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

MAX41470 Evaluation Kit

Evaluates: MAX41470

General Description

The MAX41470 evaluation kit (EV kit) contains a single MAX41470 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 EV kit operates 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.

The EV kit includes Windows® 10-compatible software that provides a simple GUI for configuration of all the MAX41470 registers through the SPI port. The GUI also controls the on-board PMIC when the MAX32630FTHR applications platform is used.

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

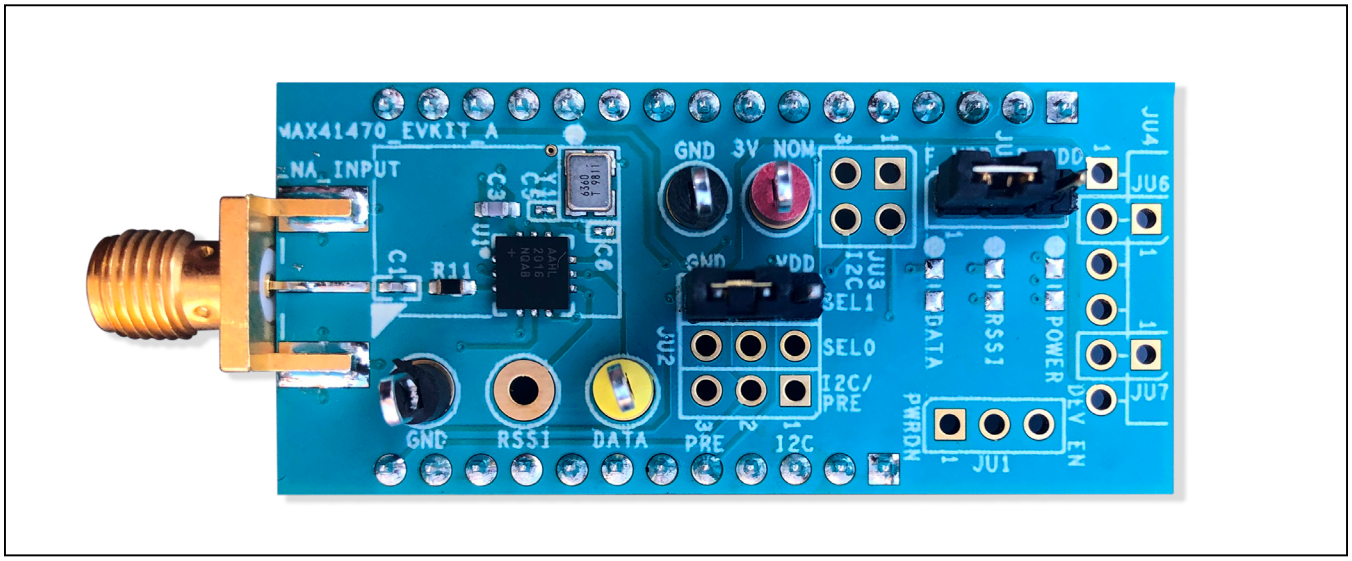
Features and Benefits

- Evaluates the MAX41470 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
- GUI Controls for MAX32630FTHR Board PMIC
  - Operation from 1.8V to 3.3V
- Proven Four-Layer PCB Design
- Fully Assembled and Tested

MAX41470 EV Kit Contents

- MAX41470 EV Kit Board
- MAX32630FTHR# Kit
  - FTHR Board
  - DAPLINK Board
  - Two USB Micro-B Cables

MAX41470 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 of Microsoft Corporation.

maxim integrated™

Quick Start

Required Equipment

- Windows PC* (Windows 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

Software and Drivers

The MAX41470 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 MAX41470 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 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 MAX41470 EV kit is 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 is connected to a local area network and has 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 MAX41470 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. MAX41470 EV Kit Installed Files and Folders

FILE NAME	DESCRIPTION
ISMRadiosGUISetup.msi	Application GUI
MaximStyle.dll	Supporting DLL file for software operation
MAX41470_Registers.xml	Register definition file for MAX41470
MAX4146x_Registers.xml*	Register definition file for MAX4146X
MAX1471_Registers.xml*	Register definition file for MAX1471
MAX41470_QuickStart.xml	Quick start configuration file for MAX41470
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.*

** Required for operation of the MAX41470 EV kit with the GUI software.*
† Required when the FTHR board is not connected to the MAX41470 EV kit.
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 MAX41470 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

FTHR Board Quick Start Procedure

Set up the MAX41470 EV kit and FTHR board hardware for MCU/GUI operation as follows:

1) Verify that all jumpers on the MAX41470 EV kit board are in the default position. (See Table 2.)
2) Connect the MAX41470 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.

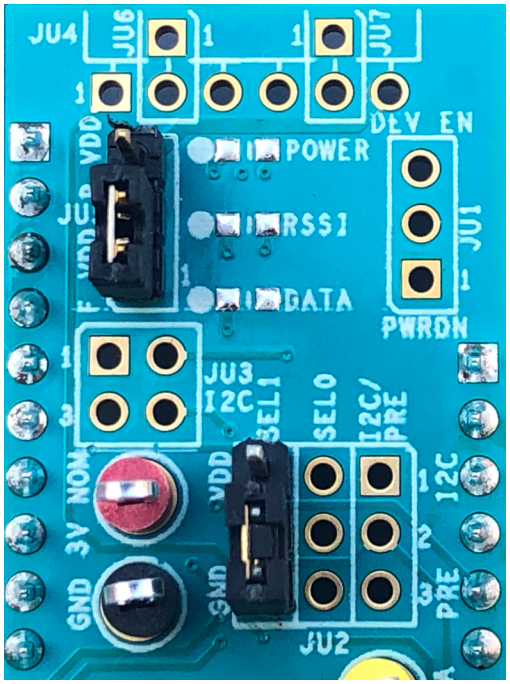


Figure 1. MAX41470 EV Kit Jumpers

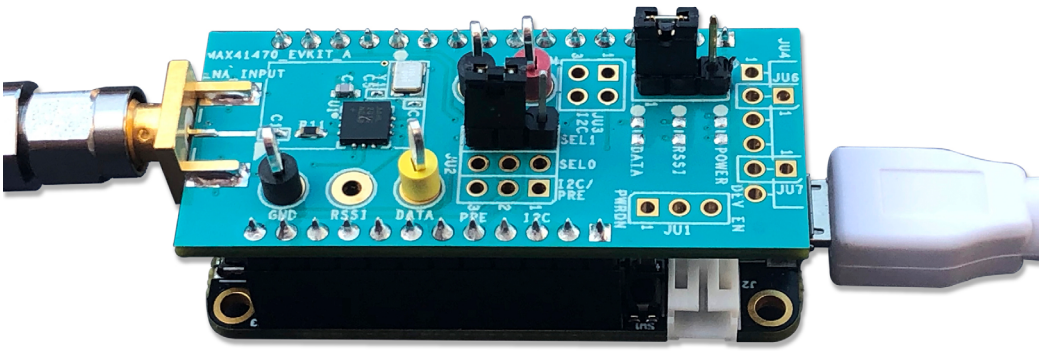


Figure 2. MAX4147X EV Kit Orientation to FTHR Board

Table 2. MAX41470 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 to VDD	I²C mode
	2-3	I2C to GND	Serial interface disabled (or open jumper); refer to the MAX41473/ MAX41474 data sheet for details
	4-5	SEL0 to VDD	Refer to the MAX41473/MAX41474 data sheet for details
	5-6	SEL0 to GND	
	7-8	SEL1 to VDD	
	8-9	SEL1 to GND	
	None Installed**	Open	SPI mode
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)	

** Default position*
*** Default position for MAX41470*
† Setting for MAX41473–MAX41474 in I²C mode
‡ Default for MAX41473–MAX41474 in Preset mode
***Note:** JU4, JU6, and JU7 not installed*

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 the board from the USB connector).

4) Connect the input signal to the LNA_INPUT SMA as the RF signal using a low-loss SMA cable.

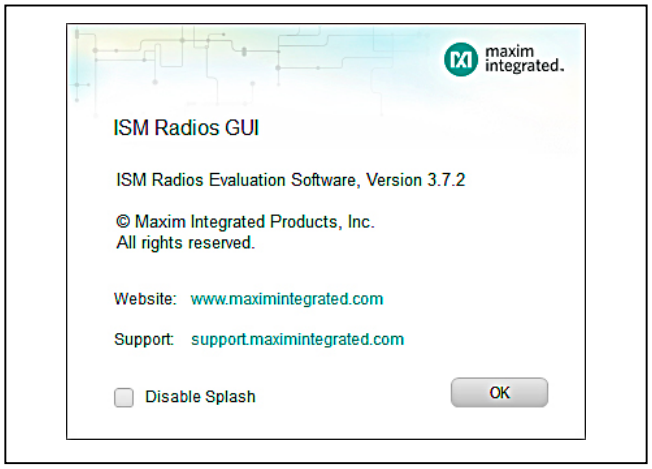


Figure 3. ISM Radios GUI Splash Screen

a) The ASK modulated signal should be centered at 433.92MHz at 2kbps Manchester (e.g., a 2kHz square wave).

5) Start the ISM Radios GUI.
   a) A GUI splash screen is displayed, as shown in Figure 3.
   b) To disable future displays of the splash screen, select **Disable Splash**.

6) The pop-up window (Figure 4) asks to select a device from the **Device** tab to get started; click **OK.**

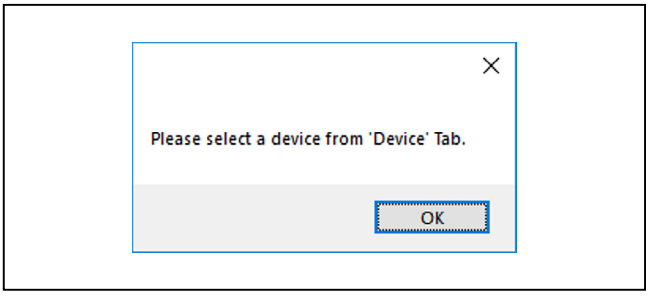


Figure 4. ISM Radios GUI Device Selection Reminder

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

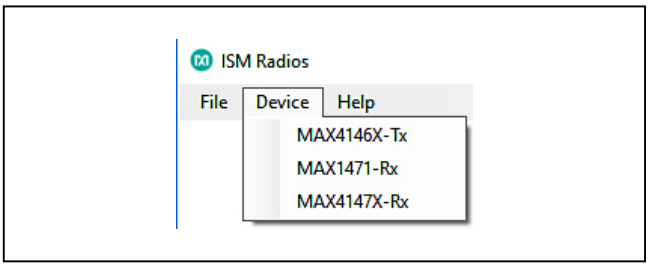


Figure 5. ISM Radios GUI Device Selection

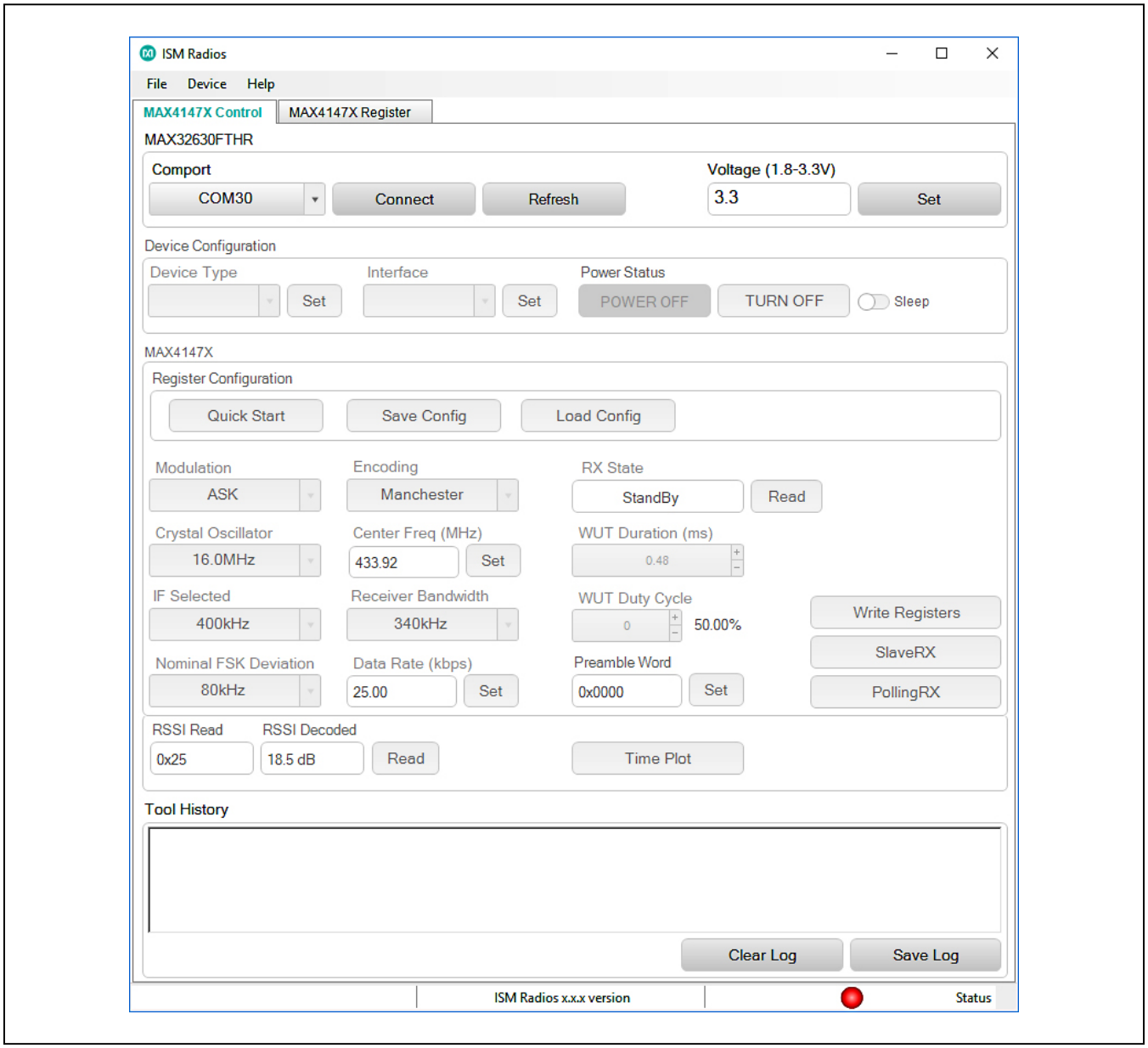


Figure 6. ISM Radios GUI Software Startup

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 (Figure 7) and click the **Connect** button. The **Connect** button changes 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 displays the associated port number.
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).
10) Enter a supply level into the **Voltage** field and click the **Set** button (Figure 9).
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.

