

# Dokumentation METH\_V2

**Dok-Rev. 1.1 vom 05.07.2016**

**Hardware-Rev. 1.0 vom 19.02.2016**

---

---

## Inhaltsverzeichnis

<b>1</b>	<b>Allgemeine Hinweise.....</b>	<b>4</b>
1.1	Handhabung	4
1.2	Installation	4
1.3	Erklärung	4
1.4	Reparaturen	4
<b>2</b>	<b>Allgemeine Informationen .....</b>	<b>5</b>
2.1	Einbau	5
2.2	Umgebungsbedingungen	5
2.3	Mechanische Abmessungen	5
2.4	Technische Eigenschaften	5
<b>3</b>	<b>Inbetriebnahme.....</b>	<b>6</b>
3.1	Belegung der VG-Leiste:	6
<b>4</b>	<b>Hardwarebeschreibung.....</b>	<b>7</b>
4.1	Serielle Schnittstellen	7
4.2	Frontansicht	7
4.2.1	Reset-Taster	7
4.2.2	Leuchtdioden VCC	7
4.2.3	Adresseinstellung High/Low	7
4.2.4	Ethernetanschluß	8
4.3	Ausgänge	8
4.4	Eingänge	8
4.5	Batterie/Goldcap	8
4.6	I/O Erweiterung (optional)	8
4.6.1	Belegung optionale Eingänge ST1	9
4.6.2	Belegung optionale Ausgänge ST2	9
<b>5</b>	<b>Speicherbelegung .....</b>	<b>10</b>
5.1	Speicheraufteilung	10
5.1.1	DRAM Belegung mit TOP-RTOS (aktuelle System Belegung)	10
5.1.2	DRAM Belegung mit BOTTOM-RTOS (alte System Belegung)	10
5.1.3	ChipRam Belegung	10
5.1.4	NandFlash Belegung	11
5.1.5	Vector-Tabelle beim MPC5125	11
5.1.6	Lokale I/O Geräte auf /I2C1/	12
5.2	RTOS-UH Sumpfzellen	12

---

<b>6</b>	<b>I/O - Geräte .....</b>	<b>13</b>
6.1	I <sup>2</sup> C	13
6.1.1	Belegung des EEPROM1	13
6.1.2	Belegung des EEPROM2 (optional)	14
6.1.3	Belegung des PWREG	14
6.2	NandFlashDisk /FD/	14
6.3	Autostart (normal)	14
6.4	Autostart (Recover)	15
6.5	NotStart / Recover RTOS	15
<b>7</b>	<b>RTOS-UH im FLASH ablegen .....</b>	<b>16</b>
<b>8</b>	<b>MPC5125 I/O Pin Belegung.....</b>	<b>17</b>
8.1	BaseBoard Setup	21
8.2	Geräte-Treiber	21
8.3	Serielle Schnittstellen Atmos-Maurer	21
8.4	Gerät DIO – Bits	22
8.5	Gerät HEX – Bits (Input Only)	23
8.6	Gerät DIP – Bits (Input Only)	23
8.7	Gerät LED – Bits (Output Only)	24
8.8	Gerät AUX – Bits (Input Only)	24
8.9	Gerät CAP – Bits (Input Only)	25
8.10	Gerät FRAM	25
8.11	Gerät SPI1 (8 Bit I/O – Extension)	26

Revisionsliste:

Rev.	Datum	Na.	Änderung
1.0	26.05.2016	Kr	Erstellung
1.1	05.07.2016	Ko	Überarbeitung und Ergänzung

---

## **1 Allgemeine Hinweise**

### **1.1 Handhabung**

1. Lesen Sie bitte zuerst sorgfältig diese Dokumentation bevor Sie die Hardware auspacken und einschalten. Sie sparen Zeit und vermeiden Probleme.
2. Beachten Sie bitte die Vorsichtsmaßnahmen bei der Handhabung elektrostatisch gefährdeter Hardware.
3. Wenn die Hardware Batterien enthält, legen Sie sie nicht auf elektrisch leitfähige Unterlagen. Die Batterie könnte kurzgeschlossen werden und Schäden verursachen.
4. Achten Sie bitte darauf, daß der spezifizierete Temperaturbereich nicht verlassen wird.

### **1.2 Installation**

1. Überprüfen Sie, ob alle Jumper entsprechend Ihrer Anwendung gesetzt sind.
2. Schalten Sie die Spannungsversorgung der externen Anschlüsse ab, bevor Sie eine Verbindung herstellen.
3. Wenn Sie sicher sind, daß alle Verbindungen korrekt installiert sind, schalten Sie die Spannungsversorgung ein.

### **1.3 Erklärung**

Wir behalten uns das Recht vor, Änderungen, die einer Verbesserung der Schaltung oder des Produktes dienen, ohne besondere Hinweise vorzunehmen. Trotz sorgfältiger Kontrolle kann für die Richtigkeit der hier gegebenen Daten, Schaltpläne, Programme und Beschreibungen keine Haftung übernommen werden. Die Eignung des Produktes für einen bestimmten Einsatzzweck wird nicht zugesichert.

### **1.4 Reparaturen**

Sollte das Produkt defekt sein, so senden Sie es bitte frei in geeigneter Verpackung mit folgender Beschreibung an uns zurück:

- Fehlerbeschreibung
- Trat der Fehler nur unter bestimmten Bedingungen auf?
- Was war angeschlossen?
- Wie sahen die angeschlossenen Signale aus?
- Garantiereparatur oder nicht?

---

## **2 Allgemeine Informationen**

### **2.1 Einbau**

Die METH\_V2 ist zum Einbau in EMV-dichte Gehäuse bestimmt. Die Verkabelung ist EMV-gerecht mit abgeschirmten Kabeln durchzuführen.

