

1. Einleitung
2. Power Management
3. Parallele I/O
4. Serielle Schnittstelle
 1. I²C
 2. USART
5. Timer
6. Interrupt Handling
7. Signalverarbeitung
8. Regelungen
 1. Software Interrupt
9. Leitungscodes
10. Weiterführende Themen

Parallele IO - Einführung

- **Parallele IO ist die einfachste Art, um mit der Außenwelt zu kommunizieren, zu steuern und zu messen.**
- **Mikrocontroller-Chip besitzt Pins (elektrische Anschlüsse), die**
 - **als Ausgang benutzt werden können**
 - **als Eingang benutzt werden können**
- **Als Ausgang genutzt, kann z.B. eine LED eingeschalten werden. Als Eingang kann z.B. eine Schalterstellung/Taster abgefragt werden.**
- **Bei Parallel IO ist alles digital:**
 - Ein / Aus**
 - High / Low**
 - 1 / 0 (bei der Pegelinterpretation ist zu beachten, ob high oder low den aktiven Zustand als ‚1‘ darstellt)**

Beispiel Filterkaffeemaschine:

Einschalter (Eingang), Wasserheizung (Ausgang)

Parallele IO – Portpins und Ports

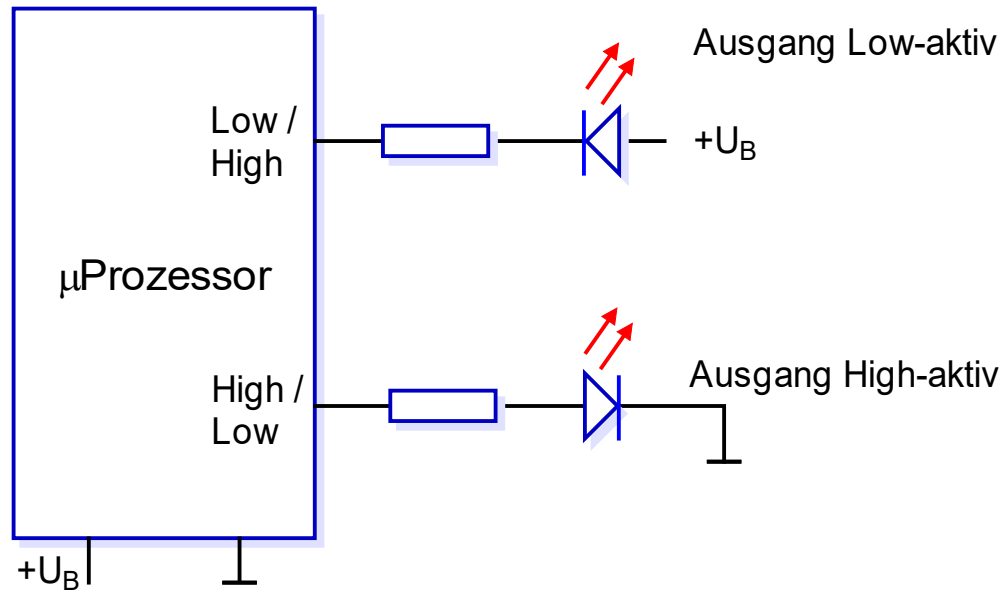
- **Der Mikrocontroller (μC) ist mit seiner Außenwelt über Portpins verbunden.**
- **Oft werden die einzelnen Portpins zusammengefasst in eine größere Einheit als Port zusammengefasst mit z.B. 32 unterscheidbaren Portpins.**
- **Oft findet man auch die Bezeichnung General Purpose Input Output (GPIO), die die Nutzbarkeit in beliebiger Form deutlich macht.**
- **Oft lässt sich die Funktion eines Portpins im weiteren konfigurieren, so dass dieser für eine Schnittstelle wie USART oder I²C eine Funktion übernimmt. Eine Schnittstelle benötigt meist mehr als einen Portpin.**
- **Analoge Werte sind ebenfalls über Portpins übertragbar, wenn analoge Peripheriekomponenten (AD / DA Wandler) als Funktion konfiguriert werden.**

