Building a Drone from scratch

Igor Stoppa

Embedded Linux Conference
October 2016

V 0.1.0

Disclaimers

Opinions expressed in these foils represent exclusively to the author’s view.

All Logos and Trademarks represented belong to their respective owners.

Summary

- Intro - what is this about? - why from scratch?
- Setting the Goals
- Requirements: must have, nice to have, long term
- Identifying the constraints: time, materials, means.
- System design: centralized / distributed, make / buy
- The gory details:
  - HW / SW selection, system architecture
  - Dealing with physical systems:
    motors in real life, inertia, max battery power
- Ideas for future expansion

Intro - what is this about?

- Learning-by-doing project
- Attempt to build a drone of acceptable quality, while:
  - keeping the cost low;
  - keeping the overall complexity low;
  - using off-the-shelf components easily obtainable through major worldwide retailers.
  - achieving sufficient modularity to support various configurations

Intro - Why from scratch?

- Many frameworks available, trying to hide the complexity.
  Useful for productization, less open to free form experimentation.
- SW platforms tend to focus on specific HW.
  It simplifies the design and the verification, at the expense of freedom of choice.
- It’s more interesting
- Challenge: use the most out of the HW selected

Setting the Goals

4WD Drone:

less glamorous than flying, but less likely to break.

Easy upgrade path:

no proprietary solutions, compartmentalize functionality.

Low cost:

stock parts from popular kits, SW to improve accuracy.

Ease of debug:

tap into standard interfaces between building blocks.

Requirements

Must Have

Speed control, Steering, Remote Control

Nice to Have

Obstacle detection, Obstacle avoidance, camera stream

Long Term

Remote Computer Vision, Onboard Computer Vision

Constraints to Development

Limited time

Only few hours per week, each week a new feature.

Costs

It shouldn't break the bank, especially when taken as educational tool/toy. This includes the tools used.

Material

It should rely only on components readily available at affordable price, through worldwide distribution channels.

System Design

Extensibility

Allow additional HW features. Ex: accelerometer.

Modularity

Segregation of different functionality.

Ease of unit-test and debug, less interference.

Real time response

Deterministic cap to reaction times, in specific cases.

Power Efficiency

Minimize power loss in major use cases (DC motors).

System Design - continued

Low Mass

Minimize negative effects of inertia:

- higher power (peak current) required to alter the state (steer, speed up/down)
- higher chance to drift

Circumscribe electrical damage

In case of electrical fault (misconnection/short, etc.), preserve the most expensive component(s) from damage.

Single Board vs Multiple Boards

Comparison	Single Board	Multi-Boards
Extensibility	Less	Yes
Power Efficiency	Yes	Less
Low Mass	Yes	Less
Modularity	Less	Yes
Real time Response	Less	Yes
Damage Control	Less	Yes

Considerations

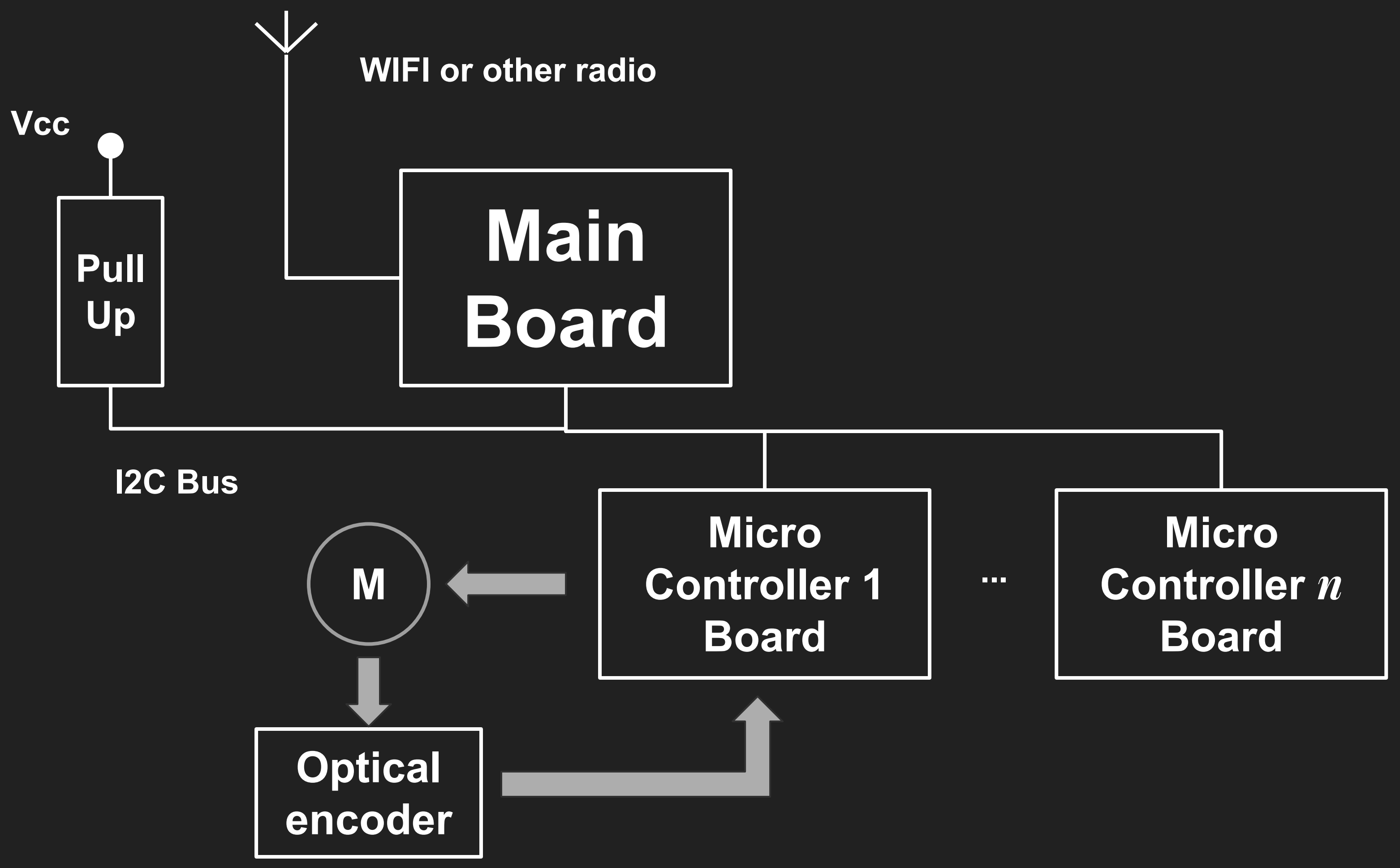
There is **no perfect solution** - unsurprisingly.

