Làm quen với Makerbot BANHMI

FPT Education
TRƯỜNG ĐẠI HỌC FPT

MAKER VIỆT

Nôi dung

- Thông tin **lập trình Makerbot BANHMI**
- Những **lưu ý** khi sử dụng **Makerbot BANHMI**
- Bài **Quizz 30’** cho **các thí sinh tham dự**

Mục tiêu

- Nắm được **cách sử dụng Makerbot BANHMI**
- Biết và hiểu cách lập trình **Makerbot BANHMI**
- Thử sức với bài QUIZZ Test

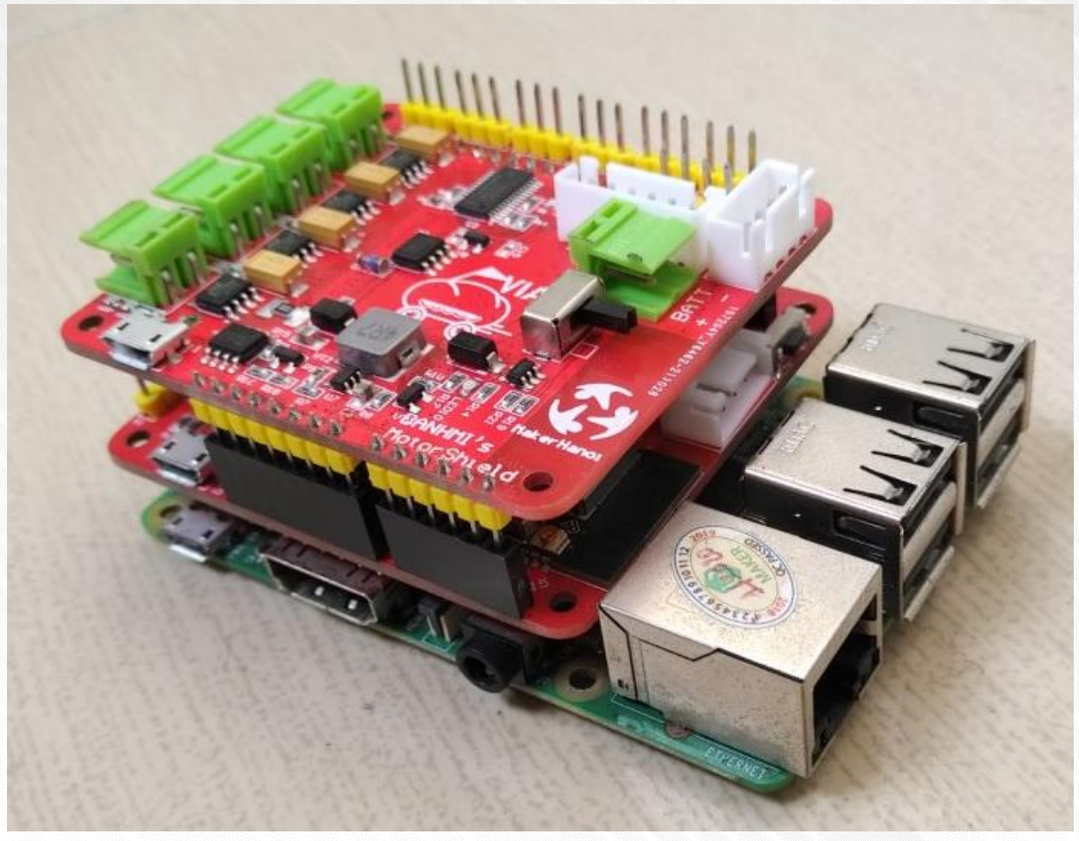
MakerBot BANHMI

TRƯỜNG ĐẠI HỌC FPT

MAKER VIỆT

Kit phát triển phần cứng mã nguồn mở

- Hướng tới phát triển robot và xe tự hành
- Có thể kết nối với Raspberry và Shield của Arduino Uno
- Hỗ trợ điều khiển lên đến 10 động cơ độc lập (4 DC,6 servo)
- Hỗ trợ các chuẩn kết nối không dây: WIFI, Bluetooth (truyền dữ liệu và điều khiển)
- Hỗ trợ mở rộng cảm biến, thiết bị với:, Uart, I2C
- Thiết kế đơn giản, dễ tiếp cận
- Thiết kế dành cho dự án mã nguồn mở VIA(Vietnam Autonomus) https://via.makerviet.org/vi/

