Silicon Labs Synthesizer Operating Guide

Portland State Aerospace Society

Evangelos Mastrogiannis – KJ7AOG

Malay Das – KG7ZVV

A circuit board

Description automatically generated

Si41xx eval board left, Nucleo-64 dev board right

# Introduction

This operating guide evaluates a custom Silicon Labs Si41xx evaluation board with a STM32 Nucleo-64 development board with the STM32F446RE (NUCLEO-F446RE). The NUCLEO-F446RE development board is used for programming the Si41xx synthesizer over a 3-wire serial interface. The Nucleo-64 board uses a real-time operating system, ChibiOS, to program the synthesizer. A remote serial console program such as Minicom can be used to interactively program and control the synthesizer as well.

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

# Connect Boards

1. Connect the serial interface between the Si41xx evaluation board and the NUCLEO-F446RE development board.

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

1. Connect the micro USB port on the NUCLEO-F446RE board to a PC with Linux.
2. Connect the Si41xx board to a battery (**3.5V to 10V max**).
3. On the Si41xx board the red AUX LED should be on. This is the lock detect output; the red LED signifies when the PLL has lost lock.

# Git Checkout and Program

1. If the STM32F446 board is already programmed with the Si41xx app then proceed to step (4).
2. In a terminal install the dependencies; E.g. Debian based.

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

1. 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 |

1. The black reset button on the STM32F446 dev board can be pressed at anytime to reprogram the Si41xx dev board.
2. The Si41xx dev board should now be programmed. The red AUX LED should have turned off if the default output frequencies have been programmed correctly.

# Configure Minicom Console

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

|  |
| --- |
| sudo apt install minicom |

1. Type the following to find the appropriate port.

|  |
| --- |
| sudo dmesg | grep tty.\*USB |

1. 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 |

1. Type the following to start the Minicom configuration.

|  |
| --- |
| sudo minicom -s |

1. Go to the ‘Serial port setup’.
2. Press ‘a’ to change the serial device and enter the correct port; example below.

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

1. Press enter twice to return to the main menu.
2. Select and press enter on ‘Save setup as dfl’ to save configuration as default.
3. Select and press enter on ‘Exit’ to start Minicom.
4. 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 |

1. 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 |

1. 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%.
2. To power down an output use the ‘reg’ command with the appropriate register data.

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

# Synthesizer Supplemental

**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 |

**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 |

# 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.silabs.com/documents/public/data-sheets/si4133.pdf>

**STM32 Nucleo-64 Development Board (NUCLEO-F446RE):**

<https://www.st.com/en/evaluation-tools/nucleo-f446re.html>