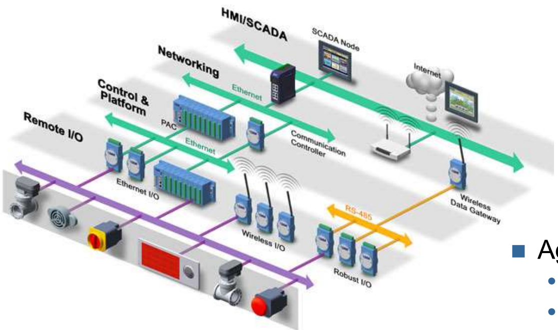
Distributed Embedded Systems

Zürcher Hochschule für Angewandte Wissenschaften
School of Engineering
InES Institute of Embedded Systems

Distributed (Embedded) Systems

Zürcher Hochschule für Angewandte Wissenschaften
School of Engineering
InES Institute of Embedded Systems

Session 1:
- **Agenda**
- Centralised Control Systems
- Remote I/O
- Fieldbus
- Real Time Ethernet
- Conclusion



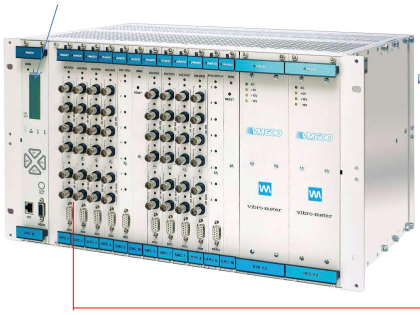
http://www2.advantech.com/eAutomation/Remote-IO/images/ARCH2.jpg

- Agenda
  - Centralised control systems
  - Distributed control systems V1.0
    - Early systems (80's)
  - Distributed control systems V2.0
    - Fieldbus technologies (90's)
  - Distributed control systems V3.0
    - Real-time Ethernet ('05++)
  - Distributed control systems V4.0
    - IoT systems

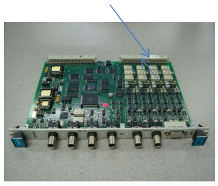
Centralised Control Systems

Session 1:
- Agenda
- **Centralised Control Systems**
- Remote I/O
- Fieldbus
- Real Time Ethernet
- Conclusion

CPU >> IPC



VME Backplane Bus System



LCR between Analog Signal and A/D Converter

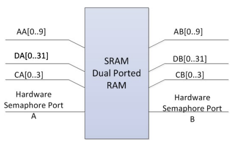
- Important Concepts
  - Long Wiring from data point to measuring system
    - High cable costs
    - high commissioning costs
    - Signal modified by wire
  - VME modules need certification
  - Access via memory mapped I/O
    - Dual-Ported Ram Interface for multiprocessors

Zürcher Hochschule für Angewandte Wissenschaften
zh aw School of Engineering
InES Institute of Embedded Systems

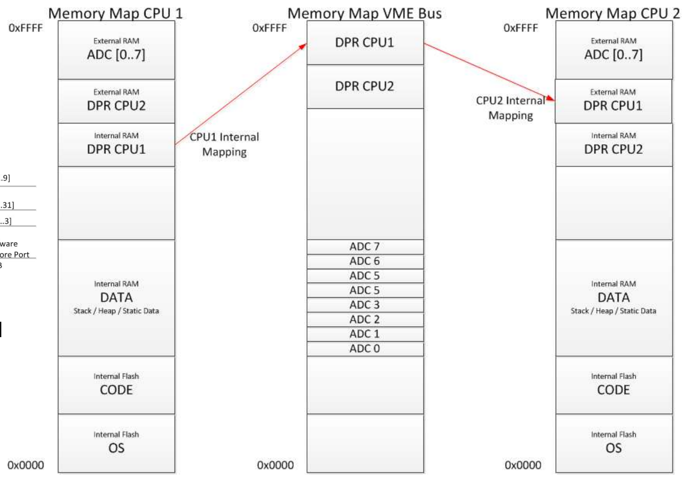
Centralised Control Systems

Session 1:
- Agenda
- **Centralised Control Systems**
- Remote I/O
- Fieldbus
- Real Time Ethernet
- Conclusion