Figure 7. COM Port

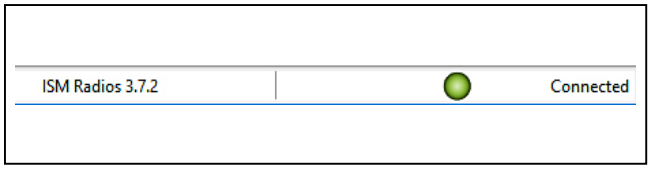


Figure 8. Connected, Indicators at Bottom of GUI

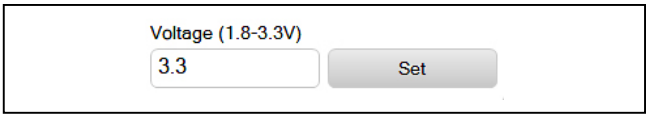


Figure 9. Supply Voltage Example

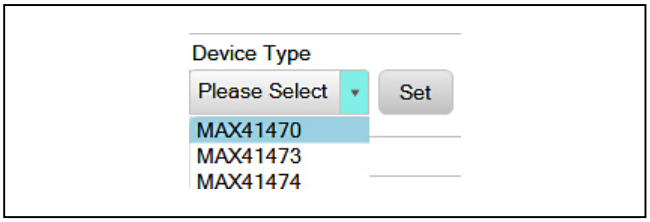


Figure 10. Part Selection

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

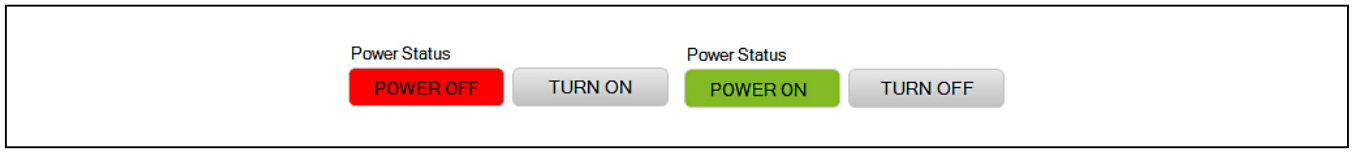


Figure 11. Power On Device

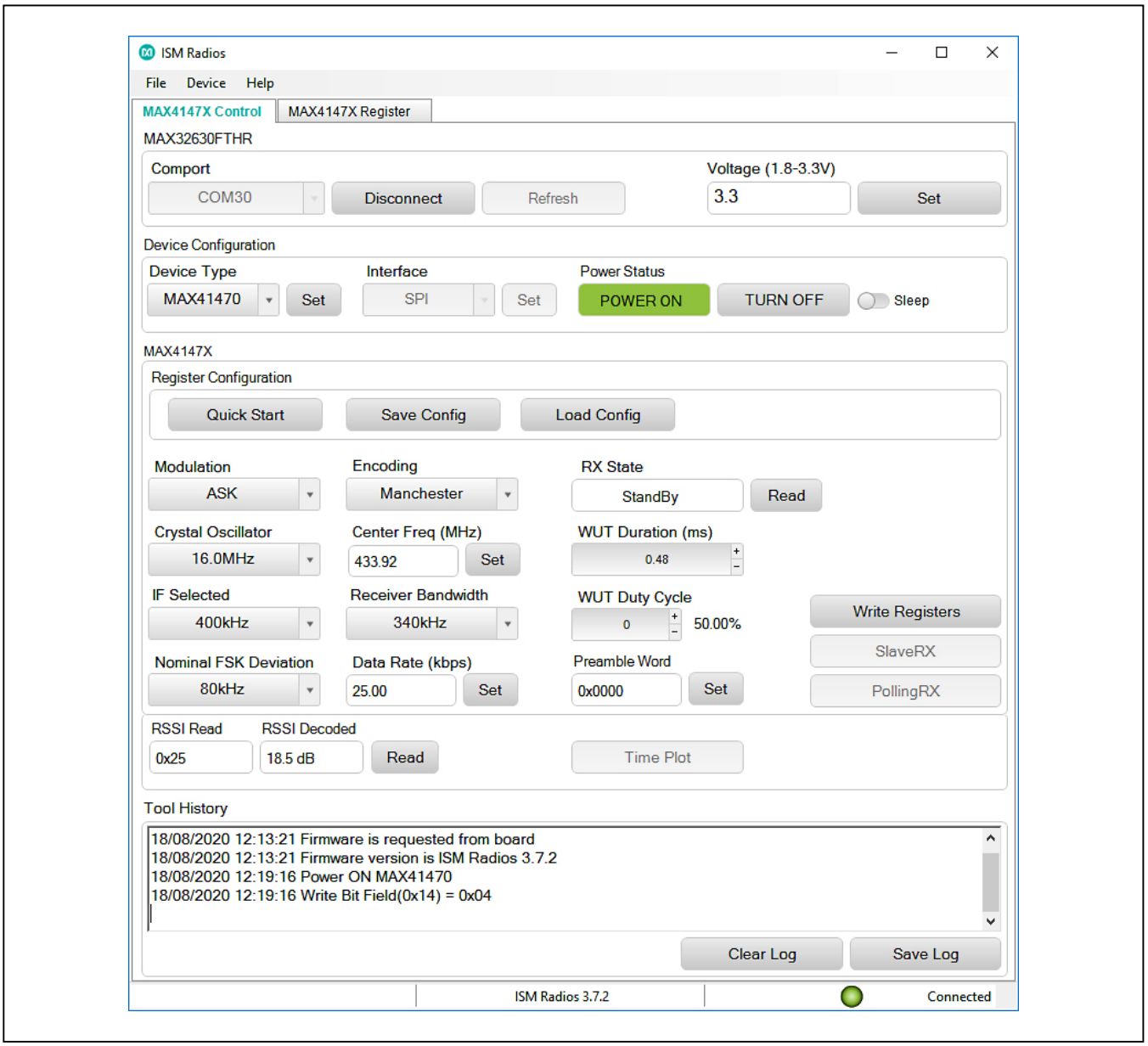


Figure 12. Connected and Powered-On GUI State

13) Configure the device, either through the Quick Start or the GUI configuration selections.
   a) Quick Start
      i. Click the **Quick Start** button (Figure 13) for an ASK configuration at 433.92MHz with IF at 200kHz and Manchester data rate at 2kbps.
   b) GUI Drop-Down and Entry Selections (Figure 14)
      i. Select **ASK** in the **Modulation** drop-down list.
      ii. Select **Manchester** in the **Encoding** drop-down list.
      iii. Enter ”433.92” in the **Center Frequency** field to represent 433.92MHz and click the **Set** button.
      iv. Select **200kHz** in the **IF Selected** drop-down list.
      v. Select **170kHz** in the **Receiver Bandwidth** drop-down list.
      vi. Enter “2” in the **Data Rate** field and click the **Set** button for a 2kbps setting.
      vii. Click the **Write Registers** button to write the configuration into the MAX41470 device.
      viii. Click the **SlaveRX** button to put the receiver into the Active SlaveRX state.
14) Connect the DATA (Yellow) test point to an oscilloscope to see the output data stream or through the **Time Plot** sampler (Figure 15). (See *Time Plot* in the *Detailed Description of Software* section.)

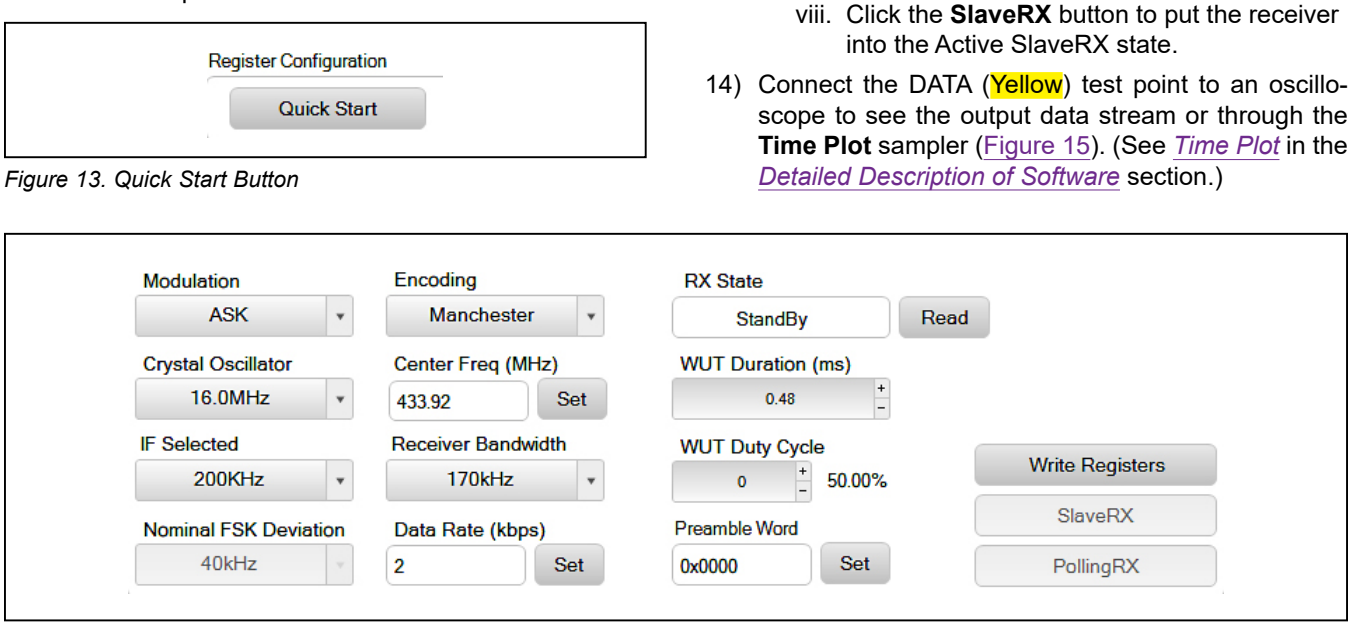


Figure 13. Quick Start Button

Figure 14. Device Settings and Write

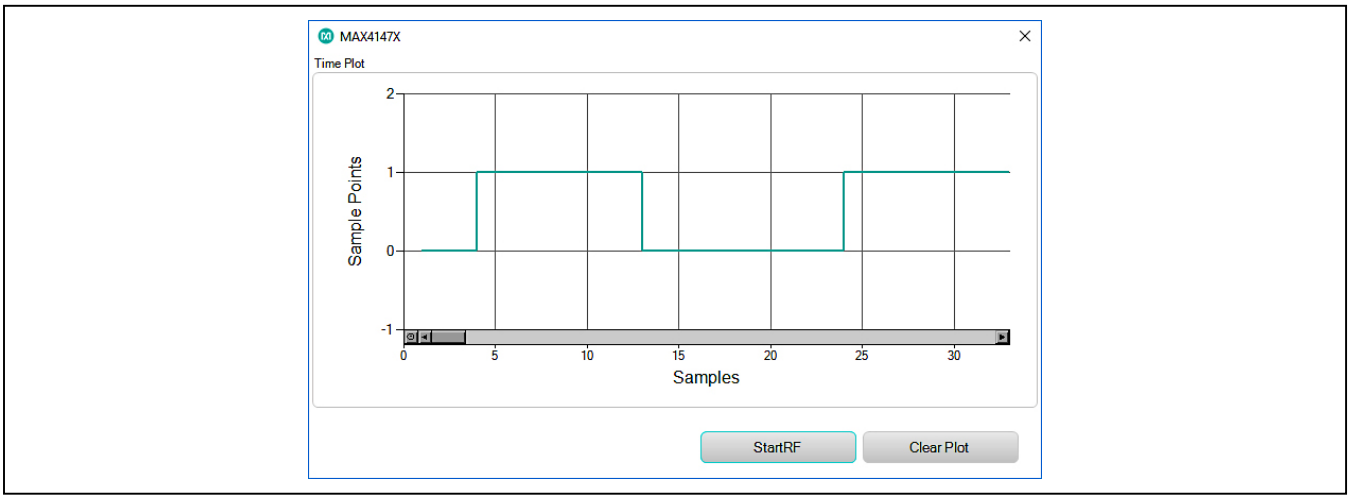


Figure 15. Time Plot of DATA

Preset Quick Start Procedure—Without FTHR Board

Set up and connect the MAX41470 EV kit hardware for standalone operation as follows:

1) Verify the jumpers on the MAX41470 EV kit board are as follows: JU1 is 2-3, JU3 is open, and JU5 is open.
2) Connect a 3.0V/20mA supply to the MAX41470 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.
4) Enable the power supply's output.
5) Connect the DATA (Yellow) test point to an oscilloscope to see the output data stream.

Table 3. MAX41470 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

MAX41470 EV Kit Printed Circuit Board

The MAX41470 EV kit PCB, as shown in Figure 16, is manufactured on a four-layer, 1oz copper, FR4 dielectric stack-up PCB. The board was designed to accommodate the MAX41470 and other members of the MAX41470 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

The MAX41470 device requires a three-wire SPI connection, and the MAX41470 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 MAX41470 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 MAX41470 EV kit comes preconfigured to directly connect the FTHR board through the H1/H2 headers to the SPI and the I^2C interfaces. The GUI determines which bus is used to communicate to the device based on the Device Type and Interface pull-downs selected in the GUI.

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 causes additional power consumption, and they are not included by default in the EV kit assembly.

PMOD Interface

The MAX41470 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 SPI PMOD interfacing. Populating this header would allow control from the MAX32600MBED kit and the MAXREFDES72# Arduino® Uno R3 to the PMOD shield adaptor. When using the PMOD interface to supply the MAX41470 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.

Arduino is a registered trademark of Arduino AG.

