

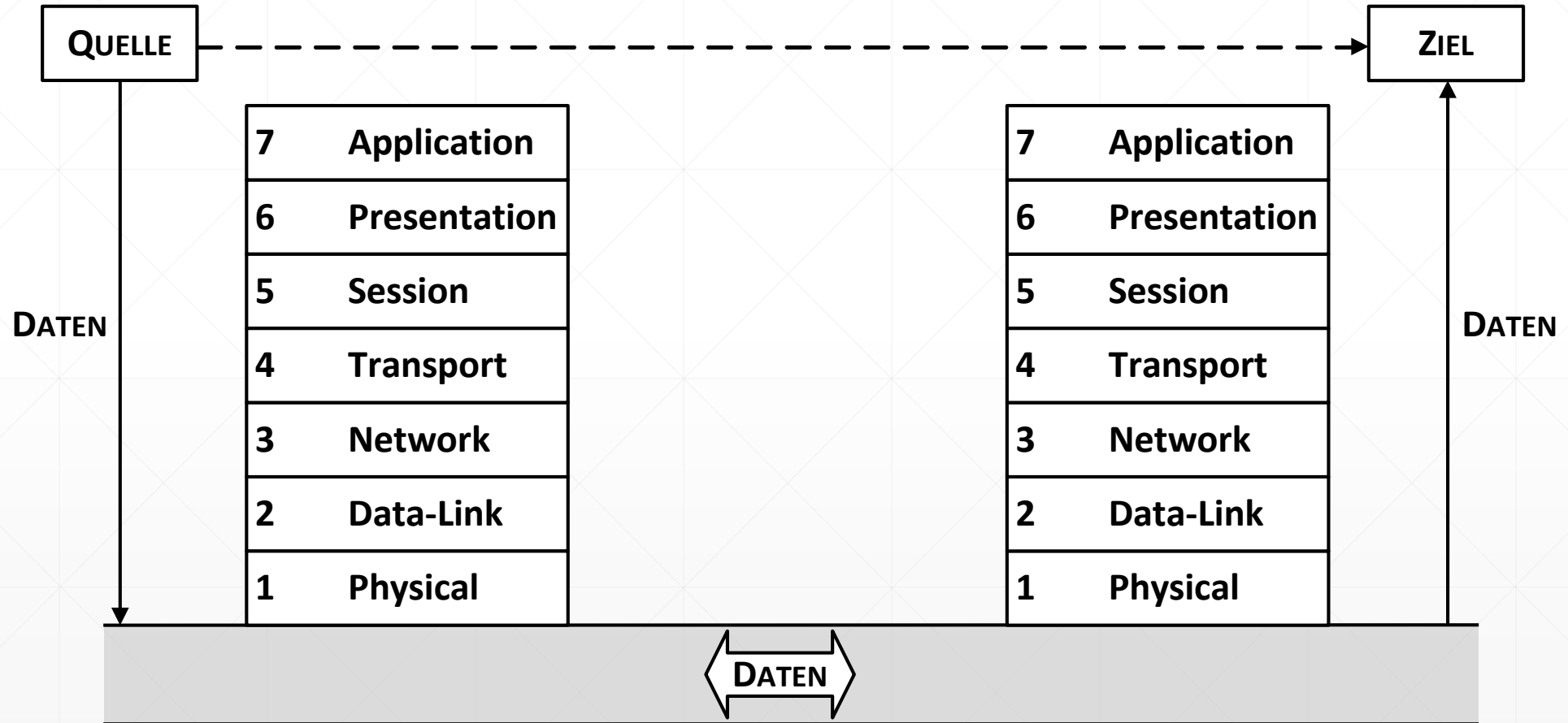
Ethernet – Embedded LAN Designs

Referat von Peter Voser
Embedded Development GmbH

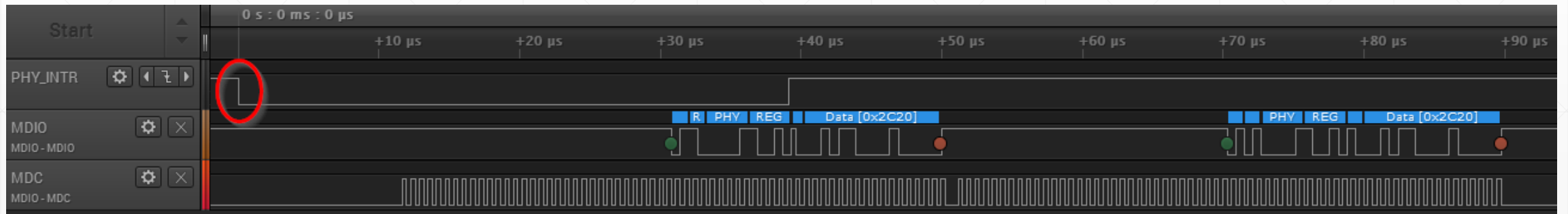
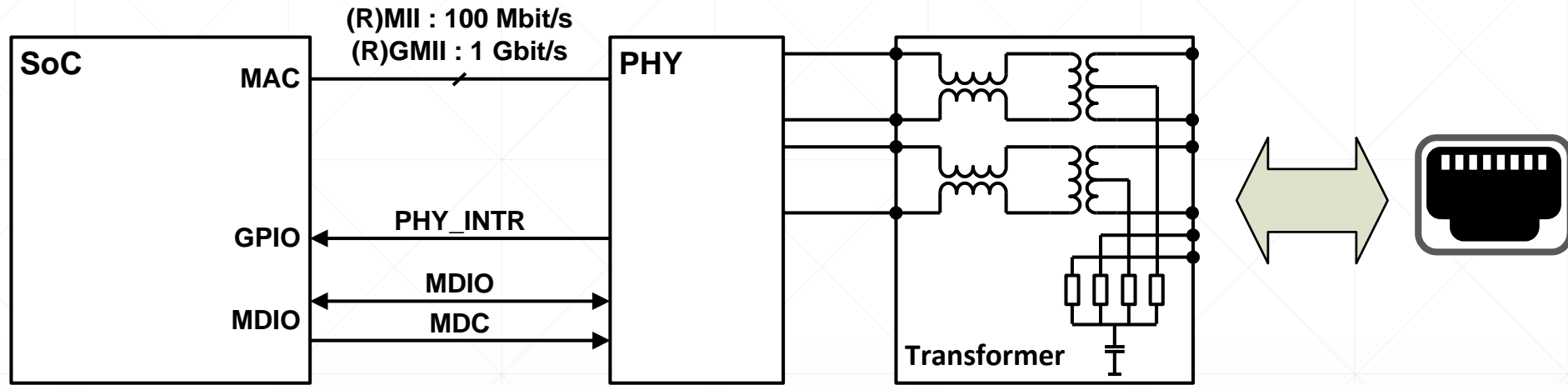


