Click here to ask about the production status of specific part numbers.

MAX4147X Evaluation Kit

Evaluates: MAX41470, MAX41473, and MAX41474

General Description

The MAX4147X evaluation kit (EV kit) contains a single MAX4147X high output power VHF/UHF sub-GHz ISM/SRD receiver designed to receive frequency-shift keying (FSK), Gaussian GFSK, or amplitude-shift keying (ASK) data in the 287MHz to 320MHz, 425MHz to 480MHz, and 860MHz to 960MHz frequency ranges.

The MAX41470, MAX41473, and MAX41474 EV kits operate in conjunction with an external microcontroller (MCU) and graphical user interface (GUI) software running on a computer. The MAX41470 uses an SPI interface for internal register configurations while the MAX41473/74 can use the preset modes or an I²C interface for register programming and control.

The MAX41473 and MAX41474 EV kits are also designed to operate with a simple, one-pin data interface, alleviating the need to program the part for nominal operation, or other high-level system (PC with GUI software) having to configure the receiver for operation. These parts allow the user to preset the operating frequencies by part selection and pin configurations. On the EV kit, selecting the frequency of operation is as simple as setting jumpers.

The EV kit includes Windows® 10-compatible software that provides a simple GUI for configuration of all the MAX4147X registers through the SPI or I²C ports. The GUI also controls the on-board PMIC when the MAX32630FTHR applications platform is used.

Features and Benefits

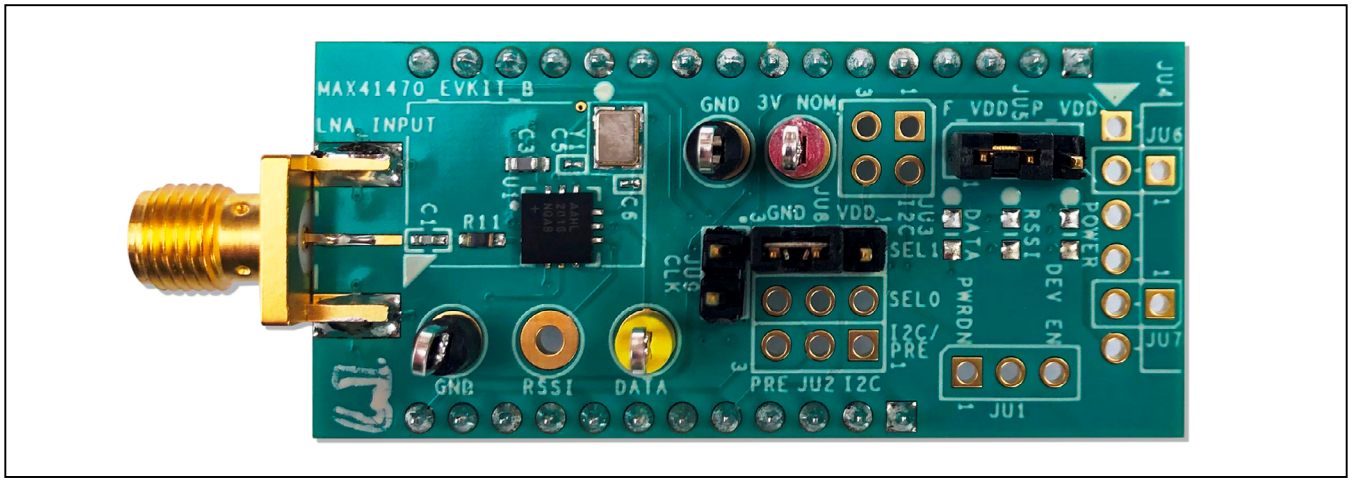
- Evaluates the MAX4147X Sub-1GHz ISM Receiver
- Single Input Voltage Supply from 1.8V to 3.6V
- Direct Interface with a MAX32630FTHR Arm® Microcontroller (MCU) Board
- Available PMOD Hardware Interface
- Windows 10-Compatible Software
- On-Board SPI Interface Control for the MAX41470 and Optional I²C Control for the MAX41473/74
- GUI Controls for MAX32630FTHR Board PMIC
  - Operation from 1.8V to 3.3V
- Proven Four-Layer PCB Design
- Fully Assembled and Tested

MAX4147X EV Kit Contents

- MAX4147X EV Kit Board
- MAX32630FTHR# Kit
  - FTHR Board
  - DAPLINK Board
  - 2x USB Micro-B Cables

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

MAX4147X EV Kit Board



Arm is a registered trademark of Arm Limited (or its subsidiaries) in the US and/or elsewhere.
Windows is a registered trademark and registered service mark of Microsoft Corporation.

maxim integrated™

Quick Start

Required Equipment

- Windows PC* (Win-10) with One USB Port Available
- Power Supply† Capable of 1.8V to 3.6V, 20mA
- RF Signal Generator with ASK/OOK and/or FSK Modulation Capabilities for LNA Input
- Oscilloscope for DATA Output Observation

**Required for operation of the MAX4147X EV kit with the GUI software.*
†Required when the FTHR board is not connected to the MAX4147X EV kit.

Software and Drivers

The MAX4147X EV kit can be used in conjunction with the Arm Cortex®-M4F microcontroller MAX32630FTHR Application Platform or "FTHR" board to provide power and control the device through a software application or GUI. For this option, additional equipment is required:

When connected to the FTHR board, the MAX4147X EV kit uses the following drivers and software components. See *Appendix I* for additional information on this installation process.

- **ISM Radios GUI**

  The software, firmware, and drivers are available from the **Maxim website**. Log in to your MyMaxim account on the website, search for the MAX41470, MAX41473 or MAX41474 IC or EV kit, click on the **Design Resources** tab, and click on the appropriate software link. Finally, click the file link on the software landing page to download the ISM Radios GUI package.

- **mBed MAX32630FTHR and DAPLINK Interface System**

  The DAPLINK system should not be required unless a firmware update to the FTHR board has been released. The FTHR board included in the MAX4147X EV kit will be preprogrammed for interfacing the GUI to the radio. The firmware programming process does not require additional software or drivers—it uses a simple USB drive, drag-and-drop file interface.

It is highly recommended that the target PC be connected to a local area network and have access to the Internet, which allows for automatic download and updates of some drivers. This process may take 15 minutes or more to complete.

Installation Procedure

The steps in this section are used when connecting the MAX4147X EV kit to a FTHR board and should only be needed once—when configuring the hardware and the PC for the first time. If these steps have already been completed, jump directly to the *FTHR Board Quick Start Procedure*.

Install the ISM Radios GUI Software

This process should take less than five minutes after downloading the software package. See *Appendix I* for detailed information on this installation process.

1) Download the ISM Radios GUI software.
2) Double-click the "ISMRadiosGUISetup.msi" setup file and follow Setup Wizard prompts.
   a. Click **Next** in the ISM Radios GUI Setup Wizard window.
   b. It is recommended to use the default Destination Folder; click **Next** to continue.
   c. Install the software by clicking the **Install** button.
   d. Click **Finish** when the ISM Radios GUI Setup Wizard installation process is complete.

Additional Register and QuickStart files may be included in future GUI versions.

Table 1. MAX4147X EV Kit Installed Files and Folders

FILE NAME	DESCRIPTION
ISMRadiosGUISetup.msi	Application GUI
MaximStyle.dll	Supporting DLL file for software operation
MAX4147X_Registers.xml	Register definition file for MAX4147X
MAX4146X_Registers.xml*	Register definition file for MAX4146X
MAX1471_Registers.xml*	Register definition file for MAX1471
MAX4147X_QuickStart.xml	Quick start configuration file for MAX4147X
MAX1471_QuickStart.xml*	Quick start configuration file for MAX1471
Firmware	Folder for current FW at the time of the GUI download

**Not used in this evaluation, but provided with the common platform.*

Cortex is a registered trademark of Arm Limited (or its subsidiaries) in the US and/or elsewhere.

Update the MAX32630FTHR Board Driver on the Host PC

No changes are needed for the FTHR board when first receiving a MAX4147X EV kit—the FTHR board has been pre-loaded with the required firmware. Updates to the driver on the host PC may be necessary depending on the operating system and whether the PC has access to the internet when first connecting to the FTHR board. See *Appendix I* for detailed information on how to update the FTHR board firmware and the driver for the FTHR board/USB interface.

Hardware Use Procedure

Table 2. MAX41470 EV Kit Jumper Settings

JUMPERS	POSITION	EV KIT FUNCTION	
JU1	None Installed	PWRDN open	Power-down state controlled by MCU
JU2	None Installed	I²C mode	
JU3	None Installed	I²C mode	
JU5	1-2*	Power from L3OUT (FTHR board)	
	2-3	Power from PMOD interface (VDD, pin 6 of JU4)	
JU8	1-2	Not used	
	2-3*	CLK output disabled	
JU9	1-2	CLK output enabled (10kΩ to GND)	

**Default position*
***Note:** JU4, JU6, and JU7 not installed*

Table 3. MAX41473/74 EV Kit Jumper Settings

JUMPERS	POSITION	EV KIT FUNCTION	
JU1	1-2	PWRDN to VDD	Preset device in power-down state
	2-3	PWRDN to GND	Preset device not in power-down state
	Not Installed*	PWRDN open	Power-down state controlled by MCU
JU2	1-2†	I²C/PRE to VDD	I²C mode
	2-3	I²C/PRE to GND	Serial interface disabled (or open jumper); see preset tables for details
	4-5	SEL0 to VDD	See preset tables for details
	5-6	SEL0 to GND	
	Not Installed	SEL0 Open	
JU3	1-2†	I²C pullup resistor R8 connected to VDD	
	3-4†	I²C pullup resistor R9 connected to VDD	
	Not Installed‡	Pullup resistors disconnected	
JU5	1-2*	Power from L3OUT (FTHR board)	
	2-3	Power from PMOD interface (VDD, pin 6 of JU4)	
JU8	1-2	SEL1 to VDD	See preset tables for details
	2-3	SEL1 to GND	
	Not Installed	SEL1 Open	
JU9	None Installed	SPI mode	

**Default position*
†Setting for MAX41473 or MAX41474 in I²C mode
‡Default for MAX41473 or MAX41474 in Preset mode
***Note:** JU4, JU6, and JU7 not installed*

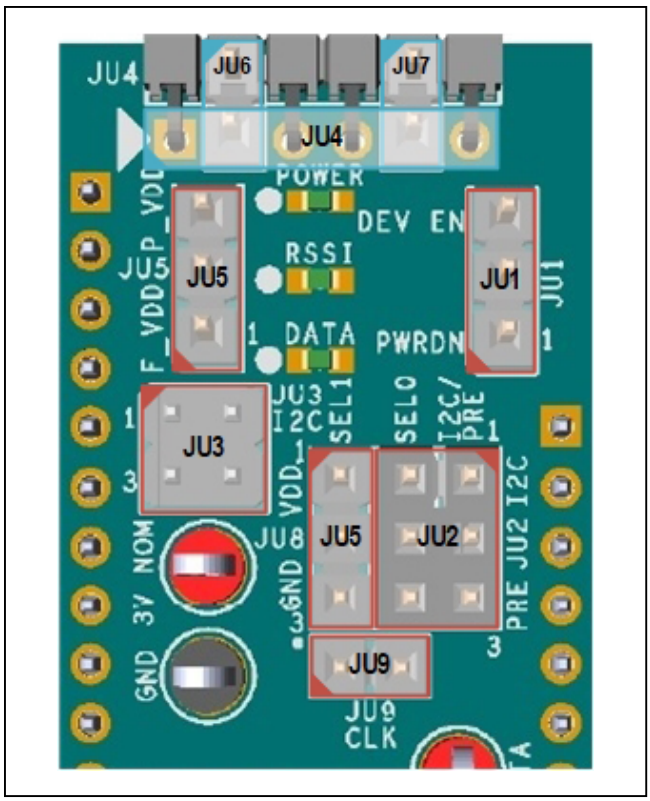


Figure 1. MAX4147X EV Kit Jumpers

FTHR Board Quick Start Procedure – SPI and I²C Interface

Set up the MAX4147X EV kit and FTHR board hardware for MCU/GUI operation as follows:

1) Verify that all jumpers on the MAX4147X EV kit board are in the default position. (See Table 2 and 3 for SPI and I²C respectively).
2) Connect the MAX4147X EV kit to the FTHR board, making certain that the USB connector is oriented on the opposite side of the SMA connector, as shown in Figure 2.
3) Connect the FTHR board to the PC using a USB Micro-B cable, and observe a “heartbeat” on the FTHR board’s red LED (on the opposite side of board from USB connector).
4) Connect the input signal to the LNA_INPUT SMA as the RF signal using a low-loss SMA cable.
   a. ASK modulated signal should be centered at 433.92MHz at 2kbps Manchester (e.g., a 2kHz square wave).

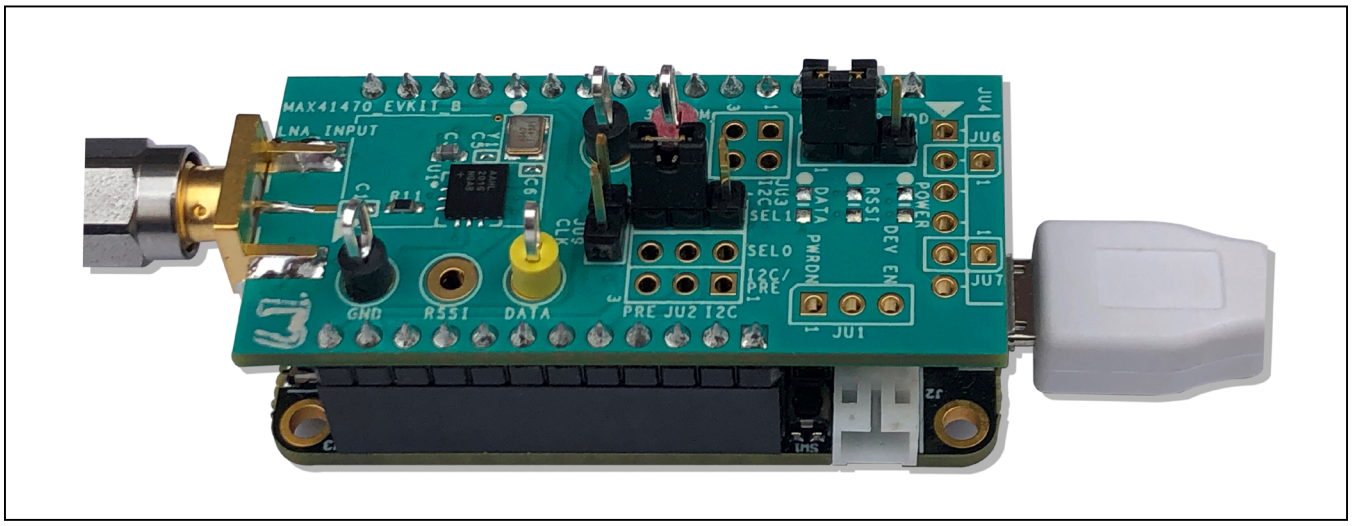


Figure 2. MAX4147XEVKIT Orientation to FTHR Board

5) Start the MAX4147XEVKIT Control Software GUI.
   a. A GUI splash screen will be displayed, as shown in Figure 3.
      i. To disable future displays of the splash screen, select **Disable Splash.**

6) The pop-up window will ask to select a device from the **Device** tab to get started; click **OK** as shown in Figure 4**.**

7) Under the Device menu option, select MAX4147X-Rx (Figure 5). The GUI tabs will populate in the window as shown in Figure 6.



Figure 3. MAX4147XEVKIT GUI Splash Screen

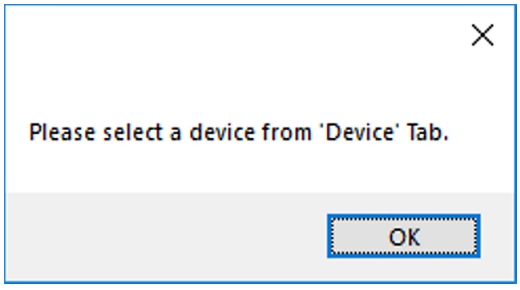


Figure 4. MAX4147XEVKIT GUI Device Selection Reminder

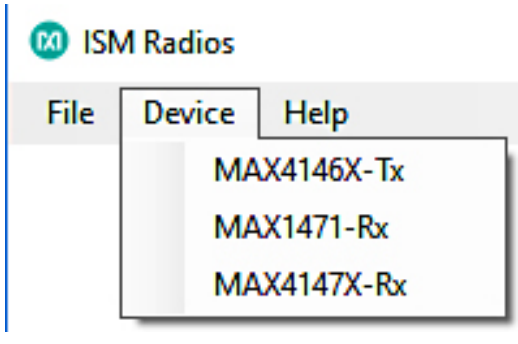


Figure 5. MAX4147XEVKIT GUI Device Selection

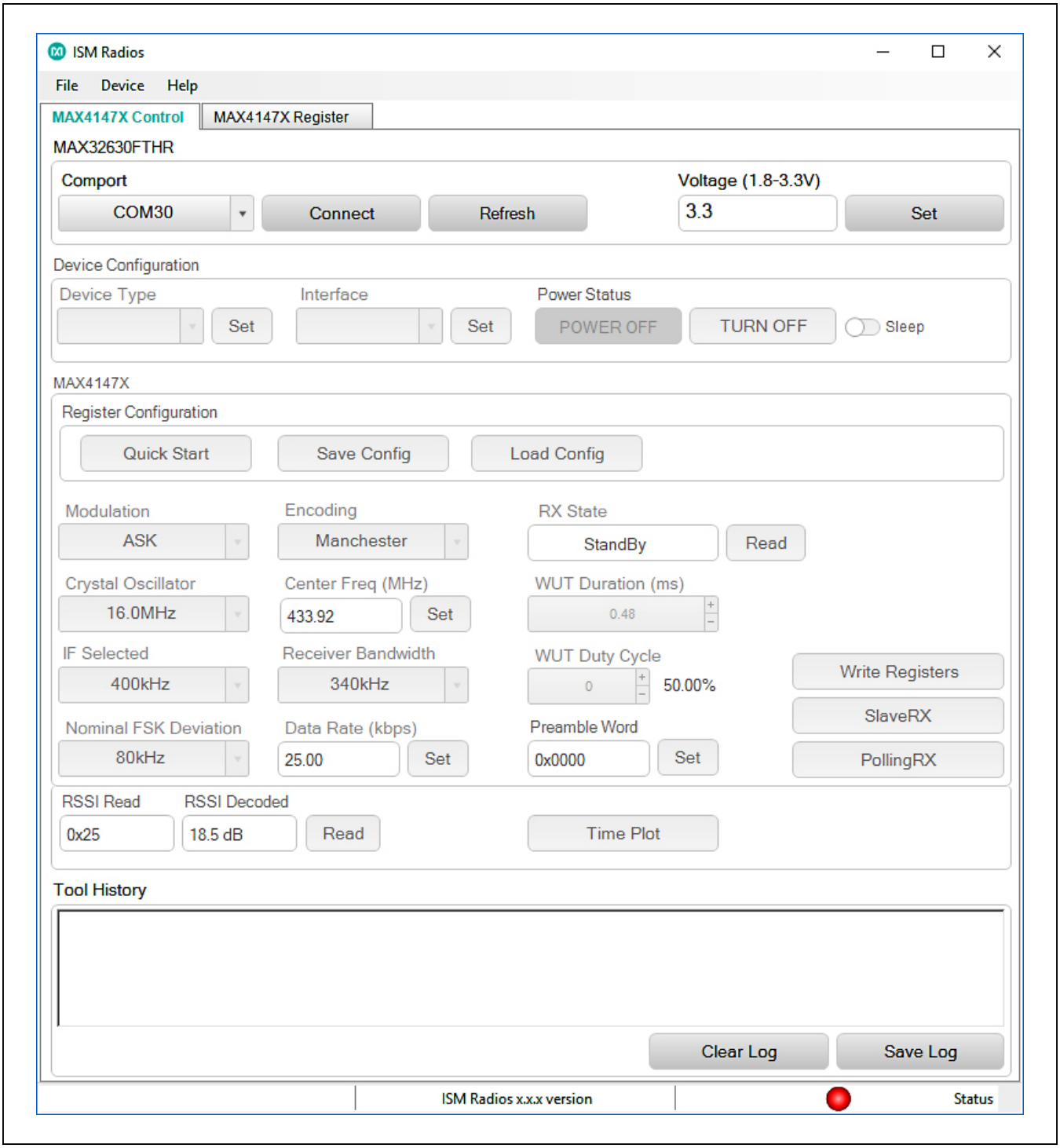


Figure 6. MAX4147XEVKIT GUI Software Startup

8) If the EV kit was connected prior to starting the GUI, the expected COM port should be displayed. Select the appropriate COM port from the drop-down list and click on the **Connect** button. The **Connect** button will change to the **Disconnect** button. The COM port can be verified through the Windows Device Manager. The FTHR board may display under the COM ports as “Teensy USB Serial” or “Maxim USB-to-UART Adapter” and will display the associated port number.



