



# Windows 10 IoT Core Internet of Things Maker Den Lab Guide

Document Version 2.0

<b>Social</b>	<b>Twitter</b> #makerden #iot #raspberrypi2 #windows10
<b>Document Authors</b>	Dave Glover   <a href="mailto:dglover@microsoft.com">dglover@microsoft.com</a>   @dglover Andrew Coates   <a href="mailto:acoat@microsoft.com">acoat@microsoft.com</a>   @coatsy
<b>Document location</b>	<a href="https://github.com/MakerDen/IoT-Maker-Den-Documentation-and-Guides">https://github.com/MakerDen/IoT-Maker-Den-Documentation-and-Guides</a>
<b>Source Code Location</b>	<a href="https://github.com/MakerDen/IoT-Maker-Den-Windows-for-IoT">https://github.com/MakerDen/IoT-Maker-Den-Windows-for-IoT</a>
<b>Disclaimer</b>	All care has been taken to ensure the accuracy of this document. No liability accepted.
<b>Copyright</b>	You are free to reuse and modify this document and associated software.

## INTRODUCTION

Welcome to the Internet of Things Maker Den Lab where you'll 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'll experience wiring circuits, deploying code, streaming sensor data to Microsoft Azure Event Hubs and visualising the data from Stream Analytics. Once you've completed these exercises, you'll be better equipped to have an IoT discussion with your customers and partners.

## GETTING STARTED

If you are setting up your own Maker Den then refer to the appendix for information on the software setup including Visual Studio and hardware setup. All source code is available at <https://github.com/MakerDen/IoT-Maker-Den-Windows-for-IoT>.

## 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. Coding skills are not essential, as long as you can follow "copy and paste" instructions.

README FIRST

Ways to Enter Code

- 1) **INSERT SNIPPETS**
- 2) **COPY AND PASTE**
- 3) **TYPE**

## 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.



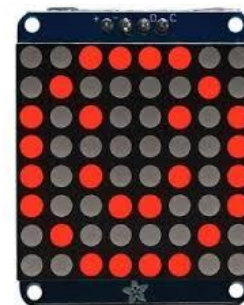
### Light Dependent Resistor (LDR) aka Photoresistor

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

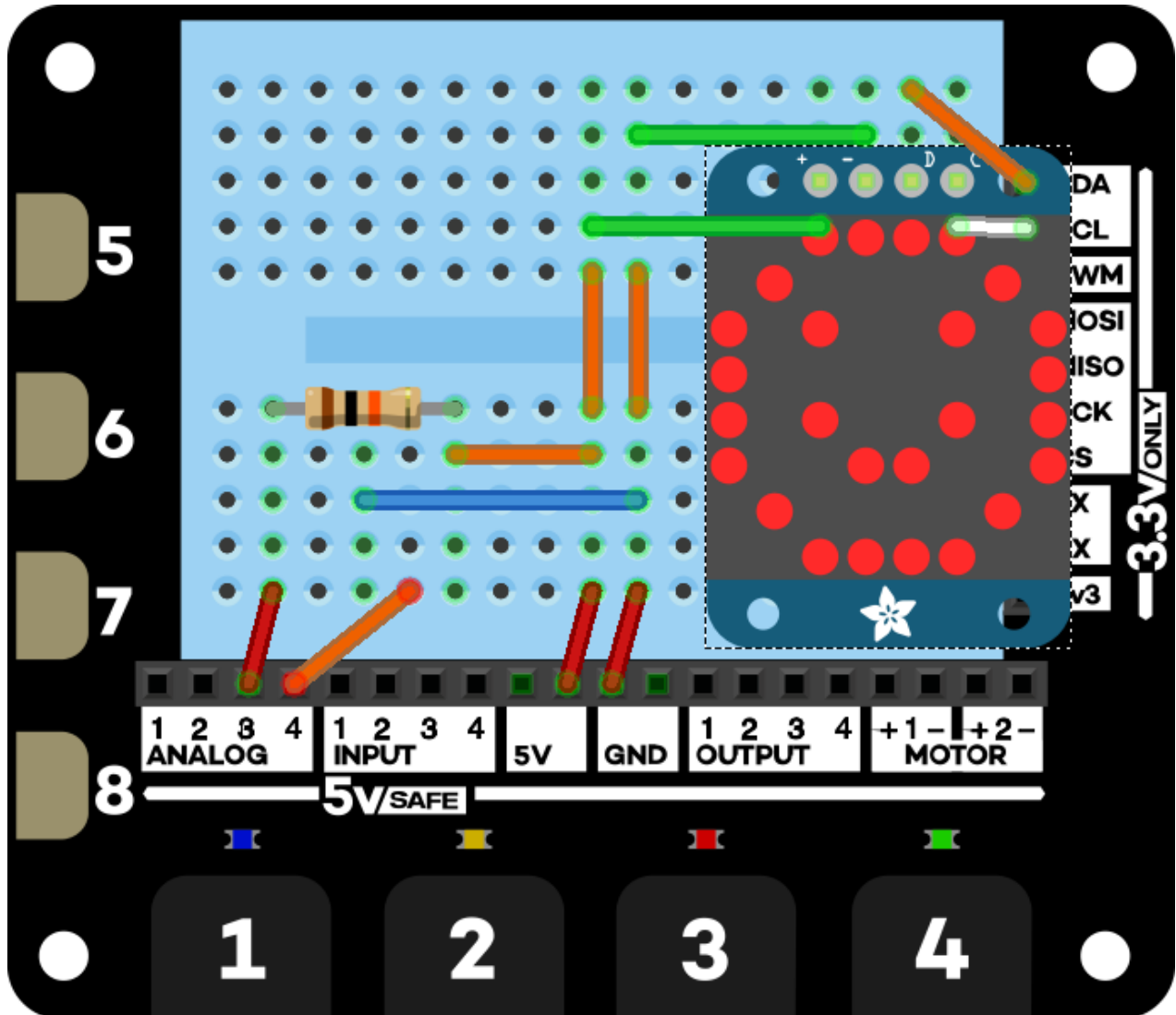
Useful prototyping HAT for RPi, has I2C ADC, I2C capacitive touch pads, motor driver, and a breadboard for prototyping.



## SETTING UP YOUR PROTOTYPE

The first task is to add two sensors plus the Adafruit Mini 8x8 LED Matrix to the Prototyping HAT.

