Application Note

Embedded Design Techniques for Verifying Serial Communications

Introduction

The typical embedded system is made up of a microcontroller and various peripheral devices interfacing with the outside world. In an increasing number of designs the communication between the microcontroller and the peripheral chips is accomplished through serial communications links.

Serial buses take up less room on a PCB than parallel alternatives, and protocols are defined by robust standards, ensuring good interoperability. Some examples of peripheral chips that use serial links include sensors / signal conditioners, conversion devices like ADCs and DACs, communications adapters, external memory, real-time clocks and display subsystems.

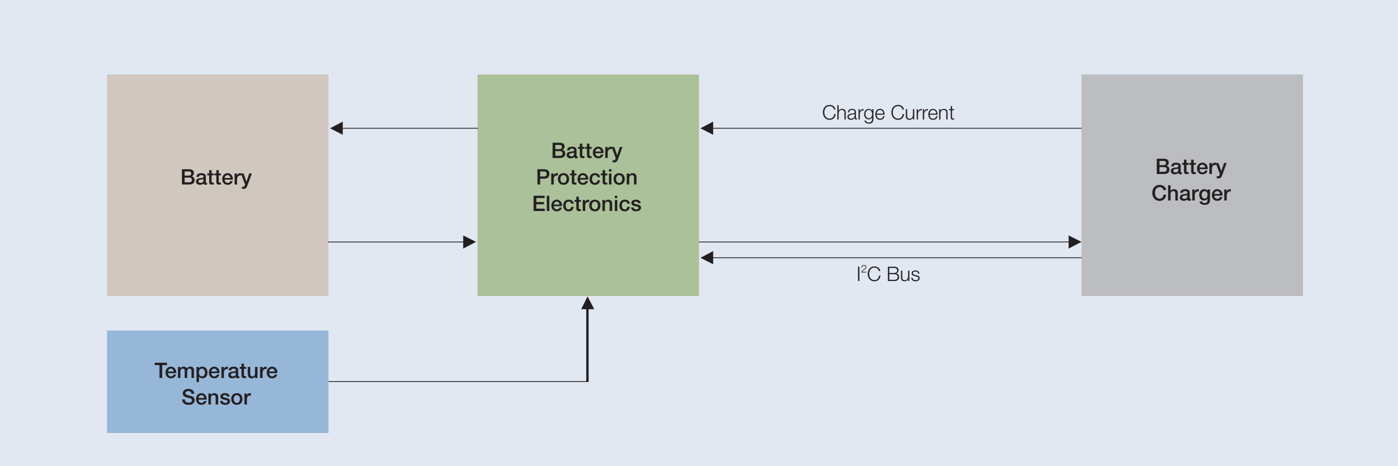

Figure 1.1. Block diagram of the battery charger system.

Before you have access to working firmware, you may want to confirm that a peripheral chip is working as expected. This can be done by correlating peripheral chip inputs or outputs with serial bus data being sent between the chip and the microcontroller. But verifying that serial buses are working and sending the correct data can be difficult with the traditional test solutions. This applications note describes three applications in which specific measurement techniques were used to correlate signals on peripheral chips with serial bus data.

Application 1. Serial Communications Controlling a Battery Charger

The charging of Lithium Ion batteries is very critical. Overcharging can result in damage to the battery and even a fire. Lithium Ion batteries have protection electronics embedded in the battery pack to monitor cell voltages and temperature as well as current and state of charge. Some versions of the protection electronics provide a means of communicating to the charger and/or the load to allow access to this the information. For a charger, knowing the battery temperature and individual cell voltages can help prevent over charging or other unsafe conditions. The same information can be used by the load device to indicate to the user when the battery is nearing the point of no longer supporting the load. The most common communications scheme used is referred to as “SMBus” which uses the I^2C protocol. In this example, the charger will periodically interrogate the battery temperature (and the cell voltages as well) using the I^2C bus. We will then look at the response of the battery protection electronics and the charger to a new analog temperature input. Figure 1.1 shows the configuration of the battery, cell electronics and charger. This charger is specially designed to adjust the charge current to prevent damage to the battery if the battery temperature is below a threshold.

The challenge in this example is to measure the response of the charger to a change in the battery temperature. Using the MSO4000 Series mixed signal oscilloscope, we are interested in finding the:

- Time from a change in temperature to when the data is available via the I^2C bus.
- Time from when the data is available to when the current starts to adjust.
- Time and stability of the current adjustment.
- Actual current values before and after the adjustment.

