Accurate USB 2.0 temperature sensor needs only a handful of parts

Silvio Lauckner, Ismaning, Germany

This Design Idea presents a simple, accurate, and reliable design to measure temperature using the USB. **Figure 1** shows the complete circuit of the temperature-sensor device. The heart of the sensor device is an FT232RQ USB-to-serial converter from FTDI (Future Technology Devices International, www.ftdichip.com). In addition to using the FT232 in its default UART mode, the FT232 works in a so-called bit-bang mode (**references 1**, **2**, and **3**). This mode changes its I/O lines into a bidirectional data bus, which the user can fully control. The connection with the USB takes place in a standard manner, and the back end of the chip interconnects to an AD7814 digital temperature sensor from Analog Devices (www.analog.com, **Reference 4**).

The temperature sensor uses a four-wire SPI but only three pins: SCK (serial clock), SS# (slave select), and SDO (serial-data out). To avoid any malfunction of the sensor, the SDI (serial-data-in) line must be grounded. The FT232 acts as an SPI master and emulates the protocol for the AD7814 by setting or clearing the appropriate port pins for SS# and SCK. The data from the sensor gets read back together with all the other bus lines. This process occurs simultaneously with the write process.

Figure 1 This circuit for a USB temperature sensor works in default UART and bit-bang modes.

designideas
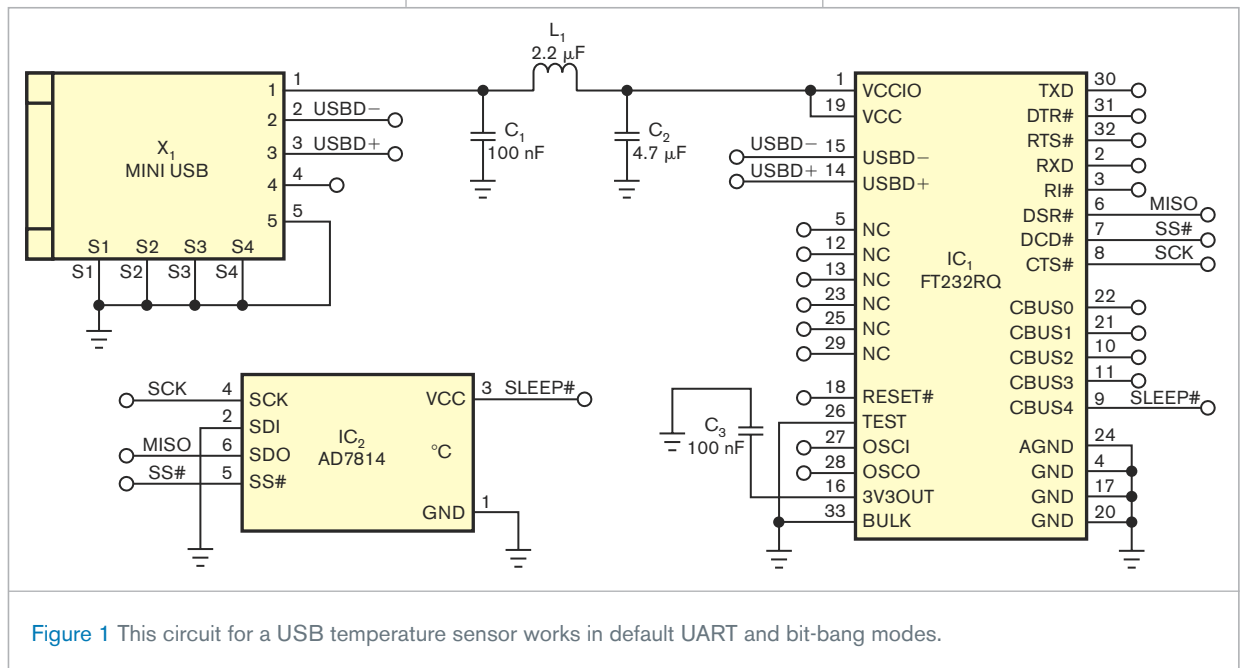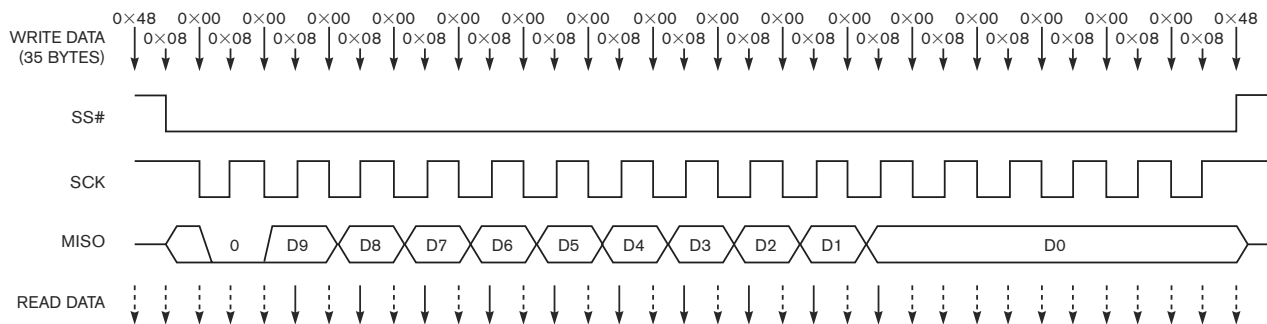
Figure 2 The timing diagram for the AD7814 shows considerable data overhead to download 10 bits of temperature data.
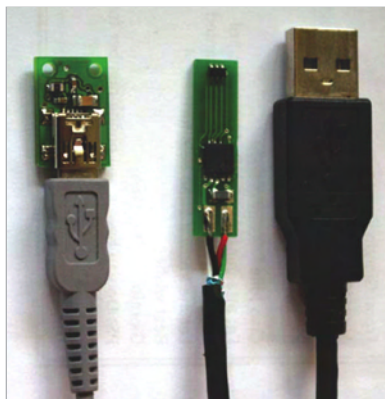
Figure 3 Two sample circuits (left and center) are smaller than a USB Type A plug (right).

