

Mecatrónica – 2013/2014

# ROS - Detector de metais

Projecto

Luís Rocha  
2010127532

José Medeiros  
2010129934

## Introdução

Neste trabalho foi incorporado um sensor indutivo ao robô Roomba para a detecção de minas terrestres através do uso de ROS. Para o controlo do robô foi necessário fazer uso de um API [2] que permitisse a comunicação com o mesmo e a plataforma ROS. Com este foi possível controlar os movimentos do robô através do teclado usando um terminal.

Quando na presença de metais e usando os dados devolvidos pelo sensor - proximidade e frequência - é possível determinar se o robô se encontra perante uma mina. Com base neste princípio é possível detectar a presença de minas.

Por os  
manu-  
ais do  
sensor



(a) Robot Roomba



Bobine LDC1000 Interface USB -  $\mu$ MSP430

(b) Componentes do LDC100EVM

Figura 1: Roomba e LDC100EVM com os componentes

## Todo list

■ Por os manuais do sensor . . . . .	1
■ Colocar melhor . . . . .	5
■ Melhorar e completar a lista de material!!! . . . . .	7
Figure: Falta a imagem do sensor soldado na PCB... . . . .	9
■ Explicar que usamos o teclado na arquitectura! . . . . .	9
■ Colocar a estrutura da bobine no robo - fotografia . . . . .	9
■ Colocar com imagens e explicações os passos em específicos . . . . .	11
■ Colocar o pseudo Código . . . . .	11
■ Colocar a simulação rviz . . . . .	13
■ Robo a andar com rviz activo . . . . .	13
■ Foto a controlar com o keyboard e a detectar minas . . . . .	13

# Índice

<b>1</b>	<b>Lista de acrónimos</b>	<b>5</b>
<b>2</b>	<b>Descrição do problema</b>	<b>6</b>
<b>3</b>	<b>Conhecimentos necessários</b>	<b>6</b>
<b>4</b>	<b>Material disponível</b>	<b>7</b>
<b>5</b>	<b>LDC100EVM</b>	<b>8</b>
<b>6</b>	<b>Solução proposta</b>	<b>9</b>
6.1	Hardware . . . . .	9
6.1.1	Arquitectura . . . . .	9
6.1.2	Implementação . . . . .	10
6.2	Software . . . . .	11
6.2.1	Pseudocódigo . . . . .	11
6.2.2	Utilização do ROS . . . . .	11
6.2.3	Comunicação com o sensor . . . . .	12
6.2.4	Protocolo VNC . . . . .	12
<b>7</b>	<b>Resultados</b>	<b>13</b>
<b>8</b>	<b>Conclusões</b>	<b>13</b>

## Índice de figuras

1	Roomba e LDC100EVM com os componentes . . . . .	1
2	Zona a tracejado . . . . .	8
3	Estrutura do robô Roomba com o sensor LDC100EVM . . . . .	9
4	Circuito do Sensor . . . . .	10
5	Circuito Simplificado . . . . .	10
6	LCD1000EVM com indicação das perfurações . . . . .	11
7	Protocolo VNC . . . . .	12

## 1 Lista de acrónimos

**ROS** Robot Operation System - Sistema Operativo para Robos

**LRC Meter** Aparelho que mede indutância, capacitância e resistência

**PCB** Printed Circuit Board - Placa de Circuito Impresso

**API** Application Programming Interface - Interface de Programação de Aplicações

**USB** Universal Serial Bus

**VNC** Virtual Network Computing

Colocar  
melhor

## 2 Descrição do problema

Pretende-se detectar minas terrestres através do uso de um robô Roomba e de um sensor indutivo LCD1000EVM capaz de medir a proximidade e a frequência associada à presença de um metal nas imediações. Este sensor é composto por uma bobine em PCB, um LDC1000 que faz uso da bobine e um microcontrolador - MSP430, tal como se pode observar na figura 1b.

Para ser possível a concretização deste projecto é necessário implementar uma interface de comunicação entre o computador e o sensor para a aquisição de dados. Esta interface vai ser baseada num protocolo de comunicação que terá que ser o mesmo que é usado pelo microcontrolador. Será portanto necessário entender bem o protocolo visto que toda a base deste projecto se baseia neste sensor.

Após a interligação entre o sensor e o computador através de uma porta USB é necessário combinar o robô Roomba com o sensor utilizando a plataforma ROS <sup>1</sup>. Para isso é obrigatório fazer uso de transformadas ( $tf's$ ) para referenciar o sensor ao eixo/referencial em relação ao referencial principal do robô. Mais á frente no relatório explicaremos melhor o conceito e a maneira como interligamos as transformações na nossa implementação Este passo é importantissimo para que se possa calcular as posições onde o sensor detecta minas.

