

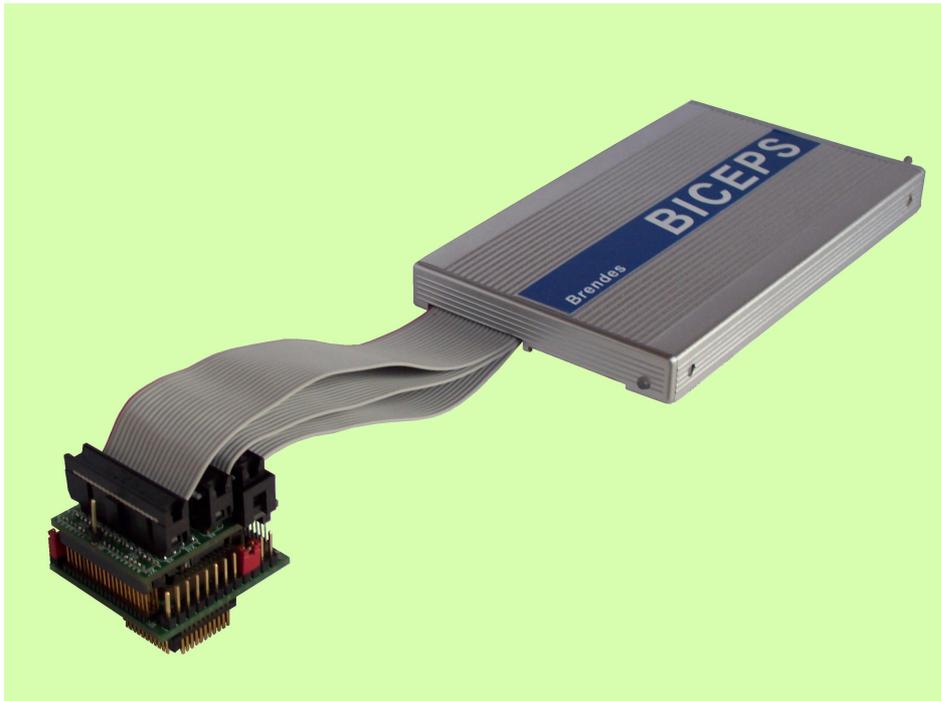
# Brendes

---

emb**e**tter tools

# *BICEPS*

emb**e**tter emulators for  
**8051** and **ARM**



Der Echtzeit-In-Circuit-Emulator **BICEPS-8051** von Brendes Datentechnik ist eine umfassende und professionelle Debugging-Lösung für 8051 Mikrocontroller.

**[www.8051-emulator.de](http://www.8051-emulator.de)**

Brendes Datentechnik GmbH  
Dresdener Str. 10  
D-26160 Bad Zwischenahn, Germany

Tel.: +49 (0) 4403 816838  
Fax: +49 (0) 4403 816839

eMail: [info@brendes.de](mailto:info@brendes.de)  
Web: [www.brendes.de](http://www.brendes.de)

## embetter **BICEPS-8051**

### Debugging Features

- In-Circuit-Emulation in Echtzeit bis zur max. Taktfrequenz
- 512k Emulation Memory mit Banking-Support
- Aktuelle, rekonfigurierbare FPGA-Technologie
- True-Single-Chip-Emulation mit Brendes-**Softhooks**™
- Debugging:
  - Echtzeit-Programmausführung
  - Zugriff auf alle internen und externen Speicherbereiche
  - keine Einschränkungen bezüglich Speicherbelegung, Interrupts , UART usw.
  - Programm Download
  - Breakpoints auf Programm- und ExtData-Speicher
- Real-Time-Trace:
  - 128k x 96 Bit Real-Time-Trace
  - 32-Bit Echtzeit-Uhr für Zeitmessungen im Bereich 100ns .. 12h, Auflösung 100ns
  - Time Stamp
  - Performance Analyse
  - Real-Time-Trace-Filter
- 16 Externe Eingänge für Break- und Trace-Zwecke
- Mapping
- Watchdog-Support
- USB-Schnittstellen
- Mitgelieferte Software:
  - Source-Level Debugger **BicWin**
  - Treiber (DLL) für **Keil µVision-DIE**
- Zum ARM-Debugging-Adapter inkl. Real-Time-Trace erweiterbar

