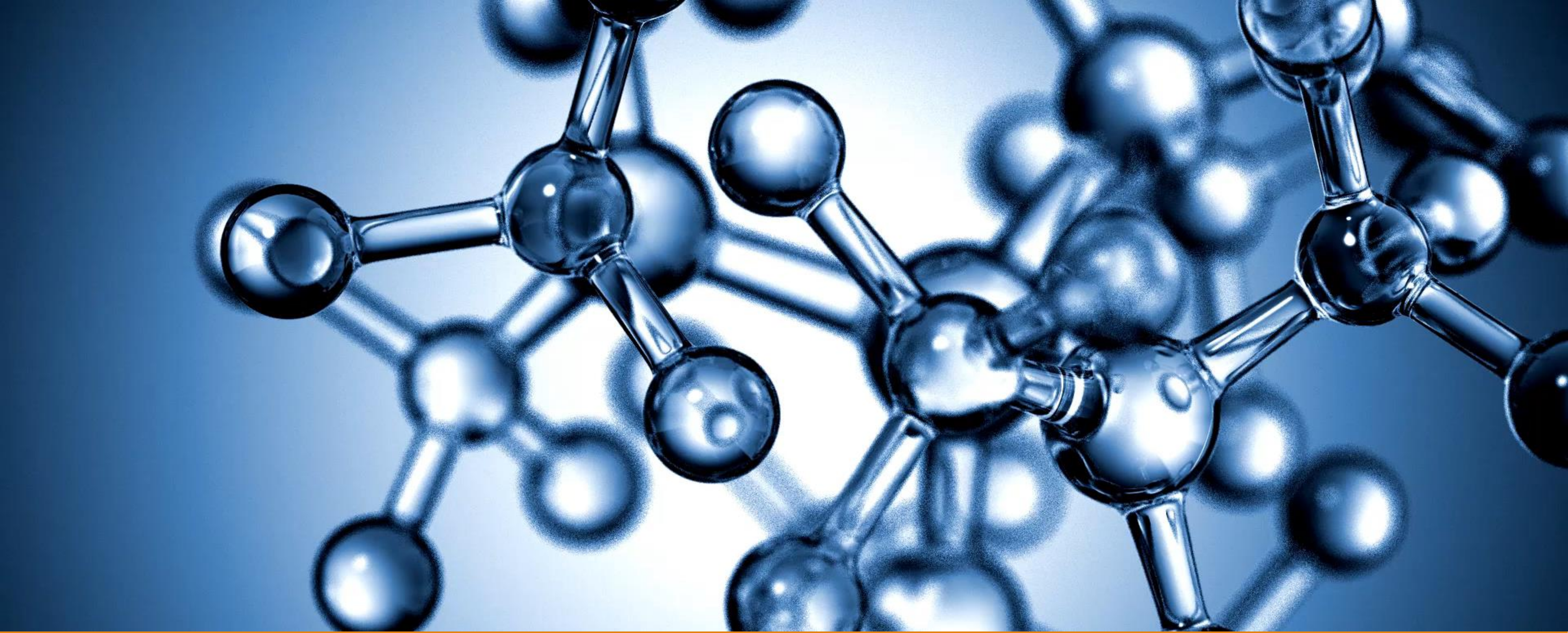


Ideas for a «Bionic Platform» - Bionic Meeting 2 Sept. 2022

Alberto Navatta – e-Nable Italia

alberto@e-nableitalia.it

What is it?

1. A «standard» **open source** platform abstracting a generic architecture and that can implement several models of bionic or electronic controlled devices.
2. A common set of libraries and tools allowing simple design of an electronic controlled device and signal and data analysis
3. A framework that can potentially be implemented on different specific hardware platforms
4. A set of general purpose hardware modules that can be combined (with also commercial products/breakout boards) to achieve several tasks

Why?

1. There are several mechanic hand projects and very few «electronic controlled solutions»
2. Electronic controlled solutions seems to be derivative of custom projects, tailored on very specific needs
3. Electronic controlled solutions seem to not exploit too much of the electronics and embedded system programming, instead they have very basic control logic (*Attention: this doesn't mean they don't work well for their specific task*)
4. An approach aiming to develop a «standard» platform has several benefits:
    1. It would facilitate vertical solutions development on top of it
    2. It will make much of the work made by volunteers reusable in other (not previously defined) contexts and environments
    3. It would allow volunteers with no skills in electronics and computer programming to reuse «implementations» based on the Bionic Platform in their projects

How?

We have identified two main development paths:

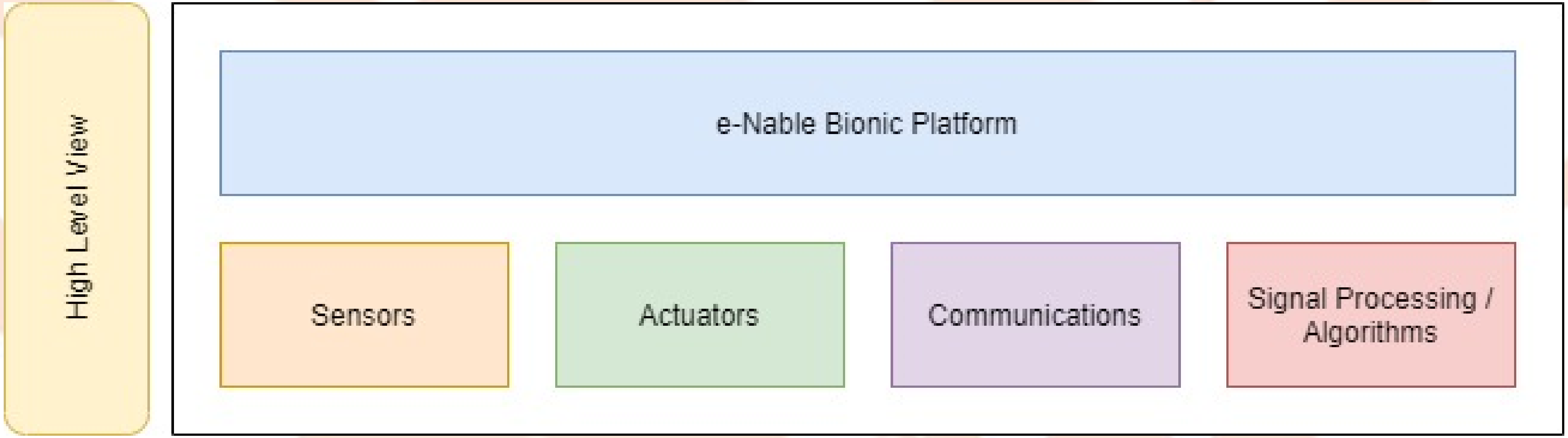
1. Create a *Bionic Software Framework:* a set of high level software libraries specialized for the development of bionic/electronics controlled devices (hopefully supporting multiple hardware implementations)

2. Develop a general purpose hardware platform that can be used to develop bionic/electronics controlled devices

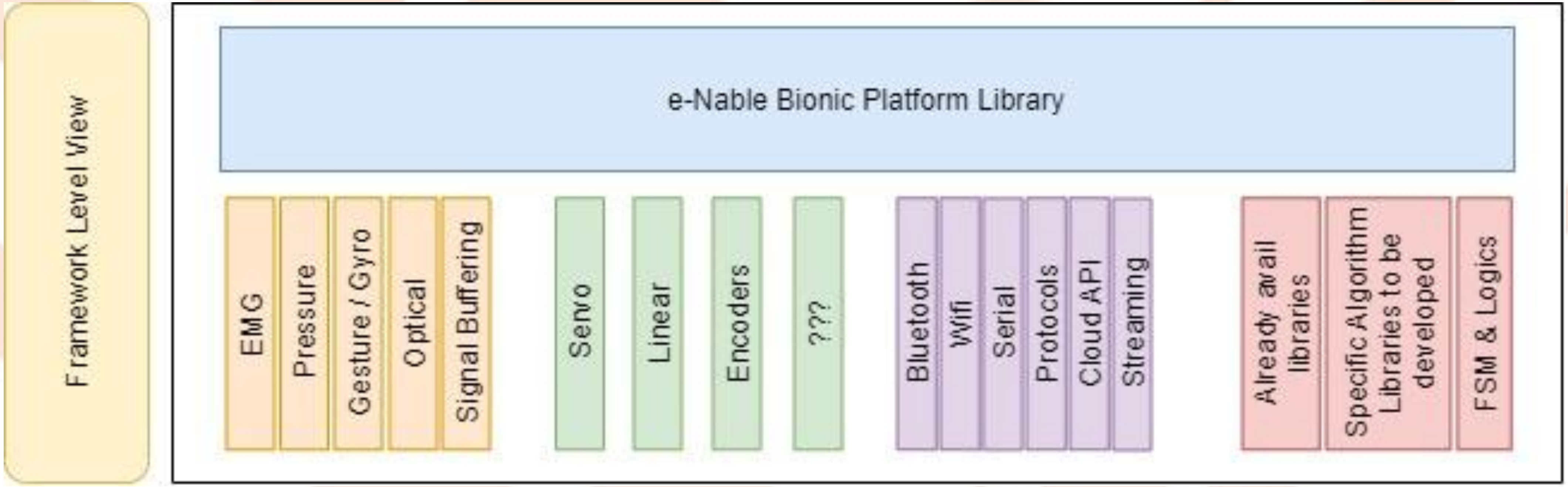
Platform Packages

1. Sensors
2. Actuators
3. Communications
4. Algorithms/Signal Processing

High Level Architecture



Framework Level Architecture



Platform Libraries #1

1. Sensors Libraries
   - EMG
   - Pressure
   - Gesture
   - Gyro
   - Optical
   - Signal Buffering: facilitates data exchange among elements allowing async collection and processing of signal data
2. Actuators
   - Servo
   - Linear Actuators
   - Encoders
   - H-Bridge interfaces
   - ??? Other

Platform Libraries #2

3. Communications
   - Bluetooth
   - Wi-Fi
   - Cloud API: allows cloud integration
   - Protocols: facilitates data exchange between processing elements
   - Streaming: allows real-time data streaming towards cloud or host systems (signal offload for external storage/processing)

4. Algorithms/Signal Processing
   - Several processing algorithms are already available as source code and can be adapted/ported on the platform
   - Newly developed libraries (for platform specific tasks)
   - Finite State Machines or more complex logics to implement control logic (pre-configured platforms)

Possible Implementations

Framework Level View

e-Nable Bionic Platform Library

EMG | Pressure | Gesture / Gyro | Optical | Signal Buffering

Servo | Linear | Encoders | ???

