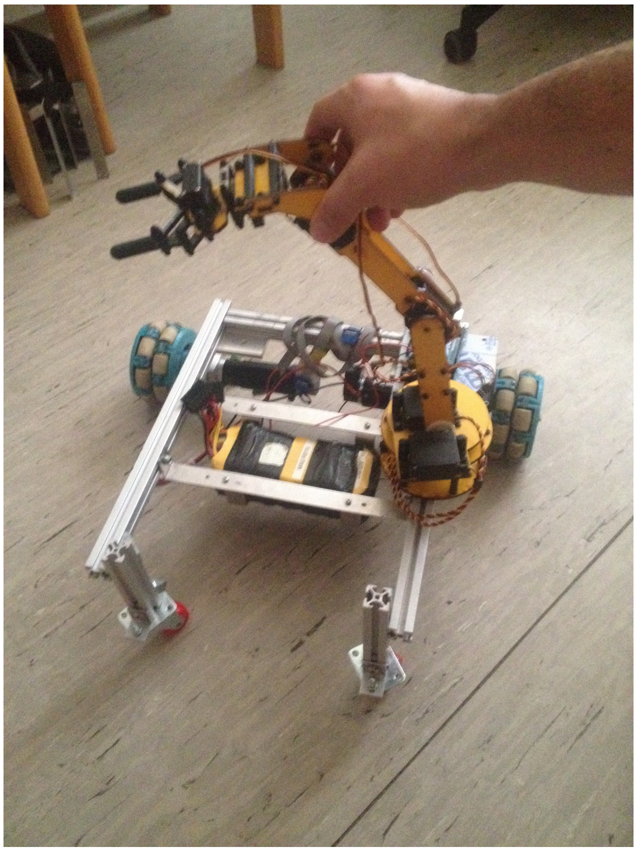
Little Bas documentation

Little bas : a open source robot

by Oscar Lima*



University of Applied Sciences Bonn-Rhein-Sieg

Master in Autonomous Systems

16.12.2014

*olima_84@yahoo.com

Abstract

Little Bas is a fully open source robotics project regarding the mechanics, electronics and software. It its well documented and currently capable of the following skills:

- autonomous navigation (ros navigation)
- face recognition (opencv)
- object recognition (bounding box approach)
- manipulating small objects

Little Bas is perfect for students who want to introduce themselves with robotics and also as a research platform.

CONTENTS

About the Author

Oscar Lima Carrión was born in Mexico city in 1984, he studied in UNAM national Mexico university Mechatronics engineering and received his bachelor diploma in 2008. Currently he is studying in Bonn Rhein Sieg University of applied sciences near Bonn, Germany, the master in autonomous systems program, which is robotics oriented.

He is proudly team member of the b-it-bots university robotics team which participates mainly in three competitions per year: German open[1], Robocup[2] and RoCKIN[3] challenge.

Also he is member of the EmoRobot[4] project as student job.

He has great passion about robotics and open source robotics code.

[1] http://www.robocupgermanopen.de/en/front

[2] http://www.robocup.org

[3] http://rockinrobotchallenge.eu

[4] http://emorobot.inf.h-brs.de

1. INTRODUCTION

Little bas is the name of the open source robot that this report describes. It has the intention of self education and also the promotion and understanding of robotics. This project will start from the very basics of what is a H bridge and will escalate in complexity until we reach how to autonomously navigate. The intention is low cost and detailed construction instructions and material purchase lists are available.

2. Related Work

Subtitle

Some text here

3. Background

Subtitle

Some text here

4. Motor Controller

H Bridge with mosfet

We start with a plain motor with encoder and two wires for power. The goal of this subsections is to provide with a strong electronic circuit which is capable of driving the motor in both directions, stop and also capable of controlling speed and position for the arm and base.

The motor which are currently having at hand is a maxon motor 248853 with a mason gear number 166932 made in germany. Further motor specs are available at the appendix.

The selected micro controller is an arduino uno with a motor shield.

This selection is nice because of its flexibility, not only you can easily connect with it and make it work very fast but you can easily adapt it for different use if needed.

For example, you can use the same motor shield to control:

- two reles
- two dc motors
- 1 stepper motor

It has also tinker kit compatibility which allows to easily connect a sensor to the board by molex pin interface, a feature that we might use in the future for adding a gyroscope and create a balancing robot.

