

# We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900

Open access books available

185,000

International authors and editors

200M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index  
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?  
Contact [book.department@intechopen.com](mailto:book.department@intechopen.com)

Numbers displayed above are based on latest data collected.  
For more information visit [www.intechopen.com](http://www.intechopen.com)



# Non-volatile memory interface protocols for smart sensor networks and mobile devices

Octavian Postolache, Pedro Silva Girão and José Miguel Dias Pereira  
*Instituto de Telecomunicações*  
*Portugal*

## 1. Introduction

Data acquisition, storage and transmission are mandatory requirements for different applications in the area of smart sensors and sensor networks.

Different architectures and scenarios can be considered. Thus for smart sensors architectures (Frank, 2002) based on the IEEE1451.X standard (IEEE, 2007) the acquired data can be processed at the smart sensor level using data of the so-called standard template TEDS (Honeywell, 2009) stored in a non-volatile memory (Brewer & Gill, 2008). The auto-identification of the smart sensor (Yurish & Gomes, 2003) unit from a sensor network is based on the Basic TEDS that also represents part of the stored information.

Considering smart sensor architectures (Song & Lee, 2008), the communication between the sensor processing unit (e.g. microcontroller) and one or multiple non-volatile memory units (e.g. Flash EEPROM memory units) is done using different communication protocols, such as SPI, I2C, 1-wire (Kalinsky & Kalinsky, 2002); (Paret & Fenger, 1997); (Linke, 2008). These protocols are thus frequently used in smart sensor implementations (IEEE, 2004)(Ramos et al., 2004).

As the name implies, smart sensors networks are networks of smart sensors, that is, of devices that have an inbuilt ability to sense information, process the information and send selected information to an external receiver (including to other sensors). A "smart sensor" is a transducer (or actuator) that provides functions beyond what is necessary to generate a correct representation of a sensed or controlled quantity. This means that such nodes require memory capabilities to store data temporarily or permanently.

In an increasingly number of applications, the nodes are required to change their spatial position (mobile nodes), which leads to wireless networks. The sensor network nodes data management and advanced data processing are carried out by a host unit characterized by high data processing capabilities, non-volatile data storage capabilities and data communication capabilities. One kind of solutions that materialize the host unit is mobile devices (e.g. phones and PDAs) with special operating systems (e.g. WindowsCE, Symbian, BalckBerry OS) and internal and extended data storage memory capabilities (CF card memory, SD card memory). Specific protocols, *CompactFlash* and *Secure-Digital* (Compact Flash, 2009) (SD Association, 2009) are associated with memory card interfaces that are used to perform the communication between the host unit processor and the memory units. In wireless sensors networks, special attention is granted to the memory read/write operations

time interval and the associated power consumption considering the mobile device autonomy requirements.

Considering the importance of non-volatile memory and the communication protocols associated with memory units as parts of smart sensors, sensor networks and distributed mobile systems (e.g. wearable sensing system for physiological parameter measurements), the proposed chapter briefly reviews some non-volatile memory solutions more used in those contexts.

## 2. Non-volatile Memory, Smart Sensors and Mobile Devices

A non-volatile memory is an important part of a smart sensor. Non-volatile memory is a general term for all forms of a solid state memory that do not need to have their memory contents periodically refreshed. This includes all forms of read-only memory (ROM) such as programmable read-only memory (PROM), erasable programmable read-only memory (EPROM), electrically erasable programmable read-only memory (EEPROM), and flash memory. It also includes the random access memory (RAM) that is powered by a battery.

Regarding smart sensors, the non-volatile memory stores a table of parameters that identify the transducer and are held in the transducer on an EEPROM for interrogation by external electronics. The table's contents can be the one defined by the IEEE committee associated with IEEE1451 standard for smart sensors (Ulivieri et al., 2009). The transducer parameters table is also known as TEDS (Transducer Electronic Datasheet) and according with the above mentioned standard, brings plug-and-play capabilities to transducers. TEDS-compatible measurement systems can auto-detect and automatically configure these "smart sensors" for measurement, reducing setup time and eliminating transcription errors that commonly occur during sensor configuration.

Different IEEE1451 family members include the TEDS module as part of particular smart sensor architecture implementations. Figure 1 presents the TEDS module localization according to IEEE1451.2, IEEE1451.3, IEEE1451.4, IEEE1451.5 and IEEE1451.6. Figure 1 shows that the TEDS that is localized at the transducer's level sends the specific transducer information to the Network Capable Information Processor (NCAP) using different kinds of interfaces. The smart transducers communicate with the NCAP using the mixed mode interface that joins the analogue signal line and the memory communication lines according to IEEE1451.4, using CAN (*Controller Area Network*) (Pfeiffer et al., 2003) interface according to IEEE1451.6, using digital point-to-point according to IEEE1451.2, using distributed bus interfaces such as I2C (Pardo et al., 2006) according to IEEE1451.3 or using wireless interfaces such as Bluetooth (Flittner, 2007) or ZigBee (Higuera, 2009) according to IEEE1451.5. Additionally, an IEEE1451 standard extension for smart sensors with RFID (IEEE1451.7) is under discussion.

From the different individuals in the IEEE1451 standard family one of the more implemented is the mixed mode transducer interfaces IEEE1451.4. The IEEE 1451.4 standard defines a mechanism for adding self-identification technology to traditional analogue sensors and actuators (IEEE, 2009) and it is described in details in the next section.

### IEEE1451.4

IEEE 1451.4 (dot 4 from now on) defines a mechanism for adding self-describing behaviour to traditional transducers with an analogue signal interface. The dot 4 defines the concept of

a transducer that supplies both analogue and digital interface, namely, *mixed-mode interface* (Fig.2) where the TEDS non-volatile memory localization is better highlighted. In this case, the non-volatile memory interface will materialize the digital interface associated with IEEE1451.4.