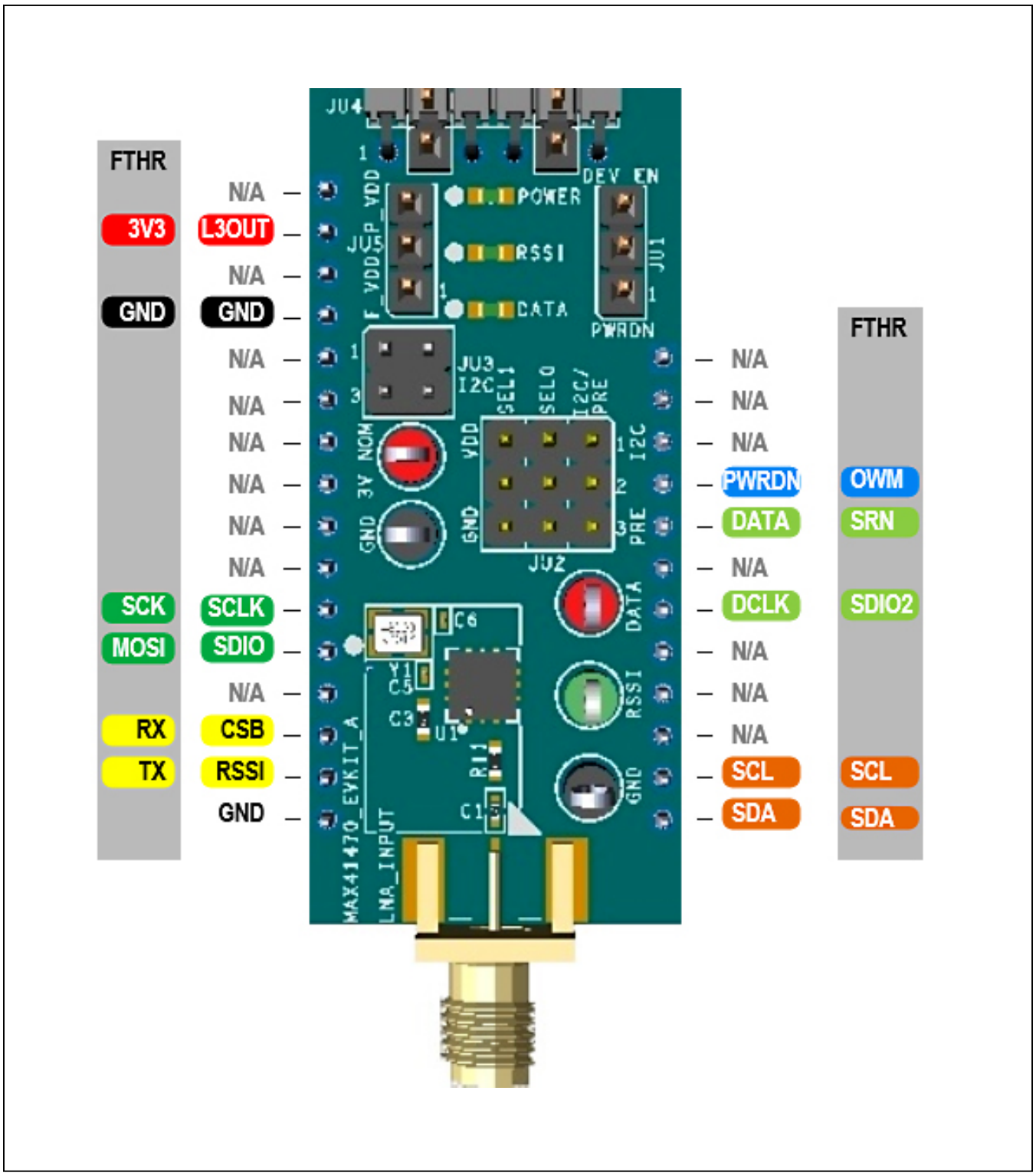


Figure 16. MAX41470 EV Kit Interface

Detailed Description of Software

The MAX41470 EV kit controller GUI software is designed to control the MAX41470 EV kit board and the MAX32630FTHR board, as shown in Figure 17. The software includes USB controls, which provide SPI and power to the MAX41470 through the FTHR board interface.

Comport

The **Comport** section provides a drop-down list of serial communication ports available for connection to a MAX41470 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 displays 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 MAX41470 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 the **Set** button. The default value of the L3OUT voltage is 3.3V.

When using the FTHR board interface to supply the MAX41470 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 choose which receiver is attached to the FTHR board. This selection configures the GUI software to interface through the SPI pins when the MAX41470 is 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 MAX41470 device.

To select the **modulation**, choose 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 MAX41470 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 is 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 MAX41470 (f_C or f_{LO}). The value entered in this section is 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 calculates the values for the PLL registers using the crystal frequency and the following formula:

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

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 are 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 is 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

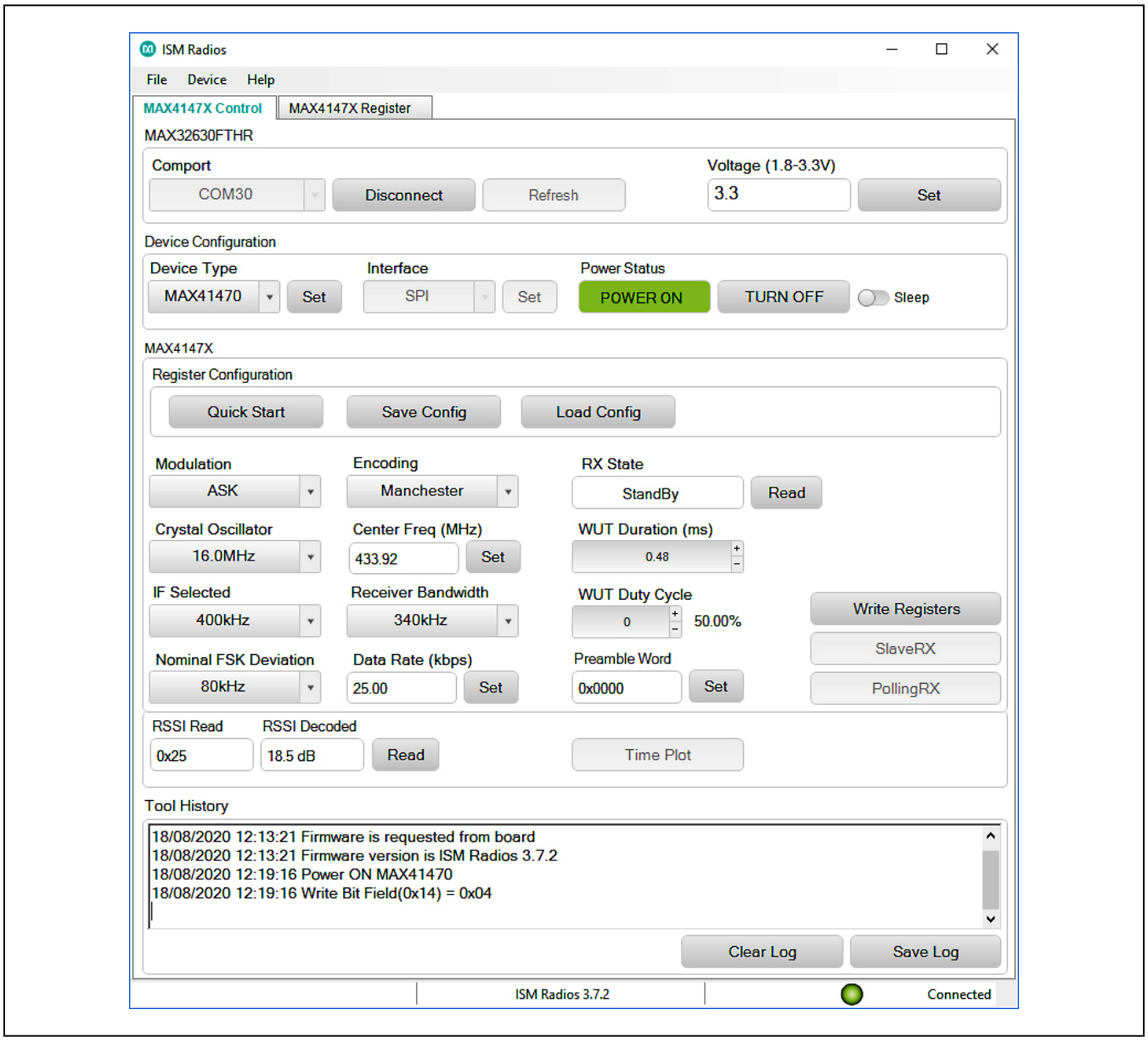


Figure 17. ISM Radios GUI Configuration

frequency and the channel filter bandwidth. This impacts 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 MAX41470 detects the preamble word. When the word is detected, the interrupt from the MAX41470 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 transitions 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 MAX41470 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 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 does 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, transitions 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 programs the PREAMB_LEN based on this entry.

Action Buttons – Write Registers

The **Write Registers** button take the combination of all the settings within the GUI and write the registers accordingly. Many of the settings within the MAX41470 are dependent on multiple bit settings within the device. This button combines all settings to determine the register writes required.

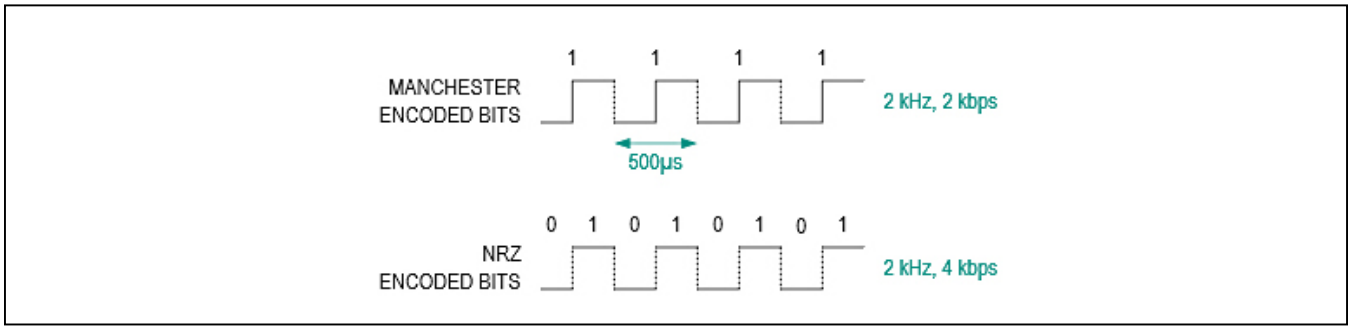


Figure 18. Manchester and NRZ Data Rates

Action Buttons – SlaveRX

The **SlaveRX** button (Figure 19) is active once the registers are programmed, either through the **Write Registers** button or the **Load Config** button. This button enables the SlaveRX state within the receiver for active data reception. In this state, the DATA test point on the EV kit displays the data received. Since the configuration registers should not be programmed in any active state, the GUI grays out the other buttons and settings during the SlaveRX state. The RX State status moves to the Slave Receiver state. To exit the **SlaveRX** state, click the SlaveRX button again.

Action Buttons – PollingRX

The **PollingRX** button is active once the registers are programmed, either through the **Write Registers** button or the **Load Config** button. This button places the device into PollingRX mode for preamble detection. When the preamble is detected, the microcontroller automatically responds appropriately, as previously described, and places 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** button (Figure 20) should be clicked to begin the display. After it is running, the **StopRF** can be clicked to stop the sampling. Clicking the **Clear Plot** button (Figure 21) clears the current display. To zoom in or out of the display samples, use the mouse scroll.

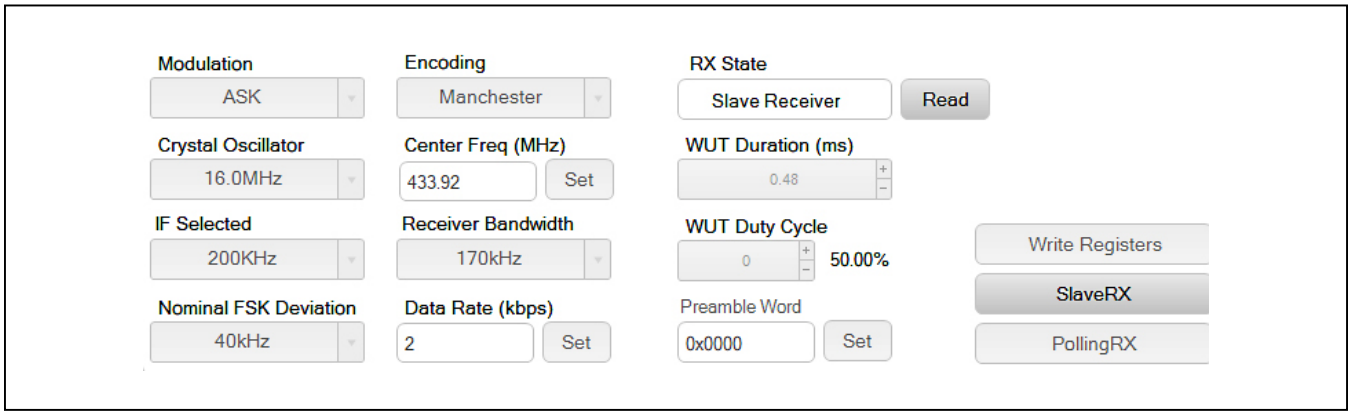


Figure 19. Active SlaveRX State

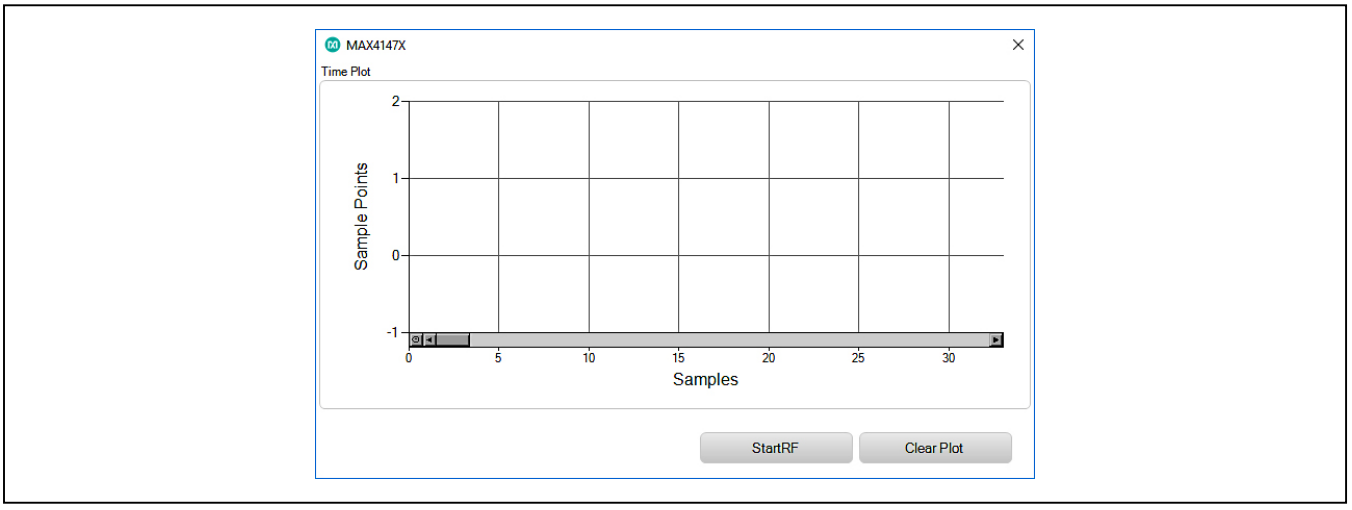


Figure 20. Initial Time Plot Window

Tool History

This portion of the GUI (Figure 22) contains a log file text block that 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** deletes the contents in the text block.

Clicking the **Save Log** button opens a **Save As** explorer window and the user is prompted to save a .txt file.

Direct Register Access

The GUI software allows for direct access to all the available registers through the MAX41470 **Register** tab (Figure 23) when interfacing with the MAX41470 SPI-based device.

