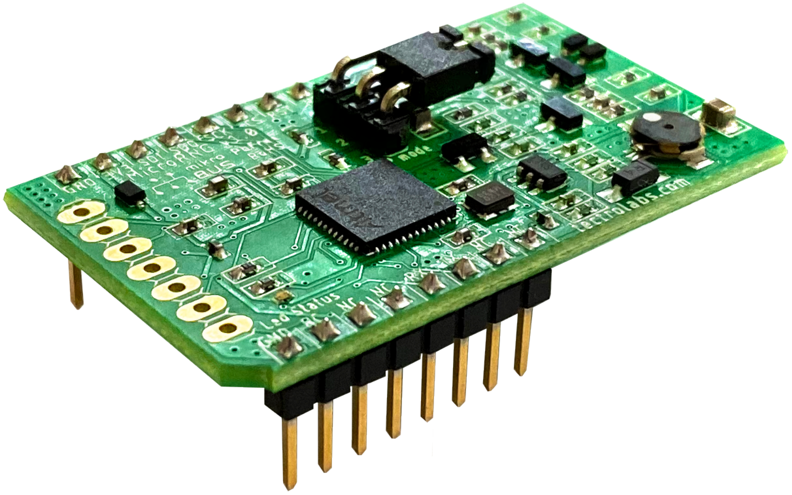
March 2024 – Rev 1.2

MicroRNG Datasheet



Description

The **MicroRNG** is a hardware (true) random number generator device that can be used in embedded systems as a reliable entropy source. It can interface with microcontrollers or microprocessors (mainboards) with integrated circuits and modules through a mikroBUS™ socket using an SPI or 2-wire UART interface.

The MicroRNG is designed to comply with NIST SP 800-90B: “Recommendation for the Entropy Sources Used for Random Bit Generation,” and can generate random numbers at a rate up to 1 Mbps in SPI mode and up to 1.5 Mbps in UART mode.

This device embeds comprehensive diagnostics and monitoring features used for validating internal random physical sources and components in real-time. It maintains an internal status, referenced in this documentation by ‘Status byte,’ which can be retrieved using the device API to check the health of the generator. Therefore, it is important to retrieve and check the status byte for error conditions when communicating with the device.

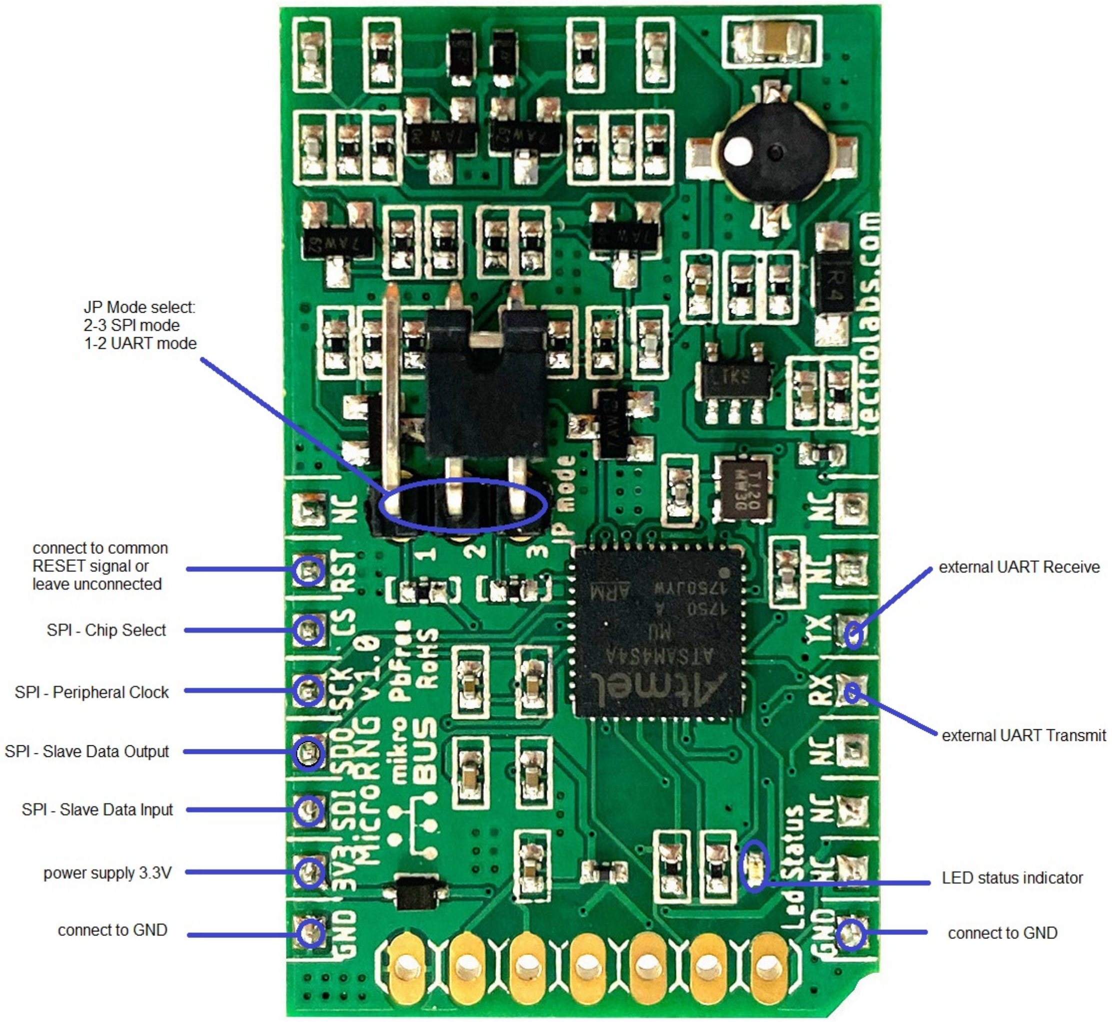
Generated random numbers can be retrieved from the MicroRNG through an API which is implemented over an SPI interface or a 2-wire UART interface. The device operates at 3.3V, consumes no more than 340mW, and has a start-up time of less than 700 milliseconds.

