UART Commands for xWRL68xx

Contents

1 UART Protocol

The UART protocol involves a simple command-response flow. Host sends the command packet to XWR device and waits for the response from the device. Host can't initiate another command before receiving the response of the previous command. Each command packet is immediately processed by the bootloader without any scheduling or processing in the background, and a valid response packet is sent. Each command has a known response packet associated with it and host application is responsible for validating the same.

2 Packet Format

2.1 Command packet format

The command packet consists of a sync pattern(0xAA) of 1 byte, 2 bytes to specify length of the packet, 1 byte of checksum for payload bytes and payload of max 252 remaining bytes.

Note - The length bytes are transmitted in big endian format with MSB byte transmitted followed by the LSB byte.

2.2 Response packet format

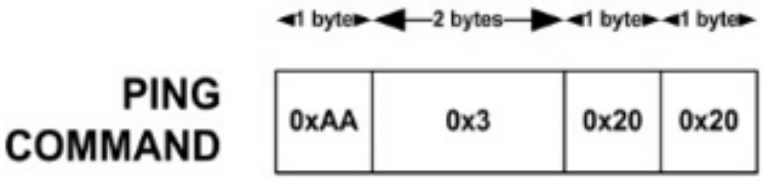
The response packet is same as the command packet except for that there is no sync byte for the response packet.

3 Supported Commands

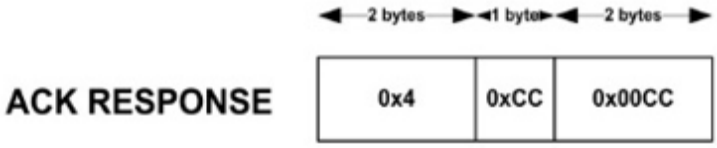
3.1 Ping command

The ping command can be issued by the host for verifying the UART connection with XWR device.

Command Packet -



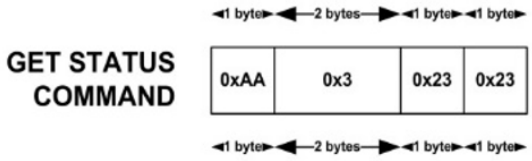
Response Packet – Ack response packet



3.2 Get status command (execution status of last command)

The get status command can be issued by the host for getting the error code associated with the execution of previous command. This should be issued only after receiving the response of the previous command. For example, after receiving NACK response for a command, this command can be issued to determine the error information which resulted in the NACK response for the latest command.

Command Packet -



Response Packet 1 -

ACK – If the command was successfully received

NACK – If the command checksum verification failed.

Response Packet 2 –

If response 1 is ACK, the below response follows:

LENGTH (2 bytes)		CHKSUM (1 byte)	PAYLOAD (8 bytes)
0x00 (MSB)	0x0A (LSB)		Error status = 0 No error in execution of previous command ≠ 0 Error code set by execution of previous command

3.3 Get version command

This command can be issued to get version information for the ROM bootloader.

Command Packet -

Response Packet 1 -

ACK – If the command was successfully received

NACK – If the command checksum verification failed.

Response Packet 2 –

If response 1 is ACK, the below response follows:

LENGTH (2 bytes)		CHKSUM (1 byte)	PAYLOAD (12 bytes)	
0x00 (MSB)	0x0E (LSB)		ROM Version information (4 bytes) byte [0] = Build version byte [1] = Minor version byte [2] = Major version byte [3] = ROM Gen version	Reserved (8 bytes)

3.4 Open download command

This command is issued by the host to open a file download context for either serial flash download or SRAM download.

Command Packet –

SYNC (1 byte)	LENGTH (2 bytes)		CHKSUM (1 byte)	PAYLOAD (17 bytes)	
0xAA	0x00 (MSB)	0x13 (LSB)		Opcode (1 byte)	0x21
				FILE_SIZE	Size of image in bytes

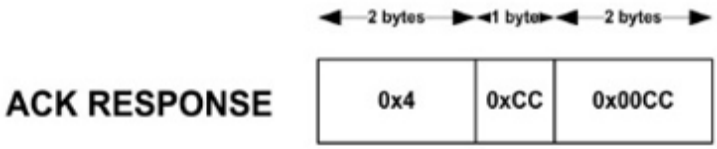
				(4 bytes)	The size should not be a multiple of 240 bytes to ensure the last chunk is not a multiple of 240.
				STORAGE_TYPE (4 bytes)	0x2 = Serial flash 0x4 = SRAM Others = Reserved
				FLASH_ADDRESS (4 bytes)	Note – This field is read only if STORAGE_TYPE is serial flash, the address has to be 4K flash sector aligned. Address in serial flash where data needs to be written.
				RESERVED (4 bytes)	0x0

Note –

1. FILE_SIZE, STORAGE_TYPE and FILE_TYPE should be transmitted in big endian fashion similar to the length bytes of the protocol packets.
2. FILE_SIZE should be multiple of 8 bytes

Response packet –

If there is no error in specified configuration ACK response is sent.



