Zephyr™ Project
Developer Summit
June 8-10, 2021 ▪ @ZephyrIoT

LIBRESOLAR

IoT-enabled Solar Power Converters with Zephyr

MARTIN JÄGER - LIBRE SOLAR

The Libre Solar Project

LIBRESOLAR

The Libre Solar Project

Zephyr™ Project Developer Summit

Open Hardware Community

FABULOUS ST. PAULI

Wind EMPOWERMENT

LIBRESOLAR

FAB CITY HAMBURG

HOOU HAMBURG OPEN ONLINE UNIVERSITY

The Libre Solar Project

Open Hardware Community

Energy Access Applications

FABULOUS ST. PAULI

Wind EMPOWERMENT

LIBRESOLAR

SCENE

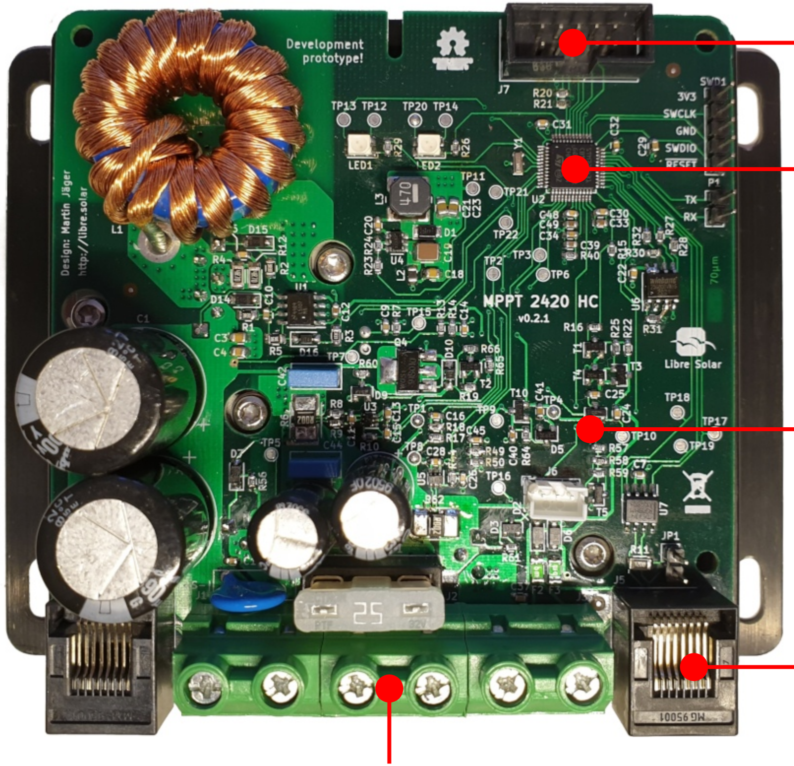
connected energy

FAB CITY HAMBURG

HOOU HAMBURG OPEN ONLINE UNIVERSITY

Open DC Grid

MPPT Charge Controller



Internal extension port
via UEXT connector

Zephyr™ Project

KiCad

External communication
ports (serial or CAN bus)

Power terminals (solar, battery, load)

Firmware Requirements

Zephyr™ Project Developer Summit

Communications / IoT

- CAN bus for system-level control
- LoRaWAN
- GSM modem

Easy Customization

- Strict separation between board definition and application firmware
- Different application features should be easily selectable

Firmware Requirements

Communications / IoT

- CAN bus for system-level control
- LoRaWAN
- GSM modem

Easy Customization

- Strict separation between board definition and application firmware
- Different application features should be easily selectable

Power Electronics

- Tight coupling between ADC, DAC and PWM signal generation
- Hard real-time control: Missed deadlines could result in system failure
- Offloading of I/O with DMA
- Watchdog supervision for individual threads

Custom Devicetree bindings (1)

Zephyr™ Project
Developer Summit

```
/ {
    pcb {
        compatible = "charge-controller";
        type = "MPPT 2420 HC";

        hs-voltage-max = <90>;
        ls-voltage-max = <32>;
        dcdc-current-max = <20>;
    };
};

&timers1 {
    status = "okay";
    halfbridge {
        compatible = "half-bridge";
        pinctrl-0 = <&tim1_ch1_pa8 &tim1_ch1n_pc13>;
        frequency = <70000>;
        deadtime = <300>;
    };
};
```

Hardware design parameters like current or voltage limits are specified via custom devicetree bindings.

