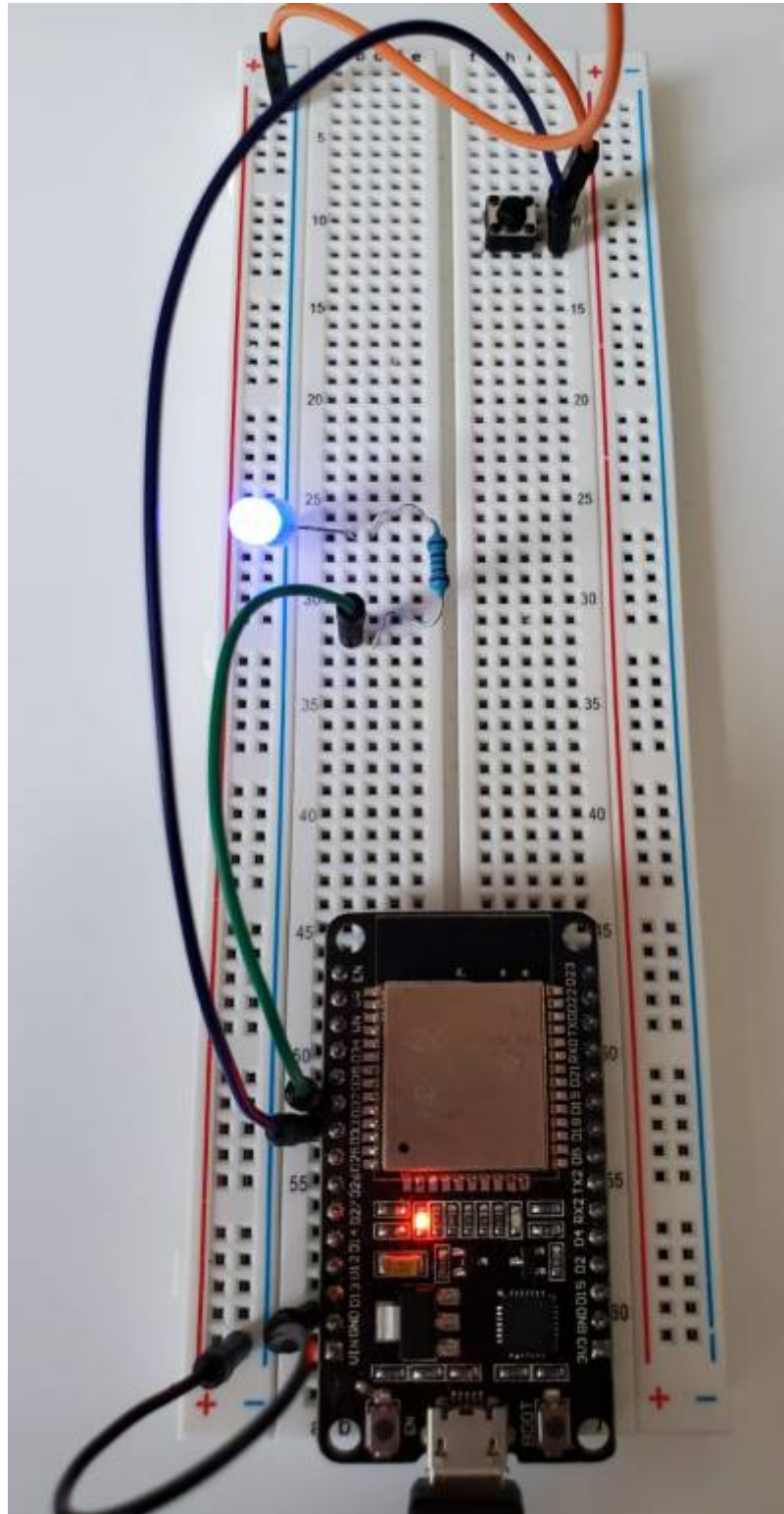
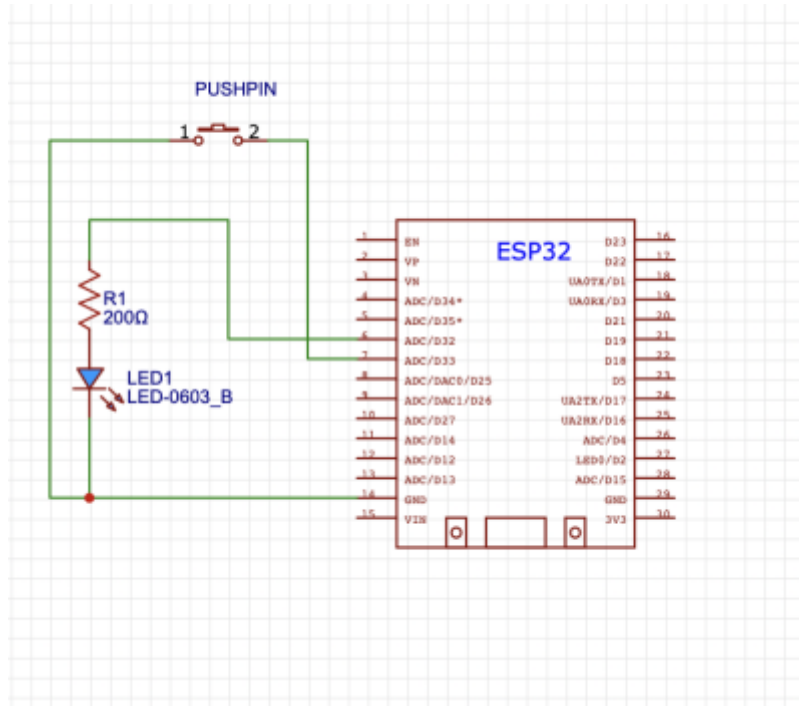


