

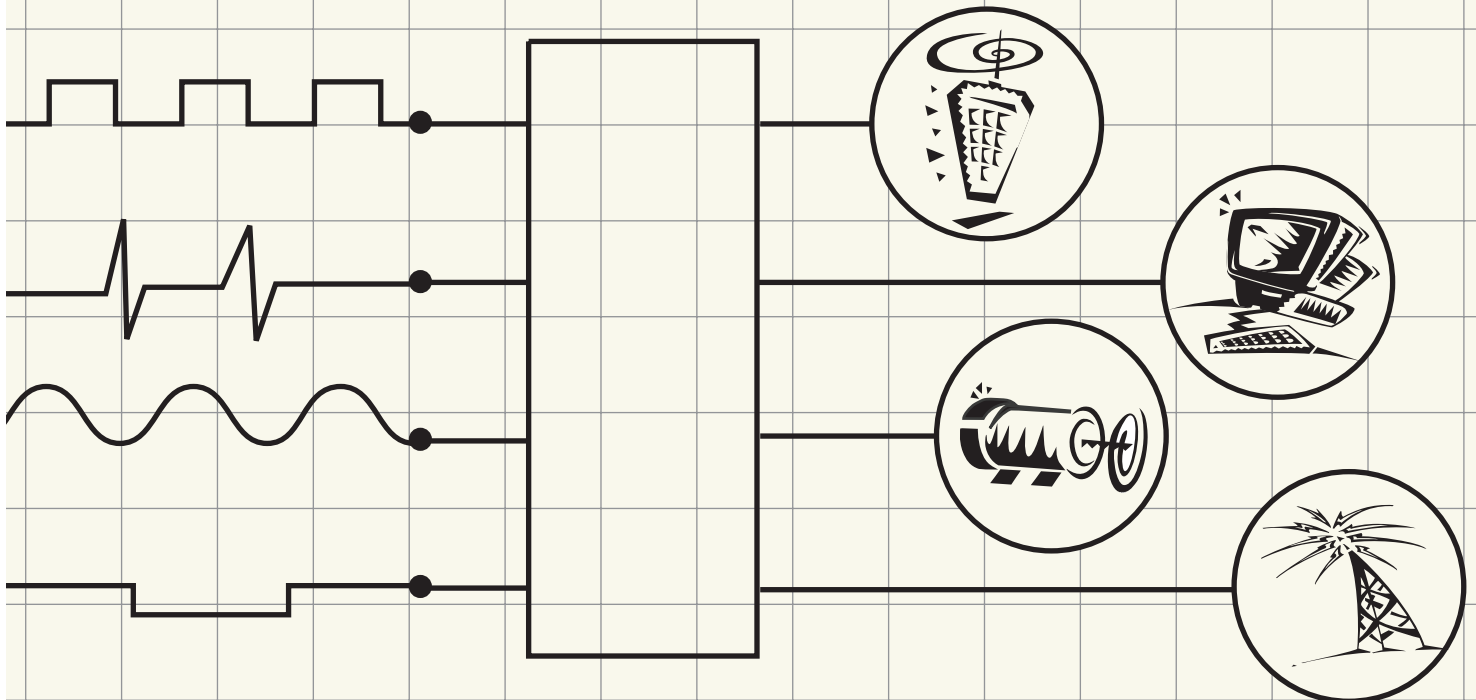
This literature was published years prior to the establishment of Agilent Technologies as a company independent from Hewlett-Packard and describes products or services now available through Agilent. It may also refer to products/services no longer supported by Agilent. We regret any inconvenience caused by obsolete information. For the latest information on Agilent's test and measurement products go to:

www.agilent.com/find/products

Or in the US, call Agilent Technologies at 1-800-452-4844 (8am-8pm EST)



Hints for Debugging Microcontroller-based Designs



Introduction

Contents

Hint 1: Tracking down elusive glitches. Using peak detect and deep memory to capture a tough glitch in a Motorola 68HC11K1-based digital radio transmitter.

Hint 2: Debugging serial I/O. Using deep memory to debug transmission problems in a digital wireless packet modem based on the Intel™ 80188 MCU.

Hint 3: Verifying PWM dead time in motor controllers. Using mixed analog and digital channels to verify proper signal timing in a Siemens C504-based system.

Hint 4: Isolating software problems with an advanced logic probe. Using one of the new generation of logic probes to debug a slot-car controller based on the Microchip PIC16C77.

Hint 5: Using arbitrary waveforms for bit-error testing. Using customized waveforms to test for bit errors in a Microchip PIC16C84-based digital data receiver.

Hint 6: Resolving hardware/software integration problems. Using a NOHAU 8031 emulator and a mixed signal oscilloscope to track down the anomalies that often plague hardware/software integration efforts.

Hint 7: Correlating software and analog outputs in a CAN controller. Using a combination of analog and digital measurements to debug program code that drives a Philips 80C51-based controller area network (CAN) system.

Hint 8: Debugging an MCU-based CCD camera controller. Using a combination of TV triggering, analog scope measurements and digital timing measurements on a Philips 80C552-based camera control system.

The challenge of debugging MCU-based designs

It's almost impossible to design an electronic or electromechanical product these days without using a microcontroller. While there are plenty of interesting design challenges, the debugging tools for MCU-based designs haven't always kept up.

If you work with 8- and 16-bit MCUs, for instance, you've probably felt stuck in the middle, between generic, basic tools (such as scopes) and higher-end tools aimed at microprocessors (such as traditional logic analyzers and emulators). At the same time, you're probably dealing with a mix of analog and digital signals, so a scope by itself or a logic analyzer by itself is only half a solution.

Moreover, you probably don't have the luxury of specializing. You have to know analog hardware, digital hardware and firmware—and be good at all three. And all the while, market windows are getting narrower, competition is getting stronger, and customers are expecting more power and capability from your MCU-based products. Your job may be a lot of things, but boring certainly isn't one of them.

Help is on the way

As the worldwide leader in test and measurement, we're working hard to help engineers like you meet the MCU challenge. One way we can help is with the information in this booklet, practical debugging hints from engineers working with a variety of MCUs. You'll see how these designers use some of the latest MCU debugging tools to get their new products on the market in less time.