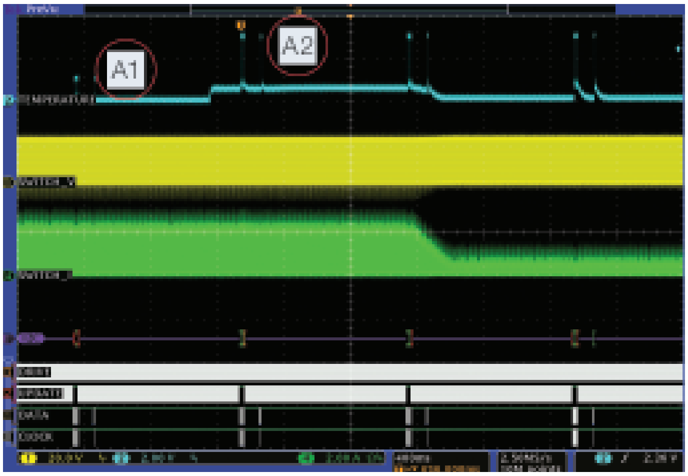

Figure 1.2. Operation of the charger over a period of 4 seconds.

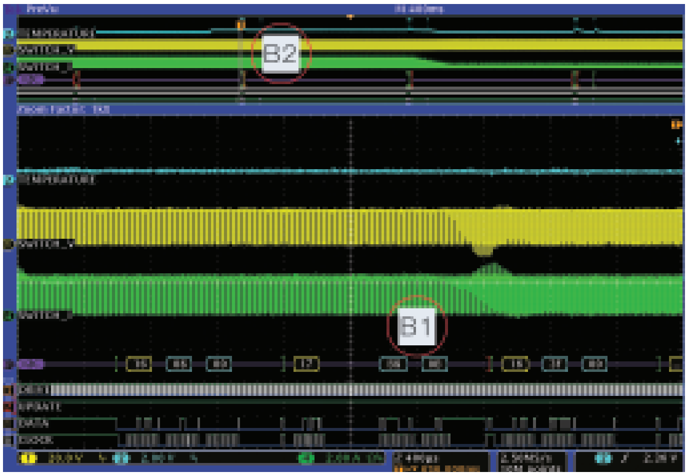

Figure 1.3. Decoded temperature reading magnified from the original capture shows detailed operation of the I²C bus and the decoded data.

Figure 1.2 shows the operation of the charger over a period of 4 seconds to show the response to a temperature change.

- The top trace labeled "TEMPERATURE" shows the sampled analog temperature reading at the battery.
- The next two analog traces labeled "SWITCH_V" and "SWITCH_I" show the power transistor's voltage and current respectively, which provide the pulse width modulated control of the charge current.
- The decoded I²C communications bus is monitored by two digital lines labeled "DATA" and "CLOCK" at the bottom of the screen.The decoded data from the I²C bus labeled I²C is displayed below the current trace simplifying debug of the communications stream.
- The digital signal to the gate of the power transistor marked "DRIVE" is also captured to allow confirmation of the power control operation.
- In addition, a digital signal marked "UPDATE" is programmed into the microcontroller to mark when the charger makes a current control correction so that the system response can be measured.

Thus we are simultaneously monitoring the stimulation (Temperature), the communications (I²C), and the effect on the charger switching power supply (Voltage and Current). We are also using the technique of inserting an additional monitoring line in the control loop (Update).

The first step from this trace is to determine when the change in temperature which occurs between the positions marked at A1 and A2 is read by the charger. The temperature is read by the battery monitoring electronics during the short pulses on the "Temperature" trace. The higher temperature reading is shown at A1 (a thermistor is used so the higher temperature is represented by a lower resistance and corresponding lower voltage). Position A2 is where the lower temperature value is first read. The I²C bus is read at the time of the longer pulse on the Temperature trace; also shown as activity on the Clock and Data digital traces.

Figure 1.3 shows the decoded temperature reading magnified from the original capture to show the detailed operation of the I²C bus and the decoded data. Here the two bytes after the read (0x17) command at the position marked B1 show the value 0x0BB8 (low byte first). This is the original temperature value. The location of the bus decode is shown at B2 in the upper portion of the screen so the engineer can always maintain the perspective of the total trace time. Since this I²C reading is taken before the new value of temperature is measured by the battery protection electronics, we would not expect the data to have changed yet.

Figure 1.4 shows the I^{2}C bus and decode at the next sample which occurs after the temperature change is measured by the battery protection electronics. Now the temperature value read after the (0x17) command at C1 has the value of 0x08D4 representing a change in temperature. This is the first I^{2}C data reading taken after the temperature reading was taken by the battery protection electronics, so the response is as expected. Note that the perspective on the total time frame is maintained in the upper portion of the screen. This magnified display is taken at position C2.

