



Prepared
IDT/Patrik Jonsson

Date
December, 2006

Approved

Information Class Reference
Unclassified

GMS

Contents

1 Introduction	2
1.1 Purpose	2
1.2 Overview	2
1.3 Definitions	3
2 References	4
3 System configuration	5
3.1 Hardware	5
3.1.1 MCU	5
3.1.2 EPROM	5
3.1.3 SRAM.....	5
3.1.4 Flash-ROM.....	6
3.1.5 Analog inputs	6
3.1.6 Digital I/O.....	6
3.1.7 Counter	6
3.1.8 RTC	6
3.1.9 Watchdog.....	7
3.1.10 Serial ports, external devices.....	7
3.1.11 Service port.....	8
3.1.12 BDM.....	8
3.1.13 Battery	9
3.2 Software.....	9
3.2.1 Overview	9
3.2.2 Boot-ROM.....	13
3.2.3 Core	14
3.2.4 Application	15

3.2.5 Software updates	17
3.2.6 Development environment	18

Supporting authors Proof read LV/ LV/Morgan Nöjdberg Tech. proof read LVÖ/
LV/Örjan Westin Lang. proof read LV

DOCUMENT HISTORY

Edition	Datum	PRR	Description
1.2	1998-05-20		First edition in English.

1. Introduction

1.1 Purpose

The purpose of this document is to specify the design and configuration of the hardware and software of GMS.

This is a design description of the GMS hardware and software. The description does not deal with internal hardware solutions, such as wiring, circuitry, etc. Neither does this document describe software details such as individual classes, functions, structures, etc. For more detailed information in these respects, refer to the circuit diagrams and the software source code.

1.2 Overview

GMS is a sensing station, primarily intended for measuring of meteorological data. The GMS station is to be included in the VViS-system used by Vägverket (the Swedish National Road Administration) as a further development of MS2. It is also to be used for collection of weather data within the MS2000 system.

GMS is based on an Enator-developed sensor board of a similar function as the digital board in the currently used MS2. A separate document, Enator TR:981002, describes how to install and connect the GMS into an existing MS2-cabinet.

The circuits of GMS have been selected for reliable function during varying climatic conditions and for many years of operation. The processor and software were selected and designed to manage high-capacity computing and to allow for simple upgrading.

A serial port is used for connection of external modem(s).

1.3 Definitions

BDM	Background Debug Mode is a special MCU-mode, enabling an external computer to take over the MCU. BDM is a very valuable tool for design and troubleshooting during software development.
DCE	Data Communications Equipment. Denominates the function of serial ports found on devices like modems, short-range modems, and radio modems. Connection is by straight cables to DET serial ports, and by crossing cables to other DCE-serial ports.
DTE	Data Terminal Equipment. Denominates the function of serial ports found on devices like personal computers, terminals, and printers. Connection is done by straight cables to DET serial ports, and by crossing cables to other DCE-serial ports.
EPROM	Eraseable Programmable Read Only Memory. Memory containing permanently saved data constantly available. This memory can be erased using UV-light. Used for starting the system and for application loading from the Flash-memory.
FAT	File Allocation Table. A file system used by Microsoft MS-DOS, Windows 95, and Windows NT. Saves file and catalogue names of eight plus three characters.
Flash-ROM	Memory for saving of programming code and test results. Can be erased and re-written a large, though finite, number of times. Memory contents are saved until the next erasing or re-writing operation. Does not require a continuous power supply to preserve contents.
GMS	General sensing station. Sensing board developed as a replacement for MS2 and quality enhancer of MS2000.
MCU	Micro Controller Unit. Denominates a microprocessor combined with additional functions. This is especially used in integrated applications, such as GMS.
MS2	Sensing station of Generation 2, popularly also called "The digital sensing station". MS2000 General sensing station based the hardware of MS3 and the operating system Windows NT.
MS3	Sensing station of Generation 3. This station is based on an industry-type PC and operating system MSDOS.
PIT	Periodic Interrupt Timer. A module of SIM, generating periodic interrupts at selectable intervals. PPI Programmable Peripheral Interface. A programmable circuit for digital I/O signals. RTC Real Time Clock. A clock circuit retaining date and time data

