FF SYSTEMS

*" AFFORDABLE REPEATER CONTROL SOLUTIONS "*

# FFLASH Programmer Rev 3.85
# for the FF-800 repeater controller

## Table of Contents

*Revised 10/11/2024*

## Warranty

FF Systems warrants its products to be free from defects for one year from the date of shipment. FF Systems may opt to repair or replace (at our option) any defective product. FF Systems does not warrant any defect due to lightning or other natural disaster. Any user modifications or repairs to any product sold by FF Systems will void this warranty. All returns must be accompanied by a Return Material Authorization (RMA) number provided by FF Systems prior to shipment. Shipments that do not have the proper return authorization prominently noted on the outside of the package will not be accepted. The purchaser is responsible for all shipping charges for any service procedure(s) performed by FF Systems (including warranty service).

## 30 Day Money Back Guarantee

FF Systems offers a 30 day trial period on all products. The 30 day trial begins 5 calendar days after shipment from FF Systems. If the purchaser decides to return a product they must contact FF Systems for a Return Material Authorization (RMA) number on or before 6:00pm central time within 35 calendar days of the original shipment date. The purchaser then has fourteen additional days to return the merchandise to FF Systems. FF Systems claims no liability for lost or misdirected shipments. The purchaser is responsible for all shipping charges.

## Product Returns

If merchandise incurs damage, it must be returned to FF Systems for warranty determination. The unit should be well packed and insured for its full retail value as FF Systems is not responsible for damages incurred during shipment. Make sure the RMA is clearly marked on the exterior of the package.

## Contact Address

e-mail: joeh@ke0ff.org

web: www.ke0ff.org

## Introduction

The FFLASH is a combination of hardware firmware for the FF-800 Repeater Controller that allows users to reprogram the FF-800 FLASH ROMs.  This allows the user to easily upgrade their FF-800 firmware without having to disassemble the controller or remove components.  The combination of the FF-Remote and FFLASH options is especially valuable to users with systems located on mountain tops or any other location where access is difficult.

The FF-Remote is an optional interface board that is used to control the access and programming modes of the FFLASH option and is required to allow remote control of the FLASH programming operations.  Control access is via DTMF commands which can be provided by a receiver or from the FF-800 telephone interface.  Several commands are present to allow control of the FFLASH system -- also, there are several utility commands that can be used to test or troubleshoot various repeater system components.

The FFLASH/FF-Remote system is designed to connect to a PC running a terminal emulator program such as Hyperterm, PROCOMM, QMODEM, or Kermit.  The appendix provides a listing of terminal parameters for some of these programs.

Throughout this manual, the terms "FFLASH" and "programmer" are used interchangeably.  The user should note that the term "FFLASH" physically refers the FF-800 while it is running the FLASH programming firmware.  This firmware "takes over" the operation of the FF-800 and is used to facilitate the transfer of data between the FLASH EEPROMs and the host computer.  When not in use, the FFLASH firmware is hidden and the normal FF-800 repeater controller firmware is accessed.

## FFLASH Overview

The FFLASH uses RS-232 serial communications to transfer program data between a host computer and the FF-800.  The communications parameters are as follows: 8 data bits, no parity, and 1 stop bit and 9600 baud (the FF-Remote allows other baud rates to enable transfers via modem).  Any host computer with a terminal program can be used with the FFLASH.  The main criteria for choosing a terminal program lies in how the terminal program handles handshaking.  The FFLASH requires software (XON/XOFF) handshaking for ASCII transfers.  The system also supports xmodem (checksum) transfers which have their own built-in handshaking and error recovery.  Therefore, XON/XOFF is not required if xmodem transfers are the only method used to send or receive data.  Also, note that the FFLASH xmodem transfers use checksum error detection.  Xmodem transfers that use CRC error detection will not work with the FFLASH.

XON/XOFF handshaking refers to the two ASCII characters that are used to perform flow control in "software" handshaking during ASCII transfers (XON/XOFF is not used in xmodem transfers).  The FFLASH is designed to offer good performance with software handshaking providing that a critical timing parameter is configured on the host computer.  In the RECEIVE mode (data from FFLASH to the host computer),  The FFLASH may be halted by an XOFF at any time.  When receiving large files (ie., a Motorola or Intel format read command on a large capacity memory device) this is useful to prevent data loss while the host system is writing to disk.  Of course, the terminal program must be designed to actually send the XOFF character when it knows it may lose data.  When TRANSMITTING data to the FFLASH, The programmer will send an XOFF character after each <cr> (0x0D) character in the input stream.  The terminal program should then stop sending data until it receives an XON from the programmer.  The critical timing parameter in the terminal program is the amount of time that the terminal program waits after each occurrence of the <cr> -- this should be at least 5ms (10ms at 1200 baud).  If this timing value is not set properly, the FFLASH may not have enough time to send the XOFF character and have it be recognized by the host computer before the host starts sending data from the next line.  If this happens, data can be lost resulting in a communications error.  Some programs only offer delays in

1/10ths of a second. Longer delays do not pose a problem to the handshaking, but they do increase read/program times. A list of tested terminal programs and their required settings can be found in appendix A.

## FFLASH connections and options

The FFLASH option includes an adapter interface that controls the programming operation. The FF-800 RS-232 port is used to communicate with the FFLASH firmware (P27 on the FF-800, or P210 on the FF-180 chassis). The adapter connects to the 10 pin connector P28 (located next to SW2 on the FF-800 board). Note that P28 is a 6 pin connector on older versions of the FF-800. This adapter is used to set the signals required to engage the FFLASH mode on the FF-800 and should remain connected tohe FF-800 at all times.

There are two versions of the adapter. The older version features a toggle switch and a push button reset switch. The latest version features a multi-function push button reset/mode switch and two status LEDs. While the operation of these modules differs, they both accomplish the same function.

## Rev B/C Single Switch Module Operation (for newer module versions)

The single switch module enters the normal mode upon power up. Normal mode is indicated when the "MODE" LED is off. A brief CW message is flashed on the MODE LED on power up. The message is "FFLASH x   I" where "x" = a single digit version number. The RESET LED is illuminated whenever a an FF-800 reset is detected. Note that the module will detect resets no matter what the source.

The FF-800 CONFIG DIP switch (SW2) must have all 4 switches in the "OFF" position while the FFLASH is connected. If the FFLASH adapter is removed, SW2-1 on the FF-800 must be moved to the "ON" position. In installations which have the FF-180 chassis, the FFLASH module is typically mounted on the back of the chassis at the A1 connector position. The FFLASH reset switch is accessed through the A2 opening just below the A1 connector position. The reset switch is recessed to prevent inadvertent operation. When viewing the chassis from the back, the REST switch is on the left, the MODE LED is on the top-right, and the RESET LED is on the bottom-right.

It should be noted that it is possible (though unlikely) for FF-800 configuration data to become corrupted during FFLASH programming operations. Before entering the PROG mode, be sure to use the FF-800 SAVE command to save a copy of the current FF-800 configuration. If a data corruption occurs, the SAVEd copy can be LOADed after the new FF-800 firmware has been programmed and reset.