Bluetooth | Wifi | Serial | Protocols | Cloud API | Streaming

Already avail libraries | Specific Algorithm Libraries to be developed | FSM & Logics

HAL – Hardware Abstraction Layer

Custom and Commercial Hardware boards and sensors

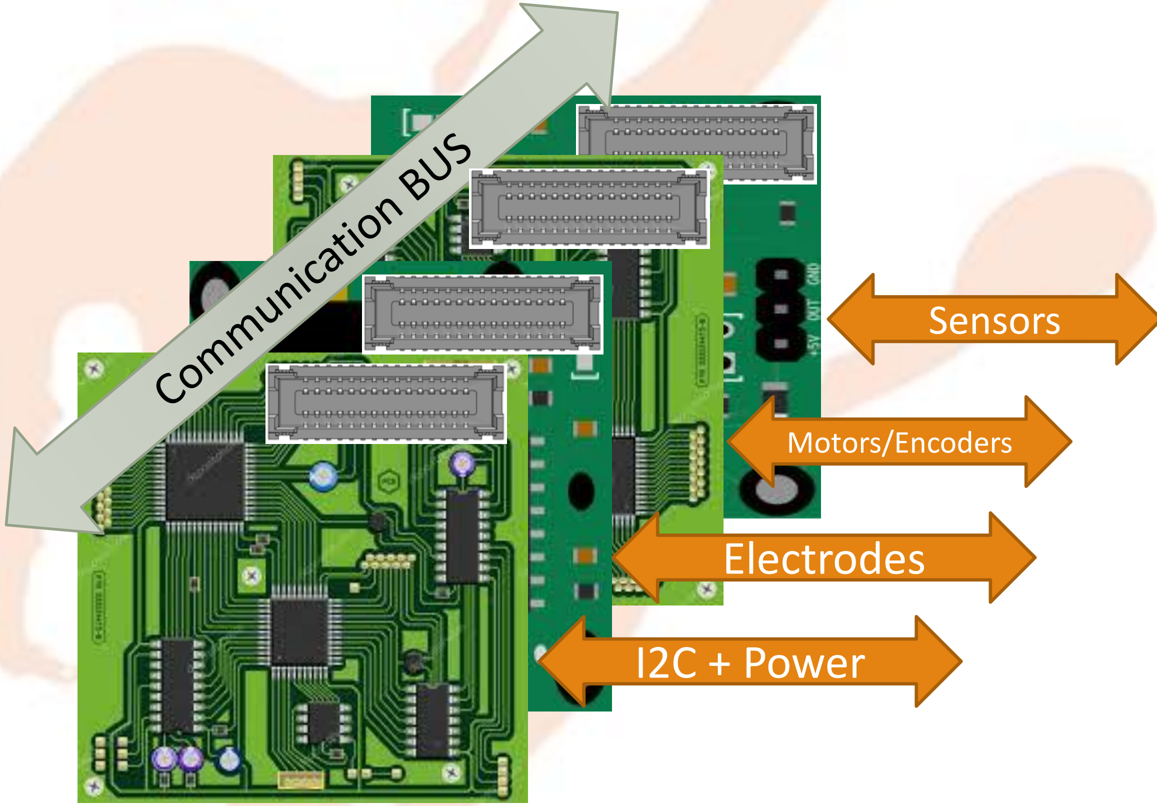
Hardware Modules

COLLABORATIVE PROJECT – SPECS ARE STILL A WORK IN PROGRESS

Hardware Modules - Concepts

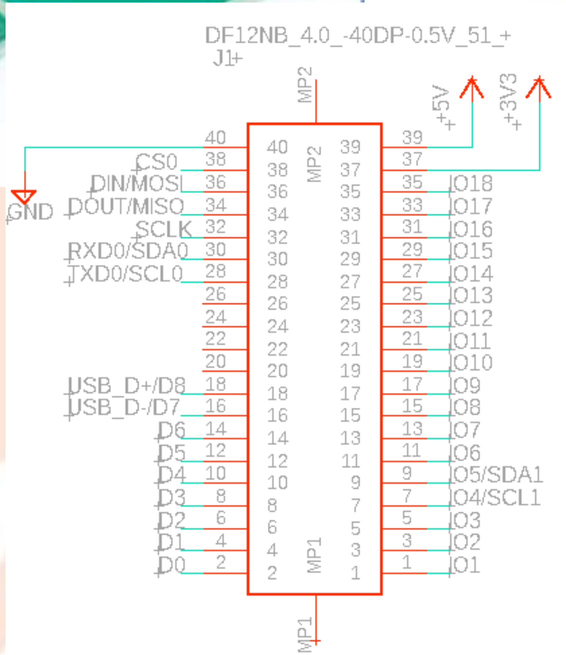
Opensource specialized hardware boards with:

1. Shared communication bus
2. Stackable PCB architecture support (allowing interconnection between PCBs), concept like the Arduino shields
3. Per board specific I/O
4. Small footprint
5. Optimized board architecture
6. Optimized power management



B2BCB - Board to Board Communication Bus

1. 40 pin connector
2. 4/6 mm height
3. Rated up to 300mA, 50V
4. Signals:
   - +5V power (in and out)
   - +3.3V power (in and out)
   - GND
   - 2 I2C (SDA0, SCL0, SDA1, SCL1)
   - SPI with 4 CS
   - 9 Digital signals (D0 – D8)
   - 18 General purpose IO (PWM/ADC)
5. Guarantees modularity, extensibility, module reusability

