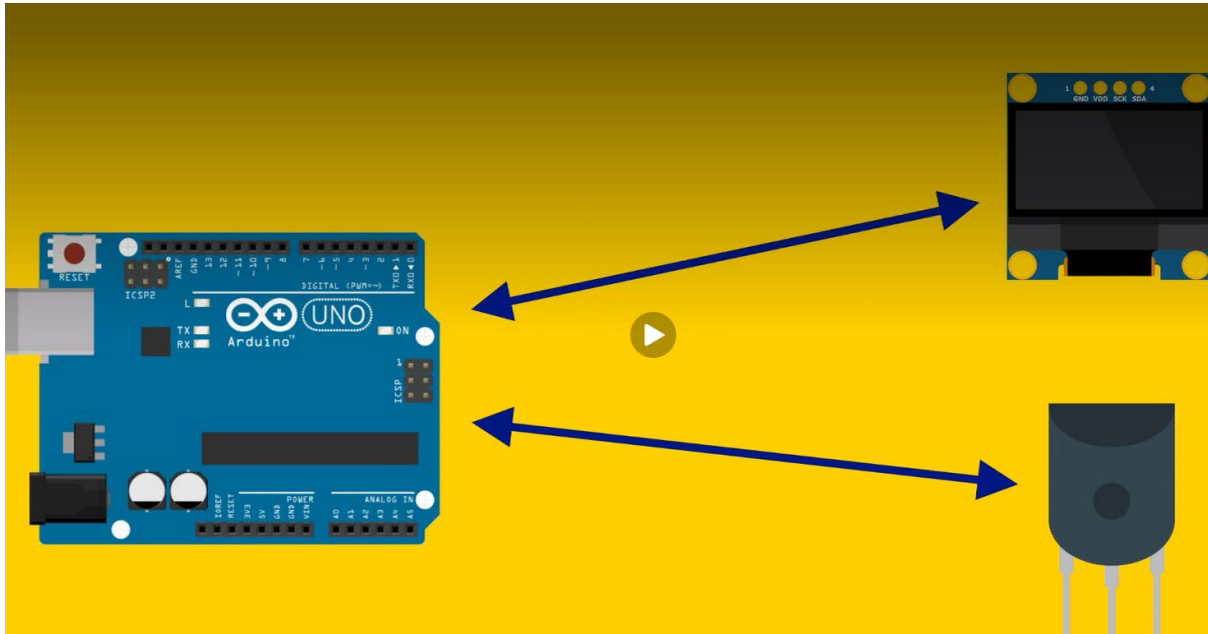


1. Einleitung
2. Power Management
3. Parallele I/O
4. **Serielle Schnittstelle**
 1. **I²C (synchron)**
 2. USART
5. Timer
6. Interrupt Handling
7. Signalverarbeitung
8. Regelungen
 1. Software Interrupt
9. Leitungscodes
10. Weiterführende Themen

I²C – Inter-Integrated-Circuit-Bus



➤ Grundlagen

- Serieller Datenbus zur Kommunikation zwischen Schaltungsteilen eines Geräts
- definiert wie ein μ C mit z.B. Displays und Sensoren kommuniziert.
- lediglich 2 Leitungen bis zu 127 Teilnehmern kommuniziert werden kann.
- I2C-Bus definiert den Ablauf wie diese Kommunikation von statten geht.

➤ Entwicklung der Schnittstelle

- Serieller Datenbus zur Kommunikation zwischen Schaltungsteilen eines Geräts
- 1982 eingeführt von Philips zur Kommunikation zwischen ICs
- 1992 Spezifikation V1 mit Geschwindigkeiten von bis zu 400 Kbit/s (Fast Mode)
- Weiterentwicklung bis 2012 zur Spezifikation V5 mit Ultra-Fast-Mode bis zu 5 Mbit/s
- **Atmel** führt die Schnittstelle unter der Bezeichnung **Two-Wire-Interface (TWI)**

➤ 2-Draht-Übertragung

- Signalleitung 1: **Serial CLock (SCL)** bzw. Two-Wire Clock (TWCK)
- Signalleitung 2: **Serial DAta (SDA)** bzw. Two-Wire Data (TWD)
- Positive Logik: „High“ entspricht einer „1“ und „Low“ entspricht einer „0“
- Bidirektionales Halbduplex-Übertragungsverfahren

