Windows 10 IoT Core

Internet of Things

Maker Den Lab Guide

Document Version 2.1

Social	**Twitter** #makerden #iot #raspberrypi2 #windows10
Document Authors	Dave Glover | dglover@microsoft.com | @dglover Andrew Coates |acoat@microsoft.com |@coatsy Fai Lai |hoongfai@microsoft.com |@faister
Document location	https://github.com/MakerDen/IoT-Maker-Den-Documentation-and-Resources
Source Code Location	https://github.com/MakerDen/IoT-Maker-Den-Windows-for-IoT
Disclaimer	All care has been taken to ensure the accuracy of this document. No liability accepted.
Copyright	You are free to reuse and modify this document and associated software.

INTRODUCTION

Welcome to the Internet of Things Maker Den Lab where you will get firsthand experience with hardware prototyping and deploying code to a Raspberry Pi 2 running Windows 10 IoT Core.

GOAL

The goal of the Maker Den is to familiarise you with some of the components and technologies associated with the Internet of Things (IoT). Along the way, you will experience wiring circuits, deploying code, and streaming sensor data to Microsoft Azure.

GETTING STARTED

If you are setting up your own Maker Den then all source code and documentation is available at https://github.com/MakerDen/IoT-Maker-Den-Documentation-and-Resources.

TIME REQUIRED

The lab will take approximately 15 minutes to complete. You are more than welcome to stay longer and delve a little deeper.

SPREAD THE WORD

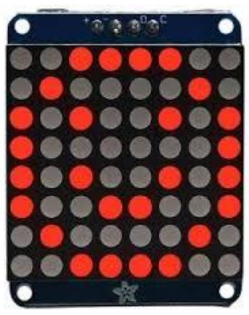
Be sure to spread the word about the Internet of Things Maker Den on Twitter. Use hash tags #makerden #iot #raspberrypi2 #windows10

SKILLS REQUIRED

Some dexterity to add a couple of sensors to a breadboard and some experience with Visual Studio will be useful but not essential.

LAB HARDWARE

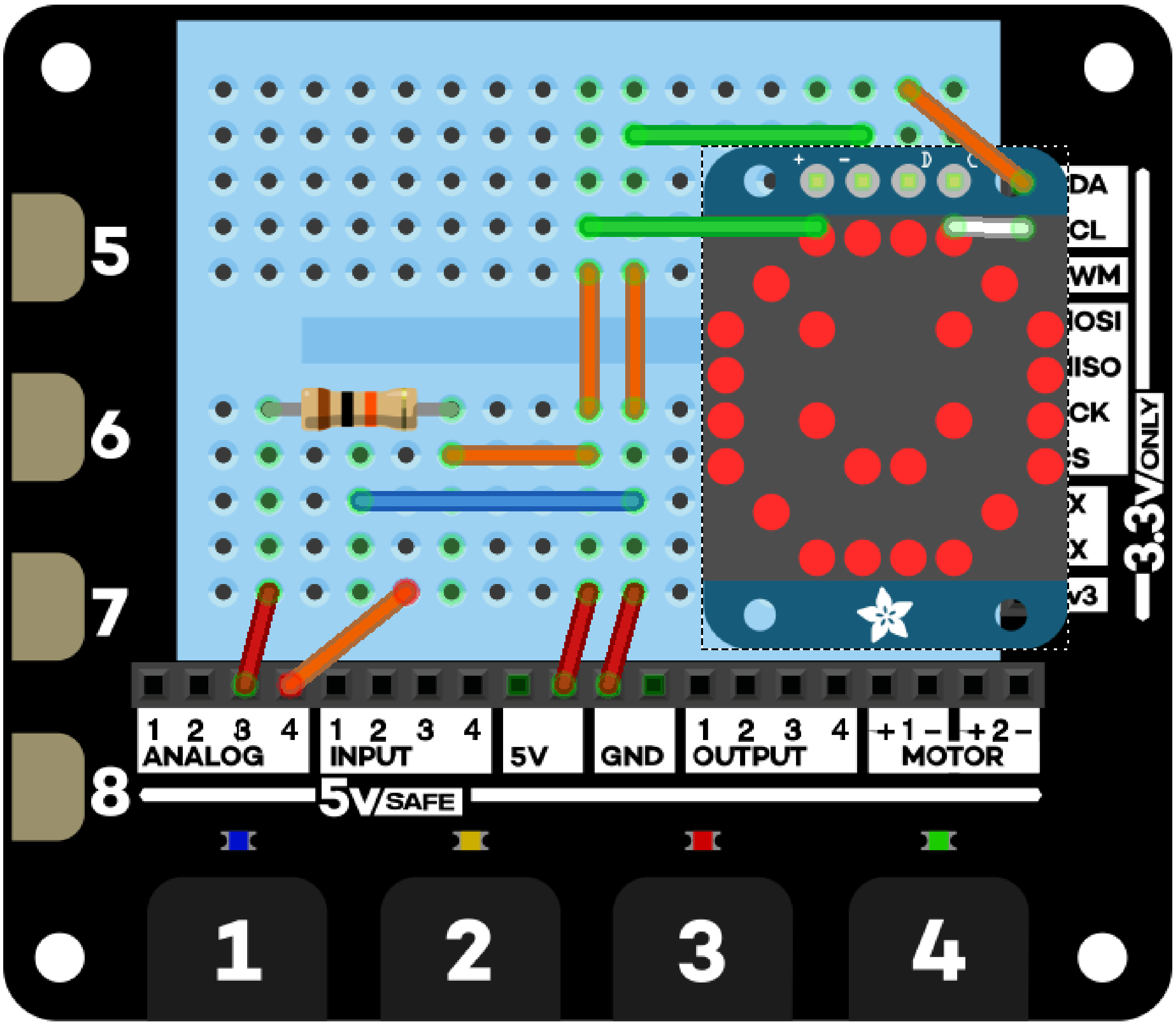
The following components are used for the Maker Den.

Raspberry Pi 2 These labs are built on the Raspberry Pi 2 running Windows 10 IoT Core. You can find out more about Windows 10 IoT Core at http://dev.windows.com/iot.	
Analogue Temperature Sensor (Microchip MCP9700A) Reads the temperature and reports it as a value in Degrees Centigrade.	
Light Dependent Resistor (LDR) aka Photoresistor A Light Dependent Resistor changes its resistance depending on light levels.	
Adafruit Mini 8x8 LED Matrix Great for displaying scrolling text and basic graphics	
Explorer HAT Pro from Pimoroni Useful prototypng HAT for RPi, has I2C ADC, I2C capacitive touch pads, motor driver, and a breadboard for prototyping.	

SETTING UP YOUR PROTOTYPE

The first task is to add two sensors to the Raspberry Pi 2 Breadboard Prototyping HAT.

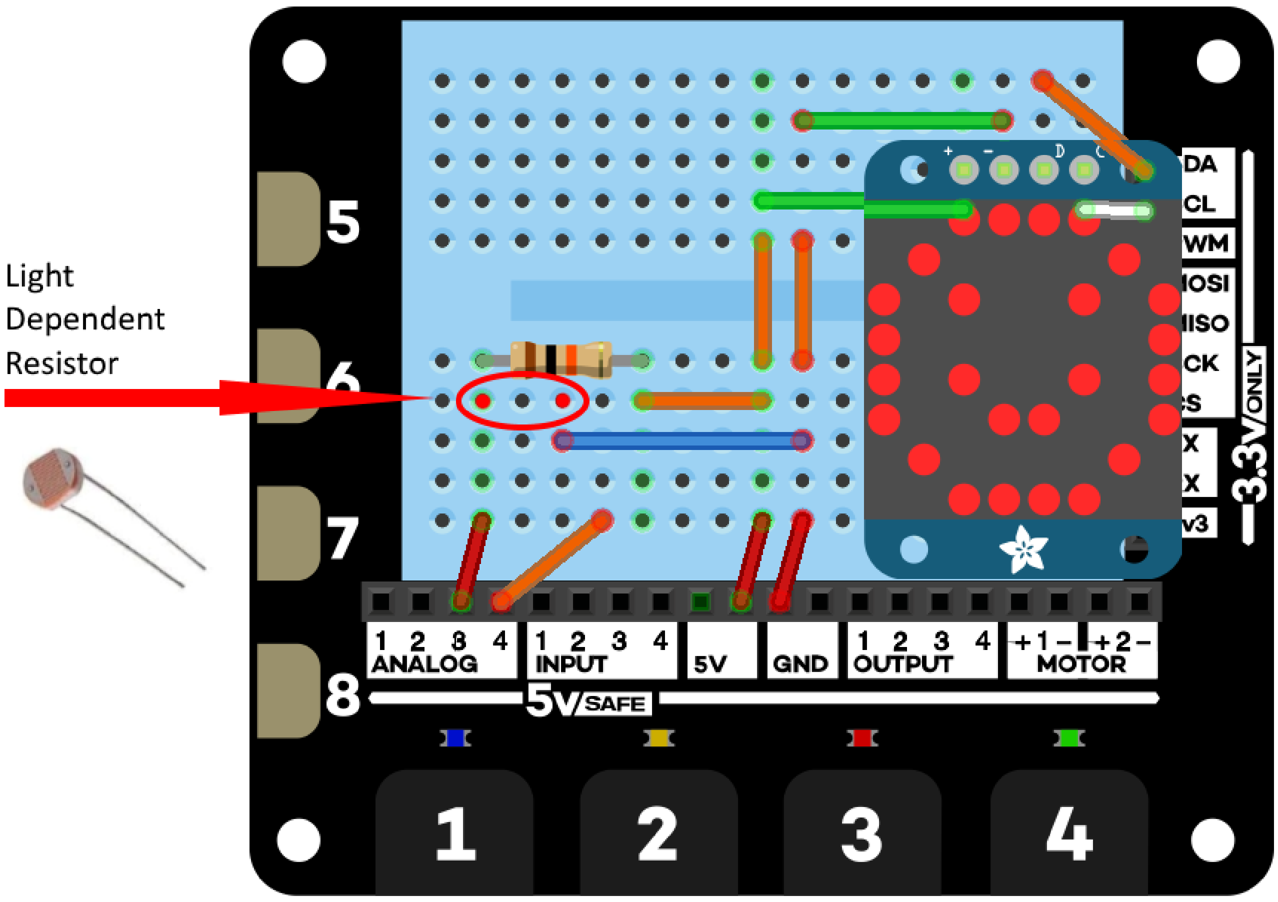
Your Breadboard Prototyping HAT should look like the image below before you add the sensors[1].

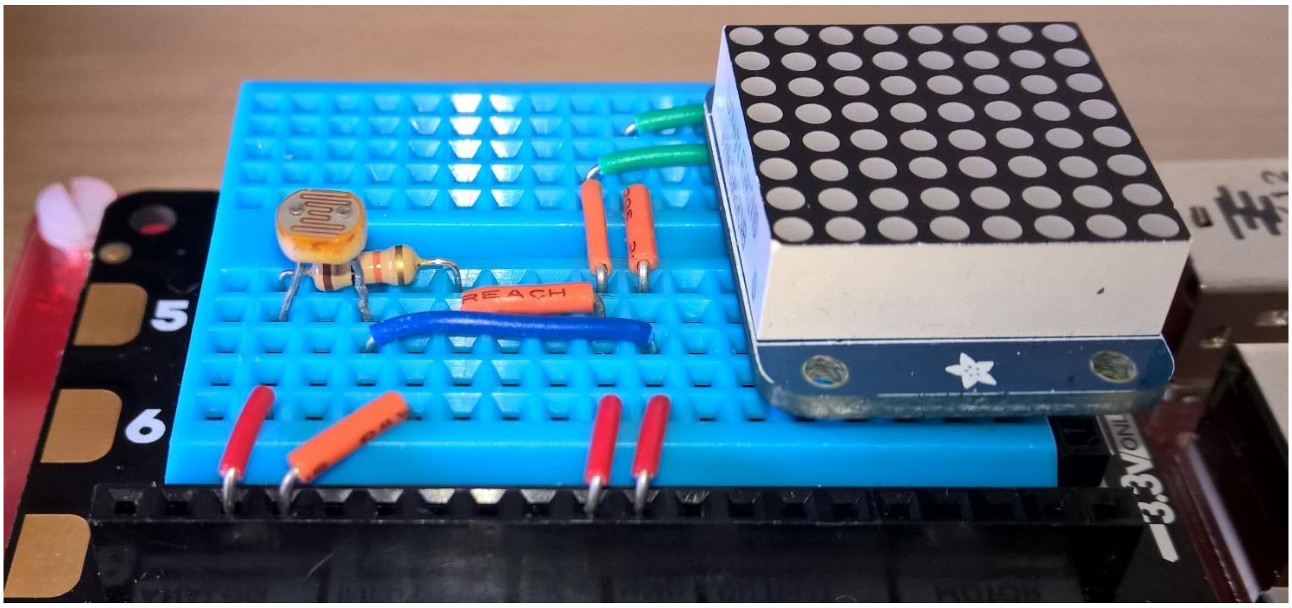


[1] Breadboards work on the principle that each of the five sockets in a column are wired in common. So, by inserting a component into a socket in the same column as another component, those components will be electrically coupled, as though their wires were joined. This makes it very simple to quickly create and modify electrical circuits without the necessity for soldering, clips or junction boxes.

ADDING THE LIGHT DEPENDENT RESISTOR (LDR)

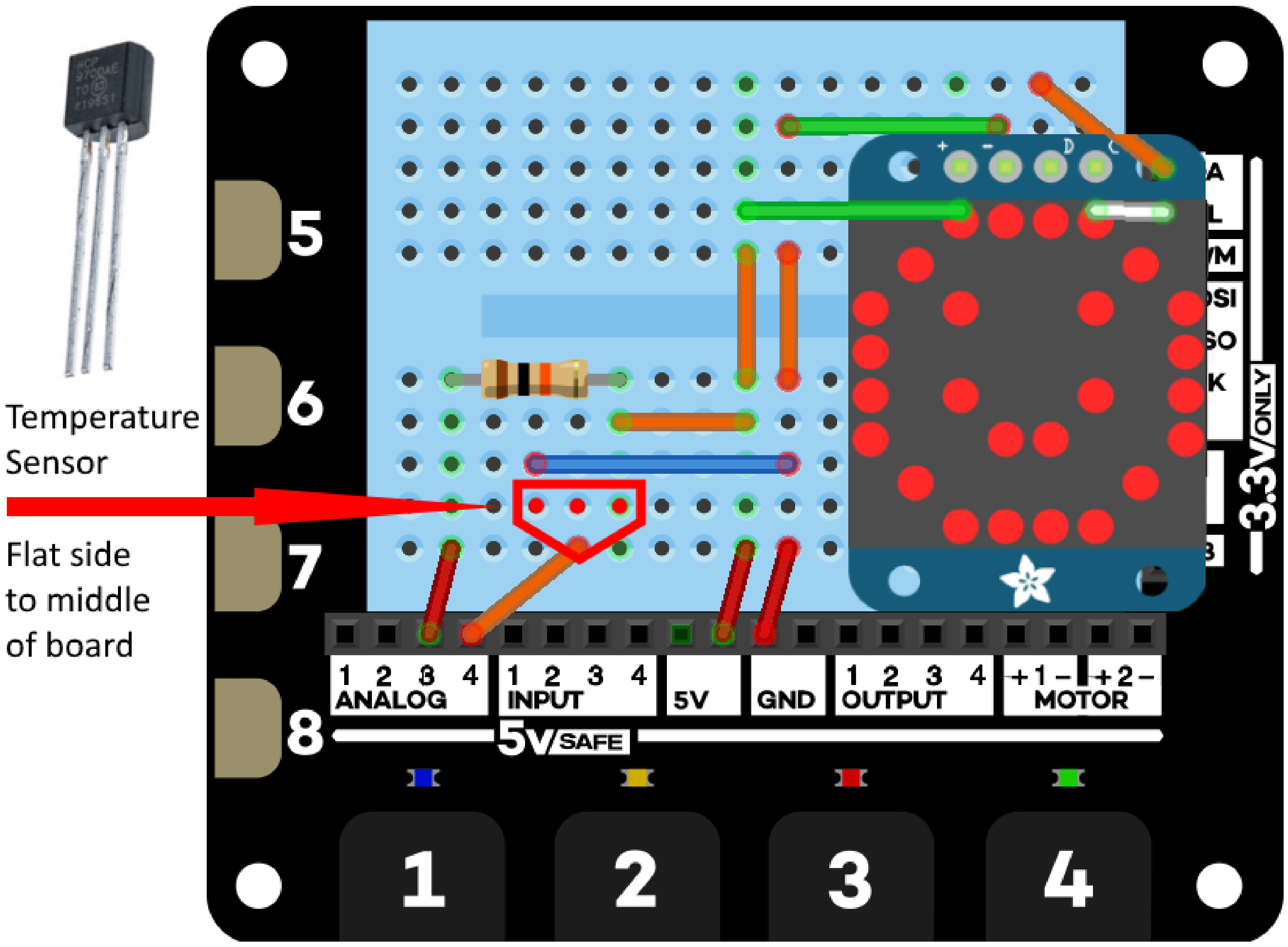
Insert the Light Dependent Resistor pins into the holes highlighted by the red dots on the breadboard image below.

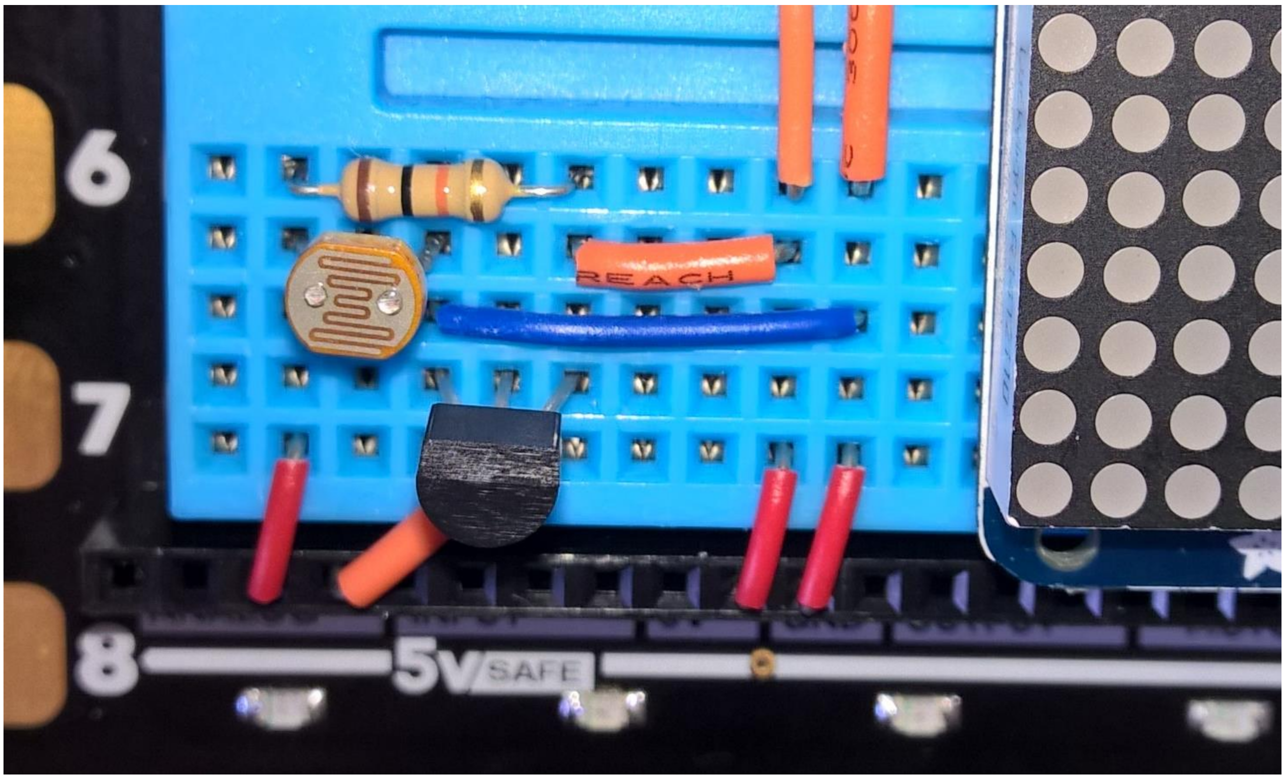




ADDING THE TEMPERATURE SENSOR

Insert the temperature sensor into the breadboard as highlighted in the image below. **The flat side of the temperature sensor must face towards the middle of the breadboard**.





EXPERIMENTS

There are five Maker Den experiments to get you started with Windows 10 IoT Core and Microsoft Azure IoT Services.

⌦ All the source code can be referenced from the Source Code folder on the Desktop.

⌦ This user guide and an architectural overview of the Maker Den can be found in the Documents folder on the Desktop.

⌦ Be sure to check out the Windows 10 IoT Core Doc, Tutorials and Samples. There is a link to this page in the Desktop Documents folder.

⌦ For the self-sufficient adventurous types, the Windows 10 IoT Core Node.Js and Python developer tools have been installed. Reference the Windows 10 IoT Core Doc, Tutorials and Samples and the Explorer HAT Pro GPIO-pins for more information.

RESETTING THE LABS

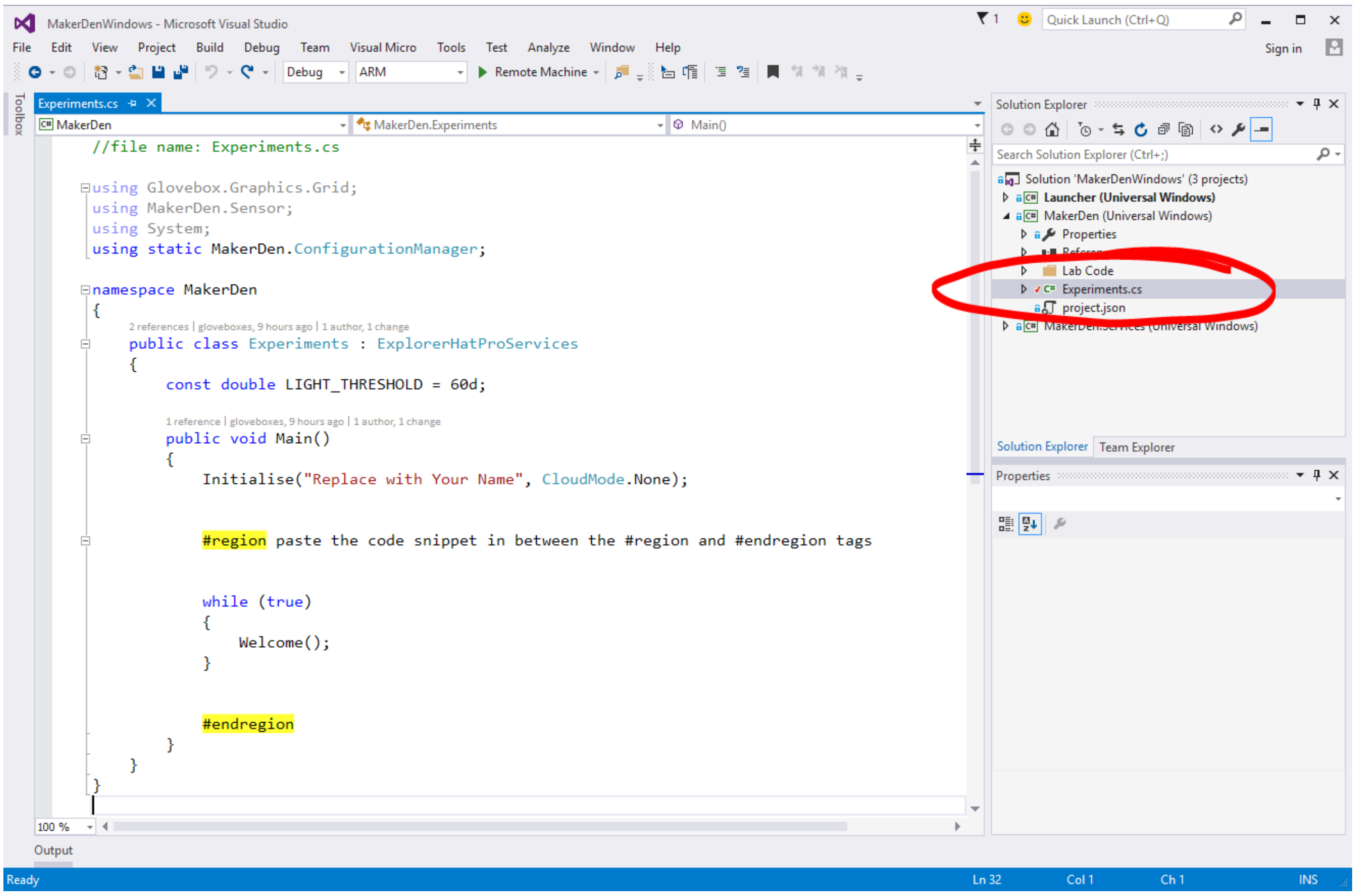
⌦ **STEP 1:** **Ensure Visual Studio is closed**.

⌦ **STEP 2:** Double click the **ResetLabs.bat** file on your desktop. This will copy the source code from a GitHub repository and launch Visual Studio with the solution opened.

EXPERIMENT 1: HELLO THERE

Deploy your first experiment to ensure everything is setup correctly and to check Visual Studio is communicating with your Raspberry Pi.

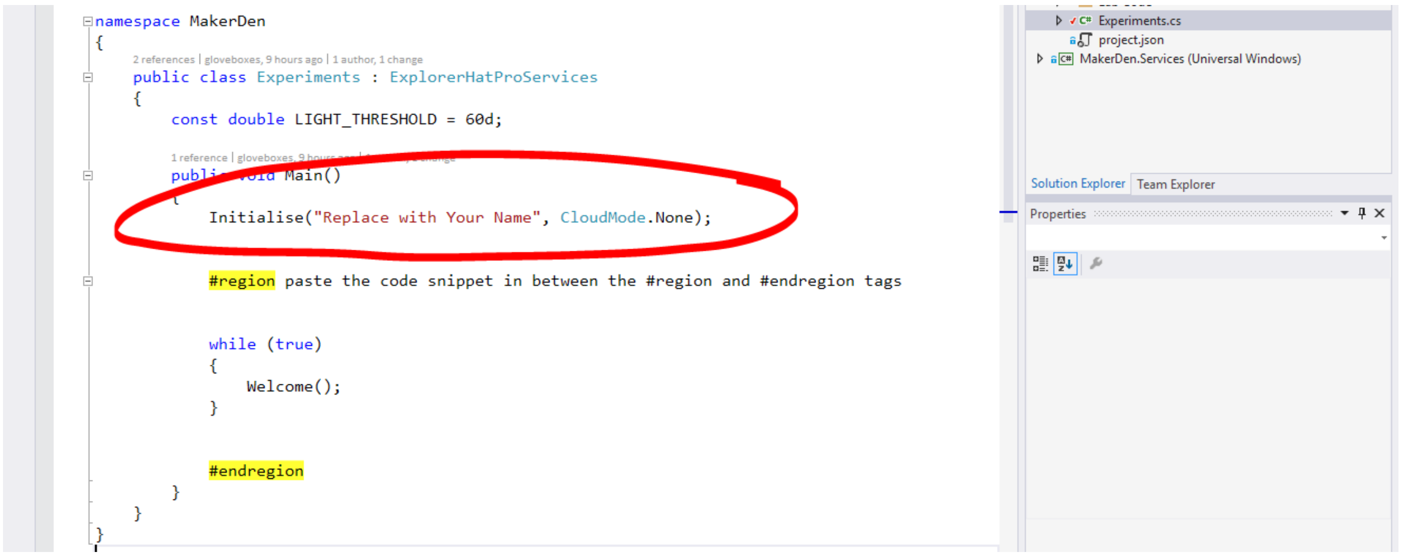
⌦**STEP 1**: Expand the **MakerDen** project then double click the **Experiments.cs** file to open it.



```
//file name: Experiments.cs

using Glovebox.Graphics.Grid;
using MakerDen.Sensor;
using System;
using static MakerDen.ConfigurationManager;

namespace MakerDen
{
    public class Experiments : ExplorerHatProServices
    {
        const double LIGHT_THRESHOLD = 60d;

        public void Main()
        {
            Initialise("Replace with Your Name", CloudMode.None);


            #region paste the code snippet in between the #region and #endregion tags


            while (true)
            {
                Welcome();
            }


            #endregion
        }
    }
}
```

MakerDenWindows - Microsoft Visual Studio
File Edit View Project Build Debug Team Visual Micro Tools Test Analyze Window Help
Debug ARM Remote Machine
Quick Launch (Ctrl+Q)
Sign in
Experiments.cs
MakerDen
MakerDen.Experiments
Main()
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'MakerDenWindows' (3 projects)
Launcher (Universal Windows)
MakerDen (Universal Windows)
Properties
References
Lab Code
Experiments.cs
project.json
MakerDen.Services (Universal Windows)
Solution Explorer Team Explorer
Properties
100 %
Output
Ready
Ln 32 Col 1 Ch 1 INS

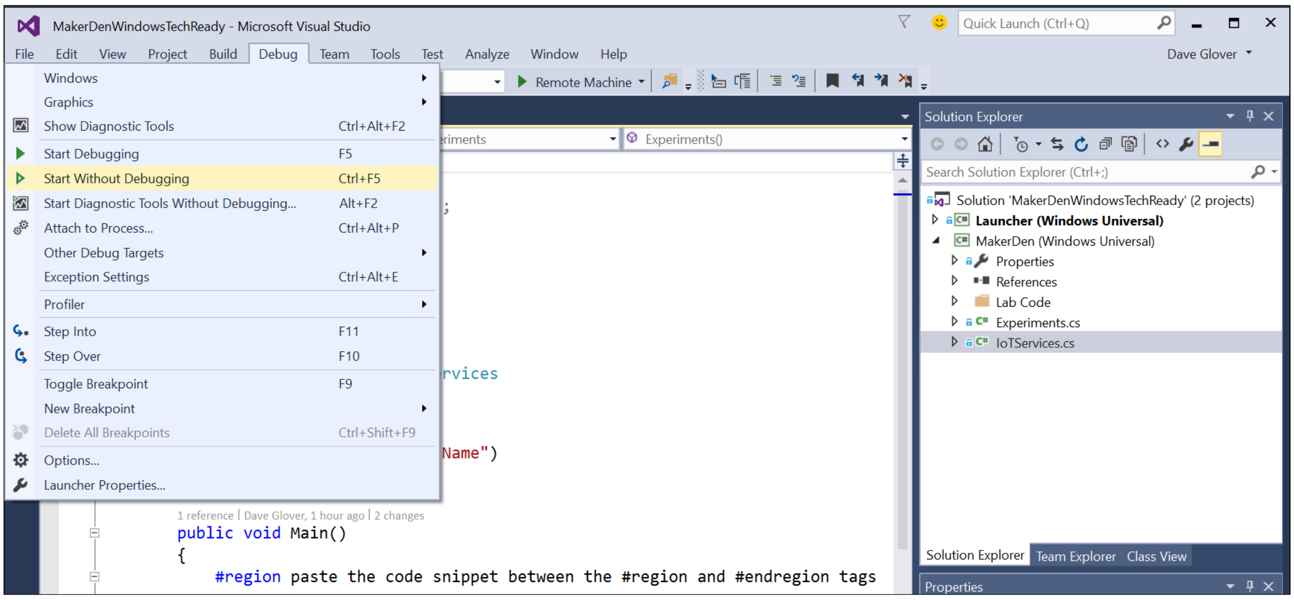
⌦**STEP 2:** Edit Experiments.cs by typing your name where it says "Replace with Your Name". Be sure to type your name inside the quotation marks.



```
namespace MakerDen
{
    public class Experiments : ExplorerHatProServices
    {
        const double LIGHT_THRESHOLD = 60d;

        public void Main()
        {
            Initialise("Replace with Your Name", CloudMode.None);

            #region paste the code snippet in between the #region and #endregion tags


            while (true)
            {
                Welcome();
            }


            #endregion
        }
    }
}
```

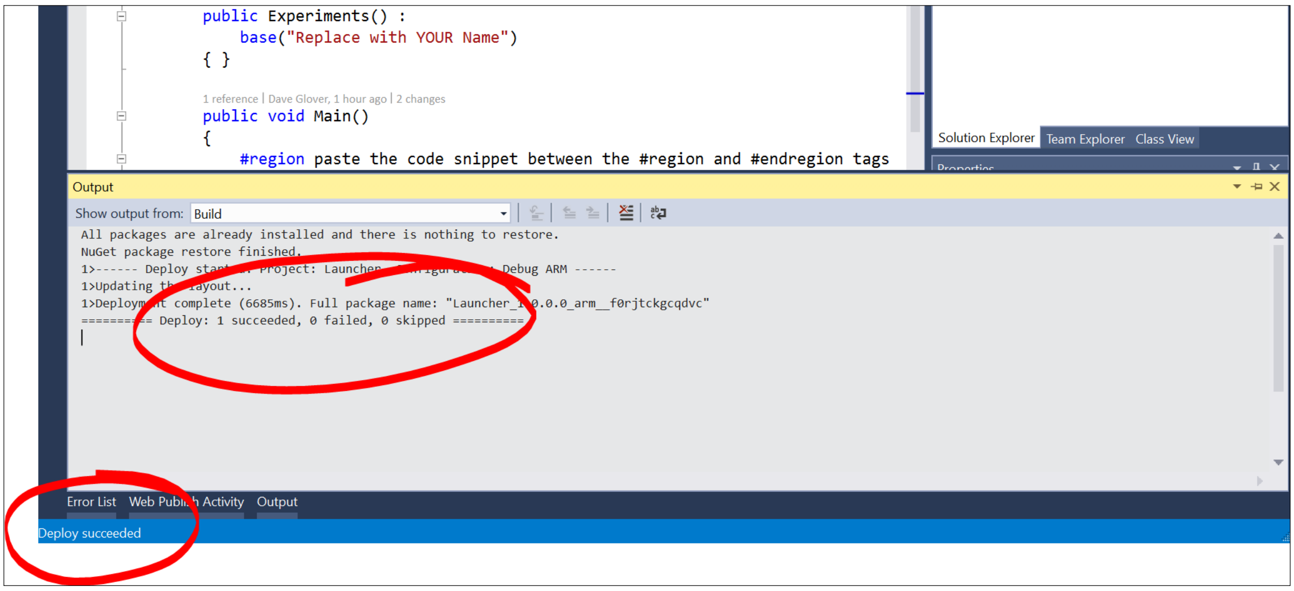
Experiments.cs
project.json
MakerDen.Services (Universal Windows)
Solution Explorer Team Explorer
Properties

☒**STEP 3:** Deploy the solution to the Raspberry Pi. From the **Debug** menu select **Start Without Debugging** or from the keyboard press **Ctrl+F5.**


MakerDenWindowsTechReady - Microsoft Visual Studio
Quick Launch (Ctrl+Q)
File Edit View Project Build Debug Team Tools Test Analyze Window Help
Dave Glover
Remote Machine
Windows
Graphics
Show Diagnostic Tools Ctrl+Alt+F2
Start Debugging F5
Start Without Debugging Ctrl+F5
Start Diagnostic Tools Without Debugging... Alt+F2
Attach to Process... Ctrl+Alt+P
Other Debug Targets
Exception Settings Ctrl+Alt+E
Profiler
Step Into F11
Step Over F10
Toggle Breakpoint F9
New Breakpoint
Delete All Breakpoints Ctrl+Shift+F9
Options...
Launcher Properties...
Experiments()
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'MakerDenWindowsTechReady' (2 projects)
Launcher (Windows Universal)
MakerDen (Windows Universal)
Properties
References
Lab Code
Experiments.cs
IoTServices.cs
Solution Explorer
Team Explorer
Class View
Properties

```
1 reference | Dave Glover, 1 hour ago | 2 changes
public void Main()
{
    #region paste the code snippet between the #region and #endregion tags
```

☒**STEP 4:** Check that Visual Studio has successfully compiled and deployed the code by looking at the output window and the status bar.



```
        public Experiments() :
            base("Replace with YOUR Name")
        { }

        1 reference | Dave Glover, 1 hour ago | 2 changes
        public void Main()
        {
            #region paste the code snippet between the #region and #endregion tags
```

```
All packages are already installed and there is nothing to restore.
NuGet package restore finished.
1>------ Deploy started: Project: Launcher, Configuration: Debug ARM ------
1>Updating the layout...
1>Deployment complete (6685ms). Full package name: "Launcher_1.0.0.0_arm__f0rjtckgcqdvc"
========== Deploy: 1 succeeded, 0 failed, 0 skipped ==========
```

☒**STEP 5:** Check the LED Matrix on the Raspberry Pi. You should see your name, the machine name and the IP Address scrolling on the LED Matrix display.

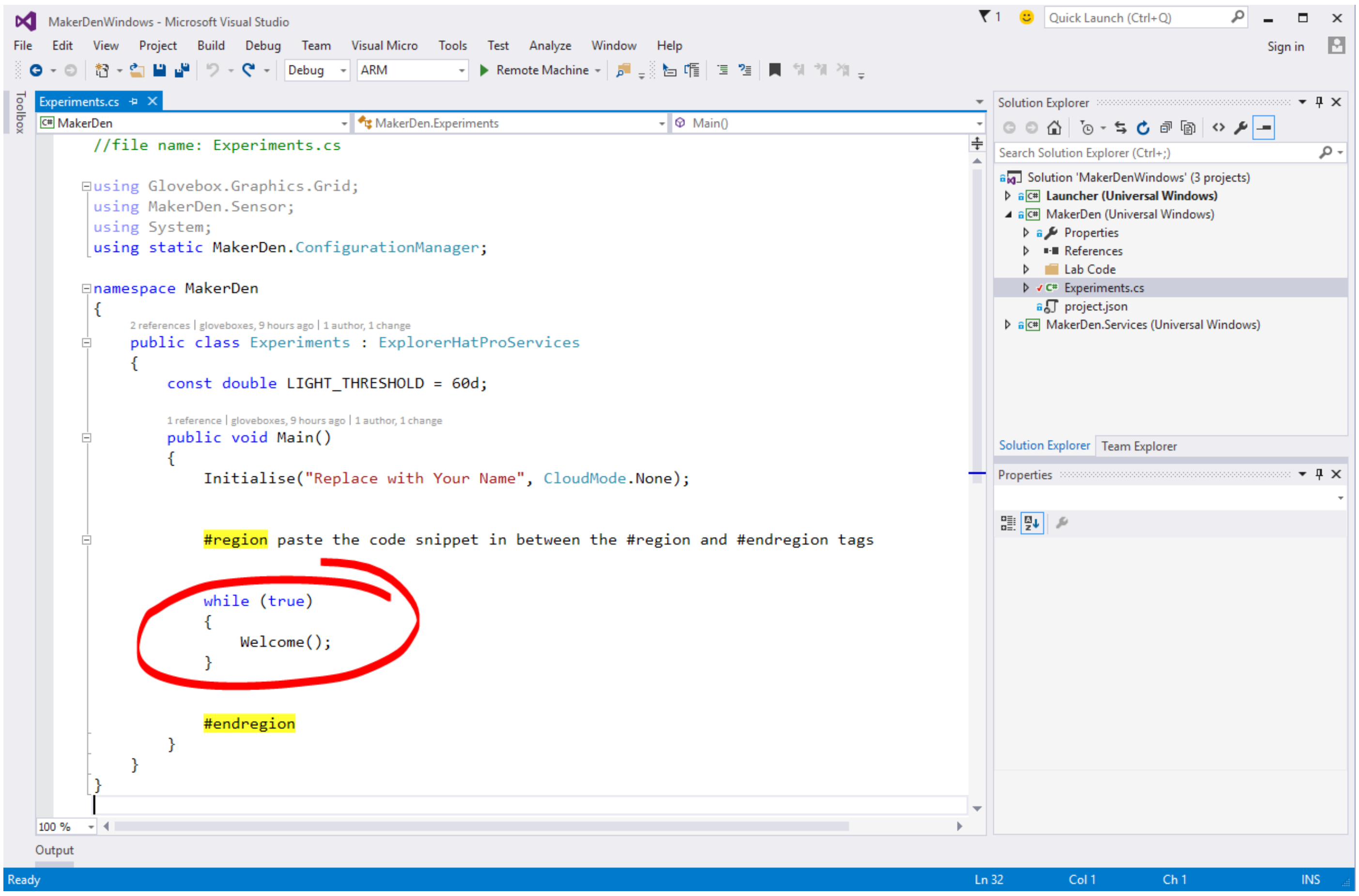
☒**STEP 6:** Pat yourself on the back, you did it☺

EXPERIMENT 2: SENSING THE WORLD

This lab reads the current light levels from the light sensor.

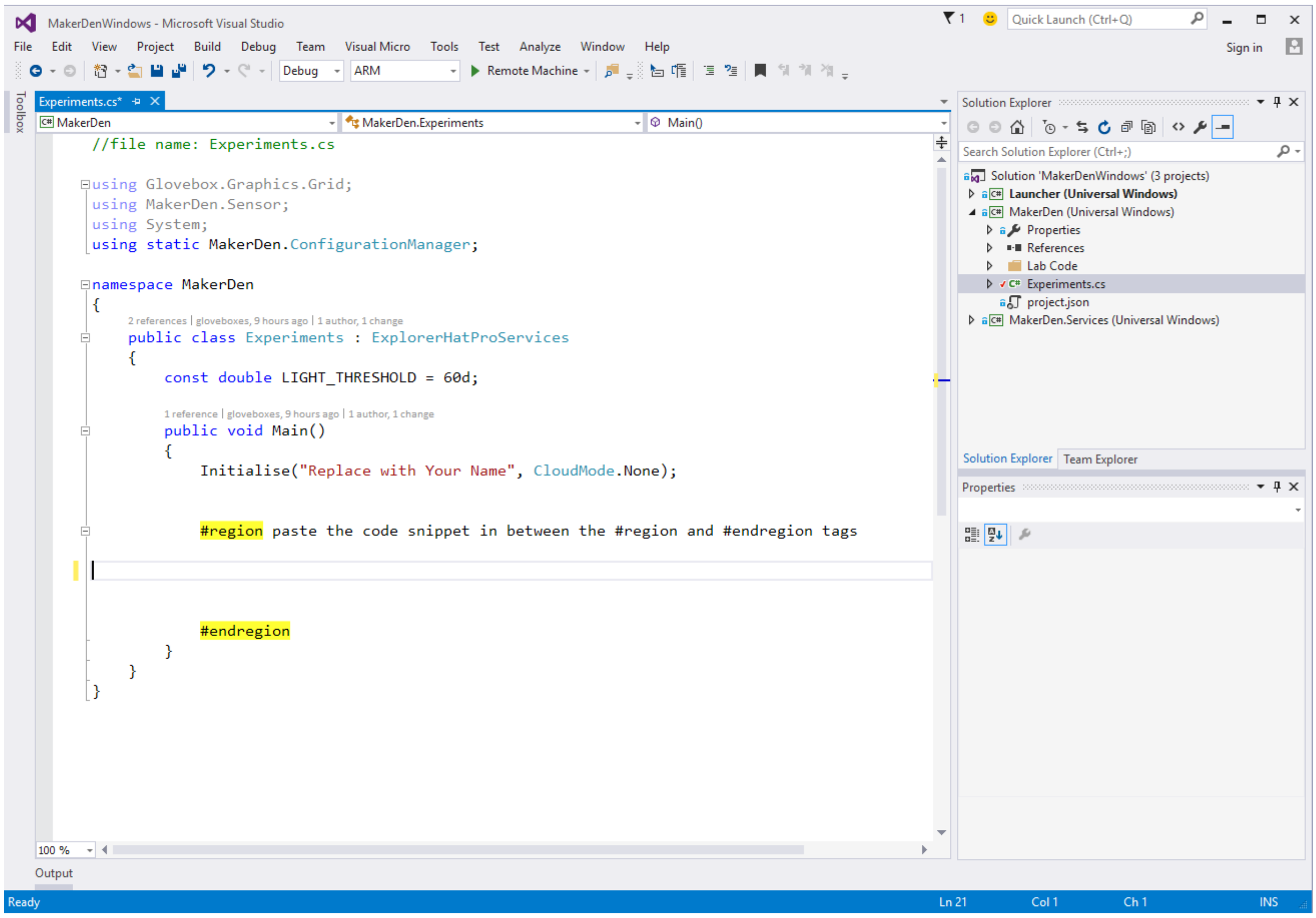
⌦**STEP 1:** Review the code in the **Experiments.cs** file. Look for the **#region** and **#endregion** tags.

⌦**STEP 2:** Delete the code circled in red **inside** the #region tags.


MakerDenWindows - Microsoft Visual Studio
File Edit View Project Build Debug Team Visual Micro Tools Test Analyze Window Help
Debug ARM Remote Machine
Quick Launch (Ctrl+Q)
Sign in
Experiments.cs
MakerDen
MakerDen.Experiments
Main()
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'MakerDenWindows' (3 projects)
Launcher (Universal Windows)
MakerDen (Universal Windows)
Properties
References
Lab Code
Experiments.cs
project.json
MakerDen.Services (Universal Windows)
Solution Explorer Team Explorer
Properties
100 %
Output
Ready
Ln 32 Col 1 Ch 1 INS

```
//file name: Experiments.cs

using Glovebox.Graphics.Grid;
using MakerDen.Sensor;
using System;
using static MakerDen.ConfigurationManager;

namespace MakerDen
{
    2 references | gloveboxes, 9 hours ago | 1 author, 1 change
    public class Experiments : ExplorerHatProServices
    {
        const double LIGHT_THRESHOLD = 60d;

        1 reference | gloveboxes, 9 hours ago | 1 author, 1 change
        public void Main()
        {
            Initialise("Replace with Your Name", CloudMode.None);


            #region paste the code snippet in between the #region and #endregion tags

            while (true)
            {
                Welcome();
            }


            #endregion
        }
    }
}
```

Your “Experiments.cs” file should look like the screenshot below after you have deleted the code. If it doesn’t look the same then **Ctrl+Z** to undo the changes you made and try again.


MakerDenWindows - Microsoft Visual Studio
File Edit View Project Build Debug Team Visual Micro Tools Test Analyze Window Help
Quick Launch (Ctrl+Q)
Sign in
Debug
ARM
Remote Machine
Experiments.cs*
MakerDen
MakerDen.Experiments
Main()
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'MakerDenWindows' (3 projects)
Launcher (Universal Windows)
MakerDen (Universal Windows)
Properties
References
Lab Code
Experiments.cs
project.json
MakerDen.Services (Universal Windows)
Solution Explorer
Team Explorer
Properties
Toolbox
100 %
Output
Ready
Ln 21
Col 1
Ch 1
INS

```
//file name: Experiments.cs

using Glovebox.Graphics.Grid;
using MakerDen.Sensor;
using System;
using static MakerDen.ConfigurationManager;

namespace MakerDen
{
    2 references | gloveboxes, 9 hours ago | 1 author, 1 change
    public class Experiments : ExplorerHatProServices
    {
        const double LIGHT_THRESHOLD = 60d;

        1 reference | gloveboxes, 9 hours ago | 1 author, 1 change
        public void Main()
        {
            Initialise("Replace with Your Name", CloudMode.None);


            #region paste the code snippet in between the #region and #endregion tags




            #endregion
        }
    }
}
```

☒ **STEP 3:** Type the following code between the #region tags **OR** using a code snippet type **lab2** and press Tab twice.

```
using (SensorMgr lightSensorMgr = new SensorMgr(light))
{
    // keep looping until the plug is pulled - this is a very common IoT pattern
    while (true)
    {
        // this is simply so we can break and see the value
        var level = light.ReadRatio * 100;

        // if the light level is above a certain level
        if (level > LIGHT_THRESHOLD)
        {
            Display(Grid8x8.Symbols.HappyFace);  // Happy Face
            ledGreen.On(); // turn on the green LED on the Explorer HAT
        }
        else
        {
            Display(Grid8x8.Symbols.SadFace);  // Sad face
            ledGreen.Off(); // turn off the green LED on the Explorer HAT
        }
        Util.Delay(500);
    }
}
```

⌧**STEP 4:** Your “Experiments.cs” file should like look like the following. If not, **Ctrl+Z** and try again.

```
//file name: Experiments.cs

using Glovebox.Graphics.Grid;
using MakerDen.Sensor;
using System;
using static MakerDen.ConfigurationManager;

namespace MakerDen
{
    public class Experiments : ExplorerHatProServices
    {
        const double LIGHT_THRESHOLD = 60d;

        public void Main()
        {
            Initialise("Replace with Your Name", CloudMode.None);

            #region paste the code snippet in between the #region and #endregion tags

            using (SensorMgr lightSensorMgr = new SensorMgr(light))
            {

                // keep looping until the plug is pulled - this is a very common IoT pattern
                while (true)
                {
                    // this is simply so we can break and see the value
                    var level = light.ReadRatio * 100;

                    // if the light level is above a certain level
                    if (level > LIGHT_THRESHOLD)
                    {
                        Display(Grid8x8.Symbols.HappyFace);  // Happy Face
                        ledGreen.On(); // turn on the green LED on the Explorer HAT
                    }
                    else
                    {
                        Display(Grid8x8.Symbols.SadFace);  // Sad face
                        ledGreen.Off(); // turn off the green LED on the Explorer HAT
                    }
                    Util.Delay(500);
                }
            }

            #endregion
        }
    }
}
```

⌧**STEP 5:** Deploy the solution to the Raspberry Pi. From the **Debug** menu select **Start Without Debugging** or from the keyboard press **Ctrl+F5.**

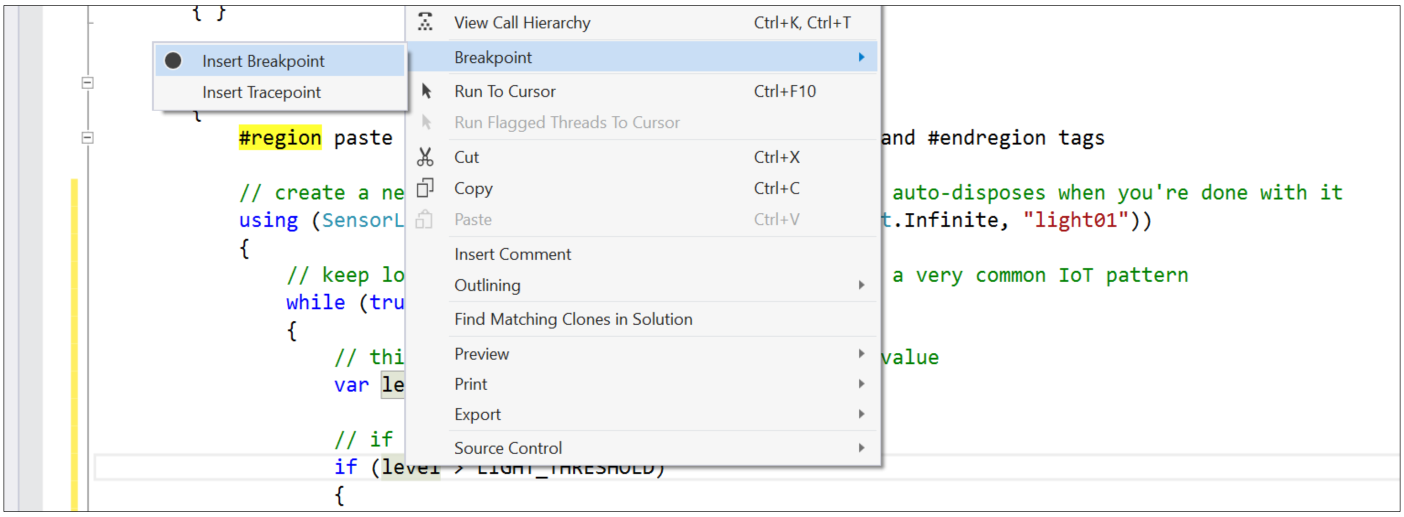
⌧**STEP 6:** Hover your hand over the light sensor and observe that the face changes and the green LED turns off when it gets dark.

EXPERIMENT 2A (OPTIONAL): REMOTE DEBUGGING

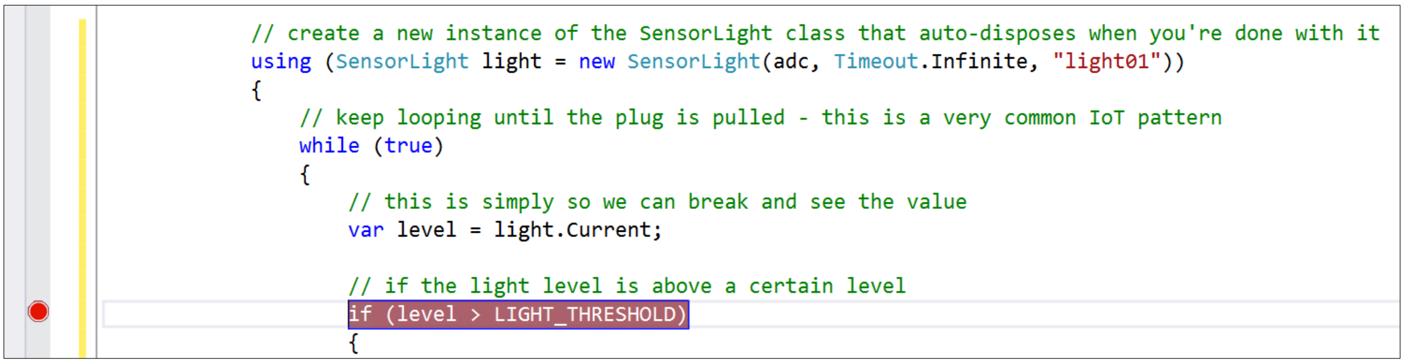
⌧ **STEP 1:** Next, set a break point to see how easy it is to debug directly on the device. This is a unique capability provided by Visual Studio.

Right-click on the line that reads `if (level > LIGHT_THRESHOLD)`

Choose Breakpoint, then Insert Breakpoint.



```
        { }
    {
        #region paste                              and #endregion tags

        // create a ne                              auto-disposes when you're done with it
        using (SensorL                              t.Infinite, "light01"))
        {
            // keep lo                              a very common IoT pattern
            while (tru
            {
                // thi                              value
                var le

                // if
                if (level > LIGHT_THRESHOLD)
                {
```



```
// create a new instance of the SensorLight class that auto-disposes when you're done with it
using (SensorLight light = new SensorLight(adc, Timeout.Infinite, "light01"))
{
    // keep looping until the plug is pulled - this is a very common IoT pattern
    while (true)
    {
        // this is simply so we can break and see the value
        var level = light.Current;

        // if the light level is above a certain level
        if (level > LIGHT_THRESHOLD)
        {
```

⌧ **STEP 2:** From the **Debug** menu select **Start Debugging** or on the keyboard press **F5** and wait for the solution to deploy and for Visual Studio to hit the breakpoint.

⌧ **STEP 3:** Hover the cursor over the variable “level” and Visual Studio will display its current value.

⌧ **STEP 4:** While holding your hand over the light sensor, press F5 a couple of times to continue and observe that the face changes and the green LED turns off when it gets dark.

⌧ **Step 5**: Press Shift-F5 to stop debugging.

EXPERIMENT 3: SENSING LIGHT AND TEMPERATURE

This lab reads the current light and temperature levels. The process is similar to Experiment 2.

⌧**STEP 1:** Delete the code between the #region tags

⌧**STEP 2:** Type the following code between the #region tags **OR** using a code snippet type **lab3** and press Tab twice.

```
using (SensorMgr lightSensor = new SensorMgr(light, SensorMgr.Sampling.Manual))
using (SensorMgr temperatureSensor = new SensorMgr(temp, SensorMgr.Sampling.Manual))
{
    while (true)
    {
        var message = String.Format("{0}C", Math.Round(temp.Temperature.DegreesCelsius, 1));
        Display(message);  // Display temp on matrix

        if (light.ReadRatio * 100 > LIGHT_THRESHOLD)
        {
            ledGreen.On();
        }
        else
        {
            ledGreen.Off();
        }
        Util.Delay(500);
    }
}
```

⌧**STEP 3:** Deploy the solution to the Raspberry Pi. From the **Debug** menu select **Start without Debugging** or from the keyboard press **Ctrl+F5** and wait for the solution to deploy.

⌧**STEP 4:** The current temperature will be displayed on the LED matrix. Try squeezing the temperature sensor between your fingers to change the temperature.

⌧**STEP 5:** Hold your hand over the light sensor now observe when it gets dark the green LED turns off.

EXPERIMENT 4: CONNECTING TO AZURE IOT HUB

This lab connects the Raspberry Pi 2 to Azure cloud services. The Maker Den supports three cloud modes.

1) Azure IoT Hub (https://azure.microsoft.com/services/iot-hub/),
2) Azure Event Hubs (https://azure.microsoft.com/services/event-hubs)
3) MQTT (Mosquitto.org service running in an Azure VM)

⌧**STEP 1:** Delete the code between the #region tags

⌧**STEP 2:** Type the following code between the #region tags **OR** using a code snippet type **lab4** and press Tab twice.

```
using (SensorMgr lightSensor = new SensorMgr(light))
using (SensorMgr tempSensor = new SensorMgr(temp))
{

    lightSensor.OnAfterMeasurement += OnAfterMeasurement;
    lightSensor.OnBeforeMeasurement += OnBeforeMeasure;
    lightSensor.OnAfterMeasurement += SetLEDMatrixBrightness;

    tempSensor.OnAfterMeasurement += OnAfterMeasurement;
    tempSensor.OnBeforeMeasurement += OnBeforeMeasure;

    DisplayTemperature().Wait();
}
```

⌧**STEP 3**: Enable IoT Hub CloudMode. Change the line that reads

```
Initialise("Replace with Your Name", CloudMode.None);
```

To

```
 Initialise("Replace with Your Name", CloudMode.IoT_Hub);
```

⌧**STEP 4:** Deploy the solution to the Raspberry Pi. From the **Debug** menu select **Start without Debugging** or from the keyboard press **Ctrl+F5** and wait for the solution to deploy.

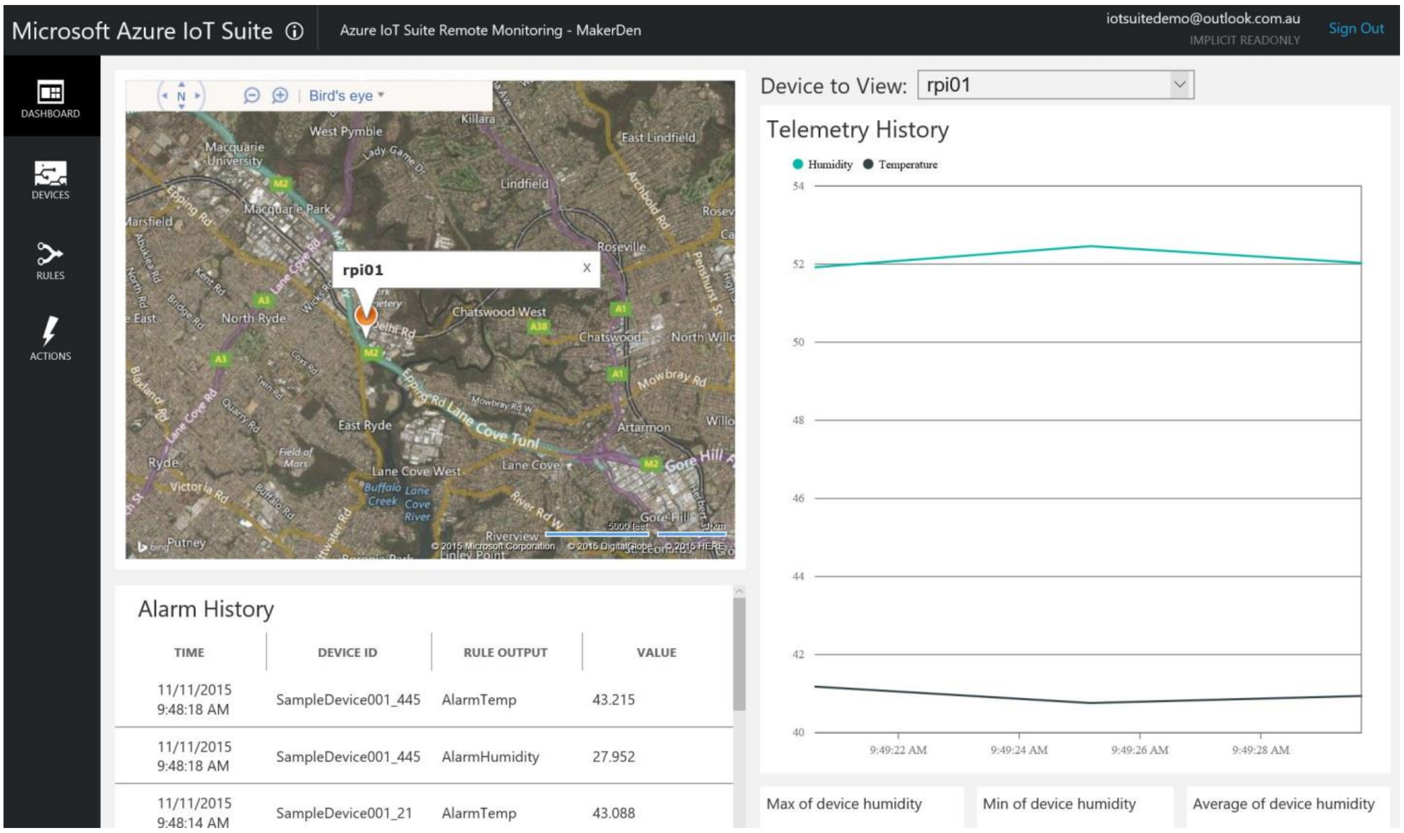
⌧**STEP 5:** Open a web browser and go to the URL https://faisterremote.azurewebsites.net/. This is the Remote Monitoring web dashboard that is part of one of the preconfigured solutions in Azure IoT Suite.

⌧**STEP 6: Sign In** using this Microsoft Account:

Email: iotsuitedemo@outlook.com.au

Password: IoThub2015

⌧**STEP 7:** In the "Device to View" drop down list, **select** your device ID. This is the same as the hostname of your Raspberry Pi 2 running Windows 10 IoT Core. Once you have selected the right device to view, you will see the Telemetry History graph, and 3 gauges for device humidity (NOTE: only temperature reading is read from the device. Humidity values are randomly generated). You may zoom in the map to locate your device's pin drop.

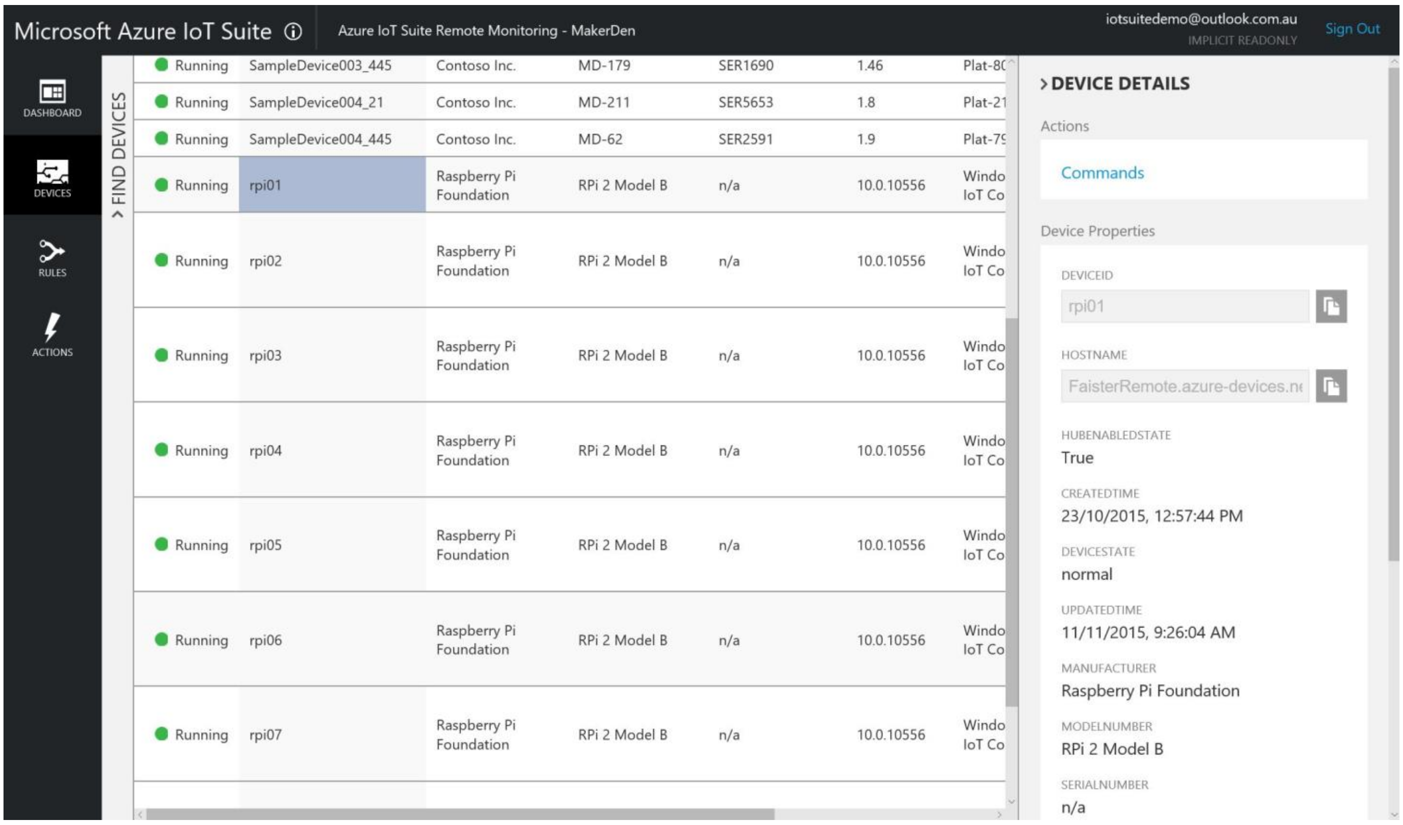


Alarm History

TIME	DEVICE ID	RULE OUTPUT	VALUE
11/11/2015 9:48:18 AM	SampleDevice001_445	AlarmTemp	43.215
11/11/2015 9:48:18 AM	SampleDevice001_445	AlarmHumidity	27.952
11/11/2015 9:48:14 AM	SampleDevice001_21	AlarmTemp	43.088

⌧**STEP 8:** Touch your finger on the temperature sensor and observe the temperature changing.

☒**STEP 9: Click DEVICES** in the left navigation bar. This brings you to the Devices List. **Select** your device. **Click DEVICE DETAILS** in the right-hand corner of the dashboard. You will find the device metadata associated to this device. This is stored within a DocumentDB store which is also preconfigured as part of the Azure IoT Suite Remote Monitoring solution.


Microsoft Azure IoT Suite
Azure IoT Suite Remote Monitoring - MakerDen
iotsuitedemo@outlook.com.au
IMPLICIT READONLY
Sign Out
DASHBOARD
DEVICES
RULES
ACTIONS
FIND DEVICES

Running	SampleDevice003_445	Contoso Inc.	MD-179	SER1690	1.46	Plat-8
Running	SampleDevice004_21	Contoso Inc.	MD-211	SER5653	1.8	Plat-21
Running	SampleDevice004_445	Contoso Inc.	MD-62	SER2591	1.9	Plat-7
Running	rpi01	Raspberry Pi Foundation	RPi 2 Model B	n/a	10.0.10556	Windo IoT Co
Running	rpi02	Raspberry Pi Foundation	RPi 2 Model B	n/a	10.0.10556	Windo IoT Co
Running	rpi03	Raspberry Pi Foundation	RPi 2 Model B	n/a	10.0.10556	Windo IoT Co
Running	rpi04	Raspberry Pi Foundation	RPi 2 Model B	n/a	10.0.10556	Windo IoT Co
Running	rpi05	Raspberry Pi Foundation	RPi 2 Model B	n/a	10.0.10556	Windo IoT Co
Running	rpi06	Raspberry Pi Foundation	RPi 2 Model B	n/a	10.0.10556	Windo IoT Co
Running	rpi07	Raspberry Pi Foundation	RPi 2 Model B	n/a	10.0.10556	Windo IoT Co

> DEVICE DETAILS

Actions

Commands

Device Properties

DEVICEID
rpi01

HOSTNAME
FaisterRemote.azure-devices.n

HUBENABLEDSTATE
True

CREATEDTIME
23/10/2015, 12:57:44 PM

DEVICESTATE
normal

UPDATEDTIME
11/11/2015, 9:26:04 AM

MANUFACTURER
Raspberry Pi Foundation

MODELNUMBER
RPi 2 Model B

SERIALNUMBER
n/a

EXPERIMENT 4A (OPTIONAL): CONNECTING TO MQTT SERVICE ON AZURE

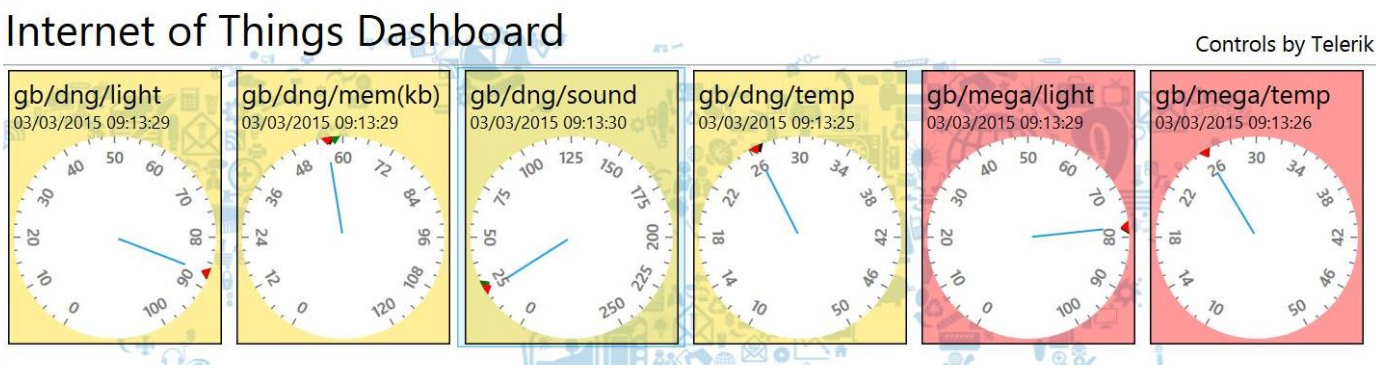
This lab connects the Maker Den to a Mosquitto (http://mosquitto.org Open Source MQTT Server) running in an Azure Virtual Machine.

⌧**STEP 1**: Enable streaming to a MQTT service. Change the CloudMode to MQTT

```
Initialise("Replace with Your Name", CloudMode.MQTT);
```

⌧**STEP 2:** Deploy the solution to the Raspberry Pi. From the **Debug** menu select **Start without Debugging** or from the keyboard press **Ctrl+F5** and wait for the solution to deploy.

⌧**STEP 3:** Press Windows Key, type "**iot**" and start the "**IoT Dashboard**".



(IoT Dashboard installable from here)

⌧**STEP 4:** Hover your hand over the light sensor, look for your Raspberry Pi name on the IoT Dashboard and observe the light level changing.

EXPERIMENT 5: NOT SO FAST COWBOY

Congratulations, you have successfully completed the Maker Den Experience. You have done some hardware prototyping, deployed a Universal Windows App to a Raspberry Pi 2, streamed data to Microsoft Azure, ingested telemetry using Azure IoT Hub and visualised data with Azure IoT Suite Remote Monitoring solution and the IoT Dashboard.

Please complete the following steps before you leave.

⌧**STEP 1**: Close Visual Studio

⌧**STEP 2**: Leave all the components in the breadboard and the Raspberry Pi running and streaming data to Azure.

⌧**STEP 3: Take 30 seconds to complete the Maker Den Evaluation at** http://aka.ms/ignite2015makerden (this is a shortcut on desktop)

⌧**STEP 4:** Make sure you get scanned to be in the draw for a Raspberry Pi 2 plus prototyping gear.

All the documentation and software for the Maker Den is available at http://www.github.com/makerden