# MAX5861 Evaluation Kit

# Evaluates: MAX5861

## General Description

The MAX5861 evaluation kit (EV kit) contains a MAX5861 high-density downstream cable SCQAM and OFDM modulator that integrates a digital up-converter (DUC) and a RF Digital to Analog Converter (RF-DAC). The MAX5861 DUC performs SCQAM and OFDM mapping, pulse shaping, and digital RF up-conversion with full agility and then drives a 14-bit 5Gsps RF-DAC. The device digitally synthesizes RF signals with up to 160 DOCSIS®-compliant 6MHz QAM channels (or up to 120 8MHz QAM channels) on a single RF port at frequencies from 47MHz to 1218MHz. The MAX5861 device provides up to 6 channels of OFDM IFFT processing. Each of the OFDM channels provide up to 192MHz of bandwidth, for a combined potential 1152MHz of modulation bandwidth. The MAX5861 can support up to six blocks powered on at the same time, where a block is defined as an OFDM channel or a 32-channel SCQAM block. The MAX5861 EV kit provides a complete system solution for high-density SCQAM and OFDM modulation targeting the DOCSIS 3.1 solution with very low power dissipation.

The MAX5861 EV kit connects to the FMC connector on the Xilinx VC707 evaluation kit, allowing the VC707 to communicate with the MAX5861's three input ports and various control signals.

The EV kit includes Windows XP®-, Windows Vista®-, Windows® 7/8-compatible software that provides a simple graphical user interface (GUI) for configuration of all of the MAX5861 registers, control of SPI interface, control of the VC707 FPGA and temperature monitoring.

## Benefits and Features

- Evaluates Up to 160 SCQAM Channels and/or 6 OFDM Channels
- Up to 6 Blocks Powered at Once, Where a Block is Defined as an OFDM Channel or a 32-Channel SCQAM Block
- Single 5.0V Input Voltage Supply
- Maximum 4.9152Gsps Update Rate
- Direct Interface with Xilinx® VC707 Data Source Board if Desired
- Windows XP-, Windows Vista-, and Windows 7/8-Compatible Software
- On-Board SPI Interface Control for the MAX5861
- On-Board SMBus Interface Control for the MAX6654 Temperature Sensor
- GUI Controls for VC707 Operation
- Pseudo Random Bit Sequence (PRBS) Test Pattern Files
- Proven 10-Layer PCB Design
- Fully Assembled and Tested

***Ordering Information* appears at end of data sheet.**



*19-7677; Rev 0; 6/15*

maxim integrated™

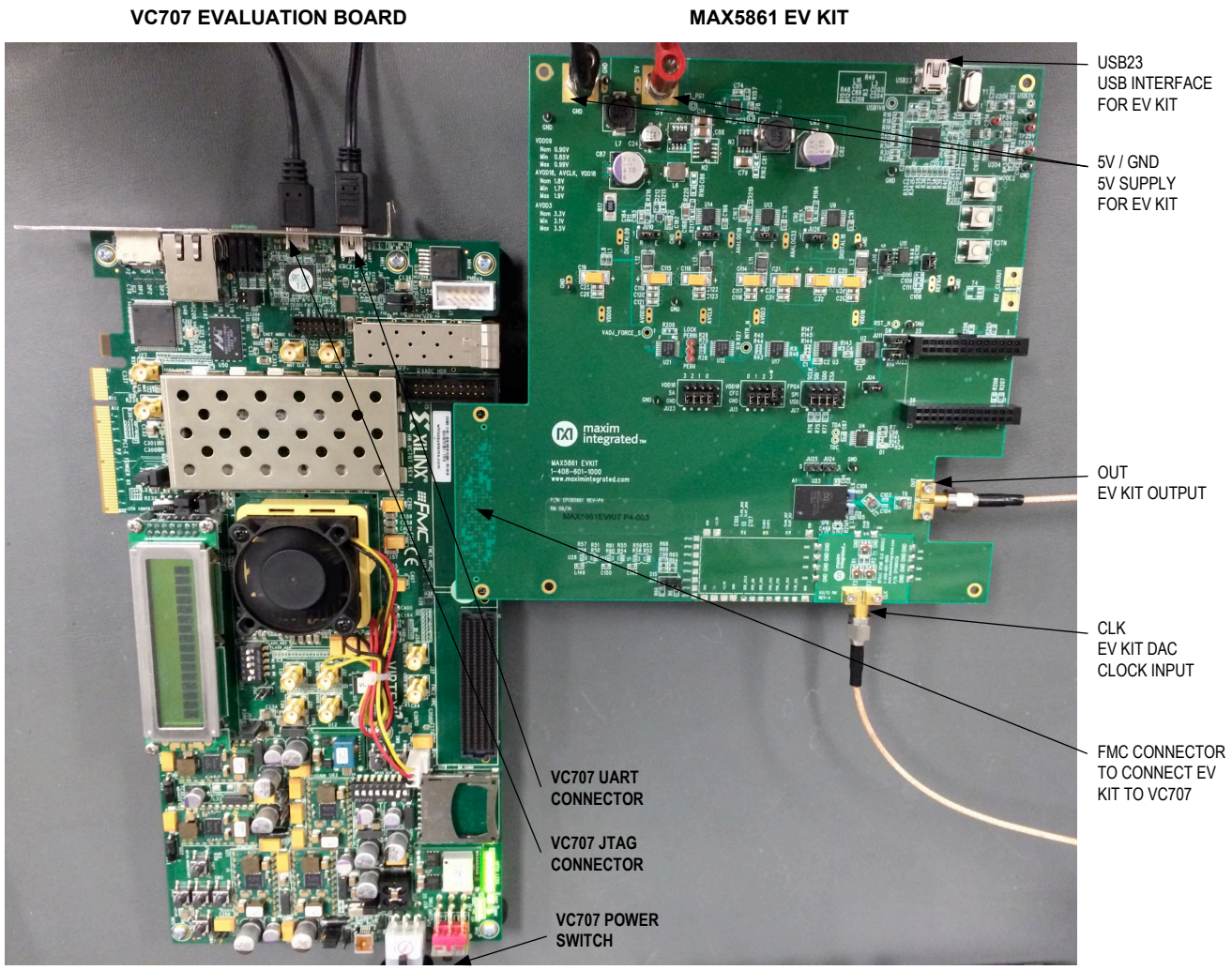



*Figure 1. MAX5861 EV Kit System with MAX5861 EV Kit and Xilinx VC707 Evaluation Board*

## MAX5861 EV Kit Files

| FILE | DECRIPTION |
|---|---|
| MAX5861EVKITSoftwareController.exe | Application program |
| ConfigurationLoadFiles | Directory with sample register configuration files to load into the MAX5861 for evaluation |
| FPGAConfigurationFiles | Directory with sample FPGA configurations files and memory patterns to load into the VC707 |
| Screenshots | Directory with sample spectrum analyzer screenshots for reference of expected performance on set of configurations |
| WindowsDriverFiles | Directory with USB supporting files |
| MAX5861ConfigurationScripts | Perl scripts and supporting files to generate new configuration files to load into the MAX5861 – see Readme.txt for details |
| MAX5861RegMap.txt | Register definition file used by the MAX5861EVKITSoftwareController for register definition display |
| Miscellaneous DLLs to include ftd2xx.dll, DTD2XX_NET.dll, libMPSSE.dll and MaximStyle.dll | Supporting DLL files for software operation |

## Quick Start

### Required Equipment

Before beginning, the following equipment is required:

- Windows XP, Windows Vista or Windows 7/8 PC with a spare HS USB port (3 USB ports if using VC707 with two of those being HS USB ports – if only 2 ports are available, the VC707 JTAG can be used first to program the BIT file into the FPGA and then it can be changed to the UART for communication)
- USB 2.0 cable for the MAX5861 EV kit, USB A male to USB Mini (supplied with kit)
- One 5.0V, minimum 3A DC power supply with banana jack cables to connect to 5V and GND to supply power to the MAX5861
- One signal generator with low-phase noise and low jitter for clock input signal to the RF-DAC at +19dBm (e.g., Rohde & Schwarz SMF100A) with SMA cable to connect to J1
- Bandpass filters for the RF-DAC clock input signal (Optional)
- One high-performance spectrum analyzer (e.g., Agilent PXA, Agilent PSA, Rohde & Schwarz FSU, or better) with SMA cable to connect to OUT
- One Xilinx Virtex 7 VC707 evaluation kit with separate USB micro and USB mini cables as well as a power cable to connect the VC707 to an outlet – for use as an external data source (Optional)

### Procedure

The MAX5861 EV kit is fully assembled and tested. Follow the steps below to verify board operation. **Caution: Do not enable the outputs of the power supplies or signal sources until all connections are completed.**

**Note:** In the following section(s), software-related items are identified by bolding. Text in **bold** refers to items directly from the EV kit software. Text in **bold and underline** refers to items from the Windows operating system.

1) Verify that the MAX5861 EV kit shunts are configured in their default state (the EV kit board indicates installation and position of jumper with a ● – see Table 1 for definitions of each of the jumpers).
2) Connect the USB cable from the PC HS-USB to the MAX5861 EV kit board USB23.
   i) Install the driver for the FTDI device if it is not automatically detected and installed by the PC.
   ii) The EVKIT USB ports can take a few minutes to fully enumerate. If the ports have not enumerated yet, the error window shown in Figure 5 will appear.
3) Set the DC power supply to 5.0V and disable the power-supply output.
4) Connect the 5.0V power-supply output to the MAX5861 EV kit 5V and GND banana jack

connectors.

5) Set the clock signal generator to the desired clock frequency (2457.6MHz) and power level at +19dBm and disable the output.
6) Connect the clock signal generator to the J1 SMA connector.
7) Connect the spectrum analyzer to the EV kit SMA connector labeled OUT.
8) <<If using optional VC707 board for SCQAM input data or OFDM data>>
   i) Connect the VC707 evaluation board per the User's Manual without turning the switch on.
   ii) Connect power supply.
   iii) Connect a USB cable between the PC and JTAG connection.
   iv) Connect a USB cable between the PC HS-USB and the UART connection.
   b) Ensure Xilinx's Impact Tools are installed on the PC and note the location of impact.exe file if programming the FPGA through the MAX5861 EV kit software GUI.
   c) Connect VC707 board to MAX5861 EV kit.
9) Install the EV kit software on your computer by running the setup.exe program. The recommended location for the installation software is C:\MaximIntegrated\MAX5861EVKIT to avoid file permission issues. The application files are copied and icons are created in the Windows **Start | Programs** menu.
10) Enable the MAX5861 EV kit DC power supply.
11) Enable the clock generator output.
   i) Approximate current draw on the 5V supply upon power-up should be 1.4A, depending on the CFG jumper settings.
12) <<If using optional VC707 board for SCQAM input data or OFDM data>>
   i) Power on the VC707 board.
13) Start the MAX5861 EV kit software by opening the icon in the **Start | Programs** menu under **Maxim Integrated**. The EV kit software window will display a splash screen, as shown in Figure 2, while the software is loading. When the software load is complete, the software GUI will display as shown in Figure 3.
   i) If the USB ports are not fully enumerated, you may receive a warning that it did not connect correctly. Cancel out of this startup and try again later. There are four ports on the USB interface of the MAX5861 EV kit – one for the SPI of the MAX5861 device, one for the $I^2C$ interface for the temperature sensor, one for bit toggling of the RST_N and MODE2 pins, and one spare. The ports can also be monitored in the device manager of the PC to determine when all four ports are ready for use (depending on the operating system, they might show up under the Universal Serial Bus Controllers as USB Serial Converter A, USB Serial Converter B, USB Serial Converter C and USB Serial Converter D).
   ii) If the PC has USB 3.0 and it is not fully compliant (i.e. not backward compatible to USB 2.0) then the USB 3.0 mode may need to be disabled in the BIOS for proper communication.
14) Verify proper communication with the board in the **Log Window** of the GUI. The software reads the register 0x000 at software initialization and should read a 0x27 in the ID code, as shown in Figure 3 below.
15) <<If using optional VC707 board for SCQAM input data or OFDM data>>
   i) On the **VC707 Tab** of the GUI, click the checkbox for Xilinx Impact Tools Are Installed
   ii) Click on the Load FPGA Bit File button, browse to the location of the Impact program (if needed – first-time programming only), and then browse to the location of the BIT file (download_lvds_xxxxx.bit) under the FPGA-ConfigurationFiles directory.—
16) Click on the Test tab of the GUI and select a Pseudo Random Bit Sequence (PRBS) file to load by clicking on the button labeled "**1 Carrier, Annex B, 256-QAM, Centered @ 1GHz**". This button will load the filename 001SB256_5861_PRBS_4915p2M_1000M.txt into the MAX5861. The SCQAM load files are in the **ABC_D_E_F**M_**G**M.txt format and the OFDM load files are in the **ABC_D_E_F**M_**G**BW_**H**k_**J**c**K**r_**L**.txt format, where the format is defined as follows.

| SYMBOL | SCQAM DEFINITION | OFDM DEFINITION |
|---|---|---|
| **A** | Number of SCQAM channel, 3 digits | Number of OFDM channel, 3 digits |
| **B** | S for SCQAM | O for OFDM |
| **C** | QAM mapping - A16, A32, A64, A160, A256, B64, B256, C64 or C256 | QAM mapping to include 16, 32, 64, 128, 256, 512, 1024, 2048 and 4096 QPSK for PRBS mode or xxx for input interface data |
| **D** | Target of the configuration - 5861 for MAX5861 or VC707 | Target of the configuration - 5861 for MAX5861 or VC707 |
| **E** | Source of the data - PRBS for internal PRBS generator or input for input interface | Source of the data - PRBS for internal PRBS generator or input for input interface |
| **F** | DAC frequency in MHz | DAC output data center frequency in MHz |
| **G** | DAC output data center frequency in MHz | OFDM Channel BW to include 24, 48, 96 and 192MHz |
| **H** | NA | 4 or 8 for 4k or 8k points IDFT |
| **J** | NA | NCP |
| **K** | NA | NRP |
| **L** | NA | NP or P for no pilot insertion or pilot insertion |

1. Observe the output on the spectrum analyzer and adjust the signal generator and spectrum analyzer, if required.


*Figure 2. MAX5861 EV Kit Software Controller Splash Screen*

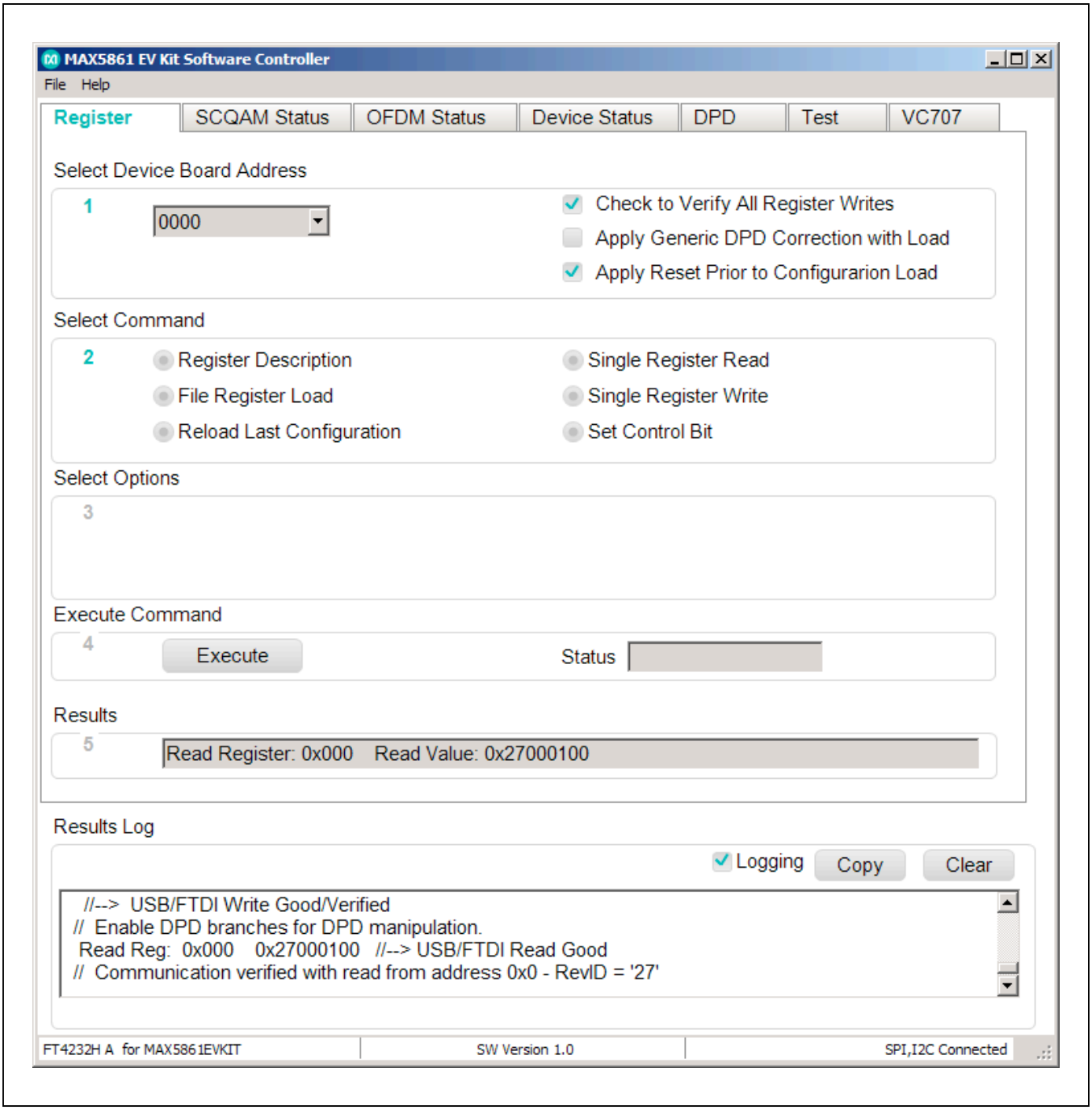



*Figure 3. MAX5861 Evaluation System Controller Software GUI Window*

## Table 1. Jumper Configuration for MAX5861 EV Kit Operation

| JUMPER | POSITION | EVKIT FUNCTION |
|---|---|---|
| JU2 | 1-2 Installed●<br>2-3 Installed | 3.3V LDO drives AVDD33_IN<br>External AVDD3ANALOG Supply drives AVDD33_IN |
| JU3 | 1-2 Installed●<br>2-3 Installed | 1.8V LDO drives AVCLK_IN<br>External ANALOG18 Supply drives AVCLK_IN |
| JU4 | Installed●<br>Not Installed | Connects SE signal to the DUT<br>SE signal not connected to the DUT |
| JU5 | 1-2,4-5●,7-8●,10-11●<br>2-3●,5-6,8-9,11-12 13-14 | CFG1, CFG2, CFG3, CFG4 connected to GND<br>CFG1, CFG2, CFG3, CFG4 connected to VDD18 |
| JU6 | 1-2 Installed<br>2-3 Installed● | Connect REF to DUT<br>Connect DAC REF to GND |
| JU7 | (1-2,4-5,7-8,10-11)●<br>2-3,5-6,8-9,11-12<br>13-14 | SCLK, SDI, SDO, CSA pins connected to USB<br>SCLK, SDI, SDO, CSA pins connected to FPGA<br>Unused connection |
| JU8 | 1-2 Installed●<br>2-3 Installed<br>Not Installed | Selects LVDS Termination<br>Selects SSTL 1.5V<br>Selects SSTL 1.2V |
| JU10 | 1-2 Installed●<br>2-3 Installed | 1.8V LDO drives AVDD18_IN<br>External ANALOG18 Supply drives AVDD18_IN |
| JU11 | 1-2 Installed<br>2-3 Installed● | RST_N_FPGA Drives RST_N<br>Pushbutton RST drives RST_N |
| JU12 | Installed●<br>Not Installed | U11 LDO Drives REFIO<br>External REFIOA drives REFIO |
| JU22 | 1-2 Installed<br>2-3 Installed● | MODE_2_FPGA Drives MODE2<br>Pushbutton MODE_2_LV drives MODE2 |
| JU23 | (1-2,4-5,7-8,10-11) ●<br>2-3,5-6,8-9,11-12<br>13-14 | SA3, SA2, SA1, SA0 pins connected to GND<br>SA3, SA2, SA1, SA0 pins connected to VDD18<br>Unused connection |
| JU24 | Installed<br>Not Installed● | Normal Operation<br>0.9V_VDDSense Monitoring |
| JU25 | Installed<br>Not Installed● | Normal Operation<br>0.9V_GND Sense Monitoring |
| JU26 | 1-2 Installed●<br>2-3 Installed | 1.8V LDO drives 1.8V_OUT<br>External DIGITAL_18 drives 1.8V_OUT |

●*Default position.*

## Detailed Description of Software

The MAX5861 EV kit Software Controller GUI is designed to control the MAX5861 EV kit and the VC707 board as shown in Figure 4. The MAX5861 Software Controller includes USB controls that provide SPI and SMBus communication to the MAX5861 and the MAX6654 interfaces. The software also controls the VC707 through the Silicon Labs COM port on the VC707 board (UART connection on the board panel). The software gives the user the capability of accessing the MAX5861 device's 1492 internal registers in each device, each 32 bits wide.

The MAX5861 EV kit software features six window tabs for operation of the MAX5861 Software Controller and are defined below:

- **Register**
  - Single Access Read And Write Operations
  - File Read/Write Loads And Downloads
  - Reload Of Last Configuration Loaded
  - Register Definition Display
  - Rstn And Mode2 Toggle Control
- **QAM Status**
  - Display Of SCQAM Channel Configuration Summary
  - Display Of The SCQAM Channel Center Frequency
- **OFDM Status**
  - Display Of OFDM Channel Configuration Summary
- **Device Status**
  - Temperature Readings And Control Of The MAX6654 Temperature Sensor IC
  - Display Of The FIFO Status, Parity Error Status And DLL Lock Status
  - Display Of The Saturation Status With The Ability To Clear The Status
- **DPD**
  - Manipulation Of The DPD Register Set Through Easy-To-Use Slide Bars Or Text Boxes
- **Test**
  - Ability To Save Off Current Device Settings
  - Quick And Easy Buttons For Device Configuration Or Generic DPD Settings For Static Linearity And $f_{dac}/2 - 2f_{out}$ Corrections
  - Display Of Sample Spectrum Analyzer Screen Capture Available for Comparison Of Expected Wideband Output

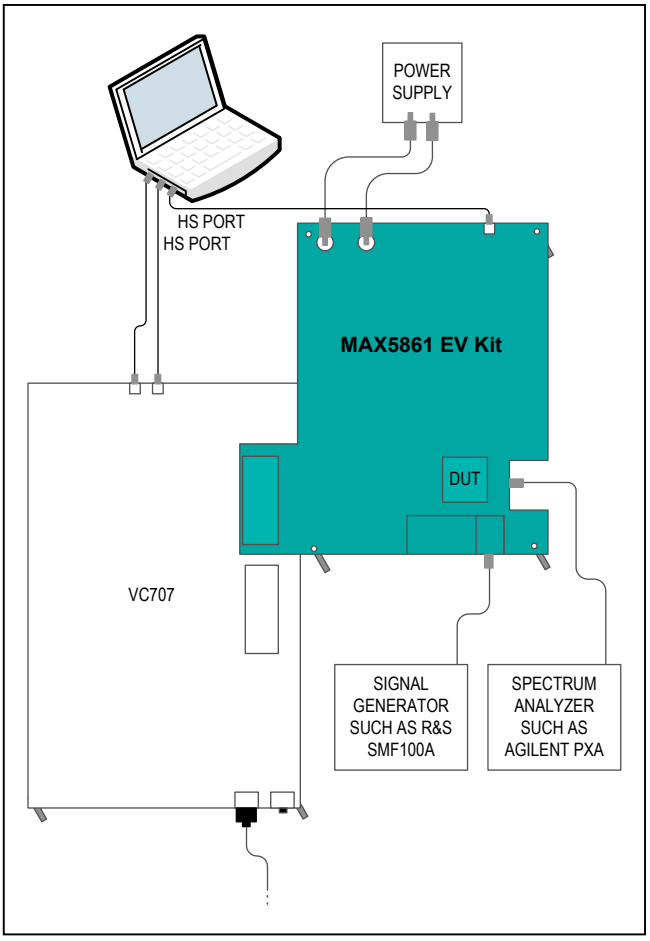



*Figure 4. MAX5861 Evaluation System Block Diagram*

- **VC707**
  - Status Of COM Port Connection and Ability to Connect
  - File Read/Write Loads and Downloads
  - Pattern File DDR3 Memory Loads for OFDM Data Transfer
  - Start Of OFDM Memory Pattern
  - Enable SDCLK Output from FPGA
  - Ability to Load A Bit File Into the FPGA
    Requires Xilinx Lab Tools for Impact Executable to Load Bit File

## MAX5861 EV Kit Software Controller

USB communication to the FTDI microcontroller's SPI is verified upon execution of the MAX5861 EV kit Software Controller. If the USB is not connected or communicating to the interface correctly, a pop-up window appears (Figure 5).

When the MAX5861 EV kit is not found, a debug setup window will appear as shown in Figure 6. If the ***Number of MPSSE/FTDI Ports*** is clicked without any devices, it would show "**No devices found**" in the window.

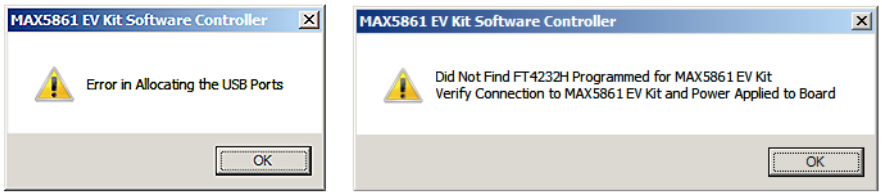



*Figure 5. Error in Allocating USB Ports and Failed to Connect to MAX5861 EV Kit Windows*

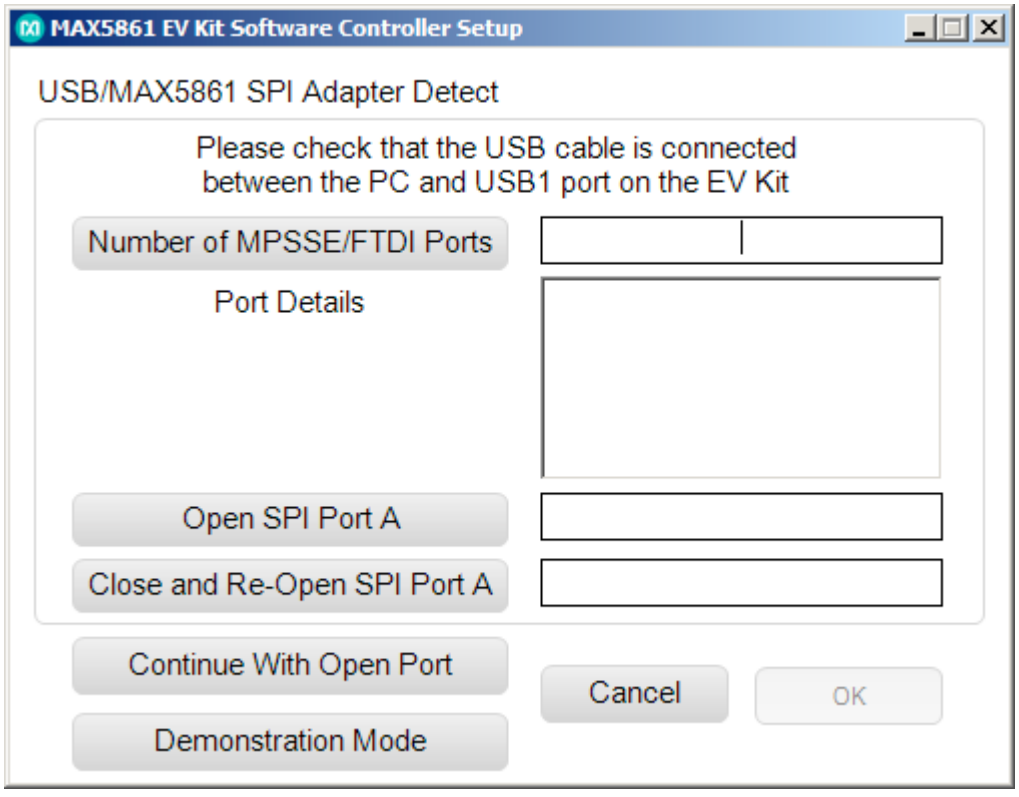



*Figure 6. MAX5861 EV Kit Software Controller Setup Window*

If desired, the ***Demonstration Mode*** button can be clicked to enter the SW GUI. Of course, there is not a board to connect to and therefore no configuration can take place. If demonstration Mode is used, the following window is displayed, with the appropriate "**DEMONSTRATION MODE**" titles and "**Not Connected**" and "**Adapter: None**" messages shown in Figure 7.

