



# ASSEMBLER

ZEITBERECHNUNG EINES PROGRAMMS

# AGENDA

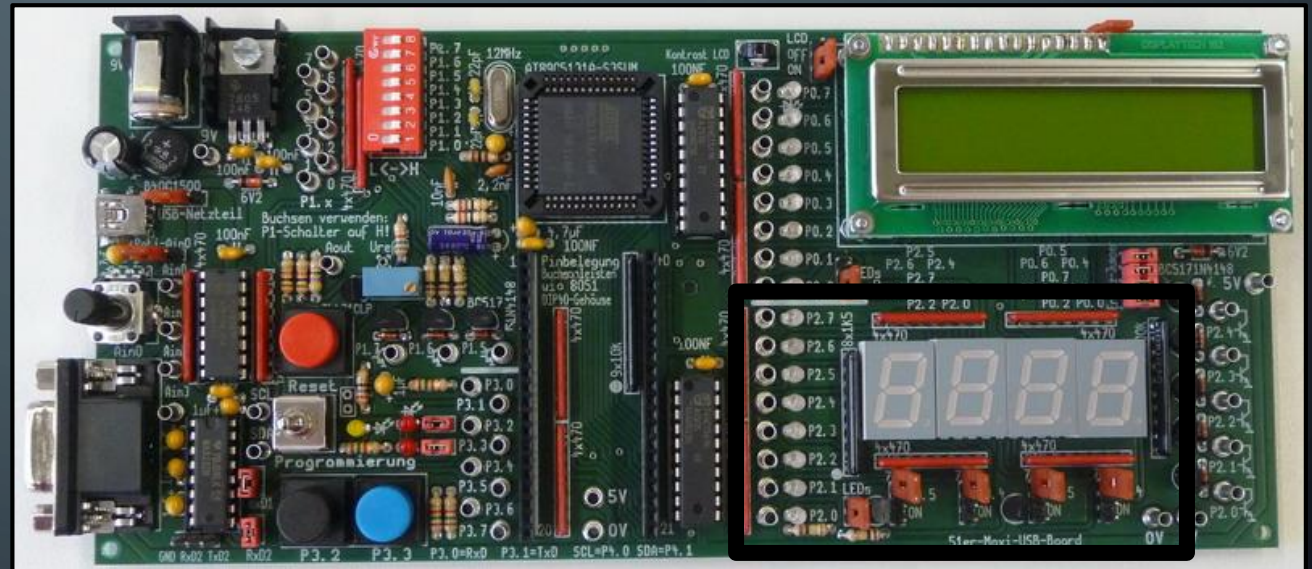
- Assembler
- Bsp. Programm Blinklicht
  - Berechnung der Dauer
- Alternatives Vorgehen

# ASSEMBLER

- Maschinennahe Sprache
  - Genaue Bestimmung des Speicherortes
  - Exakte Zeitmessung
- ⇒ Detailliertes Programmieren
- ⇒ Reduzieren des Speicherplatzes
- ⇒ Effizientere Befehle möglich

# ASSEMBLER

- Viele verwandte Sprachen
  - X86-Prozessoren:
    - Microsoft Macro Assembler MASM
  - Linux: GNU Assembler
  - Mikrocontroller:
    - Freier Macroassembler ASEM-51
  - Inline-Assembler
  - ...



# BSP. PROGRAMM BLINKLICHT

- LED soll mit Frequenz von 1 sek Blinken
- Lösung Pausenprogramm zählt bis 1.000.000

# ANALYSE

```
mov P2, #0
```

```
haupt:
```

```
    cpl P2.0  
    lcall Pause1s  
    sjmp haupt
```

```
Pause1s:
```

mov R2, #200	1 Mz	1 Mz
S2: mov R1, #100	1 Mz	200 Mz
S1: mov R0, #50	1 Mz	20.000 Mz
S0: djnz R0, S0	2 Mz	2.000.000 Mz
djnz R1, S1	2 Mz	40.000 Mz
djnz R2, S2	2 Mz	400 Mz
ret	2 Mz	2 Mz

```
End
```

---

2.060.603 Mz

# ANALYSE

mov P2, #0

haupt:

    cpl P2.0  
    lcall Pause1s  
    sjmp haupt

Pause1s:

mov R2, #100	1 Mz	1 Mz
S2: mov R1, #100	1 Mz	100 Mz
S1: mov R0, #50	1 Mz	10.000 Mz
S0: djnz R0, S0	2 Mz	1.000.000 Mz
djnz R1, S1	2 Mz	20.000 Mz
djnz R2, S2	2 Mz	200 Mz
ret	2 Mz	2 Mz

End

---

1.030.303 Mz

# ANALYSE

mov P2, #0

haupt:

cpl P2.0  
lcall Pause1s  
sjmp haupt

Pause1s:

mov R2, #100	1 Mz	1 Mz
S2: mov R1, #100	1 Mz	100 Mz
S1: mov R0, #48	1 Mz	10.000 Mz
S0: djnz R0, S0	2 Mz	960.000 Mz
djnz R1, S1	2 Mz	20.000 Mz
djnz R2, S2	2 Mz	200 Mz
ret	2 Mz	2 Mz

End

---

990.303 Mz



# ANALYSE

- Dauer eines Programms lässt sich auf Mz genau berechnen
  - Im Bsp. fehlen 9.697 Mz => neue Schleife...
- Im Allgemeinen reicht Multiplizieren der Startwerte \* 2
  - $2 * 100 * 100 * 50 = 1.000.000$

# ALTERNATIVE - TIMERINTERRUPT

**TMOD (89h) : Timermodus-Kontrollregister für Timer1 & Timer0**

Kontrolle Timer 1				Kontrolle Timer 0			
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Gate	C/T	M1	M0	Gate	C/T	M1	M0

Gate	C/T	M1	M0	
		0	0	Modus 0
		0	1	Modus 1: 16 Bit-Timer ohne Nachladen
		1	0	Modus 2: 8-Bit-Timer mit Auto-Reload
		1	1	Modus 3: 2 Stück 8-Bit-Timer
	0	Timer-Betrieb		
	1	Zähler-Betrieb		
0	Timer nur durch TR-Bit ein- und ausschalten			
1	Timer mit TR-Bit und Portpin ein- und ausschalten			