



PCT/DE95/00740

Processes to render microcontrollers secure

The invention relates to a process to secure microcontroller programs with external memories against being read and to secure microcontroller assemblies against duplication.

In many applications a multitude of microcontrollers are frequently required
5 and every microcontroller should as far as possible be unique and the contents of the microcontroller should not be accessible to third parties. An example of an area of application is the area of identification and encoding, in which special internal codes are needed but where serial production of closed circuits in EEPROM technology is not possible due to cost considerations.
10 Frequently the company with the know-how must place the production with external producers and in such cases the necessary transfer of information enables the production company to produce more or very similar components on its own account without the knowledge of the client.

In the known state of the technology to date there is a process and a circuit
15 configuration known to solve a similar task, described in publication EP 0 114 522. The aim of the process is solely to render the copying of the ROM component difficult by incorporating not just a simple ROM module but a module with encoding and decoding unit with a key memory. In this manner the information required by the microcontroller is supplied in
20 encoded form. The microcontroller requires a coding and decoding unit and a key field in order to process the information into legible code or plain text. In the initialising phase the microcontroller transmits a key to the ROM with which the information is then transmitted in encoded form. This is to ensure that no user can read the information in plain text.

25 The disadvantage of this process is that it only offers protection against the copying of the contents of the ROM by non-professional users. The costs associated with such a solution are also considerably higher than the costs of a standard ROM and the encoding and decoding unit must be installed in



PCT/DE95/00740

the ROM module as well as in the microcontroller module. Tapping the transmitted key also represents another point of attack. With knowledge of the encoding and decoding technology, which for reasons of costs cannot be very complex, duplication of the ROM is even possible for the non-professional user. The kind of encoding algorithm can also only be altered by altering the whole microcontroller system. This restricts the use of the modules to those compatible with the encoding system in the microcontroller and in which the password produces the correspondingly encoded text.

- 10 For professional users, including the producer of the module, producing such a pirate copy presents no problems. All that needs to be done is to copy the entire ROM and the circuit structure.

Furthermore there is a known process according to publication EP 0449256A2 in which programs 5 and data are stored in encoded form and are decoded when required by using the key stored in a second memory 7 using a coding and decoding device. The decoding unit is located between the CPU and the data bus and the CPU. The decoder comprises AND and OR gate arrays which in case of access to the data and program memory utilise the key from the key memory to decode the necessary data in real-time and feed it to the CPU. The decoding algorithm is determined by the hardware structure of the decoder. The key for the programs is also located in memory 5 and is transmitted to memory 7 when the system is started.

The disadvantage of the process is that such a microcontroller with external memory can be freely reproduced at all times and provides no protection against the producer of such microcontroller subassemblies. In addition to a standard microcontroller, an encoding and decoding unit is required. The decoder must be able to decode commands in real-time by using the code from memory 7. This means that the decoding unit may not be too complex because otherwise the time required to decode the commands may reduce the performance of the microcontroller. In addition, the microcontroller



PCT/DE95/00740

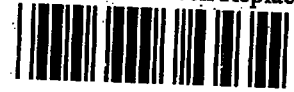
surface requirement increases in proportion to the complexity of the decoding. An additional disadvantage is the fact that the key for the program memory 5 is located in the program memory. A producer of microcontrollers is thus in a position to decode the encoded program module and gain access to the know-how of the client.

Further claims are related merely to the different methods of protecting internal ROM contents against unauthorised reading, as stated in the publication EP 468535A2. Here a process is presented whereby an internal ROM 4 and input and output port 5 and 6 are connected with a CPU via a bus and the contents of the ROM are only output via the output port after part or all of the contents of the internal ROM have been input into the input port. Only in the widest interpretation does such a process constitute an innovative solution.

The solution in publication EP 536943A2 describes a data security configuration in which an external ROM memory 24 linked to a programmable logic unit 11 not only safeguards the data against pirate copies by means of encoding but also the external ROM memory is firmly and unchangeably assigned to the logic unit.

The disadvantage of this solution is that in the production process protection against the producer of the unit is not guaranteed. The producer may manufacture any number of copies and embark on arbitrary serialisation of the system without the knowledge of the client. The safety of the system may therefore be undermined through the employees of the producer or even those of the contracting company.

Furthermore it is not possible to load large programs of variable size subsequently from the external memory. The external memory can only as large as can be mapped in the internal SRAM. This constitutes a limitation of use from the outset, especially in the highly variable area of microcontrollers.



