

**UNICEUB – CENTRO UNIVERSITÁRIO DE BRASÍLIA**  
**FATECS - FACULDADE DE TECNOLOGIA**  
**E CIÊNCIAS SOCIAIS APLICADAS**  
**CURSO DE ENGENHARIA DE COMPUTAÇÃO**

**JURACI GONTIJO COELHO**

**ACIONAMENTO REMOTO DE LÂMPADAS**

BRASÍLIA/DF

1º SEMESTRE DE 2008

**JURACI GONTIJO COELHO**

**ACIONAMENTO REMOTO DE LÂMPADAS**

Monografia apresentada ao Curso de Engenharia de Computação, como requisito parcial para obtenção do grau de Engenheiro de Computação.

Orientador: Prof. José Julimá Bezerra Junior

BRASÍLIA/DF  
1º SEMESTRE DE 2008

## Resumo

A motivação para a realização deste projeto surgiu a partir da observação do crescente índice de seqüestros relâmpagos divulgados no noticiário cotidiano. Com o intuito de diminuir este índice foi proposto o desenvolvimento de um dispositivo que permita a uma pessoa o acionamento, de dentro de seu próprio carro, de uma lâmpada, com a finalidade de observar as condições de segurança nas imediações de sua residência ou local de trabalho e, caso necessário, também, o acionamento de uma sirene. O alcance proposto para este acionamento é de 100 metros. O projeto é composto por um controle remoto, um dispositivo de controle e um dispositivo acionador. O controle remoto é composto por um microcontrolador, botões e um módulo de rádio. O microcontrolador é responsável por fazer a leitura da situação dos botões e, quando necessário, enviar comandos e informações ao módulo de rádio para que sejam transmitidos ao dispositivo de controle. O dispositivo de controle é composto por um microcontrolador e um módulo de rádio. Ele é responsável por receber as informações transmitidas pelo controle remoto, interpreta-las e comandar o dispositivo acionador. O dispositivo acionador é composto por dois relés e componentes auxiliares e é responsável por receber os comandos do dispositivo de controle e acender a lâmpada ou ligar a sirene.

Palavras chave: Transceptor, PIC16F628A, acionamento.

## Abstract

This project was motivated by the increasing rate of kidnappings disclosed in the daily news. In order to reduce this index, it has been proposed the development of a device that allows a person to trigger, from inside of your own car, a lamp. So that the conditions of security in his neighborhood or work place can be observed. And in case of need it allows also to trigger a siren. The proposed scope for this trigger is 100 meters. The project consists of a remote control, a control module and a trigger module. The remote control is composed of a microcontroller, buttons and a radio module. The microcontroller is responsible for reading the buttons' state and send commands and information to the radio module to be transmitted to the control module, which is composed of a microcontroller and a radio module. The control module is responsible for receiving the information transmitted by remote control, interpret them and command the trigger module. The trigger module is composed of two relays and auxiliary components and is responsible for receiving the commands from the control module and light the lamp or turn on the siren.

Keywords: Transceiver, PIC16F628A, trigger.

## **Agradecimentos**

Agradeço aos meus pais José Augusto e Vitória pelo afeto e carinho.

Ao Professor orientador José Julimá pelo incentivo, orientação e cobrança.

Ao meu amigo Daniel pelo apoio e idéias.

A minha esposa Cida pela paciência, compreensão, dedicação e apoio.

## Sumário

Capítulo 1. Introdução .....	1
1.1 Motivação .....	1
1.2 Objetivo .....	1
1.3 Visão geral do projeto.....	2
Capítulo 2. Referencial Teórico .....	4
2.1 Comunicação via rádio .....	4
2.2 Microcontrolador PIC16F628A .....	7
2.3 Linguagem de programação.....	14
Capítulo 3. Desenvolvimento do projeto.....	17
3.1 Hardware .....	17
3.1.1 Controle remoto.....	18
3.1.2 Dispositivo de controle .....	22
3.1.3 Dispositivo acionador .....	23
3.2 Firmware.....	25
3.3 Testes e resultados .....	43
3.4 Simulação.....	53
Capítulo 4. Conclusão .....	57
Referências Bibliográficas .....	58
Apêndice A – Código fonte do controle remoto .....	59
Apêndice B – Código fonte do dispositivo de controle .....	67
Anexo – Datasheet do transceptor .....	75

## Índice de figuras

Figura 1.1 - Diagrama geral do projeto .....	2
Figura 2.1 - Chip do PIC16F628A.....	9
Figura 2.2 - Diagrama de blocos do PIC16F628A .....	11
Figura 2.3 - Mapa da memória de programa e da pilha .....	12
Figura 2.4 - Mapa da memória RAM.....	13
Figura 3.1 - Diagrama em blocos do Projeto .....	17
Figura 3.2 - Circuito oscilador .....	18
Figura 3.3 - Esquema elétrico dos botões e dos leds de <i>status</i> .....	19
Figura 3.4 - Esquema de interligação do transceptor com o PIC.....	20
Figura 3.5 - Pinagem do Transceptor .....	20
Figura 3.6 - Esquema elétrico do controle remoto .....	22
Figura 3.7 - Esquema elétrico do dispositivo de controle .....	23
Figura 3.8 - Dispositivo acionador .....	24
Figura 3.9 - Dispositivo de controle e acionador.....	25
Figura 3.10 – Fluxograma geral.....	26
Figura 3.12 – Trecho do código da leitura dos botões.....	28
Figura 3.13 – Ação do botão 1.....	29
Figura 3.14 – Trecho do código da ação do botão 1 .....	29
Figura 3.15 – Transmissão de dados .....	30
Figura 3.16 – Trecho do código da rotina de transmissão de dados .....	31
Figura 3.17 – Recepção de dados.....	32
Figura 3.18 –Trecho do código da rotina de recepção de dados.....	32
Figura 3.19 – Acionamento do dispositivo de controle .....	33
Figura 3.20 - Trecho do código da rotina de acionamento .....	34
Figura 3.21 – Checagem do relé 1 .....	35
Figura 3.22 – Trecho do código da sub-rotina de checagem do relé 1 .....	35
Figura 3.23 – Relé 1 acionado.....	36
Figura 3.24 – Trecho do código da sub-rotina relé 1 acionado.....	36
Figura 3.25 – Relé 1 liberado.....	37
Figura 3.26 - Trecho do código da sub-rotina relé 1 liberado .....	37
Figura 3.27 - Ambiente de trabalho do MPLAB .....	38
Figura 3.28 - Definição de nome para o Workspace.....	39

Figura 3.29 - Criação do projeto .....	39
Figura 3.30 – Tela do Workspace com o Projeto Aberto .....	40
Figura 3.31 - Arquivo de código-fonte aberto.....	40
Figura 3.32 - Selecionando o PIC desejado .....	41
Figura 3.33 - Relatório de compilação em andamento .....	42
Figura 3.34 - Relatório de compilação com sucesso .....	42
Figura 3.35 - Gravador de PIC McPlus .....	43
Figura 3.36 - Ambiente de realização dos testes.....	44
Figura 3.37 - Ambiente de realização dos testes.....	44
Figura 3.38 - Portadora ligada .....	45
Figura 3.39 - Portadora desligada .....	45
Figura 3.40 - Portadora com espúrios .....	46
Figura 3.41 - Portadora sem espúrios .....	46
Figura 3.42 - Portadora modulada .....	47
Figura 3.43 – Sinal de acionamento .....	48
Figura 3.44 – Acionamento do botão.....	48
Figura 3.45 – Sinal de clock.....	49
Figura 3.46 – Sinal interferente.....	50
Figura 3.47 – Preâmbulo e sinal interferente .....	50
Figura 3.48 – Controle remoto com chaves seletoras .....	51
Figura 3.49 – Dispositivo de controle com chaves seletoras .....	52
Figura 3.50 – Controle remoto .....	52
Figura 3.51 – Dispositivo de controle e dispositivo acionador .....	53
Figura 3.52 – Recepção a cinco metros do controle remoto.....	54
Figura 3.53 – Recepção a 100m do controle remoto.....	54
Figura 3.54 – Recepção a 300m do controle remoto.....	55
Figura 3.55 – Recepção a 400m do controle remoto.....	56

**Índice de equações**

Equação 3.1 - Cálculo de taxa de transferência.....	21
---	----

## Lista de símbolos e abreviaturas

### Abreviaturas

ACLPD – Acionamento da Lâmpada

ACSNE – Acionamento da Sirene

AM – Modulação em Amplitude

bps – Bits por segundo

BT1 – Botão 1

BT2 – Botão 2

CI – Circuito Integrado

EEPROM – Memória Apenas de Leitura Programável e Apagável Eletricamente

EHF – Frequência Extremamente Alta

ELF – Frequência Extremamente Baixa

FM – Modulação em Frequência

GPS – Sistema Global de Posicionamento

HF – Frequência Alta

LCD – Display de Cristal Líquido

LED – Diodo Emissor de Luz

LF – Frequência Baixa

MF – Frequência Média

RAM – Memória de Acesso Randômico

RF – Radiofrequência

RL1 – Relé 1

RL2 – Relé 2

RTLDP – Retroalimentação da Lâmpada

RTSNE – Retroalimentação da sirene

SHF – Frequência Super Alta

UHF – Frequência Ultra Alta

ULA – Unidade Lógica e Aritmética

USART – Interface Universal de Transmissão e Recepção Síncrona e Assíncrona

VHF – Frequência muito Alta

VLF – Frequência Muito Baixa

## Símbolos

Hz – Hertz, unidade de frequência

V – Volts, unidade de tensão elétrica

A – Ampère, unidade de corrente elétrica

$\Omega$  - Ohm, unidade de resistência elétrica

dBm – Decibel relativo a 1mW, relação de potência

W – Watt, unidade de potência

## **Capítulo 1. Introdução**

Este projeto consiste num sistema que permite o acionamento de lâmpadas a uma distância de 100m, por meio de um controle remoto, o qual pode também acionar circuitos elétricos através de contatos secos de relés. O sistema é composto basicamente por um controle remoto, um dispositivo de controle e um dispositivo acionador.

### **1.1 Motivação**

A motivação para a realização deste projeto surgiu a partir da observação do crescente índice de seqüestros relâmpagos divulgados no noticiário cotidiano. Então, com o intuito de diminuir este índice foi pensado em um dispositivo que permita a uma pessoa o acionamento, de dentro de seu próprio carro, de uma lâmpada, com a finalidade de observar as condições de segurança nas imediações de sua residência ou local de trabalho e, caso necessário, também, o acionamento de uma sirene.

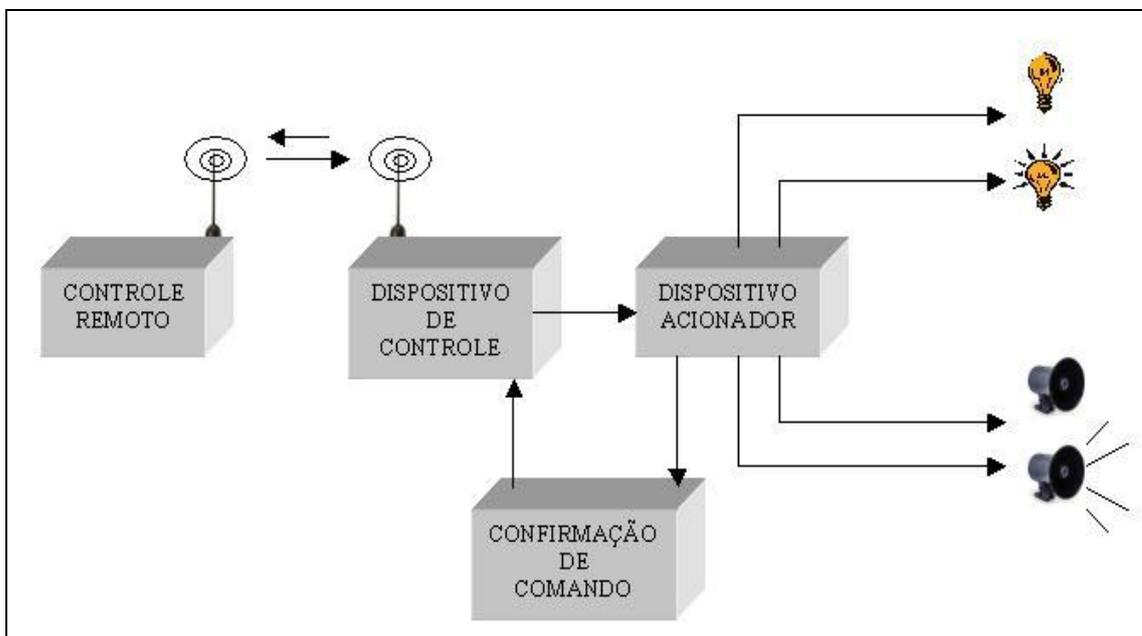
### **1.2 Objetivo**

A utilidade deste projeto é proporcionar uma melhoria para a sociedade através do desenvolvimento de um dispositivo que permita o acionamento remoto de lâmpadas e/ou circuitos elétricos. A melhoria ocorre devido à segurança e praticidade proporcionadas pelo acionamento remoto de lâmpadas permitindo, desta forma, que a exposição às situações de risco de seqüestros relâmpagos seja minimizada. Isto é possível graças à mobilidade proporcionada pelo uso das tecnologias de transmissão sem fio as quais têm proporcionado ao homem enormes possibilidades de acesso aos mais variados recursos tecnológicos.

Assim sendo, este projeto tem por finalidade específica disponibilizar um recurso que permita às pessoas evitar que se exponham a situações de risco. O conforto, a comodidade e o auxílio a pessoas com dificuldades de locomoção também são possíveis com o uso desta tecnologia.

### 1.3 Visão geral do projeto

Este projeto é composto por um controle remoto, um dispositivo de controle e um de acionamento. A transmissão a partir do controle remoto é feita através de radiofrequência. O dispositivo de acionamento é responsável por ligar ou desligar a lâmpada ou a sirene. O diagrama geral do projeto é apresentado na figura 1.1.



**Figura 1.1 - Diagrama geral do projeto**

O acionamento das lâmpadas é realizado com base em um comando enviado pelo usuário, através do controle remoto, ao dispositivo de controle do sistema, o qual o interpreta, e então, envia um sinal de acionamento a um dispositivo acionador que em seguida aciona a lâmpada ou circuito selecionado. Após o envio do sinal de acionamento ao dispositivo acionador é feita a verificação do acionamento. Para isso, uma retro-alimentação envia um sinal ao controle do sistema e este informa o *status* do acionamento ao controle remoto.

O controle remoto e a unidade de controle do sistema são compostos, basicamente, cada um, por um microcontrolador, PIC16F628A, e um transceptor

modelo NIM-434.650-10 da Radiometrix, além de componentes eletrônicos auxiliares. Já o dispositivo acionador é composto por dispositivos eletro-eletrônicos com a finalidade de realizar, a partir de um sinal recebido da unidade de controle do sistema, o acionamento, por meio de um relé, de um contato seco, o qual funciona como uma chave liga/desliga para a lâmpada ou circuito que é telecomandado.

Cada etapa deste projeto é detalhada nos capítulos seguintes da monografia.

O capítulo 2 trata do referencial teórico. Neste capítulo são comentados os tópicos teóricos mais relevantes, como a comunicação via rádio, características do microcontrolador e a linguagem de programação utilizada.

O capítulo 3 refere-se ao desenvolvimento do projeto. São descritas as implementações de hardware e software além dos testes, resultados e simulações.

O capítulo 4 apresenta a conclusão, com os problemas ocorridos e sugestões para projetos futuros.

## Capítulo 2. Referencial Teórico

Neste capítulo é feita uma breve descrição da teoria relacionada a este projeto. O item 2.1 refere-se à comunicação via rádio, o 2.2 ao microcontrolador PIC16F628A e o item 2.3 linguagem de programação utilizada.

### 2.1 Comunicação via rádio

O transceptor escolhido para esse projeto foi o NiM 2-434.650-10 que apresenta as seguintes características, de acordo com o *datasheet* em anexo:

- Modo de operação halfduplex
- Frequência de 434,65MHz (UHF)
- Velocidade de até 10kbps
- Modulação FM
- Alcance de até 500 metros
- Tensão de operação de 2,9V a 15V

A comunicação via rádio consiste em conectar dois ou mais equipamentos fazendo-se o uso da radiofrequência. Por meio de técnicas apropriadas é possível realizar qualquer tipo de comunicação utilizando as ondas de rádio. Graças a essas técnicas é que foi possível o surgimento da rádiodifusão, do celular, do GPS, das comunicações a longas distâncias, por meio de satélites e de sistemas em radiovisibilidade, e tantos outros serviços de telecomunicações.

As principais vantagens em utilizar este tipo de comunicação são a mobilidade, onde se tem a liberdade de conectar-se em qualquer rede existente, a flexibilidade, onde é feito o uso de estações de rádio para conectar usuários em redes existentes, a instalação rápida de uma rede, pois não precisa de cabos nem de infra-estrutura complexa, a modularidade já que a rede é livre para futuras expansões e, dentre várias outras, a facilidade de alcançar pontos de difícil acesso como prédios tombados pelo patrimônio histórico e locais remotos e inóspitos.

Um transmissor de rádio pode ser entendido como um elemento que provoca continuamente, através de uma antena, perturbação eletromagnética, de forma localizada, que se propaga no espaço, em todas as direções, atenuando-se com a

distância. Uma antena receptora pode sentir estas perturbações e, se estiver ligada a um equipamento conveniente (receptor), haverá recepção dos sinais daquele transmissor [4].

Para que ocorra comunicação via rádio, inicialmente, quando o transmissor é ligado este envia para o espaço ondas eletromagnéticas de frequência fixa, fazendo com que um receptor sintonizado nesta frequência, apenas saiba que o transmissor está no ar. No entanto, se for variada uma característica da onda gerada pelo transmissor, na recepção é possível detectar estas variações impressas na onda original [4].

Esta onda original é chamada de portadora ou radiofrequência e serve apenas para estabelecer o contato, através do espaço, entre o transmissor e o receptor. O sinal que representa a informação e que varia uma característica da onda portadora chama-se onda moduladora ou sinal modulante. Ao processo de variação de uma característica da onda portadora de acordo com o sinal elétrico da informação, chama-se modulação [4].

A variação da amplitude da onda portadora constitui o método de modulação em amplitude (AM) e, para a variação da frequência da onda portadora, tem-se a modulação em frequência (FM). Do lado da transmissão, o equipamento que produz a modulação chama-se modulador e normalmente está junto ao transmissor. Do lado da recepção, o equipamento que sente as variações da portadora e recupera a informação é o demodulador, estando normalmente junto ao receptor [4].

Desse modo ao se transmitir uma informação de uma localidade A para uma localidade B, esta informação é processada em A pelo modulador-transmissor, fazendo com que se tenha uma onda portadora modulada na antena transmissora. Esta onda é captada pela antena receptora na localidade B, sendo processada pelo receptor-demodulador, regenerando assim a informação transmitida em A e entregue ao usuário em B [4].

A radiofrequência (onda portadora) utilizada para a transmissão de informação da localidade A para B, constitui o canal de RF (canal de radiofrequência). Como este processo é unidirecional, para se transmitir na direção inversa, isto é, a informação de B para A, é necessário um outro canal de RF [4].

Quando existem obstáculos físicos que atrapalham a propagação das ondas no espaço, ou quando este sinal está demasiadamente enfraquecido devido às características de propagação utilizam-se estações intermediárias ao longo das rotas

de rádio, chamadas estações repetidoras, afim de regenerar ou retransmitir as ondas. O conjunto de estações terminais e repetidoras é chamado de tronco de rádio [4].

As ondas eletromagnéticas propagam-se de maneiras diferentes, dependendo da frequência emitida pelo transmissor. Devido a isto, os sistemas de rádio são classificados internacionalmente de acordo com as faixas de frequências utilizadas [4]. À medida que ocorre aumento da frequência, a propagação das ondas de rádio se aproxima da luz, ou seja, processa-se em linha reta, exceção feita ao fenômeno de refração. Estas ondas podem ser focalizadas por antenas e refletores, difratadas por dispositivos refratores ou obstáculos e ainda refletidas [1].

As ondas de rádio são classificadas internacionalmente como ELF (300Hz – 3KHz), VLF (3KHz – 30KHz), LF (30KHz – 300KHz), MF(300KHz – 3MHz), HF (3MHz – 30MHz), VHF (30MHz – 300MHz), UHF (300MHz – 3GHz), SHF (3GHz – 30GHz) e EHF (30GHz – 300GHz).

#### **Faixa ELF (*Extremely Low Frequency*)**

Faixa de frequência em que as ondas penetram razoavelmente no solo e na água, portanto, possui aplicação em comunicação com submarinos e escavações de minas. Geralmente esta faixa é operada com transmissores de alta potência e grandes antenas [1].

#### **Faixa VLF (*Very Low Frequency*)**

Nesta faixa, o mecanismo de propagação utilizado é a reflexão ionosférica, sendo considerado um ótimo condutor; pois induz pequena atenuação na onda refletida [1].

#### **Faixa LF (*Low Frequency*)**

Para esta faixa, até aproximadamente 100KHz, é empregado o mecanismo de reflexão ionosférica, muito embora a atenuação da onda seja maior que a observada na faixa de VLF [1].

### **Faixa MF (*Médium Frequency*)**

Para frequências acima de 100KHz, dentro da faixa de MF (300KHz a 3MHz), o mecanismo empregado é o das ondas de superfície, que apresenta menor atenuação que o mecanismo de reflexão ionosférica [1].

### **Faixa HF (*High Frequency*)**

Na faixa de HF, o mecanismo que possui maior domínio é o da refração ionosférica, sendo que em regiões mais próximas do transmissor ainda permanece a presença das ondas de superfície. Isto se deve ao fato que na refração ionosférica as diversas camadas da ionosfera vão desviando pouco a pouco a trajetória das ondas antes de retornarem a superfície da Terra [1].

