BUTTLOAD

AVRButterfly ISP Programmer

By Dean Camera, 2007

For Buttload V2.1

SYNOPSIS:

The ButtLoad firmware allows you to transform a cheap, widely available "AVRButterfly" demonstration and development board from Atmel into a fully-fledged ISP programmer for all AVRStudio compatible devices. Using the pre-existing Butterfly bootloader, the programmer itself can have its firmware loaded quickly via a RS-232 (serial) interface. This feature allows the construction of the programmer without the need for a bootstrap cable or pre-existing programmer – thus the ButtLoad system does not suffer the "chicken and egg" dilemma of other homebrew programmers.

The AVRButterfly's extremely small price tag and host of onboard features makes it ideal for this purpose. Compatible with the V2 protocol used in AVRStudio 4.x and the ATAVRISP, ButtLoad takes the place of the latter system. Not only cheaper, the ButtLoad system has the unique ability to store a program (including flash and EEPROM memory as well as the fuse and lock bytes for the device) on its onboard non-volatile dataflash. With this feature, the ButtLoad system can be used to perform complete upgrades or programs in the field without the need for a computer.

The construction of the ButtLoad system is easy; all that is required component-wise is the level-shifting circuitry for the USI port and the optional bicolour status LED. In the case of the former this can just be a pair of resistors, as shown in "Connection Diagram.pdf".

Power consumption:

The following table shows the typical power consumption for different ButtLoad states. It assumes a bare Butterfly setup with no external circuitry attached.

Sleeping	~30µA
Active (in a menu awaiting user input)	~100µA
Active (in a function using the serial port)	~1.7mA

LOADING/UPGRADING THE FIRMWARE:

The following instructions assume you are using a Butterfly running the default bootloader. If you have programmed in a different bootloader (or if you have removed the bootloader) you will need to find alternate directions for loading the BUTTLOAD.HEX file into the Butterfly's MEGA169. Note: if you are not using a bootloader, the BOOTRST fuse bit should be disabled.

1) Compile the project with WinAVR (AVR-GCC). WinAVR is a free collection of C tools, plus an excellent compiler for the Windows platform. You can download the latest package from http://sourceforge.net/projects/winavr/.

2) After compiling, a new file named BUTTLOAD.HEX should be present in the ButtLoad project folder. Start AVRStudio4.

3) Connect your Butterfly to your computer's serial port, and remove all power to the board.

4) Reconnect the Butterfly's power source. Although no activity is present, the system is now in Bootloader mode.

5) Push the Butterfly's joystick inwards, and hold it down. Next, open AVRPROG from the Tools menu.

6) Select the BUTTLOAD.HEX file from the ButtLoad project directory. To program the Butterfly, click the Program button.

7) After the erase, program and verify cycles are complete, release the joystick, close AVRPROG and cycle power to the Butterfly.

Once these simple steps have been followed, ButtLoad is ready to be used. To intitialise the program, see the “Operation” section of this document.

If you are using the default Butterfly bootloader and receive programming errors at the flash address 0x940c, then please visit the AVRFreaks thread at http://tinyurl.com/oudt6 for an explanation of the problem and possible solutions.

CONNECTIONS:

Because the on-board dataflash is connected to the SPI interface, the code uses the USI system in three-wire mode to communicate with the slave AVR instead. This means that the two systems can run at different data rates without switching, and is also nessesary because the slave AVR does not have a /CS pin.

USI Interface:

Pin 1 (SCK) - Slave AVR SCK
Pin 2 (DI) - Slave AVR MISO
Pin 3 (DO) - Slave AVR MOSI
Pin 4 (GND) - Slave AVR GND

PORTF (As viewed from the TOP of the board):

Pin 1 (PF4) - Green (active-high) lead of a Bicolour LED (optional)
Pin 5 (PF5) - Red (active-high) lead of a Bicolour LED (optional)
Pin 3 (PF6) - /RESET line of slave AVR
Pin 9 (PF7) – Recovery mode target clock out
Pin 10 (GND) - Status LED ground if used (optional)

USART Interface:

Pin 1 – Receive relative to Butterfly
Pin 2 – Transmit relative to Butterfly
Pin 3 – Ground

Vin Interface:

Pin 1 – External sleep input pin
Pin 2 – Ground

Level shifting circuitry must be employed that can translate the 3.3V Butterfly signals to the target AVR's voltage and vice-versa at sufficient current to prevent damage to the Butterfly. See the included connection diagrams for simple level shifting systems (alternativly view ***http://tinyurl.com/r3yw7*** for ***s***everal other level shifting circuit designs).

