

Universidade Federal de Campina Grande
Centro de Engenharia Elétrica e Informática
Departamento de Engenharia Elétrica

EVANASKA MARIA BARBOSA NOGUEIRA

**ESTUDO DAS ANTENAS RFID UTILIZANDO UM
HARDWARE PARA PCD (PLATAFORMA DE COLETA DE
DADOS)**

Relatório do projeto de conclusão de curso

CAMPINA GRANDE – PB

JULHO DE 2009

Universidade Federal de Campina Grande
Centro de Engenharia Elétrica e Informática
Departamento de Engenharia Elétrica

RELATÓRIO DO TRABALHO DE CONCLUSÃO DE CURSO

*Relatório do Trabalho de Conclusão de Curso apresentado à Coordenação
do Curso de Graduação em Engenharia Elétrica da Universidade
Federal de Campina Grande, em cumprimento às exigências
para obtenção do Grau de Engenheiro Eletricista.*

Orientador:

Prof. Edmar Candeia Gurjão

Campina Grande, julho de 2009.

Agradeço a Deus, à minha família e aos meus amigos pelo apoio,
ao professor Edmar pelo incentivo e dedicação,
assim como a Joab Leão e Thiago Euzébio
que também estiveram envolvidos e muito me ajudaram
na realização deste trabalho de pesquisa.

Sumário

1. Objetivos.....	05
2. Tecnologia RFID.....	05
2.1. História do RFID.....	05
2.2. Componentes.....	06
2.2.1. Identificador ou transponder (tag)	07
2.2.2. Antena.....	08
2.2.3. Leitor.....	11
2.3. Vantagens e desvantagens do uso do RFID.....	12
2.3.1. Vantagens no uso do RFID.....	12
2.3.2. Desvantagens no uso do RFID.....	13
2.4. O funcionamento da transmissão de dados RFID.....	13
3. Leitor Phidget RFID-1023.....	15
4. Programando em C#.....	18
4.1. Modificações no código.....	18
4.2. Design do programa.....	19
5. Conclusões.....	22
6. Referências.....	23
Anexo – Programa em C#.....	24

1. OBJETIVOS

O objetivo inicial deste trabalho foi estudar de forma detalhada o funcionamento de uma tecnologia de identificação chamada RFID (*Radio Frequency Identification*). De posse de uma base teórica formada, foi feita uma implementação para o uso do RFID. Decidiu-se elaborar um sistema que registrasse o momento da passagem de identificadores eletrônicos (*tags*), salvando esses dados em um arquivo para uma análise posterior. Por exemplo, um caso prático para a utilização desse sistema é o monitoramento de um serviço de segurança, como em um condomínio de casas em que é necessário que alguns vigilantes circulem nas ruas internas, para fiscalizar se o serviço está sendo feito de forma correta, identificadores eletrônicos (*tags*) podem ser adicionados ao crachá do vigilante e cada vez que ele passasse por um leitor fixo em alguns pontos do condomínio, as *tags* seriam identificadas e seria registrado qual vigilante passou pelo local, juntamente com a hora e a data, e esses dados seriam salvos em um arquivo para que o serviço de segurança seja em tempo real ou posteriormente analisado.

2. TECNOLOGIA RFID

Como o próprio nome indica, RFID (*Radio Frequency Identification*) é uma tecnologia de identificação que utiliza a radiofrequência. Com isso, a tecnologia de RFID possibilita que uma informação armazenada em uma *tag* seja lida sem que a mesma esteja dentro do campo visual do leitor, através de objetos tais como madeira, plástico, papel, etc.

2.1. História do RFID

A utilização da radiofrequência para a localização e identificação de objetos não é uma novidade nem um privilégio dos dias atuais. Os primeiros estudos têm sua origem por volta de 1870, com as equações de Maxwell, sobre o estudo da propagação das ondas eletromagnéticas, base para os sistemas de identificação. Em 1904 o alemão C. Hülsmeyer construiu o primeiro radar. Em 1922, Guglielmo Marconi apresentou um estudo sobre a possibilidade da rádio-detecção com base na reflexão das ondas eletromagnéticas [1].

Na década de 30, o físico escocês Robert Alexander Watson-Watt chefiou um projeto secreto que desenvolveu o primeiro sistema ativo identificador (IFF - Identify Friend or Foe), considerado atualmente como o primeiro sistema RFID ativo. O sistema consistia de um transmissor instalado no avião, o qual ao receber os sinais das estações de radar em terra começava a emitir um sinal de volta, possibilitando a identificação da aeronave [2].

Em 1940, a Universidade de Birmingham desenvolveu uma válvula capaz de produzir pulsos de elevada potência (Magnetron), possibilitando a construção de radares de pequeno porte para a instalação em aeronaves e navios. [1].

As pesquisas sobre RFID começaram de fato nos anos 50 e 60, quando vários artigos foram publicados explicando a possibilidade da utilização da tecnologia para a identificação remota de objetos. Em 23 de janeiro de 1973, Mario W. Cardullo afirma ter recebido a primeira patente de uma etiqueta RFID com memória regravável.

A partir da década de 70 surgiram as primeiras aplicações comerciais, dentre as quais podemos destacar o desenvolvimento de um componente de baixa frequência (125 kHz), um transponder encapsulado em vidro que poderia ser injetado sob a pele de vacas. Este sistema ainda é utilizado para a identificação de vacas em todo o mundo. Com o passar do tempo as empresas que forneciam a tecnologia passaram a utilizar o espectro de rádio de alta frequência (13,56 MHz), possibilitando um maior alcance e maior taxa de transferência de dados [2].

2.2. Componentes

Atualmente existe uma centena de empresas que desenvolvem e comercializam equipamentos e produtos para as mais variadas aplicações e ambientes. Basicamente esta tecnologia é composta por três componentes, um identificador ou transponder, conhecido como *Tag*, uma antena e um leitor. Basicamente o sistema RFID é ativado quando um identificador ou *Tag* é inserido no campo de atuação de um leitor, nesse momento o identificador é induzido pelas ondas de rádio geradas pelo leitor e enviadas pela antena, possibilitando ao identificador transmitir os dados armazenados, os quais são captados pela antena e recebidos pelo leitor.