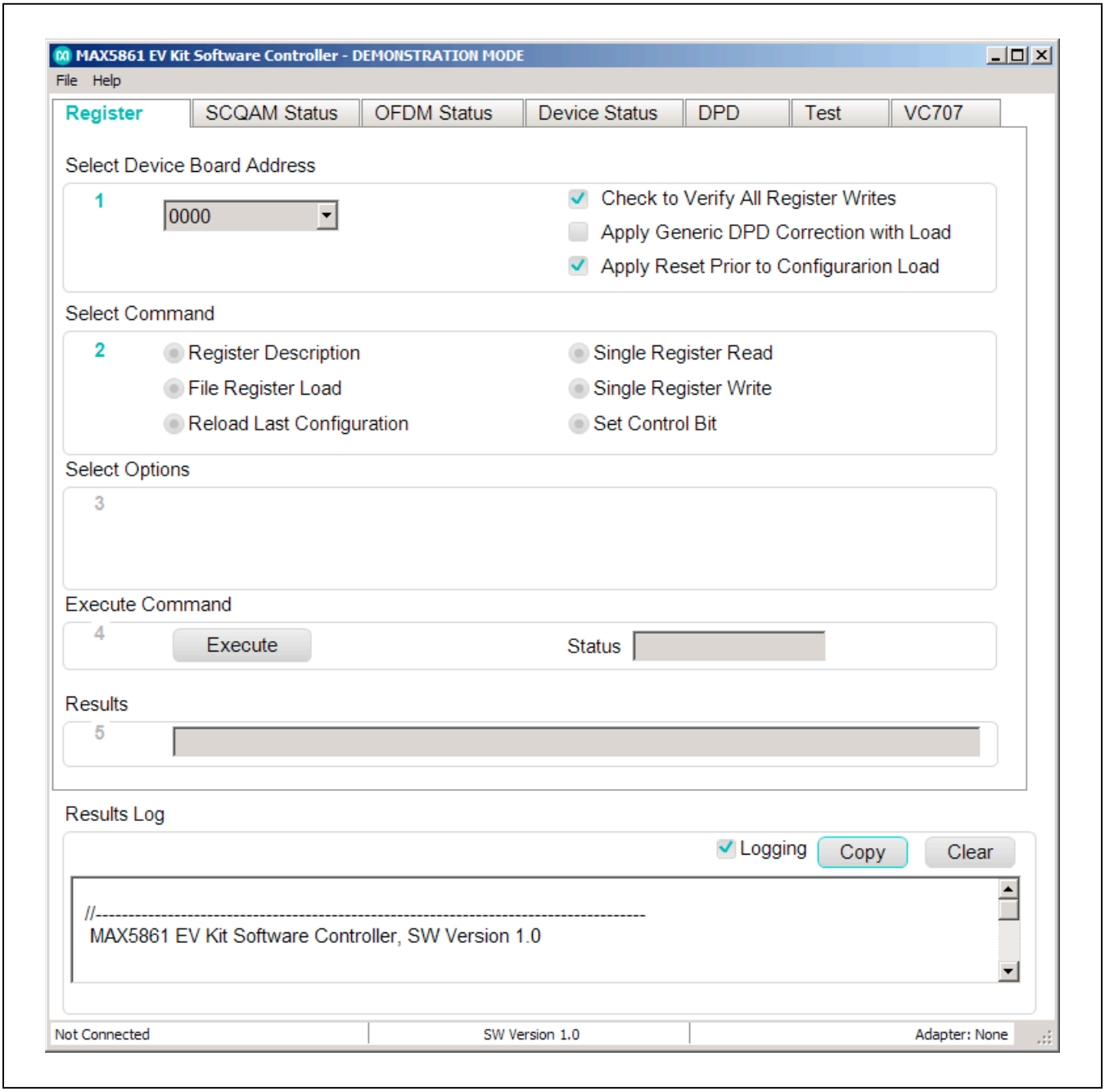



*Figure 7. Demonstration Mode (Debug Mode Display)*

When the MAX5861 EV kit is detected and the ports are connected, the screen shown in Figure 8 is displayed. The initialization of the SW will read the RevID from the ID register in the MAX5861 device at offset 0x000 to see if the communication is working between the SW and the MAX5861. The RevID should read '27'. If this is the case, then it assures that the communication is working between the SW and device. If the board is set up with a board address set to something other than '0000', then this communication will not work. However, this does not mean that it would not work once the board address was set correctly using the **Select Device Board Address** pulldown. The SW will also attempt to enable the DPD branches to allow for correction manipulation through the DPD page.

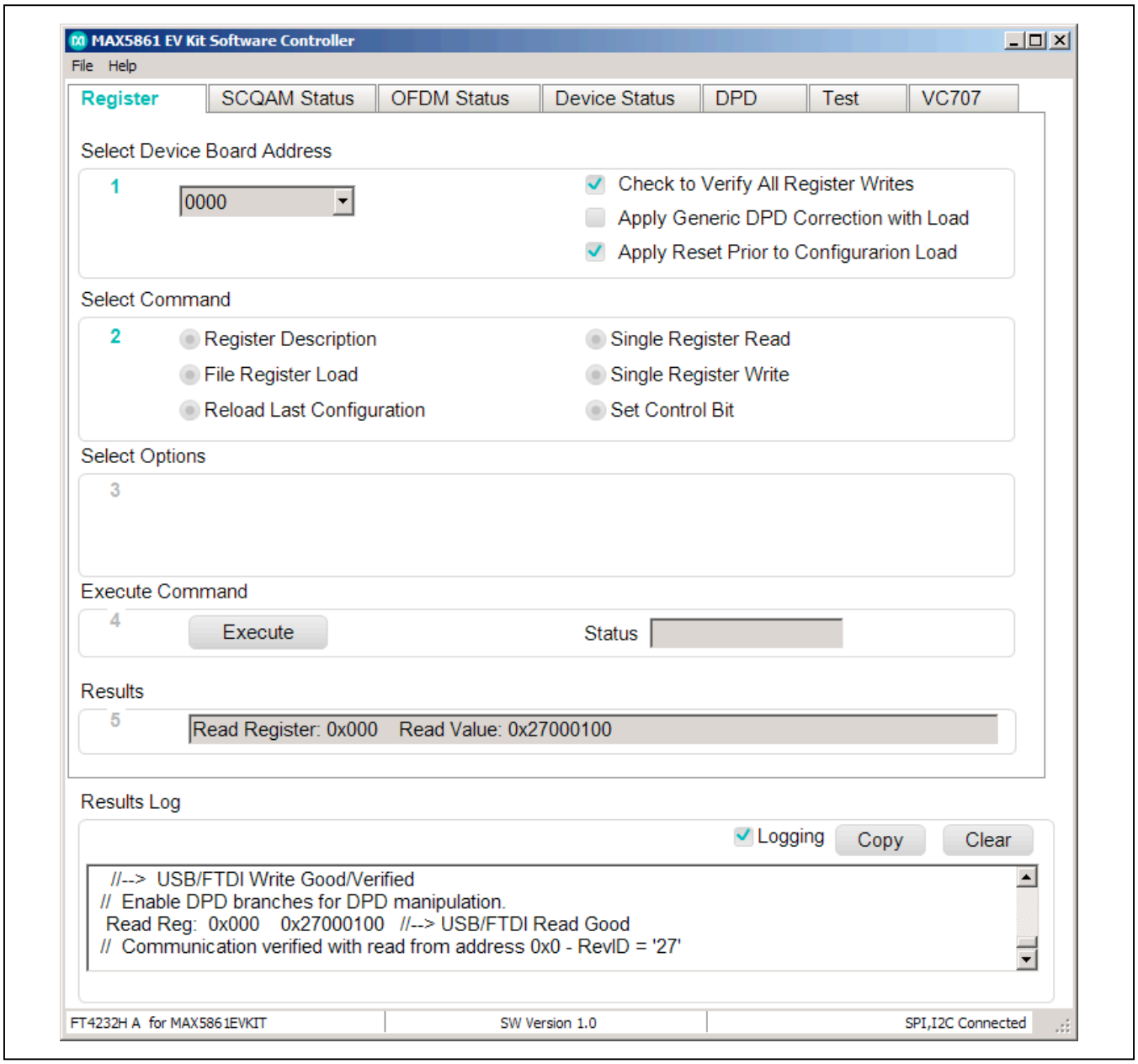



*Figure 8. Initial Display with Proper Connections*

When connecting to the MAX5861 EV kit correctly, the text in the **Results Log** will show text similar to the following text:

If Digilent devices show up in the **Results Log** text, then the VC707 board has the JTAG cable connected. The FT4232H devices are the MAX5861 EV kit board. There are four ports on the FTDI USB device where the MAX5861 EV kit uses port A for the SPI communication to the MAX5861, port B for the $I^2C$ communication to the temperature sensor, port C for GPIO expansion bus, and port D for bit-bang functions such as issuing a RST_N or MODE2 pulse to the MAX5861 device.

The status bar at the bottom of the MAX5861 EV Kit Software Controller shows the status of the connection when the GUI was opened. It also displays the version of the software.

Notes about the MAX5861 EV Kit Software Controller GUI:

Boxes that can have user entry are white in color, with the exception of the **Results Log**. Text boxes that are for displaying results of an action are grey. The **Results Log** will show grey if it has been disabled. Logging can be disabled, but this does not result in much of a savings on the processing timing and is therefore not recommended so that results feedback can be provided to the user. The entire content of the **Results Log** can be copied by clicking on the Copy button and then pasted into a text editor for viewing if desired. The **Results Log** can be cleared by clicking on the Clear button.

```
//------------------------------------------------------------------------
 MAX5861 EV Kit Software Controller, SW Version 1.0

 // Connected to FPGA//
 //Initialization:  FTDI Device List Returned
 //  Device 0 Digilent Adept USB Device A
 //  Device 1 Digilent Adept USB Device B
 //  Device 2 FT4232H A
 //  Device 3 FT4232H B
 //  Device 4 FT4232H C
 //  Device 5 FT4232H D
 //  Device/port for SPI is 2
 //
 //  Clear Lock Status after Reset
 //Register:  Read Modify Write Reg:  0x0D8  data  0x00000000  mask  0x00000020
 Read Reg:  0x0D8    0x83FF10B4   //--> USB/FTDI Read Good
 Write Reg:  0x0D8    0x83FF1094
  //--> USB/FTDI Write
 Write Reg:  0x038    0x000001F0
  //--> USB/FTDI Write Good/Verified
 //  Enable DPD branches for DPD manipulation.
 Read Reg:  0x000    0x27000100   //--> USB/FTDI Read Good
 //  Communication verified with read from address 0x0 - RevID = '27'
```

*Figure 9. Sample Text Output When Connected*

## MAX5861 Register Tab

Figure 10 displays the **Register** tab page. The board address can be selected from the pulldown (range of 0000 to 1111) in panel 1. Panel 2 contains the **Select Command** radio buttons. The radio buttons include the **Register Description, File Register Load, Reload Last Configuration, Single Register Read, Single Register Write and Set Control Bit**.

To see the register description, click on the **Register Description** radio button, enter the address in the **Address** text box located in panel 3 and click the **Execute** button in panel 4. The description of the register defined will display in the **Results Log** window.

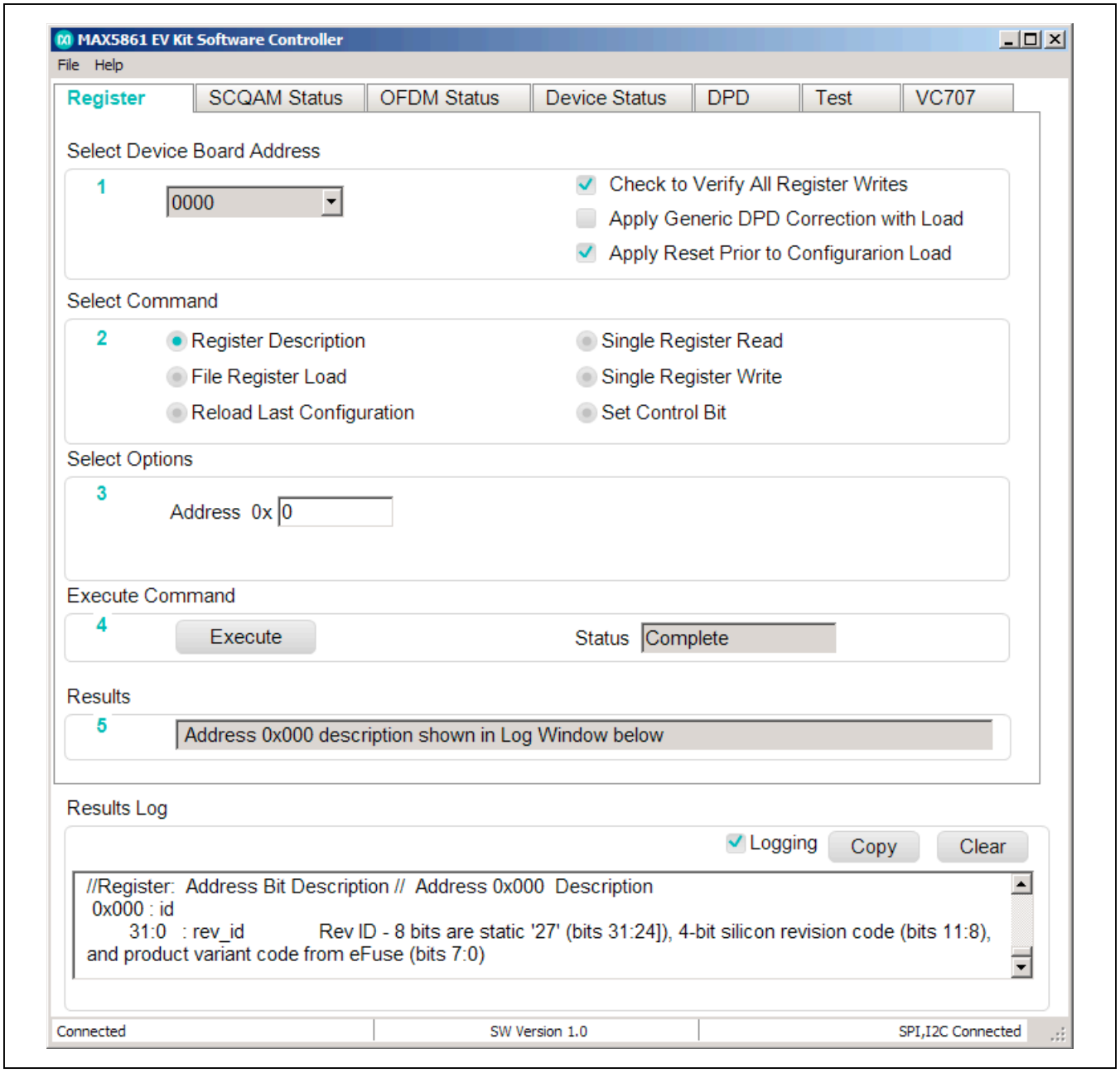



*Figure 10. Register Tab Register Description Results*

To load a MAX5861 configuration file, click on the **File Register Load** radio button and click on the **Execute** button in panel 4. This will bring up the file browse window for the configuration file selection.

There are a few selections in panel 1 that affect a file load. These are **Check to Verify All Register Writes, Apply Generic DPD Correction with Load and Apply Reset Prior to Configuration Load**. The **Check to Verify All Register Writes** checkbox will read the register after the write to verify that the contents were received. The registers will have a mask so that it only verifies the appropriate bits. The **Apply Generic DPD Correction with Load** checkbox will apply values to the $f_{DAC}/2 - 2f_{OUT}$ correction (using DPD gain 9 and gain 10 offsets – 0x03E written with 0x004A00FA) and static linearity correction (using DPD gain 11 and gain 12 – 0x03F written with 0x008007F0). The **Apply Reset Prior to Configuration Load** checkbox will initiate a toggle of the RST_N pin before loading a configuration load file.

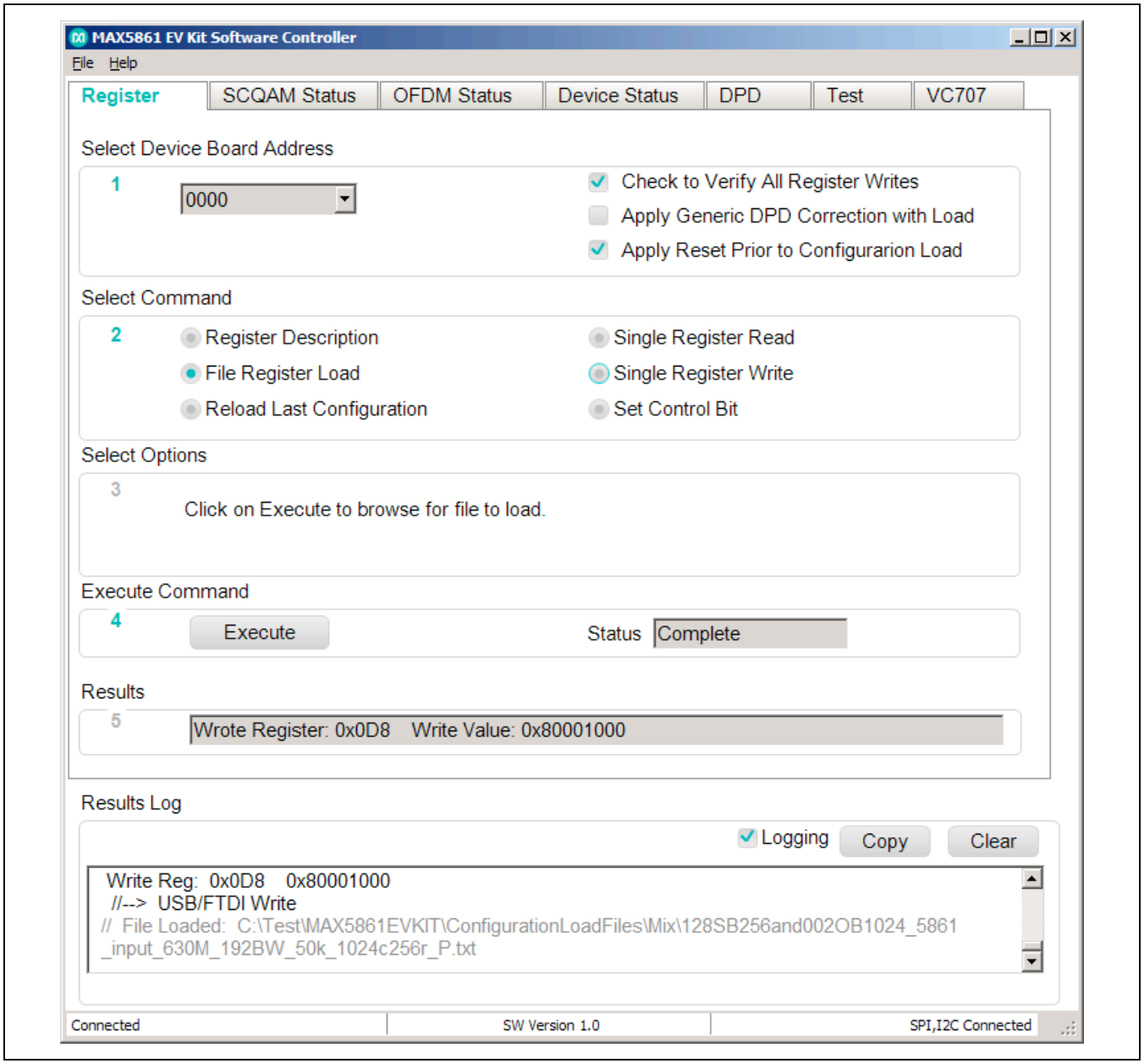



*Figure 11. Register Tab File Register Load*

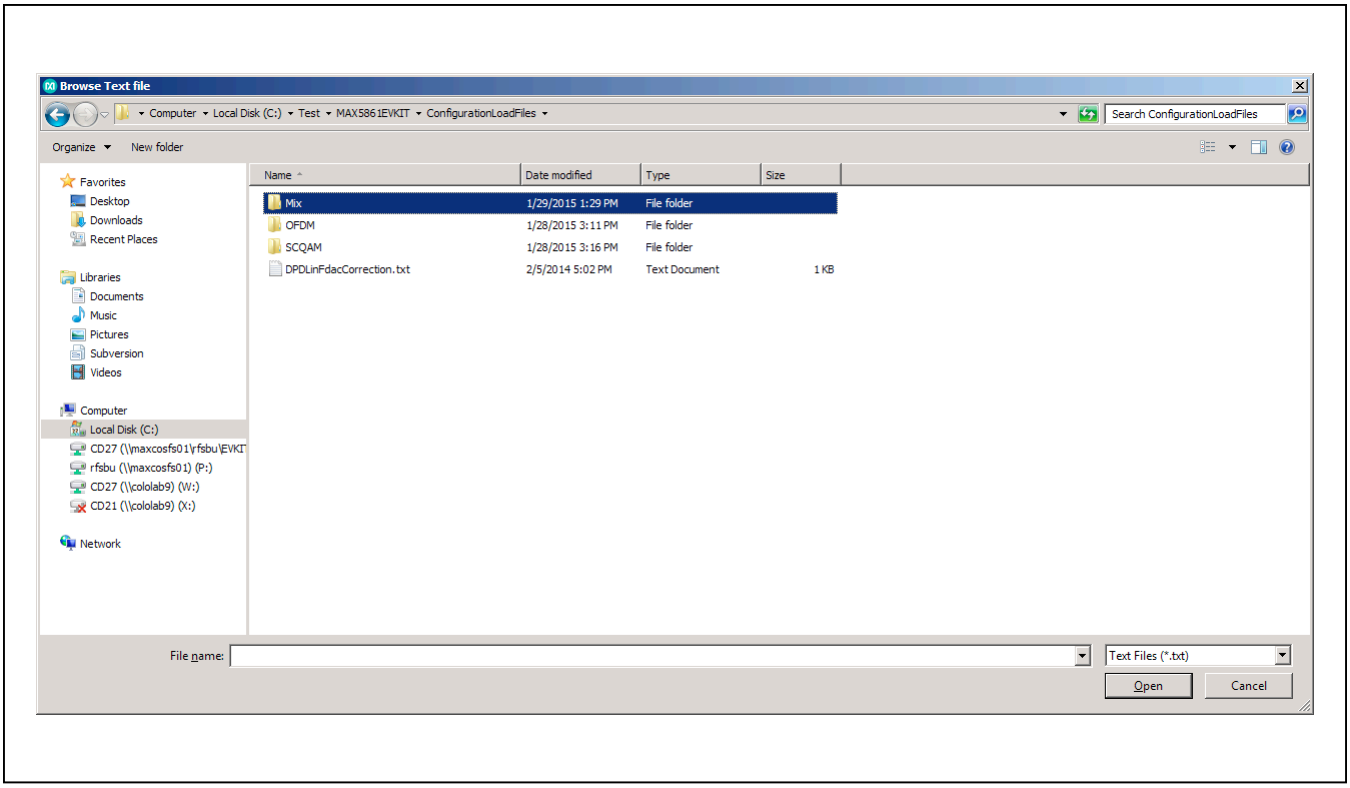



*Figure 12. Browse Window for Register File Load*

To read an individual register within the MAX5861, click on the **Single Register Read** radio button, enter the address offset into the **Address** box in panel 3 and click on the **Execute** button in panel 4. The result will be displayed in both the Results window in panel 5 as well as the **Results Log**.

To write to an individual register within the MAX5861, click on the **Single Register Write** radio button, enter the address offset into the **Address** box and the data to be written into the **Data** box in panel 3 and click on the **Execute** in panel 4. The write will be performed, verified and displayed in both the Results window in panel 5 as well as the **Results Log**.

To set a control bit to the MAX5861, click on the **Set Control Bit** radio button, select the bit(s) to toggle in panel 3 and click on the **Execute** in panel 4. The RST_N and/or MODE2 bits connected to the device will be toggled.

