



Setting up a Mechatronic System

4th Training in Rio de Janeiro, BRA

6th-9th of May 2019

Thomas Lechner



"The European Commission support for the production of this publication does not constitute an endorsement of the contents which reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein."

FOR EDUCATIONAL PURPOSE ONLY

Co-funded by the
Erasmus+ Programme
of the European Union



Agenda

- Implementing a new ECU
 - Introduction
 - Input- and Output-Signals
 - Hardware and Software Setup
 - Programming a ECU with Simulink
 - Controller Setup with Ziegler-Nichols
 - Live Demo via Skype



Choosing the ECU

Interfaces

- Speed Controller
 - Motor Speed (Input)
 - DC-Motor terminal voltage (Output)
- Position Controller
 - Rotor position (Input)
 - Motor speed and **direction** (Output → desired value for speed controller)
- DC-Motor load torque
 - Estimated via DC-Motor current



Choosing the ECU

Interfaces

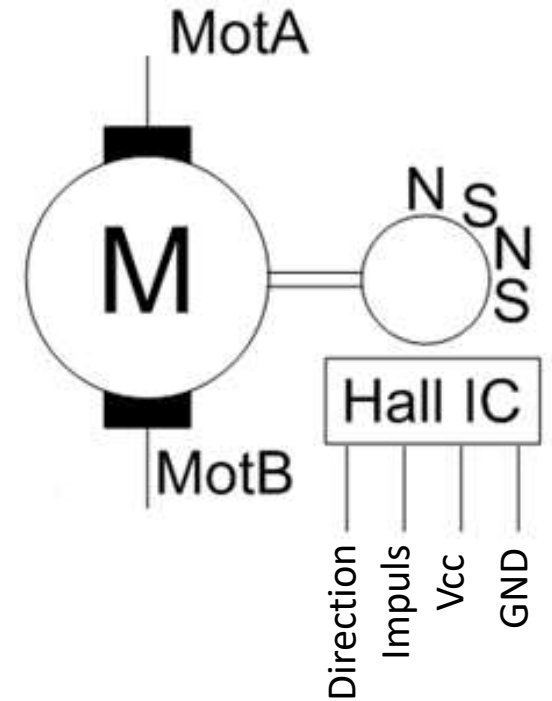
- Communication between ECU and environment
 - CAN-Interface
- ECU application
 - Can Calibration Protocol (CCP)



Choosing the ECU

Speed Measurement

- **DC-Motor → 10 Magnets**
 - Hall-Sensor measures
 - Rotor position (Input)
 - Motor speed and **direction** (Output → desired value for speed controller)
- **DC-Motor load torque**
 - Estimated via DC-Motor current



Choosing the ECU – Speed Measurement

$\tau \approx 1.3 \text{ ms} \rightarrow n \approx 2308 \text{ RPM}$

Speed measurement with timer input:
input:

$$f = \frac{1}{\tau}$$

f ... Frequency in Hz

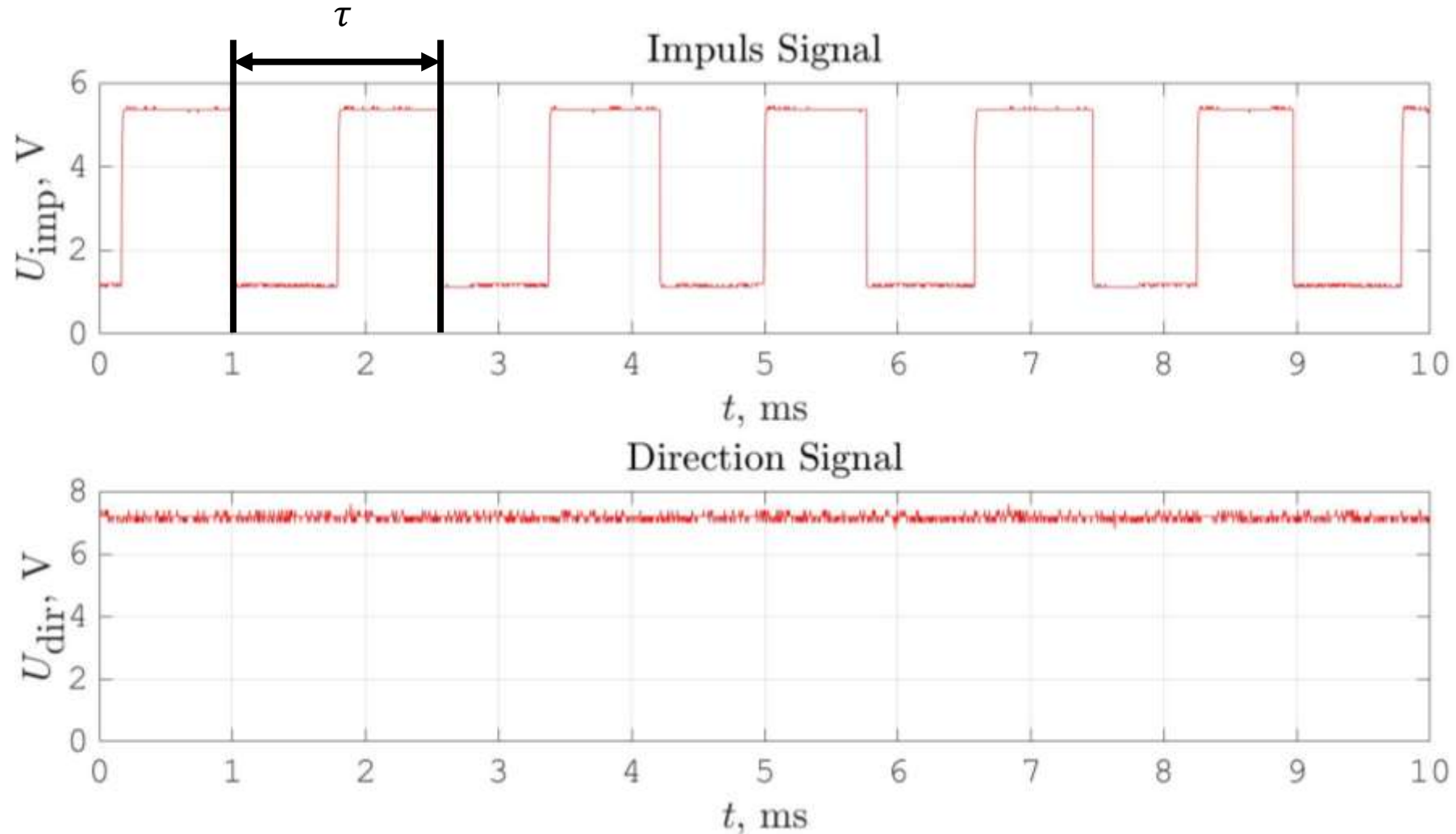
τ ... Period time in s

$f \rightarrow$ Measurement value

$$n = \frac{f}{N} \cdot 60$$

n ... engine speed in RPM

N ... Number of increments per revolution. In our case, $N=20$.



Choosing the ECU – Direction Measurement

Direction measurement with a digital input:

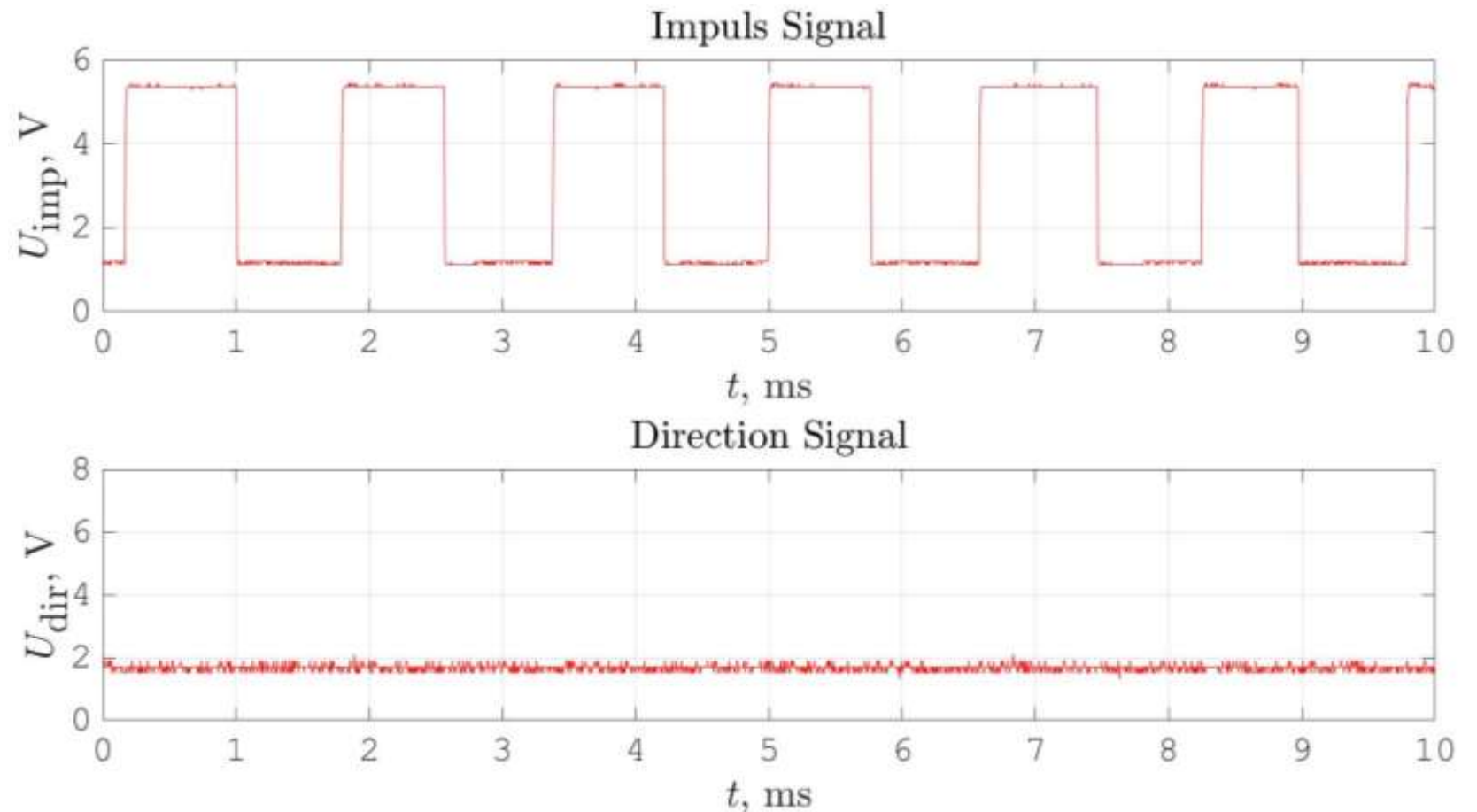
$U_{\text{dir}} \cong 1.9 \text{ V} \rightarrow$ logical 0