Both can be made to work, with ad-hoc adjustments.

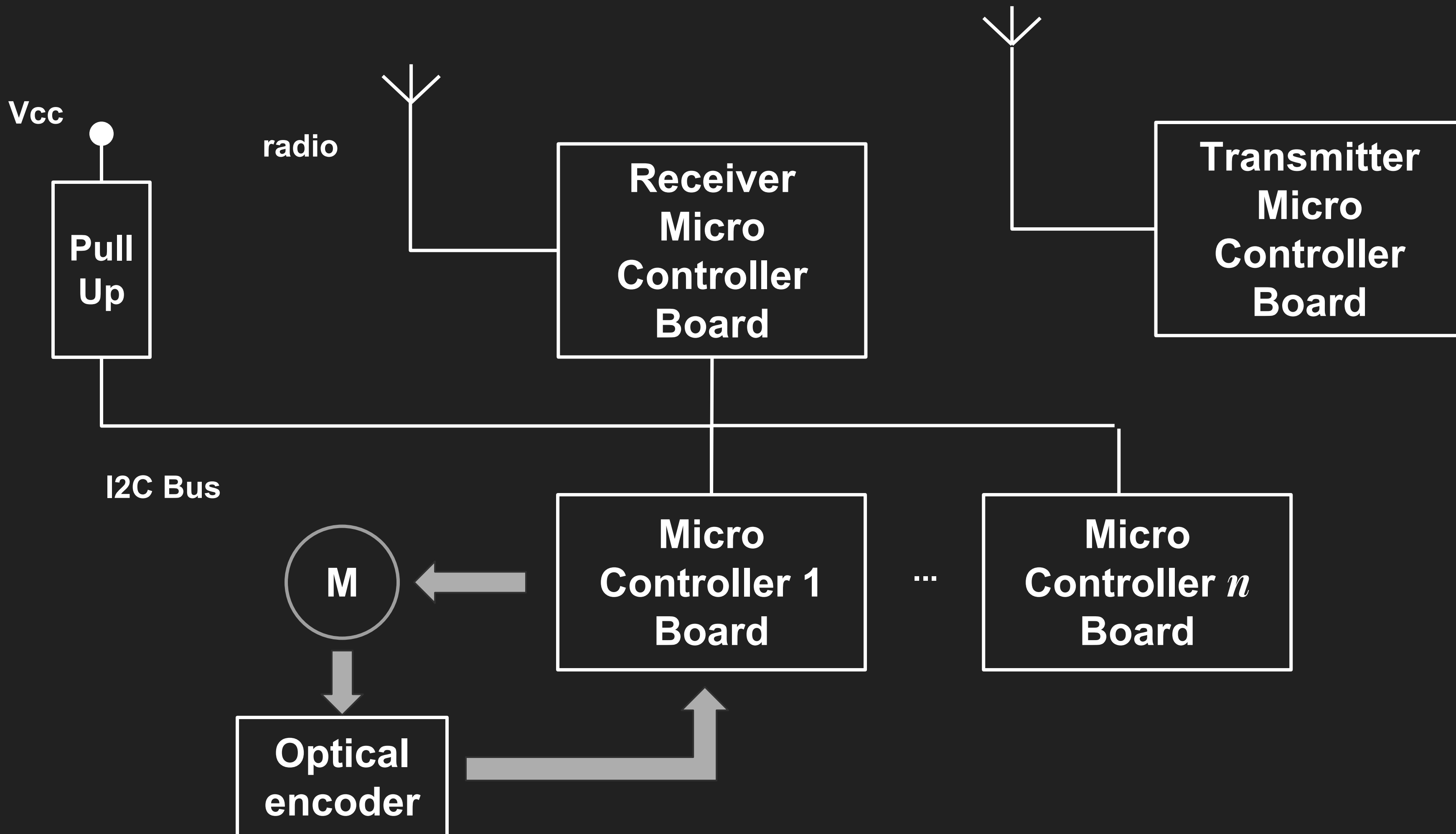
The **Multi-Boards approach wins** because:

- It is better at protecting the “Main” board.
- It can even omit the “Main” board - ex: simple RC drone.
- It enables the use of an RTOS for the time-sensitive tasks.