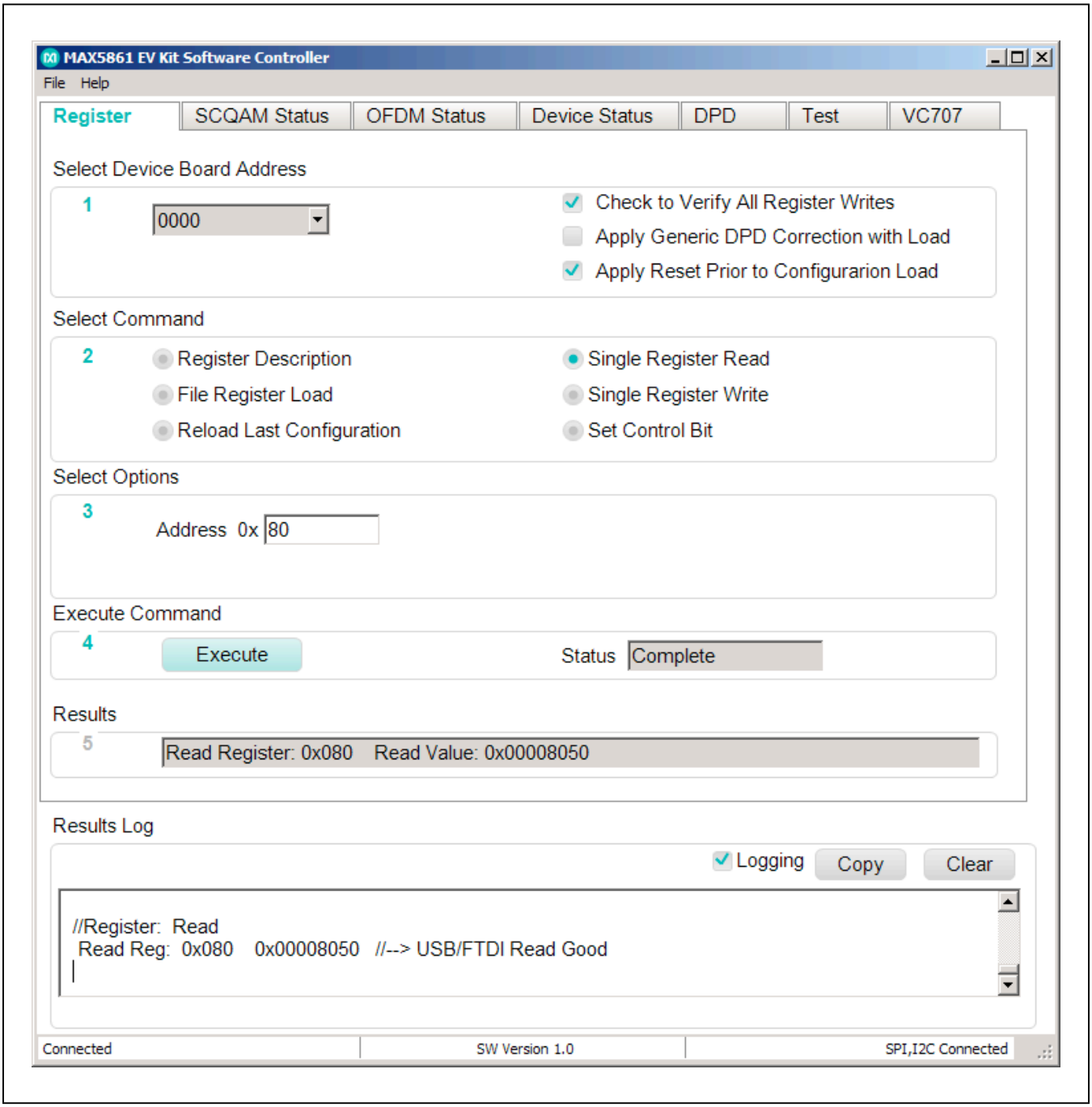



*Figure 13. Register Tab Single Register Read*

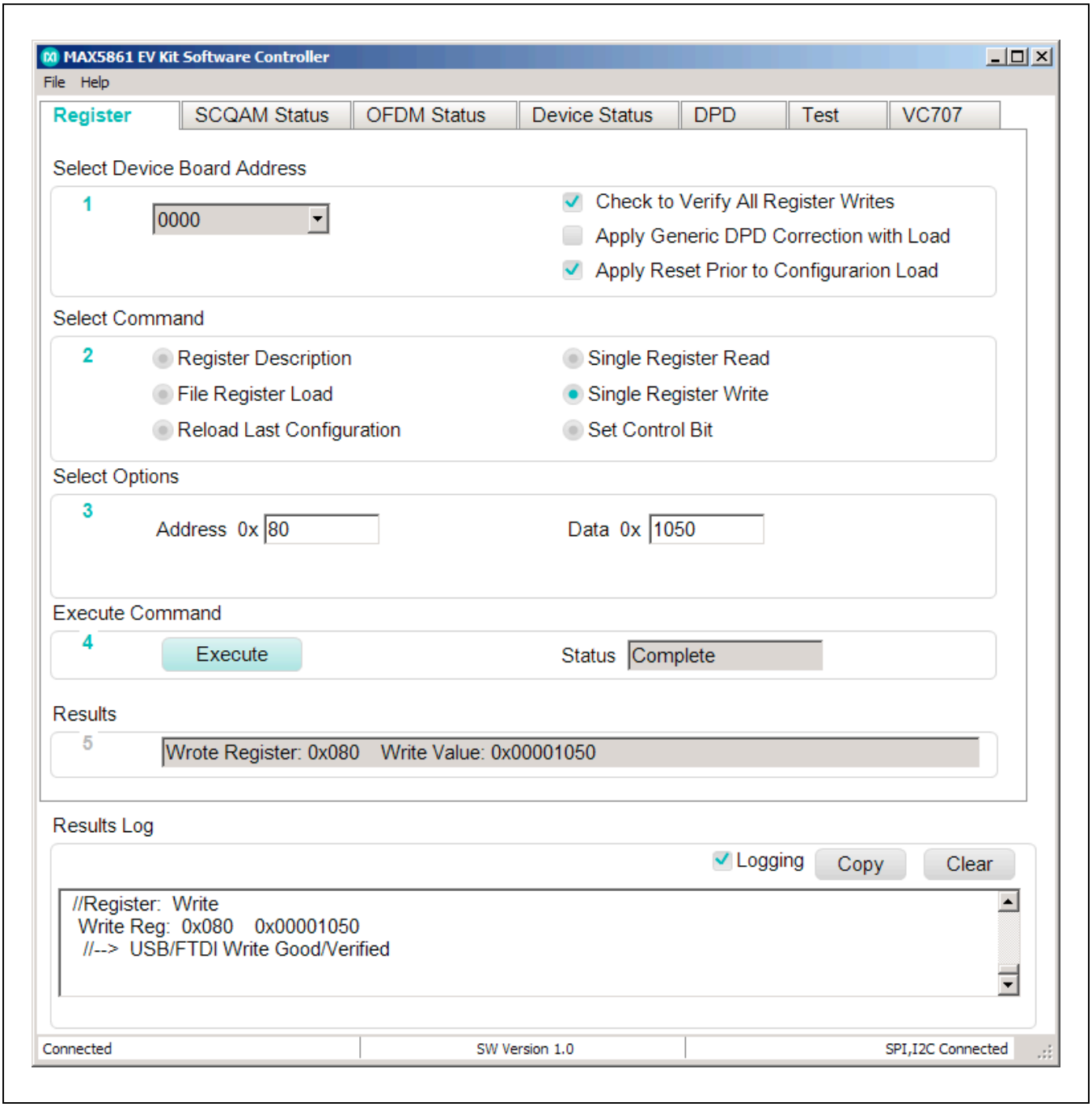



*Figure 14. Register Tab Single Register Write*

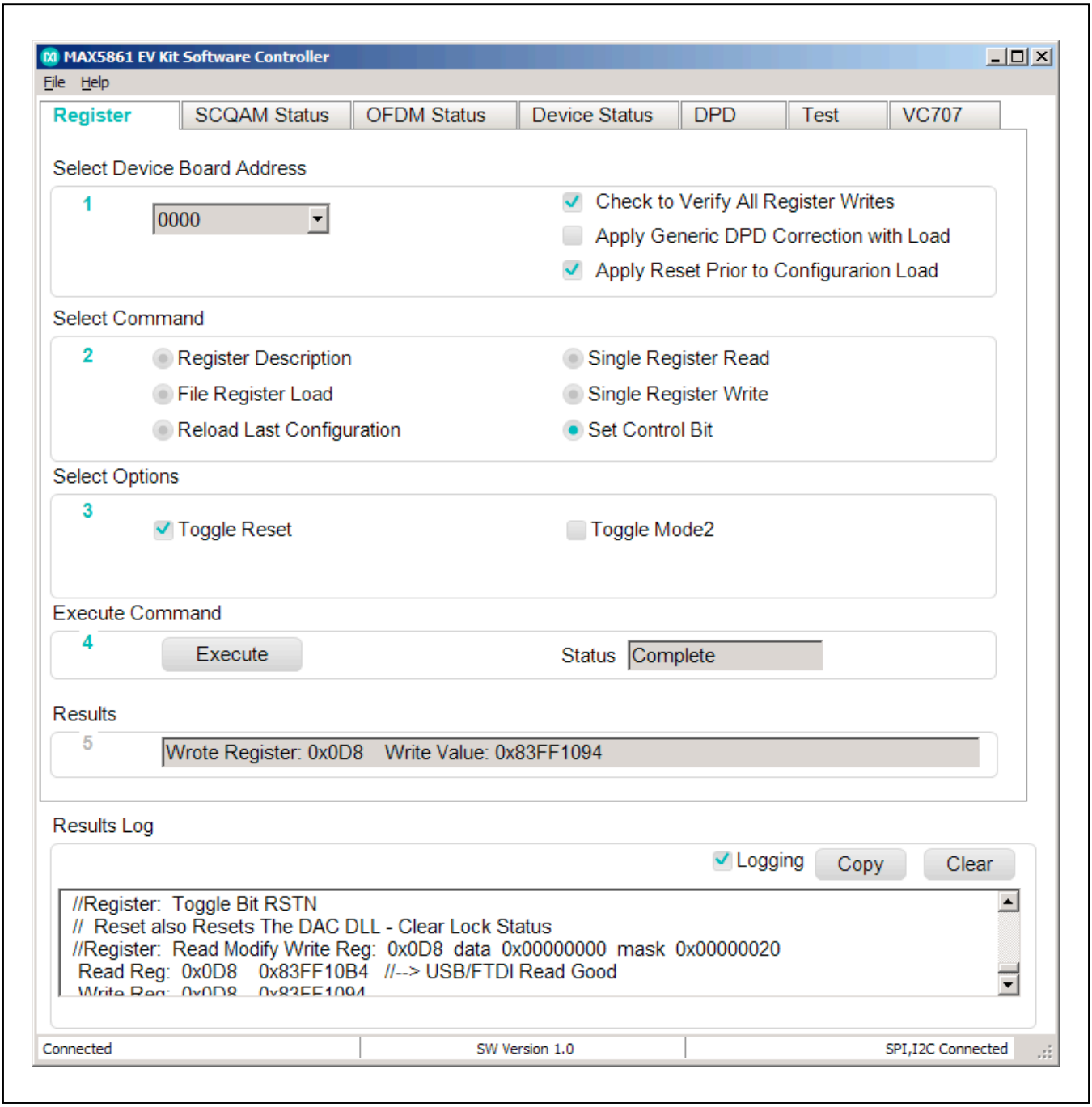



*Figure 15. Register Tab Set Control Bit*

### MAX5861 SCQAM Status Tab

There are many status qualifiers of the MAX5861. The SCQAM Status page is designed to provide status about a specific SCQAM channel's configuration. Enter the channel into the text box and click on the Display SCQAM Channel Status for Channel # button and the information about that channel will be displayed. To determine the programmed center frequency of a channel, enter the channel number, enter the $f_{DAC}$ frequency for the system and click the Calculate $f_{OUT}$ button. The result will be displayed.

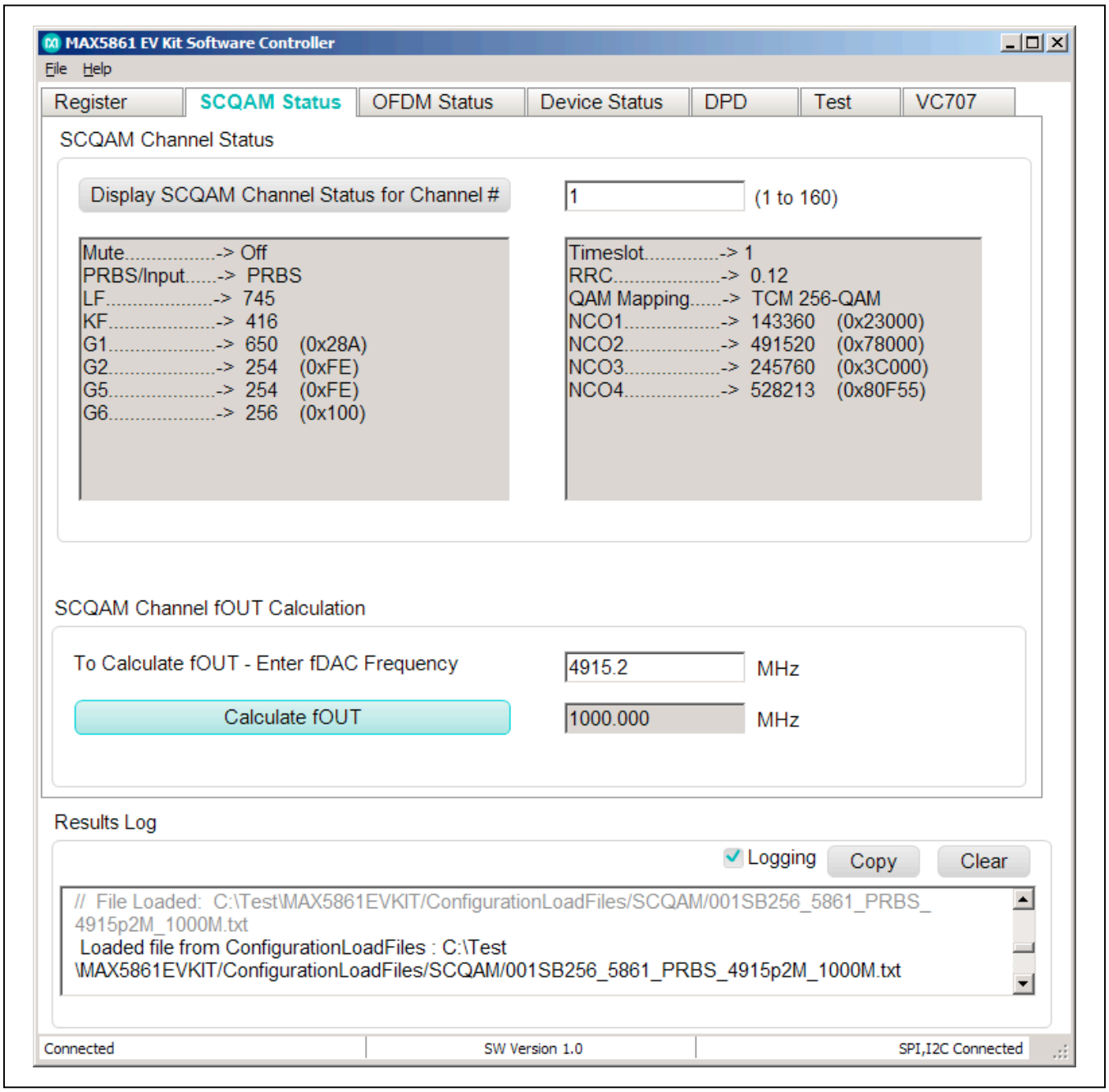



*Figure 16. QAM Status Tab*

### MAX5861 OFDM Status Tab

The **OFDM Status** tab is designed to provide status about a specific OFDM channel's configuration. Enter the channel into the text box and click on the **Display OFDM Channel Status for Channel #** button and the information about that channel will be displayed.

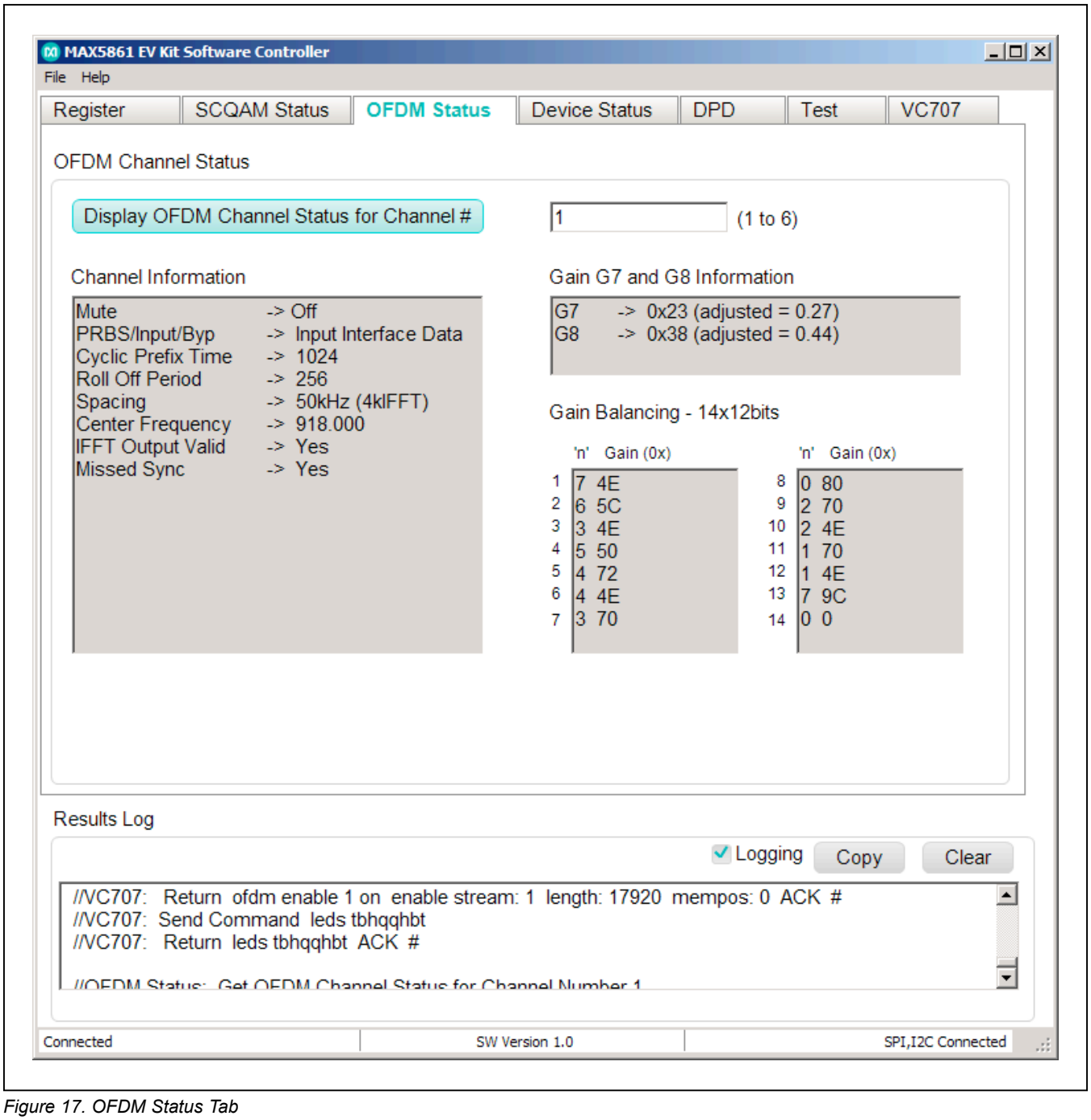



*Figure 17. OFDM Status Tab*

## MAX5861 Device Status Tab

The **Device Status** tab is designed to provide status about the overall health of the MAX5861. There are three status panels to include the Temperature, FIFO, LOCK, Parity Status and Saturation Status panels. The temperature of the MAX5861 can be read from the remote temperature sensor by clicking the **Read Temperature** button. The threshold that determines alerts (shown with the LED on the board) can be modified by setting a new temperature into the text box next to the Set Threshold button and then clicking on the button to set it. If the LED is on and the current temperature is below the current threshold, then the **Clear Temperature Alert** button can be clicked to clear the LED.

The FIFO, DLL lock status and parity error status can be seen by clicking the **Get FIFO Status**, **Get DLL Lock Status**, and **Get Parity Error Status** buttons under the **FIFO**, **LOCK**, **Parity Status** panels. Keep in mind that FIFO and Parity errors can be seen on startup depending on programming order. Clear these errors by clicking on the **Get FIFO Status** and **Get Parity Error Status** buttons right after initial configuration. After the initialization errors have been cleared, click on the buttons to determine the system's true health during normal operation. The saturation status of the stages throughout the device can be verified by clicking on the **Get Saturation Status** button. This reads many registers and will take a minute. The status will be displayed but can be cleared by clicking the **Clear Saturation Status** button.

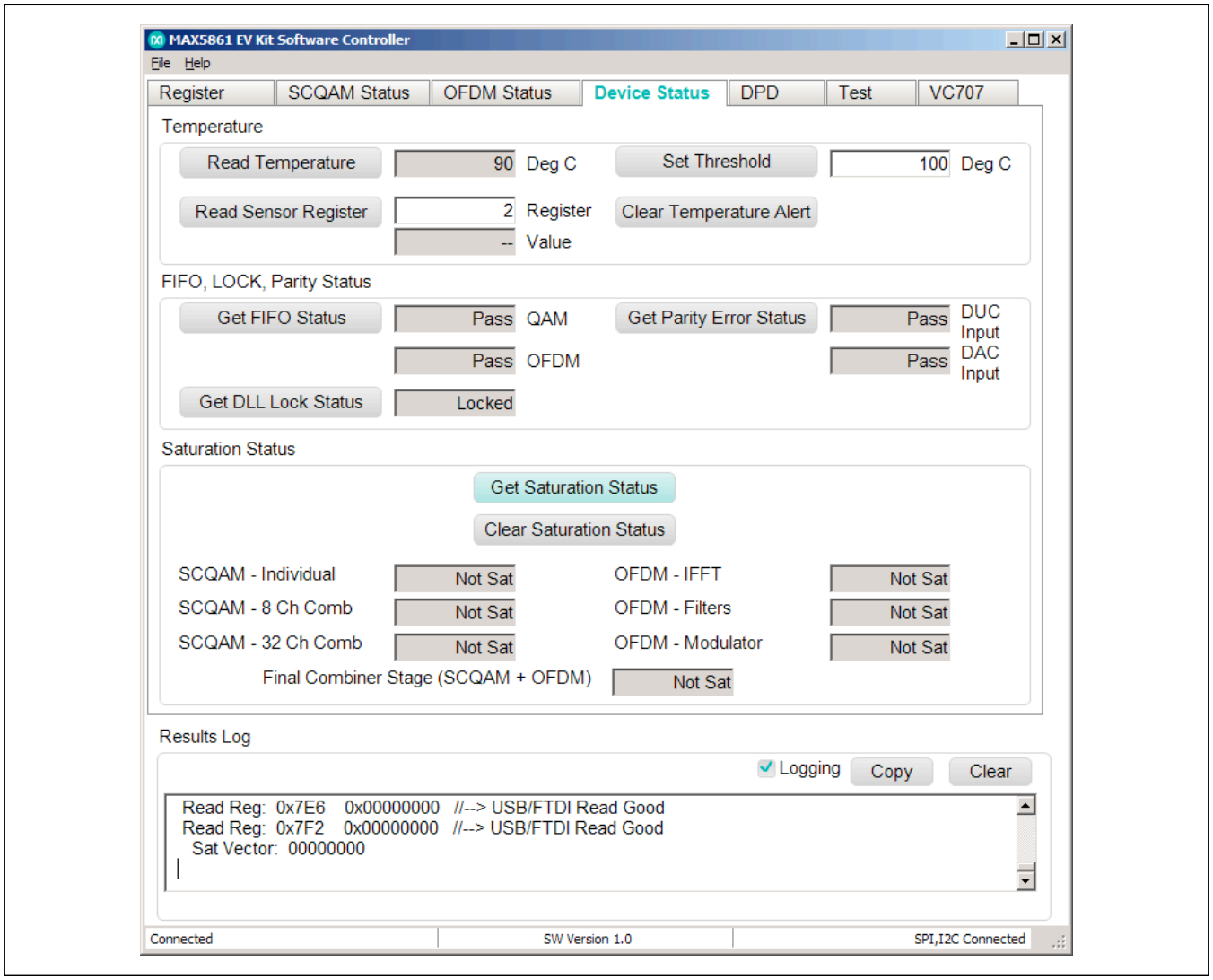



*Figure 18. Device Status Tab*

## MAX5861 DPD Tab

The **DPD** tab is designed to provide an interactive way for the user to set the DPD register values and see the results on the analyzer in real-time. When first accessing the DPD page, click the **Set to Register Gain Values** to synchronize the page up with the current register settings. After this point, either the text entry method or the slide bar method can be used. New values can be entered into the text boxes followed by the **Apply Text Box Gain Values** button. This will load the values into the registers. Another option with the text boxes is to use the left and right arrow keys. Click on a box to place the cursor in the box and then use the left or right arrows to change the values by plus or minus 1. The results of these settings can be easily seen on the spectrum analyzer where the sweep time is set to less than a second. In addition to the

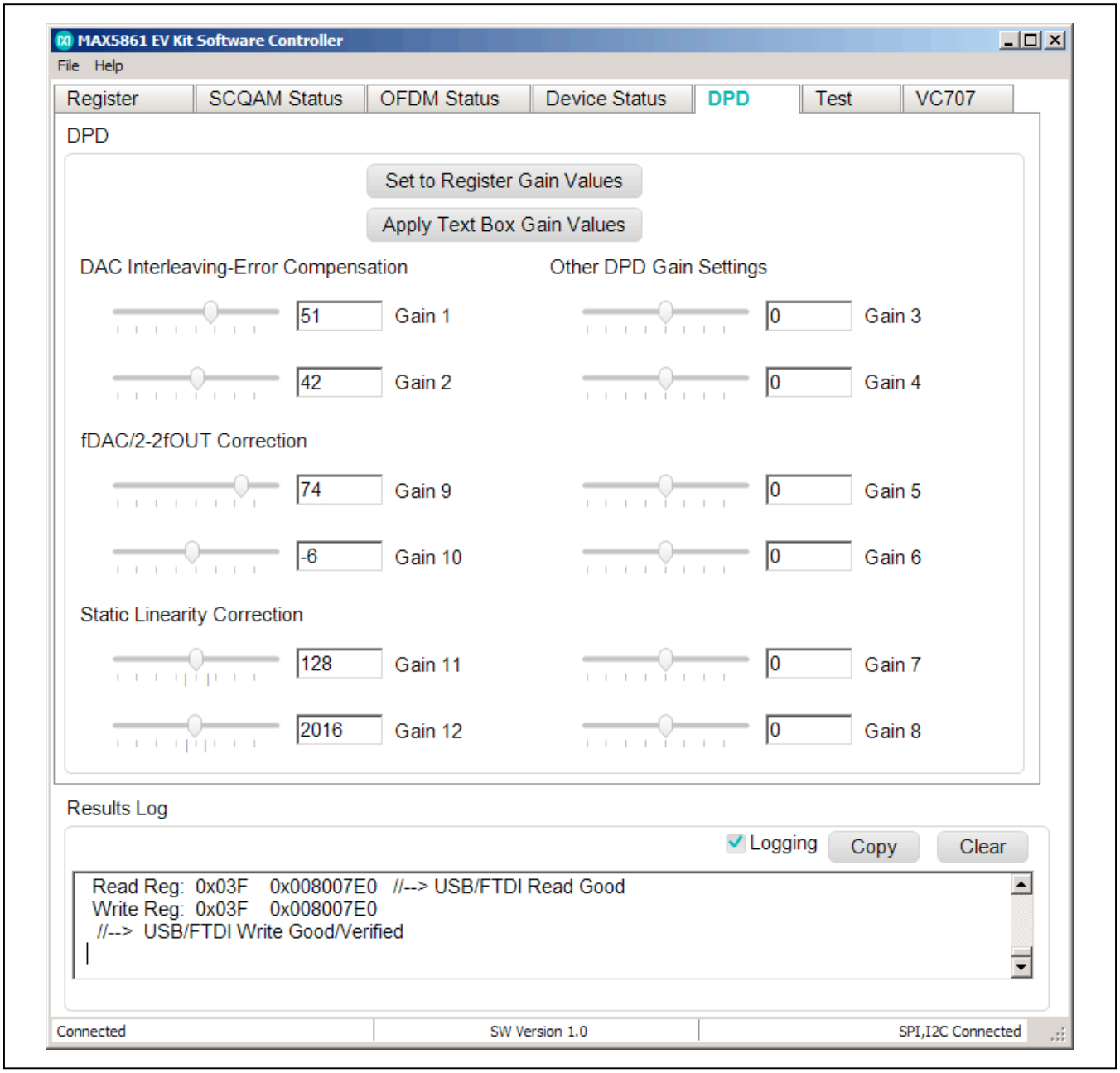



*Figure 19. DPD Tab*

text boxes, the slide bars can be moved. The software will write a new value to the register if the value has been moved from its previous location. This allows real-time monitoring of the output signal while changing the gain values with the slide bars. All of the writes to vary the DPD settings will be logged in the **Results Log**.

## MAX5861 Test Tab

The **Test** page is designed to allow capturing of the device's current configuration as well as easy initial configuration of the device. The Save Settings panel will capture the current register settings of the device. They can be captured in their entirety in numerical order or capturing only the active SCQAM channels and their settings. The Load Sample Configurations Using Internal PRBS Data panel allows a user to click a single button and load a pre-set PRBS configuration into the device. Under this panel, there is also an ***Apply Generic DPD Values – Static Lin & fDAC/2-2fOUT*** button. This writes generic DPD gain values to the Static Linearity and $f_{DAC}/2 - 2f_{OUT}$ branches of the DPD processing. See the device datasheet for further details. When loading the configurations under the SCQAM Configurations, SCQAM and OFDM Configurations and OFDM Configurations sections, the

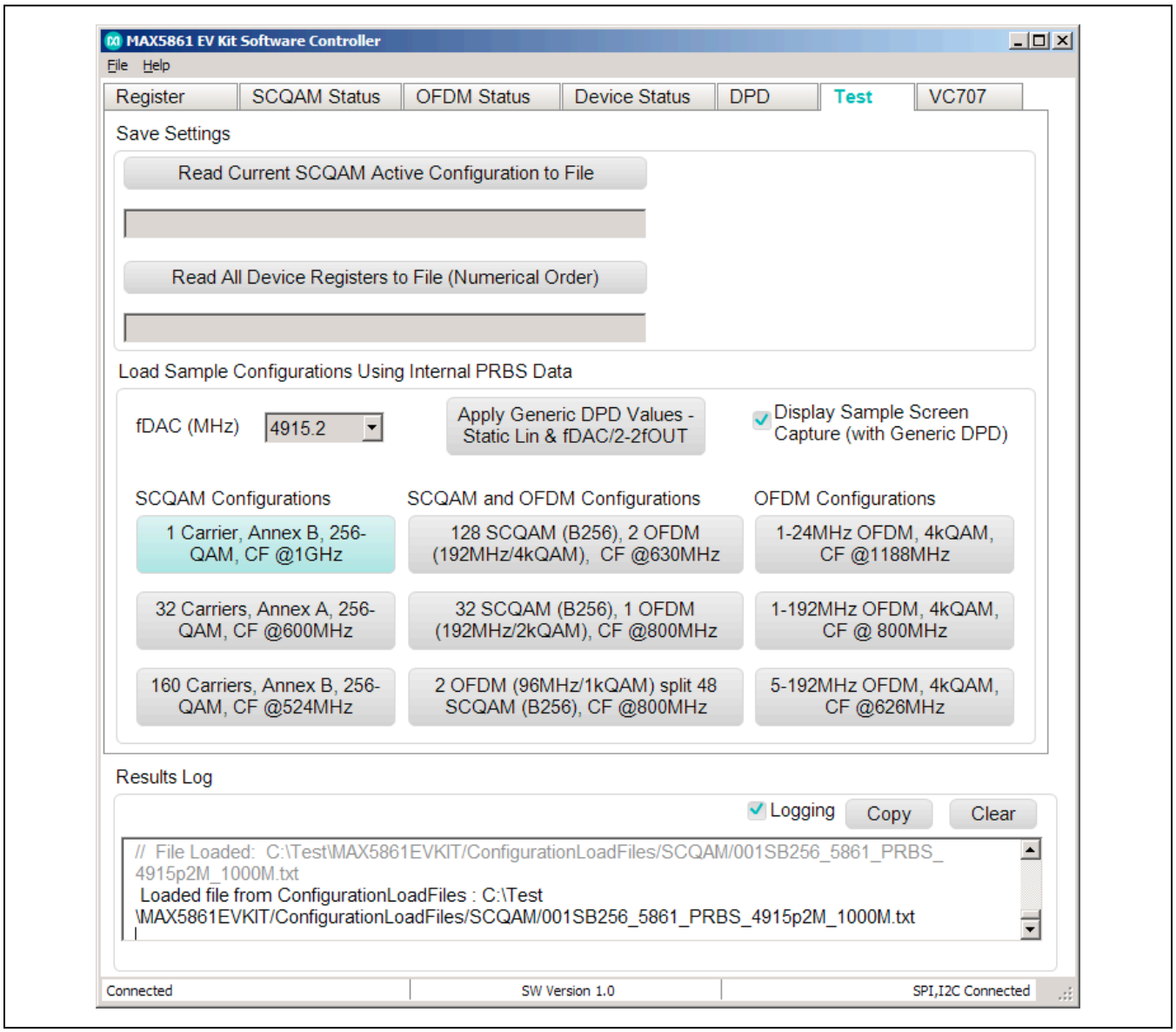



*Figure 20. Test Tab*

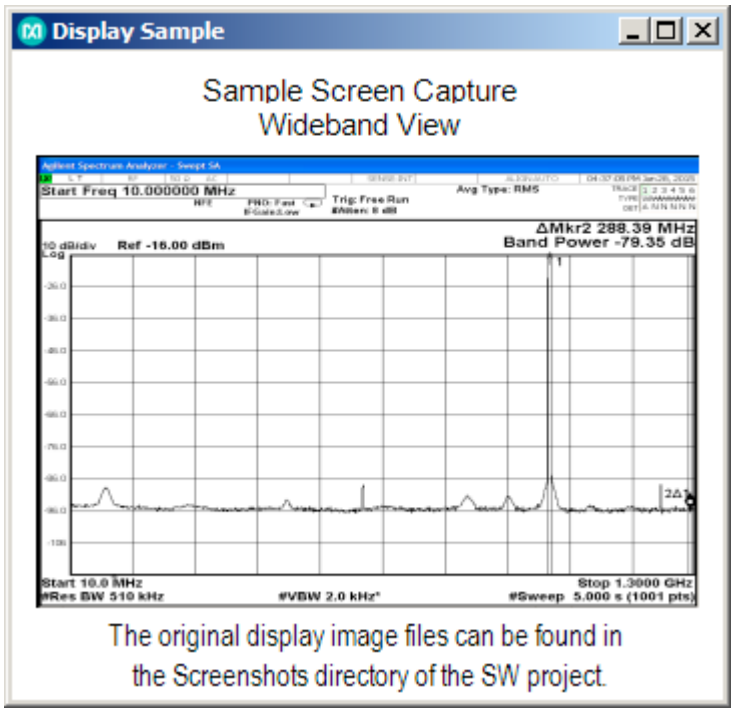



*Figure 21. Test Tab Option for Displaying the Sample Wideband Spectrum Analyzer Capture*

expected sample screen captures can be displayed by clicking on the checkmark and making it an active green checkmark. This will open up another window to show the screenshots. These files can also be viewed, if larger images are needed, by accessing the Screenshots directory of the installed software.