### **2.2 Umgebungsbedingungen**

Umgebungstemperatur (Betrieb)	0-50° C
Umgebungstemperatur (Lagerung)	-20-85° C
rel. Luftfeuchte	max. 95%, nicht kondensierend
Höhe	-300m bis +3000m

### **2.3 Mechanische Abmessungen**

Europakarte	160 x 100 mm
-------------	--------------

### **2.4 Technische Eigenschaften**

Versorgungsspannung:	8-38 Volt DC, typ. 160mA bei 24 Volt, max. 1A
Serielle Schnittstellen:	1 x 5-Draht RS-232 2 x 20mA TTY passiv, galvanisch entkoppelt 1 x RS485
Ethernet	10/100 BaseT auf RJ45 Buchse
digitale Eingänge:	4 x 24 Volt / 2 mA galvanisch entkoppelt, interruptfähig
digitale Ausgänge:	2 x 24 Volt / 0,5 A galvanisch entkoppelt
CAN-Bus:	2 x CAN-Bus mit ISO-Interface, galvanisch entkoppelt
DIP-Schalter	5 polig 2 Stück Hex-Dreh-Schalter
RAM	128 MB
NAND-FLASH	128 MB

### 3 Inbetriebnahme

#### 3.1 Belegung der VG-Leiste:

Pin	A-Reihe	Gruppe	C-Reihe	Gruppe
1	PE	Stromversorgung	PE	TTY Kanal 2
2	PE		+20mA IN2	
3	+24 Volt		-20mA IN2	
4	+24 Volt		+20mA OUT2	
5	GND		-20mA OUT2	
6	GND		GND	
7	DC_Vcc CAN1	CAN Kanal 1	DC_Vcc CAN2	CAN Kanal 2
8	Abschluß CAN1		Abschluß CAN2	
9	CAN1 High		CAN2 High	
10	CAN1 High		CAN2 High	
11	CAN1 Low		CAN2 Low	
12	CAN1 Low		CAN2 Low	
13	DC_GND CAN1		DC_GND CAN2	
14	PE		PE	
15	PE	RS 232 Kanal 1	PE	TTY Kanal 3
16	RS1_TxD		+20mA IN3	
17	RS1_RxD		-20mA IN3	
18	RS1_CTS		+20mA OUT3	
19	GND		-20mA OUT3	
20	RS1_RTS		GND	
21	IN 1 +	Eingang 24 Volt	IN 2 +	Eingang 24 Volt
22	IN 1 -		IN 2 -	
23	+24 Volt O1	Ausgang 24V/0,5A	+24 Volt O2	Ausgang 24V/0,5A
24	LED O1 (2mA)		LED O2 (2mA)	
25	Load O1		Load O2	
26	GND O1		GND O2	
27	RS485_R1	Rs485 120Ohm	IN 3 +	Eingang 24 Volt
28	RS485_R2		IN 3 -	
29	Rs485 +	Rs485 Signal	IN 4 +	Eingang 24 Volt
30	Rs 485 -		IN 4 -	
31	Rs485 Gnd	Rs485 Gnd	TTL-Eingang 1	I/O-Prozessor
32	TTL-Eingang 3	TTL-Eingang 3	TTL-Eingang 2	

---

## 4 Hardwarebeschreibung

### 4.1 Serielle Schnittstellen

Es stehen max. 3 serielle Schnittstellen zur Verfügung. Die mit **RS1** bezeichnete Schnittstelle ist die Systemschnittstelle des RTOS-UH. Sie hat einen 5-Draht Anschluß und ist direkt an den Core5125 angeschlossen.

Anschluß	Schnittstelle	RTOS
TxD	RS1	A1
RxD		
RTS		
CTS		

Die Schnittstellen 2 und 3 sind passive 20mA TTY-Schnittstellen:

Anschluß	Schnittstelle	RTOS
OUT2	TTY-Kanal 1	A2
IN2		
OUT3	TTY-Kanal 2	A3
IN3		

### 4.2 Frontansicht



#### 4.2.1 Reset-Taster

Das Drücken des Reset-Tasters führt zum sofortigen Ende aller Aktivitäten der METH\_V2 und es erfolgt ein Kaltstart wie nach dem Einschalten der Versorgungsspannung.

#### 4.2.2 Leuchtdioden VCC

Die Vcc-LED LE1 zeigt das Vorhandensein der Betriebsspannung an.

#### 4.2.3 Adresseinstellung High/Low

Die beiden Hex-Dreh-Schalter dienen der Adresseinstellung und können vom Anwenderprogramm ausgelesen werden.

---

---

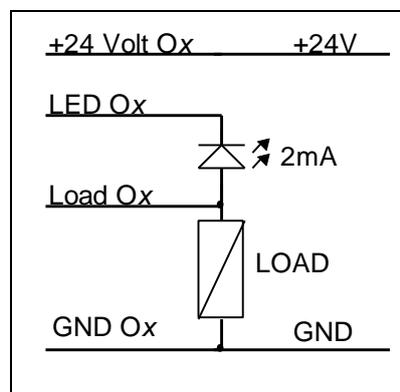
#### 4.2.4 Ethernetanschluß

Der Ethernetanschluß ist über zwei RJ45-Buchsen für Twisted-Pair-Verkabelung realisiert. In der Buchse sind 2 Leuchtdioden integriert, die anzeigen das Verbindung besteht (Link) und ob Aktivitäten (Senden/Empfangen – Activity) stattfinden.

#### 4.3 Ausgänge

Es stehen 2 galvanisch getrennte Leistungsausgänge zur Verfügung.

Die beiden Leistungsausgänge sind als galvanisch getrennte High-Side-Schalter ausgelegt:



Zusätzlich kann eine LED (2mA Strom) zur Kontrolle des Schaltzustandes angeschlossen werden.

#### 4.4 Eingänge

Die Eingänge sind galvanisch getrennt und gegen Verpolung geschützt. Der Eingangsstrom bei 24 Volt Spannung beträgt ca. 2 mA. Vor den Eingängen IN1/IN2 liegt ein Tiefpaß mit ca. 165 Hz Grenzfrequenz, vor IN3/IN4 mit ca. 16,5 KHz.

#### 4.5 Batterie/Goldcap

Das Gerät ist mit RTC ausgestattet. Die Pufferung erfolgt entweder über eine 3,0 Volt Lithiumbatterie oder einen 1 F Goldcap. Die Batterie wird erst bei der Lieferung eingesetzt und hat dann eine Mindestlebensdauer von 5 Jahren, unabhängig von der Einschaltdauer des Gerätes. Die Pufferdauer mit dem Goldcap hängt vom Ladezustand ab, bei geladenem Goldcap ist eine Pufferdauer von min. 14 Tagen gegeben.