## embetter features

# Vorteile des BICEPS-Emulators

### 1. embetter controller spectrum

#### Unterstützung aller aktuellen 8051-Flash-Versionen im True-Single-Chip Mode

Mit dem **BICEPS-8051**-Emulator können nahezu alle 8051-Controller auch im Single-Chip-Modus emuliert werden- wichtig vor allem bei den modernen Flash-Varianten. Durch die neue **Softhook™**-Emulationstechnik, die ohne Bondout-CPU's und ohne Debug-Lizenzen auskommt, können praktisch sämtliche 8051-kompatiblen Controller aller Halbleiter-Hersteller emuliert werden. Zum Teil ist der **BICEPS-8051**-Emulator der einzige Emulator, der bestimmte 8051-Typen unterstützen kann (siehe Controller-Liste)!

### 2. embetter connections

Neben dem klassischen POD-Konzept, bei dem die CPU auf dem Targetboard ersetzt wird, ist der Anschluss des **BICEPS-8051**-Emulators auch über einen Debug-Connector oder einen Universal-Adapter möglich

- **POD**: Anschluss über CPU-Sockel
- **Debug-Connector**: Anschluss über Stiftleiste und Flachbandkabel
- **Universal-Adapter**: Anschluss über Sockel für externen Programmspeicher, typischerweise bei Banking-Applikationen

### 3. embetter expandability

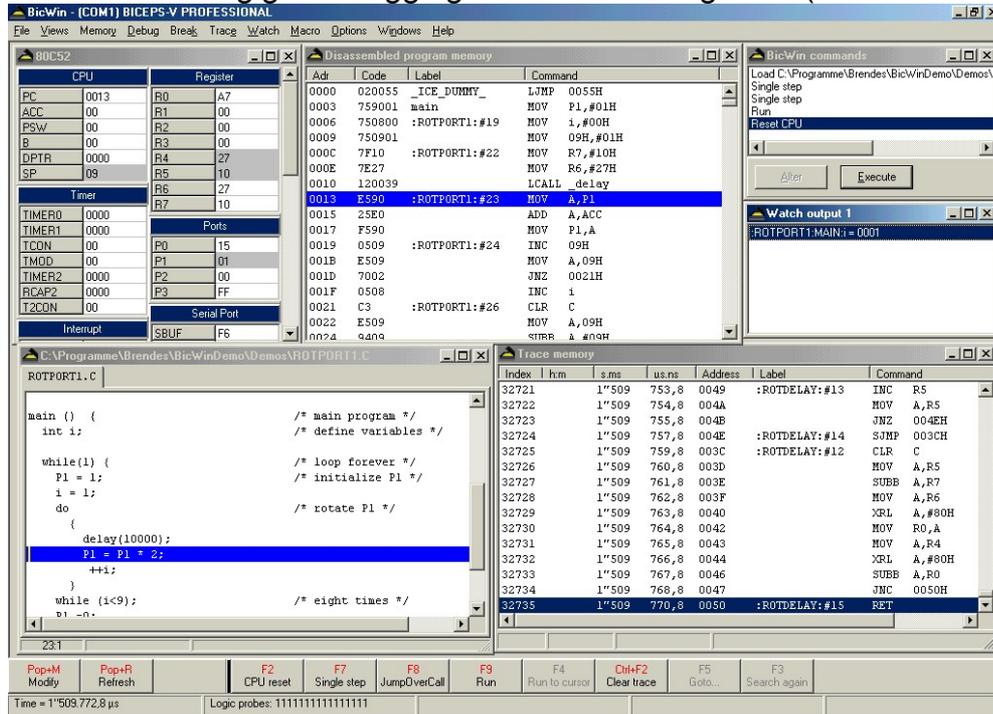
Der **BICEPS-8051**-Emulator kann per Lizenz zu einem professionellen ARM-Debugging-Adapter erweitert werden. Der **BICEPS-Professional** unterstützt neben der 8051-Emulation JTAG- und ETM-Real-Time-Trace-Debugging für ARM.

