Unlocking Software Defined Power Electronics with Zephyr

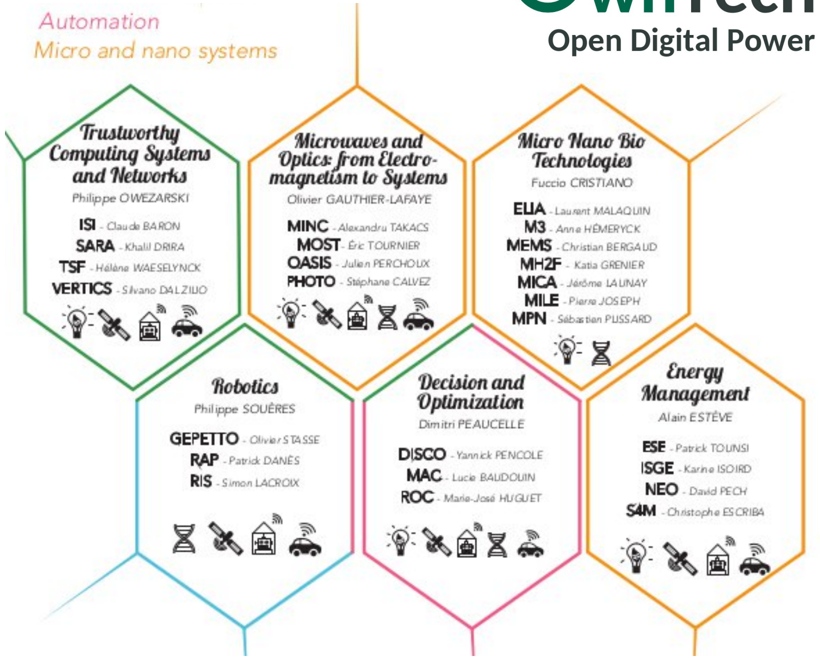
Concepts and Study case

Le LAAS-CNRS

Zephyr™ Project Developer Summit

OwnTech Open Digital Power

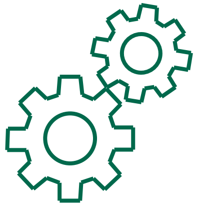
The LAAS-CNRS is the largest research laboratory in France. Strong of 300+ researchers, its field of research extend from robotics to renewables.



Dr. Luiz Fernando Lavado Villa received his PhD in power electronics for renewable energy management from the University of Grenoble in 2013.
He joined the University of Toulouse as an associate professor in 2014, where he has been conducting his research on software defined power electronics and modular energy management systems ever since.

Mr. Jean Alinei is a product engineer at CNRS. He is leading the technical development of software defined power converters with Dr Villa. Prior to joining the LAAS-CNRS, Jean worked in the renewable field for major companies such as the french utility EDF and GE renewables.
Mainly focused into making innovative ideas a reality, Jean has a dual curriculum in product engineering and innovative technology management, he received his M.Eng degree from Grenoble INP in 2018.

The OwnTech Project



Demand for power electronics is continuously growing while there is no technological mean to streamline training and foster innovation

OwnTech proposes a community-based compact, versatile, open-source and low cost technology for learning and fast prototyping power electronics.

OwnTech.org

Outline

- What is Power Electronics?
- Software Defined Power Electronics and OwnTech’s study case
- Lessons learned

Outline

- What is Power Electronics?
- Software Defined Power Electronics and OwnTech's study case
- Lessons learned

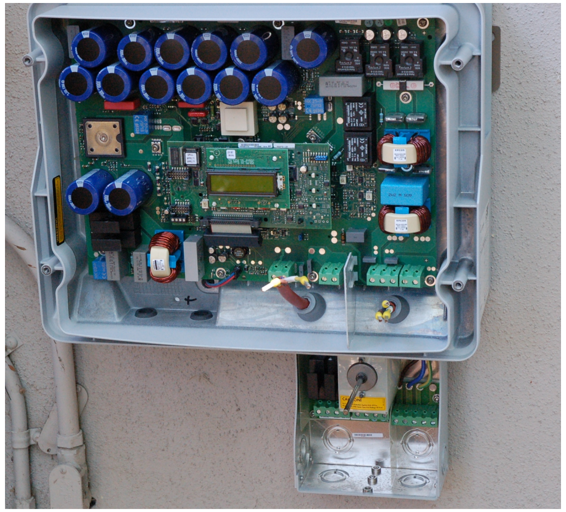
Zephyr™ Project Developer Summit

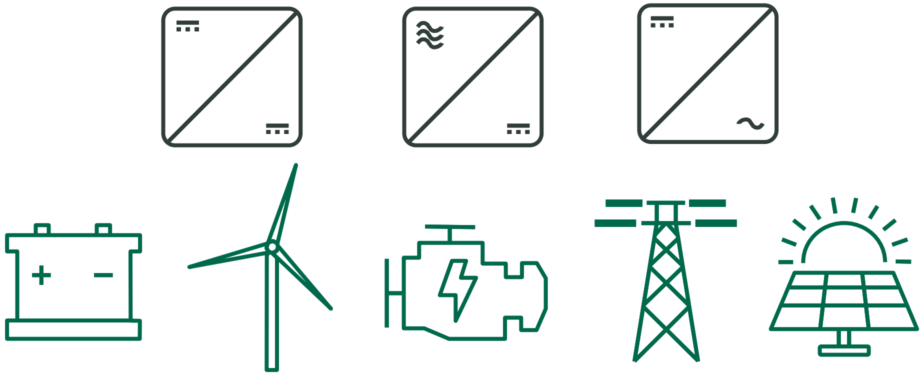
OwnTech Open Digital Power

The Great Energy Transition

What is power electronics?

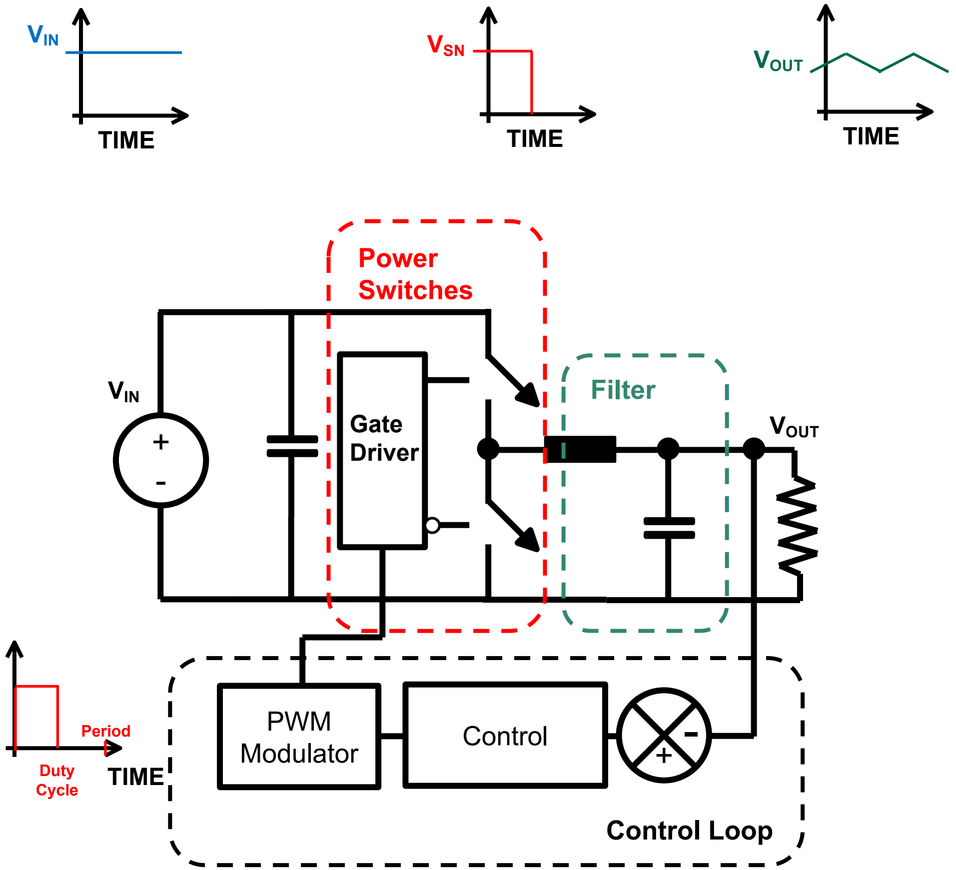
- Is a branch of electrical engineering
- Electronics that is made to handle power flows
  - Goes from mW to MW
  - DC-DC and DC-AC (1 or 3 phase)
- **Application-oriented design**
  - **Battery charging**
  - **Energy management**
  - **Motor control**
  - **Grid connection of renewables**





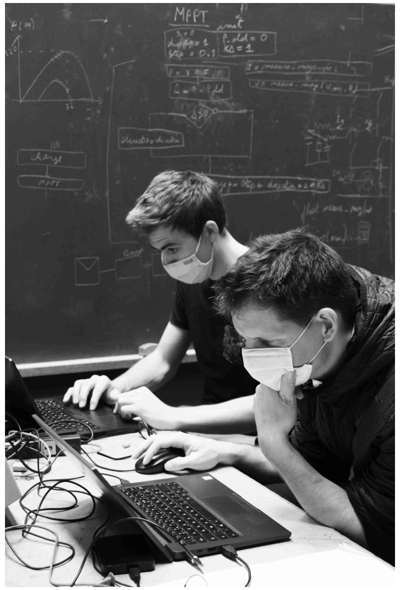
How does power electronics work?

- Power converters regulate voltages and/or currents in either DC our AC
- Regulation is done through control loops
- Control loops use measurements to regulate power flow
- Power flow is controlled via PWM signals sent to the power switches via gate driver circuits
- Output filters smooth out the output voltage and current
- **Real-Time is critical for stability**
- **The P.E. community is very analog solution oriented**
- **We believe the time is ripe for digitalizing power electronics**



Why digitalize power electronics?

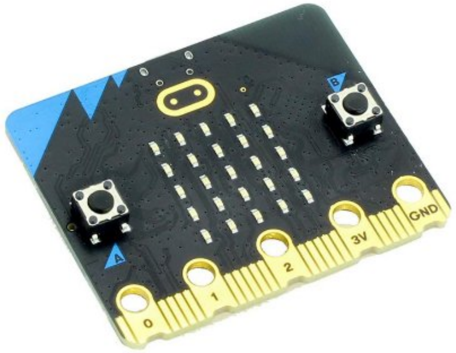
- Decarbonizing the economy will require more electricity
- More electricity will require more power electronics
- Digitalization has emerged as a means to reduce time-to-market and standardize the development of solutions
- **Energy transition will be greatly accelerated if power electronics becomes more standardized, modular and digital**
- **Where we are is a case in point**



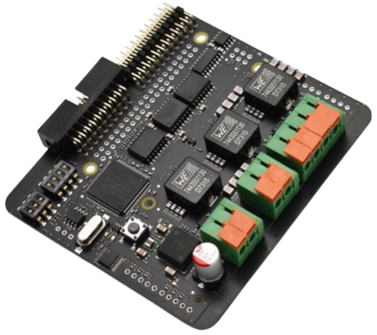
Digitalization examples

Modular hardware with standard connections for new modules

Hardware capable of many different communication outputs



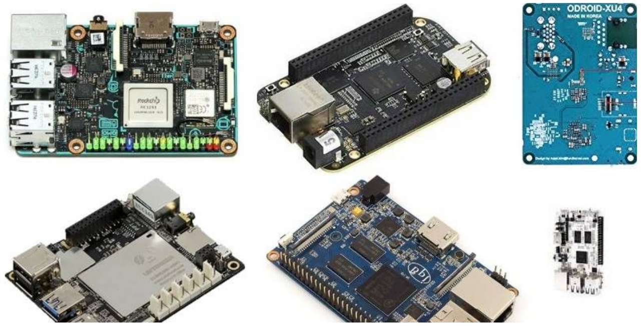
Simple to use led-based hardware

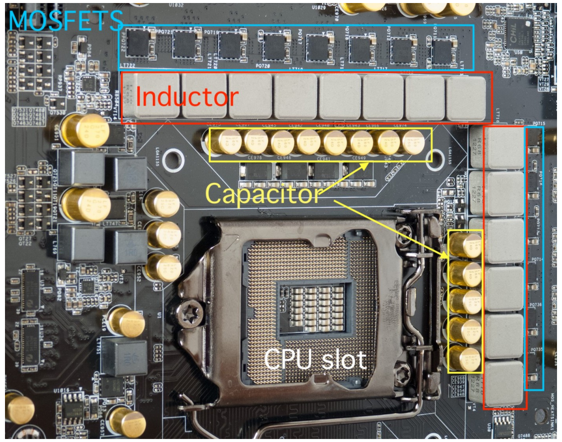


Reprogrammable Power Hardware with Raspberry-Pi based interface

Digitalization state-of-the-art

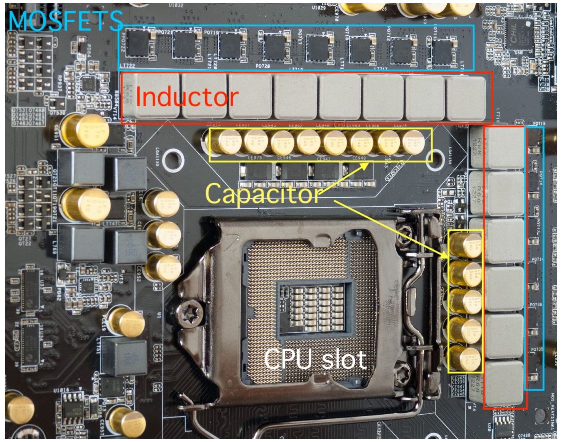
- Two examples give a good idea of the state of the art of digitalization
- IoT
  - Fully integrated software stack
  - No more bare metal
- Digital Power controllers
  - Bare metal
  - Ultra-flexible micro-controllers with specialized peripherals
  - High integration between peripherals
  - Specialized for real-time control
  - Integrated fault-management and telemetry protocols
- **The Problem: On which target can we combine the abstraction API approach of IoT with the performance of dedicated analog peripherals ?**





Digital Power Controllers examples

- Multiple silicon vendors have Digital power SoC in their portfolios
- Some are Function defined
  - Multiphase converters for CPU power delivery



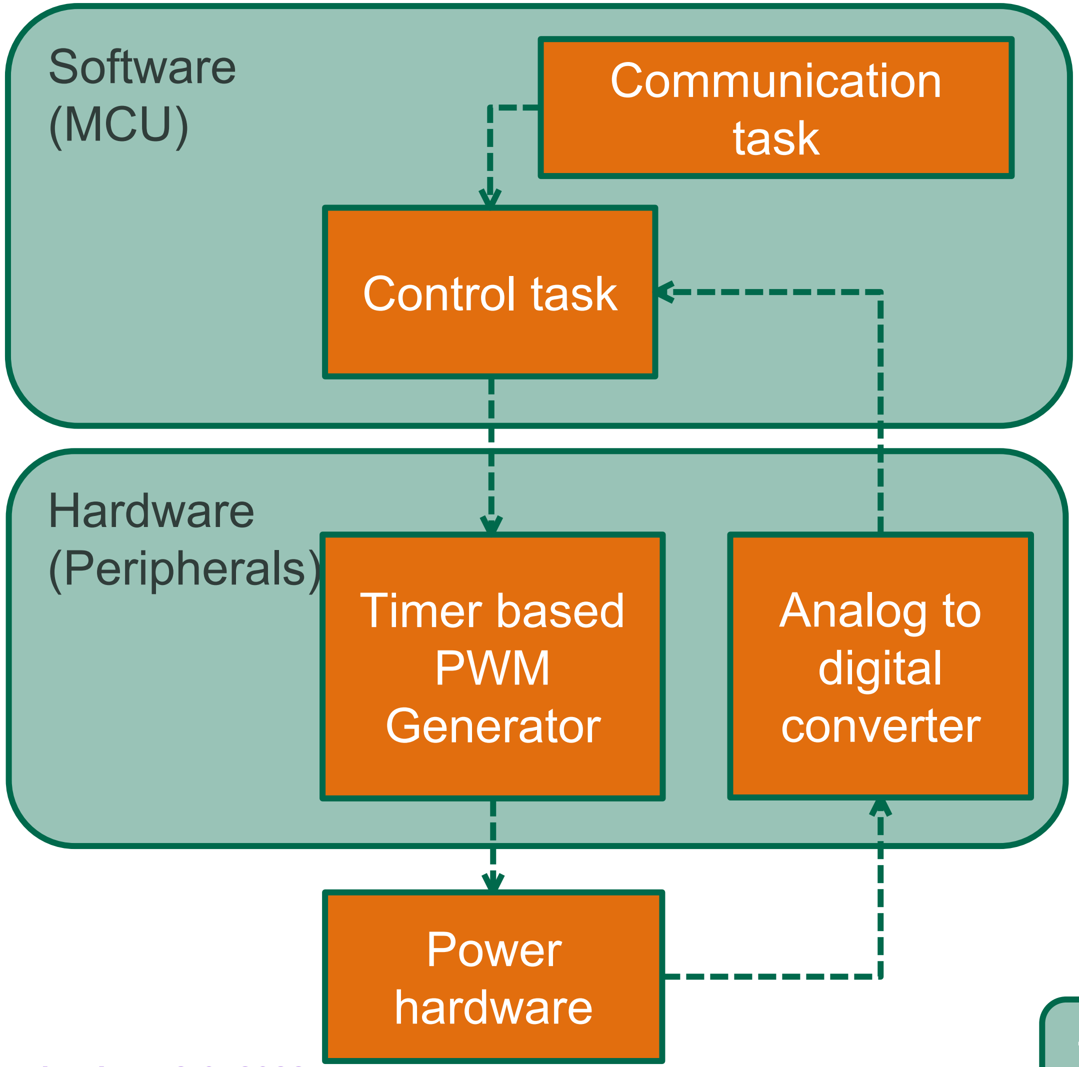
- Provide Telemetry
- Optimize control laws
- Tied to a specific use case

- Some are reprogrammable as they can cover multiple power topologies.

