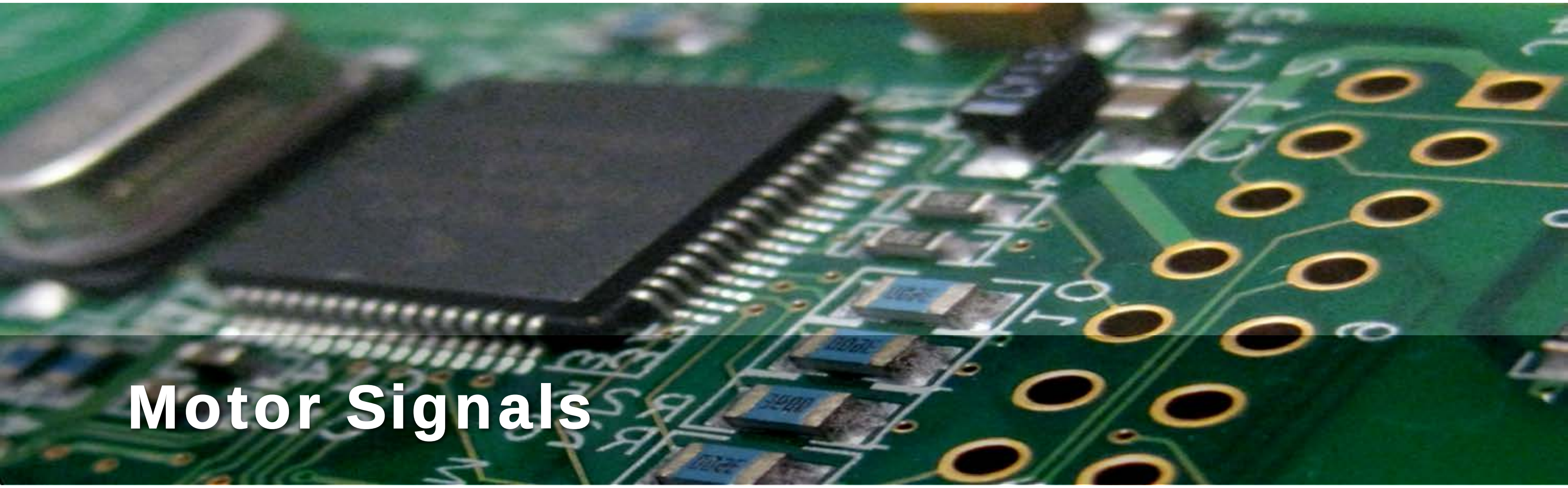
Lucerne University of Applied Sciences and Arts
HOCHSCHULE LUZERN
Technik & Architektur



Motor Signals

"It's all about the right signals…"

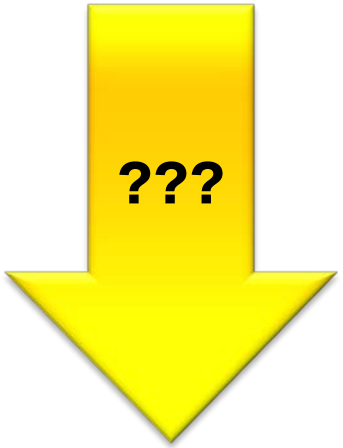
Prof. Erich Styger
erich.styger@hslu.ch
+41 41 349 33 01

Learning Goals

- Driving DC Motor
- Motor Signals
- PWM, Timer Channels
- Direction (DIR)

- Lab Goal:
  - Working motor driver
  - Shell interface
    - Direction
    - Speed/PWM

