



Why do we need calibration?

Vector Informatik
Andreas Patzer

Goal of this document

- Explanation of the basics of Measurement & Calibration (MC)

Introduction

Imagine:

- ❑ ... you are driving with your new car trough a forest
 - ❑ a wild pig is crossing the road directly in front of the car
 - ❑ you slam on the brakes
 - ❑ your car is still controllable, cause no tire is blocking
-
- ❑ Why is no tire blocking?

Introduction

- ❑ Cause your new car has an Antilock Brake System (ABS)
- ❑ Some sensors are measuring the behavior of the car and the tires. In case of blocked tires, the Electronic Control Unit (ECU) take over the control of the brake system to help you to control the car.

- ❑ How can a ECU do that?

Introduction

- ❑ The ECU is a little computer running a brake control application.
- ❑ It analyzes the sensor values and calculates activities and realize this activities with the help of actuators.
- ❑ Functions, written in software are used to calculate the activities.

- ❑ E.g.: $y = 5 * x + 2$ (x is the sensor value)

- ❑ With the compiling and linking of the software, a binary-file is generated. This binary-file contains the complete program and the data. With a copy process ("flashing") from the PC to the flash memory of the ECU, the software is running in the ECU.

Introduction

- ❑ But why is the software developer using '5' for the coefficient and '2' for the offset in the function $y=5*x+2$? If the coefficient and the offset are depending on the car physics, how can the software developer be sure about the value?
- ❑ If there are uncertainties about the right values or the values may differ in dependencies of the physic, the software developer would write the function in that way:
- ❑ $y = a * x + b$
- ❑ But the ECU needs explicit numerical values to calculate the result. Where do the values of a and b come from?

Introduction

- ❑ Normally we have separate phases in the software development.
- ❑ 1. Code development:
 - ❑ A software developer uses a programming language to realize the functions and the complete application.
- ❑ 2. Application or calibration phase:
 - ❑ An application engineer sets the parameters to the right values.
 - ❑ He is NOT changing the functions itself, he is changing the parameter values.

- ❑ How can an application engineer do that?

Introduction

Online vs. offline calibration

- ❑ **Offline calibration** = calibration of ECU binary-file and flashing into ECU memory
- ❑ As mentioned before, the program code and the data must be stored in the ECU memory.
- ❑ By changing the values of the parameters in the binary file and flashing this new file into the ECU memory, the behavior of the ECU can be changed.
- ❑ Evaluate the effects by
 - ❑ Analyzing the bus messages
 - ❑ Analyzing analog/digital sensor signals

Introduction

Online vs. offline calibration

- ❑ **Online calibration** = modify the parameters “on the fly” directly in the ECU memory
- ❑ Access to the ECU memory
- ❑ Modify the parameter „**on the fly**”
- ❑ Watch the „response” (reaction) of the ECU
- ❑ Evaluate the effects by
 - ❑ Analysing the bus messages
 - ❑ Analysing analog/digital sensor signals
 - ❑ Analyse **ECU-internal variables** (see measurement data acquisition)
 - ❑ Input and output value of the control algorithm
 - ❑ Sensor input values

