UNIVERSIDADE FEDERAL DE CAMPINA GRANDE
CENTRO DE ENGENHARIA ELÉTRICA E INFORMÁTICA
UNIDADE ACADÊMICA DE ENGENHARIA ELÉTRICA

RELATÓRIO DE ESTÁGIO

CAPTURA DE SINAL RFID

Aluno
Tiago Carvalho Leite

Orientador
Prof. Dr. Edmar Candeia Gurjão

Campina Grande, Abril de 2010

Sumário

Resumo

O estágio foi realizado na Universidade Federal de Campina Grande (UFCG) no Laboratório de Comunicações (LABCOM), orientado pelo Prof. Dr. Edmar Candeia Gurjão. Analisando os recursos do laboratório e a minha experiência acadêmica e profissional, foi decidido fazer algo relacionado à tecnologia RFID, visto que eu já conhecia algo sobre a tecnologia de trabalhos acadêmicos anteriores em outro laboratório da mesma universidade.

No laboratório havia um kit RFID, o trabalho foi feito em cima desse kit e consistia de colocá-lo para funcionar e posteriormente realizar a captura do sinal de comunicação entre a etiqueta RFID e o leitor.

Neste relatório pretende-se mostrar o que foi feito para chegar a tal feito, tentando detalhar o processo para que ele possa ser repetido por outra pessoa que venha a realizar trabalhos com o mesmo kit.

Inicialmente faremos uma introdução teórica falando a respeito da tecnologia RFID, discorrendo sobre como ela funciona, falando de cada elemento de um sistema RFID e detalhando algumas informações importantes sobre as etiquetas RFID. Dentro da mesma secção é feita uma descrição do kit trabalhado.

Na secção 3 fala-se do que é o sistema GNU Radio, que também foi usado no estágio. Na secção a seguir estão os passos para realizar a captura do sinal, passando pela montagem de um programa de leitura de etiquetas, montagem do programa de captura do sinal, construção da antena, culminando com a apresentação dos resultados.

1. Introdução

Para a realização dos trabalhos foi usado um kit já existente no laboratório, o Phidgets USB 125 kHz. O trabalho proposto foi primeiramente colocar o kit para funcionar em diversos sistemas, e paralelamente a isso realizar um guia para colocar em prática seu funcionamento, evitando assim que pessoas que venham a utilizá-lo futuramente percam tempo pesquisando sobre com fazê-lo e que possam, por conseguinte, concentrar-se mais em sua aplicação.

Feito isso, o próximo passo era realizar a captura do sinal de comunicação entre o leitor (*reader*) e a *tag*, também chamada de etiqueta e *transponder*, mostrando-o tanto no domínio do tempo quanto no da frequência. Uma vez que o sinal tenha sido capturado, a proposta era estudá-lo para tentar imitar o sinal da *tag*, para que assim pudesse ser construído um emulador da *tag*,

Houve muitas ideias sobre como fazê-lo, foi feita uma pesquisa para saber se outras pessoas tinham feito o mesmo e como elas fizeram. Ao final o sistema GNU Radio foi escolhido. O GNU Radio é um kit de desenvolvimento de rádio por software que também já estava presente no início do estágio.

2. RFID

2.1.Descrição da tecnologia

A identificação por frequência de rádio RFID (*Radio Frequency Identification*) é uma tecnologia de identificação automática (Auto-ID) que usa ondas de rádio para fazer sua identificação. Surgiu inicialmente como substituta de outra tecnologia Auto-ID bastante famosa, o código de barras. Com a evolução da tecnologia e o surgimento de *tags* com memórias regraváveis e aumento do alcance e velocidade de leitura seu uso passou a ser mais abrangente. Além de identificar produtos em ambientes comerciais e industriais (substituindo o código de barras) o RFID é também utilizado para identificar pacientes em hospitais, veículos de carga e animais silvestres e de criação, sistemas de localização e aquisição de dados. Em controle de acesso substitui as chaves comuns, fáceis de serem copiadas.

O RFID tem vantagens frente às tecnologias que procura substituir. Seu sistema é de fácil leitura, o objeto a ser lido não precisa estar necessariamente no campo de visão do leitor, como ocorre com o código de barras. O alcance de leitura também é superior, mesmo para sistemas passivos (sem bateria para aumentar a energia do sinal). A quantidade de dados a serem armazenados e a possibilidade de regravação é outra importante vantagem dele.

O custo ainda é uma desvantagem em aplicações de larga escala, como identificação de produtos em supermercados. Mesmo com grande esforço do Wal-Mart (maior rede mundial de supermercados) ver etiquetas RFID substituindo os códigos de barras nos supermercados não está tão próximo como se previu. No entanto, em aplicações de menor escala, onde a eficiência é mais importante que o preço, o RFID já conquista grande parte do mercado.

Seja qual for a aplicação que o sistema RFID tenha, os elementos que o compõe serão os mesmos, sendo eles: *tag*, *reader* e *host* (Figura 1). As *tags*, também chamadas de etiquetas, são acopladas ao objeto a ser identificado, possuindo um chip que armazena seu respectivo código de identificação. O *reader* (leitor) é o dispositivo que interroga as *tags*, por meio de ondas eletromagnéticas enviadas pelo mesmo, e adquire as informações nela contidas. Ele é conectado a um computador (*host*) ou outro componente para processamento da informação da *tag* e tomada de decisões.

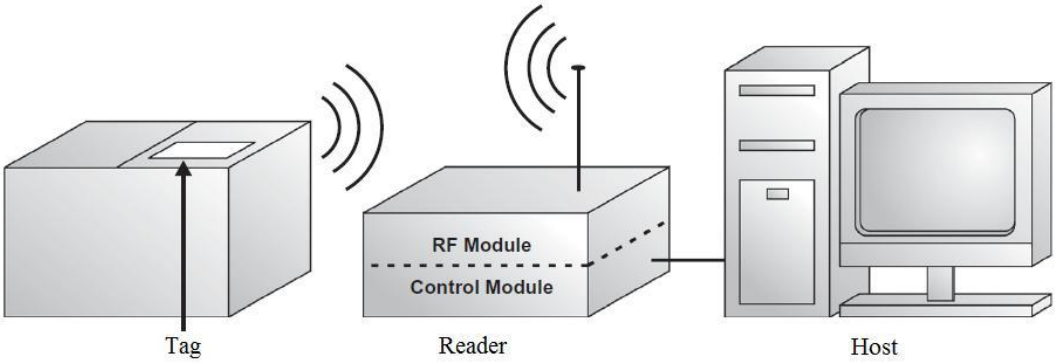

Figura 1: Componentes de um sistema RFID/Fonte [2]

O *host* pode ser não apenas um computador, mas também outros sistemas com capacidade de processamento, como microcontroladores, PDAs e CLPs. Para tanto cada *reader* tem uma ou mais portas de comunicação compatíveis com o *host* para com o qual foi projetado para trabalhar.

A transferência de dados nos sistemas RFID ocorre através de uma conexão entre a etiqueta e o leitor conhecida como acoplamento. O acoplamento pode ser do tipo magnético (indutivo) ou eletromagnético (*backscatter*). No acoplamento magnético o comprimento de onda do sinal emitido pelo leitor é várias vezes maior que a distância entre ele e a *tag*, o campo eletromagnético pode então ser tratado como um campo magnético alternado com relação a distância entre a etiqueta e o leitor. No acoplamento eletromagnético trabalha-se em frequências superiores as do caso anterior, e devido aos pequenos comprimentos de onda, não mais podemos simplificar a análise da onda em termos da componente magnética (Figura 2).

No acoplamento indutivo a *tag* absorve a energia enviada pelo leitor. Já no acoplamento *backscatter* a *tag* "reflete" a resposta de volta para o leitor (similar à operação de um radar).

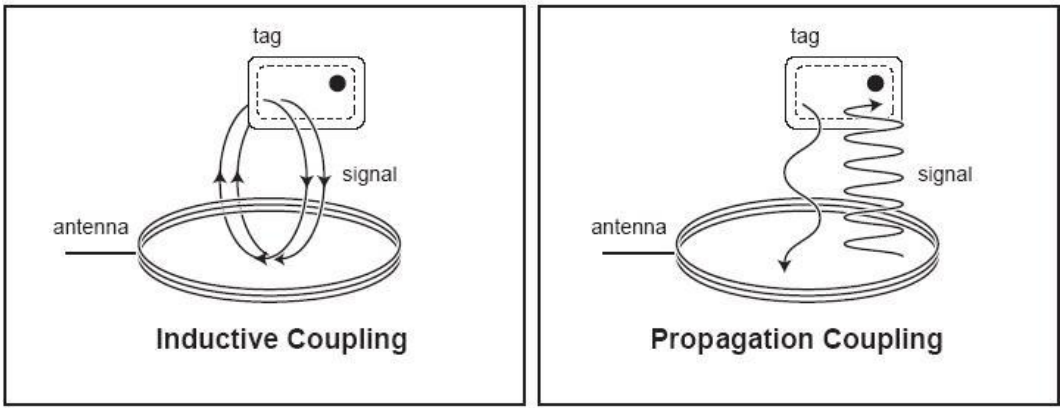

Figura 2: Acoplamento indutivo e backscater

2.2. Host

O *host* é o sistema computacional, dedicado ou não, que executará as tarefas de processamento para o sistema RFID em questão. O tipo de *software* a ser executado está associado com as características do *host*. É ele quem comanda as operações a serem realizadas pelo leitor. Cada leitor possui um protocolo de comunicação diferente com seu *host* e deve haver um *software* no *host* capaz de realizar essa comunicação.

2.3.Leitor

O leitor é o agente intermediário entre o *host* e a etiqueta. É ele quem se comunica diretamente com as etiquetas, através dos comandos do *host* ele executa uma operação (geralmente leitura de ID, escrita na memória ou leitura da memória) e responde ao *host* depois que finalizar sua comunicação com a *tag*.

No caso de *tags* passivas, o leitor fornece a energia requerida para energizar a *tag* por meio do seu campo eletromagnético. O alcance deste campo é geralmente determinado pelo tamanho da antena de ambos os dispositivos e pela potência do leitor. O tamanho da antena geralmente é definido pelas necessidades da aplicação e da frequência utilizada. Geralmente a potência é bem baixa, visto que o alcance também é, aproximadamente até 15m.