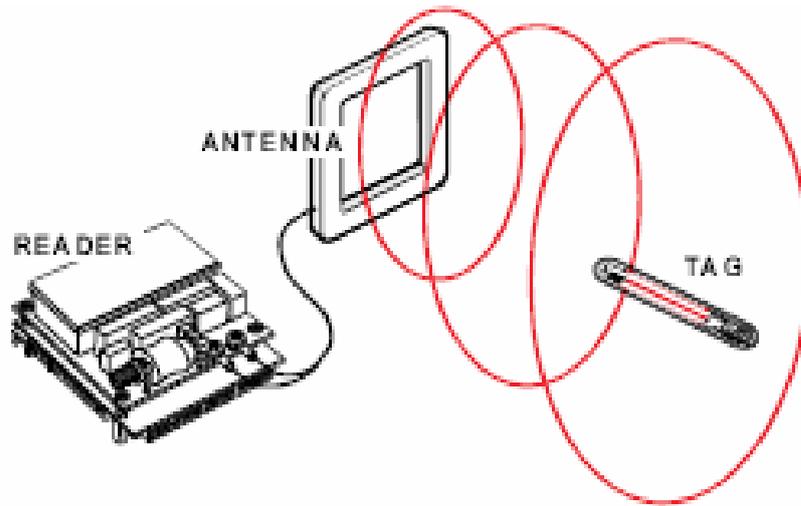


Figura 1 – Simulação de funcionamento de um sistema RFID

Fonte: Santini [10]

2.2.1. Identificador ou transponder (*tag*)

Os identificadores ou *Tags* são dispositivos que possuem funções internas e tem como características básicas o armazenamento de dados e a transmissão. Nem todos os identificadores possuem internamente um microchip ou uma fonte de energia, mas todos são dotados de uma espiral ou algum tipo de antena.

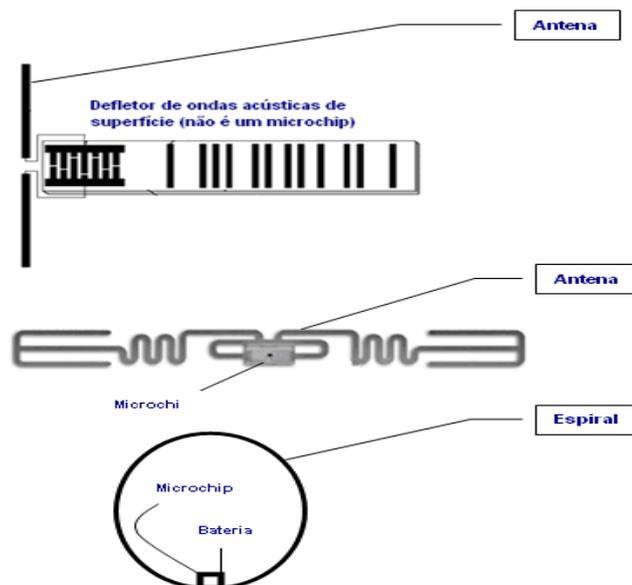


Figura 2 – Identificadores - Tag's RFID

Fonte: Santini [10]

Para cada aplicação existe um tipo de identificador dotado de uma ou mais propriedades, além de suas funções básicas:

- **Anexabilidade:** Todo identificador RFID deve ser anexado de alguma forma a um determinado item ou objeto.
- **Comunicação:** Todo identificador RFID deve ser capaz de transmitir dados por alguma frequência de rádio.
- **Desabilitar:** Alguns identificadores possibilitam que um leitor os comande para pararem de funcionar permanentemente.
- **Gravar uma vez:** Muitos identificadores são fabricados com seus dados pré-gravados ou possibilitam ao usuário a gravação uma única vez, não sendo possível a alteração após a gravação.
- **Regravações:** Alguns identificadores podem ser gravados e regravados várias vezes.
- **Anti-colisão:** Alguns identificadores sabem quando é a sua vez de se comunicar com o leitor.
- **Segurança e criptografia:** Alguns identificadores possuem a capacidade de se comunicar através de criptografia, transmitindo dados e reconhecendo senhas com essa característica.
- **Compatibilidade com padrões:** Um identificador pode ser compatível com um ou mais padrões, seja de hardware ou de software.

Os identificadores possuem diferentes formas e tamanhos para se anexar aos objetos da melhor forma possível. Devido ao fato dos identificadores serem aplicados em diversos tipos de objetos e estarem submetidos a variações ambientais, outra característica dos identificadores é estarem encapsulados em diversos tipos de materiais.

As *tags* podem se apresentar como:

- Botões e discos plásticos, geralmente incluindo um furo central para prendedores, estes identificadores são duráveis e reutilizáveis;
- Cartão de crédito, identificadores RFID no formato de cartão que são chamados de “cartões de crédito sem contato”;
- Identificadores pequenos inseridos em objetos comuns (roupas, relógios e pulseiras), estes identificadores podem ser no formato de chaves e chaveiros.

- Identificadores em cápsulas de vidro: que podem sobreviver até em ambientes líquidos ou corrosivos. [3]

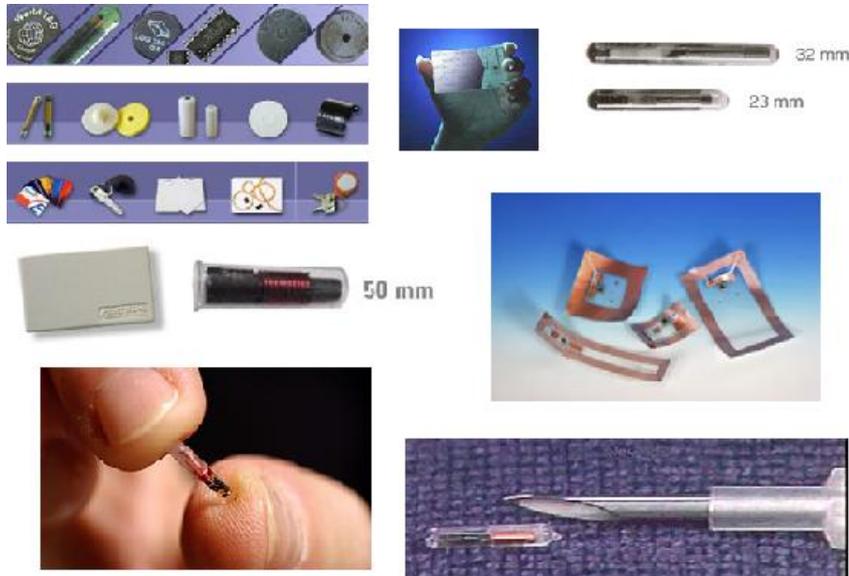


Figura 3 – Tipos de identificadores - Tag's RFID

Fonte: netrand.com

Além das suas características físicas de encapsulamento e formas, os identificadores também podem ser classificados pelas características intrínsecas propriamente ditas, tais como: fonte de energia e por sua faixa de frequência, alcance e aplicações.

Quanto à energia, o identificador pode ser classificado como:

- **Etiqueta ativa:** Possui sua própria fonte de energia, geralmente alimentada por bateria interna, que asseguram o processo de escrita e leitura, com melhor aplicação em artigos grandes e em distâncias maiores.

