Comentario técnico: CTC-097
Componente: **SNTP con ESP32 y Mongoose-OS**
Autor: Sergio R. Caprile, Senior R&D Engineer

Revisiones	Fecha	Comentarios
0	09/06/20	

Mongoose-OS incorpora un cliente SNTP, una versión simplificada de NTP (Network Time Protocol). Esto nos permite conectarnos a servidores NTP (en Internet o en nuestra red) y obtener información precisa de fecha y hora sin necesidad de contar con un RTC y evitando el inconveniente de tener que ponerlo en hora.

Configuración

Configuramos el ESP32 con Mongoose-OS para conectarse como cliente a una red WiFi. Dado que debemos conectarnos a un servidor, ésta nos resulta la forma tal vez más rápida y simple de realizar las pruebas y explicar la operación. La configuración puede realizarse manualmente mediante RPC, en un archivo de configuración JSON; o definirla en el archivo YAML que describe el proyecto. Para las pruebas elegimos esta última opción.

```
libs:
  - origin: https://github.com/mongoose-os-libs/sntp       # incorpora el cliente SNTP
config_schema:
  - ["sntp.server", "time.google.com"]           # Dirección del server (por defecto Google)
  - ["sntp.update_interval", 7200]               # Tiempo entre sincronizaciones 0 => una vez
  - ["sys.tz_spec", "<-03>3"]                    # Timezone en formato tzset
```

Ambos parámetros SNTP podemos omitirlos, la conexión se hará a los servidores de Google cada dos horas. La especificación de zona horaria es difícil de manejar a menos que dominemos el uso de la utilidad *tzset*. De todos modos existe una ayuda[1]. En este ejemplo hemos configurado la zona horaria utilizada en Argentina al momento de escribir este Comentario Técnico.

Operación

Luego de compilado el código (`mos build`) y grabado el microcontrolador (`mos flash`) mediante *mos tool*, observaremos en el log cuando se establece la sincronización.
Para obtener la información, disponemos de las siguientes funciones:

- En mJS: `Timer.now();`
- En C: `double mg_time(void);`

Ambas devuelven el tiempo UTC en segundos desde la *epoch* Unix[2], en coma flotante con doble precisión.
Ambas funciones obtienen la información de *gettimeofday()*[3], como sugiere la documentación del IDF de Espressif[4].

1 https://github.com/mamuesp/timezones. Señalamos que es un artículo genérico y debido a cambios posteriores de Mongoose-OS requiere algo de manipulación pero es fácil de incluir en nuestro código si lo necesitamos y lo deseamos.
2 La *epoch* Unix corresponde al 1 de enero de 1970. El valor devuelto es la cantidad de segundos que ha transcurrido desde el comienzo de ese día.
3 Dado que el compilador utilizado es GNU, es esperable disponer de algunas funciones POSIX de *libc*. Las funciones que detallamos aquí están documentadas y hemos comprobado que funcionan de la manera en que se muestra. Para más detalles, tanto Mongoose-OS como el IDF de Espressif son open source y puede observarse el código fuente y su respectiva documentación.
4 https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-reference/system/system_time.html

Para convertir esa información a fecha y hora local disponemos de funciones de formato, que a su vez utilizan *strftime()*, por lo que el formato es el utilizado por esta función standard:

- En mJS: `Timer.fmt(formato, time);`
- En C: `int mgos_strftime(buffer, tamaño, formato, time);`

Eventos

Dado que el entorno es event-driven, existe además la posibilidad de ejecutar una función cuando se produce la sincronización. Para ello debemos declarar un handler para el evento correspondiente. En el archivo adjunto hemos hecho uso de esto para el ejemplo. La forma de realizarlo es la siguiente:

En mJS

```
load('api_events.js');

let MGOS_EVENT_TIME_CHANGED = Event.SYS + 3;

Event.addHandler(MGOS_EVENT_TIME_CHANGED, function (ev, evdata, ud) {
        // here
}, null);
```

En C

```
#include "mgos_event.h"


static void my_time_change_cb(int ev, void *ev_data, void *userdata)
{
        // here
}


enum mgos_app_init_result mgos_app_init(void)
{
        mgos_event_add_handler(MGOS_EVENT_TIME_CHANGED, my_time_change_cb, NULL);
        return MGOS_APP_INIT_SUCCESS;
}
```