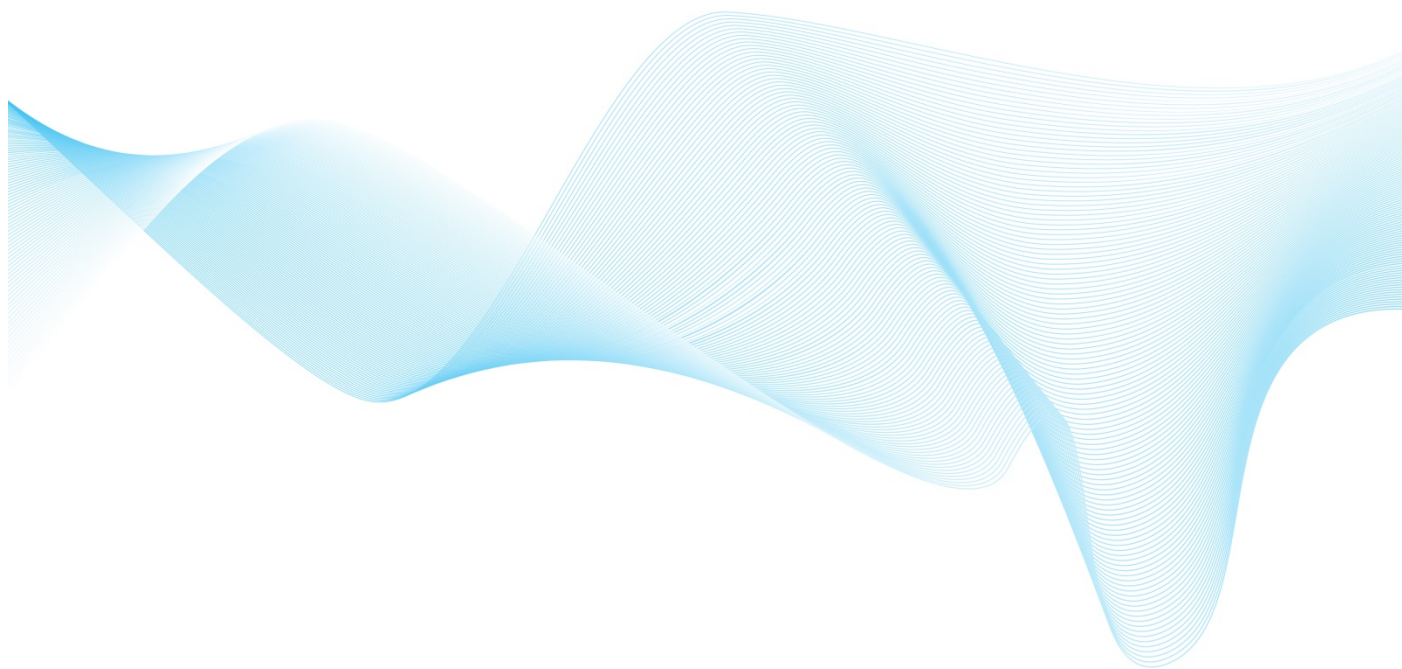




Yottacontrol Co.