	even in case of power failure. RTX C Real-Time eXecutive in C. Real-time operating system used in GMS.
RTXCio	Real-Time eXecutive in C io system. Architecture for I/O-handling in RTX C. Specifies a general interface to external devices. Requires implementation for every single I/O-device involved.
RTXCfile	Real-Time eXecutive in C file system. File system for RTX C, using RTX Cio to handle secondary-memory devices.
SCI	Serial Communications Interface. A single UART implemented in SIM directly in the MCU. This communication channel is always available in a system based on MC68332 MCU.
SIM	System Integration Module. A block of various system modules in MCU, giving access to common functions in integrated systems. Simplifies hardware design and software development.
SRAM	Static Random Access Memory. Primary memory used in GMS. Saves programs and data during system operation. The contents will be lost in case of power failure.
TPU	Time Processor Unit. A programmable controller that is integrated in MCU. Independent of the CPU, TPU executes a microcode in a time-related multiprocessing state. For example, TPU can count pulses.
TPURAM	Time Processor Unit Random Access Memory. Static RAM integrated in the MCU. Can be backed up by a battery in case of power failure.
UART	Universal Asynchronous Receiver Transmitter. Circuit for simplified handling of asynchronous serial communication.
Watchdog	Denominates the hardware that monitors the operations and, after software trouble, will restart the system.

2. References

Avtal/Beställning MS2-GMS, Vägverket AL70A 97:12225 (Agreement/Order)
GMS-kretskortsanslutning till MS2, Enator TR:981002 (GMS Circuit board wiring to MS2)

MS2-GMS gränssnittsspecifikation, T0109-SGMS01-
DOC (GMS Interface specification)

Kretsschema MS2-GMS kretskort, TELUB-300198
(Circuit diagram)

MS2000 specifikation, Enator T0110-SM2K01-SDD (MS2000 specification)

MS3 Programsystembeskrivning, Enator T0026-SMS301-SDD MS3 Software system description).

3. System configuration

Configuration of GMS is described below. Hardware and software are dealt with in separate sections.

3.1 Hardware

GMS is a circuit board of size 6 HE = 233,4 x 220 mm. It is connected by two 64-pole connectors DIN 41612 Body style C. The board front has eight modular connectors for external devices and a modular connector for the service port. All signals to the MCU are accessible in four connectors on the GMS-board, for addition of another circuit board with more functions. The GMS-board is designed to replace the currently used MS2-board in the VViS sensing station, without any major modifications being required.

Memory access to all memory hardware (EPROM, SRAM, and Flash-ROM) is done using 16-bit access, to enhance performances of the GMS-system.

3.1.1 MCU

- Motorola MC68332 contains a 32-bit CPU-module (CPU32).
- System Integration Module (SIM).
- Time Processor Unit (TPU).
- Queued Serial Module (QSM).
- TPURAM 2 KB.
- Software Watchdog Timer (SWT)
- Periodic Interrupt Timer (PIT)

TPU does not use calendar date information (no millennium problems).

GMS is prepared for inclusion of a backup battery to supply TPURAM in case of power failure.

3.1.2 EPROM

- EPROM for program booting, 128 KB.

Storage for start-up routines only; not for drive routines or applications.

3.1.3 SRAM

- Static RAM 2 MB.

3.1.4 Flash-ROM

- The board is equipped with minimum 2 MB of Flash-ROM, optionally expandable to 4 MB.

3.1.5 Analog inputs

- The GMS-board has 14 analog inputs with filters. Connection is via connector P16 on the board. A/D-conversion is done at 12 bit resolution. Two multiplexers serve for A/D-channel selection.

3.1.6 Digital I/O

- The GMS-board has 14 opto-isolated digital inputs and 8 digital outputs. Connection is via connector P17 on the board.

3.1.7 Counter

- There are 6 opto-isolated inputs to TPU channels 0œ5 available in connector P17. The remaining counter channels are accessible from the MCU-bus.