$U_{\text{dir}} \cong 5.5 \text{ V} \rightarrow$ logical 1

Direction of rotation:

1 \rightarrow clockwise

0 \rightarrow counterclockwise



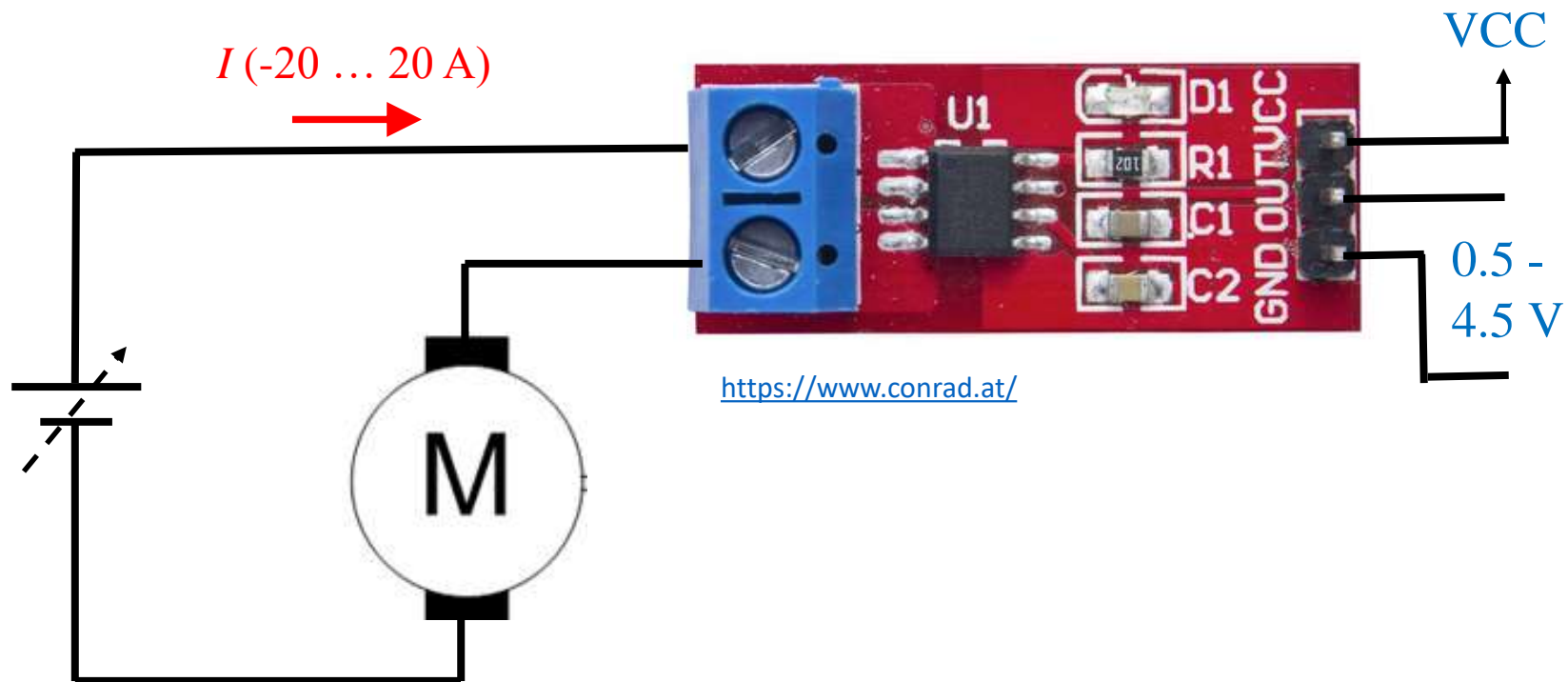
Choosing the ECU-Electrical Current Measurement

Current Measurement with a Hall-Sensor:

Supply Voltage (VCC) → 5 V

-20 A	→	0.5 V
0 A	→	2.5 V
20 A	→	4.5 V

For DAQ → Analog Input



Choosing the ECU – DC-Motor connection

DC-Motor terminal voltage

- The voltage must be variable to change the motor-speed
- The voltage must change the polarity to change the direction
- Maximum (minimum) DC-motor current is ± 12 A

PWM modulated Voltage

H-Bridge



<http://www.hessmer.org/blog/2013/12/28/ibt-2-h-bridge-with-arduino>



- | | |
|---------|--|
| 1、 RPWM | : Forward level or PWM signal input, active high |
| 2、 LPWM | : Inversion level or PWM signal input, active high |
| 3、 R_EN | : Forward drive enable input , high enable , low close |
| 4、 L_EN | : Reverse drive enable input , high enable , low close |
| 5、 R_IS | : Forward drive –side current alarm output |
| 6、 L_IS | : Reverse drive –side current alarm output |
| 7、 VCC | : +5 V power input, connected to the microcontroller 5V power supply |
| 8、 GND | : Signal common ground terminal |



ECU - Input/Output Overview

	Quantity	Range	Description
CAN	>= 1	500 kBaud	- ECU flashing - Communication with environment, dynamic measurement ...
Sensor Supply	1	5 V	- Current transducer - H-Bridge
Sensor Supply	1	10 V	- DC-Motor speed/position sensor
Voltage out 5 V	1	0 - 5 V	- H-Bridge → PWM enable
PWM out	2	10 kHz, 0 – 100 %, 0 – 5 V	- DC-Motor direction clockwise - DC-Motor direction counter clockwise
Timer in	1	2000 Hz	- DC-Motor speed
Digital in	1	1.9 V → logical 0 5.5 V → logical 1	- DC-Motor direction
Analog in	1	5 V	- Current transducer
Counter in	1	1.9 V → logical 0 5.5 V → logical 1	- DC-Motor position



ECU – Required Performance

Performance Requirement

- Minimum cycle time: 2 ms
 - This is an empirical value, estimated according to the expertise we have with a similar application. The cycle time influences the controller performance.
- Automatic software-generation out of Simulink
 - State of the art method. (language C is not longer part of our curriculum)
- Calibration via XCP or CCP
 - State of the art method for development, parameter setting, debugging ...
- Calculation with Floating Points (single, double, ...)
 - Knowledge about Integer-Arithmetic is not so important for an system engineer.



ECU - Our Choice

HY-TTC 510 from TT-Tech

Key Benefits:

- 32 bit dual-core CPU with 180MHz
- Floating-point unit
- 12 Bit ADC
- PWM-Outputs
- Digital in an Outputs
- CAN, CCP



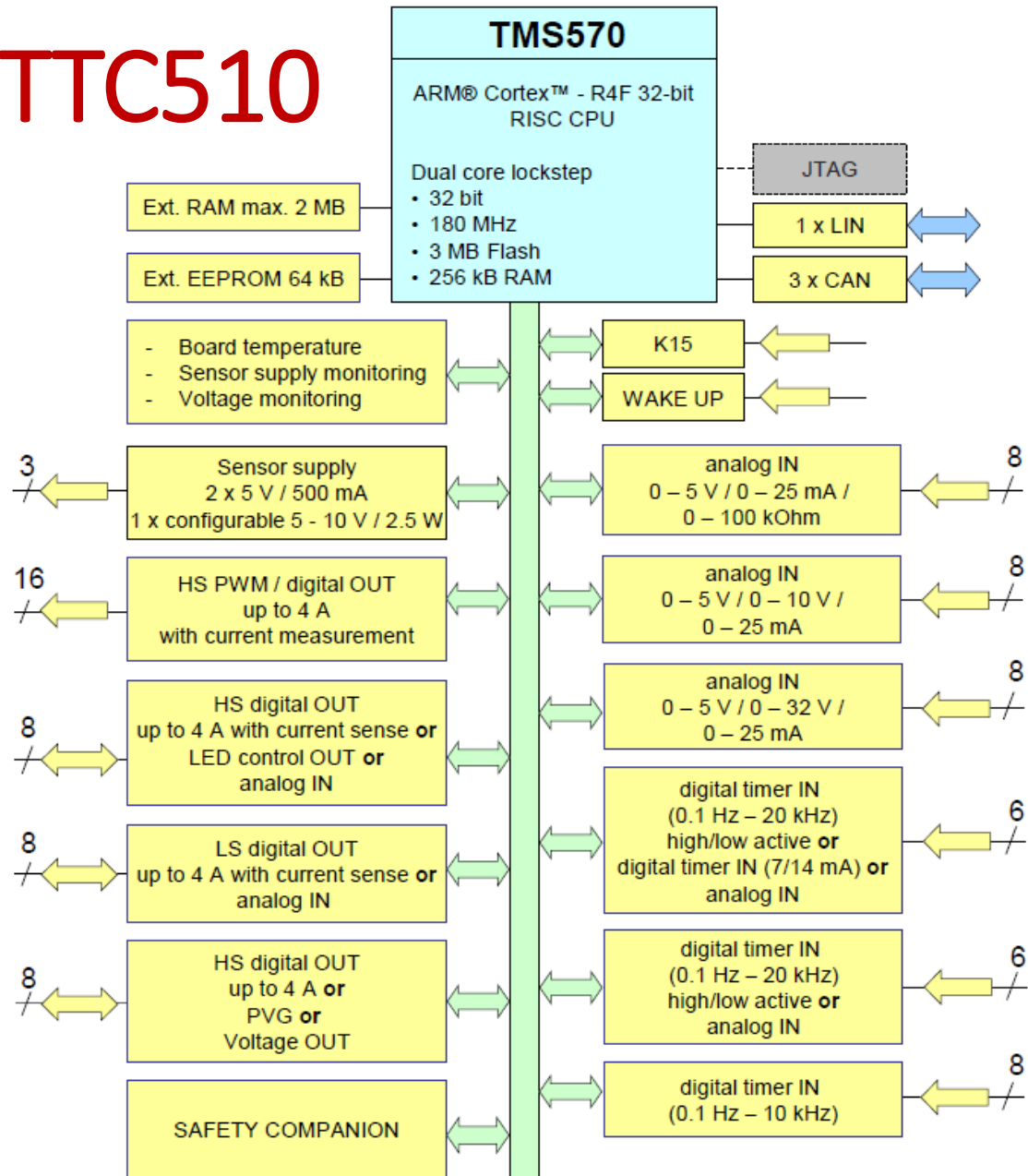
<https://www.ttcontrol.com>



Co-funded by the
Erasmus+ Programme
of the European Union



Description TTC510



ECU – Target-performance comparison

	Quantity	Range	Possible with HY TTC 510?
CAN	~2	500 kBaud	- Yes (3 CAN-Interfaces available)
Sensor Supply	1	5 V	- Yes (2 x 5 V supply on board)
Sensor Supply	1	10 V	- Yes (1 x programmable between 5 V an 10 V)
Voltage out 5 V	1	0 - 5 V	- Yes
PWM out	2	10 kHz 0 – 100 % 0 – 5 V	- No (maximum 1 kHz) - Yes - Yes/No → Voltage level must be adapted (voltage divider)
Timer in	1	2000 Hz	- Yes (maximum 20 kHz)
Digital in	1	1.9 V → logical 0 5.5 V → logical 1	- Yes
Analog in	1	5 V	- Yes
Counter in	1	1.9 V → logical 0 5.5 V → logical 1	- Yes (for Simulink, a Workaround is necessary)

