AcQuisition Technology bv

Headquarters:
Raadhuislaan 27a
5341 GL OSS
THE NETHERLANDS

Postal address:
P.O. Box 627
5340 AP OSS
THE NETHERLANDS

Phone: +31-412-651055
Fax: +31-412-651050
Email: info@acq.nl
WEB: http://www.acq.nl

M392

12-/16-bit ADC M-module with 16x Single-Ended Inputs

User Manual

Version 1.6

Disclaimer:

The information in this document has been carefully checked and is believed to be entirely reliable. However, no responsibility is assumed for inaccuracies. AcQuisition Technology does not assume any liability arising out of the application or use of any product or circuit described herein; neither does it convey any license under its patent rights nor the rights of others. AcQuisition Technology products are not designed, intended, or authorized for use as components in systems intended to support or sustain life, or for any other application in which the failure of an AcQuisition Technology product could create a situation where personal injury or death may occur, including, but not limited to AcQuisition Technology products used in defence, transportation, medical or nuclear applications. Should the buyer purchase or use AcQuisition Technology products for any such unintended or unauthorized application, the buyer shall indemnify and hold AcQuisition Technology and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that AcQuisition Technology was negligent regarding the design or manufacture of the part.

Printed in The Netherlands.

CONTENTS

AcQuisition Technology bv
P.O. Box 627, 5340 AP
Oss, The Netherlands

1. INTRODUCTION

1.1. VALIDITY OF THE MANUAL

This document is of version 1.6 and provides information on the use of the M392 R0.x running firmware of revision 1.3 and up.

1.2. PURPOSE

The manual serves as instruction for configuring the M-module, the connection of measurement sources and the integration on a M-module carrier. Furthermore it gives the user additional information for special applications and configuration possibilities of the assembly. Detailed information concerning the individual assemblies (data sheets etc.) are not part of this manual. This manual also contains a description of the provided example software.

1.3. SCOPE

The scope of this manual is the usage of the M392 12-/16-bit ADC M-module with 16x Single-Ended Inputs and the APIS based example software written in ANSI-C.

1.4. DEFINITIONS, ACRONYMS AND ABBREVIATIONS

AcQ	AcQuisition Technology bv
APIS	AcQ Platform Interface Software
M-module	Mezzanine I/O concept according to the M-module specification
Platform	Combination of hardware and operating system

1.5. NOTES CONCERNING THE NOMENCLATURE

Hex numbers are marked with a leading “0x”-sign: for example: 0x20 or 0xff.

File names are represented in italic: *filename.txt*

Code examples are printed in `courier` .

1.6. OVERVIEW

In chapter two a short description of the M392 hardware can be found. The next chapter covers the installation and setup of the M-module as well as the connection of the measurement sources. In chapter 4 the operation and usage of the M392 is described in detail. AcQ provides APIS based example software for the M392, the software is described in chapter 5. Finally this document contains an Annex containing a bibliography, component image, technical data, document history and a programming example.

This page contains no essential data.

AcQuisition Technology bv
P.O. Box 627, 5340 AP
Oss, The Netherlands

2. PRODUCT OVERVIEW

2.1. INTRODUCTION

The M392 is an 16-channel analog input M-module with single-ended inputs and an ADC resolution of 12- or 16-bit. The M392 has one ADC core with an analog multiplexer for signal routing. A local TMS320C203 DSP provides amongst other functions the signal routing, data acquisition, range selection and measurement correction. The operation of the local DSP is transparent to the user. Measurement results are continuously updated in dual ported memory without interference of the host. The front-end of the M392 is optically isolated, powered from either an on-board isolated DC/DC converter or from an externally applied source.

The M392 is available in several versions which differ with respect to the conversion resolution and isolated power supply:

- M392/V16/WDC 16-bit resolution, with on-board DC/DC converter
- M392/V16/NDC 16-bit resolution, no DC/DC converter (external supply required)
- M392/V12/WDC 12-bit resolution, with on-board DC/DC converter
- M392/V12/NDC 12-bit resolution, no DC/DC converter (external supply required)

The M392 has four input ranges which are software selectable per channel: +/-10V bipolar, 0...10V unipolar, +/-5V bipolar and 0...5V unipolar. The module features an active low-pass 2-pole input filter with a cut-off frequency of 1kHz and a programmable moving average filter for each channel individually.
The M392 family has no potentiometers since measurement correction is handled by the DSP. Correction is done by the DSP according to factory calibrated system parameters stored in an onboard EEPROM. The acquisition time is software programmable from 20 ：seconds up to 1 millisecond per channel and channels can be individually enabled or disabled.

2.2. TECHNICAL OVERVIEW

Below an overview of the functionality of the M392 is listed.

- 16x single ended analog inputs.
- 16-bit 10 : s sampling analog-to-digital converter.
- Optional 12-bit version available.
- Accuracy 0.1%.
- 16x onboard active low-pass 2-pole input filters with a cut-off frequency of 1kHz.
- No potentiometers.
- Analog front-end optically isolated.
- Optional onboard isolated power supply (DC/DC converter) for analog front-end.
- With on-board DC/DC converter the module is operated from a single 5V supply.
- Data acquisition handled by TMS320C203 DSP, transparent for the user.
- Measurement data corrected by the DSP according to calibration data stored in an onboard. EEPROM.
- Programmable moving average filter.
- Measurement results continuously updated in dual ported memory.
- System configuration programmable through a command interface in dual ported memory.
- User programmable voltage input ranges:
  - +/10V Bipolar (default)
  - 0..10V Unipolar
  - +/5V Bipolar
  - 0..5V Unipolar.
- Input range per channel configurable.
- Software programmable update rate up to 50 kSPS (default).
- A08/D16 M-module interface.
- Host interrupts supported.
- Identification EEPROM.

AcQuisition Technology bv
P.O. Box 627, 5340 AP
Oss, The Netherlands

3. INSTALLATION AND SETUP

3.1. UNPACKING THE HARDWARE

The hardware is shipped in an ESD protective container. Before unpacking the hardware, make sure that this takes place in an environment with controlled static electricity. The following recommendations should be followed:

- Make sure your body is discharged to the static voltage level on the floor, table and system chassis by wearing a conductive wrist-chain connected to a common reference point.
- If a conductive wrist-chain is not available, touch the surface where the board is to be put (like table, chassis etc.) before unpacking the board.
- Leave the board only on surfaces with controlled static characteristics, i.e. specially designed anti static table covers.
- If handling the board over to another person, touch this persons hand, wrist etc. to discharge any static potential.

IMPORTANT: Never put the hardware on top of the conductive plastic bag in which the hardware is shipped. The external surface of this bag is highly conductive and may cause rapid static discharge causing damage. (The internal surface of the bag is isolating.)