SET PROG Mode:
Connect the computer serial port to the FF-800 serial port. Start the terminal program, set its parameters for communications with the FFLASH (see appendix A), and apply power to the FF-800. Press and hold the FFLASH reset switch until the MODE LED lights (approx 6 seconds). Then send a break command from the computer (most programs use <cntl>+<break> to send an RS-232 break). Note that revision C users do not need to send a break after reset. Also, a <cntl>+<@> (ASCII NUL) may be used in place of the break sequence with some programs. For FF-800 installations that DO NOT have the U532K option installed, it is necessary to perform a reset/break sequence two more times (for a total of three). Note that once the MODE LED is lit, is is not necessary to press and hold the reset switch. At this point, the FFLASH start-up message should be visible, and the ;> prompt should be displayed. The FFLASH is now ready to accept programming commands. Appendix B shows a sample session and should be used as a guide. Most users will find that only a handful of the commands presented in this manual are required to update the FF-800 firmware. The sample session in appendix B illustrates all of the operations

normally required to program the FF-800 firmware. A typical programming session will take 10 to 15 minutes at 9600 baud.

Return to NORM Mode:
After completing the programming operation, you may either press and hold the RESET switch until the MODE LED goes off (approximately 6 seconds), or simply power cycle the FF-800. At this point, the new FF-800 firmware should begin operation, and the FF-800 power on message should be seen at the terminal port and heard at the repeater output. The programming session is now complete, and the FF-800 may resume its normal operation.

Rev C Options:
The revision C FFLASH module includes a serial monitor that samples serial data that is sent to the FF-800. This allows the module to interpret commands sent via the FF-800 serial connection. There are three basic commands available: Reset in current mode, Reset in NORM mode, and Reset in PROG mode. In addition to these basic commands, there are several maintenance commands and options that are described in Appendix E.

All of the FFLASH commands begin with at least 3 seconds of no data followed by <CNTL>-Y. The rest of the command data must be entered with no more than 3 seconds between characters.

| | |
|---|---|
| simple reset: | `<CNTL>-Y R R` |
| NORM reset: | `<CNTL>-Y R N` |
| PROG reset: | `<CNTL>-Y R P` |

Note that in PROG mode, it is not necessary to enter a break sequence after a reset with the rev C module. This applies to both the push button and serial reset modes.

## Rev A, Toggle Switch Module Operation (for older module versions)

The toggle switch should be left in the "normal" position when not accessing the FFLASH system. In addition, the FF-800 CONFIG DIP switch (SW2) must have all 4 switches in the "OFF" position while the FFLASH is connected. If the FFLASH adapter is removed, SW2-1 on the FF-800 must be moved to the "ON" position. In installations which have the FF-180 chassis, the FFLASH module is typically mounted on the back of the chassis at the A1 connector position. The FFLASH reset switch is accessed through the A2 opening just below the A1 connector position. Both the reset and mode switches are recessed to prevent inadvertent operation. When viewing the chassis from the back, the mode switch is in the NORM position when flipped to the LEFT, and in the PROG position when flipped to the RIGHT.

The toggle switch on the FFLASH controls the mode of the FF-800. In the "NORM" position, the FF-800 is configured to operate normally using the firmware that has been programmed into U3 and U4. The "PROG" mode sets the FF-800 for memory device programming and the programming firmware becomes active when the reset switch is depressed.

It should be noted that it is possible (though unlikely) for FF-800 configuration data to become corrupted during FFLASH programming operations. Before entering the PROG mode, be sure to use the FF-800 SAVE command to save a copy of the current FF-800 configuration. If a data corruption occurs, the SAVEd copy can be LOADed after the new FF-800 firmware has been programmed and reset.

SET PROG Mode:
Connect the computer serial port to the FF-800 serial port. Start the terminal program and set its parameters for communications with the FFLASH (see appendix A). Place the FFLASH mode switch in the PROG position and apply power to the controller. After power is applied, press reset and then send a break command from the computer (most programs use <cntl>+<break> to send an RS-232 break). Also, a <cntl>+<@> (ASCII NUL) may be used in place of the break sequence. Perform this reset/break sequence two more times (for a total of three). At this point,

the FFLASH start-up message should be visible, and the ;> prompt should be displayed. The FFLASH is now ready to accept programming commands. Appendix B shows a sample session and should be used as a guide. Most users will find that only a handful of the commands presented in this manual are required to update the FF-800 firmware. The sample session in appendix B illustrates all of the operations normally required to program the FF-800 firmware. A typical programming session will take 10 to 15 minutes at 9600 baud.

Return to NORM Mode:
After completing the programming operation, the FFLASH mode switch is returned to the "NORM" position, and the power to the FF-800 is disconnected for a minimum of 5 seconds. After power is reapplied, the new FF-800 firmware should begin operation, and the FF-800 power on message should be heard. At this point, the programming session is complete, and the FF-800 may resume its normal operation. NOTE: Users who have the U532K option installed may simply switch to NORM and then reset (power off cycling is not required w/ the U532K upgrade).

## In Case of Difficulty

If the FF-800 does not respond normally after the programming operation, perform following procedure:

1) Try to log on to the FF-800 serial port. Set the terminal program to the baud rate that the FF-800 was set for prior to the programming operation (even though the FFLASH operates at 9600 baud, the controller baud rate should not have been changed). Issue the FF-800 logon command. If you are able to log on, try the "ID" command. If there is no response, or if the ID is incorrect, try resetting the controller.

2) If the log on is unsuccessful, or if the controller appears to be behaving erratically, there are two possible problems: 1, the configuration was corrupted by the programming operation, or 2, there was a problem with the programming operation. First, assume that the programming operation was at fault. Return to the PROG mode and VERIFY all files. If the data all checks OK go to (3). If there is a problem with the data, repeat the erase/program procedure as shown in appendix B. If, after entering the NORM mode, the controller still does not respond correctly, go to (3).

(3) The configuration is corrupted. If you are able to log on to the FF-800 serial port proceed to (4) -- if you can not log on, the FF-800 default settings need to be restored so that the LOAD command can be executed to reload the original configuration settings. This requires that the top panel to the controller chassis be removed. After removing the panel, locate the FFLASH control module located at the back of the chassis. Remove the FFLASH connector and press reset (or apply power). If there is an audio amplifier connected to the FF-800, the INIT messages for each of the INIT functions can be heard as they are completed. If no amplifier is connected, then it may be helpful to manually key one of the transmitters connected to the FF-800 and monitor the status of the INIT procedure using a receiver tuned to the transmitter's output. If no feasible method of monitoring the audio is available, then wait at least 60 seconds before proceeding. Once the procedure is complete, the controller will reset and send its power-up message. Remove power and reinstall the P28 connector (be sure that the FFLASH is in the NORM mode). Reapply power.

(4) After logging onto the serial port, follow the procedure in the FF-800 manual for using the LOAD command to load the previously saved configuration file.

## FFLASH display status

Starting with V3.85 and above, the FFLASH provides status information via the FF-8010 display module. The display feature is automatic for those systems which have a display module, and there are no user programmable configuration options.  The following describes the display actions:

1) The DV LED is a command in progress indication.  It is illuminated when a command is executing.

2) The RING LED is an error indicator.  All errors except checksum errors are reflected here.

3) DVL is an XMODEM retry indicator.  This indicator simply indicates that an xmodem retry was required and is not an error indication.

4) The eight (8) OUTPUT LEDs form a "gas gauge" that produces a linear, "scrolling" sequence to indicate that data is being processed during programming or read operations.  The OUTPUT LEDs are all on after reset and will go off after the first FFLASH command.

## FFLASH functions

There are several commands that control the operation of the FFLASH -- these commands are entered in "command-line" format with a command keyword followed by any required data. Commands in progress can be aborted with an <esc> (0x1B) character.

