Appendix G: I²C (Inter Integrated Circuits) Interface

I²C is a synchronous 8-bit oriented serial communication bus between integrated circuits installed next to each other (normally on the same board). Created by Philips in the 1980s, it is a two-wire bus with standardized speeds ranging from 100 kbps up to 5 Mbps. I²C has some advantages over SPI (appendix F), such as the use of less wires, flow control (acknowledgement bits), a more formal construction, and the possibility of having more than one master. On the other hand, it is slower, requires pull-up resistors (for open-drain pins), and is more complex to implement.

1. I²C Bus Structure

The general structure of an I²C bus is depicted figure G.1. Its two wires are called *SCL* (serial clock) and *SDA* (serial data), which interconnect one or more master units to a number of slave units. A common ground wire (not shown) is obviously also needed for the system to function. The IC families with I²C support are the same as for SPI (appendix F).

As indicated in figure G.1, the clock (*SCL*), always generated by the master (usually a microcontroller), is unidirectional, whereas the data (*SDA*) wire is bidirectional. Because *SCL* and *SDA* are open-drain lines (the 5Mbps version allows push-pull logic), external pull-up resistors (R_{PU}) must be connected between these wires and V_{DD}. The number of devices

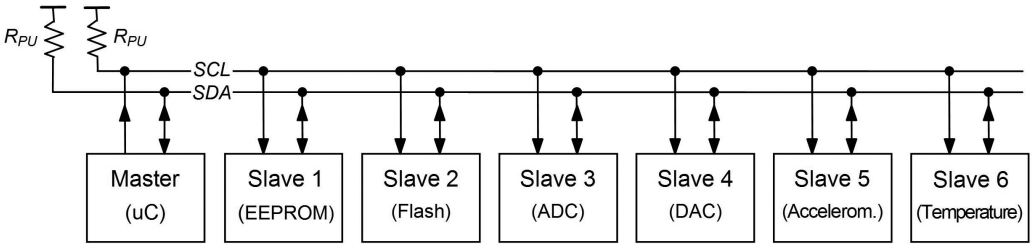


Figure G.1
General I²C bus structure.

sharing the same bus can be up to 128 (7-bit address) or 1024 (10-bit address). Advanced features of the I²C protocol include bus arbitration, clock stretching, general call, reset by software, and so on.

Common values for V_{DD} are 3.3 V and 5 V, but devices with lower voltages, like 1.8 V, are also available. The value of R_{PU} is typically in the 1 kΩ to 33 kΩ range, depending on the total *SCL* or *SDA* wire capacitance; if it is large (a long bus with many slaves), then the resistor must be small to achieve the minimum rise-time defined in the I²C specifications.

2. I²C Bus Operation

Data transfers are done in groups of 8 bits, after which an acknowledgment bit is issued by the receiving end. The general principle is depicted in figure G.2a, which shows a data transmission from master to slave. The start sequence consists of lowering *SDA* with *SCL* high (gray area), and the stop sequence (the other gray area) consists of raising *SDA* with *SCL* high. This means that during data transmission *SDA* must remain stable while *SCL* is high; otherwise, start/stop commands might occur.

While the master is transmitting (always the most significant bit [MSB] first), the slave remains in high-impedance mode (`'Z'`), so the master has control over the *SDA* wire. After the eighth bit is sent, the master goes to the `'Z'` state, so the slave can transmit its acknowledgement bit (=0). In reading procedures, it is the master who emits the ack bit (=0) after each byte received; however, if the reading includes multiple bytes, then the master sends a no-ack bit (=1) after the last byte that it wants to retrieve. As depicted in figure G.2b, data is always placed on the *SDA* wire (by either end) at falling clock edges and is read from it at rising clock edges (thus exactly as in SPI).

It is obviously necessary to respect the minimum values of the time parameters T_1 to T_3 and T_{SCL} as shown in figure G.2a (these are just the main time parameters). T_1 to T_3 are usually smaller than $T_{SCL}/2$, so $T_{SCL}/2$ or T_{SCL} can be employed for those time windows (see figure G.2a). As actual examples, the values of these parameters in the Maxim MAX11647 A/D converter device are $T_1 = T_2 = 0.6$ μs, $T_3 = 1.3$ μs, and $T_{SCL} = 2.5$ μs; and in the NXP PCF8593 real-time clock device they are $T_1 = T_3 = 4.7$ μs, $T_2 = 4$ μs, and $T_{SCL} = 10$ μs.

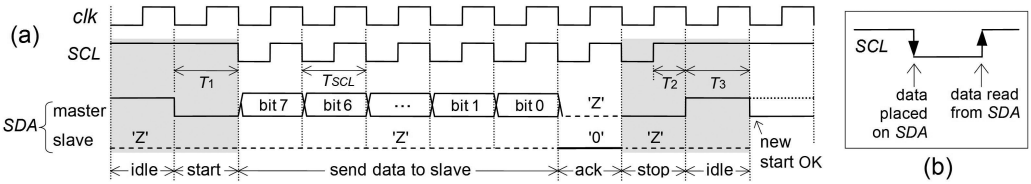


Figure G.2
I²C communication principle.

