2250074 iSSR User Manual

2G ENGINEERING

Contents

REVISION HISTORY

REV	DATE	Editor	DESCRIPTION
A	2026-01-28	JL	Initial Release

Initial Setup

Hardware

The iSSR provides multiple options for communications interfaces, including RS-232, RS-485, CAN, USB, and Ethernet. It can draw power from an external DC supply or directly from the USB port. See the iSSR datasheet or interface drawing for information on pinouts.

Software

2G Engineering's JAMBUI software is recommended for initial setup and configuration. If you did not receive a copy of it with the iSSR documentation package, it can be downloaded from https://www.2g-eng.com/subsea-products/software-downloads/.

Configuration

The default configuration of the iSSR's communication interfaces are listed below.

Serial communication

The iSSR's serial interface defaults to RS-485 mode, 115200 baud, 8N1, Modbus RTU mode, with a Modbus device ID of 1.

Ethernet communication

The iSSR's Ethernet interface defaults to DHCP mode for address assignment.

USB communication

The iSSR's USB interface supports the CDC ACM standard and appears as a virtual serial port when connected to a computer. Baud rate, parity, and bit count are ignored, the default Modbus mode is RTU, and the default Modbus device ID is 1.

Daisy Chaining

Up to 64 iSSRs can be connected in a daisy chain, allowing control and feedback through a single device designated as the master iSSR.

Connection

To connect multiple iSSRs in a daisy chain configuration, a standard 5-conductor, 1mm pitch flat flex cable (FFC) with same-side contacts such as Parlex P/N 100R5-102B must be used. To make a connection between iSSRs, gently lift the daisy chain connector latch using a tweezers or small screwdriver, insert the FFC with the contacts facing between the two connectors, then press down on the latch to engage it. Connections between iSSRs must always be between opposite connectors, i.e. the left daisy chain connector on one iSSR must always be connected to the right daisy chain connector on a different iSSR, and vice-versa.

Configuration

The master iSSR must be designated as such by writing its MASTER_MODE register to 1 and saving the configuration. All other devices in the chain do not require any additional configuration. Having multiple devices configured as the master in a single chain will result in communications errors and will prevent devices in the chain from operating properly.

Addressing

Every time a chain of iSSRs is powered up, the master iSSR will automatically discover the other devices in the chain. It will first probe the devices on the left daisy chain port, followed by the devices on the right daisy chain port. It will then assign an index number to each device on the chain. Devices will always be numbered from left to right, starting at 1, regardless of the master's position in the chain. After index numbers are assigned, all devices will briefly show their index number on their LEDs as a binary value so you can quickly confirm that they have been numbered as expected.

Connecting

Using Serial or USB

- Connect the iSSR to your computer using either a direct USB connection, or through an RS-232 or RS-485 adapter as needed.
- If connected using a direct USB connection, the iSSR will draw power from the USB port. Otherwise, you will need to apply power using the GND and VIN pins on the comms connector. Note: the USB port can be used to supply power to the iSSR even if you are using a different interface to communicate with it.
- Open JAMBUI. Go to the Setup>Connection page and select RS232 or RS485 mode. Click the button to the right of the Comm Ports field to scan for available serial ports, and select the one that your device is connected to. Unless you have changed settings on the iSSR, the defaults for baud rate, bit count, parity, and mode should not need to be changed.
- Click the "Reconnect" button below the settings area, or the "Not Connected" button in the upper left half of the window to open the connection to the device.

Using Ethernet

- Connect the iSSR to your network using a suitable cable.
- Apply power to the iSSR using either the USB connector or the GND and VIN pins on the comms connector.
- Open JAMBUI. Go to the Setup>Connection page and select Ethernet mode.
- If you know the iSSR's IP address, you can enter it in the field, otherwise click the magnifying glass to open the device search window. Click the Search button, then double-click the IP address to use it.
- Click the "Reconnect" button below the settings area, or the "Not Connected" button in the upper left half of the window to open the connection to the device.

Using JAMBUI

Once you have connected to the iSSR and JAMBUI has downloaded the register map from the device, you will be taken to the Register view.