Podem operar na faixa de 455 MHz e 5,8 GHz, além de possibilitar a comunicação com distâncias de 20 a 100 metros.

Sua vida útil é estimada em dez anos. Esta categoria de etiqueta tem um custo mais alto do que os da etiqueta passiva. [4]

• **Etiqueta Passiva:** não possui fonte própria de energia. Com custo e tamanho consideravelmente menor que a categoria de etiqueta ativa, é a base do crescimento nas implantações de RFID.

Pelo fato de possuírem apenas 2Kbits de memória, dificulta as identificações complexas. Porém, muitos esforços estão sendo destinados para aumentar a capacidade de armazenamento desta etiqueta. A sua durabilidade é indeterminada. [4]

Pode-se também realizar uma diferenciação das *tags* quanto à faixa de frequência que pode ser resumida pela Tabela 1 a seguir.

Faixa de frequência	Padrões	Alcance	Aplicações
< 135 KHz	ISO 1784/11785 ISO/IEC 18000-2 ISO 14223-1	< 0,5 m	Controle de acesso, rebanho rastreado.
13,56 MHz	ISO/IEC 18000-3 EPC class-1 ISO 15693 ISO 14443	~ 1 m	Cartão inteligente, controle de acesso, ID de pagamento, controle de bagagem, livrarias, biométricos.
860 - 960 MHz	ISO/IEC 18000-6 EPC class- 0 e 1	~ 4 - 5 m	Pagamento eletrônico, manejo de bagagem.
2,45GHz	ISO/IEC 18000-4	~ 10 m	Pagamento eletrônico, monitoramento de ambientes.

Tabela 1 – Tabela de frequência e aplicações - RFID

Observando a Tabela 1, percebemos a existência de dois padrões, o ISO e o EPC. Esses padrões surgiram da necessidade de se estabelecer normas e regras que possibilitassem a interoperabilidade entre os sistemas, possibilitando a comunicação e a troca de dados entre leitores e identificadores de fabricantes diferentes. Apesar do esforço das duas entidades, os padrões ISO (*International Organization for Standardization*) e EPC Global não são completamente compatíveis entre si.

Como toda tecnologia nova ou em expansão, os problemas de padronização são freqüentes. No início os maiores fabricantes desenvolviam sistemas próprios, o que acarretava uma diversidade de protocolos que gerava uma série de dificuldades para a comunicação entre os sistemas. [3]

2.2.2. Antena

São as responsáveis por enviar ondas eletromagnéticas que transportam a informação numa conexão sem fio entre os leitores e os identificadores, geralmente são alimentadas pelos próprios leitores, mas podem possuir alimentação própria. Para cada utilização e ambiente existe um modelo específico de antena, sendo que o que vai determinar o tipo a ser utilizado é o seu diagrama de radiação, que é a representação gráfica do ganho de uma antena num determinado plano, um exemplo desse tipo de diagrama pode ser observado na Figura 4, em que se tem representado um diagrama de uma antena com uma direção de radiação predominante.

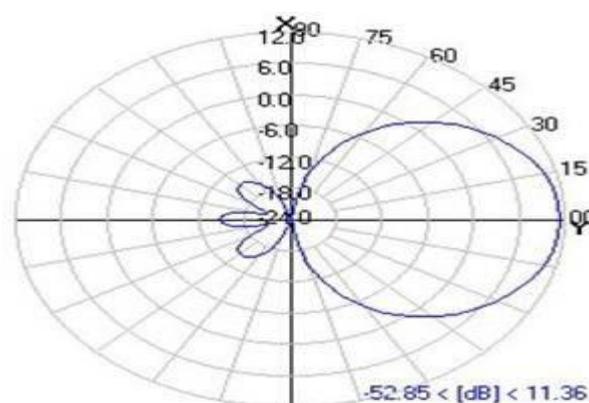


Figura 4 – Diagrama de radiação de uma Antena Diretiva.

Fonte: teleco.com.br

As antenas de desktop e de parede são as mais comuns, e em muitos casos substituem as antenas de portal, as exemplificações dessas antenas podem ser observadas na figura 5.

- A - Antena de desktop ou parede;
- B - Antena de HF;
- C - Antena em portal.

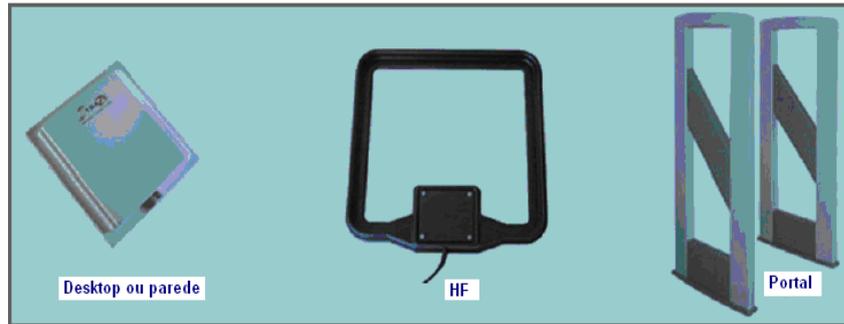


Figura 5 – Tipos de antenas – RFID

Fonte: Estado da Arte em RFID [7]

Os leitores manuais e alguns modelos de leitores fixos e móveis possuem suas próprias antenas incorporadas [5].

2.2.3. Leitor

O leitor é o hardware responsável por fazer a ligação entre o identificador e o gerenciador de eventos ou aplicação, por meio de uma interface de rede. Sendo composto por componentes físicos e lógicos. Dentre os componentes físicos de um leitor, está a antena, que pode ou não estar acoplada. O controlador, que é o dispositivo responsável pela verificação e avaliação dos dados recebidos, determinando quando e como os mesmos deverão ser enviados pela interface de rede. E a interface de rede, que são os meios pelos quais os leitores enviam os dados coletados, podendo ser serial, USB, Ethernet, Wireless, Bluetooth ou Zigbee.