Keywords and data must be separated by one or more spaces. Also, most keywords only require the first 1 to 3 characters to be recognized. The descriptions that follow include the keyword, minimum keyword, data, and a description of the command and data parameters. Parameters in brackets [] are optional. The omission of all parameters (where they are optional) will interrogate the current status of the command. The following describes the format of the command descriptions in this manual (all numeric parameters are in "0x" hexadecimal format):

(keyword) [parameters] (minimum keyword)
(command description...)

EXAMPLE:

### TEST [test value] T

Test for blank device

This command scans the device from LOLIM to ...

## TEST [test value] T

Test for blank device

This command scans the device from LOLIM to HILIM and looks for any non 0xFF byte. If the optional hex [test value] is included, then this value is compared and an error results if any locations are found that do not match [test value]. The test operation terminates at the first error location and an error message is displayed.

## READ [LOLIM] [HILIM]
## R

READ device data in Motorola S-record format

## INTEL [LOLIM] [HILIM]
## I

Read device data in INTEL hex format

These commands read the contents of the device and send the data to the host computer in the appropriate format. The current value of LOLIM and HILIM may be changed with the inclusion of the optional limit parameters (see LIMITS for more information on the function of these parameters). These values are in hex and [LOLIM] must be less than [HILIM] or the FFLASH will use the default limits of the device (see DEVICE command for the default limits of the supported devices).

The logical address of the output is formed from the physical device address plus the value of OFFSET. Thus, with offset = 0xE000, the first logical address of the device would be 0xE000.

## XREAD [LOLIM] [HILIM]
## XR

XREAD device data in Motorola S-record format (xmodem)

## XINTEL [LOLIM] [HILIM]
## XI

Read device data in XINTEL hex format (xmodem)

These commands are identical to the READ and INTEL commands except that the xmodem protocol is used to transfer the device data.

# DEVICE  [device_name]
# D

Set device type

The FF-800 contains two FLASH EEPROMs that are organized as three part numbers. Omitting [part#] causes the current device is returned.  The following lists the supported part numbers:

| | Default Limits | | |
|---|---|---|---|
| device name | LOLIM | HILIM | Description |
| U3 | 0x008000 | 0x00FFFF | 32K x 8 EEPROM (28F256) |
| U4LOW | 0x000000 | 0x00FFFF | 128K x 8 EPROM (28F010), A16 = 0 |
| U4HIGH | 0x000000 | 0x00FFFF | 128K x 8 EPROM (28F010), A16 = 1 |

It is up to the operator to correctly set the DEVICE before sending or receiving a data file.  The firmware files are typically provided as three separate text files and obey the following conventions:

| FILE NAME | Description |
|---|---|
| V???U3.S19 | U3 firmware file |
| V???U4L.S19 | U4 low firmware file |
| V???U4H.S19 | U4 high firmware file |

The "???" refers to the firmware revision minus any decimal point (.) that may be present; ie., revision 2.18 would appear as "218".  These files are available from the github site for ke0ff (www.github.com/ke0ff/FF800)

## VERIFY [LOLIM] [HILIM]
## V

VERIFY device data

This command readies the FFLASH for an incoming data file and the data are compared to the device using the current value of OFFSET.  The incoming data may be in Motorola S-record, or Intel HEX record format (the FFLASH will automatically accept either format).  The programmer displays each physical device address which fails to verify.

## XVERIFY [LOLIM] [HILIM]
## XV

XVERIFY device data (xmodem)

This command is similar to VERIFY except that the xmodem transfer protocol is used.  Because of the way that xmodem operates, there is no display of error addresses.  If there are any failures, they are indicated by a simple message at the end of the transfer.  The FFLASH only supports checksum xmodem transfers (CRC xmodem will not function correctly).

## PROG P
## WRITE W

Program device data

This command readies the FFLASH for an incoming data file and the contents of the data are programmed to the current device using the current value of OFFSET. The incoming data may be in Motorola S-record, or Intel HEX record format (the FFLASH will automatically accept either format). The programmer displays each physical device address which fails to program.

## XPROG XP
## XWRITE
## XW

Program device data (xmodem)

This command is similar to PROG/WRITE except that the xmodem protocol is used to send the data file. There is no error byte display for xmodem transfers.

## OFFSET [address] O

Value to be subtracted from logical file address on PROG or VERIFY
(also value to be added to physical device address on READ or INTEL)

The OFFSET command allows a data file to be "relocated" from the address that it was originally intended to occupy. This command is intended for special operations and should not be used under normal programming conditions.

## FLERA [ALL] FL

Erase FLASH device

This command begins a FLASH-device erase procedure on the current device. The "ALL" option may be used to erase both U3 and U4. If the U4 device is selected (high or low) the FFLASH will erase both halves. Thus, the user should only erase ONE of the U4 devices. If there are erase errors, the error addresses will be displayed. Error addresses are an indication that there may be a problem with the device or that the Vpp supply is out of specifications. If this type of error is consistently encountered, the user should contact FF-Systems for further assistance.

The erase time for U3 is approximately 10 seconds, while U4 will erase in about 40 seconds. The only response that is displayed during erasure is that of error bytes. If a byte fails the erasure process, its address is displayed as soon as it is discovered. If the device erases successfully, the "OK" message will be displayed at the end of erasure (two "OK" messages are displayed for the "ALL" option).

## FILL [byte] FI

FILL target device with data

This command will fill the device with 0xFF between LOLIM and HILIM. If the [byte] parameter is included, this is used as the fill pattern. NOTE: if FFSKIP is on, this command will have no effect when trying to fill 0xFF data.

## ERROR E

Read last error

Displays last error(s) encountered by the programmer. The following list describes the various error types:

verify error: error in PROG or VERIFY operation

checksum error: S-record or Intel Hex record checksum error

record data error: invalid character in S-record or Intel Hex record. All characters should be ASCII hex (0x30 - 0x39 or 0x41 - 0x46) -- all other characters cause this error (except for the leading "S" or ":" which is the record start character for Motorola and Intel, respectively).

serial error: too many characters received and data was most likely lost. Other errors are likely to accompany this error.

invalid line error: The "S" or ":" start character was not found on one or more data lines.

address limit error: a physical address was encountered that was outside the current LOLIM/HILIM boundaries. All data on the affected line(s) is ignored.

When any of these errors are encountered during a PROG or VERIFY operation, the operation continues until a valid end of file record is encountered, or until an <esc> (0x1B) is encountered. During non-xmodem PROG or VERIFY operations, the FFLASH will display the physical address of all data that fails to verify. When massive verify failures occur during PROG, the data flow will be severely restricted. Thus if programming is taking excessive time, it may indicate a massive verify failure (among other causes, this can be due to incorrect device type or a problem with the Vpp supply).

## CLEAR C

Clear errors

Clears any previous error(s). Some commands automatically clear errors when they are entered: TEST, PROG, WRITE, VERIFY, READ, INTEL.

## STAT S

Display programmer status

The current settings and error conditions are reported in the following format:
Device Type: (current device part#)
MFG/DEV codes: mm dd (mm = mfg code, dd = device code)
Offset = (current offset in HEX)
Prog limits (low/hi):
0xFF Skip: (on or off)
Last Error(s): (lists all current errors)

This command is useful to quickly check on the current programmer settings. If the MFG/DEV code displays: "!! VPP OFF !!" it is a signal that there is a problem with the Vpp supply OR that the selected memory device may have a problem.