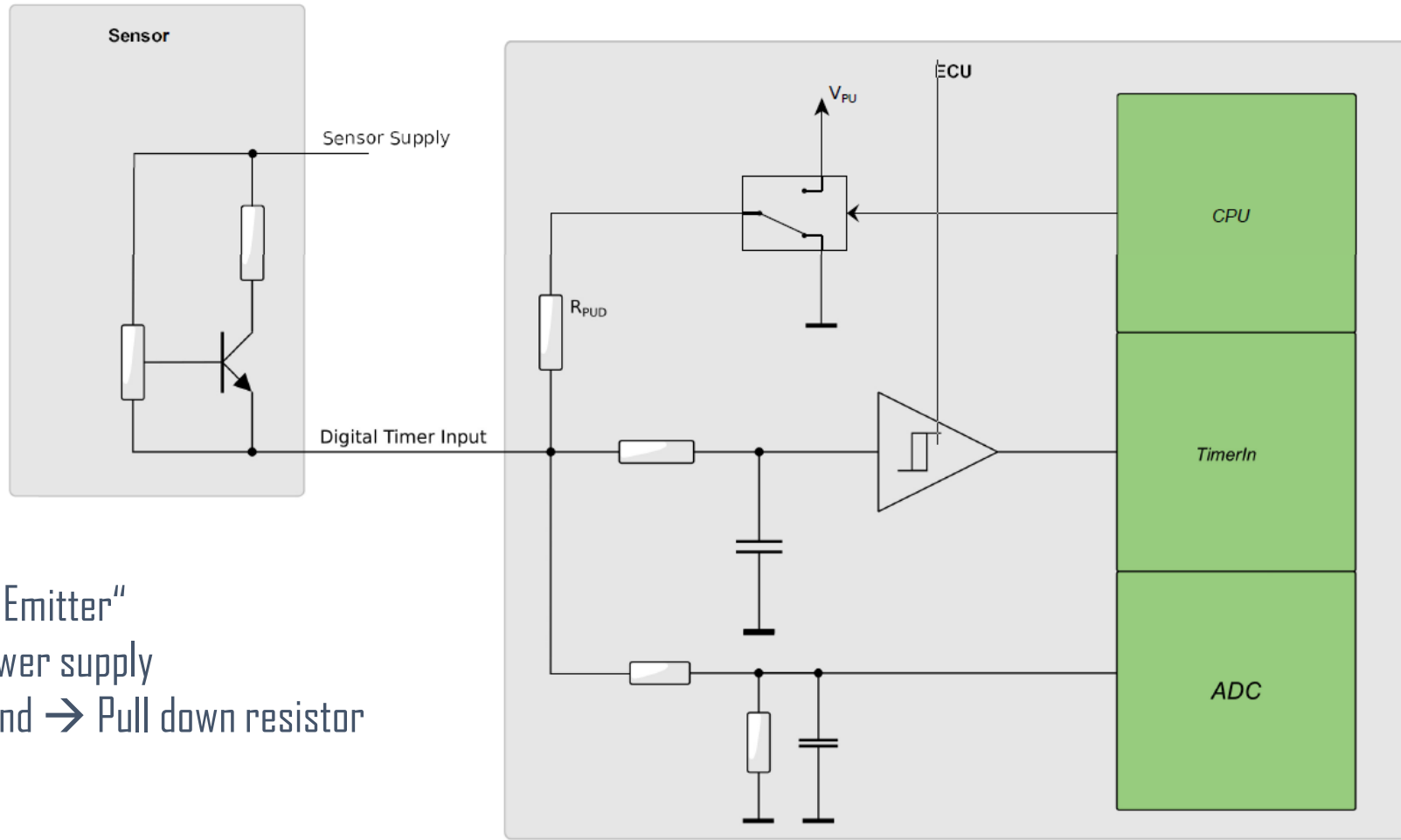


ECU – Target-performance comparison

- Minimum cycle time: 2 ms
 - **OK.** The cycle time can be adjusted in discreet steps. The minimum value is 1 ms.
- Automatic Software generation out of Simulink
 - **OK.** A Simulink-Library is included in the scope of delivery. A basic description, for correct solver settings is available.
- Calibration via XCP or CCP
 - **OK.** CCP is supported in the polling mode.
- Calculation with Floating Points (single, double, ...)
 - **OK.** The μ P has a FPU on board.