### **Faixas de VHF(*Very High Frequency*), UHF (*Ultra High Frequency*) e SHF (*Super High Frequency*)**

Sistemas de propagação em visibilidade, uma vez que as antenas permitem focalizar as ondas, diminuem a influência do terreno na energia propagada. Utiliza-se também do fenômeno da difração, pois na faixa de VHF já não se torna mais possível o uso da refração ionosférica, uma vez que as ondas não retornam à superfície terrestre pois não atingem ângulo zero ao serem irradiadas [1]

## **2.2 Microcontrolador PIC16F628A**

Em poucas palavras, o microcontrolador pode ser definido como um “pequeno” componente eletrônico, dotado de uma “inteligência” programável utilizado no controle de processos lógicos. [6]

O controle de processos deve ser entendido como o controle de periféricos, tais como: LEDs, botões, displays de segmentos e de cristal líquido (LCD), resistências, relés, sensores diversos (pressão, temperatura, etc.) e muitos outros. São chamados de controles lógicos, pois a operação do sistema baseia-se nas ações lógicas que devem ser executadas, dependendo do estado dos periféricos de entrada e/ou saída. [6]

O microcontrolador é programável, pois toda a lógica de operação é estruturada na forma de um programa e gravada dentro do componente. Depois

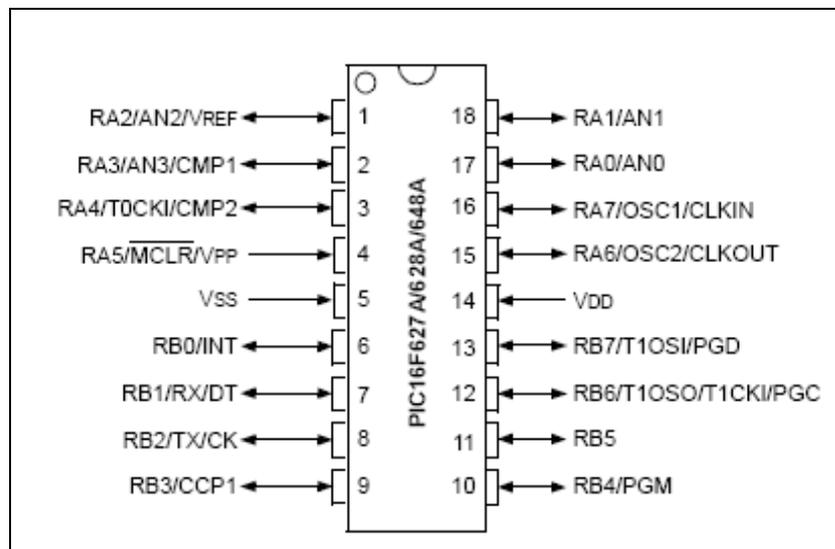
disso, toda vez que o microcontrolador é alimentado, o programa interno é executado. Quanto à “inteligência” do componente, a mesma pode ser associada à Unidade Lógica Aritmética (ULA), pois é nessa unidade que todas as operações matemáticas e lógicas são executadas. Quanto mais poderosa a ULA do componente, maior sua capacidade de processar informações. Nessa definição o microcontrolador ganhou ainda o adjetivo “pequeno”, pois em uma única pastilha de silício encapsulada (popularmente chamada de CI ou CHIP), há todos os componentes necessários ao controle de um processo, ou seja, o microcontrolador está provido internamente de memória de programa, memória de dados, portas de entrada e/ou saída paralela, contadores, comunicação serial, PWMs, conversores analógico-digitais, dentre outros. Esta é uma das características fundamentais que diferencia os microcontroladores dos microprocessadores, pois os últimos, apesar de possuírem uma ULA muito mais poderosa, não possuem todos esses recursos em uma única pastilha. [6]

Tipicamente, um microcontrolador caracteriza-se por incorporar no mesmo encapsulamento um microprocessador (com a finalidade de interpretar as instruções de programa e processar dados), memória de programa (com a finalidade de armazenar de maneira permanente as instruções do programa), memória de dados (com a finalidade de memorizar os valores associados com as variáveis definidas no programa), uma série de pinos de entrada/saída (com a finalidade de realizar a comunicação do microcontrolador com o meio externo) e vários periféricos (tais como temporizadores, controladores de interrupção, temporizadores cão de guarda (*WatchDog Timers-WDTs*), comunicação serial, geradores de modulação por largura de pulsos ou de PWM (*Pulse Width Modulation*), conversores analógico/digital etc.). [2]

Os microcontroladores PIC apresentam uma estrutura de máquina interna do tipo Harvard, enquanto grande parte dos microcontroladores tradicionais apresenta uma arquitetura tipo Von-Neumann. A diferença está na forma como os dados e o programa são processados pelo microcontrolador. Na arquitetura tradicional, tipo Von-Neumann, existe apenas um barramento (*bus*) interno (geralmente de 8 bits), por onde passam as instruções e os dados. Já na arquitetura tipo Harvard existem dois barramentos internos, sendo um de dados e outro de instruções. No caso dos microcontroladores PIC, o barramento de dados é sempre de 8 bits e o de instruções pode ser de 12,14 ou 16 bits, dependendo do microcontrolador. Esse tipo de

arquitetura permite que, enquanto uma instrução é executada, outra seja “buscada” da memória, o que torna o processamento mais rápido. Além disso, como o barramento de instruções é maior do que 8 bits, o *OPCODE* da instrução já inclui o dado e o local onde ele vai operar (quando necessário), o que significa que apenas uma posição de memória é utilizada por instrução, economizando assim muita memória de programa. [6]

O PIC16F628A foi o modelo escolhido para este projeto porque ele é versátil, compacto, rápido, poderoso e de fácil aprendizagem. A figura 2.1 mostra o encapsulamento do PIC16F628A com a descrição de suas pinagens. [6]



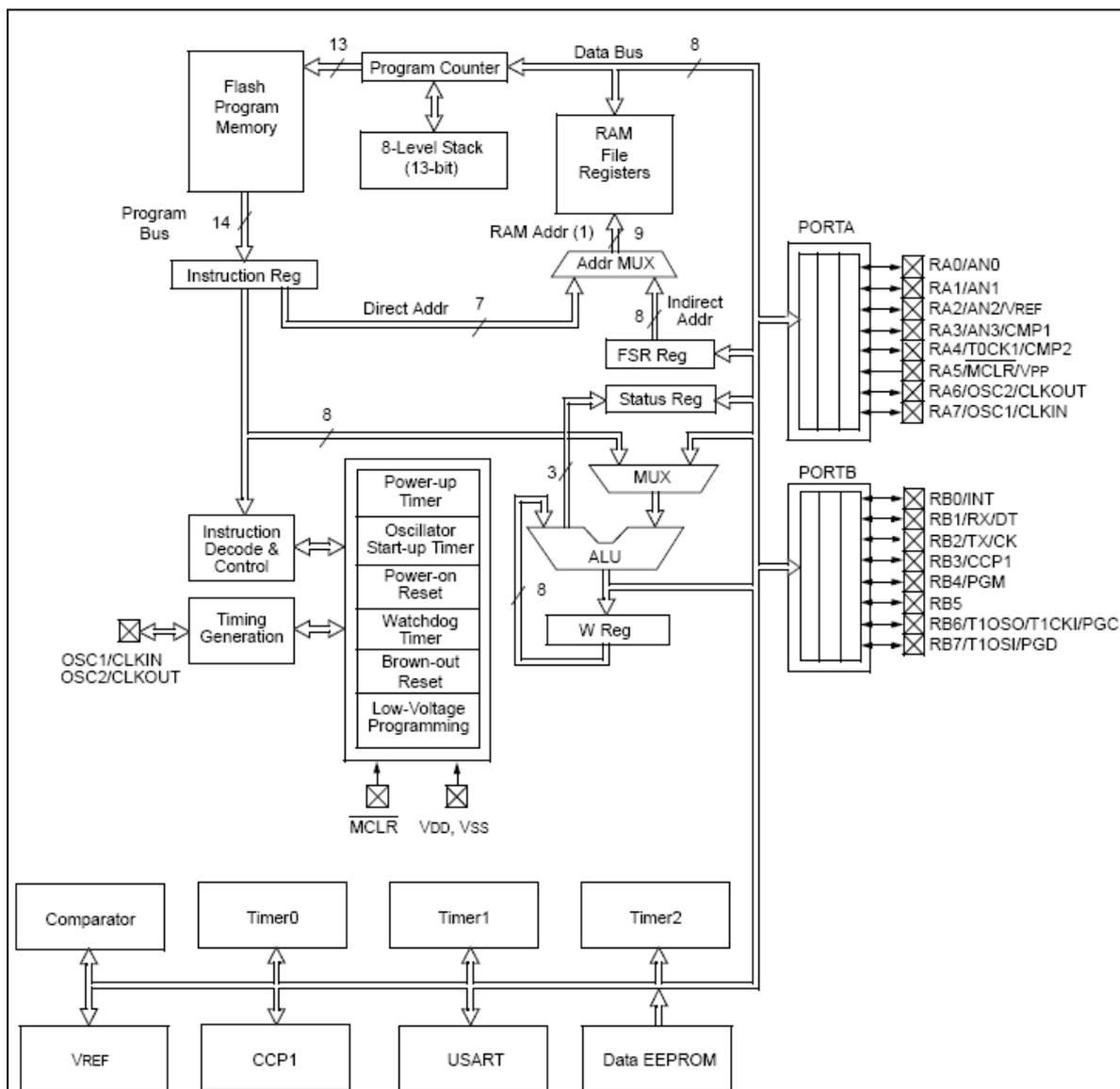
**Figura 2.1 - Chip do PIC16F628A**

Suas principais características são 16 pinos de I/O, frequência de até 20Mhz, oscilador interno de 4Mhz/37Khz, 2Kx14bits de memória FLASH de programa, 224x8 bits de memória RAM para dados, 128x8 bits de memória EEPROM interna, módulo TIMER 0 (contador de 8bits), TIMER 1 (contador de 16 bits), TIMER 2 (contador de 8 bits, módulo de referência de tensão (com 16 passos), dois comparadores analógicos com referência interna programável de tensão, um canal de comunicação serial (USART<sup>1</sup>), um *WatchDog timer*, capacidade de corrente de 25mA por pino de

<sup>1</sup> Interface Universal de Comunicação Serial Síncrona e Assíncrona

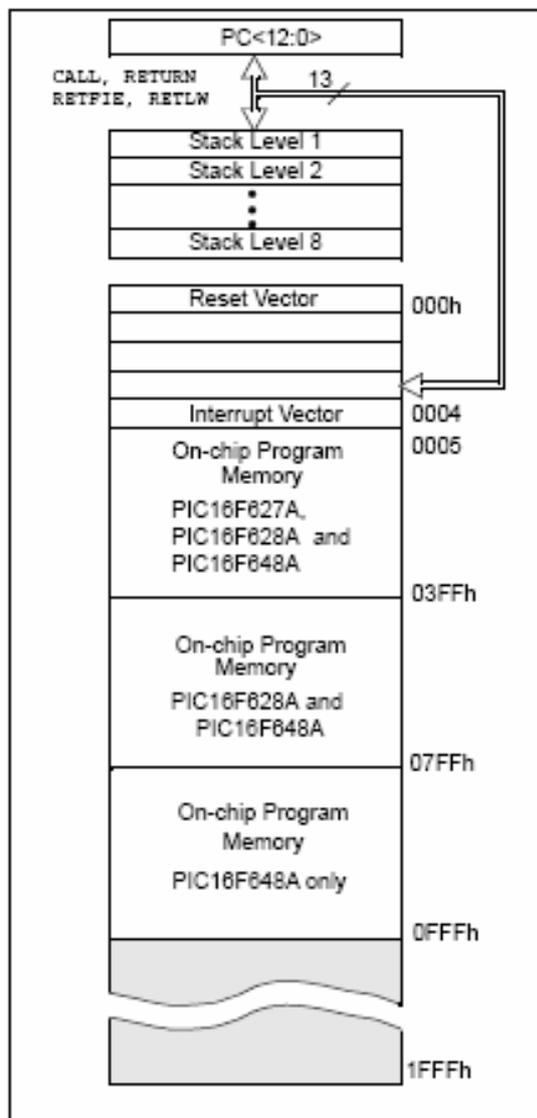
I/O, 35 instruções, 10 fontes de interrupção independentes, modo de funcionamento *sleep* e tensão de operação entre 3v a 5v. [6] [10]

No diagrama de blocos apresentado na figura 2.2 podem ser visualizadas as diversas partes que compõem o microcontrolador PIC16F628A. A ULA, diretamente ligada ao registrador *W* (*work reg*). No canto superior esquerdo temos a memória de programa, e saindo deste bloco temos um barramento de 14 bits (*program bus 14*). Mais ao centro está a memória de dados (RAM). Ela já possui um barramento de 8 bits (*data bus 8*), conforme explicado na definição da arquitetura Harvard. Do lado direito podemos visualizar as portas com todos os seus pinos de I/O. Na parte inferior, os periféricos, tais como a EEPROM (memória de dados não volátil), os *timers* (0, 1 e 2), o comparador interno, o módulo CCP (*capture compare* e PWM) e a porta serial (USART). Um pouco mais ao centro, está o registrador de *status* (*status reg*). Algumas informações importantes sobre as operações aritméticas da ULA ficam armazenadas nesse registrador. Na parte superior temos ainda o contador de linha de programa (*program counter*) e a pilha de 8 níveis (*stack*). Entre todos o periféricos, a comunicação é feita por meio de um barramento de oito vias. Há ainda os circuitos internos de *reset*, osciladores, *watchdog timer* (WDT), *power up* e *Brown-out* internos. [8] [10]



**Figura 2.2 - Diagrama de blocos do PIC16F628A**

O PIC16F628A possui 2.048 localidade de memória de programa. Todas as instruções que ele executa possuem o mesmo tamanho, 14 bits, e cada instrução ocupa uma localidade de memória de programa. O PIC16F628A possui um barramento de endereços de 13 bits para acesso à memória de programa, o que dá uma capacidade de endereçamento de 8K, porém somente 2K (2.048 localidades) foram implementadas fisicamente nesse componente, utilizando a faixa de endereços de 0000h a 07FFh. Por isso, os últimos 6K de localidades da memória de programa (0800h a 1FFFh) estão reservados, ou seja, não têm utilidade nesse microcontrolador. [10] [8] A figura 2.3 apresenta o mapa de programa e dapilha.



**Figura 2.3 - Mapa da memória de programa e da pilha**

Já a memória RAM é dividida em quatro bancos e é compartilhada pelos registradores de propósito geral (GPRs) e pelos registradores com finalidades específicas (SFRs). É importante observar que, para acessar um determinado registrador, é preciso fazer o chaveamento para o banco em que ele se encontra. Por exemplo, para acessar o registrador PIE1, é necessário fazer o chaveamento para o banco 1, onde ele está localizado, ocupando o endereço 8Ch. Para facilitar o acesso a alguns registradores, foram criados espelhos em outros bancos. Esse é o caso do registrador STATUS, que pode ser acessado de qualquer um dos bancos. [6] [8] [10]



## 2.3 Linguagem de programação

De maneira geral, programação é uma seqüência de operações que um sistema deve executar para que uma tarefa determinada seja realizada. Cada operação corresponde a uma instrução que pode ser interpretada e executada pelo microcontrolador. As instruções são constituídas por um conjunto de bits, que ficam armazenados na memória de programa do microcontrolador. Esses bits são lidos na memória de programa, são decodificados para serem acionadas as variáveis de controle internas ao sistema, para que a operação correspondente à instrução seja executada. A quantidade de bits pode variar de uma instrução para a outra, dependendo do microcontrolador. No caso particular do PIC16F628A, todas as instruções possuem o mesmo tamanho, 14 bits. De maneira geral, as instruções são formadas por dois campos distintos: o código de operação (*OPCODE*) e o operando. Os primeiros bits da instrução representam o código de operação, os demais formam o operando e podem conter dados, registrador ou endereço. Nem todas as instruções possuem operando, como no caso da instrução *RETFIE*, cuja função é retornar de uma interrupção. [8]

Código de operação é o campo da instrução que especifica qual tarefa deve ser realizada pelo microcontrolador e quando a instrução é processada. O *OPCODE* é um conjunto de bits, e decorar o código em binário referente a cada uma delas, é tarefa quase impossível. Para resolver este problema foi dado um nome a cada instrução, o *MNEMÔNICO* da instrução, uma forma abreviada de descrição em inglês da instrução. Por exemplo: *MOV* (mover), *ADD* (somar), *CLR* (limpar) e *NOP* (não faz nada). [8]

Outro campo da instrução é o operando, quando ele existir, pois, nem todas as instruções possuem operando. Existem três tipos de informações diferentes que podem estar no operando de uma instrução do PIC16F628A. Os tipos de operandos que podem aparecer em uma instrução são: registrador (F), bit de um registrador (B) e dado (L). [8]

Todo microcontrolador é capaz de executar um conjunto de tarefas denominadas instruções. O PIC16F628A é capaz de executar 35 instruções, que

estão divididas em quatro grupos: operações com registradores, com bits, com literais e operações de controle. [8]

Na linguagem *Assembly* um programa editor de texto é utilizado para editar o programa em *Assembly* (programa fonte) e um programa chamado Montador (Assembler) é o responsável pela conversão dos mnemônicos em linguagem de máquina (programa objeto). Um equipamento chamado programador é utilizado para colocar o programa em linguagem de máquina na memória de programa do microcontrolador. [8]

A linguagem *Assembly* é composta por cinco elementos básicos: *Label*, Mnemônico, Operando, Comentário e Diretivas. [8]

O *Label* é uma *string* utilizada, entre outras coisas, para representar o início de um bloco de instruções que executa uma tarefa definida dentro do programa. O bloco de instruções é normalmente chamado de sub-rotina. O *Label* deve começar na coluna 1 e pode ter um comprimento máximo de até 32 caracteres, tendo obrigatoriamente que começar com um caractere alfanumérico ou    (*underline*). O *Label* pode ser sucedido pelo : (*colon*), o qual é tratado como um operador do *Label* e não como parte dele. É muito comum o uso do *Label* em conjunto com as instruções que provocam desvio no programa, como CALL e GOTO. Neste caso o *Label* substitui o endereço para onde o programa será desviado. [8]

O Mnemônico é parte da sintaxe de uma instrução e especifica qual tarefa será executada. Como já foi mencionado, a parte da instrução que especifica a tarefa que será realizada é chamada de *OPCODE* (código de operação). [8]

O Operando pode ou não fazer parte da sintaxe de uma instrução. Ele pode ser um registrador, uma constante, valor numérico ou uma variável. Quando há mais de um operando na instrução, eles são separados por vírgula. [8]

O Comentário é um texto colocado após cada instrução para informar a tarefa que está sendo executada, fazendo com que o programa fique mais legível e fácil de entender. Todo Comentário deve ser precedido do ; (ponto-e-vírgula), fazendo com que tudo que esteja escrito a partir do ponto-e-vírgula, até o final da linha seja desprezado pelo montador. [8]

As Diretivas, conhecidas também como pseudo-instruções, são comandos que não fazem parte do *set* de instruções do microcontrolador, mas são reconhecidas pelo programa montador. As mesmas não são convertidas diretamente em linguagem de máquina. Elas são usadas para controlar o *Assembly*. As Diretivas

mais utilizadas em programas escritos em *Assembly* são: #DEFINE, #INCLUDE, ORG, QUE, MACRO, ENDM, CBLOCK, ENDC e END. [8]

## Capítulo 3. Desenvolvimento do projeto

Este capítulo apresenta a implementação, os testes e os resultados obtidos durante o desenvolvimento do projeto além da simulação do funcionamento. O item 3.1 e seus subtópicos referem-se à descrição do hardware implementado. Já o item 3.2 trata do software responsável pelo funcionamento do protótipo. O item 3.3 apresenta os testes e resultados e o 3.4 a simulação.

### 3.1 Hardware

A figura 3.1 mostra o diagrama em blocos deste protótipo. O controle remoto é composto por um microcontrolador, um transceptor, dois leds, dois botões e um cristal além de componentes eletrônicos auxiliares. O controle remoto tem a função de enviar ao dispositivo de controle o comando de acionamento, gerado quando o usuário pressiona um dos botões, e de apresentar o *status* do acionamento, o qual é indicado ao usuário por meio de leds.

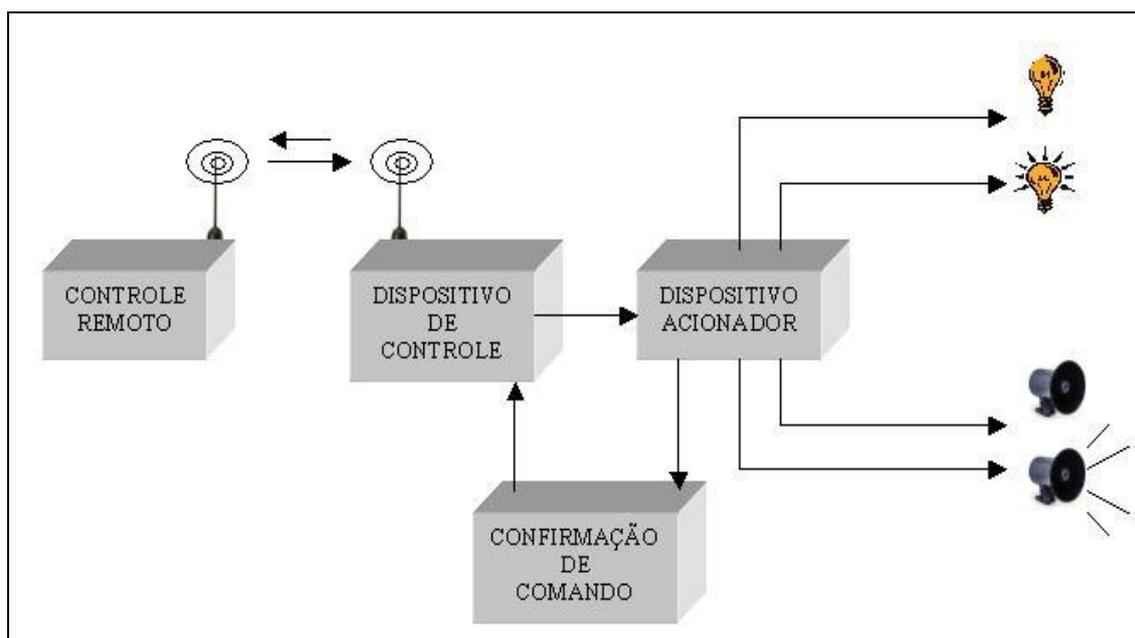


Figura 3.1 - Diagrama em blocos do Projeto

O dispositivo de controle é composto de um microcontrolador, um transceptor e componentes eletrônicos auxiliares. Este dispositivo recebe o comando de acionamento o interpreta e encaminha ao dispositivo acionador o nível lógico necessário ao acionamento. O dispositivo acionador é composto por dois relés além de componentes eletrônicos necessário ao seu funcionamento. Cada relé possui dois contatos seco, sendo um responsável pelo acionamento, e outro pela confirmação de comando.

O item 3.1.1 descreve a implementação do controle remoto. Já no item 3.1.2 é feita a descrição do dispositivo de controle e no item 3.1.3 do dispositivo acionador.

### 3.1.1 Controle remoto

Para uma melhor precisão na operação do microcontrolador é utilizado um cristal externo de 4MHz ligado ao microcontrolador conforme especificações contidas no *datasheet* do microcontrolador. Internamente a frequência do cristal é dividida por quatro, dando origem a quatro fases distintas, Q1, Q2, Q3 e Q4 as quais constituem um ciclo de máquina ou de instrução. A instalação do cristal é feita nos pinos 15 (OSC2) e 16 (OSC1) do PIC conforme indicado na figura 3.2. Para aumentar a estabilidade da frequência do cristal são utilizados dois capacitores C1 e C2 no valor de 22pF cada um conforme indicado pelo fabricante do microcontrolador. [10]

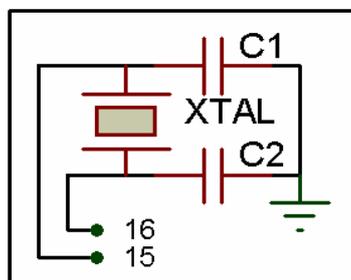
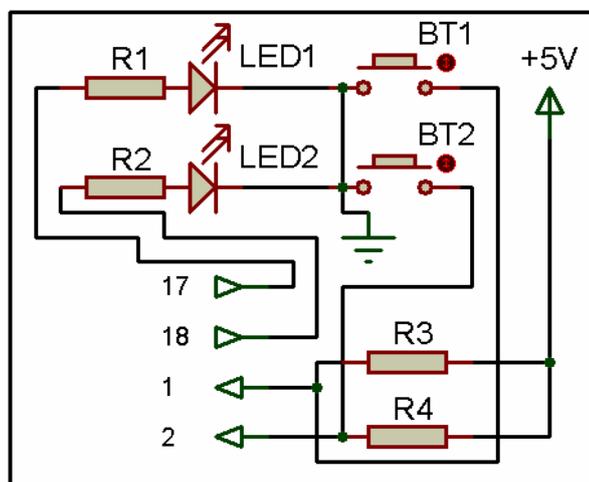


Figura 3.2 - Circuito oscilador

Os botões para o acionamento foram instalados nos pinos 1 (RA2) e 2 (RA3). Essas duas portas do PIC foram configuradas como entradas sendo que o nível lógico um (5V) corresponde ao botão solto e o nível lógico zero (0V) ao botão pressionado. Dessa forma, enquanto os botões não são pressionados tem-se como entrada a tensão da fonte de alimentação. Quando o botão é pressionado a porta correspondente é aterrada, e para evitar o curto circuito da fonte foi instalado em cada uma das portas um resistor de  $22\text{K}\Omega$  (R3 e R4) conforme ilustrado na figura 3.3. O botão um (BT1) é utilizado para acionar a lâmpada e o dois (BT2) a sirene.

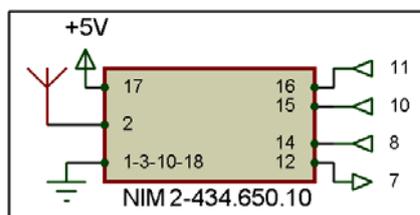


**Figura 3.3 - Esquema elétrico dos botões e dos leds de *status***

Os leds indicativos do *status* do acionamento foram instalados nas portas 17 (RA0) e 18 (RA1). Para tanto essas portas foram configuradas como saída. O LED1 indica o *status* da lâmpada e o LED2 o da sirene. Os resistores R1 e R2 são de  $330\Omega$  e foram utilizados para limitar a corrente nos leds.

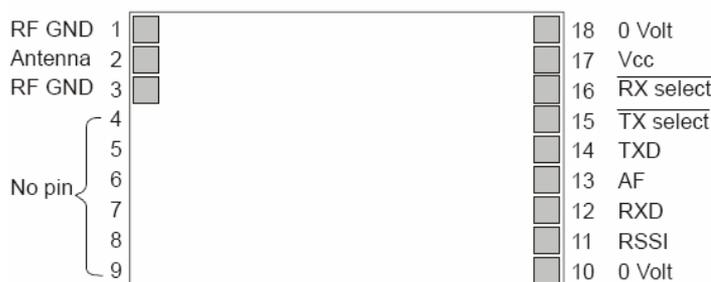
A instalação do transceptor foi feita de acordo com as recomendações de seu *datasheet*, contido no anexo. Para construção da antena foi utilizado um pedaço de fio rígido de 16,4cm e a mesma foi ligada ao pino 2 (*antenna*). Os pinos 1,3,10 e 18 foram aterrados como recomendado. Os pinos 12 (RXD) e 14 (TXD) foram respectivamente ligados aos pinos 7 (RX) e 8 (TX) do PIC. E os pinos 15 (TXSELECT) e 16 (RXSELECT) aos pinos 10 (RB4) e 11 (RB5). A alimentação do transceptor é feita com a interligação do pino 17 (VCC) ao positivo da fonte de

energia utilizada. O esquema da interligação do transceptor com o PIC é apresentado na figura 3.4.



**Figura 3.4 - Esquema de interligação do transceptor com o PIC**

O sentido da comunicação é definido pelo nível do sinal presente nos pinos RXSELECT e TXSELECT. Ambos com níveis lógicos zero (0V) ou um (5V) fazem com que o transceptor fique inoperante. Ao se aplicar 0V ao pino RXSELECT e 5V ao TX SELECT o transceptor é configurado para recepção. Ao se aplicar 0V ao pino TXSELECT e 5V ao pino RXSELECT o transceptor é configurado para transmissão. [9] Esses sentidos são selecionados pelo PIC através dos pinos RB4 e RB5. A figura 3.5 mostra a disposição dos pinos do transceptor vistos a partir do topo e sem a carcaça.



