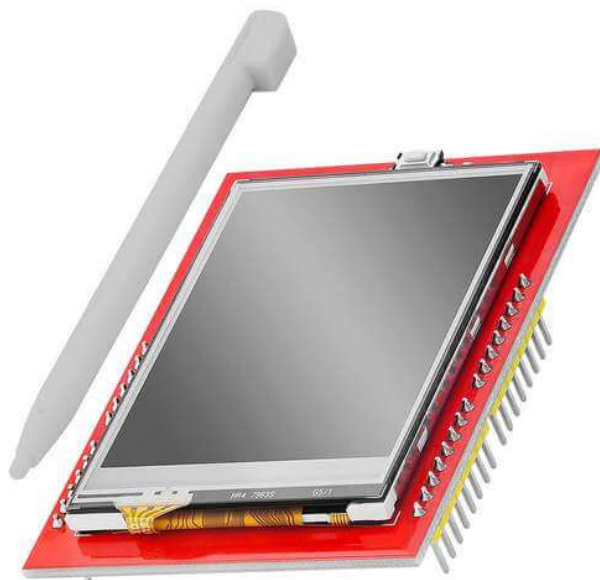


# Bộ điều khiển màn hình cảm ứng XPT2046

## Thành phần / Trung tâm

Thành phần xpt2046 cho phép sử dụng bộ điều khiển màn hình cảm ứng dựa trên chip XPT2046 (biểu [dữ liệu](#), [AZ-Delivery](#)) với ESPHome. Nhiêu màn hình LCD giá rẻ có chứa bộ điều khiển này. [SPI](#) là bắt buộc phải được thiết lập trong cấu hình của bạn để cảm biến này hoạt động.



Bộ điều khiển màn hình cảm ứng XPT2046

# Example configuration entry

xpt2046:

```
id: touchscreen
cs_pin: 17
irq_pin: 16
update_interval: 50ms
report_interval: 1s
threshold: 400
dimension_x: 240
dimension_y: 320
calibration_x_min: 3860
calibration_x_max: 280
calibration_y_min: 340
calibration_y_max: 3860
swap_x_y: false
```

binary\_sensor:

```
- platform: xpt2046
  xpt2046_id: touchscreen
  id: touch_key0
  x_min: 80
  x_max: 160
  y_min: 106
```

```
y_max: 212
on_state:
  - lambda: 'ESP_LOGI("main", "key0: %s", (x ? "touch" : "release"))';'
```

## Các biế̃n cẫu hình:

Cẫu hình được tạo thành từ hai phầ̃n: Thành phầ̃n màn hình cảm ứng và các cảm biế̃n nhĩ phân riề̃ng lē̃ tùy chọ̃n.

Cẫu hình cơ sở:

- **id** ( *Tùy chọ̃n* , [ID](#) ): Đặt ID của cảm biế̃n này.
- **cs\_pin** ( *Tùy chọ̃n* , [Lược đồ̃](#) chân): Chân chọ̃n chip. Thường được đánh dấ̃u `T_CS` trên bảng.
- **irq\_pin** ( *Tùy chọ̃n* , [Số đồ̃](#) chân): Chân phát hiệ̃n cảm ứng. Thường được đánh dấ̃u `T_IRQ` trên bảng. Nế̃u không được chỉ định, thành phầ̃n sẽ sử dụng tĩ̃nh năng thẳ̃m dò qua SPI.
- **update\_interval** ( *Tùy chọ̃n* , [Thời gian](#) ): Khoảng thời gian để kiể̃m tra cảm biế̃n. Nế̃u `irq_pin` được chỉ định, cảm ứng sẽ được phát hiệ̃n gầ̃n như ngay lập tức và cài đặt này sẽ chỉ được sử dụng để phát hiệ̃n phát hành. Mặc định là 50ms.
- **report\_interval** ( *Tùy chọ̃n* , [Thời gian](#) ): Khoảng thời gian báo cáo tọa độ định kỳ trong khi chạm vào màn hình cảm ứng. Mặc định là never.
- **ngưỡ̃ng** ( *Tùy chọ̃n* , int): Giá trị để phát hiệ̃n chạm hoặc nhả. Mặc định là 400.
- **Chiế̃u\_x** ( *Tùy chọ̃n* , int): Kích thước của màn hình theo hướng ngang. Thường tĩ̃nh bằ̃ng pixel nhưng một tỷ lệ phầ̃n trăm cũng có thể hữu ĩ̃ch. Mặc định là 100.
- **kích thước\_y** ( *Tùy chọ̃n* , int): Kích thước của màn hình theo hướng dọc. Thường tĩ̃nh bằ̃ng pixel nhưng một tỷ lệ phầ̃n trăm cũng có thể hữu ĩ̃ch. Mặc định là 100.
- **calibration\_x\_min** ( *Tùy chọ̃n* , int): Giá trị thô tương ứng với cạnh trái (hoặc trên cùng nế̃u `swap_x_y` được chỉ định) của màn hình. Xem phầ̃n [Hiệ̃u chỉnh](#) để biế̃t quy trình hiệ̃u chỉnh màn hình cảm ứng. Mặc định là 0.
- **calibration\_x\_max** ( *Tùy chọ̃n* , int): Giá trị thô tương ứng với cạnh bên phải (hoặc dưới cùng nế̃u `swap_x_y` được chỉ định) của màn hình. Mặc định là 4095.
- **calibration\_y\_min** ( *Tùy chọ̃n* , int): Giá trị thô tương ứng với cạnh trên cùng (hoặc bên trái nế̃u `swap_x_y` được chỉ định) của màn hình. Mặc định là 0.
- **calibration\_y\_max** ( *Tùy chọ̃n* , int): Giá trị thô tương ứng với cạnh dưới cùng (hoặc bên phải nế̃u `swap_x_y` được chỉ định) của màn hình. Mặc định là 4095.
- **swap\_x\_y** ( *Tùy chọ̃n* , boolean): Nế̃u đúng, trục x và y được hoán đổi. Mặc định là false.
- **on\_state** ( *Tùy chọ̃n* , [Tự động hóa](#) ): Tự động hóa để thực hiệ̃n khi nhấ̃n hoặc thả màn hình cảm ứng. Xem [Hành động on\\_state](#) .