➤ Mehrere Teilnehmer möglich (Master-Slave-Bus)

- Ein Datentransfer wird durch einen Master (Controller) initiiert
- Slaves (Targets) werden über eine 7-Bit-Adresse angesprochen, bis zu 127 Slaves möglich

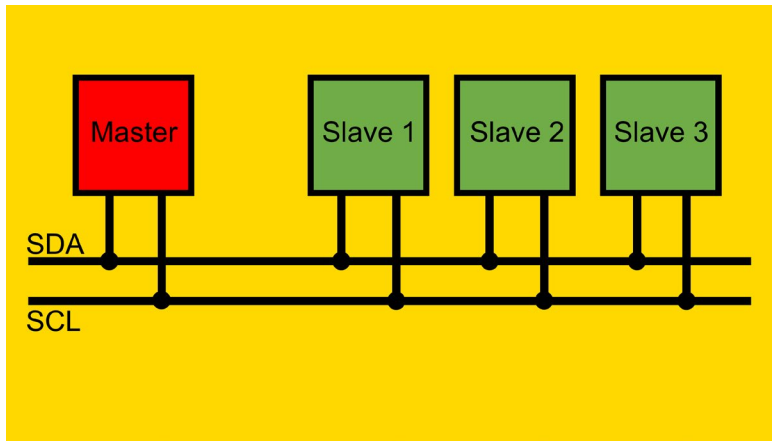
I²C – Inter-Integrated-Circuit-Bus

➤ Eigenschaften

- Übertragungsrate: 0,1 Mbit/s (Standard)
- Takt: synchron durch gemeinsame Taktleitung **SCL** (Serial Clock)
- Datenübertragung seriell, Richtung bidirektional

➤ Aufbau

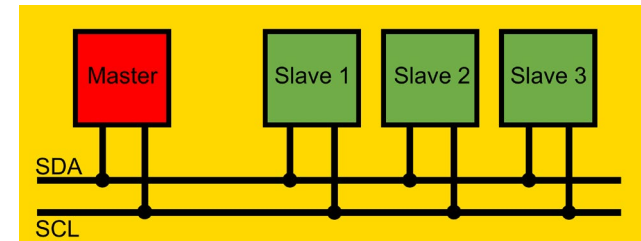
- Der I²C besteht i.d.R. aus mehreren Teilnehmern die zusammen kommunizieren.



- Ein Datentransfer wird durch einen Master (Controller) initiiert
- Slaves (Targets) werden über eine 7-Bit-Adresse angesprochen, bis zu 127 Slaves möglich
- Die Kommunikation findet immer zwischen Master und Slaves statt

➤ Leitungen:

- SDA (Serial Data): sendet (bidirektional) Daten
- SCL (Serial Clock): sendet Taktimpulse
- Teilnehmer verwenden einen gemeinsamen Bustakt deshalb handelt es sich hierbei um einen sogenannten **synchronen** Bus



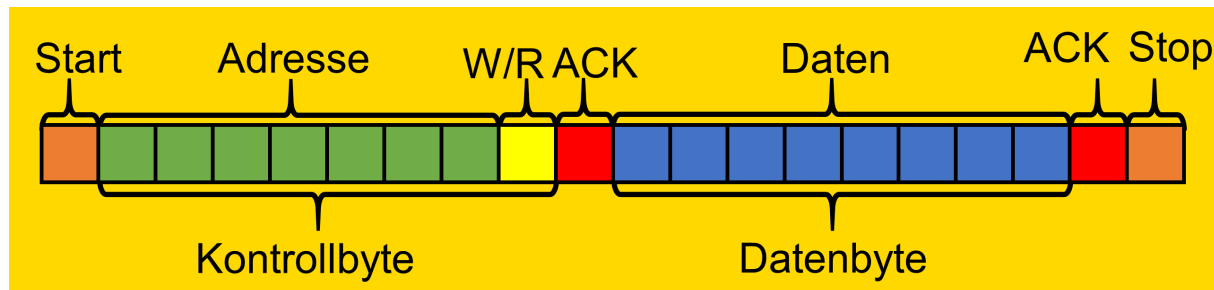
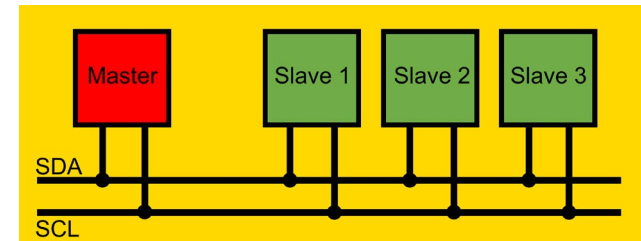
➤ Beispiel: Ansteuerung einer LED Matrix Anzeige (3. Praktikum)

