

---

Großer Beleg

**Ausarbeitung eines  
Praktikumsversuches zum Design  
eines 1-Wire-Master-Controllers**

**Falk Niederlein**



# Gliederung

---

- 1 Allgemein**
- 2 Architektur**
- 3 1-Wire-Protokoll**
- 4 Praktikumsversuch**
- 5 Zusammenfassung**

# Allgemein

---

- ein Produkt der Firma Maxim
- Low-Cost-Bussystem, welches Mikrocontroller bzw. PCs über Twisted-Pair-Kabel verbindet
- lediglich eine Datenleitung vorhanden
- Kommunikation erfolgt im Halbduplex-Verfahren
- es existiert eine Reihe von Bauelementen:
  - ➔ Feuchtigkeitssensoren
  - ➔ Temperatursensoren
  - ➔ ID-Chips (z.B. Personenkontrolle)
  - ➔ ...



# Architektur - Basiskomponenten

---

## Busmaster

- beinhaltet die Kontrollsoftware
- repräsentiert durch PC bzw. Mikrocontroller

## 1-Wire-Device

- Devices gibt es in den verschiedensten Ausführungen für die unterschiedlichsten Einsatzgebiete
- Devices unterscheiden sich durch einzigartige Adresse

## Verbindungselement

- Verbindung mit dem Master

→ einfaches Master-Slave-Prinzip



## Parasite Power

- Energie für die Devices kann über die Datenleitung bezogen werden
- auf der Datenleitung liegt eine Spannung von 5V an, welche die Devices versorgt
- während der Bitübertragung werden die Devices durch einen internen Kondensator mit Strom versorgt

## Pull-Up-Widerstand

- wichtig für die Kommunikation
- bringt Datenleitung wieder in den *high*-Zustand, nachdem sie für die Bitübertragung auf *low* gezogen wurde
- für kurze Leitungen und wenige Devices sind 5k $\Omega$  üblich



# 1-Wire-Protokoll

---

## Erkennung

- Vorhandensein von Devices überprüfen
- Senden eines Reset-Impulses durch den Master (*low*-Ziehen der Datenleitung für 480  $\mu$ s)
- Devices antworten mit einem Präsenz-Impuls

## Adressierung

- jedes 1-Wire-Device besitzt eine einzigartige 64 Bits umfassende Adresse  $\rightarrow$  dient als Netzadresse
- Adresse beinhaltet die Bauteilzugehörigkeit, eine Seriennummer und einen CRC-Wert
- die Adresse wird nach dem Präsenz-Impuls vom Master erfragt; anschließend kann er sich mit einem Device verbinden



# 1-Wire-Protokoll

---

## Datenübertragung

- die Übertragung der Bits erfolgt in Time-Slots von 60-120  $\mu\text{s}$
- Einsen und Nullen werden durch unterschiedlich langes *low*-Ziehen der Datenleitung erzeugt

**Schreiben einer 1:** der Master zieht die Datenleitung für max. 15  $\mu\text{s}$  auf *low*, der Slave tastet bei 30  $\mu\text{s}$  ab

**Schreiben einer 0:** der Master zieht die Datenleitung für min. 60  $\mu\text{s}$  auf *low*, der Slave tastet bei 30  $\mu\text{s}$  ab