- On your own:
  - Run the motors
  - Stop at line

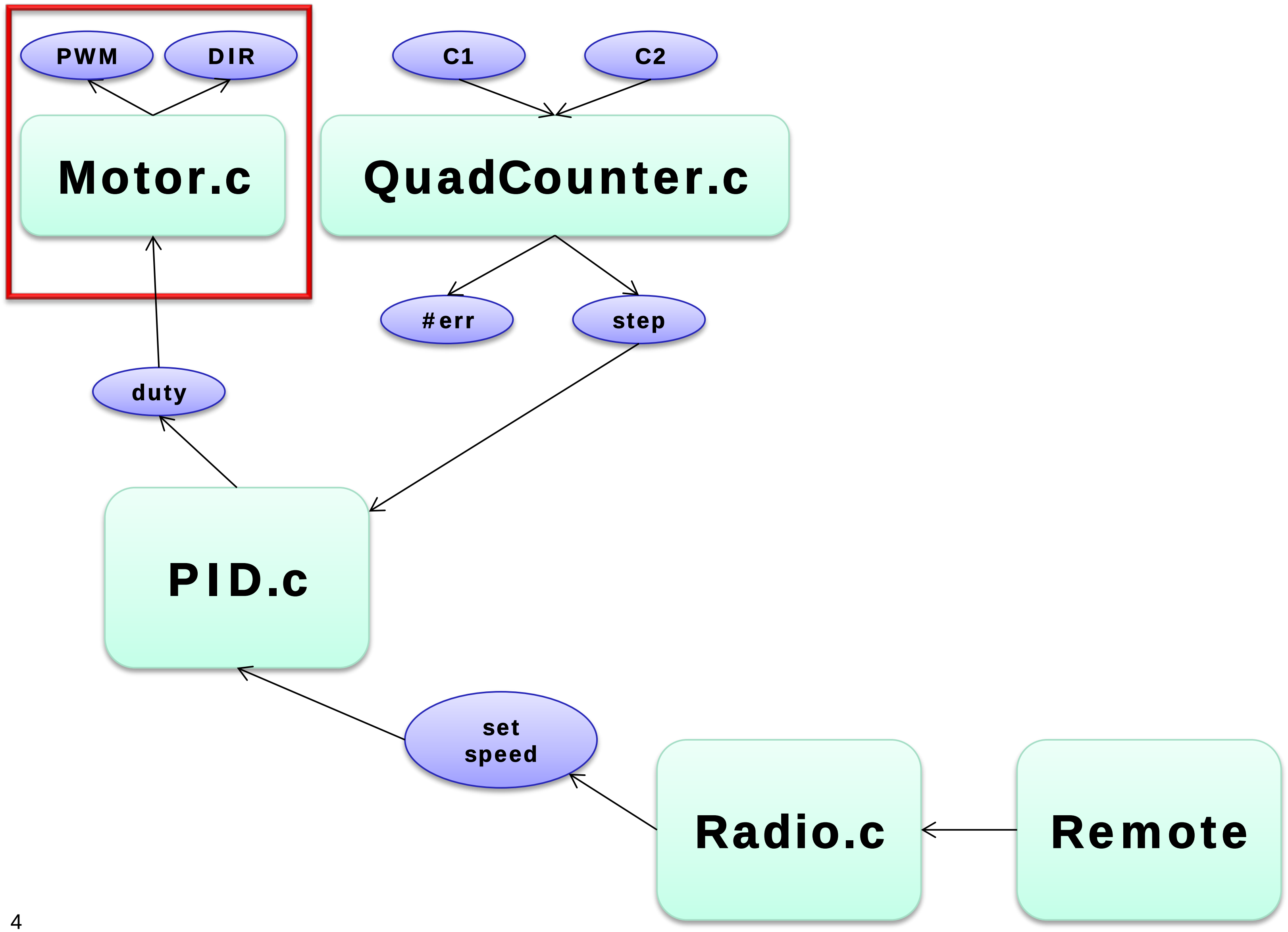


Lucerne University of Applied Sciences and Arts

HOCHSCHULE LUZERN

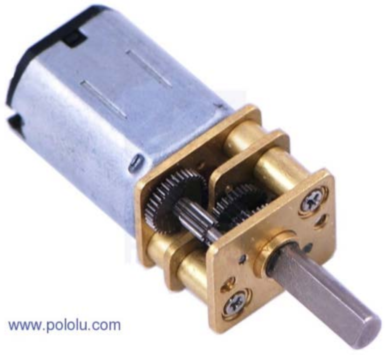
Technik & Architektur

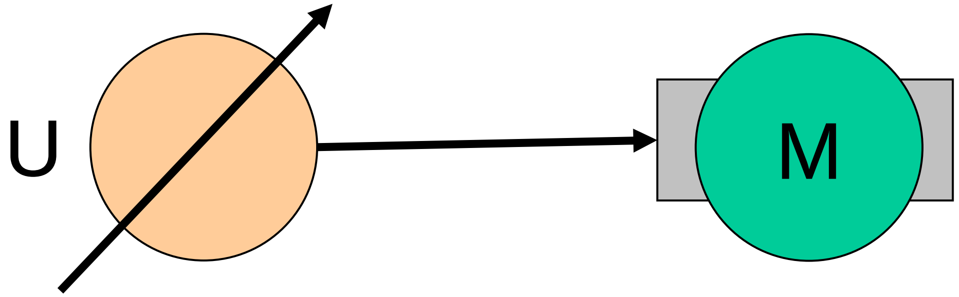
High Level Overview

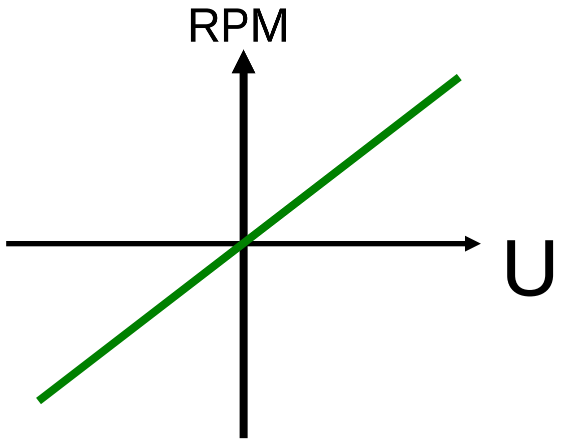


Aktuator

- DC Motor
- Speed proportional to voltage
- Disturbances
    - Mechanical
    - Load
- Goal: Closed Loop Control of Speed/Direction





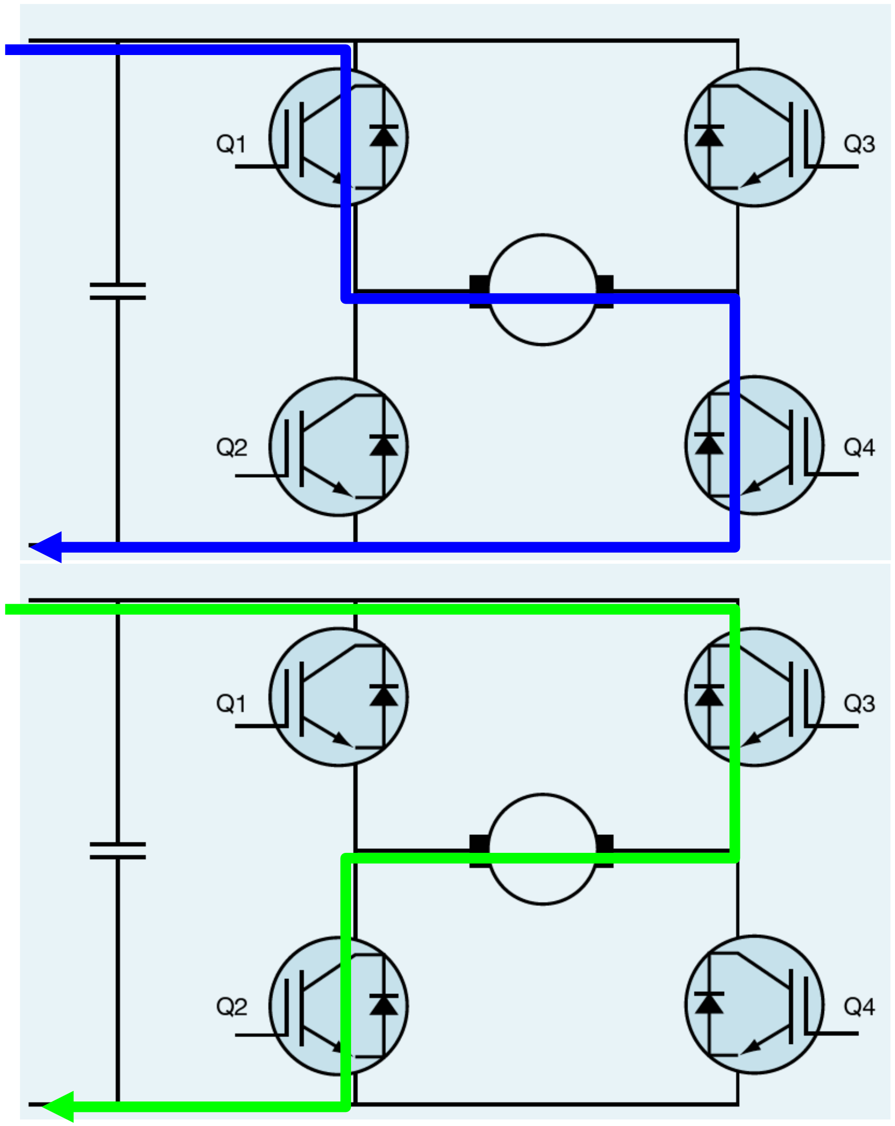


Lucerne University of Applied Sciences and Arts
HOCHSCHULE LUZERN
Technik & Architektur

