

Troubleshooting DG8SAQ AVR-USB based SI570 Controllers

September 11th, 2008
T. Baier DG8SAQ

There are several controllers based on my AVR firmware out there, e.g. by SDR-KITs or by Tony Parks KB8YIG. As I have received quite a few bug reports, I have decided, to compile this step by step troubleshooting guide.

1. Supported and tested operating systems

AVR-USB has been tested successfully on the following operating systems:

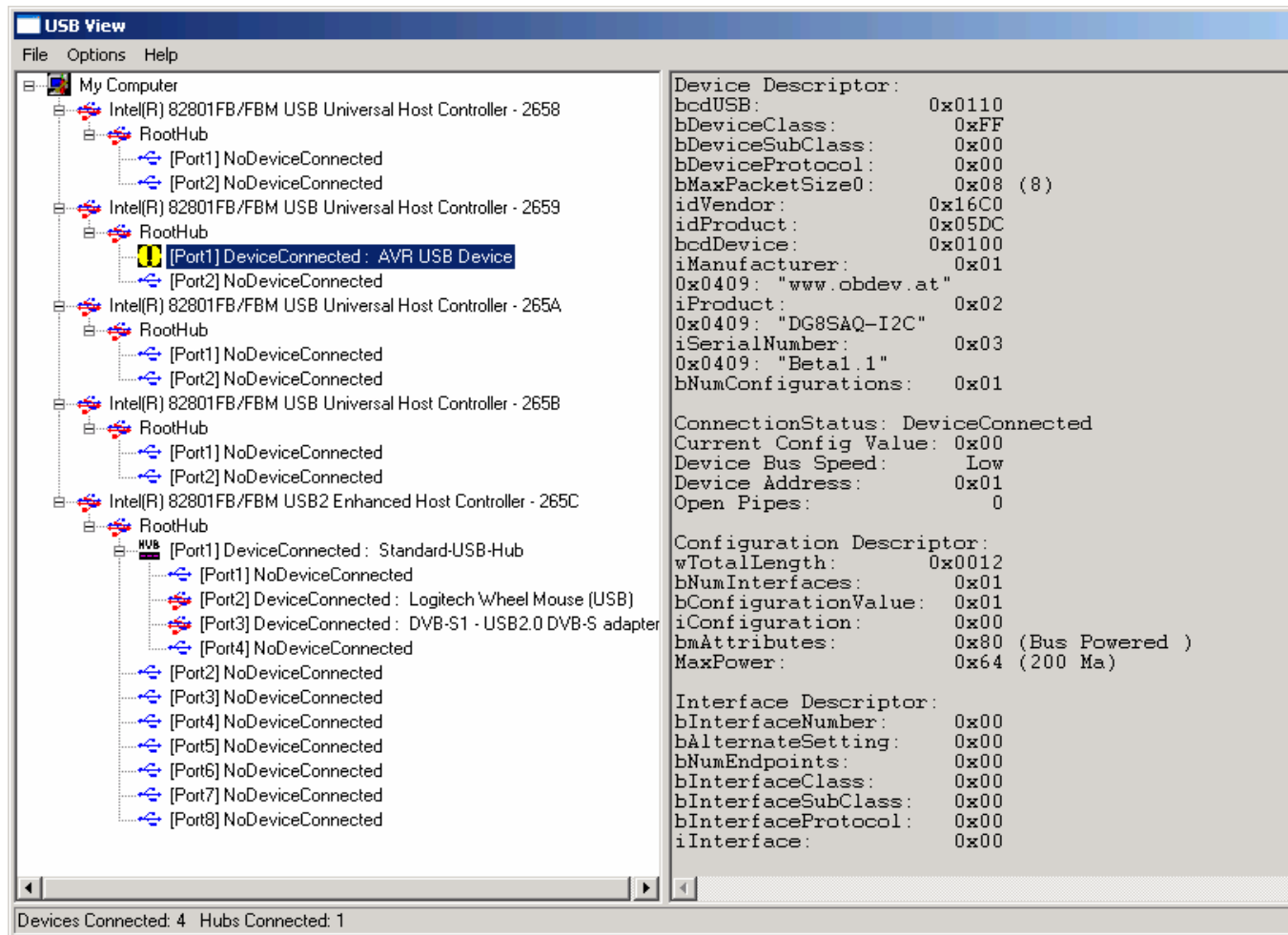
- Windows 98
- Windows ME
- Windows 2000
- Windows XP
- Windows Vista: To date it is unclear to me if both 32 bit Vista and 64 bit Vista have been tested. I do not have access to either Vista OS. LibUSB drivers are provided for both 32 bit and 64 bit and should both work. Note, that in order to install unsigned drivers on Vista 64bit, the sign-check needs to be switched off in the registry:
 1. *click Start, type regedit in the Start Search box, right-click regedit.exe, and then click Run as administrator.*
 2. *Locate, and then click the following key in the registry:
HKEY_LOCAL_MACHINE\Software\Microsoft\Non-Driver Signing*
 3. *Right-click the Policy binary value, and then click Modify.*
 4. *The Value data will appear in the following format: 0000 02
Press DELETE to remove the current value (02 in this example), and then type 0 (the current value will now appear as 00).*
 5. *Click OK, and then quit Registry Editor.*