- Digital power SoC do have
  - Dedicated High Resolution PWM generators
  - Dedicated Fast Analog peripheral
    - ADC, DAC and comparators
  - Dedicated Hardware for fast control computation
- Some require vendor specific languages and libraries

The Digital Power Electronics Stack

- The digital power electronics stack has two parts: Software and Hardware Peripherals
- Software handles communication tasks and control
- Hardware comprises PWM Generation and ADC operation
- Low level loop
  - The power hardware sends data to the ADC and receives the PWM from the Timer
  - The control task drives the PWM

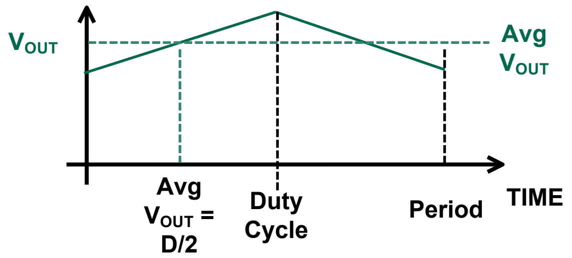


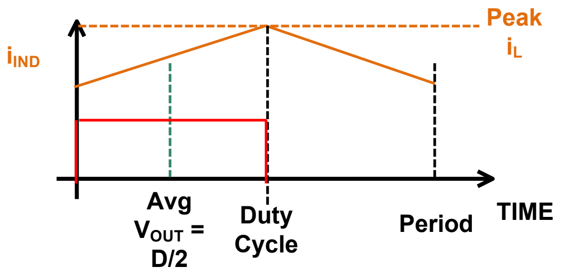
Outline

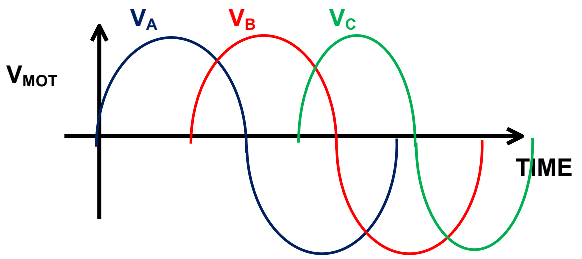
- What is Power Electronics?
- Software Defined Power Electronics and OwnTech Study Case
  - Understanding PE control
  - The SDPE stack
  - Presenting OwnTech's implementation
- Lessons learned and theoretical conclusions

Understanding P.E. Control - Overview

- Objective:
  - Deduce the main constraints on the digital power hardware
  - 3 main control schemes that covers 90% of the use cases
- Voltage mode
  - Average measurements
- Current mode
  - Average and Peak measurements
- DC-AC control
  - Average and external measurements

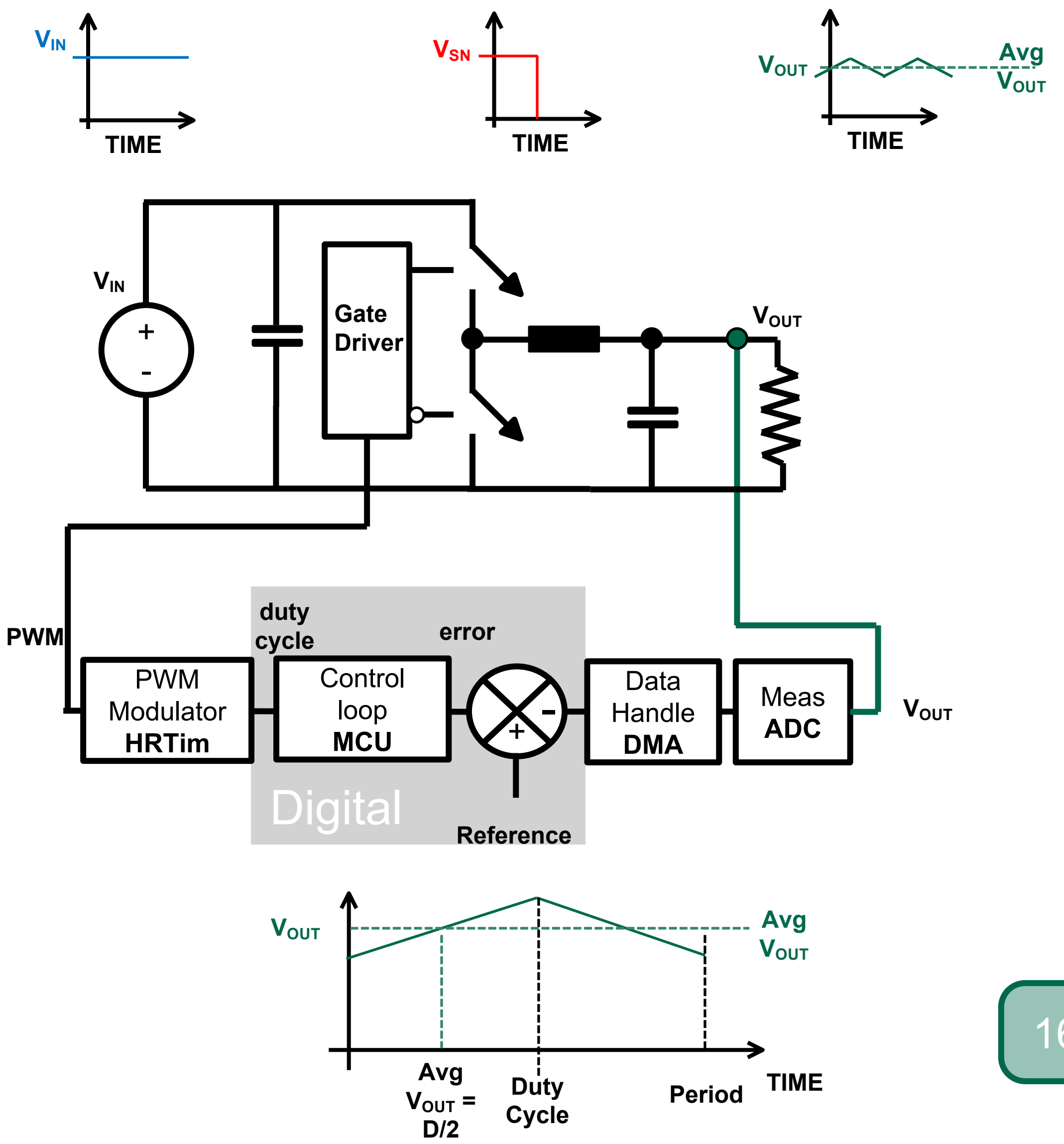






Understanding P.E. Control - Voltage mode

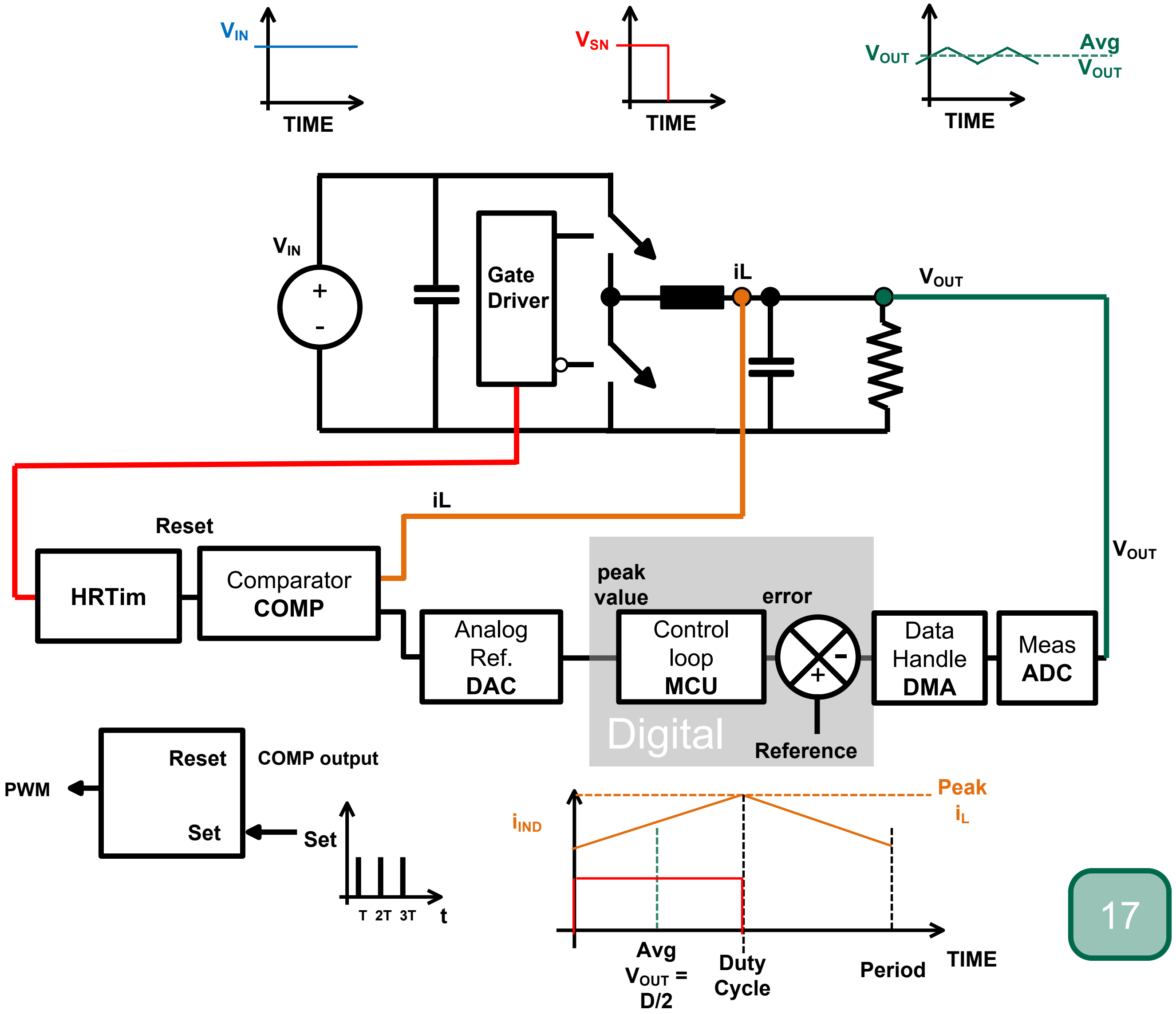
- Control of the duty cycle by the MCU
- Less hardware intensive
- Average measurement based control
- Adaptable control bandwidth
- Can be done with non-specialized micro-controller
- **Precision in the measurement trigger is necessary**



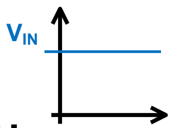
06/09/2022

Understanding P.E. Control - Current mode

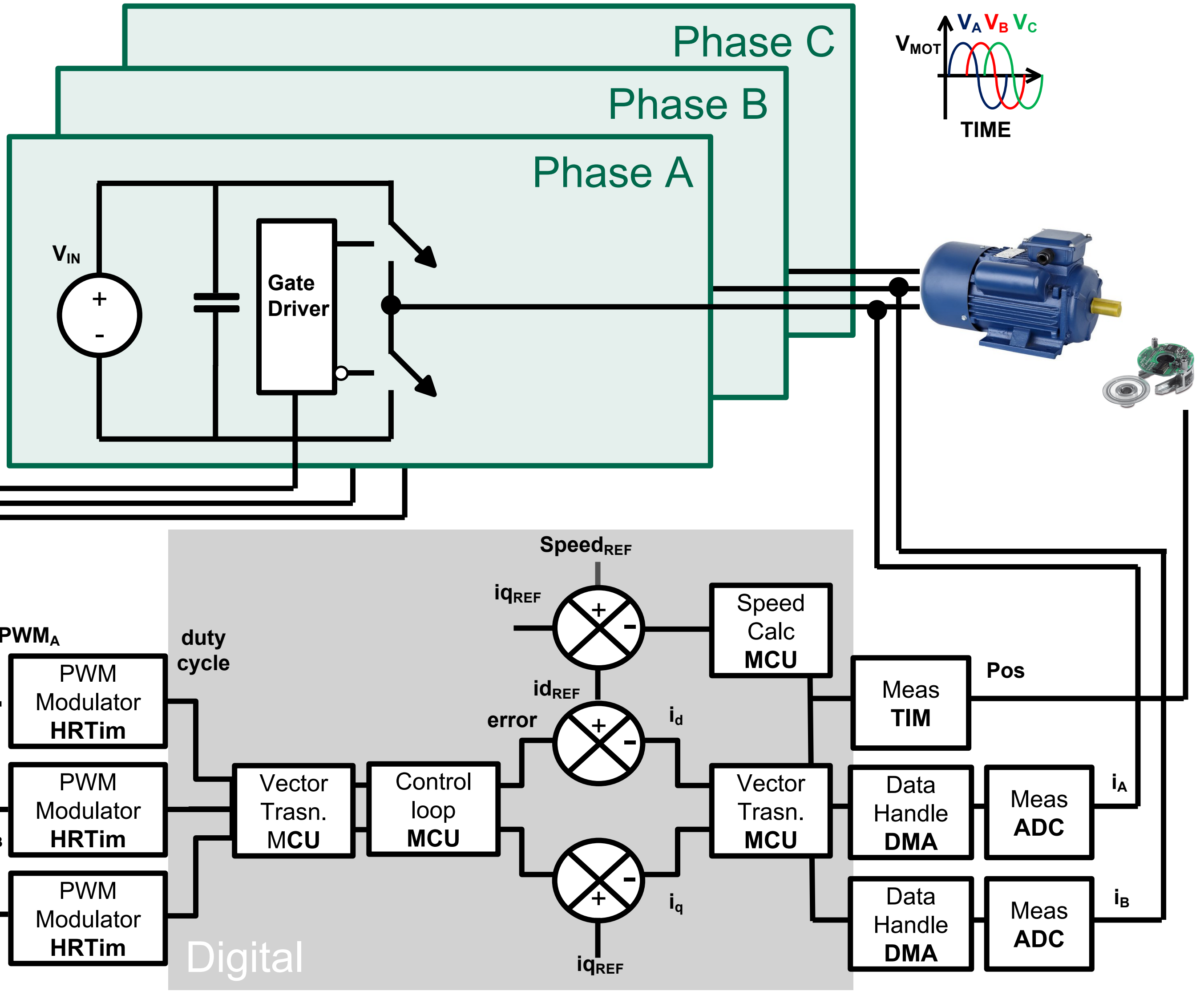
- Control of the duty cycle by a comparator
- Very hardware intensive
- Average and peak measurement based control
- Decoupled control bandwidth
- Provides safety in case of current spikes and overloads
- **Requires direct connection of the ADC, the DAC the COMP and the HRTIM**