**Figura 3.5 - Pinagem do Transceptor**

A comunicação entre o PIC e o transceptor é feita por meio da USART, uma interface que permite a comunicação serial no modo síncrono ou assíncrono, conforme a configuração escolhida. Este projeto utiliza o modo assíncrono, com uma taxa de transferência de 4807,69bps, a qual é calculada de acordo com a fórmula fornecida pelo fabricante do microcontrolador, apresentada a seguir.[10]

$$TaxaDeTransferência = \frac{Fosc}{16(X + 1)} \quad (3.1)$$

$$TaxaDeTransferência = \frac{4 \times 10^6}{16(51 + 1)} = 4807,69bps$$

Neste modo de transmissão para cada caractere transmitido é utilizado um elemento de sinalização para indicar o início do caractere (START) e um outro para indicar o término do caractere (STOP). O START (bit de partida) corresponde a uma interrupção do sinal na linha e o STOP (bit de parada) à condição de marca ou repouso, ou seja, à existência do sinal na linha (normalmente o STOP corresponde a 1,4 ou 2,0 vezes o tempo de START). [5]

Pelo bit START, o receptor é avisado da transmissão de um caractere com antecedência suficiente para que possa, através de seu próprio *clock*, sincronizar seus circuitos elétricos para ler cada bit no momento apropriado. O termo "Assíncrono" refere-se a irregularidades dos instantes de ocorrência dos caracteres, ou seja, o tempo decorrido entre dois caracteres (tempo de repouso) pode ser variado pelo equipamento transmissor sem que o equipamento receptor tome conhecimento.[5]

O ritmo de transmissão assíncrono, apesar da emissão dos caracteres ser irregular, possui um sincronismo ao nível dos bits que compõem o caractere (obtido pela identificação do START), pois o equipamento receptor deve necessariamente conhecer os instantes que separam os bits dentro do caractere. [5]

Na figura 3.6 é apresentado o esquema elétrico do controle remoto onde pode ser vista a interligação do transceptor com o microcontrolador.

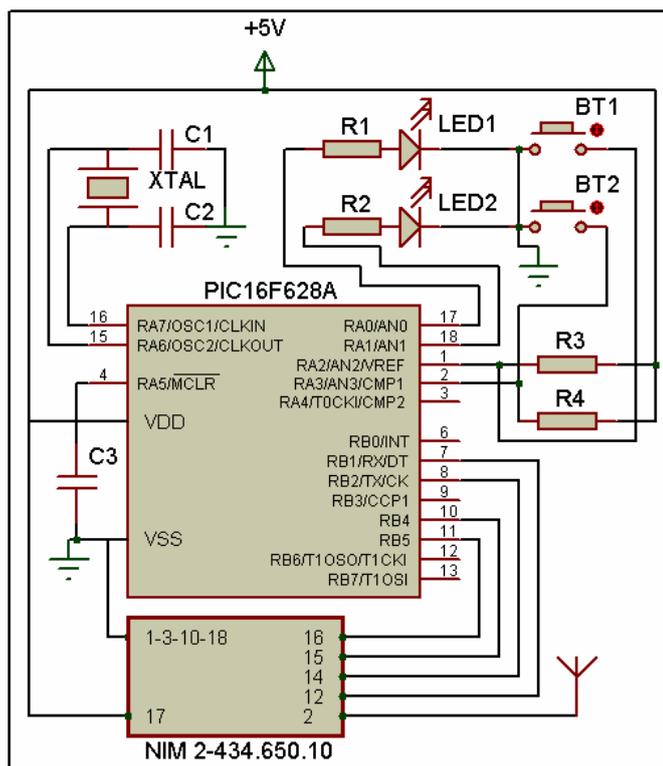


Figura 3.6 - Esquema elétrico do controle remoto

### 3.1.2 Dispositivo de controle

Neste dispositivo a alimentação, o circuito oscilador e a ligação com o transceptor são idênticas às já descritas para o controle remoto. O sinal de acionamento é disponibilizado ao dispositivo acionador através dos pinos 17 (RA0), ACLPD (acionamento da lâmpada), e 18 (RA1), ACSNE (acionamento da sirene), do PIC. O sinal referente à confirmação de comando, o qual é disponibilizado por meio de contato seco proveniente do dispositivo acionador, é detectado nos pinos 1 (RA2) e 2 (RA3). Estes pinos estão ligados conforme ilustra a figura 3.7.

Neste circuito o ponto RTLDP (retro-alimentação da lâmpada) é proveniente de um dos contatos secos do relé que aciona a lâmpada e o ponto RTSNE (retro-alimentação da sirene) do que aciona a sirene. Estes contatos, quando fechados, enviam nível lógico zero (0V) às entradas 1 e 2 do PIC. Dessa forma, como ocorre com os botões do controle remoto, os resistores R7 e R8, de 22K $\Omega$  cada, são utilizados para evitar o curto circuito da fonte de alimentação quando do recebimento do sinal da retro-alimentação.

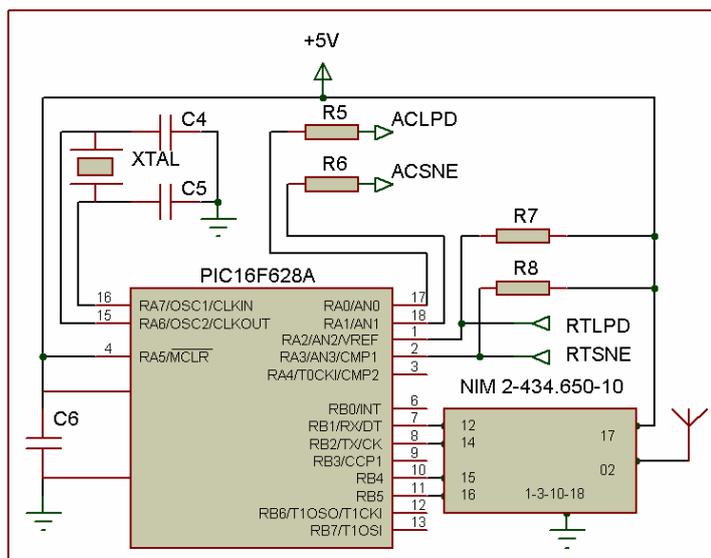
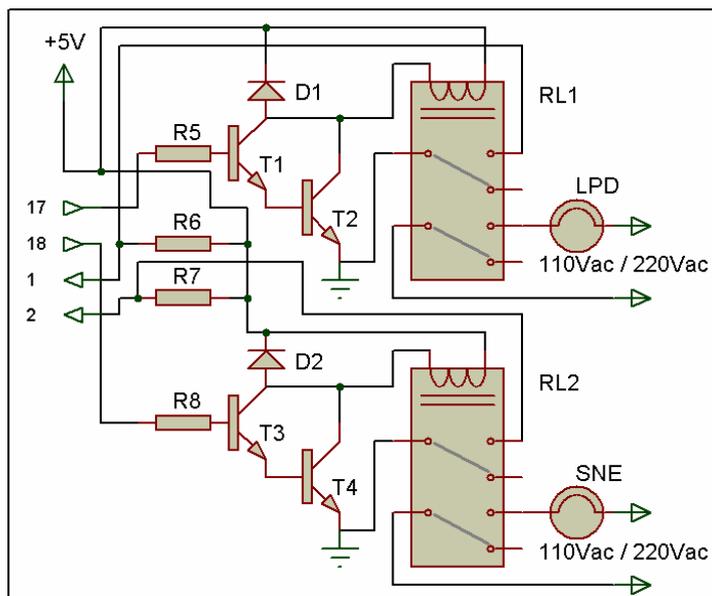


Figura 3.7 - Esquema elétrico do dispositivo de controle

### 3.1.3 Dispositivo acionador

Para cada carga a ser acionada este dispositivo possui um circuito acionador como indicado na figura 3.8. O acionamento é feito a partir do fechamento do contato seco do relé. A capacidade de condução de corrente do contato seco, de acordo com o fabricante do relé, é de no máximo 5A em corrente alternada ou 10A em corrente contínua e a tensão máxima de 250Vac ou 30Vdc. A corrente necessária para a operação do relé é da ordem de 100mA e a capacidade máxima de corrente por porta do PIC é de 25mA.[10] Então, são utilizados dois transistores ligados na forma de um par Darlington os quais funcionam como um único transistor, porém, com um ganho de corrente extremamente alto. O valor deste ganho é igual ao resultado do produto dos ganhos dos dois transistores envolvidos na ligação. [3] O resistor ligado à base do par Darlington tem a função de limitar a corrente de saída na porta do PIC fazendo com que ela fique bem abaixo dos 25mA.

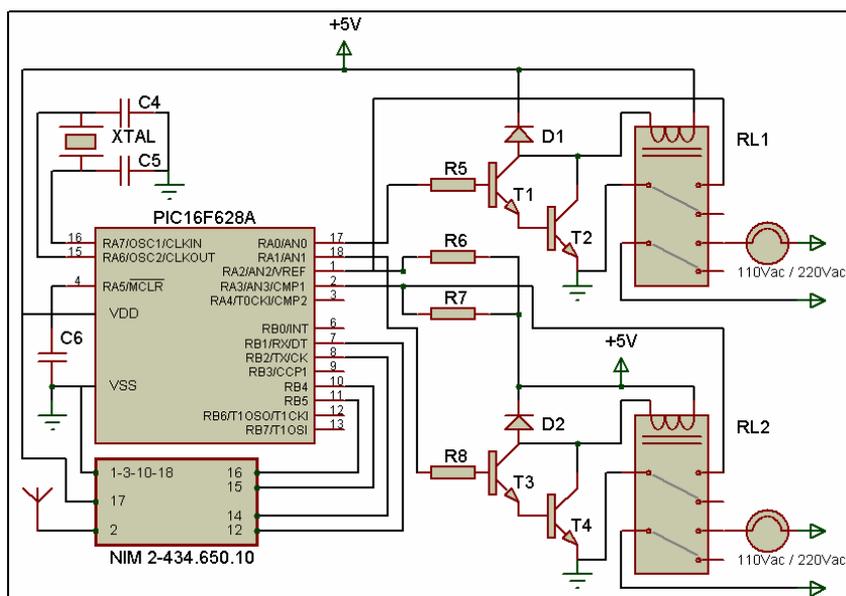


**Figura 3.8 - Dispositivo acionador**

O relé utilizado possui dois contatos secos sendo que um deles é usado para acionar a carga e o outro para informar o *status* do acionamento ao dispositivo de controle. Quando o relé entra em operação estes dois contatos são fechados. Dessa forma, enquanto a carga é acionada o PIC recebe em uma de suas entradas o nível lógico zero (0V) indicando que o acionamento foi realizado. No momento em que o acionamento é desfeito este nível lógico passa de zero para um indicando que a carga foi desligada. Os diodos D1 e D2, em paralelo com as bobinas dos relés (RL1 e RL2 respectivamente), são usados para proteger o circuito contra a tensão reversa gerada pelas bobinas dos relés no momento em que estes são desligados.

Para o acionamento da carga o dispositivo de controle entrega nível lógico um (5V) na base do transistor. Este entra em saturação conduzindo através do par Darlington uma corrente de cerca de 100mA fazendo com que o relé entre em operação e a carga seja acionada. Para desligar a carga o dispositivo de controle envia nível lógico zero (0V). E assim, o par Darlington entra em corte, o relé é desligado e a carga desativada.

O esquema elétrico do dispositivo de controle em conjunto com o dispositivo acionador é apresentado na figura 3.9.

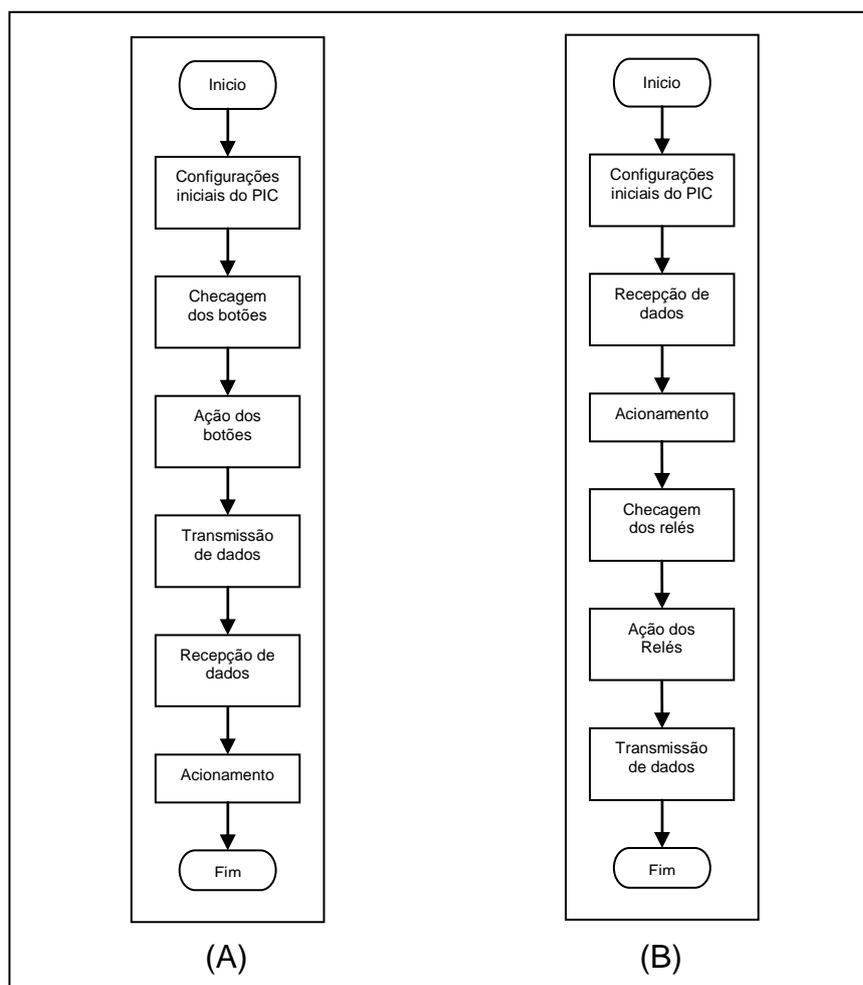


**Figura 3.9 - Dispositivo de controle e acionador**

## 3.2 Firmware

Aqui é apresentado parte do código fonte (*firmware*<sup>2</sup>) implementado neste projeto. O código fonte completo do controle remoto é apresentado no apêndice A, enquanto que o do dispositivo de controle é apresentado no apêndice B. Na figura 3.10 é apresentado o fluxograma geral desta implementação. As rotinas de transmissão e recepção de dados e de acionamento são comuns ao controle remoto e ao dispositivo de controle. A figura 3.10 (A) ilustra o fluxo de dados do controle remoto enquanto a figura 3.10 (B) a do dispositivo de controle.

<sup>2</sup> Código fonte gravado no microcontrolador.



**Figura 3.10 – Fluxograma geral**

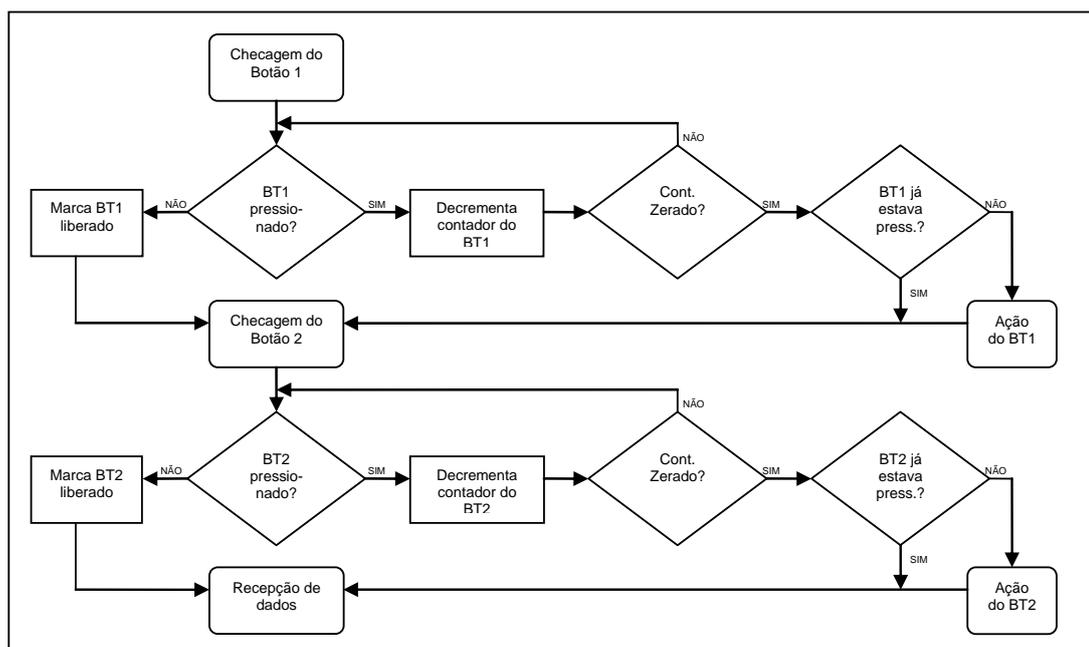
Para que o microcontrolador funcione corretamente é necessário estabelecer o modelo de PIC e o *clock* utilizado, efetuar a paginação de memória, definir as variáveis, os *flags* internos, as constantes, o vetor de *reset*, o início das interrupções e dentre outras configurações essenciais as portas de entrada e saída. Essas definições fazem parte do bloco de processo configurações iniciais do PIC constante tanto no controle remoto quanto no dispositivo de controle.

A seguir é apresentado o fluxograma e parte do código fonte dos blocos subsequentes ao bloco configurações iniciais do PIC.

As figuras 3.11 e 3.12 apresentam respectivamente o fluxograma e o trecho de código da leitura dos botões. Ao se pressionar um dos botões do controle remoto o microcontrolador realiza o *debounce* na porta correspondente ao botão pressionado.

O *debounce* é uma técnica utilizada para aguardar a estabilização do nível lógico após o fechamento do contato elétrico do botão. Esta técnica é muito útil, pois evita que o acionamento seja feito repetidas vezes indevidamente. O contador do BT1 é usado para a estabilização do botão 1 e o contador do BT2 para a do botão 2. Isto é feito por meio do decremento do contador correspondente, até que o mesmo seja zerado. Em seguida é feita a verificação do *flag* de controle do botão pressionado.

Caso o BT1 esteja marcado como “já pressionado” o fluxo é desviado para a checagem do BT2. Se o mesmo ocorre com o BT2 o desvio é feito para a rotina de recepção de dados. Quando o *flag* de controle indicar que o botão não está marcado como “já pressionado” o fluxo é desviado para a ação correspondente ao botão pressionado. Se nenhum dos botões for pressionado o programa atualiza os *flags* de controle dos botões e desvia para a rotina de recepção de dados.



**Figura 3.11 – Leitura dos botões**

```

CHECA_BT1
    BTFSC    BOTAO1
    GOTO     BT1_LIB
    DECFSZ   FILTRO1,F
    GOTO     CHECA_BT1
    BTFSS    ST_BT1
    GOTO     ACAO_BOTAO1
    GOTO     CHECA_BT2

BT1_LIB
    BCF      ST_BT1

CHECA_BT2
    BTFSC    BOTAO2
    GOTO     BT2_LIB
    DECFSZ   FILTRO2,F
    GOTO     CHECA_BT2
    BTFSS    ST_BT2
    GOTO     ACAO_BOTAO2
    GOTO     HAB_RX

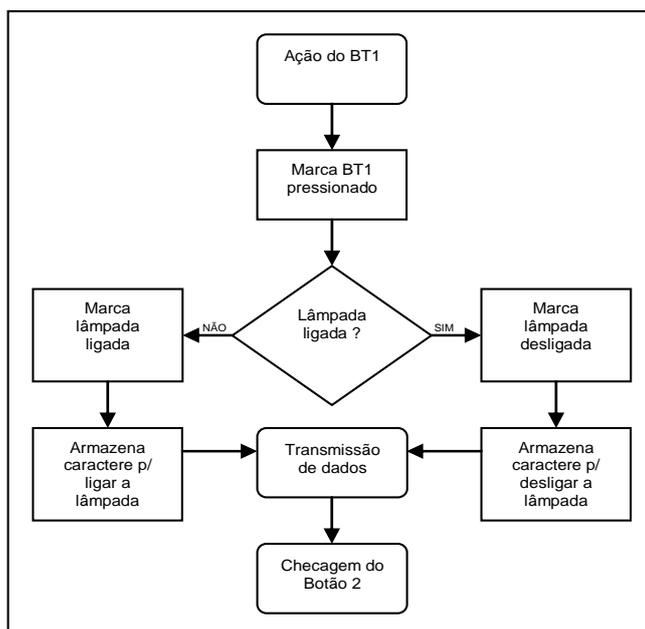
BT2_LIB
    BCF      ST_BT2
    GOTO     HAB_RX

```

**Figura 3.12 – Trecho do código da leitura dos botões**

As figuras 3.13 e 3.14 apresentam o fluxograma e o trecho de código referente à ação do botão 1. Este botão é responsável pelo acionamento da lâmpada. Inicialmente é atualizado o *flag* de controle do botão 1 e então é consultado o *flag* de *status* da lâmpada e caso a mesma esteja desligada é armazenado em uma variável de transmissão o caractere U, no código ASCII, o qual é utilizado para ligar a lâmpada. Se o flag de *status* indicar que a lâmpada está ligada, o caractere armazenado é o X. Em seguida é chamada a rotina de transmissão de dados para que seja transmitido o caractere guardado nesta variável. Após a transmissão do comando deste botão é efetuada a leitura do botão 2.

A ação do botão 2 é parecida com a descrita acima. Nesta ação os caracteres de comando são o Z e o V respectivamente utilizados para ligar e desligar a sirene. E após a transmissão o programa vai para a rotina de recepção de dados.



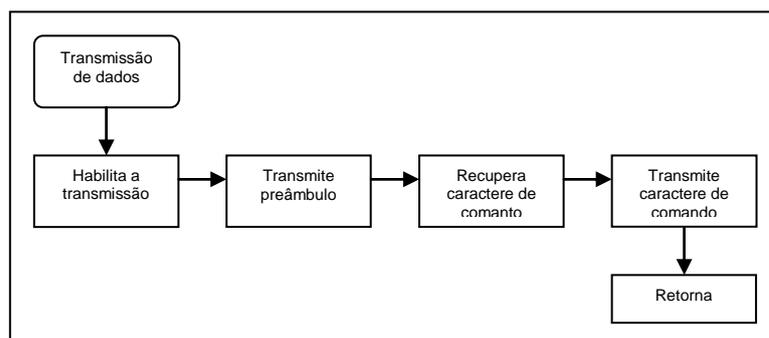
**Figura 3.13 – Ação do botão 1**

	BSF	ST_BT1
	BTFSC	ST_LPD
	GOTO	APL
	BSF	ST_LPD
	MOVLW	A'U'
	MOVWF	TX_CMD
	CALL	TX
	GOTO	CHECA_BT2
APL		
	BCF	ST_LPD
	MOVLW	A'X'
	MOVWF	TX_CMD
	CALL	TX
	GOTO	CHECA_BT2

**Figura 3.14 – Trecho do código da ação do botão 1**

As figuras 3.15 e 3.16 demonstram o fluxograma e o trecho de código da rotina de transmissão de dados. Inicialmente a transmissão do microcontrolador é habilitada, em seguida é transmitido o preâmbulo o qual é utilizado para preparar o módulo transceptor para a transmissão. O preâmbulo é composto pelo número hexadecimal AAh, utilizado para a estabilização interna do transmissor, seguido de 00h e FFh para o sincronismo do receptor, e 01h para indicar o início da mensagem. Após a transmissão do preâmbulo o comando de acionamento é recuperado da variável de transmissão e entregue ao módulo transceptor para que seja transmitido via radiofrequência. Em seguida o programa retorna ao ponto de onde a rotina foi chamada. A rotina de transmissão de dados faz uso de uma rotina auxiliar de delay de 20ms, usada para gerar um retardo entre os dados do preâmbulo.

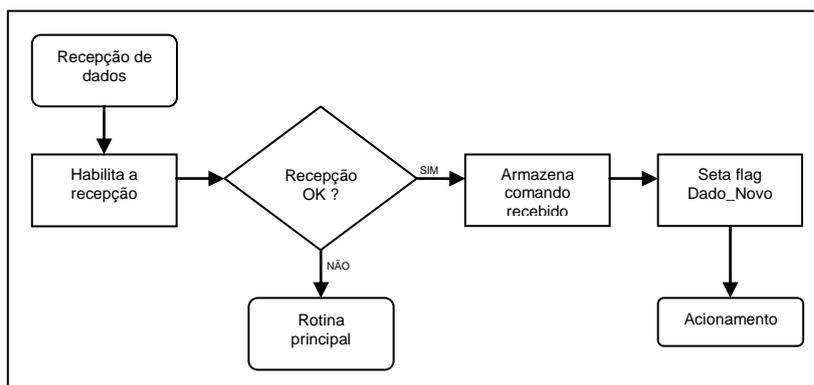
A rotina de recepção é apresentada nas figuras 3.17 e 3.18. Primeiramente é feita a configuração da USART do microcontrolador para que a recepção seja habilitada. Então é verificado se há problema de recepção, em caso afirmativo o programa é desviado para a rotina principal. Em caso negativo o caractere recebido é armazenado em uma variável, o *flag* responsável por indicar dado novo recebido é *setado* e o programa segue para a rotina de acionamento.



