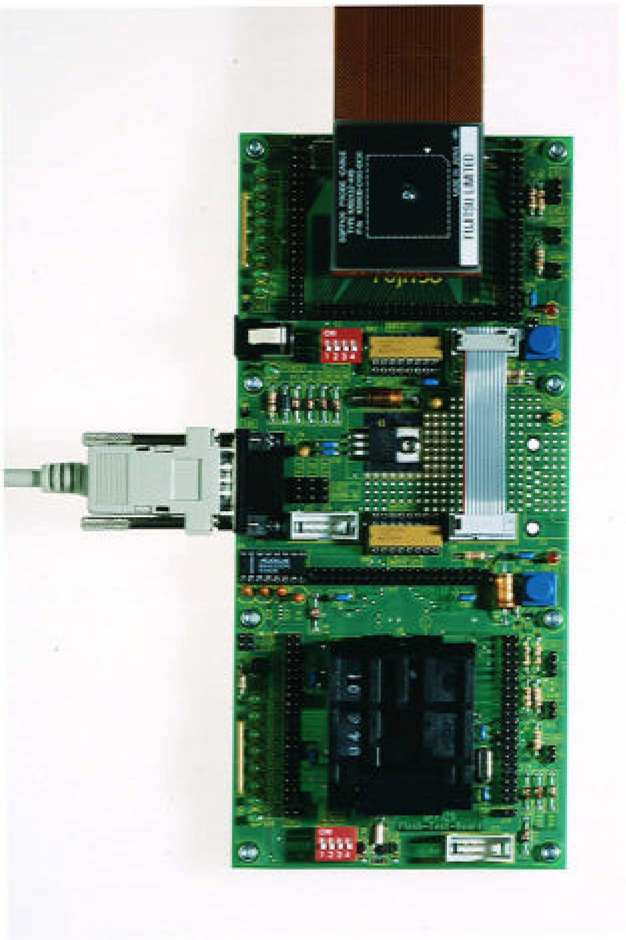
QFP120 Flash-Test-Board Documentation

© Fujitsu Mikroelektronik GmbH
Am Siebenstein 6-10
63303 Dreieich-Buchschlag, Germany



Revision: **3.2**
Date: **30/07/1999**

Warranty and Disclaimer

To the maximum extent permitted by applicable law, Fujitsu Mikroelektronik GmbH restricts its warranties and its liability for the Product **QFP120 Flash-Test-Board (FLASH-EVA2-120P-M13)**, its performance and any consequential damages, on the use of the Product in accordance with (i) the terms of the License Agreement and the Sale and Purchase Agreement under which agreements the Product has been delivered, (ii) the technical descriptions and (iii) all accompanying written materials. In addition, to the maximum extent permitted by applicable law, Fujitsu Mikroelektronik GmbH disclaims all warranties and liabilities for the performance of the Product and any consequential damages in cases of unauthorised decompiling and/or reverse engineering and/or disassembling.

1. Fujitsu Mikroelektronik GmbH warrants that the Product will perform substantially in accordance with the accompanying written materials [this manual] for a period of 90 days form the date of receipt by the customer. Concerning the hardware components of the Product, Fujitsu Mikroelektronik GmbH warrants that the Product will be free from defects in material and workmanship under use and service as specified in the accompanying written materials for a duration of 1 year from the date of receipt by the customer.

2. Should a Product turn out to be defect, Fujitsu Mikroelektronik GmbH´s entire liability and the customer´s exclusive remedy shall be, at Fujitsu Mikroelektronik GmbH´s sole discretion, either return of the purchase price and the license fee, or replacement of the Product or parts thereof, if the Product is returned to Fujitsu Mikroelektronik GmbH in original packing and without further defects resulting from the customer´s use or the transport. However, this warranty is excluded if the defect has resulted from an accident not attributable to Fujitsu Mikroelektronik GmbH, or abuse or misapplication attributable to the customer or any other third party not relating to Fujitsu Mikroelektronik GmbH.

3. To the maximum extent permitted by applicable law Fujitsu Mikroelektronik GmbH disclaims all other warranties, whether expressed or implied, in particular, but not limited to, warranties of merchantability and fitness for a particular purpose for which the Product is not designated.

4. To the maximum extent permitted by applicable law, Fujitsu Mikroelektronik GmbH´s and its suppliers´ liability is restricted to intention and gross negligence.

   NO LIABILITY FOR CONSEQUENTIAL DAMAGES

   To the maximum extent permitted by applicable law, in no event shall Fujitsu Mikroelektronik GmbH and its suppliers be liable for any damages whatsoever (including but without limitation, consequential and/or indirect damages for personal injury, assets of substantial value, loss of profits, interruption of business operation, loss of information, or any other monetary or pecuniary loss) arising from the use of the Product.

Should one of the above stipulations be or become invalid and/or unenforceable, the remaining stipulations shall stay in full effect.

Contents

Note, this board must only be used for test applications in labatory environment

Hardware description of QFP120 Flash-Test-Board

Overview

The QFP120 Flash-Test-Board is a low cost multifunctional evaluation board for the Fujitsu MB90520 and MB90570 Flash microcontroller series. It can be used as a low cost target system for software development especially for Flash programming evaluation.

Fujitsu´s Flash microcontrollers can be programmed by three methods:
1) An ordinary EPROM programmer with adapter is used as with OTP devices
2) Using Fujitsu embedded serial programming mode via on chip UART
3) Downloading program code to Flash memory using customer´s own bootstrap software

The board supports the two methods of in-circuit Flash programming (method 2 and 3).
Besides that, it is possible to connect an emulator as well as using it as a stand-alone application board.
The board allows the designer immediately to start software development before his own final target system is available.
The order code of the board is FLASH-EVA2-120P-M13.

Features

- Low cost target board for 16bit LX series in the QFP120 package, e.g. MB90520 and MB90570
- Easy evaluation of different in-circuit flash programming modes
- All resources of the LX microcontroller family are available for evaluation
- Burn-in socket for stand-alone application test
- Prototyping area

Board Description

For references mentioned in this chapter please see also the assembly drawing on page 13.
The QFP120 Flash-Test-Board is designed to have two independent evaluation areas. In the upper half there is a 120 pin SQFP socket which allows you to connect the probe cable of the in-circuit emulator. It would be also possible to assemble the board with a programmed MCU device instead of this socket, the footprint is the same. In the bottom area of the board there is a 120 pin burn-in socket which either can be used to program a MCU Flash device or to execute code in an already programmed device. As already mentioned above, programming of a MCU Flash device can be done with an external programmer (see figure 8), e.g. AF200 from Yokogawa, or if both evaluation areas are used in conjunction (see figure 7). In this case one MCU executes the serial programming algorithm and the burn-in socket contains the device which is going to be programmed via the on chip serial interface.
Between the upper and lower section of the board there is the power supply connector, the 5V voltage regulator and a 9 pin DSUB connector which can be connected to *UART CH0* of either the SQFP- or the burn-in socket. This selection is done by setting three jumpers in the appropriate position. The figure below shows the UART configured to be connected to the SQFP120 socket (default).

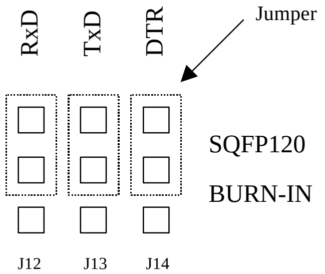


Figure 1: Jumper J12, J13 and J14 connect the UART to either SQFP120 or Burn-In socket

In addition the RS232 signal DTR (Data Terminal Ready) or the inverted /DTR can be selected to reset the selected evaluation device (J20). This selection is done as shown in the figure below. As default this jumper is left open.