## embetter user interface

# BicWin-Debugger oder Keil-µVision-IDE

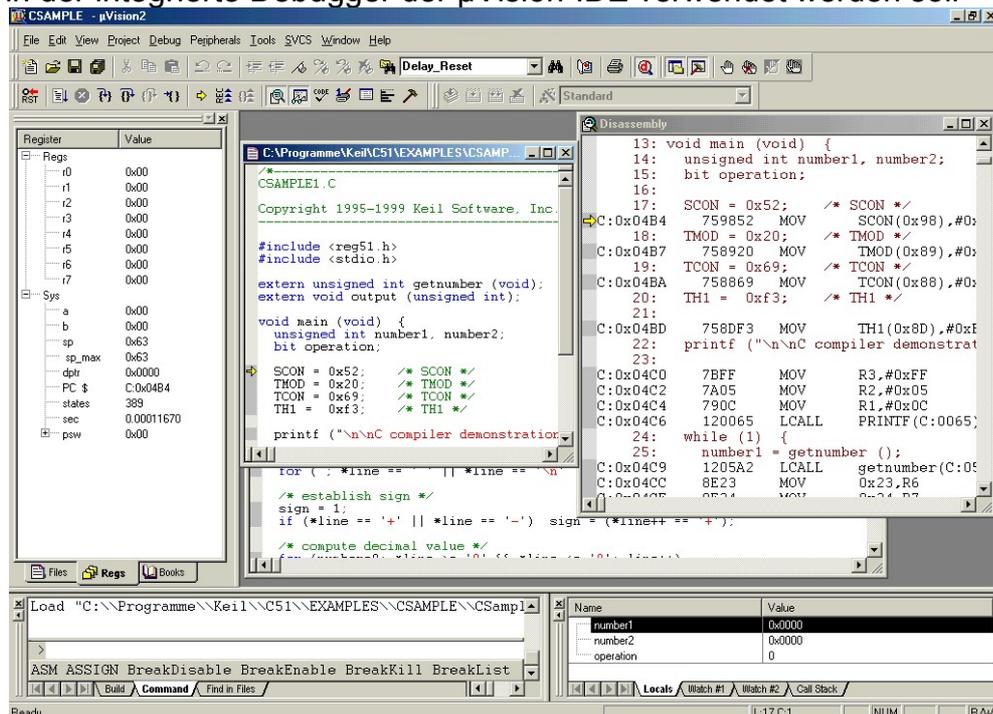
## 1. BicWin: der Brendes Sourcelevel Debugger

wenn eine unabhängige Debugging-Software benötigt wird (deutsch oder englisch)



## 2. Keil µVision: Integrierte Entwicklungsumgebung

wenn der integrierte Debugger der µVision-IDE verwendet werden soll



## embetter **BICEPS-8051**

### Real-time-Trace-Memory

Der **BICEPS-8051-Professional** -Emulator ist mit einem Real-Time-Trace-Speicher mit einer Kapazität von 128k x 96 Bit ausgerüstet. Dieser dient der Aufzeichnung von Signalen während eines Echtzeit-Programmlaufs ähnlich einem Logik-Analysator. Zusätzlich vorhanden ist ein 32-Bit-Echtzeitähler, der synchron zu jeder Programmausführung läuft, und somit exakte Zeitmessungen ermöglicht.

Im Trace-Speicher werden folgende Daten aufgezeichnet (insgesamt 96 Bit):

- 24 Bit Adressbus-Signale
- 16 Bit Datenbus-Signale
- 16 Bit externe Signale (Logic-Probes)
- 32 Bit Stand des Echtzeitählers (Zeitstempel, Time Stamp)
- 8 Bit Kontroll-Signale

Anhand der aufgezeichneten Informationen kann man sich sehr leicht einen Überblick über den Ablauf und die Zeitdauer eines Programms oder der externen Signale verschaffen. Weitere Funktionen bieten umfangreiche Möglichkeiten:

- Darstellung auch während der Programmausführung möglich, ohne dass diese beeinflusst wird ("On the fly"-Zugriff).
- Zeit-Messungen durch Bildung von Zeitdifferenzen des aufgezeichneten Zeitstempels
- Performance-Analyse
- Suchfunktionen für aufgezeichnete Daten
- Sourceline-Trace-Modus (Aufzeichnung von Quelltext-Zeilen)
- Auswahl von Adressbereichen (Echtzeit-Adress-Filter) für die Trace-Aufzeichnung
- Steuerung der Aufzeichnung durch Trigger-Ereignisse
- Programmabbruch durch Überlauf des Trace-Speichers mit vorgegebbarer Anzahl aufzuzeichnender Einträge

### Real-Time Trace-Filter-Logik

Um die Kapazität des Trace-Speichers zu erhöhen, besitzt der **BICEPS**-Emulator einen Real-Time Trace-Filter-Speicher. Er stellt für jede Programm- oder Xdata-Adresse ein Enable-Bit zur Verfügung. So kann für jede

- Source-Datei (module)
- Funktion
- Quelltextzeile oder sogar
- Assembleranweisung

festgelegt werden, ob sie aufgezeichnet werden soll oder nicht.