Kommunikation in Schichten

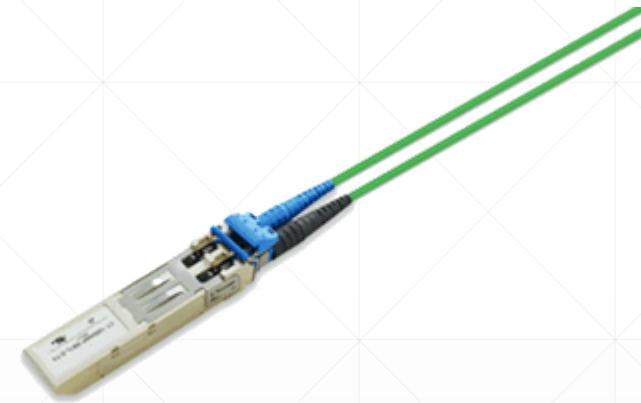
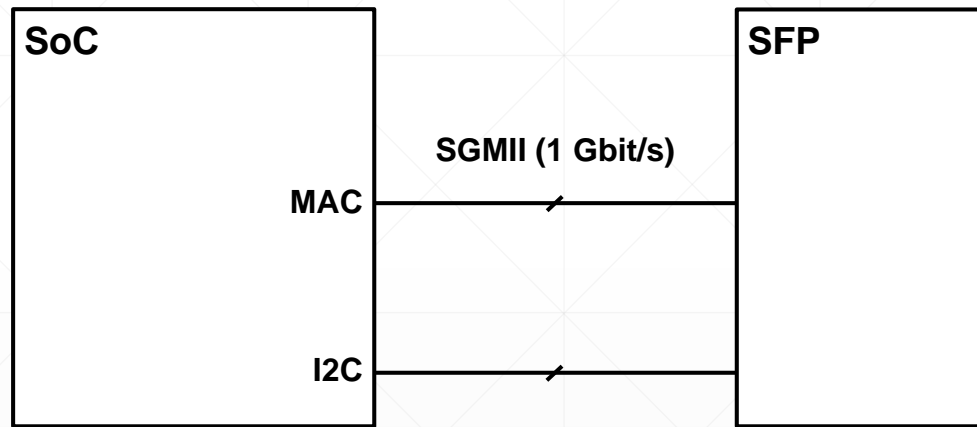
Das OSI-Modell



Layer 1 Kupfer PHY



Layer 1 Gigabit SFP für Glasfasern



SFP = Small Form-factor Pluggable

Layer 2

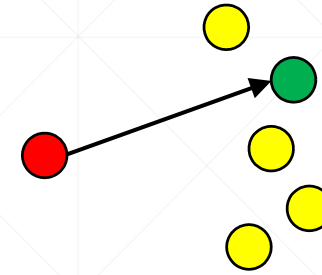
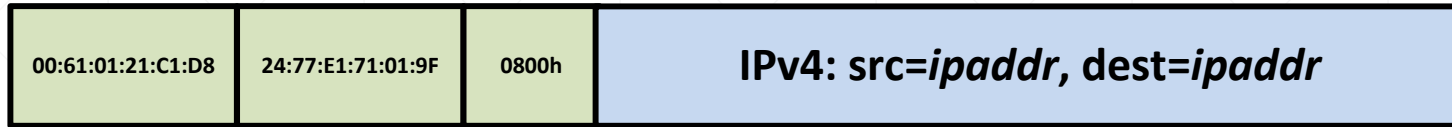
Das Ethernet Frame



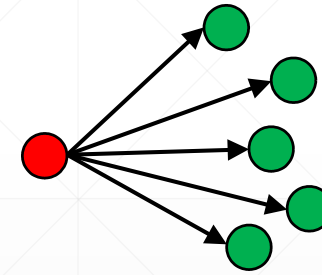
- P** Preamble
- DA** Destination MAC Address
- SA** Source MAC Address
- L/T** Length/Type
- FCS** Frame Check Sequence (CRC32)