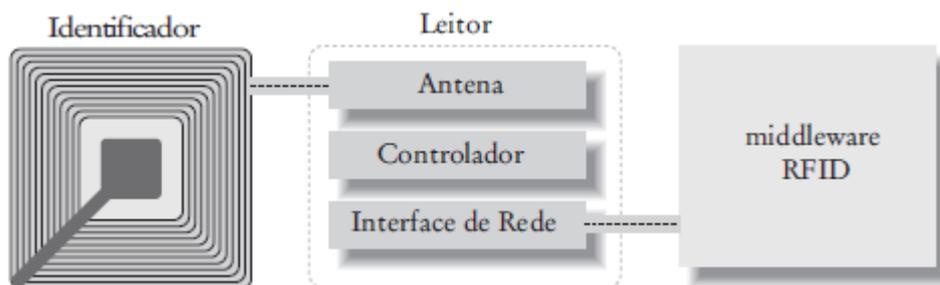


Figura 6 – Componentes físicos de um leitor

Fonte: Gloover [3]

Os componentes lógicos do leitor estão localizados no interior do controlador, eles são na verdade quatro subsistemas separados, dotados de diferentes atribuições, são eles:

- API do leitor → Responsável pela troca de informações entre as aplicações com as quais interage, referente às condições do leitor, dos identificadores, aos níveis de energia, tempo corrente e parâmetros de configuração.
- Comunicações → Responsável pelo gerenciamento dos protocolos de comunicação entre o leitor e o *middleware* ou aplicação.
- Gerenciador de eventos → Responsável pelo monitoramento dos identificadores, filtrando e enviando os eventos cuja sua configuração ou programação julgar relevante.
- Subsistemas de antenas → Responsável pela execução da configuração lógica de como os leitores controlam e interrogam os identificadores, além do controle físico das antenas, dos protocolos de RF e da implementação dos protocolos de comunicação entre os identificadores.

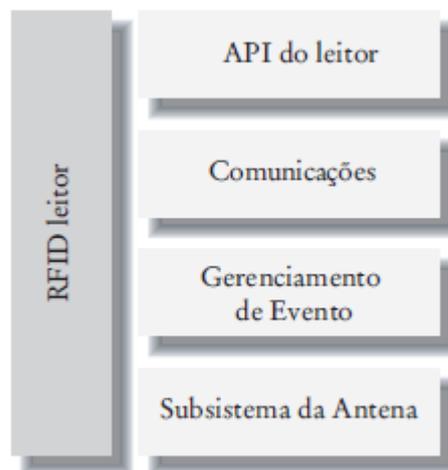


Figura 7 – Componentes lógicos de um leitor

Fonte: Glover [3]

Da mesma forma que os identificadores, existem hoje no mercado uma quantidade enorme de modelos e formatos, para as mais variadas aplicações, de diversos fabricantes. Podendo ser encontrados em tamanhos que variam desde meia polegada até o tamanho de um computador de mesa, possibilitando a sua instalação ou adaptação para as configurações do tipo portais, túneis, leitores em empilhadeiras ou prateleiras inteligentes. [3]

2.3. Vantagens e desvantagens do uso do RFID

O RFID apresenta vantagens e desvantagens no seu desenvolvimento, funcionamento e custo. Elas serão descritas com mais detalhes a seguir.

2.3.1. Vantagens no uso do RFID

Além da eliminação dos erros de escrita e leitura de dados, podem ser enumeradas outras vantagens com o uso da tecnologia de RFID, em relação às outras tecnologias de identificação e coleção de dados existentes. [6]

São estas algumas das vantagens:

- Controle com precisão, de todos os bens de um supermercado ou empresa com armazenamento de produtos, desde que possuam as etiquetas inteligentes;
- Assegurar operação segura em ambiente severo (lugares úmidos, molhados, sujos, corrosivos, altas temperaturas, baixas temperaturas, vibração, choques);
- Operação sem contato e sem necessidade de campo visual, e etiquetas com grande variedade de formatos e tamanhos;
- Aplicação de etiquetas ativas com capacidade de armazenamento para leitura e envio dos dados;
- Reconhecimento da etiqueta inteligente sem a necessidade de proximidade da leitora;
- Durabilidade das etiquetas com a possibilidade de reutilização;
- Redução dos estoques como consequência do melhor gerenciamento e controle da logística;
- Contagem instantânea dos estoques;
- Precisão nas informações de armazenamento;
- Velocidade na expedição de produtos;
- Facilidade para localização dos itens no estoque;
- Eficiência no reabastecimento dos estoques;
- Controle eficiente na validade dos produtos;
- Prevenção contra roubo e falsificação de produtos.

2.3.2. Desvantagens no uso do RFID

O custo elevado da tecnologia RFID e dos equipamentos, quando comparados com os sistemas de código de barras. Este é um dos principais obstáculos para o aumento de sua aplicação comercial. Atualmente, uma etiqueta inteligente custa nos EUA cerca de US\$ 0.25, para compra em lotes de um milhão de chips. No Brasil, segundo a ABAC (Associação Brasileira de Automação Comercial), esse custo é de aproximadamente R\$ 0,50. Comparando com a impressão do código de barras, a etiqueta do RFID custa até 10 vezes mais; [9]

- O uso em materiais metálicos e condutivos pode afetar o alcance de transmissão das antenas. Como a operação é baseada em campos magnéticos, o metal pode interferir negativamente no desempenho. Entretanto, encapsulamentos especiais podem contornar esse problema, fazendo com que automóveis, vagões de trens e contêineres possam ser identificados resguardados as limitações com relação às distâncias de leitura; Nesse caso, o alcance das antenas depende da tecnologia e frequência utilizada, podendo variar de poucos centímetros a alguns metros (cerca de 30 metros), dependendo da existência ou não de barreiras;

- Falta a padronização das frequências, que ainda está em andamento;
- Invasão de privacidade, quando as etiquetas são utilizadas como propósito de rastrear os produtos da origem até o consumidor final; [9]

2.4. O funcionamento da transmissão de dados – RFID

O funcionamento da transmissão de dados ocorre a partir da entrada do objeto ou produto contendo a etiqueta inteligente, na área coberta pelo leitor de etiqueta, que emite suas ondas constantemente. [7]

A partir da identificação da etiqueta, o leitor envia um sinal eletromagnético, que é recebido pela antena da etiqueta. No retorno da comunicação, a etiqueta transmite um sinal modulado ao leitor com as informações armazenadas.

A comunicação entre o leitor e a etiqueta passiva, é realizada utilizando um dos dois métodos existentes para modular o sinal de identificação, que são: baixa frequência, com menos de 100 Mhz, ou de alta frequência que é superior a 100 Mhz.

De forma mais detalhada, conforme descreve Roussos [8], o funcionamento da identificação por rádio frequência se dá a partir de um campo eletromagnético, formado entre antenas e algo para responder ao estímulo criado pelo campo magnético, conforme demonstrado na figura 8.

