IQS5xx I2C Bootloader v2.x
Technical User Guide

Table of Contents

1 Introduction

The goal of this document is to describe the working of the Azoteq IQS5xx I2C Bootloader v2.x from a user point of view. The user in this case will be the engineer that needs to develop the firmware upgrade routines to be used on the master MCU to update the IQS5xx.

2 Overview

The IQS5xx can run in one of two modes:

1. **Bootloader mode:**

   The IQS5xx provides the user with an interface containing numerous USB address commands that can be used to upgrade the application firmware.

2. **Application mode:**

   The IQS5xx executes the application firmware.

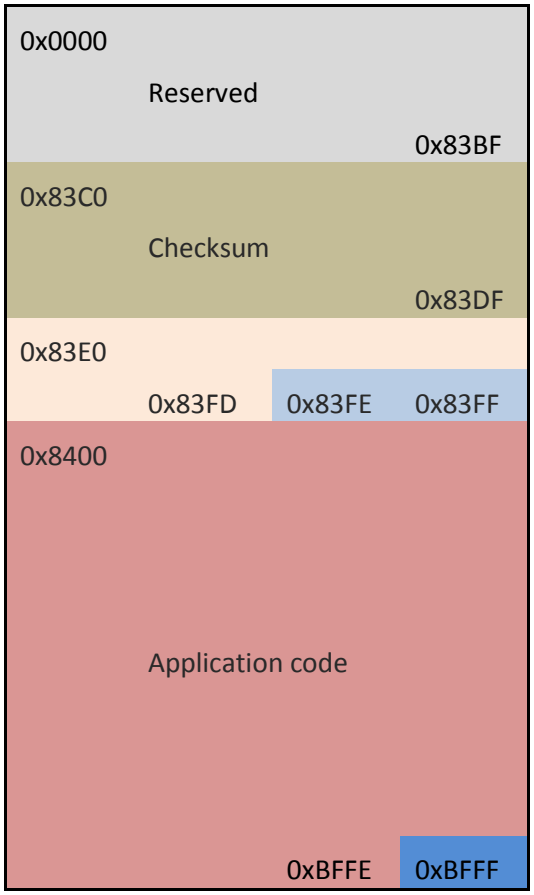
Note that the address commands associated with the bootloader mode are not available when the device runs in the application mode, and vice versa.

The next sections concern the following:

- Simplified IQS5xx memory map.
- Bootloader entry.
- I²C communication with the bootloader.
- CRC calculation.

3 Simplified IQS5xx memory map

Here follows a graphical representation and a description of the IQS5xx memory map:



1. 0x0000-0x83BFF should be considered reserved space and not written.
2. 0x83C0-0x83DF is the firmware checksum.
3. 0x83E0-0x83FF should be written as 0x00.
   Note that if the user writes 0x00 to 0x83FE and 0x83FF, and then reads the data at those addresses again, the addresses will not contain 0x00.
4. 0x8400-0xBFFE contains the application code.
5. 0xBFFF contains the configurable I^2C slave address of the IQS5xx.

When performing a firmware upgrade the master MCU must write data to addresses 0x83C0 through 0xBFFF (inclusive). Unused memory locations must be written as 0x00.

4 Bootloader entry

This section describes the two available methods that can be used to enter bootloader mode. Once in bootloader mode, the master must write the application firmware to the IQS5xx and verify that the upgrade was successful using the available I²C interface.

4.1 Polling after Power-on reset:

Shortly after device reset an I²C polling window opens. If the IQS5xx is polled (continuously addressed by the master until an ACK is received), using the bootloader I²C address (0x40 xor (IQS5xx I2C slave address)), the device enters bootloader mode. Application firmware can then be loaded. The polling window stays open for approximately 2ms. If the polling window expires, and the device has not been polled on the bootloader I²C address, the device will enter the application mode and start executing the application firmware. The following flow chart describes bootloader entry using the polling method:

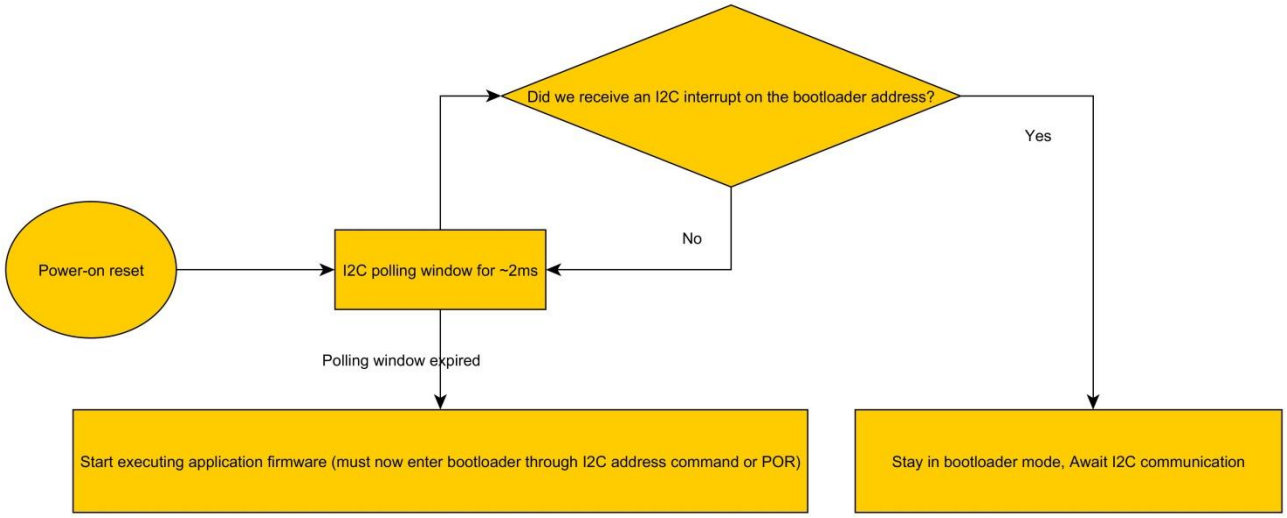


Figure 4.1 Basic working of the bootloader

4.2 Enter bootloader through application

The IQS5xx-A000 has the ability in the firmware to exit the application mode and enter the bootloader mode by writing 0xA5 to address 0xFF. The device will then reset and enter bootloader mode. Note that this command is only available if the application code section of the memory map contains a valid firmware image and the device is running in the application mode. The sequence is therefore:

- I²C start.
- Address the IQS5xx in application mode (default 0x74) and indicate that data will be written to the IQS5xx (0x74<<1).
- Write 0xFF and then 0xA5.

Note that this command will not be available if there was an error during a previous firmware update and the application section of the memory no longer contains a valid IQS5xx firmware image, or if the firmware image does not include this functionality. The recommended entry method is thus to poll and enter after power-on reset.

5 I²C Communication with the bootloader

By default the IQS5xx has an I²C address of 0x74, the bootloader address is the application address XOR 0x40, hence by default 0x34 (0x74 ^ 0x40). The following commands are sent using I²C, with "slave address" the bootloader address (default 0x34).

The following commands are supported by the bootloader:

I2C Command	Description
0x00	Read Bootloader version (Read only, 2 data bytes)
0x01	Read 64 bytes from memory map. (see description below)
0x02	Execute application firmware (Write only, 0 data bytes)
0x03	Do CRC check on current firmware. (Read one byte, 1=fail, 0=success)
Other	Write data bytes to firmware application address if valid address presented

Note that these commands are only available in bootloader mode.

5.1 Read Bootloader version(0x00):

A single command byte (0x00) is written to the slave, the first two bytes read from the device will return the bootloader version. Detailed I²C will look like this:

Start, (slave address << 1), master writes 0x00, stop.

Start, ((slave address << 1) | 0x01), master reads two bytes, stop.

It is recommended to read the bootloader version to ensure that bootloader entry was successful.

5.2 Read block from memory map (0x01):

The read command is intended for the master to verify that the firmware download has been successful. The CRC check may be trusted to verify the application program space, but note that certain blocks such as the interrupt vector, checksum descriptor and EEPROM data space is not included in the CRC check. (Refer to the section below on CRC calculation.)

