

EXTENDING BATTERY LIFE WITH SCRIPT-CONTROLLED SLEEP

1. Introduction

Senquip devices can automatically enter hibernate when external power is removed, waking periodically to measure and transmit data. This provides a simple way to reduce power consumption when equipment is switched off.

When a device is connected to permanent power, its supply voltage does not fall when the equipment stops, so Hibernate on Power Loss is not activated. Other applications may require more complex power control, with measurement and sleep intervals determined by equipment state, sensor readings or several conditions.

This application note uses a Senquip QUAD permanently connected to a vehicle battery as an example. The ignition signal is connected to IO1. When ignition is on, the QUAD measures and transmits every five seconds. When ignition is off, a script extends the interval, so the device sleeps between periodic location updates.

The script uses script function `SQ.set_next_cycle()` to schedule the next measurement cycle without changing the configured Base Interval. Configured wake sources remain active, allowing the device to wake early when an event occurs.



Figure 1 – Example of a machine with a small battery that needs to be preserved

Document Number	Revision	Prepared By	Approved By
APN0052	1.0	NGB	NB
Title			Page
Extending Battery Life with Script-Controlled Sleep			2 of 8

2. Vehicle Connection

Connect the QUAD power input to the vehicle battery and connect the ignition signal to IO1. The QUAD and ignition signal must share a common ground.

Because the QUAD remains connected to the vehicle battery, its supply voltage does not fall when ignition is switched off. Hibernate on Power Loss therefore cannot be used to control its ignition-off reporting interval. Instead, the script reads the voltage on IO1 and schedules a longer interval when ignition is off.

The example uses a threshold of 6 V to distinguish between ignition on and ignition off. This is suitable for a nominal 12 V or 24 V ignition signal. Change the threshold if another signal voltage is used.

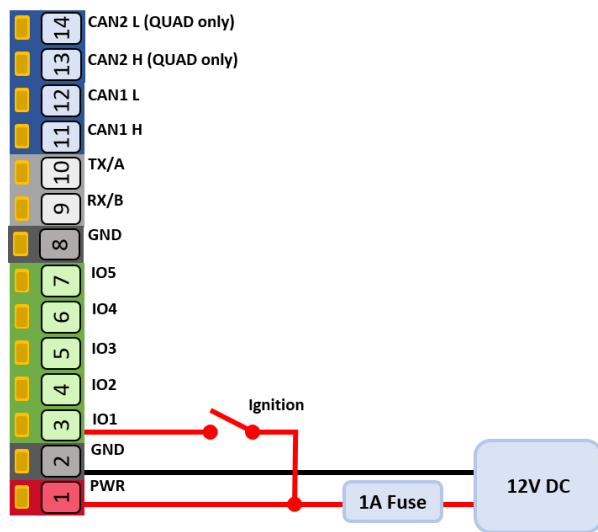


Figure 2 - QUAD connected to permanent battery power with ignition connected to IO1

Document Number
APN0052

Revision
1.0

Prepared By
NGB

Approved By
NB

Title
Extending Battery Life with Script-Controlled Sleep

Page
3 of 8

3. QUAD Configuration

The example uses a QUAD running SFW003 firmware with support for `SQ.set_next_cycle()`.

The following device settings are applied:

- Set the *Base Interval* to 5 seconds
- Disable *Device Always On* so the device can sleep when the script schedules a longer interval.
- Select *Disconnect Power During Sleep* to further reduce external battery use when the internal device battery is charged.

General

Device ID

HL9DTX35E

Device Model

QUAD-C2-G

Firmware Version

SFW003-7.1.5-beta

Hardware Revision

4

Device Name

QUAD-TX35E

Base Interval

5

Seconds

Measurement cycles will be every 5 seconds.

Transmit Interval

1

Transmissions will be every 5 seconds.

Exception Interval

1

When a warning or alarm is active, transmissions will be every 5 seconds.

Device Always On

☐ Enabled

Batch Transmit

☐ Enabled

Web Server

☐ Enabled

LEDs Always On

☐ Enabled

Internal Battery Profile

Extend Battery Lifespan

Power Input

Power Loss Alert

☒ Enabled

Disconnect Power During Sleep

☒ Enabled

Hibernate on Power Loss

☒ Enabled

Hibernate Delay Intervals

5

Hibernates 25 seconds after power loss.

