SGS-THOMSON MICROELECTRONICS

APPLICATION NOTE

CODED LOCK USING THE ST6-REALIZER

by Olivier Rouy

INTRODUCTION

The trend is to provide more and more applications with security features. This is requested to insure data confidentiality, access control, or simply user identification. Non Volatile Memory is usually required to store the identification or secret code. With the embedded EEPROM provided on the ST62 MCU, one-chip solutions can be achieved with the associated cost and density advantages.

A coded lock system has been developed with the ST6-REALIZER environment and is described in this note. This development tool is used to get an automatic generation of the application software.

1 APPLICATION OVERVIEW

The application described here consists of a coded lock for a door. A secret code is loaded into the lock system to allow the door to be unlocked only if the same access code is entered. When loaded, the secret code is stored in the non-volatile memory (EEPROM) embedded in the ST62 MCU. This ensures that the data is retained even after a voltage cut-off and insures the security is maintained along the time.

The application provides the following features:

– Secret code recording (3 digits).

– Recognition of entered access code.

– Door lock control.

All the featured functions of the system are managed in the final application under software control by the MCU. However, using the ST6-REALIZER allows this application software to be generated by a functional description of the system. Thus the software development can be achieved by a system oriented or hardware like description.

Only the core of the application, the secret code storage in EEPROM and the code recognition, are described as a generic base for various applications. Any kind of user interface could be used (keyboard, IR or RF solution for instance), while the output signal can activate any kind of circuit. In the present case, it is assumed that the user interface provides the following 4 inputs:

– Digit reception flag

– Digit value (0..9)

– Change secret code

– Lock the door

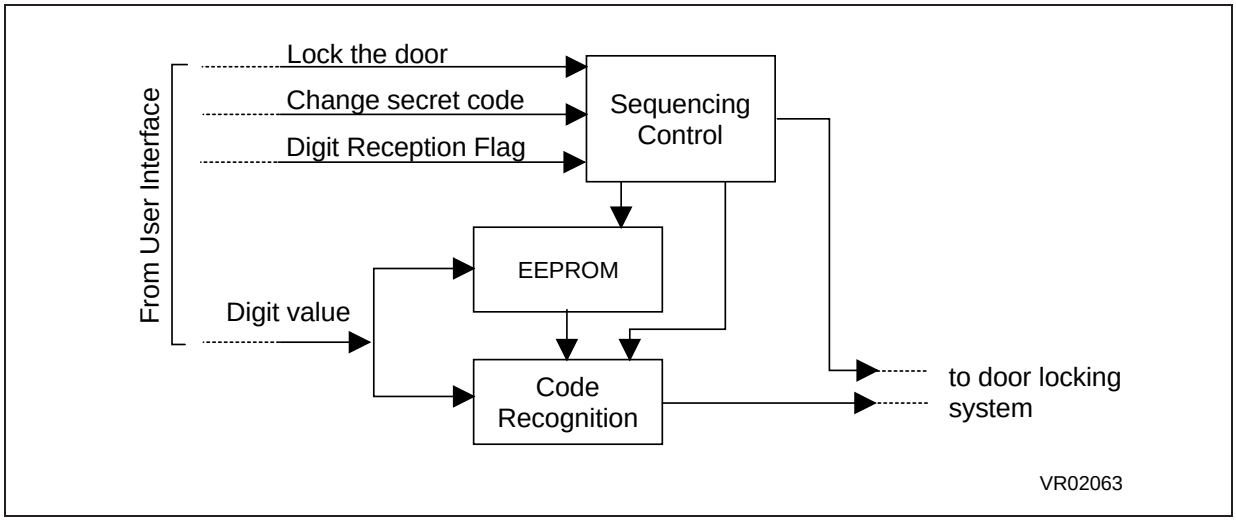
The digits used for the code are serially received, announced by the activation of a flag.

