Grabo documentation

# Version

V1: 8 October 2013 – Creation by Hugo Van hamme

# Warning

Grabo uses lithium-polymer batteries. Deep discharge of these batteries will DESTROY them. The voltage should not drop under 3.2 Volt per cell (6.4 V in total). The motors are disabled at 3.6 Volt per cell, the current drain can still destroy the battery.

DISCONNECT THE BATTERY WHEN NOT IN USE.

Regularly check battery voltage.

# Description

Grabo is a 2-wheel robot platform with the following:

* Central controller is an Arduino UNO.
* Bluetooth module (JY-MCU also called HC-06) for communication with a master computer via text commands.
* Each wheel (of 2) is equipped with a dc motor and a wheel encoder, implemented on an Arduino NANO. The NANO communicates with the UNO over I2C.
* A two-finger claw. Each finger driven by a separate servo. The current draw by the servo’s is measured to allow adjusting the gripping force.
* The claw is mounted on a lift (controlled by a servo).
* A laser pointer is also mounted on the lift.
* A 6-axis gyroscope (MPU6050) is used for driving on a straight line. It communicates with the Arduino NANO over I2C.
* A ultrasonic distance (from nose to nearest object ahead) measuring device.

## Further hardware information

The wheel encoder, lift, claw and motors require the battery as power source. The NANO, UNO, Bluetooth, echo sounder and gyro can be powered form the USB as well.

The current drawn by the servo’s is measured as follows: they are fed by a step-down voltage converter (KISS-33R modules). There is a small resistor between the converter output and the servo positive power supply, while the step-down feedback voltage is measured at the servo positive power supply, hence providing a stable supply voltage for the servos. The voltage across the resistor is amplified by an opamp configured as difference amplifier. This signal is fed to an analog input on the UNO.

The two wheel motors are controlled by an H-bridge. Each H-bridge is connected to PWM signal on one leg and a digital “direction” signal on the other leg.

# Interfacing with Grabo

## Terminal

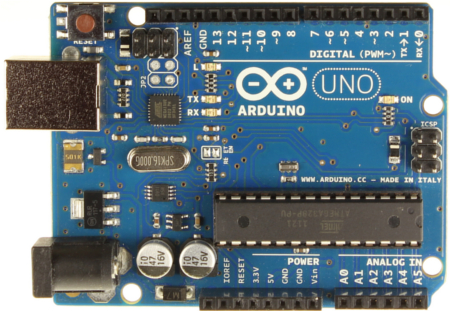
I have tried different “terminals” to issue commands to Grabo

* The Arduino serial monitor. Be sure to select “newline” to terminate keyboard input.
* Realterm. Under the “send” tab, select “EOL +LF”
* The matlab serial device

In all cases, set the baud rate to 115200 and choose the com port associated with the serial connection (on W7, see “device manager” => “Ports (COM & LPT)”.

## Hardware

Grabo can be connected via a serial line at 115200 baud, no parity, 8 bits, 1 stop bit. You can either connect the UNO via a USB cable (remove the connector on digital pins 0 and 1 (TX/RX) ) to a PC.



If you are using the Bluetooth connection, you have to connect the connector with grey wires on digital pins 0 and 1 (TX/RX). Establishing a connection over Bluetooth is a tedious and delicate process. I only have experience on Windows 7.

* Under “Devices and Printer”, select device “Linvor”
* Right-click on Linvor and go to Properties, then services
* Serial port (SPP) + the COM<N> port should be listed. If it’s not, reset the Bluetooth module on Grabo (at this moment power-off + power-on). The service should appear after some time (may require a refresh F5). (<N> is an actual number, 12 as I am trying it now)
* Under Device Manager, you should see COM<N>. If not, uncheck the service in the Linvor properties and recheck. The COM port should install automatically.
* Now connect to the Bluetooth module by starting the Arduino serial monitor on the right COM-port, pressing “change” in Realterm (tab “port”) or opening the serial connection in Matlab. The red led on the Bluetooth module should stop flashing and light up continuously.

## Commands

Default argument is 0.

"p <int>" switch pointer on/off - <int>==0 => off ; <int>!=0 => on

"s" echo status

"a <float>" set reference angle to current position + <float> (in degrees)

"x <int>" set reference wheel position to <int> (in encoder ticks; does not move anything)

"t <int>" set target position to <int> (in encoder ticks; does not cause movement)

"c <int> drive straight ahead at pulsed speed -16384 <= <int> +16383 (negative is backward; 5000 is a good value

"r <int> drive straight ahead at speed 0 <= <int> <= +16383 (negative is backward) till target from t command

"e <int> drive straight ahead at speed 0 <= <int> <= +16383 (negative is backward) till close to object (echo sounding)

"l <int>" lift claw to position 0 <= <int> <= 100

"q <int> close claw with force 1 <= <int> <= 1000; 0 is open

# The Arduino NANO

The NANO performs time-critical functions:

* Wheel encoding (left and right)
* Processing the gyro output

The NANO is connected to the UNO and the gyro via the I2C bus (pins A4 and A5). Because the GYRO is always a slave device, I made the UNO also a slave. The NANO will compute a rotation about the vertical axis (as a float between -180 and +180) and increments of the wheel encoder and send these to the UNO in its main loop. This will generate an interrupt on the UNO which then simply reads the new angle and wheel position increments.

# Software

Components:

## Grabo\_client

Code for UNO. Is client to master computer.

Measures battery voltage. Will disable motors if voltage too low.

## WheelGrabo

Code for NANO.

Wheel encoders are polled.

## Libraries

### Claw

Two-finger claw + lift driver. Used on UNO.

Uses 3 PWM outputs on the UNO, one for each servo.

### EchoSounder

Ultrasound distance measuring. Used on UNO.

Uses an interrupt.

### I2Cdev

Used by gyro code. Copied from Jeff Rowberg. Used on NANO.

### MPU6050

Gyro code copied from Jeff Rowberg. Used on NANO.

I2C bus + interrupt used.

### TwoWD

Motor driver, used on UNO.

Uses 2 PWM ouptuts of UNO. Uses the angle measurement to control the motors so Grabo drives in a straight line (using the wheel encoders does not have enough resolution).