

# Universal Gripper Controller Design with Visual Feedback

Ahmet Safa Ozturk

January 15, 2016

BIOROB EPFL

Supervisor: Mehmet Mutlu

Professor: Auke Ijspeert

# Outline

- Motivation
- Steps
- Universal Gripper
- Spotlight
- Android Interface
- Conclusion

# Motivation

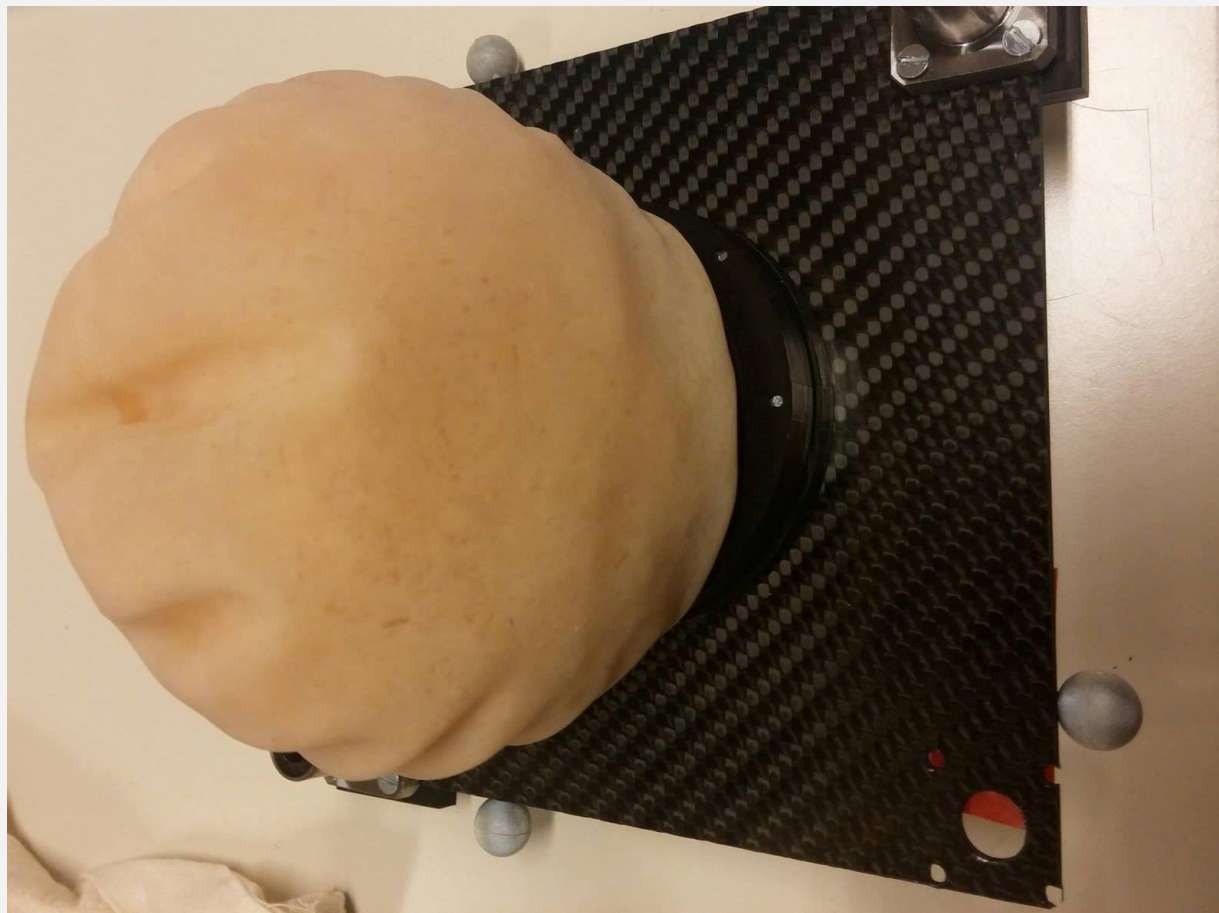
Building and testing two new Roombots compatible boards for controlling the Spotlight and the Gripper.



