***Link Github:***

**GP\_Controller:**

<https://github.com/baobuibk/SpaceLiin_Electroporator_Controller_8HBridge_FW>

Nhánh FW\_V2

**GP\_Pulse:**

<https://github.com/baobuibk/SpaceLiin_Electroporator_8_HBridge_V1.1.0_FW>

Nhánh main

***Kết nối phần cứng:***

**Cổng giao tiếp – XBEE:**

* **Mở cổng COM baudrate 115200.**
* **Gõ chữ ‘B’ để kết nối.**
* **Tiến hành sử dụng.**

***Các lệnh để calib***

***CALIB\_RUN* N**

**NOTE**:

N: Bắt đầu quá trình calib. N = 0 hoặc 1.

**EXAMPLE**: *CALIB\_RUN* 1

**CHỨC NĂNG:** Bật tắt quá trình calib.

***CALIB\_MEASURE N S***

**NOTE**:

N: GIá trị đo được từ tụ 300V. Đơn vị: mv

S: Giá trị đo được từ tụ 50V. Đơn vị: mv

**EXAMPLE**: *CALIB\_MEASURE* 300000 50000

**CHỨC NĂNG:** Lệnh dùng để nhập giá trị đo được từ tụ hv và lv để cung cấp cho quá trình calib. Lưu ý, trong quá trình chạy calib, tất cả các lệnh khác sẽ không được chạy, để dừng calib, thực hiện lệnh *CALIB\_RUN* 0.

***Quy trình calib ADC***

**Thực hiện lệnh calib:**

***CALIB\_RUN 1***

1 chương trình calib sẽ gồm 4 lần nhập giá trị đo tụ. Sau khi chạy lệnh calib\_run, tụ sẽ tự động nạp lên giá trị đầu tiên là 50V và 5V cho tụ hv và lv tương ứng. Chờ cho đến khi áp tụ ổn định, nhập lệnh với giá trị đọc được từ VOM:

*CALIB\_MEASURE* 50000 5000

Giá trị gửi cho *CALIB\_MEASURE* sẽ là giá trị đo được từ 2 đầu tụ 300V và 50V. Các giá trị ghi vào sẽ được viết theo đơn vị mv.

Sau calib lần 1, lần 2 sẽ nạp lên 100V và 10V, lần 3 sẽ nạp lên 200V và 25V, lần 4 sẽ nạp lên 300V và 50V. Tại mỗi lần calib, chương trình sẽ ghi ra màn hình “CALIB HV CAP AT 50V AND LV CAP AT 5V”, người dùng có thể dựa vào đó để xác định mình đang ở lần calib nào. Ví dụ, các lệnh bôi đen là lệnh người dùng phải thực hiện:

***CALIB\_RUN* 1**

CALIB HV CAP AT 50V AND LV CAP AT 5V

***CALIB\_MEASURE* 49000 4900**

CALIB HV CAP AT 100V AND LV CAP AT 10V

***CALIB\_MEASURE* 99000 10000**

CALIB HV CAP AT 200V AND LV CAP AT 25V

***CALIB\_MEASURE* 199253 24978**

CALIB HV CAP AT 300V AND LV CAP AT 50V

***CALIB\_MEASURE* 298845 49875**

FINNISH RUNNING CALIB, CAP IS DISCHARGING

Sau khi calib xong, tụ sẽ tự động được xả. Trong quá trình calib, nếu muốn thoát ra thực hiện lệnh:

*CALIB\_SET* 0

**Các lệnh để điều khiển tụ**

***SET\_CAP\_VOLT N S : Set cap voltage***

**NOTE**:

N: Set điện áp cho tụ 300V

S: Set điện áp cho tụ 50V

**RANGE:**

1 <= N <= 300

1 <= S <= 50

**EXAMPLE**: *SET\_CAP\_VOLT* 50 50

**CHỨC NĂNG:** Cài đặt điện áp muốn nạp cho tụ 300 và tụ 50.

***SET\_CAP\_CONTROL N S : Control charger on/off***

**NOTE**:

N = 0: Tắt nạp tụ HV, N = 1: Bật nạp tụ HV

S = 0: Tắt nạp tụ LV, S = 1: Tắt nạp tụ LV

**EXAMPLE**: *SET\_CAP\_CONTROL* 1 1

**CHỨC NĂNG:** Bật tắt nạp tụ HV và LV.

***SET\_CAP\_RELEASE N  S : Control releasing cap***

**NOTE**:

N: Chỉ có 2 giá trị 1 và 0. 1 là tụ 300V xả, 0 là tụ 300V không được xả.