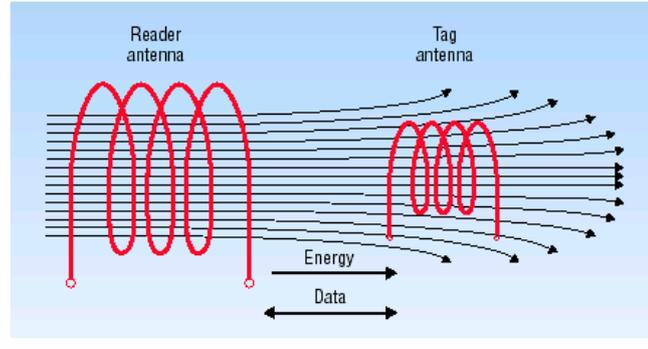


Figura 8 - Operação do RFID: A antena provê a energia para a etiqueta transmitir o número de identificação armazenado.
Fonte: Roussos [8]

O processo de leitura inicia-se quando a etiqueta recebe um sinal eletromagnético de um leitor. A etiqueta armazena a energia do sinal em um capacitor, por meio de um processo chamado “junção indutiva”. Em seguida, transmite um sinal de retorno ao leitor com as informações armazenadas na etiqueta. Como pode ser observado na figura 9.

A comunicação entre o leitor e a etiqueta passiva, pode ser feita usando duas faixas de frequências para modular o sinal de ID. São elas: Baixa frequência (de 30 a 500 kHz), e alta frequência (850 a 950 MHz e de 2,4 a 2,5 GHz).

A grande diferença entre as duas frequências, é que em baixas frequências, a etiqueta passiva é capaz de transmitir seus dados apenas para pequenas distâncias.

Em altas frequências, a distância para a leitura entre a etiqueta ativa e o leitor podem ser aumentadas, porém o aumento é limitado por legislação e imposição dos governos e dos órgãos reguladores. [4]

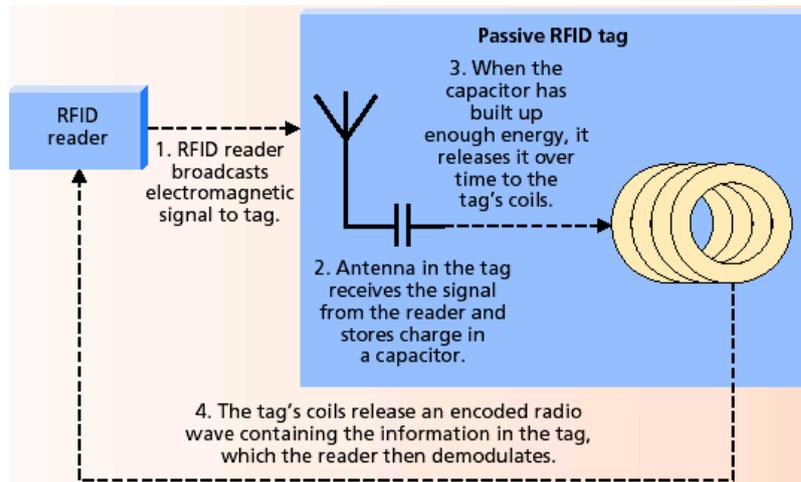


Figura 9- Etiqueta passiva RFID: transferência em baixa-frequência

Fonte: Weinstein [7]

3. LEITOR PHIDGET RFID - 1023

O produto utilizado durante esse projeto para a análise do funcionamento de um leitor RFID foi o equipamento da Phidget de código 1023, esse equipamento está representado na figura 10.



Figura 10 – 1023 Phidget RFID

Fonte: Manual da Phidget 1023

Esse leitor é ligado diretamente a uma porta USB e utiliza o protocolo EM4102 para ler as *tags*, sendo que essa leitura é a cerca de 6cm de distância da *tag* passiva. Após a comunicação, ele retorna o numero identificador do transponder. Em sua placa podemos encontrar ainda um LED e saídas de 5V.

Quando se deseja estabelecer uma comunicação com uma *tag* RFID é necessário escolher um protocolo. Nesse protocolo encontram-se regras e configurações a serem utilizadas durante a transmissão sem fio entre um leitor e uma *tag*. Como foi

anteriormente citado, para o caso do leitor Phidget e as *tags* adquiridas juntamente com o leitor, o protocolo utilizado é o EM4102.

Como foi explicado na primeira parte desse trabalho, podemos ter as *tags* ativas e passivas. Mas para esse projeto foram utilizadas apenas transponders passivos para minimizar os custos. Essas *tags* por não possuir uma fonte própria de energia, para seu funcionamento e alimentação através do leitor é necessário que ela esteja a uma distância pequena, cerca de 6cm. Essa distância irá ser variável para cada caso de *tag*, pois em alguns casos é necessário uma fixação da *tag* em um material que interfere um pouco na transmissão, sendo necessário assim uma maior proximidade do leitor.

Os três tipos de *tags* testados e utilizados durante esse trabalho foram:

- *Tags* de disco de 30mm, que geralmente são utilizadas para serem coladas a objetos.
- *Tags* em cartões
- *Tags* em chaveiros que são de fácil portabilidade.

Então o kit utilizado é composto basicamente pelo leitor PhidgetRFID 1023, um cabo usb, e as tags compatíveis. Esse material pode ser visualizado na figura 11 a seguir.



Figura 11 – 1023 Phidget RFID
Fonte: Manual da Phidget 1023

Para o funcionamento desse kit RFID, pode-se utilizar como sistema operacional o Windows 2000/XP/Vista, Windows CE, Linux ou Mac OS X.

Quanto à linguagem de programação pode-se utilizar o VB6, VB.NET, C#.NET, C++, Flash 9, Flex, Java, LabVIEW, Python, Max/MSP, e Cocoa. Mas para o nosso trabalho de conclusão de curso, foi escolhido o C# por ser uma linguagem baseada em C/C++ estudadas durante o curso de engenharia elétrica, pois ao tentarmos implementar usando diretamente a linguagem C++ não obtivemos êxito.

Programas simples para implementação nessas linguagens podem ser encontrados no site do fabricante www.phidget.com.

Algumas informações extras sobre o leitor phidget RFID 1023 podem ser encontradas na tabela 2 abaixo como a potência da antena e as distâncias para cada tipo de tag.

Potência de saída da antena (max, far diel)	< 10 μ W
Frequência da antena ressonante	125kHz – 140kHz
Protocolo de comunicação	EM4102
Taxa de atualização da leitura	30 atualizações/segundo
Tensão de alimentação externa +5V	5VDC
Corrente limite de alimentação externa +5V	400mA
Tensão de alimentação externa do LED	5VDC
Corrente limite de alimentação externa LED	16mA
Resistência de saída externa LED	250 Ohms
Tipo de fio recomendado	16-26 AWG
	5 – 6mm (0.196” – 0.236”)
USB – Especificação da potência da corrente	500mA Max
Consumo de corrente do dispositivo passivo	16Ma
Consumo de corrente do dispositivo ativo	100mA Max
Distância típica de leitura – Tag em cartão	11cm (5”)
Distância típica de leitura – Tag em disco	6cm (3”)
Distância típica de leitura – Tag em chaveiro	7cm (3.5”)

Tabela 2 – Configurações do Phidget RFID 1023

Esse produto passou por uma série de atualizações desde 2002 quando foi lançado no mercado. Alguns desses melhoramentos podem ser observados na tabela 3.

