

Gowin(GW1NSER-4C) SHA-3 (Secure Hash Algorithm-3) User Guide

Version 1.1
2021/05

Contents

1	Introduction	4
2	Features	4
3	Directory Structure	4
4	Functions.....	4
4.1	SHAKE128.....	5
4.2	SHAKE256.....	5
4.3	SHA3_224.....	5
4.4	SHA3_256.....	5
4.5	SHA3_384.....	6
4.6	SHA3_512.....	6
5	Reference Design	6

Revision History

Version	Date	Reason for Changes
1.1	2021/05	Initial

1 Introduction

This document “Gowin SHA-3 User Guide” describes the application of SHA-3 (Secure Hash Algorithm-3) on the Gowin MCU platform. It was packaged into SHA3 library, which has been successfully verified and tested on the Gowin GW1NSER-4C board DK-START-GW1NSER4C-QN48G V1.1. It is based on KECCAK (<https://github.com/XKCP/XKCP>), which was selected by NIST as the winner of the public SHA-3 Cryptographic Hash Algorithm Competition. The SHA-3 family consists of four cryptographic hash functions and two extendable-output functions.

2 Features

A hash function is a function on binary data for which the length of the output is fixed. The input to a hash function is called the message, and the output is called the digest or hash value. The digest often serves as a condensed representation of the message. The four SHA-3 hash functions are named SHA3-224, SHA3-256, SHA3-384 and SHA3-512; in each case, the suffix after the dash indicates the fixed length of the digest, e.g., SHA3-256 produces 256-bit digests.

An extendable-output function (XOF) is a function on binary data in which the output can be extended to any desired length. The two SHA-3 XOFs are named SHAKE128 and SHAKE256. The suffixes indicate the security strengths that these two functions can generally support.

3 Directory Structure

The directory structure of SHA3 library “SHA3/sha_lib” is as table 3-1.

Table 3-1 Directory structure

Level-1	Level-2	Files	Description
doc	-	– Gowin(GW1NSER-4C)_SHA3_UserGuide.pdf	Gowin(GW1NSER-4C) SHA3 user guide.
library	include	– align.h – brg_endian.h – KeccakP-1600-SnP.h – KeccakSponge.inc – KeccakSponge-common.h – KeccakSpongeWidth1600.h – SimpleFIPS202.h – SnP-Relaned.h	Header files for SHA3.
	lib	– libgowin_sha3_soc.a	Library for SHA3
	template	– main.cpp	Template for SHA3

4 Functions

The six function prototypes can be found in the header file :
GW1NSER_4C/SHA3/src/sha3_lib/library/include/SimpleFIPS202.h.

4.1 SHAKE128

/** Implementation of the SHAKE128 extendable output function (XOF) [FIPS 202].

* @param output Pointer to the output buffer.
* @param outputByteLen The desired number of output bytes.
* @param input Pointer to the input message.
* @param inputByteLen The length of the input message in bytes.
* @return 0 if successful, 1 otherwise.
*/

int SHAKE128(unsigned char *output, size_t outputByteLen, const unsigned char *input, size_t inputByteLen);

4.2 SHAKE256

/** Implementation of the SHAKE256 extendable output function (XOF) [FIPS 202].

* @param output Pointer to the output buffer.
* @param outputByteLen The desired number of output bytes.
* @param input Pointer to the input message.
* @param inputByteLen The length of the input message in bytes.
* @return 0 if successful, 1 otherwise.
*/

int SHAKE256(unsigned char *output, size_t outputByteLen, const unsigned char *input, size_t inputByteLen);

4.3 SHA3_224

/** Implementation of SHA3-224 [FIPS 202].

* @param output Pointer to the output buffer (28 bytes).
* @param input Pointer to the input message.
* @param inputByteLen The length of the input message in bytes.
* @return 0 if successful, 1 otherwise.
*/

int SHA3_224(unsigned char *output, const unsigned char *input, size_t inputByteLen);

4.4 SHA3_256

/** Implementation of SHA3-256 [FIPS 202].

```

* @param output    Pointer to the output buffer (32 bytes).
* @param input     Pointer to the input message.
* @param inputByteLen  The length of the input message in bytes.
* @return 0 if successful, 1 otherwise.
*/

```

```
int SHA3_256(unsigned char *output, const unsigned char *input, size_t inputByteLen);
```

4.5 SHA3_384

```
/** Implementation of SHA3-384 [FIPS 202].
```

```

* @param output    Pointer to the output buffer (48 bytes).
* @param input     Pointer to the input message.
* @param inputByteLen  The length of the input message in bytes.
* @return 0 if successful, 1 otherwise.
*/