# Steps

- Board Designs
- Firmware
- Tests

# Universal Gripper



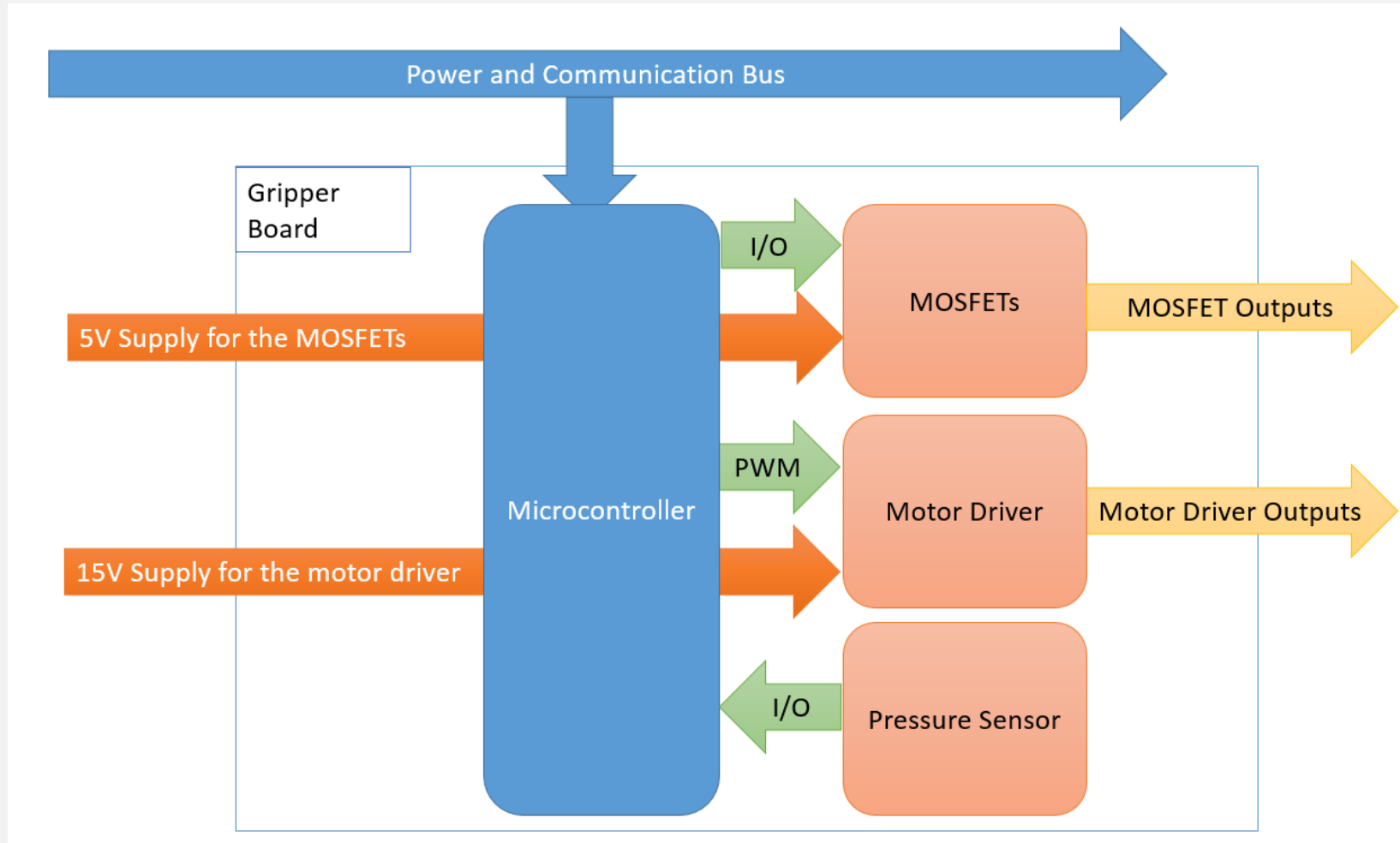
# Universal Gripper



# Universal Gripper

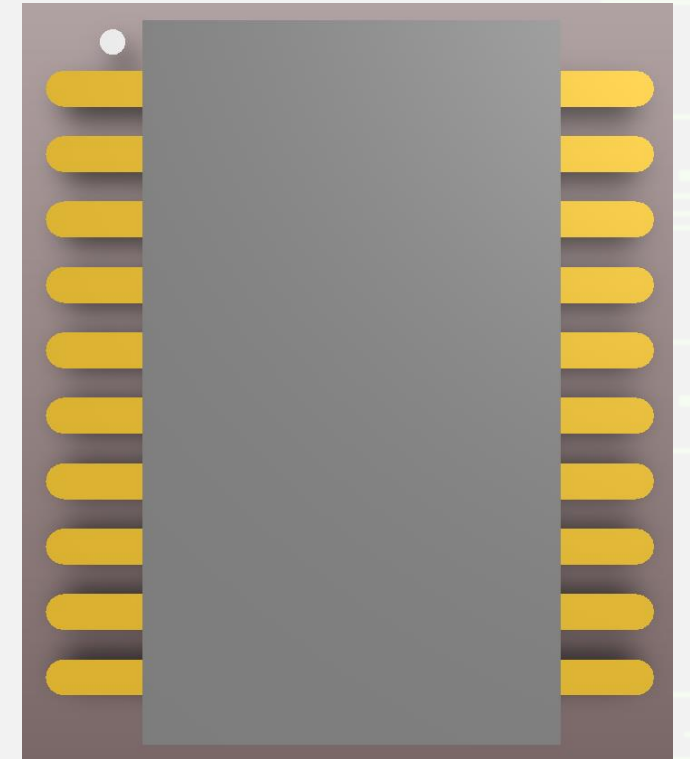
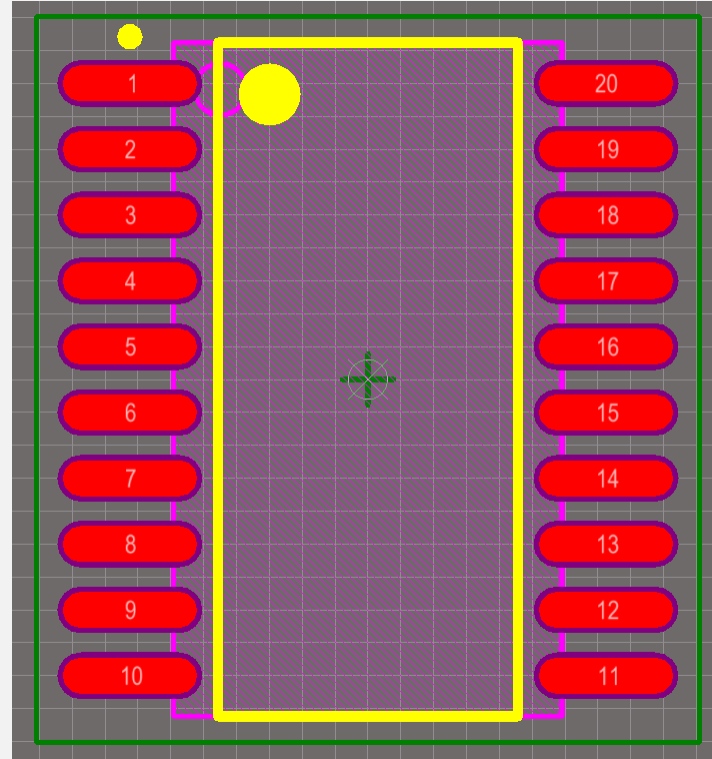
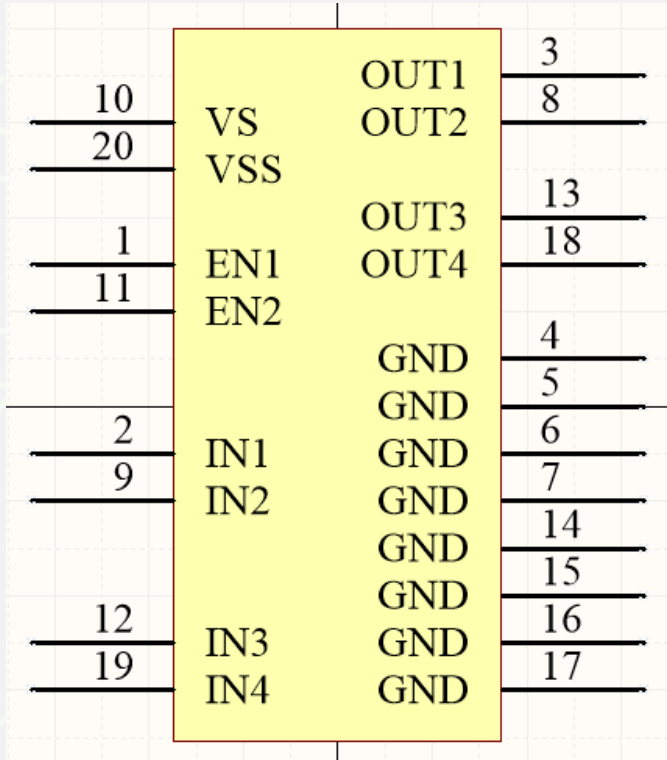