- Master (Mikrocontroller), Slave (8x8 LED Matrix)
- Der Master bestimmt, wer auf der Leitung kommuniziert, da nur eine Leitung (SDA)
- Der Master kann entweder Daten an Slaves senden oder eben Slaves auffordern ihm Daten zu schicken.
- Ein Slave kann aber nie von sich aus Daten schicken. Nur, wenn er vom Master dazu aufgefordert wird.

I²C – Inter-Integrated-Circuit-Bus

➤ Ablauf:

- Daten werden seriell (hintereinander) geschickt
- Protokoll definiert die Übertragung



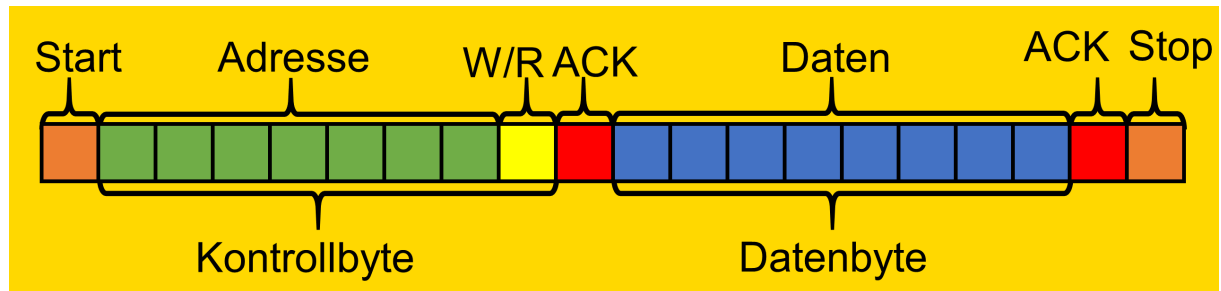
➤ Start-Condition

Als erstes sendet der Master eine sogenannte „Start-Condition“. Dadurch werden alle Slaves hellhörig und hören genau zu was auf der Datenleitung folgend gesendet wird.

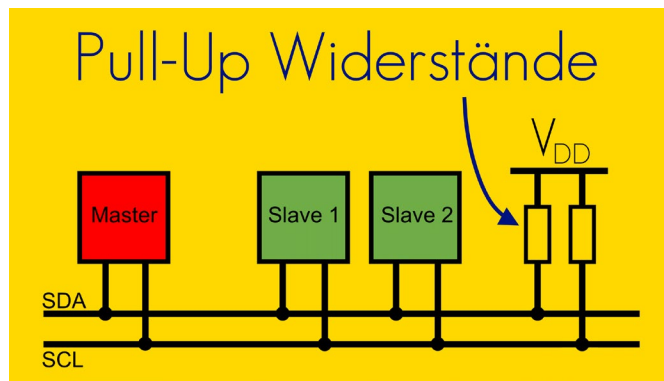
- **Adresse:** 7 Bit Adresse des Slaves, mit dem der Master kommunizieren möchte
- **Write (0) / Read (1):** Write: Daten zum Display senden, Read: Temperatur einlesen
- **Acknowledge:** der Slave sendet ein sogenanntes Acknowledge-Bit (0), also eine Bestätigung, das er bereit für die Kommunikation ist.
- **Datenpaket:** 8 Bit, also 1 Byte. Ist es gesendet, folgt ein Acknowledge-Bit und darauf kommt ggfs. das nächste Datenpaket. Es können beliebig viele Datenpakete versendet werden.