Features

- Core
    - Two noise sources based on Zener diodes
    - SPI or 2-wire UART operation modes
    - mikroBUS™ socket connectivity
    - Up to 1 Mbps data transfer speed using device API through SPI interface
    - Up to 1.5 Mbps data transfer speed using device API through 2-wire UART interface
    - Embedded statistical tests
        - Validation of all tests using ‘Fault Injection’ technique at device startup
        - ‘Frequency Table Inspection’ - runs at device startup
        - ‘Repetition Count Test’ - continuously runs on raw random byte stream
        - ‘Adaptive Proportion Test’ - continuously runs on raw random byte stream
    - Embedded post-processing algorithms available
        - Linear Corrector (P. Lacharme)
        - available only in 2-wire UART operation mode: SHA1, SHA2, SHA512 and HMAC160
- PbFree / RoHS compliant
- Pin-to-pin compatible with mikroBUS™ add-on board
- Power supply and consumption
    - 3.3V supply voltage
    - max 340mW consumption in normal operation mode and 125mW in sleep mode
- I/O - High-level voltage on any input pins should not exceed 3.3V
- Dimensions
    - 42.9 mm by 25.4 mm, same as the mikroBUS™ add-on board standard size ‘M’
- Sleep mode
    - Noise sources are shut down through the device API when in this mode
- Peripherals
    - SPI - Serial Peripheral Interface
        - 8-bits data transfer format
        - Slave mode
        - SCK pin - Peripheral Clock
            - Clock Polarity - the inactive state value of SCK is logic level zero
            - It requires a half the SCK clock period delay from CS falling edge (activation) to the first valid SCK transition
            - Clock Phase - data is changed on the rising edge of SCK and captured on the falling edge
            - 60 MHz max clock speed
        - CS pin - Chip Select, also known as Slave Select or NSS
            - The active level is logic level zero
            - Used by Master device for activating MicroRNG device (slave)
        - SDI pin - Slave Data Input
            - Used by Master device for sending commands to MicroRNG device (slave)
        - SDO pin - Slave Data Output
            - Used by Master device for receiving responses from MicroRNG device (slave)
        - Master device is required to wait for 6.8 microseconds between data transfers
    - 2-wire UART - Serial Communication Interface, 8 bits, no parity, 1 stop bit
        - Baud rates supported: 1200, 2400, 4800, 9600, 19200, 38400, 150000, 187500, 200000, 250000, 300000, 375000, 468750, 500000, 600000, 1000000, 1250000, 1500000, 1875000, 2500000, 3000000, 4000000, 4800000, 5000000
        - RX pin - external UART Transmit
            - Used by MicroRNG device for receiving commands from external UART device
        - TX pin - external UART Receive
            - Used by MicroRNG device for transmitting responses to external UART device

