Card12

Hardware Version 1.1

User Manual

Contents

1. Overview

Card12 is an easy to use, credit card-sized Controller Module based on a Motorola HC12 Microcontroller. The Card12 Controller Module provides an easy way to evaluate the Microcontroller. It is a cheap, off-the-shelf solution for low to mid volume series applications.

An MC912D60A MCU is used on the Card12.D60A module. The module can also be ordered with an MC912DG128A MCU installed. This version is labelled Card12.DG128A.

Technical Data

- Motorola 16 Bit Microcontroller MC912D60A with TQFP-112 package
- 16 MHz external clock, 8 MHz bus clock
- 60 KB Flash Memory
- 2 KB RAM
- 1 KB EEPROM
- 8 Channel 16 Bit Timer with Input Capture / Output Compare
- 16 Channel 10 Bit A/D-Converter
- 2 Channel SCI with RS232 Transceiver (MAX232A)
- 4 Channel PWM
- Synchronous serial Interface SPI
- COP Watchdog, Clock Monitor
- CAN 2.0A/B bus interface with CAN driver (PCA82C250)
- Reset controller for Power Monitoring
- up to 80 free I/O-Lines
- all I/O-Lines are available via header connectors on the edge of the board
- mechanical dimensions: 54mm x 86mm (approx. 2.1" x 3.4")
- 5V Power Supply (additional 12V not anymore required for Flash-Programming)
- typ. power consumption: 50 mA

Extended Features of Card12.DG128A:

- Motorola 16 Bit Microcontroller MC912DG128A
- 128 KB Flash Memory
- 8 KB RAM
- 2 KB EEPROM
- two CAN-Interfaces (1x incl. driver, 1x TTL)

Package Contents

- Controller Module with MC912D60A (Card12.D60A) or with MC912DG128A (Card12.DG128A)
- TwinPEEKs Monitor Program (in the Flash Memory of the MCU)
- RS232 Cable (Sub-D9)
- two 50-pin header connectors plus one power supply connector
- Hardware Manual (this document)
- CD-ROM with Assembly Software, several Data Sheets, HC12 Reference Manual, Software Examples, C-Compiler Demo Version etc.

2. Quick Start

As no one likes to read lengthy manuals, we will summarize the most important things in the following section. If you need any additional information, please refer to the more detailed sections of this manual.

Here is how you can start:

- Please check the board for any damage due to transportation
- Connect the Controller Module via RS232 to a PC. The connection between Interface SER0 (ST3) on the Card12 and the PC is simply made using the flat cable which is in the box.
- On the PC, start a Terminal Program. An easy to use Terminal Program is OC-Console, which is available at no charge from our Website!
- Select a baudrate of 19200 Bd. Disable all hardware or software protocols.
- Connect a (stabilized!) Power supply, for instance at this point: Ground on ST7 Pin 2 / +5V on ST7 Pin 1
- But **at first**: double-check voltage and polarity!
- Once powered up, the Monitor Program starts and displays a short message. The Prompt appears and now the Module is waiting for your commands.

We hope you will enjoy working with the Card12 Controller Module!

Schematic Diagram

To ensure best visibility of all details, the Schematic Diagram of Card12 is provided as a seperate document.