Tracking down elusive glitches

HINT 1

By Steven Schram, Invocon, Inc.

Infrequent, unpredictable events can present some of the toughest troubleshooting challenges around. I recently encountered such a glitch while designing a low-power data acquisition device. This wireless instrument system uses a group of remote sensor units and spread spectrum radio transceivers (Figure 1). Data collected at the system can be retrieved by a network control unit connected to a computer.

The system uses the interrupt pin on a low-power clock chip to trigger power-up events every 60 seconds. Between events, the clock and supporting logic are the only devices drawing current (approximately 50 μ A). After getting a trigger from the clock, the Motorola 68HC11K1 microcontroller powers up, collects temperature data and listens for transceiver activity. If it hears a data request on the transceiver, the MCU transmits the temperature data.

The glitch in question was showing up during this 60-second interval, when the system was supposed to be quiet. To find and analyze this anomaly, I used a deep memory (1 Mbyte) digital oscilloscope with peak detect. Since the glitch occurred so infrequently, I first set the scope's time base at 10 seconds/division in order to capture the entire 60-second sequence. Without peak detect, most narrow events would be impossible to detect at this time base setting. But as Figure 2 shows, the glitch in my system was easily captured and viewed. Peak detect showed something unusual happening approximately 15 seconds after the clock trigger event.

Once I became aware of the anomaly's presence in my system, the scope's deep memory made it easy to zoom in and analyze the glitch in more detail. With the scope's 1 Mbyte of acquisition memory, the initial waveform capture at 10 seconds/division was also sufficient to see waveform details when I zoomed in and viewed at 10 milliseconds/division (Figure 3).

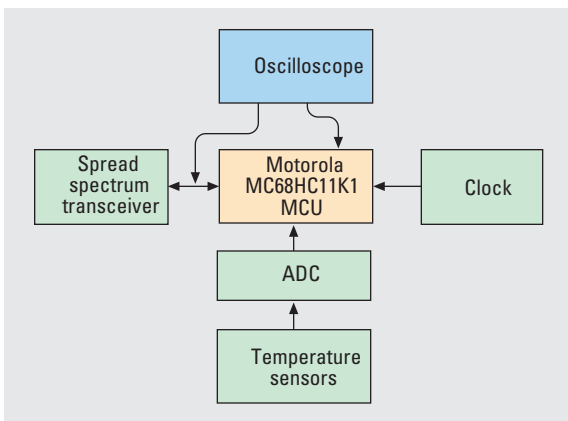


Figure 1: Basic block diagram of the wireless instrumentation system. One analog input on the scope monitored power-up on the MCU while the other monitored the carrier detect signal feeding the transceiver.

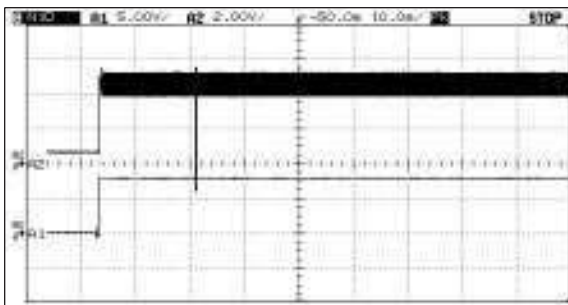


Figure 2: The initial measurement using peak detect with a 10 s/div time base setting.



Figure 3: Once the glitch was identified, zooming in to a faster time base setting provided the necessary details.

HINT

2

Debugging serial I/O

By Matthias Wandel, Research in Motion

When debugging serial I/O lines in an embedded system, a scope with deep memory can take the place of an expensive, complex serial protocol analyzer. I frequently use this technique when debugging the serial protocol that loads new firmware into our wireless modems and other devices.

In this particular case, I was debugging the serial link on our RIM900, a digital wireless packet modem that we market to companies building equipment for the Mobitex wireless network. This modem uses an Intel 80188 MCU and can communicate to other devices

using an RS-232 link. In this case, the RS-232 is connected to a PC.

The measurement setup is rather simple, as Figure 1 shows. I connected one input on my deep memory line scope to the transmit line and the other to the receive line.

I set the scope to roll mode, which shows continuous signal acquisition scrolling across the screen. Of course, it's important to make sure the digital scope's time base is set to a rate that will capture every bit transition.

Then I try to reproduce the problem. While waiting for the problem to appear, I keep one finger on the scope's stop key, then press it to freeze the acquisition when I see the problem. I've found that a reaction time of a half second or so is fast enough to capture the important parts of the signal. Thanks to its deep memory capability (1 million samples per channel), the scope captures the signal at high resolution so I can also go back and zoom in on the data to explore individual bit transitions if needed (Figure 2).

We've found that this simple technique has a number of advantages over using a serial protocol monitor, in addition to the time saved:

- Viewing activity from an analog perspective, so you can verify important signal parameters
- Verifying the baud rate using the scope's display cursors
- Checking to see if the hardware malfunctions partway through the transmission of a specific character
- Measuring software turnaround times precisely
- Identifying noisy conditions and crosstalk

In fact, this approach has worked after someone using a protocol analyzer couldn't make heads or tails of the situation.

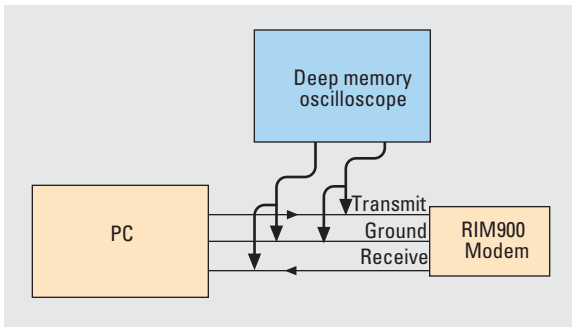


Figure 1: Block diagram of modem debugging setup.

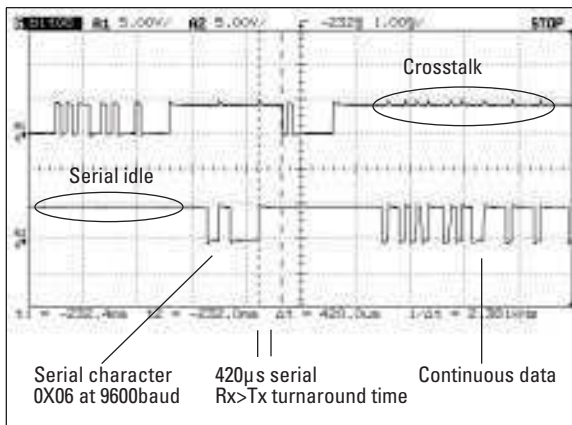


Figure 2: A scope display uncovers several important features of this serial bit stream.