## 3 Conhecimentos necessários

Após analisarmos os requisitos deste projecto chegámos á conclusão de que era necessário familiarizarmos com a plataforma ROS. Sendo assim seguimos os tutoriais existentes no website WikiRos [1]. Após concluidos os tutoriais, ficamos aptos a usar ROS usando C++ como linguagem de programação.

Os seguintes conceitos/utilidades foram usados no nosso projecto, a sua explicação breve apresenta-se abaixo:

### **Tf**

Estruturação dos referenciais do robô assim como as transformações necessárias entre os diferentes componentes <sup>2</sup> do robô para que se possa saber as posições de qualquer elemento num dado instante.

### **Marker**

Estrutura de dados que permite usar as bibliotecas do *rviz* para marcar minas no mapa. A posição das minas é obtida pela transformação da *tf* do sensor em relação ao referencial base do robô.

### **ArrayMarker**

Guardamos os markers (minas) que detectamos num array de markers. Para tal é necessário fazer *push\_back* de um marker para o array quando é detectada a presença de uma mina.

### **Serial Port**

Protocolo de comunicação usado para comunicar com o sensor através de uma porta USB.

---

<sup>1</sup>Distribuição Hydro

<sup>2</sup>Sensores, plataformas, rodas, entre outros

## 4 Material disponível

Melhorar e completar a lista de material!!!

**Robô e API** Roomba 555 com interface USB e *Wrapper* para comunicar com o robô

**Sensor de Metal** LDC100EVM

**Computador** Portatil de pequenas dimensões com o sistema operativo Ubuntu<sup>3</sup>

**Bobine** 1.5 mH

**Condensador** 19.2nF

**ROS** Plataforma ROS

---

<sup>3</sup>Ubuntu 13.04

## 5 LDC100EVM

O LDC100EVM é um sensor indutivo que permite medir a presença de metais retornando duas variáveis: posição do metal e a frequência de oscilação. É um circuito integrado que contém um microcontrolador MSP430 que é usado para fazer a interface entre o LDC (sensor) e um computador. Este integrado vêm com uma bobine que pode ser mudada se se partir a PCB na zona a tracejado e colocando ai uma bobine á nossa escolha. É possível ver na figura 2 a zona a tracejado:



Figura 2: Zona a tracejado

Para dimensionar a bobine é necessário satisfazer duas condições para que o sensor possa funcionar na gama ideal:

1. A frequência de ressonância deve estar compreendida entre 5 kHz e 5 MHz
2. A resistência de ressonância<sup>4</sup> deve estar compreendida entre 798Ω e 3.93 MΩ

Para o cálculo da frequência e a respectiva resistência usa-se as seguintes fórmulas:

$$f_{\text{ressonância}} = \frac{1}{2\pi\sqrt{LC}} \quad (1)$$

$$R_p = \frac{1}{R_s} \frac{L}{C} \quad (2)$$

Tendo em conta (1) e (2) podemos dimensionar [4] uma nova bobine para o sensor. Para garantir que a bobine juntamente com o novo condensador estão em contacto com o sensor deve-se soldar um adaptador na placa que permite a fácil colocação de elementos através de uma chave de fendas.

Para calcular a indutância [3] medida pelo sensor é necessário fazer uso da frequência medida pelo LDC100EVM. Para tal:

$$L_{\text{medida}} = \frac{1}{C * (2\pi f_{\text{sensor}})^2} \quad (3)$$

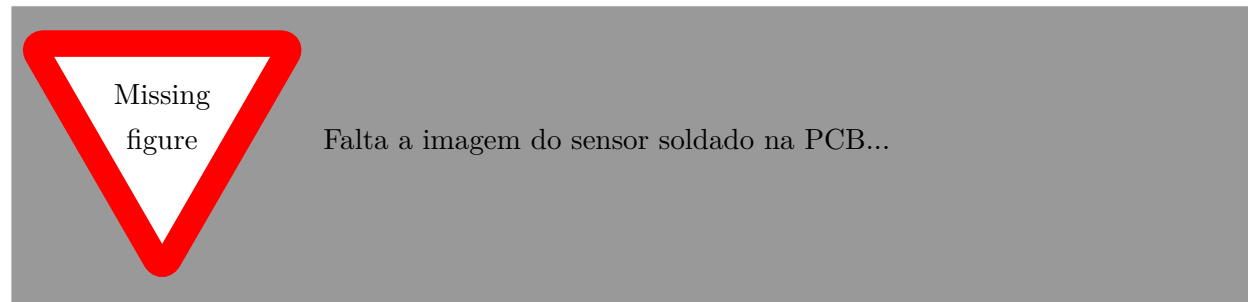
---

<sup>4</sup>Simboliza as perdas associadas às correntes de Eddy



## 6 Solução proposta

### 6.1 Hardware



Explicar que usamos o teclado na arquitectura!

