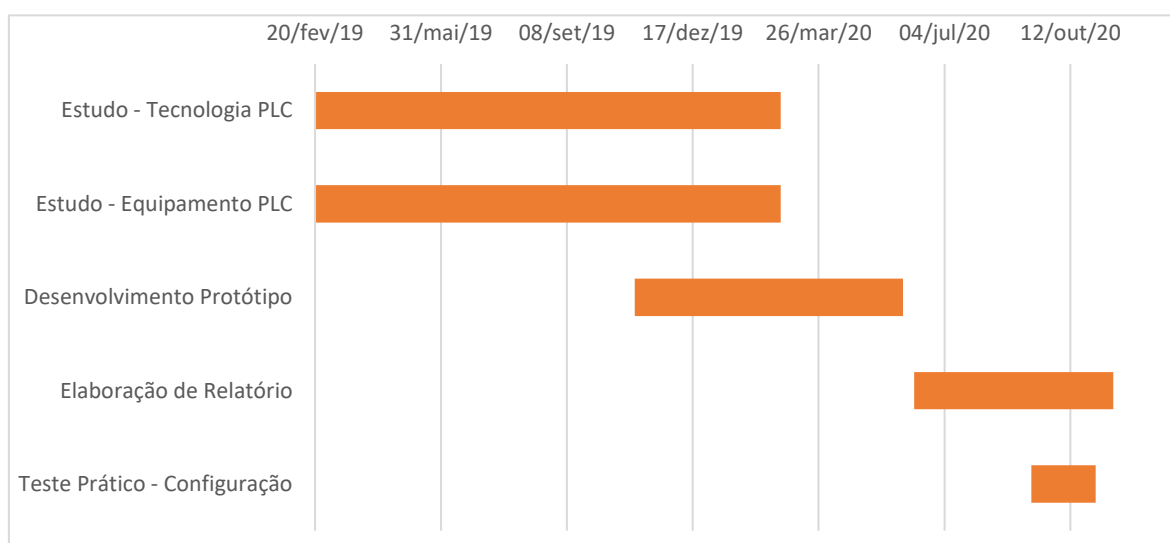
Na fase de implementação, o protótipo desenvolvido tem de ser capaz de transmitir dados de identificação de baterias, ou seja, fornecer as informações necessárias para a identificação da bateria que se encontra carregamento.

Faz também parte dos objetivos desta dissertação, a comparação com outros tipos de tecnologias de telecomunicações, analisando as suas características e a sua performance.

1.3. CALENDARIZAÇÃO

Sendo o desenvolvimento de um sistema que permite a identificação de baterias através da comunicação pela linha de alimentação DC a motivação deste trabalho, a sua prossecução conduziu à calendarização apresentada na tabela 1. Esta inclui um conjunto de tarefas, como o estudo da tecnologia PLC e os equipamentos que a implementam, o desenvolvimento de um protótipo, a elaboração do relatório, testes e resultados.

Tabela 1 Calendarização do projeto



1.4. ORGANIZAÇÃO DO RELATÓRIO

No Capítulo 1 é feita a apresentação da dissertação. No capítulo seguinte apresenta-se um estudo sobre as características da tecnologia PLC e os respetivos protocolos de comunicação. No capítulo 3, descrito o desenvolvimento de um protótipo de teste para a comunicação pela linha de alimentação DC. O capítulo 4 é detalhada toda a lógica do código desenvolvido para o envio de um sinal. No capítulo seguinte, 5, são apresentados os resultados e testes

efetuados. No último capítulo, o 6, estão reunidas as principais conclusões e as perspectivas dos futuros desenvolvimentos.

2. COMUNICAÇÃO PELA LINHA DE ALIMENTAÇÃO DC

A comunicação por PLC tem vindo a ter um enorme crescimento na sua implementação e no seu desenvolvimento nos últimos anos. Tal deve-se à necessidade de simplificar as infraestruturas, onde o constante aumento da presença de tecnologia pede igualmente um aumento da mesma que, por conseguinte, se torna mais dispendiosa.

Esta comunicação pode ser feita utilizando os dois tipos de corrente elétrica conhecidos, sendo eles a corrente contínua (DC) e a corrente alternada (AC), ambas com diferentes aplicações.

No caso de baterias de veículos industriais existe uma necessidade de identificação e monitorização tal como controlo de ciclos de carregamento. Dentro deste problema, com uma infraestrutura já existente e uma indústria com reduzida possibilidade de paragem surgiu a possibilidade de com base na tecnologia PLC, desenvolver um sistema de comunicação pela linha de carregamento das baterias.

No sistema a ser desenvolvido, será feito um estudo sobre a linha de transmissão DC, pois a comunicação dos dados vai ocorrer durante o carregamento das baterias. Estas baterias são

utilizadas em ambiente industrial, como fonte de alimentação de veículos em fábricas, onde as condições de transmissão podem não ser as mais favoráveis.

Vai ser também elaborado um estudo sobre o PLC, as suas características, e as vantagens e desvantagens associadas à sua utilização.

Após a análise do PLC e das suas características, é necessário perceber que protocolos estão disponíveis, comparando os existentes de acordo com as suas vantagens e desvantagens, tal como os tipos de modulação utilizados nesta tecnologia.

2.1. CARACTERÍSTICAS DO CANAL EM PLC

A rede de energia elétrica tem grandes diferenças de topologia, estrutura e até de propriedades em relação a outro tipo de redes, nomeadamente às redes de comunicação. Tal como referido anteriormente, o funcionamento eficaz do PLC está sujeito às condições do canal, onde existe diversos obstáculos que dificultam a comunicação[1].

Como a rede elétrica não tem como objetivo a transmissão de dados, quando a utilizamos para este fim encontramos algumas dificuldades como impedâncias variáveis, elevada atenuação e elevados níveis de ruído[1].

O ruído é, possivelmente, a principal causa de atenuação a ter em consideração na transmissão de dados em PLC, como ilustrado no diagrama da Figura 1. Na linha elétrica, o ruído pode ser impulsivo ou seletivo na frequência, ou ambos[1].

É possível apontar a existência de quatro tipos de ruído, o ruído colorido, o ruído de fundo, ruído impulsivo síncrono e assíncrono[1].

O ruído colorido é o resultado da soma espectral de todas as fontes de ruído de baixa frequência. Tem uma baixa densidade e diminui com o aumento da frequência. A denominação de ruído colorido vem da analogia às cores, por exemplo, o ruído branco é uma denominação que representa a mistura de todas as cores possíveis, daí ser uma denominação com o mínimo de informação possível. Esta analogia é estendida às outras cores, sendo mais específico no seu significado, pois certas frequências têm mais predominância com certas cores [2].

O ruído impulsivo síncrono é causado pelas fontes de energia a operarem sincronamente com a frequência principal. Já o assíncrono é causado por cargas ou fontes que se ligam e

desligam da rede, sendo este o que mais prejudica as comunicações PLC, não sendo aplicável em DC[1].



Figura 1 Diagrama de integração do ruído no sistema

Neste sistema, é estudada a possibilidade da linha de transmissão estar presente num ambiente industrial onde devido à presença elétrica nos equipamentos industriais, existe a preocupação dos campos magnéticos. Um campo magnético tem a capacidade de induzir uma corrente num objeto condutor para linhas de transmissão, mas para terem efeito perceptível isso exigiria que um objeto condutor comprido ficasse muito próximo da linha para ter um efeito perceptível[3].

2.2. PLC NA INDÚSTRIA

A tecnologia PLC acrescenta uma alternativa no mercado de transmissão de dados, nomeadamente em aplicações industriais, como redes de controlo e monitorização de maquinaria ou recolha de leituras de sensores[4].

O uso de PLC tem sido principalmente orientado para as redes inteligentes, no entanto é uma opção que não tem sido explorada na indústria, onde as comunicações sem fios e cabos de cobre e fibra são primariamente utilizadas. Em termos industriais o PLC pode ser uma opção, especialmente onde a construção de infraestrutura não é viável. O aparecimento do PLC como alternativa surge devido a um tempo reduzido de instalação, e retirando inclusive a necessidade de parar o funcionamento normal da indústria para preparação da instalação[4].

A nível industrial normalmente não existe a necessidade de taxas de transmissão altas, no entanto é necessária uma alta fidelidade, pois falhas nas comunicações podem causar grandes constrangimentos[4].

2.2.1. CARACTERÍSTICAS DA TRANSMISSÃO DE DADOS NA INDÚSTRIA

Os sistemas de comunicações no terreno devem ser capazes de transmitir dados periódicos em tempo real para monitorização e alarmes esporádicos. Para estes casos, é recomendada a comunicação por fio pois oferece uma menor possibilidade de erros e perda de pacotes[4].

A tabela abaixo apresenta, alguns dos protocolos de redes sem fio ou com fio utilizados na indústria, comparando o alcance, taxa de transmissão e número de nós[4].

Tabela 2 Características dos protocolos de redes industriais[4]

Type	Protocol Name	Data rate (Kbps)	Range (meters)	Nodes
Wired	<i>Fieldbus Foundation</i> ®	31,25	1900	8-16
	<i>Modbus</i> ®	38,4	1200	32
	<i>Profibus</i> ®	93,75	1200	32
		182,5	600	
		500	200	
	<i>DeviceNet</i> ®	125	500	64
		250	250	
500		100		
Wireless	<i>R-Fieldbus</i> [5]	2000	100	30
	<i>ELPRO</i> ® [8]	19,2	5000	95
	<i>ZigBee</i> ® [9]	250	100	64000

Para o caso das tecnologias sem fios o ambiente envolvente afeta drasticamente o alcance, uma vez que o sinal tem dificuldade em penetrar estruturas e é também influenciado pelas condições atmosféricas[4].

Para uma comunicação a tempo real e a níveis industriais devem ser cumpridos os seguintes pontos[4].

- **Área de cobertura**
- **Tempo de resposta** - Considera-se aceitável um tempo de resposta de 1-2 segundos aceitável.
- **Robustez** – No caso de perda de nó por um certo período de tempo o sistema deve conseguir compensar.
- **Segurança dos dados**

2.2.2. ATENUAÇÃO E INTERFERÊNCIAS

A comunicação PLC em ambientes industriais encontra diversas fontes de interferências e atenuação, desde motores elétricos a controladores e conversores. Dado que o motor elétrico é um dos equipamentos em maior número na indústria, é preciso ter em conta o papel que o mesmo pode ter no canal de PLC. O controle dos motores através de inversores também é um problema, devido ao ruído gerado na comutação[4].

Os motores elétricos funcionam como impedância. Esta impedância depende da frequência, que na banda de frequências do intervalo dos 10KHz aos 30MHz podem variar entre 1Ω a $10K\Omega$, bem como de onde o sinal PLC será transmitido, ou seja, em cabos de baixa tensão na mesma banda de frequências a impedância pode chegar dos 5Ω aos 50Ω [4].

Por estes motivos, conclui-se que existe incompatibilidade no alcance de frequências. Devido a esta incompatibilidade na interface entre o cabo e o motor elétrico a impedância de entrada, medida na entrada do cabo de alimentação, torna-se variável na frequência. Isto significa que um cabo que termine com um motor elétrico pode ser considerado como uma impedância de carga (componente que consome corrente de um circuito elétrico) conectada paralelamente à linha de alimentação[4].

A utilização de inversores na indústria tornou-se bastante popular, pois com estes dispositivos é possível controlar a velocidade de rotação de um motor e a corrente de arranque de motores de indução de corrente alternada. Os inversores funcionam como conversores, ou seja, convertem corrente alternada em corrente contínua, e voltam a converter para corrente alternada com a frequência desejada. No entanto, este procedimento gera ruído e variações de impedância que podem se tornar um obstáculo ao sinal PLC[4].

2.2.3. PROTOCOLOS PLC PARA INDÚSTRIA

Se compararmos os dois protocolos PLC indicados na tabela 3, é possível perceber que G3-PLC tem uma taxa de transmissão suficiente para funcionar na comunicação no terreno em ambiente industrial, com o valor de 500 Mbs e com a vantagem de ser capaz de obter um maior alcance. Outra das vantagens do G3-PLC é capacidade de atravessar transformadores de tensão sem usar circuitos de acoplamento[4].

Tabela 3 Especificações do IEEE P.1901 e IEEE P.1901.2[4]

Specification	P.1901.1	P.1901.2 (G3-PLC)
<i>Frequency Range</i>	2-28MHz	3-500KHz
<i>Data Rate (maximum)</i>	500Mbps	500Kbps
<i>Security</i>	AES-128-bit Encryption	AES-128-bit Encryption
<i>Range (Maximum)</i>	1500 meters	Over 6 kilometers
<i>Modulation</i>	OFDM	OFDM
<i>ROBO Mode</i>	Yes	Yes
<i>Subcarrier Modulation Type</i>	BPSK, QPSK and QAM (16/64/256/1024)	D8PSK, DQPSK, DBPSK and ROBO

2.3. COMUNICAÇÃO

2.3.1. LARGURAS DE BANDA

As tecnologias PLC são divididas em duas classes, que operam em valores de frequência diferentes, onde a seleção de uma em detrimento da outra depende do alcance de comunicação desejado. O alcance é inversamente proporcional à frequência e do fluxo de dados. Estas classes são caracterizadas pela sua largura de banda, sendo elas a faixa estreita e faixa larga[1].

FAIXA LARGA

A classe de faixa larga da comunicação PLC tem uma banda alargada, no intervalo de 1,6MHz a 250 MHz, sendo este intervalo variável[1].

Esta faixa de frequências permite ao PLC uma alta transmissão de dados, à volta dos 100Mbps. Como desvantagem, nesta banda só é possível transmitir dados a uma curta distância, pois utiliza frequências altas, reduzindo assim o alcance do sinal[1]. As suas principais aplicações são em espaços interiores, sendo a sua principal aplicação a distribuição de internet, nomeadamente em edifícios, onde recai mais especificamente em dados de multimédia dentro dos mesmos edifícios[1].

FAIXA ESTREITA

A faixa estreita é a banda com mais aplicação dentro do PLC, operando no intervalo de 0,1 kHz a 500 kHz, sendo uma banda de frequências mais baixa que a faixa larga[1].

Tem como característica uma baixa taxa de transmissão de dados, que podem chegar aos 500 kbit/s[1].

Como grande vantagem conseguir transmitir a longas distâncias, em alguns quilômetros (150km para as frequências mais baixas da banda), podendo este alcance ser atingido com a introdução de repetidores de sinal no sistema[1].

As tecnologias de faixa larga incluem as bandas[1]:

- *Comité Européen de Normalisation Électrotechnique* (CENELEC): 3 kHz a 148.5 kHz, da Europa.
- *Federal Communications Commission* (FCC): 10 kHz a 490 kHz, dos Estados Unidos da América.
- *Association of Radio Industries and Businesses* (ARIB): 10 kHz a 450 kHz, do Japão.
- Banda Chinesa: 3 kHz a 500 kHz.

Adicionalmente, existem dois tipos de tecnologias onde se enquadram as mono-portadora onde as taxas de transmissão são menores, e as multi-portadoras que permitem taxas de transmissão até os 500 kbit/s e onde se destacam o *PowerLine Intelligent Metering Evolution* PRIME e G3-PLC[1].

2.3.2. SERIAL PERIPHERAL INTERFACE

O *Serial Peripheral Interface* (SPI) é um protocolo de conexão síncrono. Esta comunicação é *full duplex* (comunicação em paralelo entre dispositivos), utilizando o formato *master-slave*, ou seja, o *master* e o *slave* trocam dados em simultâneo. Como se pode verificar na Figura 2, são utilizados 4 pinos:

- SCLK -*Serial clock*,
- MISO - *Master-in, Slave-out*, pino de comunicação no sentido *slave-master*.

- MOSI - *Master-out, Slave-in*, pino de comunicação de dados, no sentido *master-slave*.
- SS - Pino de seleção do *slave*.

A comunicação por SPI segue a direção de fluxo de dados apresentada na Figura.

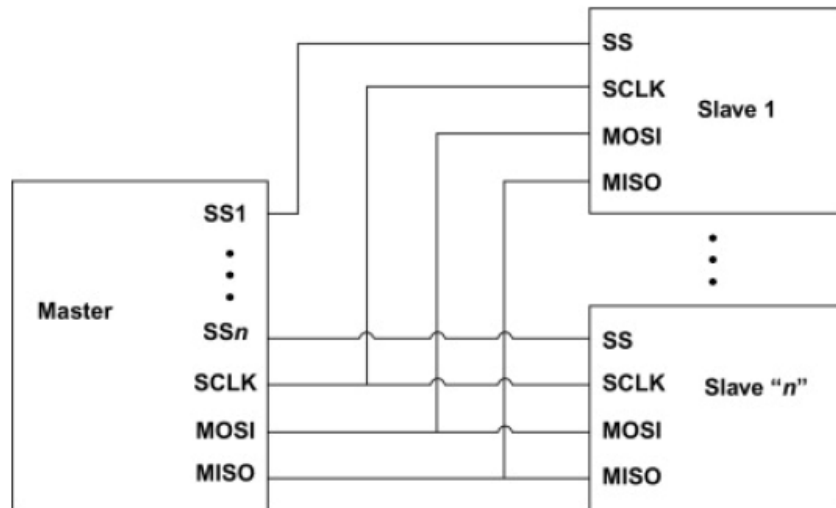


Figura 2 Comunicação SPI[5]

Os dispositivos que utilizam este protocolo têm registros de tamanho dependente da arquitetura, destinados ao armazenamento de dados a enviar e de dados recebidos.

A linha SS é utilizada pelo *master* para selecionar o *slave* com quem deseja comunicar. Cada *slave* requer uma linha individual, ou seja, para cada número de *slaves* que estão conectados existe esse mesmo número de pinos CS no *master*. As restantes 3 linhas, como o MOSI, MISO e SCLK são compartilhadas pelos *slaves*[5].

Relativamente ao funcionamento do SPI, cada operação inicia com a seleção do *slave* por parte do *master* utilizando a linha do pino SS. O *master* inicia a operação com o SCLK a uma frequência igual ou inferior à frequência máxima suportada pelo *slave*. Em cada ciclo do *clock*, o *master* envia o bit 1 para o *slave*. Para o caso de ambos não terem dados para enviar, enviam o bit 0. Caso o *master* queira enviar informação coloca os dados no MOSI, começando com o bit mais significativo. O *slave* pode eventualmente responder colocando

dados no MISO. É possível ver na Figura 3 as formas de onda para o caso de uma transmissão de 8-bits. Os bits mudam do flanco descendente do *clock*[5].

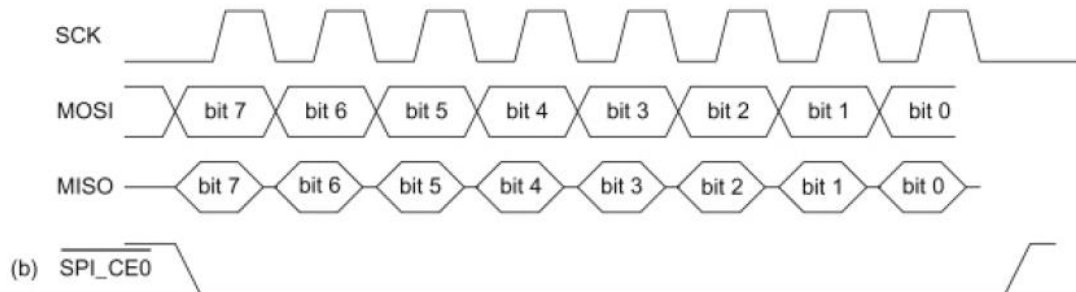


Figura 3 Formas de onda de uma ligação SPI[5]

Não existe um conjunto de frequências típicas definidas para o protocolo SPI, no entanto está entre valores a rondar os MHz. Como não existe um *overhead* adicionado como endereçamento e controlo de fluxo, a taxa de transmissão é proporcional à frequência de *clock*. Para o caso de um SPI de 50 MHz, a taxa de transmissão é 50 Mbps. Uma das grandes vantagens da utilização do SPI é o uso de altas frequências e, conseqüentemente, altas taxas de transmissão. Como a comunicação é *full-duplex*, a taxa de transmissão real poderia ser até 100 Mbps, para o caso do *master* e do *slave* enviarem dados relevantes em simultâneo[5].