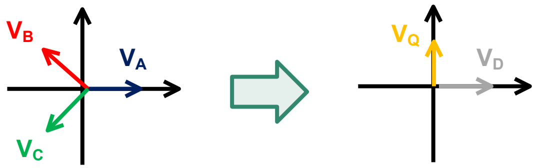


Understanding P.E. Control - DC-AC control



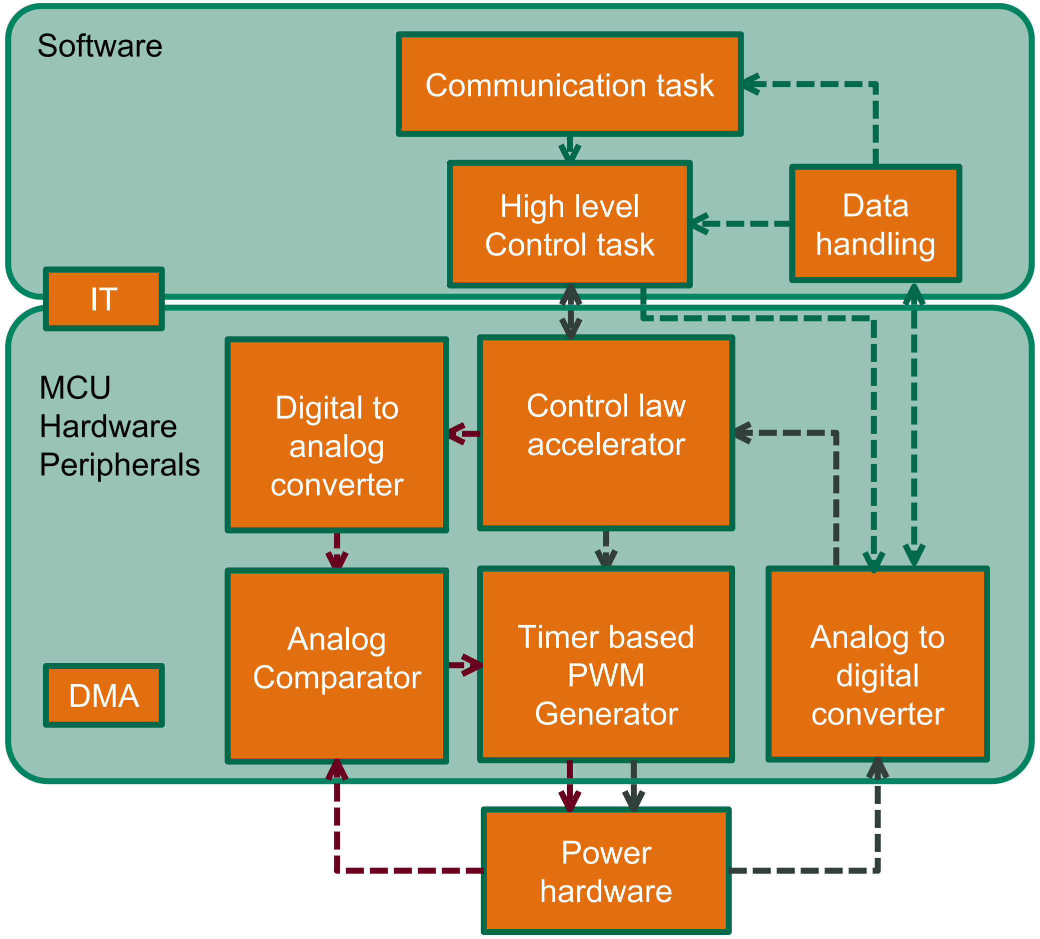
- Control of the duty cycle by the MCU
- Very calculation intensive
- Requires many measurements
- Average current measurement based control
- Control bandwidth limited by the motor
- **It may require external measurements**





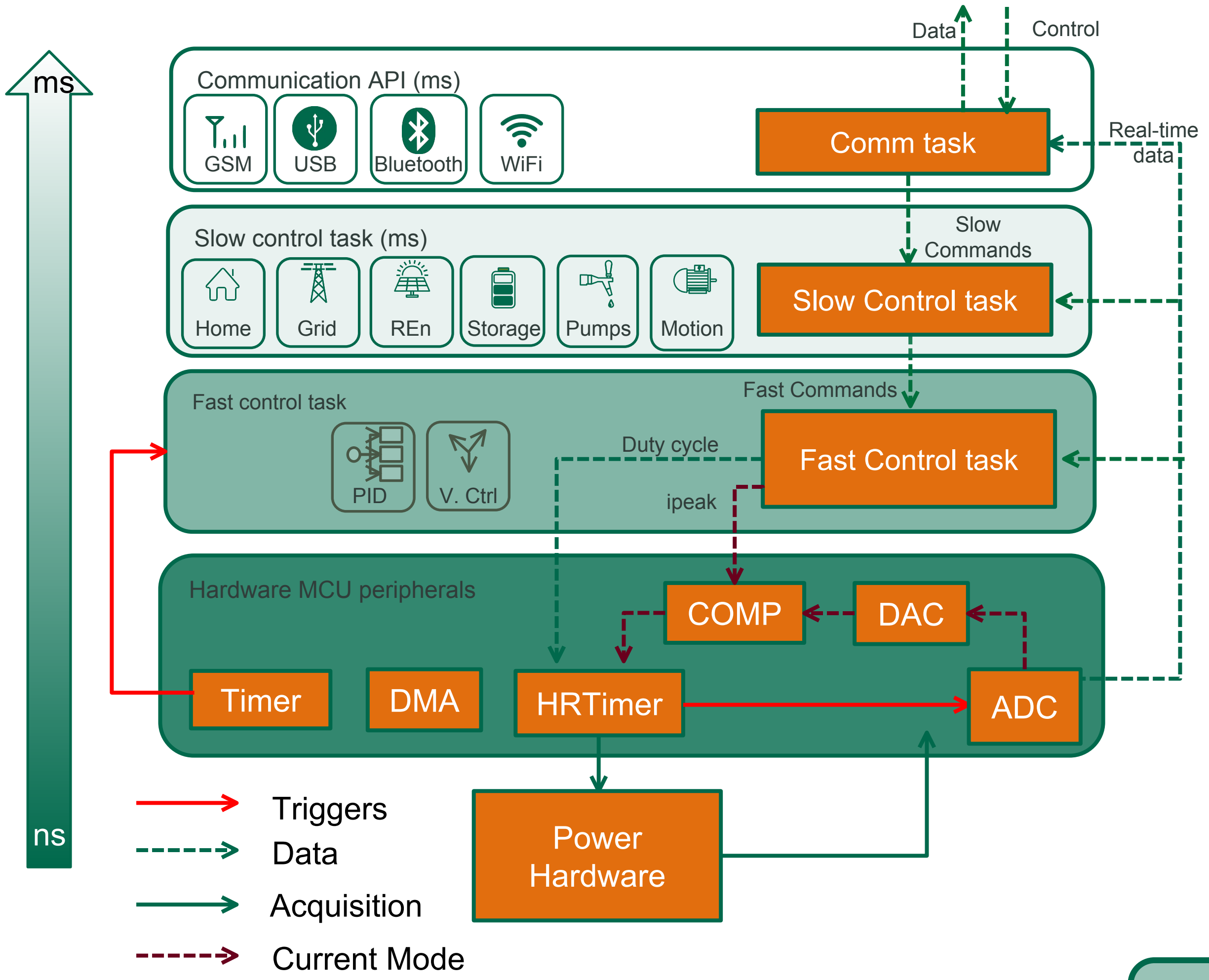
Understanding P.E. Control - Digital Hardware Overview

- Critical real-time needs
  - Fast control tasks
  - Fast acquisition
  - Fast communication needs
  - Very precise timers and PWMs
- Peripheral needs
  - Multiple triggers
  - Multiple DMA
  - ADC and PWM synchronization
  - **Dense and efficient Peripheral-to-Peripheral connection**
- Communication needs
  - Fast low-layer communication for decentralized control/synchronization
  - Slow high layer communication for hierarchical/centralized control



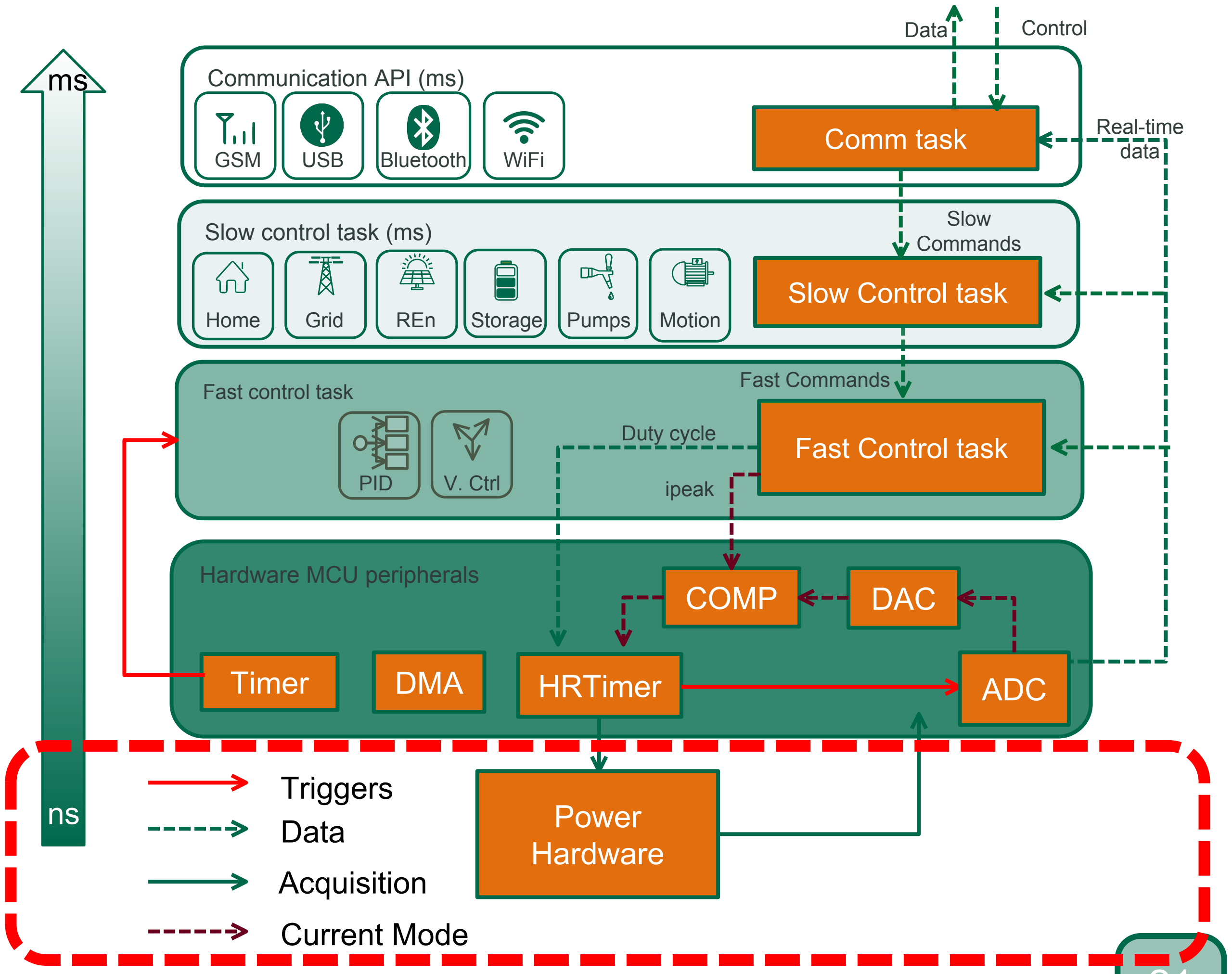
Our SDPE stack

- Communication task
  - (A)Synchronous Data uplink
  - (A)Synchronous Control Downlink
  - Flexible and compatible with multiple protocols
- Slow control task
  - Application oriented control task (e.g. state machine)
  - Updates references of the fast control
- Fast control task
  - Low-level reference tracking control
  - Receives data either for average or peak control
- Hardware MCU peripherals
  - Full integration of peripherals to achieve hardware acceleration for power electronics
  - Voltage or current control implementations
- Power Hardware
  - Flexible and reprogrammable power converter
  - Possibility of integrating external measurements



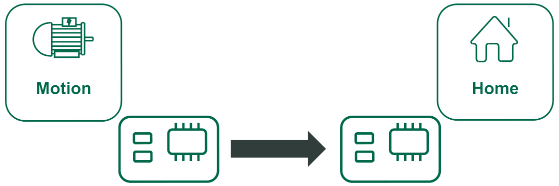
Our SDPE stack

- Communication task
  - (A)Synchronous Data uplink
  - (A)Synchronous Control Downlink
  - Flexible and compatible with multiple protocols
- Slow control task
  - Application oriented control task (e.g. state machine)
  - Updates references of the fast control
- Fast control task
  - Low-level reference tracking control
  - Receives data either for average or peak control
- Hardware MCU peripherals
  - Full integration of peripherals to achieve hardware acceleration for power electronics
  - Voltage or current control implementations
- Power Hardware
  - Flexible and reprogrammable power converter
  - Possibility of integrating external measurements