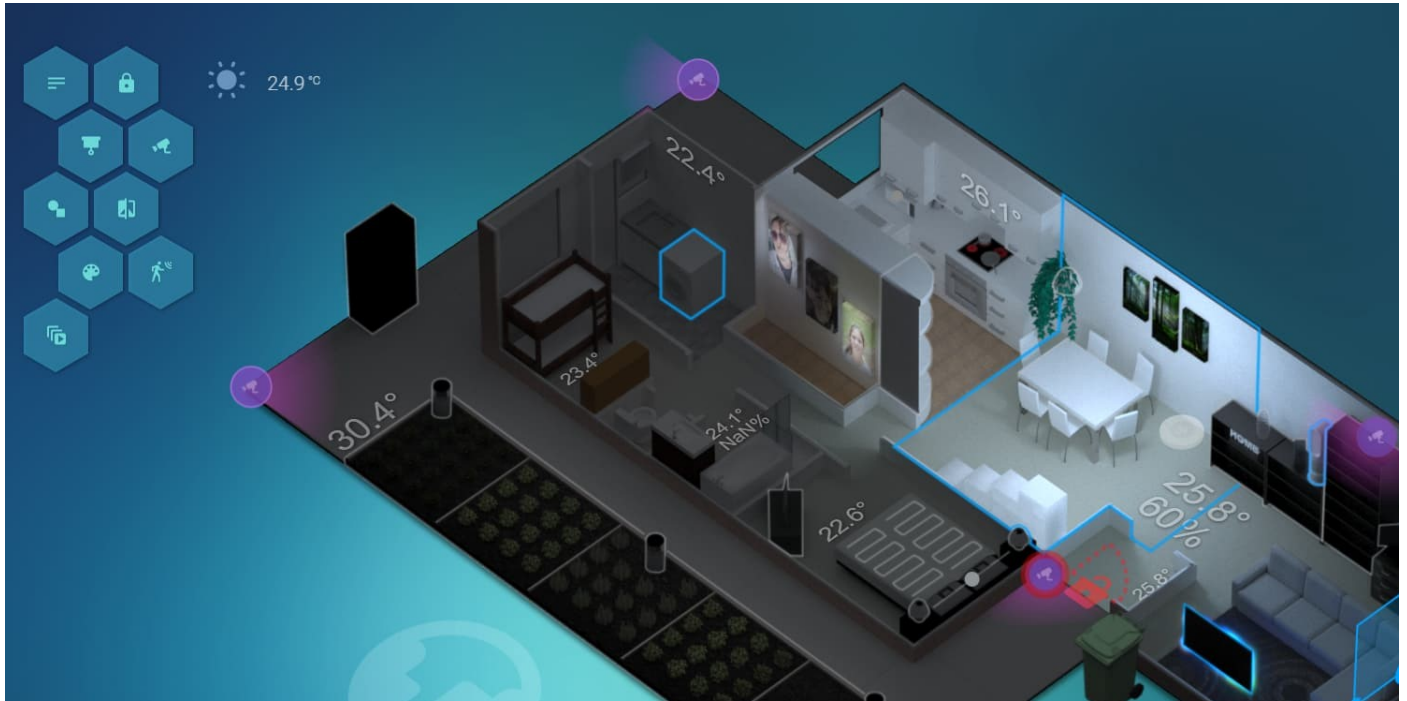
HIPC Example
Honey CombMenu



在 Home Assistant Honeycomb Menu 範例

Honeycomb 選單是一個 Home Assistant 模組（並非卡片），可應用於任何 Lovelace 卡片。當卡片上定義的操作啟動該模組時，它會顯示一個圓形排列的 Honeycomb 按鈕列表，並可選配 XY 觸控板，使與 Lovelace 的交互更加流暢。此模組的設計靈感來自@RomRider 的按鈕卡。

此模組使用層級覆蓋來控制蜂窩選項和子選項，因此您可以定義一個模板化的選項版本，然後定義另一組選項，這些選項將合併/級聯。



如何安裝

下載模組

將檔案放入 config/wwwHome Assistant 安裝路徑中。

如果你跟我一樣有強迫症，可以把檔案放在子目錄裡😊，記得要把資源 URL 指向子路徑。
將資源新增至 Lovelace 配置。有兩種方法可以做到這一點。

yaml- 找到你的 lovelace.yaml 文件，然後將以下內容放入 resources 資料夾中，例如：

resources:

```
- url: /local/{path-to-module}.js  
  type: module
```

Web Interface- (Home Assistant V0.108+):

轉到配置頁面，然後打開 Lovelace Dashboards

選擇選項 Resources 卡

點擊+右下角的（新增）按鈕

在 URL 欄位中，新增模組 js 檔案路徑（需要將 Home Assistant 配置中的 `/www/` 路徑替換為 `/local/`）。例如，已儲存檔案的路徑 homeassistant_path/config/www/module.js 應/local/module.js 作為 URL 路徑。