Introduction

Online vs. offline calibration

Online

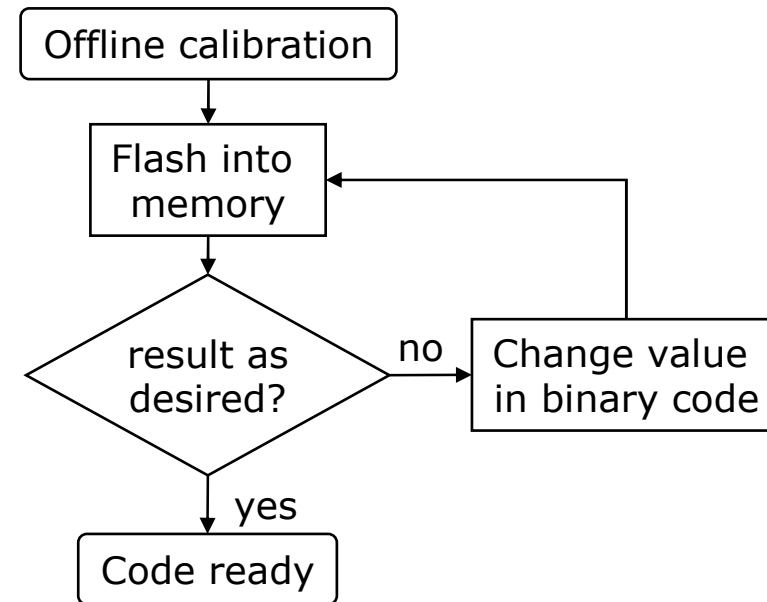
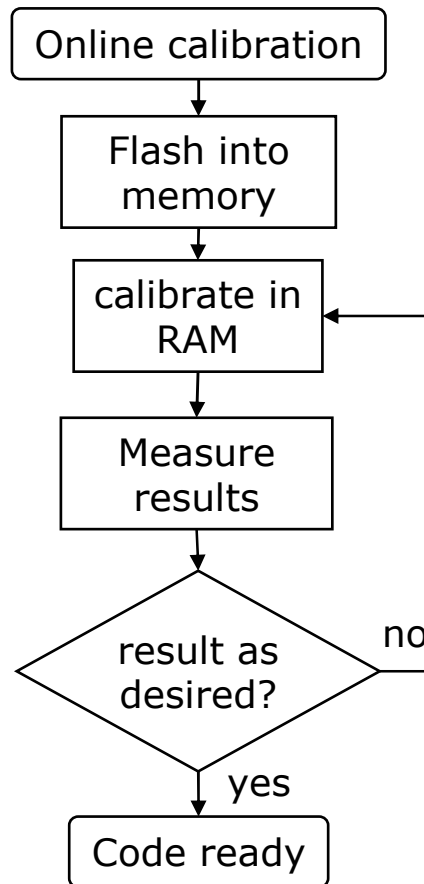
- Protocol for measurement and calibration in the ECU necessary
- + Standard conform solution for flashing, calibration and measurement
- + Without time delay the relation between calibration and result becomes visible
- + Working with symbolic names "Speed" instead of memory addresses "3F1A"

Offline

- + Only flash protocol in the ECU necessary
- The correlation between calibration and result is always disconnected with the flashing process
- User must know the memory addresses very well or - like in CANape Graph – jump to the right address by selecting the parameter name from the memory description-file (A2L)