**Figura 3.15 – Transmissão de dados**

```
BANK1
BSF      TXSTA,BRGH
MOVLW   .51
MOVWF   SPBRG
BCF     TXSTA,SYNC
BANK0
BSF     RCSTA,SPEN
MOVLW   0xAA
MOVWF   TXREG
BANK1
BSF     TXSTA,TXEN
BANK0
MOVWF   TXREG
CALL    DL
MOVLW   0x00
MOVWF   TXREG
MOVWF   TXREG
CALL    DL
MOVLW   0xFF
MOVWF   TXREG
MOVWF   TXREG
CALL    DL
MOVLW   0x01
MOVWF   TXREG
MOVWF   TXREG
MOVF    TX_CMD,W
MOVWF   TXREG
MOVWF   TXREG
RETURN
```

**Figura 3.16 – Trecho do código da rotina de transmissão de dados**



**Figura 3.17 – Recepção de dados**

```

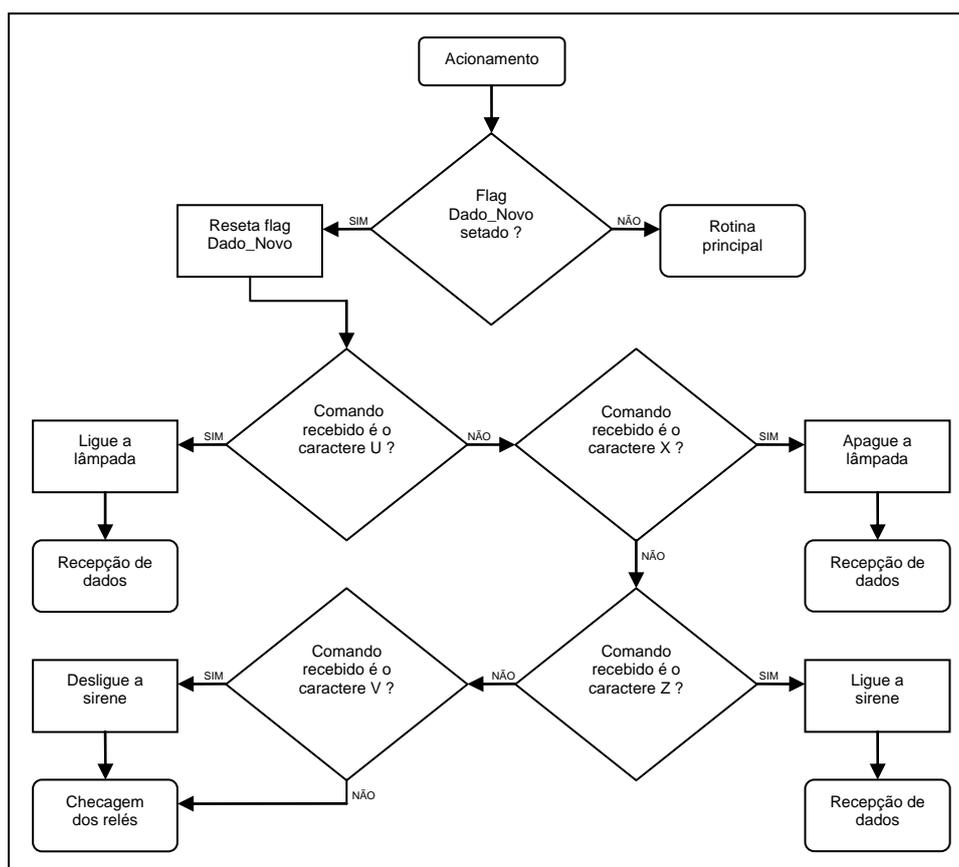
BANK1
BSF      TXSTA,BRGH
MOVLW   .51
MOVWF   SPBRG
BCF     TXSTA,SYNC
BANK0
BSF     RCSTA,SPEN
BSF     RCSTA,CREN
BTFSC   PIR1,RCIF
GOTO    $+3
BTFSC   RCSTA,FERR
GOTO    MAIN
MOVF    RCREG,W
MOVWF   DADO_RECEBIDO
BSF     DADO_NOVO
  
```

**Figura 3.18 –Trecho do código da rotina de recepção de dados**

As figuras 3.19 e 3.20 apresentam respectivamente o fluxograma e trecho do código da rotina de acionamento a qual está implementada no dispositivo de controle. Esta rotina verifica o estado do *flag* indicativo de dado novo recebido e caso não haja dado recebido o programa é desviado para a rotina principal. Se houver dado recebido o *flag* indicativo de dado novo é limpo. No próximo passo o caractere recebido é comparado respectivamente com os caracteres U, X, Z e V. Caso o caractere recebido seja U, X ou Z é executado o comando referente a este

caractere e o programa retorna para a rotina de recepção de dados. Caso seja V a sirene é desligada (ou o led indicativo do *status* da sirene é apagado no controle remoto) em seguida, o programa vai para a checagem dos relés (se o programa estiver em execução no controle remoto, o fluxo do programa é retornado para a rotina principal). Se o dado recebido não for nenhum dos comandos utilizados por esta rotina o programa vai para a checagem dos relés (no controle remoto ele vai para a rotina principal).

No dispositivo acionador o caractere U é o comando para ligar a lâmpada, o X para desligar, o Z para ligar a sirene e o V para desligar a sirene. Já no controle remoto o caractere U é usado para acender o led indicativo do *status* de lâmpada acesa e o X para apagada. O Z é utilizado para acender o led indicativo do *status* da sirene e o V para apagar.



**Figura 3.19 – Acionamento do dispositivo de controle**

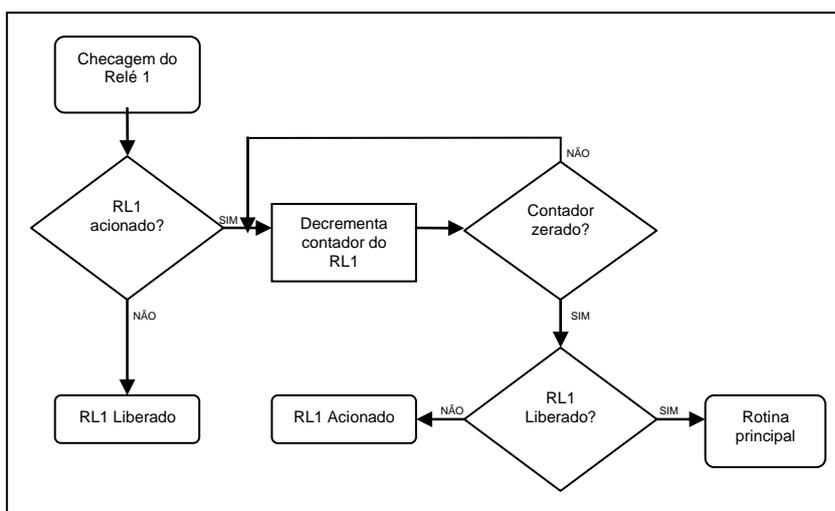
BTFSS	DADO_NOVO
GOTO	MAIN
BCF	DADO_NOVO
MOVLW	A'U'
XORWF	DADO_RECEBIDO,W
BTFSS	STATUS,Z
GOTO	\$/+3
BSF	LED1
GOTO	HAB_RX
MOVLW	A'X'
XORWF	DADO_RECEBIDO,W
BTFSS	STATUS,Z
GOTO	\$/+3
BCF	LED1
GOTO	HAB_RX
MOVLW	A'Z'
XORWF	DADO_RECEBIDO,W
BTFSS	STATUS,Z
GOTO	\$/+3
BSF	LED2
GOTO	HAB_RX
MOVLW	A'V'
XORWF	DADO_RECEBIDO,W
BTFSS	STATUS,Z
GOTO	\$/+2
BCF	LED2

**Figura 3.20 - Trecho do código da rotina de acionamento**

A rotina de checagem dos relés é composta pela checagem dos relés 1 e 2, que são muito semelhantes. As figuras 3.21 e 3.22 apresentam o fluxograma e o trecho do código da checagem do relé 1 que consiste em verificar se este relé está acionado ou não. Em caso negativo o programa segue para a sub-rotina “Relé 1 liberado” a qual faz parte da rotina de ação dos relés. Em caso positivo o contador deste relé é decrementado até que seja zerado (debounce) e então é verificado se o relé continua acionado. Se estiver liberado o programa retorna à rotina principal, se estiver acionado segue para a sub-rotina “Relé 1 acionado”, que também faz parte

da rotina de ação dos relés. A checagem do relé 2 segue os mesmos passos descritos para o relé 1.

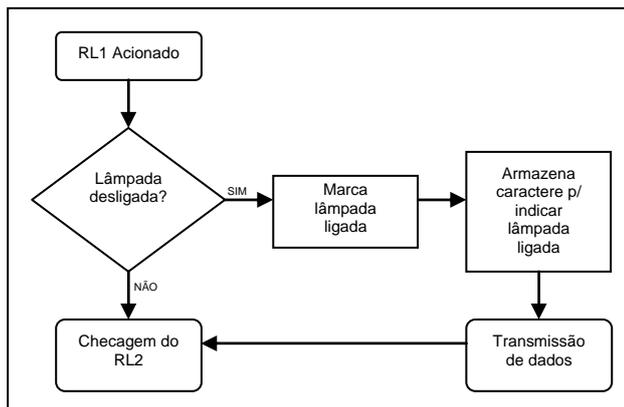
A rotina de ação dos relés é composta pelas sub-rotinas relé 1 acionado, relé 1 liberado, relé 2 acionado e relé 2 liberado. A sub-rotina relé 1 acionado verifica se a lâmpada está desligada, se não estiver o programa desvia para a checagem do relé 2, se sim marca o *flag* de controle da lâmpada, armazena na variável de transmissão o caractere U, chama a rotina de transmissão e após o retorno desta vai para a checagem do relé 2. As figuras 3.23 e 3.24 apresentam o fluxograma e trecho do código descrito acima. A sub-rotina relé 2 acionado é bastante parecida com a descrita para o relé 1, sendo que o caractere armazenado na variável de transmissão é o Z, e após a transmissão o programa vai para a rotina principal.



**Figura 3.21 – Checagem do relé 1**

BTFSC	RELE1
GOTO	APL
DECFSZ	FILTRO1,F
GOTO	-\$-1
BTFSS	RELE1
GOTO	LPD
GOTO	MAIN

**Figura 3.22 – Trecho do código da sub-rotina de checagem do relé 1**

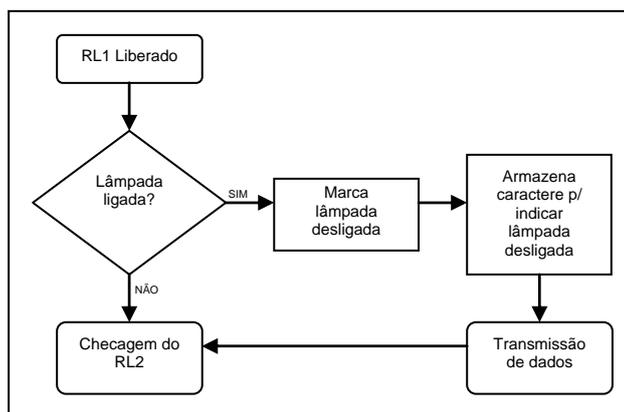


**Figura 3.23 – Relé 1 acionado**

BTFSC	ST_LPD
GOTO	CHECA_RL2
BSF	ST_LPD
MOVLW	A'U'
MOVWF	TX_CMD
CALL	TX
GOTO	CHECA_RL2

**Figura 3.24 – Trecho do código da sub-rotina relé 1 acionado**

As figuras 3.25 e 3.26 demonstram o fluxograma e o trecho do código da sub-rotina relé 1 liberado. Esta sub-rotina verifica se a lâmpada está ligada, em caso afirmativo desmarca o *flag* de controle da lâmpada, armazena o caractere X na variável de transmissão, chama a rotina de transmissão e ao retornar vai para a checagem do relé 2. Se a lâmpada estiver desligada desvia diretamente para a checagem do relé 2. Na sub-rotina relé 2 liberado o caractere armazenado e transmitido é o V e ao retornar da rotina de transmissão vai para a rotina principal, caso a sirene esteja desligada desvia diretamente para a rotina principal.



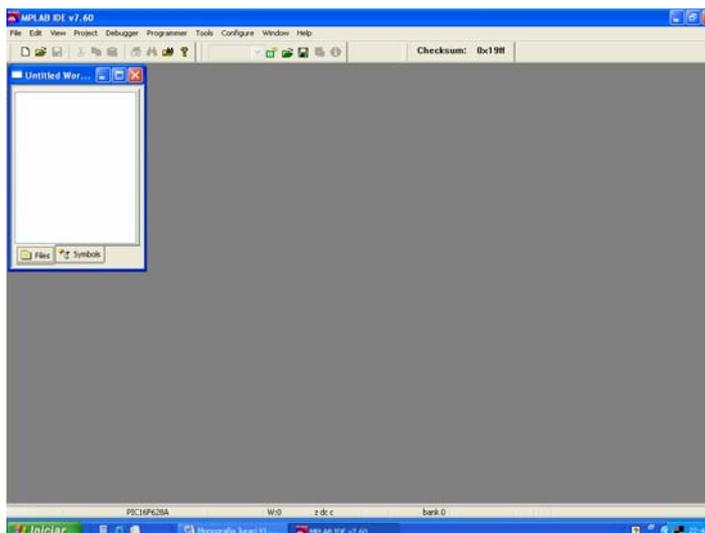
**Figura 3.25 – Relé 1 liberado**

BTFSS	ST_LPD
GOTO	CHECA_RL2
BCF	ST_LPD
MOVLW	A'X'
MOVWF	TX_CMD
CALL	TX
GOTO	CHECA_RL2

**Figura 3.26 - Trecho do código da sub-rotina relé 1 liberado**

Para o desenvolvimento do programa-fonte em *Assembly* foi utilizado o MPLAB IDE o qual é distribuído gratuitamente pela Microchip Technology. O MPLAB IDE é um conjunto de softwares que permite, entre outras coisas: criar programas, em *Assembly*, efetuar a simulação e o *debug* do programa e gravar programas no microcontrolador.

A versão do MPLAB IDE utilizada para desenvolver o software deste projeto foi a v7.60. A instalação do MPLAB é feita simplesmente executando o programa de instalação e selecionando as opções-padrão do assistente de instalação. Com um duplo-clique no ícone do MPLAB IDE tem-se acesso ao ambiente de trabalho ilustrado na figura 3.27.

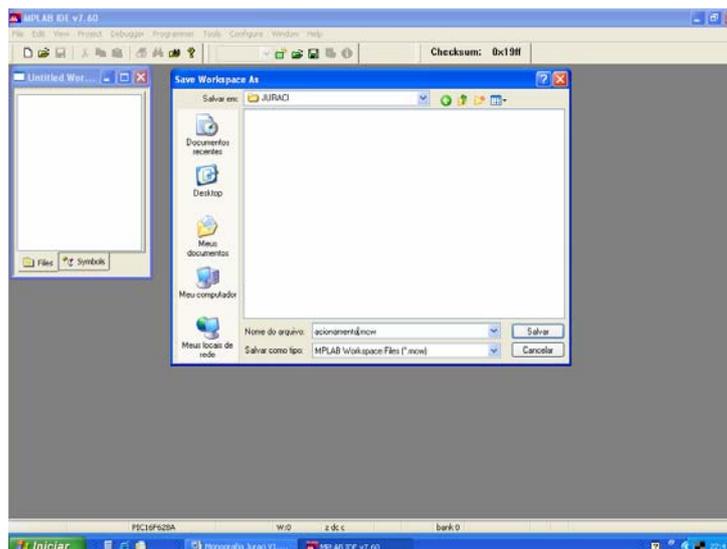


**Figura 3.27 - Ambiente de trabalho do MPLAB**

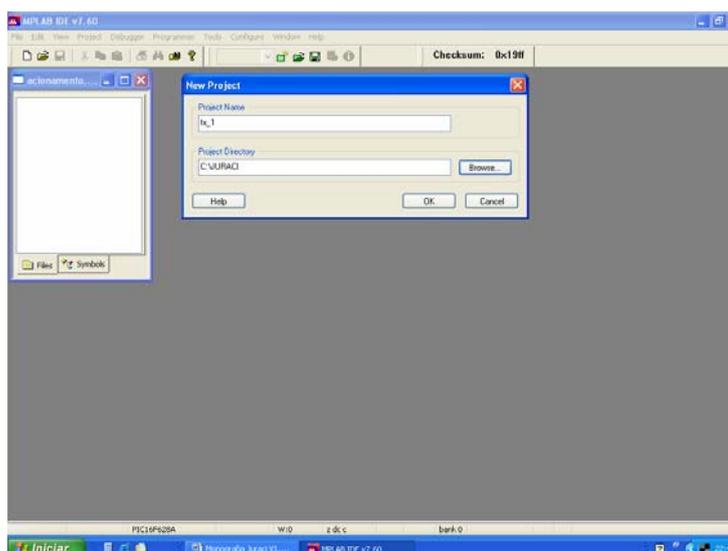
A versão 7.60 do MPLAB IDE utiliza o conceito de Projeto, e área de trabalho, conhecida como Workspace para armazenar todas as informações necessárias à compilação de um sistema. Um projeto é um arquivo que guarda as informações básicas necessárias à compilação do sistema, tais como a relação de arquivos-fonte, opções de compilação e algumas ferramentas de compilação. O conceito de Workspace é ainda mais abrangente que o de Projeto, pois ele armazena todas as demais informações necessárias ao desenvolvimento de um sistema. No arquivo do Workspace são armazenadas informações relacionadas a um ou mais projetos associados, qual está ativo no momento, quais as janelas abertas e suas posições na tela, configurações do ambiente de trabalho, etc. O MPLAB não funciona adequadamente se um Workspace/Projeto não for aberto. [8]

A figura 3.28 apresenta a janela onde foi feita a definição de nome para o Workspace. Após serem denominados o diretório e o nome do workspace, é criado um arquivo com a extensão MCW onde são armazenadas todas as informações de trabalho do MPLAB.

A figura 3.29 demonstra a criação do projeto dentro do workspace. Foi criado um subdiretório chamado **tx\_1** localizado dentro do diretório **JURACI**.



**Figura 3.28 - Definição de nome para o Workspace**



**Figura 3.29 - Criação do projeto**

Na figura 3.30 é visualizado o workspace e o projeto aberto no momento. Caso seja criado um projeto sem que seja definido o workspace, este será definido automaticamente com o mesmo nome do projeto.

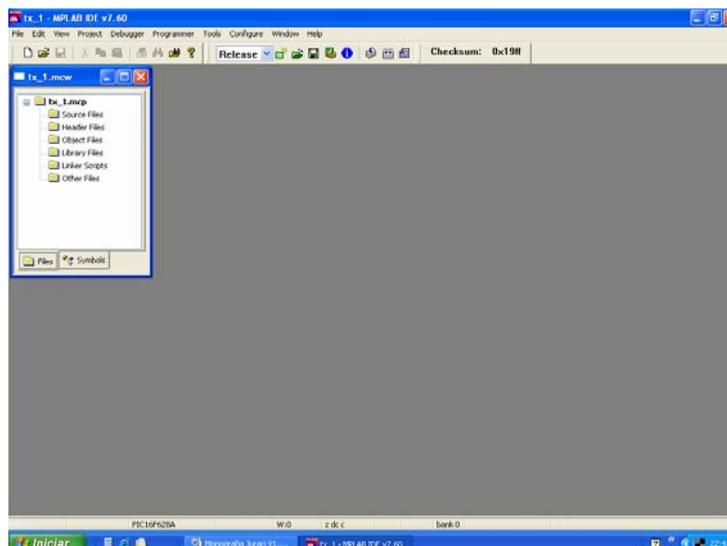


Figura 3.30 – Tela do Workspace com o Projeto Aberto

Após ter sido criado o projeto e definido o workspace o próximo passo é associar um arquivo de código-fonte ao projeto. A extensão do arquivo fonte é ASM por se tratar da linguagem Assembly. O nome do arquivo aparece na tela do workspace, dentro do grupo denominado **Source Files**. Com um duplo-clique sobre o nome do arquivo o mesmo é aberto.

A figura 3.31 apresenta a área de trabalho com o código-fonte aberto. Onde seu nome, **ACIONAMENTO\_TX1.ASM** é mostrado dentro do grupo **Source Files**.

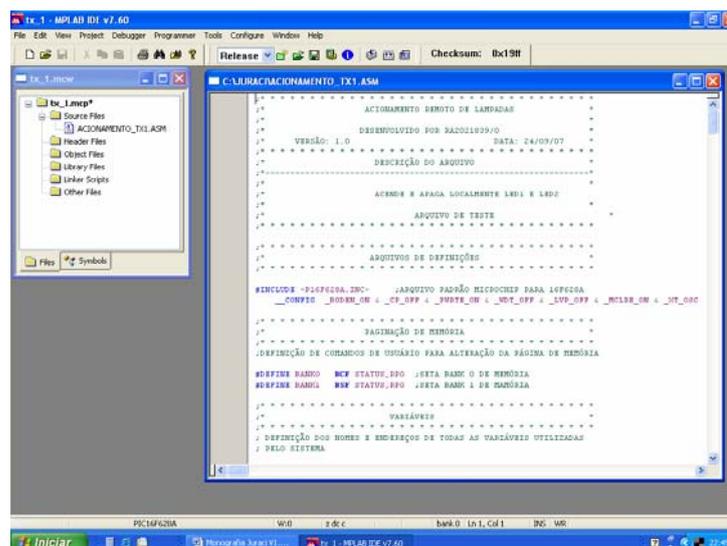
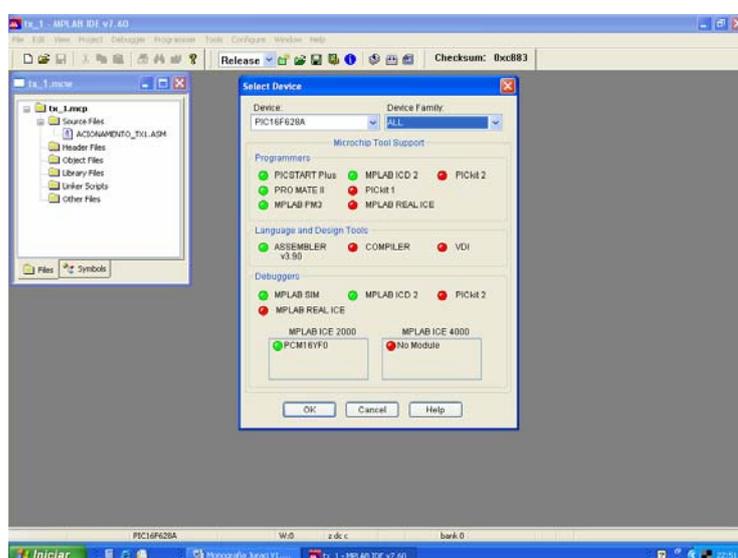


Figura 3.31 - Arquivo de código-fonte aberto

Para que o programa seja compilado é necessário a definição do microcontrolador utilizado, pois o trabalho do compilador depende dela visto que a compilação se baseia nos dados do modelo de PIC escolhido. Dessa forma, com um clique em **Configure** e outro em **Select Device** é selecionado o modelo de PIC utilizado neste projeto: PIC16F628A.

Na figura 3.32 é indicado o modelo de PIC selecionado e quais as ferramentas da Microchip estão disponíveis (LED verde) para o modelo escolhido. Estas ferramentas são o simulador (MPLAB SIM), os gravadores e os emuladores.



**Figura 3.32 - Selecionando o PIC desejado**

Na compilação do projeto, o código-fonte é processado para criar um arquivo que será entendido pelo microcontrolador. No MPLAB, quando um arquivo com a extensão ASM é compilado ele é transformado em um HEX (hexadecimal). Esse novo arquivo é que será gravado no PIC.

A figura 3.33 ilustra a compilação em andamento. A compilação foi efetuada ao clicar no ícone “Build All”, a mesma poderia ter sido efetuada através do comando **project > Build All**.

A figura 3.34 apresenta o relatório final da compilação. O importante é que a mesma foi bem sucedida, pois a mensagem “BUILD SUCCEEDED” foi apresentada no final do relatório.