Operation

The MicroRNG can operate in either SPI or 2-wire UART mode, which is selected with the 'JP Mode' switch located on the board (see picture below). When the device is powered on or restarted, the 'JP Mode' switch configuration determines which interface is to be used. The SPI mode is selected by setting 'JP Mode' switch jumper to the 2-3 position, while the 2-wire UART mode is selected by setting the jumper to the 1-2 position.



The MicroRNG utilizes random noise generated from two independent and reliable sources based on the electrical noise produced by an avalanche breakdown effect in Zener diodes. The electrical noise generated by each random source is independently amplified, filtered, and converted into digital values. The random bytes produced by both noise sources are combined and processed internally with a Linear Corrector, a technique first proposed by P. Lacharme ("Post-Processing Functions for a Biased Physical Random Number Generator". In: FSE 2008. LNCS, vol. 5086, pp. 334-342. Springer-Verlag. 2008), using a 0.5 compression rate. Alternatively, when operation in 2-wire UART mode, the raw random bytes can be processed internally with one of the following hashing methods: SHA1, SHA2, SHA512, or HMAC160.

As soon as the MicroRNG is powered on or the RST signal is asserted, it will begin running start-up health tests and will become ready in less than 700 milliseconds. If any errors are detected when running start-up health tests, the ‘LED Status’ indicator will blink several times and the ‘Status byte’ will be updated with an error code.

Communication with the MicroRNG device can be achieved through a set of API calls implemented over the SPI or 2-wire UART interface. Detailed information about the implemented API can be found in the *Serial Peripheral Interface (SPI) mode* and *2-wire UART mode* sections.

Pinout Description

The table shown below explains the usage of each pin on the MicroRNG device board in connection to pinout on the mikroBUS™ socket.

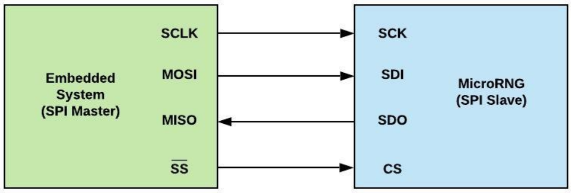
Pin number	mikroBUS™ pin	MicroRNG pin	Notes
1	AN	NC	Not connected
2	RST	RST	Connect to RESET signal or leave it unconnected
3	CS	CS	SPI Chip Select Input, active level is logical zero
4	CSK	CSK	SPI Clock Input, inactive state is logic level zero
5	MISO	SDO	SPI Master Input / MicroRNG Data Output
6	MOSI	SDI	SPI Master Output / MicroRNG Data Input
7	+3.3V	3.3V	VCC-3.3V power
8	GND	GND	Reference Ground
9	GND	GND	Reference Ground
10	+5V	NC	Not connected
11	SDA	NC	Not connected
12	SCL	NC	Not connected
13	TX	RX	UART Transmit, MicroRNG UART Receive
14	RX	TX	UART Receive, MicroRNG UART Transmit
15	INT	NC	Not connected
16	PWM	NC	Not connected

Serial Peripheral Interface (SPI) operation mode

The MicroRNG implements a serial peripheral interface (SPI) that enables external devices in Master mode to communicate with the MicroRNG device. To select the SPI operation mode, the MicroRNG board should be configured with the 'JP Mode' switch jumper set to 2-3 position. In this mode, the MicroRNG device acts as an SPI 'Slave.' A MicroRNG device is selected when the master asserts its CS signal.