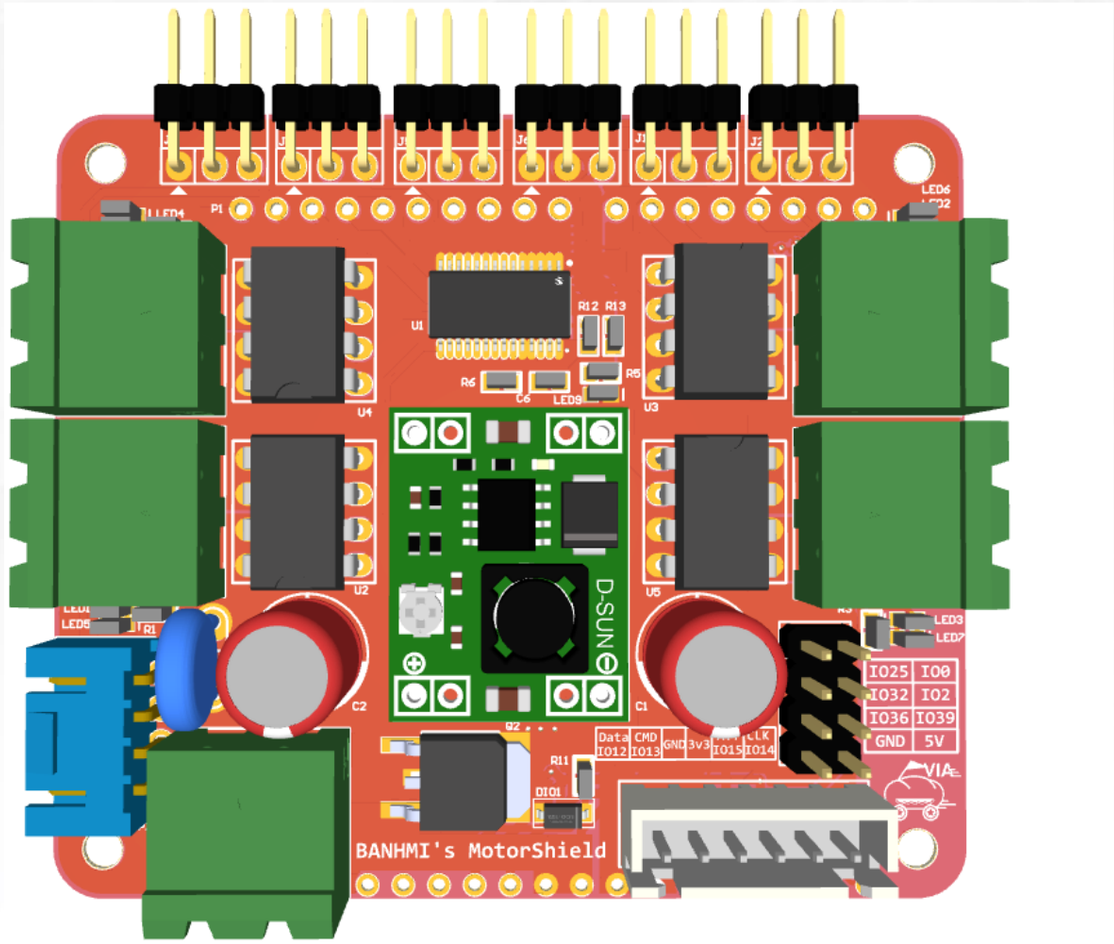


MakerBot BANHMI

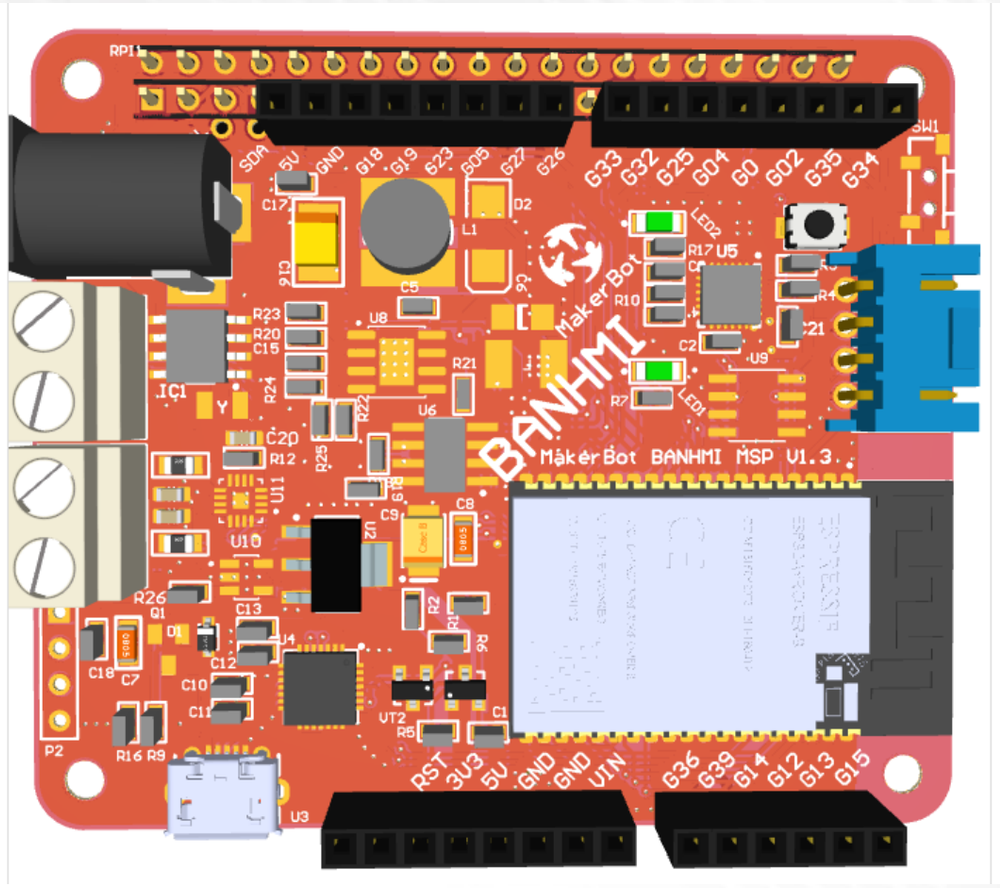
FPT Education
TRƯỜNG ĐẠI HỌC FPT

MAKER VIỆT

Thành phần chính mạch Makerbot BANHMI



Mạch công suất



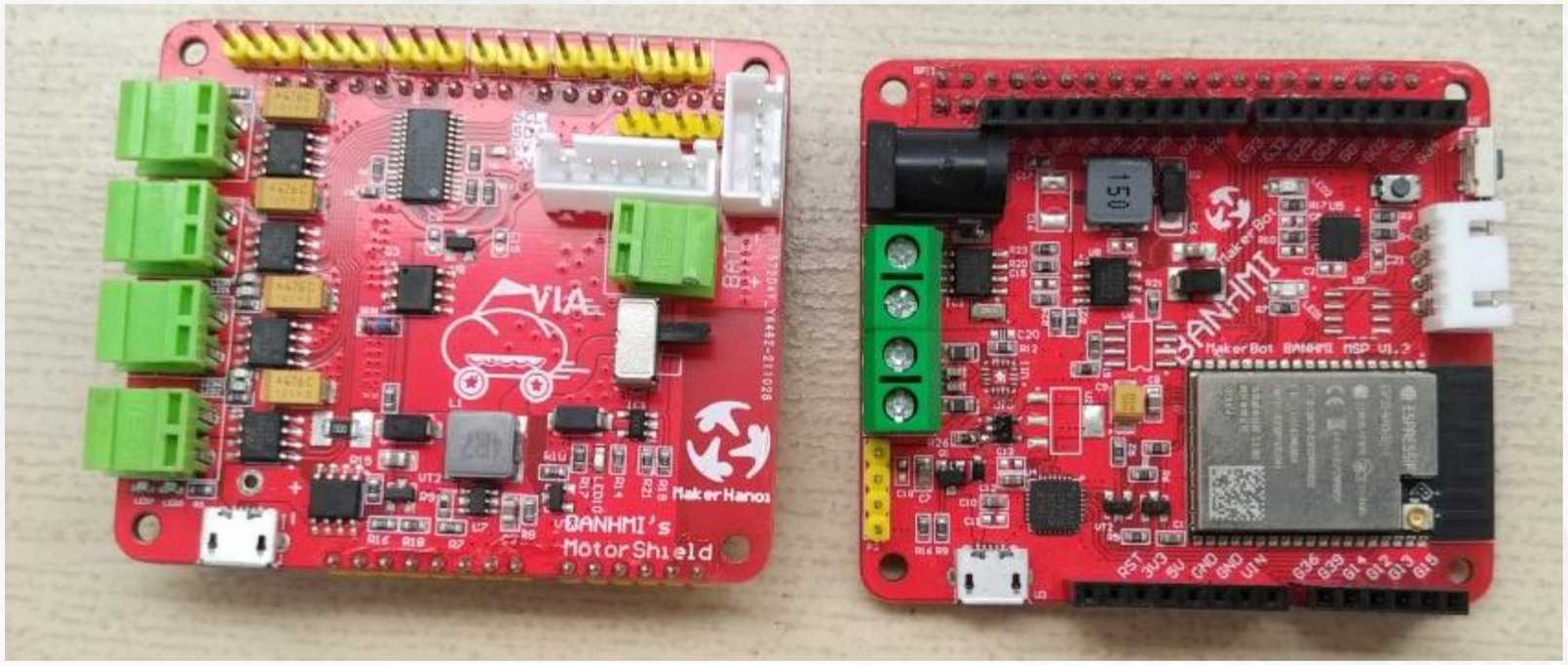
Mạch điều khiển

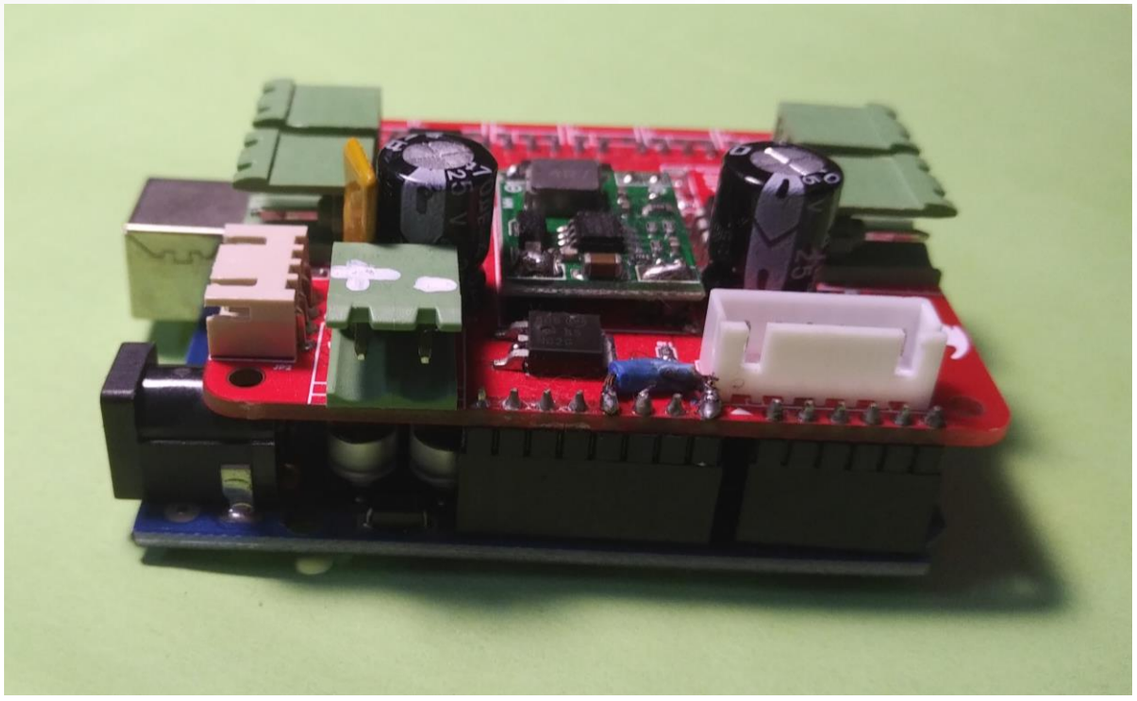
MakerBot BANHMI

FPT Education
TRƯỜNG ĐẠI HỌC FPT

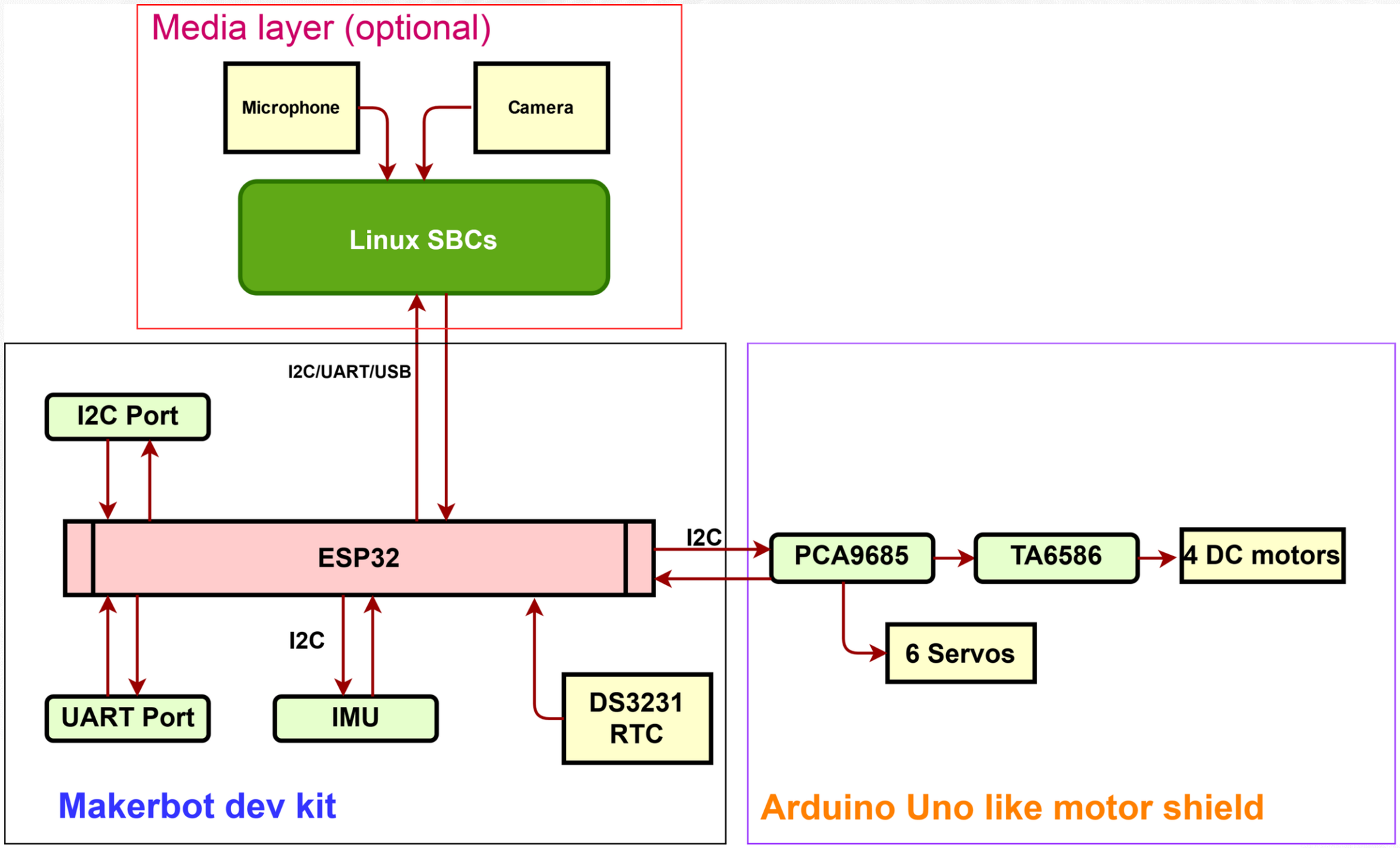
MAKER VIỆT

HÌnh ảnh MakerBot BANHMI





MakerBot BANHMI



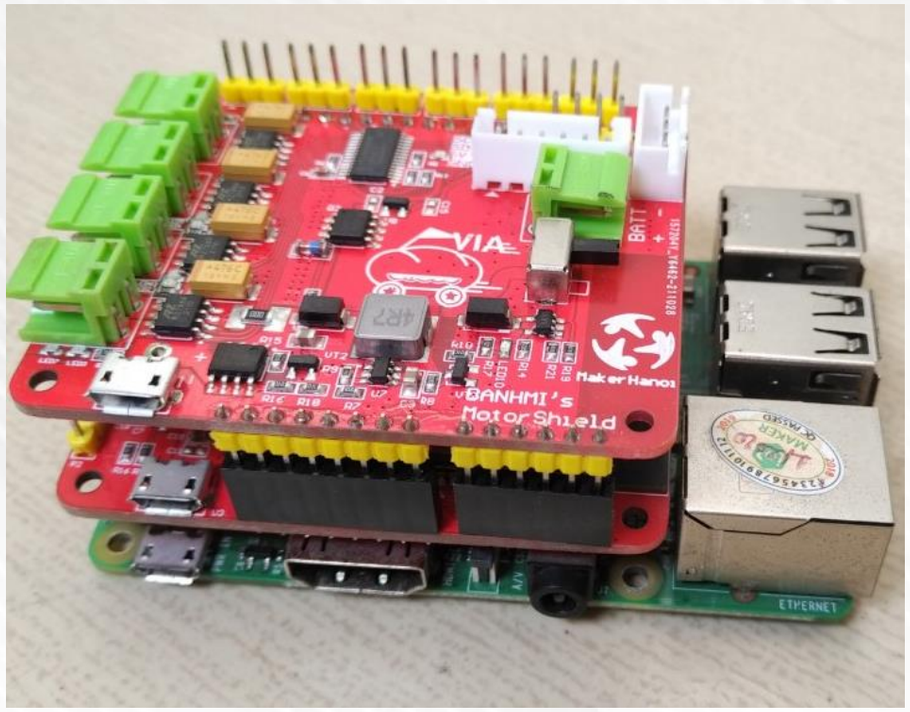
MakerBot BANHMI

TRƯỜNG ĐẠI HỌC FPT

MAKER VIỆT

Mạch điều khiển Makerbot BANHMI

- Mạch điều khiển hệ thống động, cơ cảm biến
- Có khả năng kết nối truyền dữ liệu qua WIFI, BLE
- Nhỏ gọn, có header tương thích với Raspberry PI
- Tương thích ngược với Arduino UNO shield
- Ngoài ra, 1 cổng UART, 1 cổng I2C mở rộng



Mạch công suất Makerbot BANHMI

- 4 đầu ra động cơ DC 5v , 6 đầu ra động cơ Servo 5-7V
- 1 cổng I2C, 1 cổng SPI
- Header mở rộng 6 chân GPIO

MakerBot BANHMI

TRƯỜNG ĐẠI HỌC FPT

MAKER VIỆT

Cấu Hình chi tiết

Khối điều khiển

- ESP32 @ 2x240Mhz,8 MB flash,8MB Psram, 34 GPIOs, built in WIFI,Bluetooth
- IMU: 6DOF MPU6050
- RTC: DS1307

Khối công suất

- PWM: PCA9685
- H-bridge: TA6586/RZ7886
- 6 Ports Servo, 4 Ports DC

FPT Education

TRƯỜNG ĐẠI HỌC FPT

MAKER VIỆT

Bắt đầu với MakerBot BANHMI

FPT Education
TRƯỜNG ĐẠI HỌC FPT

MAKER VIỆT

Lưu ý quan trọng khi sử dụng Makerbot BANHMI

- **Nên** sử dụng dây cáp ***micoUSB có chất lượng tốt***
- ***Không nên*** cắm mạch makerbot qua bộ chia ***USB (USB HUB)***
- Nếu có thể, ***nên*** cắm mạch MakerBot vào ***cổng USB 3.0*** để đảm bảo nguồn cung cấp năng lượng cho mạch MakerBot
- ***Không nên*** cắm ***ngược chiều nguồn điện*** cho mạch công suất Makerbot
- ***Không*** cắm ***nguồn điện*** trực tiếp ***vào chân IO*** của mạch Makerbot, hay ***bất cứ connector nào*** không phục vụ cho việc cấp nguồn
- ***Không*** cắm các thiết bị chưa rõ tương thích chân vào mạch makerbot.
- ***Tuyệt đối không*** cắm ***pin*** vào ***connector 4 chân hoặc 6 chân trên mạch công suất***

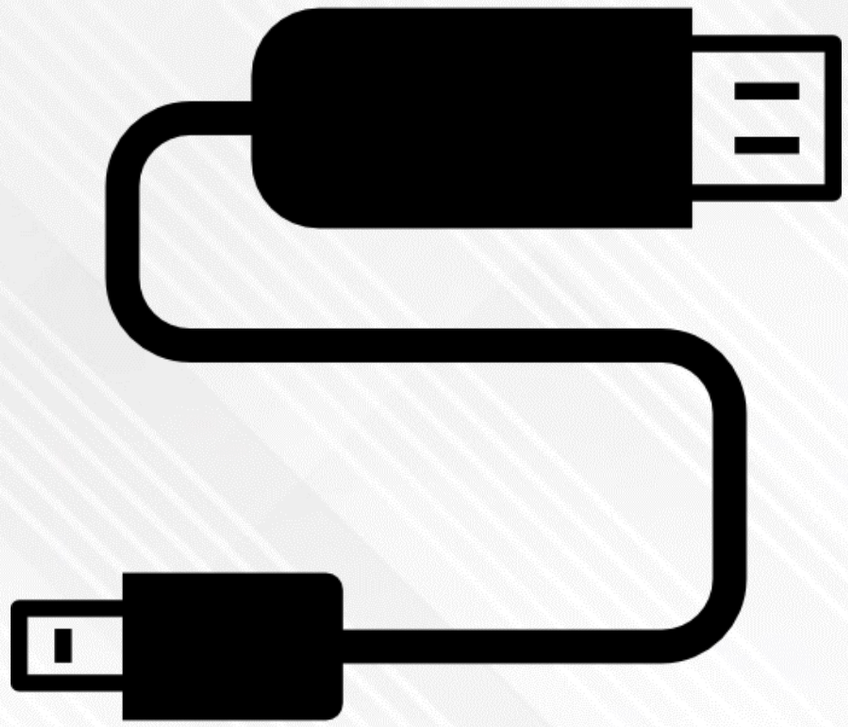
Bắt đầu với MakerBot BANHMI

FPT Education
TRƯỜNG ĐẠI HỌC FPT

MAKER VIỆT

Kết nối mạch Makerbot với máy tính

Kết nối MakerBot với máy tính qua cáp microUSB



Lưu ý

- ***Mạch điều khiển động cơ tuy sáng đèn khi cắm usb nhưng chỉ động cơ chỉ hoạt động khi có nguồn 12V cắm vào mạch***
- *Nên sử dụng dây cáp micoUSB có chất lượng tốt*
- *Không nên cắm mạch makerbot qua bộ chia USB (USB HUB)*
- *Nếu có thể, nên cắm mạch MakerBot vào cổng USB 3.0 để đảm bảo nguồn cung cấp năng lượng cho mạch MakerBot*

Bắt đầu với MakerBot BANHMI

TRƯỜNG ĐẠI HỌC FPT

MAKER VIỆT

Kết nối mạch Makerbot với máy tính

Khi kết nối với máy tính, máy tính sẽ tự động tiến hành cài đặt driver cho mạch MakerBot, sau khi cài đặt driver xong trên máy tính sẽ xuất hiện thiết bị *Silicon Labs CP210x USB to UART Bridge*

Đối với Windows

- Sau khi cài đặt driver, mach MakerBot sẽ được gán với 1 cổng COM trên máy, điền bên cạnh tên thiết bị

  ví dụ Silicon Labs CP210x USB to UART Bridge (COM 3)

- Chú ý ghi nhớ tên cổng COM này (COM3) để thuận lợi cho các bước tiếp theo

Bắt đầu với MakerBot BANHMI

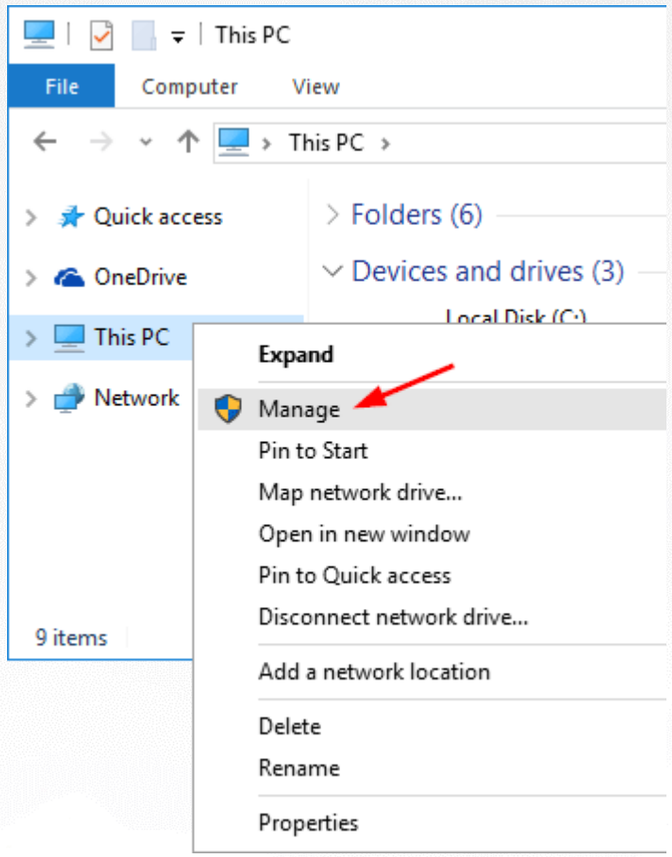
FPT Education
TRƯỜNG ĐẠI HỌC FPT

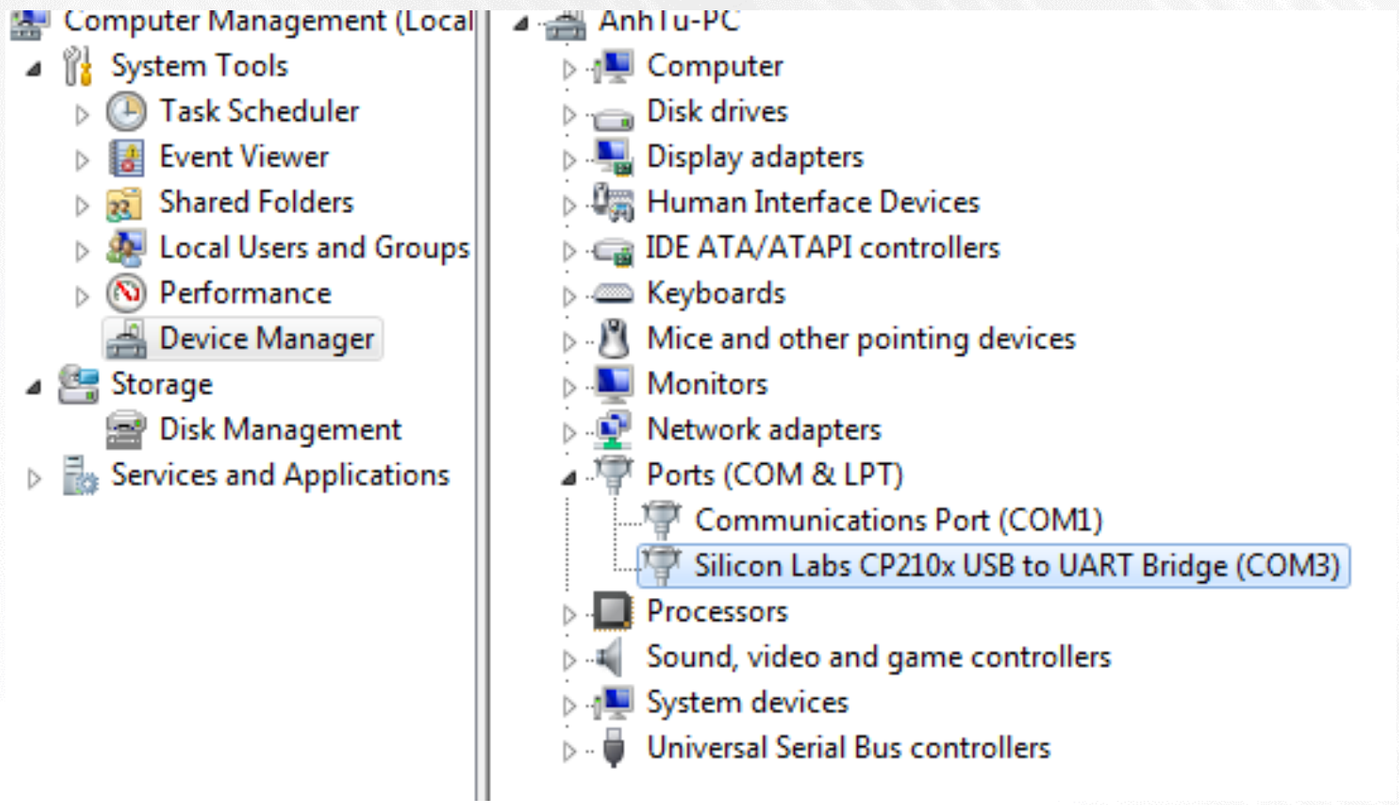
MAKER VIỆT

Kết nối mạch Makerbot với máy tính

Đối với Windows

Để kiểm tra lại kết nối: click chuột phải ***This PC->manage->Device Manager -> Port(COM&LPT)***



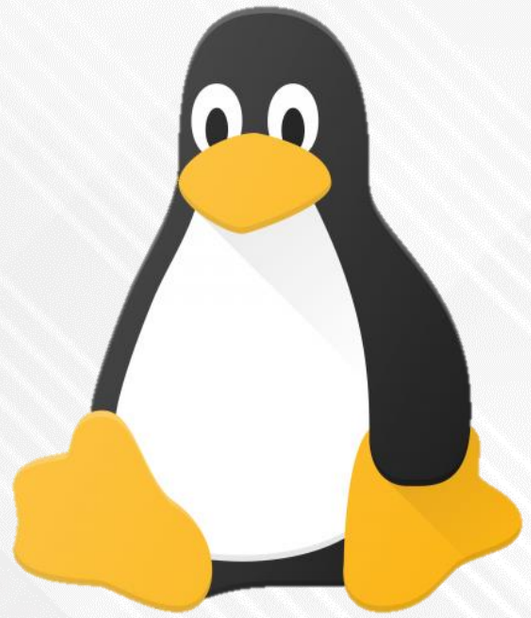


Bắt đầu với MakerBot BANHMI

TRƯỜNG ĐẠI HỌC FPT

MAKER VIỆT

Kết nối mạch Makerbot với máy tính



Đối với Linux

- Có thể kiểm tra bằng lệnh *lsusb* và lệnh *ls /dev/ttyUSB* hoặc ls /dev/ttyACM**

Đối với MacOS

- *Cài đặt driver: https://www.silabs.com/developers/usb-to-uart-bridge-vcp-drivers*

Sau khi cài driver

- *V*ào "System Preferences" -> "Security & Privacy"
- Ở góc dưới cửa sổ sẽ có chữ "System software from developer "SiLabs" was blocked from loading."
- Ấn nút "Allow"
- Khởi động lại Mac

Sau khi hoàn thành Mac sẽ nhận cổng COM với tên “/dev/cu.SLAB_USBtoUART “

Bắt đầu với MakerBot BANHMI

TRƯỜNG ĐẠI HỌC FPT

MAKER VIỆT

Kết nối mạch Makerbot với máy tính

Lỗi có thể gặp phải

Lỗi máy tính không nhận thiết bị, kết nối chập chờn, không ổn định sau 30s kể từ khi kết nối

Cách khắc phục

- *Đổi cáp microUSB, đổi cổng kết nối USB,*
- *Cắm nguồn 12v vào jack DC rồi kết nối với máy tính*
- *Nhấn giữ nút BOOT(nút tròn màu đen ở giữa mạch) rồi kết nối với máy tính*

FPT Education

TRƯỜNG ĐẠI HỌC FPT

MAKER VIỆT

Arduino và MakerBot BANHMI

Bắt đầu với MakerBot BANHMI

TRƯỜNG ĐẠI HỌC FPT

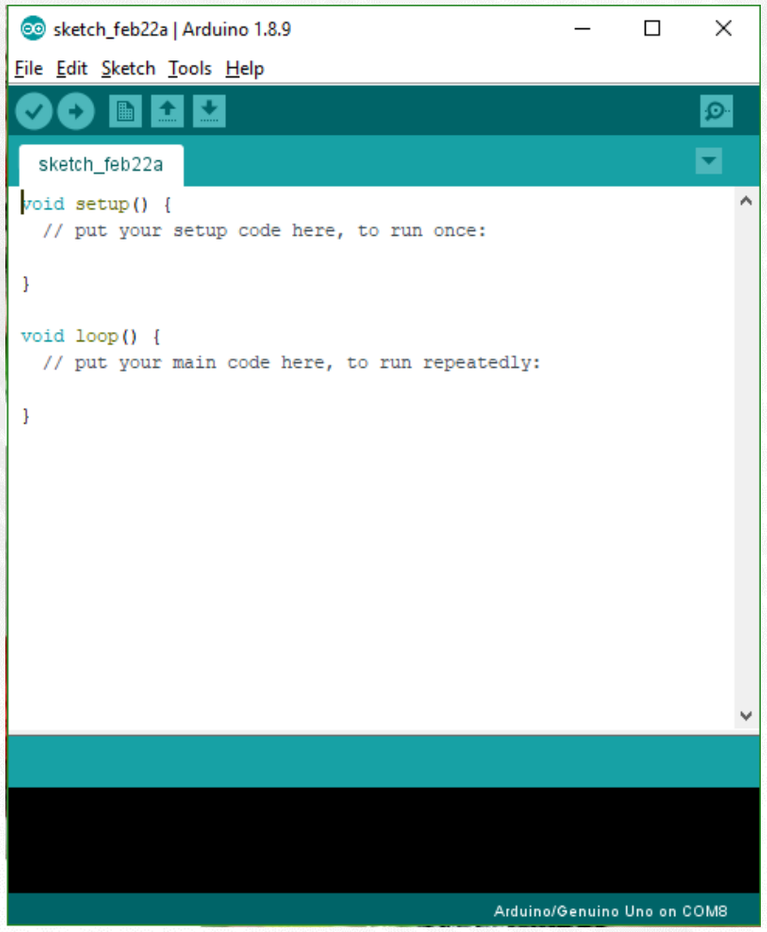
MAKER VIỆT

Cài đặt MakerBot BANHMI với Arduino

Arduino IDE

- Arduino là một nền tảng mã nguồn mở được sử dụng để xây dựng các dự án điện tử. Arduino bảng mạch Arduino và Arduino IDE
- Arduino IDE (Arduino Integrated Development Environment) là một trình soạn thảo và biên dịch chương trình để nạp cho mạch Arduino

Mạch Makerbot sẽ sử dụng trình biên dịch và bộ thư viện của arduino



Bắt đầu với MakerBot BANHMI

FPT Education
TRƯỜNG ĐẠI HỌC FPT

MAKER VIỆT

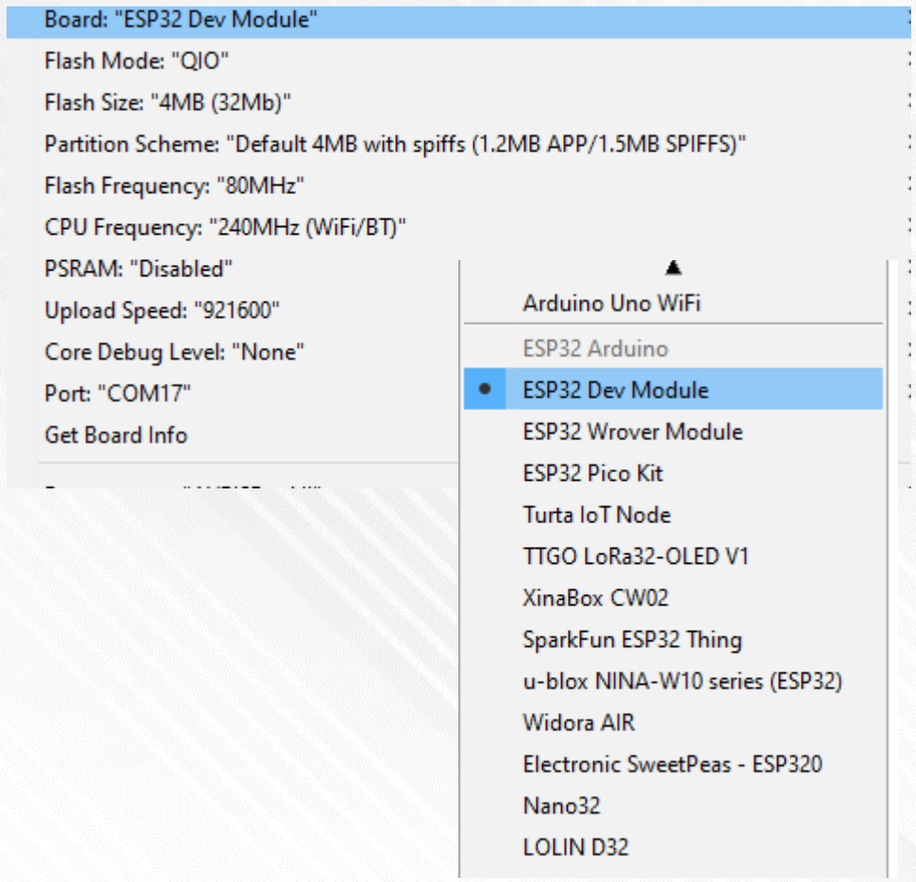
Cài đặt MakerBot BANHMI với Arduino

Cài Đặt ESP32 Arduino

- Mở **Arduino IDE>File>Preferences**
- Tại phần **Additional Board Manager URLs:**

https://dl.espressif.com/dl/package_esp32_index.json

- Chọn **Tool** > **Board** > **Boards Manager**
- Tìm kiếm với từ khóa “esp32” , trong danh sách kết quả chọn “**ESP32 by Espressif Systems”** và nhấn **Install**



Bắt đầu với MakerBot BANHMI

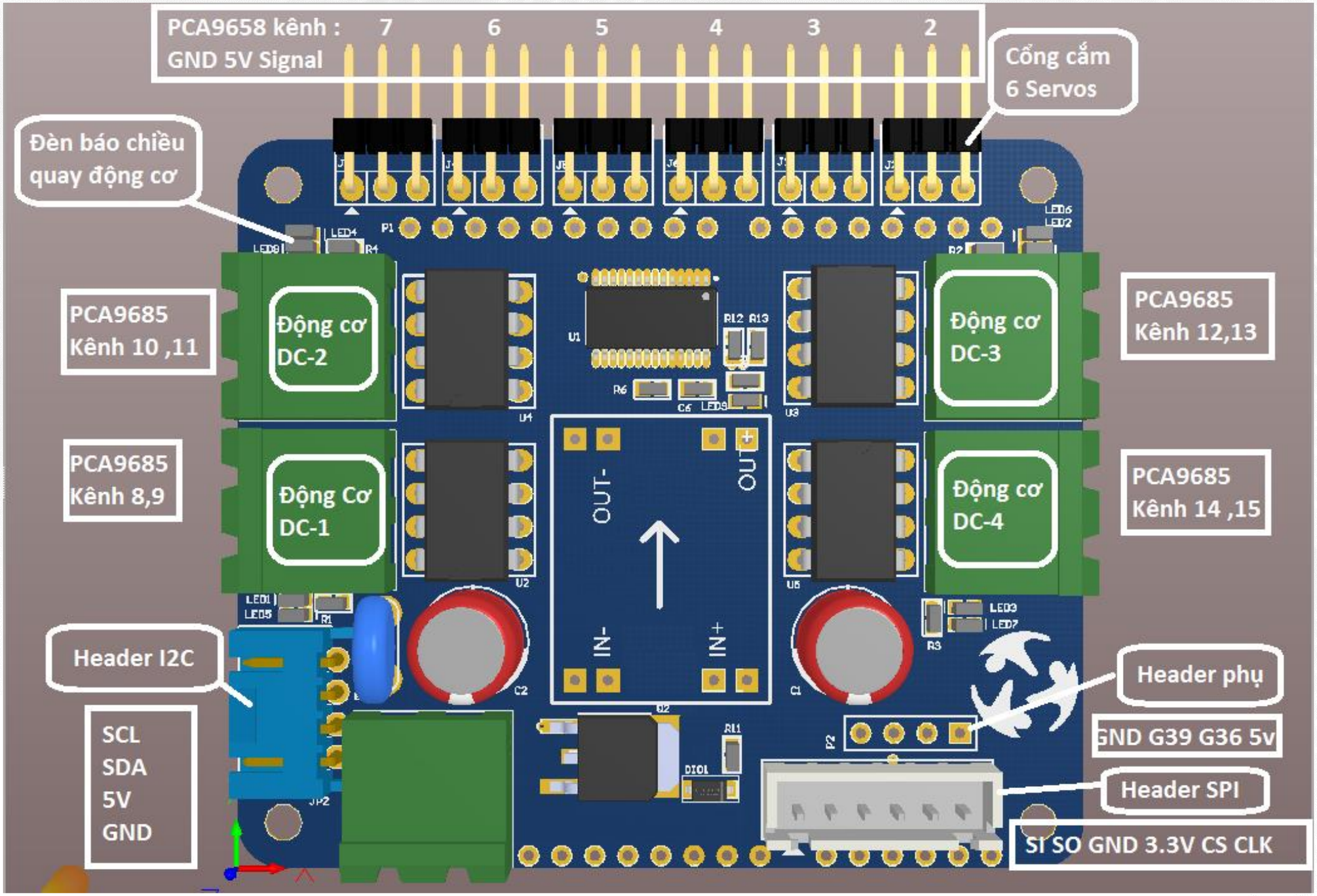
FPT Education
TRƯỜNG ĐẠI HỌC FPT

MAKER VIỆT

Cài đặt MakerBot BANHMI với Arduino

Platform IO

Lập trình MakerBot BANHMI với Arduino



Bắt đầu với MakerBot BANHMI

FPT Education
TRƯỜNG ĐẠI HỌC FPT

MAKER VIỆT

Điều khiển động cơ Servo và DC

Sử dụng thư viện Adafruit PCA9685:

Link tải thư viện: https://github.com/adafruit/Adafruit-PWM-Servo-Driver-Library

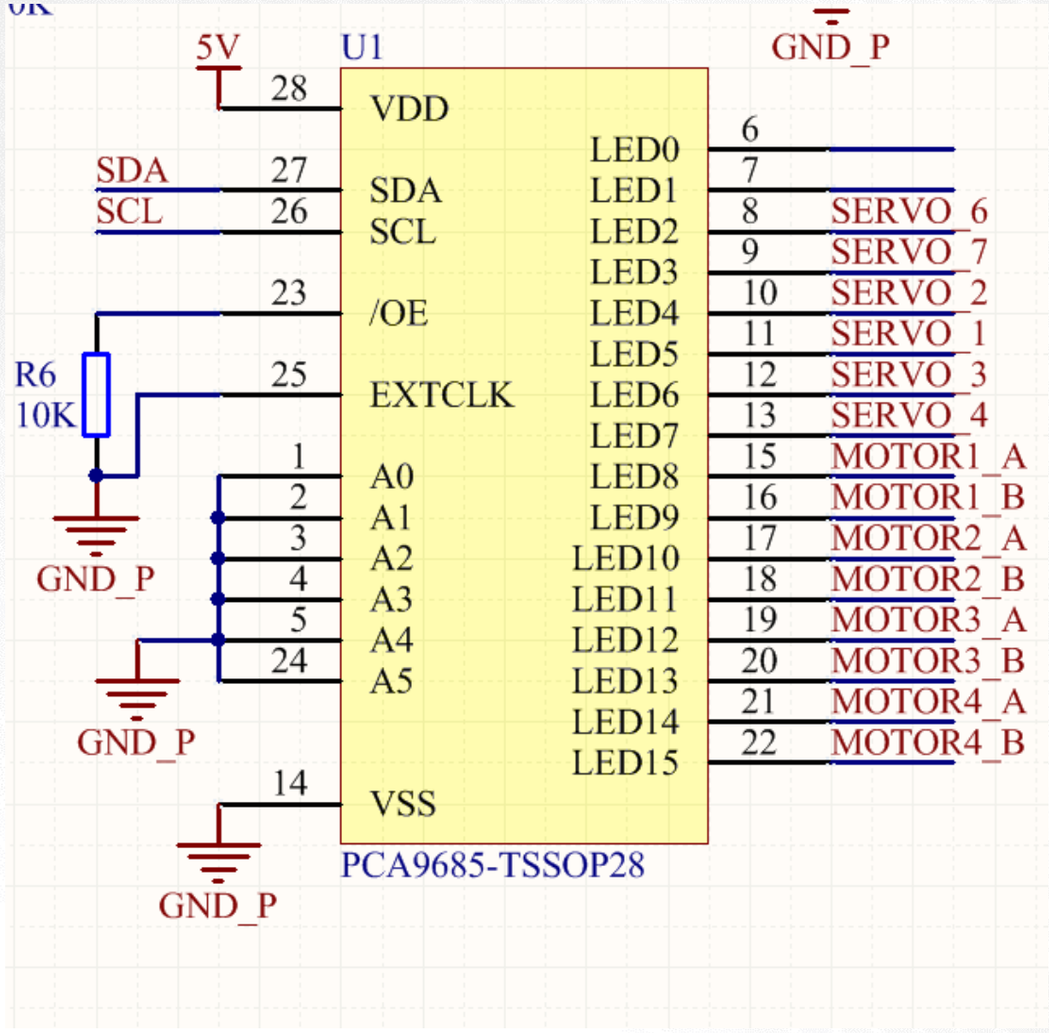
http://adafruit.github.io/Adafruit-PWM-Servo-Driver-Library/html/class_adafruit___p_w_m_servo_driver.html

Ví dụ mẫu điều khiển servo **File->examples->Adafruit-PWM-Servo-Driver-Library>servo**

Ví dụ PWM: **File->examples->Adafruit-PWM-Servo-Driver-Library->PWMtest**

Tài liệu tham khảo PWM:

http://arduino.vn/reference/xung-pwm



Điều khiển động cơ Servo và DC

Khai báo thư viện

```
#include <Wire.h> //thư viện I2c của Arduino, do PCA9685 sử dụng
chuẩn giao tiếp i2c nên thư viện này bắt buộc phải khai báo
#include <Adafruit_PWMServoDriver.h> // thư viện PCA9685
```

Khởi tạo clas của thư viện với địa chỉ gốc

```
Adafruit_PWMServoDriver pwm = Adafruit_PWMServoDriver();
```

Hoặc khởi tạo với địa chỉ tùy biến, kết nối nhiều mạch

```
Adafruit_PWMServoDriver pwm1 = Adafruit_PWMServoDriver(0x70);
Adafruit_PWMServoDriver pwm2 = Adafruit_PWMServoDriver(0x71);
```

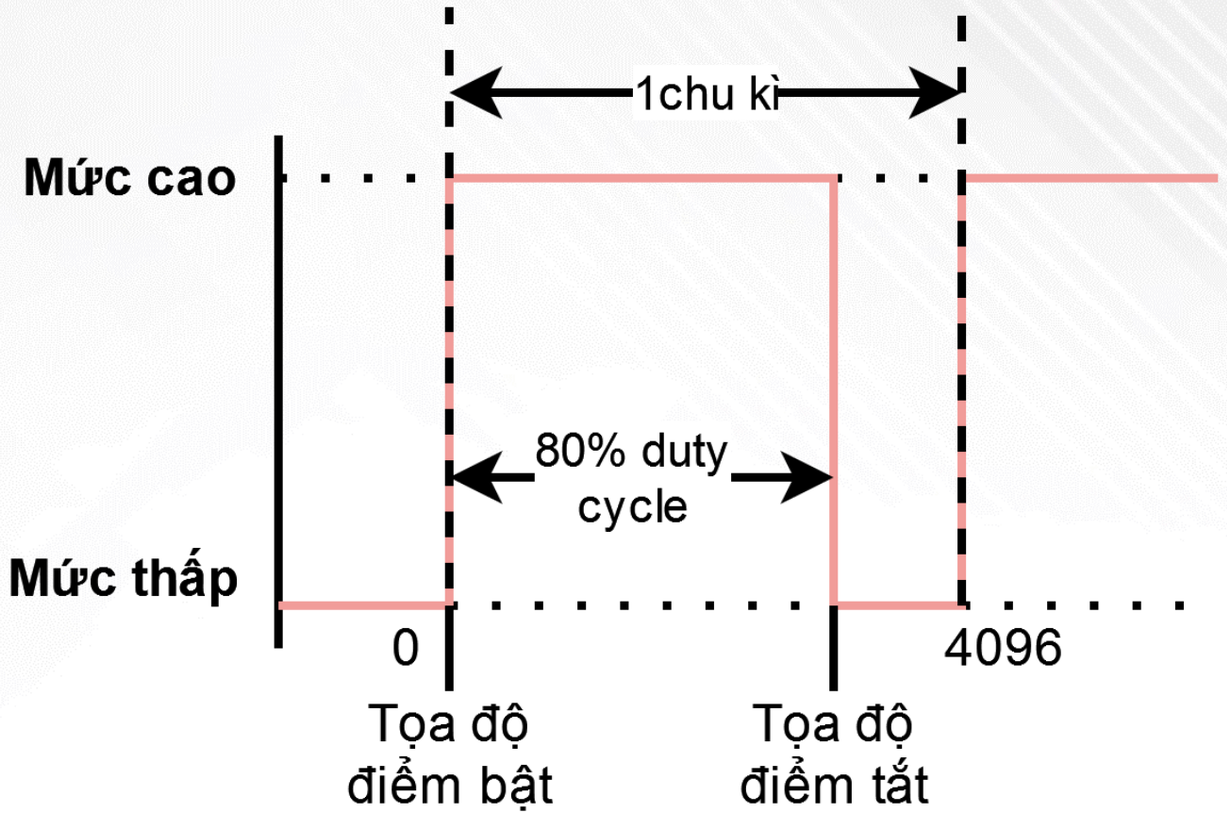
Điều khiển động cơ Servo và DC

```
Khởi tạo trong hàm setup
pwm.begin(); //khởi tạo PCA9685
pwm.setOscillatorFrequency(27000000); // cài đặt tần số giao động
pwm.setPWMFreq(50);// cài đặt tần số PWM //tần số PWM được cài đặt từ 24-1600 HZ, tần số này được cài đặt tùy thuộc vào nhu cầu xử dụng

Để điều khiển được cả servo và động cơ DC cùng nhau, tần số PWM điều khiển được cài đặt trong khoảng50-60Hz Khởi tạo trong hàm setup

Wire.setClock(400000); // cài đặt tốc độ giao tiếp i2c ở tốc độ cao nhất(400 Mhz) , hàm này có thể bỏ qua nếu gặp lỗi hoặc không có nhu cầu tử dụng I2c tốc độ cao
```

Điều khiển động cơ Servo và DC

Cấu trúc Hàm băm xung PWM

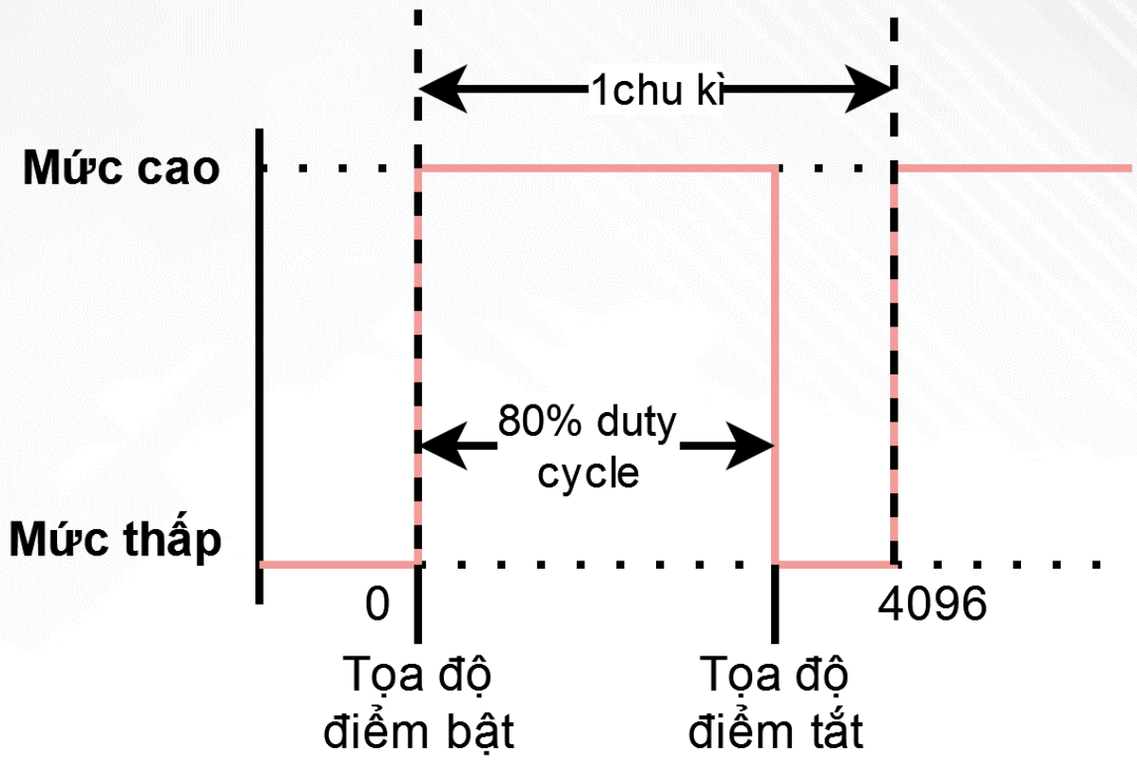
Tên class	Đối tượng	Tham số		
pwm	setPWM	Kênh PWM số	Tọa độ bật (0-4096)	Tọa độ tắt (0-4096)
pwm.setPWM(kênh PWM, tọa độ bật, tọa độ tắt);				



Điều khiển động cơ Servo và DC

Cấu trúc Hàm băm xung PWM

```
pwm.setPWM(kênh PWM,toa độ bật, toa độ tắt);
//kênh PWM, kênh đầu ra có thế xem hình ở đầu slide và điền vào số kênh muốn
điều khiển trong khoảng 0-15
//toa độ trị bật, tọa đổ điểm bắt đầu nâng mức logic lên cao 0-4095 (2^12)
//giá trị bật quyết định tốc độ của động cơ (duty cycle), toa độ trị bắt, tọa
đổ điểm kết thúc hạ nâng mức logic xuống thấp
```



Điều khiển động cơ Servo và DC

Cấu trúc Hàm băm xung PWM

Ví dụ so sánh với hàm **analogWrite()**

```
pwm.setPWM(13 ,0, 255); // chọn chân số 13, giá trị pwm 255
 tương đương
analogWrite(13 , 255)
```

Hoặc

```
pwm.setPWM(13 ,0, 0); // chọn chân số 13, giá trị pwm 0
 tương đương
analogWrite(13 , 0)
```

Điều khiển động cơ Servo và DC

Điều khiển động cơ DC

```
pwm.setPWM(chan2, 0, val);
pwm.setPWM(chan1, 4096, 0); //makerbot Sử dụng 2 kênh của PCA9685 , để điều khiển động cơ qua 1 chân luôn ở trạng thái tắt
```

Điều khiển động cơ Servo và DC

Điều khiển động cơ DC

Ví dụ: điều khiển động cơ số 1 tốc độ quay 50%, chiều quay thuận

```
pwm.setPWM(8, 0, 2048); //chân số 8 set chiều dương là PWM 50%
pwm.setPWM(9, 0, 0);    //chân số 9 set chiều âm
//điều khiển kênh 8 và 9 của động cơ 1, tốc độ 50% = 4096/2
```

Ví dụ: điều khiển động cơ số 1 tốc độ quay 75%, chiều quay ngịch

```
pwm.setPWM(8, 0, 0);    //chân số 8 set chiều âm
pwm.setPWM(9, 0, 2730); //chân số 9 set chiều dương là PWM 75%
//điều khiển kênh 8 và 9 của động cơ 1, tốc độ 75% = 4096/1.5
```

Bắt đầu với MakerBot BANHMI

TRƯỜNG ĐẠI HỌC FPT

MAKER VIỆT

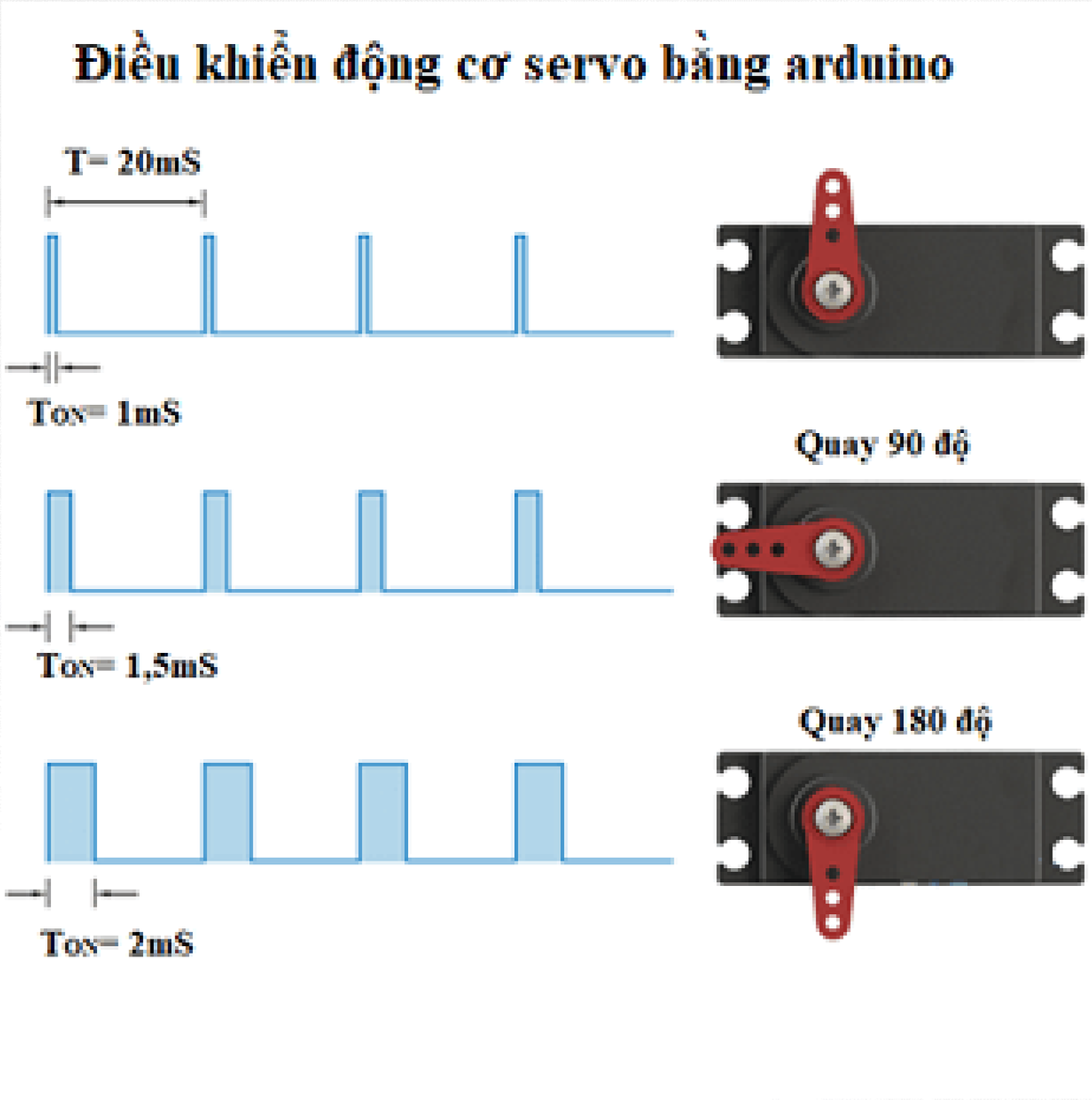
Điều khiển động cơ Servo và DC

Điều khiển Servo

Để điều khiển động cơ servo chúng ta sử dụng xung PWM ở tần số 50Hz

Cách điều khiển góc (đối với Servo 180) dựa theo độ rộng xung bật như hình bên

Đối với Servo 360 ta không điều khiển được góc mà chỉ điều khiển được tốc độ quay và chiều quay dựa vào độ rộng xung bật tương tự như servo 180

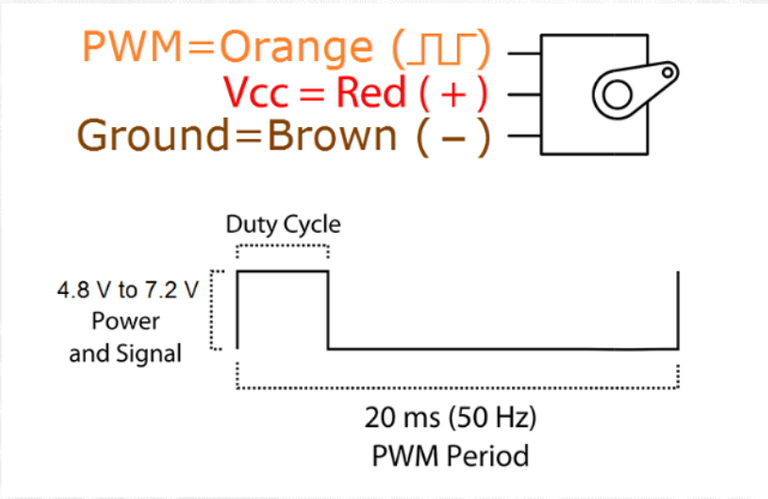


Điều khiển động cơ Servo và DC

Điều khiển Servo

Điều khiển sử dụng hàm băm xung PWM

- Tính độ giá trị PWM dựa theo thời gian
  ví dụ ở góc 180 độ
  pwm_val = T_on/(Ts/4096) = 2/(20/4096) = 409.6



Vậy để set góc 180 độ ta cần sử dụng:

```
pwm.setPWM(5, 0, 410); // chọn kênh servo số 5
```

Điều khiển sử dụng hàm set thời gian

- Đưa vào giá trị thời gian chính xác ở đơn vị micro giây

```
pwm.writeMicroseconds(kênh PWM, microsec);
//Microsec, thời gian xung ở mức cao trong 1 chu kì( trạng thái bật)
```

Ví dụ set góc 180 độ:

```
pwm.writeMicroseconds(5, 2000); // chọn kênh servo số 5
```

Tay điều khiển PS2

TRƯỜNG ĐẠI HỌC FPT

MAKER VIỆT

Tay điều khiển PS2 không dây

Sử dụng thư viện tay cầm PS2:

Link tải thư viện: https://github.com/makerhanoi/Arduino-PS2X-ESP32-Makerbot

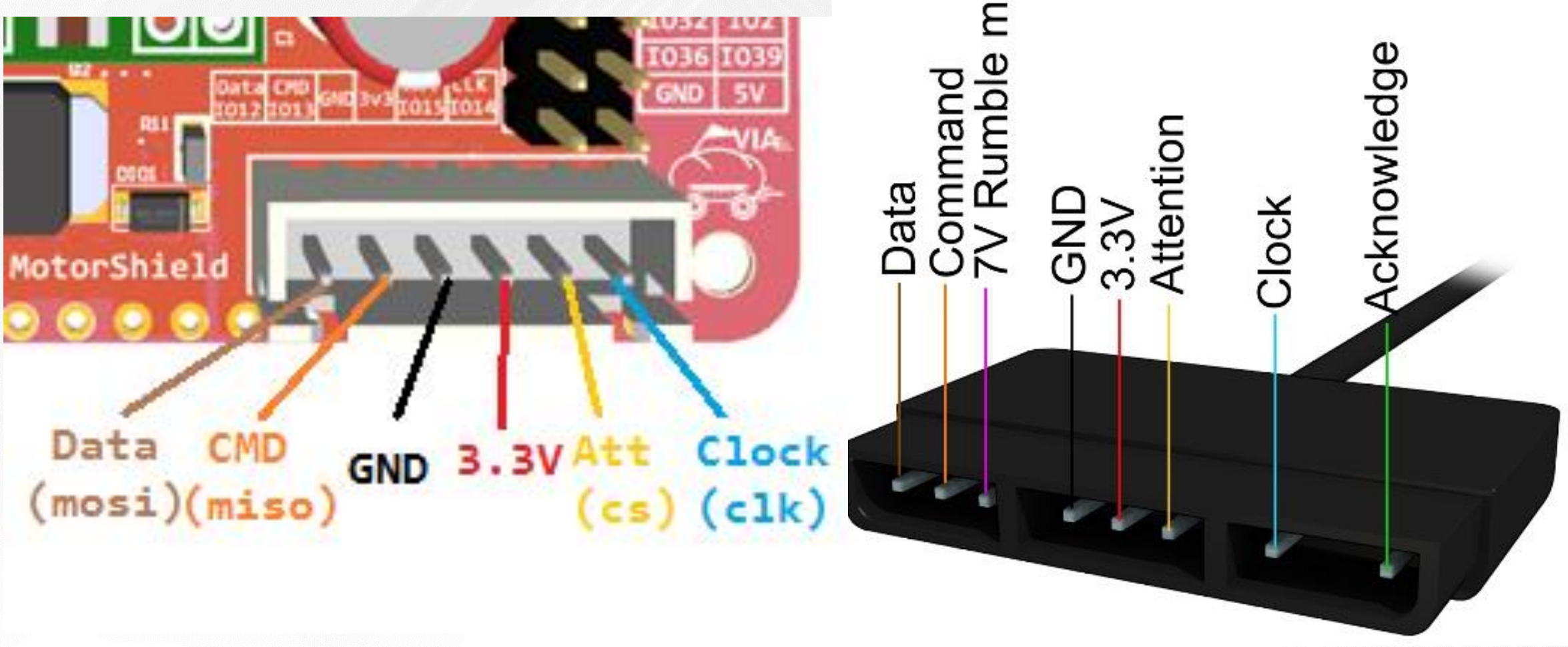
Ví dụ mẫu **File->examples->Arduino-PS2X-ESP32-Makerbot>PS2X_Example_VIA_Makerbot_ESP32**

Tay điều khiển

FPT Education
TRƯỜNG ĐẠI HỌC FPT

MAKER VIỆT

Sơ đồ kết nối tay điều khiển PS2



Lưu ý có 2 chân ta không sử dụng là chân số 3 (Rumble motor) và chân số 7 (Acknowledge)

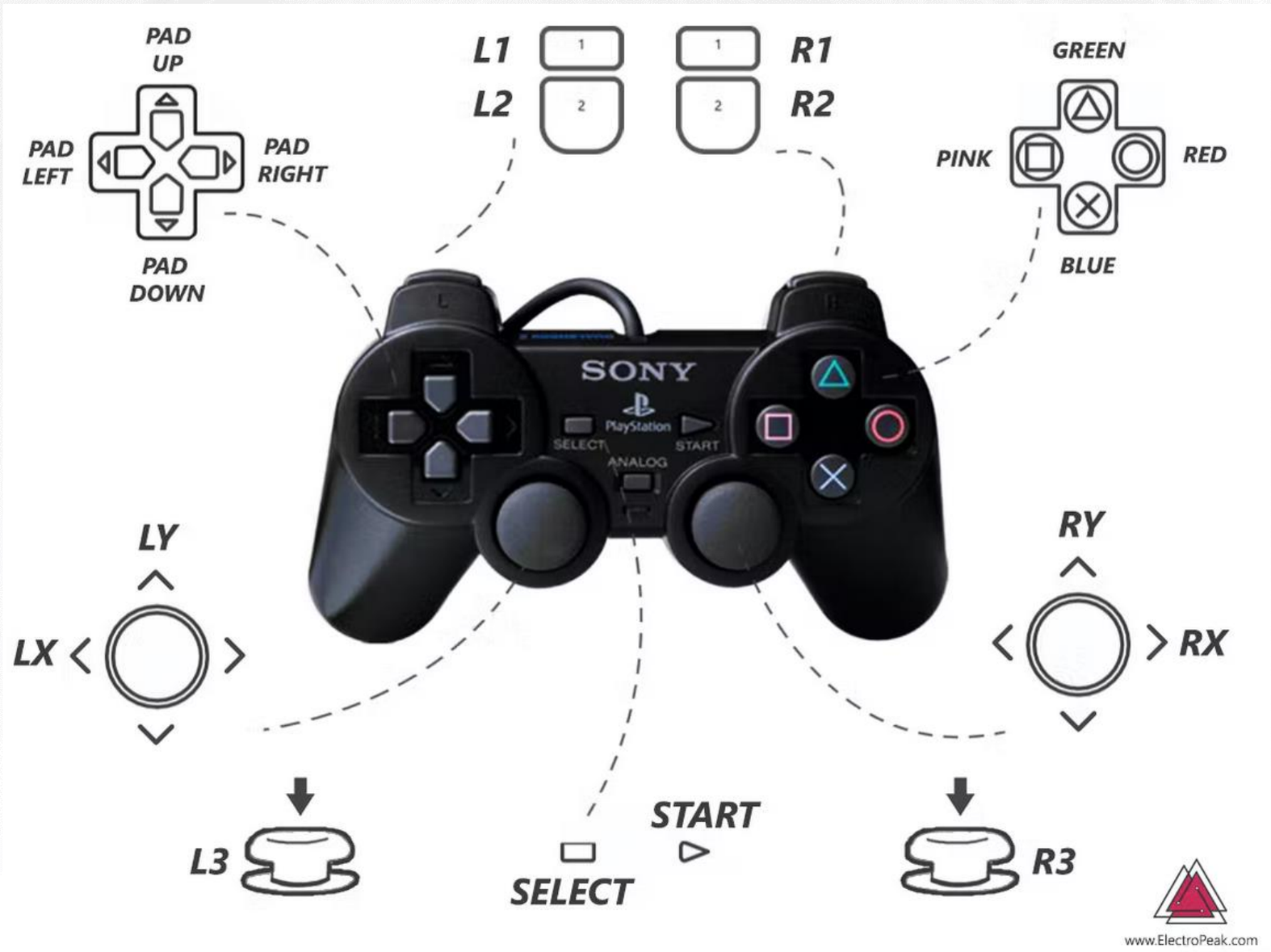
Tay điều khiển

FPT Education
TRƯỜNG ĐẠI HỌC FPT

MAKER VIỆT

Lập trình tay điều khiển PS2

Bản đồ nút bấm tay điều khiển PS2



Lập trình tay điều khiển PS2

Bản đồ nút bấm tay điều khiển PS2

FPT Education
TRƯỜNG ĐẠI HỌC FPT
MAKER VIỆT

Lập trình tay điều khiển PS2

***Bắt đầu với ví dụ đọc các nút bẩm gửi từ tay cầm (*PS2X_Example_VIA_Makerbot_ESP32)**

Khai báo thư viện cho tay cầm:

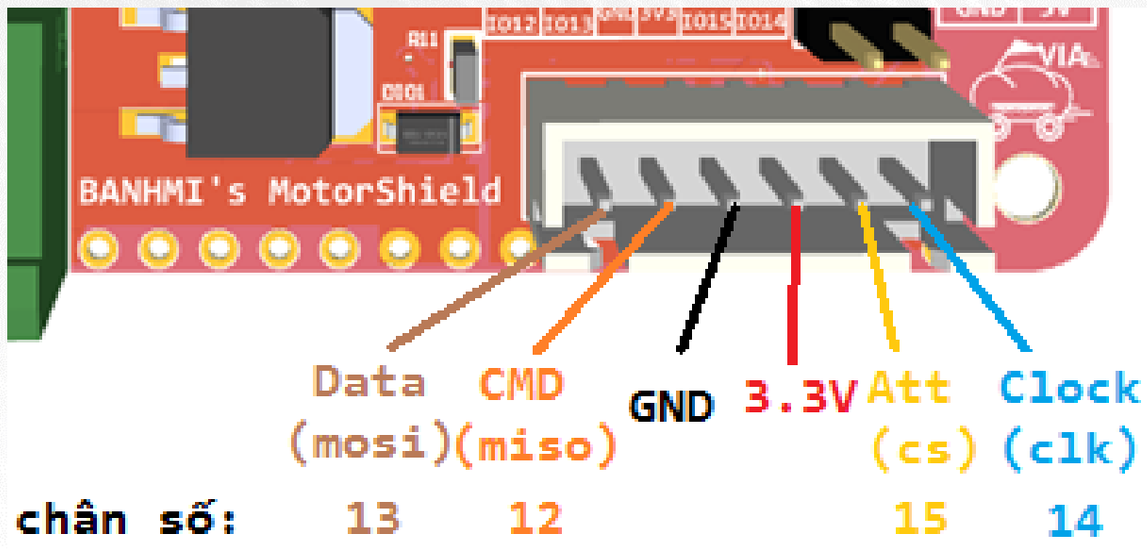
```
#include <PS2X_lib.h> // Khai báo thư viện
```

Định nghĩa các chân điều khiển

```
#define PS2_DAT 12 // MISO
#define PS2_CMD 13 // MOSI
#define PS2_SEL 15 // SS
#define PS2_CLK 14 // SLK
```

Khởi tạo class của thư viện

```
PS2X ps2x; // khởi tạo class PS2x
```



Lập trình tay điều khiển PS2

***Bắt đầu với ví dụ đọc các nút bấm gửi từ tay cầm (*PS2X_Example_VIA_Makerbot_ESP32)**

Bắt đầu kết nối trong hàm setup()

Khởi tạo Serial monitor với tốc độ 115200

```
Serial.begin(115200);
```

Kết nối với tay cầm bằng hàm ps2x.config_gamepad, thử kết nối lại trong vòng 10 lần nếu quá 10 lần không kết nối được với tay cầm thì sẽ dừng lại

```
int error = -1;
for (int i = 0; i < 10; i++) // thử kết nối với tay cầm ps2 trong 10 lần
{
  delay(1000); // đợi 1 giây
  // cài đặt chân và các chế độ: GamePad
  error = ps2x.config_gamepad(PS2_CLK, PS2_CMD, PS2_SEL, PS2_DAT, pressures, rumble);
  Serial.print(".");
  if(!error) //kiểm tra nếu tay cầm đã kết nối thành công
      break; // thoát khỏi vòng lặp
}
```

Lập trình tay điều khiển PS2

***Bắt đầu với ví dụ đọc các nút bẩm gửi từ tay cầm (*PS2X_Example_VIA_Makerbot_ESP32)**

Đọc tay điều khiển trong hàm hàm loop()

Cập nhật các giá trị của tay điều khiển

```
ps2x.read_gamepad(false, false); // gọi hàm để đọc tay điều khiển
```

Đọc nút bấm với nhiều cách khác nhau, đọc theo kiểu digital, True khi nhấn, False khi không bám:

```
// các trả về giá trị TRUE (1) khi nút được giữ
if (ps2x.Button(PSB_START)) // nếu nút Start được giữ, in ra Serial monitor
      Serial.println("Start is being held");
```

Đọc nút bẩm theo kiểu Analog, lấy giá trị độ lớn, ấn mạnh hay nhẹ:

```
if (ps2x.Button(PSB_PAD_UP)) // tương tự như trên kiểm tra nút Lên (PAD UP)
{
   Serial.print("Up held this hard: ");
   Serial.println(ps2x.Analog(PSAB_PAD_UP), DEC); // đọc giá trị analog ở nút
   này, xem nút này được bấm mạnh hay nhẹ
}
```

Lập trình tay điều khiển PS2

***Bắt đầu với ví dụ đọc các nút bấm gửi từ tay cầm (*PS2X_Example_VIA_Makerbot_ESP32)**

Đọc nút bấm theo sự thay đổi trạng thái (bật, bật hoặc tắt, tắt)

```
if (ps2x.ButtonPressed(PSB_CIRCLE)) // Trả về giá trị TRUE khi nút được ấn (từ tắt sang bật)
   Serial.println("○ just pressed");
if (ps2x.NewButtonState(PSB_CROSS)) // Trả về giá trị TRUE khi nút được thay đổi trạng thái (bật sang tắt, hoặc tắt sang bật)
   Serial.println("× just changed");
if (ps2x.ButtonReleased(PSB_SQUARE)) // Trả về giá trị TRUE khi nút được thả ra (từ bật sang tắt)
   Serial.println("□ just released");
```

Đọc nhiều nút bấm theo sự thay đổi trạng thái

```
if (ps2x.NewButtonState()) { // Trả về giá trị TRUE khi nút được thay đổi
 trạng thái (bật sang tắt, hoặc tắt sang bật)
   if (ps2x.Button(PSB_L3)) Serial.println("L3 pressed");
  if (ps2x.Button(PSB_R3)) Serial.println("R3 pressed");
  if (ps2x.Button(PSB_L2)) Serial.println("L2 pressed");
}
```

Lập trình tay điều khiển PS2

***Bắt đầu với ví dụ đọc các nút bẩm gửi từ tay cầm (*PS2X_Example_VIA_Makerbot_ESP32)**

Đọc giá trị joystick

```
Serial.print("Stick Values:");
Serial.print(ps2x.Analog(PSS_LY)); // đọc trục Y của joystick bên trái.
Serial.print(","); Serial.print(ps2x.Analog(PSS_LX), DEC); Serial.print(",");
Serial.print(ps2x.Analog(PSS_RY), DEC); Serial.print(",");
Serial.println(ps2x.Analog(PSS_RX), DEC); }
```

Tay điều khiển

Lập trình tay điều khiển PS2

TRƯỜNG ĐẠI HỌC FPT

MAKER VIỆT

Key	Function	Digital/Analog
PSB_SELECT	OK	Digital
PSB_START	OK	Digital
PSB_PAD_UP	UP	Analog
PSB_PAD_DOWN	DOWN	Analog
PSB_PAD_LEFT	LEFT	Analog
PSB_PAD_RIGHT	RIGHT	Analog
PSB_BLUE	X	Analog
PSB_GREEN	Triangle	Analog
PSB_PINK	Square	Analog
PSB_RED	Circle	Analog
PSB_L3	L3	Digital
PSB_R3	R3	Digital
PSB_L2	L2	Analog
PSB_R2	R2	Analog
PSB_L1	L1	Analog
PSB_R1	R1	Analog
PSB_RX	Joystick right x	Analog
PSB_RY	Joystick right y	Analog
PSB_LX	Joystick left x	Analog
PSB_LY	Joystick left y	Analog

FPT Education

TRƯỜNG ĐẠI HỌC FPT

MAKER VIỆT

QUIZZ time !!!

QUIZZ time !!!

TRƯỜNG ĐẠI HỌC FPT

MAKER VIỆT

Bài tập thực hành

Sử dụng Tinkercad circuit hãy viết 1 chương trình sử dụng Arduino uno mô phỏng sao cho in ra Serial monitor dòng chữ sau :

" hello Via \n " in ra lúc bắt đầu và xuất hiện lại sau mỗi 3 giây

" VRC 2022\n" in ra lúc bắt đầu và xuất hiện lại sau mỗi 2 giây

Ví dụ :

00:00 Hello VIA

VRC2022

.....

00:02 VRC2022

00:03 Hello VIA

00:04 VRC2022

00:06 Hello VIA