ECU – Circuit diagram of Input's



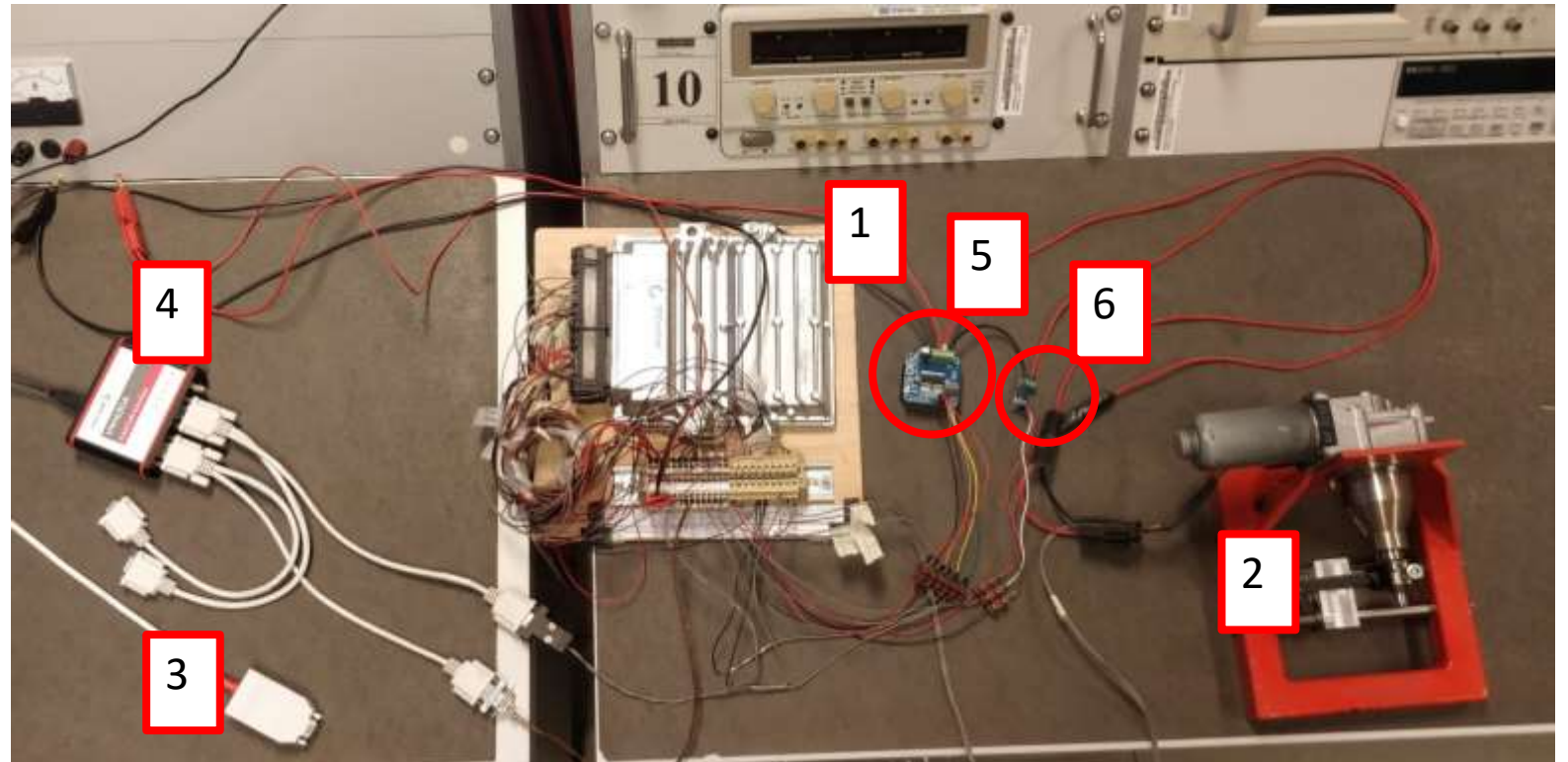
Sensor output is „Open Emitter“

- External sensor power supply
- Connection to ground → Pull down resistor

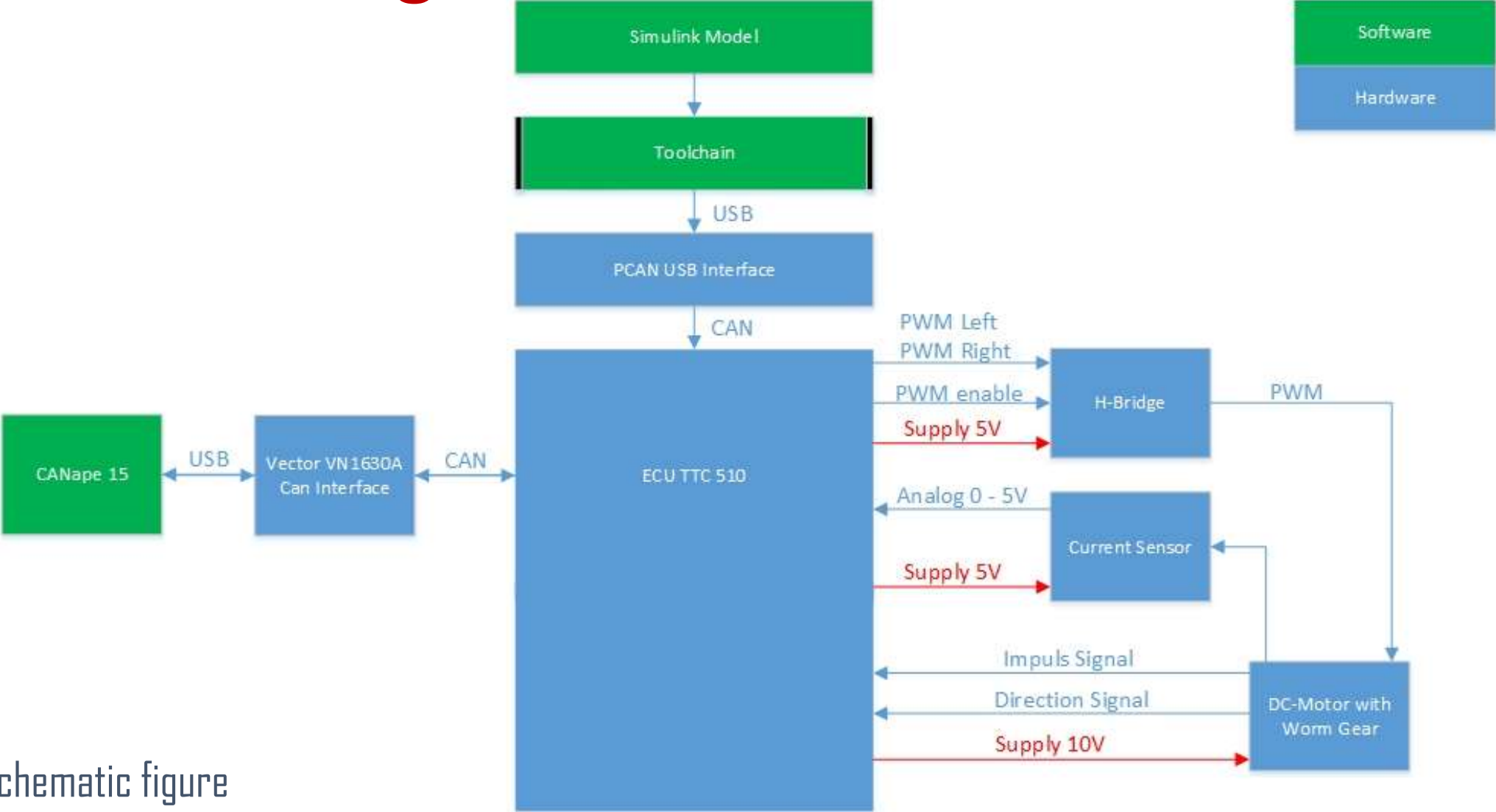


System overview

- 1) ECU HY-TTC 510
- 2) Device under Test (DUT)
- 3) PCAN-USB Interface for flashing
- 4) Vector VN1630 USB to CAN Interface for application (CCP) and measurement
- 5) H-Bridge
- 6) Current transducer



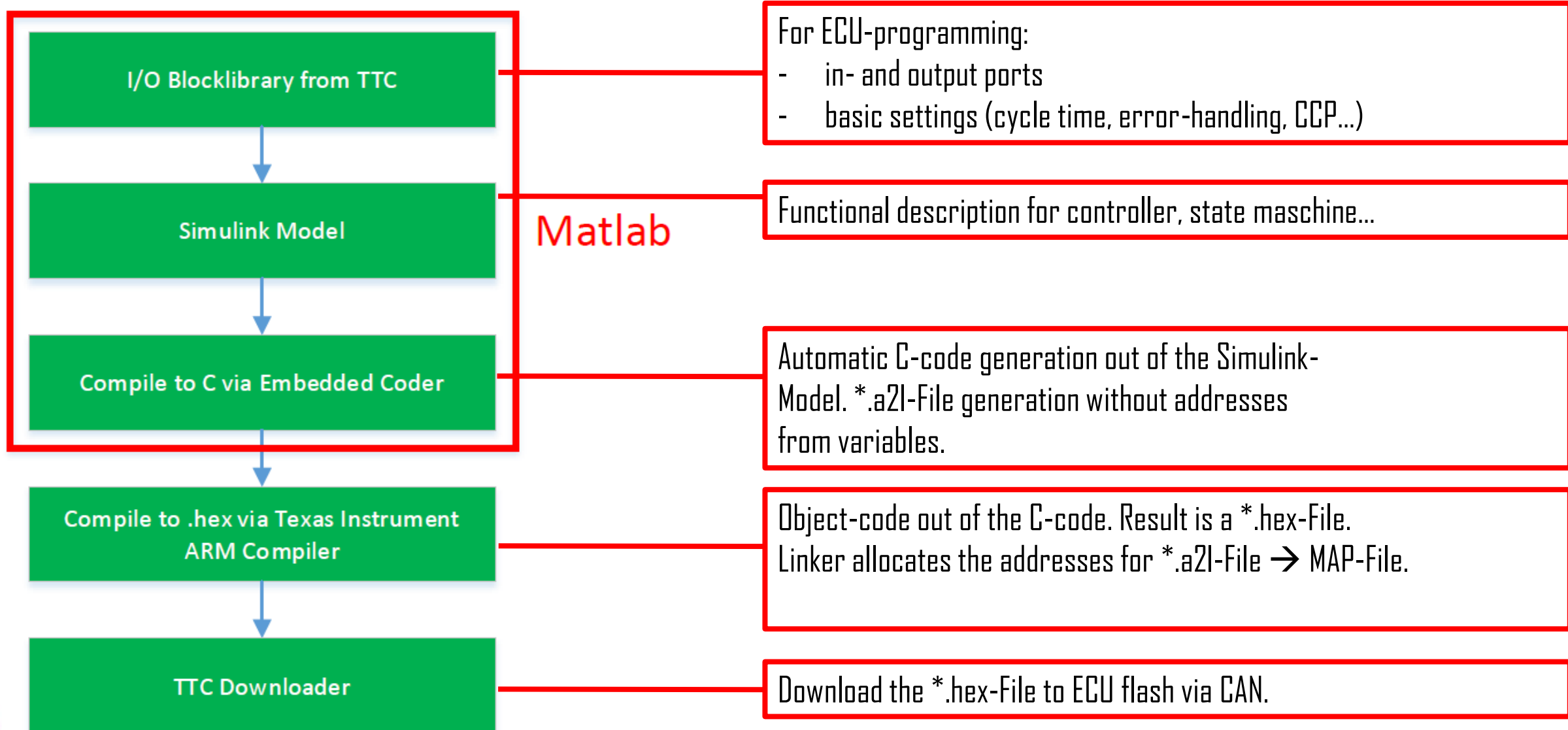
System block diagram



Schematic figure



Tool chain description



Matlab

For ECU-programming:

- in- and output ports
- basic settings (cycle time, error-handling, CCP...)

Functional description for controller, state machine...

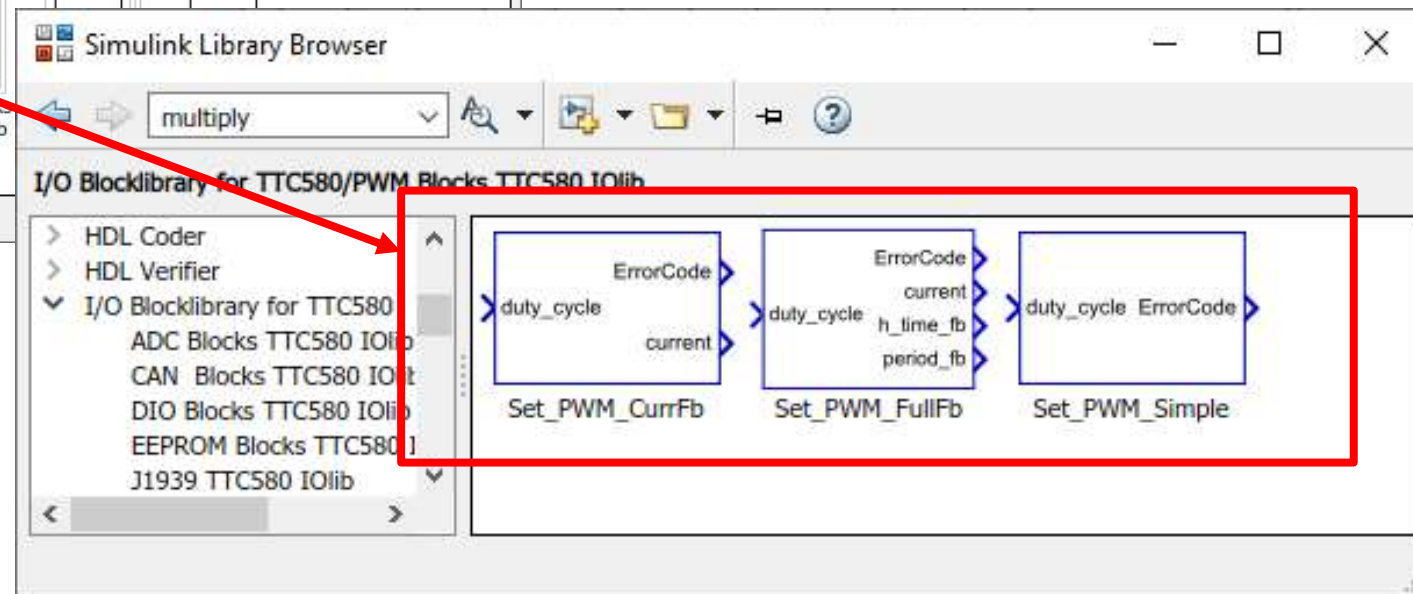
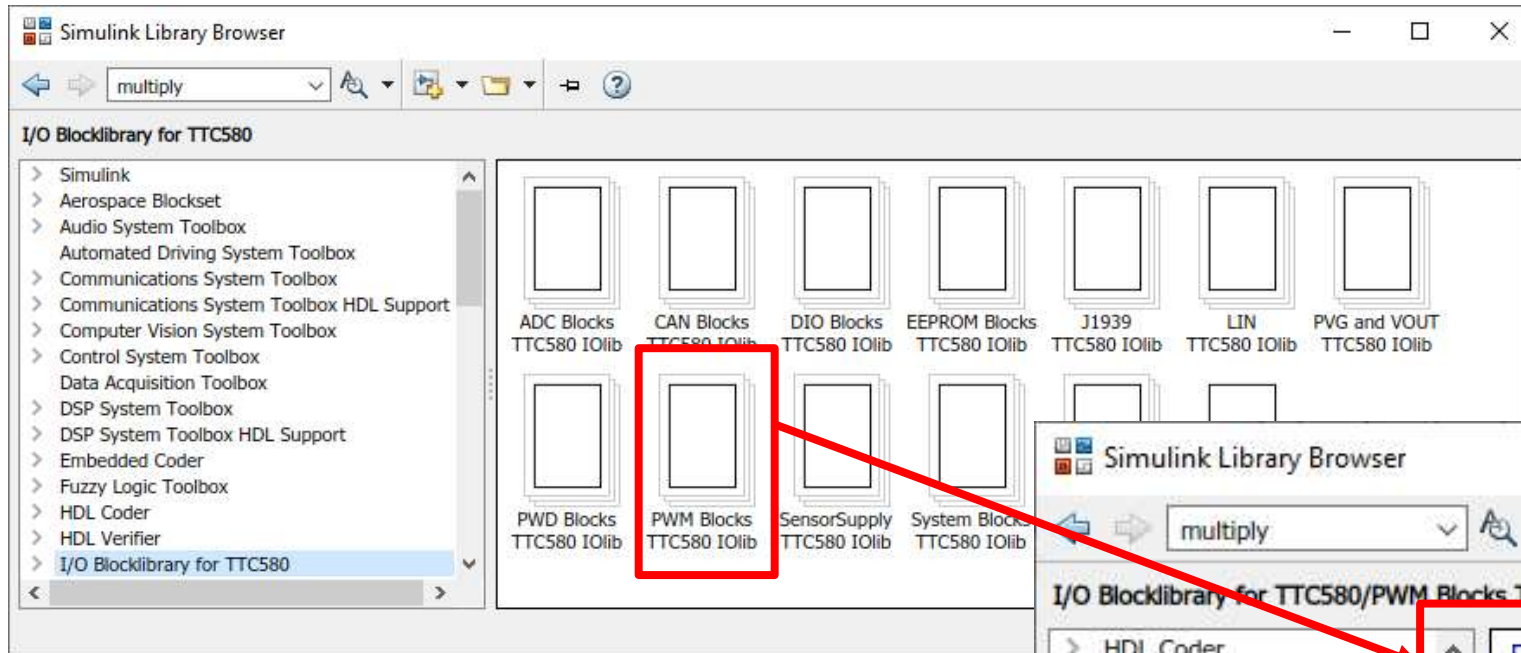
Automatic C-code generation out of the Simulink-Model. *.a2l-File generation without addresses from variables.