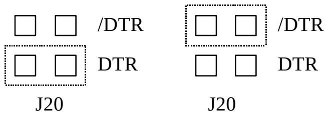


Figure 2: Selection of DTR signal polarity

Furthermore there is a small prototyping area with a footprint of a second 9 pin DSUB connector for application purposes. A 10 pin connector with power supply, i.e. VCC and ground (GND), is provided (J15). Also several pads with VCC are available to supply additional circuitry. The solder mask on the bottom side has a cut out to make shortest connections to the GND copper plane possible.
The points, which are mentioned from now on, are available for each section separately.
All microcontroller I/O pins are connected to edge connectors which allow interfacing to any application environment.
In addition eight port LEDs support simple program debugging.
A reset switch and a power supervisory device ensure proper power on reset conditions.

Three DIP switches let the user choose one of the available memory access modes. For further details please look in the memory section of the respective hardware manual or onto the schematic of the board. The following table gives you the relationship between the DIP switch settings and the logic levels of the mode pins. Note that there is a difference in the circuitry between MD0 and MD1,2 so that the programming device is able to overdrive the user selected mode for programming purposes (See also figure 9 in the next chapter and the note below figure 8).

	DIP switch setting	Logical value
MD0	ON (closed)	0 (low)
	OFF (open)	1 (high)
MD1	ON (closed)	1 (high)
	OFF (open)	0 (low)
MD2	ON (closed)	1 (high)
	OFF (open)	0 (low)

▨ default

Figure 3: DIP switch settings and polarities (default MD2,1,0 = 011, single chip mode)

Sub- and main-clock crystals are available for both, the SQFP120- and the Burn-in socket to evaluate easily the different clock and power saving modes.

Two resistor networks can be placed in two positions each to choose between a MB90F520 or a MB90F570 device. The respective settings correct the different pinouts of the supported series.

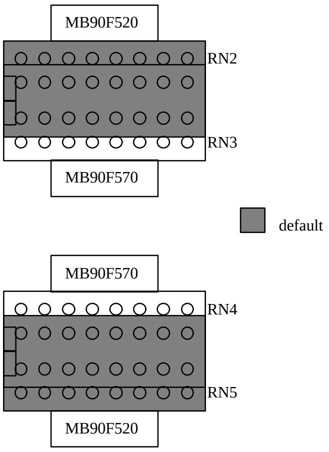


Figure 4: Selectors of MCU series

The reset signal, either derived from the reset switch or any programmer via the UP connector can be used as the RST (reset) or HST (hardware standby) signal of the microcontroller. J4 and J21 do that selection. The default setting is RST as shown in the following figure.

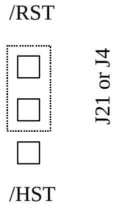


Figure 5: Reset selector

Two jumpers are provided to simply connect the analogue reference signals AVRL and AVRH to Vss and Vcc respectively, but they also allow the user to supply a seperate reference voltage for the analogue to digital converter. The jumpers are set per default as you can see below.
JP1 and JP2 below to the upper SQFP120 socket and JP3, JP4 to the lower evaluation area with the burn-in socket.

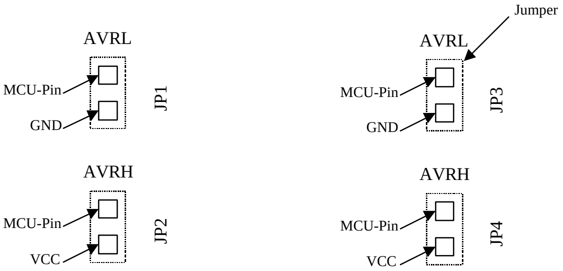


Figure 6: Jumpers to supply the analogue reference voltage

Serial Programming Interface

On the QFP120 Flash-Test-Board there are three ribbon cable connectors which comply to the below definition of the serial programming interface.

Pin Number of 10pin connector	**DOWN** connector	**UP** connector	Description
6	SIN1	SOT0	Ser. I/F
4	SOT1	SIN0	
5	SCK1	SCK0	
8	P00	P00	Programming startup pin
9	P01	/TICS	*)
7	P02	Reset	Reset of the MCU
2	P03	MD0	Mode pins
1	P04	MD1	
3	P05	MD2	
10	V_{SS}	V_{SS}	Signal ground

Table 1: Definition of serial programming interface

*) The user circuitry can be cut off by using the /TICS as a multiplexer control signal during serial programming. On the QFP120 Flash-Test-Board this signal is used to drive the yellow LED *PROGR.*, which indicates, that the serial flash programming is in progress.

The figures below illustrate the philosophy of the UP and DOWN connectors as well as the different configurations, in which the Flash-Test-Board can be used.

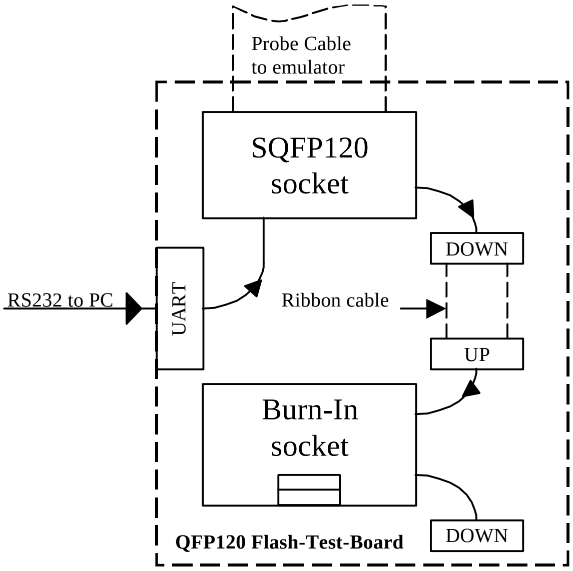


Figure 7: A one by one connection of the DOWN to the UP connector makes it possible to program a Flash device in the Burn-In socket. In this case either the emulator or a MCU, which is assembled instead of the SQFP120 socket, works as a programmer and executes the serial programming algorithm. Dedicated software must run on the emulator or MCU for this purpose.

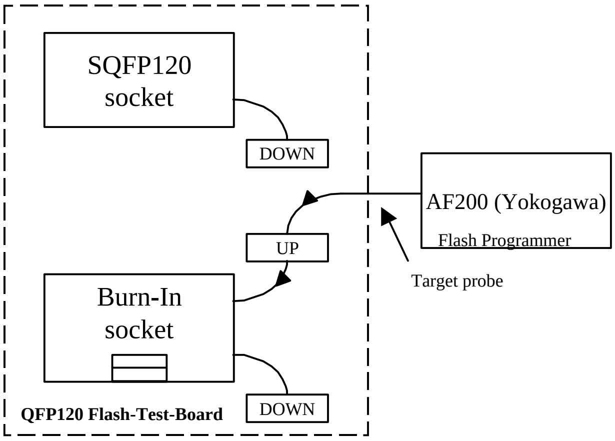


Figure 8: In this configuration an external programmer like AF200 from Yokogawa is used to program the Flash MCU.

Note: If an external programmer like Yokogawa AF200 is used, be sure that the DIP switches of the Burn-In socket, to set the mode pins, are in open position. Otherwise the programmer might be damaged. This is not necessary if you use this board as shown in figure 7.

If the user adds additional circuitry to his target board in order to establish an UP connector, he can program the Flash MCU via the DOWN connector of the QFP120 Flash-Test-Board, as it is shown in figure 11. The user of course does not have to provide switches to the mode pins as it shown below and as it done on this Flash-Test-Board in order to keep the flexibility of mode settings. Only the mode pins which have to be configured different from MD2,1,0 = 110 have to be supplied with pull-up or pull-down resistors respectively and have to be connected to the UP connector.