## LIM [LOLIM] [HILIM] L

Set device address limits

## DEFAUL DEF

Set default address limits

These limits restrict programming or read operations to occur between the physical device addresses contained between LOLIM and HILIM (inclusive). During a program or verify operation, the FF-PGMR will generate an error if any address is encountered that is outside the LOLIM/HILIM window. The programming/verify operation will continue, however, and any addresses outside the limits will not be programmed or verified.

This command is intended for special operations and should not be used under normal programming conditions. The default limits are described with the DEVICE command.

## ? ?

List command keywords (?)

This command generates a list of the keywords used by the FF-PGMR. It is useful as a quick reminder but is not a "help list" as it does not contain any information about parameters or actions for any of the commands listed in this manual.

## FFSKIP [yes/no] FF

Set 0xFF skip status

For an EPROM that erases to 0xFF, considerable time savings is usually obtained by skipping the programming of 0xFF bytes. However, SRAM devices do not have an "erased state" and should not skip 0xFF programmed bytes. This command should be "yes" or "on" for EPROM or flash devices, and "no" or "off" for RAM devices.

This command is for factory test and should be set to "yes" for FFLASH programming operations.

## VERS VERS

Read FFLASH firmware version message

This command interrogates the firmware version message for the FFLASH system.

## BAUD [rate#] B

Set baud rate

This command allows the baud rate to be changed and is primarily for use by the FF-Remote interface. [rate#] is defined as follows: 0=9600 baud, 1=4800 baud, 2=2400 baud, 3=1200 baud, 4=600 baud, 5=300 baud, 6=150 baud, 7=75 baud.

## MODEMI [string] MODEM

Enter or examine modem initialization string

This command allows the user to enter a string that is intended to initialize a modem or TNC connected to the FF-800. This string is sent after the BAUD command IF MODENA = ON. This feature is only meaningful on FF-800 systems which have a 32K x 8 RAM or EEPROM installed at U5. It is intended for use with the FF-Remote to allow modem or TNC control when first enabling the FFLASH programming system.

Strings can contain up to 63 characters and are entered with the following protocols:

- Control characters are entered using the ^ character (ASCII 94, decimal) followed by the letter of the control code. Thus, and end of line (CR) character would be entered as ^M. NOTE: ^Z is used by the FFLASH to indicate end of string and should not be used in the input string (the FFLASH will ignore all characters following the ^Z).

- The repeat character function is indicated by the sequence: ^^cn. ^ = ASCII 94, decimal, c = character to repeat, and n = number of times to repeat the character (1 (ASCII 49 decimal) to 9 (ASCII 57 decimal). This sequence would be used to send an escape sequence to a modem (^^+3 would send 3 "+" characters to the modem). To place a single "^" character into the target string, the repeat function must be used as follows: ^^^1

- The ~ character (ASCII 126, decimal) is a delay command and each ~ results in 1 second of delay.

An example initialization string would reset the modem and place it in auto-answer mode:

```
MODEM ~~~^^+3~~~ATZ^M~~~ATS0=1^M
```

This string would delay 3 seconds, send "+++" and wait 3 seconds more. The modem reset command "ATZ" is next, followed by another 3 second delay. The last command "ATS0=1" sets the modem to answer on the first ring.

## MODENA [ON/OFF] M

Enable or disable modem initialization string

This command enables the modem initialization string. If disabled, the response to the BAUD command is "Set baud rate".

## DUPLEX [FULL/HALF] DU

Enable or disable character echo

Systems which are connected to a modem or TNC should have their duplex set at half to suppress character echo. Otherwise, any echoed characters from the modem would form an endless loop of echoed characters.

## BUFFALO BU

Enable MCU ROM

This is a test command and should not be used.

## HE addr HE

Read data at address

This is a test command and should not be used.

## HW addr data HW

Write data to address

This is a test command and should not be used.

The following commands are used to support advanced modem operations when used in conjunction with the FF-Remote:

**MED1 [string] ME**
Enter or examine modem control string#1

**AUX1 [string] AU**
Enter or examine modem aux. string#1

**MED2 [string] MED2**
Enter or examine modem control string#2

**AUX2 [string] AUX2**
Enter or examine modem aux. string#2

**LOGOFF LO**
Send modem logoff string (AUX1)

# Appendix A: Compatible terminal programs and settings

## Macintosh:

Program: **Crossbow** cross-assembler (this application has a terminal sub-function)
by: Onset Computers
P.O. Box 3450
Pocasset, Ma 02559
(508) 563-9000

Revision: 1.99N or higher
Settings: (all are under "TERMINAL" menu)
• BAUD RATE & PROTOCOL
Set baud rate to 9600 baud
Data bits: 8
Stop bits: 1
Parity: none
X-on/X-off: turn on local AND remote

• FILE TRANSFER
Send Prefix: empty
Send Suffix: empty
Character delay: 0
Line Delay: 5
Discard echoed text: leave unchecked (off)

• TERMINAL PREFERENCES
Display non-ASCII as Hex: checked (on)
Filter control characters: unchecked (off)
Input buffer: 200,000 bytes

To read a device in ASCII, simply type the appropriate read command and clear the terminal screen (command-B) before pressing the <return> key. After the FFLASH has finished sending data to the Mac, choose "Select All" (under the EDIT menu) and then choose "Save Selection" from the FILE menu. If the resulting file is to be transported to other systems, some editing may be required to remove extraneous lines. These lines may cause checksum or invalid line errors if the saved file is used to program or verify with other programmers. The FFLASH ignores any line that does not begin with an "S" or ":", so the extraneous lines will not cause any problems (as long as the "extraneous lines" do not contain valid commands).

An easier method to read is to use the XREAD command in conjunction with the Crossbow XMODEM RECEIVE function (under the TERMINAL menu).

## Appendix A: Compatible terminal programs and settings (continued)

## Macintosh:

Program: **Microsoft Works**

Revision: 2.00 and higher
Settings:
• COMMUNICATION SETTINGS (this window pops-up when a new comm. document is opened)

    TTY
    Baud rate: 9600
    Data bits: 8
    Stop bits: 1
    Parity: none
    X-on/X-off: selected
    Line delay: 5
    Character delay: 0

• COMMUNICATION

    Echo: off

To read a device, use the "CAPTURE TEXT" option under the COMMUNICATION menu. After the text capture has begun, issue the appropriate read command and wait for the FFLASH to complete the data send. The resulting file must be edited to remove invalid characters (such as the read command, the "OK" error report, and new command prompt) before the file can be used to program or verify.

This program may exhibit problems during program or verify operations and is not recommended as a primary interface.

Other Macintosh applications may be used to communicate with the FFLASH. If an application does not support XON/XOFF, programming can only be done using XMODEM (checksum) transfers.

## Appendix A: Compatible terminal programs and settings (continued)

## IBM (and compatibles):

Program: **Qmodem**
by: Mustang Software

Revision: 4.5 (test drive)
Settings:
There are a multitude of settings for this program, those listed below are the bare minimum required to allow the FFLASH function. Other settings may need adjustment according to the user's taste and application:

- SETUP PROGRAM
  MODEM
  COMMUNICATION PARAMETERS
  Baud rate: 9600 baud
  Data bits: 8
  Parity: none
  Stop bits: 1
  Set the COM port and IRQ -- Note: In order to avoid problems, be sure there are no IRQ conflicts with other drivers, interfaces, or internal modems. The FFLASH may appear to function with a conflict, but data may be lost without indication.