Inspect the hardware to verify that no mechanical damage appears to have occurred. Please report any discrepancies or damage to your distributor or to AcQuisition Technology immediately and do not install the hardware.

3.2. CONNECTING THE MODULE

This section describes the pin-assignments of the 25-pole D-sub female front connector and the 24-pole female P2 header.

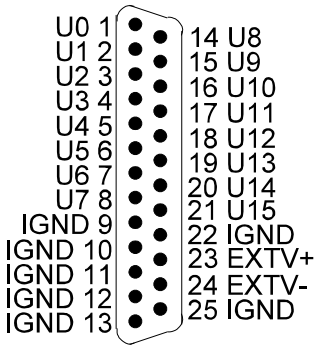


Figure 1 25 pole D-sub

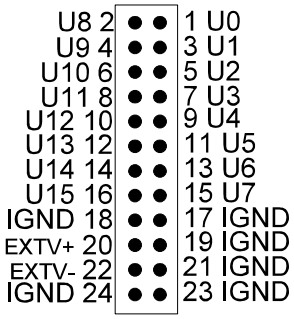


Figure 2 24 pole female header

The signals labelled with Ux are single-ended inputs where x is the channel number. The IGND pins are connected to the isolated ground of the M392 and are provided as a reference for the single ended inputs.
The analog front-end is powered from either an onboard DC/DC converter (M392/V16/WDC and M392/V12/WDC) or from an externally applied power supply (M392/V16/NDC and M392/V12/NDC) depending on the module version.
If the module has an board DC/DC converter the external power pins EXTV+ and EXTV- must not be connected. If the module has no onboard DC/DC converter an external power supply of +/-15V DC must be applied: +15V DC must be applied to EXTV+ and -15V DC must be applied to EXTV-, the power supply ground must be connected to IGND. For details on the power requirements please refer to section 6.3.
For a description of the input circuitry refer to section 4.3.

Note: Although the digital subsystem of the M392 is powered from the M-module bus the module does not initialize when no isolated power supply is available.

AcQuisition Technology bv
P.O. Box 627, 5340 AP
Oss, The Netherlands

4. FUNCTIONAL DESCRIPTION

This chapter gives a detailed description of the M392. The M-module interface is described as well as the operation of the M-module.

4.1. BLOCK DIAGRAM

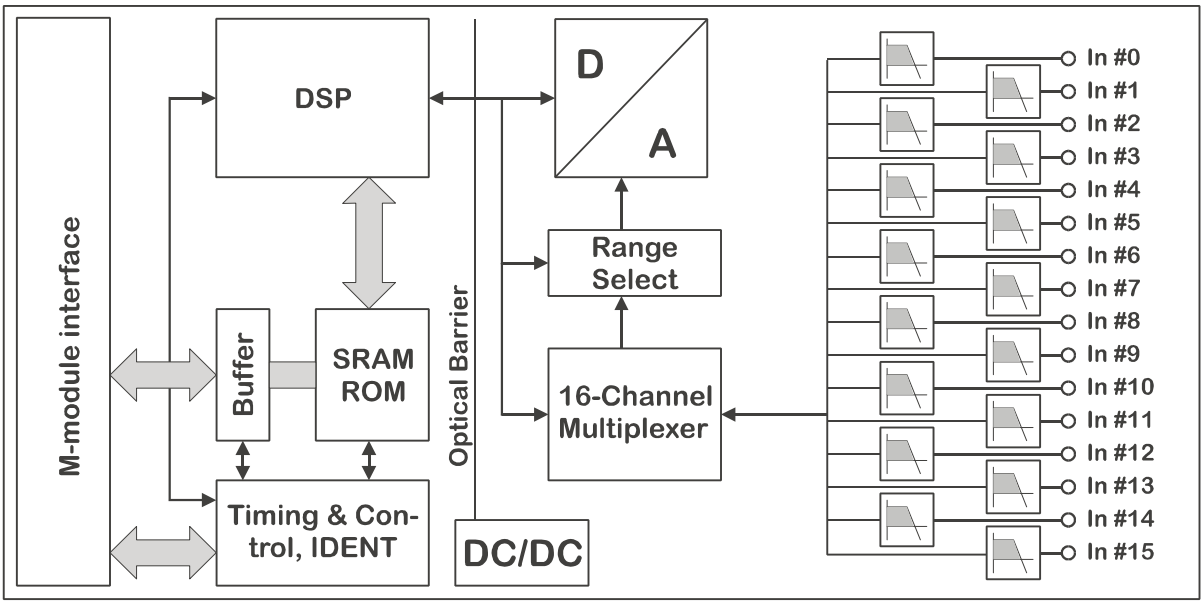


Figure 3 M392 Block Diagram

4.2. M-MODULE INTERFACE

The M392 has a A08/D16 INT A compliant M-module interface. The following sections give an overview of the M-module interface and memory organization of the M392.

4.2.1. MEMORY MAP OVERVIEW

The following table shows the address map of the M392 M-module. All addresses are relative to the base address of the M-module.

Offset	Name	Description
0x00	MDR0	Measurement data channel 0
....	....	....
....	....	....
0x1e	MDR15	Measurement data channel 15
0x20	RES	Reserved
....	....	....
....	....	....
0x3e	RES	Reserved
0x40	C_CMD	Command interface: command field
0x42	C_RES	Command interface: result field
0x44	C_PRM0	Command interface: parameter 0
....	....	....
....	....	....
0x62	C_PRM15	Command interface: parameter 15
0x64	GLOBSTAT	Global status
0x66	IMASK	Interrupt mask
0x68	IFLAG	Interrupt flags
0x6a	RES	Reserved
....	....	....
....	....	....
0x7e	RES	Reserved
0x80	PAGEREG	Page register, must be written with 0x100
0x82	CTRLREG	Host control register
0x84	RES	Reserved
....	....	....
....	....	....
0xfc	RES	Reserved
0xfe	EEPREG	Identification EEPROM register

Caution: Only 16-bit wide accesses are allowed. Do not write reserved fields.

Fields described in the memory map are either located in dual ported memory, offset 0x00 to offset 0x80 provided that the page register is set to 0x0100 or are implemented as hardware registers: page register, host control register and the EEPROM register.
The following sections give a detailed description of memory locations and hardware registers.

AcQuisition Technology bv
P.O. Box 627, 5340 AP
Oss, The Netherlands

4.2.2. MEASUREMENT DATA WORDS

The memory locations MDR0 to MDR15 at offset 0x00* to offset 0x20* contain the measurement results of channel 0 to channel 15. The data is represented in straight binary format. The following table shows the relation between the binary output codes and the corresponding ideal input values.