Digital Full H-Bridge

- Idea
  - 4 Switches
  - Individually controlled
  - Full control
- Needs exact timing
- Switches need to be in sync

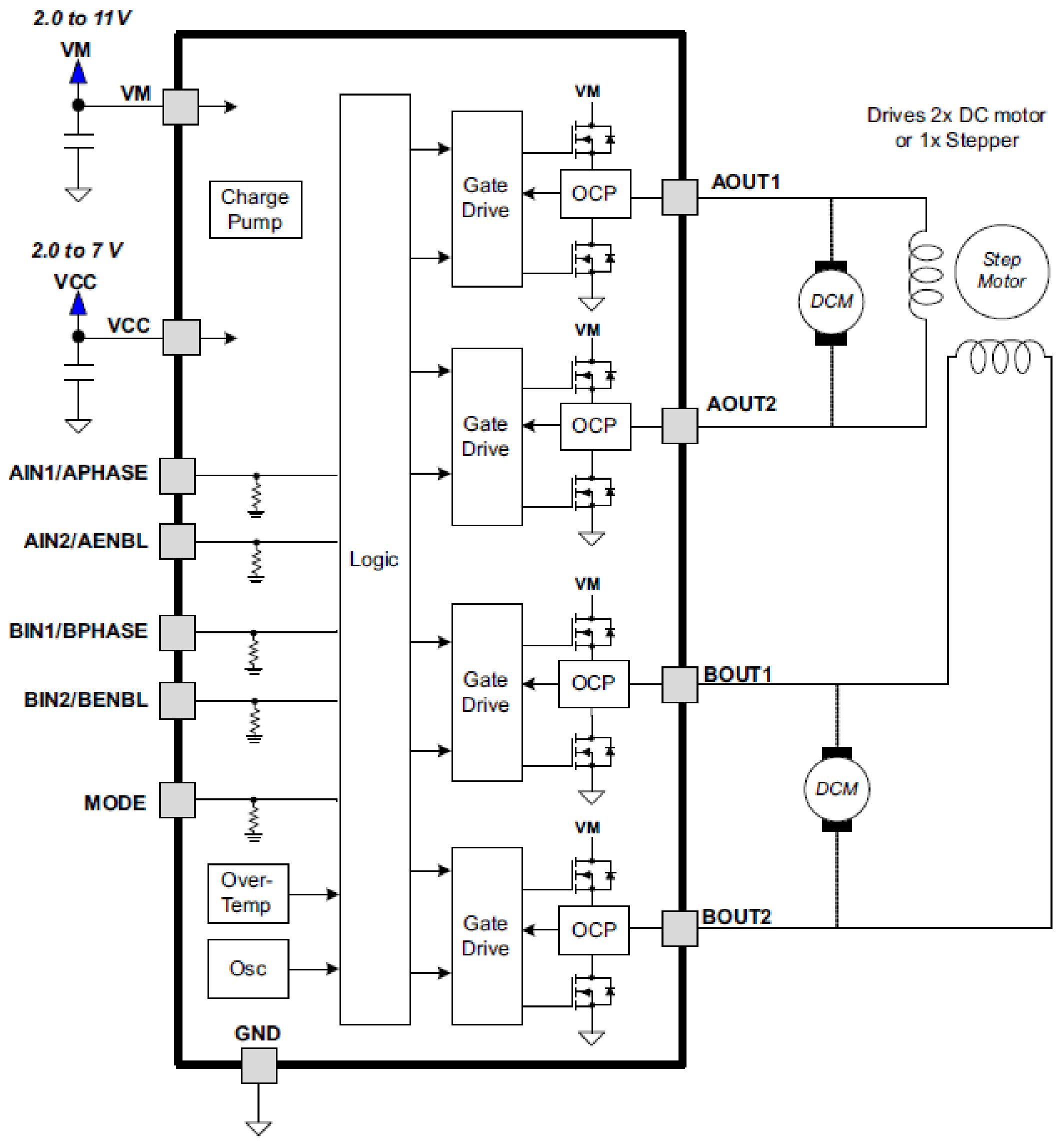
- Motor Driver
  - Direction
  - PWM (Voltage, Speed)
  - Others (emergency stop, etc)



Source: Maxon

TI DRV8835

- Dual-H-Bridge
- 4 MOSFET's
- 1.5 A max per H-Bridge
- (3 A if combined)
- Thermal shutdown:
  1.5 A @ 15 sec (Pololu)
- ➔ realistic: 1.2 A
- PWM'ing: additional
  heating



Source: TI DRV8835 Datasheet

Lucerne University of Applied Sciences and Arts

HOCHSCHULE LUZERN

Technik & Architektur

IN/IN and PHASE/ENABLE

- MODE
  - 0: additional coast mode
  - 1:
    - PWM on ENABLE: speed
    - PHASE: forward/backward

Table 2. IN/IN MODE

MODE	xIN1	xIN2	xOUT1	xOUT2	FUNCTION (DC MOTOR)
0	0	0	Z	Z	Coast
0	0	1	L	H	Reverse
0	1	0	H	L	Forward
0	1	1	L	L	Brake

