



Info-display voor thuis

met Windows op de Raspberry Pi

Dr. Veikko Krypczyk (Duitsland)

Informatiesystemen tonen informatie over het weer, het laatste nieuws, de vertrektijden van de metro om de hoek of andere actuele gegevens. Zulke informatiesystemen zijn ideaal voor gebruik in de huiskamer en op het werk. Een slimme keuze van hard- en software maakt het systeem flexibel.

Informatiesystemen zijn in de mode en kunnen voor allerlei doelen worden gebruikt, zowel privé als voor het werk. Een eigen Home Information System of kortweg *HomeInfoSystem* zou de gebruiker bijvoorbeeld de volgende informatie kunnen verschaffen:

- het weer hier en nu, en indien nodig, een prognose voor de komende dagen;
- nieuws: lokaal, nationaal of globaal, eventueel beperkt tot een specifiek onderwerp zoals sport of cultuur;
- foto's uit de cloud, gepresenteerd in de vorm van een diashow;
- programmeergegevens van TV of de bioscoop;

- de vertrektijden van tram en bus, met actuele informatie over mogelijke vertragingen;
- waar de gezinsleden zijn (als hun smartphone dat ondersteunt);
- afspraken uit de agenda, nu en straks.

Ook voor het bedrijfsleven (*EnterpriseInfoSystem*) zijn allerlei toepassingen denkbaar. Hier zijn enkele voorbeelden:

- weergave van werkplannen en afspraken, automatisch uit een teamkalender in de cloud opgehaald;
- gewenste en werkelijke productieparameters;

- actuele interne bedrijfsinformatie;
- informatie voor klanten, deelnemers aan evenementen – enzovoort.

Deze lijst kan naar believen worden uitgebreid. U kunt vast nog veel meer interessante toepassingen bedenken. Tot nu toe waren de voorbeelden alleen gericht op een passieve presentatie van informatie, maar er kunnen ook interactieve functies worden geïntegreerd, bijvoorbeeld met behulp van een aanraakscherm.

Dergelijke informatiesystemen kunnen kant-en-klaar worden gekocht of zelf worden gebouwd. De juiste interactie tussen hard- en software is daarbij erg belangrijk. Als u uw eigen systeem bouwt, blijft u flexibel; u kunt het systeem aanpassen aan uw eigen behoeften en u bent niet afhankelijk van de software van een provider. In dit artikel bespreken we de aanpak voor een eenvoudig realiseerbaar en modificeerbaar HomeInfoSystem (HIS). De software wordt op [1] gratis beschikbaar gesteld voor gebruik en verdere ontwikkeling. We lichten het principe toe aan de hand van webservices (weer, nieuws, foto's), waarna u het HomeInfoSystem naar eigen behoefte kunt gaan uitbreiden.