16-bit resolution		
Range	Code	Input
+/- 10 V (default)	0x0000 0x8000 0xffff	-10 V 0 V +9.999695 V
0..10 V	0x0000 0x8000 0xffff	0 V +5 V +9.999847 V
+/- 5 V	0x0000 0x8000 0xffff	-5 V 0 V +4.999847 V
0..5 V	0x0000 0x8000 0xffff	0 V +2.5 V +4.999924 V

12-bit resolution		
Range	Code	Input
+/- 10 V (default)	0x000 0x800 0xfff	-10 V 0 V +9.9951 V
0..10 V	0x000 0x800 0xfff	0 V +5 V +9.9976 V
+/- 5 V	0x000 0x800 0xfff	-5 V 0 V +4.9998 V
0..5 V	0x000 0x800 0xfff	0 V +2.5 V +4.9976 V

The measurement results are continuously updated in the dual ported memory by the DSP.
For details on the data acquisition refer to section 4.4.

* The page register must be set to 0x100

4.2.3. COMMAND INTERFACE

The M392 features a command interface for communication between the host and the local DSP. Through the command interface it is possible to configure special features of the M392, like setting the acquisition time, programming ranges etc. For operation in default mode (+/- 10V input range @ 20 :s acquisition time) no commands have to be executed.

The command interface consists of a command field, a result field and a parameter field:

Offset*	Name	Description
0x40	command	command field
0x42	result	result field
0x44	parm0	first parameter
....	....	....
....	....	....
0x62	parm15	last parameter

* The page register must be set to 0x100

The M392 can accept commands at any time provided that a previous requested operation is completed. Before a command is given make sure the command field is empty (no outstanding command present). Then write the parameters if any to the parameter field. After that the command must be written to the command field. When a command is handled by the DSP, return parameters if any, are stored in the parameter field, the result is updated in the result field and finally the command field is cleared by the DSP.
The DSP is ready to accept new commands.

For a complete overview of the available commands and the usage refer to section 4.6.1.

4.2.4. GLOBAL STATUS WORD

The global status word at offset 0x64 (with the page register set to 0x0100) contains status information of the M-module. Under normal conditions the global status word is zero. A non-zero status word indicates a possible defect of the M-module and measurement results are not available. If after a power-up the problem still exists contact AcQ for repair.

Under normal circumstances the global status word does not have to be accessed by the host.

4.2.5. INTERRUPT MASK WORD

The interrupt mask word is defined as follows:

IMR Offset: 0x66 (page = 0x0100)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RDY	res	res	res	res	res	res	res	res	res	res	res	res	res	res	res
Reset:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Write only															

RDY Measurement cycle ready
RDY signals a measurement cycle ready (global status word is zero) or a fault situation (global status word is non-zero).

The interrupt mask register may only be written by the host. A bit set un-masks the corresponding interrupt. A bit cleared masks the interrupt.
Setting RDY will cause host interrupts to be generated at the end of every measurement cycle.

AcQuisition Technology bv
P.O. Box 627, 5340 AP
Oss, The Netherlands

4.2.6. INTERRUPT FLAG WORD

The interrupt flag word is defined as follows:

IFR Offset: 0x68 (page = 0x0100)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Read only

Since the M392 has currently one interrupt source the interrupt flag word is not used. However the interrupt flag register is implemented for compatibility reasons only.

4.2.7. PAGE REGISTER

The dual ported memory of the M392 is accessed using a page register for the upper address bits. The first 128 bytes of the address space of the M392 M-module is used as a “window” into the dual ported memory as described in the previous section.

PAGEREG Offset: 0x80

15	9	8	7	6	5	4	3	2	1	0
not used	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7
Reset:										
	x	1	x	x	x	x	x	x	x	x

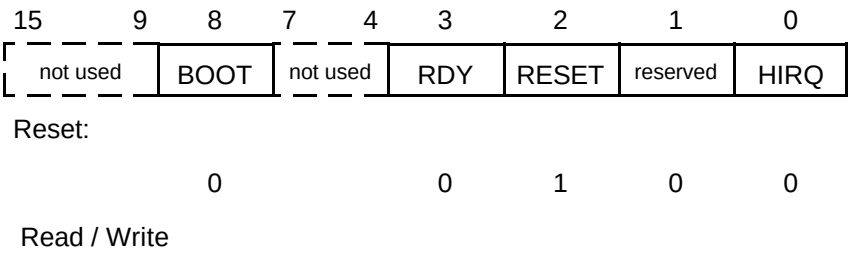
Write only

A16 - A7 Dual ported memory address bits
These bits determine the current dual ported memory page selected. A page has a length of 128 bytes (A6 - A0).

For normal operation the page register must be set to 0x0100. The page register is available for service purposes and must not be set to any other value than 0x0100.

4.2.8. HOST CONTROL REGISTER

Offset: 0x82



RDY Ready Bit
This bit must be cleared by the host by writing a logic ‘1'. The bit will be set by the M-module when all channels are updated.

RESET Local reset
When asserted, the M392 will be kept in reset state. After power-up the M-module is in reset.

HIRQ Host Interrupt Request
This bit reflects the state of the host interrupt line, one ‘1' means asserted. Writing a one ‘1' to this location clears the pending interrupt.

BOOT Boot selection
Default the local firmware is booted from ROM, however it is possible to boot the DSP from dual ported memory, therefore the BOOT bit must be set to ‘1' prior to a reset. The boot bit is available for service and debug purposes, so details are beyond the scope of this manual.

Caution: Do not use read-modify-write operations on the host control register (AND/OR instructions).
Reserved bits must be written by a logic ‘0‘.

AcQuisition Technology bv
P.O. Box 627, 5340 AP
Oss, The Netherlands

4.2.9. IDENTIFICATION EEPROM

The identification of an M-module is implemented using a serial EEPROM with a 64*16 word organisation. The industry-standard component 93C46 is used in order to make the identification compatible throughout the complete range of available M-modules. Access to the identification EEPROM takes place through the following register:

EEPROM Offset: 0xFE

15 … 4	3	2	1	0
not used	unused	CS	CLK	DI/O
Reset:				
	0	0	0	x
Read / Write				

CS Chip-Select
This bit corresponds to the chip select input of the EEPROM.

CLK Clock
This bit corresponds to the clock input of the EEPROM.

DI/O Data input/output
This bit corresponds to the data input of the EEPROM when writing and data output of the EEPROM when reading.

For information on controlling the EEPROM refer to the NM93C46 data sheets. For information on the memory organization refer to the M-module Specification.

Note: The identification EEPROM on the M392 is only accessible by the host when the M-module is kept in reset. (Bit #2 in the host control register ‘set’).

Caution: On the M392 the EEPROM is not only used for identification purposes but also for storage of calibration data used by the local DSP . Therefore writing to or clearing from the EEPROM by the host will result in erroneous behaviour.

4.3. INPUT CIRCUITRY

This section describes the input circuitry of the M392, this might be useful for connecting measurement sources.