Data	Revisão do Produto	Comentário
Junho 2002	Versão do dispositivo 100	Liberação do Produto
Abril 2004	Versão do dispositivo 200	Acréscimo do LED e de 2 saídas digitais. Acréscimo da habilidade de habilitar e desabilitar a antena.
Janeiro 2005	Versão do dispositivo 201	Circuito RF adaptado para melhorar a confiabilidade.
Janeiro 2006	Versão do dispositivo 202	Melhoramento para a versão flash do microprocessador
Junho 2006	Versão do dispositivo 204	Baixa tensão de reset sendo 4,7V

Abril 2007	PCB Rev 0, Versão do dispositivo 205	Mudança de erros no protocolo de comunicação com a tag.
Julho 2007	PCB Rev 1, Versão do dispositivo 206	Abertura no centro da antena, conexões na borda do PCB

Tabela 3 – História do Leitor PhidgetRFID 1023

4. PROGRAMANDO EM C#

C# (CSharp) é uma linguagem de programação orientada a objetos desenvolvida pela Microsoft como parte da plataforma .Net. A sua sintaxe orientada a objeto foi baseada no C++, mas inclui muitas influências de outras linguagens de programação.

4.1. Modificações no código

Para tornar o programa mais completo e com algumas funções desejadas, foram feitas algumas alterações no código fornecido pela phidgets.com para a programação em C#. Essas alterações compreendem os seguintes aspectos:

- Registrar a hora e a data em que a *tag* passou pelo leitor Phidget RFID
 - Retirar informações irrelevantes para obter um programa mais compacto e funcional.
 - Mostrar na tela do programa o histórico das *tags* que foram identificados pelo leitor, juntamente com o horário e data.
 - Salvar os dados obtidos com o programa em um arquivo no computador no formato .txt
 - Fazer a equivalência dos nomes dos funcionários para a sua *tag* correspondente.
- Essas modificações foram feitas no código do programa e podem ser analisadas a seguir.

Primeiramente para se registrar a hora e a data em que o leitor RFID identifica a *tag*, utilizou-se o seguinte comando dentro da função que registra a *tag*:

```
DateTime.Now.ToString()
```

Após essa primeira fase, para retirar informações não necessárias ao nosso aplicativo, foram comentadas no código as partes equivalentes às saídas output, led, considerações da antena, pois essa estará sempre acoplada ao leitor, então não será

necessário modificar seus parâmetros. E na parte de design foram retiradas as partes correspondentes a essas funções.

Para mostrar o histórico das tags identificadas com os horários correspondentes para cada tag, houve a necessidade de na parte de design acrescentar um “Rich TextBox” para que vários dados possam ser mostrados, e podendo criar barra de rolagem caso tenha um espaço insuficiente para mostrar as tags identificadas. Denominamos essa caixa de texto com o nome “completo” e com a seguinte instrução

```
completo.Text = texto;
```

definimos que na caixa de texto estará presente o conteúdo de “texto”. Sendo a “texto” definido como:

```
texto = texto + funcionario + " " + " " +
DateTime.Now.ToString() + "\n";
```

Assim, teremos na caixa de texto correspondente ao histórico o registro dos funcionários que passaram anteriormente, e os novos dados coletados, ou seja, o nome do funcionário, a hora e a data que a tag foi identificada.

Para fazer a equivalência do valor associado a cada tag e o nome do funcionário utilizou-se um swich da seguinte forma:

```
switch (e.Tag)// atribui ao numero do tag o nome do
funcionario
{
    case "1f00be131b":
        funcionario="Antonio";
        break;
    case "046011a670":
        funcionario="Rafael";
        break;
    case "1f008c1757":
        funcionario="Thiago";
        break;
    case "2100454bc7":
        funcionario="Markus";
        break;
    default:
        funcionario = "Não cadastrado";
        break;
}
```

Pode ser colocado todo o quadro de funcionários desejado, mas colocamos nomes quaisquer para exemplificar algumas tags e os outros códigos serão citados como “Não cadastrado”.

Por fim, para não haver perda de dados quando o programa fosse fechado, os dados foram salvos em um arquivo “*.txt” primeiramente deve-se criar um arquivo teste.txt e depois salva o conteúdo presente em “texto”, que é exatamente o conteúdo

presente no histórico em “teste.txt” no final do programa, através dos seguintes comandos:

```
private                                     string
arquivo="D:\\UNIVERSIDADE\\UFCG\\tcc\\teste.txt";
System.IO.File.WriteAllText(@arquivo, texto);
```

Com esses comandos foi possível realizar as modificações e o programa começou a operar de forma desejada e satisfatória.

4.2. Design do programa

No *Microsoft Visual Studio* (programa gratuito para compilação de códigos na linguagem C#) realizamos algumas modificações na estrutura do nosso programa, para que esse ficasse mais compacto e funcional.

Primeiramente utilizamos o programa adquirido no site www.phidgets.com para o leitor RFID phidget 1023. É importante notar uma atualização na biblioteca .lib, e que antes de compilar o programa é necessário que o leitor tenha instalado e conectado o leitor ao computador via porta USB.

O programa apresentava uma aparência semelhante a encontrada na Figura 12.

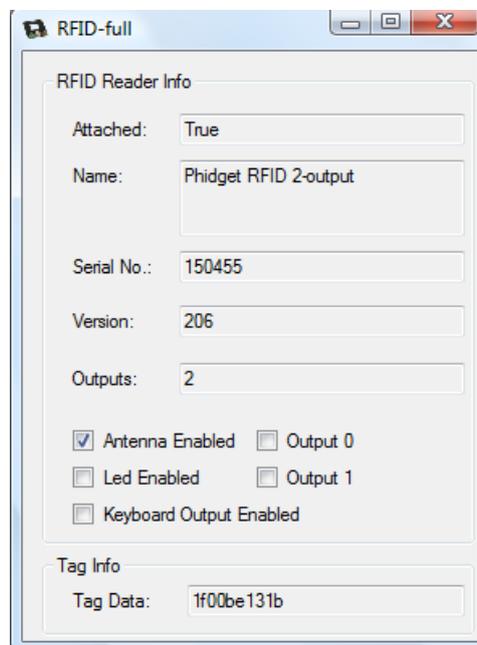


Figura12 – Programa RFID inicial

Mas nesse programa obtínhamos apenas o número identificador da *tag* e somente no instante em que ela era conhecida, sendo assim, o programa não apresentava os dados necessários para atingir os objetivos desse projeto.

