

BRS-100-GW1NR9

An FPGA development board



BRisbaneSilicon

www.brisbanesilicon.com.au

Table of Contents

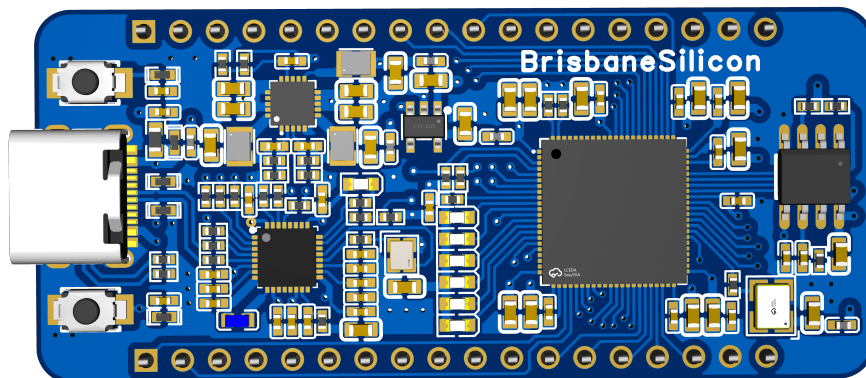
1	About	3
1.1	Image.....	3
2	Board Specifications	4
3	PC Setup.....	5
3.1	Prerequisites	5
3.2	Optional	5
3.3	License Setup.....	6
3.4	Environment Variables	8
4	Example Project.....	10
4.1	Building.....	11
4.2	Program.....	13
4.3	Board Demonstration Mode.....	15
4.4	Extending.....	17

1 About

The BRS-100-GW1NR9 is a FPGA development board. This document details the following:

1. Specifications of the board.
2. Setting up your PC to build and program the board.
3. Building and programming the provided example project to the board.
4. Extending the example project.
5. Example project roadmap.

1.1 Image



1 BRS-100-GW1NR9

2 Board Specifications

- [GOWIN GW1NR-9](#)¹ FPGA.
 - 8MB internal PSRAM.
 - Internal bitstream Flash with AUTO BOOT.
- Non-volatile memory.
 - 8 MB User Flash.
- Peripherals
 - 16 x User GPIO.
 - 6 x User LED.
 - 2 x User Pushbuttons.
- USB-C connector
 - UART over USB (FTDI)
 - JTAG over USB
- 1 x Power LED.
- 3.3V and 5V rails.
- [Example project](#)² with a fully scripted workflow.
 - Includes a guided setup of the build prerequisites and environment.
 - Demonstrates board features (PSRAM, Flash, Peripherals etc).
 - [Extensible](#)³, i.e. user can add their own application specific RTL modules.
 - No GUI-based program required.
 - Simple [build](#)⁴ and [program](#)⁵ workflow stages.

1. <https://au.mouser.com/new/gowin/gowin-gw1nr-9-fpgas>

2. <https://github.com/BrisbaneSilicon/BRS-100-GW1NR9>

3. <https://github.com/BrisbaneSilicon/BRS-100-GW1NR9?tab=readme-ov-file#development>

4. <https://github.com/BrisbaneSilicon/BRS-100-GW1NR9?tab=readme-ov-file#build>

5. <https://github.com/BrisbaneSilicon/BRS-100-GW1NR9?tab=readme-ov-file#program>

3 PC Setup

Fulfill the below prerequisites to setup your PC to build and program FPGA firmware to the BRS-100-GW1NR9.

3.1 Prerequisites

1. A PC running an x64 compatible, Debian-based flavour of Linux.
 - Other flavours of Linux may work but aren't officially supported.
 - We recommend [Ubuntu](#)⁶.
 - Support for Windows is coming soon! See section [roadmap](#)⁷ below.
2. An installation of GOWIN EDA V1.9.12.
 - Available from the official GOWIN EDA [download page](#)⁸ or direct link to [V1.9.12](#)⁹.
 - You may need to first register as a GOWIN member [here](#)¹⁰.
 - On Linux, please ensure that GOWIN EDA is installed to one of the following directories (or be prepared to modify the build script):
 - `$HOME/Applications /opt/gowin /opt/GOWIN /opt/Gowin $HOME/Documents/Applications/`
3. A free license for GOWIN EDA. See section [License Setup](#)¹¹ below.
4. A copy of the [BRS-100-GW1NR9](#)¹² repository.
 - Launch a terminal program.
 - Navigate to the directory in which you wish to host the BRS-100-GW1NR9 repository.
 - `git clone https://github.com/BrisbaneSilicon/BRS-100-GW1NR9.git`