This view shows all the registers in the device. Each register corresponds to a single control, configuration, or status value. Registers are broadly grouped by function, but you may find them easier to navigate using the following tips:

- There is a list of tags on the left side of the window. Selecting a tag will show you only the registers related to that tag.
- The registers can be sorted alphabetically by clicking on the "Name" column header.
- You can choose to view a custom subset of registers by pressing the F4 key and selecting only the registers you wish to see in the Query column. Multiple registers can be enabled by selecting them and holding down the control key while clicking in the Query column. Press the F4 key again to hide the registers which you have not selected.

Data in the register list will be continually be updated as long as "Audit by Schedule, Repeated" is selected under the audit menu. The update rate can be changed using the Audit Schedule field on the Setup>Connection page. In general, the data will update more quickly when fewer registers are displayed.

Data can be written to any R/W register by either double-clicking on the register in the list, highlighting it, and pressing F2, or right-clicking on it and selecting "Write". After entering your desired value and clicking OK, the new value will be queued up for transmission to the device. It may take several seconds for this to actually occur, depending on the amount of queued data that is pending.

Operating the iSSR

Now that communications have been established with the iSSR, you will be able to control it and view its status. Some registers have special characteristics, which will be listed before their description in this document. The types of special registers are listed below:

Configuration

Parameters labeled *Configuration* will be lost when the device is reset or power-cycled unless the parameters are saved by writing a 1 to the WRITE_CONFIG register. Factory default settings can be restored by writing a 1 to the LOAD_DEFAULTS register. The restored default settings will also be lost on the next reset or power cycle unless saved using WRITE_CONFIG.

Remote Configuration

Parameters labeled *Remote Configuration* are similar to those labeled *Configuration*, but they can be sent to either the local device or one of the iSSRs on the chain, depending on which one is selected in the CFG_DEVICE_SEL register. These parameters will be lost when the device in question is reset or power cycled unless they are saved by writing the device number to the CHAIN_WRITE_CONFIG register.

Calibration

Parameters labeled *Calibration* will be lost when the device is reset or power-cycled unless the parameters are saved by writing a 1 to the WRITE_CAL register. Factory default calibration can be restored by writing a 1 to the RESET_CAL register. The restored calibration will also be lost on the next reset or power cycle unless saved using WRITE_CAL.

Communication

Ethernet

Ethernet configuration changes will take effect after a reset or power cycle.

IP_CONFIG

Configuration. Bit field which configures various aspects of Ethernet and IP communication. The following bits are defined:

Bit	Description
0	Use DHCP – if set, the system will obtain an IP address via DHCP. Otherwise, it will use the static IP configuration defined in the IP4_ADDR, IP4_MASK, and IP4_GW registers.
1-14	Reserved
15	Disable Ethernet – if set, the Ethernet subsystem will be completely disabled. This will save a significant amount of power (~10mA at 24V).
16-31	Reserved

IP4_ADDR

Configuration. Specifies the static IP address to use if DHCP is not enabled. JAMBUI will display this value as 4 integers representing the 4 octets of the IPv4 address.

IP4_MASK

Configuration. Specifies the static subnet mask to use if DHCP is not enabled. JAMBUI will display this value as 4 integers representing the 4 octets of the IPv4 address.

IP4_GW

Configuration. Specifies the static default gateway to use if DHCP is not enabled. JAMBUI will display this value as 4 integers representing the 4 octets of the IPv4 address.

Serial

SERIAL_BAUD

Configuration. Specifies the baud rate that will be used for RS-232 and RS-485 communication. Allowable values are 80 to 400000. Default is 115200. Changes take effect immediately.

SERIAL_TERM

Configuration. Specifies if a 120 ohm termination resistor is enabled on the system's serial port. Termination is enabled if the corresponding bit is set. Changes take effect immediately.

Bit	Termination
0	Port 1
1-15	Reserved

PHY1_MODE

Configuration. Specifies the active physical layer mode for the comms pins on the external connector. Changes take effect immediately. The following values are defined:

Value	Port Mode
0	RS-232
1	RS-485 (Default)
2	CAN

3-65535	Reserved

Modbus

MODBUS_ADDR

Configuration. Specifies a Modbus address for the device. Address will take effect immediately.

MODBUS_MODE

Configuration. Specifies the Modbus mode to use over serial (RS-232 or RS-485) interfaces. The new mode will take effect immediately on all serial interfaces.

