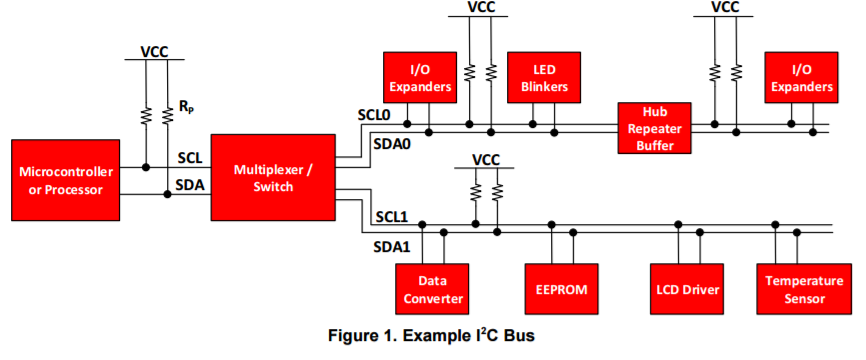
I2C

Overview:

The I2C bus is a very popular and powerful bus used for communication between a master (or multiple masters) and a single or multiple slave devices.

It only uses 2 wires: SDA (Serial Data) and SCL (Serial Clock).



A slave may not transmit data unless it has been addressed by the master. Data transfer may be initiated only when the bus is idle. A bus is considered idle if both SDA and SCL lines are high after a STOP condition.

The general procedure for a master to access a slave device is the following:

1. Suppose a master wants to send data to a slave:

• Master-transmitter sends a START condition and addresses the slave-receiver

• Master-transmitter sends data to slave-receiver

• Master-transmitter terminates the transfer with a STOP condition

1. If a master wants to receive/read data from a slave:

• Master-receiver sends a START condition and addresses the slave-transmitter

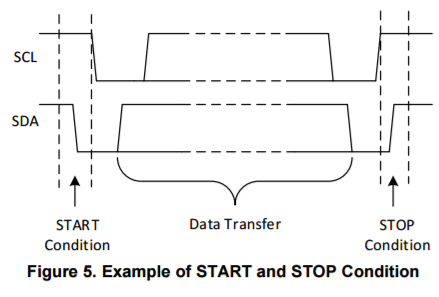
• Master-receiver sends the requested register to read to slave-transmitter

• Master-receiver receives data from the slave-transmitter

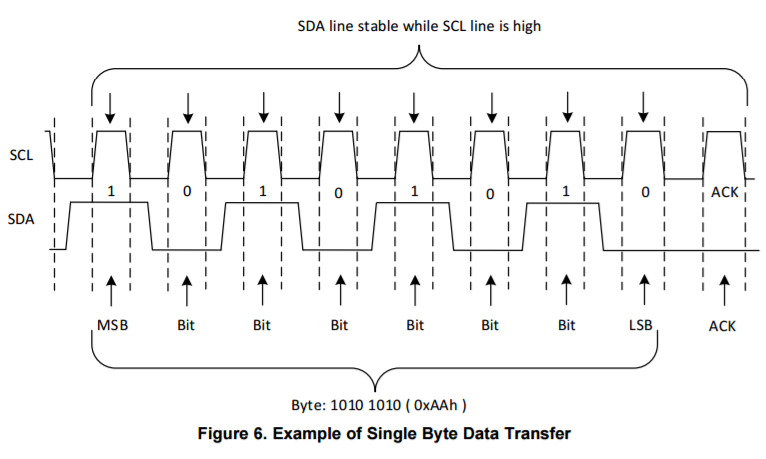
• Master-receiver terminates the transfer with a STOP condition

Start and Stop conditions:

I2C communication with this device is initiated by the master sending a START condition and terminated by the master sending a STOP condition. A high-to-low transition on the SDA line while the SCL is high defines a START condition. A low-to-high transition on the SDA line while the SCL is high defines a STOP condition.



Data Validity and Byte Format:

One data bit is transferred during each clock pulse of the SCL. Data is transferred Most Significant Bit (MSB) first. Any number of data bytes can be transferred from the master to slave between the START and STOP conditions. 