  TOGGLES
  Xon/Xoff: on
  RTS/CTS flow: off
  RTS when writing: off

  MODEM COMMANDS
  Initialization command: leave blank

To program devices, use the UPLOAD (Pgup) function and use DOWNLOAD (Pgdn) to read devices -- choose xmodem transfer mode for each. Because of broad variations in XON/XOFF implementations, PC/compatible users should only use XMODEM transfers to program, verify, or read data. FFLASH programming commands must be issued prior to the PC/compatible's UPLOAD command, and issue read commands AFTER the DOWNLOAD command. Be sure to use the XWRITE, XREAD, XVERIFY commands for programming and reading.

If ASCII transfers are used for read operations, sure to verify read files against the device from which they are read before removing the device from the programmer. While Qmodem has exhibited good reliability when configured as shown above, it is possible for the PC to loose a character during the file read. If any errors are detected in the verify, reread the device and check the read file again. Persistent problems may be indicative of a setup problem or IRQ conflict. If you are consistently losing data during device reads, check your setup and make sure that there are no IRQ conflicts. If you examine the read file with a text editor, check the line lengths -- except for the last two data lines, they must all be the same length.

## Appendix A: Compatible terminal programs and settings (continued)

## IBM (and compatibles):

Program: **TerraTerm** (www.github.com/TeraTermProject/teraterm)

Revision: 5.2 (or later)
Settings:
There are a multitude of settings for this program, those listed below are the bare minimum required to allow the FFLASH function. Other settings may need adjustment according to the user's taste and application:

- "Setup -> Serial Port"
  Set the COM port number.
  Set baud rate to 9600 baud
  Set data format to: 8 bits, 1 stop bit, no parity
  Tx Delay = 20ms/line

To program devices, use the "File-> Send File" for ASCII transfers, or "File-> Transfer-> XMODEM-> Send" for X-modem transfers. Because of broad variations in XON/XOFF implementations, PC/compatible users should only use XMODEM transfers to program, verify, or read data. FFLASH programming commands must be issued prior to the PC/compatible's UPLOAD command, and issue read commands AFTER the DOWNLOAD command. Be sure to use the XWRITE, XREAD, XVERIFY commands for programming and reading.

## Appendix B: FFLASH Programming Session Example

```
                                                      1st reset/break (no response)
rom-                                                  2nd reset/break
BUFFALO 3.61ff (int)
rx)
FF-800 FFlash programming firmware v3.82              3rd reset/break
copyright (c) FF-Systems, 1995                        power-up message
;>?                                                   list keywords
Command list:
TEST       READ       DEVICE     VERIFY     PROG
OFFSET     VERS       ERROR      STAT       LIM
INTEL      DEFAULT    ?          WRITE      FFSKIP
CLEAR      FLERA      FILL       HE         HW
BAUD       XVERIFY    XPROG      XWRITE     XREAD
XINTEL     MODENA     MODEMI     DUPLEX     BUFFALO
MED1       AUX1       MED2       AUX2       LOGOFF

;>d u3                                                 set device
Device Type: U3
MFG/DEV codes: 01 A1
;>t                                                   test for blank (0xFF)
Error!

;>flera all                                           erase all devices
OK                                                    U3 erase operation was OK
OK                                                    U4erase operation was OK

;>s                                                   interrogate  status
Device Type: U3
MFG/DEV codes: 01 A1                                  check mfg/dev codes
Offset = 008000
Prog limits (low/hi): 008000 00FFFF
0xFF Skip: OFF
Last Error(s): OK

;>xw                                                  program U3 using xmodem
Press <cntl-X> to abort...
                                                      send  U3.S19 using
                                                      XMODEM (checksum)
OK or checksum error (this is OK, but any other error indicates a problem)
;>d u4
Device Type: U4LOW
MFG/DEV codes: 20 07
;>t                                                   test for blank (0xFF)
OK
```

## Appendix B: FFLASH Programming Session Example (continued)

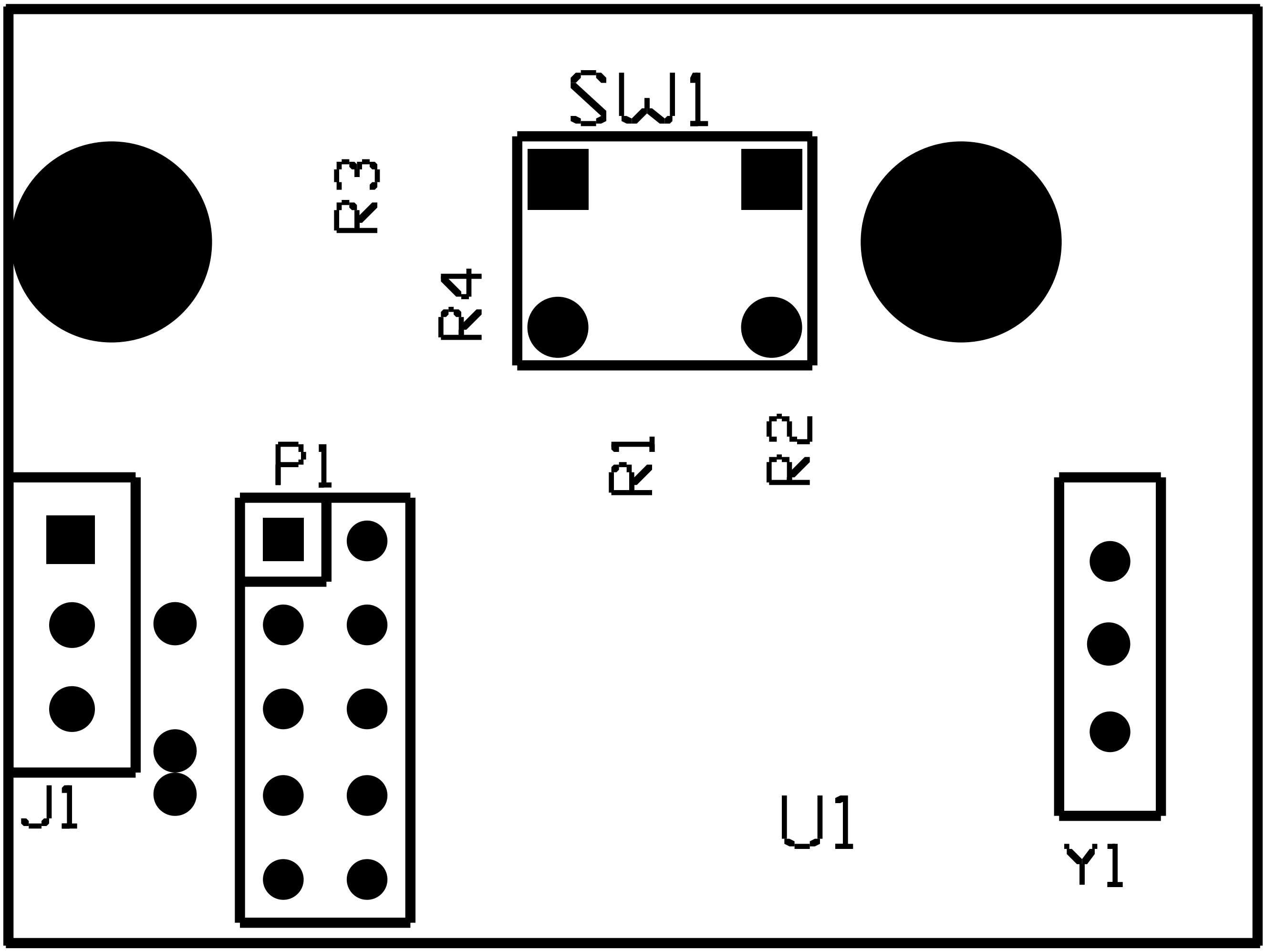
```
;>xw
Press <cntl-X> to abort...
                                                      send  U4L.S19 using
                                                      XMODEM (checksum)
OK or checksum error (this is OK, but any other error indicates a problem)
;>d u4h
Device Type: U4HIGH
MFG/DEV codes: 20 07
;>t                                                   test for blank (0xFF)
;>xw
Press <cntl-X> to abort...
                                                      send  U4H.S19 using
                                                      XMODEM (checksum)
OK or checksum error (this is OK, but any other error indicates a problem)
                                                      switch FFLASH to "norm"
                                                      and  reset the FF-800
```