2. USB connection

In order to test the USB connection, only the powered AVR is required. These steps can be done without SI570 in place and I highly recommend to do this test after partial assembly.

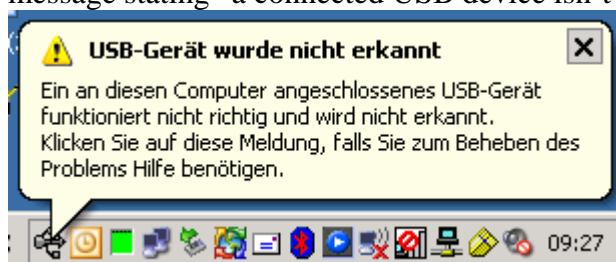
- Plug the device into a free USB port of your PC.
The PC should notify, that new hardware has been found and ask you to install a driver. In this case you may continue to install the driver according to the manual on the SDR-KITs webpage <http://www.sdr-kits.net/> or you may still do the next test.
- For the following connectivity test no drivers need to be installed. All you need is the Microsoft program USBView, which I have put on my webpage, as I couldn't find the Microsoft download any longer: www.mydarc.de/dg8saq/hidden/usbview.zip

Running usbview.exe, you should find your AVR device on the left half of the window, and, if you click on it, you should also obtain the detailed device information on the right.

