# Farmbeats Labs GrovePI Gateway

## Overview

The Farmbeats labs GrovePI Gateway application is a Microsoft .Net Core[[1]](#endnote-1) Universal Windows Platform[[2]](#endnote-2)(UWP) Windows 10 IoT Core[[3]](#endnote-3) Background application[[4]](#endnote-4). This application uploads data from SeeedStudio Grove sensors[[5]](#endnote-5) attached to a Dexter Industries GrovePI[[6]](#endnote-6) and images from an optional Web Camera[[7]](#endnote-7).

The Grove sensor data is uploaded to a Microsoft Azure IoT Central[[8]](#endnote-8) hosted Azure IoT Hub[[9]](#endnote-9) for analysis and display. The web camera images are uploaded to Microsoft Azure Blob storage[[10]](#endnote-10) for post processing and analysis.

## Development Tools

The application can be deployed to a Raspberry PI[[11]](#endnote-11) device running Windows 10 IoT Core by downloading the latest installer from (https://github.com/?) or, by downloading the source code from (https://github.com/?) then building and deploying it using Microsoft Visual Studio 2017[[12]](#endnote-12).

To build and deploy the application Visual Studio 2017 needs the Windows 10 IoT Core Project Templates for VS 2017[[13]](#endnote-13) and associated dependencies installed. The application was built using the Visual C# template, there are also templates for C/C++, JavaScript and Arduino Wiring projects.

The application uses the Azure IoT Hub device client library[[14]](#endnote-14) (available via Nuget[[15]](#endnote-15)) to keep the device settings and properties up to date, and upload telemetry data to the Azure IoT Hub.

The application reads values from the sensors attached to the GrovePi using the GrovePi NuGet package[[16]](#endnote-16). Dexter industries have created sample applications which show how other SeeedStudio Grove sensors and actuators should be used[[17]](#endnote-17).

The application uses the Newtonsoft Json.Net[[18]](#endnote-18) framework (available via NuGet) for serialising message payloads.

## Application Structure

The application consists of a core file StartupTask.cs which contains the application logic, plus services for device provisioning, retrieving the device Mac Address[[19]](#endnote-19), and uploading images.

### Provisioning

On start-up the application calls the FarmBeats device provisioning service every 30 seconds until it successfully retrieves an Azure IoT Hub connection string.

### Device Twin[[20]](#endnote-20)

Once the device is connected to an Azure IoT Hub, device settings are synchronised with the configuration stored in Azure IoT Central. The Device Twin functionality is only available for MQ Queue Telemetry Transport[[21]](#endnote-21)(MQTT) protocol connected devices (Sept 2018) so TransportType.Mqtt is required when creating the Azure IoT Hub connection.

If any of the device twin settings are changed the application will need to be restarted for them to take effect.

### Device Properties

The device supports the following properties which are uploaded to the Azure IoT Hub every time the application starts. These used in the provisioning process and to help with debugging.

* Timezone – configured via Device Portal so both UTC[[22]](#endnote-22)/local times are consistent
* OSVersion – Version of the Windows 10 IoT Core operating system installed on the device
* MachineName – Network name of the device
* ApplicationDisplayName –User friendly name of the application
* ApplicationName – The publisher name of the application
* ApplicationVersion – Version of the application
* SystemID – Publisher specific unique device identifier used for debugging/provisioning.

### Application Commands

The application supports three commands, which can be initiated from Azure IoT Central. The execution of each command is also tracked with an event.

#### Sensor Update

Initiates an out of sequence sensor data update. (Useful for checking the installation/configuration of sensors)

#### Image Update

Takes an out of sequence image and uploads it to Azure storage. (Useful for aligning the camera)

#### Restart Device

The application forces the device to restart after stopping the Sensor and Camera update timers (this also forces device twin settings to be updated and updates to be applied)

### Application Events

The application notifies the Azure IoT hub of four different events

#### Sensor Update Manual

An out of sequence telemetry data update has just completed

#### Image Update Manual

An out of sequence image has just been uploaded

#### Device Restart Manual

A device restart has just been initiated

#### Application Started

The application has just started. This could have been initiated by a user or the device restarted automatically after applying updates.

### Grove PI Configuration

The GrovePI has two device twin settings

* GrovePIInstalled – A Boolean flag indicating whether the GrovePI should be configured.
* SensorUpdatePeriod – The period for sensor data updates in seconds.

If the SensorUpdatePeriod is 0 the GrovePI is configured but the update timer is not started. The GrovePI firmware version is logged to help with debugging.

### Camera Configuration

The Camera has two device twin settings

* CameraInstalled – A Boolean flag indicating whether a Camera should be configured.
* ImageUpdatePeriod - The period for camera image updates in seconds.

If the ImageUpdatePeriod is 0 the Camera is configured but the update timer is not started.

### Sensor Value Upload

The telemetry data point Centigrade, Fahrenheit & Humidity values are nullable because the GrovePI Sensirion[[23]](#endnote-23) DHT22 driver randomly returns Not a Number(NaN[[24]](#endnote-24)).

The soil moisture, and light level analog inputs also every so often returns zero[[25]](#endnote-25) but as zero is a valid value it was difficult to identify incorrect values.

public sealed class TelemetryDataPoint

{

[JsonProperty("Centigrade", NullValueHandling = NullValueHandling.Ignore)]

public double? Centigrade {get; set;}

[JsonProperty("Fahrenheit", NullValueHandling = NullValueHandling.Ignore)]

public double? Fahrenheit {get; set;}

[JsonProperty("Humidity", NullValueHandling = NullValueHandling.Ignore)]

public double? Humidity {get; set;}

public double Light {get; set;}

public double SoilMoisture1 {get; set;}

public double SoilMoisture2 {get; set;}

}

### Image Upload

The application uploads images to an HTTPS endpoint as a stream of bytes. When built in debugging mode the application also stores a copy of the most recent image (timelapse.jpg) in “User Folders\Pictures” on the device.