## MAX5861 VC707 Tab

The **VC707** tab is designed to provide control of the VC707 FPGA through the MAX5861 EV kit GUI. The page allows the user to send single commands, load configuration files, load memory files, enable SDCLK, start memory data flow and stop memory data flow. The user can send individual commands to the VC707 with the use of the **Single Command to Send** button. To load a configuration into the VC707 for SCQAM setup, click the **File – Load Configuration** button. These configurations set up the time slots, PRBS information for the channels, clocks, and align the RDY interface appropriately. As the signals come up, it will go through a training sequence on the ready signals so only the second through the eighth channels (in each 8-channel combiner) will show until the training is complete. These can take a couple of minutes to load. To run the OFDM from the FPGA, a pattern memory file must be loaded into the VC707 board. This file can be loaded via the **File – Load Memory Pattern** button. It will bring up a file browse window to search for the desired memory file. The memory files can take a little while to load into the VC707 memory (i.e. a 4k IFFT pattern with 128 symbols for demodulation can take approximately 10 minutes and an 8k IFFT pattern with 128 symbols can take approximately 20 minutes – shorter patterns of 14 symbols will only take a couple of minutes). There is a CRC verification on the data written into the memory to confirm expected data values. The SDCLK to the MAX5861 must also be transmitting. This

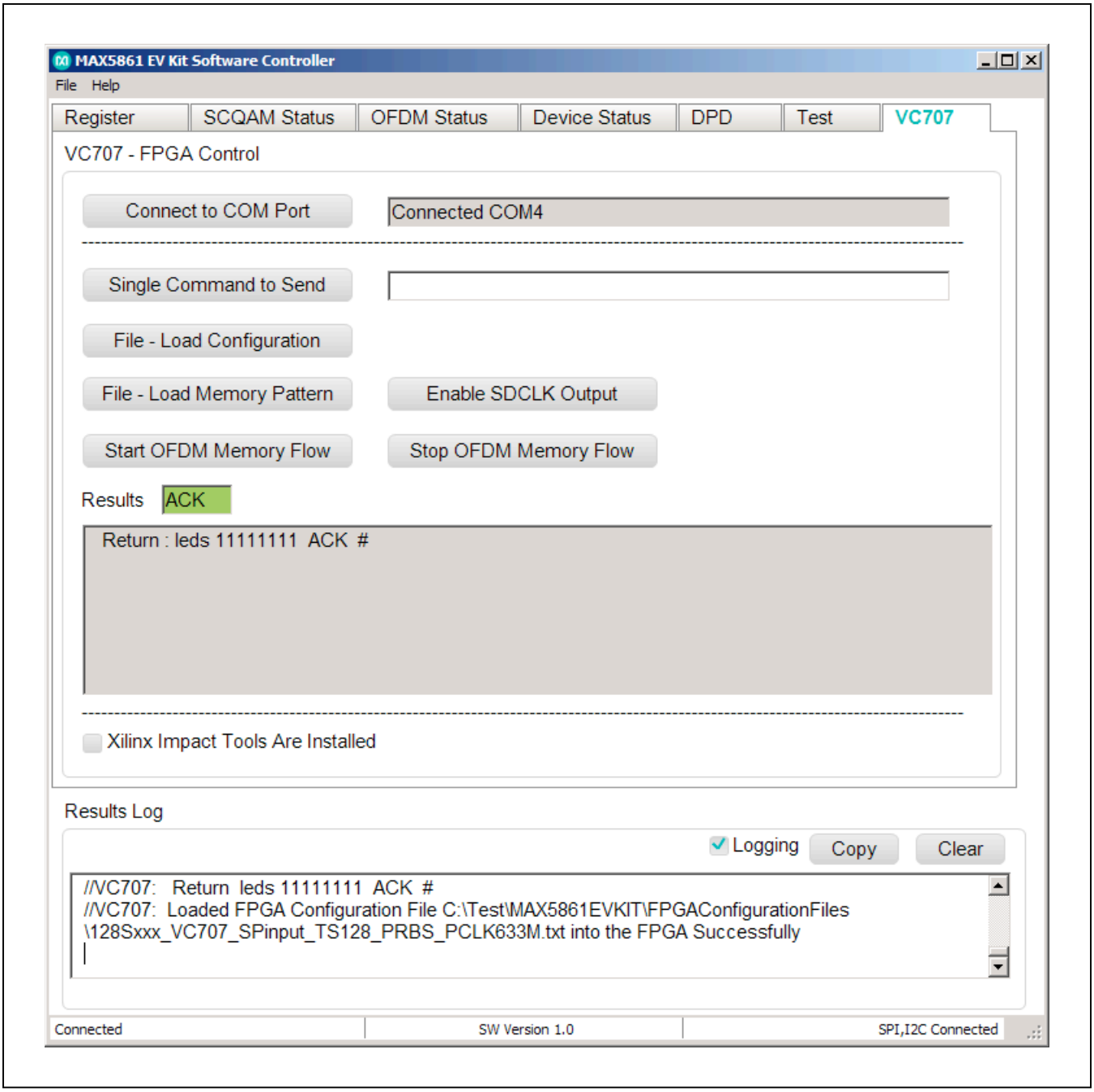



*Figure 22. VC707 Tab*

can be accomplished with the **Enable SDCLK Output** button. The final step for the OFDM pattern to drive the MAX5861 is to send the OFDM memory pattern from the VC707 to the DUC. To send the data, click the **Start OFDM Memory Flow** button and to stop the flow of data, click the **Stop OFDM Memory Flow** button.

## MAX5861 Configuration Load Order

The optimal order of operation for the system with the MAX5861 receiving data from the VC707 is the following:

1) Configure the MAX5861 for the SCQAM and/or OFDM channel
2) Program the FPGA to send data to the MAX5861 channel
3) With data flowing from the FPGA to the MAX5861, clear the status registers of the MAX5861 of the startup condition triggers (FIFO, parity, saturation, etc)
4) Monitor the MAX5861 for health as needed

For the initial configuration of the device, the optimal order of operation for the registers within the MAX5861 is the following:

1) Set the Gain5 and Gain6 to zero (GAIN56 register)
2) Power up the blocks (GBL_CFG2)
3) Program channel(s) to include all NCO load pulses
4) Set Gain5 and Gain6 to desired values

To program another channel when the device is already configured, program in the order defined:

1) Make sure the channel is muted (CHAN_x_x for SCQAM and OFDM_CFG_x for OFDM)
2) Power up the additional block (GBL_CFG2)
3) Set the Gain1/Gain2 (G1G2_x) for SCQAM or Gain7/Gain8 (GAIN_x) for OFDM to zero
4) Unmute the additional channel(s)
5) Program the channel(s) to include all NCO load pulses (if using OFDM, this includes the GAIN_x register for the NCO3 load, while keeping Gain7/Gain 8 at zero)
6) Set Gain1/Gain2 or Gain7/Gain8 to the desired gain values

This procedure should ensure that the channels come up cleanly in the spectrum.

## Detailed Description of Hardware

The MAX5861 EV kit contains a MAX5861 high-density downstream cable SCQAM and OFDM modulator that integrates a DUC and RF-DAC. The MAX5861 DUC performs SCQAM and OFDM mapping, pulse shaping and digital RF up-conversion with full agility and then drives a 14-bit 4.6Gsps RF-DAC. The device digitally synthesizes RF signals with up to 160 DOCSIS-compliant 6MHz QAM channels (or up to 120 8MHz QAM channels) on a single RF port at frequencies from 47MHz to 1218MHz. The MAX5861 device provides up to 6 channels of OFDM IFFT processing. Each of the OFDM channels provide up to 192MHz of bandwidth. The MAX5861 can support up to 6 blocks powered on at one time, where a block is defined as an OFDM channel or a 32-channel SCQAM block. The MAX5861 EV kit provides a complete system solution for high-density QAM modulation with very low power dissipation. The EV kit operates from a single 5.0V/3A input power supply.

The MAX5861 EV kit provides a Samtec FMC connector J1 to drive the IC input ports. The MAX5861 output is available for viewing at the OUT SMA connector (see Figure 4).

The EV kit incorporates a MAX6654 device for monitoring the die temperature. The LED D1 turns on when the MAX5861 die temperature rises above the programmed high temperature threshold within the MAX6654. See the *MAX5861 Device Status Tab* section for configuring the high temperature threshold. Also refer to the MAX6654 IC data sheet for additional register information.

The EV kit provides on-board SPI and SMBus interfaces and is connected to the computer through the USB connector, USB23. The EV kit Windows XP- , Vista- and Windows 7/8- compatible software provides a GUI for control of the MAX5861, MAX6654, and VC707 programmable features. Logic-level translators provide proper interface translation from the MAX5861 digital signals to the USB and VC707 circuitry.

The EV kit includes a hardware kit that allows the board to be lifted off of a lab bench as well as mate to the VC707 board. The hardware kit includes standoffs, screws, metal washers, and nylon washers. The order from the top of the board to the bottom should be as follows: pan screw, metal washer, MAX5861 EV kit board mounting through-hole, nylon washer, and standoff. The hardware kit components should be placed on the outer 3 corners of the EV kit as well as the mounting hole inside the FMC connector holes.

### Power Supplies

The EV kit operates from a single 5.0V/3A power supply applied at the 5.0V_IN and GND PCB pads. The MAX15023 dual synchronous step-down controller,

MAX1793 LDO regulator, and MAX8902A low-noise LDO regulator provide power to the EV kit's power rails. The MAX15023 devices are configured to 0.9V and are used for on-board regulation of the MAX5861 core supply input, VDD09. The MAX1793 LDO regulators are configured to 1.8V and 3.3V and are used for on-board regulation of the MAX5861 VDD18, AVCLK, and AVDD3 supply inputs. The MAX8902A LDO regulators are configured to 1.8V and are used for on-board regulation of the AVDD18 supply inputs. The EV kit also provides the option of using external power supplies. When using an external supply for the VDD09 supply rail, remove resistor R17 and apply a 0.9V/5A power supply at the DIGITAL09 relative to GND PCB pads. See Table 1 for proper shunt settings for all of the other EV kit power-supply inputs. A 1.8V/3A power supply will then be applied on the DIGITAL18, ANALOG18, ANALOGCLK, and ANALOG33 pads.

Test points are available at VDD09, AVDD18, AVCLK, AVDD3, VDD18_IN, and GND for monitoring the EV kit power-supply voltages.

Jumpers JU24 and JU25 are provided for sensing the MAX5861 VDD09 voltage. Remove the shunts at jumper JU1 and JU8 and connect the voltmeter positive and negative terminals at JU24 pin 1 and JU25 pin 2 respectively to monitor VDD09 voltage as monitored inside the MAX5861 IC.

### SPI and SMBus Interface Control

The EV kit communicates to the MAX5861 SPI interface using the on-board USB circuitry. Place shunts across pins 1-2, 4-5, 7-8, 10-11 and 13-14 of jumper JU7 to control the SPI interface using the USB circuitry.

The EV kit communicates to the MAX6654 SMBus interface using the on-board USB circuitry. Place shunts across pins 1-2 and 4-5 of jumper JU23 to control the SMBus interface using the USB circuitry.

### Global Reset

Momentary pushbutton switch RSTN is used as a global reset to clear all MAX5861 configuration registers. A global reset is required when uploading a new test configuration on the **MAX5861 Register** tab. Press the RSTN switch to clear the registers before uploading a new test configuration file.

Test configuration files are available for loading into the MAX5861, via the SPI interface. The MAX5861 output is available for viewing at the external OUT SMA connector.

### MODE2 Pushbutton Switch

Momentary pushbutton switch MODE2 is available for providing a low-to-high pulse at the MAX5861 port MODE2 input when pressed.

The pushbutton switch is used to perform various functions within the MAX5861. Refer to the MAX5861 IC data sheet for additional information.

### Using the VC707 Virtex FPGA Board with the MAX5861 EV Kit

The MAX5861 device has the ability to produce PRBS data for both the SCQAM and OFDM ports. This PRBS is inserted in place of the input interface data if programmed. The MAX5861 device can be fully evaluated without the need for the VC707. But if input data is desired, the FPGA platform (VC707) can be used as a data source for the MAX5861 EV kit. The VC707 drives the MAX5861 multiplexed input ports, control signals and monitors the ready logic outputs. The EV kit software provides the **VC707** tab for controlling the VC707.

The VC707 and MAX5861 EV kit boards can be connected using the available FMC interface connector. See Figure 4 for details in connecting the boards together. Alternatively, the VC707 and MAX5861 EV kit can be connected with coaxial ribbon cables (Part No.: Samtec HDR-169468-01). Note that it is necessary to use recommended hardware and/or cables to get a reliable electrical connection between the boards.

## Component List, Schematics, and PCB Layout Diagrams

Refer to the following files attached to this data sheet for component information, schematics, PCB layout diagrams:

- **BOM_MAX5861EVKIT_REVA.xlsx**
- **Schematic_MAX5861_evkit_reva.PDF**
- **PCB_MAX5861_evkit_reva.PDF**

## Component Suppliers

| SUPPLIER | PHONE | WEBSITE |
|---|---|---|
| Fairchild Semiconductor | 888-522-5372 | www.fairchildsemi.com |
| Hong Kong X'tals Ltd. | 852-35112388 | www.hongkongcrystal.com |
| Murata Electronics North America, Inc. | 770-436-1300 | www.murata-northamerica.com |
| Panasonic Corp. | 800-344-2112 | www.panasonic.com |
| Taiyo Yuden | 800-348-24120 | www.t-yuden.com |
| TDK Corp. | 847-803-6100 | www.component.tdk.com |

***Note:*** *Indicate that you are using the MAX5861s when contacting these component suppliers.*

## Ordering Information

| PART | TYPE | TYPE |
|---|---|---|
| MAX5861EVKIT# | EV Kit | 160-Channel SCQAM and 6-Channel OFDM Cable Modulator |
| EK-V7-VC707-G | FPGA Board | Xilinx Virtex 7 FPGA Board – Ordered through a qualified Xilinx Vendor |

## Revision History

| REVISION NUMBER | REVISION DATE | DESCRIPTION | PAGES CHANGED |
|---|---|---|---|
| 0 | 6/15 | Initial release | — |

## Configuration Load Sequence
## Output Example

The following is a Results Log capture of a configuration load sequence. To make the configuration easier to follow, repeated functions and/or writes (i.e. same or similar register values to channels 1 through 128, etc) will be condensed and noted as such with ellipses between the initial write and the final write within the configuration load file. The sequence uses the **Test** tab and configures the device for a single-carrier SCQAM. It then checks the **SCQAM Status** tab for the current settings. This is followed by a reset and reconfiguration using the **Register** tab with a (128) SCQAM plus (2) 192MHz bandwidth OFDM configuration from the input interface. The FPGA is then configured for a (128) channel configuration, the memory is loaded with a pattern, the SDCLK is enabled and the memory is started. The next step was to check the **OFDM Status** tab for the current settings. The final step is to look at the **Device Status** tab. Especially given the order of operation, the FIFOs and parity might have startup errors. The reading of these registers clears these errors and subsequent reads will be valid for operating status.

**Test Tab – Load Single SCQAM:**

```
//Test:  Display Sample Pictures for Configurations
//  File Configuration Load, Board Address  0x0
 Write Reg:  0x026    0x000A000A
  //-->  USB/FTDI Write Good/Verified
 Write Reg:  0x008    0x000007BF
  //-->  USB/FTDI Write Good/Verified
 Write Reg:  0x081    0x000FFFFE
  //-->  USB/FTDI Write Good/Verified
 Write Reg:  0x080    0x0000105E
  //-->  USB/FTDI Write Good/Verified
 Write Reg:  0x088    0x00000401
  //-->  USB/FTDI Write Good/Verified
 Write Reg:  0x109    0x000001A0
  //-->  USB/FTDI Write Good/Verified
 Write Reg:  0x10A    0x000002E9
  //-->  USB/FTDI Write Good/Verified
 Write Reg:  0x10B    0x00023000
  //-->  USB/FTDI Write Good/Verified
 Write Reg:  0x10C    0x00FE028A
  //-->  USB/FTDI Write Good/Verified
 Write Reg:  0x101    0x00078000
  //-->  USB/FTDI Write Good/Verified
 Write Reg:  0x061    0x0003C000
  //-->  USB/FTDI Write Good/Verified
 Write Reg:  0x041    0x00080F55
  //-->  USB/FTDI Write Good/Verified
 Write Reg:  0x108    0x00D00616
  //-->  USB/FTDI Write Good/Verified
 Write Reg:  0x100    0x00000001
  //-->  USB/FTDI Write Good/Verified
 Write Reg:  0x060    0x00000001
```

//--> USB/FTDI Write Good/Verified

Write Reg: 0x040 0x00000000

//--> USB/FTDI Write Good/Verified

Write Reg: 0x0D8 0x80000000

//--> USB/FTDI Write

Write Reg: 0x042 0x010000FE

//--> USB/FTDI Write Good/Verified

Write Reg: 0x047 0x00008000

//--> USB/FTDI Write Good/Verified

Write Reg: 0x040 0x00000003

//--> USB/FTDI Write Good/Verified

Write Reg: 0x001 0x87800000

//--> USB/FTDI Write Good/Verified

//Register: Read Modify Write Reg: 0x0D8 data 0x00000000 mask 0x00000020

Read Reg: 0x0D8 0x80001010 //--> USB/FTDI Read Good

Write Reg: 0x0D8 0x80001010

//--> USB/FTDI Write

// File Loaded: C:\Test\MAX5861EVKIT/ConfigurationLoadFiles/SCQAM/001SB256_5861_PRBS_4915p2M_1000M.txt

Loaded file from ConfigurationLoadFiles : C:\Test\MAX5861EVKIT/ConfigurationLoadFiles/SCQAM/001SB256_5861_PRBS_4915p2M_1000M.txt

**QAM Status Tab:**

//QAM Status: Get QAM Channel Status for Channel Number 1

Read Reg: 0x088 0x00000001 //--> USB/FTDI Read Good

Read Reg: 0x080 0x0000105E //--> USB/FTDI Read Good

// 32 chan mute bit 0 with result of 0

Read Reg: 0x081 0x000FFFFE //--> USB/FTDI Read Good

// 8 chan mute bit 0 with result of 0

Read Reg: 0x009 0x00000040 //--> USB/FTDI Read Good

// 32 chan powerdown 6 with result of 1

// SYM_INTF 0x00000040

Read Reg: 0x108 0x00100616 //--> USB/FTDI Read Good

// SYMIF 0x00100616

Read Reg: 0x10A 0x000002E9 //--> USB/FTDI Read Good

// LF 0x000002E9

Read Reg: 0x109 0x000001A0 //--> USB/FTDI Read Good

// KF 0x000001A0

Read Reg: 0x10C 0x00FE028A //--> USB/FTDI Read Good

// G1G2 0x00FE028A

Read Reg: 0x042 0x010000FE //--> USB/FTDI Read Good

// G5G6 0x010000FE

Read Reg: 0x10B 0x00023000 //--> USB/FTDI Read Good
// NCOA 0x00023000
Read Reg: 0x101 0x00078000 //--> USB/FTDI Read Good
// NCO2 0x00078000
Read Reg: 0x061 0x0003C000 //--> USB/FTDI Read Good
// NCO3 0x0003C000
Read Reg: 0x041 0x00080F55 //--> USB/FTDI Read Good
// NCO4 0x00080F55

//QAM Status: Get Channel fOUT for Channel Number 1
Read Reg: 0x10B 0x00023000 //--> USB/FTDI Read Good
// NCOA 0x00023000
Read Reg: 0x101 0x00078000 //--> USB/FTDI Read Good
// NCO2 0x00078000
Read Reg: 0x061 0x0003C000 //--> USB/FTDI Read Good
// NCO3 0x0003C000
Read Reg: 0x041 0x00080F55 //--> USB/FTDI Read Good
// NCO4 0x00080F55
// Value Signed Mag (1) 143360.000 (2) 491520.000 (3) 245760.000 (4) 528213.000
// NCO Freq Offset in Hz (1) 21000000.000 (2) 72000000.000 (3) 288000000.000 (4) 618999609.375
// Calculated Channel Offset ==> 999.999609375 MHz

**Register Tab – Toggle RSTN:**

//Register: Toggle Bit RSTN
// Reset also Resets The DAC DLL - Clear Lock Status
//Register: Read Modify Write Reg: 0x0D8 data 0x00000000 mask 0x00000020
Read Reg: 0x0D8 0x83FF10B4 //--> USB/FTDI Read Good
Write Reg: 0x0D8 0x83FF1094
//--> USB/FTDI Write

//Register: Toggle Bit RSTN

**Register Tab: Load File Configuration (128 SCQAM plus 2 OFDM):**

//Register: File IO Browse, Board Address 0x0
Write Reg: 0x026 0x000A000A
//--> USB/FTDI Write Good/Verified
Write Reg: 0x008 0x0000043C
//--> USB/FTDI Write Good/Verified
Write Reg: 0x081 0x000F0000
//--> USB/FTDI Write Good/Verified
Write Reg: 0x080 0x00008050
//--> USB/FTDI Write Good/Verified
Write Reg: 0x0B1 0x04540453
//--> USB/FTDI Write Good/Verified
Write Reg: 0x74F 0x00000F00

//--> USB/FTDI Write Good/Verified
Write Reg: 0x0B2 0x04560455
//--> USB/FTDI Write Good/Verified
…
Write Reg: 0x093 0x04180417
//--> USB/FTDI Write Good/Verified
Write Reg: 0x7BC 0x00040008
//--> USB/FTDI Write Good/Verified
Write Reg: 0x7B0 0x00040008
//--> USB/FTDI Write Good/Verified
Write Reg: 0x0A2 0x04360435
//--> USB/FTDI Write Good/Verified
…
Write Reg: 0x0B5 0x045C045B
//--> USB/FTDI Write Good/Verified
Write Reg: 0x109 0x000001A0
//--> USB/FTDI Write Good/Verified
Write Reg: 0x10A 0x000002E9
//--> USB/FTDI Write Good/Verified
…
Write Reg: 0x5F1 0x000001A0
//--> USB/FTDI Write Good/Verified
Write Reg: 0x5F2 0x000002E9
//--> USB/FTDI Write Good/Verified
Write Reg: 0x10B 0x00063000
//--> USB/FTDI Write Good/Verified
Write Reg: 0x113 0x00059000
//--> USB/FTDI Write Good/Verified
…
Write Reg: 0x5F3 0x00023000
//--> USB/FTDI Write Good/Verified
Write Reg: 0x10C 0x000A028A
//--> USB/FTDI Write Good/Verified
…
Write Reg: 0x4DC 0x000A028A
//--> USB/FTDI Write Good/Verified
Write Reg: 0x744 0x065C074E
//--> USB/FTDI Write Good/Verified
…
Write Reg: 0x758 0x0000079C
//--> USB/FTDI Write Good/Verified
Write Reg: 0x101 0x00178000
//--> USB/FTDI Write Good/Verified

…
Write Reg: 0x381 0x00178000
//--> USB/FTDI Write Good/Verified
Write Reg: 0x061 0x0013C000
//--> USB/FTDI Write Good/Verified
Write Reg: 0x065 0x00114000
//--> USB/FTDI Write Good/Verified
Write Reg: 0x069 0x00014000
//--> USB/FTDI Write Good/Verified
Write Reg: 0x06D 0x0003C000
//--> USB/FTDI Write Good/Verified
Write Reg: 0x7B1 0x00064000
//--> USB/FTDI Write Good/Verified
Write Reg: 0x7BD 0x0008C000
//--> USB/FTDI Write Good/Verified
Write Reg: 0x041 0x0005B400
//--> USB/FTDI Write Good/Verified
Write Reg: 0x108 0x00C00000
//--> USB/FTDI Write Good/Verified
…
Write Reg: 0x5F0 0x00C00000
//--> USB/FTDI Write Good/Verified
Write Reg: 0x100 0x00000001
//--> USB/FTDI Write Good/Verified
…
Write Reg: 0x5B0 0x00000001
//--> USB/FTDI Write Good/Verified
Write Reg: 0x060 0x00000001
//--> USB/FTDI Write Good/Verified
…
Write Reg: 0x06C 0x00000001
//--> USB/FTDI Write Good/Verified
Write Reg: 0x7B2 0x00013823
//--> USB/FTDI Write Good/Verified
Write Reg: 0x7BE 0x00013823
//--> USB/FTDI Write Good/Verified
Write Reg: 0x040 0x00000000
//--> USB/FTDI Write Good/Verified
Write Reg: 0x0D8 0x80000000
//--> USB/FTDI Write
Write Reg: 0x042 0x010000FE
//--> USB/FTDI Write Good/Verified
Write Reg: 0x047 0x00008000

//--> USB/FTDI Write Good/Verified

Write Reg: 0x040 0x00000003

//--> USB/FTDI Write Good/Verified

Write Reg: 0x001 0x87800000

//--> USB/FTDI Write Good/Verified

//Register: Read Modify Write Reg: 0x0D8 data 0x00000000 mask 0x00000020

Read Reg: 0x0D8 0x80001010 //--> USB/FTDI Read Good

Write Reg: 0x0D8 0x80001010

//--> USB/FTDI Write

// File Loaded: C:\Test\MAX5861EVKIT\ConfigurationLoadFiles\Mix\128SB256and002OB1024_5861_input_630M_192BW_50k_1024c256r_P.txt

//VC707: File I/O

**VC707 Tab – Load Configuration for 128 Channels:**

// Loading File Into FPGA : C:\Test\MAX5861EVKIT\FPGAConfigurationFiles\128Sxxx_VC707_SPinput_TS128_PRBS_PCLK633M.txt

//VC707: Send Command leds 00000000

//VC707: Return leds 00000000 ACK #

//VC707: Send Command si570 set 633.0

//VC707: Return si570 set 633.0 00 42 C4 F6 63 11 ACK #

//VC707: Send Command leds 0000000b

//VC707: Return leds 0000000b ACK #

//VC707: Send Command type set all prbs

//VC707: Return type set all prbs ACK #

//VC707: Send Command prbs set 1 0 1558 1 off 0 even

//VC707: Return prbs set 1 0 1558 1 off 0 even ACK #

…

//VC707: Send Command prbs set 128 0 2356 1 off 0 even

//VC707: Return prbs set 128 0 2356 1 off 0 even ACK #

//VC707: Send Command leds 000000b1

//VC707: Return leds 000000b1 ACK #

//VC707: Send Command prbs reset all off

//VC707: Return prbs reset all off ACK #

//VC707: Send Command prbs enable all on

//VC707: Return prbs enable all on ACK #

//VC707: Send Command leds 00000111

//VC707: Return leds 00000111 ACK #

//VC707: Send Command pll reset qam on

//VC707: Return pll reset qam on ACK #

//VC707: Send Command pll clksel qam clkin1

//VC707: Return pll clksel qam clkin1 ACK #

//VC707: Send Command pll reset qam off

//VC707: Return pll reset qam off ACK #

```
//VC707:  Send Command  pll qam info
//VC707:   Return  pll qam info  inclk - ok | fbclk - ok | outclk - locked  ACK  #
//VC707:  Send Command  oserdes reset adat on
//VC707:   Return  oserdes reset adat on  ACK  #
//VC707:  Send Command  oserdes reset adat off
//VC707:   Return  oserdes reset adat off  ACK  #
//VC707:  Send Command  leds 00001111
//VC707:   Return  leds 00001111  ACK  #
//VC707:  Send Command  bank set adat 16
//VC707:   Return  bank set adat 16  ACK  #
//VC707:  Send Command  ready reset on
//VC707:   Return  ready reset on  ACK  #
//VC707:  Send Command  ready channel 1
//VC707:   Return  ready channel 1  ACK  #
//VC707:  Send Command  delay reset pulse
//VC707:   Return  delay reset pulse  ACK  #
//VC707:  Send Command  oserdes enable pclk on
//VC707:   Return  oserdes enable pclk on  ACK  #
//VC707:  Send Command  delay set pclk 7
//VC707:   Return  delay set pclk 7  ACK  #
//VC707:  Send Command  oserdes enable psync on
//VC707:   Return  oserdes enable psync on  ACK  #
//VC707:  Send Command  leds 00011111
//VC707:   Return  leds 00011111  ACK  #
//VC707:  Send Command  ready bypass on
//VC707:   Return  ready bypass on  ACK  #
//VC707:  Send Command  ready bypass set 0 FEFEFEFE
//VC707:   Return  ready bypass set 0 FEFEFEFE  ACK  #
//VC707:  Send Command  ready bypass set 1 FEFEFEFE
//VC707:   Return  ready bypass set 1 FEFEFEFE  ACK  #
//VC707:  Send Command  ready bypass set 2 FEFEFEFE
//VC707:   Return  ready bypass set 2 FEFEFEFE  ACK  #
//VC707:  Send Command  ready bypass set 3 FEFEFEFE
//VC707:   Return  ready bypass set 3 FEFEFEFE  ACK  #
//VC707:  Send Command  ready bypass set 4 FEFEFEFE
//VC707:   Return  ready bypass set 4 FEFEFEFE  ACK  #
//VC707:  Send Command  ready bypass set 5 00000000
//VC707:   Return  ready bypass set 5 00000000  ACK  #
//VC707:  Send Command  leds 00011111
//VC707:   Return  leds 00011111  ACK  #
…
//VC707:  Send Command  leds 00111111
//VC707:   Return  leds 00111111  ACK  #
```