Register List

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

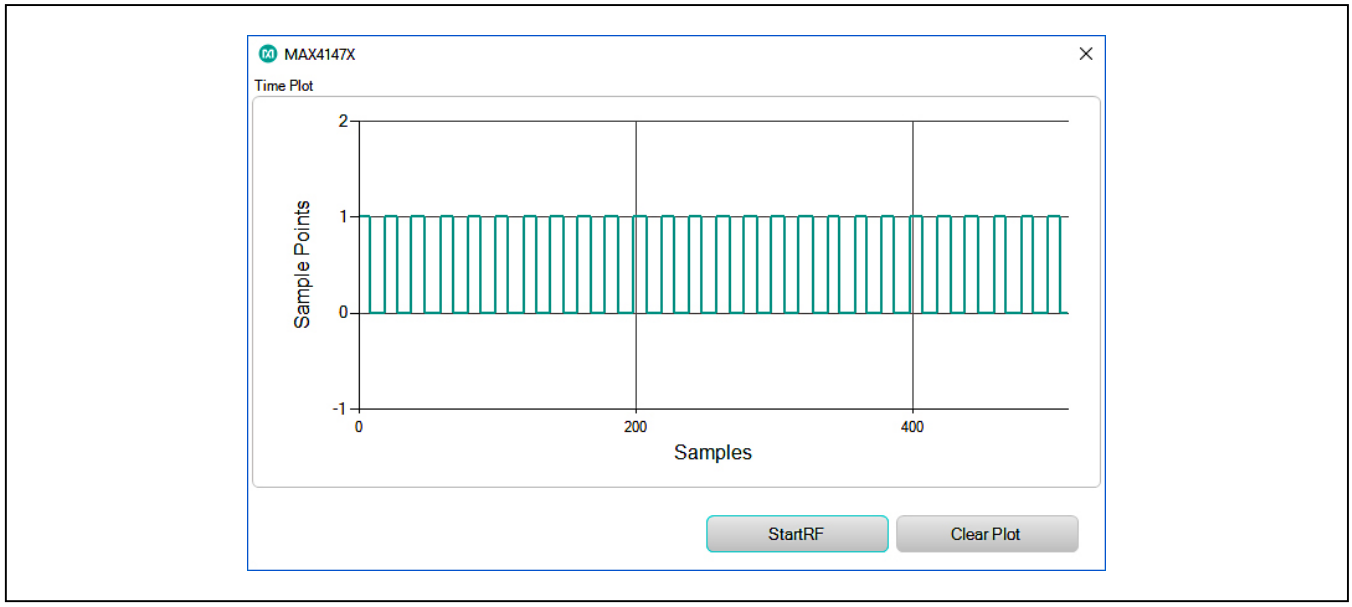


Figure 21. Time Plot Display

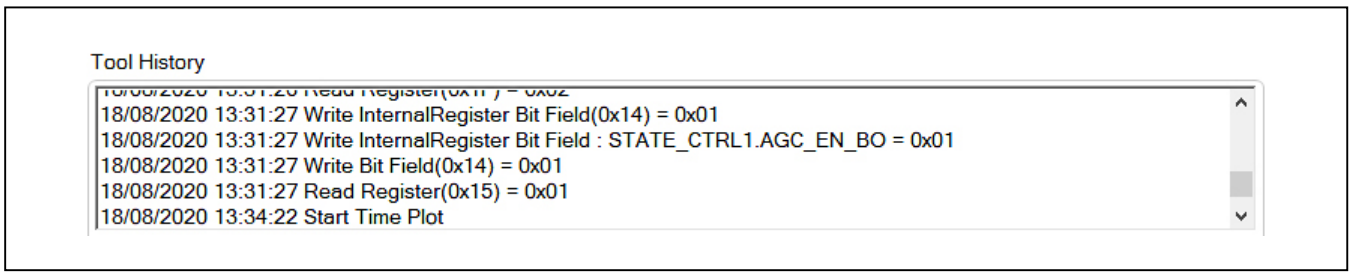


Figure 22. Tool History

Register Value

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

The body of this tab 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.

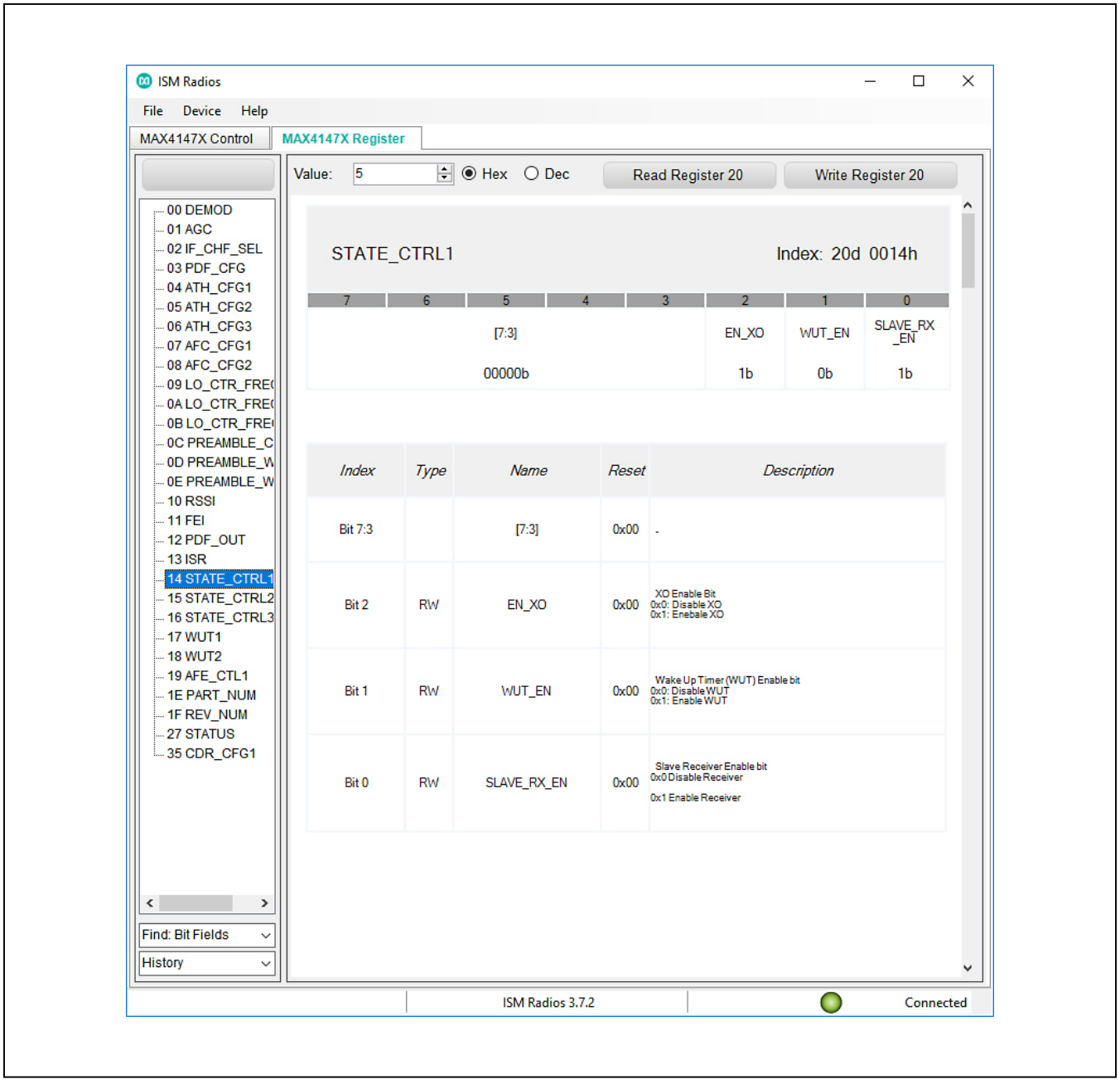


Figure 23. Register Tab

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 press **Enter**—the new value is automatically reflected in the **Value** field and is 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 closes 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 displays 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 ISM Radios 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 MAX41470 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 two options. It relies on the part number to choose the modulation and jumper settings (or tri-level pin connections) to configure the part for a defined carrier frequency. 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.

SPI Mode

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 the receive frequency, data rate, thresholds, bandwidths, etc.

The MAX41470 device allows for a three-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 States

When communicating with a MAX41470 device, 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.

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 is reported as 0x70.

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

MAX41470 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

PART	QTY	DESCRIPTION
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+
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		www.panasonic.com
Sullins	760-744-0125	www.sullinscorp.com
Vishay Dale	800-433-5700	www.vishay.com

Note: *Indicate that you are using the MAX41470 when contacting these component suppliers.*

Ordering Information

PART	TYPE
MAX41470EVKIT#	EV Kit

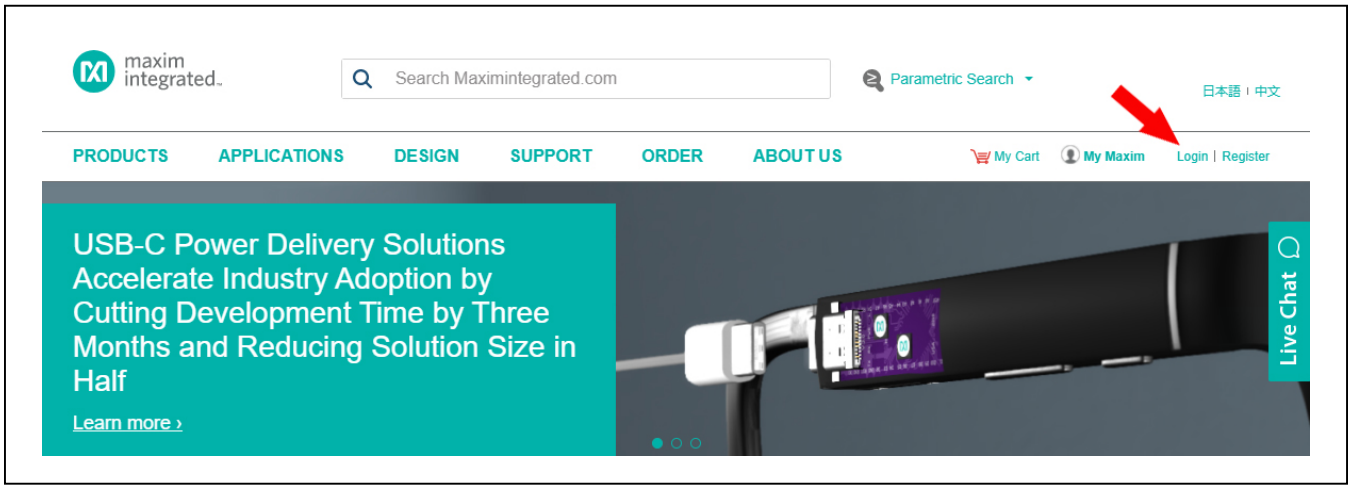
#Denotes a RoHS-compliant device.

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

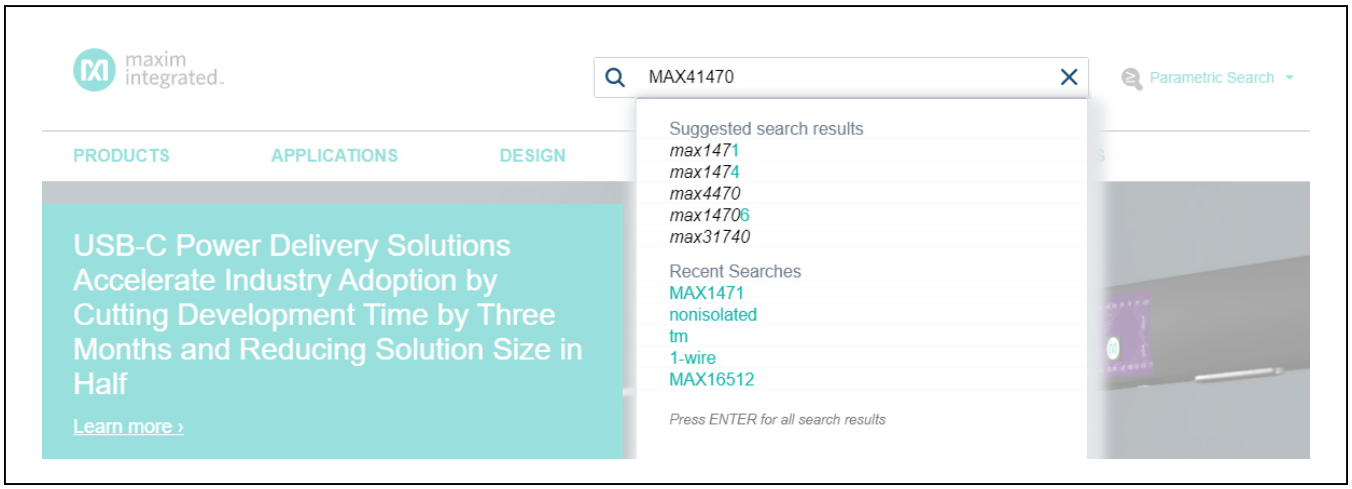
Download the ISM Radios GUI

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

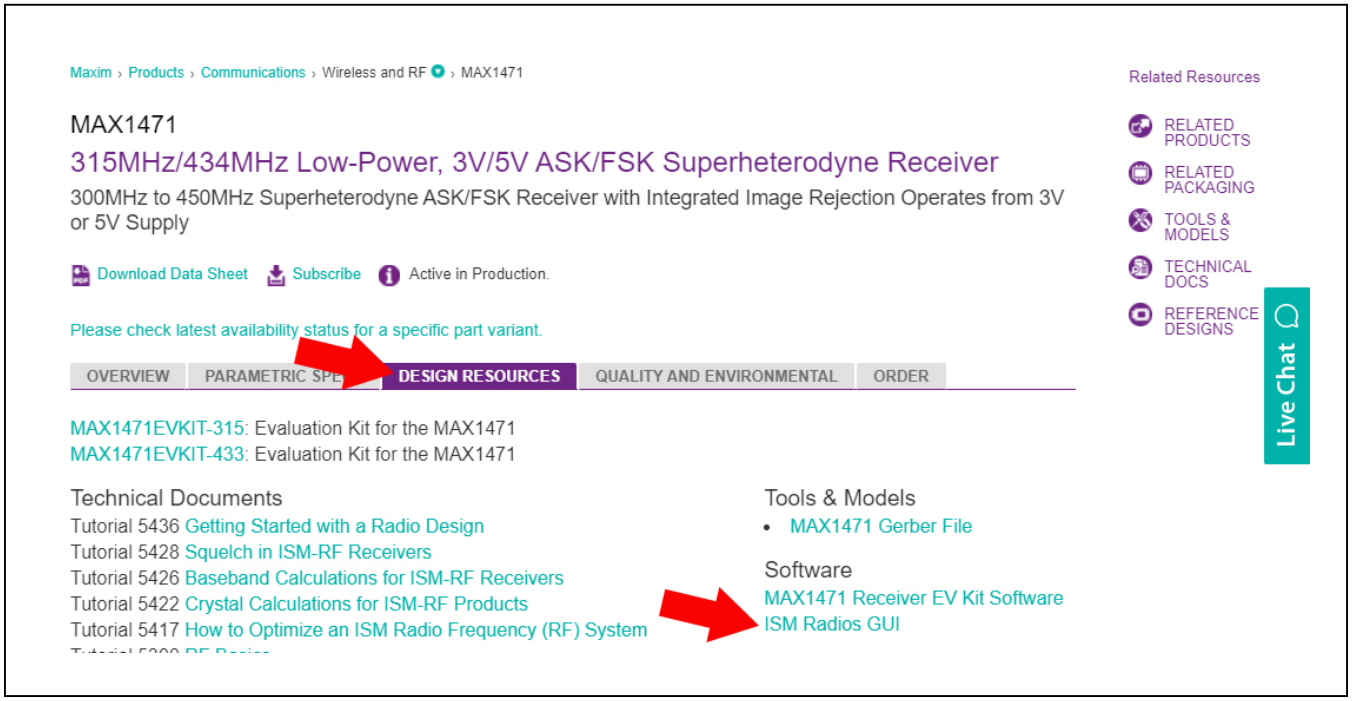
1) Log in to your MyMaxim account on the website.



