Windows 10 IoT Core Internet of Things Maker Den Lab Guide

Document Version 2.0

Social	Twitter #makerden #iot #raspberrypi2 #windows10
Document Authors	Dave Glover | dglover@microsoft.com | @dglover Andrew Coates |acoat@microsoft.com |@coatsy
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.

The following components are used for the Maker Den.

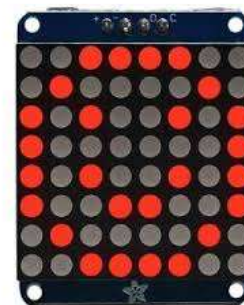|	
Raspberry Pi 2 These labs are built on the Raspberry Pi 2 running Windows 10 IoT Core. You can find out more about Windows 10 IoT Core at http://dev.windows.com/iot.	
Analogue Temperature Sensor (Microchip MCP9700A) Reads the temperature and reports it as a value in Degrees Centigrade.	
Light Dependent 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.	

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].

---
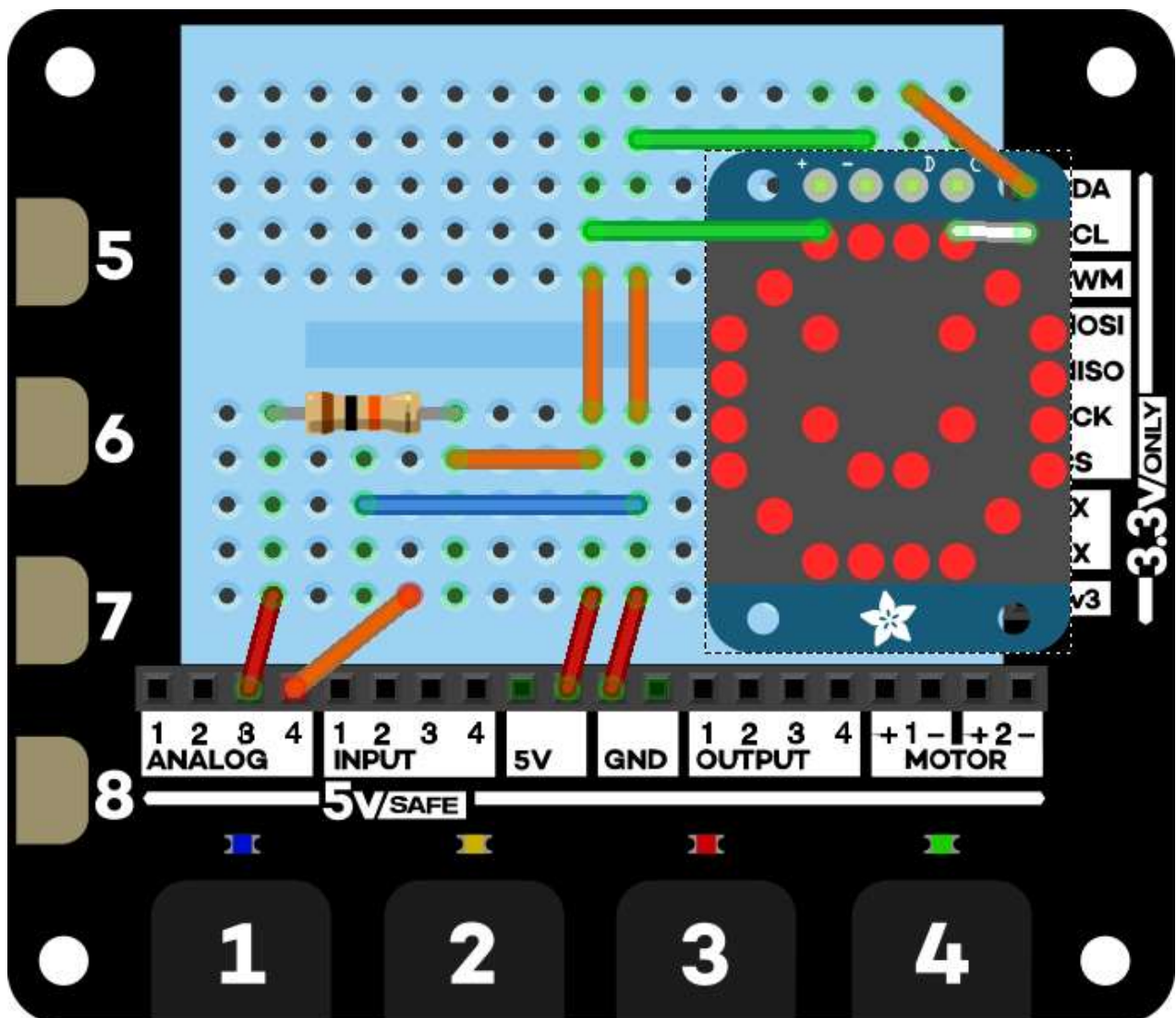
[1] Breadboards work on the principle that each of the five sockets in a column are wired in common. So, by inserting a component into a socket in the same column as another component, those components will be electrically coupled, as though their wires were joined. This makes it very simple to quickly 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.
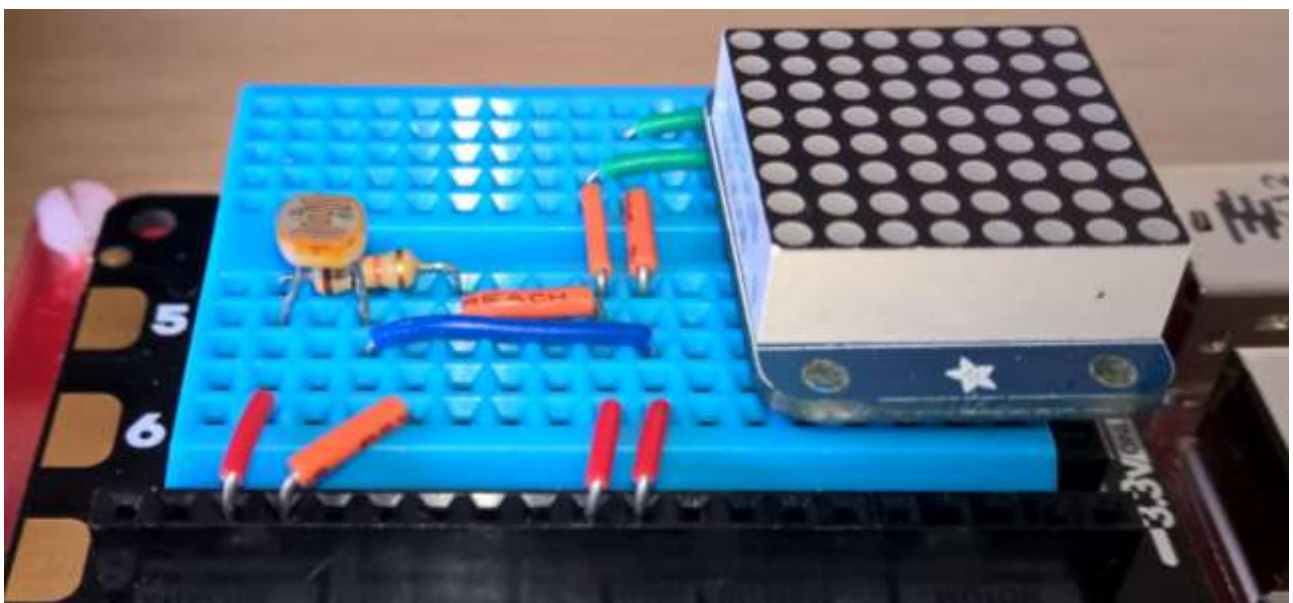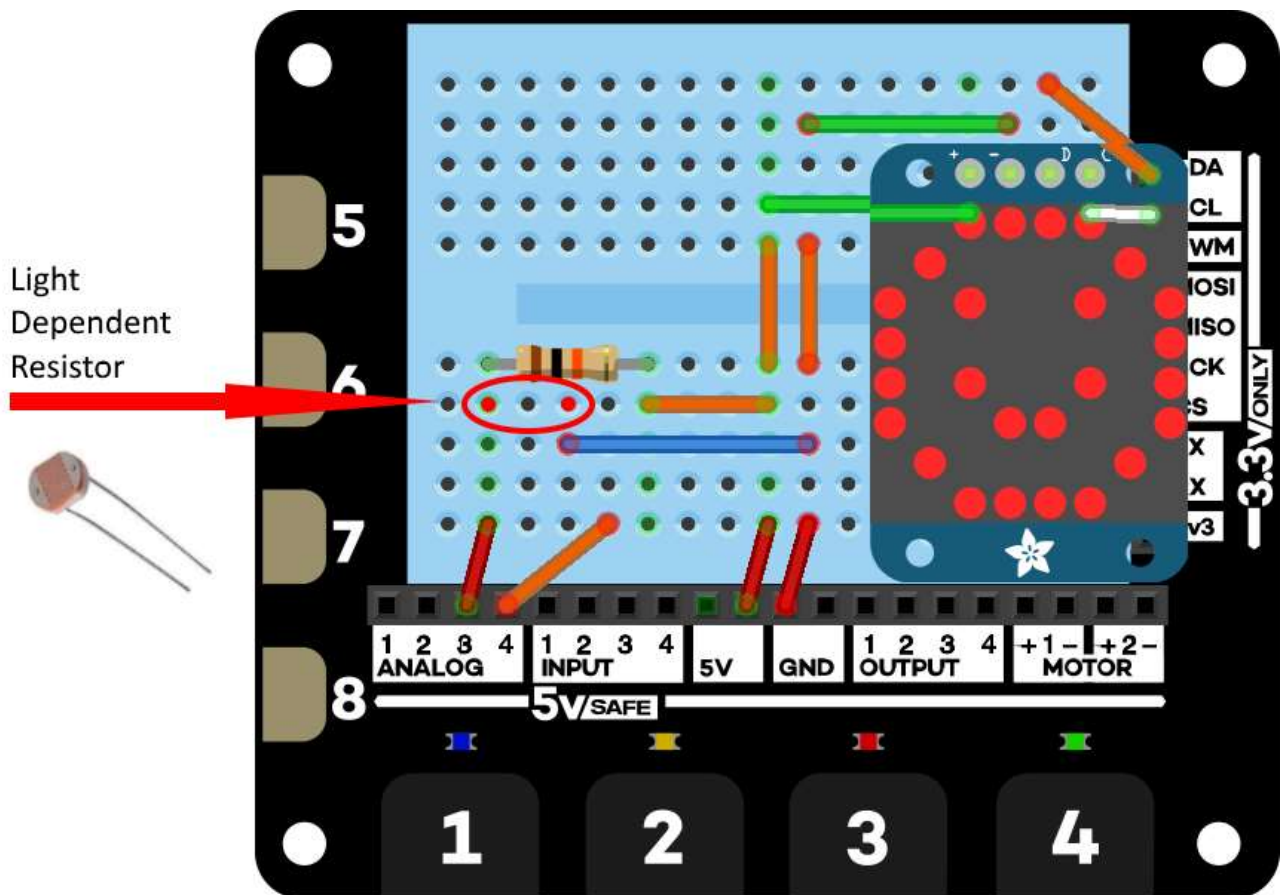
ADDING THE TEMPERATURE SENSOR