I²C – Inter-Integrated-Circuit-Bus

➤ Ablauf:

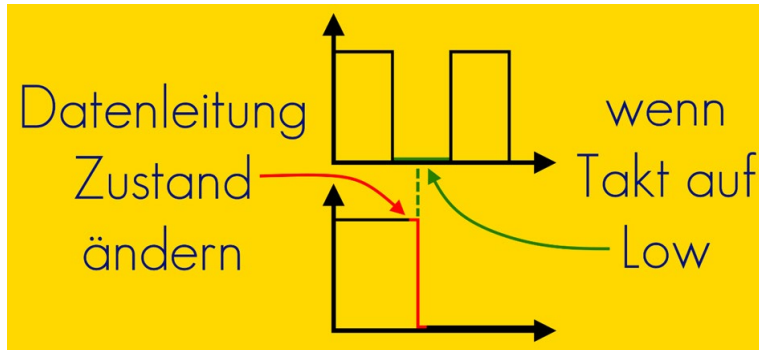


- **Stop-Condition:** Die Datenleitung geht auf High, während wie Clockleitung (SCL) auch auf High ist.
- **Pull-Up Widerstände:**
- Wenn auf dem I²C Bus nichts kommuniziert wird, sind die beiden Busleitungen SDA und SCL beide auf High (idle high).
- Um dieses Verhalten zu erreichen, wird an beide Leitungen ein Pull-Up Widerstand verbaut.

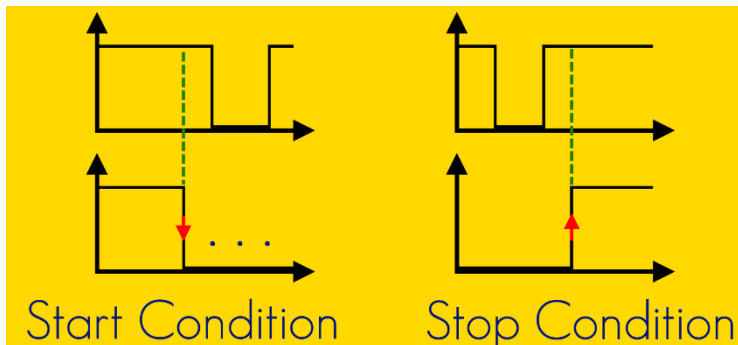


➤ Details:

- **Zustandsänderung:** Die Datenleitung ändert den Zustand, wenn der Takt auf Low ist