PCT/DE95/00740

With the further development of computer technology and cryptography, high-quality encoding algorithms need to be used, especially in the area of security systems intended for payments and identification of persons. The deployment of a complex encoding system in line with this solution would
5 far exceed the budgeted costs and the complexity of the system.

The interchangeability of the decoding algorithms in line with this solution also requires a cost-intensive change of the hardware. Given the progressive development of computer technology and mathematics, algorithms which are regarded as safe today, may be deciphered by simple PC
10 programs tomorrow. The development of the algorithms must keep pace with the development of mathematics (cryptology) and computer sciences. Maintaining the variability of the algorithm is therefore a precondition for the long-term deployment of an economical module.

Accurate serialisation of the individual data security devices in line with this
15 publication is not possible as serial numbers can only be transmitted to the device which secures the data in plain text or using the key word. This means that the producer of the device is in a position to produce any number of securing devices bearing further serial numbers without the knowledge of the know-how holder. The security of production for electronic
20 components of this type is particularly important as, for economic reasons, such components are usually produced outside the know-how holding companies in 'low-wage countries'. At the same time they are used in financial transfer and personal identification systems. The temptation to produce additional copies and to commit plagiarism is great. Many video
25 encoding systems, for example, are cracked or copied because of the "information loopholes" in the individual manufacturing units of the producing companies which serve as sources for forgers and producers of replicas.

PCT/DE95/00740



In order to eliminate the above disadvantages of the prior art, prevent the unauthorised copying and multiplication of microcontroller modules with external memories and to prevent the deciphering and duplication of external microcontroller memories as well as for the supplementation of commands and information for microcontrollers, especially by a manufacturer of such modules, it is the task of the invention to create a process to permit the manufacture of any number of selectively identifiable microcontrollers combining simple circuitry and conventional assembly conditions in such a manner that it renders impossible the duplication of the modules or their contents by unauthorised persons, including the producer of such components.

In line with the invention, the task is solved through the characteristics given in Patent Claim 1. Preferred realisations of the invention are shown in the subclaims.

15 The advantages of the invention comprise the unified production of standard microcontrollers (modules) and the fact that any external programs of any length can be reloaded from the external memory in such a manner that the external program cannot be read or analysed by the producer of the assembly or by any third party. This process does not require any special hardware decoder, which means that the decoding algorithm can be changed at low cost. Also very demanding encoding processes can be implemented in the system as the external ROM is not decoded in real-time. The production of such microcontroller modules is possible without any special security precautions. As the process permits the subsequent loading of missing commands in the actual microcontroller, the possibility of the producer of the modules manufacturing excessive or subsequently duplicating more microcontrollers than have been ordered can be excluded.

The invention is explained in the following using Figure 1 to Figure 3.

The invention shall be explained in the framework of a sample application of an encoding microcontroller. In principle the process can be used for any

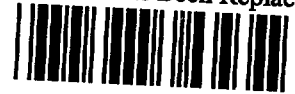


PCT/DE95/00740

other application, e.g. credit cards, telephone cards, etc.

The microcontroller module 1 shown in Fig. 1 consists of a microcontroller 2 in which an - internal CPU - 3 is connected via an - internal bus - 4 to an - internal ROM - 5, an - internal RAM - 6, a selection module 7, a - ROM interface module - 8 and an interface 9, and of an - internal programming signal - 10, an - external programming signal - 11 and an - external memory (PROM) - 12 which is connected with an - external bus - 13 via the - ROM interface module - 8 to the microcontroller 2.

When the microcontroller module 1 is started, the microcontroller 2 loads the contents of the - external memory (PROM) - 12 into the - internal RAM - 6. In this - internal RAM - 6 which can only be accessed via the - internal CPU - 3, the content of the - internal RAM - 6 is decoded into program and data with the help of the selection module 7, the contents of which can serve as a key and with the help of the data transmitted from the - external memory (PROM) - 12 which can also be used as key, using the decoding algorithm which is located in the - internal ROM - 5. The program in the - internal ROM - 5 may contain additional program parts, such as self-test procedures or routines for block data transmission of the contents of the - external memory (PROM) - 12. Once the program or the required program parts from the - external memory (PROM) - 12 have been converted from the specifically encoded form to a form which can be processed, the - internal CPU - 3 executes the commands contained in the - internal RAM - 6. The interface 9 can contain any communication parts of a standard microcontroller module, e.g. PIO, SIO or even external architecture similar to a bus. The microcontroller module 2 can communicate via this interface 9 with other units necessary to the completion of the task. The selection module 7 merely serves as storage for the microcontroller-specific internal key which can be utilised to decode the - external memory (PROM) - 12. By

