How to recover from a "Fuse Brick" with 9X radio

By Telemachus (Thanks to GDENTON3 for first reporting this)

What is a fuse brick?

In the AVR forums, there are several examples of a microcontroller (the little microchip that runs the 9X radio, also called an Atmega64) suddenly not responding to programming, after having been programmed successfully one or more times.

When you hook up your programmer and try to access it through AVR dude, you get the following error:

Code:

```
avrdude: AVR device not responding
avrdude: initialization failed, rc=-1
avrdude: Yikes!  Invalid device signature.
avrdude: Expected signature for ATMEGA64 is XX XX XX
avrdude: AVR device not responding
 ***failed;
avrdude: verification error, first mismatch at byte 0x0000
          0xca != 0xff
avrdude: verification error; content mismatch
```

Assuming that your connections have not become dislodged (this needs to be checked with a multi-meter), you have likely run into a "fuse brick." This can be caused by any number of reasons, which are out of the scope of this document.

In a nutshell, the microcontroller is now looking for an external clock in order to take programming, and the USB programmers often used to program the 9X do not provide this clock.

What is needed to fix this? (Warning: TECHNICAL TALK)

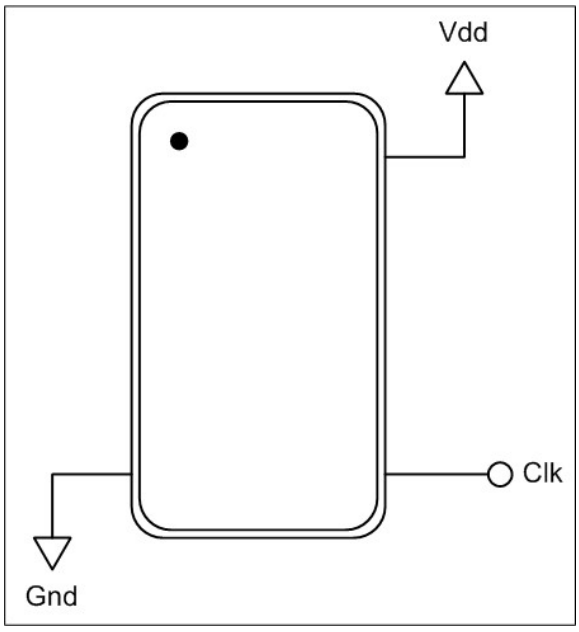
In essence, what needs to be provided is an external clock to get the microcontroller working again. The datasheets for the Atmega64 recommend a "1MHz Oscillator." This needs to be attached to the "XTAL1" pin of microcontroller, or pin 24. There are a few different ways to do this. Another working microcontroller, such as another Atmega or a Microchip PIC can be used to provide the clock, but this will require configuring the microcontroller to output a 1MHz clock. For those who are comfortable with microcontrollers, a PIC 12F675 is a cheap (free with samples) solution. It just needs to be set to internal clock at 4MHz, with OSCOUT on, and internal MCLR. Then, Pin 3 will provide the clock needed to clock the Atmega64. No code is needed at all.

Forget all the Mumbo-Jumbo technical talk, HOW DO I FIX THIS?

By far, the easiest solution would be to purchase a 1MHz oscillator from Digikey or similar distributor. These cost about $2 plus shipping:

http://search.digikey.com/scripts/DkSearch/dksus.dll?Detail&name=XC229-ND

These are self-contained oscillator units that don't need to be programmed in any way. They have 4 metals pins on the bottom of them, which will have a pinout similar to the one below (check the datasheet on the one you buy):