The picture below shows the simplified schematics of input channel 0 of the M392, channel 1 to channel 15 are similar.

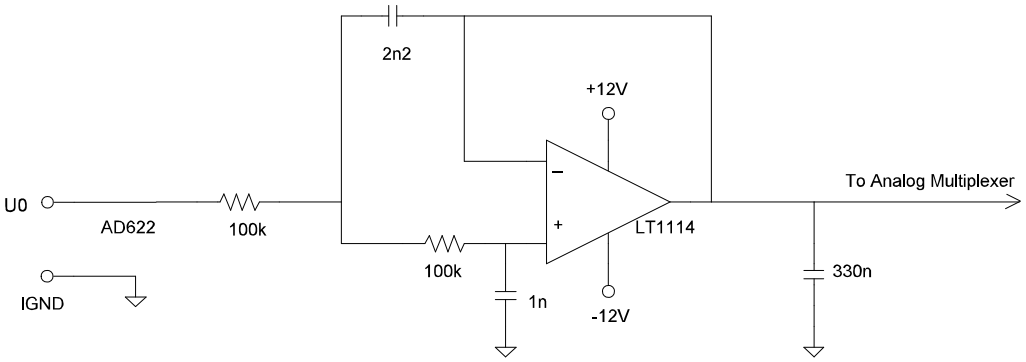


Figure 4 M392 Input Circuitry

The input circuitry consists of an 2-pole active low pass filter wit a cut-off frequency of 1Khz build around a precision amplifier, the LT1114. The filter output goes to an input of an analog multiplexer. The output of the multiplexer is connected to the range selection circuit and finally to the input of an ADC. For details on the multiplexer and the ADC refer to section 4.4.

Detailed information on connecting sensors or other measurement sources are beyond the scope of this manual.

AcQuisition Technology bv
P.O. Box 627, 5340 AP
Oss, The Netherlands

4.4. DATA ACQUISITION

The data acquisition is handled by the onboard TMS320C203 DSP without any user intervention. However this section describes the acquisition process for a better understanding of the operation and usage of the M392.

The operation will be described according to the simplified block diagram below:

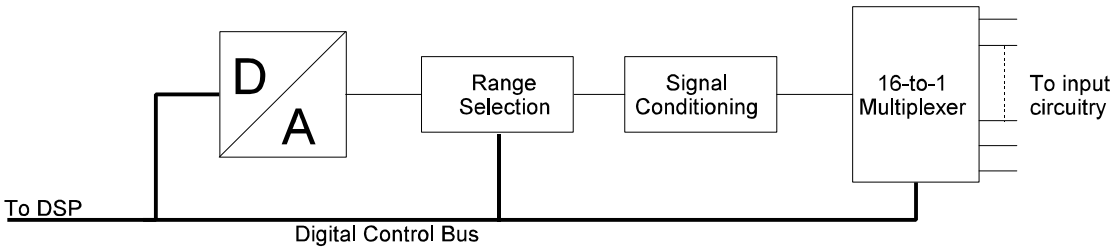


Figure 5 M392 Data Acquisition Control

Via the digital control bus the DSP is able to select an input channel and to set the range selector to the required position. The range will be set according to the configuration of the corresponding input channel. The acquisition loop will continuously scan the input channels starting at channel 0 followed by channel 1 and after channel 15 a new loop is initiated. Channels can be individually disabled or enabled from servicing.
The acquisition time is the time between two samples and is software programmable between 20 µs and 1ms. The acquisition loop time depends on the number of enabled channels and the programmed acquisition time. The maximum throughput is 20 µs (one channel enabled).

The used ADC is an ADS7809 from Burr-Brown with a resolution of 16-bits. The DSP takes care of correcting the samples according to calibration parameters stored in EEPROM.
Furthermore the DSP is used for implementation of a digital moving average (MA) filter for noise compensation and better DC accuracy. The MA filter can be configured for averaging over 8, 16, 32, 64 or 128 samples and it can be disabled.

The flowchart below shows the acquisition loop:

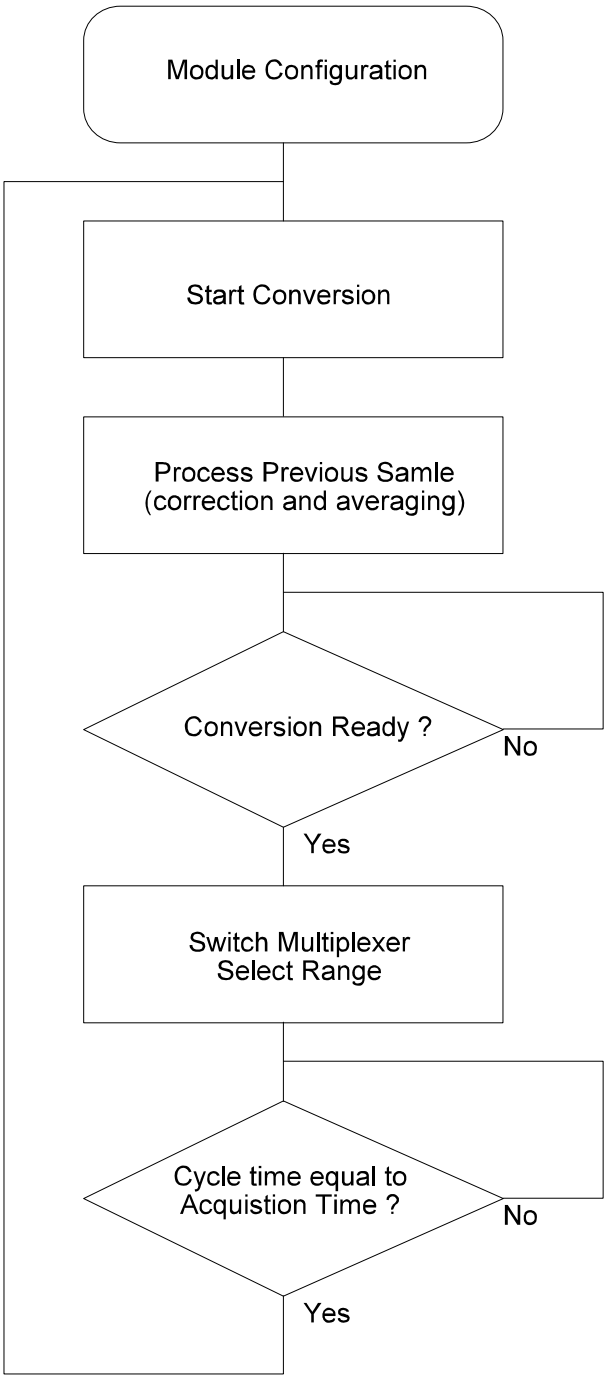


Figure 6 M392 Acquisition Cycle

AcQuisition Technology bv
P.O. Box 627, 5340 AP
Oss, The Netherlands

The resolution of the Analog-to-Digital process is either 12-bit or 16-bit depending on the version of the M-module. The table below contains the specification of the A-to-D conversion.