S: Chỉ có 2 giá trị 1 và 0. 1 là tụ 50V xả, 0 là tụ 50V không được xả.

**EXAMPLE**: *SET\_CAP\_RELEASE* 1 1

**CHỨC NĂNG:** Bật tắt xả tụ.

***GET\_CAP\_VOLT : Get set voltage on cap***

**EXAMPLE**: *GET\_CAP\_VOLT*

**CHỨC NĂNG:** Lấy thông tin về giá trị điện áp được set cho 2 tụ.

***GET\_CAP\_CONTROL : Get info whether cap is charging or not***

**EXAMPLE**: G*ET\_CAP\_CONTROL*

**CHỨC NĂNG:** Lấy thông tin tụ có đang nạp hay không.

***GET\_CAP\_RELEASE : Get info whether cap is releasing or not***

**EXAMPLE**: *GET\_CAP\_RELEASE*

**CHỨC NĂNG:** Lấy thông tin tụ có đang xả hay không

***GET\_CAP\_ALL : Get all info about cap***

**EXAMPLE**: *GET\_CAP\_ALL*

**CHỨC NĂNG:** Lấy tất cả thông tin từ tụ, điện áp set trên tụ, tụ có đang nạp hay xả.

***Các lệnh để điều khiển xung cầu H***

***SET\_SEQUENCE\_INDEX N: Set current sequence index***

**NOTE**:

N: Set số thứ tự của sequence hiện tại

**RANGE:**

1 <= N <= 10

**EXAMPLE**: *SET\_SEQUENCE\_INDEX 1*

**CHỨC NĂNG:** Chọn thứ tự của sequence hiện tại

***SET\_SEQUENCE\_DELETE: Delete current sequence index***

**EXAMPLE**: *SET\_SEQUENCE\_DELETE*

**CHỨC NĂNG:** Xóa sequence được chọn hiện tại.

**NOTE:** FW version hiện tại chỉ cho phép xóa tại index hiện tại lớn nhất, vd: user set tổng cộng 5 sequence thì lệnh này chỉ cho phép xóa tại index thứ 5, số sequence tổng cộng mới sẽ là 4 sequence, nếu user xóa sequence tại index 2 hay 3 thì sẽ bị báo lỗi.

***SET\_SEQUENCE\_CONFIRM: Confirm current sequence index***

**EXAMPLE**: *SET\_SEQUENCE\_CONFIRM*

**CHỨC NĂNG:** Xác nhận các thông số đã set cho sequence đang được chọn.

**NOTE:** Tại mỗi một index mới tạo, sau khi user đã nhập xong hết tất cả các thông số, thì phải thêm một bước xác nhận thông số trước khi user có thể chuyển index qua số khác, điều này để đảm bảo người dùng chỉ set sequence cần thiết, tránh các lỗi lúc đánh lệnh

***SET\_SEQUENCE\_DELAY N: Set delay between sequence***

**NOTE**:

N: Số thời gian delay (ms)

**RANGE:**

1 <= N<= 100

**EXAMPLE**: *SET\_SEQUENCE\_DELAY 15*

**CHỨC NĂNG:** Set thời gian chờ giữa các sequence.

***SET\_PULSE\_POLE N M : Set pos and neg pole of a pulse***

**NOTE**:

N: Set kênh cực dương

M: kênh cực âm

**RANGE:**

1 <= N, M <= 8; N != M

**EXAMPLE**: *SET\_PULSE\_POLE 2 8*

**CHỨC NĂNG:** Chọn cực đầu ra, 2 -> cực dương, 8-> cực âm

***SET\_PULSE\_COUNT N M X Y : Set number of pulse***

**NOTE**:

N: Set số xung dương cho kênh điện áp tụ 300V

M: Set số xung âm cho kênh điện áp tụ 300V

X: Set số xung dương cho kênh điện áp tụ 50v

Y: Set số xung âm cho kênh điện áp tụ 50V

**RANGE:**

1 <= N, M, X, Y <= 20

**EXAMPLE**: *SET\_PULSE\_COUNT* 2 2 10 10

**CHỨC NĂNG:** Set số xung dương và xung âm cho kênh điện áp tụ 300V và tụ 50V. Nếu set N = 10, thì sẽ phát ra 10 xung dương và 10 xung âm, tổng cộng có 20 xung.

