Keywords: plug and play, plug n play, electronic memory label, circuit board identification, PCB identification, 1-wire devices

APPLICATION NOTE 178

Printed Circuit Board Identification Using 1-Wire® Products

By: Bernhard Linke, Principal Member Technical Staff
Jan 06, 2010

Abstract: There are two ways to identify a printed circuit board (PCB). The 'minimalist' approach is based on devices using ROM technology. The 'nameplate' approach requires user-programmable NV memory to store the PnP-related information. The most obvious way to implement Plug-and-Play capabilities is by means of a memory chip that holds the required information and does not lose it if power is switched off. 1-Wire chips for PCB identification are available in ROM, EEPROM, and EPROM (One-Time-Programmable) technology.

Introduction

Anyone who has ever tried to add a modem- or network-card to a PC knows that such an undertaking may produce unwanted results and can take more time than expected. To achieve cost savings from high volume production, the add-in board is designed to be logically compatible to a large variety of motherboards from various vendors. This flexibility makes it necessary to configure the board for its operating environment before it is ready to function. To perform the configuration one needs two pieces of information: sufficient knowledge of the system that the board is expected to reside in and knowledge of the board itself. Once the suitable settings are determined one needs to set jumpers or flip switches on the board. This completes the tricky part of the hardware installation.

As long as such upgrades or changes are made routinely and by trained personnel, they are time-efficient and do not involve much risk. Every time one has to deal with an unfamiliar product, a new learning cycle begins with studying the associated documentation (which may not be well-written) and a phase of trial and error, which may result in damage to the product or the system it is supposed to work in. In 1994, the concept of "Plug-and-Play" (PnP) was introduced to get out of this awkward situation, particularly with PCs. In order to function, each PnP device must have all of the following capabilities:

1. It must be uniquely identified.
2. It must state the services it provides and the resources it requires.
3. It must identify the software driver that supports it.
4. It must allow software to configure it.

(source: Lee Fisher, "Win32 Application Support for Plug and Play"[Online] 18 January 1995

(1995-01-18))

Although these requirements were originally specified for PC products, all electronic systems that consist

of more than one device (or circuit board) do benefit from Plug and Play capabilities.

The most obvious way to implement PnP capabilities is by means of a memory chip that holds the required information and does not lose it if power is switched off. In some cases, suitable memory is already on the board (e.g., inside a microcontroller or as a byte-wide memory chip or serial EEPROM). In other cases or if the entire memory is already assigned to other functions, an additional memory chip is needed. Due to their smaller physical size and minimal additional I/O requirements, serial memories are preferred over byte-wide memories. Of all serial memories, devices with 1-Wire interface are easiest to interface because they require only a single data line plus ground reference. Most 1-Wire devices do not even have a V_{CC} pin; they draw the energy for operation right from the data line. All 1-Wire devices have a unique registration number, which allows for traceability and performs as an address if multiple 1-Wire devices (not just memories) are connected in parallel to form a 1-Wire net. This feature is not found with other serial devices. Parts of the registration number can be customized, which turns a 1-Wire device into a custom chip that authenticates the board and makes it impractical to clone.

Spare memory bytes not used for PnP functions are available to enable/disable some of the resources of the circuit board, opening the door to license management rather than changing hardware for upgrades. Depending on storage capacity and chip technology, the memory could hold manufacturing data for quality control, warranty information, and repair history. Having such information readily accessible reduces the time to repair.

Device Technologies

1-Wire chips for printed circuit board identification need to maintain stored data independent of any power supply. This limits the choice to devices that are manufactured in nonvolatile (NV) technologies such as ROM, EEPROM, and EPROM (**Table 1**). The lack of a window in the plastic package or the mounting technique of chip scale packages with the active side towards the circuit board converts EPROM chips to One-Time-Programmable (OTP) memories.