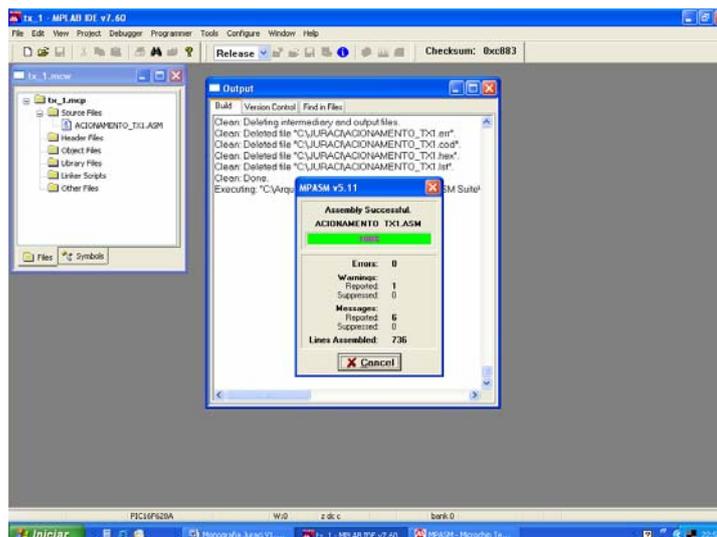


Figura 3.33 - Relatório de compilação em andamento

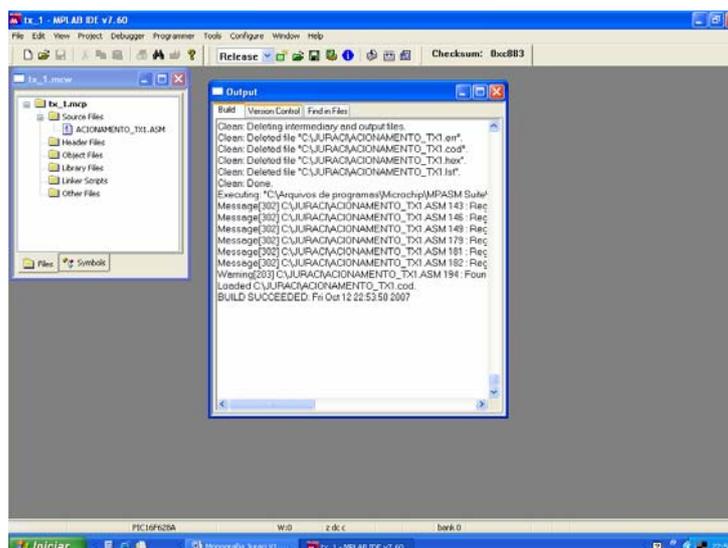


Figura 3.34 - Relatório de compilação com sucesso

Após ter obtido êxito na compilação do programa o passo seguinte foi efetuar a gravação do programa na memória do PIC16F628A. Para isso foi utilizado o gravador de PIC McPlus que é um gravador nacional de baixo custo e que opera diretamente com o MPLAB. Este gravador é composto basicamente por duas placas: o gravador propriamente dito e o soquete de gravação onde é colocado o PIC a ser gravado. O McPlus opera com todos os modelos de PIC da família FLASH, de 8 a 40 pinos (DIP). A gravação é feita por meio da porta de comunicação serial do PC.

Na figura 3.35 é apresentado o McPlus utilizado para gravação dos dois PICs utilizados neste projeto.



**Figura 3.35 - Gravador de PIC McPlus**

### **3.3 Testes e resultados**

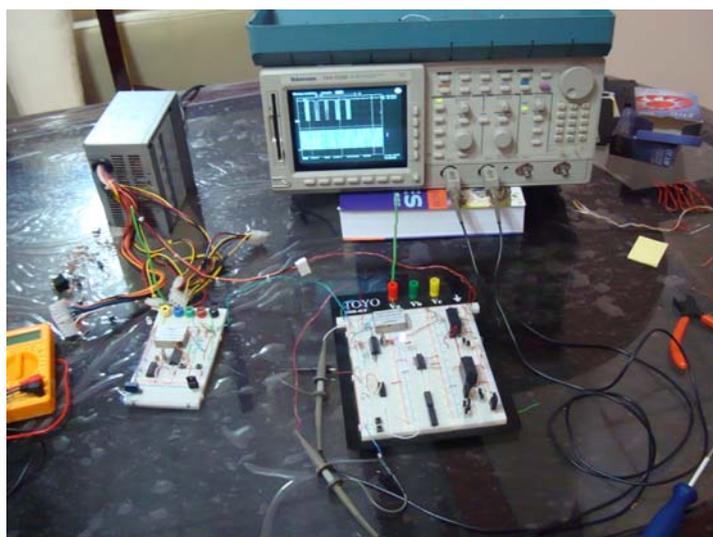
Após a fase de implementação, constatou-se que o projeto não estava funcionando como esperado. Dessa forma, vários testes foram realizados com o intuito de solucionar este problema. A partir dos resultados obtidos nos testes a implementação foi retificada e o projeto elaborado com as modificações pertinentes.

Este capítulo apresenta os testes realizados no desenvolvimento do projeto, os resultados e as ações tomadas.

As figuras 3.36 e 3.37 ilustram o ambiente onde os testes foram realizados.

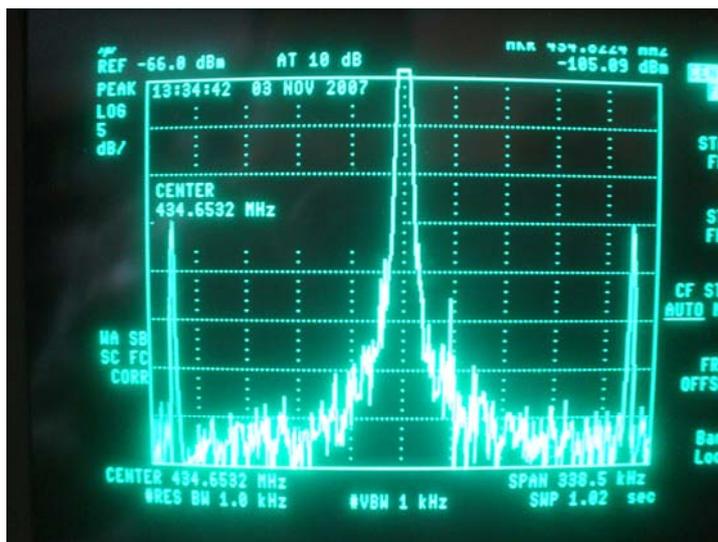


**Figura 3.36 - Ambiente de realização dos testes**



**Figura 3.37 - Ambiente de realização dos testes**

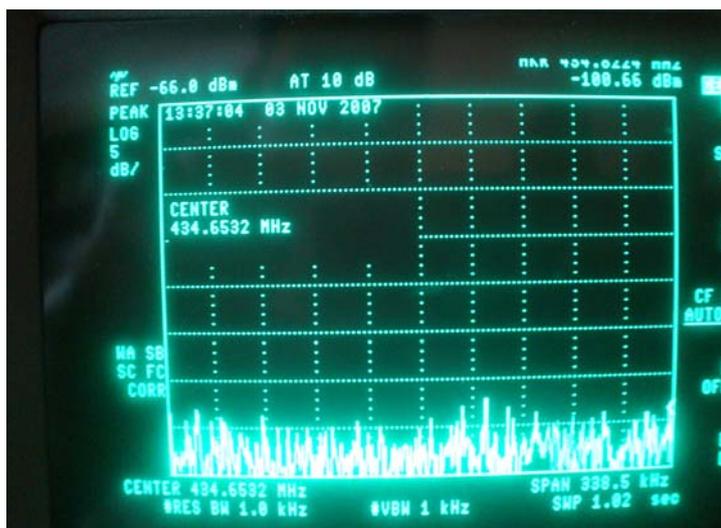
Na figura 3.38 é apresentado o espectro de frequência com a portadora do transceptor ligada onde pode ser visto a frequência de 434.6532MHz com suas bandas laterais superior e inferior. A frequência de operação do modulo transceptor é de 434.65Mhz. Dessa forma, a frequência emitida pelo transceptor encontra se correta.



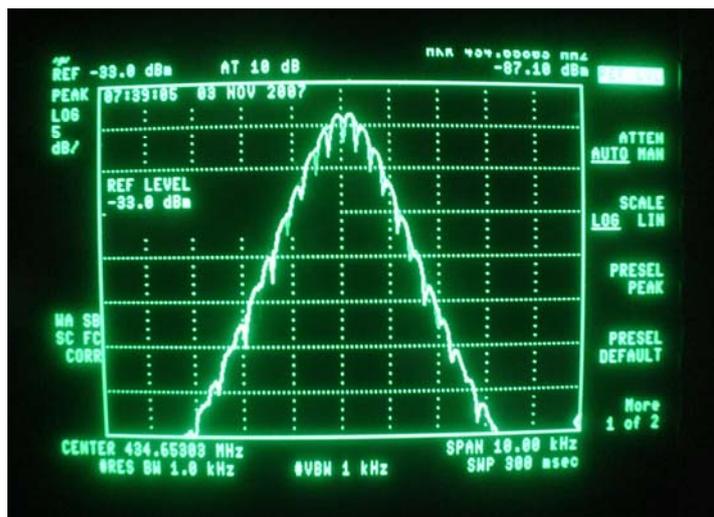
**Figura 3.38 - Portadora ligada**

A figura 3.39 apresenta o espectro de freqüência com a portadora desligada onde pode ser visto que não há interferência de freqüências externas no espectro do sinal de rádio.

A figura 3.40 demonstra a portadora com interferência de sinal elétrico. Foi constatado que o mesmo é proveniente da rede elétrica.

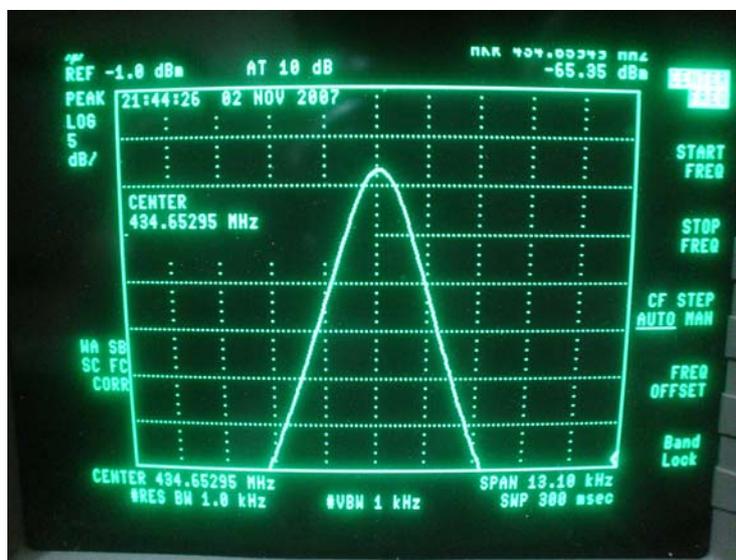


**Figura 3.39 - Portadora desligada**



**Figura 3.40 - Portadora com espúrios**

Após a substituição da fonte de alimentação e o uso de um estabilizador de energia a interferência na portadora foi eliminada como indica a figura 3.41.



**Figura 3.41 - Portadora sem espúrios**

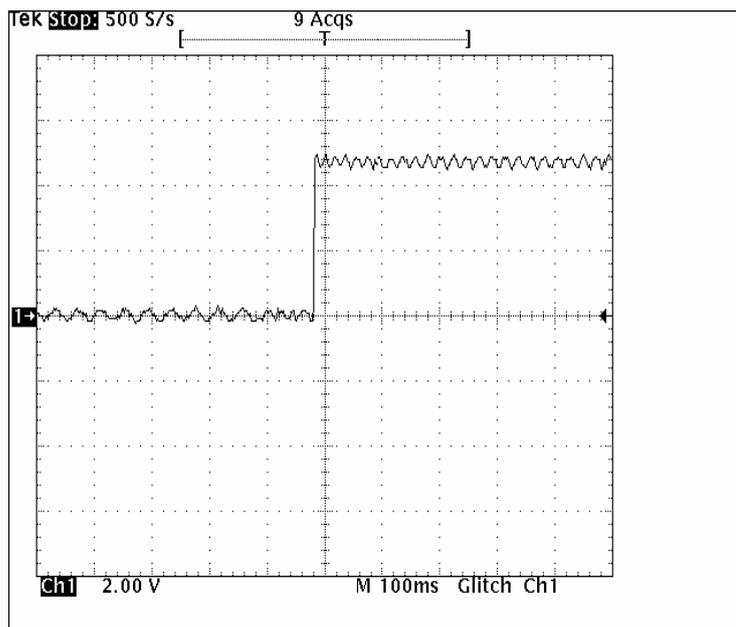
A figura 3.42 apresenta a portadora sendo modulada. O sinal modulante foi gerado ao ser pressionado um dos botões do controle remoto. Com isso verificou se que o transmissor estava operando sem restrição.



**Figura 3.42 - Portadora modulada**

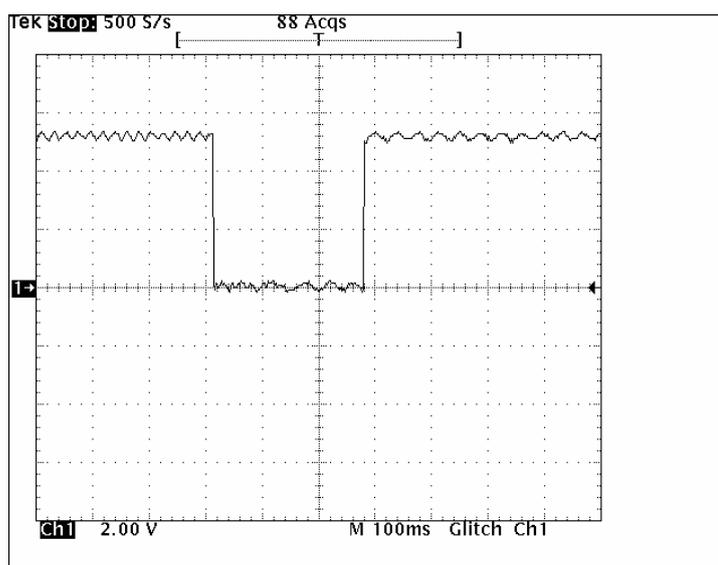
Apesar de o transceptor transmitir o sinal gerado com o pressionar dos botões não houve comunicação entre o controle remoto e o dispositivo de controle. Para solucionar este problema foi introduzido um tempo de retardo entre a transmissão de cada byte do preâmbulo.

A figura 3.43 apresenta o sinal de acionamento enviado pelo dispositivo de controle ao dispositivo acionador. Este sinal apresenta a transição de nível baixo para alto responsável pelo acionamento do relé.



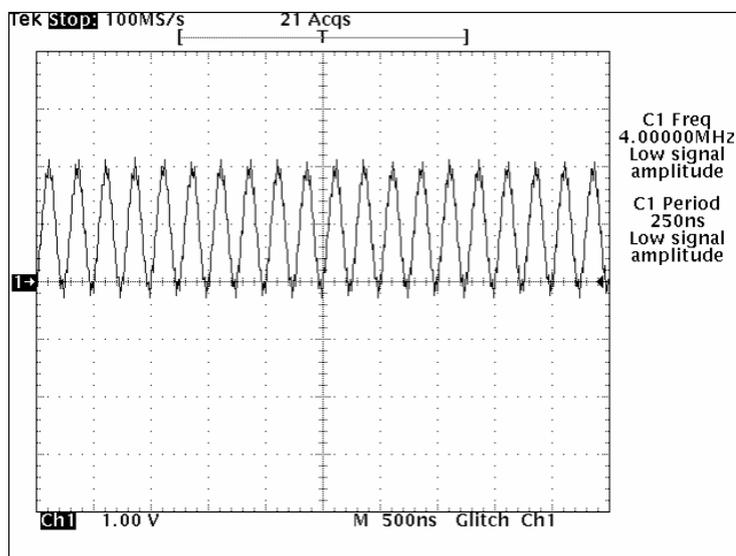
**Figura 3.43 – Sinal de acionamento**

A figura 3.44 apresenta a transição de nível alto para baixo quando o botão é pressionado. Até que o nível baixo seja estabilizado decorre um certo tempo de instabilidade. Com isso a leitura do botão somente é feita depois de decorrido o *debaunce* do botão.



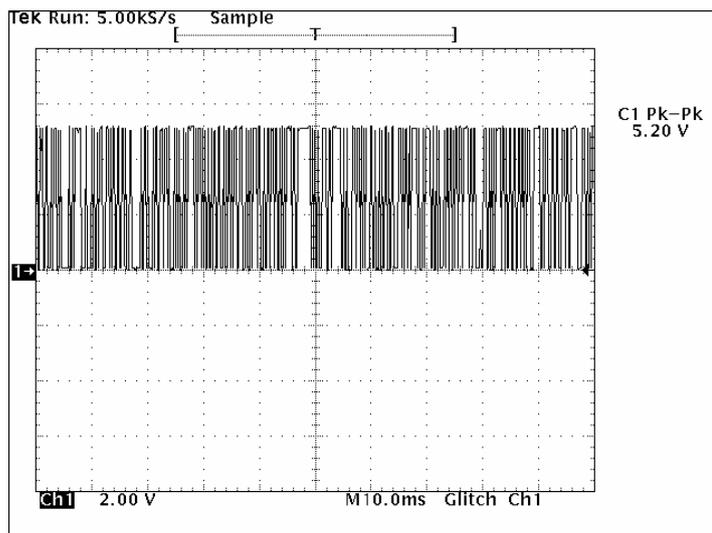
**Figura 3.44 – Acionamento do botão**

O clock usado pelo microcontrolador é apresentado na figura 3.45 onde pode ser vista a frequência de 4MHz, com um período de 250ns.



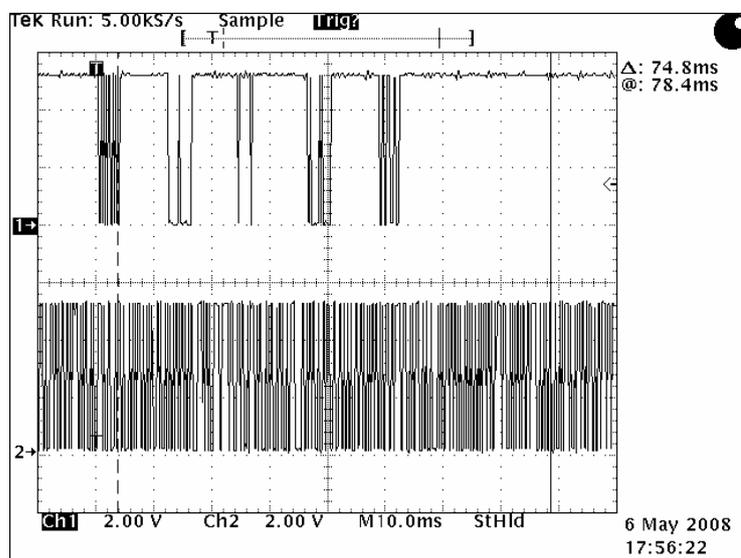
**Figura 3.45 – Sinal de clock**

Na figura 3.46 é visualizado o sinal gerado pelo receptor quando este não detecta a portadora no momento em que o controle remoto passa de transmissor para receptor. Este sinal interfere no funcionamento do dispositivo de controle causando falso comando. Conseqüentemente é gerado, indevidamente, por várias vezes, sinal de acionamento e entregue ao dispositivo acionador.



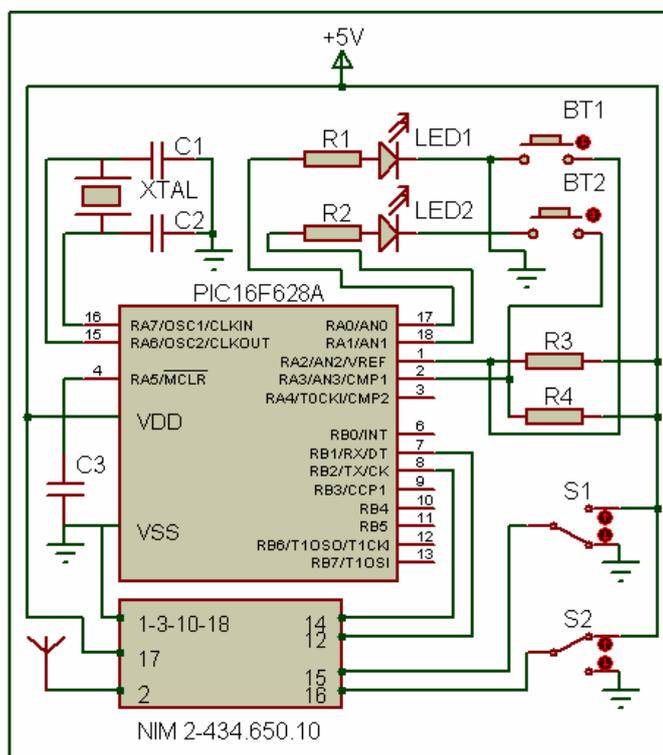
**Figura 3.46 – Sinal interferente**

A figura 3.47 apresenta o preâmbulo entregue pelo microcontrolador ao transceptor em conjunto com o sinal interferente gerado pelo receptor no momento em que o mesmo perde o sinal da portadora.



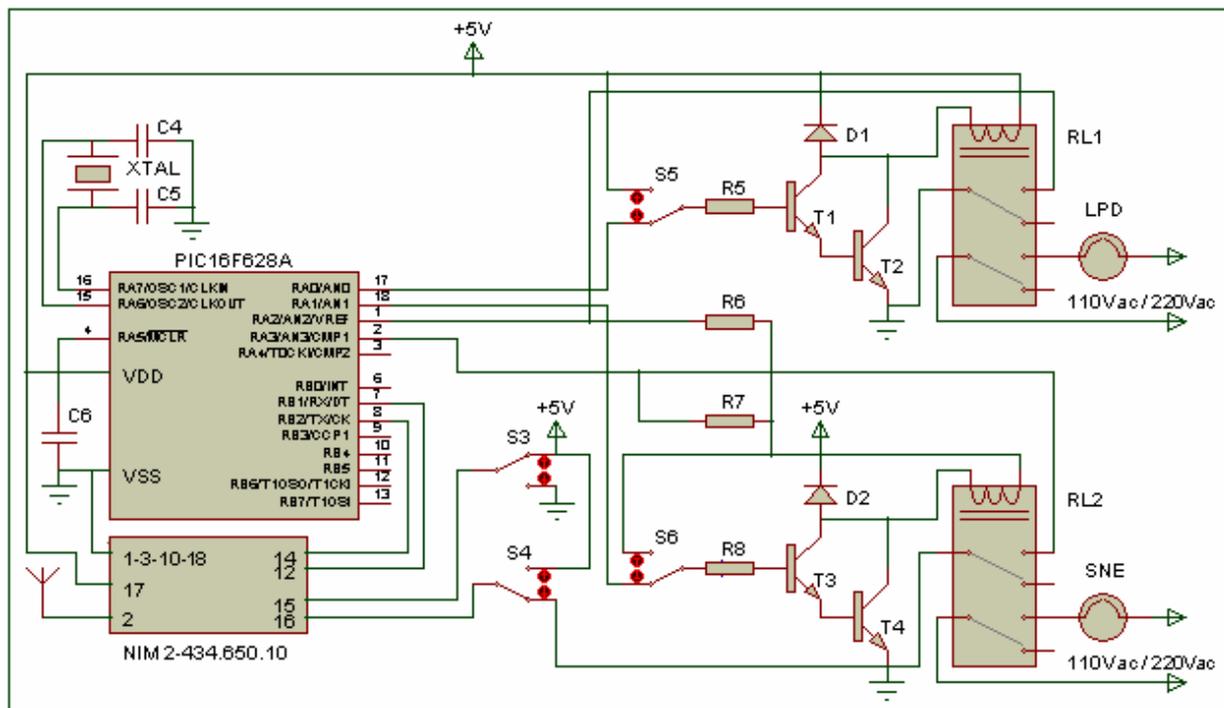
**Figura 3.47 – Preâmbulo e sinal interferente**

De forma a solucionar este problema, a seleção do sentido de transmissão no controle remoto passou a ser feito manualmente, com o uso das chaves seletoras S1 e S2 e não mais pelo microcontrolador, conforme apresenta a figura 3.48.



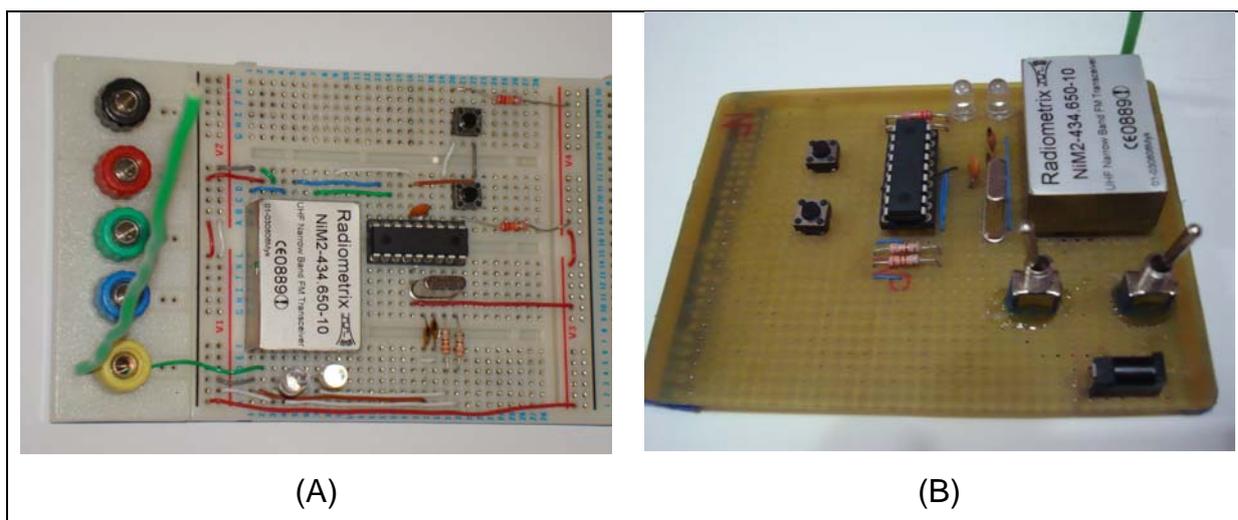
**Figura 3.48 – Controle remoto com chaves seletoras**

No dispositivo de controle o sentido da comunicação é selecionado através das chaves seletoras S3 e S4, enquanto que as chaves S5 e S6 são utilizadas para acionar manualmente os relés, simulando, desta forma, a retro-alimentação. O esquema deste dispositivo com as chaves seletoras descritas acima é ilustrado na figura 3.49.