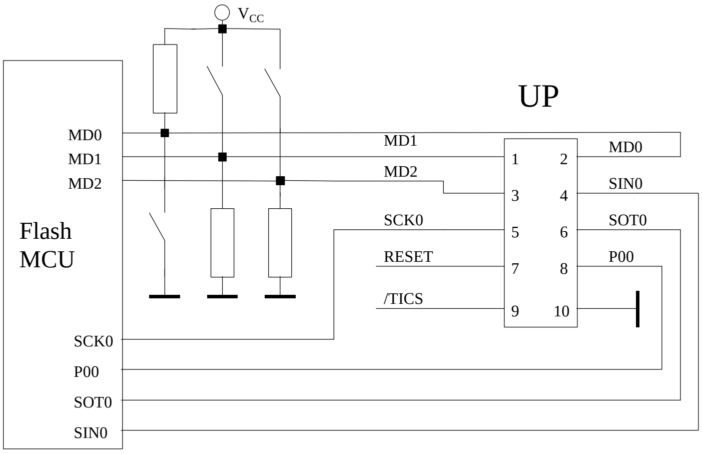


Figure 9: UP connector of QFP120 Flash-Test-Board

The next figure, figure 10, shows an application mode pin setting of MD2,1,0 = 011. The programmer, i.e. the QFP120 Flash-Test-Board or an external programmer like AF200 from Yokogawa however, is still able to drive the mode pins to MD2,1,0 = 110. This is the setting for the embedded serial programming mode.

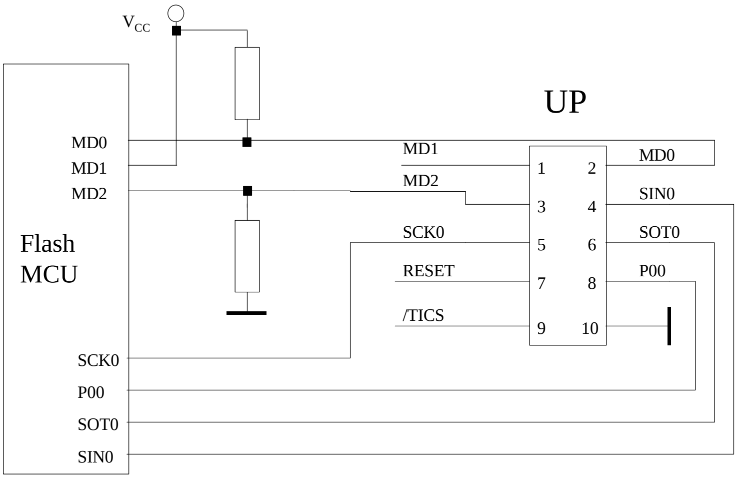


Figure 10: UP connector with mode pin setting of MD2,1,0 = 011 (single chip mode)

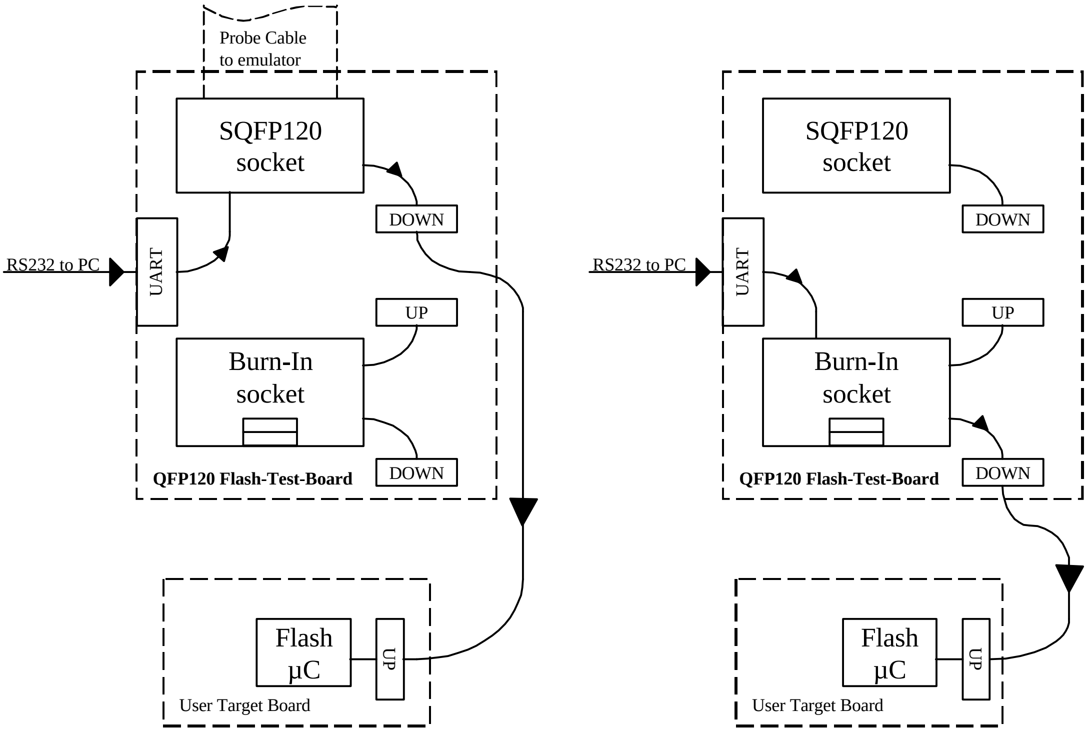
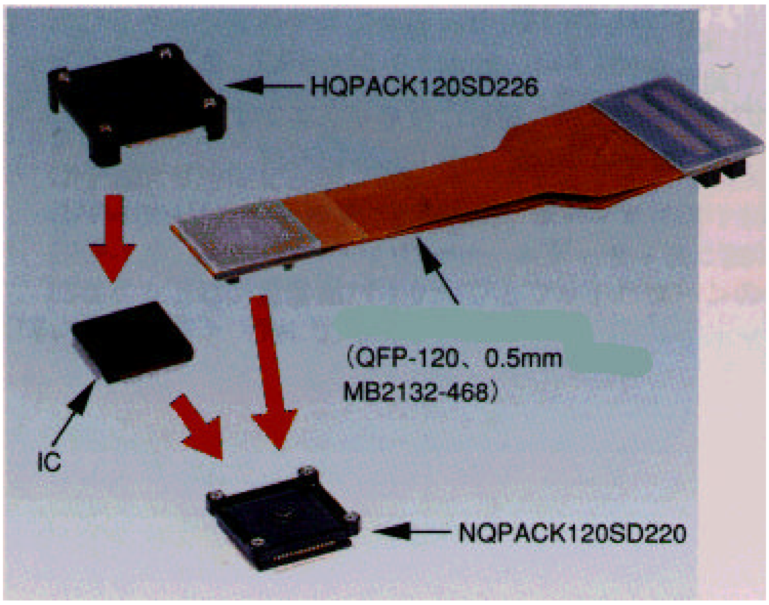


Figure 11: The DOWN connector to program the Flash MCU on the user target system is provided for both, the SQFP120 and Burn-In socket. Therefore the serial programming algorithm can be run from emulator or from programmed MCU device which is assembled instead of the SQFP120 socket (see left side) or put into the Burn-In socket (see right side).