Parallele IO – Portpins als Ressource

- Pins sind eine der knappsten Ressourcen, da die Anzahl der Anschlüsse an den Mikrocontroller begrenzt ist.
- Oft bietet der Mikrocontroller mehr Funktionen als Pins. Dazu kann eine Anwendung die Funktion eines Portpins konfigurieren. Dies nennt man Pin Multiplex (PinMux)
- Anwendung A: Viel Parallel IO, 1 Timer/Counter
- Anwendung B: Wenig Parallel IO, 3 Timer/Counter, 2 Serielle Schnittstellen
- Konfiguration von Parallel IO über Register
- Festlegen ob Ein- oder Ausgang
- Gegebenenfalls Pull-Up- oder Pull-down einschalten
- Gegebenenfalls Sonderfunktion / Pin Mux einstellen
- Definition der Register: Datenblatt

Parallele IO – LED

Ausgang Low-aktiv ist häufiger,
da dann der Strom für die Leuchtdiode nicht vom Prozessor zur
Verfügung gestellt werden muss



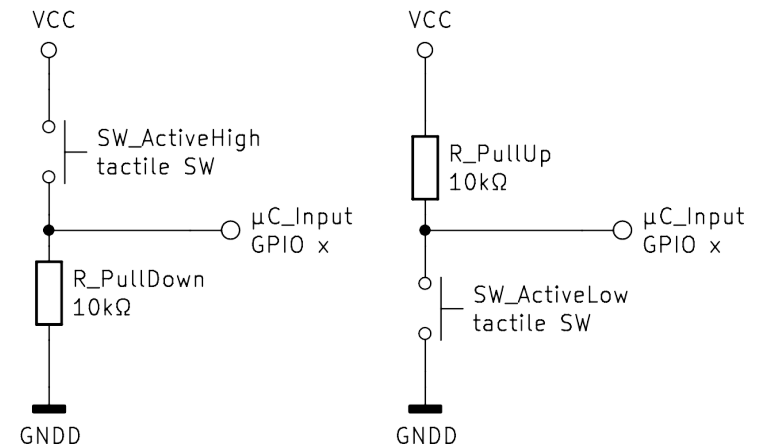
Parallele IO – Tasten und Schalter

Anschluss von Tastern / Schaltern

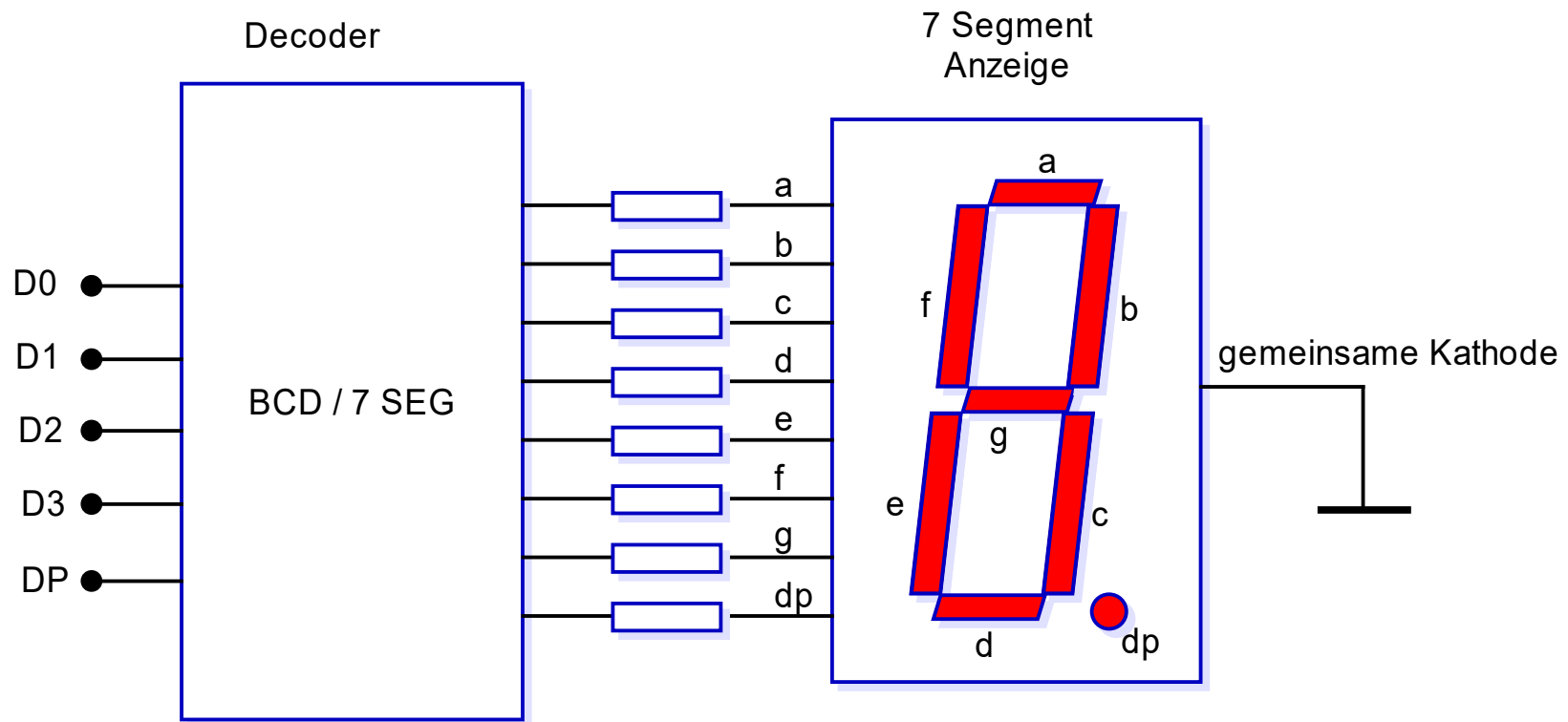