If you wish to program 3V AVR parts, omit the two protection resistors shown in the schematics.

KNOWN ISSUES:

1) A maximum of 20 fuse bytes and 20 lock bytes can be stored in memory at any one time (writing the same fuse overwrites the existing value). If it is attempted to write more than this maximum, the extra bytes will be ignored.

OPERATION:

Initialising the system:

Once the firmware has been loaded using the bootloader, the power must be cycled to the device in order to reset it. To initiate the new firmware using the Atmel bootloader, push the joystick upwards briefly; this will cause the main menu to show.

Interfacing with ButtLoad:

ButtLoad communicates with AVRStudio in the exact same manner as the AVRISP. Details on the STK500 communication protocol (a subset of which is the AVRISP communication protocol) is avaliable from the Atmel website, application note 061 (avaliable from http://atmel.com/dyn/products/app_notes.asp?family_id=607). Details on the actual programming method can be found in application note 910 (also avaliable from http://atmel.com/dyn/products/app_notes.asp?family_id=607).

Status LEDs:

Adding the bicolour status LED to ButtLoad is optional, but can serve as a quick indication of the system status when the user is not in a position to easily read the LCD display. The status LEDs can show one of three colours:

RED	Programming (or error if flashing)
ORANGE	Busy
GREEN	Ready

The main menu:

The default function in the main menu is "AVRISP MODE". To scroll through the available menu items, push the joystick up or down. To select a function, press the joystick inwards briefly. You can access program info by pushing the joystick right (then push up or down to scroll and left to return).

When ButtLoad starts it is in the main menu only, NOT in the AVRISP mode. You will need to enter a function via clicking the joystick on the desired function name before use.

Before any function which requires the use of the USART ButtLoad will calibrate the internal RC osccilator. This process should take no longer than one second to complete.

AVRISP MODE:

This mode will cause the Butterfly to emulate an AVRISP, as its name suggests. While "*AVRISP MODE*" scrolls across the display, you can connect to the ButtLoad system using the AVRStudio programmer as if it was a normal ATAVRISP and perform tasks on a connected AVR. To interact with ButtLoad in this mode, select *Tools->Program AVR->Connect...* from AVRStudio, select the AVRISP option and click OK.

When entering programming mode, ButtLoad will attempt to syncronise with the attached AVR. If this fails, "SYNC ERROR" will show on the display briefly and the AVRStudio programmer will show an error message.

To exit the function, push the joystick left. For safety reasons, the function can only be exited while not in programming mode or during the reception of a data packet from the computer.

STORE PRGM:

This mode behaves in exactly the same manner as AVRISP MODE, except "*STORAGE MODE*" will scroll across the display, and the target will be the internal memory rather than an attached AVR. You can read or write to the non-volatile memory as if it was an attached AVR, including setting the fuses, lockbytes, EEPROM and FLASH contents. To interact with ButtLoad in this mode, select *Tools->Program AVR->Connect...* from AVRStudio, select the AVRISP option and click OK.

As the different data types are programmed, they are enabled for programming automatically in PROGRAM AVR mode. **Erasing the virtual AVR will clear these flags and the stored EEPROM/Flash data but will not delete the stored fuse/lock bytes.**

In this mode the signature bytes will always read back as 0x01 0x01 0x01.

To exit the function, push the joystick left. For safety reasons, the function can only be exited while not in programming mode or during the reception of a data packet from the computer.

PROGRAM AVR:

Once data has been loaded into ButtLoad, it may be programmed into an attached AVR. This may be EEPROM, FLASH (Data), Fuse Bytes, Lock Bytes or a combination.

START:

This instructs ButtLoad to commence programming of the target in accordance with the current settings.

If the one of the selected types is not present in memory (for instance, you have elected to program the stored fuse bytes into the target AVR, but no fuse bytes have been loaded into memory) the system will show a "NO *x*" error – where *x* is the item that cannot be programmed.

As the programming commences, the screen will update to show both a progress bar along the top of the display as well as a letter indicating the currently executing task. This takes the form of a "PRG>" prefix followed by one of the following letters:

Letter	Process
C	Target Erase (memory clear)
D	Target Flash programming
E	Target EEPROM programming
F	Target Fusebytes programming
L	Target Lockbytes programming

Once the programming cycle completes sucessfully, the display will show "PROG DONE" for a short period before returning to the main menu. If an error occurred during the programming (such as a data type missing) then "PROG FAIL" will appear. This will show even if the other chosen data types were programmed sucessfully.