Então efetuamos uma série de modificações tanto no código como foi dito anteriormente, como nos dados a serem exibidos. Encontramos como resultado a versão apresentada na figura 13.

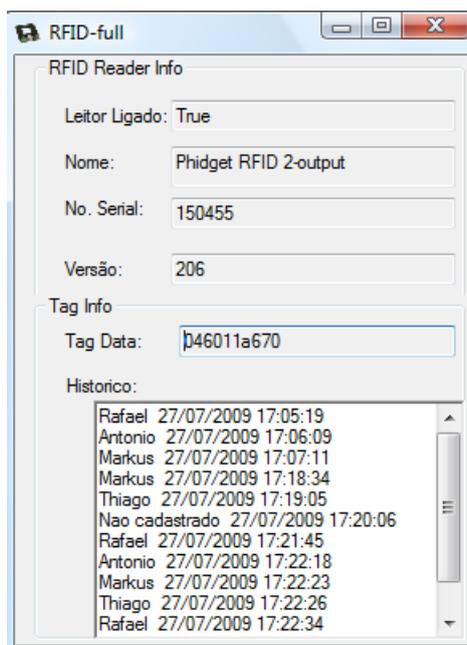


Figura 13 – Programa RFID final

Como pode ser observado na figura 13, primeiramente o programa informa se o leitor RFID está ligado, ou seja, conectado a porta a USB e devidamente reconhecido. Mostra ainda qual leitor está conectado ao computador, assim como o seu número serial e a sua versão.

Na segunda parte, em que se tem as informações da *tag*, podemos observar primeiramente o número da *tag* que está sendo identificada naquele exato momento, e depois mostra o histórico de todos os funcionários, que passaram a partir de quando o programa foi aberto, assim como a data e o horário. Caso o número de funcionários identificados seja grande, uma barra de rolagem será criada.

Após o fechamento do programa, um arquivo teste.txt será criado, para o caso exemplificado anteriormente, o conteúdo está representado na figura 14 a seguir.

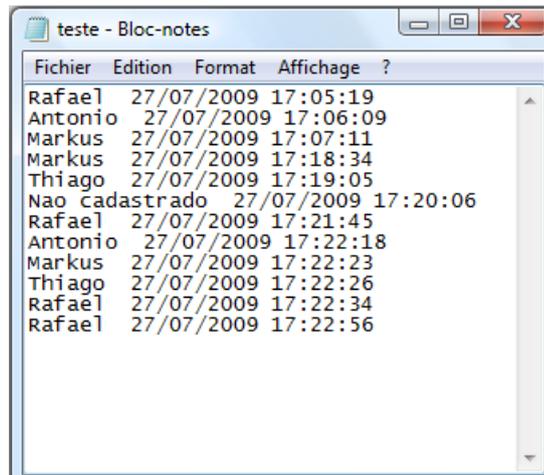


Figura 14 – Arquivo teste.txt

No arquivo “teste.txt” temos uma representação dos dados da mesma forma que foi mostrado durante a execução do programa na parte referente ao histórico.

5. CONCLUSÕES

Nesse trabalho, foi realizado um aprofundamento dos conceitos anteriormente adquiridos sobre o sistema de identificação RFID colocando-os em prática.

Com esse trabalho foi elaborado um sistema de monitoramento de um conjunto de pessoas que trabalham no setor de segurança por meio de tags associadas a cada pessoa e com a ajuda um leitor de RFID em um local fixo, consegue-se saber a que horas e data cada vigilante passou pelo leitor.

Foi adquirido ainda um conhecimento em uma linguagem que eu ainda não tinha estudado anteriormente que é o Csharp (C#), resultando em ampliações nos conhecimentos de programação orientada a objeto.

Então, durante o projeto de conclusão de curso, foi obtido um conjunto de conceitos antigos que foram aprimorados e novos estudos reunidos em um só trabalho, reunindo conceito e prática. Resultando em um projeto funcional e interessante, que pode ser aprimorado e posteriormente colocado em prática.

6. REFERÊNCIAS

- [1] MARINHA do Brasil. **Navegação Radar: Navegação costeira, estimada e em águas restritas**. 84p. Disponível em <http://www.mar.mil.br/dhn/bhmn/download/cap14>. Acessado em: 21 ago.2008.
- [2] RFID Journal - 1338/1/129. **WHAT is RFID?**. Disponível em <http://www.rfidjournal.com/article/articleview/1338/1/129/> Acessado em: 21 ago.2008.
- [3] GLOVER, Bill; BHATT, Himanshu. **Fundamentos de RFID: Teoria em Prática**. Rio de Janeiro: Alta Books, 2007. 227p.
- [4] ROSA, Luiz Antonio. **Aplicação do RFID na cadeira logística**. Universidade de São Paulo – Escola Politécnica MBA Tecnologia de Informação.2007.
- [5] SYBASE Portugal - RSC (RFID Solutions Center). **Estado da Arte em RFID**. 1.0, 2006. Disponível em <http://www.portalfid.net/documents/EstadoDaArte.pdf>. Acesso em: 26 nov.2008.
- [6] ACURA, Technologies Ltd. - Casos e Aplicações da Tecnologia RFID. Disponível em www.acura.com.br/aplic_log_20.php (acesso em 13/11/2006).
- [7] WEINSTEIN, Ron, **RFID a technical Overview and its application to the Enterprise**, IEEE Computer Society, May / Jun-2005.
- [8] ROUSSOS, George, **Enabling RFID in retail**, IEEE Computer Society and Birkbeck College, University of London, March 2006.
- [9] http://www.fasul.edu.br/pasta_professor/arquivos//26/1073_05_-_tecnologia_de_rfid.ppt Acesso em 28 jul.2009.
- [10] SANTINI, Arthur Gambin. **RFID (Radio Frequency Identification) – Conceitos Aplicabilidade e Impactos**. Ed. Ciência Moderna. 96p. 1 edição. 2008