3.1.8 RTC

- Real-time clock with battery backup. The RTC has two 16-register arrays, each with 4-bit operation. One array serves for reading/setting of real-time information. The other array handles alarm settings. Programming of alarm settings enables generation of regular interrupts. In this way GMS can be made to switch on/off itself regularly by turning on/off the power supply. This signal is accessible in connector P17.

S1	Seconds units
S10	Seconds tens
Mi1	Minutes units
Mi10	Minutes tens
H1	Hours units
H10	Hours tens
D1	Days units
D10	Days tens
Mo1	Months units
Mo10	Months tens
Y1	Years units

Y10	Years tens
W	Day of week
CtlD	Control reg. D
CtlE	Control reg. E
CtlF	Control reg. F

Since the year figures in the real-time clock consists only of the two least significant digits, no problems will occur when arriving at the millennium 2000. The real-time clock handles every fourth year as a leap year, so the year 2000 will be a leap year. However, the year 2100 will produce an error, since it will be considered a leap year as well. The years 98 and 99 will be presented as 1998 and 1999 relative to the superior system. All other years are presented as 20xx. Without software updating the clock of GMS will be functioning up to and including 2097-12-31 23.59.59 before the clock presentation returns to 1998-01-01 00.00.00.

3.1.9 Watchdog

- A programmable watchdog timer for board reset and system-reset.

3.1.10 Serial ports, external devices

- The GMS-board contains two UART-circuits, each with 4 serial ports, for connection to a modem or external sensors via RS232C-drivers. Connection is done using 8 board-front connectors (P2, P1, P4, P3, P8, P7, P10, P9). The connector type is Modular connector 8/8 (RJ45). The serial connector is of type DTE. The serial ports are numbered from 1 to 8, where port number 1 is at the top.

1	DSR
2	RTS
3	DCD
4	TXD
5	RXD
6	GND
7	CTS
8	DTR

The modular connectors have gold plated, nickel-based contact wires.

3.1.11 Service port

- This is 1 serial port with interface RS232C intended for service. The port is wired to the MCU-pins MOSI, RXD, TXD, and MISO via an RS232C driver. Connection is done using a board-front connector P6. The connector type is Modular connector 6/6. The serial connector is of type DTE.

1	RTS
2	GND
3	RXD
4	TXD
5	GND
6	CTS

The modular connector has gold plated, nickel-based contact wires. Signals RXD and TXD are also accessible in mother board connector P17.

3.1.12 BDM

- For communication with the processor during software development, the GMS-board has 2 connectors for connecting to an emulator board of a PC. The connector pins are directly wired to the corresponding MCU pins. The connectors consist of a two-line pin strip P13 (each line with five pins), and a stand-alone pin P14; this array enables use of the Motorola BDM.

1	DS*
2	BERR*
3	GND
4	BKPT*/DSCL
5	GND
6	FREEZE
7	RESET*
8	IFETCH*/DS1

9	VDD
10	IPIPE*/DS0

1	CPUCLK
---	--------

3.1.13 Battery

- For continuous supply of the real-time clock, the GMS is equipped with a lithium battery having a capacity of 170mAh. The battery voltage level is 3,0 V and its storage time is 10 years. The estimated operational time before battery change is 2 years. Load to the battery is applied only when the board is not powered by its ordinary supply.

3.2 Software

The software of GMS has three main parts: a Boot-ROM containing initiating routines and functions for file transfer; a core consisting of drive routines complying with the RTXcio-model together with the file system RTXfile; and an application consisting of one or more processes. The core and application are saved as one entity in the secondary storage.