2.4. MODULAÇÃO OFDM

OFDM é uma técnica de modulação que consiste na transmissão paralela de dados em diversas subportadoras, espaçadas por igual, dividindo o espectro de frequências, sendo moduladas, cada uma delas, por taxas de transmissão tão baixas quanto maior o número de subportadoras. Pode-se ver na Figura 4 o espectro de frequências OFDM[6].

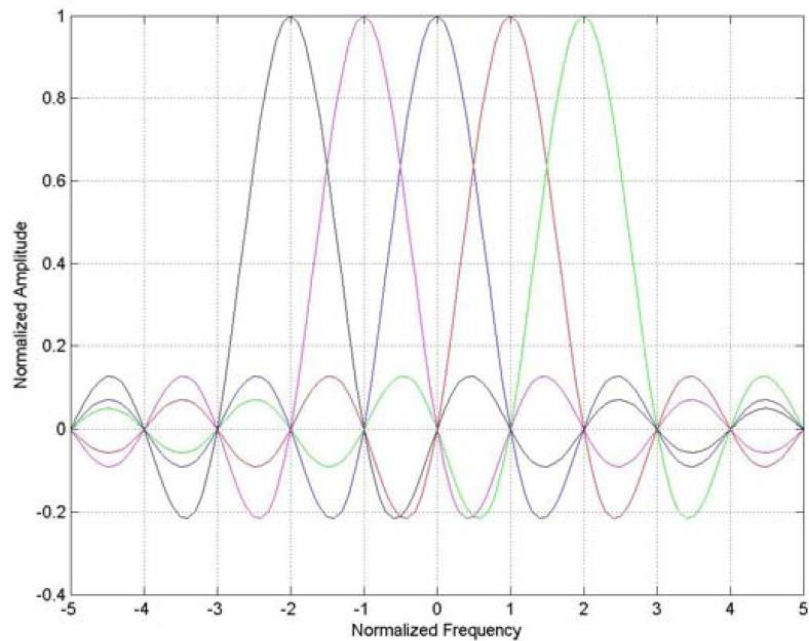


Figura 4 Espectro de frequências OFDM[6]

Quando as subportadoras têm um espaçamento apropriado (definido pela banda de dados) para satisfazer a ortogonalidade, os seus espectros vão se sobrepor, resultando numa vantagem espectral. Em comparação ao sistema clássico de dados paralelos (FDM - *Frequency Division Multiplexing*), em que é necessária uma banda de guarda entre as portadoras o que resulta no desperdício de espectro. A OFDM usa sobreposição de subportadoras e daí resulta um uso mais efetivo da largura de banda, como se pode ver na Figura 5.

A ortogonalidade entre subportadoras no domínio do tempo, implica que duas diferem exatamente por um número inteiro de ciclos durante um intervalo de tempo que corresponde à duração de um símbolo OFDM[6].

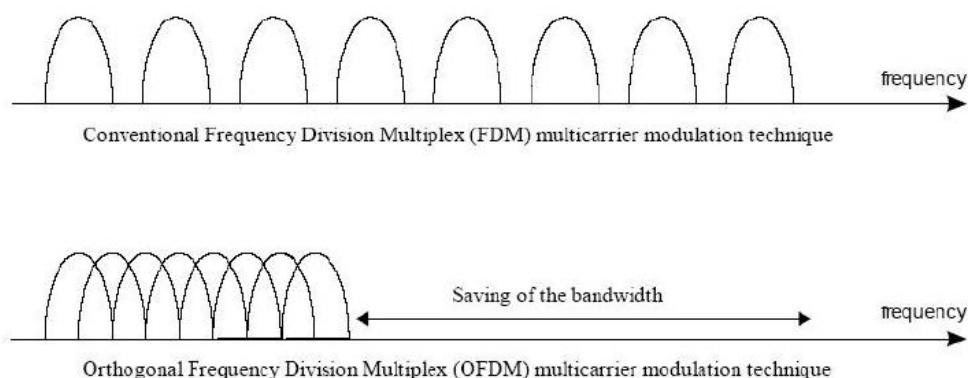


Figura 5 Largura de banda utilizada por OFDM e FDM[6]

Na tabela 4 vemos algumas aplicações que incorporam OFDM.

Tabela 4 Aplicações que incorporam OFDM[6]

	HIPERLAN/2	DAB	802.11a/g	DVB-T
No of carriers	48 subcarriers	1,705 sub-carriers in 2K DFT	48 subcarriers, 64 FFT	1,705 sub-carriers in 2K DFT 6,817 sub-carriers in 8K DFT.
Modulation scheme	16-QAM/8-PSK modulation	DQPSK/OFDM	64-QAM	64-QAM (or any choice)
Capacity	25Mbps	2Mbps	54Mbps	12-24 Mbps
Bandwidth	25MHz	1,536MHz (overall)	20MHz	8MHz RF channel
Spectral Region	5.2GHz	Band 3 174-240MHz Band L 1452-1492MHz	802.11a in 5.8GHz 802.11g in 2.4GHz	VHF/UHF band
Technology	WLAN Technology	Broadcasting technology	Wireless LAN Technology	Broadcasting technology

OFDM é um esquema de modulação e pode ser combinado com outros esquemas como *Time Division Multiple Access* (TDMA) onde a largura de banda é dividida no domínio do tempo, *Frequency Division Multiple Access* (FDMA) onde a largura de banda é dividida em diferentes bandas de frequência e *Code Division Multiple Access* (CDMA) onde a transmissão de dados é feita em simultâneo, com uma sequência de código única[7].

2.4.1. PRINCÍPIO DE MODULAÇÃO

O OFDM tem como princípio a divisão de um sinal com uma taxa de transmissão elevada num maior número de taxas de transmissão mais pequenas que são transmitidas simultaneamente sobre um número de subportadoras. Isto acontece porque a duração do símbolo aumenta para as subportadoras paralelas de baixa taxa de transmissão e a dispersão no tempo causada pelo atraso do multipercurso é reduzido[8].

A ortogonalidade entre subportadoras: no domínio do tempo, implica que duas subportadoras quaisquer diferem exatamente por um número inteiro de ciclos durante um intervalo de tempo T_0 que corresponde à duração de um símbolo OFDM, como mostra o exemplo da Figura 6[8].

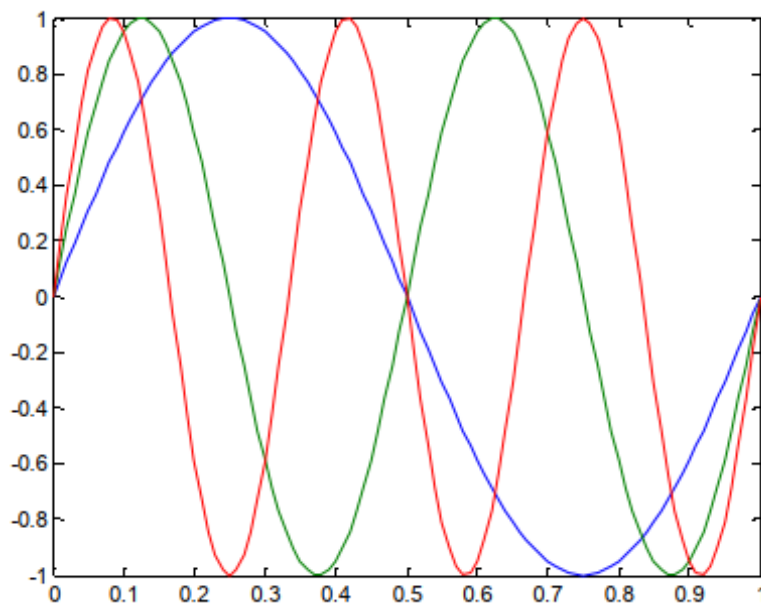


Figura 6 Exemplo de ortogonalidade no domínio do tempo

Os sinais das subportadoras podem ser descritos pela seguinte expressão:

$$s_k(t) = \begin{cases} \sin\left(\frac{2\pi}{T_0} t\right) & 0 \leq t \leq T_0 \quad k = 1, 2, \dots, N \\ 0 & \text{outros } t \end{cases} \quad (1)$$

A correlação cruzada entre cada par de subportadoras é representada por:

$$\int_0^{T_0} \sin\left(m \frac{2\pi}{T_0} t\right) \sin\left(n \frac{2\pi}{T_0} t\right) dt = \begin{cases} C & m = n \\ 0 & m \neq n \end{cases} \quad (2)$$

Um sinal OFDM consiste num somatório de subportadoras que são moduladas por PSK ou Quadrature Amplitude Modulation (QAM). Se d_i representa o símbolo complexo QAM, N_s o número de subportadoras, T a duração do símbolo e f_c a frequência da portadora então um símbolo OFDM com $t = t_s$ na notação complexa de banda base pode ser escrita como[8]:

$$s(t) = \sum_{i=-\frac{N_s}{2}}^{\frac{N_s}{2}-1} d_i + \frac{N_s}{2} * e^{(j2\pi\frac{i}{T}(t-ts))}, \quad ts \leq t \leq ts + T \quad (3)$$

$$s(t) = 0, \quad t < ts \wedge t > ts + T \quad (4)$$

2.4.2. SUB-PORTADORA PILOTO

Para obter melhor condições do canal é normal utilizar algumas subportadoras para funcionarem como piloto (*pilot carriers*). Estas subportadoras (*sub-carriers*) têm uma distribuição uniforme ao longo da faixa, e com isso servem para calcular ganhos e defasamentos para equalização de cada sub-canal. Na Figura 7 é possível ver a distribuição das portadoras pilotes (*pilot carriers*)[8].

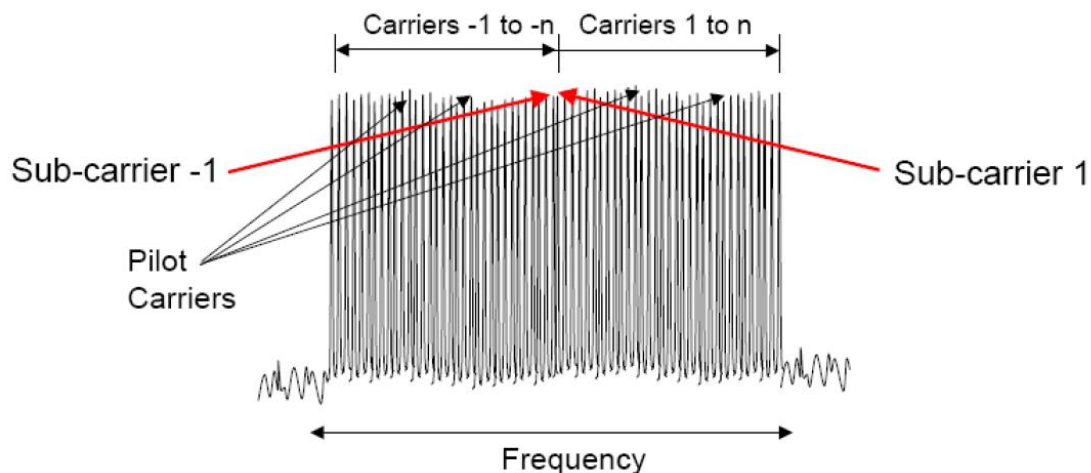


Figura 7 Subportadoras piloto sobre o campo das frequências (*frequency*)[8]

2.4.3. CODIFICAÇÃO

O controlo de erros lida com erros que acontecem devido ao canal de comunicação. Genericamente existe dois tipos de erros, que são os erros aleatórios e erros explosivos. Existe esquemas de codificação com o intuito de reduzir a probabilidade de erro[6].

No multipercurso da modulação, algumas subportadoras podem ficar completamente perdidas devido aos desvanecimento. Embora maior parte das subportadoras possa ser detetada sem erros, o *bit error ratio* (BER) será largamente dominado por algumas subportadoras com menor amplitude, daí a probabilidade de erros de bit é próxima de 0,5. Para evitar o domínio das subportadores mais fracas, a codificação de correção de erros é essencial[6].

Na Figura 8 é possível ver o diagrama de blocos com todos os passos da modulação OFDM na transmissão[6].

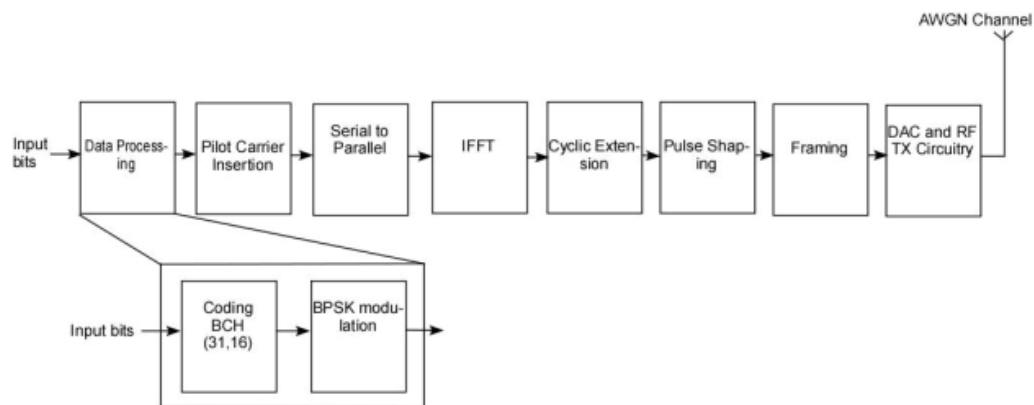


Figura 8 Diagrama de blocos de transmissão OFDM[6]

Devido ao desvanecimento seletivo de frequência, grandes desvanecimentos no espectro de frequência pode causar erros de bit em explosão em vez de aleatórios, onde maior parte da codificação de correção de erros não tem a capacidade de lidar com erros explosivos[6].

O *interleaving* é aplicado para tornar aleatório a ocorrência de erros de bit antes da decodificação. Do lado da transmissão o *interleaving* é feito de uma certa forma e do lado da recepção será feito da forma oposta previa à decodificação. A técnica mais utilizada de de *interleaving* é *Block interleaver*[6].

2.4.4. ESQUEMAS DE MODULAÇÃO

Na modulação OFDM podem ser implementados esquemas como PSK e QAM. Esta escolha depende de diversos fatores como a taxa de transmissão e a sensibilidade para erros[6].

Na modulação as rotações de fase e a escalada de amplitude aumentam consideravelmente a taxa de erros e até pode destruir as comunicações por completo. No entanto estes problemas

podem ser ultrapassados utilizando uma equalização de sinal antes da desmodulação, ou seja, a rotação de fase e a escalada de amplitude é medida utilizando símbolos piloto[6].

Para fazer o rastreamento no canal é necessário atualizações contínuas na equalização do canal, daí ser inserido símbolos piloto na transmissão. Quanto maior o número de sinais piloto mais rápido e eficaz será o rastreamento[6].

2.4.5. TEMPO DE GUARDA E PREFIXO CÍCLICO

O OFDM é uma técnica eficiente a lidar com o multipercurso, devido à utilização de várias subportadoras. O multipercurso causa atrasos no sinal a ser recebido, podendo criar conflitos no caso da utilização de várias portadoras como mostra na Figura 9. Para evitar a interferência inter-simbólica é introduzido um tempo de guarda por cada símbolo. O tempo de guarda não tem necessidade de conter qualquer sinal, no entanto isto levaria a interferência inter-portadora. A interferência inter-portadora consiste no cruzamento de sinais entre subportadoras, ou seja, deixam de ser ortogonais. Para resolver o problema utiliza-se um prefixo cíclico. Este tempo deve ser definido de forma a que seja maior do que o atraso, para que os componentes do multipercurso de um símbolo não interfiram com o símbolo seguinte[6].

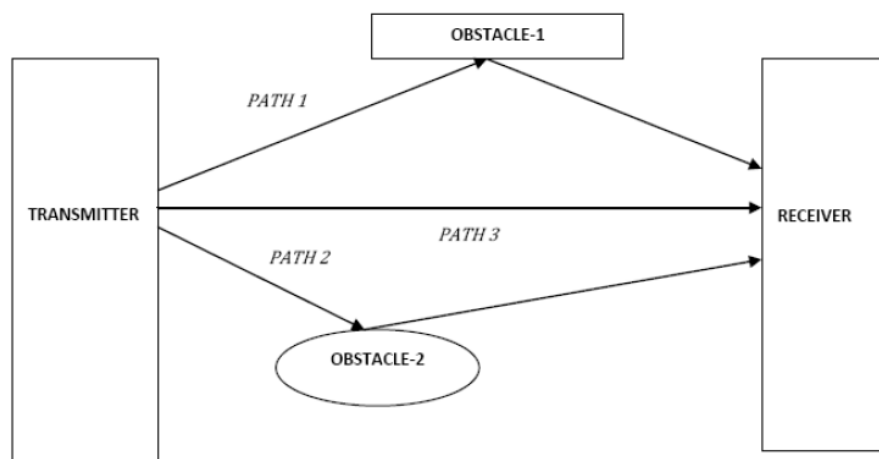


Figura 9 Diagrama explicativo da ISI, devido ao multipercurso[6].

O Prefixo cíclico no tempo de guarda garante que as réplicas têm um número inteiro de ciclos no intervalo da transformada de Fourier, desde que o atraso seja menor que o tempo de guarda como se pode ver na Figura 10[8].

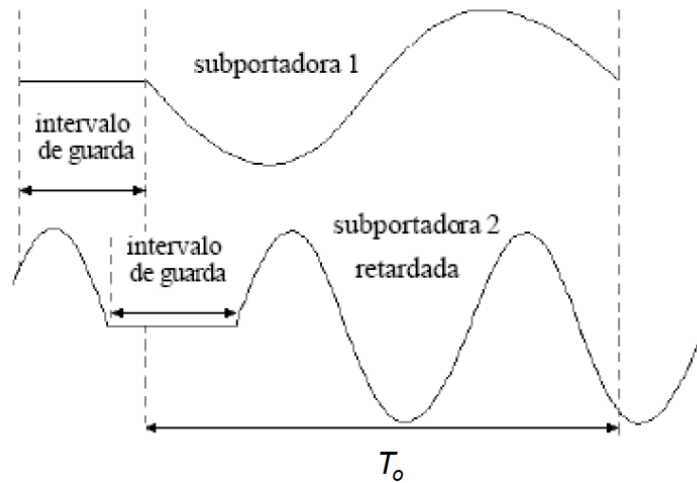


Figura 10 Tempo de guarda e prefixo cíclico[8]

2.4.6. FSK

Frequency Shift Keying (FSK) é uma técnica de modulação que utiliza mudanças discretas de frequência para transmitir e receber dados. Uma das modulações derivadas do FSK é o Binary Shift Keying[9].

Esta técnica consiste na comutação entre duas frequências discretas, a *Mark Frequency* ('1') e a *Space Frequency* ('0'), onde as mesmas correspondem diretamente com o respectivo bit que transmissão como se pode ver na Figura 11[9].

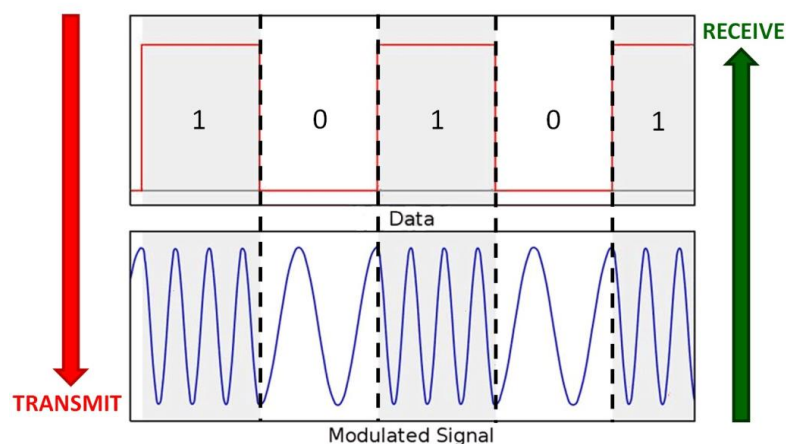


Figura 11 BFSK no domínio do tempo[9]

No Binary Shift Keying é utilizado duas frequências diferentes, por exemplo, f_1 e $f_2 = f_1 + \Delta f$, para transmitir uma sequência de informação binária. As duas formas de onda podem ser expressas da seguinte forma[10]:

$$u_1(t) = \sqrt{\frac{2\varepsilon_b}{T_b}} \cos 2\pi f_1 t, \quad 0 \leq t \leq T_b \quad (5)$$

$$u_2(t) = \sqrt{\frac{2\varepsilon_b}{T_b}} \cos 2\pi f_2 t, \quad 0 \leq t \leq T_b \quad (6)$$

Nas duas formas de onda FSK, ε_b é sinal de energia do bit e o T_b é a duração do intervalo do bit[10].

