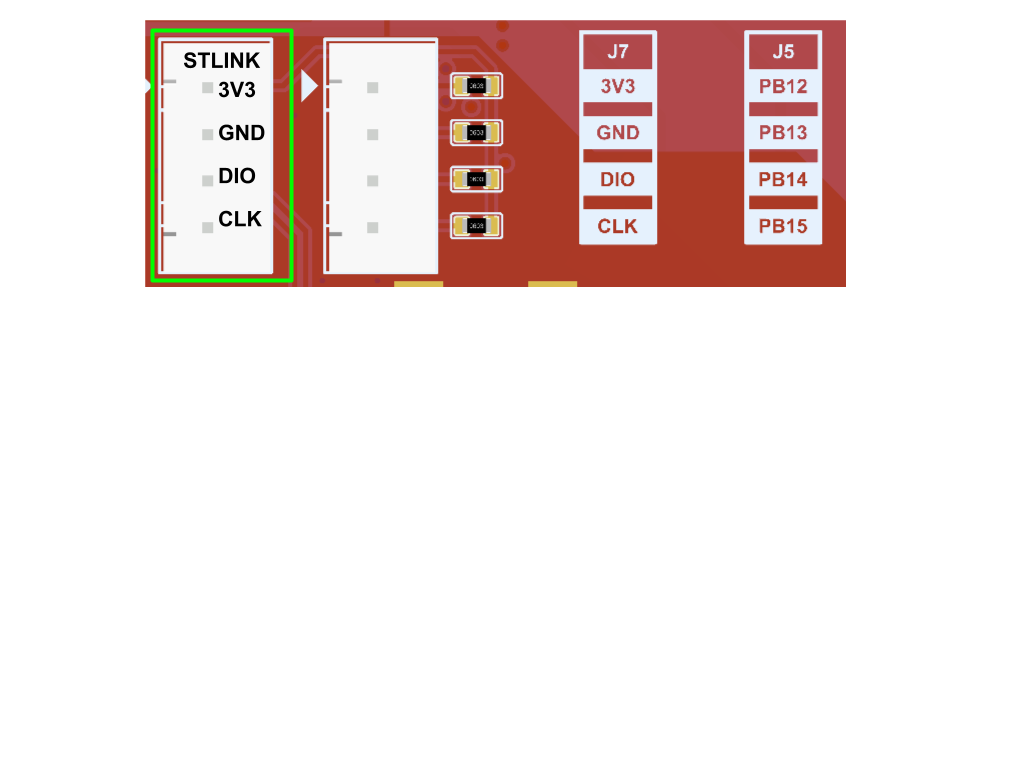
***Github Link***

Controller: [GP\_Controller.bin](https://github.com/baobuibk/SpaceLiin_Electroporator_V2.0_Controller/tree/main/Debug)