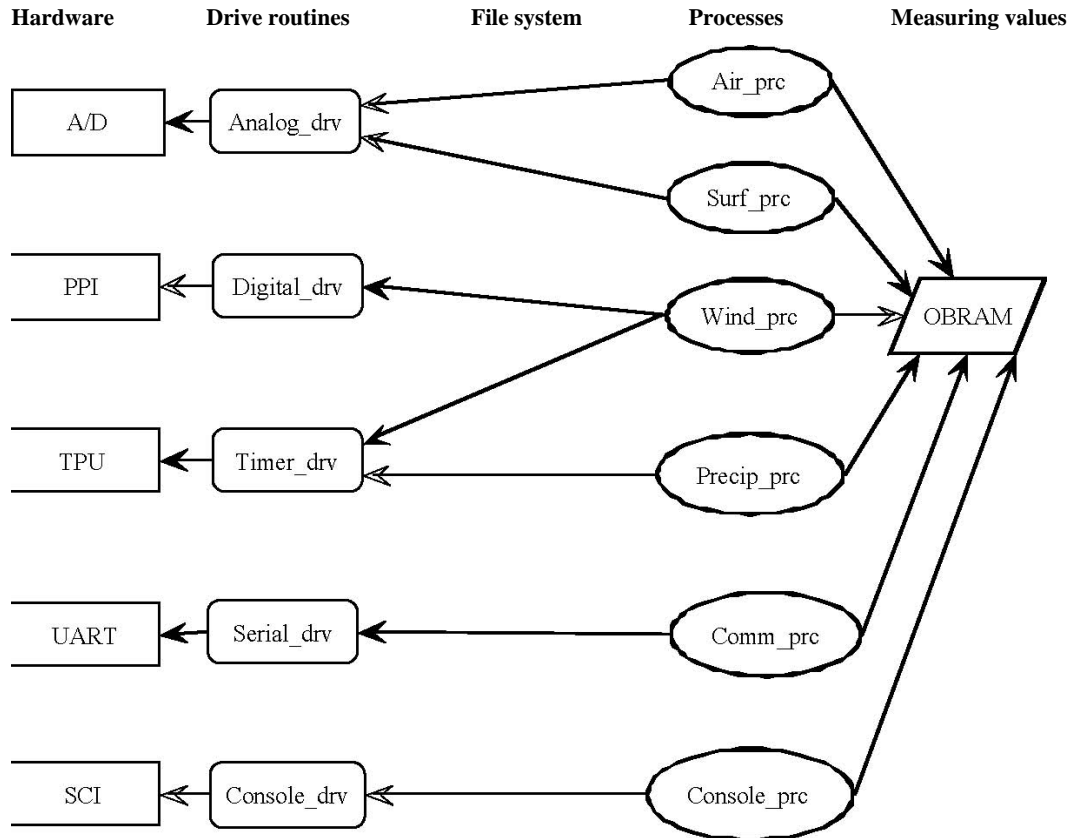
The GMS-application is divided into a processor that handles related sensors and pertinent measuring values derived from those sensors.

3.2.1 Overview

The GMS-application is divided into a measuring function and a control function. They are described below together with design considerations.

3.2.1.1 Measuring function

The figure below illustrates the measuring function.



Measuring function, overview

The symbols represent various system parts.

- Squares represent I/O-circuits.
- Squares with rounded corners represent drive routines according to the RTXCio-model. Each process consists of an interrupt routine and a process.
- The parallelogram represents a storage volume.
- Ellipses represent processes.

Arrows represent calling sequences within the system.

The measuring processes (Air_prc, Surf_prc, Wind_prc, and Precip_prc) make calls to drive-routine functions for reading of measuring values, date, and time.

The measuring processes send a message to continue operations to the monitoring process. Measuring values provided with time stamps are saved in TPURAM, where individual access by various processes is enabled.

The communication process (Comm_prc) communicates with the superior system. It calls drive-routine functions to handle serial ports, the secondary storage, and the real-time clock. A request for measuring values brings about reading of saved measuring values from TPURAM, which are sent to the superior system. Files can be down-loaded to the secondary storage. Updating of the core and/or application is done by down-loading a new version into the board. The communication process sends a message to the monitoring process to continue operations.

- Modem is connected to Port 1
- Fixed connections are entered on Port 2

The console process (Console_prc) permits access to the system via the service port. This process calls drive-routine functions to handle the service port serial), the secondary storage, and the real-time clock. Connections are established using terminal programs, where the console process offers access to functions for investigations and trouble-shooting. Every application must contain a console process.

