PROFIBUS-DP to G-64 CONFIGURABLE INTERFACE

E. Carlier, A. Moreno Forrellad# (SL/BT), J. Rochez (IT/CO) and J. Serrano (PS/CO)

Abstract

Since many years, the SL/BT group has developed specific G-64 [1] hardware modules for the control of the fast pulsed magnets. Up to now, the integration of these modules into the accelerator control system has been done using a MIL1553 fieldbus in command/response mode connected to a LynxOS front-end.

In 1996 and 1998, recommendations for the use of industrial fieldbuses [2] and programmable logic controllers [3] were issued at CERN. In order to integrate the existing G-64 hardware into this environment based on industrial components, an interface between the G-64 bus and the PROFIBUS-DP fieldbus [4] has been developed.

This interface is based on a modular and open hardware architecture with on one side an off-the-shelf industrial, fieldbus independent, interface and on the other side a FPGA (Field Programmable Gate Array). Required functionality has been implemented within the FPGA with VHDL (Very high-speed integrated circuit Hardware Description Language).

# On leave, Technical Student from 09/1999 to 12/2000, now at Motorola Switzerland, SPS/WSSG/WCD Group.

Geneva, Switzerland
6 April, 2001
PROFIBUS-DP to G-64 CONFIGURABLE INTERFACE

E. Carlier, A. Moreno Forrellad# (SL/BT), J. Rochez (IT/CO) and J. Serrano (PS/CO)
CERN, Geneva, Switzerland

Abstract
Since many years, the SL/BT group has developed specific G-64 [1] hardware modules for the control of the fast pulsed magnets. Up to now, the integration of these modules into the accelerator control system has been done using a MIL1553 fieldbus in command/response mode connected to a LynxOS front-end.

In 1996 and 1998, recommendations for the use of industrial fieldbuses [2] and programmable logic controllers [3] were issued at CERN. In order to integrate the existing G-64 hardware into this environment based on industrial components, an interface between the G-64 bus and the PROFIBUS-DP fieldbus [4] has been developed.

This interface is based on a modular and open hardware architecture with on one side an off-the-shelf industrial, fieldbus independent, interface and on the other side a FPGA (Field Programmable Gate Array). Required functionality has been implemented within the FPGA with VHDL (Very high-speed integrated circuit Hardware Description Language).

1 INTRODUCTION

1.1 General
The fast controls of the “SPS fast pulsed magnets” requires complex timing and pulsed signal acquisition systems. As hardware modules, like a synchronous timing delay with nanosecond precision or a fast track-and-hold optically isolated 12 bit ADC, are not available directly from industry, specific modules based on the G-64 standard microprocessor bus have been developed. The slow controls used commercially available G-64 modules.

The emergence of industrial standard components in the area of fieldbuses and PLCs (programmable logic controllers) make them advantageous to use for the modernisation of the slow controls. This conversion has been started last year using SIEMENS S7-300 and S7-400 PLCs, which have been connected through a PROFIBUS-DP fieldbus to a SCADA (Supervisory Control and Data Acquisition) system.

In this framework, it appeared to be useful to have the possibility to interconnect the existing specific fast electronics to this industrial environment. As no solution was available from industry for the connection of the G-64 bus, a specific hardware module, the “PROFIBUS-DP to G-64 Configurable Interface”, has been developed.

1.2 Interface
The interface controls continuously the G-64 bus. It maps the G-64 registers located in the VPA (Valid Peripheral Address) field in different manners on the PROFIBUS-DP message frame.

A modular approach has been chosen for both the hardware and the software architecture with the aim to avoid in the future a strong dependence on the interface, and to obtain an easy way to connect specific hardware to the upper layers of control system.

The hardware is composed of an OEM (Original Equipment Manufacturer) PROFIBUS-DP communication module from HMS Industrial Networks AB [5] and a XILINX FPGA for the control of the G-64 bus. The required software tasks are provided through an application, based on state machines, that has been developed from a set of structured functions written in VHDL.

As the HMS module is not “fieldbus specific” it is possible to port the interface to other fieldbuses. Its functionality can also easily be adapted to other specific user requirements.

2 TECHNICAL SOLUTION

2.1 Hardware
Figure 1 shows the hardware (architecture) divided into three blocks:

- An AnyBus-S fieldbus interface mezzanine board;
- A XILINX FPGA and its configuration ports (PROM and parallel cable);
- Line drivers for the G-64 data, address and control lines.

The AnyBus-S board has the PROFIBUS-DP fieldbus protocol implemented and exchanges data with the FPGA through a DPRAM (dual-port static RAM) of 2Kx8.

The FPGA application is uploaded at power-on, either from an on-board PROM, or directly from a computer through a parallel interface. After the initialisation of the AnyBus-S board and the configuration of the interface, the FPGA bridges in both directions the G-64 VPA field to the Anybus-S DPRAM with a synchronous read/write data access mode.

