

Igor Araújo de Souza
Lucas Steinle Duarte Silva

DISPOSITIVO PARA GESTÃO DE FILA DE ESPERA VIA SISTEMA DE
PAGERS

São Paulo
2010

Igor Araújo de Souza
Lucas Steinle Duarte Silva

DISPOSITIVO PARA GESTÃO DE FILA DE ESPERA VIA SISTEMA DE
PAGERS

Trabalho de formatura apresentada à Escola
Politécnica da Universidade de São Paulo para
obtenção do título de Engenheiro Mecatrônica
Área de concentração: Engenharia Mecatrônica

Orientador: Prof. Dr. Marcos Barreto

São Paulo
2010

FICHA CATALOGRÁFICA

Souza, Igor Araújo de

**Dispositiva para gestão de fila de espera via sistema de
pagers / I.A. de Souza, L.S.D. Silva -- São Paulo, 2010.
65 p.**

**Trabalho de Formatura - Escola Politécnica da Universidade
de São Paulo. Departamento de Engenharia Mecatrônica e de
Sistemas Mecânicos.**

**1. Mecatrônica I. Silva, Lucas Steinle Duarte II. Universidade
de São Paulo. Escola Politécnica. Departamento de Engenharia
Mecatrônica e de Sistemas Mecânicos III. t.**

RESUMO

Filas de esperas estão presentes em diversos lugares como: bancos, restaurantes, hospitais, lojas e vários outros estabelecimentos comerciais. Muitas das vezes, os clientes ficam “presos” ao local, com uma pequena senha de papel, esperando que a sua vez seja anunciada num painel eletrônico. Visando aqueles estabelecimentos que procuram se diferenciar da maioria, oferecendo um serviço de melhor qualidade para seus clientes, este trabalho tem como objetivo desenvolver de um sistema de pagers para substituir as tradicionais senhas, possibilitando o deslocamento do cliente naquele período de espera na fila. As fases de planejamento, projeto e implementação abrangem a escolha das funcionalidades relevantes, estudo e seleção das interfaces viáveis de comunicação Wireless, projeto de softwares utilizando técnicas de modelagem SysML, escolha dos componentes de hardware, fabricação de protótipos e programação dos módulos de Cliente e Servidor.

ABSTRACT

Standing in a Queue is something that we do almost every day in several places like: banks, restaurants, hospitals, stores e others establishments. Very often, the clients get “stuck” in the line with a password made with paper, waiting to see his number on the electronic device. Looking for those establishments that want to differentiate themselves from the others by offering a better service to their customers, this project aims to develop a pager system to substitute the traditional password made with paper, releasing the costumer from the line. The planning, design and implementation’s steps will define the relevant functionalities, the definition of the Wireless’s communication interfaces, the software’s design using SysML modeling, the hardware’s components, the prototypes manufacture and programing the Client’s and Server’s module.

LISTA DE ILUSTRAÇÕES

Figura 1 - Módulos RT4 e RR3.....	16
Figura 2 - Módulo da Radiocrafts	16
Figura 3 - Xbee, da empresa Digi.....	17
Figura 4 - Multipath	21
Figura 5 - Influência da Baud Rate no sinal recebido.....	21
Figura 6 - Padrão de radiação para a antena isotrópica hipotética de 0 dBi	22
Figura 7 - Padrão de radiação para uma antena omnidirecional de 3 dBi.....	22
Figura 8 - Representação 3D para o padrão de radiação de uma omni 5.8 dBi	24
Figura 9 - Padrão de radiação para uma antena patch de 8 dBi.....	24
Figura 10 - Representação em 3D do padrão de radiação de uma antena patch.....	25
Figura 11 - Diagrama de circuito do módulo	27
Figura 12 - Diagrama da PCB	27
Figura 13 - Conceito para o pager.....	28
Figura 14 - Central de Comando	29
Figura 15 - Diagrama de circuito do pager	32
Figura 16 - Diagrama da PCB do pager	32
Figura 17 - Diagrama de circuito da central	43
Figura 18 - Diagrama da PCB da central	44
Figura 19 - Matriz do circuito e placa cobreada.....	46
Figura 20 - Transferência térmica e remoção do papel.....	46
Figura 21 - Papel removido e detalhes da resolução alcançada	47
Figura 22 - Processo de corrosão e a placa já corroída.....	47
Figura 23 - Placa finalizada, pronta para furação.....	47
Figura 24 - Placa para o módulo RC2500HP	48

Figura 25 - Placa da Central pronta	48
Figura 26 - Placa de um dos pagers pronta	49
Figura 27 - Placa já com o módulo no lugar	49

LISTA DE DIAGRAMAS

Diagrama 1 - Caso de Uso dos módulos dos pagers	30
Diagrama 2 - Diagrama de atividades do pager	31
Diagrama 3 - Caso de Uso da Central de Comando	34
Diagrama 4 - Diagrama de atividades da central	36

LISTA DE TABELAS

Tabela 1 - Matriz de Decisão para Método de Comunicação.....	18
Tabela 2 - Palavras de controle para comunicação	26
Tabela 3 - Custo estimado dos componentes para o circuito do pager.....	33
Tabela 4 - Custo estimado dos componentes para o circuito da central	45

SUMÁRIO

1. INTRODUÇÃO	12
1.1. MOTIVAÇÃO.....	12
1.2. OBJETIVOS.....	13
1.3. ORGANIZAÇÃO DO TRABALHO	13
2. REQUISITOS	14
2.1. REQUISITOS FUNCIONAIS.....	14
2.2. REQUISITOS TÉCNICOS	14
2.3. REQUISITOS COMERCIAIS	15
3. ANÁLISE DE ALTERNATIVAS PARA COMUNICAÇÃO	16
3.1. TRANSMISSOR RT4 E RECEPTOR RR3.....	16
3.2. RC1140 OU RC2500HP	16
3.3. XBEE	17
3.4. MATRIZ DE DECISÃO.....	18
4. PROJETO BÁSICO	19
4.1. LIMITE DE ALCANCE.....	19
4.2. INTERFERÊNCIAS E RUÍDOS	20
4.2.1. Fading	20
4.2.2. <i>Multipath</i> e propagação de atraso	20
4.3. TIPOS DE ANTENAS	22
4.4. SELEÇÃO DO MÓDULO E DO PROTOCOLO DE COMUNICAÇÃO	25
4.2.1. Diagrama de circuito e de PCB do módulo	27
4.3. SELEÇÃO DE HARDWARE DOS PAGERS	28
4.2.1. Interface Homem-Máquina do Pager	28

4.3. SELEÇÃO DE HARDWARE DA CENTRAL DE COMANDO	29
4.3.1. Interface Homem-Máquina da Central	29
5. PROJETO DO <i>PAGER</i>	30
5.1. CASOS DE USO.....	30
5.1.1. Diagrama e Descrições	30
5.1.2. Diagrama de Atividade	31
5.2. HARDWARE DO PAGER	32
5.2.1. Diagrama de Circuito.....	32
5.2.2. Diagrama da PCB	32
5.2.2. Lista de Materiais	33
6. PROJETO DA CENTRAL DE COMANDO	34
6.1. CASOS DE USO.....	34
6.1.1. Diagramas e Descrições	34
6.1.2. Diagramas de Atividades.....	36
6.2. HARDWARE DA CENTRAL.....	43
6.2.1. Diagrama de Circuito.....	43
6.2.2. Diagrama da PCB	44
6.2.3. Lista de Materiais	45
7. FABRICAÇÃO DOS PROTÓTIPOS	46
7.1. CONFECÇÃO DE PCBS POR TRANSFERÊNCIA TÉRMICA DE TONER	46
7.2. BREAKOUT BOARD RC2500HP	48
7.3. CENTRAL	48
7.4. PAGER	49
8. IMPLEMENTAÇÃO DO SOFTWARE	50
9. DISCUSSÃO DO PROTÓTIPO.....	51
10. CONCLUSÃO	52
11. BIBLIOGRAFIA.....	53

ANEXO 1 – CÓDIGO FONTE DA CENTRAL	55
ANEXO 2 - CÓDIGO FONTE DOS PAGERS	63

1. INTRODUÇÃO

1.1. MOTIVAÇÃO

O tempo para o homem atual está se tornando um recurso cada vez mais escasso e a busca por uma eficiência maior no dia-a-dia é um fato. O aproveitamento do tempo pode ser comparado ao de um recurso não-renovável presente no nosso planeta, o seu uso consciente pode levar a melhorias em áreas aparentemente desconectadas daquele evento, que acarreta uma série de modificações para a melhoria da sociedade.

Um dos pontos de desperdício de tempo ocorre quando pessoas ficam ociosas aguardando para realizar alguma outra tarefa. O tempo que é perdido nesta fila poderia ser mais bem aproveitado se o indivíduo não ficasse “preso” ao local para preservar seu lugar na tal fila. Isso é um caso comum na vida humana que causa estresse, desconforto e insatisfação do cliente com qualquer que seja o estabelecimento comercial: bancos, lojas, restaurantes, bares, hospitais, *pet shops*, entre outros.

Conhecendo esta deficiência no atual cenário comercial, começaram a surgir produtos para atender esta demanda para otimização do tempo em filas. Tais produtos procuram dar liberdade ao cliente sem que o mesmo perca a sua vez de ser atendido, minimizando problemas causados pelos fatores citados anteriormente.

O gerenciamento de filas via *paggers* é um sistema novo no mercado nacional e até mesmo no internacional, sendo que os únicos produtos semelhantes no país são importados. Isso faz com que a popularização do produto se torne difícil devido ao alto custo agregado a implantação do sistema.

Tendo como objetivo suprir a demanda por um sistema mais acessível, este trabalho procurar desenvolver um sistema de gestão de filas que atenda as necessidades comerciais existentes no mercado com um custo mais baixo.

1.2. OBJETIVOS

O principal objetivo deste trabalho é o projeto de um sistema capaz de auxiliar no controle de uma fila qualquer, ou seja, algo que substitua o tradicional bilhete numerado por algo que possa interagir com o usuário, eliminando a necessidade de que o mesmo tenha que ficar restrito ao local, aumentando a satisfação do cliente e as possibilidades de negócio para o estabelecimento.

Em segundo plano, os objetivos secundários deste trabalho abrangem o aprofundamento do conhecimento na área de transmissão sem fio, em metodologia de projeto e na estruturação de um documento formal para exposição de tal experiência.

1.3. ORGANIZAÇÃO DO TRABALHO

O trabalho é estruturado da seguinte forma:

Capítulo 2: Apresentação dos requisitos funcionais, técnicos e comerciais;

Capítulo 3: Análise das alternativas propostas para solução do problema de comunicação sem fio via rádio, com posterior escolha através de uma matriz de decisão.

Capítulo 4: Projeto básico envolvendo a escolha do *protocolo de comunicação* e dos componentes para o módulo do *paggers* e da central de controle.

Capítulo 5: Utilizando os componentes previamente definidos, apresenta-se aqui um projeto detalhado do hardware e diagramas a serem utilizados no desenvolvimento do módulo do *pager*.

Capítulo 6: Apresentação do projeto de hardware da central de comando, fazendo considerações sobre sua interação com o operador através de diversos diagramas, além da apresentação do circuito propriamente dito.

Capítulo 7: Método para confecção de placas de circuito impresso e fotos dos protótipos.

Capítulo 8: Implementação do software, juntamente com os códigos fonte.

Capítulo 9: Conclusão e considerações do trabalho.

Capítulo 10: Bibliografia

2. REQUISITOS

2.1. REQUISITOS FUNCIONAIS

Os requisitos funcionais podem ser descritos em função de casos de uso acionados pelo usuário. Tais casos de uso serão encontrados em capítulos adiante junto com outros diagramas formais de desenvolvimento de software.

Por hora, segue a seguinte descrição dos casos necessários para o funcionamento básico do sistema:

- **Adicionar:** Permite ao usuário adicionar mais um integrante na lista de espera, esse integrante ocupará a ultima posição da lista.
- **Remover:** Permite ao usuário remover um integrante qualquer da lista de espera.
- **Chamar:** Permite ao usuário acionar o dispositivo relacionado ao próximo integrante da lista de espera.
- **Consultar:** O usuário deve ser capaz de consultar a lista de espera.

Deve-se lembrar que haverá a possibilidade de existirem duas filas de espera, cada uma com as quatro funções acima. Além disso, o sistema deverá ser capaz de gerar, para cada fila, relatórios de uso com informações de tempo médio de espera, fluxo de pessoas atendidas por hora e quantidade de cancelamentos.

2.2. REQUISITOS TÉCNICOS

Os aparelhos de *paggers* para filas de espera existentes hoje no mercado são todos importados e, por isso, seu custo é bastante elevado. A intenção desse projeto é fazer um produto nacional com um preço mais baixo, mas atendendo os requisitos citados abaixo, pois sem eles, não há como competir nesse mercado.

Os critérios mais importantes que serão atendidos são:

- Alcance mínimo de 100m;
- Adequação a regras vigentes nos mercados de interesse;
- Interface amigável
- Baixo custo de manutenção
- Possibilidade de uso de mais um sistema em um mesmo ambiente.

O custo mais elevado de alguns componentes, as dificuldades de manufatura de um protótipo sem máquinas especializadas e as limitações de tempo devem ser considerados ao decidir entre a fabricação fiel ao produto, o uso de plataformas de desenvolvimento industriais ou uma solução híbrida.

2.3. REQUISITOS COMERCIAIS

Uma vez que mais de uma solução satisfaz determinados requisitos técnicos, é imperativo que estas soluções sejam analisadas com base em requisitos comerciais. Um estudo detalhado no mercado sobre os preços praticados pelos concorrentes mostra que todos eles estão na faixa de R\$ 200,00 por Pager e R\$ 500,00 a central. Uma estimativa para o projeto terá um custo de produção de R\$ 90,00 para um Pager e de R\$ 110,00 para a central.

Depois de concluída a fase de desenvolvimento, deve-se averiguar se tal requisito comercial foi satisfeito.

3. ANÁLISE DE ALTERNATIVAS PARA COMUNICAÇÃO

3.1. TRANSMISSOR RT4 E RECEPTOR RR3

Os módulos RT4 e RR3 são transmissores e receptores de rádio frequência (RF) simples, não possuem qualquer tipo de protocolo de comunicação e toda a gestão deste fica a critério do microcontrolador. Os módulos operam nas faixas de frequência de 315, 418 e 434 MHz e, segundo o fabricante, possuem alcance de até 100 metros.

A empresa italiana Telecontrolli realiza o ajuste de frequência tanto do transmissor quanto do receptor através de um processo a laser na própria fábrica, o que elimina um grande problema no desenvolvimento de sistemas baseados em RF: falhas na comunicação ou na qualidade de transmissão devido uma má sintonização do par TX-RX.

Ressalta-se que a grande desvantagem do uso deste conjunto seria a ausência de um protocolo de comunicação definido, já que estes módulos simplesmente transmitem e recebem dados.

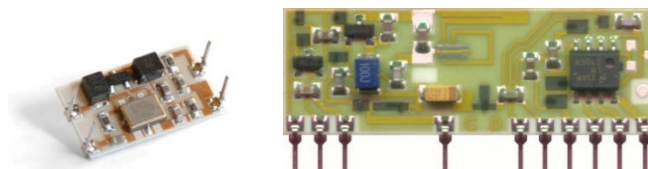


Figura 1 - Módulos RT4 e RR3

3.2. RC1140 OU RC2500HP

A empresa Radiocrafts apresenta diversas opções que podem ser usadas para comunicação sem fio, entre elas estão o RC1140 (que é um módulo de rádio similar ao RT4/RR3) e o RC2500HP, que operam na banda de 2.4 GHz. São módulos de comunicação serial RS232, sendo que o modelo RC2500HP possui grande potência de transmissão, podendo ampliar o limite de alcance do projeto em várias dezenas e metros.