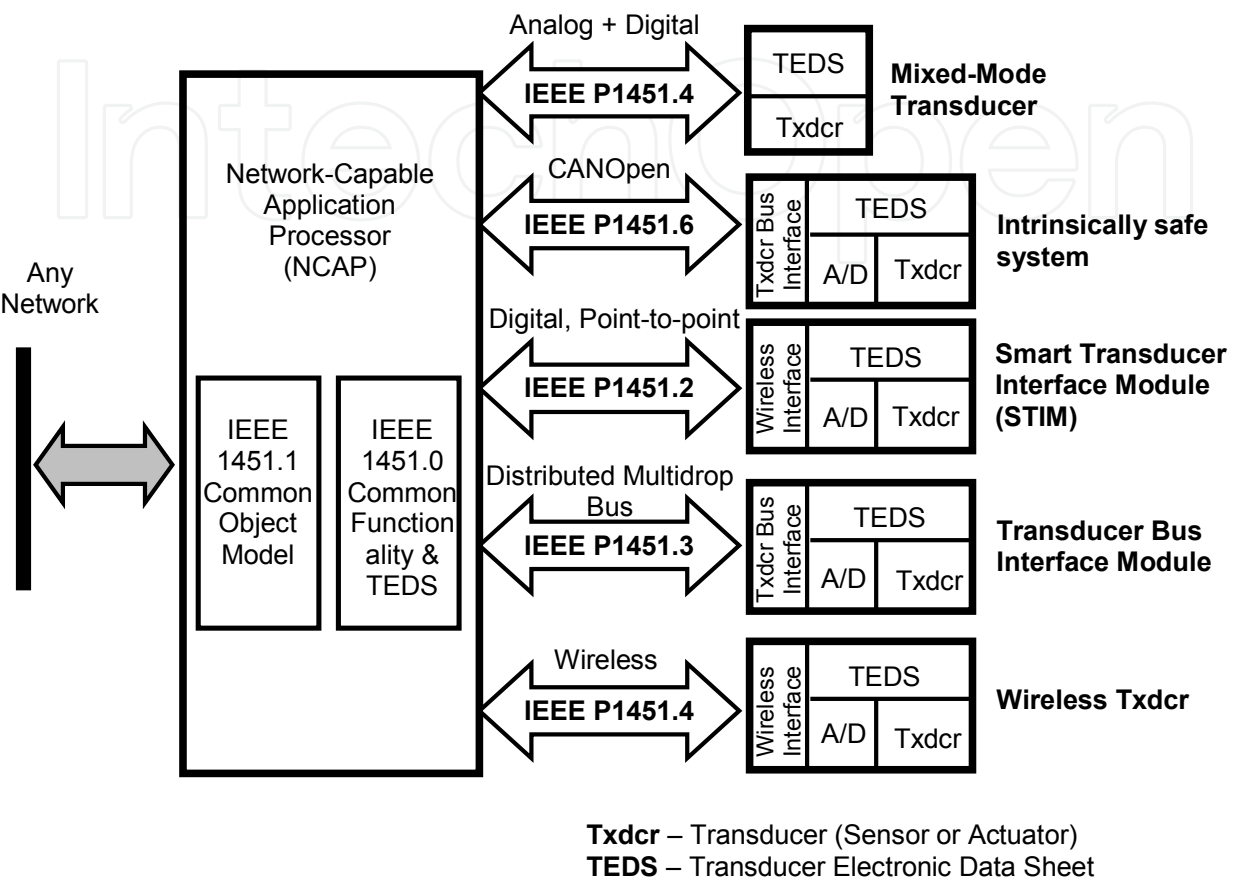


Fig. 1. TEDS on IEEE 1451 family of smart transducer interface standards network (Txdcrc – transducer sensor or actuator)

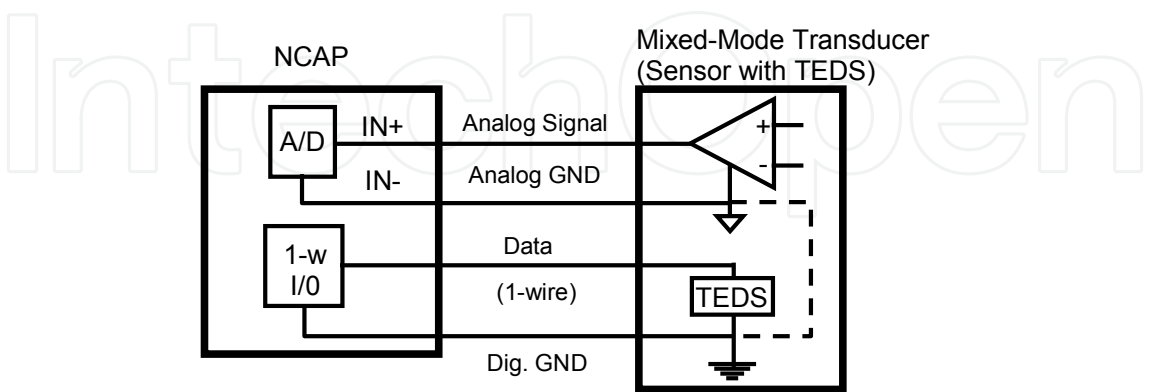


Fig. 2. Non-volatile memory for TEDS associated with *mixed-mode interface* for smart sensors (NCAP – network capable application processor)

The memory interfaces that are mainly used in the IEEE1451.4 implementation are 1-wire and I2C. In the following paragraph a brief description of the 1-wire and 2-wire (I2C) memory interface protocols particularly used in smart sensor implementation will be presented.

3. Memory Interfaces Protocols for Smart Sensors

The 1-wire interface provides advantages over other industry interfaces such as I<sup>2</sup>C or SPI™ in applications where contacts between the host controller and the memory device are limited or must be minimized, and/or where factory-programmed unique device serialization is required or valuable. 1-wire devices require a total of two contacts for total device operation, compared to four contacts for I<sup>2</sup>C memory or five contacts for SPI memory.

3.1 1-wire interface protocol for smart sensor memories

1-wire protocol was developed by Dallas Semiconductor in order to permit digital communications over twisted-pair cables with 1-wire components over a 1-wire network. A 1-wire bus uses only one wire for signalling and power. Communication is asynchronous and half-duplex, and it follows a strict master-slave scheme. One or several slave devices can be connected to the bus at the same time. Only one master should be connected to the bus. 1-wire bus electronics implements an open-drain (wired-AND) master/slave multidrop architecture with resistor pull-up to a nominal supply at the master. A block diagram of 1-wire network is presented in Fig. 3.

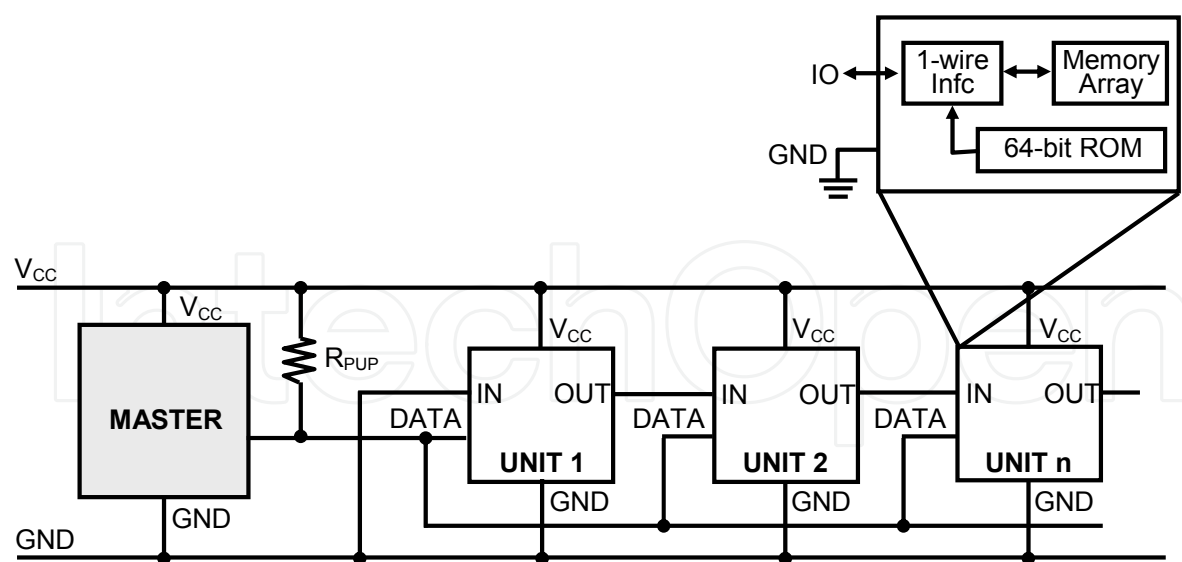


Fig. 3. - 1-wire network of non-volatile memories

The 1-wire network has three main components: a bus master with controlling software, wiring and associated connectors (e.g. Esensors connectors) and 1-wire devices (unit1, unit2, unit3,...unitn) that can be sensors, actuators and memories (e.g. DS2430A from Maxim). The

protocol uses conventional CMOS/TTL logic levels (maximum 0.8V for logic “zero” and a minimum 2.2V for logic “one”) with operation specified over a supply voltage range of 2.8V to 6V. System clock is not required; each 1-wire part is self-clocked by an internal oscillator synchronized to the falling edge of the master.