- Active high
- Active low

Bei mechanischen Tastern ist eigentlich immer ein Entprellen notwendig

- Durch Hardware
z.B. mit FlipFlop, Kondensator und Widerstand oder durch Optionen im Mikrocontroller selbst
- Durch Software
z.B. durch Zeitmessung mittels Timerfunktionen



Parallele IO – 7 Segment Anzeige



BCD = Binary Coded Decimal

Beispiel: 10 entspricht 0001 0000

Parallele IO – 7 Segment Anzeige mit Decoder

(xn) und 7 Ausgänge (a...g) für Segmente

Wert	x3	x2	x1	x0	Zeichen	a	b	c	d	e	f	g
0	0	0	0	0	0	1	1	1	1	1	1	0
1	0	0	0	1	1	0	1	1	0	0	0	0
2	0	0	1	0	2	1	1	0	1	1	0	1
3	0	0	1	1	3	1	1	1	1	0	0	1
4	0	1	0	0	4	0	1	1	0	0	1	1
5	0	1	0	1	5	1	0	1	1	0	1	1
6	0	1	1	0	6	1	0	1	1	1	1	1
7	0	1	1	1	7	1	1	1	0	0	0	0
8	1	0	0	0	8	1	1	1	1	1	1	1
9	1	0	0	1	9	1	1	1	1	0	1	1

Anschluß über Decoder spart
Leitungen / Pins

Mehrere Stellen über
Zeitmultiplex: Einzelne Stellen
werden nacheinander angeschaltet

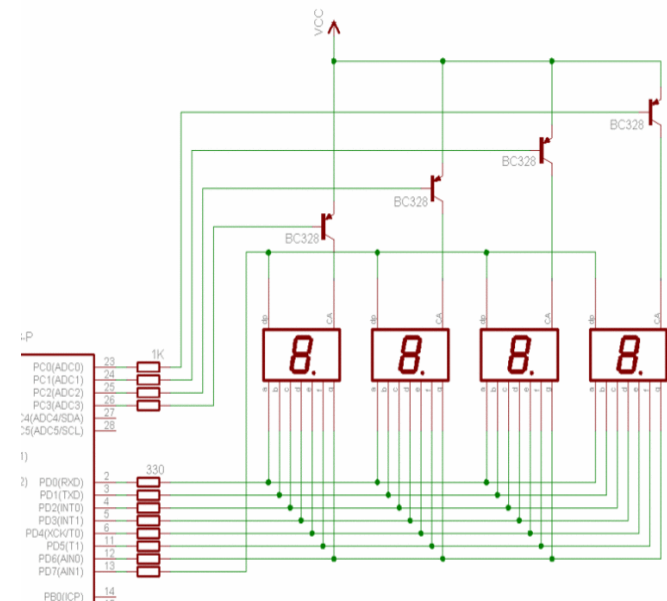
4 Anzeigen

Ohne Decoder:

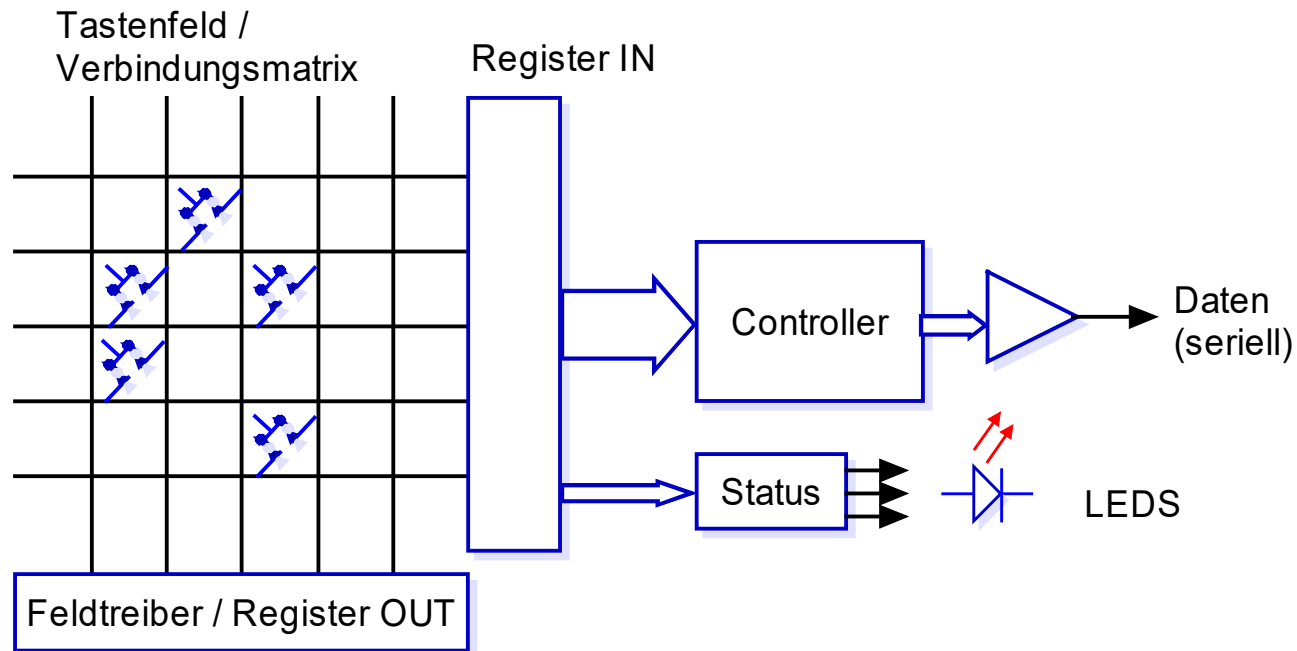
8 + 4 statt 8 x 4
12 statt 32

Mit Decoder:

5 + 4 statt 5 x 4
9 statt 20



Parallele IO – Tastatur



- Pins sparen durch Matrixanordnung
- Abfrage auch im Zeitmultiplex:
 - Spalten werden nach und nach eingeschaltet (Parallel Output)
 - Reihen werden gemessen (Parallel Input)
- Im Allgemeinen Pull-up/Pull-down Widerstände nötig, am besten im μC

Parallele IO – Laborhardware

Mikrocontroller	Fähigkeiten der Parallelen IO
AT91	• Zwei PIO-Ports mit jeweils 32 Portpins • User Interface an fester Speicheradresse • Zustandsänderungen pro Portpin durch Enable/Disable-Konzept • aufbauende Funktionen für Schnittstellen wie USART, Timer etc. konfigurierbar • Konfigurierbar zur Nutzung mit Interrupt Controller
SAM3X	• Vier PIO-Ports vergleichbar zu AT91 • Port Multiplexing für aufbauende Funktionen wie USART, Timer etc. • Etwas zusätzliche Funktionalität im Vergleich zu AT91
RP2040	• Programmable IO State Machines erlauben eines Ports flexible Verwendung unabhängig von der CPU • Digitale Schnittstellen wie UART oder I2C lassen sich per Software mit 9 speziellen Assemblerbefehlen mittels der State Machines bilden • GPIO mit SIO für direkten Zugriff • Zustandsänderungen pro Port • Zusätzliche Register für Set/Clear/Toggle