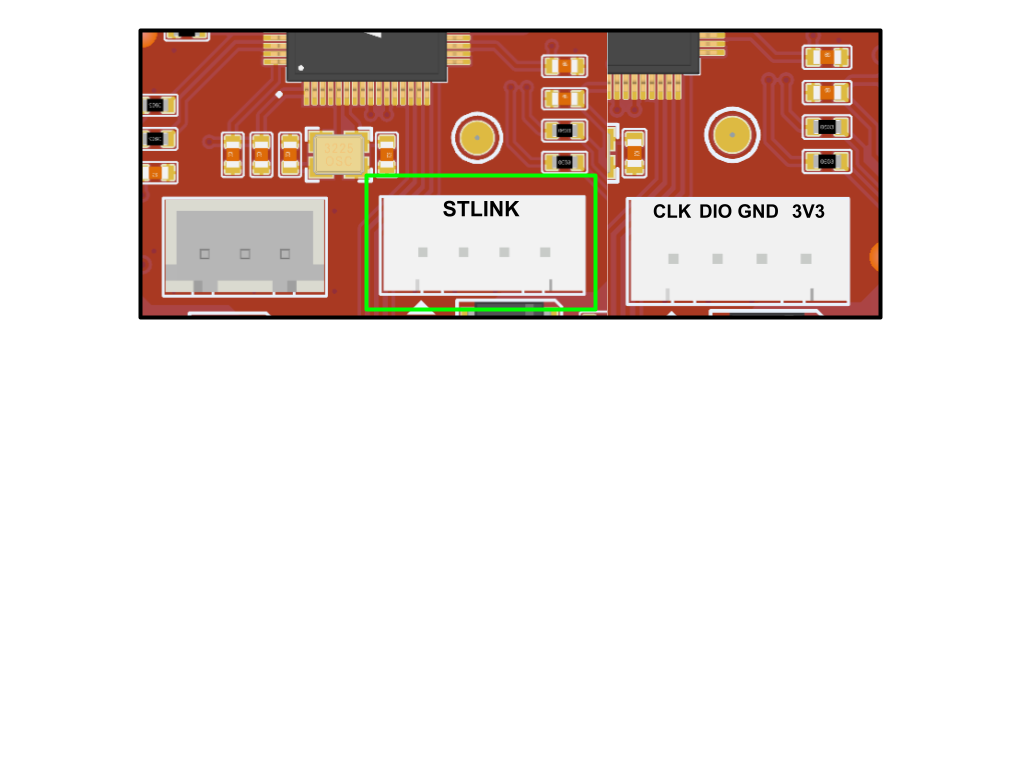
Switch: [GP\_Switch](https://github.com/baobuibk/SpaceLiin_Electroporator_V2.0_Switch/tree/main/Debug)

***Kết nối STLINK***

Controller:



Switch:

****

***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 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***

***Cài đặt sequence cho 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: 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

***Cài đặt thông số cho xung cầu H***

***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:**

0 <= X, Y <= 100

0 <= 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\_POS N S : Set hs pos 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:**

5 <= N <= 20

5 <= S <= 20

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

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

***SET\_PULSE\_HV\_NEG N S : Set hs neg 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:**

5 <= N <= 20

5 <= S <= 20

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

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

***SET\_PULSE\_LV\_POS N S : Set ls pos 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 <= 90

1 <= S <= 90

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

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

***SET\_PULSE\_LV\_NEG N S : Set ls neg 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 <= 90

1 <= S <= 90

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

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

***Các lệnh bắt đầu quá trình băm xung***

***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, đây là chế độ bình thường, sau khi nhập lệnh xong thì bo sẽ chạy ngay lập tức.

***SET\_THRESHOLD\_ACCEL Z :***

**NOTE**:

Z: Giới hạn gia tốc trục Z

**RANGE:**

1 <= Z <= 2000 mg

**EXAMPLE**: *SET\_THRESHOLD\_ACCEL 2000*

**CHỨC NĂNG:** Cài đặt giới hạn gia tốc cho chế độ băm xung theo giới hạn gia tốc, firmware ver này chỉ xét theo trục Z. Đơn vị g là đơn vị của trọng trường Trái Đất.

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

**NOTE**:

N = 0: Tắt phát xung cầu H theo giới hạn gia tốc

N = 1: Bật phát xung cầu H theo giới hạn gia tốc

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

**CHỨC NĂNG:** Bật chế độ băm xung cầu H theo giới hạn của gia tốc. Sau khi nhập lệnh xong, bo sẽ luôn so sánh giá trị hiện tại với giới hạn gia tốc đã cài đặt, nếu giá trị hiện tại lớn hơn giá trị đã cài đặt thì cho xung cầu H chạy.

***Lấy giá trị đã cài đặt cho sequence***

***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.

***Lấy giá trị đã cài đặt cho xung cầu H***

***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\_POS : Get hs pos pulse on time and off time***

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

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

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

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

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

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

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

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

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

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

**CHỨC NĂNG:** Lấy thông tin về số thời gian on và off (ms) của xung âm 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\_IMPEDANCE N M: Measure cuvette impedance***

**NOTE**:

N: Kênh cực dương.