Verifying PWM dead time in motor controllers

Technical staff, Microcontroller Group, Siemens Semiconductors

Generating pulse width modulated (PWM) signals with an MCU is a common way to control AC motors with sine-wave shaped currents. A typical application for an 8-bit MCU is controlling a three-phase induction drive with variable speed in an open-loop configuration.

However, the MCU can't drive an induction motor directly, so you need to amplify the three-phase signals first. Instead of using analog amplifiers, a more efficient way is to digitally amplify the PWM outputs with power switches, such as MOSFETs or IGBTs. The three-phase inverter shown in Figure 1 accomplishes this function.

The hardware for each phase of the inverter consists of two power switches (high side and low side) in a push-pull configuration. This creates a potential problem, though, if the control signals for the switches are exact complements of each other. During PWM switching, both power switches might momentarily conduct simultaneously due to different transistor turn-on and turn-off latencies. This can create a high-current short circuit and may destroy the inverter. It's therefore important to use an MCU optimized for motor control, such as the Siemens C504 (an 8051 derivative) or C164 (16-bit architecture). Both can be programmed to insert "dead time" in the PWM outputs by hardware without any software overhead. The dead time ensures that the two switches never conduct at the same time.

After programming the microcontroller to create the PWM output signals with dead time, the next step is testing the wave shape and timing. A four-channel scope can do the basic measurement, but if one is available, a mixed signal scope is a better choice because you can measure multiple analog and digital waveforms simultaneously and set up complex logic triggers.

Figure 2 verifies that the programmed dead time is sufficient for safe PWM switching. This zoomed-in display shows the impact of the dead time on the analog gate-source voltage of the power switch MOSFETs. The scope's cursors simplify the correct timing measurement and help characterize the circuit precisely.

With combined digital and analog measurement channels, you can easily monitor all six PWM signals and the phase currents. Figure 3 shows the two phase currents and corresponding digital PWM pattern. The time-qualified trigger mode lets you synchronize the scope's display to an adjustable pulse width corresponding to a well-defined phase angle.

HINT

3

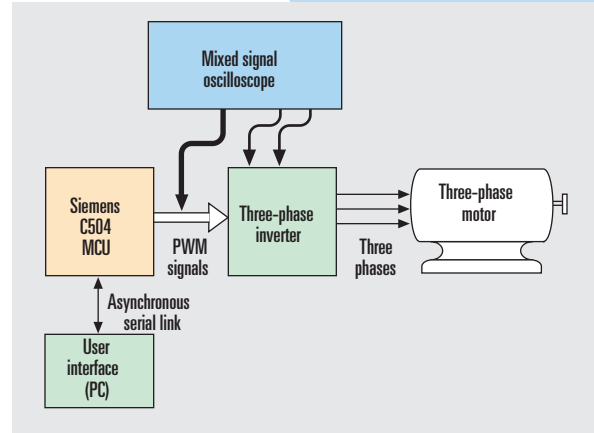


Figure 1: Block diagram of an open-loop configuration for generating safe PWM signals to drive a three-phase motor.

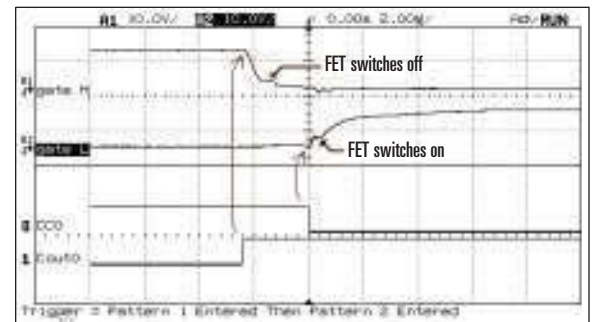


Figure 2: Verifying the dead time between the high-side and low-side PWM outputs.

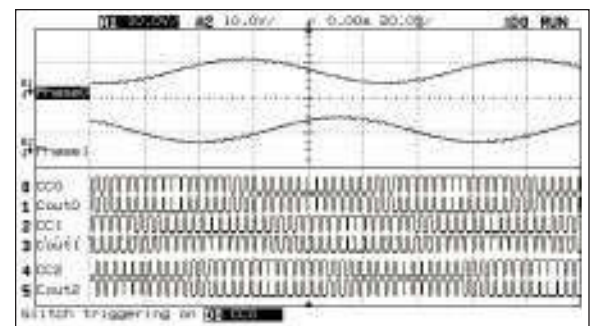


Figure 3: Monitoring all six PWM signals and the phase currents on the high-side and low-side switches.

HINT

4

Isolating software problems with an advanced logic probe

By Dave Brobst, Solutions Cubed

As consulting engineers for the embedded control market, we have the opportunity to work on a variety of interesting projects. While they range from inexpensive consumer devices to complex industrial systems, nearly all these designs involve some form of serial communication. Consequently, we deal with a number of different serial buses, including RS-232, I²C[™] and the new Universal Serial Bus. Troubleshooting these buses can be a challenge, thanks to the irregular waveforms we frequently encounter.

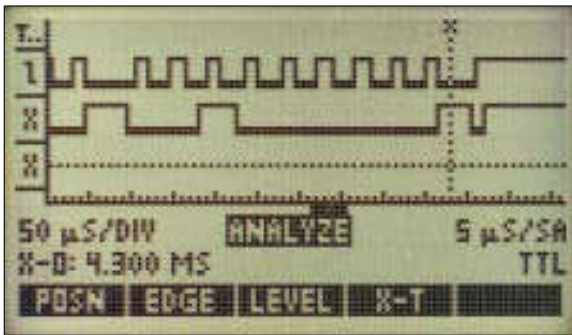


Figure 1: A good write sequence on the EEPROM.

