Application Note

AS72xx – EEPROM Updating

EEPROM Updating using I2C

Content Guide

1 General Description

This Application Note describes using the AS72xx I^{2}C slave interface for updating the external program EEPROM. This app note applies to I^{2}C slave capable AS72xx devices: AS7225, AS7261, AS7262 and AS7263. See Appendix A for first time EEPROM programming.

Note, the application note only applies to Scotty Sensor Systems with the firmware versions v3.x and 4.0.xx. Please use serial flash's for later firmware version. For further details refer to our application note "External Flash program and update".

2 I^{2}C EEPROM Update Methodology

The I2C update method uses the I2C hardware interface of the I2C slave capable AS72xx devices. Refer to the AS72xx datasheets for more information on the protocol for accessing the set of virtual I2C registers using the set of three actual, physical I2C registers.

The technique is described below, I^{2}C registers are detailed in the tables that follow.

Get the latest "*.bin" file from ams for the specific AS72xx device. The EEPROM device has two 56KB firmware partitions, one where the current firmware is stored and executed out of, and a second space where the new firmware is downloaded. These images can be switched via software by selecting the FW_IMAGE bit.

To start the download of a new image set the START_XFR bit in the FW_UPDATE register to 1. At this time the Firmware byte count register FWBC_LOW and FWBC_HIGH registers are set to zero. The user then sends the 56Kbyte image to the device one byte at a time by repeated sequential writes to FWLOAD register. Since the device does not support a burst mode on the I^{2}C port, it is required to repeatedly address the FWLOAD register and write the next sequential byte of the image to it. The user can query the Firmware Byte count registers FWBC_LOW and FWBC_HIGH at any time to know what the current byte count of the image received. Reading the FWLOAD register will give the last byte written in. The FWBC counter will read 1 when the first byte is written into FWLOAD register (it is initialized to zero at the start if the transfer, when START_XFR is set to 1) and auto increment to a max of 0xE002 (57,346 bytes) when the last byte of the 56Kbyte + 2 Byte Checksum transfer is written. If a user attempts to send more than (56Kbytes + 2 Byte Checksum) it is ignored. Once 56Kbytes + 2 Byte Checksum of data are transferred into the device the START_XFR bit is cleared to zero and the XFR_56K bit will be set to 1. If at any time during the data transfer if the link has hung up or the user wishes to stop the transfer and restart the process from the beginning, the KILL_XFR bit can be set to 1 to do this.

To validate the data transfer, a checksum needs to be verified. If the received checksum for the image is matched against the internally generated checksum then the CKS_OK flag is set to 1.

When both XFR_56K and CKS_OK are 1, the user can then update the device to the new firmware by setting the FW_PART to 1. When this occurs the device will go through a software reset switching to the newly loaded partition. No persistent data set by the user is changed with this firmware update. If the user wants to revert to the older image, simply set FW_PART again to toggle the active partition.

I^{2}C Registers

Figure 1: I^{2}C EEPROM Update Register Summary

Register Name	Address	R/W	Register Function	Reset Value
FW_UPDATE	0x48 = R 0xC8 = W	R/W	Control for firmware updates	
FWBC_LOW	0x49 = R 0xC9 = W	R/W	Firmware byte count (low byte)	
FWBC_HIGH	0x4A = R 0xCA = W	R/W	Firmware byte count (high byte)	
FWLOAD	0x4B = R 0xCB = W	R/W	Firmware download register	

Figure 2: I²C EEPROM Update Register Details (FW_UPDATE): Read/Write

7	6	5	4	3	2	1	0
Start_ XFR	KILL_ XFR	XFR_ 56K	CKS_ OK	FW_ PART	Reserved	Reserved	Reserved

Fields	Bits	Description
START_XFR	7	1 = Start new firmware transfer, clears bits 6:4
KILL_XFR	6	1 = Kill this transfer, 0 = No OP
XFR_56K	4	1 = Transfer successful (56kBytes transferred, no checksum validated) 0 = problem with transfer
CKS OK	5	Firmware checksum OK 1 = Yes, 0 = No
FW_PART	3	Selection of Firmware Partition 1 = Select new firmware partition 0 = Select previous firmware Partition
Reserved	2	Reserved
Reserved	1	Reserved
Reserved	0	Reserved

Figure 3: Other I²C EEPROM Update Register Details: Read/Write

Register	Address	Bits	Description
FWBC_LOW	0x49 = R 0xC9 = W	7:0	Firmware byte count Low Byte Number of bytes downloading

FWBC_HIGH	0x4A = R 0xCA = W	7:0	Firmware byte count Low Byte Number of bytes downloading
FWLOAD	0x4B = R 0xCB = W	7:0	Firmware download register – write firmware bytes to this address when START_XFR = 1

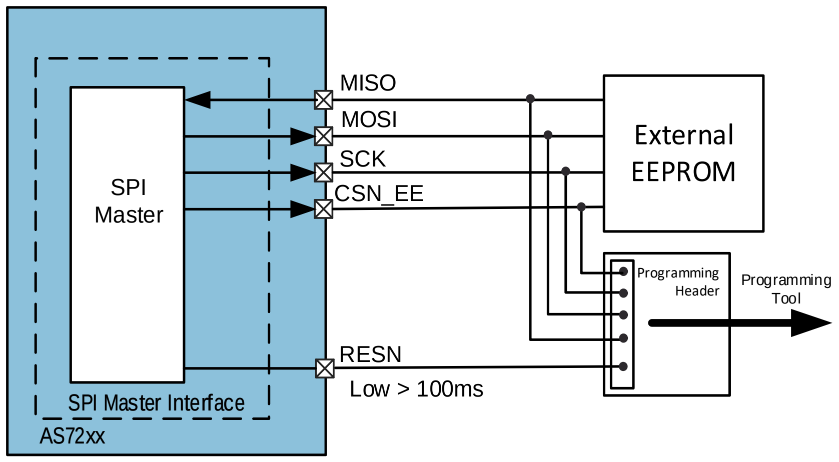
A. First Time EEPROM Programming

Use one of these two options:

Get the latest “*.bin” file (256KB) from ams for the specific AS72xx device. Have the EEPROM manufacturer (STMicro, etc.) or one of their distributors use the “*.bin” file to program the EEPROM. This is a service they routinely provide.

Or, use the SPI Programming setup as shown below to program the EEPROM (assumes programming tool access to the EEPROM is available). Make sure to keep RESN Low during programming as shown.

Figure 4: Direct SPI Programming of EEPROM after System Assembly



Contact Information

For further information and requests, e-mail us at:

ams_sales@ams.com

For sales offices, distributors and representatives, please visit:

www.ams.com/contact

Headquarters

ams AG

Tobelbaderstrasse 30

8141 Unterpremstaetten

Austria, Europe

Tel: +43 (0) 3136 500 0

Website: www.ams.com

Copyrights & Disclaimer

Revision Information

Initial version v1-01