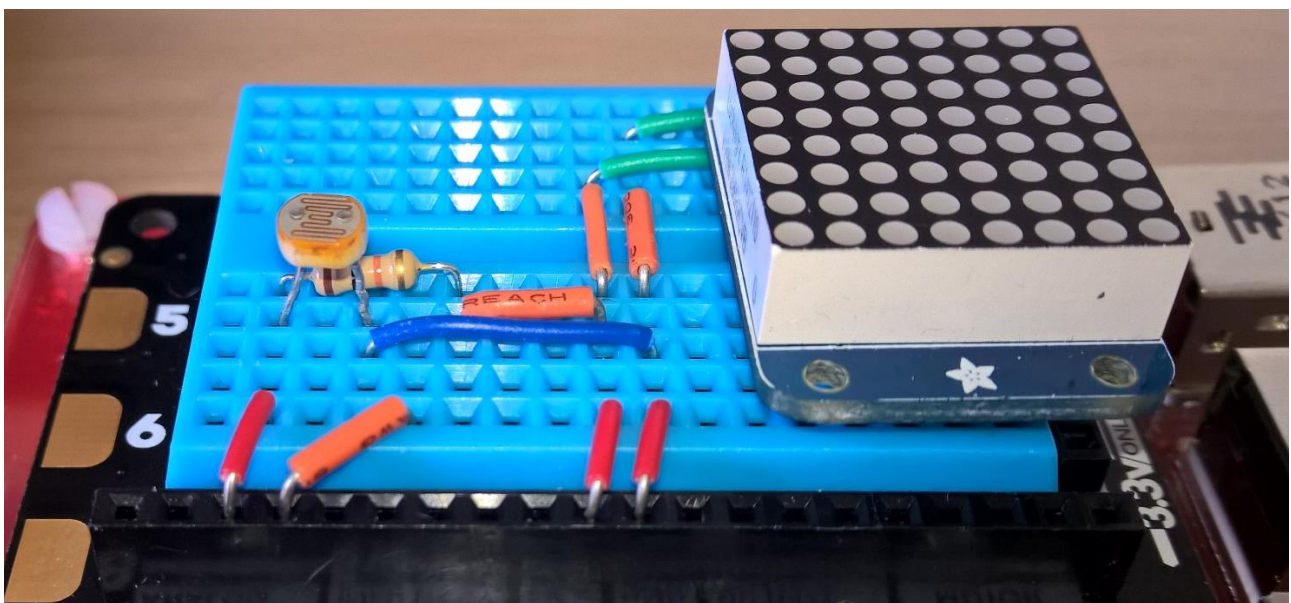
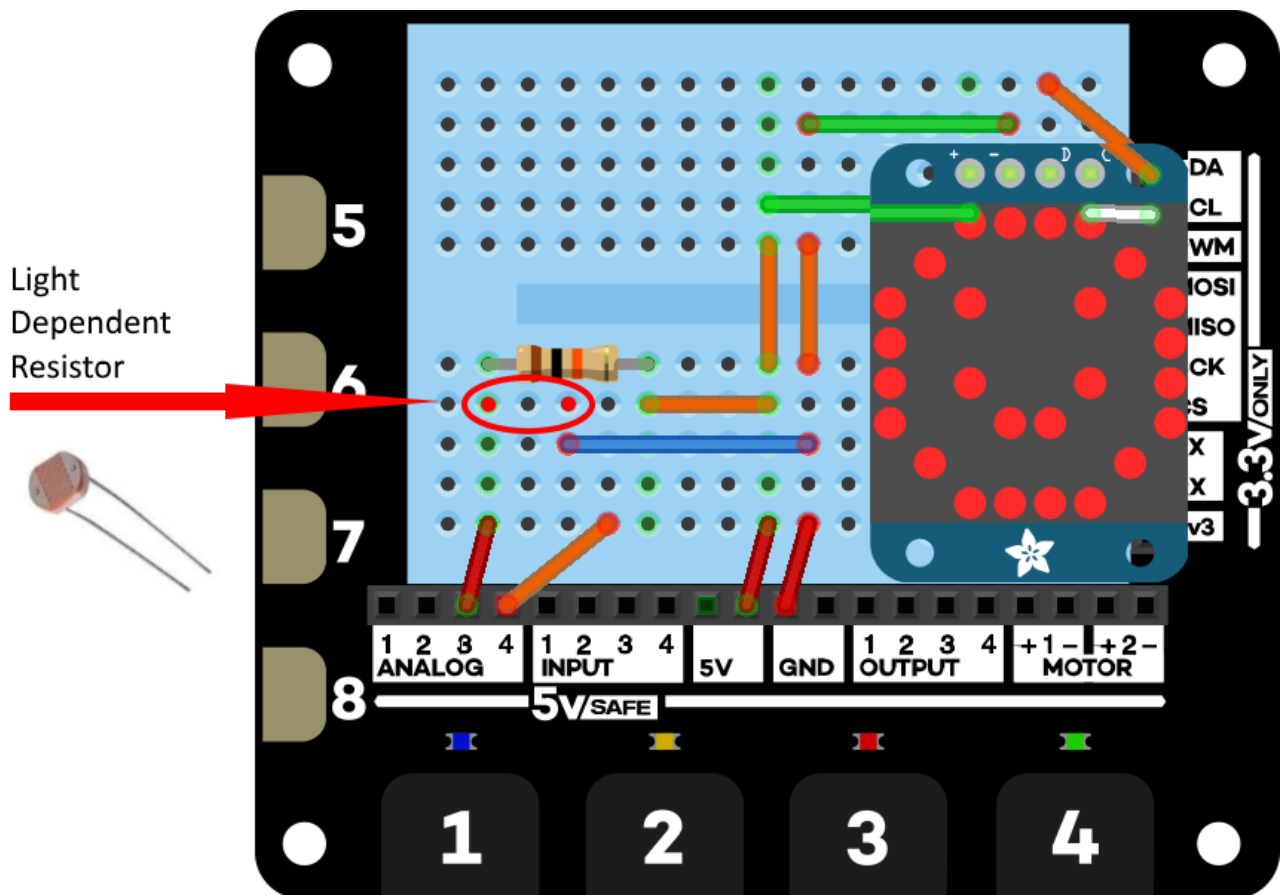
Your Breadboard Protoshield should look like the image below before you add the sensors and the Adafruit Mini Led Matrix<sup>1</sup>.



<sup>1</sup> 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)