#### 4.6 I/O Erweiterung (optional)

Über eine Erweiterungsplatine können weitere je 8 Ein- und Ausgänge zur Verfügung gestellt werden. Die Ein- und Ausgänge sind jeweils für sich galvanisch getrennt, d.h. die Eingänge haben eine gemeinsame Masse, die Ausgänge eine gemeinsame 24V Versorgung und Masse. Die Anschlüsse werden über 2 Stück 10 polige Pfostensteckverbinder zur Verfügung gestellt.

---

Die DC24V für die Ausgänge müssen von extern zugeführt werden!

#### 4.6.1 Belegung optionale Eingänge ST1

Belegung	PIN	PIN	Belegung
IN1	1	2	IN2
IN3	3	4	IN4
IN5	5	6	IN6
IN7	7	8	IN8
DC24V	9	10	DC0V

#### 4.6.2 Belegung optionale Ausgänge ST2

Belegung	PIN	PIN	Belegung
O1	1	2	O2
O3	3	4	O4
O5	5	6	O6
O7	7	8	O8
DC24V	9	10	DC0V

---

## 5 Speicherbelegung

### 5.1 Speicheraufteilung

Speichertyp	Startadresse	Endadresse	Kommentar
DRAM	\$00000000	\$07FFFFFFF	128 MB DRAM
Bus	\$80000000	\$8FFFFFFF	256 MB Muxed AD[32] Bus opt. CS0
Prozessor	\$F0000000	\$F00FFFFF	1 MB Register des MPC5125
ChipRam	\$F0200000	\$F0207FFF	32 KB chiplokales RAM
Ext.I/O	\$F7000000	\$F701FFFF	128 KB ext. I/O opt. CS0
NandFlash	\$FFF00000	\$FFFFFFF	NandFlash Controller Space

#### 5.1.1 DRAM Belegung mit TOP-RTOS (aktuelle System Belegung)

PPC-Vector	\$00000000	\$00003FFF	16 KB
Anwenderbereich	\$00004000	\$075FFFFF	118 MB - 16 KB
Oberhalb der RTOS-Speicherverwaltung Feste Vergabe der Adressen			
DUT	\$07600000	\$077FFFFF	2 MB Display
MMU-PTAB	\$07800000	\$078FFFFF	1 MB MMU PageTab
..	\$07900000	\$079FEFFF	1 MB – 4KB . frei
PREBOOT	\$079FF000	\$079FFFFF	4KB prebooter
RTOS-UH	\$07A00000	\$07FFFFFF	Max 6 MB

#### 5.1.2 DRAM Belegung mit BOTTOM-RTOS (alte System Belegung)

PPC-Vector	\$00000000	\$00003FFF	
RTOS-Variablen	\$00004000	\$0001C7FF	
PREBOOT	\$0001C800	\$0001FFFF	Prebooter
RTOS-UH	\$00020000	RT_SCAN_LEN	Je nach Länge des Scanbereiches
Anwenderbereich	RT_SCAN_LEN+1	\$07DFFFFF	
DUT	\$07E00000	\$07FFFFFF	2 MB Display

#### 5.1.3 ChipRam Belegung

#FEC	\$F0200000	\$F020008F	FEC I/O Bufferpointer
FEC-Buffer	\$F0200090	\$F0200CFF	2 FEC Tx-Buffer
USB-Verw.	\$F0201000	\$F02022FF	USB Verwa. Pointer
	\$F0202300	\$F0207FFF	#Frei#

---

### 5.1.4 NandFlash Belegung

Die gesamte Größe des NandFlash beträgt 128 MB (64K Blöcke je 2 KB)

Start Block	End Block	Size	Bereich
0	255	512 KB	..frei
256	511	512 KB	1.Booter Bereich
512	767	512 KB	Safe Booter Bereich
768	1023	512 KB	..frei..
1024	4095	6 MB	1. RTOS Bereich
4096	8191	8 MB	Safe RTOS-Bereich
8192	65535	112 MB	Nand-Flash Disk

### 5.1.5 Vector-Tabelle beim MPC5125

Die Vectortabelle bei einem PowerPC liegt unter RTOS-UH ab der Adresse \$4000. Hier die Zuordnung der einzelnen Vektoren für die interne Peripherie, sie beginnt ab \$4120. Die vollständige Liste ist dem Referenzmanual für den MPC 5125 zu entnehmen.

Adresse	Hardwarefunktion
\$00004124	GPT10 .. GPT2[2] Kanal
\$00004128	GPT11 ..GPT2[3] Kanal
\$0000415C	GPT0 .. GPT1[0] Kanal
\$00004160	GPT1 .. GPT1[1] Kanal
\$000041B8	GPT8 .. GPT2[0] Kanal
\$000041BC	GPT9 .. GPT2[1] Kanal
\$00004240	GPT2 .. GPT1[2] Kanal
\$00004244	GPT3 .. GPT1[3] Kanal
\$00004248	GPT4 .. GPT1[4] Kanal
\$0000424C	GPT5 .. GPT1[5] Kanal
\$00004250	GPT6 .. GPT1[6] Kanal
\$00004254	GPT7 .. GPT1[7] Kanal
\$00004258	GPIO 0..31 Kanal
\$00004278	GPIO 32..64 Kanal
\$00004290	GPT12 .. GPT2[4] Kanal
\$00004294	GPT13 .. GPT2[5] Kanal
\$00004298	GPT14 .. GPT2[6] Kanal
\$0000429C	GPT15 .. GPT2[7] Kanal

---

### 5.1.6 Lokale I/O Geräte auf /I2C1/

Funktion	Bus	Adress	Kommentar
RTC	I2C1	\$D0 (208)	RealTimeClock
EEPROM1	I2C1	\$A0 (160)	EEPROM 2KB
EEPROM2	I2C1	\$A8 (168)	EEPROM 2KB ( optional )
PWRREG	I2C1	\$86 (134)	PowerRegister

### 5.2 RTOS-UH Sumpfzellen

Folgende Sumpfzellen sind mit speziellen Informationen belegt:

Adresse	Zugriff	Beschreibung
\$00005300	Langwort	Taktfrequenz in KHz (\$60AE0=396.000)
\$00005304	Langwort	Prozessortyp PPC5125 (\$8000 5125)
\$00005308	Langwort	Offset MMU-PTAB zu RAMEND von feste Adresse (TOP)
\$000053EA	Byte	xxxxxxxB der Wert wird jede Sekunde aktualisiert (Watch-dog)
		B SRAM Batterie OK
\$000053EB	Byte	xxxRx0xx Zustand der RTC beim Reset
		R RTC Batterie OK
		0 Oszillator Down (Beim Reset normal)

---

## 6 I/O - Geräte

Name	LDN	DRV	Beschreibung
/ETH	17	0	Ethernet-Treiber für 10/100 Mbit
/CAN	123	0	Kanal 1
		1	Kanal 2
A1	0	0/2/6	1. serielle Schnittstelle RS232
A2	2	0/2/6	2. serielle Schnittstelle RS232 opt.
A3	21	0/2/6	3. serielle Schnittstelle RS232 opt.
/I2C	90	0	Zugriff auf das EEPROM
/SPI	91	0..7	SPI-Interface ( optional )
/H0/	3	0	uSD-Card
/FD/	92	0	NandFlashDisk ( optional )

### 6.1 I<sup>2</sup>C

Das 2048 Byte große EEPROM ist über die /I2C/-Dation zu erreichen. Die Adresse, von der gelesen oder geschrieben werden soll, wird im Dateinamen übergeben. Wird keine Adresse übergeben, ist die Startadresse 0. Der Dateiname muß mit einem A beginnen, danach folgt die Adresse 3 stellig:

```
/I2C/C0D160A48    IC2Bus1, Gerät 160 ($A0) lesen/schreiben ab Adresse 48
```

Beispiel:

```
handle := OPEN( '/I2C/C0D160A48' ) ; (* ab Adresse 48 *)
_READ( handle, 20, ADR(buffer) ) ; (* 20 Bytes lesen *)
_CLOSE( handle ) ;
```

Das EEPROM benötigt bis zu 4 ms zum Schreiben eines Bytes, das Lesen kann bis zu 100 µs dauern.

Weiterhin stehen zum Zugriff auf das EEPROM die folgenden PEARL-Routinen zur Verfügung:

```
I2C_RD_EEPROM( unsigned short adr, unsigned short len, unsigned char *data)
I2C_WR_EEPROM( unsigned short adr, unsigned short len, unsigned char *data)
```

#### 6.1.1 Belegung des EEPROM1

Ein 2048 Byte großes EEPROM ist über die /I2C/-Dation erreichbar.

```
/I2C/C0D160A48    IC2Bus1, Gerät 160 ($A0) lesen/schreiben ab Adresse 48
```

Im EEPROM sind die ersten 48 Byte für das System reserviert und dürfen vom Anwender nicht verändert werden!

---

Adresse	Belegung
\$0000-\$0005	Physikalische Ethernetadresse (MAC)
\$0006-\$002F	reserviert für System (42 Byte)
\$0030-\$07FF	frei für Anwenderdaten (2000 Byte)

### 6.1.2 Belegung des EEPROM2 (optional)

/I2C/C0D168A0 I2CBus1, Gerät 168 (\$A8) lesen/schreiben ab Adresse 0

Das optionale EEPROM kann komplett mit Anwenderdaten belegt werden.

Adresse	Belegung
\$0000-\$07FF	frei für Anwenderdaten (2048 Byte)

### 6.1.3 Belegung des PWREG

Es werden 5 I/O-Signale zur freien Nutzung zur Verfügung gestellt. Die weiteren 3 Pins werden auf dem CORE5125 verwendet und dürfen nicht verändert werden. Alle PINs des PWREG sind auf Output geschaltet. Das Output-Register ist auf Offset \$05 erreichbar.

/I2C/C0D134B5 I2C\_BUS 1, Gerät 134 (\$86), Offset \$05

BitMask	Ausgang	Bedeutung
\$80	I2C_IO7	#frei#
\$40	I2C_IO6	#frei#
\$20	I2C_IO5	#frei#
\$10	I2C_IO4	#frei#
\$08	I2C_IO3	#frei#
\$04	RESET_USB	High: No Reset für USB-PHY
\$02	RESET_ETH	High: No Reset für ETH-PHY
\$01	PWR_ETH	High: Power für Ethernet Phy

## 6.2 NandFlashDisk /FD/

Die resetfeste NandFlashDisk /FD/ liegt direkt im lokalen Nand-Flash, sie ist maximal 106 MB groß.

Der Formatbefehl lautet: FORM D /FD/FULL

## 6.3 Autostart (normal)

Ist auf der µSD-Card /H0/ eine Datei AUTO.EX vorhanden, so wird diese beim Programmstart mit der EX-Shell abgearbeitet, d.h. alle Zeilen dieser Datei werden der Reihe nach als Befehl interpretiert und abgearbeitet.

---

Ist auf der NandFlashDisk /FD/ eine Datei AUTO.EX vorhanden, so wird diese beim Programmstart mit der EX-Shell abgearbeitet, d.h. alle Zeilen dieser Datei werden der Reihe nach als Befehl interpretiert und abgearbeitet.

Die AUTO.EX auf /H0/ hat die höhere Priorität, d.h. wenn sie vorhanden ist wird eine AUTO.EX auf /FD/ nicht mehr eingelesen.

#### **6.4 Autostart (Recover)**

Ist auf der µSD-Card /H0/ eine Datei RECOVER.EX vorhanden, so wird diese beim Programmstart mit der EX-Shell abgearbeitet, d.h. alle Zeilen dieser Datei werden der Reihe nach als Befehl interpretiert und abgearbeitet.

Ist auf der NandFlashDisk /FD/ eine Datei RECOVER.EX vorhanden, so wird diese beim Programmstart mit der EX-Shell abgearbeitet, d.h. alle Zeilen dieser Datei werden der Reihe nach als Befehl interpretiert und abgearbeitet.

Die RECOVER.EX auf /H0/ hat die höhere Priorität, d.h. wenn sie vorhanden ist, wird eine RECOVER.EX auf /FD/ nicht mehr eingelsen.

