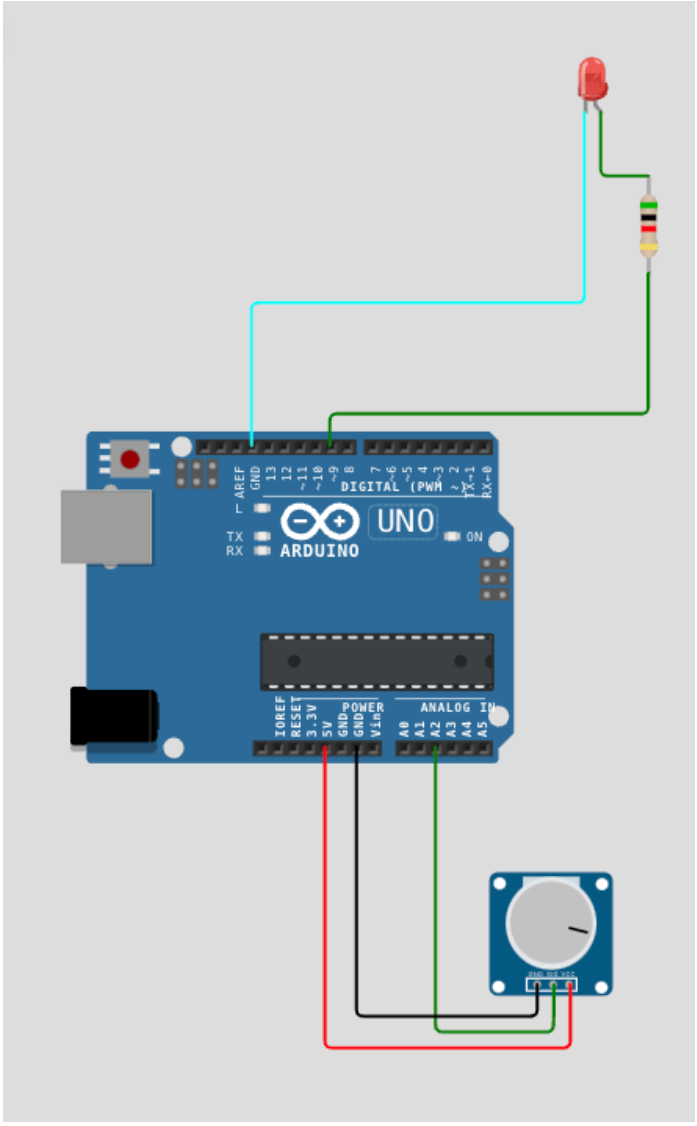
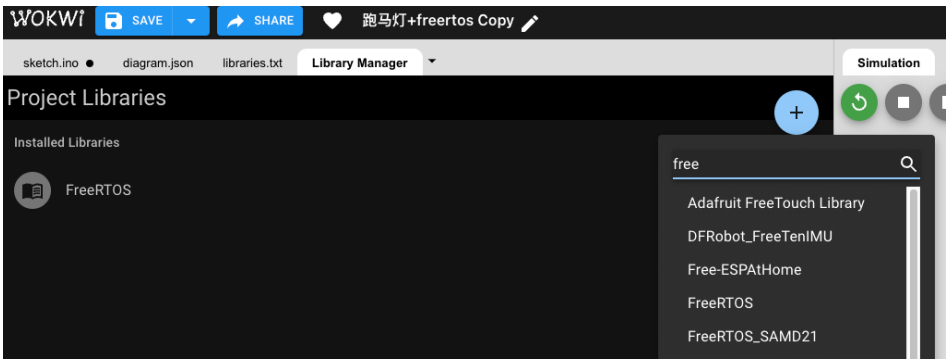
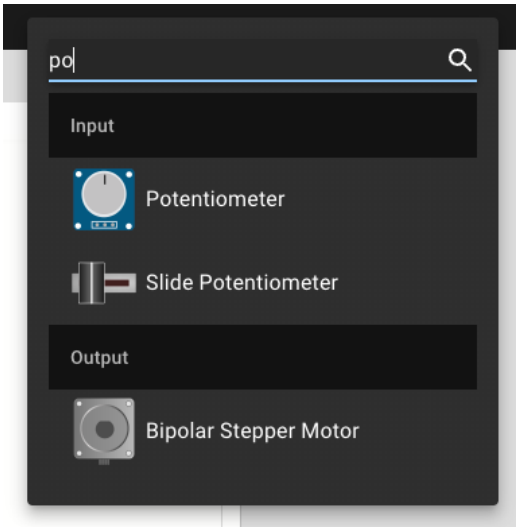
参考