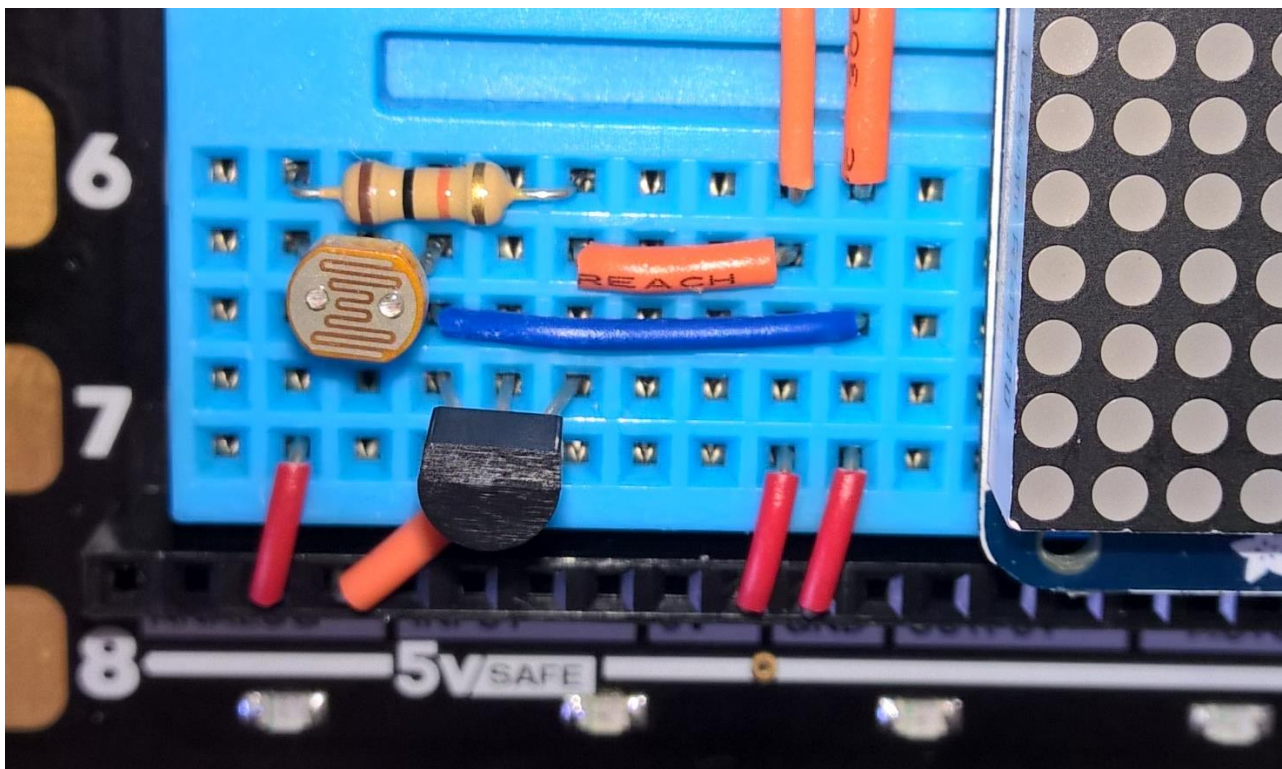
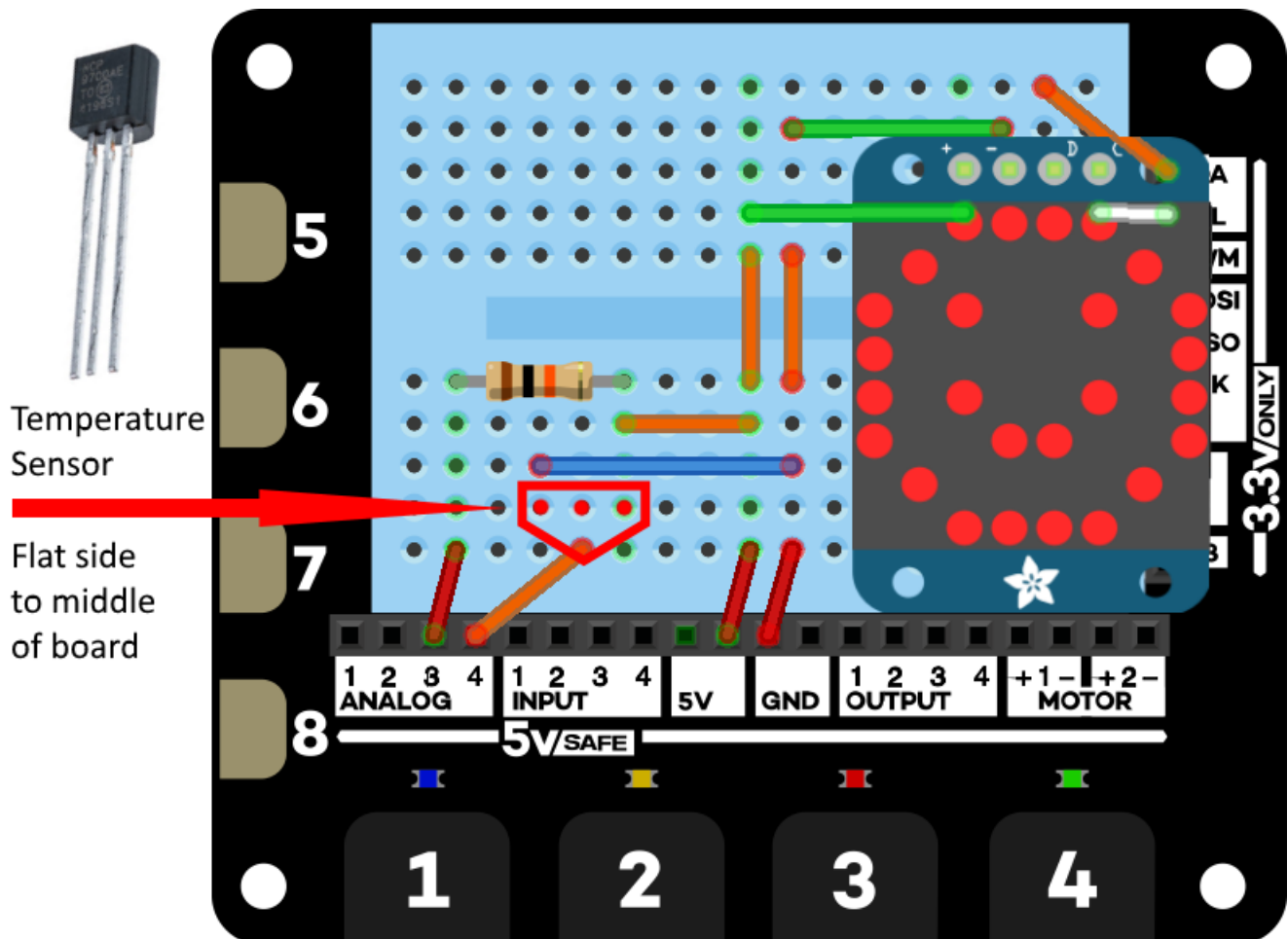
Insert the Light Dependent Resistor pins into the holes highlighted by the red dots on the breadboard image below.





## 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 experiments, all source code is provided. Feel free to explore, delve, be curious, and have fun.

---

## RESETTING THE LABS

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

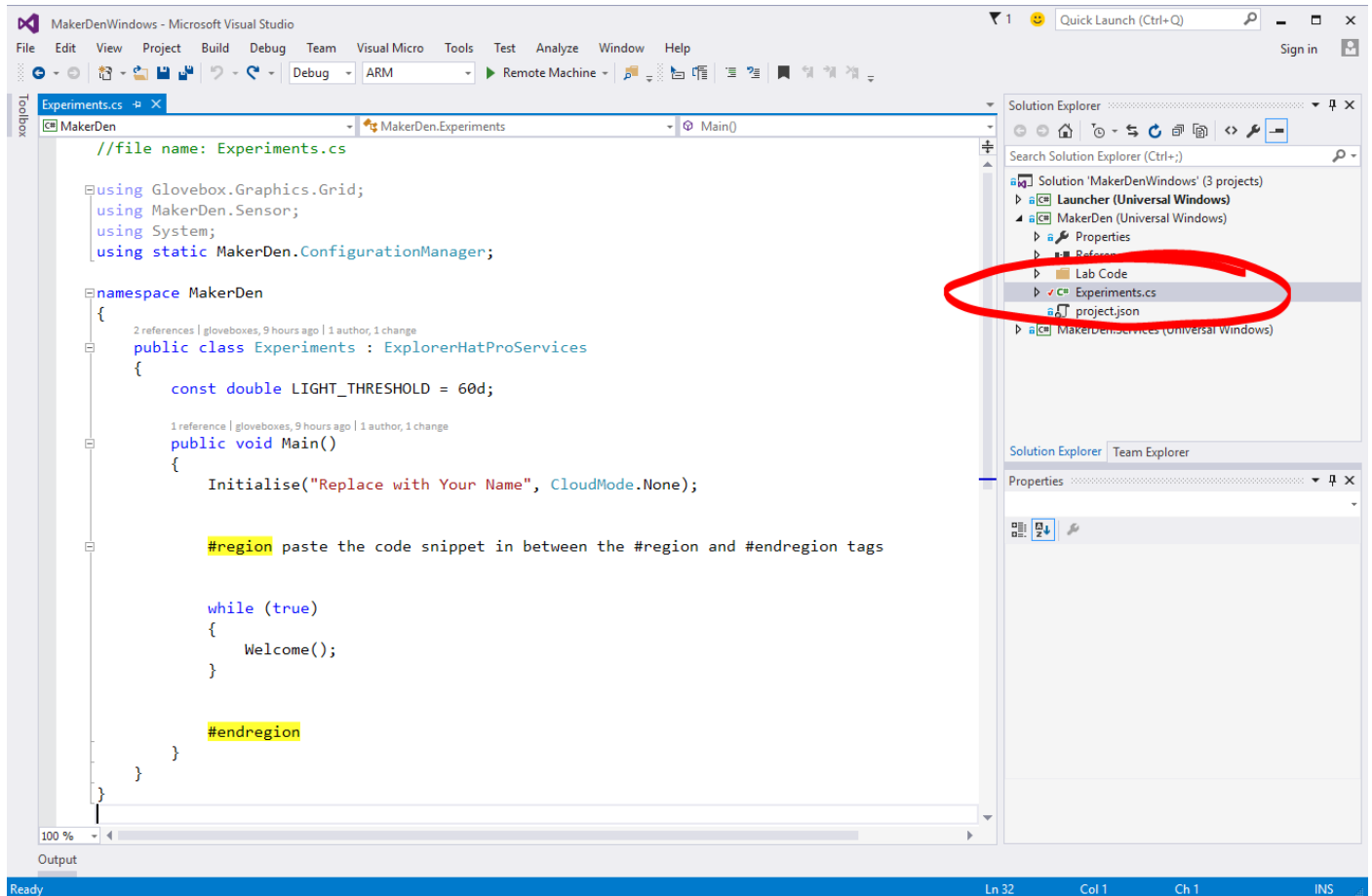
☒ **STEP 2:** Double click the **ResetLabs.bat** file on your desktop. This will copy the source code down from a GitHub repository and launch Visual Studio with the solution opened.