Resolution	16 bits or 12 bits
Accuracy	16 bit voltage inputs: 0.1% 12-bit voltage inputs: 0.5 %
Maximum throughput rate	50 kHz

For details on the A-to-D converter refer to the Burr-Brown ADS7809 data sheet.

For more information on channel configuration, configuring acquisition timing and dis-/enabling channels refer to section 4.2.3.

4.5. RESET PROCEDURE

After power-up the M392 is kept in reset. The Identification EEPROM can be read by the host. Before any control operation are possible the M392 must be started according to the following reset procedure written in ANSI-C:

```
#define CTRLREG   ((volatile unsigned short *)(<base>+0x82))
#define PAGEREG   ((volatile unsigned short *)(<base>+0x80))
#define GLOBSTAT  ((volatile unsigned short *)(<base>+0x64))

*CTRLREG = 0x0004;            /* put M-module in reset */
*PAGEREG = 0x0100;            /* write 0x0100 to page register */
*CTRLREG = 0x0000;            /* release reset */

while(!(*CTRLREG & 0x00008)); /* wait until RDY bit is set */

if (GLOBSTAT == 0)            /* check global status */
      /* M392 ready for usage */
else
      /* Hardware failure */
```

The global status word at offset 0x64 contains status information of the M-module. Under normal conditions the global status word is zero. A non-zero status word indicates a possible defect of the M-module and measurement results are not available. If after a power-up the problem still exists contact AcQ for repair.

Make sure only 16-bit write and read accesses are performed. After the reset sequence the M392 will continuously acquire the input channels.

4.6. M-MODULE OPERATION

The M392 is configured through the command interface described in section 4.2.3. This paragraph gives an overview of the available user commands and their usage.

4.6.1. COMMAND SET OVERVIEW

This section lists the commands available to the user, the possible return codes and contains an programming example of command execution on the M392.

Command set:

Command	Hex	Description
DONE	0x0000	command completed
SYNC	0x5a5a	synchronize firmware, sign of live
VERSION	0x1000	returns version information
SETUP	0x0001	setup channel configuration
SETACQTIME	0x0002	set acquisition time
MAFILTER	0x0003	configure the moving average filter

Result codes:

Result	Hex	Description
NOERR	0x0000	no error
UNKCMD	0x8001	unknown command
ADCFAIL	0x8002	a-to-d converter failure
ILLREQ	0x8003	illegal request

Below you can find a software example written in ANSI-C:

```
#define C_CMD     ((volatile unsigned short *)(<base>+0x40))
#define C_RES     ((volatile unsigned short *)(<base>+0x42))
#define C_PRM(x)  ((volatile unsigned short *)(<base>+0x44+((x&0xf)<<1))


*C_PRM(0) = 0;                          /* write parameter 0 */
*C_CMD = 5a5a;                          /* execute synchronization command */
while (*C_CMD);                         /* wait for completion */

if (*C_RES) {                           /* check result */
      /* handle error */
}
else
      /* command successful */
```

4.6.2. SYNC COMMAND

SYNC **Sign of live**

Command: 0x5a5a

Inputs: None

Outputs: None

Function: This function is intended to check whether or not the DSP is running. The local software takes no special action and terminates the execution normally by setting the result field to NOERR and clearing the command field.

AcQuisition Technology bv
P.O. Box 627, 5340 AP
Oss, The Netherlands

4.6.3. VERSION COMMAND

VERSION	**Get M-module information**

Command: 0x1000

Inputs: None

Outputs:

Parameter 0:	M-module number (should be 392d)
Parameter 1:	Firmware version number
Parameter 2:	Reserved
Parameter 3:	Resolution: (16d or 12d)
Parameter 4:	DSP clock frequency 0: 57MHz, 1: 40MHz

Function: This function returns information about the hardware configuration and the firmware. When the function returns, parameter 0 contains the M-module number which should be 392(d). Parameter 1 contains the firmware version number: for example 10(d) which means version 1.1. Parameter 2 is reserved and is merely provided for compatibility reasons and parameter 3 specifies the resolution in number of bits: either 16(d) or 12(d). Parameter 4 passes information about the DSP clock frequency, normally this parameter is zero which indicates a DSP frequency of 57MHz. Optionally the M392 can be factory configured for 40MHz, in this case parameter 4 returns 1.

4.6.4. SETUP COMMAND

SETUP	**Setup Acquisition Time**

Command: 0x0001

Inputs: Channel 0 control
Channel 1 control
.....
Channel 15 control

Outputs: None

Function: This command is provided to setup the channel configuration. The channel configuration specifies the input range for a channel or causes the channel to be disabled. Parameter 0 contains the channel 0 control word, parameter 1 contains the control word for channel 1 etc.

All control words must be specified (parameter 0 to 15) before the command is given. The table below lists the available control words:

Input Range	Control Word
+/-10 V bipolar	0x0000 (default)
0...10 V unipolar	0x0001
+/-5 V bipolar	0x0002
0...5 V unipolar	0x0003
Channel disabled	0x0004

AcQuisition Technology bv
P.O. Box 627, 5340 AP
Oss, The Netherlands

4.6.5. SETACQTIME COMMAND

SETACQTIME — Setup acquisition time

Command: 0x0002

Inputs: Acquisition time

Outputs: None

Function: This command is provided to configure the acquisition time. Input parameter 0 specifies the throughput. The table below shows the relation between the input parameter and the acquisition time.

Acquisition time	Input parameter
20 µs	0x0000 (default)
100 µs	0x0001
500 µs	0x0002
1000 µs	0x0003

4.6.6. MAFILTER COMMAND

MAFILTER — Configure moving average filter

Command: 0x0003

Inputs: Filter selection

Outputs: None

Function: This command is provided to configure the digital moving average (MA) filter. Input parameter 0 specifies the filter behaviour. The table below shows the relation between the input parameter and the MA filter.

Filter	Input parameter
disabled	0x0000
8 samples	0x0001
16 samples	0x0002
32 samples	0x0003 (default)
64 samples	0x0004
128 samples	0x0005

4.7. READING SAMPLES

After reset the M392 starts gathering samples without user intervention. The default operation is all channels enabled and the range is set to +/-10V (or 0...20mA in case of current inputs). Data can be obtained from the measurement data registers MDRx (x is the channel number) at any time. However the M392 features a signalling mechanism which indicates that all channels are refreshed. This is done with the RDY bit in the control register CTRLREG.