By writing 0x01 to the slave, followed by a two byte starting address, the slave will return 64 consecutive bytes starting at the specified address.

It is recommended to verify the entire memory map written to the slave with this command.

Detailed I²C:

Start, (slave address << 1), master writes 0x01, start address MSB, start address LSB, stop.

Start, ((slave address << 1) | 0x01), master reads up to 64 bytes from slave, stop.

5.3 Execute application firmware (0x02):

The master writes a single command byte (0x02) to the slave, at which point the slave will jump to the application start address. I²C details:

Start, (slave address << 1), master writes 0x02, stop.

5.4 Calculate CRC of current application firmware (0x03):

The master writes a single command byte (0x03) to the slave, one byte is read from the slave, which will be the CRC check pass or fail. A 1 indicates failure, 0 indicates success.

Start, (slave address << 1), master writes 0x03, stop

Start, ((slave address << 1) | 0x01), master reads single byte result, stop.

5.5 Program application firmware to device:

The device will program a block of application firmware once the entire block has been received via I^2C and the starting address of the block is valid.

Block size: 64 bytes.

Valid starting addresses are in multiples of 64 byte. (Hence the start address has to end with 6 zero bits, e.g. 0x83C0, 0x8400, 0x8440, 0x8480 etc.)

I^2C details:

Start, (Slave address << 1), (starting address MSB), (starting address LSB), data byte 1, data byte 2, … , data byte 64, stop.

NOTE: All 64 bytes MUST be sent in order to trigger the write on the bootloader.

Writes are allowed on blocks from 0x83C0 to 0xBFFF. The firmware is located from address 0x8400 to 0xBFFF and it is recommended to write 0x00 to every unused location, to insure that previous data is overwritten.

The firmware will be provided in standard intel hex format.

6 CRC Checksum calculation

The bootloader can be set to calculate a checksum on the application firmware present to validate the firmware. It is recommended to verify the firmware upgrade by reading all addresses from the slave with the read block command, but it is of equal importance to ensure that the bootloader passes the CRC check before finishing the firmware upgrade.

6.1 CRC Checksum formula

The checksum is calculated on the addresses used by the program firmware. The detail of the checksum is saved in a checksum descriptor located at address 0x83C0 in the memory map.

The checksum has the following format:

- 0x83C0: 0x04
- 0x83C1-0x83C2: 16-bit address for the start address for CRC calculation.
- 0x83C3-0x83C4: 16-bit address for the (end address + 1) for CRC calculation.
- 0x83C5: 0x00
- 0x83C6-0x83C7: 16-bit result of the checksum calculated on this space.

The checksum is automatically calculated on the IQS5xx and therefore the user does not need to do any CRC calculating. It is only included here for clarity.

7 Summary of firmware upgrade procedure

Suggested order of events:

1. Get device in bootloader mode (either by polling after power on, or application enter bootloader command)
2. Read the bootloader version number to verify bootloader entry success.
3. Write the new application firmware to the device (write the entire space 0x8400 – 0xBFFF, unused blocks may be omitted, but it is recommended to fill unused space with 0x00.)
4. Verify that the write was successful by reading all the data with the read block command.
5. Once the write has been verified program the checksum descriptor, which is located in the memory map block 0x83C0-0x83FF.
6. Execute the bootloader check CRC command and verify that a zero is returned to indicate success.
7. Exit bootloader mode.

	USA	Asia	South Africa
Physical Address	6507 Jester Blvd Bldg 5, suite 510G Austin TX 78750 USA	Rm1725, Glittery City Shennan Rd Futian District Shenzhen, 518033 China	109 Main Street Paarl 7646 South Africa
Postal Address	6507 Jester Blvd Bldg 5, suite 510G Austin TX 78750 USA	Rm1725, Glittery City Shennan Rd Futian District Shenzhen, 518033 China	PO Box 3534 Paarl 7620 South Africa
Tel	+1 512 538 1995	+86 755 83035294 ext 808	+27 21 863 0033
Fax	+1 512 672 8442		+27 21 863 1512
Email	kobusm@azoteq.com	linayu@azoteq.com.cn	info@azoteq.com

Please visit www.azoteq.com for a list of distributors and worldwide representation.