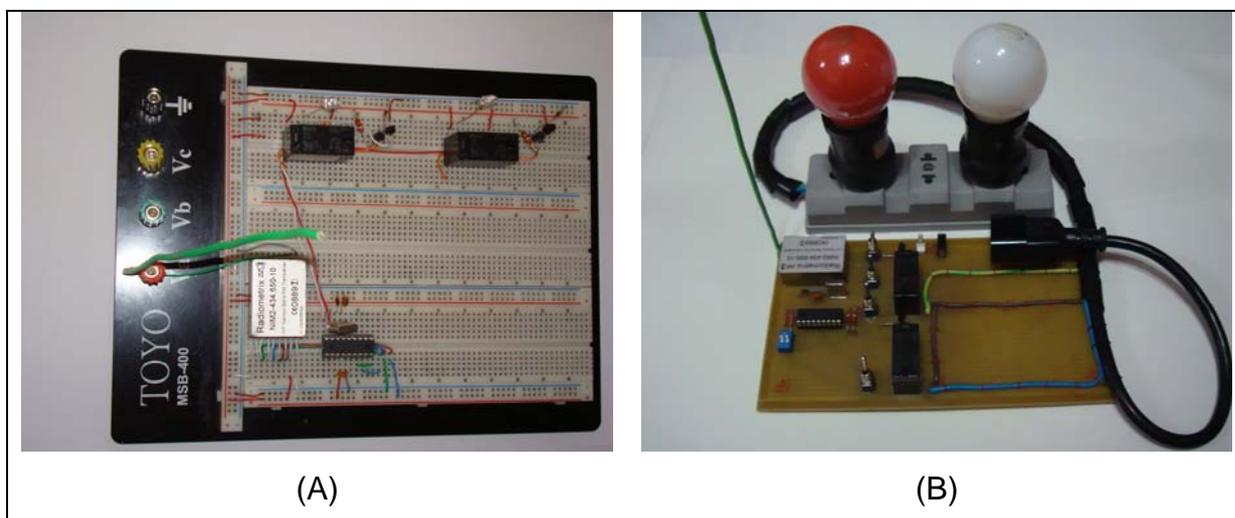
**Figura 3.49 – Dispositivo de controle com chaves seletoras**

A figura 3.50 (A) apresenta o circuito do controle remoto montado em protoboard enquanto a figura 3.50 (B) apresenta o mesmo circuito montado em uma placa de circuito universal.



**Figura 3.50 – Controle remoto**

A figura 3.51 (A) apresenta o circuito do dispositivo de controle e do dispositivo acionador montado em protoboard enquanto a figura 3.51 (B) apresenta o mesmo circuito montado em uma placa de circuito universal.



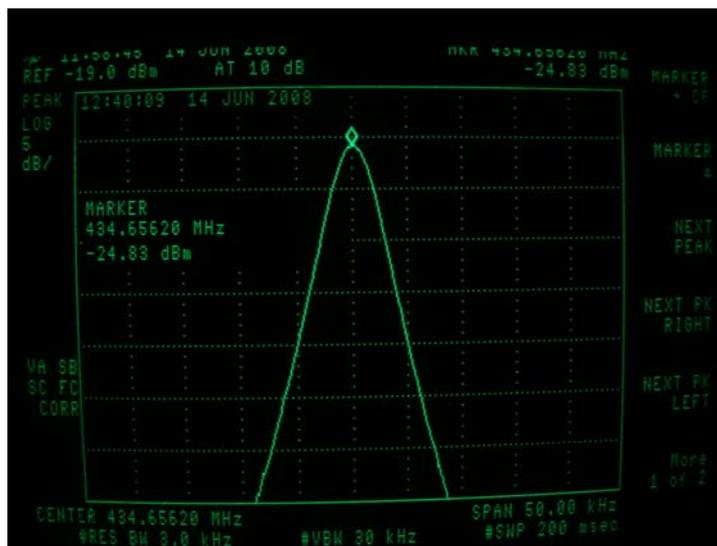
**Figura 3.51 – Dispositivo de controle e dispositivo acionador**

### 3.4 Simulação

A simulação foi realizada com a finalidade de avaliar o funcionamento do projeto e confirmar seu funcionamento a uma distância de 100 metros. Para isso, foram utilizados o controle remoto, o dispositivo de controle com o acionador, um analisador de espectro para realizar as medições de nível de sinal, um veículo com hodômetro parcial para medir a distância entre o controle remoto e o acionador e, por último, um inversor para fornecer energia elétrica para alimentar o controle remoto.

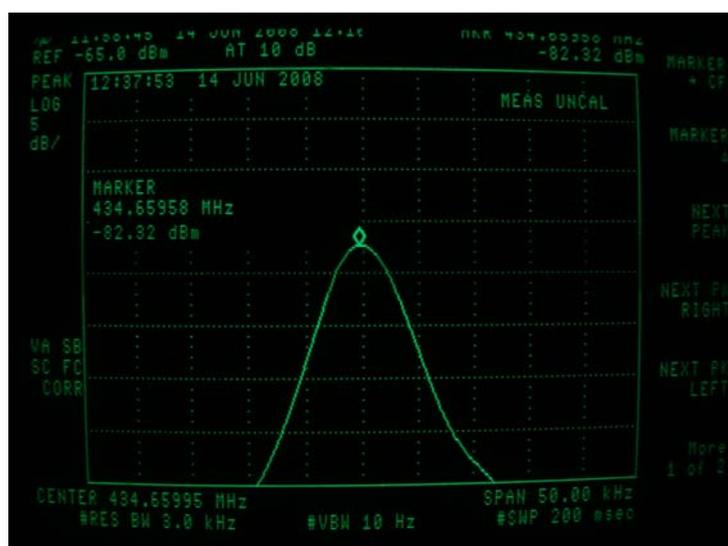
O analisador de espectro ficou instalado junto ao dispositivo de controle, o controle remoto e o inversor foram instalados no veículo.

A primeira medição foi realizada a uma distância de cinco metros. Nessa condição o projeto funcionou corretamente e o nível de sinal recebido foi -24,83dBm. A figura 3.52 apresenta a medição realizada no analisador de espectro.



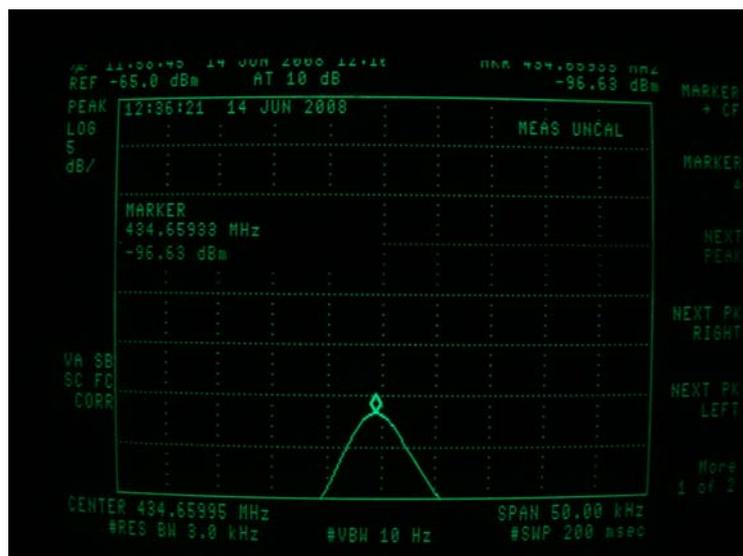
**Figura 3.52 – Recepção a cinco metros do controle remoto**

Em seguida o controle remoto foi deslocado para 100 metros distante do dispositivo de controle. A essa distância o nível de recepção caiu para -82,32dBm, como pode ser visto na figura 3.53, mesmo assim o funcionamento apresentou-se normal.



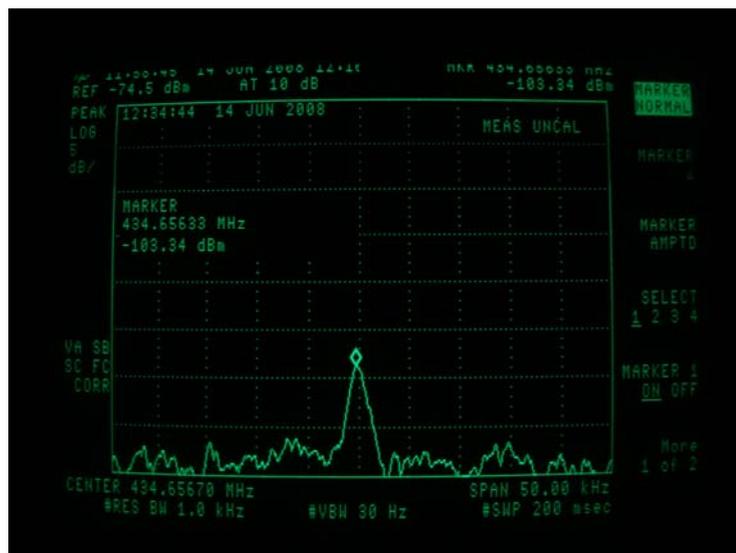
**Figura 3.53 – Recepção a 100m do controle remoto**

A título de conhecimento, foram realizadas outras duas medições. A primeira foi relativa ao limiar de funcionamento tanto para o nível de recepção quanto para a distância. Como resultado, a uma distância de 300 metros o nível de recepção foi de -96,63dBm, como pode ser comprovado na figura 3.54, e o projeto continuou funcionando corretamente.



**Figura 3.54 – Recepção a 300m do controle remoto**

A segunda medição realizada definiu o nível e a distância para os quais o projeto não mais funcionou. Como visto na figura 3.55 o nível de recepção medido foi de -103,34dBm, já a distância medida foi de 400 metros.



**Figura 3.55 – Recepção a 400m do controle remoto**

Todos estes testes foram realizados em campo aberto, sendo que o analisador de espectro e o acionador ficaram próximo à janela de um apartamento no primeiro andar e o controle remoto foi deslocado em linha reta.

## Capítulo 4. Conclusão

Este projeto realiza o acionamento remoto de uma lâmpada e uma sirene por meio de contato seco de relés. Para que seja acionada uma quantidade maior de lâmpadas ou circuitos elétricos basta adequar o *software* e o *hardware* deste projeto ao limite de portas disponíveis no microcontrolador utilizado. Durante a implementação deste projeto surgiram problemas dos quais alguns foram superados e outros não.

Um problema resolvido foi quanto à dificuldade de gravação do microcontrolador, mesmo com um gravador novo, ocorriam mensagens de erro na gravação. A solução surgiu após a verificação de mau contato na placa do gravador, todas as soldas foram refeitas e o funcionamento voltou ao normal.

Com a análise do sinal transmitido foi verificado um excesso de ruído na portadora do rádio, afetando a comunicação. Como solução foi trocada a fonte de alimentação e utilizado um estabilizador.

Uma questão não superada foi a instabilidade apresentada quando o receptor não detecta sinal proveniente do transmissor. Neste caso o transceptor entrega ao microcontrolador um sinal interferente, este sinal gera falsos comandos de acionamento causando instabilidade no dispositivo acionador. Por esse motivo não foi possível executar a inversão do sentido de transmissão, que por sua vez inviabilizou a retro-alimentação. Contudo, após a instalação das chaves seletoras de sentido de transmissão, a retro-alimentação foi simulada.

Concluído o desenvolvimento, o projeto apresenta o seguinte funcionamento:

O controle remoto faz a leitura dos botões e transmite a informação necessária ao dispositivo de controle através de radiofrequência. O dispositivo de controle, por sua vez, recebe as informações transmitidas pelo controle remoto, interpreta e comanda o dispositivo acionador. Este, por sua vez, recebe os comandos do dispositivo de controle e acende a lâmpada ou liga a sirene além de retornar ao dispositivo de controle o *status* do acionamento.

Ficam aqui sugestões para projetos futuros baseados nesta implementação:

- Dispositivo para transmissão de canal de voz;
- Transmissão de dados a uma taxa de até 10Kbps;
- Implementar uma rede de sensores sem fio.

## Referências Bibliográficas

- [1]- ALVES, Luiz. **Comunicação de Dados**. São Paulo: Makron, McGraw-Hill, 1992.
- [2]- MARTINS, Nardênio Almeida. **Sistemas Microcontrolados**. 1ª Ed. São Paulo: Novatec, 2005.
- [3]- MALVINO, Albert Paul. **Eletrônica Volume 1**. 2ª Ed. São Paulo: McGraw-Hill, 1987.
- [4]- PINES, José. BARRADAS, Ovídio César Machado. **Telecomunicações Sistemas Multiplex**. 2ª Ed. Rio de Janeiro: Livros Técnicos e Científicos, Embratel, 1978.
- [5]- SILVEIRA, Jorge Luis da. **Comunicação de Dados e Sistemas de Teleprocessamento**. São Paulo: Makro Books, McGraw-Hill, 1991.
- [6]- SOUZA, David José de. **Desbravando o PIC**. 11ª Ed. São Paulo: Érica, 2007.
- [7]- SOUZA, David José de. LAVINIA, Nicolas César. **Conectando o PIC**. 2ª Ed. São Paulo: Érica, 2003.
- [8]- ZANCO, Wagner da Silva. **Microcontroladores PIC16F628A/648A**. 1ª Ed. São Paulo: Érica, 2005.
- [9]- NiM2 *transceiver data sheet*.
- [10]- <http://www.microchip.com> Acessado em: 18 de agosto de 2007 e em 10 fevereiro de 2008.

## Apêndice A – Código fonte do controle remoto

```

*****
;
.*          ACIONAMENTO REMOTO DE LÂMPADAS          *
;
*****
;
.*          *
;
.*          DESENVOLVIDO POR RA2021839/0          *
;
.*          *
;
*****
;
.*          DESCRIÇÃO DO ARQUIVO          *
;
.*          *
;
.*          CONTROLE REMOTO          *
;
.*          ENVIA COMANDO PARA O ACIONAMENTO          *
;
.*          REMOTO DE UMA LÂMPADA E/OU UMA SIRENE          *
;
.*          *
;
*****
;
*****
;
.*          ARQUIVOS DE DEFINIÇÕES          *
;
*****
;
#INCLUDE <P16F628A.INC>          ;ARQUIVO PADRÃO MICROCHIP PARA 16F628A
          __CONFIG _BODEN_ON & _CP_OFF & _PWRTE_ON & _WDT_OFF & _LVP_OFF
& _MCLRE_ON & _XT_OSC

*****
;
.*          PAGINAÇÃO DE MEMÓRIA          *
;
*****
;DEFINIÇÃO DE COMANDOS DE USUÁRIO PARA ALTERAÇÃO DA
;PÁGINA DE MEMÓRIA
#DEFINE      BANK0 BCF STATUS,RP0          ;SETA BANK0 DE MEMÓRIA
#DEFINE      BANK1 BSF STATUS,RP0          ;SETA BANK1 DE MAMÓRIA
*****
;
.*          VARIÁVEIS          *
;
*****
;DEFINIÇÃO DOS NOMES E ENDEREÇOS DE TODAS AS VARIÁVEIS
;UTILIZADAS PELO SISTEMA

          CBLOCK      0x20          ;ENDEREÇO INICIAL DA MEMÓRIA
;DE USUÁRIO
          CONT1          ;CONTADOR PARA ROTINA DE DELAY
          CONT2          ;CONTADOR PARA ROTINA DE DELAY

```

```

        FLAGS                ;ARMAZENA OS FLAGS DE CONTROLE
        FILTRO1              ;FILTRAGEM PARA O BOTÃO 1
        FILTRO2              ;FILTRAGEM PARA O BOTÃO 2
        DADO_RECEBIDO        ;ARMAZENA DADO RECEBIDO
        TX_CMD               ;ARMAZENA DADO PARA TRANSMISSÃO
    ENDC                    ;FIM DO BLOCO DE MEMÓRIA

;*****
;
;*                          FLAGS INTERNOS                          *
;
;*****
; DEFINIÇÃO DE TODOS OS FLAGS UTILIZADOS PELO SISTEMA
#DEFINE    ST_BT1          FLAGS,0          ;STATUS DO BOTÃO 1
#DEFINE    ST_BT2          FLAGS,1          ;STATUS DO BOTÃO 2
#DEFINE    ST_LPD          FLAGS,2          ;STATUS DA LÂMPADA
#DEFINE    ST_SNE          FLAGS,3          ;STATUS DA SIRENE
#DEFINE    DADO_NOVO       FLAGS,4          ;DADO NOVO NA USART

;*****
;
;*                          CONSTANTES                              *
;
;*****
; DEFINIÇÃO DE TODAS AS CONSTANTES UTILIZADAS PELO SISTEMA

T_FILTRO   EQU    .255    ;FILTRO PARA BOTÃO E CONTADOR

;*****
;
;*                          ENTRADAS                                *
;
;*****
; DEFINIÇÃO DE TODOS OS PINOS QUE SERÃO UTILIZADOS COMO ENTRADA
#DEFINE    BOTAO1          PORTA,2          ;PORTA DO BOTÃO1
                                                ; 0 -> PRESSIONADO
                                                ; 1 -> LIBERADO
#DEFINE    BOTAO2          PORTA,3          ;PORTA DO BOTÃO2
                                                ; 0 -> PRESSIONADO
                                                ; 1 -> LIBERADO

;*****
;
;*                          SAÍDAS                                  *
;
;*****
; DEFINIÇÃO DE TODOS OS PINOS QUE SERÃO UTILIZADOS COMO ;SAÍDA
#DEFINE    LED1            PORTA,0          ;STATUS DA LÂMPADA
#DEFINE    LED2            PORTA,1          ;STATUS DA SIRENE

```

```

;*****
;
;*                               *
;                               VETOR DE RESET
;*****
;
;   ORG    0x00                ;ENDEREÇO INICIAL DE
;                               ;PROCESSAMENTO
;
;   GOTO   INICIO
;*****
;
;*                               *
;                               INÍCIO DA INTERRUPTÃO
;*****
; AS INTERRUPTÕES NÃO SÃO UTILIZADAS.
;
;   ORG    0x04                ;ENDEREÇO INICIAL DA INTERRUPTÃO
;   RETFIE                ;RETORNA DA INTERRUPTÃO
;*****
;
;*                               *
;                               USART
;*****
;RB1 -> RX
;RB2 -> TX
;RB4 -> TXSELECT
;RB5 -> RXSELECT
;RX HABILITADO
;
;RXSELECT -> 0
;TXSELECT -> 1
;TX HABILITADO
;
;RXSELECT -> 1
;TXSELECT -> 0
;
;#DEFINE    TXSELECT    PORTB,4    ;TXSELECT
;
;0 -> HABILITA TX
;1 -> DESABILITA TX
;
;#DEFINE    RXSELECT    PORTB,5    ;RXSELECT
;
;0 -> HABILITA RX
;1 -> DESABILITA RX
;*****
;
;*                               *
;                               INICIO DO PROGRAMA
;*****
;
; INICIO
;
;   BANK1                ;ALTERA PARA O BANCO 1
;
;   MOVLW    B'00001100'
;   MOVWF   TRISA        ;RA2 E 3 ENTRADA DEMAIS SAÍDAS
;
;   MOVLW    B'00000110'
;   MOVWF   TRISB        ;DEFINE TODO O PORTB COMO SAÍDA

```

```

        MOVLW      B'10000000'
        MOVWF     OPTION_REG      ;PRESCALER 1:2 NO TMR0
                                   ;PULL-UPS DESABILITADOS
                                   ;DEMAIS CONFIG. IRRELEVANTES

        MOVLW      B'00000000'
        MOVWF     INTCON          ;INTERRUPÇÕES DESLIGADAS
        BANK0
        MOVWF     CMCON           ;COMPARADOR ANALÓGICO
                                   ;COMPARADORES DESLIGADOS

;*****
;
;*          INICIALIZAÇÃO DAS VARIÁVEIS          *
;*****
;
        CLRF     PORTA           ;LIMPA O PORTA
        CLRF     PORTB          ;LIMPA O PORTB
        CLRF     FLAGS          ;LIMPA TODOS OS FLAGS

;*****
;
;*          ROTINA PRINCIPAL                      *
;*****
;
MAIN
        MOVLW     T_FILTRO
        MOVWF     FILTRO1       ;INICIALIZA FILTRO1 = T_FILTRO
        MOVWF     FILTRO2       ;INICIALIZA FILTRO2 = T_FILTRO

;*****
;
;*          CHECAGEM DOS BOTÕES                  *
;*****
;
CHECA_BT1
        BTFSC    BOTAO1        ;CHECAGEM DO BOTÃO 1
                                   ;BOTÃO 1 PRESSIONADO?
        GOTO     BT1_LIB       ;NÃO. MARCA BT1 LIBERADO
        DECFSZ   FILTRO1,F     ;SIM, DECREMENTA CONTADOR DO BT1
                                   ;CONTADOR ZERADO?
        GOTO     CHECA_BT1     ;NÃO, CONTINUA ESPERANDO
        BTFSS    ST_BT1        ;SIM, BOTÃO JÁ ESTAVA PRESSIONADO?
        GOTO     ACAO_BOTAO1   ;NÃO, EXECUTA AÇÃO DO BOTÃO1
        GOTO     CHECA_BT2     ;SIM, CHECA BOTÃO 2

BT1_LIB
        BCF      ST_BT1        ;MARCA BOTÃO 1 LIBERADO

CHECA_BT2
        BTFSC    BOTAO2        ;CHECAGEM DO BOTÃO 2
                                   ;BOTÃO 2 PRESSIONADO?
        GOTO     BT2_LIB       ;NÃO. MARCA LIBERADO

```

```

DECFSZ    FILTRO2,F          ;SIM, DECREMENTA DO BT2
          ;CONTADOR ZERADO?

GOTO      CHECA_BT2         ;NÃO, CONTINUA ESPERANDO
BTFSS     ST_BT2            ;SIM , BOTÃO JÁ ESTAVA PRESSIONADO?
GOTO      ACAO_BOTAO2       NÃO, EXECUTA AÇÃO DO BOTÃO2
GOTO      HAB_RX            ;SIM, ROTINA DE RECEPÇÃO DE DADOS

BT2_LIB
BCF       ST_BT2           ;MARCA BOTÃO 2 LIBERADO
GOTO      HAB_RX           ;ROTINA DE RECEPÇÃO DE DADOS

.*****
,*
;          AÇÃO DOS BOTÕES          *
,*
.*****

ACAO_BOTAO1
BANK0    ;ALTERA PARA BANCO 0
BSF      ST_BT1           ;MARCA BOTÃO 1 PRESSIONADO
BTFSC    ST_LPD           ;LÂMPADA LIGADA?
GOTO     APL              ;SIM, MARCA LÂMPADA DESLIGADA
BSF      ST_LPD           ;NÃO. MARCA LÂMPADA LIGADA
MOVLW    A'U'             ;ARMAZENA CARACTERE PARA
MOVWF    TX_CMD           ;LIGAR A LÂMPADA
CALL     TX               ;CHAMA ROTINA DE TRANSMISSÃO
GOTO     CHECA_BT2        ;CHECAGEM DO BOTÃO 2

APL
BANK0
BCF      ST_LPD           ;MARCA LÂMPADA DESLIGADA
MOVLW    A'X'             ;ARMAZENA CARACTERE PARA
MOVWF    TX_CMD           ;DESLIGAR A LÂMPADA
CALL     TX               ;CHAMA ROTINA DE TRANSMISSÃO
GOTO     CHECA_BT2        ;CHECAGEM DO BOTÃO 2

ACAO_BOTAO2
BANK0    ;ALTERA PARA BANCO 0
BSF      ST_BT2           ;MARCA BOTÃO 2 PRESSIONADO
BTFSC    ST_SNE           ;SIRENE LIGADA?
GOTO     ASNE             ;SIM, MARCA SIRENE DESLIGADA
BSF      ST_SNE           ;NÃO MARCA SIRENE LIGADA
MOVLW    A'Z'             ;ARMAZENA CARACTERE PARA
MOVWF    TX_CMD           ;LIGAR A SIRENE
CALL     TX               ;CHAMA ROTINA DE TRANSMISSÃO
GOTO     HAB_RX           ;ROTINA DE RECEPÇÃO DE DADOS

ASNE

```

```

BANK0
BCF      ST_SNE      ;MARCA SIRENE DESLIGADA
MOVLW   A'V'        ;ARMAZENA CARACTERE PARA
MOVWF   TX_CMD      ;DESLIGAR A SIRENE
CALL    TX          ;CHAMA ROTINA DE TRANSMISSÃO

.*****
;
;*          ROTINA DE RECEPÇÃO DE DADOS          *
;*****
;
HAB_RX

BANK1      ;ALTERA PARA O BANCO 1
BSF      TXSTA,BRGH ;ATIVA ALTA VELOCIDADE
MOVLW   .51        ;BAUND RATE 4800 bps
MOVWF   SPBRG
BCF      TXSTA,SYNC ;ATIVA COMUNICAÇÃO ASSÍNCRONA
BANK0      ;ALTERA PARA O BANCO 0
BSF      RCSTA,SPEN ;HABILITA A PORTA SERIAL
BSF      RCSTA,CREN ;HABILITA A RECEPÇÃO
BTFSC   PIR1,RCIF  ;RECEPÇÃO OK?
GOTO    $+3        ;SIM, ARMAZENA COMANDO RECEBIDO
BTFSC   RCSTA,FERR ;NÃO
GOTO    MAIN       ;NÃO, ROTINA PRINCIPAL
MOVF    RREG,W     ;SIM, ARMAZENA COMANDO RECEBIDO
MOVWF   DADO_RECEBIDO ;ARMAZENA O COMANDO RECEBIDO
BSF     DADO_NOVO  ;SETA FLAG DADO NOVO

.*****
;
;*          ROTINA DE ACIONAMENTO          *
;*****
;
ACIONADOR

BTFSS   DADO_NOVO  ;FLAG DADO_NOVO SETADO?
GOTO    MAIN       ;NÃO. ROTINA PRINCIPAL
BCF     DADO_NOVO  ;SIM, RESETA FLAG DADO_NOVO
MOVLW   A'U'      ;COMPARA DADO RECEBIDO COM
XORWF   DADO_RECEBIDO,W ;CARACTERE U
BTFSS   STATUS,Z   ;DADO_RECEBIDO = U
GOTO    $+3        ;NÃO, SALTA TRÊS LINHAS
BSF     LED1       ;SIM, LIGUE A LÂMPADA
GOTO    HAB_RX     ;ROTINA DE RECEPÇÃO DE DADOS
MOVLW   A'X'      ;COMPARA DADO RECEBIDO COM
XORWF   DADO_RECEBIDO,W ;CARACTERE X
BTFSS   STATUS,Z   ;DADO_RECEBIDO = X ?