***SET\_PULSE\_DELAY X Y N : Set delay between pulse hv and lv***

**NOTE**:

X: Set khoảng thời gian delay giữa xung dương và âm của kênh HV

Y: Set khoảng thời gian delay giữa xung dương và âm của kênh LV.

N: Set khoảng thời gian delay giữa xung HV và xung LV, đơn vị là ms.

**RANGE:**

1 <= X, Y <= 100

1 <= N <= 1000

**EXAMPLE**: *SET\_PULSE\_DELAY* 100 50 1000

**CHỨC NĂNG:** Set khoảng thời gian delay giữa xung HV và xung LV, delay giữa xung dương và âm của HV và của LV, trong khoảng thời gian chờ đó, áp tại cầu H = 0V. Đơn vị: ms. Khoảng thời gian chờ sẽ cộng nối tiếp với thời gian off time của xung tương ứng.

***SET\_PULSE\_HV N S : Set hs pulse on time and off time***

**NOTE**:

N: Set thời gian on time cho 1 xung kênh hv. Đơn vị: ms.

S: Set thời gian off time cho 1 xung kênh hv. Đơn vị: ms.

**RANGE:**

1 <= N <= 20

1 <= S <= 20

**EXAMPLE**: *SET\_PULSE\_HV* 10 10

**CHỨC NĂNG:** Set số on time và off time cho 1 xung kênh hv. Đơn vị: ms.

***SET\_PULSE\_LV N S : Set ls pulse on time and off time***

**NOTE**:

N: Set thời gian on time cho 1 xung kênh lv. Đơn vị: ms.

S: Set thời gian off time cho 1 xung kênh lv. Đơn vị: ms.

**RANGE:**

1 <= N <= 1000

1 <= S <= 1000

**EXAMPLE**: *SET\_PULSE\_LV* 10 10

**CHỨC NĂNG:** Set số on time và off time cho 1 xung kênh lv. Đơn vị: ms.

***SET\_PULSE\_CONTROL N : Start pulsing***

**NOTE**:

N = 0: Tắt phát xung cầu H

N = 1: Bật phát xung cầu H

**EXAMPLE**: *SET\_PULSE\_CONTROL* 1

**CHỨC NĂNG:** Bật tắt phát xung cầu H.

***GET\_SEQUENCE\_INDEX : Get current index of the sequence***

**EXAMPLE**: *GET\_SEQUENCE\_INDEX*

**CHỨC NĂNG:** Lấy thông tin về số thự tự hiện tại của sequence.

***GET\_SEQUENCE\_DELAY: Get the delay of the current sequence.***

**EXAMPLE**: *GET\_SEQUENCE\_DELAY*

**CHỨC NĂNG:** Lấy thông tin về thời gian chờ giữa 2 sequence của sequence hiện tại.

***GET\_SEQUENCE\_ALL: Get all info about of the total sequence that the user set***

**EXAMPLE**: *GET\_SEQUENCE\_ALL*

**CHỨC NĂNG:** Lấy tất cả thông tin về sequence kể cả xung trong sequence đó mà người dùng đã set, lệnh này sẽ xuất ra tất cả sequence mà user đã set. Để chỉ lấy thông tin về số xung của sequence hiện tại, vui lòng sử dụng GET\_PULSE\_ALL.