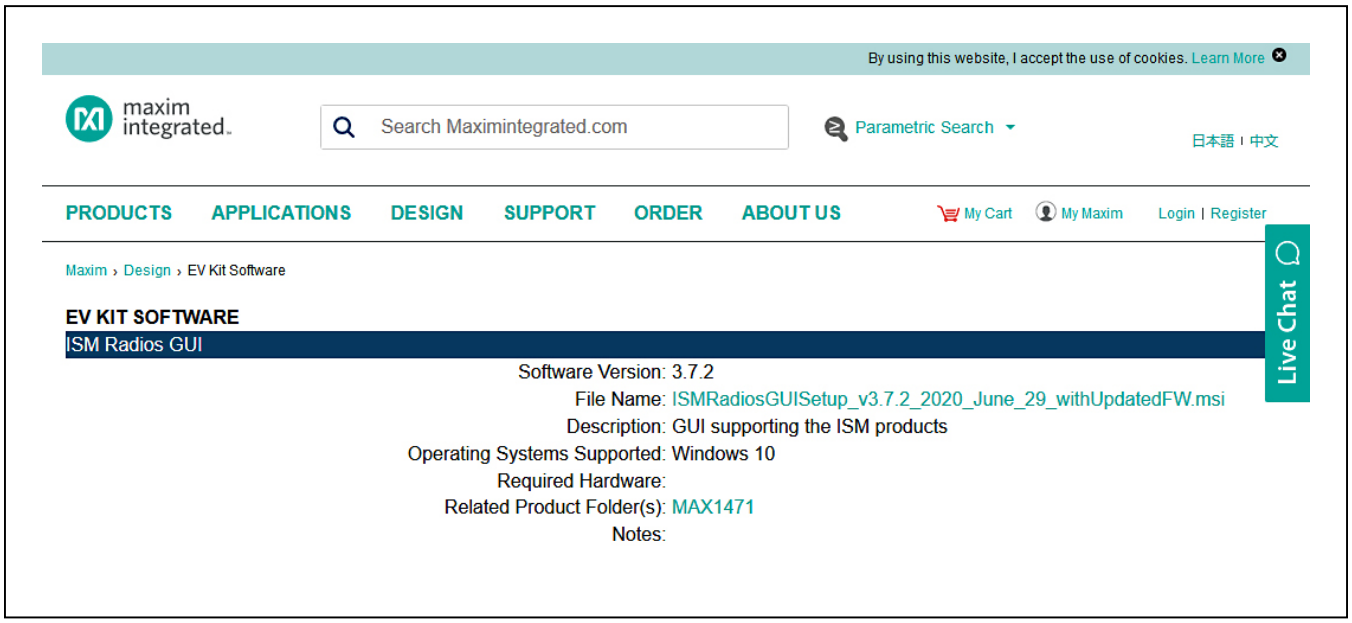
2) Click on the magnifying glass and search for the **MAX41470**.



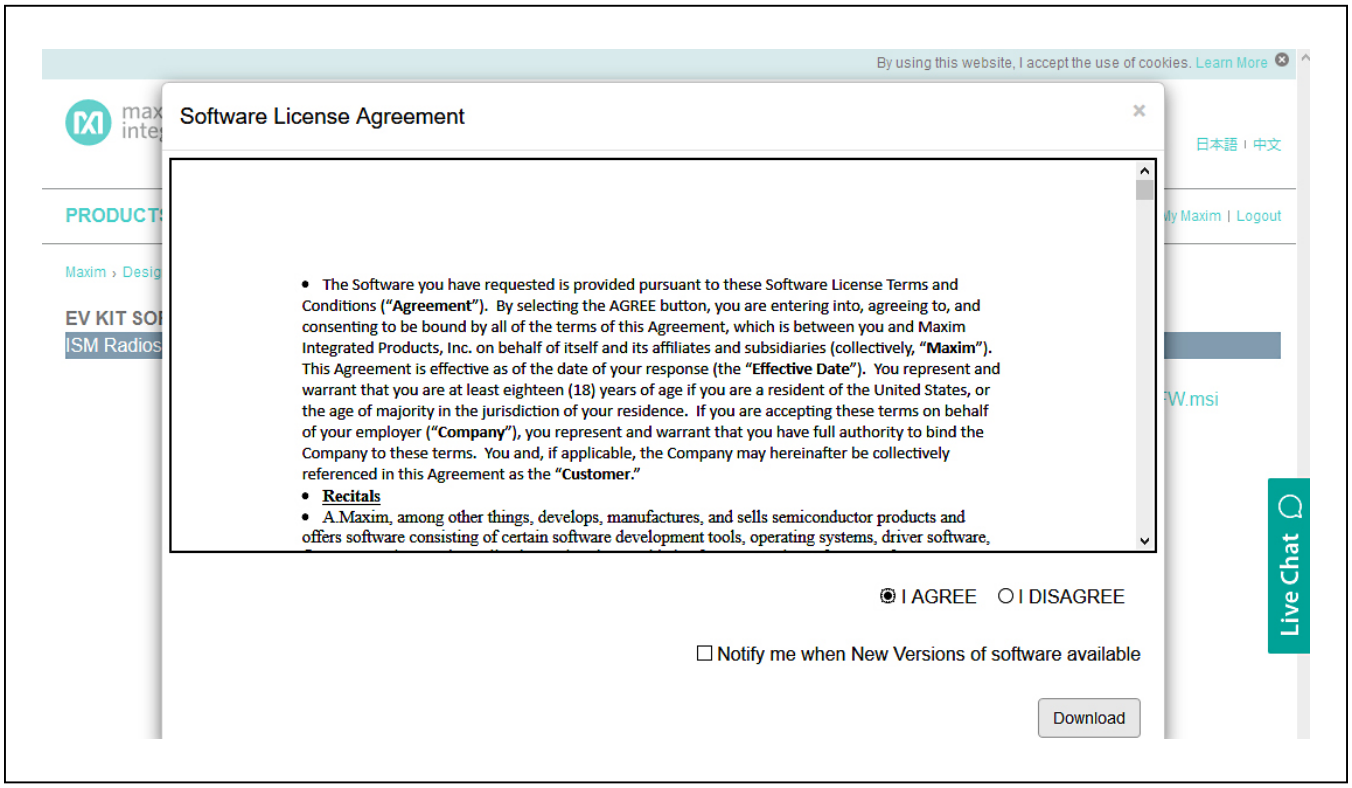
3) Click on the **Design Resources** tab on the appropriate product web page and click on the software link.



4) Click the file link on the software landing page to download the MAX41470 EV kit software package.



5) Review the Maxim Software License Agreement (SLA), and accept the terms by selecting **I AGREE** and clicking the **Download** button.



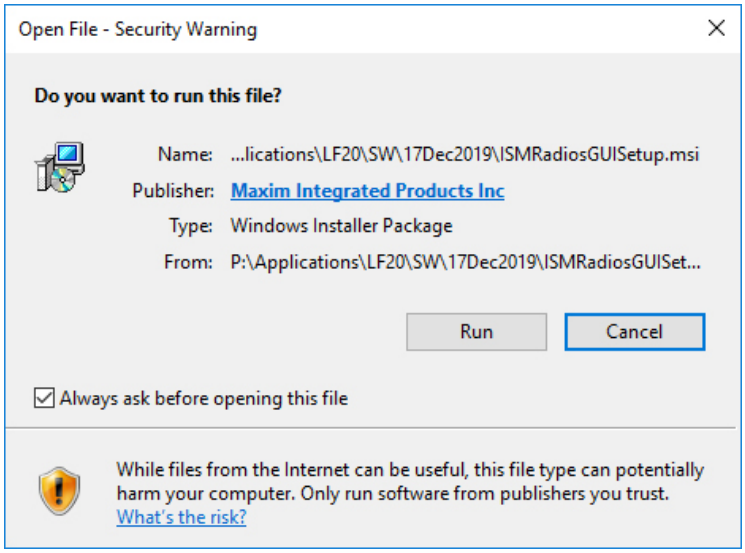
6) Save the EV kit software package to your desktop or other accessible location for later install.

Install the ISM Radios GUI

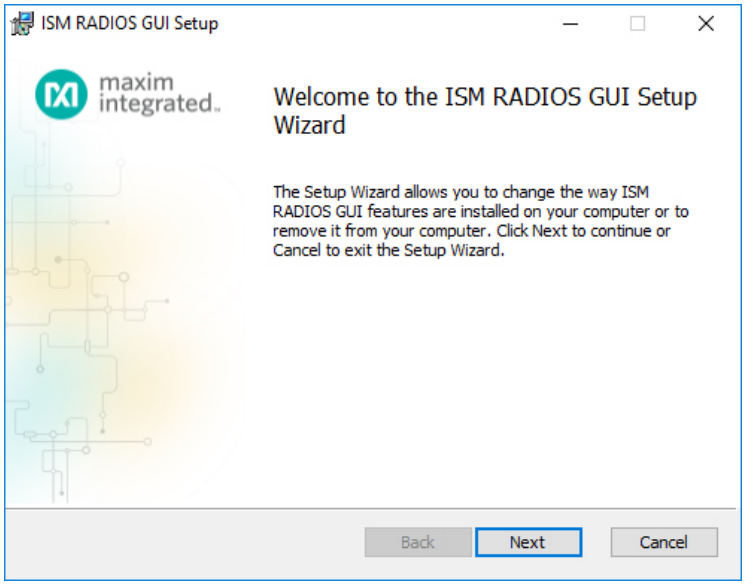
This software and firmware are available from the **Maxim website**. See the *Download the ISM Radios GUI* section for information on obtaining the latest firmware from Maxim.

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

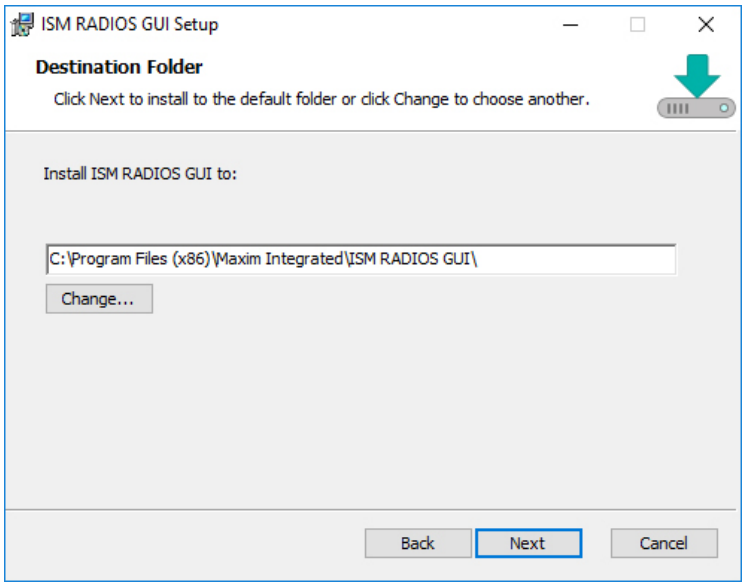
1) Double-click the ISMRadiosGUISetup.msi setup file and follow the Setup Wizard prompts.
   a) If a security warning appears, click **Run**.



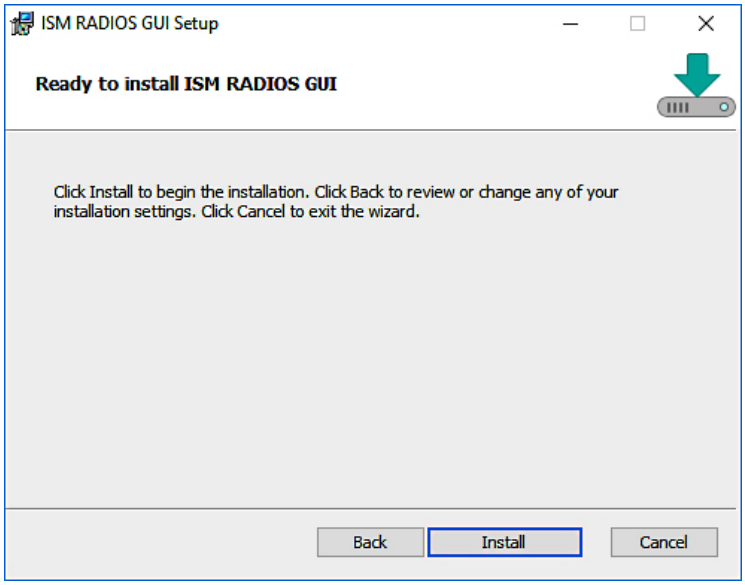
   b) In the **Welcome** window, click **Next**.



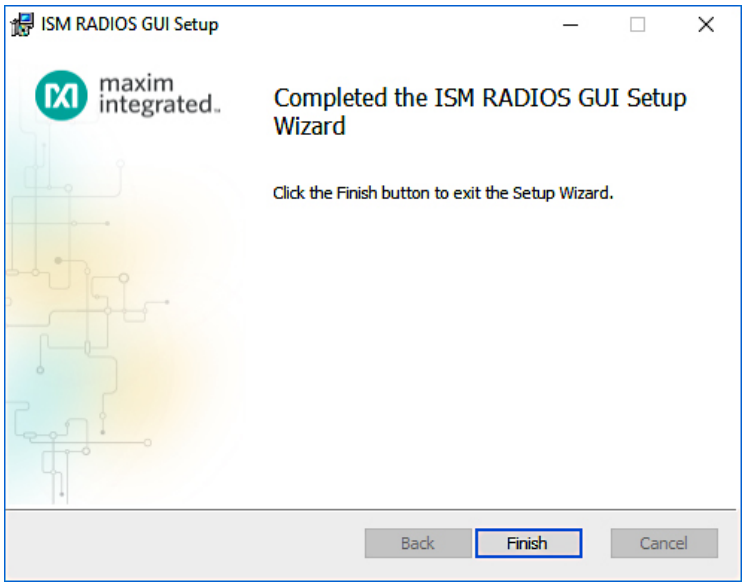
c) Use the default **Destination Folder**, and click **Next**.



