

HYDA CONTROLLER

HƯỚNG DẪN SỬ DỤNG PHẦN MỀM

1. THÔNG TIN CHUNG

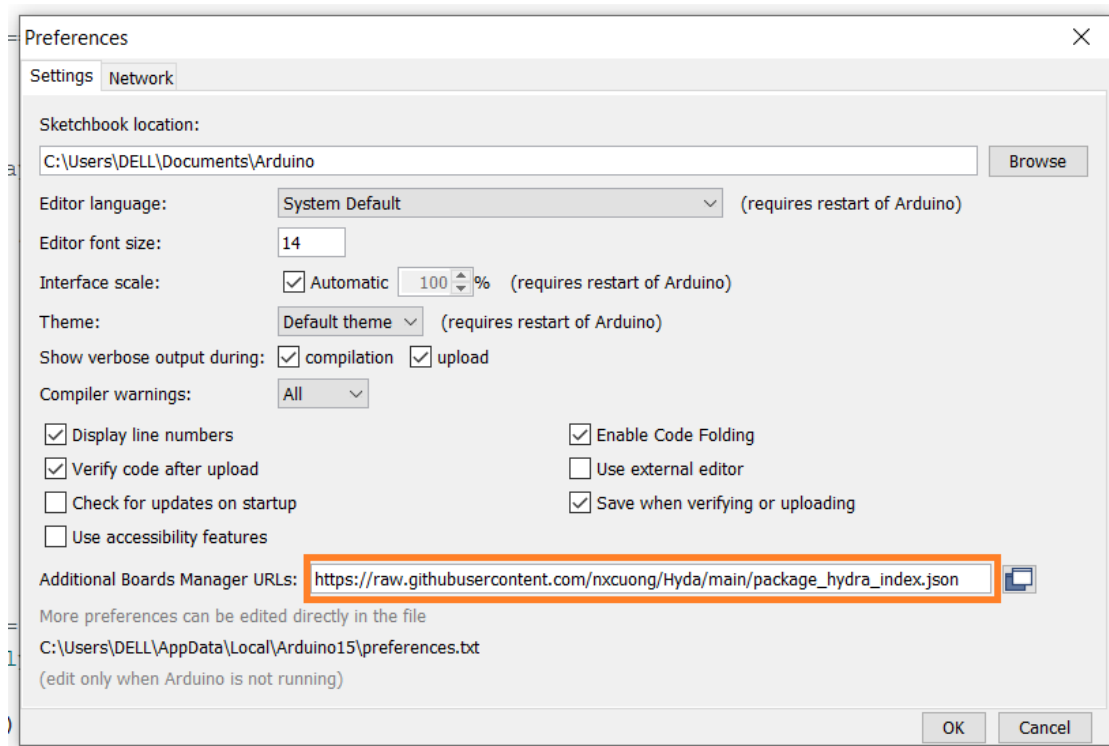
Phần mềm

Bộ mạch Hyda sử dụng **arduino IDE** lập trình bằng cách tích hợp Hyda vào arduinoIDE.