#### **6.5 NotStart / Recover RTOS**

Das System überprüft den Booter bzw. das RTOS-UH im Nand mit einer Checksumme.

Wenn der 1. Booter OK ist und das 1. RTOS-UH (die normalen Arbeitsversionen) fehlerfrei sind und die Checksumme übereinstimmt, dann wird dieses RTOS-UH gestartet und der Autostart (normal) durchgeführt. Falls die Checksumme nicht stimmt, dann wird das Safe-RTOS gestartet und der Autostart (Recover) durchgeführt.

---

## **7 RTOS-UH im FLASH ablegen**

Nachdem das RTOS-UH als S-Record auf die Adresse \$800000 geladen und der Befehl START\_RTOS ausgeführt wurde, kann das dann laufende neue RTOS mit dem Befehl

NAND S System wird im Nand-Flash abgelegt.

NAND B schreibt den Booter in das Nand-Flash  
im Flash abgelegt werden.

NAND F schreibt den SafeBooter und das SafeRTOS in das Nand-Flash auf die Safe Bereiche.

## 8 MPC5125 I/O Pin Belegung

Hier wird die Default-Einstellung (**Fett**) der I/O Pins dargestellt.

Vom PreBooter wird der Prozessor entsprechend der folgenden Tabelle aufgesetzt, Änderungen können als Kalt/Warm-Scheibe bei Bedarf vorgenommen werden.

<i>Pin</i>	<i>Cfg- Adress</i>	<i>Alt 0</i>	<i>Alt 1</i>	<i>Alt 2</i>	<i>Alt 3</i>
R4	\$F000A004	LPC_CLK	TPA1		<b>GPIO04</b>
N2	\$F000A005	LPC_OE	PSC3_3		<b>GPIO05</b>
N3	\$F000A006	LPC_RW	PSC3_4		<b>GPIO06</b>
M1	\$F000A007	LPC_CS0			<b>GPIO07</b>
P3	\$F000A008	LPC_ACK	<b>NFC_CE1</b>	LPC_CS1	GPIO08
L1	\$F000A009	LPC_TS	NFC_CE2	<b>LPC_CS2</b>	
L2	\$F000A00A	<b>LPC_TSIZ1/LPC_AX0 2</b>	NFC_CE3	LPC_CS3	
N4	\$F000A00B	<b>LPC_TSIZ0/LPC_AX0 1</b>		LPC_CS4	
K1	\$F000A00C	<b>LPC_ALE/LPC_AX00</b>			
J1	\$F000A00D	LPC_AD31	PSC_MCLK		<b>GPIO09</b>
M3	\$F000A00E	LPC_AD30	CAN_CLK		<b>GPIO10</b>
L3	\$F000A00F	LPC_AD29			<b>GPIO11</b>
H1	\$F000A010	LPC_AD28			<b>GPIO12</b>
M4	\$F000A011	LPC_AD27			<b>GPIO13</b>
J2	\$F000A012	LPC_AD26			<b>GPIO14</b>
G1	\$F000A013	LPC_AD25			<b>GPIO15</b>
K3	\$F000A014	LPC_AD24			<b>GPIO16</b>
J3	\$F000A015	LPC_AD23			<b>GPIO17</b>
K4	\$F000A016	LPC_AD22		RCLPCTS	<b>GPIO18</b>
F1	\$F000A017	LPC_AD21			<b>GPIO19</b>
H3	\$F000A018	LPC_AD20			<b>GPIO20</b>
J4	\$F000A019	<b>NFC_ALE/LPC_AD19</b>		RST_CFG_LP CWA	
G2	\$F000A01A	<b>NFC_CLE/LPC_AD18</b>		RST_CFG_LP CMX	
G3	\$F000A01B	<b>NFC_RE/LPC_AD17</b>		RST_CFG_PL L_LOCK	
F3	\$F000A01C	<b>NFC_WE/LPC_AD16</b>			
E1	\$F000A01D	LPC_AD15	<b>PSC2_0</b>	RCSYSOC	GPIO21
H4	\$F000A01E	LPC_AD14	PSC2_1	RCPReDiv2	<b>GPIO22</b>

