

PA-H8/3332-(T)80 Data Sheet

80 pin QFP(TQFP) socket/28 pin DIP 0.6" plug

Supported Device/Footprints

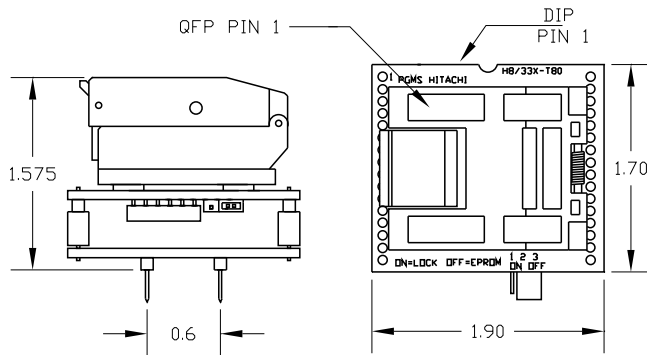
These adapters allow EPROM programming of Hitachi H8/3332 & H8/3334 devices in their QFP or TQFP packages on 28 pin DIP programmers.

Mfgr	Device		Footprint	
	Device	Package	Device	Plug
Hitachi	H8/3332, H8/3334	QFP or TQFP	27C256	28 pin DIP

The QFP package supported has a 14mm square body and 0.65mm lead spacing. Hitachi package code FP-80A

The TQFP package supported has a 12mm square body and 0.5mm lead spacing. Hitachi package code TFP-80A

Adapter Dimensions



Adapter Parts & Part Numbers

The following chart shows the various socket and board part numbers that make up these adapters.

Adapter	Socket	Top Board	Bottom Board
PA-H8/3332-80	80QG-02A	H8/33X-80	H8/53X-28
PA-H8/3332-T80	80QJ-808	H8/33X-T80	H8/53X-28

Adapter Construction

The adapter is made up of 3 sub-assemblies. They assemble via connectors making the adapter modular. This way the sub-assemblies can be replaced when they wear out.

When disassembling the adapter take care not to bend the pins. When reassembling the adapter note the pin 1 indicators to align the parts correctly.

Test Socket

QFP Lidded ZIF test socket:
 Enplas Part #: FPQ-80-0.65-02 LSC #: 80QG-02A
 TQFP Lidded ZIF test socket:
 Yamaichi Part #: IC51-0804-808 LSC #: 80QJ-808

H8/33X-(T)80

Accepts the test socket and connects to the 32 pin header footprint. Provides circuitry for security programming.

H8/53X-28

Connects the top board to the programmer and the 27C256 footprint.

EPROM Security

The Hitachi H8 family provides EPROM security. By programming one security location four EPROM bytes are protected. After programming, the protected bytes read 00h and cannot be programmed.

To program security locations, install the security jumper and program the device; the data value is ignored. The security jumper is marked SW1 on the top board. Remove the jumper to allow EPROM programming.

SW1 jumper setting

Pins	Mode	Function
1-2	ON	Program Lock Bits
2-3 or None	OFF	Program/Read EPROM

To secure the whole device it is simplest to install the jumper and program the whole device to 00.

To secure one block of EPROM (4 bytes) program any byte of the block with the security jumper installed. The block addresses begin at 0 and are arranged at 0-3, 4-7, 8-B, C-F and so on through all the memory.

The security locations are EPROM. On erasable devices they are erased when the EPROM is erased. On One-Time Programmable devices they cannot be erased.

EPROM Address Mapping

The H8's EPROM is at the same address for MCU mode (normal execution) and EPROM programming. The valid EPROM addresses for each device are as follows.

Device	EPROM size	EPROM addresses	MCU addresses
H8/3332	16K bytes	0000-3FFF	0000-3FFF
H8/3334	32K bytes	0000-7FFF	0000-7FFF

The H8/3332 has less memory than the standard EPROM it emulates. The following table shows the device, its EPROM programming footprint, and the unused addresses of that footprint. Those addresses can be accessed by the EPROM programmer. When those addresses are read the value will be either FF or undetermined. When programming, either prevent those addresses from being programmed or be sure to set the data for those unused addresses to FF.

Device	Footprint	Unused addresses	Reads	Program to
H8/3332	27C256	4000-7FFF	FF	FF
H8/3334	27C256	NONE	-	-

Adapter Wiring

The following chart shows the connections from the QFP device to the adapter's DIP plug.

SOCKET	PLUG	SOCKET	PLUG
1	1	41	-
2	-	42	-
3	-	43	-
4	14	44	-
5	14	45	-
6	24	46	-
7	14	47	28
8	28	48	20
9	-	49	27
10	-	50	26
11	-	51	2
12	14	52	23
13	-	53	21
14	-	54	22
15	-	55	25
16	-	56	14
17	-	57	3
18	-	58	4
19	-	59	5
20	-	60	6
21	-	61	7
22	-	62	8
23	-	63	9
24	10K +	64	10
25	10K +	65	11
26	-	66	12
27	-	67	13
28	-	68	15
29	-	69	16
30	-	70	17
31	-	71	18
32	-	72	19
33	-	73	14
34	-	74	28
35	-	75	SW1-2
36	-	76	-
37	-	77	-
38	14	78	-
39	-	79	-
40	-	80	-

VCC (28) to GND (14) .1uf bypass
 10K + 10K Pullup
 SW1 Jumper settings:
 1-2 for ON (Program Lock Bits)
 2-3 (or none) OFF (Program/Read EPROM)