It is recommended to use the RDY bit in the host control register to check if new samples are available. The RDY bit must be cleared by the host by writing a logic '1' to bit #3 of the host control register. The RDY bit will be set by the M392 when a measurement cycle is completed.
An example programmed in ANSI-C:

```
#define CTRLREG   ((volatile unsigned short *)(<base>+0x82))
#define PAGEREG   ((volatile unsigned short *)(<base>+0x80))

#define MDR0      ((volatile unsigned short *)(<base>+0x00))
#define MDR1      ((volatile unsigned short *)(<base>+0x02))
etc.

unsinged short data[16];

/*
 *    Execute infinite loop for logging samples
 */

*CTRLREG = 0x0008;            /* clear RDY bit by writing a logic '1' */

for (;;) {
      while(!(*CTRLREG & 0x0008));  /* wait until RDY bit is set */

      *CTRLREG = 0x0008;      /* clear RDY bit by writing a logic '1' */

      data[0] = *MDR0;        /* get measurement results */
      data[1] = *MDR1;
      ....

      /* store/compute measurement results */

}
```

4.8. INTERRUPT HANDLING

The M392 is capable of generating interrupts of type A, software-end-of-interrupt. Because the M392 is not capable of delivering an interrupt vector, this must be handled by the carrier board.

When the RDY-bit in the IMASK register at offset 0x66 is set the M392 will generate an interrupt request when a measurement cycle is complete.

The interrupt service routine must acknowledge the interrupt request by writing a logic "1" to bit #0 of the host control register.

AcQuisition Technology bv
P.O. Box 627, 5340 AP
Oss, The Netherlands

An example for using interrupts:

```
#define CTRLREG   ((volatile unsigned short *)(<module base>+0x82))
#define PAGEREG   ((volatile unsigned short *)(<module base>+0x80))
#define IMASK     ((volatile unsigned short *)(<module base>+0x66))

void m392isvc(void)
{
      if (*CTRLREG & 0x0001)        /* if interrupt caused by M392 */
            *CTRLREG = 0x0001;      /* then clear pending interrupt */

      if (*CTRLREG & 0x0008) {      /* if ready flag is set */
            *CTRLREG = 0x0008;      /* then clear RDY flag */
            ...                     /* signal measurement cycle is ready */
      }
}

void main(void)
{
      ...
      install_irq(..., &m392isvc);  /* install interrupt handler */
      *IMR = 0x8000;                /* enable RDY interrupt */
      ...
      ...                           /* program body */
}
```

Note: To avoid performance problems when the RDY interrupt is enabled, most applications will require a lower acquisition time than the default value of 20 µs.

This page contains no essential data.

AcQuisition Technology bv
P.O. Box 627, 5340 AP
Oss, The Netherlands

5. SOFTWARE

This chapter describes the example software which is available for the M392 12-/16-bit ADC M-module with 16x Single-Ended Inputs. The example software is available in ANSI-C source code and consists mainly of an M392 function library which provides functions for easy access to the M392 and a demo program which illustrates the usage of the software library.
The M392 library functions are APIS based, physical accesses and interrupt support are handled by APIS, AcQ Platform Independent Interface Software. The next section contains general information on APIS, for detailed information please refer to the APIS Programmer's Manual.

5.1. APIS SUPPORT

AcQ produces and supports a large number of standard M-modules varying from networking and process I/O to motion control applications. Physically, the M-modules are supported by a large number of hardware platforms: VMEbus, PCI, CompactPCI as well as a wide variety of operating systems: OS-9, Windows NT, Linux etc.
APIS offers a way to program platform independent applications, example- and test software for controlling hardware. Application software written for APIS only needs re-compiling for a particular platform and is operational with little effort (provided that the application is operating system independent).

5.1.1. CONCEPT

Hardware accesses to registers and memory are handled by APIS. Some minor operating system dependent functions frequently used in hardware related software, such as interrupt handling and a delay function are also provided by APIS.
APIS platform support consists of an Application Programming Interface in the form of definition files coded in ANSI-C and platform dependent M-modules e.g. source files, libraries and/or drivers.
In the most simple outline, a platform dependent APIS module consist of nothing more then macro definitions in which APIS calls are substituted by direct hardware accesses. But in most cases an APIS module will consist of a library with interface routines and in some implementations a device driver is needed for interaction with the operating system.

5.1.2. API

The Application Programming Interface for APIS is implemented in two ANSI-C coded definition files: *apis.h* which contains general definitions and *platform_apis.h* which contains platform specific definitions and references to the APIS function calls.
The application source file must include the APIS header file *apis.h*. Porting of the application to a platform, consists of re-compiling the source code with a defined pre-processor macro for selection of the used platform. The APIS header file contains generic APIS definitions and includes a platform specific header file according to the platform selection macro.
APIS calls are translated to the platform specific calls in the APIS header file and the platform specific definition file *platform_apis.h* (*platform* is a name that identifies a hardware and operating system combination, e.g. i4000os9).
The macro PLATFORM must be defined, either via a pre-processor definition provided at compile time or via a macro-definition in the application source.

5.1.3. CODE GENERATION

APIS based example software is available in ANSI-C source code. Source code files must be compiled with the pre-processor definition PLATFORM set to a valid value, conforming the target platform.
Building of the example software for the M392 is platform dependent, for details refer to the release notes of the APIS support package of the target platform and the APIS Programmer's Manual.
Examples of APIS supported platforms are i4000os9, i2000dos, i3000win etc.

5.2. TYPE DEFINITIONS AND STRUCTURES

The table below contains a list and description of all types and structures used in the M392 example software (standard ANSI-C types are not listed).

Name	Type	Description
INT8	char	8-bit signed data
UINT8	unsigned char	8-bit unsigned data
INT16	short	16-bit signed data
UINT16	unsigned short	16-bit unsigned data
INT32	long	32-bit signed data
UINT32	unsigned long	32-bit unsigned data
PHA8	volatile unsigned char *	8-bit physical access
PHA16	volatile unsigned short *	16-bit physical access
PHA32	volatile unsigned long *	32-bit physical access
APIS_PATH	unsigned long	APIS physical path ID
APIS_HANDLE	void *	APIS physical path handle
APIS_WIDTH	int	APIS access size in bytes
IDCODE	union {	ID EEPROM contents
	struct {	
	short synccode,	Sync code (0x5346)
	modnum,	M-module number (binary coded)
	revision,	revision number
	modchar,	M-module characteristics
	res[4];	reserved
	char manstr[16];	manufacturer string
	} id;	ID data
	short reg[16];	raw data
	}	

AcQuisition Technology bv
P.O. Box 627, 5340 AP
Oss, The Netherlands

5.3. FUNCTION REFERENCE

This section contains the reference of the functions provided by the M392 software library.