Figure 1.5 shows the period immediately after the I^{2}C reading taken in Figure 1.4 magnified to show the gradual current adjustment. The markers for the current control process are shown in the trace marked “Update”. At each “Update” pulse, there is an incremental correction to the current until the current reaches the new lower value that is required by the lower temperature. The smooth adjustment from the higher current to the lower level can be seen as well as the very short interval (about 10 milliseconds) between the I^{2}C communications event near the left of the trace at D1 to the start of the ramp down of the current. The current is adjusted in about 200 milliseconds after the new temperature reading is received by the charger. The value of the current before and after the change can be measured as well as it drops from about 2 Amps to about 0.25 Amps.

Figure 1.6 shows the benefit of deep memory in this oscilloscope. The same trace can be further magnified to show the detailed current and voltage wave forms during the operation of the switching power supply charge control. A current probe is used to measure the current in the power transistor. The drive signal is shown at the same time as the analog signals so that the drive delay and the switching times can be measured. This trace also allows measurement of the current ripple and indicates that the power supply inductor ripple is well controlled during the transition from the higher current the new lower current level.

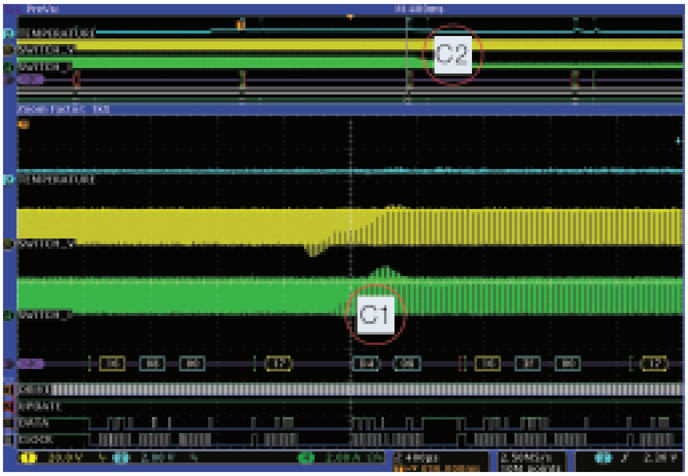

Figure 1.4. The I^{2}C bus and decode at the next sample which occurs after the temperature change is measured by the battery protection electronics.

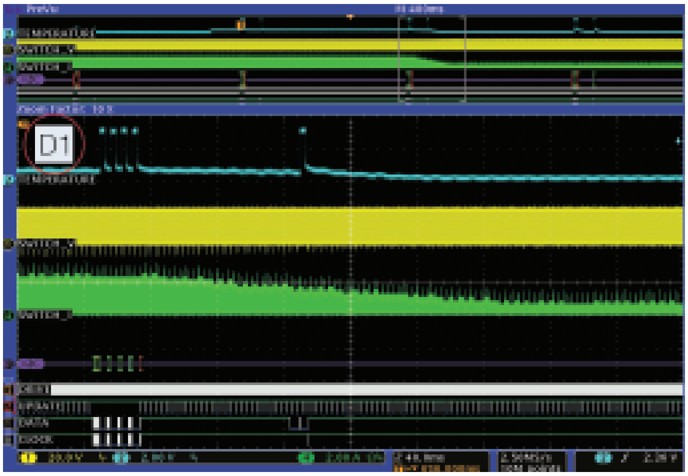

Figure 1.5. Magnified portion immediately after the I^{2}C reading shows the gradual current adjustment.

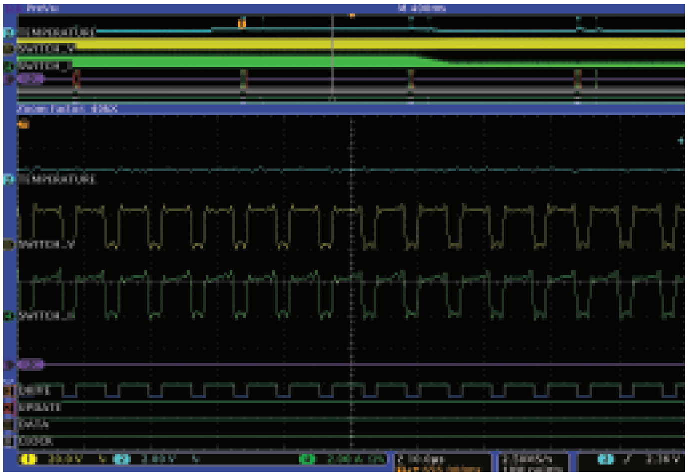

Figure 1.6. Deep memory enables the same trace to further magnified, thus showing the detailed current and voltage wave forms during the operation of the switching power supply charge control.