//VC707: Send Command bank enable adat on

//VC707: Return bank enable adat on ACK #

//VC707: Send Command oserdes enable adat on

//VC707: Return oserdes enable adat on ACK #

//VC707: Send Command leds 00011111

//VC707: Return leds 00011111 ACK #

…

//VC707: Send Command leds 00111111

//VC707: Return leds 00111111 ACK #

//VC707: Send Command ready reset off

//VC707: Return ready reset off ACK #

//VC707: Send Command leds 00111111

//VC707: Return leds 00111111 ACK #

…

//VC707: Send Command leds 01111111

//VC707: Return leds 01111111 ACK #

//VC707: Send Command ready bypass off

//VC707: Return ready bypass off ACK #

//VC707: Send Command ready bypass set 0 FFFFFFFF

//VC707: Return ready bypass set 0 FFFFFFFF ACK #

//VC707: Send Command ready bypass set 1 FFFFFFFF

//VC707: Return ready bypass set 1 FFFFFFFF ACK #

//VC707: Send Command ready bypass set 2 FFFFFFFF

//VC707: Return ready bypass set 2 FFFFFFFF ACK #

//VC707: Send Command ready bypass set 3 FFFFFFFF

//VC707: Return ready bypass set 3 FFFFFFFF ACK #

//VC707: Send Command ready bypass set 4 00000000

//VC707: Return ready bypass set 4 00000000 ACK #

//VC707: Send Command ready bypass set 5 00000000

//VC707: Return ready bypass set 5 00000000 ACK #

//VC707: Send Command leds 11111111

//VC707: Return leds 11111111 ACK #

//VC707: Loaded FPGA Configuration File C:\Test\MAX5861EVKIT\FPGAConfigurationFiles\128Sxxx_VC707_SPinput_TS128_PRBS_PCLK633M.txt into the FPGA Successfully

**VC707 Tab – Load Memory File for OFDM:**

//VC707: Load Memory File Into VC707

//VC707: Send Command delay set sdclk 5

//VC707: Return delay set sdclk 5 ACK #

// DDR3 Memory File Load Update: Lines Written is -->2048

// DDR3 Memory File Load Update: Lines Written is -->4096

…

// DDR3 Memory File Load Update: Lines Written is -->71680

//VC707: Send Command to Read Memory ddr3 read 0 1

//VC707: Read 1 Blocks of DDR3 Memory

//VC707: Reading Block #0 of 1

//VC707: Reading Complete - Now Formatting Data

//VC707: Bytes From Block 2 : Read 1 Blocks of DDR3 Memory

//VC707:

11111100 11111100 11111100 11111100

11111100 11111100 11111100 11111100

00000000 00000000 00000000 00000000

00000000 00000000 00000000 00000000

//VC707: Read From DDR3 Memory: Read 1 Blocks of DDR3 Memory

//VC707: DDR3 Read Results Written to File 'VC707DDR3ReadMemoryResults.txt'

// CRC Verfied Good and All FPGA Communications Returned ACKs

//VC707: Loaded DDR3 Memory File C:\Test\MAX5861EVKIT\FPGAConfigurationFiles\VC707Memfile_Fill_50kSpace4kIFFT_BW192M_1024QAM_CP1024RP256_14symb.txt into Memory Successfully

//VC707: Send Command ofdm memory 1 0 17920

//VC707: Return ofdm memory 1 0 17920 ACK #

**VC707 Tab – Enable SDCLK:**

//VC707: Send Command leds tt0000tt

//VC707: Return leds tt0000tt ACK #

//VC707: Enable the SDCLK Output

//VC707: Send Command leds 00111111

//VC707: Return leds 00111111 ACK #

//VC707: Send Command delay reset pulse

//VC707: Return delay reset pulse ACK #

//VC707: Send Command oserdes enable pclk on

//VC707: Return oserdes enable pclk on ACK #

//VC707: Send Command leds 01111111

//VC707: Return leds 01111111 ACK #

//VC707: Send Command pll reset ofdm on

//VC707: Return pll reset ofdm on ACK #

//VC707: Send Command pll clksel ofdm clkin1

//VC707: Return pll clksel ofdm clkin1 ACK #

//VC707: Send Command pll reset ofdm off

//VC707: Return pll reset ofdm off ACK #

//VC707: Send Command pll ofdm info

//VC707: Return pll ofdm info inclk - ok | fbclk - ok | outclk - locked ACK #

//VC707: Send Command oserdes reset sdclk on

//VC707: Return oserdes reset sdclk on ACK #

//VC707: Send Command oserdes reset sdclk off

//VC707: Return oserdes reset sdclk off ACK #

//VC707: Send Command oserdes enable sdclk on
//VC707: Return oserdes enable sdclk on ACK #
//VC707: Send Command leds 00000111
//VC707: Return leds 00000111 ACK #
//VC707: Send Command leds 00000001
//VC707: Return leds 00000001 ACK #
//VC707: Send Command oserdes enable bcsync on
//VC707: Return oserdes enable bcsync on ACK #
//VC707: Send Command oserdes enable bdat on
//VC707: Return oserdes enable bdat on ACK #
//VC707: Send Command oserdes enable cdat on
//VC707: Return oserdes enable cdat on ACK #
//VC707: Send Command leds tt1111tt
//VC707: Return leds tt1111tt ACK #

**VC707 Tab – Start Memory Flow for OFDM:**

//VC707: Start OFDM Data Flow from Memory
//VC707: Start Memory from Stream 1
//VC707: Send Command ofdm disable
//VC707: Return ofdm disable ACK #
//VC707: Send Command ofdm enable 1 on
//VC707: Return ofdm enable 1 on enable stream: 1 length: 17920 mempos: 0 ACK #
//VC707: Send Command leds tbhqqhbt
//VC707: Return leds tbhqqhbt ACK #

**OFDM Status Tab:**

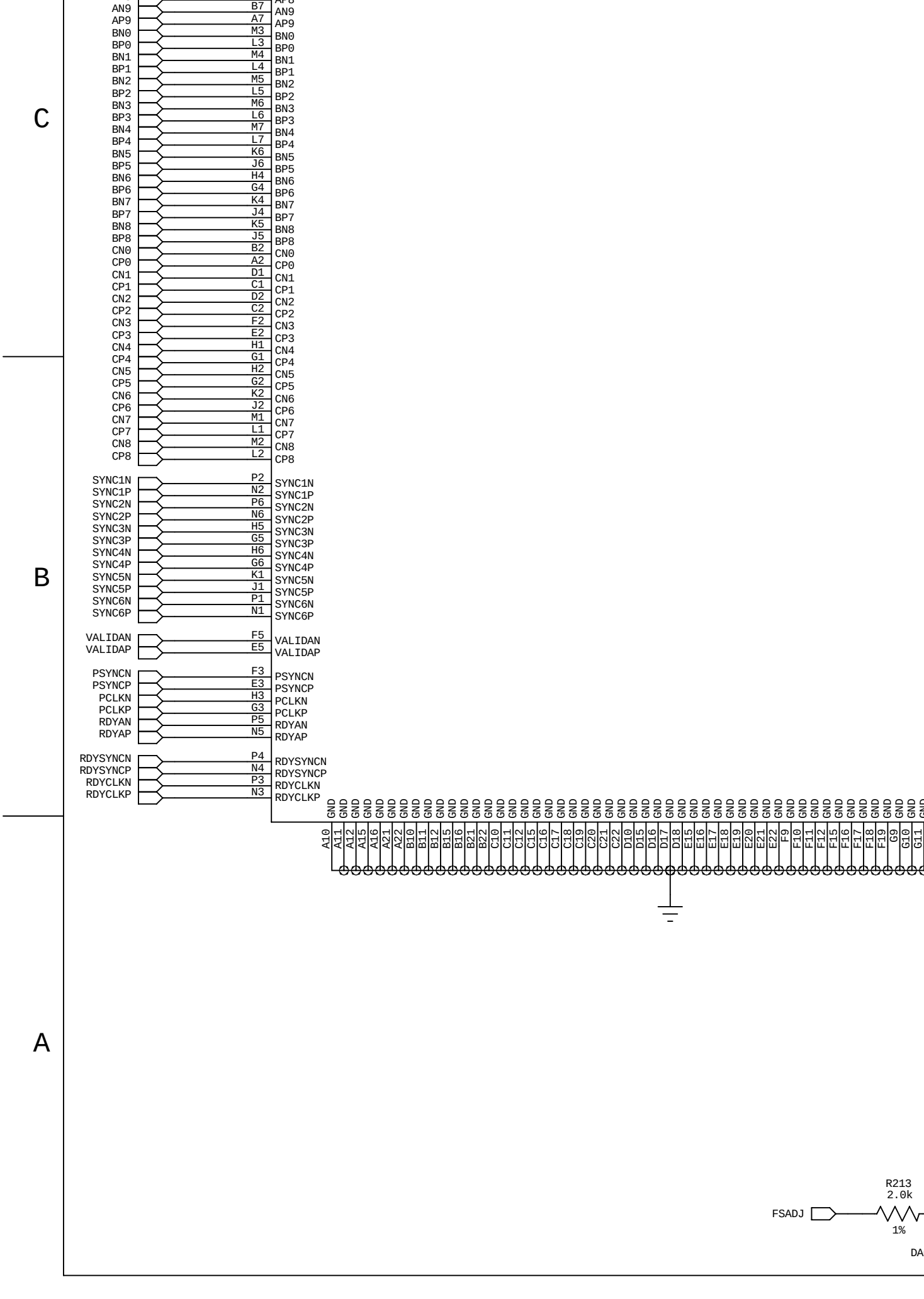
//OFDM Status: Get OFDM Channel Status for Channel Number 1
Read Reg: 0x741 0x0000FF00 //--> USB/FTDI Read Good
Read Reg: 0x009 0x000003C3 //--> USB/FTDI Read Good
// OFDM chan powerdown 0 with result of 1
// OFDM_CFG 0x000003C3
Read Reg: 0x74B 0x00000006 //--> USB/FTDI Read Good
Read Reg: 0x74C 0x01EFE080 //--> USB/FTDI Read Good
Read Reg: 0x740 0x00029FFE //--> USB/FTDI Read Good
Read Reg: 0x7B0 0x00040008 //--> USB/FTDI Read Good
Read Reg: 0x7B1 0x00064000 //--> USB/FTDI Read Good
Read Reg: 0x041 0x0005B400 //--> USB/FTDI Read Good
// Value Signed Mag (3) 409600.000 (4) 373760.000
// NCO Freq Offset in Hz (3) 480000000.000 (4) 438000000.000
Read Reg: 0x7B5 0x1100006F //--> USB/FTDI Read Good
//Register: Read Modify Write Reg: 0x741 data 0x00004000 mask 0x00004000
Read Reg: 0x741 0x0000FF00 //--> USB/FTDI Read Good
Write Reg: 0x741 0x0000FF00
//--> USB/FTDI Write Good/Verified
Read Reg: 0x7B2 0x00003823 //--> USB/FTDI Read Good

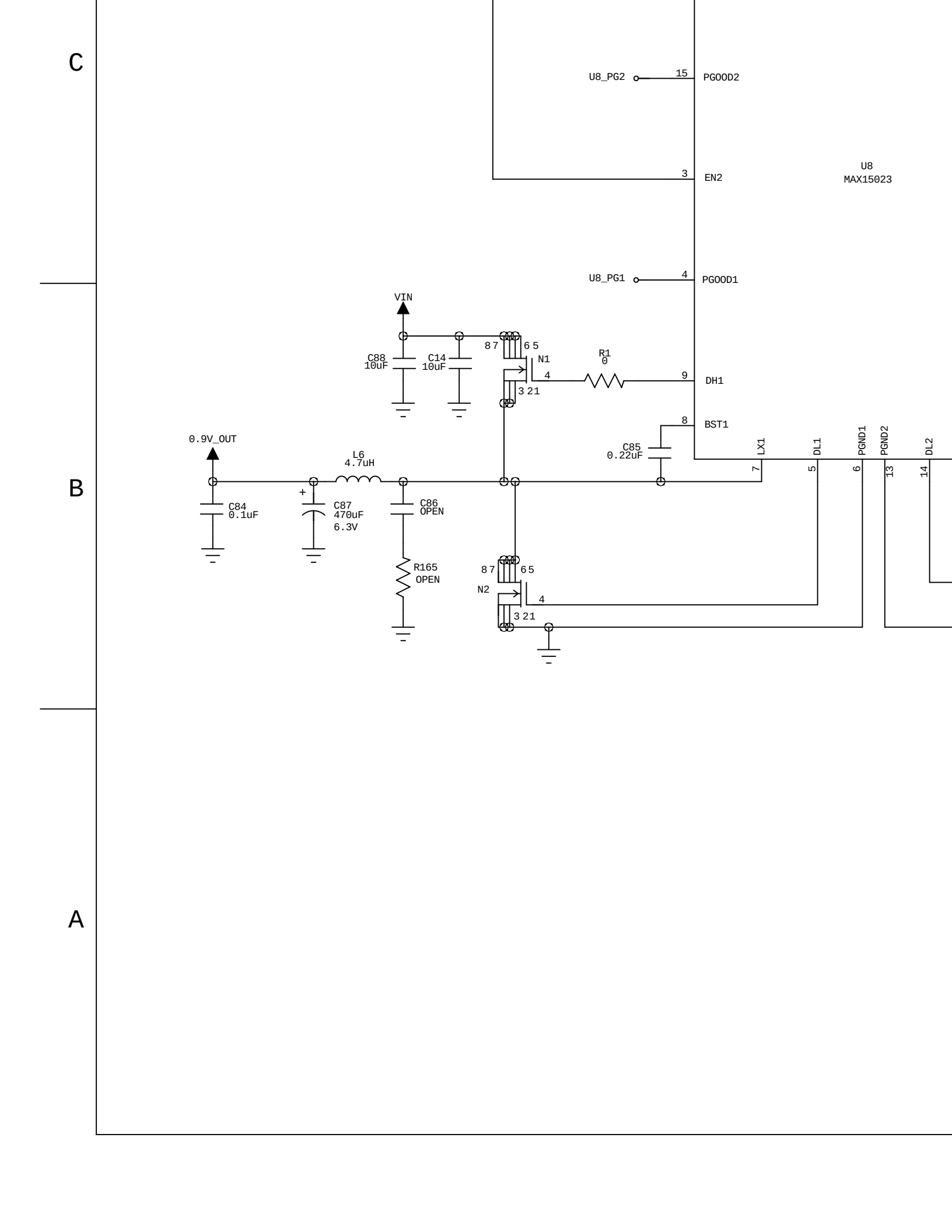
Read Reg: 0x744 0x065C074E //--> USB/FTDI Read Good
Read Reg: 0x745 0x0550034E //--> USB/FTDI Read Good
Read Reg: 0x746 0x044E0472 //--> USB/FTDI Read Good
Read Reg: 0x747 0x00800370 //--> USB/FTDI Read Good
Read Reg: 0x748 0x024E0270 //--> USB/FTDI Read Good
Read Reg: 0x749 0x014E0170 //--> USB/FTDI Read Good
Read Reg: 0x74A 0x0000079C //--> USB/FTDI Read Good

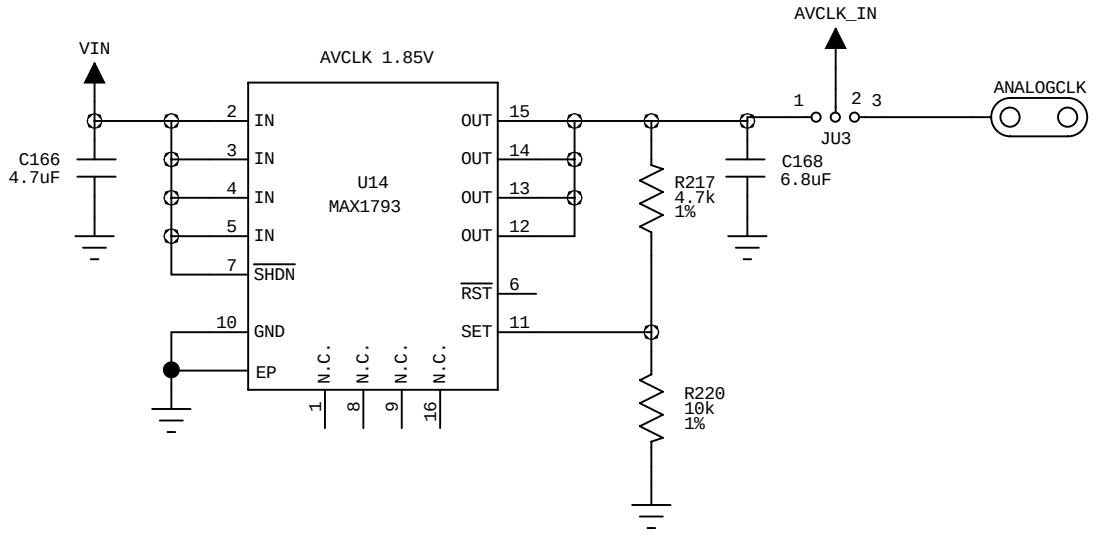
**Device Status Tab:**

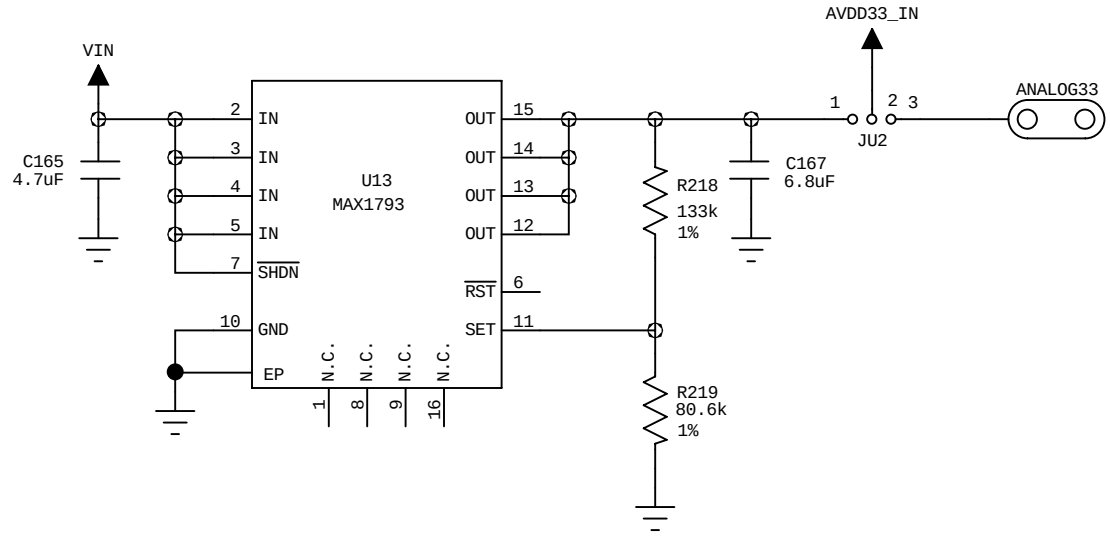
//Device Status: I2C Temperature Read
Write I2C Address: 78 Command 9 Data 32 //--> USB/FTDI Write Good
Write I2C Address: 78 Command 1 Read I2C Data: 93 //--> USB/FTDI Read Good
//Device Status: FIFO Status
Read Reg: 0x0D9 0x00086F0F //--> USB/FTDI Read Good
//Register: Read Modify Write Reg: 0x0DA data 0x00000000 mask 0xFFFFFFFF
Read Reg: 0x0DA 0xFFFFFFFF //--> USB/FTDI Read Good
Write Reg: 0x0DA 0x00000000
//--> USB/FTDI Write Good/Verified
…
//Register: Read Modify Write Reg: 0x0E3 data 0x00000000 mask 0xFFFFFFFF
Read Reg: 0x0E3 0x00000000 //--> USB/FTDI Read Good
Write Reg: 0x0E3 0x00000000
//--> USB/FTDI Write Good/Verified
Read Reg: 0x741 0x0000FF00 //--> USB/FTDI Read Good
Read Reg: 0x74F 0x0000FF00 //--> USB/FTDI Read Good
Read Reg: 0x75D 0x00020F01 //--> USB/FTDI Read Good
Read Reg: 0x76C 0x00020F01 //--> USB/FTDI Read Good
Read Reg: 0x77A 0x00020F01 //--> USB/FTDI Read Good
Read Reg: 0x788 0x00020F01 //--> USB/FTDI Read Good
//Register: Read Modify Write Reg: 0x741 data 0x00000000 mask 0x00003000
Read Reg: 0x741 0x0000FF00 //--> USB/FTDI Read Good
Write Reg: 0x741 0x0000CF00
//--> USB/FTDI Write Good/Verified
…
//--> USB/FTDI Write Good/Verified
//Register: Read Modify Write Reg: 0x788 data 0x00000000 mask 0x00003000
Read Reg: 0x788 0x00020F01 //--> USB/FTDI Read Good
Write Reg: 0x788 0x00020F01
//--> USB/FTDI Write Good/Verified
//Device Status: Parity Status
Read Reg: 0x0D8 0x80001098 //--> USB/FTDI Read Good
//Register: Read Modify Write Reg: 0x0D8 data 0x00000000 mask 0x00000058
Read Reg: 0x0D8 0x80001098 //--> USB/FTDI Read Good
Write Reg: 0x0D8 0x80001080

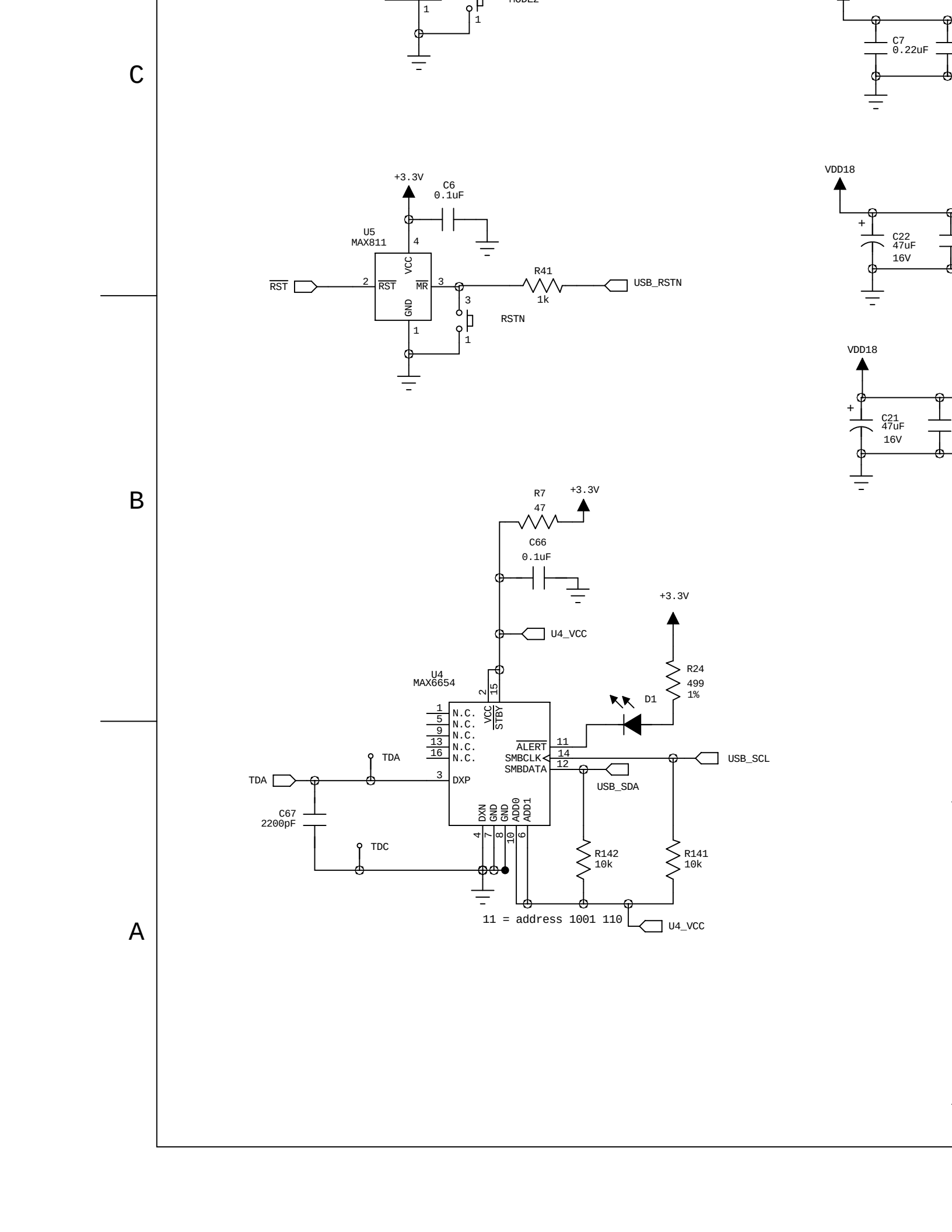
```
  //-->  USB/FTDI Write
 //Device Status:  Lock Status
  Read Reg:  0x0D8    0x80001080   //--> USB/FTDI Read Good
 //Register:  Read Modify Write Reg:  0x0D8  data  0x00000000  mask  0x00000020
  Read Reg:  0x0D8    0x80001080   //--> USB/FTDI Read Good
  Write Reg:  0x0D8    0x80001080
  //-->  USB/FTDI Write
 //Device Status:  FIFO Status
  Read Reg:  0x0D9    0x00086000   //--> USB/FTDI Read Good
 //Register:  Read Modify Write Reg:  0x0DA  data  0x00000000  mask  0xFFFFFFFF
  Read Reg:  0x0DA    0x00000000   //--> USB/FTDI Read Good
  Write Reg:  0x0DA    0x00000000
  //-->  USB/FTDI Write Good/Verified
 …
  //-->  USB/FTDI Write Good/Verified
 //Register:  Read Modify Write Reg:  0x0E3  data  0x00000000  mask  0xFFFFFFFF
  Read Reg:  0x0E3    0x00000000   //--> USB/FTDI Read Good
  Write Reg:  0x0E3    0x00000000
  //-->  USB/FTDI Write Good/Verified
  Read Reg:  0x741    0x0000CF00   //--> USB/FTDI Read Good
  Read Reg:  0x74F    0x0000CF00   //--> USB/FTDI Read Good
  Read Reg:  0x75D    0x00020F01   //--> USB/FTDI Read Good
  Read Reg:  0x76C    0x00020F01   //--> USB/FTDI Read Good
  Read Reg:  0x77A    0x00020F01   //--> USB/FTDI Read Good
  Read Reg:  0x788    0x00020F01   //--> USB/FTDI Read Good
 //Register:  Read Modify Write Reg:  0x741  data  0x00000000  mask  0x00003000
  Read Reg:  0x741    0x0000CF00   //--> USB/FTDI Read Good
  Write Reg:  0x741    0x0000CF00
  //-->  USB/FTDI Write Good/Verified
 …
 //Register:  Read Modify Write Reg:  0x788  data  0x00000000  mask  0x00003000
  Read Reg:  0x788    0x00020F01   //--> USB/FTDI Read Good
  Write Reg:  0x788    0x00020F01
  //-->  USB/FTDI Write Good/Verified
 //Device Status:  Parity Status
  Read Reg:  0x0D8    0x80001080   //--> USB/FTDI Read Good
 //Register:  Read Modify Write Reg:  0x0D8  data  0x00000000  mask  0x00000058
  Read Reg:  0x0D8    0x80001080   //--> USB/FTDI Read Good
  Write Reg:  0x0D8    0x80001080
//-->  USB/FTDI Write
```