Related items to QFP120 Flash-Test-Board

1. FLASH-EVA2-120P-M13, QFP120 Flash-Test-Board
2. MB90F523PFV, Flash MCU device 0,5mm pitch
3. MB2132-468, Probe cable
4. NQPACK120SD220 (Socket to connect probe cable or to carry IC)
5. HQPACK120SD226 (Cap for IC)



Pin header cross reference list

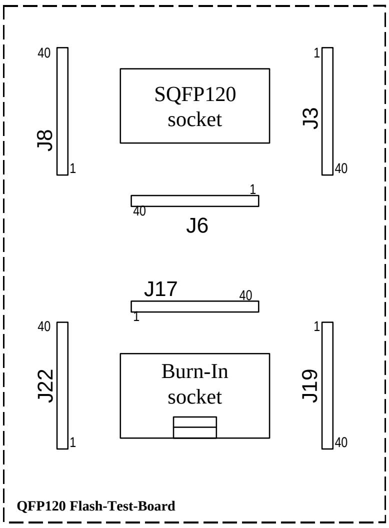


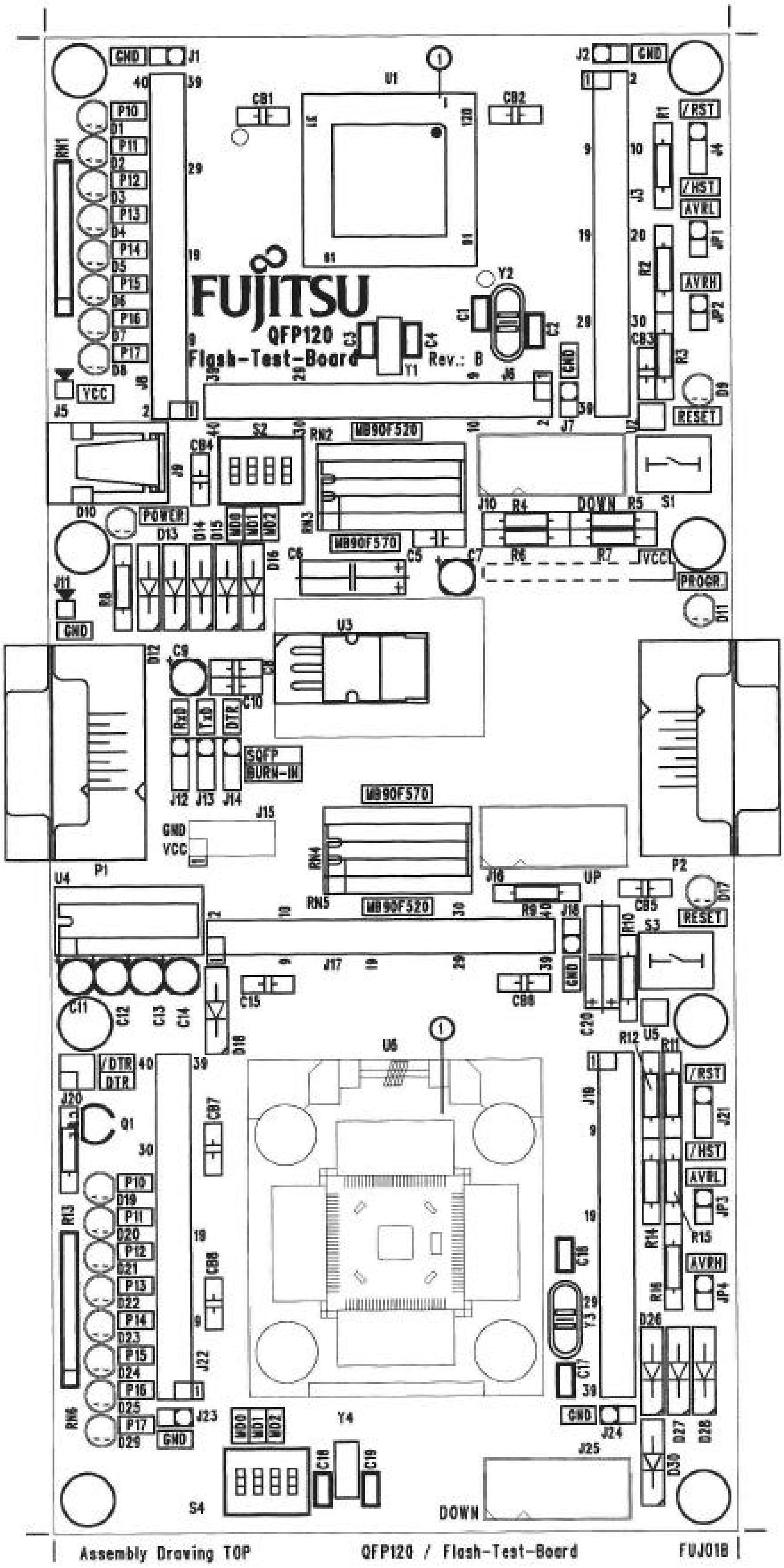
Figure 12: Pin header layout on QFP120 Flash Test Board

MCU Pin No.	SQFP120 socket area	Burn-In socket area	MB90F520	MB90F570
1	J3 / 15	J17 / 35	P31 / CKOT	P31 / RDX
2	J3 / 14	J17 / 34	P32 / OUT0	P32 / WRLX
3	J3 / 13	J17 / 33	P33 / OUT1	P3 / WRHX
4	J3 / 12	J17 / 32	P34 / OUT2	P34 / HRQ
5	J3 / 11	J17 / 31	P35 / OUT3	P35 / HAKX
6	J3 / 10	J17 / 30	P36 / PPG00	P36 / RDY
7	J3 / 9	J17 / 29	P37 / PPG01	P37 / CLK
8	J3 / 8	J17 / 28	Vcc	Vcc
9	J3 / 7	J17 / 27	P40 / PPG10	P40 / SIN0
10	J3 / 6	J17 / 26	P41 / PPG11	P41 / SOT0
11	J3 / 5	J17 / 25	P42 / SIN0	P42 / SCK0
12	J3 / 4	J17 / 24	P43 / SOT0	P43 / SIN1
13	J3 / 3	J17 / 23	P44 / SCK0	P44 / SOT1
14	J3 / 2	J17 / 22	P45 / SIN1	P45 / SCK1
15	J3 / 1	J17 / 21	P46 / SOT1	P46 / PPG0
16	J8 / 40	J17 / 20	P47 / SCK1	P47 / PPG1
17	J8 / 39	J17 / 19	SEG00	P50 / SIN2
18	J8 / 38	J17 / 18	SEG01	P51 / SOT2
19	J8 / 37	J17 / 17	SEG02	P52 / SCK2
20	J8 / 36	J17 / 16	SEG03	P53 / SIN3
21	J8 / 35	J17 / 15	SEG04	P54 / SOT3
22	J8 / 34	J17 / 14	SEG05	P55 / SCK3
23	J8 / 33	J17 / 13	SEG06	P56 / IN0
24	J8 / 32	J17 / 12	SEG07	P57 / IN1
25	J8 / 31	J17 / 11	PA0 / SEG08	P60 / SIN4
26	J8 / 30	J17 / 10	PA1 / SEG09	P61 / SOT4
27	J8 / 39	J17 / 9	PA2 / SEG10	P62 / SCK4
28	J8 / 28	J17 / 8	PA3 / SEG11	P63 / CKOT
29	J8 / 27	J17 /7	PA4 / SEG12	P64 / OUT0
30	J8 / 26	J17 /6	PA5 / SEG13	P65 / OUT1
31	J8 / 25	J17 /5	PA6 / SEG14	P66 / OUT2
32	J8 / 24	J17 /4	PA7 / SEG15	P67 / OUT3
33	J8 / 23	J17 /3	Vss	Vss