請確保資源類型保留為 Javascript 模組

如何使用

在 fire-dom-event 操作中，您可以呼叫 honeycomb 選單，同時將配置選項傳遞給 honeycomb_menu

屬性。

```
- type: vertical-stack
  cards:
  - type: button
    entity: light.kitchen
    hold_action:
      action: fire-dom-event
    honeycomb_menu:
      template: light
      autoclose: false
      active: true
    ...
```

honeycomb:選項

選項 價值觀 預設 細節

實體 any:entity_id card:entity 這將對指定的實體 (entity_id) 呼叫操作。如果省略，則將使用卡片實體。

按鈕 list[0-5]: 按鈕 | skip | break null | template_buttons 按鈕就是你的蜂巢 😊。最多可定義 6 個按鈕。* 注意：清單索引從 0 開始 0。與 template_buttonskip 匹配的索引將被覆蓋。在索引上使用字串將使用 template_button 該索引對應的預設設置，而使用字串 break 則會停用該蜂巢位置，無論 template_button 該索引的值是什麼。

積極的 true | false | template false 將其設為 true 將根據其所指派的實體套用活動樣式。您也可以選擇使用範本並傳回布林值。請參閱“模板”。

自動關閉 true | false true 按下按鈕後關閉選單

聲音的 any:url_path null 指向一個音訊文件，點擊按鈕後該文件將播放。

xy_pad XYPad null 這樣就可以在蜂巢結構的中間添加一個 xy 引腳，該引腳可以根據 x 或 y 值執行相應的服務。

尺寸 int:px 225 蜂窩狀選單的尺寸(以像素為單位)。每個按鈕項都會隨著尺寸的增加而增加。

間距 int:px 2 這將為每個蜂窩狀元素設定內邊距(以像素為單位)。

變數 object null 可以透過父模板和模板程式碼存取變數，其中 `variables.foo<variableName>foo` 是變數名稱。請參閱範例

動畫速度 int:ms 100 調整選單顯示動畫的速度，每毫秒增加一個按鈕的顯示時間...因此，50 毫秒 * 5 = 250 毫秒，總時間(第一個按鈕立即顯示)。

Button 選項

選項 價值觀 預設 細節

類型 any:card custom:button-card 按鈕要使用的基礎卡。請務必將基礎卡片的高度設定為 100%，否則可能無法正確顯示。

積極的 true | false honeycomb:active 覆蓋此按鈕項目的蜂窩狀激活屬性

展示 true | false true 是否顯示此按鈕。如果父範本中此位置已有啟動的按鈕，則會顯示父範本中的按鈕。模板可用於傳回布林值。

自動關閉 `true | false` `honeycomb:autoclose` 覆蓋此按鈕的蜂窩自動關閉屬性

聲音的 `any:url_path` `honeycomb:audio` 覆蓋此按鈕的蜂窩音訊屬性

實體 `any:entity_id` `honeycomb:entity` 您可以定義此按鈕的目標實體。如果省略，則預設使用蜂巢實體。可以透過存取實體的變數在模板中使用它。`entity`

位置 `int null` 按鈕的放置位置索引。這有助於覆蓋模板按鈕。

圖示 `any:icon null` 此處新增僅用於引用 `custom:button-card`，以便您可以顯示項目的圖示。

顏色 `any:css_color var(--honeycomb-menu-icon-color)` 圖示或背景顏色（取決於 `custom:button-card` 配置）。保留預設值則由主題處理顏色。

顯示名稱 `true | false` `false` 僅適用於支援此選項的卡片

還有其他選擇嗎？ `Button:type` - - -

XYPad 選項

`x` 和 `y` 屬性具有相同的選項。如果省略 `x` 或 `y` 屬性，則填滿將僅沿著定義的方向進行。此控制板可用於調整燈光亮度、旋轉色調、打開和關閉百葉窗或捲簾門等。