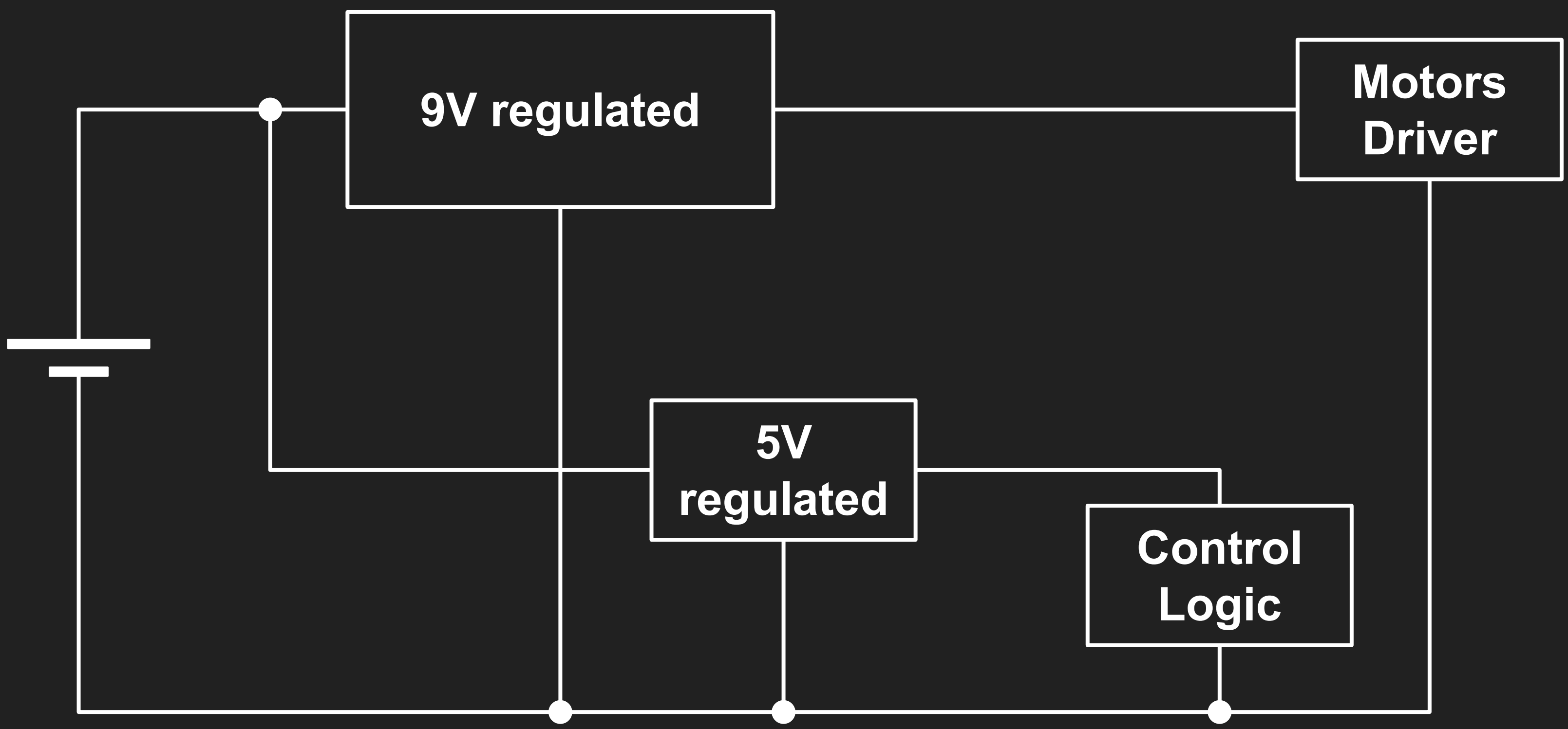
Overall System Architecture



RC-Variant


Vcc
radio
Pull Up
Receiver Micro Controller Board
Transmitter Micro Controller Board
I2C Bus
M
Micro Controller 1 Board
...
Micro Controller n Board
Optical encoder