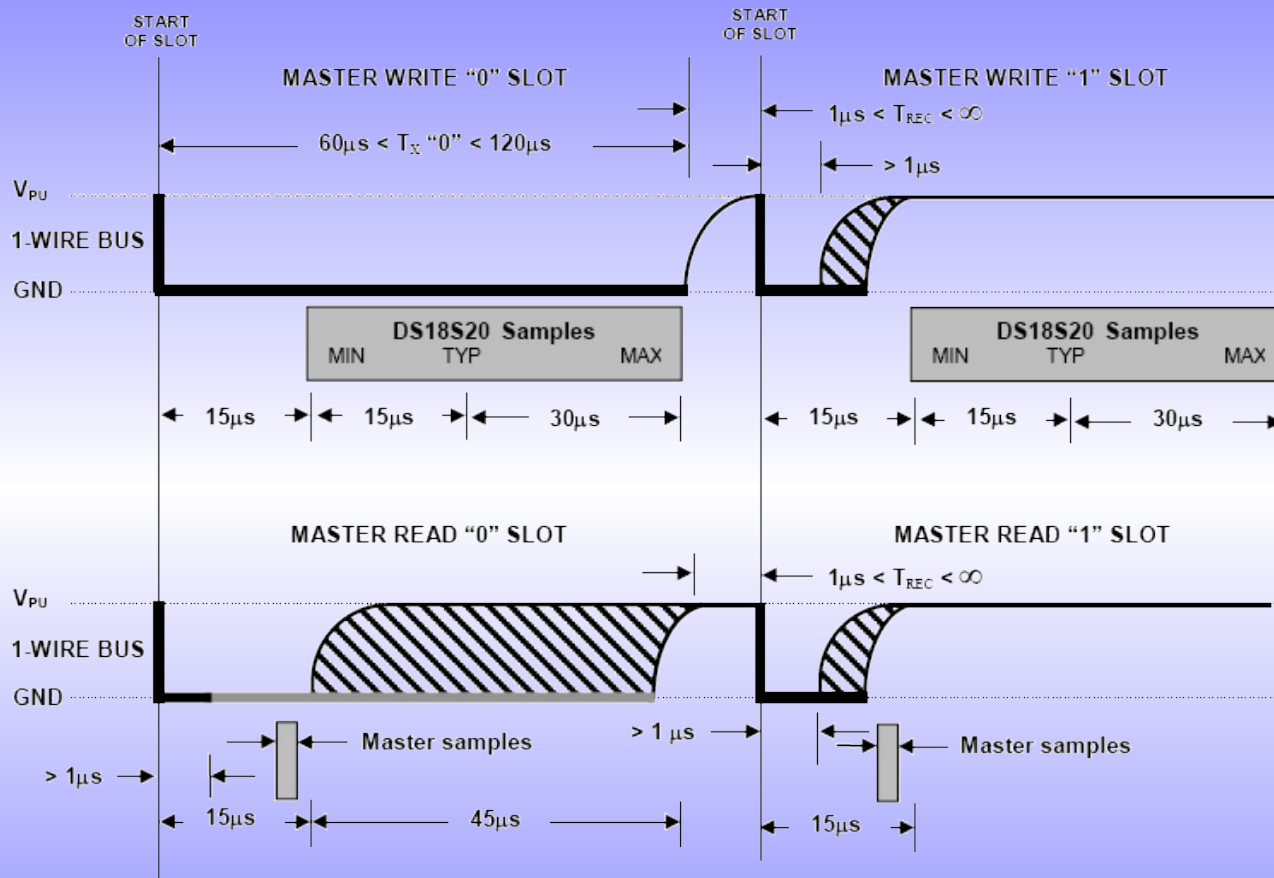
Um Daten zu lesen, muss der Master zuerst einen Impuls von min. 1  $\mu\text{s}$  generieren. Der Slave reagiert auf diese Signal entsprechend.