# On leave, Technical Student from 09/1999 to 12/2000, now at Motorola Switzerland, SPS/WSSG/WCD Group.
2.1.1 Fieldbus interface

The first part of the hardware is based on the AnyBus-S PROFIBUS-DP slave module. This module can immediately be connected to a PROFIBUS-DP segment as a slave partner and allows the bus master to read and write data directly from/to its internal DPRAM. Any memory location in the DPRAM can be accessed from both sides simultaneously.

The module has the PROFIBUS-DP communication protocol fully implemented in accordance with the EN 50170 norm. Its specific technical features are the following:

- Parallel or serial access to the process data in the AnyBus DPRAM memory
- Maximum cyclic I/O data size: 208 bytes “in”, 208 bytes “out”, 416 bytes in total
- RS485 optically isolated PROFIBUS interface with an on board DC/DC converter
- Auto baud-rate detection, baud-rate range: 9,6 kbit-12 Mbit

The AnyBus-S product family offers also several DPRAM compatible modules for other fieldbusses. By changing only this AnyBus-S board on the interface, a bridge to the G-64 bus via different fieldbusses is possible without any modification to the application implemented in the FPGA. This option results in an open solution, not only for PROFIBUS-DP fieldbus, but also for the CANopen fieldbus recommended at CERN. And not to forget, the possibility to create a bridge between Ethernet IP and G-64 bus.

2.1.2 FPGA

All the intelligence of the interface has been implemented inside the FPGA chip. With this flexible solution, mapping of the G-64 registers, size of data exchange on the PROFIBUS-DP segment, and other functional parameters, it is possible to adapt the application easily to the requirements.

During normal operation, the AnyBus-S configuration is uploaded at power-on from the PROM or the computer and is fixed. However, the configuration of the G-64 can be integrated into the FPGA in case of a fixed configuration or be modified on-line during normal execution.

2.2.3 G-64 interface

Special tri-state line drivers have been mounted on the board that matches the electrical specification of the G-64 standard with the FPGA logical output levels.

Bi-directional drivers have been used for data lines (D0 to D7). The address (A0 to A9), the enable clock (ENB), the valid peripheral address (VPA), the read/write (R/W) and data transfer acknowledge (DTACK) control lines are directly controlled by the FPGA for synchronous mode data access. G-64 hardware interrupts are not handled by the interface.

2.2 Software

The VHDL programming language, recommended at CERN, has been used to describe digitally the functional aspects of the models and the timing behaviour needed for the implementation of the communication between the configurable G-64 hardware that has to be accessed and the message format for PROFIBUS-DP.

Two models have been developed and implemented in the application program:

- Fixed memory mapping;
- Dynamic memory mapping.

During development and coding, the ESA (European Space Agency) VHDL modelling guidelines [6] have been followed in order to ensure a high quality of the VHDL models. The models can thus efficiently be used, maintained with a low effort throughout the full life cycle of the modelled hardware, and be ported between simulators.

The software development environment for code production has been based on three different CERN wide supported software tools:

- “Leapfrog” [7] for compilation and simulation;
- “Synplify 522a” [8] for code synthesis;

In order to avoid future memory limitations, to maintain a high level of performance, and to satisfy speed constraints, a XILINX Virtex XCV-200PQ240 FPGA with a XC18V02-PC44 socket mounted PROM has been selected for this application. Table 1 shows the specific technical features of the FPGA.

<table>
<thead>
<tr>
<th>System gates</th>
<th>236 666</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configurable logic blocks array</td>
<td>28 x 42</td>
</tr>
<tr>
<td>Logic cells</td>
<td>5292</td>
</tr>
<tr>
<td>Maximum available I/O</td>
<td>166</td>
</tr>
<tr>
<td>Block RAM bits</td>
<td>57 344</td>
</tr>
</tbody>
</table>

In addition, the use of a FPGA instead of discrete components offers the possibility to program small control algorithms inside the FPGA without modifying the hardware.

Table 1: FPGA parameters

In order to avoid future memory limitations, to maintain a high level of performance, and to satisfy speed constraints, a XILINX Virtex XCV-200PQ240 FPGA with a XC18V02-PC44 socket mounted PROM has been selected for this application. Table 1 shows the specific technical features of the FPGA.

- Parallel Cable
- J-Tag
- PROM
- XC18V02-PC44
- Drivers
- XILINX Virtex XCV200 PQ240
- AnyBus-S PROFIBUS-DP
2.2.1 Architecture

The basic structure of the software consists of the concurrent execution of different processes declared as a mutually exclusive state machine like shown in Figure 2. Processes are based on a set of low-level hierarchical functions and a structured data flow. They can be classified into three categories:

- System management (Anybus-S initialisation and G-64 configuration);
- Communication (read, write, single data read and single data write);
- Monitoring (mirroring and internal parameters control).

![Software Architecture Diagram](image)

Figure 2: Software Architecture

2.2.2 Models

2.2.2.1 Fixed memory mapping

The “Fixed memory mapping” model consists of a direct mapping of a predetermined number of G-64 8 bit registers to the PROFIBUS-DP message frame.

Operational configuration parameters, like the G-64 register addresses and the PROFIBUS-DP message configuration, are defined during the FPGA development and are embedded inside the application program after compilation and routing. Once the program has been started it is not possible to change or modify these parameters. This model is useful to work with very specific systems.