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



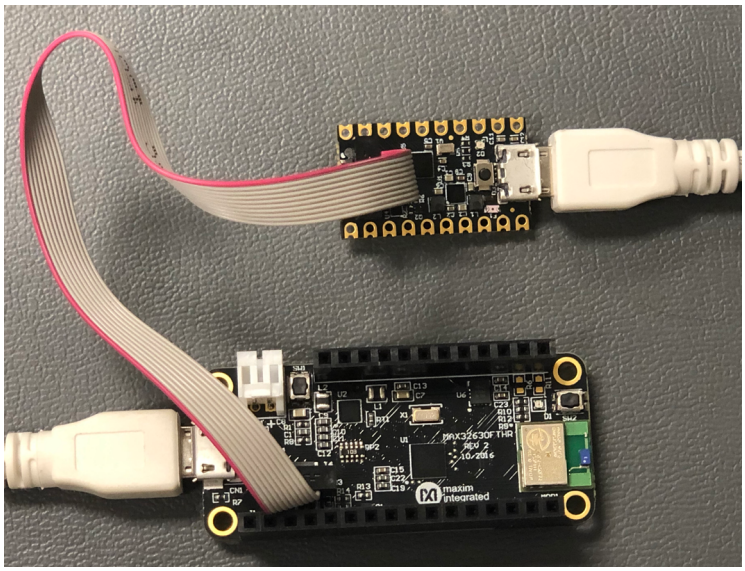
e) Click **Finish** when the setup process is complete.



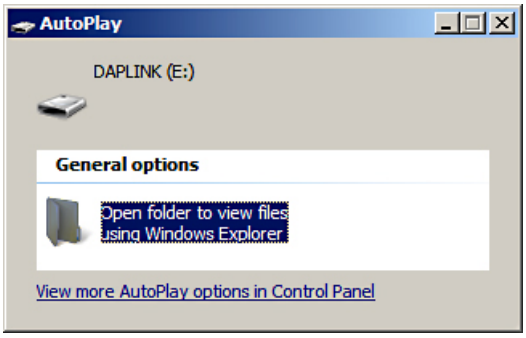
Program the MAX32630FTHR Board with the MAX41470 Firmware

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.

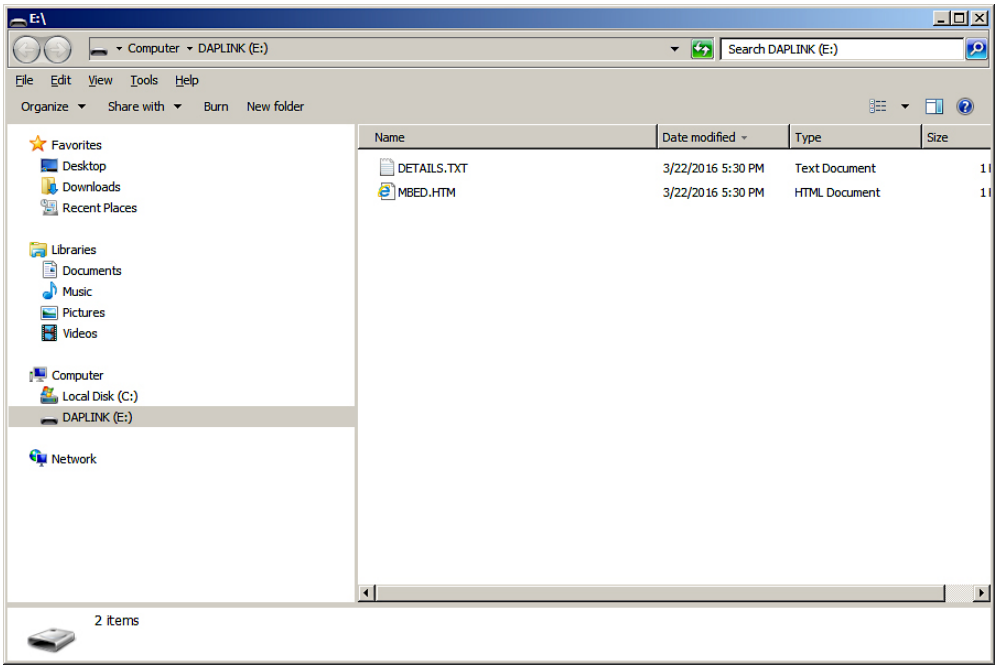
1) Connect the MAX32630FTHR to the MAX32625PICO.
   a) Use the fine-pitch, 10-pin ribbon cable to connect the boards from the SWD (J3) header on the HDK to J4 on the MAX32630FTHR.



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 (the white USB cable off the right side of the photo).



   b) The status LED on the DAPLINK board blinks red when connecting.
   c) After a few seconds of activity, the PC recognizes 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 shuts off and the LED on the MAX32625PICO slowly flashes red as the FTHR board is being programmed.

b) Once the programming is complete, the DAPLINK USB drive disconnects from the PC and reconnects 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 prepare 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 restarts and the newly programmed application begins 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 **Sign up** and create your mbed account. Otherwise, log in with your normal username and password. This gives you access to the website, tools, libraries, and documentation.

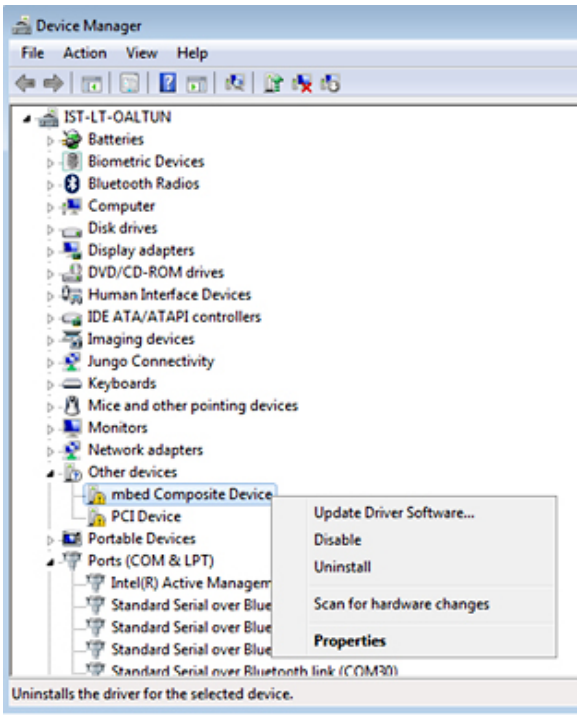
You must load the matching HDK image (**MAX32630FTHR DAPLINL image**) for the platform you are programming in order for drag-and-drop programming to work.

Update the MAX32630FTHR Board Driver

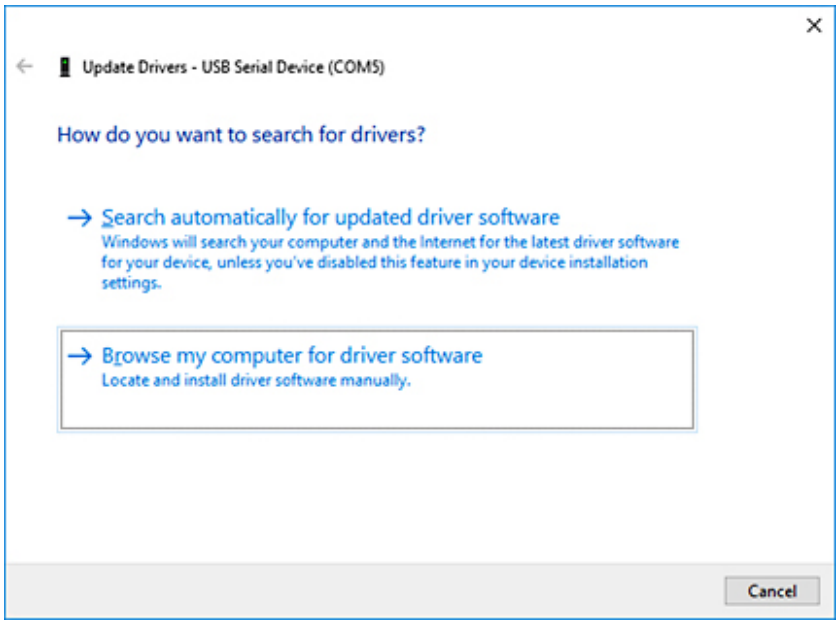
The required driver is available from the **Maxim website**. See the *Download the ISM Radios GUI* section in this documentationfor information on obtaining the latest driver from Maxim.

1) Connect the MAX32630FTHR to the PC’s USB port.

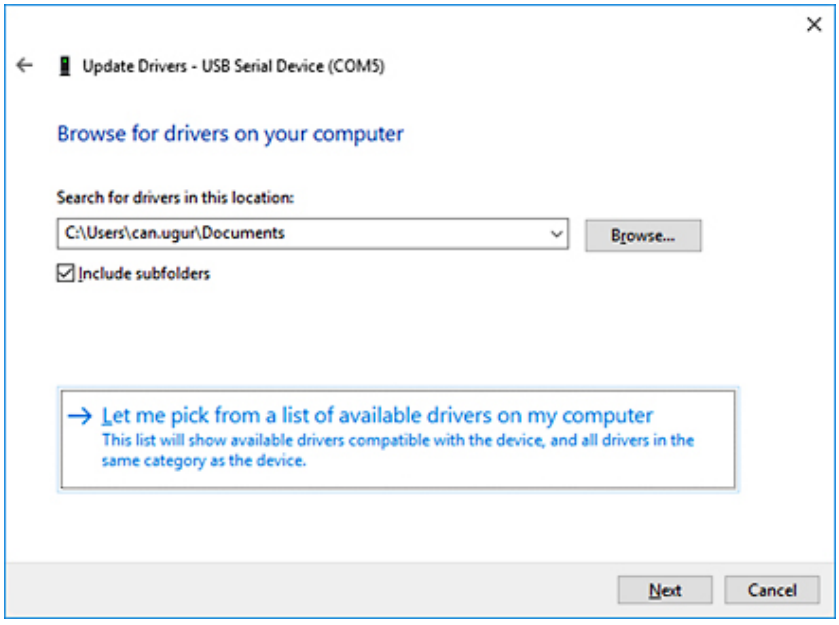
2) In **Device Manager**, right-click **Other devices** > **CDC Device** or **mbed Composite Device**.



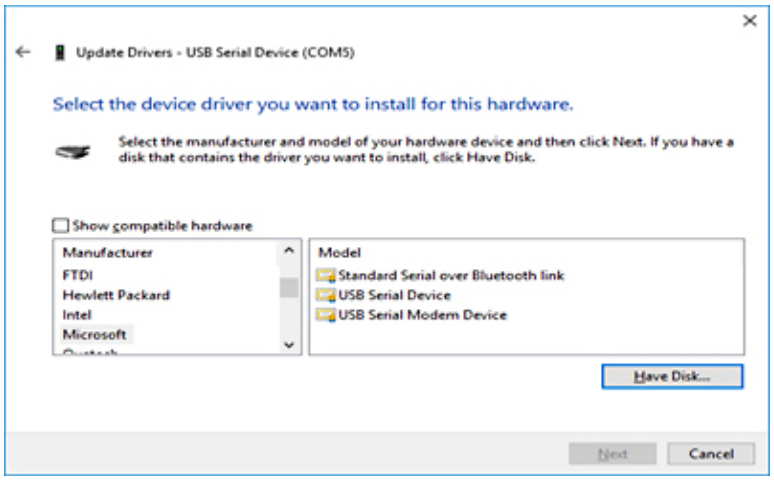
3) Click **Update Driver Software**, then select **Browse my computer for driver software.**



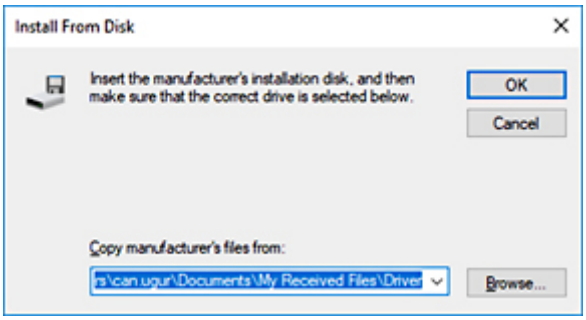
4) Select **Let me pick from a list of available drivers on my computer**, and click **Next**.



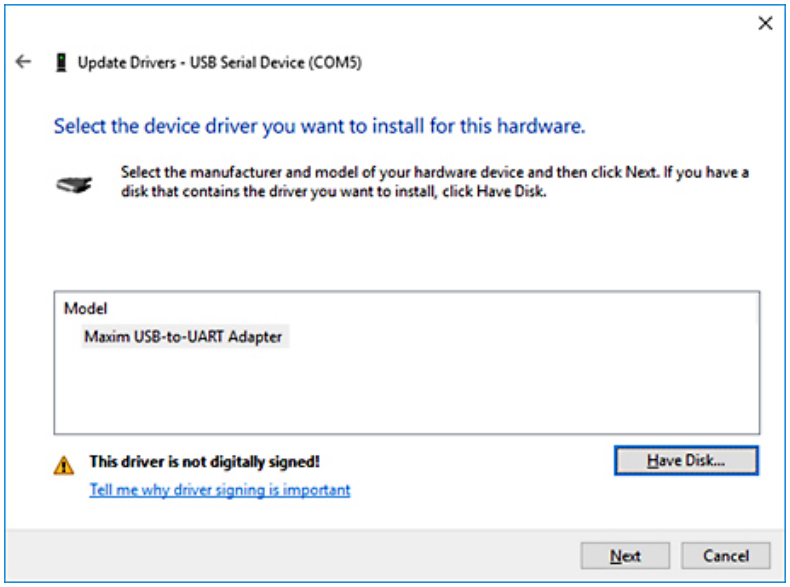
5) On a Windows 10 operating system, click the **Have Disk…** button.



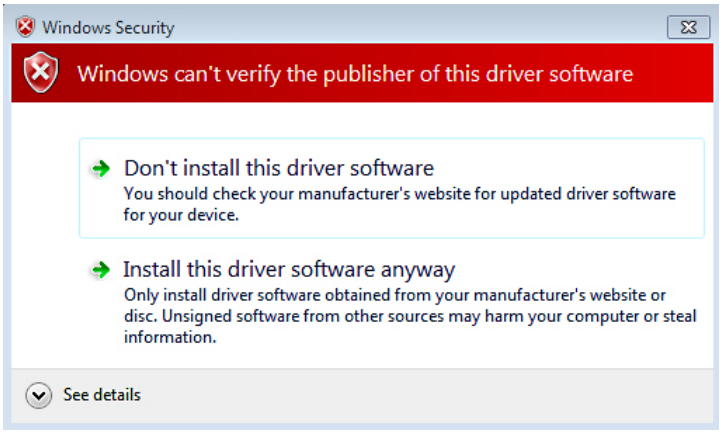
6) Browse to the path of the driver folder and click **OK**.