選項 價值觀 預設 細節

重複 `int:ms | false` `false` 如果 `xy` 引腳移動但處於空閒狀態，則 `repeat` 將繼續呼叫該服務；否則，僅當 `xy` 引腳移動時才會呼叫該服務。

發布 `true | false` `false` 僅當釋放 `xy` 引腳並提供其當前 `xy` 值時才呼叫服務。

`x` XY 配置 `null` 屬性和值如下所示。 `*null` 將停用 `x` 軸移動

是 XY 配置 `null` 屬性和值如下所示。 `*null` 將停用 `y` 軸移動。

XY 配置

選項 價值觀 預設 細節

倒置 `true | false` `false` `x` 或 `y` 軸會交換正負值，因此向上移動 `xy` 軸的接腳會得到正值，向下移動就會得到負值。

服務 `any:service` `null` 要呼叫的服務例如 `light.turn_on`。如果省略此值，則球形銷將不會對該軸產生任何影響。

數據或服務數據 `dict null` 請以字典/物件的形式提供任何服務資料。此屬性將透過模板系統進行處理，從而允許存取變數和 JavaScript。請參閱模板部分。

主題樣式和預設設定

將以下樣式屬性新增至主題 `.yaml` 檔案將覆蓋預設值。

```
/* Styles selector is used as a placeholder for syntax highlighting */
```

```
styles {
```

```
  --honeycomb-menu-icon-color: var(--paper-item-icon-color);
```

```
  --honeycomb-menu-icon-active-color: var(--paper-item-icon-active-color);
```

```
  --honeycomb-menu-background-color: var(--paper-card-background-color);
```

```
  --honeycomb-menu-active-background-color: var(--paper-card-active-background-color,
var(--paper-card-background-color));
```

```
  --honeycomb-menu-disabled: #9a9a9a6e;
```

```
}
```

範本

所有可模板化的屬性都將被解析為模板字面量。如果您需要完整的 JavaScript 功能，則 HCJS: 必須在

開頭聲明 `JavaScript` (也稱為 **Honeycomb JavaScript**)，以表示該字串將作為 **JavaScript** 運行 (請記住返回您的值!)。如果使用此功能，則可以透過物件存取已定義變數的配置 **variables** 以及 **xypad** 資料。

目前所有 **XYConfig:data** 屬性和部分配置選項均支援模板功能。範本功能提供了靈活性，並可根據 **XY** 座標軸位置或配置選項提供相應的值。

僅包含單字 “entity” 的屬性將被轉換為該 **honeycomb:entity** 值。

xypad 會將下列內容傳遞給變數物件。**variables**

x: Pixels from the x center position

y: Pixels from the y center position

x_percentage: Percentage of the x position (-10 to 10)

y_percentage: Percentage of the y position (-10 to 10)

HCJS: return `variables.x_percentage / 10;`

或透過模板字面量 (無需返回)

`${variables.x_percentage / 10}`

請注意，某些服務資料需要整數類型，而您可能發送的是浮點數，如果需要，您需要將其轉換為整數類型。

範例

注意：以下服務 **light.relative_brightnesss** 並非 **Home Assistant** 的一部分，而是我自訂的服務，用於相對調整燈光亮度。您可以使用該 **light.turn_on** 服務，並結合 **JavaScript** 模板解析器進行一些計算，來實現相同的效果。

...

action: fire-dom-event

honeycomb_menu:

entity: light.kitchen

xy_pad:

x:

invert: true

service: light.relative_color

data:

The word entity will become light.kitchen

entity_id: entity

hue: `'${variables.x / 18 * variables.x_percentage}'`

y:

service: light.relative_brightnesss

data:

```
# We can define another entity like normal
entity_id: light.bathroom
brightness: '${variables.y_percentage}'
percentage: true
```

例如，可以使用模板化的配置選項來確定風扇狀態，並在風扇目前處於該狀態時將按鈕設為啟動狀態，例如…由於 Honeycomb 選單項目預設使用 `custom:buton-card` 作為按鈕，您可以預設使用其 JS 語法，也可以使用 HCJS：