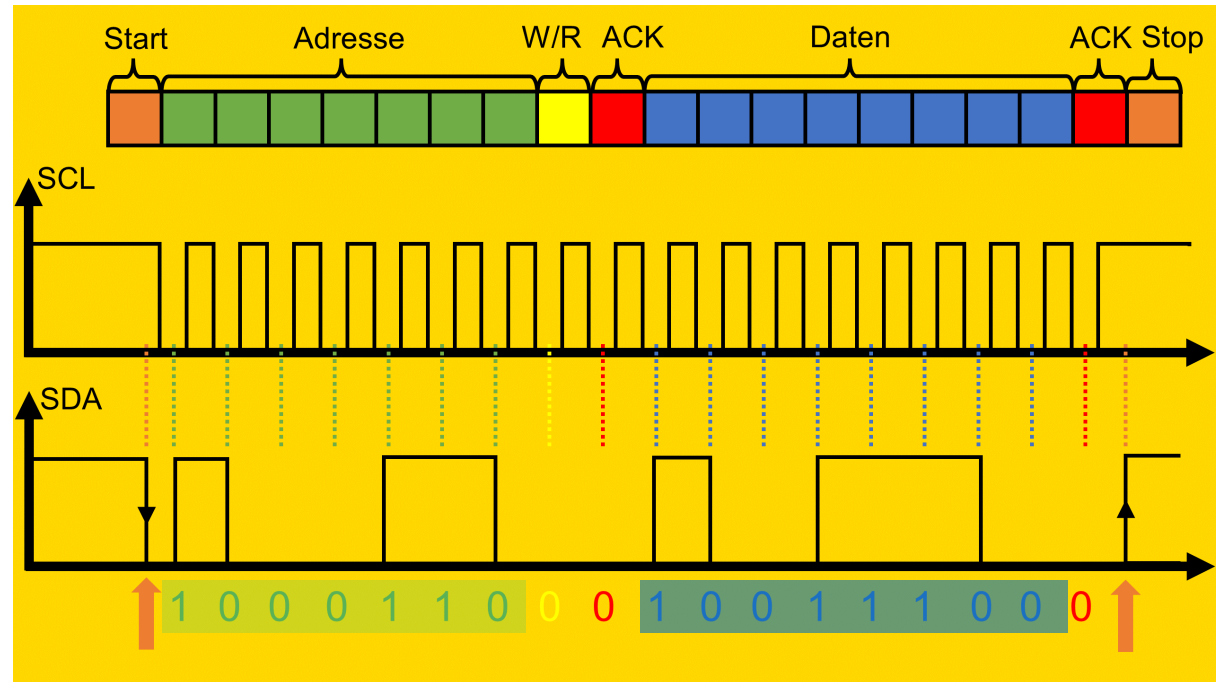


- **Ausnahme Start-Stop-Condition:** Die Datenleitung ändert den Zustand, wenn der Takt auf High ist



I²C – Inter-Integrated-Circuit-Bus

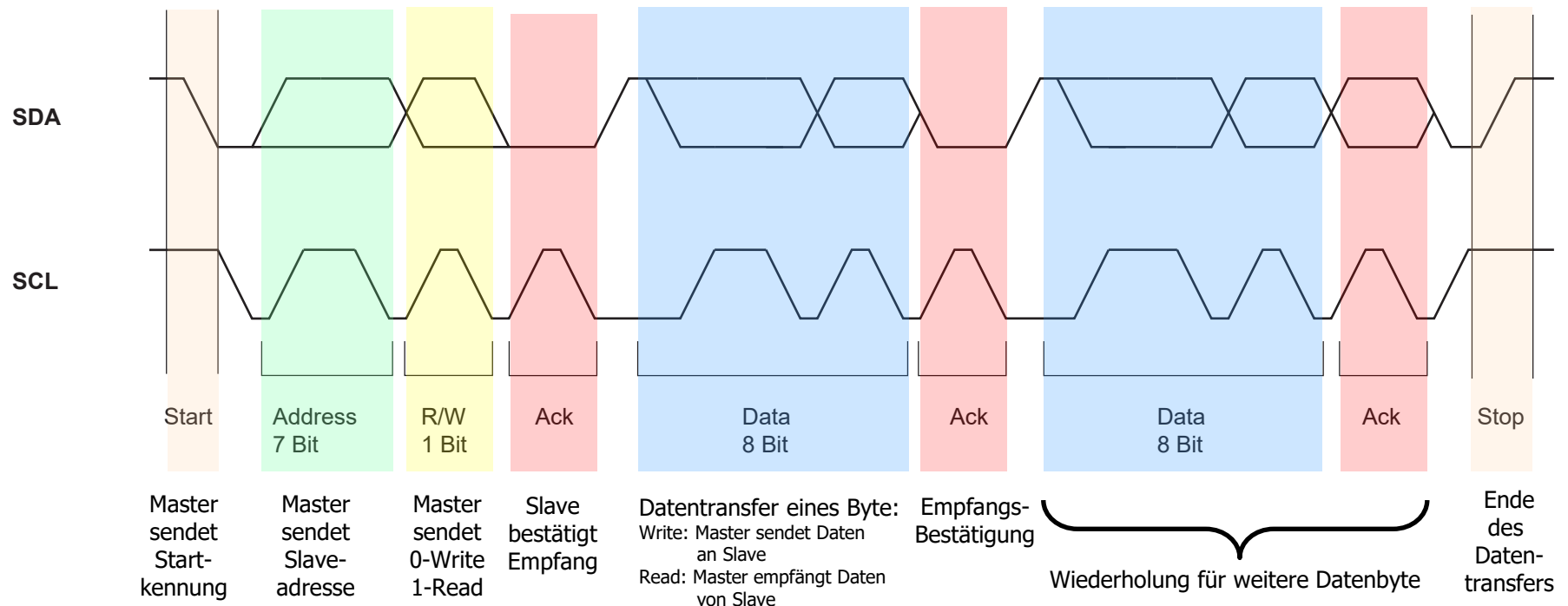
➤ Zeitdiagramm:



- **Start:** Dateneingang auf Low, während die Taktleitung auf High ist
- **Adresse:** 7 Bit, , R/W Bit: Read (0) Write (1)
- **Acknowledge:** Bestätigung Low (0)
- **Datenpaket:** 1 Byte
- **Acknowledge:** Bestätigung Low (0), ggfs. Wiederholung
- **Stop:** Die Datenleitung geht auf High, während CLK auf High ist.

I²C – Übertragungsformat auf dem Bus

- Die auf der SDA-Leitung übertragenen Daten sind jeweils 8 Bit lang.
- Die Daten werden mit dem höchstwertigen Bit (MSB) zuerst übertragen.
- Jedes übertragene Byte muss durch den Empfänger bestätigt werden (Ack).
- Jede Übertragung beginnt mit einer Start-Bedingung und endet mit einer Stopp-Bedingung.



- **Der TWI-Bus (Two-Wire Interface, kompatibel zu I²C) ist synchron**
 - Der Takt (TWCK bzw. SCL) wird immer vom Master erzeugt, sobald eine Übertragung stattfindet.
 - Solange keine Kommunikation aktiv ist, bleiben SCL und SDA beide auf High (durch Pull-up-Widerstände).
 - Erst wenn der Master sendet oder empfängt, beginnt er den Takt auf TWCK (bzw. SCL) zu toggeln.
 - Die Startsequenz (SDA fällt auf Low, während SCL High ist) und die Stopsequenz (SDA steigt auf High, während SCL High ist) werden ebenfalls vom Master erzeugt.
 - TWCK führt nur während der aktiven Datenübertragung Taktimpulse.
 - In der Bus-Leerlaufzeit bleibt TWCK konstant auf High.
 - Der Bus ist „multi-master-fähig“, aber stets synchron getaktet.

➤ 1. Bus-Leerlauf

- Solange keine Übertragung läuft, liegen SCL und SDA beide auf High (Pull-ups).
- Alle Master überwachen den Bus.
- Wer etwas senden will, prüft:→ Sind SDA und SCL = High, ist der Bus frei (idle).
- Dann darf dieser Master eine Startbedingung erzeugen (SDA fällt bei SCL=High).