A freqüência de operação pode variar de acordo com as especificações, padrões e regulações. O RFID utiliza as faixas de frequência regulamentadas peta ITU (*International Telecommunication Union*) conhecidas como ISM (*industrial, scientific and medical*). As faixas de freqüências mais comuns são: baixa freqüência (LF), 135 kHz ou menos, geralmente se utiliza 125 kHz; alta freqüência (HF), 13,56 MHz; ultra

alta freqüência (UHF) a partir de 433 MHz a 950 MHz; e microondas, 2,45 GHz ou 5,8GHz.

A freqüência está intimamente relacionada com a taxa de transferência de dados entre a *tag* e o leitor. Quanto menor a freqüência, menor será a taxa de transferência. *Tags* passivas são construídos em frequências mais baixas, de 125 kHz até as faixas de UHF, enquanto a maioria das *tags* ativas está na frequência de 2,45 GHz, mas encontram-se também nas frequências de 433 MHz e 5,8 GHz.

2.4.Tags

As *tags* são compostas basicamente de um chip, onde está um circuito integrado (CI) com memória, uma antena para transmissão, e o encapsulamento, que protege as partes internas e possui estrutura própria para se acoplar ao objeto a ser identificado. As *tags* mais comuns hoje em dia possuem um circuito integrado (CI) com memória, essencialmente um microprocessador, mas existem etiquetas que não possuem CI. Estes *tags* são mais efetivos em aplicações onde simples funcionalidades são requeridas. A tarefa mais comum de uma *tag* quando ela é interrogada é transmitir o ID gravado em sua memória, ela pode ainda executar outras tarefas básicas (leitura/escrita) ou manipular os dados em sua memória. Na Figura 3 encontram-se alguns exemplos de *tags*. Já na Figura 4 é mostrada uma etiqueta em close e de encapsulamento transparente, evidenciando suas partes internas principais, o chip e a antena.

Figura 3: Exemplos de tags

Figura 4: Estrutura de uma tag/Fonte: www.sagedata.com

Quanto à memória as *tags* podem ser *read-write* (RW), quando elas possuem memória que pode ser alterada, ou *read-only* (RO), quando todo o conteúdo de sua memória é fixo. A capacidade de escrever na memória eleva o custo de um *tag*, assim como a capacidade de executar funções de alto nível. Uma vantagem de *tags* somente leitura (RO) é a eliminação do risco de escrita acidental.

Em *tags* com memória RW ela em geral divide-se em duas partes. Uma fixa, chamada de ID, e outra regravável chamada de memória de dados. Também é possível encontrar *tags* que não fazem distinção dos dois tipos de memória, como também *tags* configuráveis, cuja memória pode ser uma memória de dados ou um ID, de acordo com a configuração.

Quanto à sua fonte de alimentação as *tags* podem ser passivas ou ativas.

***Tags* Passivas:** As *tags* passivas não possuem, em sua placa, uma fonte de alimentação (bateria), eles usam a energia emitida pelo leitor para suprir sua demanda e transmitir a informação de sua memória. Possuem boa resistência a ambientes rigorosos, pois não possuem partes móveis. Para este tipo de *tag*, o leitor sempre inicia a comunicação. Sua faixa de alcance de leitura vai desde menos de 1cm até, aproximadamente, 15m. Uma grande vantagem das *tags* passivas é que sua vida útil é virtualmente infinita. Uma vez que não necessitam de fonte de energia interna, enquanto

não forem danificados estarão em funcionamento, assim ele possui vida útil virtualmente infinita. Existem *tags* passivas com memórias RO e RW.

***Tags* Ativas:** As *tags* ativas possuem uma fonte de energia interna e circuitos (ou equipamentos) eletrônicos para tarefas específicas. Utiliza sua fonte para transmitir dados ao leitor, não precisando da energia deste. Os circuitos eletrônicos podem conter microprocessadores, sensores e portas de entrada/saída alimentadas pela fonte interna. Desta forma, por exemplo, estes componentes podem medir a temperatura ambiente e gerar uma média. Os componentes podem utilizar esta informação para determinar outros parâmetros tais como a data de vencimento do produto ao qual a *tag* está anexada, a *tag* então envia estas informações para o leitor (junto com seu ID).

A comunicação pode ser iniciada pela *tag* ou pelo leitor. A *tag* pode transmitir continuamente mesmo na ausência de um leitor. Outro tipo de *tag* ativa permanece em estado de espera (estado de baixo consumo de energia - *sleep state*) na ausência de um leitor. O leitor então desperta a *tag* do *sleep state* a partir de um comando apropriado. A distância de leitura de uma etiqueta ativa pode chegar a 1km. Mas você não precisa trabalhar sempre na distância máxima. Para fins de economia de energia do leitor e da etiqueta você pode configurá-los para trabalhar com área de cobertura menores.

Uma desvantagem comercial da *tag* ativa é que ela, ao contrário da passiva, possui vida útil finita, determinada pela sua bateria. Como dito antes a *tag* pode entrar em modo de espera, com o objetivo de economizar sua bateria e aumentar sua vida útil. Em condições de operação, o tempo de vida útil médio de uma *tag* ativa varia de 3 a 7 anos, mudando com o fabricante. A maioria dos produtos disponíveis no mercado apresenta vida útil de 5 anos.

Sistemas Ativos x Sistemas Passivos O que define um sistema RFID como ativo ou passivo é basicamente o tipo de *tag* utilizada. Já foi visto a diferença básica entre as duas *tags*. A ativa possui fonte de energia interna e a passivo não. Essa diferença faz com que com a etiqueta ativa possua alcances de leitura e escrita muito maiores, assim suas aplicações se diferem baseadas nessa característica.

Por possuírem maior alcance de leitura as *tags* ativas são usadas em sistemas de localização e rastreamento. Para aplicações de curta e média distância *tags* passivas são mais recomendadas, visto que o longo alcance das *tags* ativas é desnecessário ou mesmo desfavorável a suas aplicações. Sistemas passivos são usados em controle de

acesso, sendo um substituto para chaves em portas tradicionais e para senhas em portas com trava elétrica. Para tanto usa-se geralmente cartões de 125kHz RO, como não necessariamente será preciso gravar informações no cartão de acesso uma memória RW adicional é dispensável. Mas caso deseje-se gravar algum tipo de informação, cartões com memórias também são encontrados.

2.5. Phidgets USB 125 kHz

Para o desenvolvimento do trabalho de estágio foi usado o kit Phidgets USB 125 kHz. O kit consiste basicamente do leitor Phidgets USB 125kHz e de um conjunto de *tags* EM4102. Como dito no próprio nome do leitor, ele se comunica com o *host*, neste caso um computador, através da porta USB. O leitor não possui botão de on-off, assim ao ter sua entrada USB conectada no computador ele já estará ligado. Existem diversos tipos de *tag*, que se diferem quanto ao encapsulamento, há cartões, discos opacos, *tags* para implantes, uma pulseira, discos transparentes que permitem a visualização da antena e do chip.

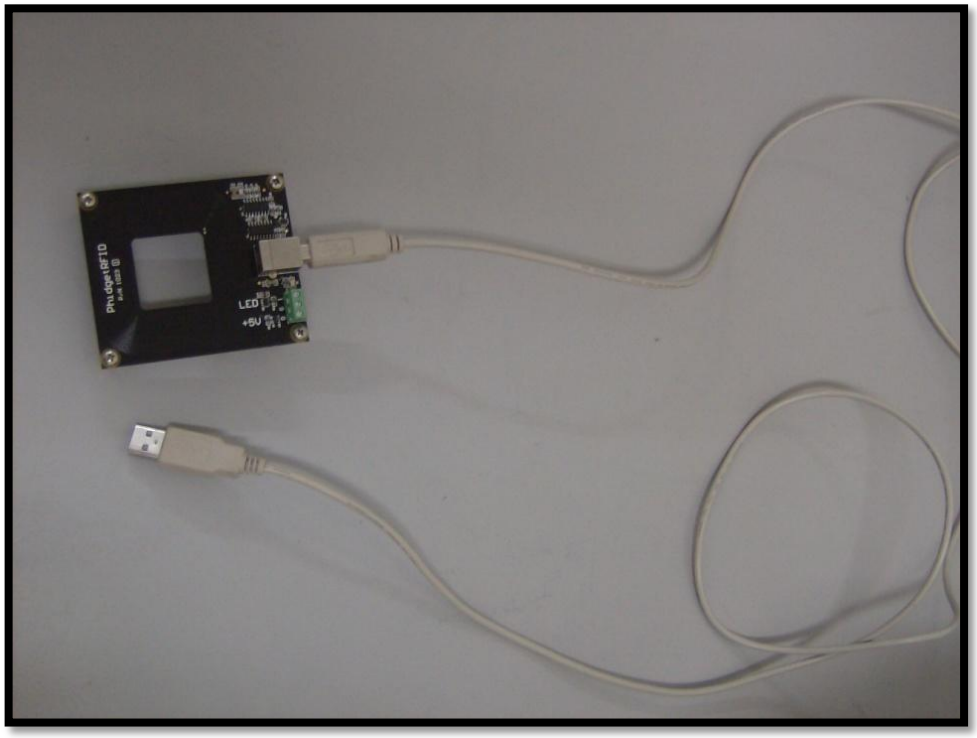

Figura 5: Leitor Phidgets USB 125 kHz

Figura 6: Conjunto de tags EM4102

Existem alguns programas exemplos disponíveis no endereço www.phidgets.com na secção *Programming* que servem para testar o funcionamento do leitor. No endereço www.phidgets.com/drivers.php estão disponíveis as bibliotecas, nelas estão contidas as funções utilizadas pelos programas que utilizam o kit, essas bibliotecas são chamadas de *Phidgets*. Para o sistema operacional *Windows XP* é necessário ter o ServicePack2 instalado para fazer a instalação do *Phidgets*. Existe uma biblioteca para o sistema *Linux* também, que requer a versão *Linux Kernal 2.6+* para sua instalação. Existem várias versões do arquivo de bibliotecas, quando for rodar algum programa é necessário conferir que tipo de versão da biblioteca ele utiliza, para evitar problemas de compatibilidade.

2.6. Etiquetas EM4102

Os dados que serão mostrados nesta secção dizem respeitos às etiquetas EM4102, que são as usadas pelo kit Phidgets. Os dados foram retirados de um manual descritivo técnico da empresa EM Microeletronic.

O EM4102 (antes chamado de H4102) é um circuito integrado do tipo CMOS usado em *transponders* RF *Read Only*. O circuito é alimentado por um campo magnético externo, recebendo dele também o *clock master*. Quando ele é energizado ele manda um sinal RF de resposta com 64 bits de informação contidos em sua memória.