Figure 7. COM Port

9) Confirm that the firmware status bar has changed from “ISM Radios x.x.x” to “ISM Radios 3.7.2” or later, the software LED is lit green, and the port status is noted as “Connected”.

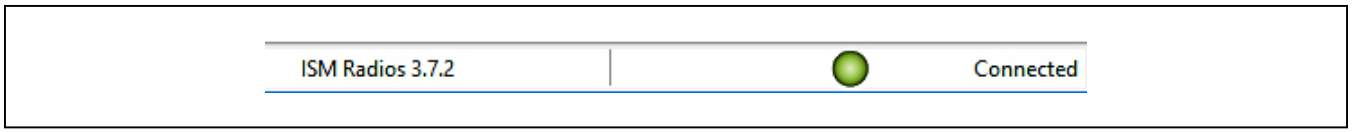


Figure 8. Connected, Indicators at Bottom of GUI

10) Enter a supply level into the **Voltage** field and click the **Set** button.

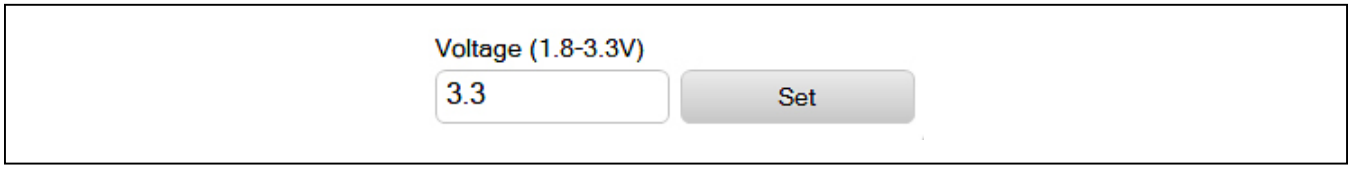


Figure 9. Supply Voltage Example

11) Select the appropriate part in the **Device Type** drop-down list and click the **Set** button.

a. If the MAX41470 is selected, the **Interface** is automatically populated with **SPI**, as it is the only selection. If either MAX41473 or MAX41474 are selected, use the “Interface” pull-down to select either ‘Preset’ or ‘I2C’ and click <Set>.

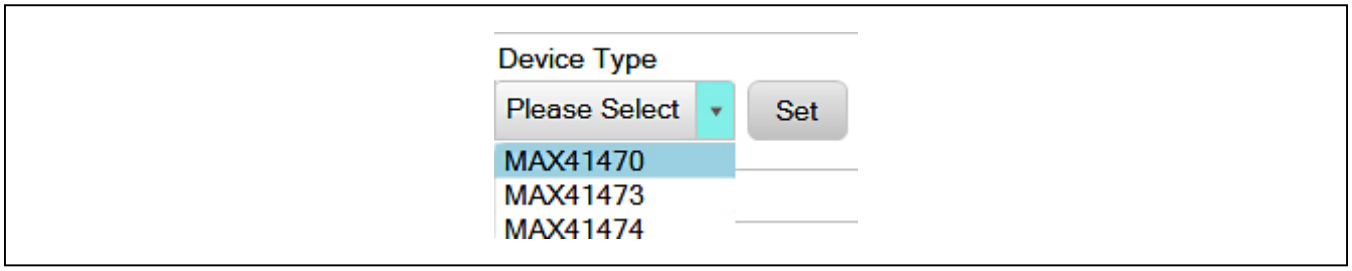


Figure 10. Part Selection

12) Power on the device by clicking **TURN ON**, which will update the status.

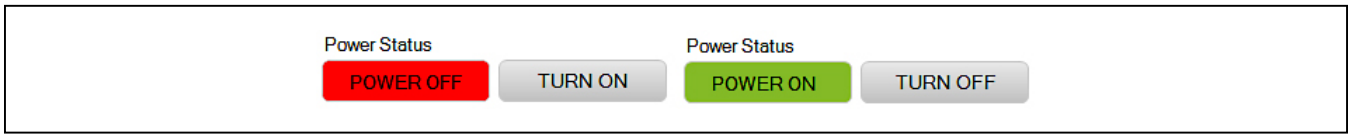


Figure 11. Power On Device

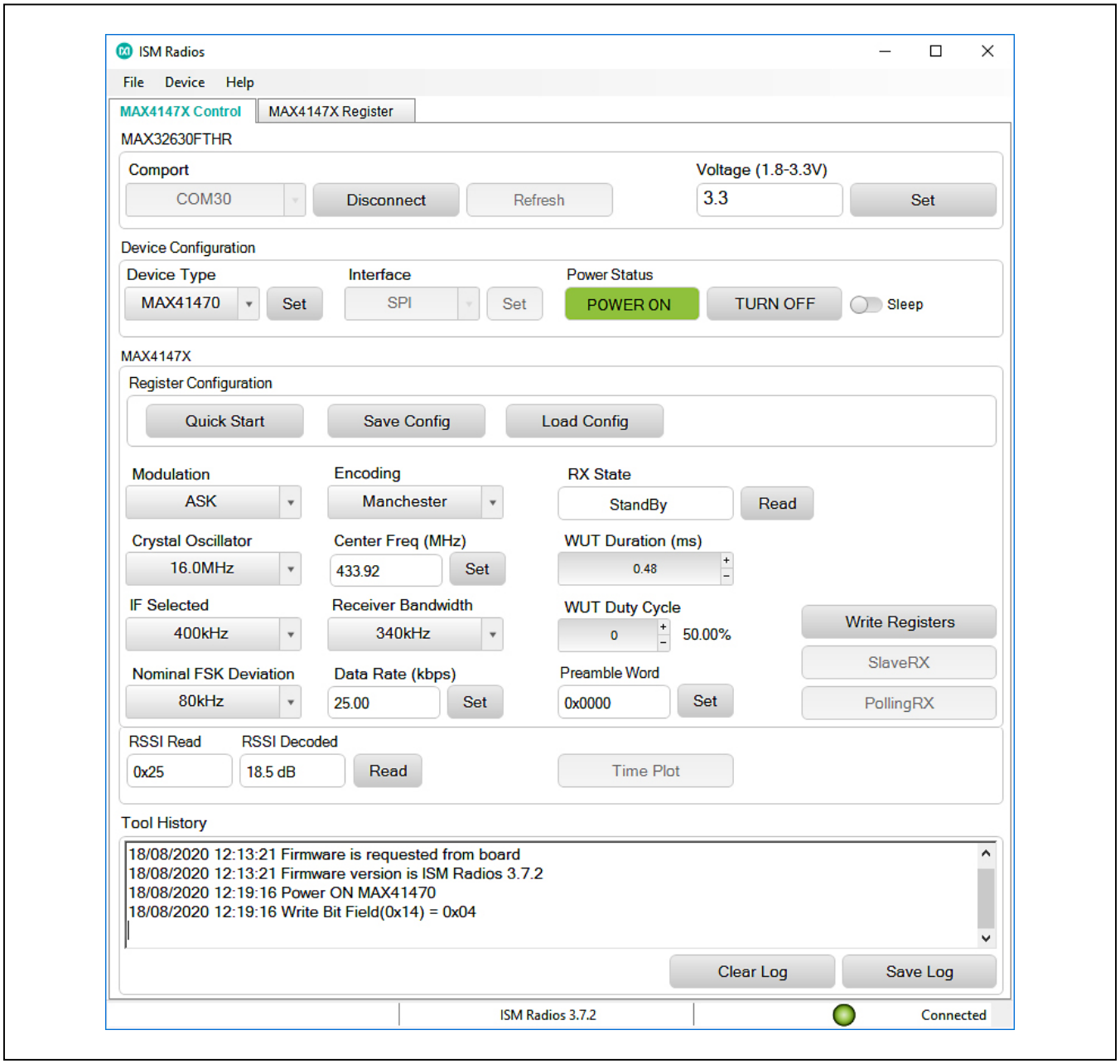


Figure 12. Connected and Powered-On GUI State

13) Configure the Device, either through the Quick Start or the GUI configuration selections.

a. Quick Start

i. Click the **Quick Start** button for an ASK configuration at 433.92MHz with IF at 200kHz and Manchester data rate at 2kbps.

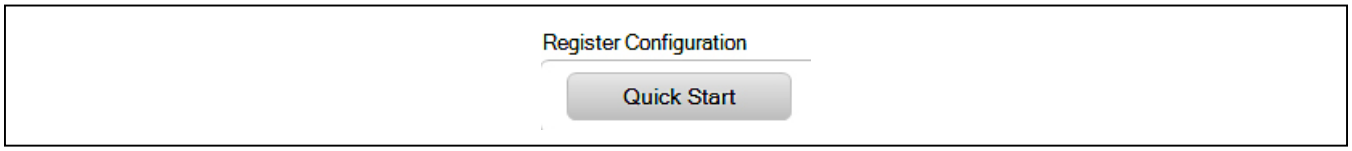


Figure 13. Quick Start Button

b. GUI Drop-Down and Entry Selections.

ii. Select **ASK** in the **Modulation** drop-down list.

iii. Select **Manchester** in the **Encoding** drop-down list.

iv. Enter ”433.92” in the **Center Frequency** field to represent 433.92MHz and click the **Set** button.

v. Select **200kHz** in the **IF Selected** drop-down list.

vi. Select **170kHz** in the **Receiver Bandwidth** drop-down list.

vii. Enter ”2” in the **Data Rate** field and click the **Set** button for a 2kbps setting.

viii. Click the **Write Registers** button to write the configuration into the MAX4147X device.

ix. Click the **SlaveRX** button to put the receiver into the Active SlaveRX state.

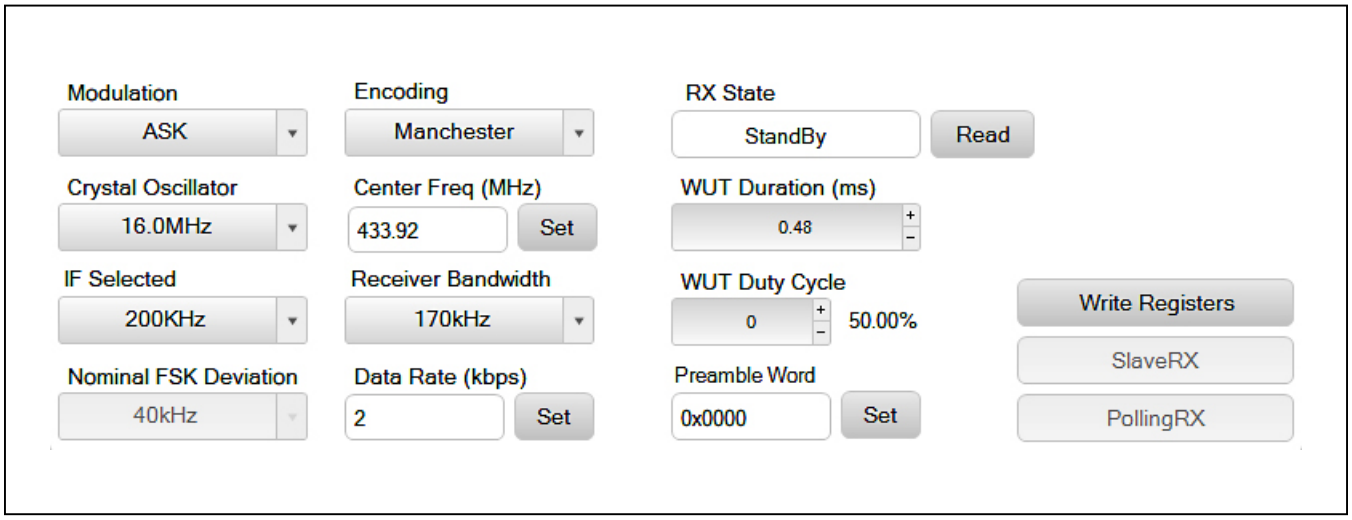


Figure 14. Device Settings and Write

14) Connect the DATA (Yellow) test point to an oscilloscope to see the output data stream or through the **Time Plot** sampler (See the *Time Plot* section in the *Detailed Description of Software* section).

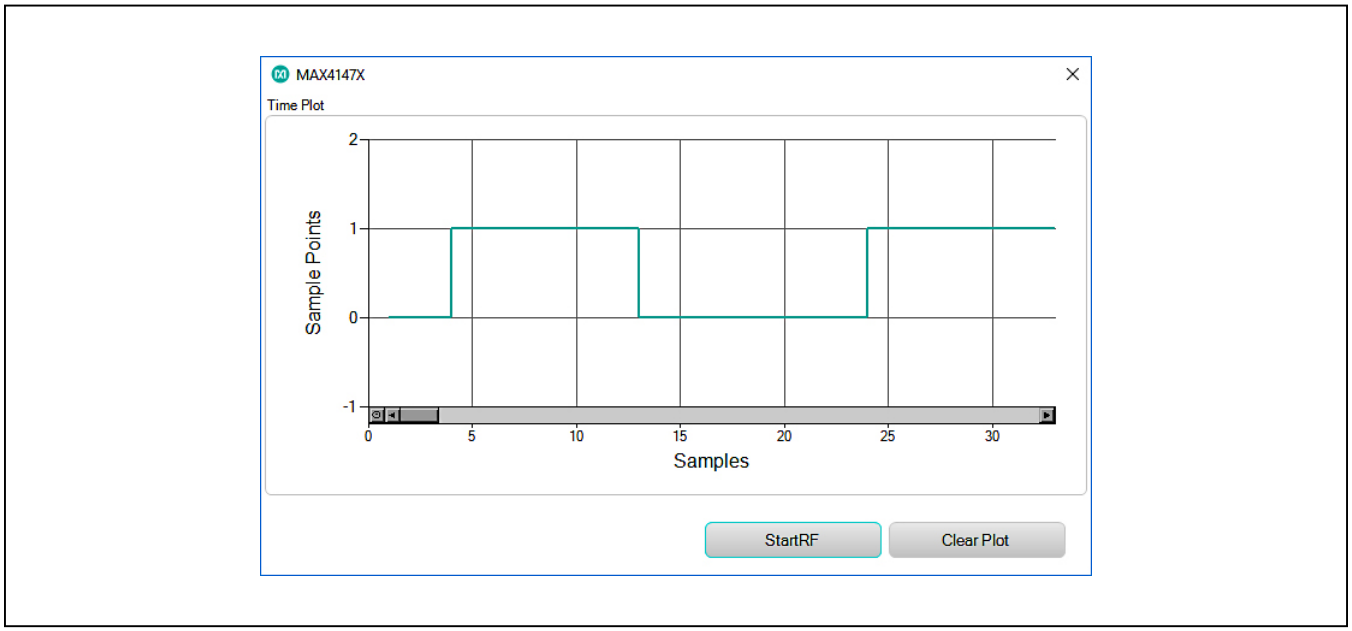


Figure 15. Time Plot of DATA

FTHR Board Quick Start Procedure – Preset Interface

Setup and Connect the MAX41473/74EVKIT Hardware to the FTHR Board for power source and/or to evaluate the RSSI or DATA output.

1) Verify jumpers on the MAX4147XEVKIT board are as follows: JU1 and JU3 are open, JU5 is 1-2 and JU2 is configured according to the Table 4 for MAX41473 or Table 5 for MAX41474.
2) Connect the MAX4147XEVKIT to the FTHR Board, be sure the USB connector is oriented on the opposite side of the SMA connector (see Figure 3).
3) Connect the FTHR Board to the PC using a micro/B USB cable and observe a “heartbeat” on the FTHR board’s red LED.
4) Connect the input signal to the LNA_INPUT SMA as the RF signal using a low-loss SMA cable.
   a. If MAX41473, ASK modulated signal within the defined data rate range.
   b. If MAX41474, FSK modulated signal within the defined data rate and frequency deviation range.
   c. Signal centered at jumper setting preset frequency noted in Table 4 and Table 5.
5) Run the MAX4147XEVKIT Control Software GUI.
   a. Select the appropriate COM port and click on the <Connect> button (see Figure 8).
   b. Confirm the firmware status bar has changed from “ISM Radios x.x.x” to “ISM Radios 3.7.2” or similar, the software LED is lit green, and the port status is noted as “Connected” (see Figure 9).
   c. Enter a supply level into the “Voltage” text box in units of V and click the <Set> button; for example, enter “3.3” for a 3.3V supply (see Figure 10).
   d. Select the appropriate part in the “Device Type” drop-down box (see Figure 11).
   e. Select ‘Preset’ on the “Interface” drop-down.
6) Connect the DATA (Yellow) test point to an oscilloscope to see the output data stream or through the <Time Plot> sampler (See *Time Plot* section under *Detailed Description of Software*).

Preset Quick Start Procedure—Without FTHR Board

Setup and connect the MAX41473/74 EV kit hardware for stand-alone operation as follows:

1) Verify jumpers on the MAX41473/4 EV kit board are as follows: JU1 is 2-3, JU3 is open, JU5 is open. Verify that JU2 and JU8 are configured according to the preset tables in Tables 4 and 5.
2) Connect a 3.0V/20mA supply to the MAX4147X EV kit at the 3V NOM (**Red**) and GND (**Black**) points, output disabled.
3) Connect the input signal to the LNA_INPUT SMA as the RF signal using a low-loss SMA cable.
   a. If MAX41473, ASK modulated signal within the defined data rate range.
   b. If MAX41474, FSK modulated signal within the defined data rate and frequency deviation range.
   c. Signal centered at jumper settings preset frequency noted in Table 4 and Table 5.
4) Enable the power supply's output.
5) Connect the DATA (**Yellow**) test point to an oscilloscope to see the output data stream.