OPTIONS:

This section sets the items to program into the target AVR. To scroll between data types, push the joystick up or downwards. If a datatype is to be programmed it will have the letter "Y" on the right of the display, otherwise the letter will be an "N". To toggle a data type on or off, push the joystick inwards.

To save the current settings and return to the PROGRAM AVR mode menu, push the joystick left.

DATAFLASH STATS:

This function contains a submenu, with options for viewing different aspects of the data (if any) stored in ButtLoad's onboard non-volatile memory via the STORAGE MODE function.

STORAGE SIZES:

This function shows the size (in bytes) of the stored types of data. Pushing up or down in this function will scroll between the program data, EEPROM data, total fuse bytes and total lock bytes.

Stored Flash/EEPROM sizes may be slightly overstated due to the nearest-page rounding system implemented.

To return to the DATASTORE INFO submenu, push the joystick left.

VIEW DATA TAGS:

To help identify characteristics about the a program stored in the dataflash, ButtLoad can display "ButtTag" data embedded in the program as strings. To learn how to place your own custom ButtTags in your program, see the "ButtTag" section of this manual.

When entered, this mode will scan the stored program data for tags. Once encountered, the first tag data will be displayed onto the LCD. Pressing down advances to the next tag (if present) – pushing the joystick upwards has no effect. To exit this mode and return to the main menu, push the joystick left.

SETTINGS:

This function will allow you to change ButtLoad's settings. If selected, the available settings will appear in a new menu. This submenu functions in the same manner as the main menu; pushing the joystick left will exit the settings menu and return to the main menu.

SET ISP SPEED:

If you want to manually change the ISP programming frequency, use this option. Pushing the joystick up or down will cycle through the preset

speeds (shown in Hz). To save the current speed and return back to the main menu, push the joystick left.

This speed will be used for all ISP programming, including programming of target devices from the dataflash (PROGRAM AVR mode). It can also be changed by a PC when in AVRISP mode.

Some speed values are not present in the AVRStudio frontend, and so can only be selected via the ButtLoad settings menu. When read from AVRStudio, these will return the closest AVRStudio value.

RECOVERY speed is designed to correct AVRs with fuses misconfigured for an external clock as a clock source. It will output a clock on pin 9 of the Butterfly's JTAG port, which should be connected to the XTAL1 pin of the slave AVR. RECOVERY should only be used to change the target fuses; it should not be used for flash or EEPROM programming. The recovery clock is only present when outputting ISP commands.

SET RESET MODE:

By default ButtLoad tristates it's target reset pin (PF.7) so that it does not intefere with the target circuit when not programming. If you are using an external buffering of the ISP lines which expects defined logic levels at all times (or otherwise require a defined logic reset signal), change this setting to LOGIC. To use the default tristating of the reset line when inactive, choose FLOAT. To save the current reset mode and return back to the main menu, push the joystick left.

SET FIRM VERSION:

In a normal AVRISP, AVRStudio determines the firmware version of the attached device and performs an upgrade if nessesary. Because ButtLoad is custom firmware for completely different hardware, such an automated upgrade is not possible or nessesary.

You can change the firmware version returned by ButtLoad to the latest your version of AVRStudio4 supports to prevent the upgrade message from appearing each time you connect to ButtLoad.

The default firmware version returned is V2.10 (consistant with AVRStudio SP4). The version minor parameter can be changed from 0 to 20 in order to be consistant with the value expected by your version of AVRstudio.

SET SLEEP TIMEOUT:

ButtLoad will automatically go into sleep mode after a user-settable period of time of no joystick activity. This auto-sleep timer is always active **except in functions which use the serial port**.

To change the timeout duration, push the joystick up or down. To save the selected value and have it take effect immediately push the joystick left.

SET TONE VOLUME:

ButtLoad is capable of giving several audio feedback tones at selected points to give an aural indication of ButtLoad's status. This includes a startup tone, target sync success tone and several others.

In this function, you can set the tone volume from OFF to 10. To increase the volume, push the joystick upwards – similarly, a push downwards on the joystick will decrease the volume.

To exit this mode and save the current value, push the joystick left.

SET STARTUP MODE:

It may be preferable to be able to set a mode which ButtLoad will automatically enter on startup.

In this function, you can set the startup mode to NORMAL (enter main menu), PRODUCTION (enter PROGRAM AVR mode) or AVRISP (enter AVRISP mode).

To exit this mode and save the current value, push the joystick left.

CLEAR SETTINGS:

Selecting this function will show a confirmation request. To confirm the command, push the joystick right – to cancel, push the joystick left to return to the settings menu.

