

Real-Time Development Interface Embedded in a Compact Motion Controller

Josef Reill, Cristina Serrano Gonzalez, Volker Senft and Martin Pfau

DLR (German Aerospace Center) – Robotics and Mechatronics Center
 Institute of Robotics and Mechatronics
 Münchener Str. 20, 82234 Weßling, Germany

E-mail: Josef.Reill@dlr.de

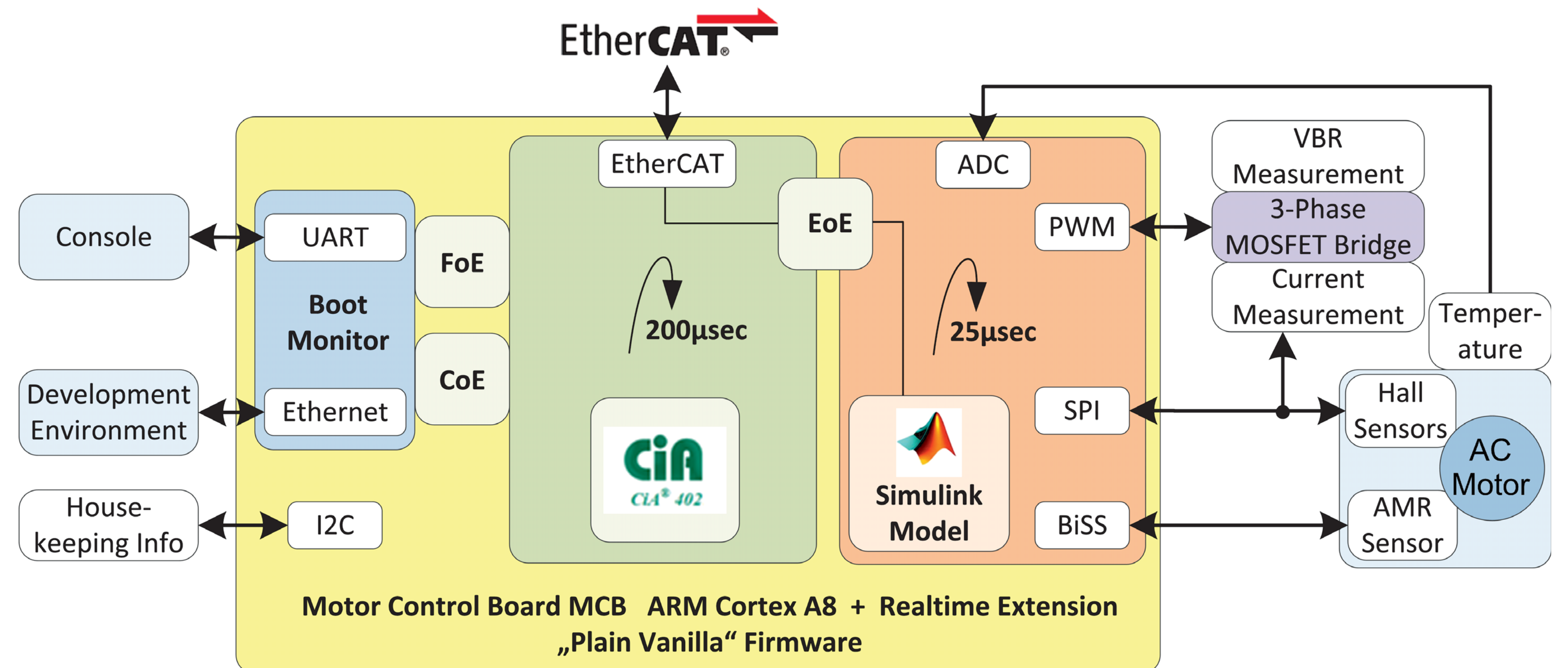
Web: <http://rmc.dlr.de/rm>



Motion Controller with EtherCAT and DC-Power connection only and fully featured Real-Time interface to the motor inverter

Principle

- ARM Cortex A8 with Real-Time extension used
- Firmware handles hardware IOs and EtherCAT communication
- Simulink model interfaces hardware via s-functions only
- Simulink model is focused on the development and investigation of the control algorithm
- Inputs are the measured signals (motor current, dc-link voltage and motor position)
- Outputs are the three PWM signals to interface the 3-phase voltage source inverter



How it works:

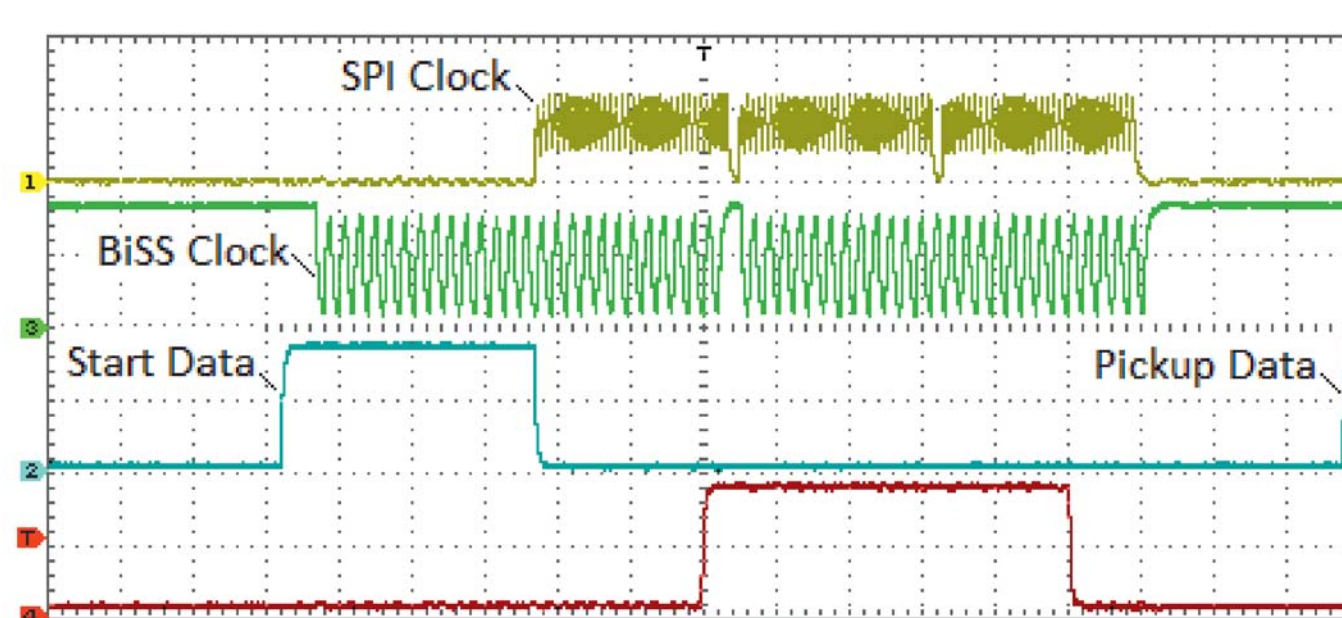
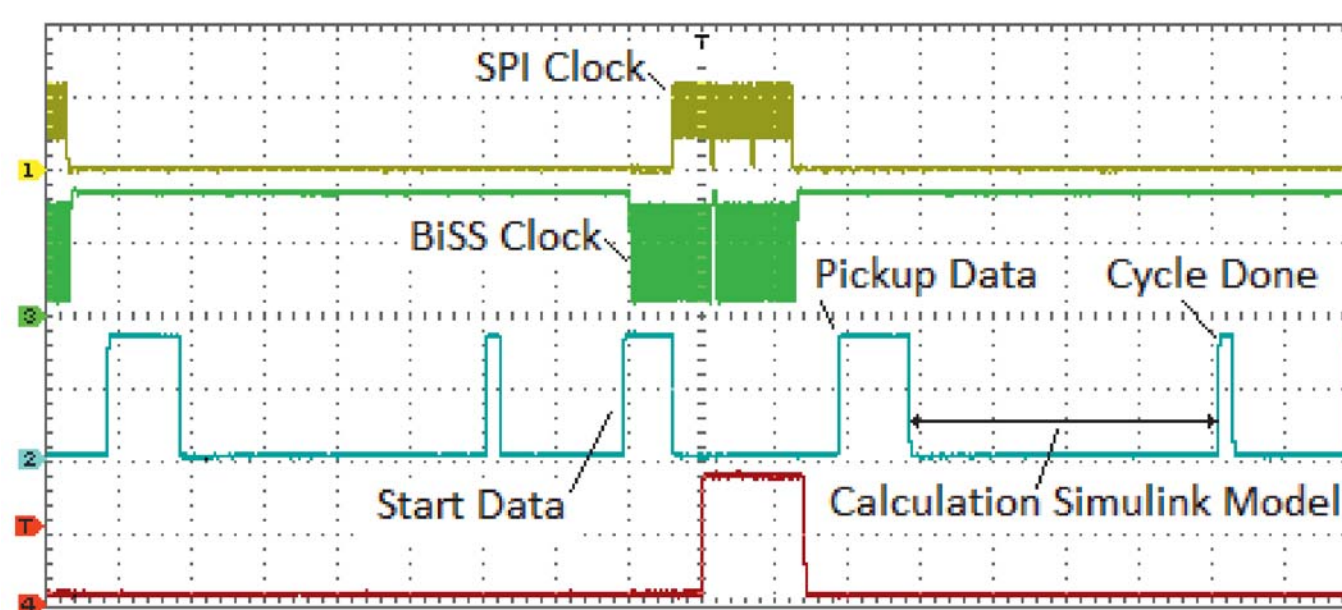
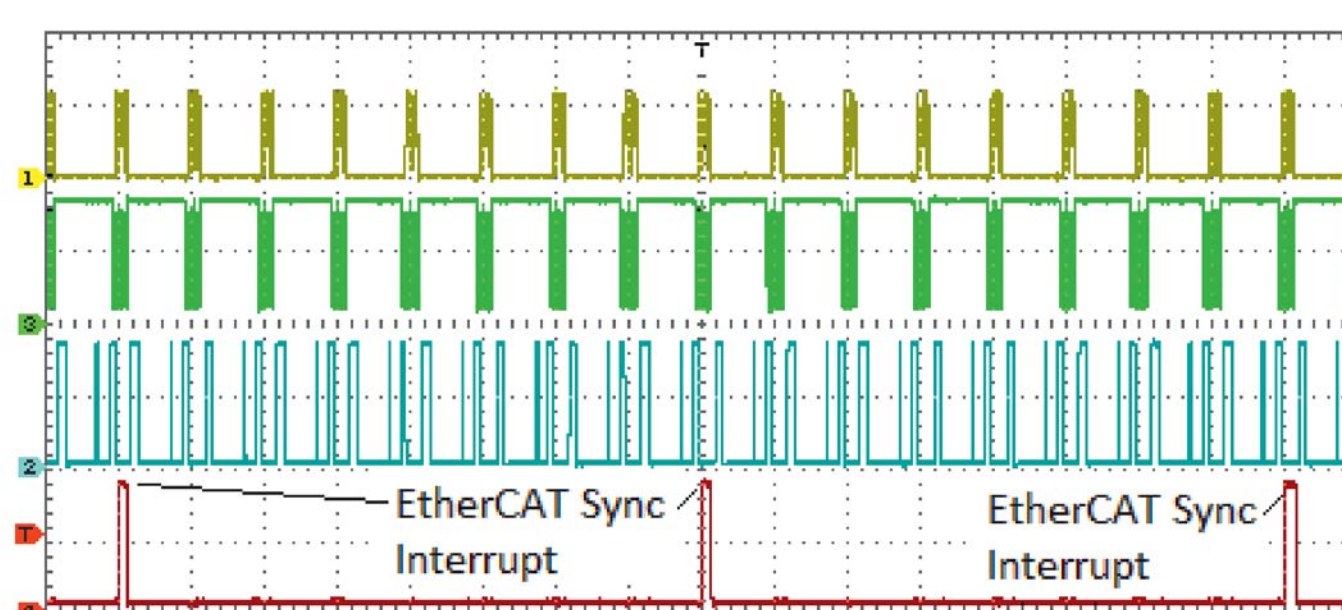
- ARM Cortex A8 is running a firmware that
 1. supports EtherCAT as an industrial Real-Time bus
 2. provides an interface to the hardware registers (PWM, SPI communication...)
 3. computes the controller algorithm that is developed with Matlab/Simulink
- Generate C-Code from Simulink Model with Simulink Coder
- Controller algorithm code is embedded in the ARM firmware
- Connect Host-PC to the target via external mode
- All Simulink signals are visible and changeable