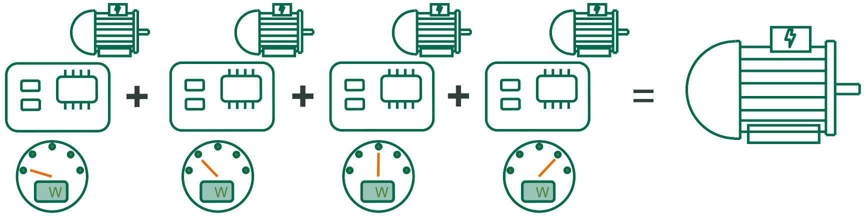


Power Hardware - Theory and specs

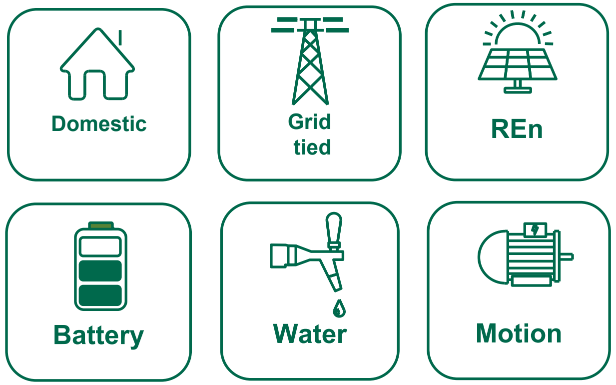
- Reprogrammable functions



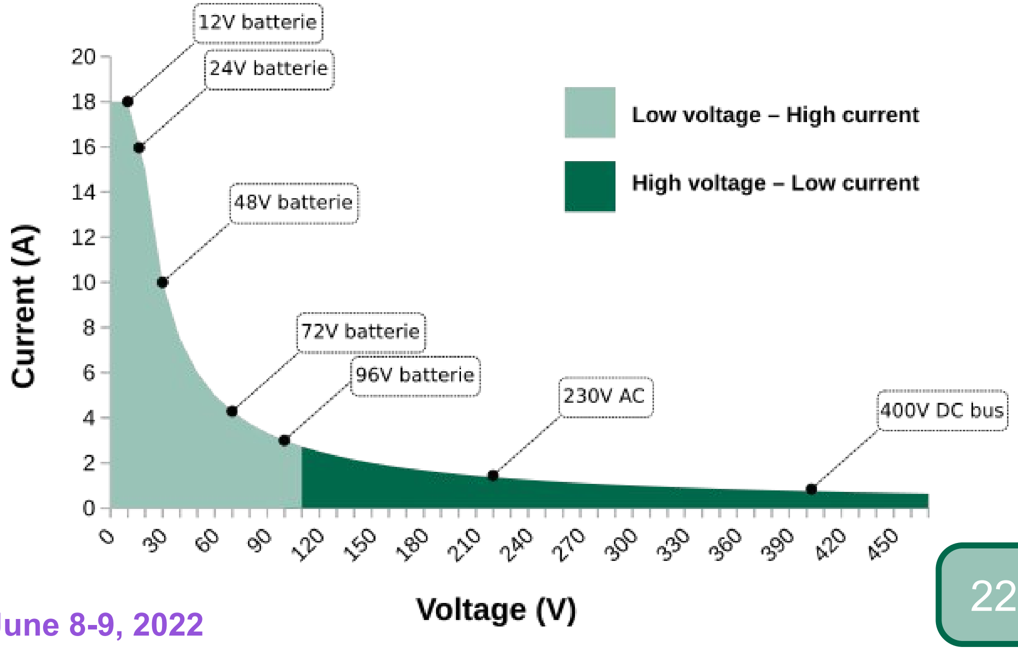
- Standard and Stackable Hardware



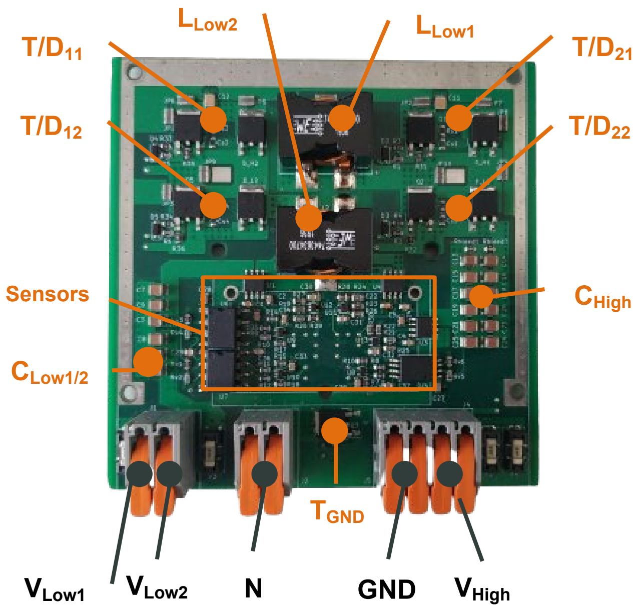
- Iso-power curve theory
    - Low-voltage functions
    - High-voltage functions



300W Iso power locus



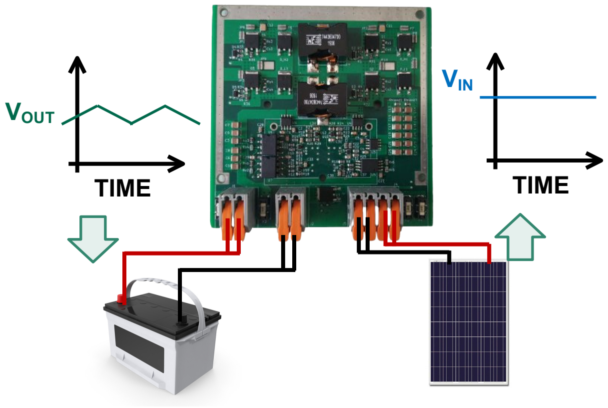
Chosen Power Hardware - 5 switch Dual Synchronous Buck

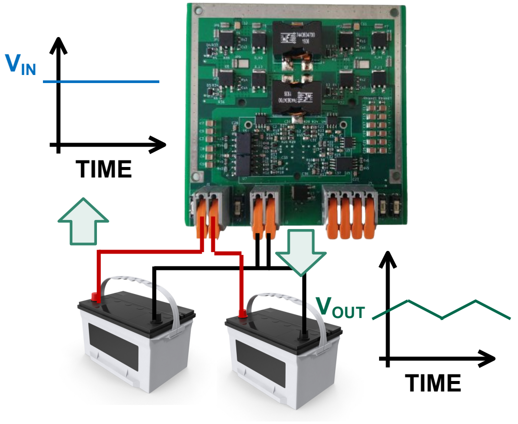


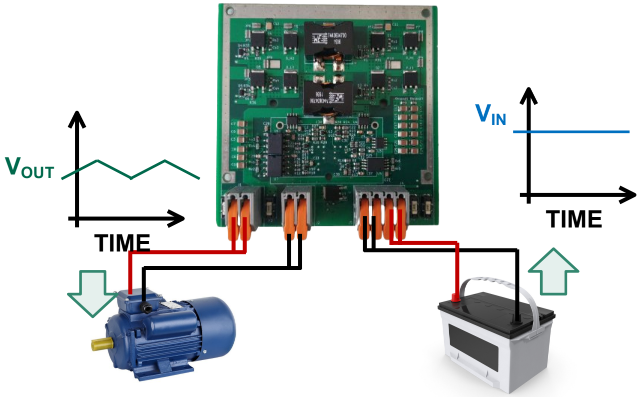
Characteristic	Details
Rated Power	300W
Number of Power Channels	Dual on Low-Side and Single on High-Side
Current Ratings	8A per channel 16A in parallel
Voltage Ratings	12V to 72V low side 40V and 110V high side
Functions	DC-DC buck or boost DC-AC single phase
Sensors	**V_{1Low}, V_{2Low}, V_{High} I_{1Low}, I_{2Low}, i_{High} Temp**

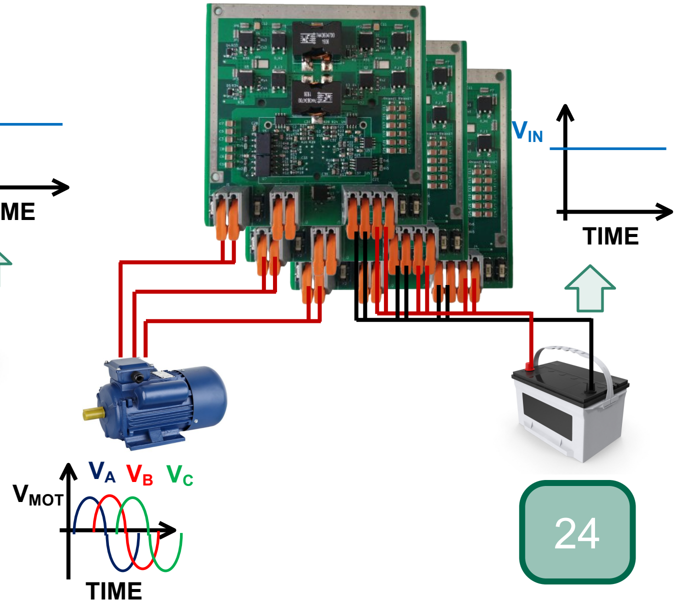
Power Hardware - Functions and Use cases

Conversion	Function	Vin	Vout	P	Vmax	Use Case
DC-DC	Buck interleaved	V_{HIGH}	$V_{LOW1/2}$	300	0.9 V_{HIGH}	Battery
	Boost interleaved	$V_{LOW1/2}$	V_{HIGH}	300	$4*V_{LOW}$	Battery
	Buck-Boost	V_{LOW1} OR V_{LOW2}	V_{LOW2} OR V_{LOW1}	150	$0.9*4*V_{LOW}$	Battery to Battery
DC-AC	1 phase	V_{HIGH}	$V_{LOW1/2}$	300	V_{HIGH}	Single phase motor control
	2 phase - 2 boards	V_{HIGH}	$V_{LOW1/2}$	600	V_{HIGH}	Stepper motor control
	3 phase - 3 boards	V_{HIGH}	$V_{LOW1/2}$	900	V_{HIGH}	Brushless motor control



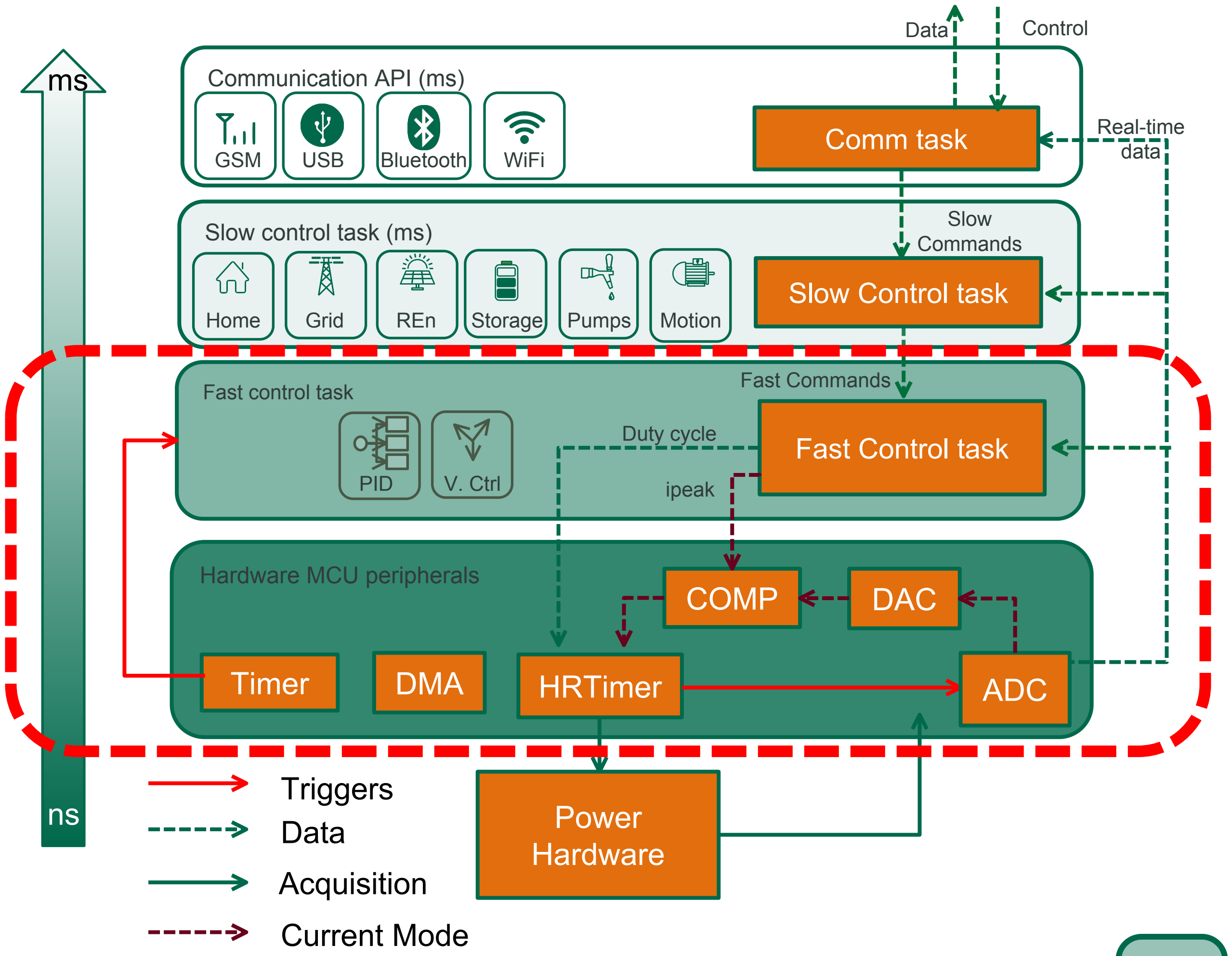






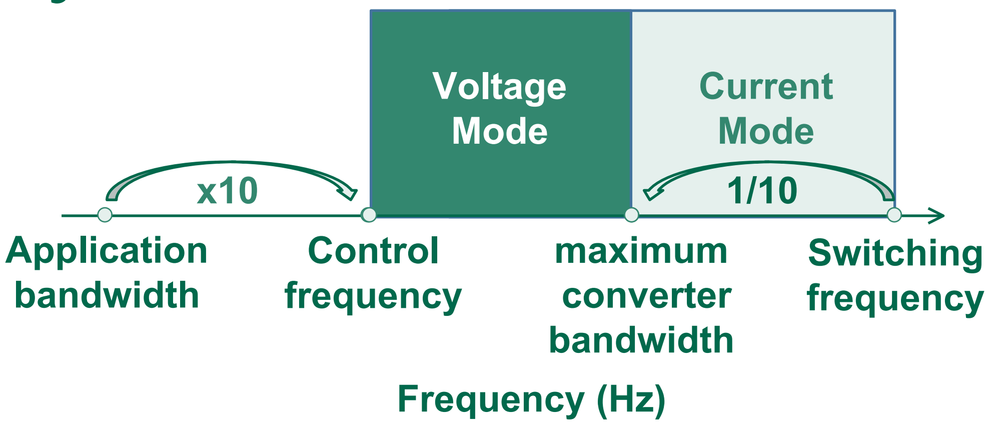
Our SDPE stack

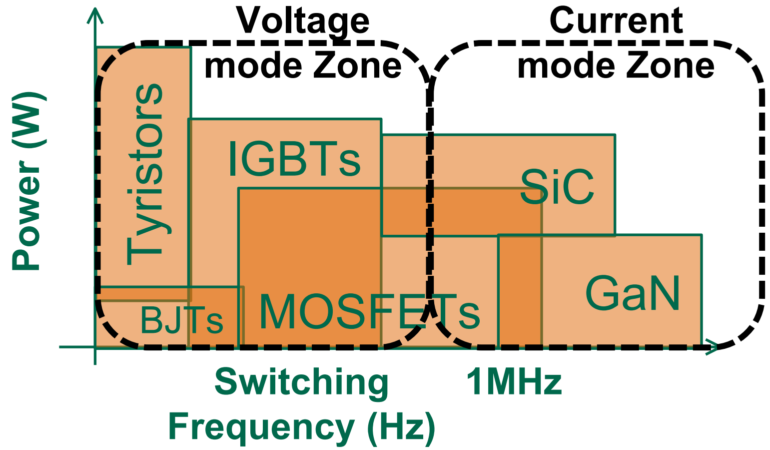
- Communication task
  - (A)Synchronous Data uplink
  - (A)Synchronous Control Downlink
  - Flexible and compatible with multiple protocols
- Slow control task
  - Application oriented control task (e.g. state machine)
  - Updates references of the fast control
- Fast control task
  - Low-level reference tracking control
  - Receives data either for average or peak control
- Hardware MCU peripherals
  - Full integration of peripherals to achieve hardware acceleration for power electronics
  - Voltage or current control implementations
- Power Hardware
  - Flexible and reprogrammable power converter
  - Possibility of integrating external measurements