```

```

GOTO      $+3           ;NÃO, SALTA TRÊS LINHAS
BCF       LED1          ;SIM, APAGUE A LÂMPADA
GOTO      HAB_RX        ;ROTINA DE RECEPÇÃO DE DADOS
MOVLW    A'Z'           ;COMPARA DADO RECEBIDO COM
XORWF    DADO_RECEBIDO,W ;CARACTERE Z
BTFSS    STATUS,Z      ;DADO_RECEBIDO = Z ?
GOTO      $+3           ;NÃO, SALTA TRÊS LINHAS
BSF      LED2           ;SIM, LIGUE A SIRENE
GOTO      HAB_RX        ;ROTINA DE RECEPÇÃO DE DADOS
MOVLW    A'V'           ;COMPARA DADO RECEBIDO COM
XORWF    DADO_RECEBIDO,W ;CARACTERE V
BTFSS    STATUS,Z      ;DADO_RECEBIDO = V ?
GOTO      $+2           ;NÃO, SALTA DUAS LINHAS
BCF      LED2           ;SIM, DESLIGUE A SIRENE
GOTO      MAIN          ;ROTINA PRINCIPAL

```

```

;*****
;

```

```

;*      ROTINA DE TRANSMISSÃO DE DADOS      *
;

```

```

;*****
;

```

```
TX
```

```

BANK1           ;ALTERA PARA O BANCO 1
BSF      TXSTA,BRGH ;ATIVA ALTA VELOCIDADE DE
           ;TRANSFERÊNCIA
MOVLW    .51      ;CONFIG. BAUND RATE PARA 4800 bps
MOVWF    SPBRG
BCF      TXSTA,SYNC ;ATIVA COMUNICAÇÃO ASSÍNCRONA
BANK0           ;ALTERA PARA O BANCO 0
BSF      RCSTA,SPEN ;HABILITA A PORTA SERIAL
MOVLW    0xAA     ;INICIA TRANSMISSÃO DO PREÂMBULO
MOVWF    TXREG
BANK1           ;ALTERA PARA O BANCO 1
BSF      TXSTA,TXEN ;HABILITA A TRANSMISSÃO
BANK0           ;ALTERA PARA O BANCO 0
MOVWF    TXREG
CALL     DL       ;CHAMA ROTINA DE DELAY
MOVLW    0x00
MOVWF    TXREG
MOVWF    TXREG
CALL     DL       ;CHAMA ROTINA DE DELAY
MOVLW    0xFF
MOVWF    TXREG

```



## Apêndice B – Código fonte do dispositivo de controle

```

*****
;
.*          ACIONAMENTO REMOTO DE LÂMPADAS          *
;
*****
;
.*          *
;
.*          DESENVOLVIDO POR RA2021839/0          *
;
.*          *
;
*****
.*          DESCRIÇÃO DO ARQUIVO                    *
;
.*          *
.*          DISPOSITIVO DE CONTROLE                *
;
.*          RECEBE COMANDO PARA O ACIONAMENTO      *
;
.*          REMOTO DE UMA LÂMPADA E/OU UMA SIRENE  *
;
.*          ENCAMINHA O COMANDO E ENVIA O STATUS  *
;
.*          DO ACIONAMENTO                          *
;
.*          *
;
*****
;
*****
;
.*          ARQUIVOS DE DEFINIÇÕES                  *
;
*****
#INCLUDE <P16F628A.INC>          ;ARQUIVO PADRÃO MICROCHIP PARA 16F628A
      _CONFIG _BODEN_ON & _CP_OFF & _PWRTE_ON & _WDT_OFF & _LVP_OFF
& _MCLRE_ON & _XT_OSC
;
*****
.*          PAGINAÇÃO DE MEMÓRIA                    *
;
*****
;DEFINIÇÃO DE COMANDOS DE USUÁRIO PARA ALTERAÇÃO DA
;PÁGINA DE MEMÓRIA
#DEFINE     BANK0 BCF STATUS,RP0      ;SETA BANK0 DE MEMÓRIA
#DEFINE     BANK1 BSF STATUS,RP0      ;SETA BANK1 DE MAMÓRIA
;
*****
.*          VARIÁVEIS                               *
;
*****
;DEFINIÇÃO DOS NOMES E ENDEREÇOS DE TODAS AS VARIÁVEIS
;UTILIZADAS PELO SISTEMA
      CBLOCK      0x20                ;ENDEREÇO INICIAL DA MEMÓRIA
;DE USUÁRIO
      CONT1                ;CONTADOR PARA ROTINA DE DELAY

```

```

CONT2                ;CONTADOR PARA ROTINA DE DELAY
FLAGS                ;ARMAZENA OS FLAGS DE CONTROLE
FILTRO1              ;FILTRAGEM PARA O RELÉ 1
FILTRO2              ;FILTRAGEM PARA O RELÉ 2
DADO_RECEBIDO        ;ARMAZENA DADO RECEBIDO
TX_CMD               ;ARMAZENA DADO PARA TRANSMISSÃO
ENDC                 ;FIM DO BLOCO DE MEMÓRIA

;*****
;
;*                   FLAGS INTERNOS                   *
;
;*****
;DEFINIÇÃO DE TODOS OS FLAGS UTILIZADOS PELO SISTEMA
#DEFINE ST_RL1      FLAGS,0      ;STATUS DO RELÉ 1
#DEFINE ST_RL2      FLAGS,1      ;STATUS DO RELÉ 2
#DEFINE ST_LPD      FLAGS,2      ;STATUS DA LÂMPADA
#DEFINE ST_SNE      FLAGS,3      ;STATUS DA SIRENE
#DEFINE DADO_NOVO   FLAGS,4      ;DADO NOVO NA USART

;*****
;
;*                   CONSTANTES                       *
;
;*****
;DEFINIÇÃO DE TODAS AS CONSTANTES UTILIZADAS PELO SISTEMA

T_FILTRO EQU .255 ;FILTRO PARA BOTÃO E CONTADOR

;*****
;
;*                   ENTRADAS                         *
;
;*****
;DEFINIÇÃO DE TODOS OS PINOS QUE SERÃO UTILIZADOS COMO ENTRADA
#DEFINE RELE1      PORTA,2      ;PORTA DO RELÉ 1
                                ; 0 -> ACIONADO
                                ; 1 -> LIBERADO
#DEFINE RELE2      PORTA,3      ;PORTA DO RELÉ 2
                                ; 0 -> ACIONADO
                                ; 1 -> LIBERADO

;*****
;
;*                   SAÍDAS                           *
;
;*****
;DEFINIÇÃO DE TODOS OS PINOS QUE SERÃO UTILIZADOS COMO SAÍDA
#DEFINE LED1 PORTA,0      ;PORTA DA LÂMPADA
#DEFINE LED2 PORTA,1      ;PORTA DA SIRENE

```

```

*****
;
.*
;          VETOR DE RESET          *
;
*****
;
          ORG          0x00          ;ENDEREÇO INICIAL DE PROCESSAMENTO
          GOTO          INICIO

*****
;
.*
;          INÍCIO DA INTERRUPÇÃO          *
;
*****
; AS INTERRUPÇÕES NÃO SÃO UTILIZADAS.
          ORG          0x04          ;ENDEREÇO INICIAL DA INTERRUPÇÃO
          RETFIE          ;RETORNA DA INTERRUPÇÃO

*****
;
.*
;          USART          *
;
*****
;RB1 -> RX
;RB2 -> TX
;RB4 -> TXSELECT
;RB5 -> RXSELECT
;RX HABILITADO
;RXSELECT -> 0
;TXSELECT -> 1

;TX HABILITADO
;RXSELECT -> 1
;TXSELECT -> 0
#DEFINE      TXSELECT      PORTB,4      ;TXSELECT
; 0 -> HABILITA TX
; 1 -> DESABILITA TX
#DEFINE      RXSELECT      PORTB,5      ;RXSELECT
; 0 -> HABILITA RX
; 1 -> DESABILITA RX

*****
;
.*
;          INICIO DO PROGRAMA          *
;
*****
INICIO
          BANK1          ;ALTERA PARA O BANCO 1
          MOVLW          B'00001100'
          MOVWF          TRISA          ;RA2 E 3 ENTRADA DE MAIS SAÍDAS
          MOVLW          B'00000110'

```

```

MOVWF    TRISB                ;DEFINE TODO O PORTB COMO SAÍDA
MOVLW    B'10000000'
MOVWF    OPTION_REG          ;PRESCALER 1:2 NO TMR0
                                           ;PULL-UPS DESABILITADOS
                                           ;DEMAIS CONFG. IRRELEVANTES

MOVLW    B'00000000'
MOVWF    INTCON              ;INTERRUPÇÕES DESLIGADAS
BANK0
MOVWF    CMCON               ;COMPARADOR ANALÓGICO
                                           ;COMPARADORES DESLIGADOS

;*****
;
;*          INICIALIZAÇÃO DAS VARIÁVEIS          *
;
;*****

CLRf     PORTA                ;LIMPA O PORTA
CLRf     PORTB                ;LIMPA O PORTB
CLRf     FLAGS                ;LIMPA TODOS OS FLAGS

;*****
;
;*          ROTINA PRINCIPAL                    *
;
;*****

MAIN

MOVLW    T_FILTRO
MOVWF    FILTRO1              ;INICIALIZA FILTRO1 = T_FILTRO
MOVWF    FILTRO2              ;INICIALIZA FILTRO2 = T_FILTRO

;*****
;
;*          ROTINA DE RECEPÇÃO DE DADOS        *
;
;*****

HAB_RX

BANK1                ;ALTERA PARA O BANCO 1
BSF      TXSTA,BRGH    ;ATIVA ALTA VELOCIDADE
MOVLW    .51          ;BAUND RATE 4800 bps
MOVWF    SPBRG
BCF      TXSTA,SYNC    ;ATIVA COMUNICAÇÃO ASSÍNCRONA
BANK0
BSF      RCSTA,SPEN    ;HABILITA A PORTA SERIAL
BSF      RCSTA,CREN    ;HABILITA A RECEPÇÃO
BTFSC   PIR1,RCIF     ;RECEPÇÃO OK?
GOTO    $+3           ;SIM, ARMAZENA COMANDO RECEBIDO
BTFSC   RCSTA,FERR    ;NAO
GOTO    MAIN          ;NÃO, ROTINA PRINCIPAL

```

```

MOVF      RCREG,W          ;SIM, ARMAZENA COMANDO
MOVWF     DADO_RECEBIDO    ;RECEBIDO
BSF       DADO_NOVO        ;SETA FLAG DADO_NOVO

```

```

;*****
;

```

```

;*          ROTINA DE ACIONAMENTO          *
;

```

```

;*****
;

```

#### ACIONADOR

```

BTFSS     DADO_NOVO        ;FLAG DADO_NOVO SETADO?
GOTO      MAIN             ;NÃO. ROTINA PRINCIPAL
BCF       DADO_NOVO        ;SIM, RESETA FLAG DADO_NOVO
MOVLW     A'U'             ;COMPARA DADO RECEBIDO COM
XORWF     DADO_RECEBIDO,W  ;CARACTERE U
BTFSS     STATUS,Z         ;DADO_RECEBIDO = U
GOTO      $+3              ;NÃO, SALTA TRÊS LINHAS
BSF       LED1             ;SIM, LIGUE A LÂMPADA
GOTO      HAB_RX           ;ROTINA DE RECEPÇÃO DE DADOS
MOVLW     A'X'             ;COMPARA DADO RECEBIDO COM
XORWF     DADO_RECEBIDO,W  ;CARACTERE X
BTFSS     STATUS,Z         ;DADO_RECEBIDO = X ?
GOTO      $+3              ;NÃO, SALTA TRÊS LINHAS
BCF       LED1             ;SIM, APAGUE A LÂMPADA
GOTO      HAB_RX           ;ROTINA DE RECEPÇÃO DE DADOS
MOVLW     A'Z'             ;COMPARA DADO RECEBIDO COM
XORWF     DADO_RECEBIDO,W  ;CARACTERE Z
BTFSS     STATUS,Z         ;DADO_RECEBIDO = Z ?
GOTO      $+3              ;NÃO, SALTA TRÊS LINHAS
BSF       LED2             ;SIM, LIGUE A SIRENE
GOTO      HAB_RX           ;ROTINA DE RECEPÇÃO DE DADOS
MOVLW     A'V'             ;COMPARA DADO RECEBIDO COM
XORWF     DADO_RECEBIDO,W  ;CARACTERE V
BTFSS     STATUS,Z         ;DADO_RECEBIDO = V ?
GOTO      $+2              ;NÃO, SALTA DUAS LINHAS
BCF       LED2             ;SIM, DESLIGUE A SIRENE

```

```

;*****
;

```

```

;*          CHECAGEM DOS RELÉS          *
;

```

```

;*****
;

```

```

CHECA_RL1          ;CHECAGEM DO RELÉ 1
      BTFSC     RELE1      ;RELÉ 1 ACIONADO?

```

```

GOTO      APL      ;NÃO, RL1 LIBERADO
DECFSZ    FILTRO1,F ;SIM, DECREMENTA CONTADOR DO RL1
GOTO      $-1      ;NÃO, CONTINUA ESPERANDO
BTFSS     RELE1    ;SIM, RELÉ 1 LIBERADO?
GOTO      LPD      ;NÃO RL1 ACIONADO
GOTO      MAIN     ;SIM, ROTINA PRINCIPAL
CHECA_RL2
BTFSS     RELE2    ;RELÉ 2 ACIONADO?
GOTO      ASNE     ;NÃO, RL2 LIBERADO
DECFSZ    FILTRO2,F ;SIM, DECREMENTA CONTADOR DO RL2
GOTO      $-1      ;NÃO, CONTINUA ESPERANDO
BTFSS     RELE2    ;SIM, RELÉ 2 LIBERADO?
GOTO      SNE      ;NÃO, RL2 ACIONADO
GOTO      MAIN     ;SIM, ROTINA PRINCIPAL

;*****
;*
;          ACÃO DOS RELÉS
;*
;*****

LPD
;RL1 ACIONADO
BTFSS     ST_LPD   ;LÂMPADA DESLIGADA?
GOTO      CHECA_RL2 ;NÃO, CHECAGEM DO RL2
BSF       ST_LPD   ;SIM. MARCA LÂMPADA LIGADA
MOVLW    A'U'     ;ARMAZENA CARACTERE PARA INDICAR
MOVWF    TX_CMD   ;LÂMPADA LIGADA
CALL     TX        ;CHAMA ROTINA DE TRANSMISSÃO
GOTO      CHECA_RL2 ;CHECAGEM DO RL2

APL
;RL1 LIBERADO
BTFSS     ST_LPD   ;LÂMPADA LIGADA?
GOTO      CHECA_RL2 ;NÃO, CHECAGEM DO RL2
BCF       ST_LPD   ;SIM, MARCA LÂMPADA DESLIGADA
MOVLW    A'X'     ;ARMAZENA CARACTERE PARA INDICAR
MOVWF    TX_CMD   ;LÂMPADA DESLIGADA
CALL     TX        ;CHAMA ROTINA DE TRANSMISSÃO
GOTO      CHECA_RL2 ;CHECAGEM DO RL2

SNE
;RL2 ACIONADO
BTFSS     ST_SNE   ;SIRENE DESLIGADA?
GOTO      MAIN     ;NÃO. ROTINA PRINCIPAL
BSF       ST_SNE   ;SIM. MARCA SIRENE LIGADA
MOVLW    A'Z'     ;ARMAZENA CARACTERE PARA INDICAR
MOVWF    TX_CMD   ;SIRENE LIGADA
CALL     TX        ;CHAMA ROTINA DE TRANSMISSÃO

```

```

        GOTO      MAIN          ;ROTINA PRINCIPAL
ASNE
        BTFSS    ST_SNE        ;SIRENE LIGADA?
        GOTO      MAIN          ;NÃO. ROTINA PRINCIPAL
        BCF      ST_SNE        ;SIM. LIMPA FLAG
        MOVLW    A'V'          ;ARMAZENA CARACTERE PARA INDICAR
        MOVWF    TX_CMD        ;SIRENE DESLIGADA
        CALL     TX            ;CHAMA ROTINA DE TRANSMISSÃO
        GOTO      MAIN          ;ROTINA PRINCIPAL

```

```

;*****
;
;*          ROTINA DE TRANSMISSÃO DE DADOS          *
;
;*****
;

```

```

TX
        BANK1    ;ALTERA PARA O BANCO 1
        BSF      TXSTA,BRGH    ;ATIVA ALTA VELOCIDADE DE
                                ;TRANSFERÊNCIA
        MOVLW    .51           ;CONFIG. BAUND RATE PARA 4800 bps
        MOVWF    SPBRG
        BCF      TXSTA,SYNC    ;ATIVA COMUNICAÇÃO ASSINCRONA
        BANK0    ;ALTERA PARA O BANCO 0
        BSF      RCSTA,SPEN    ;HABILITA A PORTA SERIAL
        MOVLW    0xAA          ;INICIA TRANSMISSÃO DO PREAMBULO
        MOVWF    TXREG
        BANK1    ;ALTERA PARA O BANCO 1
        BSF      TXSTA,TXEN    ;HABILITA A TRANSMISSÃO
        BANK0    ;ALTERA PARA O BANCO 0
        MOVWF    TXREG
        CALL     DL            ;CHAMA ROTINA DE DELAY
        MOVLW    0x00
        MOVWF    TXREG
        MOVWF    TXREG
        CALL     DL            ;CHAMA ROTINA DE DELAY
        MOVLW    0xFF
        MOVWF    TXREG
        MOVWF    TXREG
        CALL     DL            ;CHAMA ROTINA DE DELAY
        MOVLW    0x01
        MOVWF    TXREG        ;TERMINA TRANSMISSÃO DO
        MOVWF    TXREG        ;PREAMBULO
        CALL     DL            ;CHAMA ROTINA DE DELAY

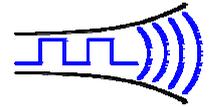
```



## **Anexo – Datasheet do transceptor**

**NEW**

# Radiometrix



27 April 2007

Hartcran House, 231 Kenton Lane, Harrow, HA3 8RP, England  
Tel: +44 (0) 20 8909 9595, Fax: +44 (0) 20 8909 2233

NiM2-xxx.xxx-10

## UHF Narrow Band Transceiver

*The narrow band NiM2 transceiver offers a low power, reliable data link in a Radiometrix transceiver standard pin out and footprint. This makes the NiM2 ideally suited to those low power applications where existing single frequency wideband UHF modules have insufficient range.*



Figure 1: NiM2-434.650-10

### Features

- Conforms to ETSI EN 300 220-3 (radio) and EN 301 489-3 (EMC)
- Standard frequency: 434.65 and 434.075MHz
- Custom frequencies available in 433MHz (EU) band
- Data rates up to 10kbps
- Usable range over 500m
- 25kHz Channel spacing
- Longer range compared to Wide Band FM modules

Available for licence-exempt operation in the 433MHz EU band, the NiM2 modules combine effective screening with internal filtering to minimise spurious radiation and susceptibility thereby ensuring EMC compliance. They can be used in existing low data rate (<10kbps) applications where the operating range of the system using wide band transceivers need to be extended. Because of their small size and low power consumption, NiM2 is ideal for use in battery-powered portable applications.

NiM2 is also available as separate NiM2T transmitter and NiM2R receiver, which can be used as dual-in-line equivalents of NTX2 transmitter and NRX2 receiver respectively.

### Applications

- EPOS equipment, barcode scanners
- Data loggers
- Industrial telemetry and telecommand
- In-building environmental monitoring and control
- High-end security and fire alarms
- DGPS systems
- Vehicle data up/download

### Technical Summary

- 3 stage crystal controlled VCXO
- Data bit rate: 10kbps max.
- Transmit power: +10dBm (10mW)
- Double conversion FM superhet
- SAW band pass filter, image rejection: 50dB
- Data bit rate: 10kbps max.
- Receiver sensitivity: -118dBm (for 12dB SINAD)
- RSSI output with >60dBm range
- Adjacent Channel: -70dBm
- Supply: 2.9V - 15V @ 20 mA transmit, 15mA receive (internal 2.8V voltage regulator)

# NiM2 Single channel transceiver

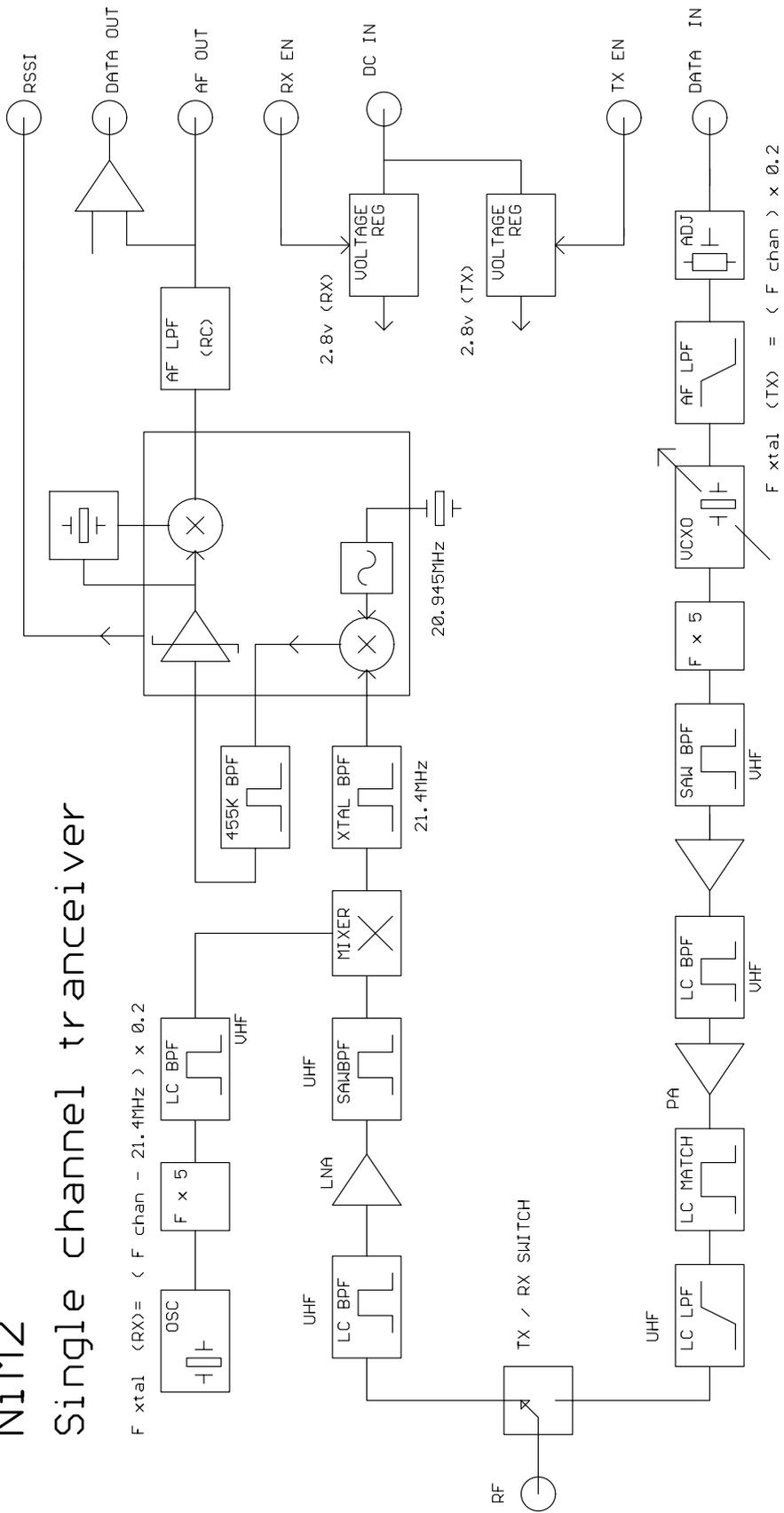


Figure 2: NiM2-434.650-10

## Functional description

The transmit section of the NiM2 consists of a frequency modulated Voltage Controlled Crystal Oscillator (VCXO) feeding a frequency doubler with two stage amplifier and RF filter. Final Power Amplifier stage is factory pre-set to appropriate band power level. Operation is controlled by a Tx Select line, the transmitter achieving full RF output typically within 5ms of this line being pulled low. The RF output is filtered to ensure compliance with the appropriate radio regulations and fed via a fast Tx/Rx changeover switch to the 50Ω antenna pin.