7) Select the device driver and click **Next**.



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

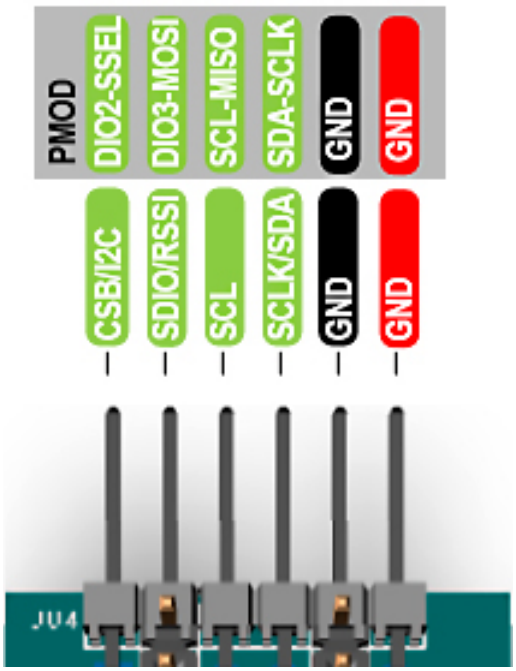


Appendix II – Hardware Modifications

PMOD Header Interface

The MAX41470 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 MAX41470 EV kit with power, be sure to connect the JU1 jumper between pins 2-3.

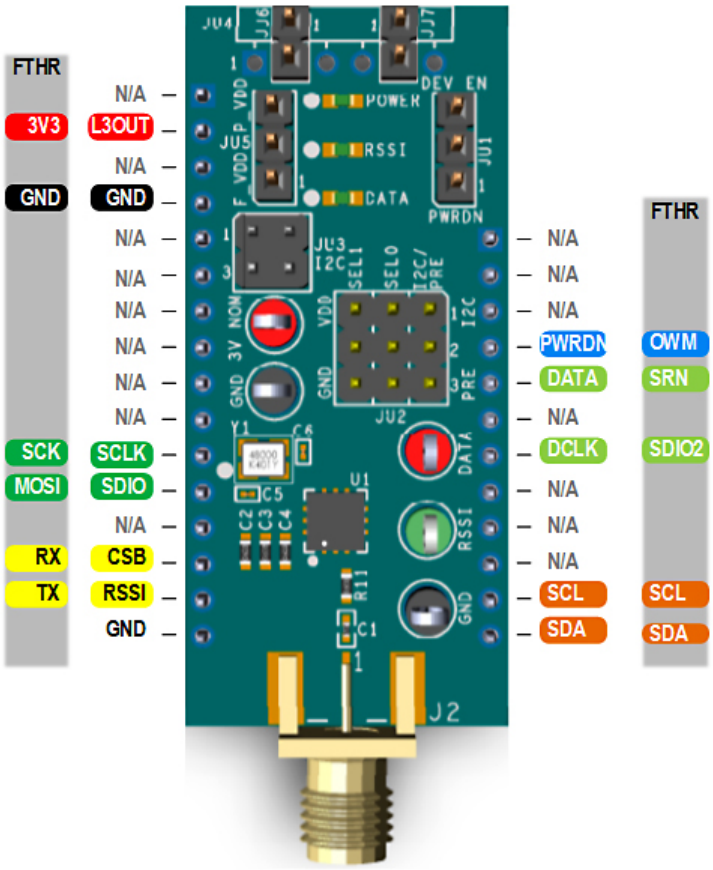


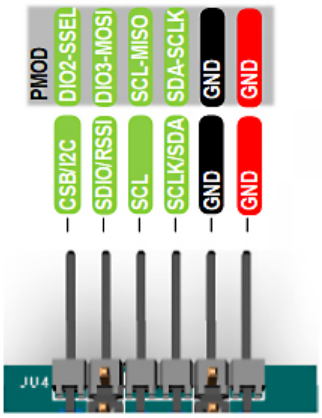
Appendix III – Pinout Sheets

MAX41470 EV Kit

FTHR Board Connectors

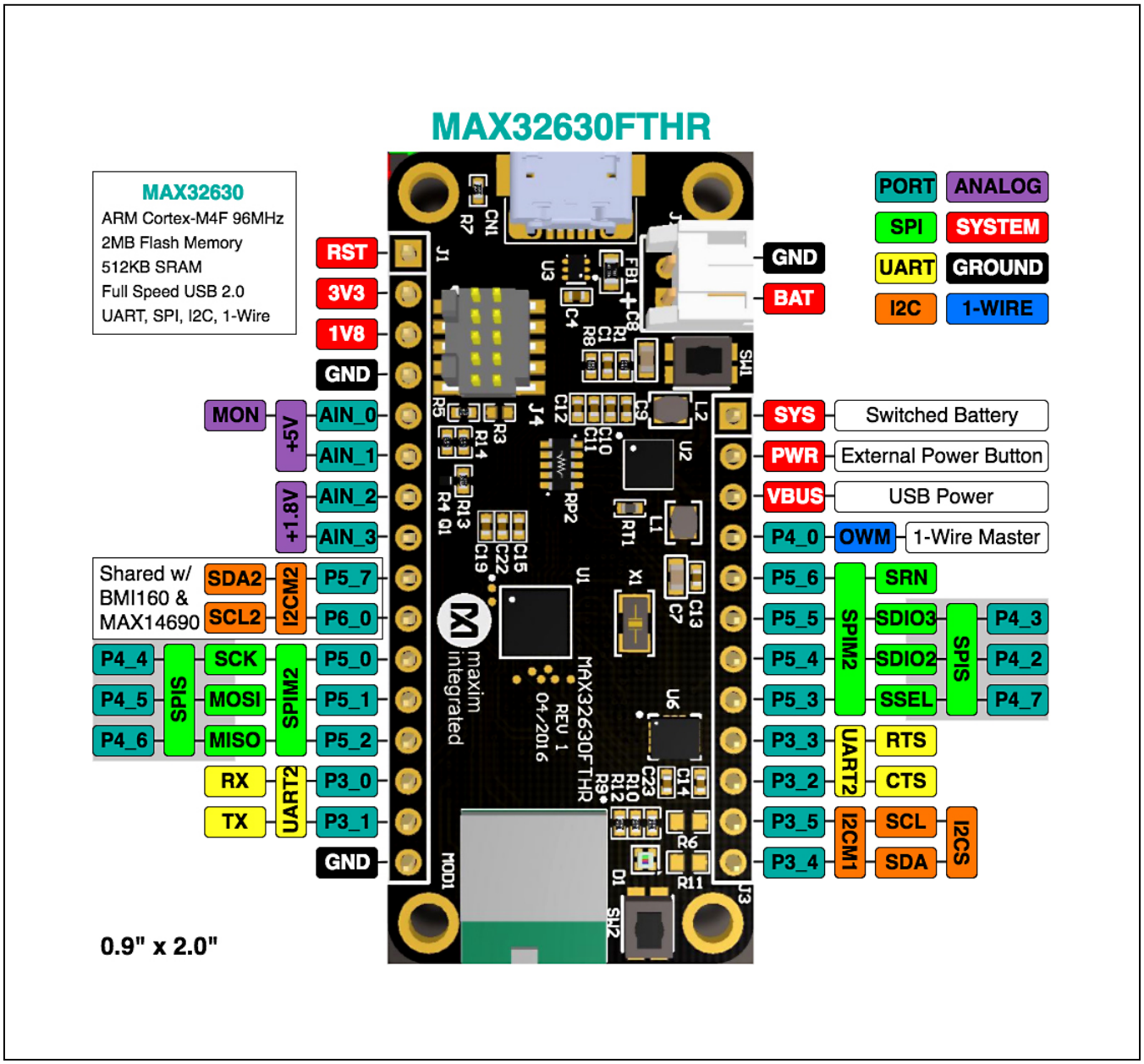
Peripheral Module Connector



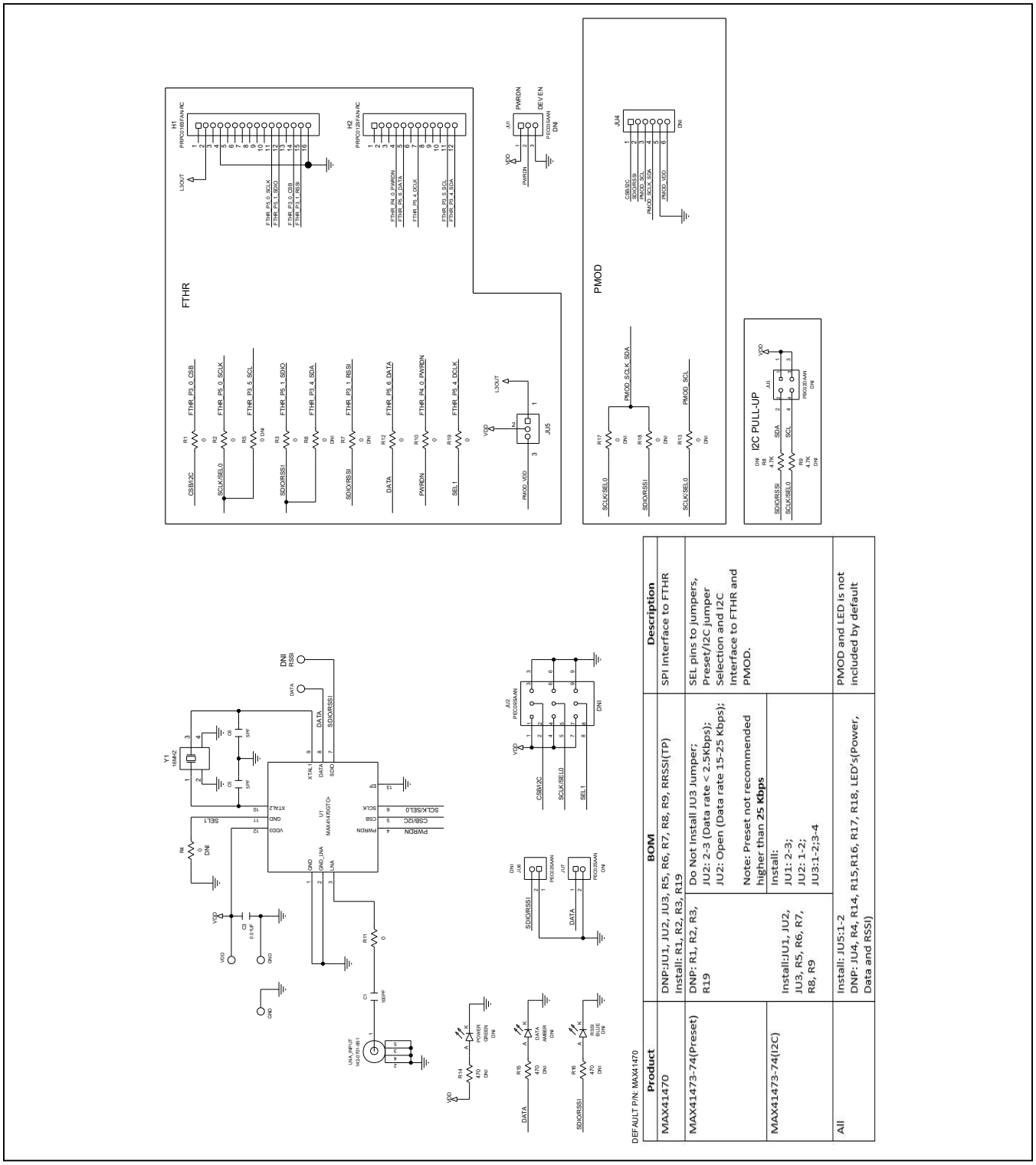


MAX32630FTHR

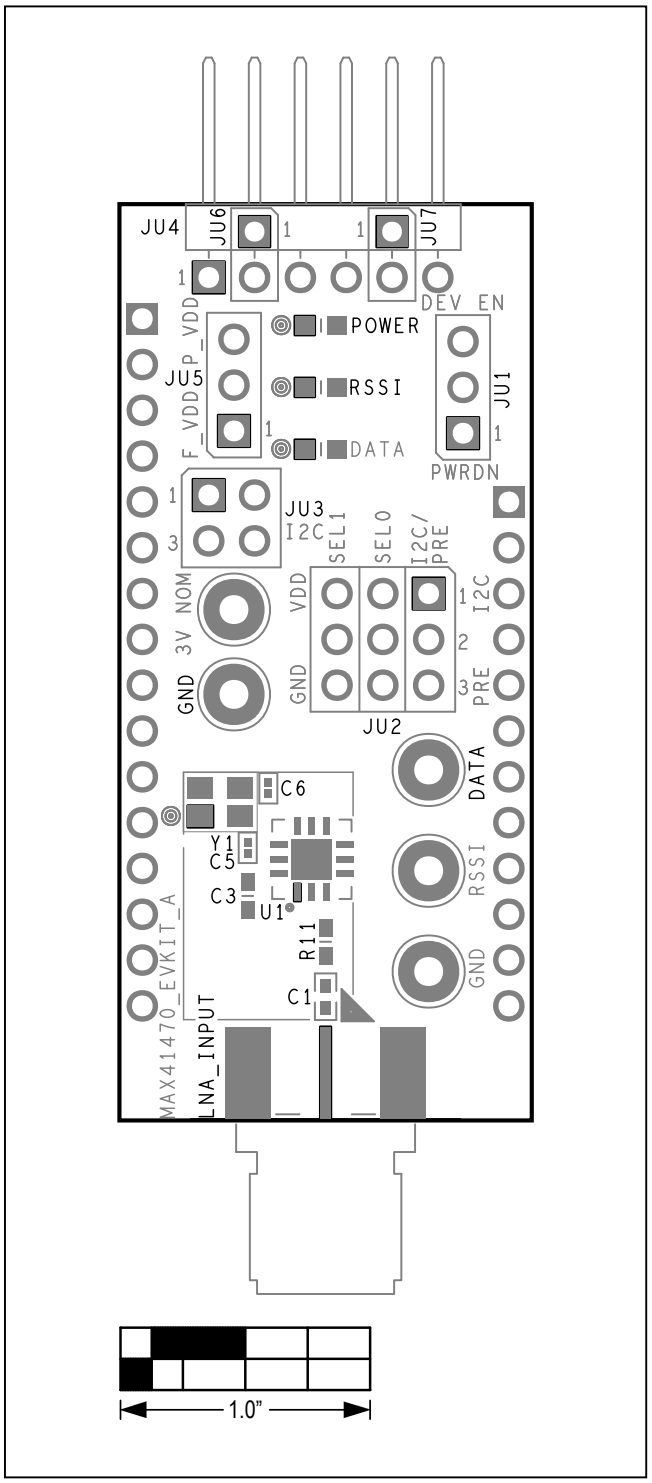
Arm Cortex-M4F microcontroller rapid development platform.