Introduction

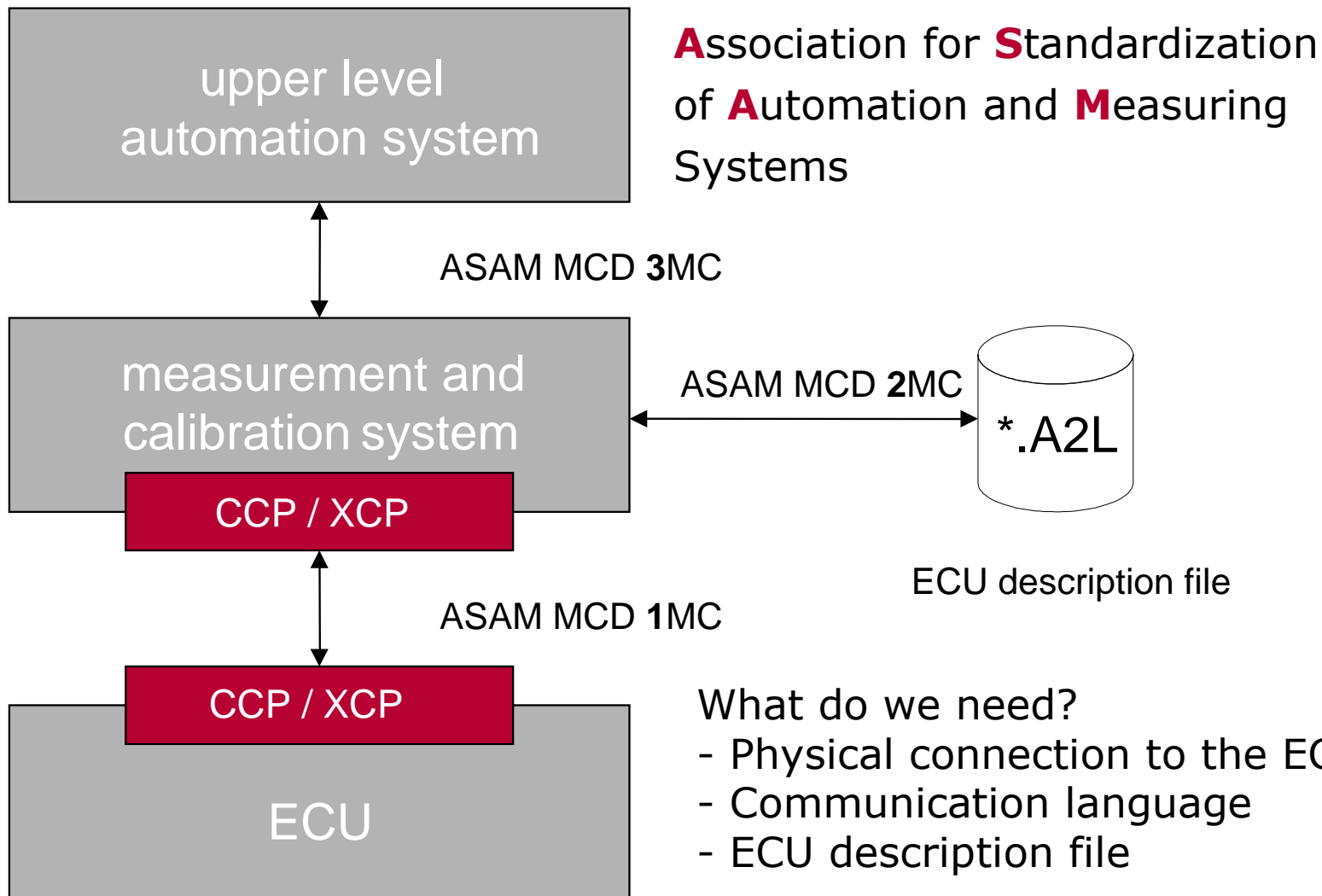
Online vs. offline calibration



- ❑ CANape can be used for online **and** offline calibration
- ❑ What is necessary to realize online calibration?