Table 4. MAX41473EVKIT Preset States, ASK

SEL1 (JU2,8)	SEL0 (JU2,5)	I2C/PRESET	FREQUENCY (MHz)	DATA RATE (kbps)*	IF FREQUENCY (kHz)	RECEIVER BW [CHF] (kHz)
Gnd	Gnd	Gnd	315	0.25 to 2.55	200	170
Gnd	Gnd	Open	315	15 to 25	400	340
Gnd	Open	Gnd	318	0.25 to 2.55	200	170
Gnd	Open	Open	318	15 to 25	400	340
Gnd	VDD	Gnd	319.5	0.25 to 2.55	200	170
Gnd	VDD	Open	319.5	15 to 25	400	340
Open	Gnd	Gnd	433.42	0.25 to 2.55	200	170
Open	Gnd	Open	433.42	15 to 25	400	340
Open	Open	Gnd	433.92	0.25 to 2.55	200	170
Open	Open	Open	433.92	15 to 25	400	340
Open	VDD	Gnd	868.3	0.25 to 2.55	400	340
Open	VDD	Open	868.3	15 to 25	400	340
VDD	Gnd	Gnd	868.5	0.25 to 2.55	400	340
VDD	Gnd	Open	868.5	15 to 25	400	340
VDD	Open	Gnd	915	0.25 to 2.55	400	340
VDD	Open	Open	915	15 to 25	400	340
VDD	VDD	Gnd	868.35	0.25 to 2.55	400	340
VDD	VDD	Open	868.35	15 to 25	400	340

**Assuming Manchester encoded data*

Table 5. MAX41474EVKIT Preset States, FSK

SEL1 (JU2,8)	SEL0 (JU2,5)	I2C/PRESET	FREQUENCY (MHz)	DATA RATE (kbps)*	IF FREQUENCY (kHz)	RECEIVER BW [CHF] (kHz)	DEVIATION (kHz)
Gnd	Gnd	Gnd	315	0.5 to 2.55	200	170	32 to 47
Gnd	Gnd	Open	315	15 to 25	400	340	64 to 94
Gnd	Open	Gnd	318	0.5 to 2.55	200	170	32 to 47
Gnd	Open	Open	318	15 to 25	400	340	64 to 94
Gnd	VDD	Gnd	319.5	0.5 to 2.55	200	170	32 to 47
Gnd	VDD	Open	319.5	15 to 25	400	340	64 to 94
Open	Gnd	Gnd	433.42	0.5 to 2.55	200	170	32 to 47
Open	Gnd	Open	433.42	15 to 25	400	340	64 to 94
Open	Open	Gnd	433.92	0.5 to 2.55	200	170	32 to 47
Open	Open	Open	433.92	15 to 25	400	340	64 to 94
Open	VDD	Gnd	868.3	0.5 to 2.55	400	340	64 to 94
Open	VDD	Open	868.3	15 to 25	400	340	64 to 94
VDD	Gnd	Gnd	868.5	0.5 to 2.55	400	340	64 to 94
VDD	Gnd	Open	868.5	15 to 25	400	340	64 to 94
VDD	Open	Gnd	915	0.5 to 2.55	400	340	64 to 94
VDD	Open	Open	915	15 to 25	400	340	64 to 94
VDD	VDD	Gnd	868.35	0.5 to 2.55	400	340	64 to 94
VDD	VDD	Open	868.35	15 to 25	400	340	64 to 94

**Assuming Manchester encoded data*

Table 6. MAX4147X EV Kit Test Points

NAME	COLOR	EV KIT FUNCTION
3V NOM	Red	1.8V to 3.6V power supply pin
GND	Black	Ground
DATA	Yellow	RX data
RSSI	Green	RSSI data (MAX41473/MAX41474 preset only)

Detailed Description

Detailed Description of Hardware

MAX4147X EV Kit Printed Circuit Board

The MAX4147X EV kit PCB is manufactured on a 4-layer, 1oz copper, FR4 dielectric stack-up PCB. The board was designed to accommodate all three versions of the ISM receiver: MAX41470, MAX41473 and MAX41474. The board was designed to accommodate the MAX41470 and other members of the MAX4147X family with the use of resistors and jumpers. Layer 1 is used to route the receiver and oscillator signals. Layer 2 is a ground layer. Layers 3 and 4 are used to route the jumper connections to support the versatility of the board and its support for multiple parts and configurations.

Control Interface

There are three forms of interfacing to the MAX4147X device depending on the part installed: 3-wire SPI, I²C control interface or preset/pin-configured. The MAX41470 device will require a 3-wire SPI connection and the MAX4147X EV kit was designed to use the provided FTHR board interface through the H1/H2 headers. Other MCU connections can be made through the JU4 PMOD header, if desired (see the *PMOD Interface* section).

Power

The MAX4147X EV kit board can be powered directly from the FTHR board PMIC through the H1 header, directly from the supply test points, or through the user-installed PMOD header. A single +1.8V to +3.6V, 20mA power supply can be connected to the board using the two wire loops (marked 3V NOM and GND) when JU5 is not populated. Jumper JU5 selects the source of power when not using the direct connection test points: from the L3OUT of the FTHR board or the PMOD_VDD of the PMOD connector.

Data Interface

The MAX4147X EV kit comes preconfigured to directly connect the FTHR board through the H1/H2 headers to the SPI and the I2C interfaces. The GUI will determine which bus is used to communicate to the device based on the Device Type and Interface pull-downs selected in the software.

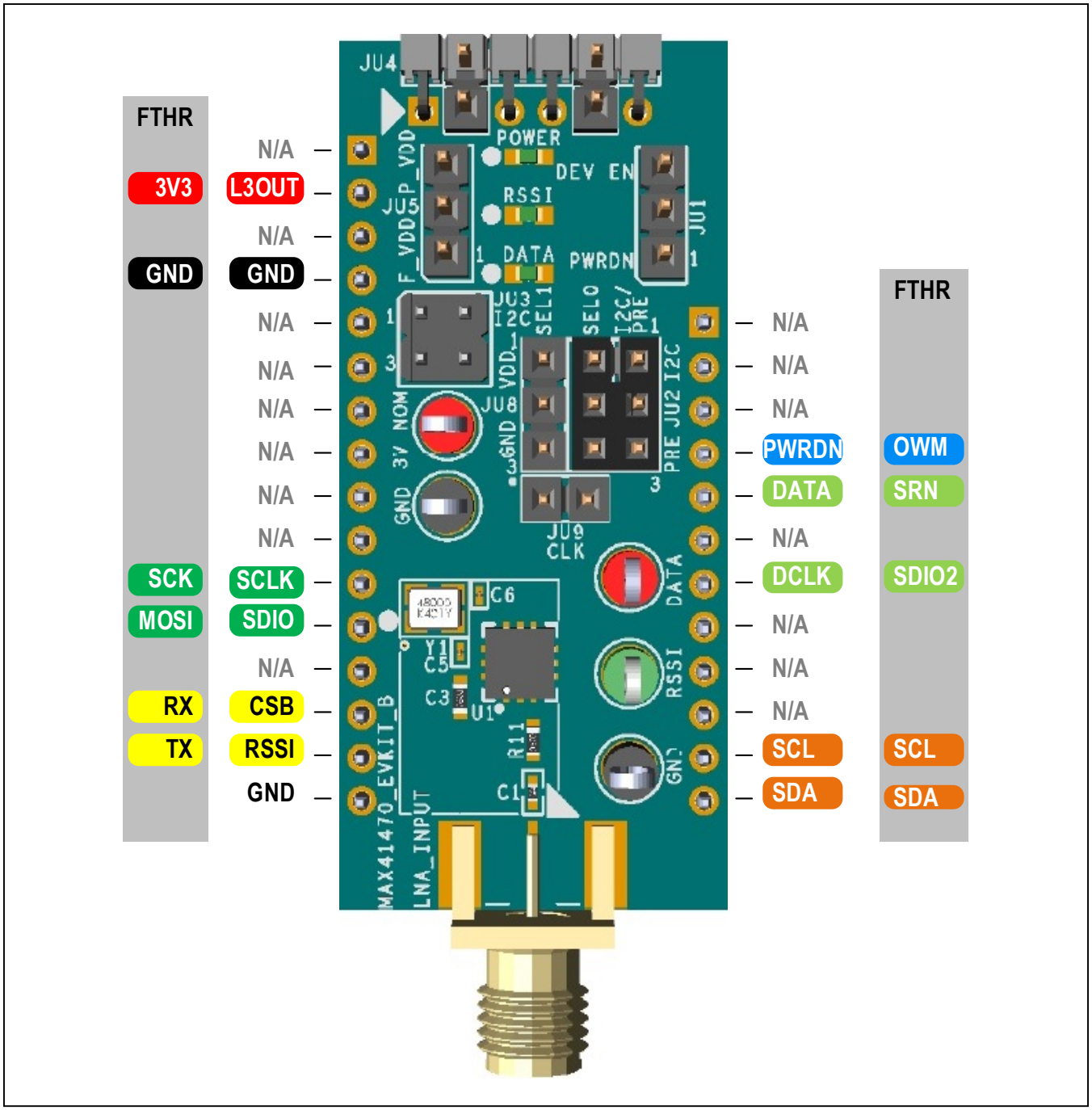


Figure 16. MAX4147X EV Kit Interface

I²C Pull-Ups

Resistors R8 and R9 along with jumper JU3 have been provided as on-board pull-ups which are required for proper I²C interfacing and termination. These pins are open-collector (or open-drain) outputs from the MCU and need to have pull-up resistors to operate properly. Two 4.7kΩ resistors are pre-populated on the MAX4147XEVKIT and can be connected to the positive supply by shorting the JU3 jumper 1-2 and 3-4. This should only be connected when the I2C/PRESET pin is connected to logic high, thus selecting the I²C interface mode of the MAX41473/74. It should be noted that the FTHR board also has footprints for I²C pull-up resistors at R6 and R11. Both sets of pull-up resistors (on the FTHR board and the MAX4147XEVKIT) should not be populated simultaneously, otherwise incorrect I²C signal levels may result. Refer to Appendix II for detailed information on Evaluation Kit Hardware Modifications.

Data Indicator

An option available on the EV kit layout is the ability to connect a surface-mount LED (POWER, DATA and/or RSSI, 0603) and resistors (R14, R15 and R16 respectively, 0603, 470Ω recommended) to provide visual feedback of the activity on the supply, DATA and RSSI lines. Populating these LEDs and resistors will cause additional power consumption and they are not included by default in the EV kit assembly.

PMOD Interface

The MAX4147X EV kit provides a PMOD-compatible header footprint to interface with the transmitter. The JU4 connector can be populated with a 6-pin, 100mil, right-angle header allowing direct connections to the CSB, RSSI, SCL, SCLK_SDA, ground, and VDD lines, making it compatible with either SPI or I²C PMOD interfacing. Populating this header would allow control from the MAX32600MBED kit and the MAXREFDES72# Arduino Uno R3 to PMOD shield adaptor. When using the PMOD interface to supply the MAX4147X EV kit with power, make sure to connect the JU5 jumper between pins 2-3. See *Appendix II* for detailed information on EV kit hardware modifications.

Detailed Description of Software

The MAX4147X EV kit controller GUI software is designed to control the MAX4147X EV kit board and the MAX32630FTHR board, as shown in Figure 17. The software includes USB controls, which provide SPI, I²C, and power to the MAX4147X through the FTHR board interface.

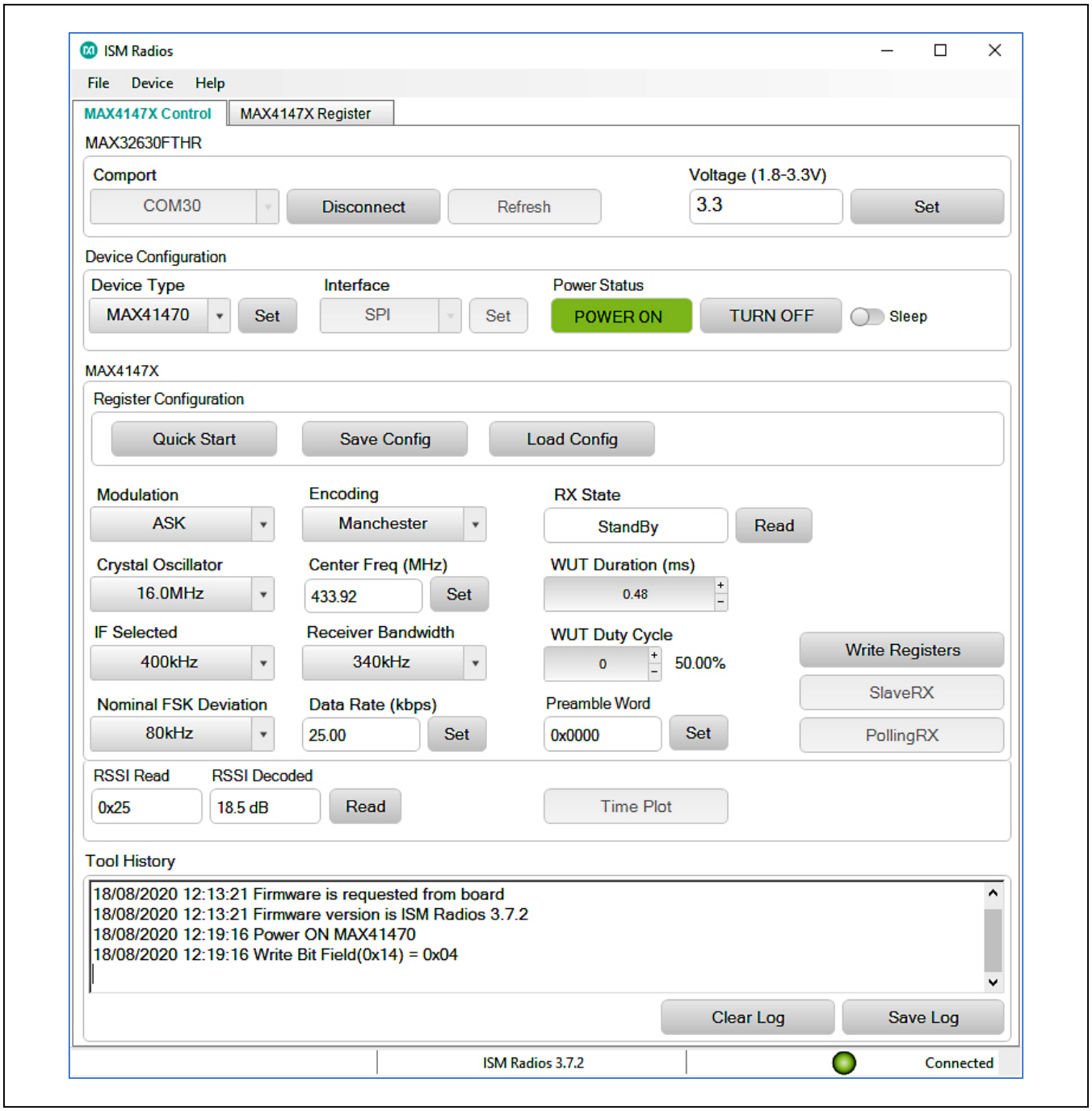


Figure 17. MAX4147X EV Kit GUI Configuration

Comport

The **Comport** section provides a drop-down list of serial communication ports available for connection to a MAX4147X EV kit through a FTHR board. When the GUI is run after connecting the EV kit hardware, the drop-down list should default to the proper COM port. If the hardware is connected to the computer after the GUI is started, click on the **Refresh** button to scan for compatible ports. Once the appropriate COM port is selected in the drop-down list, click on the **Connect** button. (See Figure 7.)

After properly connecting to the COM port with the FTHR board, the GUI will display the revision of FTHR board firmware detected, display a **Green** "LED", and display "Connected" in the status bar along the bottom of the GUI window. (See Figure 8.)

Voltage (1.8V to 3.3V)

The **Voltage** section provides a user-adjustable power supply from the FTHR board MAX14690N power management IC (PMIC) to the MAX4147X EV kit and can be used as the primary VDD supply. The PMIC L3OUT can be set to voltages between 1.8V to 3.3V and it applies to the level of the logic interface lines as well as the device supply. (See Figure 9.)

To program the supply **voltage**, enter a valid level in the Voltage field and click on the **Set** button. The default value of the L3OUT voltage is 3.3V.

When using the FTHR board interface to supply the MAX4147X EV kit with power, make sure to connect the JU5 jumper between pins 1-2.

Device Type

The **Device Type** section must be set by the user to properly chose which receiver is attached to the FTHR board. This selection will configure the GUI software to interface through the SPI pins (when MAX41470 is selected) or through the I^2C pins (when the MAX41473 or MAX41474 are selected).

To select the receiver, chose the appropriate part in the **Device Type** drop-down list and click on the **Set** button. (See Figure 10.)

Modulation

The **Modulation** section allows the user to set the form of modulation for the MAX4147X device. When a MAX41473 is selected, only ASK modulation will be available in the drop-down box. Similarly, when MAX41474 is selected, only FSK will be available in the drop-down box.

To select the modulation, chose ASK or FSK in the **Modulation** drop-down list. (See Figure 14.)

Encoding

The **Encoding** section allows the user to define the type of data being transmitted. The two options here are Manchester or NRZ and are available in the drop-down list. (See Figure 14.)

Crystal Oscillator

The **Crystal Oscillator** section allows the user to match the GUI calculation to the frequency of the crystal installed on the MAX4147X EV kit (f_{XTAL}). All EV kits come prepopulated with a 16.000MHz crystal, and the default setting in the GUI is assumed to be 16.0MHz. This value can be adjusted to 12.8MHz or 19.2MHz and will be used when programming frequency-based registers that are dependent on the f_{XTAL} value.

To change the reference oscillator, select the oscillator frequency in the drop-down list. (See Figure 14.)

Center Frequency

The **Center Freq** section is used to set the carrier or "center" frequency of the MAX4147X (f_C or f_{LO}). The value entered in this section will be used to calculate the three-word fractional-N value programmed into the LO_CTR_FREQ3 (0x09), LO_CTR_FREQ2 (0x0A), and LO_CTR_FREQ1 (0x0B) registers. The GUI will calculate the values for the PLL registers using the crystal frequency and the following formula:

$$LO_{CTR_{FREQ}}[23:0] = \text{Round}\left(\frac{65536 \times (f_{RF} - f_{IF})}{f_{XTAL}}\right)$$

To program the carrier, enter a valid frequency (in MHz) into the **Center Freq** field and click the **Set** button. (See Figure 14.)

IF Selected

The IF selection selects the IF frequency to be configured. Through the drop-down list, either 200kHz or 400kHz IF frequencies will be selected for configuration in the device. (See Figure 14.)

Receiver Bandwidth

The channel filter bandwidth programmed into the device is selected using the **Receiver Bandwidth** drop-down list. The population of this list is dependent on the IF frequency selected. The options correspond to those listed in register IF_CHF_SEL (0x002). When 200kHz IF is selected, the options are 170kHz, 60kHz, 26kHz, 12kHz, or 6kHz. If 400kHz IF is selected, the options are 340kHz, 120kHz, 52kHz, 24kHz, or 12kHz. Through the drop-down list, the desired channel filter bandwidth (CHF) is selected for configuration in the device. (See Figure 14.)

Nominal FSK Deviation

This is only valid when the modulation selected is FSK. When FSK is selected, the **Nominal FSK Deviation** value will be available through the drop-down list. The available ranges in the drop-down list are based on the configuration factors already defined to include the IF frequency and the channel filter bandwidth. This will impact the DEMOD_FSK setting in the DEMOD register (0x00).

Data Rate

Enter the configured data rate of the transmitted signal into the **Data Rate** field in kbps. Then click the **Set** button to set the programming of the device.

As shown in Figure 18, a Manchester 2kbps signal could look like a 2kHz square wave to represent all 1's (or all 0's if shifted in phase) in Manchester encoded data. And an NRZ 4kbps signal could look like a 2kHz square wave to represent alternating 0's and 1's.

Intention of Polling

Polling allows the system to go into a lower-power state until the receiver and microcontroller are awakened for active receiving of the signal. The microcontroller can be in a pause or sleep state until the MAX4147X detects the preamble word. When the word is detected, the interrupt from the MAX4147X goes to the microcontroller to wake it up for 'listening'. The DATA line during the PollingRX state is kept high. At the time of the detection, the DATA line will transition low as the interrupt. The microcontroller acknowledges the interrupt by reading the ISR (0x13) register. If the PREAM_DET bit is set in the ISR register, then the microcontroller places the MAX4147X into the SlaveRX state through the STATE_CTRL1 register, the same register used to place the device into the PollingRX state (SLAVE_RX_EN bit versus the WUT_EN bit). When in the SlaveRX state, the WUT Duration, WUT Duty Cycle, and Preamble Word do not impact the operation. These are only used in the PollingRX state.

Polling – WUT Duration

The WUT Duration is a configurable duration that defines the time the receiver is in the PollingRX state. This is programmable between 0.48ms and 20.88ms. This programmed value is only used when in Polling mode and will not impact the SlaveRX state.

Polling – WUT Duty Cycle

The WUT Duty Cycle defines the duty cycle of the detection duration versus the wait time. See the device datasheet for *Wake-Up Timer in Self-Polling Mode* diagram for more details. The ratio is programmed in the WUT2 (0x18) register where:

$$\text{Duty Cycle} = \frac{1}{(2 + \text{TSBY_TDET_RATIO})}$$

Polling – Preamble Word

The **Preamble Word** is the Manchester encoded word that is compared against the incoming data stream. When the word is detected, the PREAMB_DET bit is set in the ISR register and the DATA line, which is held high during the PollingRX state, will transition from high to low. This is the interrupt for the microcontroller. It is important to note that the preamble word is Manchester encoded. This means that a standard square wave into the device as the preamble would require that the preamble word be programmed with 0x0000 or 0xFFFF (depending on the length desired for the preamble word, as defined in the PREAMBLE_CFG1 register through the PREAM_LEN bits). After entering the desired value in the **Preamble Word** field, click the **Set** button. The GUI detects the number of nibbles through the entry in the **Preamble Word** field and will program the PREAMB_LEN based on this entry.