...

```
action: fire-dom-event
```

```
honeycomb_menu:
```

```
entity: fan.master_bedroom
```

```
buttons:
```

```
- icon: 'mdi:information-variant'
```

```
tap_action:
```

```
action: more-info
```

```
- icon: 'mdi:fan-speed-1'
```

```
active: '[[[ return entity.attributes.speed == "low" ]]]'
```

```
tap_action:
```

```
action: call-service
```

```
service: fan.set_speed
```

```
service_data:
```

```
entity_id: entity
```

```
speed: low
```

```
- icon: 'mdi:fan-speed-2'
```

```
active: '[[[ return entity.attributes.speed == "medium" ]]]'
```

```
tap_action:
```

```
action: call-service
```

```
service: fan.set_speed
```

```
service_data:
```

```
entity_id: entity
```

```
speed: medium
```

```
- icon: 'mdi:fan-speed-3'
```

```
active: '[[[ return entity.attributes.speed == "high" ]]]'
```

```
tap_action:
```

```
action: call-service
```

```
service: fan.set_speed
```

```
service_data:
```

```
entity_id: entity
```

```
speed: high
```

使用變數的範例

假設你已經為燈光定義了一個蜂巢式選單模板，那麼你可以讓燈光模板對如何渲染做出邏輯條件。

編輯蜂巢式選單模板檔案（*不要將模板化與蜂巢模板混淆）

light:

variables:

timer: null

motion: null

audio: /local/audio/pin-drop.ogg

xy_pad:

repeat: 500

y:

invert: true

service: light.turn_on

data:

entity_id: entity

brightness_step_pct: 'HCJS: return variables.y_percentage / 10;' # or '\${variables.y_percentage /

10}'

buttons:

- icon: 'mdi:information-variant'

tap_action:

action: more-info

- icon: 'mdi:lightbulb'

active: true

- show: 'HCJS: return (variables.motion);' # or '\${variables.motion}'

entity: 'HCJS: return variables.motion;' # or '\${variables.motion}'

icon: 'mdi:motion-sensor'

position: 4

active: true

tap_action:

action: toggle

- show: 'HCJS: return (variables.timer);' # or '\${variables.timer}'

entity: 'HCJS: return variables.timer;' # or '\${variables.timer}'

icon: 'mdi:timer'

position: 5

active: true

tap_action:

action: toggle

以上述範例為例，如果選單繼承了燈光模板，那麼任何將計時器或動作設為變數的選單項目都會顯示為實體屬性等於該變數屬性的值。如果未設定該變量，則不會顯示該選單項，而是顯示該位置的下一個繼承項，或不顯示任何內容。

以下舉例說明如何為上述範例指派選單。

- type: 'custom:button-card'
entity: light.back_yard_garden_leds
hold_action:
 action: fire-dom-event
honeycomb_menu:
 template: light
 variables:
 motion: automation.motion_back_yard_garden_leds

相關影片

<https://www.youtube.com/watch?v=MCNxgb0mrSA&list=PLNaeA0iHcKdQ5T4ZBqRtoYvHaCgIP-c1D&index=2>

<https://www.youtube.com/watch?v=LWYmXQL30Hw&list=PLNaeA0iHcKdQ5T4ZBqRtoYvHaCgIP-c1D&index=3>

<https://www.youtube.com/watch?v=FBcbXH62fBQ&list=PLNaeA0iHcKdQ5T4ZBqRtoYvHaCgIP-c1D&index=4>

<https://www.youtube.com/watch?v=VXhcl0QBdKA&list=PLNaeA0iHcKdQ5T4ZBqRtoYvHaCgIP-c1D&index=5>

<https://www.youtube.com/watch?v=oJ9Yr2dSqUk&list=PLNaeA0iHcKdQ5T4ZBqRtoYvHaCgIP-c1D&index=6>

<https://www.youtube.com/watch?v=e9ePvfNCKK8&list=PLNaeA0iHcKdQ5T4ZBqRtoYvHaCgIP-c1D&index=7>