

BLINKING LED

Contents

1 Tujuan Percobaan	2
2 Hasil yang diharapkan	2
3 Komponen yang digunakan	2
4 Circuit	2
5 Warning	2
6 Code Description	3
7 Software Code	3
8 Analisa	5
9 Pengembangan dan latihan	5

1 Tujuan Percobaan

Mengatur nyala mati nya LED yang terhubung ke ESP8266, melalui software.

2 Hasil yang diharapkan

Led menyala selama 1 detik, mati selama 1 detik dan berulang.

3 Komponen yang digunakan

- 1 pcs Power supply 5v atau bisa dari USB yang terhubung dengan PC.
- 1 pcs LED merah diffused (yang bright bikin sakit mata).
- 1 pcs Resistor dengan nilai 1K Ohm.

4 Circuit

Power Supply 5v hanya dibutuhkan jika rangkaian tidak terbubung dengan USB komputer, jika USB terhubung dengan komputer maka sumber tegangan 5V didapatkan langsung dari port USB komputer. Pada percobaan ini LED terhubung dengan ESP8266 melalui R 1K yang bertujuan untuk membatasi arus yang mengalir pada LED. Lebih lengkap nya bisa dilihat pada Figure 1.

5 Warning

Jangan sampai terbalik menghubungkan polaritas dari power supply ke ESP8266, kesalahan penyambungan berakibat rusak nya rangkaian ESP8266.

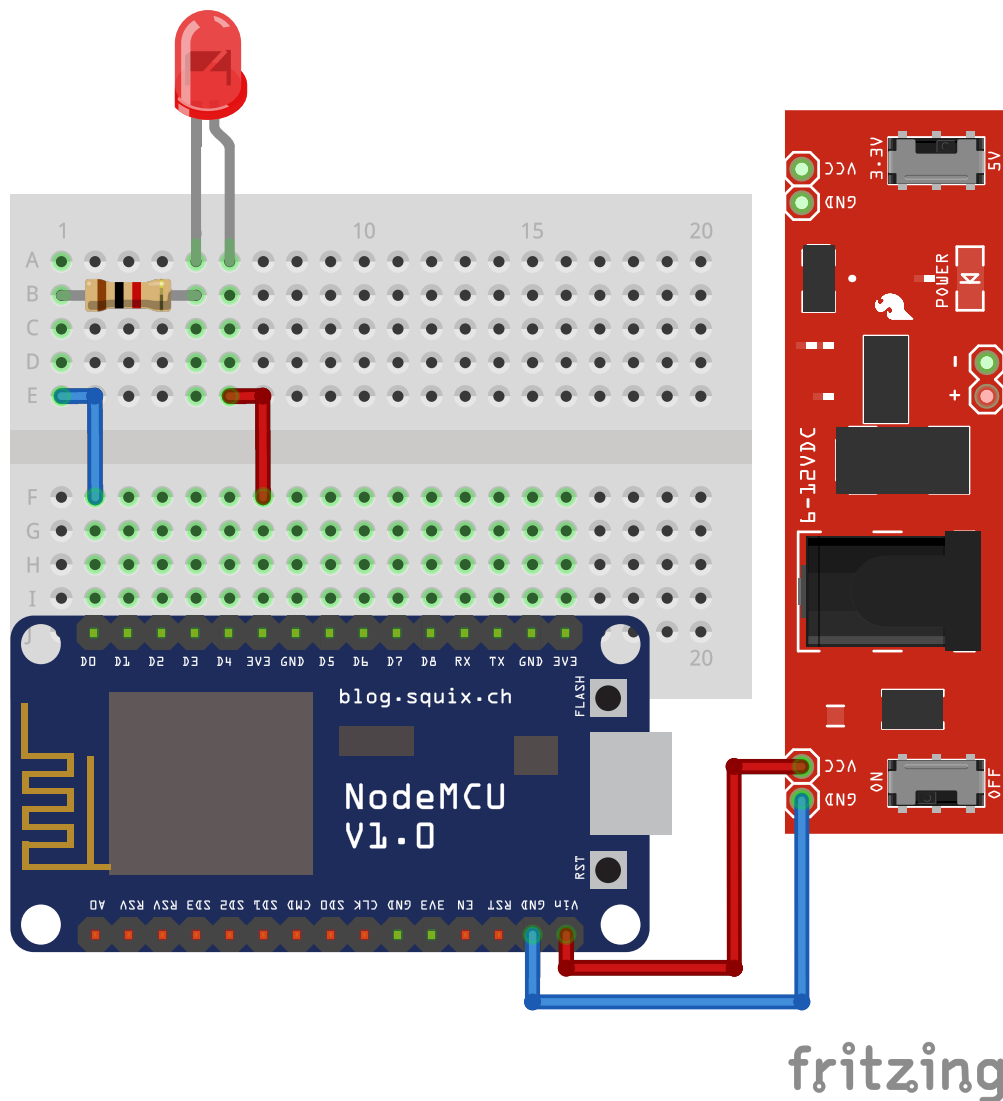


Figure 1: Rangkaian LED Blink.