3. Parts Location Diagram
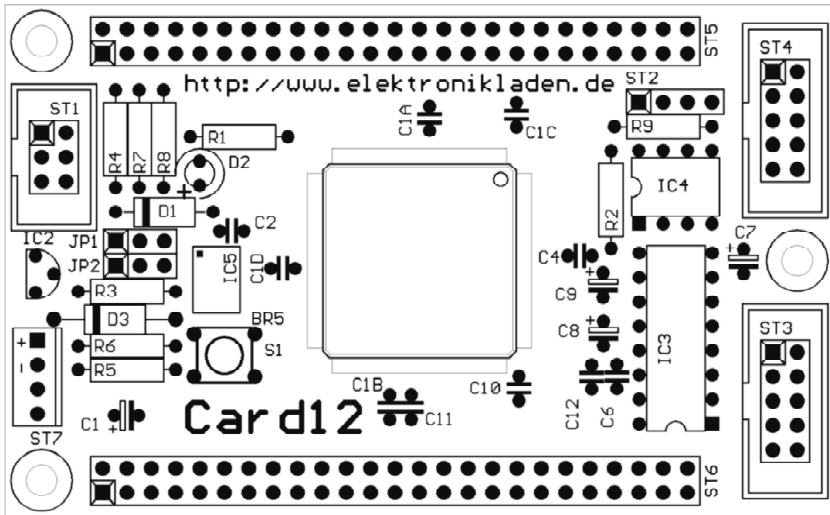
Place Plan - Component Side
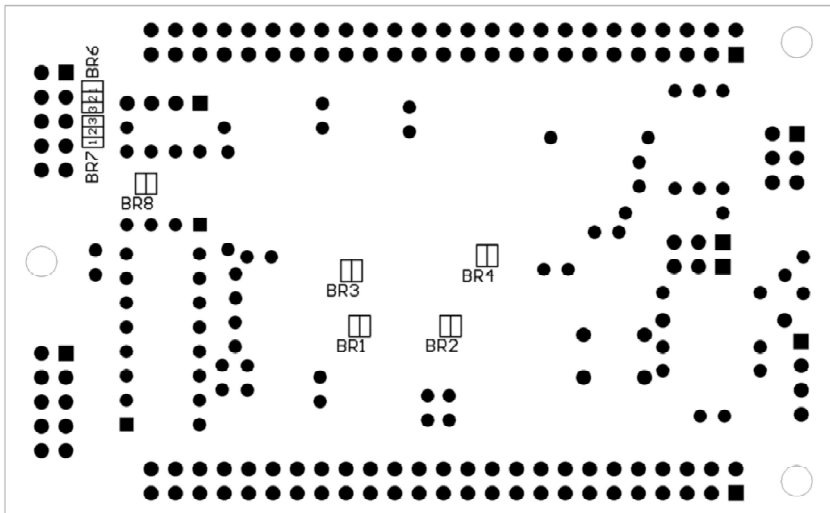
Solder Bridges on the Back Side of the PCB

4. Jumpers and Solder Bridges

Jumpers

JP1, JP2: MODA, MODB

1-2*	MODA/MODB=L (Single Chip Mode)
2-3	MODA/MODB=H (Expanded Wide Mode)

Solder Bridges

BR1: VDDAD

open	external supply of VDDAD
closed*	connected to VCC on board

BR2: VRH

open	external supply of VRH
closed*	connected to VCC on board

BR3: VFP

open	VFP Pin not connected (D60A, DG128A!!)
closed	VFP Pin connected with Programming Power Supply Input (12V required for D60, DG128)

BR4: VSTBY (DG128 only)

open*	external supply of VSTBY
closed	connected to VCC on board

BR5: EXTAL

open	Clock (EXTAL) supplied externally
closed*	Oscillator IC5 delivers Clock (EXTAL)

* = Factory Default Setting

BR6, BR7: RS232 TxD/RxD Select (SER1)

1-2*	RS232 configured as "Device"
	(for connection to a PC etc.)
2-3	RS232 configured as "Host"
	(for connection to a serial LCD etc.)

BR8: LCD Power Supply (SER1)

open*	VCC not available on RS232 port SER1
	(standard Sub-D connector layout)
closed	VCC available on RS232 port SER1
	(at Pin 9 of the Sub-D connector)

* = Factory Default Setting

5. Mechanical Dimensions

The following table summarizes the mechanical dimensions of the Card12. The values provide a basis for the design of carrier boards etc. Please note: Always check all mechanical dimensions using the real hardware module!

The reference point (0,0) is located at the "south/west" corner of the PCB. The PCB is orientated horizontally, as shown in the Parts Location Diagram (see above).

All data for holes/drills (B) refer to the center of the hole/drill, connectors (ST) are referenced by pin 1.

	X in Inch	Y in Inch
ST1	0.150	1.575
ST2	2.600	1.700
ST3	3.150	0.675
ST4	3.150	1.825
ST5	0.400	1.900
ST6	0.400	0.100
ST7	0.133	0.725
B1	0.150	0.150
B2	0.150	1.950
B3	3.250	1.050
PCB	3.400	2.100

6. Application Hints

In this section some details will be given on how to program the HC12 in general and the Card12 Controller Module in particular.

Please be aware that, even if this manual can provide some specific hints, it is impossible to cover all kinds of knowledge and techniques required to design a microcontroller program. Please refer to the data sheets of the silicon vendors and to the manuals of the software tools you use to find additional help.

Behaviour after Reset

As soon as the Reset input of the microcontroller is released, the MCU reads the Interrupt Vector at memory address $FFFE/F and then jumps to the address found there.