Advantages:

- Ease of use and comfort of developing control algorithms that is known from expensive rapid control development systems
- EtherCAT communication is used solely to interface the hardware
- External mode features make use of Ethernet over EtherCAT technology
- File Access over EtherCAT is used to update the control algorithm from remote
- The development hardware is already the final hardware
- Investigation of the running control algorithm from remote possible whenever needed

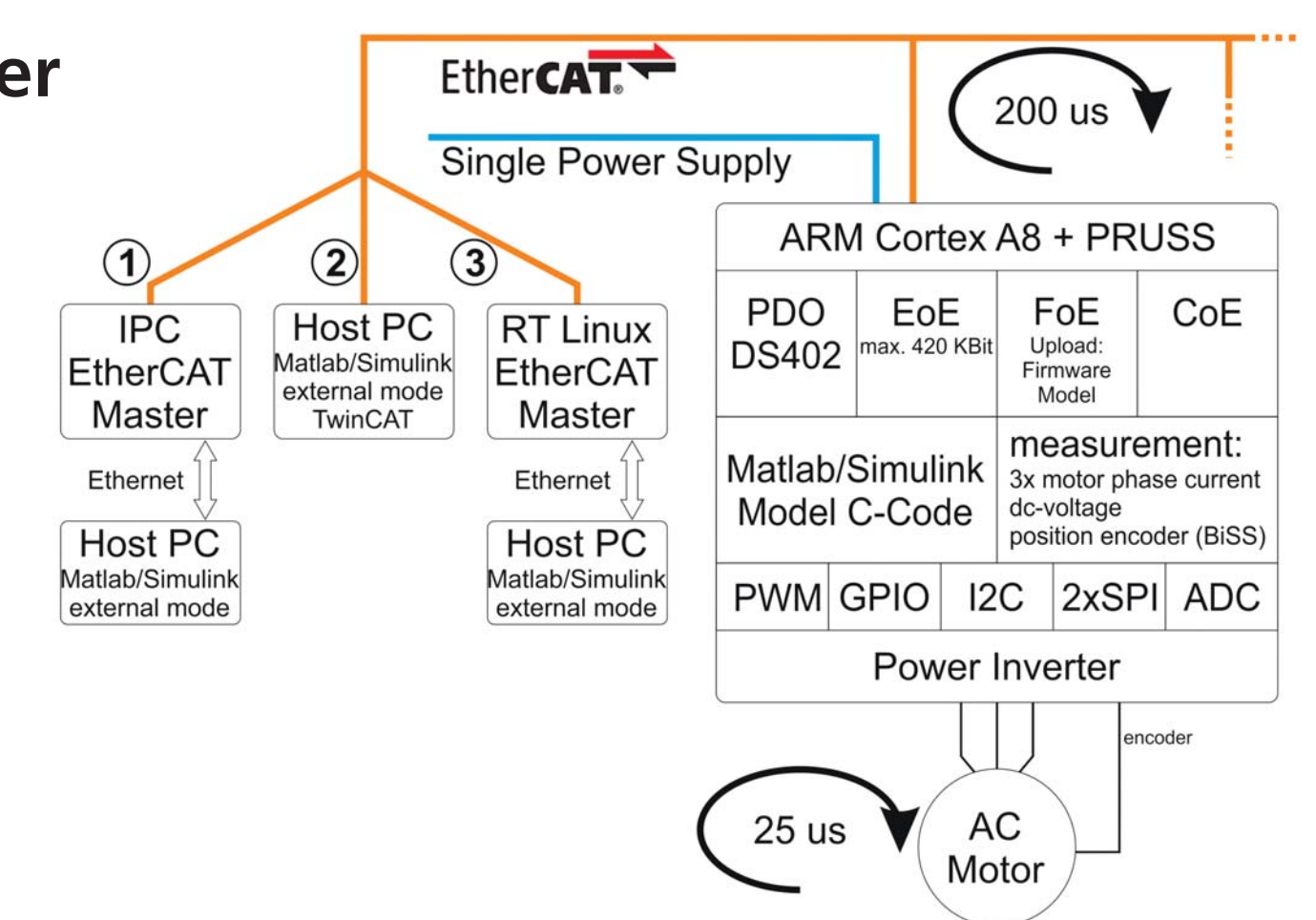
Timing

- The control algorithm cycle is triggered by EtherCAT
- EtherCAT cycle and controller loop cycle run synchronously
- Every control loop cycle starts with sensor reading (overall 25 µs is split in 10 µs sensor reading and 10 µs to calculate the Simulink model)
- Motor phase current and dc-link voltage are read by SPI channel 1
- Motor position is read by BiSS protocol that is implemented on SPI channel 2
- Right after data is picked up the Simulink model calculation is started