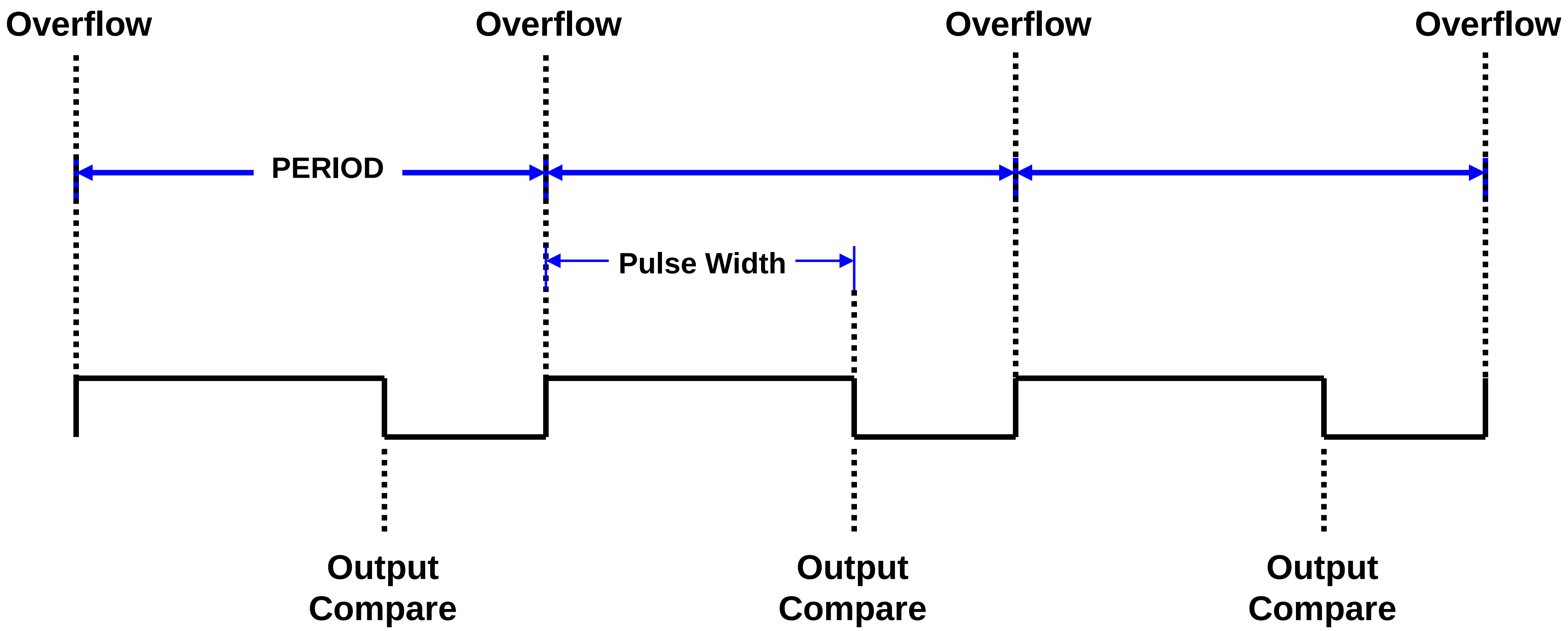
Leerlauf

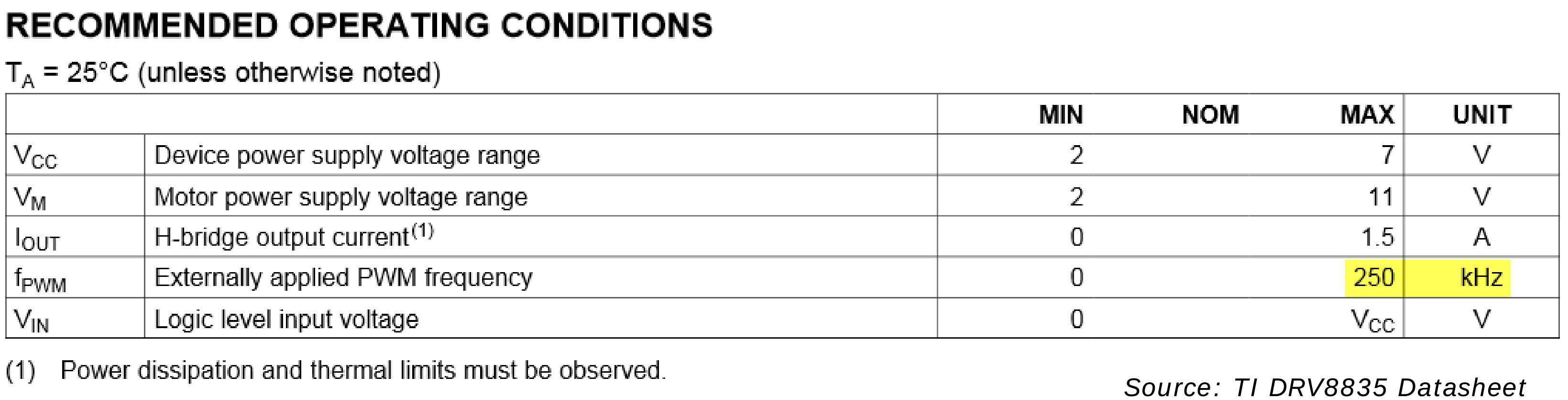
Table 3. PHASE/ENABLE MODE

MODE	xENABLE	xPHASE	xOUT1	xOUT2	FUNCTION (DC MOTOR)
1	0	X	L	L	Brake
1	1	1	L	H	Reverse
1	1	0	H	L	Forward

Source: TI DRV8835 Datasheet

PWM





RECOMMENDED OPERATING CONDITIONS

T_A = 25°C (unless otherwise noted)

		MIN	NOM	MAX	UNIT
V_{CC}	Device power supply voltage range	2		7	V
V_M	Motor power supply voltage range	2		11	V
I_{OUT}	H-bridge output current[1]	0		1.5	A
f_{PWM}	Externally applied PWM frequency	0		250	kHz
V_{IN}	Logic level input voltage	0		V_{CC}	V