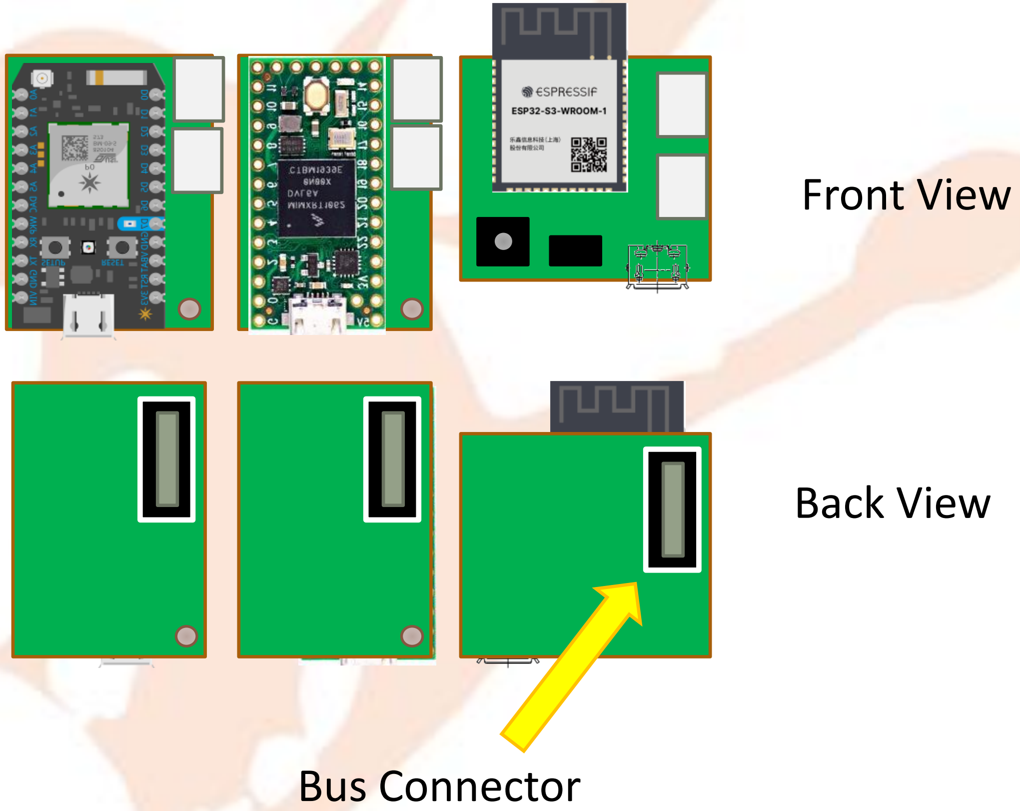


CPU Boards

1. B2BCB interface (female connector)
2. Multiple CPU support (e.g. ESP32, Teensy 4.0, Particle Photon, Arduino Nano BLE, etc.)
3. Powered from:
   - USB
   - External VCC
   - BUS
4. I2C external interface (Slave on bus, remote control)
5. Power external interface (Remote power)

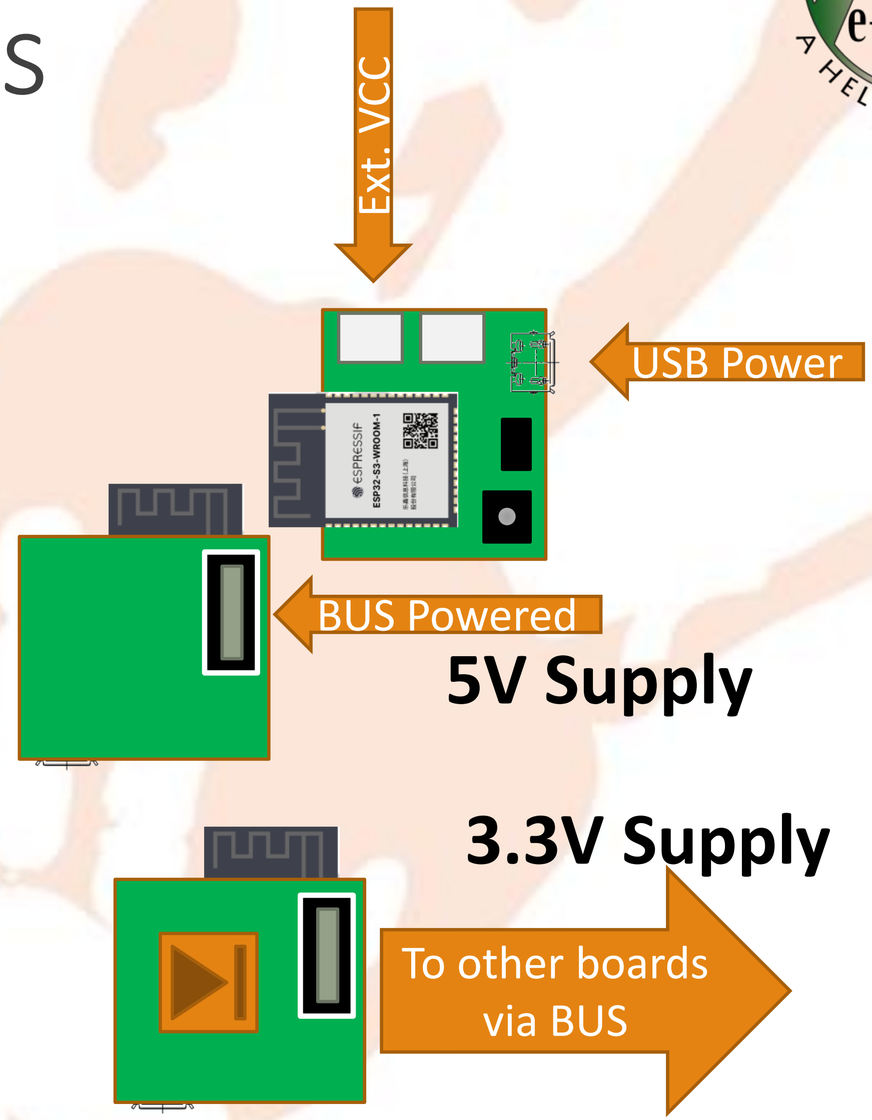
Design Status:

ESP32 S3 → Schematics 95%, Board 95%
Teensy Module → Schematics 90% (Standby)
Particle Photon → Schematics 90% (Standby)