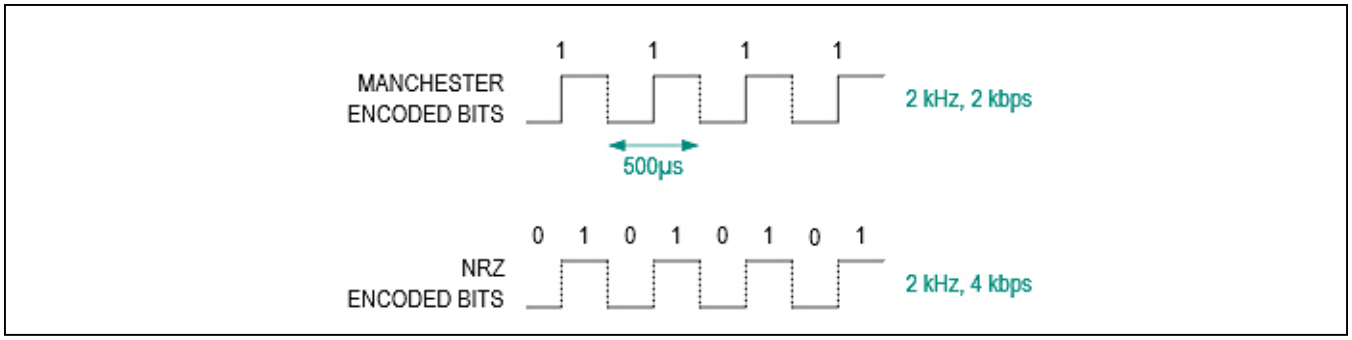


Figure 18. Manchester and NRZ Data Rates

Action Buttons – Write Registers

The **Write Registers** button will take the combination of all the settings within the GUI and write the registers accordingly. Many of the settings within the MAX4147X are dependent on multiple bit settings within the device. This button will combine all settings to determine the register writes required.

Action Buttons – SlaveRX

The **SlaveRX** button will be active once the registers are programmed, either through the **Write Registers** button or the **Load Config** button. This button will enable the SlaveRX mode within the receiver for active data reception. In this state, the DATA test point on the EV kit will display the data received. Since the configuration registers should not be programmed in any active state, the GUI will grey out the other buttons and settings during the SlaveRX state. The RX State status will move to the Slave Receiver state. To exit the SlaveRX state, click the **SlaveRX** button again.

Action Buttons – PollingRX

The **PollingRX** button will be active once the registers are programmed, either through the **Write Registers** button or the **Load Config** button. This button will place the device into PollingRX mode for preamble detection. When the preamble is detected, the microcontroller will automatically respond appropriately as described above and place the device into SlaveRX mode.

Time Plot

The **Time Plot** button allows the user to visually see the DATA pin displayed in a plot. This plot works best at around the 10kbps rate and lower for proper oversampling of the received signal.

The **StartRF** buttons should be clicked to begin the display. After it is running, the **StopRF** can be clicked to stop the sampling. Clicking **Clear Plot** will clear the current display. To zoom in or out of the display samples, use the mouse scroll.

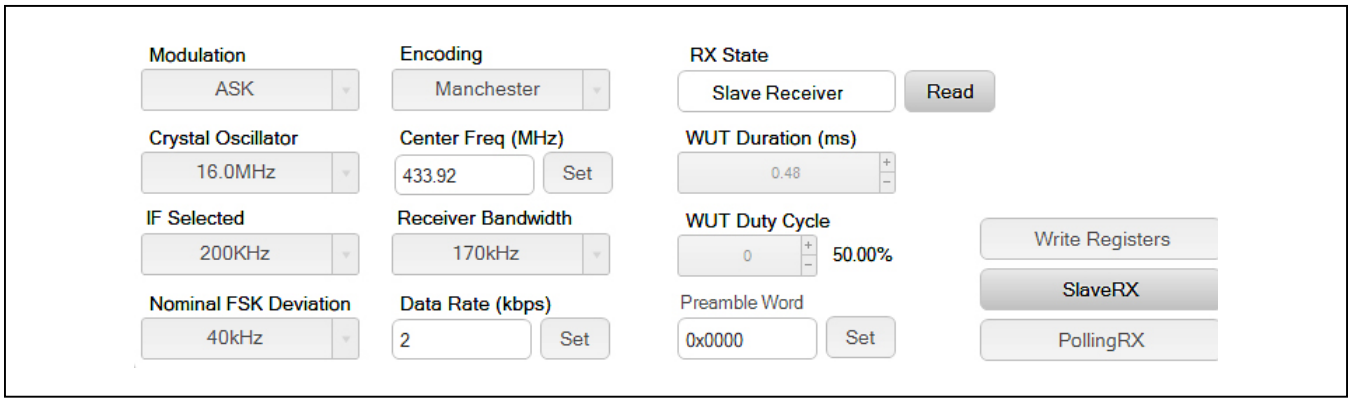


Figure 19. Active SlaveRX State

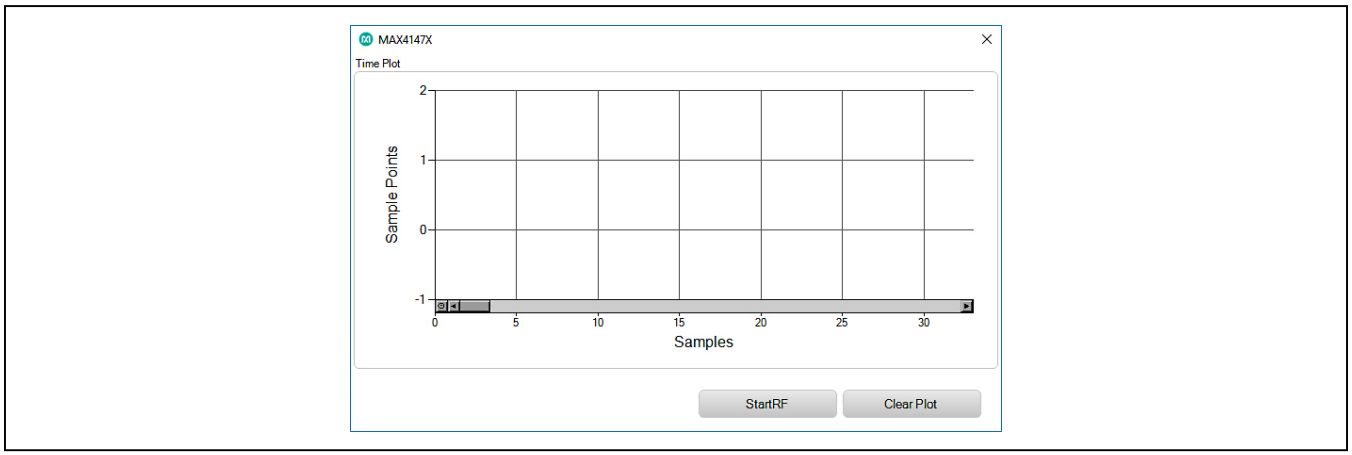


Figure 20. Initial Time Plot Window

The **StartRF** buttons should be clicked to begin the display. After it is running, the **StopRF** can be clicked to stop the sampling. Clicking **Clear Plot** will clear the current display. To zoom in or out of the display samples, use the mouse scroll.

Tool History

This portion of the GUI contains a Log File text block which is used to record activity within the GUI.

Logging

For every **Set**, connection effort, or register programming action, the GUI activity is logged in this text block. The user can add notes and make edits to the content of the Log File text block.

Clicking on the **Clear Log File** will delete the contents in the text block.

Clicking the **Save Log** button will open a **Save As** explorer window and the user will be prompted to save a .txt file.

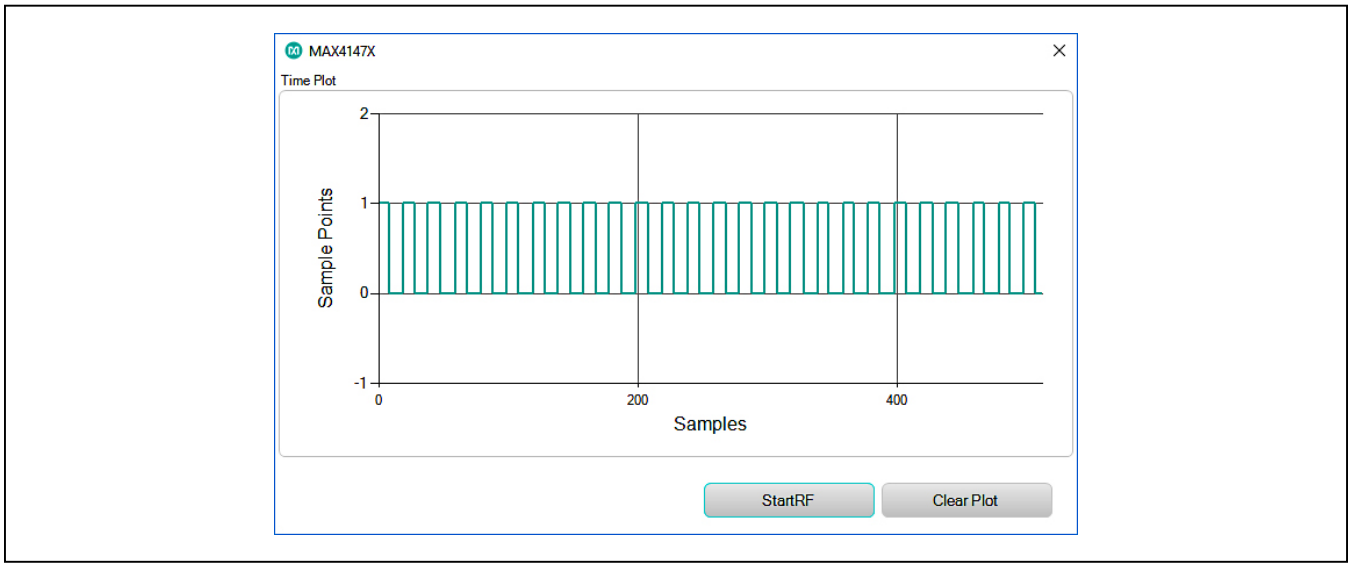


Figure 21. Time Plot Display

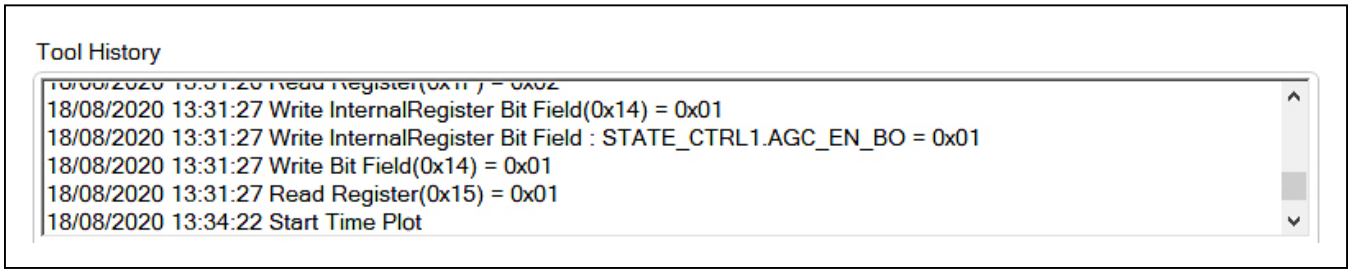


Figure 22. Tool History

Direct Register Access

The GUI software allows for direct access to all the available registers through the MAX4147X **Register** tab when interfacing with the MAX41470 SPI-based device or the MAX41473/74 devices in I²C mode.

Register List

On the left-hand side of the **Register** tab is a list of the device's internal registers. Each register address/name (e.g., "14 STATE_CTRL1") acts as an active control and by clicking on an individual register, the contents will be presented in the *Register Value* section.

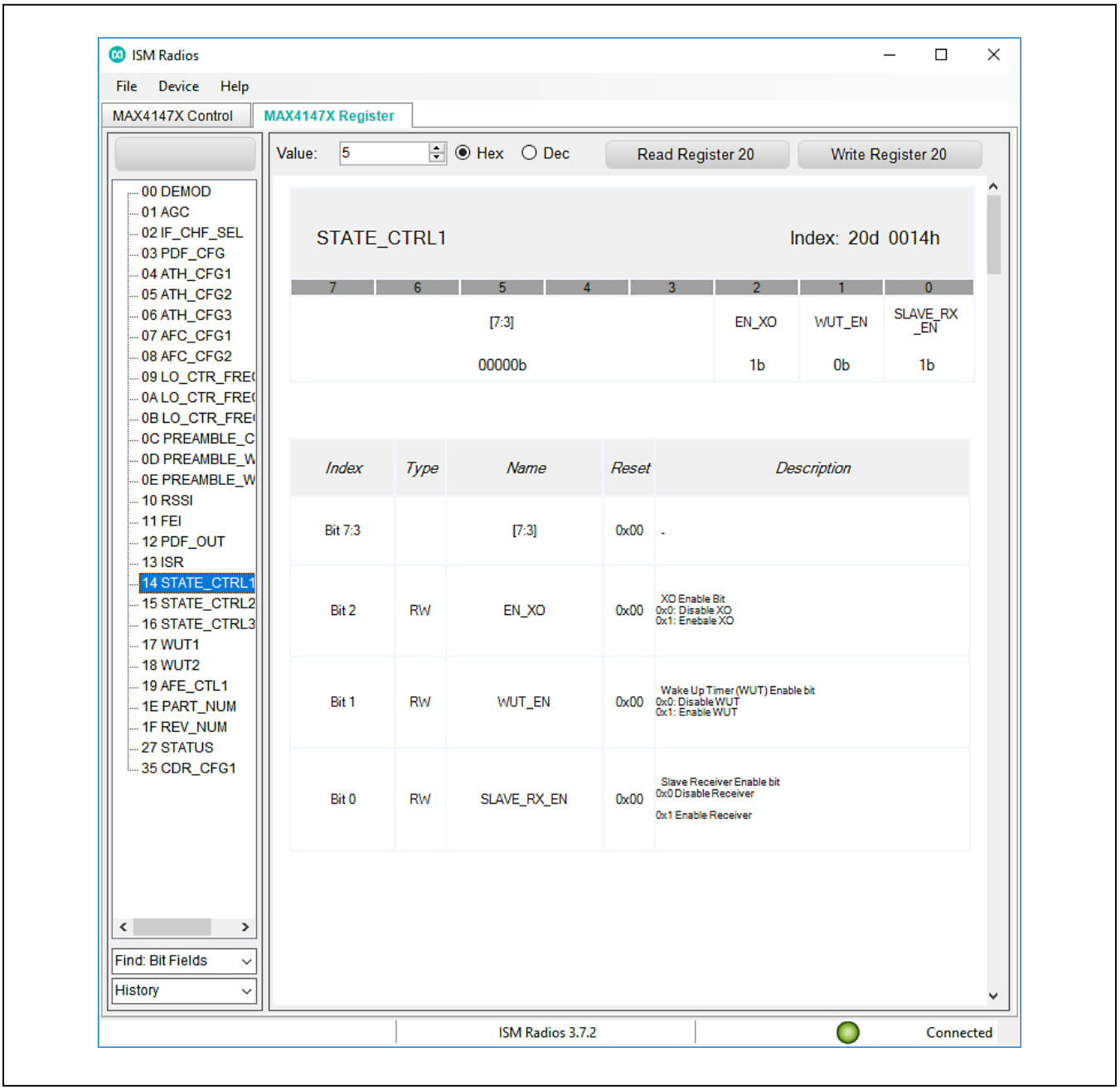


Figure 22. Tool History

Register Value

The right-hand side of the **Register Interface** section displays the content of the selected device register. At the top of the block, a header displays the name of the selected register (eg “STATE_CTRL1”), the “Index” or address of the register in both decimal (“0d”) and hexa-decimal (“0000h”) form.

The body of this section shows a table with the names of the individual bits for the selected 8b register along with the current value programmed into each bit or bit group.

The remaining portion of the body shows a table with the bit indexes, the type of register (write/read), the name of the bit or bit group, the Reset value, and a description of the bit or bit group.

Read and Write Registers

Most of the registers in the MAX4147X are both readable and writable. The read-only registers are RSSI (0x10), FEI (0x11), PDF_OUT (0x12), ISR (0x13), STATE_CTRL2 (0x15), PART_NUM (0x1E), and REV_NUM (0x1F). Writing values to a register can be accomplished by selecting the register of interest, typing a Hex or Dec value into the **Value** field, and clicking on the **Write Register X** button (where X is the decimal address of the register). Reading the register content is similar: select the register of interest and click on the **Read Register X** button.

Register Bit Field

Individual bits or bit groups can be programmed without having to enter the full value of the register. To program a bit or group of bits, first select the register of interest (WUT2, 0x18 for example), next select the bit or bit group to be changed (TSBY_TDET_RATIO as an example), enter the binary code for the new value (0000011b), and hit **Enter**—the new value will automatically be reflected in the **Value** field and will be written to the device.

Miscellaneous Software Information

The tool bar along the top of the GUI software provides a couple of options to the user.

File and Help Menu

Selecting **File** > **Exit** from the tool bar will close the GUI program. This has the same effect as clicking the **X** button in the upper-right corner of the GUI software.

Selecting **Help** > **About** from the tool bar will display the splash screen. This window shows the name of the software, the revision number, a copyright notice, a link to the Maxim website, a link to the support website, and a checkbox to enable or disable the splash screen during startup. Click the **OK** button to close the **About** window.

.xml File

The register descriptions for the MAX4147X GUI are available in an .xml file which is stored with the executable in the application directory. The default file loaded when the GUI is initialized is `MAX4147X_Registers.XML`. This file can be edited as needed to adjust the names of fields, provide simple indicators to the GUI user, or allow for flexible updates to the GUI interface in the future.

Use Cases

Two Interface Modes for Data Transmission and Control

The MAX4147X allows a great deal of flexibility when it comes to receiving data. Typically, the fewer pins used to interface with the device, the simpler it is to control and transmit data.

Preset Mode

Preset mode is the simplest interface of the three options. It relies on the part number to choose the modulation (MAX41473 for ASK; MAX41474 for FSK) and jumper settings (or tri-level pin connections) to configure the part for a defined carrier frequency (See Tables 4 and 5). With the Preset mode, the data rate and frequency deviation, in the case of FSK modulation, need to be within expected ranges for proper operation.

I²C Mode

The I²C interface mode allows the user to access the internal registers of the MAX41473/74 devices, permitting full control over the receive frequency, data rate, thresholds, bandwidths, etc.

This mode only requires two digital pins to interface with the transmitter but has a more complicated, packet-based protocol for interacting with the device. Once configured for I²C interfacing by connecting the I2C/PRESET pin to logic high, the MAX41473/74 can support a 2-wire I²C-compatible serial interface consisting of a serial-data line (SDA) and a serial-clock line (SCL). SDA and SCL facilitate bidirectional communication between the MAX41473/74 and the master (microcontroller) at clock frequencies up to 1MHz. The master device initiates a data transfer on the bus and generates the SCL signal to permit data transfer. The MAX41473/74 functions as an I²C slave device that transfers and receives data to and from the master. It is necessary to pull SDA and SCL high with external pullup resistors via population of the R8 and R9 resistors and the installation of JU3 1-2 and 3-4.

One bit transfers during each SCL clock cycle. A minimum of nine clock cycles is required to transfer a byte into or out of the MAX41473/74 (8 bits and an ACK/NACK). The data on SDA must remain stable during the high period of the SCL clock pulse. Both SDA and SCL remain high when the bus is not busy.

All the MAX41473/74 devices are identified with an I2C address of 0xD6 for write and 0xD7 for read sequences. Packet transmission is described in the various device datasheets within the "*Two-Wire I2C Serial Interface*" section.

SPI Mode

The SPI interface is only available on the MAX41470 device. Similar to the I2C devices, the SPI interface allows access to the internal registers of the receiver. This permits the user to have full control over the same properties of over the receive frequency, data rate, thresholds, bandwidths, etc.

The MAX41470 device allows for a 3-wire read/write interface. The transaction is defined by the CSB low cycle followed by the SCLK transitions for the data on the SDIO pin. A full description of the SPI interface can be found in the *Serial Peripheral Interface (SPI)* section of the MAX41470 device data sheet.

Shutdown, Standby, and Program Modes

When communicating with a MAX41470 device or the MAX41473/74 device in I2C mode, the part can be programed to power-down into one of three low-current, non-receiving states after completing a transmission: Shutdown, Sleep, and Standby modes.