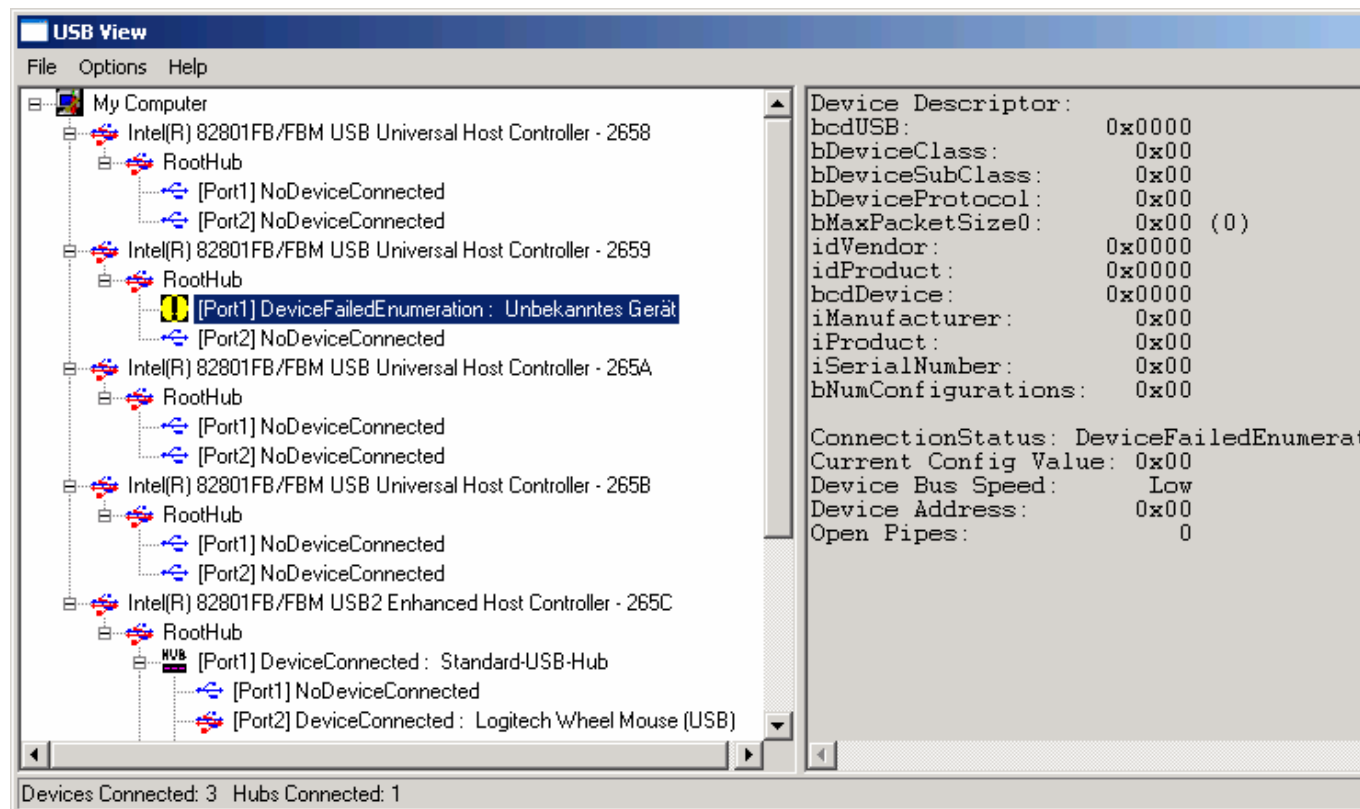


If you see this, the AVR device works properly and you may continue to install the drivers.

If Windows doesn't recognize the AVR device, it will notify you with the following message stating "a connected USB device isn't working properly".



In this case, the usbview screen looks like this:



This indicates a **HARDWARE** problem, not a driver problem. Windows is unable to read the device information from the AVR device, which means the device simply doesn't work.

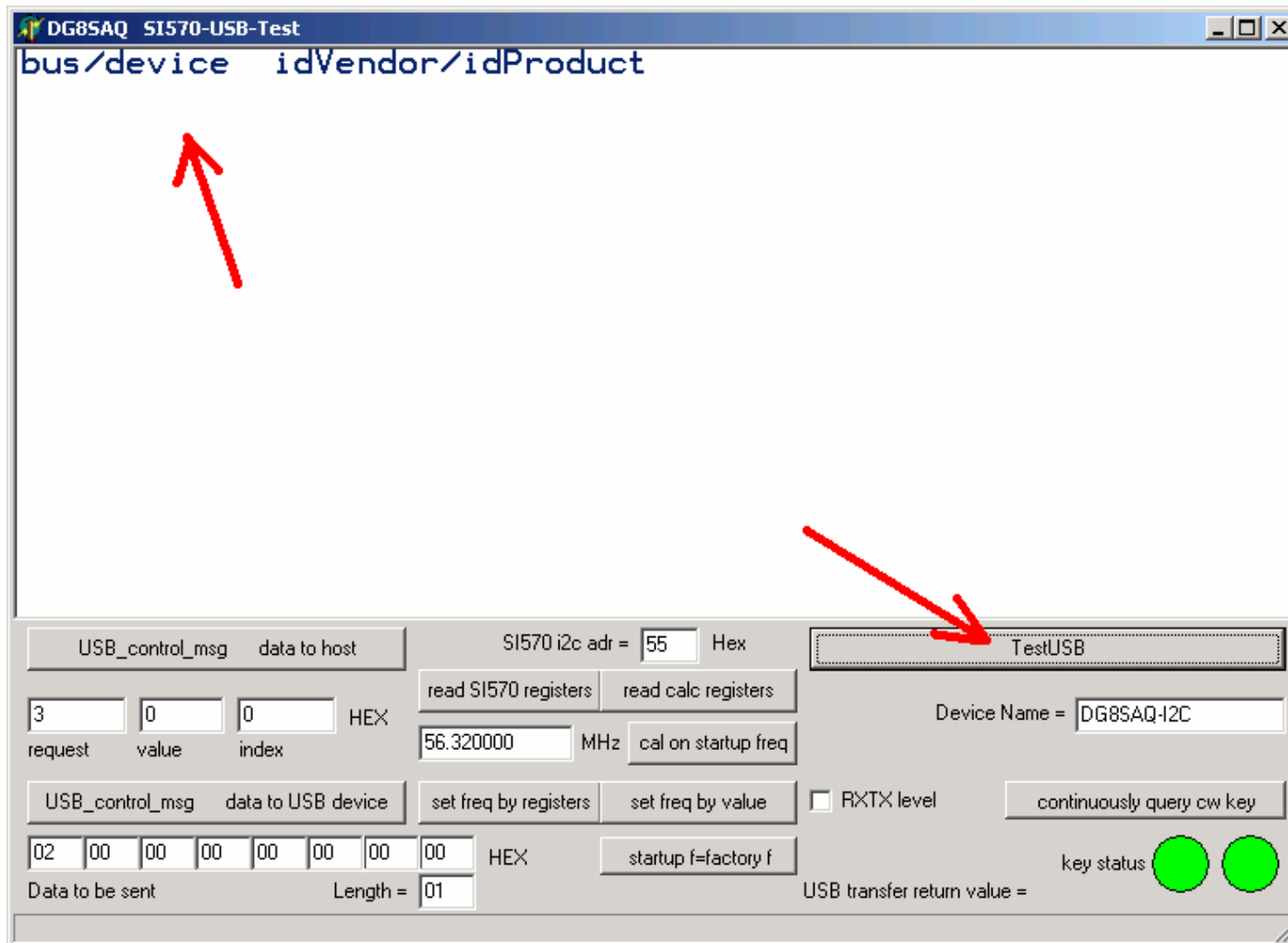
Possible causes:

- AVR not powered up or too low supply voltage for AVR. AVR needs 5V DC. Beware, that the USB interface must provide up to 200 mA if the SI570 is connected and powered from USB. Also note, that for bus powered USB hubs the limit is 100 mA.
- AVR not properly in socket, e.g. bent pins
- Wrong orientation of Zener diodes or wrong or missing resistors around the USB connector
- If no USB connector but a cable is used, then swapped D+ and D- signals at the USB connection
- Broken or corrupt AVR.
- Broken PC USB interface.

3. I2C connection to SI570

If the AVR device is recognized by the PC and the drivers are installed properly, but the SI570 fails to tune, then you should continue troubleshooting with the program SI570_USB_Test provided in the firmware zip archive from my webpage in the zip folder `si570\hostware\SI570_USB_Test.exe`.

Start the program and press the button TestUSB. Do not modify any input fields!



If you see the above almost blank following screen without device information, but you successfully performed the USB connectivity test, then you have a driver problem. Install or reinstall the LibUSB driver provided in my firmware zip archive.

If the device and the driver works properly, then you should see the following screen with device details:

DG8SAQ SI570-USB-Test

```

bus\device   idVendor\idProduct
bus-0\...\libusb0-0001--0x16c0-0x05dc   0x16C0/0x05DC
- Manufacturer : www.obdev.at
- Product      : DG8SAQ-I2C
- Serial Number: Beta1.1
wTotalLength:      18
bNumInterfaces:    1
bConfigurationValue: 1
iConfiguration:    0
bmAttributes:      80h
MaxPower:          100
  bInterfaceNumber: 0
  bAlternateSetting: 0
  bNumEndpoints:    0
  bInterfaceClass:   0
  bInterfaceSubClass: 0
  bInterfaceProtocol: 0
  iInterface:        0

```

USB_control_msg data to host SI570 i2c adr = 55 Hex TestUSB



3 0 0 HEX read SI570 registers read calc registers

request value index 56.320000 MHz cal on startup freq

Device Name = DG8SAQ-I2C

USB_control_msg data to USB device set freq by registers set freq by value RXTX level continuously query cw key

02 00 00 00 00 00 00 00 HEX startup f=factory f

Data to be sent Length = 01 key status  

USB transfer return value =

Next, check communication with the SI570 by pressing the “read SI570 registers” button. You should see the following if a successful I2C communication with the SI570 has been established:

The screenshot shows the DG8SAQ SI570-USB-Test application. The top window displays the following register values:

```

Register 7 = 05 Hex
Register 8 = 42 Hex
Register 9 = B5 Hex
Register 10 = 76 Hex
Register 11 = 9D Hex
Register 12 = D4 Hex
-----
RFREQ      = 43.3414591103792
N1         = 21
nN1        = 22
HS_DIV     = 0
nHS_DIV    = 4
frequency  = 56.287257436701 MHz

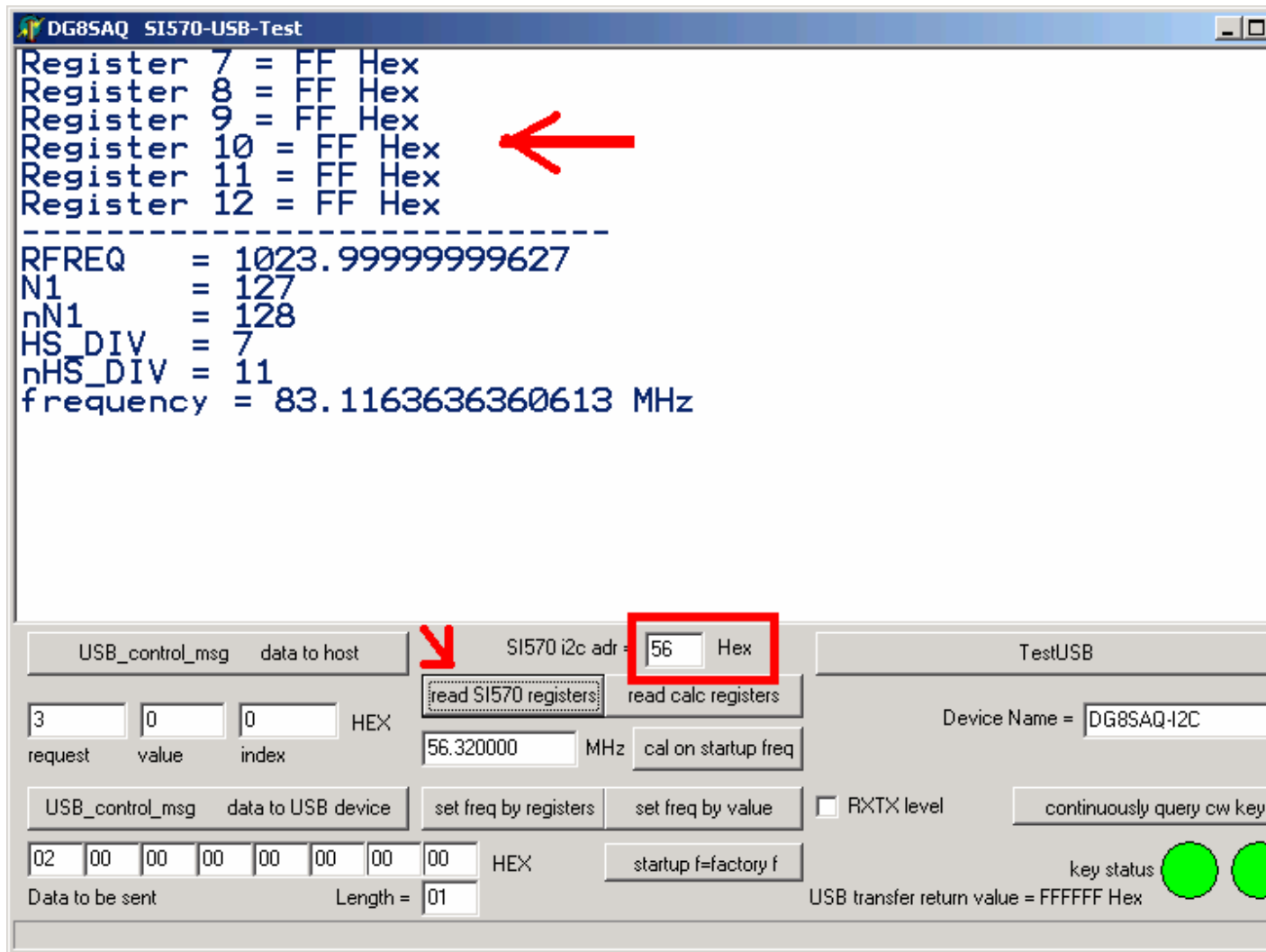
```

The bottom control window shows the following fields and buttons:

- SI570 i2c address: 55 Hex (highlighted with a red box and arrow)
- Buttons: read SI570 registers, read calc registers, set freq by registers, set freq by value, startup f=factory f
- Fields: request (3), value (0), index (0), HEX, 56.320000 MHz, cal on startup freq, Device Name = DG8SAQ-I2C, RXTX level (checkbox), continuously query cw key, key status (two green circles), USB transfer return value = B54205 Hex

You should see the hexadecimal register contents of the SI570, from which the approximate startup frequency is calculated and displayed below.

