

Using the HC08 Microcontroller Family to Enhance System Security

Overview

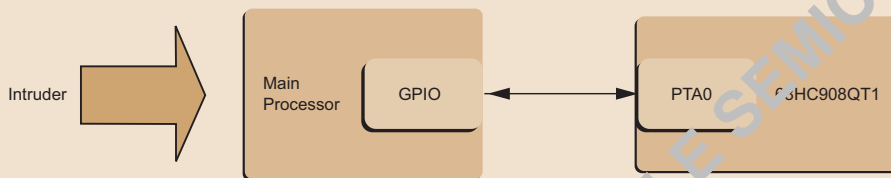
Prevention of unauthorized reproduction of an embedded system intellectual property (IP) is achievable with the use of a device that provides on-chip IP protection capabilities. An example of such a device is a microcontroller unit (MCU) with embedded

Flash memory featuring memory lock functionality. When a device without IP protection features is used in a design, the overall system can still be protected with the use of low-cost MCU like those offered as part of the 68HC08QT family.

Key Benefits

- > Add security to hardware and firmware designs
- > Prevents unauthorized duplication of intellectual property
- > Provides authentication capabilities to existing systems
- > Programmable security solution

COMMUNICATION LINK BETWEEN TWO PROCESSORS



Freescale Ordering Information

Part Number	Product Highlights	Additional Information
68HC908QT1	Member of the low-cost, high-performance M68HC08 Family of 8-bit microcontroller units (MCUs).	www.freescale.com ^{Note}
68HC908QT2	Member of the low-cost, high-performance M68HC08 Family of 8-bit microcontroller units (MCUs).	
68HC908QT4	Member of the low-cost, high-performance M68HC08 Family of 8-bit microcontroller units (MCUs).	

Note: Search on the listed part number.

Design Challenges

Embedded systems are increasingly requiring security features to guard against unauthorized access to the intellectual property contained within the system. Due to a variety of systems requirements, some of these systems utilize processors that do not include integrated security protection features. For these systems, a low-cost system security feature is desirable.

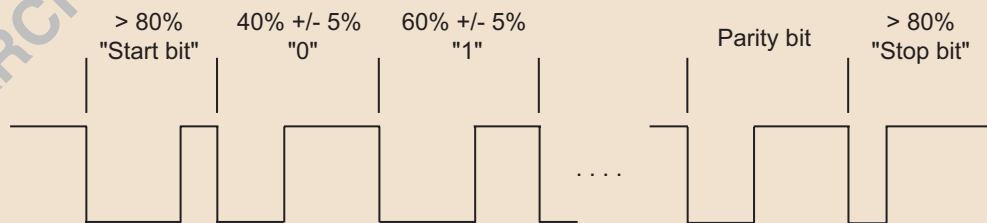
Freescale Semiconductor Solution

The figure on page 1 shows a hardware implementation with an 68HC908QT1 controller, which prevents an unauthorized party from duplicating an entire system. Although the hardware, printed circuit board, and object code stored in the main processor can be copied exactly, the code stored in the 68HC908QT1 cannot be copied, thanks to its built-in Flash security mechanism.

Communication between the two processors uses single-wire asynchronous protocol. To enable communication, each processor uses one GPIO which works as both input and output to prevent signal contention on the communication line. Both GPIOs are initialized as inputs by enabling the internal pull-up resistor and setting the corresponding bit in the GPIO data register to "0" after the device is reset. When GPIO is set as input, it also represents output "1". If the GPIO is required to send an output "0", simply set GPIO as output, because the corresponding bit in the GPIO data register has been set to "0". By toggling the corresponding bit of the GPIO direction register, outputs "0" and "1" can be sent from GPIO without communication line contention. The advantage of using single-wire communication protocol is that it is difficult to distinguish the sender from the receiver.

The 68HC908QT1 is a low-cost MCU which includes an on-chip oscillator. The internal oscillator circuit is designed for use without external components to provide a clock source with a tolerance of 25% without trimming, the internal clock can provide $\pm 5\%$ accuracy after trimming. Implement a timing-insensitive communication protocol as shown in the figure below to generate a different duty cycle pulse to represent binary data ("1" or "0"), the start bit and the stop bit. A timer monitors the duration of each bit. If a time-out occurs, the counter will record this time out; please refer to the figure below.

