ADuC814 to ADM1032 via I2C® Interface
By Brian Moss

INTRODUCTION

Even though the ADuC814 has an on-chip ±1.5°C temperature sensor, configured through the on-board ADC registers, it is often required that a remote or external temperature sensor be used. The ADM1032 temperature sensor (±1°C accurate) comes in a small 8-lead package and returns a temperature, measured in centigrade, via an SMBus interface.

ADM1032

The ADM1032 is a dual channel digital thermometer with overtemperature and undertemperature alarms. It is primarily intended for use in the computer and thermal management industries. It comes in an 8-lead mini small outline package (MSOP) and communicates with the outside world via a 2-wire SMBus interface (I2C compatible). It has a default device address of 0x4C. See Technical Note uC001, MicroConverter® I2C Compatible Interface, in the Technical Library section of the Analog Devices website at www.analog.com/microconverter. Several registers are

accessible on the ADM1032, but for this application, this application note covers only one: Register Address 0x00, the address of the data register for the on-chip sensor. This means that connectivity will be at a minimum with only standard I2C and ADuC814 components being required.

How It Works

On power-up, the ADM1032 continuously converts, by default, on its internal sensor and updates its data register, Address 0x00, automatically. In this mode of operation, very little work is required in gathering the samples, except writing the appropriate I2C commands and addresses to it.

The device address of the ADM1032 is 0x4C, which must be correctly formatted for use in an I2C communications protocol. This means shifting 0x4C to the left by one place, as is shown in the I2C bit pattern below (Figure 1). The LSB now becomes the read/write bit (See Technical Note uC001 for details).

A scope plot, captured via GPIB, of the I2C waveform diagram for the entire operation is as shown in Figure 1.
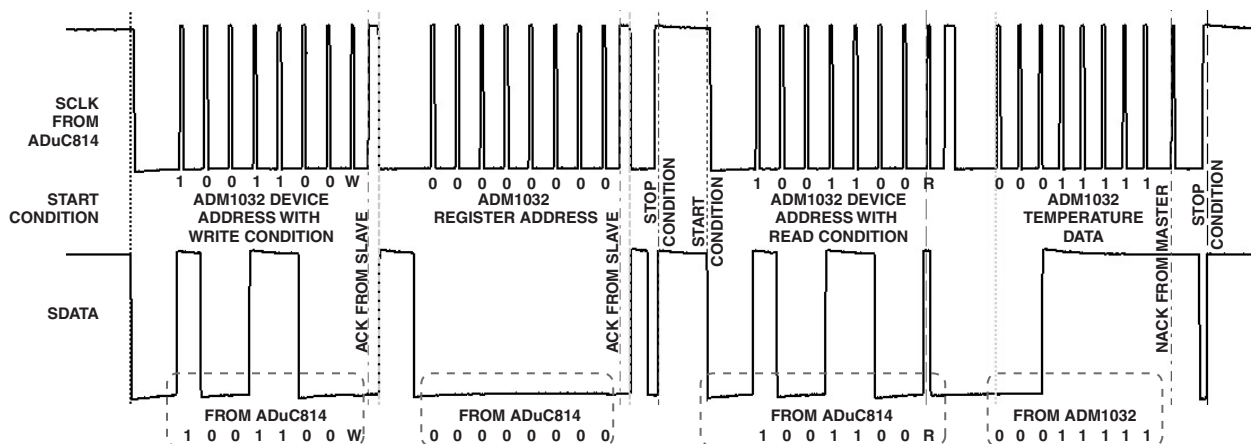

Figure 1. I2C Waveform

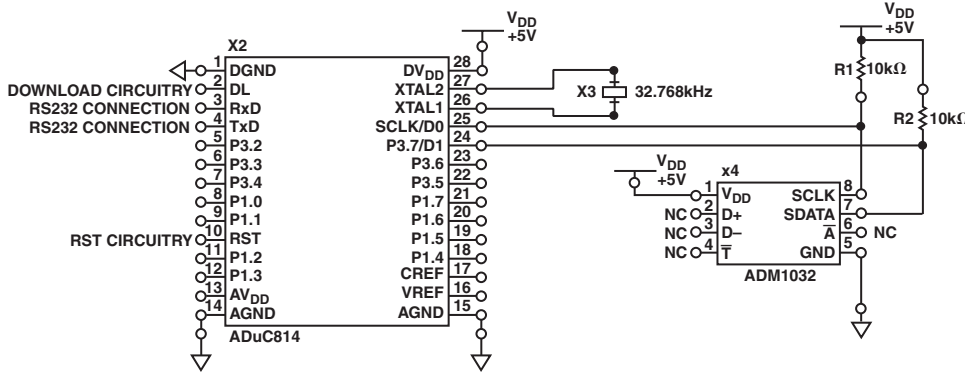In all operations, the ADuC814 is the master device and the ADM1032 the slave. The master device always

outputs the SCLK. Note that the temperature returned in the above example is 00011111b = 31°C.