<i>Pin</i>	<i>Cfg- Adress</i>	<i>Alt 0</i>	<i>Alt 1</i>	<i>Alt 2</i>	<i>Alt 3</i>
E2	\$F000A01F	LPC_AD13	<b>PSC2_2</b>	RCPreDiv1	GPIO23
E3	\$F000A020	LPC_AD12	<b>PSC2_3</b>	RCPreDiv0	GPIO24
C1	\$F000A021	LPC_AD11	PSC2_4	RCSPMF3	<b>GPIO25</b>
D2	\$F000A022	LPC_AD10	PSC3_0	RCSPMF2	<b>GPIO26</b>
C2	\$F000A023	LPC_AD9	PSC3_1	RCSPMF1	<b>GPIO27</b>
E4	\$F000A024	LPC_AD8	PSC3_2	RCSPMF0	<b>GPIO28</b>
C3	\$F000A025	<b>LPC_AD7</b>		RCCPLL4	
B2	\$F000A026	<b>LPC_AD6</b>		RCCPLL5	
D5	\$F000A027	<b>LPC_AD5</b>		RCCPLL6	
C1	\$F000A028	<b>LPC_AD4</b>		RCLPCDBW1	
B3	\$F000A029	<b>LPC_AD3</b>		RCLPCDBW0	
B4	\$F000A02A	<b>LPC_AD2</b>		RCBMS	
A3	\$F000A02B	<b>LPC_AD1</b>		RCLOC1	
A4	\$F000A02C	<b>LPC_AD0</b>		RCLOC0	
P1	\$F000A02D	<b>NFC_CE0</b>			GPIO29
N1	\$F000A02E	<b>NFC_R/B0</b>			GPIO30
AB8	\$F000A02F	DIU_CLK	<b>PSC4_0</b>	USB1_DATA0	LPC_AX04
AA8	\$F000A030	DIU_DE	<b>PSC4_1</b>	USB1_DATA1	LPC_AX05
W11	\$F000A031	DIU_HSYNC	<b>PSC4_2</b>	USB1_DATA2	LPC_AX06
Y17	\$F000A032	DIU_VSYNC	<b>PSC4_3</b>	USB1_DATA3	GPIO31
AB9	\$F000A033	CAN3_RX	CLK_OUT2	DIU_LD00	<b>GPIO32</b>
Y10	\$F000A034	CAN3_TX	CLK_OUT3	DIU_LD01	<b>GPIO33</b>
AA10	\$F000A035	<b>DIU_LD02</b>	PSC4_4	USB1_DATA4	LPC_AX07
Y11	\$F000A036	DIU_LD03	<b>PSC5_0</b>	USB1_DATA5	LPC_AX08
AA11	\$F000A037	DIU_LD04	<b>PSC5_1</b>	USB1_DATA6	LPC_AX09
AB10	\$F000A038	DIU_LD05	<b>PSC5_2</b>	USB1_DATA7	GPIO34
AB11	\$F000A039	DIU_LD06	<b>PSC5_3</b>	USB1_STOP	GPIO35
Y12	\$F000A03A	DIU_LD07	PSC5_4	USB1_CLK	<b>GPIO36</b>
W13	\$F000A03B	CAN4_RX	PSC6_0	DIU_LD08	<b>GPIO37</b>
AB12	\$F000A03C	CAN4_TX	PSC6_1	DIU_LD09	<b>GPIO38</b>
Y13	\$F000A03D	DIU_LD10	<b>PSC6_2</b>	USB1_NEXT	GPIO39
AA13	\$F000A03E	DIU_LD11	<b>PSC6_3</b>	USB1_DIR	GPIO40
AB13	\$F000A03F	DIU_LD12	PSC6_4	USB2_DATA0	<b>GPT2[0]</b>
W14	\$F000A040	DIU_LD13	PSC7_0	USB2_DATA1	<b>GPT2[1]</b>
Y14	\$F000A041	DIU_LD14	PSC7_1	USB2_DATA2	<b>GPT2[2]</b>
AB14	\$F000A042	DIU_LD15	<b>PSC7_2</b>	USB2_DATA3	GPT2[3]
AA15	\$F000A043	CLK_OUT1	I2C3_SCL	DIU_LD16	<b>GPIO41</b>
Y15	\$F000A044	CLK_OUT2	I2C3_SDA	DIU_LD17	<b>GPIO42</b>

<b>Pin</b>	<b>Cfg- Adress</b>	<b>Alt 0</b>	<b>Alt 1</b>	<b>Alt 2</b>	<b>Alt 3</b>
AB15	\$F000A045	DIU_LD18	<b>PSC7_3</b>	USB2_DATA4	GPT2[4]
AB16	\$F000A046	DIU_LD19	PSC7_4	USB2_DATA5	<b>GPT2[5]</b>
AB17	\$F000A047	DIU_LD20	<b>PSC8_0</b>	USB2_DATA6	GPT2[6]
W16	\$F000A048	<b>DIU_LD21</b>	PSC8_1	USB2_DATA7	GPT2[7]
Y16	\$F000A049	DIU_LD22	<b>PSC8_2</b>	USB2_DIR	GPIO43
AA17	\$F000A04A	DIU_LD23	<b>PSC8_3</b>	USB2_NEXT	GPIO44
V4	\$F000A04B	I2C2_SCL	PSC8_4	USB2_CLK	<b>GPIO45</b>
U4	\$F000A04C	I2C2_SDA	PSC9_4	USB2_STOP	<b>GPIO46</b>
B15	\$F000A04D	<b>CAN1_TX</b>	PSC9_0	I2C2_SCL	GPIO47
A16	\$F000A04E	<b>CAN2_TX</b>	PSC9_1	I2C2_SDA	GPIO48
B18	\$F000A04F	I2C2_SCL	<b>PSC9_2</b>	CAN3_RX	GPIO49
A19	\$F000A050	I2C2_SDA	<b>PSC9_3</b>	CAN3_TX	GPIO50
AA1	\$F000A051	<b>FEC1_TXD2</b>	PSC2_0	USB2_DATA0	GPIO51
Y1	\$F000A052	<b>FEC1_TXD3</b>	PSC2_1	USB2_DATA1	GPIO52
Y4	\$F000A053	<b>FEC1_RXD2</b>	PSC2_2	USB2_DATA2	GPIO53
AA2	\$F000A054	<b>FEC1_RXD3</b>	PSC2_3	USB2_DATA3	GPIO54
U1	\$F000A055	<b>FEC1_CRCS</b>	PSC2_4	USB2_DATA4	GPIO55
W2	\$F000A056	<b>FEC1_TXER</b>	PSC3_0	USB2_DATA5	GPIO56
AA3	\$F000A057	<b>FEC1_RXD1</b>	PSC3_1	USB2_DATA6	GPIO57
W3	\$F000A058	<b>FEC1_TXD1</b>	PSC3_2	USB2_DATA7	GPIO58
V1	\$F000A059	<b>FEC1_MDC</b>	PSC3_3	USB2_DIR	GPIO59
Y5	\$F000A05A	<b>FEC1_RXER</b>	PSC3_4	USB2_NEXT	GPIO60
V2	\$F000A05B	<b>FEC1_MDIO</b>		USB2_CLK	GPIO61
AB2	\$F000A05C	<b>FEC1_RXD0</b>		USB2_STOP	GPIO62
W4	\$F000A05D	<b>FEC1_TXD0</b>		NFC_R/B1	GPIO63
W1	\$F000A05E	<b>FEC1_TCLK</b>	PSC0_0		GPIO04
AB4	\$F000A05F	<b>FEC1_RCLK</b>	PSC0_1	NFC_R/B2	GPIO05
AB3	\$F000A060	<b>FEC1_RXDV</b>	PSC0_2	NFC_R/B3	GPIO06
Y3	\$F000A061	<b>FEC1_TXEN</b>	PSC0_3		GPIO07
U3	\$F000A062	<b>FEC1_COL</b>	PSC0_4		GPIO08
Y9	\$F000A063	<b>USB1_DATA0</b>	PSC1_0	FEC2_RXD1	
W9	\$F000A064	<b>USB1_DATA1</b>	PSC1_1	FEC2_TXD1	
AB7	\$F000A065	<b>USB1_DATA2</b>	PSC1_2	FEC2_MDC	
AB6	\$F000A066	<b>USB1_DATA3</b>	PSC1_3	FEC2_RXER	
AA7	\$F000A067	<b>USB1_DATA4</b>	PSC1_4	FEC2_MDIO	
Y7	\$F000A068	<b>USB1_DATA5</b>	PSC4_0	FEC2_RXD0	
Y6	\$F000A069	<b>USB1_DATA6</b>	PSC4_1	FEC2_TXD0	
AB5	\$F000A06A	<b>USB1_DATA7</b>	PSC4_2	FEC2_TCLK	