Confirming the command will immediately clear the internal EEPROM. This will have the effect of removing all references to the program or EEPROM data (if any) stored in dataflash, as well as reset all other non-volatile settings (such as the sleep timeout) to their defaults.

Once complete, the function will show a success message and return automatically to the main menu.

JUMP TO BOOTLOADER:

This function will re-enable the JTAG interface momentarity and, assuming the BOOTRST fuse is programmed, jump to the bootloader (if no bootloader is present ButtLoad will simply restart). The jump is executed AFTER the joystick is released; if you are programming via the JTAG interface, connect to the AVR while holding in the joystick to enter the function. While the JTAG interface is enabled, the message “*JTAG ON*” will be displayed.

The auto-sleep timeout is disabled while the JTAG interface is enabled.

SET CONTRAST:

To preserve ButtLoad battery life, or for aesthetic reasons, the ButtLoad display's contrast can be set. This value is saved into non-volatile memory and thus the contrast value will be retained on power-off. To change the contrast, push the joystick up (to darken) and down (to lighten) - the contrast range can vary from 1 (very light) to 15 (very dark). To exit the SET CONTRAST mode and save the new value into memory, push the joystick left.

SLEEP MODE:

This function will place the Butterfly in sleep mode. When activated the LCD controller and much of the peripherals will be turned of to save power. This mode can be used instead of physically removing device power but keep in mind a very small current draw will still exist.

To wake up the Butterfly and resume normal operation, push the joystick upwards.

Errors:

ButtLoad errors are prefixed with the text “E>”. If an error occurs, the error text will continuously scroll across the display until the joystick is pushed inwards. An exception to this is sync errors in AVRISP MODE or dataflash errors in PRGM DATAFLASH MODE; these will clear automatically after a sucessful retry and will not have the “E>” prefix. When an error occurs, the error tone will play and the red status LED will flash.

ButtLoad can show one of the following error messages:

BADISR: Internal error. Please send a report detailing what occurred to show this message to my email address. You will need to power-cycle your Butterfly to clear this error.

DATAFLASH ERROR: Could not recognise the onboard Dataflash. It may be damaged or corrupt.

SYNC ERROR: ButtLoad cannot syncronise its ISP communications to a connected target AVR. No AVR may be connected, the ISP cable may be attached improperly or the target AVR may be damaged.

NOTHING SELECTED: Buttload will not attempt to program a slave AVR in PROGRAM AVR mode, because no data types have been selected for programming.

NO ERASE CMD: You are trying to program a slave AVR's flash from the stored program data, but no erase command has been saved by ButtLoad. You will need to execute a flash erase while in STORAGE MODE and reload your program data.

TIMEOUT: A communication timeout occurred while waiting for the target to respond during programming in PROGRAM AVR mode. This may be caused by having the ISP SPEED setting set to RECOVERY in AVRISP mode.

RECOVERY MODE: You are trying to program a slave AVR from the stored data, but the ISP SPEED setting is set to recovery. Change the ISP speed to a normal value and retry.

NO DATA: No program flash data has been stored in memory, but you are trying to program a slave AVR's flash.

NO EEPROM: See NO DATA error description.

NO FUSE BYTES: See NO DATA error description.

NO LOCK BYTES: See NO DATA error description.

NO STORED PRGM: No program has been stored into the dataflash, and so ButtLoad cannot check for tags.

NO TAGS: No tags were found in the entire stored program data. Either no tags are present, or the tag headers are incorrect/invalid.

BUTTTAGS:

ASM Language:

The Atmel ASM2 assembly language supports a C-style preprocessor. To include ButtTags in your assembly project, add the following define to the top of your source file:

```
#define BUTTLOADTAG(text) .db "@(#)" text
```

To add a tag to your project, use the macro at the **end** of your source file.

C Language:

For ease of use, the following header file can be used in a project to allow for the easy insertion of ButtTags into a project.

ButtTag.h contents:

```
#ifndef BLTAG_H
#define BLTAG_H

    // INCLUDES:
    #include <avr/io.h>
    #include <avr/pgmspace.h>

    // STRUCTS:
    struct ButtLoadData
    {
```

```
        char MagicString[4];
        char TagData[];
};

// DEFINES:
#define BT_TAGHEADER          {'@','(','#',')'}
#define BUTTLOADTAG(id, data) const struct ButtLoadData           \
                                          BUTTTAG_##id PROGMEM = \
                                          {BT_TAGHEADER, data}

#endif
```

The ButtTag header file must be included in your project before any tags or you will receive compiler errors. The header file should be included in each file in your project which includes ButtTags.

The header file is protected against multiple-inclusion errors or compiler loops.