Acknowledge (ACK) and Not Acknowledge (NACK):

Each byte of data (including the address byte) is followed by one ACK bit from the receiver. The ACK bit allows the receiver to communicate to the transmitter that the byte was successfully received and another byte may be sent.

Before the receiver can send an ACK, the transmitter must release the SDA line. To send an ACK bit, the receiver shall pull down the SDA line during the low phase of the ACK/NACK-related clock period (period 9), so that the SDA line is stable low during the high phase of the ACK/NACK-related clock period.

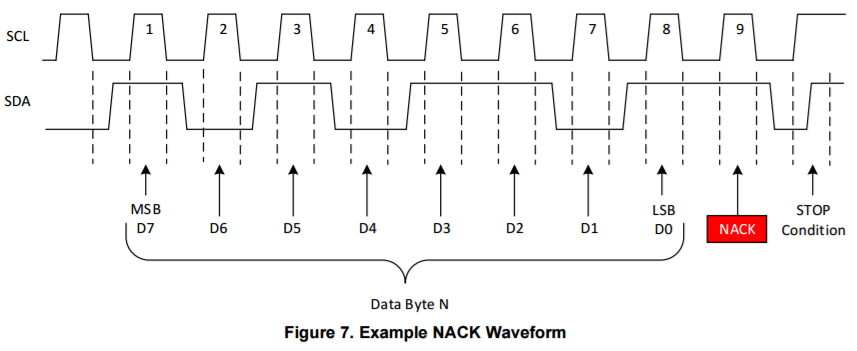
When the SDA line remains high during the ACK/NACK-related clock period, this is interpreted as a NACK. There are several conditions that lead to the generation of a NACK:

1. The receiver is unable to receive or transmit because it is performing some real-time function and is not ready to start communication with the master.

2. During the transfer, the receiver gets data or commands that it does not understand.

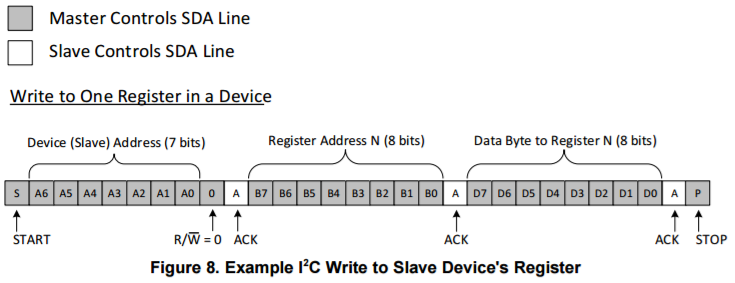
3. During the transfer, the receiver cannot receive any more data bytes.

4. A master-receiver is done reading data and indicates this to the slave through a NACK



Writing to a Slave On The I2C Bus:

To write on the I 2C bus, the master will send a start condition on the bus with the slave's address, as well as the last bit (the R/W bit) set to 0, which signifies a write. After the slave sends the acknowledge bit, the master will then send the register address of the register it wishes to write to. The slave will acknowledge again, letting the master know it is ready. After this, the master will start sending the register data to the slave, until the master has sent all the data it needs to (sometimes this is only a single byte), and the master will terminate the transmission with a STOP condition



Reading from a Slave On The I2C Bus:

Reading from a slave is very similar to writing, but with some extra steps. In order to read from a slave, the master must first instruct the slave which register it wishes to read from. This is done by the master starting off the transmission in a similar fashion as the write, by sending the address with the R/W bit equal to 0 (signifying a write), followed by the register address it wishes to read from. Once the slave acknowledges this register address, the master will send a START condition again, followed by the slave address with the R/W bit set to 1 (signifying a read). This time, the slave will acknowledge the read request, and the master releases the SDA bus, but will continue supplying the clock to the slave. During this part of the transaction, the master will become the master-receiver, and the slave will become the slave-transmitter. The master will continue sending out the clock pulses, but will release the SDA line, so that the slave can transmit data. At the end of every byte of data, the master will send an ACK to the slave, letting the slave know that it is ready for more data. Once the master has received the number of bytes it is expecting, it will send a NACK, signaling to the slave to halt communications and release the bus. The master will follow this up with a STOP condition.