Signalling on the 1-wire bus is divided into time slots of 60us. One data bit is transmitted on the bus per time slot. Units are allowed to have a time base that differs significantly from the nominal time base. This however, requires the timing of the master to be very precise, to ensure correct communication with slaves with different time bases.

**Addressing**

All 1-wire devices have a unique address laser-registered into the chip. Dallas Semiconductors guarantees that the address is unique. The individual address is expressed by a 64-bit serial number that is stored in the device memory. It is composed of eight bytes divided into three main sections as presented in Table 1.

Family code	ID	CRC8
8 bits	48 bits (unique within family)	8 bits

Table 1. – 1-wire address format. Left to right: increasing time, increasing bit order

Starting with the least significant bit (LSB), the first byte stores the 8-bit family codes that identify the device type. For the particular case of 1-wire memories the family codes are presented in Table 2.

1-wire memory	Family code
1k memory iButton	08
4k memory iButton	06
16k memory iButton	0A
64k memory iButton	0C
4k EEPROM	23
1k EEPROM	2D
256 EEPROM	14
1k EEPROM protected with SHA-1	33

Table 2. - 1 wire address family code for several memory devices

The next six bytes store a customizable 48-bit individual address or ID that guaranteed unique within a family. A few types of chips have sequences of IDs reserved for special manufacturing runs, but in general, there are no special characteristics to the ID. The last byte, the most significant byte (MSB), contains a cyclic redundancy check (CRC) with a value based on the data contained in the first seven bytes. This allows the master to determine if an address was read without error.

With a 2<sup>48</sup> serial number pool, conflicting or duplicate node addresses on the net are never a problem. For maximum data security the 1-wire memories can implement US government-certified Secure Hash Algorithm (SHA-1).

### Basic bus signals

As mentioned before, 1-wire memory interface protocol uses a single wire (plus ground) to accomplish both communication and power transmission. A single bus master can feed multiple slaves over a single twisted-pair cable. Thus, the master initiates every communication on the bus down to the bit-level. This means that for every bit that is to be transmitted, regardless of direction, the master has to initiate the bit transmission. This is always done by pulling the bus low, which will synchronize the timing logic of all units.

### 1-wire bus commands and operations

The communication between the master and slaves uses a set of five basic commands of the 1-wire bus: "Write 1", "Write 0", "Read", "Reset" and "Presence".

- **Write 1** - The master pulls the bus low for 1 to 15  $\mu$ s. It then releases the bus for the rest of the time slot (Fig.4).



Fig. 4. - Write 1 - command on the 1-wire bus

- **Write 0** - The master pulls the bus low for a period of at least 60  $\mu$ s, with a maximum length of 120  $\mu$ s (Fig.5).



Fig. 5. - Write 0 - command on the 1-wire bus

- **Read** - The master pulls the bus low for 1 to 15  $\mu$ s. The slave then holds the bus low if it wants to send a '0'. If it wants to send a '1', it simply releases the line. The bus should be sampled 15 $\mu$ s after the bus was pulled low. As seen from the master's side, the "Read" signal is in essence a "Write 1" signal. It is the internal state of the slave, rather than the signal itself that dictates whether it is a "Write 1" or "Read" signal (Fig. 6).



Fig. 6. - Read command on the 1-wire bus

- **Reset & Presence** - The master pulls the bus low for at least 8 time slots or 480 $\mu$ s and then releases it. This long low period is called the "Reset" signal. If there is a slave present, it should then pull the bus low within 60 $\mu$ s after it was released by the master and hold it low for at least 60 $\mu$ s. This response is called a "Presence" signal. If no presence signal is issued on the bus, the master must assume that no device is present on the bus, and further communication is not possible (Fig.7).



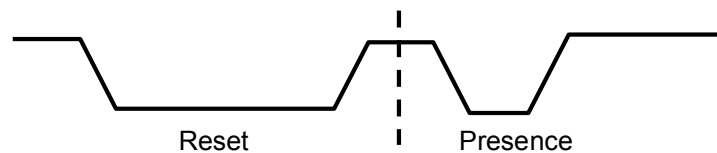


Fig. 7.- Reset & Presence – command on the 1-wire bus

The communication between the master and slave is performed using the 1-wire commands according with the flowchart that is presented in Fig.8.

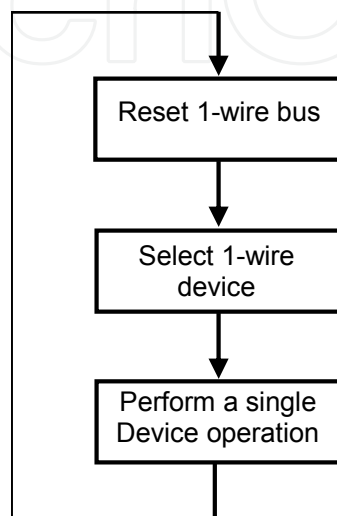


Fig. 8. - 1-wire master-slave communication flowchart

The first step of the communication is materialized by the „reset” command that is delivered by the master synchronizing the entire 1-wire bus. One of the unit1, unit2, ..., unit n (slave devices) is selected for the next communication. The selection of the specific slave is done using the serial number of the device or using a binary search algorithm (Maxim, 2002). Once a specific device has been selected, all other devices drop out and ignore subsequent communications until the next reset is carried out.

Because each device type performs different functions and serves a different purpose, each has a unique protocol once it has been selected. For the particular case of a non-volatile memory, a set of particular commands are mentioned:

- **Write Scratchpad [0Fh]** - applies to the data memory and the writable addresses in the register page. After issuing the Write Scratchpad command, the master must first provide the 2-byte target address, followed by the data to be written to the scratchpad;
- **Read Scratchpad command [AAh]** - allows verifying the target address and the integrity of the scratchpad data. After issuing the command code, the master begins reading. The master should read through the end of the scratchpad, after which it receives an inverted CRC16, based on data as it was sent by the 1-wire memory.
- **Copy Scratchpad command [55h]** - is used to copy data from the scratchpad to the data memory and the writable sections of the register page.
- **Read Memory command [F0h]** - is the general function to read from the 1-wire memory. After issuing the command, the master must provide a 2-byte target



address, which should be in the range of 0000h to 0A3Fh. If the target address is higher than 0A3Fh, for the particular case of DS28EC20 (1-wire memory), the upper four address bits are changed to “0”. After the address is transmitted, the master reads data starting at the (modified) target address and can continue until address 0A3Fh. If the master continues reading, the result is FFh. The Read Memory command sequence can be ended at any point by issuing a reset pulse.

- **Extended read memory [A5h]-** works essentially the same way as Read Memory, except for the 16-bit CRC that the DS28EC20 generates and transmits following the last data byte of a memory page. The Extended Read Memory command sequence can be ended at any point by issuing a reset pulse.

