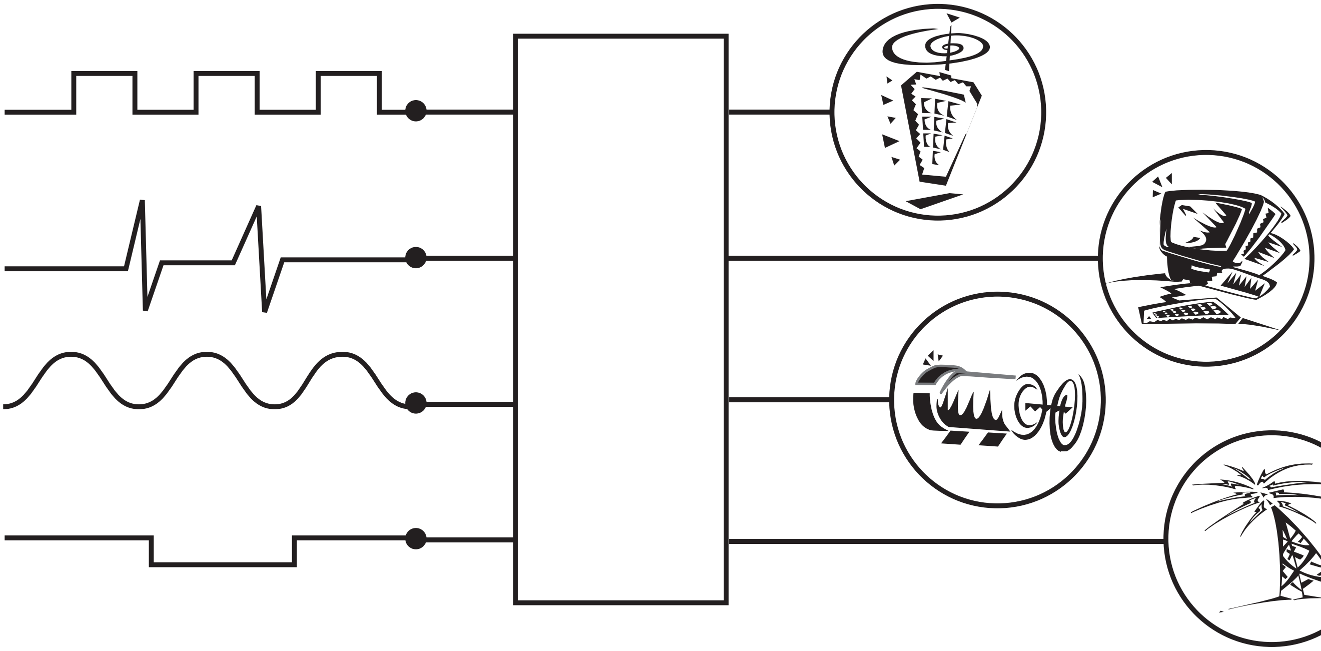
Time-saving tips for Siemens
C166 and C500 microcontrollers

HEWLETT® PACKARD
SIEMENS

8 Hints for Debugging Siemens MCU-based Designs



Introduction

Contents

Filling the gap in MCU debugging tools

Klaus Hommann

Klaus Hommann is a technical consultant and sales advisor for Willert Software Tools, a supplier of Siemens MCU development tools in Bueckeburg, Germany.

The steadily decreasing size and steadily increasing speed and capability of advanced microcontrollers, such as the Siemens C500/166 family of MCUs, give designers some impressive options for designing new products. On the downside, these new MCUs also give designers some advanced headaches during testing and system integration.

Traditional tools aren't always up to the task

To complicate matters, the debugging tools most designers have at their disposal aren't providing enough support. Emulators and ROM monitors won't work unless the MCU is up and running, so they're not much help if you're struggling with power lines, clock signals, drivers, missing pull-up resistors and other common prototype problems.

And even when you can operate them, emulators and ROM monitors can't address many MCU debugging problems, including interference between analog and digital signal lines on the printed circuit board and on most anything involving analog I/O signals.

Many engineers have relied on oscilloscopes for debugging MCU-based designs, but the move from 4-bit to 8- and 16-bit MCUs is leaving scopes behind. Plus, scopes can't offer the complex triggering you often need in MCU debugging. On the other hand, logic analyzers can handle these newer chips, of course, but they can't always handle the analog signals and they usually offer more power than you really need for most C500/166–based designs.

The best of both worlds

A new type of instrument, the mixed signal oscilloscope (MSO), closes the gap in MCU debugging tools. The HP 54645D from Hewlett-Packard (see inside back cover) combines two analog scope channels with 16 digital logic channels, so you can monitor analog and digital lines at the same time. This MSO offers more powerful triggering than a scope, including the ability to define pattern triggers across both analog and digital lines. Of course, you can also trigger it from your emulator if you need to synchronize code and signal analysis. Deep memory is another key feature in this new instrument, giving you up to a million samples on each channel.

Digital signals from an analog perspective

As MCU clock speeds increase, the analog nature of your digital signals becomes increasingly important. Is the output threshold of the serial interface or the CAN bus high enough? Is the timing of the digital outputs correct? The MSO delivers reliable answers to these and many other questions, including such problems as electromagnetic radiation, capacitive loading and power supply faults.

Whether you use an MSO by itself or in conjunction with an emulator, you'll find it offers a new dimension of MCU debugging power.

HINT 1

Verifying PWM dead time in motor controllers

Technical staff, Microcontroller Group, Siemens Semiconductors *www.sci.siemens.com*

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 such as the HP 54645D 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.