We will now build and understand the motor controller that will control our robot.

Exercises with a DC motor controller

In this section we will start with a lab exercise, this will have the following objectives:

1. Create, design and test a simple open loop motor controller, wire the encoder of the motor for external ground truth observation but this will not really be passed as feedback.
2. Design a velocity interface for the motor by using a P, PD and PID controller
3. Create a torque controller for the motor, this should be capable of setting the desired current you want the motor to consume and the control loop should adapt to this desired input

4. Create a servomotor, this should be able to send the motor to a certain angle and remain there even if external forces are applied to the motor
5. Add compliance to the previous servo motor controller, even if a human hand holds the shaft of the motor, the motor should stop or not accept more current than a certain level and then slowly the motor has to go to the commanded position
6. Create a acceleration controller, this should be able to receive acceleration values which are then executed by the motor

Open loop motor control

The goal is to set a velocity, send this velocity to the controller and by making a model of the motor be able to predict the needed voltage to drive the motor in open loop to a certain speed. This will be later on verified and compared with ground truth data coming from the enconder.

For modeling the motor we will use information from the following matlab website:

http://ctms.engin.umich.edu/CTMS/index.php?example=MotorSpeed&section=SystemModeling

Reading the encoder values

A full example (code) is available under this link:

http://playground.arduino.cc/Main/RotaryEncoders#Example1

more examples in general for rotary encoders are available under:

http://playground.arduino.cc/Main/RotaryEncoders

For the hardware part we require first to understand how the encoder is constructed and the main logic behind it.

5. Experiment Results and Analysis

Subtitle

Some text here

6. CONCLUSIONS

Title

Some text here

7. FUTURE WORK

Title

Some text here

APPENDIX

A. Base motors specifications

The base of our robot is having two motors in the base (differential drive) and as mentioned before is a maxon motor 248853 with a mason gear number 166932 made in germany. Specifications were taken from[5], there are two components: motor and gearbox.

Motor specifications

Gearbox specifications

It has a weight of 110 g, reduction of 5.8:1 (23/4) a maximum continuos torque of 1 Nm with peaks of 1.25 Nm a efficiency of 80%, backlash of 0.7°, inertia of 1.5 gcm2, maximum status radial load of 140 N (10 mm from flange) while having a dynamic max. radial load of 120 N, maximum continuos speed of 8000 rpm.

Full datasheet available on document with name gearbox_specs.pdf that should be available and distributed among with this report.

Encoder specifications

The encoder HEDS-5540 is used, you can consult the data sheet in the attached files. It is an absolute encoder of 500 ticks per revolution. The reader is encouraged to read the following wikipedia link to get familiar with absolute encoders:

http://en.wikipedia.org/wiki/Rotary_encoder#Incremental_rotary_encoder

The encoder has 3 signals: A, B and a third signal which tells you when a full rotation has been made to the encoder. The signals A and B are there shifted by 90 degrees to provide with information about the spinning orientation (clockwise or counterclockwise) and they provide with a binary count from 1 to 4 and backwards depending on the spinning orientation.

Motor assembly price

The motor assembly part number is 248853, this is a package that includes the motor, encoder and gearbox, as follows:

[5] http://www.maxonmotor.com/maxon/view/product/gear/planetary/gp32/166932

Artikel Nr.: 248853
maxon Motor-Getriebe-Encoder-Kombination Nr. 248853 bestehend aus:
118777 DC motor RE35 GB 90W KL 2WE
166932 gear GP32C 1.0NM 1ST KL i=5.8 : 1, Katalog 14/15 S.276
110513 tacho ENC HEDS 5540 500IMP 3K WE Ø4, Katalog 14/15 S.325

part	price in euros	price in CHF (swiss franc)	
DC motor	248.68	299	
Encoder	71.53	86	
planetary gearbox	116.6	140.2	
total	436.81		

Keep in mind that both product numbers 248853 and 118777 are no longer available, but equivalent 273754 (just the motor) is still available, so the presented price corresponds to this last presented equivalent motor. Encoder and gearbox are still having the same part number and prices are presented as such.

REFERENCES

7. Mosfet H bridge : http://www.cadvision.com/blanchas/hexfet/index.html

8. e