In the default delivery condition of Card12, the Flash Boot Block of the MCU ($E000-$FFFF) contains the Monitor Program TwinPeeks. The reset vector points to the start of this Monitor Software. As a result, TwinPeeks will start immediately after reset.

To automatically start a user program instead of the Monitor, TwinPeeks contains a special Autostart feature. The Autostart feature can be activated by connecting two I/O-lines of the MCU (see section Monitor Program for details).

Startup Code

Every Microcontroller firmware starts with a number of hardware initialization commands. For an MC912D60A/DG128A application program two issues have top priority: setting the Stack Pointer and disabling (or using) the COP Watchdog.

The following assembly lines show an example (which is actually derived from the starting sequence of the TwinPeeks Monitor):

```
main    lds  #StackPtr     ; Init Stack Pointer
        clr  COPCTL         ; Disable Watchdog
```

Important Notice: VFP-Pin of the Controller

The members of the first HC12 generation (D60, DG128 etc.) need a Flash programming voltage of 12V (see data sheet!). On the Card12, this voltage can be supplied via the power supply connector ST7 or via the BDM connector ST1. The solder bridge BR3 must be closed.

The new A-types (D60A, DG128A), which started to beccome available in the middle of the year 2001, have a modified Flash memory module with internal generation of programming power. **Solder bridge BR3 on the Card12 must always be open for A-type MCUs! This is to protect Pin 97 of the MCU. If this Pin was to be exposed to 12V, the microcontroller could be destroyed!**

Additional Information on the Web

Additional information about hard- and software of the Card12 Controller Module will be published on our Website, as it becomes available:

http://elmicro.com/en/card12.html

7. Monitor Program TwinPEEKs

Software Version 1.6

Serial Communication

The monitor program TwinPEEKs comminicates using the first RS232 interface ("SER0") at **19200 Baud**. The settings are: 8N1, no hardware or software handshake, no protocol.

Autostart Function

After Reset the TwinPEEKs Monitor checks, whether Pins PH6 and PH7 are connected or not. If there is a connection the Monitor jumps to address $8000 (*in the case of DG128(A): $4000*). In this way it is possible to automatically start a user program without changing the Reset Vector, which resides in the protected Flash Boot Block area.

Redirected Interrupt Vectors

The interrupt vectors reside at the end of the 64KB memory map, which falls in the area of the protected Flash Boot Block. In order to process interrupts of an application program, the interrupt vectors are redirected to a RAM area by the Monitor. The method is similar to the behaviour of the HC11 in Special Bootstrap Mode.

The user's application program installs an interrupt vector by placing a jump instruction at the RAM pseudo vector location. Example: Follow these steps to use the SPI interrupt:

```
ldaa #$06           ; JMP Opcode
staa $07C7          ; SPI Pseudo Vector
ldd  #isrFunc       ; Jump Address
std  $07C8          ; SPI Pseudo Vector + 1
```

The following listing shows, which interrupt is redirected to which pseudo address in RAM by the monitor program:

```
FFC2 :                  org $ffc2                 ; 912D60 *and* 912DG128

FFC2 : 07A6             dc.w  RAMTOP-90           ; CGM LLH
FFC4 : 07A9             dc.w  RAMTOP-87           ; MSCAN Tx
FFC6 : 07AC             dc.w  RAMTOP-84           ; MSCAN Rx
FFC8 : 07AF             dc.w  RAMTOP-81           ; MSCAN Err
FFCA : 07B2             dc.w  RAMTOP-78           ; Pulse Accu B Overflow
FFCC : 07B5             dc.w  RAMTOP-75           ; MDCU
FFCE : 07B8             dc.w  RAMTOP-72           ; KWUG, KWUH
FFD0 : 07BB             dc.w  RAMTOP-69           ; MSCAN Wake Up
FFD2 : 07BE             dc.w  RAMTOP-66           ; ATD
FFD4 : 07C1             dc.w  RAMTOP-63           ; SCI1
FFD6 : 07C4             dc.w  RAMTOP-60           ; SCI0
FFD8 : 07C7             dc.w  RAMTOP-57           ; SPI
FFDA : 07CA             dc.w  RAMTOP-54           ; Pulse Accu Input Edge
FFDC : 07CD             dc.w  RAMTOP-51           ; Pulse Accu Overflow
FFDE : 07D0             dc.w  RAMTOP-48           ; Timer Overflow
FFE0 : 07D3             dc.w  RAMTOP-45           ; TC7
FFE2 : 07D6             dc.w  RAMTOP-42           ; TC6
FFE4 : 07D9             dc.w  RAMTOP-39           ; TC5
FFE6 : 07DC             dc.w  RAMTOP-36           ; TC4
FFE8 : 07DF             dc.w  RAMTOP-33           ; TC3
FFEA : 07E2             dc.w  RAMTOP-30           ; TC2
FFEC : 07E5             dc.w  RAMTOP-27           ; TC1
FFEE : 07E8             dc.w  RAMTOP-24           ; TC0
FFF0 : 07EB             dc.w  RAMTOP-21           ; RTI
FFF2 : 07EE             dc.w  RAMTOP-18           ; IRQ
FFF4 : 07F1             dc.w  RAMTOP-15           ; XIRQ
FFF6 : 07F4             dc.w  RAMTOP-12           ; SWI
FFF8 : 07F7             dc.w  RAMTOP-9            ; Illegal Opcode
FFFA : 07FA             dc.w  RAMTOP-6            ; COP Fail
FFFC : 07FD             dc.w  RAMTOP-3            ; Clock Monitor Fail
FFFE : F000             dc.w  main                ; Reset
```