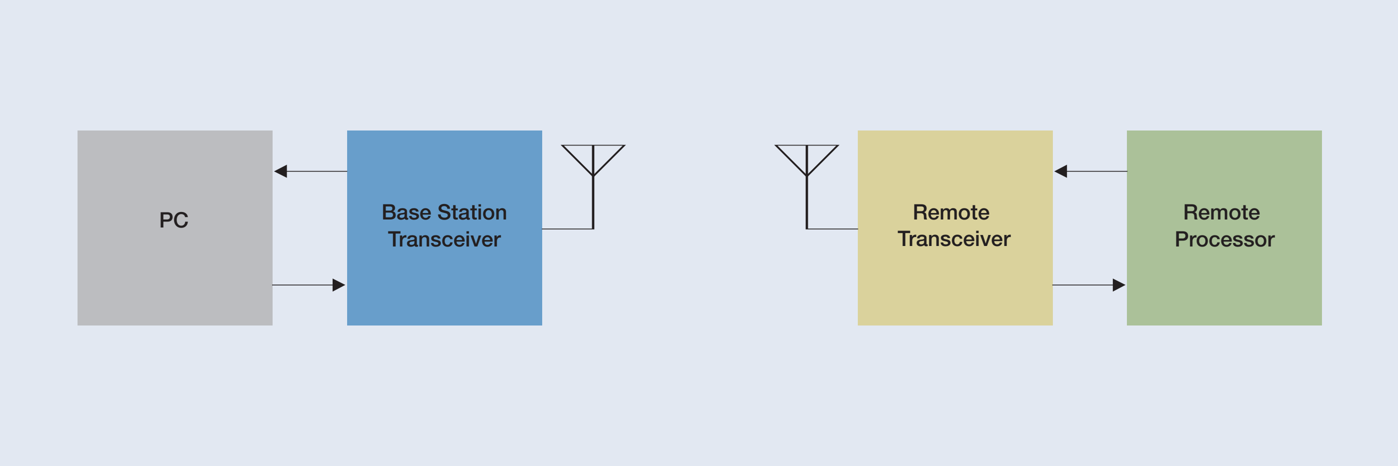

Figure 2.1. Block diagram of the two way radio system.

Thus, we have observed that the battery protection electronics correctly captured the change in temperature, the charger was able to read this temperature change, and that the charger was able to promptly and smoothly adjust the charge current to respond to the change in temperature to reduce the charge current. All of the information to measure the time intervals and current response was captured in one trace from which the events in the control system could be evaluated by the engineer as needed.

Application 2. Short Range Data Radio Communications

In this example, we will confirm the correct packet transmission and the timing of each stage of a two-way radio communications system. Figure 2.1 shows the PC communicating to a base station using RS-232 serial communications. The base station then communicates to a remote unit using a proprietary radio protocol. We will use the RS-232 decode capability to look at the message sent from the PC to the base station and from the base station to the PC. We will use the analog and digital channels to look at the radio transmitted and received signals on both the base station and remote unit. The radio scheme used here is AM to allow very simple circuitry on the remote side.

The packet structure consists of eight bytes as follows:

- Header byte = 0x21 for commands (0x28 is used for long data packets not used in this example)
- Two serial number bytes = 0x0000 for the base station and any other value for remotes
- Command byte = 0x30 to 0x3F
- Two argument or data bytes = 0x0000 to 0xFFFF with the low byte first
- Two checksum bytes = the sum of the first 6 bytes with the low byte first

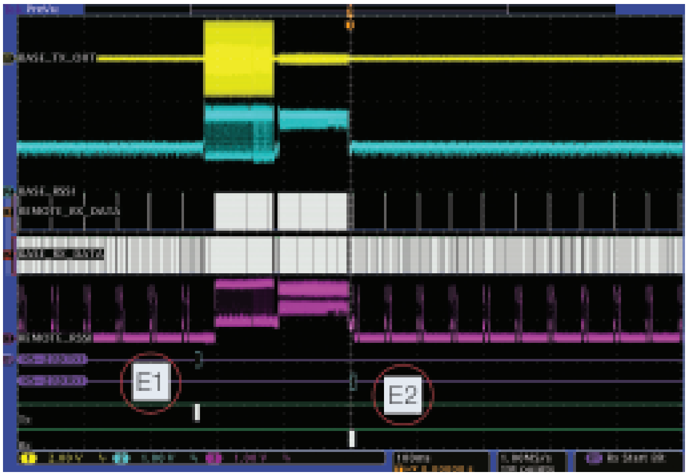

Figure 2.2. Full communications cycle from the PC back to the PC.

