AUTOMATIC GENERATION OF INTELLIGENT INSTRUMENTS FOR IEEE 1451

L.Cámara, O.Ruiz, A.Herms, J.Samitier, J.Bosch.

Departament d'Electrònica. Universitat de Barcelona. Martí i Franquès 1, 08028 Barcelona, Spain
Tel: +34 934029070 Fax: +34 934021148 e-mail: lourdes@el.ub.es

Abstract - *In this work we present the design of a smart transducers interface module (STIM) that meets the IEEE 1451 normative. The improvement of this prototype is that it can work with any configuration of transducers without the necessity of changing the software and hardware of the STIM. The system automatically adapts its functions from the TEDS (Transducers Electronic Data Sheets). We only need to write the appropriate TEDS of our transducers configuration in a non-volatile memory of the STIM. We also have developed a PC TEDS editor that allows us to introduce them in a friendly way in a PC and save them into the STIM through the parallel port.*

Keywords - Intelligent control; Smart transducer interface; Distributed measurement and control; Plug and play; Network protocols.

1. INTRODUCTION

Nowadays the transducers available in the market have many different ways of producing their outputs (4-20mA, frequency output, voltage output, etc.). Because of this diversity the development of instrumentation systems that involve transducers with different types of interface becomes complex, expensive and laborious. This means a drawback for designers, transducer manufacturers and final users. The development of the standard IEEE-1451 [1,2] intends to normalize and simplify the interconnection between transducer systems using networks, such as Ethernet [3] or CAN [4]. This normative gives very useful guidelines to make interconnection of transducers to instruments and networks easy and standard [5,6]. The standard is divided in two parts:

- One part that describes the STIM (Smart Transducer Interface Module), giving rules for organising the transducers in channels, the definition of a standard connector (signals and timings) and the TEDS (Transducer Electronic Data Sheets). This last item is very important because they include complete information about the transducers and allow a lot of interesting features to be introduced, such as Plug-and-Play [7] and hot start-up connection of nodes in the network.
- And another part that defines the NCAP (Network Capable Application Processor), the element that connects the STIM to the specific network used in each system. This element reads the TEDS from the STIM and uses them to notify the system about the available resources and to manipulate data from/to the STIM.

Big companies in the electronics industry are already working in the development and commercialization of STIMs [8] and NCAPs [9] mainly for Ethernet based networks. Being this implication an important leap forward for the IEEE 1451 standard, the STIM still is an element designed and produced for each specific application to be considered.

2. STARTING POINT

The work presented is the consequence of the evolution of our investigations, during the two last years, on the development of complete nodes according to the IEEE 1451 standard. Each node includes two subsystems, a STIM and an NCAP.

The NCAP has been specifically designed for a CAN network.

The first version of the STIM was generated by strictly following the philosophy of the IEEE 1451 standard. This first STIM was designed for a system composed by a temperature sensor and an actuator formed by 6 LEDs, and was developed using and FPGA as a controller. After this first prototype we made a new design using a small microcontroller instead the FPGA. This alternative is much more flexible and implies a smaller development time.

Despite the flexibility of using a microcontroller to implement the STIM, it still remains a problem in the development of real systems. The software of the microcontroller must be specifically designed for each application, depending on the transducers used in each node. Moreover, the hardware may be different depending on these transducers.

3. AUTOMATIC GENERATION OF STIMS

From this starting point we established a new goal that represents an important improvement. The idea is simple: automatically generate the STIM from the specific system TEDS. In other words, we have designed and implemented a specific hardware and, with the information extracted from the transducers TEDS, our tools develop the STIM software that suits the specific configuration of transducers in each application.

3.1 *Controller*.

The STIM controller is based in a small Analog Devices data acquisition system integrated in a single chip. The most remarkable feature of this chip is that it includes a microcontroller core (8052), a precision 12-bit multiplexed (8 channels) ADC (Analog to Digital Converter) with

ISBN 972-98115-4-7

autocalibration and temperature sensor, and a 2 DAC (Digital to Analog Converter). There are three blocks of memory in the chip: 8 Kbytes FLASH ROM for program, 640 bytes FLASH for data and 256 bytes RAM also for data. Moreover the chip has an external bus that allows the expansion of memory up to 16 Mbytes (this will be helpful for big TEDS). One of the µC (microcontroller) ports is used to implement the standard digital connector to the NCAP. As can be seen in Fig. 1, the complete STIM control circuitry is very simple and is made with only the ADuC812, a 512 Kbytes NVRAM memory, and registers to demultiplex the address/data bus of the ADuC812.