Figura 2 - Módulo da Radiocrafts

Este módulo possui protocolo de transmissão proprietário, o RC232, isso significa que o uso deste modelo seria um tipo de “caixa preta”, ou seja, os dados seriam enviados pela porta serial, codificados e transmitidos, recebidos e decodificados do outro lado e enviados via serial para o destinatário. Observe que este dispositivo substitui diretamente o cabo de uma comunicação RS232 sem muitas modificações no software e no hardware.

3.3. XBEE

A empresa Digi fabrica os módulos Xbee que se baseiam no protocolo ZigBee de comunicação. Este protocolo foi desenvolvido para redes sem fio que atendessem a demanda por sistemas que necessitem de baixa taxa de transmissão de dados, como redes de automação predial, sistemas de segurança domésticos, controle industrial e sensoriamento remoto.

Apesar de que isto poderia ser tratado no desenvolvimento do software, a vantagem desta solução é a facilidade de implementação da rede de dispositivos, amenizando as dificuldades de alocação de endereços, já que o projeto prevê a existência de um número relativamente significativo de células de comunicação.



Figura 3 - Xbee, da empresa Digi

3.4. MATRIZ DE DECISÃO

Neste ponto, deve-se escolher por uma das três alternativas propostas, esta situação de múltiplas alternativas disponíveis é muito comum de ocorrer durante projetos e, normalmente, o artifício conhecido como matriz de decisão é utilizado para se escolher a opção mais correta. A apostila da matéria PNV2100, Introdução à Engenharia, define matriz de decisão do seguinte modo:

Uma forma relativamente simples de implementar o procedimento acima esboçado é o que se chama de Matriz de Decisão. Consiste em selecionar a melhor alternativa pela determinação da maior média ponderada das notas. Assim, colocamos em uma tabela, de um lado, as soluções propostas, por exemplo ao longo da primeira linha. Na primeira coluna listamos os critérios de avaliação. Pode-se reservar a 2a. coluna para atribuir os pesos atribuídos aos critérios. Os demais espaços da tabela são utilizados para a atribuição de notas a cada uma das alternativas segundo os critérios adotados.

Tomando as soluções propostas, resta determinarmos os critérios de avaliação e seus respectivos pesos, entretanto deve-se ressaltar que tal processo dá-se de maneira subjetiva. Abaixo são listados os critérios os escolhidos para a análise:

- Custos de Produção
- Facilidade de implementação do software
- Fornecedores
- Confiabilidade da rede
- Capacidade de implementação de novos recursos

Tabela 1 - Matriz de Decisão para Método de Comunicação

		Módulos da empresa		
Crítérios	Pesos	Telecontrolli	Radiocrafts	Digi
I	8	10	7	7
II	6	5	9	9
III	6	8	7	7
IV	10	6	8	5
V	7	3	8	8
	Total	239	288	258

4. PROJETO BÁSICO

O projeto básico consiste da definição do módulo de transmissão, do módulo de recepção de rádio e do microcontrolador a ser utilizado em cada parte do sistema. Estes componentes são a base para a escolha dos demais elementos do projeto de hardware e são responsáveis pela definição das capacidades do software.

De acordo com a matriz de decisão mostrada no item 3.1.4, o módulo da Radiocrafts RC2500HP é a melhor alternativa para o sistema de comunicação do projeto, entre suas características destacam-se a facilidade de implementação da comunicação e a confiabilidade da rede por conter um protocolo próprio de checagem de erros.

No caso dos microcontroladores optou-se por utilizar um fabricado pela Atmel, o escolhido foi o Atmega88 com encapsulamento TQFP, devido ao baixo custo envolvido já que o encapsulamento DIP custa cerca de três vezes mais. O código é baseado na linguagem da interface Arduino, que possui uma interface bastante amigável, tendo muita semelhança com a linguagem de programação C++, conhecida pela maioria dos programadores.

4.1. LIMITE DE ALCANCE

A distância máxima que se pode transmitir através de um cabo é limitada principalmente pela capacitância do fio, que progressivamente atenua o sinal até não ser mais possível a leitura do sinal proveniente do transmissor.

Essa atenuação também ocorre na comunicação via rádio e um dos fatores é a simples dispersão das ondas pelo espaço, entretanto questões ambientais, de terreno e outros motivos também podem influenciar significativamente a qualidade do sinal recebido pelo receptor. No ar, a atenuação é proporcional ao quadrado da distância.

Sabe-se que o sinal recebido está em função da frequência da portadora, a potência transmitida, a distância entre TX e RX, os ganhos de antenas, etc. A título de curiosidade, verifica-se que dobrando a potência não se dobra o alcance devido ao decaimento quadrático do sinal. Então para o dobro de potência, há um aumento aproximado de 1,414 na distância atingida.

4.2. INTERFERÊNCIAS E RUÍDOS

4.2.1. Fading

Define todas as variações da atenuação do sinal devido a sua propagação em um ambiente real. O sinal do radio interage de várias maneiras com o ambiente e por isso, varia muito de acordo com a configuração do ambiente. Quanto maior a distância, maior será o impacto causado pelo Fading, e o sistema irá degradar-se com o aumento da distância até perder a comunicação.

4.2.2. *Multipath* e propagação de atraso

As ondas de radio refletem ou difratam em obstáculos, e são atenuadas de forma diferente de acordo com o material. É exatamente o que ocorre com a luz exceto que existem mais materiais transparentes ou refletores para o rádio.

Em um ambiente real como em uma casa ou escritório, existem várias superfícies refletoras das ondas de radio (paredes, tetos, metais), várias outras que são semitransparentes (paredes, tetos, humanos) e até opacas como alguns metais. Isso causa muitos problemas na hora de estimar o alcance do sistema e significa inclusive que o sinal recebido em um nó poderá vir de diferentes direções e com força diferente, mas o receptor apenas detecta a combinação de todas essas reflexões. Este fenômeno é chamado de *multipath*.

Na maioria das vezes, o *multipath* é bom pois a adição de todas as reflexões aumentam a força do sinal, entretanto o seu principal efeito é a dificuldade de avaliar o alcance do sistema.

O grande problema do *multipath* é que ele cria uma propagação atrasada e dependendo do número de reflexões e da velocidade de propagação dos sinais refletidos, eles não chegam ao mesmo tempo no receptor. É como se fosse um eco, o sinal enviado diretamente será mais rápido do que o sinal refletido em uma parede por exemplo.

Como as ondas de radio se propagam na velocidade da luz, essa diferença é muito pequena, mas quando a taxa de bits do sistema aumenta, essa diferença se torna significativa a ponto de criar interferências destrutivas nas mensagens enviadas. Uma taxa de bits abaixo de 1 Mbps é relativamente imune a esse fenômeno de propagação do atraso, mas aumentando essa taxa de transferência acima de 1

Mbps, esse fenômeno aumenta. Sistemas que possuem velocidades de transferência acima de 5 Mbps necessitam de alguma técnica para contornar o problema do *multipath*.

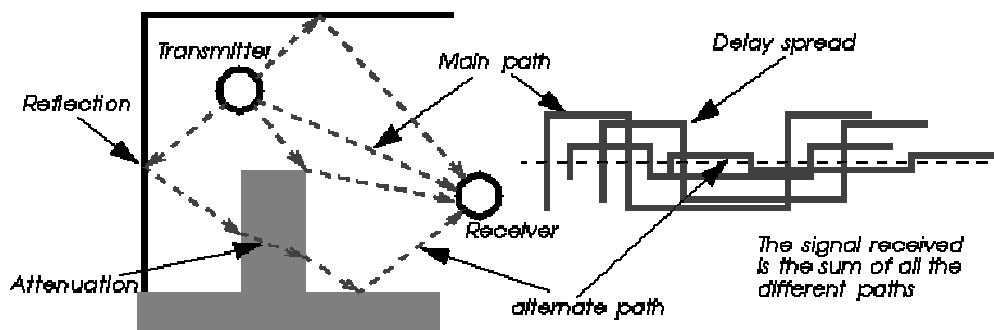


Figura 4 - Multipath

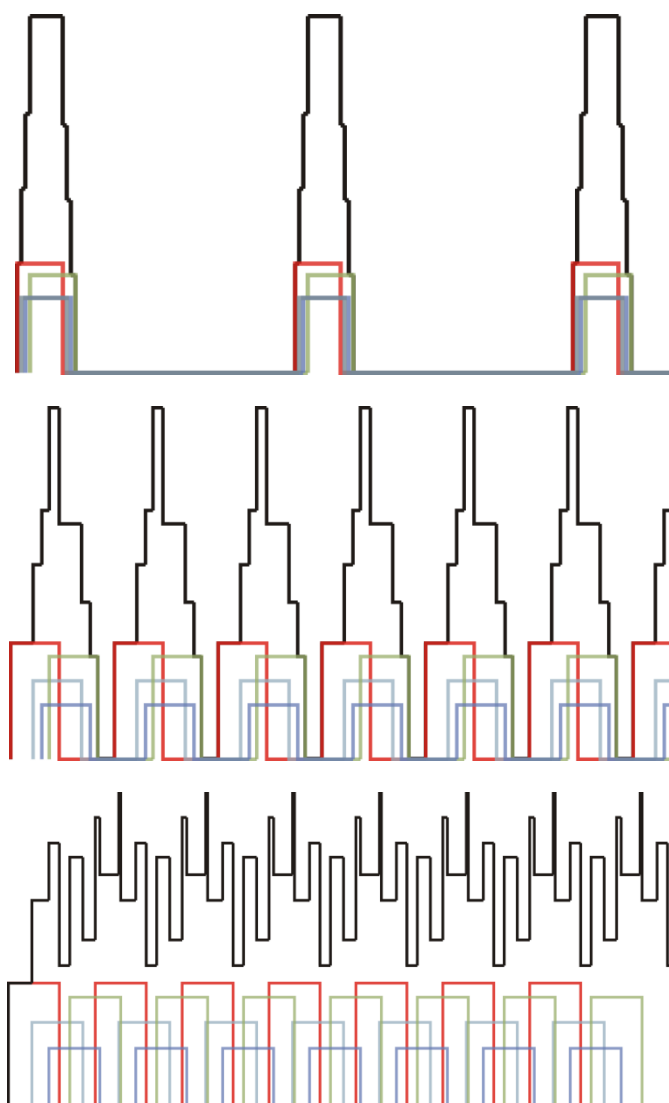


Figura 5 - Influência da Baud Rate no sinal recebido

A principal técnica para superar esse problema é usar um equalizador. Um equalizador é um grande circuito digital que procura estimar os diferentes compo-

nentes do sinal recebido. O equalizador precisa ser “treinado” para determinar quais são as mensagens erradas, separando os diferentes componentes do sinal e recalculando o sinal correto removendo o atraso.

4.3. TIPOS DE ANTENAS

O tipo de antena e o ganho da mesma podem fazer uma grande diferença no alcance e na qualidade do sinal, entretanto a escolha adequada é fundamental para que o resultado obtido não deixe de ser atendido.

Primeiramente define-se que decibel (dB) “é uma unidade de medida logarítmica que expressa a magnitude de uma quantidade física (como de potência ou intensidade) relativa a um valor de referência específico ou imposto”. Ou seja, dB pode ser considerado uma unidade de potência.

Um antena com 0 dBi de ganho propagaria o sinal esfericamente perfeito, mas é claro que essa situação é impossível de ser realizada na vida real. Na escala logarítmica, a cada 3 dB de acréscimo significa o dobro de potência de transmissão, que representa um aumento de 1,414 vezes no alcance.

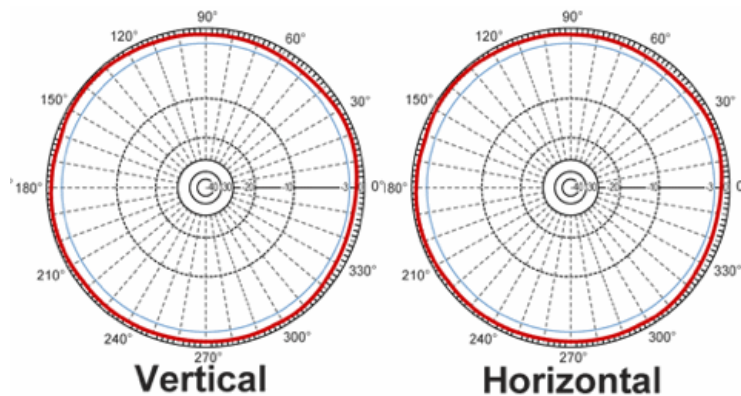


Figura 6 - Padrão de radiação para a antena isotrópica hipotética de 0 dBi

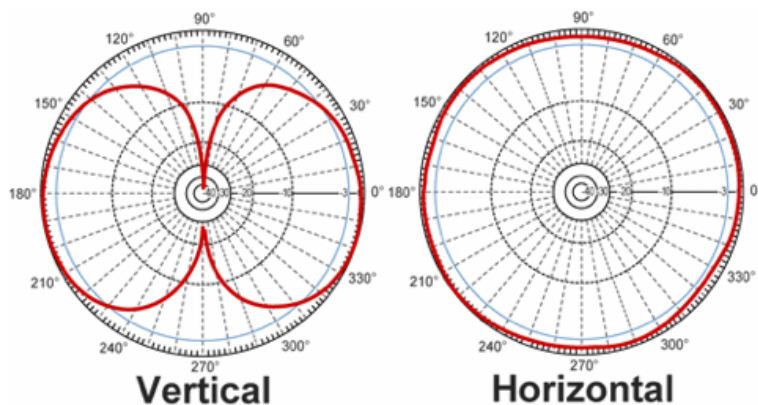


Figura 7 - Padrão de radiação para uma antena omnidirecional de 3 dBi

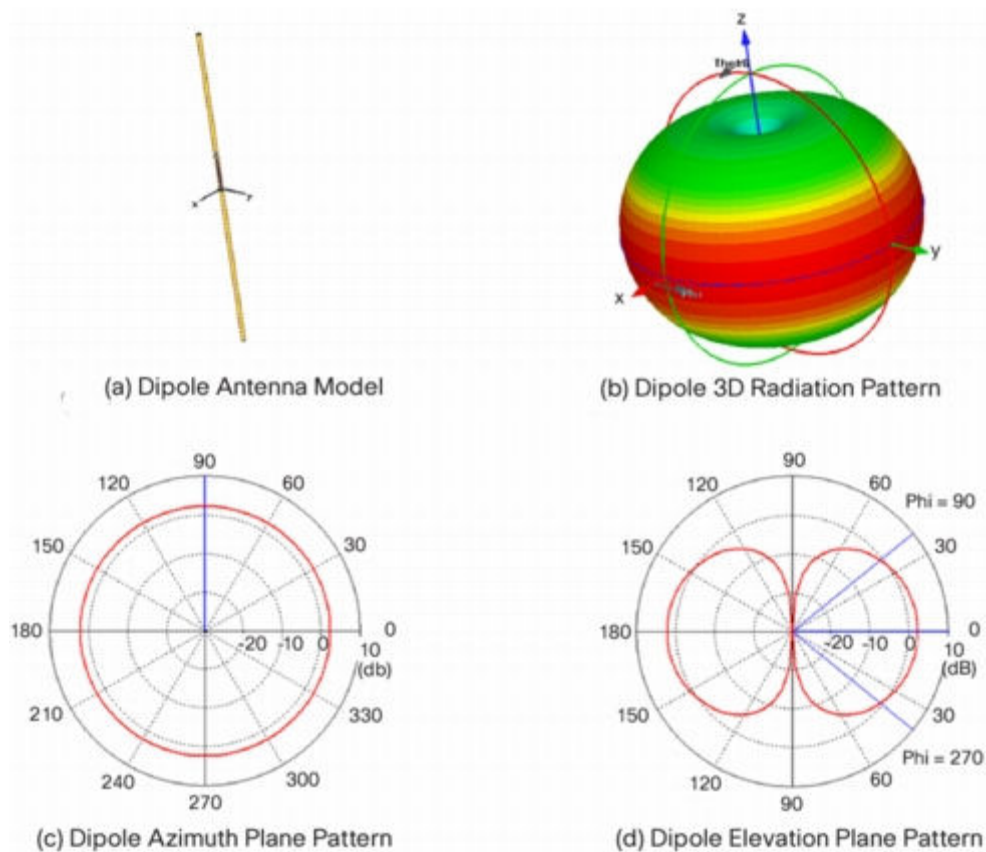


Figura 8 - Representação 3D para padrão de radiação de uma antena dipolo

Note que houve uma mudança na distribuição vertical, percebe-se que no eixo da antena a transmissão/recepção de sinal é praticamente nulo.