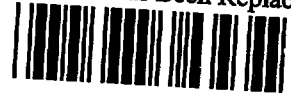


PCT/DE95/00740

virtue of the fact that every selection module 7 may contain a different random key, the - external memory (PROM) - 12 cannot be exchanged even if the uncoded original contents are the same. Because of this, no microcontroller module 1 can be copied by a non-entitled and unauthorised manufacturer by taking an - external memory (PROM) - 12 where he has knowledge of the contents and purchasing the microcontroller 2 in order to produce the microcontroller module 1. He cannot know the value of the selection module 7. The selection module 7 can be executed as a long PROM cell using a value derived from the programming by the - internal CPU - 3 which can only be read after this, or it can also be executed as an indistinct noise generator which on being enabled always supplies the same value but, due to the limited tolerance per circuit, always has a different statistical value. This can also occur in an area which has been irradiated and where, as a result of the changes in structure, different memory states have arisen, or it can occur in an area containing different bonding islands which have been bonded differently in the circuit bonding process. The generation of internal numbers is such a broad field that only a few examples could be shown.

The internal and external programming signal 10, 11 is for the once only programming of the - external memory (PROM) - 12 by the microcontroller module 2 with the assistance of the - ROM interface module - 8 via the - external BUS - 13. This process is shown in Figure 2 and Figure 3 below.

The flow diagram shown in Figure 2, consisting of a - primary ROM contents file - 14, composed of the - external ROM program code - 15, an - external variable part - 16 and the - external serial number part - 17 as well as of an encoding algorithm (which is known to the know-how holder) 18, a - known, user-specific production key - 19 and a generated production file 20.



PCT/DE95/00740

The function is characterised by the fact that the holder of the know-how produces the contents of the - primary ROM contents file - 14. This data is then transmitted in the - external ROM program code -15. The contents of the file can be organised as required. In this form of execution, which is not intended as a general example but relates to this specific application, the - primary ROM contents file - 14 consists of the - external ROM program code - 15, which also permits supplementation and modification of the program parts of the microcontroller 2 after production, and the- external variable part - 16 which serves to partially serialise the microcontroller 2 in such a manner that the number of the system can be ascertained and therefore it can be ensured that no microcontroller 2 has been duplicated, which in turn guarantees the uniqueness of the microcontroller module 1 without the know-how holder being able to produce microcontrollers at all. For example, in the - external variable part - 16 specific personnel data or specific key generation algorithms for the encoding microcontroller may be stored. This area can also be used for storing sequential serial numbers which can be accessed by an external microcontroller command and additionally sales and production control systems can be introduced. The allocation of the microcontroller module 1 to persons becomes possible with this selection.

After the know-how holder has defined the content of the - primary ROM contents file - 14, he encodes the - primary ROM contents file - 14 using the known encoding algorithm 18 and the user-specific key 19. In this way the production file 20 is generated. The - encoding algorithm known to the know-how holder - 18 can be provided to the know-how holder in form of a program e.g. on a PC. Even if the same conversion algorithm 21 is located in the microcontroller 2, several users (know-how holders) can still produce - primary ROM contents files - 14 for the same microcontroller 2 without loss of know-how, as the - known, user-specific production key - 19 can be



PCT/DE95/00740

selected differently for every user.

The production file 20 obtained in this way is now installed after establishing the memory programming tension via the interface 9 to the microcontroller 2 on completion of the components assembly.

- 5 The internal flow diagram shown in Figure 3 consists of the microcontroller module 1, the microcontroller 2, the selection module 7, the interface 9, the - external programming signal - 11, the - external memory - 12, the - known, user-specific production key - 19, the - generated production file - 20 and the conversion algorithm 21.
- 10 On completion of the physical production process (assembly, soldering, etc.), the microcontroller assembly is contacted via the interface 9 and the - external programming signal - 11 is linked to the - external memory (PROM) - 12. If required, the - internal programming signal - 10 can also be linked . After this, the - known, user-specific production key - 19 is transmitted to
- 15 microcontroller 2 via the interface 9. Following this transmission, the production file 20 is transmitted to microcontroller 2. Using the - known, user-specific key - 19 and the production file 20 and by applying the conversion algorithm 21, the - primary ROM contents file 14 - is decoded in the microcontroller 2. Following this decoding, depending on the application,
- 20 the content of the selection module 7 may be filled with the value of the decoded production file 20 to later serve as internal key. However, only one value from the decoded production file 20 may be used to recode the conversion algorithm 21. The conversion algorithm encodes the decoded production file 20 which is located in the - internal RAM - 6 using the selected
- 25 internal key from the production file 20 or from the selection module 7 or from a combination of both. The newly encoded data is permanently stored via the - ROM interface module - 8 in the - external memory (PROM) - 12. In this way a specifically encoded - external memory (PROM) - 12 is gener-