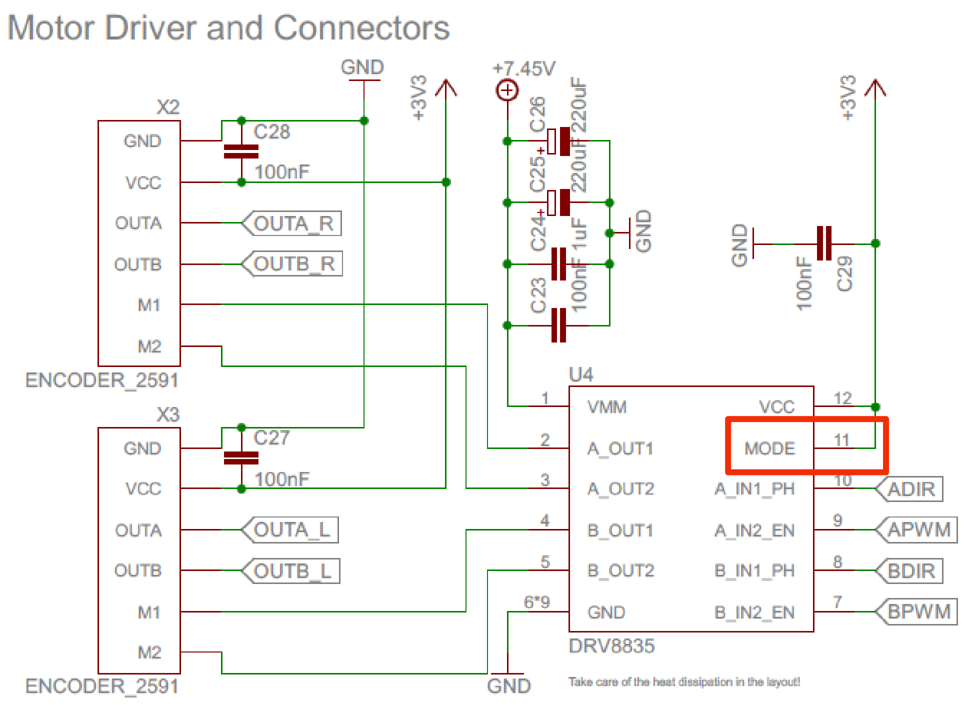
(1) Power dissipation and thermal limits must be observed.

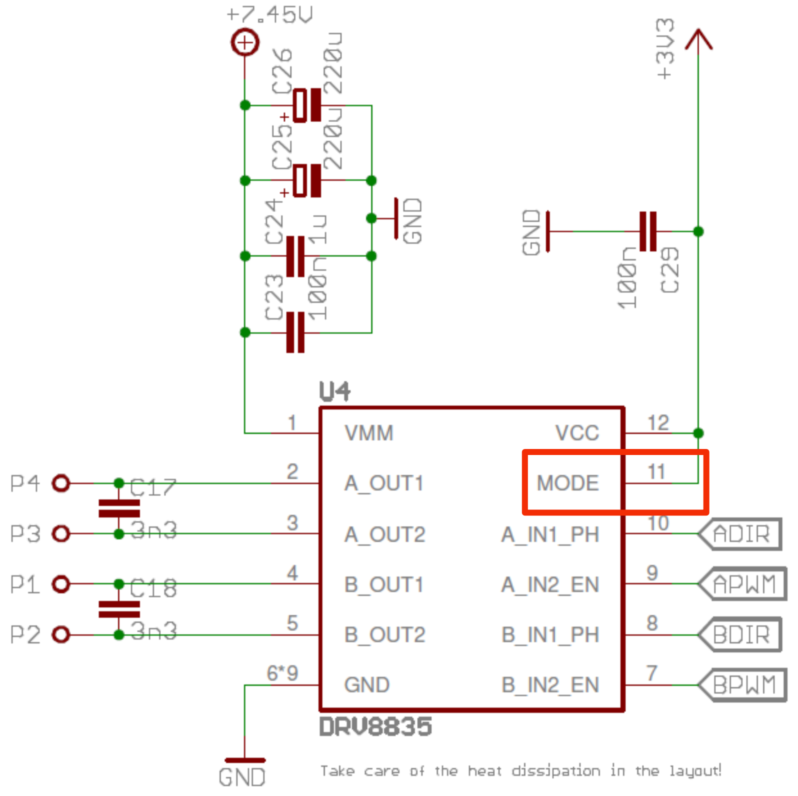
Source: TI DRV8835 Datasheet

Lucerne University of Applied Sciences and Arts
HOCHSCHULE LUZERN
Technik & Architektur

Motor Driver Schematic (V1 / V2 Robot)

- MODE set to HIGH: PWM & DIR mode

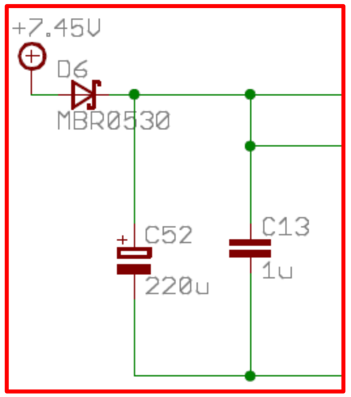




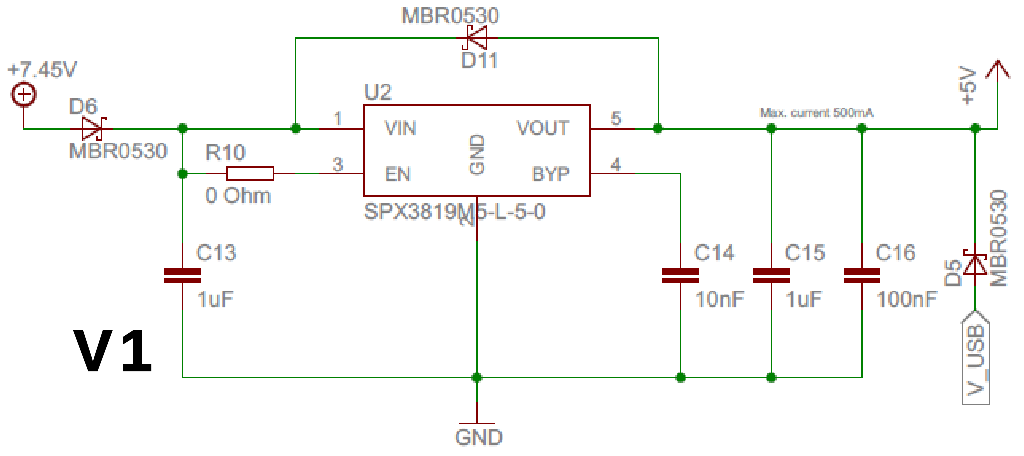
5V Buffers

- Issue: high current might drop VBAT/7.45V Voltage
- V1 Robot: 5V Output might drop for high motor current

