IRACST - International Journal of Computer Science and Information Technology & Security (IJCSITS), ISSN: 2249-9555
Vol. 3, No.1, February 2013

Design of a Microcontroller-based Circuit for Software Protection

Eshtiag Jah Alrasool Alsideg Mohammed Ahmed.
Faculty of Science And Art - King AbdAlaziz University
KAU
Jeddah, Saudi Arabia.

El-nzeer El-ameen Mohammed Ali
Gezira Collage of Technology
GCT
medani, Sudan.

***Abstract*—In light of technological development, exploit the potential of computer and benefit from its services has become very important, so the computer software has played a major role in various fields in our life. Facing the use of these software and their manufacturing many of risks such as piracy and unauthorized usage. The object of this research is to contribute in getting rid of those problems by develop a microcontroller-based circuit for software protection. In which a key is completely encrypted in the hardware and that is an easy to use license manager that creates professional and secure license keys to protect your software against piracy and unauthorized usage. The designed circuit has been successfully tested. It shows a reliable software protection.**

Keywords - hard key, software protection, serial port microcontroller-based circuit, piracy, encryption algorithm.

I. INTRODUCTION

computer programs have become more important and with the number of personal computers and Internet users grow, the incidence of software piracy, reverse engineering, modification, break-once run everywhere (BORE) – attacks, copy and illegal usage are some problem faced the software development and distribution, that denies software companies and vendors their rightful return on investment. Due to the large loss revenues of software companies, the software vendors have to pay more attention to develop new protection technique against unauthorized and illegal usage. Although legal protection tools like trade secrets, copyright, patents and trademarks have been put into use, they are not adequate for the software protection. Other methods.such as using serial numbers or user name/password offer only weak protection, since programs are digital products they can be copied bit by bit entirely. Without any help from hardware side, protected software eventually can be cracked by professional crackers.[1] Authors of computer software always feel aggrieved that their works are copied and stalling by unauthorized pirates. Consequently program vendors have been researching extensively to invent a foolproof device to prevent their software, and then they could sell their protected programs at a low price to achieve a large market. [2] For that they develop some techniques for software copy and license protection that prevents the applications from being run on different machines, Volume ID(This method restricts the user to run the application only on the specified drive volume), MAC Address (The MAC Address (Media Access Control) is the hardware addressof anetwork adapter, Hostname(The windows PC name is unique within a network and can be used for copy protection), NetBIOS Computer Name (The NetBIOS computer name is available on every PC. It can be used for software copy protection only on a single user installation (available only with the Professional Edition)) , Volume ID + UNC Pathname (This Installation Code combines the Volume ID with the absolute path of the license file. Assume a situation where an application is installed several times on the same machine, but in different directories: The Volume ID is identical on all installations and therefore the installation may be used by many users (3 installations with 20 licenses allow 60 users to work with your software, but you only sold 20 licenses). With this installation code type the problem of several installations is eliminated), MAC Address + UNC Pathname (This Installation Code combines the MAC address with the absolute path of the license file (for details why using the pathname see Volume ID + UNC pathname), Combination Volume ID + MAC Address + Hostname + UNC Pathname (This Installation Code offers the most restrictive protection as all criteria have to match), Combination Volume ID + MAC Address + Hostname (Restrictive protection for single user applications), IP Address (Uses the IP address of the licensed hardware. Only possible if a static IP address is available. [3], and tools which creates professional and secure license keys to protect software against piracy , Quick License Manager (QLM) (QLM is an easy to use license manager that creates professional and secure license keys to protect your software against piracy. You can create permanent or evaluation (trial) license keys in a snap. Integrate QLM with your application). [4], Smart Dongle (The primary function of Smart dongle is to protect software from piracy. This robust unit can be used to carry passwords, signatures, executable code, or other sensitive data, making it as difficult as possible to steal your software. The affordable price makes it a valuable asset for developers who are looking for an easy, cost-effective security solution for their software while providing portability and convenience to end-users), Matrix-Dongle (Matrix is a reliable safety system for the protection of your software from unauthorized usage and reproduction. Whole purpose is to protect software licenses against software piracy. [5], Cryp Key DAL (Distributor Authorizing License) (Cryp Key DAL is gives vendors the ability to grant others permission to authorize software. With Cryp Key DAL, the vendor achieves control over the number of licenses a distributor can issue by providing a distributor with a pre-configured copy of Cryp Key's SKG (Site Key Generator). The vendor uses its Master SKG to authorize the distributor's SKG