Layer 2 Ethernet Frame Beispiele

Unicast Frame



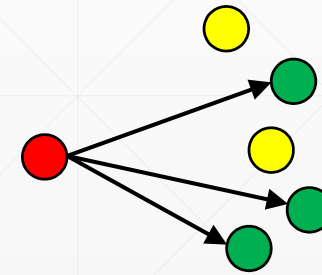
Broadcast Frame



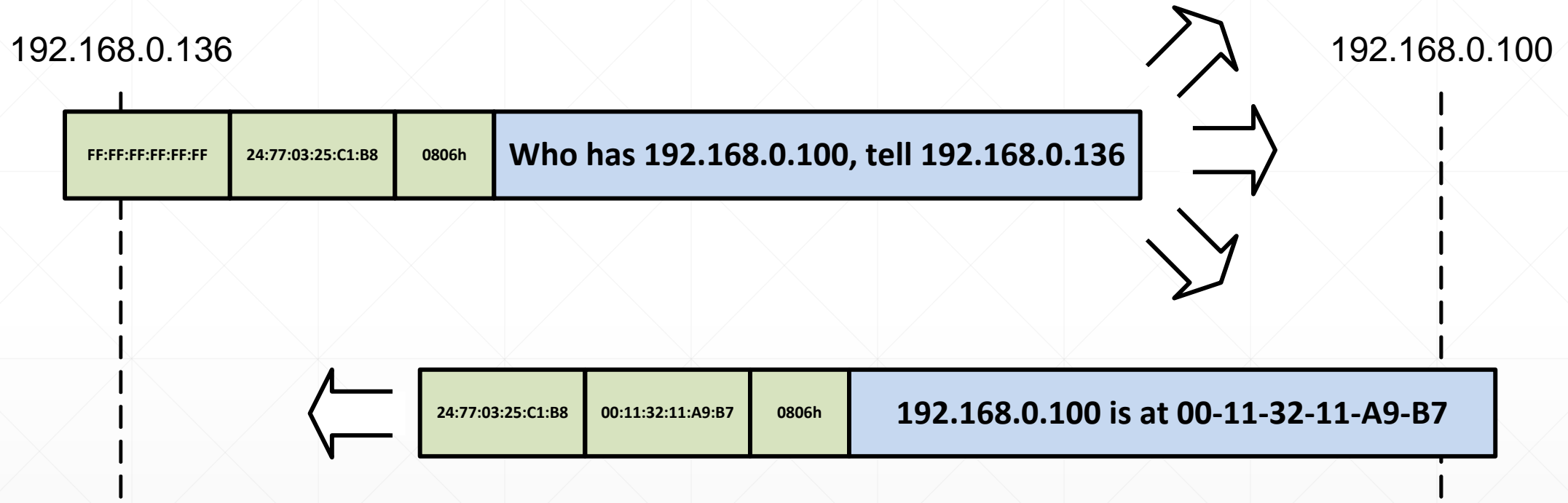
Multicast Frame



Bit0 = 1

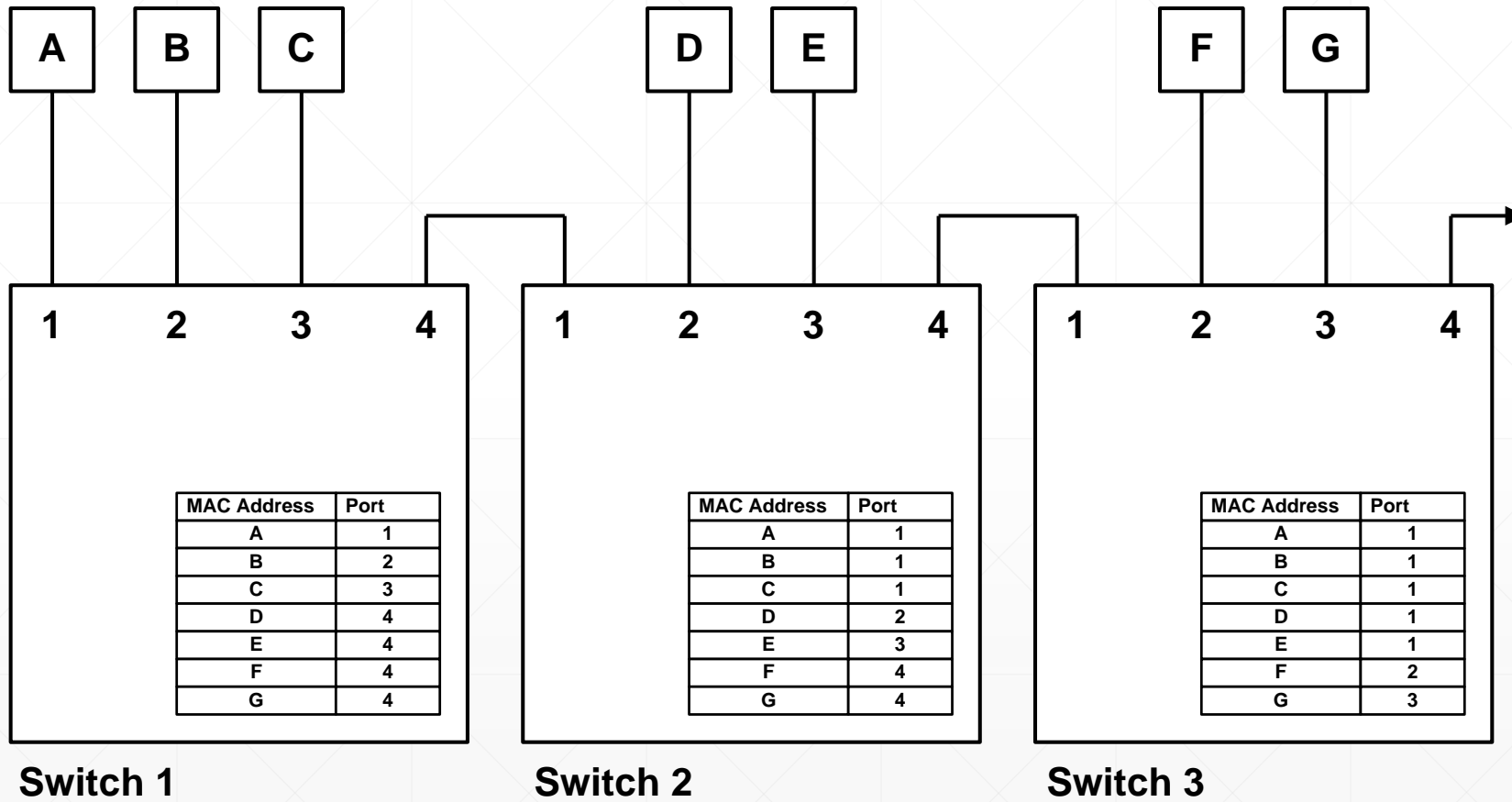


Bindeglied zwischen Network und Link Layer ARP Address Resolution Protocol



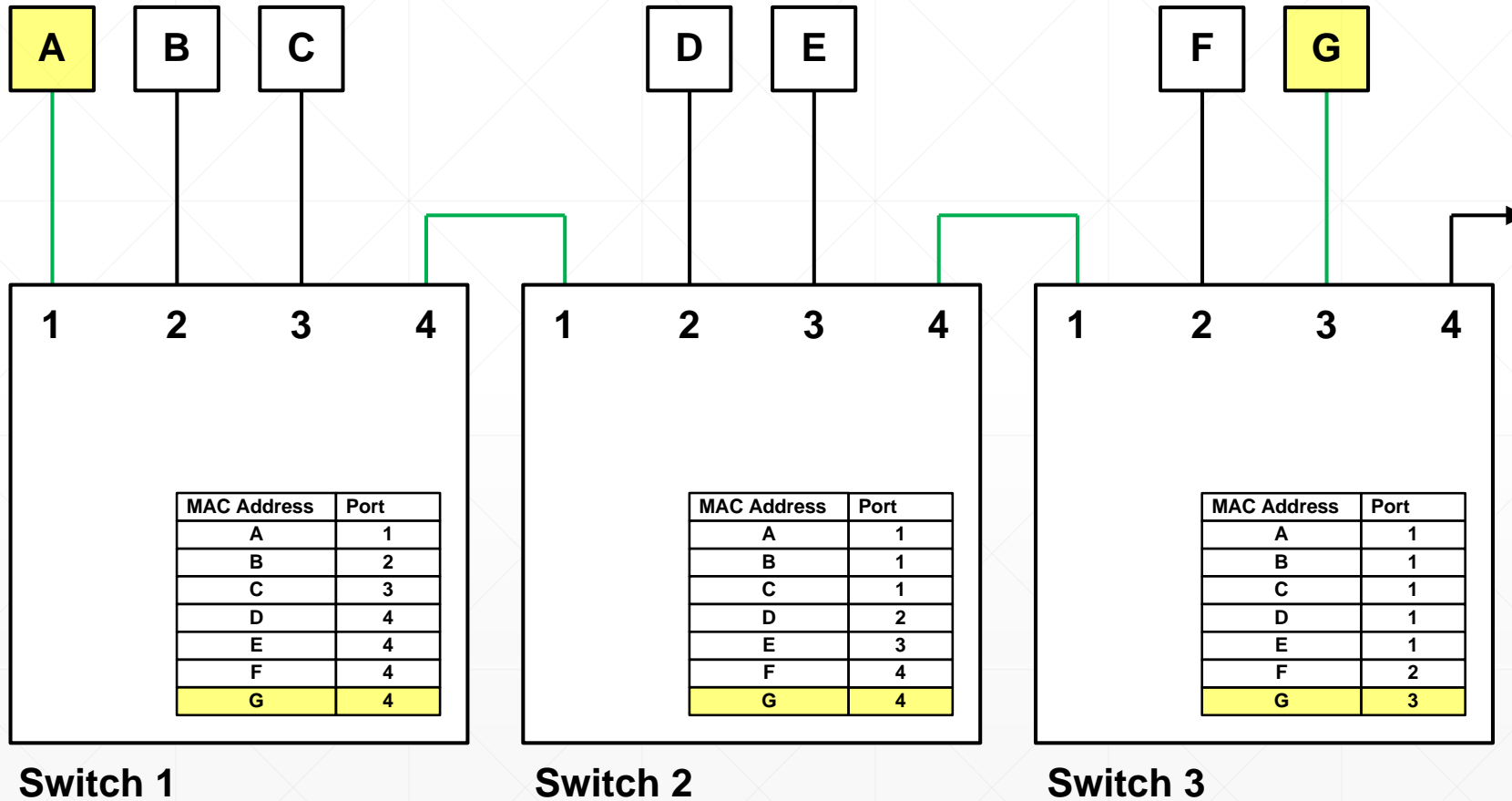
| No. | Time | Source | Destination | Protocol | Length | Info |
|-----|----------|-------------------|-------------------|----------|--------|---|
| 1 | 0.000000 | 24:77:03:25:c1:b8 | ff:ff:ff:ff:ff:ff | ARP | 42 | Who has 192.168.0.100? Tell 192.168.0.136 |
| 2 | 0.003460 | 00:11:32:11:a9:b7 | 24:77:03:25:c1:b8 | ARP | 60 | 192.168.0.100 is at 00:11:32:11:a9:b7 |
| 3 | 0.003479 | 192.168.0.136 | 192.168.0.100 | ICMP | 74 | Echo (ping) request id=0x0001, seq=4688/20498, ttl=128 (reply in 4) |
| 4 | 0.005444 | 192.168.0.100 | 192.168.0.136 | ICMP | 74 | Echo (ping) reply id=0x0001, seq=4688/20498, ttl=64 (request in 3) |

Layer 2 Ethernet Switch Bilden eines Netzwerks



Layer 2 Ethernet Switch

Auf direktem Weg von A nach G

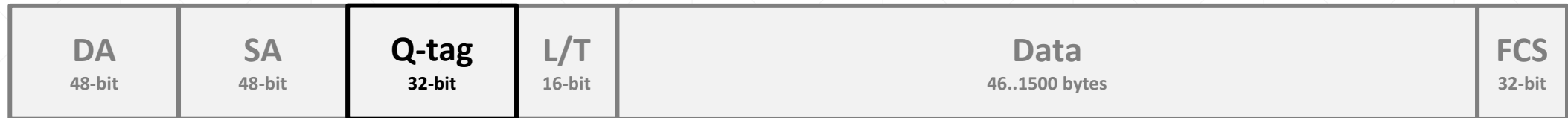


Layer 2 Ethernet Switch Address Learning

| Hash(MacAddr) | Bin #1 | Bin #2 | Bin #3 |
|----------------|-----------------------------|-----------------------------|--------|
| H ₁ | | | |
| H ₂ | 00:91:0B:11:7F:32 Port 3 | | |
| H ₃ | | | |
| H ₄ | 00:50:03:FF:00:01 Port 3 | 04:A7:10:78:51:11 Port 7 | |
| H ₅ | | | |
| ... | | | |
| H _n | | | |

