

If Your Cerebot™ Won't Program (Faulty Clock Source)

Revision: 12/10/07



www.digilentinc.com

215 E Main Suite D | Pullman, WA 99163
(509) 334 6306 Voice and Fax

The Problem: The Cerebot Board Won't Program

If the Cerebot won't program and you see this error message in Digilent AVR Programmer:

```
Creating SPI communications interface
Connecting
Entering Programming Mode
*** Error - Unable to place device into programming mode
```

then the problem may have been caused by setting the clock source to an invalid configuration (such as EXTERNAL).

This is a common problem with the AVR in-system programming (ISP). The AVR ISP uses the current system clock for programming. But since the Cerebot doesn't come with an external clock, if you configure the fuse settings for the AVR ATmega processors to use an external clock, you will be unable to use ISP.

A valid clock signal must be present whenever you program the AVR device using in-system programming. The AVR device typically generates the clock signal internally, but if the clock-select fuse bits are changed, the AVR device could have no clock source at all. If the clock signal is no longer generated by the internal RC oscillator (which is the Cerebot default configuration) then a clock signal must be supplied to the micro-controller. If there is no valid clock source, the device will fail to program and any further attempts to program the device will fail.

The Solution: Supply a Valid Clock Signal

To program the Cerebot to use the internal oscillator again you must first supply a clock signal to the micro-controller. That clock signal can come from a working Cerebot board, an external oscillator, or a function generator. The clock source must:

- provide a square wave of no less than 2Mhz
- have a high clock voltage in the range of 3.0 to 5.0 volts (preferably 3.3 volts)
- have the low clock voltage be ground (or zero volts).

Using a Function Generator to Supply a Clock Signal

1. Obtain a function generator capable of producing a 3.3v 0-p (zero to peak) square wave at 2Mhz (for the external clock source).
2. Ground the function generator to the Cerebot by attaching the ground lead to connector J16 or J17.
3. A programming cable needs to be connected to the inoperative board, and the Digilent AVR programming software running.
4. Locate the two solder pads on the inoperative board that are surrounded by an oval and marked X1 (this is right next to JE). Touch the wire from the signal lead carrying the 3.3v

- 0-p 2Mhz square wave from the function generator to the solder pad farthest from the AVR chip on the inoperative board (see Figure 1).
5. In Digilent AVR Programmer, select the Fuses tab and select [CKSEL = 0100] Calibrated Internal Oscillator, 8.0Mhz (see Figure 2).
 6. With the wire still touching the solder pad, click Program. The device should program successfully and run without the external clock source.

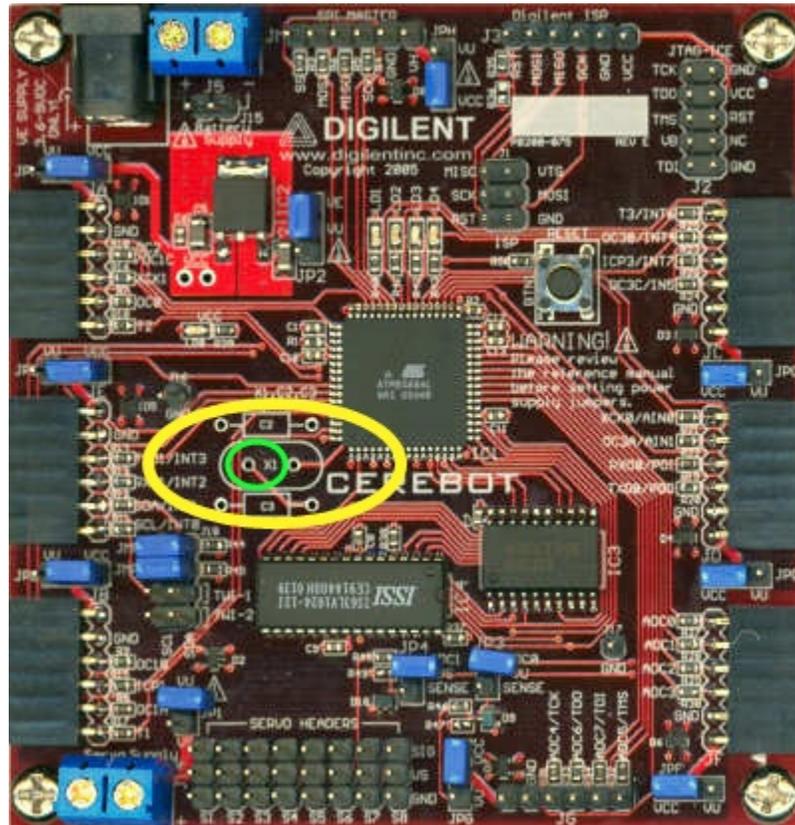


Figure 1
The Clock-Source Solder Pad (Circled in Green)