Introduction

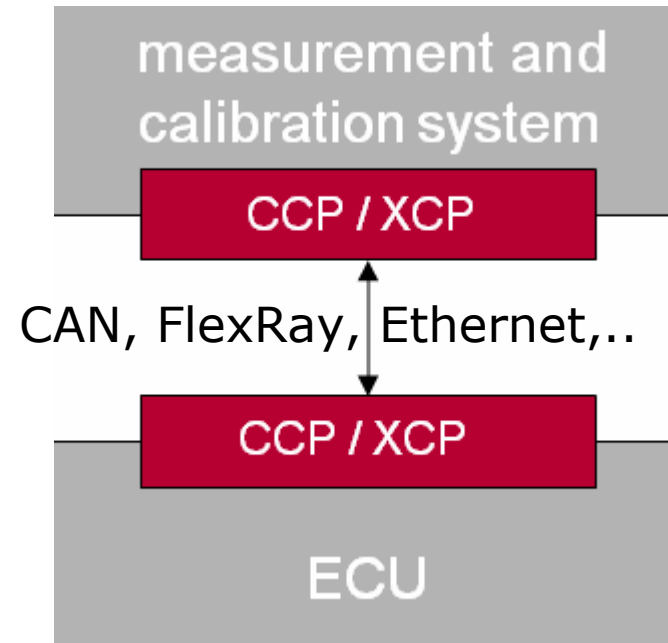
Online calibration concept



Introduction

Physical connection to the ECU

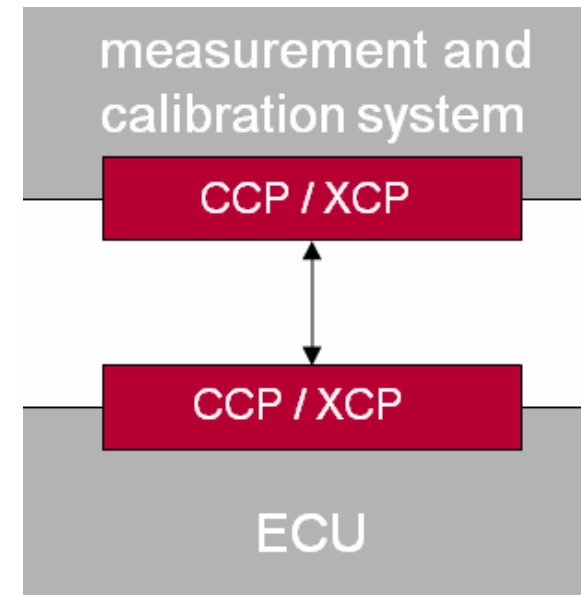
- ❑ Different physical connection can be used:
- ❑ CAN
- ❑ FlexRay
- ❑ LIN
- ❑ SPI / SCI
- ❑ USB
- ❑ DPRAM
- ❑ NEXUS
- ❑ JTAG
- ❑ ..



Introduction

Communication language

- ❑ CAN Calibration Protocol (CCP) or
- ❑ X Calibration Protocol (XCP *)
- ❑ XCP is the successor of CCP
- ❑ In the ECU system resources are necessary for the CCP or XCP driver
 - ❑ Effort: CCP/XCP protocol implementation
 - ❑ ROM: 1-3 kByte
 - ❑ RAM: 100 Byte (depends on the max. number of measurement signals)
 - ❑ CPU load (e.g. Measurement data collection, CAN-communication)
 - ❑ Bus load (CAN)
 - ❑ Calibration concept needed



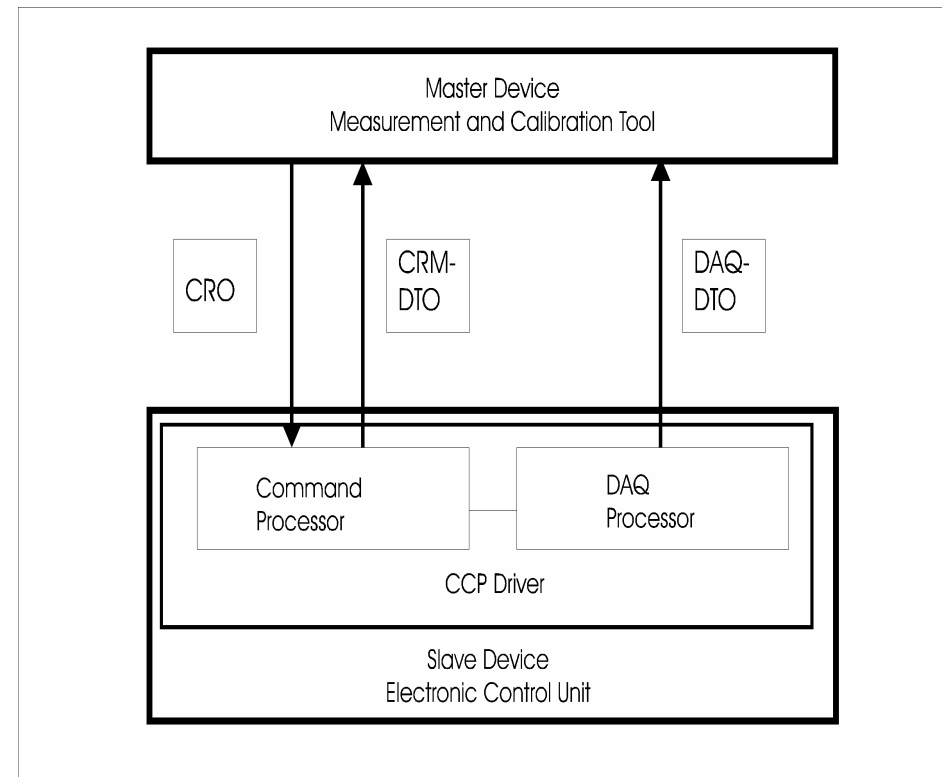
* X means "any" or "universal"