Code example:

```
hs_voltage_max =
    DT_PROP(DT_PATH(pcb), hs_voltage_max);
```

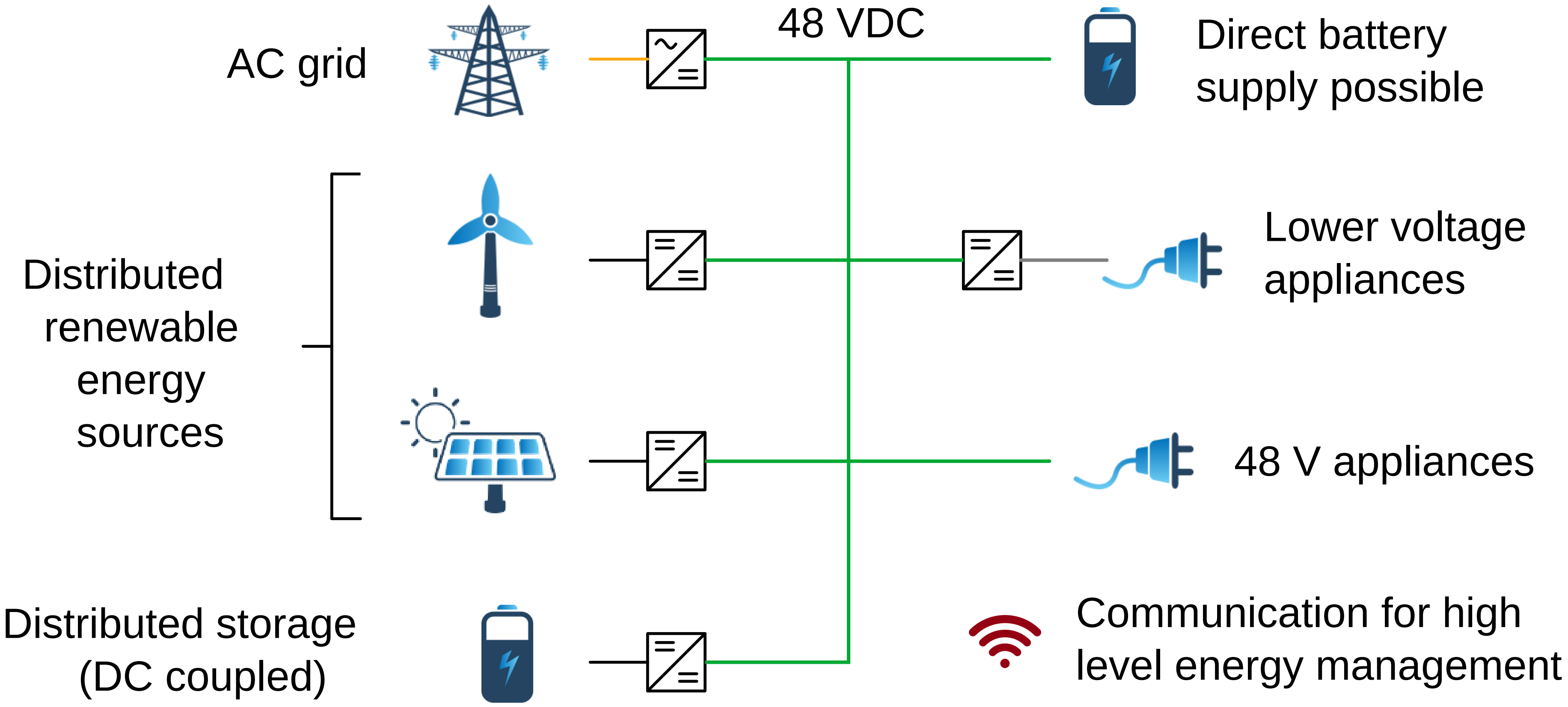
Custom Devicetree bindings (2)


```
/ {
    adc-inputs {
        compatible = "adc-inputs";
        v-low {
            io-channels = <&adc1 12>;
            multiplier = <105600>;
            divider = <5600>;
        };
        v-high {
            io-channels = <&adc1 15>;
            multiplier = <102200>;
            divider = <2200>;
        };
        i-dcdc {
            io-channels = <&adc2 1>;
            // amp gain: 25, resistor: 2 mOhm
            multiplier = <1000>;    // 1000
            divider = <50>;         // 2*25
        };
    };
};
```

Some preprocessor magic to find position of single ADC measurement in the array written by the DMA controller with e.g. `ADC_POS(i_dcdc)`:

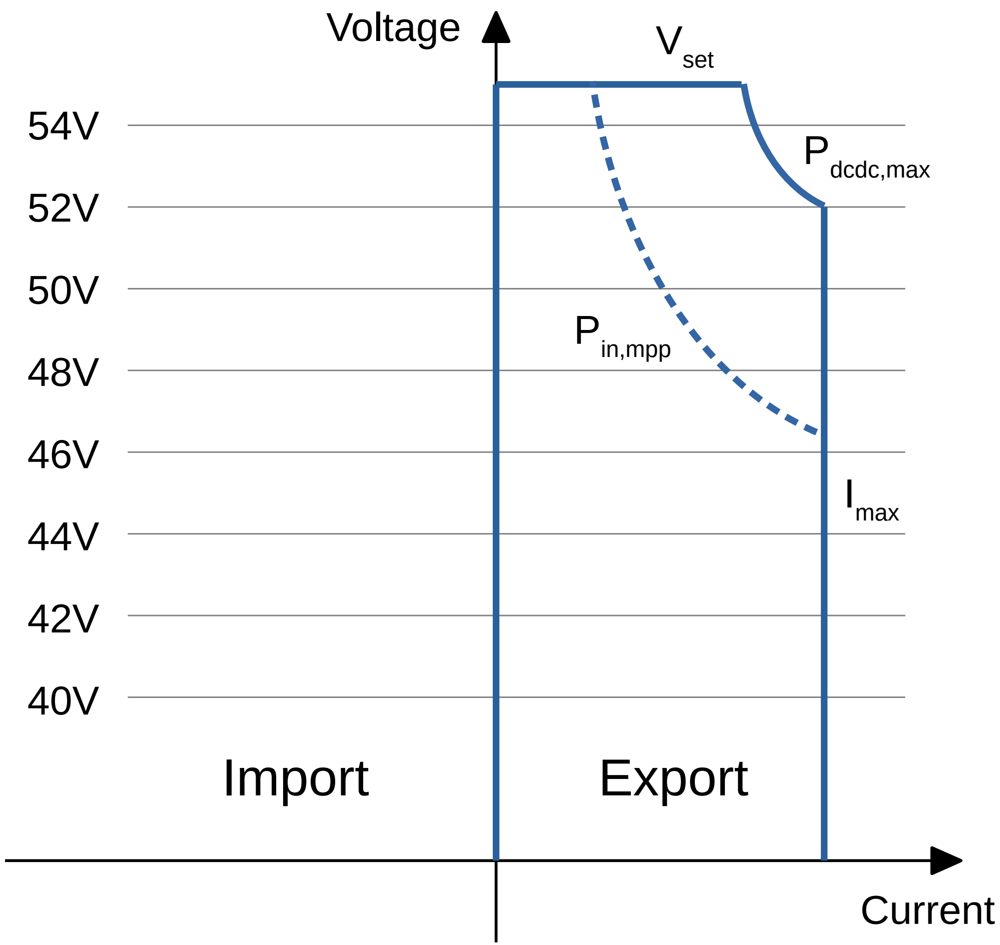
```
/*
 * Find out the position in the ADC reading array
 * for a channel identified by its Devicetree node
 */
#define ADC_POS(node) \
    DT_N_S_adc_inputs_S_##node##_ADC_POS

#define ADC_ENUM(node) node##_ADC_POS,

enum {
    DT_FOREACH_CHILD(DT_PATH(adc_inputs), ADC_ENUM)
    NUM_ADC_CH   // trick to get number of elements
};
```

48V DC Grid

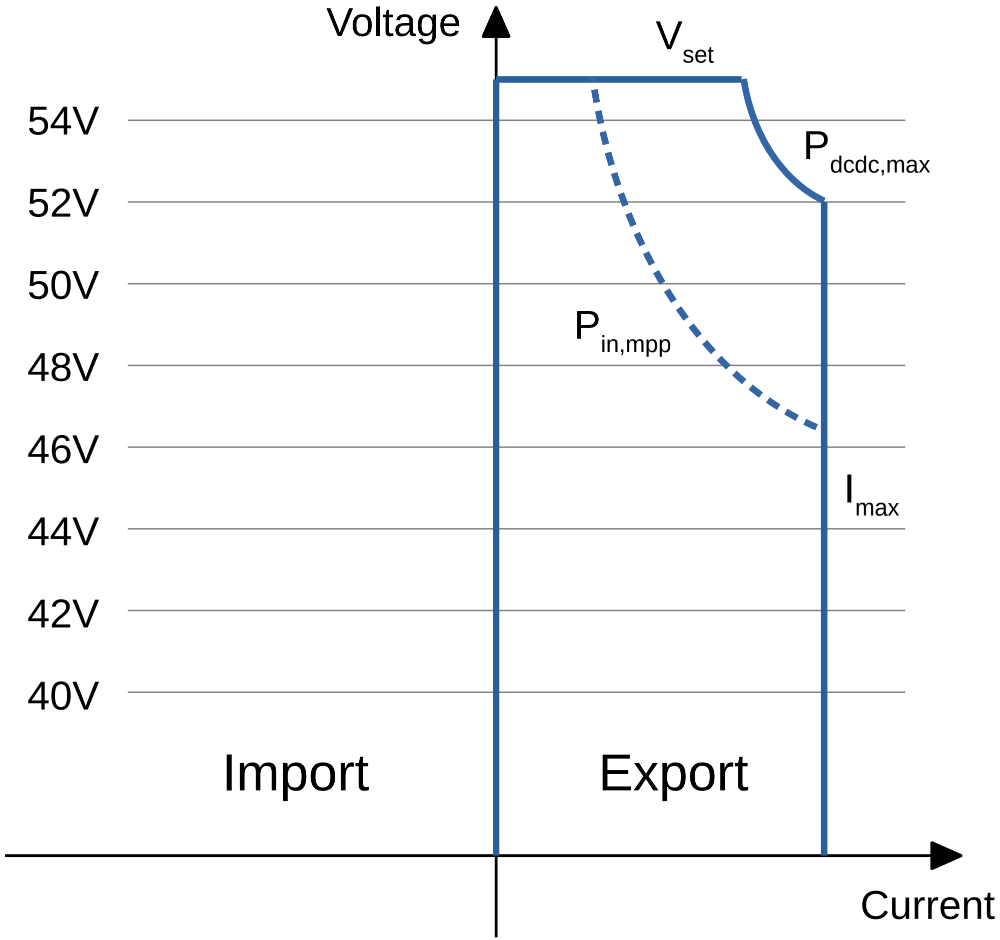

AC grid
48 VDC
Direct battery
supply possible
Distributed
renewable
energy
sources
Lower voltage
appliances
48 V appliances
Distributed storage
(DC coupled)
Communication for high
level energy management

