

MC-32 Multiboard Programming Adapter

The multiboard adapter is currently available for the **Galep5** programming software and supports the Fujitsu **MB95Fxxx** series. A **GalepX** version and support for further MCU types can be implemented on request.

1. General

The multiboard programming adapter connects up to 32 MCU boards to a single GALEP-5/5D programmer for programming up to 32 devices simultaneously using a serial 1-wire interface. It can switch between multichannel and single channel mode. In single channel mode, any of the 32 boards can be selected for blank check, read, and programming. In multichannel mode all boards are programmed in parallel.

Serial numbers and further data can be programmed separately for any device, even in multichannel mode. Device specific trim values as well as programming results (pass / fail) can be stored through an external DLL. Programming and pass/fail status can be also controlled through external signals.

	Multichannel	Single channel	Remarks
Number of channels	1-32	1-32	
Number of devices	1-32	1	
Program	Yes	Yes	
Erase	Yes	Yes	
Blank check	Yes	Yes	
Verify	No	Yes	
Verify (fast)	Yes	Yes	
Read	No	Yes	
Serial numbers	Yes	Yes	With external DLL
Pass / Fail	Yes	Yes	With external DLL

2. Galep 5 / DIL48 Pin connection

DATAIO_CH17	--	25	24	--	DATAIO_CH16
DATAIO_CH18	--	26	23	--	DATAIO_CH15
DATAIO_CH19	--	27	22	--	DATAIO_CH14
DATAIO_CH20	--	28	21	--	DATAIO_CH13
DATAIO_CH21	--	29	20	--	DATAIO_CH12
DATAIO_CH22	--	30	19	--	DATAIO_CH11
DATAIO_CH23	--	31	18	--	DATAIO_CH10
DATAIO_CH24	--	32	17	--	DATAIO_CH9
DATAIO_CH25	--	33	16	--	DATAIO_CH8
DATAIO_CH26	--	34	15	--	DATAIO_CH7
DATAIO_CH27	--	35	14	--	DATAIO_CH6
DATAIO_CH28	--	36	13	--	DATAIO_CH5
DATAIO_CH29	--	37	12	--	DATAIO_CH4
DATAIO_CH30	--	38	11	--	DATAIO_CH3
DATAIO_CH31	--	39	10	--	DATAIO_CH2
DATAIO_CH32	--	40	9	--	DATAIO_CH1
n.c.	--	41	8	--	RSTX
CRTL0	--	42	7	--	VCC (n.c.)
CRTL1	--	43	6	--	VPP (n.c.)
CRTL2	--	44	5	--	VCC_ON
CRTL3	--	45	4	--	VPP_ON
CRTL4	--	46	3	--	SERST_DATA
+5V	--	47	2	--	SERST_CLK
GND	--	48	1	--	SERST_CLK_ST

2.1 Data In/Out channels:

Galep5/DIL48 Pin No.	Name	Desc
9 - 40	CHN1 - CHN32	32 Data Rx/Tx Channels

2.2 Reset control:

Galep5/DIL48 Pin No.	Name	Desc
8	RSTX	Unused

2.3 Vcc control:

The **VCC_ON** pin can be used as power supply by Galep for the target systems (max. 200 mA total), or as a control pin for external power supply switching.

Galep5/DIL48 Pin No.	Name	Desc
4	VPP_ON	VPP Switch control (0-5V)

5	VCC_ON	VCC Switch control (0-5V)
6	n.c.	
7	n.c.	

2.4 Action result SYNC out:

Galep5/DIL48 Pin No.	Name	Desc
1	SERSTATE_CLK_ST	Data ready clock (CPOL=0)
2	SERSTATE_CLK	Clock CPOL=0
3	SERSTATE_DATA	Data out

2.5 ICT control:

This option supports simple ICT (In Circuit Tester program control) functions and enables a programming loop controlled by adapter pins. The result and status is reported via pins too. Only when the **'Program'** action is activated, the ICT related pins are active.

When the programming loop is ready for next start or abort, the text *'Control is waiting'* is displayed.

If a signal (pull down) is applied to the the Start pin, *'Action is running'* is displayed.