Hibernate Period

21600

Seconds

Wakes and transmits every 6 hours during hibernate.

Count Hours

☐ Enabled

Figure 3 – QUAD General settings

Document Number	Revision	Prepared By	Approved By
APN0052	1.0	NGB	NB
Title			Page
Extending Battery Life with Script-Controlled Sleep			4 of 8

The following IO settings are applied:

- Enable IO1 *Wake on Change* to allow the device to wake from sleep when the ignition voltage changes.
- Select IO1 *Default State* as *OFF*, and the *Measurement State* as *No Change*. You could select *PULLUP* as the *Measurement State* if the ignition feed is a switch to ground signal.
- Configure IO1 to measure voltage on every cycle.

IO1 (Ignition) ✓ ⬆

Name

Ignition

Interval

1

Samples this input every 5 seconds.

Wake on Change

☒ Enabled

Wakes the device from sleep on a digital change. Set the default state to PULLUP for this to work reliably.

Output

Default State

OFF

Measurement State

No Change

Measurement Time

0

Seconds

Setting time in the Measurement State before a reading is taken.

Select a measurement type to configure.

Digital

Volts ✓

mA

Hz

Duty

Pulse

Voltage Measurement

Measure Voltage

☒ Enabled

Voltage Calibration

Voltage Alert

Figure 4 – QUAD IO1 settings

Set other required measurement intervals to 1 if they are to be collected on every cycle. GPS acquisition and transmission are handled by the device’s normal measurement cycle; the script only schedules the next cycle. A location update requires a valid GPS fix.

Document Number
APN0052

Revision
1.0

Prepared By
NGB

Approved By
NB

Title
Extending Battery Life with Script-Controlled Sleep

Page
5 of 8

4. Scheduling the Next Measurement

A script can adjust the time until the next measurement cycle using the function `SQ.set_next_cycle()`. This allows the device to take frequent measurements when equipment is operating and sleep for longer periods when it is idle.

The function takes the interval until the next measurement cycle, in seconds:

```
SQ.set_next_cycle(seconds);
```

It applies to the next measurement cycle only and does not change the configured Base Interval. The script must call it again on each cycle where a longer interval is required.

The device completes its current measurements and transmission before sleeping. Sleep is possible when the scheduled interval is long enough and Device Always On is disabled.

In this example, the ignition-off interval is defined at the top of the script:

```
let off_sec = 300;
```

A value of 300 gives a five-minute interval for testing. Change the value to 43200 for a twelve-hour interval or as required.

5. Checking the Ignition State

The data handler runs when a measurement cycle completes. It receives the measurements as a JSON string, which the script converts to an object using `JSON.parse()`.

IO1 voltage is available as `analog1`. The script checks that this value is a number and compares it with the ignition threshold:

```
if (typeof obj.analog1 === 'number' && obj.analog1 < ign_v) {  
    SQ.set_next_cycle(off_sec);  
}
```

When IO1 is below the threshold, the script schedules the next cycle after `off_sec` seconds.

When IO1 is at or above the threshold, the script does not change the next cycle. The device continues using its normal five-second Base Interval. If the voltage measurement is unavailable, it also remains on the normal interval.

The script does not override transmission scheduling. With Transmit Interval set to 1, the device transmits each completed measurement cycle.

6. Waking on Ignition Change

Wake on Change allows IO1 to wake the device before the scheduled interval expires.

Ignition state or event	Device operation
Ignition on	Measure and transmit every five seconds
Ignition off	Sleep between updates at the configured interval
Ignition changes while sleeping	Wake immediately and perform a measurement cycle

For example, the QUAD may be sleeping between five-minute location updates when ignition is switched on three minutes after its previous measurement. The IO1 change wakes the device immediately, without waiting for the

remaining two minutes. The next measurement detects that ignition is on, and normal five-second operation resumes.

When ignition is switched off, the next measurement detects the low voltage and schedules the longer interval.