COMMUNICATION PROTOCOL BASED ON PWM DUTY CYCLE

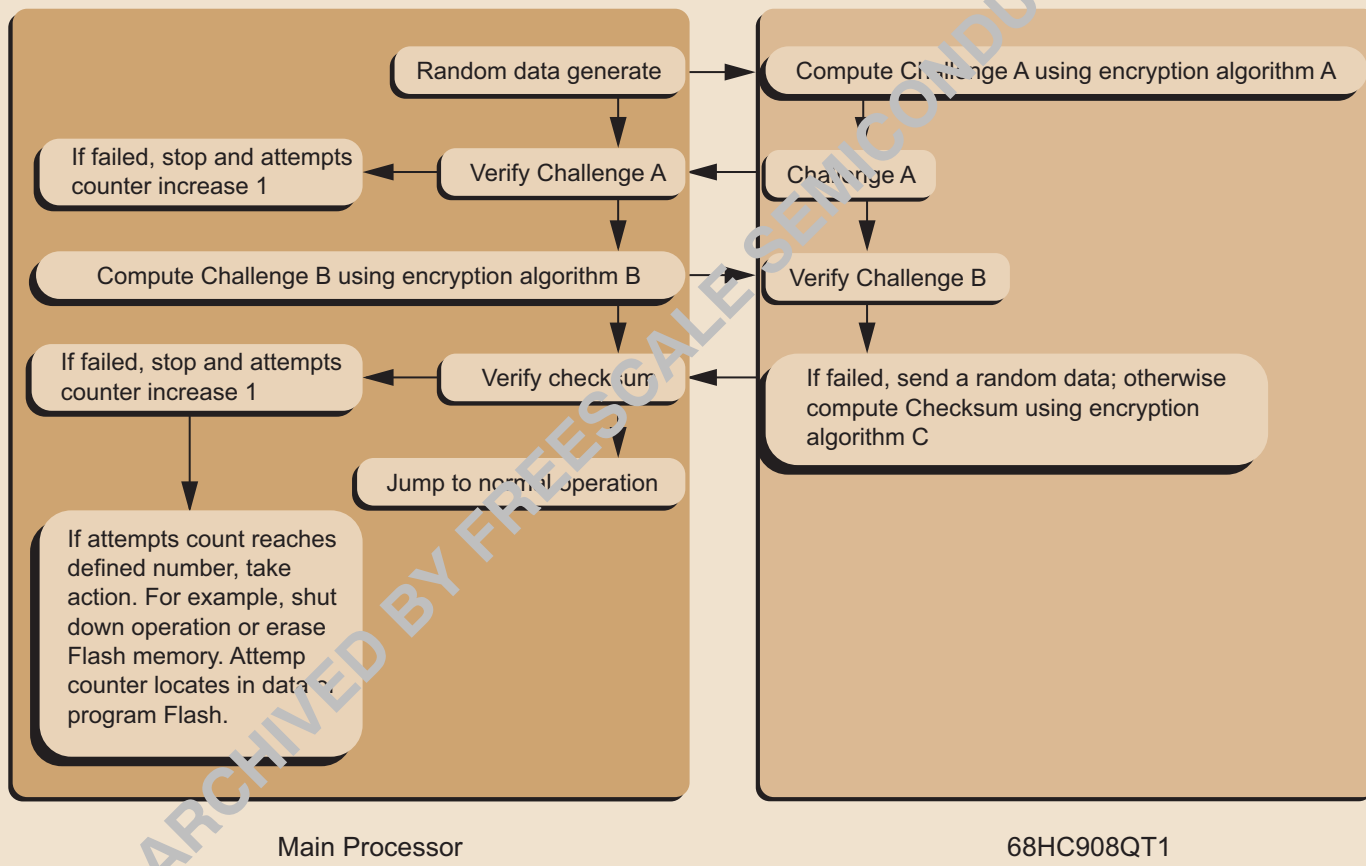


Although the code in the 68HC908QT1 cannot be copied, the data exchanged between the two processors can be intercepted by an intruder. To prevent unauthorized access to the information transmitted between the two processors, data encryption algorithms must be implemented in both processors in order to cipher and decipher the transmitting data. Encryption is a security method which prevents intruders or unwanted access from inside or outside your system by using a string of letters and

numbers as keys to encode your communication data. Any controller that connects to communication lines must know the encryption key in order to access data. If the algorithm itself is secure in its design, then the data is secure as long as the key is secure, even if the encryption algorithm is known. There are many ways to hide the key. One method is to create a key generation algorithm written in a high-level language such as C and to insert the data table into the program code.

This will create a certain level of complexity and make it difficult to understand the disassembled object code. The figure below shows the steps of the authentication operation. If a pre-defined number of verification attempts fail, both processors stop work and, if desired, erase the internal Flash memory.

AUTHENTICATION OPERATION



Development Tools

Tool Type	Product Name	Vendor	Description
Hardware	Emulation Module	Freescale Semiconductor	M68EML08QTQY
Hardware	Modular Development System (MMDS)	Metrowerks	KITMMDS08QTQY
Software	CodeWarrior Development Studio for HC(S)08 Special Edition	Metrowerks	CDCWSEHC08

Disclaimer

This document may not include all the details necessary to completely develop this design. It is provided as a reference only and is intended to demonstrate the variety of applications for the device.

ARCHIVED BY FREESCALE SEMICONDUCTOR INC.

Learn More: Contact the Technical Information Center at +1-800-521-6247 or +1-480-768-2130.
For more information about Freescale products, please visit www.freescale.com.