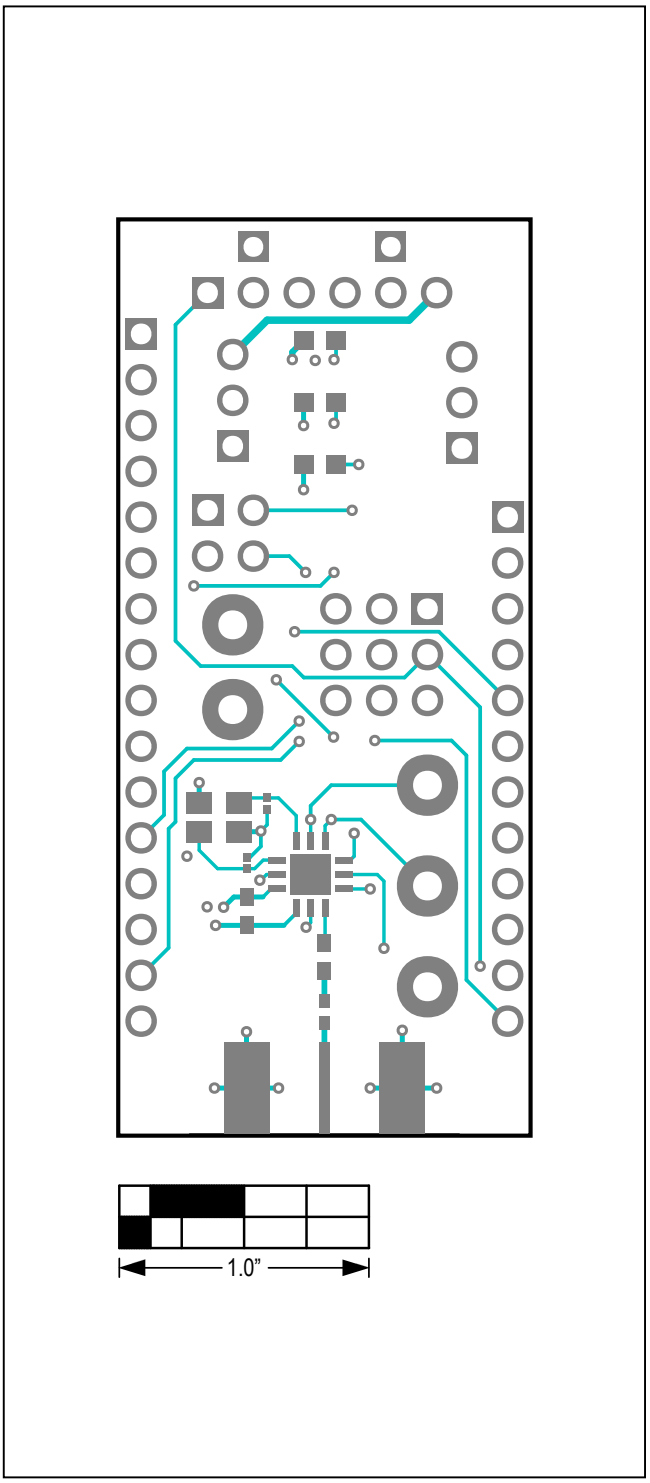
MAX41470 EV Kit Schematic Diagram



MAX41470 EV Kit PCB Layout Diagrams

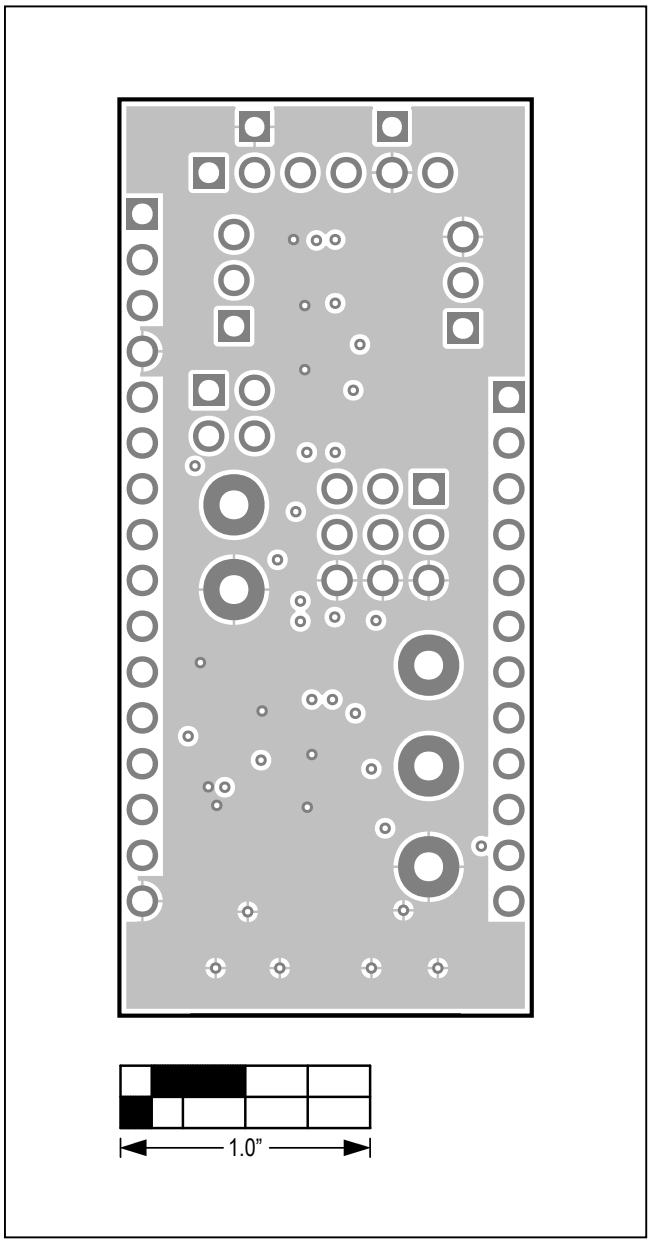


MAX41470 EV Kit PCB Layout – Silkscreen Top

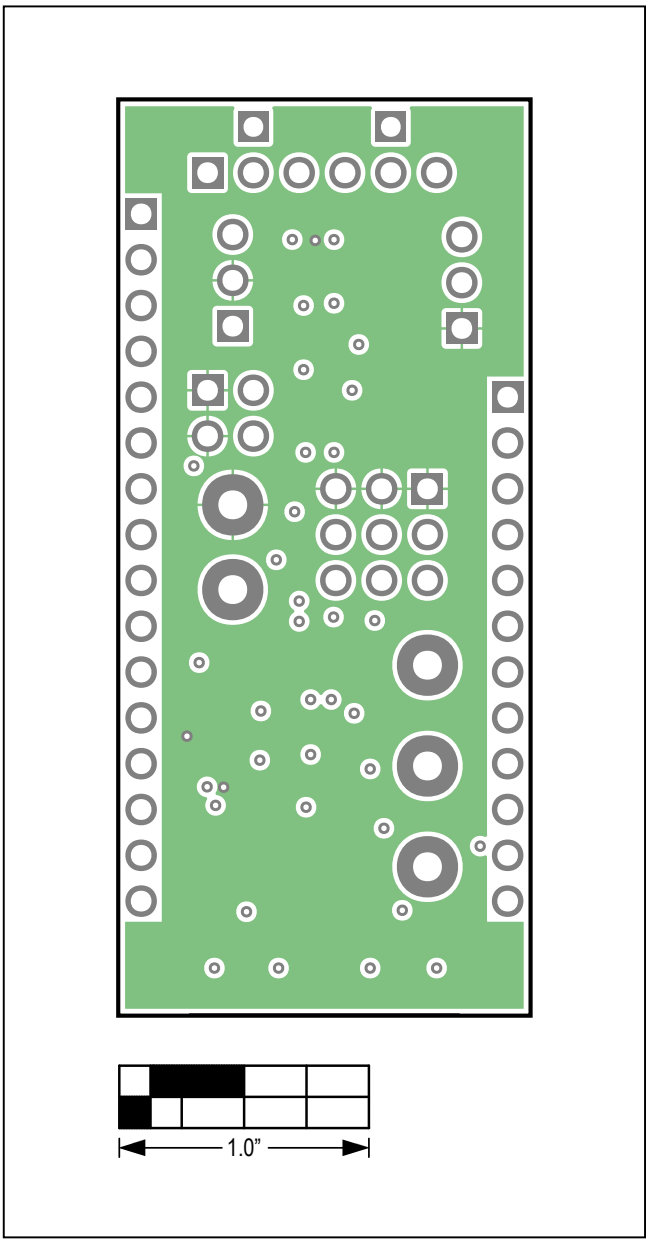


MAX41470 EV Kit PCB Layout – Top Layer

MAX41470 EV Kit PCB Layout Diagrams (continued)

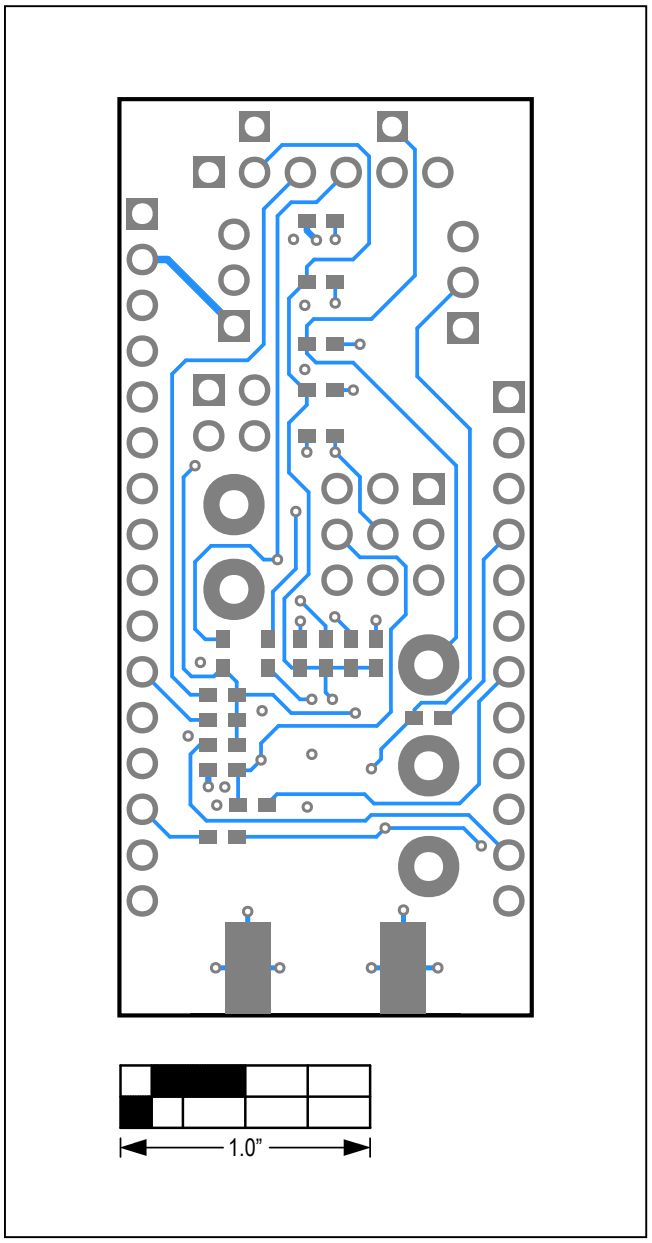


MAX41470 EV Kit PCB Layout – Internal 2

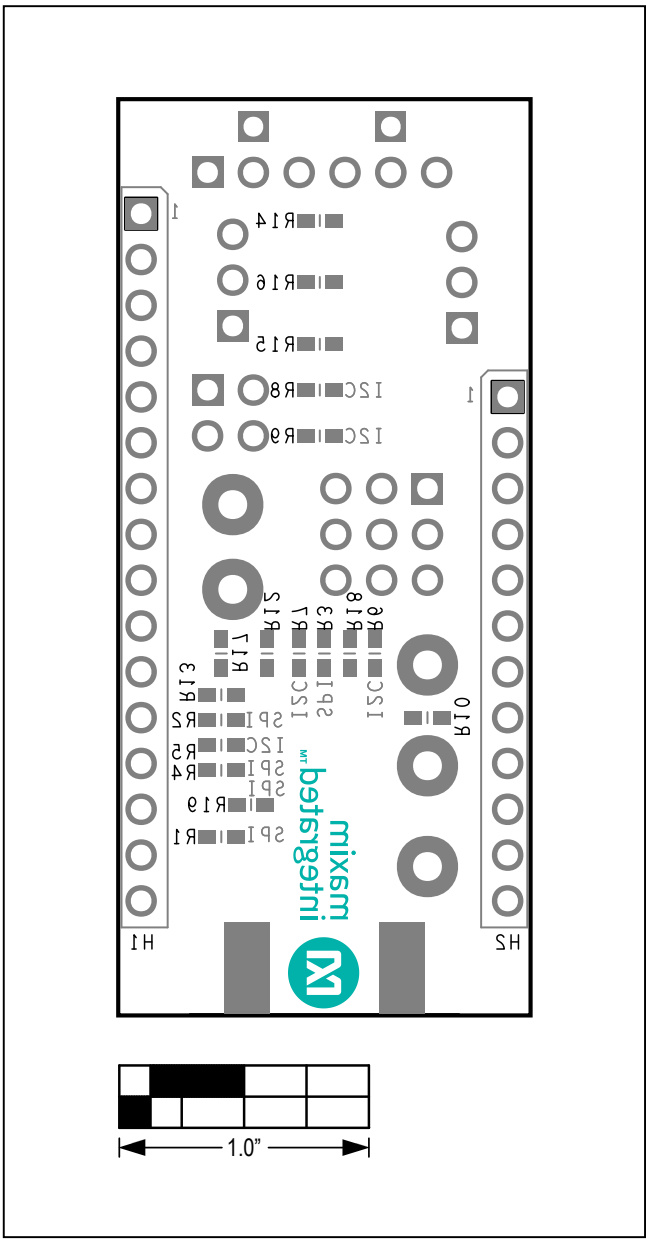


MAX41470 EV Kit PCB Layout – Internal 3

MAX41470 EV Kit PCB Layout Diagrams (continued)



MAX77655 EV Kit PCB Layout – Bottom Layer



MAX77655 EV Kit PCB Layout – Silkscreen Bottom

Revision History

REVISION NUMBER	REVISION DATE	DESCRIPTION	PAGES CHANGED
0	9/20	Initial release	—

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