Werden häufig ausgeführte Warteschleifen oder Interruptroutinen vor der Trace-Aufzeichnung ausgefiltert, kann die Kapazität von Millisekunden auf Sekunden oder Minuten gesteigert werden!

## Darstellung von Tracespeicher-Inhalten

Der **BICEPS**-Emulator bietet verschiedene Möglichkeiten zur Darstellung der Trace-Aufzeichnung. Typischerweise werden die Inhalte auf Quelltextebene angezeigt, d.h. ausgeführte Zeilen werden als Liste mit Zeitstempel präsentiert (klassische Darstellung).

Index	Time	Address	Instruction
148199	0''563.664,2 µs	main:90 :93 :94	IOCLR1 = 0x00F80000; // Turn off LEDs 4-8 } }
148200	0''563.664,3 µs	main:60	for (n = 0; n <= 3; n++) {
148202	0''563.664,5 µs	main:61	if ((n==1)    (n==3)) {
148203	0''563.664,9 µs	main:67 :68	IOSET1 = 0x00800000; // set LED 8 delay(); // Delay ...
148215	0''563.667,5 µs	main:69 :71	IOCLR1 = 0x00810000; // Turn off LEDs 8 + 1 switch (n) {
148216	0''563.667,9 µs	:73 :74	IOSET1 = 0x00010000; // set LED 1 delay(); // Delay ...
148227	0''563.670,4 µs	main:75 :76	IOCLR1 = 0x00FF0000; // Turn off LEDs 1-8 break; }
148228	0''563.670,5 µs	main:93	}
148229	0''563.670,6 µs	main:61	if ((n==1)    (n==3)) {
148230	0''563.671,0 µs	main:62 :63	IOSET1 = 0x00100000; // set LED 5 delay(); // Delay ...
148241	0''563.673,5 µs	main:64 :65	IOCLR1 = 0x00F00000; // Turn off LEDs 5-8 } }
148242	0''563.673,6 µs	main:71	switch (n) {
148243	0''563.674,0 µs	main:78 :79	IOSET1 = 0x00020000; // set LED 2 delay(); // Delay ...
148254	0''563.676,5 µs	main:80 :81	IOCLR1 = 0x00FF0000; // Turn off LEDs 2-8 break; }
148255	0''563.676,6 µs	main:93	}

Collapsed function  
(traced lines not  
displayed)

Time stamp

Executed source lines

(**BICEPS-8051** Real-Time-Trace Source-level-Modus)

Mit der "Collapse/Expand"-Funktion können aufgezeichnete Programmteile ausgeblendet werden, um die Darstellung übersichtlicher zu machen. Dies gilt für jede Funktion oder Quelltext-Zeile ( in diesem Beispiel wird die **delay()**-Routine nicht dargestellt). Eine Detail-Darstellung auf Assembler-Ebene ist ebenfalls verfügbar.

## Kontext-Darstellung

In diesem Modus wird der Trace-Memory-Cursor im Quelltext-Fenster als Balken dargestellt. Damit ist es möglich, wie bei Einzelschritten durch das Programm vorwärts und rückwärts zu "steppen". Die direkt vor oder nach dem Cursor ausgeführten Zeilen werden hervorgehoben. So erkennt man auf einen Blick, welcher Teil eines Programms (**then**- oder **else**-Zweig, welcher **case**-Zweig einer **switch**-Anweisung) wirklich ausgeführt wurde.

The screenshot displays the BICEPS-8051 Real-Time-Trace Kontext-Modus interface. It features a 'Trace memory' window on the left and a source code window on the right. Red arrows point to various elements:

- Trace memory:** A table with columns for Index, Time, and Address. The entry at index 148270 is highlighted in yellow.
- Source text:** C code for a serial interface loop. The code includes a while loop with an if-else switch statement. The code is color-coded: white for executed lines and grey for non-executed lines.
- Executed source lines:** Lines 61-63, 67-69, 73-75, 79-81, 83-85, and 89-91 are highlighted in white.
- Source lines not executed (grey):** Lines 64-66, 70-72, 76-78, 82, 86-88, 90-92, and 94 are highlighted in grey.
- Annotations:**
  - Trace memory cursor:** Points to the highlighted entry in the trace memory window.
  - Time stamp:** Points to the time value in the trace memory window.
  - Collapsed function (traced lines not displayed):** Points to the 'init\_serial()' function call at the top of the source code.
  - Source lines executed (white): If statement and else branch:** Points to the if-else structure in the code.
  - Source lines executed (white): Switch statement and case branch:** Points to the switch statement and its cases.
  - Source lines not executed (grey):** Points to the non-executed parts of the code.

(BICEPS-8051 Real-Time-Trace Kontext-Modus)

## embetter

# Break-Logik

Der **BICEPS-8051**-Emulator bietet umfangreiche Unterbrechungsmöglichkeiten für ein in Echtzeit ausgeführtes Programm an. Dazu ist im Emulator eine komplexe Kontroll-Logik vorhanden, die bei Speicherzugriffen auf den Programmspeicher oder den externen Datenspeicher **Adressbus**, **Datenbus** und **externe Signale** in Echtzeit auf das Eintreten vordefinierte Bedingungen überprüft.

Es können zwei Unterbrechungsereignisse definiert werden, die einzeln oder zusammen eine Unterbrechung der Programmausführung auslösen können. Jedes Ereignis kann beliebig viele Einzeladressen beinhalten, außerdem stehen Ereigniszähler mit bis zu 24 Bit Kapazität zur Verfügung.

Die Unterbrechungsereignisse können auf verschiedene Weise verknüpft werden, so dass auch bedingte Unterbrechungen möglich sind. Weitere Möglichkeiten sind:

- Unterbrechung bei einem bestimmten Zustand der externen Signale
- Unterbrechung beim Überlauf des Real-Time-Trace-Speichers (Overflow)

Neben der Hauptfunktion, die Programmausführung zu unterbrechen, können die definierten Ereignisse auch zur Steuerung des Real-Time-Trace-Speichers und zum Auslösen externer Vorgänge, z.B. Triggerung eines Oszilloscopes, verwendet werden.

## embetter port replacement

### P0/P2-Emulation mit BICEPS-Softhooks®

Bei der Emulation von 80C51-Mikrocontrollern stellt sich das grundsätzliche Problem, dass durch den Anschluss des Emulationsspeichers Port-Signale verloren gehen, da diese als Bus-Signale benötigt werden. Dadurch können Single-Chip-Applikationen ohne externen Bus nicht emuliert werden. Dieses Problem wurde bisher gelöst durch

- spezielle Bondout-Versionen des Controllers oder
- Standard-Chips mit einem speziellen Emulationsmodus (Hook-Mode), bei dem I/O- und Bus-Signale mit höherer Frequenz gemultiplext und die I/O-Ports durch einen zweiten Baustein nachgebildet werden.

Beide Lösungen stehen jedoch nicht für alle Controller zur Verfügung.

Ein neues, von Brendes Datentechnik entwickeltes Verfahren ermöglicht es, alle 80C51-kompatiblen Controller mit P0/P2 als Ports zu emulieren. Dies gilt für die Emulation mit einem POD oder mit dem **BICEPS**-Debug-Connector. Dieses Emulationsverfahren heißt **Softhook**-Emulation. Es kann praktisch alle Bit-Zugriffe und die meisten Byte-Zugriffe (aber nicht alle!) auf die Ports P0 und P2 in Echtzeit emulieren, obwohl der Controller während der Emulation im normalen Bus-Modus läuft. Wenn in C51 programmiert werden soll, wird der Compiler durch eine Erweiterung veranlasst, **Softhook**-kompatible Programme zu erzeugen.

Die **BICEPS-Softhooks** bieten

- Nachbildung von P0 und P2 und der Operationen, die diese Ports ansprechen, durch Emulator-Hardware
- Keine Abweichung zwischen Emulator und Controller bei Ausführungszeit oder Programmgröße
- Kompatibel mit C51-Compiler:
  - Mehr als 90% des erzeugten Codes können direkt emuliert werden
  - für die restlichen P0/P2-Anweisungen werden schnelle **Softhook**-kompatible Intrinsic-Funktionen (3 Bytes) verwendet (siehe Beispiel)
  - die erzeugte Object- bzw. Hex-Datei kann direkt in den Emulator geladen oder in den Controller programmiert werden

Die **Softhooks** sind z.Zt. auf den **Keil-C51**-Compiler abgestimmt. Der Compiler kann durch ein Erweiterungs-Tool direkt in der Bedienoberfläche **µVision** dazu veranlasst werden, **Softhook**-kompatible Programme zu erzeugen, ohne dass die vorhandenen Sourcedateien geändert werden müssen. Eine Einbindung in Assembler-Programme ist ebenfalls möglich.

Die **Softhook**-Technik von Brendes Datentechnik ist die einzige Emulationstechnik, die es erlaubt, alle aktuellen und zukünftigen 80C51-kompatiblen Controllerderivate unabhängig vom Hersteller zu emulieren. Durch die **BICEPS-Softhooks** ist eine Bondout- oder Hook-Modus-Emulation nicht mehr unbedingt notwendig, obwohl auch diese weiterhin vom **BICEPS-8051**-Emulator unterstützt werden.

Im folgenden ein Beispielprogramm, das zeigt, wie ein einfaches C51-Programm mit P0- und P2-Zugriffen umgesetzt wird und welche P0/P2-Zugriffe vom **BICEPS**-Emulator emuliert werden können.

```
/* *****  
**      Softhook demo program  
**          P2.0=0 : toggle P0.3  
**          P2.0=1 : read P2,  
**                      invert value and transfer to P0  
** *****  
#include <reg52.h>  
  
sbit P2_0 = P2 ^0;  
sbit P0_3 = P0 ^3;  
  
void main (void)  
{  
    while (1)  
    {  
        if (!P2_0)  
        {    P0_3 = 0;  
            P0_3 = 1;  
        }  
        else P0 = P2 ^ 0xFF; // invert  
    }  
}
```

Die **Softhook** Compiler-Erweiterung erkennt inkompatible Zuweisungen (hier: Schreib-Zugriff auf P0). Diese werden ersetzt durch Intrinsic function calls:

```
/* *****  
**      Softhook demo program  
**          P2.0=0 : toggle P0.3  
**          P2.0=1 : read P2,  
**                      invert value and transfer to P0  
** *****  
#include <reg52.h>  
#include <shook.h>  
  
sbit P2_0 = P2 ^0;  
sbit P0_3 = P0 ^3;  
  
void main (void)  
{  
    while (1)  
    {  
        if (!P2_0)  
        {    P0_3 = 0;  
            P0_3 = 1;  
        }  
        else WriteP0(P2 ^ 0xFF); // invert  
    }  
}
```

Die 3 Byte **Softhook** Intrinsic-Funktion wird in den Object-Code eingesetzt, der für Emulator und Controller verwendet wird:

```
; FUNCTION main (BEGIN)
                                ; SOURCE LINE #15
0000      ?C0001:
                                ; SOURCE LINE #17
0000 20A006  JB      P2_0,?C0003
                                ; SOURCE LINE #18
0003 C283   CLR      P0.3
                                ; SOURCE LINE #19
0005 D283   SETB     P0.3
                                ; SOURCE LINE #19
0007 80F7   SJMP     ?C0001
0009      ?C0003:
                                ; SOURCE LINE #21
0009 E5A0   MOV      A,P2
000B F4     CPL      A
000C FF     MOV      R7,A
000D 8F80  MOV      P0,R7    ; 3 BYTES INTRINSIC FUNCTION
000F EF     MOV      A,R7
                                ; SOURCE LINE #15
0010 80EE   SJMP     ?C0001
; FUNCTION main (END)
```

## embetter **BICEPS-8051**

# Emulations-Adapter

Der **BICEPS-8051**-Emulator bietet drei verschiedene Möglichkeiten zur Emulation:

### 1. **BICEPS** Prozessor-Adapter (POD)

Der Prozessor-Adapter (POD) ersetzt den Mikrocontroller der Testschaltung. Auf ihm ist der emulierte Controller untergebracht, so dass nur der POD gewechselt werden muss, falls ein anderer Controller emuliert werden soll.

