Introduction

Connected Standby is a new feature introduced by Microsoft in Windows 8* for SOC-based platforms. The use case on the tablet/mobile systems is similar to that on phones like Instant ON and Always Connected. Intel® dual-core Atom™ Z2760 (code name CloverTrail) is the first Intel platform to support Connected Standby. Connected Standby is critical to achieve the lower power targets for many devices and be able to demonstrate power performance leadership. Interesting applications can be developed for Connected Standby to further showcase the platform capabilities like push email, multiplayer games, etc.

In this sample application, we showcase the use of Connected Standby and demonstrate the use of the API to register the background task with the timer for periodic execution in the background, so when the system enters standby, the task executes and generates an execution log. It is a simple interface and application for any ISV to use inside their apps.

How it works

Connected Standby is primarily designed for very low power consumption and achieving the battery targets. It is part of the Microsoft certification, and devices need to conform to the requirement set by Microsoft for Connected Standby. Devices enter Connected Standby when the “on” button is pushed or after idle timeout. Based on current testing, the power is < 100 mW1. This is the requirement set by Microsoft. Instant On is about < 300ms1 from button press to display on. Overall, the platform is mostly asleep; only select apps execute occasionally. Figure 1 shown below represents the state representation in form of a flowchart, designed to illustrate the system transitions. Figure 2 shown below shows the more detailed view of the transitions and the events associated with each transition.
Figure 1: State Transition

- **Screen On (Active)**: User present and using
- **Screen Off (Connected Standby)**: User not present, still connected
- **OFF**
Applications Conforming to Connected Standby

Microsoft has designed the API and all the various interfaces for applications to use the Connected Standby feature and be power efficient even when applications are running. Windows® UI and Store apps are designed to be power efficient from the ground up. Windows Store apps (using the new Windows UI features) are suspended when the device enters Connected Standby. Microsoft has created the concept of background tasks that run while the device is in standby. Windows Store Apps can include background tasks that run in connected standby. The included background task/tasks run even if the corresponding application is suspended or has exited. To save power, Windows 8 imposes tight restrictions on background tasks. Figure 3 illustrates the concept.

- Background tasks either run:
  - Periodically (less than once every 15 min)
- When triggered by an event: Network becomes available, network event, push notification

- Background tasks have limited CPU time to run:
  - 7 “lock screen” apps get 2 sec every 15 mins
  - Non “lock screen” apps get 1 sec every 2 hrs

- Limited network bandwidth:
  - Lock screen apps get 4.69 MB every 15 min (daily max of 450 MB)
  - Non Lock screen apps get 6.25 MB every 2 hrs (daily max of 75 MB).

---

**Figure 3: Application in Connected Standby**
## Writing Background Tasks

<table>
<thead>
<tr>
<th>Trigger Type</th>
<th>Description</th>
<th>Lock screen app only</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time Trigger</td>
<td>Time event</td>
<td>Yes</td>
</tr>
<tr>
<td>Control Channel trigger</td>
<td>Data on open TCP channel</td>
<td>Yes</td>
</tr>
<tr>
<td>Push notification trigger</td>
<td>Data from Windows* Push Service</td>
<td>Yes</td>
</tr>
<tr>
<td>System trigger</td>
<td>Various network and system events</td>
<td>No</td>
</tr>
<tr>
<td>- Network up</td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Network down</td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Network status</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Maintenance trigger</td>
<td>Same time trigger, except fires only when device is plugged in</td>
<td>No</td>
</tr>
</tbody>
</table>

## Registering a timed task

Below is a code snippet based on the Microsoft classes and APIs that allow developers to register a timed task with the application builder class. This can be used by any Windows 8 application to register the timed task with the trigger value specified. The trigger value is taken in seconds. You can refer to the MSDN for further information regarding the API and the classes (http://msdn.microsoft.com/en-us/library/windows/apps/br224847.aspx)

```c++
using namespace Windows::ApplicationModel::Background;

//Associate an app with a trigger
auto builder = ref new BackgroundTaskBuilder();
builder->Name = "SampleBackgroundTask";
builder->TaskEntryPoint = "Tasks.SampleBackgroundTask";

//Create a time Trigger
IBackgroundTrigger^ trigger = new TimeTrigger(15, false);
```
builder->SetTrigger(trigger);

//Register task with OS
builder->Register();

**Background Task Execution**

The following code snippet shows instantiating the background task, which then executes when the event is triggered. This is the same task that was created earlier and registered with the builder class. The Sample Background task’s “run” method can be populated with the required execution flow.

```cpp
using namespace Windows::ApplicationModel::Background;

namespace Tasks
{    public ref class SampleBackgroundTask sealed:public IBackgroundTask
{
    ...
    virtual void Run(IBackgroundTaskInstance^ taskInstance) {
        // Code to be executed in the background
    }
    ...
};
}
```

The task generation is part of the samplebackground.xaml.cpp file, and the timetriggeredtask.xaml.cpp has the code for the timed task.

**Summary**

The sample code showing a background registered task and its usage and flow. When the system enters Connected Standby, the background task will generate messages that are sent to the lock screen to show as notifications, thus demonstrating the capability of Connected Standby.
Notices

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF INTEL PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

UNLESS OTHERWISE AGREED IN WRITING BY INTEL, THE INTEL PRODUCTS ARE NOT DESIGNED NOR INTENDED FOR ANY APPLICATION IN WHICH THE FAILURE OF THE INTEL PRODUCT COULD CREATE A SITUATION WHERE PERSONAL INJURY OR DEATH MAY OCCUR.

Intel may make changes to specifications and product descriptions at any time, without notice. Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The information here is subject to change without notice. Do not finalize a design with this information.

The products described in this document may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Contact your local Intel sales office or your distributor to obtain the latest specifications and before placing your product order.

Copies of documents which have an order number and are referenced in this document, or other Intel literature, may be obtained by calling 1-800-548-4725, or go to: http://www.intel.com/design/literature.htm

Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests, such as SYSmark* and MobileMark*, are measured using specific computer systems, components, software, operations, and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products.

Any software source code reprinted in this document is furnished under a software license and may only be used or copied in accordance with the terms of that license.

Intel and the Intel logo are trademarks of Intel Corporation in the US and/or other countries.

Copyright © 2012 Intel Corporation. All rights reserved.

*Other names and brands may be claimed as the property of others.