## on\_stateHành động

Tĩ̃nh năng tự động hóa này sẽ được kích hoạt khi màn hình cảm ứng XPT2046 phát hiệ̃n một lầ̃n chạm, một lầ̃n nhả hoặc định kỳ mỗi lầ̃n `report_interval` chạm vào.

Trình kích hoạt này cung cẫp ba đố̃ĩ số̃ : `x` và `y` thuộc loại int và chỉ định tọa độ của lầ̃n chạm và một bool `touched` chỉ định xem có phát hiệ̃n được một lầ̃n chạm hoặc thả hay không.

Ngoài các tọa độ, trạng thái cảm ứng và các giá trị thô cẫn thiế̃t cho việc hiệ̃u chuẩn có thể được truy cập dưới dạng các biế̃n thành việ̃n.

Đoạn mã sau

```
xpt2046:
  on_state:
    - lambda: |-
        ESP_LOGI("main", "args x=%d, y=%d, touched=%s", x, y, (touched ? "touch" : "release")
        ESP_LOGI("main", "member x=%d, y=%d, touched=%d, x_raw=%d, y_raw=%d, z_raw=%d",
            id(touchscreen).x,
            id(touchscreen).y,
            (int) id(touchscreen).touched,
            id(touchscreen).x_raw,
            id(touchscreen).y_raw,
            id(touchscreen).z_raw
        );
```

sản xuất

```
[20:17:37][I][main:065]: args x=145, y=261, touched=touch
[20:17:37][I][main:073]: member x=145, y=261, touched=1, x_raw=1686, y_raw=3218, z_raw=424
[20:17:37][I][main:065]: args x=145, y=261, touched=release
[20:17:37][I][main:073]: member x=145, y=261, touched=0, x_raw=0, y_raw=0, z_raw=0
```

## Cảm biến nhị phân

Cảm xpt2046biến nhị phân cho phép bạn thiết lập các khu vực trên màn hình cảm ứng dưới dạng các nút ảo. Đầu tiên, thiết lập [Thành phần / Trung tâm](#) và sau đó sử dụng nền tảng cảm biến nhị phân này để tạo các cảm biến nhị phân riêng lẻ cho từng nút ảo.

- **name** ( *Tùy chọn* , chuỗi): Tên cho cảm biến nhị phân.
- **id** ( *Tùy chọn* , [ID](#) ): Chỉ định thủ công ID được sử dụng để tạo mã.
- **xpt2046\_id** ( *Tùy chọn* , [ID](#) ): Chỉ định ID của thành phần mà cảm biến là một phần. Hữu ích khi bạn có nhiều màn hình cảm ứng.
- **x\_min** ( **Bắt buộc** , int): Tọa độ bên trái của vùng màn hình được phát hiện là nút ảo.
- **x\_max** ( **Bắt buộc** , int): Tọa độ bên phải của vùng màn hình được phát hiện là nút ảo.
- **y\_min** ( **Bắt buộc** , int): Tọa độ trên cùng của vùng màn hình được phát hiện là nút ảo.
- **y\_max** ( **Bắt buộc** , int): Tọa độ dưới cùng của vùng màn hình được phát hiện là nút ảo.
- Tất cả các tùy chọn khác từ [Cảm biến nhị phân](#) .

## Hiệu chuẩn

Để khớp điểm chạm với tọa độ hiển thị, màn hình cảm ứng phải được hiệu chỉnh. Thành phần XPT2046 trả về các giá trị thô trong phạm vi 0 đến 4095. Các giá trị thô đó có sẵn dưới dạng biến `x_raw` và `y_raw` biến thành viên và ví dụ như viết chúng ra như trong ví dụ [về Hành động on\\_state](#) . Mục tiêu của hiệu chuẩn là xác định các giá trị thô tương ứng với các cạnh của màn hình.