**Lesen einer 1:** Slave tut nichts

**Lesen einer 0:** Slave zieht die Datenleitung weiterhin auf *low*

# 1-Wire-Protokoll

READ/WRITE TIME SLOT TIMING DIAGRAM Figure 11



LINE TYPE LEGEND	
	Bus master pulling low
	Resistor pullup
	DS18S20 pulling low

Quelle:  
DS18S20 High-Precision 1-Wire  
Digital Thermometer





# Praktikumsversuch

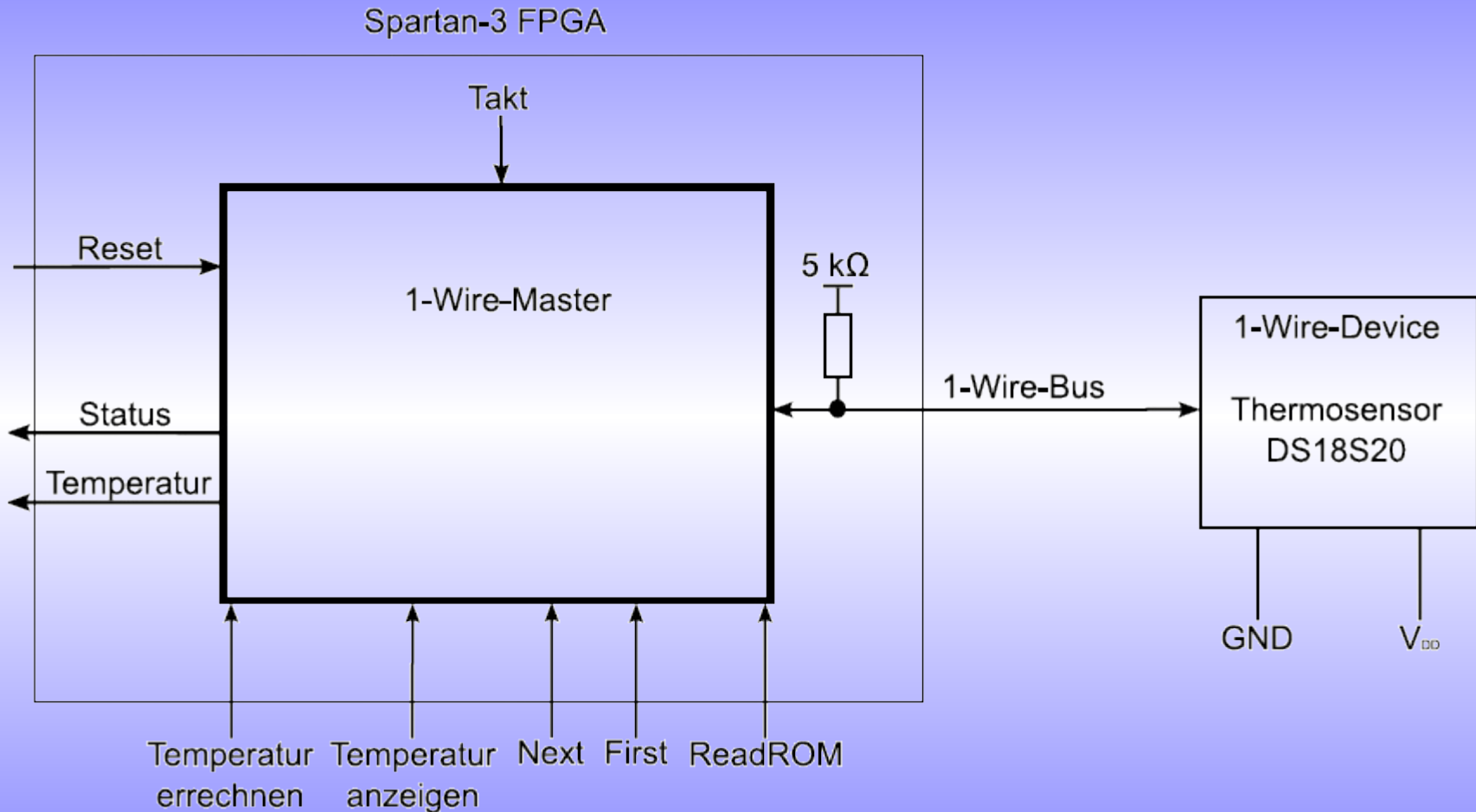
---

Ziel: einen 1-Wire-Master auf dem Spartan-3-FPGA implementieren, welcher alle Devices erkennt und mit ihnen kommunizieren kann

Funktionen:

- alle Devices identifizieren
- den 64-ROM-Code mittels CRC verifizieren
- Temperatur ermitteln und anzeigen
- Statusanzeige

# Praktikumsversuch





# Praktikumsversuch

---

Um mit den 1-Wire-Devices zu kommunizieren, legt der Master Befehle auf den Bus, die von den Devices erkannt werden und entsprechend reagieren.