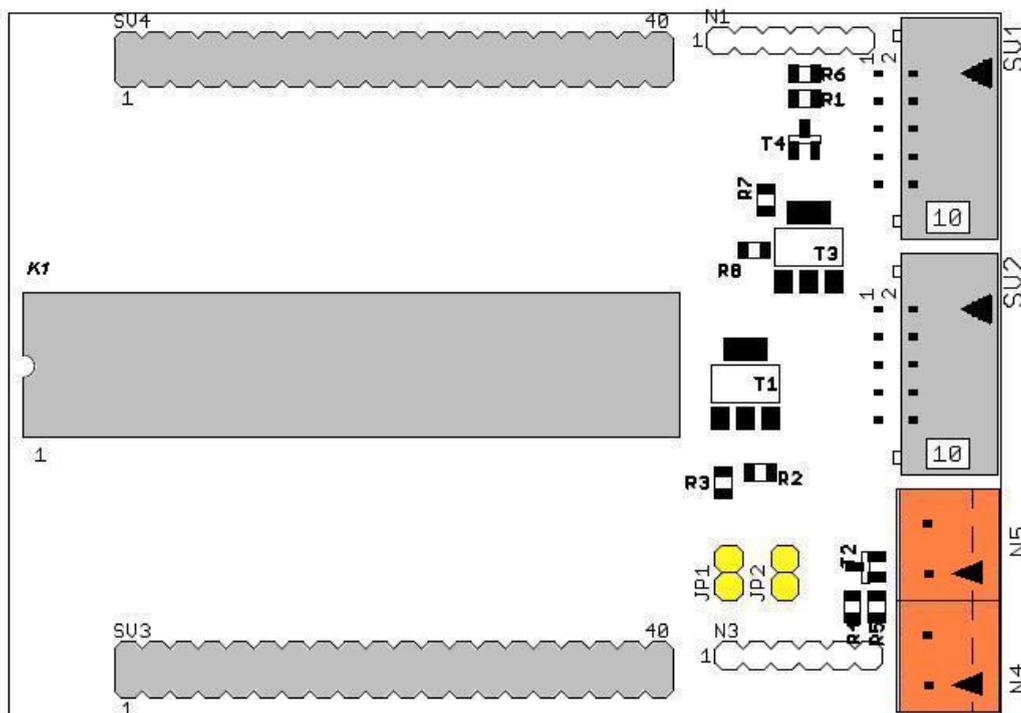
If a signal (pull down) is applied to the Abort pin, *'Control break'* is displayed.

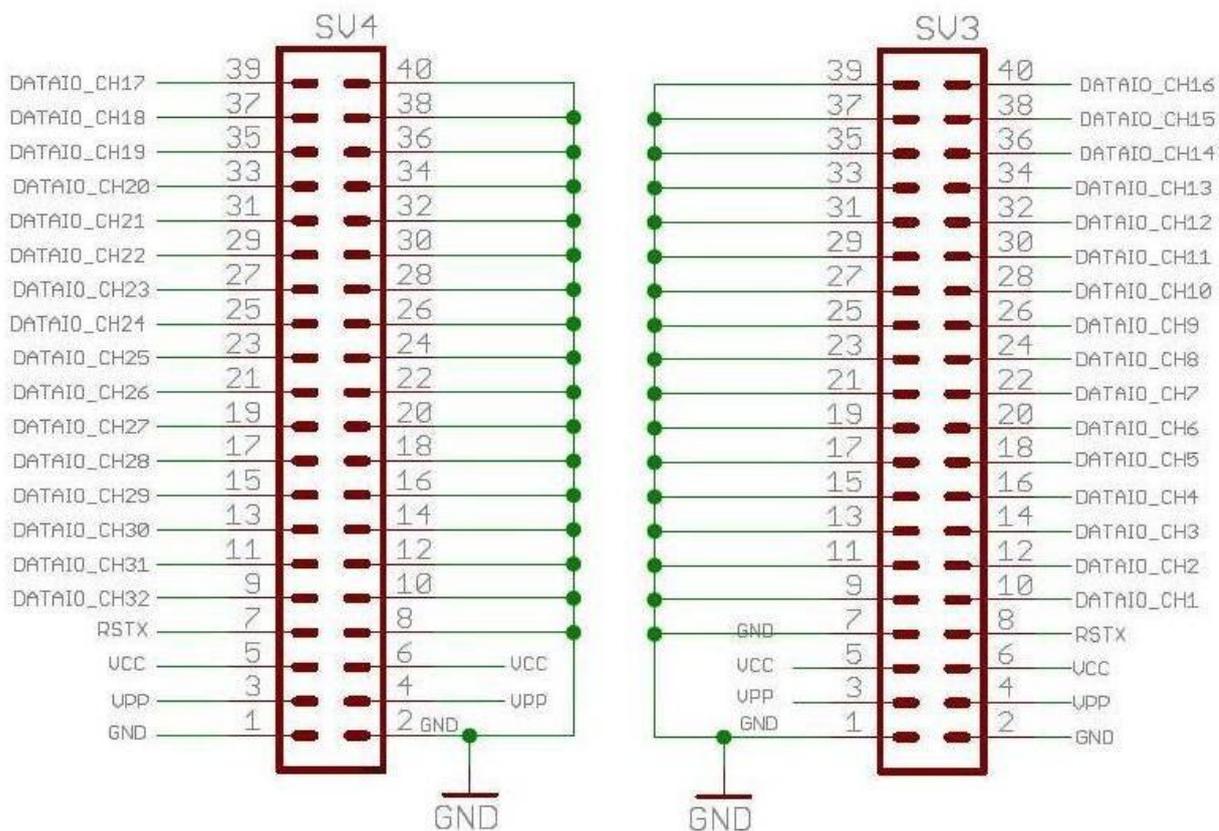
Galep5/DIL48 Pin No.	Name	Description
42	CRTL0	Input, Start pin. User can start the next programming cycle when dropped to LOW. Connected with a pullup resistor to +5V.
43	CRTL1	Output/5V. Will be set to HIGH if the Galep5 is prepared and ready for next Start signal. Will be set LOW when Start is pulled to LOW.
44	CRTL2	Output/5V. Will be set HIGH if any error occurs. Otherwise it will be held LOW. Will be set LOW when Start is pulled to LOW.
45	CRTL3	Output/5V. Will be set HIGH if the complete programming action has finished. Will be set LOW when Start is pulled to LOW.
46	CRTL4	Input, Abort pin. User can abort the programming loop when dropped to LOW. Connected with a pullup resistor to +5V.

