www.elnec.com

AN–ELNEC–EN–ISP-HCS08

Application note for In-System Programming of Motorola/Freescale HCS08 microcontrollers

Introduction

The HCS08 is low-cost, high-performance family of 8-bit microcontroller units (MCUs). All MCUs in the family use the enhanced HCS08 core and are available with a variety of modules, memory sizes, memory types, and package types.

In system programming (ISP) of HCS08 microcontrollers is performed via the single–wire BDM (Background Debug Mode). Bellow are the most important requirements and advices to get high reliability while working with our programmers. For further information look at the MCU specific datasheet.

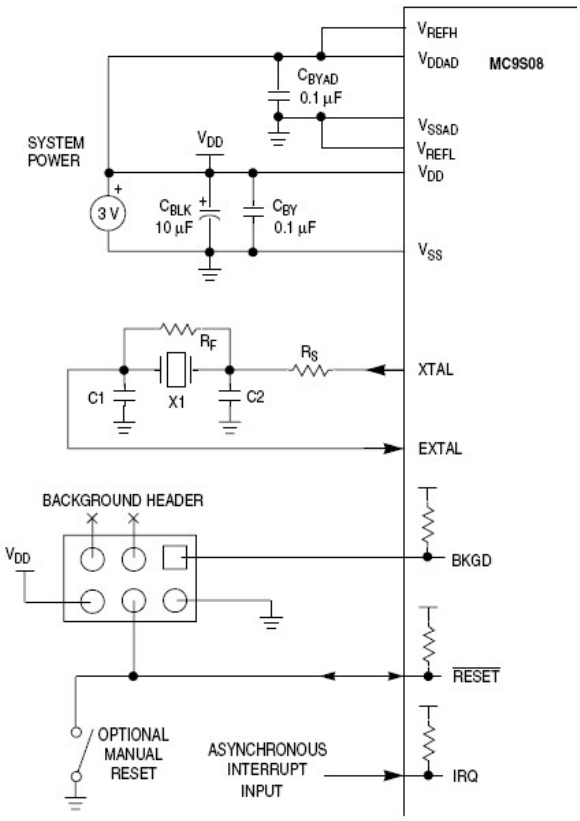

Figure 1. Typical HCS08 system connection

Used signals:

Pin name	***Function***	***Signal level***
VDD, VDDAD, VREFH	supply voltage	3.3 V
VSS, VSSAD, VREFL	ground	0 V
RESET\	reset input	H, L
BKGD	background debug communication i/o, mode select input	H, L, PU
EXTAL	oscillator, clock input	H, L
XTAL	oscillator output	H, L

Table 1. HCS08 MCU ISP related signals description

Recommended target circuit design

In the following, you can find important notices applying to recommended connection of target MCU to the target system.

Implementation of BDM in our programmers comply with Motorola/Freescale BDM connection specification. So, 6-pin header with 2 pins unconnected can be used in design stage of target system.

Purpose of the R1, R2 resistors is to isolate the programmed chip from rest of target system. Recommended value of resistors for particular programmer is specified in *Device info (see Figure 6)*. You can also use jumpers instead of the resistors.

During reset, BKGD is driven LOW and the MCU enters **Special Single Chip Mode**. This allows programmer to communicate through BDM.

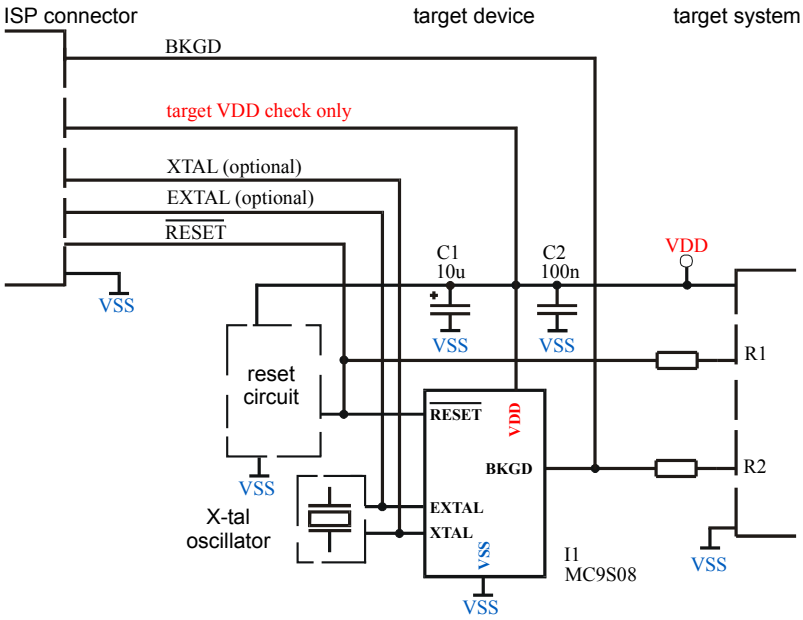

Figure 2. Recommended target circuit design using BDM interface (MCUs with self-clock mode have optional clock signal disabled)

Note: Because communication with target board is performed only by 1 wire (pin BKGD), programming speed may vary depending on system configuration and programmer control SW settings (from x-sec to x-minutes per operation). We recommend to engage MCU's FLL circuitry (if implemented) to get MCU running on appropriate bus/bdm frequency.

