

Источник - Сергей Звездин: персональный блог (<http://blog.zvezdin.com/ru/245>)

## Приложения для Windows Phone 7, зависящие от источника питания

---

Не всегда мобильное приложение удастся сделать экономичным по отношению к батарее. Яркий пример таких приложений – автомобильная навигация. В этом случае для предотвращения быстрого разряда батареи следует использовать внешнее питание. В этой заметке я покажу каким образом отслеживать такие ситуации.

В случаях, когда сценарии использования нашего приложения предполагают наличие внешнего питания, следовало бы предупредить пользователя об этом. Напрашивается следующая логика: перед запуском ресурсоемкого программного кода мы должны проверить присутствует ли внешний источник питания и, если нет, то предупредить пользователя о том, что батарея может быть быстро истощена. То же самое следует проделывать всякий раз, когда внешнее питание отключается.

Эту логику просто реализовать, используя класс [DeviceStatus](#). Для наших целей он содержит:

- свойство **PowerSource** – позволяет определить текущий источник питания (батарея или внешний);
- событие **PowerSourceChanged** – позволяет отловить момент, когда источник питания изменяется.

Теперь мы можем построить требуемую для нас логику. Предположим, что у нас есть два метода – `StartWorkingHard()` и `StopWorkingHard()`, которые запускают и останавливают запись GPS-трека. Тогда код для отслеживания источника питания может выглядеть так:

```
01 public MainPage ()
02 {
03     InitializeComponent ();
04
05     DeviceStatus.PowerSourceChanged += PowerSourceChanged;
06     CheckPowerStatus ();
07 }
08
09 private void PowerSourceChanged(object sender, EventArgs e)
10 {
11     CheckPowerStatus ();
12 }
13
14 private void CheckPowerStatus ()
```

```
15 {
16     if (DeviceStatus.PowerSource == PowerSource.Battery)
17     {
18         if (MessageBox.Show("Используется питание от батареи. Продолжить запись трека?",
19 "Изменен источник питания", MessageBoxButtons.OKCancel) == DialogResult.OK)
20         {
21             StartWorkingHard();
22         }
23         else
24             StopWorkingHard();
25     }
26     else
27         StartWorkingHard();
28 }
29 private void StartWorkingHard()
30 {
31 }
32
33 private void StopWorkingHard()
34 {
35 }
```