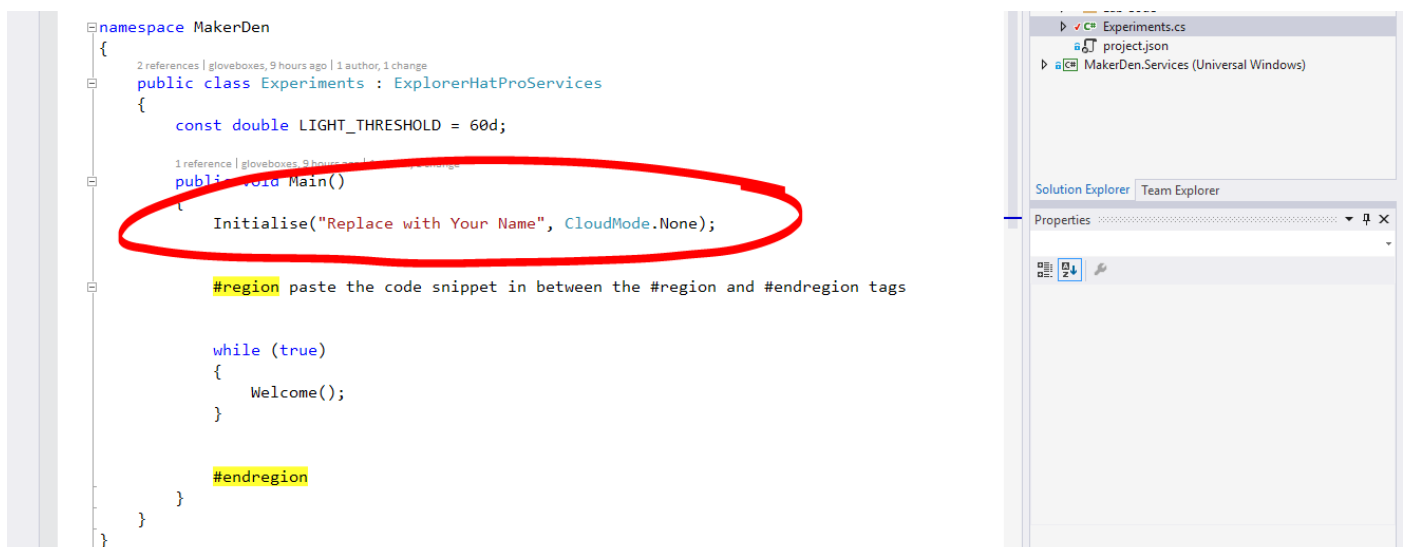
## EXPERIMENT 1: HELLO THERE

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

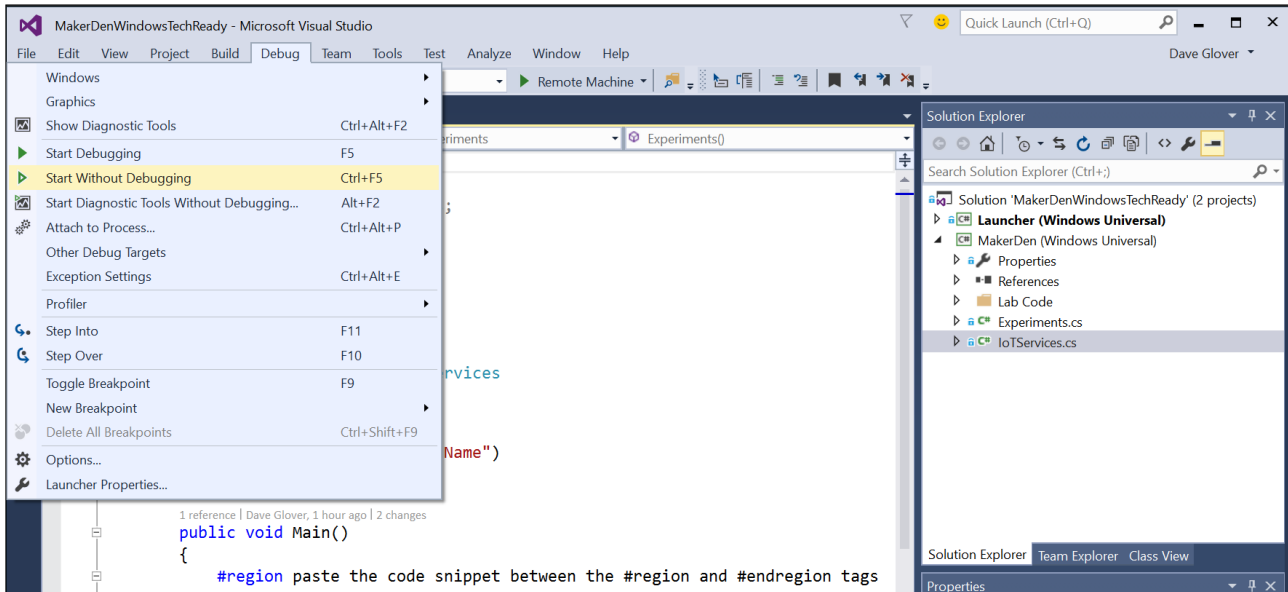
✳️ **STEP 1:** Expand the **MakerDen** project then double click the **Experiments.cs** file to open it.