6 Code Description

Pada Esp8266 terdapat 16 GPIO yang bisa digunakan baik sebagai input atau sebagai output. Percobaan ini menggunakan GPIO D0 yang disetting sebagai output. Sesuai dengan Figure 1... maka untuk menyalakan LED, D0 harus berada pada kondisi LOW atau 0, sedang untuk menyalakan LED D0 harus pada kondisi HIGH atau 1. Sedangkan untuk durasi nyala dan mati nya LED menggunakan fungsi wait yang telah disediakan oleh OS (Operating System).

7 Software Code

Pembuatan software diawali dengan setup project dengan cara menjalankan perintah seperti pada Listing 1... pada terminal.

```

1 $ mkdir led_blink
2 $ cd led_blink
3 $ pio init --ide=codeblocks --board=d1 --project-option "framework=esp8266-rtos-sdk"

```

Listing 1: Setup project blink

Jika tidak ada error maka akan terdapat file platformio.cbp yang merupakan project file untuk codeblocks. Buka platformio.cbp dengan codeblocks dan buat file baru dengan nama blink.c yang berisi dengan code seperti pada Listing 2.

```

1 #include "esp_common.h"
2 #include "freertos/task.h"
3 #include "gpio.h"
4
5 /*****
6  * FunctionName : user_rf_cal_sector_set
7  * Description  : SDK just reversed 4 sectors, used for rf init data and paramters.
8  *               We add this function to force users to set rf cal sector, since
9  *               we don't know which sector is free in user's application.
10 *               sector map for last several sectors : ABCCC
11 *               A : rf cal
12 *               B : rf init data
13 *               C : sdk parameters
14 * Parameters   : none
15 * Returns      : rf cal sector
16 *****/
17 uint32 user_rf_cal_sector_set(void)
18 {
19     flash_size_map size_map = system_get_flash_size_map();
20     uint32 rf_cal_sec = 0;
21     switch (size_map)
22     {
23     case FLASH_SIZE_4M_MAP_256_256:
24         rf_cal_sec = 128 - 5;
25         break;
26
27     case FLASH_SIZE_8M_MAP_512_512:
28         rf_cal_sec = 256 - 5;
29         break;
30
31     case FLASH_SIZE_16M_MAP_512_512:
32     case FLASH_SIZE_16M_MAP_1024_1024:
33         rf_cal_sec = 512 - 5;
34         break;
35
36     case FLASH_SIZE_32M_MAP_512_512:
37     case FLASH_SIZE_32M_MAP_1024_1024:
38         rf_cal_sec = 1024 - 5;
39         break;
40
41     default:
42         rf_cal_sec = 0;
43         break;
44     }
45
46     return rf_cal_sec;
47 }
48
49 void task_blink(void* ignore)
50 {
51
52     ///! deklarasi pin 16 atau D0 untuk
53     GPIO_ConfigTypeDef io_out_conf;
54     io_out_conf.GPIO_IntrType = GPIO_PIN_INTR_DISABLE; ///! tidak menggunakan
55     interrupt
56     io_out_conf.GPIO_Mode = GPIO_Mode_Output; ///! sebagai output

```

```

56     io_out_conf.GPIO_Pin      = 16; /// pin 16 atau D0
57     io_out_conf.GPIO_Pullup  = GPIO_PullUp_EN; /// aktifkan pullup internal
58     gpio_config(&io_out_conf);
59
60
61     while(true) /// ulang selama lama lama nya
62     {
63         GPIO_OUTPUT_SET(16,0); /// D0 = 0, LED nyala
64         vTaskDelay(1000/portTICK_RATE_MS); /// tunggu 1000ms = 1s detik
65         GPIO_OUTPUT_SET(16,1); /// D0 = 1, LED mati
66         vTaskDelay(1000/portTICK_RATE_MS); /// tunggu 1000ms = 1s detik
67     }
68
69     vTaskDelete(NULL);
70 }
71
72 /* *****
73  * FunctionName : user_init
74  * Description  : entry of user application, init user function here
75  * Parameters   : none
76  * Returns      : none
77 ***** */
78 void user_init(void)
79 {
80     xTaskCreate(&task_blink, "startup", 2048, NULL, 1, NULL);
81 }

```

Listing 2: Pogram utama LED blink

Pastikan semua perintah ditulis dengan benar dan build project, jika tidak ada error yang dilaporkan oleh codeblocks maka project bisa diupload ke board dengan menggunakan perintah

```

1 $ pio --target upload

```

Listing 3: Upload Project ke board

8 Analisa

Pogram Listing 2... pada dasarnya hanya membuat D0 menjadi output, dan memberikan fungsi tunda nyala dan mati pada LED. Perintah untuk membuat D0 menjadi output terdapat pada baris 53 sampai 58, dan untuk fungsi tunda 1 detik terdapat pada baris 64 dan 66.

9 Pengembangan dan latihan

- Ganti rangkaian dan pogram untuk D1
- Ganti delay dari 1s ke 500ms