Value	Modbus Mode
0	Modbus RTU (Default)
1	Modbus ASCII
2-65535	Reserved

HOST_COMM_TIMEOUT

Configuration. Specifies the amount of time, in milliseconds, with no packets received on any interface, which will set the communications timeout fault.

MODBUS_ZERODELAY

Configuration. By default, on serial interfaces, the Modbus standard requires a certain amount of delay time between when a message is sent to a device, and a response is returned. Setting this setting to 1 will disable that artificial delay time. If your host system can handle this, you will be able to significantly increase your overall throughput.

MODBUS_NUM_REMAP_REGS

Configuration. The system allows you to remap registers from anywhere in the map to a single cluster at the end of it. This allows you to read/write a group of registers in a single Modbus transaction, which can significantly increase your throughput. This register sets how many registers can be remapped into this remap region. Allowable values are 0 to 100, inclusive.

SSR Control Configuration

MASTER_MODE

Configuration. Set to 1 to configure this iSSR as a master device, i.e. one that will control other iSSRs over the iSSR's daisy chain bus. Only one iSSR can be configured as a master device on a single iSSR bus. If set to 0, this iSSR will function as a slave or standalone device.

STARTUP_MODE

Configuration. Configures how the iSSR will set its output when it is first powered on, assuming the POWERON trigger source is enabled (see the SW_POWERON register). The current behavior of the POWERON trigger is to only assert the main switch trigger. Thus if you need the precharge circuit to activate before the main switch at power on, you will need to select Equivalent or Main Only as the trigger mode. The following startup modes are supported:

Value	Startup state
0	Output off (Default)
1	Output on

2	Restore previous state
3-65535	Reserved

SW_EXCLUSION

Configuration. Specifies how the iSSR will handle multiple trigger sources. The following modes are supported:

Value	**Exclusion Mode**
0	LAST: The last enabled trigger to change state will set the output state (Default).
1	OR: If any enabled trigger is true, the output will turn on.
2	AND: The output will only turn on if all enabled triggers are true.
3-65535	Reserved

SW_POWERON

Configuration. Enables the Power On switch trigger. When set to 1, the iSSR will trigger its output when it is first powered on, depending on the value set in the STARTUP_MODE register and the configured exclusion mode.

SW_SERIAL

Configuration. Enables the Serial switch trigger. When set to 1, the iSSR will trigger its output when commanded over RS-232 or RS-485, depending on the configured exclusion mode.

SW_CAN

Configuration. Enables the CAN switch trigger. When set to 1, the iSSR will trigger its output when commanded over CAN bus, depending on the configured exclusion mode.

SW_USB

Configuration. Enables the USB switch trigger. When set to 1, the iSSR will trigger its output when commanded over the USB connection, depending on the configured exclusion mode.

SW_ETHERNET

Configuration. Enables the Ethernet switch trigger. When set to 1, the iSSR will trigger its output when commanded over the Ethernet connection, depending on the configured exclusion mode.

SW_ANALOG

Configuration. Enables the Analog switch trigger. When set to 1, the iSSR will trigger its output when commanded via its analog trigger input pins, depending on the configured exclusion mode.

MAIN_VIH

Configuration. Specifies the voltage threshold to assert (in active high mode) or deassert (in active low mode) the main (“ON”) analog input pin.

MAIN_VIL

Configuration. Specifies the voltage threshold to deassert (in active high mode) or assert (in active low mode) the main (“ON”) analog input pin.

PRECHG_VIH
Configuration. Specifies the voltage threshold to assert (in active high mode) or deassert (in active low mode) the precharge ("PCHG") analog input pin.

PRECHG_VIL
Configuration. Specifies the voltage threshold to deassert (in active high mode) or assert (in active low mode) the precharge ("PCHG") analog input pin.

MAIN_EDGE_TRIG
Configuration. Specifies if the main analog input pin is edge sensitive or level sensitive and its polarity.

Value	Trigger Mode	Active Level
0	Analog input is rising edge sensitive (Default).	High
1	Analog input is high level sensitive	
2	Analog input is falling edge sensitive	Low
3	Analog input is low level sensitive	
4-65535	Reserved	

PRECHG_EDGE_TRIG
Configuration. Specifies if the main precharge input pin is edge sensitive or level sensitive and its polarity.