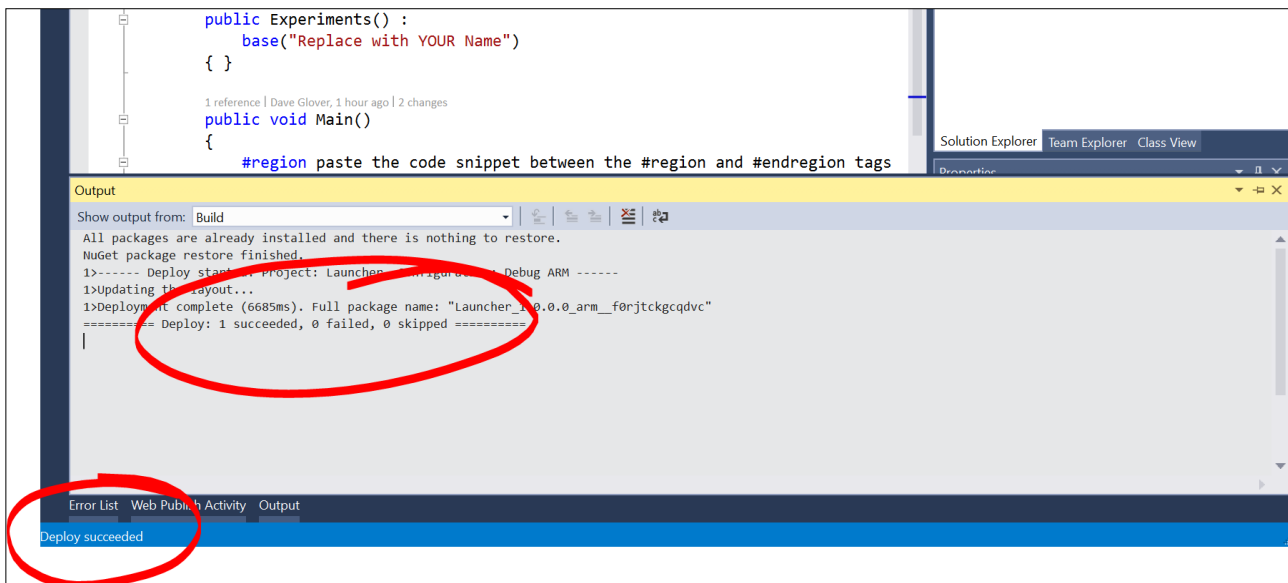
✳️ **STEP 2:** Edit Experiments.cs by typing your name where it says “Replace with Your Name”. Be sure to type your name inside the quotation marks.



✗ **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.

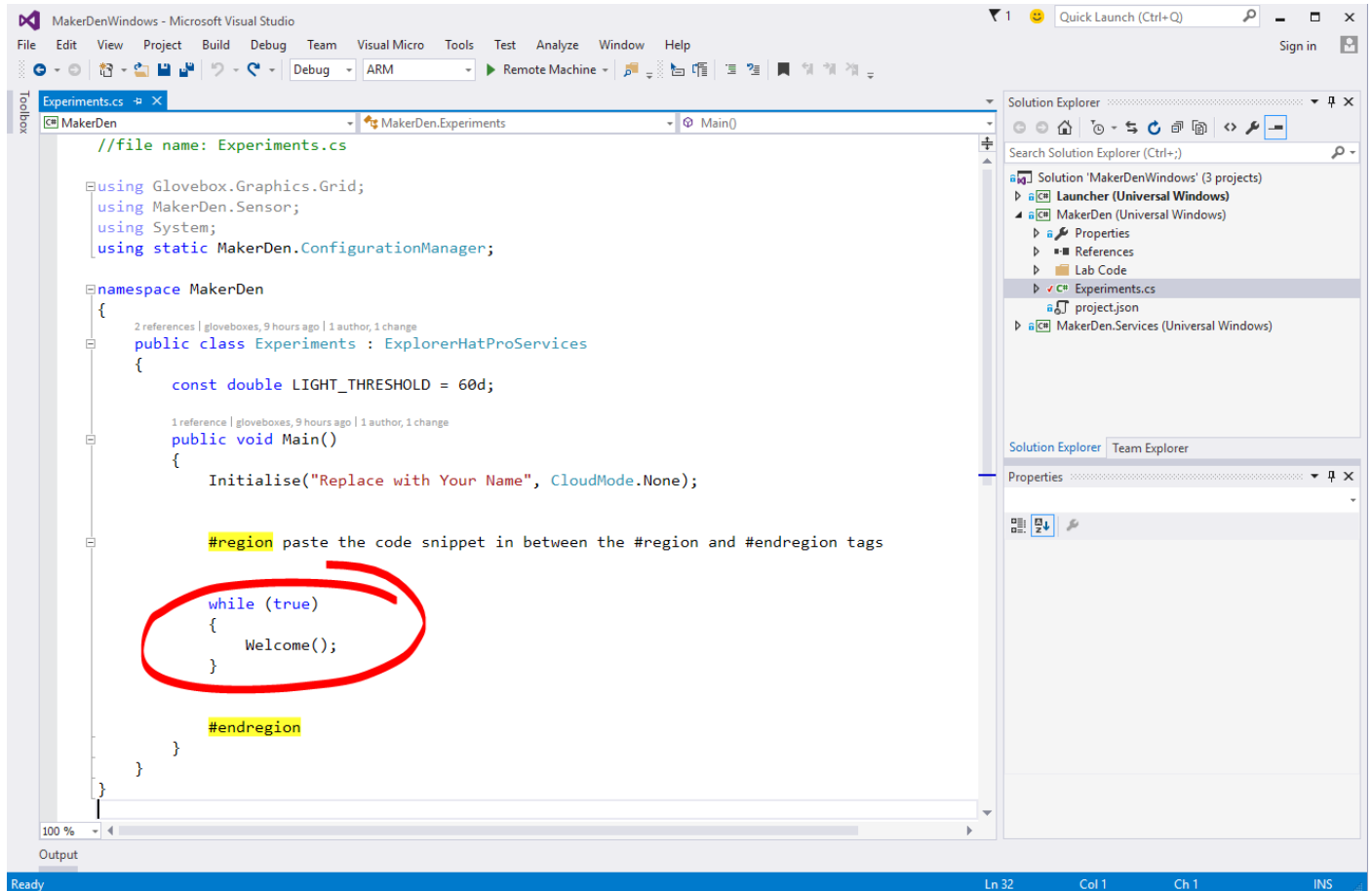
✗ **STEP 6:** Pat yourself on the back, you did it 😊

## EXPERIMENT 2: SENSING THE WORLD

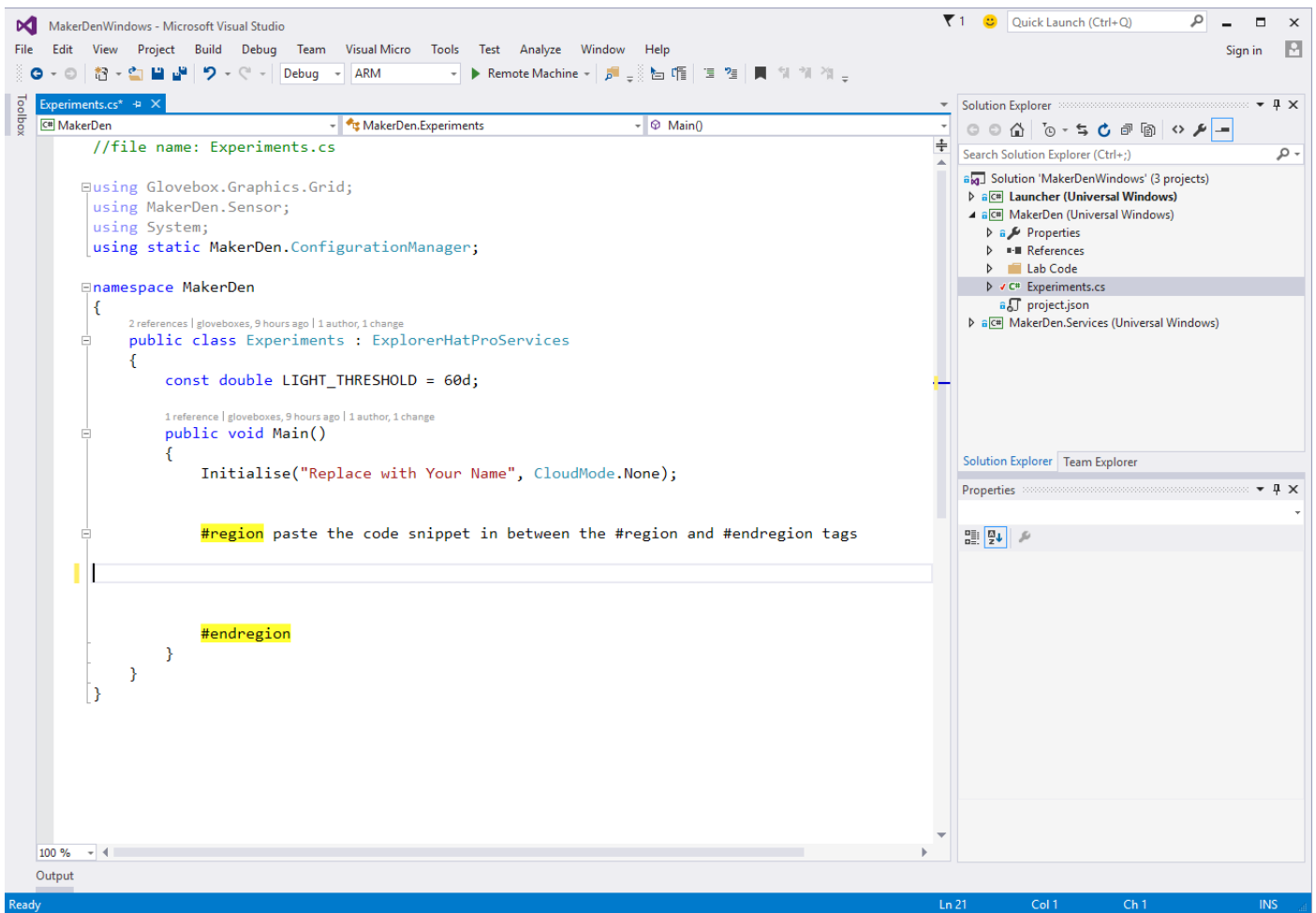
This lab reads the current light levels from the light sensor.