---

<b>Pin</b>	<b>Cfg- Adress</b>	<b>Alt 0</b>	<b>Alt 1</b>	<b>Alt 2</b>	<b>Alt 3</b>
W6	\$F000A06B	<b>USB1_STOP</b>	PSC4_3	FEC2_RCLK	
Y8	\$F000A06C	<b>USB1_CLK</b>	PSC4_4	FEC2_RXDV	
AA5	\$F000A06D	<b>USB1_NEXT</b>		FEC2_TXEN	GPIO09
W7	\$F000A06E	<b>USB1_DIR</b>		FEC2_COL	GPIO10
T1	\$F000A06F	<b>SDHC1CLK</b>	NFC_CE1	FEC2_TXD2	GPIO11
T2	\$F000A070	<b>SDHC1CMD</b>	PSC5_0	FEC2_TXD3	GPIO12
T3	\$F000A071	<b>SDHC1_D0</b>	PSC5_1	FEC2_RXD2	GPIO13
T4	\$F000A072	<b>SDHC1_D1</b>	PSC5_2	FEC2_RXD3	LPC_CS5
R1	\$F000A073	<b>SDHC1_D2</b>	PSC5_3	FEC2_CRS	LPC_CS6
R2	\$F000A074	<b>SDHC1_D3</b>	PSC5_4	FEC2_TXER	LPC_CS7
D6	\$F000A075	<b>PSC_MCLK</b>			GPIO14
C11	\$F000A076	PSC0_0	SDHC2CMD	<b>GPT1[0]</b>	GPIO15
A12	\$F000A077	PSC0_1	SDHC2_D0	<b>GPT1[1]</b>	GPIO16
A13	\$F000A078	PSC0_2	SDHC2_D1	<b>GPT1[2]</b>	GPIO17
B13	\$F000A079	PSC0_3	SDHC2_D2	<b>GPT1[3]</b>	GPIO18
D11	\$F000A07A	PSC0_4	SDHC2_D3	<b>GPT1[4]</b>	CAN1_TX
C12	\$F000A07B	PSC1_0	SDHC2CLK	<b>GPT1[5]</b>	CAN2_TX
C13	\$F000A07C	PSC1_1	CAN_CLK	<b>GPT1[6]</b>	IRQ0
B14	\$F000A07D	PSC1_2	TPA2	<b>GPT1[7]</b>	IRQ1
D13	\$F000A07E	PSC1_3	CKSTPIN	NFC_R/B2	GPIO19
A15	\$F000A07F	PSC1_4	<b>CKSTPOUT</b>	NFC_CE2	GPIO20
B5	\$F000A080	J1850TX		NFC_CE3	<b>I2C1_SCL</b>
C6	\$F000A081	J1850RX		NFC_R/B3	<b>I2C1_SDA</b>

Feste, nicht programmierbare, I/O Pins:

<b>Pin</b>	<b>Funktion</b>
C9	<b>CAN1_RX</b>
B8	<b>CAN2_RX</b>
B6	<b>GPIO00</b>
A5	<b>GPIO01</b>
A6	<b>GPIO02</b>
C7	<b>GPIO03</b>

---

## 8.1 BaseBoard Setup

Folgende digitale Eingänge erzeugen ein RTOS-UH Event:

Hardware	RTOS-UH Event
CAP-Charge	\$80000000
CAP-Full	\$40000000
PowerFail	\$20000000
GPT1[0]	\$00008000
GPT1[1]	\$00004000
GPT1[2]	\$00002000
GPT1[3]	\$00001000
GPT1[4]	\$00000800
GPT1[5]	\$00000400
GPT1[6]	\$00000200
GPT1[7]	\$00000100

## 8.2 Geräte-Treiber

Device	Ldn	Drive	Size	Funktion
DIO	122	0	2 Byte	Digital In/Output
HEX	122	1	2 Byte	HexDreh Schalter Input
DIP	122	2	2 Byte	DipSwitch Input
LED	122	3	2 Byte	LED 2..4 Output
AUX	122	4	2 Byte	Aux Input
CAP	122	5	2 Byte	CAP Input
FRAM	91	0		FRAM
SPI1..7	91	1..7	1 Byte	Ext. SPI

## 8.3 Serielle Schnittstellen Atmos-Maurer

Es sind insgesamt 4 serielle Schnittstellen unterstützt und auf der VG64-Steckleiste verfügbar:

- /A1/ RS232 Bedienschnittstelle (38400 Baud) mit User-Interface
- /A2/ 1. RS422 Schnittstelle (9600 Baud)
- /A3/ 2. RS422 Schnittstelle (9600 Baud)
- /A4/ RS485 Schnittstelle (9600 Baud)

---

Zusätzlich ist auf der Pfostenleiste auf dem CPU-Modul die /A5/ (115200 Baud) mit User-Interface verfügbar.

