

Procedure to Upgrade Firmware in the ALS-U Bipolar Power Converter

D. Bergman

9/28/25

Version 1.0

Introduction

This document details the steps needed to upgrade the firmware in an ALS-U bipolar power converter (BPC). Firmware source code for the BPC is written in C, and the program runs on an Atmega2560-16AU microcontroller. Firmware is stored in nonvolatile FLASH memory in the Atmega chip.

To upgrade the firmware, a host computer running Arduino IDE is needed along with the new firmware .ino file. A suitable “In-Circuit Serial Programming” device (ICSP) for Arduino is also needed. This is usually done by programming an Arduino board to serve as a programmer, then selecting “Arduino as ISP” under “Tools => Programmer” in the IDE.

The controller board inside the BPC uses an SD card to configure the BPC. The file “config.txt” on the SD card is read once at startup. Some firmware upgrades may require that the contents of the config.txt file be changed, but the SD card does not contain any firmware code, only configuration data. The SD card does not need to be physically accessed to upgrade the firmware.

Firmware included with the BPCs shipped to LBNL for the AR was version 092324. Firmware beginning with version 061025 adds capability to allow for DHCP IP address assignment.

Once this new firmware is uploaded to the controller board, additional steps are needed to add a new line to the SD card’s config.txt file to select static or dynamic IP address assignment. These additional steps require the use of BNL furnished python scripts which are described below.

If any problems are encountered while performing this procedure, contact David Bergman at BNL for assistance (dbergman@bnl.gov).

Procedure

1. Install Arduino IDE on a host computer if it is not already installed.
2. Install python on the host computer if it is not already installed.
3. In the Arduino IDE Library Manager, add libraries “UIPEthernet” version 2.0.12 and “SdFat” version 2.2.0.
4. In the Arduino library directory, add OneWire library files furnished by BNL.

Note that line 336 in file DS18B20_DS2482.cpp has been commented out in order to avoid the “block till conversion complete” condition.

Library files are usually stored in the home directory in a folder named Arduino\libraries\ under both Windows and Linux systems.

5. Replace file “pins_arduino.h” with the BNL furnished version.

In windows, the file is likely to be stored in

“C:\Program Files (x86)\Arduino\hardware\arduino\avr\variants\mega\”.

In linux, file is likely to be stored in

“/home/user/Arduino/hardware/Arduino/avr/variants/mega/”.

Restart Arduino IDE after replacing the file.

6. Change DHCP_TIMEOUT macro from 60000 to 30000 in /Arduino/libraries/UIPEthernet/Dhcp.h
7. With the BPC turned off, connect the Arduino programmer to the BPC chassis front panel In-Circuit Serial Programming (ICSP) connector (3x2 header).
8. Compile firmware version 061025. Upload to the BPC using “Sketch => Upload Using Programmer” in the IDE.
9. After uploading the new firmware, disconnect the programmer, and turn on 120 VAC or 208 VAC power to the chassis.

10. Connect an ethernet cable between the BPC rear panel RJ45 connector and the host computer's ethernet jack. Configure the host's network adaptor for an IP address of 192.168.0.240.
11. Confirm that you can ping the BPC using target address 192.168.0.x where x is the serial number of the BPC chassis.
12. To save the contents of the SD card to file, run python program pullSDdib.py by typing

```
python pullSDdib.py 192.168.0.x model_serial
```

where x is the serial number of the BPC chassis and *model_serial* is the model number/serial number combination of the chassis.

Examples:

```
python pullSDdib.py 192.168.0.10 6201_0010 for slow BPC serial number 0010
```

```
python pullSDdib.py 192.168.0.20 6202_0020 for fast BPC serial number 0020
```

In the latter example, the program will write the information on the SD card to a file named "config_6202_0020.txt"

13. To update the contents of the SD card, run program pushSDdib.py by typing

```
pushSDdib.py 192.168.0.x
```

where x is the serial number of the BPC chassis.

The program responds with:

```
"Enter line number to change (1, 2, or 7):"
```

To modify the static IP address stored on the SD card, type 1, then enter the IP address as ddd.ddd.ddd.ddd

To modify the MAC address stored on the SD card, type 2, then enter the MAC address with 2 digit hex numbers as xx,xx,xx,xx,xx,xx

To select static or DHCP IP address assignment, type 7, then type enter 0 for static or 1 for DHCP.

The program does not allow you to modify lines 3, 4, 5, and 6 on the SD card which are for 1-wire sensor ID codes, Hall sensor gain adjustments, and model number/serial number respectively.

Note that the BPCs shipped to LBNL for the AR only had 6 lines in the config.txt file. Running pushSDdib.py will add a seventh line to the file for static/DHCP selection.

At the completion of these steps, the firmware and config.txt file will be upgraded, and the BPC will be able to work with either static or dynamically assigned IP addresses.