Um transmissor simplificado FSK consiste em dois blocos de duas fontes com relógio interno. Juntamente com uma entrada binária sequencial para controlar a posição do interruptor, como se pode ver na Figura 12[9].

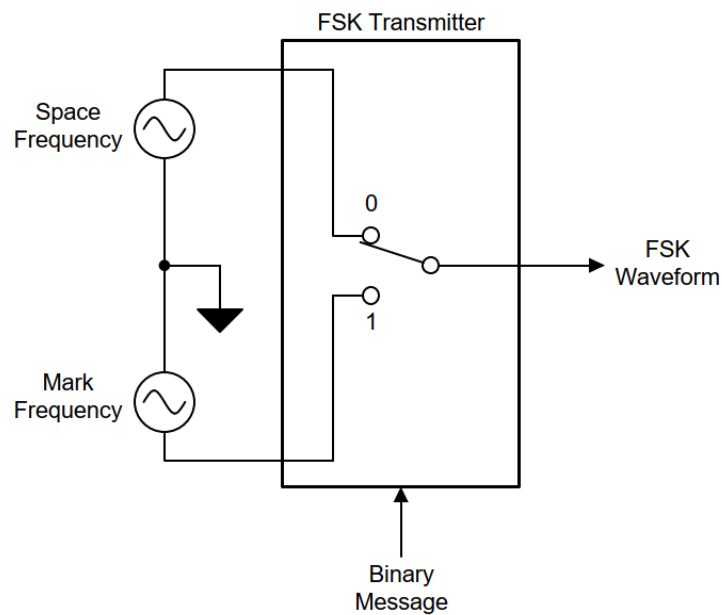


Figura 12 Diagrama de transmissor simplificado FSK[9]

As duas fontes osciladoras, produzem dois sinais, uma com frequência alta e outra baixa, estão conectadas a um interruptor. A sequência de entrada binária é aplicada para escolher as frequências de acordo com a entrada binária como se pode ver na Figura 13[9].

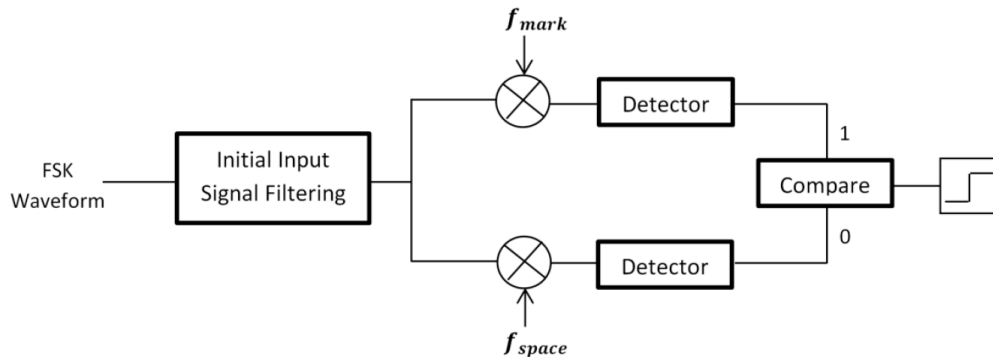


Figura 13 Diagrama de transmissão FSK[9]

2.5. PROTOCOLOS PLC

A escolha de um protocolo de comunicação adequado garante a segurança e disponibilidade da informação na sua transmissão, sendo que o custo e a eficiência do sistema são critérios importantes na seleção de uma solução.

Os dois protocolos escolhidos para análise são o *Powerline Intelligent Metering Evolution* (PRIME) e o G3-PLC.

2.5.1. PRIME

O PRIME é um dos protocolos PLC mais maduros. Sendo baseada em OFDM é um protocolo capaz de lidar com grandes interferências em condições difíceis. Estas interferências dificultam uma comunicação fiável provocadas pela transmissão partilhada dos canais[11].

Esta norma define as duas camadas mais baixas do modelo *Open Systems Interconnection* (OSI) como se pode ver na Figura 14.



Figura 14 Modelo OSI[12]

A pilha de protocolos é a proposta na Figura 15.

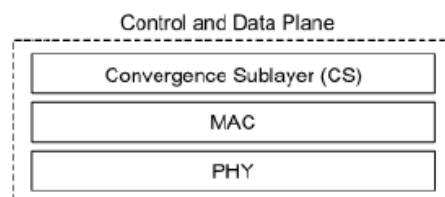


Figura 15 Pilha de protocolos norma PRIME[1]

A camada *Convergence Sublayer* (CS) classifica o tráfego, associando-o com a sua respetiva ligação *Medium Access Control* (MAC), fazendo o mapeamento de qualquer tipo de tráfego para ser incluído nas *MAC Service Data Units* (MSDU's). A camada MAC possibilita o acesso ao canal, a alocação de largura de banda, gestão da ligação e resolução de endereços. Neste caso, a camada MAC está desenvolvida de forma a operar num ambiente Cliente-Escravo orientado à conexão e está otimizada para ambientes de linha elétrica de LV. Já a camada física (PHY) assume o papel de transmitir e receber as *MAC Protocol Data Units* (MPDU's) entre os nós vizinhos[1][11].

CAMADA FÍSICA

Como já foi referido anteriormente a camada física do PRIME é baseada na modulação OFDM, utilizando frequências na banda de 3 kHz até 95 kHz. As frequências abaixo do 40 kHz mostram diversas dificuldades nomeadamente nas linhas elétricas de baixa tensão como por exemplo o ruído colorido, que está sempre presente nas linhas elétricas. No entanto o PRIME também usa a banda de 41,992 k Hz até 88,867 kHz, sendo isto possível usando a modulação OFDM com um sinal com 97 subportadoras espaçadas igualmente entre si. A modulação diferencial também é utilizada, como *Differential Phase-Shift Keying* (DBPSK), *Differential Quadrature Phase Shift Keying* (DQPSK) ou *Differential 8-Phase Shift Keying* (D8PSK)[11].

ASPETOS FUNDAMENTAIS

Do lado da transmissão a camada física é responsável por receber uma MPDU, gerando uma trama designada *Physical Protocol Data Unit* (PPDU)[1][11].

Pode ser observado o processamento da trama na Figura 16.

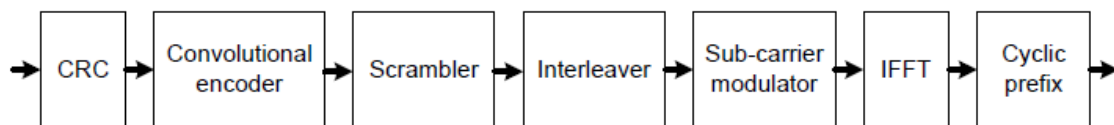


Figura 16 Processamento da PPDU[11]

1. **Cyclic Redundancy Check (CRC):** Este bloco adiciona um código de redundância ao cabeçalho para melhorar a deteção de erros. No *payload* o CRC é adicionado na camada MAC[1].
2. **Codificação Convolutacional:** O codificador surge no PRIME como um bloco opcional. No caso de o canal apresentar condições que garantem a robustez necessária, pode-se aumentar o ritmo de dados ao desativar este bloco, mas em contrapartida as comunicações ficam mais sujeitas a erros. No início de cada transmissão PPDU o estado do codificador é colocado a zero[1][11].

Como se pode ver na Figura 17, oito zeros são inseridos no fim da informação do cabeçalho para limpar o codificador e forçá-lo a regressar ao estado zero. Se o

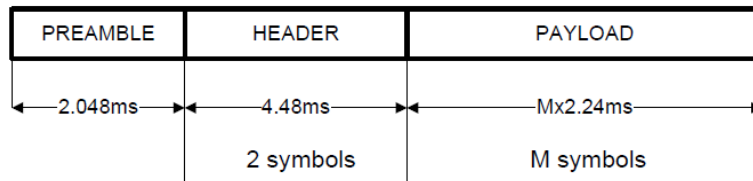


Figura 18 Formato da trama[11]

Preâmbulo (*Preamble*): É utilizado no início de cada PDU para sincronizar. É um envelope constante utilizado em vez de símbolos OFDM, conseguindo assim o máximo de energia, tendo de ser ágil nas frequências para permitir a sincronização na presença de atenuação seletiva de frequências.

- **Cabeçalho (*Header*):** O cabeçalho é composto por dois símbolos OFDM, que são enviados utilizando a modulação DBSK e com o código convolucional acionado (FEC)[11].

Tabela 5 Parâmetros do cabeçalho[11]

	DBPSK
Código Convolucional (1/2)	<i>On</i>
Nº de bits de informação por portadora (N_{BPSC})	0.5
Nº de bits de informação por símbolo OFDM (N_{BPS})	42

- ***Payload*:** O *Payload* é modulado por DBPSK, DQPSK ou D8PSK dependendo da configuração da camada MAC. É esta camada que seleciona a melhor modulação através de erros em transmissões anteriores para o mesmo recetor, ou utilizando informação do SNR[11].

PARÂMETROS

Na tabela 6 e 7 é possível ver os parâmetros OFDM da norma PRIME na sua camada física. Nos parâmetros inclui-se a lista de frequências e tempo utilizado. Estes parâmetros são comuns para todas as combinações de codificação[11].

Tabela 6 Parâmetros PRIME PHY[11]

Parâmetro	Valores	
Relógio em banda de base (Hz)	250000	
Espaçamento entre portadoras (Hz)	488.28125	
Número de portadoras para dados	84 (cabeçalho)	96 (payload)
Número de portadoras para canal piloto	13 (cabeçalho)	1 (payload)
Intervalo FFT ⁴ (amostras)	512	
Intervalo FFT (μs)	2048	
Cyclic Prefix (amostras)	48	
Cyclic Prefix (μs)	192	
Intervalo de símbolo (amostras)	560	
Intervalo de símbolo (μs)	2240	
Tempo do preâmbulo (μs)	2048	

Tabela 7 Taxas de transmissão e tamanho de pacotes para modulação e codificação[11]

	DBPSK		DQPSK		D8PSK	
	On	Off	On	Off	On	Off
Código Convolutacional (1/2)						
Nº de bits de informação por portadora (N_{BPSK})	0.5	1	1	2	1.5	3
Nº de bits de informação por símbolo OFDM (N_{BPS})	48	96	96	192	144	288
Ritmo de dados bruto (kbit/s apróx.)	21.4	42.9	42.9	85.7	64.3	128.6
Tamanho máximo (em bits) da MSDU com 63 símbolos)	3016	6048	6040	12096	9064	18144

2.5.2. G3-PLC

A norma G3-PLC™ foi desenvolvida para responder a uma necessidade do setor energético um padrão de comunicações na linha energética que possibilite uma visão á rede de rede inteligente [13].

As especificações do G3-PLC foram escolhidas para endereçar os seguintes objetivos:

- Providenciar uma comunicação robusta em linhas elétricas com condições adversas à comunicação[14].
- Obter um mínimo de 20kbps de taxa de transmissão no modo normal de funcionamento[14].
- Capacidade de coexistência nas frequências selecionadas, coabitando com comunicação *Spread Frequency Shift Keying* (S-FSK) de banda estreita[14].

- Garantir uma comunicação robusta com a capacidade de selecionar no canal as frequências que não tenham muita interferência[14].

O G3-PLC agrega as camadas MAC e PHY como se pode ver na Figura 19 com o objetivo de medir a energia elétrica. Esta norma utiliza o protocolo *Internet Protocol* versão 6 (IPv6), o que aumenta as escolhas para potenciais aplicações e serviços. No entanto para permitir uma boa interoperabilidade entre a camada MAC e o IPv6 é utilizada uma subcamada para a adaptação denominada 6LoWPAN[15].

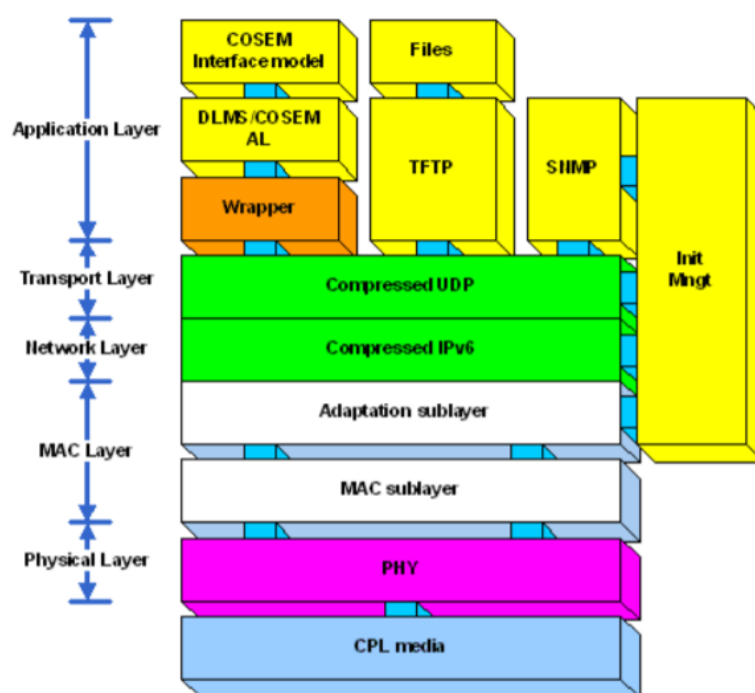


Figura 19 Pilha de protocolos G3-PLC[15]

CAMADA FÍSICA

A camada da física do G3-PLC utiliza a modulação OFDM dentro da banda de frequências CENELEC (3 kHz a 148.5 kHz). A OFDM é uma técnica que consegue eficientemente utilizar a banda CENELEC e permitindo o uso de técnicas de codificação de canal avançadas[1][14].

Num sistema OFDM a banda CENELEC é dividida num conjunto de sub-canais que podem ser vistos como portadoras independentes moduladas com PSK, cada uma com diferentes frequências de portadora sem interferência umas com as outras. A codificação

Convolutacional e de *Reed-Solomon* têm bits de redundância que permitem ao recetor recuperar bits perdidos devido ao ruído impulsivo e de fundo[14].

O sinal OFDM é gerado por uma Transformada Inversa de Fourier produzida por uma modulação de fase codificada diferencial que é alocada a cada sub-portadora. É usada uma técnica de estimador de canal cego, que com base na qualidade de receção do sinal, o recetor decide qual o esquema de modulação a ser usado. Além disso o sistema diferencia as subportadoras com um mau SNR e não transmite dados nelas[14].

ASPETOS FUNDAMENTAIS

Como se pode verificar na Figura 20 que o FEC é mais complexo que no G3-PLC comparativamente ao protocolo PRIME.

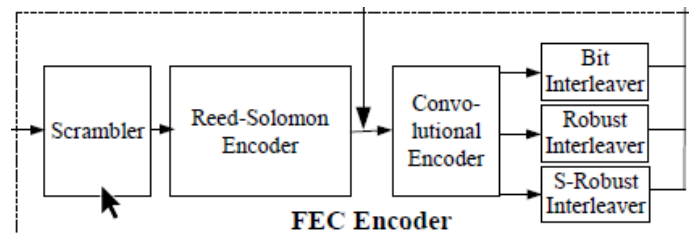


Figura 20 Processamento de transmissão G3-PLC[14]

No caso do G3-PLC, em relação ao PRIME acrescenta um Codificador Reed- Solomon (RS) e um Codificador de Repetição (*Repetition Coder* - RC) que está embebido no bloco “*Interleaver*”[1].

1. **Scrambler:** Tal como no PRIME, este bloco tem como função tornar aleatória a sequência de bits.
2. **Codificador Reed-Solomon:** O objetivo deste bloco é aumentar a robustez da comunicação a erros de canal através de informação redundante.

Os dados vindos do “*Scrambler*” são codificados por códigos sistemáticos RS (N=255, K=239, T=8) ou RS (N=255, K=247, T=4)., onde N é o número de símbolos de código gerado por cada K símbolos à entrada do codificador e T é correspondente à capacidade de correção de erros do código. Para o funcionamento em modo Robusto usa-se T=4. O tamanho das palavras de dados usadas no bloco *Reed-Solomon* é fixado em 8 bits (1 byte)[1].

3. Codificador Convolutacional: Para o G3-PLC, o bloco do Codificador Convolutacional utilizado é o mesmo que é usado na tecnologia PRIME, descrito na secção X[1].

4. Interleaver: O *Interleaver* foi concebido de forma a conseguir dar proteção contra dois tipos diferentes de erros:

- **Erros de rajada** que podem corromper alguns símbolos OFDM consecutivos[1].
- **Desvanecimento na frequência** que pode corromper algumas frequências adjacentes para um grande número de símbolos OFDM[1].

a. Modo Normal

Quando este bloco está a trabalhar em Modo Normal não é acrescentada redundância à sequência de bits[1].

b. Modo Robusto

Em Modo Robusto quando se transmite, à saída do Codificador Convolutacional, os bytes à saída são repetidos 4 ou 6 vezes, dependendo da utilização de um RC de fator 4 ou 6. O RC de fator 4 é usado exclusivamente para a transmissão de dados e no caso do RC de fator 6 é uma forma mais Robusta para codificação do *Frame Control Header* (FCH)[1].

5. Modulação

É escolhido um esquema de modulação OFDM, onde as portadoras são moduladas através de DBPSK ou DQPSK para se conseguir uma taxa de transmissão máxima de 33,4 kbit/s, em modo normal de operação. A escolha da modulação para cada portadora simplifica muito o recetor, não sendo assim necessário um circuito para detetar a fase[1].

ESTRUTURA DA TRAMA

A camada PHY de G3-PLC suporta 2 tipos de trama. A trama típica para a PHY do OFDM pode ser vista na Figura 21.

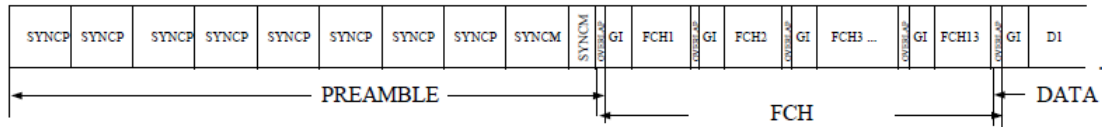


Figura 21 Estrutura típica da trama G3-PLC[14]

Cada trama começa com o Preambulo (*Preamble*), que é utilizado para sincronização. O Preambulo é seguido de 13 símbolos de dados alocados ao FCH, que tem a informação de controlo necessária para desmodular a trama de dados (*Data*). Os dados são transmitidos de seguida, onde existe um intervalo de guarda[14].

A camada PHY suporta também uma trama ACK/NACK, que consiste apenas no Preambulo e no FCH, onde os campos dos bits no FCH vão funcionar com sinalização de *acknowledged* (ACK) e *not acknowledged* (NACK) como se pode ver na Figura 22[14].

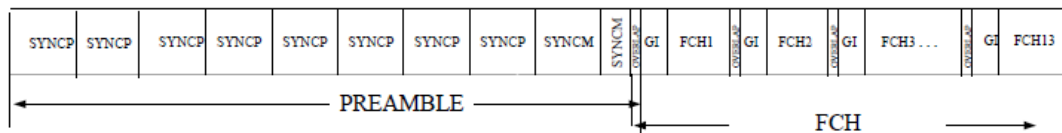


Figura 22 Estrutura típica da trama G3-PLC[14]

PARÂMETROS

Para o G3-PLC, pode-se ver na Tabela 8 o número de configurações possíveis mediante a escolha do modo de operação e do número de símbolos usado. Na tabela 8 pode-se ver as taxas de transmissão para os diferentes tipos de modulações possíveis.