This table is valid for both D60(A) and DG128(A), despite the fact that the highest used RAM address is $3FFF with the DG128(A) compared to $07FF with the D60(A). In addition, the DG128(A) has five additional interrupt vectors:

```
FFB8 :                  org $ffb8                 ; 912DG128 *only!*

FFB8 : 3F97             dc.w  RAMTOP-105          ; MSCAN1 Tx
FFBA : 3F9A             dc.w  RAMTOP-102          ; MSCAN1 Rx
FFBC : 3F9D             dc.w  RAMTOP-99           ; MSCAN1 Err
FFBE : 3FA0             dc.w  RAMTOP-96           ; MSCAN1 Wake Up
FFC0 : 3FA3             dc.w  RAMTOP-93           ; IIC
```

8. Monitor Commands

TwinPEEKs Command Summary:

Dump: D (adr1 (adr2))

- Display Memory from Address "adr1" to "adr2" ("adr2" is not included)

Edit: E (addr (byte))

- edit memory starting at address "addr"
- If "byte" is specified, this "byte" is simply written to "addr". Otherwise the command is executed in an interactive mode.

Fill: F adr1 adr2 byte

- Fills the memory area from "adr1" to "adr2" ("adr2" not included) with the value of "byte"

Goto: G (addr)

- Jumps to a user program starting at "addr"

Help: H

- Command Summary

Load: L (offs)

- Load an executable program in Motorola S-Record format (S1 type as well as S2 type records are supported)
- The sending terminal program (PC side) must be able to wait for an acknowledge (*) after each line!
- If "offs" is specified, every address in the S-Record file is moved by an offset.

Move: M adr1 adr2 adr3

- Copy memory starting at "addr1" and ending at "addr2"-1 to "addr3".

Page: P (page)

- Select a memory page ("page"=0..7). This page will be visible in the paged area ($8000 to $BFFF = 16 KB).
  DG128(A) only!

Bulk: X (array)

- Erase the Flash Memory
- If "array" is *not* specified: erase the whole Flash Memory (excl. upper Boot Block)
- if "array" = 0: erase only the lower 28 KB Flash Array
- if "array" = 1: erase only the upper 32 KB Flash Array
- The Monitor (residing in the upper Boot Block) will *not* be erased, anyway. (to overwrite the Monitor code a BDM loader/programmer is required)
  DG128(A) only: array = 0..3 (32 KB each)

9. Memory Map

Card12.D60A

Start	Ende	Belegung
$0000	$01FF	Control Registers
$0200	$07FF	1.5KB (of 2KB) RAM (the lower 0,5KB are hidden by the Control Registers) TwinPEEKs uses the top 512 bytes
$0C00	$0FFF	1KB EEPROM
$1000	$7FFF	28KB (lower) Flash Memory Array
$8000	$FFFF	32KB (upper) Flash Memory Array TwinPEEKs uses the top 8KB

Card12.DG128A

Start	Ende	Belegung
$0000	$03FF	Control Registers
$0800	$0FFF	2KB EEPROM
$2000	$3FFF	8KB RAM TwinPEEKs uses the top 512 bytes
$4000	$7FFF	16KB Flash (equals page 6)
$8000	$BFFF	16KB Flash page 0 (any page 0..7, selectable by PPAGE)
$C000	$FFFF	16KB Flash (equals page 7) TwinPEEKs uses the top 8KB