## Benötigte Befehle

- **Read ROM:** Auslesen der 64-Bit-Adresse bei nur einem vorhandenem Device
- **Search ROM:** leitet einen universellen Suchalgorithmus ein, der es ermöglicht alle Devices auf dem Bus zu identifizieren
- **Match ROM:** adressieren eines Devices
- **Convert T:** veranlasst den Sensor die Temperatur zu messen
- **Read Scratchpad:** Auslesen der Temperatur

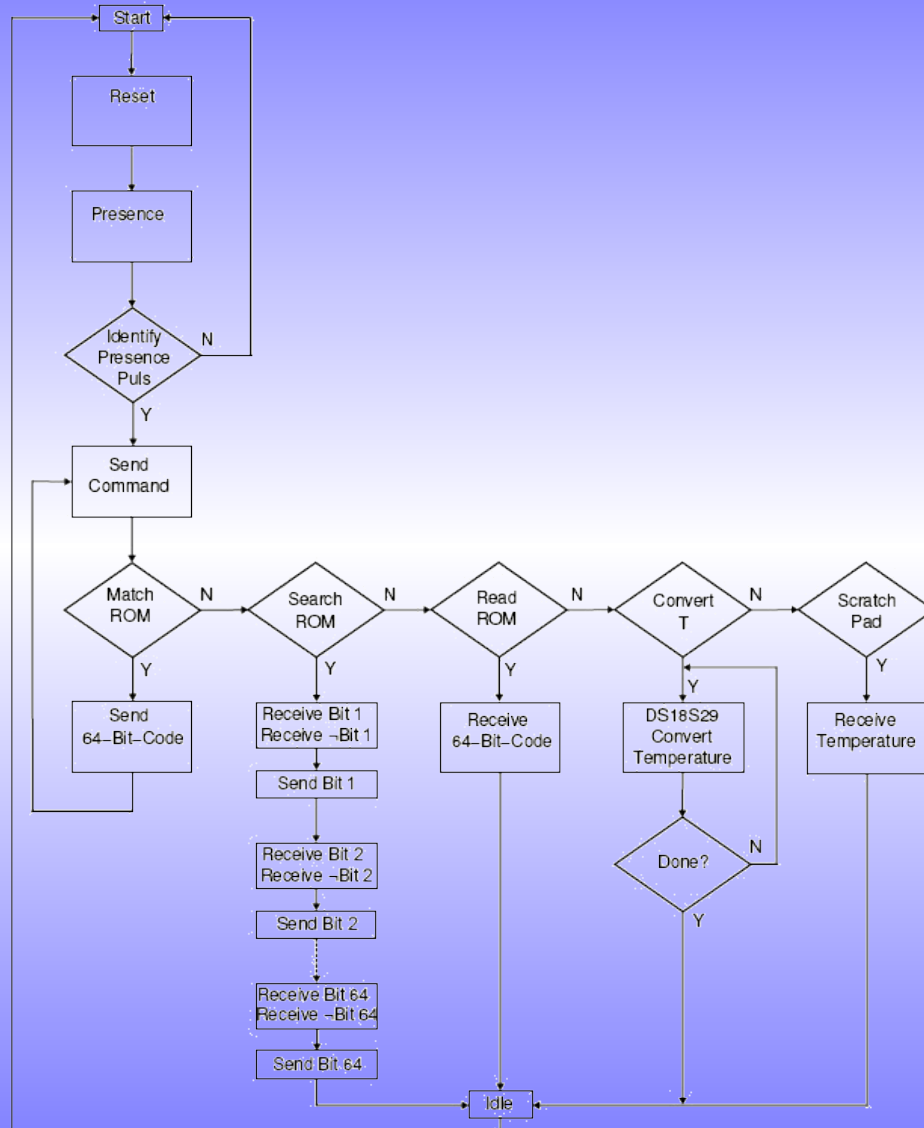
# Praktikumsversuch

## Suchalgorithmus

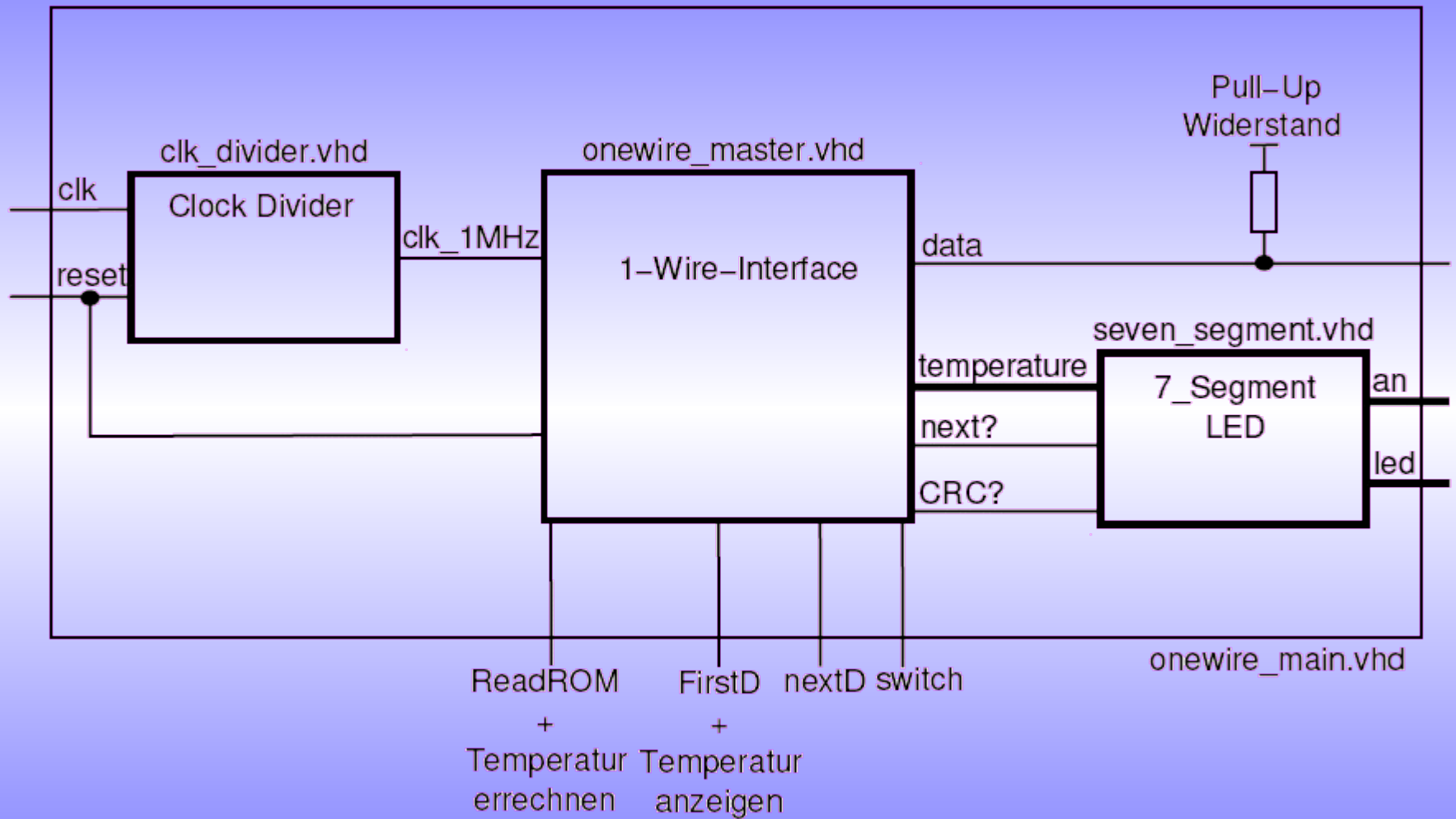
- auf dem Versuchsboard realisiert durch eine First- und Next-Taste
- der Algorithmus beginnt, indem der Master das erste Bit und dessen Komplement aller Devices verlangt
- der Bus verhält sich wie ein logisches AND
- abhängig vom gesendeten Bit, Komplement und eventuellen Ergebnis eines früheren Suchdurchlaufes, entscheidet der Master über die Suchrichtung
- der Master legt das errechnete Bit auf den Bus, alle Devices die nicht damit übereinstimmen gehen in den Wartemodus
- Wiederholung bis zum 64. Bit

Bit	Komplement	Suchrichtung	Bit-Position vs. letzte Diskrepanz	Suchrichtung
0	0	→	=	1
0	1	0	<	gleiche Richtung wie letzte Suche
1	0	1	>	0
1	1	Fehler		