Tabela 8 Parâmetros G3-PLC[1]

	Blocos Reed Solomon DQPSK	Blocos Reed Solomon DBPSK	Blocos Reed Solomon Robusto
Número de símbolos OFDM	(Out/In) Bytes P16 ⁷	(Out/In) Bytes P16 ⁷	(Out/In) Bytes P8 ⁸
12	(53/37)	(26/10)	N/A
20	(89/73)	(44/28)	N/A
32	(143/127)	(71/55)	N/A
40	(179/163)	(89/73)	(21/13)
52	(233/217)	(116/100)	(28/20)
56	(251/235)	(125/109)	(30/22)
112	N/A	(251/235)	(62/54)
252	N/A	N/A	(141/133)

Tabela 9 Taxas de transmissão para diferentes modulações[1]

Número de símbolos OFDM	Ritmo de Dados (DQPSK) bit/s P16 ⁷	Ritmo de Dados (DBPSK) bit/s P16 ⁷	Ritmo de Dados (Robusto) bit/s P8 ⁸
12	12103	3271	N/A
20	19456	7462	N/A
32	26489	11471	N/A
40	29693	13298	2423
52	33221	15309	3121
56	34160	15844	3257
112	N/A	20009	4647
252	N/A	N/A	5592

2.5.3. COMPARAÇÃO ENTRE PRIME E G3-PLC

Resumindo as características das duas tecnologias estudadas, pode-se ver na tabela 10 as principais diferenças e semelhanças entre as mesmas.

Tabela 10 Parâmetros G3-PLC e PRIME[16].

	PLC G3	PRIME
frequency range	35–91 kHz	42–89 kHz
sampling frequency f_s	400 kHz	250 kHz
OFDM		
FFT size M	256	512
length of cyclic prefix L_{CP}	30	48
windowing	yes	no
subcarrier spacing Δf	1.5625 kHz	488 Hz
No. of carriers used (one-sided)	36	97
max. data rate	33.4 kbps	128.6 kbps
Forward Error Correction	Reed Solomon code, convolutional code, repetition code	convolutional code
interleaving	per data packet	per OFDM symbol
modulation differential encoding	DBPSK, DQPSK in time	DBPSK, DQPSK, D8PSK in frequency

O PRIME e G3-PLC são os protocolos em análise, que permite concluir que apesar de diferentes características como o intervalo de frequências, ambos utilizam modulação OFDM e apesar de a codificação utilizada por ambos ser FSK, o G3-PLC é modulado no tempo e o PRIME na frequência[1][16].

3. PROTÓTIPO

Para o desenvolvimento do sistema de identificação de baterias pela linha de alimentação DC, foi necessário fazer o dimensionamento, ou seja, fazer o levantamento das necessidades do mesmo.

Como é ilustrado na Figura 23, os dados a transmitir são transportados pela linha de alimentação de uma bateria, tendo numa das extremidades do diagrama o carregador e em lado oposto a bateria. O equipamento previamente existente no sistema, não tem qualquer comunicação, mas sim o transporte de corrente contínua.

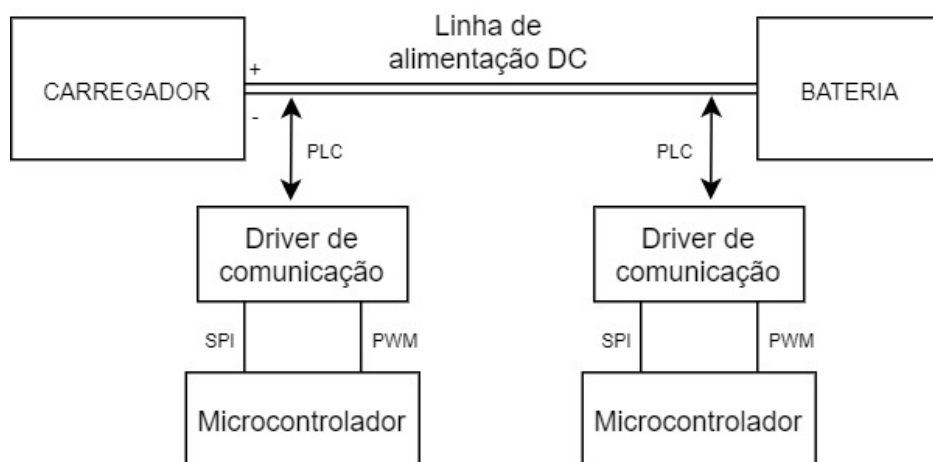


Figura 23 Diagrama de blocos do sistema de comunicação PLC

Foi realizada uma pesquisa sobre as ferramentas existentes no mercado que possam permitir a comunicação PLC, com acoplamento à linha DC.

Durante a pesquisa de equipamentos foram encontradas duas possíveis hipóteses:

- AFE031
- PLC360

Ambos os equipamentos são idênticos nas suas características, como na utilização dos protocolos PLC certificados, como o PRIME e o G3; utilização das bandas de frequência CENELEC A, B, C e D, tal como a utilização de SPI, PWM e DAC, ambos com capacidade de acoplamento à linha.

Durante a pesquisa, o AFE031 da *Texas Instruments* destacou-se por ser um equipamento especificamente desenhado para a transmissão numa linha de alimentação DC. Este integrado contém uma enorme quantidade de documentação, juntamente com um módulo de desenvolvimento *BOSTXL-AFE031-DF1*, que permite o acoplamento à linha e envio de dados PLC.

O PLC360 da *Microship* apesar de ter características idênticas ao AFE031, tem uma menor qualidade na documentação apresentada e tem igualmente um kit de desenvolvimento denominado *PLC360-EK Evaluation Kit*.

Apesar das características idênticas dos dois equipamentos, a decisão recaiu sobre o AFE031, devido a dois pontos: a possibilidade de acesso ao integrado sem custos e o maior número e maior facilidade de interpretação da documentação.

3.1. AFE031

O AFE031 é um dispositivo integrado *analog front end* (AFE) que permite a comunicação sobre linhas de tensão AC e DC, capaz de acoplamento capacitivo ou por transformador sobre o controlo de um microcontrolador, funcionando como *driver* de comunicação.

Este dispositivo opera com alimentações que variam de 7V até 24V. Onde o circuito de processamento funciona a uma alimentação de 3,3V.

O recetor deteta sinais PLC até $20 \mu V_{RMS}$, e o amplificador integrado opera linhas de baixa impedância que requerem até 1,5 A para cargas reativas.

Para possível utilização deste integrado a *Texas Instruments* tem um módulo de teste, *BOSTXL-AFE031-DF1* demonstrado na Figura 24 que oferece uma plataforma robusta para as aplicações PLC.

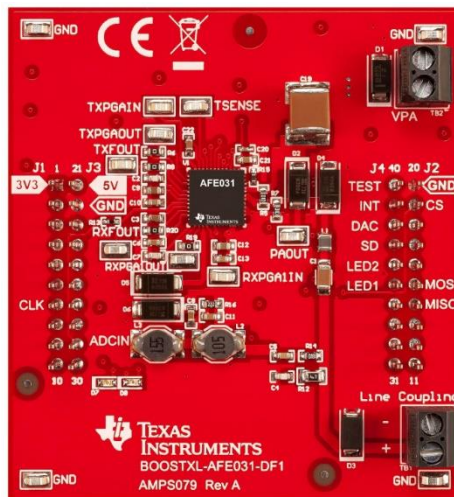


Figura 24 Modulo de avaliação BOSTXL-AFE031-DF1

O BOSTXL-AFE031-DF1 funciona como peça central num sistema integrado de PLC.

Este modulo tem aplicações em diversas áreas como monitorização, automação, carregamento de baterias e energia solar.

Na Figura 25 está representado o diagrama do sistema do kit de desenvolvimento, onde temos o bloco do controlador, com a comunicação por SPI e ADC o BOSTXL-AFE031-DF1 e o acoplamento à linha.

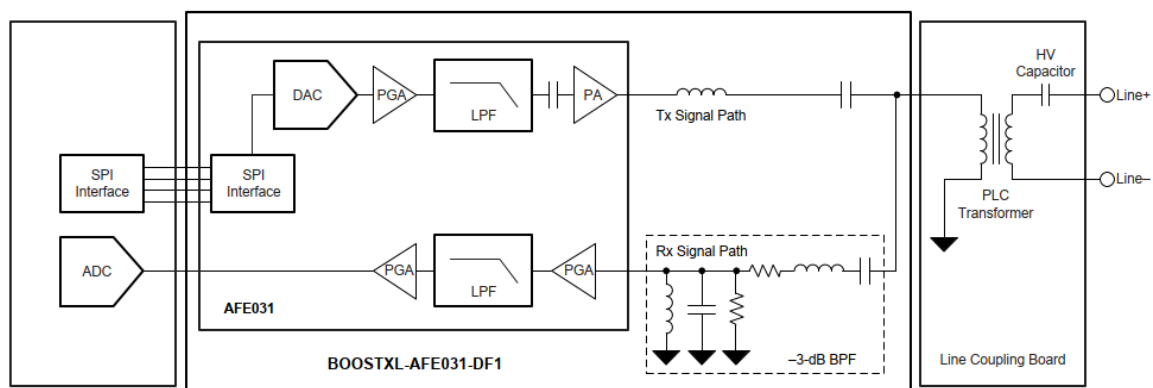


Figura 25 Diagrama do sistema PLC[17]

É também incluído no modulo um filtro passa-banda a -3dB com intuito de atenuar o sinal de ganho de tensão criado no transformador externo da linha de acoplamento e melhorar a

rejeição do ruído. O sinal filtrado é amostrado por um *analog-to-digital-converter* (ADC) e desmodulado do lado do controlador para descodificar os dados PLC.

Relativamente ao acoplamento à linha DC, o fornecedor do AFE031, a *Texas Instruments* recomenda o esquema na Figura 26, onde à saída do AFE031 é inserido um condensador de $10\mu\text{F}$ antes do acoplamento à linha, neste caso como exemplo uma linha de 24V. Este esquema mostra que o AFE031 é possível criar uma rede com diversos equipamentos a comunicar entre si pela mesma linha DC.

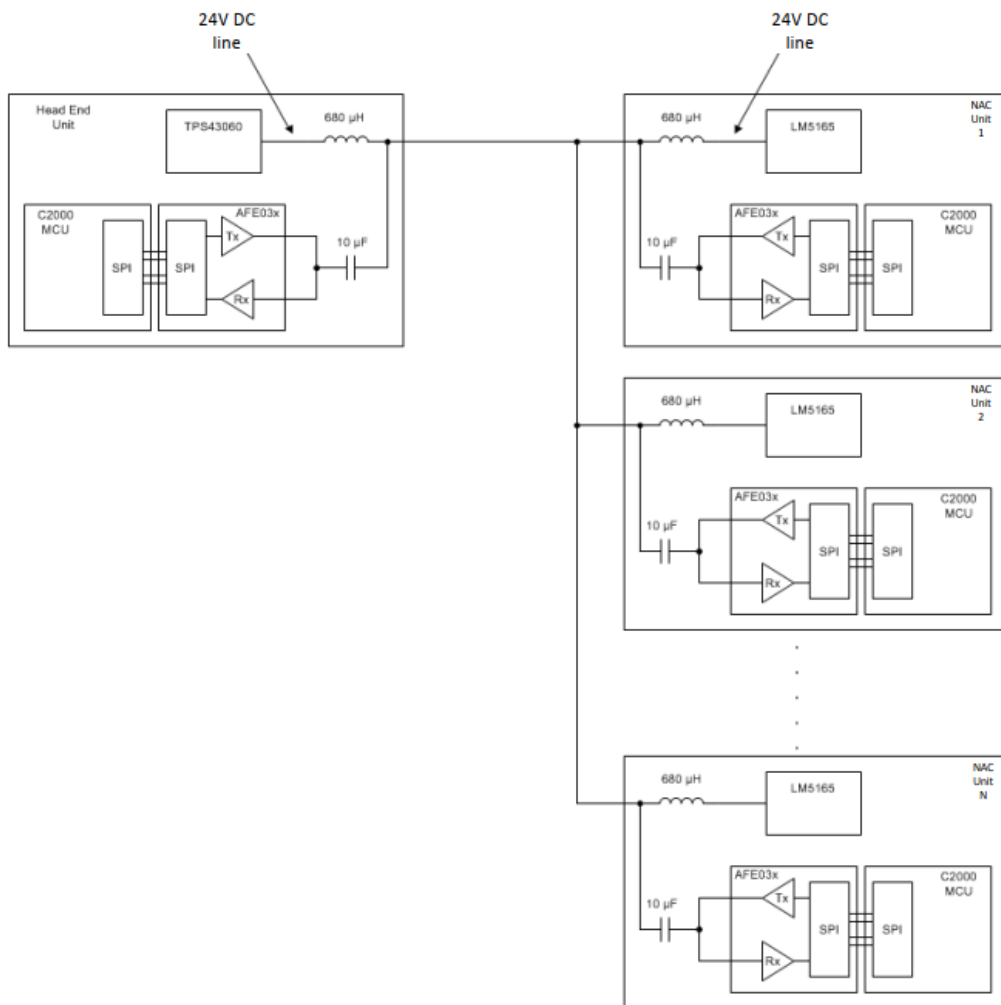


Figura 26 Exemplo de vários módulos acoplados à linha DC[9]

Dentro do esquema temos vários pontos de entrada no sistema. O VPA é o ponto onde o circuito é alimentado sendo a alimentação segundo o *datasheet* do integrado a 15V. Esta alimentação serve o amplificador de tensão (PA), utilizado para a injeção de sinal na linha de transmissão[18].

A alimentação do integrado em si para o seu funcionamento normal será de 3.3V, incluindo assim um circuito abaixador, alimentação denominada no circuito de VDD. Na Figura 28 pode-se ver o circuito abaixador, com filtro à entrada e à saída do conversor utilizado, o L7805.

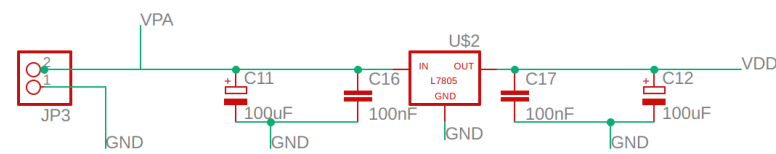


Figura 28 Circuito abaixador de tensão

Para uma análise mais cuidada do esquema elétrico é preciso analisar o diagrama de blocos funcional do AFE031, demonstrado na Figura 29.

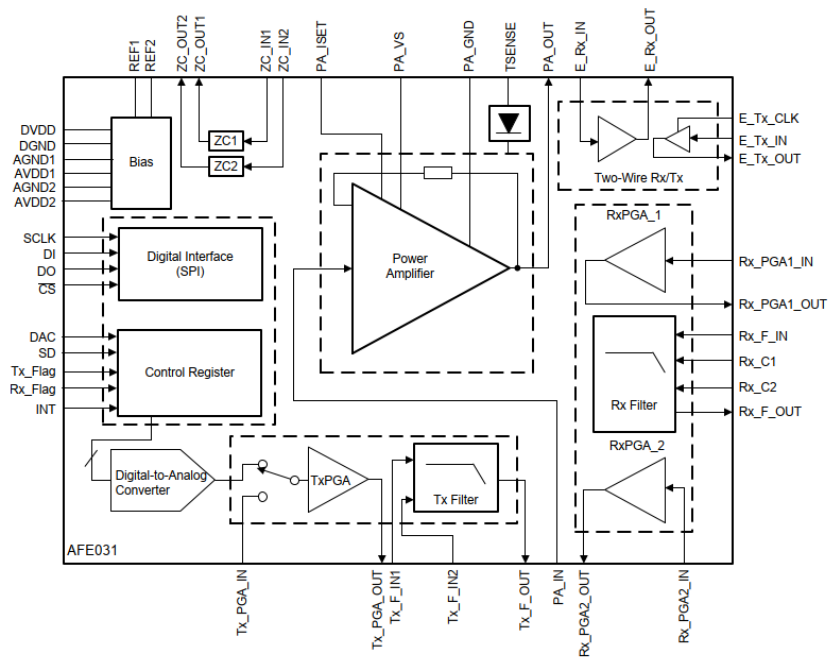


Figura 29 Diagrama de blocos funcional do AFE031[18]

A ligação ao amplificador de tensão numa conexão típica para uma aplicação PLC requiere dois componentes adicionais: um condensador de acoplamento AC e uma resistência de limitação de corrente, como mostra na Figura 30, à entrada do pino PA_IN e no PA_ISET respetivamente. O condensador C1 introduz características de passa-alto à função de transferência do amplificador de tensão. O valor da frequência de corte depende do valor do condensador. A resistência R8 é respetiva ao cálculo da limitação de corrente no amplificador de tensão. O PA_OUT é a saída do bloco do PA e a saída do sinal a transmitir seguindo para o circuito de proteção ao acoplamento à linha.

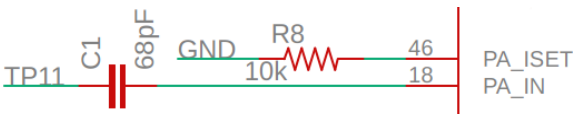
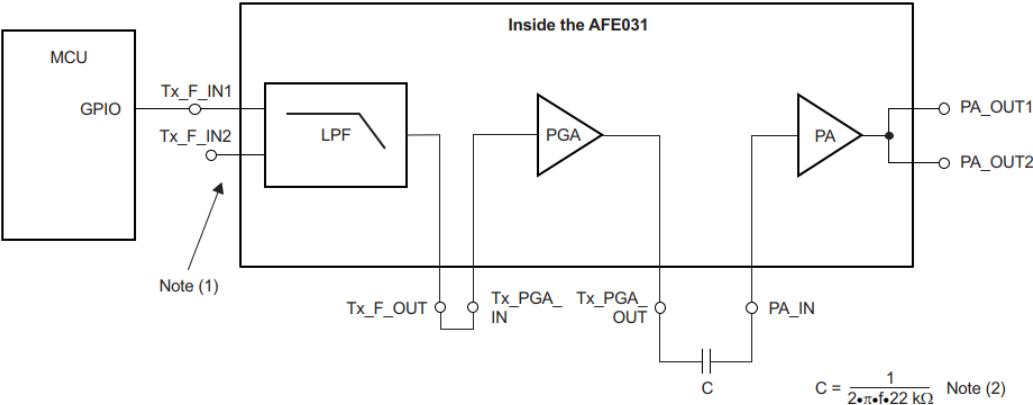


Figura 30 Ligações ao amplificador de tensão

Relativamente à transmissão, o bloco Tx consiste no Tx PGA e o Tx Filter. O Tx PGA é um amplificador de ganho. O ganho deste amplificador é programado por SPI, sendo que as opções disponíveis de ganho são 0.25V/V, 0.5V/V, 0.707V/V e 1V/V. O Tx Filter é um filtro passa-baixa de quarta ordem. A frequência de corte é seleccionável entre os modos CENELEC A ou CENELEC B, C e D.

Para o modo PWM o GPIO do microprocessador pode ser ligado diretamente à entrada do Tx Filter, Tx_F_IN, como é mostrado na Figura 31.



$$C = \frac{1}{2 \cdot \pi \cdot f \cdot 22 \text{ k}\Omega} \text{ Note (2)}$$

Figura 31 Ligações em modo PWM[18]

Relativamente à receção o bloco Rx consiste nos Rx PGA1, Tx PGA 2 e Rx Filter. Os Rx PGA1 e 2 são amplificadores de ganho, podendo ser programados para fazer a atenuação ou ganho. O Rx Filter é um filtro passa-baixo de quarta ordem, onde a frequência de corte entre os modos CENELEC A ou CENELEC B, C e D. Devido ao facto do Rx Filter ser um filtro analógico de baixo ruído, são necessários dois condensadores externos para o configurar, como se pode ver na Figura 32.

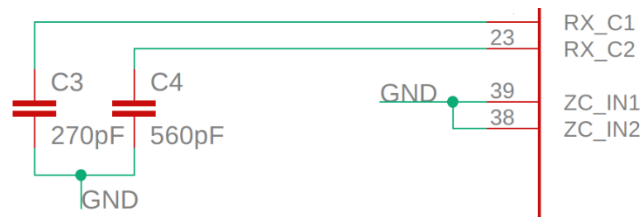


Figura 32 Condensadores externos do Rx Filter

No entanto um filtro passivo passa-banda de quarta ordem é recomendado, como é mostrado na Figura 33, pois remove qualquer sinal no caminho fora da banda de frequências prevenindo a chegada dos mesmos aos filtros ativos de entrada dentro do integrado.