2 PRACTICAL REALISATION

2.1 Functional description

By restricting the application description to its core (the operations sequencing, the secret code storage in the EEPROM and the code recognition), the resulting application block diagram is represented in Figure 1

Figure 1. Block diagram of the coded door lock



2.1.1 Sequencing control

The most important feature is the sequencing control of all the operations. Two main items must be considered: the transition between the different working modes (Door locked, Access code entry, Door unlocked and Secret code entry) and the management of the serial flow of the numeric values received through the user interface.

The transition between the working modes follows conditional rules described in the ST6-REALIZER with the help of a state-machine as described in Figure 2 where four states are existing:

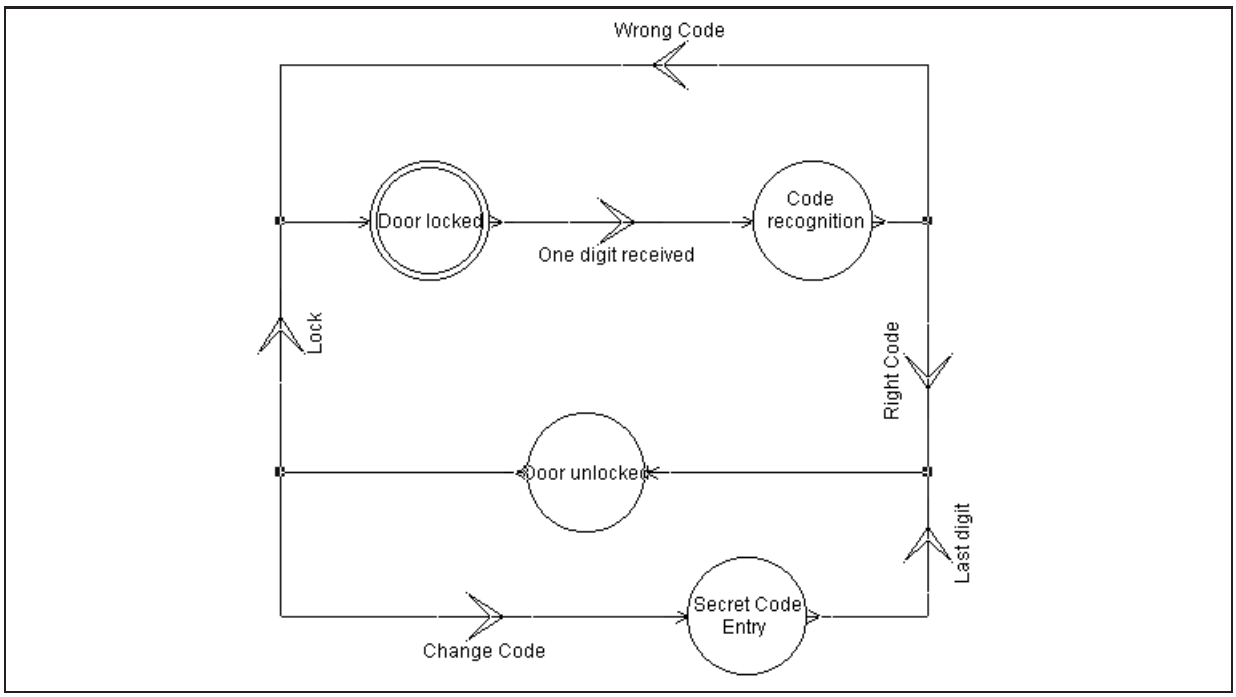
– Door locked

– Code recognition

– Door unlocked

– Secret code entry