Vejamos agora uma antena omnidirecional com ainda mais ganho, note que apesar do alcance maior no plano perpendicular a antena, o volume de propagação é ainda mais “achatado” neste caso.

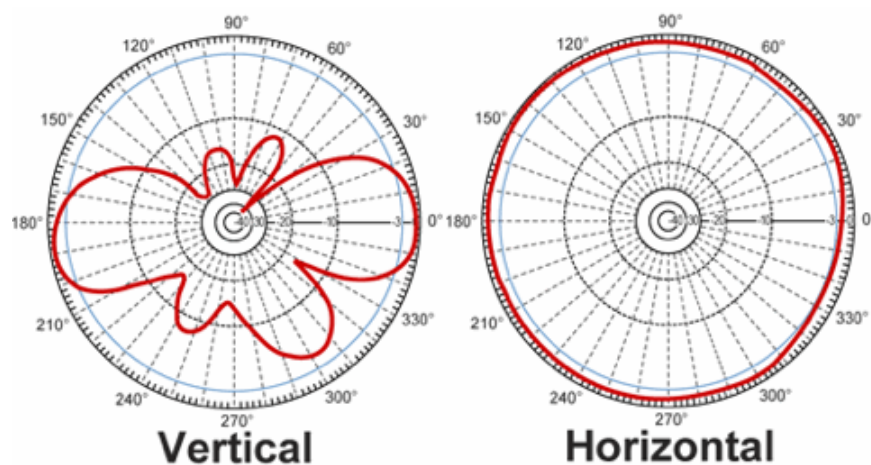


Figura 9 - Padrão de radiação para uma antena omnidirecional de 7 dBi,

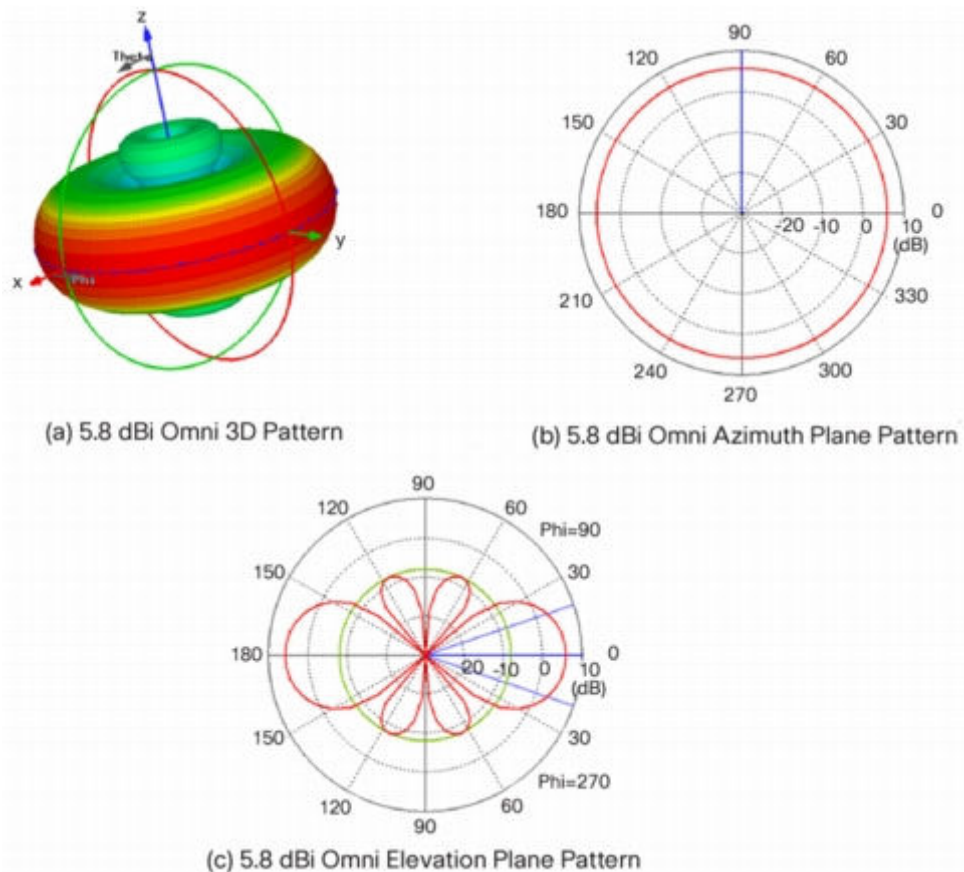


Figura 8 - Representação 3D para o padrão de radiação de uma omni 5.8 dBi

Uma forma de conseguir ganhos ainda maiores é utilizar antenas direcionais. Este tipo concentra o padrão de radiação em apenas uma direção, não sendo difícil alcançar ganhos que multiplica em várias vezes a potência transmitida.

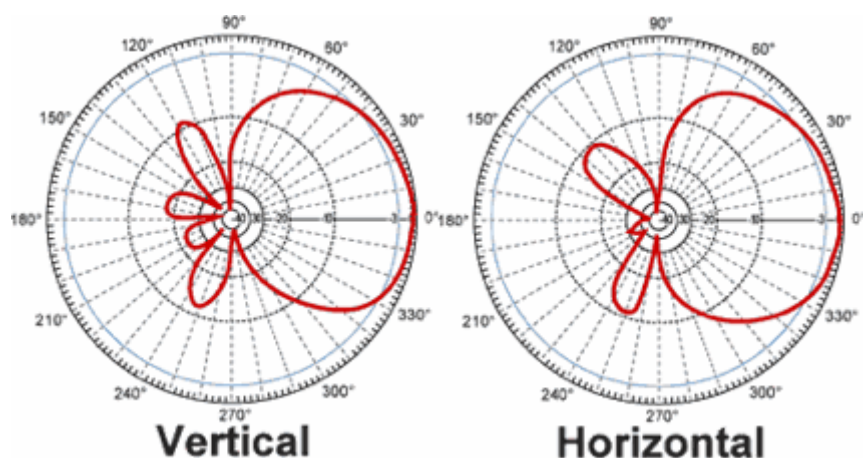


Figura 9 - Padrão de radiação para uma antena patch de 8 dBi

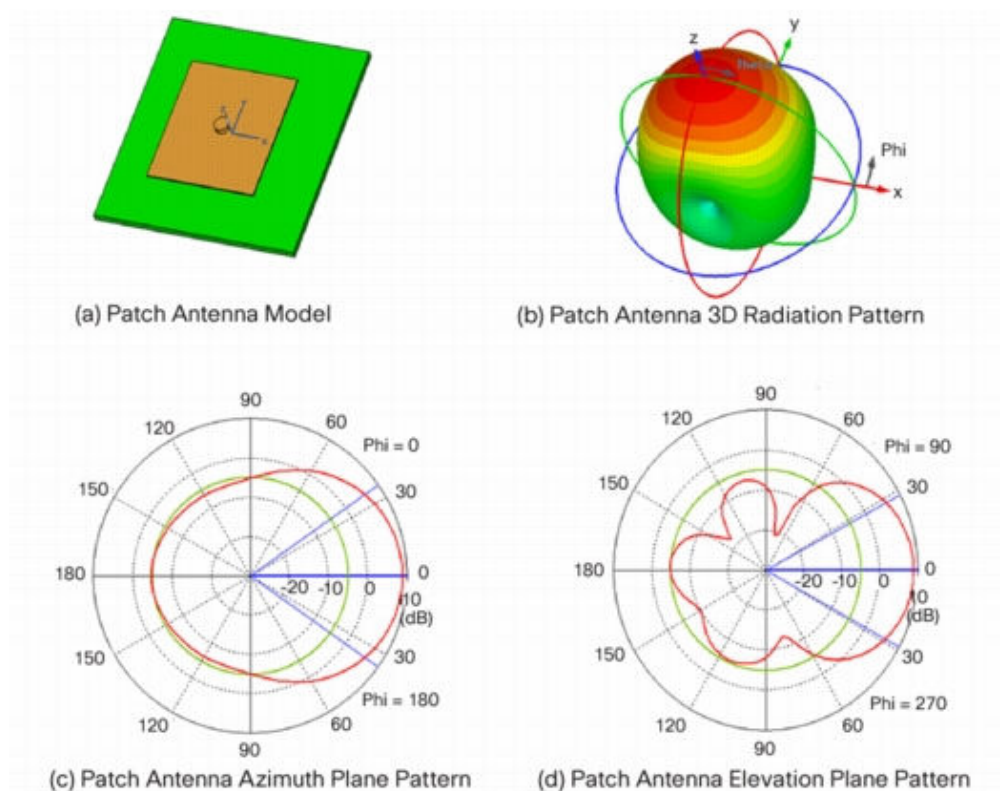


Figura 10 - Representação em 3D do padrão de radiação de uma antena patch

4.4. SELEÇÃO DO MÓDULO E DO PROTOCOLO DE COMUNICAÇÃO

O módulo utilizado (RC2500HP) já possui um protocolo de comunicação próprio, o RC232, que estabelece uma transferência de dados bem confiável entre os *transceivers*. Com isso, a comunicação resume-se a definir quais palavras de controle serão utilizadas para envio e quais serão esperadas no recebimento.

Foi definido que são necessárias três operações diferentes para que os casos propostos possam ser realizados. São elas:

- 1) Adição de um *pager* em uma fila de espera;
- 2) Chamada de um *pager*;
- 3) Consulta à posição na fila por um *pager* qualquer.

As duas primeiras operações são iniciadas pela central e esta aguarda uma palavra de confirmação do *pager* em questão. A terceira operação é iniciada por um *pager* qualquer, que aguarda uma resposta da central. Então é necessário que haja um "ID" único para identificar cada *pager* na rede e um código para a operação, assim como um outro para a sua respectiva resposta.

Como a central precisa gerenciar todos os *paggers* do sistema, foi estabelecido o padrão para recebimento de mensagens para a mesma seja que ela sempre receba primeiro a operação a ser realizada e depois o ID do remetente da mensagem, além de um caractere que sinalize o fim da palavra (*end character*). Para o envio, definiu-se que primeiro seja enviado o ID do destinatário e depois a operação a ser feita, seguido do *end byte*.

Foi-se escolhido que o *end byte* seria o caractere ‘&’ e que as letras para as outras operações seriam: ‘a’, ‘c’ e ‘p’, com as respectivas respostas: ‘b’, ‘d’ e ‘q’. Representando o ID como #, obteve-se as seguintes palavras de controle para a comunicação serial:

Tabela 2 - Palavras de controle para comunicação

Operação	Remetente	Destinatário	Envio	Resposta
Adicionar	Central	<i>Pager</i>	#a&	b#&
Chamar	Central	<i>Pager</i>	#c&	d#&
Posição	<i>Pager</i>	Central	p#&	#q@&

Note que “@” está representando a posição daquele *pager* na fila de espera correspondente.

Definidos os parâmetros supracitados, deve-se escolher quais são as configurações adequadas para o *transceiver*. De acordo com o datasheet fornecido pelo fabricante, verificou-se que as opções seguintes tinham que ser alteradas da configuração de fábrica:

- 1) Taxa de dados RF: 19,2 kbit/s
- 2) Tempo limite para receber pacote: 500 ms
- 3) *End character* para pacote: ‘&’

Isso garante que ao terminar a mensagem com o ‘&’, o *transceiver* envie imediatamente via rádio sem ter que esperar que o tempo limite para *buffer* do pacote ser atingido.

4.2.1. Diagrama de circuito e de PCB do módulo

Devido ao encapsulamento de características SMD do RC2500HP, foi necessário o desenvolvimento de uma placa de circuito impresso para melhor acessar os pinos do mesmo. Também se decidiu incorporar uma antena de menor ganho possível para o módulo diretamente na placa de circuito impresso. O ganho pequeno possibilita uma irradiação de sinal mais simétrica e mais adequada à aplicação.

Para o desenvolvimento das placas de circuito impresso, utilizou-se o software gratuito Eagle, da empresa Cadsoft, sendo que os diagramas do circuito elétrico e do circuito impresso encontram-se abaixo:

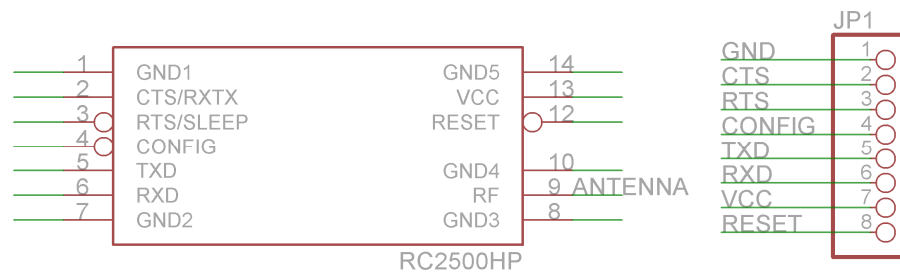


Figura 11 - Diagrama de circuito do módulo

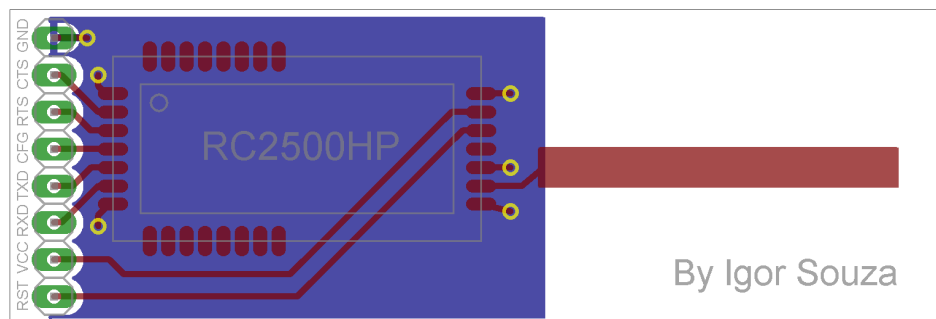


Figura 12 - Diagrama da PCB

4.3. SELEÇÃO DE HARDWARE DOS PAGERS

Como mencionado anteriormente, foi escolhido o microcontrolador ATmega88 TQFP devido à diferença de custos com relação ao DIP, mesmo considerando a maior dificuldade de montagem devido ao encapsulamento SMD. Optou-se também pela utilização de um oscilador externo de 16 MHz, não tanto pela velocidade do *clock*, mas pela estabilidade do mesmo. Utilizou um ressonador cerâmico visto que um cristal comum exigiria a inserção de dois capacitores extras no circuito.