**Master host computer interfacing**

In a 1-wire network associated with smart sensors the master is generally a microcontroller. An example of DS28EC20 connection to the microcontroller as part of an IEEE1451.4 implementation is presented in Fig. 9. It presents the advantage of compactness but requires 1-wire implementation protocol at the microcontroller level. 1-wire protocol implementation for Atmel microcontroller (Atmel, 2004) and PIC microcontroller (Maxim, 2003) are referred in the literature.

Using a serial 1-wire line driver, the 1-wire memory device can be connected to the UART port of the microcontroller reducing the design time. A solution in this field is the MAXIM’s DS2480B chip. It connects directly to UARTs and 5V RS232 systems. Interfacing to RS232C ( $\pm 12V$  levels) requires a passive clamping circuit and one 5V to  $\pm 12V$  level translator such as MAX232. Internal timers relieve the host of the burden of generating the time-critical 1-wire communication waveforms. The DS2480B can be set to communicate at four different data rates, including 115.2kbps, 57.6kbps, and 19.2kbps, with 9.6kbps being the power-on default. Command codes received from the host's crystal controlled UART serve as a reference to continuously calibrate the on-chip timing generator. The various control functions of the DS2480B are optimized for MicroLAN 1-wire networks and support the special needs EPROM-based add-only memories and EEPROM devices as well as the other 1-wire devices (e.g. ibutton, 1-wire thermometers).

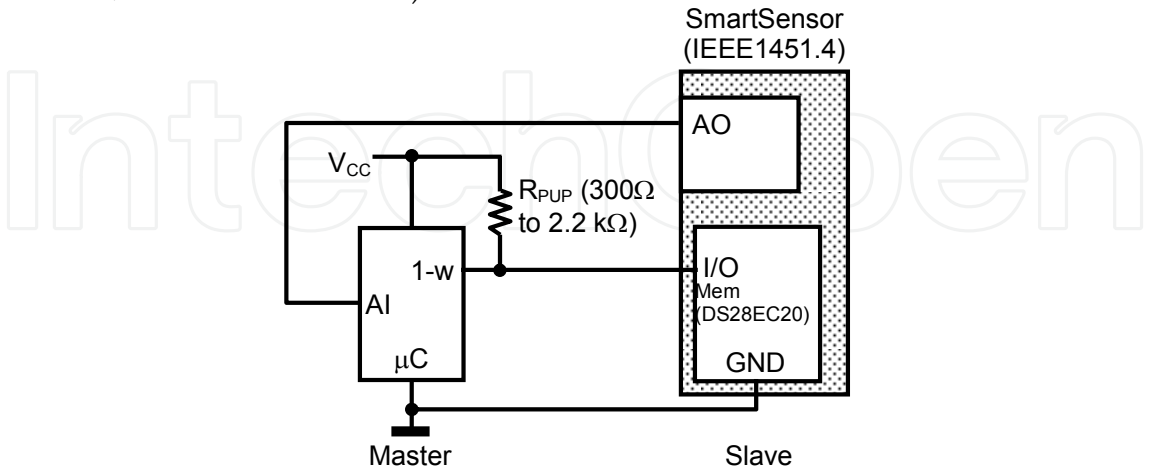


Fig. 9. - The interfacing of memory as part of IEEE1451.4 smart sensor to the microcontroller ( $\mu C$ )

Referring to the implementation of 1-wire software that works under Windows OS, an 1-wire *Software Development KIT (SDK)* can be used to develop applications associated with IEEE1451.4 smart sensor data management. The 1-wire SDK includes 1-wire API for .NET, API examples in VB.NET and C#, along with TMEX API examples in C, C++, Pascal (Borland Delphi), and Microsoft Visual Basic that assures high degree of flexibility and reduced time of software design and implementation.

3.2 2-wire (I2C) interface protocol for smart sensor memories

In the early 1980's, Philips Semiconductors developed a simple bi-directional 2-wire bus for developing networks of integrated circuits (IC), including memories. This bus is called the Inter-Integrated Circuit or I2C-bus. At present, Philips' IC range includes more than 150 CMOS and bipolar I2C-bus compatible types for performing communication functions between intelligent control devices (e.g. microcontrollers), general-purpose circuits (e.g. LCD drivers, remote I/O ports, memories) and application-oriented circuits (e.g. digital tuning and signal processing circuits for radio and video systems). Taking into account the capabilities of I2C bus on device interfacing, it was considered as an interesting solution for smart sensors (SmartS) interfacing, including those compatibles with the IEEE1451 protocol. An example of I2C network, including a set of memories as part of IEEE1451.4 smart sensor network implementation, is presented in Fig. 10.

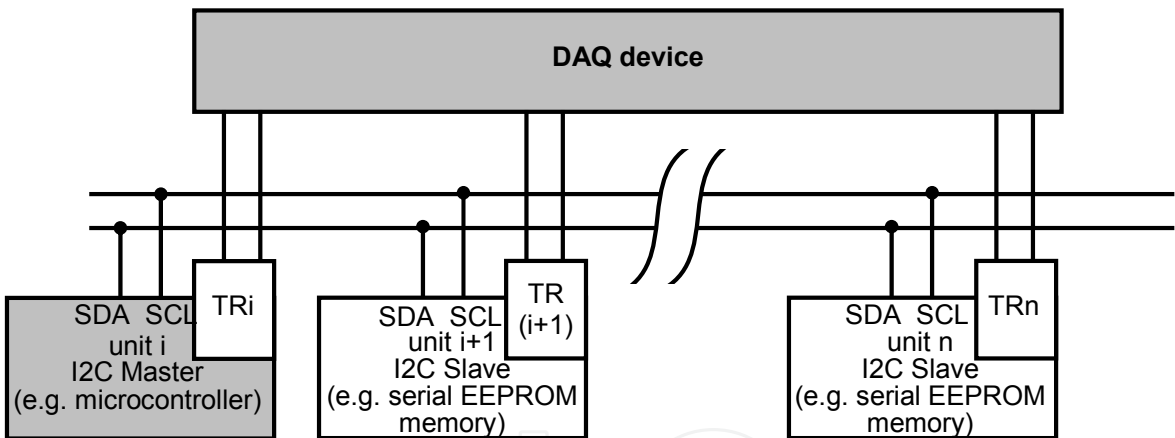


Fig. 10. - I2C device network including 1 master device and n – slave devices ( TR<sub>i</sub>, TR<sub>(i+1)</sub>, TR<sub>n</sub> – analog transducers, DAQ device – analog signal acquisition devices)

Each device has a particular address. I2C bus supports two addressing schemes: 7-bit address and 10-bit address. Up to 1024 devices are allowed to be connected to the bus. The 7-bit address scheme has shorter message length and requires less complex hardware. Devices with 7 and 10 bit addresses can be mixed in the same system. The bus can operate in three modes with different data rates. Data on the bus can be transferred at rates of up to 100kbit/s in the standard-mode, up to 400 kbit/s in the fast-mode, or up to 3.4 Mbit/s in the high-speed mode. The number of interfaces connected to the bus is dependent on the bus capacitance limit of 400pf (Phillips, 2000).

**Basic bus signals**

The I2C bus wires are called SDA (Serial DATA line) and SCL (Serial CLock line). They are both bi-directional and assure the communication between the network devices through the SDA and SCL signals (Fig.11). As can be observed, the communication signals are characterized by: the start bit, the 7 bits (B6...B0) I2C device address, the RD/nWR bit associated with master-slave read or write operations signalling, the ACKNOWLEDGE bit, the 8 (D7...D0) data bits, and the STOP bit.