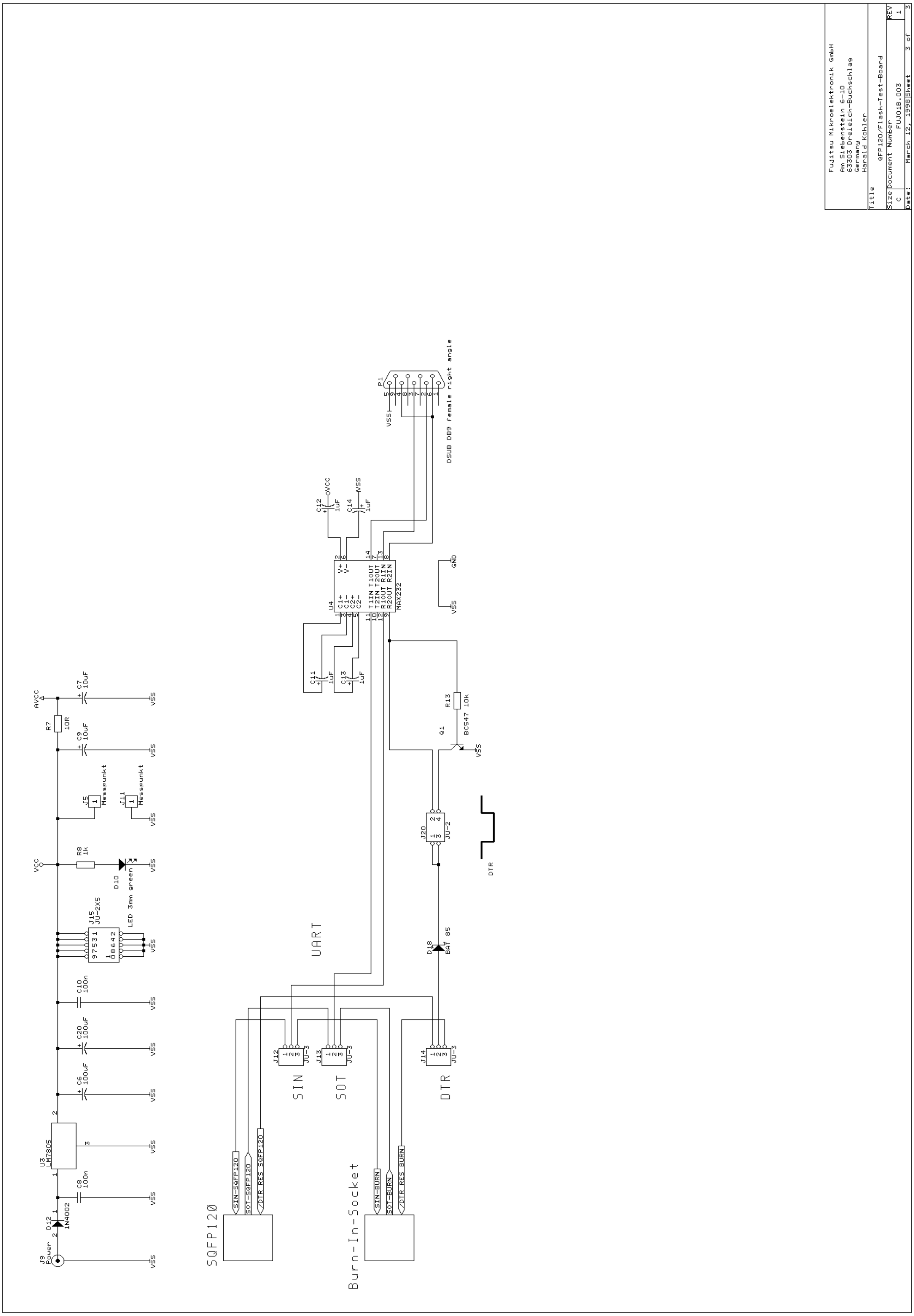
34	J8 / 22	J17 / 2	C	C
35	J8 / 21	J17 / 1	P50 / SIN2	P70
36	J8 / 20	J22 / 40	P51 / SOT2	P71
37	J8 / 29	J22 / 39	P52 / SCK2	P72
38	J8 / 18	J22 / 38	DVcc	DVcc
39	J8 / 17	J22 / 37	DVss	DVss
40	J8 / 16	J22 / 36	P53 / DA0	P73 / DA0
41	J8 / 15	J22 / 35	P54 / DA1	P74 / DA1
42	J8 / 14	J22 / 34	Avcc	AVcc
43	J8 / 13	J22 / 33	AV_{RH}	AV_{RH}
44	J8 / 12	J22 / 32	AV_{RL}	AV_{RL}
45	J8 / 11	J22 / 31	AV_{SS}	AV_{SS}
46	J8 / 10	J22 / 30	P60 / AN0	P80 / AN0
47	J8 / 9	J22 / 29	P61 / AN1	P81 / AN1
48	J8 / 8	J22 / 28	P62 / AN2	P82 / AN2
49	J8 / 7	J22 / 27	P63 / AN3	P83 / AN3
50	J8 / 6	J22 / 26	P64 / AN4	P84 / AN4
51	J8 / 5	J22 / 25	P65 / AN5	P85 / AN5
52	J8 / 4	J22 / 24	P66 / AN6	P86 / AN6
53	J8 / 3	J22 / 23	P67 / AN7	P87 / AN7
54	J8 / 2	J22 / 22	Vcc	Vcc
55	J8 / 1	J22 / 21	P70 / TI0 / OUT4	P90 / CS0
56	J6 / 40	J22 / 20	P71 / TI1 / OUT5	P91 / CS1
57	J6 / 39	J22 / 19	P72 / TO0 / OUT6	P92 / CS2
58	J6 / 38	J22 / 18	P73 / TO1 / OUT7	P93 / CS3
59	J6 / 37	J22 / 17	P74 / COM0	P94 / CS4
60	J6 / 36	J22 / 16	P75 / COM1	P95 / CS5
61	J6 / 35	J22 / 15	P76 / COM 2	P96 / CS6
62	J6 / 34	J22 / 14	P77 / COM3	P97 / CS7
63	J6 / 33	J22 / 13	Vss	Vss
64	J6 / 32	J22 / 12	P80 / SEG16	PA0 / AIN0 / IRQ6
65	J6 / 31	J22 / 11	P81 / SEG17	PA1 / BIN0
66	J6 / 30	J22 / 10	P82 / SEG18	PA2 / ZIN0
67	J6 / 29	J22 / 9	P83 / SEG19	PA3 / AIN1
68	J6 / 28	J22 / 8	P84 / SEG 20	PA4 / BIN1
69	J6 / 27	J22 / 7	P85 / SEG21	PA5 / ZIN1
70	J6 / 26	J22 / 6	P86 / SEG22	PA6 / SDA
71	J6 / 25	J22 / 5	P87 / SEG23	PA7 / SCL
72	J6 / 24	J22 / 4	P90 / SEG24	PB0 / IRQ0
73	J6 / 23	J22 / 3	X1A	X1A
74	J6 / 22	J22 / 2	X0A	X0A
75	J6 / 21	J22 / 1	P91 / SEG25	PB1 / IRQ1
76	J6 / 20	J19 / 40	P92 / SEG26	PB2 / IRQ2
77	J6 / 19	J19 / 39	P93 / SEG27	PB3 / IRQ3
78	J6 / 18	J19 / 38	P94 / SEG28	PB4 / IRQ4
79	J6 / 17	J19 / 37	P95 / SEG29	PB5 / IRQ5
80	J6 / 16	J19 / 36	P96 / SEG30	PB6 / ADTG
81	J6 / 15	J19 / 35	P97 / SEG31	PB7
82	J6 / 14	J19 / 34	V0	PC0
83	J6 / 13	J19 / 33	V1	PC1
84	J6 / 12	J19 / 32	V2	PC2
85	J6 / 11	J19 / 31	V3	PC3
86	J6 / 10	J19 / 30	/HST	HSTX
87	J6 / 9	J19 / 29	MD2	MD2
88	J6 / 8	J19 / 28	MD1	MD1
89	J6 / 7	J19 / 27	MD0	MD0
90	J6 / 6	J19 / 26	/RST	RSTX
91	J6 / 5	J19 / 25	Vss	Vss
92	J6 / 4	J19 / 24	X0	X0