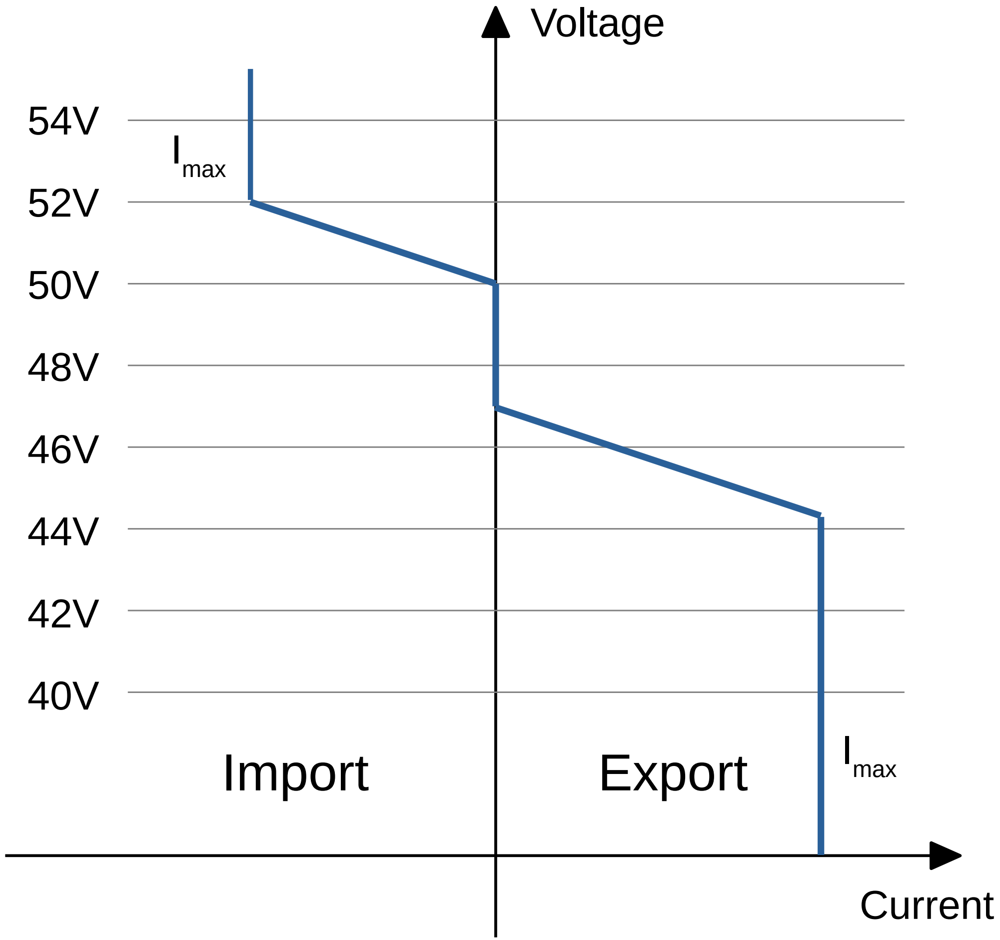
Droop Control: Solar Panel and Battery

Zephyr™ Project Developer Summit

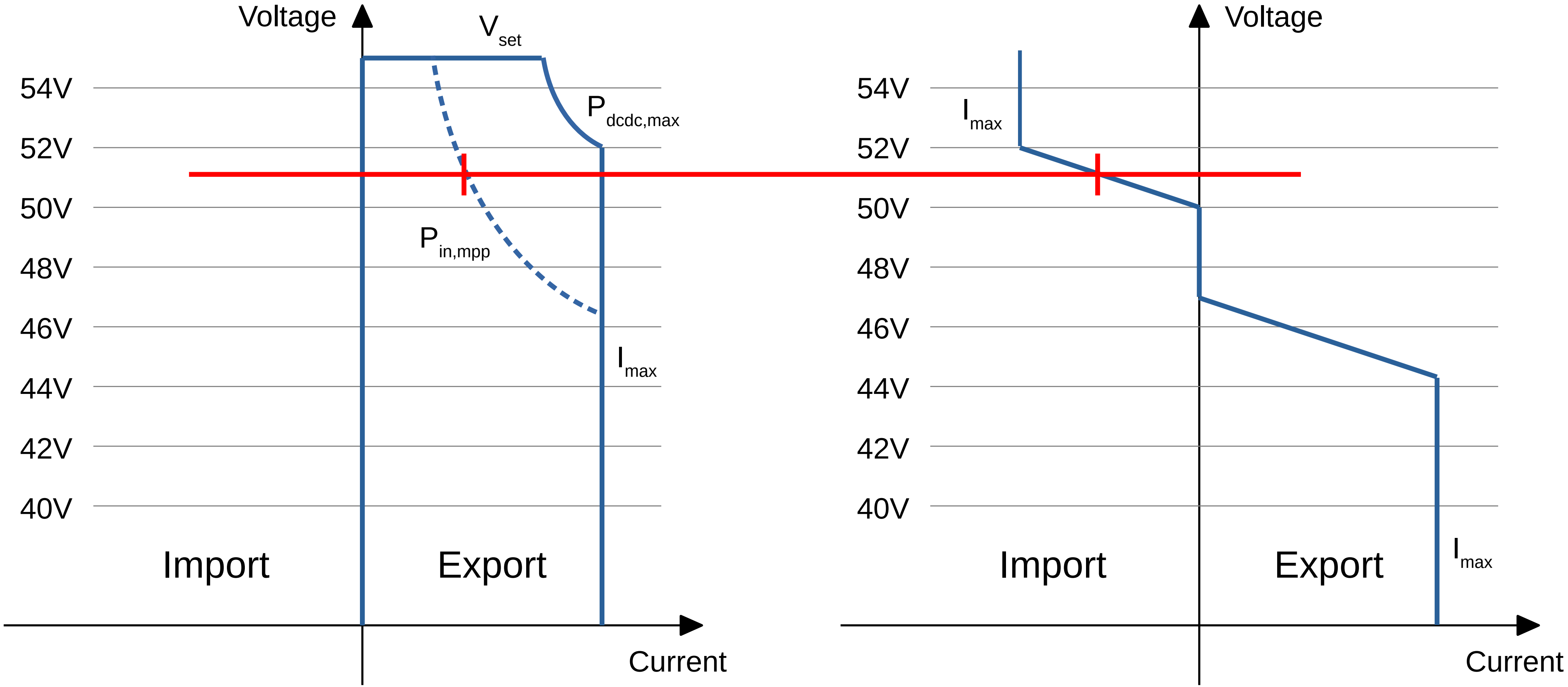


Droop Control: Solar Panel and Battery

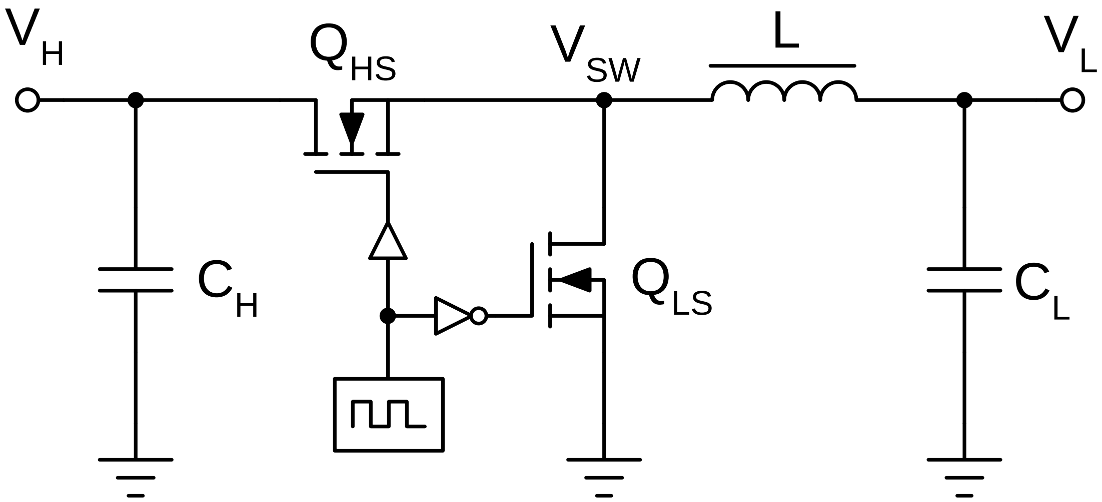




Droop Control: Solar Panel and Battery