- Memory Map and Dual-Ported RAM



Dual-Ported RAM



Centralised Control Systems

Code

```c
// Partial Memory Map CPU 1
#define ADC0_ADD 0xF800
#define ADC1_ADD 0xF900
#define ADC2_ADD 0xFA00
#define ADC3_ADD 0xFB00
#define ADC4_ADD 0xFC00
#define ADC5_ADD 0xFD00
#define ADC6_ADD 0xFE00
#define ADC7_ADD 0xFF00

#define DPR_CPU2 0xE000;
#define DPR_CPU1 0xC000;

void raise_alarm(void);
```

```c
//CPU 1
void main (void) {
    char monitoring = TRUE;
    // AD Converter 0
    int *pADC0_value;
    pADC0_value = ADC0_ADD;

    // DPR of CPU 2
    int *alarm;
    alarm = 0xE000;
    *alarm = FALSE;

    while ( monitoring ) {
        if ( *pADC0_value > MAX_VALUE ) {
            raise_alarm();
            *alarm = TRUE;
        }
    }
}
```

```c
// CPU 2
void main (void) {
    char monitoring = TRUE;

    // DPR of CPU 2
    int *alarm;
    alarm = 0xC000;

    while ( monitoring ) {
        if ( *alarm == TRUE ) {
            raise_alarm();
        }
    }
}
```

Max possible speed of loop and speed of communication (i.e CPU + bus latencies) determines speed of application

Centralised Control Systems: Summary

School of Engineering
InES Institute of Embedded Systems

Session 1:
- Agenda
- **Centralised Control Systems**
- Remote I/O
- Fieldbus
- Real Time Ethernet
- Conclusion

What do we need to remember

- First industrial measurement and control systems with centralised CPU
- Long wires between signal source and signal conversion
- Fast I/O (~RAM latency) due to memory mapping
- Dual Ported RAM as a controller-controller interface
- Loop processing time strictly dependent on CPU speed and communication latency

Problems

- Centralised systems tend to be more expensive
- Problems if long distances between signal source and signal conversion (noise)

Solutions

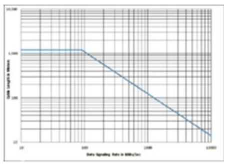
- Use cheaper HW
- Connect signal conversion directly at signal source

Remote I/O (RIO) Measurement and Control Systems

Session 1:
- Agenda
- Centralised Control Systems
- **Remote I/O**
- Fieldbus
- Real Time Ethernet
- Conclusion

RS232 full-duplex; >20'000 bit/s; max 50m;
replaced by USB
point 2 point

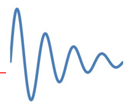
RS422 full-duplex,
10Mbit/s (20m)
19kBit/s (1200m)
1Tx max 10Rx



https://upload.wikimedia.org/wikipedia/commons/thumb/5/59/RS-422_CableLength-DataRate.png/220px-RS-422_CableLength-DataRate.png

RS485 half-duplex,
30Mbit/s (10m)
100kBit/s (1200m)
Multidrop

50 m / 1.2 km / 1.5 km



CE

http://www.directindustry.com/industrial-manufacturer/interface-card-76180-_2.html
http://jieshengda.sell.curiousexpeditions.org/pz6dd74e3-4-20ma-0-10v-to-rs485-rs232-a-d-converter.html
https://commons.wikimedia.org/wiki/File:Signal_processing.png

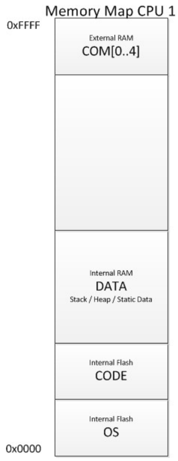
7

Remote I/O (RIO) Measurement and Control Systems

Zürcher Hochschule für Angewandte Wissenschaften
zh aw School of Engineering
InES Institute of Embedded Systems