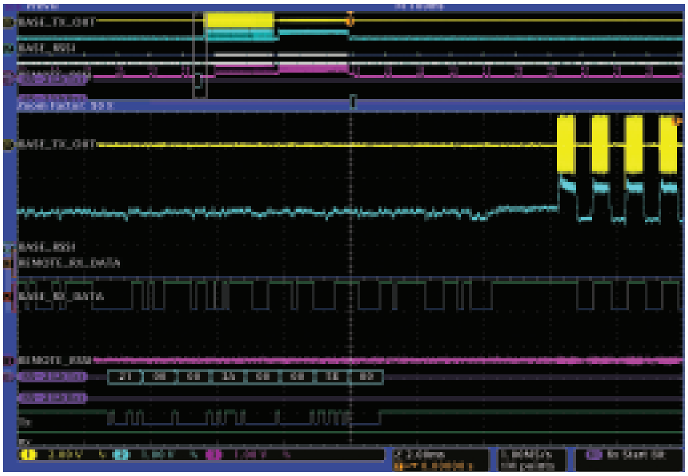

Figure 2.3. Decode of the initial RS-232 from the PC.

Figure 2.2 shows the entire transaction from the RS-232 signal being sent to the base station at E1 to the return message at E2. The RS-232 signals are captured by two digital lines labeled “Tx” and “Rx”. The corresponding decoded information is displayed on the lines labeled “RS-232(Tx)” and “RS-232(Rx)”. The actual decoded signals will be shown in later figures. The yellow trace labeled “BASE_TX_OUT” is the analog transmitted signal. The blue trace labeled “BASE_RSSI” is the base station received analog signal. This signal’s digital version is captured by digital signal labeled “BASE_RX_DATA”. On the remote side, the purple signal labeled “REMOTE_RSSI” shows the analog version of the received signal, and the digital signal labeled “REMOTE_RX_DATA” is the digital version of this signal. As can be seen from this capture, the entire transaction takes about 230 milliseconds for a full command packet and the return data. Note that the capture is triggered by the start bit of the return RS-232 signal to assure that the entire transaction is captured.

In Figure 2.3 the capture is magnified to show the decoded version of the data transmitted from the PC to the base station. With this decode, it is easy to check the data to confirm that the correct packet is being sent, as is the case. Note that the gray zoom window in the upper section of the screen shows the area being displayed in the magnified view. The decoded data shows that the packet is correctly formed with eight bytes. The first byte correctly reflects the command 0x21. Since this is the base station we expect the address to be 0x0000 which it is. The command 0x3A is the request for the battery voltage. Since this is a request, there is no argument so the bytes are zeros. The checksum of 0x5B is the correct sum of the previous bytes.

Figure 2.4 shows the base station transmitted waveform after most of the wakeup preamble has been sent. The data is represented by short and long periods of carrier signal. Note that the receiver is showing the exact same pattern in both the analog (REMOTE_RSSI) and digital versions (REMOTE_RX_DATA) as we expect.

In Figure 2.5, the remote is responding. Note that the time from receipt of the message to the transmission of the response is only about 10 milliseconds. The return message is shown by the base station receiver analog signal (BASE_RSSI) and the digital version (BASE_RX_DATA). The return RS-232 signal is also visible and only has a delay of about 5 milliseconds after the receipt of the message from the remote.

In Figure 2.6, the decoded RS-232 message is shown, and can be used to confirm that the return data is what is expected. Again the decoded data shows that the packet is correctly formed with eight bytes. The first byte is again the command header of 0x21. The address is 0xFFFF which is a legitimate value. The command byte of 0x3A is echoed so that the base station can confirm that the response is due to the command sent and not a misread command. The data value returned is 0x12F which corresponds to 2.5 Volts, which is what was applied. The checksum is the correct sum of the first 6 bytes: 0x21 + 0xFF + 0xFF + 0x3A + 0x2F + 0x01 = 0x0289.

We have used the RS-232 data decode feature of the oscilloscope to confirm the correct two-way communications in this radio system. The packets are correctly formed and the header and command values and the checksums are correct. We have been able to show the transmitted and received analog and digital information in one capture and the ability to magnify the captured data to look at the analog signal quality and the recovered radio data. We have also confirmed that the receiver response time is in the range we expected and the entire two way transaction takes about 250 milliseconds as expected.

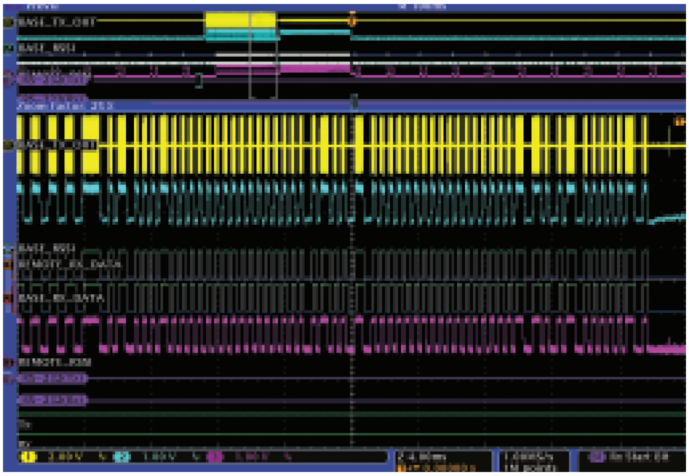

Figure 2.4. Base station transmitted waveform and remote receiver signals.