Việc hiệu chuẩn giả định màn hình được định hướng theo cách mà bạn sẽ sử dụng nó, tức là thành phần [Công cụ kết xuất màn hình](#) của bạn phải có tọa độ logic [0,0] ở trên cùng bên trái. Đặt các kích thước là `dimension_x` và `dimension_y` và cũng không đặt bất kỳ giá trị hiệu chuẩn nào `swap_x_y`.

```
# Touchscreen
xpt2046:
  id: touchscreen
  cs_pin: 17
  irq_pin: 16
```

```

dimension_x: 240
dimension_y: 320
on_state:
  - lambda: |-
      if (touched)
        ESP_LOGI("cal", "x=%d, y=%d, x_raw=%d, y_raw=%d",
          id(touchscreen).x,
          id(touchscreen).y,
          id(touchscreen).x_raw,
          id(touchscreen).y_raw
        );

```

Lấy bút stylus hoặc một vật tương tự, chạy dự án và chạm vào các góc của màn hình ở các pixel cạnh. Lặp lại nhiều lần và lưu ý các giá trị thô x và y tối thiểu và lớn nhất.

```

... top left ...
[21:07:48][I][cal:071]: x=217, y=34, x_raw=3718, y_raw=445
[21:07:49][I][cal:071]: x=222, y=32, x_raw=3804, y_raw=419
... top right ...
[21:07:52][I][cal:071]: x=19, y=36, x_raw=334, y_raw=370
[21:07:52][I][cal:071]: x=22, y=35, x_raw=386, y_raw=347
... bottom left ...
[21:08:00][I][cal:071]: x=224, y=299, x_raw=3836, y_raw=3835
[21:08:00][I][cal:071]: x=225, y=303, x_raw=3848, y_raw=3878
[21:08:01][I][cal:071]: x=223, y=299, x_raw=3807, y_raw=3829
... bottom right ...
[21:08:11][I][cal:071]: x=16, y=299, x_raw=281, y_raw=3839
[21:08:12][I][cal:071]: x=19, y=302, x_raw=328, y_raw=3866
[21:08:13][I][cal:071]: x=20, y=296, x_raw=358, y_raw=3799

```

Điều đó có nghĩa là x nguyên tối thiểu là 281, tối đa 3848, y tối thiểu 347 và tối đa 3878.

Xác định giá trị thô nào là hướng x của màn hình và giá trị nào là hướng y. Trong trường hợp của chúng ta, việc di chuyển sang phải làm giảm giá trị nguyên x và đi xuống làm tăng giá trị y để các trục khớp nhau và chúng ta *không* cần sử dụng `swap_x_y`. Nếu x thô là y của màn hình, hãy sử dụng `.swap_x_y = true`

Nếu một trong các tọa độ đi theo hướng "sai", nó cần phải được đảo ngược. Việc đảo ngược được thực hiện bằng cách hoán đổi các giá trị nhỏ nhất và lớn nhất. Trong trường hợp của chúng ta, hướng nằm ngang được biểu thị bởi nguyên x (không hoán đổi) bị đảo ngược nên `calibration_x_min` cần phải lớn hơn `calibration_x_max`. Hướng thẳng đứng là được. Do đó, cấu hình sẽ là

```

xpt2046:
  calibration_x_min: 3848
  calibration_x_max: 281
  calibration_y_min: 347
  calibration_y_max: 3878

```

Biên dịch, chạy và nhấp lại vào các cạnh. X và y bây giờ sẽ khớp với tọa độ của màn hình.

```

[21:32:34][I][cal:071]: x=7, y=6, x_raw=3755, y_raw=407
[21:32:37][I][cal:071]: x=237, y=4, x_raw=313, y_raw=385
[21:32:43][I][cal:071]: x=239, y=318, x_raw=284, y_raw=3845
[21:33:05][I][cal:071]: x=2, y=313, x_raw=3821, y_raw=3793

```

Lưu ý rằng màn hình cảm ứng không cực kỳ chính xác và có thể có các lỗi phi tuyến tính hoặc các lỗi tương tự, vì vậy đừng mong đợi độ chính xác hoàn hảo đến từng pixel. Bạn có thể xác minh điểm tiếp xúc bằng lambda hiển thị tương tự như sau.

```
display:  
  - platform: ili9341  
    lambda: |-  
      it.fill(BLACK);  
      if (id(touchscreen).touched)  
        it.filled_circle(id(touchscreen).x, id(touchscreen).y, 10, RED);
```

Nói một cách chính xác, thành phần thực hiện như sau

- đọc x và y thô và chuẩn hóa nó bằng cách sử dụng các giá trị tối thiểu và tối đa (không đảo ngược)
- hoán đổi x và y nếu cần
- đảo ngược nếu cần
- tỷ lệ với kích thước hiển thị

## Xem thêm

- [Bộ lọc cảm biến nhị phân](#)
- [Tham chiếu API](#)
- [Thư viện XPT2046 của Paul Stoffregen](#)
- [Chỉnh sửa trang này trên GitHub](#)