Value	Trigger Mode	Active Level
0	Analog input is rising edge sensitive (Default).	High
1	Analog input is high level sensitive	
2	Analog input is falling edge sensitive	Low
3	Analog input is low level sensitive	
4-65535	Reserved	

SSR Device Configuration

CFG_DEVICE_SEL
Selects the remote device (1-64) to configure when accessing Remote Configuration registers. A value of 0 can also be used to select the device you are communicating with.

CFG_DEVICE_READY
Reports 1 if the Remote Configuration registers are up to date. Will report 0 temporarily when changing devices using the CFG_DEVICE_SEL register, or if the specified device number is not available.

DEVICE_NAME
Remote Configuration. Sets the name of the device. This is a user-defined text string for informational purposes only; it does not have any effect on device operation.

LED_MODE
Remote Configuration. Sets the mode of the iSSR's 4 external LEDs. Modes are TBD.

LED_BRIGHTNESS

Remote Configuration. Sets the brightness of the iSSR's 4 external LEDS, on a scale from 0 to 100%.

TRIG_MODE

Remote Configuration. Configures the triggering mode of the iSSR. The following triggering modes are available:

Number	Name	Description
0	Independent	Precharge trigger and Main trigger operate fully independently. The precharge timeout value is enforced - If the precharge trigger is asserted continually, it will automatically shut off after the timeout expires, and will need to be triggered again. Asserting the main trigger at any point will energize the main switch and will automatically disable the precharge circuit. Releasing the precharge trigger at any time will also disable the precharge circuit if it was active. Setting the precharge timeout to 0 will allow the precharge to remain on indefinitely.
1	Independent Sequential	Precharge trigger and Main trigger operate semi-independently. When the precharge trigger is asserted, the precharge circuit will be energized immediately. After the precharge timeout expires, the main switch will be automatically engaged, at which point the precharge circuit will be turned off. The main trigger can be asserted at any time to turn on the main switch and end the precharge cycle early. The precharge trigger can be deasserted during the precharge phase to cancel the startup sequence and disable the precharge circuit. The main switch will remain energized as long as either the precharge or main triggers are asserted.
2	Equivalent	Precharge trigger and Main trigger are logically OR-ed together. When either trigger is asserted, the precharge circuit will be energized immediately. After the precharge timeout expires, the main switch will be automatically engaged, at which point the precharge circuit will be turned off. The main switch will remain on as long as either trigger remains asserted. Both triggers must be released to either cancel the precharge or disengage the main switch.
3	Main Only	Precharge trigger is disabled, but the system will otherwise function identically to the Equivalent mode.

PRECHG_DC

Remote Configuration. Specifies the duty cycle to use when precharging. Currently only 100% is supported.

PRECHG_TIMEOUT

Remote Configuration. Specifies the timeout, in milliseconds, to be used when precharging. The exact behavior of this parameter depends on the selected trigger mode. Set this value to 0 to disable the precharge timeout.

PSTART_I_LIMIT

Remote Configuration. Specifies the current limit to be used when pulse starting.

PSTART_TIMEOUT
Remote Configuration. Specifies the time, in milliseconds, that pulse start will be active before switching to normal operation mode. Set this value to 0 to disable pulse start.

PSTART_MIN_INT
Remote Configuration. Specifies the minimum time interval, in microseconds, between pulses when pulse starting.

DEFAULT_SW_DIR
Remote Configuration. Specifies the default direction(s) to allow current flow. The value selected here will be copied to the SW_DIR register on system startup. After startup, this register will have no additional effects. Available options are as follows:

Number	Description
0	Reserved
1	Forward current only
2	Reverse current only
2	Bidirectional current

OC_FWD
Remote Configuration. Specifies the soft overcurrent limit in the forward direction in Amps.

OC_REV
Remote Configuration. Specifies the soft overcurrent limit in the reverse direction in Amps.

MAX_TEMP
Remote Configuration. Specifies the maximum allowable operating temperature in degrees C.

MIN_TEMP
Remote Configuration. Specifies the minimum allowable operating temperature in degrees C.