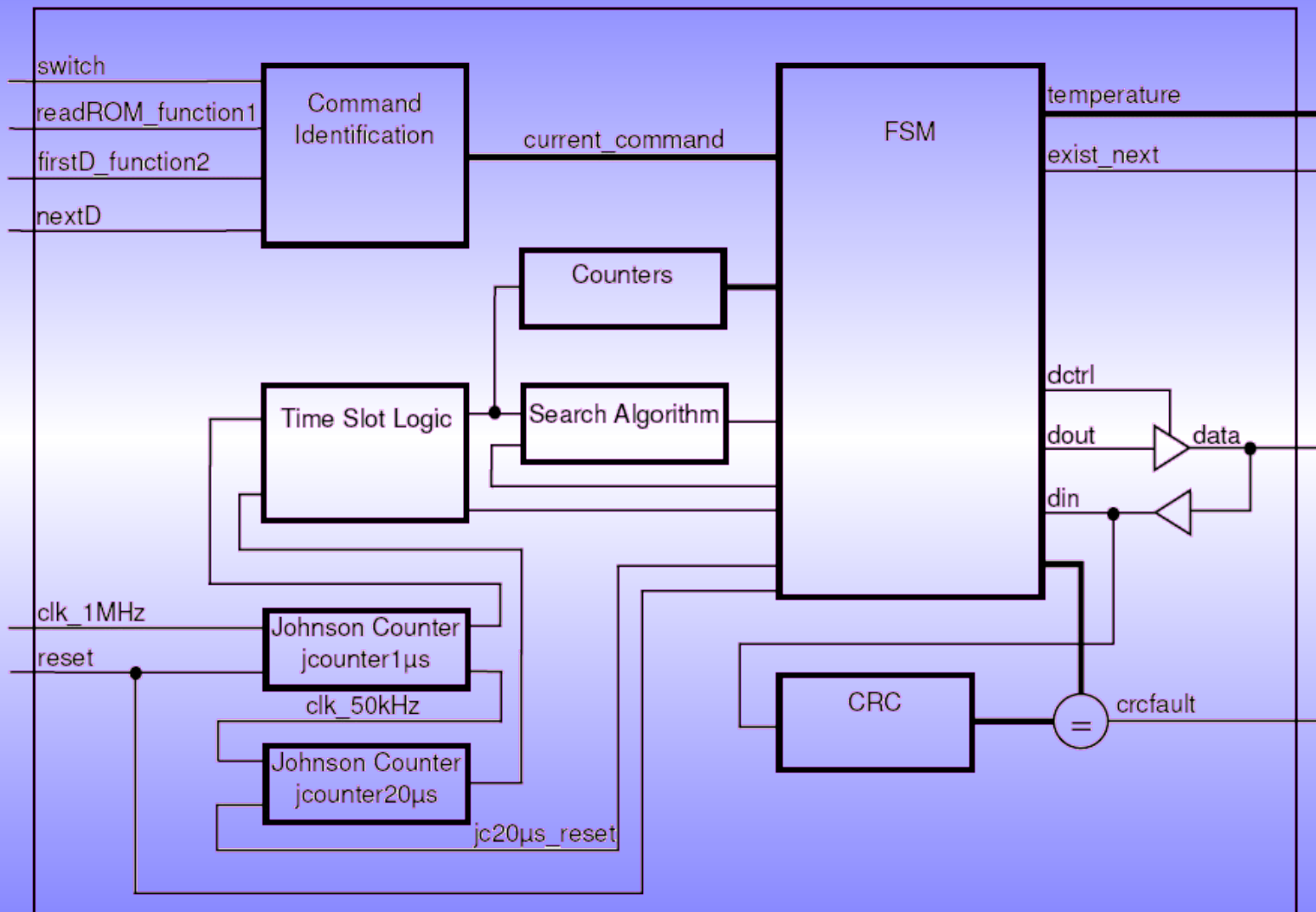
# Praktikumsversuch

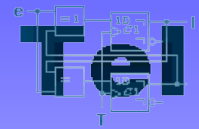


# Praktikumsversuch



# Praktikumsversuch





# Zusammenfassung

---

- es wird nur eine Datenleitung benötigt, an der üblicherweise Sensoren hängen
- Übertragung erfolgt in Time-Slots; durch unterschiedlich langes *low*-Ziehen werden Einsen oder Nullen erzeugt
- der Praktikumsversuch gibt einen guten Einblick in die 1-Wire-Technologie
- nicht alle Befehle, die der Sensor beherrscht, sind für das Praktikum notwendig
- komplexere Aufbauten realisieren