✕ **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 and jump to step 8 OR follow the **Copy and Paste** instruction below.

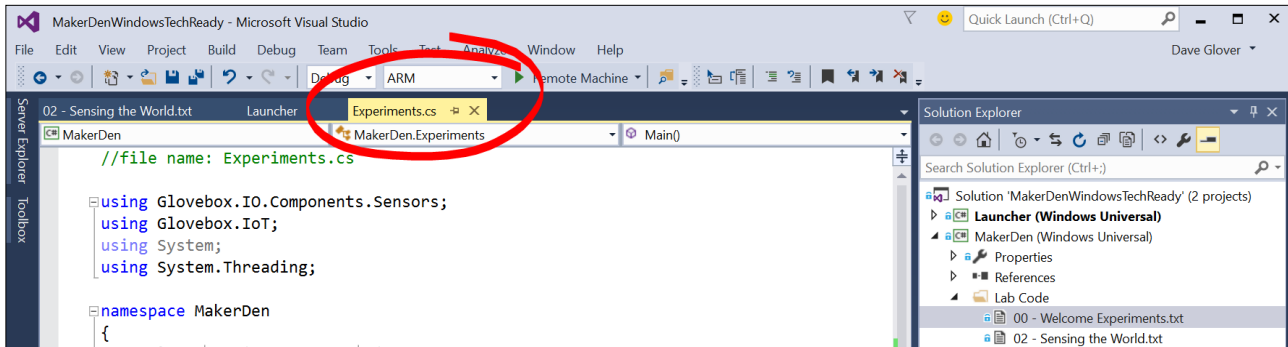
```
using (SensorMgr lightSensorMgr = new SensorMgr(light))
{
    // keep looping until the plug is pulled - this is a very common IoT pattern
    while (true)
    {
        // this is simply so we can break and see the value
        var level = light.ReadRatio * 100;

        // 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:** To copy and paste, open by clicking on the “02 - Sensing the World.txt” file from the “Lab Code” folder.

✕ **STEP 5:** Type **Ctrl+A** **Ctrl+C** to Select All and Copy all the code in this file.

✕ **STEP 6:** Reselect “Experiments.cs” from the page tabs.



✕ **STEP 7:** Place the cursor between the #region tags and type **Ctrl+V** to paste the code you copied from the “02 - Sensing the World.txt” file.

✖ **STEP 8:** Your “Experiments.cs” file should 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.IoT_Hub);

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

✖ **STEP 10:** Hover your hand over the light sensor and observe that the face changes and the green LED turns off when it gets dark.

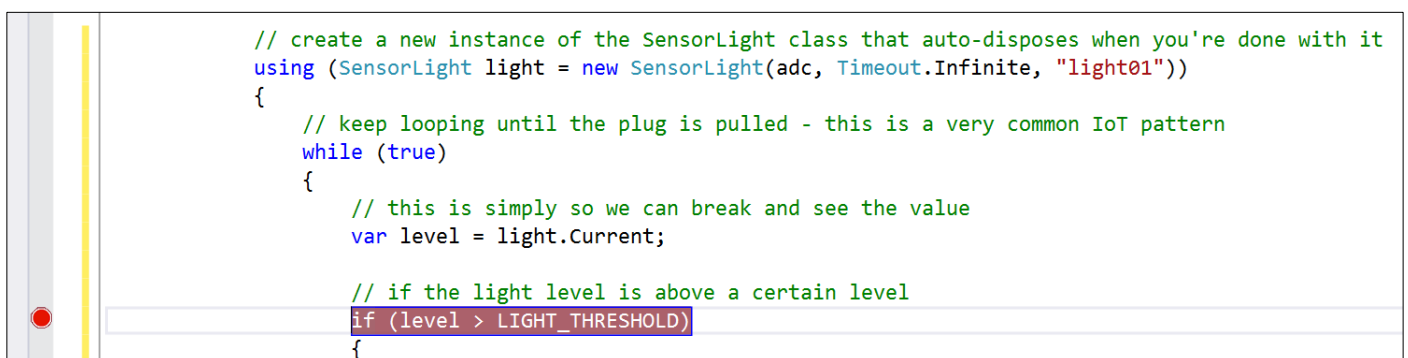
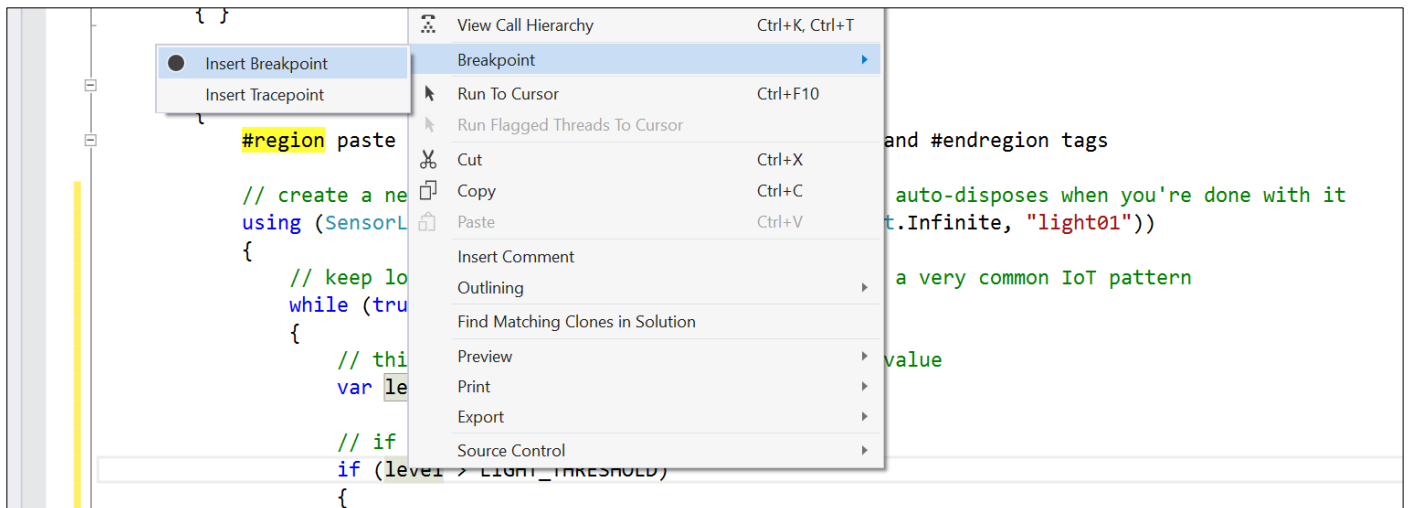


## EXPERIMENT 2A (OPTIONAL): REMOTE DEBUGGING

- ✕ **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 says