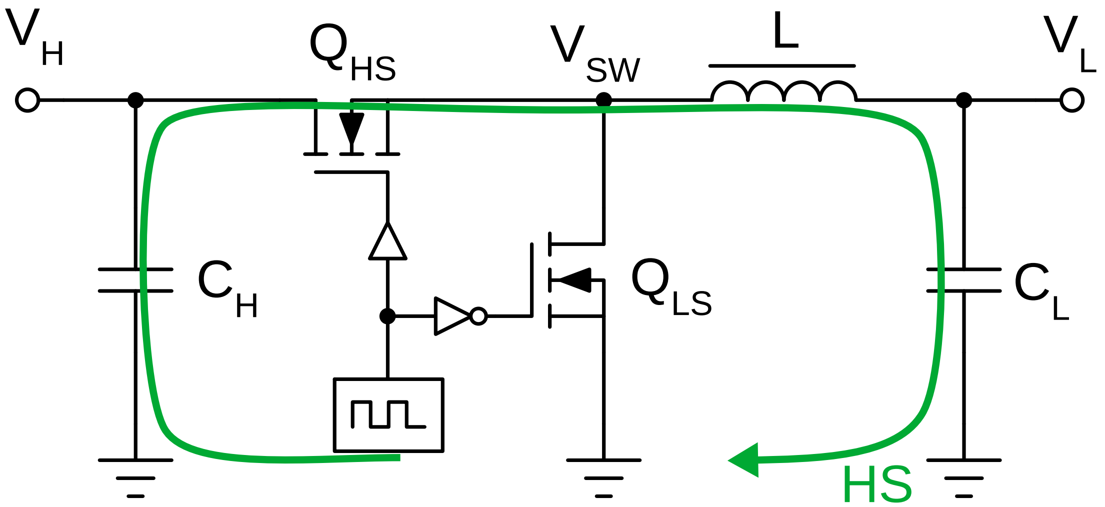


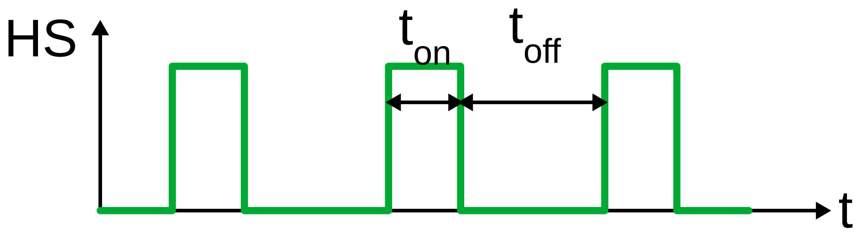
Synchronous DC/DC Converter



$$D = t_{on} / (t_{on} + t_{off}) = V_L / V_H$$

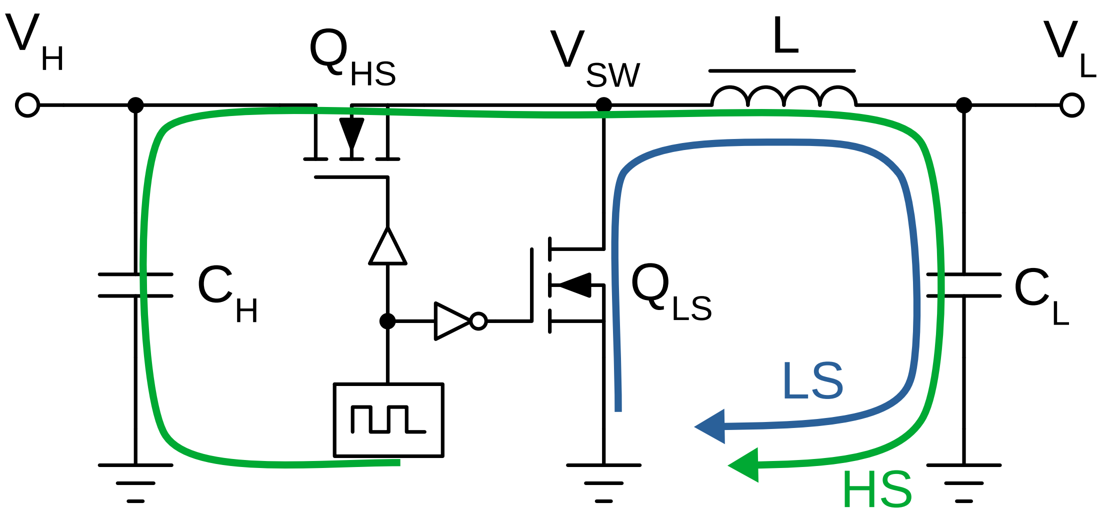
Synchronous DC/DC Converter

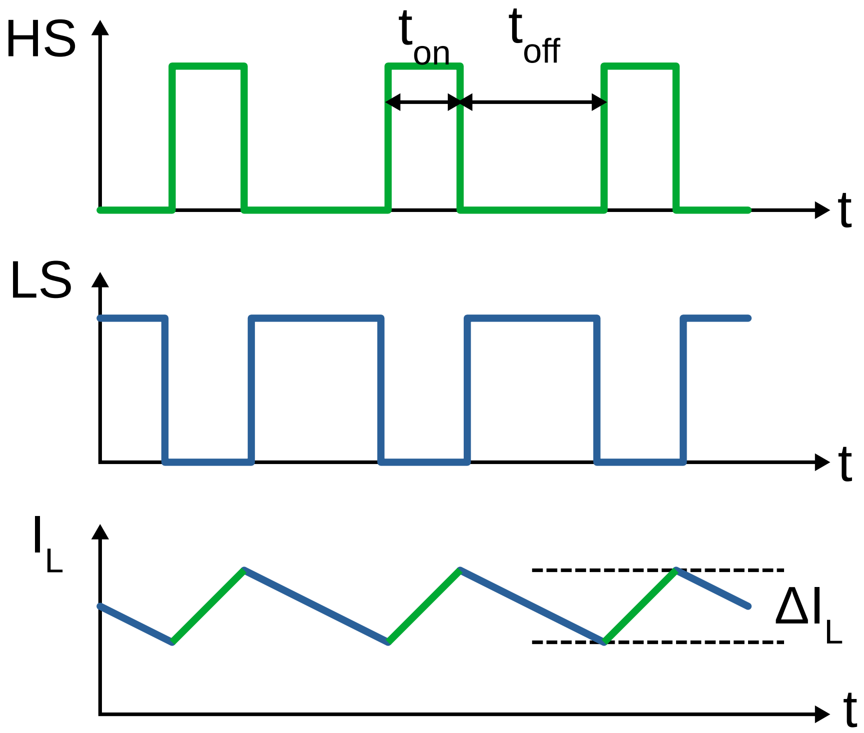




$$D = t_{on} / (t_{on} + t_{off}) = V_L / V_H$$