➤ 2. Takt-Erzeugung

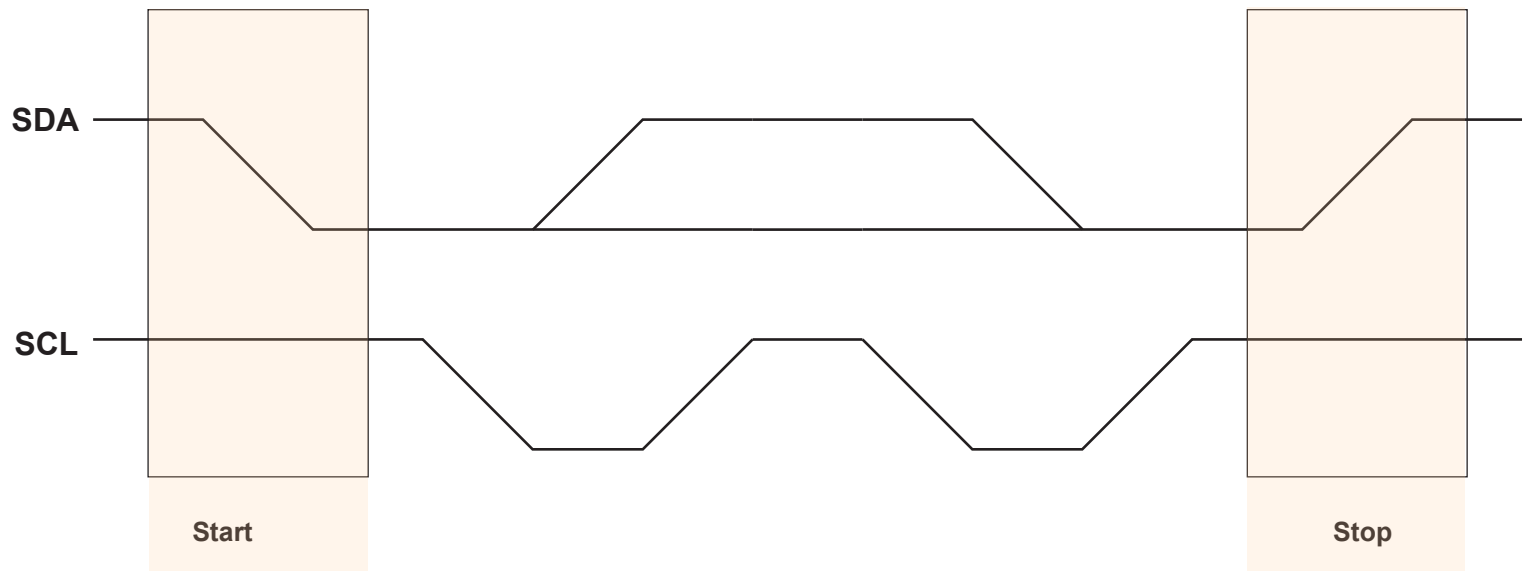
- Nur der aktive Master erzeugt den Takt auf SCL (TWCK).
- Der Master bestimmt die Taktfrequenz (z. B. 100 kHz oder 400 kHz) für diese eine Übertragung.
- Andere Master warten, bis der Bus wieder frei ist.
- Wenn mehrere Master gleichzeitig eine Startsequenz versuchen, greift die Arbitration (Bus-Arbitrierung, Bus-Zugriffsregelung).

➤ 3. Arbitrierung (Bus-Zugriffsregelung)

- Falls zwei Master gleichzeitig starten:
- Beide beobachten SDA während des Sendens.
- Wenn ein Master „1“ sendet, aber eine „0“ liest → hat er den Bus verloren (weil ein anderer Master SDA auf Low zieht).
- Der Master, der die Arbitrierung verliert, schaltet sich sofort ab und wartet, bis der Bus wieder frei ist.
- Damit funktioniert das Multi-Master-Prinzip ohne zentralen Taktgeber und ohne Konflikte.
- Die Taktfrequenz ist also nicht global festgelegt, sondern wird pro Übertragung vom aktiven Master bestimmt.

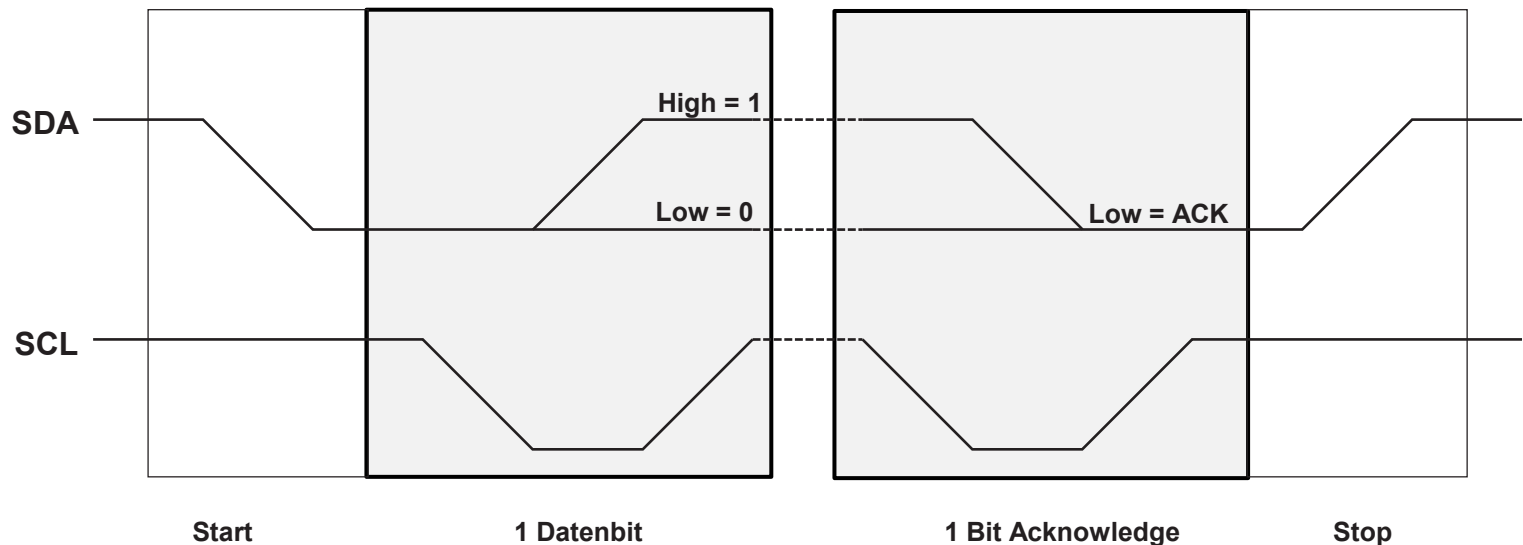
I²C – Leitungscodierung – Start- und Stopp-Bedingung