A taxa de dados pode ser igual a 64, 32 ou 16 vezes o período da frequência da portadora, como mostrado na Figura 7. Os dados são codificados com código de Manchester, Bifásico ou PSK (*Phase Shift Keying*). É preciso apenas uma bobina para suprir as funções do chip.

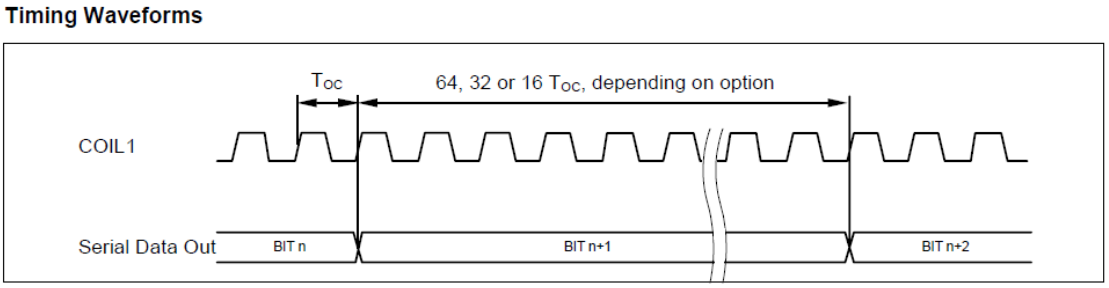

Figura 7: Fonte [4]

O fabricante define três aplicações para suas etiquetas:

- *Transponder* implantável em animais
- *Transponder* para orelhas de animais
- *Transponders* industriais

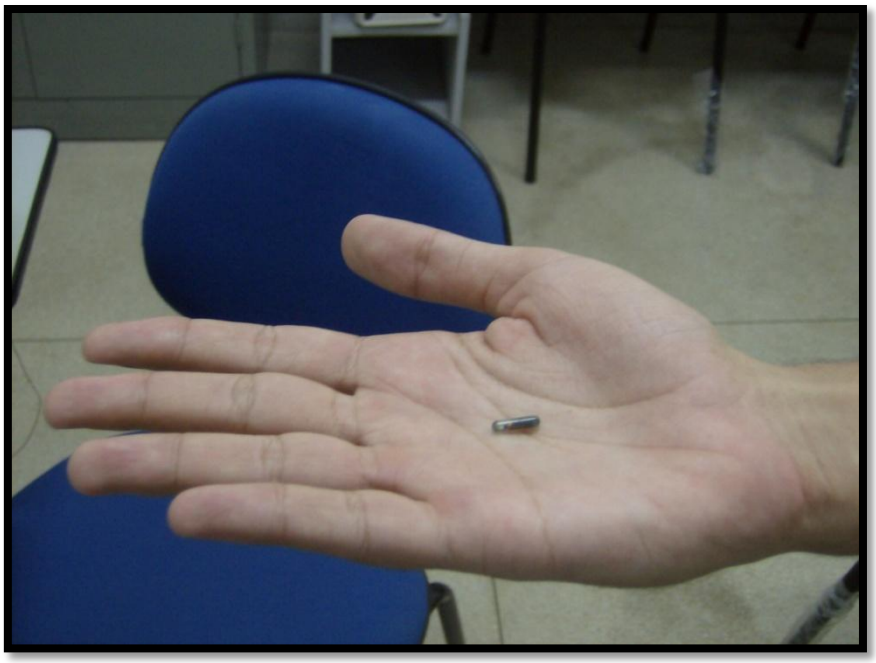

Figura 8: Transponder para implante em animais

Na Figura 9 está um esquema simplificado do *tranceiver* (leitor) e do *transponder* (etiqueta), na Figura 10 o esquema do *transponder* é mostrado mais detalhadamente.

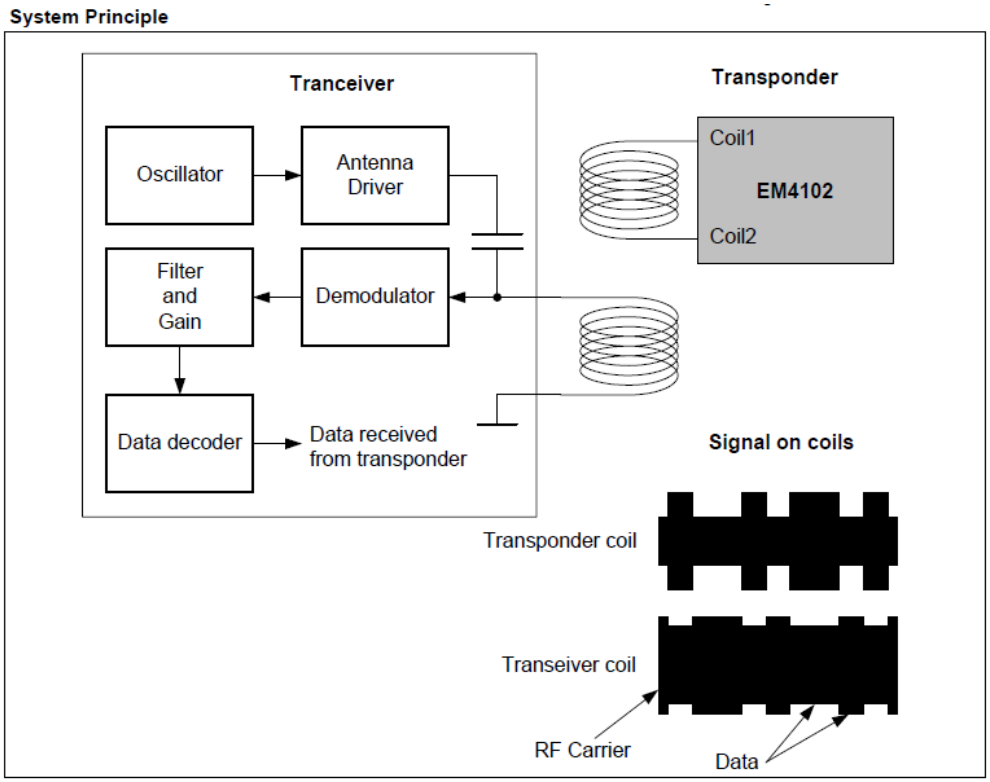

Figura 9: Tranceiver e transponder/Fonte [4]

Block Diagram

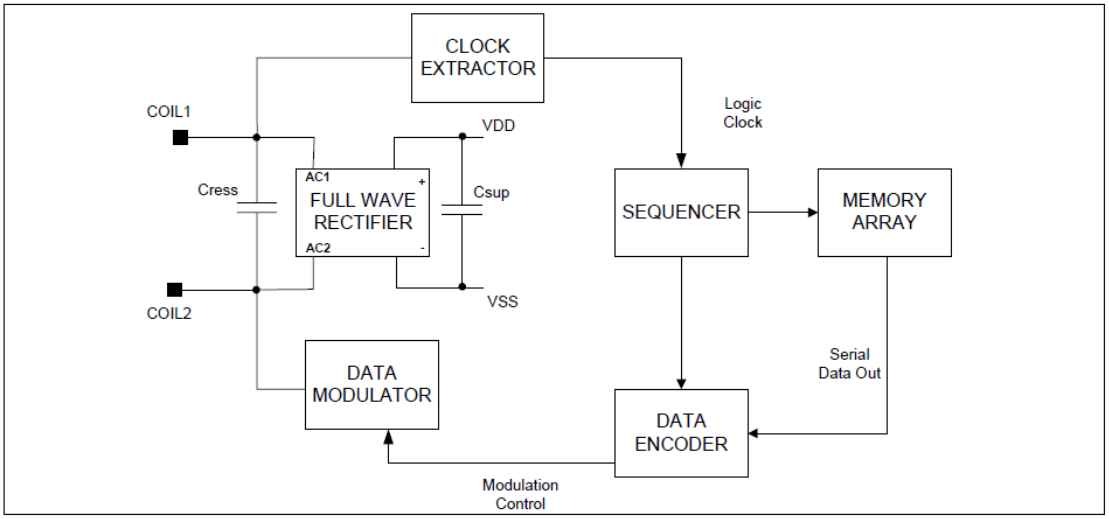

Figura 10: Transponder/Fonte [4]

A seguir é explicado cada bloco:

- **Geral:** O EM4102 é alimentado com um campo eletromagnético induzido na bobina do *transponder*. A voltagem AC é retificada para prover uma fonte de tensão interna DC. Quando o último bit é enviado, o chip continuará a mandar o sinal a partir do primeiro bit até que a fonte de energia desapareça.
- **Full Wave Rectifier:** A entrada AC induzida pelo campo incidente na bobina é retificado por uma ponte de Graetz. A ponte limita a tensão DC interna para evitar problemas com campos fortes.
- **Clock Extrator:** Um dos terminais da bobina (Coil 1) é usado para gerar um *master clock* para funções lógicas. A saída do *clock extractor* se dirige ao *sequencer*.
- **Sequencer:** O *sequencer* provê todo sinal necessário para endereçar a sequência de memória e codificar a saída serial de dados. Três tipos de codificações lógicas são possíveis, código de Manchester, bifásico e PSK. A taxa de bit para os dois primeiros pode ser 64 ou 32 períodos da frequência da portadora. Para a versão PSK, a taxa é 16. O *sequencer* recebe o *clock* do *clock extractor* e gera todo sinal interno que controla a memória e o codificador lógico de dados.
- **Data Modulator:** O modulador de dados é controlado pelo Modulation Control para induzir uma alta corrente na bobina. O transistor de Coil 2 carrega a alta corrente. Isto vai afetar o campo magnético de acordo com os dados estocados na memória.

- **Memory Array para codificação de Manchester e bifásico:** O EM4102 contém 64 bits divididos em 5 grupos de informação, 9 bits são usados para o cabeçalho, 10 bits de paridade para linhas (P0 – P9), 4 bits de paridade para colunas (PC0 – PC3), 40 bits de dados (D00 – D93), e 1 bit de parada com valor lógico 0. A estrutura dos dados é mostrada na Figura 11.