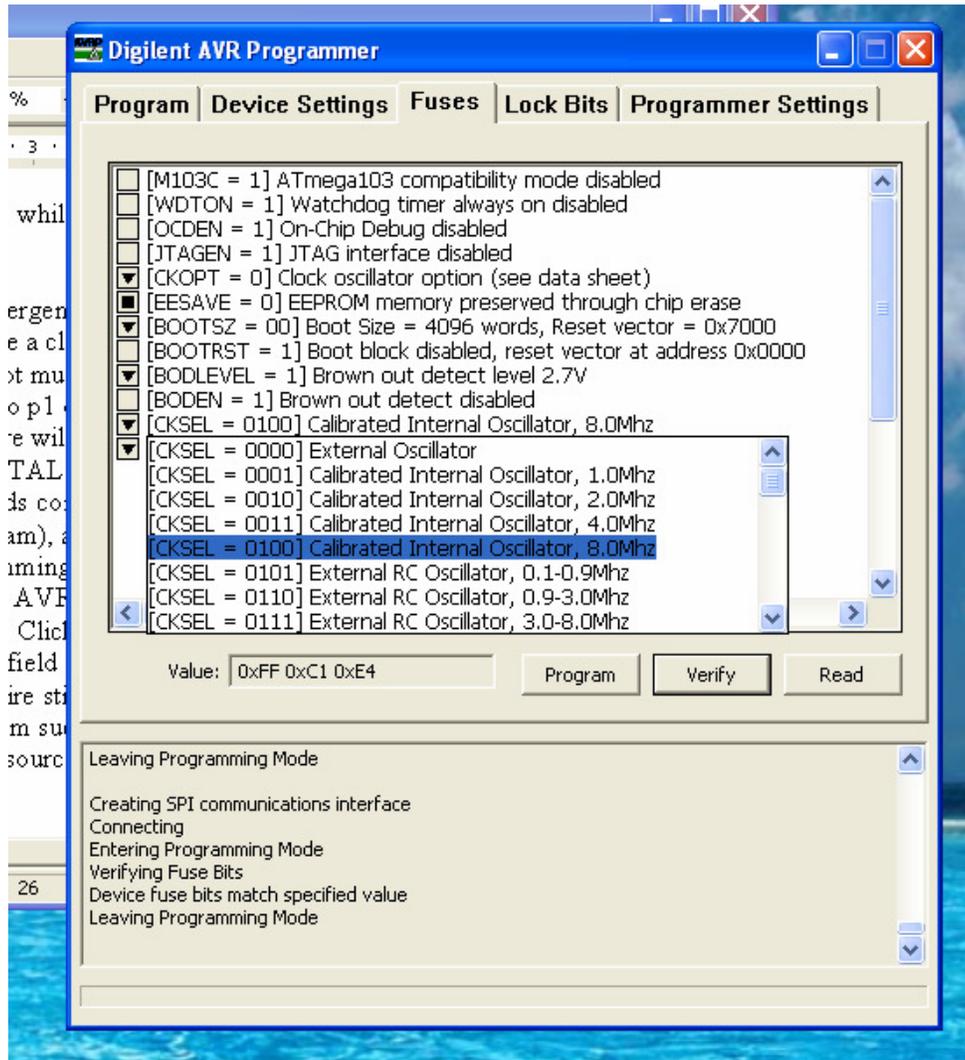


Figure 2
Select This Clock Source

Using a Working Cerebot to Supply a Clock Signal

1. Power up both the inoperative and working Cerebot boards and launch the Digilent AVR Programmer software.
2. Connect a programming cable to the working Cerebot board.
3. Load the Emergency_Clock hex file onto the working Cerebot (see the source code, below).
4. Connect the programming cable to the inoperative Cerebot board.
5. Connect the ground on the working Cerebot to the ground on the inoperative Cerebot.
6. Insert a wire into the pin 1 position of connector JD on the working Cerebot.
7. Locate the two solder pads on the inoperative board that are surrounded by an oval and marked X1 (this is right next to JE). Touch the free end of the wire to the solder pad farthest

from the AVR chip on the inoperative board (see Figure 1). This wire will now carry a 4Mhz clock signal from the working Cerebot to the inoperative Cerebot.

8. In Digilent AVR Programmer, select the Fuses tab and select [CKSEL = 0100] Calibrated Internal Oscillator, 8.0Mhz (see Figure 2).
9. With the wire still touching the solder pad, click Program. The device should program successfully and run without the external clock source.

Source Code for the Emergency_Clock Program

The following is the source code for creating the necessary hex file using WinAVR development tools. For more information on creating a project using WinAVR, see "Cerebot First Program Reference Design" at www.digilentinc.com.

```

/*****
/*
/* Emergency_Clock.c -- Generates ~ 4 mhz clock from cerebot
/*
/*
/*****
/* Author: Chris Keeser
/* Copyright 2005 Digilent
/*
/*****
/* Module Description:  Toggles all the PORTE pins at a rate of ~ 4mhz
/* any pin on cerebot's port JC or JD (besides power and ground)
/* can be used to supply a clock signal to a defunct cerebot board
/*
/*
/*****
/* Revision History:
/* created: (ChrisK) 9/1/05
/*
/*
/*****

/* ----- */
/*          Include File Definitions          */
/* ----- */

#include <avr/io.h>          // Allows use of PORTX, PINX, DDRX etc.

int main(void)
{
    // Set up port E as outputs
    DDRE = 0xFF;

    // loop forever
    while(1)
    {
        // bring pins high
        PORTE = 0xFF;

        // bring pins low
        PORTE = 0x00;
    }
}

```