Else NACK response is sent –

LENGTH (2 bytes)		CHKSUM (1 byte)	PAYLOAD (2 bytes)
0x00 (MSB)	0x4 (LSB)	0x33	0x0033

3.5 Send SFLASH download chunk command

This command is used to send the image chunks, 1…..n, after the “open download command” is issued.

Command packet -

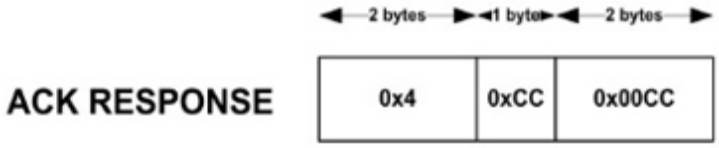
SYNC (1 byte)	LENGTH (2 bytes)	CHKSUM (1 byte)	PAYLOAD Chunks 1…n-1: 240 + 1(for opcode) bytes

				Chunk n: remaining bytes < 240 + 1(for opcode) Chunk n should be less than 240 bytes	
0xAA	0x00 (MSB)	Chunks 1…n-1: 243 bytes (LSB) Last chunk n: Remaining image bytes + 0x1 (for opcode) + 0x2 (for length bytes of protocol packet)		0x24 (opcode)	Image data

Note: Chunk n should be less than 240 bytes, TI image generation tool(metaimage_creator.exe) internally pads additional bytes if needed to respect this constraint. Customers would also have to respect this constraint while generating binaries without using TI tools.

Response packet –

If there is no error in specified configuration ACK response is sent.



Else NACK response is sent –

LENGTH (2 bytes)		CHKSUM (1 byte)	PAYLOAD (2 bytes)
0x00 (MSB)	0x4 (LSB)	0x33	0x0033

3.6 Send SRAM download chunk command

This command is used to send the image chunks, 1…..n, after the “open download command” is issued.

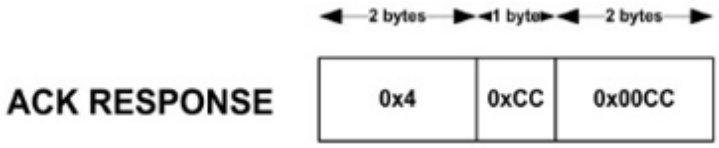
Command packet -

SYNC (1 byte)	LENGTH (2 bytes)		CHKSUM (1 byte)	PAYLOAD Chunks 1…n-1: 240 + 1(for opcode) bytes Chunk n: remaining bytes < 240 + 1(for opcode) Chunk n should be less than 240 bytes	
0xAA	0x00 (MSB)	Chunks 1…n-1: 243 bytes (LSB) Last chunk n: Remaining image bytes + 0x1 (for opcode) + 0x2 (for length bytes of protocol packet)		0x26 (opcode)	Image data

Note: Chunk n should be less than 240 bytes, TI image generation tool(metaimage_creator.exe) internally pads additional bytes if needed to respect this constraint. Customers would also have to respect this constraint while generating binaries without using TI tools.

Response packet –

If there is no error in specified configuration ACK response is sent.



Else NACK response is sent –

LENGTH (2 bytes)		CHKSUM (1 byte)	PAYLOAD (2 bytes)
0x00 (MSB)	0x4 (LSB)	0x33	0x0033

3.7 Close download command (SFLASH)

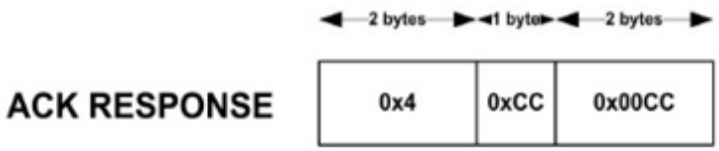
This command is used to close the download context for SFLASH.

Command packet -

SYNC (1 byte)	LENGTH (2 bytes)		CHKSUM (1 byte)	PAYLOAD (5 bytes)	
0xAA	0x00 (MSB)	0x7 (LSB)		Opcode (1 byte)	0x22
				STORAGE_TYPE (4 bytes) (big endian)	0x2 = Serial flash

Response packet –

If there is no error in specified configuration ACK response is sent.