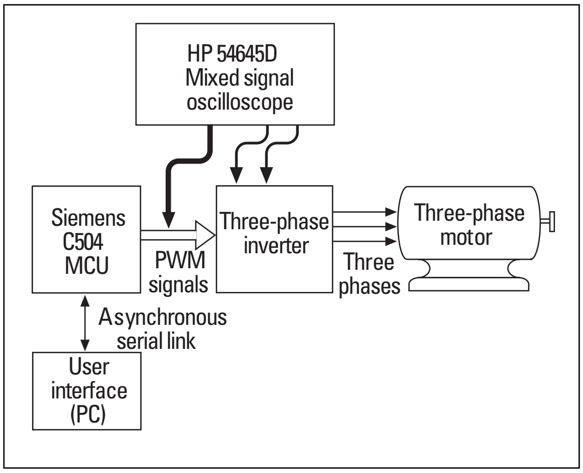
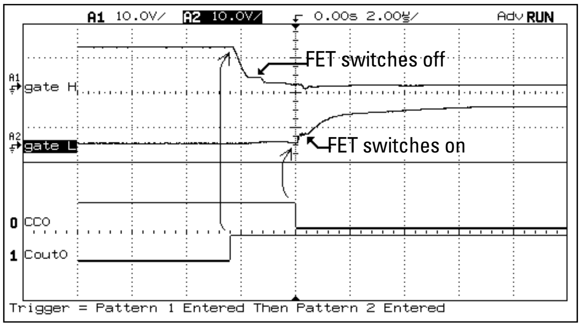


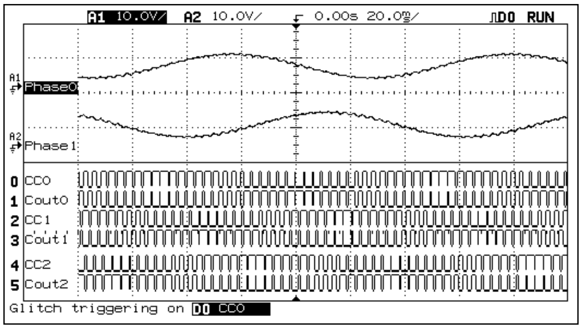
Figure 1: Block diagram of an open-loop configuration for generating safe PWM signals to drive a three-phase motor.

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.



Trigger = Pattern 1 Entered Then Pattern 2 Entered

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



Glitch triggering on D0 CC0

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

HINT 2

Debugging external interrupts in Siemens C500 MCUs

By David Bordui *www.qxi.com*

David Bordui a Field Application Engineer with QXI, a Siemens representative in Longwood, Florida, USA.

Is a peripheral in your system screaming for attention? Under certain conditions, the Siemens C500 and derivatives (as well as other 8051-based microcontrollers from other manufacturers) can miss interrupt requests from peripherals.

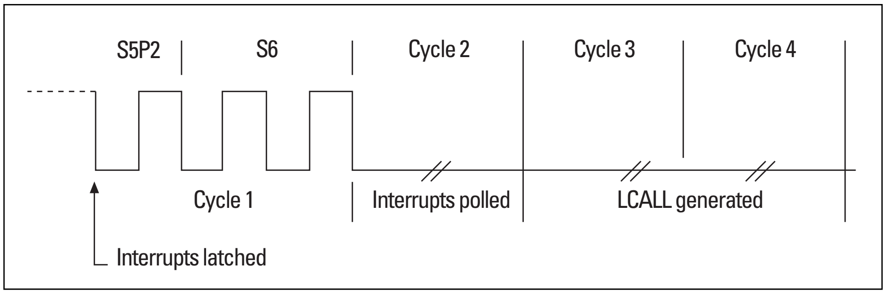


Figure 1: The CPU timing diagram shows why LCALL gets blocked after the RETI command is generated.

The potential for trouble exists whenever the MCU is set to edge-sensitive interrupt mode and the peripheral requires the MCU to reset its interrupt line. Common culprits include multi-channel USART devices that randomly interrupt the MCU upon reception of serial data from a peripheral. The MCU can miss the interrupt edge and therefore fail to reset the interrupt line. Once this happens, the peripheral appears to be locked up.

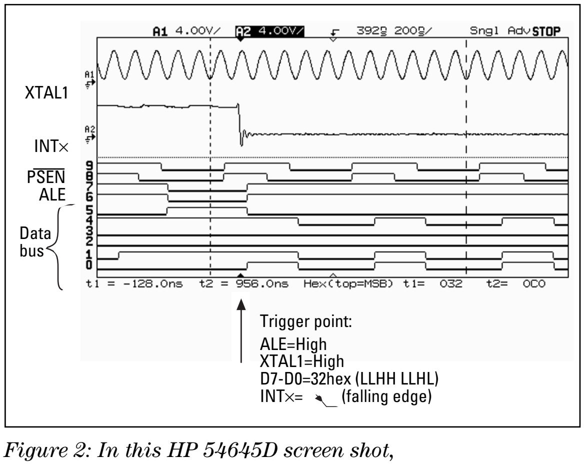


Figure 2: In this HP 54645D screen shot, a closer inspection of the external interrupt edge (analog line A2) relative to the RETI fetch (digital lines 0-7) and oscillator segment and phase reveals the offending condition.

The trouble starts when the LCALL instruction gets blocked, which can happen in several different situations with C500 devices (two of which are particularly hard to diagnose because all the relevant action happens inside the CPU core).

A close look at the CPU timing diagrams in the C5xx databook helps explain the problem. C5xx devices divide the external oscillator by 12 to form a single instruction cycle, and each cycle is divided into 6 segments (S1-S6) with 2 phases of the oscillator in each (P1 and P2). An address latch enable (ALE) always occurs on the S1P2..S2P1 and S4P2..S5P1 time intervals, and ALE transitions indicate a fetch of the next opcode. The interrupt reset instruction RETI is a 1-byte, 2-cycle instruction, which means the next three opcodes are discarded while the interrupts remain latched (Figure 1).

To isolate this problem, I connected digital channels from an HP 54645D mixed signal oscilloscope (MSO) to ALE, $\overline{\text{PSEN}}$ and data lines D0-D7. To avoid excessive loading on the oscillator, I used the two analog high-impedance channels to capture and display the oscillator (XTAL1) and the external interrupt signal (INTx).

The MSO's pattern trigger made it easy to capture the problem scenario by watching for a "high" on ALE, an $0X32_{HEX}$ (RETI instruction) on the data bus, a "high" on the oscillator input, and an active falling edge on INTx. This makes it simple: if the scope triggers, the problem is there (Figure 2).

Switching to level-sensitive interrupts will avoid the problem whenever that's an option. When it isn't, a simple hardware fix will do the trick. Connect the ALE line (or $\overline{\text{PSEN}}$) to one input of an OR gate and the interrupt output from the external USART to the other input, then connect the OR output to INT×. The USART's interrupt line will gate the pulsing ALE line, thereby creating multiple edges for the MCU. (For rising edge interrupts, use an AND gate.)

