SNTP

```
#include <string.h>
#include <time.h>
#include <sys/time.h>
#include <freertos/FreeRTOS.h>
#include <freertos/task.h>
#include <esp_system.h>
#include <esp_event.h>
#include <esp_log.h>
#include <esp_attr.h>
#include <esp_sleep.h>
#include <nvs_flash.h>
#include <esp_sntp.h>
#include "protocol_examples_common.h"


static const char *TAG = "example";
RTC_DATA_ATTR static int boot_count = 0;


static void obtain_time (void);
static void initialize_sntp (void);


#ifdef CONFIG_SNTP_SYNC_METHOD_CUSTOM
void sntp_sync_time (struct timeval *tv) {
    settimeofday (tv, NULL);
    ESP_LOGI (TAG, "Time is synchronized from custom code");
    sntp_set_sync_status (SNTP_SYNC_STATUS_COMPLETED);
}
#endif
```

```
void time_sync_notification_cb (struct timeval *tv) {
  ESP_LOGI (TAG, "notification of a time synchronization event");
}


void app_main () {
  ++boot_count;
   ESP_LOGI (TAG, "Boot count: %d", boot_count);
   time_t now;
   struct tm timeinfo;
     time (&now);
     localtime_r (&now, &timeinfo);
      if (timeinfo.tm_year < (2016 - 1900)) {
         ESP_LOGI (TAG, "time is not set yet. connecting to
           WiFi and getting time over NTP");
         obtain_time ();
          time (&now);
      }
#ifdef CONFIG_SNTP_TIME_SYNC_METHOD_SMOOTH
      else {
        {
           ESP_LOGI (TAG, "add a error for test adjtime");
           struct timeval tv_now;
           gettimeofday (&tv_now, NULL);
           int64_t cpu_time = (int64_t) tv_now.tv_sec
              * 1000000L + (int64_t) tv_now.tv_usec;
           int64_t error_time = cpu_time + 500 * 1000L;
           struct timeval tv_error = { .tv_sec =
              error_time / 1000000L, .tv_usec =
              error_time % 1000000L };
        settimeofday (&tv_error, NULL);
      }
      ESP_LOGI (TAG, "time was set, now just adjusting it. Use SMOOTH SYNC method");
      obtain_time();
      time (&now);
    }
#endif

    char strftime_buf[64];
    setenv ("TZ", "EST5EDT, M3.2.0/2, M11.1.0", 1);
    tzset ();
     localtime_r (&now, &timeinfo);
     strftime (strftime_buf, sizeof(strftime_buf), "%c", &timeinfo);
     ESP_LOGI (TAG, "the current date/time in new york is : %s",
                                                      strftime_buf);

     setenv("TZ", "CST-8", 1);
     tzset ();
     localtime_r (&now, &timeinfo);
     strftime(strftime_buf, sizeof (strftime_buf), "%c", &timeinfo);
     ESP_LOGI (TAG, "the current date/time in shanghai is : %s",
                                                      strftime_buf);

     if(sntp_get_sync_mode() == SNTP_SYNC_MODE_SMOOTH) {
        struct timeval outdelta;
        while (sntp_get_sync_status() == SNTP_SYNC_STATUS_IN_PROGRESS) {
          adjtime (NULL, &outdelta);
          ESP_LOGI (TAG, "waiting for adjusting time ...
              outdelta = %li sec: %li ms : %li us",
              outdelta. tv_sec, outdelta. tv_usec / 1000,
              outdelta. tv_usec % 1000 );
```

```
            vTaskDelay(2000 / portTICK_PERIOD_MS);
        }
    }

    Const int deep_sleep_sec = 10;
    ESP_LOGI(TAG, "entering deep sleep for %d seconds",
                                    deep_sleep_sec);
     esp_deep_sleep(1000000LL * deep_sleep_sec);
}



static void obtain_time(void) {
    ESP_ERROR_CHECK( nvs_flash_init() );
    tcpip_adapter_init();
    ESP_ERROR_CHECK( esp_event_loop_create_default() );
    ESP_ERROR_CHECK( example_connect() );

     initialize_sntp();
     time_t now = 0;
     struct tm timeinfo = {0};
      int retry = 0;
      while (sync_get_sync_status() == SNTP_SYNC_STATUS_RE
        && ++retry < retry_count) {
         ESP_LOGI(TAG, "waiting for system time to be set...
              (%d/%d)", retry, retry_count);
         vTaskDelay(2000 / portTICK_PERIOD_MS);
      } time(&now);
         localtime_r(&now, &timeinfo);
         ESP_ERROR_CHECK(example_disconnect());
}
```

```
static void initialize-sntp (void) {
   ESP_LOGI (TAG, "initializing SNTP");
   sntp_setoperatingmode ( SNTP_OPMODE_POLL );
   sntp_setservername (0, "pool.ntp.org")
   sntp_set_time_sync_notification_cb (time_sync_notification_cb);
#ifdef CONFIG_SNTP_TIME_SYNC_METHOD_SMOOTH
   sntp_set_sync_mode (SNTP_SYNC_MODE_SMOOTH);
# endif
   Sntp_init ();
}
```