3.2.1.2 Design selections

A software design capable of grouping various measuring values using only a few processes was designed to satisfy the following wants:

- 1 Employ the real-time operating system RTX for process control, process communication, and access control.
- 2 Increase the flexibility to simplify introduction of future applications, such as frost-depth sensing, freezing-point sensing, vehicle counting devices, cameras, air-pressure sensing, etc. In addition, a high flexibility makes it easier to cut down the functionality to exclude disturbances like false error reports and measurement values entering without any sensors connected.
- 3 Reuse the MS3 source code as much as possible, with respect to the above requirements 1 and 2.
- 4 Preserve the same reliability level and options of remote-operated software updating as used in MS3.

The proposed design satisfies the requirements by reducing the complexity (number of processes) while simplifying addition of new measuring processes.

3.2.1.3 Information flow, example

Example: For air humidity the flow of control signals and data is as follows.

- 1 The timer of the processor issues an initiating signal for measuring the analog input channel where the air-humidity sensor is connected.
- 2 Drive routine Analog_drv awakens, reads the analog channel several times, and saves an average value of the values obtained to increase reliability.
- 3 If the station is configured to measure air humidity, the process Air_prc will call drive routine Analog_drv. It then reads the values saved for the analog channels where sensors for air temperature and air humidity are connected.
- 4 Read measuring values are processed in the same way as in MS3 and are integrated over a time of 30 seconds.
- 5 Air_prc saves the air temperature, air humidity, and dew point in a battery-powered memory of the processor, together with date and time information.
- 6 Air_prc issues signal Monitor_prc, thus telling it is in continuous operation.
- 7 Process Comm_prc answers a phone call via a modem, reads the three measuring values from the memory and sends them to a superior system.
- 8 Comm_prc issues a signal Monitor_prc, thus reporting it is in operation.

The other measuring processes in the diagram function in a similar way, the differences being the sensors involved and processing of the values read. Control signals and data have flows according the pattern discussed above.

3.2.1.4 Pros and cons

Obvious advantages of the design are these facts:

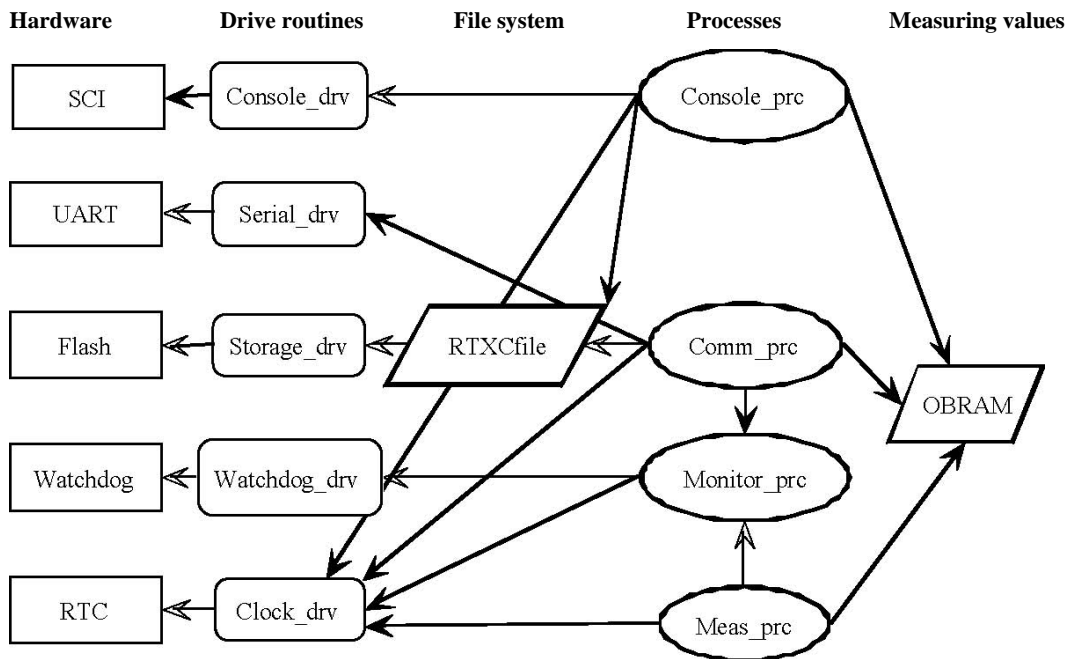
- Closely measured/derived values are handled within a single process.
- The number of measuring processes is low.
- Retrieval of measuring values is separated from the communication.
- New functionality can be added or removed in a simple way, without affecting existing/remaining measuring processes.