Else NACK response is sent –

LENGTH (2 bytes)		CHKSUM (1 byte)	PAYLOAD (2 bytes)
0x00 (MSB)	0x4 (LSB)	0x33	0x0033

3.8 Close download command (SRAM)

This command is used to close the download context for SRAM.

Command packet -

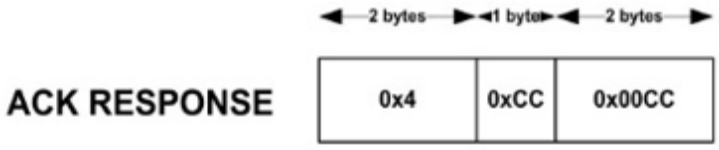
SYNC (1 byte)	LENGTH (2 bytes)		CHKSUM (1 byte)	PAYLOAD (5 bytes)	
0xAA	0x00 (MSB)	0x7 (LSB)		Opcode (1 byte)	0x22
				STORAGE_TYPE (4 bytes) (big endian)	0x4 = SRAM

Response packet –

There are two response packets associated with this command.

Response packet #1 -

If there is no error in specified configuration ACK response is sent.



Else NACK response is sent –

LENGTH (2 bytes)	CHKSUM (1 byte)	PAYLOAD (2 bytes)

0x00 (MSB)	0x4 (LSB)	0x33	0x0033

3.9 Change Baud (Deprecated)

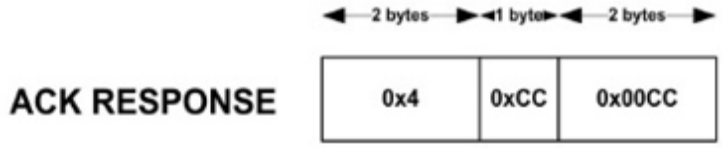
This command is used to change the UART communication baud rate. The radar device continues to operate in the existing baud rate till the response for this command is completely sent. Subsequently, the device switches the newly requested baud rate. The host shall wait for 5 ms after receiving the response before sending a new command to the radar device.

Command packet -

SYNC (1 byte)	LENGTH (2 bytes)		CHKSUM (1 byte)	PAYLOAD (5 bytes)	
0xAA	0x00 (MSB)	(LSB)		Opcode (1 byte)	0x27
				BAUD_RATE (4 bytes) (big-endian)	9600 19200 57600 115200 (default) 460800 576000 921600

Response packet –

If there is no error in specified configuration ACK response is sent.



Else NACK response is sent –

LENGTH (2 bytes)	CHKSUM (1 byte)	PAYLOAD (2 bytes)

0x00 (MSB)	0x4 (LSB)	0x33	0x0033

4 Error codes

Error Code Bits Definition (8 bytes)	Error Description
0x0000000000000001	M_BOOT_UART_DWNLD_INCOMPLETE Occurs when close download command while the SFLASH/SRAM image is only partially transferred.
0x0000000000000002	M_BOOT_UART_INVALID_TARGET_MEM Occurs when 1) SRAM storage type command used in SOP Device management 2) SFLASH storage type command used in SOP Functional mode
0x0000000000000004	M_BOOT_UART_INVALID_FLASH_ADDR The FLASH_ADDRESS specified in in open download command is not aligned to a sector boundary.
0x0000000000000008	M_BOOT_UART_CRC_INIT_FAILURE CRC module initialization failed
0x0000000000000010	M_BOOT_UART_INVLD_SIZE The frame sizes don’t match the expected UART frame size.
0x0000000000000020	M_BOOT_UART_CRC_MISMATCH UART SFLASH download CRC mismatch – Occurs when calculated CRC not match with the expected CRC in flashing
0x0000000000000040	M_BOOT_UART_PARSE_FAIL Parsing of the image failed. Possible return for close download in functional mode.

5 Sequence for Download to Flash

The flash programming is as depicted in Figure 4-2.

1. Step 1: Issue a UART break signal and wait for reception of acknoledgement
2. Step 2: The default timeout for bootloader is ~21 seconds after which it stops processing the UART commands. However, the flashing time of the images depends on the image size and the host/usb-uart converter latencies and might exceed the ~21 seconds. This step issues a sequence of commands to increase the timeout of the bootloader to ~300 seconds. The following commands have to be issued in sequence. This step is implemented in the TI provide ar_programmer utility.
    a. Open download command – Use the following parameters: FILE_SIZE = 8, STORAGE_TYPE = 0x2, FLASH_ADDRESS = 0xE6040000. The flash address translates to a SoC register to control the CPU clock dividers.
    b. Send SFLASH download chunk command – Use payload as [0x15, 0x02, 0x80, 0x61,0x30, 0x33, 0xFF, 0x0F]. This translates to a timeout of ~300 seconds.
    c. Close download command (SFLASH) – No parameters present. The expected response for this command in this scenario is a NACK and can be ignored.

