Add trim to the motors

If you find that your GoPiGo starts turning a bit when it moves forward then this is something for you.

We have added a trim function in the Firmware and the software from firmware v1.3 on wards so that you can increase or decrease the speed of one of the wheels just by a bit (add trim) so it does not turn by a lot when it is moving forward. The GoPiGo usually starts going a bit left or right because of a lot of external factors like the wheel alignment, motor placement, friction etc.

Here is how to use trim with your GoPiGo.

Before you start, **make sure that you disconnect the motors**. When the firmware update is done and you have verified that it has upgraded, connect the motors again.

Make sure that you have updated to firmware v1.3.

```
cd Desktop/GoPiGo/Firmware sudo chmod +x firmware_update.sh
```

```
sudo ./firmware_update.sh
```

```
pi@dex: ~/Desktop/GoPiGo/Firmware
pi@dex ~ $ cd Desktop/GoPiGo/Firmware/
pi@dex ~/Desktop/GoPiGo/Firmware $ sudo chmod +x firmware_update.sh
pi@dex ~/Desktop/GoPiGo/Firmware $ sudo ./firmware_update.sh
ATTENTION! Important!
BEFORE PROGRAMMING THE GOPIGO FIRMWARE, DISCONNECT THE MOTORS.
Please confirm that you've disconnected the motors.
Have you disconnected the motors before programming the firmware? (y/n)
y
Updating the GoPiGo firmware
============================
Tue Dec 29 06:25:02 UTC 2015

avrdude: AVR device initialized and ready to accept instructions

Reading | ################################################## | 100% 0.00s

avrdude: Device signature = 0x1e950f
avrdude: reading input file "0x7F"
avrdude: writing lfuse (1 bytes):

Writing | ################################################## | 100% 0.00s

avrdude: 1 bytes of lfuse written
avrdude: verifying lfuse memory against 0x7F:
avrdude: load data lfuse data from input file 0x7F:
avrdude: input file 0x7F contains 1 bytes
avrdude: reading on-chip lfuse data:
```

and also update the gopigo.py library

```
cd Desktop/GoPiGo/Software/Python/ sudo python setup.py install
```

```
pi@raspberrypi ~ $ cd Desktop/GoPiGo/Software/Python/
pi@raspberrypi ~/Desktop/GoPiGo/Software/Python $ sudo python setup.py install
running install
```

Now run the **basic_test_all.py**

```
sudo python basic_test_all.py
```

```
pi@raspberrypi ~/Desktop/GoPiGo/Software/Python $ sudo python basic_test_all.py

Cmd:
```

Check the firmware version first:

```
Cmd: f
v 1.3
```

Then check for the trim value with the **tr** command:

Now you should try various trim values to see what works for you.

```
Cmd: tr
-3, Trim Value Not set
```

- A trim value of **0** means that the left motor will run at the same speed as the right motor.
- A trim value of **50** means that the left motor will run at the **150%** the speed of the right motor and the GoPiGo will turn more towards the right. This can be used to compensate for the GoPiGo normally turning left when it is supposed to go right.
- A trim value of **-50** means that the left motor will run at the **50%** the speed of the right motor and the GoPiGo will turn more towards the left. This can be used to compensate for the GoPiGo normally turning right when it is supposed to go left.
- Start with small trim values in **multiple of 5** and find out the value that works for you
- Give **w** to move forward and **x** to stop to test the trim values

Once you have found the trim value that works best for you, write it to the EEPROM so that the GoPiGo uses them whenever it starts:

```
Cmd: tt
Enter trim value to test(-100 to 100): 10
Value in EEPROM:  0

Cmd: w

Cmd: x
```

Give **tw** as the command and enter the desired trim value. If it reports back that the value in EEPROM is the same as the value you gave, that means that you have set the values correctly.

```
Cmd: tw
Enter trim value to write to EEPROM(-100 to 100): 10
Value in EEPROM:  10
```

The values revert back to 0 whenever yo do a firmware update.