Na parte de alimentação, escolheu-se reguladores lineares, como o comum LM7805 e o LM1117-3.3. Este último é um regulador *low-drop*, pois o *V_{in}* dele será 5V e só é permitida uma pequena queda de tensão na regulação de tensão.

4.2.1. Interface Homem-Máquina do Pager

A figura abaixo mostra uma idéia de como poderá ser a aparência dos pagers:



Figura 13 - Conceito para o pager

A interface homem-máquina do *pager* é bastante simples, ele possui apenas um botão que, quando pressionado, informa a posição do *pager* na fila correspondente através de três *LEDs*. A resolução dessa informação está limitada aos três primeiros da fila.

4.3. SELEÇÃO DE HARDWARE DA CENTRAL DE COMANDO

A central será responsável por todas as funções e rotinas de gerenciamento de filas. Nela o operador comandará, por exemplo, as funções de adicionar, remover ou consultar a lista por meio de botões e tendo um *display* LCD para exibir as informações relevantes.

O hardware escolhido para a central é idêntico ao dos *paggers*, entretanto há o acréscimo do *display LCD* de 16x2 e dos botões extras para o teclado. Outra diferença é a ausência dos *LEDS*, já que a interface será através do *LCD*.

4.3.1. Interface Homem-Máquina da Central

A figura abaixo mostra uma idéia de como poderá ser a central.



Figura 14 - Central de Comando

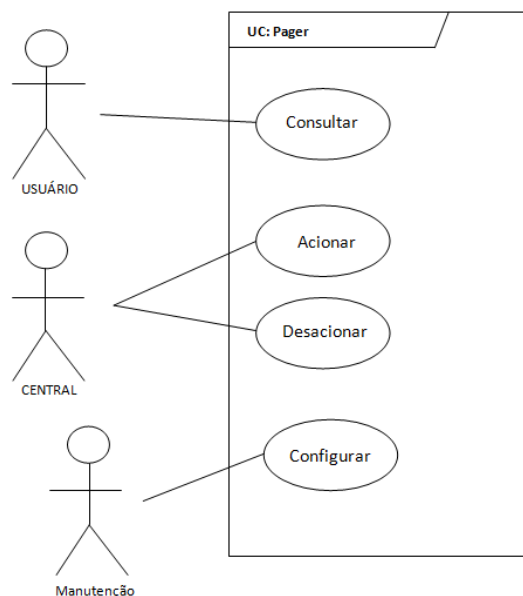
5. PROJETO DO *PAGER*

5.1. CASOS DE USO

5.1.1. Diagrama e Descrições

O caso de uso para os módulos dos *paggers* é bastante simples e está demonstrado no diagrama abaixo.

Diagrama 1 - Caso de Uso dos módulos dos paggers



Consultar: Permite ao usuário consultar a lista de espera e se informar de quantos números existem na sua frente.

Acionar Pager: Após o usuário pressionar o comando “chamar” na central, ela envia um sinal que acionará o *pager* correspondente.

Desacionar Pager: permite que o usuário através da central desacione o *pager*.

Estes dois últimos casos estão descritos em um diagrama de atividade próprio, na seção 5.1.2.

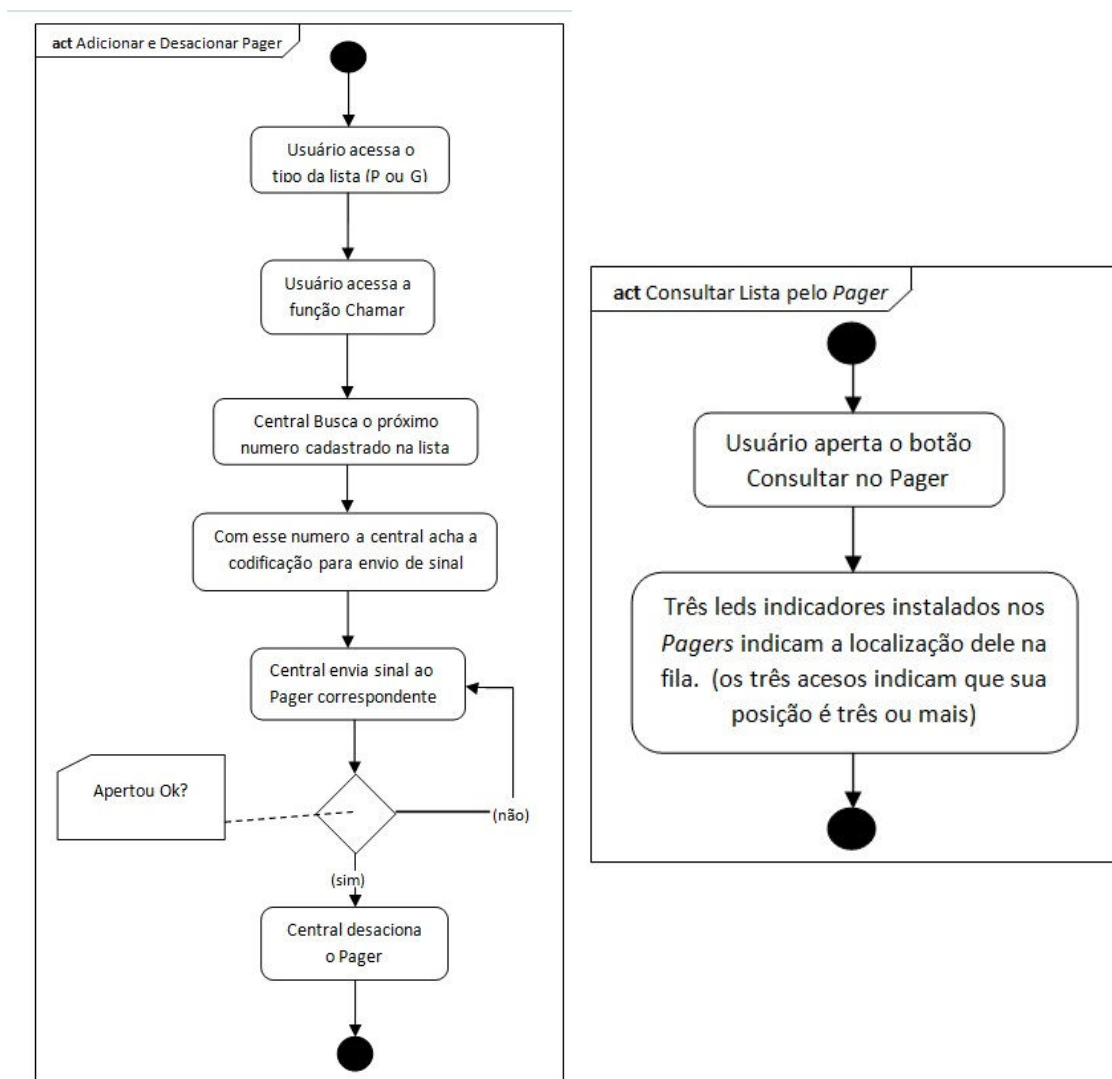
Configurar: O técnico responsável pela manutenção pode configurar o *pager* para ajustar seu funcionamento.

5.1.2. Diagrama de Atividade

Como citado previamente, os casos acima estão demonstrados em diagramas de atividade mostrados abaixo.

Quando a central deseja acionar o próximo *Pager* da lista de espera, basta acessar a lista desejada (pequena ou grande), e em seguida selecionar a função chamar. A central busca o número que está na lista, tal número remete a um código que foi previamente cadastrado na central, que corresponde ao endereço do *pager* a ser chamado. Esse ID faz com que o sinal seja aceito apenas por um módulo, ou seja, apenas um *pager* consegue interpretar o sinal como válido.

Diagrama 2 - Diagrama de atividades do pager



5.2.2. Lista de Materiais

Tabela 3 - Custo estimado dos componentes para o circuito do pager

QTDE	Símbolo	Componente	Valor	Fabricante	Preço Médio (R\$)
1	MEGA8-AI	ATMEGA88 TQFP	-	ATMEGA	4,80
1	JRADIO	RC2500HP	-	Radiocrafts	56,00
1	X1	Ressonador	16MHz	-	1,00
1	IC2	L7805CV	12V-5V	ST	0,40
1	-	Placa de Fenolite	-	-	0,70
1	-	Caixa de Plástico	PB112/2	Patola	12,00
1	IC3	LM1117T	5V-3,3V	National	6,80
1	BC548	Transistor	-	-	0,15
3	LED1, 2, 3	LED alto brilho	Vermelho	-	0,70
2	D2, D3	LED comum	Verde, Amarelo	-	0,20
1	D1	Diodo 1N4001	-	-	0,15
4	R2, R3, R4, R6	Resistor	1K	-	0,05
1	R5	Resistor	2K2	-	0,05
2	R1, R7	Resistor	10K	-	0,10
3	R9, R10, R11	Resistor	560Ω	-	0,05
3	C1, C2, C5	Capacitor	0,1μF	-	0,15
2	C3, C6	Capacitor	22μF	-	0,15
1	C4	Capacitor	100μF	-	0,20
3	ISP; JP1, JP2	Barra de Pino	-	-	0,80
3	Botões	Microswitches	-	-	0,20
					89,05

6. PROJETO DA CENTRAL DE COMANDO

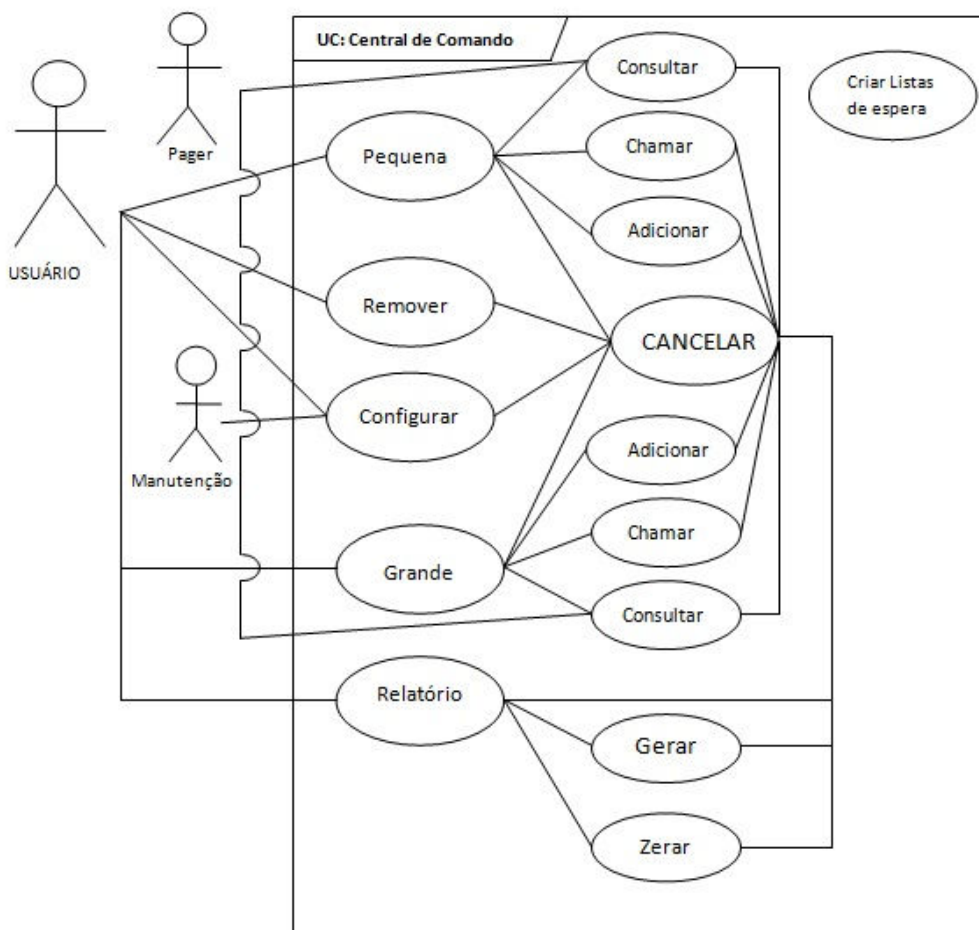
6.1. CASOS DE USO

6.1.1. Diagramas e Descrições

A central de comando é responsável por praticamente todas as funções realizadas pelo produto, monitorando, comunicando, calculando e armazenando todas as informações necessárias.

Seu diagrama de casos de uso está demonstrado abaixo.

Diagrama 3 - Caso de Uso da Central de Comando



Criar Lista de Espera: Existem duas listas internas que podem ser habilitadas se necessário (pequena e grande). O próprio microcontrolador irá construir as listas de espera de *pager*, sendo que elas começam vazias. Essas listas são apagadas toda vez que a central for desligada ou reiniciada.

Adicionar na Lista: Permite ao usuário adicionar o *pager* de identificação desejada na lista de espera acessada.

Remover da Lista: Permite ao usuário remover o *pager* de identificação desejada de qualquer lista de espera da central.

Chamar Pager: Permite ao usuário chamar um *pager* que está no topo da lista de espera correspondente, podendo excluir ou não o mesmo da fila. Não é necessário inserir nenhuma identificação para o *pager*, as listas rodam automaticamente.

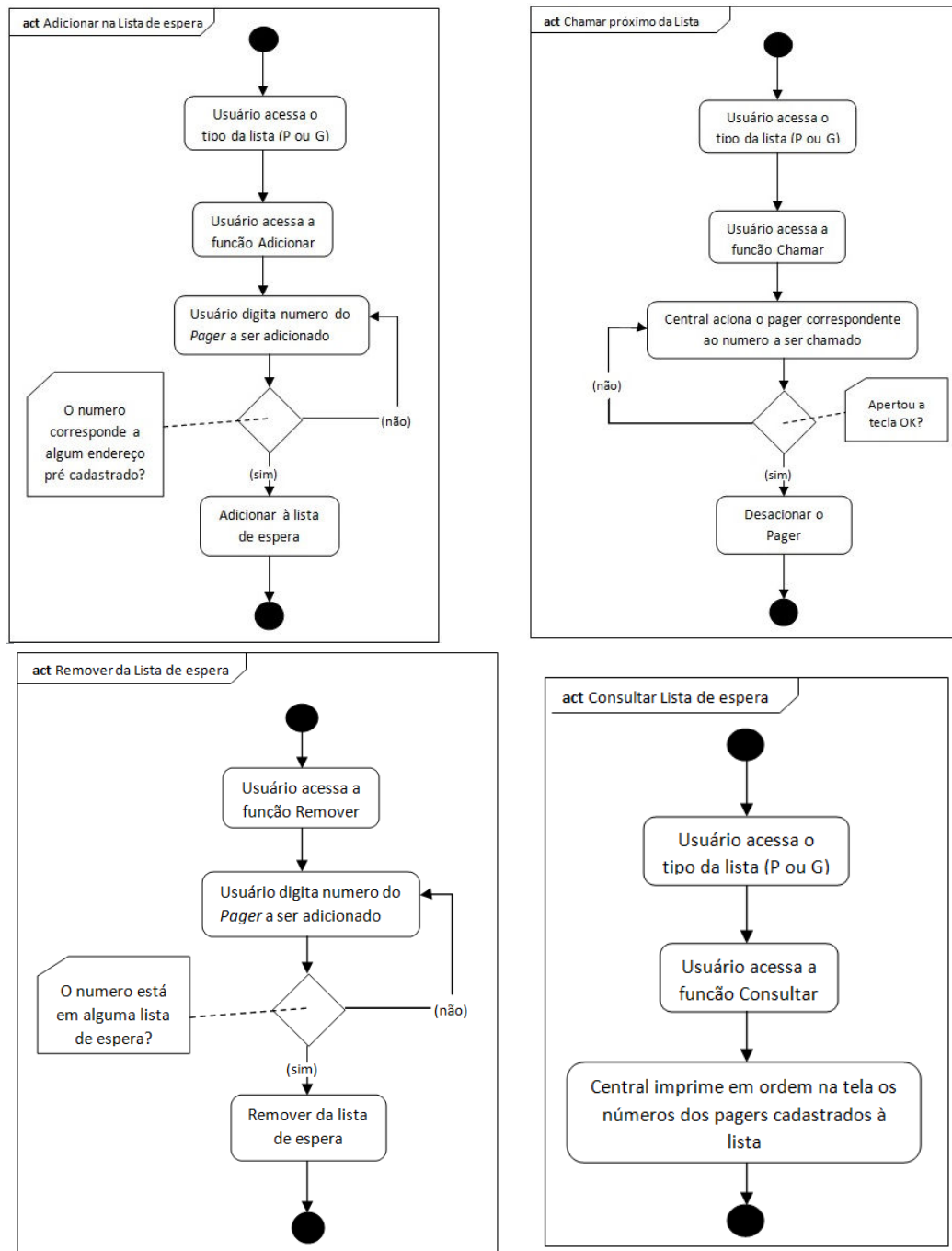
Consultar Lista: Permite aos usuários tanto da central como os dos pagers consultarem as listas de espera da central.