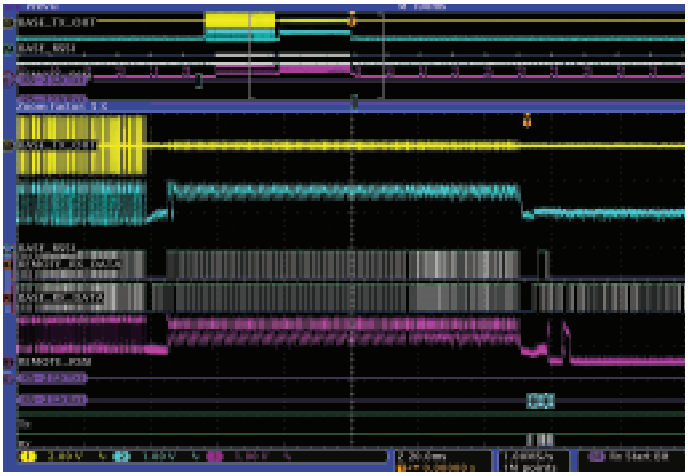

Figure 2.5. Return transmitted and received signals.

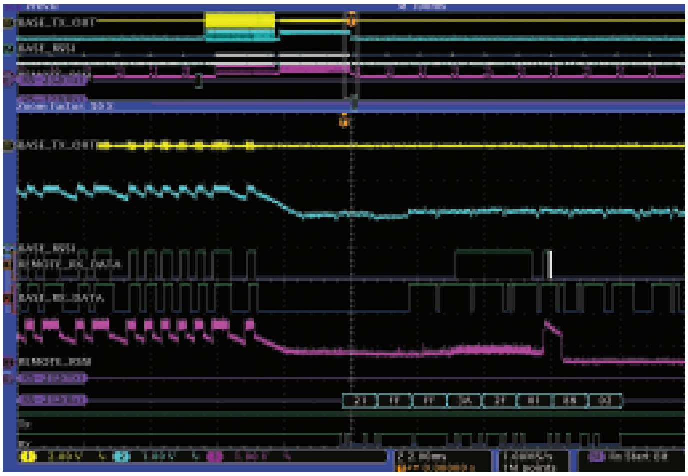

Figure 2.6. Decode of returned message to the PC.

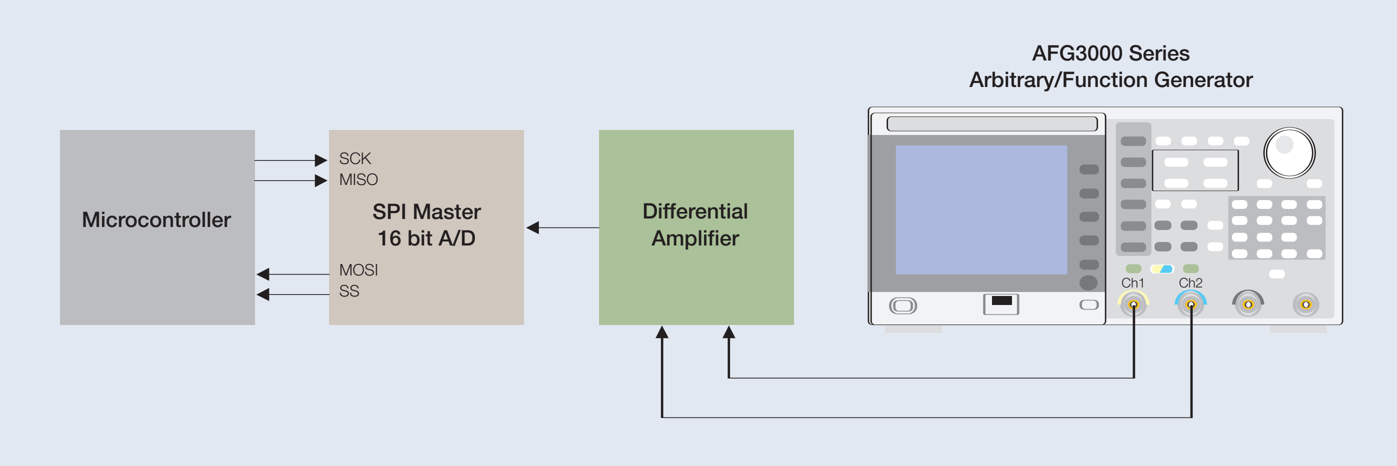

Figure 3.1. Block diagram of data acquisition subsystem.