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**.

Temperature Sensor

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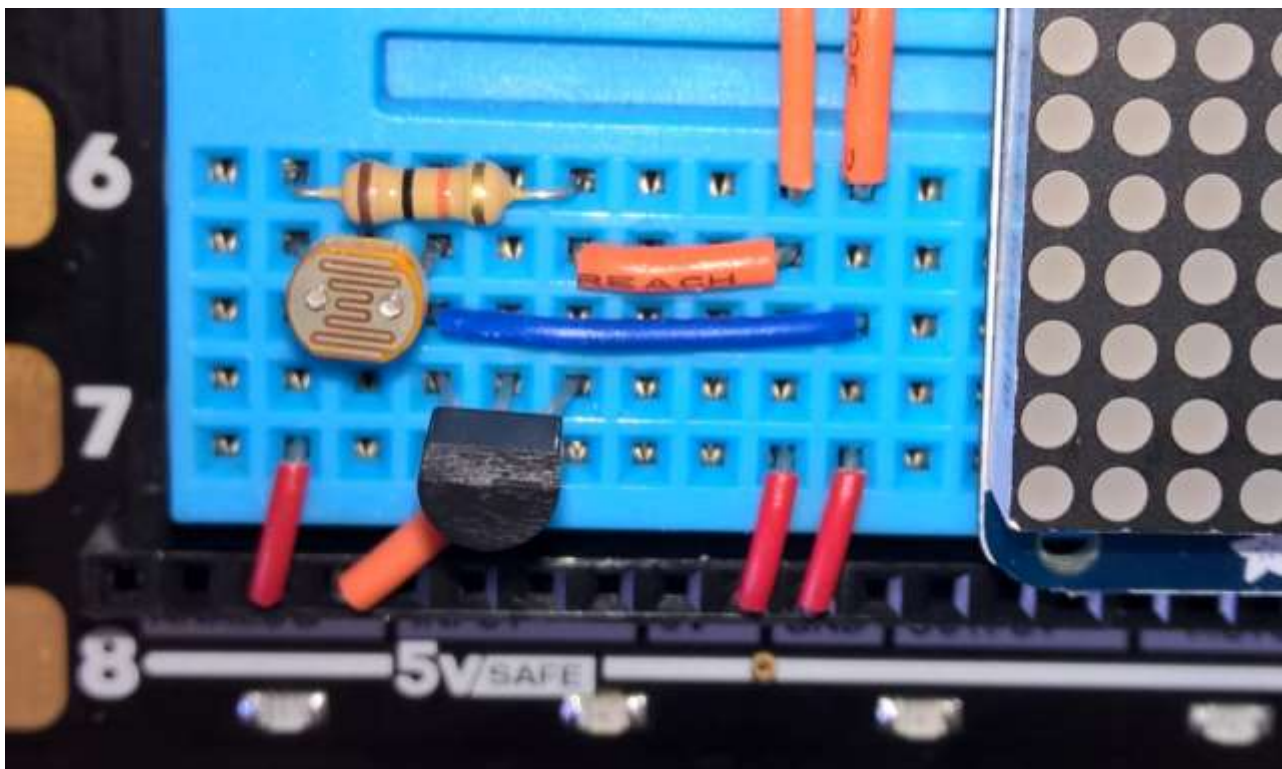