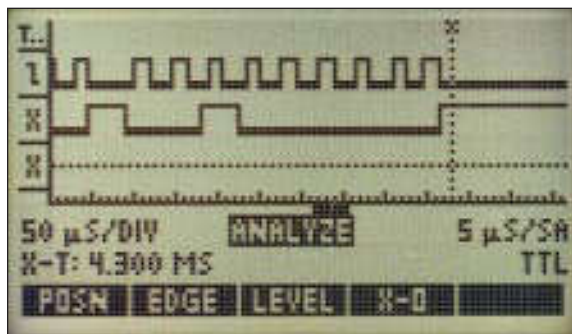


Figure 2: A bad write cycle, with the cursor highlighting the deviation from the expected signal.

One of our recent projects involved an advanced slot car control system we developed for Carolina Tracks Ltd. Based on the Microchip PIC16C77 8-bit MCU, this system reinvents the slot cars many of us remember from childhood. It aims to provide a realistic racing experience, complete with sound, displays, pit stops and simulated driving dynamics.

A key component in the system is an EEPROM that stores such data as high scores and the names of frequent players. The MCU communicates with this EEPROM over the I²C

bus, a bidirectional, multidevice bus created by Philips. This two-channel bus (a data line and a clock line) can support data rates up to 400 kHz.

Late in the design of the control system, the EEPROM began failing to write the data from its buffer to the EEPROM cells. Because the problem occurred after a software revision, we knew it was software related.

An advanced logic probe (the model I used is the HP LogicDart) proved to be the perfect debugging tool. With 2k samples of memory and three input channels, this probe could capture a full EEPROM write cycle. Using the analyze mode with an edge trigger on the clock line and single shot acquisition, I first collected a good write cycle for comparison (Figure 1). Then I captured a bad write cycle (Figure 2) and went to work.

Using the zoom and scroll features in the logic probe, I compared the good and bad waveforms until I found a difference. I discovered that we had inadvertently deleted a line of code during the last software revision. The result was that the MCU failed to send the stop bit, so the EEPROM didn't get the signal to transfer data from its buffer to the storage cells.

The size and location of the EEPROM chip would've made a scope or logic analyzer a cumbersome alternative. The advanced logic probe was a fast and painless way to find and fix the problem.

I²C is a trademark of Philips Corporation.

Using arbitrary waveforms for bit-error testing

By Jim Clark, LPA Design

As digital communications designers, we often need to test data reception software to make sure that bit errors are properly detected. Whether it's a wired or wire-

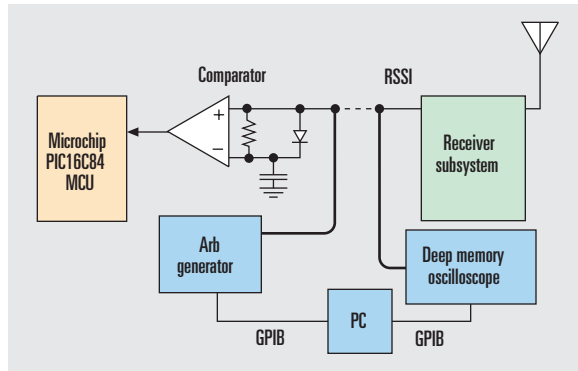


Figure 1: Block diagram of the On / Off Keyed data receiver.

less system, when we test over short distances on the lab bench, reception is usually too good to encounter random bit errors. When an error does occur, it is usually not very repeatable. This is where an arbitrary waveform generator comes in handy.

Digitizing a very clean and verified waveform that represents a digital data packet and storing it on a computer for replay with an arb generator opens up a whole new means of bit error testing. By capturing the clean waveform on a deep memory digital scope and transferring it to a PC using instrument software such as HP BenchLink or LabVIEW®, we can then change the waveform and introduce noise or other interference to fully verify proper reception on the device under test (DUT). We play the modified signal back to the DUT by downloading the new waveform into the arb generator and substituting this waveform for the original signal source.

Figure 1 shows the block diagram of an OOK (On / Off Keyed) data receiver we're designing. Figure 2 shows a clean (error free) packet of data that I captured from the received signal strength indicator (RSSI) output using a deep memory scope. I then uploaded this waveform to my PC using HP BenchLink software. Then it was a simple matter of cutting the waveform from the HP BenchLink software and pasting it directly into the arb editor. I saved this

waveform as my "perfect" original. I then used the math and edit tools in BenchLink to add random noise and reduce the amplitude to simu-

late path loss. The resulting signal was downloaded into the arb generator to generate a substitute RSSI signal (Figure 3).

The receiver is fed this new noisy signal to see if its software still recognizes the data. It's easy to go back and add more and more noise until bit errors occur. Then it's a simple matter to verify that errors are detected and/or corrected, depending on the receiver.

The arb generator also makes it possible to edit a specific bit or set of bits to make sure that errors in all positions are detected. Another key advantage is saving the modified waveforms to ensure a consistent signal from test to test.

LabVIEW® is a U.S. registered trademark of National Instruments Corporation.

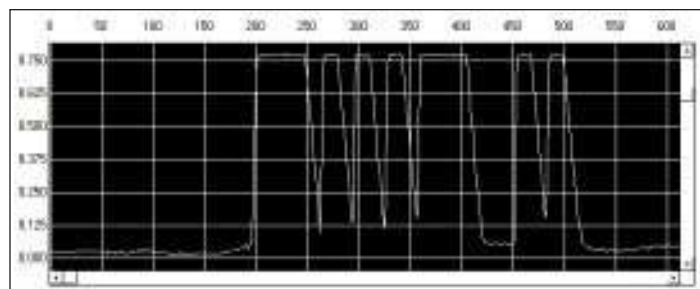


Figure 2: A clean waveform captured and transferred to the PC for editing.

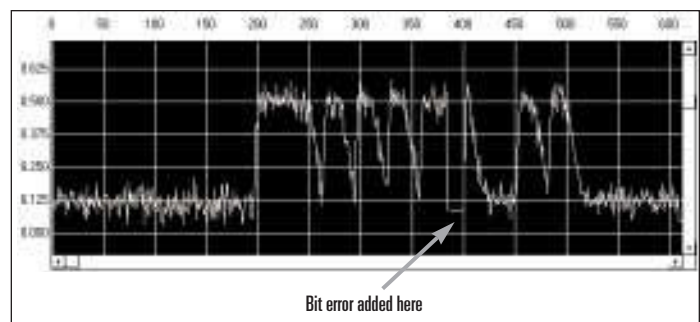


Figure 3: The edited waveform with noise and a bit error added to the clean original.