1	1	1	1	1	1	1	1	1	9 header bits
8 version bits or customer ID				D00	D01	D02	D03	P0	
				D10	D11	D12	D13	P1	
				D20	D21	D22	D23	P2	
32 data bits				D30	D31	D32	D33	P3	
				D40	D41	D42	D43	P4	
				D50	D51	D52	D53	P5	
				D60	D61	D62	D63	P6	
				D70	D71	D72	D73	P7	
				D80	D81	D82	D83	P8	
				D90	D91	D92	D93	P9	10 line parity bits
				PC0	PC1	PC2	PC3	S0	
				4 column parity bits					

Figura 11: Estrutura da memória/Fonte [4]

O cabeçalho é composto de 9 bits com todos os valores 1. Assim com a organização de dados e bits de paridade essa sequência não pode mais ser repetida. O cabeçalho é seguido por 10 grupos de 4 bits, permitindo 100 bilhões de combinações, há para esse grupo 1 bit de paridade par para cada linha. O último grupo consiste de 4 bits de paridade para colunas, sem um bit de paridade de linha para esta linha. S0 é o bit de parada que é sempre 0. Os bits D00 a D03 e os bits D10 a D13 são identificações específicas do cliente. Esses 64 bits são enviados de forma serial e quando o último bit é enviado o primeiro bit é enviado novamente, repetindo a sequência continuamente até que cesse a fonte de energia.

- **Memory Array para PSK:** O PSK é programado com paridade ímpar para os bits P0 e P1 e sempre com lógica zero. A paridade dos bits para P2 a P9 é par. A paridade de coluna PC0 a PC3 são calculadas incluindo os bits de versão e são de paridade par.

A seguir são explicadas as três codificações lógicas possíveis, na Figura 12 há um exemplo de sinal com cada uma das codificações para melhor explicá-las.

- **Manchester:** Há sempre uma transição de ON para OFF ou de OFF para ON no meio do período do bit. Se a mudança é de OFF para ON o bit lógico será 1, e se é de ON para OFF será 0. Assim quando ocorre uma mudança de fase no sinal do modulador ocorrerá uma mudança de bits nos dados.
- **Código Bifásico:** No início de cada bit uma transição sempre irá ocorrer. Se ocorrer uma transição de valor lógico no meio do período do bit o valor lógico do bit de dados será 0, se o valor lógico se mantiver durante todo o período do bit o valor lógico será 1.
- **PSK:** Quando uma mudança de fase ocorre, um 0 lógico é lido da memória. Se nenhuma mudança de fase ocorre depois do ciclo da taxa de dados um 1 lógico é lido.

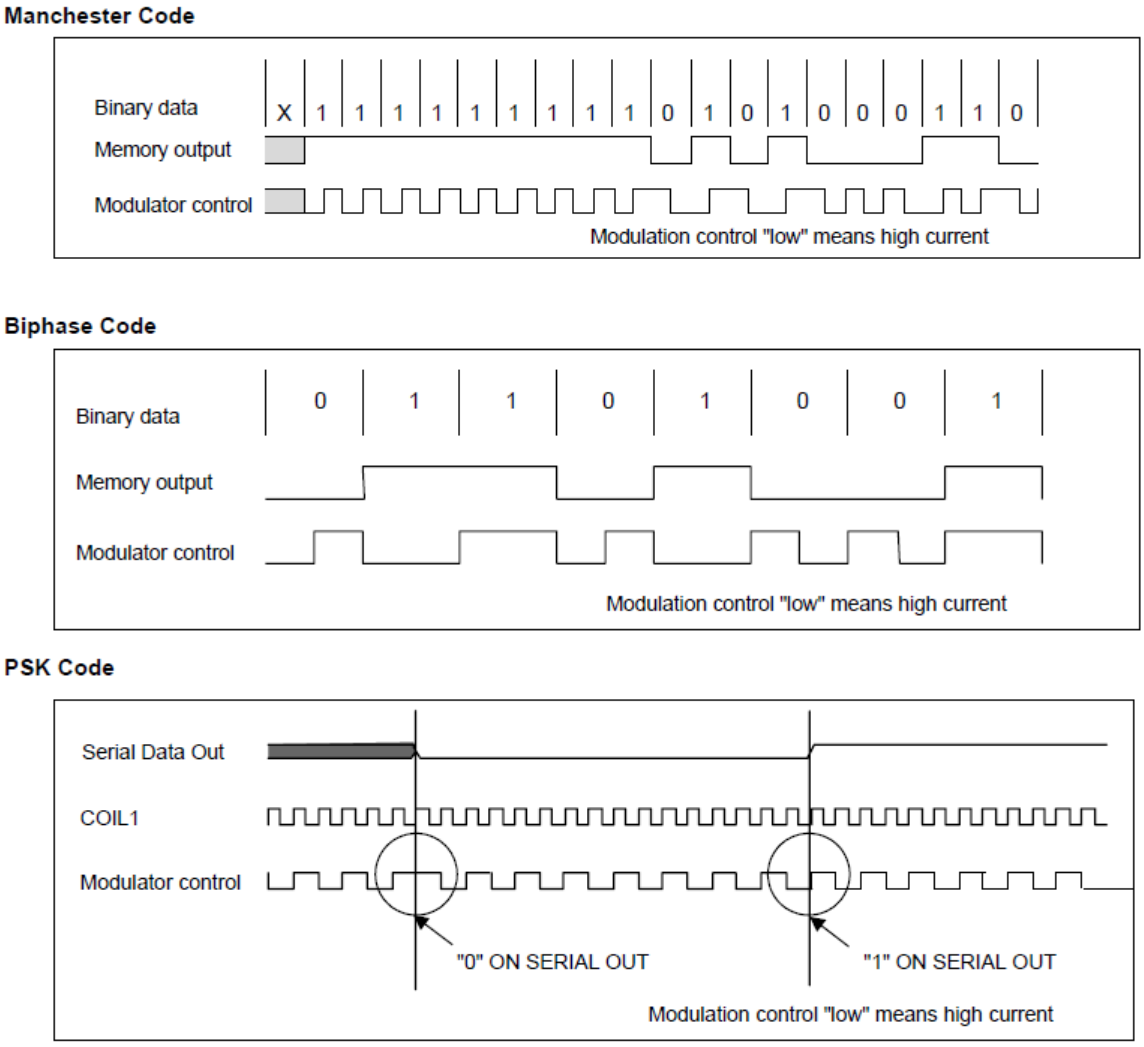

Figura 12: Codificações/Fonte [4]

3. GNU Radio

GNU Radio é um kit de ferramentas de desenvolvimento de software livre que provê processamento de sinais em tempo real e blocos de processamento para implementar radio por software. É usado em ambientes comerciais, acadêmicos ou mesmo por hobbie para ajudar em pesquisas de comunicações sem fio bem como para implementar sistemas de rádio do mundo real.

As aplicações de GNU Radio são escritas usando a linguagem de programação Python, já o processamento de sinais é implementado em linguagem C++.

O GNU Radio é feito para o sistema Linux. Existe um conjunto de ferramentas incluindo programas exemplos que são instalados conjuntamente. Ele permite fazer programação em Python e C++ (mais baixo nível, mexendo diretamente com o processamento de sinais). Ele também conta com uma ferramenta na qual você pode montar sistemas usando blocos funcionais.

Seu hardware é composto por uma placa USRP. O laboratório possui duas destas placas, na Figura 13 é mostrada uma delas.

Figura 13: Placa USRP

4. Captura do Sinal

4.1.Programa de leitura de *tags*

Como o GNU Radio é feito para o sistema *Linux*, foi instalado na máquina o programa *RFID-simple*, feito para *Linux*, um programa simples para leitura das *tags*. Para fazer a instalação os seguintes passos foram seguidos:

- Baixar a biblioteca *Phidget* no endereço www.phidgets.com/drivers.php;
- Decompactar o arquivo baixado;
- Entrar na pasta descompactada pelo terminal do *Linux* e executar o comando *sudo make install.*

Os procedimentos acima instalam a biblioteca, os procedimentos a seguir instalam o programa em si.

- Baixar o *Code Sample* (dentro de C/C++), do mesmo site na secção *Programming*;
- Descompactar o arquivo baixado;
- Entrar na pasta descompactada e executar o comando *./ configure*;
- Posteriormente executar o comando *make*;
- Estes comandos são responsáveis pela compilação. Se o processo ocorreu com sucesso os arquivos *RFID-simple* e *RFID-simple.o* foram gerados.
- Rodar o programa através do comando *sudo ./RFID-simple*

Ao executar o comando *sudo ./RFID-simple* o leitor já deve está conectado ao computador, caso ele não esteja a seguinte mensagem de erro aparecerá:

Waiting for RFID to be attached....Problem waiting for attachment: Given timeout has been exceeded.

O programa faz uma varredura constante à procura de *tags*, quando ele acha alguma mostra no terminal toda a informação guardada na memória que foi enviada pela *tag*, que consiste de 8 bits de versão ou ID do cliente acrescidos de mais 32 bits de dados, totalizando 40 bits que são mostrados em hexadecimal. Enquanto ela estiver no alcance do leitor seu ID será mostrado, quando ela sair do alcance do leitor a mensagem

Tag Lost: (dados da tag) irá aparecer. A Figura 14 mostra um exemplo do programa em funcionamento, nele foi aproximado um cartão RFID com dados 1f00d9c251, que posteriormente foi afastado, em seguida aproximou-se outro cartão, este com dados 1f0074d439.

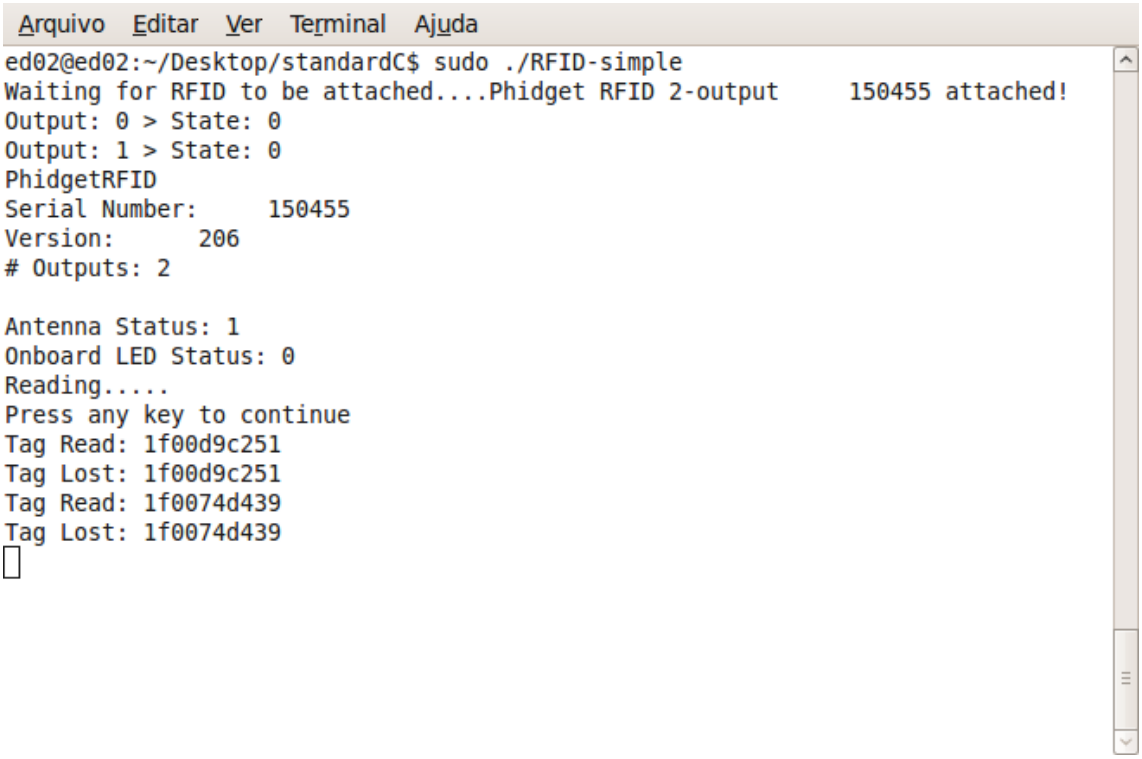

Figura 14: Programa leitor em funcionamento

O leitor não lê mais de uma *tag* ao mesmo tempo. Ao se aproximar uma ou mais etiquetas ele irá identificar apenas uma. Assim se você aproximar duas *tags* a mensagem *Tag Read: (dados da tag)* irá aparecer para apenas uma.

4.2.Programa de visualização no GNU Radio

Com o leitor devidamente instalado e testado houve depois um processo de familiarização com a ferramenta GNU Radio, o objetivo era procurar um programa exemplo que permitisse a visualização do sinal na frequência e no tempo na frequência de 125 kHz. Alguns programas foram olhados, ao fim o programa de visualização do sinal foi feito usando a ferramenta de montagem de programas por blocos funcionais. Para acessar a interface de montagem de blocos usa-se o comando grc no terminal do Linux.

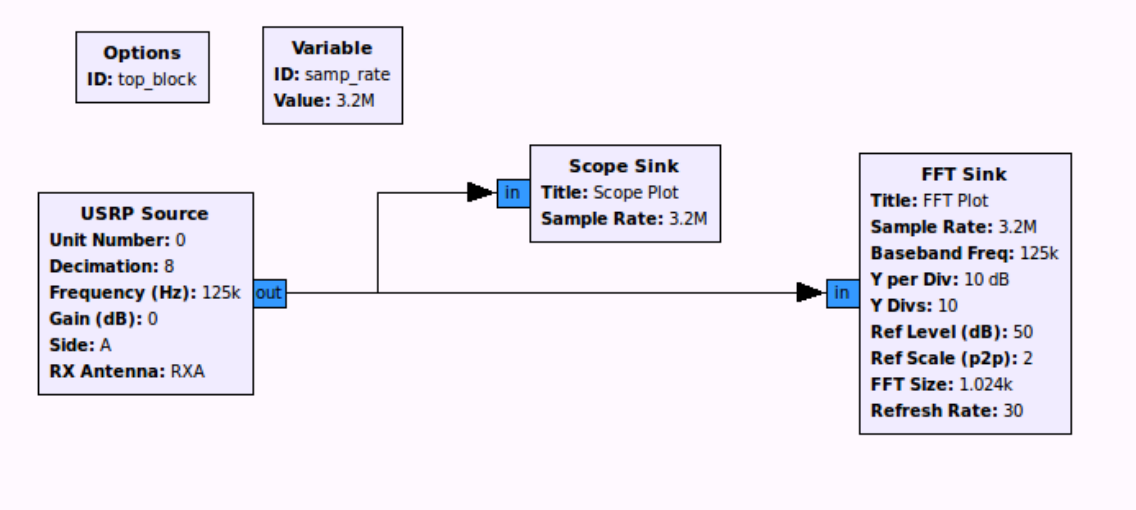

Figura 15: Diagrama de Blocos do GNU Radio

Foram usados três blocos, como mostrado na Figura 15, que é o diagrama de blocos final do programa de visualização do sinal. O USRP Source é a fonte do sinal, que é a entrada do módulo USRP, assim a fonte de entrada para o processamento a ser feito é a fonte externa, na qual foi acoplada uma antena para baixas frequências, fato que será discutido na próxima secção. Seu bloco está contido na aba USRP.

Acoplada à entrada existem duas saídas, o Scope Sink é o osciloscópio, ele é o responsável por mostrar o sinal que vem do USRP Source no domínio do tempo. O FFT Sink é o analisador de espectro, responsável por mostrar o sinal no domínio da frequência. Na Figura 16 estão em destaque os blocos, mostrando as abas de onde eles foram retirados.

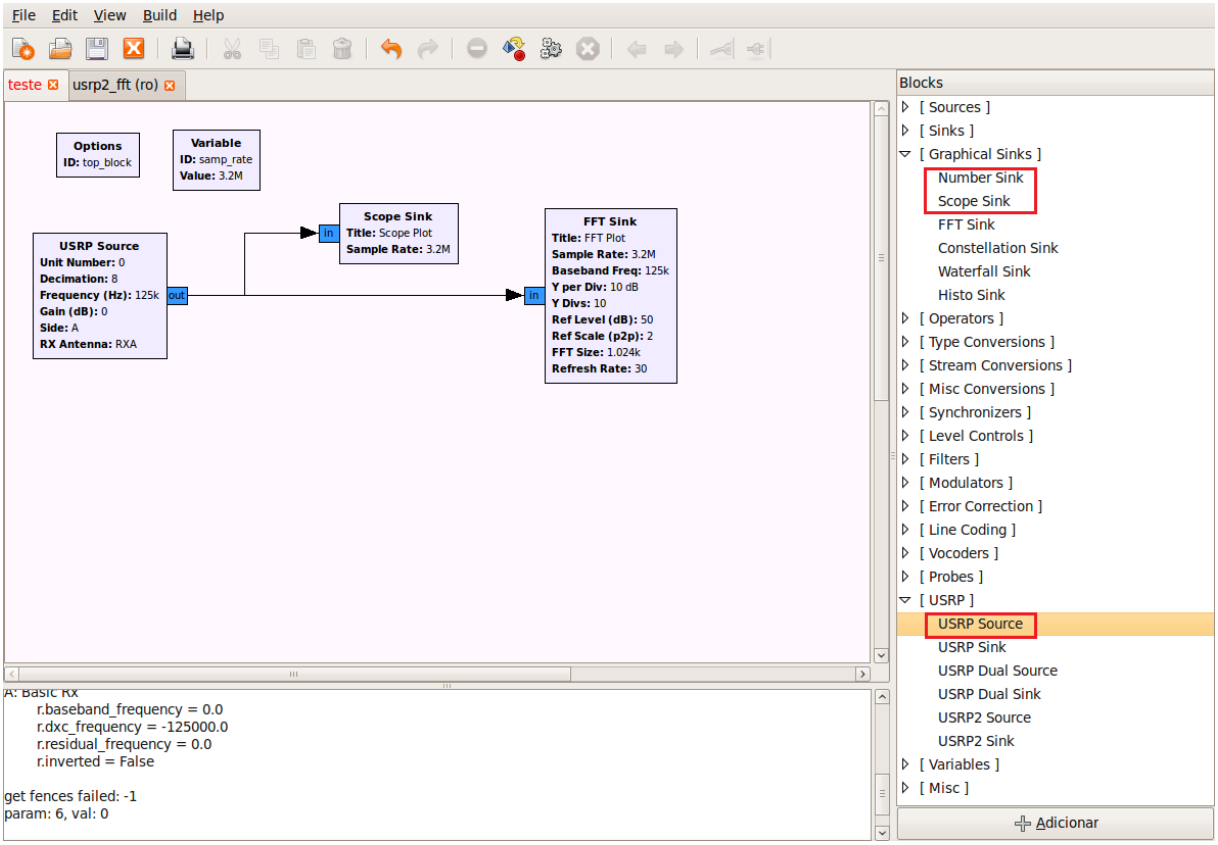

Figura 16

É necessário configurar os 3 blocos para que a captura ocorra e seja mostrada na frequência e intervalos desejados. No USRP Source foi mudado o valor da frequência para 125 kHz inserindo o valor 125e3. O FFT Sink passou por essa mesma alteração. Em relação à faixa de frequência do FFT Sink, vimos que ela é sempre de 1,5 MHz à direita e à esquerda da frequência central, para qualquer frequência central que seja escolhida.

O Scope Sink não passou por nenhuma alteração de seus valores iniciais. A configuração final é mostrada nas Figuras 17, 18 e 19.