ANEXO 1 - Programa Principal em C#

```

using System;
using System.IO;
using System.Runtime.Serialization;
using System.Runtime.Serialization.Formatters.Binary;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using Phidgets; //Necessario para a classe RFID e a classe PhidgetException
using Phidgets.Events; //Necessario para phidget event handling classes

namespace RFID_full
{
    public partial class Form1 : Form
    {
        private RFID rfid; //Declara um objeto RFID
        private ErrorEventBox errorBox;
        private string texto="";
        private string funcionario;
        private string arquivo="D:\\UNIVERSIDADE\\UFCG\\tcc\\teste.txt";
        private FileStream output;
        public Form1()
        {
            InitializeComponent();
            errorBox = new ErrorEventBox();
        }

        //inicializa o leitor Phidgets RFID
        private void Form1_Load(object sender, EventArgs e)
        {
            rfid = new RFID();

            rfid.Attach += new AttachEventHandler(rfid_Attach);
            rfid.Detach += new DetachEventHandler(rfid_Detach);
            rfid.Error += new Phidgets.Events.ErrorEventHandler(rfid_Error);

            rfid.Tag += new TagEventHandler(rfid_Tag);
            rfid.TagLost += new TagEventHandler(rfid_TagLost);

            openCmdLine(rfid);

            void rfid_Attach(object sender, AttachEventArgs e)
            {
                RFID attached = (RFID)sender;
                attachedTxt.Text = attached.Attached.ToString();
                nameTxt.Text = attached.Name;
                serialTxt.Text = attached.SerialNumber.ToString();
                versionTxt.Text = attached.Version.ToString();

                if (rfid.outputs.Count > 0)
                {
                    rfid.Antenna = true;
                }
            }

            //detach event handler...clear all the fields, display the attached status, and disable the checkboxes.
            void rfid_Detach(object sender, DetachEventArgs e)
            {
                RFID detached = (RFID)sender;
                attachedTxt.Text = detached.Attached.ToString();
            }
        }
    }
}

```

```

nameTxt.Text = "";
serialTxt.Text = "";
versionTxt.Text = "";

if (rfid.outputs.Count > 0)
{
}

void rfid_Error(object sender, Phidgets.Events.ErrorEventArgs e)
{
    Phidget phid = (Phidget)sender;
    DialogResult result;
    switch (e.Type)
    {
        case PhidgetException.ErrorType.PHIDGET_ERREVENT_BADPASSWORD:
            phid.close();
            TextInputBox dialog = new TextInputBox("Error Event",
                "Authentication error: This server requires a password.", "Please enter the password, or cancel.");
            result = dialog.ShowDialog();
            if (result == DialogResult.OK)
                openCmdLine(phid, dialog.password);
            else
                Environment.Exit(0);
            break;
        case PhidgetException.ErrorType.PHIDGET_ERREVENT_NETWORK:
            result = MessageBox.Show("Network Error: " + e.Description +
                Environment.NewLine + "Keep trying to connect?", "Error Event", MessageBoxButtons.YesNo);
            if (result == DialogResult.No)
            {
                phid.close();
                Environment.Exit(0);
            }
            break;
        default:
            if (!errorBox.Visible)
                errorBox.Show();
            break;
    }
    errorBox.addMessage(DateTime.Now.ToLongDateString() + " " + DateTime.Now.ToLongTimeString() + ":
" + e.Description);
}

//Tag event handler...we'll display the tag code in the field on the GUI
void rfid_Tag(object sender, TagEventArgs e)
{
    switch (e.Tag) // atribui ao numero do tag o nome do funcionario
    {
        case "1f00be131b":
            funcionario="Antonio";
            break;
        case "046011a670":
            funcionario="Rafael";
            break;
        case "1f008c1757":
            funcionario="Thiago";
            break;
        case "2100454bc7":
            funcionario="Lucas";
            break;
        default:
            funcionario = "Nao cadastrado";
            break;
    }
    texto = texto + funcionario + " " + DateTime.Now.ToString() + "\n";
    tagTxt.Text = e.Tag;
}

```

```

completo.Text = texto;
//This sends the RFID tag and an enter to the active application
if (keyboardCheckBox.Checked == true)
{
    SendKeys.Send(e.Tag);
    SendKeys.Send("{ENTER}");
}

}

//Quando a tag for retirada ira limpar o campo do numero da tag
void rfid_TagLost(object sender, TagEventArgs e)
{
    tagTxt.Text = "";
}

//When the application is being terminated, close the Phidget.
private void Form1_FormClosing(object sender, FormClosingEventArgs e)
{
    rfid.Attach -= new AttachEventHandler(rfid_Attach);
    rfid.Detach -= new DetachEventHandler(rfid_Detach);
    rfid.Tag -= new TagEventHandler(rfid_Tag);
    rfid.TagLost -= new TagEventHandler(rfid_TagLost);

    //run any events in the message queue - otherwise close will hang if there are any outstanding events
    Application.DoEvents();

    rfid.close();
    System.IO.File.WriteAllText(@arquivo, texto);
}

//Parses command line arguments and calls the appropriate open
#region Command line open functions
private void openCmdLine(Phidget p)
{
    openCmdLine(p, null);
}
private void openCmdLine(Phidget p, String pass)
{
    int serial = -1;
    int port = 5001;
    String host = null;
    bool remote = false, remoteIP = false;
    string[] args = Environment.GetCommandLineArgs();
    String appName = args[0];

    try
    { //Parse the flags
        for (int i = 1; i < args.Length; i++)
        {
            if (args[i].StartsWith("-"))
                switch (args[i].Remove(0, 1).ToLower())
                {
                    case "n":
                        serial = int.Parse(args[++i]);
                        break;
                    case "r":
                        remote = true;
                        break;
                    case "s":
                        remote = true;
                        host = args[++i];
                        break;
                }
        }
    }
}

```

```

        case "p":
            pass = args[++i];
            break;
        case "i":
            remoteIP = true;
            host = args[++i];
            if (host.Contains(":"))
            {
                host = host.Split(':')[0];
                port = int.Parse(host.Split(':')[1]);
            }
            break;
        default:
            goto usage;
    }
else
    goto usage;
}

if (remoteIP)
    p.open(serial, host, port, pass);
else if (remote)
    p.open(serial, host, pass);
else
    p.open(serial);
return; //success
}
catch {}
usage:
    StringBuilder sb = new StringBuilder();
    sb.AppendLine("Invalid Command line arguments." + Environment.NewLine);
    sb.AppendLine("Usage: " + appName + " [Flags...]");
    sb.AppendLine("Flags:\t-n serialNumber\tSerial Number, omit for any serial");
    sb.AppendLine("\t-r\t\tOpen remotely");
    sb.AppendLine("\t-s serverID\tServer ID, omit for any server");
    sb.AppendLine("\t-i ipAddress:port\tIp Address and Port. Port is optional, defaults to 5001");
    sb.AppendLine("\t-p password\tPassword, omit for no password" + Environment.NewLine);
    sb.AppendLine("Examples: ");
    sb.AppendLine(appName + " -n 50098");
    sb.AppendLine(appName + " -r");
    sb.AppendLine(appName + " -s myphidgetserver");
    sb.AppendLine(appName + " -n 45670 -i 127.0.0.1:5001 -p paswr");
    MessageBox.Show(sb.ToString(), "Argument Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

    Application.Exit();
}
#endregion
}
}

```