Object-code out of the C-code. Result is a *.hex-File. Linker allocates the addresses for *.a2l-File → MAP-File.

Download the *.hex-File to ECU flash via CAN.



TTC IO-Library



The IO-Library

- Developed from TTTech
- included in scope of delivery



A simple Simulink example

Change PWM ratio as a function of a voltage signal

Global Settings for the ECU → Block *MainDlg*

Setup for:

- CAN Baudrate (max. 1000 kHz)
- Cycle (Duration) time
- CCP Addresses

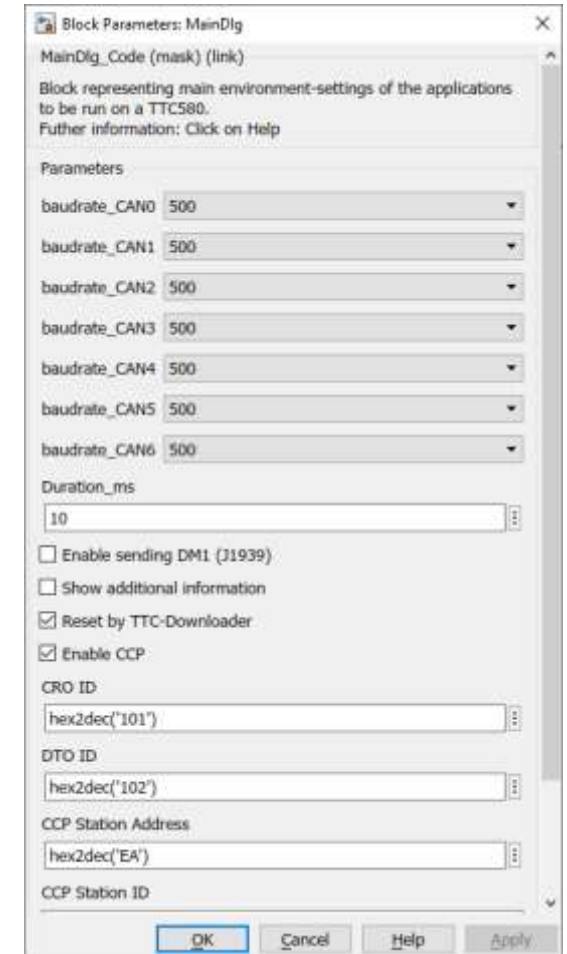
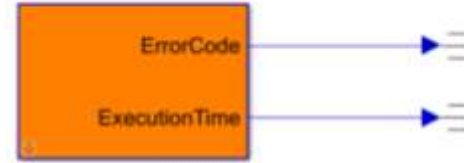
Power outputs must be enabled

- Block *Power_Enable*

0 → disable

1 → enable

Data type: Boolean



A simple Simulink example

Change PWM ratio as a function of a voltage signal

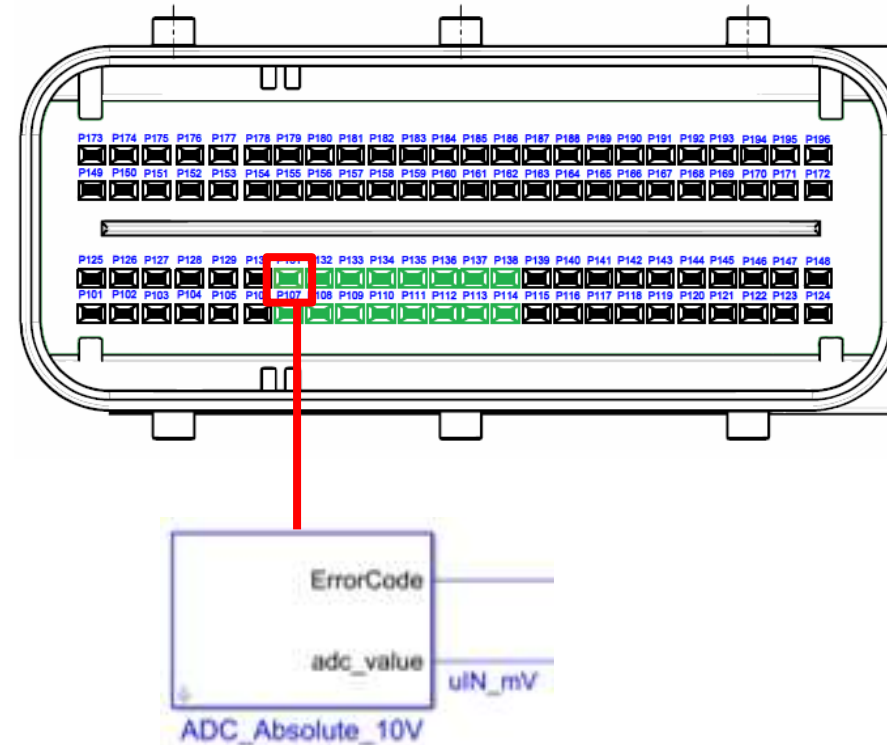
Input: Voltage Signal

Choosing an Analog-Input port → Block
ADC_Absolute_10V

Choose the input port that fits to the connector
pinning:

Pin 131 is connected → *IO_ADC_09*

For more infos see [1] 4.10 Analog Input 2
Modes



Pin No.	Function 1	Function 2	SW-define
P107	Analog 0... 5 V, 0... 10 V Input	Analog 0... 25 mA Input	IO_ADC_08
P131	Analog 0... 5 V, 0... 10 V Input	Analog 0... 25 mA Input	IO_ADC_09
P108	Analog 0... 5 V, 0... 10 V Input	Analog 0... 25 mA Input	IO_ADC_10
P132	Analog 0... 5 V, 0... 10 V Input	Analog 0... 25 mA Input	IO_ADC_11
P109	Analog 0... 5 V, 0... 10 V Input	Analog 0... 25 mA Input	IO_ADC_12



A simple Simulink example

Change PWM ratio as a function of a voltage signal

Output: PWM-Signal

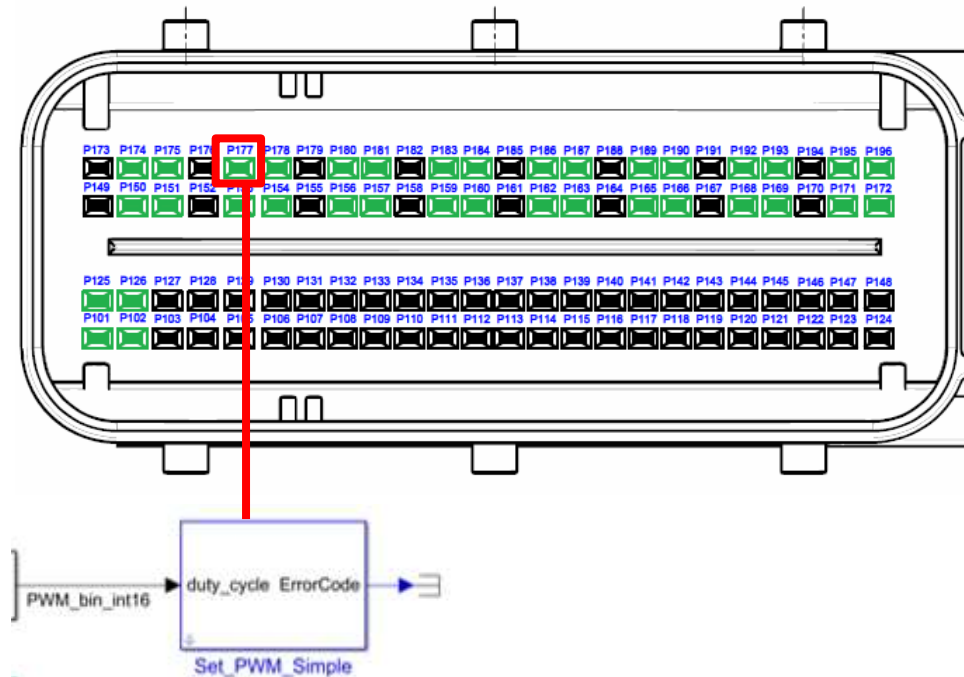
Choosing a PWM output port → Block

ADC_Absolute_10V

Choose the input port that fits to the connector pinning:

Pin 177 is connected → IO_PWM_01

For more infos see [1] 4.12 High-Side PWM Outputs



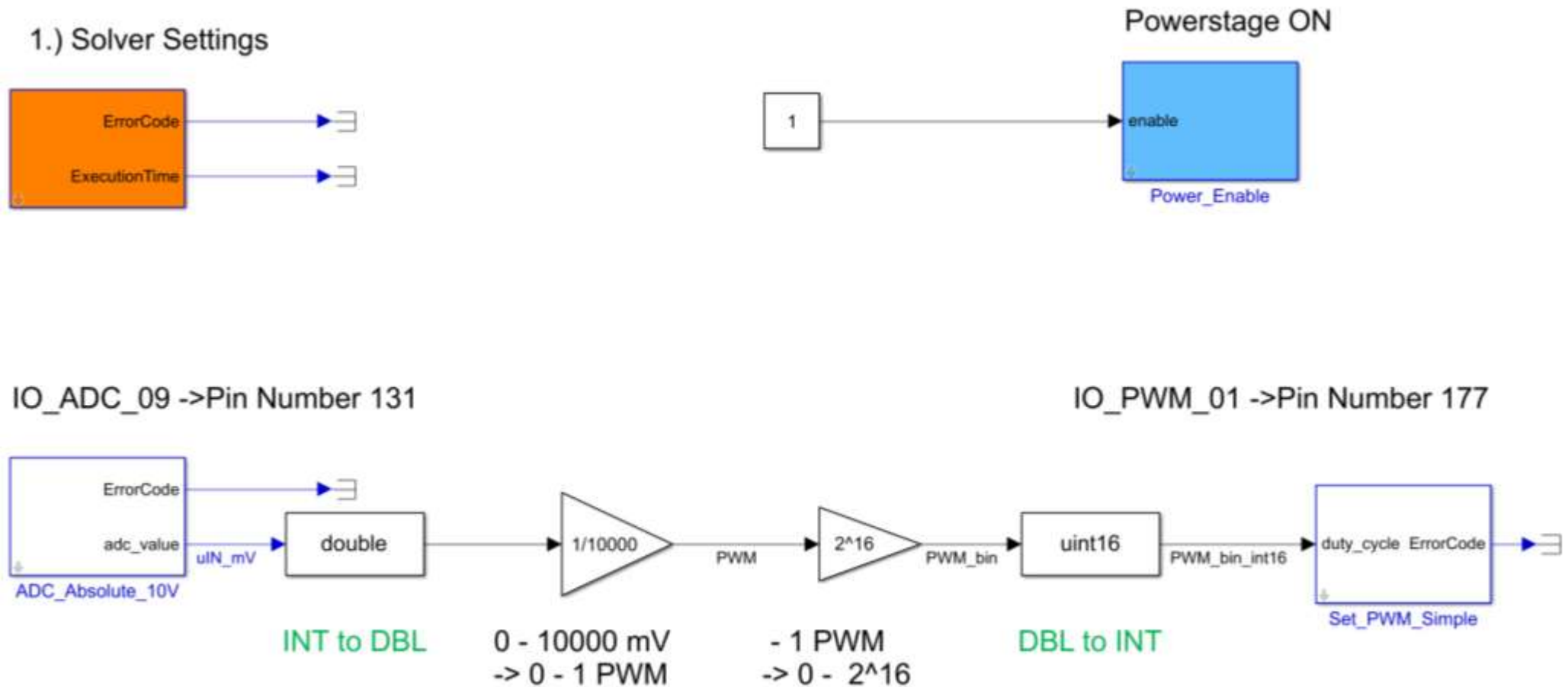
Pin No.	Function	SW-define	Ext./Second. Shut-off	Power stage
P153	High-Side PWM Output	IO_PWM_00	A	a
P177	High-Side PWM Output	IO_PWM_01	A	a
P156	High-Side PWM Output	IO_PWM_02	A	b
P180	High-Side PWM Output	IO_PWM_03	A	b
P159	High-Side PWM Output	IO_PWM_04	A	c
P183	High-Side PWM Output	IO_PWM_05	A	c
P186	High-Side PWM Output	IO_PWM_06	A	d



A simple Simulink example

Change PWM ratio as a function of a voltage signal

1.) Solver Settings



build → C-Code generation

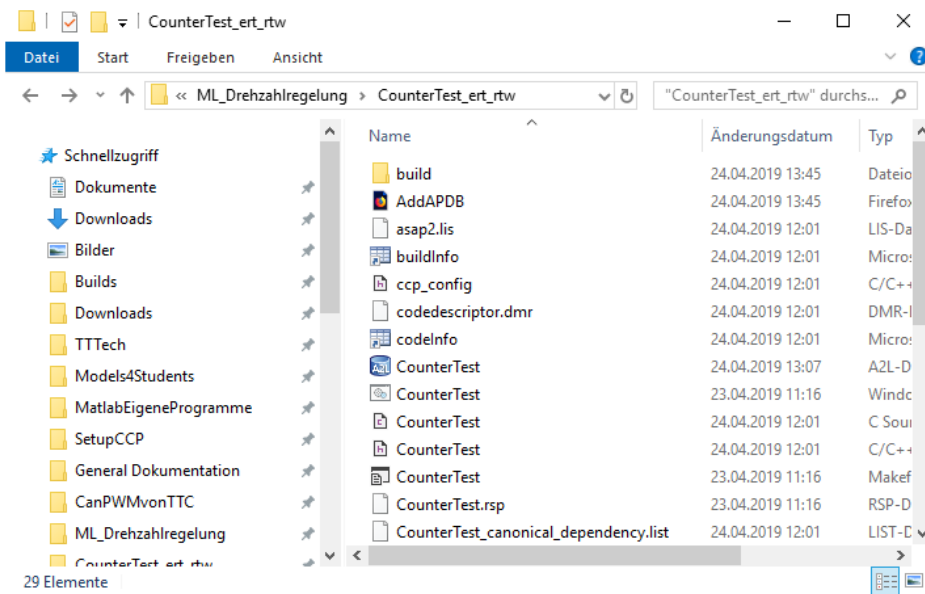


Embedded Coder

Embedded Coder:

<https://mathworks.com/products/embedded-coder.html>

The correct Simulink solver settings must be set before compiling the model.



```
PROJCT VERSION          : 1.1
* Simulink Coder version    : 8.14 (R2018a) 06-Feb-20
* C/C++ source code generated on : Wed Apr 24 12:01:23 201
*
* Target selection: ert.tlc
* Embedded hardware selection: Texas Instruments->TMS570 C
* Emulation hardware selection:
*   Differs from embedded hardware (MATLAB Host)
* Code generation objectives: Unspecified
* Validation result: Not run
*/
```

```
/* Includes
*****
#include "APDB.h"
#include "IO_Driver.h"
#include "IO_RTC.h"
#include "CounterTest.h"
#include "ccp.h"
#include "ccp_config.h"

/* Defines
*****
/* modify to adjust application version */
#define MAJOR_NUMBER          0U
#define MINOR_NUMBER          1U
...
*/
```

make → object files (*.hex)

TI Code Composer Studio:

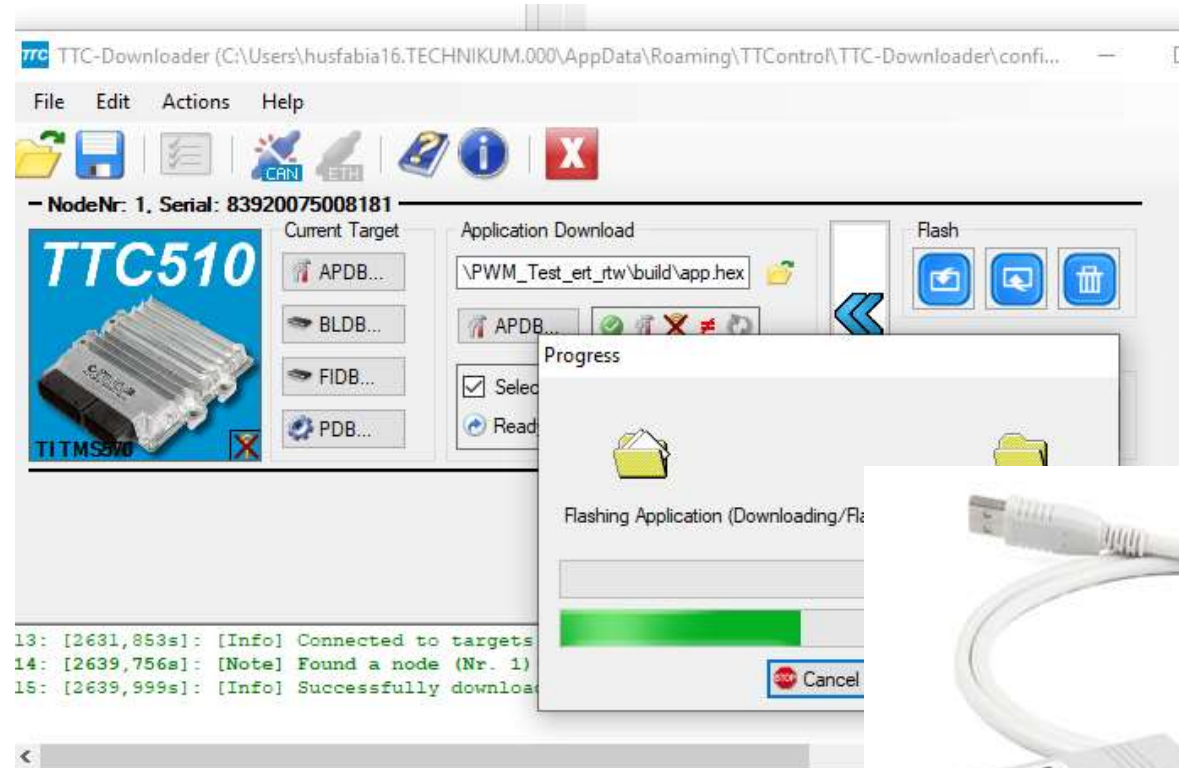
http://processors.wiki.ti.com/index.php/Download_CCS



TTC-Downloader

Flashing the ECU:

- Upload the *.hex-File from the PC to the ECU
- Physical Connection between PC and ECU
→ CAN
- Download Software → TTC-Downloader
(included in scope of delivery)
- For CAN-Connection → USB to CAN-Interface PCAN from Peak-System

