

# I<sup>2</sup>C bus and devices

## What is I<sup>2</sup>C?

I<sup>2</sup>C is probably the most common protocol used to exchange information between microcontrollers and sensors, displays or actuators. I<sup>2</sup>C is for instance the protocol used under the hood during IDP to link your different boards together.

In this section, you will find a brief introduction to I<sup>2</sup>C, with probably just enough information to get you started. Not everything will be crystal clear, but hopefully most of the relevant sections of the examples code will make sense, and you will manage to modify them to get them to do what you want.

Start with the following intro to I<sup>2</sup>C, from NXP, the company who developed it a while ago, and also the manufacturer of the sensor you will use ([https://www.youtube.com/embed/qeJN\\_80CiMU](https://www.youtube.com/embed/qeJN_80CiMU)).

Of course you could go much deeper into the subject if you want. A few external links are pasted below for reference, but reading them can wait for now.

You may need to refer to them later on to better understand the code examples.

I<sup>2</sup>C bus specification: [short](#) and [extended](#).

# Connecting an I<sup>2</sup>C device to your microcontroller

You need first to identify how to connect I<sup>2</sup>C devices to the microcontroller board. I<sup>2</sup>C sensors need at least 4 connections, two for power (VCC and GND), and two for communications (CLK for the clock signal and SDA for the data).

The [user manual](#) of your microcontroller should allow you to identify the default pins for these communications. You can also find the description of the pins on the [spec sheet online](#).

## Task

**Identify the different I2C connections available on your microcontroller. We will use the default I2C pins, called I2C1\_SCL and I2C1\_SDA. Can you identify them on the board? We will also need to power the sensor using the 3.3V pin to VCC, and GND. Look also for these pins on the board. They may appear multiple times. We will use the female headers to connect to the make jumper wires.**

Most of the microcontrollers will come with libraries and tutorials to quickly get your system to communication through I<sup>2</sup>C. Here is the link to the mbed documentation: [mbed I2c class](#). Note that the protocol is simple enough that with a bit of experience it is easy to understand what signals microcontroller and attached devices are exchanging.

## The data sheet of an I<sup>2</sup>C device

For any device you wish to connect using I<sup>2</sup>C, you will need to:

- identify or set the device address;
- configure the device if needed, by setting the value of different registers controlling the operation of the device;
- read and send data to the device.

All this information can be found in the data sheet of the device you are planning to use. Make sure you check the availability of a good documentation before selecting a device. We will work with a sensor produced by NXP, with a rather good documentation to support the work of the developer.

The data sheet of the LM75 sensor you will use is available [here](#).

This is daunting document! It probably contains many technical words you are not familiar with. However, you probably don't need to understand it all to get to use the device in your application. Have a quick scan through it at first. If you know what you are looking for, you will get this information out of it quickly enough afterwards.

Most of the time, there will also be sample codes available to you online. The sensor we use here is fairly popular, and the mbed compiler even contains a fully functional template! Of course, we will use this to make sure that we have an easy start.

### Task

**Look at the data-sheet of the sensor. What is the address range? How to set it? Your sensor is already soldered to a breakout board. Look at the back of the breakout board, and try to understand how to set the address of the device. The next section will show you how to implement it.**