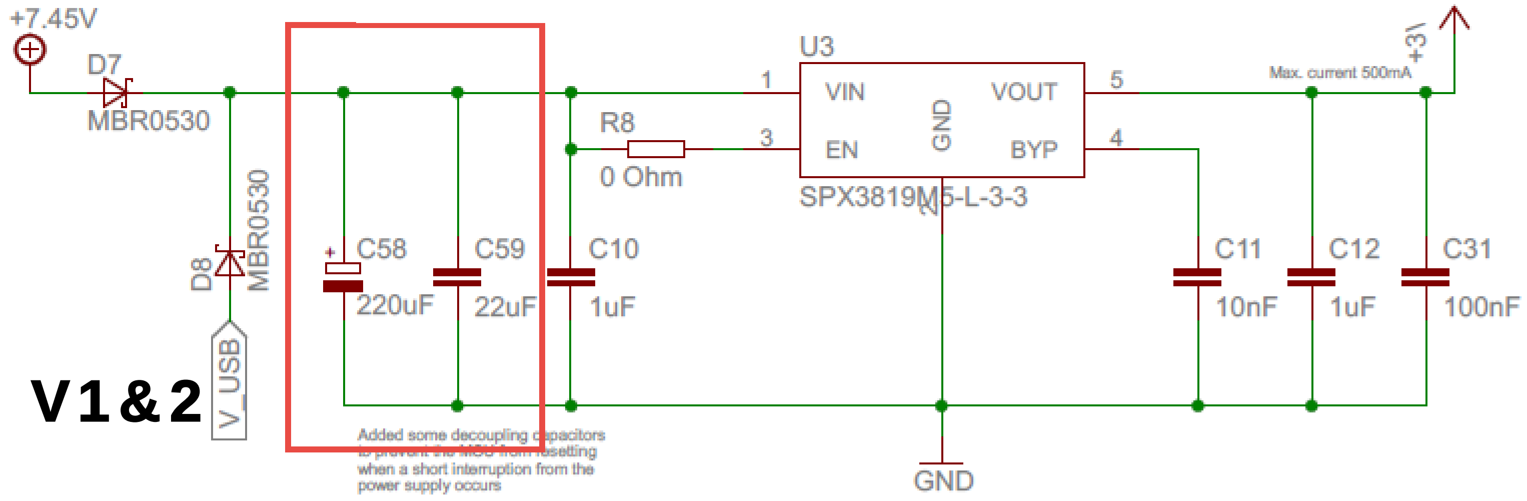
V2



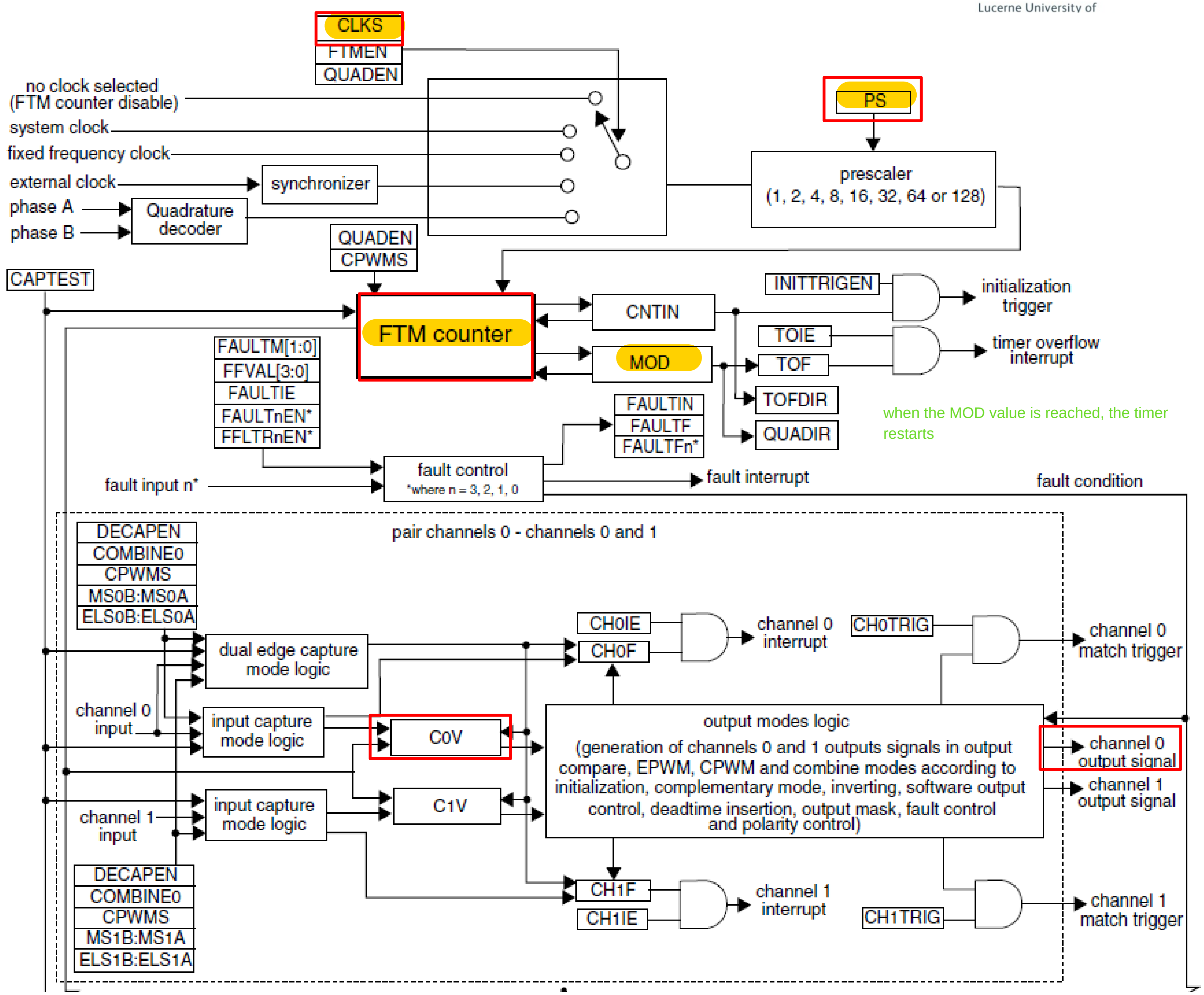
V1



V1&2



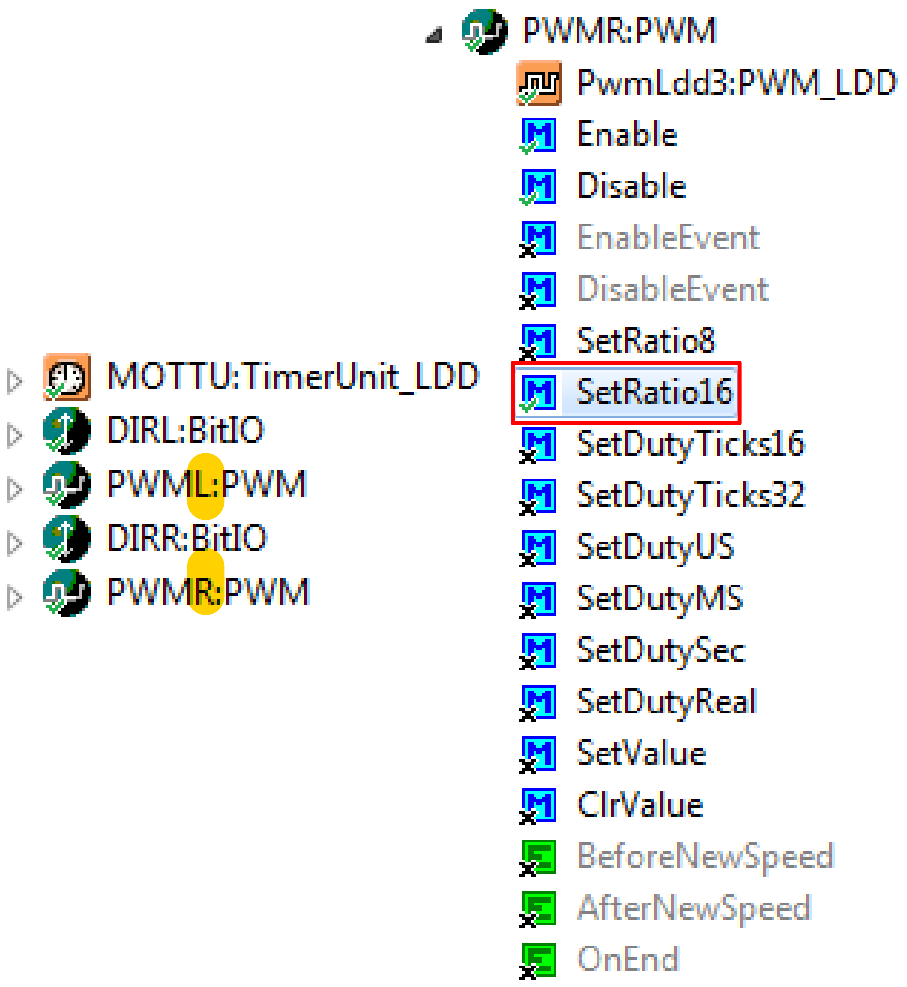
TPM