The SCL signal is an explicit clock signal on which the communication synchronizes and is imposed by the master unit. When the slave unit is not able to follow with the clock rate given by the master a mechanism clock stretching is activated on the slave level (Leens, 2009).

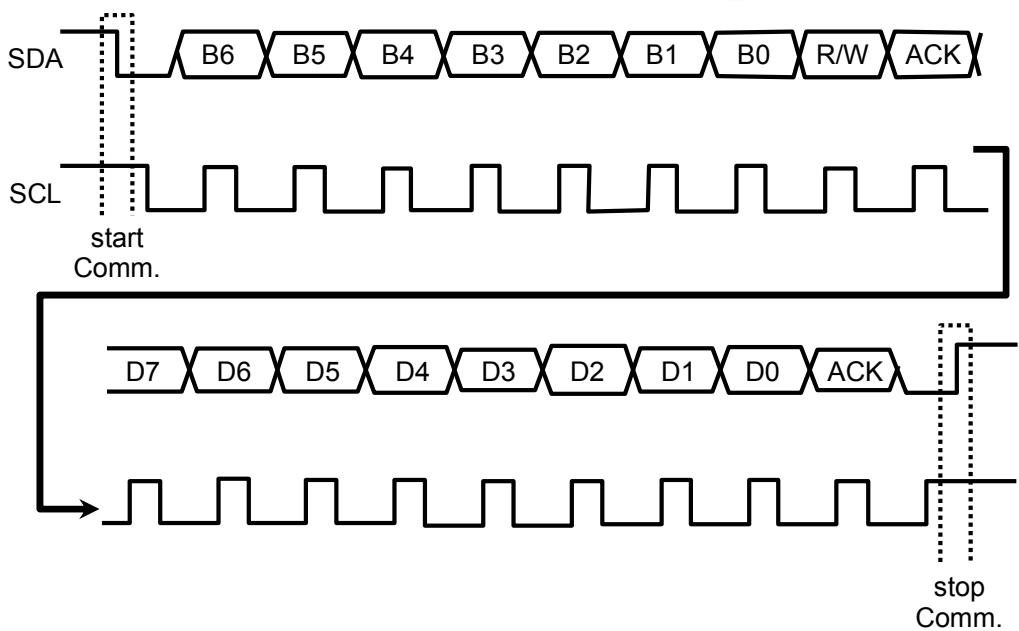


Fig. 11. - The SDA and SCL signals associated with the I2C bus communication

The I2C bus does not support plug-n-play or interrupt functions that are important for many sensor networks or memory networks. To save energy, some sensor nodes should be in sleep mode most of the time and woken up by a timer or sensing event. For each transported information, the microcontroller should initiate the request and provide the clock to the sensors. To get the updated information from the sensor, the microcontroller has to poll each sensor node connected on the bus very often to make sure it will not miss new information or unexpected events. These features make I2C unsuitable for the applications which have strict requirement for the power efficiency and emergency processing. Although the sensor node could be clipped to or from the system easily, the microcontroller can not detect an event or configure the system during operation. This limits the application of I2C in sensor networks which are usually dynamically reconfigurable and/or demand high energy-efficiency.

### Commands and operations

Every device connected to the bus has its own unique address and can act as a receiver and/or transmitter, depending on its own functionality.

The I2C bus is a multi-master bus. Thus, more than one device with I2C interface are capable of initiating a data transfer. The I2C protocol specification states that the unit that initiates a data transfer on the bus is considered the *Bus Master*. Consequently, at that time, all the other units are regarded to be *Bus Slaves*. The master can send data to a slave or receive data from a slave. Slaves do not transfer data between themselves.

Considering the particular scenario of a microcontroller as master of I2C bus that includes memories associated with smart sensor TEDS as bus slaves, the commands and operation associated with master-slave, slave-master communication are next presented.

First, the master will send out a START signal. This acts as an “Attention” signal sent to the bus units. All of the units connected to the bus will listen for incoming data. Then, the master sends the ADDRESS of the device it wants to access, along with an indication of whether the access is a Read or Write operation (“1” logic for read operation and “0” for write operation).

Having received the address, all ICs will compare it with their own address. If it does not match, they simply wait until the STOP command from the master.

Considering that the received address by the slave matches its proper address, the response is materialized by the ACKNOWLEDGE signal. Once the master unit receives the ACKNOWLEDGE, it can start transmitting or receiving DATA according to the RD/nWR bit.

When all the data transmission is done the master will send the STOP command. This is the signal that the bus has been released and that the connected units may expect another transmission to start any moment.

Referring to memories as I2C bus units, there are several important manufactures of very broad range of memory capacities with I2C bus interface. As common used memories in smart sensor implementation can be mentioned the Microchip memories, that are characterized by capacities from 16 bytes, for the 24C00, up to 32k bytes, for the 24C256 family.

## 4. Memory Interfaces Protocols for Mobile Devices

Mobile devices must have memories that assure the capability to store applications and data files. However, data logging tasks are always related to the utilization of memory extension cards, such as CompactFlash memories (CF card) or Secure Digital memories (SD card). Considering the importance of mobile devices external memories, two of the communications protocols associated with this kind of memories are now described.

### 4.1 Compact Flash memory protocol

CompactFlash® (CF) has established itself as a dominant flash memory card technology in applications where small form factor, low-power dissipation and ease-of-design are crucial considerations. Introduced by SanDisk Corporation in 1994, CF Storage Cards provide the capability to transfer all types of digital information and software between different types of digital systems and are compatible with future CF designs, eliminating interoperability restrictions. As a result, CF is a powerful solution that can be found in digital SLR cameras, PDAs (e.g. iPAQ 2700 series), embedded systems, single-board computers and data recorders.

CF based data storage is used especially at the master device level such as PDAs (Postolache, 2006) or touch panel computers (e.g. TPC2106T) (Postolache, 2007).

Two types of CF non-volatile memory cards are considered:

- Type I cards or CF, dimensionally characterized by 43mm (1.7") x 36mm (1.4") x 3.3mm (0.13");
- Type II cards or CF2, dimensionally characterized by 43mm (1.7") x 36mm (1.4") x 5mm (0.19").

CompactFlash cards are designed with flash technology [(Kingmax Digital Inc., 2009)]. According to the CompactFlash card specification version 4.1 it can be characterized by data storage capacities up to 137GB. However, the current commercial CF solutions have data storage capacities up to only 64GB (e.g. Pretec CF memory).

The CF classification by speed is:

- original CF;
- CF High Speed (using CF+/CF2.0);
- CF3.0 standard.

Compact Flash supports data rates up to 133MB/sec. However, values associated with current commercially CF devices are up to only 45Mb/s (e.g. 8GB ScanDisk Extreme).

### **CF electrical interface**

The internal configuration of the CF protocol associated to CF non-volatile memory cards includes a CF controller connected through I/O digital lines to a host interface that can be associated with PDAs, laptop computers, tablet PC, etc. The CF block diagram and the CF connector pin-out are presented in Fig. 12.