Logische Gruppen mit VLAN

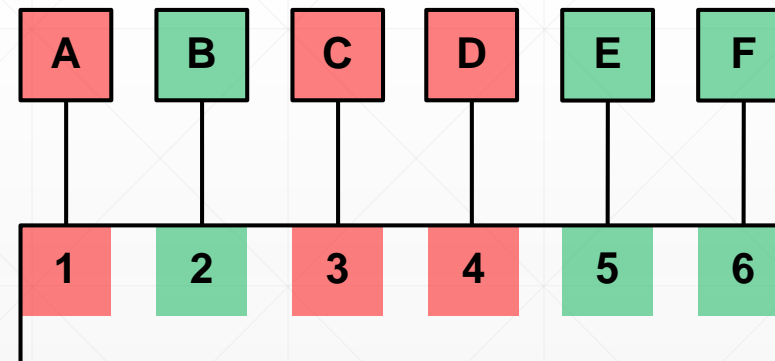
L2 IEEE 802.1Q tagged Frames



TPID 16-bit Erkennung 8100h
VID 12-bit VLAN ID 0..4094
DEI 1-bit drop indicator
PCP 3-bit priority code point

VLAN ID 100 Ports 1, 3, 4

VLAN ID 101 Ports 2, 5, 6



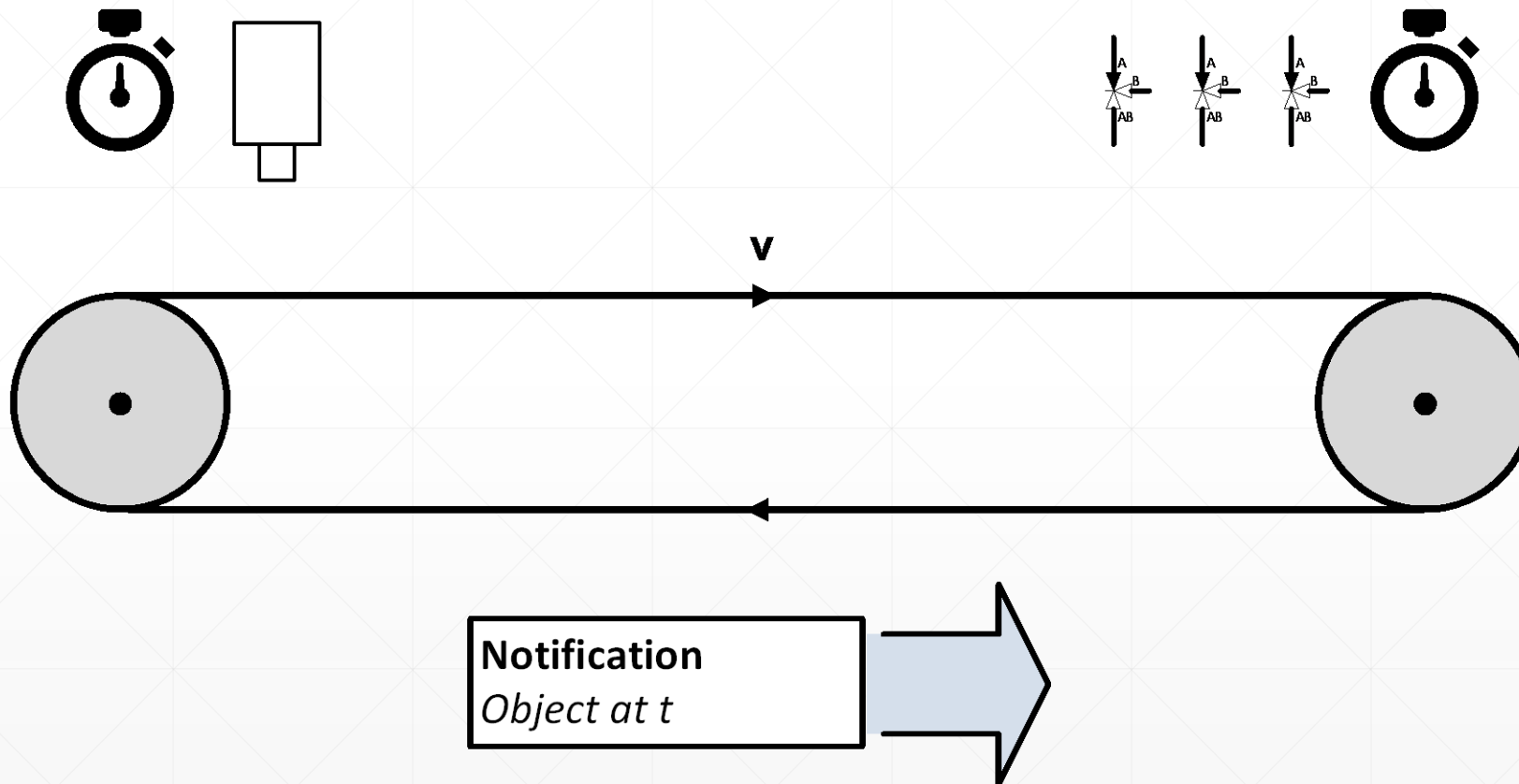
Switch

Marvell 88E6390 «Peridot»

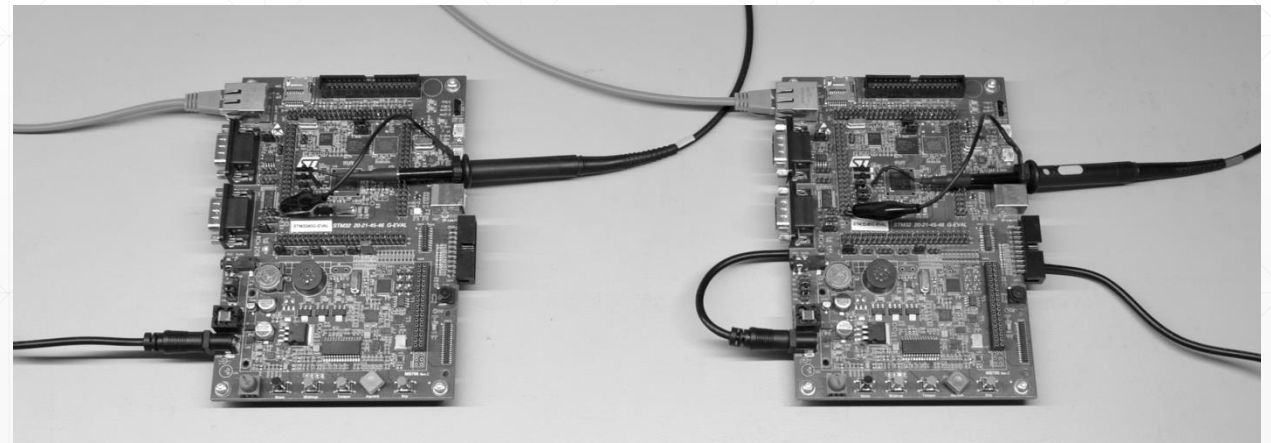
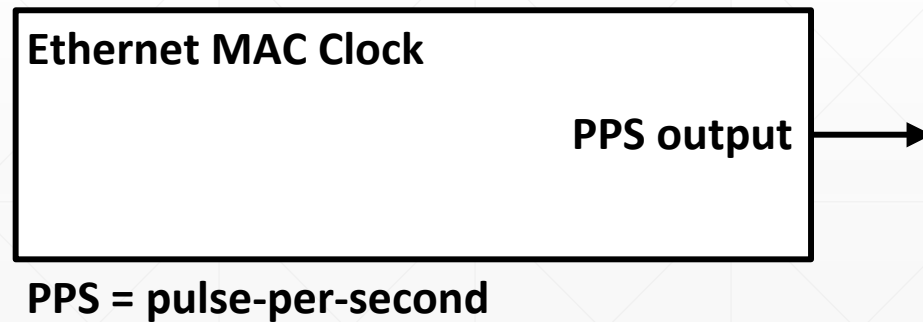
Ein moderner Switch für embedded Systeme