```
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 and jump to step 7 **OR** follow the **Copy and Paste** instruction below.

```
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:** To copy and paste, open by clicking on the “03 -Sensing light and Temp.txt” file from the “Lab Code” folder.

✖ **STEP 4:** Type **Ctrl+A Ctrl+C** to Select All and Copy all the code in this file.

✖ **STEP 5:** Reselect “Experiments.cs” from the page tabs.

✖ **STEP 6:** Place the cursor between the #region tags and type **Ctrl+V** to paste the code you copied from the “03 -Sensing light and Temp.txt” file.

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

✖ **STEP 9:** Hold your hand over the light sensor now observe when it gets dark the green LED turns off.

---

## EXPERIMENT 4: CONNECTING TO AZURE

This lab connects the Raspberry Pi 2 to Azure Event Hubs.

When you have successfully completed this lab your data will be visible on the Connect The Dots Maker Den Dashboard at <http://makerden.azurewebsites.net>.

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

✕ **STEP 2:** Type the following code between the #region tags and jump to step 7 **OR** follow the **Copy and Paste** instruction below.

```
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:** To copy and paste, open by clicking on the “**04 - Connecting to Azure Service Bus.txt**” file from the “Lab Code” folder.

✕ **STEP 4:** Type **Ctrl+A Ctrl+C** to Select All and Copy all the code in this file.

✕ **STEP 5:** Reselect “Experiments.cs” from the page tabs.

✕ **STEP 6:** Place the cursor between the #region tags and type **Ctrl+V** to paste the code you copied from the “04 - Connecting to Azure Service Bus.txt” file.

✕ **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:** From the desktop browser navigate to <http://makerden.azurewebsites.net> to view your data. You will see your data plus data from other Raspberry Pis that are streaming data to Azure Event Hubs.



✕ **STEP 9:** You can filter the data displayed by clicking on the sensors listed on the left hand side of the chart.

---

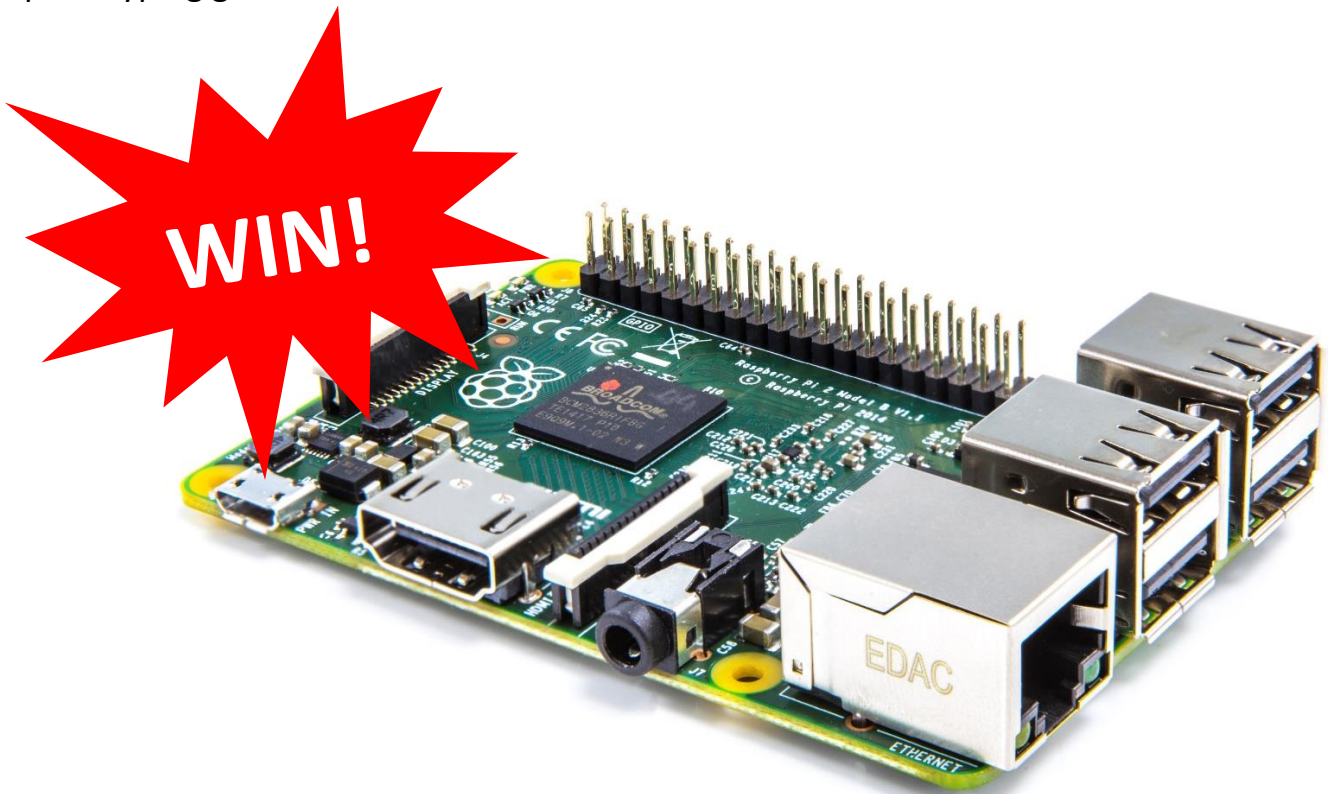
## EXPERIMENT 5: NOT SO FAST COWBOY

Congratulations, you've successfully completed the Maker Den Experience. You've done some hardware prototyping, deployed a Universal Windows App to a Raspberry Pi 2, streamed data to Microsoft Azure Event Hub, and visualised aggregate sensor data with Stream Analytics via Azure Websites.

Please complete the following steps before you leave.

☒ **STEP 2:** Take 30 seconds to complete the Maker Den Evaluation at <http://aka.ms/TR21MakerdenEval>

☒ **STEP 3:** Make sure you get scanned to be in the draw for a Raspberry Pi 2 plus prototyping gear.