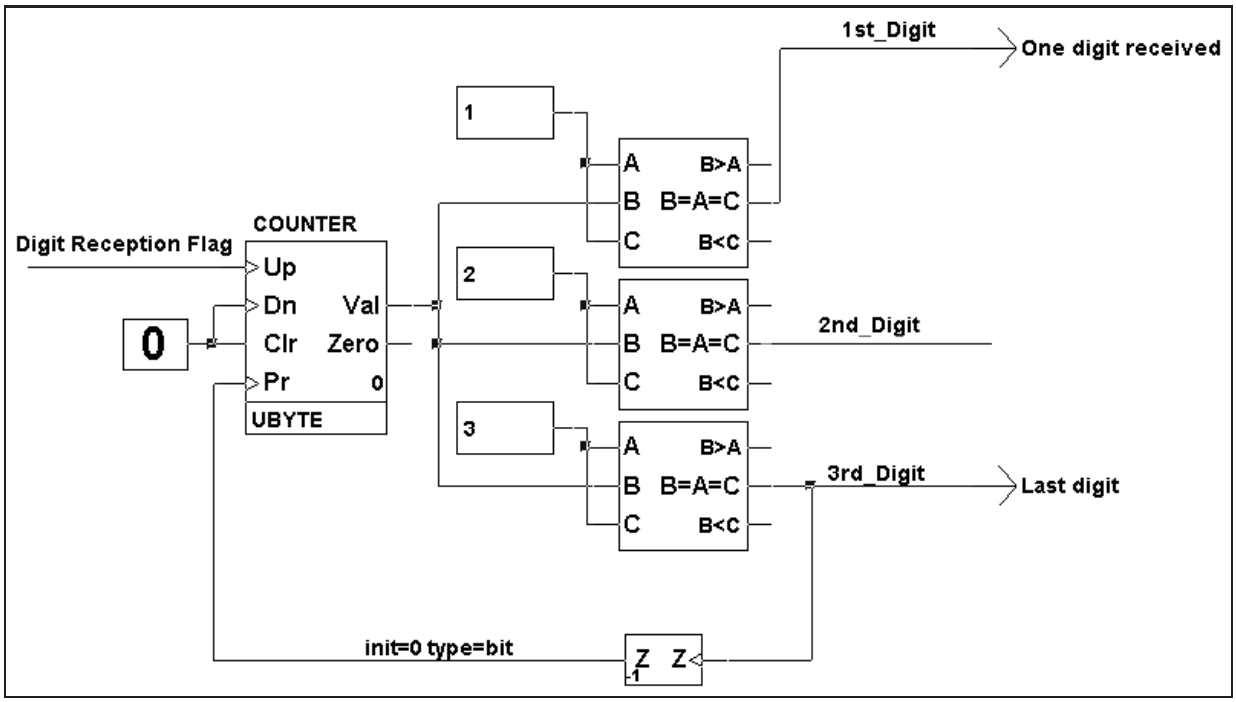
Figure 2. State machine description for sequencing control



The conditional occurrences are generated either through the user interface (*Lock, Change secret code)*, or through some functional sub-blocks of the application (*One digit received, Right code, Wrong code, Last digit*). For instance, the occurrence of the condition *One digit received* in the Door locked state initiates the process Code recognition.

The management of the serial flow of the digits entered (0..9) is achieved with a counter of received digits (Figure 3). Depending whether it is the 1st, the 2nd or the 3rd digit (In the case of a 3 digit code), the numeric value issued by the interface is stored in the associated memory location. When the 3rd digit is received, the counter is reset. In addition, the reception flags of the 1st digit and 3rd digit are used in the state-machine to initiate respectively the code recognition and conclude the Secret code entry.