- 11-Port Gigabit AVB/TSN Switch
- 8 integrierte PHYs und 2 SERDES (Fiber)
- IMP (Integrated Management Processor) für Steuerungen und Paket Inspektion
- TCAM für Filter und Rate Limiting
- PTP Support
- Management und Konfiguration über Registermodell

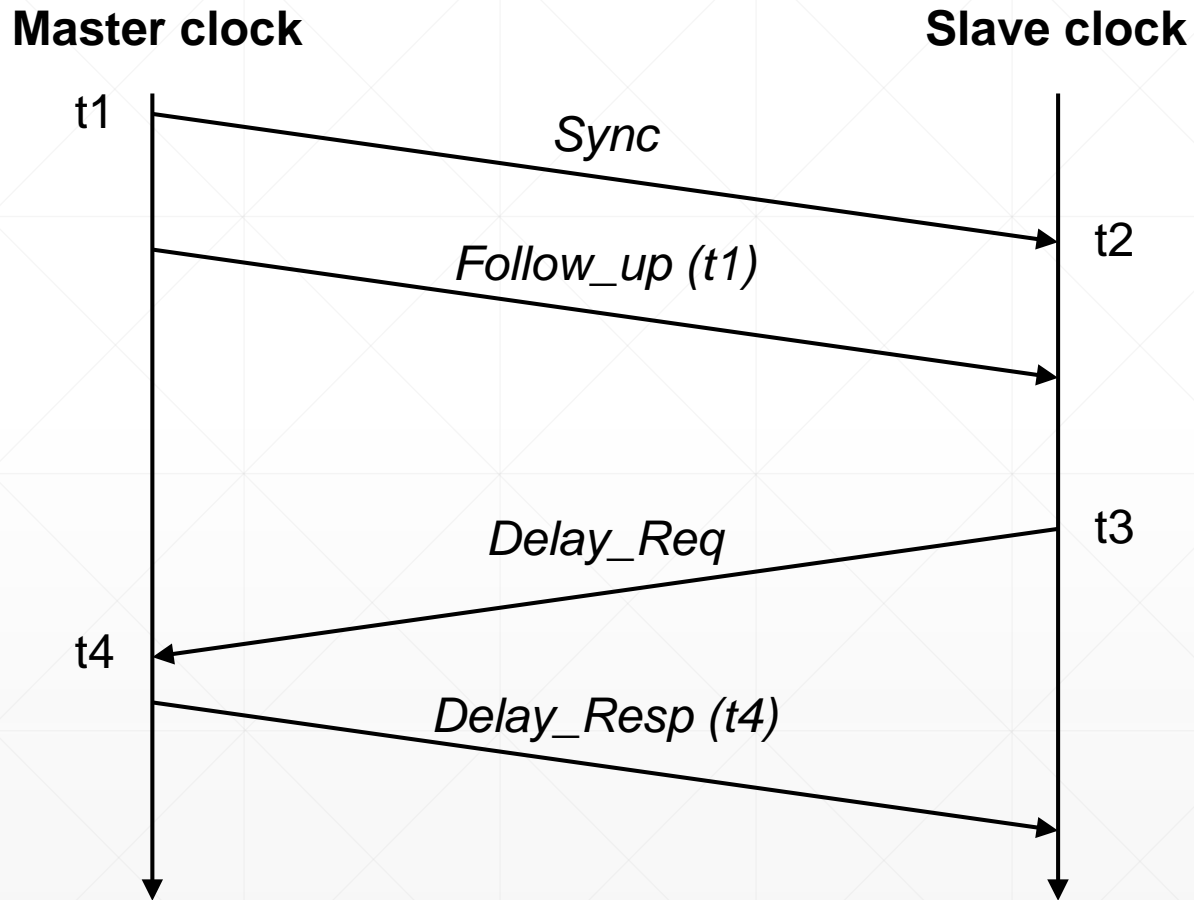
Precision Time Uhren synchronisieren



Precision Time Überblick

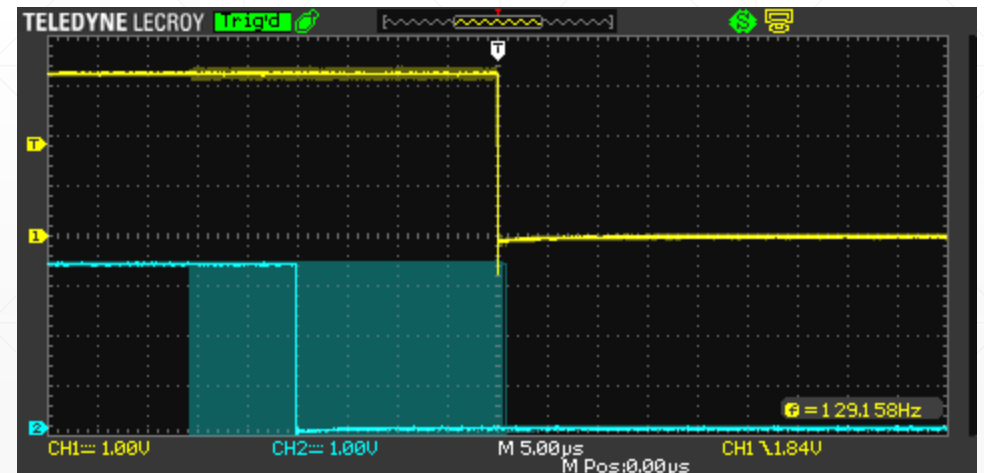


Precision Time End to End (Phase Correction)