A drawback might be the fact that:

- The software has a different design than that of MS3, where retrieval takes place in parallel and simultaneously from all sensors.

All in all, the general assessment is that the advantages overrule any drawback.

3.2.1.5 Control function



Control function, overview

Symbols have the same meanings as in the diagram of measuring functions.

- Meas_prc i represents the four measuring-value processes (Air_prc, Precip_prc, Surf_prc, and Wind_prc), all using the same control functions.

Arrows represent calling sequences within the system.

The monitoring process (Monitor_prc) checks the system function. All processes have to signal to the monitoring process during operation. When a process no longer is signalling regularly, a count-down is started. After the time allowance for process is elapsed, the system will restart. By operating in this

manner the system will not stop by operational disturbances. Applications where operational reliability is very crucial must include a monitoring process.

3.2.2 Boot-ROM

Boot-ROM contains routines for initiating MCU, SIM, SCI, PIT, SWT, and TPU according to the hardware configuration, for preparing system operation,

for handling of system errors and system interrupts, and file transfer functions. The core uses specific interrupt vectors, but the others are handled by general routines in Boot-ROM. The RAM, TPURAM, and Flash-ROM are configured for operation.

By start-up a check is made whether the user wants to interrupt the start-up and instead hand over the system control to a single console. That console enables down-loading of files to the secondary storage on the board, thus providing a common update of drive routines and measuring-board application. Also, the console offers optional trouble-shooting by investigation and allocation of the memory contents to various memory addresses.

If the user does not intervene the signalling, or when operation of the single console is ended, the core and application are loaded into the RAM from the secondary storage. System control is handed over to the application.

If hardware memory configuration is changed (sizes of ROM, SRAM, or Flash-ROM), the Boot-ROM (hard-coded information) must be replaced.

3.2.3 Core

Below every drive routine is described, referring to hardware and function.

Drive routine	I/O-circuit used	Description
Analog_drv	A/D	Reads the analog inputs regularly, computes an average value out of several value readings, and saves it for the application. Uses PIT as the start signal for measuring.
Digital_drv	PPI	Reads the digital inputs regularly and saves the value for the application. Uses PIT as the start signal for measuring.
Clock_drv	RTC	Handles date and time information saved in the real-time clock circuit. Date can be written into the circuit and be read from it. Extra signals wired to the connectors are NOT used.

Console_drv	SCI	Handles interrupts from the serial communication circuit integrated into MCU. Reads and sends off characters from/to the serial port wired to MCU. The drive routine saves received/request-to-send characters internally on request by the application.
RTXCfile	Storage_drv	A file system compatible with MS-DOS FAT. Uses storage_drv for handling of the secondary storage. Functions from RTXCfile are compiled together with those processes that save data into the secondary storage (serving as a hard disk).
Serial_drv	UART	Handles interrupts from the external serial communication circuits. Reads and sends off characters from/to the serial ports of the board. The drive routine saves received/request-to-send characters internally on request by the application.
Storage_drv	Flash-ROM	Handles read/write operations for the secondary storage. Reading and writing are operated as blocks. Using RTXCfile requires that the storage is capable of block handling.
Timer_drv	TPU	Handles interrupts from the timer circuit, reads and saves counter values for the application.
Watchdog_drv	SWT	Handles start, stop, and updating of the watchdog integrated into the MCU.

3.2.4 Application

Below every process is described, referring to resource needs and function.

Process	Resources needed	Description
Air_prc	analog_drv, clock_drv, Monitor_prc, TPURAM	Reads values of air temperature and air humidity from the same analog inputs as MS2. Reads date and time from the real-time clock. Computes dew point. Processes these values like MS3 and saves time-stamped values of air temperature, air humidity, and dew point in TPURAM. Issues signal Monitor_prc for continued operation.
Comm_prc	clock_drv, Monitor_prc, TPURAM, storage_drv, serial_drv	Communicates with superior system via the serial ports. Connection enabled by modem or direct connection via a service cable. Measuring values are read from TPURAM. New versions of drive routines and application are received and saved in the secondary storage. The real-time clock serves for time-stamping of data and files, and for date comparisons. Issues signal Monitor_prc for continued operation.
Console_prc	console_drv, clock_drv, TPURAM, RTXCfile, storage_drv	Handles user's inputs/outputs via the service port. The communication is run by a terminal program using a command-line-based interface giving access to functions for process control, memory investigation, and trouble-shooting. The real-time clock can be read and adjusted.
Monitor_prc	watchdog_drv	Checks functions of the other system processes. When a process stops, a process-specific countdown starts. After time allowance is elapsed, the system restarts.