Generalized Module Layer - Digital Power Bandwitdh theory

- There are 2 constraints for controlling a power converter
  - The maximum bandwidth of a converter is 1/10 of its switching frequency
  - The minimum frequency of the control is x10 of the application bandwidth
- Voltage Mode lies between both constraints
  - It is often associated with higher-order safety features for peak current
- Current Mode requires hardware acceleration
- Sensor bandwidth
  - Voltage sensor = 10 x application bandwidth
  - Current sensor = 10 x switching frequency
- Number of switching legs is a complexity factor
- Switch tech is linked with choice of uControllers

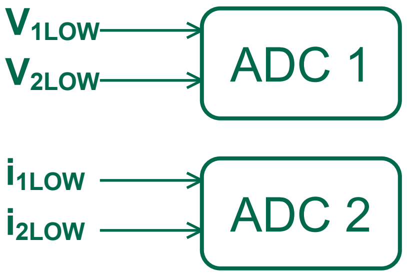




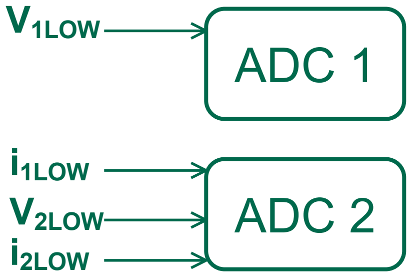
Digital Hardware - Data Acquisition Specs

- Reprogrammable ADC peripherals
  - ADC mode of operation depends on the control method
  - Flexibility in terms of data and control
  - **Need to accelerate calculations**
  - **Need for interconnection matrix**
- Multi-mode data acquisition
  - Measurements scheme depends on control objectives of the application
  - Application-level control may need intricate data measurements
  - **Need for flexible channel allocation**

Synchronous measurements for accurate power measures



Fast measurement for accelerated control



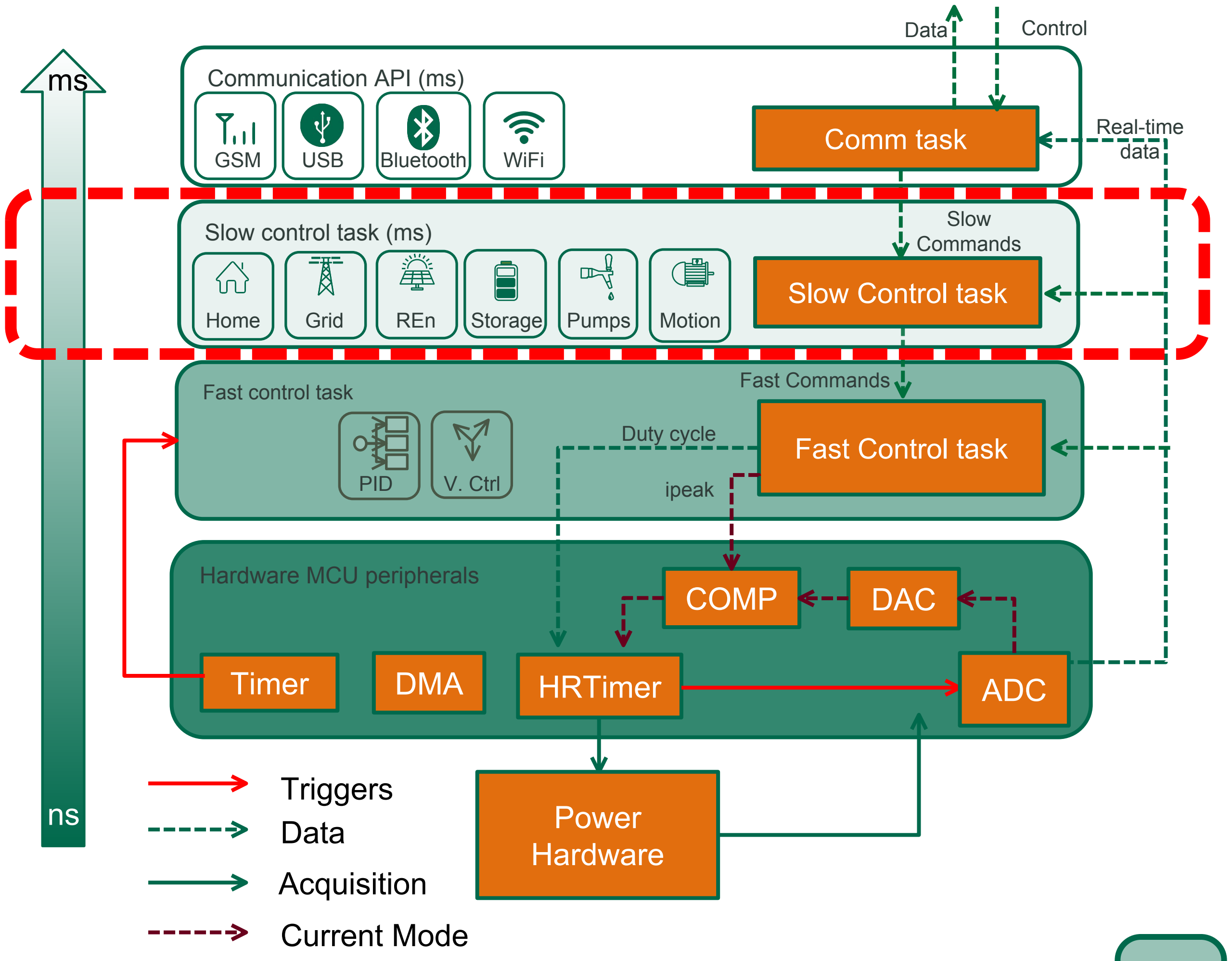
Slow measurements for Telemetry

Chosen digital power controller - STM32G474RE

Peripheral	Details
HRTimer	13000 points at 200kHz
2 ADCs	Reassignable ADC channels for flexible control
2 DMAs	Flexible data handling
FMAC	Filter for accelerated control calculations
CORDIC	Accelerated trigonometric functions
FPU	Fast software math functions

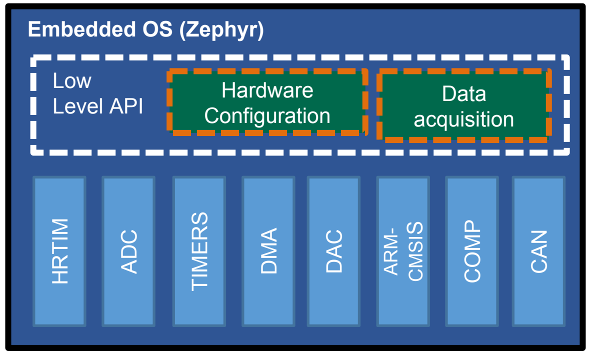
Our SDPE stack

- Communication task
  - (A)Synchronous Data uplink
  - (A)Synchronous Control Downlink
  - Flexible and compatible with multiple protocols
- Slow control task
  - Application oriented control task (e.g. state machine)
  - Updates references of the fast control
- Fast control task
  - Low-level reference tracking control
  - Receives data either for average or peak control
- Hardware MCU peripherals
  - Full integration of peripherals to achieve hardware acceleration for power electronics
  - Voltage or current control implementations
- Power Hardware
  - Flexible and reprogrammable power converter
  - Possibility of integrating external measurements



RTOS-based modular firmware - Control, Application and Communication functions

- RTOS have stacks and natively implement abstraction from hardware
- Out-of-the-box IoT compatible communication stack
- Driver-oriented programming and abstraction
- Low-level API abstraction possible
  - Using RTOS modules to create Power-Electronics oriented abstraction
  - Data Acquisition and handling also abstracted for easier control implementation



Software Inspiration

Modular software that is agnostic of the hardware

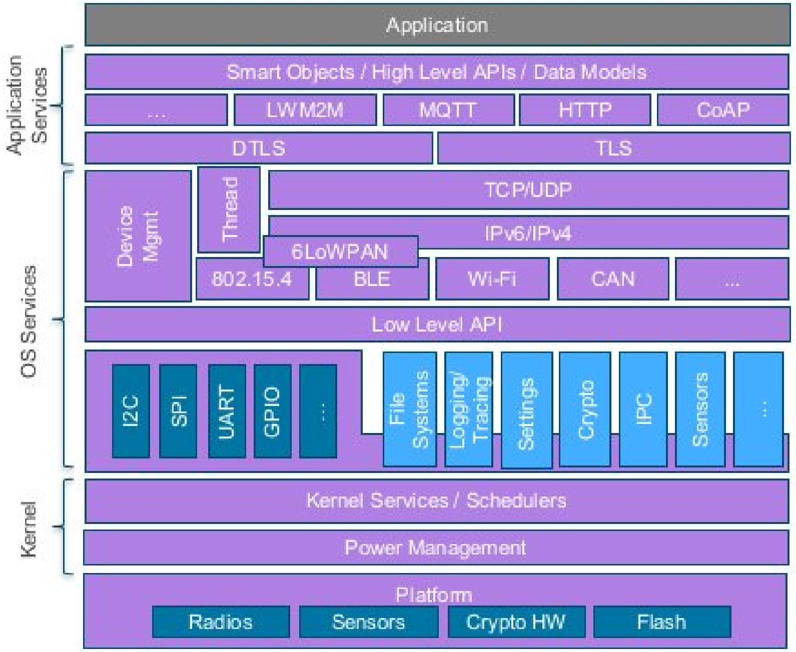
Lightweight code and cross vendor compatible out-of-the-box

Simple to use IDE with plenty of examples and easy to add-on libraries

Intuitive interface for control systems and tuning

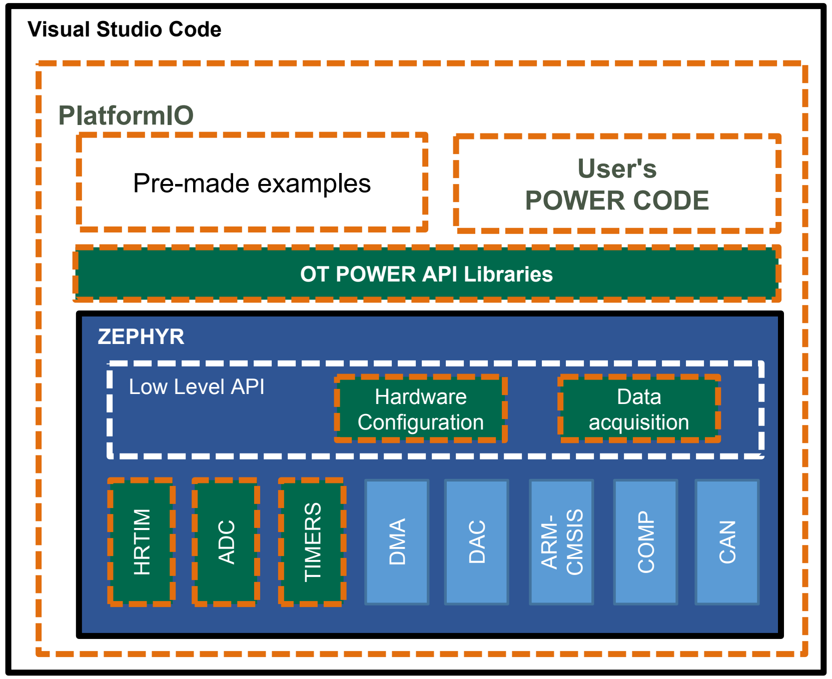
Chosen firmware platform - Zephyr OS

- Open source real-time OS
- Vibrant Community participation
- Safety and Security built-in
- Cross-architecture with broad SoC and development board support
- Vendor Neutral Governance
- Permissively licensed - Apache 2.0
- Complete, fully integrated, highly configurable, modular for flexibility



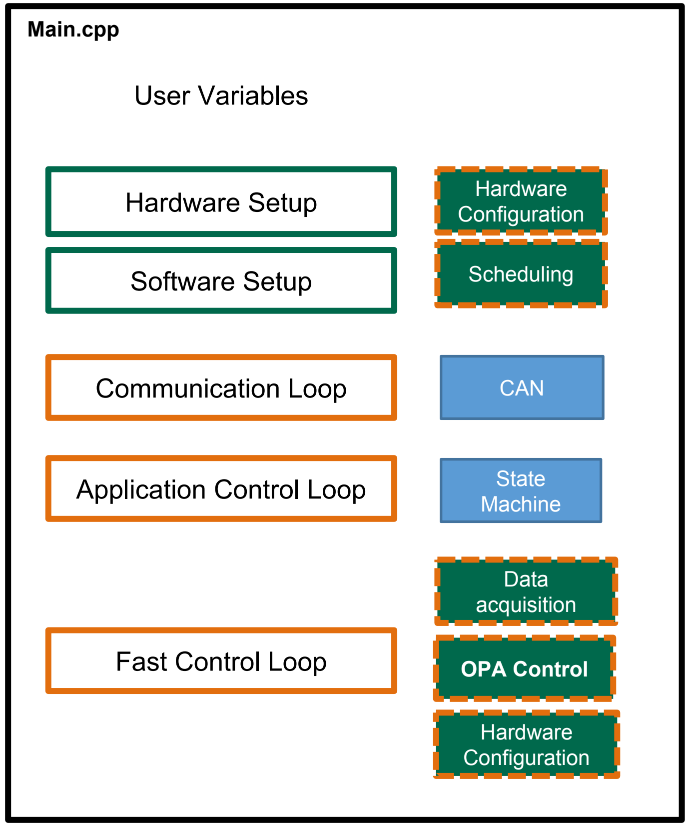
The ergonomics of our Zephyr OS implementation

- VSCode outer enveloppe
- PlatformIO based usage
- Gitlab hosted Libraries
- Automatic handling of other 3rd party modules
- Low-Level API for abstracting the drivers
- In-House HRTIM driver
- Rewritten ADC driver for peripheral-to-peripheral connection
- In-house Timer module (to be reviewed)



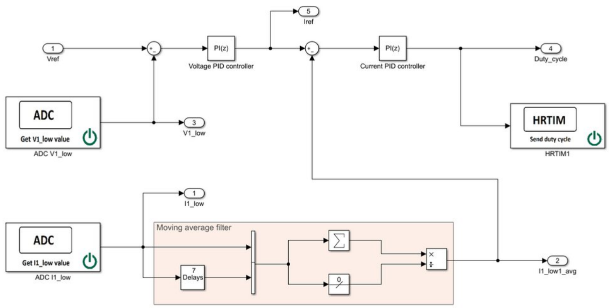
The ergonomics of our Zephyr OS implementation - The main

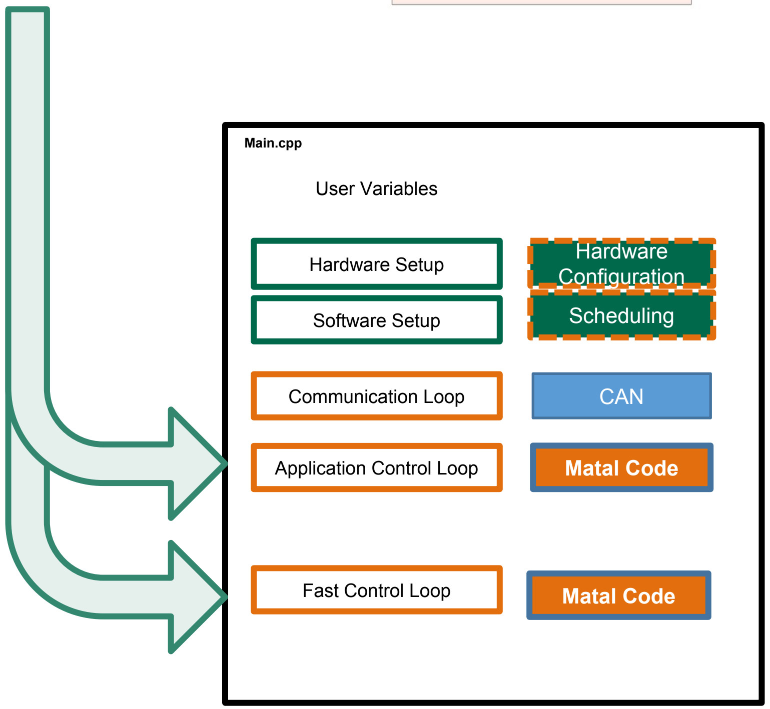
- Single main code for “Arduino-like” experience
- Setup and Loop ergonomics
- Hardware and Software Setup
- Communication, Application and Fast Control Loop