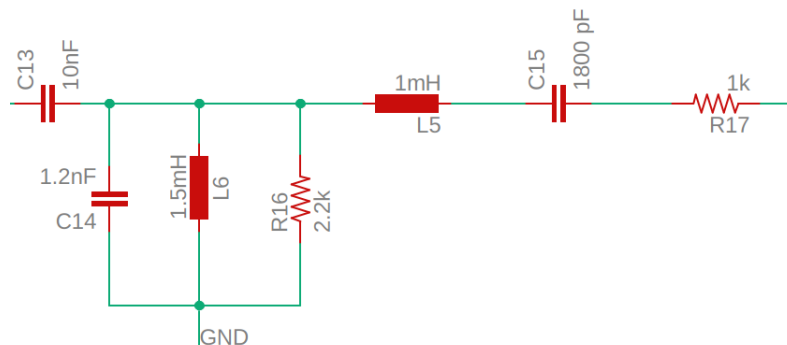


Figura 33 Filtro passa banda à entrada do RX PGA1

Relativamente aos blocos REF1 e REF2, cada um divide a sua respetiva tensão de alimentação a metade com um divisor de tensão resistivo. O REF1 fornece uma tensão utilizada para o amplificador de tensão, enquanto REF2 fornece uma tensão usada para o TX PGA, Tx Filter, RX PGA1. Rx Filter e Rx PGA2. Cada bloco tem um pino externo como saída que pode ser usado para filtragem e redução de ruído. Podemos ver na figura 34 o circuito com as ligações aos condensadores externos, redutores de ruído. Estes condensadores são opcionais, mas recomendados.

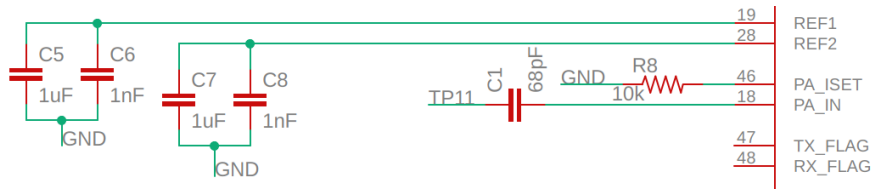


Figura 34 Circuito de ligações aos condensadores externos

As resistências R1, R2, R3, R4, R5, R6 e R7 funcionam como resistências de *pull-up*, nos pinos dedicados à comunicação por SPI.

Os pontos de contacto com o AFE031 para a comunicação foram transpostos separadamente como pinos de conexão como se pode ver na Figura 35. Nestes pinos encontram-se os pontos destinados à comunicação por SPI, DAC e diferentes *test points* (TP), estes com diferentes funcionalidades. O TP10 é o *test point* que conecta o TX_F_IN, sendo este o pino do AFE031 que em caso de modo PWM, recebe o sinal vindo do microcontrolador.

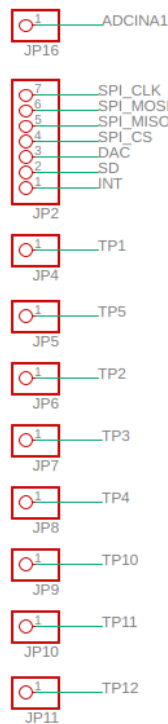


Figura 35 Pontos de conexão para a comunicação com o AFE031

Por último, o ponto de acoplamento à linha, ou seja, a saída do sinal PLC é o ponto designado como JP1 no circuito e como TB1 no *datasheet* do integrado como se pode ver na Figura 36.

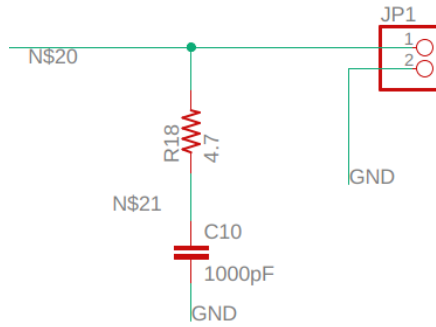


Figura 36 Parte do circuito de saída de acoplamento à linha

3.2.2. DESENVOLVIMENTO PCB

Após a finalização do desenho das ligações, uma das opções do *EAGLE* é a transposição do esquema de ligações para a placa. Os equipamentos já estão inseridos no *software*, onde o programa tem a informação do modelo dos componentes, ou seja, o seu tamanho e os seus pontos de ligação.

O próximo passo foi definir as dimensões da placa, estas que foram definidas pelo departamento na instituição de ensino, que restringe o tamanho das placas PCB a serem contruídas no departamento, ou seja, 10x8cm.

Após a distribuição dos componentes pelo espaço disponível, foi desenvolvido no departamento de Engenharia Eletrotécnica.

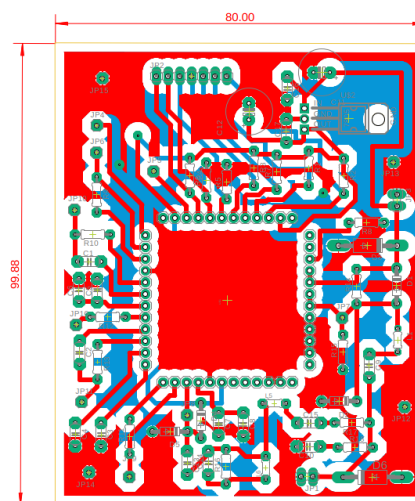


Figura 37 Placa em PCB do protótipo

Para este processo foi impresso o desenho da placa, das duas faces, para, por radiação infravermelho, marcar as pistas na PCB.

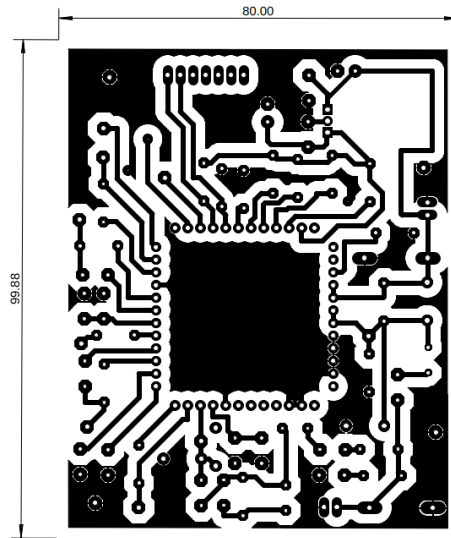


Figura 38 Face superior a desenhar na placa PCB

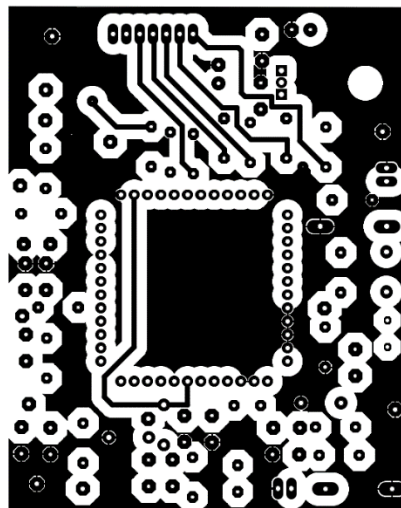


Figura 39 Face inferior a desenhar na placa PCB

Por último foi feito a furação da placa, a montagem dos componentes e a respetiva soldadura. Na Figura 40 pode-se ver o protótipo final.

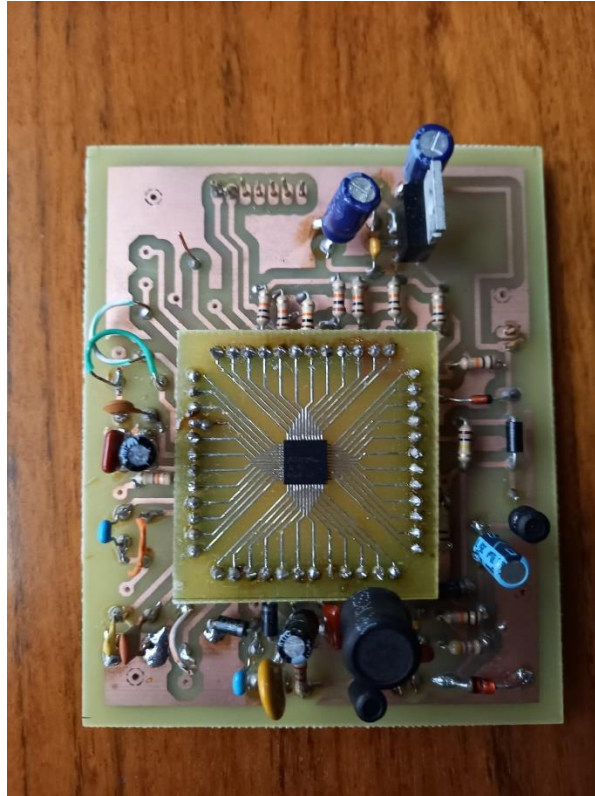


Figura 40 Protótipo montado

Foi feito um teste de continuidade em todas as ligações da placa, onde a mesma se verificou em todos os pontos.

3.3. ATMEGA 328P

O microcontrolador escolhido para o sistema é o Atmega 328P da *Microchip*. Apesar do fornecedor do AFE031 sugerir um microcontrolador do seu fornecimento, da família de microcontroladores C2000, foi escolhido o Atmega328P em detrimento deste, devido a um baixo custo, grande disponibilidade no mercado, de fácil acesso, grande comunidade de desenvolvimento de produtos que integram o ATmega 238P utilizados em larga escala para placas de desenvolvimento e uma arquitetura simples.

Este microcontrolador pertence a família dos AVR de 8 Bit, com as seguintes características:

- Dois *Timer/Counters* de 8 bits com *Prescaler* e *Compare Mode*.
- Um *Timer/Counter* de 16 bits com *Prescaler*, *Compare Mode* e *Capture Mode*.
- Seis canais de PWM

- Quatorze canais ADC
- USART
- Comunicação SPI *master/slave*
- Alimentação de 1.8V a 5.5V
- Velocidade de *Clock* é suficiente 8MHz

Este integrado tem todas as características necessárias para a execução do projeto, ou seja, para a comunicação com o AFE031 é necessário comunicar por SPI e para a comunicação para linha de transmissão tem compatibilidade com os dois modos possíveis de funcionamento do AFE031, o modo PWM e o modo DAC.

A alimentação do microcontrolador é compatível com a do AFE031 entre os 1.8V aos 5V, tal como a velocidade máxima do *clock* do integrado é compatível com o valor de frequências que são pretendidas na comunicação, valores que podem chegar aos 8MHz.

4. IMPLEMENTAÇÃO

4.1. COMUNICAÇÃO

O sistema ilustrado no diagrama de blocos da Figura 41, consiste no microcontrolador ATmega328P que comunica por SPI com o AFE031, *driver* de comunicação PLC. O sinal a injetar através do circuito de acoplamento à linha é gerado no microcontrolador, uma onda PWM enviada à ao AFE031.

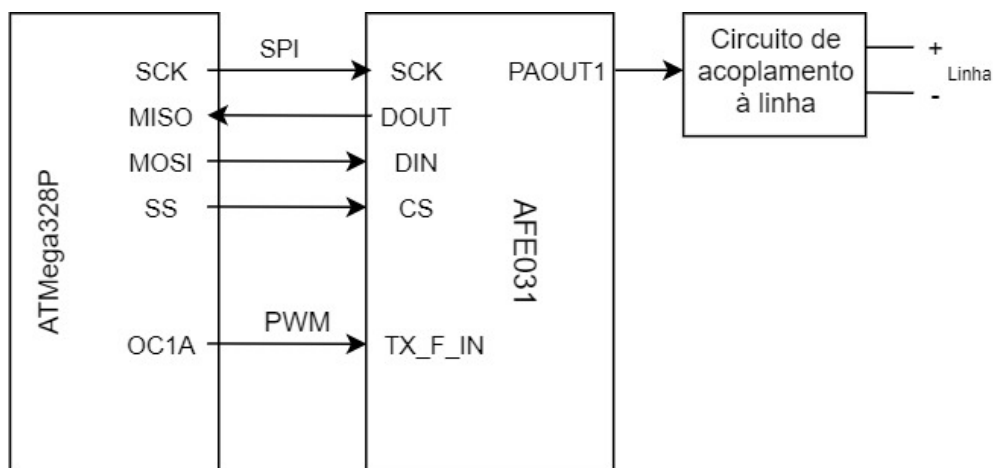


Figura 41 Diagrama de blocos da interface entre o controlador e o AFE031

O AFE031 suporta duas formas de transmissão de dados, *Pulse Width Modulation* (PWM) e DAC. Tem igualmente múltiplos registos internos que permitem a configuração interna do chip AFE, como a seleção dos modos, filtros e ganho. Estes registos são configurados por SPI.

Para configurar o AFE031 como transmissor, o primeiro passo é decidir o modo de transmissão, DAC ou PWM.

Em modo DAC, os dados são enviados por SPI para atualizar o DAC interno do AFE031, onde o valor de saída é filtrado e amplificado. Neste modo, a tabela de sinal é corrida ciclicamente e cada ponto é enviado por SPI.

Para o caso do modo PWM, a comunicação por SPI é só usada para configurar os registos no AFE031. De seguida são gerados um ou dois sinais PWM e enviados para o AF031.

Na Figura 42 é apresentado o fluxograma da função principal do *software* desenvolvido. Após dar início, é feita a configuração do microcontrolador, de seguida é feita a configuração do AFE031 e por fim é configurado uma interrupção durante o envio contnuo de dados.

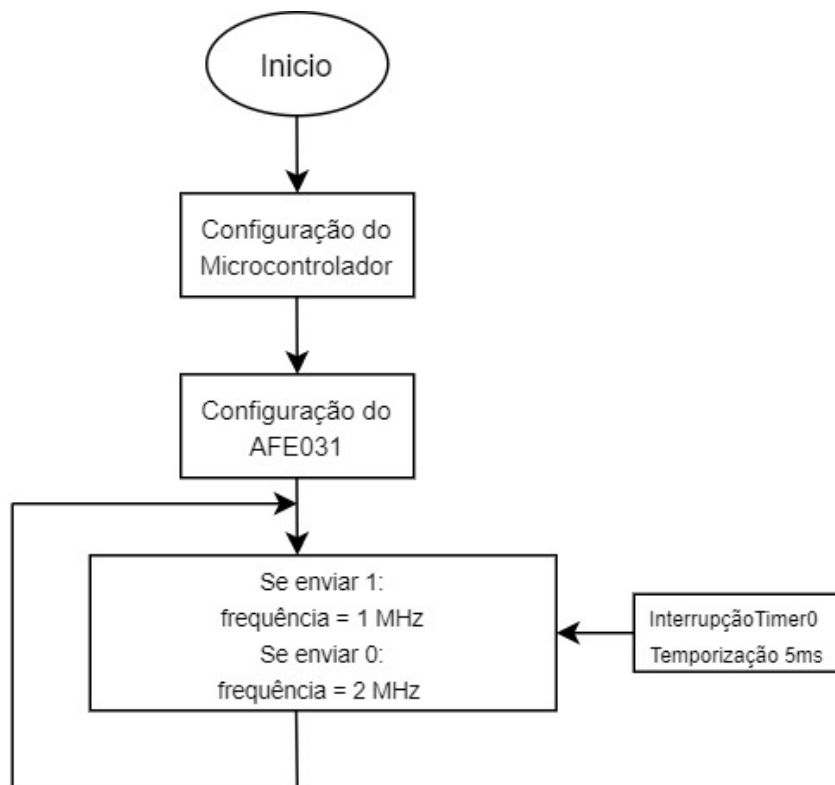


Figura 42 Fluxograma da função principal

4.1.1. CONFIGURAÇÃO AFE031

Há dois passos para configurar o AFE031, colocar os dois principais *General Purpose Input/Output* (GPIO), o *Mode Select* e *System Shutdown* a *Low* e configurar o dispositivo por SPI.

Para a configuração inicial do AFE031 demonstrada no fluxograma da Figura 43, são seguidos os seguintes passos:

1. Configurar os GPIO – configurar os pinos SD e DAC como *low*.
2. Configurar o SPI para caracteres de 16 bits.

3. Fazer um *soft reset*- Escrever um 0x14 no registo de *reset*.
4. Habilitar o *Bias*-Escrever 0x03 no *enable2*.
5. Escolher a frequência de banda-Escrever 1 ou 0 ao bit *CA_CBCD*, para escolher CENELEC B, C, D ou CENELEC A respetivamente.
6. Limpar interrupções – Escrever 0x00 no registo *control2*.
7. Configurar Interrupções – Escrever no registo *control2*.
8. Habilitar ZC – Escrever 1 no bit *zc* no registo *enable2*.
9. Escrever o ganho para o caso de transmissão-Escrever 0, 1, 2, 3 nos TXG bits no registo de seleção de ganho.
10. Escrever o ganho para o de receção.
11. Finalizar as configurações necessárias baseados na implementação a ser utilizada para a transmissão e receção.

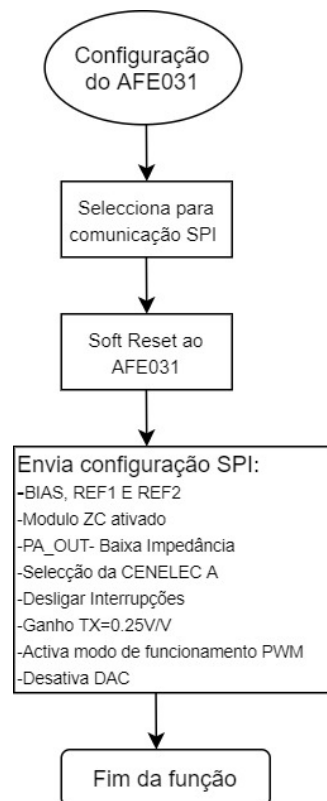


Figura 43 Fluxograma da configuração do AFE031

A plataforma de desenvolvimento utilizada para a programação foi o *Atmel Studio*, pois é o IED (*Integrated Development Environment*) oficial do fornecedor do microcontrolador, *Atmel*.

4.1.2. CONFIGURAÇÃO DO MICROCONTROLADOR

Para a configuração do microcontrolador como é possível verificar no fluxograma da Figura 44, consiste na configuração dos portos do Atmega328P, seguida da configuração da comunicação por SPI. É feita a configuração do Timer1 e do Timer0, para gerar a onda de sinal PWM e uma interrupção com temporização de 5ms.

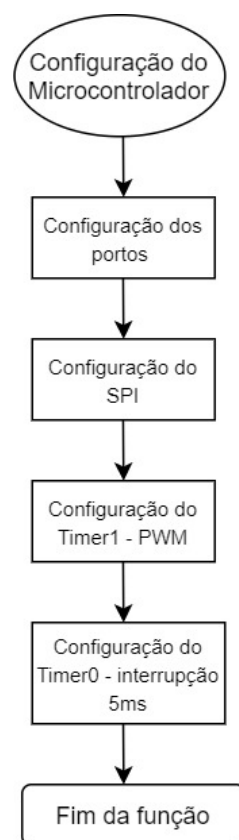


Figura 44 Fluxograma da configuração do microcontrolador

CONFIGURAÇÃO SPI

A comunicação por SPI é necessária para a configuração inicial do dispositivo AFE031. Para isso tem de se estabelecer uma comunicação entre os dois dispositivos, o controlador Atmega 328P e o dispositivo AFE031.

Para o caso do Atmega 328P, os quatro pinos destinados à comunicação SPI são, o SCK, o MISO, o MOSI e o SS, que correspondem no controlador respectivamente aos pinos, PB5, PB4, PB3, PB2 como se pode ver na Figura 45.

