

# I2C Transport

## Introduction

Each I2C bus consists of two lines (signals): a clock line (SCL) and a data line (SDA). The devices on the I2C bus are either masters or slaves. The master is always the device that drives the clock line. The slaves are the devices that respond to the master. Only the master can initiate a transfer over the I2C bus. Each slave has a unique address.

**i** All bytes in this section are written in hexadecimal form, if not indicated otherwise.

**i** The Touch Sensor Module does not communicate using registers or memory accesses, so the documentation for the STM32 microcontrollers is not applicable.

## The DataReady signal

The Touch Sensor Module uses a signal called DataReady (DR) to inform the host that there is data to read and that a read operation can be initialized. Refer to [Electrical Integration](#) for more information.

## Module Address

The slave I2C address of the sensor module is 0x50 (7 bit). The address itself consists of 7 bits but 8 bits are always sent. The extra bit informs the slave if the master is reading from or writing to it. If the bit is 0 the master is writing to the slave. If the bit is 1 the master is reading from the slave.

The resulting address bytes are 0xA1 (read) and 0xA0 (write).

## Syntax

The I2C Transport Protocol is simple and the syntax is identical in both directions (read or write):

Byte number	1	2	3 to n
Description	FrameStart, a constant. The value is always EE.	DataSize, the number of bytes in the payload. Datasize >= 1.	Payload (serialized ASN.1 message).

Examples:

1. The payload is 8 bytes long. The transmission reads:

```
EE 08 [payload]
```

2. The payload is 25 bytes long. The transmission reads:

```
EE 19 [payload]
```

## Sending Data

To send data to the Touch Sensor Module, the host initiates a write operation for the sensor module's address and writes the full payload.

(Do not confuse the I2C FrameStart constant with the type byte that indicates that a serialized message is a request. The value is EE for both bytes, but the meaning is completely different.)

Example: Sending a request with an ENABLE command to the sensor module:

```
EE 0B EE 09 40 02 02 00 65 03 81 01 00
```



### DataReady

If the sensor module signals DataReady, it is NOT permitted to initiate an I2C write operation as the sensor module is waiting for the host to initiate a read operation.

Always wait for the corresponding response from one command before sending another command. Note that not all commands have a response command. If there is no corresponding response command, the host is free to issue another command.

## Receiving Data

The sensor module triggers the DataReady signal when there is data for the host to receive. To maximize the performance and minimize the load on the I2C bus, the host is expected to read data in a certain sequence:

1. The sensor module asserts DataReady
2. The host initiates a two bytes I2C read operation for the I2C 7-bit address 0x50.
3. The sensor module fills in the first two bytes, FrameStart and DataSize.
4. The host initiates an I2C read operation for address 0x50 and reads XX bytes (as indicated by DataSize).
5. The sensor module deasserts DataReady.

## BootComplete

When the sensor module has finished booting, and hardware such as the I2C module have been configured, the command BootComplete is put into the send queue. The DataReady signal is triggered and the host needs to read the data to acknowledge that the module is ready to operate. If the sensor module powers on before the host system does, the host system needs to check if the DataReady signal is active, in which case it needs to read the data.



Do not send any commands to the sensor module before BootComplete has been read.

## Read More about the Communication Protocol

- [Serialization Protocol Quick Start](#)
- [USB HID Transport](#)
- [I2C Transport](#)
- [zForce Message Specification](#)
- [Understanding the zForce ASN.1 Serialization Protocol](#)

## Read More

- [Introduction](#)
- [Getting started with Touch Sensor Module Evaluation](#)
- [Getting Started with Software Integration](#)
- [Mechanical Integration](#)
- [Electrical Integration](#)
- [Software Integration](#)
- [Implementation Examples](#)
- [Specifications](#)
- [Legal Notice](#)