HINT

5

HINT

6

Resolving hardware/software integration problems

By Charlie Howard, Embedded Technologies Associates, Inc.

One of the most common problems in MCU debugging is figuring out whether an anomaly is based in hardware or in software. This can be tricky enough if a single person designs both, and it's magnified many times over if a team of designers is involved.

Traditionally, hardware designers use an oscilloscope and logic analyzer to prove it's a software problem, while software designers use an emulator to prove it's a hardware problem. Unfortunately, these one-sided methods often only reconfirm that the problem exists. What we really need is a

way to witness the problem as it occurs, while observing how the software and hardware behave and/or misbehave.

Tying a logic analyzer to an emulator can help, but this can involve a lot of configuration and connection work. A faster, easier alternative that's more than adequate for most MCU-based designs uses an emulator's trace and triggering capability to trigger a mixed signal oscilloscope. At the same time, the emulator selectively stores the suspect software instructions.

In one recent debug scenario, I used a NOHAU 8031 emulator with a mixed signal scope to explore some

trouble in the timing signals derived from the 8031 Port1 and an analog signal's relationship to these signals. Setup involved just three signals from the board to the emulator, one signal to the scope, and a trigger connection between the emulator and the scope.

As Figure 1 shows, the emulator captured the cycles in question while triggering the scope. (Note the time stamping.) The scope triggered on the fetch of the write to the port and captured the anomalous event (the slower transitions on lines P1-1 and P1-0) as well as the analog signal in question, indicating a hardware problem, as we can see from the scope display in Figure 2.

If this had been a software problem, I could've scrolled the trace buffer while synchronizing it to the source and program windows, making it easy to correlate program code to the error event. If there is more than one programmer writing to the same port, this method can save tremendous amounts of time and money by identifying the responsible software module. Plus, hardware engineers can continue to use the emulator's trigger-out and the second analog probe to isolate the cause of the problem further.

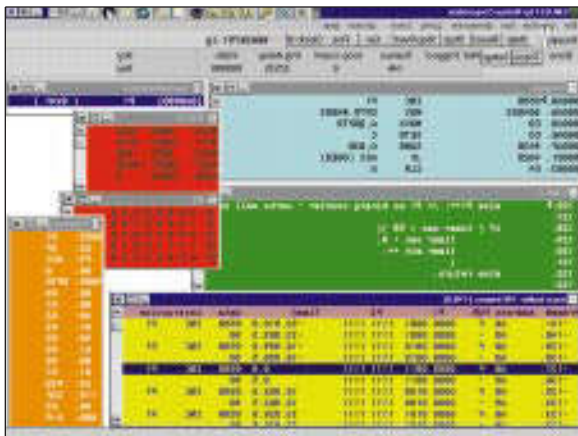


Figure 1: The emulator's trace buffer shows the point ($t=0$) at which the emulator triggers the scope.

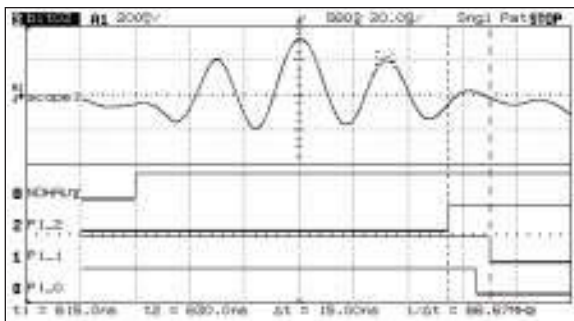


Figure 2: A detailed look with the scope shows a signal delay on lines P1-1 and P1-0.

Correlating software and analog outputs in a CAN controller

By Pascal Mestdagh, EUROCORPS, Telecommunications Division

Until recently, troubleshooting mixed signal designs, where you need exact time coherency between analog signals and MCU code, was extremely difficult. The problem could be partially solved by combining a logic analyzer and an oscilloscope with common time bases and triggering them simultaneously. However, time base differences between the two instruments could lead to incorrect results. Moreover, differences in memory made things even more difficult. An alternative is to use a hybrid scope/logic analyzer. These instruments enhance cross-domain measurement accuracy and can reduce debugging time for mixed-signal designs.

In my application, where a Philips 80C51 MCU interacts with an 82C200 CAN (Controller Area Network) control chip to establish low-speed data communication between several domotics (home automation) devices, it is not always easy to determine the cause of an emerging problem. In this specific case, problems arose when I tried to send data to a remote device. It seemed as though several bytes were not arriving at their destination.

I connected the digital inputs to the MCU data bus and connected the analog inputs to the transmission line (Figure 1). I then used pattern triggering to synchronize the measurement to the specific transmission request code word for the 82C200. Next, I set the trigger pattern in such a way that the measurement system triggered when the code word and the

desired transmission frame occurred simultaneously. I quickly discovered that I had a software problem and had to review the code. Contrary to my first assumption, the test revealed that a data loss existed between the MCU and the CAN controller, and not on the transmission line (Figure 2).

The integrated scope and logic channels made it possible to compare with great accuracy the analog signals with their digital originators (the MCU code). In addition, deep memory is a big plus, since it let me sample the full length of the transmission frame (approximately 300 ns) and at the same time have enough detail to investigate the micro-controller code (approximately 150 ns).

Although conventional test equipment probably could've solved this problem, I saved a considerable amount of time using a hybrid analog-digital solution.

HINT

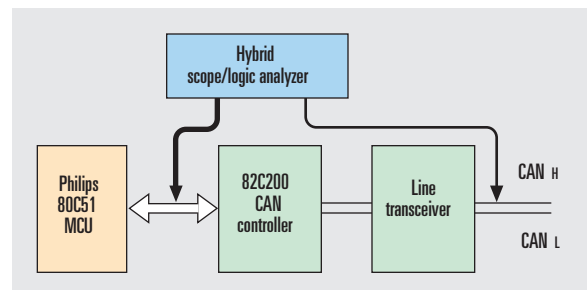


Figure 1: Measurement connections used to debug the CAN controller setup.