The ergonomics of our Zephyr OS implementation - The Matlab

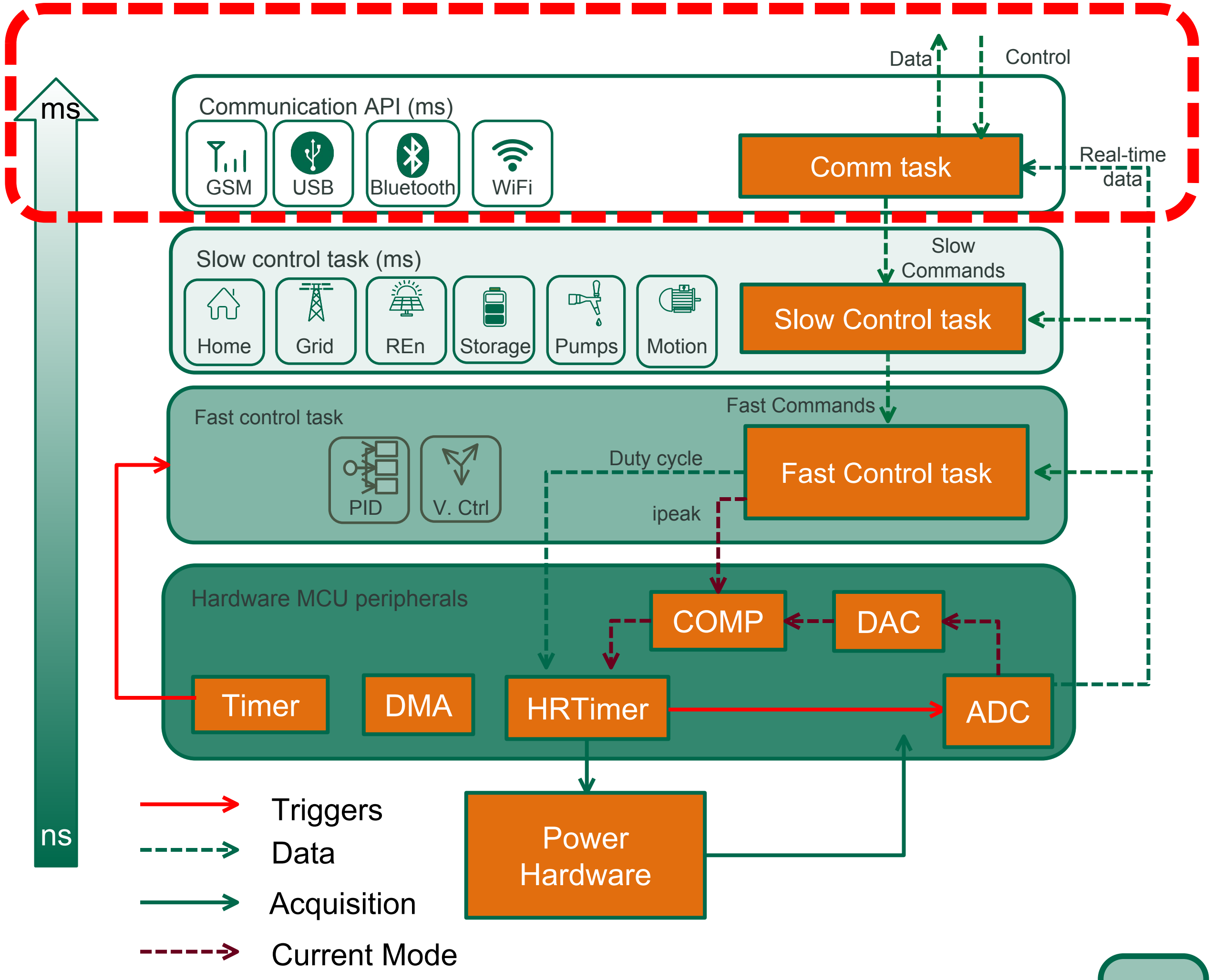
- Matlab blocs can be generated to call the abstraction layer modules
- Application code is automatically generated
- The code is automatically put in the the folder tree
- The code is called from the application loop or the fast control loop





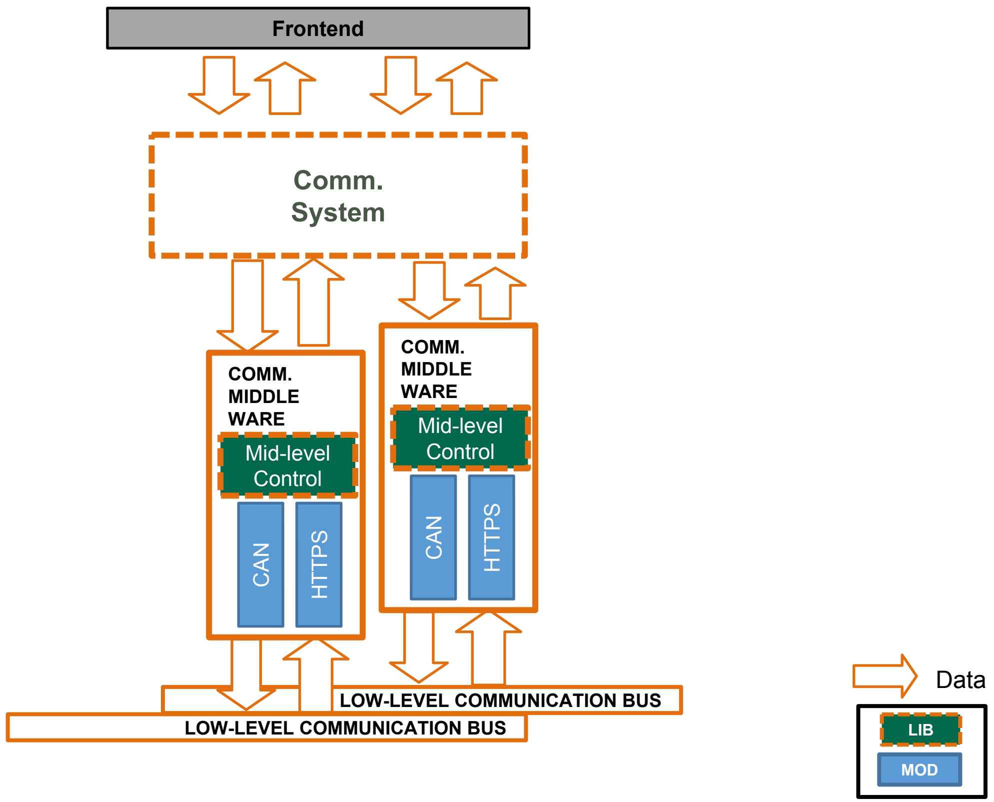
Our SDPE stack

- Communication task
  - (A)Synchronous Data uplink
  - (A)Synchronous Control Downlink
  - Flexible and compatible with multiple protocols
- Slow control task
  - Application oriented control task (e.g. state machine)
  - Updates references of the fast control
- Fast control task
  - Low-level reference tracking control
  - Receives data either for average or peak control
- Hardware MCU peripherals
  - Full integration of peripherals to achieve hardware acceleration for power electronics
  - Voltage or current control implementations
- Power Hardware
  - Flexible and reprogrammable power converter
  - Possibility of integrating external measurements

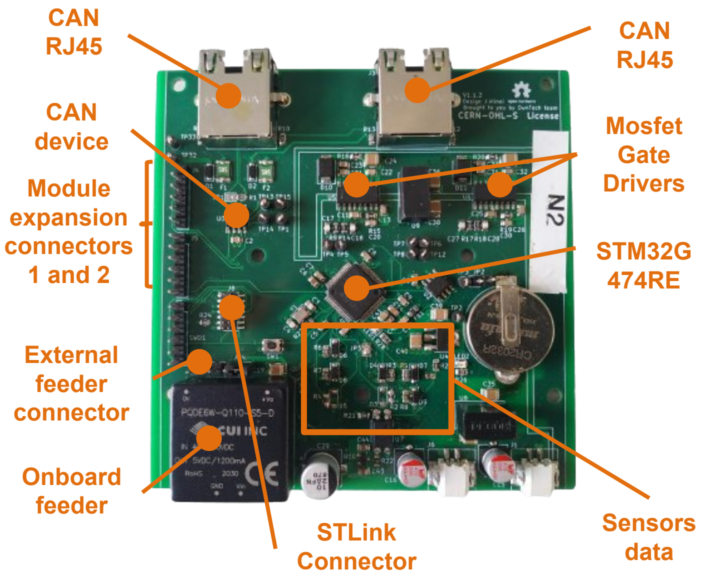


Communication MiddleWare - Specs

- Multi-purpose communication
  - Low level synchronization
  - High level data aggregation
- Data parsing and bi-directional flow
  - Uplink data harvesting
  - Downlink parameter update
- Data treatment
  - Data aggregation and system diagnosis
- Flexible control architectures
  - Decentralized
  - Hierarchical
  - Centralized



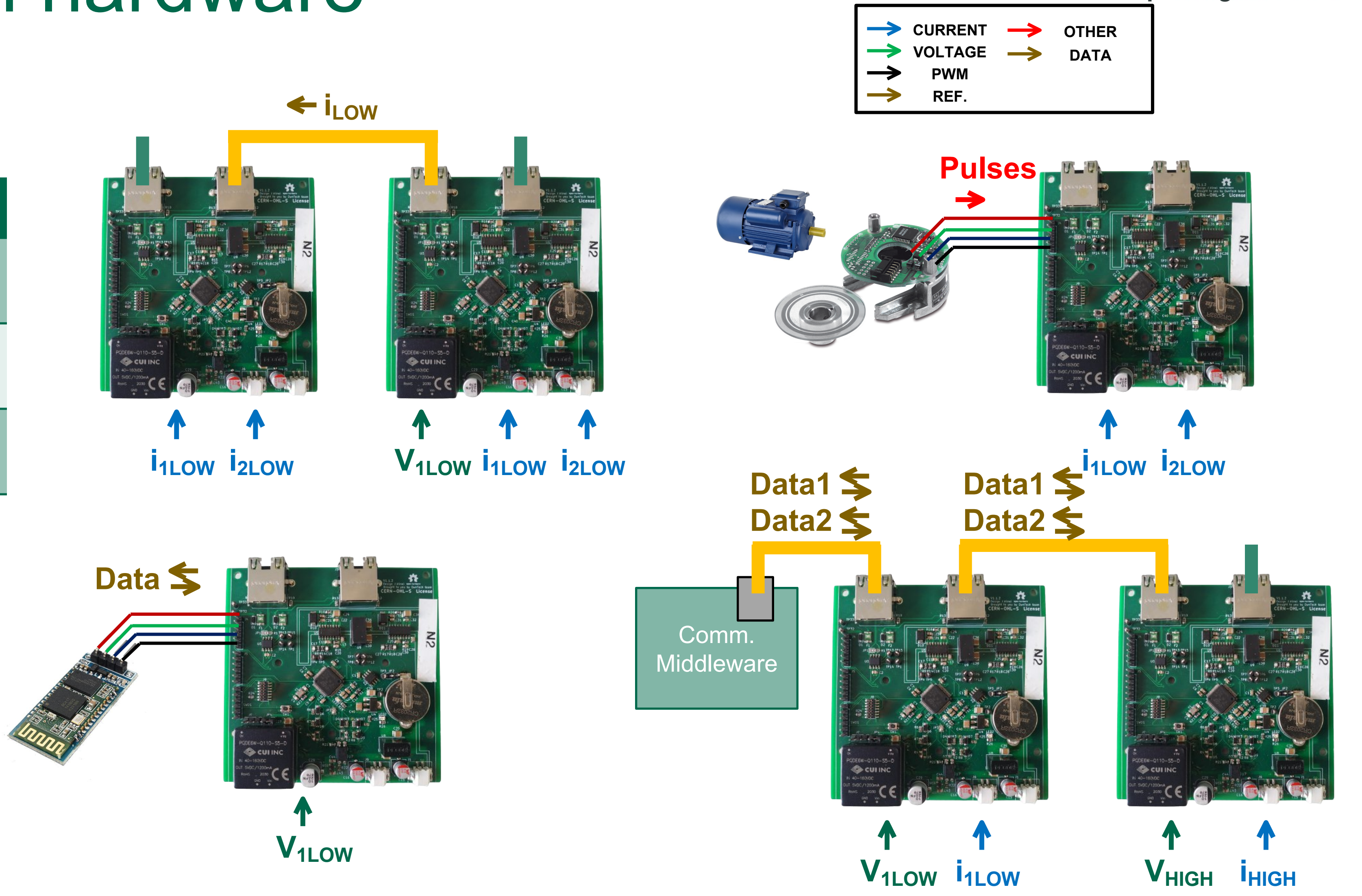
Communication hardware - Digital controller integration



Peripheral	Details
Measurements	$V_{Low1/2}$, $i_{Low1/2}$, V_{High}, i_{High}
CAN-based communication	2 RJ45 connectors to create daisy-chain CAN
Isolated gate drivers	PWM to power stage driver
Expansion connector 1	USART, Rotary Encoder, GPIOs
Expansion connector 2	SD Card module, Rotary Encoder, GPIOs
STLink Connector	Programming and debugging

Communication hardware - Use cases

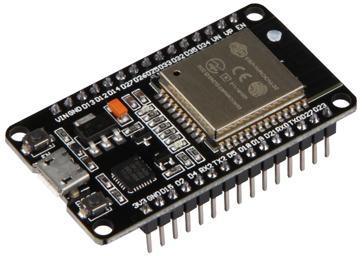
Function	Use case
CAN-Based communication	Master-slave control, System monitoring
Expansion connector 1	Bluetooth via USART, Rotary Encoder, GPIOs
Expansion connector 2	SD Card module, Rotary Encoder, GPIOs

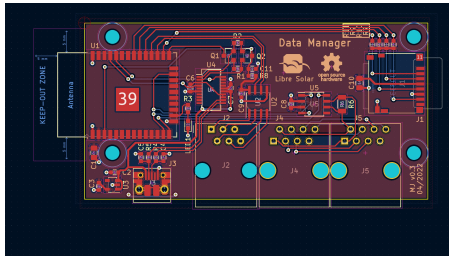


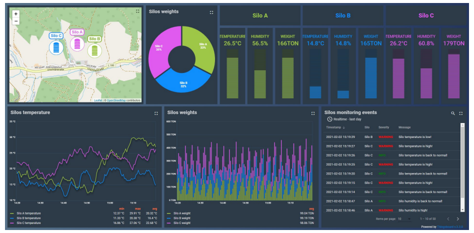
Communication MiddleWare - ThingSet + ESP32

- ThingSet
  - Created by Martin Jäger
  - thingset.io
  - Target agnostic data parsing system
  - Possibility of creating data structures
- ESP32
  - Low-cost and ubiquitous IoT solution
  - Separated cores allow for data pre-treatment
- Dedicated dongle
  - Created by Martin Jäger and LibreSolar
  - Open-Source ESP32 based solution compatible with CAN
- FontEnd
  - Data Aggregation
  - Using ThingsBoard