### Application Status

The application reports two status values, one for the camera the other for the GrovePI. These can be used to track the progress of the application as it starts.

public enum CameraStatus

{

Undefined = 0,

NotInstalled,

Installed,

Enabled,

Automatic

}

public enum GrovePiStatus

{

Undefined = 0,

NotInstalled,

Installed,

Enabled,

Automatic

}

In Azure IoT Central these enumeration vales are mapped to user friendly text labels and colours.

## Extension Projects

### Soil Moisture Sensor Evaluation

The Grove Soil Moisture sensors are designed for short-term indoor use and will degrade over time. Trialling other sensors and technologies for longer-term and/or outdoor use could be a standalone project

This capacitive soil moisture sensor is plug compatible with the existing analog input

<https://makerfabs.com/index.php?route=product/product&product_id=389>

This I2C Soil Moisture Sensor is available in indoor and outdoor versions and would free up an analog port for another sensor.

https://www.tindie.com/products/miceuz/i2c-soil-moisture-sensor/

This SoilWatch 10 -Soil Sensor is an analog sensor designed for outdoors use (3V version)

<https://www.tindie.com/products/pinotech/soilwatch-10-soil-moisture-sensor/>

This Soil Temperature and Humidity Sensor is designed for long-term outdoors use and uses the same GrovePI driver as the temperature and humidity sensor in the kit.

<https://makerfabs.com/index.php?route=product/product&product_id=384>

This I2C Sunlight Sensor could be used to monitor light levels and would free up a port for another analog sensor.

<https://www.seeedstudio.com/Grove-Digital-Light-Sensor-p-1281.html>

This I2C UV Light Sensor could be used to monitor UV levels and would free up a port for another analog sensor.

<https://www.seeedstudio.com/Grove-I2C-UV-Sensor-VEML607-p-3195.html>

For sensors with unterminated cables these could be used

<https://www.seeedstudio.com/Grove-Screw-Terminal-p-996.html>

### Automatic Watering

The software on the device could be extended to control a pump and measure the amount of water used.

385 DC Diaphragm Pump

<https://makerfabs.com/index.php?route=product/product&product_id=95>

Grove Relay

<https://www.seeedstudio.com/Grove-Relay-p-769.html>

1/8” Water Flow Sensor

<https://www.seeedstudio.com/G1-8-Water-Flow-Sensor-p-1346.html>

## Summary

The Farmbeats Labs Student Kit is intended to provide an introduction to Precision Agriculture with a selection of environment sensors.

This project is just the beginning and over the next couple of months we will be releasing additional functionality and working with students from all over the world to explore how the Internet of Things (IoT) can improve agriculture and the environment.

Share your ideas and projects with others at (https://github/?)

1. https://dotnetfoundation.org/ [↑](#endnote-ref-1)
2. https://docs.microsoft.com/en-us/windows/uwp/get-started/universal-application-platform-guide [↑](#endnote-ref-2)
3. https://developer.microsoft.com/en-us/windows/iot [↑](#endnote-ref-3)
4. https://docs.microsoft.com/en-us/windows/iot-core/develop-your-app/backgroundapplications [↑](#endnote-ref-4)
5. https://www.seeedstudio.com/category/Grove-c-1003.html [↑](#endnote-ref-5)
6. https://www.dexterindustries.com/grovepi/ [↑](#endnote-ref-6)
7. https://docs.microsoft.com/en-us/windows/iot-core/learn-about-hardware/hardwarecompatlist [↑](#endnote-ref-7)
8. https://azure.microsoft.com/en-us/services/iot-central/ [↑](#endnote-ref-8)
9. https://azure.microsoft.com/en-us/services/iot-hub/ [↑](#endnote-ref-9)
10. https://azure.microsoft.com/en-us/services/storage/blobs/ [↑](#endnote-ref-10)
11. https://www.raspberrypi.org/ [↑](#endnote-ref-11)
12. <https://visualstudio.microsoft.com/> Community version is sufficient [↑](#endnote-ref-12)
13. https://marketplace.visualstudio.com/items?itemName=MicrosoftIoT.WindowsIoTCoreProjectTemplatesforVS15 [↑](#endnote-ref-13)
14. https://docs.microsoft.com/en-us/dotnet/api/microsoft.azure.devices.client [↑](#endnote-ref-14)
15. https://www.nuget.org/ [↑](#endnote-ref-15)
16. https://www.nuget.org/packages/GrovePi/ [↑](#endnote-ref-16)
17. https://github.com/DexterInd/GrovePi/tree/master/Software/CSharp [↑](#endnote-ref-17)
18. https://www.newtonsoft.com/json [↑](#endnote-ref-18)
19. https://en.wikipedia.org/wiki/MAC\_address [↑](#endnote-ref-19)
20. https://docs.microsoft.com/en-us/azure/iot-hub/iot-hub-devguide-device-twins [↑](#endnote-ref-20)
21. https://mqtt.org/ [↑](#endnote-ref-21)
22. https://en.wikipedia.org/wiki/Coordinated\_Universal\_Time [↑](#endnote-ref-22)
23. https://www.sensirion.com [↑](#endnote-ref-23)
24. https://en.wikipedia.org/wiki/NaN [↑](#endnote-ref-24)
25. https://forum.dexterindustries.com/t/issue-with-analog-ports-reads/695 [↑](#endnote-ref-25)