Table 1. 1-Wire Devices For Circuit Board Identification

Part #	Technology	Memory	Appl. Type	Voltage	Packages	Extras
DS2401	ROM	64 bits	Minimalist	2.8V to 5.5V	Plastic, CSP	-----
DS2430A*	EEPROM	32 + 8 bytes	Nameplate	2.8V to 5.5V	Plastic, CSP	8 bytes OTP
DS2431	EEPROM	128 + 7 bytes	Nameplate	2.8V to 5.25V	Plastic, CSP	Write protection
DS2432	EEPROM	128 + 16 bytes	Nameplate	2.8V to 5.5V	Plastic, CSP	Challenge & response
DS28E01	EEPROM	128 + 17 bytes	Nameplate	2.8V to 5.25V	Plastic	Challenge & response
DS2433	EEPROM	512 bytes	Nameplate	2.8V to 5.5V	Plastic, CSP	-----
DS28EC20	EEPROM	2560 + 16 bytes	Nameplate	3.14V to 5.25V	Plastic	Write protection
DS2406	OTP EPROM	128 bytes	Nameplate	2.8V to 5.5V, 12V write	Plastic, CSP	2 controlled I/O pins
DS2502	OTP EPROM	128 bytes	Nameplate	2.8V to 5.5V, 12V write	Plastic, CSP	Write protection
DS2505	OTP EPROM	2048 bytes	Nameplate	2.8V to 5.5V, 12V write	Plastic	Write protection
DS2506*	OTP EPROM	8192 bytes	Nameplate	2.8V to 5.5V, 12V write	Plastic	Write protection

*The DS2506 and DS2430A are no longer recommended for new designs.

Circuit Board Identification Overview

As indicated in column APPL. TYPE of Table 1, there are two ways to identify a printed circuit board. The **minimalist** approach is based on devices using ROM technology. The **nameplate** approach requires user-programmable NV memory to store the PnP-related information. **Table 2** shows the identification alternatives of both approaches. Reference numbers beginning with 1 indicate the minimalist approach. For the nameplate approach there are devices in EEPROM technology (reference numbers beginning with 2) and OTP EPROM (numbers beginning with 3). Within each of these groups, the reference letter A stands for generic, off-the-shelf parts. The letter B is used for parts that have a customized ROM. The ROM customization is explained in section *Minimalist Approach*. The letter C indicates protected EEPROM (2C). For more details on this see section *Nameplate Approach*.

With customized devices, the cloning protection is achieved by restricted supply. Clone manufacturers will not be able to buy the customized parts. To achieve protection with generic devices (excluding the DS2432 and DS28E01) it is necessary to manipulate the data, e.g., by means of encryption.

Common encryption methods (e.g., cipher block chaining) rely on a key ("secret") and an initialization vector. Embedding the unique ROM registration number of the memory chip in the initialization vector ensures that identical source data looks different for every individual device. This way, if data is copied

from one chip to another, the copy will not work in the application. Details about encryption algorithms and software are available on the Internet, for example at the National Institute of Standards and Technology website.

Table 2. Overview of Identification Alternatives

	Reference						
	1A	1B	2A	2B	2C	3A	3B
Approach	Minimalist	Minimalist	Nameplate	Nameplate	Nameplate	Nameplate	Nameplate
Identification	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Cloning Protection	No	Yes	With encryption	Yes	Yes	With encryption	Yes
Plug & Play	No	Limited	Yes	Yes	Yes	Yes	Yes
Automation	Low	Medium	High	High	High	High	High
Min. Order	N/A	> 10k parts	N/A	> 10k parts	N/A	N/A	> 10k parts
Typical Volume	Any	> 100k p.a.	Any	> 100k p.a.	Any	Any	> 100k p.a.

Minimalist Approach

The lowest-cost method of identifying circuit boards (1A) uses generic (non-customized) DS2401 silicon serial numbers and relies on a central database that associates the ROM registration number of the 1-Wire chip to an individual board. Due to lack of user-programmable memory, this approach does not support PnP.

A somewhat more costly alternative (1B) uses devices with a customized ROM registration number. This allows the customer (board manufacturer) to specify 8, 12, or 16 bits of the 64-bit ROM content (**Figure 1**). Having control over some of the ROM bits allows for limited PnP compliance. A central database is still required to associate the ROM content to an individual board. For PnP to function, the circuitry on the board must be able to read the 1-Wire device and be able to forward the information to host CPU.

Since the customized chips are only sold to the original customer and its authorized agents, they are controlled products that are not available for clone manufacturers. To take advantage of this, the firmware must be able to distinguish between legitimate customized chips and generic chips or customized chips made for other customers.

Generic devices (1A) are easy to order and, in small quantities, are usually shipped from stock. Customized devices (1B) are manufactured on order and have a lead-time of 13 weeks. Contact Maxim or its sales organization for a copy of the registration form.

The typical chip for the minimalist approach is the DS2401.

Minimalist Approach—No User Memory—Comparison Table

1A	Generic Device
Method	Database lookup based on registration number.
Precondition	Database that uses registration number as key to access detailed product information.
Administration	Enter registration number of every board into database and associate (reference to) detailed product information.
Cloning Protection	None
Advantages	Lowest chip cost.
Disadvantages	Database automation is limited to the tester reading the registration numbers of a batch of boards and then transmitting them as a list to the database; the association to the product information is created manually. Requires access to (remote) database to read a board's detailed information.