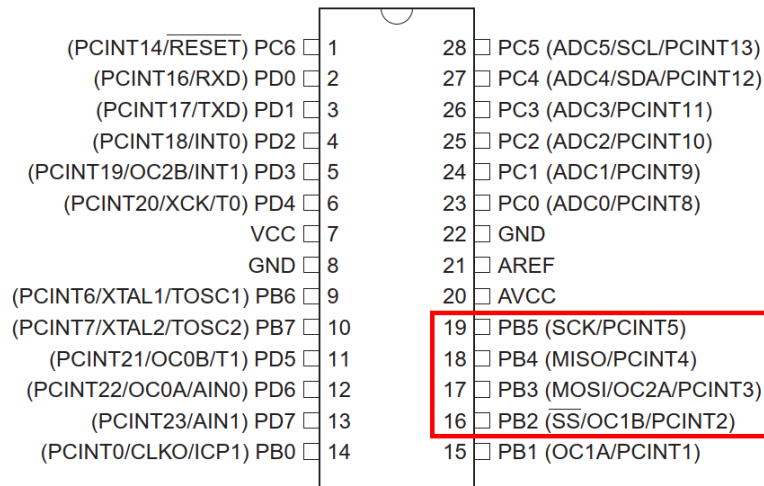


Figura 45 Configuração dos pinos no Atmel 328P – Pinos destinados à comunicação SPI

No lado oposto da comunicação, os quatro pinos destinados à comunicação SPI são, o SCLK, o DIN, o DOUT e o CS, que correspondem no controlador respectivamente aos pinos, 3, 4, 5 e 6 como se pode ver na Figura 46.

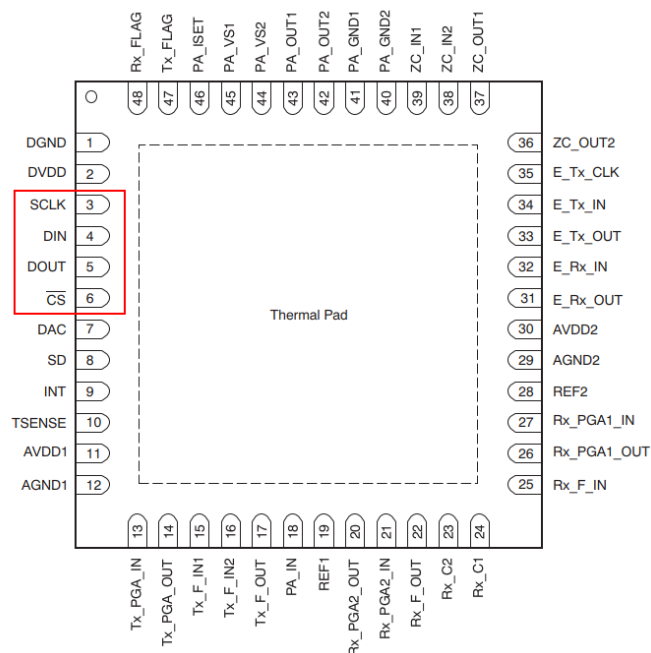
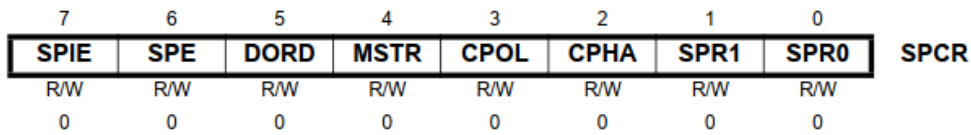


Figura 46 Configuração dos pinos no AFE031 – Pinos destinados à comunicação SPI

Para a configuração o SPI é preciso configurar três registos, o *SPI Control Register (SPCR)*, *Status Register (SPSR)* e *Data Register (SPDR)*.

SPCR:



- **SPIE** – Habilita a interrupção SPI, este bit será posto a 0.
- **SPE**-Habilita o SPI, onde este bit será colocado a 1.
- **MSTR**-É colocado a 1 para seleccionar o modo de funcionamento como *Master*.
- **CPOL**-Este bit é posto a 0 de acordo com a tabela 11..

Tabela 11 Funcionalidade CPOL

CPOL	Leading Edge	Trailing Edge
0	Rising	Falling
1	Falling	Rising

- **CPHA**- Neste bit determina se o *sample* de dados é feito na primeira parte ou na última do SCK.

Tabela 12 Funcionalidade CPHA

CPHA	Leading Edge	Trailing Edge
0	Sample	Setup
1	Setup	Sample

- **SPR1 e SPR0**

Na tabela é escolhido o valor de *prescaler* que neste caso será o valor 2.

Tabela 13 Relação entre SCK e a frequência de oscilador

SPI2X	SPR1	SPR0	SCK Frequency
0	0	0	$f_{osc}/4$
0	0	1	$f_{osc}/16$
0	1	0	$f_{osc}/64$
0	1	1	$f_{osc}/128$
1	0	0	$f_{osc}/2$
1	0	1	$f_{osc}/8$
1	1	0	$f_{osc}/32$
1	1	1	$f_{osc}/64$

Este registo é configurado da seguinte forma:

$$SPCR = 01010000$$

SPCSR:

7	6	5	4	3	2	1	0	
SPIF	WCOL	–	–	–	–	–	SPI2X	SPSR
R	R	R	R	R	R	R	R/W	
0	0	0	0	0	0	0	0	

- **SPIF**-Bit relativo à *flag* de interrupção
- **SPI2X**-Bit relativo à duplicação da velocidade de SPI.

Este registo é configurado da seguinte forma:

$$SPSR = 00000001$$

O conjunto de configurações iniciais acima para comunicação SPI, traduzem-se no código em C abaixo:

```
void config_io()
{
    DDRB |= (1 << PB3) | (1 << PB5) | (1 << PB2);
    DDRB &= ~(1 << PB4);
    PORTB |= (1 << PB1);
    SPCR = (1 << SPE) | (1 << MSTR);
    SPSR |= (1 << SPI2X);
}
```

Foi criada uma função genérica para o envio de dados SPI. Esta função denominada SPI_enviar(uint8_t endereço, uint8_t dados) recebe o endereço e os dados a enviar, ou seja,

o registo a escrever nesse mesmo endereço. Nesta função é ativado o início da transmissão com a ativação do *slave*.

```
void SPI_enviar(uint8_t endereco, uint8_t dados)
{
    PORTB &= ~(1 << PB2);
    SPDR = ( (0 << 6) + (endereco));
    while(!(SPSR & (1<<SPIF)));
    SPDR = dados;
    while(!(SPSR & (1<<SPIF)));
    PORTB |= (1 << PB2);
}
```

Com a função de envio de dados SPI criada, foi construída outra função para fazer a configuração necessária do AFE031. Os passos descritos nesta configuração já foram descritos no capítulo 4. É utilizada a função “SPI_enviar” para escrever um endereço no respetivo registo.

```
void config_afe031()
{
    DAC_PORT &= ~(1 << DAC_PIN);
    SD_PORT &= ~(1 << SD_PIN);
    SPI_enviar(RESET, 0x14);
    SPI_enviar(ENABLE2, 0b00001111);
    SPI_enviar(CONTROL1, 0x00);
    SPI_enviar(CONTROL2, 0x00);
    (GAIN_SELECT, TX_GAIN_0_25VV);
    SPI_enviar(ENABLE1, 0b00000011);
}
```

CONFIGURAÇÃO TIMER 1 - PWM

O modo de comunicação escolhido no AFE031 é o PWM. A escolha recaiu sobre o modo de comunicação PWM, devido à sua maior simplicidade em comparação com o modo DAC que se previa uma implementação mais complexa, apesar de, segundo o fornecedor do AFE031, o modo PWM oferecer menos qualidade de sinal. Outra das razões é a existência de um maior conhecimento técnico sobre PWM, adquirido durante o curso.

O objetivo do código é gerar uma sequência de 8 bits (duração de uma palavra), onde as alterações de frequência do PWM indicam a alteração do estado. Cada bit tem a duração de 5ms, ou seja, durante este tempo não pode haver alteração de frequência.

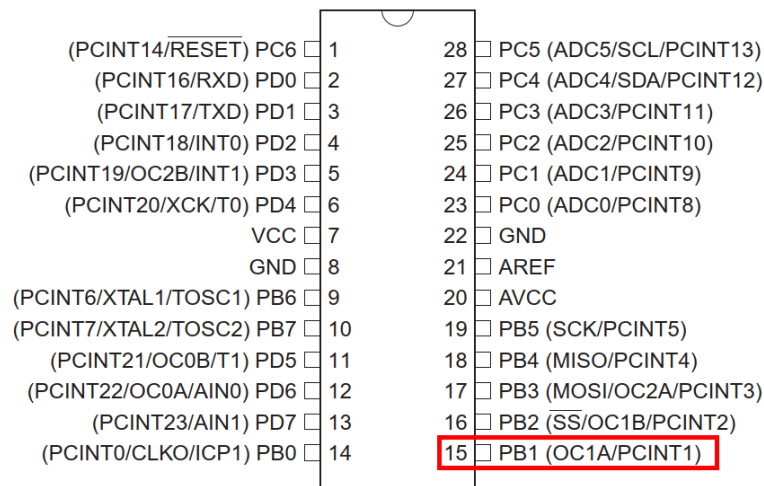


Figura 47 Configuração dos pinos do Atmel 328P – Pinos destinados ao sinal PWM

O primeiro passo para configuração PWM no Atmega 328P é a escolha das frequências de transmissão, uma a funcionar como o estado 0 e outra a funcionar como estado 1:

$$f_{PWM_NIVEL0} = 2MHz$$

$$f_{PWM_NIVEL1} = 1MHz$$

Para conseguirmos este valor de frequência temos de escolher um modo de operação que o permita.

O modo selecionado é o *Fast* PWM do Timer 1 que permite PWM de 16 bit, pois é possível a utilização de frequências mais altas.

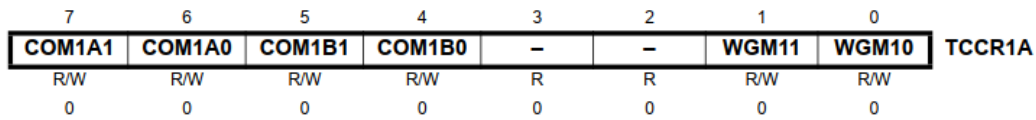
$$f_{OCnxPWM} = \frac{f_{CLK_{IO}}}{N \times (1 + TOP)}$$

$$1MHz = \frac{8MHz}{1 \times (1 + 7)}$$

Onde o valor TOP é um valor entre 0 a $(2^{16} - 1)$ valor máximo a 16 bits, sendo que neste interessa ter o valor mais baixo possível, e o N é o valor de *prescaler*, de acordo tabela 16, sendo que neste caso interessa ser o menor valor possível, ou seja 1.

Iniciando a configuração dos registos:

TCCR1A (Time/Counter 1 Control Register A):



O bit COM1A1/0 e COM1B1/0 controlam o comportamento dos pinos de saída OC1A e OC1B, para este caso só tem importância o OC1A, sendo o *Output Compare* escolhido.

A opção escolhida para o registo COM1A1 e COM1A0 de acordo com a tabela, é o *Toggle* em caso de *Compare Match*.

Tabela 14 Modo de comparação de saída Timer1

COM1A1/COM1B1	COM1A0/COM1B0	Description
0	0	Normal port operation, OC1A/OC1B disconnected.
0	1	WGM13:0 = 14 or 15: Toggle OC1A on Compare Match, OC1B disconnected (normal port operation). For all other WGM1 settings, normal port operation, OC1A/OC1B disconnected.
1	0	Clear OC1A/OC1B on Compare Match, set OC1A/OC1B at BOTTOM (non-inverting mode)
1	1	Set OC1A/OC1B on Compare Match, clear OC1A/OC1B at BOTTOM (inverting mode)

Juntamente com os bits WGM12 e WGM13 presentes no registo TCCR1B, estes bits controlam a sequência de contagem do contador, o valor TOP e que tipo de geração de onda será usado.

Na tabela seguinte, para termos a configuração dos registos devemos escolher o modo de operação utilizado e qual será o valor de *TOP* a utilizar. Sendo que será *Fast PWM* e para o valor de *TOP* o OCR1A.

Tabela 15 Descrição de bit do modo de geração de onda do Timer1

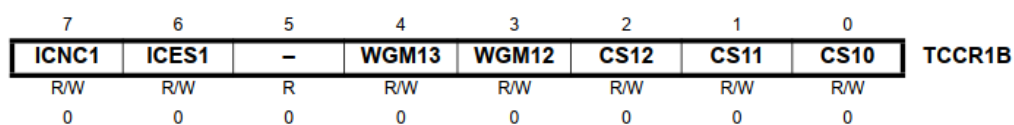
Mode	WGM13	WGM12 (CTC1)	WGM11 (PWM11)	WGM10 (PWM10)	Timer/Counter Mode of Operation	TOP	Update of OCR1x at	TOV1 Flag Set on
0	0	0	0	0	Normal	0xFFFF	Immediate	MAX
1	0	0	0	1	PWM, Phase Correct, 8-bit	0x00FF	TOP	BOTTOM
2	0	0	1	0	PWM, Phase Correct, 9-bit	0x01FF	TOP	BOTTOM
3	0	0	1	1	PWM, Phase Correct, 10-bit	0x03FF	TOP	BOTTOM
4	0	1	0	0	CTC	OCR1A	Immediate	MAX
5	0	1	0	1	Fast PWM, 8-bit	0x00FF	BOTTOM	TOP
6	0	1	1	0	Fast PWM, 9-bit	0x01FF	BOTTOM	TOP
7	0	1	1	1	Fast PWM, 10-bit	0x03FF	BOTTOM	TOP
8	1	0	0	0	PWM, Phase and Frequency Correct	ICR1	BOTTOM	BOTTOM
9	1	0	0	1	PWM, Phase and Frequency Correct	OCR1A	BOTTOM	BOTTOM
10	1	0	1	0	PWM, Phase Correct	ICR1	TOP	BOTTOM
11	1	0	1	1	PWM, Phase Correct	OCR1A	TOP	BOTTOM
12	1	1	0	0	CTC	ICR1	Immediate	MAX

Mode	WGM13	WGM12 (CTC1)	WGM11 (PWM11)	WGM10 (PWM10)	Timer/Counter Mode of Operation	TOP	Update of OCR1x at	TOV1 Flag Set on
13	1	1	0	1	(Reserved)	–	–	–
14	1	1	1	0	Fast PWM	ICR1	BOTTOM	TOP
15	1	1	1	1	Fast PWM	OCR1A	BOTTOM	TOP

Este registo é configurado da seguinte forma:

$$TCCR1A = 01000011$$

TCCR1B (Time/Counter 1 Control Register B):



Os bits WGM13 e WGM12 são consequência da tabela 15.

Ao colocar o bit ICNC1 a 1 é ativado o *Input Capture Noise Canceler*, ou seja, o sinal de entrada no pino *Input Capture (ICPC1)* é filtrado.

No bit ICES1 (*Input Capture Edge Select*) é selecionado qual o limite no IPC1 que é utilizado para acionar um evento. Quando o ICES1 é escrito com o valor 0, é usado o limite negativo e para o caso de ser escrito o valor 1 é utilizado o limite positivo.

O bit CS1X seleciona a fonte de *clock* a ser usada pelo *Timer*. Na tabela seguinte escolhemos a configuração relativa ao valor de *prescaler* escolhido, sendo neste caso o valor 1.

Tabela 16 Descrição de bit para escolha de *Clock Timer1*

CS12	CS11	CS10	Description
0	0	0	No clock source (Timer/Counter stopped).
0	0	1	$clk_{IO}/1$ (No prescaling)
0	1	0	$clk_{IO}/8$ (From prescaler)
0	1	1	$clk_{IO}/64$ (From prescaler)
1	0	0	$clk_{IO}/256$ (From prescaler)
1	0	1	$clk_{IO}/1024$ (From prescaler)
1	1	0	External clock source on T1 pin. Clock on falling edge.
1	1	1	External clock source on T1 pin. Clock on rising edge.

Este registo é configurado da seguinte forma:

$$TCCR1B = 00011001$$

TCNT1 (Timer/Counter 1)

$$TCNT1H = 0$$

$$TCNT1L = 0$$

OCR1AH (Output Compare Register 1 A):

Este registo é respetivo ao valor TOP selecionado de acordo com a frequência pretendida, ou seja, o estado do bit a ser enviado em determinado momento.

Para obter uma frequência de 1MHz:

$$f_{OCnXPWM} = \frac{f_{CLK_{IO}}}{N \times (1 + TOP)}$$

$$TOP = \frac{8MHz}{1MHz} - 1$$

$$TOP = 7$$

$$OCR1AH = 0$$

$$OCR1AL = 7$$

Para obter uma frequência de 2MHz:

$$f_{OCnxPWM} = \frac{f_{CLK_{IO}}}{N \times (1 + TOP)}$$

$$TOP = \frac{8MHz}{2MHz} - 1$$

$$TOP = 3$$

$$OCR1AH = 0$$

$$OCR1AL = 3$$

TIMSK1 (Timer 1 Interrupt Mask Register):

Este registo só é necessário configurar caso haja interrupções para configurar, neste caso há

$$TIMSK1 = 0$$

O conjunto de configurações iniciais acima para comunicação utilizando PWM, traduzem-se no código em C abaixo:

```
Void config_io()
{
    ...

    DDRB |= (1 << 1);
    TCCR1A |= (1 << COM1A0);
    TCCR1A |= (1 << WGM10);
    TCCR1A |= (1 << WGM11);
    TCCR1B |= (1 << WGM12);
    TCCR1B |= (1 << WGM13);
    TCCR1B |= (1 << CS10);
    OCR1AH = 0;
    OCR1AL = (F_CPU / F_PWM_NIVEL0) - 1;
    ...
}
```

CONFIGURAÇÃO TIMER 0 - INTERRUPÇÃO

Para fazer a comutação de frequência, foi utilizada a interrupção de *Output Compare A*, do *Timer /Counter 0*. Descrita pela função *ISR()*, apontada ao vetor *TIMER0-COMPA* do microcontrolador.

Esta interrupção é acionada a cada 5ms, de forma a verificar o valor lógico a ser enviado através do sinal PWM. Esta configuração segue a lógica apresentada no fluxograma da Figura 48.

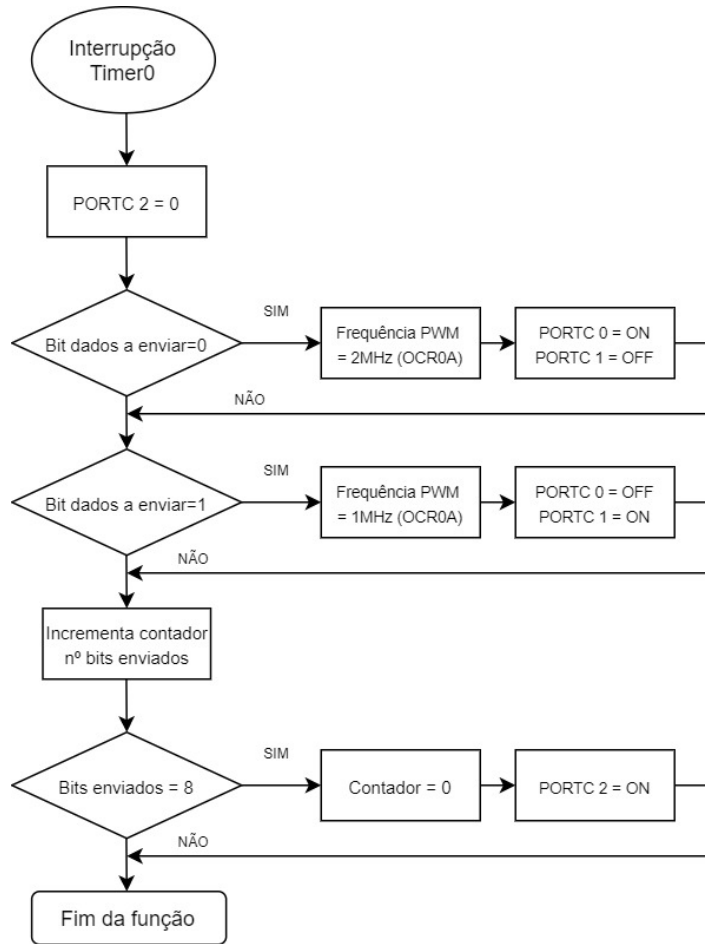


Figura 48 Fluxograma das configurações do Timer 0

Para o timer0 a escrita dos registros foi efetuada da seguinte forma:

TCCR0A (Time/Counter Control Register A):

7	6	5	4	3	2	1	0	
COM0A1	COM0A0	COM0B1	COM0B0	-	-	WGM01	WGM00	TCCR0A
R/W	R/W	R/W	R/W	R	R	R/W	R/W	
0	0	0	0	0	0	0	0	

Como o objetivo é criar uma interrupção, segundo a tabela 17, é escolhida o COM0A1/COMA0/COMB1/COMB0, a operação de porto normal, com o OC0A e OC0B desconectados, ou seja, todos os quatros bits a 0.