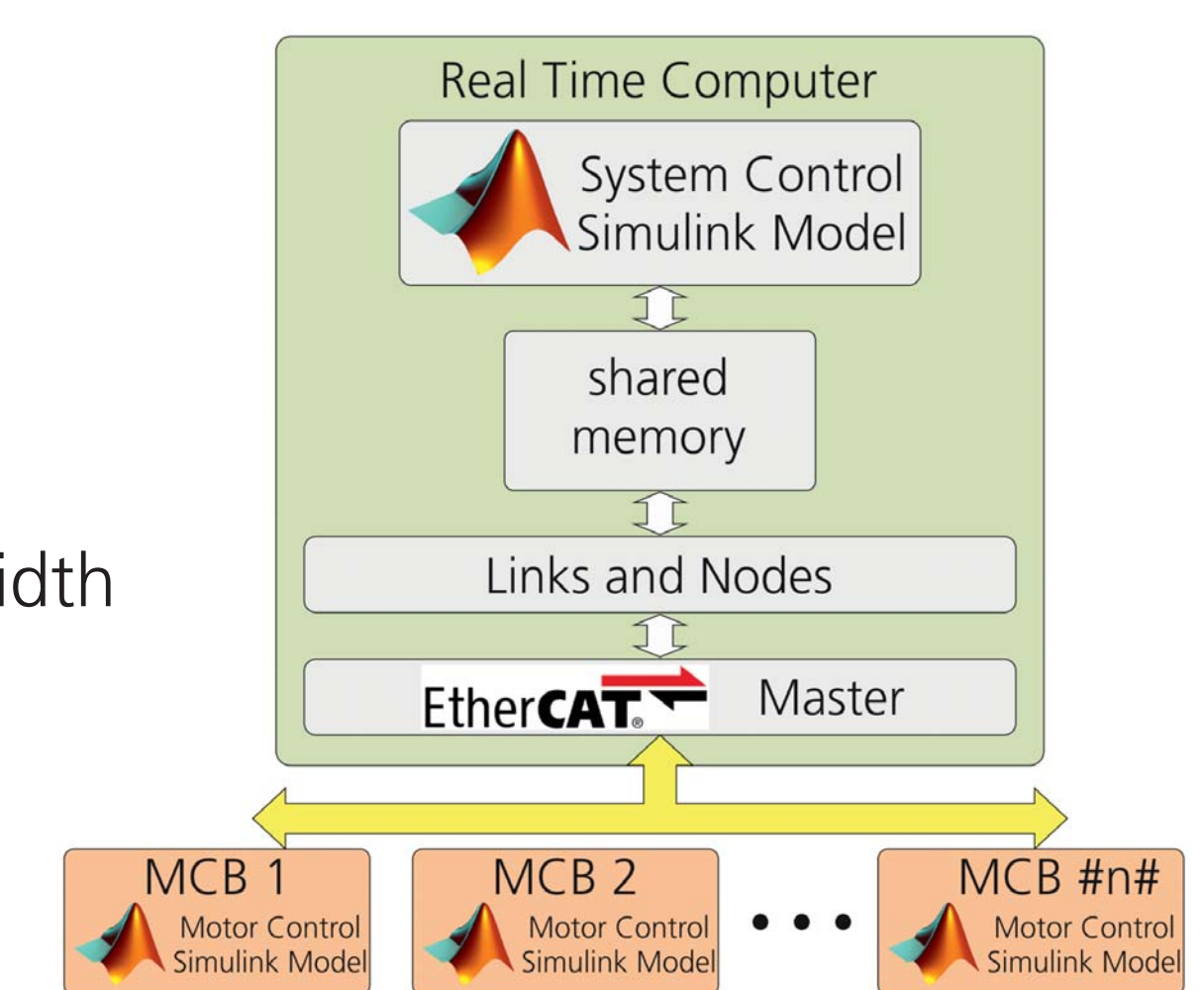
Different ways to access the motion controller

1. IPC is EtherCAT Master
Host PC with Simulink connected via Ethernet
2. Windows PC with Simulink and TwinCAT software is EtherCAT master and Host-PC
3. RT-Linux with EtherCAT master
Host-PC with Simulink connected via Ethernet



Expansion to Multi-Axis Systems - Increasing system complexity

- Several Motion Controller Boards connected to one EtherCAT bus
- Real-Time Computer for overall system control
- Software development for high level controller is done by use of Matlab/Simulink
- Software „Links and Nodes“ is used to get access to controller values in Real-Time at EtherCAT bandwidth
- Controller development on the motion controller board MCB is also done with Matlab/Simulink
- All kind of control levels use the same approach and toolchain!



Workflow

1. Controller algorithm development in block diagram form
2. Implementation in Matlab/Simulink environment
3. Generate C-Code with embedded coder
4. Bind and compile with firmware C-Code
5. Upload to motion controller via File-over-EtherCAT
6. Use Simulink external mode via Ethernet-over-EtherCAT
7. Analyze all signals that were put on Simulink scopes in the model
8. Values of Simulink blocks may be changed online, changes in the model structure require new compilation of the code