for a limited number of runs. Cryp Key Instant is a software-based solution that embeds its licensing protection directly into your application executable. It protects applications by wrapping and encrypting your executable (EXE) or dynamic link library (DLL) files with Cryp Key's proven protection technologies). [6], HardKey License Manager: (allows to generate cryptographically strong serial numbers based on asymmetric crypto algorithm.[7].

Design of a microcontroller-based circuit for software protection (hard key) is an extension to that tools and it is protects software from piracy and unauthorized usage by exchange encrypted key with protected software.

II. SYSTEM DESIGN

Reverse engineering, modification, and break-once run everywhere (BORE)– attacks, copy and illegal usage are some problems faced the software development and distribution. Design of a microcontroller-based circuit for software protection (hard key) protects software from all above. Hard key still costive and not more available. For these reasons there should be a method of designing of a microcontroller based circuit for software protection (hard key) that is far less cheap. The following sections describe the design steps of the proposed system

A. Hardware Design

The system is composed of three main parts namely; serial port, max 232 and microcontroller. The serial port is a computer port which is opens by the protected software that sends its key to the max232 which converts signals from an RS-232 serial port to signals suitable for use in TTL compatible digital logic circuits. The microcontroller which receive that key and sends it encrypted to the protected software via max232 works under software control.

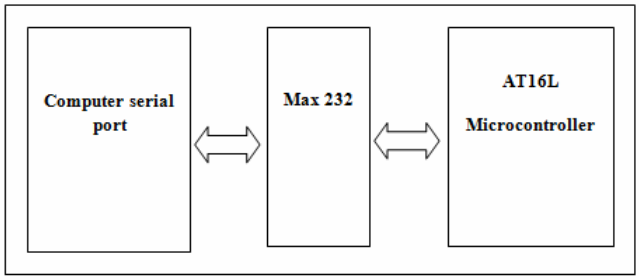

Figure 1. System Block Diagram

Figure 1: System Circuit Diagram explain the need of MAX232 which is compatible with RS-232 standard, and consists of dual transceiver. Each receiver converts TIA/EIA-232-E levels into 5V TTL/CMOS levels. Each driver converts TTL/COMS levels into TIA/EIA-232-E levels.