Power Distribution - 1 Battery



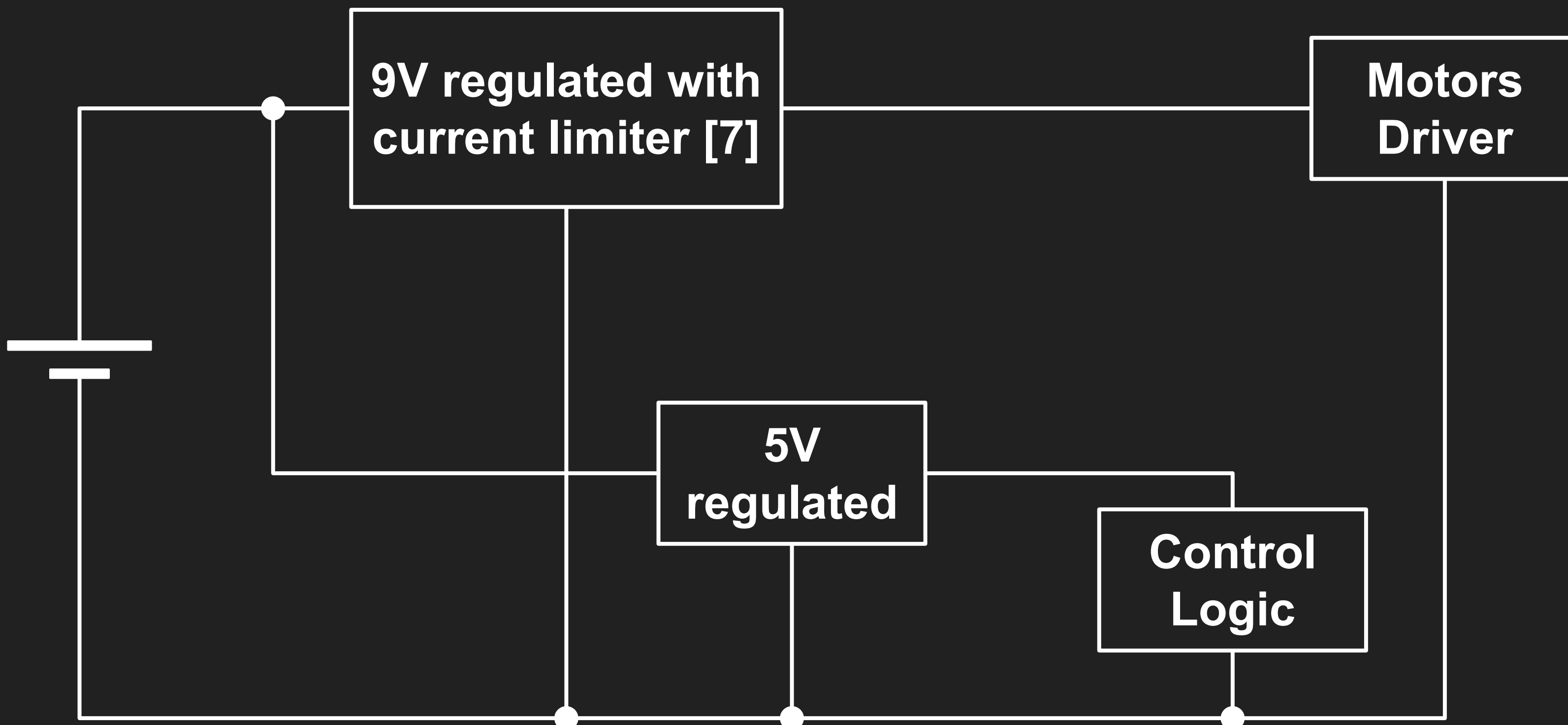
Power Distribution - 1 Battery

1 single battery for powering both logic and actuators

- Actuators can try to draw more current than the battery provides while accelerating. Ex: inversion of rotation, start.
- Voltage across the battery pack can drop.
- The drop can be enough to starve the regulator feeding the logics.