The MicroRNG SPI interface consists of two data lines and two control lines:

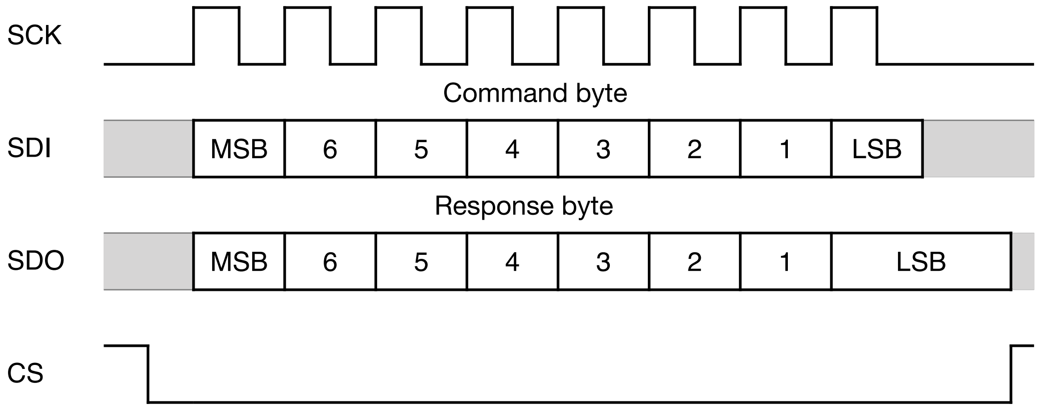
- SCK (Peripheral Clock)
  - This input signal is driven by the Master device and regulates the flow of the data bits
- CS (Chip Select, also known as Slave Select or NSS)
  - This input signal allows Master to select the MicroRNG (slave)
- SDI (Slave Data Input)
  - Used by Master for sending commands to MicroRNG (slave)
- SDO (Slave Data Output)
  - Used by Master for receiving responses from MicroRNG (slave)



MicroRNG SPI embedded characteristics:

- 8 bits data transfer format
- SCK
  - 60 MHz max clock speed provided by the master device
  - Clock Polarity - the inactive state value of SCK is logic level zero
  - It requires a half the SCK clock period delay from CS falling edge (activation) to the first valid SCK transition
  - Clock Phase - data is changed on the rising edge of SCK and captured on the falling edge
- CS
  - The active level is logic level zero
- The master device is required to insert a delay for 6.8 microseconds after each transfer and before removing the CS if needed.

SPI data transfer timing diagram



When an SPI data transfer is in initiated, the master device sends an 8-bit command to the MicroRNG device while the MicroRNG sends an 8-bit response (for the previous command) to the master device. The master device, therefore, will need two transfers to send a command and receive its response value.

The MicroRNG implements the following commands accessible through the SPI device API:

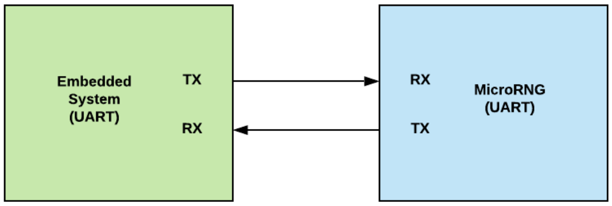
CMD as decimal number	CMD as ASCII character	Description
108	l	Generates random byte value by applying embedded Linear Corrector (P. Lacharme) processing on raw, random bytes. The response byte represents the random byte value.
114	r	Generates a raw (unprocessed) random byte value. It should only be used for verification or when used with external post-processing implementations. The response byte represents the raw, random byte value.
115	s	Retrieves the MicroRNG device internal status byte. The response byte represents the device status byte value.
116	t	Retrieves the internal SPI transfer ID which is incremented with each transfer. Used for validation of the SPI communication between the master device and the MicroRNG device. Primarily used in the development phase to detect SPI misconfigurations of the master device. The response byte represents the latest SPI transfer ID.
68	D	Shuts down the random noise sources of the MicroRNG device. This command is used to enable the sleep mode when the device is not in use. The response byte value is the decimal number 200.
85	U	Starts up the random noise sources of the MicroRNG device. This command is used to leave the sleep mode to restore normal functionality of the device. The response byte value is the decimal number 0.
82	R	Resets the UART configuration baud rate to the factory default value 19200. It will take effect after the device is powered off and turned back on or after RST signal assertion. This command is used to restore the factory default MicroRNG device UART baud rate when it has been misconfigured through the 2-wire UART API. The response byte represents the device status byte value.

2-wire UART operation mode

Asynchronous serial communication in the MicroRNG device is implemented using 2 wires. To select 2-wire UART operation mode, the MicroRNG board should be configured by setting the ‘JP Mode’ switch jumper to the 1-2 position.