5.3.1. GENERAL FUNCTIONS

m392_boot() — Boot the M392

Syntax: `int m392_boot (APIS_HANDLE handle)`

Description: Boot the M392 from ROM.

Arguments: APIS_HANDLE handle
Hardware path handle returned by apis_open()

Returns: 0 or -1 in case of hardware failure

Example: m392_boot(handle); /* boot m392 */

m392_ramboot() — Boot the M392 from RAM

Syntax: `int m392_ramboot (APIS_HANDLE handle)`

Description: Download bootfile to dual port memory, the DSP will load the bootfile from RAM. To enable this function the macro BOOT_FROM_RAM must be defined as TRUE and a boot-image must be linked to the application.

Arguments: APIS_HANDLE handle
Hardware path handle returned by apis_open()

Returns: 0 or -1 in case of hardware failure

Example: m392_boot(handle); /* boot m392 */

m392_cmd() — Execute command

Syntax:
```
int m392_cmd (APIS_HANDLE handle, UINT16 command, UINT16 *pArgList,
      int ni, int no)
```

Description: Copy command parameters to the parameter field in shared ram of the M-module. Execute command and wait for the command to be completed. Copy the result parameters and return with the result code obtained from the firmware.

Below is a table with all available commands:

Command	Description
SYNC	Synchronize firmware, sign of live
VERSION	Returns version information
SETUP	Setup channel configuration
SETACQTIME	Set acquisition time
MAFILTER	Configure the moving average filter

Arguments:

APIS_HANDLE handle
Hardware path handle
UINT16 command
Firmware command
void *pArgList
Pointer to the command parameter list
int ni
Number of input command parameters
int no
Number of output command parameters

Returns: Command result code

Command result codes:

Result	Description
NOERR	No error
UNKCMD	Unknown command
ADCFAIL	A-to-D converter failure
ILLREQ	Illegal request

Example:
```
m392_cmd( handle, SYNC, (UINT16*)(&params), 0, 0);
```


AcQuisition Technology bv
P.O. Box 627, 5340 AP
Oss, The Netherlands

m392_install_irq() — Install interrupt handler

Syntax: `int m392_install_irq( APIS_HANDLE handle)`

Description: Install interrupt handler and unmask interrupt.

Arguments: APIS_HANDLE handle
hardware path handle

Returns: Returns NOERR or -1 in case of error installing the interrupt handler

Example:
```
m392_install_irq( handle);  /* install interrupt handler */
```

m392_remove_irq() — Remove interrupt handler

Syntax: `int m392_remove_irq( APIS_HANDLE handle)`

Description: Removes interrupt handler and masks interrupt.

Arguments: APIS_HANDLE handle
Hardware path handle

Returns: NOERR

Example:
```
m392_remove_irq( handle); /* remove interrupt handler */
```

m392_irqh() — Interrupt handler

Syntax: `int m392_irqh( APIS_HANDLE handle)`

Description: The interrupt handler will clear the interrupt of the m392. Pending interrupts are cleared with the IACK command. Code which has to be executed when the interrupt occurs must be added to this function.

Arguments: APIS_HANDLE handle
Hardware path handle

Returns: 0 or -1 if interrupt is not from M392

m392_setupchan() — Setup channels

Syntax: `int m392_setupchan( APIS_HANDLE handle, UINT16 *inp_range_list)`

Description: Configure input range of each channel. See following table for valid input ranges.

Input range	Description
V10BI	+/- 10 V bipolar
V10UNI	0 .. 10 V unipolar
V5BI	+/- 5 V bipolar
V5UNI	0 .. 5 V unipolar
DISCHAN	Disable channel

Arguments: APIS_HANDLE handle
Hardware path handle
UINT16 *inp_range_list
Pointer to an array with input ranges for channel 0-15

Returns: NOERR

Example:
```
m392_setupchan(handle, (UINT16*)(&inp_range_list));
```

m392_setupallchan() — Setup all channels

Syntax: `int m392_setupallchan( APIS_HANDLE handle, UINT16 inp_range)`

Description: Configures channel (0-15) with the same input range. See table in m392_setupchan() for valid input range.

Arguments: APIS_HANDLE handle
Hardware path handle
UINT16 inp_range
Input range for channel 0-15

Returns: NOERR

Example:
```
m392_setupallchan( handle, V10BI);
```


AcQuisition Technology bv
P.O. Box 627, 5340 AP
Oss, The Netherlands

m392_setupacqtime() — Setup acquisition time

Syntax: `int m392_setupacqtime( APIS_HANDLE handle, UINT16 throughput)`

Description: Configure the acquisition time for each channel. See following table for valid acquisition times.

throughput	Description
TIME20us	20 microseconds
TIME100us	100 microseconds
TIME500us	500 microseconds
TIME1000us	1000 microseconds

Note: The acquisition time is per channel.
E.g. If the acquisition time is set to 20 μs and all channels are enabled, each channel is updated after 16 * 20 = 320 μs.

Arguments: APIS_HANDLE handle
Hardware path handle
UINT16 throughput
Acquisition time

Returns: NOERR

Example:
```
m392_setupacqtime(handle, TIME1000us);
```

m392_configmafilter() — Configure MA filter

Syntax: `int m392_configmafilter( APIS_HANDLE handle, UINT16 filter_beh)`

Description: Configures moving average filter. If MA filter is enabled, it will be applied to all channels. See following table for valid values for filter_beh.

filter_beh	Description
DISFILT	Disable filter
SAMPLE8	8 samples
SAMPLE16	16 samples
SAMPLE32	32 samples (default)
SAMPLE64	64 samples
SAMPLE128	128 samples

Arguments: APIS_HANDLE handle
hardware path handle
UINT16 filter_beh
filter behaviour

Returns: NOERR or ILLREQ when filter_beh has an illegal value

Example:
```
m392_configmafilter(handle, SAMPLE64);
```

m392_readchannel() — Read data from channel

Syntax:
```
int m392_readchannel( APIS_HANDLE handle, UINT16 channel, UINT16
    *data)
```

Description: Reads data from one channel.

Arguments: APIS_HANDLE handle
hardware path handle
UINT16 channel
number of channel
UINT16 *data
pointer to data from channel

Returns: NOERR or -1 when read command has failed.

Example:
```
m392_readchannel(handle, 3, &data);
```


AcQuisition Technology bv
P.O. Box 627, 5340 AP
Oss, The Netherlands

m392_check_rdybit() — Check if RDY bit is set

Syntax: `int m392_check_rdybit( APIS_HANDLE handle)`

Description: Checks if RDY bit is set. The RDY bit is set when one measurement cycle is completed.

Arguments: APIS_HANDLE handle
hardware path handle

Returns: Returns TRUE when RDY bit is set and FALSE when it is not set.