3.2 Optional

1. An FTDI driver is required if you wish to communicate with the BRS-100-GW1NR9 via UART. On most Linux distributions they are part of the default installation of the OS. On Windows, you may need to install the driver from [here](#)¹³. FTDI also provide installation guides, available [here](#)¹⁴.

6. <https://ubuntu.com/>

7. <https://github.com/BrisbaneSilicon/BRS-100-GW1NR9#roadmap>

8. https://www.gowinsemi.com/en/support/download_eda/

9. https://cdn.gowinsemi.com.cn/Gowin_V1.9.12_linux.tar.gz

10. <https://www.gowinsemi.com/en/member/>

11. <https://github.com/BrisbaneSilicon/BRS-100-GW1NR9#license-setup>

12. <https://github.com/BrisbaneSilicon/BRS-100-GW1NR9>

3.3 License Setup

There are two options (both free) for licensing GOWIN EDA.

1. Using a local license file.
2. Using a floating license server.

Option (1) requires applying for a license from GOWIN [here](#)¹⁵. It can take up to a few working days for GOWIN to provide you with a license, which will be valid for a period of one year. Option (2) requires the same initial step if you wish to host your own license server. Alternatively you can point the GOWIN license manager at a public license server; this is likely the quickest path forwards. See [Public License Servers](#)¹⁶ for a list of public license servers.

3.3.1 Linux

Launch a bash terminal and perform the following:

1. Change directory to the GOWIN IDE installation 'bin' directory.
 - `cd <GOWIN IDE install directory>/IDE/bin/`
2. Run the licensing manager.
 - `./license_config_gui`
 - Alternatively you can run the GOWIN IDE `./gw_ide` and click 'Help' - 'Manage License'.
3. Either point the licensing manager at your local license file (Option 1 above) or a floating license server.
4. Press 'Check' to validate the license.
 - If the license has been successfully validated, it should produce a popup window **INFO** with the message **Server is OK**.
5. Click 'Save' to save your license setup.



Sometimes the first license check (step 4) will fail - simply repeat the step to validate the license.

13. <https://ftdichip.com/drivers/vcp-drivers/>

14. <https://ftdichip.com/document/installation-guides/>

15. <https://www.gowinsemi.com/en/support/license/>

16. <https://github.com/BrisbaneSilicon/BRS-100-GW1NR9/tree/public?tab=readme-ov-file#public-license-servers>

3.3.2 Windows

Support for Windows is coming soon! See section [roadmap](#)¹⁷ below.

3.3.3 Public License Servers

A list of public GOWIN EDA license servers is below. These are community reported and might not be official or stable (see the previous instructions on how to check their validity).

IP Address	Port
106.55.34.119	10559
43.128.7.128	10559

17. <https://github.com/BrisbaneSilicon/BRS-100-GW1NR9/tree/public?tab=readme-ov-file#roadmap>

3.4 Environment Variables

3.4.1 Linux

1. Prior to building the project, set 'LD_LIBRARY_PATH' to point at the GOWIN IDE libraries at `<GOWIN IDE install directory>/IDE/lib`. For example:

```
export LD_LIBRARY_PATH=/opt/Gowin/Gowin_V1.9.12_linux/IDE/lib
```

2. Prior to programing the BRS-100-GW1NR9 board, add `<GOWIN IDE install directory>/Programmer/bin/` to your PATH. For example:

```
export PATH=$PATH:/opt/Gowin/Gowin_V1.9.12_linux/Programmer/bin/
```



When attempting to build the project, you may get the following error:

```
symbol lookup error: /<system library path>/libfontconfig.so.1: undefined
symbol: FT_Done_MM_Var.
```