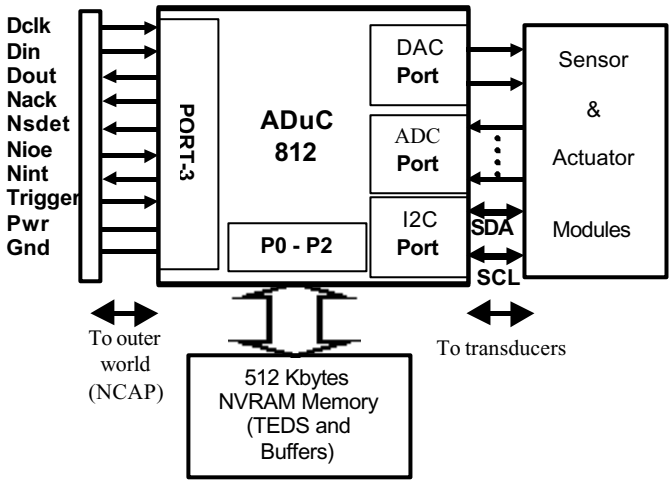

Fig. 1 - Block diagram of the STIM.

Other facilities of the ADuC812 are interrupt dealing and integrated timers that allow the generation of basetimes to implement some characteristics of the normative such as channel measurement repetitions or sequences generation.

3.2 *Memory Organization.*

One of the crucial points of the design is the memory organization. That is, which kind of data we need to store and how to distribute them between the internal µC memory and the NVRAM. We have to take in mind that the software of the µC will be the same independently of the specific transducers. The only thing that controls the STIM functions are the TEDS, so in the memory system we will store information that the program on the µC will use, together with the TEDS information.

The µC memory is organised as follows:

- 8 Kbytes FLASH: The program that controls the system. We had to make the software in assembler in order to minimize code size, because we wanted to use exclusively the internal memory for code.
- 640 bytes FLASH: Information about the **type** of each channel (up to 255 channels) and also the **size** of the corresponding buffer in the external NVRAM. It is organized in two sections, and it is permanent once the system has been configured. See Table I.
- 256 bytes RAM: For stack and the necessary program data variables.

SECTION	INFORMATION	PAGES
A	Number of channels in the system (channel 0) Number of bytes in the buffer for each cannel.	0-128F
B	Type of channel	129-160
Each page is 4 bytes long		

Table I – Organisation of the internal FLASH

The organization of the NVRAM gets much more complicated when the system has to offer good performances. The data that reside in the NVRAM memory are grouped in seven sections. In Table II we show these sections.

SECTION	INFORMATION	ADDRESS
A	Status and Mask registers for each channel. Defined by the normative.	0000-07FF
B	One Control register for each channel.	0800-08FF
C	Module address and address in module for each channel: Two bytes containing I2C address for each channel.	0900-0AFF
D	Index of addresses for buffers and TEDS blocks of each channel. Four bytes each.	0B00-22FF
E	Buffer counters. Two bytes per channel to know the number of bytes already stored in each buffer.	2300-24FF
F	Buffers.	002500-03FFFF
G	TEDS	040000-FFFFFF

Table II – Organisation of the NVRAM

3.3 *Sensor and Actuator Modules*

Some additional modules have been developed to allow the connection of up to 255 channels of any type in order to make the system the most flexible and versatile within the standard. The channels are connected through different stackable modules, depending on the type of channel. Three different modules have been implemented:

1. Analog inputs, up to 32 channels per module.
2. Analog outputs, up to 32 channels per module.
3. Digital channels (Input and Output), 16 lines per module.

The first two types of modules are basically analog multiplexers that redirect the input/output signals of the transceivers to the analog input/output lines of the ADuC, As an example, in Fig. 2 we show the block diagram of the Analog Output Module.

In the case of digital channels, the module is based in a small and economical µC. The aim of this organization, in the digital case, is to provide a method to define channels grouping lines of input/output, and to release from this work the ADuC.

The communication between the STIM controller and the modules are made via an I^2C bus. This communication consists of the selection of the channel to be measured or

activated. In the case of digital channels, in addition to this, the information of the transducers is also transmitted via I^2C.

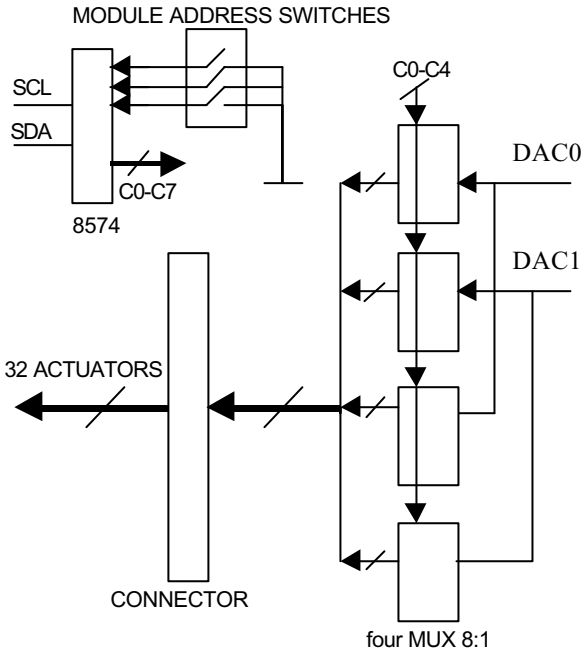

Fig. 2 - Block diagram of an analog actuator module.