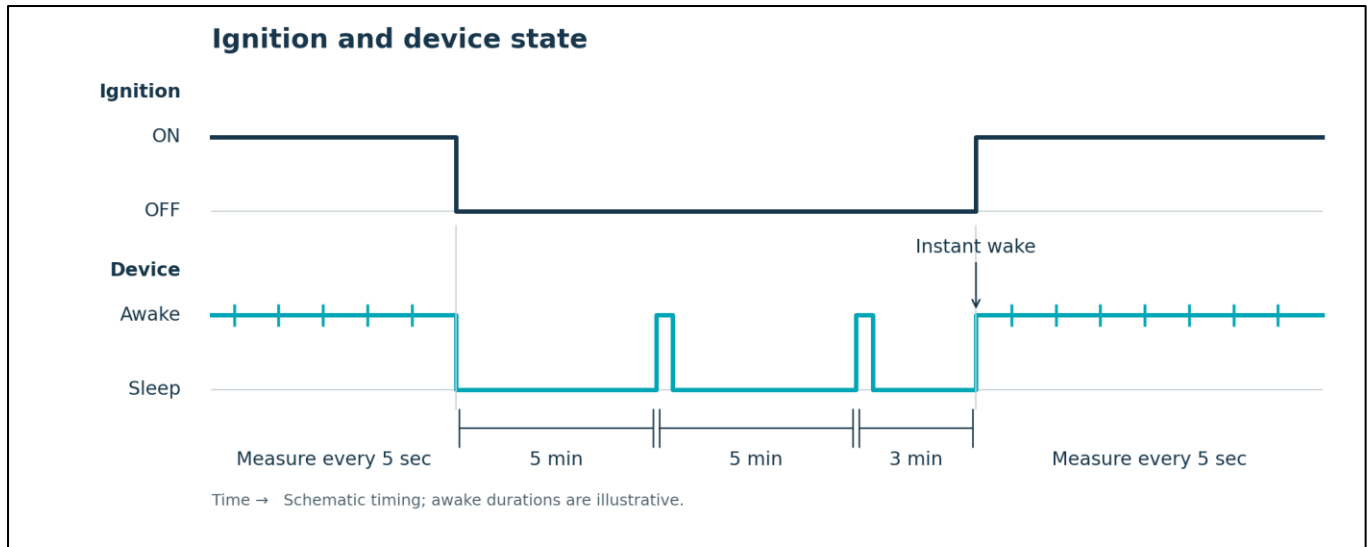


Figure 5 - QUAD ignition timing

7. Other Wake Sources

The device can also wake in response to other configured events. These can be used alongside scheduled reporting to collect data when equipment is accessed or moved.

Wake source	Example use
IO1 or IO2 digital change	Detect ignition or another external signal
Tamper sensor	Detect the enclosure being opened
Magnetic sensor in Wake mode	Initiate a measurement using a magnet
Accelerometer vibration	Detect movement or vibration, such as a trailer starting to move

Enable the required wake source in the device settings. For vibration wake, select a threshold appropriate to the installation so normal background vibration does not cause unnecessary wakes.

A wake event starts a measurement cycle. The script then evaluates IO1 again. If ignition remains off, it schedules another long interval after that cycle. A vibration or tamper event therefore causes an early report but does not, by itself, select continuous five-second operation.

8. Adapting the Method

Change off_sec to suit the required idle reporting interval and change ign_v to suit the ignition signal.

The same approach can use another measured condition to select the next interval. For example, a script could extend the interval when a pump is stopped or shorten it when a monitored value crosses a threshold.

Enabled wake sources may cause additional measurement cycles before the scheduled time. Each ignition-off cycle in this example schedules the next timed cycle relative to that cycle; the reports are not aligned to fixed clock times.

Document Number

APN0052

Revision

1.0

Prepared By

NGB

Approved By

NB

Title

Extending Battery Life with Script-Controlled Sleep

Page

7 of 8

9. Conclusion

The Senquip QUAD uses IO1 voltage to select between frequent measurements during vehicle operation and periodic reporting while ignition is off. `SQ.set_next_cycle()` extends the idle interval while configured wake sources allow the device to respond immediately to an event.

Document Number

APN0052

Revision

1.0

Prepared By

NGB

Approved By

NB

Title

Extending Battery Life with Script-Controlled Sleep

Page

8 of 8

Appendix A – Full Script

```
let off_sec = 300; // change this to the desired sleep period when ignition is off
let ign_v = 6.0;

load('senquip.js');

SQ.set_data_handler(function(data) {
  let obj = JSON.parse(data);

  // Check ignition state
  if (typeof obj.analog1 === 'number' && obj.analog1 < ign_v) {
    SQ.set_next_cycle(off_sec);
  }
}, null);
```