Gerar Relatório: Permite ao usuário ter conhecimento estatístico da movimentação de seu estabelecimento. Essa função dá acesso a informações como número total de pagers adicionados, número total de Pager removidos, tempo médio de espera por período de uma hora, número de pagers adicionados por período de uma hora e número total de pagers removidos por período de uma hora.

Zerar Relatório: Caso se deseje recomeçar o relatório, esta função zera as variáveis correspondentes, mas deixando a fila intacta.

6.1.2. Diagramas de Atividades

Diagrama 4 - Diagrama de atividades da central



As opções aparecem no *display* de LCD sempre em pares e com os botões de *Up* e *Down* é possível caminhar entre elas e para selecionar a função desejada, basta apertar a tecla *OK*. Para voltar, a lógica continua seguindo do mesmo jeito, basta apertar a tecla *Cancel*.

Caso o usuário aperte a tecla *Cancel* várias vezes, eventualmente ele voltará ao menu principal e a central ficará aguardando para realizar alguma outra operação. Ao final de qualquer operação com sucesso, a tela voltará ao menu principal.

Abaixo é exemplificado a estrutura básica da interface e seus comandos.

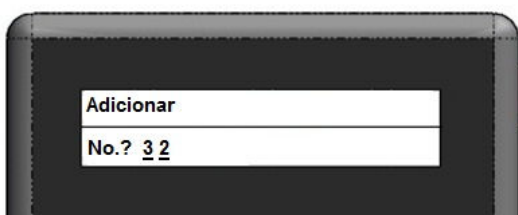
6.1.2.1. Adicionar Pager

Para adicionar um *pager* à Lista, deve-se seguir as seguintes etapas:



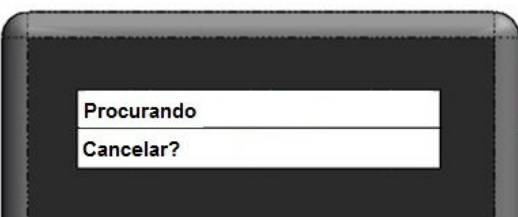
Etapas 1: Escolhe-se o tipo da fila onde o *pager* será adicionado (pequena ou grande), e em seguida a função adicionar.

Por exemplo, será adicionado o *pager* nº 32.

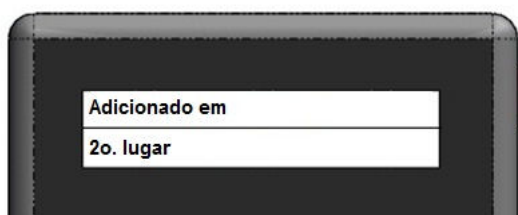


Etapas 2: O usuário escolhe o número através das teclas *Up* e *Down* e alterna entre as casas de dezena e unidade atrás das teclas *Left* e *Right*.

Terminado a escolha, o usuário deverá apertar a tecla *OK* para confirmar.



Etapas 3: Na tela aparecerá a mensagem: "Procurando" significando que a central está verificando se o *pager* está ligado e dentro dos limites de alcance.

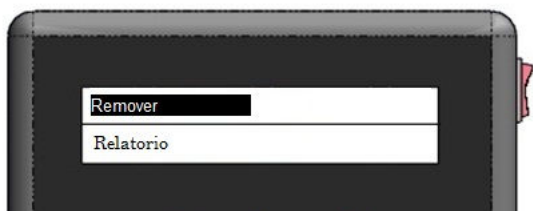


Etapas 4: Após a central reconhecer o *pager*, ela mostrará que ele foi adicionado e a sua posição na fila em que foi adicionado. Após 3 segundos, retorna-se ao menu principal.

6.1.2.2. Remover Pager

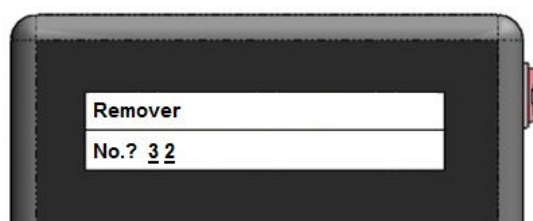
Para remover um *pager* não é necessário especificar a lista no qual ele está, o software faz uma busca entre elas e remove o *pager* escolhido automaticamente.

Para remover um *pager* da Lista, deve-se seguir as seguintes etapas:

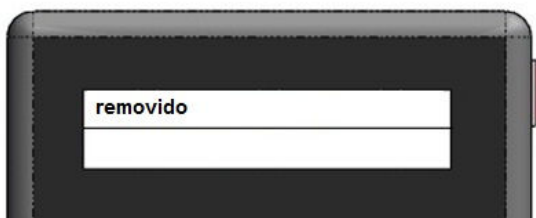


Etapa 1: Escolhe-se a função “Remover” no menu principal.

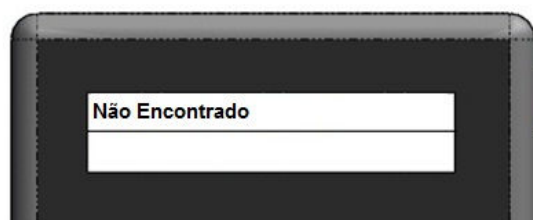
Será removido, por exemplo, o *pager* nº 32.



Etapa 2: Similar ao “Adicionar”, o usuário escolhe o número através das teclas direcionais e aperta OK para confirmar.



Etapa 3: Caso aquele número seja encontrado, aparecerá na tela a mensagem “Removido”, ela ficará na tela por 2 segundos e, em seguida, a tela volta para o menu principal. (*)



(*) Caso o número do *pager* escolhido não exista em nenhuma lista, a mensagem “Não Encontrado” aparecerá no display.

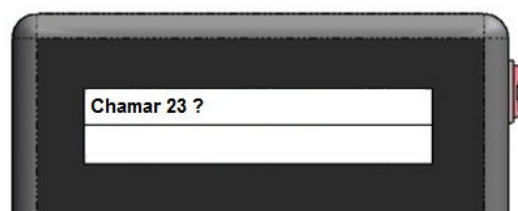
6.1.2.3. Chamar próximo pager das listas

Para chamar um *pager* de uma das listas, deve-se seguir as etapas:

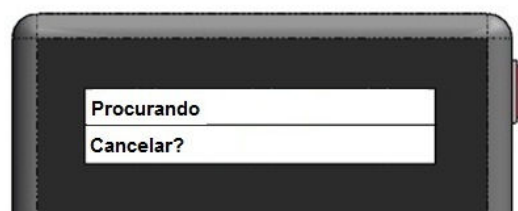


Etapas 1: Escolhe-se qual da lista que deseja-se chamar o próximo *pager* (pequena ou grande) e, em seguida, a função "Chamar".

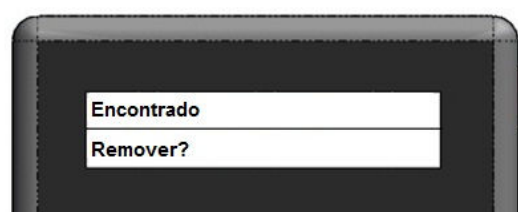
No exemplo, o *pager* nº 32 será acionado. (*)



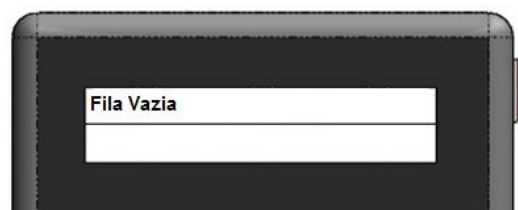
Etapas 2: Na tela aparecerá uma mensagem de confirmação, mostrando o número do primeiro *pager* da fila. Para acioná-lo, o usuário deve apertar a tecla OK.



Etapas 3: A central irá buscar o *pager* e acioná-lo. Caso o usuário queira cancelar a ação, basta apertar a tecla *Cancel* e, com isso, a tela voltará ao menu anterior.



Etapas 4: Após encontrar e acionar o *pager*, a central perguntará se deseja removê-lo da lista. Se sim, o usuário deverá apertar a tecla OK para confirmar, caso não queira, basta ele apertar a tecla *Cancel*, voltando para o menu anterior.



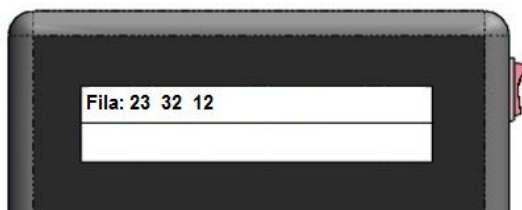
(*) Caso a lista de espera esteja vazia, a central mostrará a mensagem "Fila Vazia".

6.1.2.4. Consultar a lista

Para consultar um *pager* de uma das listas, deve-se seguir as etapas:

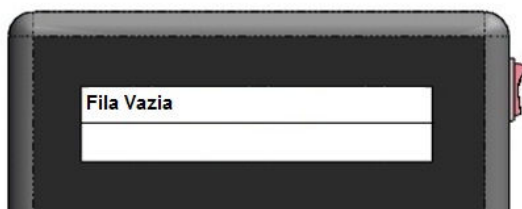


Etapas 1: Escolhe-se o tipo da lista que deseja-se consultar (pequena ou grande), e em seguida a função "Consultar".



Etapas 2: Na tela, será listado todos os *paggers* inclusos naquela fila de espera, sendo que os primeiros da lista estão no esquerdo da tela. As teclas *Left* e *Right* são utilizadas para percorrer a lista.

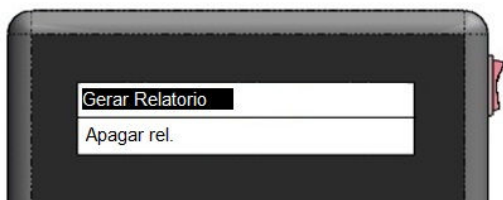
Para voltar ao menu inicial, o usuário deve apertar a tecla *OK* ou *Cancel*. (*)



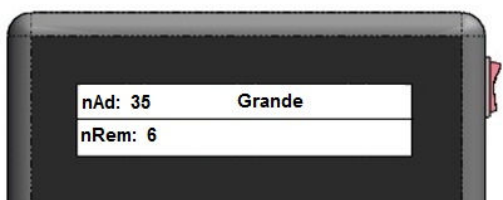
(*) Caso a lista de espera esteja vazia, a central mostrará a mensagem "Fila Vazia".

6.1.2.5 Gerar Relatório

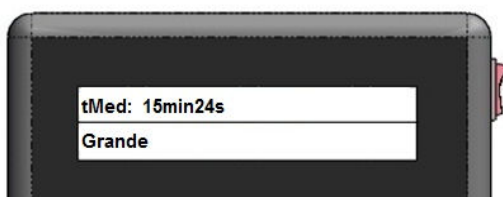
Para gerar o relatório da movimentação, deve-se seguir as seguintes etapas:



Etapa 1: Escolhe-se “Relatório” no menu principal.



Etapa 2: Serão mostrados dados como “nAd” (número total de *pagars* adicionados), “nRd” (número total de *pagars* removidos) e o tipo da fila de espera.

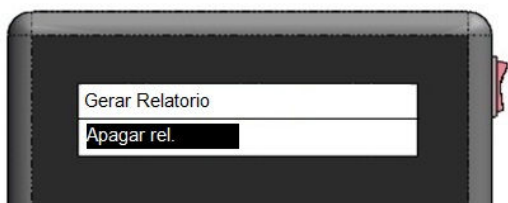


Apertando as teclas *Up* e *Down* aparecerá outra página mostrando “tMed” (tempo médio de espera), assim como a fila ao qual o dado se refere.

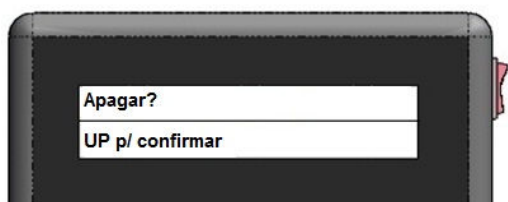
Para voltar ao menu inicial, o usuário deve apertar a tecla *OK* ou *Cancel*.

6.1.2.6 Apagar Relatório

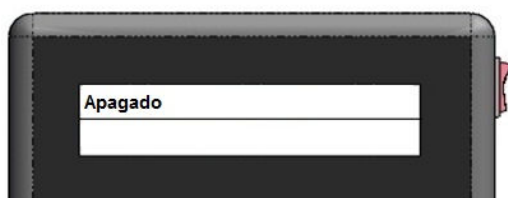
Para apagar a memória onde são armazenados os dados que geram o relatório, segue-se os seguintes passos:



Etapa 1: Escolhe-se “Apagar relat.” no menu principal.



Etapa 2: Na tela irá aparecer uma mensagem para confirmação, para garantir que o usuário realmente deseja apagar o relatório. Para confirmar basta apertar a tecla *Up*.



Etapa 3: A central informa que o relatório foi apagado. Ela exibe essa mensagem durante 2 segundos e volta ao menu principal.