- Memory Map and interface code

Session 1:
- Agenda
- Centralised Control Systems
- **Remote I/O**
- Fieldbus
- Real Time Ethernet
- Conclusion




```c
int open_port(void) {
    int fd; // file description for the serial port
    char *init_string = "hello sensor";

    fd = open("/dev/ttyS0", O_RDWR | O_NOCTTY | O_NDELAY);
    write(fd, init_string, 13);

    return (fd);
}
```


8

Remote I/O (RIO) Summary

Session 1:
- Agenda
- Centralised Control Systems
- **Remote I/O**
- Fieldbus
- Real Time Ethernet
- Conclusion

- What do we need to remember
  - First remote IO implemented as character streams
  - Distance and speeds dependent on cabling, termination, (weather?)
  - Long distances possible
- Problems
  - Each manufacturer had his own protocol and message format
    - Vendor lock-in and/or one cable per device
  - Slow transmission speeds == slow sampling rates
  - Horrendous cable problems (wrong termination, crossed cables …)
- Solutions
  - Strictly define cable, type and length; connector; termination, colour.
  - Define a protocol and a standard application interface

9

Fieldbus

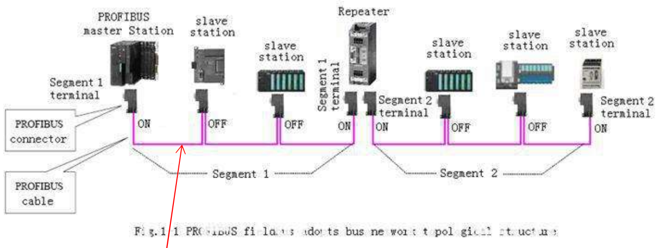
Session 1:
- Agenda
- Centralised Control Systems
- Remote I/O
- **Fieldbus**
- Real Time Ethernet
- Conclusion

- Why
  - Problems with RIO cause a substantial overhaul
- What
  - Applications
  - Architectures
  - Traffic types
  - Profiles
  - System Start-up

10

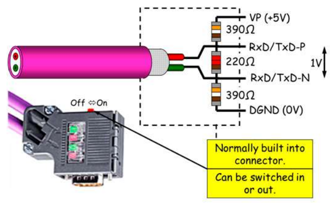
Fieldbus: Solution Example PROFIBUS (1)

Session 1:
- Agenda
- Centralised Control Systems
- Remote I/O
- **Fieldbus**
- Real Time Ethernet
- Conclusion



RS485

CAN based protocols (CANopen, DeviceNet) also hugely popular



http://www.profibus.com/newsroom/profinews-newsletter/profinews-2014-v2/profinews-117-singapore-takes-center-stage/
http://www.aliexpress.com/store/product/Model-FS-PBHUB-2-Isolated-PROFIBUS-MPI-PPI-repeater-Factory-direct-sale-Technical-support-tech-fourstar/101406_845502212.html

11

Fieldbus Example PROFIBUS (1)

Session 1:
- Agenda
- Centralised Control Systems
- Remote I/O
- **Fieldbus**
- Real Time Ethernet
- Conclusion

- What’s so important about the fieldbus?
  - Traffic types defined
  - Separation of rate of communication and rate of application
  - First use of OSI model to describe protocols for process and machine control
  - Profiles take part of application<->communication interface
  - Communication models standardised
    - Point-to-point
    - Publisher-subscriber
    - Producer-consumer

- Conformance testing necessary
- Interoperability testing necessary
  - Plugfest

	User program	Application profiles
7	Application Layer	PROFIBUS DP Protocol (DP-V0, DP-V1, DP-V2)
6	Presentation Layer.	Not used
5	Session Layer	
4	Transport Layer	
3	Network Layer	
2	Data link Layer	Fieldbus Data Link (FDL): Master Slave principle Token principle
1	Physical Layer	Transmission technology
	OSI Layer Model	OSI implementation at PROFIBUS

http://www.tesla-institute.com/index.php/automation-articles/156-introduction-to-profibus