# Universal Gripper

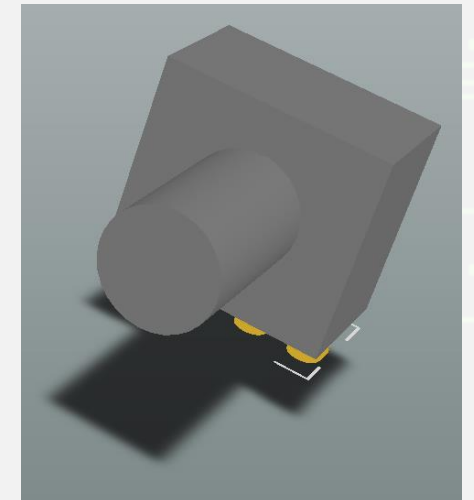
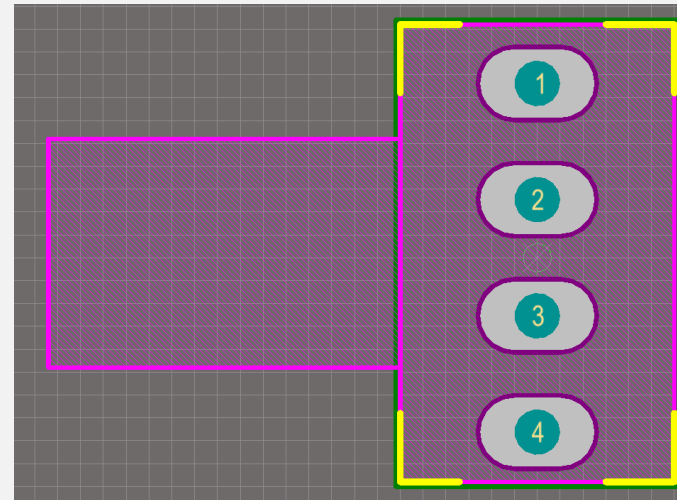
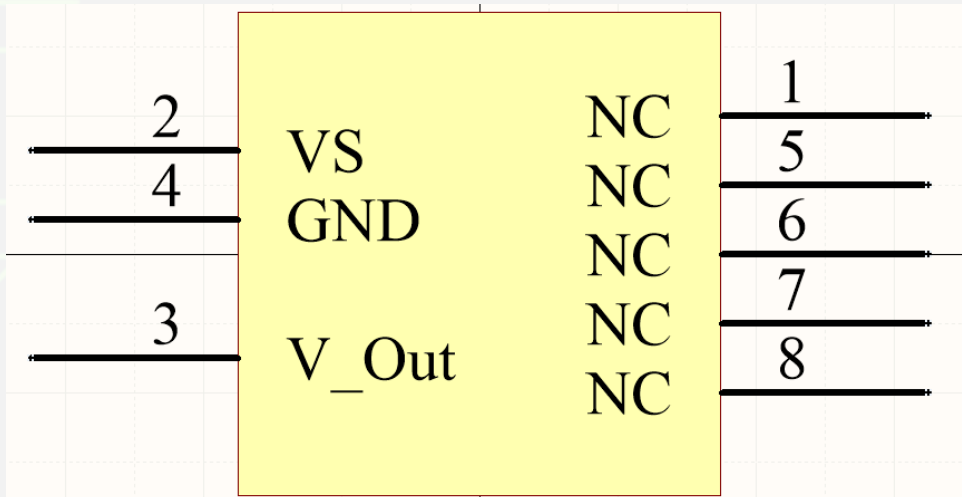




# Motor Driver



# Pressure Sensor

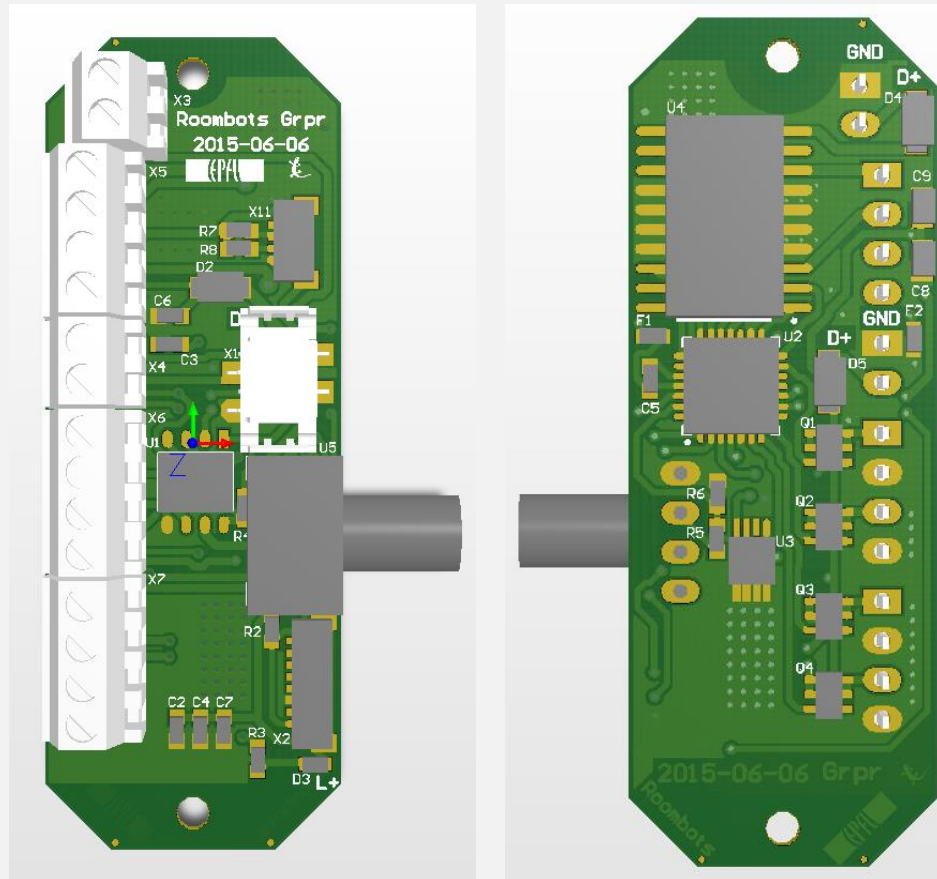


The schematic diagram illustrates the electrical connections for the Gripper Control Board. It features several key components:

- Microcontroller:** Micro-Match 7-338069-4, connected to VBUSL and GND.
- Transceiver:** R5485A, RS485 Transceiver Half-Duplex 16Mbps, connected to VDD, GND, and signal lines.
- Sensors:** Pressure Sensor (U5) and four RTQ040P02TR temperature sensors (Q1-Q4).
- Connectors:** X1 (Power bus low voltage: 6V), X3 (SM08B-SUR.S), X6 (SM2834-4), X7 (SM2834-4), and X11 (SM04B-SUR.S Connector to Push buttons).
- Passive Components:** Various capacitors (C1-C9), resistors (R1-R10), and diodes (D1-D5).
- Regulation:** A 3.3V regulator (U3) and a 5V regulator (U4).

