

Unit Grove to Grove

SKU:U148



Description

Unit Grove to Grove is a **1-in-1-out** GROVE expansion control unit that provides **on/off control** + **current measurement**. On/off control is implemented using a digital signal, and current measurement is an analog signal in the range of 0 ~ 1000mA.

Features

- On/off control: 5V/1A
- Current measurement: 0~1000mA

Includes

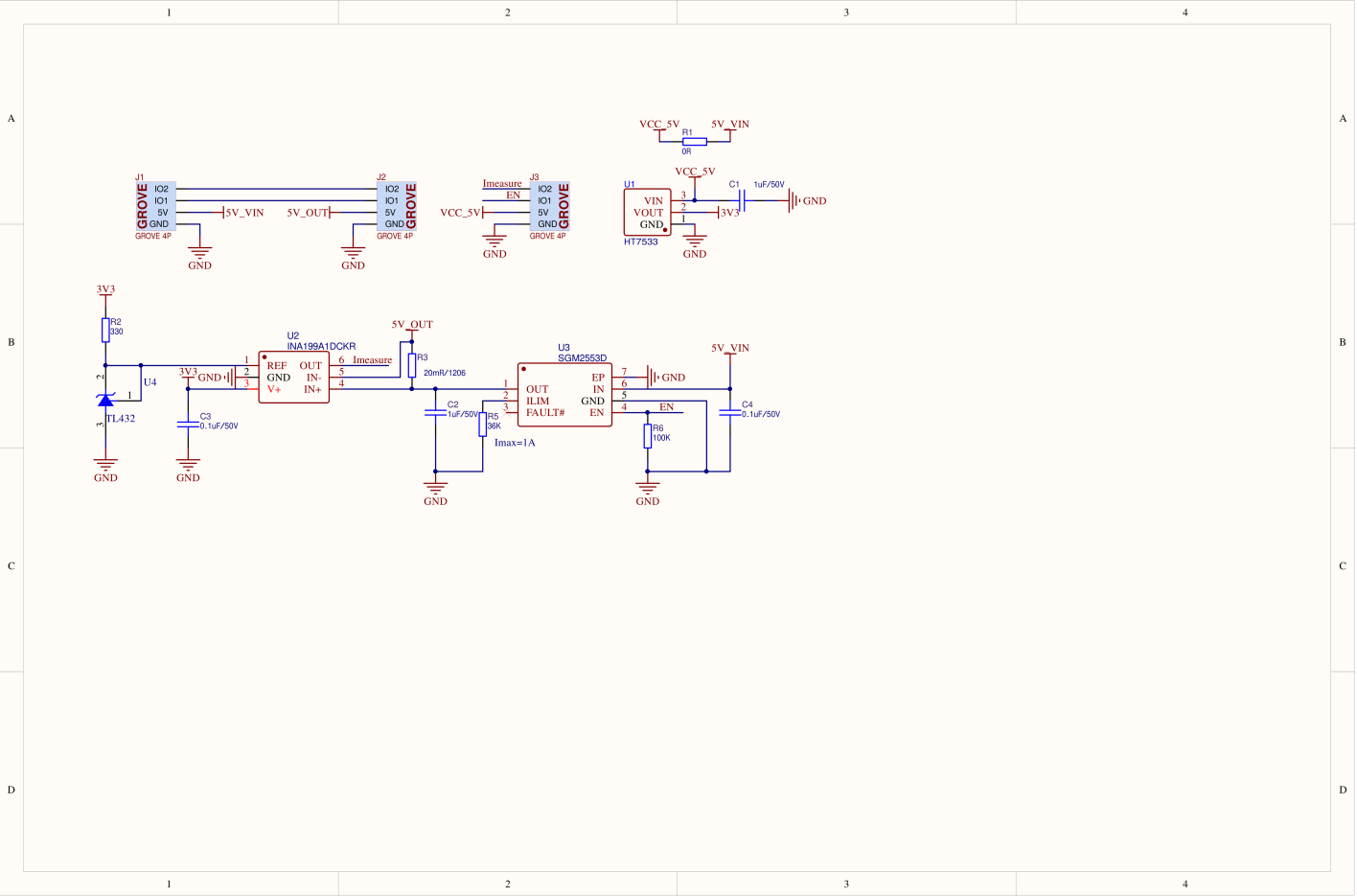
- 1 x Unit Grove to Grove
- 1 x HY2.0-4P Grove Cable (20cm)

Specifications

Specification	Parameter
Operating Voltage	5V
On/Off Current	1000mA
On/Off Loop Voltage	5V
Current Measurement Range	0~1000mA
Product Size	32.0 x 24.0 x 8.1mm
Product Weight	4.9g
Package Size	138.0 x 93.0 x 9.1mm
Gross Weight	10.0g

Schematics

- [Unit Grove to Grove Schematics PDF](#)