Of course, it's always a good idea to use software checks to make sure each interrupt event is serviced only once, and it's particularly important in this case since you're generating multiple interrupt signals.

HINT 3

Testing a C167–based PWM solenoid circuit

By Mark Fabiny *www.sci.siemens.com*

Mark Fabiny is a Microcontroller Applications Specialist for Siemens Microelectronics in Dallas, Texas, USA.

Microcontrollers provide a level of programmability and functionality that makes them a viable alternative to application-specific semiconductors such as pulse width modulation controllers. However, this increased capability can present new challenges for testing and integrating new designs.

The example presented here comes from a customer's need to drive sixteen solenoids in a medical analyzer. Previous designs used three complex programmable logic devices driving discrete power drivers in saturation mode. The new design required that the solenoids also be driven with a "peak-and-hold" driver to limit current draw and to prevent damage if a low-impedance solenoid is accidentally installed.

The customer investigated various PWM controllers and drivers but this approach did not satisfy the cost or functionality requirements. We proposed a new design based on an MCU. As shown in Figure 1, the SABC167SR MCU has 32 capture/compare (capcom) channels and 16 analog inputs. The functionality and I/O capabilities of the C167 let the customer use just one MCU to drive 16 solenoids while accomplishing all of the design goals.

The HP 54645D mixed signal oscilloscope (MSO) permits simultaneous analysis and complete debugging of the closely coupled analog and digital signals in the circuit (Figure 2). The MSO's deep memory captures several cycles of the solenoid, which aids in debugging the section that uses pulse width modulation of the power driver to modify the holding current. Also, the MSO's 16 digital channels make it possible to view all 16 solenoids at once—particularly useful when integrating the entire system because the solenoids are actuated in specific sequences that must be verified.

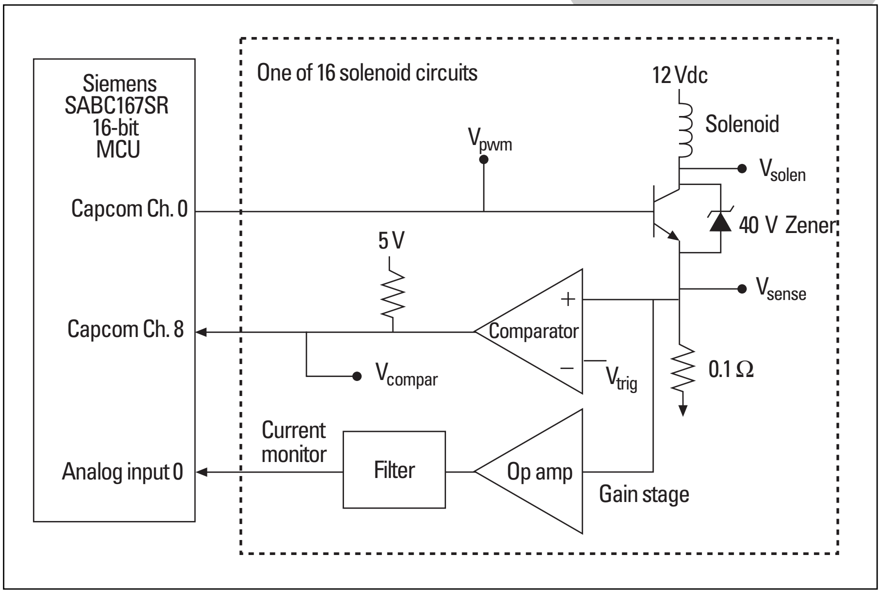


Figure 1: The capcom channels are individually programmed for "input capture" and "output compare" functions, which drive each solenoid and sense the peak current threshold, respectively. Capcom channel 0 drives a power transistor; the sense resistor at the emitter leg is for current monitoring. The op-amp conditions this signal and provides feedback to Analog Input 0, which monitors the holding current. The comparator signals when peak current is reached.

Circuits of this nature can be difficult to examine because the PWM portion requires a time base that is much faster (less than 50 µs) than the overall solenoid on-time (greater than 1 s). The events surrounding the transition from peak mode to hold mode are the most critical, and the MSO's glitch triggering is a convenient and reliable way to capture these signals.

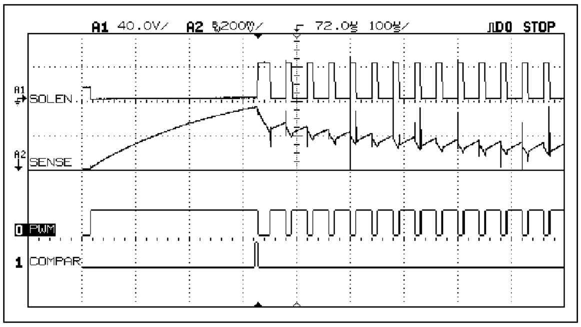


Figure 2: Four key signals from Figure 1. SOLEN is an analog signal from the low side of the solenoid; the zener diode clamps inductive spikes at 40 V. SENSE, measured across a 0.1Ω resistor, peaks around 380 mV, which corresponds to 3.8 A PWM drives the power transistor. The initial 300 µs pulse is the time required to achieve peak current. COMPAR triggers at 380 mV across the sense resistor and is fed into capcom channel 8 to signal the MCU to enter hold mode until the solenoid is turned off.

HINT 4

Evaluating high-precision servo drives

By Frank Stolle and Jörg Droßmann *www.sican.com*

Frank Stolle and Jörg Droßmann are Project Managers with SICAN Braunschweig GmbH, Braunschweig, Germany.

Our servo positioning systems typically use incremental encoders for precise position control. These encoders generate two sine waves offset by 90 degrees, then we use comparators to derive square-shaped position counter signals from the sine waves. Every square wave edge increments or decrements a position counter, with rotation direction indicated by the edge direction and the level of the encoder signals. For maximum precision, we also sample and digitize the sine waves.