<u>Solution: limit the max current used by the actuators.</u>

Power Distribution - 1 Battery



Motors - options

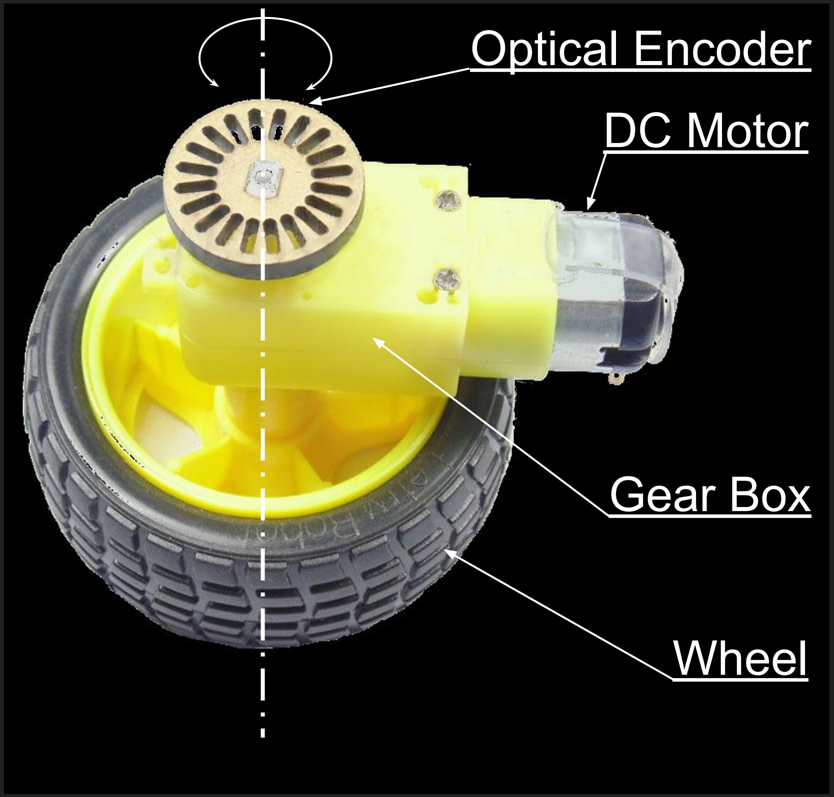
DC Motor

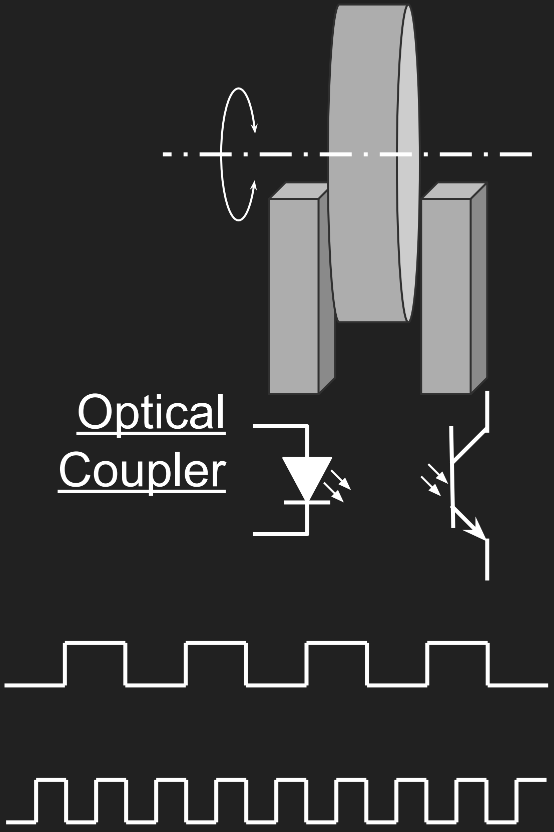
- Pros: fast, naturally continuous, robust.
- Cons: needs additional circuitry for speed/position control

Servo Motor

- Pros: fast, high torque
- Cons: needs modification to be continuous, can vibrate when idle, more expensive.

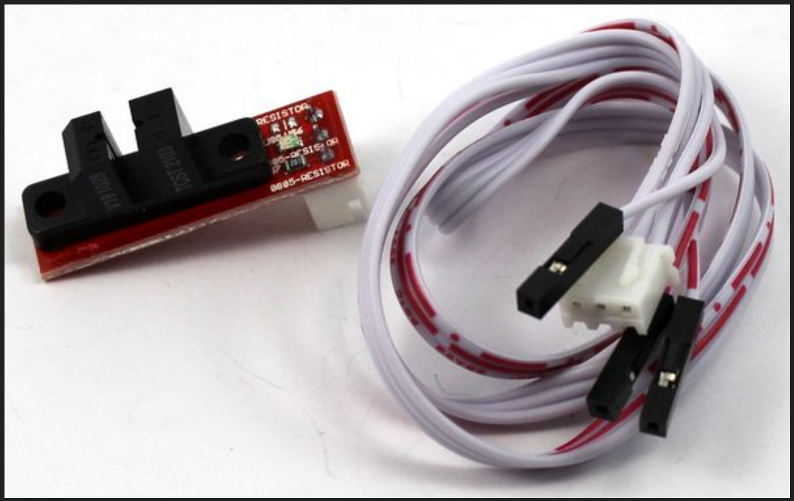
Choice: DC Motor





Frequency proportional
to the rotation speed

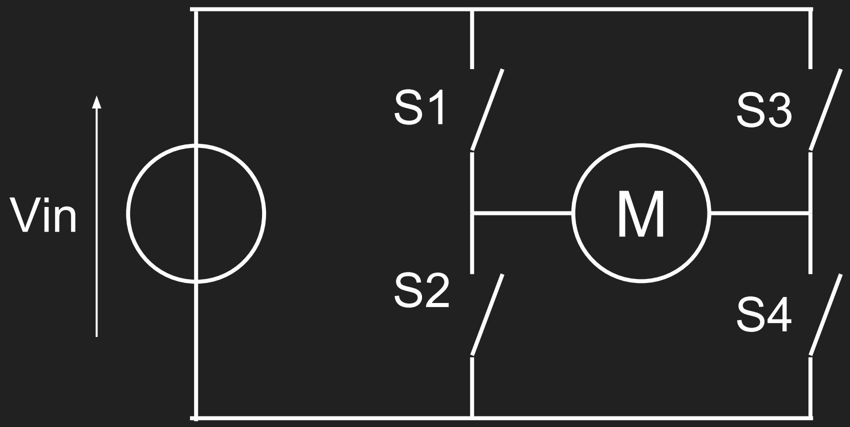
Optical Coupler



End Stop for 3D printer TCST2103 [1]

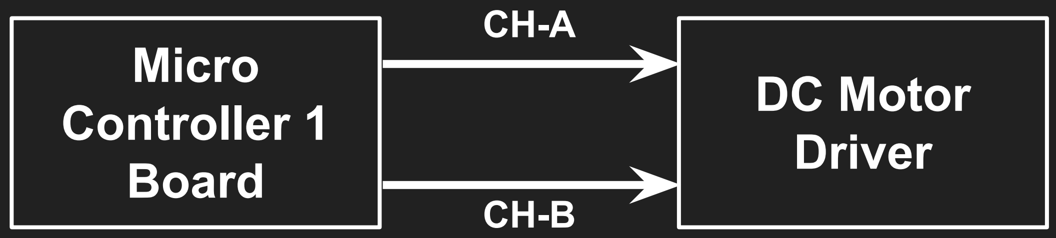
- Fairly cheap
- Sufficiently accurate
- Compatible with the dimensions of the optical encoder.