Control

SW_CTRL
Write a 1 to trigger the main switch of the device you are directly communicating with. If the interface used to write this register is not enabled, it will be ignored. For example, if you are communicating with the iSSR using an RS-232 interface, the SW_SERIAL register must be set to 1 for this register to have any effect. The exact behavior of the iSSR when the main switch is triggered will depend on the triggering mode selected in TRIG_MODE.

SW_DIR
Selects the allowed direction of current flow for the device you are directly communicating with. Available options are as follows:

Number	Description
0	Reserved
1	Forward current only

2	Reverse current only
2	Bidirectional current

PRECHG_CTRL
Write a 1 to trigger the precharge switch of the device you are directly communicating with. If the interface used to write this register is not enabled, it will be ignored. For example, if you are communicating with the iSSR using an RS-232 interface, the SW_SERIAL register must be set to 1 for this register to have any effect. The exact behavior of the iSSR when the precharge switch is triggered will depend on the triggering mode selected in TRIG_MODE.

SW_CTRL_ALL
Allows control of the main switch states for all the iSSRs in a chain with a single register write. This register is organized as an array of bytes, which should be treated as a bit field. Each bit represents the on/off state of a single iSSR in the chain. The bit field is organized in little-endian order, so byte 0, bit 0 represents the first iSSR, byte 0 bit 1 represents the second iSSR, byte 1 bit 0 represents the eighth iSSR, and so on. The exact behavior of the iSSR when the main switch is triggered will depend on the triggering mode selected in TRIG_MODE.

SW_DIR_ALL
Allows control of the main switch directions for all the iSSRs in a chain with a single register write. This register is organized as an array of bytes, which should be treated as a bit field. Each pair of bits represents the direction control state of a single iSSR in the chain. Bit definitions are the same as for the SW_DIR register. The bit field is organized in little-endian order, so byte 0, bit 0-1 represents the first iSSR, byte 0 bit 2-3 represents the second iSSR, byte 1 bit 0-1 represents the eighth iSSR, and so on.

PRECHG_CTRL_ALL
Allows control of the precharge switch states for all the iSSRs in a chain with a single register write. This register is organized as an array of bytes, which should be treated as a bit field. Each bit represents the precharge state of a single iSSR in the chain. The bit field is organized in little-endian order, so byte 0, bit 0 represents the first iSSR, byte 0 bit 1 represents the second iSSR, byte 1 bit 0 represents the eighth iSSR, and so on. The exact behavior of the iSSR when the precharge switch is triggered will depend on the triggering mode selected in TRIG_MODE.

SW_CTRL_n
Write a 1 to trigger the main switch of the n-th device in the chain. The exact behavior of the iSSR when the main switch is triggered will depend on the triggering mode selected in TRIG_MODE.

SW_DIR_n
Controls the main switch direction for the n-th device in the chain. Values used in this register are the same as in the SW_DIR register.

PRECHG_CTRL_n
Write a 1 to trigger the precharge switch of the n-th device in the chain. The exact behavior of the iSSR when the precharge switch is triggered will depend on the triggering mode selected in TRIG_MODE.

Status/Feedback

MY_INDEX

Reports the index of the device that you are communicating with if the device is part of a chain. Note that communicating directly with a device in a chain other than the master device is generally not recommended.

CHAIN_LENGTH

On a master device, reports the number of devices in the chain.

IS_CHAINED

Reports 1 if the device is part of a chain.

SYSTEM_FAULTS

Reports general system fault conditions. Fault bits are set when the condition occurs and are cleared by writing a 1 to them unless otherwise noted. As of firmware version 0.4, bits in this register are defined as follows:

Bit	**Description**
0	Power-on reset. Indicates that the system has rebooted.
1	General internal logic error.
2	Firmware update error.
3	Parameter fault. Indicates that a command sent to the unit contained invalid data.
4	Host communications timeout. Indicates that no valid communications packets have been received on any valid communication interface in the interval defined by the HOST_COMM_TIMEOUT register.
5	File system failure.
6	Configuration file missing or damaged.
7	Metaconfiguration file missing or damaged.
8	Product definition file missing or damaged.
9	Register remapping file missing or damaged.
10	Calibration data file missing or damaged.
11	Statistics tracking data missing or damaged.
12	Persistent system state data missing or damaged.
13	MOSFET over temperature limit.
14	MOSFET under temperature limit.
15	Auxiliary sensor over temperature limit.
16	Auxiliary sensor under temperature limit.
17	Board over temperature limit.
18	Board under temperature limit.
19	Hard overcurrent limit exceeded.
20	Soft overcurrent limit exceeded.
21-23	*Reserved*
24	Precharge timeout expired.
25	Daisy chain configuration error or device discovery failure.
26	Multiple devices are configured as masters in a single daisy chain.