Introduction

Communication language

❑ Two CAN identifiers are needed for the CCP communication between CANape and the ECU:

- ❑ CRO-ID
(Command Receive Object)
- ❑ CRM / DTO-ID
(Command Return Message/
Data Transmission Object)



- ❑ The Command Processor is responsible for the execution of the transmitted CCP commands, e.g. read/write on a RAM location.
- ❑ The DAQ Processor is responsible for synchronous data transmission, which of course first has to be initialized by CANape.

Introduction

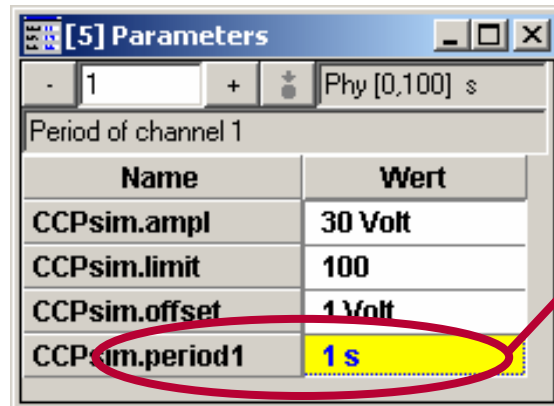
A2L description file

- ❑ User want's to use symbolic names, e.g. "period1".
- ❑ ECU does not know anything about names. The ECU can only handle the addresses of memory cells, e.g. "410D8" were the content of the parameter period1 is stored.
- ❑ Among other things, the mapping between the names and their addresses is documented in the A2L-File (ASAM 2nd interface Language)

Name	Address	Data type	Comment	Conversion	Unit
Test_Paramet...			Test Parameters, BYTE		
ampl	410DC	FLOAT			Volt
limit	410E0	FLOAT	Limit for channel1		
offset	410E4	FLOAT	Offset for channel1		Volt
period1	410D8	FLOAT	Period of channel 1		s