Shutdown is the lowest-current power-down state and is the default condition for all devices. No programming of the device can happen in the Shutdown state, and any previous programming is lost. Sleep allows the interface to be awake, but without the internal clocks running. Standby allows the receiver to start up quicker than shutdown by keeping the crystal oscillator circuit running. Standby is the proper state to configure the device for operation.

To set the power-down state, drive the PWRDN pin high. To transition to Sleep state, drive the PWRDN pin low. To transition to the Standby state, program the STATE_CTRL1 (0x14) EN_XO bit to a 1b.

Chip ID

Register PART_NUM (0x1E) provides a readable identification number for the device part number. With the EN_XO='1', the MAX41470 device ID value will be reported as 0x70. When communicating with a MAX41473 device in I2C mode, the ID value will be reported as 0x73, and similarly the MAX41474 would be 0x74.

The REV_NUM register (0x1F) provides a readable revision number for the device. The default value for this register is 0x02.

MAX4147X EV Kit Bill of Materials

PART	QTY	DESCRIPTION
C1	1	100pF ± 5% Capacitor (0402) Murata GRM1555C1H101JA01
C3	1	0.01µF ± 5% Capacitor (0603) Murata GRM1885C1H103JA01
C5, C6	1	5pF ± 0.25pF Capacitor (0402) Murata GRM0225C1H5R0CA03
DATA	1	Test Point Keystone 5014
SDIO/RSSI	1	Test Point Keystone 5126
GND, GND1	2	Test Point Keystone 5011
3V NOM	1	Test Point Keystone 5010
H1	1	Connector Male Through Hole Sullins PRPC016SFAN-RC
H2	1	Connector Male Through Hole Sullins PRPC012SFAN-RC
J2/LNA_ INPUT	1	Connector End Launch Johnson Components 142-0701-851
JU1, JU5	2	Connectors Male Through Hole Sullins PEC03SAAN
JU2	1	Connectors Male Through Hole Sullins PEC09SAAN
JU3	1	Connectors Male Through Hole Sullins PBC02DAAN
SU1–SU6	6	Test Point Jumpers Sullins STC02SYAN
R1-R4, R10-R12, R19	8	0Ω ± 0% Resistor (0603) Vishay Dale CRCW06030000Z0
R8, R9	2	4.7kΩ ± 5% Resistor (0603) Panasonic ERJ-3GEYJ472V
U1	1	MAX41470GTC+, MAX41473GTC+, MAX41474GTC+
Y1	1	16MHz Crystal Epson TSX-3225 16.0000MF18X-AC0

Component Suppliers

SUPPLIER	PHONE	WEBSITE
Epson America	562-290-4677	www5.epsondevice.com
Johnson Components/Cinch		belfuse.com/cinch
Keystone	800-221-5510	www.keyelco.com
Murata Electronics North America, Inc.	770-436-1300	www.murata.com
Panasonic		
Sullins	760-744-0125	www.sullinscorp.com
Vishay Dale	800-433-5700	www.vishay.com

Note: *Indicate that you are using the MAX4147X when contacting these component suppliers.*

Ordering Information

PART	TYPE
MAX41470EVKIT#	EV Kit
MAX41473EVKIT#	Preset / I²C – ASK
MAX41474EVKIT#	Preset / I²C – FSK

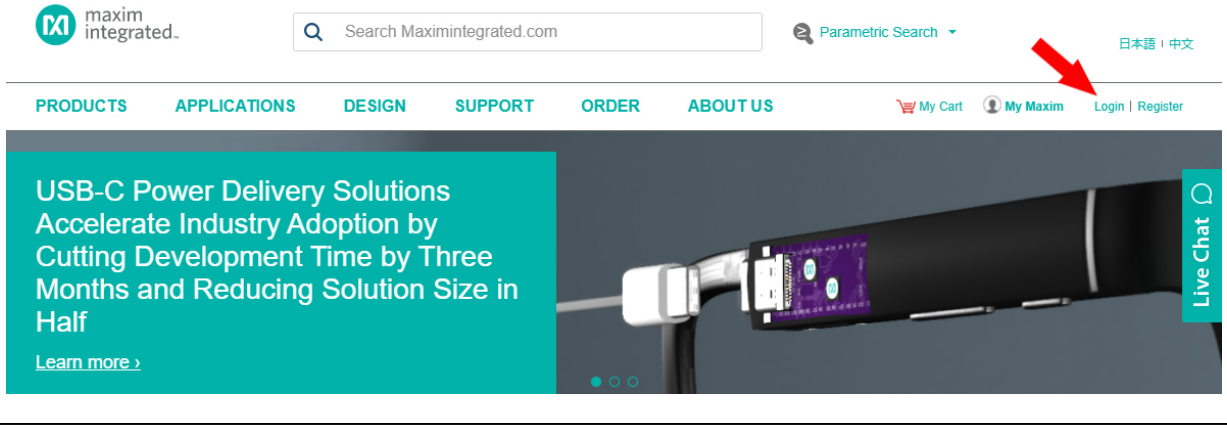
#Denotes RoHS compliant.

Appendix I – Detailed Software, Firmware, and Driver Installation Procedures

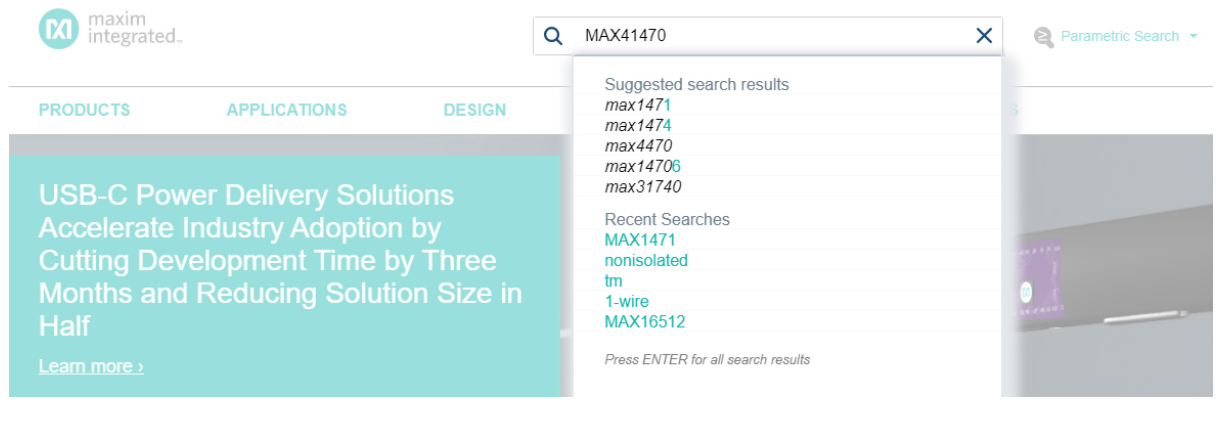
Download the ISM Radios GUI

This software and firmware are available from the **Maxim website**.

1) Log in to your MyMaxim account on the website.

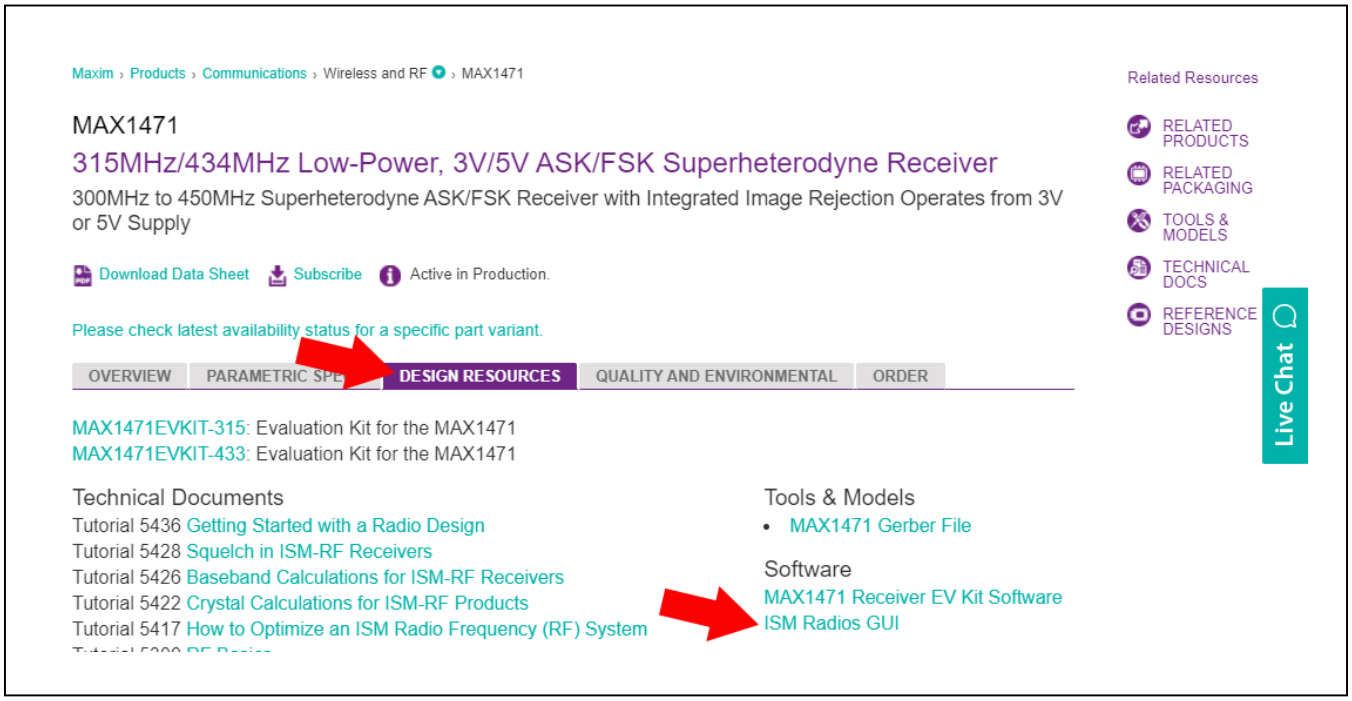


2) Click on the magnifying glass and search for the **MAX41470**, **MAX41473** or **MAX41474.**

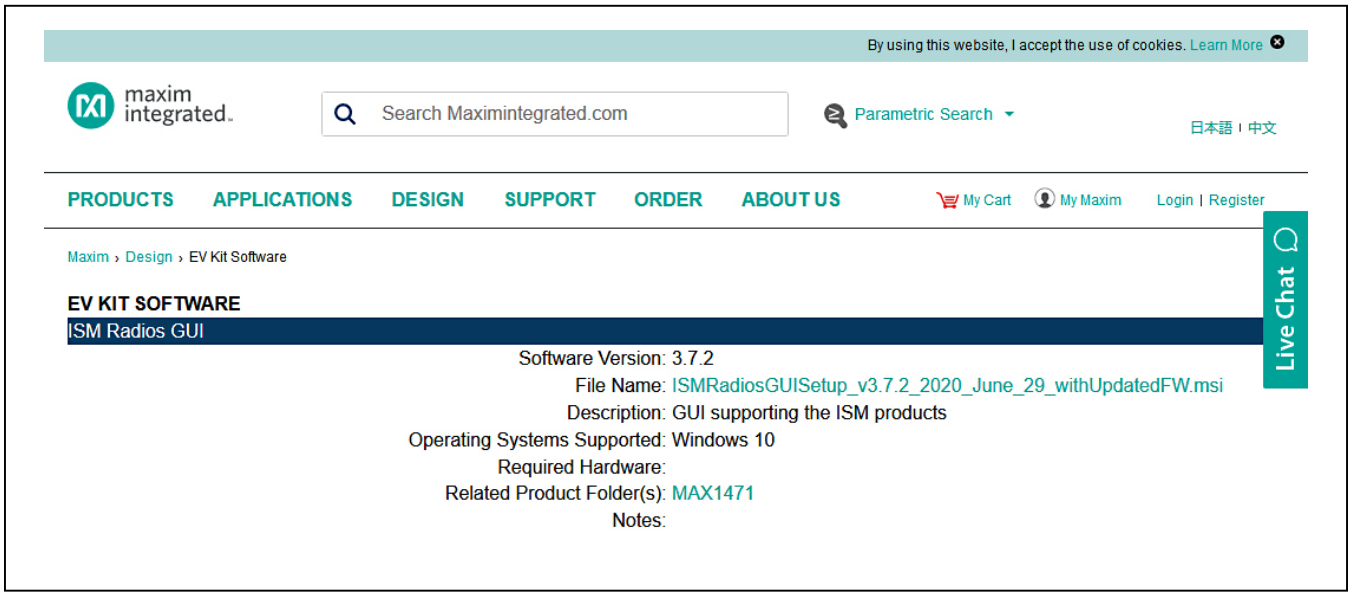


3) Click on the **Design Resources** tab on the appropriate product web page.

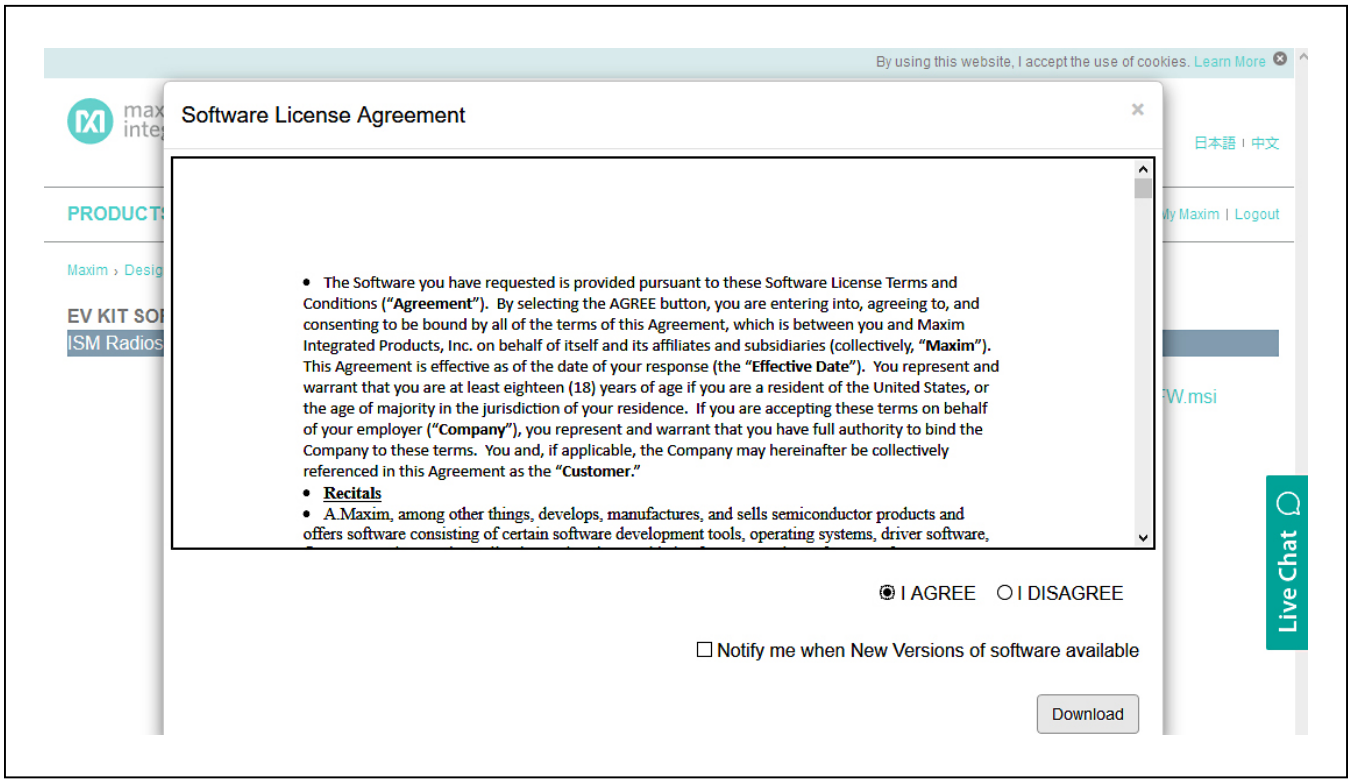
4) Click on the appropriate software link.



5) Click the file link on the software landing page to download the MAX4147X EV kit package.



6) Review the Maxim Software License Agreement (SLA) and accept the terms by clicking on the **Accept** button.



7) Save the EV kit distribution package to your desktop or other accessible location for later install.

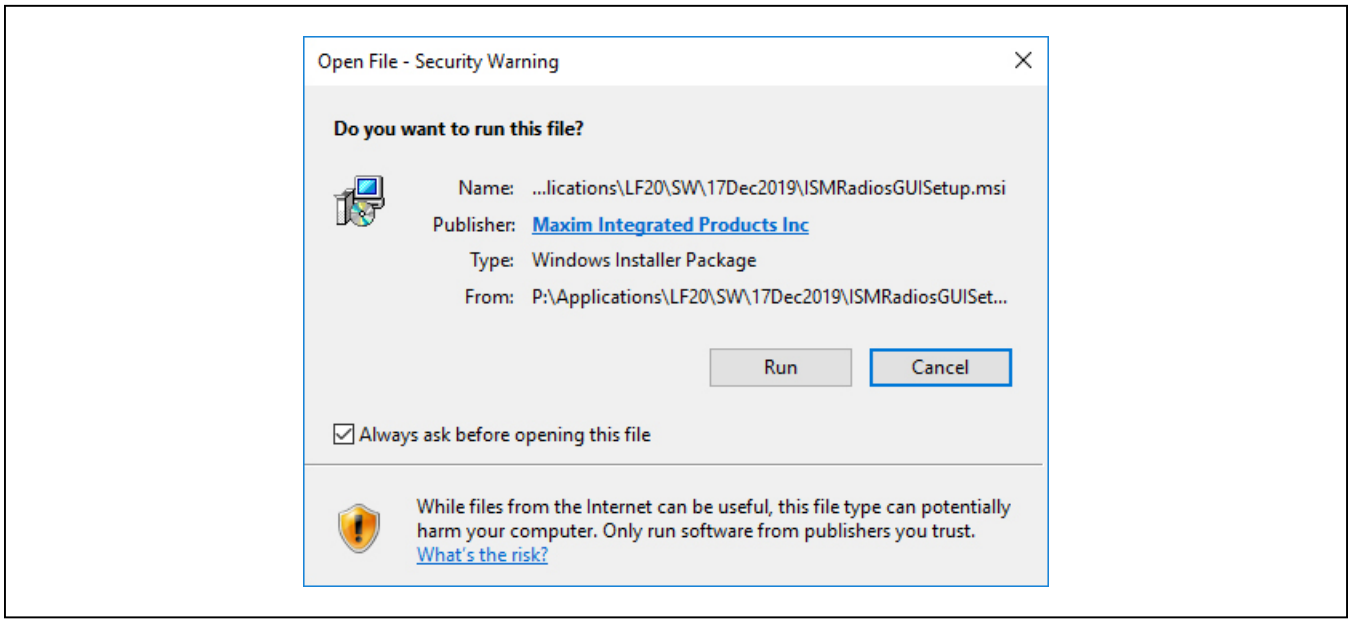
Install the ISM Radios GUI

This software and firmware are available from the **Maxim website**. See the *Download the ISM Radios GUI* section for information on obtaining the latest firmware from Maxim.