PROD_VALID
Reports 1 if the device's Product Definition file is valid.

MCFG_VALID
Reports 1 if the device's Metaconfiguration file is valid.

MONITOR_PASS/MONITOR_FAIL
Reports the general status of various subsystems within the iSSR. A bit set in the MONITOR_PASS register means that the associated subsystem is known to have been functional at some point since the iSSR was powered on. A bit set in the MONITOR_FAIL register means that the associated subsystem is known to have encountered an error at some point since the iSSR was powered on. It is possible for the same bit in both MONITOR_PASS and MONITOR_FAIL to be set at the same time. Bits will remain set until they are manually cleared by writing 1 to them. The following bits are defined for the iSSR:

Bit	**Description**
0	System initialization
1	Flash memory
2	File system
3	Configuration
4	Metaconfiguration
5	Product definition
6	Statistics
7	Persistent system state
8	Calibration
9	System watchdog
10	Ethernet
11	LEDs
12-31	*Reserved*

Analog Status/Feedback

SWITCH_I
Reports the current through the switch terminals. Positive current is defined as current flowing from the left to the right terminal of the iSSR.

TRIG_V
Reports the voltage applied to the main trigger pin.

PRECHG_TRIG_V
Reports the voltage applied to the precharge trigger pin.

V_IN
Reports the power supply input voltage to the iSSR.

TEMP1
Reports the temperature of the main control board inside the iSSR.

TEMP2
Reports the base plate temperature of the iSSR.

SWITCH_I_n
Reports the current through the switch terminals of the n-th device in the chain.

V_IN_n
Reports the power supply input voltage of the n-th iSSR in the chain.

TEMP1_n
Reports the main control board temperature of the n-th iSSR in the chain.

TEMP2_n
Reports the base plate temperature of the n-th iSSR in the chain.

Switch Status/Feedback

SW_STATE
Reports the actual state of the main switch of the device you are currently communicating with. The actual state may differ from the requested state if the iSSR has tripped off due to a fault condition such as over current or over temperature.

PRECHG_STATE
Reports the actual state of the precharge switch of the device you are currently communicating with. The actual state may differ from the requested state depending the configured startup mode and timeouts.

SW_STATE_ALL
Reports the actual state of the main switches of all iSSRs in the chain. This register is organized as an array of bytes, which should be treated as a bit field. Each bit represents the on/off state of a single iSSR in the chain. The bit field is organized in little-endian order, so byte 0, bit 0 represents the first iSSR, byte 0 bit 1 represents the second iSSR, byte 1 bit 0 represents the eighth iSSR, and so on. The actual state may differ from the requested state if the iSSR has tripped off due to a fault condition such as over current or over temperature.

PRECHG_STATE_ALL
Reports the actual state of the precharge switches of all iSSRs in the chain. This register is organized as an array of bytes, which should be treated as a bit field. Each bit represents the precharge state of a single iSSR in the chain. The bit field is organized in little-endian order, so byte 0, bit 0 represents the first iSSR, byte 0 bit 1 represents the second iSSR, byte 1 bit 0 represents the eighth iSSR, and so on. The actual state may differ from the requested state depending the configured startup mode and timeouts.

SW_STATE_n
Reports the actual state of the main switch of the n-th device in the chain. The actual state may differ from the requested state if the iSSR has tripped off due to a fault condition such as over current or over temperature.

PRECHG_STATE_n
Reports the actual state of the precharge switch of the n-th device in the chain. The actual state may differ from the requested state depending the configured startup mode and timeouts.