3. Example: I²C Interface for an A/D Converter

To illustrate the use of I²C, the MAX11647 device (figure G.3a) from Maxim is used as an example. It is a two-channel, 10-bit analog-to-digital converter (ADC), with I²C access at 400 kHz or 1.7 MHz. It operates with V_{DD} = 3.3 V. The reference voltage can be internal (V_{DD} or 2.048 V) or external (between 1 V and V_{DD}). Its address (set at manufacturing) is 0110110.

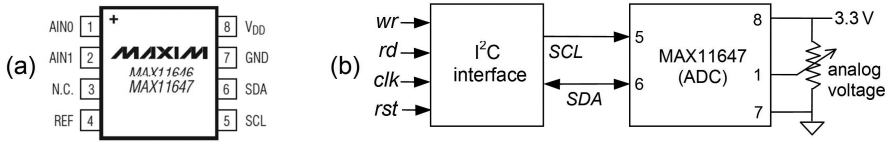
A setup for the experiments is shown in figure G.3b. However, before using (reading from) this device, its two registers, called *setup* and *configuration* registers (figures G.3c and d), must be programmed. The following will be adopted in the experiments developed later with VHDL:

Setup register: MSB = 1 (it is a setup byte), then 101 (reference voltage is 2.048 V, internal, always on), followed by 0 (conversion clock is internal), 0 (unipolar), 1 (internal reset disabled), and finally LSB = – (don't care). In summary: 1101001–.

Configuration register: MSB = 0 (it is a configuration byte), then 11 (use input selected by CS0), followed by – – – (don't care), then 0 (use input 0), and finally LSB = 1 (single ended). In summary: 011 – – – 01.

Figure G.4a shows how the registers can be programmed. Communication starts when *SDA* is lowered with *SCL* high. The first byte consists of the slave address plus a write (= 0) bit, after which the master goes to the high impedance state (`'Z'`), so the slave can transmit its acknowledgment bit (= 0). Next, the master sends the setup byte, to which the slave again responds with an ack bit. Finally, the configuration byte is transmitted, so after the ack bit the master ends the write procedure.

The read procedure is depicted in figure G.4b. The master again starts the communication by lowering *SDA* with *SCL* high. Next, it transmits the slave address plus a read (= 1) bit, to



(c) Setup Register	
Bit	Description
7 - REG	Byte purpose: 1=setup; 0=configuration
6 - SEL2 5 - SEL1 4 - SEL0	Reference voltage: 00X=VDD; 01X=external; 100=internal, always off; 101=internal, always on; etc.
3 - CLK	Conversion clock: 0=internal; 1=external
2 - BIP/UNI′	Polarity: 0=unipolar; 1=bipolar
1 - RST′	Reset: 0=default parameters; 1=no action
0 - X	Don't care

(d) Configuration Register	
Bit	Description
7 - REG	Byte purpose: 1=setup; 0=configuration
6 - SCAN1 5 - SCAN0	11=scan input selected by CS0; 00=scan from AIN0 to that selected by CSO; etc.
4 - X 3 - X 2 - X	Don't care
1 - CS0	Input selection: 0=AIN0; 1=AIN1
0 - SGL/DIF′	Input type: 1=single ended; 0=differential

Figure G.3
(a) MAX11647 pins; (b) Setup for the experiments; (c) Setup register; (d) Configuration register.

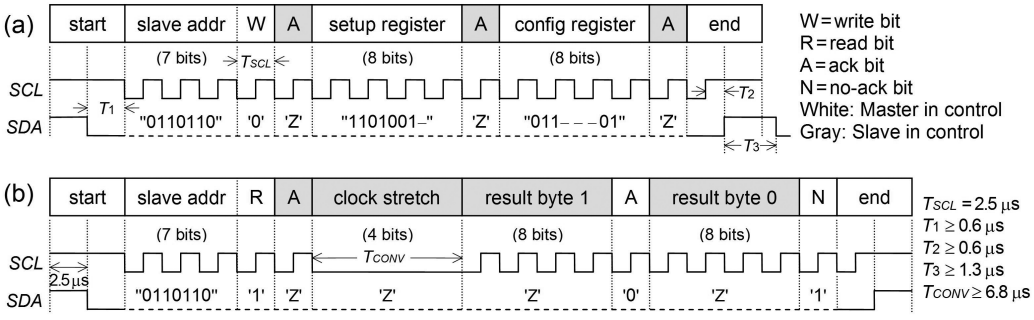


Figure G.4
I²C write and read procedures for the MAX11647 ADC.

which the slave responds with an ack bit. After this, the slave takes over the *SDA* wire (gray shade). First, a time large enough (clock stretch) is given to the slave to perform the analog-to-digital conversion; the device specifications say that $t_{CONV} = 6.8$ µs, so three clock periods suffice if the speed is 400 kHz (four periods were employed in the experiments). Next, the two-byte result is transmitted to the master (the six MSBs of byte 1 are zero; recall that this is a 10-bit ADC). If the master wants a single reading, it sends a no-ack bit (= 1) after byte 0 is received, so the slave stops sending data and the master ends the communication.

The values of T_1, T_2, T_3, and T_{CONV} are listed in figure G.4. The reader is invited to check whether they are satisfied by the waveforms presented there.

The circuit discussed in this example is finalized and implemented with VHDL in sections 17.3 and 17.4.