Because of high frequencies and fast signal transitions, please, do not omit the bypass capacitors as proposed by manufacturer.

Device operation options

You can edit the Device operation options in menu *Operation options* of control program (Device->Device options->Operation options <*Alt+O*>).

In the case that your programmer supports target system power supply and you intend to use this option, it is needed to set supply voltage parameters according to your requirements.

You may also select level of ISP signals after operation (see *Figure 3*). Description of particular parameters you can find in the menu *Help* <F1>.

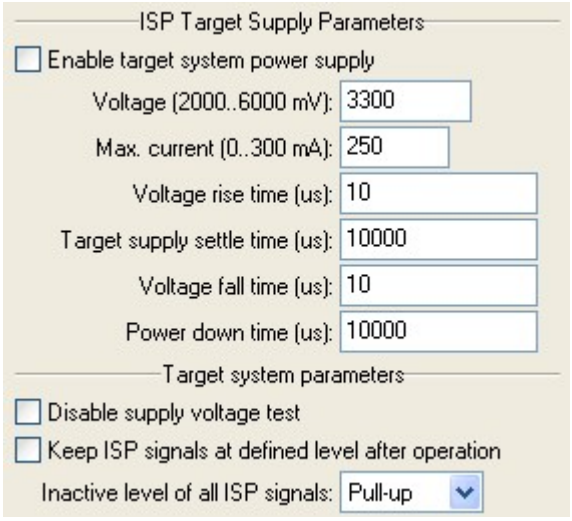

Figure 3. Settings ISP target supply <Alt+O>

There are some additional options to give you more adjustability. Besides *Target system power*, for some members of HCS08 family, the programmer can provide an *optional clock* signal (2 different frequencies).

If you prefer to use your own system oscillator (or clock), there is another option, which allows you to get the best performance – engaging MCU's FLL (implemented only in some members of HCS08 family). This option is available in 3 different frequencies.

Notice: may be, the programmer will not be able to communicate through BDM at your system frequency. In this case, the only way to solve this problem is to engage MCU's FLL. Programmer will set FLL to give the appropriate BDM frequency.

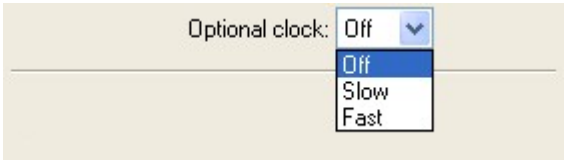

Figure 4. Additional options, optional clock <Alt+O>

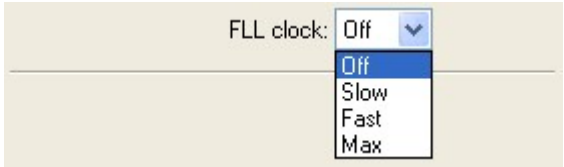

Figure 5. Additional options, FLL <Alt+O>

*[*1)] Remark: Programming algorithm skips blank ($FF) bytes of FLASH area to be programmed. Therefore, sometimes the programming procedure can take a shorter time as verification, and programming time may vary depending on programmed data.*

Tables 2, 3. show all possible MCU clock sources/modes/settings while operating with our programmers.

Oscillator	***Optional clock***	***MCU runs on***
no	off	not running
no	on	optional clock
yes	no	oscillator clock
yes	on	optional clock

Table 2. HCS08 clock source settings/dependence (MCUs without FLL circuitry)

Oscillator	***FLL***	***MCU runs on***
no	off	self-clock (SCM)
no	on	FLL clock generated from internal clock (FEI)
yes	no	self-clock (SCM), oscillator ignored
yes	on	FLL clock generated from internal clock (FEI), oscillator ignored

Table 3. HCS08 clock source settings/dependence (MCUs with FLL circuitry)

Device information

Additional information about selected MCU, such as ISP connector pin numbering, recommended target circuit design for particular MCU etc., you can find in the menu *Device info* of control program (Device->Device info <Ctrl+*F1*>) (see *Figure 6*).