Tabela 17 Modo de comparação de saída Timer0

COM0A1	COM0A0	Description
0	0	Normal port operation, OC0A disconnected.
0	1	Toggle OC0A on Compare Match
1	0	Clear OC0A on Compare Match
1	1	Set OC0A on Compare Match

O modo de operação pretendido é o CTC, ou seja, segunda a tabela 18, o modo 2.

Tabela 18 Descrição de bit do modo de geração de onda do Timer0

Mode	WGM02	WGM01	WGM00	Timer/Counter Mode of Operation	TOP	Update of OCRx at	TOV Flag Set on ⁽¹⁾⁽²⁾
0	0	0	0	Normal	0xFF	Immediate	MAX
1	0	0	1	PWM, Phase Correct	0xFF	TOP	BOTTOM
2	0	1	0	CTC	OCRA	Immediate	MAX
3	0	1	1	Fast PWM	0xFF	BOTTOM	MAX
4	1	0	0	Reserved	–	–	–
5	1	0	1	PWM, Phase Correct	OCRA	TOP	BOTTOM
6	1	1	0	Reserved	–	–	–
7	1	1	1	Fast PWM	OCRA	BOTTOM	TOP

Este registo é configurado da seguinte forma:

$$TCCR0A = 00000010$$

TCCR0B (Time/Counter Control Register B):

7	6	5	4	3	2	1	0	
FOC0A	FOC0B	–	–	WGM02	CS02	CS01	CS00	TCCR0B
W	W	R	R	R/W	R/W	R/W	R/W	
0	0	0	0	0	0	0	0	

Na tabela seguinte escolhemos a configuração relativa ao valor de *prescaler* escolhido, sendo neste caso o valor 256.

Tabela 19 Descrição de bit para escolha de *Clock Timer0*

CS02	CS01	CS00	Description
0	0	0	No clock source (Timer/Counter stopped)
0	0	1	clk _{IO} /(No prescaling)
0	1	0	clk _{IO} /8 (From prescaler)
0	1	1	clk _{IO} /64 (From prescaler)
1	0	0	clk _{IO} /256 (From prescaler)
1	0	1	clk _{IO} /1024 (From prescaler)
1	1	0	External clock source on T0 pin. Clock on falling edge.
1	1	1	External clock source on T0 pin. Clock on rising edge.

Este registo é configurado da seguinte forma:

$$TCCR0B = 00000100$$

TIMSK0 (Timer /Counter Interrupt Mask Register):

7	6	5	4	3	2	1	0	
–	–	–	–	–	OCIE0B	OCIE0A	TOIE0	TIMSK0
R	R	R	R	R	R/W	R/W	R/W	
0	0	0	0	0	0	0	0	

É pretendido que exista uma ativação da interrupção quando existe uma comparação entre o TCNT0 e o OCR0A., ou seja, o registo fica configurado da seguinte forma:

$$TIMSK0 = 00000010$$

OCR0A (Output Compare Register A):

A frequência de interrupção pretendida é de 5ms, ou seja, este valor tem de ser convertido para Hz. A escolha do valor OCR0A foi definido através dos seguintes cálculos:

$$f_{OCRxPWM} = \frac{1}{5ms}$$

$$f_{OCRxPWM} = 200 \text{ Hz}$$

$$f_{OCRxPWM} = \frac{f_{CLK_{IO}}}{N \times (1 + TOP)}$$

$$TOP = \frac{8MHz}{256 \times 200} - 1$$

$$TOP = 159$$

$OCR0A = 159$

É possível verificar no código apresentado abaixo que, o PORTB1 é apresentado como saída.

```
Void config_io()
{
    ...
    DDRD |= (1 << 6);
    TCCR0A |= (1 << WGM01);
    TCCR0B |= (1 << CS02);
    TIMSK0 |= (1 << OCIE0A);
    OCR0A = 159;
    sei();
}
```

Para a fazer a comutação da frequência, foi utilizada a função do Timer0, ISR().

O vetor “dados[]”, que armazena a palavra de 8 bits a ser enviada, é avaliado através do apontador de índice i.

Em cada iteração da função o valor de “dados[]” na posição i é comparado com os valores a:

- 0 – Em caso afirmativo, a frequência do sinal PWM é alterado para 2MHz. O pino PC0 é acionado a nível logico 1 e o pino PC1 é acionado a nível logico 0, indicando para efeitos de *debugging*, o envio de um “1” no sinal.
- 1 – A frequência do sinal PWM é alterada para 1MHz. De forma contrária ao caso anterior, PC0 é acionado a nível logico 0 e o pino PC1 é acionado a nível logico 1.

Após a alteração da frequência do sinal o apontador “i” é incrementado, preparando a próxima iteração dos testes.

No final desta preparação, o índice é em si verificado contra o valor 8, indicador de que a palavra foi transmitida na totalidade. Em caso negativo, o microcontrolador, termina a interrupção e regressa ao seu ciclo de funcionamento padrão. Em caso afirmativo, o índice “i” é acionado de forma a indicar o fim da transmissão da palavra de dados, antes do termino da função. Este PC2 é retornado a valor logico zero, sempre que a função de interrupção é iniciada.

```
ISR (TIMER0_COMPA_vect)
{
    PORTC &= ~(1 << PC2);
    if ( dados[i] == 0)
```

```

    {
        OCR1AL = (F_CPU / F_PWM_NIVEL0) - 1;
        PORTC |= (1 << PC0);
        PORTC &= ~(1 << PC1);
    }
    if ( dados[i] == 1)
    {
        OCR1AL = (F_CPU / F_PWM_NIVEL1) - 1;
        PORTC |= (1 << PC1);
        PORTC &= ~(1 << PC0);
    }
    i++;
    if (i == 8 )
    {
        i = 0;
        PORTC |= (1 << PC2);
        //OCR1AL = 0;
        //send_data = 0;
    }
}

```

5. TESTES E RESULTADOS

5.1. TESTE MICROCONTROLADOR – ANALISADOR LOGICO

Para testar o código desenvolvido à saída do controlador, ou seja, testar o sinal PWM a enviar, foi montado um esquema com um programador USBASP ligado ao Atmega e um analisador lógico com ligação USB ao computador como se pode ver na Figura 49.

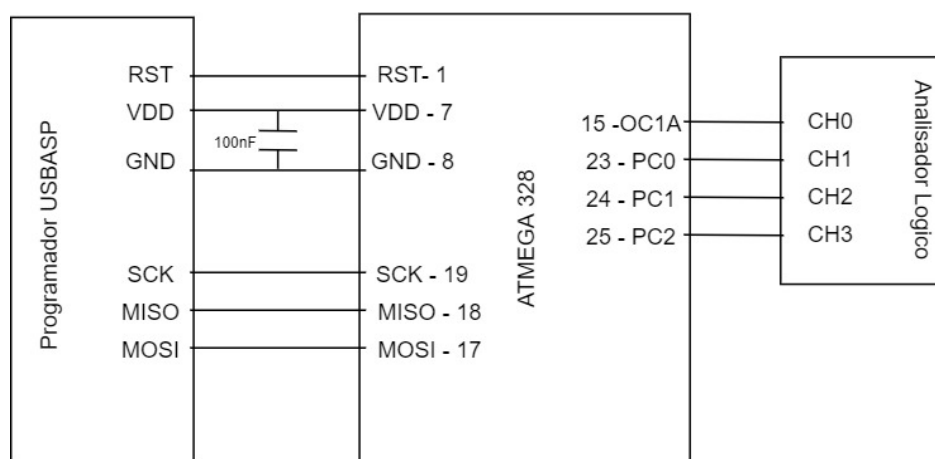


Figura 49 Esquema de ligações para teste do PWM com analisador logico.

Este esquema foi montado numa *breadboard* como se pode ver na Figura 50, na qual o analisador logico utilizado foi o *Hobby Components USB 8CH 24MHz 8 Channel Analyser*, desenhado para funcionar com a plataforma de desenvolvimento de analise de sinal *open-source* da *sigrok PulseView*.

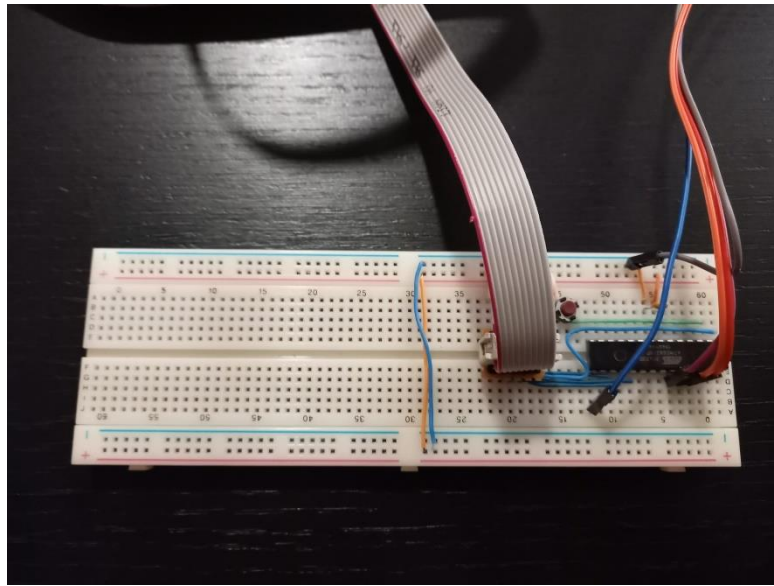


Figura 50 Circuito com analisador logico

No *PulseView*, fez-se a recolha da leitura feita pelo analisador logico do sinal PWM enviado.

Como já referido no capítulo anterior, a mensagem enviada é o número de identificação 231 em binário, daí a leitura do sinal tem de indicar o valor 11100111, sendo este valor um exemplo.

Para fazer uma correta leitura do sinal, é preciso compreender a interface do *PulseView*. Para facilitar a análise gráfica, foi acrescentado mais três sinais, funcionando como “marcadores”. Na Figura 51 pode-se ver o sinal PWM que está a ser transmitido, um sinal que marca quando o sinal PWM está na frequência referente ao estado 0, denominado “ESTADO 0”, o sinal seguinte referente ao estado 1 e por fim o sinal denominado “ULTIMO BIT” referente ao final da palavra, de 8 bits. Cada bit tem um período de 5ms, valor mínimo requerido pelo AFE031.

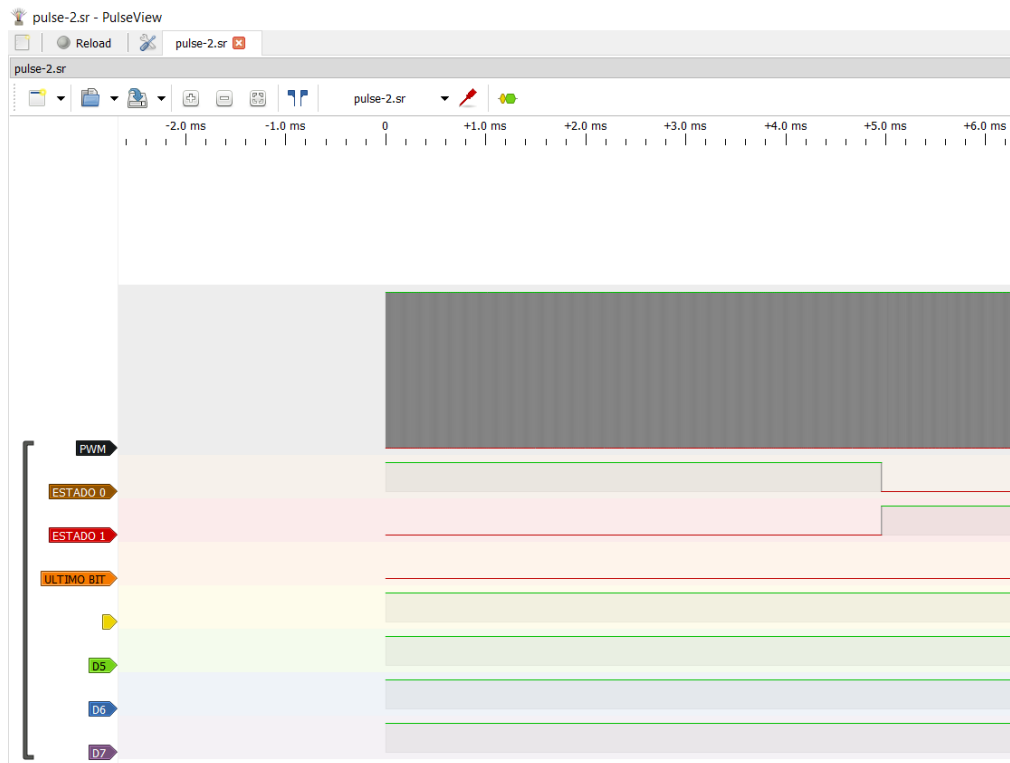


Figura 51 Período de 1 bit

Na Figura 52, é possível analisar uma palavra completa a ser transmitida. Devido à alta frequência de transmissão, a análise fica mais fácil de visualizar com a utilização dos “marcadores” já referidos anteriormente. O início da palavra começa quando o sinal “ULTIMO BIT” muda para o estado lógico 0. A partir daí lê-se 15ms onde o “ESTADO 1” está a 1, ou seja, dividindo 15 por 5 é igual a 3, portanto, durante 3 bits foi enviado o valor 1. Seguindo a mesma lógica, durante os próximos 10ms o “ESTADO 0” está a 1, ou seja, foram enviados dois 0. E por fim foi enviado três 1, completando a mensagem, onde o “ULTIMO BIT” fica a 1, mantendo-se nesse estado durante 5ms, regressando assim ao estado 0, indicando o fim da palavra e o início da seguinte. O envio da palavra é contínuo estando constantemente a enviar a mesma mensagem, sendo que fica comprovado que a mensagem enviada é 11100111, ou seja, a número de identificação 231 em binário.

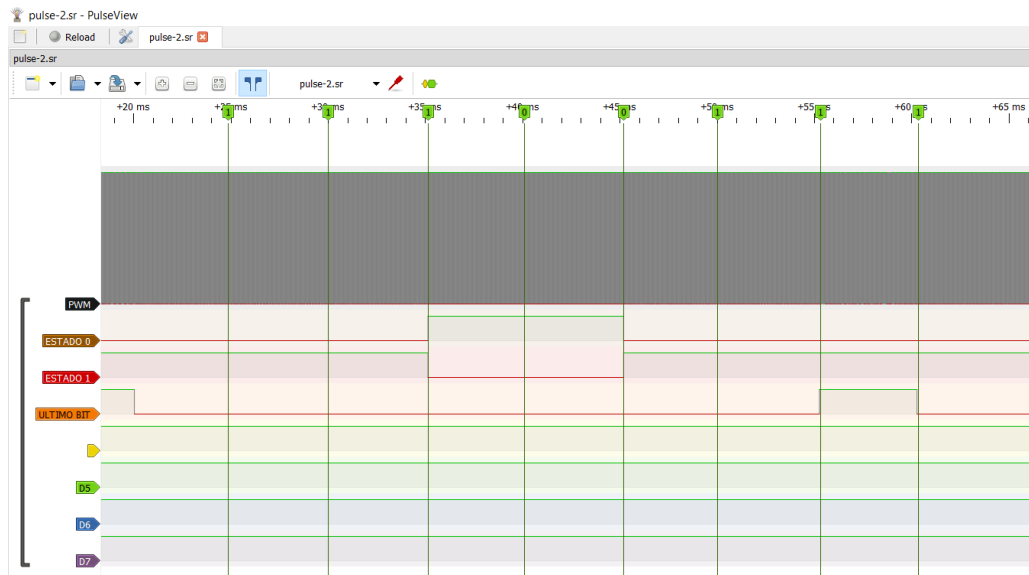


Figura 52 Mensagem enviada por PWM-Leitura com “marcadores”

Para uma análise sem “marcadores” é possível ampliar o gráfico e perceber que na alteração de estado existe uma alteração de frequência. Na Figura 53, vemos a transição do estado 1 para o estado 0, ou seja, uma frequência mais baixa lida como estado 1, para uma frequência mais alta lida como o estado 0.

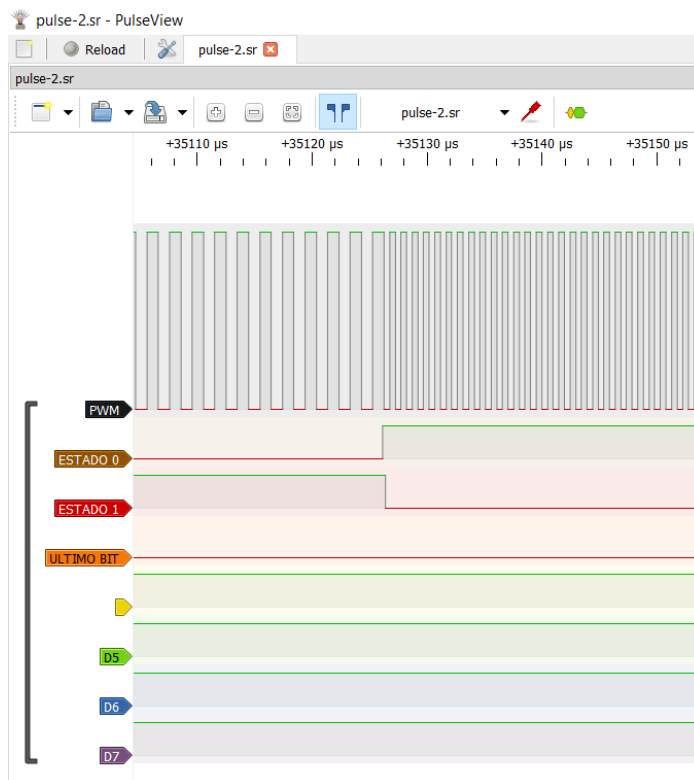


Figura 53 Mensagem enviada por PWM – Leitura do sinal PWM

5.2. TESTE AFE031

Para os testes práticos do protótipo, foi necessário a utilização de uma fonte de alimentação e um osciloscópio para análise do sinal transmitido.

Infelizmente, houve um erro na soldadura do integrado AFE031 na placa PCB, que obrigou a dessoldar o mesmo e a voltar a soldar. No entanto durante o processo de dessoldadura com a tentativa à força de retirada do componente da placa, o AFE031 estragou-se.

Apesar de tudo havia de reserva outro integrado, no entanto com o tempo disponível não foi possível voltar a soldar o AFE031, pois é um processo muito demorado devido ao facto do componente ter os pinos muito pequenos e consequentemente pistas muito pequenas.

Com o tempo restante e para poder realizar pelo menos um teste com o AFE031 e para perceber se o código desenvolvido funciona, fiz uma análise de todos os pinos estritamente necessários, ou seja, foi retirado todos os componentes existentes para proteção e redução de ruído. A única possibilidade de teste foi ligar a alimentação e a comunicação diretamente aos pinos do integrado, não testando assim o protótipo em si, mas sim só o integrado.

Devido a limitações de acesso a equipamento, só havia uma fonte de alimentação, foi então necessário improvisar e utilizar a mesma alimentação para o Atmega 328P e para o AFE031, sendo este valor 5V. Este valor foi alimentado da fonte de alimentação para o Atmega no pino VCC e no AFE031 no pino do VDD, onde as ligações já estão documentadas no capítulo 4.

Para as comunicações foram ligados os quatro pinos da comunicação SPI (SCK, MOSI, MISO, SS), o pino DAC do AFE031 foi ligado à massa, pois nas configurações o mesmo tem de ser levado a estado logico 0 para não definir o modo de transmissão DAC, o pino OC1A ligou ao TX_F_IN para a transmissão do sinal PWM e por fim, a saída do AFE031 no pino PA_OUT1 para o osciloscópio com uma ponta de prova. O esquema de ligações pode ser visto na figura 54 e a montagem deste circuito pode ser vista na Figura 55.

