Silicon Labs Synthesizer Operating Guide

Portland State Aerospace Society
12-Jul-2023
V1.1

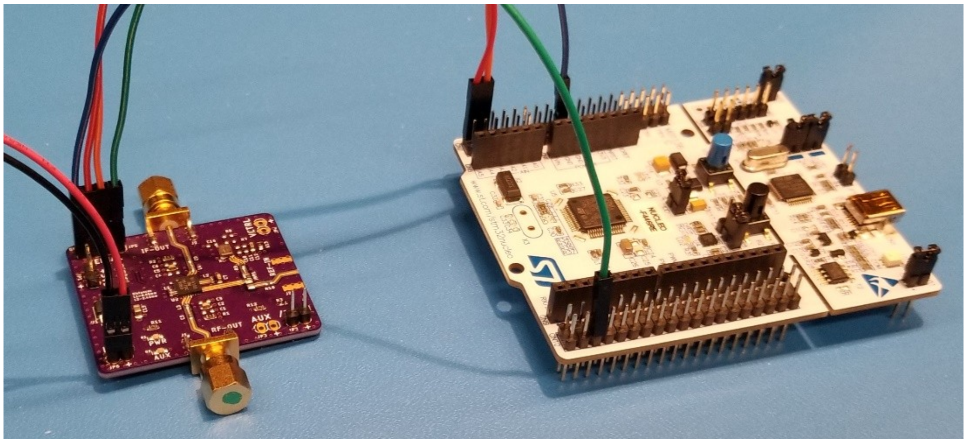


Figure 1: Si4123 eval board left, NUCLEO-F446RE dev board right

Introduction

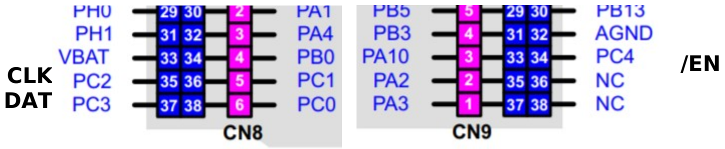
This operating guide evaluates a custom Silicon Labs Si41xx evaluation board with a STM32 Nucleo-64 development board which includes the STM32F446RE; a 32-bit ARM Cortex-M4 MCU (NUCLEO-F446RE). The NUCLEO-F446RE development board is used for programming the Si41xx synthesizer over a 3-wire serial interface. The NUCLEO-F446RE board uses a real-time operating system, ChibiOS, to setup the synthesizers registers. A a serial communication program such as Minicom is used to interactively operate the setup procedure after startup.

The Silicon Labs Si41xx synthesizer family is available in five variations with up to three synthesizers. However, RF1 and RF2 share an output. See the end of this document for supplemental information, datasheets, and GitHub repos with EAGLE CAD files of the Si4123 evaluation board; Figure 1 shown above.

Connect Boards

1. Connect the serial interface between the Si41xx evaluation board and the NUCLEO-F446RE development board. Refer to Figure 2 in the Appendix for the complete NUCLEO-F446RE pin layout.

Si41xx	Nucleo-64
/EN	PC4
CLK	PC2
DATA	PC3
GND	GND



2. **ATTENTION:** Connect the Si41xx board to power. The **Si4123 evaluation board shown in Figure 1 above accepts ~3.5V to 10V max due to incorporating a LDO regulator**; the TI LP38692 3.3V fixed LDO. The **Si41xx IC operates within 2.7V to 3.6V**; however, a TCXO on the evaluation board may have strict voltage requirements.
3. Connect the micro USB port on the NUCLEO-F446RE board to a PC with Linux.
4. On the Si41xx board the red AUX LED should be on. This is the lock detect output; the red LED indicates when the PLL has lost lock.