NOTE: Depending on the specific firmware changes for a given update, the U4H.S19 file may not be provided. If it is not, then skip the programming procedure for that file. All revisions above V3.00 will require the U4H.S19 file.

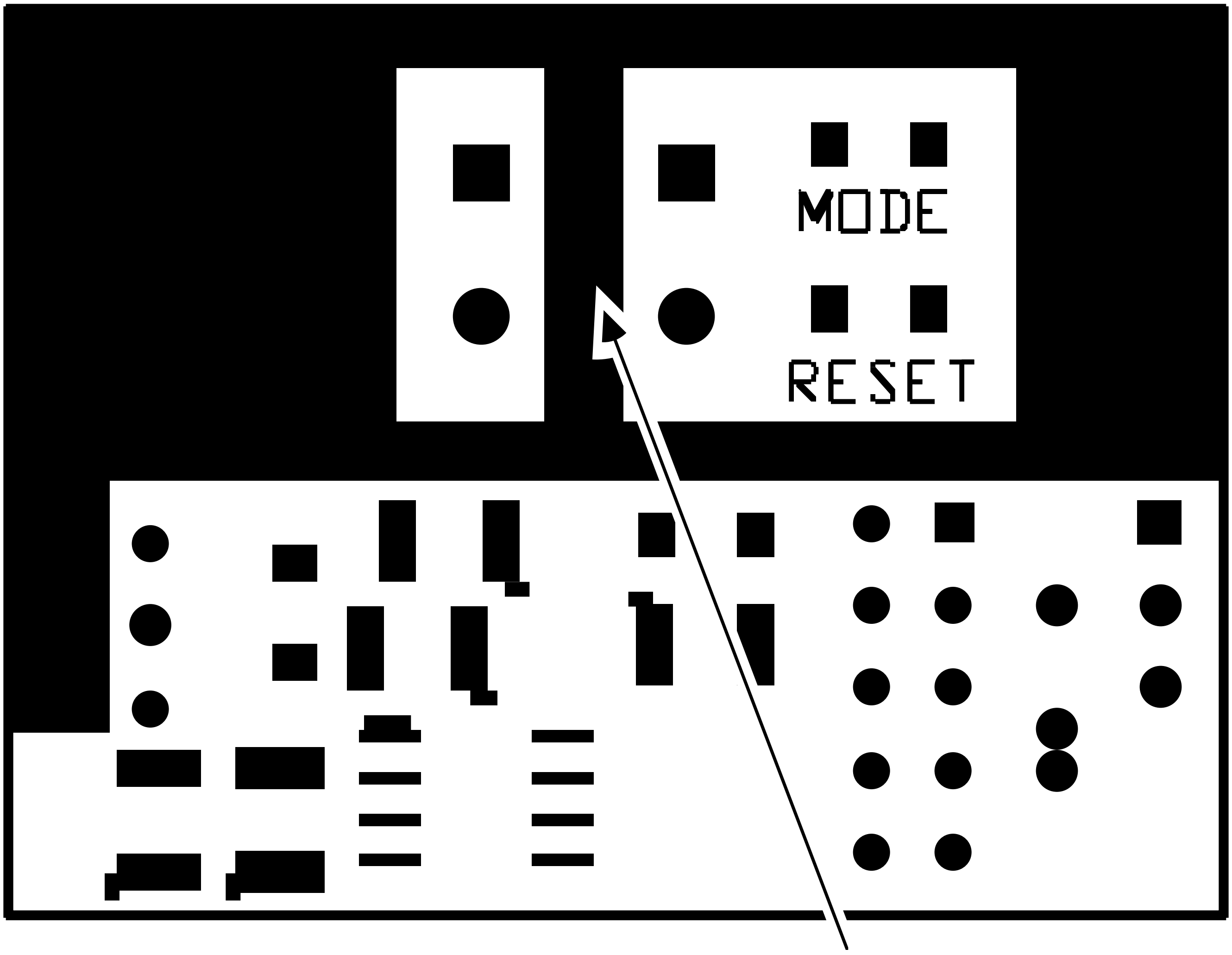
## Appendix C: Single Switch (Rev B) FFLASH Module Parts List & Layout

*Note: See FF-800 Technical Manual (latest edition) for Rev C (or later) schematic and parts list.*

| Parts List | |
|---|---|
| C1, 2, 7 | 0.1 µF ceramic capacitor, 50V |
| C3, 4, 5, 6 | 10 µF electrolytic capacitor, 16V |
| | |
| R1 | 8.2 KΩ, 1/8 watt resistor |
| R2 | 68 KΩ, 1/8 watt resistor |
| R3, 4 | 470Ω, 1/8 watt resistor |
| | |
| X1 | 4.00MHz ceramic resonator |
| | |
| D1 | Amber LED, surface mount |
| D2 | Green LED, surface mount |
| | |
| U1 | 68HC705K1 microcontroller I.C. |
| U2 | MAX662 DC-DC converter I.C. |
| Q1 | 2N2222A NPN transistor |
| | |
| SW1 | SPST momentary push button switch |