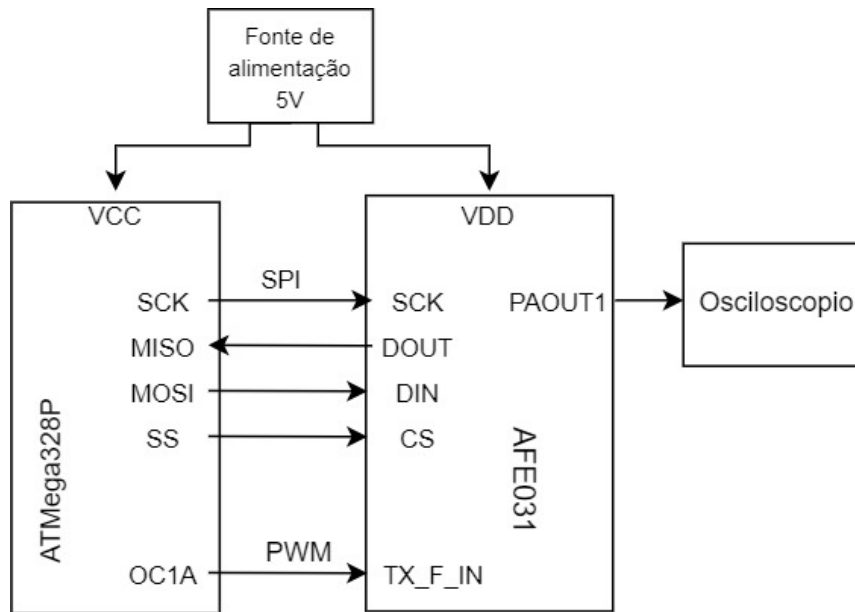


Figura 54 Esquema de ligações do teste do AFE031

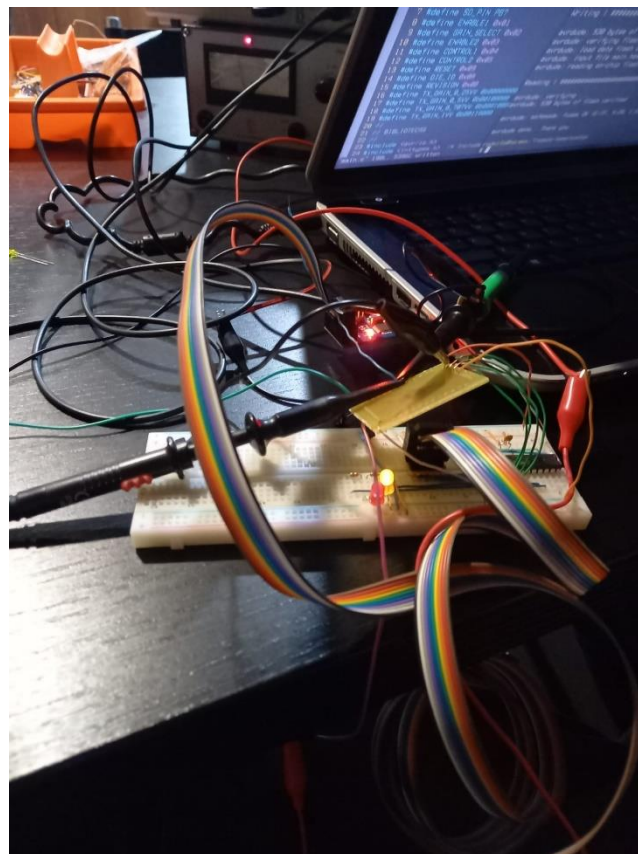


Figura 55 Montagem do circuito de teste do AFE031

O resultado obtido no osciloscópio não foi totalmente claro, como era expectável neste teste. Na leitura da Figura 56 verifica-se um sinal próximo de um PWM com uma frequência com uma leitura de medida flutuante, mas próxima do 1 MHz, que era o valor de frequência esperado respetivo ao valor logico 1. Este sinal como era expectável tem muito ruído, devido á falta de componentes redutores de ruído, o que torna a análise difícil.

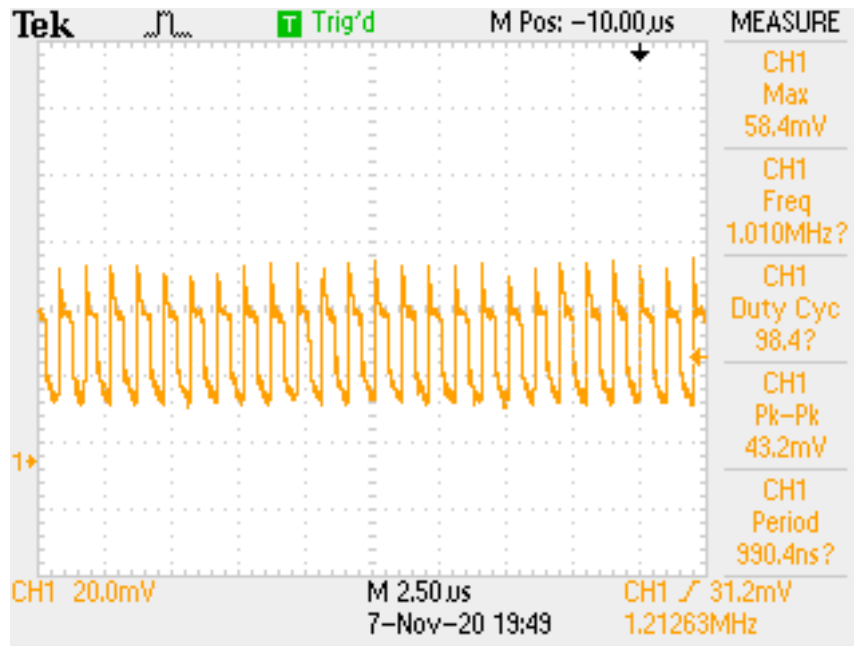


Figura 56 Leitura do sinal à saída do AFE031

No entanto para perceber se a onda enviada é uma mensagem, é preciso encontrar uma alteração de frequência, que se pode verificar na Figura 57. Devido ao ruído, é muito difícil perceber se o sinal enviado é o número de identificação 231 em binário. Este valor é em enviado de forma cíclica de 40ms em 40ms, sendo uma palavra de 8 bits, cada um com 5ms.

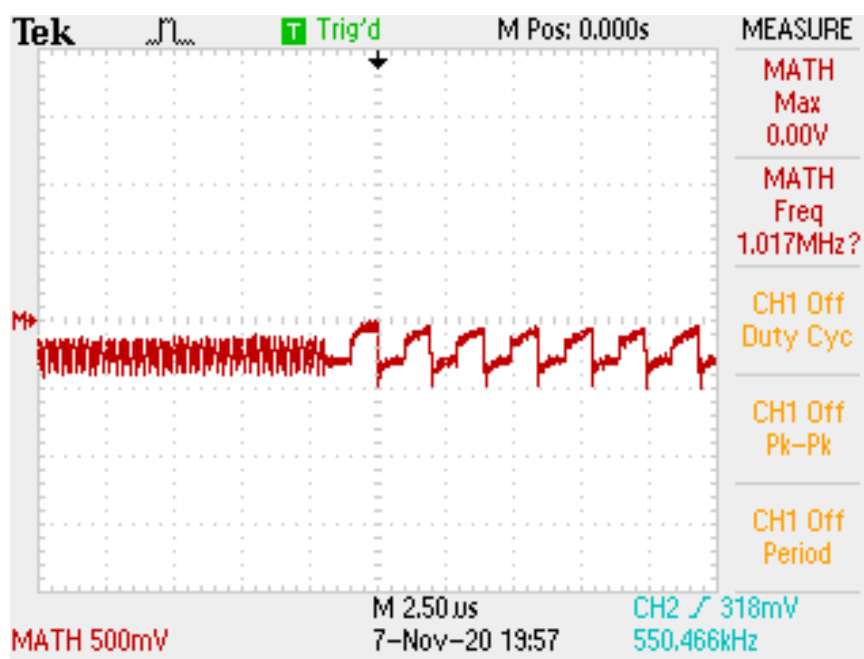


Figura 57 Leitura do sinal à saída do AFE031

Apesar de, para o objetivo do teste realizado ser cumprido na sua totalidade, seria necessário testar o código desenvolvido no protótipo. No entanto havendo alteração de frequência, e o código PWM já validado anteriormente no analisador lógico, juntando os dois testes, é possível afirmar que o objetivo do teste foi cumprido, de enviar um sinal de identificação de uma bateria.

Relativamente à amplitude do sinal não é possível fazer qualquer tipo de validação pois como o sinal tem muito ruído e o osciloscópio não consegue obter uma leitura válida, sendo um valor em constante oscilação. À saída do Atmega do sinal PWM tem 5V de amplitude e tendo em conta que o AFE031 foi configurado para aplicar uma atenuação de 0.25 V/V, o valor expectável de saída do sinal seria aproximadamente 1,25V.

6. CONCLUSÕES

O resultado deste estudo demonstra que é possível criar um sistema de comunicação pela linha de alimentação DC.

Este sistema é capaz de fazer a integração entre o equipamento existente no mercado como o AFE031 e o Atmega 328P, onde é criado um link de comunicação entre os dois dispositivos. O sistema dimensionado tem a capacidade de enviar um sinal pela linha de alimentação DC, sendo esse sinal um possível valor de identificação.

Como conclusão do estudo provou-se que este sistema pode ser implementado. O protótipo dimensionado tal como o código adjacente tem a capacidade para comunicar por uma linha de alimentação DC de baterias de veículos em ambiente industrial, enviando um valor como identificação de uma bateria.

Durante a implementação prática relativamente à validação do sinal enviado os resultados foram muito bons, e de acordo com o pretendido. O sinal PWM enviado foi validado como a mensagem definida no código, ou seja, o código desenvolvido para o sinal a transmitir foi também validado. Relativamente ao teste do protótipo, tal não foi possível, no entanto como alternativa foi realizado um teste diretamente no integrado, onde é possível analisar um sinal no osciloscópio com características idênticas ao esperado, validando apenas a frequência e a comutação da mesma. Mas com o nível de ruído observado, não é possível fazer uma análise quantitativa do sinal, nem perceber a palavra que está a ser transmitida. Com a junção dos dois testes realizados, o objetivo pratico desta dissertação fica cumprido.

Este estudo incidiu unicamente sobre a transmissão, pelo que seria interessante uma adição ao estudo um protótipo utilizado como recetor. Mais também poder criar uma rede de dispositivos a comunicar entre si pela linha de alimentação.

Como trabalho futuro uma validação prática do AFE031 com o protótipo desenvolvido, tal como a respetiva validação de *hardware* e de *firmware*.

Referências Documentais

- [1] R. d. S. Augusto, “Simulação de comunicações baseadas em Power Line Communication (PLC) para Smart Grids,” 2003.
- [2] D. Ellis, “About Colored Noise,” [Online]. Available: <https://www.ee.columbia.edu/~dpwe/noise/>. [Acedido em 02/11/2020 Novembro 2020].
- [3] J. Hannigan, “Effects of Electric and Magnetic Fields on Transmission Line Design,” 2013.
- [4] S. C. Pereira, I. R. S. Casella e A. Caporali, “Power Line Communication Technology in Industrial Networks,” 2015.
- [5] P. Barry e P. Crowley, “Serial Peripheral Interface,” 2012.
- [6] A. V. Rakesh Rajbanshi, “OFDM systems Design,” 2004.
- [7] Geeks for Geeks, “Difference between FDMA, TDMA and CDMA,” 24 Agosto 2020. [Online]. Available: <https://www.geeksforgeeks.org/difference-between-fdma-tdma-and-cdma/>. [Acedido em 20 Janeiro 2020].
- [8] S. Santos, *Electrónica das Telecomunicações*, 2018.
- [9] Texas Instruments, *Interfacing the C2000 With and AFE030/1: FSK Example*, 2020.
- [10] J. G. Proakis e M. Salehi, *Communication Systems Engineering*, 2002.
- [11] Texas Instruments, *Implementing PRIME for Robust and Reliable Power Line Communication (PLC)*, 2013.
- [12] P. Pinto, “Redes: Modelo de referência OSI,” [Online]. Available: <https://medium.com/@pedrompinto/redes-modelo-de-refer%C3%Aancia-osi-827cbd67e882>.
- [13] G3-PLC Alliance, “G3-PLC Overview,” [Online]. Available: <https://g3-plc.com/what-is-g3-plc/g3-plc-overview/>.
- [14] ERDF, *PLC G3 Physical Layer Specification*.
- [15] *G3 Specifications metering profile*, 2011.
- [16] M. Hoch, “Comparison of PLC G3 and PRIME,” 2011.
- [17] Texas Instruments, *BOOSTXL-AFE031-DF1 User's Guide*, 2019.
- [18] Texas Instruments, *Datasheet*, 2019.

[19] Eletronic Wings, “SPI in AVR Atmega16/Atmega32,” [Online]. Available: <https://www.electronicwings.com/avr-atmega/atmega1632-spi>. [Acedido em 02 Setembro 2020].

Anexo B. – Código desenvolvido

```
// DEFS
#define F_CPU 8000000
#define F_PWM_NIVEL0 2000000
#define F_PWM_NIVEL1 1000000
#define DUTY 50
#define DAC_DDR DDRB
#define DAC_PORT PORTB
#define DAC_PIN PB6
#define SD_DDR DDRB
#define SD_PORT PORTB
#define SD_PIN PB7
#define ENABLE1 0x01
#define GAIN_SELECT 0x02
#define ENABLE2 0x03
#define CONTROL1 0x04
#define CONTROL2 0x05
#define RESET 0x09
#define DIE_ID 0x0A
#define REVISION 0x0B
#define TX_GAIN_0_25VV 0b00000000
#define TX_GAIN_0_5VV 0b00100000
#define TX_GAIN_0_707VV 0b00010000
#define TX_GAIN_1VV 0b00110000
//
// BIBLIOTECAS
//
#include <avr/io.h>
#include <inttypes.h> /* Include integer type header file */
//
// BIBLIOTECAS
//
#include <avr/io.h>
#include <stdio.h>
#include <string.h>
#include <avr/interrupt.h>
#include <util/delay.h>
#include <inttypes.h> /* Include integer type header file */
//
// BIBLIOTECAS LOCAIS
//
// DECLARACAO FUNCOES
//
void config_io(void);
void SPI_enviar(uint8_t endereco, uint8_t dados);
void config_afe031();
//
```

```

// DECLARACAO VARIAVEIS GLOBAIS
//
char dados[8]={1,1,1,0,0,1,1,1};
uint8_t i, send_data;
//
// MAIN
//
int main(void)
{
    config_io();
    config_afe031();
    TCNT0 = 0; // Reset Timer/Counter0 - reset temporizador frame
5ms
    TCNT1H = 0;      // Reset Timer/Counter1 - reset a onda pwm
    TCNT1L = 0;
    while(1)
    {
        send_data = 1;
        //if( (PIND & (1 << 0)) == 1)
        //{
        //    while((PIND & (1 << PD0)) == 1) // aguarda que o
sw seja liberto
        //    {
        //    }
        //    send_data = 1;
        //}
    }
}
void config_io()
{
    //
    // CONFIG IO
    //
    DAC_DDR |= (1 << DAC_PIN); // Define pino controlo DAC como
saida
    SD_DDR |= (1 << SD_PIN);    // Define pino controlo SD como
saida
    //
    DDRC |= (1 << PC1); // Define pino indicacao transmissao 1
logico como saida
    DDRC |= (1 << PC0); // Define pino indicacao transmissao 0
logico como saida
    DDRC |= (1 << PC2); // Pino utilizado para indicacao de
final de palavra 8 bits configurado como saida
    //
    // CONFIG SPI
    //
    // Configuracao SPI MASTER
    // Modo 0 -> CPOL = 0; CPHA = 0;
    // Envia primeiro MSB
    // SCLK <= 20 MHz -> Fmicro/2 = 4 MHz
    //
    // SS = PORTB2
    // MOSI = PORTB3
    // MISO = PORTB4
    // SCK = PORTB5

```

```

//
DDRB |= (1 << PB2) | (1 << PB3) | (1 << PB5); // SS, MOSI
e SCK como saidas
DDRB &= ~(1 << PB4); // MISO como entrada
PORTB |= (1 << PB1); // SS a 1 - sem transmissao
SPCR = (1 << SPE) | (1 << MSTR);
SPSR = (1 << SPI2X);
//
// CONFIG PWM
//
// Frequencia micro = 8 MHz
// TIMER/COUNTER1 16bits Fast PWM pino OC1A
// TOP = OCR1A
// Duty Cycle a 50% (modo WGM13:0)
// F = Fmicro/( N*(1 + TOP))
// Prescaler -> N = 1
//
DDRB |= (1 << 1); // PORTB1 como saida - PORTB1 = OC1A
TCCR1A |= (1 << COM1A0); // Toogle da pino PB1, quando
existe compare match entre TCNT1 e OCR1A
TCCR1A |= (1 << WGM10); // Modo FAST PWM
TCCR1A |= (1 << WGM11); // com TOP = OCR1
TCCR1B |= (1 << WGM12); // duty 50%
TCCR1B |= (1 << WGM13);
TCCR1B |= (1 << CS10); // Prescaler = 1
OCR1AH = 0;
OCR1AL = (F_CPU / F_PWM_NIVEL0) - 1; // Atribuicao de valor
padrao de frequencia
//
// CONFIG TEMPORIZACAO 5ms
//
// Frequencia micro = 8 MHz
// TIMER/COUNTER0 8bits CTC
// OCR0A valor de compare
// F = Fmicro/( 2*N*(1 + OCR0A))
// Prescaler -> N = 1
//
//
DDRD |= (1 << 6); // PORTB1 como saida - PORTB1 = OC1A
TCCR0A |= (1 << WGM01); // Modo CTC
TCCR0B |= (1 << CS02); // Prescaler = 256
TIMSK0 |= (1 << OCIE0A); //Ativacao da interrupcao quando
existe compare entre TCNT0 e OCR0A
OCR0A = 159;
sei();
}
void SPI_enviar(uint8_t endereco, uint8_t dados)
{
PORTB &= ~(1 << PB2); // Ativa Slave, indica inicio de
transmissao
SPDR = ( (0 << 6) + (endereco)); // Bit 6 = 1, escrever +
endereco
while(!(SPSR & (1<<SPIF))); // Aguarda transmissao dos 8 bits
SPDR = dados; // Envia os dados pretendidos
while(!(SPSR & (1<<SPIF)));
}

```

```

        PORTB |= (1 << PB2); // Indica ao Slave que
a transmissao terminou
    }
void config_afe031()
{
    DAC_PORT &= ~(1 << DAC_PIN); // Pino DAC = 0;
    SD_PORT &= ~(1 << SD_PIN); // Pino SD = 0;
    SPI_enviar(RESET, 0x14); // Soft reset do AFE
    // Ativar Bias AFE - REF1 on, REF2 on
    // Ativar Modulo ZC
    // PA_OUT = 1, baixa impedancia ???
    SPI_enviar(ENABLE2, 0b00001111);
    SPI_enviar(CONTROL1, 0x00); // CENELEC A, calibracao
desligada
    SPI_enviar(CONTROL2, 0x00); // Limpa interrupts do AFE
    SPI_enviar(GAIN_SELECT, TX_GAIN_0_25VV); // Define Ganho
TX
    SPI_enviar(ENABLE1, 0b00000011); // Ativa PA e TX, desativa
DAC
}
ISR (TIMER0_COMPA_vect)
{
    PORTC &= ~(1 << PC2);

    if ( dados[i] == 0)
    {
        OCR1AL = (F_CPU / F_PWM_NIVEL0) - 1;
        PORTC |= (1 << PC0); // Sinaliza transmissao de
valor logico 0 na onda PWM
        PORTC &= ~(1 << PC1);
    }
    if ( dados[i] == 1)
    {
        OCR1AL = (F_CPU / F_PWM_NIVEL1) - 1;
        PORTC |= (1 << PC1); // Sinaliza transmissao de
valor logico 1 na onda PWM
        PORTC &= ~(1 << PC0);
    }
    i++;
    if (i == 8 )
    {
        i = 0;
        PORTC |= (1 << PC2); // Sinaliza final de palavra
de 8 bits
    }
}

```