As can be seen in Figure 1, the slave device address 0x98 (shifted 0x4C with the LSB set to write) is written from the master device through the SDATA line with the master generating eight clock pulses on the SCLK line. A ninth SCLK pulse is also generated for the slave acknowledgment (ACK). The master then outputs the specific register address required (slave Register Address 0x00 is the one needed). The master again outputs a ninth plus a tenth SCLK for the slave acknowledgment and stop condition, respectively. Following this, the master outputs the slave device address again, but this time with the R/W bit (LSB of the slave device address byte) set to read (high), resulting in a slave device address byte of 0x99 being sent. This read bit tells the slave on the next series of SCLKs, also generated by the master, to output the contents of the reg-

ister address previously received (Register Address 0x00) to the SDATA line. The master then sends the required eight SCLKs to the slave to transfer the contents of the data register from the slave to the master. The master again also sends a ninth and tenth SCLK for the no-acknowledgment (NACK) and the stop condition, respectively. These signals are explained in Technical Note uC001.

SCHEMATIC
Figure 2 shows the schematic for ADuC814 to ADM1032 via the I²C interface. The connectivity for this circuit is at a minimum and involves only two wires, SCLK and SDATA. Note that for clarity, some pin details for the ADuC814 and circuitry are omitted. This circuitry is important and should not be omitted in practice.

SOME PIN NAMES ARE INCOMPLETE FOR CLARITY.
SOME EXTERNAL CIRCUITRY IS IMPLIED FOR THE ADuC814.

Figure 2. ADuC814 to ADM1032 Interface

CODE
The code used can be found in the accompanying ADM1032.ASM file. The data is transmitted (or received) via the SDATA line bit by bit on every rising edge of SCLK. For transmission from the master to the slave, the data is initially moved into the MDO bit of the I2CCON register of the master bit by bit and gets outputted on SDATA on the rising edge of SCLK. SCLK is controlled by setting and clearing the MCO bit (again in I2CCON) in sequence. This is the operation for transmitting the device address,

register address, and again the device address. Following the second device address, the SDATA line on the ADuC814 must be turned around to accept data from the ADM1032 (slave). This is done by clearing the MDE bit in I2CCON. Again, all data is transmitted and received on the rising edge of SCLK as generated by the bit banging of the MCO bit. The slave transmits its data via the SDATA line on each SCLK from the master and is received into the MDI bit on each SCLK. Please review Technical Note uC001 for details about the start and stop conditions.

OPERATION

The operation of the system is very simple. Once the code and hardware have been downloaded and set up, simply toggle reset, configure HyperTerminal (for 9600 baud in this case) or whatever other serial interrogation method you decide (LabVIEW™, Visual Basic®, etc.), and read the data. The temperature data from the ADM1032 is in centigrade by default. Figure 3 shows a typical output, the nominal temperature when the system started to run was 0x1A (26°C). As the temperature of the ADM1032 was increased (by touching it), the output can be seen to increase to 0x1BH (27°C) and 0x1C (28°C).
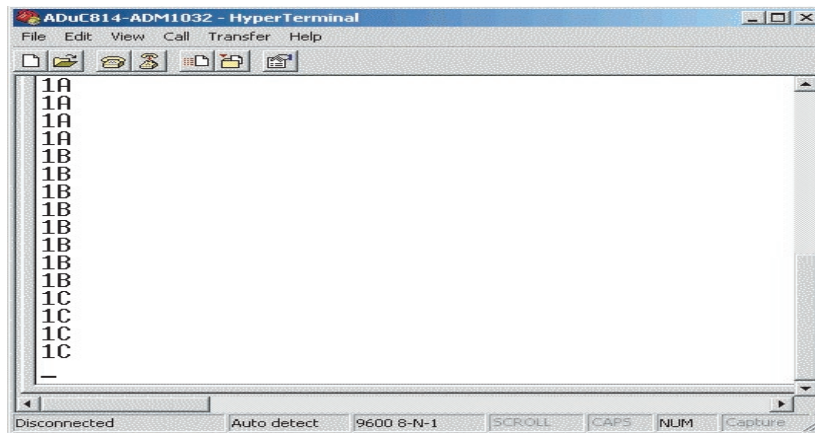

Figure 3. Graphical Display of Temperature Output

For more information about the ADM1032, see:
http://www.analog.com/Analog_Root/productPage/productHome/0%2C2121%2CADM1032%2C00.html

For more information about the ADuC814, see:
http://www.analog.com/Analog_Root/productPage/productHome/0%2C2121%2CADUC814%2C00.html