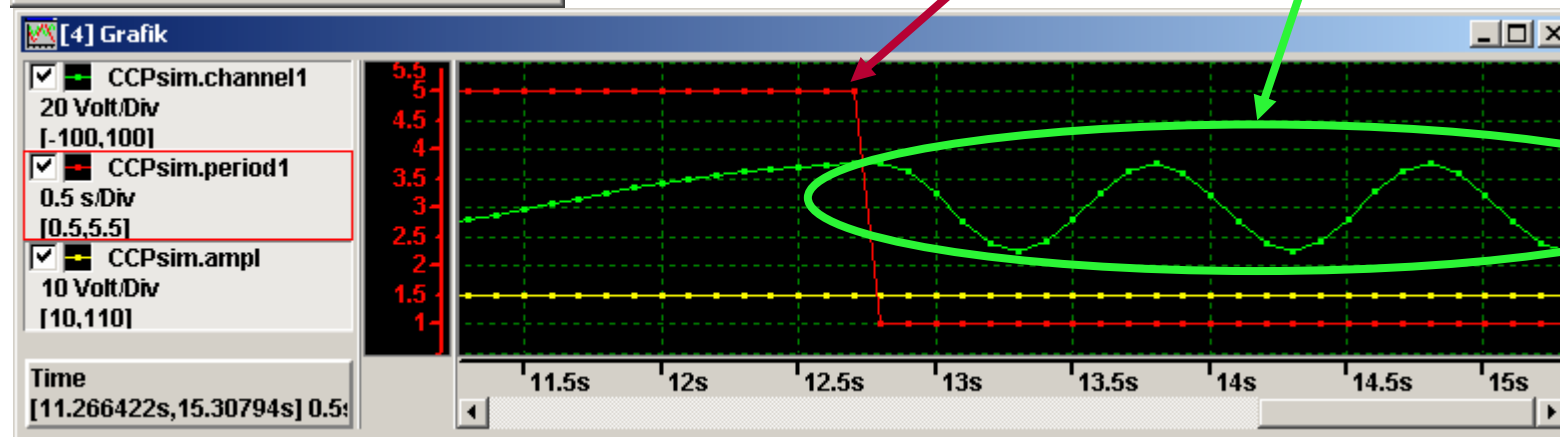
Introduction

Online calibration



Change parameter "period1" (ECU=CCPsim) from 5 seconds to 1 second

See the new behavior directly in the measurement window.