ThingSet





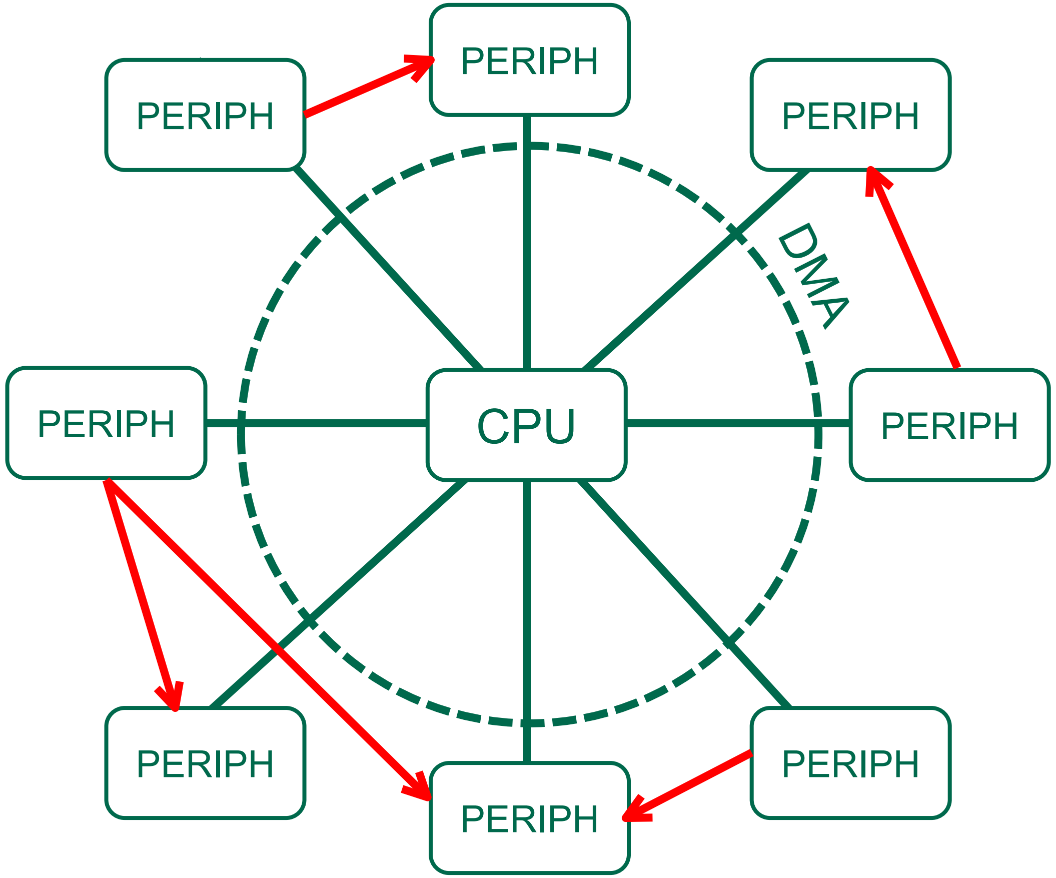


Outline

- Overview of the monolithic approach
- Study case
- Lessons learned

Lessons learned – The star and the circle

- Power electronics have a very niche need of combining CPU flexibility and hardware acceleration
- Manufactures have solutions, but they require bare-metal coding
- Interfaces needed by SDPE
  - **Peripheral-to-peripheral via DMA**
  - **Peripheral interconnection matrix**
- Everything is possible, but not always available

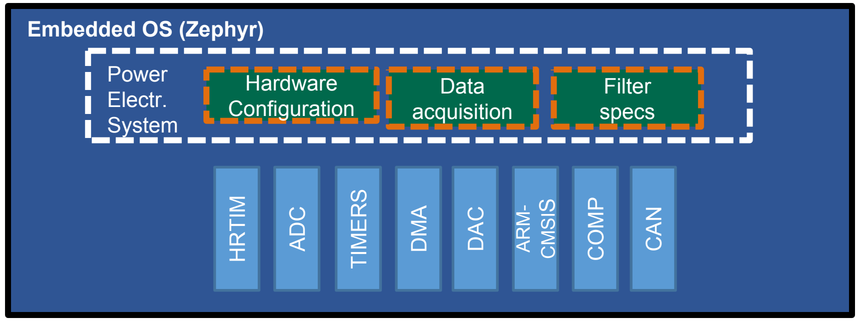


Zephyr™ Project Developer Summit

OwnTech Open Digital Power

Lessons learned – Empower the trees

- It is necessary to create an abstraction for power hardware in Zephyr
- This would allow for a truly vendor neutral support for power electronics systems
- **How to write this abstraction in Zephyr?**
  - Single block in the DTS using handles

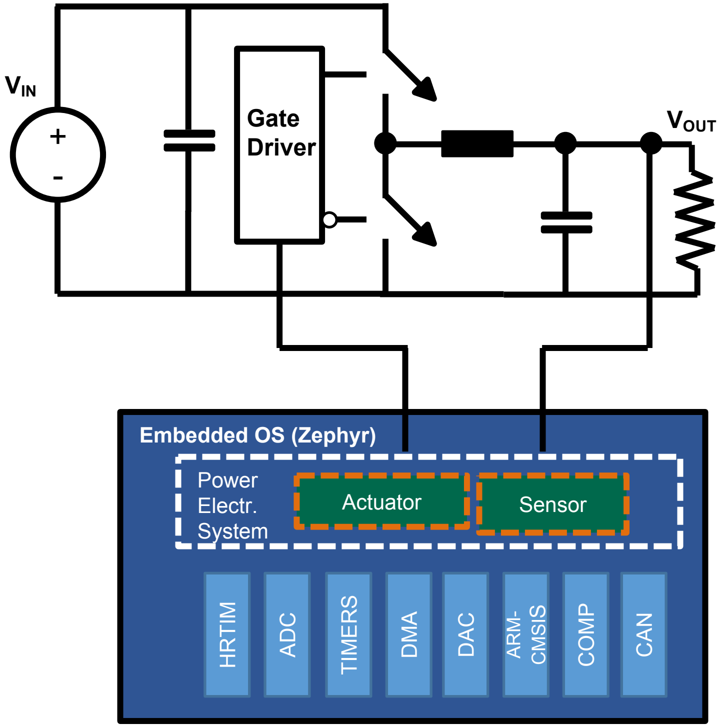



```
/ {
  power-hardware {
    compatible = "st,nxp";

    powerlegs {
      leg1: leg1 {
        compatible = "buck, boost";
        hrtimer = < &hrtim1 > ;
        high-switch = < &hrtim1_cha1_pa8 > ;
        low-switch = < &hrtim1_cha1_pa9 > ;
        status = "okay";
        inductor-max-current = < 10 >;
        vlow-max-voltage = < 72 >;
        vhigh-max-voltage = < 110 >;
        label = "LEG1";
        phandle = < 0x70 >;
    };
```

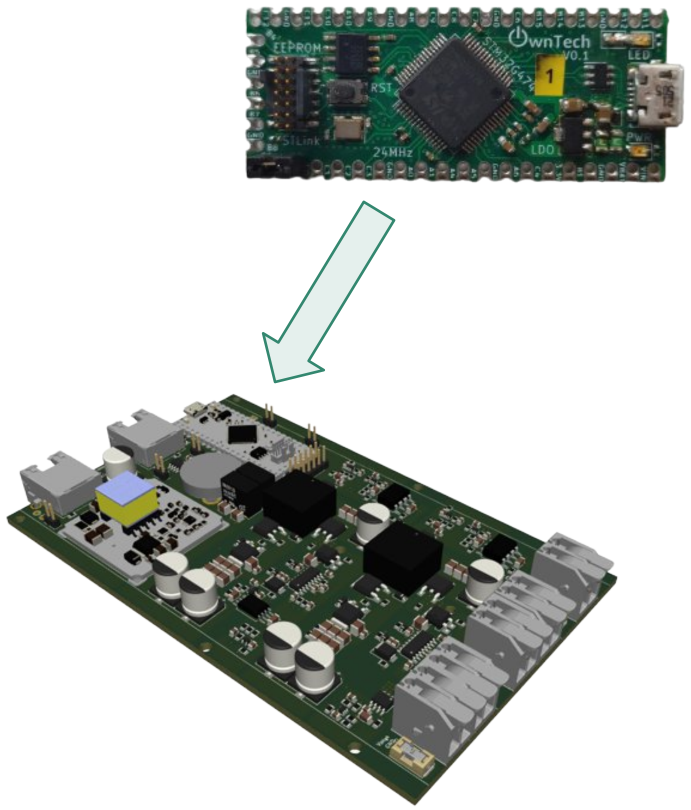
Lessons learned – What goes up **should** come down

- Embedded systems need to take data and control targets
- Zephyr has the start of an uplink with the sensor API, but it still has **a complex workflow**
- Zephyr would benefit enormously from an **“actuators” API to close the loop**



Lessons learned – Breakdown big problems in small parts

- The problem is complex and the market fragmented
- It seems important to separate the brains from the muscle
- Dedicated board for the SoC
  - Feather footprint
  - Leverages highly specialized peripherals
- Dedicated board for power
  - 160mmx100mm footprint (Euroboard)



Thank you!

contact@owntech.org
https://gitlab.laas.fr/owntech

Luiz Villa and Jean Alinei
LAAS-CNRS

OwnTech is a Technology Sandbox

OwnTech Open Digital Power

Zephyr™ Project Developer Summit

OwnTech user-centric technology sandbox provides all elements to foster a community of users and developers

The OwnTech foundation ensures open and fair access to the technology sandbox

Academia

Industry

Makers and fablabs

Standard Hardware

Intuitive IDE

Unified documentation

Open Data Monitoring

Technology Sandbox: A combination of open-hardware, open-software and with community-based activities to foster bottom up innovation

This fertile open sandbox produces impact

New ideas

The technology can be used for limitless amount of applications such as Smart Grids, electrical mobility, energy storage and much more

New talents

The open technology enables learning by doing and fosters community exchange

OwnTech can teach and do

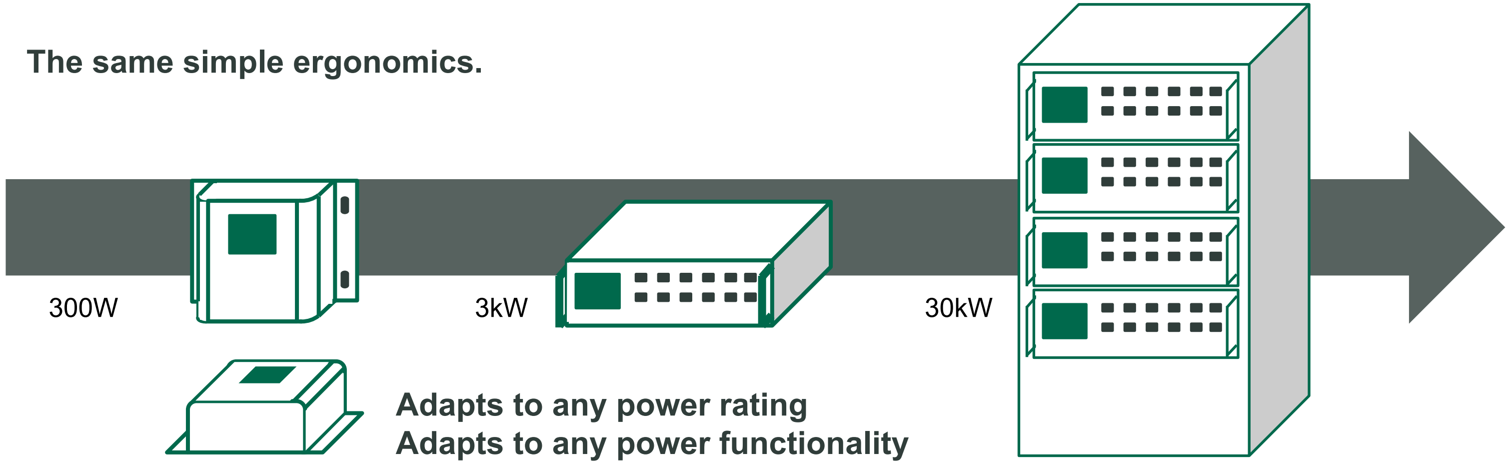
A standard module

- 300W per module
- Compact 100x100mm form factor
- Around 100€

As a stand alone module..

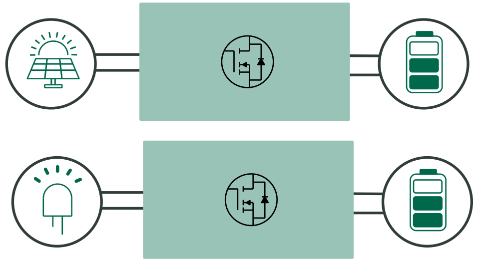
Or in industrial 19” rack..

The same simple ergonomics.



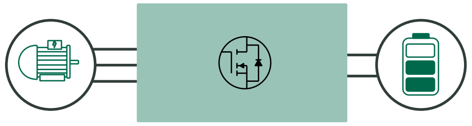
Adapts to any power rating
Adapts to any power functionality

Use case example for Education



Instructors can easily setup their practical classes and manage a group of students who are reprogramming the target while following their class instructions.

Use case example for Industry

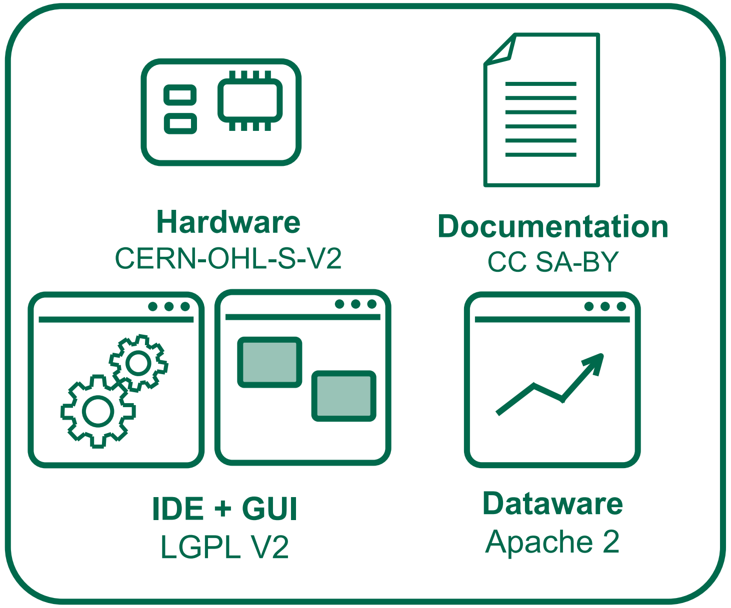


Control engineers setting up their test benches in short periods of time. They are also seamlessly changing their models and reprogramming the converter to control their target application.