***GET\_PULSE\_POLE : Get the pos and neg pole of a pulse***

**EXAMPLE**: *GET\_PULSE\_POLE*

**CHỨC NĂNG:** Lấy thông tin về cực xung dương và âm.

***GET\_PULSE\_COUNT : Get number of pulse***

**EXAMPLE**: *GET\_PULSE\_COUNT*

**CHỨC NĂNG:** Lấy thông tin về số xung dương và âm của tụ HV và tụ LV.

***GET\_PULSE\_DELAY : Get delay between pulse hv and lv***

**EXAMPLE**: *GET\_PULSE\_DELAY*

**CHỨC NĂNG:** Lấy thông tin về khoảng thời gian chờ giữa xung dương và âm của tụ HV, tụ LV, và giữa 2 xung của 2 tụ.

***GET\_PULSE\_HV : Get hs pulse on time and off time***

**EXAMPLE**: *GET\_PULSE\_HV*

**CHỨC NĂNG:** Lấy thông tin về số thời gian on và off (ms) của xung tụ HV.

***GET\_PULSE\_LV : Get ls pulse on time and off time***

**EXAMPLE**: *GET\_PULSE\_LV*

**CHỨC NĂNG:** Lấy thông tin về số thời gian on và off (ms) của xung tụ LV.

***GET\_PULSE\_CONTROL : Get info whether pulse starting pulsing***

**EXAMPLE**: *GET\_PULSE\_CONTROL*

**CHỨC NĂNG:** Lấy thông tin xem cầu H có đang băm xung hay không.

***GET\_PULSE\_ALL : Get all info about pulse***

**EXAMPLE**: *GET\_PULSE\_ALL*

**CHỨC NĂNG:** Lấy tất cả thông tin của cầu H.

***Các lệnh để lấy dữ liệu quá trình chạy***

***MEASURE\_VOLT : Measure cap voltage***

**EXAMPLE**: *MEASURE\_VOLT*

**CHỨC NĂNG:** Lấy điện áp hiện tại trên tụ.

***MEASURE\_CURRENT : Measure cuvette current***

**EXAMPLE**: *MEASURE\_CURRENT*

**CHỨC NĂNG:** Lấy dòng điện hiện tại.

***MEASURE\_IMPEDANCE X Y Z: Measure cuvette impedance***

**NOTE**:

X: Điện áp để đo trở kháng.

Y: Thời gian đo trở kháng (đơn vị: ms)

Z: Biến yêu cầu xả áp sau khi đo xong

**RANGE:**

1 <= X <= 300

0 <= Y <= 2000

0 <= Z <= 1

**EXAMPLE**: *IMPEDANCE\_MEASURE* 50 1000 1

**CHỨC NĂNG:** Khởi động quy trình đo trở kháng. Sau khi đo xong tụ sẽ xả nếu như người dùng cho phép xả, nếu không thì không xả tụ.

***GET\_SENSOR\_GYRO : Get gyro***

**EXAMPLE**: *GET\_SENSOR\_GYRO*

**CHỨC NĂNG:** Lấy thông tin cảm biến.

***GET\_SENSOR\_ACCEL : Get accel***

**EXAMPLE**: *GET\_SENSOR\_ACCEL*

**CHỨC NĂNG:** Lấy thông tin cảm biến.

***GET\_SENSOR\_LSM6DSOX : Get accel and gyro***

**EXAMPLE**: *GET\_SENSOR\_LSM6DSOX*

**CHỨC NĂNG:** Lấy thông tin cảm biến.

***GET\_SENSOR\_TEMP : Get temp***

**EXAMPLE**: *GET\_SENSOR\_TEMP*

**CHỨC NĂNG:** Lấy thông tin cảm biến nhiệt độ.

***GET\_SENSOR\_PRESSURE : Get pressure***

**EXAMPLE**: *GET\_SENSOR\_PRESSURE*

**CHỨC NĂNG:** Lấy thông tin cảm biến áp suất.

***GET\_SENSOR\_BMP390 : Get temp, pressure and altitude***

**EXAMPLE**: *GET\_SENSOR\_BMP390*

**CHỨC NĂNG:** Lấy thông tin cảm biến.

***Các lệnh để set thông số accel cho xung cầu H chạy***

***SET\_THRESHOLD\_ACCEL X Y Z: Set threshold accel***

**NOTE**:

X: Set threshold accel cho trục X. Đơn vị: mG.

Y: Set threshold accel cho trục Y. Đơn vị: mG.

Z: Set threshold accel cho trục Z. Đơn vị: mG.

**UNIT:**

Đơn vị cho X, Y, Z ở trên là mG, với G là *gravitational constant G*,

**RANGE:**

1 <= X <= 30000

1 <= Y <= 30000

1 <= Z <= 30000

**EXAMPLE**: *SET\_THRESHOLD\_ACCEL 1000 1000 2500*

**CHỨC NĂNG:** Set giá trị accel threshold cho 3 thông số X, Y, Z; Ở FW version hiện tại, chương trình sẽ sử dụng thông số threshold Z, còn 2 thông số X, Y sẽ bỏ qua.

***GET\_THRESHOLD\_ACCEL:***

**EXAMPLE**: G*ET\_THRESHOLD\_ACCEL*

**CHỨC NĂNG:** Lấy các thông số threshold X, Y, Z đã set

***SET\_AUTO\_ACCEL N:***

**NOTE**:

N = 0: Tắt phát xung cầu H khi chạy ở chế độ tự động dựa trên accel.