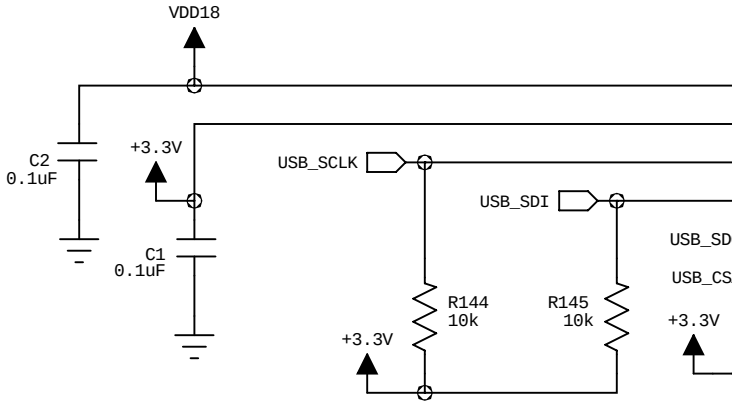

C
B
A
AN9 B7 AN9
AP9 A7 AP9
BN0 M3 BN0
BP0 L3 BP0
BN1 M4 BN1
BP1 L4 BP1
BN2 M5 BN2
BP2 L5 BP2
BN3 M6 BN3
BP3 L6 BP3
BN4 M7 BN4
BP4 L7 BP4
BN5 K6 BN5
BP5 J6 BP5
BN6 H4 BN6
BP6 G4 BP6
BN7 K4 BN7
BP7 J4 BP7
BN8 K5 BN8
BP8 J5 BP8
CN0 B2 CN0
CP0 A2 CP0
CN1 D1 CN1
CP1 C1 CP1
CN2 D2 CN2
CP2 C2 CP2
CN3 F2 CN3
CP3 E2 CP3
CN4 H1 CN4
CP4 G1 CP4
CN5 H2 CN5
CP5 G2 CP5
CN6 K2 CN6
CP6 J2 CP6
CN7 M1 CN7
CP7 L1 CP7
CN8 M2 CN8
CP8 L2 CP8
SYNC1N P2 SYNC1N
SYNC1P N2 SYNC1P
SYNC2N P6 SYNC2N
SYNC2P N6 SYNC2P
SYNC3N H5 SYNC3N
SYNC3P G5 SYNC3P
SYNC4N H6 SYNC4N
SYNC4P G6 SYNC4P
SYNC5N K1 SYNC5N
SYNC5P J1 SYNC5P
SYNC6N P1 SYNC6N
SYNC6P N1 SYNC6P
VALIDAN F5 VALIDAN
VALIDAP E5 VALIDAP
PSYNCN F3 PSYNCN
PSYNCP E3 PSYNCP
PCLKN H3 PCLKN
PCLKP G3 PCLKP
RDYAN P5 RDYAN
RDYAP N5 RDYAP
RDYSYNCN P4 RDYSYNCN
RDYSYNCP N4 RDYSYNCP
RDYCLKN P3 RDYCLKN
RDYCLKP N3 RDYCLKP
GND
A10 A11 A12 A15 A16 A21 A22 B10 B11 B12 B15 B16 B21 B22 C10 C11 C12 C15 C16 C17 C18 C19 C20 C21 C22 D10 D15 D16 D17 D18 E15 E16 E17 E18 E19 E20 E21 E22 F9 F10 F11 F12 F15 F16 F17 F18 F19 G9 G10 G11
R213
2.0k
FSADJ
1%
DA


U8_PG2
15
PGOOD2
U8
MAX15023
3
EN2
U8_PG1
4
PGOOD1
VIN
C88
10uF
C14
10uF
8 7
6 5
N1
4
3 21
R1
0
9
DH1
8
BST1
0.9V_OUT
L6
4.7uH
C85
0.22uF
LX1
7
DL1
5
PGND1
6
PGND2
13
DL2
14
C84
0.1uF
+
C87
470uF
6.3V
C86
OPEN
R165
OPEN
8 7
6 5
N2
4
3 21

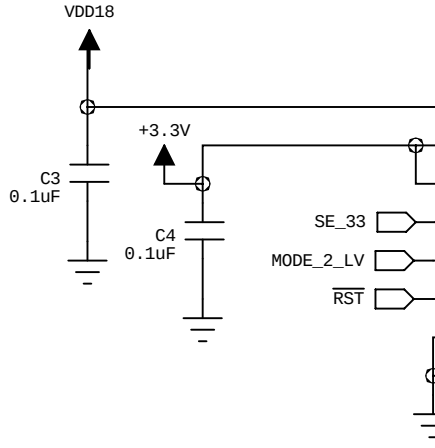

AVCLK_IN
VIN
AVCLK 1.85V
ANALOGCLK
C166
4.7uF
U14
MAX1793
IN
OUT
SHDN
RST
SET
GND
EP
N.C.
R217
4.7k
1%
C168
6.8uF
JU3
R220
10k
1%

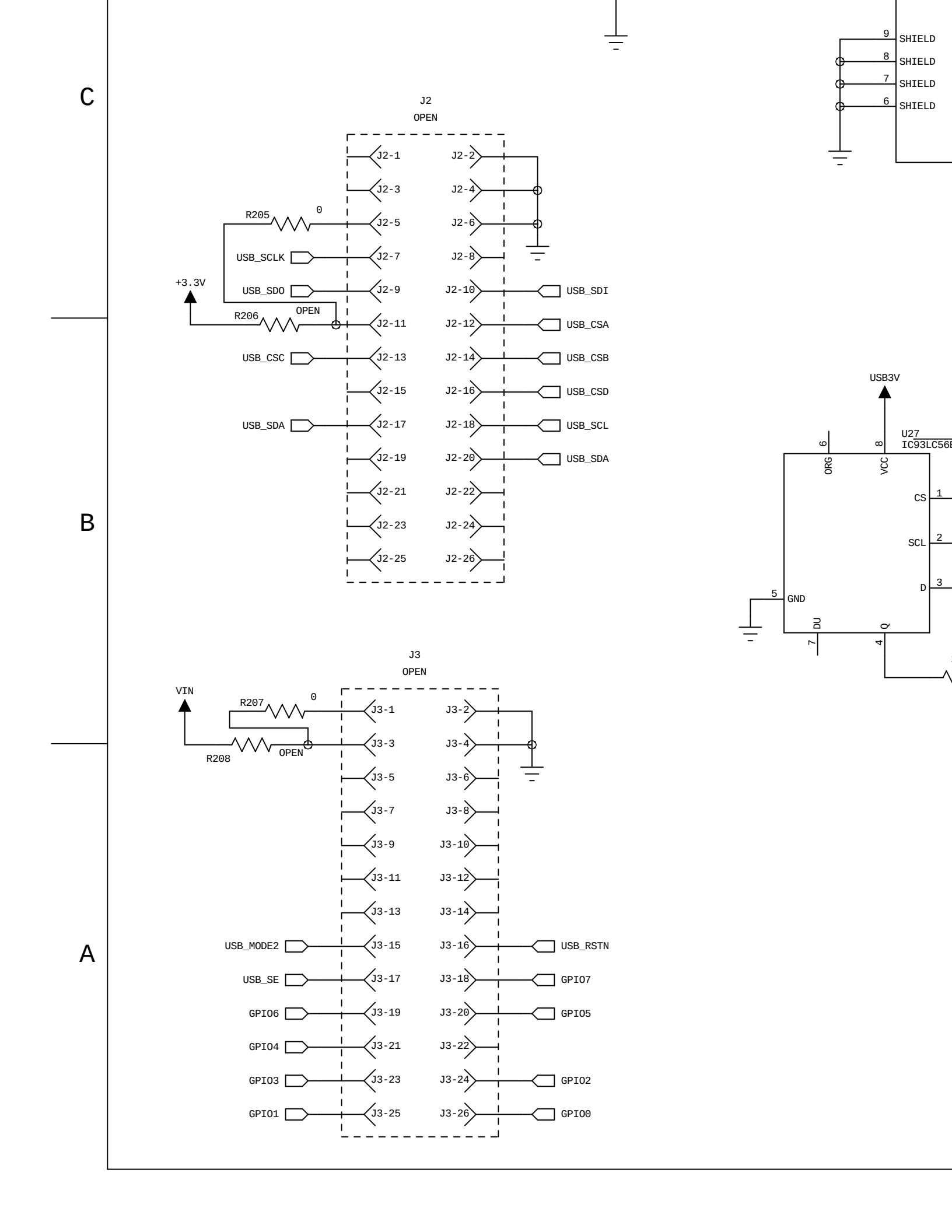

AVDD33_IN
VIN
ANALOG33
C165
4.7uF
U13
MAX1793
IN
OUT
SHDN
RST
SET
GND
EP
N.C.
R218
133k
1%
C167
6.8uF
JU2
R219
80.6k
1%

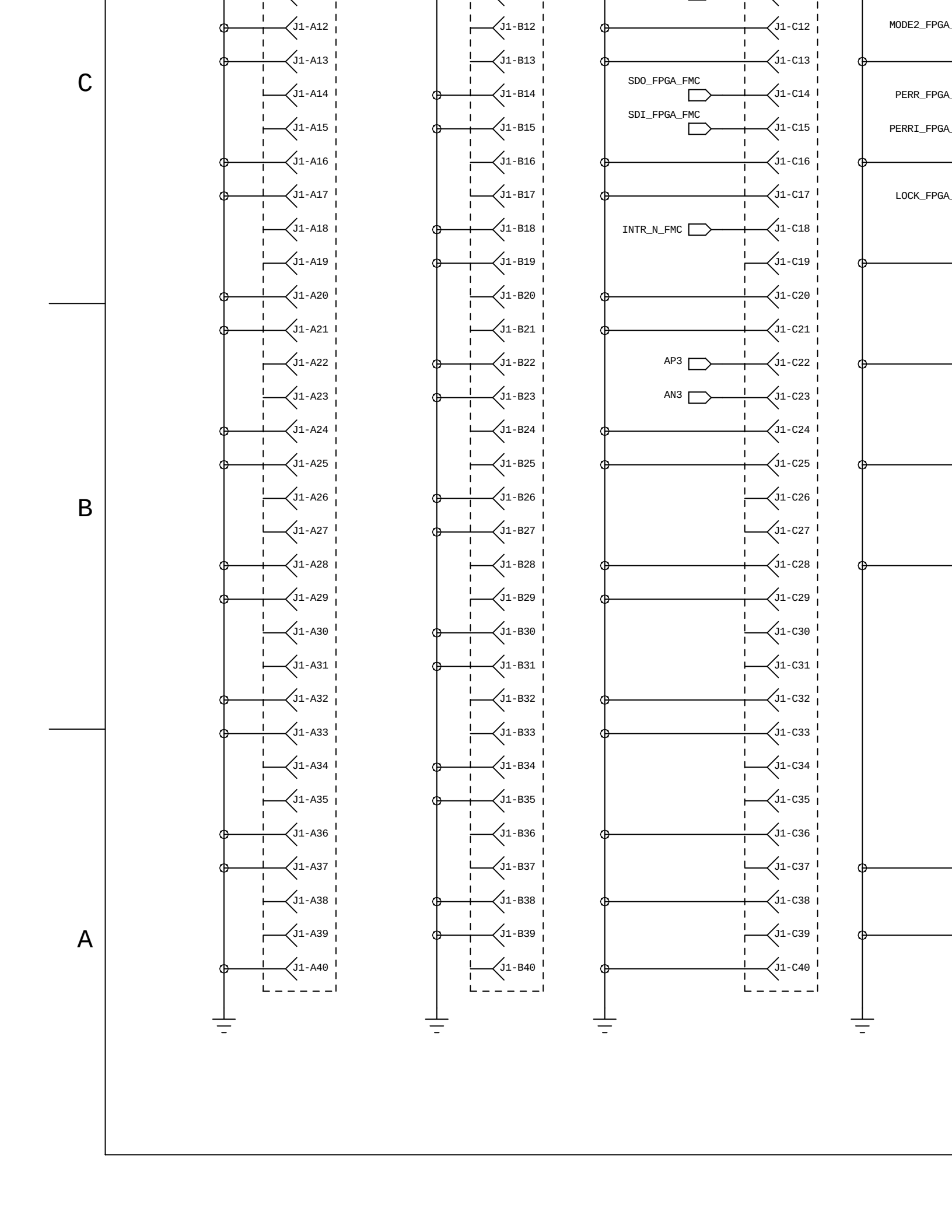

C
B
A
C7
0.22uF
+3.3V
C6
0.1uF
U5
MAX811
VCC
RST
MR
GND
R41
1k
USB_RSTN
RSTN
VDD18
C22
47uF
16V
C21
47uF
16V
R7
47
C66
0.1uF
U4_VCC
U4
MAX6654
N.C.
VCC
STBY
ALERT
SMBCLK
SMBDATA
DXP
DXN
GND
ADD0
ADD1
R24
499
1%
D1
USB_SCL
USB_SDA
TDA
TDC
C67
2200pF
R142
10k
R141
10k
11 = address 1001 110

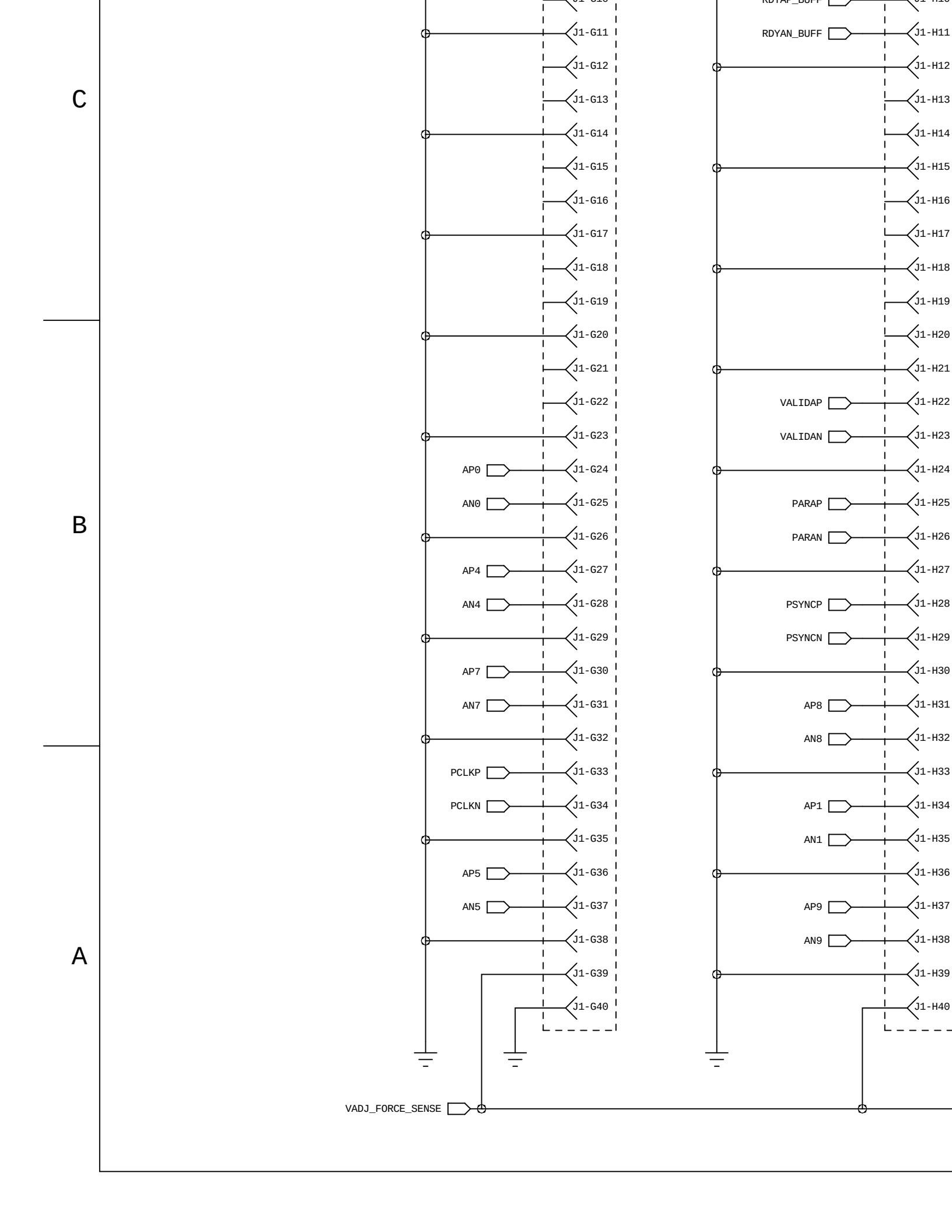

VDD18
C2
0.1uF
+3.3V
C1
0.1uF
USB_SCLK
USB_SDI
USB_SD
USB_CS
R144
10k
R145
10k
+3.3V
+3.3V

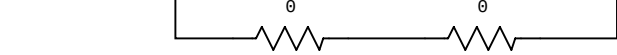

LEV_TRANS
PERR_
PERRI_
LOCK_


VDD18
+3.3V
C3
0.1uF
C4
0.1uF
SE_33
MODE_2_LV
RST

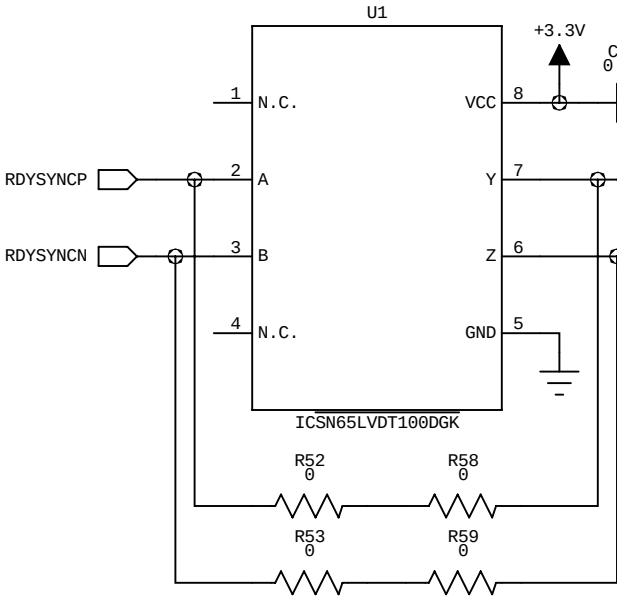

C
B
A
J2
OPEN
J2-1
J2-2
J2-3
J2-4
R205
0
J2-5
J2-6
USB_SCLK
J2-7
J2-8
+3.3V
USB_SDO
J2-9
J2-10
USB_SDI
R206
OPEN
J2-11
J2-12
USB_CSA
USB_CSC
J2-13
J2-14
USB_CSB
J2-15
J2-16
USB_CSD
USB_SDA
J2-17
J2-18
USB_SCL
J2-19
J2-20
USB_SDA
J2-21
J2-22
J2-23
J2-24
J2-25
J2-26
9
SHIELD
8
SHIELD
7
SHIELD
6
SHIELD
USB3V
U27
IC93LC56B
6
ORG
8
VCC
CS
1
SCL
2
D
3
5
GND
7
DU
Q
4
J3
OPEN
VIN
R207
0
J3-1
J3-2
R208
OPEN
J3-3
J3-4
J3-5
J3-6
J3-7
J3-8
J3-9
J3-10
J3-11
J3-12
J3-13
J3-14
USB_MODE2
J3-15
J3-16
USB_RSTN
USB_SE
J3-17
J3-18
GPIO7
GPIO6
J3-19
J3-20
GPIO5
GPIO4
J3-21
J3-22
GPIO3
J3-23
J3-24
GPIO2
GPIO1
J3-25
J3-26
GPIO0


C
B
A
J1-A12
J1-A13
J1-A14
J1-A15
J1-A16
J1-A17
J1-A18
J1-A19
J1-A20
J1-A21
J1-A22
J1-A23
J1-A24
J1-A25
J1-A26
J1-A27
J1-A28
J1-A29
J1-A30
J1-A31
J1-A32
J1-A33
J1-A34
J1-A35
J1-A36
J1-A37
J1-A38
J1-A39
J1-A40
J1-B12
J1-B13
J1-B14
J1-B15
J1-B16
J1-B17
J1-B18
J1-B19
J1-B20
J1-B21
J1-B22
J1-B23
J1-B24
J1-B25
J1-B26
J1-B27
J1-B28
J1-B29
J1-B30
J1-B31
J1-B32
J1-B33
J1-B34
J1-B35
J1-B36
J1-B37
J1-B38
J1-B39
J1-B40
J1-C12
J1-C13
SDO_FPGA_FMC
J1-C14
SDI_FPGA_FMC
J1-C15
J1-C16
J1-C17
INTR_N_FMC
J1-C18
J1-C19
J1-C20
J1-C21
AP3
J1-C22
AN3
J1-C23
J1-C24
J1-C25
J1-C26
J1-C27
J1-C28
J1-C29
J1-C30
J1-C31
J1-C32
J1-C33
J1-C34
J1-C35
J1-C36
J1-C37
J1-C38
J1-C39
J1-C40
MODE2_FPGA_
PERR_FPGA_
PERRI_FPGA_
LOCK_FPGA_

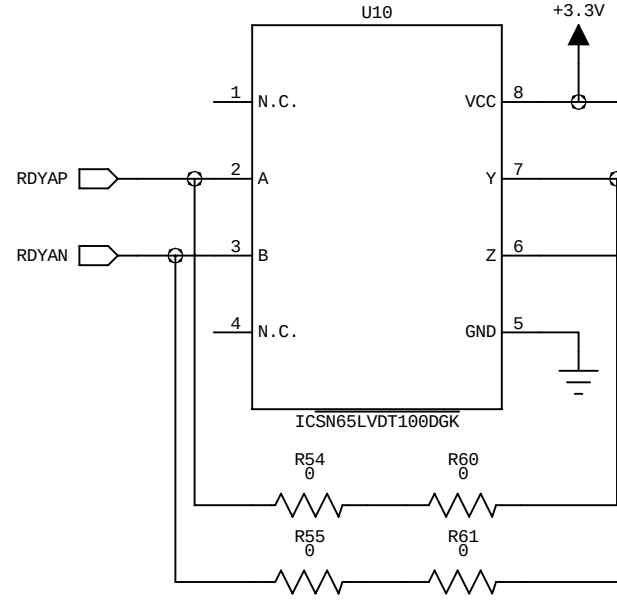

C
B
A
J1-G11
J1-G12
J1-G13
J1-G14
J1-G15
J1-G16
J1-G17
J1-G18
J1-G19
J1-G20
J1-G21
J1-G22
J1-G23
AP0
J1-G24
AN0
J1-G25
J1-G26
AP4
J1-G27
AN4
J1-G28
J1-G29
AP7
J1-G30
AN7
J1-G31
J1-G32
PCLKP
J1-G33
PCLKN
J1-G34
J1-G35
AP5
J1-G36
AN5
J1-G37
J1-G38
J1-G39
J1-G40
RDYAN_BUFF
J1-H11
J1-H12
J1-H13
J1-H14
J1-H15
J1-H16
J1-H17
J1-H18
J1-H19
J1-H20
J1-H21
VALIDAP
J1-H22
VALIDAN
J1-H23
J1-H24
PARAP
J1-H25
PARAN
J1-H26
J1-H27
PSYNCP
J1-H28
PSYNCN
J1-H29
J1-H30
AP8
J1-H31
AN8
J1-H32
J1-H33
AP1
J1-H34
AN1
J1-H35
J1-H36
AP9
J1-H37
AN9
J1-H38
J1-H39
J1-H40
VADJ_FORCE_SENSE



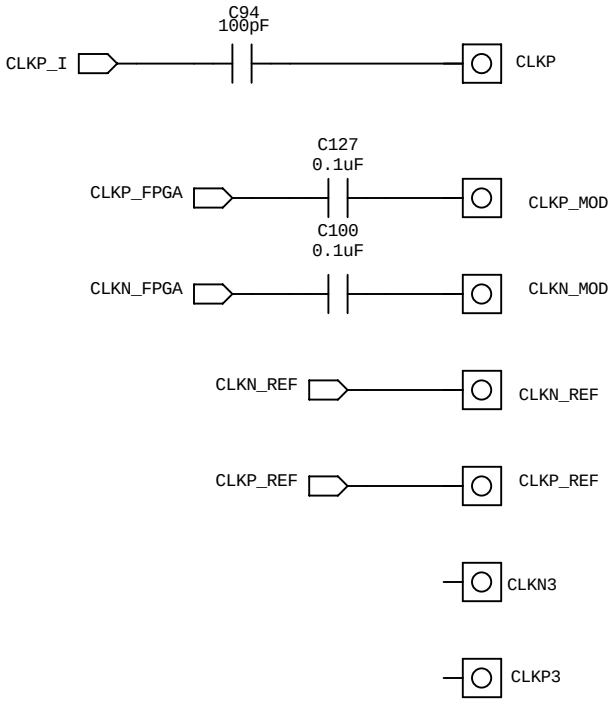

ALL BYPASS RESISTORS ARE LOCATED UNDER THE BUFFER
AND SHARE COMMON PINS FROM INPUT TO OUTPUT

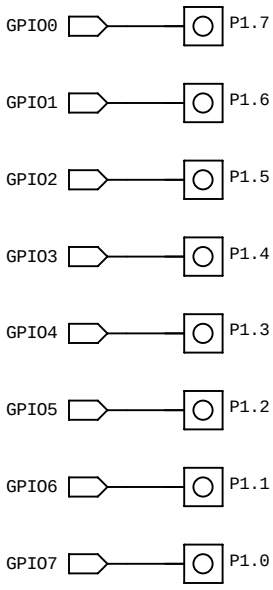



ALL BYPASS RESISTORS ARE LOCATED UNDER THE BUFFER
AND SHARE COMMON PINS FROM INPUT TO OUTPUT

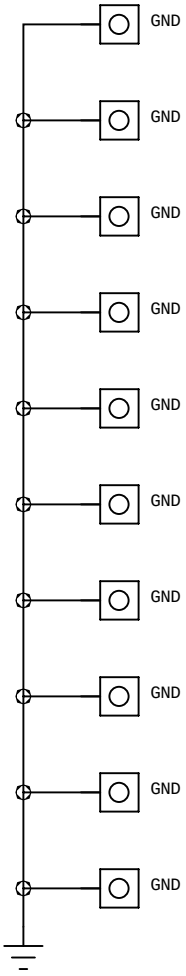



ALL BYPASS RESISTORS ARE LOCATED UNDER THE BUFFER FO
AND SHARE COMMON PINS FROM INPUT TO OUTPUT


C94
100pF
CLKP_I
CLKP
C127
0.1uF
CLKP_FPGA
CLKP_MOD
C100
0.1uF
CLKN_FPGA
CLKN_MOD
CLKN_REF
CLKN_REF
CLKP_REF
CLKP_REF
CLKN3
CLKP3