PCT/DE95/00740

ated without the code being known to the know-how holder or his staff. It is not necessary for the key (or the contents of the selection module 7) to be known. Thus a very high level of security is attained and the producer of the microcontroller module 1 cannot illegally manufacture the
5 microcontroller 2 for himself.



PCT/DE95/00740

Numeric reference system

1. microcontroller module
2. microcontroller
3. - internal CPU -
- 5 4. - internal bus -
5. - internal ROM -
6. - internal RAM -
7. selection module
8. - ROM interface module -
- 10 9. interface
10. - internal programming signal -
11. - external programming signal -
12. - external memory (PROM) -
13. - external bus -
- 15 14. - primary ROM contents file -
15. - external ROM program code -
16. - external variable part -
17. - external serial number part -
18. - encoding algorithm known to know-how holder -
- 20 19. - known, user-specific production key -
20. production file
21. conversion algorithm



PCT/DE95/00740

Patent claims

1. Method for rendering microcontrollers secure, according to which reading of the external microcontroller program and duplication of the microcontroller assembly is prevented, characterised by the fact that serial-
5 ised microcontrollers (2) are produced which cannot be copied by third parties and the issue of further serial numbers, particularly by the manufacturer of a microcontroller module (1), is prevented, and that highly sophisticated encoding methods are implemented in the system, whereby any programs of varying sizes can be loaded from the external memory PROM
10 (12) at a later stage, and that decoding algorithms can be exchanged with little effort, a - primary ROM contents file (14), consisting of an -external variable part (16), and an - external serial number part - (17) and a production file (20) being generated in sequence by the know-how holder for the production of the microcontroller assembly, whereby the production file
15 (20) is transmitted to a microcontroller (2) which consists of an - internal CPU - (3), an - internal bus - (4), an - internal ROM - (5), an - internal RAM - (6), a selection module (7), a - ROM interface module - (8) an interface (9) and an internal programming signal (10), at the same time as which the a - known, user-specific production key - (19) is sent via the interface (9) to
20 the microcontroller (2), and, with the aid of the conversion algorithm (21), using the - known, user-specific production key - (19), a - primary ROM contents file - (14) is generated from the production file (20), and an encoded data stream, consisting of parts of the - primary ROM contents file - (14) or of the contents of the selection module (7) or of combinations of
25 the selection module (7) and the - primary ROM contents file - (14), is transferred via the - ROM interface module - (8), depending on the configuration of the - external memory (PROM) - (12) once or if required several times using the - external programming signal - (11) and the - internal programming signal - (10) to the - external memory (PROM) - (12), a pro-



cess which guarantees that after every process for every microcontroller module (1) the - external memory (PROM) - (12) is uniquely configured, preventing duplication of the microcontroller module (1) and ensuring that external programs cannot be read or analysed by the producers of the
5 assembly or by other third parties.

2. Method for rendering microcontrollers secure according to Claim 1, characterised by the fact that the selection module (7) can be configured as an EPROM cell or as an unfocused noise generator which always provides the same value on start up and provides scattered values with other circuits
10 because of the narrow tolerance conditions, whereby this is an area which has been radiated and where structural changes have therefore given rise to a different memory status or an area in which various bond islands are present which can be bonded differently when bonding the circuit.

3. Method for rendering microcontrollers secure according to Claim 1 and
15 Claim 2, characterised by the fact that the conversion algorithm (21) can consist of various part algorithms.

4. Method for rendering microcontrollers secure method according to Claim 1 to Claim 3, characterised by the fact that the contents of the selection module (7) are known neither to the manufacturer nor the know-how
20 holder.

SEE THREE SEPARATE PAGES OF DRAWINGS!

This Sheet Has Been Replaced