6.2. HARDWARE DA CENTRAL

6.2.1. Diagrama de Circuito

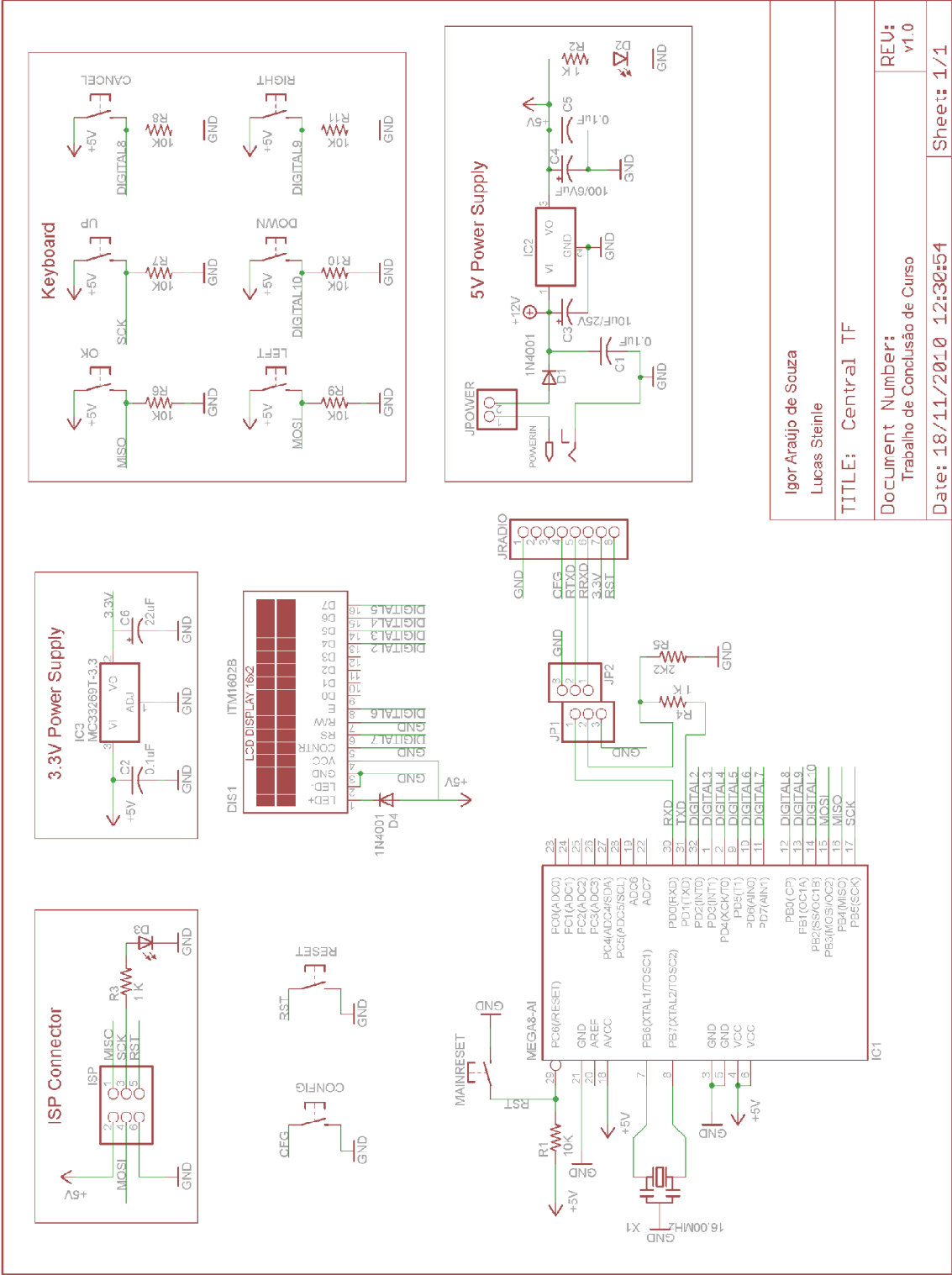


Figura 17 - Diagrama de circuito da central

6.2.2. Diagrama da PCB

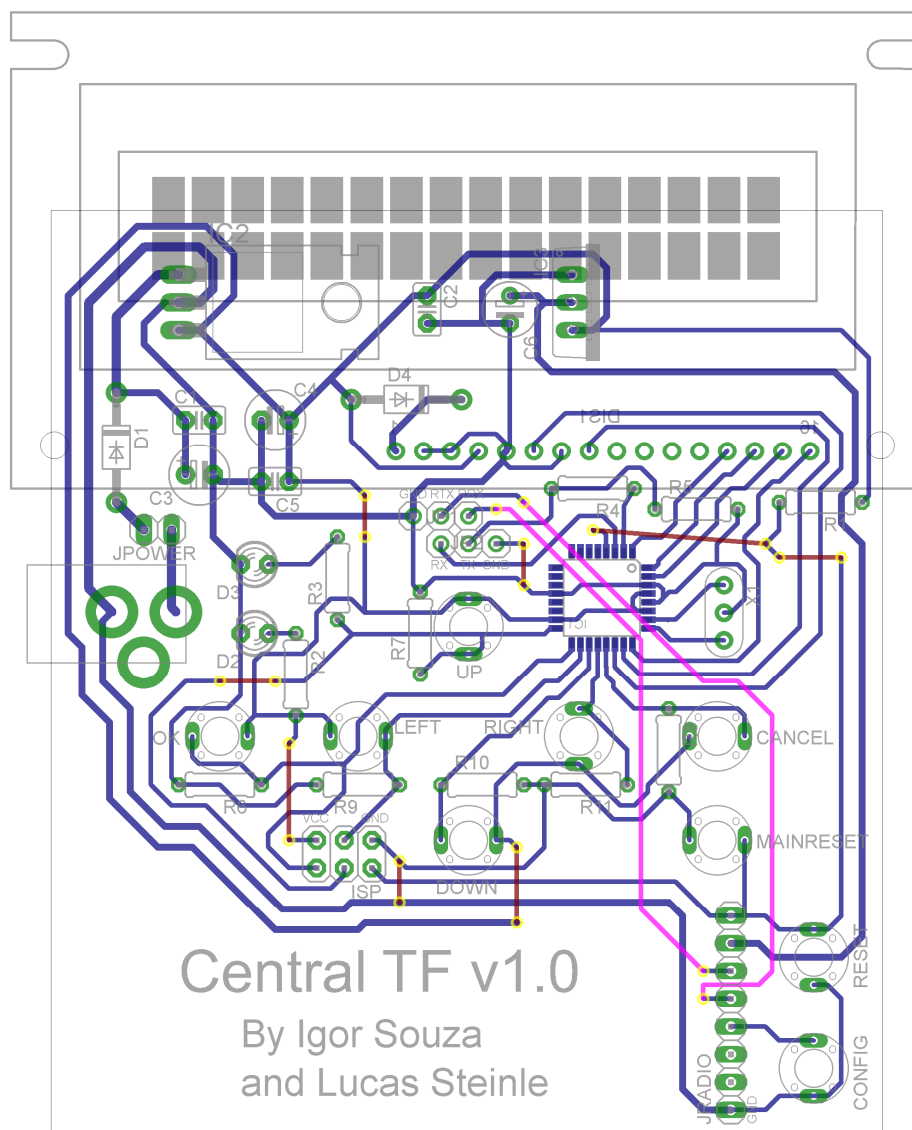


Figura 18 - Diagrama da PCB da central

6.2.3. Lista de Materiais

Tabela 4 - Custo estimado dos componentes para o circuito da central

QTDE	Símbolo	Componente	Valor	Fabricante	Preço (R\$)
1	MEGA8-AI	ATMEGA88 TQFP	-	ATMEGA	4,80
1	JRADIO	RC2500HP	-	Radiocrafts	56,00
1	X1	Ressonador	16MHz	-	1,00
1	DIS1	Display LCD	16-2seg	InTech	17,00
1	IC2	L7805CV	12V-5V	ST	0,40
1	-	Placa de Fenolite	-	-	0,70
1	-	Caixa de Plástico	PB114	Patola	15,00
1	IC3	LM1117T	5V-3,3V	National	6,80
3	LED1, 2, 3	LED alto brilho	Vermelho	-	0,70
2	D2, D3	LED comum	Verde, Amarelo	-	0,20
1	D1	Diodo 1N4001	-	-	0,15
4	R2-R4	Resistor	1K	-	0,05
2	R1, R6-R11	Resistor	10K	-	0,10
1	R5	Resistor	2K2	-	0,05
3	C1, C2, C5	Capacitor	0,1µF	-	0,15
2	C3, C6	Capacitor	22µF	-	0,15
1	C4	Capacitor	100µF	-	0,20
3	ISP; JP1,JP2	Barra de Pino	-	-	0,80
9	Botões	Microswitches	-	-	0,20
					109,95

7. FABRICAÇÃO DOS PROTÓTIPOS

Para a fabricação das placas de circuito impresso utilizou-se o processo de transferência térmica de toner, através da impressão em papel fotográfico e posterior corrosão com peróxido de ferro. Atingiu-se excelentes resultados, com resolução de trilhas de até 0,025mm (ou um milésimo de polegada).

7.1. CONFEÇÃO DE PCBS POR TRANSFERÊNCIA TÉRMICA DE TONER

A confecção de PCBs pelo método citado segue o seguinte procedimento:

- 1) Inicialmente, é necessário a impressão do circuito em papel fotográfico (glossy), em uma impressora a laser. Deve-se atentar para caso haja a necessidade da matriz estar espelhada. Evidentemente, é necessário a placa cobreada para circuito impresso:



Figura 19 - Matriz do circuito e placa cobreada

- 2) Uma vez com esse material, é feito a transferência térmica do toner, posicionando a matriz no local correto e aplicando calor em cima do papel. Com a exposição correta à alta temperatura, se consegue com quase totalidade a transferência do toner do papel para a placa, entretanto deve-se aguardar o resfriamento da placa para poder retirar o papel:



Figura 20 - Transferência térmica e remoção do papel

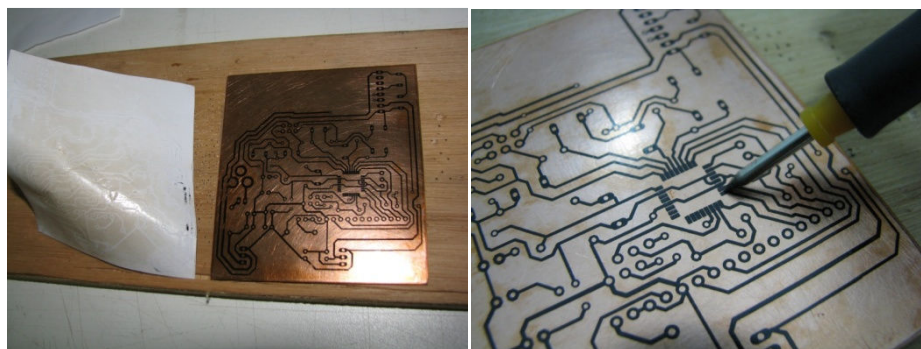


Figura 21 - Papel removido e detalhes da resolução alcançada

- 3) Checa-se se há falhas no procedimento, retirando pontes indevidas ou refazendo trilhas incompletas com caneta de tinta permanente (caneta para retro-projetor). Em seguida, corrói-se a placa, limpando o toner no fim do processo com acetona. Finalmente fura-se a placa e se for desejado, pode-se repetir parte do processo para aplicar um “silkscreen”, com os locais dos componentes e similares:

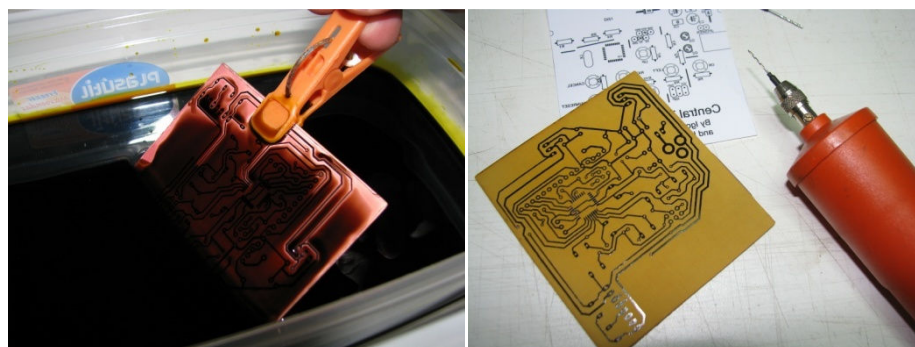


Figura 22 - Processo de corrosão e a placa já corroída

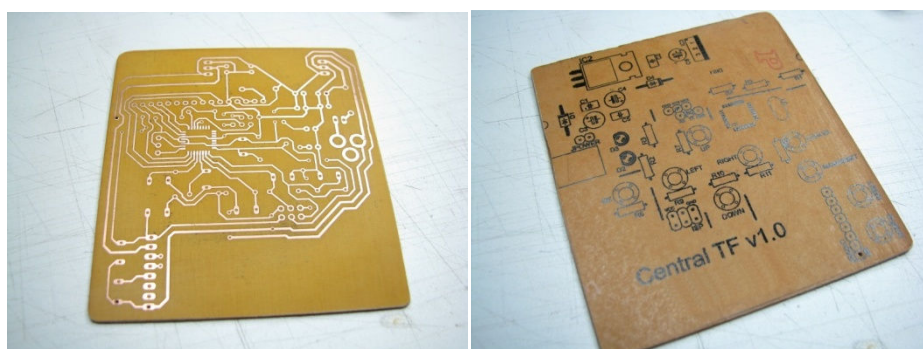


Figura 23 - Placa finalizada, pronta para furação

7.2. BREAKOUT BOARD RC2500HP

Placa “adaptadora” para o módulo RC2500HP, com antena incorporada à placa.