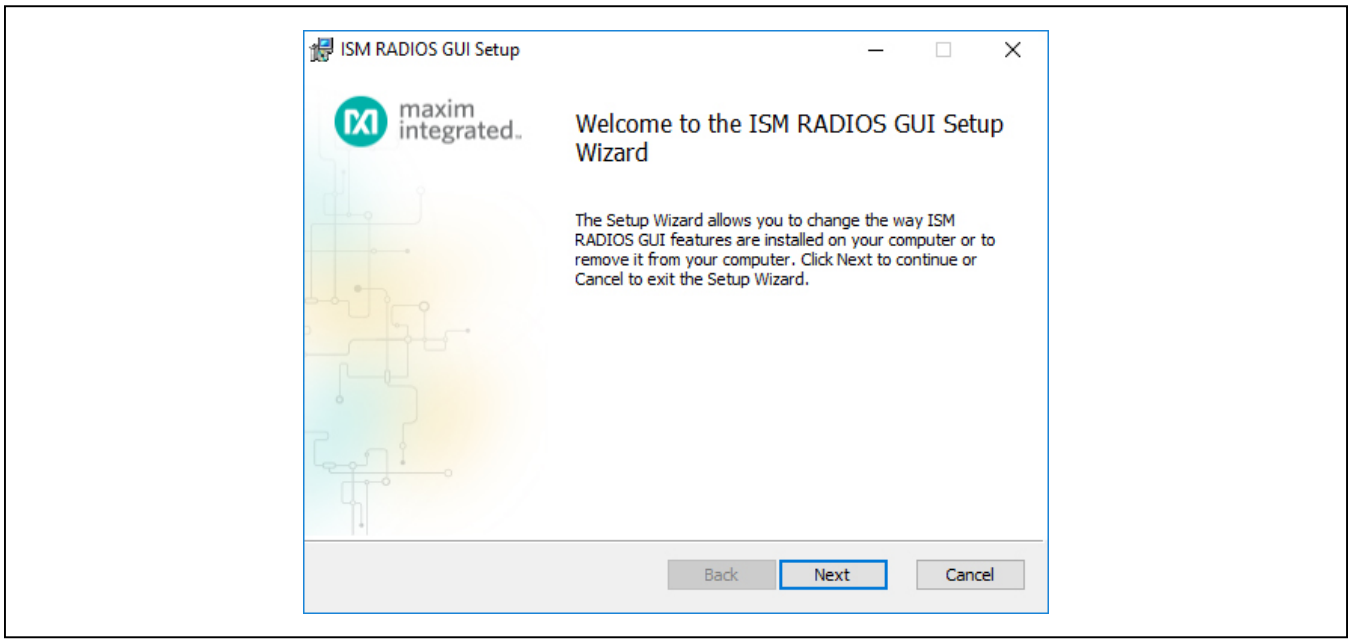
This process should take less than **10 minutes** after downloading the software, firmware, and driver package.

1) Double-Click the ISMRadiosGUISetup.msi setup file and follow the Setup Wizard prompts.

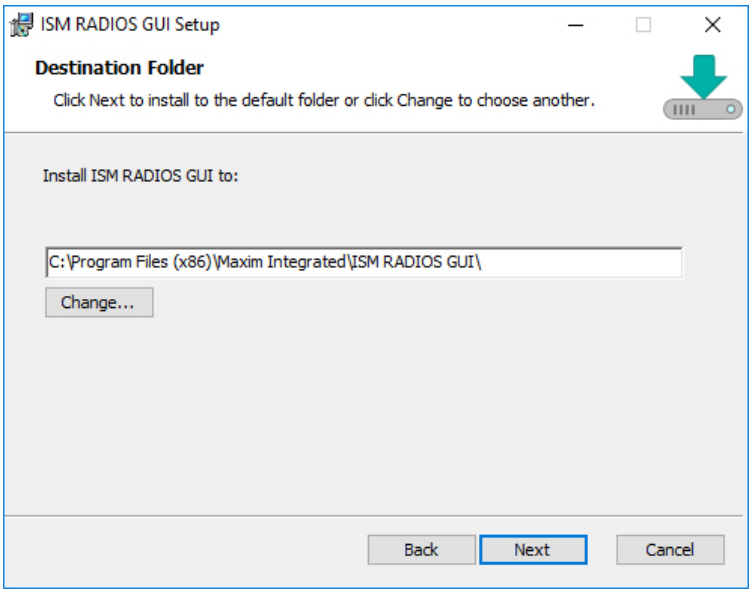
   a. If Security Warning appears, click **Run**.



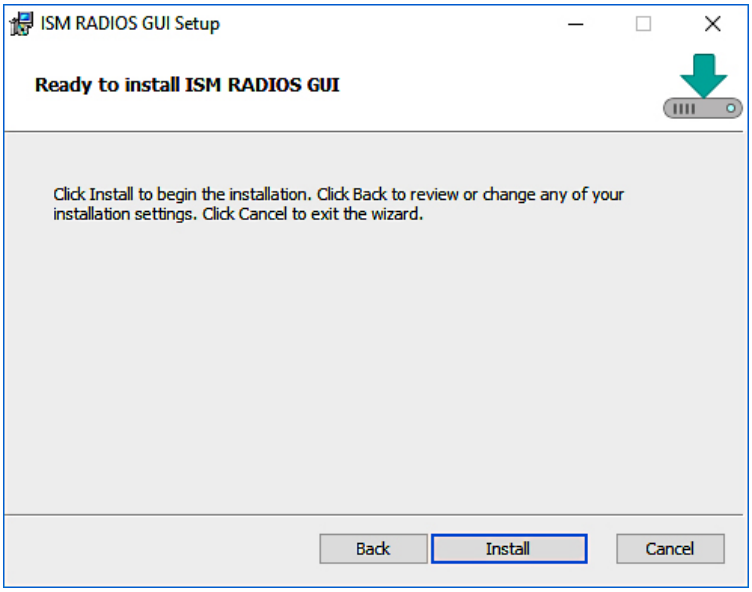
   b. Click **Next**.



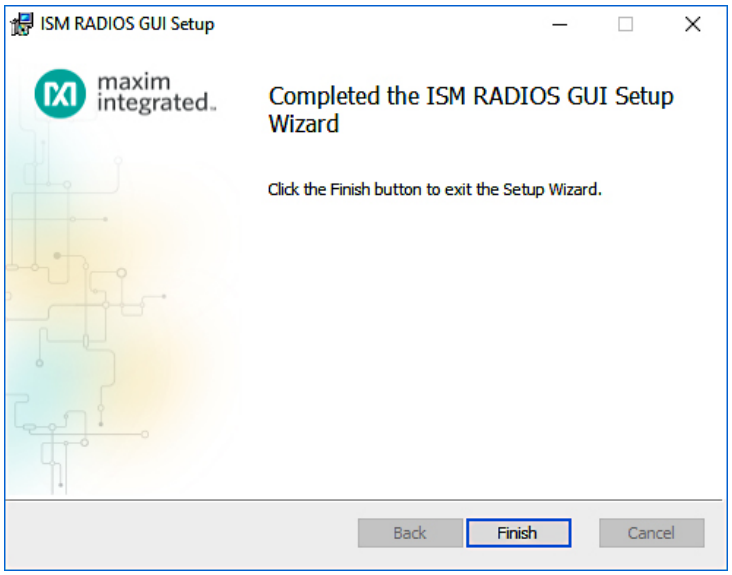
c. Use the default **Destination Folder** and click **Next**.



d. Install the software by clicking **Install**.



e. Click **Finish** when the setup process is complete.

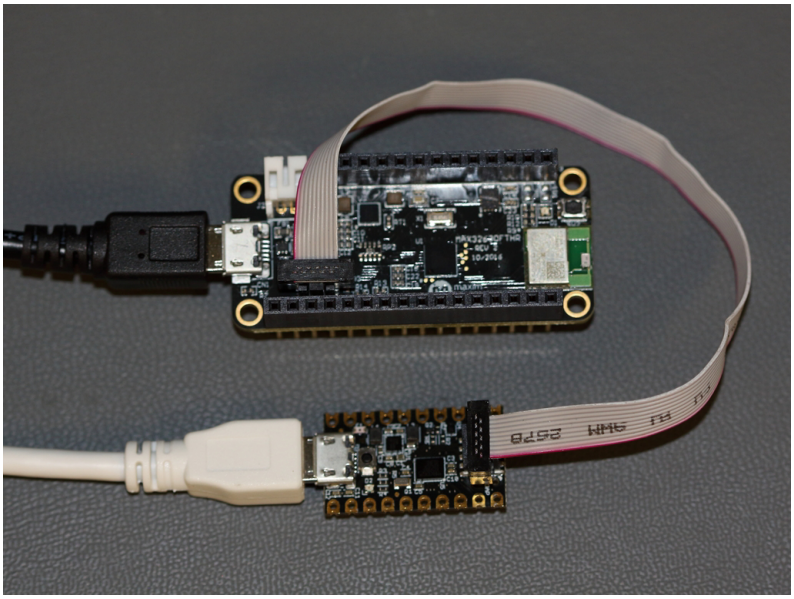


Program the MAX32630FTHR Board with the MAX4147X Firmware

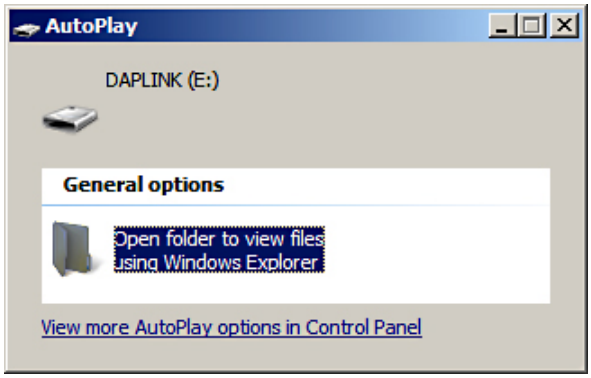
This software and firmware are available from the **Maxim website**. See the *Download the ISM Radios GUI* section above for information on obtaining the latest firmware from Maxim.

1) Connect the MAX32630FTHR to the MAX32625PICO.

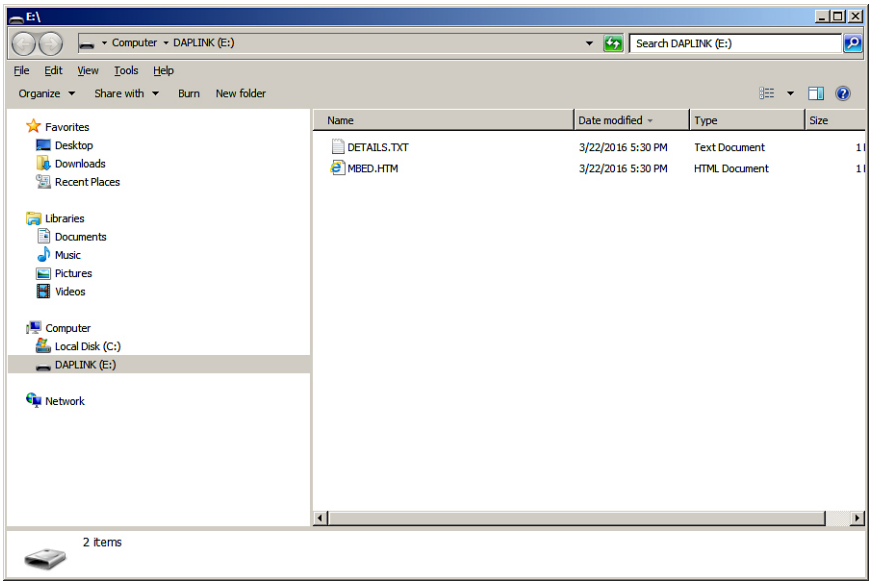
a. Use the fine-pitch 10-pin ribbon cable to connect the boards from the SWD (J3) header on the HDK to J4 on the MAX32630FTHR.



2) Connect the MAX32630FTHR to a power source.
   a. Use a USB Micro-B cable to connect the MAX32630FTHR board to a suitable power source (no USB connectivity is required). Alternatively, you can power the board from a charged battery as long as you remember to turn it on by pressing the power/reset button next to the battery connector. The board turns on automatically when powered from the USB supply.
   b. The status LED on the FTHR board should be lit a steady red.
3) Connect the MAX32625PICO to a PC.
   a. Use a USB Micro-B cable to connect the HDK to a PC, through the connector marked HDK (The white USB cable in the photos).



   b. The status LED on the DAPLINK board will blink red when connecting.
   c. After a few seconds of activity, the PC will recognize the DAPLINK as a standard USB drive.
4) Drag-and-drop or save a the ism_radios_fw.bin program binary to the mbed or DAPLINK USB drive.



a. The FTHR board LED will shut off and the LED on the MAX32625PICO will slowly flash red as the FTHR board is being programmed.

b. Once the programming is complete, the DAPLINK USB drive will disconnect from the PC and reconnect as a USB drive again.

c. If the programming was successful, the contents of the DAPLINK USB drive should include a DETAILS.TXT file. If an ERROR.TXT file exists on the drive, check that the FTHR board had power during the programming process and repeat steps 3 and 4.

5) To ready the FTHR board for use, disconnect the DAPLINK board (ribbon cable) and press the **Reset** button on the FTHR board or disconnect the FTHR board from the USB power supply.

a. When the **Reset** button is pressed, the microcontroller will restart and the newly programmed application will begin to run, or you can disconnect and reconnect the USB cable if using a PC for power.

The latest information and these firmware update instructions can be found on the **MAX32630FTHR board mBed web site** or by visiting the **mBed home page** and searching for “MAX32630FTHR”.

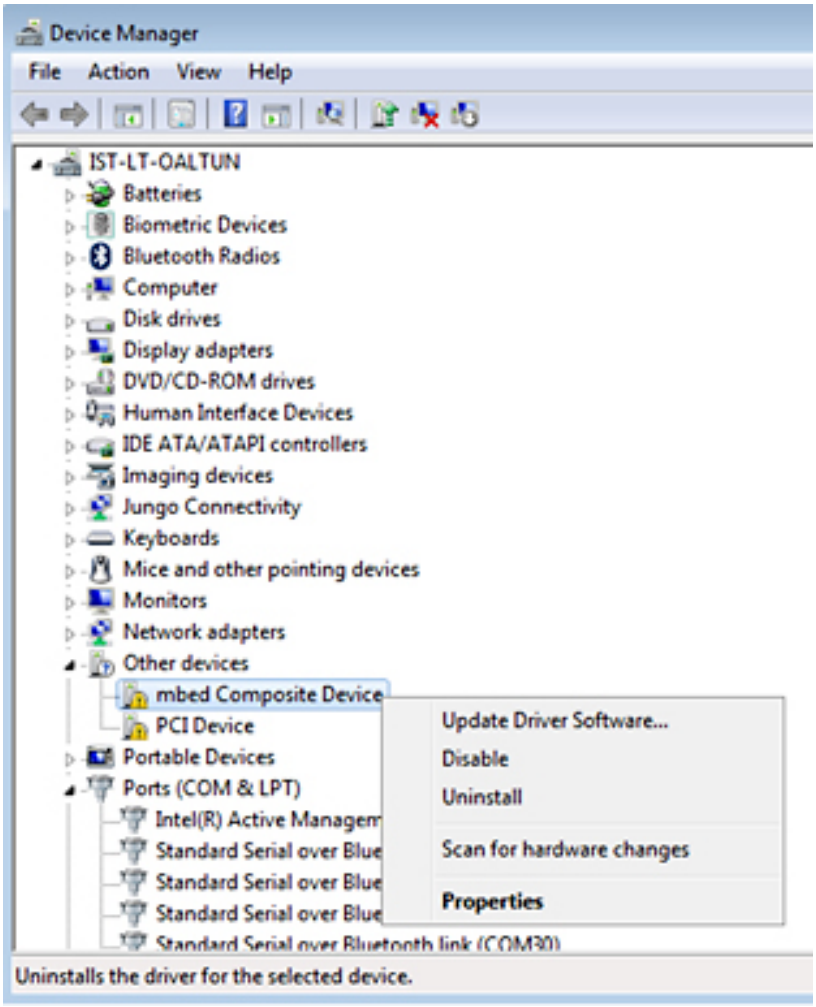
If you do not have an mbed account, choose **Signup** and create your mbed account. Otherwise, log in with your normal username and password. This will give you access to the website, tools, libraries, and documentation.

You must load the matching HDK image (**MAX32630FTHR DAPLINL image**) for the platform you are programming in order for drag-n-drop programming to work.

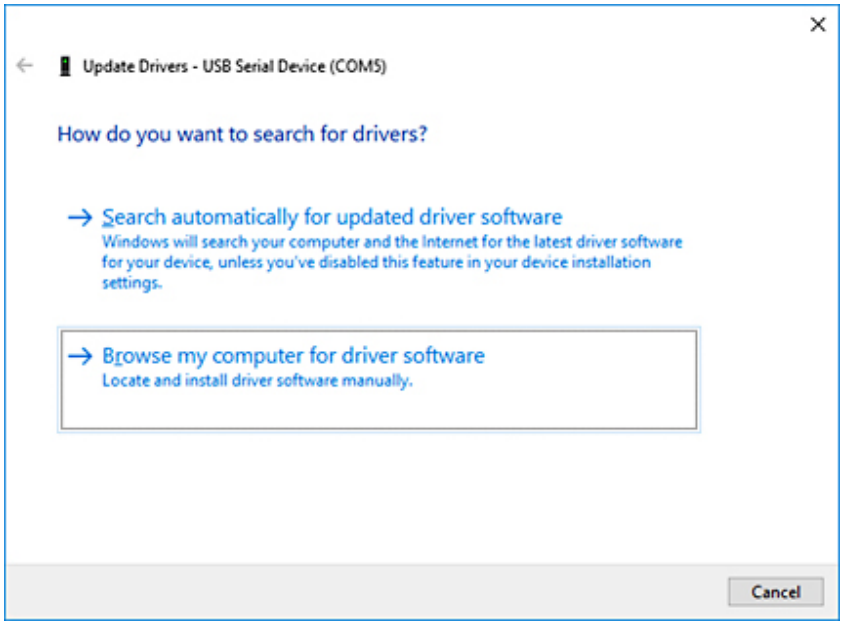
Update the MAX32630FTHR Board Driver

The required driver is available from the **Maxim website**. See the *Download the ISM Radios GUI* section in this documentation for information on obtaining the latest driver from Maxim.

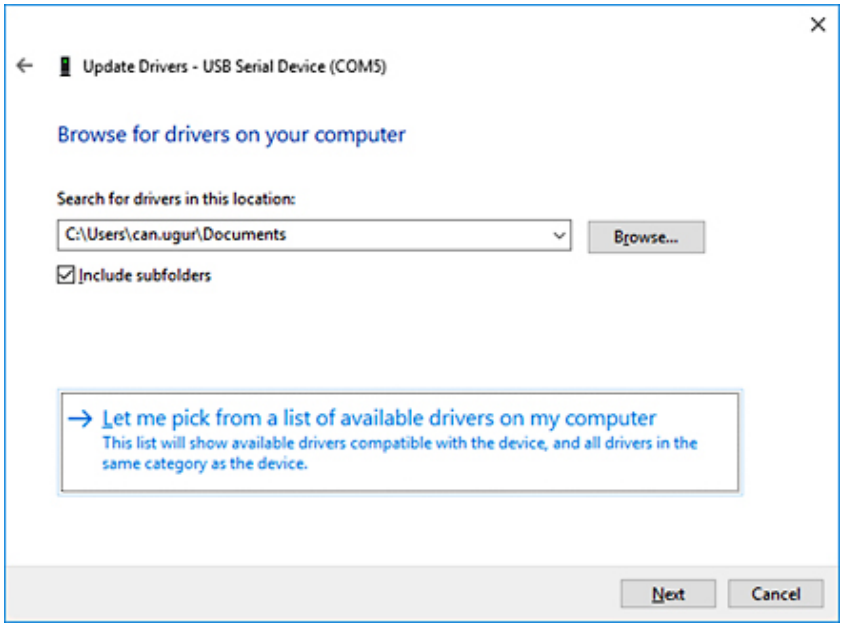
1) Connect the MAX32630FTHR to the PC's USB port.
2) In **Device Manager**, right-click **Other devices** => **CDC Device** or **mbed Composite Device**.