when the MOD value is reached, the timer restarts

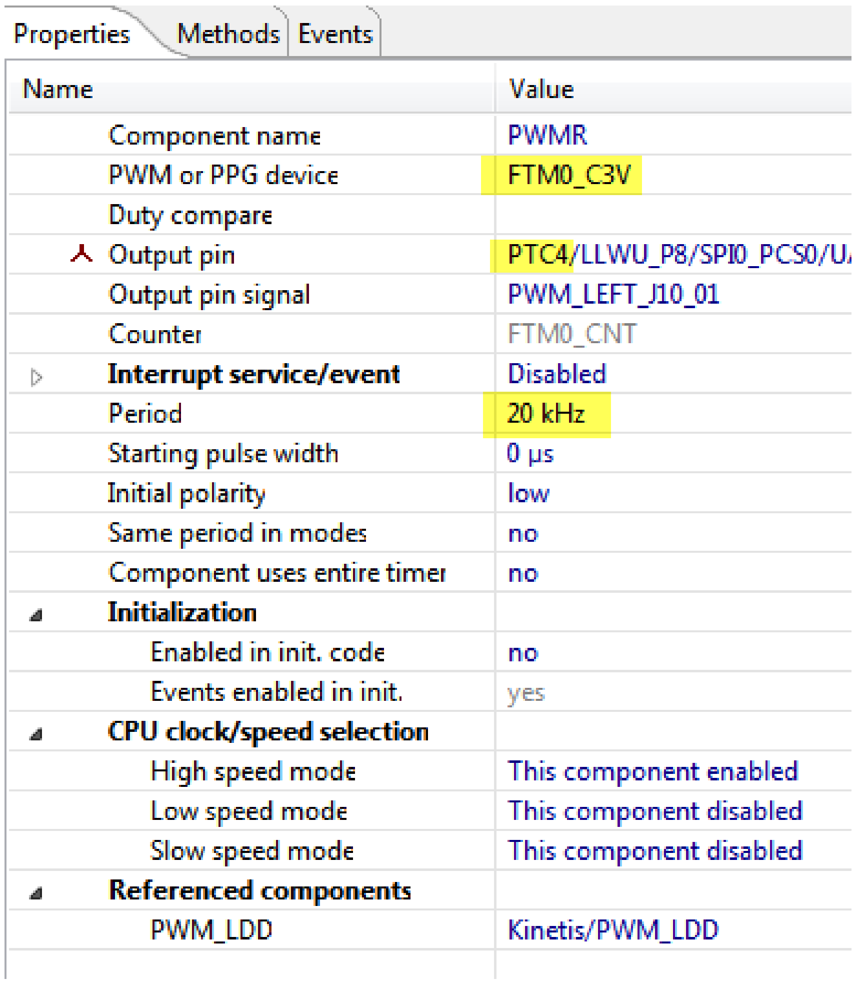
Lucerne University of Applied Sciences and Arts
HOCHSCHULE LUZERN
Technik & Architektur