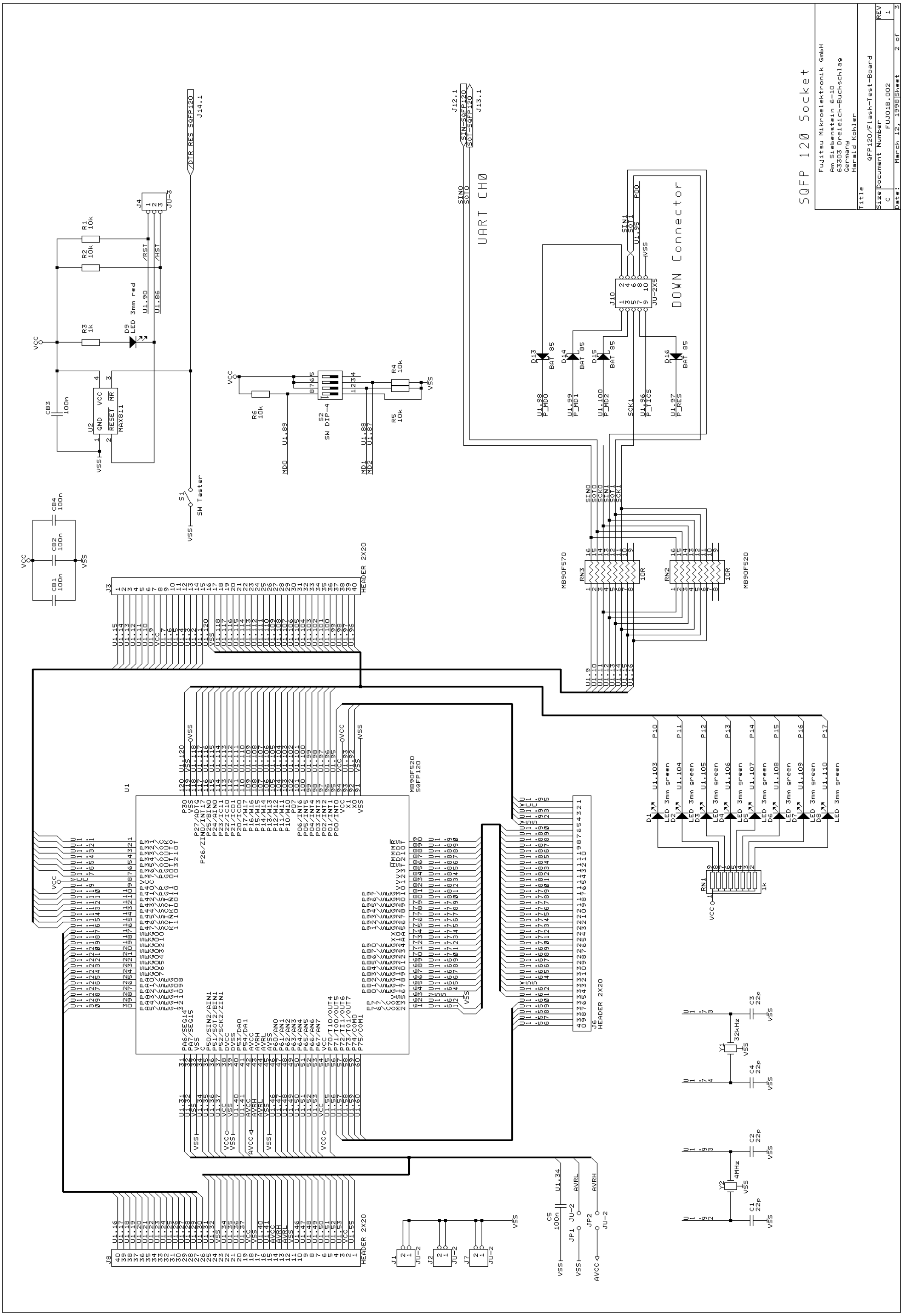
93	J6 / 3	J19 / 23	X1	X1
94	J6 / 2	J19 / 22	Vcc	Vcc
95	J6 / 1	J19 / 21	P00 / INT0	P00 / AD00
96	J3 / 40	J19 / 20	P01 / INT1	P01 / AD01
97	J3 / 39	J19 / 19	P02 / INT2	P02 / AD02
98	J3 / 38	J19 / 18	P03 / INT3	P03 / AD03
99	J3 / 37	J19 / 17	P04 / INT4	P04 / AD04
100	J3 / 36	J19 / 16	P05 / INT5	P05 / AD05
101	J3 / 35	J19 / 15	P06 / INT6	P06 / AD06
102	J3 / 34	J19 / 14	P07	P07 / AD07
103	J3 / 33	J19 / 13	P10 / WI0	P10 / AD08
104	J3 / 32	J19 / 12	P11 / WI1	P11 / AD09
105	J3 / 31	J19 / 11	P12 / WI2	P12 / AD10
106	J3 / 30	J19 / 10	P13 / WI3	P13 / AD11
107	J3 / 29	J19 / 9	P14 / WI4	P14 / AD12
108	J3 / 28	J19 / 8	P15 / WI5	P15 / AD13
109	J3 / 27	J19 / 7	P16 / WI6	P16 / AD14
110	J3 / 26	J19 / 6	P17 / WI7	P17 / AD15
111	J3 / 25	J19 / 5	P20 / IC00	P20 / A16
112	J3 / 24	J19 / 4	P21 / IC01	P21 / A17
113	J3 / 23	J19 / 3	P22 / IC10	P22 / A18
114	J3 / 22	J19 / 2	P23 / IC11	P23 / A19
115	J3 / 21	J19 / 1	P24 / AIN0	P24 / A20
116	J3 / 20	J17 / 40	P25 / BIN0	P25 / A21
117	J3 / 19	J17 / 39	P26 / ZIN0	P26 / A22
118	J3 / 18	J17 / 38	P27 / ADTG	P27 / A23
119	J3 / 17	J17 / 37	Vcc	Vss
120	J3 / 16	J17 / 36	P30	P30 / ALE

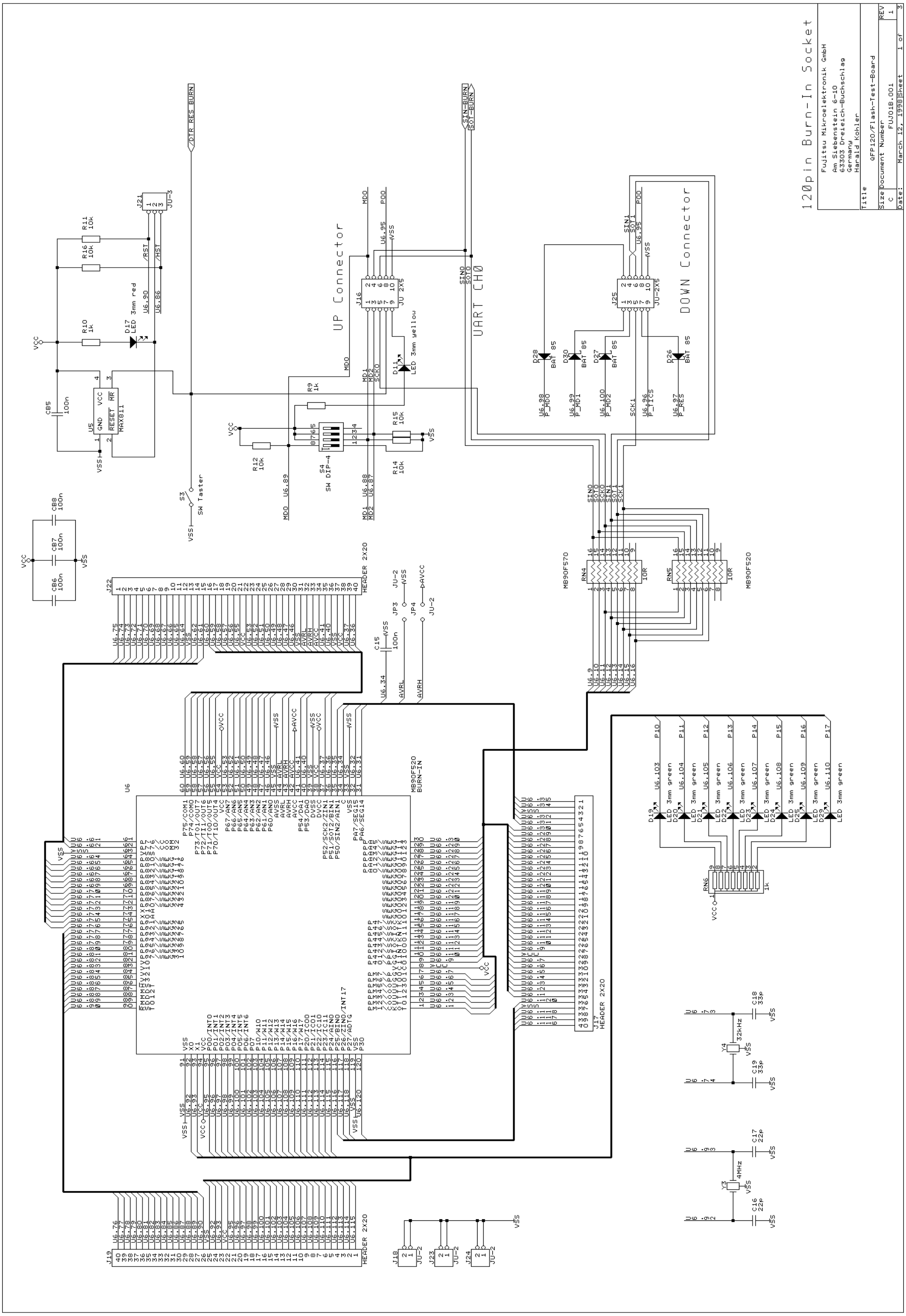
Assembly drawing of QFP120 Flash Test Board