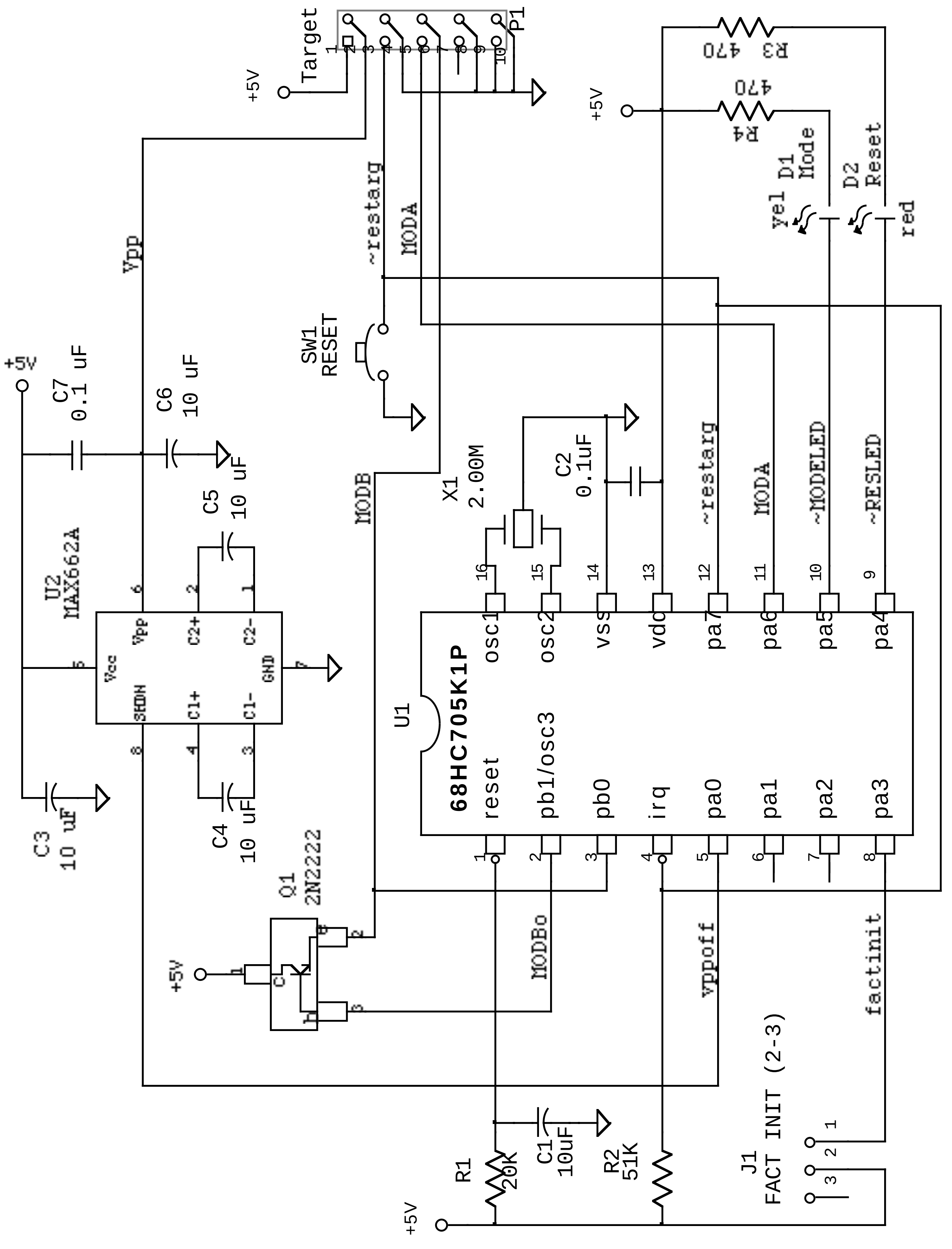


Revision B Top View (inside chassis)



reset switch

Revision B Bottom View (via chassis access hole)

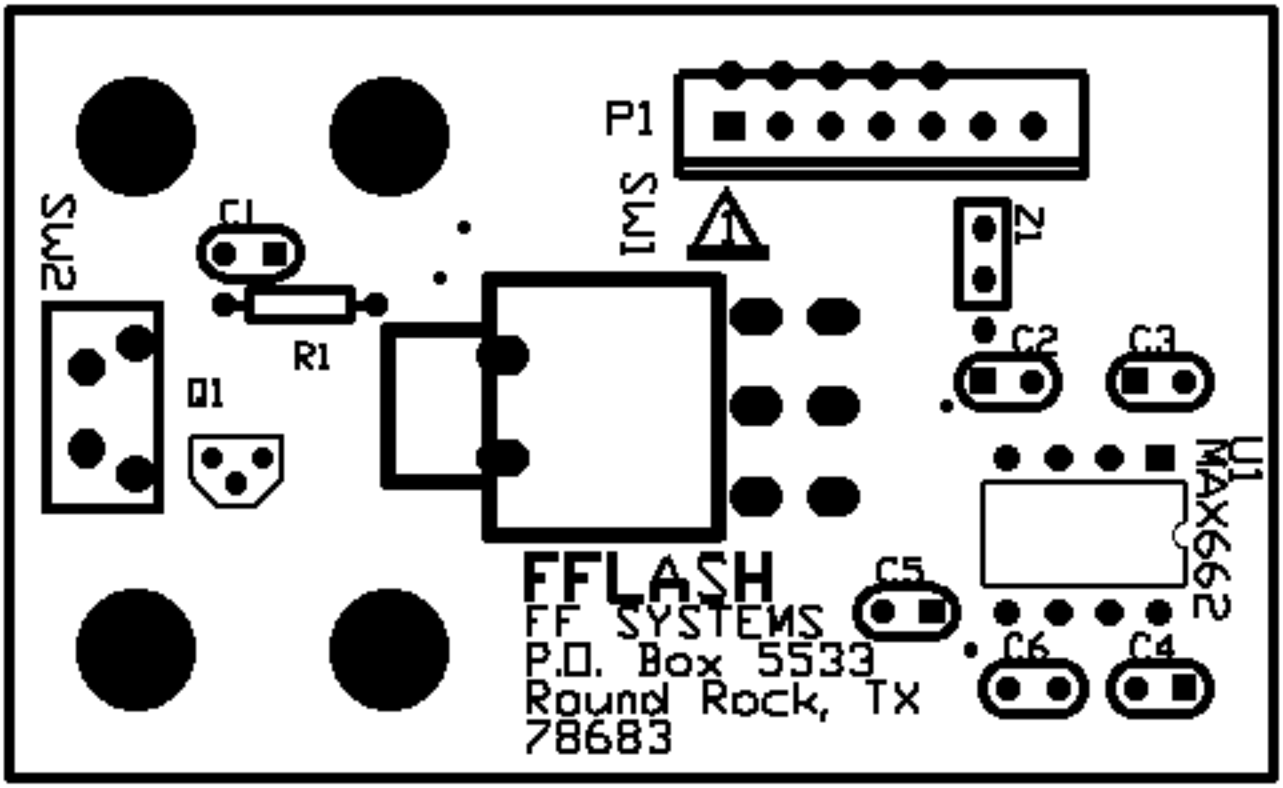


Revision B Schematic

# Appendix D: Dual Switch (Rev A) FFLASH Module Parts List & Layout

<u>Parts List</u>

| | |
|---|---|
| C1, 6 | 0.1 µF ceramic capacitor, 50V |
| C2, 3, 4, 5 | 10 µF electrolytic capacitor, 25V |
| R1 | 2.4 KΩ, 1/4 watt resistor |
| U1 | MAX662 DC-DC converter I.C. |
| Q1 | 2N2222A NPN transistor |
| SW1 | DPDT toggle switch |
| SW2 | SPST momentary push button switch, right angle mount |
| Z1 | 100mA PTC self-resetting fuse |



## Appendix E: Rev C Extended Commands and Options

Since the revision C FFLASH module features a serial command interface, it has the ability to offer greater configuration flexibility than previous versions of the reset interface. This section describes the extended commands and options. Note that these features and options do not generally require attention on the part of the operator since the revision C module is operationally a superset of the revision B modules. However, the added serial functions will greatly aid the operator and allow remote programming sessions to be performed if needed.

### Command summary:

| | |
|---|---|
| `<cntl-Y> RR` | toggle FF-800 reset |
| `<cntl-Y> RN` | set NORM mode, toggle reset |
| `<cntl-Y> RP` | set PROG mode, toggle reset |
| `<cntl-Y> RQ` | querry FFLASH module software version |
| `<cntl-Y> !nnnn` | Set password to "nnnn", where "n" is any alphanumeric character (must be 4 characters), "----" forces password to be invalid |

If a valid password is programmed, the above commands become:

| | |
|---|---|
| `<cntl-Y> nnnnRR` | toggle reset |
| `<cntl-Y> nnnnRN` | set NORM mode, toggle reset |
| `<cntl-Y> nnnnRP` | set PROG mode, toggle reset |
| `<cntl-Y> nnnnRQ` | querry FFLASH module software version |

NOTE: Prior to entering a command, there must be at least 3 seconds of delay with no characters sent for the <cntl>-Y commands to be recognized.