1.连接框图


DIGITAL (PWM
UNO
ARDUINO
TX
RX
ON
POWER
ANALOG IN

2. 库


WOKWi
SAVE
SHARE
跑马灯+freertos Copy
sketch.ino
diagram.json
libraries.txt
Library Manager
Simulation
Project Libraries
Installed Libraries
FreeRTOS
free
Adafruit FreeTouch Library
DFRobot_FreeTenIMU
Free-ESPAtHome
FreeRTOS
FreeRTOS_SAMD21



3. Sketch.ino

```
#include <Arduino_FreeRTOS.h>
#include <queue.h>

//
#define LED1_PIN 9
//#define LED_PIN 10
//#define LED_PIN 11

#define ANALOG_PIN A2      // Pin Analog A2

QueueHandle_t integerQueue;

// Task Function Prototypes
void TaskBlink(void *pvParameters);
void TaskAnalogRead(void *pvParameters);
void TaskIdentitas(void *pvParameters);

// 旋钮控制可变电阻（线性电位器）
int analogValue = 0;

void setup() {

   // Serial Monitor
  Serial.begin(9600);
  //  LED_PIN MODE = OUTPUT
  pinMode(LED1_PIN, OUTPUT);
```

```
  //pinMode(LED2_PIN, OUTPUT);
  //pinMode(LED3_PIN, OUTPUT);

   integerQueue = xQueueCreate(10, // Queue length
                         sizeof(int) // Queue item size
                         );


  // Task 1: LED Blink
  xTaskCreate(
    TaskBlink,
    "Task Blink",
    64,
    NULL,
    2,
    NULL);

  // Task 2:
  xTaskCreate(
    TaskAnalogRead,
    "Task Analog Read",
    64,
    NULL,
    1,
    NULL);

  // Create Task 3:
  xTaskCreate(
    TaskSendInfo,
    "Task Send Info",
    64,
    NULL,
    3,
    NULL);

}

void loop() {
  //

}

// Task 1 - LED Blink
void TaskBlink(void *pvParameters) {
  //
```

```
  TickType_t xLastWakeTime = xTaskGetTickCount(); //获取系统当前运行的时钟节拍数
  // Periode delay task (2000 ms)定时
  const TickType_t xFrequency = 2000 / portTICK_PERIOD_MS;

  for(;;) {
    digitalWrite(LED1_PIN, HIGH);
    Serial.println("LED ON");
    vTaskDelayUntil(&xLastWakeTime, xFrequency);  // 绝对延时

    digitalWrite(LED1_PIN, LOW);
    Serial.println("LED OFF");
    vTaskDelayUntil(&xLastWakeTime, xFrequency);
  }
}

// Task 2 -
void TaskAnalogRead(void *pvParameters) {
  //  potentiometer可变电阻
  TickType_t xLastWakeTime = xTaskGetTickCount();
  // Periode delay  task (500 ms)
  const TickType_t xFrequency = 500 / portTICK_PERIOD_MS;

  for(;;) {
    analogValue = analogRead(ANALOG_PIN);  // read pin A2

    xQueueSend(integerQueue, &analogValue, portMAX_DELAY);

    Serial.print("可变电阻值为 : ");
    Serial.println(analogValue);
    vTaskDelayUntil(&xLastWakeTime, xFrequency);  // Delay
  }
}

// Task 3 - Serial Monitor
void TaskSendInfo(void *pvParameters) {
  //
  int receivedADCValue;
  TickType_t xLastWakeTime = xTaskGetTickCount();
  // Periode delay task (5000 ms)
  const TickType_t xFrequency = 5000 / portTICK_PERIOD_MS;

  for(;;) {
    Serial.print("SendInfo-传输数据：");
```

```
    if (f) {
      Serial.println(receivedADCValue);}

    vTaskDelayUntil(&xLastWakeTime, xFrequency);
  }
}
```