# Build



**figure 4.1** Deep-Sleep set up for ESP32/

## Schematics



**figure 4.2** Graph depicting Deep-Sleep schematics set up for ESP32/

## Code & Description

```

*/There are 2 ways to wake up a system, those are called Interrupts.
*/hardware Interrupts are based on external events where signals are sent
to the GPIO.
*/Software Interrupts occur when we program the device, like through a wake
up alarm or timer.

#define uS_TO_S_FACTOR 1000000           */Equation to convert milliseconds
to Minutes
#define TIME_TO_SLEEP  10                 */Determined amount of minutes at
sleep

RTC_DATA_ATTR int bootCount = 0;

*/Data has to be stored in the Real time Clock (RTC) fast memory because
the CPU memory is wiped on every boot.
*/ The amount of data has to be minimal due to size limitations : 8bit fast
memory and 8bit Slow memory.

*/Power outages and resetting the board will erase the RTC memory,
therefore its use should be limited to non essential information.
*/Therefore we decided to just Keeping data of times awaken in internal
RTC, this will help us see if any issues occur.

*****Wake Up Print
*****

```

```
void print_wakeup_reason(){                                     */Set up to list reasons
for system wake-up
    esp_sleep_wakeup_cause_t wakeup_reason;

    wakeup_reason = esp_sleep_get_wakeup_cause(); */System wakes up due to
3 reasons

    switch(wakeup_reason)
    {
        case ESP_SLEEP_WAKEUP_EXT0 : Serial.println("PUSHED BUTTON caused the
system to WAKEUP"); break;
        case ESP_SLEEP_WAKEUP_TIMER : Serial.println("TIMER caused the system to
WAKEUP"); break;
        default : Serial.printf("DeepSleep didn't wake up ESP32:
%d\n",wakeup_reason); break; */Initial Boot will yield this reason
    }
}

*/We made the system wake up due to the internal timer and a pushed button,
*/ the reason for the push button is that in the case we can take a
measurement at our will without having to wait for the timers.
*/An external clock could be added to the device, however Grafana already
designates time and date of transferred data.

void setup(){
    Serial.begin(115200);
    delay(1000);

    pinMode(32,OUTPUT); */Illuminates a LED when the system is awake.
For visual confirmation.

    pinMode(33,INPUT_PULLUP);

    */ESP32 has pull-up resistors built on the pins, when we activate it
    */it avoids the use of external resistors
    */INPUT_PULLUP keeps the signal HIGH by default
    */floating currents which can produce erroneous readings are avoided by
    pull up/down resistors

    for(int i=0;i<5;i++)
    {
        digitalWrite(32,HIGH); */ These are the parameters for the LED
flashing
        delay(1000);
        digitalWrite(32,LOW);
        delay(1000);
    }
}
```

```
++bootCount;
Serial.println("Reboot count number: " + String(bootCount));    */We
want to know how many times the system has booted
                                                                    */easy
way to know if there are issues internal or battery
                                                                    */issues

print_wakeup_reason();

esp_sleep_enable_ext0_wakeup(GPIO_NUM_33,0);

*/During Sleep only Pins connected to the RTC are operational
*/A General Purpose Input/output pin are used to perform digital readings
and output functions.
*/By default those pins have no predefined purpose.
*/The pin used has to be named after their GPIO

esp_sleep_enable_timer_wakeup(TIME_TO_SLEEP * uS_TO_S_FACTOR);
*/conversion factor to minutes

Serial.println("ESP32 is going into DeepSleep for " +
String(TIME_TO_SLEEP) +    */Script detailing the process
" Seconds");

Serial.println("Going to sleep now.....");
delay(1000);
Serial.flush();
*/To avoid mistakes in data transmission by
*/clearing buffer
esp_deep_sleep_start();

}
```

\*\*\* There is no loop section in our coding because the micro controller will go into deep sleep before reaching that part of the code, which means everything has to be written in the void set up section of the sketch.

## Results

```
rst:0x1 (POWERON_RESET),boot:0x13 (SPI_FAST_FLASH_BOOT)
config: 0, SPIWP:0xee
clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00
mode:DIO, clock div:1
load:0x3fff0030,len:1344
load:0x40078000,len:13516
load:0x40080400,len:3604
entry 0x400805f0
Reboot count number: 1
DeepSleep didn't wake up ESP32: 0
ESP32 is going into DeepSleep for 10 Seconds
Going to sleep now.....
ets Jun  8 2016 00:22:57
```

```
rst:0x5 (DEEPSLEEP_RESET),boot:0x13 (SPI_FAST_FLASH_BOOT)
config: 0, SPIWP:0xee
clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00
mode:DIO, clock div:1
load:0x3fff0030,len:1344
load:0x40078000,len:13516
load:0x40080400,len:3604
entry 0x400805f0
Reboot count number: 2
TIMER caused the system to WAKEUP
ESP32 is going into DeepSleep for 10 Seconds
Going to sleep now.....
ets Jun  8 2016 00:22:57
```

```
rst:0x5 (DEEPSLEEP_RESET),boot:0x13 (SPI_FAST_FLASH_BOOT)
config: 0, SPIWP:0xee
clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00
mode:DIO, clock div:1
load:0x3fff0030,len:1344
load:0x40078000,len:13516
load:0x40080400,len:3604
entry 0x400805f0
Reboot count number: 3
PUSHED BUTTON caused the system to WAKEUP
ESP32 is going into DeepSleep for 10 Seconds
Going to sleep now.....
ets Jun  8 2016 00:22:57
```

From:

<https://student-wiki.eolab.de/> - **HSRW EOLab Students Wiki**

Permanent link:

<https://student-wiki.eolab.de/doku.php?id=amc2022:grouph:here&rev=1662764076>

Last update: **2023/01/05 14:38**