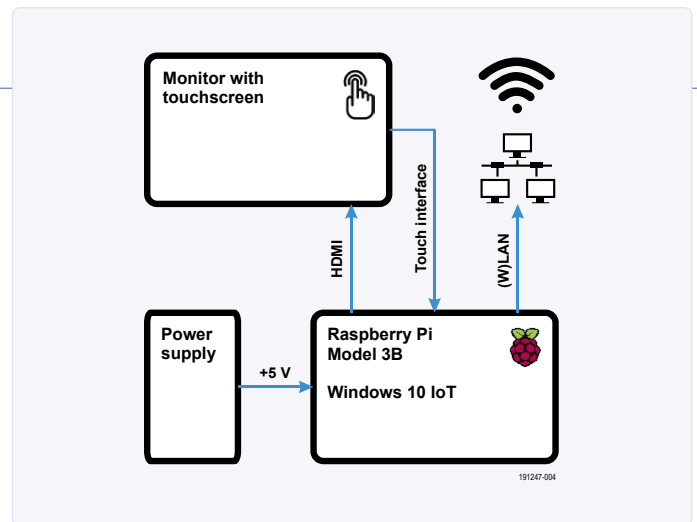
Opzet van het systeem

Het systeem moet geschikt zijn voor continubedrijf (24/7). Het energieverbruik moet dus zo laag mogelijk zijn, ongeacht de grootte van het scherm. Daarom is het raadzaam om een mini-PC in de vorm van een enkelkaartcomputer te gebruiken, bijvoorbeeld een Raspberry Pi, een Rock Pi 4, een Asus Tinkerboard S of een Banana Pi M3. Voordelen van de Raspberry Pi zijn de prijs, de prestaties, de goede ondersteuning voor de beoogde taak en vooral onze uitgebreide ervaring met deze minicomputer. In **figuur 1** ziet u de structuur van het systeem. Het is opgetrokken rond de Raspberry Pi. Model 3B is het meest geschikt, maar ook het oudere model 2B is te gebruiken, omdat Windows 10 IoT als besturingssysteem wordt gebruikt. Het nieuwste model 4 presteert weliswaar meer, maar wordt niet officieel ondersteund door Windows 10 IoT.

In het blokschema in figuur 1 is te zien dat de Raspberry Pi zoals gebruikelijk wordt gevoed uit een goede, gestabiliseerde netadapter die 5 V en werkelijk (!) 2 A kan leveren. De netwerkverbinding kan bedraad (LAN) of draadloos (WLAN) zijn. Voor het display hebben we een ruime keuze qua grootte, resolutie en de beschikbaarheid van een touch-interface. De grootte van het scherm is natuurlijk afhankelijk van de beoogde toepassing. Die kan variëren van een klein scherm dat op tafel ligt tot een infodisplay dat ook op grotere afstand gelezen kan worden. De Raspberry Pi (model 2 en 3) haalt een maximale resolutie van 1920x1080 pixels (full HD). Een hogere resolutie geeft een scherper beeld en er kan *meer* content worden weergegeven, maar natuurlijk moet u bij uw keuze ook rekening houden met de kosten, zeker als het om een aanraakscherm gaat. Of zoiets nodig is, hangt van verschillende criteria af:

- kan/moet de gebruiker/kijker iets selecteren?
- is er sprake van interactie?
- is het scherm binnen reikwijdte van de gebruiker?

De verbinding tussen de Raspberry Pi en de monitor gaat altijd via een HDMI-kabel. Als er een aanraakscherm wordt gebruikt, is er een extra verbinding tussen de monitor en de Raspberry Pi nodig. De Raspberry Pi maakt gebruik van een USB-poort, maar de aansluitingen van de monitor kunnen anders zijn. Nieuwere modellen gebruiken meestal een USB-C-poort.



Figuur 1. De opzet van het HomeInfoSystem (HIS).

De monitor moet natuurlijk ook van stroom worden voorzien. Voor mijn HomeInfoSystem gebruik ik een mobiel full-HD aanraakscherm met een beeld diagonaal van 13 inch en met een HDMI-aansluiting, USB-C voor touch, USB-C voor de voeding en een ingebouwde accu. Het is bijvoorbeeld mogelijk om de Raspberry Pi 'piggyback' op een grote monitor te monteren en deze aan de muur of het plafond te bevestigen. De HDMI-kabel kan dan heel kort zijn.

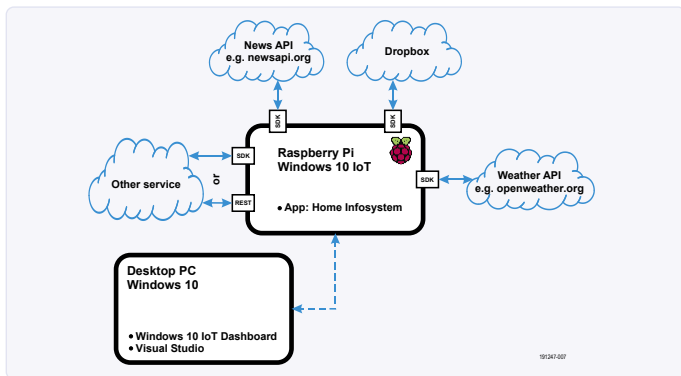
Architectuur van de software

Het besturingssysteem dat op de Raspberry Pi is geïnstalleerd is Windows 10 IoT. Dit biedt een aantal voordelen bij het opzetten van een informatiesysteem:

- actieve en directe kioskmodus; dat wil zeggen dat een app direct wordt uitgevoerd met beperkte toegangsrechten voor de gebruiker;
- uitgebreide mogelijkheden voor het ontwerpen van de gebruikersinterface in de beschrijfstaal XAML;
- eenvoudig beheer van de Raspberry Pi via de grafische webinterface;
- ontwikkelen, testen en inzetten direct vanuit de geïntegreerde ontwikkelomgeving Visual Studio;
- automatisch updaten van de app mogelijk via de store (scenario voor commercieel gebruik).