MicroRNG 2-wire UART interface consists of two data lines:

- MicroRNG pin ‘RX’ - MicroRNG UART Receive (remote UART Transmit)
  - This line is used by the MicroRNG device for receiving commands from the remote UART device
- MicroRNG pin ‘TX’ - MicroRNG UART Transmit (remote UART Receive)
  - This line is used by the MicroRNG device for sending data to the remote UART device

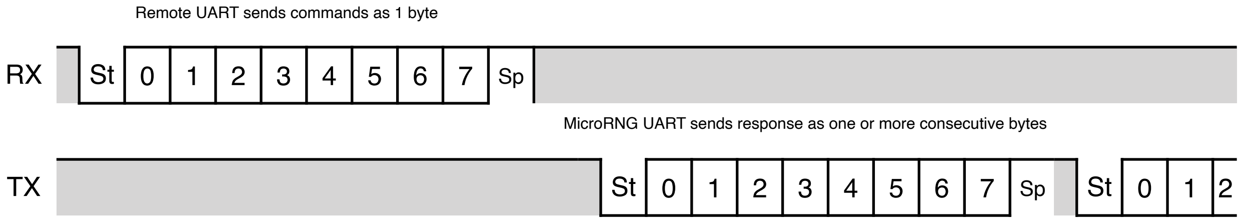


MicroRNG 2-wire UART embedded characteristics:

- 8 bits data transfer format
- One stop bit
- No parity bit
- Baud rates supported: 1200, 2400, 4800, 9600, 19200, 38400, 150000, 187500, 200000, 250000, 300000, 375000, 468750, 500000, 600000, 1000000, 1250000, 1500000, 1875000, 2500000, 3000000, 4000000, 4800000, 5000000
- Factory default baud rate: 19200

The MicroRNG device is shipped with default baud rate 19000. The default baud rate can be modified through the MicroRNG 2-wire UART API. A remote UART device communicates with a MicroRNG device using 1, 2 and 3 byte commands. As soon as the command is sent out to the MicroRNG device, the remote UART device should immediately be waiting for the expected response.

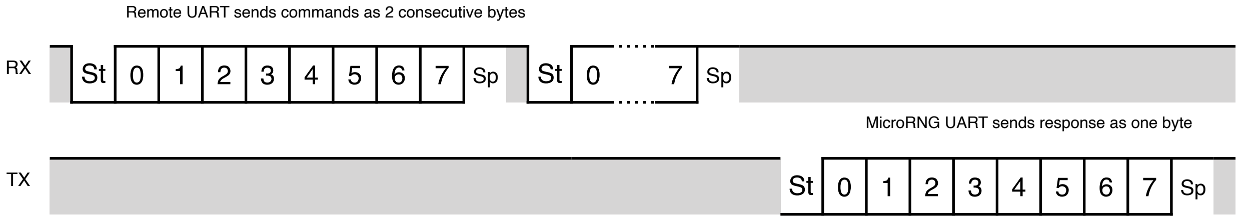
2-wire UART - one byte command timing diagram


Remote UART sends commands as 1 byte
RX
St
0
1
2
3
4
5
6
7
Sp
MicroRNG UART sends response as one or more consecutive bytes
TX
St
0
1
2
3
4
5
6
7
Sp
St
0
1
2

2-wire UART API - one byte commands

CMD as decimal number	CMD as ASCII character	Bytes in response	Description
97	a	1	Generates random byte value by applying embedded SHA1 processing on raw random bytes. The response byte represents the final random byte value.
98	b	1	Generates random byte value by applying embedded SHA2 processing on raw random bytes. The response byte represents the final random byte value.
99	c	1	Generates random byte value by applying embedded SHA512 processing on raw random bytes. The response byte represents the final random byte value.
100	d	1	Generates random byte value by applying embedded Linear Corrector (P. Lacharme) processing on raw random bytes. The response byte represents the final random byte value.
101	e	1	Generates random byte value by applying embedded HMAC160 processing on raw random bytes. The response byte represents the final random byte value.
102	f	1	Generates a raw (unprocessed) random byte value. It should only be used for verification or when used with external post-processing implementations. The response byte represents the raw random byte value.
83	S	1	Retrieves the MicroRNG device internal status byte. The response byte represents the device status byte value.
68	D	1	Shuts down the random noise sources of the MicroRNG device. This command is used to enable sleep mode when the device is not in use. The response byte value is the decimal number 200.
85	U	1	Starts up the random noise sources of the MicroRNG device. This command is used to leave the sleep mode to restore normal functionality of the device. The response byte value is the decimal number 0.
118	v	4	Retrieves the MicroRNG device version as 3 ASCII characters followed by the status byte value.
109	m	7	Retrieves the MicroRNG device model as 6 ASCII characters followed by the status byte value.
115	s	31	Retrieves the MicroRNG device serial number as 30 ASCII characters followed by the status byte value.
71	G	1	The response byte value is the current device baud rate profile number, a decimal number between 1 and 24.