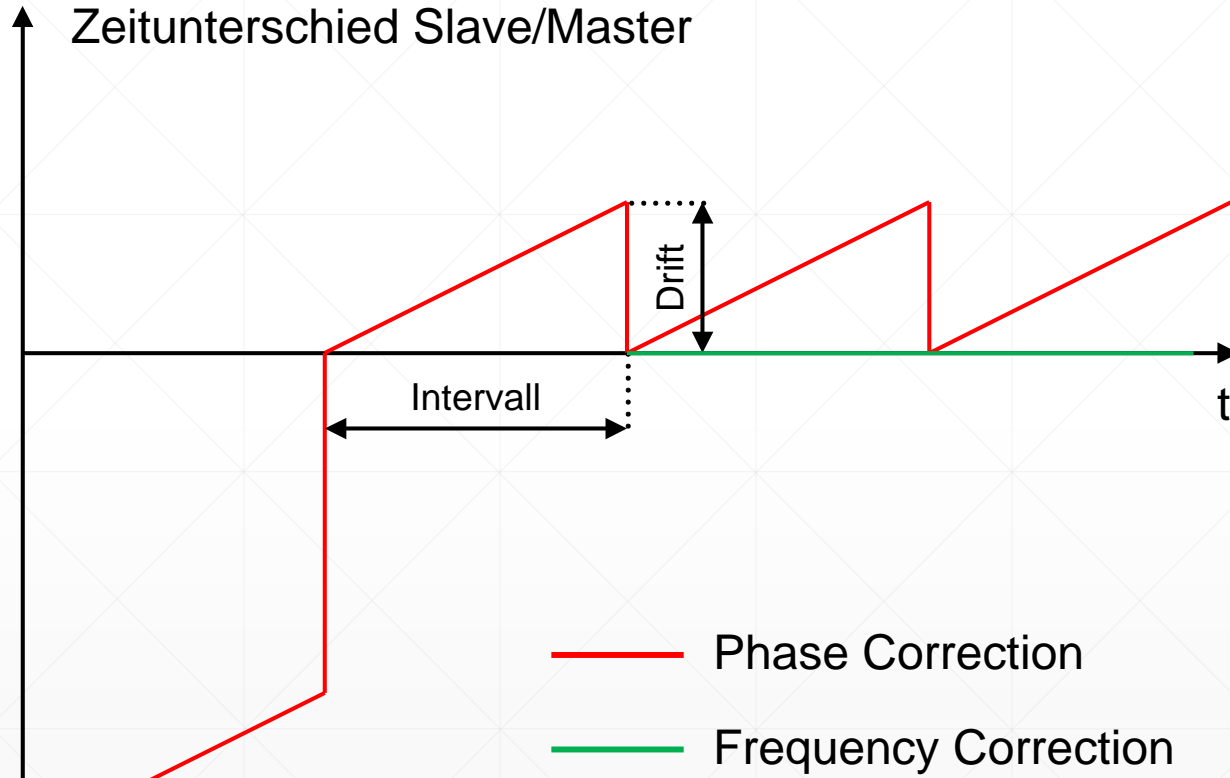


Verifikation am PPS Ausgang

- Sync alle 2'000ms
- Oszilloskope 5 μ s/div, persistence mode
- Drift infolge Quartz Ungenauigkeit

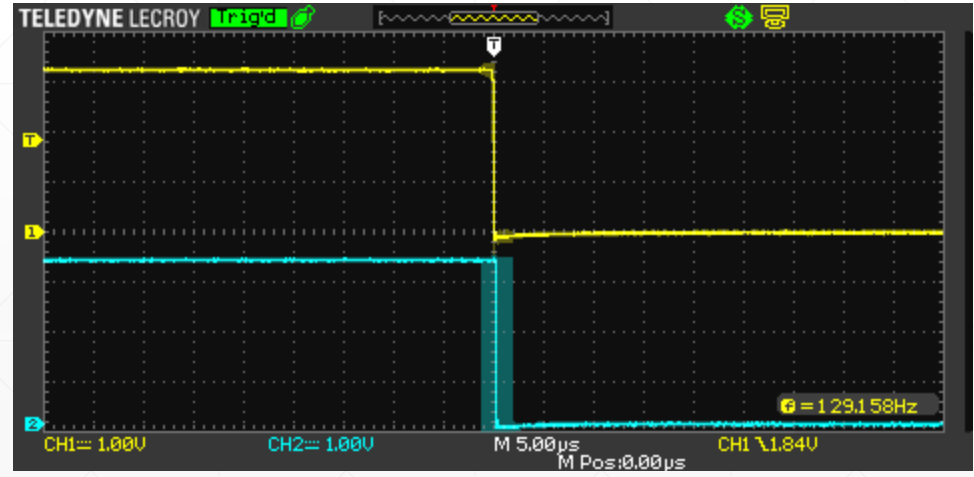


Precision Time End to End (Frequency Correction)

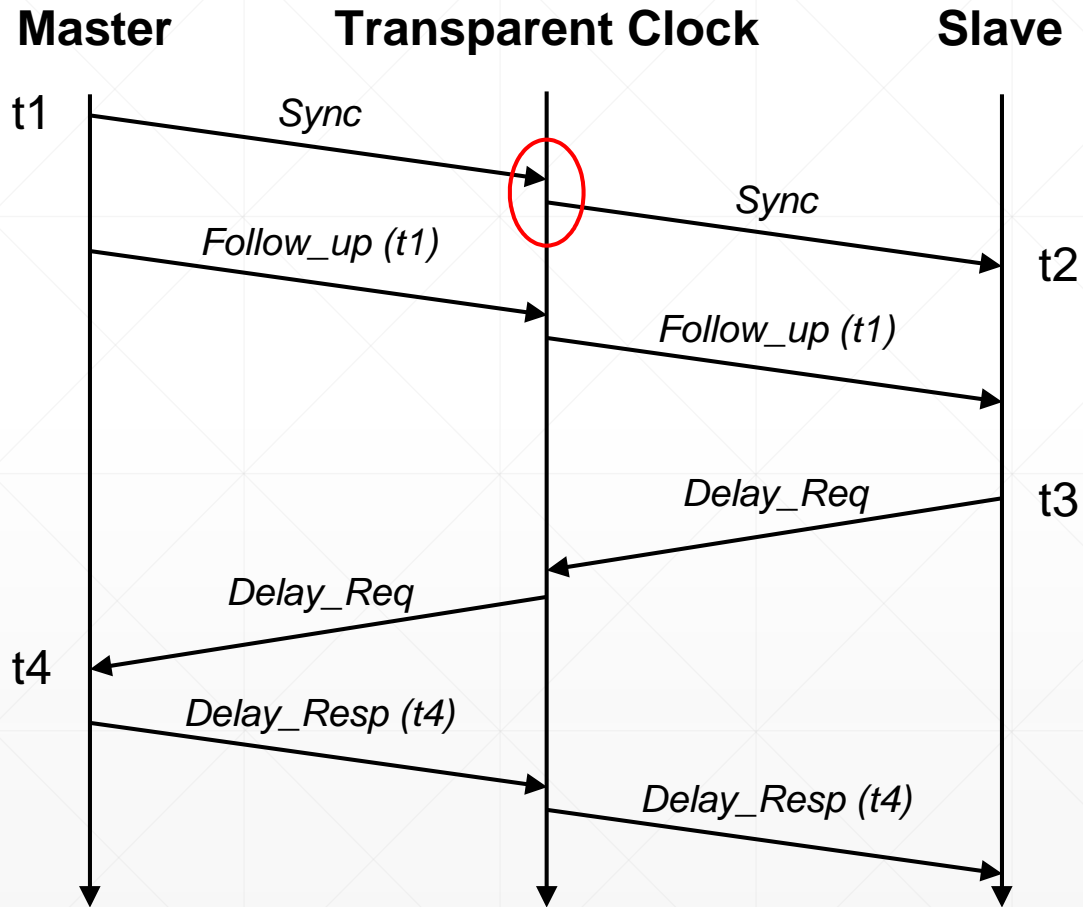


Verifikation am PPS Ausgang

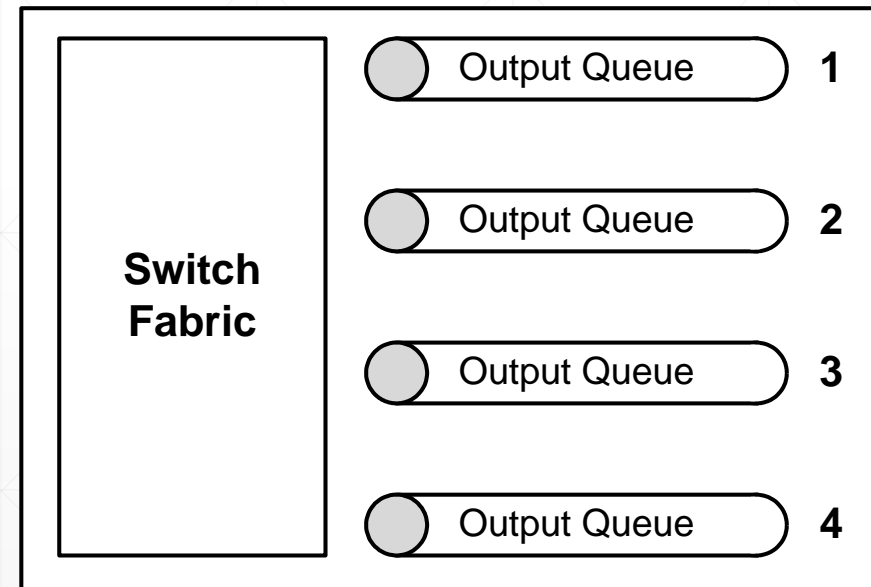
- Sync alle 2'000ms
- Oszilloskope 5µs/div, persistence mode
- Slave passt Frequenz dem Master an