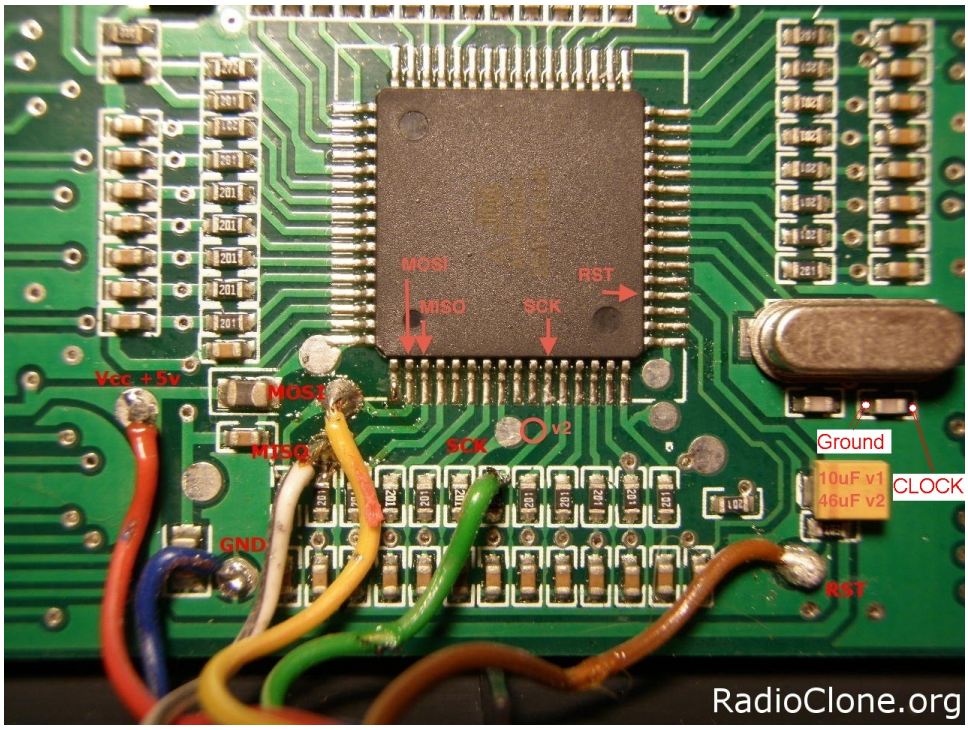


VDD is also known as the "power" pin, and will have to be attached to a 5 volt source. I think it is easiest just to hook it up to the VCC pin that we already hook the programmers up to, as shown in the picture below.

Clk is where the 1MHz signal is output from the oscillator, and this needs to be attached to pin 24 of the ATMEGA64. The spot designated CLOCK in the picture below is probably the easiest access point.

Gnd needs to go to ground. There are many, MANY options for this, but the easiest is probably to connect to the spot labeled GROUND in the picture below.

(Thanks to Rafit for the excellent picture of the 9X access points)



Once the oscillator is connected, you simply connect the 9X to your programmer as you normally would. The oscillator will be powered through the programmer, and will output a 1MHz oscillator signal to the ATMEGA64, and you will be able to access your radio again to reprogram it.

FIX THOSE FUSES

The first thing you should so is write the correct fuses to the radio. These are documented in Thus manual as follows:

Fuses low: 0e
Fuses high:89
Fuses ext: FF

Simply type them in, and hit WRITE next to the fuses (this will vary slightly based on the graphical user interface you use to access AVRdude, but should be obvious when you look at it).

Once the fuses are fixed, you should be able to power off the radio, desolder the oscillator, and then access the programmer without it there to program your firmware. It would also be an option to install a switch on the ground side of the oscillator, so that if this happens again, you just switch on the oscillator and fix the fuses, then switch it off when you don't need it anymore.

Hope this helps!

```
Using AVRDUDE from the command line, assuming you had the smartieparts.com pogo
board programmer (USBasp) the following command will program the fuse values
recommended for gruvin9x running on the stock ATmega64A radio hardware ...

avrdude -p m64 -c USBasp -u -U efuse:w:0xff:m -U hfuse:w:0x89:m -U
lfuse:w:0x0e:m

(Substitute your programmer type for -c, for example: -c usbtiny
or -c stk500v2. Do NOT use these values for the ATmega2560/1! Gruvin.
```