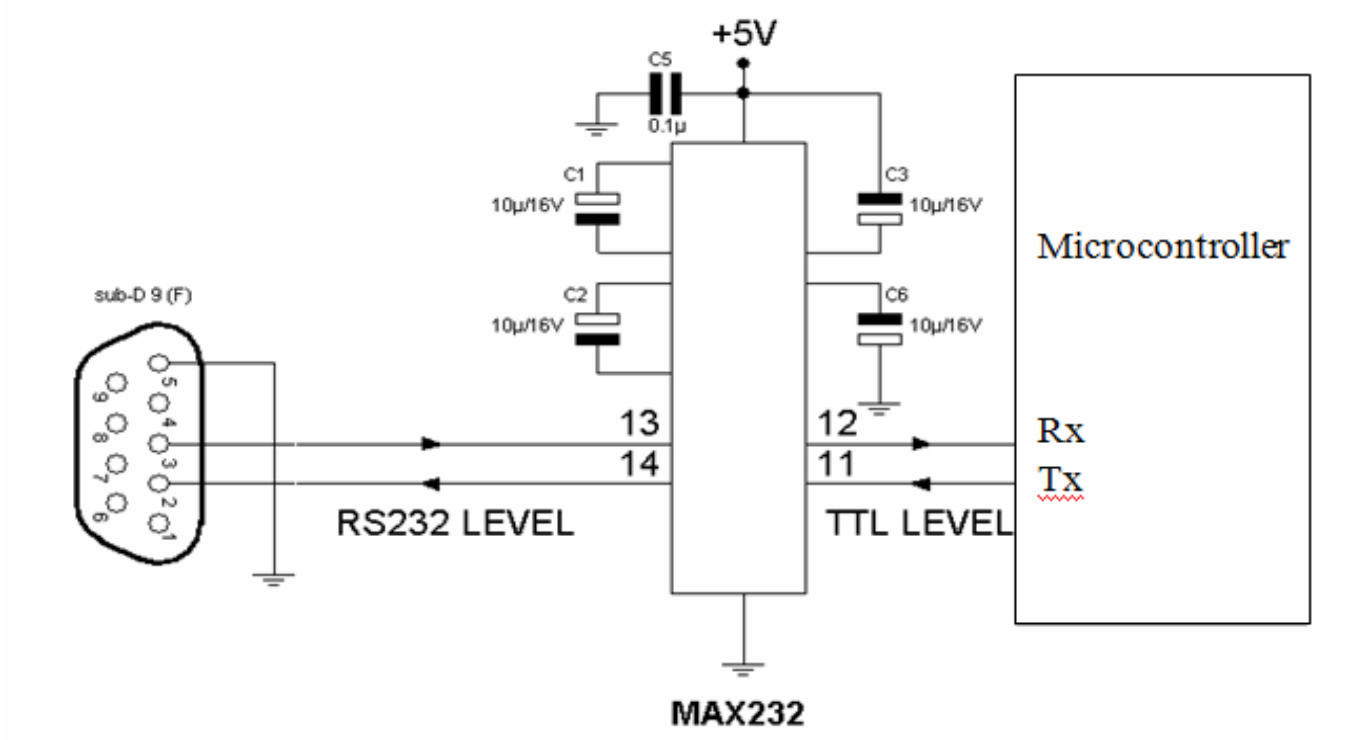

Figure 2. System Circuit Diagram

Figure 3. Method of key checking

Figure 2: the circuit diagram of the system, consist of AT16L microcontroller, max 232 and serial port. It shows the max 232 circuit and it's connection with of AT16L microcontroller rand the serial port

B. Software Development

The functionality of the system describes a simple technique to exchange a key between the circuit and protected software which is waiting for specific response to run in a full user permission or run in a limit user permission area.

.B/Method of key exchange

The protected software checking specific port (serial port) searching for its hard key when any user run it.

then make decision in a fallowing :

1. If it's not find the hard key then run in a limit user permission area.
2. If it finds the hard key then sends a key to it, after that still waiting for response to make decision:
3. With availed key run in full user permission.
4. With invalid hard key run in a limit user permission
5. If it runs in a limit user permission or in a full user permission area, The protected software still checking the hard key.