Como podemos ver na figura, soldamos os componentes numa placa PCB de modo a facilitar a mudança de bobines e para garantir que o circuito está em contacto com os componentes.

Em termos de ligações é mais facil de compreender na figura abaixo, onde é possível ver a ligação dos componentes ao Roomba e ao sensor.

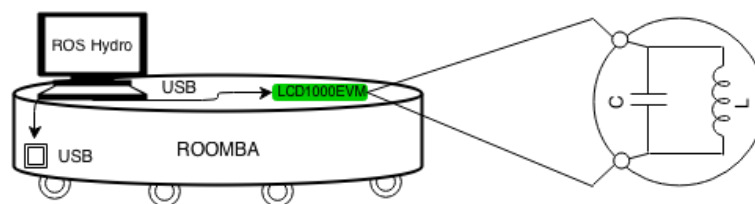


Figura 3: Estrutura do robô Roomba com o sensor LDC100EVM

#### 6.1.1 Arquitectura

Em termos de hardware para este trabalho foi utilizado um computador que vai controlar o Roomba e o sensor LDC1000EVM. No sensor foi ainda substituída a bobine que vinha com o mesmo por uma bobine e um condensador de valores por nós pretendidos.

O robô Roomba encontra-se conectado por usb a um computador que o controla. Este computador encontra-se ainda ligado a um sensor LDC1000EVM que por sua vez tem ligado a si uma bobine e um condensador por nós projectados e adaptados.

Colocar a estrutura da bobine no robo - fotografia

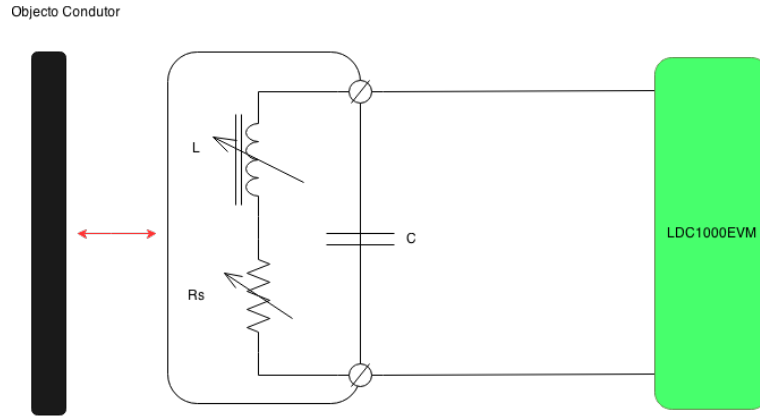


Figura 4: Circuito do Sensor

Por intermédio de (2) é possível aproximar o circuito da seguinte forma:

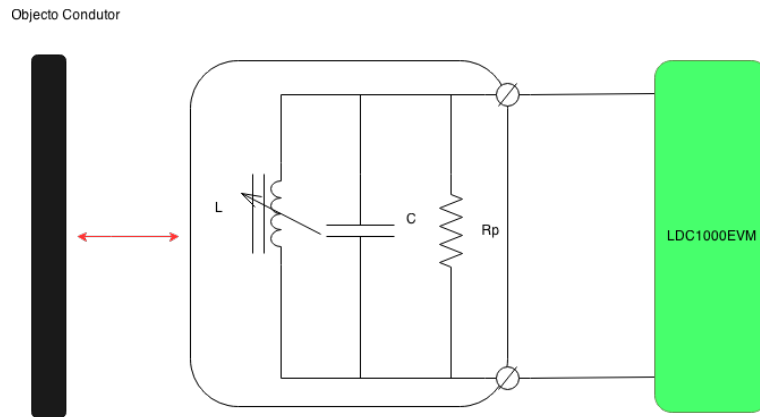


Figura 5: Circuito Simplificado

### 6.1.2 Implementação

Como foi anteriormente explicado, o circuito pode ser alterado e dimensionado para se trabalhar na gama de valores desejada. A seguir iremos explicar os passos necessários para implementar o novo circuito e adaptá-lo ao sensor.