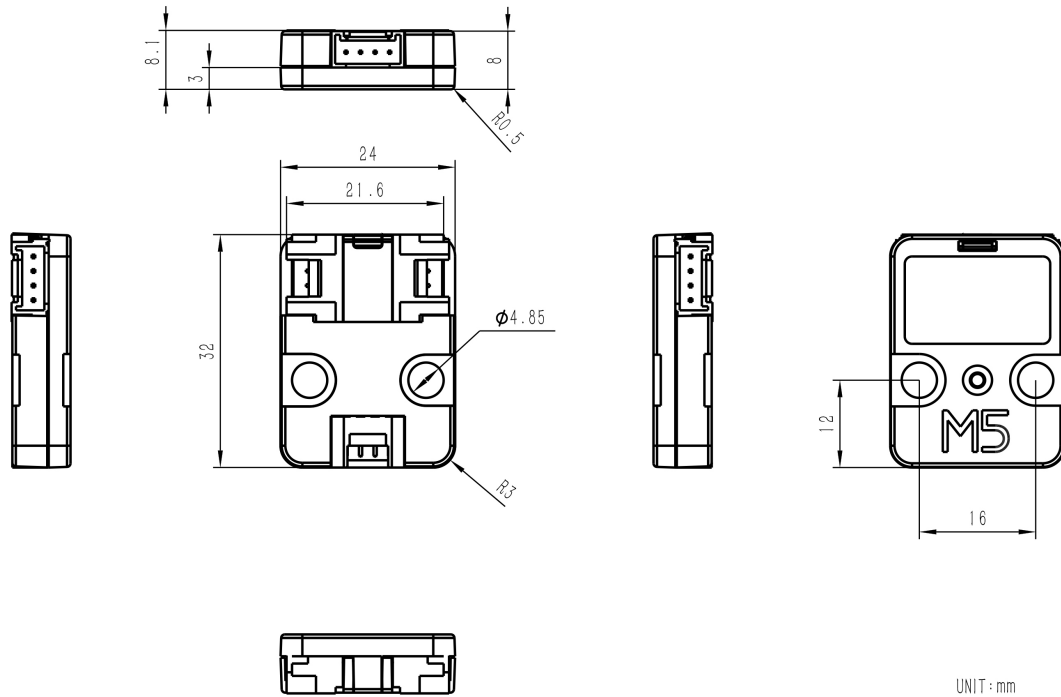


PinMap

Unit Grove to Grove

HY2.0-4P	Black	Red	Yellow	White
PORT.B	GND	5V	PWR_EN	Analog Output

Model Size



Structure

- [Unit Grove to Grove Structure Files](#)

Softwares

Arduino

```
#include <Arduino.h>
#include "driver/adc.h"
#include "esp_adc_cal.h"
#include "math.h"
#include <M5GFX.h>

#define Din_Pin 26
#define Aout_Pin 36
#define groveOn HIGH
#define groveOff LOW

esp_adc_cal_characteristics_t *adc_chars;
float groveVref;

M5GFX display;
M5Canvas canvas(&display);
```

```

int get_battery_voltage(void) {
    uint32_t adc_reading = 0;
    // Multisampling
    for (int i = 0; i < 64; i++) {
        adc_reading += adc1_get_raw((adc1_channel_t)ADC1_CHANNEL_0);
    }
    adc_reading /= 64;
    // Convert adc_reading to voltage in mV
    uint32_t voltage =
        (uint32_t)(esp_adc_cal_raw_to_voltage(adc_reading, adc_chars));
    // Serial.printf("Raw: %dtVoltage: %dmVrn", adc_reading, voltage);
    return voltage;
}

void getVerf() {
    float sampleVari = 1.0f;
    while (sampleVari > 0.20f) {
        sampleVari = 1.0f;
        float sampleVol[100] = {};
        float sampleVolAll = 0;
        groveVref = 0;
        for (int i = 0; i < 100; i++) {
            sampleVol[i] = get_battery_voltage();
            groveVref = groveVref + get_battery_voltage();
            // Serial.println(sampleVol[i]);
        }
        // Serial.println(groveVref);
        for (int i = 0; i < 100; i++) {
            // Serial.println(sampleVol[i]);
            float avrAll = sampleVol[i] - (groveVref / 100.0f);
            // Serial.println(avrAll);
            sampleVolAll += avrAll * avrAll;
        }
        // Serial.println(sampleVolAll);
        sampleVari = sampleVolAll / 99.0f;
        Serial.println(sampleVari);
        Serial.println(groveVref);
    }
    // return groveVref;
}

void setup() {
    Serial.begin(115200);
    pinMode(Din_Pin, OUTPUT);
    digitalWrite(Din_Pin, groveOn);

    display.begin();
    if (display.width() < display.height())
    {
        display.setRotation(display.getRotation() ^ 1);
    }

    // ADC initialization
    gpio_pad_select_gpio(Aout_Pin);

```

```

gpio_set_direction((gpio_num_t)Aout_Pin, GPIO_MODE_INPUT);
adc1_config_width(ADC_WIDTH_BIT_12);
adc1_config_channel_atten(ADC1_CHANNEL_0, ADC_ATTEN_DB_11);
adc_chars = (esp_adc_cal_characteristics_t *)calloc(
    1, sizeof(esp_adc_cal_characteristics_t));
esp_adc_cal_characterize(ADC_UNIT_1, ADC_ATTEN_DB_11, ADC_WIDTH_BIT_12,
    3300, adc_chars);

getVerf();
groveVref = groveVref / 100.0f / 1000.0f;

canvas.setColorDepth(1); // mono color
canvas.createSprite(display.width(), display.height());
canvas.setTextSize((float)canvas.width() / 160);
canvas.setTextScroll(true);
}

void loop() {
    float groveVol = get_battery_voltage() / 1000.0f;
    Serial.printf("Voltage is: %fVrn", groveVol);
    canvas.printf("Voltage is: %fVrn", groveVol);
    float groveCurrent = ((groveVol - groveVref) / 50.0f / 0.02f);
    Serial.printf("Current is: %fArn", groveCurrent);
    canvas.printf("Current is: %fArn", groveCurrent);
    canvas.pushSprite(0, 0);
    delay(1000);
}

```

UiFlow1

- [Unit Grove to Grove UiFlow1 Docs](#)

UiFlow2

- [Unit Grove to Grove UiFlow2 Docs](#)

| Video