12

Fieldbus – Applications

Session 1:
- Agenda
- Centralised Control Systems
- Remote I/O
- **Fieldbus**
- Real Time Ethernet
- Conclusion

- Open Loop Control (Steuerung)
- Closed Loop Control (Regelung)
- All applications well understood
  - Little scope for inventing new paradigms
    - Automation (Manufacturing, injection moulding) 32 -> 4 ms -> 1 ms
    - Motion Control (printing, polishing) 1ms -> 125 us -> 31.25 us

- Control cycle based
  - Sample – calculate - actuate
  - Digital control ->deadlines need to be kept -> real time component
  - Cycle times
    - Process, (Water, chemicals …) Building Automation x s -> 32 ms

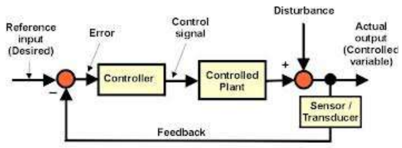


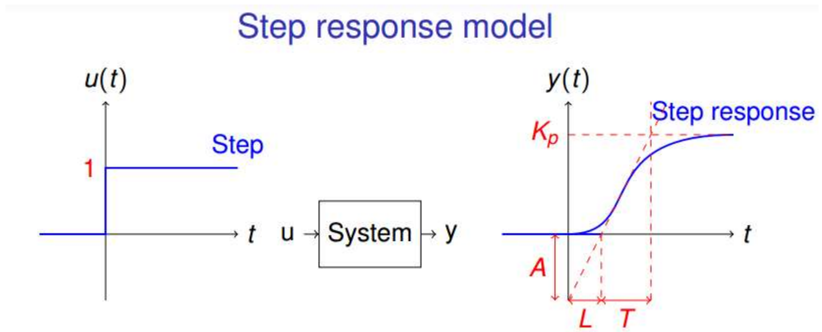
Fig.15: Feedback control system

http://blog.oureducation.in/open-loop-and-closed-loop-control-system/

Fieldbus – Control Applications

Session 1:
- Agenda
- Centralised Control Systems
- Remote I/O
- **Fieldbus**
- Real Time Ethernet
- Conclusion

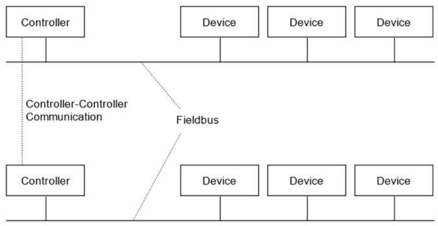
- Closed Loop Control (Regelung) – Responses
  - A system shows a lag time L
  - A constant gain K and
  - A time constant



http://http://www.hh.se/download/18.7502f6fd13ccb4fa8cb2f6/1360676928481/PID.pdf/

Fieldbus architectures

Zürcher Hochschule für Angewandte Wissenschaften
School of Engineering
InES Institute of Embedded Systems



source distributed_1.svg

Session 1:
- Agenda
- Centralised Control Systems
- Remote I/O
- **Fieldbus**
- Real Time Ethernet
- Conclusion

- Communication part of system philosophy
- Most applications best thought of as **controller-device**
  - Used to be called master-slave
    - Easy to understand and program
  - Communication follows this thinking
  - Often separate protocols for controller-controller communication
  - Asymmetric computing power
    - higher in controller than device
- **Peer to peer** tends to be implemented on a higher logical levels