System Operations

SYS_RESET
Write 1 to reset the iSSR you are communicating with. This will result in the iSSR's main switch turning off. All setting changes that have not been saved using the WRITE_CONFIG register will be lost. With the current firmware release, the iSSR will also lose its place in the daisy chain, if it is part of one.

WRITE_CONFIG
Write 1 to save configuration changes to non-volatile storage on the iSSR you are communicating with. If you do not do this, changes will be lost the next time the iSSR is reset or power cycled.

LOAD_DEFAULTS
Write 1 to load default settings on the iSSR you are communicating with. This may include changes to the serial PHY type, baud rate, and Modbus address, so you may loose communication with the device until you correct these settings on your host device. The new settings will be lost the next time the iSSR is power cycled or reset unless they are saved using the WRITE_CONFIG register.

PUSH_CONFIG
Copies all *Remote Configuration* registers from the master SSR to slave SSRs. Write 0 to copy to all connected slave SSRs, or write the device number (1-64) to copy to a specific one.

CHAIN_WRITE_CONFIG
Saves configuration changes to non-volatile storage on a slave SSR. Write 0 to save changes on all connected slave SSRs, or write the device number (1-64) to save changes on a specific one.

CHAIN_LOAD_DEFAULTS
Load default configuration settings on a slave SSR. Write 0 to load default settings on all connected slave SSRs, or write the device number (1-64) to load default settings on a specific one.

IDENTIFY
Write a device number (1-64) to flash the LEDs on that device in order to confirm its position on the chain.

Product Definition File
The product definition file (cfg/prod.ini) is used to configure various aspects of the iSSR that should only need to be configured once and are not expected to need to be changed in normal operation. Changes to parameters in this file will be applied after a reset. If the prod.ini file is valid, the backup file (prod.bak) will automatically be updated to match it on startup. Extreme caution should be exercised when editing the product definition file; damage to the system may result from improper edits to this file. Such damage is not covered under warranty.

Updating Firmware

As new features and functionality improvements become available, 2G Engineering will provide firmware updates. These firmware updates can be user-installed using JAMBUI.

1. Connect the iSSR to your computer via RS232, RS485, CAN, or Ethernet as described earlier in this document.
2. Make sure Programming is enabled under the File > File Transfer menu.
3. Switch to the File Transfer tab.
4. On the right pane of the window, click on the file path and navigate to the location of the firmware update.
5. Select the firmware update file and click on the "<-- Copy" button.
6. Allow the transfer to complete.
7. Highlight the firmware update file on the right pane of the window and click the CRC+ button.
8. Highlight the firmware update file on the left pane of the window and click the CRC+ button.
9. Compare the two CRC values in the output section and verify that they match.
10. Click the Burn0 button and click OK. Wait for the update to complete.

The ability to update multiple devices in a chain from the master device will be provided in a future firmware version.

Pulse Start

For some loads (e.g. large capacitive loads with high quiescent draw), the precharge circuit on the iSSR may be insufficient to raise the load voltage to a safe operating level. On these systems, the pulse start feature is available as an alternative way to start up your load. The pulse start feature works by turning on the main switch repeatedly, tripping off when it hits a specified current limit and restarting after a specified time interval. After a timeout period, the system will then move to the main switch on state, at which time the normal operating current limits will be in effect. At this point, the system will latch off if the current limits are exceeded.

The pulse start approach generates a significant amount of heat and EMI, and is discouraged unless you have no other alternatives. If you plan on using this feature, we recommend contacting us to discuss your application before implementing it.

If enabled, the pulse start stage is inserted between the precharge and main switch phases during startup.

LED Status Indication

The 4 LEDs on the front of the iSSR are used to indicate system state. The possible LED patterns and their corresponding meanings are listed below:

LED State				Description
1	2	3	4	
●	❍	❍	❍	Master module is discovering iSSR modules to the left
❍	❍	❍	●	Master module is discovering iSSR modules to the right

❍	❍	❍	🟢	Module is displaying its address
⚪	⚪	⚪	⚪	Module is idle
🔵	🔵	🔵	🔵	Precharge switch is on
🟢	🟢	🟢	🟢	Main switch is on
🔴 🟢 🔵 ⚪	🔴 🟢 🔵 ⚪	🔴 🟢 🔵 ⚪	🔴 🟢 🔵 ⚪	Identification was triggered for this module