



PROGRAMMING VIA USB

VERSION 0.0
05/05/2011

Am29F032B ROM introduction

The Alien Flash cartridge contains four Am29F032B 4MB Flash ROMs. The ROMS can be addressed and read like any other ROM. However, to program or erase the ROMS a command sequence must be sent to the ROM chips. A command sequence is simply a series of reads or writes of specific Data to specific address locations.

PROGRAM COMMAND SEQUENCE

Address	Write	Address	Write	Address	Write	Address	Write
\$555	#\$AA	\$2AA	#\$55	\$555	#\$A0	PA	#PD

PA = Address of the memory location to be programmed.

PD = Data to be programmed at location PA.

Note: Address bits A21–A11 are don't cares except for where PA is required.

As can be seen, four write cycles are required to program one ROM byte. Programming a byte can only program a bit from a "1" to a "0", therefore before programming the ROM, the sector that is going to be programmed must be erased first. Similarly to programming a byte, there is also a command sequence for erasing a sector.

PROGRAM BYTE COMMAND SEQUENCE

Address	Write										
\$555	#\$AA	\$2AA	#\$55	\$555	#\$80	\$555	#\$AA	\$2AA	#\$55	SA	#\$30

SA = Address of the sector to be erased. Address bits A21–A16 select a unique sector.

Note: Address bits A21–A11 are don't cares except for where SA is required.

A sector can take several seconds to erase, it's possible to monitor when the sector has completed erasing by the reading toggle bits, for more information on reading toggle bits, other command sequences and additional information for the Am29F032B ROM, refer to the [Am29F032B Data Sheet](#).

The four ROMS in the Alien Flash total 16MB and are arranged as follows:

FLASH ROM 0: addressed at \$000000 - \$3FFFFFF

FLASH ROM 1: addressed at \$400000 - \$7FFFFFF

FLASH ROM 2: addressed at \$800000 - \$BFFFFFF

FLASH ROM 3: addressed at \$C00000 - \$FFFFFF

Care must be taken when righting a command sequence to a ROM that the entire command sequence is written to the correct target ROM.

Do not simultaneously erase sectors on separate ROM chips (during a ROM erase an Am29F032B consumes higher current than usual).

FT232R CHIP

The Alien Flash incorporates a Future Technology Devices FT232R integrated circuit. The FT232R is a USB to serial UART interface. Future Technology Devices provides royalty-free Virtual Com Port (VCP) and Direct (D2XX) drivers for the FT232R, eliminating the requirement for USB driver development. Information on the FT232R can be found on the [FTDI website](#) or the [FT232R Data Sheet](#).

As default, the FT232R in the Alien Flash is set to work with the Direct (D2XX) drivers as this is many times faster than using the Virtual Com Port (VCP) drivers. The Alien Flash uses a Baud rate of 3000000.

Information for using the D2XX drivers for application development can be found in the [D2XX Programmer's Guide](#).

Alien Flash Commands

While the Alien Flash is being programmed via the USB, the host has direct access to the four FLASH ROMS in the Alien Flash. The Alien Flash can accomplish this by tristating the normal C64 lines to the ROM chips and switching the lines from the USB logic in.

To tristate the ROM chips so that the USB logic can have access, the "Tristate Mode" of the Alien Flash must be entered, this is done by sending \$00 followed by \$55 \$55 \$55 \$55. Tristate mode is exited at any time that the USB logic receives an invalid command, for example, when programming the Alien Flash is completed simply send \$00 to hand the access to the ROM chips back to the normal C64 lines.

Commands to read and write bytes from the ROM can be sent to the Alien Flash.

Read Command (\$D5):

A read command is issued by sending \$D5 followed by 3 address bytes and a blank byte "\$00".

e.g. sending \$D5 \$FF \$01 \$00 \$00 will read a byte from \$001FF

Write Command (\$55):

A write command is issued by sending \$55 followed by 3 address bytes and a data byte.

e.g. sending \$55 \$01 \$00 \$00 \$40 will write #\$40 to address \$000001

To program a byte in a ROM, the "Program" command sequence must be written to the FLASH ROM first.

the command sequence is #\$AA to \$0555, #\$55 to \$02AA and #\$A0 to \$0555.

e.g. \$55 \$55 \$05 \$00 \$AA, \$55 \$AA \$02 \$00 \$55, \$55 \$55 \$05 \$00 \$A0, \$55 \$01 \$00 \$00 \$40 would be required to program #\$40 to address \$000001

Care must be taken to ensure that the command sequence is written to the correct ROM, for example if we are programming an address somewhere between \$400000 - \$7FFFFFFF the command sequence would have to be \$55 \$55 \$05 \$40 \$AA, \$55 \$AA \$02 \$40 \$55, \$55 \$55 \$05 \$40 \$A0