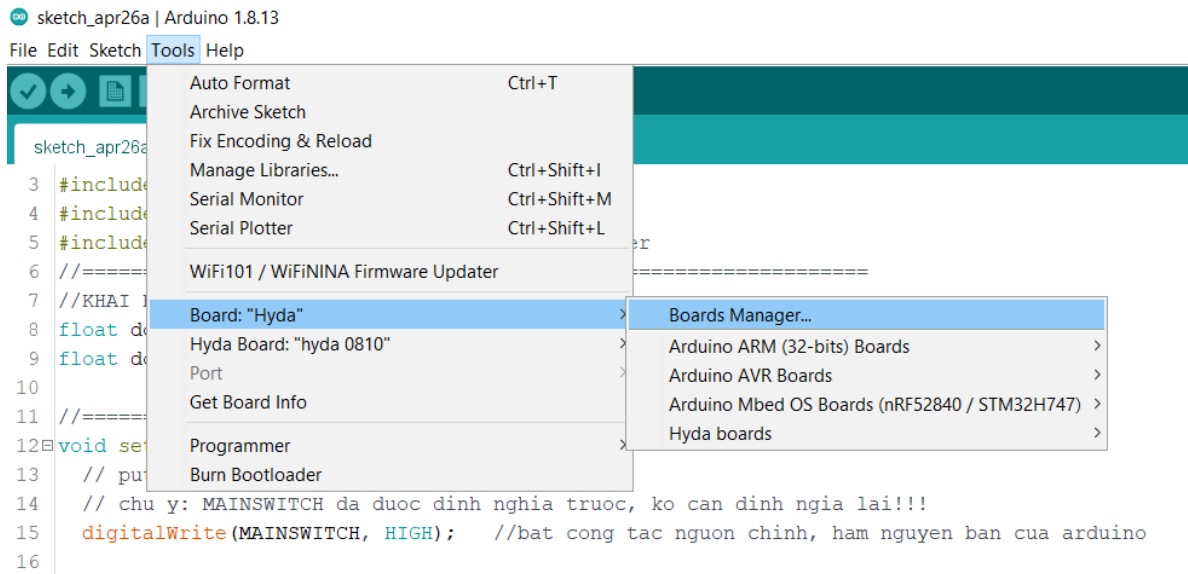
- Bước 1. Tải và cài đặt arduino IDE <https://www.arduino.cc/en/software>
- Bước 2: Mở chương trình arduino IDE, tại cửa sổ File/Preferences
- Copy và paste đường link vào ô được đánh dấu:

https://raw.githubusercontent.com/nxcuong/Hyda/main/package_hydra_index.json

- Nhấn OK sau khi paste xong.