GPIO0
P1.7
GPIO1
P1.6
GPIO2
P1.5
GPIO3
P1.4
GPIO4
P1.3
GPIO5
P1.2
GPIO6
P1.1
GPIO7
P1.0


GPIO0
GPIO1
GPIO2
GPIO3
GPIO4
GPIO5
GPIO6
GPIO7

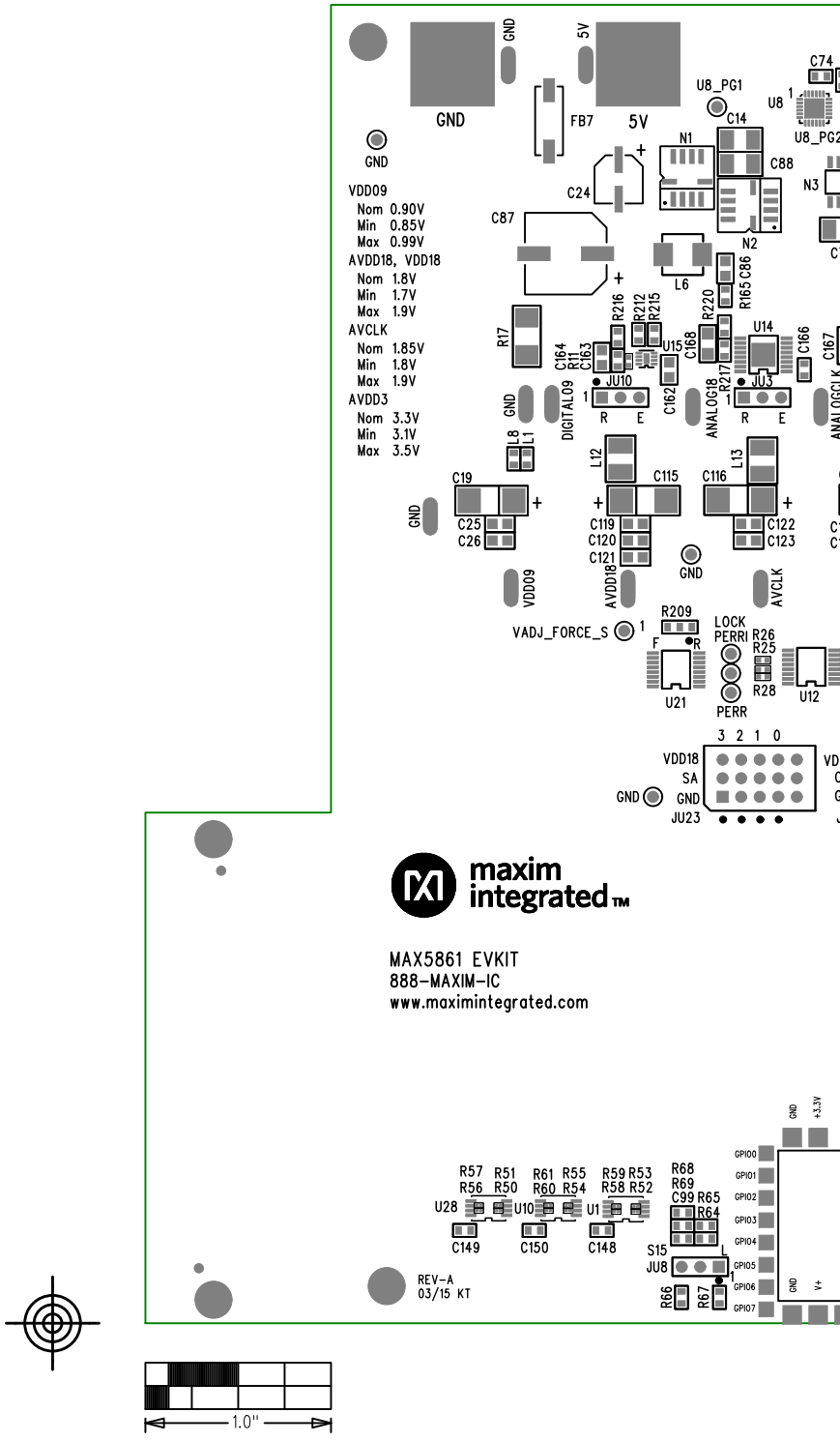

GND
GND
GND
GND
GND
GND
GND
GND
GND
GND

MAX5861 E

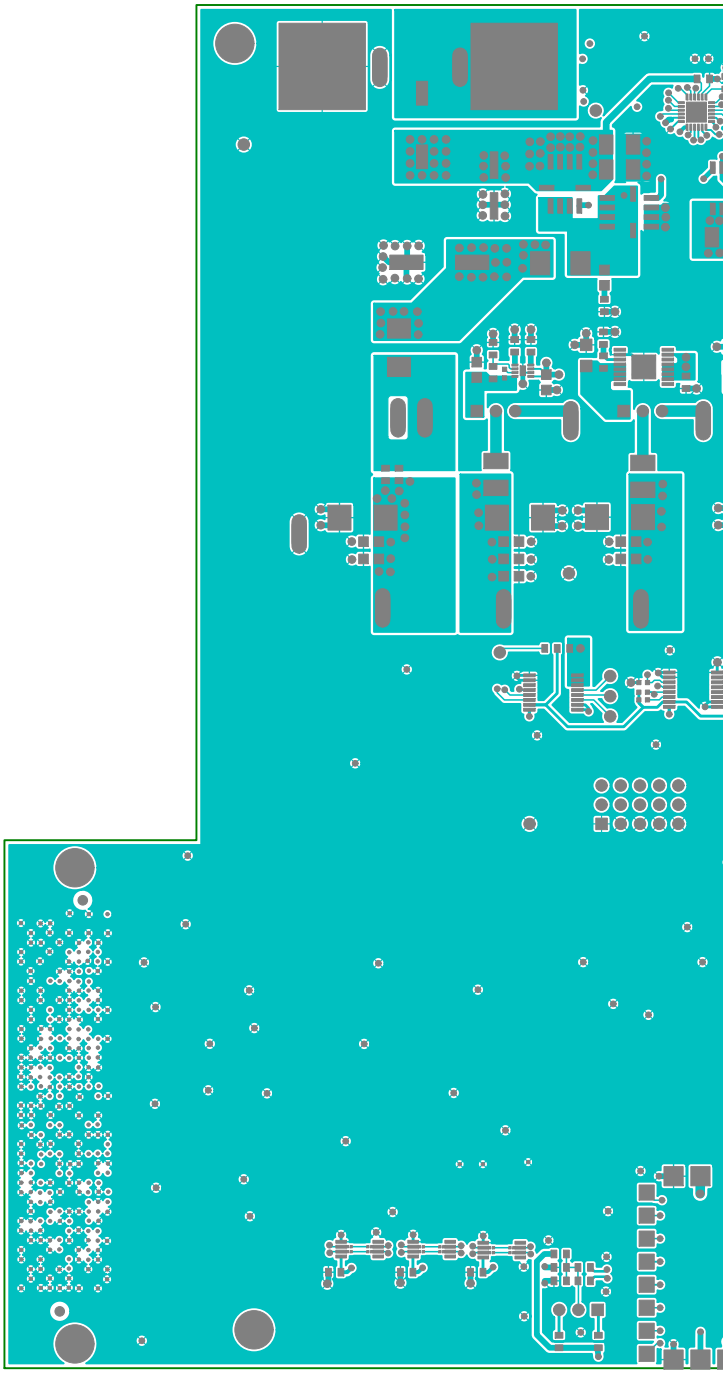
P/N:

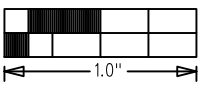
LAYER SILKSCREEN T

DATE:

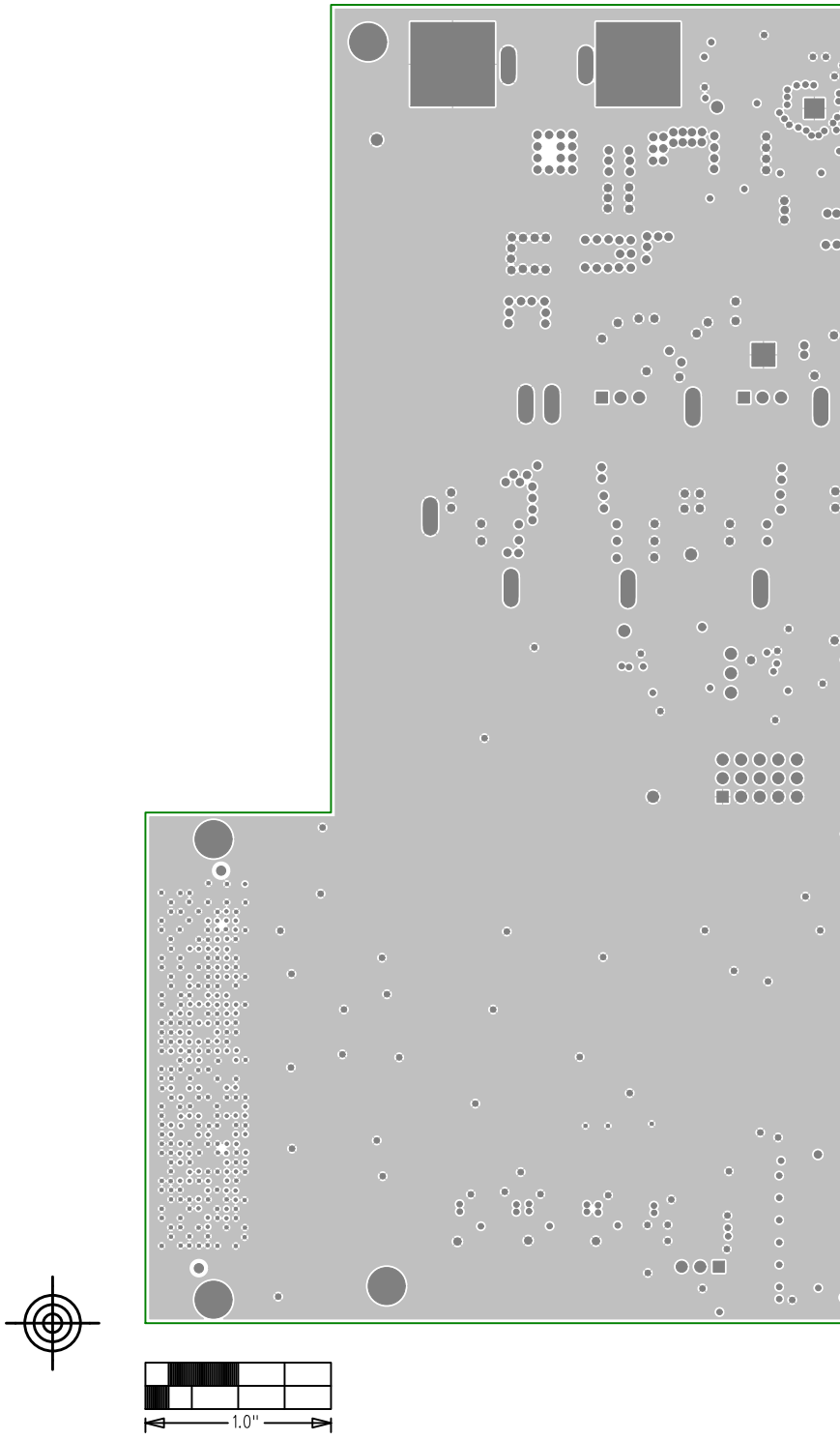


| MAX5861 E |
| --- |
| P/N: |
| |
| LAYER COMPONENT S |
| DATE: |

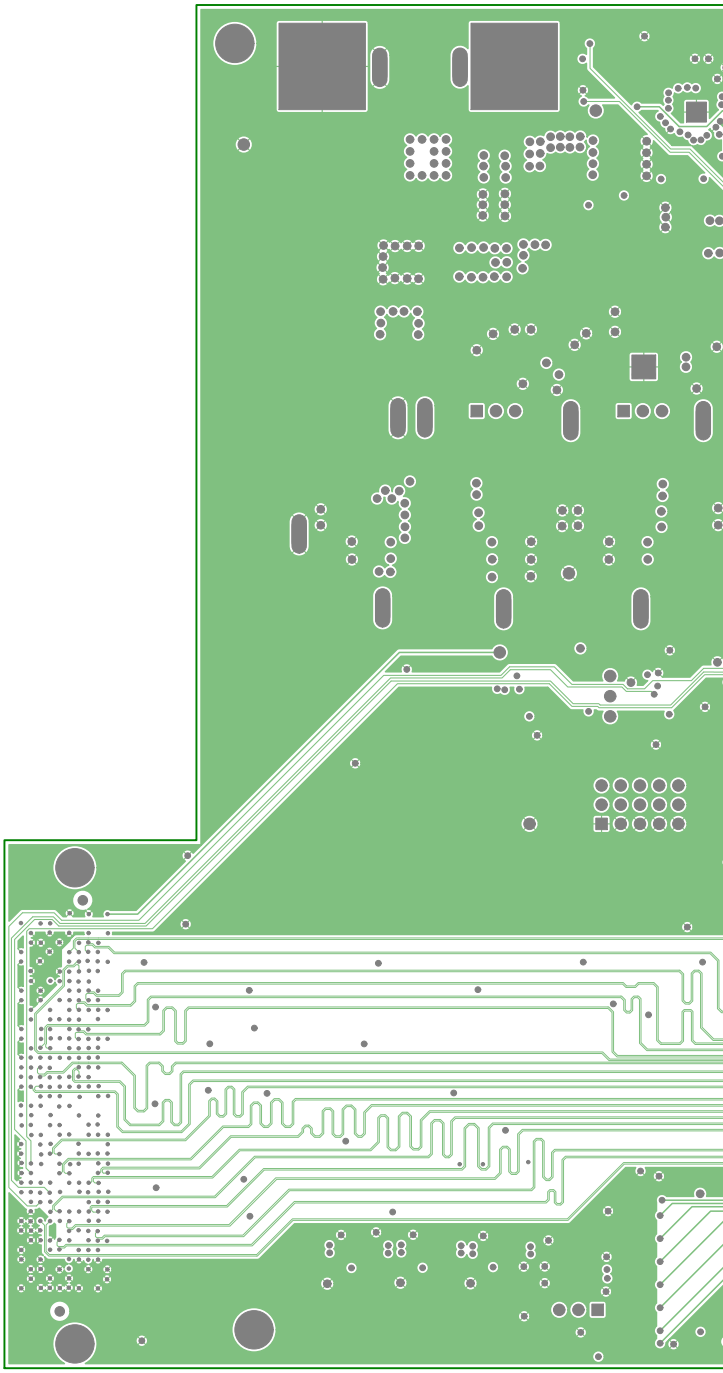




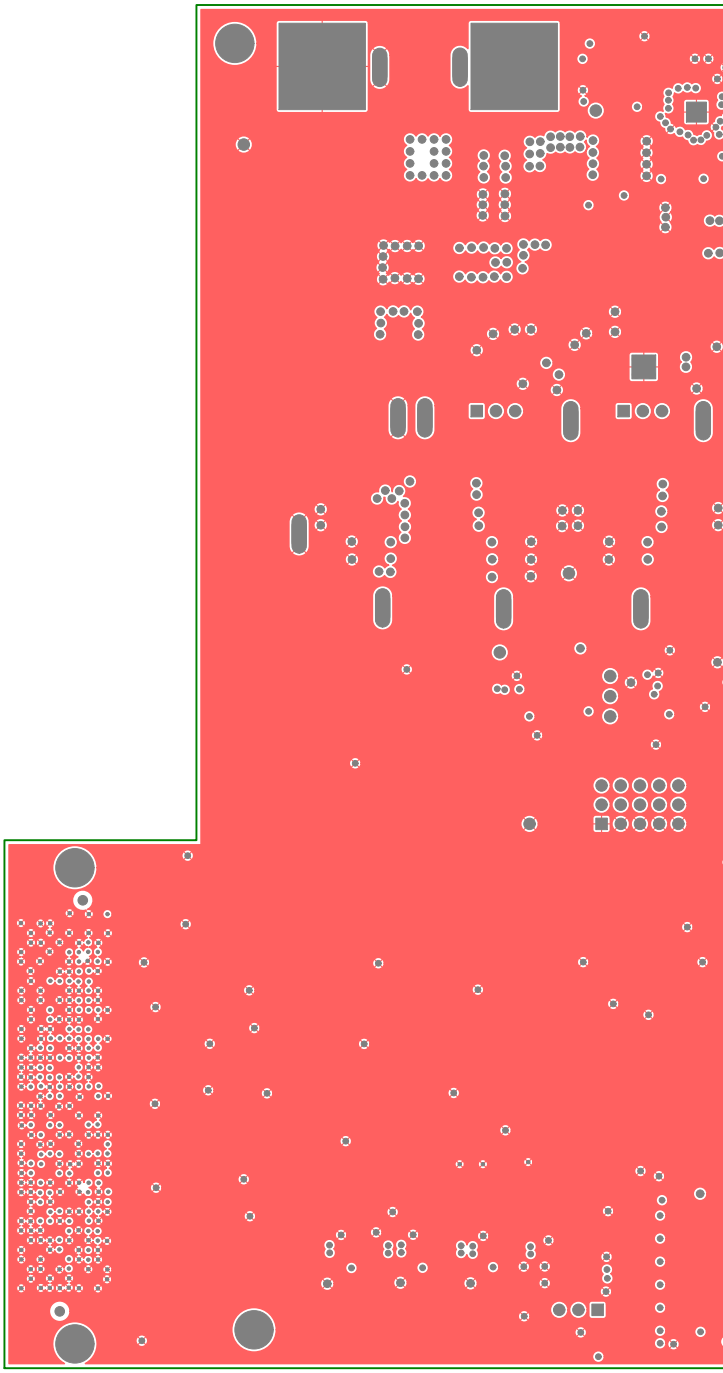
| MAX5861 E |
| --- |
| P/N: |
| |
| LAYER LAYER 2 GND |
| DATE: |



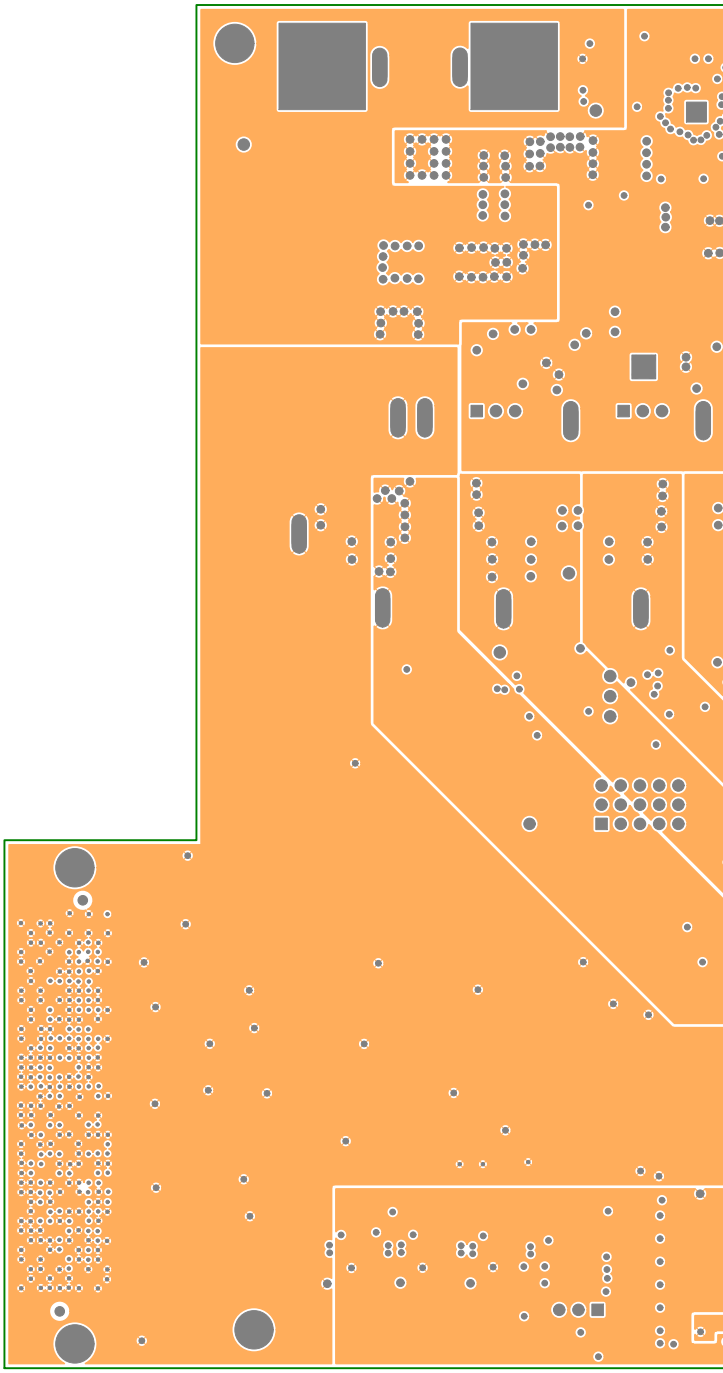
| | |
|---|---|
| MAX5861 E | |
| P/N: | |
| LAYER | LAYER 3 SIGN |
| DATE: | |



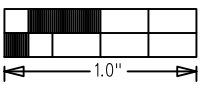
| | |
|---|---|
| MAX5861 E | |
| P/N: | |
| | |
| LAYER | LAYER 4 GND |
| DATE: | |



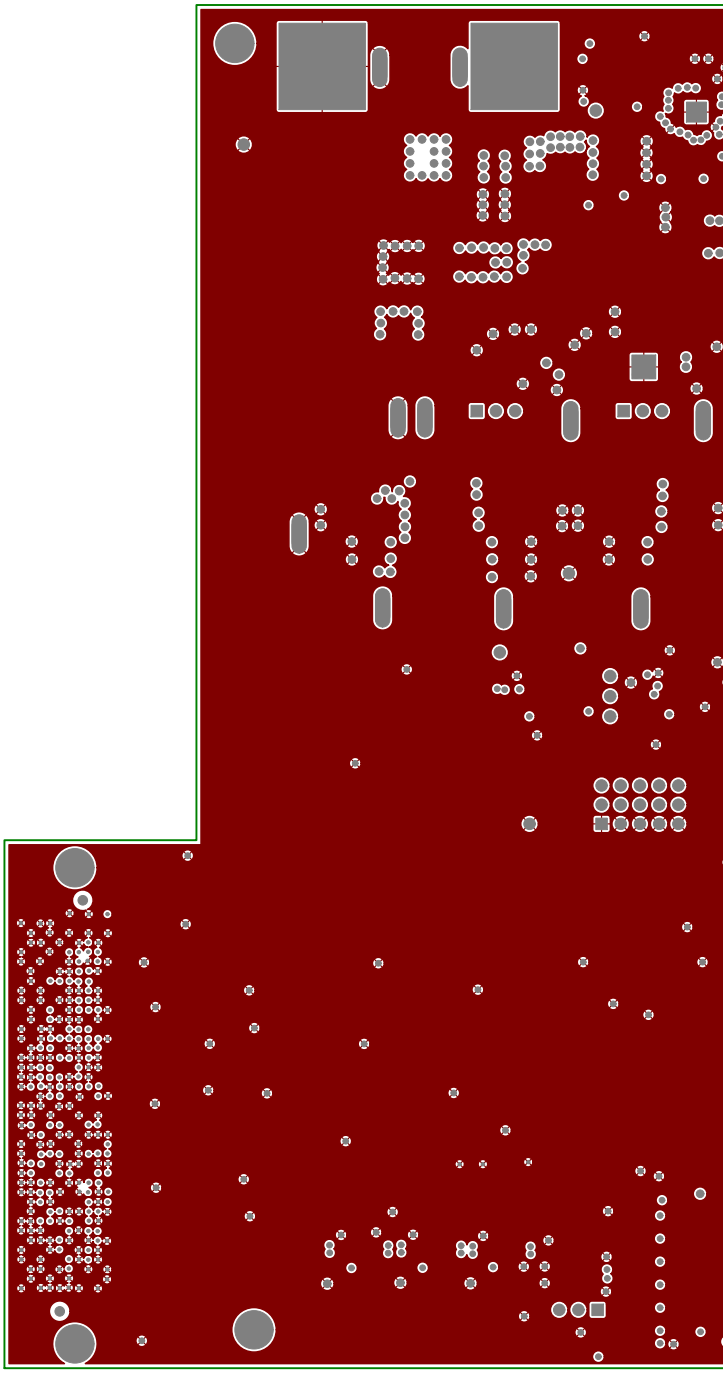
| | |
|---|---|
| MAX5861 E | |
| P/N: | |
| | |
| LAYER | LAYER 5 POW |
| DATE: | |



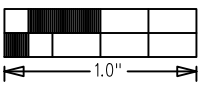




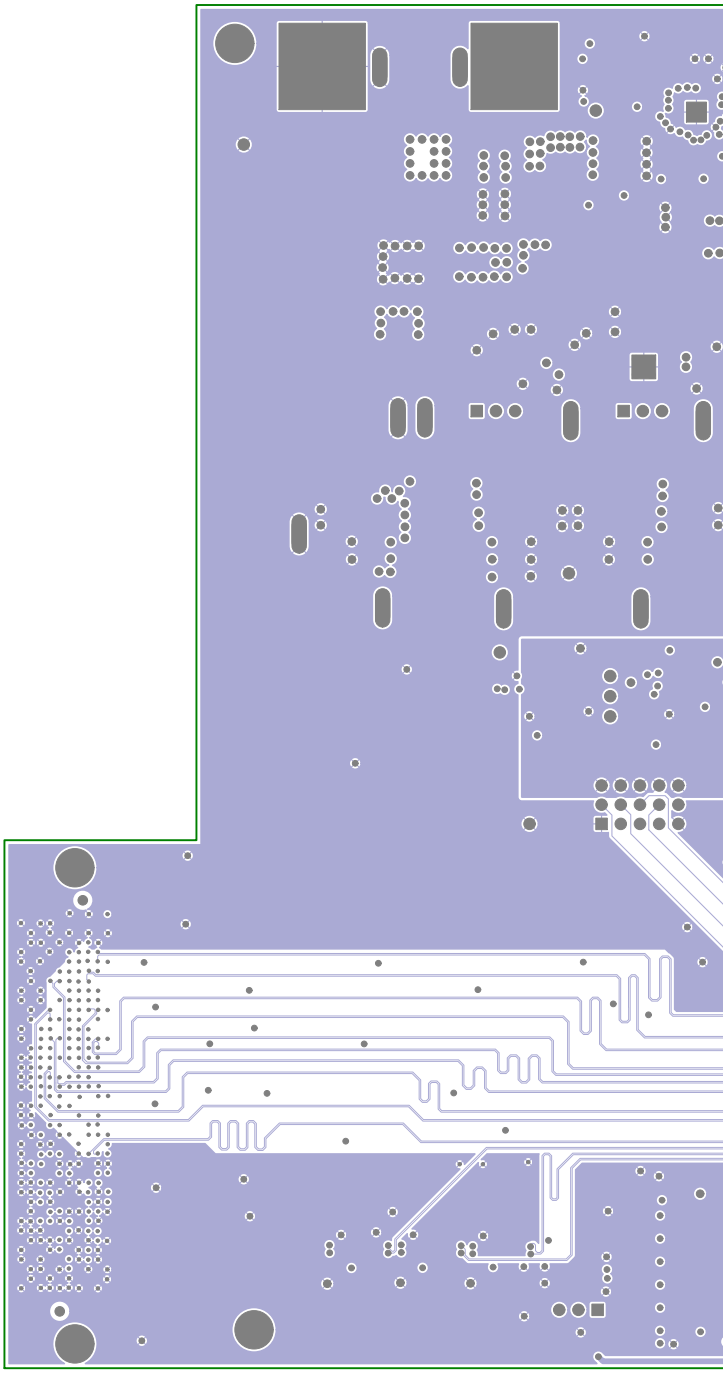
| | |
|---|---|
| MAX5861 E | |
| P/N: | |
| | |
| LAYER | LAYER 6 GND |
| DATE: | |



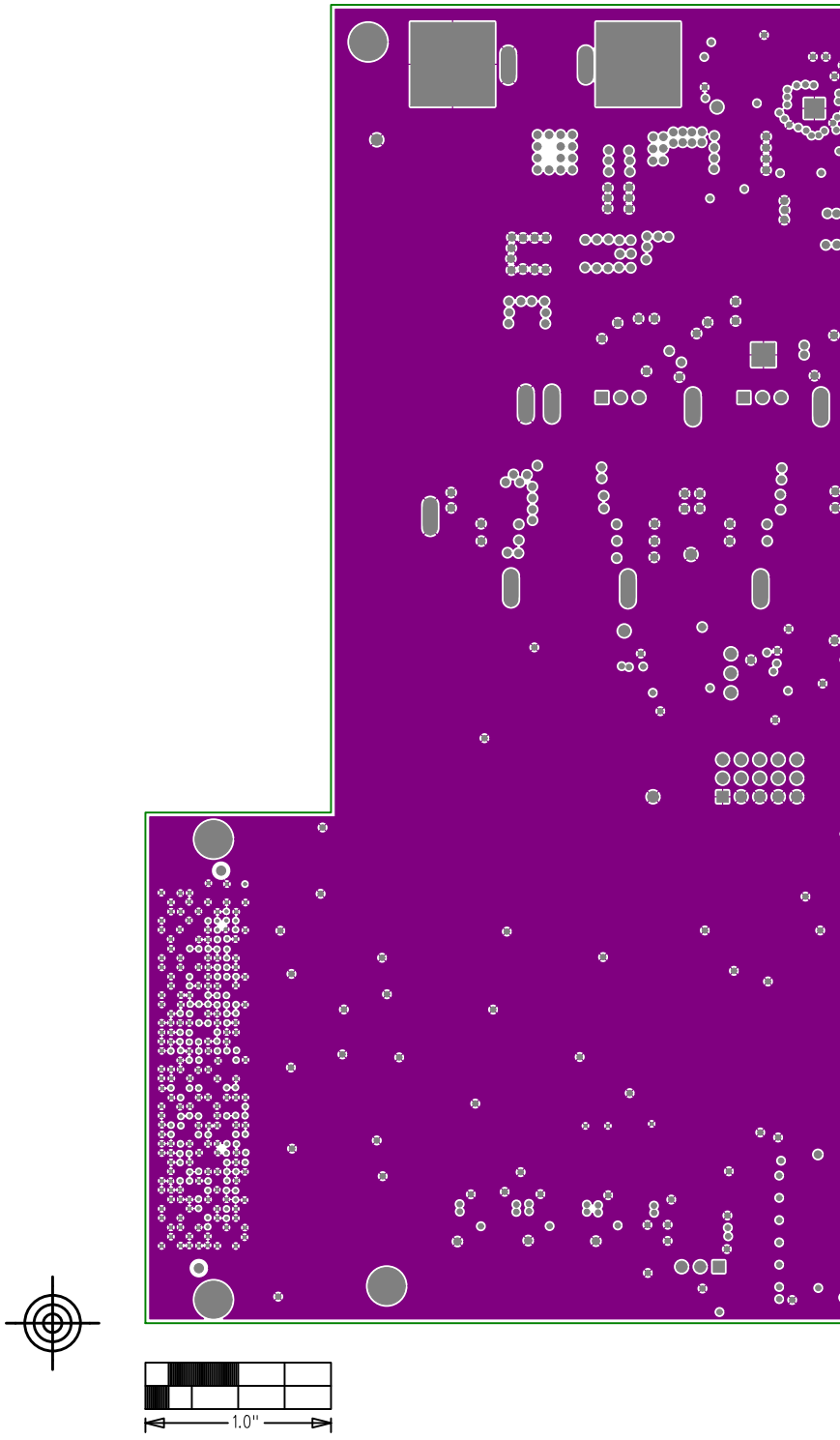




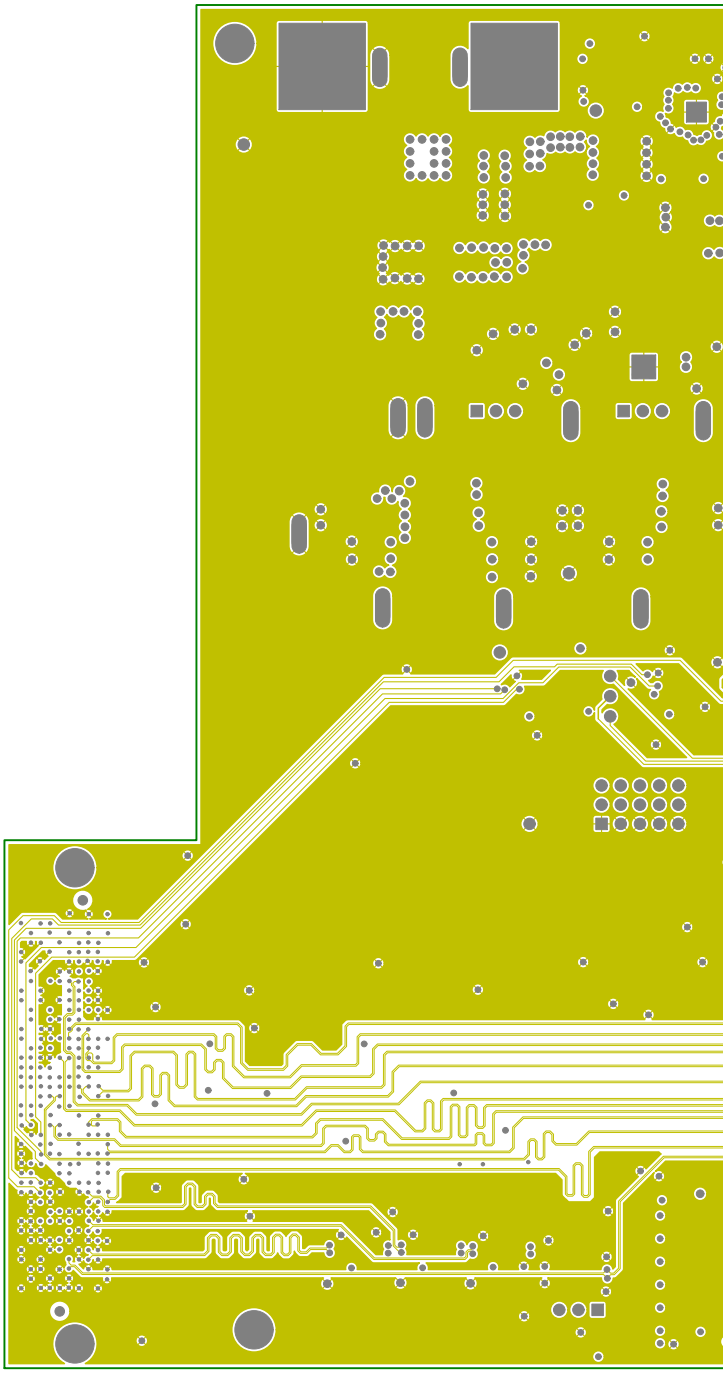
| | |
|---|---|
| MAX5861 E | |
| P/N: | |
| | |
| LAYER | LAYER 7 SIG |
| DATE: | |