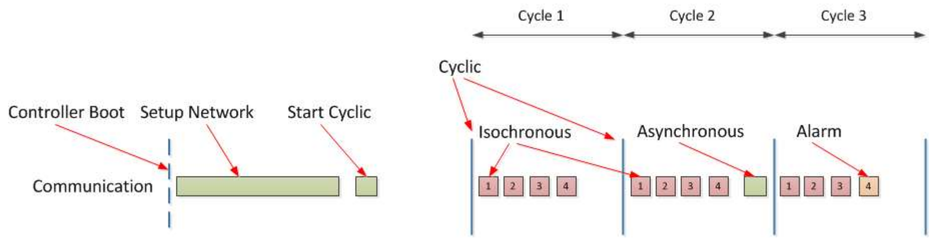
15

Fieldbus – Traffic Types

Zürcher Hochschule für Angewandte Wissenschaften
School of Engineering
InES Institute of Embedded Systems

Session 1:
- Agenda
- Centralised Control Systems
- Remote I/O
- **Fieldbus**
- Real Time Ethernet
- Conclusion

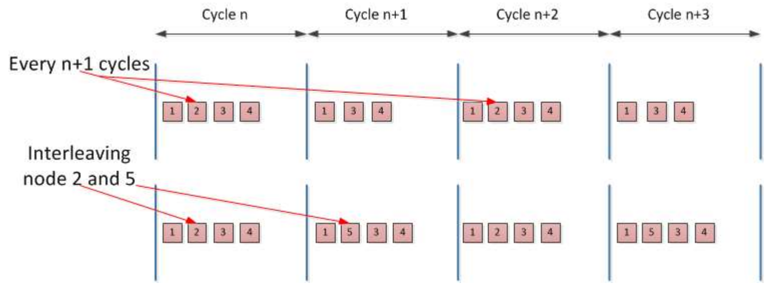
- Cyclic communication -> Isochronous
  - **Cyclic traffic** = normal operation
    - Data has **hard** deadlines but only 1 cycle validity
  - **Setup**
    - Setup has no hard deadlines but **must** be delivered
  - **Alarms**
    - Alarms have **hard** deadlines and **must** be delivered



16

Fieldbus - Cyclic communication

- Scheduling can be applied



- Traffic types seen in other protocols: example USB
  - Isochronous (real time), bulk (asynchronous), interrupt (alarm), control (setup)

Session 1:
- Agenda
- Centralised Control Systems
- Remote I/O
- **Fieldbus**
- Real Time Ethernet
- Conclusion

17

Important Lessons

- Distributed Systems generally implemented as Controller-Device systems

- Current distributed systems run in cycle times from seconds to 31.25 us

- Three general traffic/data categories
  - Isochronous Data (real time)
  - Control Data (setup)
  - Alarms

Session 1:
- Agenda
- Centralised Control Systems
- Remote I/O
- **Fieldbus**
- Real Time Ethernet
- Conclusion

18

Exercise

Session 1:
- Agenda
- Centralised Control Systems
- Remote I/O
- **Fieldbus**
- Real Time Ethernet
- Conclusion

Application: Sleep control (monitoring ?). Wireless sensors placed across the mattress sensing small motions, like breathing and heart rate and large motions caused by tossing and turning during sleep, providing data available through an app on the smart phone.

- Lets presume this thing is in a hospital and lets make the following assumptions:
  - If the heart rate and breathing sensors don't detect a heartbeat after 5 seconds and the patient isn't tossing and turning then a warning is sent to the app. After 10 seconds and an alarm is sent
  - The doctor wants to see the relationship between tossing and turning and heartbeat every morning
  - As an option the nightwatch can elect to project the heartbeat onto a screen in his/her office
- Determine the transfer types for each case

Fieldbus - Standardised Profile (1)

Session 1:
- Agenda
- Centralised Control Systems
- Remote I/O
- **Fieldbus**
- Real Time Ethernet
- Conclusion

- The most important innovation from fieldbus technology
  - Is a standardised interface for both application and controller
- Can be abstracted as a DPR with globally defined addresses for globally defined variables/parameters
- Instead of addresses we speak of indexes

Index Range	Description
0000h	Reserved
0001h-0FFFh	Data Types
1000h-1FFFh	Communication Entries
2000h-5FFFh	Manufacturer Specific
6000h-9FFFh	Device Profile Parameters
A000h-FFFFh	Reserved

