

# Sensor & Location platform Windows 7

## HOE MAAK JE CONTEXT-AWARE APPLICATIES VOOR WINDOWS 7?

Anko Duizer

Een nieuw onderdeel van de Windows 7 SDK is het Sensor & Location platform. Programmeren met sensoren is niet alleen zeer interessant en toepasbaar, maar ook vooral erg leuk. Dankzij dit platform kunnen applicaties op een uniforme wijze bekend worden gemaakt met hun omgeving, de zogenaamde context-aware applicaties. Er is ook een managed API beschikbaar waardoor het platform zeer toegankelijk is voor iedere .NET ontwikkelaar.

Dat Windows 7 veel nieuws biedt voor de ontwikkelaar is inmiddels wel bekend. Een zeer interessant onderdeel van al dit nieuws is het sensor & location platform. Met dit platform wordt het op een eenvoudige manier mogelijk om applicaties zich zelf te laten aanpassen aan de omgeving. Met alle hightech mobiele devices, die we tegenwoordig hebben, is dit een zeer interessant gegeven. Afhankelijk van de hoeveelheid licht kan jouw applicatie zich bijvoorbeeld aanpassen door de resolutie, de fonts en het contrast te veranderen waardoor de gebruiker beter kan blijven werken met de desbetreffende applicatie in de zon.

Het sensor & location platform geeft een standaard manier om diverse sensoren (devices) te integreren met Windows 7. Het platform zorgt ook voor een standaard programmeermodel waar applicaties en hardware leveranciers gebruik van kunnen maken.

### Historie

Wat is er anders? Je kon toch altijd al sensoren aanspreken vanuit je applicaties? Niets nieuws, of toch? In het recente verleden was het zeker mogelijk om sensoren aan te spreken. In de praktijk gebruikte je hiervoor vaak een COM-poort, waarvan het aantal beperkt is. Daarbij had ieder device weer zijn eigen specifieke API. Ondanks het managed tijdperk was dit veelal een Win32 API, lastig in het gebruik.

Al met al was het niet eenvoudig om een universeel programmeer model te introduceren. Het gevolg hiervan is dat er nauwelijks gebruikt wordt gemaakt van sensoren in applicaties. Dit is natuurlijk jammer omdat de voordelen en de mogelijkheden evident zijn. Met het sensor & location platform zou dit wel eens kunnen gaan veranderen!

### Sensor & Location platform

Het sensor & location platform is onderdeel van de Windows 7

SDK. De belangrijkste doelstelling van het platform is zorgen dat de ontwikkelaars op een uniforme wijze kunnen omgaan met sensoren in een applicatie. Aan de andere kant zorgt het platform ook voor een uniforme API voor de hardware leveranciers.

Zoals voor de meeste Windows 7 API's geldt, is ook de sensor & location API een native API. Dit betekent concreet dat C++ en COM programmering de standaard manier is om het platform aan te spreken.

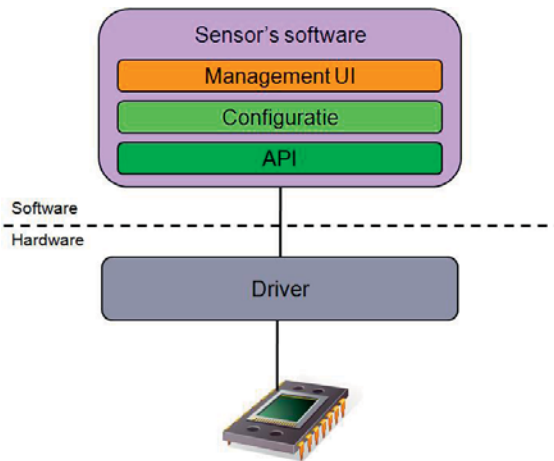
Het platform maakt het mogelijk dat een applicatie ontwikkelaar sensoren kan aanspreken en daardoor hun gedrag kan integreren in een applicatie. Denk hierbij bijvoorbeeld aan GPS (waar ben ik?), ambient (Hoeveel licht is er?), accelerator (beweging) of touch sensoren. Met deze sensoren is het mogelijk om een applicatie bekend te maken met de omgeving. Afhankelijk van de locatie kunnen er andere adressen, in de buurt worden getoond (location based services). Afhankelijk van de hoeveelheid licht kan de applicatie worden aangepast voor een beter beeld. De accelerator en touch sensoren kunnen dienen als (alternatieve) input voor een applicatie.

Het platform kan met diverse sensoren omgaan. Een sensor meet een fysiek fenomeen of een fysieke interactie. Het platform organiseert sensoren (type) in een aantal categorieën. Een categorie representeert een groep van gemeenschappelijke sensoren. Een 3-D accelerometer device als onderdeel van een game device valt binnen de categorie: Orientation sensor. De categorieën en sensoren worden in code gerepresenteerd door een GUID, die veelal zijn voorgedefinieerd door Windows 7. Een hardware leverancier kan een nieuw type en categorie toevoegen aan het platform wanneer dit noodzakelijk is.

Sensor type	Sensor positie	Gebruikt voor?
GPS	In de computer of verbonden met de computer	Locatie en oriëntatie
Accelerometer	In de computer of verbonden met een persoon	Beweging, locatie en oriëntatie
Light	In de computer, of verbonden dichtbij of op afstand	Licht condities, zoals zonlicht of een donkere presentatie ruimte
RFID	In de computer of verbonden met een persoon	Identificatie van fysieke onderdelen, automatische logon
Compass	Verbonden aan een persoon	Oriëntatie
Camera	In de computer, of verbonden aan persoon of op afstand	Opname, object identificatie
Microphone	In de computer, of verbonden aan persoon of op afstand	Geluid meten
Temperature	In de computer, of op afstand	Temperatuur meten
Motion detector	In de computer, of op afstand	Beweging in de buurt van de computer, beveiliging

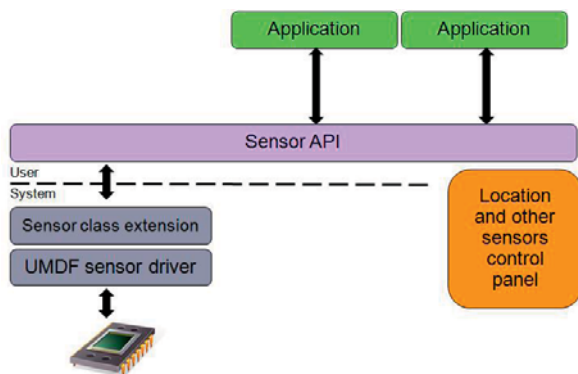
TABEL 1: VOORBEELDEN VAN SENSOREN.

## Architectuur



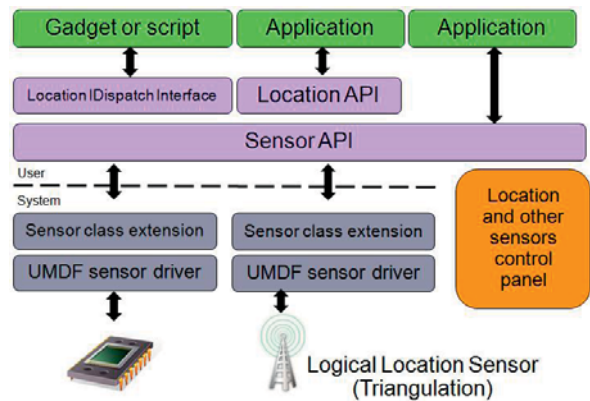
FIGUUR 1: TRADITIONELE SENSOR DRIVER ARCHITECTUUR.

In het recente verleden werd er volgens een architectuur gewerkt zoals weergegeven in figuur 1. Iedere sensor had een specifieke driver met een eigen API en configuratie. Communicatie met sensoren was dus mogelijk, maar niet eenvoudig.



FIGUUR 2: SENSOR PLATFORM ARCHITECTUUR.

Het sensor & location platform lost dit anders op door een beproefde architectuur te gebruiken. Een architectuur gebaseerd op lagen, waaronder de drivers werken. Voor de ontwikkelaar is er slechts één API beschikbaar, wat zorgt voor een uniform programmeermodel. Aan de "onderkant" moeten hardware leveranciers voldoen aan de door Microsoft gespecificeerde API. Een bekend driver model zoals we dat bijvoorbeeld ook kennen van ADO.NET. In figuur 2 is dit schematisch weergegeven. Het sensor & location platform werkt volgens dit zelfde driver model (zie figuur 3). Bovenop de Sensor API is echter een verbijzondering gemaakt voor locatie, waaraan bijvoorbeeld de concrete vraag kan worden gesteld: "Waar ben ik?".

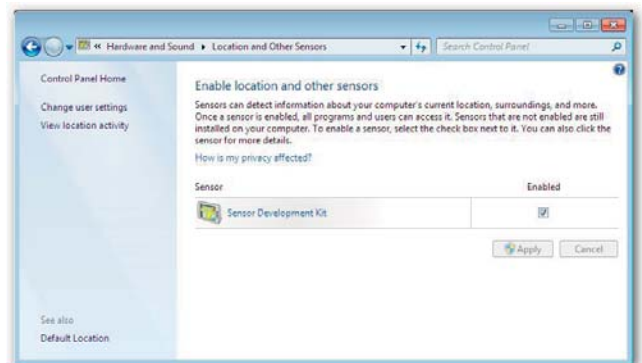


FIGUUR 3: LOCATION PLATFORM ARCHITECTUUR.

In basis is het werken met sensoren gebaseerd op een event architectuur. Wanneer nieuwe data beschikbaar komt, dan wordt een event afgevuurd waarop de applicatie kan reageren. Denk hierbij bijvoorbeeld aan status veranderingen zoals 'connected/disconnected' and 'active/deactive'. Hoe vaak een event wordt afgevuurd is veelal configureerbaar per sensor. Wanneer een event is afgevangen door de applicatie kan er aan de desbetreffende sensor aanvullende informatie worden gevraagd.

Het platform zorgt ervoor dat meerdere applicaties gebruik kunnen maken van dezelfde sensor. Er ontstaan geen problemen met COM poorten, zoals in het verleden. De architectuur is uitbreidbaar en het aantal sensoren dat werkt binnen het platform zal sterk toenemen.

Als onderdeel van de architectuur is in figuur 2 en 3 een control panel weergegeven. Dit control panel is onderdeel van Windows 7. Hierin wordt duidelijk dat het sensor & location platform volledig is geïntegreerd in het operating systeem. Via het control



FIGUUR 4: ENABLING SENSOREN IN WINDOWS 7.

# feel free



# to innovate

**XCESS**  
expertise center

Dertien-in-een-dozijn developers zijn er al genoeg. Doorsnee ICT-bedrijven trouwens ook.

Bij XCESS mag je je onderscheiden, want eigenwijs, dat zijn we zelf ook. Feel free to join!

[www.xcess.nl](http://www.xcess.nl)

panel heeft de gebruiker controle over de sensoren, en kunnen sensoren expliciet worden 'enabled' of 'disabled'. Een voorbeeld van het control panel is weergegeven in figuur 4. Er is een API call beschikbaar om het control panel op te roepen vanuit een applicatie, wanneer een sensor moet worden 'enabled' voor de werking van de applicatie.

## Kan ik het al gebruiken?

Er is een Windows SDK beschikbaar, waarvan het sensor & location platform een onderdeel is. De SDK bevat inmiddels de nodige documentatie en zijn er diverse voorbeelden beschikbaar. Wanneer je aan de slag gaat met de SDK, dan kom je al snel tot de conclusie dat de API een native is. Dit betekent, zoals eerder vermeld, dat je aan de slag moet met C++ en COM.

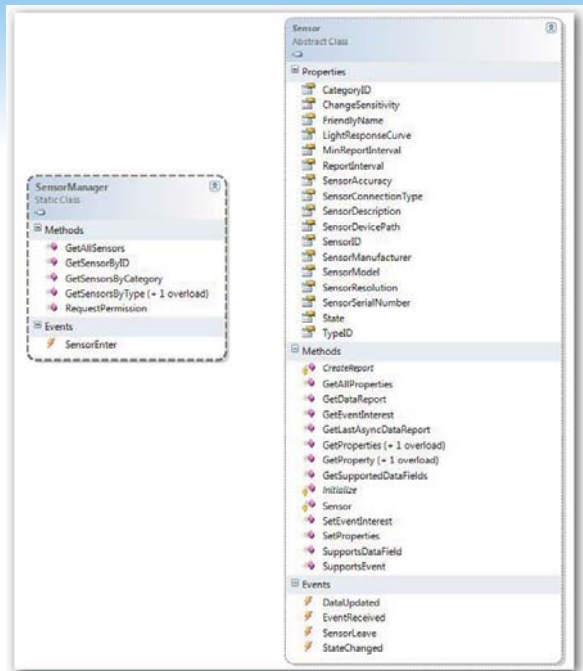
Voor sommige ontwikkelaars zal dit vertrouwd voelen, en zeker niet als een probleem. Voor de gemiddelde .NET ontwikkelaar is dit toch wel een teleurstelling. Kan dit niet anders? Gelukkig is er een kleine managed library beschikbaar gekomen boven op de standaard API waarmee wel degelijk managed tegen het platform kan worden geprogrammeerd. Deze .NET interop library moet wel apart worden geïnstalleerd.

Om het sensor & location platform te leren kennen, moet je er mee kunnen spelen. Hiervoor is een interessante sensor beschikbaar van Freescale. Menigeen heeft dit bord ontvangen als onderdeel Windows 7 Sensor Development Kit die op diverse plaatsen is uitgedeeld. Wanneer je nog geen bord hebt, dan kun je deze voor een klein bedrag alsnog aanschaffen. Het bord heeft verschillende sensoren zoals een beweging, licht en touch sensor.

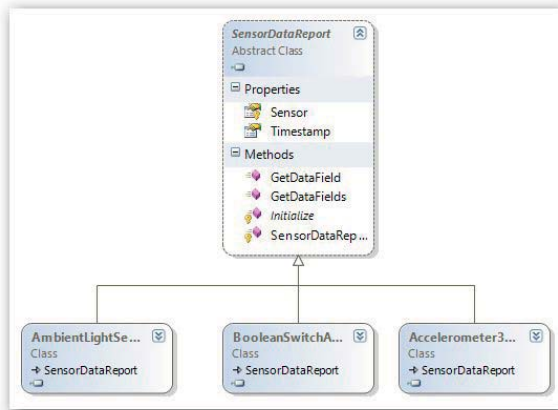
## Hoe te programmeren?

Wanneer je de .NET interop library voor het platform heb geïnstalleerd, kun je meteen aan de slag. Om tegen een sensor aan te programmeren zal je eerst een instantie moeten hebben, die de desbetreffende sensor representeert. Om tot een dergelijke pointer in code te komen is de SensorManager vaak het startpunt. De SensorManager is een static class die onderdeel is van de library. Via de SensorManager kunnen eenvoudigweg alle sensoren worden opgehaald, maar ook per categorie, type of zelfs een unieke sensor (via een GUID). De SensorManager kan er ook voor zorgen dat de gebruiker wordt geconfronteerd met de standaard gebruikersinterface om een sensor te 'enablen'. Verder vuurt de SensorManager een event af wanneer een sensor beschikbaar komt. Kortom, de SensorManager is de start van de meeste applicaties die gebruik willen maken van het sensor & location platform. De SensorManager class is weergegeven in figuur 5.

Iedere sensor wordt gerepresenteerd door het type Sensor (zie figuur 5). Zodra er een sensor beschikbaar is kan er data worden verkregen middels een abstracte class SensorDataReport. Deze class wordt verbijzonderd per sensor. Deze verbijzondering is logisch daar een licht sensor andere informatie terug geeft dan bijvoorbeeld een touch sensor. Zoals in figuur 6 is te zien is de SensorDataReport in de library een drietal keren verbijzonderd, bijvoorbeeld voor een ambient light sensor.



FIGUUR 5: DE SENSORMANAGER EN SENSOR CLASS.



FIGUUR 6: DE SENSORDATAREPORT CLASS.

In voorbeeldcode 1 is een standaard programmeerstructuur weergegeven om gebruik te maken van het sensor & location platform. In de form load wordt via de SensorManager een lijst opgehaald met alle beschikbare sensoren. De gebruiker kiest vervolgens via een combobox de gewenste sensor. Na een button click wordt bepaald welke sensor er is geselecteerd. In de voorbeeldcode wordt gewerkt met een AmbientLightSensor, alleen dit type sensor wordt dan ook geaccepteerd. Wanneer de sensor klaar is wordt er data opgehaald. Hierbij moet de data wordt gecast naar een specifiek type: AmbientLightSensorDataReport. Nu de data binnen is, kan deze worden gebruikt binnen de applicatie. Andere sensoren volgen precies hetzelfde stramien.

```
private void Form1_Load(object sender, EventArgs e)
{
    comboBoxSensors.Items.Clear();
    foreach(Sensor s in SensorManager.GetAllSensors())
    {
        comboBoxSensors.Items.Add(s);
        comboBoxSensors.DisplayMember = "FriendlyName";
    }
}

private void button1_Click(object sender, EventArgs e)
{
    Sensor s = (Sensor)comboBoxSensors.SelectedItem;
```

```
if (s.GetType() != typeof(AmbientLightSensor))
{
    MessageBox.Show("Sensor is not an ambient light sensor");
    return;
}

switch (s.State)
{
case SensorState.Ready:
    AmbientLightSensorDataReport datareport =
        (AmbientLightSensorDataReport)s.GetDataReport();
    labelDataReport.Text = "Luminance: "
        + datareport.IlluminanceLux + " LUX";
    break;
case SensorState.Error:
    MessageBox.Show("Sensor error...");
    break;
default:
    break;
}
}
```

VOORBEELDCODE 1: GEBRUIK MAKEN VAN SENSOREN

Wat zijn de gevolgen voor je applicatie wanneer je het sensor & location platform gaat gebruiken? Wat zijn de mogelijkheden? In de basis geeft het platform je de gelegenheid om je applicatie bekend te maken met de omgeving, een zogenaamde 'context aware applicatie'. Dit betekent dat de applicatie zich kan aanpassen op basis van omgevingsinformatie die binnenkomt via de sensoren. In het geval van een GPS sensor kunnen we dit ons voorstellen. Menig ontwikkelaar zal een device in zijn of haar auto hebben waarop een context gevoelige GPS applicatie draait. Z'ou applicatie laat bijvoorbeeld de tankstations of hotels in de omgeving zien. Een mooi voorbeeld waarbij gebruik wordt gemaakt van een sensor die in staat is de content van de applicatie aan te passen. Maar er zijn ook andere voorbeelden denkbaar, zeker bij mobiele devices zoals een laptop. Een laptop kun je overal mee naar toe nemen: binnen, buiten, vergaderzaal, conferentieoord, etc. Een bekend probleem bij de laptop schermen is dat de leesbaarheid van het scherm nogal wisselt naar gelang er meer of minder licht op het scherm valt. Nu stelt iedere gebruiker dit handmatig in. Kun je hier nog meer aan doen? Op basis van een ambient light sensor zou je de hoeveelheid licht kunnen meten. Wanneer er veel licht is, bijvoorbeeld in de zon, kun je het font groter maken, de resolutie veranderen en de kleuren in je applicatie meer contrasterend laten zijn.

Vandaag de dag kan dit ook allemaal zonder het sensor & location platform, en toch gebeurt het niet. Met het platform is het in Windows 7 eenvoudig om gebruik te maken van de sensoren. De eenvoud trekt ontwikkelaars over de streep. En ook hier geldt het spreekwoord: "Wanneer er één schaap over de dam is, volgen er meer". De daad is aan ons om het platform te gaan gebruiken en creatieve ideeën te verzinnen en te realiseren om sensoren goed in te passen in onze applicaties. Ik kan haast niet wachten...

Referenties

<http://www.microsoft.com/whdc/device/sensors/default.aspx>  
<http://msdn.microsoft.com/en-us/library/dd318953%28VS.85%29.aspx>  
[http://www.freescale.com/webapp/sps/site/prod\\_summary.jsp?code=JMBADGE2008-B](http://www.freescale.com/webapp/sps/site/prod_summary.jsp?code=JMBADGE2008-B)

**Anko Duizer**, is werkzaam als trainer/coach bij Class-A. Daarvoor heeft hij 5 jaar gewerkt bij Microsoft als consultant. Sinds begin 2001 is hij bezig met .NET. Anko volgt nieuwe ontwikkeling op de voet, waaronder Windows 7 en het Azure Services Platform. Hij is te bereiken op [anko.duizer@class-a.nl](mailto:anko.duizer@class-a.nl) en blogt op <http://www.ankoduizer.nl/>