Placing ButtTags in your program:

To place ButtTags in your program, first you must place the above code into a new header file, and include that into your program. Once included, the new macro "BUTTLOADTAG" becomes avaliable for your use.

The syntax for the BUTTLOADTAG macro is:

BUTTLOADTAG(*Name***,** *TagDataInQuotes***)**

All ButtTags must be given a unique name, due to C language constraints. These names are not stored in the resulting binary, and thus no extra restrictions except normal C language constraints are present.

TagData can be any null-terminated string equal to or less than 20 characters in length. Tags are read out from the program binary sequentially. You may use ButtTags to store copyright information, the name of the program, its version or anything else alpha-numeric which can be displayed onto the Butterfly's LCD. If a ButtTag is over 20 characters in length, the superflous characters will be stored in the resulting binary but ignored by ButtLoad when read.

If you wish to extract a tag's textual data in your program code, it is avaliable via the syntax **BUTTTAG_**{Tag Name}**.TagData.**

TROUBLESHOOTING:

There are several things to check when you encounter trouble with ButtLoad. If, once you have tried all the below, ButtLoad is still failing, please post in the ButtLoad AVRFreaks thread (http://tinyurl.com/cco7k) or send me a private email.

Firstly, check the power source. ButtLoad **must** be powered with 3V. Attempting to power the Butterfly with voltages higher than this may damage the dataflash, rending program and data storage inoperable or problematic. Ensure that the battery (if used)

is fully charged, as a weak battery may affect programming, serial communications and/or storage performance.

Ensure that there are no shorts in the ISP line, that the ISP connector is not reversed and that all the ISP lines are fitted if you receive SYNC ERROR messages. Also ensure that the target's /RESET line can be pulled low – avoid strong pullups on the target.

Avoid placing the bare Butterfly board on conductive surfaces, and avoid touching the bare LCD pins. Damage from doing the latter is unlikely, but it will effect the LCD display's performance.

THANKS TO:

I'm very grateful to many people who helped make this project possible. First and foremost, thanks to the members of AVRFreaks.net for their wonderful input and encouragment during the project's development.

Special thanks to Barry (AVRFreaks username Bpar) for donating a significant amount of his time over the initial development period, remotley testing and debugging the code.

Also thanks to Tom (AVRFreaks username ZoomCityZoom) for the gifts of his JTAG ICE and a Butterfly, and “Smokey” Joe (www.smileymicros.com) for his donation of a pair of Butterflies for the cause after I damaged my own Butterfly. Joe sells Butterflies on his website at US$20 each and he's written a fantastic introduction to the C language – based around the Butterfly board and the free GCC compiler – which I highly recommend.

Eternal thanks to the members of the AVR-GCC project for their wonderful and free C compiler which is avaliable for everyone in a convienient toolset for Windows made by Eric of AvrFreaks.net (winavr.sourceforge.net).

My grattitude is also extended to David Bourgeois, Mike Henning, Nard Awater and Scott Coppersmith, who gratiously gave their time and input to test and debug the beta versions of ButtLoad. Scott also did a lot of extra testing work during the alpha stages which greatly eased my workload and submitted some sample schematic diagrams.

Final thanks to Sylvian for his connection diagram for ButtLoad and Collin O'Flynn for his OSCCAL calibration example code, which was used as a template for the OSCCAL calibration routine in early versions of ButtLoad. Both are also active members of AVRFreaks.net.

ButtLoad's cycle-accurate CPU busy-wait delay loops are provided by the excellent header file written by Hans-Juergen Heinrichs.

CONTACT:

I'm always happy to receive legitemate emails and take the care to answer every single one. Whether it be a bug report, feedback, AVR-related anecdote, an “attaboy!”, cohherant letter of abuse, picture of your mounted ButtLoad ISP programmer or anything else, please don't hesitate to congest the information highway.

My email address is dean_camera@hotmail.com. I am also avaliable via PM at www.AVRFreaks.net under the username “abcminiuser”.

If you wish to browse my personal webpage (listing my main past and current projects), the address is http://home.pacific.net.au/~sthelena.

ButtLoad development is currently being discussed in the ButtLoad AVRFreaks Academy thread, avaliable at http://tinyurl.com/cco7k. You can post suggestions, bug reports and the like there as an alternative to contacting me directly.

The latest version of ButtLoad is always avaliable at its AVRFreaks academy project entry, at http://tinyurl.com/z55sb.

Document icons sourced with permission from
http://www.deviantart.com/view/32215071.

ButtLoad logo based on stock photo located at http://www.deviantart.com/deviation/2264235/.