Index Range	Description
001h-001Fh	Standard Data Types
0020h-0023h	Pre-Defined Complex Data Types
0024h-003Fh	Reserved
0040h-005Fh	Manufacturer Complex Data Types
0060h-007Fh	Device Profile Standard Data Types
0080h-009Fh	Device Profile Complex Data Types
00A0-025Fh	Multiple Device Module Types
0260h-0FFFh	Reserved

CAN Open mapping

Fieldbus - Standardised Profile (2)

Session 1:
- Agenda
- Centralised Control Systems
- Remote I/O
- **Fieldbus**
- Real Time Ethernet
- Conclusion

- Communication profile used in setup
- The application profile, that what makes the node, in this case a temperature sensor
- The vendor specific profiles – hidden from view it allows vendors to implement special features
- USB offers similar profiles called Device Descriptors

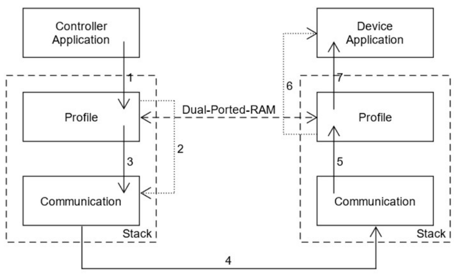
Index	Name
1000h	Device Type
1001h	Error Register
1002h	Manufacturer Status Register
1003h	Pre-defined Error Field
1005h	COB ID SYNC
1006h	Communication Cycle Period
1007h	Synchronous Window Length
1008h	Manufacturer Device Name
1009h	Manufacturer Hardware Version
100Ah	Manufacturer Software Version
100Ch	Guard Time
1200h-127Fh	Server SDO Parameters
1280h-12FFh	Client SDO Parameters
1400h-15FFh	RxPDO Communication Parameters
1600h-17FFh	RxPDO Mapping Parameters
1800h-19FFh	TxPDO Communication Parameters
1A00h-1BFFh	TxPDO Mapping Parameters

CAN Open Communication Profile

Fieldbus - Standardised Profile (3)

Session 1:
- Agenda
- Centralised Control Systems
- Remote I/O
- **Fieldbus**
- Real Time Ethernet
- Conclusion

- The Controller application writes data into his copy of the profile.
- The communication stack transfers this data, according to the correct data traffic type to the node
- The node stack receives this data and maps it into the correct position in the device profile
- The node stack notifies the node application which then takes appropriate action



source distributed_2.svg

Important Lessons

Zürcher Hochschule für Angewandte Wissenschaften
zh aw School of Engineering
InES Institute of Embedded Systems

Session 1:
- Agenda
- Centralised Control Systems
- Remote I/O
- **Fieldbus**
- Real Time Ethernet
- Conclusion

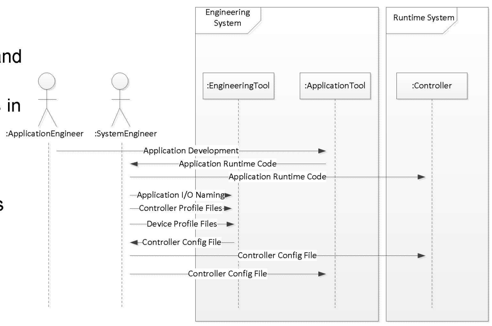
- Profiles allow every master in a network to access device capabilities in a standard way
- This means that a system engineer can swap a device from vendor A for one from vendor B ***without*** modifying the application software
- The standard/device tells what format the data is in
- The master can configure the **format** of the data delivery from device!