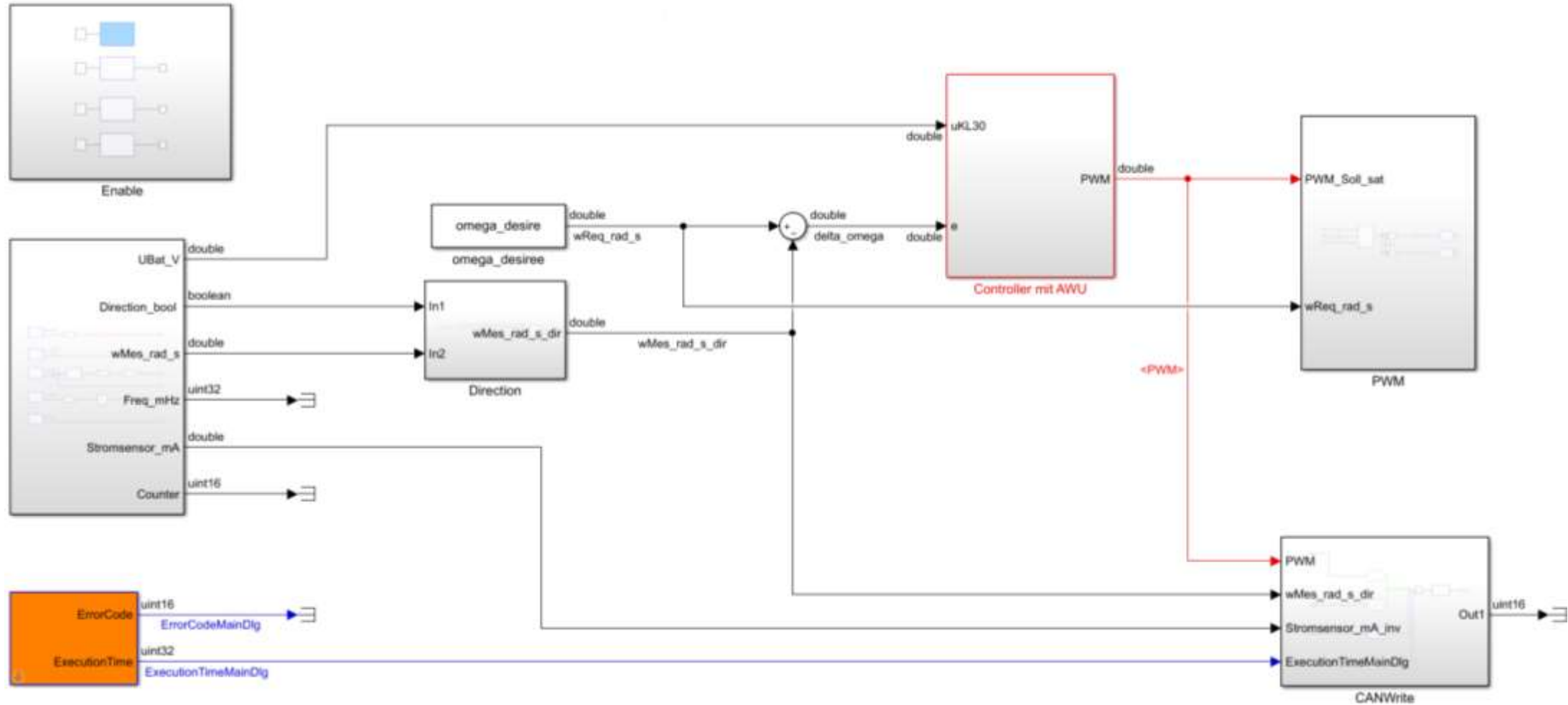


<https://www.peak-system.com>

Co-funded by the
Erasmus+ Programme
of the European Union



Speed Controller

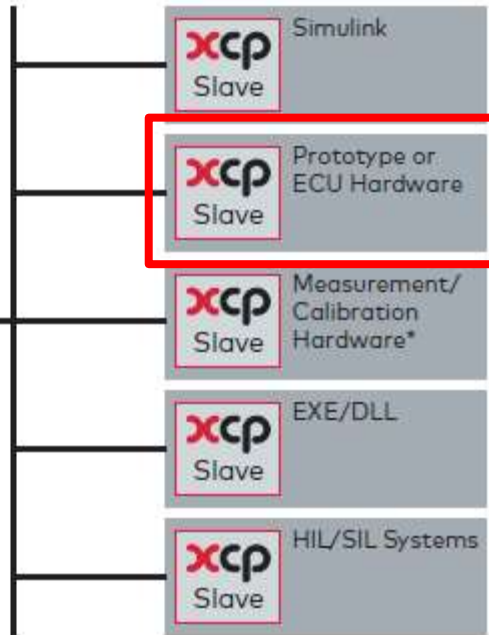


ECU-Calibration with CCP

Host PC with CANape



[2]



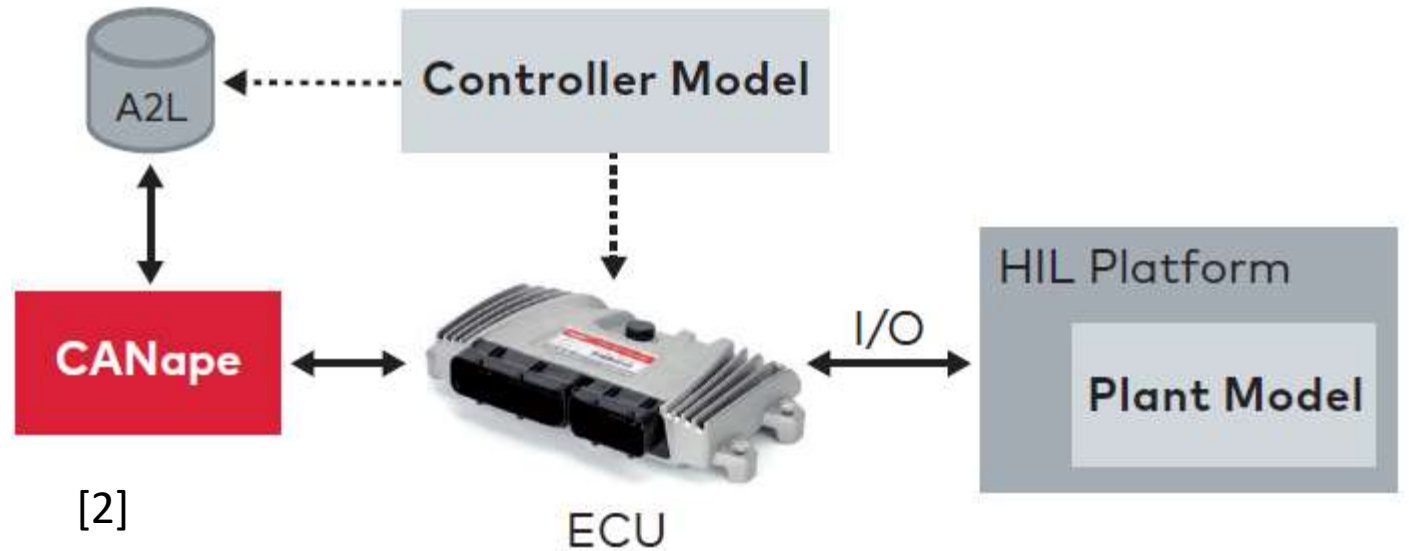
ECU HY-TTC 510



ECU-Calibration with CCP

Schematic figure of the calibration concept

- Calibration Software CANape from Vector Informatics - <https://www.vector.com/int/en/>
- Connection CANape <-> ECU with VN1630 CAN Interface
- A2L-File generated by Simulink Embedded Coder and TI ARM Compiler
- Controller Model from Simulink



[2]

ECU



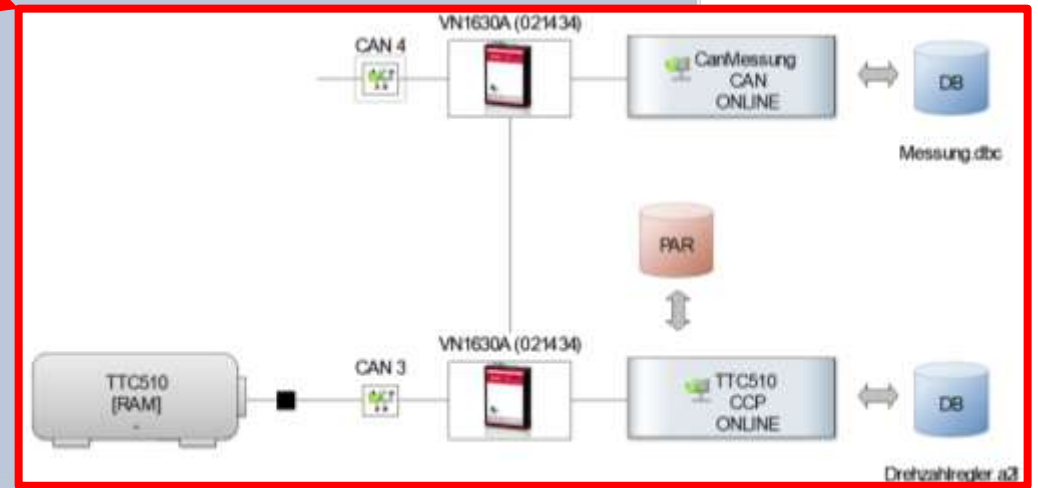
CANape Example

The screenshot displays the Vector CANape software interface. The top menu includes Start, Display, Devices, Calibration, Teams, Analysis, Tools, and Graphic. The toolbar contains various icons for connection, measurement setup, data acquisition, and display. The main workspace is divided into several panels:

- Symbol Explorer:** Shows a tree view of the project files.
- Device window:** A diagram showing the hardware setup, including a TTC510 device connected to a CAN interface and a VN1630A CAN module.
- Parameter:** A table listing parameters and their values:

Name	Value
omega_desire	1.200
sp1	0.0099999644
Tn_1	31.5
KWU	1000
- Numeric:** A list of numerical values for various signals:

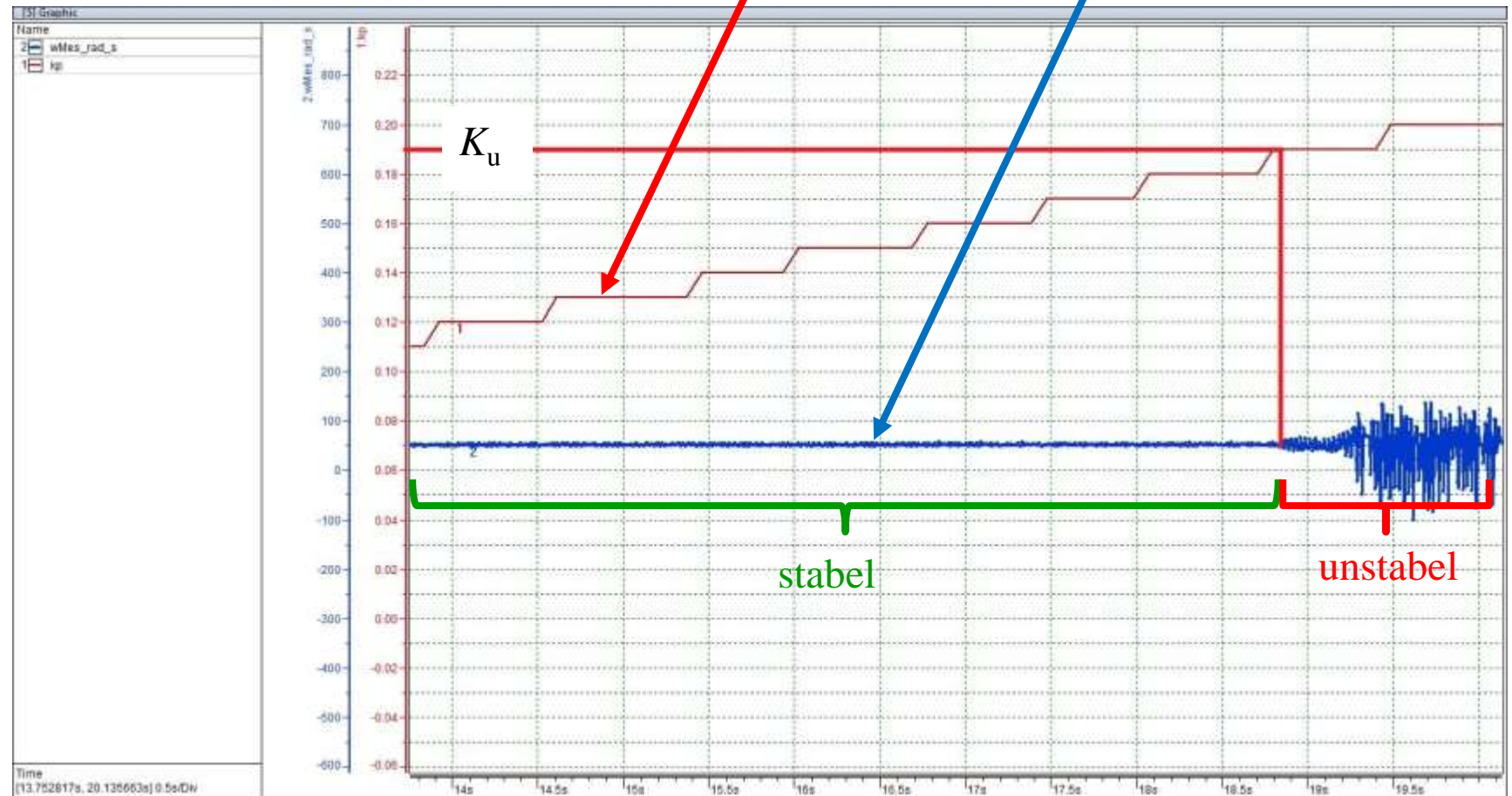
Signal	Value
UBat_V	11.8099975793
wReq_rad_s	200
wMes_rad_s	205.3325197321
delta_omega	10.9753952231
delta_t_Tn	-20.1979828244
delta_Tn_KWU	35.2879639303
delta_Add	6.055042268
delta_Memory	5.9711875981
ueCtrl0	5.4126527023
UBat_V	11.8099975793
PWM0	0.5962924973
PWM	0.4399449828
- Graphic:** A graph showing the signal 'wMes_rad_s' over time. The y-axis ranges from -1000 to 1000, and the x-axis shows time from 1m 20s to 1m 38s. The signal is a green line that fluctuates around a mean value of approximately 200.