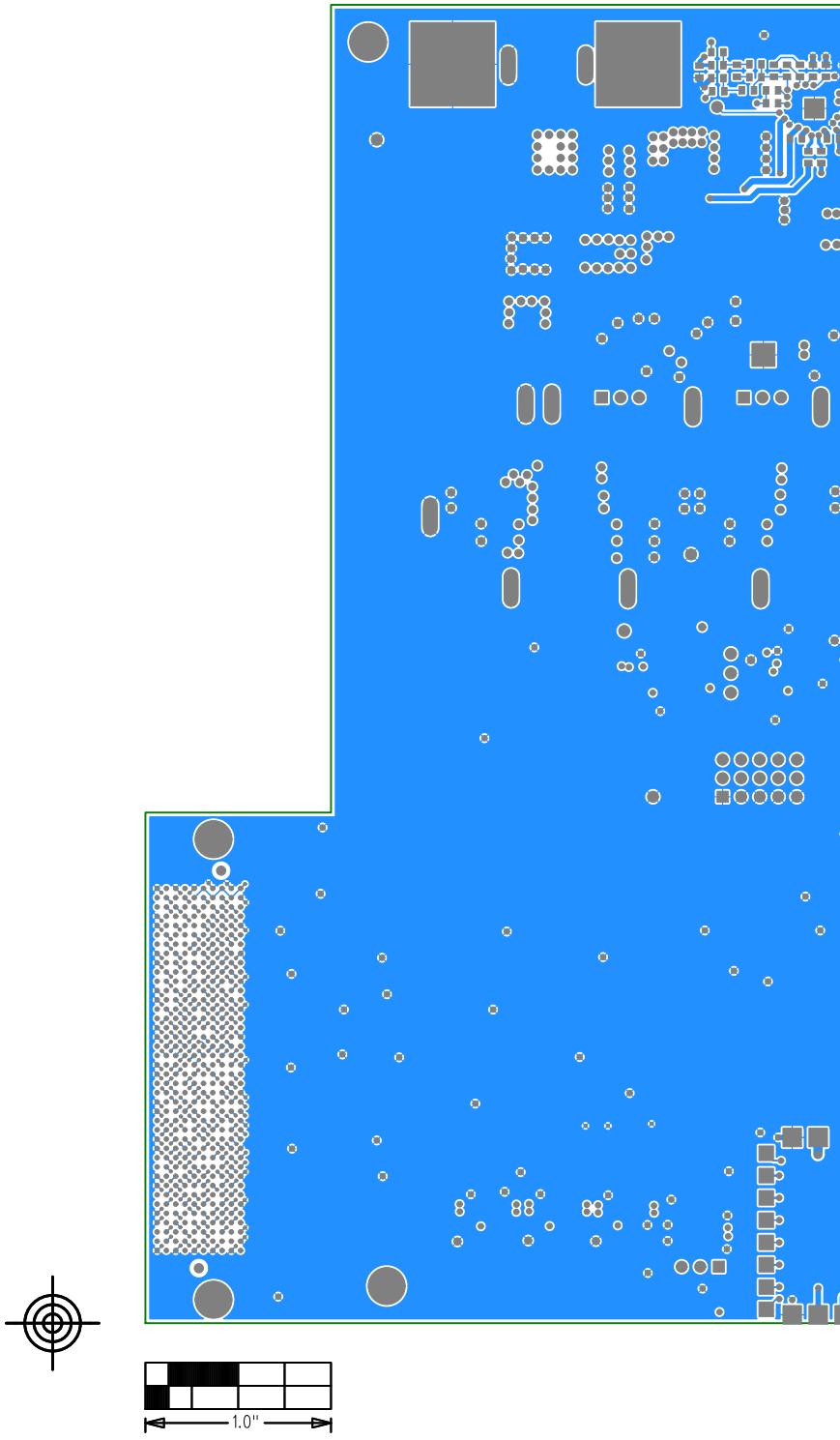
| MAX5861 E |
| --- |
| P/N: |
| |
| LAYER LAYER 8 |
| DATE: |



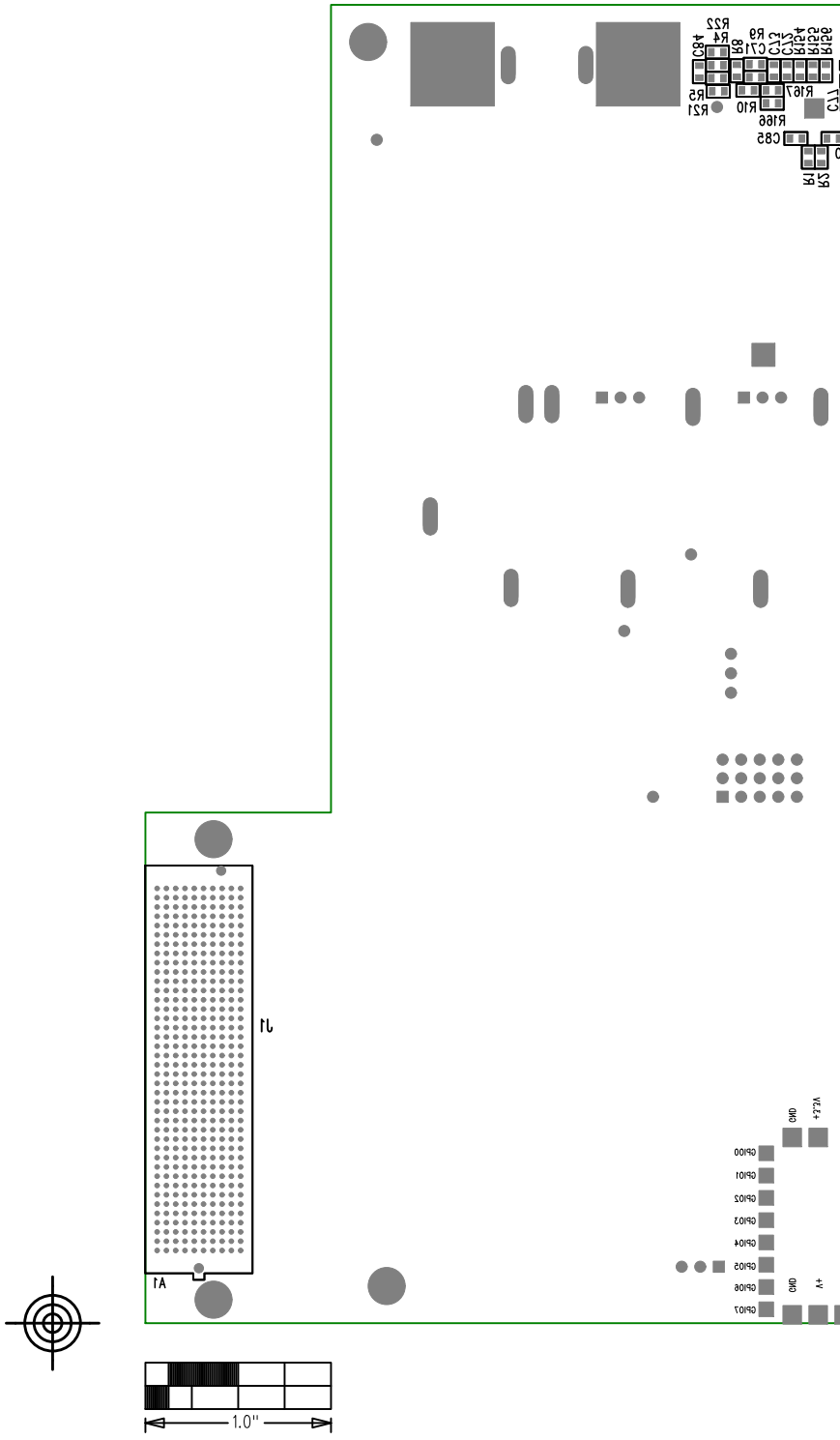
| | |
|---|---|
| MAX5861 E | |
| P/N: | |
| LAYER | LAYER 9 SIG |
| DATE: | |



| | |
|---|---|
| MAX5861 E | |
| P/N: | |
| LAYER | SOLDER SIDE |
| DATE: | |



| | |
|---|---|
| MAX5861 E | |
| P/N: | |
| | |
| LAYER | SILKSCREEN E |
| DATE: | |



**Bill of Materials (BOM)**
**(Rev 0, 6/15)**

| Item | Component Description | QTY Per | Reference Designators | Manufacturer Part Number |
|---|---|---|---|---|
| 1 | Uninsulated Banana Jacks | 2 | 5.0V_IN, GND | JOHNSON 108-0740-001 |
| 2 | PC Test point, red | 11 | TP25V, TP33V, LOCK, PERR, PERR, INTR_N, USB1VB, TPUSB3V, TP5V, RST_N, TDA | Keystone Electronics 5000 |
| 3 | 0.1uF ±10% 16V X7R ceramic capacitors (0603) | 14 | C1, C2, C3, C4, C5, C8, C66, C82, C84, C96, C99, C148, C149, C150 | TDK C1608X7R1C104K |
| 4 | 10uF ±20% 10V X5R ceramic capacitors (1210) | 3 | C14, C79, C89 | TDK C3225X5R1A106M |
| 5 | 47uF ±20% 16V tantalum capacitors (C-Case) | 7 | C19, C20, C21, C22, C114, C115, C116 | AVX TPSC476M016R0250 |
| 6 | 150uF ±20% 10V celectrolytic capacitor (6.6mmx 6.6mm) | 1 | C24 | Panasonic EEEFK1A151AP |
| 7 | 10uF ±20% 6.3V X5R ceramic capacitors (0805) | 18 | C25, C26, C28, C29, C50, C51, C52, C117, C118, C119, C120, C121, C122, C123, C162, C164, C202, C213 | TDK C2012X5R0J106M |
| 8 | 100pF ±5% 50V C0G ceramic capacitors (0402) | 2 | C8, C94 | TDK C1005C0G1H101J |
| 9 | 1.0uF ±20% 6.3V X5R ceramic capacitors (0402) | 18 | C35, C36, C39, C40, C41, C42, C43, C105, C106, C107, C108, C124, C125, C126, C128, C129, C130, C131 | TDK C1005X5R0J105M |
| 10 | 0.1uF ±20% 6.3V X5R ceramic capacitors (0201) | 45 | C46, C47, C48, C49, C50, C51, C100, C102, C127, C132, C133, C134, C135, C136, C137, C138, C139, C140, C141, C142, C143, C144, C145, C146, C147, C153, C154, C155, C156, C157, C158, C159, C160, C175, C176, C177, C178, C179, C205, C206, C207, C208, C209, C210, C211 | TDK C0603X5R0J104M |
| 11 | 0.22uF ±20% 6.3V X5R ceramic capacitors (0201) | 33 | C7, C12, C13, C15, C16, C17, C18, C23, C27, C33, C34, C37, C38, C44, C45, C52, C53, C54, C55, C56, C57, C58, C59, C60, C61, C62, C63, C64, C65, C68, C69, C70, C101 | Taiyo Yuden JMK063BJ224MP-F |
| 12 | 2200pF ±10% 50V C0G ceramic capacitor (0603) | 4 | C67, C71, C73, C76 | TDK C1608C0G1H222 |
| 13 | 33pF ±5% 50V C0G ceramic capacitors (0603) | 2 | C72, C77 | TDK C1608C0G1H330J |
| 14 | 1uF ±10% 10V X5R ceramic capacitor (0603) | 3 | C74, C201, C214 | TDK C1608X5R1A105K |
| 15 | 3300pF ±10% 50V C0G ceramic capacitors (0603) | 1 | C75 | TDK C1608C0G1H332K |
| 16 | 4.7uF ±20% 10V X5R ceramic capacitor (0603) | 7 | C78, C81, C95, C97, C165, C166, C203 | TDK C1608X5R1A475M |
| 17 | 0.22uF ±10% 10V X5R ceramic capacitor (0603) | 2 | C80, C85 | TDK C1608X5R1A224K |
| 18 | 470uF ±20% 6.3V Elecrolytic capacitor (10.3mmx 10.3mm) | 2 | C83, C87 | SANYO 6SVP470M |
| 19 | 6.8uF ±10% 10V X5R ceramic capacitor (1206) | 4 | C90, C98, C167, C168 | TDK C3216X5R1C685M |
| 20 | 0.47uF ±20% 4V X7S ceramic capacitors (0204) | 2 | C92, C93 | Murata LLL153C70G474ME17E |
| 21 | 0.01uF ±10% 25V X5R ceramic capacitors (0402) | 1 | C163 | TDK C1005X5R1E103K |
| 22 | 0.1uF ±10% 16V X5R ceramic capacitors (0402) | 4 | C103, C104, C112, C113 | TDK C1005X5R1A104K |
| 23 | | | | |
| 24 | 0.1pF ±10% 100V ceramic capacitors (0603) | 2 | C89, C204 | GRM188R72A104K |
| 25 | 3.3uF ±10% 10V ceramic capacitor (0603) | 1 | C11 | TDK C1608X5R1A335K |
| 26 | 8.0pF 50V NP0/C0G 0603 | 2 | C9, C10 | TDK C1608C0G1H080C |
| 27 | 6.8pF ±0.25pF 50V C0G ceramic capacitor (0402) | 1 | C466 | Murata GRM1555C1H6R8C |
| 28 | 0.92" SMA connectors (PC edge mount) | 1 | OUT | Rosenberger 32K243-40ML5 |
| 29 | LED Green (0603) | 1 | D1 | Panasonic LNJ314G8TRA |
| 30 | Momentary Pushbutton Switches | 3 | MODE2, SE, RSTN | Omron B3S-1000 |
| 31 | Small Black Test Points | 9 | GND (x8), TDC | Keystone Electronics 5001 |
| 32 | Power filter ferrite bead | 1 | FB1 | StewartLaird Signal Integ Prod 28F0121-1SR-10 (Digikey 240-2436-1-ND) |
| 33 | FMC Connector | 1 | J1 | Samtech ASP-134488-01 |
| 34 | 2 Pin Header(OK to substitute with manufactured header, do not cut to fit) | 4 | JU4, JU12, JU24, JU25 | Sullins Electronics Corp. S1012EC-02-ND |
| 35 | 3x5 Header(OK to substitute with manufactured header, do not cut to fit) | 3 | JU5, JU7, JU23 | TSW-105-15-T-T-ND |
| 36 | 3 pin headers | 8 | JU2, JU3, JU6, JU8, JU10, JU11, JU22, JU26 | TSW-103-07-T-S |
| 37 | Chip bead cores (0603) | 2 | L1, L8 | Panasonic EXC-ML16A270U |
| 38 | Chip bead cores (1812) | 4 | L2, L11, L12, L13 | Panasonic EXC-CL4532U1 |
| 39 | 10uH 3A Inductor | 1 | L5 | Sumida CDRH104RNP-100N |
| 40 | 4.7uH 5.5A Inductor | 1 | L6 | Vishay IHLP2020CZER4R7M11 |
| 41 | 390nH wire-wound chip inductors (2520) | 2 | L9, L10 | Coilcraft 1008CS-391XJLB |
| 42 | 1.3mH 1.2A Inductor (0402) | 1 | L17 | Murata LQW15AN1N3C10D |
| 43 | FERRITE BEAD 28 OHM 4.0A 0603 | 2 | L16, L3 | Wurth 742792603 |
| 44 | 30V 9A n-channel MOSFET ( 8 SO) | 2 | N1, N2 | Fairchild FDS6692A |
| 45 | Dual n-channel 30V MOSFET (8 SO) - | 1 | N3 | Fairchild Semiconductor FDS6982AS |
| 46 | 0 Ohm 5% resistors (0603) | 30 | R1, R2, R11, R15, R16, R18, R19, R20, R21, R22, R29, R30, R31, R32, R33, R34, R35, R36, R37, R38, R39, R40, R49, R164, R212, R202, R203, R204, R205, R207 | Venkel 0603-000T |
| 47 | 0 Ohm 5% resistors (0603) - Dual_Opt | 1 | R209 | Venkel 0603-000T |
| 48 | 0 Ohm 5% resistors (2010) | 1 | R17 | Venkel 2010-000T |
| 49 | 0 Ohm 1% resistors (0201) | 12 | R50, R51, R52, R53, R54, R55, R56, R57, R58, R59, R80, R81 | PanasonicERJ-1GE0R00C |
| 50 | 1k Ohm 1% resistor (0603) | 4 | R6, R12, R41, R42 | Venkel 0603-1001T |
| 51 | 143 Ohm 1% resistor (0603) | 2 | R8, R158 | Venkel 0603-1430T |
| 52 | 8.45k Ohm 1% resistor (0603) | 1 | R9 | Venkel 0603-8451T |
| 53 | 16.9k Ohm 1% resistor (0603) | 1 | R10 | Venkel 0603-1692T |
| 54 | 10k Ohm 1% resistor (0603) | 9 | R47, R64, R65, R75, R76, R77, R154, R161, R220 | Venkel 0603-1002T |
| 55 | 499 Ohm 1% resistor (0402) | 2 | R24, R214 | Venkel 0402-4990T |
| 56 | 47 Ohm 1% resistor (0603) | 1 | R7 | |
| 57 | 590 Ohm 1% resistor (0603) | 1 | R69 | |
| 58 | 866 Ohm 1% resistor (0603) | 1 | R67 | |
| 59 | 1.87k Ohm 1% resistor (0603) | 1 | R68 | Venkel 0603-103T |
| 60 | 4.7k Ohm 1% resistor (0603) | 1 | R217 | |
| 61 | 5.23k Ohm 1% resistor (0603) | 1 | R66 | Venkel 0603-2372T |
| 62 | 8.06k Ohm 1% resistor (0603) | 1 | R156 | Venkel 0603-8061T |
| 63 | 33.2k Ohm 1% resistor (0603) | 2 | R155, R157 | Venkel 0603-3322T |
| 64 | 9.53k Ohm 1% resistor (0603) | 1 | R159 | Venkel 0603-9531T |
| 65 | 3.01k Ohm 1% resistor (0603) | 1 | R160 | Venkel 0603-3011T |
| 66 | 10 Ohm 5% resistor (0603) | 4 | R4, R5, R166, R167 | Venkel 0603-100T |
| 67 | 2.2k Ohm 5% resistor (0603) | 1 | R23 | Venkel 0603-222T |
| 68 | 2.0k Ohm ±1% resistors (0603) | 1 | R213 | Venkel 0603-2001T |
| 69 | 10.0k Ohm 5% resistor (0402) | 19 | R13, R14 , R25, R26, R27, R28, R43, R44, R45, R46, R141, R142, R143, R144, R145, R146, R147, R200, R201 | |
| 70 | 12.0 kOhm 1% resistor (0603) | 1 | R3 | |
| 71 | 133k Ohm ±1% resistors (0603) | 2 | R62, R218 | Venkel 0603-1333T |
| 72 | 80.6k Ohm ±1% resistors (0603) | 2 | R63, R219 | Venkel 0603-8062T |
| 73 | 1:1 3000MHz RF transformers | 2 | T5, T6 | Mini-Circuits TC1-1-13M-34+ |
| 74 | Digital Up-converter and RF DAC (280 LFBGA) | 1 | U15 | Maxim MAX5881Uxx+ |
| 75 | Quad Level Translator (16-TSSOP) | 5 | U2, U3, U12, U17, U21 | Texas Instrument SN74AVC4T774IPW |
| 76 | Temperature sensor (16-QSOP) | 1 | U4 | Maxim MAX6654MEE+T |
| 77 | 3.3V Supervisory Circuit (SOT-143) | 1 | U5 | Maxim MAX811TEUS+ |
| 78 | 3.3V Supervisory Circuit (SOT-143) | 2 | U6, U16 | Maxim MAX812TEUS+ |
| 79 | Dual DC-DC Controller (24 TQFN-EP) | 1 | U8 | Maxim MAX15023ETG+ |
| 80 | 1.8V LDO Regulator (16 TSSOP-EP) | 2 | U14,U9 | Maxim MAX1793EUE18+ |
| 81 | 1.25V precision voltage reference (8 SO) | 1 | U11 | Maxim MAX6161AESA+ |
| 82 | | | | |
| 83 | EEPROM 128X16 | 1 | U27 | Microchip 93LC56B-E/MS |
| 84 | 3.3VLDO Regulator (16 TSSOP-EP) | 2 | U13, U204 | Maxim MAX1793EUE33+ |
| 85 | LDO Regulator (8 TDFN-EP) | 1 | U15 | Maxim MAX8902AATA+ |
| 86 | | | | |
| 87 | UART to USB Converter(TQFP-32L 7mmx7mm) | 1 | U7 | FTDI: FT4232HQ-QFN |
| 88 | 2.5V LDO Regulator (5 SC70) | 1 | U205 | Maxim MAX8511EXK25+ Top Mark |
| 89 | 3.3V LDO Regulator (5 SC70) | 1 | U206 | Maxim MAX8511EXK33+ Top Mark |
| 90 | Mini USB Receptacle | 1 | USB23 | 575-89743005 |
| 91 | 12MHz crystal | 1 | Y1 | ECS-120-18-28A CSM-4AX |
| 92 | Shunts | 22 | | Sullins Electronics Corp. STC02SYAN |
| 93 | PCB: MAX5881 EVALUATION KIT | 1 | 1 Oz Impedance Controlled | Network PCB |
| 94 | PCB: MAX5881 EVALUATION KIT | 1 | 1 Oz Impedance Controlled | Cirtex |
| 95 | PCB: MAX5881 EVALUATION KIT | 1 | 1 Oz Impedance Controlled | MEI |
| 96 | | | | |
| 97 | Not installed capacitors (0805) | 2 | C81, C86 | |
| 98 | Not installed, capacitor (0603) | 3 | C109, C110, C111 | |
| 99 | Not installed resistor (0603) | 5 | R162, R163, R165, R206, R208 | |
| 100 | Not installed resistor (0603) | 3 | R215, R216, R48 | |
| 101 | Not installed, testpoints | 2 | U8_PG1, U8_PG2 | Keystone 5000 |
| 102 | Not installed, IC's | 3 | U1, U10, U28 | |
| 103 | Not installed, 2x13 header socket 0.1" okay to substitute (2.54mm pitch, >3.9mm ht) | 2 | J2, J3 | Sullins Connector Solutions SFH11-PBPC-D13-ST-BK |
| 104 | Not installed, 0.92" SMA connectors (PC edge mount) | 1 | REF_CLKOUT | Rosenberger 32K243-40ML5 |
| 105 | Not installed, 1:1 3000MHz RF transformers | 1 | T4 | Mini-Circuits TC1-1-13M+ |
| 106 | Not installed version of device, 1.25V precision voltage reference (8 SO) (MAX6161AESA+ installed) | 0 | U11 | Maxim MAX6161BESA+ |
| 107 | | | | |
| 108 | | | | |
| 109 | LARGE BROWN 9 3/8" x 71 /4" x 2 1/2"" | | Pack-out | |
| 110 | Label | | Pack-out | |
| 111 | WEB instructions for Maxim Data Sheet | | Pack-out | |
| 112 | BAG, STATIC SHIELD ZIP 8"x10", W/ ESD LOGO | | Pack-out | |
| 113 | FOAM, ANTI-STATIC PE 12"x12"X5MM | | Pack-out | |
| 114 | CABLE, USB-A male to USB-mini 6ft | 1 | | |
| 115 | 4-40 x 0.75" machine screws, pan | 4 | Item A | Grainger 1ZB43 (part of Hardware Kit) |
| 116 | flat washer (ID - 0.125", OD- 0.312") | 6 | Item B | Grainger 2WA22 (part of Hardware Kit) |
| 117 | 4-40 x 1 " aluminum hex standoff | 4 | Item C | Kyestone 2205 |
| 118 | Nylon flat washers (ID - 0.140", OD-0.375") | 4 | Item D | Keystone Electronics 3213 |
| 119 | Hardware Kit (containing Items A, B, C, D, E and F) | | Contains Items A-D | |
| 120 | BAG, STATIC SHIELD 5"X8",WIESD LOGO | | Hardware Kit-Contains Items A-D | |
| 121 | BAG, STATIC SHIELD ZIP 4"x6", W/ ESD LOGO | | Items A-D | |
| 122 | | | | |
| 123 | XFMR_CLK_Module (Installed) | 1 | Install on board | |
| 124 | | | JUMPER TABLE | |
| | | | JUMPER | SHUNT POSITION |
| | | | JU4 JU12 | Installed 1-2 |
| | | | JU7 JU23 | 1-2, 4-5, 7-8, 10-11 |
| | | | JU5 | 2-3, 4-5, 7-8, 10-11 |
| | | | JU2 JU3 JU8 JU10 JU26 | 1-2 |
| | | | JU6 JU11 JU22 | 2-3 |
| | | | R209 | 2-3 |

# X-ON Electronics

Largest Supplier of Electrical and Electronic Components

***Click to view similar products for*** RF Development Tools ***category:***

***Click to view products by*** Maxim ***manufacturer:***

Other Similar products are found below :

MAAM-011117 MAAP-015036-DIEEV2 EV1HMC1113LP5 EV1HMC6146BLC5A EV1HMC637ALP5 122410-HMC686LP4E ADL5363-EVALZ 130437-HMC1010LP4E EKIT01-HMC1197LP7F SKYA21001-EVB SMP1331-085-EVB EVAL01-HMC1041LC4 MAAL-011111-000SMB MAAM-009633-001SMB 107712-HMC369LP3 107780-HMC322ALP4 SP000416870 EV1HMC520ALC4 EV1HMC244AG16 EV1HMC539ALP3 124694-HMC742ALP5 SC20ASATEA-8GB-STD MAX2692EVKIT# SKY12343-364LF-EVB 108703-HMC452QS16G 119197-HMC658LP2 EV1HMC647ALP6 ADL5725-EVALZ 106815-HMC441LM1 UXN14M9PE SIMSA868-DKL SIMSA868C-DKL SKY65806-636EK1 SKY68020-11EK1 SKY67159-396EK1 SKY66181-11-EK1 SKY65804-696EK1 SKY13396-397LF-EVB SKY13380-350LF-EVB SKY13322-375LF-EVB SKY12207-478LF-EVB SE5023L-EK1 SE5004L-EK1 SE2436L-EK1 Se2435L-EK1 SIMSA915C-DKL SIMSA915-DKL SIMSA433C-DKL SKY12211-478LF-EVB EVK-R202-00B