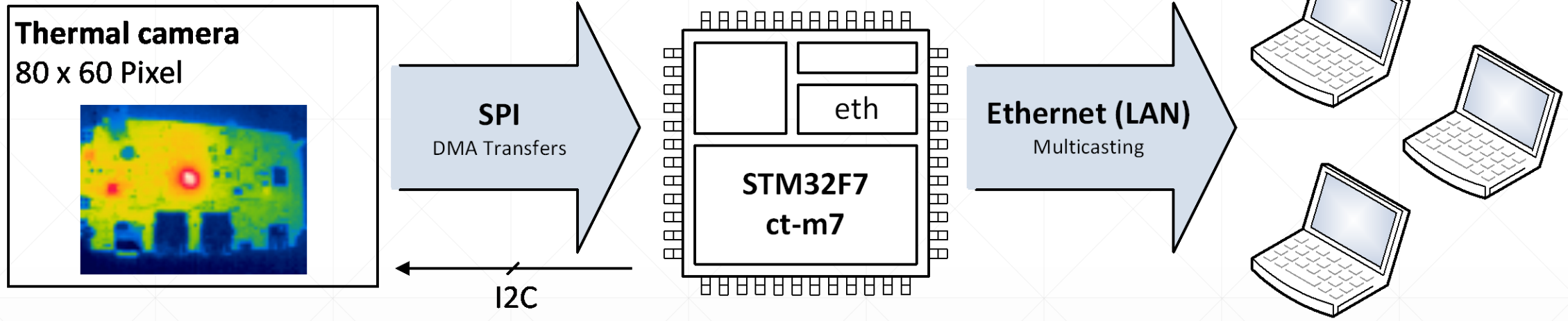
Precision Time mit Switch End to End Transparent Clock



Der Switch misst die Transitzeit der PTP Messages und stellt sie zur Verfügung.



Demo Wärmebild Kamera Multicasting



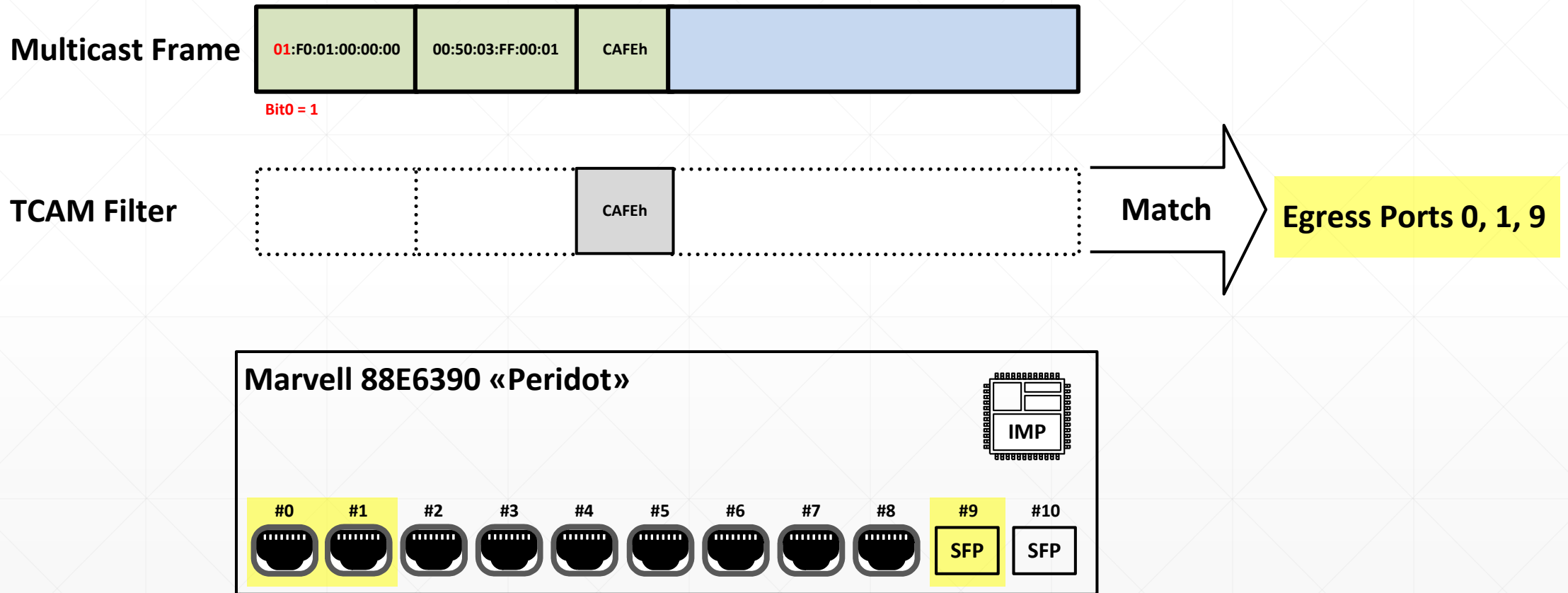
Aufteilung eines 4800-Byte Bilds in Frames:

- FN** Frame number
- CH** Chain ID (0=begin, 1=continue, 2=end)