N = 1: Bật phát xung cầu H khi chạy ở chế độ tự động dựa trên accel.

**EXAMPLE**: *SET\_AUTO\_ACCEL 1*

**CHỨC NĂNG:** Cho phép chương trình chạy ở chế độ tự động dựa trên thông số accel threshold đã set từ trước, khi giá trị accel đạt hoặc vượt ngưỡng giá trị threshold, xung cầu H sẽ được tự động chạy, trước đó nếu accel chưa đạt threshold, thì xung cầu H sẽ chưa được chạy.

**Note về FW version này**

**Thông báo đã nạp tụ xong và đã xả tụ xong:**

Ở version này, sau khi tụ đã nạp tới điện áp đã set thì trên console sẽ hiện lên thông báo tụ đã nạp xong tới mức điện áp mà mình đã set. Tương tự cho việc tụ đã xả xong.

**QUY TRÌNH CÀI ĐẶT THRESHOLD ACCEL:**

* Sau khi đã cài đặt xong sequence, và cap volt như trước.
* Tại đây user có 2 lựa chọn, 1 là sử dụng lệnh SET\_PULSE\_CONTROL 1, 2 là sử dụng lệnh SET\_AUTO\_ACCEL 1.
* Để cầu H tự động chạy bằng accel threshold, người dùng đầu tiên phải nhập lệnh SET\_THRESHOLD\_ACCEL 1000 2000 2500. Quan trọng là thông số Z, thông số Z sẽ là thông số chương trình dùng để thực hiện việc chạy cầu H.
* Sau khi đã set xong threshold accel thì người dùng nhập lệnh SET\_AUTO\_ACCEL 1 và chương trình sẽ bắt đầu Stream dữ liệu accel vào console, cho đến khi đạt accel threshold thì xung sẽ chạy.

**ĐỂ THOÁT KHỎI STREAM ACCEL:**

* Khi dữ liệu accel đang stream trên console, để tắt stream, user nhấn nút bất kỳ thì stream accel sẽ tự động tắt.

**Quy trình chạy**

* **SET\_CAP\_VOLT 100 50 -> Cài đặt điện áp**
* **SET\_CAP\_CONTROL 1 1 -> Sạc tụ lên điện áp vừa cài đặt**
* **SET\_SEQUENCE\_INDEX 1 -> Cài đặt thứ tự sequence hiện tại**
* **SET\_SEQUENCE\_DELAY 5 -> Cài đặt TG delay giữa 2 sequence**
* **SET\_PULSE\_POLE 3 8 -> Chọn cực đầu ra**
* **SET\_PULSE\_COUNT 5 6 7 8 -> Chọn số xung**
* **SET\_PULSE\_HV 5 15 -> Cài đặt thời gian ON – OFF của xung HV**
* **SET\_PULSE\_LV 20 30 -> Cài đặt thời gian ON – OFF của xung LV**
* **SET\_PULSE\_DELAY 5 6 7 - > Cài đặt thời gian delay giữa các xung.**
* **SET\_SEQUENCE\_CONFIRM -> Xác nhận thông số đã cài đặt.**
* **SET\_SEQUENCE\_INDEX 2 -> Cài đặt thứ tự sequence hiện tại**
* **SET\_SEQUENCE\_DELAY 15 -> Cài đặt TG delay giữa 2 sequence**
* **SET\_PULSE\_POLE 1 7 -> Chọn cực đầu ra**
* **SET\_PULSE\_COUNT 15 20 13 8 -> Chọn số xung**
* **SET\_PULSE\_HV 16 20 -> Cài đặt thời gian ON – OFF của xung HV**
* **SET\_PULSE\_LV 100 200 -> Cài đặt thời gian ON – OFF của xung LV**
* **SET\_PULSE\_DELAY 20 100 200 - > Cài đặt thời gian delay giữa các xung.**
* **SET\_SEQUENCE\_CONFIRM -> Xác nhận thông số đã cài đặt.**
* Nếu xài accel threshold:
* **SET\_THRESHOLD\_ACCEL 100 200 2500 -> Cài đặt thông số threshold accel**
* **SET\_AUTO\_ACCEL 1 -> Cho phép chương trình chạy tự động dựa trên thông số accel threshold Z đã set**
* Nếu xài bình thường:
* **SET\_PULSE\_CONTROL 1 -> Cho phép xung cầu H chạy.**