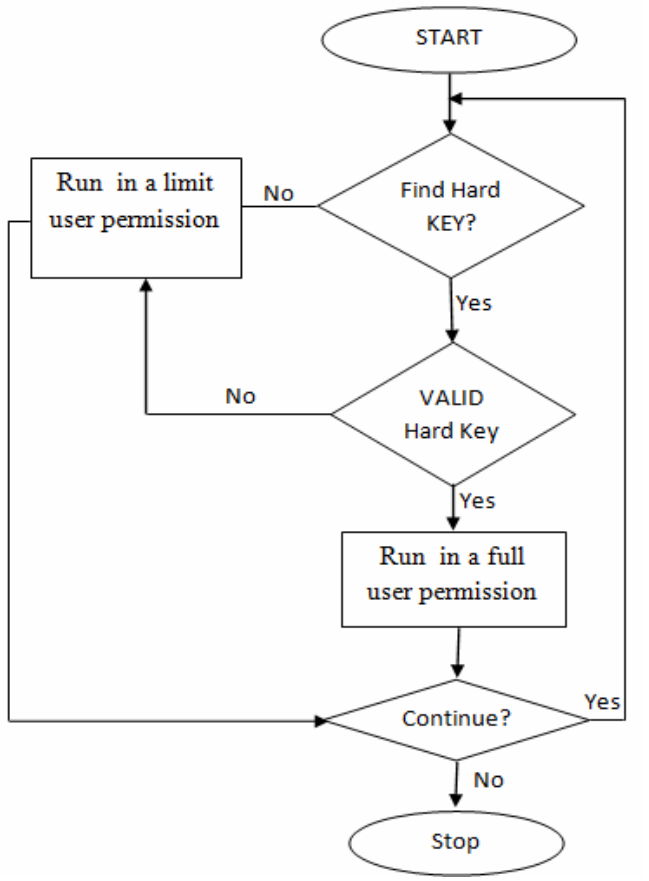

Figure 3 represents the steps of exchanging key

B.2 Encryption Algorithm

Encryption is the act of encoding text so that others not privy to the decryption mechanism (the key) cannot understand the content of the text.

Algorithm is any set of detailed instructions which results in a predictable end-state from a known beginning.

Here is an algorithm that the researcher use to exchange an encrypted key between the hard key and the protected program ,algorithm Instructions go through the following:

1. The protected program chose k0 is a number randomly, calculate (k_1= 3 k_0+37) and then sends a key (k_1) to the hard key.
2. A hard key receives that key and make three things:
   First: find the key (k =(k_1-37)/3)
   Second: multiplex the key by three then add (17) to it (k_2 =3k+17)
   Third: send the key (k_2) to protected program.
3. The protected program receives that key and make three things:
   First: find (k_1) from the receive key (k_1= k_2+20).
   Second: find (k) using (k =(k_1-37)/3)
   Third: make comparison between two keys (k_0 , k).
4. Protected program run in a full version when the tow keys are equivalent (k0 = K), or run in a limited version if they are not(k0 ≠ K).

III. Implementation And Experimental Result

A. Hardware Implementation

An electronic circuit is composed of individual electronic components , such as resistors, transistors, capacitors, inductors and diodes, connected by conductive wires or traces through which electric current can flow.The components used for the circuit design shown in the followingtable .

Table 1. Circuit Components

Component	Specifications	No
Microcontroller	AT16L	1
Crystal	16.9344MHz	1
Capacitor	27Pf	2
Capacitor	1Pf	4
Switches	Push Button	1
Normal diode	N11.4	1
Max 232	Lm78L05817	1

The microcontroller Reset (RS), Vcc, Gnd, SCL, XTAL1and XTAL2 pins, are connected to 16.9344MHz Crystal which is used to generating clock signal for the microcontroller, transmit (Tx) and receive (Rx) pins of microcontroller are connected to receive (Rx) and transmit (Tx), respectively in TTL MAX level. Transmit (Tx) and

receive (Rx), in serial port are connected to input and output in MAX232 level, as shown in figure 4.

