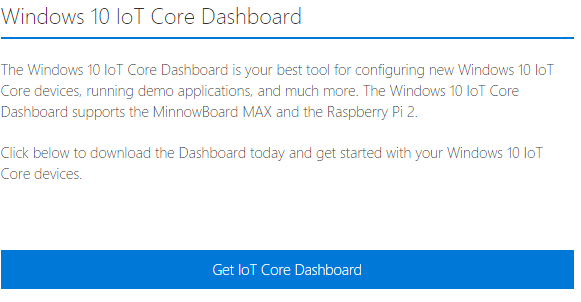
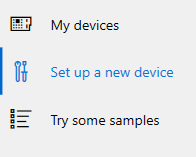
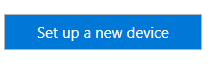
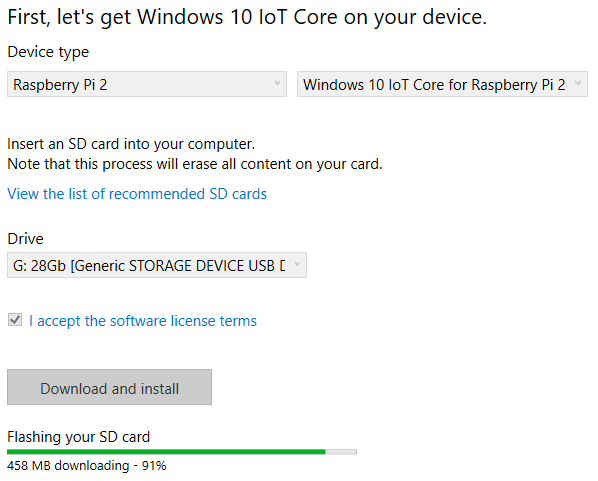
Microsoft IoT Hackathon DX

Raspberry Pi

# Configuring Windows 10 IoT Core

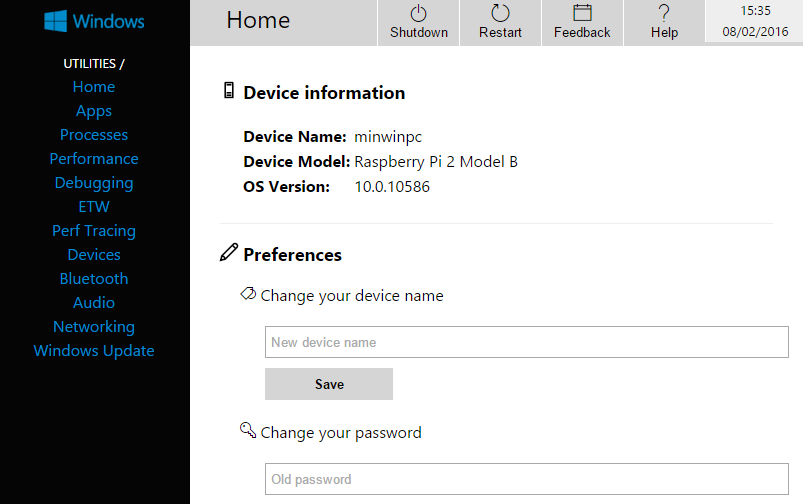
The easiest way to install Windows 10 IoT Core on your Raspberry Pi is by installing the Microsoft IoT Core Dashboard. This tutorial will talk you through using the dashboard to install Windows 10 IoT Core.

1. Go to <http://ms-iot.github.io/content/en-US/Downloads.htm> and click the **Get IoT Core Dashboard**.  
   
2. Download the file to your computer and **install** it.
3. Once installed the program will automatically open. If you already have it installed, you can access it from the Windows 10 IoT Core folder on your start menu.
4. Click the **“Set up a new device”** option on the left hand side of the application. 
5. Click the **“Set up a new device”** button.   
   
6. Set **“Device Type”** to Raspberry Pi 2, the second drop down should be auto populated.   
   ***Note:*** *if the device type doesn’t show any options wait a few seconds and they should appear.*
7. **Insert a Micro SD** card into your computer.   
   ***Note:*** *While the majority of SD cards are compatible with Windows 10 IoT Core some are not. It’s worth checking the list of officially supported SD cards if you have any issues with the one you have.* [*http://ms-iot.github.io/content/en-US/win10/SupportedInterfaces.htm*](http://ms-iot.github.io/content/en-US/win10/SupportedInterfaces.htm)
8. If the drive for the SD card isn’t already selected, then **select the correct drive**.
9. **Accept** the terms and conditions.
10. Finally click the **“Download and install”** button. Depending on the speed of your internet this may take a while to complete.  
    ***Note****:* Your SD card will erased as part of this process so back-up any files that you want to keep on the SD card.  
    