1. O primeiro passo será escolher simplesmente a nova bobine por nós pretendida, de seguida escolhemos um condensador que permita cumprir as condições impostas à frequência de ressonância(1) e à resistência de ressonância(2).
2. O próximo passo será com o auxílio de um “LCR meter”, colocando as pontas deste medidor em paralelo com o paralelo do condensador e da bobine que escolhemos temos que anotar o valor da resistência  $R_p$  lido no medidor, na ausência de qualquer objecto magnético.
3. Neste passo temos que colocar um objecto magnético o mais próximo possível da bobine e anotar o novo valor de  $R_p$ .
4. É necessário anotar também a frequência de varrimento que se encontra definida no medidor.

5. O passo seguinte é actualizar os registos do sensor para que se adapte ao novo circuito. Para modificar os registos é necessário conectar o sensor a um computador coondutoresm windows e utilizando a gui application fornecida pela texas instruments na secção de alteração dos registos terêmos que modificar três registos. O valor de  $R_{pMAX}$  deve ser alterado para o dobro do valor anotado na ausência de objectos c, como os valores disponíveis são discretos deve ser escolhido o mais próximo deste valor por excesso. O valor de  $R_{pMIN}$  deve ser alterado para metade do valor anotado na presença de um objecto condutor, como os valores disponíveis são discretos deve ser escolhido o mais próximo deste valor por defeito. O registo da frequência deve ser alterado para no máximo  $\frac{8}{10}$  do valor da frequência de varrimento.
6. Por fim é necessário partir o sensor onde diz “coil perforation” na figura 6 e soldar o paralelo da bobine e do condensador no sítio onde diz “conections for custom LC tank” na figura 6.

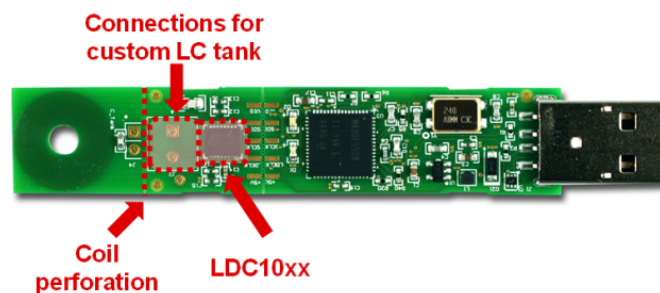


Figura 6: LDC1000EVM com indicação das perfurações

Colocar com imagens e explicações os passos em específicos

## 6.2 Software

Colocar o pseudo Código

### 6.2.1 Pseudocódigo

### 6.2.2 Utilização do ROS

Utilizámos o Robot Operating System (ROS) como plataforma de comunicação entre os dois periféricos por nós utilizados e o computador. Com o ROS é possível implementar bibliotecas de cliente ROS como roscpp, rospy ou roslisp. Por outras palavras significa que facilmente se consegue adaptar programas criados em outras línguas (de entre C++, Python e LISP). No nosso caso em específico adaptámos o programa de leitura do sensor em cpp para a plataforma ROS. Numa segunda parte através do api disponibilizado pelo Gonçalo Cabrita e pelo Bruno Gouveia conseguimos controlar o Roomba pelo teclado do computador. Através da leitura dos encoders e do controlo dos motores foi possível controlar a velocidade angular e linear do robô.

Utilizando os nós do ROS foi possível assim controlar o robô e utilizá-lo para a detecção e marcação nas minas. De uma forma mais detalhada foi usada como referência a transformada "ODOM" que nos fornece a posição inicial em que o robô se encontra, com o nó entre esta transformada e o

base link do Roomba sabe-se sempre a posição relativa à posição inicial. De seguida criando o nó da transformada sensor(criada por nós) é assim possível, com a informação devolvida pelo sensor, saber se o robô se encontra na presença de uma mina. Caso seja verificada a presença de uma mina são enviadas as coordenadas (x,y,z) para uma função "markers" que irá marcar no mapa as minas em relação a "ODOM", ou seja em relação à posição inicial do robô.

### 6.2.3 Comunicação com o sensor

A primeira parte deste trabalho foi concluir que o sensor utiliza porta série como protocolo de comunicação quando comunica com o computador. Como tal, para facilitar o uso futuro deste sensor criámos código para as leituras do sensor em três plataformas diferentes de programação. Adaptámos por isso o nosso código às plataformas [C++](#), [Matlab](#) e [Python](#). Estes três códigos encontram-se mais a baixo na tabela(Fazer referencia à tabela).