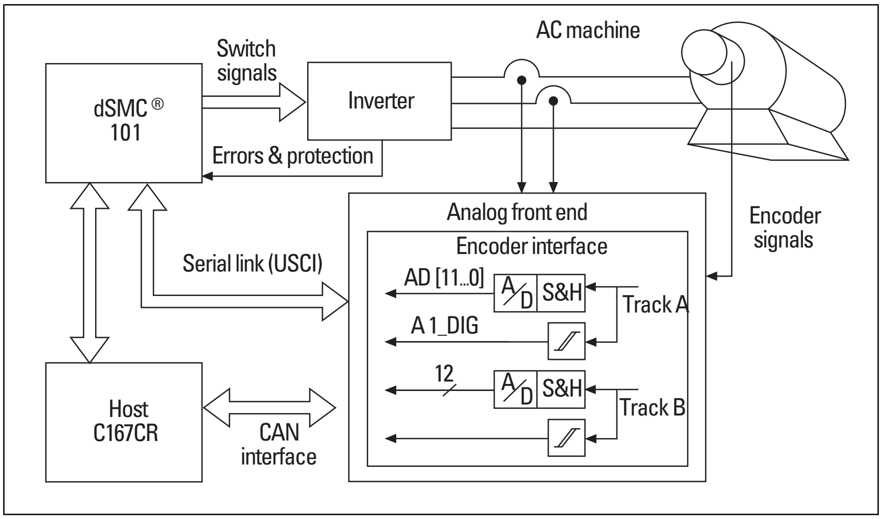


Figure 1: Block diagram of a high-precision position controller.

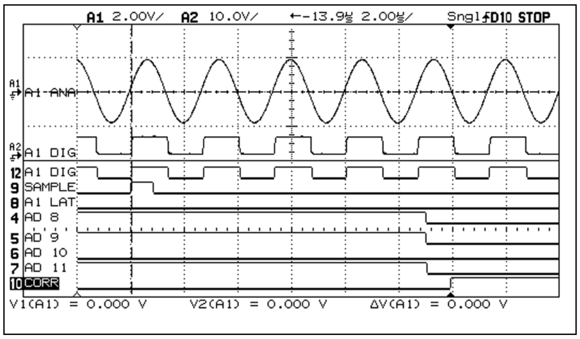


Figure 2: In this HP 54645D screen shot, signals AD 11 and A1 LAT give conflicting information about the sampled signal.

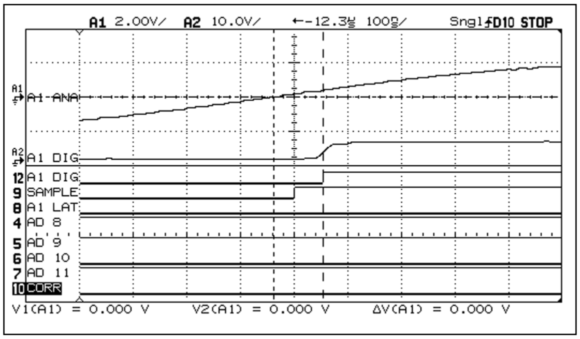


Figure 3: The cause uncovered: the sampling edge occurred immediately after the zero crossing.

Figure 1 shows a servo system that includes a Siemens C167CR MCU and a *digital Smart Motion Controller* (dSMC) from SICAN. The C167CR manages communication interfaces and high-level control tasks. The dSMC contains a PWM unit, a position counter, ADC interfaces and an embedded DSP that performs torque, speed and position control.

Figure 2 highlights a problem we recently encountered. At the beginning of every control loop, the dSMC (see the SAMPLE signal) latches the position counter values and the encoder signals. Simultaneously, the encoder sinusoids are sampled. After conversion (11 µs later), the signal AD11 (MSB of the ADC output) goes low, indicating that the digitized analog signal had a positive polarity when sampled (SAMPLE). However, the synchronous comparator output (labeled A1 DIG) remains low, indicating a negative polarity.

Figure 3 explains the inconsistency. The sampling edge occurred immediately after the rising sine signal crossed zero. However, the comparator output (A1 DIG) had not yet changed, possibly due to hysteresis or a zero offset difference in the ADC's level. The dSMC chip can solve this problem by recognizing such inconsistencies and correcting the reading of the position counter.

Examining behavior such as this is nearly impossible with a conventional oscilloscope. The MSO's deep memory also let us trigger on the inconsistent ADC result and then explore the related signals.

HINT 5

Using deep memory to measure transceiver disable time

By Patrick Pettibon *www.sci.siemens.com*

Patrick Pettibon is a Field Applications Engineer with Siemens Microelectronics in Dallas, Texas, USA.

Even the most powerful scope triggering features aren't much help if you can't access the signals you'd like to use as trigger events. If you've ever probed dense circuit boards or the fine-pitch leads of small but powerful chips such as the Siemens C166 MCU family, you understand the problem.

On a recent project, I encountered a transceiver controlled by a C161 MCU; a PC controlled the entire system through its serial port (Figure 1). I wanted to measure the disable time of the transceiver. Sounds easy enough: just trigger on the chip select and write strobe signals to the peripheral.

While my HP 54645D mixed signal oscilloscope (MSO) could certainly trigger on such a combination, the software running on the C161 was confusing the issue by writing to the peripheral for more than one purpose. I needed to qualify the trigger with an address, but I didn't have a test clip for the MCU's tiny MQFP package, leaving me without access to the data/address bus.

With a firmware change, I could've found a way to cycle the transceiver's state, but the MSO's MegaZoom feature saved me the trouble. By entering a disable command on the PC and then pressing the Run/Stop key on the scope, I was able to capture the event manually. Even though the interesting part of the signal lasted only a few microseconds, HP MegaZoom captured enough samples to see the results in great detail.

I first set the time base to 10 ms/div in order to capture 100 ms of the transceiver signal. Next, I disabled the transceiver from my PC, then quickly pressed Run/Stop on the scope. After several tries, I was able to capture the disable event (Figure 2). At 10 ms/div, I lacked sufficient visual resolution to measure the event, but HP MegaZoom let me zoom in to 2 µs/div (Figure 3), providing more than enough resolution. In other words, using my finger as the trigger source, I was able to capture a transient event only 2 µs long.

You can capture similar events using other manual triggers such as push buttons, serial commands or timer interrupts. With HP MegaZoom's one million samples and simple pan and zoom, you just need to trigger somewhere near the signal to see it.

