Features:

- Small size (1, 2 or 8 Kbyte depending on the device)
- Uses UART RX and TX pins, and optional Bootloader Entry pin
- Allows updating itself
- Optional image verification using CRC32

Hardware configuration

UART pins used by the bootloader depending on the device type are listed in the table below.

Device	UART TX	UART RX	Entry	Size, bytes
SAM C21	PA22	PA23	PA19	1024
SAM D20	PA22	PA23	PA19	1024
SAM D21	PA8	PA9	PA19	1024
SAM D10	PA10	PA11	PA25	1024
SAM L10	PA22	PA23	PA19	1024
SAM L11	PA22	PA23	PA19	1024
SAM G55	PA10	PA9	PA11	2048
SAM E54	PA4	PA5	PA27	8192

Bootloader Entry pin has an active low level. Its value is sampled at the beginning of the bootloader execution. Although internal pull-up resistor is enabled before sampling the pin, it is recommended that Bootloader Entry pin is pulled-up externally for improved noise immunity.

UART settings used by this bootloader are 115200 8N1.

This bootloader supports UART baudrate auto tuning. This feature may be useful for hosts that can’t set exact value of the baudrate. Auto tuning can be initiated at any point of operation. To perform auto tuning, host must send a break signal (at least 11 bits of low level) followed by a character 0x55. No response is expected for this request.

Note that baudrate auto tuning is not supported on SAM D20 and SAM G55.

Method of entry

Bootloader can be invoked in a number of ways:

1. Bootloader will run automatically if there is no valid firmware. Firmware is considered valid if the first word is not 0xFFFFFFFF. Normally this word contains initial stack pointer value, so it will never be 0xFFFFFFFF unless device is erased.
2. Bootloader will run on external request if the value of the Bootloader Entry pin is low on bootloader execution start.
3. Bootloader will run on application (internal) request if the first 4 locations in the SRAM are equal to 0x78656C41.

External reset takes priority over any other method of entry.

The following code can be used by the application to request the bootloader execution:

```
{
  uint32_t *ram = (uint32_t *)HSRAM_ADDR;
   __disable_irq();
  ram[0] = 0x78656c41;
  ram[1] = 0x78656c41;
  ram[2] = 0x78656c41;
  ram[3] = 0x78656c41;
  NVIC_SystemReset();
}
```

Bootloader commands

All bootloader commands have the same general format shown in the table below.

Command ID	Guard Value	Data 0	…	Data N
1 byte	4 bytes	4 bytes	…	4 bytes

The number and meaning of the data words varies with the command. All data words must be sent in a little-endian (LSB first) format.

Guard Value must be a constant value of 0x78656C41. This value provides additional protection against spurious commands.

All bytes of the command frame must be sent within 100 ms of each other. After 100 ms of idle time, incomplete command is discarded and bootloader goes back to waiting for a new Command ID. This behavior allows host to re-synchronize in the case of synchronization loss.

Bootloader understands the following commands:

1. Unlock (0xA0)
2. Data (0xA1)
3. Verify (0xA2)
4. Reset (0xA3)

Unlock command must be issued before the first Data command. It has the following payload:

- Data 0 – Starting Offset
- Data 1 – Image Size

Starting Offset is the offset from the beginning of the flash memory. To upgrade the bootloader itself this value must be set to 0. Application image offset is device-dependent and valid values are listed in the table below. Image offset must be aligned at an Erase Unit Size boundary, which is also device dependent. Image size must be in increments of Program Unit bytes.

Device	**Application Offset, bytes**	**Program Unit Size, bytes**	**Erase Unit Size, bytes**
SAM G55	2048	512	2048 if address is below 16384, 8192 otherwise
SAM E54	8192	512	8192
All others	1024	256	256

Data command is used to send image data. It has the following payload:

- Data 0 – Starting Offset
- Data 1 – Data N – Image Data (Program Unit Size bytes)

Starting offset must be located inside the region previously unlocked via the Unlock command. Attempts to request the write outside of the unlocked region will result in error and supplied data will be discarded.

This bootloader supports simultaneous Flash memory write and reception of the next block of data. The next block of data may be transmitted as soon as the status code is returned for the first one.

Because of this behavior, the status code for the last block will be sent before this block is written into the Flash memory. To ensure that this block is written, host must send another command and wait for the response. So either Verify or Reset command must be sent after the last block of data. In case if Verify command is used, but actual verification is not required, fields of the Verify command must be set to include at least one block of Program Unit Size bytes, and Image CRC can be set to 0. In that case CRC Fail status will be reported, which may be safely ignored.

Verify command is used to verify the image data. It has the following payload:

- Data 0 – Expected CRC32

Image CRC is a standard IEEE CRC32 with a polynomial of 0xEDB88320.

Internal CRC is calculated based on the values actually read from the Flash memory after programming, so it verifies the whole chain. Image CRC is calculated over the previously unlocked region.

Reset command is used to exit the bootloader and run the application. It is necessary if the host has no control over the reset pin. But it can also be useful even if host has control over the reset, since this command allows host to communicate up to 4 words of arbitrary information to the application. It has the following payload:

- Data 0 – Arbitrary Value 0
- Data 1 – Arbitrary Value 1
- Data 2 – Arbitrary Value 2
- Data 3 – Arbitrary Value 3

Supplied arbitrary values are passed to the application in the first 4 locations in the SRAM.

Supplied values must not be all equal to 0x78656C41, as this will request the bootloader execution.

Response Codes

Bootloader will send a single character response code in response to each command. Sequential commands can only be send after the response code is received for a previous command, or after 100 ms timeout without a response.

Possible response codes are:

1. OK (0x50) – command was received and processed successfully
2. Error (0x51) – there were errors during the processing of the command
3. Invalid (0x52) – invalid command is received
4. CRC OK (0x53) – CRC verification was successful
5. CRC Fail (0x54) – CRC verification failed

Programming algorithm

After issuing each command, host must wait for the response code for at least 100 ms. If no response code is received during this time, the command may be considered lost and can be repeated again.

Host controller must perform the following actions in order to update the firmware:

1. Request the bootloader entry.
2. Wait for at least 5 ms for bootloader to start.
3. Issue the Unlock command with required image parameters.
4. Send the Data command with the firmware data.
5. Repeat the item 4 until the entire image is transferred.
6. Issue the Verify command and check the response code.
7. If valid CRC was supplied in the Verify command, but response code is not CRC OK, then repeat the update starting from the item 3.
8. Issue the Rest command.