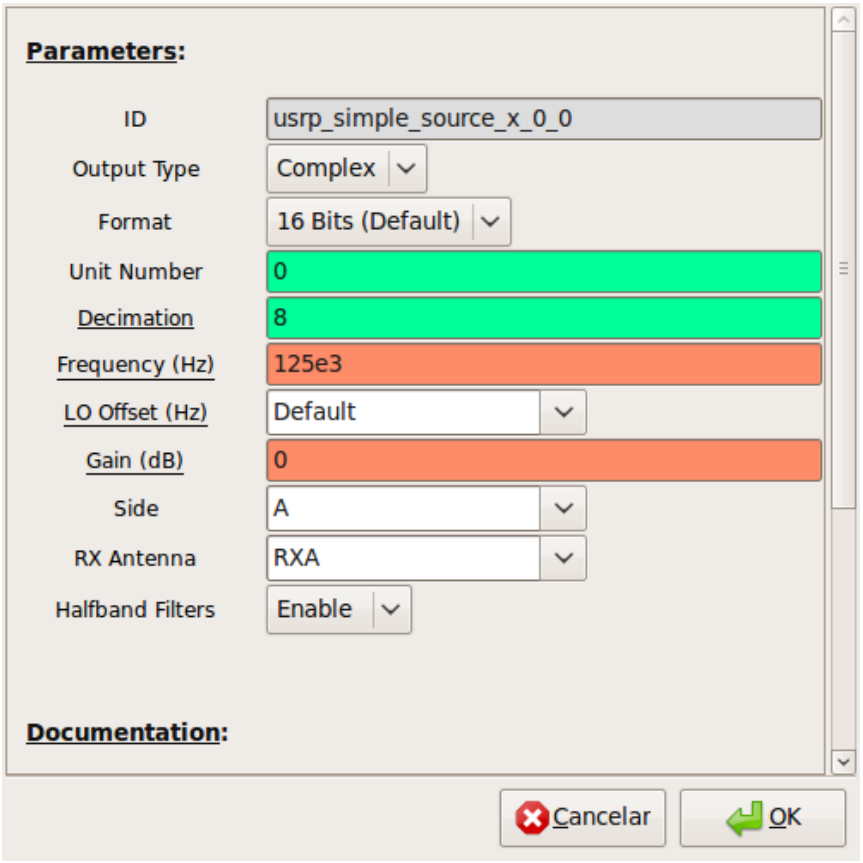

Figura 17: USRP Source

Figura 18: Scope Sink

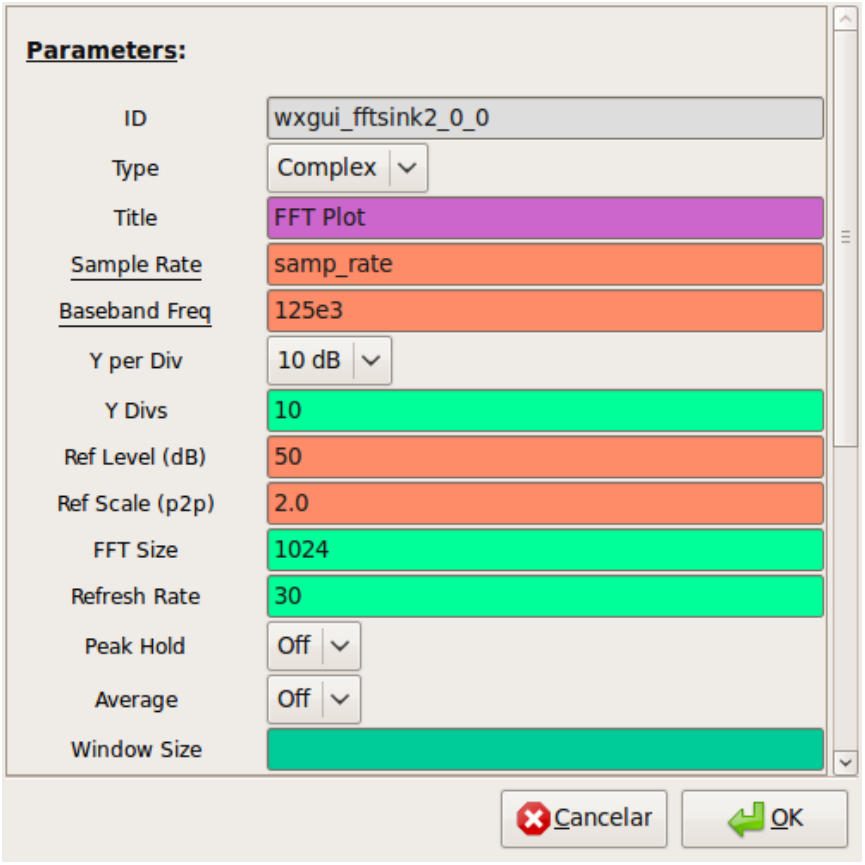

Figura 19: FFT Sink

Algumas configurações são alteradas na interface do osciloscópio para melhor visualização do sinal, uma vez que ele se adequada à amplitude e frequência do sinal inicial, e como inicialmente há apenas ruído em sua leitura, que é um sinal totalmente aleatório e que tem pouquíssimas características em comum com o sinal que realmente queremos ver, principalmente em relação à amplitude. As alterações a serem feitas na visualização do sinal na interface final do programa serão discutidas na secção 4.4, quando serão abordados os resultados obtidos.

4.3. Construção da antena

Ficou clara a necessidade de uma antena acoplada à entrada do GNURadio para fazer a visualização do sinal. Como não havia nenhuma antena na faixa de frequência desejada teve-se de construir uma. Foi feita uma antena loop com medidas de 7cm x 7cm , de 34 voltas. Ela foi feita com fio esmaltado de 0,25 mm de espessura.

Inicialmente tentamos fazer uma antena de medidas 6cm x 6cm , seguindo um guia encontrado para fazer uma antena na faixa de 125 kHz, então procuramos objetos que pudessem servir de apoio para enrolar o fio no formato da antena, o que acabou não sendo encontrado, mas encontrou-se um rolo de papel durex de perímetro 28cm, muito próximo do perímetro desejado, que era de 4x6cm = 24cm. Então enrolou-se o fio em volta do rolo de papel durex, dando as 34 voltas desejadas. Depois de terminar de dar as voltas amarramos com os fios com fita isolante em dois pontos, para assegurar que eles não se separassem. No final ficamos com uma antena circular de 28cm de perímetro.

Usando uma régua para medir o tamanho de cada lado, dobramos a antena circular para dar um formato retangular a ela. Depois que foi dado a ela o formato desejado passamos uma cola específica para fixar um fio no outro, evitando que eles ficassem soltos entre si. Depois de usada a cola, esperamos 15 minutos para que ele secasse. Pronto, a antena em si estava pronta, o resultado pode ser visto na foto da Figura 20.

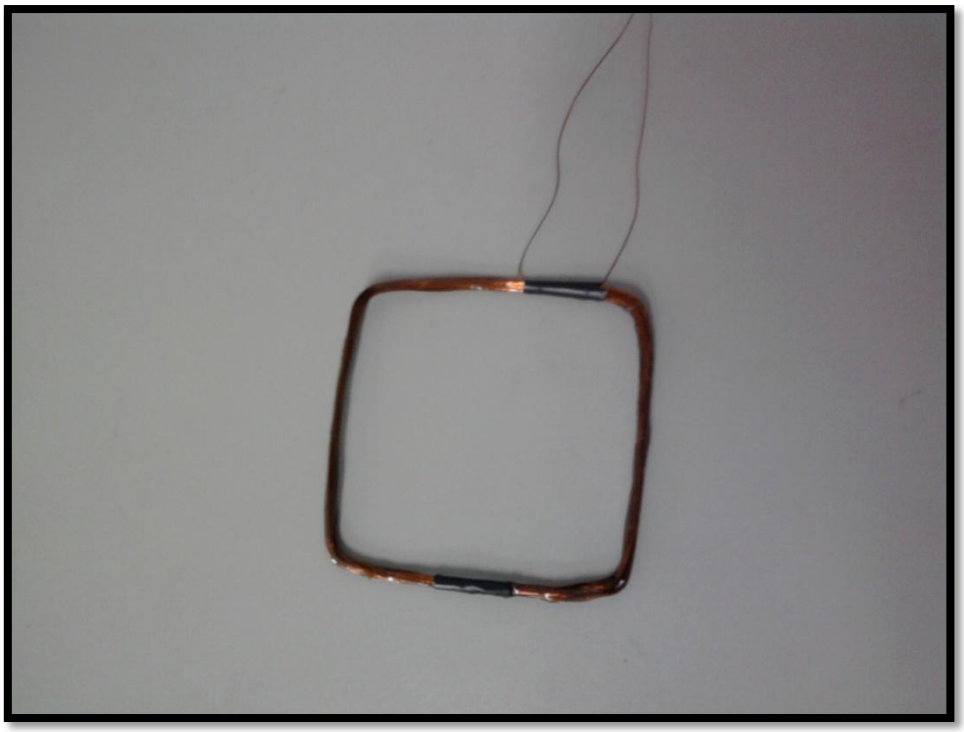

Figura 20: Antena

Pronto, agora restava fazer a conexão da antena com o GNU Radio. O GNU Radio tem uma entrada própria para antenas. Havia vários conectores com uma das pontas soltas no laboratório. Usamos um e simplesmente soldamos os terminais de um lado do conector com os terminais da antena. Fica evidente pelo processo que há

problemas com casamento de impedâncias, mas para a aplicação final o resultado foi satisfatório.

4.4. Resultados

Uma vez que o programa de leitura das *tags* e o programa de visualização do sinal estão prontos podemos efetuar a captura do sinal. Para isso primeiramente conectamos o leitor Phidgtes ao computador, e em seguida executamos o programa RFID-simple, como já dito anteriormente. Depois executamos o programa de visualização do sinal, o que é feito na própria tela de edição, usando o caminho Build → Execute. Para a correta captura do sinal, a antena deve estar acoplada à entrada do GNURadio.

Inicialmente, considerando que a antena esteja longe do alcance do leitor, haverá apenas ruído e um sinal totalmente aleatório será visto. O Scope Sink tem a opção Autorange, que faz um ajuste automático de amplitude de acordo com o sinal que é recebido. Como temos certo conhecimento do sinal que iremos receber e faremos teste afastando e aproximando a antena do leitor o Autorange é indesejado, pois sem ele teremos melhor noção da influência da distância e a diferença entre o sinal e o ruído.

Assim desativamos o Autorange e aumentamos consideravelmente o valor inicial de Counts/Div. Em algumas medições também ajustamos o Secs/DIV que cuida da escala no eixo x (tempo). Já no FFT Plot usamos a ferramenta Autoscale que ajusta a escala para o sinal que está sendo recebido no momento. O sinal visualizado com o leitor desligado é mostrado na Figura 21.