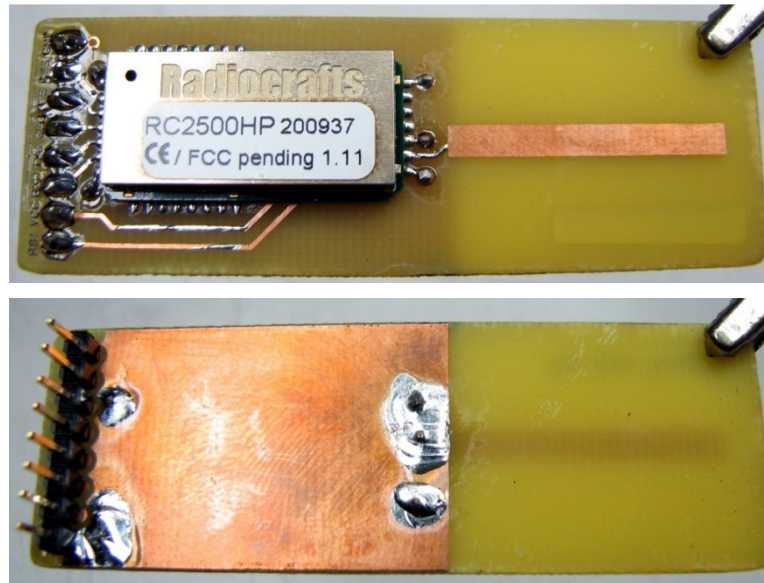


Figura 24 - Placa para o módulo RC2500HP

7.3. CENTRAL

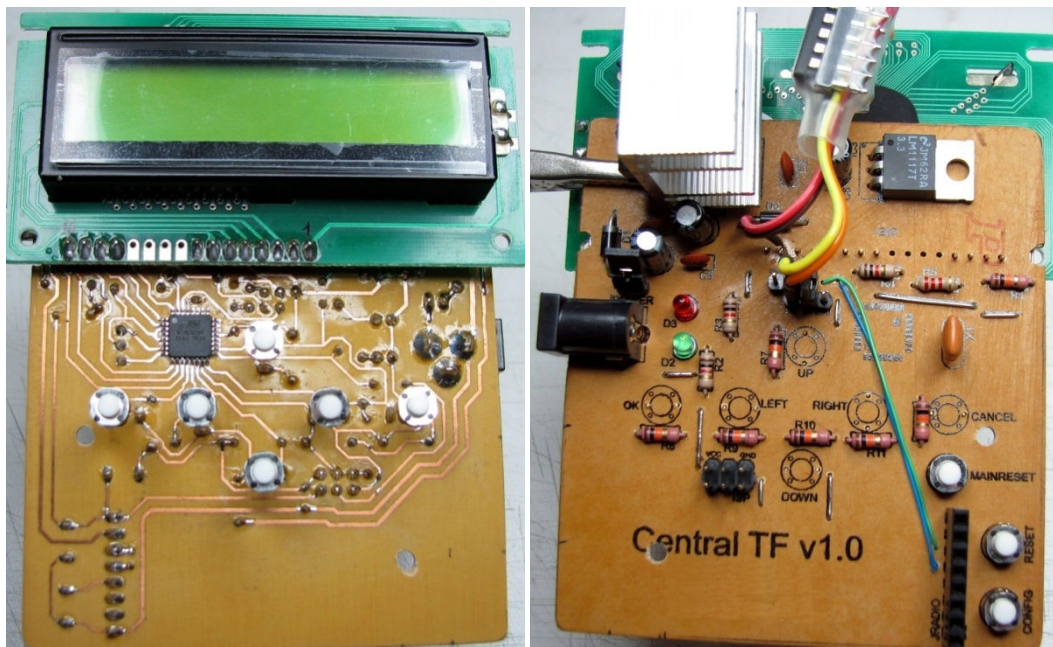


Figura 25 - Placa da Central pronta

7.4. PAGER

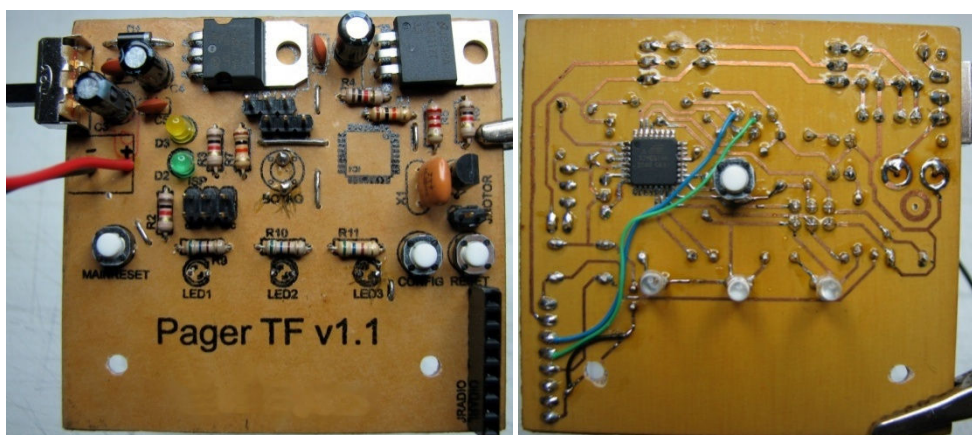


Figura 26 - Placa de um dos pagers pronta

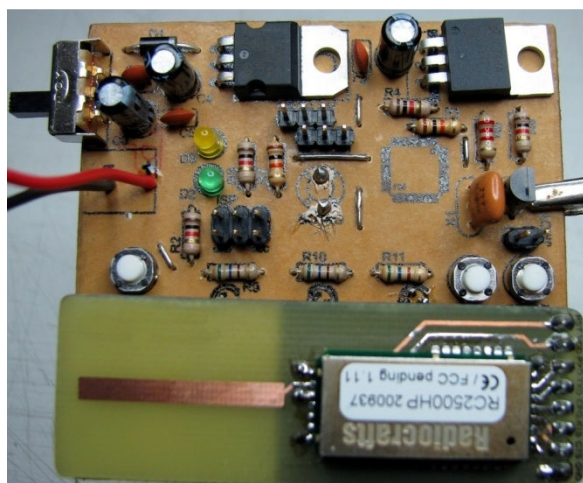


Figura 27 - Placa já com o módulo no lugar

8. IMPLEMENTAÇÃO DO SOFTWARE

Apesar de todo o estudo de casos, houve um problema sério na estimativa inicial do tamanho do programa da central, sendo que a programação proposta nesse trabalho ultrapassou os 10 kbytes. Devido ao limite de memória do Atmega88 ser de 8 kbytes, o programa teve suas funcionalidades reduzidas para apenas uma fila de espera e com simplificações no texto mostrado no LCD. Foi conseguido um tamanho final para o programa da central de aproximadamente 7300 bytes.

O código fonte do programa da central e dos pagers estão apresentados nos Anexos 1 e 2, respectivamente.

9. DISCUSSÃO DO PROTÓTIPO

O início do projeto exigiu cuidado na pesquisa para definir os métodos de comunicação e controle do sistema, pois existem muitas possibilidades e variáveis como, por exemplo, custo, facilidade de implementação, capacidade de armazenamento e confiabilidade, todos fatores que afetam a decisão final.

A construção dos equipamentos demandou um tempo além do esperado, principalmente devido à necessidade de aperfeiçoar a técnica para confeccionar as placas de circuito impresso. O uso de alguns componentes SMD exigiu um cuidado a mais na soldagem dos mesmos, entretanto o esforço se mostrou recompensador tendo em vista a redução de custos com uma peça fundamental do circuito.

No desenvolvimento dos protótipos viu-se que a experiência necessária para soldar os módulos de radio nos circuitos é bastante alta, pois qualquer mau posicionamento pode deixar trilhas em curto.

A escolha equivocada do microcontrolador impediu a plena implementação do software pretendido, fato que poderia ter sido evitado caso um ATmega328 ou até mesmo um ATmega168 tivesse sido utilizado no lugar do ATmega88.

A meta de alcance inicial não foi alcançada, chegando a apenas 40% do valor pretendido. Tal fato possivelmente está ligado escolha de antena, que não se adequou corretamente ao caso.

10. CONCLUSÃO

A pesquisa realizada em estabelecimentos que possuem produtos similares mostrou que a concorrência no mercado brasileiro é pequena, possibilitando o real lançamento de um produto similar.

O funcionamento do dispositivo para gestão de fila de espera se mostrou muito promissor se for escolhida uma antena adequada para a atividade. A pesquisa também mostrou que estabelecimentos que não possuem nenhum tipo de controle da fila de espera adquiririam um produto para essa função, caso o preço for acessível.

11. BIBLIOGRAFIA

(1) PORTA PARALELA.

Disponível em: <<http://www.rogercom.com/pparalela/LptWireless.htm>>

Último Acesso em: 28/06/2010

(2) BC648 DATA SHEET

Disponível em: <<http://www.datasheetcatalog.org/datasheet/siemens/BC639.pdf>>

Último Acesso em: 18/06/2010

(3) MC145026 DATA SHEET

Disponível em: <<http://www.datasheetcatalog.org/datasheet2/4/09gy1ga79yr6l4igj17j3sggtsfy.pdf>>

Último Acesso em: 28/06/2010

(4) RADIOCRAFTS

Disponível em: <<http://www.radiocrafts.com/>>

Último Acesso em: 20/06/2010

(5) RR3 DATA SHEET

Disponível em: <<http://www.telecontrolli.com/download/product-datasheets/receivers/14-rr3/download.html>>

Último Acesso em: 20/06/2010

(6) RT4 DATA SHEET

Disponível em: <www.telecontrolli.com/download/product-datasheets/transmitters/108-rt4/download.html>

Último Acesso em: 20/06/2010

(7) ARDUINO

Disponível em: <<http://www.arduino.cc/>>

Último Acesso em: 28/04/2010

(8) LASER TRIMMING

Disponível em: <<http://www.ls-laser-systems.com/>>

Último Acesso em: 21/05/2010

(9) LS LASER SYSTEMS GMBH

Disponível em: <http://www.gsig.com/system_circuit_trimming.html>

Último Acesso em: 21/09/2010

(10) ZigBee 2006 Protocol Stack

Disponível em:

<http://www.microchip.com/Microchip.WWW.SecureSoftwareList/secsoftwaredownload.aspx?device=en538049&lang=en&ReturnURL=http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=1406&dDocName=en538049#>

Último Acesso em: 20/06/2010

(11) SOLDAGEM EM SMD

Disponível em: < <http://www.eletronica.com/soldagem-em-smd/>>

Último Acesso em: 20/11/2010

(12) RC232 USER MANUAL

Disponível em: < http://www.radiocrafts.com/uploads/rc232_user_manual_1_5.pdf>

Último Acesso em: 23/11/2010

(13) CREATING PCBs WITH THE TONER TRANSFER METHOD

Disponível em: < <http://www.dr-lex.be/hardware/tonertransfer.html>>

Último Acesso em: 18/11/2010

(14) ANTENNA PATTEERNS AND THEIR MEANING

Disponível em: < http://www.cisco.com/en/US/prod/collateral/wireless/ps7183/ps469/prod_white_paper0900aecd806a1a3e.html>

Último Acesso em: 16/11/2010

(15) UNDERSTANDING ANTENNA GAIN.

Disponível em: < <http://www.rcexplorer.se/page14/gain/gain.html>>

Último Acesso em: 08/11/2010

ANEXO 1 – CÓDIGO FONTE DA CENTRAL

```

/*****
/*  Trabalho Final - PMR2550 (2010)
/*
/*  São Paulo, 1 de dezembro de 2010
/*
/*  Dispositivo para gestao de fila de espera
/*  via sistema de pagers.
/*
/*  Orientador: Prof. Marcos Barretto
/*  Alunos: Igor Araújo de Souza - 5694199
/*          Lucas Steinle Duarte Silva - 5696159
/*
/*  Programa: CENTRAL
/*  Versao: 5.2 (2010/nov/29)
/*
*****/
#include <LiquidCrystal.h>

#define RS 7          // Pinos do LCD
#define ENABLE 6
#define D4 2
#define D5 3
#define D6 4
#define D7 5

#define B_OK 8        // Pinos do teclado
#define B_CANCEL 12
#define B_UP 13
#define B_DOWN 10
#define B_RIGHT 11
#define B_LEFT 9

#define LPRINC 6      //No. de linhas no menu principal

// inicializacao do LCD com os pinos correspondentes
LiquidCrystal lcd(RS, ENABLE, D4, D5, D6, D7);

boolean atualiza = true, //atualizacao do LCD
        digito = 0;      //digito do cursor
char endereco,           //endereço da mensagem recebida
    codigo = 'n',        //comando interpretado via serial
    linha = 1,           //linha atual do menu
    dezena = 0,          //armazenamento da dezena
    unidade = 0,         //idem para unidade
    temp,               //variavel para manipulacao temporaria
    nFila = 0,          //proximo lugar na fila
    fila[99],           //armazena enderecos dos pagers
    nAdic = 0,          //total de adicionados
    nRemov = 0,         //total de chamados/removidos
    opcao,              //define qual o menu atual
    incomingByte;        //para mensagem recebida
unsigned long tFila[99], //armazena a hora de adicao na fila
        tMedio;         //tempo medio entre chamadas

void setup() {
    Serial.begin(19200); //Inicializacao da serial e
    lcd.begin(16, 2);    // do LCD
    lcd.blink();
    //Adicao de dois pagers para demonstracao
    temp = adicionarPager(23);
    temp = adicionarPager(51);
}

void loop() {
    if(opcao == 0){ //Menu Principal
        if(digitalRead(B_UP) == HIGH){
            linha--;
            if(linha < 1)
                linha = LPRINC;
            while(digitalRead(B_UP) == HIGH);
            delay(200);
            atualiza = true;
        }//if
    }
}
```



```

else if(digitalRead(B_DOWN) == HIGH){
    linha++;
    if(linha > LPRINC)
        linha = 1;
    while(digitalRead(B_DOWN) == HIGH);
    delay(200);
    atualiza = true;
} //else if
else if(digitalRead(B_OK) == HIGH){
    opcao = linha;
    linha = 1;
    while(digitalRead(B_OK) == HIGH);
    delay(200);
    atualiza = true;
} //else if
else if (atualiza == true){ //Atualiza LCD
    if (linha == 1 || linha == 2){ //pagina 1 do menu principal
        lcd.clear();
        lcd.print("Chamar");
        lcd.setCursor(0,1);
        lcd.print("Adicionar");
    } //if
    else if (linha == 3 || linha == 4){ //pagina 2 do menu principal
        lcd.clear();
        lcd.print("Fila");
        lcd.setCursor(0,1);
        lcd.print("Remover");
    } //else if
    else if (linha == 5 || linha == 6){ //pagina 3 do menu principal
        lcd.clear();
        lcd.print("Relatorio");
        lcd.setCursor(0,1);
        lcd.print("Apagar rel.");
    }
    lcd.setCursor(0, (linha+1)%2);
    atualiza = false;
} //if atualiza
} //else if Menu Principal

else if(opcao == 1){ //Chamar pager atual
    if(digitalRead(B_OK) == HIGH){
        opcao = 11;
        while(digitalRead(B_OK) == HIGH);
        delay(200);
        atualiza = true;
    } //else if
    else if(digitalRead(B_CANCEL) == HIGH){
        opcao = 0;
        while(digitalRead(B_CANCEL) == HIGH);
        delay(200);
        atualiza = true;
    } //else if
    else if(nFila < 1){
        lcd.clear();
        lcd.print("Fila vazia");
        delay(2000);
        opcao = 0;
    }
    else if(atualiza == true){
        lcd.clear();
        lcd.print("Chamar ");
        lcd.print(fila[0], DEC);
        lcd.print("?");
        atualiza = false;
    } //else if atualiza
} //else if Chamar

else if(opcao == 11){ //Chamando...
    if(digitalRead(B_CANCEL) == HIGH){
        opcao = 1;
        while(digitalRead(B_CANCEL) == HIGH);
        delay(200);
        atualiza = true;
    } //if
    else if(atualiza == true){
        Serial.print(fila[0]);
        Serial.print("c&");
    }
}

```

```

        lcd.clear();
        lcd.print("Procurando");
        lcd.setCursor(6,1);
        lcd.print("Cancelar?");
        atualiza = false;
    } //else if
    else if(codigo == 'd' && endereco == fila[0]){
        lcd.clear();
        lcd.print("Encontrado");
        lcd.setCursor(0,1);
        lcd.print("Remover?");
        codigo = 'n';
    } //else if
    else if(endereco == fila[0] && digitalRead(B_OK) == HIGH){ //se for para remover
        tMedio = tMedio + (millis() - tFila[0]);
        temp = removerPager(fila[0]);
        lcd.clear();
        lcd.print("Removido");
        opcao = 0;
        delay(2000);
        while(digitalRead(B_OK) == HIGH);
        delay(200);
        atualiza = true;
    } //else if
} //else if Chamando

else if(opcao == 2){ //Adicionar pager
    if(digito)
        unidade = linha;
    else
        dezena = linha;
    if(digitalRead(B_OK) == HIGH){
        opcao = 21;
        while(digitalRead(B_OK) == HIGH);
        delay(200);
        atualiza = true;
    } //if
    else if(digitalRead(B_CANCEL) == HIGH){
        opcao = 0;
        linha = 2;
        while(digitalRead(B_CANCEL) == HIGH);
        delay(200);
        atualiza = true;
    } //else if
    else if(digitalRead(B_UP) == HIGH){
        linha++;
        if(linha > 9)
            linha = 0;
        while(digitalRead(B_UP) == HIGH);
        delay(200);
        atualiza = true;
    } //else if
    else if(digitalRead(B_DOWN) == HIGH){
        linha--;
        if(linha < 0)
            linha = 9;
        while(digitalRead(B_DOWN) == HIGH);
        delay(200);
        atualiza = true;
    } //elseif
    else if(digitalRead(B_LEFT) == HIGH){
        digito = 0;
        linha = dezena;
        while(digitalRead(B_LEFT) == HIGH);
        delay(200);
        atualiza = true;
    } //else if
    else if(digitalRead(B_RIGHT) == HIGH){
        digito = 1;
        linha = unidade;
        while(digitalRead(B_RIGHT) == HIGH);
        delay(200);
        atualiza = true;
    } //else if
    else if(atualiza == true){
        lcd.clear();
        lcd.print("Adicionar");
    }
}

