**what is I2C:**

 I2C stands for Inter-Integrated Circuit. It is a bus interface connection protocol incorporated into devices for serial communication. It was originally designed by Philips Semiconductor in 1982. Recently, it is a widely used protocol for short-distance communication. It is also known as Two Wired Interface (TWI).

Diagram

Description automatically generated

**Working of I2C Communication Protocol:**

It uses only 2 bi-directional open-drain lines for data communication called SDA and SCL. Both these lines are pulled high.

Serial Data (SDA) – Transfer of data takes place through this pin.

Serial Clock (SCL) – It carries the clock signal.

**I2C operates in 2 modes –**

Master mode

Slave mode

Each data bit transferred on SDA line is synchronized by a high to the low pulse of each clock on the SCL line.

According to I2C protocols, the data line cannot change when the clock line is high, it can change only when the clock line is low. The 2 lines are open drain; hence a pull-up resistor is required so that the lines are high since the devices on the I2C bus are active low. The data is transmitted in the form of packets which comprises 9 bits. The sequence of these bits is –

Start Condition – 1 bit

Slave Address – 8 bits

Acknowledge – 1 bit

Start and Stop Conditions:

START and STOP can be generated by keeping the SCL line high and changing the level of SDA. To generate START condition the SDA is changed from high to low while keeping the SCL high. To generate STOP condition SDA goes from low to high while keeping the SCL high, as shown in the figure below.

Table

Description automatically generated with medium confidence

**Repeated Start Condition:**

Between each start and stop condition pair, the bus is considered as busy and no master can take control of the bus. If the master tries to initiate a new transfer and does not want to release the bus before starting the new transfer, it issues a new START condition. It is called a REPEATED START condition.

**Read/Write Bit:**

A high Read/Write bit indicates that the master is sending the data to the slave, whereas a low Read/Write bit indicates that the master is receiving data from the slave.

**ACK/NACK Bit:**

After every data frame, follows an ACK/NACK bit. If the data frame is received successfully then ACK bit is sent to the sender by the receiver.

**Addressing:**

The address frame is the first frame after the start bit. The address of the slave with which the master wants to communicate is sent by the master to every slave connected with it. The slave then compares its own address with this address and sends ACK.

**I2C Packet Format:**

In the I2C communication protocol, the data is transmitted in the form of packets. These packets are 9 bits long, out of which the first 8 bits are put in SDA line and the 9th bit is reserved for ACK/NACK i.e. Acknowledge or Not Acknowledge by the receiver.

START condition plus address packet plus one more data packet plus STOP condition collectively form a complete Data transfer.

**Features of I2C Communication Protocol:**

Half-duplex Communication Protocol –

Bi-directional communication is possible but not simultaneously.

Synchronous Communication –

The data is transferred in the form of frames or blocks.

Can be configured in a multi-master configuration.

Clock Stretching –

The clock is stretched when the slave device is not ready to accept more data by holding the SCL line low, hence disabling the master to raise the clock line. Master will not be able to raise the clock line because the wires are AND wired and wait until the slave releases the SCL line to show it is ready to transfer next bit.

**Arbitration –**

I2C protocol supports multi-master bus system but more than one bus can not be used simultaneously. The SDA and SCL are monitored by the masters. If the SDA is found high when it was supposed to be low, it will be inferred that another master is active and hence it stops the transfer of data.

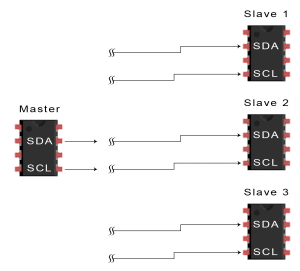
Serial transmission –

I2C uses serial transmission for transmission of data.

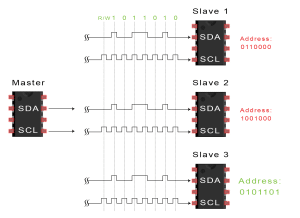
Used for low-speed communication.

**STEPS OF I2C DATA TRANSMISSION**

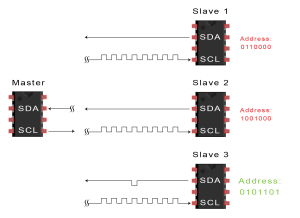
1. The master sends the start condition to every connected slave by switching the SDA line from a high voltage level to a low voltage level before switching the SCL line from high to low:

[](https://www.circuitbasics.com/wp-content/uploads/2016/01/Introduction-to-I2C-Data-Transmission-Diagram-START-CONDITION-3.png)

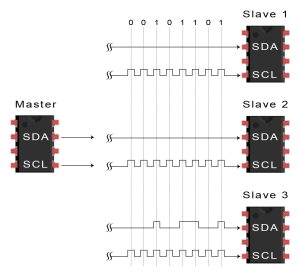
2. The master sends each slave the 7 or 10 bit address of the slave it wants to communicate with, along with the read/write bit:

[](https://www.circuitbasics.com/wp-content/uploads/2016/01/Introduction-to-I2C-Data-Transmission-Diagram-ADDRESS-FRAME-2.png)

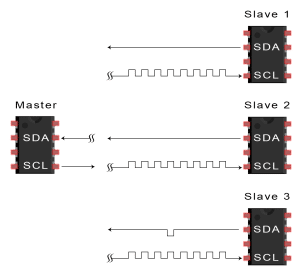
3. Each slave compares the address sent from the master to its own address. If the address matches, the slave returns an ACK bit by pulling the SDA line low for one bit. If the address from the master does not match the slave’s own address, the slave leaves the SDA line high.