2.2.2.2 Dynamic memory mapping

The “Dynamic memory mapping” model allows to link dynamically the DPRAM addressing map, i.e. the translation of the PROFIBUS-DP message frame to the G-64 address range.

The G-64 configuration process fills a configuration table that contains a software description of the G-64 hardware. This description contains the number of modules present on the bus, their base address and their data size. This configuration table can be loaded after power-on and can remotely be modified at runtime. This model offers a flexible, hardware independent solution.

2.2.3 Algorithm

After power-on, the AnyBus-S module is initialised with operational settings depending on the selected model. Initialisation settings are chosen during modelling of the application and cannot be changed at runtime. The Anybus-S initialisation process configures the module in order to match the PROFIBUS-DP message size with DPRAM addressing range. Once the module is initialised the PROFIBUS-DP communication is enabled.

Two handshake registers, control and status bytes, are then used to manage the communication synchronisation between the two buses. The application program that is continuously blocked in a one-millisecond timeout wait state controls the handshaking procedure. After reception of an interrupt from the AnyBus-S module or expiration of its internal timeout, the application program reads the control byte and triggers the execution of the process that has to be executed.

Data exchange between the two buses can be done with two different types of processes:

- Read/write process,
- Single data read/write process.

Depending on the selected model, the read process reads cyclically all the data from the G-64 crates in accordance with the loaded configuration. Respectively, the write process writes once the data to the corresponding G-64 address after reception of the related data exchange request. As the mapping of the addresses between the two buses is preloaded, no G-64 addresses have to be specified anymore and the G-64 registers can be visible as deported I/O on the PROFIBUS-DP segment.

Single data read/write processes allow to read from or to write to the bus master one data byte directly at a specific G-64 address. In this case, the G-64 address to be accessed has to be specified at each data exchange and has to be included inside the PROFIBUS-DP message frame.

Additional processes are implemented that provide remote monitoring facilities of the application program such as mirror or internal parameters processes.

The mirror process permits to map through the FPGA internal register the DPRAM output area on its input area. This process is realised cyclically and offers the possibility to control the PROFIBUS-DP frame. It is mainly used for testing. The internal parameters process show some internal AnyBus-S registers and some details
of the FPGA software such as software version, internal status, and watchdog status.

Finally, a watchdog that avoids a blocking or unknown state is implemented into the software. An internal one-second counter is reset at each state transition and, in case of expiration, an automatic re-initialisation of the program is issued.

3 APPLICATION

3.1 AnyBus-S module evaluation

The correctness of the low-level hierarchical functions has been demonstrated with an HMS development kit for the AnyBus-S module.

This kit, installed in a PC, has been used to carry out a manual initialisation of the board and then to execute a mirrored function of the whole input PROFIBUS-DP data frame to the output PROFIBUS-DP data buffer.

A second PC has been configured with an APPLICOM [10] interface board as PROFIBUS-DP segment master. Two BridgeView [11] applications were communicating with the APPLICOM board, one acting as a SCADA process sending information to the AnyBus-S module and the other one acting as the FPGA chip on every bounced data buffer.

3.2 Interface integration

The PROFIBUS-DP to G-64 interface has been integrated and tested in different standard industrial architectures.

Figure 3: G-64 to PROFIBUS-DP interface showing the AnyBus-S mezzanine board (left), the FPGA and its programming PROM (centre) and the G-64 drivers (right).

In the first application, the G-64 hardware has been accessed by a SIEMENS S7-300 PLC acting as fieldbus master. In this case, a fixed memory-mapping model has been used at the interface level, which allows direct access to the G-64 registers through the internal DBs (data block) of the CPU.

In the second application, the G-64 hardware has been connected through the fieldbus directly to a SCADA system. In this case, the flexible G-64 configuration mechanism has been used through the interface dynamic memory-mapping model. The SCADA system maps its internal variable to the different G-64 registers and manages the G-64 hardware configuration.

4 CONCLUSIONS

The objective, the interconnection of the PROFIBUS-DP fieldbus to the G-64 microprocessor parallel bus, has been fulfilled.

The choice to realise an interface board by the integration of an OEM module has resulted in a reduction of the development time and the overall project costs. The combination of this interface with the modelling of the functional requirements inside a FPGA has resulted in an open and modular solution.

As the G-64 address area is larger than the single message transmission capabilities of the PROFIBUS-DP protocol, a direct data mapping of the two buses is not possible. For this reason, data exchange mechanisms based either on a dynamic configuration of the interface or on an indirect access mode have been implemented and provide a flexible solution.

Finally, the integration as deported I/O of our specific G-64 hardware has been successfully realised inside a standard industrial control architecture.

5 ACKNOWLEDGMENTS

Many thanks to all the people who have contributed to this project: to Serge Brobecker (IT/CE) for his frequent assistance with the software development environment, to Virgina Prieto (PS/BD) and Jean-Louis Bretin (SL/BT) for their advices during the design and the test of the electronic circuit, and specially to Rosa Maria Cantí from Polytechnic University of Catalonia who has supported the project since its beginning.

6 REFERENCES