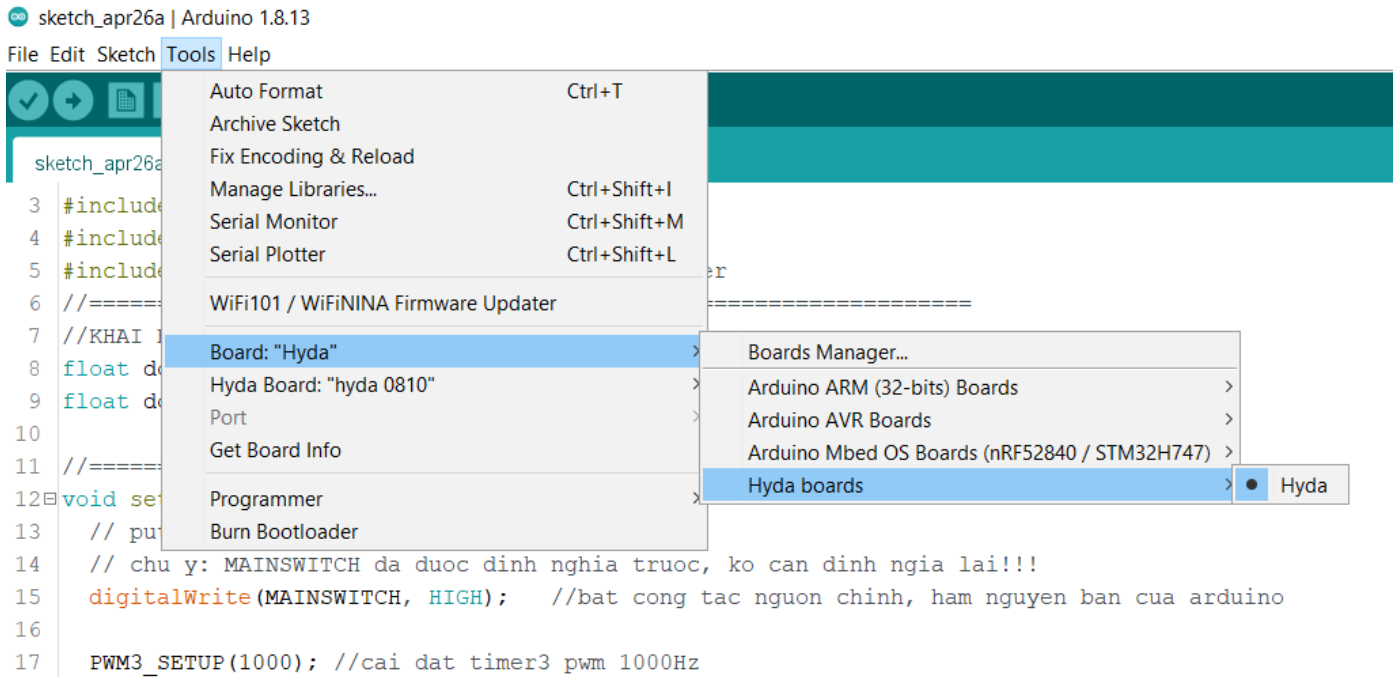


- Bước 3: Vào Tools/Boards Manager



- Kéo con trượt xuống dưới cùng tìm tới phần: Hyda family, kích vào cài đặt – install
- Nhấn Close khi quá trình cài đặt kết thúc.
- Chọn bo mạch để lập trình: *Tools/Boar/Hyda boards/hyda*
- *Tools/ Hyda Board/hyda0810*

Phần chọn bo mạch lập trình đã hoàn tất.



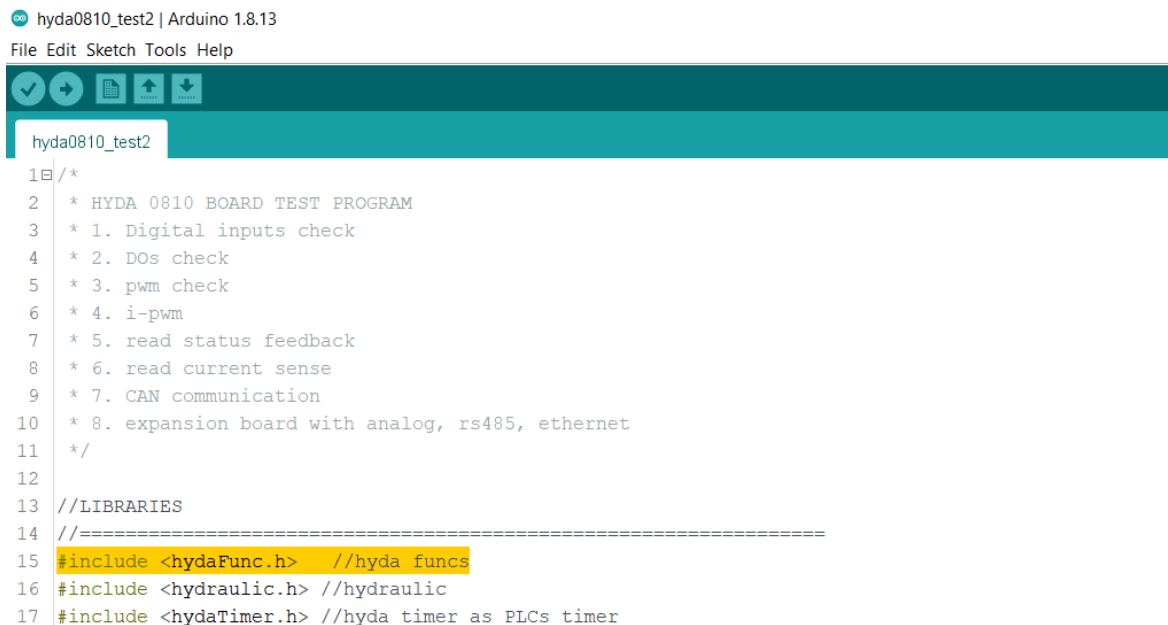
2. HƯỚNG DẪN LẬP TRÌNH

Để lập trình với bo mạch Hyda, chúng ta cần sử dụng thư viện viết riêng cho mạch.

Thư viện sử dụng cho dòng sản phẩm Hyda là "hydaFunc.h"

Trong chương trình Arduino IDE ta cần thêm dòng lệnh

#include <hydaFunc.h>



Thư viện cho phép sử dụng các hàm viết sẵn cho dòng sản phẩm HYDA để có thể đọc tín hiệu kênh vào, ghi đầu ra số, kích PWM, hay đọc kênh analog,...

a. ĐỌC KÊNH VÀO SỐ

Để đọc tín hiệu kênh vào số ta sử dụng hàm sau: **DIN(kênh)**, kênh từ 1 tới 8.

Khai báo trong chương trình như sau:

Ví dụ:

```
int kenhVao1;           // khai bao bien kenhVao1
kenhVao1 = DIN(1);      //đọc kênh vào 1 lưu vào biến kenhVao1
```

b. LÀM VIỆC VỚI KÊNH RA

HYDA0810

Có 10 kênh đầu ra, đều có thể cấu hình ra ON/OFF (logic) với các van logic hoặc cấu hình ra PWM để điều khiển các van tỷ lệ.

Tần số PWM cho các kênh từ 1 tới 8 sẽ giới hạn ở mức 250Hz, không sử dụng tần số cao hơn cho các kênh này!

Riêng kênh DO9 và DO10 có thể sử dụng để băm xung ở tần số trên 1kHz tới 10kHz và có mạch đo dòng hồi tiếp chạy thuật toán PID.

Lưu ý: Chỉ sử dụng 1 cấu hình cho đầu ra nhất định, nếu đã sử dụng logic sẽ không thể dùng PWM và ngược lại.

- HÀM DOUT

Cấu trúc: DOUT(kênh, giá trị ghi); //hàm ghi giá trị kênh ra

Trong đó kênh: từ 1 tới 10

Giá trị là logic 0 hoặc 1.

Ví dụ: đọc đầu vào 1, ghi giá trị đầu vào 1 ra kênh đầu ra 6

(mô hình LADDER)

```
|-----| DI1|------(DQ6)-----|
```

Dùng thư viện hyda: **DOUT(6, DIN(1));** //ghi ra kênh DO6 giá trị DI1

- HÀM PWM

Trước khi sử dụng hàm PWM cần cấu hình tần số của kênh được sử dụng trong hàm SETUP của chương trình.

Hyda0810 sử dụng 4 timer để tạo PWM là timer1, timer3, timer4 và timer5.

Timer1 cho các kênh DO1, DO3, DO5

Timer3 cho các kênh DO2, DO4, DO6

Timer4 cho các kênh DO7, DO8

Timer5 cho các kênh DO9, DO10

Lưu ý: Chỉ có thể cài đặt 1 tần số cho mỗi timer, do đó các kênh cùng nhóm sẽ có cùng tần số PWM.

Để cài đặt tần số PWM ta sử dụng hàm **PWMn_SETUP(unsigned interger Tanso);**

Với n = 1, 3, 4, 5 tương ứng từng Timer

Tanso: là giá trị tần số PWM.

Ví dụ: đặt tần số 1000Hz cho timer3 ta khai báo như sau trong phần setup:

sketch_apr26a | Arduino 1.8.13

File Edit Sketch Tools Help

```

sketch_apr26a $
1 //LIBRARIES
2 //=====
3 #include <hydaFunc.h> //hyda funcs
4 #include <hydraulic.h> //hydraulic
5 #include <hydaTimer.h> //hyda timer as PLCs timer
6 //=====
7
8 void setup() {
9     // put your setup code here, to run once:
10    PWM3_SETUP(1000); //cai dat timer3 pwm 1000Hz
11 }
12
13 void loop() {
14     // put your main code here, to run repeatedly:
15
16 }

```

Cấu trúc: PWMOUT(kênh, duty);

Trong đó: unsigned char kênh = 1 tới 10

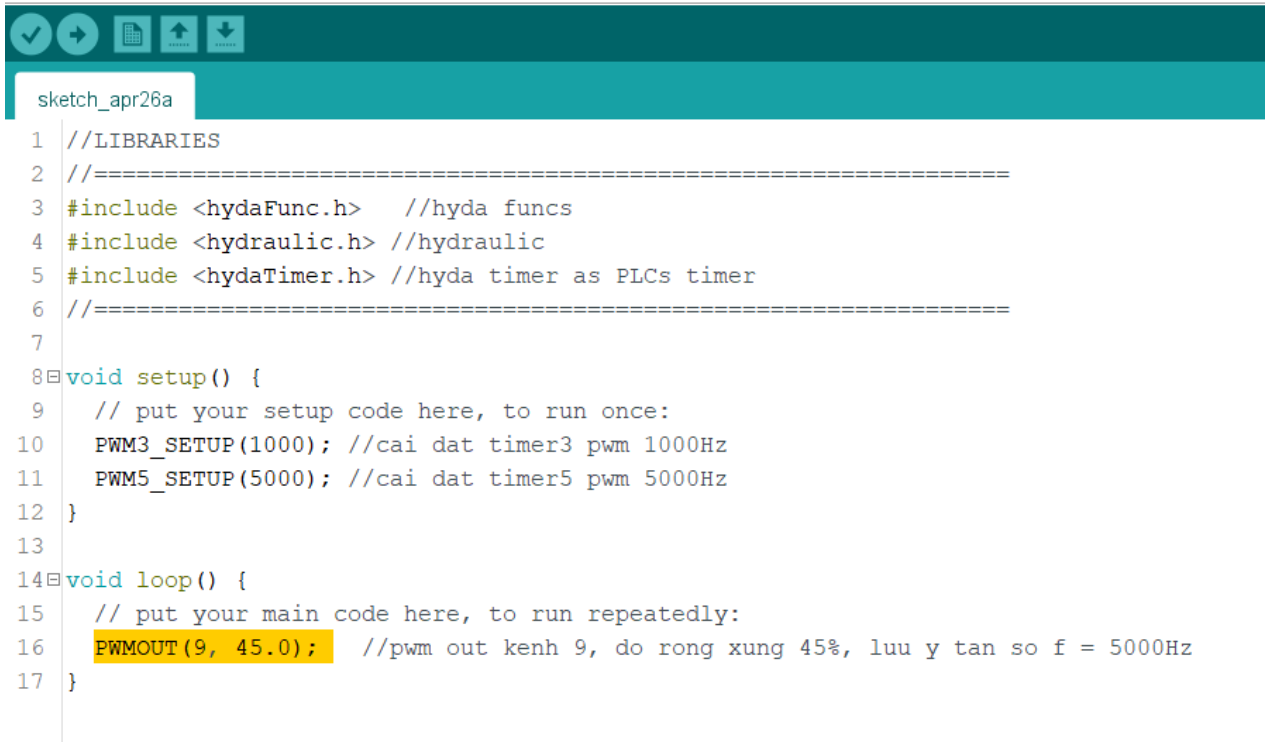
Độ rộng xung Duty = 0 tới 100.0 (float)

Ví dụ:

PWMOUT(9, 45.0); //PWM out kênh 9, độ rộng xung 45%

sketch_apr26a | Arduino 1.8.13

File Edit Sketch Tools Help



```

1 //LIBRARIES
2 //=====
3 #include <hydaFunc.h> //hyda funcs
4 #include <hydraulic.h> //hydraulic
5 #include <hydaTimer.h> //hyda timer as PLCs timer
6 //=====
7
8 void setup() {
9     // put your setup code here, to run once:
10    PWM3_SETUP(1000); //cai dat timer3 pwm 1000Hz
11    PWM5_SETUP(5000); //cai dat timer5 pwm 5000Hz
12 }
13
14 void loop() {
15     // put your main code here, to run repeatedly:
16    PWMOUT(9, 45.0); //pwm out kênh 9, do rong xung 45%, lưu ý tần số f = 5000Hz
17 }