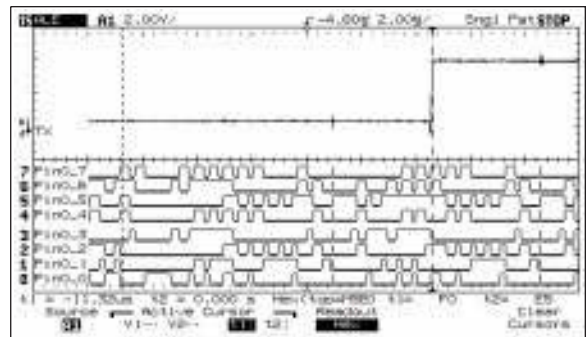


Figure 2: The simultaneous occurrence of the transmission request code word and the analog transmission frame revealed an inconsistency in my software code. Bytes did not arrive at their destination because the MCU didn't verify the "transmission complete" bit of the status register in the CAN controller.

HINT

8

Debugging an MCU-based CCD camera controller

by Jan Fischer, Petr Kocourek, and Petr Navratil

Like many MCU-based designs, the CCD camera systems we've been designing require simultaneous measurement of digital and analog signals, often with complex trigger requirements.

As Figure 1 shows, the horizontal synchronization pulses are first separated from the video signal. Using these Hsync pulses, the phase lock loop (PLL) generates the 10 MHz ADCLK signal. The falling edge of ADCLK samples the TV signal into the ADC; its rising edge updates the ADC output. The programmable logic device (PLD) converts ADCLK to generate the \overline{WR} signal. The \overline{WR} rising edge writes the data from the ADC into

the FIFO memory, which then contains a digitized signal of one TV line. The Philips 80C552 MCU reads data from the FIFO and calculates the feedback control data for camera positioning and zooming. Systems like this are commonly used in applications that need to track and measure objects visually, such as navigation and non-contact measurement.

Using the single-shot TV trigger mode and Autostore function of our mixed signal scope (MSO), we discovered a 25 ns edge instability or jitter on the ADCLK signal. With the MSO, we easily captured and stored a complete 20 ms stream of one-half of the TV picture at 50 MSa/s for further processing and analysis (Figure 2).

Figure 3 shows the critical timing of writing the data from ADC into the memory; 5 ns was not enough time for the memory to store the data. This problem, which was impossible to find with a conventional scope, was obvious when we measured with a hybrid scope/logic analyzer. Using these results, we were able to reprogram the PLD to avoid the problem.

The ability to trigger the MSO with a TV signal made it easier to debug the MCU software routine. The combination of analog and digital acquisition gave us a complete view of some rather complex behavior in our design.

Jan Fischer and Petr Kocourek are with Czech Technical University and Petr Navratil is with T&M Direct.

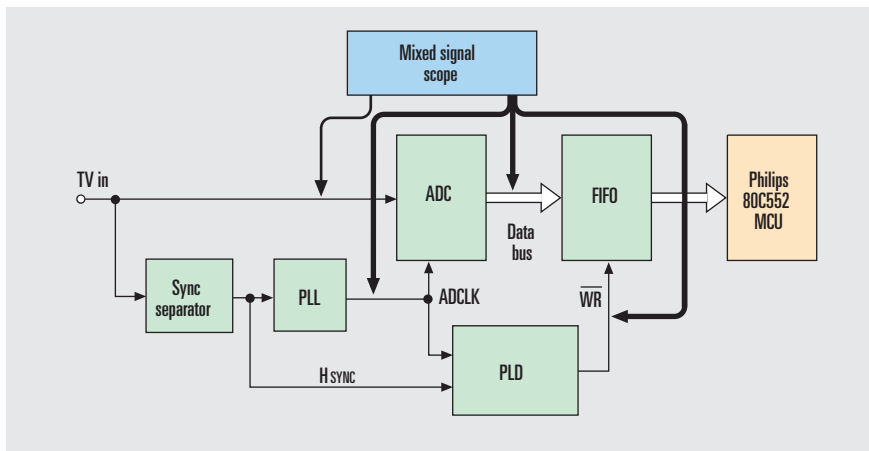


Figure 1: Block diagram of the CCD camera controller showing analog and digital test connections.

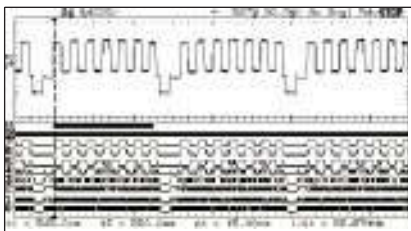


Figure 2: The analog output of the CCD camera and the relevant digital signals in the control system. The output of the ADC is on lines 0-7, the ADCLK signal is on line 10, and the \overline{WR} signal is on line 11.

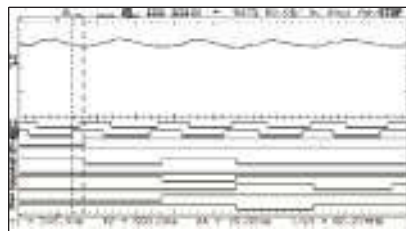


Figure 3: The display markers highlight the timing relationship between ADCLK and \overline{WR} .