To rectify this, setup the following environment variable as part of your pre-build environment, substituting '<system library path>' to be as per your specific error message:

```
export LD_PRELOAD=$LD_PRELOAD:/<system library path>/libfreetype.so
```

When attempting to program the board, you may get the following error:

```
programmer_cli: /<GOWIN EDA install path>/Programmer/bin/libz.so.1: version
`ZLIB_1.2.3.4' not found (required by /lib/x86_64-linux-gnu/libpng16.so.16)
```

To rectify this, setup the following environment variable as part of your pre-build environment:

```
export LD_PRELOAD=$LD_PRELOAD:/<system library path>/libz.so.1
```

Note that the '<system library path>' may differ from the build error - you might need to use the 'ldd' program to locate libz.so.1.



If you are going to be building the project often, we recommend (on Linux) you add a bash alias for the above environment variables to your '.bashrc'. For example:

```
alias gowin_ini='export LD_LIBRARY_PATH=/opt/Gowin/Gowin_V1.9.12_linux/IDE/lib; \
export PATH=$PATH:/opt/Gowin/Gowin_V1.9.12_linux/Programmer/bin/; export \
LD_PRELOAD=/lib/x86_64-linux-gnu/libfreetype.so'
```

Then to initialise the build environment, you can simply run `gowin\_ini` from bash.

3.4.2 Windows

Support for Windows is coming soon!

4 Example Project

The example project allows the user to build an FPGA bitstream and program it onto the [BRS-100-GW1NR9](#)¹⁸ FPGA development board.

The example project can be built in [board demonstration](#)¹⁹ mode, which demonstrates the hardware functionality offered by the BRS-100-GW1NR9.

It can also be extended by the user to include their custom, application-specific, RTL modules.

The example project workflow is fully scripted (fetch, build, program); it does not require the use of a GUI-based program at any point.

18. <https://brisbanesilicon.com.au/devboards/brs-100-gw1nr9>

19. <https://github.com/BrisbaneSilicon/BRS-100-GW1NR9/blob/public/README.md#board-demonstration>

4.1 Building

After fulfilling all of the prerequisites, and initializing your build environment, you are ready to build the BRS-100-GW1NR9 firmware! Simply perform the following:

4.1.1 Linux

```
cd <BRS-100-GW1NR9 repository directory>/BRS-100-GW1NR9/build
./build.sh
```

That's all there is to it!

The build script supports various customizations via command line arguments. To view the full set of supported command line arguments, simply perform the following:

```
./build.sh -h
```

The most commonly used are listed below.

Build Argument	Description
-y, --list_supported_system_clock_frequencies	List the supported system clock frequencies and exit.
-k, --clock_frequency FREQUENCY_MHZ	Use a frequency of FREQUENCY_MHZ for the system clock (default 51 MHz).
-u, --uart_baud	Set the user comms baud rate (default 115200).
-b, --board_demonstration	Perform a build of the board demonstration bitstream.
-a, --clean_all_platforms	Perform cleanup of the entire build and exit.
-p, --proj_only	Only generate the project file, then exit. Useful if the user wishes to utilize GOWIN IDE.
-s, --synth_only	Only proceed with build until synthesis is complete, then exit. Useful to check FPGA utilization, timing etc.

4.1.2 Windows

Support for Windows is coming soon!

4.2 Program

After building the BRS-100-GW1NR9 firmware you are ready to program it to the board! It is worth noting that this stage also programs the internal non-volatile bitstream flash, so your firmware will auto-load after board power on!

After plugging the board into your PC via the USB-C cable, simply perform the following:

4.2.1 Linux

The '<this repository directory>' is the directory in which you performed Step (4) of [prerequisites](#)²⁰ i.e. the directory in which you cloned this repository.

```
cd <this repository directory>/BRS-100-GW1NR9/prog
./program.sh
```

That's all there is to it!



The 'program' script will trigger a build the BRS-100-GW1NR9 firmware (with no command line customization arguments) if it detects it hasn't already been built.

Again, the program script supports various customizations via command line arguments. To view the full set of supported command line arguments, simply perform the following:

```
./program.sh -h
```

The most commonly used are listed below.