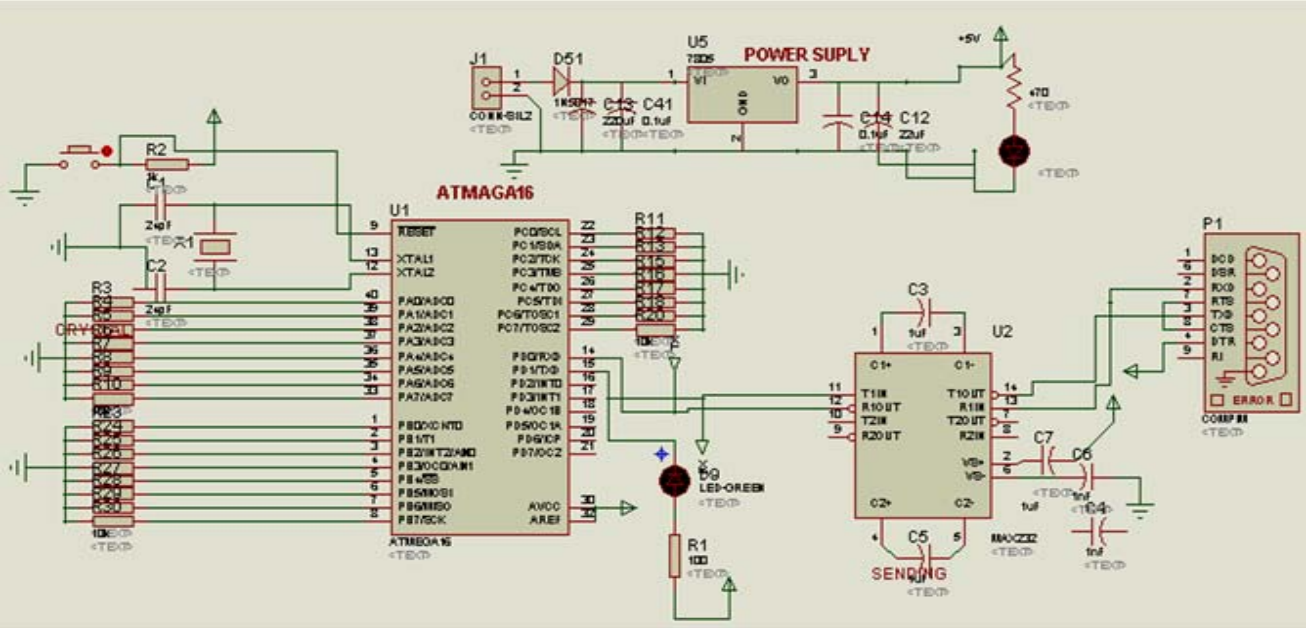

Figure 4. Proposed Circuit

Figure 4: explain Proposed Circuit

In circuit design the researcher use, A microcontroller that is a small computer on a single integrated circuit consisting of a relatively simple CPU combined with support functions such as a crystal oscillator, timers, and watchdog. Neither program memory in the form of NOR flash or One Time Programmable Read Only Memory (OTPROM) is also often included on a chip, as well as a typically small read/write memory. Microcontrollers are designed for dedicated applications. Thus, in contrast to the microprocessors used in the personal computers and other high – performance applications, simplicity is emphasized. Microcontrollers are used to automatically control products and devices, such as automobile machine control systems, remote controls, office machines, appliances, power tools, and toys, Crystal oscillator,that is an electronic circuit that uses the mechanical resonance of a vibrating crystal of piezoelectric material to create an electrical signal with a very precise frequency. This frequency is commonly used to keep track of time (as in quartz wristwatches), to provide a stable clock signal for digital integrated circuits, and to stabilize frequencies for radio transmitters and receivers and MAX232, that converts signals from an RS-232 serial port to signals suitable for use in TTL compatible digital logic circuits, it's a dual driver/receiver that includes a capacitive voltage generator to supply EIA-232 voltage levels from a single 5-V supply. Each receiver converts EIA-232 inputs to 5-V TTL/CMOS levels into EIA-232 levels [12].The input to the max232IC will be of Transistor Transistor Logic (TTL) and its voltage range will be of 5 volt. The output of the max 232 ic will be of Recommended Standard 232 Logic (RS232 logic) and its voltage range will be of 12 volt. The max 232 ic which is operating on 5 volt supply has to step up the 5 volt input to 12 volt. Hence a capacitor is used for pumping the voltage and called as pumping capacitor.

B. software implementation

The microcontroller work under software control which is written in C language. To program the microcontroller using the C programming language, needs two tools:

1. AVR Studio which is an integrated development environment that includes an editor, the assembler, HEX file downloader and a microcontroller emulator. WinAVR which is for a GCC-based compiler for AVR. It appears in AVR Studio as a plug-in.
2. WinAVR also includes a program called Programmer's Notepad that can be used to edit and compile C programs, independently of AVR Studio.

To load the program in the microcontroller, go through four major stages:

- creating an AVR Studio project,
- compiling C code to HEX file,
- debugging C program using the simulator,
- downloading HEX file to the Atm16L development board and running it.