The receive section is a double conversion FM superhet with IF at 21.4MHz and 455kHz fed by a Low Noise Amplifier (LNA) on the RF front-end. The receiver is controlled by RX Select line and will power up typically <2ms. Quadrature detector output is available as Audio Frequency (AF) output and transmitted digital data is regenerated from AF using adaptive data slicer. A Received Signal Strength Indicator (RSSI) output with some 60dB of range is provided.

## User interface

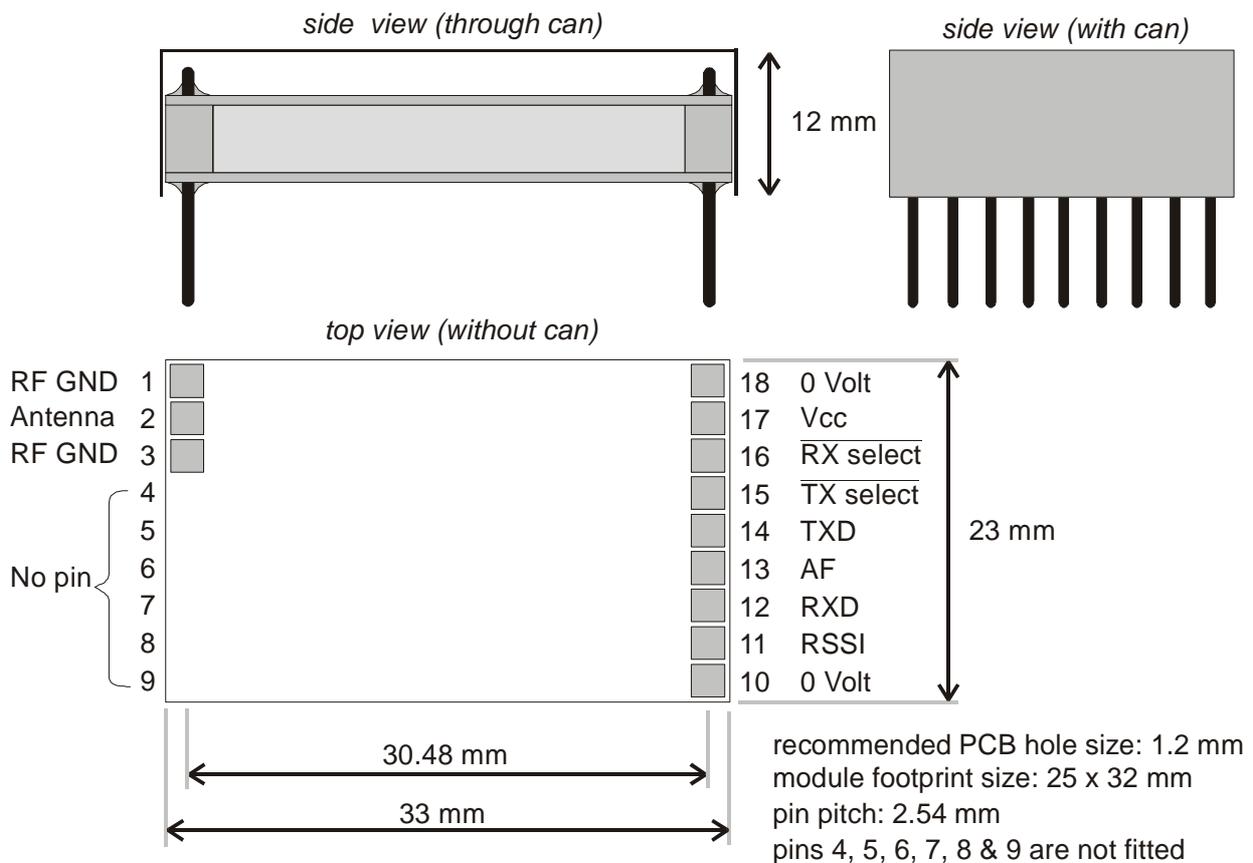


Figure 3: NiM2 pin-out and dimension

NiM2 Pin	Name	Function
1, 3, 9, 10, 18	0V	Ground
17	VCC	2.9 – 15V DC power supply
16	$\overline{RX}$	Pull low to enable Receiver
15	$\overline{TX}$	Pull low to enable Transmitter
14	TXD	DC coupled input for 3V CMOS logic. $R_{in} = 100k\Omega$
13	AF	500mV <sub>pk-pk</sub> audio. DC coupled, approx 0.8V bias
12	RXD	2.5V <sub>pk-pk</sub> logic output of data slicer. Suitable for Biphase codes
11	RSSI	DC level between 0.5V and 2V. 60dB dynamic range

### NOTES:

- $\overline{RX}$  and  $\overline{TX}$  have internal (10kΩ approx.) pull-ups to Vcc
- Avoid  $\overline{RX}$  and  $\overline{TX}$  both low: undefined module operation (but damage will not result)
- RXD is an open collector output, with a 10kΩ pullup to Vcc.
- Pin out is as BiM1, BiM2. On RF connector end only pins 1, 2 and 3 are present.

## Absolute maximum ratings

Exceeding the values given below may cause permanent damage to the module.

Operating temperature	-10°C to +60°C
Storage temperature	-30°C to +70°C
RF in (pin 1)	±50V @ <10MHz, +13dBm @ >10MHz
All other pins	-0.3V to +15.0V

## Performance specifications:

(Vcc = 3V / temperature = 20 °C unless stated)

General	pin	min.	typ.	max.	units	notes
<b>DC supply</b>						
Supply voltage	17	2.9	-	15	V	
TX Supply current (10mW)	17		20		mA	
RX Supply current	17		15		mA	
Antenna pin impedance	2		50		Ω	
RF centre frequency			434.650		MHz	
			434.075		MHz	
Channel spacing			25		kHz	
Number of channels			1			
<b>Transmitter</b>						
<b>RF</b>						
RF power output	2	+9	+10	+11	dBm	1
Spurious emissions	2			-36	dBm	2
Adjacent channel TX power			-37		dBm	
Frequency accuracy		-2.5	0	+2.5	kHz	3
FM deviation (peak)		±2.5	±3.0	±3.5	kHz	4
<b>Baseband</b>						
Modulation bandwidth @ -3dB		0		5	kHz	
TXD input level (logic low)	14		0		V	5
TXD input level (logic high)	14		3.0		V	5
<b>Dynamic timing</b>						
TX select to full RF				5	ms	
<b>Receiver</b>						
<b>RF/IF</b>						
RF sensitivity @ 12dB SINAD	2, 13		-118		dBm	
RF sensitivity @ 1ppm BER	2, 12		-112		dBm	
RSSI range	2, 11		60		dB	6
IF bandwidth			TBA		kHz	
Blocking	2		80		dB	
Image rejection	2		55		dB	
Adjacent channel rejection	2		70		dB	2
Spurious response rejection	2		65		dB	
LO leakage, radiated			-60		dBm	3
<b>Baseband</b>						
Baseband bandwidth @ -3dB	13		5		kHz	
AF level	13		500		mV <sub>P-P</sub>	7
DC offset on AF out	13		0.8		V	
Distortion on recovered AF	12		TBA		%	
Load capacitance, AF / RXD	12,13		TBA		pF	
<b>Dynamic timing</b>						
<b>Power up with signal present</b>						
Power up to stable AF output	16, 13		2			

Power up to stable RXD output	16, 12		10		ms	
<i>Signal applied with supply on</i>						
Signal to valid AF	2, 11		TBD		ms	
Signal to stable data	2, 12		TBD		ms	
Time between data transitions	12			0.1	ms	8
Mark : space ratio	12	20	50	80	%	8

**Notes:**

1. Measured into 50Ω resistive load.
2. Exceeds EN/EMC requirements at all frequencies.
3. Total over full supply and temperature range.
4. With 0V – 3.0V modulation input.
5. To achieve specified FM deviation.
6. See applications information for further details.
7. For received signal with ±3kHz FM deviation.
8. For 50:50 mark to space ratio (i.e. squarewave).

# Applications information

## Power supply requirements

The NiM2 have built-in regulators which deliver a constant 2.8V to the transmitter and the receiver circuitry when the external supply voltage is 2.9V or greater. This ensures constant performance up to the maximum permitted rail, and removes the need for external supply decoupling except in cases where the supply rail is extremely poor (ripple/noise content  $>0.1V_{p-p}$ ).

## TX modulation requirements

The module is factory-set to produce the specified FM deviation with a TXD input to pin 14 of 3V amplitude, i.e. 0V “low”, 3V “high”

If the data input level is greater than 3V, a resistor must be added in series with the TXD input to limit the modulating input voltage to a maximum of around 2V on pin 7. TXD input resistance is 100k $\Omega$  to ground, giving typical required resistor values as follows:

Vcc	Series resistor
$\leq 3V$	-
3.3V	10 k $\Omega$
5V	68k $\Omega$
9V	220k $\Omega$

## RX Received Signal Strength Indicator (RSSI)

The NiM2 wide range RSSI which measures the strength of an incoming signal over a range of 60dB or more. This allows assessment of link quality and available margin and is useful when performing range tests.

The output on pin 11 of the module has a standing DC bias of up to 0.5V (approx.) with no signal, rising to around 2.0V at maximum indication. DVmin-max is typically 1V and is largely independent of standing bias variations. Output impedance is 56k $\Omega$ . Pin 11 can drive a 100 $\mu$ A meter directly, for simple monitoring.

Please note that the actual RSSI voltage at any given RF input level varies somewhat between units. The RSSI facility is intended as a relative indicator only - it is not designed to be, or suitable as, an accurate and repeatable measure of absolute signal level or transmitter-receiver distance. Typical RSSI characteristic is as shown below:

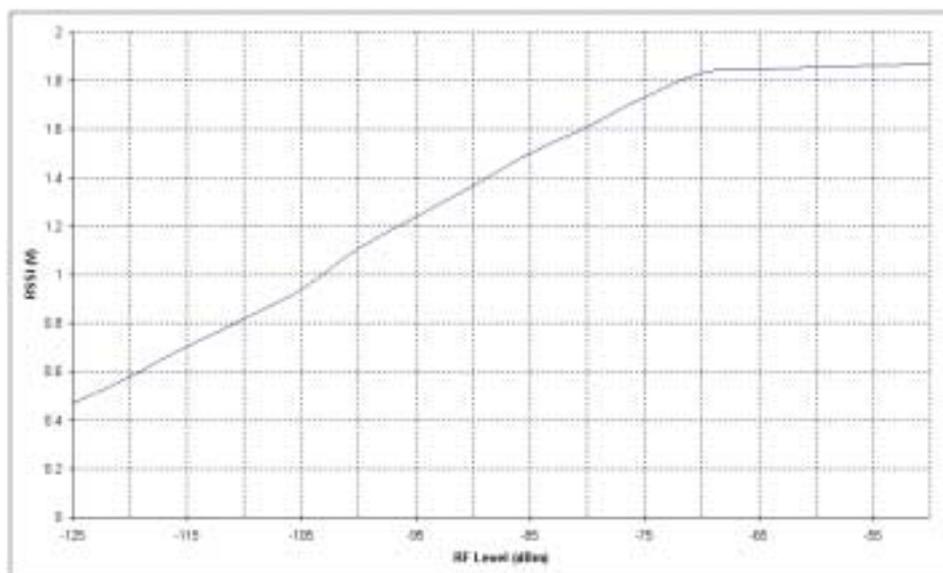


Figure 4: RSSI level with respect to received RF level at NiM2 antenna pin

## Expected range

Predicting the range obtainable in any given situation is notoriously difficult since there are many factors involved. The main ones to consider are as follows:

- Type and location of antennas in use
- Type of terrain and degree of obstruction of the link path
- Sources of interference affecting the receiver
- “Dead” spots caused by signal reflections from nearby conductive objects
- Data rate and degree of filtering employed

## Data formats and range extension

The NiM2 TXD input is normally driven directly by logic levels but will also accept analogue drive (e.g. 2-tone signalling). In this case it is recommended that TXD (pin 14) be DC-biased to 1.2V approx. with the modulation ac-coupled and limited to a maximum of  $2V_{P-P}$  to minimise distortion over the link. The varactor modulator in the NiM2 introduces some 2<sup>nd</sup> harmonic distortion which may be reduced if necessary by predistortion of the analogue waveform. At the other end of the link the NiM2 RXD output is used to drive an external decoder directly.

Although the modulation bandwidth of the NiM2 extends down to DC it is not advisable to use data containing a DC component. This is because frequency errors and drifts between the transmitter and receiver occur in normal operation, resulting in DC offset errors on the NiM2 audio output.

The NiM2 in standard form incorporates a low pass filter with a 5kHz nominal bandwidth. This is suitable for transmission of data at raw bit rates up to 10kbps.

## Antennas

The choice and positioning of transmitter and receiver antennas is of the utmost importance and is the single most significant factor in determining system range. The following notes are intended to assist the user in choosing the most effective antenna type for any given application.

*The following types of integral antenna are in common use:*

**Quarter-wave whip.** This consists simply of a piece of wire or rod connected to the module at one end. At 434MHz the total length should be 164mm from module pin to antenna tip including any interconnecting wire or tracking. Because of the length of this antenna it is almost always external to the product casing.

**Helical.** This is a more compact but slightly less effective antenna formed from a coil of wire. It is very efficient for its size, but because of its high Q it suffers badly from detuning caused by proximity to nearby conductive objects and needs to be carefully trimmed for best performance in a given situation. The size shown in figure 5 below is about the maximum commonly used at 433MHz and appropriate scaling of length, diameter and number of turns can make individual designs much smaller.

**Loop.** A loop of PCB track having an inside area as large as possible (minimum about 4cm<sup>2</sup>), tuned and matched with 2 or 4 capacitors. Loops are relatively inefficient but have good immunity to proximity detuning, so may be preferred in shorter range applications where high component packing density is necessary.

*Integral antenna summary:*

	whip	helical	loop
Ultimate performance	***	**	*
Ease of design set-up	***	**	*
Size	*	***	**
Immunity to proximity effects	**	*	***

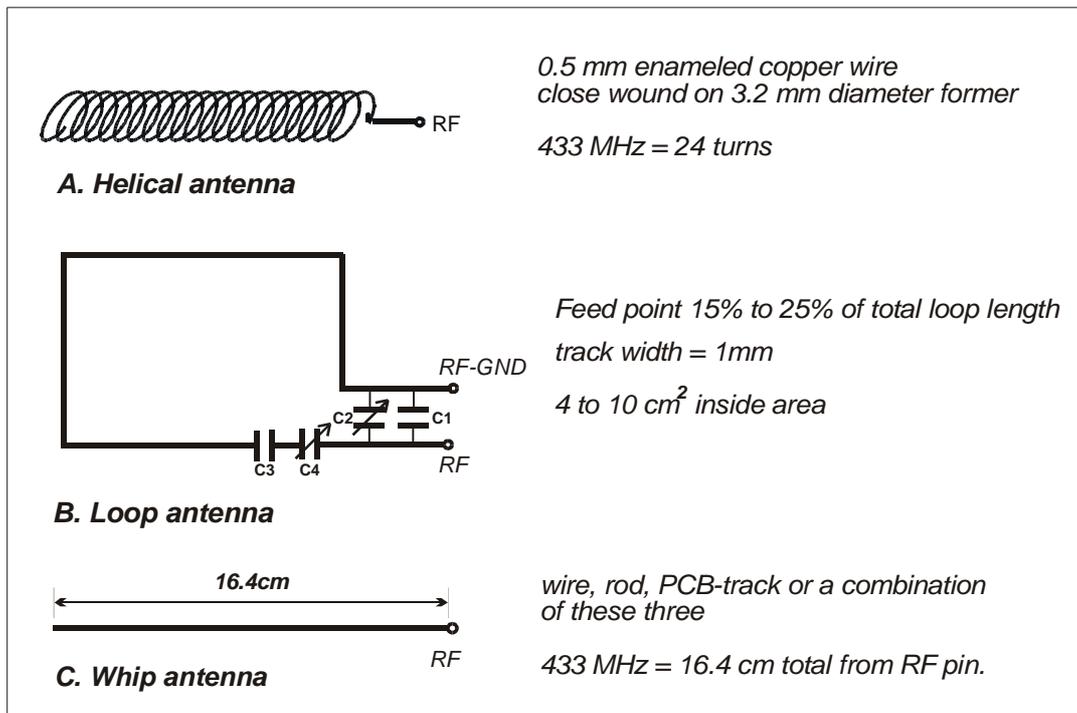


Figure 5: integral antenna configurations

## Packet data

In general, data to be sent via a radio link is formed into a serial "packet" of the form :-

### **Preamble - Control - Address - Data - CRC**

- Where: **Preamble:** This is mandatory for the adaptive data slicer in the receiver in the NiM2 to stabilise. The NiM2 will be stable after 10ms. Additional preamble time may be desired for decoder bit synchronisation, firmware carrier detection or receiver wake up.
- Control:** The minimum requirement is a single bit or unique bit pattern to indicate the start of message (frame sync.). Additionally, decoder information is often placed here such as: packet count, byte count, flow control bits (e.g. ACK, repeat count), repeater control, scrambler information etc.
- Address:** This information is used for identification purposes and would at least contain a 16/24 bit source address, additionally - destination address, site / system code , unit number and repeater address's may be placed here.
- Data:** User data , generally limited to 256 bytes or less (very long packets should be avoided to minimise repeat overheads on CRC failure and channel hogging).
- CRC:** 16/24 Bit CRC or Checksum of control-address-data fields used by the decoder to verify the integrity of the packet.

The exact makeup of the packet depends upon the system requirements and may involve some complex air-traffic density statistics to optimise through-put in large networked systems.

## Networks

NiM2's may be used in many different configurations from simple pair's to multi-node random access networks. The NiM2 is a single frequency device thus in a multi node system the signalling protocol must use Time Division Multiple Access (TDMA). In a TDMA network only one transmitter may be on at a time, 'clash' occurs when two or more transmitters are on at the same time and will often cause data loss at the receivers. TDMA networks may be configured in several ways - Synchronous (time slots), Polling (master-slave) or Random access (async packet switching e.g. X25). Networked NiM2's allow several techniques for range / reliability enhancement:

**Store and forward Repeaters:** If the operating protocol of the network is designed to allow data path control then data may be routed via intermediate nodes. The inclusion of a repeating function in the network protocol either via dedicated repeater/router nodes or simply utilising existing nodes allows limitless network expansion.

**Spatial Diversity:** In buildings multi-path signals create null spots in the coverage pattern as a result of signal cancellation. In master-slave networks it is cost effective to provide 2 NiM2's with separate antenna at the master station. The null spot patterns will be different for the two NiM2's. This technique 'fills in' the null spots, i.e. a handshake failure on the first NiM2 due to a signal null is likely to succeed on the 2nd NiM2.

## "RS232" Serial data

It is possible to transmit "RS232" serial data directly at 600 to 9600bps baud between a pair of NiM2 transceivers in half duplex mode. The data must be "packetised" with no gaps between bytes. i.e. The data must be preceded by >10ms of preamble (55h or AAh) to allow the data slicer in the NiM2 to settle, followed by one 00h and one FFh bytes to allow the receive UART to lock, followed by a unique start of message byte, (01h), then the data bytes and finally terminated by a CRC or check sum. The receiver data slicer provides the best bit error rate performance on codes with a 50:50 mark:space average over a 5ms period, a string of FFh or 00h is a very asymmetric code and will give poor error rates where reception is marginal. Only 50:50 codes may be used at data rates above 1kbps.

We recommend 3 methods of improving mark:space ratio of serial codes, all 3 coding methods are suitable for transmission at 10kbps:-

- **Method 1 - Bit coding**

Bit rate , Max 10kbps , Min 250bps  
Redundancy (per bit) 100% (Bi-phase)

Each bit to be sent is divided in half, the first half is the bit to be sent and the second half, it's compliment. Thus each bit has a guaranteed transition in the centre and a mark:space of 50:50. This is Bi-phase or Manchester coding and gives good results, however the 100% redundancy will give a true throughput of 5kbps.

Another variation of this code is to encode a '1' as a long bit with one transition and '0' as a short bit with two transition or vice versa. Each encoded bit starts with a guaranteed transition to reverse the voltage level even if stream of 00h/FFh is encoded. This is called Differential Manchester Encoding. This encoding method is easier to decode as the decoder has to sample encoded bit several times and if the sample value is more than 75% of a long bit period, then it is decoded as '1' and if there was transition then it is decoded as '0' or vice versa.

- **Method 2 - FEC coding**

Bit rate , Max 10kbps, Min 2.4kbps  
Redundancy (per byte) 100%

Each byte is sent twice; true then it's logical compliment. e.g. even bytes are true and odd bytes are inverted. This preserves a 50:50 balance.

A refinement of this simple balancing method is to increase the stagger between the true and the inverted data streams and add parity to each byte. Thus the decoder may determine the integrity of each even byte received and on a parity failure select the subsequent inverted odd byte. The greater the stagger the higher the immunity to isolated burst errors.

### **Digitised analogue data**

Linear operation of NiM2 transceivers will allow direct transfer of analogue data, however in many applications the distortion and low frequency roll off are too high (e.g. bio-medical data such as ECG). The use of delta modulation is an excellent solution for analogue data in the range 1Hz up to 4kHz with less than 1% distortion. A number of proprietary IC's such as Motorola's MC3517/8 provide CVSD Delta mod/demod on a single chip.

Where the signal bandwidth extends down to DC , such as strain gauges, level sensing, load cells etc. then Voltage to Frequency / Frequency to Voltage chips (such as Nat Semi LM331) provide a simple means of digitising.

### **Module mounting considerations**

Good RF layout practice should be observed. If the connection between module and antenna is more than about 20mm long use 50Ω microstrip line or coax or a combination of both. It is desirable (but not essential) to fill all unused PCB area around the module with ground plane.

### **Variants and ordering information**

The NiM2T transmitters, NiM2R receivers and NiM2 transceivers are manufactured in the following variants as standard:

<i>At 434.650MHz:</i>	NiM2-434.65-10	Transceiver
	NiM2T-434.65-10	Transmitter
	NiM2R-434.65-10	Receiver
<i>At 434.075MHz:</i>	NiM2-434.075-10	Transceiver
	NiM2T-434.075-10	Transmitter
	NiM2R-434.075-10	Receiver

*Other frequency variants can be supplied to individual customer requirements in the 433MHz (European) licence exempt bands*

# **Radiometrix Ltd**

**Hartcran House**

**231 Kenton Lane**

**Harrow, Middlesex**

**HA3 8RP**

**ENGLAND**

**Tel: +44 (0) 20 8909 9595**

**Fax: +44 (0) 20 8909 2233**

**sales@radiometrix.com**

**www.radiometrix.com**

## **Copyright notice**

*This product data sheet is the original work and copyrighted property of Radiometrix Ltd. Reproduction in whole or in part must give clear acknowledgement to the copyright owner.*

## **Limitation of liability**

*The information furnished by Radiometrix Ltd is believed to be accurate and reliable. Radiometrix Ltd reserves the right to make changes or improvements in the design, specification or manufacture of its subassembly products without notice. Radiometrix Ltd does not assume any liability arising from the application or use of any product or circuit described herein, nor for any infringements of patents or other rights of third parties which may result from the use of its products. This data sheet neither states nor implies warranty of any kind, including fitness for any particular application. These radio devices may be subject to radio interference and may not function as intended if interference is present. We do NOT recommend their use for life critical applications.*

*The Intrastat commodity code for all our modules is: 8542 6000*

## **R&TTE Directive**

After 7 April 2001 the manufacturer can only place finished product on the market under the provisions of the R&TTE Directive. Equipment within the scope of the R&TTE Directive may demonstrate compliance to the essential requirements specified in Article 3 of the Directive, as appropriate to the particular equipment.

Further details are available on The Office of Communications (Ofcom) web site:

*<http://www.ofcom.org.uk/radiocomms/ifi/>*

### *Information Requests*

*Ofcom*

*Riverside House*

*2a Southwark Bridge Road*

*London SE1 9HA*

*Tel: +44 (0)845 456 3000 or 020 7981 3040*

*Fax: +44 (0)20 7783 4033*

*information.requests@ofcom.org.uk*

### *European Radiocommunications Office (ERO)*

*Peblingehus*

*Nansensgade 19*

*DK 1366 Copenhagen*

*Tel. +45 33896300*

*Fax +45 33896330*

*ero@ero.dk*

*www.ero.dk*

---