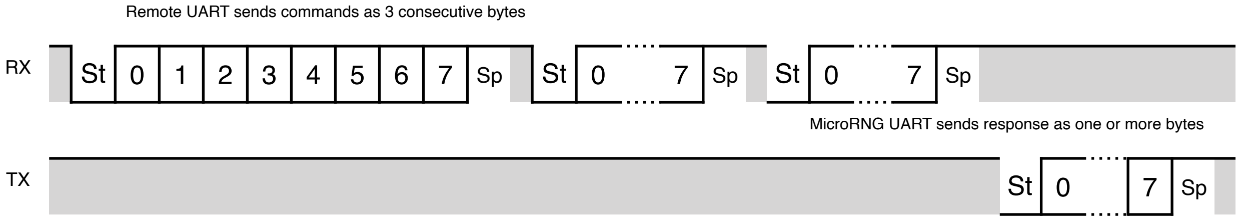
2-wire UART - two byte command timing diagram


Remote UART sends commands as 2 consecutive bytes
RX
St 0 1 2 3 4 5 6 7 Sp
St 0 7 Sp
MicroRNG UART sends response as one byte
TX
St 0 1 2 3 4 5 6 7 Sp

2-wire UART API - two byte commands

CMD as decimal number	CMD as ASCII character	CMD argument	Bytes in response	Description
66	B	baud rate profile number	1	This 2 byte command is used for changing the MicroRNG 2-wire UART baud rate. The first byte is the command, and the second byte is the argument that represents the new baud rate profile number. The profile number should be a decimal number between 1 to 24. The new baud rate will only take effect after the MicroRNG device is powered off and on or after RST signal is asserted. There should not be delays longer than 90 milliseconds between moments when sending each byte as part of command, otherwise the device will ignore the command. The response byte represents the device status byte value.

2-wire UART - three byte command timing diagram


Remote UART sends commands as 3 consecutive bytes
RX
St 0 1 2 3 4 5 6 7 Sp
St 0 7 Sp
St 0 7 Sp
MicroRNG UART sends response as one or more bytes
TX
St 0 7 Sp

2-wire UART API - three byte commands

CMD as decimal number	CMD as ASCII character	CMD arguments	Description
52	4	16-bit unsigned integer	This 3 bytes command is for retrieving true random bytes from the device using embedded Linear Corrector (P. Lacharme). The 16-bit unsigned integer represents the requested amount of bytes (max value is 50,000), the lower byte is sent first and then the higher byte. The MicroRNG device expects to receive all three bytes within a reasonable amount of time. There should not be delays longer than 90 milliseconds between moments when sending each byte, otherwise the device will ignore the command. The last byte in the response is the status byte.
114	r	16-bit unsigned integer	This 3 bytes command is used to retrieve raw (unprocessed) random bytes from the MicroRNG device. The 16-bit unsigned integer represents the requested amount of bytes (max value is 50,000), the lower byte is sent first and then the higher byte. The MicroRNG device expects to receive all three bytes within a reasonable amount of time. There should not be delays longer than 90 milliseconds between moments when sending each byte, as part of command, otherwise the device will ignore the command. The last byte in the response is the status byte.
49	1	16-bit unsigned integer	This 3 bytes command is used to retrieve true random bytes from the MicroRNG device by hashing the low bias random byte stream with SHA-160. The 16-bit unsigned integer represents the requested amount of bytes (max value is 50,000), the lower byte is sent first and then the higher byte. The MicroRNG device expects to receive all three bytes within a reasonable amount of time. There should not be delays longer than 90 milliseconds between moments when sending each byte, as part of command, otherwise the device will ignore the command. The last byte in the response is the status byte.
50	2	16-bit unsigned integer	This 3 bytes command is used to retrieve true random bytes from the MicroRNG device by hashing low bias random byte stream with SHA-256. The 16-bit unsigned integer represents the requested amount of bytes (max value is 50,000), the lower byte is sent first and then the higher byte. The MicroRNG device expects to receive all three bytes within a reasonable amount of time. There should not be delays longer than 90 milliseconds between moments when sending each byte, as part of command, otherwise the device will ignore the command. The last byte in the response is the status byte.

