ULE and Battery Lifetime

SYNAPTICS CONFIDENTIAL DISCLOSED UNDER NDA
PN: 506-001241-01 REV A
Contents

1. Distinguish between Pageable and non-Pageable Devices
2. Account for various operation mode contributions
3. Provide some examples for various Use Cases
1. Pageable and non-Pageable Devices
1.1 Pageable Versus Non-Pageable Devices

- Most ULE Home Automation Devices are **non-pageable** - they are not “immediately” available for updates/re-configurations/queries from the ULE Hub (Controller). Examples of such devices are Smoke & Motion Detectors, Open/Close Detectors. They exit hibernation mode only to send maintenance packet (Keep Alive) or an Alert/Event (=Transactions). At such junctures, the ULE Hub can initiate a transaction as well. The average current of such devices is typically dictated by the interval of the Keep Alive transaction.

- Battery-Powered **Pageable** devices are also found in the home. Examples: Battery-powered Alarms or Warning Lights or Door Locks. Such ULE Devices need to wakeup and check for a page at an interval which satisfies the response latency tolerance. The average current of such devices is dictated by the wakeup interval.
1.2 Non-Pageable Device Activity Profiles

- Devices spend the large majority (>99%) of their lifetime in hibernation. In this mode, only the ULE block within the DHX91 is active, draining 1.7uA.

- ULE devices are designed with a “Keep Alive” (maintenance) function where the ULE timer is programmed to occasionally wake up the device and allow it to Sync and Exchange a single 32-Byte Data Packet with the ULE Controller. Typical Wakeup interval is 15 minutes.

- Home Automation Devices will also awake from hibernation to transfer data (e.g., temperature, humidity), Control (light On-Off) and alarms (door-window entry, smoke) with the Controller via a 32-Byte Data Packet.
1.3 Computing Battery Life for Non-Pageable Device

• Add the depletion caused by each of the 3 activity modes to obtain the Total average current:
  • Hibernate: 1.7uA x Duty Cycle (~1) = 1.7uA, average
  • Keep Alive: Charge Depletion (µC) per 32-Byte Data Transaction x Duty Cycle (Duty Cycle = # of transactions per day/86400seconds)
  • Events, Alarms, Etc: Computed the same as Keep Alive

• Note that this calculation only accounts for “ULE Communication Pipe”. Additional budget must be allocated for external sensors and/or controllers

• Battery Lifetimes are typically specified in mAh (ie AAA = 1000mAh, CR123A= 1500mAh). So to predict lifetime it is convenient to convert total average current to mAh/year. Each µA of current depletes at the rate of 8.76mAh/year!
2. Operation Mode
Contribution
2.1 Charge Depleted During 32-Byte Transaction

- Each 32-Byte transaction “costs” 2mC (and ~100mS) – See Backup slide for breakdown
- The contribution of these transactions to average current increases linearly with the # of transactions – see the tabulation below:

<table>
<thead>
<tr>
<th>charge/transfer(uC)</th>
<th>events/day</th>
<th>average current (uA)</th>
<th>Battery Drain (mAh/yr)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2000</td>
<td>10</td>
<td>0.2</td>
<td>2</td>
</tr>
<tr>
<td>2000</td>
<td>100</td>
<td>2.3</td>
<td>20</td>
</tr>
<tr>
<td>2000</td>
<td>500</td>
<td>11.6</td>
<td>100</td>
</tr>
</tbody>
</table>
3. Use Cases
3.1 Use Case Example: Motion Detector

• A motion detector with Keep Alive every 15 minutes (≈~100 time/day) and 100 detections per day:
  • 4.6µA contributed by 200 Events (per tabulation in previous page)
  • 1.7µA contributed by Hibernate
  • Total of 6.3µA avg (=55mAh/yr) for ULE “communication pipe”
  • Assume another 7µA for actual motion sensor, gives 13.3µA avg or 116mAh/year
  • Supplied by a CR123A LiOn Battery (specified with 1500mAh capacity), this corresponds to 1500/116 = 12 years

• Door and Window Magnet, Smoke Detector, Thermostat and Light Switch would have similar battery drain profiles
3.2 Current Versus Time Template for 32-Byte Transaction

X-Axis: Time, 20mS/Div
Y-Axis: Current, 50mA/Div

Device Transmits 32-byte packet
Device Receives Beacon
Device Scans for Clearance
Device Receives ACK
Hibernation

Hibernation

X: Time, 20mS/Div
Y: Current, 50mA/Div
• In addition to hibernating (99% of the time, as previously) and occasional 32-byte transactions, pageable devices wake up at short intervals to check for incoming page

• Each wakeup “costs” 75µC (new DHX101 SOC) and 6mS (Note: Its predecessor, DHX91, requires ~500µC per wakeup)

• The contribution of wakeups to average current decrease linearly with the interval between wakeups – see table below:

<table>
<thead>
<tr>
<th>Pageable Device Battery Drain as a function of latency</th>
</tr>
</thead>
<tbody>
<tr>
<td>charge/wakeup (µC)</td>
</tr>
<tr>
<td>----------------------</td>
</tr>
<tr>
<td>100</td>
</tr>
<tr>
<td>100</td>
</tr>
<tr>
<td>100</td>
</tr>
<tr>
<td>100</td>
</tr>
</tbody>
</table>
3.4 Use Case Example: Door Lock

- A Door Lock with Keep Alive every 15 minutes (~100 time/day) and 10 Open/Close Events per day and 1.28s Latency in response tolerance:
  - 1.7µA contributed by Hibernate
  - 2.3µA contributed by 100 Keep Alives
  - 0.2µA contributed by Open/Close Events
  - 65µA contributed by Wakeups (See Table on previous page)
  - Total of 69.2µA avg (=606mAh/yr) for ULE “communication pipe”
  - Supplied by a 2xAAA LiOn Battery (specified with 750mAh capacity), this corresponds to 750/606 = 1.23 Years
3.5 Pageable Device: DHX101 Measurements

- Test Platform: EVB with DHX101
- SW Image: ULE SDK 37.05, Voice Call
- Paging with PMSS Enabled
- Wakeup Interval (= Response Time, 1.28s)
3.6 Current Drain Profile: Wakeup, Check for Page

X-Axis: Time, 10mS/Div
Y-Axis: Current, 20mA/Div

Wakeup Duration ~6mS
Charge Consumed ~76uA

Charge consumed = 76uC
Duration = 6mS
3.7 Wakeup Interval: Checking for Page Every 1.28s

Wakeup Interval = 1.28s
3.8 Average Current Drain for 1.28s Wakeup Interval

Average Current = 65uA
## Revision History

<table>
<thead>
<tr>
<th>Revision</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Initial release.</td>
</tr>
<tr>
<td>A</td>
<td>Initial production release.</td>
</tr>
</tbody>
</table>