1B	Customized Device
Method	Database lookup based on registration number. Board types or board versions are distinguished by different codes in the customer-specified section (subfield) of the registration number.
Precondition	Database that uses registration number as key to access detailed product information.
Administration	Enter registration number of every board into database and associate (reference to) detailed product information. A portion of the product information can automatically be identified by the content of the customer-specified bits (subfield).
Cloning Protection	Yes
Advantages	The chip is less costly than memory chips for the nameplate approach. Less prone to human error, provided that the right chip is mounted on the board. If different products or product versions require different test programs, the name of test program can be used to verify that the right chip was mounted on the board.
Disadvantages	Database automation is limited to the tester reading the registration numbers of a batch of boards and then transmitting them as a list together with test program name to the database. Requires access to (remote) database to read a board's detailed information. Logistics for stocking and using different customized chips.

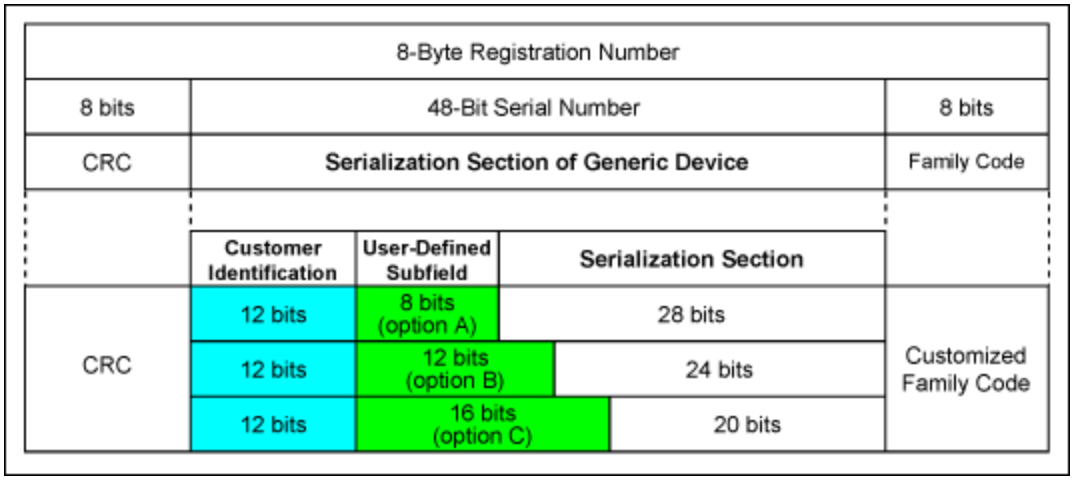
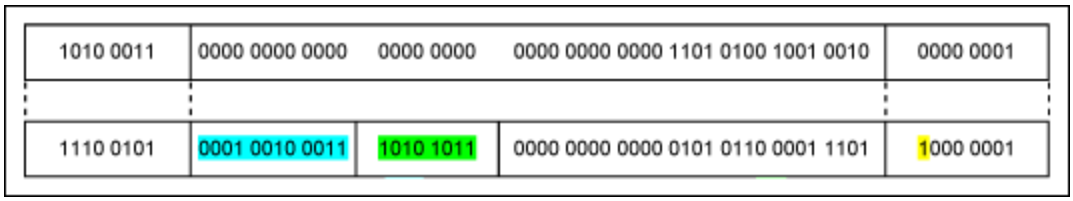


Figure 1. Generic and customized ROM registration number.
Option A: 268 million serial numbers per subfield value
Option B: 16.7 million serial numbers per subfield value
Option C: 1.04 million serial numbers per subfield value

Examples

Generic DS2401
Hex representation of sample registration number: A3 00 00 00 00 D4 92 01



Customized DS2401, customer ID 123 hex, option A, subfield contents AB hex
Hex representation of sample registration number: E5 12 3A B0 00 56 1D 81
The most significant bit of the customized family code is always 1.

From reading a customized ROM one cannot tell what the selected option is. However, for a particular customer identification code and device, the option remains the same for all variations of the user-defined subfield content.

Nameplate Approach

All devices suitable for the nameplate approach have user-programmable NV memory to store the PnP-related information. For the nameplate approach it is essential to list the data items that are required for PnP to function, to define a way to identify each data item, and to decide on a format in which the (numerical, logical, text) value of each data item is represented. The format may be customized for a particular application to save space or one may use an industry-standard, but larger, format like XML (extendible markup language). Further data items can be added to the list, provided that they can be identified and a format has been selected. The data items then can be stored one after another in any sequence in the memory of the board identification chip. Alternatively, one can embed the data between a length byte at the beginning and a CRC check at the end, and store it as a data file. More information on a file format for 1-Wire devices is found in application note 114, "1-Wire File Structure."