The Ziegler-Nichols method

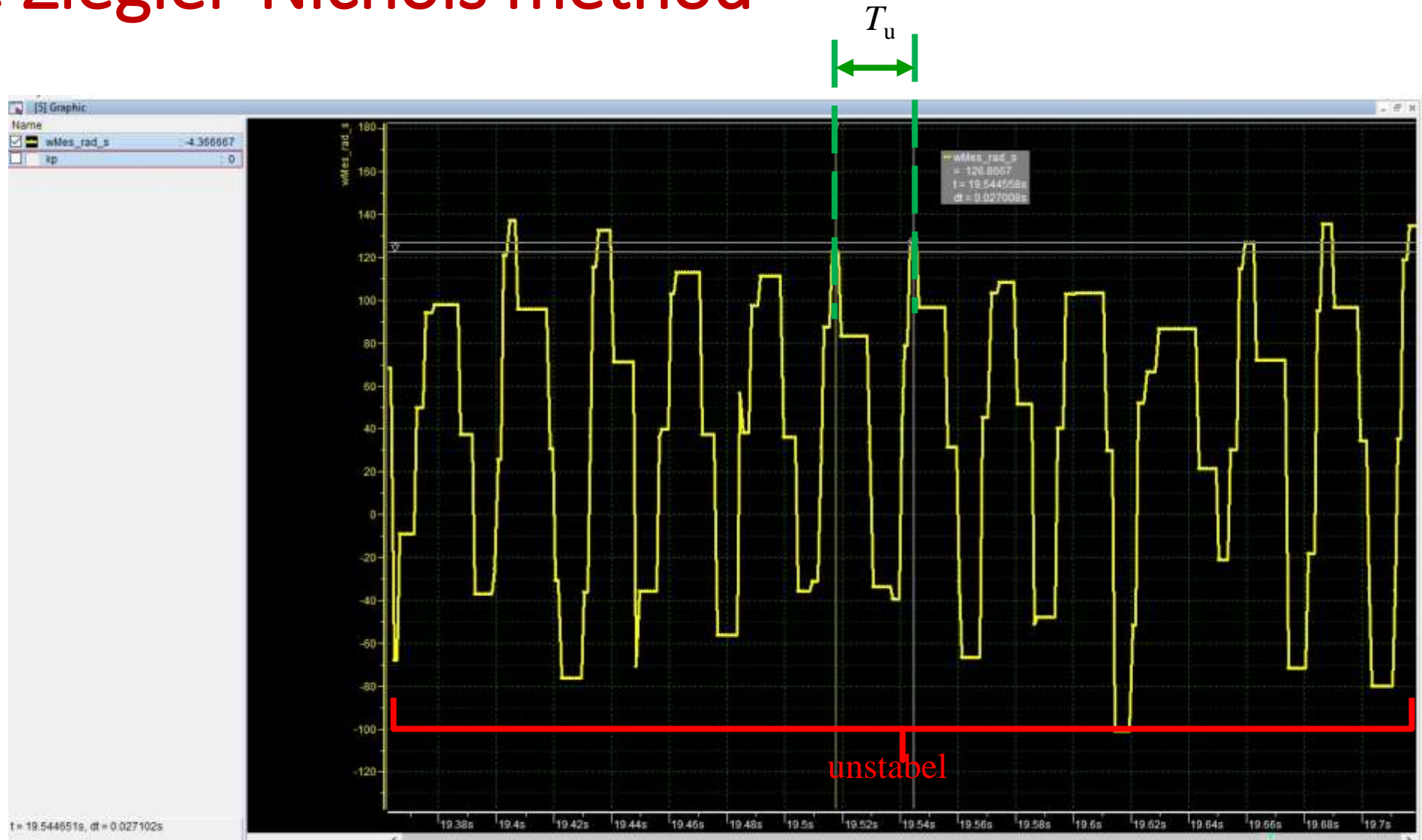
Setup for the speed controller (PI)

- Goal: Find optimal values for K_p and T_n
- Increasing K_p to the ultimate gain K_u .
- Adjustment via CCP out of CANape
- PI-controller \rightarrow
 $K_p = 0.45 K_u$.



The Ziegler-Nichols method

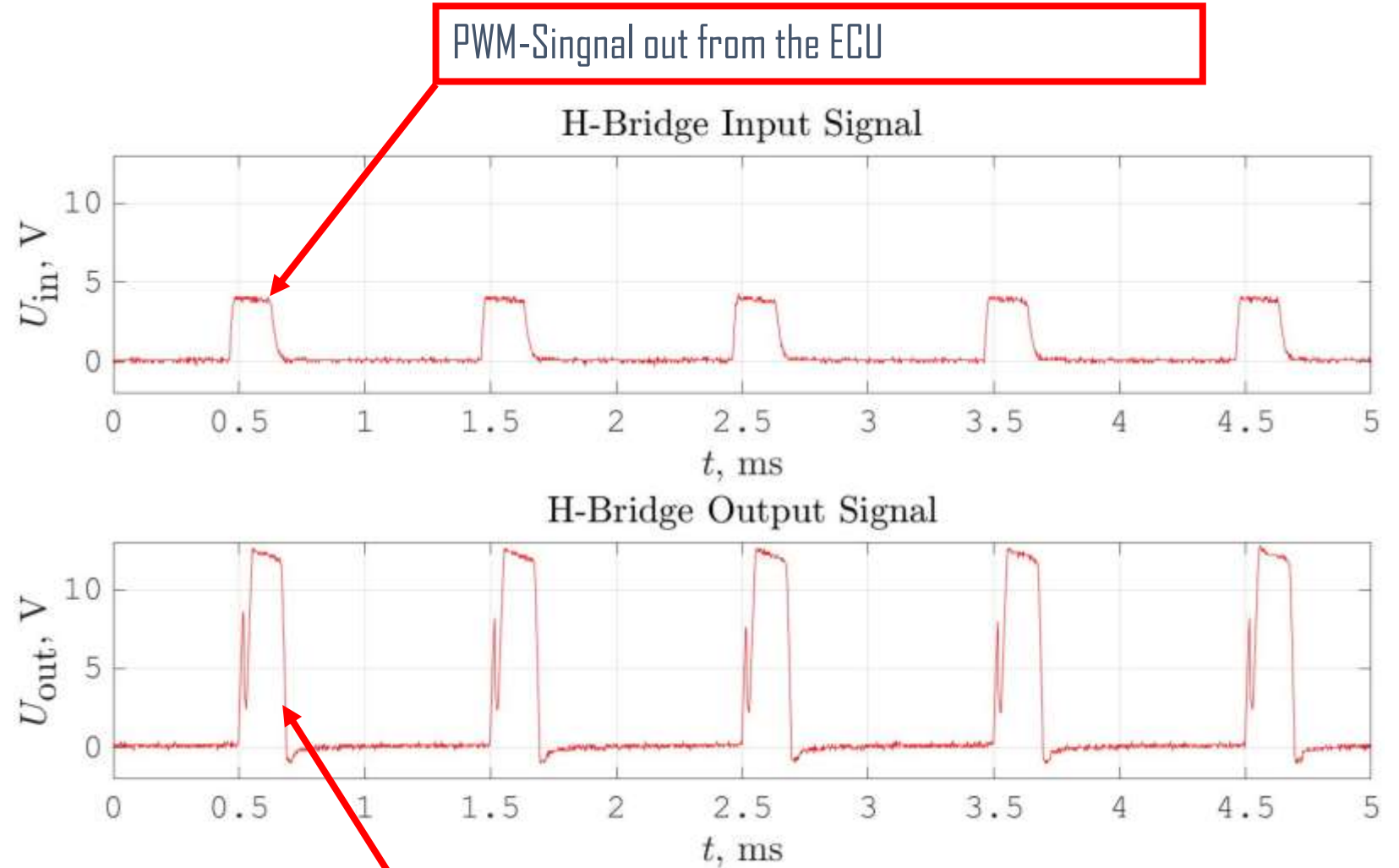
$$T_i = 0.8 T_u$$



ECU – adjust the DC motor speed

The TTC510-ECU has no H-Bridge included.

- External device must be used
- The ECU controls the H-Bridge with a PWM-Signal
- Maximum PWM-frequency from ECU is 1 kHz → Problem: structure-borne sound



Output Voltage from the H-Bridge to DC-Motor



References

- [1] TT Control GmbH: *HY-TTC 500 System Manual Programmable ECU for Sensor-Actuator Management Product Version 01.04*; 28 June 2017
- [2] Andreas Patzer | Rainer Zaiser: *XCP – The Standard Protocol for ECU Development*; Vector Informatik GmbH - Stuttgart, Germany ([Free download](#))





Setting up a Mechatronic System

4th Training in Rio de Janeiro, BRA

6th-9th of May 2019

Thomas Lechner



"The European Commission support for the production of this publication does not constitute an endorsement of the contents which reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein."

FOR EDUCATIONAL PURPOSE ONLY

Co-funded by the
Erasmus+ Programme
of the European Union