CPU Board Power modes

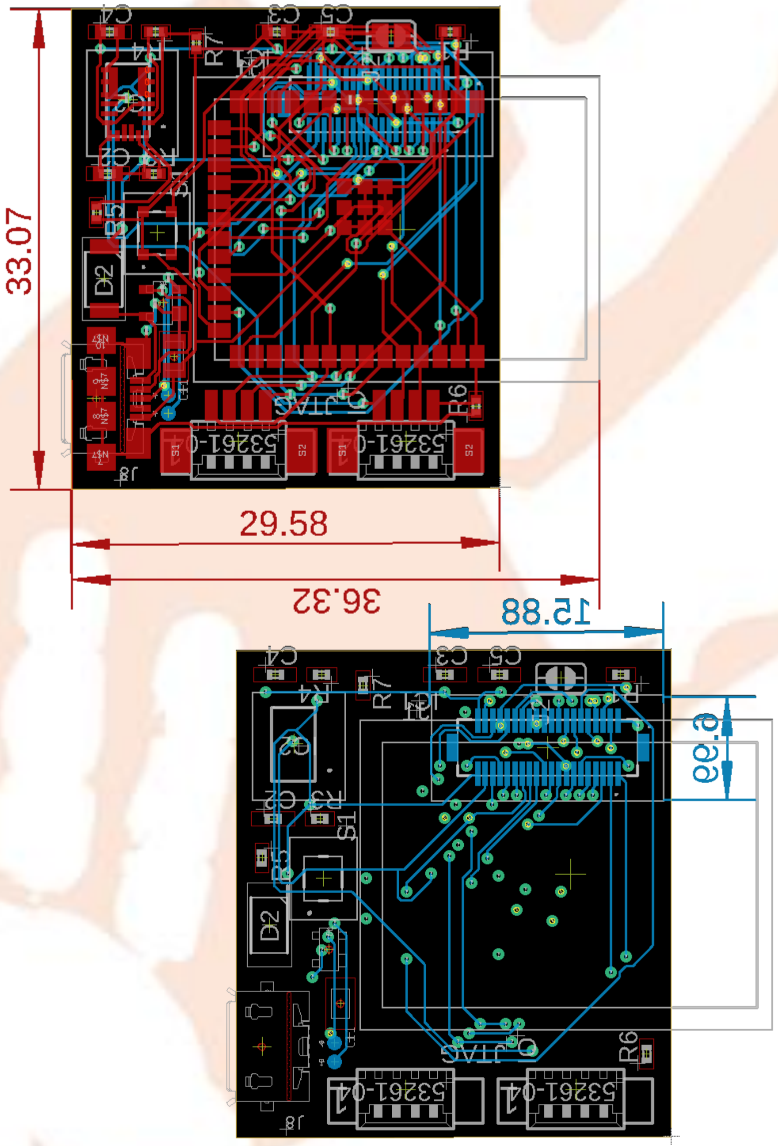
1. Operating voltages:
   - +5V (external power supply)
   - +3.3V (generated by internal step-down)
2. +5V Supply can be provided from multiple sources (alternative):
   - USB Power, board is powered from USB connector, +5V supply is extracted from USB interface, useful for development phase (programming/debugging)
   - External VCC, board is powered from +5V lines on the external I/O connector
   - BUS Power, +5V supply arrives from B2BCB (inter-board bus, typically from another board)
3. +3.3V is generated internally and feed also on the B2BCB (to provide +3.3V voltage to other boards connected on B2BCB)
4. USB Current Protection (prevent flow back of current from USB connector)



ESP32-S3 CPU Board

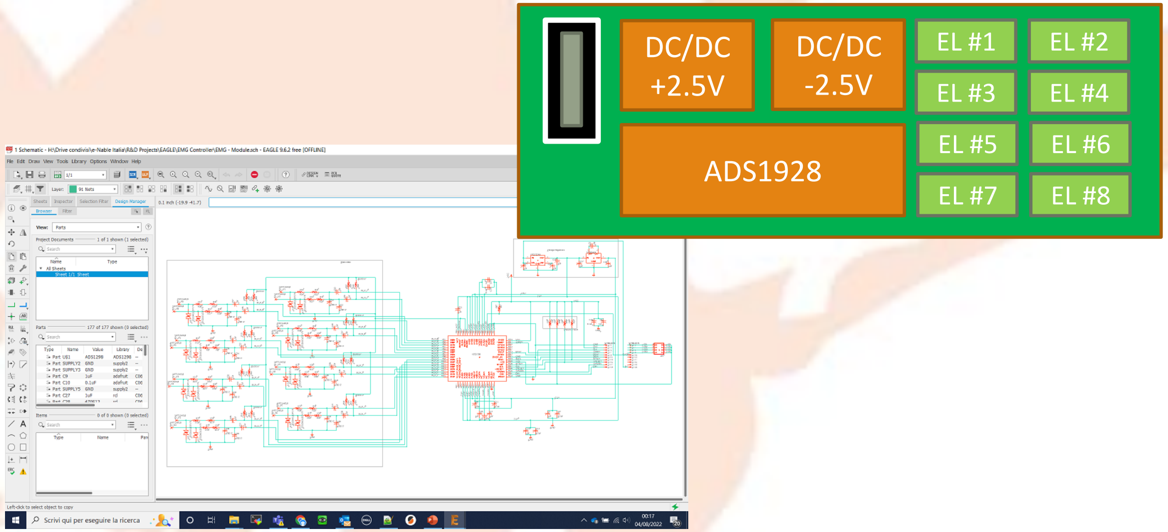
1. ESP32-S3-WROOM-1-N8R2 module:
   - Dual-core 32-bit LX7 microprocessor @ 270 MHz
   - 2 MB PSRAM
   - 8 MB SPI Flash
   - PCB Antenna (Wi-Fi, Bluetooth LE)
2. Peripherals: 2 I2C, 1 SPI, 36 GPIO
3. External connectors:
   - USB-C
   - Power-in +5V
   - I2C
   - JTAG
4. Software support:
   - Arduino
   - TensorFlow lite
   - Platform IO