If I2C communication fails, you might see the following screen instead:



The registers seem to contain all the values FF hexadecimal. Actually, the SI570 is not read at all in this case, but you only see the high level of the SDA pullup resistor. I have provoked this error by entering a wrong I2C address in the field surrounded by the red box. If the pullup resistor is missing, then you might also read back all zeros in such a failure case.

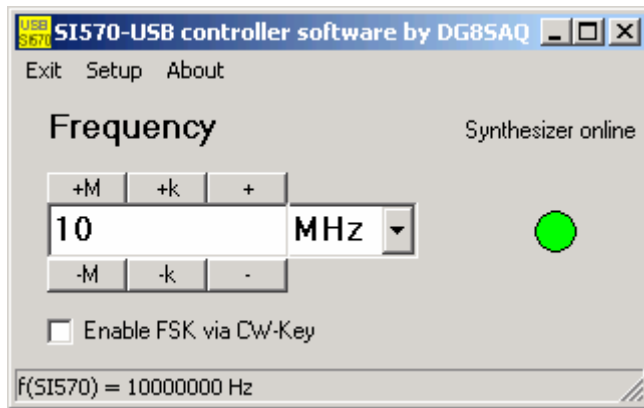
Possible causes for failing I2C communication:

- wrong I2C address. Consult the SiLabs webpage to find out the I2C address for your type number.
- missing pullup resistors at the SI570 SCL or SDA lines.
- poor solder connections underneath the SI570. Resolder!
- missing power supply for SI570
- connection break between AVR and SI570. Resolder! Check if AVR has bent pins.

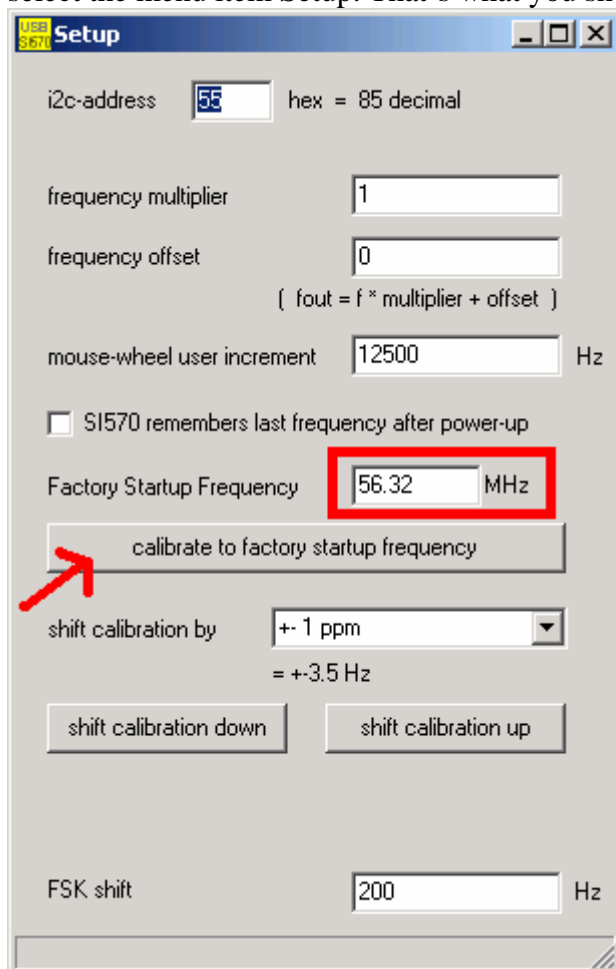
DO NOT USE ANY OTHER BUTTONS ON SI570_USB_Test UNLESS YOU EXACTLY KNOW WHAT YOU ARE DOING. YOU MIGHT CORRUPT THE AVR EEPROM.