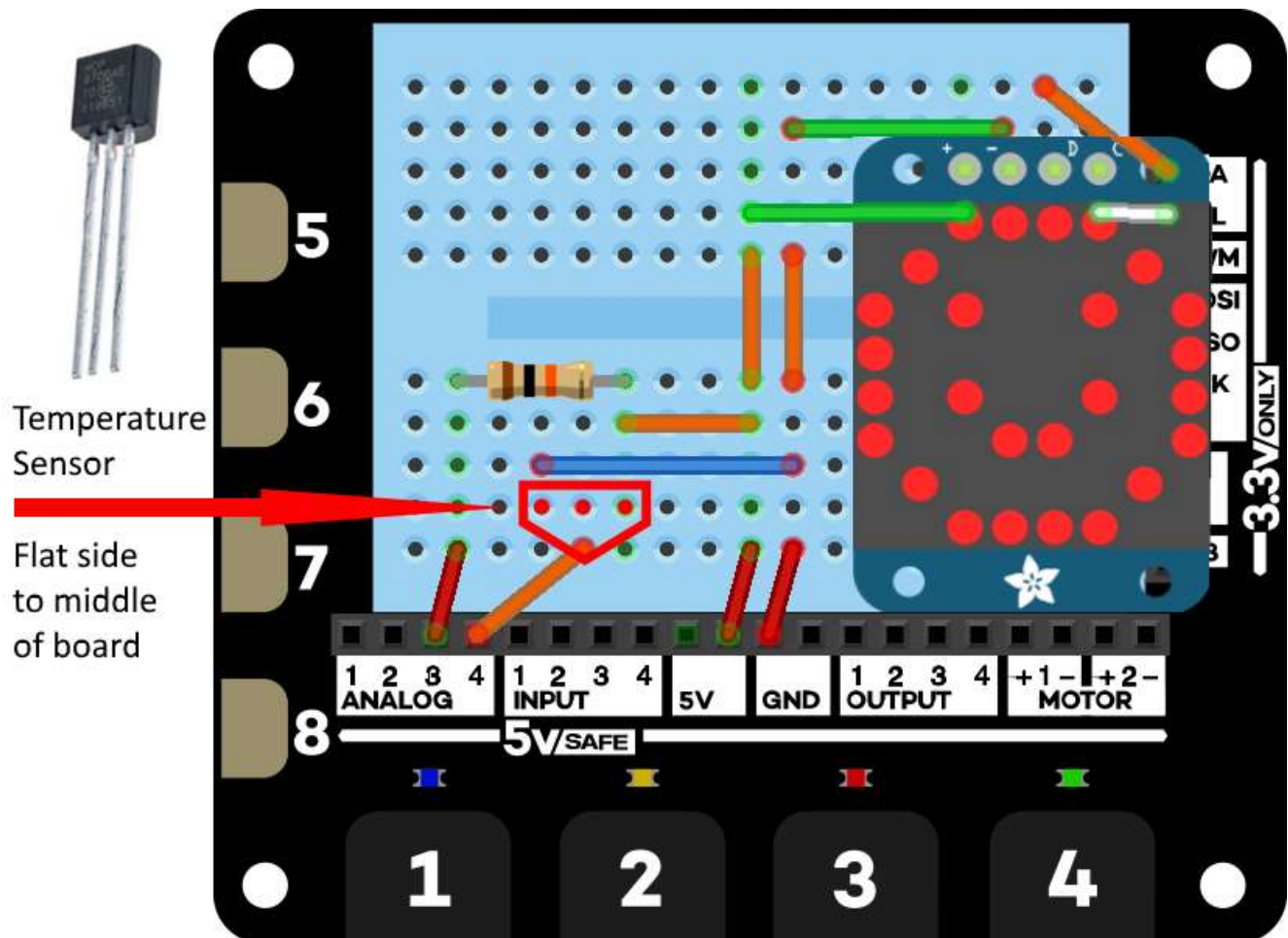
There are five Maker Den experiments to get you started with Windows 10 IoT Core and Microsoft Azure IoT Services.

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.
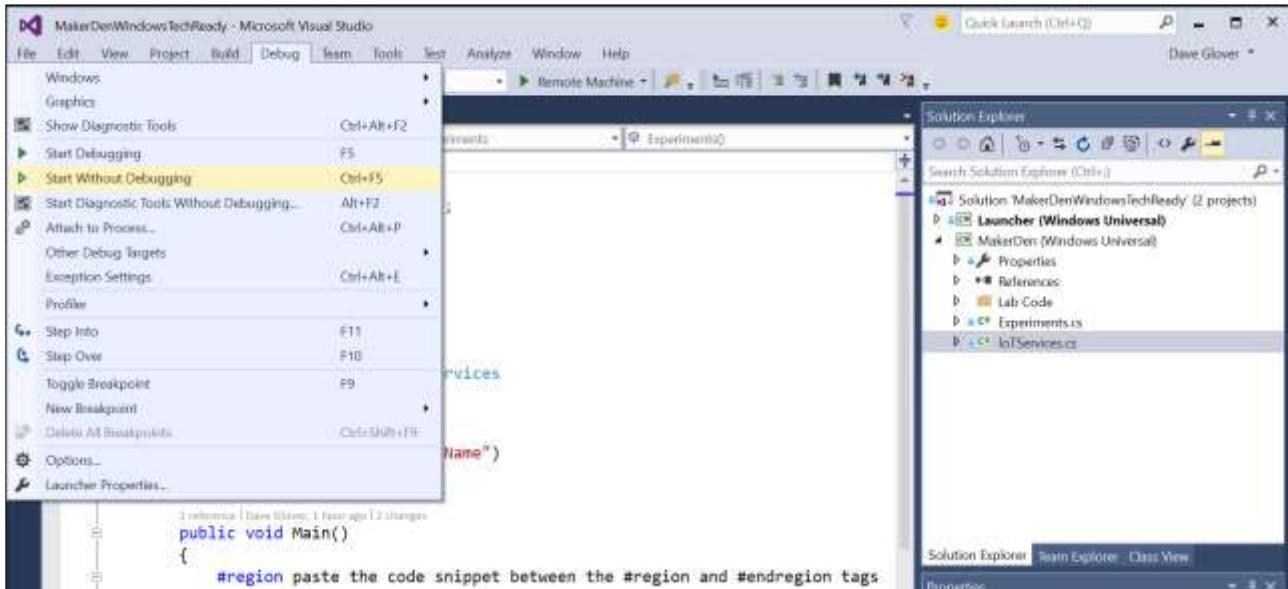
☒ **STEP 1**: Expand the **MakerDen** project then double click the **Experiments.cs** file to open it.
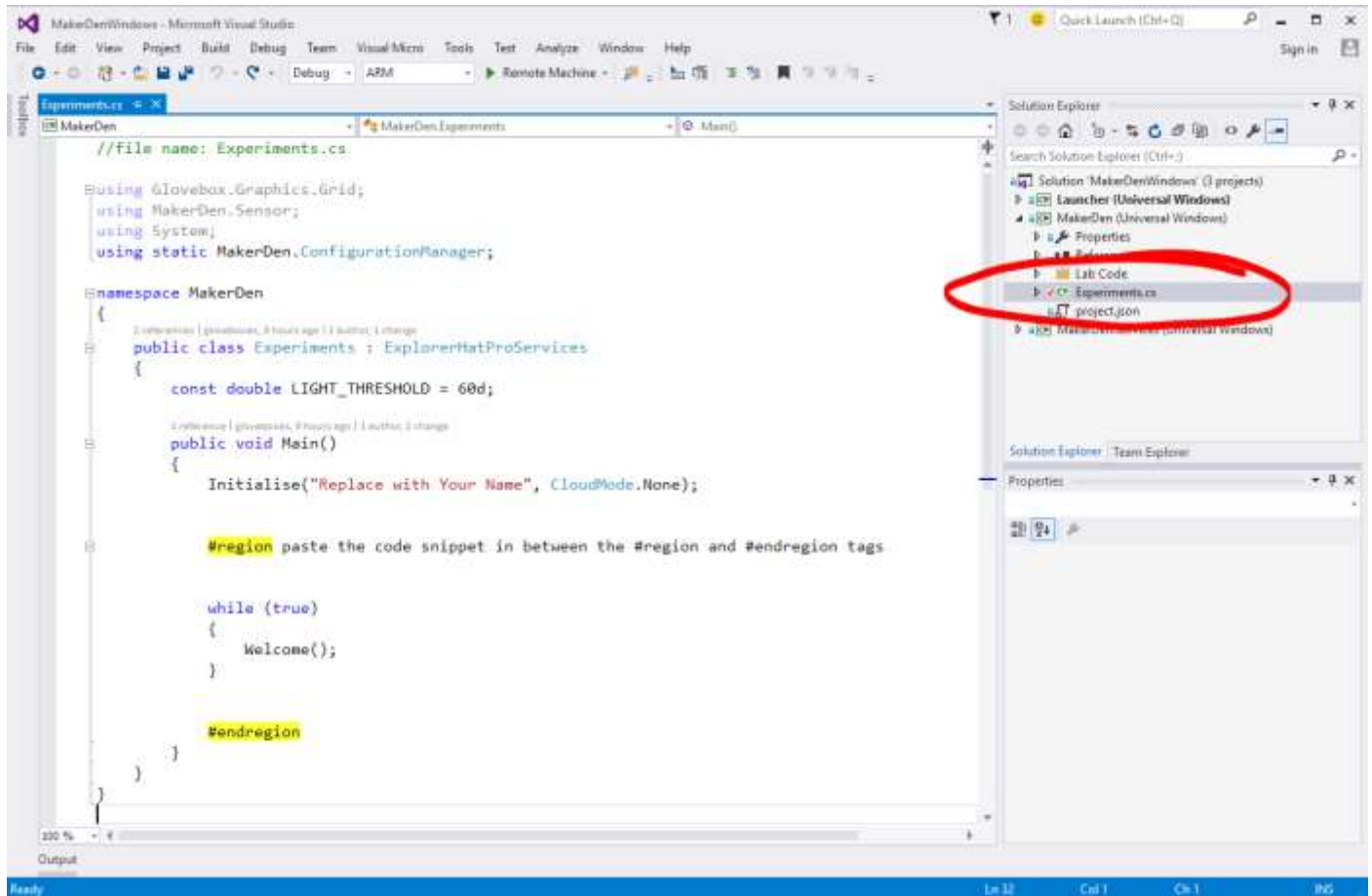
☒ **STEP 2:** Edit Experiments.cs by typing your name where it says "Replace with Your Name". Be sure to type your name inside the quotation marks.

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

⊠ **STEP 4:** Check that Visual Studio has successfully compiled and deployed the code by looking at the output window and the status bar.
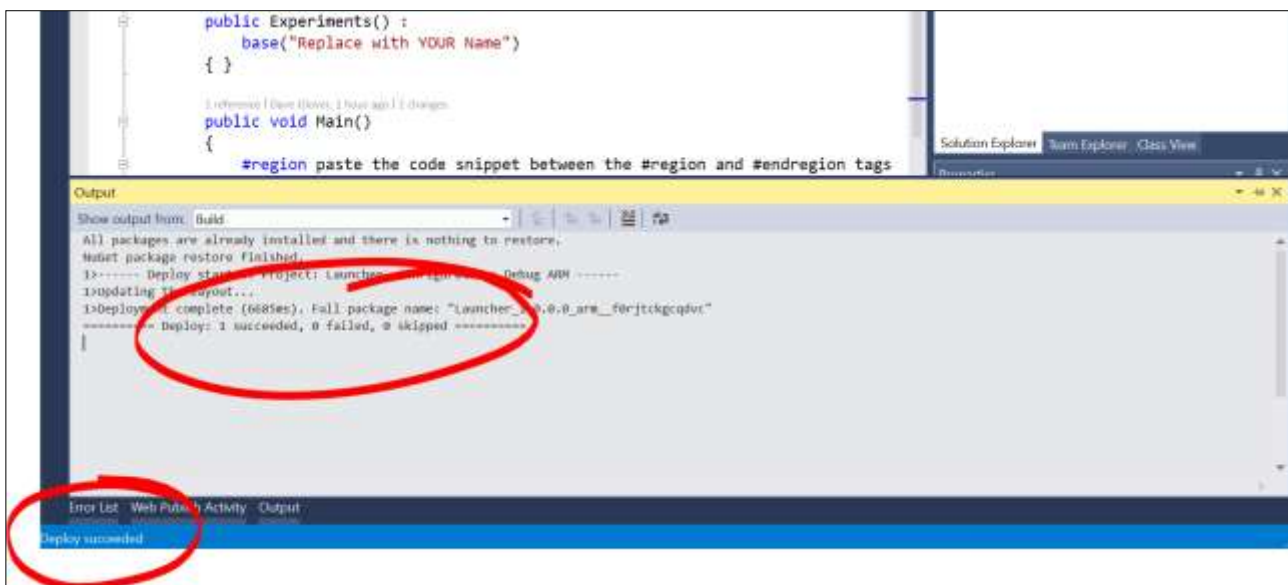
⊠ **STEP 5:** Check the LED Matrix on the Raspberry Pi. You should see your name, the machine name and the IP Address scrolling on the LED Matrix display.
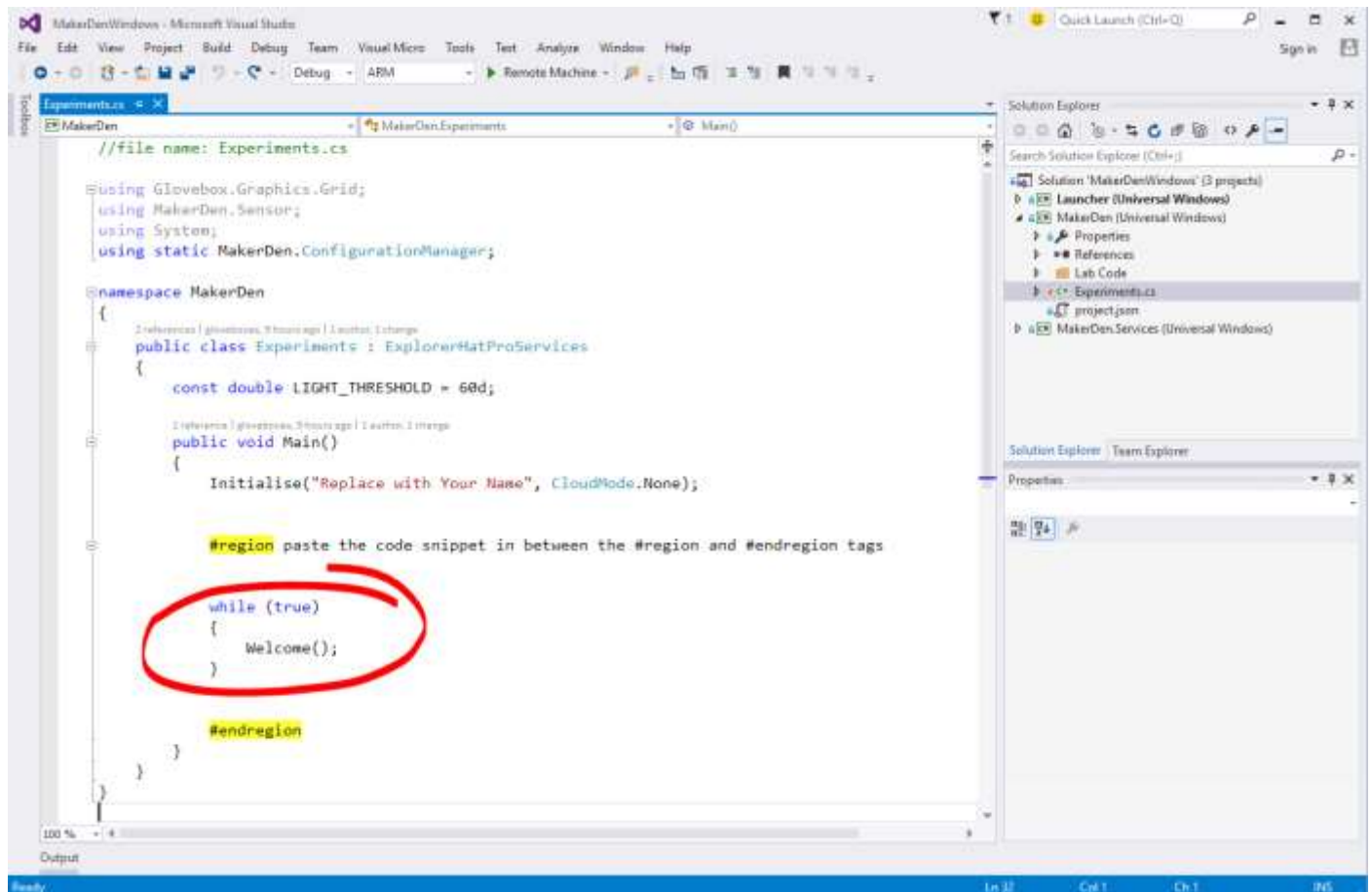
⊠ **STEP 6:** Pat yourself on the back, you did it☺

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.

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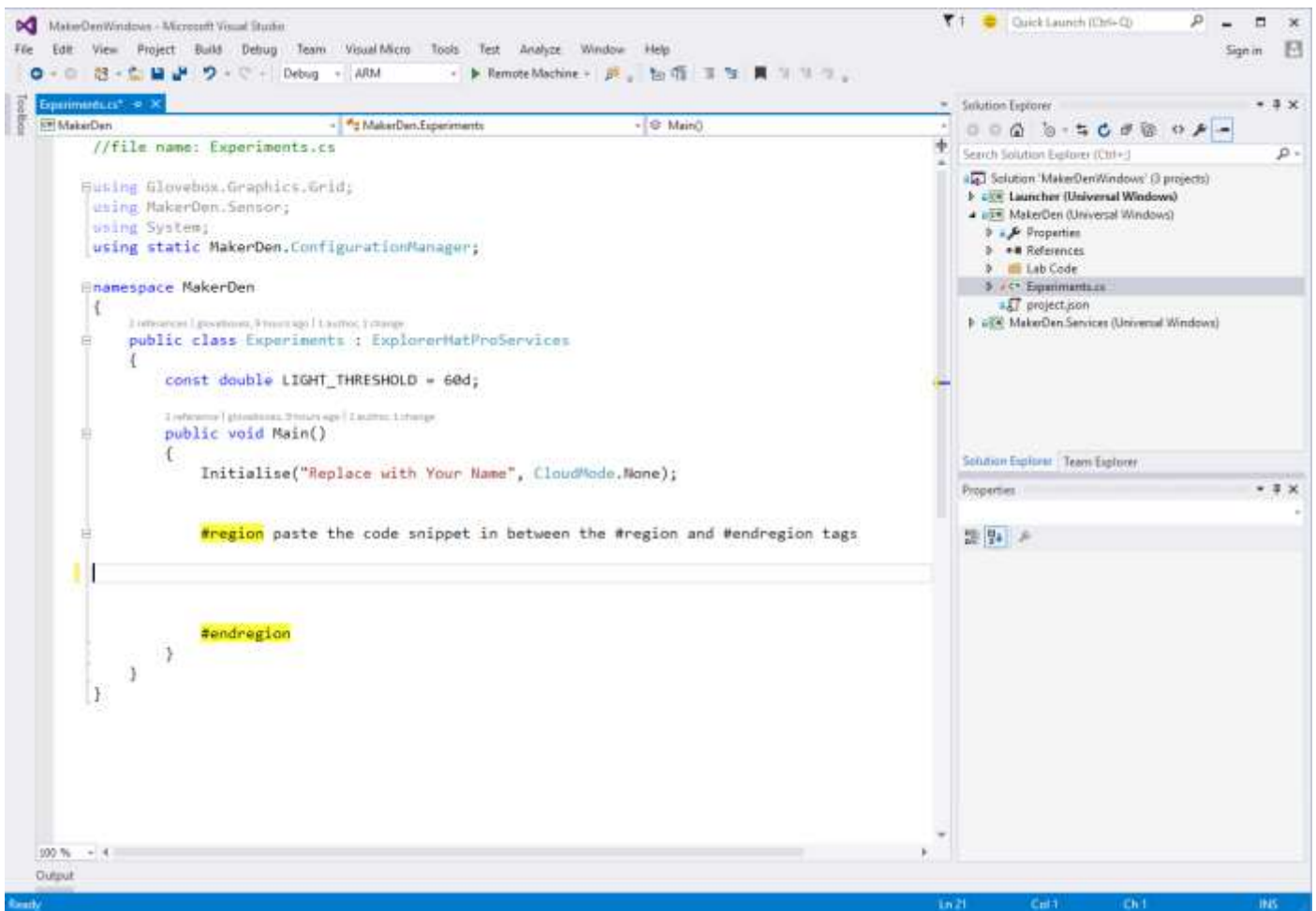
☒**STEP 3:** Type the following code between the #region tags **OR** using a code snippet type **lab2** and press Tab twice.

```
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.

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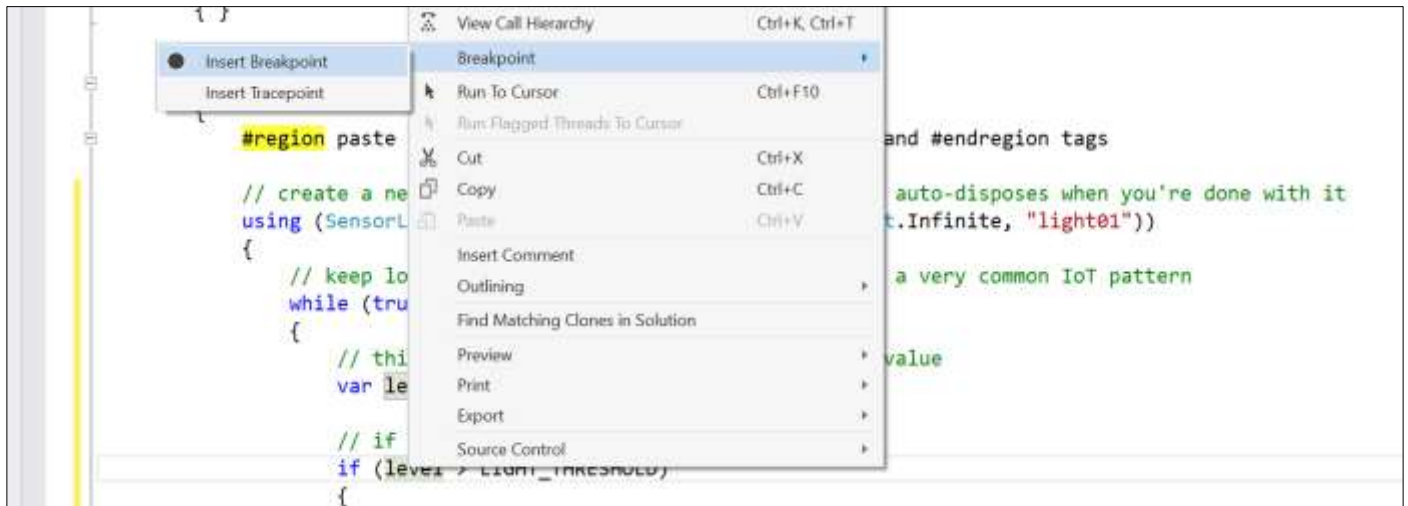
⊠**STEP 6:** Hover your hand over the light sensor and observe that the face changes and the green LED turns off when it gets dark.

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.

☒ **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.

```csharp
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.

14

EXPERIMENT 4: CONNECTING TO AZURE

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 IoTHub 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:** <mark>Note about IoT Hub</mark>

More Instructions

Screen shot required

⊠ **STEP 6**: Enable streaming to a MQTT service. Change the CloudMode to MQTT

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

⊠ **STEP 7:** 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 8:** Press Windows Key, type "**iot**" and start the "**IoT Dashboard**".
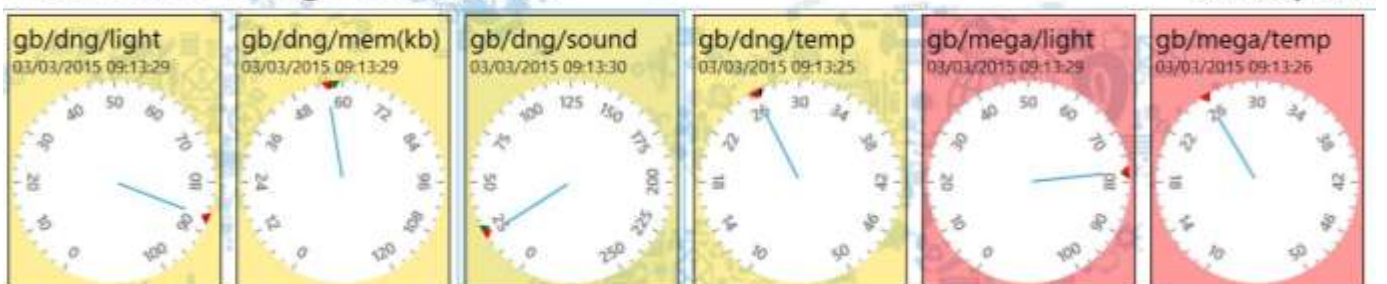
(IoT Dashboard installable from here)

⊠ **STEP 9:** 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, and visualised data with Azure IoT hub 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/TR21MakerdenEval

⊠ **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