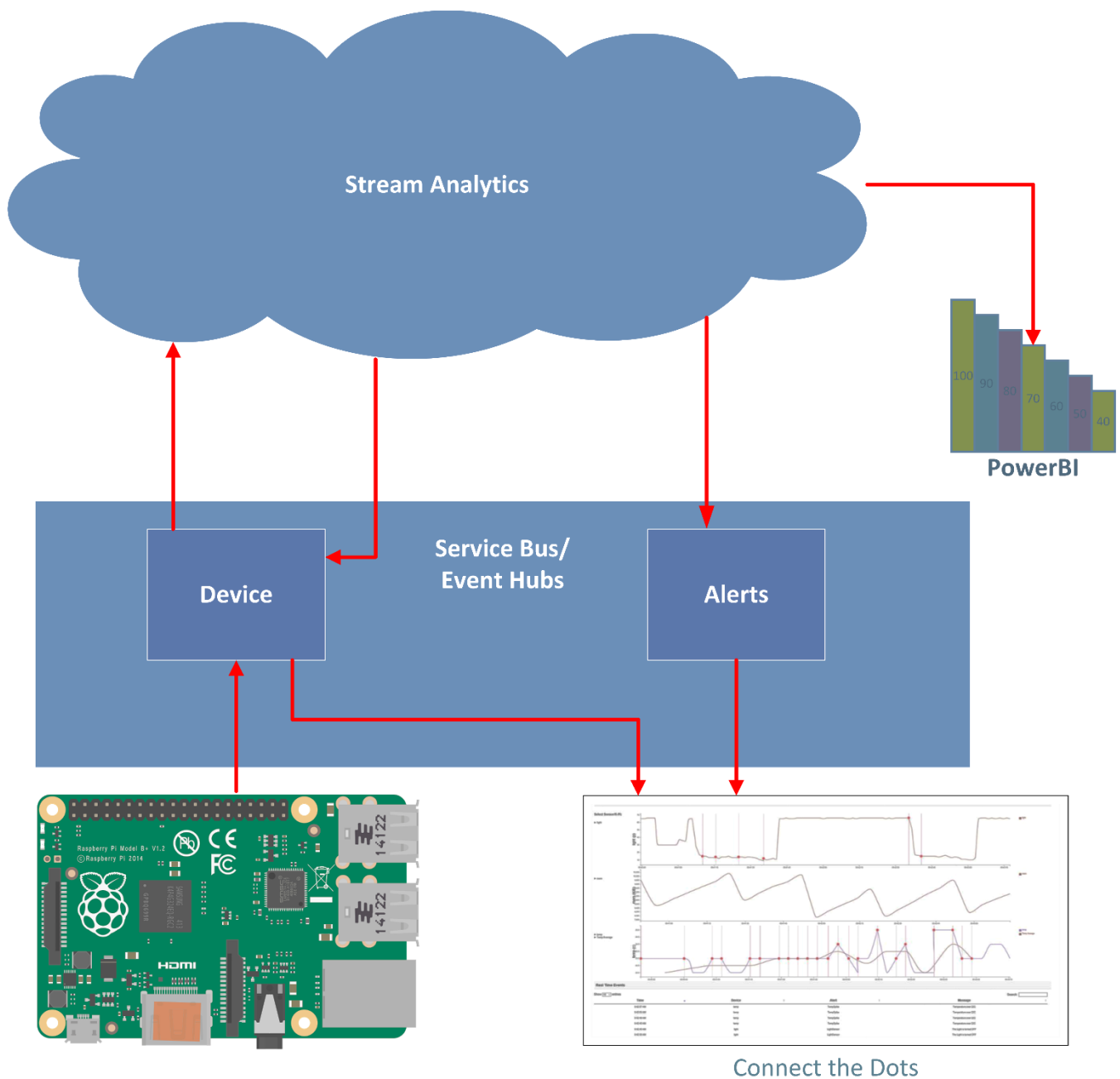
All the documentation and software for the Maker Den is available at <http://www.github.com/makerden>

## ARCHITECTURAL OVERVIEW

The Maker Den architecture can be viewed at two levels.

At a high level, devices stream data to the Azure “Devices” Event Hub. In turn, Stream Analytics consolidates and summarises the data back to the Devices Event Hub for visualisation purposes and generates alerts based on predefined thresholds.

Both the Devices and Alerts queues push data to the Connect the Dots service (using SignalR) for display on the dashboard.



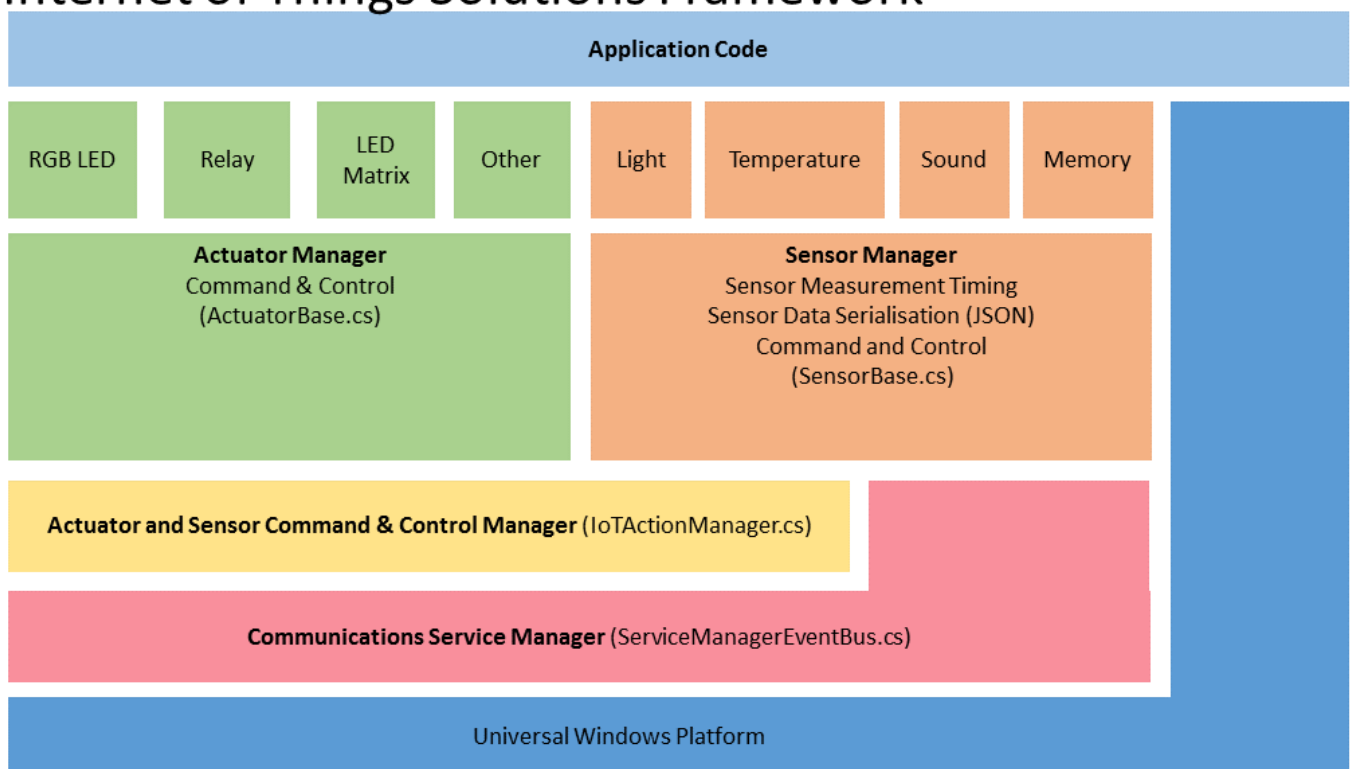


At a lower level, the Maker Den framework sits on top of the Universal Windows Platform. There's a communications layer and a Actuator and Sensor command and control base class on top of which actuator and sensor base classes are built.

On top of the actuator and sensor base classes, the framework implements concrete classes for specific actuators (things that do something like move a servo or flash a light) and sensors (things that measure and report readings like temperature, light level, sound, pH, and so on).

Finally, the Maker Den application code takes advantage of this framework to interact with the physical world.

## Internet of Things Solutions Framework



---

### BLOGS

- [Windows 10 IoT Core](#)
- [Microsoft Hackster.IO](#)
- [Embedded 101](#)