In der Regel ist der Prozessor das einzige aktive Bauteil auf dem POD. Die Port-Signale sind entweder direkt mit der zu testenden Schaltung verbunden oder werden von der Emulator-Logik erzeugt, falls die Signale für Emulator-interne Zwecke benötigt werden. Dies gilt vor allem für Port0, Port2, /PSEN und ALE.

Die mechanische Anpassung an die Bauform des Controllers (DIL, PLCC oder QFP) erfolgt über einen Converter.

Die Taktgeschwindigkeit wird durch den Emulator bestimmt (wählbar zwischen 1..40 MHz). Außerdem besteht die Möglichkeit, den Takt der Testschaltung zu verwenden.

### 2. **BICEPS** Universal-Adapter

Der Universal-Adapter des **BICEPS**-Emulators kann alle Applikationen mit externem Bus emulieren. Er kontaktiert die Signale P0.0-P0.7, P2.0-P2.7 sowie ALE, PSEN und RESET.

Der Adapter hat das gleiche Pinout wie Standard-Speicher-Bausteine und ist für Applikationen mit externem Programmspeicher gedacht. Er ersetzt den Programmspeicher auf der Testschaltung. Dies gilt auch für Banking-Anwendungen

Gegenüber einem Prozessor-Adapter bieten der Universal-Adapter folgende neue Möglichkeiten:

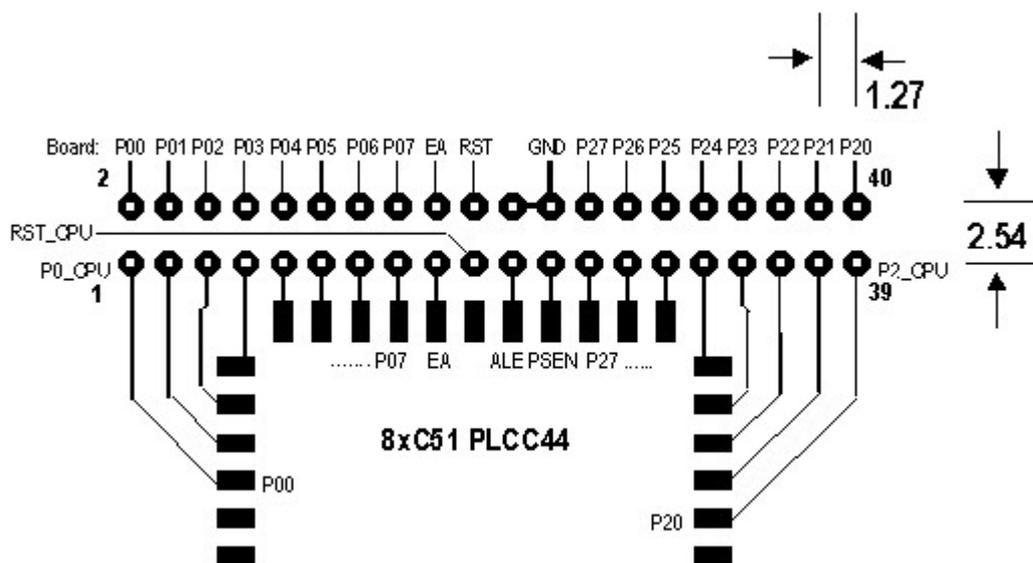
- Der Prozessor bleibt auf der Platine. Dadurch sind die Adapter universell auch für SMD-Versionen, Mini-Module, zukünftige Prozessoren usw. geeignet
- Geringere Abweichung vom Original-Prozessor, keine Kontaktierungsprobleme, Verwendung des Original-Taktes
- Absolut universell für alle 80C51-kompatiblen Controller-Derivate einsetzbar

Trotzdem bleiben die Debugging-Eigenschaften (Zugriff auf alle externen und internen Ressourcen ohne Einschränkungen, Unterbrechungsmöglichkeiten, Real-Time-Trace usw ...) erhalten.