Application 3. Evaluation of SPI-based A/D Converter

In this example we are investigating the performance of a 16-bit A/D converter that communicates using the SPI type serial bus. We are using both channels of a two channel signal generator, such as the AFG3000 Series to create a true differential input waveform, along with the SPI decode capability of the MSO4000 Series oscilloscope to view the data converted from the incoming signal. The data acquisition hardware consists of a differential input amplifier, the A/D converter, and a microcontroller as shown in Figure 3.1.

The objective of this measurement is to check that the A/D converter is set up properly as the SPI master, and that it is correctly reading the full input voltage range. We are using an arbitrary function generator to produce a full-scale input to the converter. The deep memory of the oscilloscope allows capturing multiple samples in one trace as shown in Figure 3.2. The SPI decode capability of the oscilloscope is used to read the value converted by the A/D. Subsequently this same acquisition can then be magnified to show samples at different places along the waveform and the decoded digital value produced by the A/D converter. In Figure 3.2, the yellow and blue traces labeled “IN+“ and “IN-” are the voltages from the signal generator. The red trace labeled “INPUT_DIFF” is the mathematical subtraction of the two input signals and represents the total input to the differential amplifier. The purple trace labeled “A/D_IN” is the actual input to the A/D converter after the differential amplifier. These analog traces allow confirming the correct operation of the analog circuits in front of the A/D. The digital signals labeled “CLOCK”, “DATA”, and “CONVERT” are the SPI control and data signals. The oscilloscope also provides decode of the data on the traces labeled “SPI(MOSI)” and “SPI(MISO)”.

In Figure 3.3, the trace is magnified at the bottom of the input voltage waveform. The location of the magnified trace can be seen in the original trace at the top of the screen. The data can be seen decoded on the “SPI(MOSI) line. The output is 0x00D8, a very low value which is in the range to be expected.

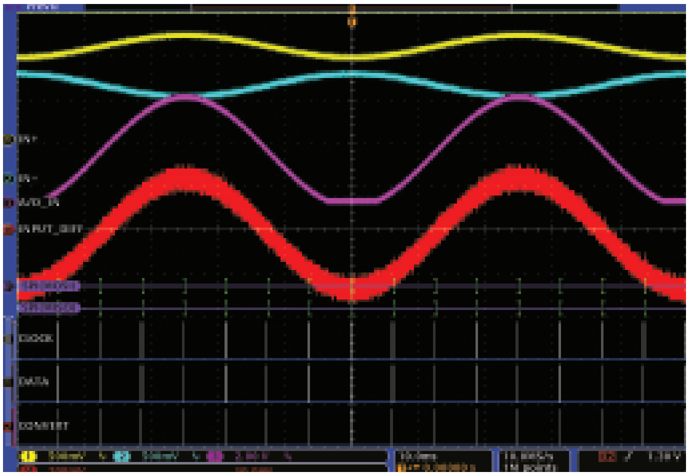

Figure 3.2. Multiple acquisitions over a sine wave input.

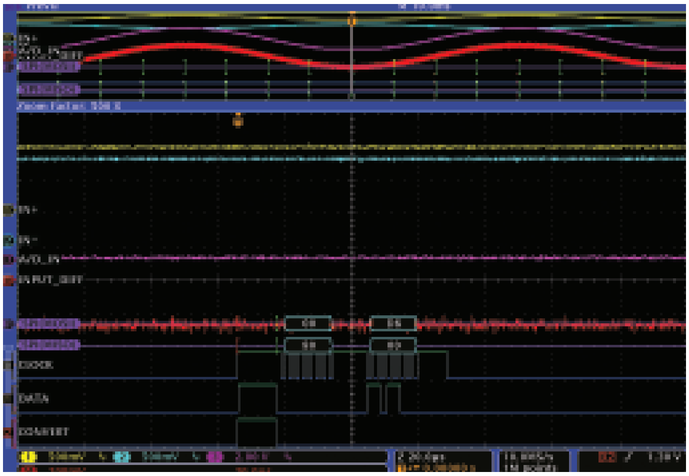

Figure 3.3. Data taken at the bottom of the waveform.

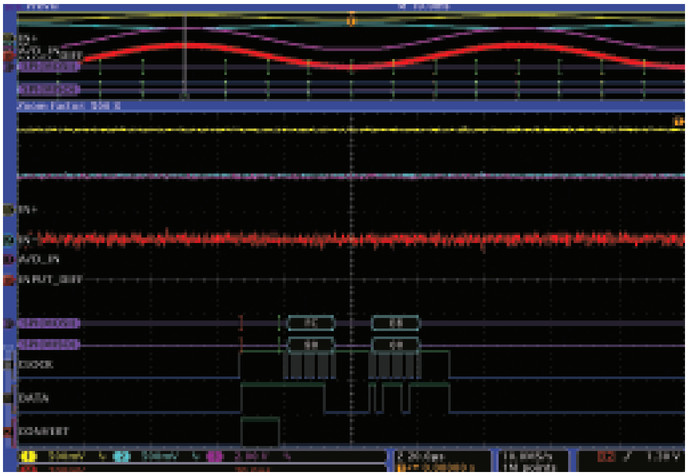

Figure 3.4. Data taken at the top of the waveform.