This solution optimizes the utilization of the μC ports, and simplifies the hardware connections.

We can make any combination of these modules, stacking them over a STIM controller. The are only two hardware restrictions:

1. No more than 255 channels (normative restriction).
2. The buffer size needed by the transducers can not exceed the total buffer size available in the NVRAM.

4. STIM SOFTWARE

The basic idea of our design is that the STIM controller adapts its work according to the TEDS information. To fit this in an operational way (velocity and simplicity), we need to include additional information in the system. This information is organized in tables and stored in the FLASH data memory of the ADuC and in the external NVRAM. The information of these tables are mostly indexed to point to the different interesting data of each specific channel: TEDS, information of the channel, buffer size and location... So, the ADuC software uses these tables basically to access in a simple and fast way to the TEDS information (also saved in the NVRAM) and to access to the buffers during the measurement and read process.

We have developed different procedures, in assembler, in order to deal with all the possible kind of sensor and actuator channels defined in the normative. The program can be divided in three parts. The initial one is a first-time configuration routine, which stores the basic data from TEDS before the normal working of the STIM. After that, there is a waiting loop that works in parallel with a periodical interruption routine. This routine scans all the channels defined in the TEDS to perform the periodical actions required in each case. And finally, there are two additional blocks of routines, also called via interruption, to perform the answer to the trigger and communication requests made by the NCAP.

5. ADDITIONAL TOOLS

The key of the IEEE1451 standard, like in our design, is the structure of TEDS and its inclusion in the STIM. Therefore, an operative system should include effective methods to introduce TEDS information in the STIM. We have developed an additional PC software tool in Delphi. At first, this tool was thought and used as pseudo-NCAP to test our first STIM. The next step was to include some windows facilities to collect the TEDS information, to translate it to the standard format and, finally, to write it into the STIM, that is a TEDS editor. In our actual system, this tool has been modified to use it also as a previous step to the STIM tables and indexes generation by means of the capture, treatment and classification of the TEDS data.

The first step to start working with the STIM is to configure it. That means to connect it through the parallel port to the PC and transfer critical classified TEDS information, from files created by the TEDS editor, to the ADuC, as well as the complete TEDS. The ADuC will store them into the different memories of the system in order to have them available during the normal STIM operation.

6. FUTURE WORKS

Our future work will be focused in two ways:

1. NVRAM is an expensive solution for non-volatile information. Our aim is to replace the NVRAM by two chips: a FLASH chip for TEDS and indexes of different channels in every section, and a RAM for buffers, status and mask registers, and intermediate buffer to save information in the FLASH memory. This solution is cheaper and powerful because we can increase memory size easily.
2. Development of an NCAP for Ethernet in order to perform tele-instrumentation with distributed systems within the IEEE-1451 standard in Internet.

ACKNOWLEDGEMENTS

We acknowledge the support of the CYCIT Spanish investigation project: “Arquitecturas de control distribuido. Análisis y diseño de estructuras con soporte de tecnologías Internet-Intranet”.

REFERENCES

[1] IEEE P1451.2 D3.01 Standard for a Smart Transducer Interface for Sensors and Actuators. June 1997.

[2] IEEE P1451.1/D1.83 Draft Standard for a Smart Transducer Interface for Sensors and Actuators – Network Capable Application Processor (NCAP) Information Model. December 1996.
[3] Kang B.Lee, Richard D. Schneeman
Distributed Measurement and Control Based on the IEEE 1451 Smart Transducer Interface Standards.
IMTC99. Venice (Italy). May 1999.
[4] L.Cámara, O.Ruiz, J.Samitier.
Complete IEEE-1451 Node, STIM and NCAP, Implemented for a CAN Network.
IMTC2000. Baltimore (USA). May 2000.
[5] Stan P. Woods
The IEEE-P1451 Tranducer to Microprocessor Interface.
SENSORS. June 1996
[6] J.Bryzek, R.Grace, K.Lee, P.Madan, J.Warrior, S.Woods.
Common Communication Interfaces for Networked Smart Sensors and Actuators.
SENSORS. September 1995.
[7] Robert N. Johnson.
Building Plug-and-Play Networked Smart Transducers.
SENSORS, October 1997.
[8] P. Conway, D.Heffernan, B.O'Hara, Prof. P.Burton, T.Miao
IEEE 1451.2: An interpretation and example implementation
IMTC 2000, Baltimore, MD, USA. May 2000.
[9] E.J. Mauders, L.A. Barford
Diagnosis of a continous dynamic system from distributed measurements
IMTC 2000, Baltimore, MD, USA. May 2000.