2.6 Adapter ground:

Galep5/DIL48 Pin No.	Name	Desc
47	ADRVCC	Adapter VCC (5v)
48	GND	Ground

3. Adapter main board





3.1 Power supply (Vcc)

The power is controlled by VCC_ON (Pin5, K1). It can be connected to the target boards through pins 6..7 (Vcc) and pins 1..2 (ground) of the SV3 and SV4 connectors.

3.1.1 Power supply from GALEP

Jumper JP1 - closed.

No external power supply must be connected to connector N4. The total power consumption must not exceed 200 mA.

3.1.2 Power supply from external source

Jumper **JP1** - open.

Connect an external power supply to connector **N4**, pin1 = Vcc, pin2 = ground.

3.2 ICT Connector

The **SV1** connector is used for controlling the programming process through external signals (see above):

(Switch:Start PGM)	CRTL0	--	1	2	--	GND
(Out: Ready=HI)	CRTL1	--	3	4	--	GND
(Out: Error=HI)	CRTL2	--	5	6	--	GND
(Out: PGM END=HI)	CRTL3	--	7	8	--	GND
(Switch:Break Loop)	CRTL4	--	9	10	--	GND

SV1

3.3 Serial Status Output

An external shift register can be used for storing pass / fail results. The shift register is controlled with the signals SERST_CLK_ST, SERST_CLK, SERST_DATA at the pins 1,2 und 3 at the **K1** and **SV2** connectors.

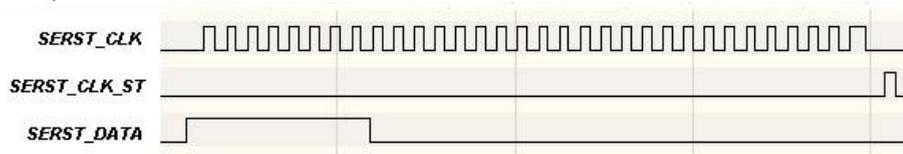
SERST_CLK	--	1	2	--	GND
SERST_DATA	--	3	4	--	GND
VCC	--	5	6	--	GND
VPP	--	7	8	--	+5V
unused	--	9	10	--	SERST_CLK_ST

SV2

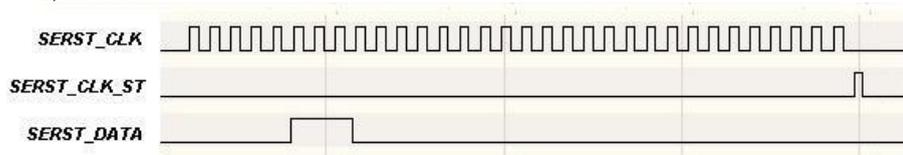
The pass / fail state is sent in 3-block sequences: [used channels], [pass state], [fail state]. Bit 0 represents channel 1, Bit 2 channel 2 etc. The block width in bits can be set up from 1 to 32 in the **.cfg** file.

Example: 32 bit blocks, 8 used channels, boards 1..5 passed and boards 6..8 failed:

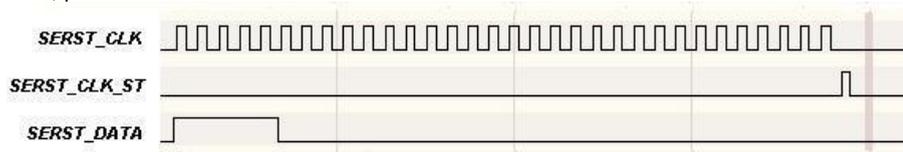
Block 1, used channels:



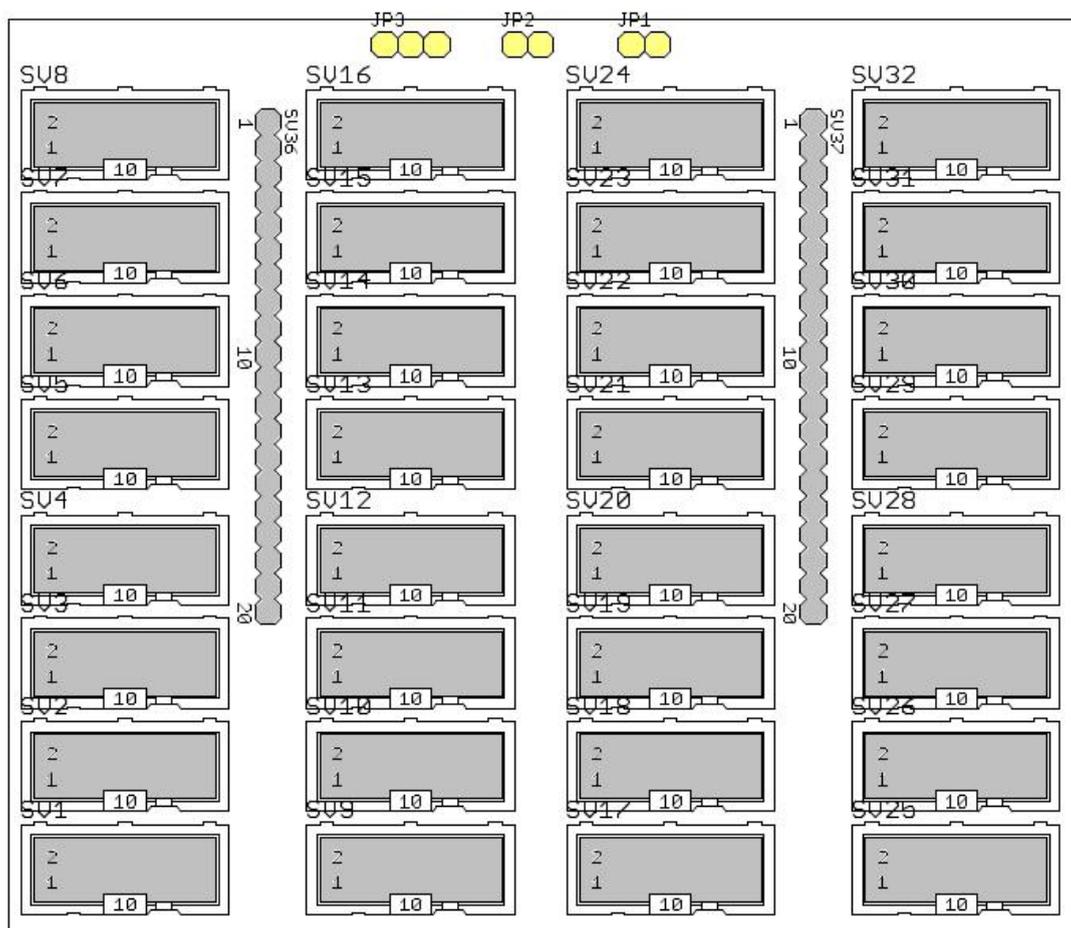
Block 2, fail state:



Block 3, pass state:

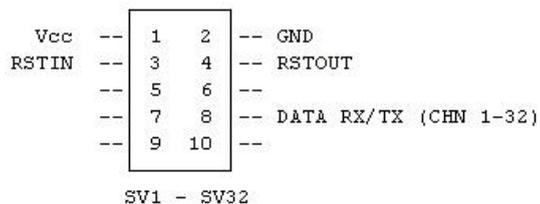


4. Adapter top board



4.1 Connectors for MCU Boards

For connecting the MCU boards, 32 10-pin connectors SV1 - SV32 are used.



4.2 Configuration

JP1 closed: power supply (Vcc) through main board.

JP2 closed: RSTIN connected to Vpp at the main board.

JP3 1-2 closed: RSTIN connected to RSTX at the main board.

JP3 2-3 closed: RSTOUT connected to RSTX at the main board.

4.3 Configuration for Programming MB95F3xx and MB95F5xx Devices

Main board:

Power supply through GALEP: **JP1** closed

External power supply: **JP1** open, external power source connected to **N4**, pin1 +5V, pin2 ground.

Top board:

JP1 closed, **J2** open, **J3** open.

5. Algo Options in the .CFG File

Keyword	Value	Desc
USCHS	0-0xFFFFFFFF	32 bit mask, one channel per bit. bit 0 = channel 1, bit 1 = channel 2 etc.
PGMFW	0-1	Enable (1) or disable (0) serial number programming.
SNDDTA	0-1	Enable (1) or disable (0) sending pass/fail data and adjustment data per channel after every programming process.

TRVAD	0-0xFFFF	Address of trim value in the NVR.
TRVSZ	1-2	Number of bytes (1 or 2) for the trim value.
GFI	[name.DLL]	Name of an external DLL containing functions for serial number programming and pass/fail data storage.
PGMNUM	0-0xFFFFFFFF	Initial value for the pgmNum variable that is incremented after every programming process.
SRSTWD	1-32	Block width in bits for the shift register to receive pass / fail signals (see above).
SNGLCHN	0-32	Channel number (1..32) for single channel mode, 0 for multichannel mode.

The other data fields in the .cfg file must not be modified.

The USCHS, PGMFW, SNDDTA, SNGLCHN variables can also be set up in the Galep5 [Action Options] menu.

6. External DLL

The DLL is used for programming serial numbers and/or storing trim values and pass / fail results. An example DLL (mb95fxxx.dll) is included and can be used as a template. The DLL name must be defined in the .cfg file (**GFI = name.dll**). The DLL must be copied in the "**galep5bin\ext**" subfolder.

The following header file should be used for the DLL. It declares 3 export functions and a struct for the data to be stored:

```
namespace exMB95Fxxx
{
    #pragma pack(1)
    struct sMB95FxxxActionDataStc
    {
        unsigned long pgmNum;    // incremented after every programming process.
        unsigned char trimVal1;  // trim value CRTH
        unsigned char trimVal2;  // trim value CRTL
        unsigned short errStAdr; // first error address (in case of an error)
        unsigned short errEndAdr; // last error address
        unsigned char errCode;    // error code
        unsigned char bUsedChn;  // channel number (1-32)
    };
    #pragma pack()
};

#ifdef __cplusplus
extern "C" {
#endif
    MB95FXXX_API unsigned long InitializeData ( unsigned long id);
    MB95FXXX_API const unsigned char * GetData ();
    MB95FXXX_API void SetData ( unsigned long algoSign, const exMB95Fxxx::sMB95FxxxActionDataStc * data, unsigned long n);
#ifdef __cplusplus
}
#endif
#endif
```

Function: MB95FXXX_API unsigned long _cdecl InitializeData (unsigned long in_idval)

Parameter: PGMNUM

returns: data size

'**InitializeData**' is called by the Galep5 software when the serial number or other firmware data is required for programming a device. The parameter is the current program counter (PGMNUM). The function has to initialize the data that are retrieved with the subsequent call of '**GetData**'. The return value is the size of the data.

The data has to be in the following format:

Header:

Offset	Size	Typ	Funktion
0x0000	32 bits	uint32 ID	Value passed to the function
0x0004	16 bits	uint16 NUM	Number of data records
0x0006	data record size * NUM	data record	Data records

Data record:

Offset	Size	Typ	Funktion
0x0000	16 bit	uint16 ADR	Address in NVR
0x0002	16 bit	uint16 NUM	Date size in bytes
0x0004	8 Bit * NUM	Data	Data

Funktion: MB95FXXX_API MB95FXXX_API const unsigned char * _cdecl GetData ()

Parameters: -

Returns: Pointer to the data

'**GetData**' is called by the Galep5 software immediately after '**InitializeData**' and returns a pointer to the data in the above format.

Funktion: MB95FXXX_API void _cdecl SetData (unsigned long algoSign, const exMB95Fxxx::sMB95FxxxActionDataStc * data, unsigned long n)

Parameter1: unsigned long algoSign - algo identifier, f.i. 0x95F564 for MB95F564.

Parameter2: const exMB95Fxxx::sMB95FxxxActionDataStc * data - list of data structs of type **sMB95FxxxActionDataStc**.

Parameter3: unsigned long n - data list size

Returns: -

The function is called by the Galep5 software after every programming process. It gets a list of **sMB95FxxxActionDataStc** structs for further processing.