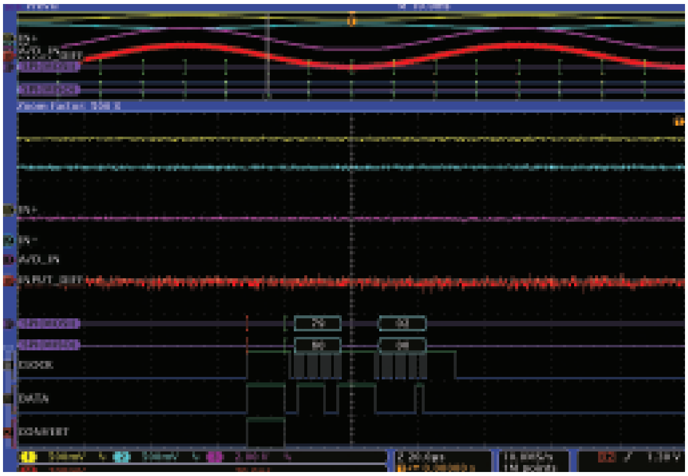

Figure 3.5. Data taken in the middle of the waveform.

In Figure 3.4, the magnified portion of the trace is shown near the top of the input voltage waveform. The decoded output value is 0xFCBB, which is close to the top of the 16 bits of A/D range, again as expected.

In Figure 3.5, the magnified portion of the trace is taken at a middle point in the input voltage waveform. The decoded output value is 0x7902, which is near the center of the A/D range.

In this example we have used the decoded digital channels to confirm that the A/D converter is correctly converting the input voltage over its full range. At the same time, the analog channels have been used to confirm that the input amplifier is working as expected. The deep memory of the oscilloscope facilitated this analysis from a single acquisition, and additional sample points could easily be checked to confirm that the digital output values correspond to the analog input at various points.

Summary

With an increasing number of designs utilizing serial communications links between the microcontroller and the peripheral chips, you need a proven test solution to help verify these serial buses are operating as expected. The specific measurement techniques highlighted in this application note demonstrate a few of the ways in which signals on peripheral chips can be correlated with serial bus data.

The Tektronix MSO4000 Series of oscilloscopes has deep memory to allow you to capture enough meaningful data, and can decode, trigger and search on serial bus data. The AFG3000 Series arbitrary/function generator's dual channel capability allows a completely independent selection of waveforms and frequencies and represents a big advantage in versatility.

Contact Tektronix:

ASEAN / Australasia	(65) 6356 3900
Austria	+41 52 675 3777
Balkan, Israel, South Africa and other ISE Countries	+41 52 675 3777
Belgium	07 81 60166
Brazil & South America	(11) 40669400
Canada	1 (800) 661-5625
Central East Europe, Ukraine and the Baltics	+41 52 675 3777
Central Europe & Greece	+41 52 675 3777
Denmark	+45 80 88 1401
Finland	+41 52 675 3777
France	+33 (0) 1 69 86 81 81
Germany	+49 (221) 94 77 400
Hong Kong	(852) 2585-6688
India	(91) 80-22275577
Italy	+39 (02) 25086 1
Japan	81 (3) 6714-3010
Luxembourg	+44 (0) 1344 392400
Mexico, Central America & Caribbean	52 (55) 5424700
Middle East, Asia and North Africa	+41 52 675 3777
The Netherlands	090 02 021797
Norway	800 16098
People's Republic of China	86 (10) 6235 1230
Poland	+41 52 675 3777
Portugal	80 08 12370
Republic of Korea	82 (2) 6917-5000
Russia & CIS	+7 (495) 7484900
South Africa	+27 11 206 8360
Spain	(+34) 901 988 054
Sweden	020 08 80371
Switzerland	+41 52 675 3777
Taiwan	886 (2) 2722-9622
United Kingdom & Eire	+44 (0) 1344 392400
USA	1 (800) 426-2200

For other areas contact Tektronix, Inc. at: 1 (503) 627-7111

Updated 17 October 2007

For Further Information

Tektronix maintains a comprehensive, constantly expanding collection of application notes, technical briefs and other resources to help engineers working on the cutting edge of technology. Please visit **www.tektronix.com**

11/07 DM 54W-21285-0

Tektronix®

Enabling Innovation