Op de Raspberry Pi installeert u eerst Windows 10 IoT (zie **kader**). Voor de ontwikkeling gebruikt u een PC (desktop/notebook) met een actuele versie van Windows 10. Daarop wordt de app gemaakt en ook getest. Visual Studio wordt gebruikt als ontwikkelomgeving. Alleen cross-device apps voor de Universal Windows Platform (UWP) runtime-omgeving kunnen worden uitgevoerd op Windows 10 IoT, klassieke Windows-applicaties draaien niet. Apps voor UWP zijn gebaseerd op .NET Core. Ze worden met behulp van de ontwikkelomgeving Visual Studio geschreven in C# of in Visual Basic .Net. De gratis community-editie [2] is voldoende.

Selecteer in het installatieprogramma van Visual Studio de juiste workload voor het ontwikkelen van apps voor UWP. Dan kunt u beginnen aan de eerste app. Vergeet niet het applicatietype te wijzigen van X86 naar ARM, want de Raspberry Pi heeft een ARM-processor. De apps worden volledig getest op de ontwikkelcomputer, en pas als een versie foutloos draait, wordt het zogenaamde app-pakket gemaakt en via de webinterface geïnstalleerd op de Raspberry Pi. Tijdens de



Figuur 2. De HomeInfoSystem-software van het systeem.

ontwikkeling is het ook mogelijk om de app meteen op de Raspberry Pi te draaien voor het testen en debuggen. Het enige wat nodig is voor dit debuggen op afstand is dat beide computers van hetzelfde lokale netwerk deel uitmaken. Omdat apps voor UWP gebaseerd zijn op .NET-Core, zijn alle mogelijkheden van de bijbehorende API's beschikbaar. Voor veel standaardproblemen kunt u gebruik maken van kant-en-klare klassen. Als programmeertaal kunt u kiezen tussen C# en Visual Basic .NET.

Een app draait op de Raspberry Pi op de voorgrond en in schermvullende modus (kioskmodus). De gebruikersinterface is declaratief ontworpen in de beschrijvingstaal XAML. Daarbij worden voorgedefinieerde besturingselementen in layout-containers geplaatst. Op die manier wordt een relatieve layout gecreëerd die zich automatisch aanpast aan verschillende schermafmetingen en resoluties. Het scala aan beschikbare bedieningselementen omvat de gebruikelijke elementen zoals knoppen en tekstvelden. Start een nieuw project, ontwerp een minimale gebruikersinterface, voer de app uit op uw PC om die te testen en breng het app-pakket vervolgens over naar de Raspberry Pi. Daarmee is de ontwikkelcyclus voltooid.

Het belangrijkste element van het softwaresysteem is de app *HomeInfoSystem* (figuur 2), die met een app-pakket via de webinterface op de Raspberry Pi wordt geïnstalleerd. De app wordt vooral gebruikt om de content te presenteren: de gegevens die bij de externe services worden opgehaald. Uitzonderingen zijn statische functies en zaken zoals de tijdweergave.

In een menu aan de linkerkant van de app kan men kiezen welke inhoud wordt weergegeven. De app schakelt automatisch tussen de verschillende content om met een instelbaar tijdsinterval. In de actieve presentatiemodus is het menu volledig verborgen. Alleen het zogenaamde hamburger-menu-pictogram  aan de bovenrand herinnert eraan dat het opengeklapt kan worden. De automatische omwisseling van de inhoud kan voor elke service afzonderlijk worden geconfigureerd. De weergave van de foto's uit de Dropbox mag gerust iets langer duren (bijvoorbeeld 45 s), terwijl voor het weer 20 s wel voldoende zal zijn. Content en services kunnen op drie manieren in de app worden geïntegreerd.