- **Startbedingung (Master fordert den Bus an):**
 1. Ruhezustand SCL-Leitung auf High; SDA auf High
 2. Master legt SDA auf Low:
eine fallende Flanke auf der SDA-Leitung, während SCL High ist.
 3. Master legt SCL auf Low: eine fallende Flanke auf der SCL-Leitung
- **Stoppbedingung (Master gibt den Bus frei):**
 1. Abschluss der letzten Übertragung
 2. Master belässt SCL auf High
 3. Master legt SDA auf High: Eine steigende Flanke auf der SDA-Leitung, während SCL High ist.



I²C – Leitungscodierung – Daten und Acknowledge

- Die Pegelübergänge auf der SDA-Leitung bei der Start- und Stoppbedingung erfolgen, wenn SCL auf High-Pegel ist.
- Alle weiteren Pegelübergänge zur Übertragung von Daten auf der SDA-Leitung erfolgen nur, wenn SCL auf Low-Pegel ist.
- **Empfangsbestätigung mit Acknowledge**
 - Der Empfänger von Daten bestätigt den Empfang mit Acknowledge.
 - Low (0) = Acknowledge (ACK): erfordert aktives Reagieren des Empfängers
High (1) = Negative ACK (NACK): Signalpegel, wenn Reaktion des Empfängers ausbleibt, Master sendet Stop-Condition, um Bus freizugeben (Bus ist „idle high“)



I²C – Modi der Busgeschwindigkeit

I ² C Modes	Geschwindigkeit
Standard Mode	100 KBit/s
Fast Mode	400 KBit/s
Fast Mode Plus	1 MBit/s
High Speed Mode	3,4 MBit/s
Ultra Fast Mode	5 MBit/s

- **Einstellung der Geschwindigkeit**
 - Master gibt die Taktrate bei den Modi bis 1 Mbit/s direkt vor
- **High Speed Mode**
 - Rückwärtskompatibel zu Geräten mit langsameren Geschwindigkeiten
 - Umschalten in die hohe Übertragungsrate durch Bytesequenz
- **Ultra Fast Mode**
 - Unidirektional, unterstützt nur Schreiben
 - Weitere Modifikationen am Protokoll

I²C – Vergleich der Laborhardware

Mikrocontroller	Fähigkeiten der Parallelen IO
AT91	• Keine Unterstützung von I²C
SAM3X	• Zwei Two-Wire-Interface (TWI)-Schnittstellen (TWI0 und TWI1) • Atmel als Hersteller verwendet aus namensrechtlichen Gründen den Begriff TWI statt I²C. Funktional gibt es keine Unterschiede. • Standard- und Fast Mode • 7 oder 10 Bit Slave-Adressen • Konfigurierbar zur Nutzung mit Interrupt Controller
RP2040	• Zwei I²C-Schnittstellen • Standard-, Fast und Fast Mode Plus • 7 oder 10 Bit Slave-Adressen • Konfigurierbar zur Nutzung mit Interrupt Controller