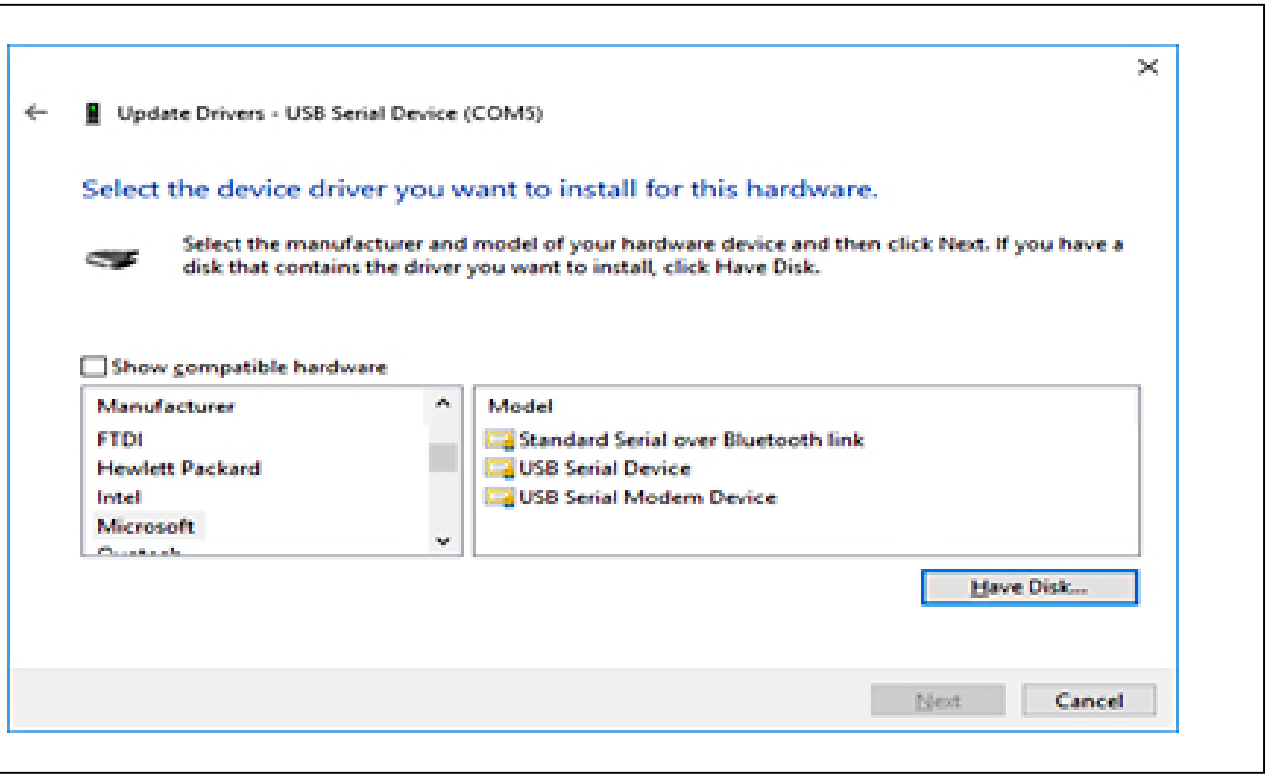
3) Click **Update Driver Software** then select **Browse my computer for driver Software**.



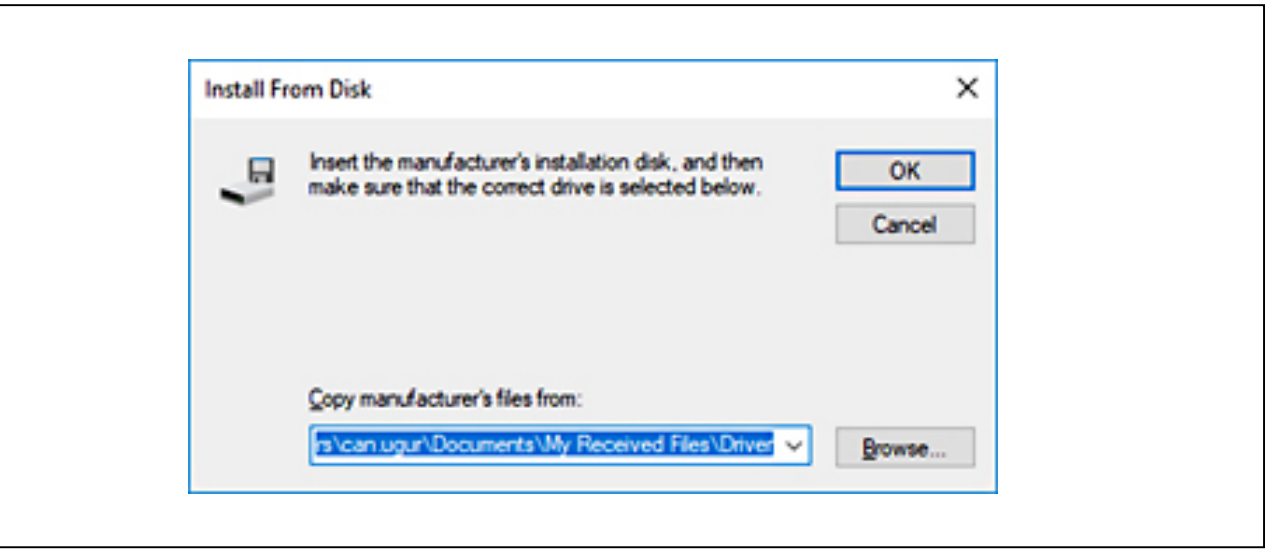
4) Select **Let me pick from a list of available drivers on my computer**.



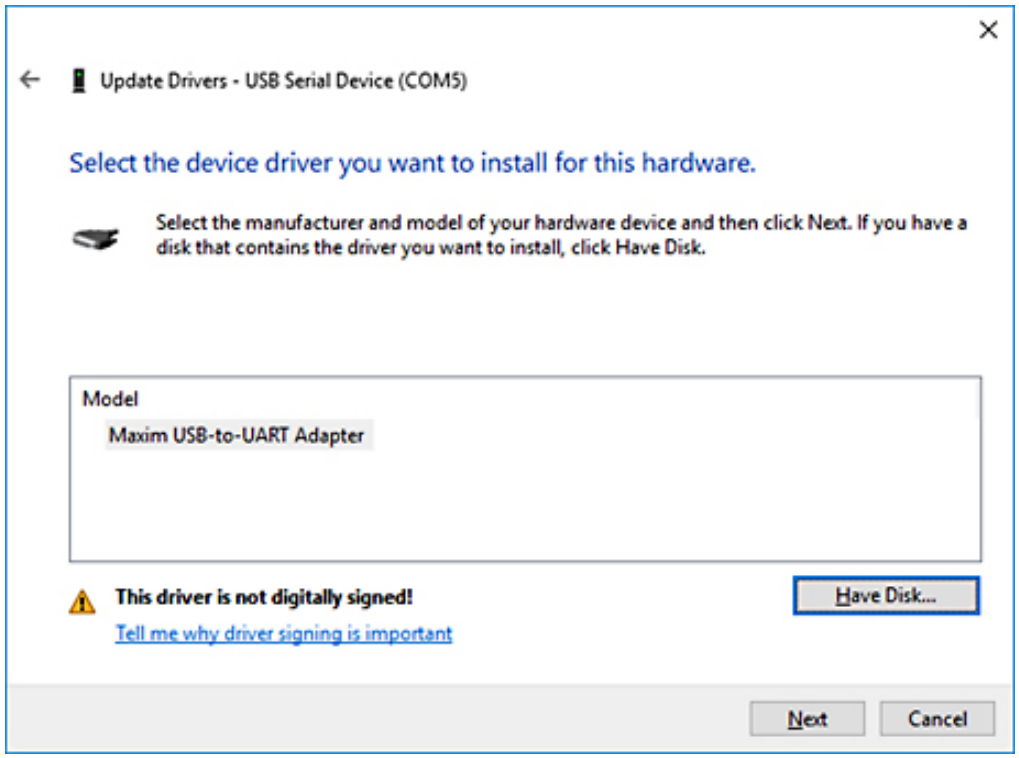
5) On a Windows 10 operating system, click the **Have Disk…** button.



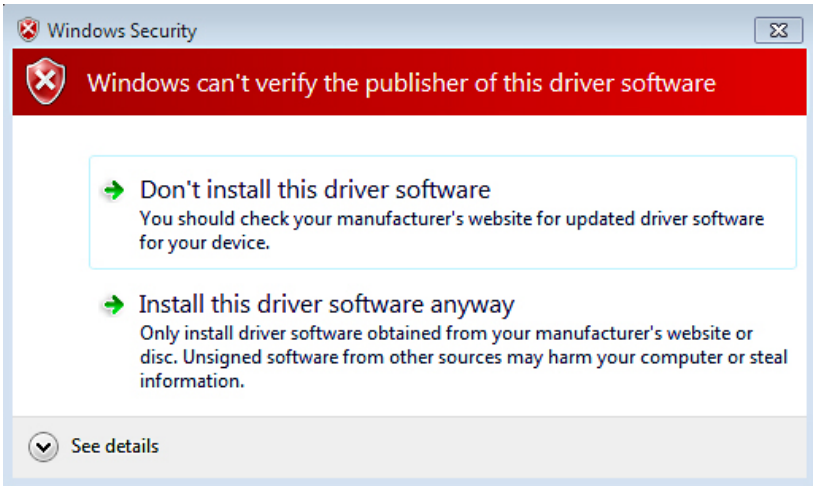
6) Browse to the path of the driver folder and click **OK**.



7) Click **Next**.



8) Ignore the warnings and click **Install this driver software anyway**.



Win 10: Unverified publisher warning

Appendix II – Hardware Modifications

I²C Pull-Up Resistors

To accommodate the various operating modes of the MAX4147X products on one board, the shared digital pins have many interface modes over which they operate.

To properly establish the open-drain topology of the two-pin serial interface the user must have the SDA and SCL lines pulled-up to the supply voltage. The resistor footprints: R8 and R9 are provided on the MAX4147XEVKIT for this purpose. By connecting JU3 (1-2 and 3-4) both lines will be pulled-up to the VDD supply.

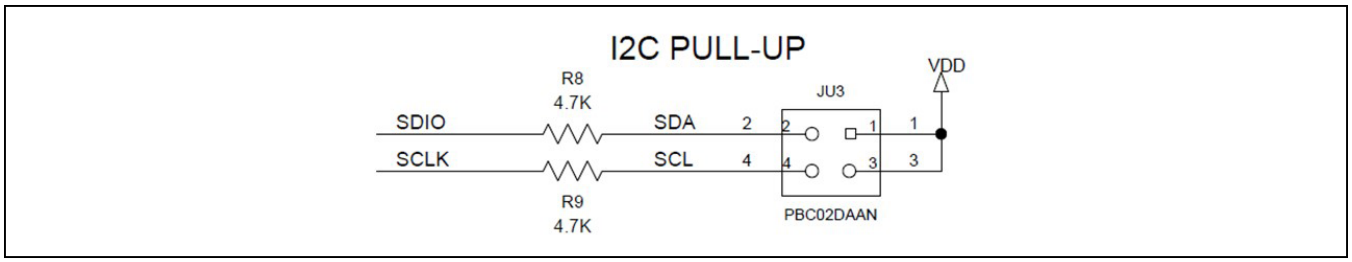


Figure A2-1. I²C Pull-Up Resistors

These pull-ups should only be connected when the I2C/PRESET pin is connected to VDD (JU2 1-2), thus selecting the I²C interface mode of the MAX4147X.

It should be noted that the FTHR board also has footprints for I²C pull-up resistors at R6 and R11. Both sets of pull-up resistors (on the FTHR board and the MAX4147XEVKIT) should not be populated simultaneously, otherwise incorrect I²C signal levels may result. Likewise, if other I²C slaves are added to the bus, only one set of pull-up resistors should be used.

PMOD Header Interface

The MAX4147X EV kit provides a PMOD-compatible header footprint, which provides yet another built-in interface to the transmitter. The JU4 connector can be populated with a 6-pin, 100mil, right-angle header such as a SAMTEC TSW-106-25-T-S-RA, allowing direct connections to the CSB/I2C, SDIO/RSSI, SCL, SCLK/SDA, ground, and VDD lines.

The PMOD interface can be used in combination with the Maxim MAX32600MBED kit and the MAXREFDES72# Arduino Uno R3-to-PMOD shield adaptor. When using the PMOD interface to supply the MAX4147X EV kit with power, be sure to connect the JU1 jumper between pins 2-3.

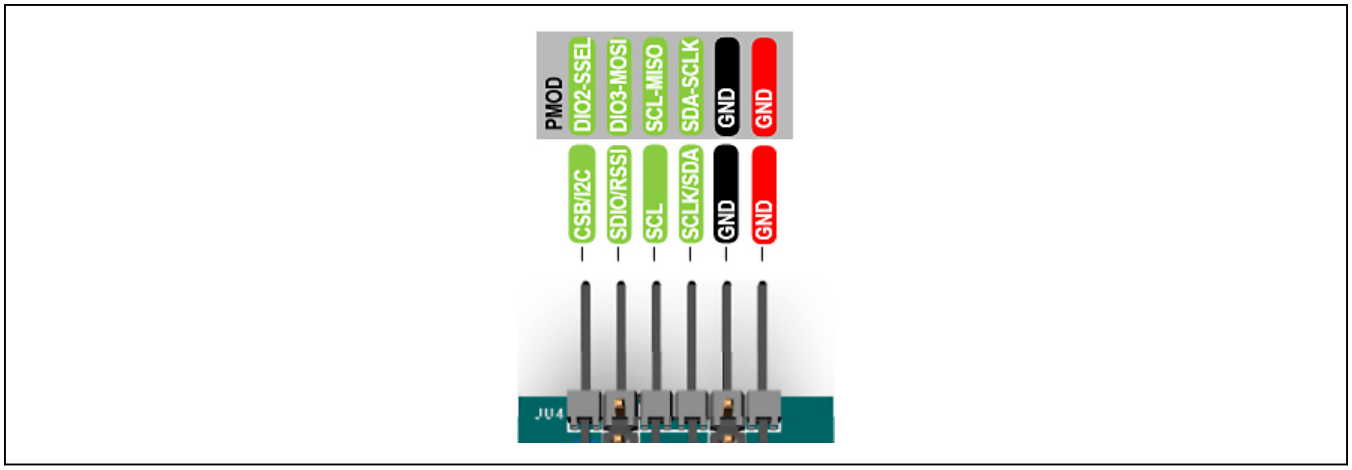
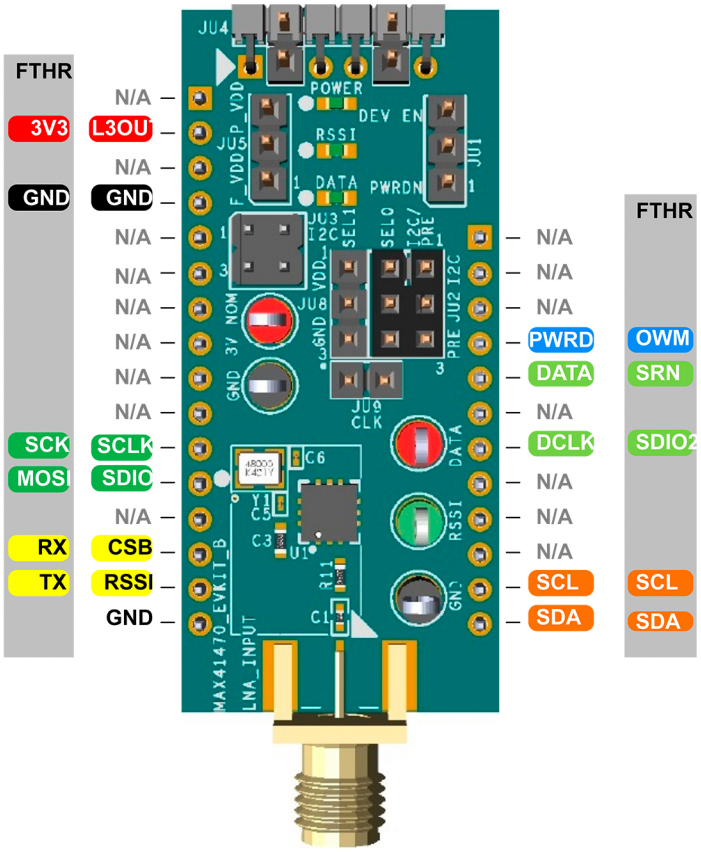