Design Status: Schematics 95% , PCB 95%



EMG Boards

1. Passthrough B2BCB interface (both male and female connector)
2. 2, 4 or 8 Electrodes support (Biosensing ADC TI ADS129x)
3. Low Power: 0.75 mW/channel
4. On board voltage references (TBV)
5. SPI interface (CPU control bus)
6. Daisy-Chain configuration (allows up to 10 boards with 8 channels @ 24bit sampling at 2KHz, CPU bus SPI running at 4Mhz)



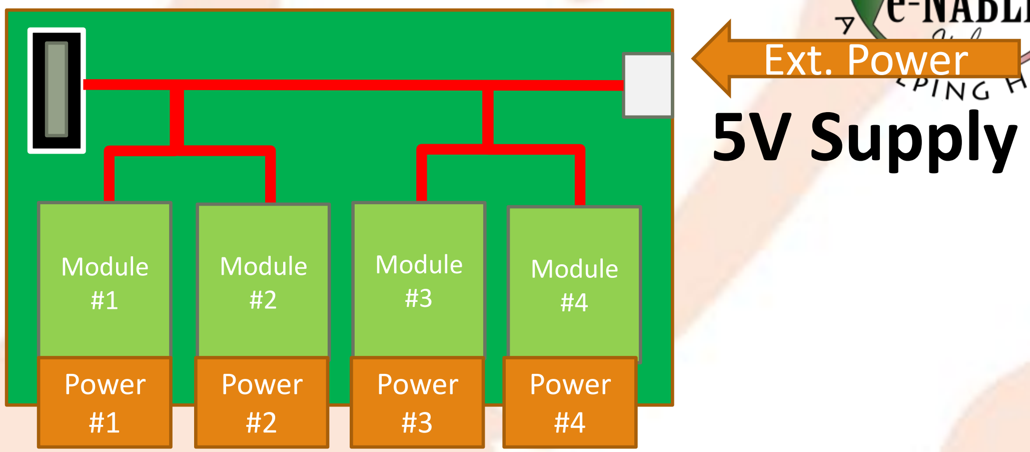
Design Status: Schematics 90% , PCB 80%

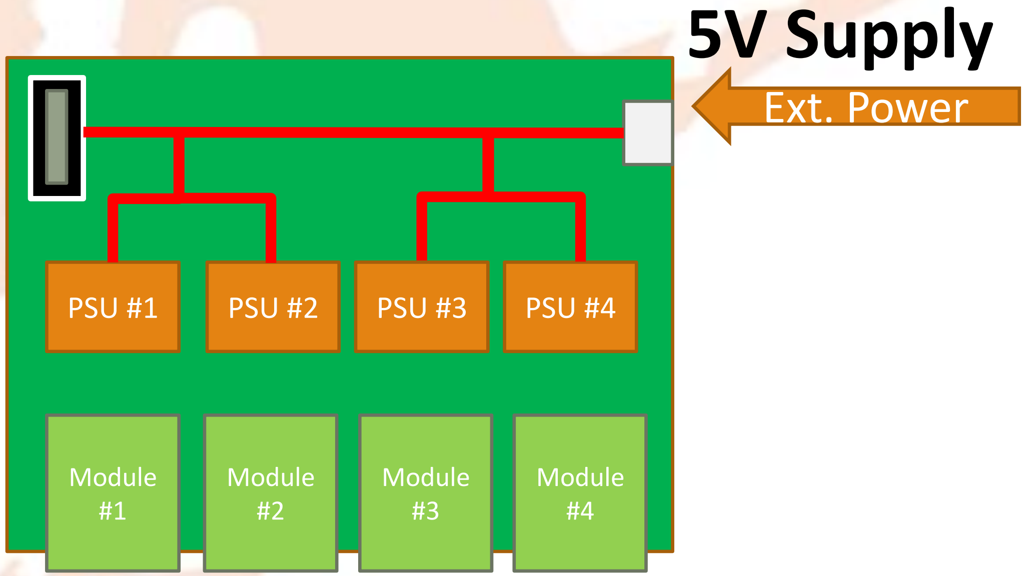
Main Board

1. Passthrough B2BCB interface (both male and female connector)
2. Powered from external VCC only, provides +5V on the BUS
3. Up to 4 driver modules, each controls up to 2 motors/actuators (max 8 motors/actuators)
4. Internal power distribution, two options:
   a) 4 PSU with programmable voltage level, one for each driver module
   b) each expansion module generates its voltage level from +5V

Design Status:

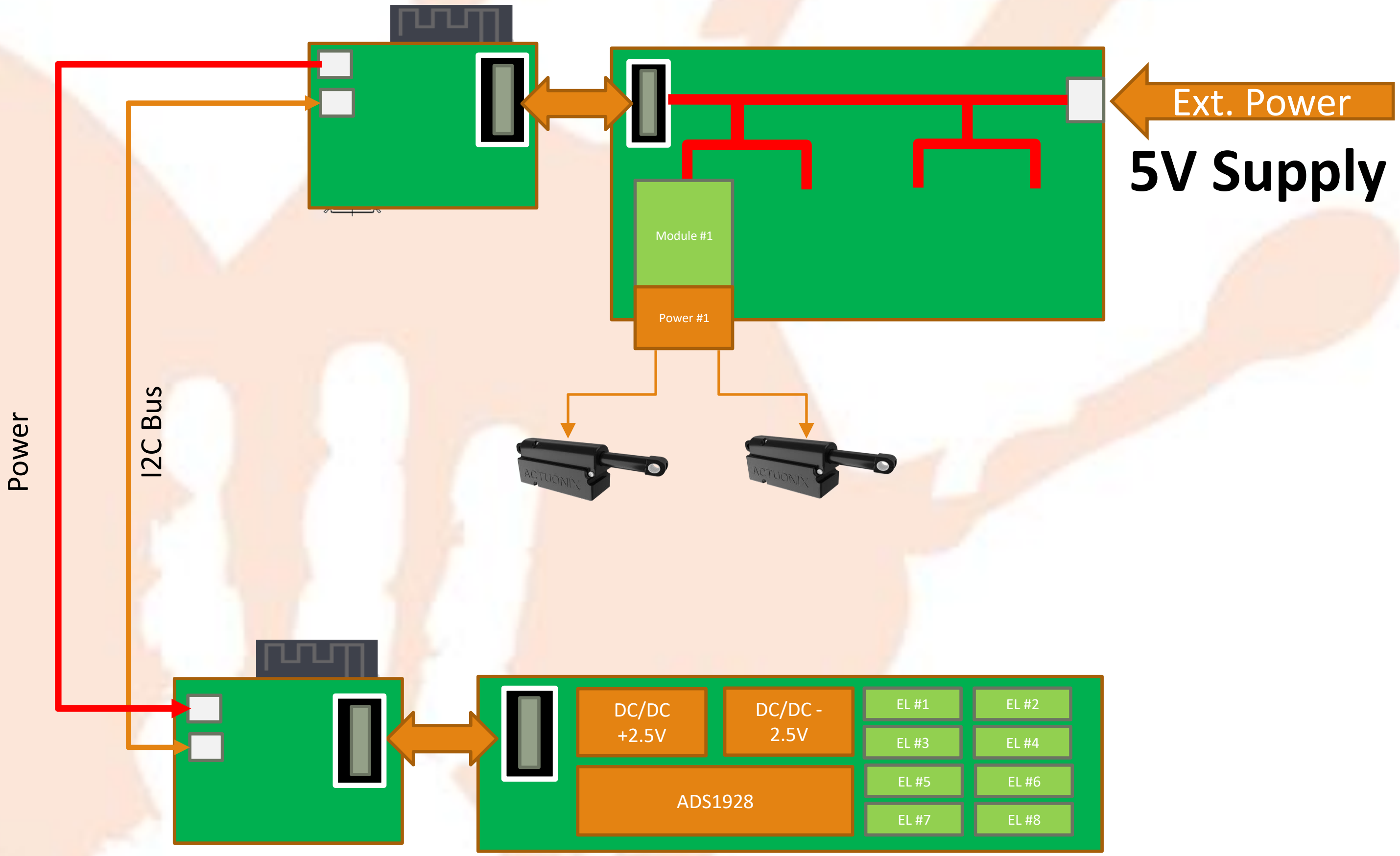
- Main board: Schematics 95% (version 4.b)
- Motor Board with 2 Actuonix Linear Actuators: Schematics 90%





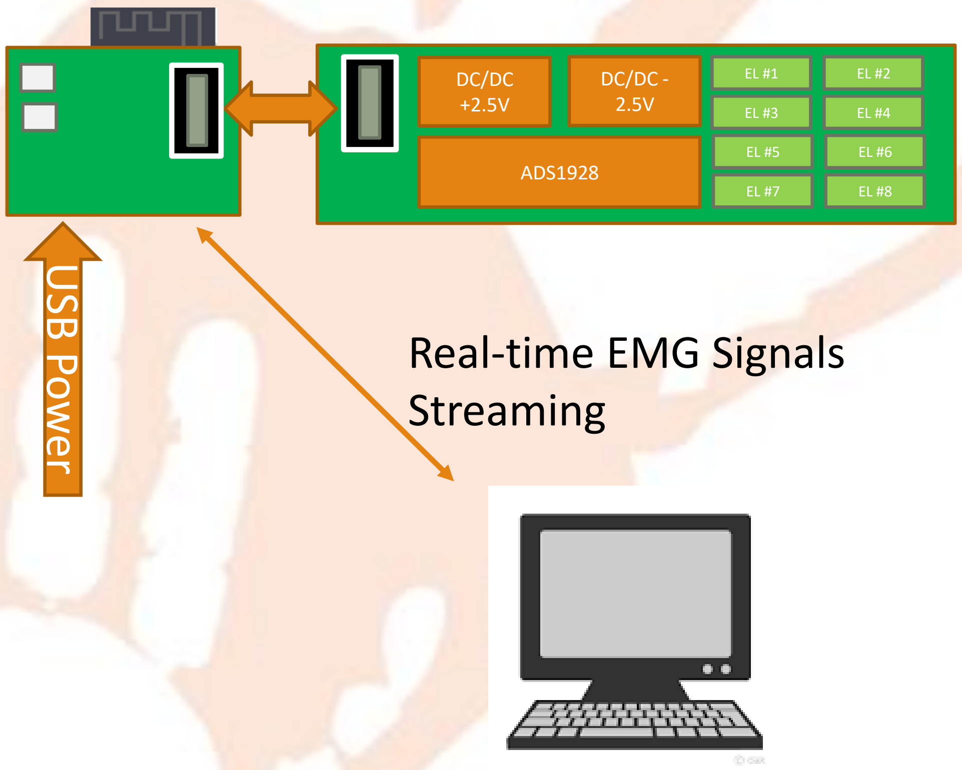
Example #1 – Simple EMG Controlled System