C. Experimental Results

The design of a microcontroller-based circuit for software protection (hard key) and its implementation enabled us to access the protected software in a full version which all system function activated when a valid hard key used, and with invalid hard key the protected software stays in a limited version,

which is some menu and some sub menu disabled and other enabled.

When we use invalid hard key the protected software send a message shown in figure 5.

Figure 5. The message display with invalid hard key

Figure 5: shows the message displayed when the protected program runs with invalid hard key.

Thus protected program open in limited user permission area, that shown in figure 6.

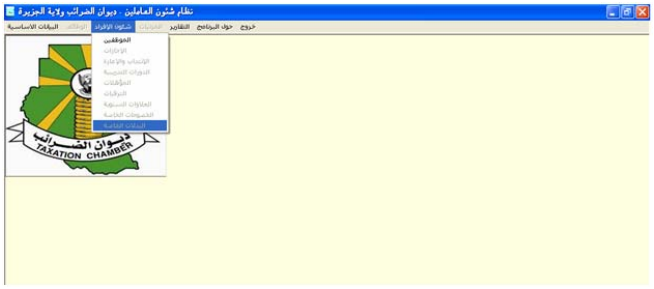

Figure 6. a limited user permission area.

Figure 6: shows the protected program when it runs in a limited user permission area, where some menu (jobs and salary) and some menu options (employee, qualification, etc, in employee effort menu), disabled .

When we use availed hard key the protected software send a message shown in figure 7.

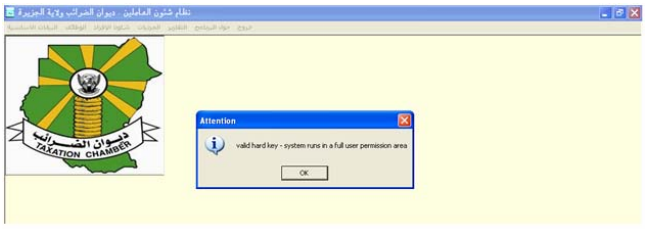

Figure 7. The message display with a valid hard key

Figure 7 shows the message display when the protected program runs with valid hard key.

Thus protected program open in limited user permission area, that shown in figure 8.

Figure 8. a full user permission area.

Figure 8: shows the protected program when it runs in a full user permission area, where all menu and all menu option are enabled. See the menu (jobs and salary) and menu option (employee, qualification, etc, in employee effort menu) which are disabled in a limit version, here in a full version are enabled

CONCLUSION

This design was implemented by using: Microcontroller AT16L, Crystal Oscillator 16.93MHz, max232, Switch, and Capacitors. The system makes them easy to store the key inside hard key and cannot be read from it, receive data, encrypt and send the encrypted key. The design of a microcontroller-based circuit for software protection (hard key) and its implementation, show us very strong tool for software protection.

Thus, we say that we have developed a method and an excellent tool to protect software from piracy and unauthorized usage.

REFERENCES

[1] Qiang Liu ,"Techniques using exterior component against software piracy,", Department of Computer Science, University of Auckland.

[2] Qiang Liu ,"Techniques using exterior component against software piracy,", Department of Computer Science, University of Auckland.

[3] Software copy protection,http://www.miragesystems.de/products/licence-protector/overview-home/software-copy-protection/print.html,Retrieved on February 2013

[4] Interactive solution,
http://www.interactivestudios.net/Products/QuickLicense Manager.aspx. Retrieved on February 2013

[5] Software copy protection, http://www.matrixlock.de/english/e_allgem.htm, Retrieved on February 2013

[6] CrypKey Instant User Manual, www.crypkey.com/support/manual_request.php?,Retrieved on February 2013

[7] Strong bit technology,
http://www.strongbit.com/hardkey_inside.asp,Retrieved on February 2013

AUTHORS PROFILE

Ms Eshtiag Jah Alrasool Alsideg Mohammed Ahmed, has been working as a Lecturer at Faculty of Science And Art - King Abd Alaziz University - Jeddah, Saudi Arabia.

Mr El nzeer El ameen Mohammed Ali, has been working as a Lecturer at Gezira Collage of Technology - medani, Sudan.