FRONTGRADE

FACT SHEET

UT32MOR500

32-Bit-Microcontroller

9/14/2021

Version #:1.0.0

Frequently Asked Questions

This document is intended to answer common questions about the UT32M0R500. For more in-depth information, please consult the Datasheet or the Functional Manual, both of which can be found here: https://frontgrade.com/product/ut32m0r500

Q. What is the processor core of the UT32M0R500?

A. The UT32M0R500 has a 32-bit RISC based ARM™ Cortex® M0+ processor.

- Compact Thumb-2 Instruction Set
- Multi-Level Interrupt support through the Nested Vector Interrupt Controller (NVIC) enables advanced real- time processing
- Memory Protect Unit helps prevent application code and data corruption improving reliability
- Wake Up Interrupt Controller interface enables event drive power up
- Debug Access Port

Q. How fast is the UT32M0R500?

A. The UT32M0R500 has an internal 50MHz oscillator and supports an external clock source and crystal oscillator.

Q. What peripherals are included?

A. The following peripherals are included:

- 2x CAN 2.0B Controllers
- 2x UART
- SPI
- 2x I2C
- JTAG
- 4x General purpose timers
- 3x PWM
- Watchdog Timer
- Real Time Clock
- 48x GPIO (21 dedicated)
- 8x Hardware Interrupts (shared with GPIO)
- 12-bit ADC 100 ksps with PGA (16 Single Ended or 8 Differential Channels)
- 1 mA Precision Current Source
- 2x 12-bit DACs
- 2x Analog Voltage Comparators
- Temperature Sensor

Q. What Firmware Development Environment is recommended?

A. Frontgrade officially supports the Keil® µVision® IDE for firmware development. Keil® offers a free version of the software that has a code size restriction of 32Kbytes, or paid versions that can be found at https://www2.keil.com/mdk5. Additionally, to perform debugging Frontgrade recommends the use of the Keil® ULINK2® debug adapter. Both the IDE and the debug adapter are sold by Keil®, and are not included with the UT32M0R500.

Q: Will one of the other Keil® ULINK® debuggers work?

A: Yes, any debugger on the following link will work, the IDE settings (see above) will need to be configured to match the chosen debugger: https://www2.keil.com/mdk5/ulink

Q: What is provided inside Frontgrade's Software Development Kit (SDK)?

A: Inside of the SDK is:

- The Standard Peripheral Library files and UT32M0R500 Specific ARM files required for firmware creation
- Example application files for various peripherals to jump-start firmware development
- Documentation:
    - An application note about creating projects in the Keil® µVision® IDE
    - The Quick Start Guide full of useful product information
    - The UT32M0R500 EVB Users' Guide provides information on how the evaluation board functions, and how to program/debug the UT32M0R500

Q: How much memory does the UT32M0R500 have?

A: The UT32M0R500 contains 96KB of embedded Dual Port SRAM with EDAC + Scrubbing, and 64Mb of internal Flash Memory. The flash memory allocates 384KB for user application firmware in four independent 96KB sections, plus an image override sector.

Q: What is the difference between programming and debugging the UT32M0R500?

A: The UT32M0R500 can have the firmware images (dedicated Flash Memory) programmed via the UART0 , CAN0, or JTAG interfaces. Once the application is loaded into a firmware image, it can be loaded into SRAM by changing the boot configuration pins and resetting the device. The application will remain in Flash Memory regardless of resets or loss of power, and can be erased or reprogrammed through the bootloader or JTAG.

The UT32M0R500 uses the Keil® ARM™ development tools to debug software. The development tools interface with the UT32M0R500's JTAG connections to load code directly into the device SRAM. The development tools then allow the user to free-run, stop, or step through application code using the JTAG connection. Because the application code is loaded into SRAM, resetting or loss of power to the UT32M0R500 will clear the SRAM.

Application Notes for programming via UART0 (App-Note-UT32M0R500-UART-Flash-Download.pdf), via CAN0 (App- Note-UT32M0R500-CAN-Update-Protocol.pdf), and via JTAG (App-Note-UT32M0R500-and-Keil-Flash-Download.zip) can be found on the product page.

Q: What is the recommended pullup/pulldown resistor configuration for the JTAG pins?

A: When being used for debugging, 10KΩ pullup resistors should be placed upon the TRST, TMS, TCK, and TDI signals, with TDO floating.

When JTAG is NOT being used (Ex. During Flight) TRST MUST have a 10KΩ pulldown resistor, TMS, TCK, and TDI should stay pulled up with 10KΩ resistors, and TDO should stay floating.

Q: What operating systems run on the UT32M0R500?

A: Both RTX and FreeRTOS runs on the UT32M0R500.

Q: What is the state of the UT32M0R500's I/O's after a reset?

A: Specific I/O's become unknown during a Power on Reset (POR), and then change to a known value after the POR. For a complete list, please consult the UT32M0R500 datasheet pinlist.

Q: What is the weight of the UT32M0R500?

A: The Column Grid Array (CGA) package option weighs 4.116 grams. The Land Grid Array (LGA) package option weighs grams.

Q: Can the sectors of the NOR Flash Memory that are not allocated for user application firmware be used for other things?

A: Yes, users can use sectors 0-7 and 17-141 to store data. Note that the NOR Flash must remain powered off 90% of the time in order to meet radiation requirements.

Q: Why are some Reserved pins marked "NC" and others "NUIL"?

A: Pins listed as RESERVED in the datasheet have one of two descriptions.

- NUIL, or "Not Used Input Low" are reserved input pins, and MUST be tied to VSS through a >10kΩ±10% resistor
- NC, or "No Connect" are reserved output pins, and MUST be left floating