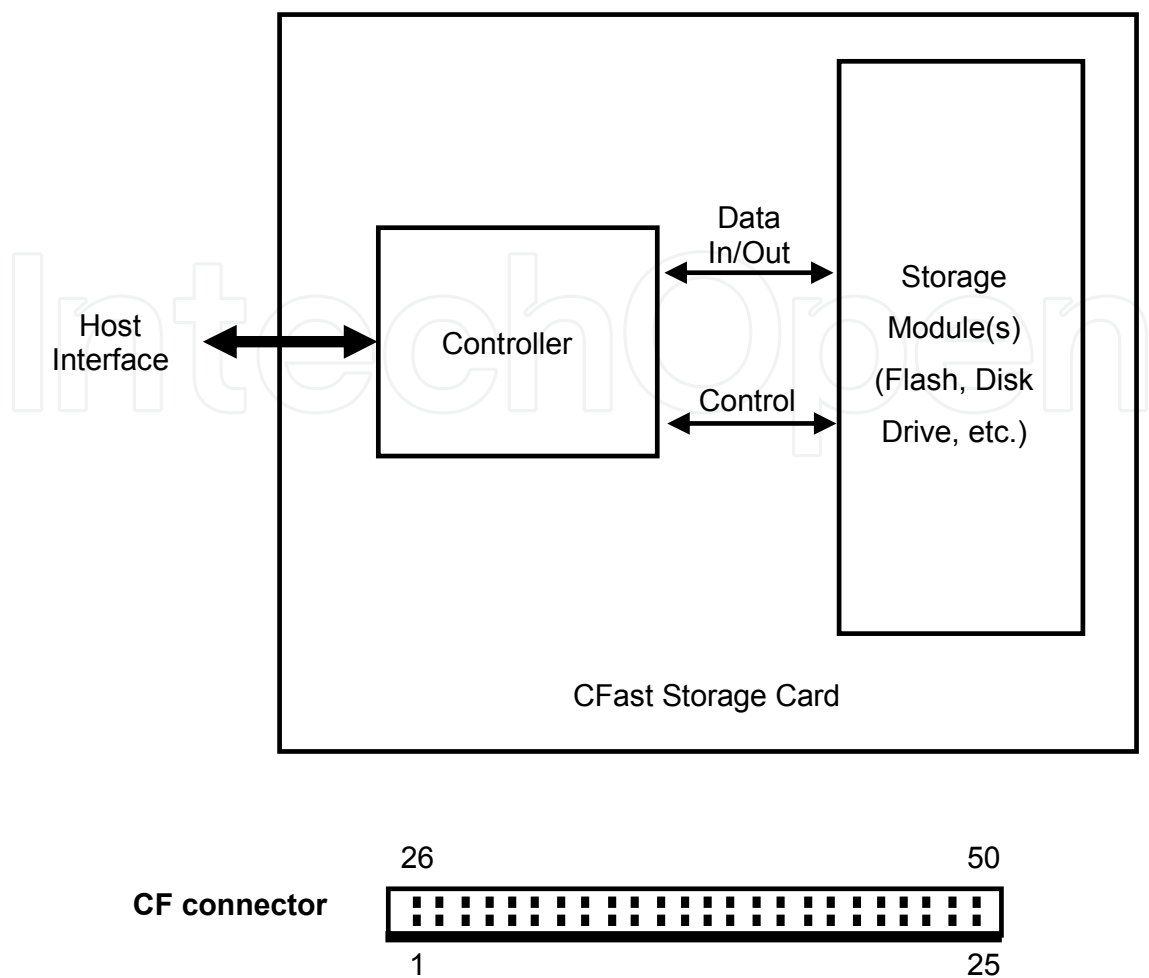


Fig. 12. - Compact Flash block diagram and CF connector pin-out

Pins 13 and 38 correspond to power supply, which according to the standard can be either 3.3 volts or 5 volts. Pins 1 and 50 correspond to GND. The data stored on the CF can be accessed through 8 or 16 bit data bus associated to the CF connector. The data and address bits for 8 and 16 bits memory access bus are shown in Table 3.

Pins	21	22	23	2	3	4	5	6
Data lines	D00	D01	D02	D03	D04	D05	D06	D07
Pins	47	48	49	27	28	29	30	31
Data lines	D08	D09	D10	D11	D12	D13	D14	D15

Table 3. - CF pins and data line correspondence for 8bit and 16 bit data bus for memory access.

The address bus (A0 to A10 lines) pin assignment is given in Fig. 13.

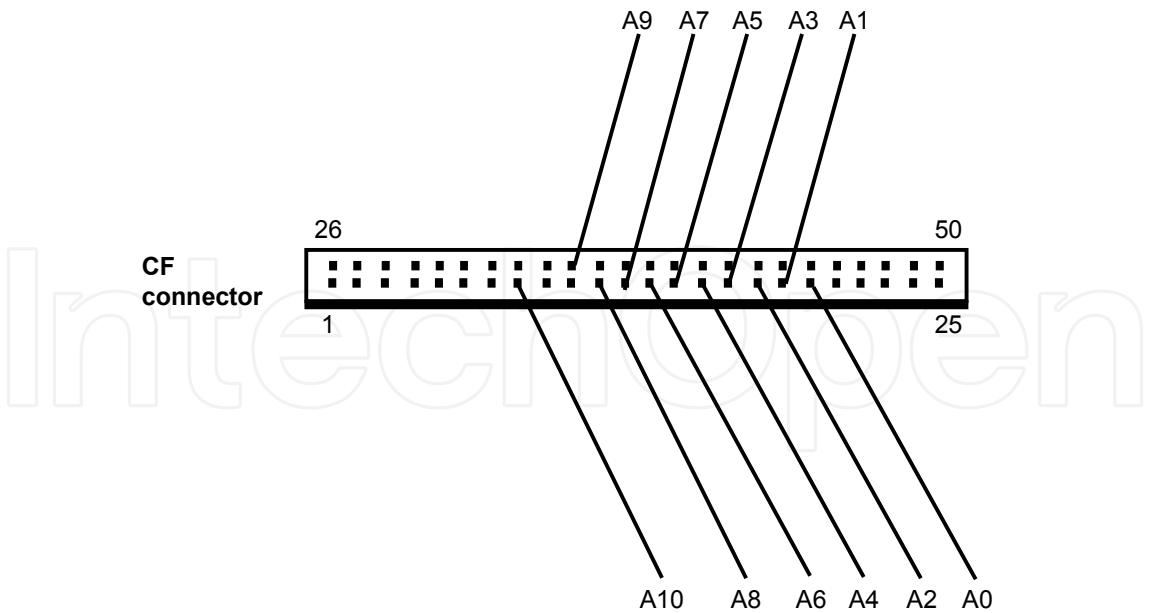


Fig. 13. - CF's connector address bus pin out

4.2 Secure digital (SD) memory protocol

The SD card standard is a standard for removable memory storage designed and licensed by the SD Card Association (Compact Flash, 2009). It is the result of the collaboration between three manufacturers, Toshiba, SanDisk, and MEI (Panasonic) and is specifically designed to meet the security, capacity, performance, and environment requirements inherent in newly emerging audio and video consumer electronic devices as well as an important component of smart sensing networks that include smart mobile devices such as smart phones and PDAs, assuring data logging capabilities for individual nodes.

At the same time, the SD standard is not limited to removable memory storage devices and has been adapted to many different classes of devices, including 802.11 cards, Bluetooth devices, and modems.

The SD Memory Card includes a content protection mechanism that complies with the security of the SDMI (Secure Digital Music Initiative) standard and is faster and capable of higher memory capacity. Nowadays, SD high-capacity (SDHC) cards start at 4GB and going up to 32GB.

In what concerns the operating rate of the SD card working in the default mode, typical values are up to 12.5 MB/sec interface speed for a variable clock rate 0 - 25 MHz, while for high-speed working mode are reported values of up to 25 MB/sec interface speed for a variable clock rate from 0 - 50 MHz.