Once programmed, the password can only be erased by placing a temporary jumper accross the "irq" pads on the FFLASH and applying power (an "OK" message will flash on the mode LED after the power on version message). The password can only be changed by first erasing and then re-programming. In situations where a password is not desired, it is recomended that a NULL, or invalid, password be programmed. A NULL password is any password that begins with a dash "-" character. Otherwise, it is possible for an unknown password to be inadvertently programmed which would prevent access to the serial commands.

It should be noted that the reply to the software version command is actually sent to the FF-800 processor serial input, not to the remote terminal directly. However, if the FF-800 serial port is in LOGON mode, or in FFLASH programming mode, the reply will be echoed to the remote terminal by the FF-800 software. If the FF-800 is in LOGOFF mode the reply will not be echoed.

### Baud Rate Selection

The FFLASH supports 4 different baud rates. The baud rate is selected by a solder pad jumper. There are 4 jumpers along the lower edge of the module. If no jumper is shorted (the factory default condition), or if the "9600" jumper is shorted, the baud rate is 9600. Moving right to left from the "1200" jumper, the baud rates are 2400, 300, and 9600 (the "9600" pad). Note that the 9600 pad is obstructed by P1 and is generally not used. When changing the baud rate jumper pads, be sure to completely remove the solder from the existing jumper before continuing. If multiple pads are shorted, the resulting baud rate will be unpredictable. Also, be sure to remove power when changing the baud rate setting.

The mix of selectable baud rates for the FFLASH does not exactly match the selectable baud rates for the FF-800. However, the four available rates were selected to offer coverage for the most likely rates modem that might be encountered. The FFLASH baud rate is generally set to the baud rate supported by the modem used with the FF-800 (or 9600 baud if using a direct connection). This would be the same rate at which the FF-800 operates.

## PROG Mode Reset

The FFLASH programming software in the FF-800 always defaults to 9600 baud on reset. Therefore, the FFLASH module must send a command to the FF-800 to set the correct baud rate. This is done any time the FFLASH module baud rate is set to 300, 1200, or 2400 baud.  The user only needs to be aware of this activity as it adds 2-3 seconds to the reset sequence.  Also, the FFLASH "V3.85" boot message will not be visible on systems that operate at baud rates other than 9600.  This message will be viewed as "garbage" when you reset in PROG mode.  This is normal and should not cause any serious problems with the system.

Once the PROG mode reset is complete, you should see a command prompt, and the FFLASH software should accept commands.

FFLASH Rev-C Install Notes
DOC date: 10/11/24, REV -

The FFLASH is a small control board that allows the software load of the FF-800 repeater controllers. It controls the reset modes and facilitates the programming configuration process. It is intended to attach to an unused DB-9 cutout on the FF-180 (or similar) chassis. Since the Revision C device allows the reset of the FF-800 to be accomplished using serial commands (<CTRL-Y> sequences), it is not required that the module be accessible via a DB-9 style cutout. This can be useful in the event that there is not a suitable cutout available. The photo below illustrates a typical mounting scenario.

A 10-conductor ribbon cable connects the FFLASH P1 to the FF-800 P28 connector (P28b on FF-800 boards with revision F artwork and a U29 interposer).

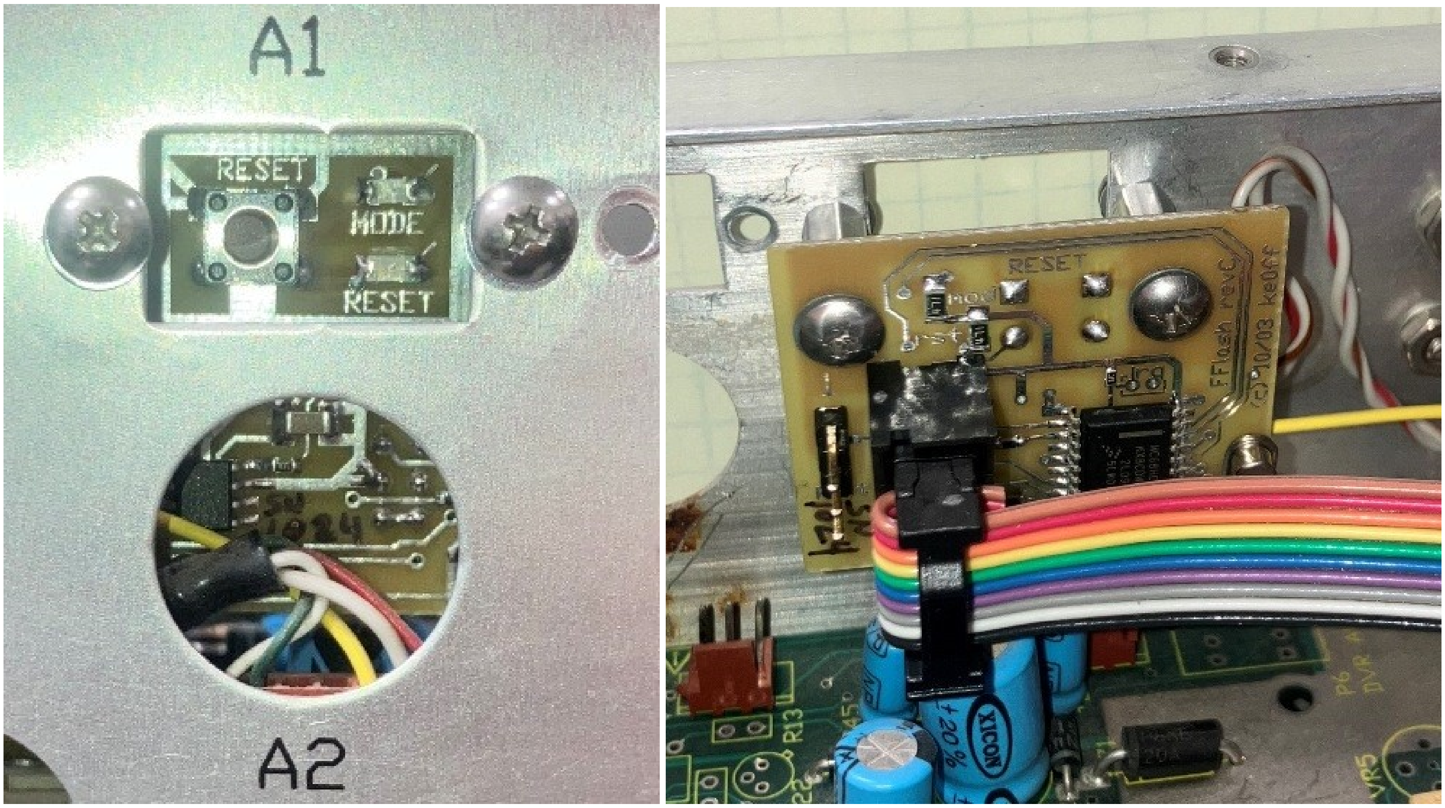


Photo: FFLASH FF-180 Chassis mounting.