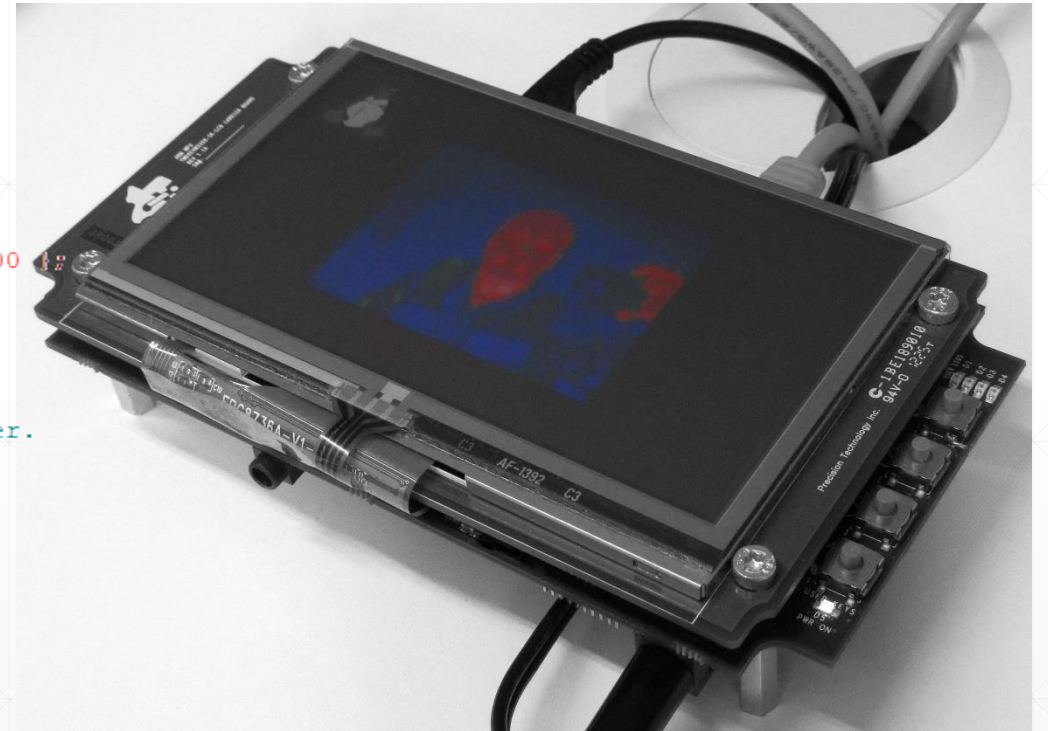
Demo

Switch lenkt MC Frame mittels TCAM

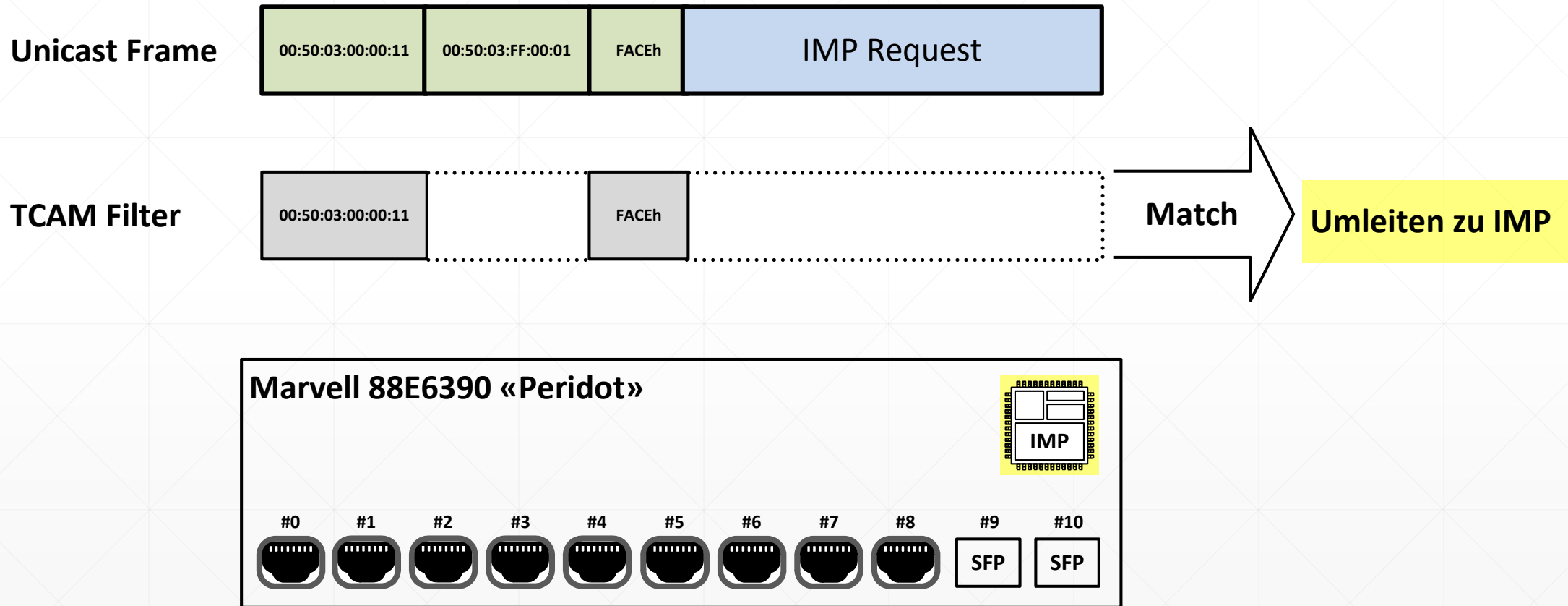


Demo Linux für L2

```
3 #include <linux/netdevice.h>
4
5 static struct packet_type flirFrameType __read_mostly =
6 {
7     .type = cpu_to_be16(0xcafe),
8     .func = handleFlirFrame,
9 };
10
11 static const uint8_t MULTICAST_ADDR[] = { 0x01, 0xf0, 0x01, 0x00, 0x00, 0x00 };
12
13 // i) Register handler for Ethertype 0xcafe.
14 dev_add_pack(&flirFrameType);
15
16 // ii) Allow frames with this MC address to pass hardware MAC address filter.
17 dev_mc_add(picture.dev, MULTICAST_ADDR);
18
19 static int handleFlirFrame(struct sk_buff *skb,
20                          struct net_device *dev,
21                          struct packet_type *pt,
22                          struct net_device *orig_dev)
23 {
24     // Decode FLIR frame chain.
25 }
26
```



Demo Eigenes L2 Switch Management Protokoll



Demo

Eigenes L2 Switch Management Protokoll

IMP Management Interface

Discovered IMP Devices

| MAC Address | Information |
|-------------------|-----------------------------------|
| 00:50:0A:00:00:01 | V1.5.1 Build Jun 26 2017 11:34:37 |
| 00:50:0A:00:00:02 | V1.5.1 Build Jun 26 2017 11:34:37 |
| 00:50:0A:00:00:03 | V1.5.1 Build Jun 26 2017 11:34:37 |

Update : 5 sec-1

Network interface : [1] : Intel(R) Gigabit CT Desktop Adapter

Close Update

Switch Ports

| Port | Link | Speed | Duplex | ATU Entries |
|------|------|----------|--------|--|
| 0 | up | 1 gbps | Full | 14:D6:4D:18:2A:BD (7) , 68:05:CA:31:B3:84 (7) |
| 1 | down | - | - | |
| 2 | down | - | - | |
| 3 | up | 100 mbps | Full | B8:27:EB:59:00:00 (7) |
| 4 | up | 100 mbps | Full | B8:27:EB:59:00:01 (6) |
| 5 | down | - | - | |
| 6 | down | - | - | |
| 7 | up | 1 gbps | Full | 00:50:0A:00:00:01 (7) |
| 8 | up | 100 mbps | Full | 00:50:02:00:00:01 (7) |
| 9 | up | 1 gbps | Full | 00:50:0A:00:00:03 (7) , B8:27:EB:00:00:07 (6) , B8:27:EB:00:00:08 (7) , B8:27:EB:00:00:09 (7) , B8:27... |
| 10 | down | - | - | |

Device 00:50:0A:00:00:02 open

| Time | Source | Destination | Length | Info |
|----------|-------------------|-------------------|--------|-------------|
| *REF* | 68:05:ca:31:b3:84 | 01:50:43:00:00:00 | 18 | Ethernet II |
| 0.000861 | 00:50:0a:00:00:02 | 68:05:ca:31:b3:84 | 60 | Ethernet II |
| 0.000861 | 00:50:0a:00:00:03 | 68:05:ca:31:b3:84 | 60 | Ethernet II |
| 0.000864 | 00:50:0a:00:00:01 | 68:05:ca:31:b3:84 | 60 | Ethernet II |
| *REF* | 68:05:ca:31:b3:84 | 00:50:0a:00:00:02 | 18 | Ethernet II |
| 0.000846 | 00:50:0a:00:00:02 | 68:05:ca:31:b3:84 | 60 | Ethernet II |
| 0.032188 | 68:05:ca:31:b3:84 | 00:50:0a:00:00:02 | 18 | Ethernet II |
| 0.037101 | 00:50:0a:00:00:02 | 68:05:ca:31:b3:84 | 68 | Ethernet II |

Fragen?

Embedded Development GmbH
www.embedded-development.ch

Peter Voser

077 405 70 05 Mobile

peter.voser@voser-development.ch