Fieldbus - System/Network setup

Zürcher Hochschule für Angewandte Wissenschaften
zh aw School of Engineering
InES Institute of Embedded Systems

Session 1:
- Agenda
- Centralised Control Systems
- Remote I/O
- **Fieldbus**
- Real Time Ethernet
- Conclusion

- Fieldbus networks well defined
  - Each device has an Electronic Data Sheet (EDS)
    - The EDS contains the device profile as well as pre-defined values
    - Often in XML format
  - The system engineer reads the EDS of all devices and the controller into an engineering tool
  - The system engineer configures system parameters in the engineering tool
  - The engineering tool spits out a EDS-like file for the controller
  - The controller uses this EDS-like file to configure his copies of the device profiles in his own memory
  - When the system is turned on the controller checks whether all devices are there the application runs
    - If «mandatory» devices missing, the controller shuts the network down
    - If «optional» devices missing the controller starts the network



Fieldbus – The Good, the Bad and the Ugly

zh aw School of Engineering InES Institute of Embedded Systems

- Good
  - Well defined and understood systems
    - With added effort - highly reliable and functionally safe systems
- Bad
  - Support for asynchronous-type communications
  - Modular Machinery
    - Substantial effort required to implement applications that use modular machinery
  - Communication Errors
    - Frequently lead simply to machine turn-off
- Ugly
  - Ad-hock changes
    - Any change has to go through system engineering

Summary

zh aw School of Engineering InES Institute of Embedded Systems

Session 1:
- Agenda
- Centralised Control Systems
- Remote I/O
- **Fieldbus**
- Real Time Ethernet
- Conclusion

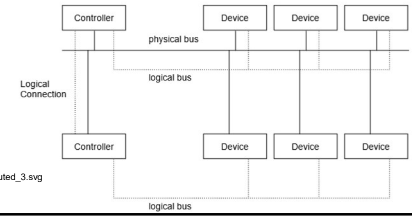
- Fieldbus hugely influential in many industries
  - Energy
  - Automotive
  - Manufacturing
  - Building automation

  - Didn't have much influence in medical or city environments
- Most important innovation is the profile
  - Defines the purpose of the device
- Question: to what degree will IoT replace the fieldbus?

Real Time Ethernet - Fieldbus V2.0

Session 1:
- Agenda
- Centralised Control Systems
- Remote I/O
- Fieldbus
- **Real Time Ethernet**
- Conclusion

- Real Time Ethernet (RTE)
- Highly deterministic communication over Ethernet
  - Varies from best-effort traffic (EtherNet/IP)
  - Highly deterministic communication (EtherCAT / PROFINET / Sercos III)
- Innovation: supports co-existance of legacy ethernet traffic
- Innovation: Advanced Physical Layer (APL)
  - Unshielded twisted pair
  - Power over Ethernet
  - > 1km lengths possible



source distributed_3.svg

Real-Time Ethernet: Fieldbus V2.0

Session 1:
- Agenda
- Centralised Control Systems
- Remote I/O
- Fieldbus
- **Real Time Ethernet**
- Conclusion

- Some RTE protocols use legacy Ethernet protocols
- EtherNet/IP uses UDP/IP for RT frames
- PROFINET uses UDP/IP for RTC1 frames (ca. 10 ms cycles)
  - But proprietary frames for RTC2 (1 ms) and RTC3 (31.25 us)
- PROFINET uses Precision Time Protocol (PTP/PTCP) for time synchronisation
- PROFINET uses Link Layer Discovery Protocol (LLDP) for neighbour/topology discovery
  - But proprietary DCP for initial setup (hostname, IP address ..)
- PROFINET uses RPC for system setup
- PROFINET uses SNMP for network management

Conclusion

Session 1:
- Agenda
- Centralised Control Systems
- Remote I/O
- Fieldbus
- Real Time Ethernet
- **Conclusion**

- Dual Ported RAM as Communication Interface
  - Known position of variables
- Cost, technical and handling reasons for serial communication/distributed control
- Manufacturer independent communication protocols required
  - Define standardised physical layers
  - Define different traffic types
  - Define application profiles residing in DPR-like structures
  - Define a process for setting up distributed systems
- Ethernet in Distributed Embedded Systems
  - “Legacy” tools for embedded systems
  - Long distances due to APL physical layer.