Figure 3. Management of the digits reception



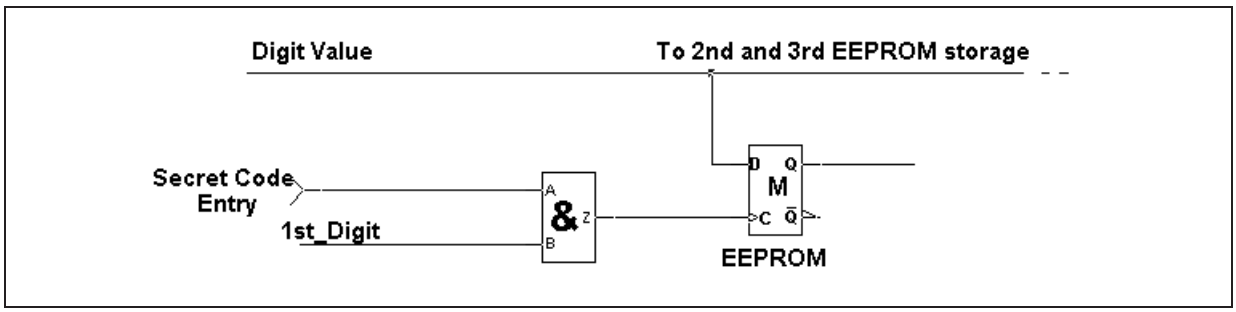
2.1.2 Secret code storage in the EEPROM

Thanks to the ST6-REALIZER, the EEPROM Non Volatile Memory can be managed as a standard bistable component: R/S Latch, D Latch, Shift Register or Counter.

In the present case, each EEPROM location is symbolised by a specific D Latch component providing the Non Volatility feature. This component, provided within the ST6-REALIZER library, has the advantage of being multitype. This means the same symbol can be used whatever the input type: Bit, Byte, Word, Integer or Signed variable.

In the present case, the data input comes directly from the interface as the Digit value, in one byte. The number of D Latches must be equal to the number of digits used for the secret code, 3 in the present case. There is a specific clock for each of the 3 D Latches, controlled by the sequencing control module. Data can be written (Clock activated) into a D Latch only if the active mode is Secret code entry, and if the received bit corresponds to this location (Figure 4). Practically, the clock is generated through an AND function between the Secret code entry mode and the ranking of the received digit. This ranking is issued by the counter used for the serial flow reception.

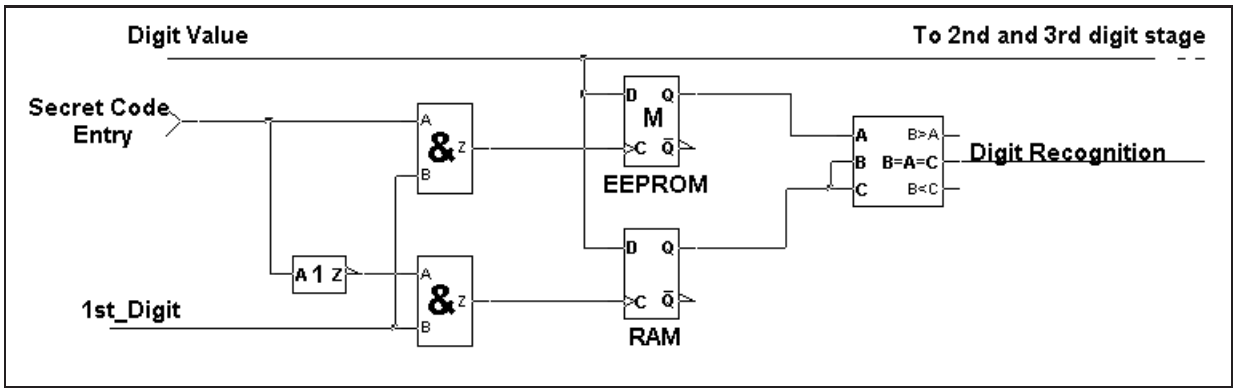
Figure 4. Digit storage into EEPROM memory



2.1.3 Access code entry and recognition

The digits are received serially as in the Secret code entry mode, and stored in Volatile Memory locations (RAM). The RAM location are described with standard D Latch symbols. The Input/Output of this component are exactly the same as for the Non Volatile one allowing a clear analogy between the functional description of the two modules. The data input channel is the same while the clocks are logically always validated except if the Secret code entry mode is activated (Figure 5).

Figure 5. Digits management for code recognition.



The code recognition is performed by comparison one-to-one between the values stored in EEPROM and RAM.

NOTES: