

ใบงานที่ 8

ADC/DAC

อุปกรณ์ที่ต้องใช้ในการทดลอง

1. สาย USB
2. บอร์ด ESP32
3. ตัวความต้านทานที่มีค่า 40k – 50k ohms
4. ตัวต้านทานปรับค่าได้ 10k Ohms

วงจร

ให้นิสิตต่อวงจรแบ่งแรงดัน โดยใช้ตัวต้านทานค่าคงที่ต่ออนุกรมกับตัวต้านทานปรับค่าเพื่อทำให้แรงดันตกคร่อมที่ตัวต้านทานปรับค่าได้มีค่าไม่เกิน 1.1V

ขาของ ADC1 CH#4 เป็นขา GPIO32

ขาของ ADC1 CH#5 เป็นขา GPIO33

**** ข้อควรระวัง เนื่องจาก ADC ของ ESP32 ไม่สามารถทนแรงดันได้เกินกว่า 1.1V**

การใช้งาน ADC แบบ One short

1. โค้ดส่วนของการ include header

```
1  #include <stdio.h>
1  #include "freertos/FreeRTOS.h"
2  #include "freertos/task.h"
3  #include "esp_log.h"
4  #include "esp_adc/adc_oneshot.h"
5
6  const static char *TAG = "ADC";
7
8  /*-----
9  |      ADC General Macros
10 -----*/
11 //ADC1 Channels
12 #define EXAMPLE_ADC1_CHAN0      ADC_CHANNEL_4
13 #define EXAMPLE_ADC1_CHAN1      ADC_CHANNEL_5
14
15 #define EXAMPLE_ADC_ATTEN        ADC_ATTEN_DB_11
16
17 static int adc_raw[2][10];
18
```

2. โค้ดในส่วนของ app_main โดยกำหนดใช้งาน ADC#1 จากนั้นกำหนด CHANNEL ที่ต้องการเพื่ออ่านค่าจากขาอื่นๆ

```
30 void app_main(void)
29 {
28     //-----ADC1 Init-----//
27     adc_oneshot_unit_handle_t adc1_handle;
26     adc_oneshot_unit_init_cfg_t init_config = {
25         .unit_id = ADC_UNIT_1,
24     };
23     ESP_ERROR_CHECK(adc_oneshot_new_unit(&init_config, &adc1_handle));
22
21     //-----ADC1 Config-----//
20     adc_oneshot_chan_cfg_t config = {
19         .bitwidth = ADC_BITWIDTH_DEFAULT,
18         .atten = EXAMPLE_ADC_ATTEN,
17     };
16     ESP_ERROR_CHECK(adc_oneshot_config_channel(adc1_handle, EXAMPLE_ADC1_CHAN0, &config));
15     ESP_ERROR_CHECK(adc_oneshot_config_channel(adc1_handle, EXAMPLE_ADC1_CHAN1, &config));
14
13     while (1) {
12         ESP_ERROR_CHECK(adc_oneshot_read(adc1_handle, EXAMPLE_ADC1_CHAN0, &adc_raw[0][0]));
11         ESP_LOGI(TAG, "ADC%d Channel[%d] Raw Data: %d", ADC_UNIT_1 + 1, EXAMPLE_ADC1_CHAN0, adc_raw[0][0]);
10         vTaskDelay(pdMS_TO_TICKS(1000));
9
8         ESP_ERROR_CHECK(adc_oneshot_read(adc1_handle, EXAMPLE_ADC1_CHAN1, &adc_raw[0][1]));
7         ESP_LOGI(TAG, "ADC%d Channel[%d] Raw Data: %d", ADC_UNIT_1 + 1, EXAMPLE_ADC1_CHAN1, adc_raw[0][1]);
6         vTaskDelay(pdMS_TO_TICKS(1000));
5     }
4
3     //Tear Down
2     ESP_ERROR_CHECK(adc_oneshot_del_unit(adc1_handle));
1 }
50
```

บรรทัดที่ 12 เป็นการสั่งให้ ADC อ่านค่าและเก็บผลลัพธ์ที่ได้ลงในตัวแปร adc_raw

3. ทำการปรับค่าตัวต้านทานปรับค่าได้และสังเกตค่าที่อ่านได้จาก monitor

ใบงานท้ายการทดลอง

ให้นักเรียนโค้ดอ่านค่าจากตัวต้านทานปรับค่าได้ โดย

- เมื่อค่าตัวต้านทานมีค่าน้อยที่สุด (หมุนมาทางซ้ายสุด) ให้ MCU สร้างสัญญาณ PWM เพื่อหมุน servo motor ไปที่มุม -90 องศา
- หากค่าตัวต้านทานมีค่ามากที่สุด (หมุนไปทางขวาสุด) ให้ MCU สร้างสัญญาณ PWM เพื่อหมุน servo motor ไปที่มุม 90 องศา
- เก็บภาพหน้าจอจากออสซิลโลสโคปใส่ในผลการทดลองด้วย