Depending on the memory size (in bytes) there may be space for other information, such as

manufacturing data. Since the information is stored in a chip on the board, there is no real need to access a central database to get information about the board. However, it may still be good practice to maintain such a database to store information that does not fit into the memory chip or is intentionally kept away from public access (e.g., bill of material, names of component vendors). As with the minimalist approach, the 64-bit ROM registration number is the key to that database.

The price of the identification chip depends on the storage capacity (memory size) and the technology it is manufactured in. As shown in an example later in this document, the information that is essential for PnP fits easily into 128 bytes or even less. If storing of additional information is considered, the 128 bytes of the small memory chips may not be enough. Conversely, with a 2048-byte chip there is plenty of room to store an entire bill of material of more than 150 items in a compact format. Using the less storage-efficient XML tagging format, 2048 bytes may still be enough to include a bill of material of up to 50 items.

Suitable memory chips for the nameplate approach are manufactured in two technologies: EEPROM and OTP-EPROM. In either technology there are generic chips (referenced as 2A for EEPROM and 3A for EPROM) as well as ROM-customized chips (2B and 3B, respectively). The ROM customization is the same as described for the minimalist approach (see Figure 1) and the same restrictions and recommendations apply.

The protected EEPROM (2C) is customized by the user by installing an 8-byte secret in the device. Once installed, the secret—in contrast to a password—is never exposed on the 1-Wire line. When writing to a protected EEPROM, the host computer needs to send a 160-bit message authentication code, which is calculated from the secret and other data. When reading the EEPROM, the device can also generate an authentication code that allows the master to verify whether the chip knows the correct secret without ever exposing the secret itself. See application note 3675, "Protecting the R&D Investment—Two-Way Authentication and Secure Soft-Feature Settings," for more information on the special uses of protected EEPROMs.

Generic devices (2A, 3A, and 2C) are easy to order and in small quantities are usually shipped from stock. Customized devices (2B, 3B) are manufactured on order and have a lead-time of 13 weeks. Contact Maxim or its sales organization for a copy of the registration form.

The smallest EEPROM chip for the nameplate approach (2A, 2B) is the DS2430A. As a special feature, this chip, in addition to 32 bytes re-programmable memory, has an 8-byte one-time programmable register. The DS2431, DS2433, and DS28EC20 are organized as 4, 16 or 80 pages of 32 bytes each. The DS2431 and DS28EC20 support write protection and OTP-EPROM emulation. The DS2432 and DS28E01 protected EEPROM (2C) can store 128 bytes (4 pages of 32 bytes) of application data and one 8-byte secret. They have several more registers for control functions. In EPROM technology (3A, 3B) there are four devices to chose from: DS2502 (128 bytes), DS2505 (2048 bytes), DS2506 (8192 bytes), and DS2406 (128 bytes). All these devices are organized in memory pages of 32 bytes. In addition to memory, the DS2406 has two extra pins that under software control can function as open-drain output (actuator) or as digital input (sensor).

Nameplate Approach—EEPROM—Comparison Table

2A	Generic Device
Method	User-programmable memory chip stores relevant PnP data on the circuit board. **Optional**: Database lookup based on registration number for additional data.
Precondition	Definition of PnP data, additional data (as desired), data representation, and formatting. Board tester that programs memory chip at final test. **Optional**: Database to access additional product information.
Administration	Ensure that the board tester uses the correct software. **Optional**: Create link between tester and database to store board registration number, copy of chip data, and board test results.
Cloning Protection	No, unless data is encrypted.
Advantages	Highly automated database. If the memory chip was programmed with incorrect data, it can be reprogrammed.
Disadvantages	More costly than minimalist approach. Unless encrypted or write protected, data is not safe from unauthorized changes.

2B	Customized Device
Method	(Same as 2A)
Precondition	(Same as 2A)
Administration	(Same as 2A)
Cloning Protection	Yes, encryption is optional.
Advantages	(Same as 2A)
Disadvantages	More costly than generic device. Unless encrypted or write protected, data is not safe from unauthorized changes.

2C	Protected Device (Customization by User)
Method	(Same as 2A)
Precondition	Board tester, which at final test installs a secret in the memory chip and programs it with the desired data. (Otherwise same as 2A)
Administration	Ensure that the board tester uses the correct software and the correct secret. (Otherwise same as 2A)
Cloning Protection	Yes, encryption is optional.
Advantages	Data can be read but not altered without knowing the secret. The secret can be write-protected. (Otherwise same as 2A)
Disadvantages	More costly than minimalist approach.

