UNIT-GROVE2GROVE

SKU:U148

Description

UNIT-GROVE2GROVE is a Grove expansion Unit with `On/Off Control` + `Current Meter` functions. On/Off control adopts switch value, Current meter is 0 - 3.3V analog signal.

Product Feature

- On/Off control: 5V/1A
- Current meter: 0~1000mA

Include

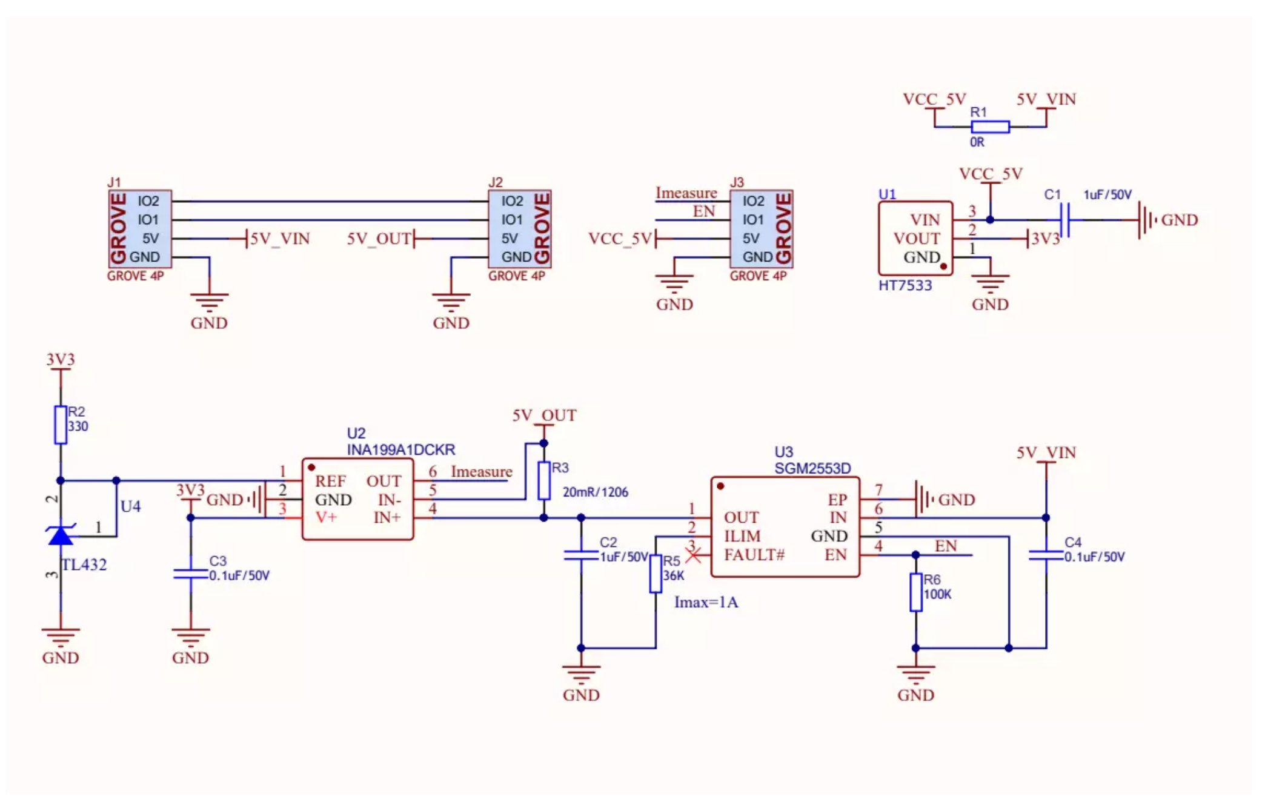
- 1x UNIT-GROVE2GROVE
- 1x HY2.0-4P cable (20cm)

Specification

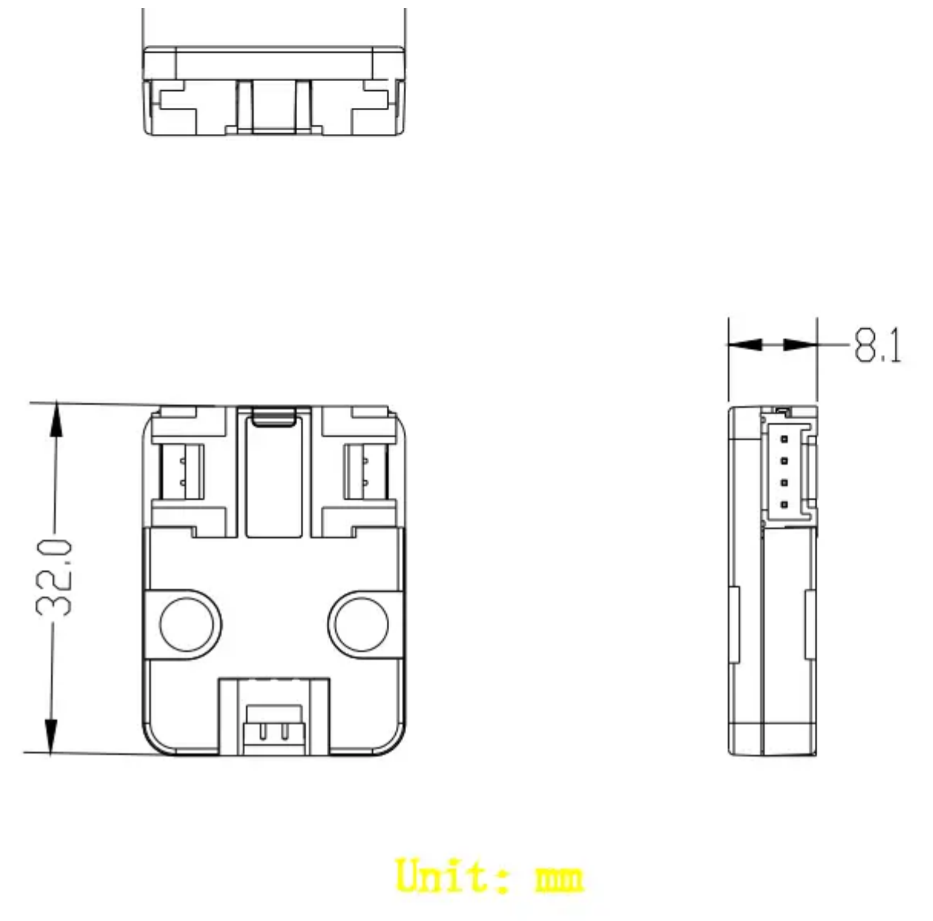
Spec	Parameter
Net Weight	5.0g
Gross Weight	10.5g