Git Checkout and Programming

1. If the STM32F446 board is already programmed with the Si41xx app then the Si41xx board will most likely already be programmed, and you can skip this section. The Si41xx app is designed to program the Si41xx board when the STM32F446 board is powered on.
2. In a terminal install the dependencies; E.g. Debian based.

```
sudo apt install git make gcc-arm-none-eabi gdb-multiarch openocd
```

3. Type the following to checkout c3_synth and program the Si41xx board.

```
git clone https://github.com/oresat/oresat-firmware.git
cd oresat-firmware
git checkout c3_synth
git submodule update --init
cd src/f4/app_si41xx
make clean
make
make write
```

4. The black reset button on the STM32F446 dev board can be pressed at any time to reprogram the Si41xx dev board.
5. The Si41xx dev board should now be programmed, and the red LED should be off.

NOTE: Lock detect output will remain high, red LED on, if any of the synthesizers (RF1, RF2, or IF) have failed. This includes variations of the Si41xx family which excludes those synthesizers. It's good practice to not program any of the respective N or R registers if the variation in use don't include those synthesizers. In addition, don't enable an output if that output is not available on the variation in use. See page 21 of the Si41xx datasheet for register details; link in 'References' section.

Configure Minicom Console

1. If Minicom has already been configured for your environment, then skip this step. - In a terminal install Minicom; E.g. Debian based.

   ```
   sudo apt install minicom
   ```

2. Type the following to find the appropriate port.

   ```
   sudo dmesg | grep tty.*USB
   ```

3. The output may look something like this, and the correct port is ttyACM0 in this example.

   ```
   [ 2246.756830] cdc_acm 2-2:1.2: ttyACM0: USB ACM device
   ```

4. Type the following to start the Minicom terminal program.

   ```
   sudo minicom -s
   ```

5. Go to the 'Serial port setup'.
6. Press 'a' to change the serial device and enter the correct port; example below.

   ```
   A -    Serial Device      : /dev/ttyACM0
   ```

7. The default bitrate of both the Si41xx app and Minicom are 115200 bps, so the settings here are sufficient.

   ```
   E -    Bps/Par/Bits       : 115200 8N1
   ```

8. Press enter twice to return to the main menu.
9. Select and press enter on 'Save setup as dfl' to save configuration as default.
10. Select and press enter on 'Exit' to start Minicom.
11. While in Minicom use the key combination 'CTRL-A X' to exit.

Control the Si41xx Dev Board

1. In a terminal type the following to start Minicom if it's already configured.

   ```
   sudo minicom
   ```

2. Once connected and pressing enter a few times you will see the ch> prompt. Type ‘?’ and press enter to view the Si41xx app commands.

```
Available commands:
    reg:   Update registers,Usage reg <register address> <register value>
    rf1:   Update RF1 registers,Usage: rf1 <frequency in KHz> <Phase detector in KHz>
    rf2:   Update RF2 registers,Usage: rf2 <frequency in KHz> <Phase detector in KHz>
    if:   Update IF registers,Usage: ifr <frequency in KHz> <Phase detector in KHz>
    ?:   provides list of commands
```

3. Use the ‘rf1’, ‘rf2’, and ‘if’ commands to change the output frequency of the respective output. Since the RF1/2 share the same output, the register written to last will determine the chosen synthesizer and turn off the other. The synthesizers can tune about +/-15%.
4. To shut-down an output use the ‘reg’ command with the appropriate register setting.

```
reg 2 0b00     Disable all outputs
reg 2 0b01     Disable IF output
reg 2 0b10     Disable RF1/2 output
reg 2 0b11     All enabled
```

Typical use example

A typical use case is to setup the synthesizer to output 436.5 MHz, and 1265 MHz signals simultaneously on IF and RF1 ports respectively. Following are the commands to achieve that.

```
ch> reg 2 0b11
ch> rf1 1265000 500
ch> if 436500 500
```

The output signal should now be seen on each port. The output signal strength for each is expected to be approximately -8 dBm to +1 dBm, with a typical level being -3 to -4 dBm. See the data sheet for details.