Analizando as partes mais importantes do código, o primeiro passo é definir a porta série pela qual estamos a comunicar, no nosso caso em particular foi a porta "COM9" em *Windows* e `"/dev/ttyACM0"` em *Linux*. De seguida definimos o tipo de controlo de fluxo, que neste caso é do tipo "software flow control". É necessário definir a frequência de comunicação, Baud Rate, que foi definida por nós a 9600. O próximo passo é definir a paridade e o número de bits da comunicação, neste caso não existe paridade e são utilizados 8bits na comunicação. Tal como na maioria dos dispositivos electrónicos na comunicação por porta série apenas é necessário de 1 "stop bit", foi ainda definido o tempo de espera para leitura máximo("Timeout") a 5 segundos. Definimos o modo de leitura contínua para funcionar de forma assíncrona. Por fim ao escrever na porta série o comando 0x33(hexadecimal) ou 3(ASCII), iniciamos a stream de dados.

### 6.2.4 Protocolo VNC

Este protocolo baseia-se no conceito de partilha remota de um computador. Para isso existe um computador que partilha o monitor e, se o utilizador permitir, os controlos do mesmo (servidor) e um ou mais computadores (clientes) que se conectam ao servidor e são capazes de realizar as acções permitidas pelo mesmo .

Como a base do VNC foi construída/projectada sobre o protocolo TCP/IP, este é capaz de funcionar via internet. É usado em aulas interactivas onde um professor permite a visualização do seu monitor e os alunos conectam-se podendo assim visualizar os passos do professor.

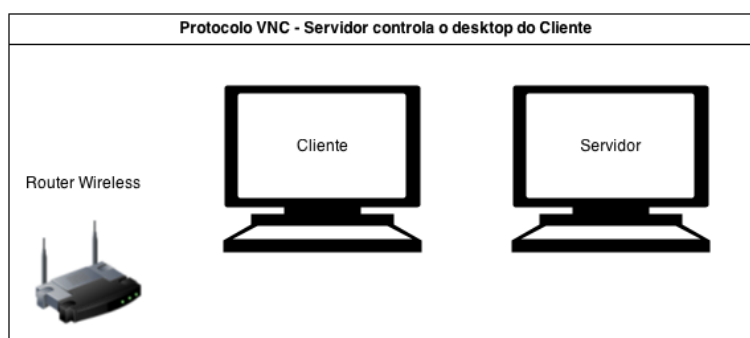


Figura 7: Protocolo VNC

**Nota:** Rápido e eficaz numa rede local com boa largura de banda.

## 7 Resultados

Colocar a simulação rviz

Robo a andar com rviz activo

Foto a controlar com o keyboard e a detectar minas

## 8 Conclusões

Com este projecto foi possível adaptar o robô ROOMBA para a detecção de minas com o auxílio do sensor LCD1000EVM. Numa primeira parte do projecto foi criada a interface para que o sensor pudesse ser utilizado em Linux. Com o auxílio da plataforma ROS (Robot Operating System) foi possível fazer a comunicação dos três periféricos ( computador, robô e sensor) esta plataforma permitiu ainda criar o controlo por teclado da velocidade e direcção do robô. Por fim foi possível adaptar o sensor para uma bobine por nós desejada.

Este projecto permitiu perceber a potencialidade tanto da plataforma ROS como do sensor LCD1000EVM. A grande vantagem da plataforma ROS é que no futuro muito facilmente qualquer pessoa poderá adaptar este sensor a outros robôs ou outros sistemas robóticos, é a vantagem do sistema de transformadas em que o ROS trabalha. O sensor LCD1000EVM tem a grande vantagem de com alguns pequenos passos se poder modificar para qualquer bobine que o utilizador pretenda criando-se assim um sensor à medida do que o utilizador precisa.

## Referências

- [1] Robot operating system. <http://wiki.ros.org/>, 2007. Hydro Edition.
- [2] Gonçalo Cabrita. Roomba api, 2013. <http://wiki.ros.org/Robots/Roomba>.
- [3] Texas Instruments. *Datasheet LDC1000*, 2013. <http://www.ti.com/lit/ds/symlink/ldc1000.pdf>.
- [4] Texas Instruments. *Quick Start Guide*, 2013. <http://www.ti.com/lit/ml/slyw022/slyw022.pdf>.