Tool and Solution

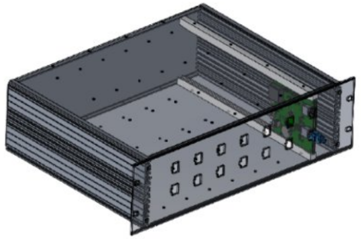
Notion of Tool

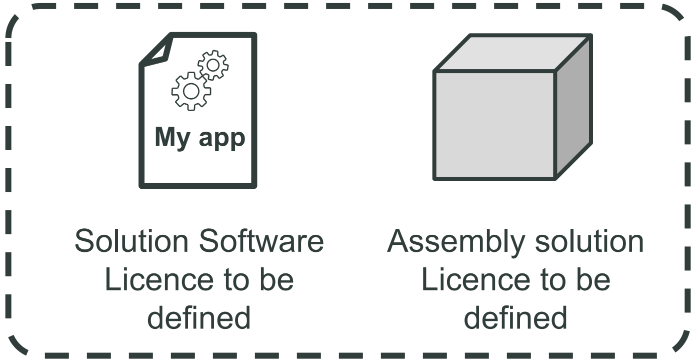
- All the **open-source IP**
- **Hardware, software and data**
- Belongs to the CNRS and the University of Toulouse
- IP declared for precedence



Notion of Solution

- **An application of the Tool**
- **Can be open or closed-source**
- **Belongs to whoever develops it**





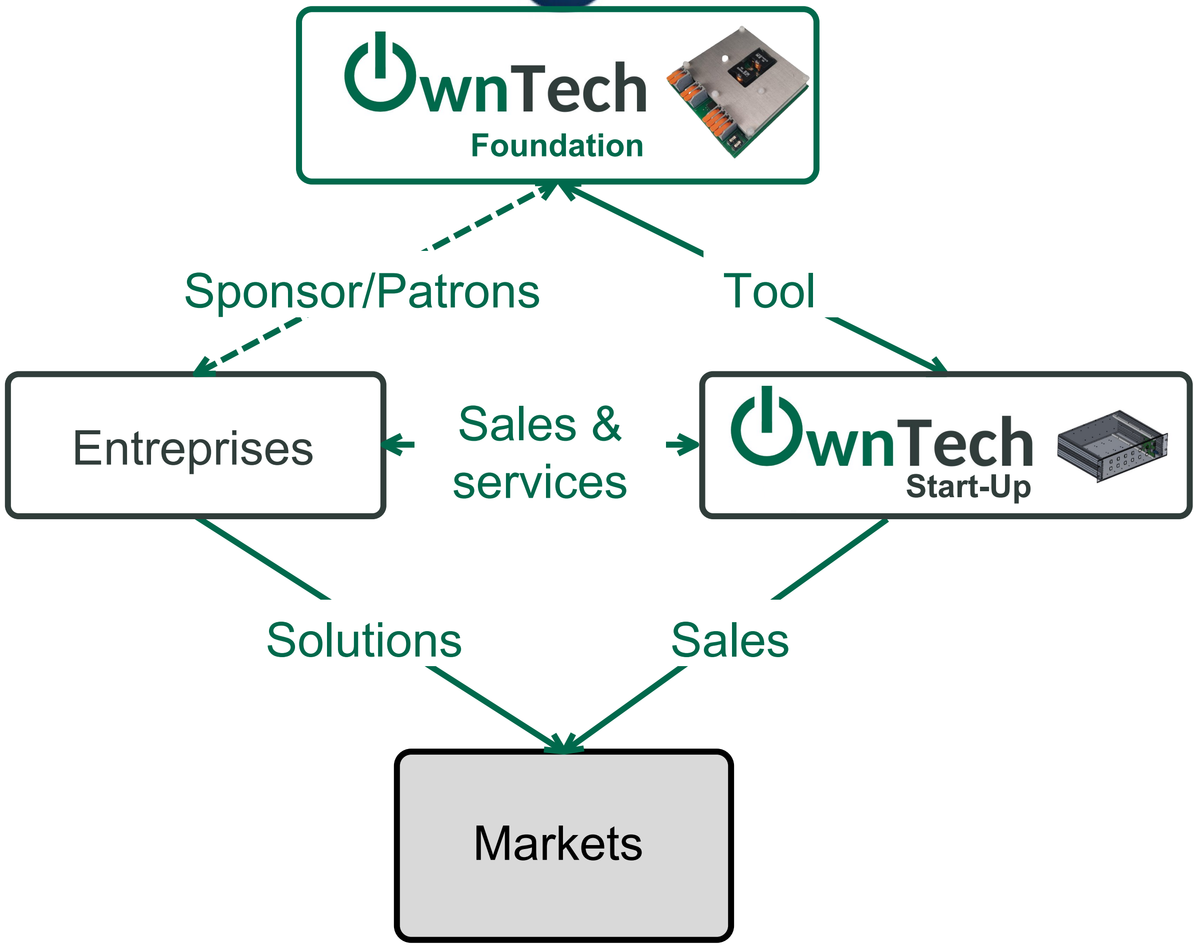
OwnTech will be a hybrid entity

The Fondation

- Manages the **Tool** (usable by all)
- Manages the **Community**
- Creates **educational** content
- Has a mandate **over the IP and the Tool**

The Start-Up

- **Create solutions** from the Tool
- Manages **hardware sale**
- Provides **services** to companies
  - Support and guarantees
  - Solutions and taylored functions
- Trains companies on the technology



Our Traction

In one year we've grown already

Tens of early adopters

800k€ secured funding to refine the technology

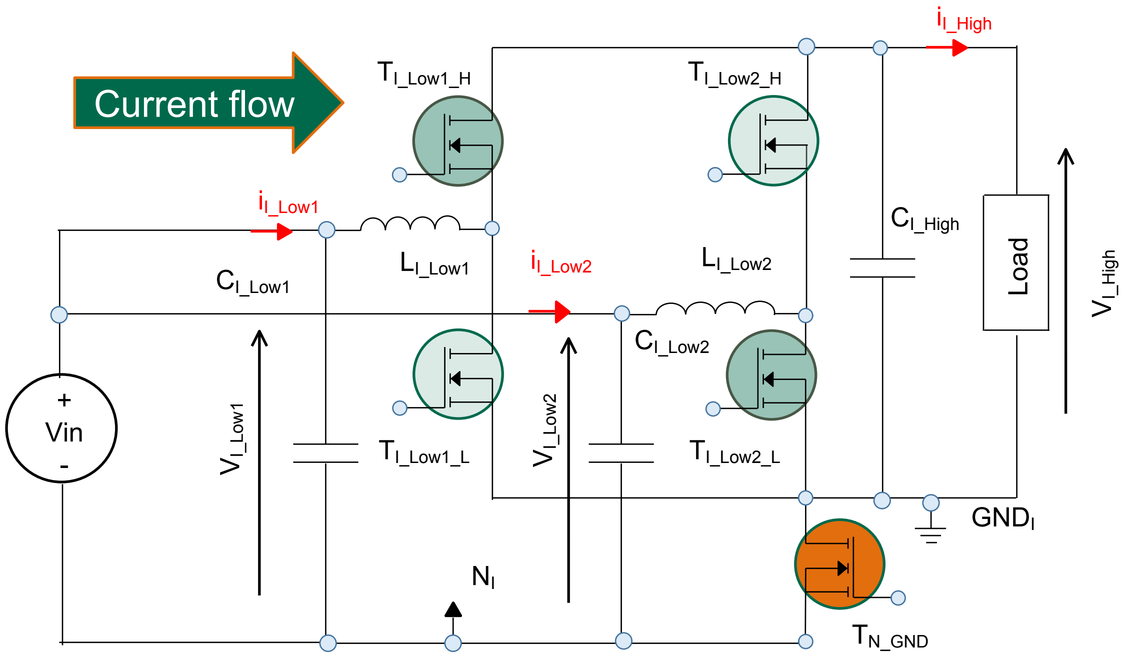
2 Industrial contracts

5 professors ready to use the technology for their teaching

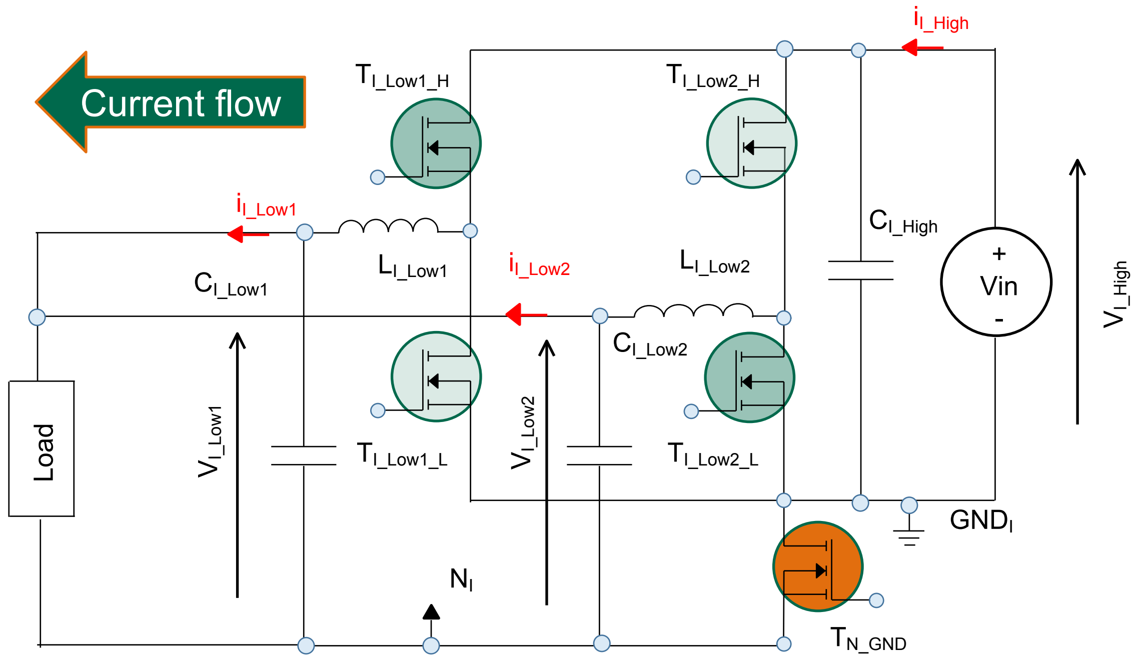
OwnTech Foundation has been accepted to be hosted by the CNRS foundation

OwnTech persues advanced discussions with silicon vendors to become Sponsors and or Patrons

Multi-function Power Hardware - Functions



Variable	Value
T_{N_GND}	ON
Function	**Interleaved Boost**
Vin	24V
Vout	V_{I_High}
Vref	50V

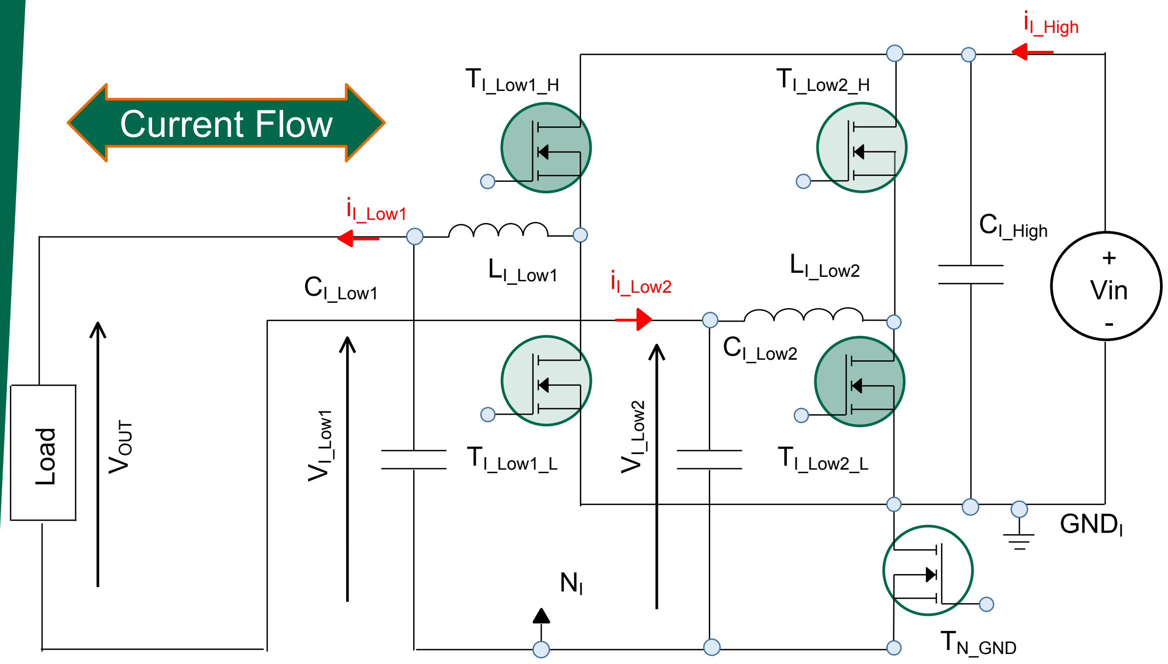


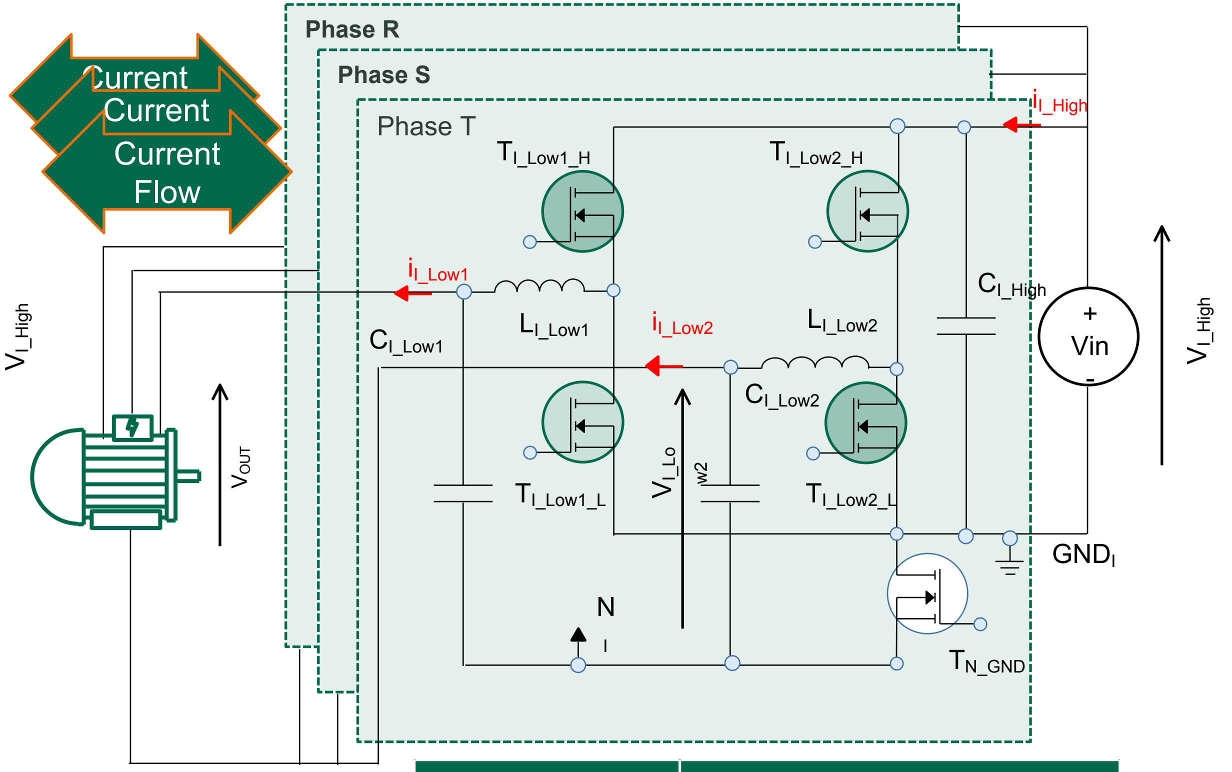
Variable	Value
T_{N_GND}	ON
Function	**Interleaved Buck**
Vin	50V
Vout	V_{I_Low1} and/or V_{I_Low2}
Vref	24V

Multi-function Power Hardware - Functions

Zephyr™ Project Developer Summit

OwnTech Open Digital Power





Variable	Value
T_{N_GND}	**OFF**
Function	**Buck 1phase inverter**
Vin	110V
Vout	$V_{I\_Low1} - V_{I\_Low2}$
$Vref_{PK}$	55V
$Vref_{RMS}$	38.9V

Variable	Value
T_{N_GND}	**OFF**
Function	Buck 3phase inverter
Vin	110V
Vout	$V_{I\_Low1} - V_{I\_Low2}$
$Vphase_{RMS}$	38.9V
$Vline_{RMS}$	67.4V

Generalized Module Layer - P.E. Control Techniques

- Three techniques handle the majority of cases
- Voltage mode
  - Control of the duty cycle
  - Average values control only
- Current mode
  - Control of the current peak
  - Average voltage control
- DC-AC
  - Very calculation intensive
  - Several measurements