Driving DC motors - H bridge



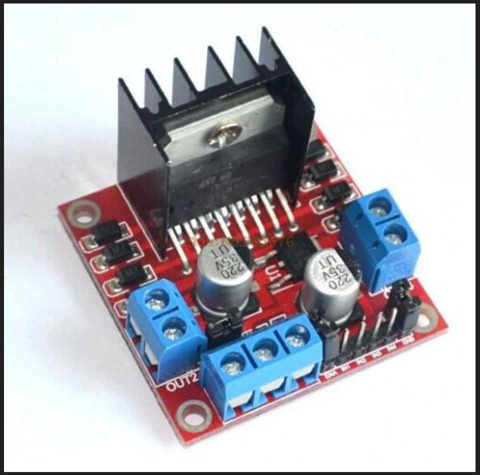
- Allows to apply voltage across a load in either direction.
- Various technologies used to implement S1..S4
- Different levels of efficiency.

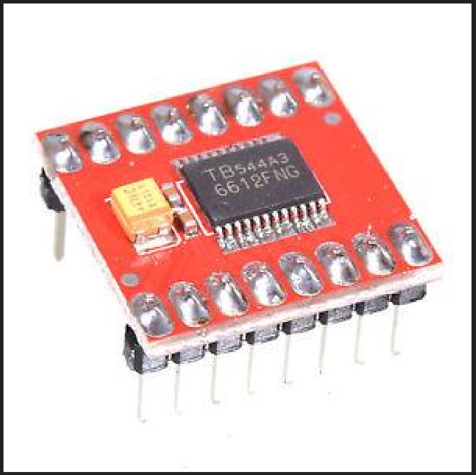
Driving DC motors - signals



CH-A/B			(A/B)O2	(A/B)O1
(A/B)IN2	(A/B)IN1	PWM(A/B)		
0	0	DON’T CARE	FREE SPINNING	
0	1	PWM	CLOCKWISE	
1	0	PWM	COUNTER CLOCKWISE	
1	1	DON’T CARE	LOCKED	

Motors Drivers - options [2]





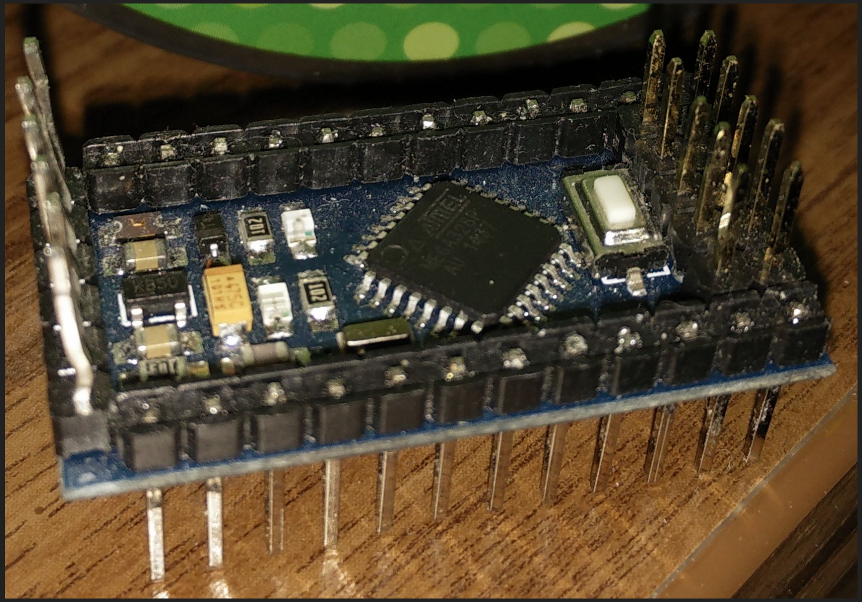
L298N

- Cheap
- Big Internal Power Loss
- Large (HeatSink)

TB6612FNG

- More expensive
- Small Internal Power Loss
- Small (no need to dissipate power)

Low Level Automation - uC



Arduino Pro Mini (AVR328p) [3]

- Has I2C interface
- Sufficiently powerful to perform the required calculations
- For each motor:
  - Drive status
  - Dedicate PWM line
  - Optical Encoder input

Motor Control and Feedback

Motor status control

- 2 independent GPIOs for each motor

PWM

- 2 independent counters, each feeding into 2 dividers
- Independent control for each motor, allows for calibration