Note: Step 2 will lead to erasure of sector 64 (i.e. Flash address 0x40000)

The following image download order is recommended for first time flashing to ensure the sector 64 is re-written. This recommendation is implemented in the TI provided ar_programmer utility.

    a. Download the flash headers to sector 1 & sector 2
    b. Download the metaimage 1 at sector 3, address offset 0x2000.
    c. Backup metaimages shall be downloaded in back to back sectors after the primary metaimage(i.e. metaimage 1).

3. Step 3: Write the image data to flash
    a. Issue an open download command by specifying the flash address and the length of data that needs to be written to the serial flash.
    b. Divide the data to be written to the flash into multiple chunks. Every chunk except the last chunk should contain 240 bytes of data (refer “Send SFLASH download chunk command”).
    c. Once all the chunks are sent, the device needs to issue a Close download command (SFLASH) and wait for ACK.

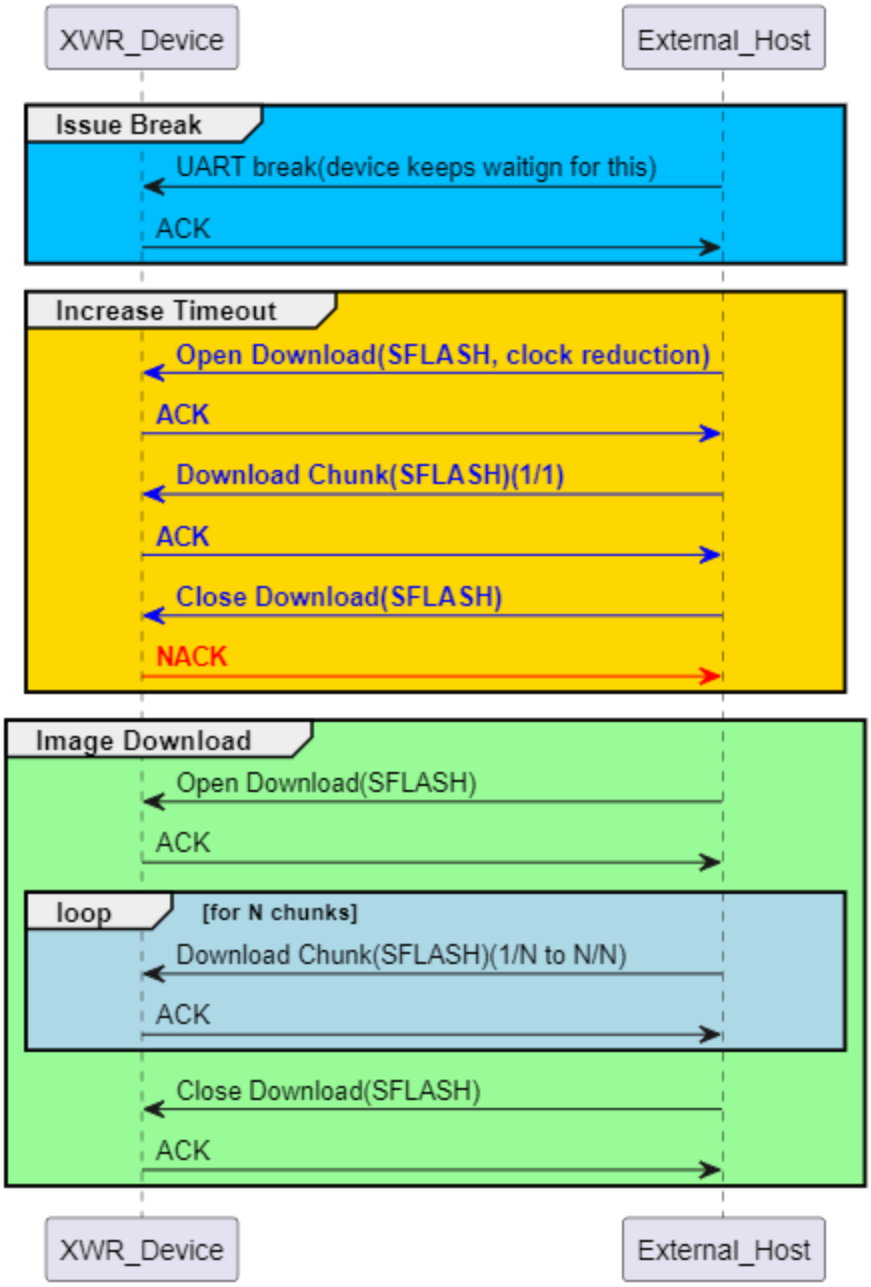


Figure 5-1 – Flashing/Booting Sequence

6 Sequence for Booting

The booting sequence using UART is as depicted in Figure 4-3. The initial handshake starts with a UART break issued by the external host.