Nameplate Approach—OTP EPROM—Comparison Table

3A	Generic Device
Method	(Same as 2A)
Precondition	Board tester with 12V capability that programs memory chip at final test. (Otherwise same as 2A)
Administration	(Same as 2A)
Cloning Protection	No, unless data is encrypted.
Advantages	Highly automated database. If the memory chip was programmed with incorrect data, incorrect data can be invalidated and new data can be written to an unused memory section (if available). Data can be write-protected in blocks of 32 bytes.
Disadvantages	More costly than minimalist approach. Requires 12V for writing to the memory chip.

3B	Customized Device
Method	(Same as 2A)
Precondition	(Same as 3A)
Administration	(Same as 2A)
Cloning Protection	Yes, encryption is optional.
Advantages	(Same as 3A)
Disadvantages	More costly than generic device. Requires 12V for writing to the memory chip.

Nameplate Design Considerations

The electronic nameplate has to include all data items that are required for PnP, a means to identify each data item, and a value for each item in a suitable format. Space permitting, there may be additional information added to the nameplate. As with metal or paper nameplates, one might want to include the company's name and address information. Since OTP-EPROM bits can only be programmed from 1 to 0, a format definition that allows changing programmed data under such constraints is desirable.

The following **sample design concept** meets these requirements. The nameplate begins with an ASCII string of the company name and website address, followed by a 00 byte as delimiter. All subsequent data is organized as groups. Each group contains data items for a particular purpose. Each data item has an identification number (1 to 255) that is unique within the group. The length of all data item values within a group is the same. Each group begins with an introductory byte that identifies the group number (1 to 15) and the value length (1 to 15 bytes) that applies to the group. The group number is stored in the higher nibble of the introductory byte, leaving the lower four bits to specify the value length. After the last value of a group there is a 00h byte as delimiter.

The viability of this definition is demonstrated on the MxTNI™-board, a miniature web server of the size

of a SIMM board. The MxTNI server consists of approximately 65 components, including a microprocessor. Some components and jumpers are installed only on certain versions of the board. The data is organized into two groups, manufacturing data and PnP data, as shown in **Table 3**.

Table 3. Sample Nameplate Contents

Group	Function	Items	Ident. #	Length	Format
1	Manufacturing Data	Product name Part number Hardware revision Date code/lot number	1 2 3 4	12 bytes each	ASCII
2	PnP Data	RAM chip select RAM size Ethernet chip presence Ethernet chip to SIMM connection Non-volatizer presence	1 2 3 4 5	1 byte each	Numeric Numeric (Mbits) Boolean Boolean Boolean

The manufacturing data includes everything that is typical for integrated circuits. All this information can be represented as ASCII text of no more than 12 bytes for each item.

The PnP data consists of five parameters: 1) number of RAM chip select lines, 2) size of each RAM chip, 3) presence of Ethernet chip, 4) connection of Ethernet chip to SIMM pins, and 5) presence of nonvolatizer. The first two of these parameters have a numeric value; the three others are "Boolean", which means they can be either true or false. Although these Boolean parameters could be merged into three bits of a single byte, this is not done in this example. Instead, a full byte FFh for "true" and 00h for "false" is used. The full nameplate including company name will look like this:

```
(C)MAXIM www.ibutton.com<00>
<1C><01>MxTNI SERVER <02>90-TINI1-512<03>REVB1 030900<04>0023B/117989<00>
<21><01><01><02><04><03><FF><04><FF><05><FF><00>
```

Information between angled brackets <> represents bytes in hexadecimal format. All the other information is plain ASCII text. The whole nameplate requires just 106 bytes, which easily fit in a DS2502 chip. If a bigger memory chip is chosen, one could include a full parts list or bill of material in the nameplate. For this purpose the parameter groups could be organized like this:

Group	Function	Length	Format
1	Manufacturing data	12 bytes each	ASCII
2	Plug-and-play data	1 byte each	(varies)
3	Batteries	9 bytes each	ASCII
4	Capacitors	9 bytes each	ASCII
5	Diodes	9 bytes each	ASCII
6	Transistors	9 bytes each	ASCII
7	Resistors	9 bytes each	ASCII
8	ICs part 1	9 bytes each	ASCII
9	ICs part 2	12 bytes each	ASCII
10	Crystals	9 bytes each	ASCII
11	Circuit board	9 bytes each	ASCII

Applied to the same example board, all this information would fit into less than 1kB. For this estimate, a proprietary part numbering system was chosen that assigns a 12-character part number to anything including component parts. Since the first three characters of the component parts are always the same they were omitted, reducing the length to nine characters. The part numbers of the integrated circuits in group nine start with a different code. For this reason the full length of 12 was used.

Since its identification number precedes each item within a group, the sequence of items is not critical. The sequence of groups can be changed also, because each group is identified by its introductory byte. This flexibility allows invalidating a group or an individual parameter within a group and to redefine it later. This is of particular interest with OTP-EPROM devices. Changing the group number to 0 invalidates a group; don't change the bits that specify the length of parameters within the group. A new copy of the whole group is then appended after the last parameter group. To invalidate and redefine a parameter, change the parameter number to 00 at the original location and append a new version of the parameter group with the new parameter value. It is not necessary to also include those parameters of the group that have not changed. Invalidation and redefinition of values or groups is not necessary with EEPROM devices. If anything changes, the entire data can be re-written.

The sample design concept described above is one way to create a nameplate. The mix of binary and ASCII data minimizes memory consumption, but is not standardized. Less memory-efficient but standardized is the XML format, which uses ASCII codes only. In XML format, the same example may look like this:

```
<NamePlate name="MxTNI SERVER">

     <Company> (C)MAXIM www.ibutton.com </Company>
     <ProdData>
     <PartNo> 90-TINI1-512 </PartNo>
     <Rev> B1 030900 </Rev>
     <Date> 0023B/117989 </Date>
     </ProdData>
     <Config>
     <MemChips> 1 </MemChips>
     <MemSize> 4Mbits </MemSize>
     <EthChip> true </EthChip>
     <EthSimm> true </EthSimm>
     <NonVol> true </NonVol>
     </Config>

</NamePlate>
```

This XML file is 380 bytes long; formatted in 1-Wire file structure as described in application note 114, this XML nameplate will fill 15 of the 16 memory pages of a DS2433 chip. The size of the XML file can be reduced by choosing shorter abbreviations for each parameter.

Obtaining a Conventional Board Serial Number

Despite having an electronic board identification or nameplate, there may be a desire to also print a conventional, human-readable serial number on the circuit board. A hexadecimal form of serial number is already included in the 64-bit ROM registration number. If the nameplate approach is chosen, one can embed a decimal serial number in the design of the nameplate. Altogether there are three cases to be distinguished:

1. **Generic Silicon, Minimalist Approach (Reference 1A)**
   To get a conventional serial number, take the 48-bit serial number (see Figure 1) and convert it into a decimal number. Since the DS2401, the typical chip of the minimalist approach, is produced in large quantities the resulting decimal number is 9 or 10 digits long.

2. **Customized Silicon, Minimalist and Nameplate Approach (References 1B, 2B, and 3B)**
   To get a conventional serial number, take the content of the serialization section and convert it into a decimal number. As shown in Figure 1, the serialization section of customized parts does not exceed 28 bits. This limits the length of the serial number to 9 decimal digits (option A) and 8 digits under option B. Depending on the forecasted demand, a 7-digit printed serial number may be acceptable. When defining the format of the printed serial number one should decide on whether and how to include the user-defined subfield. The hexadecimal format functions in any case; depending on the subfield content, one could take the digits as binary-coded decimal (BCD) and print them straight as decimal numbers. When printing the whole ROM registration number or parts of it in hexadecimal format, the CRC should be printed to the left and the family code to the right. Example: 2500206600 7A8081. Omitting family code and CRC, leaves 002066007A80, the customer identification and a combination of user-defined subfield and serial number. Knowing the ROM option (B in this example), one can tell that the actual serial number of this device is 007A80 in hexadecimal or 31360 decimal. In this example, the subfield can be seen as decimal as well as hexadecimal.

3. **Generic Silicon, Nameplate Approach (References 2A, 2C, and 3A)**
   The preferred way to get a conventional serial number is to include it in decimal form in the nameplate. This type of serialization needs to be managed by the software that controls the board tester, a task that may not be as easy as expected. Since the serial number by definition needs to be different in every part, gang-programming the identification chips before they are mounted on the circuit board is not an option. Instead of including a serialization in the nameplate, one could fall back to the same procedure as in case *1) Generic Silicon, Minimalist approach*.

Connection to Main Circuit

To get the PnP data to the host CPU, the board identification chip must be connected to the circuitry on the board. The interface shown in **Figure 2** fits to a bi-directional port. Due to its open-drain characteristic, port 0 of the 8051-compatible microcontroller is the preferred choice to drive a 1-Wire line.

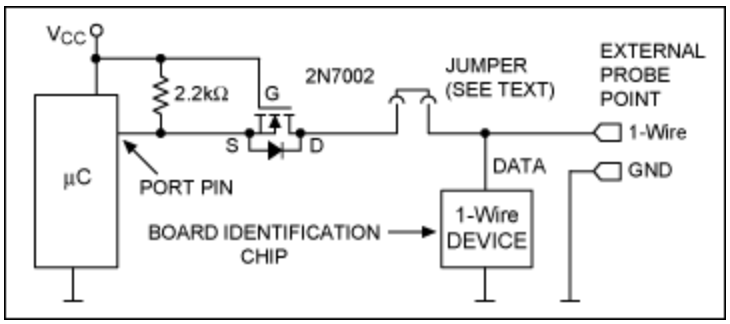


Figure 2. Dual-Read Mode Interface.

During normal operation, the port pin is pulled high to V_{CC} through the 2.2kΩ resistor. The parasitic diode inside the MOSFET ensures that the voltage at the 1-Wire device gets close to V_{CC} even if V_{GS} is zero and the MOSFET is off. If the microcontroller puts a logic 0 (0V) on the port pin, V_{GS} is high enough to make the transistor conducting. This way the voltage at the port pin is propagated through the transistor to the 1-Wire device. To transmit a 0-bit to the microcontroller, the 1-Wire device shorts the 1-Wire line to Ground for a short time. During that time current flows from V_{CC} through the resistor and through the parasite diode of the MOSFET through the 1-Wire device. Despite the voltage drop between source and drain of the MOSFET, the voltage on the port pin remains in the permissible range to safely be recognized as logic 0. To transmit a 1-bit, the 1-Wire device remains in high impedance. As a result,

the 1-Wire line is pulled to V_{CC} by the resistor as soon as the microcontroller stops pulling low. If the board is not powered, one can read the 1-Wire chip through the external probe point. The 2.2kΩ resistor ensures that V_{GS} of the MOSFET is 0, which keeps the transistor non-conducting, preventing current flow from the external 1-Wire reader into the other circuitry on the board. The jumper shown in Figure 2 is only required with EPROM devices. It must be removed to protect the MOSFET from the 12V pulses, which the tester applies to program the EPROM chip with the nameplate data. After the programming is completed, the jumper must be reinstalled to give the microcontroller access to the identification chip.

The concept of the circuit in Figure 2 can be applied to other types of serial interface; however, a decoupling transistor will be necessary for the data line as well as clock and control signals. It may also be necessary to de-couple the power line of the serial memory chip. Instead of the simple 2-pin probe point, a 4- or 5-pin connector may be required. As a further disadvantage, non-1-Wire serial memory devices don't come with a built-in customizable serialization.

Real World Example

The MxTNI-board, which was chosen as an example for the nameplate design considerations, actually includes an electronic nameplate. Due to its Ethernet interface, each of these boards needs an Ethernet Address. The easiest way to implement such addresses in silicon is by means of the DS2502-E48, where one page is preprogrammed with an Ethernet address and administrative data, which leaves three memory pages available for the nameplate at **no extra cost**.

The actual MxTNI nameplate design is somewhat different from the sample design concept. This is acceptable as long as all relevant data is included, identifiable, and has a value. The MxTNI nameplate contains seven data entities or tags, as shown in **Table 4**.

Table 4. MxTNI Nameplate Contents

Tag Name	Tag #	Tag Length	Tag Format	Possible Tag Values
TAG_TINI_HW_VERSION	01	35 bytes	ASCII	(text string)
TAG_DATE_CODE	02	4 bytes	numeric	(UNIX 32-bit time format)
TAG_RAM_CHIP_SELECT_CONFIG	03	1 byte	numeric	1, 2
TAG_RAM_SIZE_EACH_CONFIG	04	3 bytes	numeric	0 to 16777216 bytes
TAG_SMC_NOT_PLACED	05	1 byte	numeric	0 = Ethernet available 1 = Ethernet unavailable
TAG_SMC_PORT_PINS_CONNECTED	06	1 byte	numeric	0 = pins not connected 1 = pins connected
TAG_NONVOLATIZER_NOT_PLACED	07	1 byte	numeric	0 = non-volatizer used 1 = non-volatizer not used

Each tag is identified by a number that is followed by a length byte and the tag value. Beginning with tag number 1, the seven tags are concatenated to form the MxTNI configuration data. The configuration data is reformatted into data packets of 32 bytes each, matching the size of a memory page. Each packet begins with a length byte and ends with a CRC16. The resulting data pattern including the preprogrammed data of a sample device is shown in **Table 5**.