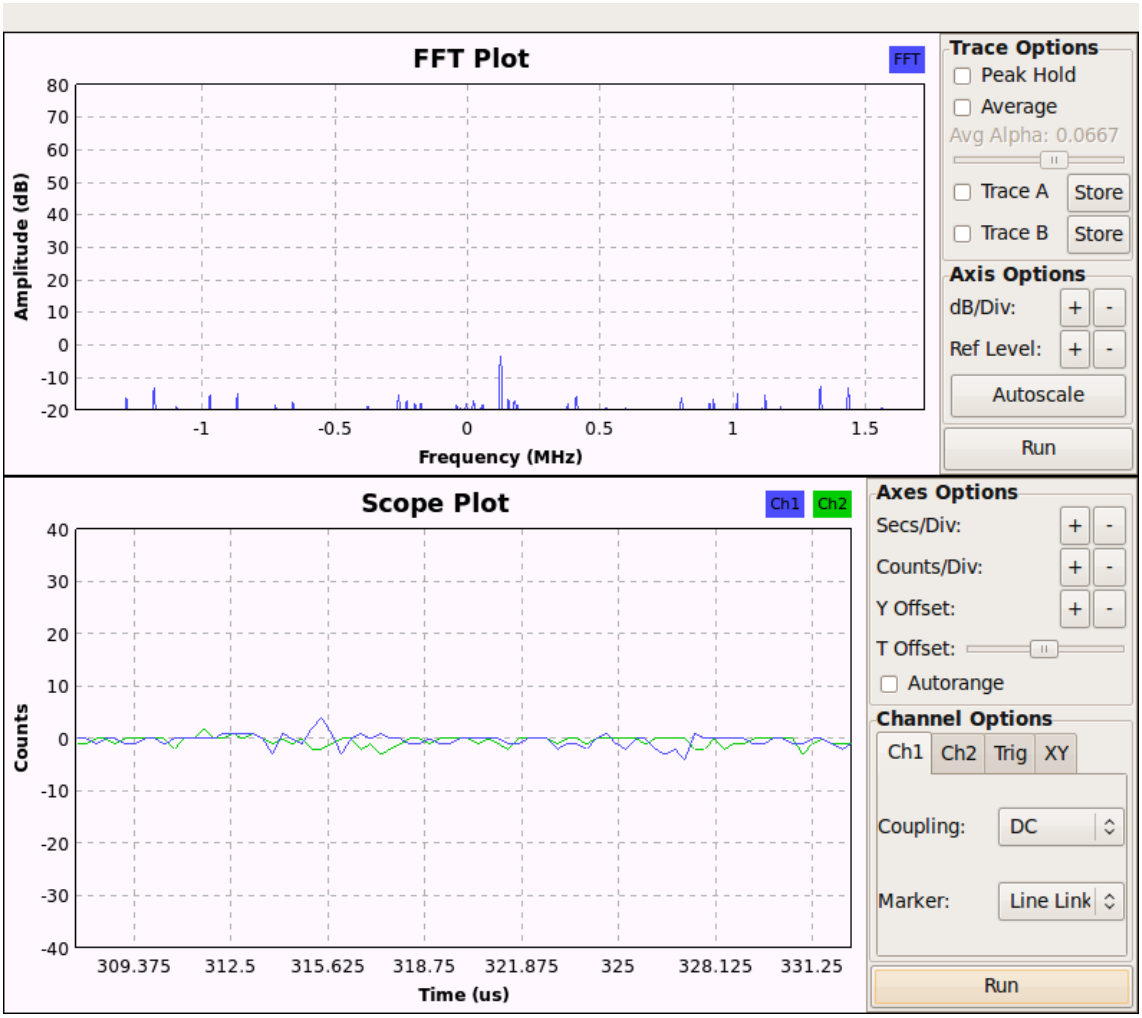

Figura 21: Leitor desligado

Notamos que mesmo fora do alcance de leitura o sinal do leitor ainda é captado a valores acima do ruído local passíveis de serem detectados. Nas Figuras 22 e 23 estão os sinais detectados nessas condições, na primeira o leitor está bem longe, na segunda um pouco mais perto, mas em ambas a antena de captação do sinal está fora do alcance de leitura. Na Figura 23 percebe-se já uma aproximação do sinal à senóide que é enviado pelo leitor para energizar a etiqueta.

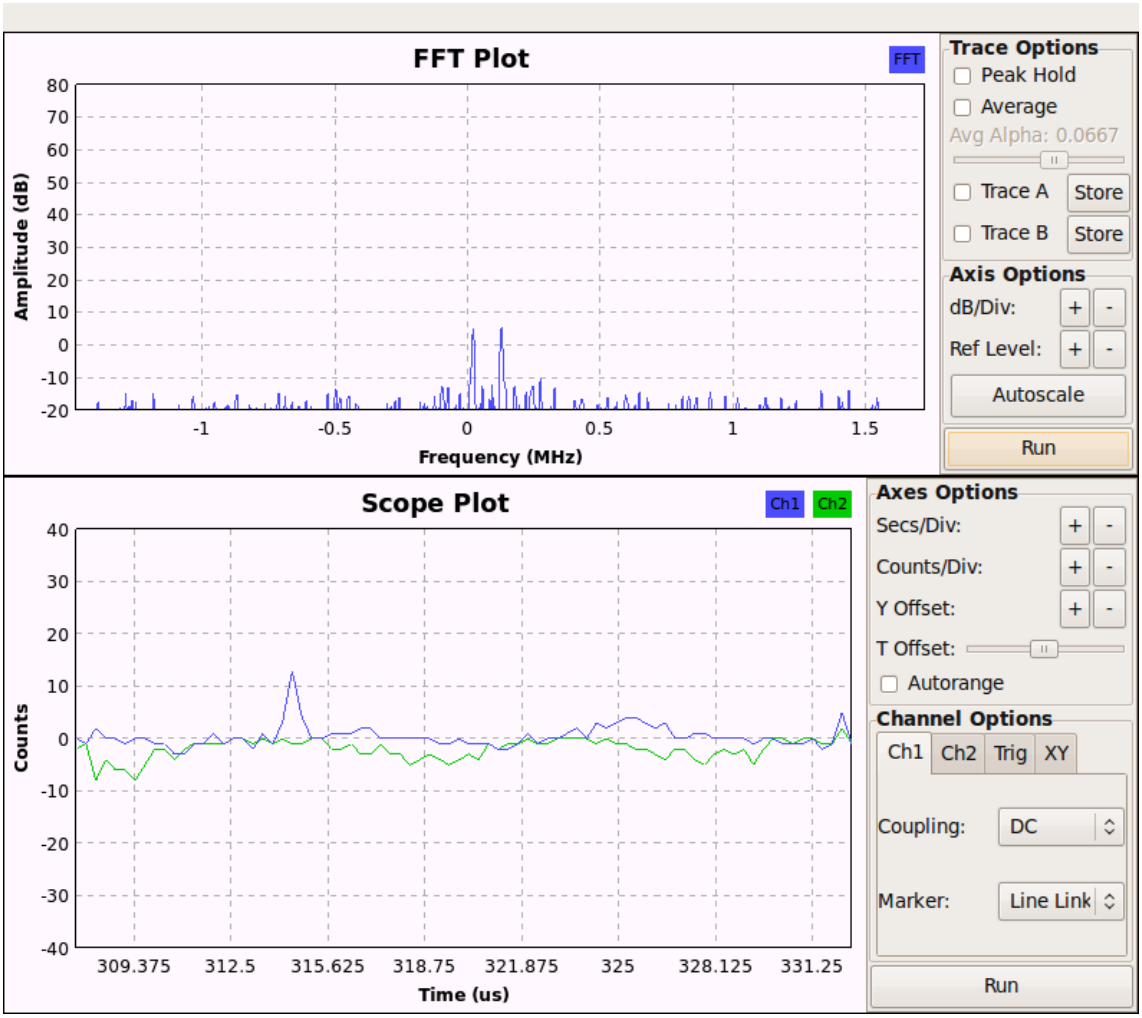

Figura 22: Leitor distante

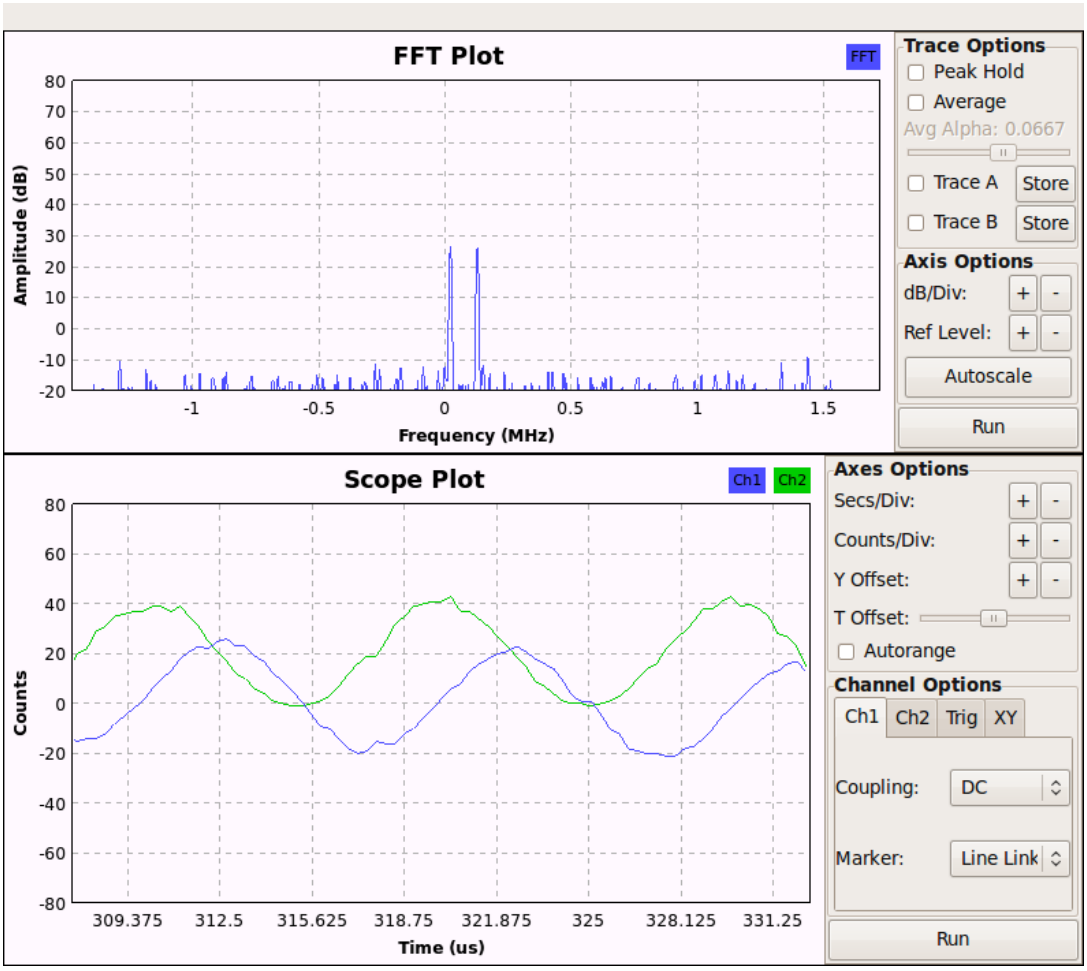

Figura 23: Leitor distante

Na Figura 25 temos o sinal apenas do leitor, já dentro do alcance de leitura, neste caso a antena está a 5cm do leitor. Percebe-se claramente a senóide enviada pelo leitor para energizar a *tag*. Medindo a frequência captada obtivemos o valor de 104,35 kHz. Fora do valor exato de 125 kHz, que era o esperado inicialmente, mas dentro da faixa de operação das *tags*, como mostrado na Figura 24, que é uma tabela do manual da EM Microeletronic.