Figure A2-2. MAX4147XEVKIT PMOD Interface

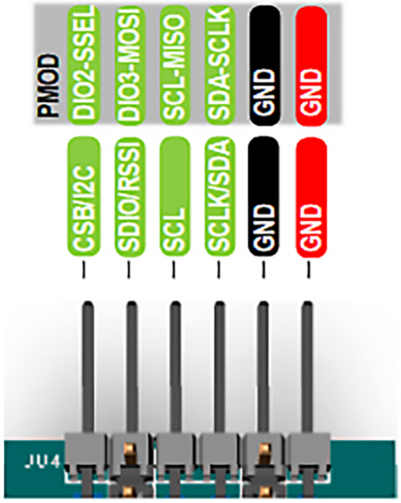
Appendix III – Pinout Sheets

MAX4147X EV Kit

FTHR Board Connectors

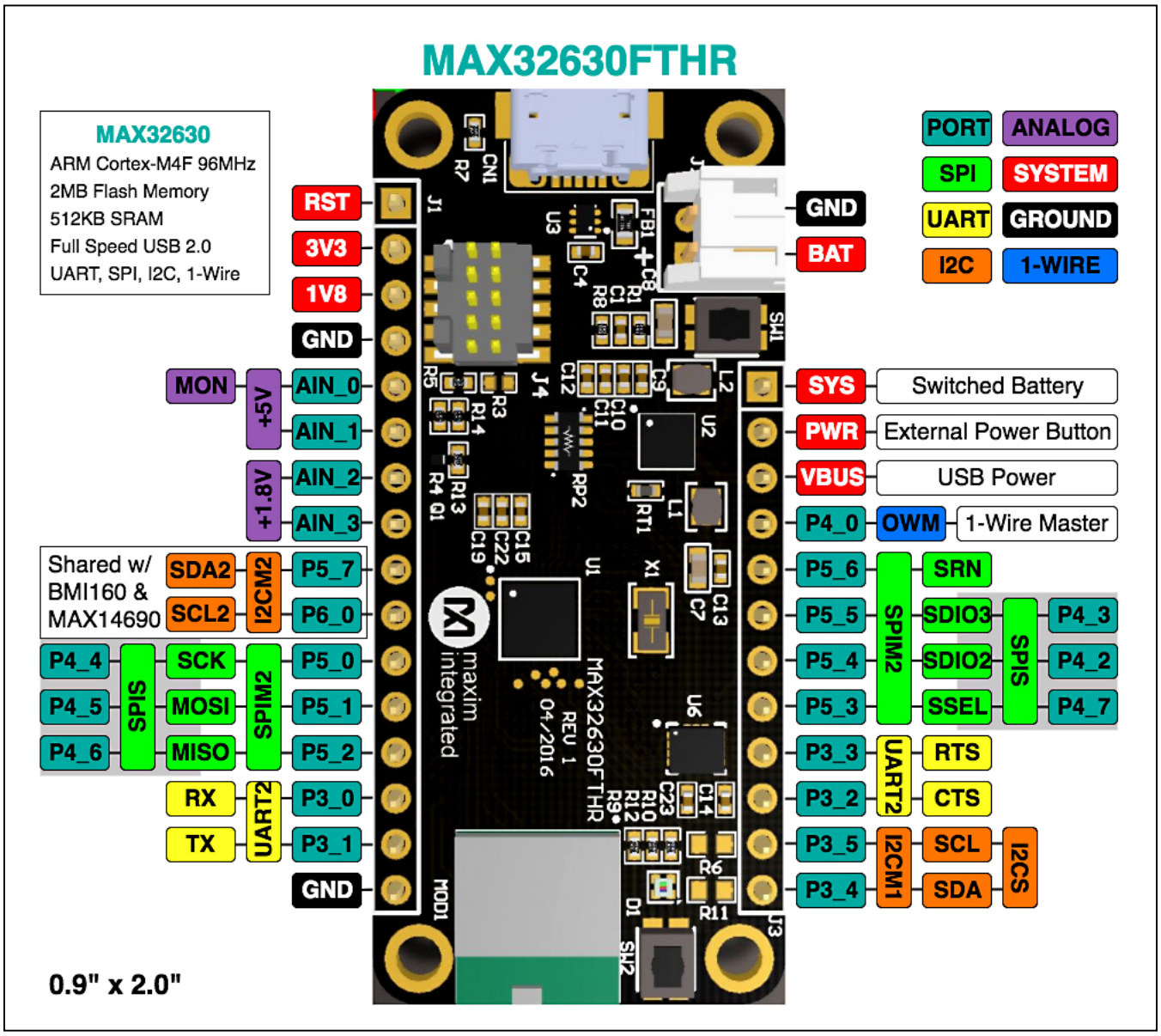


Peripheral Module Connector

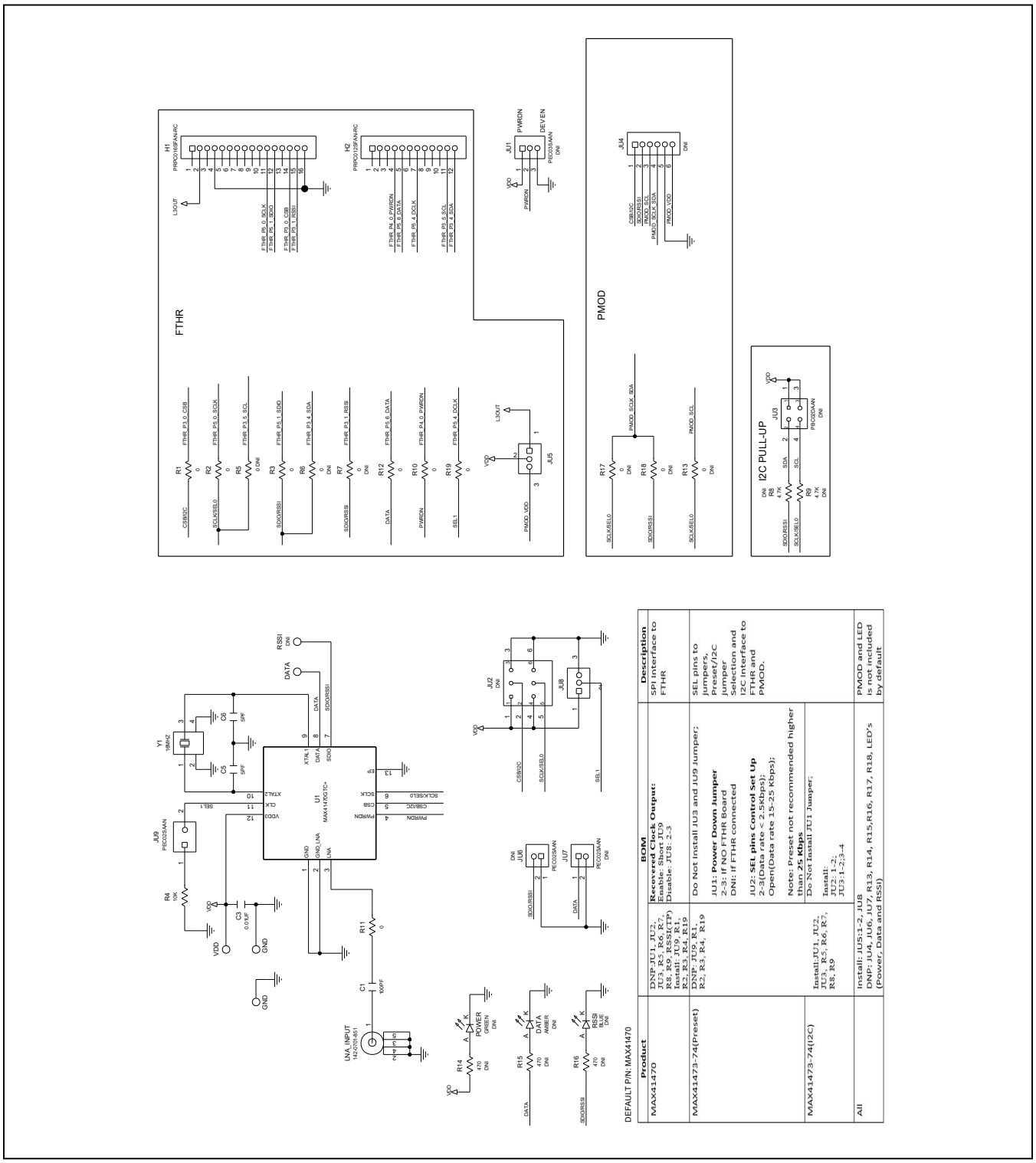


MAX32630FTHR

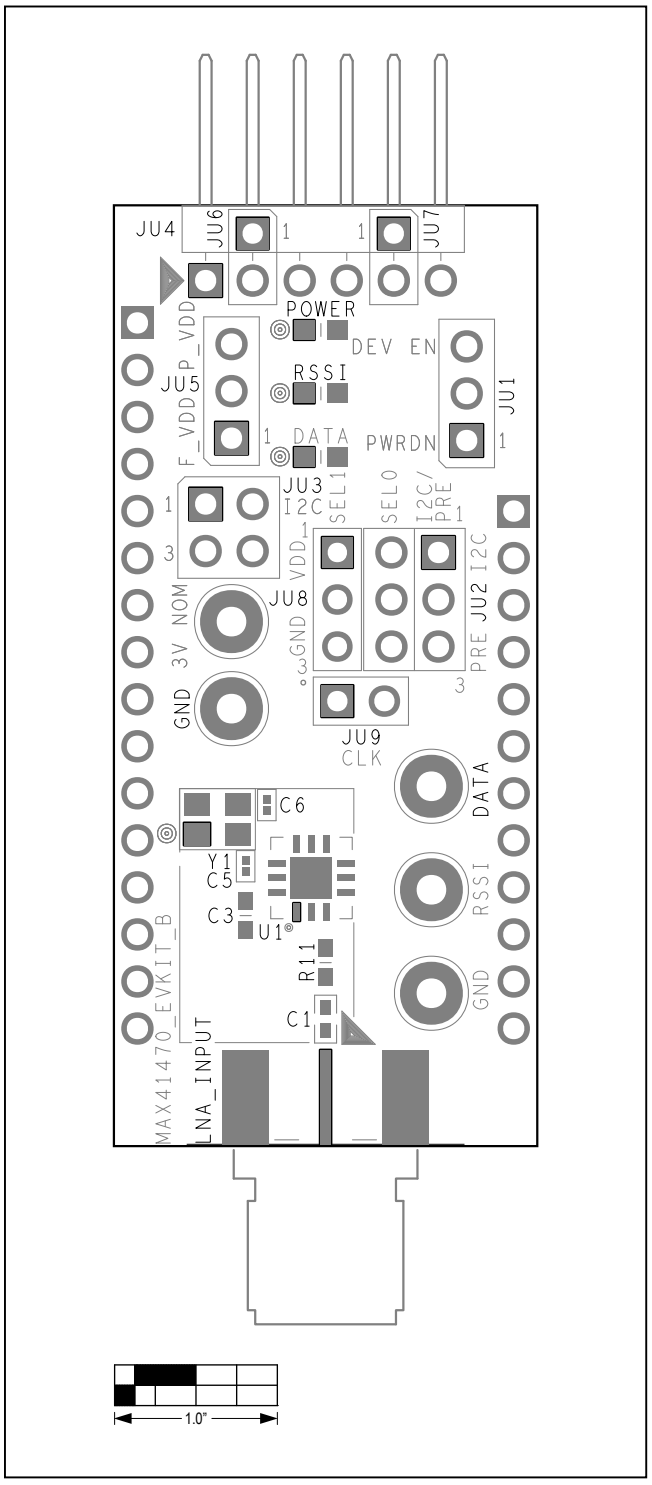
Arm Cortex-M4F microcontroller rapid development platform.



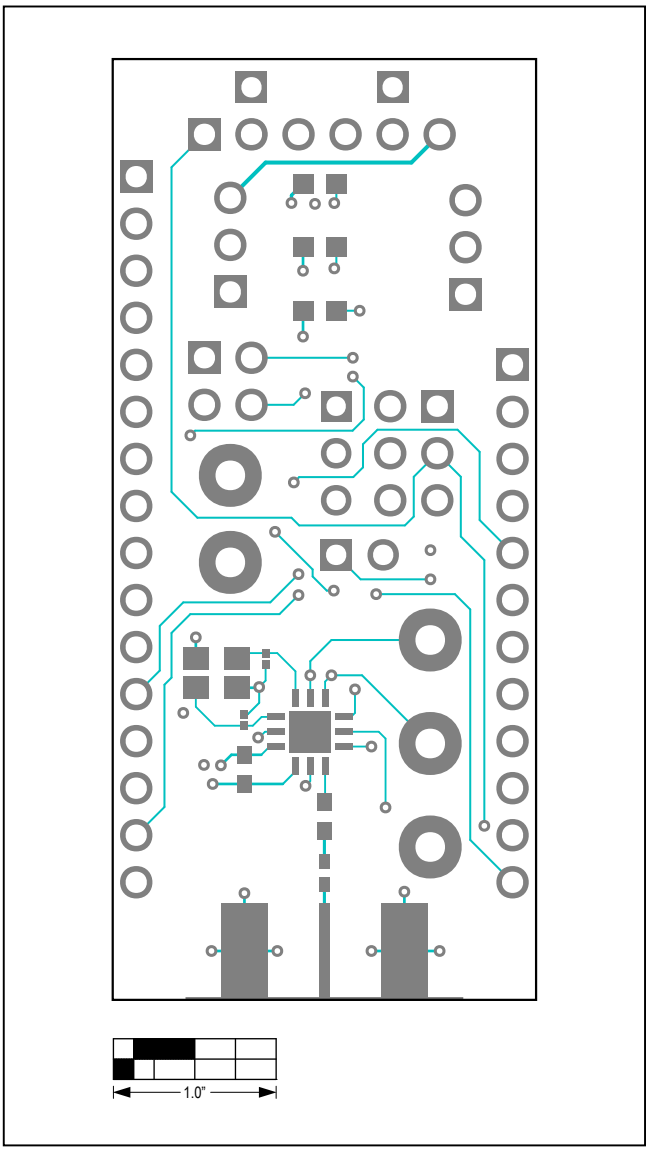
MAX4147X EV Kit Schematic Diagrams



MAX4147X EV Kit PCB Layout Diagrams

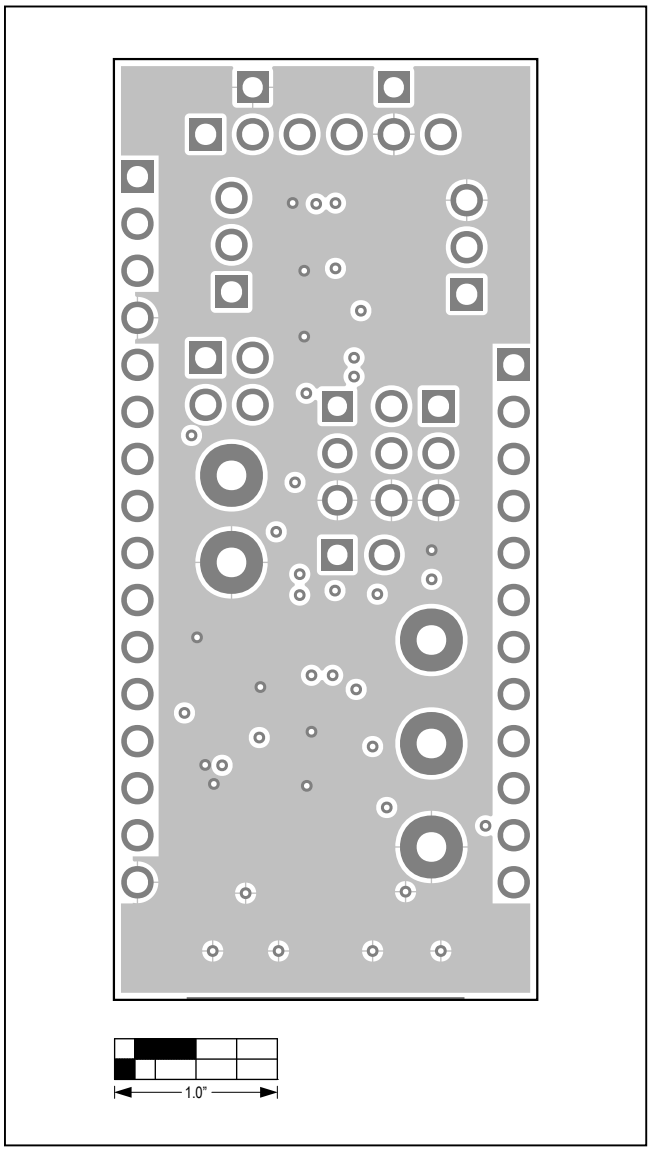


MAX4147X EV Kit PCB Layout—Silkscreen Top

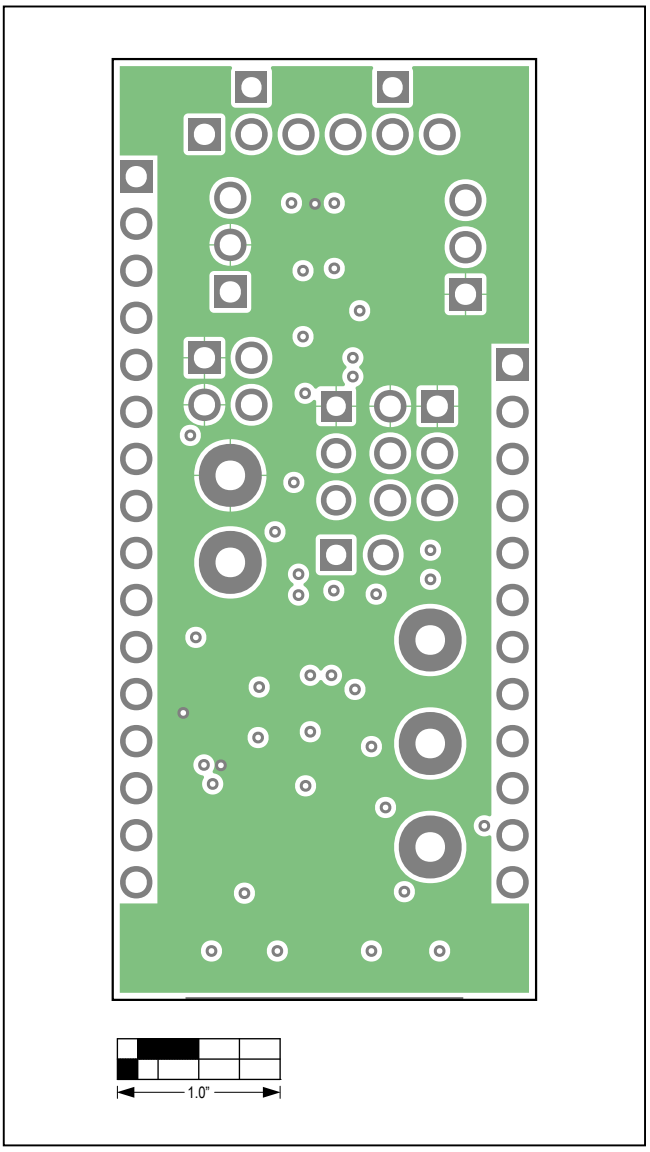


MAX4147X EV Kit PCB Layout—Top Layer

MAX4147X EV Kit PCB Layout Diagrams (continued)

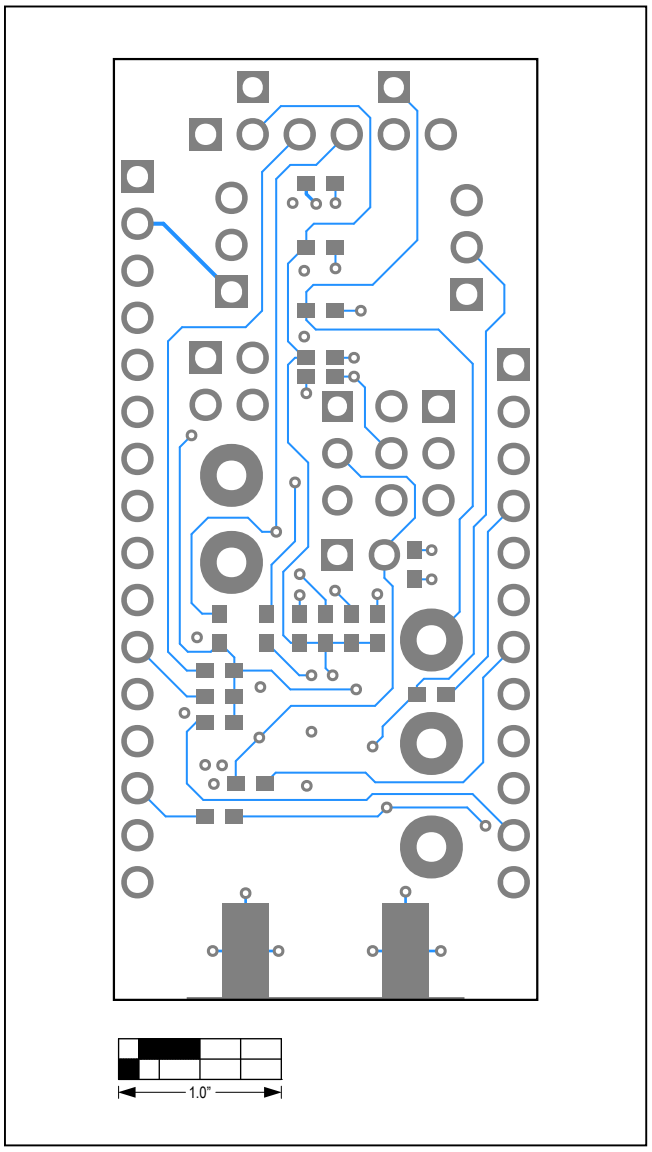


MAX4147X EV Kit PCB Layout—Internal 2

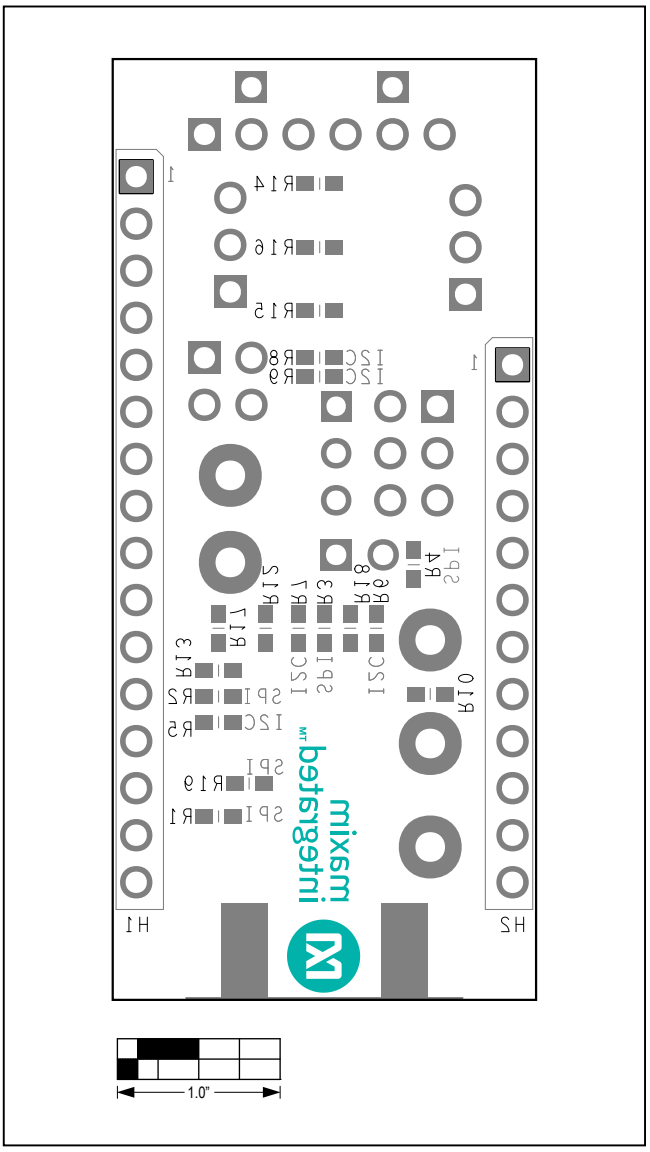


MAX4147X EV Kit PCB Layout—Internal 3

MAX4147X EV Kit PCB Layout Diagrams (continued)



MAX4147X EV Kit PCB Layout—Bottom Layer



MAX4147X EV Kit PCB Layout—Silkscreen Bottom

Revision History

REVISION NUMBER	REVISION DATE	DESCRIPTION	PAGES CHANGED
0	9/20	Initial release	—
1	10/20	Added MAX41473 and MAX41474	All

For pricing, delivery, and ordering information, please visit Maxim Integrated's online storefront at https://www.maximintegrated.com/en/storefront/storefront.html.

Maxim Integrated cannot assume responsibility for use of any circuitry other than circuitry entirely embodied in a Maxim Integrated product. No circuit patent licenses are implied. Maxim Integrated reserves the right to change the circuitry and specifications without notice at any time.