- **Intern:** de functionaliteit haalt geen extra gegevens (zoals de weergave van de tijd) uit de buitenwereld. De hele functie inclusief de gegevens is volledig statisch geïntegreerd in de broncode van de app.
- **Extern (SDK):** voor veel services kunt u gebruik maken van

diensten van aanbieders. Dat betreft bijvoorbeeld het weer, nieuws, toegang tot cloud-services zoals Dropbox en integratie van een agenda. Voor veel van deze diensten zijn zogenaamde software development kits (SDK's) beschikbaar. Een SDK is speciaal afgestemd op de programmeeromgeving, in dit geval .NET Core en C#. Het voordeel in vergelijking met een generieke programmeerinterface zoals REST is dat de integratie en het programmeren een stuk eenvoudiger zijn. Zo'n SDK wordt meestal geleverd door de serviceprovider of is via de community beschikbaar als bibliotheek. In Visual Studio wordt een SDK

Listing 1. Voorbeeld-weerdata van <https://openweathermap.org>.

```
{
  "coord": {
    "lon": 139,
    "lat": 35
  },
  "weather": [
    {
      "id": 800,
      "main": "Clear",
      "description": "clear sky",
      "icon": "01n"
    }
  ],
  "base": "stations",
  "main": {
    "temp": 281.52,
    "feels_like": 278.99,
    "temp_min": 280.15,
    "temp_max": 283.71,
    "pressure": 1016,
    "humidity": 93
  },
  "wind": {
    "speed": 0.47,
    "deg": 107.538
  },
  "clouds": {
    "all": 2
  },
  "dt": 1560350192,
  "sys": {
    "type": 3,
    "id": 2019346,
    "message": 0.0065,
    "country": "JP",
    "sunrise": 1560281377,
    "sunset": 1560333478
  },
  "timezone": 32400,
  "id": 1851632,
  "name": "Shuzenji",
  "cod": 200
}
```

INSTALLATIE VAN WINDOWS 10 IOT OP DE RASPBERRY PI

De installatie van het besturingssysteem op de SD-kaart van de Raspberry Pi gebeurt (evenals de ontwikkeling van de software) vanaf een Windows 10-PC. Actualiseer, indien nodig, vooraf de versie van Windows 10. Windows 10 IoT Core is momenteel beschikbaar in versie 1809 (Build 17763). Op uw PC moet minstens deze versie draaien. De SD-kaart wordt in een geschikt (intern of extern) slot op de PC geplaatst.

- **Dashboard:** als alles klaar is, downloadt u het Windows 10 IoT Dashboard setup-bestand van [4]. Hiermee wordt de volledige applicatie geladen.
- **Naar de SD-kaart schrijven:** selecteer in het dashboard (figuur A) het menu-item *Setup a new device* en selecteer daar het juiste apparaattype (Raspberry Pi 2 of 3) en het loopwerk, de SD-kaart. Het is handig als u de gegevens van uw WLAN, de gebruikersnaam en het wachtwoord voor de Raspberry Pi meteen op de SD-kaart kunt schrijven. Na acceptatie van de licentievoorwaarden start de download en schrijft het dashboard het besturingssysteem naar de SD-kaart.
- **Raspberry Pi starten:** steek de zojuist gevulde SD-kaart in de Raspberry Pi en sluit de hardware aan (monitor,

aanraakscherm, voedingseenheid), plus muis, toetsenbord en eventueel een LAN-kabel. De Raspberry Pi heeft geen aan/uit-schakelaar en start direct als de spanning wordt ingeschakeld. De eerste keer duurt het opstarten tamelijk lang. Na het selecteren van de taal verschijnt na enkele ogenblikken het startscherm van Windows 10 IoT. Er draait dan nog geen aangepaste app.

- **Raspberry Pi configureren:** dit gaat vanaf een PC via het dashboard of het Windows Device Portal (WDP) (figuur B). De Raspberry Pi model 3 heeft WiFi aan boord en Windows 10 IoT detecteert automatisch de hardware, zodat de installatie soepel verloopt. Bij model 2, zonder WiFi, moet een externe dongle worden aangesloten en ingericht, wat iets gecompliceerder is dan bij model 3. Ook het grotere rekenvermogen spreekt in het voordeel van model 3 van de Raspberry Pi.

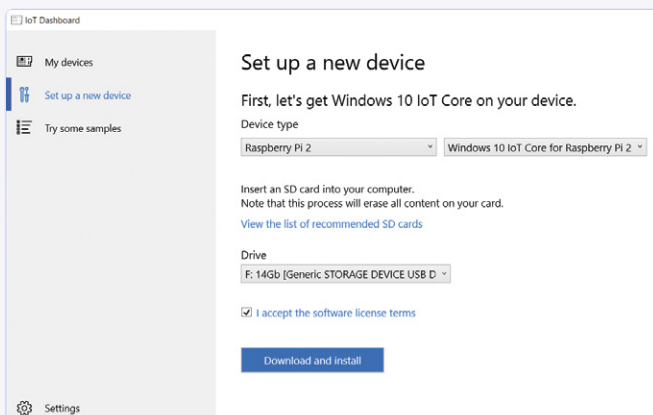
De Raspberry Pi met Windows 10 IoT kan vanuit een webbrowser worden benaderd. Het configuratie-dashboard is toegankelijk via het IP-adres van de Raspberry Pi. U kunt daar een app installeren, starten en afsluiten, statusinformatie over de systeembelasting bekijken, netwerkverbindingen beheren of het systeem afsluiten en herstarten. Als alternatief kunt u ook de Power-Shell (commandoregel) gebruiken. Op die manier kunnen meer configuratieopties worden gebruikt, maar het is niet erg comfortabel. Als er extra hardwaredrivers moeten worden ingericht, is dit echter de enige manier. We doen hier alles via de webinterface.

Gebruik en verdere ontwikkeling van de broncode

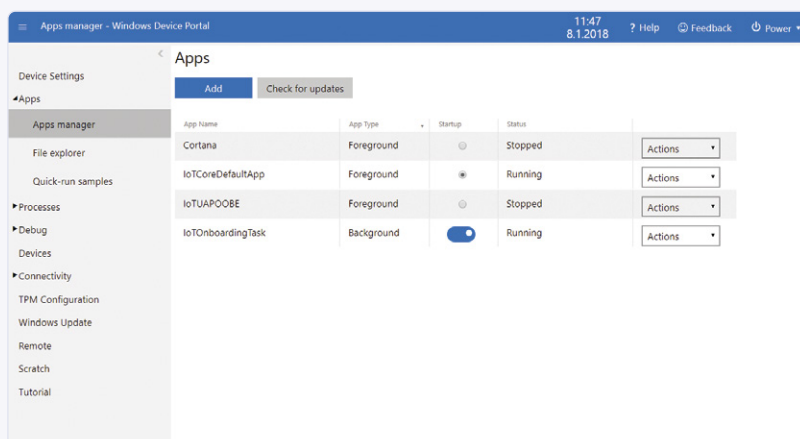
De software voor het project op [1] kan zonder problemen meteen worden gebruikt. Deze bevat: startpagina (weergave van een welkomstekst), klok, diashow met Dropbox-integratie, weergave van weergegevens en nieuws. De online diensten moeten vóór gebruik worden geconfigureerd:

- **Dropbox:** ga naar het ontwikkelgebied (console) in uw eigen Dropbox-account en maak een app aan. Geef toegang tot de app-directory en genereer app secret (key). Deze key wordt later ingevoerd in de instellingen van de app.
- **Nieuws:** registreer u met [5] en genereer gratis een persoonlijke sleutel voor toegang tot de nieuwszone. De sleutel kan niet algemeen beschikbaar worden gesteld, omdat het aantal operaties beperkt is. U slaat de sleutel direct op in het bestand *NewsControlViewModel.cs* (regel 95).
- **Weer:** genereer een key voor vrije toegang en sla deze op in het bestand *OpenWeatherMapProxyForecast.cs* (regel 13).

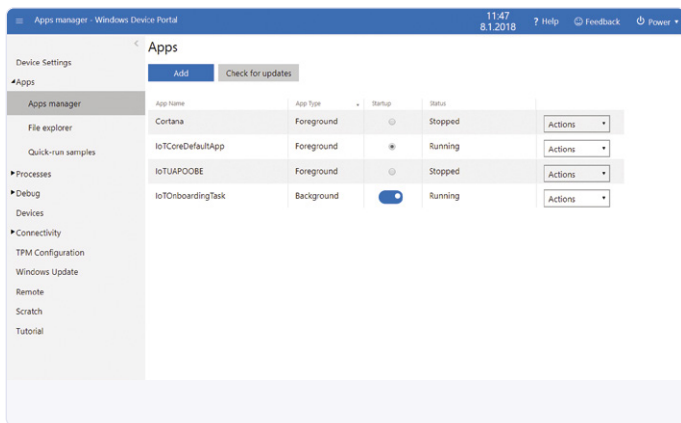
U kunt de keys voor de nieuws- en weersdiensten rechtstreeks invoeren in Visual Studio. Compileer het project daarna opnieuw en maak de app-pakketten aan voor installatie op de Raspberry Pi.



Figuur A. Dashboard van Windows 10 IoT.



Figuur B. Webinterface voor de configuratie van de Raspberry Pi.



Figuur 3. Opties van de OpenWeather-API.

geïntegreerd met behulp van de pakketbeheerder NuGet.

- **Extern (REST):** als er geen SDK is, kan elke service worden geïntegreerd als een REST-interface via de algemene webinterface. Met de gebruikelijke webmethode (GET) stuurt u een verzoek naar de service en ontvangt u een antwoord in een gestructureerd gegevensformaat (XML, JSON). Daarna kunt u de gegevens parsen, interpreteren, grafisch voorbereiden en presenteren zoals u dat wilt. Cloud-services die door applicaties van derden kunnen worden gebruikt, bieden een dergelijke interface plus bijbehorende documentatie.

Voor het gebruik van externe services, via SDK of via REST, moet u zich daar doorgaans registreren. Om de dienst te kunnen gebruiken is een unieke identificatie nodig. Meestal krijgt u daarvoor een **key** die bij elke aanvraag moet worden meegestuurd. Veel diensten

worden gratis aangeboden voor particulier en beperkt commercieel gebruik, soms is dan het aantal toegestane aanvragen per dag beperkt of zijn de gegevens van mindere kwaliteit. Toch kan een groot aantal services heel goed en uitgebreid worden gebruikt, zelfs in de gratis versie.

Services

De HomeInfoSystem-app bevat een combinatie van verschillende content-opties (services), die naar wens kunnen worden uitgebreid en gecombineerd. We beschrijven hier de koppeling met en implementatie van enkele services. Zo leert u hoe u zelf uitbreidingen kunt implementeren.

- **Integratie van de weerservice:** we maken hier gebruik van de OpenWeather-weerservice [3]. Deze dienst biedt verschillende mogelijkheden voor het ophalen van weergegevens. Er zijn betaalde diensten, maar bijvoorbeeld *Current weather* en *5 days/3 hour forecast* kunnen gratis worden gebruikt (figuur 3). Na registratie wordt een API-key gegenereerd waarmee de weergegevens kunnen worden opgevraagd. Zo'n oproep kan er bijvoorbeeld als volgt uit zien:

`api.openweathermap.org/data/2.5/weather?q={city name}&appid={your api key}`

Daarbij moet u de parameters `{city name}` en `{your api key}` zelf invullen. Het antwoord is een **record** in JSON-formaat (listing 1), dat ook 'handmatig' kan worden gelezen. Het begint met de coördinaten van de locatie, gevolgd door informatie over temperatuur, luchtdruk, bewolking enzovoort. Het record wordt in detail beschreven op de website van de dienst. Voor de app is een SDK ontwikkeld, zodat de service eenvoudig gebruikt kan worden. In listing 2 kunt u dat gedeelte van de broncode zien.

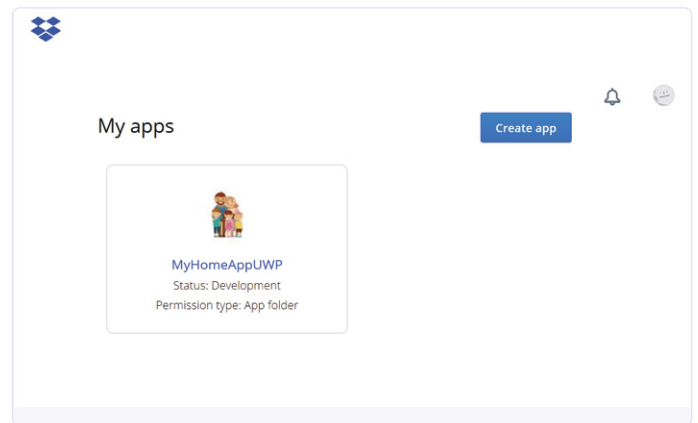
- **Integratie van cloud-opslag:** om foto's weer te geven kunt u het beste een van de bekende cloud-opslagsystemen zoals Dropbox

Listing 2. SDK (fragment) om de openweathermap.org-service te gebruiken.

// Note: The RootObject data type contains the weather data in a tree structure.

```
public async static Task<RootObject> GetWeather(string location)
{
    try
    {
        var httpClient = new HttpClient();
        string uri = basisUri + endPoint + "?q=" + location + "&appid=" + apiKey + "&units=metric";
        var response = await httpClient.GetAsync(uri);
        var result = await response.Content.ReadAsStringAsync();
        var serializer = new DataContractJsonSerializer(typeof(RootObject));
        var ms = new MemoryStream(Encoding.UTF8.GetBytes(result));
        var data = (RootObject)serializer.ReadObject(ms);
        return data;
    }
    catch
    {
        return null;
    }
}
```

of OneDrive gebruiken. Deze diensten hebben eigen API's en ook een SDK voor veel programmeertalen voor toegang via externe apps. De gratis versie van Dropbox biedt 5 GB opslagruimte voor (bijvoorbeeld) foto's. In het online-portal kunt u direct toegang krijgen tot het ontwikkelgebied, waar u een app kunt inrichten (**figuur 4**). U kunt de portal gebruiken om de app te configureren, bijvoorbeeld om aan te geven welke map binnen de Dropbox-structuur moet worden gebruikt. Er zijn verschillende manieren om van buitenaf toegang te krijgen tot de gegevens (foto's) in Dropbox: met behulp van gebruikersnaam en wachtwoord (authenticatie) of zonder gebruikersinteractie met een unieke sleutel (Key, App Secret). Deze laatste methode is voor onze toepassing optimaal, omdat de app zich met zo'n key gemakkelijk kan identificeren bij Dropbox. We gebruiken de SDK voor .NET Core, die we rechtstreeks kunnen integreren in onze



Figuur 4. U moet een app inrichten in het ontwikkel-portal van Dropbox.

Listing 3. SDK om toegang te krijgen tot gegevens in Dropbox (fragment).

```
public static async Task<ObservableCollection<string>> GetFilesFromDropBoxAsync()
{
    ObservableCollection<string> itemsList = new ObservableCollection<string>();
    // Path for storing the files
    StorageFolder storageFolder = ApplicationData.Current.LocalFolder;
    try
    {
        using (var dbx = new DropboxClient(ProgrammSettings.DropBoxAppToken))
        {
            // List all files in the App folder
            var list = await dbx.Files.ListFolderAsync(string.Empty);
            // loop through all of files
            for (int i = 0; i < list.Entries.Count; i++)
            {
                string localFileName = storageFolder.Path + "/" + list.Entries[i].Name;
                if (File.Exists(localFileName) == false)
                {
                    // Download the files
                    var file = await dbx.Files.DownloadAsync(list.Entries[i].PathLower);
                    // Extract the data stream
                    Stream stream = await file.GetContentAsStreamAsync();

                    // Write the data stream to a local file
                    using (var fileStream = File.Create(localFileName))
                    {
                        (await file.GetContentAsStreamAsync()).CopyTo(fileStream);
                    }
                }
                itemsList.Add(localFileName);
            }
        }
    }
    catch
    {
    }
    return itemsList;
}
```

app. Het ophalen van foto's uit Dropbox is dan kinderspel: dit deel van de code is te zien in **listing 3**. Gebruikers kunnen bijvoorbeeld foto's maken met hun smartphone en van daaruit direct in Dropbox opslaan.

- **Statische inhoud:** statische inhoud wordt direct in de app geïmplementeerd met behulp van programmacode. Dat betreft bijvoorbeeld het weergeven van de tijd met een speciale visuele weergave.

Dit waren enkele voorbeelden van content-implementaties voor de HomeInfoSystem-app. Voor toepassingen in bedrijven (EnterpriselnfoSystem) kunnen op dezelfde manier koppelingen worden gemaakt met voor het bedrijf interessante services.

Uitbreidingen

De hierboven beschreven services zijn slechts voorbeelden om u aan te moedigen om uw eigen HIS-project te starten. U zult zeker snel services vinden die u graag wilt implementeren. Hieronder nog enkele voorbeelden.

- Het HomeInfoSystem kan ook de waarden van verschillende sensoren weergeven. Die worden dan doorgestuurd naar een backend van het IoT en worden regelmatig door de app ingelezen. Elektronici kunnen allerlei hardware-uitbreidingen bouwen en deze flexibel in het systeem integreren.
- **Inhoud weergeven op bepaalde tijden:** sommige content kan op bepaalde, vaste tijdstippen interessant zijn. Het is bijvoorbeeld nuttig om op werkdagen tussen 7 en 8 uur 's ochtends actuele informatie over het openbaar vervoer te tonen. Maar in het weekend bent u misschien meer geïnteresseerd in informatie over welke films er draaien in de lokale bioscoop. Die info is te vinden met behulp van een bioscoopprogramma-API.
- **Mobiele services:** u kunt een mobiele app toevoegen aan de systeemomgeving. Gebruikers kunnen gegevens naar een server (backend) sturen en de app haalt deze gegevens op. Daarvoor zijn verschillende toepassingen denkbaar. Een andere interessante toepassing is het gebruik van geoservices. Mobiele gebruikers kunnen bijvoorbeeld (na toestemming te hebben gegeven) hun huidige locatie doorgeven aan een backend. De app haalt deze positie regelmatig op en toont deze op een kaart. Zo weet u waar uw gezinsleden zich op dat moment bevinden.
- **Online services:** andere nuttige services die met een publieke API kunnen worden geïntegreerd zijn diensten zoals Microsoft of Google Calendar, en dienstregelingen van het openbaar vervoer.
- **Alternatieve interfacestructuur:** in plaats van handmatige of tijdgestuurde omschakeling van de inhoud, kunt u ook een totaaloverzicht maken. Dat kan vooral interessant zijn voor grotere schermen. Zo kunt u bijvoorbeeld het actuele weer aan

de ene kant en het nieuws en de actuele foto's aan de andere kant laten zien.

U ziet dat er genoeg mogelijkheden zijn om het HomeInfoSystem uit te breiden. Een bijzondere attractie is het aansluiten van externe sensoren (IoT) om bijvoorbeeld een snelle blik op uw vakantiehuis te werpen (met een webcam). Ook op zakelijk gebied zijn er gegarandeerd veel toepassingsmogelijkheden.

Conclusie en vooruitzichten

Een informatiesysteem zoals hier beschreven, voor thuis of voor een bedrijf, is iets fascinerends. Thuis kan het actuele informatie verstrekken (weer, nieuws, data) en in het bedrijf kan het belangrijke bedrijfsgegevens presenteren (voorraden, data van teamvergaderingen). Het mooie is dat het met beperkte inspanning kan worden aangepast aan uw eigen behoeften. Na de keuze van de hardwareconfiguratie, met name het scherm, biedt de hier beschreven software al een basisfunctionaliteit, die vervolgens flexibel kan worden uitgebreid en aangepast. De broncode is open source. U kunt de code [1] in Visual Studio 2019 openen en naar eigen behoefte verder ontwikkelen en aanpassen. ◀

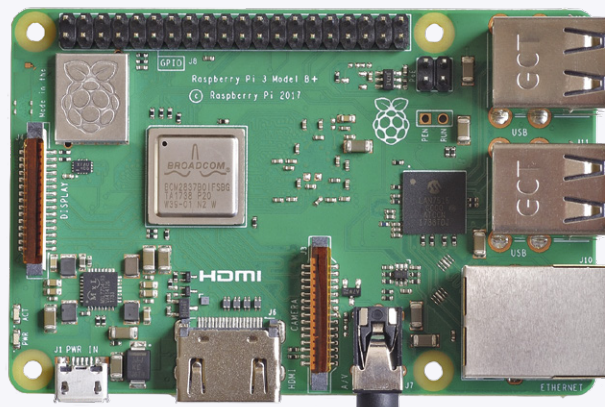
191247-04



IN DE STORE

➤ Raspberry Pi 3 B+

www.elektor.nl/raspberry-pi-3-model-b-plus



➤ Elektor Raspberry Pi Electronics kit

www.elektor.nl/elektor-raspberry-pi-electronics-kit

WEBLINKS

- [1] **Projectpagina bij dit artikel:** <http://www.elektormagazine.nl/191247-04>
- [2] **VisualStudio:** <https://visualstudio.microsoft.com/downloads/>
- [3] **OpenWeather:** <https://openweathermap.org/>
- [4] **Windows 10 IoT-Core:** <https://docs.microsoft.com/en-us/windows/iot-core/downloads>
- [5] **News-API:** <https://newsapi.org/>