CMD as decimal number	CMD as ASCII character	CMD arguments	Description
51	3	16-bit unsigned integer	This 3 bytes command is used to retrieve true random bytes from the MicroRNG device by hashing low bias random byte stream with SHA-512. The 16-bit unsigned integer represents the requested amount of bytes (max value is 50,000), the lower byte is sent first and then the higher byte. The MicroRNG device expects to receive all three bytes within a reasonable amount of time. There should not be delays longer than 90 milliseconds between moments when sending each byte, as part of command, otherwise the device will ignore the command. The last byte in the response is the status byte.
104	h	16-bit unsigned integer	This 3 bytes command is used to retrieve true random bytes from the MicroRNG device with HmacSHA256 post-processing. The 16-bit unsigned integer represents the requested amount of bytes (max value is 50,000), the lower byte is sent first and then the higher byte. The MicroRNG device expects to receive all three bytes within a reasonable amount of time. There should not be delays longer than 90 milliseconds between moments when sending each byte, as part of command, otherwise the device will ignore the command. The last byte in the response is the status byte.

A MicroRNG device, in 2-wire UART operation mode, can only operate at baud rates defined in the table below:

2-wire UART baud rate profile numbers

MicroRNG baud rate profile number	Corresponding baud rate
1	1200
2	2400
3	4800
4	9600
5	19200 (factory default)
6	38400
7	150000
8	187500
9	200000
10	250000
11	300000
12	375000
13	468750
14	500000
15	600000
16	1000000
17	1250000
18	1500000
19	1875000
20	2500000
21	3000000
22	4000000
23	4800000
24	5000000

The MicroRNG internal status values and descriptions

Device status byte decimal value	Description
0	Device is healthy and ready to accept commands.
1	Device internal ‘Repetition Count Test’ has failed. Device is not operable and should not be used. This error condition indicates an internal problem with the device.
2	Device internal ‘Adaptive Proportion Test’ has failed. Device is not operable and should not be used. This error condition indicates an internal problem with the device.
3	Device encountered a serial transmission communication error when operating in in 2-wire UART mode. Usually, this error occurs when there is a mismatch of serial communication parameters between the MicroRNG UART and the remote UART.
4	Device internal ‘Frequency Table Test’ has failed. Device is not operable and should not be used. This error condition indicates an internal problem with the device.
5	Remote UART tried to set an invalid baud rate profile number in 2-wire UART operation mode. The profile number should be a decimal number between 1 to 24.
6	Device received an invalid command when operating in SPI mode.
200	Device internal random noise sources are turned off, and random numbers cannot be generated. Remote device should send the appropriate command to turn random noise sources ‘on’ to resume MicroRNG normal operation.

Operation Conditions

Parameter	Min	Type	Max	Unit
Ambient Temperature Range	0	-	35	°C
DC supply	3.3	3.3	3.3	V
Input Low-level Voltage	-0.3	-	0.8	V
Input High-level Voltage	1.96	-	3.3	V
Output High-level Voltage	2.4	-	-	V
Master SPI clock frequency	-	20	60	MHz
UART baud rate	1200	-	5000000	Baud Rate
Actual data transfer rate in SPI mode	-	-	1	Mbps
Actual data transfer rate in 2-Wire UART mode	-	-	1.5	Mbps
Device start-up time	-	400	700	Milliseconds

Revision History

Date	Changes
14-Dec-2019	Added MicroRNG connection diagrams to sections “Serial Peripheral Interface (SPI) operation mode” and “2-wire UART operation mode”.
	Updated section “Operation Conditions”.
9-March-2024	Fixed grammar typos.

Table of Contents