4. Tuning the SI570

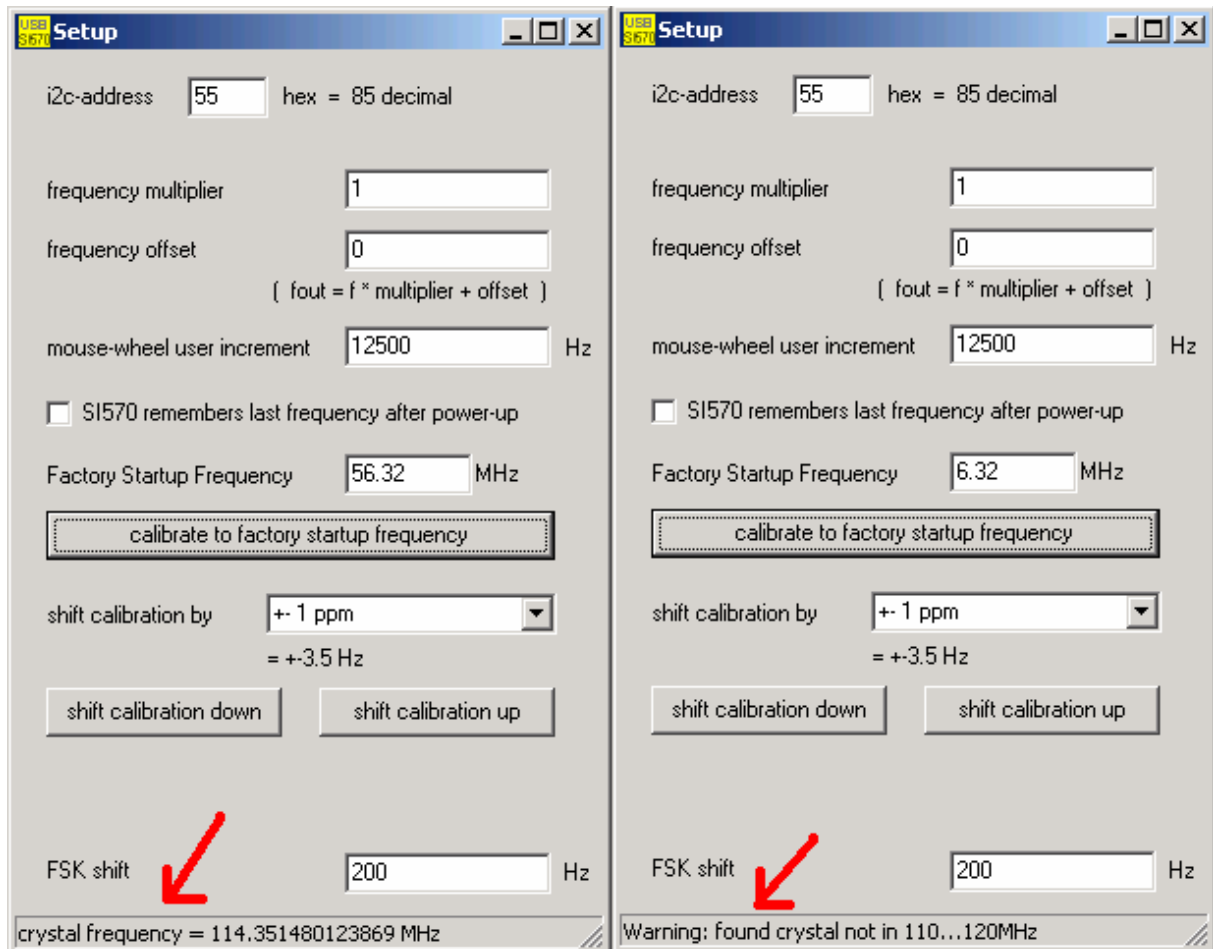
To control the SI570, use the program USB_Synth also available from my webpage:



Use the pushbuttons or the mouse wheel to control the frequency. If there is an offset between displayed frequency and the actual frequency, you need to calibrate the device. To do so, select the menu item Setup. That's what you should see:



Make sure to enter the correct factory startup frequency into the field with the red box around and press the “calibrate to factory startup frequency” button. Wait for a few seconds until the status line at the bottom display success or a warning:



If a warning is issued, you have very likely entered a wrong startup frequency. A wrong calibration value has then be stored in the AVR EEPROM which might cause the device to fail to tune. Recalibrate until success is displayed.

Note: Upon calibration the whole AVR EEPROM is erased. Thus calibration is a means to recover devices with corrupt EEPROM. This feature has only been implemented on the latest version of USB_Synth.

Possible causes for device failing to tune:

- Corrupt EEPROM => calibrate device
- Corrupt calibration => calibrate device
- failing communication => check troubleshooting steps 1...3

I hope this document helps you to solve any proplems with devices using my AVR USB firmware.

73's Tom DG8SAQ