#### 8.4 Gerät DIO – Bits

Lokale Digital I/O

Bit	15	14	13	12	10	9	8	7	6	5	4	3	2	1	0
Output														DO2	DO1
Input												DI4	DI3	DI2	DI1

Programmbeispiel in IEC1131:

```
...
VAR
  mDIO : DWORD := 0 ;
  inval : WORD := 0 ;
  outval: WORD := 0 ;
  pIN   : POINTER TO WORD ;
  pOUT  : POINTER TO WORD ;
  Stat  : DINT := 0 ; (* Status *)
END_VAR
...
IF mDIO = 0 THEN
  mDIO := _OPEN( '/DIO/mDIO' ) ;
  pIN  := ADR( inval ) ;
  pOUT := ADR( outval ) ;
ELSE
  Stat := _WRITE( Handle:=mDIO, Wait:=FALSE, LEN:=2, Buffer:=pOUT ) ;
  Stat := _READ( Handle:=mDIO, LEN:=2, Buffer:=pIN ) ;
END_IF
...
```

---

## 8.5 Gerät HEX – Bits (Input Only)

HexDreh Schalter

Bit	15	14	13	12	10	9	8	7	6	5	4	3	2	1	0
Input								Hex 1				Hex 2			

Programmbeispiel in IEC1131:

```
...
VAR
  mHEX : DWORD := 0 ;
  inval: WORD  := 0 ;
  pIN  : POINTER TO WORD ;
  Stat : DINT  := 0 ; (* Status *)
END_VAR
...
IF mHEX = 0 THEN
  mHEX := _OPEN( '/HEX/mHEX' ) ;
  pIN := ADR( inval ) ;
ELSE
  Stat := _READ( Handle := mHEX, LEN:=2, Buffer:=pIN ) ;
END_IF
...
```

## 8.6 Gerät DIP – Bits (Input Only)

DIP-Schalter

Bit	15	14	13	12	10	9	8	7	6	5	4	3	2	1	0
Input										S6	S5	S4	S3	S2	S1

Programmbeispiel in IEC1131:

```
...
VAR
  mDIP : DWORD := 0 ;
  inval: WORD  := 0 ;
  pIN  : POINTER TO WORD ;
  Stat : DINT  := 0 ; (* Status *)
END_VAR
...
IF mDIP = 0 THEN
  mDIP := _OPEN( '/DIP/mDIP' ) ;
  pIN := ADR( inval ) ;
ELSE
  Stat := _READ( Handle := mDIP, LEN:=2, Buffer:=pIN ) ;
END_IF
...
```

---

---

## 8.7 Gerät LED – Bits (Output Only)

User LEDs

Bit	15	14	13	12	10	9	8	7	6	5	4	3	2	1	0
Input													LE4	LE3	LE2

Programmbeispiel in IEC1131:

```

...
VAR
  mLED : DWORD := 0 ;
  outval: WORD :=0 ;
  pOUT : POINTER TO WORD ;
  Stat : DINT := 0 ; (* Status *)
END_VAR
...
IF mLED = 0 THEN
  mLED := _OPEN( '/LED/mLED' ) ;
  pOUT := ADR( outval ) ;
ELSE
  Stat := _WRITE( Handle:= mLED, Wait:=FALSE, LEN:=2, Buffer:=pOUT ) ;
END_IF
...

```

## 8.8 Gerät AUX – Bits (Input Only)

Auxiliary digital TTL-Inputs

Bit	15	14	13	12	10	9	8	7	6	5	4	3	2	1	0
Input					GPIO16	GPIO15	GPT25		GPT22	GPT21	GPT20			GPIO33	GPIO20

Programmbeispiel in IEC1131:

```

...
VAR
  mAUX : DWORD := 0 ;
  inval: WORD := 0 ;
  pIN : POINTER TO WORD ;
  Stat : DINT := 0 ; (* Status *)
END_VAR
...
IF mAUX = 0 THEN
  mAUX := _OPEN( '/AUX/mAUX' ) ;
  pIN := ADR( inval ) ;
ELSE
  Stat := _READ( Handle := mAUX, LEN:=2, Buffer:=pIN ) ;
END_IF
...

```

---

---

## 8.9 Gerät CAP – Bits (Input Only)

Status der Backup-Kondensatoren

Bit	15	14	13	12	10	9	8	7	6	5	4	3	2	1	0
Input													CAPLOW	CAPOK	CHRG

Programmbeispiel in IEC1131:

```
...
VAR
  mCAP : DWORD := 0 ;
  inval: WORD  := 0 ;
  pIN  : POINTER TO WORD ;
  Stat : DINT  := 0 ; (* Status *)
END_VAR

...
IF mCAP = 0 THEN
  mCAP := _OPEN( '/CAP/mCAP' ) ;
  pIN  := ADR( inval ) ;
ELSE
  Stat := _READ( Handle := mCAP, LEN:=2, Buffer:=pIN ) ;
END_IF

...
```

## 8.10 Gerät FRAM

Es steht ein 512 KB grosses FRAM zur Verfügung. Dort können Daten über einen Spannungsausfall sicher gespeichert werden.

Programmbeispiel in IEC1131:

```
...
VAR
  mFRAM : DWORD := 0 ;
  dLEN  : DINT  := 512000 ; (* 512000 Byte sichern *)
  pDAT  : POINTER TO WORD ;
  Stat  : DINT  := 0 ; (* Status *)
END_VAR

...
IF mFRAM = 0 THEN
  mFRAM := _OPEN( '/FRAM/mFRAM' ) ;
  pDAT  := ADR( mein_buffer ) ; (* Adresse des BackupBuffer *)
ELSE
  Stat:=_WRITE(Handle:=mDIO, Wait:=TRUE, LEN:=dLEN, Buffer:=pDAT);
  Stat := _READ( Handle := mDIO, LEN:=dLEN, Buffer:=pDAT ) ;
END_IF

...
```

---

## 8.11 Gerät SPI1 (8 Bit I/O – Extension)

I/O Extension auf AdapterBoard (optional)

Bit	7	6	5	4	3	2	1	0
Output	O8	O7	O6	O5	O4	O3	O2	O1
Input	E8	E7	E6	E5	E4	E3	E2	E1

Programmbeispiel in IEC1131:

```
...
VAR
  mEXT : DWORD := 0 ;
  inval : BYTE := 0 ;
  outval: BYTE := 0 ;
  pIN   : POINTER TO BYTE;
  pOUT  : POINTER TO BYTE;
  Stat  : DINT := 0 ; (* Status *)
END_VAR
...
IF mEXT = 0 THEN
  mEXT := _OPEN( '/SPI1/mEXT' ) ;
  pIN  := ADR( inval ) ;
  pOUT := ADR( outval ) ;
ELSE
  Stat := _WRITE( Handle:=mEXT, Wait:=FALSE, LEN:=1, Buffer:=pOUT ) ;
  Stat := _READ( Handle := mEXT, LEN:=1, Buffer:=pIN ) ;
END_IF
...
```