Processor Expert PWM



MOTTU:TimerUnit_LDD
DIRL:BitIO
PWML:PWM
DIRR:BitIO
PWMR:PWM

PWMR:PWM
- PwmLdd3:PWM_LDD
- Enable
- Disable
- EnableEvent
- DisableEvent
- SetRatio8
- SetRatio16
- SetDutyTicks16
- SetDutyTicks32
- SetDutyUS
- SetDutyMS
- SetDutySec
- SetDutyReal
- SetValue
- ClrValue
- BeforeNewSpeed
- AfterNewSpeed
- OnEnd



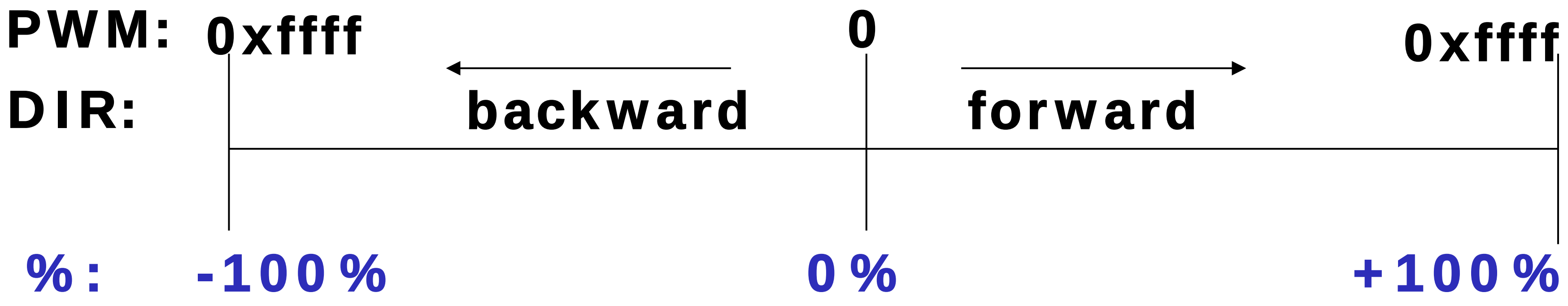
Name	Value
Component name	PWMR
PWM or PPG device	FTM0_C3V
Duty compare	
Output pin	PTC4/LLWU_P8/SPI0_PCS0/U
Output pin signal	PWM_LEFT_J10_01
Counter	FTM0_CNT
Interrupt service/event	Disabled
Period	20 kHz
Starting pulse width	0 µs
Initial polarity	low
Same period in modes	no
Component uses entire timer	no
Initialization	
Enabled in init. code	no
Events enabled in init.	yes
CPU clock/speed selection	
High speed mode	This component enabled
Low speed mode	This component disabled
Slow speed mode	This component disabled
Referenced components	
PWM_LDD	Kinetis/PWM_LDD

Motor Interface

```
typedef enum {
  MOT_DIR_FORWARD,  /*!< Motor forward direction */
  MOT_DIR_BACKWARD  /*!< Motor backward direction */
} MOT_Direction;

typedef enum {
  MOT_MOTOR_LEFT, /*!< left motor */
  MOT_MOTOR_RIGHT /*!< right motor */
} MOT_MotorSide;


MOT_MotorDevice *MOT_GetMotorHandle(MOT_MotorSide side);

void MOT_SetDirection(MOT_MotorDevice *motor, MOT_Direction dir);

void MOT_SetSpeedPercent(MOT_MotorDevice *motor, int8_t percent);
```

DIR & PWM Mapping

- Hardware
  - PWM duty cycle ratio (0x0000..0xffff)
  - DIR (binary signal) (0 or 1)
- Software
  - % speed (-100% ... +100%)
  - PWM (0x0000-0xffff)
  - Dir (boolean)
- Use ***single*** representation for data/state?



Lucerne University of Applied Sciences and Arts

HOCHSCHULE LUZERN

Technik & Architektur

Motor Direction (Motor.c)

```
typedef struct MOT_MotorDevice_ {
  MOT_SpeedPercent currSpeedPercent;
  uint16_t currPWMvalue;
  uint8_t (*SetRatio16)(uint16_t);
  void (*DirPutVal)(bool);
} MOT_MotorDevice;
```

Use Pin value instead?

```
MOT_Direction MOT_GetDirection(MOT_MotorDevice *motor) {
  if (motor->currSpeedPercent<0) {
    return MOT_DIR_BACKWARD;
  } else {
    return MOT_DIR_FORWARD;
  }
}
```

Motor Direction (Motor.c)

```
void MOT_SetDirection(MOT_MotorDevice *motor, MOT_Direction
  dir) {
   if (dir==MOT_DIR_BACKWARD) {
     motor->DirPutVal(1);
     if (motor->currSpeedPercent>0) {
       motor->currSpeedPercent = -motor->currSpeedPercent;
     }
   } else if (dir==MOT_DIR_FORWARD) {
     motor->DirPutVal(0);
     if (motor->currSpeedPercent<0) {
       motor->currSpeedPercent = -motor->currSpeedPercent;
     }
   }
}
```

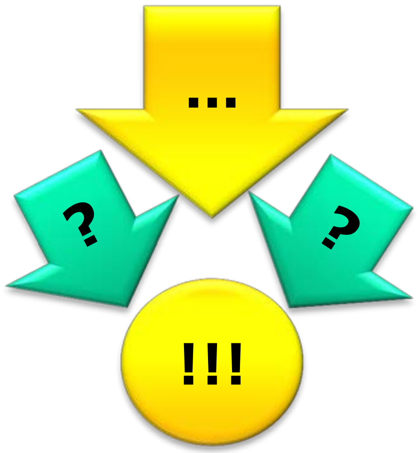
Optimize this?

accessing the pin can be quite expensive, Optimierung: If we are already in this direction, don't write it again

What about PWM?

Summary

- Motor Platform Signals
  - DIR
  - PWM and Timer channels
  - Dual H-Bridge
- Realtime aspects
- Concept of device handle
- Data vs. Pin value
  - Data consistency
  - Access/Reentrancy



Lab: Motor Signals

- Add/Enable components for Motors
  - BitIO: Direction
  - PWM: Speed
- Motor.c/.h
  - Check/Change
    - SetDirection()
    - GetDirection()
    - SetSpeedPercent()
  - Shell support
    - Direction (forward, backward)
    - Duty (%)