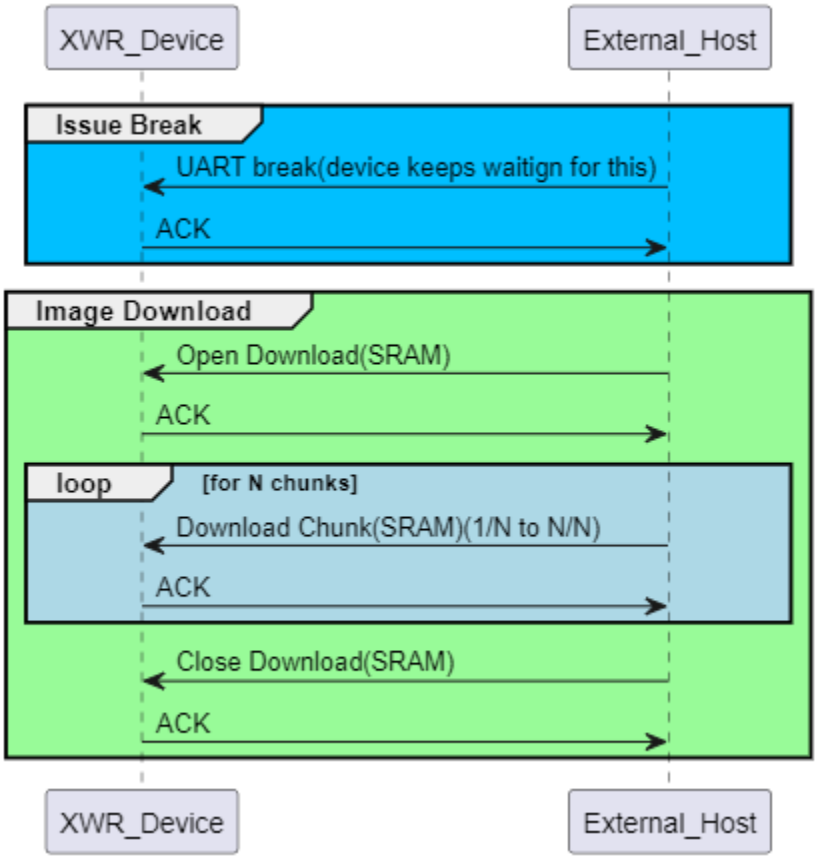


Figure 6-3 – Booting Sequence

7 Protocol careabouts

1. For UARTA SRAM download, the device timeout has to be increased from ~20 seconds to ~300 seconds. The timeout change related information is communicated to RBL via timeout_update.bin file, the timeout_revert.bin file communicates to the RBL to revert back the timeout. Hence, the first image in the meta image needs to be timeout_update.bin and the last image in the meta image needs to be timeout_revert.bin. Please refer sample metaimage_uart_gp.json file.
2. Boot interface detection timeout is 10 sec in GP device and 800msec in HSSE devices.
    a. The host must initiate UART communication within 10 sec after the NRESET in GP and within 800msec in HSSE

3. The minimum core image size shall be >240 bytes for UART based boot
4. In UARTB flashing, the RBL timer begins counting down as soon as the power is turned on and lasts for 20 seconds. To maximize this timeout, we need to reduce the timer's tick rate as quickly as possible after the power-up. Therefore, it's essential to initiate the flashing process immediately in order to extend the timeout period.
5. Specific to uniflash:
    a. The address offsets for the images are fixed [0x2000, 0xfc000, 0x1f6000, 0x2f0000], 1 MB is reserved for each metaimage
    b. In case the metaimage size exceeds 1 MB, use offsets for metaimage 1 & metaimage3(i.e. 0x2000 & 0x1f6000) to allow 2 images up to 1.5 MB
    c. Nreset has to be asserted to program each backup metaimage similar to xWRL6432
    d. In case of flashing multiple metaimages, flash the metaimage 1(i.e. offset 0x2000) at the last
    e. For example, if you want to program 4 images, program image 2, image 3 , image4 followed by image 1