Example:
```
m392_wait4rdy(handle);
```

m392_wait4rdybit() — Wait till RDY bit is set

Syntax: `void m392_wait4rdybit( APIS_HANDLE handle)`

Description: Polls RDY bit until it is set. The RDY bit is set when one measurement cycle is completed.

Arguments: APIS_HANDLE handle
hardware path handle

Returns: Nothing

Example:
```
m392_wait4rdy(handle);
```

m392_clear_rdybit() — Clear RDY bit

Syntax: `void m392_clear_rdybit( APIS_HANDLE handle)`

Description: Clears RDY bit.

Arguments: APIS_HANDLE handle
hardware path handle

Returns: Nothing

Example:
```
m392_clear_rdybit(handle);
```

5.4. SOFTWARE DISTRIBUTION

This section gives an overview of the software distribution.

File	Description
M392\SOFTWARE\m392rel.txt	Release notes
M392\SOFTWARE\LIB\m392lib.c	APIS based ANSI-C M392 software library
M392\SOFTWARE\LIB\m392defs.h	Definitions for M392 software library
M392\SOFTWARE\EXAMPLE\m392demo.c	Demo program
M392\SOFTWARE\EXAMPLE\makefile.os9	Maker for OS-9
M392\SOFTWARE\EXAMPLE\makefile.bor	Maker for Borland-C
M392\SOFTWARE\EXAMPLE\makefile.lin	Maker for Linux-gcc
MMODID\SOFTWARE\LIB\modideep.c	M-module ID EEPROM software library
MMODID\SOFTWARE\LIB\ideeprom.h	M-module ID EEPROM definitions

M392 example software is APIS based, therefore APIS support for the target platform is required for code generation.

The following figure is an example of the M392 software integrated in the APIS environment with as target platform the i4000/OS-9.

```
+--PROJECT                          AcQ's distribution
     +--APIS                        APIS basis distribution
     |    +--DOC                    APIS documentation
     |    +--SOFTWARE
     |         | readme.txt         Distribution overview
     |         +--COMMON
     |         |    +--DEFS
     |         |    |    apis.h     General definitions
     |         |    +--OS9TRAP
     |         |    |   +--CMDS
     |         |    |       apistrap  OS-9 trap handler
     |         |    +--.....
     |         +--I4000OS9          i4000/OS-9 support
     |         |    relnotes.txt    Release notes / version info
     |         |    apis_i4000os9.h Platform specific definitions
     |         |    apis_i4000os9.c Platform support library
     |         +--.....
     +--M392                        M392 distribution
     |    +--DOC                    M392 documentation
     |    +--SOFTWARE
     |         | m392rel.txt        Release notes / version info
     |         +--LIB               M392 support libraries
     |         |    m392defs.h
     |         |    m392lib.c
     |         +--EXAMPLE           M392 example software
     |              m392demo.c
     +--MMODID
          +---SOFTWARE
               +---LIB
                    modideep.c      M-module ID EEPROM library
                    ideeprom.h      M-module ID definitions
```

Code generation is platform dependent, for information on building the software please refer to the release notes of the target platform and the APIS Programmer's Manual.

AcQuisition Technology bv
P.O. Box 627, 5340 AP
Oss, The Netherlands

6. ANNEX

6.1. BIBLIOGRAPHY

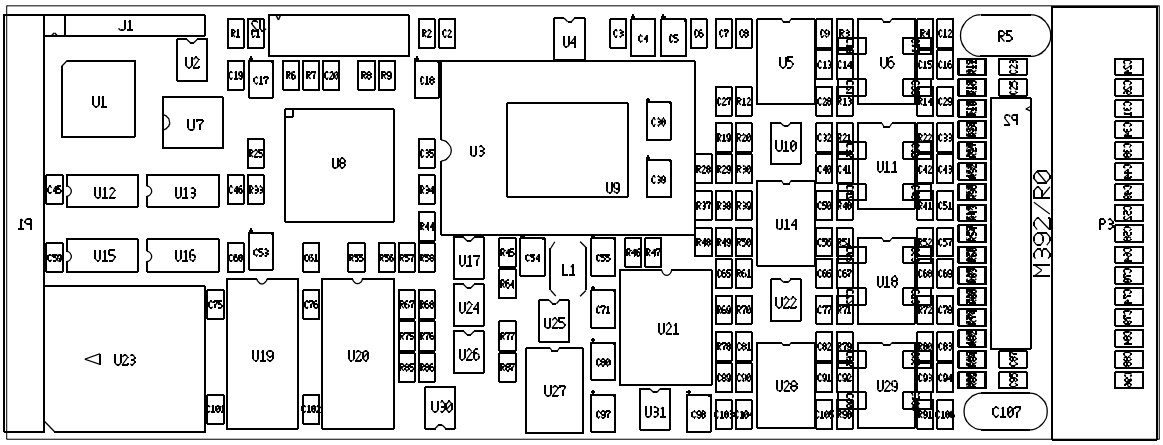
Specification for M-module interface and physical dimensions:
M-module specification manual, April 1996, MUMM.
Simon-Schöffel-Strasse 21, D-90427 Nürnberg, Germany.

APIS Programmer's Manual
AcQuisition Technology
P.O. Box 627, 5340 AP Oss, The Netherlands.

Data Sheet of the NM93C46
Memory Databook 1992 edition (400069)
National Semiconductor Corporation
1111 West Bardin Road; Arlington, TX76017, United States

Data Sheet of the ADS7809
Mixed Signal Products
Burr-Brown IC Data Book
Burr-Brown Corporation; P.O. Box 11400 Tucson, AZ 85706-1400, United States

6.2. COMPONENT IMAGE



6.3. TECHNICAL DATA

Slots on the base-board:
Requires one 16-bit M-module slot.

Connection:
To base-board via 40 pole M-module interface.
To peripheral via 25 pole D-sub connector.
To peripheral via 24 pole header connector

Power supply with on-board DC/DC converter:
+5VDC ±10%, typical 850mA.

Power supply without on-board DC/DC converter:
+5VDC ±10%, typical 250mA (obtained from M-module bus)
+15VDC ±5%, maximal 200mA (externally applied)
-15VDC ±5%, maximal 100mA (externally applied)

Temperature range:t
Operating: 0..+60°C.
Storage : -20..+70°C.

Humidity:
Class F, non-condensing.

6.4. DOCUMENT HISTORY

- Version 1.0
  First release
- Version 1.1
  Adapted for local firmware
- Version 1.2
  APIS support added and new lay-out
- Version 1.3
  Example code changed
- Version 1.4
  Modified for firmware version R1.3 and up
- Version 1.5
  Function m392_ramboot() added to software library
- Version 1.6
  Description of M392 for external isolated power supply (M392/V16/NDC, M392/V12/NDC) added.

AcQuisition Technology bv
P.O. Box 627, 5340 AP
Oss, The Netherlands