```

```

        lcd.setCursor(0,1);
        lcd.print("No.? ");
        lcd.print(dezena, DEC);
        lcd.print(unidade, DEC);
        temp = 5 + digito;
        lcd.setCursor(temp,1);
        atualiza = false;
    } //if atualiza
} //else if Adicionar

else if(opcao == 21){ //Adicionando...
    if(digitalRead(B_CANCEL) == HIGH){
        opcao = 2;
        linha = dezena;
        digito = 0;
        while(digitalRead(B_CANCEL) == HIGH);
        delay(200);
        atualiza = true;
    } //else if
    else if(atualiza == true){ //verifica-se se o pager esta ligado
        temp = dezena*10 + unidade;
        Serial.print(temp);
        Serial.print("a&");
        lcd.clear();
        lcd.print("Procurando");
        lcd.setCursor(6,1);
        lcd.print("Cancelar?");
        atualiza = false;
    }
    else if(codigo == 'b' && endereco == temp){ //se estiver no alcance
        fila[nFila] = endereco;
        tFila[nFila] = millis();
        nFila++;
        nAdic++;
        lcd.clear();
        lcd.print("Adicionado em");
        lcd.setCursor(0,1);
        lcd.print(nFila, DEC);
        lcd.print("o. lugar");
        delay(3000);
        opcao = 0;
        linha = 2;
        codigo = 'n';
        atualiza = true;
    }
} //else if Adicionando

else if(opcao == 3){ //Mostrar fila de espera
    if(nFila < 1){
        lcd.clear();
        lcd.print("Fila vazia");
        delay(2000);
        opcao = 0;
        linha = 3;
    } //if
    else if(digitalRead(B_OK) == HIGH || digitalRead(B_CANCEL) == HIGH){
        opcao = 0;
        linha = 3;
        while(digitalRead(B_OK) == HIGH);
        delay(200);
        atualiza = true;
    } //else if
    else if(digitalRead(B_LEFT) == HIGH){
        linha--;
        if(linha < 1)
            linha = nFila;
        while(digitalRead(B_LEFT) == HIGH);
        delay(200);
        atualiza = true;
    } //else if
    else if(digitalRead(B_RIGHT) == HIGH){
        linha++;
        if(linha > nFila)
            linha = 1;
        while(digitalRead(B_RIGHT) == HIGH);
        delay(200);
        atualiza = true;
    }
}

```

```

    }//else if
    else if (atualiza == true){
        lcd.clear();
        lcd.print("Fila:");
        lcd.setCursor(0,1);
        lcd.print(fila[linha - 1], DEC);
        lcd.print(" ");
        lcd.print(fila[linha], DEC);
        lcd.print(" ");
        lcd.print(fila[linha + 1], DEC);
        lcd.print(" ");
        lcd.print(fila[linha + 2], DEC);
        lcd.print(" ");
        lcd.print(fila[linha + 3], DEC);
        atualiza = false;
    }//if atualiza
} //else if Mostrar

else if(opcao == 4){ //Remover um pager da fila
    if(digito)
        unidade = linha;
    else
        dezena = linha;
    if(nFila < 1){
        lcd.clear();
        lcd.print("Fila vazia");
        delay(2000);
        opcao = 0;
        linha = 4;
    }//if
    else if(digitalRead(B_OK) == HIGH){
        opcao = 41;
        while(digitalRead(B_OK) == HIGH);
        delay(200);
        atualiza = true;
    }//else if
    else if(digitalRead(B_CANCEL) == HIGH){
        opcao = 0;
        linha = 4;
        while(digitalRead(B_CANCEL) == HIGH);
        delay(200);
        atualiza = true;
    }//else if
    else if(digitalRead(B_UP) == HIGH){
        linha++;
        if(linha > 9)
            linha = 0;
        while(digitalRead(B_UP) == HIGH);
        delay(200);
        atualiza = true;
    }//if
    else if(digitalRead(B_DOWN) == HIGH){
        linha--;
        if(linha < 0)
            linha = 9;
        while(digitalRead(B_DOWN) == HIGH);
        delay(200);
        atualiza = true;
    }//if
    else if(digitalRead(B_LEFT) == HIGH){
        digito = 0;
        linha = dezena;
        while(digitalRead(B_LEFT) == HIGH);
        delay(200);
        atualiza = true;
    }//else if
    else if(digitalRead(B_RIGHT) == HIGH){
        digito = 1;
        linha = unidade;
        while(digitalRead(B_RIGHT) == HIGH);
        delay(200);
        atualiza = true;
    }//else if
    else if(atualiza == true){ //pergunta qual o pager
        lcd.clear();
        lcd.print("Remover");
        lcd.setCursor(0,1);

```

```

        lcd.print("No.? ");
        lcd.print(dezena, DEC);
        lcd.print(unidade, DEC);
        temp = 5 + digito;
        lcd.setCursor(temp,1);
        atualiza = false;
    } //else if atualiza
} //else if Remover

else if(opcao == 41){ //Removendo...
    temp = dezena*10 + unidade;
    if(removerPager(temp)){
        lcd.clear();
        lcd.print(temp, DEC);
        lcd.print(" removido");
        delay(2000);
        opcao = 0;
        linha = 4;
    } //if remover
    else{
        lcd.clear();
        lcd.print("Nao encontrado");
        delay(3000);
        opcao = 4;
    } //else
} //else if Removendo

else if(opcao == 5){ //Relatorio
    if(digitalRead(B_OK) == HIGH || digitalRead(B_CANCEL) == HIGH){
        opcao = 0;
        linha = 5;
        while(digitalRead(B_OK) == HIGH);
        delay(200);
        atualiza = true;
    } //if
    else if(digitalRead(B_UP) == HIGH){
        linha--;
        if(linha < 0)
            linha = 1;
        while(digitalRead(B_UP) == HIGH);
        delay(200);
        atualiza = true;
    } //else if
    else if(digitalRead(B_DOWN) == HIGH){
        linha++;
        if(linha > 1)
            linha = 0;
        while(digitalRead(B_DOWN) == HIGH);
        delay(200);
        atualiza = true;
    } //else if
    else if (atualiza == true){ //Atualiza LCD
        if (linha == 0){
            lcd.clear();
            lcd.print("nAd: ");
            lcd.print(nAdic, DEC);
            lcd.setCursor(0,1);
            lcd.print("nRem: ");
            lcd.print(nRemov, DEC);
        } //if
        else if (linha == 1){
            temp = (tMedio/(nAdic - nRemov))/1000;
            lcd.clear();
            lcd.print("tMed: ");
            lcd.print(temp/60, DEC);
            lcd.print("min");
            lcd.print(temp%60, DEC);
            lcd.print("s");
        } //else if
        atualiza = false;
    } //if atualiza
} //else if Relatorio

else if(opcao == 6){ //Apagar relatorio
    if(digitalRead(B_UP) == HIGH){
        nAdic = 0;
        nRemov = 0;
    }
}

```

```

        tMedio = 0;
        lcd.clear();
        lcd.print("Apagado");
        delay(2000);
        while(digitalRead(B_UP) == HIGH);
        delay(200);
        atualiza = true;
        opcao = 0;
    } //if
    else if(digitalRead(B_CANCEL) == HIGH){
        opcao = 0;
        linha = 6;
        while(digitalRead(B_CANCEL) == HIGH);
        delay(200);
        atualiza = true;
    } //else if
    else if (atualiza == true){ // Atualiza LCD
        lcd.clear();
        lcd.print("Apagar?");
        lcd.setCursor(0,1);
        lcd.print("UP p/ confirmar");
        atualiza = false;
    } //if atualiza
} //else if Apagar

if(Serial.available() > 0){ //Verifica se ha algo disponivel na serial
    delay(10); //tempo necessario para toda a mensagem ser armazenada no buffer
    incomingByte = Serial.read();
    if(incomingByte != ' ' && (incomingByte=='b' || incomingByte=='d' || incomingByte=='p')){
        codigo = incomingByte;
        endereco = Serial.read();
    } //if incoming
    else{
        codigo = 'n';
    } //else
} //if Serial

if(codigo == 'p'){ //Quando for a posicao for perguntado por um pager
    for(int i = 0; i < nFila; i++){
        if(fila[i] == endereco){
            temp = i;
            Serial.print(endereco);
            Serial.print("q");
            Serial.print(temp);
            Serial.print("&");
        }
    }
    codigo = 'n';
} //if posicao
} //loop

/*****/
/* Adiciona um pager na fila */
/* */
/* Recebe: num = pager a ser adicionado */
/* Devolve: nFila = posicao do pager na fila */
/*****/
char adicionarPager(char num){
    fila[nFila] = num;
    tFila[nFila] = millis();
    nFila++;
    nAdic++;
    return nFila;
} //adicionarPager
/*****/
/* Remove um pager da fila */
/* */
/* Recebe: num = pager a ser procurado */
/* Devolve: Confirmacao booleana */
/*****/
boolean removerPager(char num){
    for(int i = 0; i < nFila; i++){
        if(fila[i] == num){
            for(int j = i; j < nFila; j++){
                fila[j] = fila[j+1];
                tFila[j] = tFila[j+1];
            } //for
        }
    }
}

```

```
        nFila--;  
        nRemov++;  
        fila[nFila] = '\\0';  
        tFila[nFila] = '\\0';  
        return true;  
    }//if fila  
}//for  
return false;  
}//removerPager
```

ANEXO 2 - CÓDIGO FONTE DOS PAGERS

```

/*****
/*  Trabalho Final - PMR2550 (2010)
/*  São Paulo, 1 de dezembro de 2010
/*
/*  Dispositivo para gestao de fila de espera
/*  via sistema de pagers.
/*
/*  Orientador: Prof. Marcos Barretto
/*  Alunos: Igor Araújo de Souza - 5694199
/*          Lucas Steinle Duarte Silva - 5696159
/*
/*  Programa: PAGER
/*  Versao: 4.0 (2010/Nov/29)
/*
/*  Descricao: - Na inicializacao sao ligados os leds
/*              e o motor para teste. Em seguida, o
/*              programa entra em loop aguardando uma
/*              entrada valida via serial no formato:
/*              ID|codigo|endbyte
/*              onde: endbyte = '&'
/*              - Identificado o ID correto e um codigo
/*              valido, executa-se uma das rotinas a
/*              seguir:
/*              a = Adicao na fila pela central
/*                  - naFila = true
/*                  - Pisca LEDs e envia:
/*                    ID|b|&
/*              c = Central chamando este pager
/*                  - naFila = false
/*                  - Pisca LEDs por CICLOS e envia:
/*                    ID|d|&
/*              q = Devolvido o lugar na fila
/*                  - Acende LEDs correspondentes.
/*              - Caso o pager esteja adicionado e se
/*              aperte o botao, solicita-se a posicao
/*              enviando:
/*                  ID|p|&
/*              - O LED2 é piscado por 100ms a cada 6s
/*              para mostrar que o pager esta ligado.
*****/

#define ID 23          //ID deste pager
#define CICLOS 15      //Duracao da chamada (segundos)

#define LED1 11        //pinos dos leds, motor e botao
#define LED2 10
#define LED3 9
#define MOTOR 3
#define BOTAO 13

boolean naFila = false; //true = pager inserido na fila
int incomingByte;        //para gerenciamento de mensagem recebida
char id = ID,            //o envio do ID via serial exige um char
     codigo = 'n',        //comando interpretado via serial
     endereco,            //endereco da mensagem recebida
     posicao;              //posicao recebida via serial
unsigned long tempo;

void setup() {
  Serial.begin(19200);
  pinMode(LED1, OUTPUT);
  pinMode(LED2, OUTPUT);
  pinMode(LED3, OUTPUT);
  pinMode(MOTOR, OUTPUT);
  pinMode(BOTAO, INPUT);

  digitalWrite(LED1, LOW); //teste de leds e motor na inicializacao
  digitalWrite(LED2, LOW);
  digitalWrite(LED3, LOW);
  digitalWrite(MOTOR, HIGH);
  delay(100);
  digitalWrite(LED1, HIGH);
  digitalWrite(LED2, HIGH);

```



```

digitalWrite(LED3, HIGH);
delay(100);
digitalWrite(LED1, LOW);
digitalWrite(LED2, LOW);
digitalWrite(LED3, LOW);
delay(100);
digitalWrite(LED1, HIGH);
digitalWrite(LED2, HIGH);
digitalWrite(LED3, HIGH);
digitalWrite(MOTOR, LOW);
delay(500);
digitalWrite(LED1, LOW);
digitalWrite(LED2, LOW);
digitalWrite(LED3, LOW);
digitalWrite(MOTOR, HIGH);
delay(100);
digitalWrite(LED1, HIGH);
digitalWrite(LED2, HIGH);
digitalWrite(LED3, HIGH);
delay(100);
digitalWrite(LED1, LOW);
digitalWrite(LED2, LOW);
digitalWrite(LED3, LOW);
delay(100);
digitalWrite(LED1, HIGH);
digitalWrite(LED2, HIGH);
digitalWrite(LED3, HIGH);
digitalWrite(MOTOR, LOW);
} //setup

void loop(){
  if(Serial.available() > 0){ //se receber algo via serial
    delay(10); //tempo para armazenar toda mensagem no buffer
    incomingByte = Serial.read();
    if(incomingByte != '&' && incomingByte == ID){ //caso este pager seja chamado
      endereco = incomingByte;
      codigo = Serial.read();
      if(codigo == 'q'){
        posicao = Serial.read();
      } //if codigo
    } //if incoming
    else{
      codigo = 'n';
    } //else
  } //if Serial

  if(codigo == 'a'){ //a = Quando for adicionado na fila
    naFila = true;
    Serial.print("b");
    Serial.print(id);
    Serial.print("&");
    digitalWrite(LED1, LOW);
    digitalWrite(MOTOR, HIGH);
    delay(200);
    digitalWrite(LED2, LOW);
    delay(200);
    digitalWrite(LED3, LOW);
    digitalWrite(MOTOR, LOW);
    delay(200);
    digitalWrite(LED1, HIGH);
    delay(200);
    digitalWrite(LED2, HIGH);
    delay(200);
    digitalWrite(LED3, HIGH);
    codigo = 'n';
    tempo = millis();
  } //if adicionado
  else if(codigo == 'c'){ //c = Quando for chamado
    naFila = false;
    Serial.print("d");
    Serial.print(id);
    Serial.print("&");
    for(int i = 0; i < CICLOS; i++){
      digitalWrite(LED1, LOW);
      digitalWrite(LED2, LOW);
      digitalWrite(LED3, LOW);
      digitalWrite(MOTOR, HIGH);
    }
  }
}

```

```

        delay(500);
        digitalWrite(LED1, HIGH);
        digitalWrite(LED2, HIGH);
        digitalWrite(LED3, HIGH);
        digitalWrite(MOTOR, LOW);
        delay(500);
    }
    codigo = 'n';
    tempo = millis();
} //if chamado
else if(codigo == 'q'){ //q = resposta para lugar na fila
    if(posicao == 0){
        digitalWrite(LED1, LOW);
        delay(2000);
        digitalWrite(LED1, HIGH);
    }
    else if(posicao == 1){
        digitalWrite(LED1, LOW);
        digitalWrite(LED2, LOW);
        delay(2000);
        digitalWrite(LED1, HIGH);
        digitalWrite(LED2, HIGH);
    }
    else if(posicao > 1){
        digitalWrite(LED1, LOW);
        digitalWrite(LED2, LOW);
        digitalWrite(LED3, LOW);
        delay(2000);
        digitalWrite(LED1, HIGH);
        digitalWrite(LED2, HIGH);
        digitalWrite(LED3, HIGH);
    }
    codigo = 'n';
    tempo = millis();
} //if posicao

if(digitalRead(BOTAO) == HIGH && naFila == true){ //mostra posicao na fila no leds
    Serial.print("p");
    Serial.print(id);
    Serial.print("&");
    while(digitalRead(BOTAO) == HIGH);
    delay(200);
} //if botao
else if(millis() - tempo > 6000){
    digitalWrite(LED2, LOW);
    delay(100);
    digitalWrite(LED2, HIGH);
    tempo = millis();
} //if millis
} //loop

```