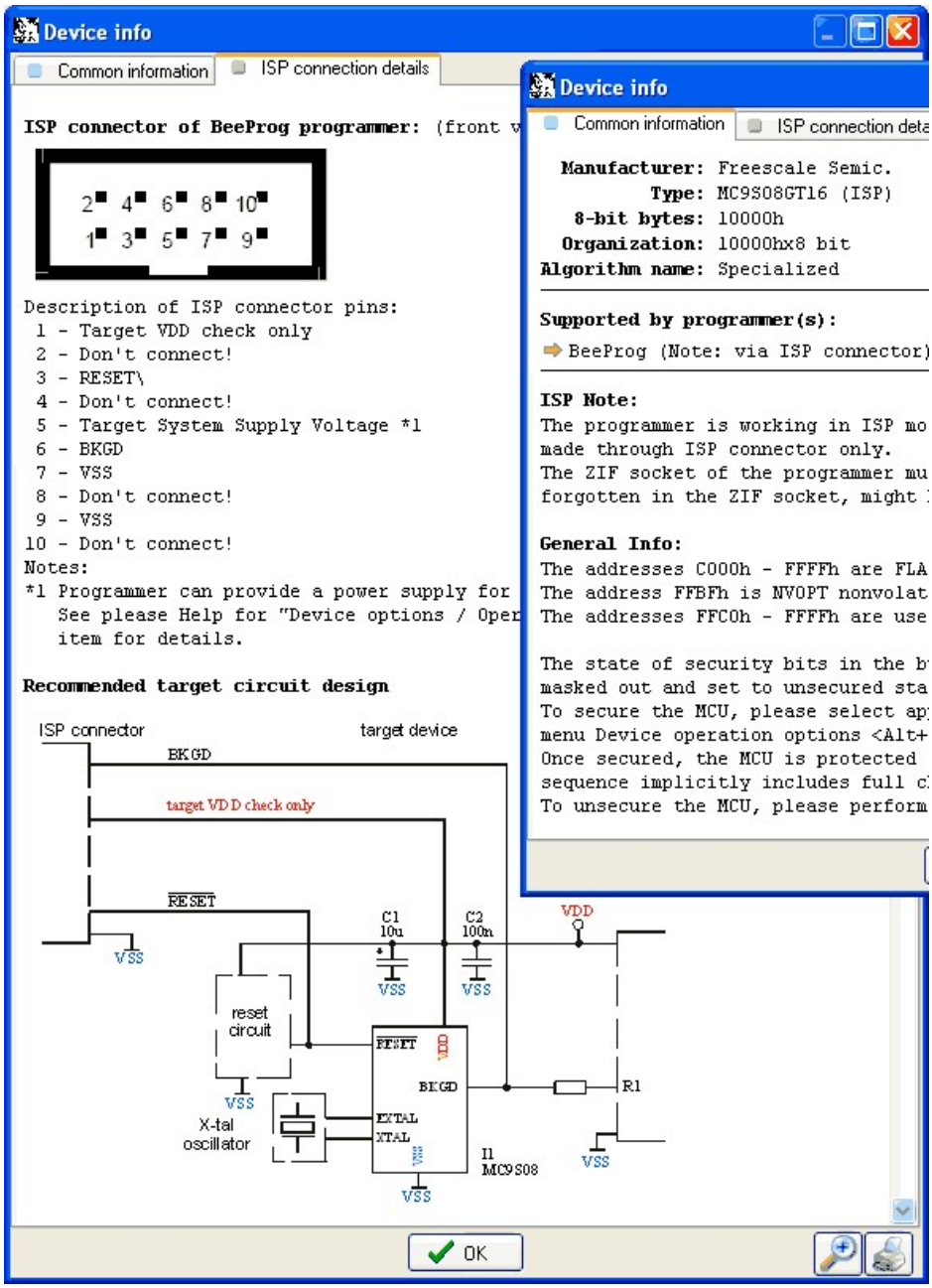

Figure 6. Device info

Good advices and troubleshooting

Connecting programmer to target system:

Turn off power supply of system before connecting/disconnecting programmer to/from system.

Before starting an operation:

Before starting an operation with target MCU, please make sure, that the **ISP cable is correctly connected** to the target system and programmer. Also make sure that no device is inserted to *ZIF* socket of the programmer.

Details about pins assign for each MCU and short description of circuit design you can find in control program (*Device Info <Ctrl+F1>).*

Correctly selected values of resistors R1, R2 provide reliable signal level recognition (for both, programmer and system) and ensure successful finishing of desired operation (see **Recommended target circuit design**).

If something goes wrong:

If the programmer reports **signal interference error**, may be, a signal interference occurred between programmer and target system. Please make sure, your design meets connection recommendations. Check the minimal values of resistors R1, R2, from programmer's point of view (in order to programmer be able to put L/H level on the pin).

If the programmer reports that the **device does not respond**, or the **operation with MCU behaves suspiciously**, check the following:

1. System frequency may result in BDM frequency, which don't comply with programmer's capability. If operations still give errors, you should engage MCU's FLL circuitry (if available) to achieve the appropriate BDM frequency.
2. RC network on RESET\ pin (if connected) hasn't long time constant.

Be aware, that longer ISP cable (longer than 20cm/0,7ft) may cause an unpredictable signal interference. Make sure you are using correct cable.

Revision history

07/2006:

Initial Release.