### 3. BICEPS Debug connector

Der **BICEPS**-Emulator bietet die Möglichkeit, den Emulator über eine Stiftleiste (Debug-Connector) auf der zu testenden Schaltung anzuschließen. Als Verbindung zwischen Emulator und Schaltung sind möglich:

- 26-poliges Standard-Flachbandkabel (Stiftleiste RM 2,54) für Anwendungen mit externem Bus
- 34-poliges Standard-Flachbandkabel (Stiftleiste RM 2,54) für Banking-Anwendungen
- 40-poliges High-Density-Kabel (Stiftleiste RM 1,27) sowohl für Bus- als auch für Port-Anwendungen (Softhooks)



Wenn die Schaltung ohne Emulator betrieben werden soll, können an Stelle des Debug-Kabels Kurzschluss-Brücken eingesetzt werden.

## embetter BICEPS-8051

### Liste der unterstützten Bausteine

Vendor	Controller	Mode	POD	Debug Connector	Universal Adapter	
All	80C31/32	Bus	B51	✓	✓	
	8xC51/52/54/55/58	Port (SH)		✓		
		Port(HO)				
Winbond	77C32 77x58 77x516 77x532	Bus	B51	✓	✓	
		Port (SH)		✓		
	77x468	Bus		✓	✓	
		Port (SH)		✓		
	78x32/33 78x51/52/54/58 78x516 78x365 78x858	Bus	B51	✓	✓	
		Port (SH)		✓		
		78x438		Bus	✓	✓
	Texas Instruments	MSC 121x	Bus		✓	✓
			Port (SH)		✓	
Maxim / Dallas	80C320, 80C323 87C520, 87C530	Bus	B51	✓	✓	
	80C390 standard, paged, contiguous	Bus	B390	✓	✓	
	80C400, 80C410	Bus			✓	
Atmel	89C5131/5132	Bus	B5131USB	✓	✓	
		Port(SH)		✓		
	89C51CC01 89C51CC03	Bus	B51CC or B51CC-52	✓	✓	
		Port (SH)		✓		
	89C51AC2	Bus		✓	✓	
		Port (SH)		✓		
	89C51CC02 89C5115	Port (SH)	B51CC + CP4428			
	8xC51RA2/RB2/IB2 8xC51RC2/IC2 8XC51RD2ED2 8xC51U2 8xC51/52/54/58X2	Bus	B51 or B51-68	✓	✓	
		Port (SH)		✓		
	89C1051/2051/4051	Port (BO)	B51			
	AT89S8252 AT89LS8252 AT89S8253	Bus	B51	✓	✓	
		Port (SH)		✓		
AT89C51SND1	Bus	B51SND1	✓	✓		
	Port (SH)		✓			

Vendor	Controller	Mode	POD	Debug Connector	Universal Adapter
NXP / Philips	8xC5x 8xC51/52/54/58/31/32X2 89C6xX2 8xC591 89C66x, 8966xX2	Bus	B51	✓	✓
		Port (HO)			
		Port (SH)		✓	
	8xC552 8xC554 8xC562	Bus	B552	✓	✓
		Port (HO)			
		Port (SH)		✓	
	8xC592	Bus	B592	✓	✓
		Port (HO)			
		Port (SH)		✓	
	89C51RA+/RB+ 89C51RC+/RD+ 89C51RA2/RB2 89C51RC2/RD2	Bus	B51	✓	✓
		Port (HO)			
		Port (SH)		✓	
Infineon (Siemens)	C501, C502, C504 C505C, C505CA C541	Bus	B51	✓	✓
		Port (SH)		✓	
	C515, C515C, 80C515A, 80C535	Bus	B517 +CP8468	✓	✓
		Port (SH)		✓	
	C517, C517A 80C537, 80C517A	Bus	B517	✓	✓
		Port (SH)		✓	
	C509	Bus		✓	✓
		Port (SH)		✓	
Analog Devices	ADuC812/816/824	Bus	BuC8xx	✓	✓
		Port(SH)		✓	
SST	89x554 89x564	Bus		✓	✓
		Port(SH)		✓	

Single chip emulation: SH = Softhook, HO = Hardware Hook

**embetter**

## Kontakt

**BRENDES DATENTECHNIK GMBH**

Dresdener Str. 10

D-26160 Bad Zwischenahn

Germany

Tel.: +49 (0) 4403 816838

FAX: +49 (0) 4403 816839

eMail: [info@brendes.de](mailto:info@brendes.de)

Web: <http://www.brendes.de>