To comply with the USB specification, you power down the temperature sensor using the sleep signal while the USB logic is in suspend mode. The sensor device receives its power through the USB and draws only about 20 mA. On the software side, you need only to open the device and switch the chip into the bit-bang mode. After that action, you can send the fixed pattern to emulate an SPI master from the host PC to the FT232 (**Figure 2**). The software returns a data array of the port samples of both the PC and the FT232, whose ports are inputs and outputs.

Because the FT232 chips come with a unique serial number, you can identify the correct device within a multichip environment. So, you can put more than one FT232-based sensor onto a computer. The core of this Design Idea is not limited to measuring temperature. You can use other sensors with digital interfaces, as well.

To get the current temperature, you must write 35 fixed bytes into the port register. The sensor expects 16 clock pulses on the SCK line while the SS# is low. The clock frequency is 1 MHz. The device samples sensor-read data during the write operation. After the protocol on the back end finishes, you can retrieve the data from the host PC for further processing. To get just 10 bits out of the sensor involves considerable data overhead (**Figure 2**). The dashed-line arrows mark the bytes, which need no further evaluation.

This Design Idea realized two sample circuits on a two-layer PCB (printed-circuit board) measuring only 18×12 mm (0.7×0.47 in.) and 7.6×30.5 mm (0.3×1.2 in.). **Figure 3** shows them in comparison to the size of a USB Type A plug.**EDN**

REFERENCES

[1] "FT232R USB UART IC," Future Technology Devices International, 2005, www.ftdichip.com/Documents/DataSheets/DS_FT232R.pdf.

[2] "AN232R-01 Bit Bang Modes for the FT232R and FT245R," Future Technology Devices International, 2005, www.ftdichip.com/Documents/AppNotes/AN232R-01_FT232RBitBangModes.pdf.

[3] "D2XX Programmer's Guide," Future Technology Devices International, 2005, www.ftdichip.com/Documents/ProgramGuides/D2XXPG33.pdf.

[4] Analog Devices, Data sheets AD7814 (www.analog.com/en/prod/0,,764_814_AD7814,00.html), ADT7301 (www.analog.com/UploadedFiles/Data_Sheets/ADT7301.pdf), and ADT7302 (www.analog.com/en/prod/0,,764_811_ADT7302,00.html).