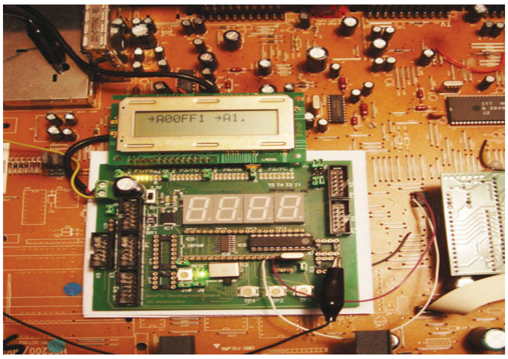
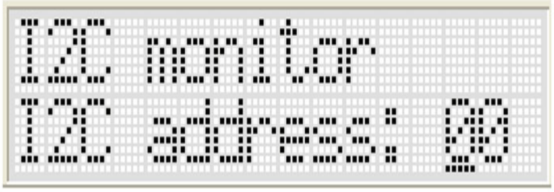


Integrated circuits within telecommunications and industrial electronic devices and complex consumer devices (for example, in a TV set) often communicate using the I2C protocol. This article describes a simple device that can monitor traffic on the I2C bus. Let’s find out what kind of messages these chips exchange!

I2C fundamentals

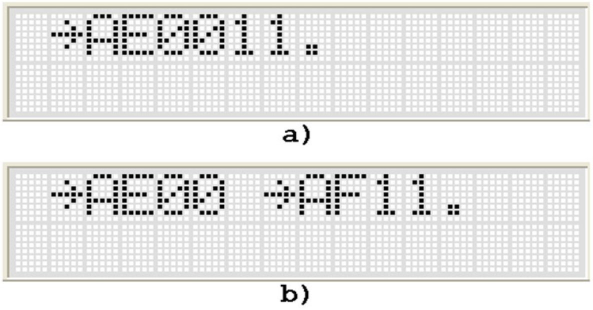


```
I2C monitor
I2C adress: 00
```

The I2C bus is a simple bidirectional 2-wire bus for efficient inter-IC control (which is where the abbreviation I2C comes from). These two lines are called the serial data line (SDA) and the serial clock line (SCL). Serial, 8-bit bidirectional data transfers can be made at up to 100 kbit/s in the standard mode or up to 400 kbit/s in the fast mode. The traffic on the bus is controlled by the master IC (usually a microcontroller) that generates clock signals and initiates and terminates a transfer. Each slave device connected to the bus is software addressable by a unique address and will only respond if addressed by the master.

Communication begins with the START condition, generated by the master. The START condition is indicated by a High to Low transition on the SDA line while the SCL is High. The bus is considered to be busy after the START condition. Following the START condition, the master will address one of the slave ICs connected to the I2C bus. The address is a 7-bit number. The 8th bit is a R/W bit. If this bit is Low, the master will send one or more blocks of instructions or data to the addressed slave. If this bit is High, the master is ready to receive one or more blocks of data from the addressed slave. The data transferred via the I2C bus consists of 8-bit packets, each followed by the 9th - acknowledge bit. The data is transferred with the most significant bit (MSB) first and the number of bytes that can be transmitted per transfer is not restricted. The new bit is placed on the SDA line while SCL is Low. The master ends the communication with a Low to High transition on the SDA line while SCL is High, which is considered the STOP condition. The STOP condition releases the addressed slave and the I2C bus.

The 7-bit addressing scheme, explained above, allows up to 112 different addresses. Most new chips use a 10-bit addressing scheme, which provides more address combinations and thus more unique addresses. The 10-bit addressing scheme uses two bytes for addressing purposes and does not affect compatibility with 7-bit addressing in any way. Thus 7- and 10-bit devices can be connected to the same I2C bus.



```
→AE0011.
```

a)

```
→AE00 →AF11.
```

b)

Let’s see how this communication works, using for example the AT24C02 serial EEPROM. Assuming that all the address pins on the device have been wired to a logic High, this serial EEPROM will have an I2C address of “AE”. We want to write an “11” at EEPROM address “00” (addresses and data in this example are given as hexadecimal numbers). The communication will look like that shown in the following example (see Table 1).

Reading the same data from the AT24C02 is shown in the following example, (see Table 2). The acknowledge bits are omitted from this example for simplicity. In reality, the receiving device confirms the reception of each byte with an ACK bit.

I2C monitor

An ATtiny2313-based I2C monitor looks very simple, but all the “clever things” are hidden in the microcontroller’s program. The program is written in Bascom-AVR with several embedded assembler routines. Assembler routines are written for monitoring the traffic on the I2C bus, where all events take place at high speed. For example, in I2C fast mode, with a bit rate up to 400 kHz, the program has only 2.5 µs to read and process each bit. Simultaneously, the program must properly react to not only the expected data patterns, but to accidental START and STOP conditions, which can also occur during communication. To simplify the hardware, the START and STOP conditions are handled by software, which also dictates specific programming requirements. To speed up program execution, a 20 MHz crystal is used in the final version of the monitor, although a somewhat simpler version of the monitor, written during development, worked well even with an 8 MHz crystal.

I2C monitor

2012_AVR_UK_163

Continue reading