Appendix

Silicon Labs Si41xx Synthesizer Family Summary:

Part	RF1	RF2	IF
Si4133	X	X	X
Si4123	X	-	X
Si4122	-	X	X
Si4113	X	X	-
Si4112	-	-	X

Si41xx Main Register Configuration Summary:

Bit	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	0	0	0	0	AUXSEL [1:0]		IFDIV [1:0]		0	0	0	0	LPWR	0	Auto PDB	Auto Kp	RF PWR	0
Default	0	0	0	0	11		00		0	0	0	0	0	0	0	1	0	0

NUCLEO-F446RE Pin Layout:

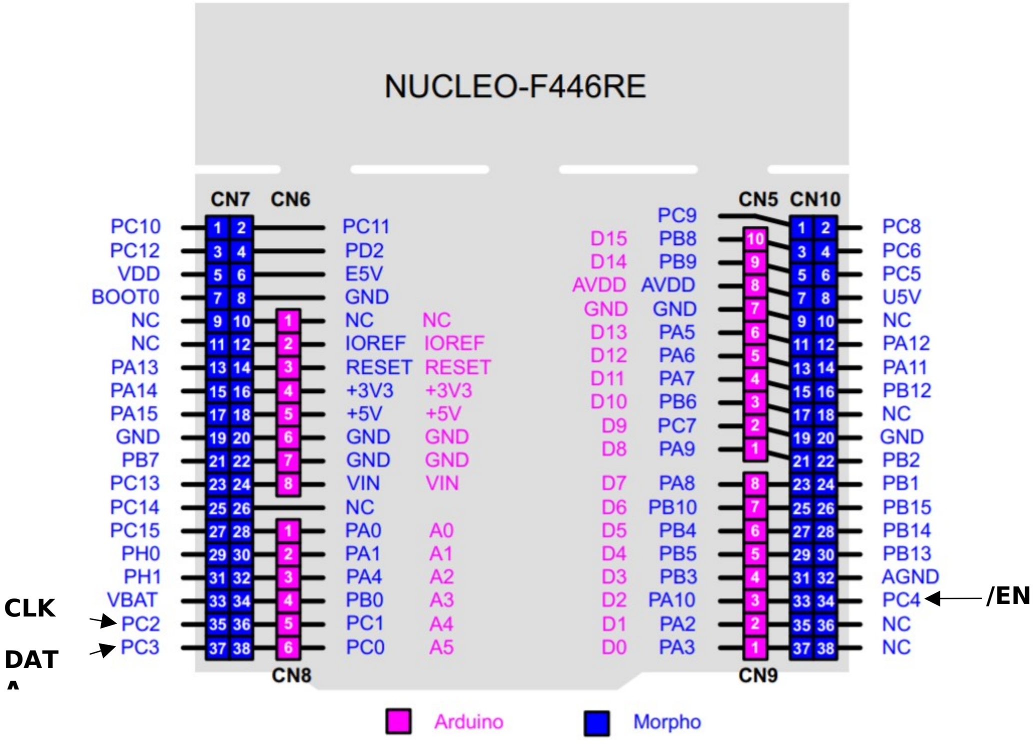


Figure 2: NUCLEO-F446RE pin layout with associated Si41xx connections

GitHub Repos

Si41xx Synthesizer Firmware:
https://github.com/oresat/oresat-firmware/tree/c3_synth/src/f4/app_si41xx

Si41xx EAGLE CAD Schematics/Layouts (synthesizer* files):
https://github.com/oresat/oresat-c3-rf/tree/master/eagle

References

Silicon Labs Si41xx Synthesizer Datasheet:
https://www.skyworksinc.com/-/media/Skyworks/SL/documents/public/data-sheets/si4133.pdf

STM32 Nucleo-64 Development Board (NUCLEO-F446RE):
https://www.st.com/en/evaluation-tools/nucleo-f446re.html

Document History

- V1.0 – 2019-10-12 – Original document preparation.
  - Evangelos Mastrogiannis, KJ7AOG
  - Malay Das, KG7ZVV
- V1.1 – 2023-07-12 – Minor improvements for clarification and updates.
  - G.N. LeBrasseur, KJ7SU