InSystProgC868, V 2.0, Jan. 2003

C868 Application Hint

In System Programming C868-1S

This document describes an in system programming of an EEPROM with the C868. It is explained how the user can manage to reprogam an already programmed EEPROM.

Software examples and ready routines are provided in C-language.

Author: Andreas Jansen, AI MC IGM

Microcontrollers

Revision History:	**2003-01** **V 2.0**
Previous Version:	-
Page	Subjects (major changes since last revision)
	Use Compiler Keil V7.01
	Use DAvE C868 B-step
	Use different eepro_A5.hex for C868 - BAstep
	Change SW in main-loop and in UART-ISR

1.1 Booting the C868 from an EEPROM

After RESET and enabled bootmode (BSL pin low) the C868 boots via SPI from an external EEPROM or via UART from a host. Therefore the C868 checks for a password at the first address of the EEPROM. If this password is correct it starts booting the code from the EEPROM to the C868's SRAM. After a complete download it jumps to the SRAM address 0000h and executes the code.

1.2 Unlocking the EEPROM Boot-Sequence

Once the EEPROM contains the correct password the C868 always boots from the EEPROM. For practical development it is useful to reprogram the contents of the EEPROM very often. This can be done by

- disconnecting the EEPROM while booting, e.g. opening the connection between the chip select of the EEPROM and the C868 or
- reprogramming the password.

Ofcourse the reprogramming of the password is the prefered option because it works without manually changing the hardware and can easily be done in system. But this requires that the user's code contains the following code:

- EEPROM read- and write-routines
- a routine which starts the unlocking sequence on the user's command.

This unlocking routine can be invoked by serial interrupt for example.

1.3 Example Project and Guideline

The following gives a simple example for programming and reprogramming the EEPROM in system by using a serial interrupt together with the MiniDebugger and the C868 StarterKit.

1.3.1 Programming the EEPROM

1.3.1.1 Programming Step 1

Press RESET and connect with the MiniDebugger.

1.3.1.2 Programming Step 2

Download code "rprotect.hex" to XRAM.

1.3.1.3 Programming Step 3

Execute code in XRAM. Now the EEPROM is prepared for first time programming. Once the protection is removed step 2 and 3 can be skipped for programming.

1.3.1.4 Programming Step 4

Download code "b11_ee_isys.hex" to SRAM.

1.3.1.5 Programming Step 5

Download code "eproA5_g.hex" to XRAM.

1.3.1.6 Programming Step 6

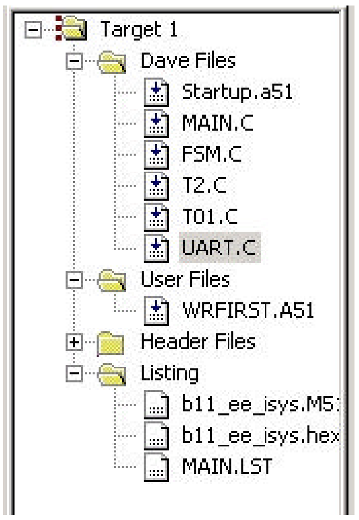
Execute code in XRAM. Now the EEPROM is programmed with the code which is in SRAM and the password is programmed to the first address of the EEPROM. With the next RESET the C868 will boot from EEPROM and jump to SRAM.

1.3.2 Reprogramming the EEPROM

Connect the running program to the MiniDebugger. By this connect the MiniDebugger host sends the byte 0x80 to C868 via the serial interface. Now the "b11_ee_isys.hex" jumps to the serial interrupt service routine and checks for the received byte. If this byte equals 0x80 it invokes the reprogramming of the password of the EEPROM. Now the bootsequence is unlocked and after the next RESET the bootstraploader will boot from serial interface.

1.3.3 Description of the Software Example

The project is generated by DavE (C868 B-step) and compiled by a Keil C-Compiler evaluation verions 7.01. It uses the following modules:



```
Target 1
  Dave Files
    Startup.a51
    MAIN.C
    FSM.C
    T2.C
    T01.C
    UART.C
  User Files
    WRFIRST.A51
  Header Files
  Listing
    b11_ee_isys.M5:
    b11_ee_isys.hex
    MAIN.LST
```

There are in principle two routines. The main routine which only toggles the LED and depending on the programming state of the EEPROM (locked with password or unlocked) it changes the colours. The second routine is the UART interrupt service routine which is only activated when receiving a byte from serial line. After the receive interrupt activates this routine the first byte of the EEPROM is reprogrammed with 0x01 (EEPROM unlocked). Ofcourse this routine can be modified on the user's demand.

After downloading the code "ee_isys.hex" to the C868 and executing the code the left two LEDs toggle green (unlocked EEPROM).

After Programming Step 6 the C868 will boot from EEPROM and the left 2 LEDs will toggle red (locked EEPROM).

After a serial receive interrupt the LEDs will toggle green again and the next boot will start from the UART.

Please find a short extraction of the software below.

Main Routine

```
void main(void)
{
MAIN_vInit();

// USER CODE BEGIN (MAIN_Main,3)
  while (1)
  {
        for (i=0xf000; i>0; i--) ;        // wait loop
        ReadFirst();
        if (ee_rbyte == 0xa5) P3 ^= 0xC0; // toggle green, EEPROM contains passw.
        else                  P3 ^= 0x30; // toggle red, EEPROM not with password
  }
  // USER CODE END} //  End of function main
```

Serial Interrupt Service Routine

```
void void UART_viIsr(void) interrupt UARTINT
{
  if (TI)
  {    TI = 0;
  }
  if (RI)
  {
      if (SBUF == 0x80)       // check for MiniDebugger Connect CMD
      {
            ReadFirst();      // delivers ee_rbyte
            if (ee_rbyte == 0xa5)   // read the first byte of the EEPROM
            {
            WriteFirst();     // now program first byte with FillByte (01h)
            }
      }
          // USER CODE END
          RI = 0;
  }

} //  End of function UART_viIsr
```

Infineon goes for Business Excellence

“Business excellence means intelligent approaches and clearly defined processes, which are both constantly under review and ultimately lead to good operating results.
Better operating results and business excellence mean less idleness and wastefulness for all of us, more professional success, more accurate information, a better overview and, thereby, less frustration and more satisfaction.”

Dr. Ulrich Schumacher

http://www.infineon.com

Published by Infineon Technologies AG