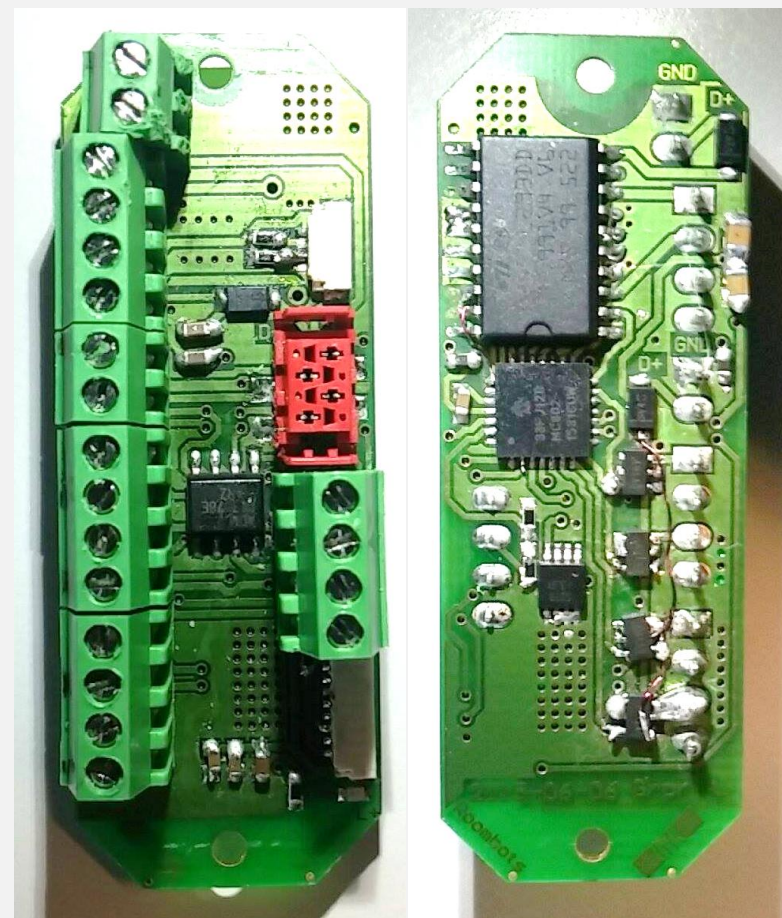
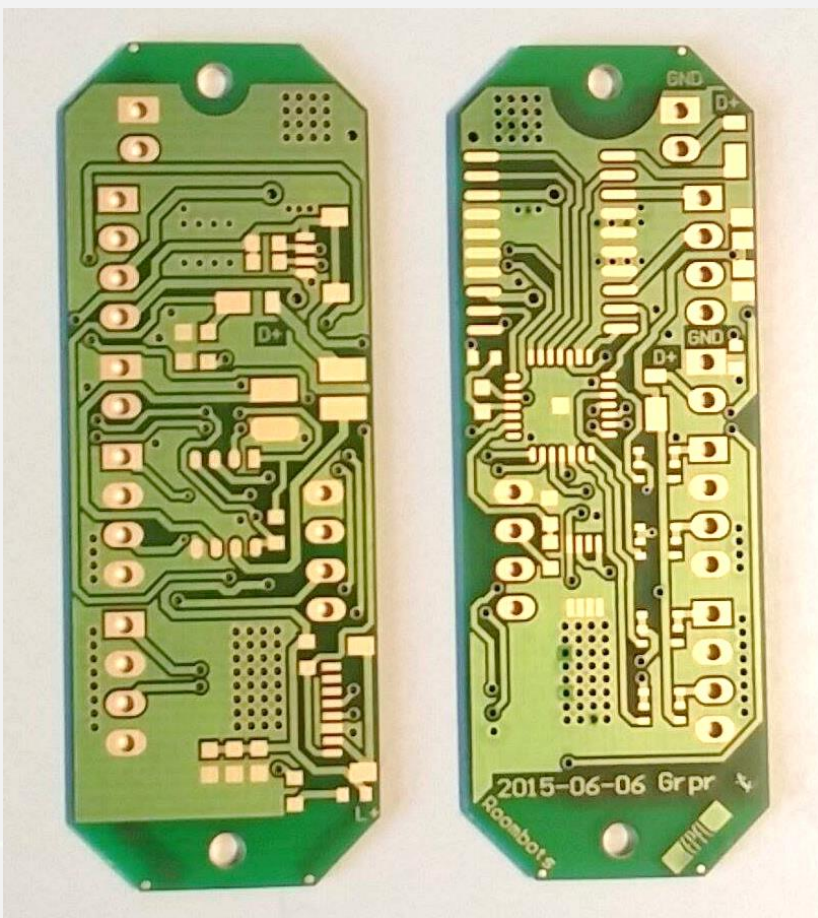
The board is divided into sections labeled A through D, corresponding to different functional areas. The title block at the bottom right identifies the project as "Gripper Control Board Main sheet" from the Biobionics Laboratory at the Ecole Polytechnique Fédérale de Lausanne, dated 10.11.2015.

# Layout





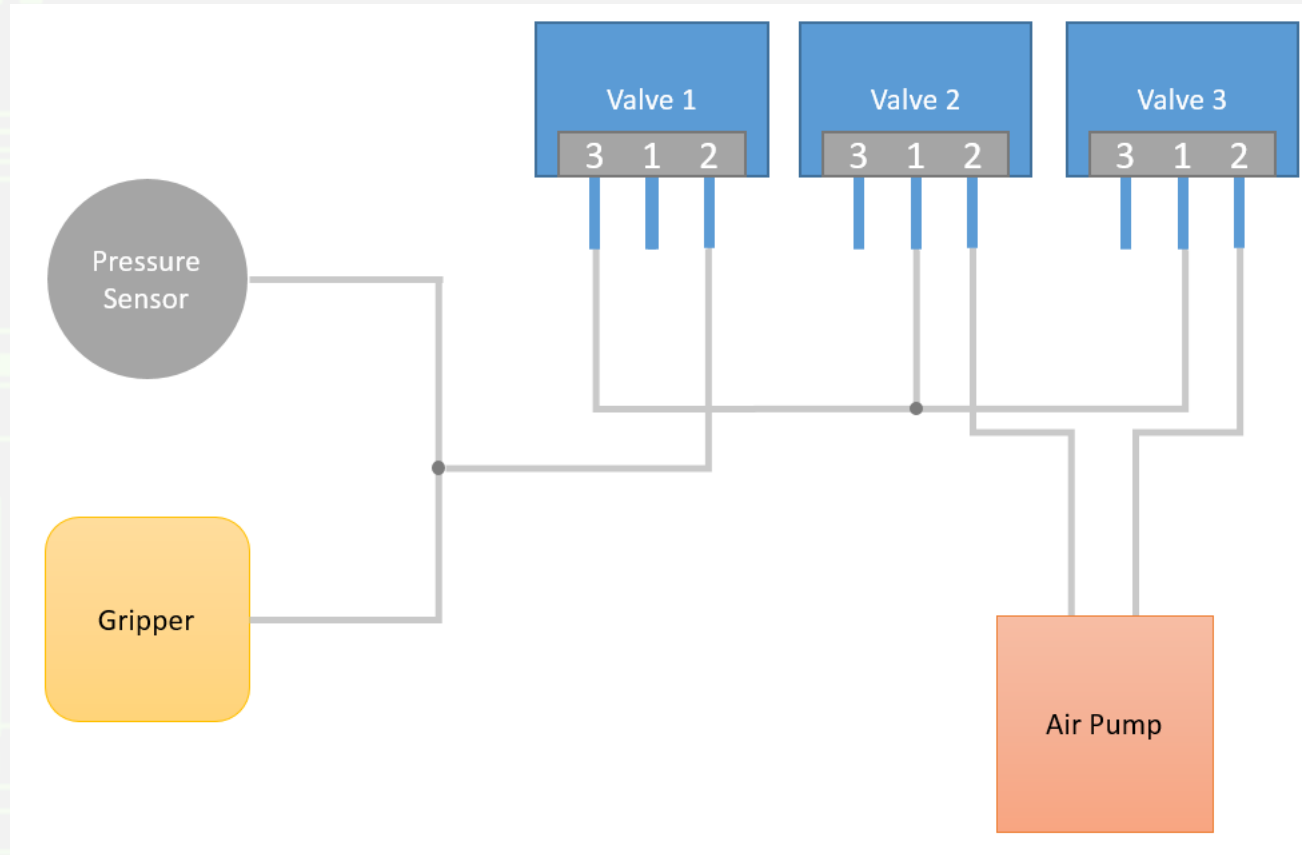
# Boards



# Firmware

- Previous firmware from other boards
- PWM and ADC configuration
- Air pump on/off control
- Hardware test commands and functions
- User commands

# Gripper Operations



#	Valve States			Operation
	Valve 1	Valve 2	Valve 3	
1	0	0	0	Hold; Gripper keeps its pressure
2	1	0	0	Atmospheric; System keeps the pressure of the Gripper at 1 atm.
3	0	1	0	Vacuum; Air Pump is able to vacuum the air inside of the Gripper
4	0	0	1	+Pressure; Air Pump is able to pump air into the Gripper

# Commands

Command	Related Function	Operation
"hold"	command_hold()	Sets system to "hold" position and stops the air pump.
"atm" & "patm"	command_atm() command_patm()	Sets system to "atmospheric" position with and without the air pump.
"grp"	command_grip()	Sets system to "vacuum" position and opens the air pump until minimum allowable pressure level reached.
"rls"	command_release()	Sets system to "+pressure" position and opens the air pump until maximum allowable pressure level reached.

Command	Related Function	Operation
"prconfig"	command_configure_pressure()	Sets and gets the different pressure types.
Pressure Type	Appended Letter	Can be read or set?
Maximum allowable pressure or releasing pressure	"r"	both
Minimum allowable pressure or gripping pressure	"g"	both
Atmospheric pressure	"a"	both
Desired pressure	"d"	both
Measured pressure	"m"	just read
Error margin	"e"	both
Command	Related Function	Operation
"pwmconfig"	command_configure_pwm()	Sets and gets the speed of the air pump.

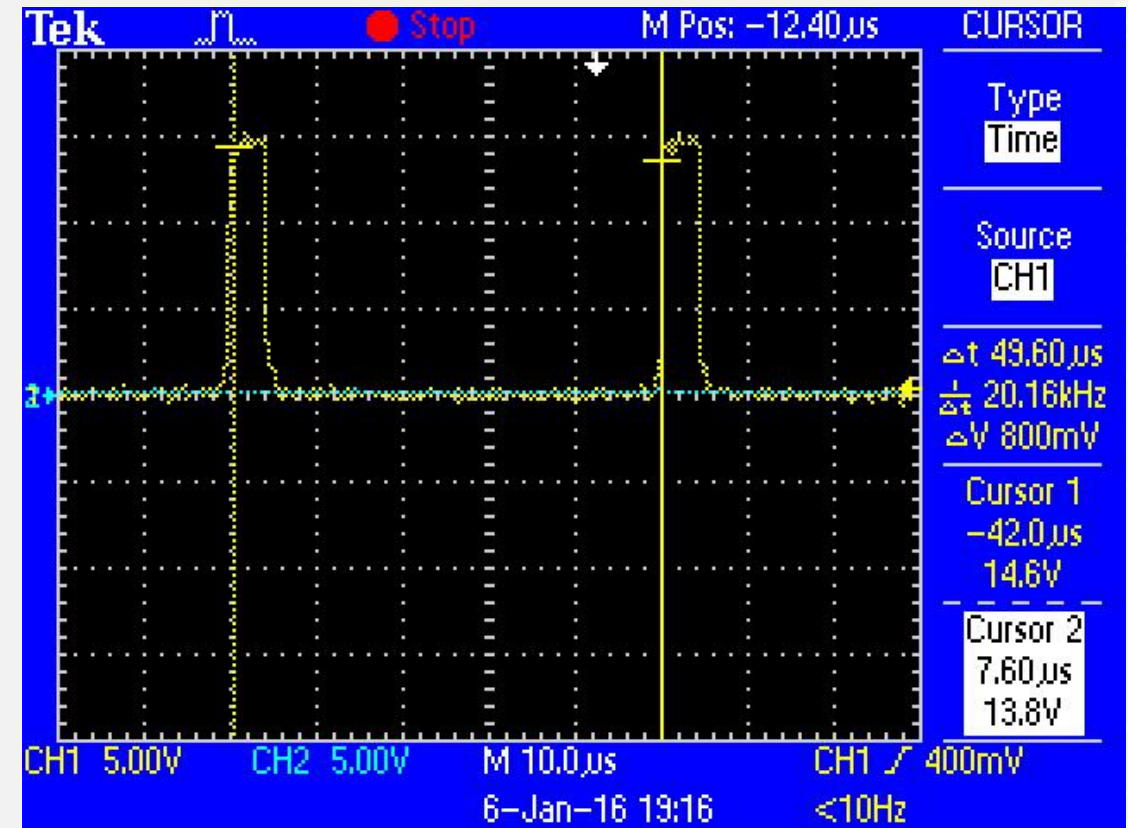


# Results

Output Voltage	12-bit Reading	Expected Pressure	Measured Pressure
3.3 V	1958	< 1atm	0.75 atm
4.3 V	3405	1 atm	0.97 atm
4.5 V	3868	> 1 atm	1 atm

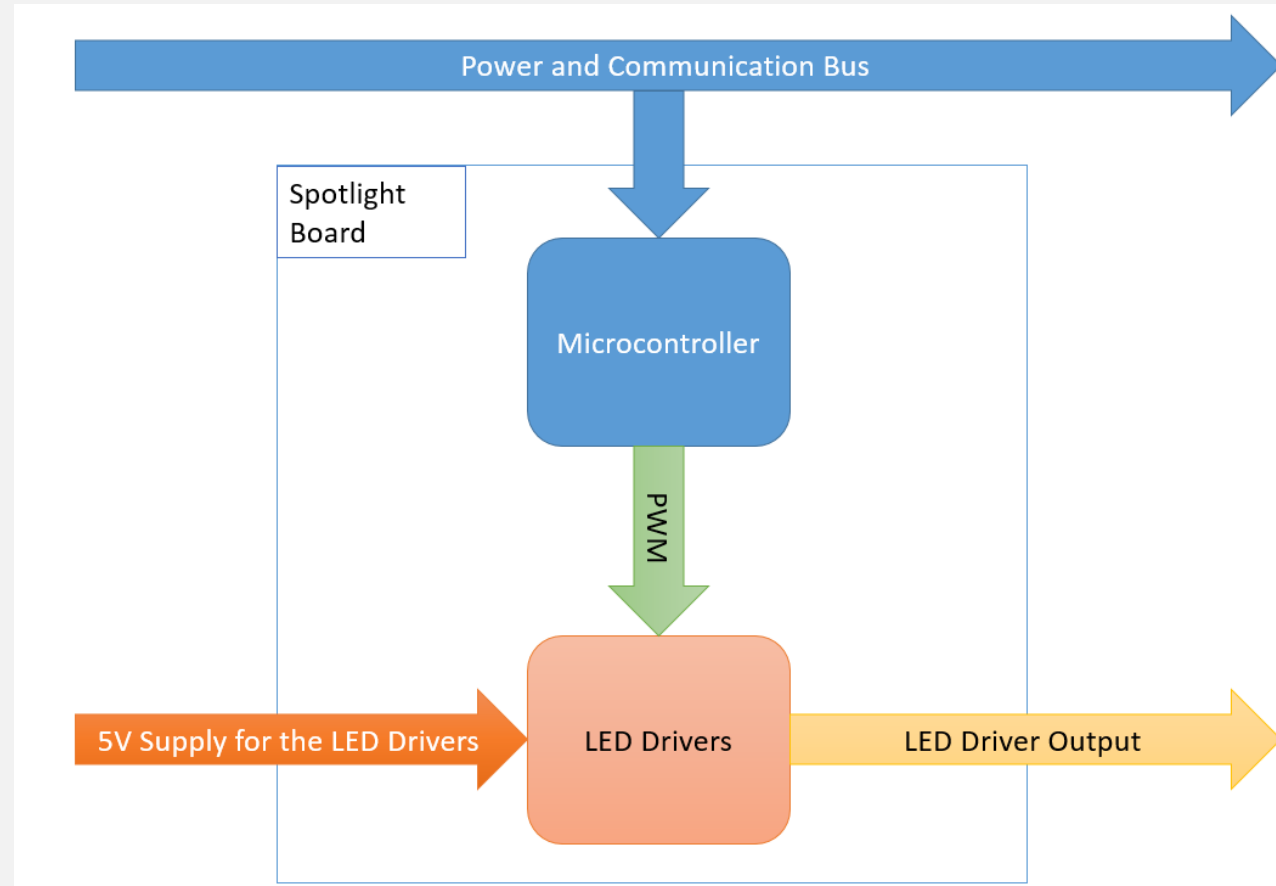
Expected Value	Measured Value	Error
10%	6.45%	35.5%
25%	23.2%	7.2%
50%	48.8%	2.4%
75%	74.6%	0.53%
90%	90.4%	0.44%



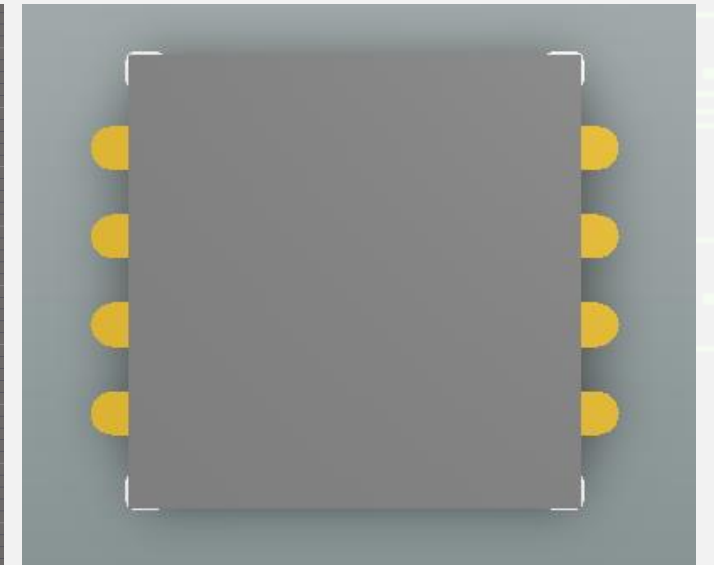
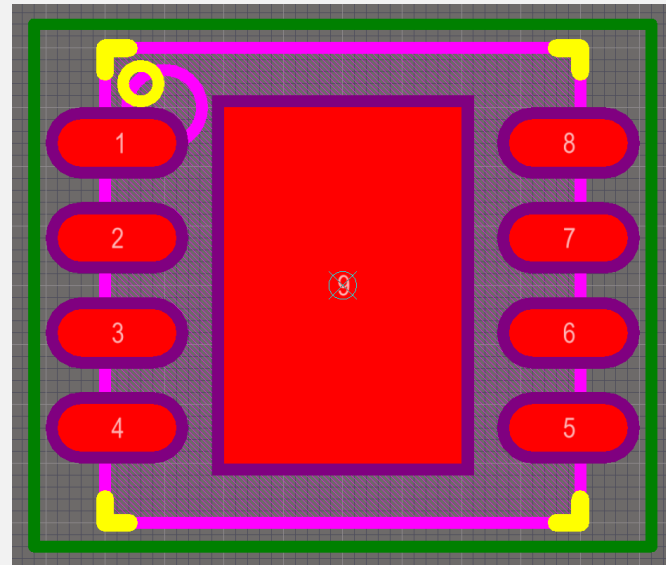
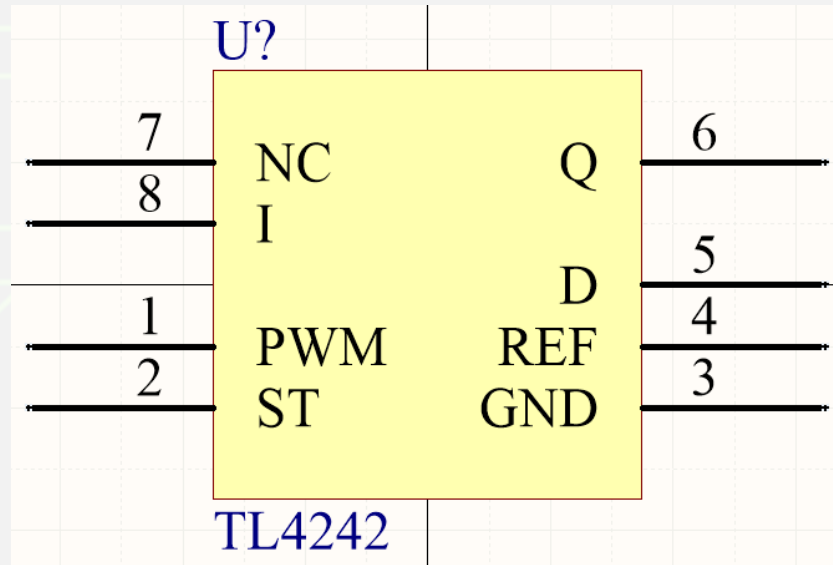
# Results



# Spotlight

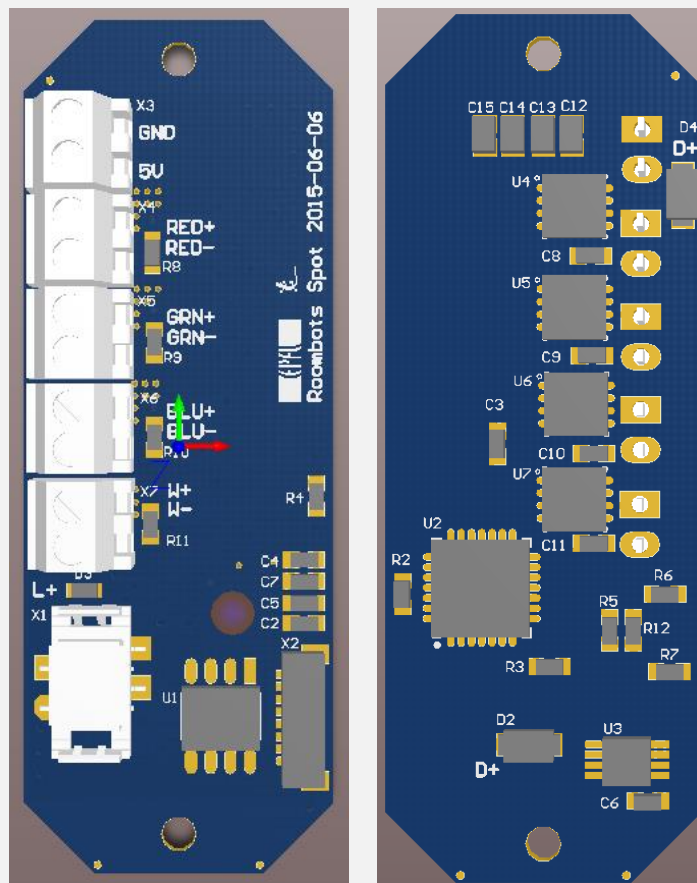


# LED Driver



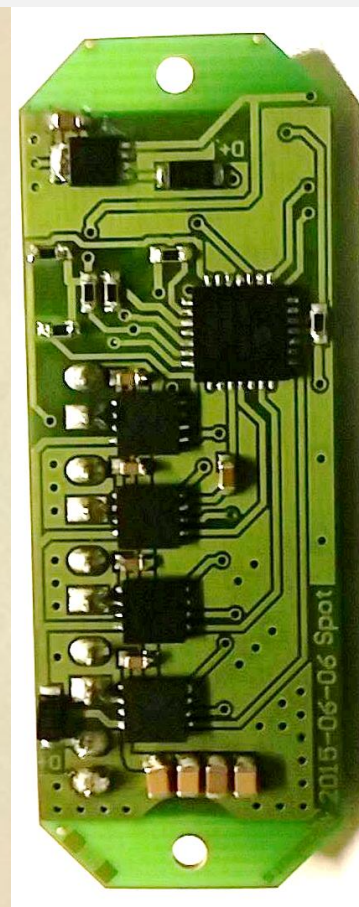
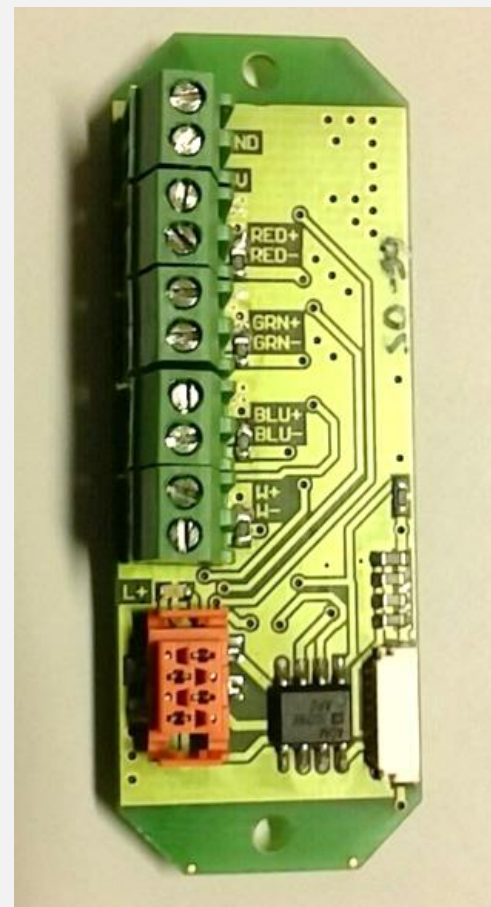
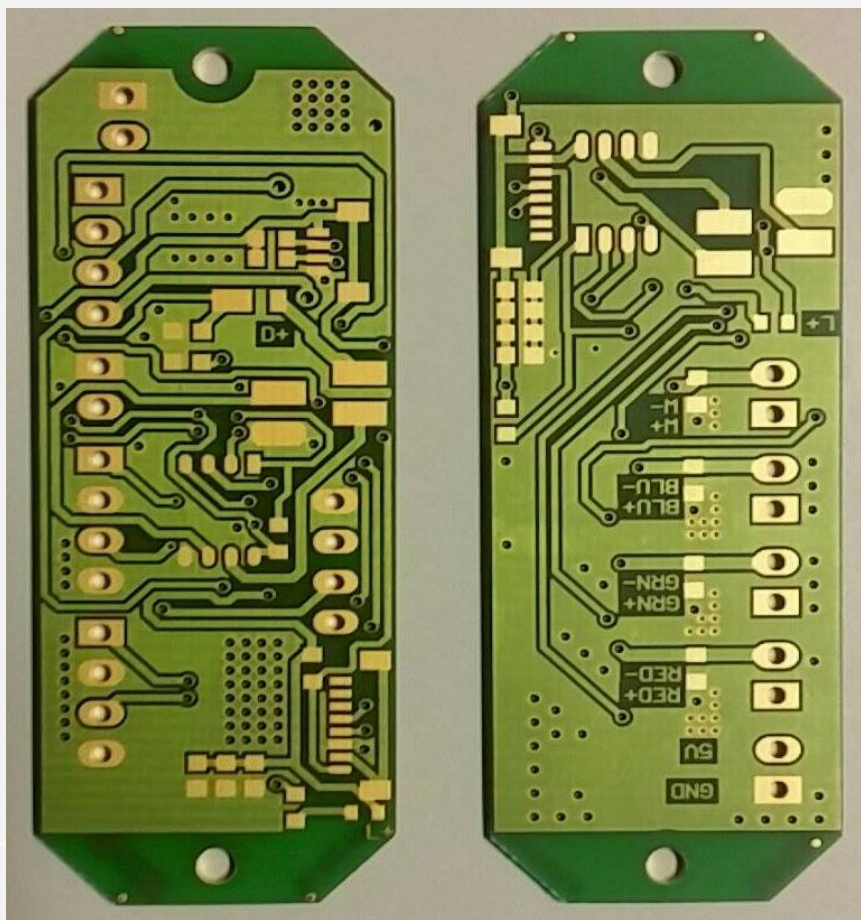
[illegible]

# Layout

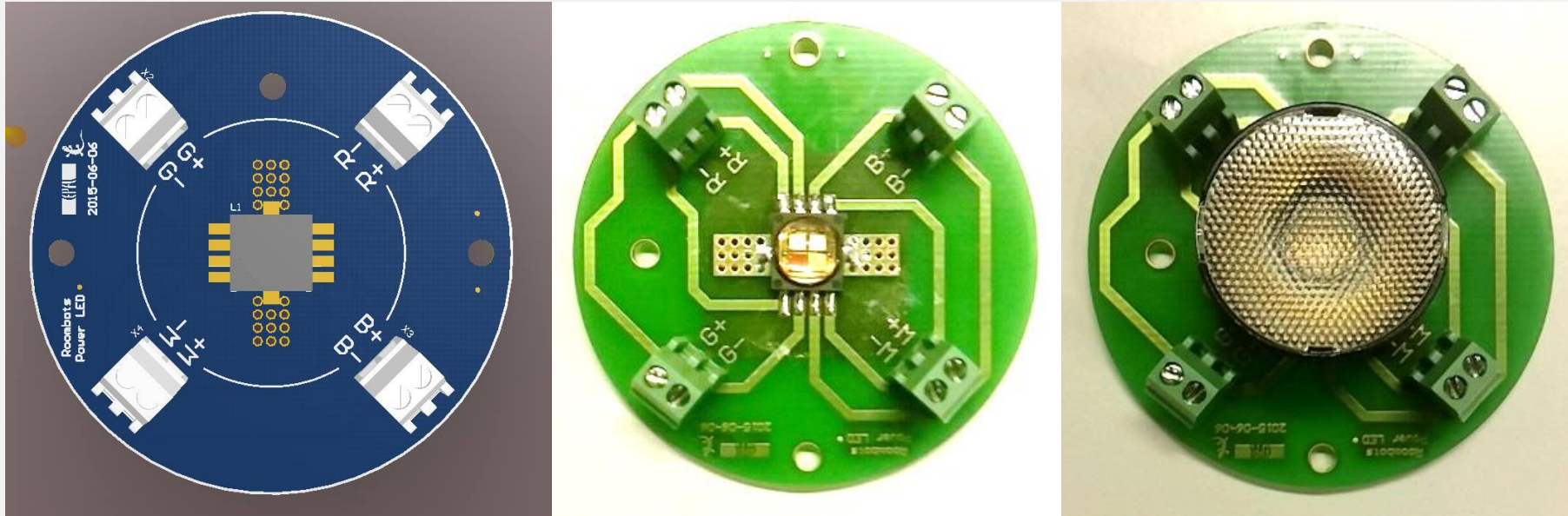




# Boards



# Power LED Board





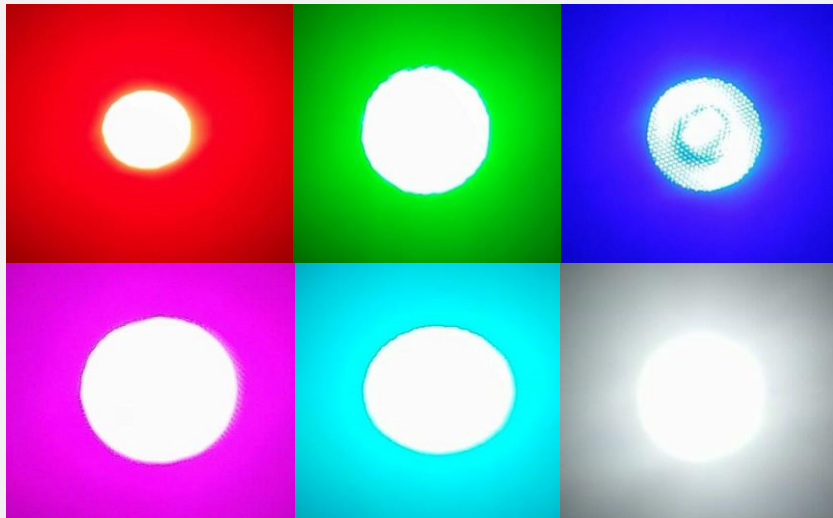
# Firmware

- Previous firmware from other boards
- PWM configuration
- Hardware test commands and functions
- User commands

# Commands

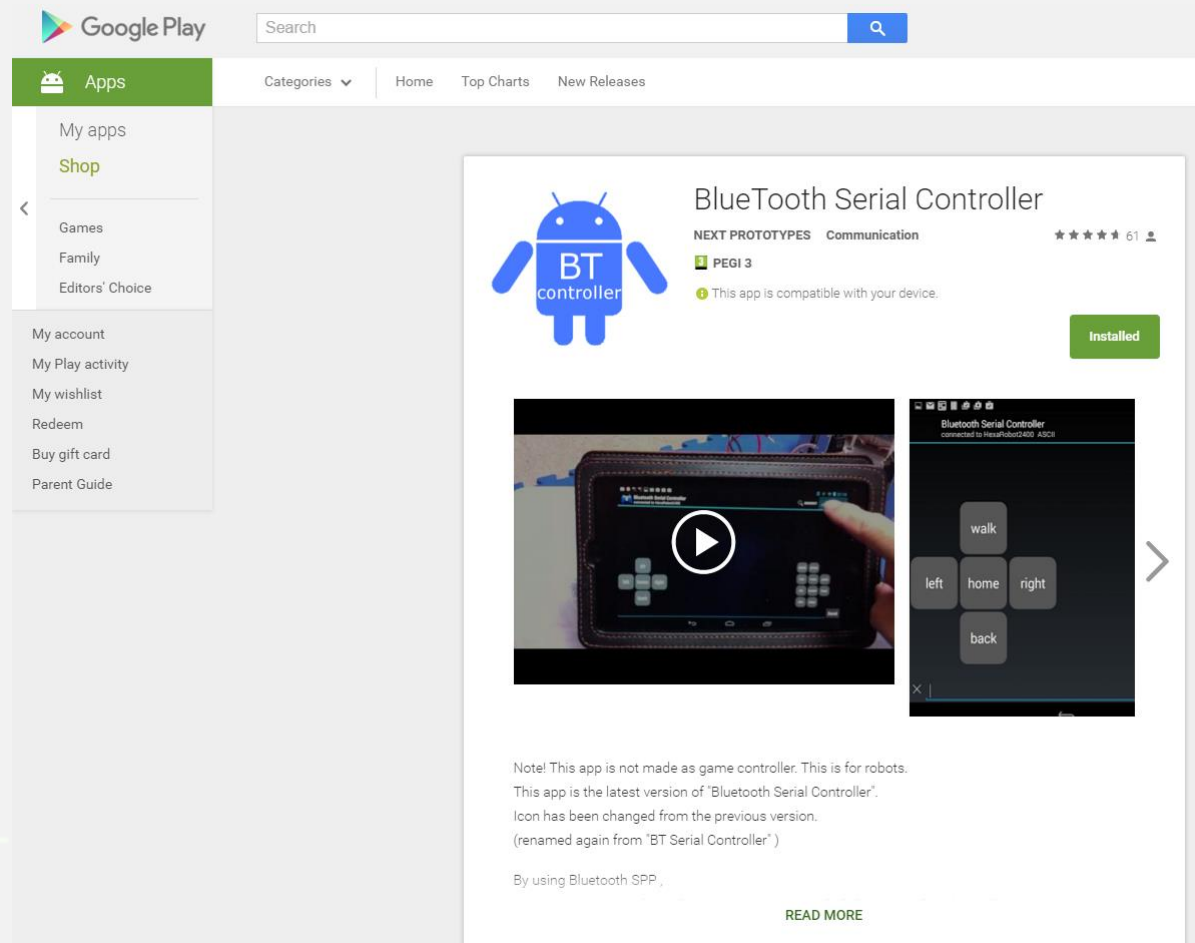
Command	Related Function	Operation
"c"	command_rgbw()	Sets the color of the spotlight.
"r"	command_rainbow()	Starts rainbow effect.
"p"	command_pulsating()	Starts pulsating effect.
"b"	command_blink()	Starts blink effect.

# Results



#	Brightness Level				Expected Current Consumption (mA)	Measured Current Consumption (mA)
	RED	GREEN	BLUE	WHITE		
1	25%	0%	0%	0%	87 mA	70 mA
2	50%	0%	0%	0%	175 mA	160 mA
3	75%	0%	0%	0%	262 mA	250 mA
4	99%	0%	0%	0%	350 mA	310 mA
5	0%	25%	0%	0%	87 mA	70 mA
6	0%	50%	0%	0%	175 mA	160 mA
7	0%	75%	0%	0%	262 mA	260 mA
8	0%	99%	0%	0%	350 mA	320 mA
9	0%	0%	25%	0%	87 mA	70 mA
10	0%	0%	50%	0%	175 mA	170 mA
11	0%	0%	75%	0%	262 mA	270 mA
12	0%	0%	99%	0%	350 mA	330 mA
13	0%	0%	0%	25%	87 mA	70 mA
14	0%	0%	0%	50%	175 mA	170 mA
15	0%	0%	0%	75%	262 mA	250 mA
16	0%	0%	0%	99%	350 mA	320 mA
17	25%	25%	0%	0%	175 mA	150 mA
18	25%	25%	25%	0%	262 mA	220 mA
19	25%	25%	25%	25%	350 mA	300 mA
20	50%	50%	0%	0%	350 mA	320 mA
21	50%	50%	50%	0%	525 mA	490 mA
22	50%	50%	50%	50%	700 mA	660 mA
23	75%	75%	0%	0%	525 mA	520 mA
24	75%	75%	75%	0%	787 mA	800 mA
25	75%	75%	75%	75%	1050 mA	1060 mA
26	99%	99%	0%	0%	700 mA	640 mA
27	99%	99%	99%	0%	1050 mA	1000 mA
28	99%	99%	99%	99%	1400 mA	1290 mA

# Android



# Conclusion

- Design two boards for Roombots modules
- Achievements:
  - Color Control & Effects
  - Pressure Control & Gripper Operations

# Future Improvements

- Firmware correction
- Pressure Sensor filtering
- Android

# Demo & Questions?