Operating Conditions

Parameter	Symbol	Min	Typ	Max	Units
Operating Temp.	T_{op}	-40		+85	°C
Maximum Coil Current	I_{COIL}	-10		10	mA
AC Voltage on Coil	V_{coil}	3	14*		Vpp
Supply Frequency	f_{coil}	100		150	kHz

Figura 24: Fonte [4]

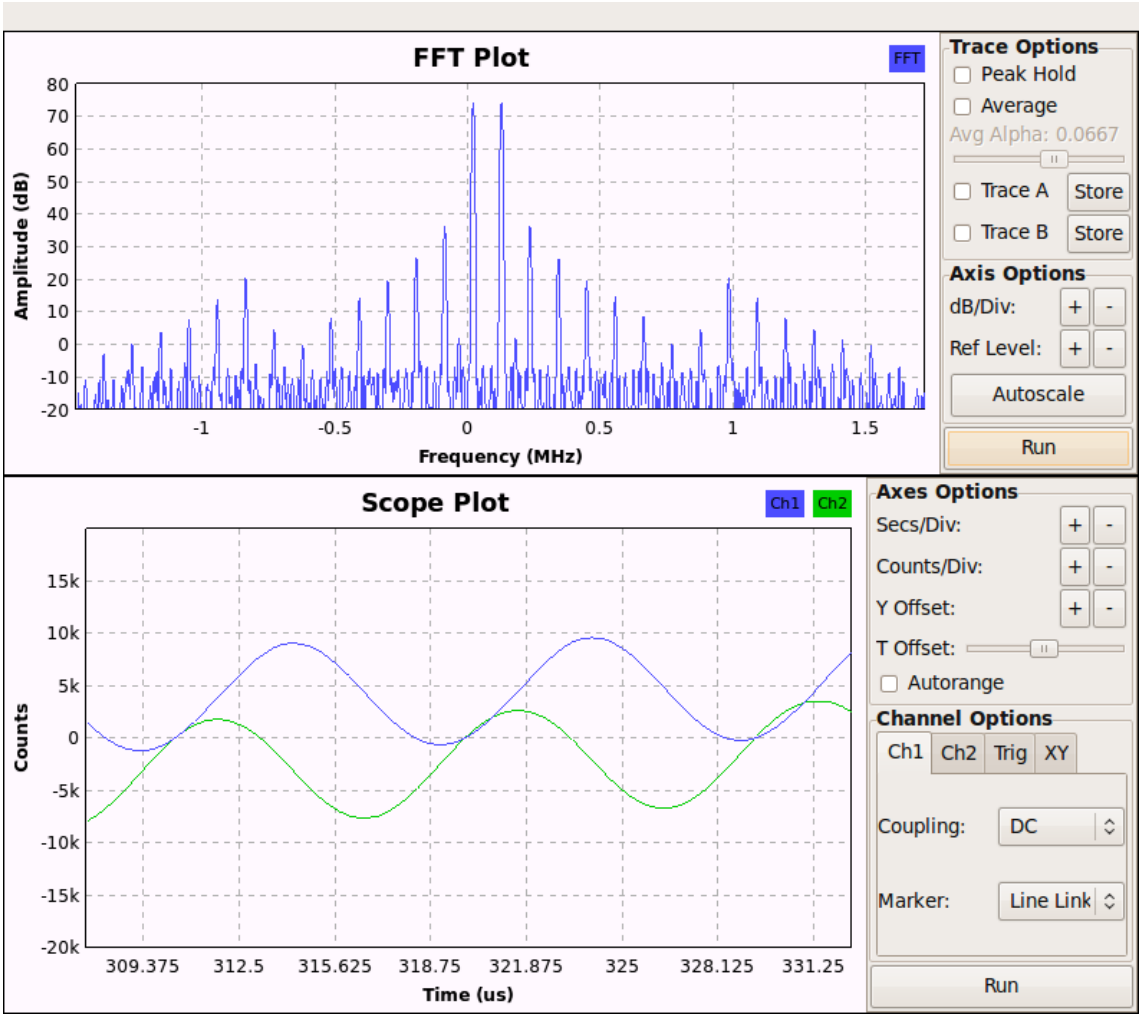

Figura 25: Leitor a 5cm

Com a aproximação da etiqueta e realização da leitura obtivemos oscilações de menor amplitude facilmente perceptíveis no FFT Sink. No domínio do tempo percebemos deformações na senóide que antes era perfeita, ao afastar a *tag* essas deformações diminuem, mostrando claramente que elas são um resultado da interferência do sinal de resposta da *tag*. Os resultados estão na Figura 26.

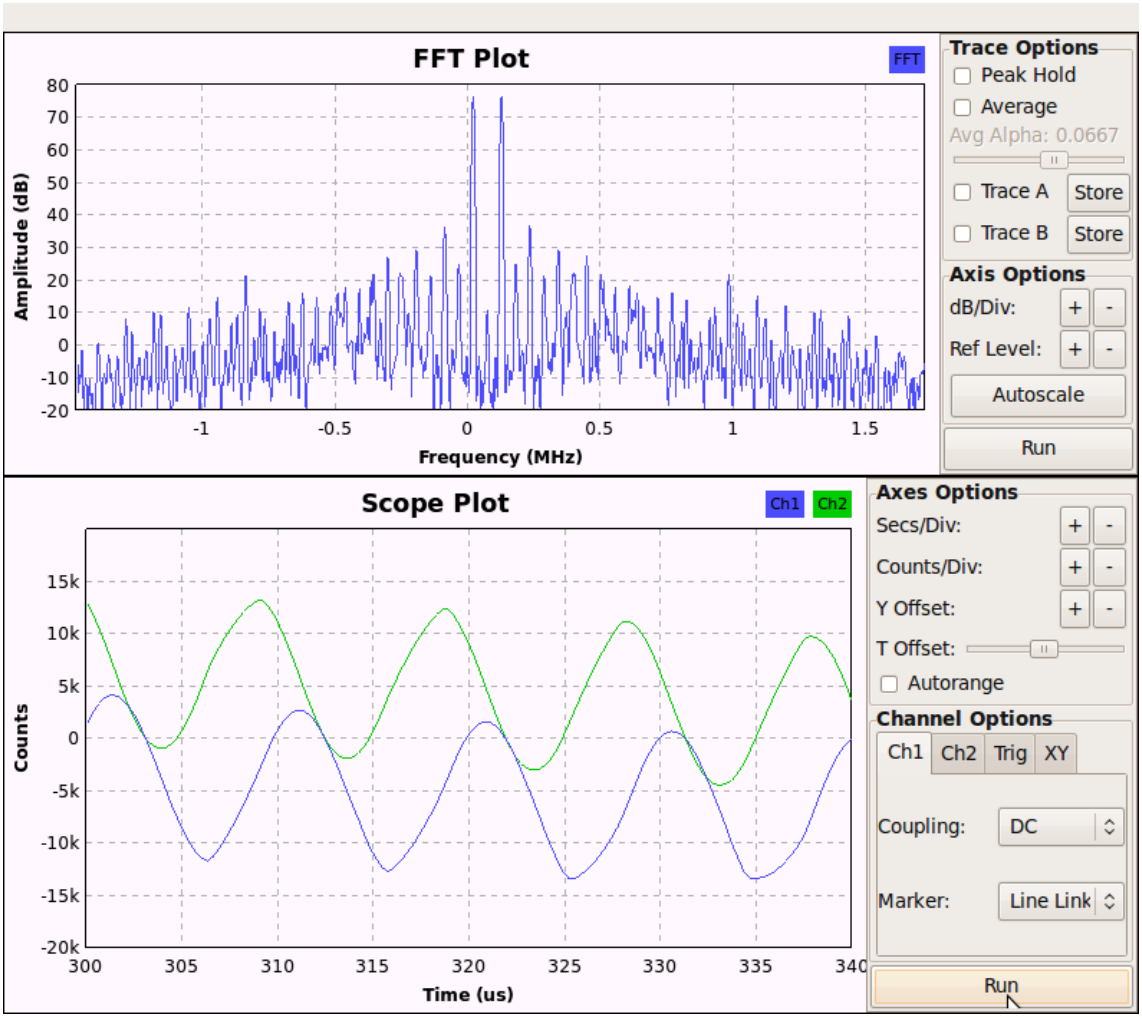

Figura 26: Leitor com tag sendo lida

5. Conclusão

A proposta inicial do estágio era fazer com que o GNU Radio se passasse por uma *tag*, emitindo um sinal válido para o leitor. Para chegar a esse objetivo devia-se passar pelos passos de colocar o kit para funcionar no sistema Linux e realizar a visualização do sinal. Ao final do estágio consegui-se chegar até a visualização do sinal.

Também não se conseguiu separar o sinal da *tag* do sinal do leitor, o que possibilitaria um melhor estudo do sinal. Mas muito da demora na execução do trabalho passou por escolher como seria feito o emulador de *tags*, o GNU Radio não foi a primeira escolha. Uma vez que o GNU Radio foi escolhido encontrou-se certa dificuldade para achar um bom ponto de partida para fazer o visualizador de sinal, porque a maioria dos programas exemplos e guias são para rádio comercial AM e FM.

Fica então a proposta para que outro aluno possa continuar um trabalho a partir desse, para que ele faça o emulador de *tag* desejado inicialmente, mas desta vez não mais partindo do zero. Mas que este trabalho sirva de primeiro passo.

6. Bibliografia

[1] RFID Toys

Amal Graafstra, Editora Wiley, 2006

[2] RFID Handbook: Fundamentals and Applications in Contactless Smart Cards and Identification

Klaus Finkenzeller, Editora Wiley, Segunda Edição, 2003

[3] RFID Security

Frank Thornton, Brad Haines, Anand M. Das, Hersh Bhargava, Anita Campbell, John Kleinschmidt

Editora SynGress, 2008

[4] **Read Only Contactless Identification Device**

Manual da EM Microeletronic

[5] **www.phidgets.com**

Site da empresa Phidgets, fabricante do kit, acessado em: 16/09/2009

[6] **www2.eletronica.org/projetos/leitor-rfid**

Site com guia de leitor RFID, acessado em: 09/09/2009