Program Argument	Description
-d, --list_default_target	List the default build target.
-c, --clean_target_prior	Clean TARGET build prior to building and programming the BRS-100-GW1NR9 board.

20. <https://github.com/BrisbaneSilicon/BRS-100-GW1NR9/blob/public/README.md#prerequisites>

Program Argument	Description
-b, --check_if_target_built	Print firmware built status of provided target board and exit.
-t, --custom_target_device CUSTOM_TARGET	Instead of the default target, target 'CUSTOM_TARGET'.



If you have previously loaded an FTDI driver in order to utilise the user comms, you will need to remove it prior to programming the board:

```
sudo rmmod ftdi_sio
```

4.2.2 Windows

Support for Windows is coming soon!

4.3 Board Demonstration Mode

4.3.1 Linux

This project can be built in 'board demonstration' mode to showcase the various hardware features (GPIO, UART, PSRAM - although as of 12/2025 not all features are demonstrated, see [roadmap](#)²¹). To run the board demonstration firmware, build the firmware with the '-b' command line argument (below) and then program the board as per [program](#)²².

```
./build.sh -b
```

Once the board has been programmed, load your FTDI driver and connect a terminal program to UART1, as per the below (I have used the terminal program [tio](#)²³ but any should work (Putty, Screen etc).

```
sudo modprobe ftdi_sio
sudo stty -F /dev/ttyUSB1 115200
sudo tio -b 115200 /dev/ttyUSB1 --output-delay 20
```

Next, reset the board by pressing pushbutton one (the pushbutton closest to Pin 1). The terminal should clear and then print out something similar to the following:

```
BOARD: BRS-100-GW1NR9
FW: c2ce822|2025-12-09 15:30:43
```

The 'FW' tag can be described as follows:

```
<GIT repo commit SHA> <-dirty> (only if it was built with local changes) | <build date> <build time>.
```

So I built commit SHA ID [c2ce822](#)²⁴, with no local changes, on the **12/09/2025** at approximately **3:30pm**. Note, there may be more printed, see [roadmap](#)²⁵.

In board demonstration mode, the state of GPIO<1..16> (the demonstration firmware configures these pins as inputs) will be reflected on GPIO<17..32> (the demonstration firmware configures these pins as outputs). For example, GPIO17 will reflect the state of GPIO1. To test this, simply connect GPIO1 alternatively to VCC or GND (both next to GPIO32) and monitor GPIO17 on your oscilloscope or multimeter.

21. <https://github.com/BrisbaneSilicon/BRS-100-GW1NR9?tab=readme-ov-file#roadmap>

22. <https://github.com/BrisbaneSilicon/BRS-100-GW1NR9?tab=readme-ov-file#program>

23. <https://github.com/tio/tio>

24. <https://github.com/BrisbaneSilicon/BRS-100-GW1NR9/commit/c2ce822baf2d6a4d22435ef2df5be94f321e9bb4>

25. <https://github.com/BrisbaneSilicon/BRS-100-GW1NR9/tree/public?tab=readme-ov-file#roadmap>

4.3.2 Windows

Support for Windows is coming soon!



If you notice that after programming the board all six user LEDs are slowly flashing, it means you have built the example project without the '-b' switch. This results in the 'user.sv²⁶' module (which by default only flashes all LEDs) being included in the firmware instead of the 'board_demonstration.sv' module.

26. <http://user.sv>

4.4 Extending

Extending the project with your custom firmware is quite straightforward, simply modify the [user.sv](#)²⁷ file (located in <BRS-100-GW1NR9 repository directory>/BRS-100-GW1NR9/proj/common/systemverilog/). You can also instantiate your own Systemverilog or VHDL modules, but ensure you add them to the appropriate build script file [synth.tcl](#)²⁸ ('scripts' directories).



More information will be added to this section in good time.

27. <https://github.com/BrisbaneSilicon/BRS-100-GW1NR9/blob/public/proj/common/systemverilog/user.sv>

28. <https://github.com/BrisbaneSilicon/BRS-100-GW1NR9/blob/public/proj/common/scripts/synth.tcl>