Optical Encoder input

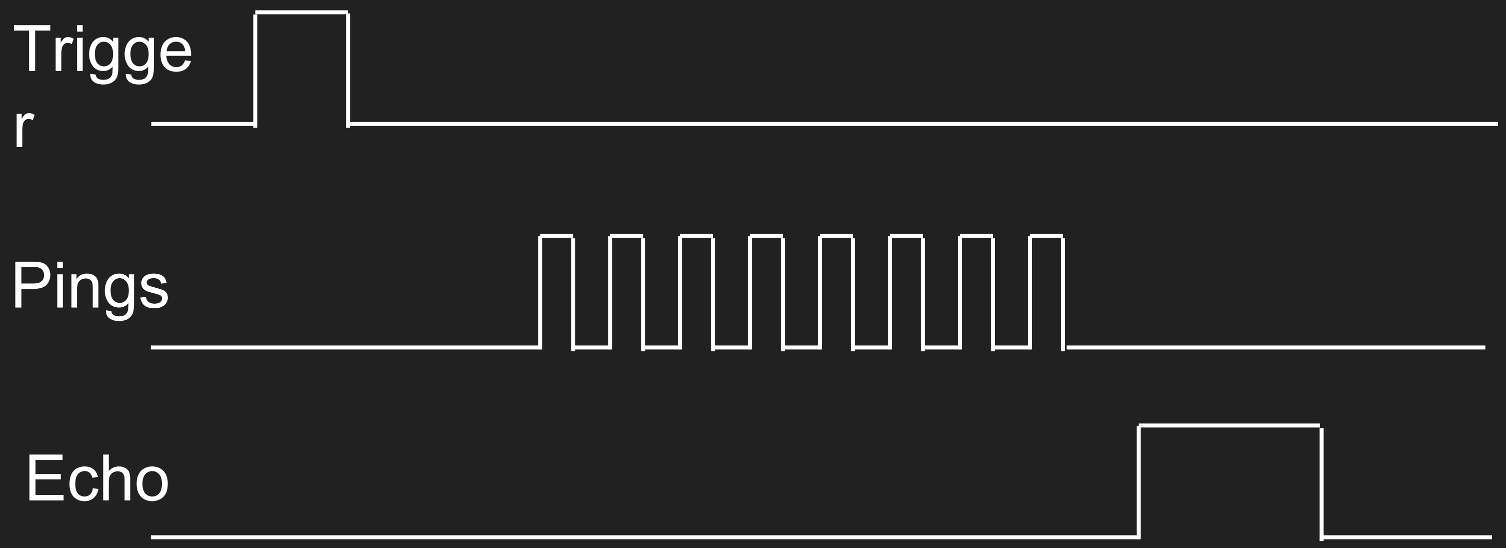
- 1 GPIO for each motor encoder, as IRQ, to avoid polling
- Only the counters are bumped in IRQ context, the rest as bottom half

Proximity Sensor

Bat-like: send a burst of waves, waits for the echos [8]

2cm - 400cm range

15 degrees aperture

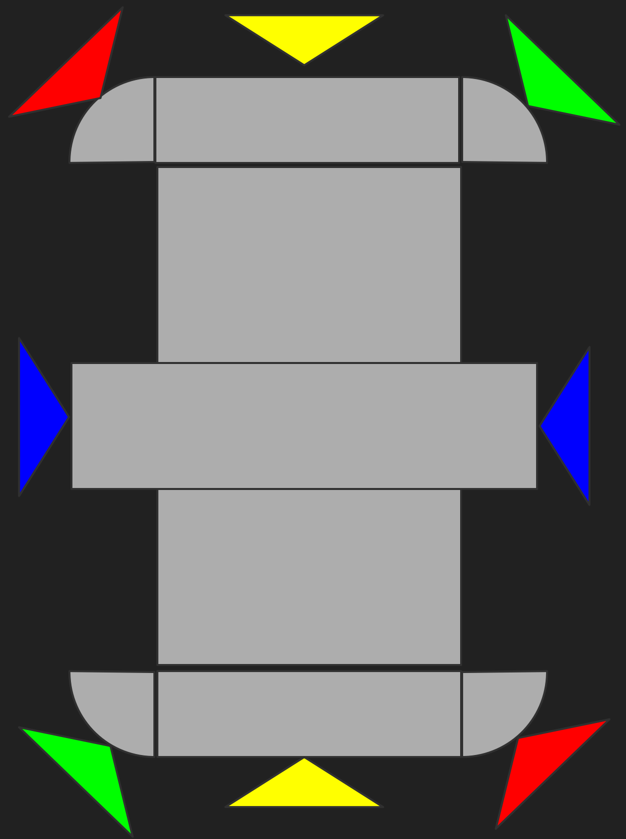


Proximity Sensor

Create pairs that do not interfere with each other.

Activate the pairs clockwise.

Possible improvement: create double pairs that are orthogonal.



Running the microController

main() Program

- Main loop with functions
- interrupt handlers

8-bit RTOS

- Interrupt handlers
- Tasks Scheduling
- Semaphores
- Mailboxes

RTOS selection

FreeRTOS [4]

- GPLv3 for non commercial
- Only for ATMega323, but not for ATMega328p
- Many (mostly dead) unofficial ports to Mini Pro
- Not very small memory footprint.

ChibiOS [5]