S: Kênh cực âm.

**EXAMPLE**: *IMPEDANCE\_MEASURE* 1 8

**CHỨC NĂNG:** Khởi động quy trình đo trở kháng. Sẽ bắt đầu quá trình đo tụ tự động, áp tụ sẽ nhảy từ từ theo từng mức để đo các mức điện trở khác nhau, sau khi đo xong thì tụ sẽ KHÔNG tự động xả.

***Các lệnh để lấy dữ liệu cảm biến***

***Cảm biến LSM6DSOX***

***GET\_SENSOR\_GYRO : Get gyro sensor data from LSM6DSOX***

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

***GET\_SENSOR\_ACCEL : Get accel sensor data from LSM6DSOX***

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

***GET\_SENSOR\_LSM6DSOX : Get all sensor data from LSM6DSOX***

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

***Cảm biến BMP390***

***GET\_SENSOR\_TEMP : Get temperature sensor data from BMP390***

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

***GET\_SENSOR\_PRESSURE : Get pressure sensor data from BMP390***

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

***GET\_SENSOR\_BMP390 : Get all sensor data from BMP390***

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

**Cảm biến H3LIS**

***GET\_SENSOR\_H3LIS : Measure cap voltage***

**EXAMPLE**: *GET\_SENSOR\_H3LIS*

**Note về FW version này**

* Lệnh đo trở kháng bây giờ chỉ cần nhập giá trị 2 cực dùng để đo giá trị điện trở:
  + Quá trình đo trở sẽ bắt đầu từ áp 30V và tăng dần cho đến đo được giá trị điện trở, nếu như giá trị điện trở vượt ngưỡng toàn thang đo thì kết quả cuối cùng sẽ trả về giá trị gần đúng nhất sau khi đã đo tại áp 300V.
  + Sau khi đo xong thì điện áp trên tụ sẽ không được xả.
  + Bảng giá trị điện áp được dùng để đo theo thứ tự sau:
    - 30V
    - 50V
    - 100V
    - 200V
    - 300V
* Ton và Toff tối đa cho bên tụ LV sẽ tạm thời giảm xuống từ 1000ms còn 90ms cho ver này do vấn đề hiệu suất của máy biến áp bên tụ LV. Tại 90ms thì công suất băm xung cầu H sẽ được lái bởi tụ LV – thay vì máy biến áp như kế hoạch ban đầu và sau mỗi 90ms thì giá trị điện áp sẽ rớt tối đa 10%.
* Ton và Toff tối thiểu cho cả tụ HV và LV sẽ tạm thời tăng lên từ 1ms đến 5ms.
* Cách xài SET\_SEQUENCE\_DELETE và SET\_SEQUENCE\_INDEX:
  + SET\_SEQUENCE\_INDEX cho phép người dùng nhảy qua lại giữa các sequence khác nhau, tuy nhiên, nếu như người dùng muốn tạo một sequence mới thì người dùng chỉ được phép nhảy sang thứ tự sequence tiếp theo. Ví dụ:
    - Số sequece index đã được confirm hiện tại là 2.
    - Khi nhảy sang một index mới thì người dùng chỉ được phép nhảy đến số 3, không được phép nhảy cóc tới số 4 hoặc 5.
    - Người dùng được phép quay về index cũ mà mình đã SET\_SEQUENCE\_CONFIRM.
  + Đối với SET\_SEQUENCE\_DELETE:
    - Số sequece index đã được confirm hiện tại là 4.
    - Để thực hiện việc delete, người dùng phải nhảy đến thứ tự sequence lớn nhất là số 4 và thực hiện hàm delete, nếu không sẽ bị báo lỗi.
* Trước khi khởi động bo người dùng nên gắn đầy đủ 2 cảm biến nối dây: BMP390, LSM6DSOX theo thứ tự bất kỳ.

**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.**
* **SET\_PULSE\_CONTROL 1 -> Bắt đầu chạy.**