Table 5. Sample MxTNI Identification Chip

Preprogrammed Section																
Address																
0000	0A	29	11	00	00	B8	93	00	35	60	00	68	59	FF	FF	FF
0010	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
Nameplate Section																
Address																
0020	1D	01	23	44	53	20	54	49	4E	49	20	4D	6F	64	65	6C
0030	20	33	39	30	20	52	65	76	20	44	20	44	53	54	F3	3C
0040	1D	49	4E	49	31	2D	35	31	32	02	04	A0	3E	36	3A	03
0050	01	01	04	03	00	00	08	05	01	00	06	01	01	07	C5	01
0060	02	01	00	AF	AF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
0070	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF

Color Code Legend

Data Packet Length Byte	Constant data (IEEE® company Identification)
Data Packet CRC	(no color = unused byte)
Data Pattern Identification Number	Nameplate Parameter Number and Parameter Length
Serialization	Nameplate Parameter Value

To read the nameplate, one needs to identify the tag numbers that are followed by the tag size. As many subsequent bytes as indicated by the tag length are the tag value. The next byte again is a tag number, and so on. **Table 6** shows the data of Table 5 in decoded form. When decoding the tag values, it is important to know that multi-byte numbers are stored with the least significant byte at the lower address ("Little Endian" byte order). To restore the natural sequence of digits one needs to write down the bytes (not digits) in reverse order (i.e., from right to left as shown in the "Explanation" column). Text is stored in its natural sequence (i.e., the first character at the lower address).

Table 6. Nameplate Decoded

	Tag #	Value	Explanation
Preprogrammed Section	(N/A)	29 11 00 00	Data pattern ID# 00001129
	(N/A)	B8 93 00	Ethernet extension ID# 0093B8 (serialization)
	(N/A)	35 60 00	Ethernet company ID# 006035 (constant data)
Nameplate Section	01	(text)	Text = "DS MxTNI Model 390 Rev D DSTINI1-512"
	02	A0 3E 36 3A	3A363EA0 = December 12, 2000, 15:05:04 hours (see note)
	03	01	1 chip select
	04	00 00 08	08 00 00 = decimal 524288 bytes (512K)
	05	00	SMC Ethernet chip is available
	06	01	SMC port pins are connected to SIMM edge connector
	07	00	Non-volatizer is used

NOTE: The date code 3A363EA0 represents 976633504 seconds after 0 hours UTC of January 1st, 1970. This time format assumes that a day is exactly 24 hours. The conversion from seconds to time of day therefore is very straightforward. The computation of the date includes the leap year compensation and is more complex. For a description of the algorithm to convert from conventional time and date to seconds and vice versa see application note 517, "DS1371/DS1372/DS1374 32-Bit Binary Counter Time Conversion."

Since the configuration data is packetized using a length byte at the beginning and a CRC16 at the end, it is not possible to invalidate a parameter (e.g., to update the nameplate after a second memory chip is installed). This would have been possible without packetizing. This packetizing, however, protects the data in the chip against memory bit errors. The CRC16 would not match if a memory bit had flipped. To safeguard against bit errors during communication, all 1-Wire EPROM chips support a read function that automatically generates a CRC based on memory data and transmits it after the end of a memory page.

Related Parts		
DS2401	Silicon Serial Number	Free Samples
DS2406	Dual Addressable Switch Plus 1Kb Memory	Free Samples
DS2411	Silicon Serial Number with V_{CC} Input	Free Samples
DS2431	1024-Bit 1-Wire EEPROM	Free Samples
DS2432	1Kb Protected 1-Wire EEPROM with SHA-1 Engine	Free Samples
DS2433	4Kb 1-Wire EEPROM	
DS2502	1Kb Add-Only Memory	Free Samples
DS2502-E48	48-Bit Node Address Chip	Free Samples
DS2505	16Kb Add-Only Memory	Free Samples

DS2506	64Kb Add-Only Memory	
DS28E01-100	1Kb Protected 1-Wire EEPROM with SHA-1 Engine	Free Samples
DS28EC20	20Kb 1-Wire EEPROM	Free Samples

More Information
For Technical Support: http://www.maximintegrated.com/support
For Samples: http://www.maximintegrated.com/samples
Other Questions and Comments: http://www.maximintegrated.com/contact

Application Note 178: http://www.maximintegrated.com/an178
APPLICATION NOTE 178, AN178, AN 178, APP178, Appnote178, Appnote 178