1. Boards:
    - 2 CPU boards
    - 1 Main Board
    - 1 Driver Module (Actuonix)
    - 1 EMG Board
2. Features:
    - Control of two linear actuators
    - 8 EMG Signal Acquisition



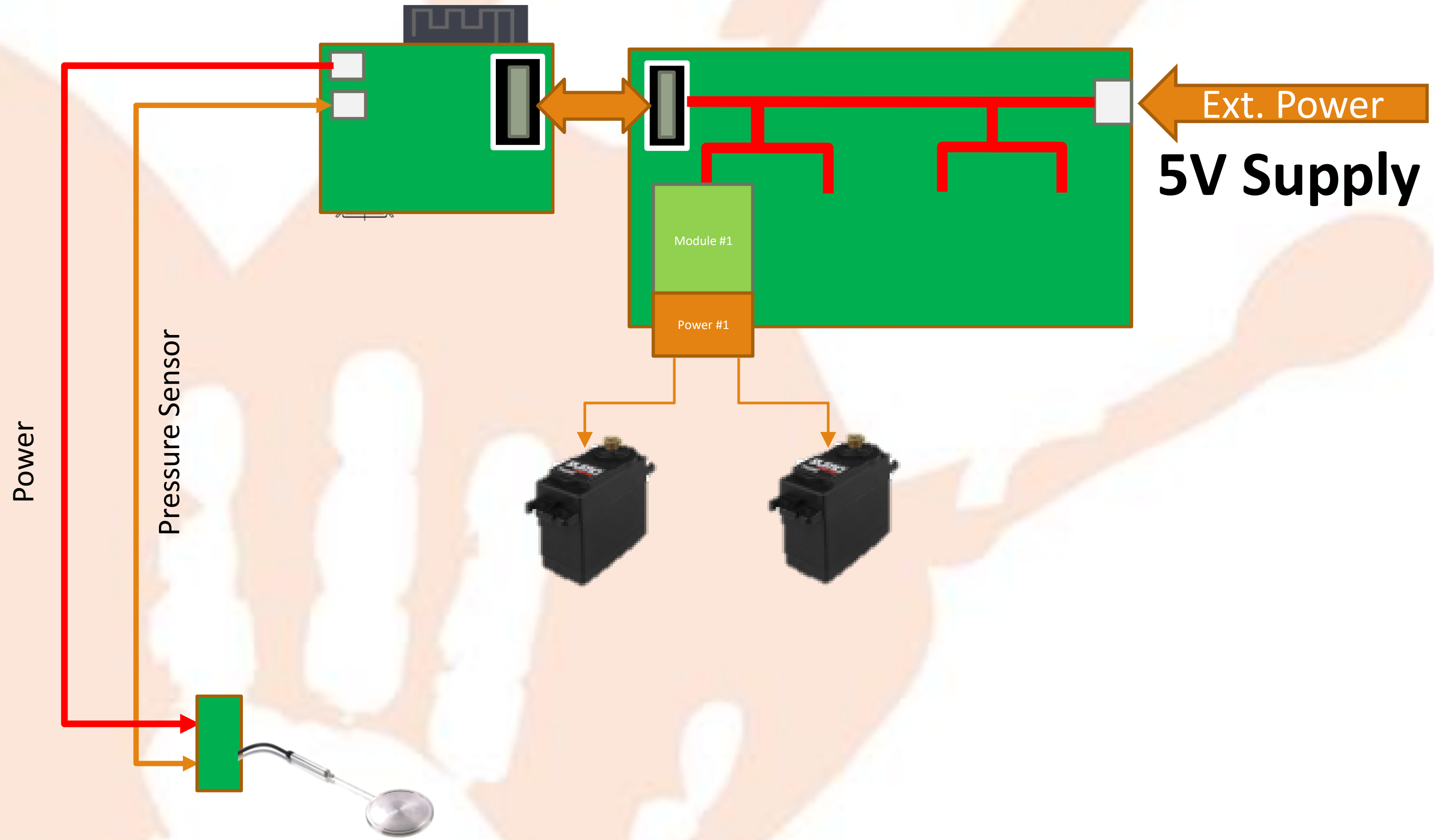
Example #2 – EMG Acquisition/Classification System

1. Boards:
    - 1 CPU boards
    - 1 EMG Board
2. Features:
    - 8 EMG Signal Acquisition
    - Realtime signal streaming via Wi-Fi
    - Host signal processing



Example #3 – Pressure Sensor + 2 Servo

1. Boards:
   - 1 CPU boards
   - 1 Main Board
   - 1 Driver Module (Servo)
2. Features:
   - Control of two Servos
   - Pressure sensor



Project Phases

Design

Design Review

Prototyping

Validation

Challenges (and Risks)

- ## Community Acceptance

  The projects can become valuable if they are accepted from the community and deliverables are effectively used by volunteers, students, etc.

- ## HW Supply chain

  We plan to build a small number of boards for validation and for development purpose, we are not a manufacturing company neither a commercial company at all, the project is released as open source and a commitment for hardware/board production is not in our scope.

Challenges (and Risks)

- Complexity

  The projects it selves are born to manage the intrinsic complexity of this kind of technology to allow also people with not specific skills in electronic and software development to deal with them, anyway it carry a great complexity and it is required a huge work to achieve the promised results, if we rely only on our resources, I suppose we'll be able to achieve only a part of the result

A collaborative approach is the only way to overcome risks and to draw a successful path

Thanks

For more information please contact: alberto@e-nableitalia.it or info@e-nableitalia.it