Synchronous DC/DC Converter

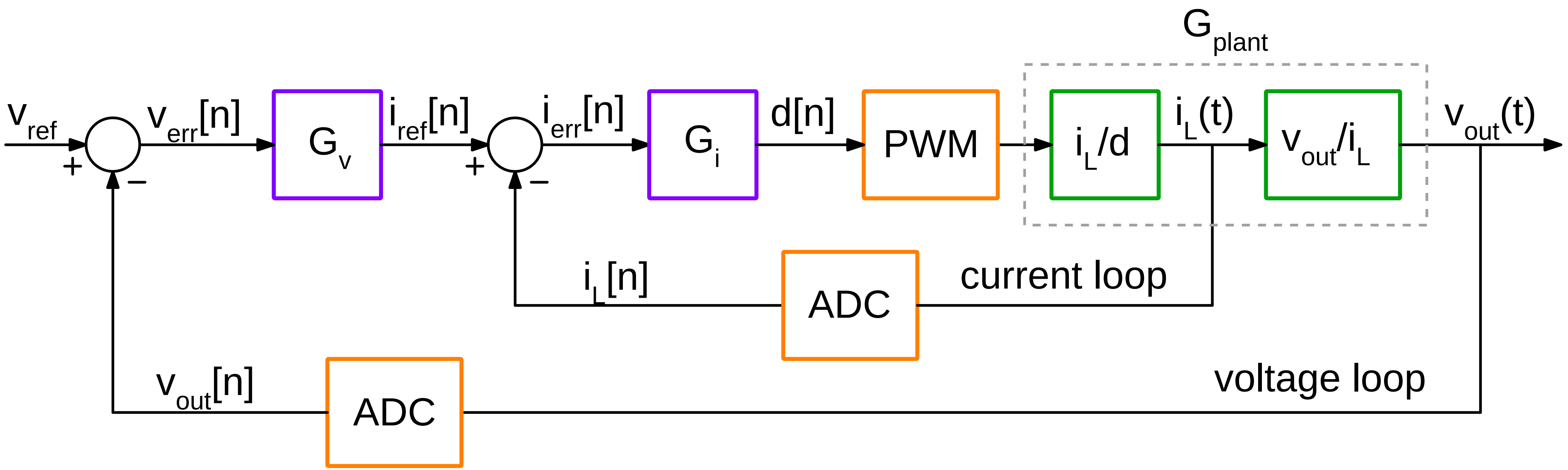




$$D = t_{on} / (t_{on} + t_{off}) = V_L / V_H$$

Digital Control of DC/DC converter

Zephyr™ Project
Developer Summit

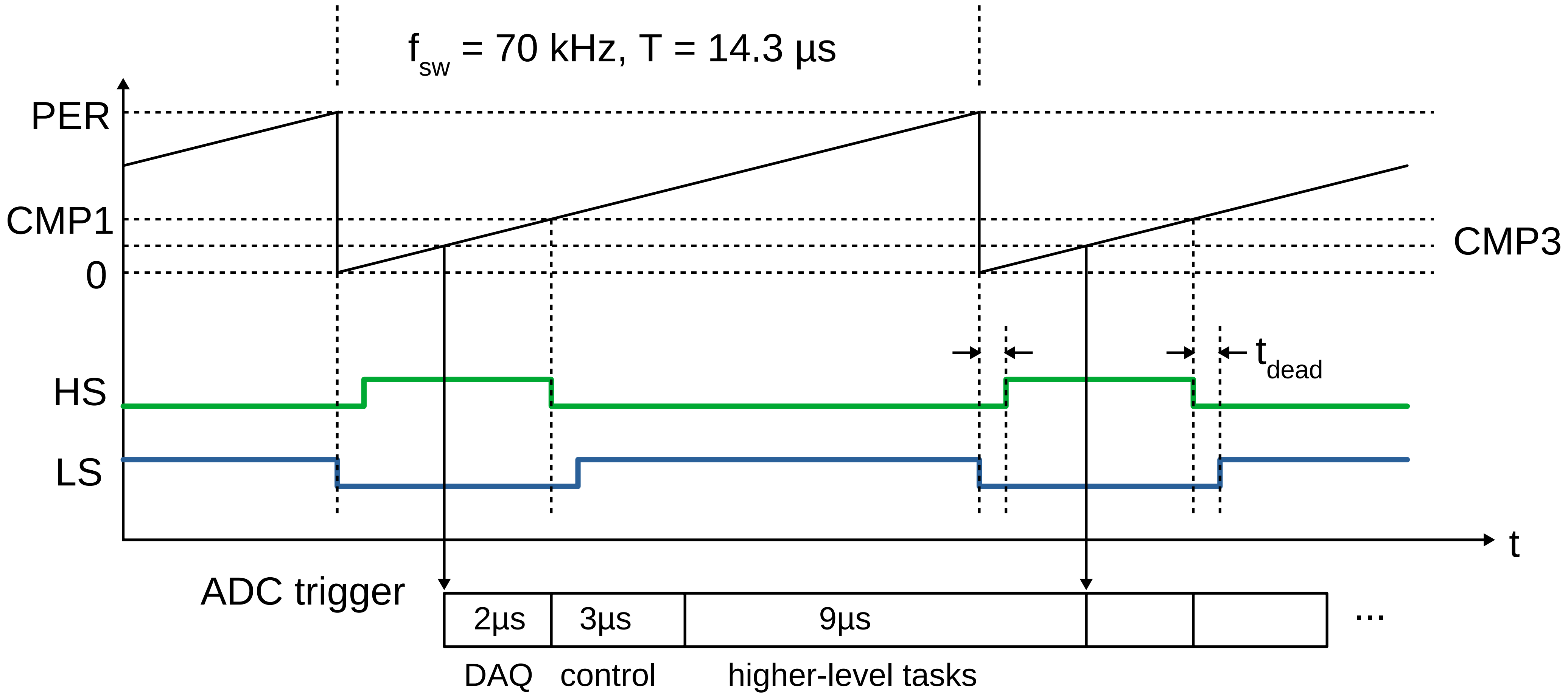


software

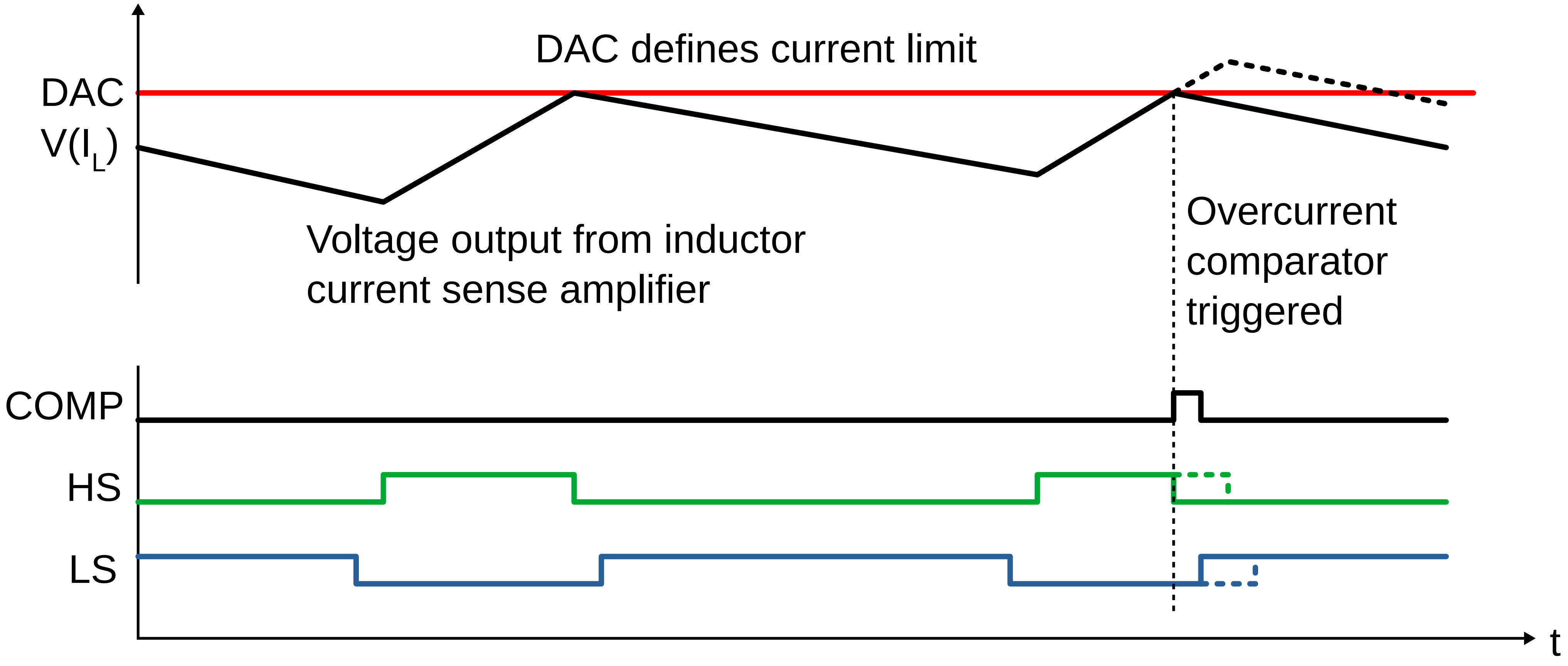
bridge

analog hardware

Timing: HRTIM with ADC trigger



Cycle-by-cycle current limiting



Summary

- Zephyr is a great basis for power electronics development
  - IoT-related functions available out of the box
  - Also industrial protocols like Modbus and CANopen supported
  - Vendor HALs allow easy customization for specific hardware
- Possible future additions
  - High-resolution timer driver
  - Hardware-triggers to connect timers, ADC, DAC and comparators
  - Further offloading of peripherals using DMA

Zephyr™ Project

Developer Summit

June 8-10, 2021 ▪ @ZephyrIoT