```

c. HÀM ĐỌC GIÁ TRỊ DÒNG ĐIỆN RA TẢI

Mỗi kênh đầu ra đều có kênh đo dòng phản hồi (dòng điện cấp ra tải), để đọc được giá trị ta sử dụng hàm sau:

```
unsigned int CUR(unsigned char channelNumber);
```

giá trị trả về có kiểu dữ liệu “int” từ 0-1023 (đơn vị)

Dòng điện được tính như sau:

Kênh DO từ 1- 8:

$$I = \text{gia_tri_ham} * 23.46 \text{ (mA)}$$

Kênh DO9 & DO10

$$I = \text{gia_tri_ham} * 4.8876 \text{ (mA)}$$

Ví dụ:

Trong chương trình dưới đây dòng ra tải kênh DO5 và DO9 được đọc lưu vào 2 biến dong5 và dong9

sketch_apr26a | Arduino 1.8.13

File Edit Sketch Tools Help

```

sketch_apr26a $
3 #include <hydaFunc.h> //hyda funcs
4 #include <hydraulic.h> //hydraulic
5 #include <hydaTimer.h> //hyda timer as PLCs timer
6 //=====
7 //KHAI BAO BIEN TOAN CUC
8 float dong5 = 0; //khai bao 2 bien dong5 va dong9
9 float dong9 = 0;
10
11 //=====
12 void setup() {
13     // put your setup code here, to run once:
14     // chu y: MAINSWITCH da duoc dinh nghia truoc, ko can dinh ngia lai!!!
15     digitalWrite(MAINSWITCH, HIGH); //bat cong tac nguon chinh, ham nguuyen ban cua arduino
16
17     PWM3_SETUP(1000); //cai dat timer3 pwm 1000Hz
18     PWM5_SETUP(5000); //cai dat timer5 pwm 5000Hz
19 }
20
21 void loop() {
22     // put your main code here, to run repeatedly:
23     PWMOUT(9, 45.0); //pwm out kênh 9, do rong xung 45%, luu y tan so f = 5000Hz
24     // kích dàu ra 5
25     DOUT(5, 1); //bat kênh 5 = ON
26
27     //doc dong dien ra tai kênh 5 va 9
28     // gia tri dong5 va dong9 mang kieu du lieu float, CUR(i) mang kieu int do do can chuyen sang float
29     dong5 = float(CUR(5)) * 23.46; //I = gia_trị_hàm * 23.46 (mA) kênh tu 1 >> 8
30     dong9 = float(CUR(9)) * 4.8876; //I = gia_trị_hàm * 4.8876 (mA) kênh tu 9 >> 10
31
32 }

```

d. ĐỌC TÍN HIỆU CHUẨN ĐOÁN LỖI

Mỗi kênh đầu ra từ 1 tới 8 đều có 1 kênh chuẩn đoán lỗi, dựa vào tín hiệu này và dòng cấp ra tải có thể đưa ra các chuẩn đoán như:

- Ngắn mạch
- Hở mạch
- Chập sang nguồn cấp 24V
- Quá dòng

Sử dụng hàm sau để đọc tín hiệu chuẩn đoán lỗi

```
unsigned char STA(unsigned char channelNo)
```

- channelNo: kênh cần đọc
- giá trị trả về là 0 hoặc 1

Kết quả chuẩn đoán tuân theo bảng logic sau:

Chỉ Áp dụng cho kênh ra từ 1 tới 8

Trạng thái kênh	Đầu vào	Đầu ra	Kênh chuẩn đoán	Dòng ra tải
Bình thường	L	L	H	0
	H	H	L	Thông thường
Quá dòng	L	L	H	0
	H	H	H	0
Ngắn mạch	L	L	H	0
	H	L	H	0
Quá nhiệt IC	L	L	H	0
	H	L	H	0
Chập sang nguồn	L	H	L	0
	H	H	L	< thông thường
Hở mạch (đứt dây)	L	L	H	0
	H	H	L	0

Với kênh DO9 và DO10

Nếu IC bị quá nhiệt, tín hiệu trạng thái báo quá nhiệt STA sẽ bị kéo về 0.