SD electrical interface

A block diagram of an SD card interface associated with a SD card non-volatile memory is presented in Fig. 14 and the corresponding pin-out, together with the SD function are presented in Table 4.

The SD card is clocked by an internal clock generator. The interface driver unit synchronizes the DAT and CMD signals from external CLK to the internal used clock signal. The card is controlled by the six line SD card interface containing the signals: CMD, CLK,DAT0~DAT3.



For the identification of the SD card in a stack of SD card, a card identification register (CID) and a relative and address register (RCA) is foreseen [(Kingmax Digital Inc., 2009)]. An additional register, (CSD), contains different types of operation parameter.

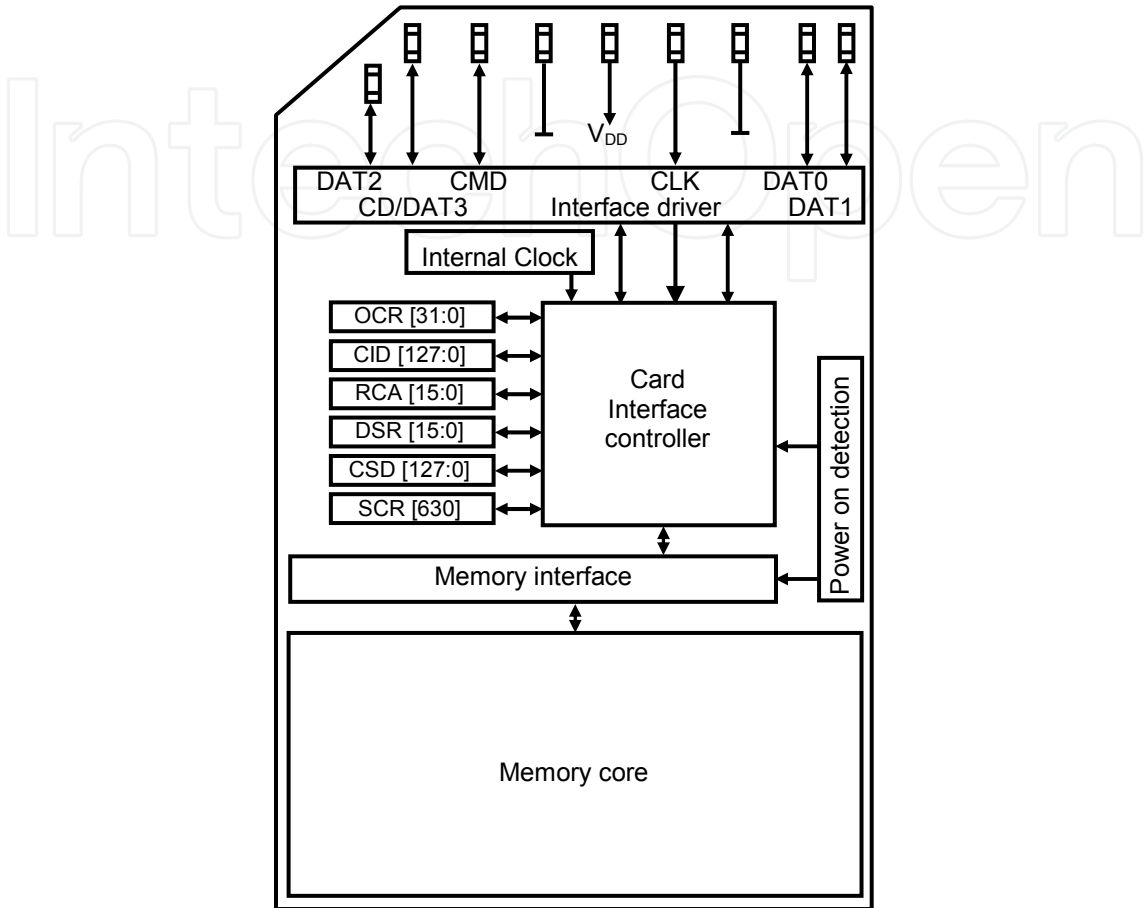


Fig. 14. - SD card block diagram

Pin	Name	SD function
1	DATA3/CS	Data Line 3
2	CMD/DI	Command Line
3	VSS1	Ground
4	VDD	Supply Voltage
5	CLK	Clock (SCK)
6	VSS2	Ground
7	DAT0/D9	Data Line 0
8	DAT1/IRQ	Data Line 1
9	DAT2/NC	Data Line 2

Table 4. – SD pin out

The card has its own power on detected unit. No additional master reset signal is required to setup the card after power on. It is protected against short circuit during insertion and removal while the SD card system is powered up

The communication using the SD card lines to access either the memory field or the register is defined by the SD card standard. Different protocols are supported by SD cards.

### **SD 1 bit protocol**

It is a synchronous serial protocol with one data line, used for bulk data transfers, one clock line for synchronization, and one command line, used for sending command frames. The SD 1-bit protocol explicitly supports bus sharing. A simple single-master arbitration scheme allows multiple SD cards to share a single clock and DAT0 line.

### **SD 4 bit protocol**

It is nearly identical to the SD 1-bit protocol. The main difference is that bulk data transfers use a 4-bit parallel bus instead of a single wire. With proper design, this has the potential to quadruple the throughput for bulk data transfers. Both the SD 1-bit and 4-bit protocols by default require a Cyclic Redundancy Check (CRC) protection of bulk data transfers.

Cyclic Redundancy Check is a simple method for detecting the presence of simple bit-inversion errors in a transmitted block of data. In SD 4-bit mode, the input data is multiplexed over the four bus (DAT) lines and the 16-bit CRC is calculated independently for each of the four lines, which imply an increased software complexity of CRC calculation. Hardware implementation of 4-bit parallel CRC calculation represents an interesting alternative and can be materialized using an Application Specific Integrated Circuit (ASIC) or field programmable gate arrays (FPGA).

### **SPI mode – SD card protocol**

It is distinct from the 1-bit and 4-bit protocols in that the protocol operates over a generic and well-known bus interface, Serial Peripheral Interface (SPI). SPI is a synchronous serial protocol that is extremely popular for interfacing peripheral devices with microcontrollers (Leens, 2009). Most modern microcontrollers, including the MSP430, support SPI natively at relatively high data rates. The SPI communications mode supports only a subset of the full SD Card protocol. However, most of the unsupported command sets are simply not needed in SPI mode. A fully-functional SD Card implementation can be realized using only SPI.

This flexibility of electrical interfaces is a significant advantage to a designer. A designer may opt for a fast parallel interface, or depending on the application, may prefer a slower implementation using SPI. Due to the popularity of the SPI protocol and its efficient implementation on the MSP430, we mention here only the SPI mode of the SD Card protocol. Minor differences in the initialization sequence exist between the two major modes. For more information, the interested reader should consult the full SD Card Specification available from the SD Card Association.