3.2.5 Software updates

Precip_prc	clock_drv, Monitor_prc, TPURAM, timer_drv	Counts precipitation pulses from the same input as used by MS2. Reads date and time information from real-time clock. Integrates values like MS3. Computes drop sizes. Saves latest, time-stamped intensity and size values in TPURAM. Issues signal Monitor_prc for continued operation.
Surf_prc	analog_drv, clock_drv, Monitor_prc, TPURAM	Reads surface temperatures and any remote surface temperatures from the same analog inputs as MS2. Reads date and time information from the real-time clock. Processes those values like MS3. Saves latest, time-stamped surface temperatures values in TPURAM. Issues signal Monitor_prc for continued operation.
Wind_prc	clock_drv, digital_drv, Monitor_prc, TPURAM, timer_drv	Reads wind direction from the same digital inputs and wind speed from the same counter input as MS2. Reads date and time information from the real-time clock. Computes average values of wind direction and speed, and maximum speed like MS3. Saves latest, time-stamped wind-speed and wind-direction values in TPURAM. Issues signal Monitor_prc for continued operation.

The software stored in Flash-ROM on GMS can be updated in two ways: 1) Via the service port, or 2) Via Serial Port 1 and Serial Port 2 during operation. The two methods are described in more detail below.

Both updating methods can be operated automatically by means of specially designed software products.

For more information of the communication interface towards GMS, refer to the MS2-GMS interface specification.

3.2.5.1 Service port

Updating of the software for the Flash-ROM of the board is ensured through the option of interrupting the start-up procedure.

To use this option, connect a computer via a crossing cable to the service port of GMS, then start a communication program capable of sending a file with the file-transfer protocol X-modem.

Start GMS by first connecting it to power, then pushing the reset button. After that push repeatedly on the computer's Return key until a prompt is displayed in the window for the communication program. Type the command `FILERX` and push the Return key. Now start the file transfer of the software, selecting X-modem as the file-transfer protocol, and select a file of the new software. Start up-loading of the software.

When up-loading is finished and the software is stored in the Flash-ROM of the board, type command `RESET` and push the Return key or the restart button. Now MS2-GSM restarts and executes the new software version.

This method can be used for updates only when the service port is available.

3.2.5.2 Remote updates

The application shall include an option enabling remote-operated updates.

Establish contact with GMS in the same way as by an ordinary collection of measurement values. Use a communication program capable of sending a file with the file-transfer protocol X-modem.

Log-on the station (to on-line mode). When the prompt appears, type the command `FILERX` and push the Return key. Then start the file transfer of the software and select a file of the new software. Start up-loading of the software.

When up-loading is finished and the software is stored in the Flash-ROM of the board, type command `RESET` and push the Return key. Now MS2-GSM restarts, switches to off-line mode, and executes the new software version.

This method can be used if the software version used responds to the contacts attempted, thus enabling a proper log-on. Additionally, the new software version must respond to contacting attempts to enable its remote-operated updating (in the future).

3.2.6 Development environment

- Language: 68000 assembler, C++
- Target operating system: RTXC 3.2c together with med RTXCio 1.3 and RTXCfile 1.3b
- Development environment: Microtec ASM68K assembler 7.1J, Microtec CCC68K C++-compiler 2.0k and Microtec MCC68K C-compiler 4.5r
- BDM-debugger: Nohau Emul 16/300 2.3m
- Evaluation board: MCT Paul & Scherer Scotty332 with the same Flash-ROM and RTC as GMS.
- Development operating system: Windows NT 4.0 Workstation