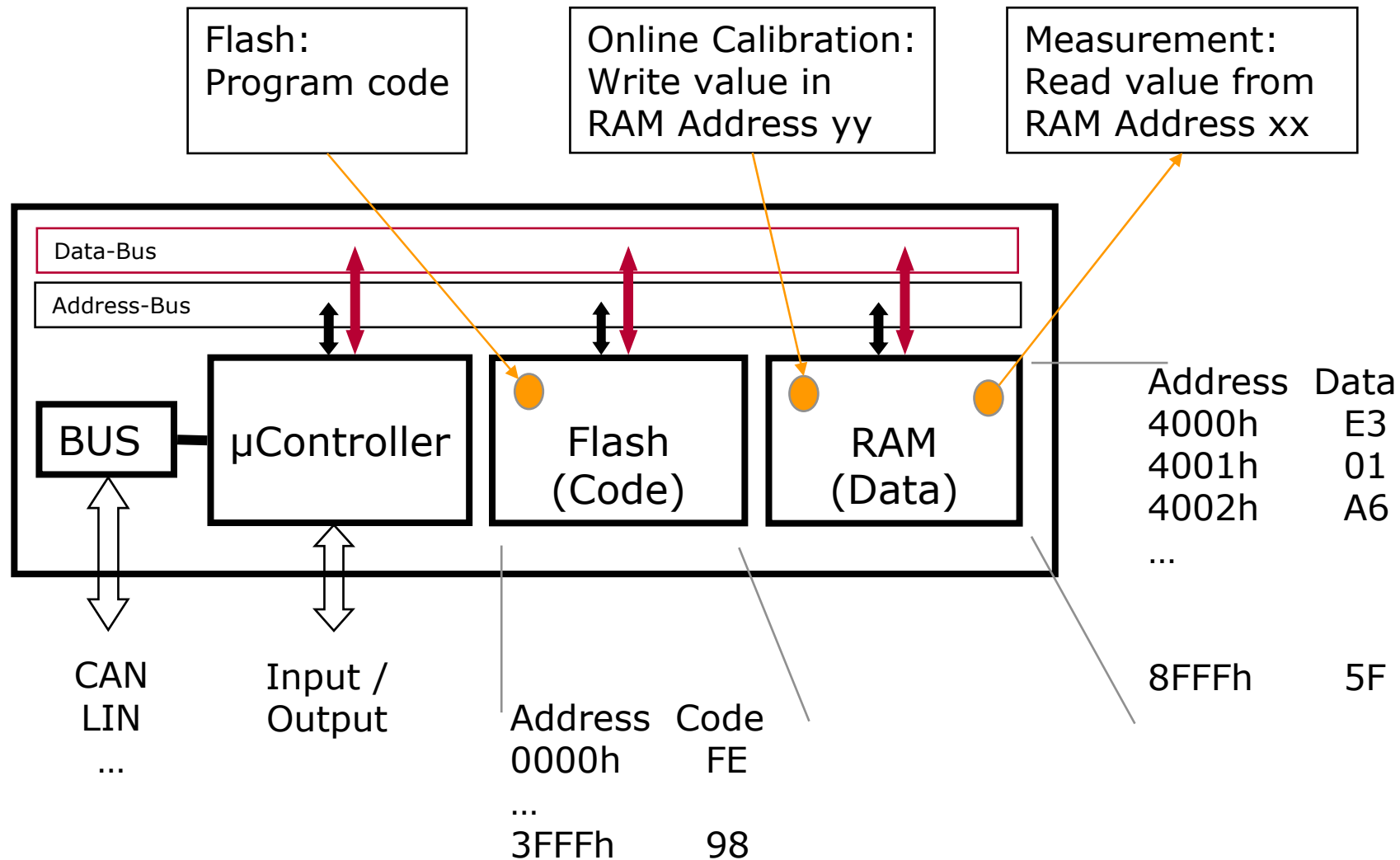


But what is necessary to change the content of the parameter "period1" to another value?

- ❑ To understand the basic concept of changing the content of a parameter in an ECU, it's necessary to take a look at the memory concept.
- ❑ Cause: Only the content in the RAM is changeable.
- ❑ Only parameters that are available in the RAM can be calibrated!

Introduction

Online calibration



- ❑ Measurement: Read value from RAM Address xxxx
- ❑ Calibration: Write value in RAM Address yyyy
- ❑ Flash: Copy code from the PC into flash memory

- ❑ In the end, measurement and calibration is nothing else than read and write access to RAM. You need
 - ❑ Physical access (e.g. CAN)
 - ❑ A language to tell the ECU what you want to do (CCP, XCP)
 - ❑ The knowledge witch RAM cell information you want to change (A2L file)

Introduction

Online calibration

- ❑ Now we are ready!

- ❑ Or not?

- ❑ Mmmmh...

- ❑ RAM memory stands for Random Access Memory. After power off, the content in the RAM is lost. Cause of this, we have two more questions to discuss:
 - ❑ After power on, there is no content in the RAM. But the ECU needs real values for the parameters. What is the way to set the parameters in the RAM?
 - ❑ At the end of the day, the application engineer changed a lot of parameters. In the RAM! How can he store the changes?

Introduction

Calibration concept

- ❑ At the boot process in the ECU, the content of the RAM has to be initialized by the application.
- ❑ E.g., 1000 Byte of data has to be copied from the flash into the RAM. After this initializing phase, the application can start and run with the content of the RAM.
- ❑ The way to initialize the RAM and to change its content, is called “calibration concept”. There are a lot of different calibration concepts on the market.
- ❑ If your customer is talking to you about calibration concepts, please talk to your responsible Vector contact person. The contact person will route you to the right PMC colleagues (PMC=Productline Measurement and Calibration).

Introduction

Save your calibration work

- ❑ The content in the RAM is changed by the application engineer. With CANape he can store the changed parameters into parameter files on the PC.
- ❑ CANape allows a lot of different formats and with the integrated Calibration Data Manager (CDM) he can manage (e.g. copy, control, merge, ..) the content of the files in a very comfortable way.
- ❑ With our tool "eASEE.CDM" we have a server based solution to manage parameter files for complete companies.

CANape USPs

- ❑ Vector tool and hardware compatibility
- ❑ Fully integrated Diagnostic Feature Set
- ❑ Flexible script driven flash programming
- ❑ Powerful programming language
- ❑ Integrated ASAP editor and updater
- ❑ Integrated CDM Studio and eASee support
- ❑ LIN and FlexRay support
- ❑ Special features:
 - ❑ Video and Audio
 - ❑ GPS
 - ❑ Panels
 - ❑ Visualizing of Simulink Models



Thank you for your attention.

For detailed information about Vector
and our products please have a look at:
www.vector-informatik.com

Author:

Andreas.Patzer@vector-informatik.de

Vector Informatik GmbH

Ingersheimer Str. 24

70499 Stuttgart