## **5. References**

Atmel (2004). AVR318: Dallas 1-Wire master 8-bit Microcontrollers, on-line at: [www.atmel.com/dyn/resources/prod\\_documents/doc2579.pdf](http://www.atmel.com/dyn/resources/prod_documents/doc2579.pdf), Sept-2004

- Brewer, J. & Gill, M. (2008). *Nonvolatile Memory Technologies with Emphasis on Flash: A Comprehensive Guide to Understanding and Using Flash Memory Devices*, Wiley-IEEE Press, February 2008
- Compact Flash Org. (2009). CF+ & CompactFlash Specification Revision 4.1, on line at [http://www.compactflash.org/spec\\_download.htm](http://www.compactflash.org/spec_download.htm), May 2009
- Flittner P. (2007). CSR (Bluetooth Subgroup Chair) Thurston Brooks, 3eTI, *IEEE P1451.5 Wireless Sensor Interface Working Group Bluetooth Subgroup. Proposal* on-line at: [grouper.ieee.org/groups/1451/5/.../P1451.5\\_Bluetooth2.pdf](http://grouper.ieee.org/groups/1451/5/.../P1451.5_Bluetooth2.pdf)
- Frank, R. (2002). *Understanding Smart Sensors*, Artech House Publishers, April 2002
- Higuera J.; Polo J.; Gasulla M. (2009). A Zigbee wireless sensor network compliant with the IEEE1451 standard, *Proceedings of IEEE Sensors Applications Symposium, SAS 2009*, pp.309-313, 2009
- Honeywell (2009). "TEDS - plug and play sensor configuration", on-line at <http://www.sensotec.com/pnpterm.shtml>, June 2009
- IEEE Std 1451.4-2004 (2004), Standard for a Smart Transducer Interface for Sensors and Actuators- Mixed-Mode Communication Protocols and Transducer Electronic Data Sheet (TEDS) Formats, *IEEE Standards Association*, Piscataway, NJ, sub classe 5.1.1, 2004
- IEEE STD 1451.0-2007 (2007). Standard for a Smart Transducer Interface for Sensors and Actuators - Common Functions, Communication Protocols, and Transducer Electronic Data Sheet (TEDS) Formats, IEEE Instrumentation and Measurement Society, TC-9, The Institute of Electrical and Electronics Engineers, Inc., New York, N.Y. 10016, SH99684, October 5, 2007
- IEEE Standard Association (2009). IEEE Standard 1451.4-2004 Tutorials, on-line at: <http://standards.ieee.org/regauth/1451/Tutorials.html>, 2009
- Kalinsky D. & Kalinsky R. (2002). Introduction to Serial Peripheral Interface, on-line at <http://www.embedded.com/story/OEG20020124S0116>, January '02
- Kingmax Digital Inc. (2009). "SD Card Specification" on line at: [downloads.amilda.org/MODs/SDCard/SD.pdf](http://downloads.amilda.org/MODs/SDCard/SD.pdf)
- Leens F. (2009). An introduction to I2C and SPI protocols, *Instrumentation and Measurement Magazine*, vol.12, No. 1, pp. 8-13, February 2009
- Linke B. (2008). Overview of 1-Wire® Technology and Its Use, on-line at [http://www.maxim-ic.com/appnotes.cfm/an\\_pk/1796](http://www.maxim-ic.com/appnotes.cfm/an_pk/1796), June 2008
- Maxim Inc. (2002), 1-Wire Search Algorithm, on-line at [http://www.maxim-ic.com/appnotes.cfm/an\\_pk/187](http://www.maxim-ic.com/appnotes.cfm/an_pk/187), March 2002
- Maxim Inc. (2003), *1-Wire Communication with a Microchip PICmicro Microcontroller* on-line at [http://www.maximic.com/appnotes.cfm/an\\_pk/2420](http://www.maximic.com/appnotes.cfm/an_pk/2420), Sept. 2003
- Pardo A.; Camara L.; Cabre J.; Perera A.; Cano X.; Marco S.; Bosch J. (2006). Gas measurement systems based on IEEE1451.2 standard, *Sensors and Actuators B: Chemical*, Vol.116, Issues 1-2, pp. 11-16, July 2006
- Paret D.& Fenger, C. (1997). *The I2C Bus: From Theory to Practice*, Wiley, John & Sons, 1997
- Pfeiffer O.; Ayre A.; Keydel C. (2003) *Embedded Networking with CAN and CANopen*, Annabooks Publisher, 2003
- Phillips Semiconductor (2000). The I2C-BUS specification, Version 2.1, 2000, online at: [i2c2p.twibright.com/spec/i2c.pdf](http://i2c2p.twibright.com/spec/i2c.pdf). 2000

- Postolache O.; Postolache G.; Girão P. M. (2006). Non-invasive mobile homeostasis Instrument", *Proc IEEE International Workshop on Medical Measurements and Applications*, Benevento, Italy,.Vol. I, pp. 94 – 98, May 2006
- Postolache O.; Dias Pereira J. M.; Girão P. M. (2007). Optical Fiber Based Turbidity Sensing System, *Proc IMEKO TC19 Symp*, Iasi, Romania, Vol. II, pp. 71 - 75, September, 2007
- Ramos H.; Pereira J.M.; Viegas V.; Postolache O.; Girão P. M. (2004). A Virtual Instrument to Test Smart Transducer Interface Modules (STIMs), *IEEE Transactions on Instrumentation and Measurement*, vol. 53, no. 4, pp. August 2004
- SD Association (2009). SD Card Compact and Powerful, Ideal for Small Products, on-line at: <http://www.sdcard.org/developers/tech/sdcard/>, May 2009
- Song E. Y. & Lee K (2008). Understanding IEEE 1451-Networked smart transducer interface standard, *IEEE Instrumentation & Measurement Magazine*, Vol.11, No. 2, pp.11-17, February 2008
- Ulivieri N.; Distanto C.; Luca T.; Rocchi S.; Siciliano P. (2006). IEEE1451.4: A way to standardize gas sensor, *Sensors and Actuators B: Chemical*, Volume 114, Issue 1, pp. 141-151, March 2006
- Yurish, S. & Gomes M. (2003). *Smart Sensors and MEMS*, NATO Science Series Mathematics, Physics and Chemistry, September 2003

IntechOpen



## **Data Storage**

Edited by Florin Balasa

ISBN 978-953-307-063-6

Hard cover, 226 pages

**Publisher** InTech

**Published online** 01, April, 2010

**Published in print edition** April, 2010

The book presents several advances in different research areas related to data storage, from the design of a hierarchical memory subsystem in embedded signal processing systems for data-intensive applications, through data representation in flash memories, data recording and retrieval in conventional optical data storage systems and the more recent holographic systems, to applications in medicine requiring massive image databases.

### **How to reference**

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Octavian Postolache, Pedro Silva Girao and Jose Miguel Dias Pereira (2010). Non-Volatile Memory Interface Protocols for Smart Sensor Networks and Mobile Devices, Data Storage, Florin Balasa (Ed.), ISBN: 978-953-307-063-6, InTech, Available from: <http://www.intechopen.com/books/data-storage/non-volatile-memory-interface-protocols-for-smart-sensor-networks-and-mobile-devices>

**INTECH**  
open science | open minds

### **InTech Europe**

University Campus STeP Ri  
Slavka Krautzeka 83/A  
51000 Rijeka, Croatia  
Phone: +385 (51) 770 447  
Fax: +385 (51) 686 166  
[www.intechopen.com](http://www.intechopen.com)

### **InTech China**

Unit 405, Office Block, Hotel Equatorial Shanghai  
No.65, Yan An Road (West), Shanghai, 200040, China  
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元  
Phone: +86-21-62489820  
Fax: +86-21-62489821

© 2010 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](https://creativecommons.org/licenses/by-nc-sa/3.0/), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.

IntechOpen

IntechOpen