- GPLv3 for non commercial
- Essential BSP for Mini Pro
- Small footprint

I2C Development and Debugging

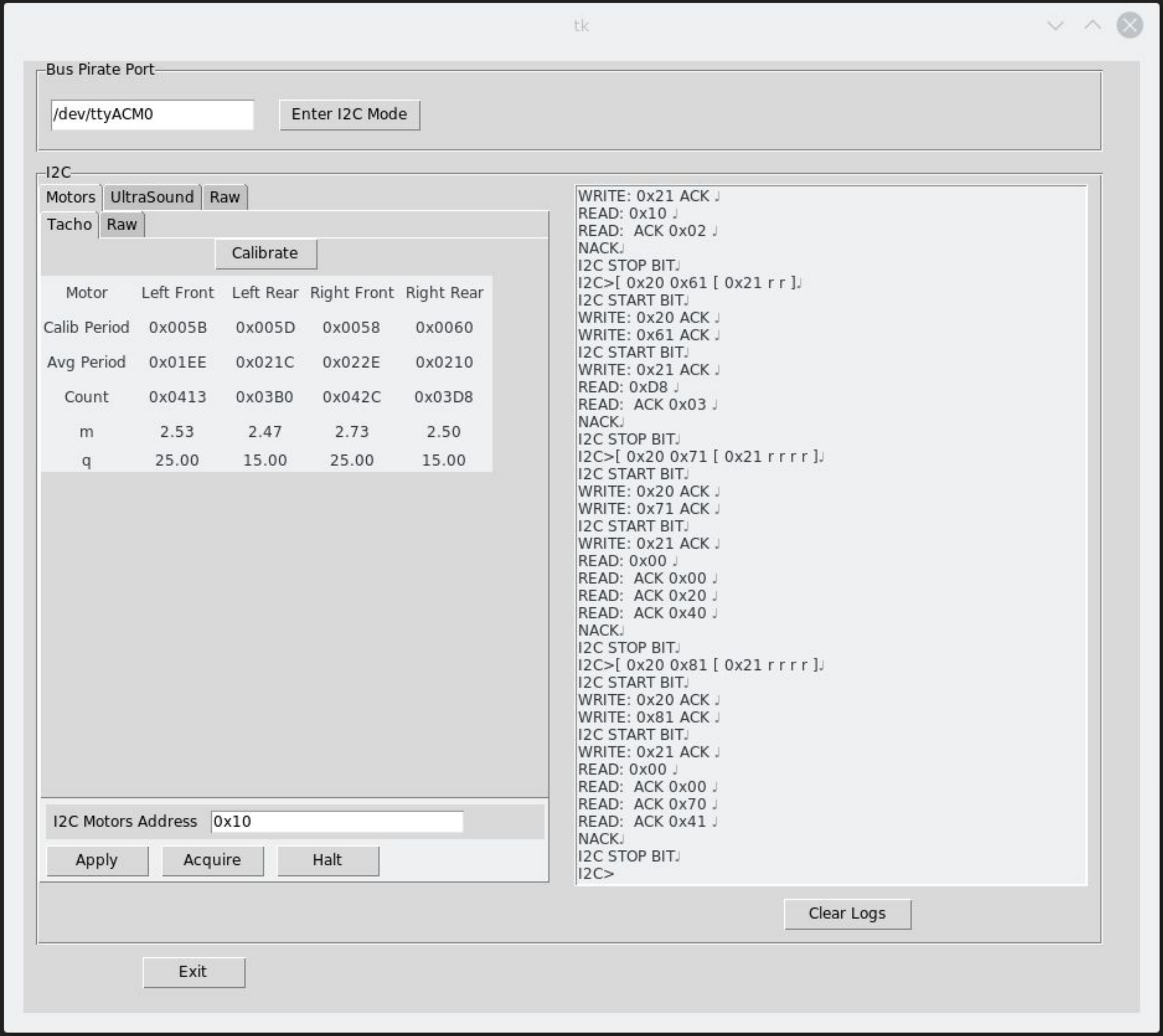
HW tools summary:

- HW debugger/flasher - AVR Dragon
- Bus low level protocol analyzer/snooper - Bus Pirate
- Logical analyzer - SigRok + Pulseview
- USB scope - Hantek + Openhantek

Full dissertation on I2C from ELC NA 2016 [6].

I2C High Level Protocol debugging

Need to create custom tools, for non-trivial testing of both the protocol and the implementation of the API.



Main Board Selection

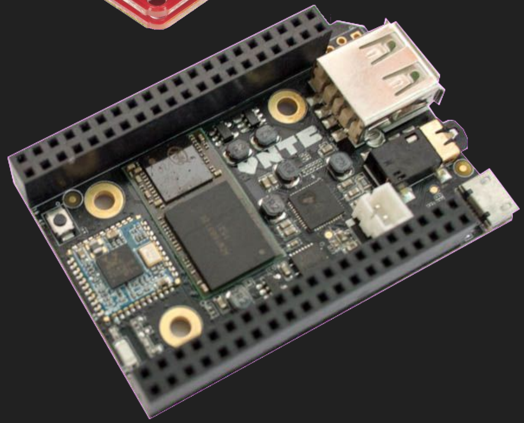
Requirements

- It must run linux
- Low power consumption
- I2C interface - master
- WiFi interface
- Small form factor
- USB OTG/Master

Main Board Selection

Options

- **[Intel Edison [9]]()**
  - Pros: powerful, small.
  - Cons: $$, modules $$
- **Next Thing CHIP [10]**
  - Pros: cheap
  - Cons: delayed
- **Intel Joule [11]**
  - Pros: powerful
  - Cons: $$$,
    Geppetto PCB $$$



Future

- Accelerometer
- Optical Flow cameras on the sides
- Computer Vision
- GPS
- LIDAR
- Port to quadcopter.

Questions?

Thank you!

Backup Info

References

[1]. http://www.alldatasheet.com/datasheet-pdf/pdf/26411/VISHAY/TCST2103.html
[2]. http://forum.makeblock.cc/t/the-review-of-dc-motor-drivers-l298n-tb6612fng-and-lv8406t/372
[3]. https://www.sparkfun.com/products/11113
[4]. http://www.freertos.org/
[5]. http://www.chibios.org/dokuwiki/doku.php
[6]. http://events.linuxfoundation.org/sites/events/files/slides/ELC%202016%20-%20I2C%20hacking%20demystified_0.pdf
[7]. http://www.ti.com/product/LM2596
[8]. http://www.micropik.com/PDF/HCSR04.pdf
[9]. http://www.intel.com/content/www/us/en/do-it-yourself/edison.html
[10]. https://getchip.com/
[11].