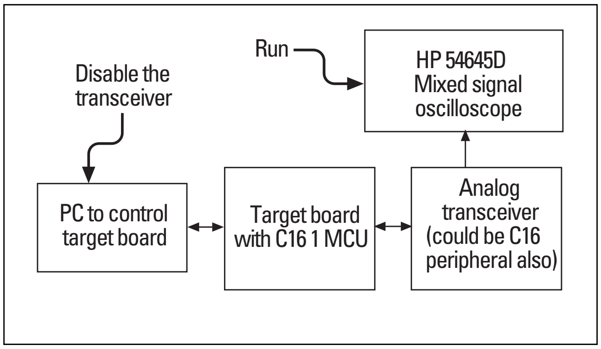


Figure 1: Measuring the transceiver disable time.

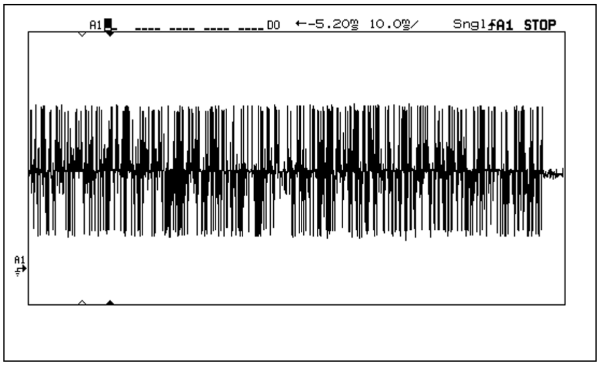


Figure 2: The long data stream captured in HP MegaZoom memory.

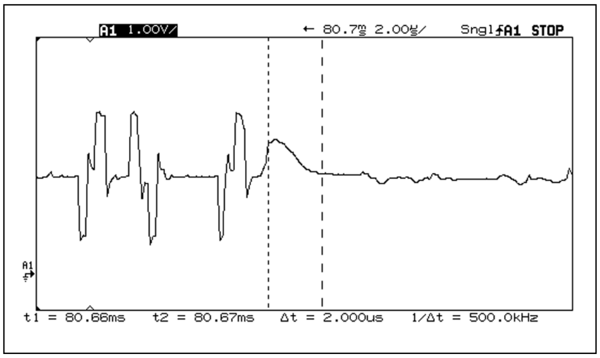


Figure 3: By panning and zooming, I found the trigger disable time—an event only 2 µs long.

HINT 6

Verifying A/D conversion in a C164 single-chip system

By Thomas Hammerschmidt and Wolfgang Schmitt *www.hitex.com*

Thomas Hammerschmidt and Wolfgang Schmitt are with Hitex Development Tools in Eching, Germany.

We recently deployed a Siemens C164CI MCU in an automotive application (Figure 1) to monitor and respond to a variety of operating parameters, including pressure, temperature and acceleration. Eight analog signals are connected through Port 5 to the analog-digital converter, which converts channels 0-3 continuously (the other channels are converted based on timer inputs). The results are transferred to internal memory via a DMA method known as peripheral event control (PEC). ROM-based analysis software watches for failures and mismatches among the various signals and controls the outputs of Port 4 depending on the detected failure. Failures are reported to the on-chip RAM along with the real time clock value and can be monitored via a serial link or CAN interface.

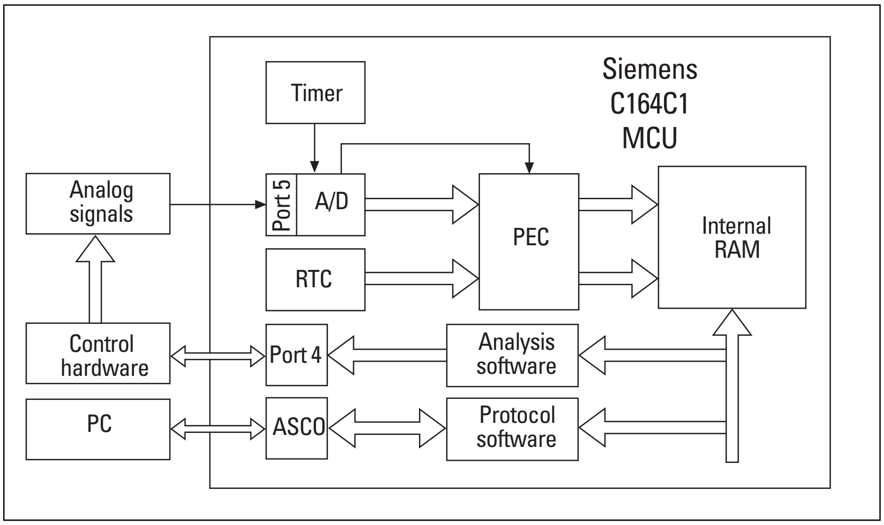
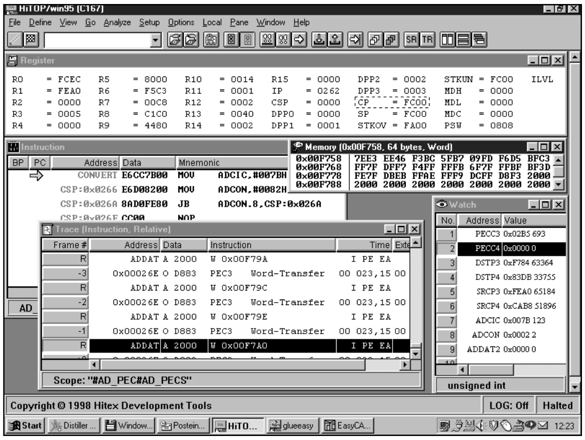


Figure 1: Block diagram of the C164CI automotive application.

During design integration, we needed to verify system behavior when one of the analog channels reported a failure, which required simultaneous capture of the program flow and the external analog (Port 4) and digital (Port 5) signals.

To analyze the program flow, an in-circuit emulator with the ability to debug and modify ROM code proved indispensable (Figure 2). The enhanced trace and trigger functions, along with integrated performance analysis, also made the investigations much easier.