Schematics of QFP120 Flash Test Board




SQFP 120 Socket
Fujitsu Mikroelektronik GmbH
Am Siebenstein 6-10
63303 Dreieich-Buchschlag
Germany
Harald Kohler
Title
QFP120/Flash-Test-Board
Size
C
Document Number
FUJ01B.002
REV
1
Date:
March 12, 1998
Sheet
2
of
3
UART CH0
DOWN Connector
U1
MB90F520
SQFP120
J3
HEADER 2X20
J6
HEADER 2X20
J8
HEADER 2X20
J10
JU-2X5
J4
JU-3
U2
MAX811
S1
SW Taster
S2
SW DIP-4
RN1
1k
RN2
10R
MB90F520
RN3
10R
MB90F570
Y1
32kHz
Y2
4MHz
D9
LED 3mm red
LED 3mm green
BAT 85
/DTR_RES_SQFP120
J14.1
SIN-SQFP120
SOT-SQFP120
J12.1
J13.1
VCC
VSS
AVCC
AVRH
AVRL
JP1
JP2
JU-2



120Øpin Burn-In Socket
Fujitsu Mikroelektronik GmbH
Am Siebenstein 6-10
63303 Dreieich-Buchschlag
Germany
Harald Kohler
Title
QFP120/Flash-Test-Board
Size C
Document Number FUJ01B.001
REV 1
Date: March 12, 1998
Sheet 1 of 3
UP Connector
DOWN Connector
UART CHØ
/DTR RES BURN
SIN-BURN
SOT-BURN
MAX811
SW DIP-4
SW Taster
MB90F520 BURN-IN
MB90F570
MB90F520
HEADER 2X20
LED 3mm red
LED 3mm yellow
LED 3mm green
BAT 85
32kHz
4MHz

Software documentation

Overview

This manual shows how to use the flash-programming software, provided with the QFP-120 Flash Test Board (FLASH-EVA2-120P-M13). Follow these instructions step by step in order to reprogram an MB90F523 Flash-microcontroller with your own application code. The software can be downloaded via RS232 to the board, which is equipped with a flash microcontroller, preprogrammed with a "flash-monitor".

What you'll need

The following components are necessary :

- ✓ Flash Test Board "FLASH-EVA2-120P-M13"
- ✓ MB90F523 with programmed Flash monitor and simple application (on Micro-CD-Rom V3.1)
- ✓ RS232-Cable (included)
- ✓ Power-supply 7,5V DC, 400mA (not included)
- ✓ Fujitsu's Micros CD-ROM V3.1 or higher

Software Installation

NOTE: The following descriptions refer to the software development environment *SOFTUNE Workbench*.

If you do not have installed **Softune Workbench** for 16-Bit Microcontrollers yet, please install now from Micro CD V3.1 onto your PC.
Furthermore, in order to follow the next chapters explanation, you will need the following software tools, which you can find under C:\Softune\Utility after installing Softune Workbench.
These tools are

SKWIZARD : The latest version of Starterkit Wizard (Terminal/Download tool)
HEXLOADW : Hexloader, can also be used to transfer data to the flash-board

Register these tools under the *Setup/Tool/* menu of *SoftuneWorkbench* with the following options:
SKWIZARD.EXE
Option:1 -sk16 -i19200 –d

HEXLOADW.EXE
Option: 1 -Flash -w -C -i19200 %A.mhx
Executing directory: %D

Now simply copy the provided software project from
the CD-drive:\StarterKit\Flash120\90520\IOPort\ onto your local harddisc and open the project *IOPort.prj* with Softune Workbench.
To check, if your configuration is correct, start "Build" and make sure that also the *load module converter* is started.

Hardware Setup

This section shows how to setup the hardware of your Flash Board :

a) Use a power supply of approx 7.5V DC at 300 mA. Connector : shield : **+** , Center : **-**
b) Be sure the Mode-Pins (Switch S4) are set to 1,3,4=OFF, 2=ON.
c) UART configured to “burn-in”-connector, i.e. JP12, JP13, JP14 (RxD, TxD, DTR) in “burn-in” position.
d) Device-Selector RN4 in MB90F520 position.
e) To set the correct Reset-conditions for downloading code later on, set JP20 to DTR and JP21 to /HST.
f) The preprogrammed device MB90F523 must be placed into the burn-in socket (Chip-marking towards ①). Be sure, the power is switched off while inserting the device !
g) Connect the Flash Board with your PC (Default is COM1) using the RS232 cable.
h) Power up the board (Check the green Power LED)
i) If you wait for 1 second approximately, your preprogrammed application is going to be executed and the LEDs are lighted in different patterns.
j) To reprogram your flash device with another application see next chapter.

Reprogramming of your application

SkWizard and HEXLOADW- download tools

As an example, it is shown how the provided IOPort -project is programmed into the flash memory using two different tools, provided with this Flash Test Board.
Note: A program to be downloaded with the bootloader, needs certain memory mapping to avoid overlapping with the bootloader itself. Therefore use the provided project which you can find on CD-drive:\StarterKit\Flash120\90520\IOPort\IOPort.prj.

a) Start SoftuneWorkbench and open the provided project *IOPort*.
b) Select “Build” to recompile, assemble and link the project. (“No Error.” should appear in the output box).
c) To transfer the converted linker-output file (OUT.MHX) to the Flash Board, you may use the provided tool SKWIZARD from the SoftuneWorkbench-Setup-ToolExecution-Menu. If you can not find SkWizard there, register the tool first, as it was mentioned before.

Select “SkWizard” from the Setup-ToolExecution-menu. After pressing RESET (F2) “MB90523 Flash loader – V3.0” should appear as the first prompt. Respond with “ESC” within 1 second and you should see the monitor prompt “>”. Then click on “Load” and select OUT.MHX to download, which you can find under the \ABS directory of the IOPort-Project.

d) RESET (F2) again will start executing the program.
e) You will see the LEDs (next to the burn-in socket) flashing. This program has now be programmed to the flash memory of the MB90F523.
f) You can modify the program, e.g. change the wait-time-value in Main.c, and repeat the steps b) to d).

Instead of using the SkWizard you can also use the HEXLOADW download-utility to transfer the linker-output file to the Flash Board.
HEXLOADW automatically erases the flash memory sectors respectively and downloads the application program code. Then press the RESET-button of the Flash Test Board and the application will start.