[](https://www.circuitbasics.com/wp-content/uploads/2016/01/Introduction-to-I2C-Data-Transmission-Diagram-ACK-Bit-Slave-to-Master-2.png)

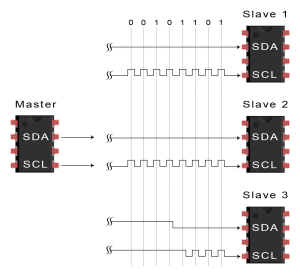
4. The master sends or receives the data frame:

[](https://www.circuitbasics.com/wp-content/uploads/2016/01/Introduction-to-I2C-Data-Transmission-Diagram-Data-Frame.png)

5. After each data frame has been transferred, the receiving device returns another ACK bit to the sender to acknowledge successful receipt of the frame:

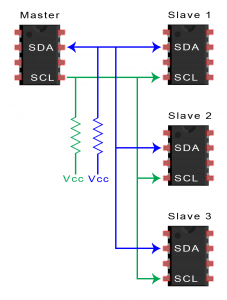
[](https://www.circuitbasics.com/wp-content/uploads/2016/01/Introduction-to-I2C-Data-Transmission-Diagram-ACK-bit-slave-to-master-2A.png)

6. To stop the data transmission, the master sends a stop condition to the slave by switching SCL high before switching SDA high:

[](https://www.circuitbasics.com/wp-content/uploads/2016/01/Introduction-to-I2C-Data-Transmission-Diagram-Stop-Condition.png)

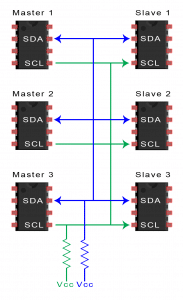
**SINGLE MASTER WITH MULTIPLE SLAVES**

Because I2C uses addressing, multiple slaves can be controlled from a single master. With a 7 bit address, 128 (27) unique address are available. Using 10 bit addresses is uncommon, but provides 1,024 (210) unique addresses. To connect multiple slaves to a single master, wire them like this, with 4.7K Ohm pull-up resistors connecting the SDA and SCL lines to Vcc:

[](https://www.circuitbasics.com/wp-content/uploads/2016/01/Introduction-to-I2C-Single-Master-Multiple-Slaves-2.png)

**MULTIPLE MASTERS WITH MULTIPLE SLAVES**

Multiple masters can be connected to a single slave or multiple slaves. The problem with multiple masters in the same system comes when two masters try to send or receive data at the same time over the SDA line. To solve this problem, each master needs to detect if the SDA line is low or high before transmitting a message. If the SDA line is low, this means that another master has control of the bus, and the master should wait to send the message. If the SDA line is high, then it’s safe to transmit the message. To connect multiple masters to multiple slaves, use the following diagram, with 4.7K Ohm pull-up resistors connecting the SDA and SCL lines to Vcc:

[](https://www.circuitbasics.com/wp-content/uploads/2016/01/Introduction-to-I2C-Multiple-Masters-Multiple-Slaves-2.png)

**Advantages:**

 Can be configured in multi-master mode.

Complexity is reduced because it uses only 2 bi-directional lines (unlike SPI Communication).

Cost-efficient.

It uses ACK/NACK feature due to which it has improved error handling capabilities.

**Limitations:**

Slower speed.

Half-duplex communication is used in the I2C communication protocol.

**Qualcomm I2C Details:**

This section describes the QUP engine. The QUP engine is an advanced high performance slave port that provides a common data path (an output FIFO and an input FIFO) for the device interintegrated circuit (I2C) and serial peripheral interface (SPI) interfaces.

**QUP overview**

Below figure, shows a high-level overview of the six QUP identical instances integrated within the bus access manager (BAM) low speed peripheral (BLSP) in the SoC.

Diagram, schematic

Description automatically generated

QUP -SE will support I2C and SPI use-cases across SOC. Like sensors, camera, PMIC use-cases and subsystems.

**QUP external connectivity**

Figure2, shows the QUP I2C and SPI external interface signals

Diagram

Description automatically generated

**NOTE:** An I2C and SPI interface within a single QUP share pads that cannot be used at the same time. A single QUP can be configured to use up to two pads for the I2C and up to four pads for the SPI. In addition, QUP[1-3] have two additional CS lines, and can be configured to use up to six pads for SPI.

I2C and SPI signals are labeled BLSPx (where x = 1 to 6 for the six QUP instances).

**QUP I2C protocol and data format**

shows a typical bus transfer with valid data on the I2C bus

A picture containing text, indoor

Description automatically generated

shows the start and stop transfer conditions on the I2C bus

Shape, polygon

Description automatically generated

shows the byte format and acknowledgments on the I2C busDiagram

Description automatically generated

shows a typical data transfer format in I2C FS mode.

Diagram

Description automatically generated

**Project:**

**Dragon Board Schematic Explanation:**

**I2C + ADXL345**

**Code**

**Platform Driver**

**Driver flow**

**Client driver**