HP 54645D Mixed Signal Oscilloscope	
Total channels	2 scope + 16 logic
Scope channels	
Bandwidth	100 MHz (75 MHz @ < 10 mV/div)
Number of channels	2
Maximum sample rate	200 MSa/s
Memory depth	1 M points/channel
Peak detect	5 ns minimum
Input impedance	1 M Ω , 13 pF
Maximum input	400 V (dc + peak ac)
Range	1 mV/div to 5 V/div
Resolution	8 bits
Coupling	ac, dc, ground
Logic channels	
Number of channels	16 (two 8-channel pods)
Maximum sample rate	400 MSa/s one pod only; 200 MSa/s both pods active
Memory depth	2 M points/channel one pod only; 1 M both pods active
Input R & C	100 k Ω , 8 pF
Input level	\pm 40 V max, 500 mVp-p min
Threshold range	\pm 6.0 volts in 50 mV increments
Predefined thresholds	TTL (1.4 V), CMOS (2.5 V), ECL (-1.3 V)
Peak detect	5 ns minimum
Timebase	
Range (main & delayed)	5 ns to 50 s/div
Accuracy (non-vernier ranges)	
Scope, same channel	\pm 0.01% of reading \pm 0.2% of screen width \pm 40 ps
Scope, channel to channel	\pm 0.01% of reading \pm 0.2% of screen width \pm 80 ps
Logic, same channel	\pm 0.01% of reading \pm 0.2% of screen width \pm (1 logic sample period, 2.5 or 5 ns) \pm chan-to-chan skew
Logic, channel to channel	\pm 0.01% of reading \pm 0.2% of screen width \pm (1 logic sample period, 2.5 or 5 ns) \pm chan-to-chan skew
Triggering	
Sources	All channels and line
Glitch triggering	Minimum width 8 ns. Operators: <, >, or range
Logic trigger modes	Edge, pattern, glitch, advanced pattern, TV Advanced pattern operators: And, Or, Then, Entered, Exited, Duration time, Duration >, Duration <
Power	100–240 Vac, 45–440 Hz, 90 VA
Weight	6.4 kg
Size (excl. handle)	172 mm H x 322 mm W x 317 mm D
Warranty	3 years
Ordering information	Options
HP 54645D Mixed signal oscilloscope Includes two scope probes (HP 10074), one logic cable (HP 54620-61601), power cord, and manual	101 HP 10098 Accessory pouch and front panel cover
HP 54645A 100 MHz 2-Channel oscilloscope with HP MegaZoom	103 HP 54654A Operator's training kit
HP 54650A HP-IB Interface module	1CM 5062-7345 Rack mount kit
HP 54652B RS-232/Parallel Interface module	106 HP 34810B BenchLink Scope software for Windows®
HP 54657A HP-IB Measurement/Storage module	W50 Additional 2-year warranty
HP 54659B RS-232/Parallel Measurement/ Storage module	
HP 1185A Carrying case	
HP E2613B Wedge 0.5 mm 3-signal probe tip adapter (qty 2)	
HP E2614B Wedge 0.5 mm 8-signal probe tip adapter (qty 1)	
HP E2615B Wedge 0.65 mm 3-signal probe tip adapter (qty 2)	

Windows® is a U.S. registered trademark of Microsoft Corporation.

The HP Wedge

The HP Wedge probe tip adapter solves the problem of connecting your scope or logic analyzer to fine-pitch thin quad flat pack (TQFP) and plastic quad flat pack (PQFP) surface-mount ICs. It works by inserting compressible dual conductors between adjacent IC legs. The flexible conductors conform to the size and shape of each leg to ensure tight contact. It's then a simple matter to connect your scope or logic analyzer to the HP Wedge.

The HP Wedge's unique design delivers secure, redundant contact on each leg, with no chance of shorting to adjacent legs. Plus, it's mechanically noninvasive, so it won't damage your device under test.

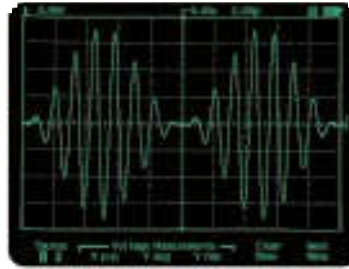
HP Wedges are available with 3- and 8-leg connections for 0.5 and 0.65 mm IC packages. Please call your local HP sales office for current availability.



The HP Wedge provides dependable, trouble-free connection to fine-pitch ICs.

HP 33120A Function/Arbitrary Waveform Generator

- Ten standard waveforms, with sine and square to 15 MHz
- Build arbitrary waveforms with 40 MSa/s speed and storage for four 16,000-point waveforms
- Clean signals: THD less than 0.04% and flatness as low as ±0.1 dB



Internal AM, FM, FSK, and burst modulation eliminate the need for a second modulation source.

The HP 33120A function/arb generator offers the rock-solid stability of digital synthesis at a price even your accounting department will feel good about.

And not only do you get a full complement of standard functions with better performance, you also get arbitrary waveforms. Just imagine the ways you could use complex custom waveforms (with 12-bit resolution), from simulating heartbeats and vibrations to testing MCU-based designs in ways never before possible.

Low cost doesn't have to mean messy harmonics and other extra baggage, either. Try to find another function/arb generator that matches the HP 33120A harmonic distortion specs at this price.

Sweep and modulation expand your test options without expanding your equipment list. Plus, you get full programmability using Standard Commands for Programmable Instruments (SCPI) with standard HP-IB and RS-232.

The Option 001 phase lock/timebase increases the HP 33120A frequency stability and creates new system possibilities. Generate precise phase-offset signals, phase lock two HP 33120As, or sync your generator to a 10 MHz frequency standard. Or you can even tie an entire ATE system to a master clock.

Waveforms	
Standard	Sine, square, triangle, ramp, noise, sin(x)/x, exponential rise and fall, cardiac, dc volts
Arbitrary	
Waveform length	8 to 16,000 points
Nonvolatile memory	Four waveforms (each from 8 to 16,000 points)
Amplitude resolution	12 bits
Sample rate	40 MSa/s
Frequency characteristics	
Sine	100 μ Hz–15 MHz
Square	100 μ Hz–15 MHz
Triangle	100 μ Hz–100 kHz
Ramp	100 μ Hz–100 kHz
White noise	10 MHz bandwidth
Resolution	10 μ Hz or 10 digits
Accuracy	10 ppm in 90 days (18 °C–28 °C)
Sinewave harmonic distortion	
dc to 20 kHz	–70 dBc
20 kHz to 100 kHz	–60 dBc
100 kHz to 1 MHz	–45 dBc
1 MHz to 15 MHz	–35 dBc
Amplitude characteristics	
Level (into 50 Ω)	50 mVp-p–10 Vp-p
(into open circuit)	100 mVp-p–20 Vp-p
Accuracy (at 1 kHz)	±1% of specified output
Modulation	
AM (int./ext.), FM (ext.), FSK (int./ext.), burst (int./ext./gate)	
Option 001 Phase Lock/TCXO Timebase	
Timebase accuracy	
Stability	±1 ppm 0 °C–50 °C
Aging	<2 ppm in first 20 days 0.1 ppm/month
External reference/ Input lock range	10 MHz ±50 Hz
Internal reference/ Output frequency	10 MHz
Power	
100 V/120 V/220 V/240 V	
Physical	
Net weight	4 kg (8.8 lbs)
Size	103.6 mm H x 254.4 mm W x 374 mm D (4.1 x 10.0 x 14.8 in)
Ordering information	
HP 33120A Function/Arb generator Opt. 001 Phase Lock/TCXO Timebase Opt. 106 HP BenchLink Arb software	