HiTOP/win95 (C167)
File Define View Go Analyze Setup Options Local Pane Window Help
SR TR
Register
Instruction
Memory (0x00F758, 64 bytes, Word)
Watch
Trace (Instruction, Relative)
AD_
Start Distiller ... Window... Postein... HiTO... glueeasy EasyCA... 12:23

```
R0    = FCEC   R5    = 8000   R10   = 0014   R15   = 0000   DPP2  = 0002   STKUN = FC00   ILVL
R1    = FEA0   R6    = F5C3   R11   = 0001   IP    = 0262   DPP3  = 0003   MDH   = 0000
R2    = 0000   R7    = 00C8   R12   = 0002   CSP   = 0000   CP    = FC00   MDL   = 0000
R3    = 0005   R8    = C1C0   R13   = 0040   DPP0  = 0000   SP    = FC00   MDC   = 0000
R4    = 0000   R9    = 4480   R14   = 0002   DPP1  = 0001   STKOV = FA00   PSW   = 0808
```

```
BP  PC       Address  Data      Mnemonic
             CONVERT  E6CC7B00  MOV      ADCIC,#007BH
      CSP:0x0266  E6D08200  MOV      ADCON,#0082H
      CSP:0x026A  8AD0FE80  JB       ADCON.8,CSP:0x026A
      CSP:0x026E  CC00      NOP
```

```
0x00F758  7EE3 EE46 F3BC 5FB7 09FD F6D5 BFC3
0x00F768  FF7F DFF7 F4FF FFFB 6F7F FFBF BF3D
0x00F778  FE7F DBEB FFAE FFF9 DCFF D8F3 2000
0x00F788  2000 2000 2000 2000 2000 2000 2000
```

```
No.  Address  Value
1    PECC3    0x02B5 693
2    PECC4    0x0000 0
3    DSTP3    0xF784 63364
4    DSTP4    0x83DB 33755
5    SRCP3    0xFEA0 65184
6    SRCP4    0xCAB8 51896
7    ADCIC    0x007B 123
8    ADCON    0x0002 2
9    ADDAT2   0x0000 0
unsigned int
```

```
Frame #      Address  Data    Instruction                 Time  Exte
R             ADDAT  A 2000  W 0x00F79A               I PE EA
-3       0x00026E  O D883    PEC3    Word-Transfer  00 023,15 00
R             ADDAT  A 2000  W 0x00F79C               I PE EA
-2       0x00026E  O D883    PEC3    Word-Transfer  00 023,15 00
R             ADDAT  A 2000  W 0x00F79E               I PE EA
-1       0x00026E  O D883    PEC3    Word-Transfer  00 023,15 00
R             ADDAT  A 2000  W 0x00F7A0               I PE EA
Scope: "#AD_PEC#AD_PECS"
```

```
Copyright © 1998 Hitex Development Tools    LOG: Off    Halted
```

Figure 2: Hitex DProbe167 emulator screen with display of PEC transfers and register values.

To measure the analog and digital signals, we used the HP 54645D mixed signal oscilloscope (MSO), with the emulator providing a trigger (displayed as TRIGG in Figure 3) for the MSO. In addition to the simultaneous analog and digital measurements, the MSO's deep memory helped us get a complete picture of the overall system behavior.

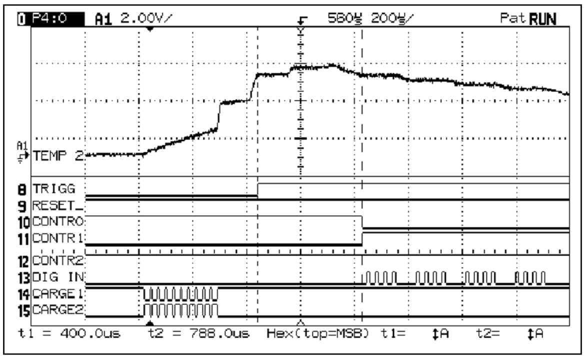


Figure 3: Correlation between analog and digital channels.

HINT 7

Using propagation delay to measure CAN cable lengths

By Ingo Pohle and Renate Schultes *www.microconsult.com*

Ingo Pohle and Renate Schultes are with MicroConsult, a training center for microelectronics and information technology in Munich, Germany.

The maximum acceptable length of an installed CAN cable is limited by baud rate, propagation delay through each component and any offset between the oscillators supplying the baud rate clock. If you know the delay through the transceivers and the delay through the entire network, you can derive the delay through the cable itself and from there, the actual cable length. Using an HP 54645D mixed signal oscilloscope (MSO), we first measure the transceiver delay (Figure 1).

Next, we measure the total delay from node to node, using a trigger signal from the master (CAN end node #1) to start data acquisition on the MSO (Figure 2).

The master starts to transmit a frame immediately after sending the trigger signal. Figure 3 shows the trigger signal (on the MSO's digital channel 5) and the T×D and R×D signals on the MSO's two analog channels. Using the MSO's display cursors, we then measure the length of the 12-bit interval from the beginning of the acknowledge bit to the next following start bit.

At the 400 Kbps baud rate in this system, the nominal time interval is 30 µs (12 bits × 2.5 µs/bit). The 900 ns difference between the nominal value and the measured interval shown in Figure 3 (Δ t = 29.1 µs) consists of the transceiver's signal delay and the signal propagation delay along the cable. Knowing this, we can compute the delay through the cable:

$$\begin{aligned} T_{\text{cable}} &= T_{\text{total}} - (4 \times T_{\text{transceiver}}) \\ &= 900\text{ ns} - (4 \times 87\text{ ns}) \\ &= 552\text{ ns} \end{aligned}$$

Next, we can compute the actual cable length:

$$\begin{aligned} L_{\text{cable}} &= T_{\text{cable}} \times 0.17\text{ m/ns} \\ &= 552\text{ ns} \times 0.17\text{ m/ns} \\ &= 93.84\text{ m} \end{aligned}$$

Knowing the exact length of the installed cable, we can confirm proper operation at the operating baud rate.

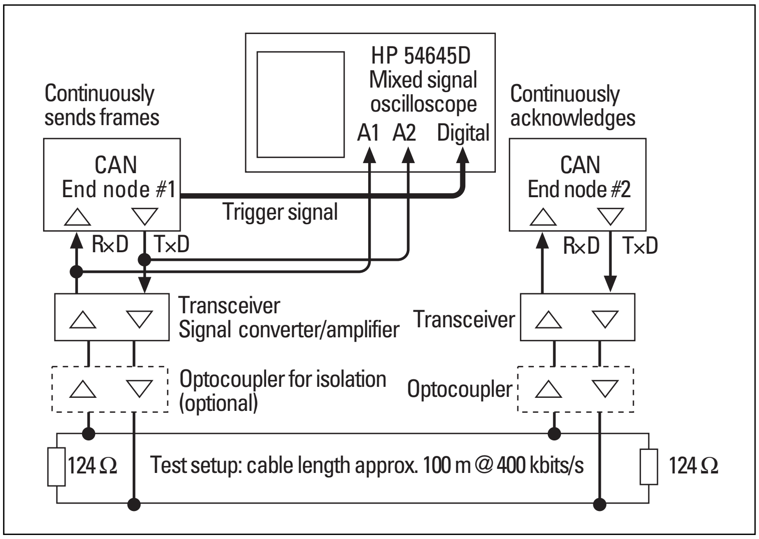


Figure 2: Test system configuration for cable length measurement; a mixed signal oscilloscope can measure both the analog and digital signals in this point-to-point CAN system. If the CAN network also includes optocouplers, we need to measure and account for their delay as well.

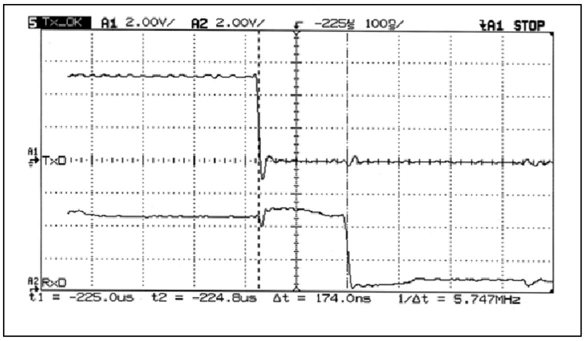


Figure 1: Signal propagation delay (TxD - RxD) through one transceiver is 174 ns (87 ns in each direction).

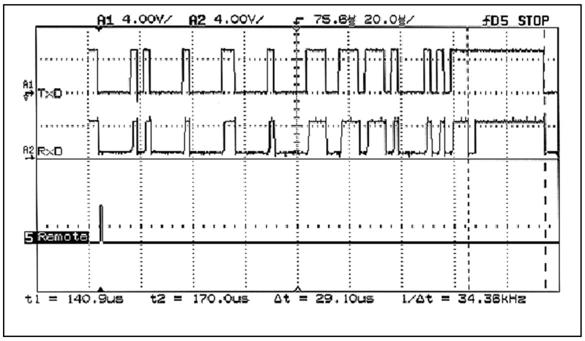


Figure 3: The time interval for this 12-bit sequence is 29.1 µs from the acknowledge bit to the next start bit.

HINT 8

Accessing fine-pitch I/O pins on Siemens MCUs

By Johnnie Hancock www.hp.com/go/bi

Johnnie Hancock is the worldwide program manager for MCU debugging tools at Hewlett-Packard in Colorado Springs, Colorado, USA.

As Siemens chip designers pack more functions into their C500/166 family of microcontrollers, pin counts of these powerful controllers have grown and the space between pins has shrunk. Pin spacings of 0.5 mm and 0.65 mm are not at all uncommon for these devices. The power and added functionality of these new MCUs is wonderful, to be sure, but troubleshooting designs based on these controllers can be a chore because connecting scopes and logic analyzers has become much more difficult and less dependable.

Engineers have tried a variety of techniques to access the pins they need to test, but these probing tricks cause as many headaches as they try to eliminate. For example, special fine-pitch IC clips can be both expensive and fragile. Soldering lead extenders onto the pins of the microcontroller can cause all kinds of problems, from heat damage to the IC and shorts between adjacent pins, not to mention the chore of removing the kludge before you ship the finished product. In addition, the added inductance of the wire extenders can cause excessive distortion in captured waveforms.

The HP Wedge probe adapter (see back cover) is a handy new solution to this problem. With this device, you can probe up to 8 adjacent I/O signals on Siemens MCUs with 0.5-mm and 0.65-mm pin spacing. All you have to do is insert the HP Wedge between the IC pins you want to probe, where each compressible segment makes secure mechanical contact between a pair of pins (Figure 1). You then connect your scope or logic analyzer to the other end of the HP Wedge. The HP Wedge also holds your probe in place, so it's a hands-free solution (Figure 2). It’s easy to use, cost effective, provides excellent electrical performance, and doesn’t damage your microcontroller.

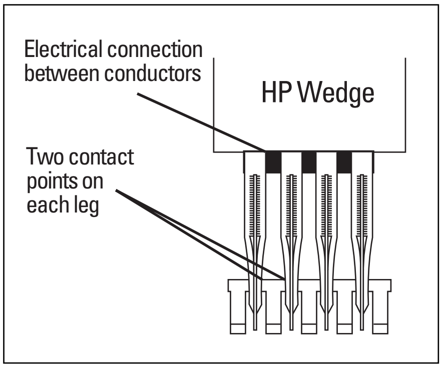

Electrical connection between conductors
HP Wedge
Two contact points on each leg

Figure 1: The HP Wedge inserts between adjacent pairs of pins on a fine-pitch IC package, providing secure, redundant, mechanically noninvasive contact.

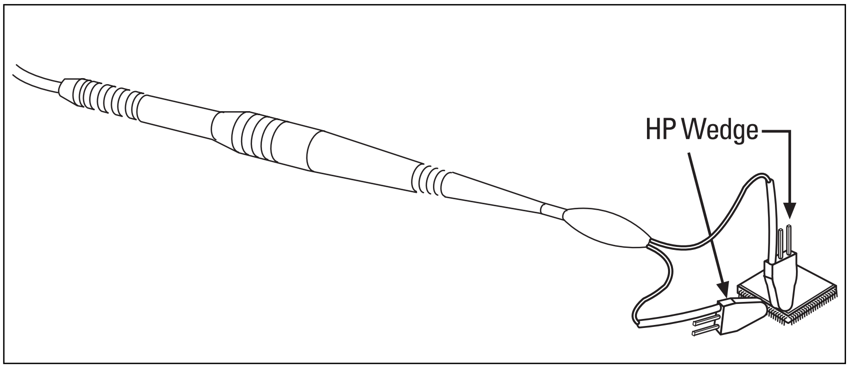
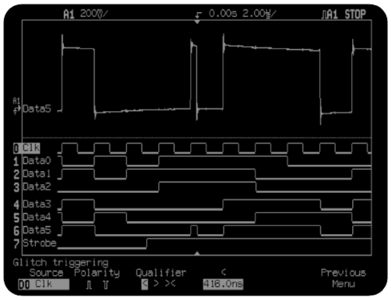


Figure 2: The HP Wedge holds your scope or logic analyzer probes in place, leaving your hands free to run your system and your test equipment.