Spec	Parameter
Voltage	5V
Switch Current	1000mA
Circuit Voltage	5V
Current Meter Range	0~1000mA
Product Size	32.04*24.01*8.05 mm
Package Size	90*135mm

Schematic



Size

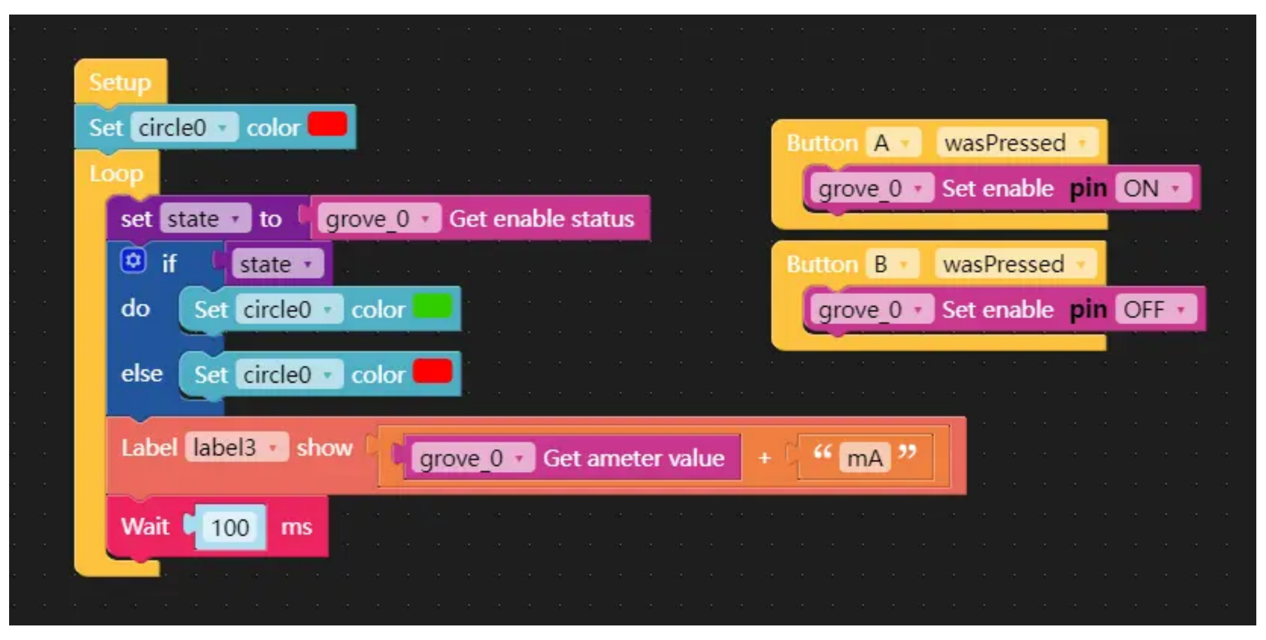


Example

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: %d\tVoltage: %dmV\r\n", 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初始化
    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);
    // groveVref = get_battery_voltage();
    // for (size_t i = 0; i < 5; i++) {
    //     groveVref = groveVref + get_battery_voltage();
    //     Serial.println(groveVref);
    // }
    // groveVref = groveVref / 5.0f / 1000.0f;
    getVerf();
    // Serial.println(groveVref);
    groveVref = groveVref / 100.0f / 1000.0f;
    // Serial.println(groveVref);

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


void loop() {
    // Serial.printf("Raw is %d\n", analogRead(Aout_Pin));
    float groveVol = get_battery_voltage() / 1000.0f;
    // Serial.println(groveVol);
    Serial.printf("Voltage is: %fV\r\n", groveVol);
    canvas.printf("Voltage is: %fV\r\n", groveVol);
    // float groveCurrent = ((groveVol - groveVref) / 50.0f / 0.01f);
    // float groveCurrent = ((groveVol - groveVref) / 83.0f / 0.01f);
    float groveCurrent = ((groveVol - groveVref) / 50.0f / 0.02f);
    Serial.printf("Current is: %fA\r\n", groveCurrent);
    canvas.printf("Current is: %fA\r\n", groveCurrent);
    //digitalWrite(Din_Pin, groveOff);
    canvas.pushSprite(0, 0);
    delay(1000);
}
```

UIFlow



```
Setup
Set circle0 color 🟥
Loop
    set state to grove_0 Get enable status
    if state
    do   Set circle0 color 🟩
    else Set circle0 color 🟥
    Label label3 show grove_0 Get ameter value + " mA "
    Wait 100 ms

Button A wasPressed
    grove_0 Set enable pin ON

Button B wasPressed
    grove_0 Set enable pin OFF
```