HP 54645D Mixed Signal Oscilloscope

- **Seamlessly integrates 2 scope channels and 16 logic channels**
- **Powerful triggering isolates important events and signal relationships**
- **HP's MegaZoom technology allows deep memory capture and a responsive display**



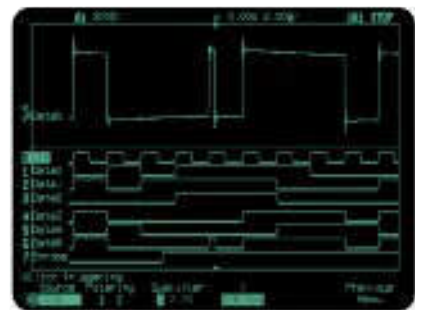
The HP 54645D mixed signal oscilloscope combines the detailed signal analysis of a scope with the multichannel timing measurements of a logic analyzer. Plus, it offers the exclusive HP MegaZoom for the benefits of deep memory without the usual

drawbacks of sluggish response and complex operation.

By being able to see both the analog and digital sides of a problem, you can analyze the signals and relationships that matter most. No more guesswork and no more poking around a few channels at a time.

Because the HP 54645D is built on a scope foundation, it looks and feels like a familiar scope—not like a complicated logic analyzer.

The HP 54645D can easily conquer MCU debugging problems that a normal scope can't begin to address, such as correlating a serial data sequence with variations in an analog signal or triggering on a mix of digital bus states and analog signal details. The combination of scope channels, logic channels and HP MegaZoom deep memory provides totally new ways to debug mixed-signal and MCU-based designs.



View circuit operation in ways you've never been able to see before.

HP LogicDart Advanced Logic Probe

Start with a scope. Then a logic probe. Then a multimeter. Switch back to the logic probe. Go borrow the logic analyzer. Every time you reach for a new instrument, you interrupt your thought process.



If you'd like to think more about your circuits and less about your equipment, pick up the HP LogicDart advanced logic probe. With one handheld tool, you can monitor logic activity on three channels, measure dc voltages, check frequencies, and even get timing diagrams with 10 ns resolution. You get both audio and visual feedback, and you can store up to 10 waveforms for later comparison.

The same probe input (small enough to handle fine-pitch IC leads) works for all measurements, so you can walk through a circuit without stopping to change equipment. By staying closer to your circuit, you'll get closer to the answer.

For more information about Hewlett-Packard test & measurement products, applications, services, and for a current sales office listing, visit our web site, <http://www.hp.com/go/tmdir>. You can also contact one of the following centers and ask for a test and measurement sales representative.

Austria:
Tel: (01) 25000-7171
Fax: (01) 25000-7172

Belgium:
Tel: 02 778 3413
Fax: 02 778 3414

Czech Republic:
Tel: 02 2435 5808
Fax: 02 2435 5809

Denmark:
Tel: 45 99 15 15
Fax: 45 82 06 30

Finland:
Tel: (09) 8872 2100
Fax: (09) 8872 2923

France:
Tel: (0) 1 69.29.41.14
Fax: (0) 1 69.82.65.09

Germany:
Tel: (0180) 524 6333
Fax: (0180) 524 6336

Greece:
Tel: 01 726 40 26
Fax: 01 726 40 20

Hungary:
Tel: 36-52 311 222
Fax: 36-52 311 222

Ireland:
Tel: 01 615 8222
Fax: 01 284 5134

Israel:
Tel: 03 53 80 377
Fax: 03 53 76 505

Italy:
Tel: 02.92.122.241
Fax: 02.92.104.069

Netherlands:
Tel: 020 547 6222
Fax: 020 547 7765

Norway:
Tel: 22 73 57 59
Fax: 22 73 56 19

Poland:
Tel: 48 22 360072
Fax: 48 22 360072

South Africa:
Tel: 011 806 1193
Fax: 011 806 1213

Spain:
Tel: (91) 631 13 23
Fax: (91) 631 14 69

Sweden:
Tel: 08 444 2277
Fax: 08 444 2525

Switzerland (German speaking):
Tel: 01-735 72 00
Fax: 01-735 72 90

Switzerland (French speaking):
Tel: 022-780 44 85
Fax: 022-780 44 95

UK:
Tel: 01344-366663
Fax: 01344-362852

For countries not listed, contact Hewlett-Packard International Sales Europe Geneva, Switzerland
Tel: (41-22) 780 8111 Fax: (41-22) 780 8542.

HP LogicDart Advanced Logic Probe	
Logic families	TTL, 5 V CMOS, 3.3 V CMOS, ECL, USER 1, USER 2
Logic monitor	100 MSA/s sample rate, 15ns glitch detect, LEDs plus beeper
Timing analyzer	3 channels, 100 MSA/s sample rate 2048 samples/channel, with pan and zoom 10 ns/div to 20 s/div
Triggering	Edge, pattern, edge/pattern combination 15 ns trigger glitch detect
dc voltage	±35 V (3 1/2 digits), ±(0.5% of reading + 2 counts)
Frequency	1 Hz to 33.0 MHz
Resistance	0.01 K to 200 k
Continuity threshold	80 minimum, 140 typical
Time & date	Real-time clock, user-selected format
Languages	Selectable in English, German, French, Spanish, and Italian
Power	(3) 1.5 V AA alkaline or lithium batteries ac adapter included
Physical	8.9 x 19.8 x 3.8 cm, 0.4kg
Ordering information:	
HP E2310A Advanced Logic Probe Includes carrying case, probe kit, User's Guide, name plate, ac adapter, batteries, and Certificate of Calibration HP E2320A Assembled probe with browser HP E2321A Replacement probe HP E2322A Probe accessory kit HP 82240B Portable thermal printer	

Copyright ©1998
Hewlett-Packard Company

Printed in the Netherlands 5/98
5966-3688EE