```

```
int SHA3_384(unsigned char *output, const unsigned char *input, size_t inputByteLen);
```

4.6 SHA3_512

```
/** Implementation of SHA3-512 [FIPS 202].
```

```

* @param output    Pointer to the output buffer (64 bytes).
* @param input     Pointer to the input message.
* @param inputByteLen  The length of the input message in bytes.
* @return 0 if successful, 1 otherwise.
*/

```

```
int SHA3_512(unsigned char *output, const unsigned char *input, size_t inputByteLen);
```

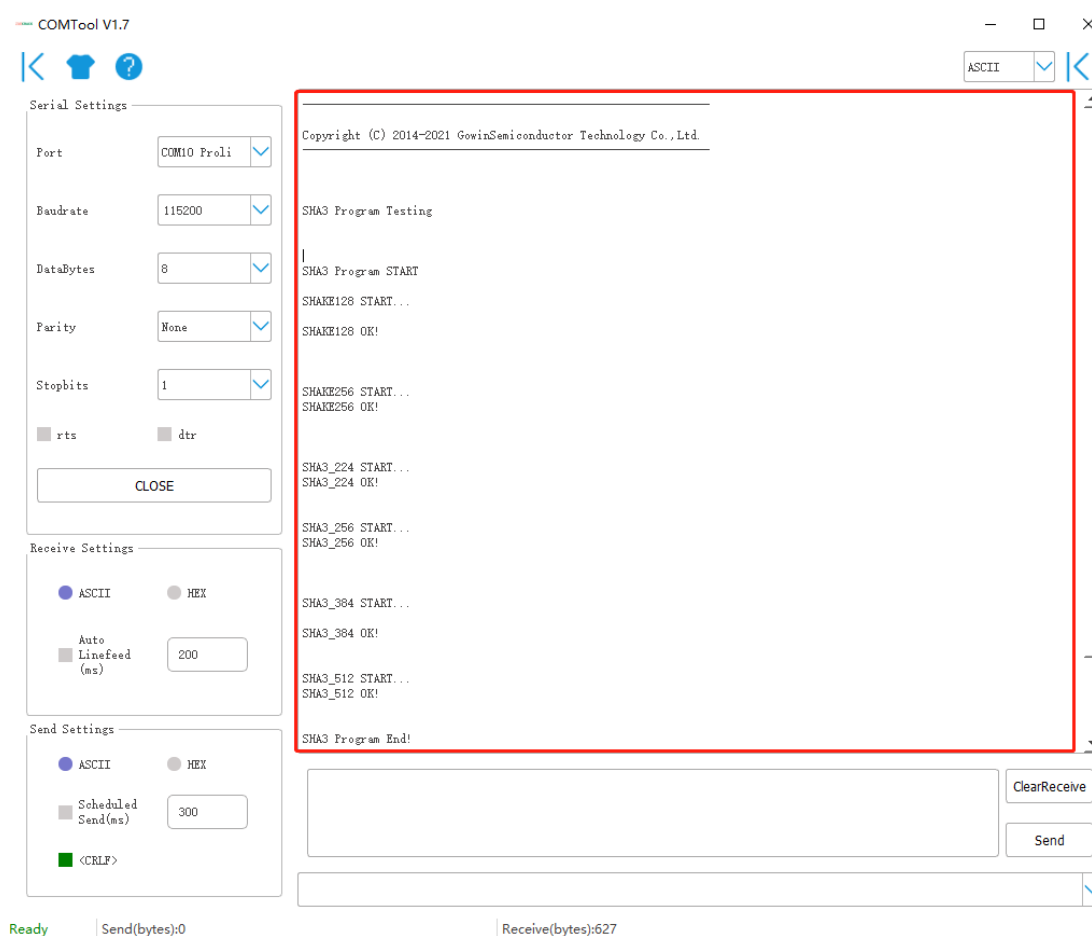
5 Reference Design

We tried to run all the six SHA-3 functions on a Gowin GW1NSER-4C board DK-START-GW1NSER4C-QN48G V1.1 and the result is as figure 5-1.

This is reference design as below.

- GW1NSER_4C/SHA3/ref_design/MCU_RefDesign/SHA3
- GW1NSER_4C/SHA3/ref_design/FPGA_RefDesign/DK_START_GW1NSER4C_QN48G_V1.1/secur
e_fpga

Figure 5-1 SHA3 Running Result



The UART output showed “SHAKE128 OK!”, “SHAKE256 OK!”, “SHA3_224 OK!”, “SHA3_256 OK!”, “SHA3_384 OK!” and “SHA3_512 OK!”, which means that the output exactly matches the expected result.