Different methods to reprogram the flash memory

There are two methods to download programs into Flash memory, ***Method A*** and ***Method B***.
In the previous example, the TEST520-project has been programmed into the flash-memory using *Method A.*
Method A: The entire sector will be erased before a byte will be written to it. Method A does not check, wheather the content of the sector is different from the data that should be written to it or not. Hexloadw supports only this method. SkWizard also works with *Method B.*
Method B: In this case the code will be transferred twice to the Flash-Board. During the first transmission, the flash-monitor checks for differences between the actual contents and the received data. During the second transmission, it will only erase those sectors, to which new data should be written to. This method also takes into account, that a flash memory cell only has to be erased, if their data-bit has to be reprogrammed from logical “0” to “1”.
To choose Method B with the SkWizard add an the *–flash* option in the command line :

SKWIZARD.EXE 1 **–flash** -sk16 -i19200 –d

Refer to Appendix for more details on options.

The Flash-Monitor for MB90523 – V3.0

The MB90F523-device you have used so far has a simple monitor-program with bootloader functions preprogrammed in flash. Besides of the communication commands you have used indirectly with one of the download tools, you can also enter some additional commands from the SkWizard or any terminal program.
To do so, press RESET (F2), “MB90523 Flash loader – V3.0” appears as the first prompt. Respond with “ESC” within 1 second and you should see the monitor prompt “>”, as already mentioned before. At this point you can enter the following commands:

Command overview

➢ RM AAAAAA NN	: “Read Memory”: NN bytes from location AAAAAA[*)]
➢ ES NN	: “Erase Sector” : Erases sector NN
➢ G	: “Go” : Call user application
➢ CALL AAAAAA	: “Call” : Call address AAAAAA
➢ PROTECT_RESET_VECTOR OFF	: Disables write-protect function for reset-vector
➢ PROTECT_RESET_VECTOR ON	: Enables write-protect function for reset-vector
➢ PROTECT_SECTORS OFF	: Disables erase-protect function
➢ PROTECT_SECTORS ON	: Enables erase-protect function
➢ CS OFF	: Switch checksum-algorithm off
➢ RST	: Software reset of MCU
➢ HEXDWL	: Switch to Download-Mode (Method A)
➢ HEXDWL1	: Switch to Download-Mode (Method B – first step)
➢ HEXDWL2	: Switch to Download-Mode (Method B – second step)
➢ (Escape)	: any ESC aborts the current function

[*)] AAAAAA: 24-bit upper case hex address
NN: two-digit upper case hex number

Defaults:
PROTECT_SECTORS **ON**
PROTECT_RESET_VECTOR **ON**
to protect the monitor from erasing the reset vector and itself.

Application Call

After Reset of the MCU, the bootloader always takes control, at least for the first steps to check the cause of reset.

After *PowerOn* and *HardwareStandby* (*HST*-pin) the version string of the loader “MB90523 Flash loader – V3.0” is output and you have the chance to respond with *ESC* in order to enter the flash monitor. Otherwise, the application is called after a timeout of approximately 1 second.

If the *RST*-pin or a software reset caused the reset of the MCU, the application is called immediately.

As you can see, you can choose between these two different behaviours by setting the reset-jumper J21 in the appropriate position (default = *HST*).

The following table summarizes this context again:

Reset Event	Call of …
Power On	BootLoader -> TimeOut -> Application or BootLoader -> „ESC“ -> Flash Monitor *)
HST	BootLoader -> TimeOut -> Application or BootLoader -> „ESC“ -> Flash Monitor *)
RST	Application
Software RST	Application

Table 2: Reset sources

*) If „ESC“ is pressed during the timeout period, the program execution branches to the Flash Monitor.

Note that the loader reads the WDTC (WatchDogTimer-Control) register in order to find out the cause for reset. While reading WDTC, the content is destroyed, but through the R0 register, a copy of the content is passed to the application software for further evaluation.

If you download an application, which normally has its own reset vector at FFFFDChex, this “application vector” will be stored in INT#7 location at address FFFFE0hex. Therefore a vector call for INT#7 is no longer available in conjunction with this boot loader. The boot loader itself takes care of this displacement. This way of handling with the reset vector is very flexible because the “application vector” can be defined in the software project inside *Softune* as the normal reset vector and is not fixed to a certain address.

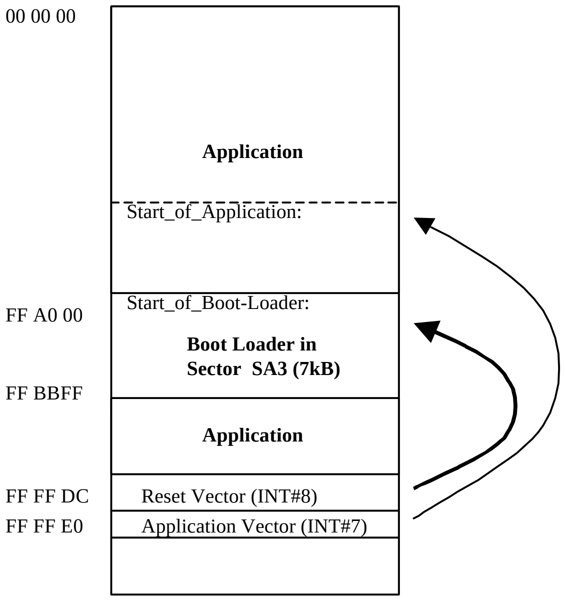


Figure 13: Location of reset vector and application vector

Using your own Flash-Loader

The flash-monitor which is also provided on the Micro CD V3.1 should act as an example and can be adapted to your own needs.
How to overwrite older versions of the bootloader with the MB90523 Flash loader – V3.0 or with your own one, read the following instructions carefully :

Before downloading the new flash-loader to the flash device, the reset-vector protection and sector-protection of the current flash-monitor must be switched off. Otherwise, any attempt to overwrite the actual monitor will result in an error. Use the commands :

PROTECT_RESET_VECTOR **OFF** and
PROTECT_SECTORS **OFF**

from the SkWizard or any terminal program to allow rewriting the monitor-area.

Appendix: Command Line Options

HEXLOADW :

HEXLOADW {PORT} [-R|-C] [-Flash] [-Ibaud] [file]

PORT This is the communication port that the Flashboard is connected to.
-W Wait for target (Power on)
-C Close HexloadW after download
-Ibaud Initialise serial port to baud (300, 1200, 2400, 4800, 7200, 9600, 19200, 38400).
-Flash Flashprogramming using Method B

Examples:

HEXLOADW 1 myprog.cnv
Programs myprog to flash via COM-port 1 using method A

HEXLOADW 2 –flash -r -c myprog.cnv
Programs myprog to flash via COM-port 2 using method B, executes the program and shuts down

STARTERKIT WIZARD :

SKWIZARD [P] [-IBaud] [-SKType] [-D] [-R [-C]] [File]

P COM-Port Number (1,2,3,4)
Baud Baudrate (300,1200,2400,4800,9600,19200,38400)
Type Starterkit-Type (8,16,32)
-D Detect Baudrate (syncronizes Baudrate with board)
-R Run after load (only useful if file is specified)
-C Close program after a successful load and execution (only if -R is specified)
-F Flashprogramming using Method B

File (last parameter given) should contain full path and filename+ext of hex-file

Examples :

SKWIZARD 1 –SK16 –i19200
Simply opens the application for use with the Flash-Board via COM-port 1

SKWIZARD 1 -SK16 -i19200 myprog.cnv
As above, but will program myprog to flash using method A

SKWIZARD 2 -SK16 -i19200 –f –r -c myprog.cnv
Now uses COM 2 and will program myprog to flash using method B, then execute it and shuts down

BINHEX :

BINHEX /o=-FF0000 /z=FFA000 /m test.cnv /o=FF0000 /a

/o is the read offset (in this case negative)
/z is new offset for records below 1000 hex
/m is for Motorola format
test.cnv is the input file
/o is the write offset (to put the record back where it started)
/a is to change addresses only