

## Das I2C Protokoll

Eine Kommunikation startet der Master mit dem Senden eines Start-bits und mit der Adresse vom Teilnehmer mit der er die Kommunikation aufnehmen möchte, gefolgt vom RW (read/write) bit, dass signalisiert ob der Master Daten zum Slave schreiben oder vom Slave lesen möchte. Alle Slaves im Bus werden nach dem Start-bit hellhörig und vergleichen ihre Adresse mit der Adresse die vom Master gesendet wurde. Der Teilnehmer mit der richtigen Adresse sendet dem Master ein Acknowledge (akzeptieren) Bit zurück um zu signalisieren, dass er bereit ist mit dem Datenaustausch. Danach sendet der Master die Adresse für das interne Register vom Slave um darin Daten lesen oder schreiben zu können. Der Slave bestätigt das erneut und jetzt können Daten mit dem Master ausgetauscht werden bis der Master ein Stop-bit sendet und der Bus wieder freigegeben wird.



### Das Start und Stop-bit

**Start-bit:** Die Datenleitung wird von HIGH auf LOW gesetzt, noch bevor das Clock Signal von HIGH auf LOW geht.

**Stop-bit:** Genau umgekehrt wird das Stopp-bit erzeugt, die Datenleitung geht von LOW auf HIGH, nachdem das Clock Signal von LOW auf HIGH gesetzt wird.

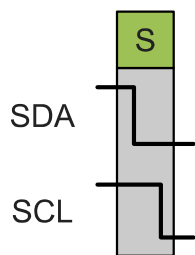


Bild 4.3.6: Start-bit

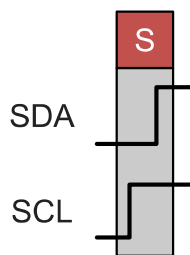


Bild 4.3.7: Stop-bit

### Adressierung (Ax) und das Read/Write bit

Nach dem Start-bit sendet der Master die Adresse vom Slave, mit dem er Daten austauschen möchte. Die Adresse kann 7 (Standard) oder 10 bit gross sein. In unserem Beispiel verwenden wir eine 7 bit Adresse. Als achttes bit (RW) sendet der Master, ob er den Slave beschreiben will (0 - write) oder ob er vom Slave Daten lesen möchte (1 - read). Beim Beschreiben vom Slave wird RW (read/write) auf 0 gesetzt und beim Daten lesen vom Slave auf 1.

Als Beispiel in diesem 1 Byte (8 bit) grossen Block möchte der Master auf dem Slave mit der Adresse 0x59 Daten schreiben und sendet dafür folgende Daten (10110010), mit MSB (most significant bit) first, also vom Datensatz links gestartet bei bit 6.

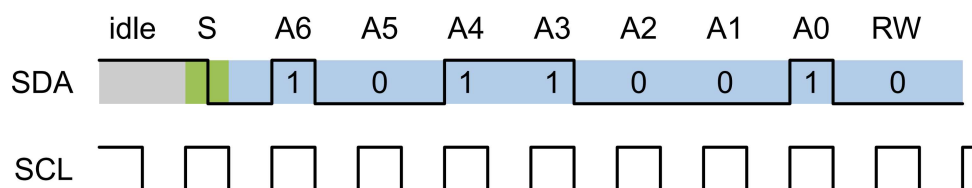


Bild 4.3.8: Start-bit und Adressierung über I2C

## Acknowledge (ACK) und Not Acknowledge (NACK)

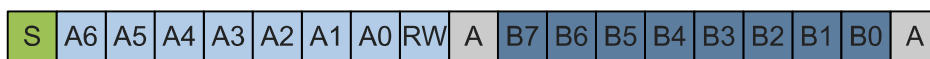
Nach jedem Byte Daten wird vom Empfänger ein Bestätigung-bit gesendet. Wenn das Bit LOW ist (ACK), dann signalisiert das dem Master, dass der Slave die Adresse oder die Daten empfangen hat und alles in Ordnung ist.

Sollte ein Problem auftreten wie zum Beispiel, dass der Slave die Daten nicht lesen kann oder gerade mit was anderem Beschäftigt ist und keine Zeit hat mit dem Master zu kommunizieren, dann sendet der Slave ein HIGH Signal (NACK)

## Adressierung vom internen Register (Bx)

Ein Gerät wie z. B. der BME280 Sensor hat eine Adresse, aber er hat verschiedene Register in denen die Werte wie Temperatur, Luftdruck und die Luftfeuchtigkeit gespeichert sind. Diese Register können nicht nur gelesen werden, man kann auch Register beschreiben wie z. B. das 0xF5 „config“ Register um verschiedene Optionen zu setzen.

Nachdem Daten Byte für die Register Adressierung wird vom Slave wieder ein ACK bit gesendet.



Start   Geräte (Slave) Adresse   Ack.   Int. Register Adresse   Ack.

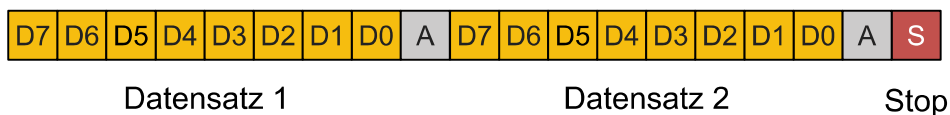
## Die Datenübertragung (Dx)

Nachdem der Master das ACK bit von der Int. Register Adresse vom Slave erhalten hat, kann die eigentliche Übermittlung der Daten erfolgen. Ein Block von Daten ist immer 1 Byte gross, also 8 bit und startet mit dem Most Significant Bit (MSB) zuerst. Zum besseren Verständnis machen wir ein Beispiel.

Wenn wir als Datenpaket den HEX Wert 0xD0 senden wollen, dann ergibt das als 8 bit Binary Wert 11010000. Sollen die Daten mit MSB first gesendet werden, dann starten wir mit dem linken Wert zuerst, also mit der 1 gefolgt von der zweiten 1 und so weiter bis wir alle 8 bits gesendet haben.

Es gibt auch das Least Significant Bit (LSB), hier starten wir auf der rechten Seite mit den Daten. Zuerst die 0 gefolgt von der zweiten und dritten 0 und so weiter.

Nach jedem Byte, das gesendet wird, bestätigt der Slave mit einem ACK bit das er die Daten erhalten hat. Die Daten können jetzt Byte für Byte gesendet werden, bis der Master ein Stop-bit sendet und damit die Kommunikation beendet.



Least significant bit

1010011

Most significant bit

1010011

## Daten lesen vom Slave

Das Daten lesen von einem Slave funktioniert fast gleich wie das Beschreiben von einem Slave. Der Master sendet die Geräte Adresse, gefolgt von einer 1 bei RW (lesen). Nachdem der Master das ACK bit vom Slave erhalten hat, sendet er die Register Adresse aus der er Daten lesen möchte.



Start   Geräte (Slave) Adresse   Ack.   Int. Register Adresse   Ack.

Die Daten werden vom Slave zum Master gesendet, das Clock Signal wird aber dennoch vom Master bereitgestellt. Nach jedem empfangenen Byte sendet der Master das ACK bit zum Slave und der sendet so lange weiter bis der Master die Übertragung mit einem NACK-bit gefolgt von einem Stop-bit beendet.



Datensatz 1

Datensatz 2

Stop