Build



Schematics

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



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_T0_S_FACTOR 1000000
to Minutes
#define TIME_T0_SLEEP 10
sleep

#define uS_T0_S_FACTOR 1000000 *//Equation to convert milliseconds

*//Determined amount of minutes at

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.

```
2025/08/19 22:03
                          3/5
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)
  Ł
    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(32,0UTPUT); *//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++)</pre>
```

```
{
```

```
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(GPI0 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_T0_SLEEP * uS_T0_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) configsip: 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) configsip: 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) configsip: 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=1662756726



Last update: 2023/01/05 14:38