HP 54645D Mixed Signal Oscilloscope

- **2 scope channels and 16 logic channels**
- **Powerful triggering**
- **HP MegaZoom deep memory**

MEGAZoom



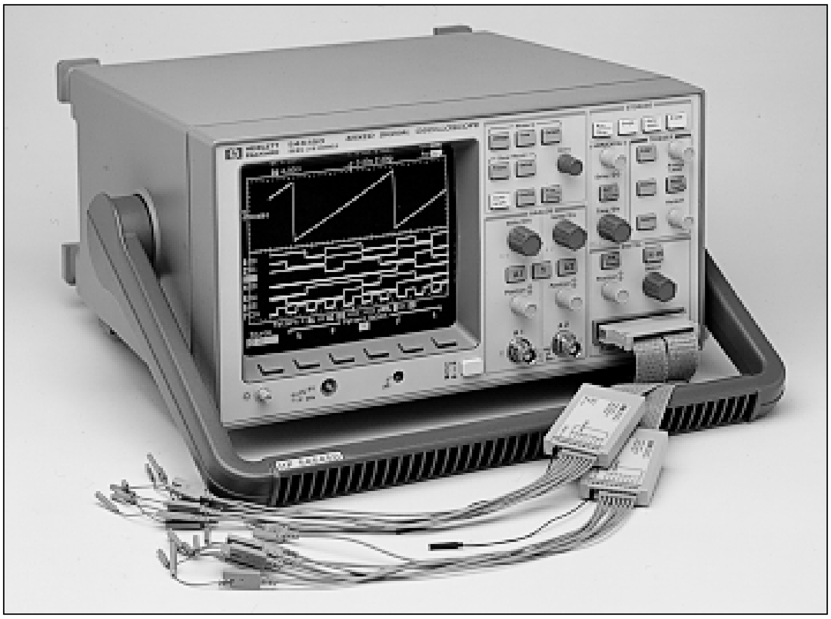
View circuit operation in ways you've never been able to see before.

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.

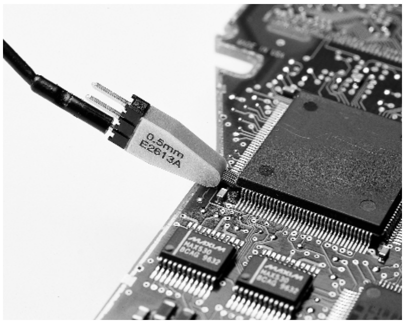
Because the HP 54645D is built on a scope foundation, it looks and feels like a familiar scope—- not like a complicated logic analyzer. And the combination of scope channels, logic timing channels and HP MegaZoom deep memory provides totally new ways to debug mixed analog-digital and MCU–based designs.



HP 54645D Mixed Signal Oscilloscope	
2 Scope channels	
Bandwidth	100 MHz (75 MHz @ < 10 mV/div)
Maximum sample rate	200 MSa/s
Memory depth	1 M points/channel
Peak detect	5 ns minimum
Input impedance	1 MΩ, 13 pF
Maximum input	400 V (dc + peak ac)
Range	1 mV/div to 5 V/div
Resolution	8 bits
16 Logic channels	
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 kΩ, 8 pF
Input level	±40 V max, 500 mVp-p min
Threshold range	±6.0 volts in 50 mV increments
Peak detect	5 ns minimum
Timebase	
Range (main & delayed)	5 ns to 50 s/div
Accuracy (non-vernier ranges)	
Scope, same channel	±0.01% of reading ±0.2% of screen width ±40 ps
Scope, chan to chan	±0.01% of reading ±0.2% of screen width ±80 ps
Logic , same channel	±0.01% of reading ±0.2% of screen width ± (1 logic sample period, 2.5 or 5 ns) ± chan-to-chan skew
Logic , chan to chan	±0.01% of reading ±0.2% of screen width ± (1 logic sample period, 2.5 or 5 ns) ± chan-to-chan skew
Triggering	
Sources	All channels and line
Logic trigger modes	Edge, pattern, glitch, advanced pattern, TV Advanced pattern operators: And, Or, Then, Entered, Exited, Duration time, Duration >, Duration <
Warranty	3 years
Ordering information and selected options	
HP 54645D Mixed signal oscilloscope Includes two scope probes (HP 10074), one logic cable (HP 54620-61601), power cord and manual HP 54645A 100 MHz 2-Channel oscilloscope with HP MegaZoom (analog-only version of HP 54645D) HP 54650A HP-IB Interface module HP 54652B RS-232/Parallel Interface module HP 54657A HP-IB Measurement/ Storage module HP 54659B RS-232/Parallel Measurement/Storage module HP 1185A Carrying case 106 HP BenchLink Scope software 1CM 5062-7345 Rack mount kit W50 Additional 2-year warranty	

The HP Wedge probe tip adapter

Although many accessories are available to connect scopes and logic analyzers to fine-pitch ICs, some can cause as many problems as they claim to solve. For example, typical spring-loaded alligator clips fall off, they short adjacent pins, and they often lack the electrical performance to replicate the signal faithfully. Specialized 0.5-mm clips are expensive and fragile. Poking around with a standard scope probe risks damaging the chip, and soldering wires onto the IC certainly won't impress your customers.



The HP Wedge provides accurate, mechanically noninvasive contact with the pins of the IC under test by inserting compressible dual conductors into the space between adjacent pins. Its unique design delivers secure, redundant contact on each pin, with no chance of shorting adjacent pins or damaging the DUT. Plus, the HP Wedge doesn't latch directly onto IC pins, so you can insert it while the board is active. After you've established a solid connection, you can easily attach the HP Wedge to scopes or logic analyzers with the appropriate accessories.

The engineers at HP DIRECT can help you define the right solution for your MCU debugging needs. Call 1-800-452-4844.

Product	IC pin spacing	Number of signals	Number of HP Wedges
HP E2613A	0.5 mm	3	1
HP E2613B	0.5 mm	3	2
HP E2614A	0.5 mm	8	1
HP E2615A	0.65 mm	3	1
HP E2615B	0.65 mm	3	2
HP E2616A	0.65 mm	8	1

Printed in USA 11/98
5968-2498E