11. Once completed you need to eject the SD card from your computer and place the SD card in the Micro SD card port of your Raspberry Pi.
12. Connect your Raspberry Pi to a monitor, keyboard and mouse and a power supply and Windows 10 IoT Core will boot up. It may take a few minutes for windows to load.
13. If you are using **Ethernet** connect your Ethernet cable to the Pi. If using a **USB Wi-Fi** adapter, make sure it is on the list of approved Wi-Fi adapters. <http://ms-iot.github.io/content/en-US/win10/SupportedInterfaces.htm#WiFi-Dongles>.
14. Once windows 10 IoT Core has loaded confirm your **language** and click **“Next”**.
15. **WIFI TODO**

# Connecting to Windows 10 IoT Core from your PC

1. In the IoT Dashboard click the **“My Devices”** link on the left hand side.
2. You will now see all your Windows 10 IoT Core devices listed. Click the globe icon in the **“Open Device in Portal”** column.
3. You will now be asked to login. Login using user name **“Administrator”** and password **“p@ssw0rd”**.

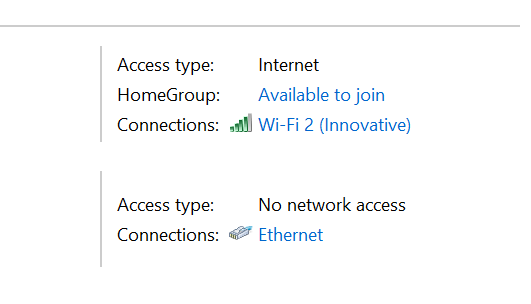
From the Device Portal you can manage many different aspects of the device including passwords, installed apps, processes and hardware. You can also monitor device performance and as well as shutting down and restarting the Pi.



If you are going to be setting up more than one Rasperry Pi with Windows 10 IoT Core, then I would recommend you change the Pi’s name. You can do this from the **“Home”** screen in the portal.

# Using a direct Ethernet connection with your Pi and PC

You can connect a Pi directly to a computers Ethernet port using a network cable and share your computers wireless connection with your Raspberry Pi. This is handy if you need to use your Raspberry Pi without a monitor and aren’t able to plug it into an Ethernet port on the network. At the very least this will allow you to setup a wireless connection.

1. Using a standard Ethernet cable plug one end into your Raspberry Pi and the other end into your computers Ethernet port. The Raspberry Pi and most modern computers Ethernet ports now automatically switch the signal when a direct connection is detected. Using an Ethernet cross over cable might not work.
2. Using Windows 10 launch the network and sharing centre. You can either do this from the control panel or by typing **“Network”** into the Cortana search box.
3. The network centre will show the network connections you have available. Select the Wi-Fi connection by clicking on the name of the network you are connected to.  
   
4. This will bring up the details for the connection. Next click on the **“Properties”** button.
5. You should see two tabs – Networking and Sharing. Select the **“Sharing”** tab.
6. Tick the “**Allow other network users to connect through this computer’s Internet connection**” box.
7. Click “**Ok**” and close the windows to exit the network and sharing centre.
8. Your Raspberry Pi will have a different IP address to the one assigned to it earlier. Close and re-open the Windows 10 IoT Core Dashboard and the new IP address for the Raspberry Pi should be shown. If it’s not shown restart the Rasbperry Pi.

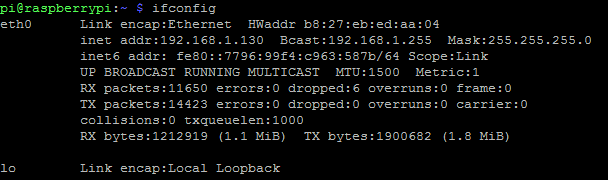
# Configuring Raspbian

When it comes to installing an OS on the Raspberry Pi there are several options. By far the most popular option is Raspbian. You can install this on your Pi by downloading the OS to an SD card but you can also download Noobs. Noobs is an easy to use OS installer for the Pi and as well as including Raspbian it also includes a range of other OS’s that can be installed.

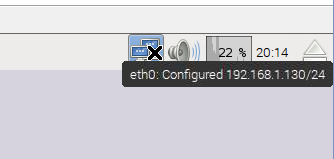
1. Download Noobs from <https://www.raspberrypi.org/downloads/noobs/>
2. Insert a Micro SD card into your computer and format the card. The card should be at least 8GB in size. <https://www.sdcard.org/downloads/formatter_4/> have an excellent easy to use utility for formatting SD cards which is both Windows and Mac compatible.
3. Extract the downloaded Noobs files to your Micro SD card.
4. Once the files have been extracted to your Micro SD card eject the card from your computer.
5. Place the Micro SD card into the Raspberry Pi, connect a monitor, keyboard and mouse and either a USB Wi-Fi dongle or an Ethernet cable and connect the power.
6. Once powered up a window will be shown with a list of different OS’s you can install. Select Raspbian by ticking the box next to it and click the **“Install”** button.
7. Raspbian will now begin to install. This can take a while to complete.
8. Once installed you will be able to login to your Pi using the username **“Pi”** and password **“raspberry”**. When typing your password, no characters will be shown. This is a security feature.

# Obtaining the IP address

If you plan on connecting to your Raspberry Pi using SSH or VNC you can use either the Pi’s network name (defaults to RaspberryPi) or the Raspberry Pi’s IP address. To obtain the Raspberry Pi’s IP address do the following: -

1. On the command line type **“ifconfig”**. To get the IP address
2. The IP address is then shown as “inet address”. Make sure you look at the correct adapter.  
   

If you are on the Raspbian desktop you can simply put the mouse over the network icon on the top right hand corner of the desktop which will show the IP addresses for the different network adapters.

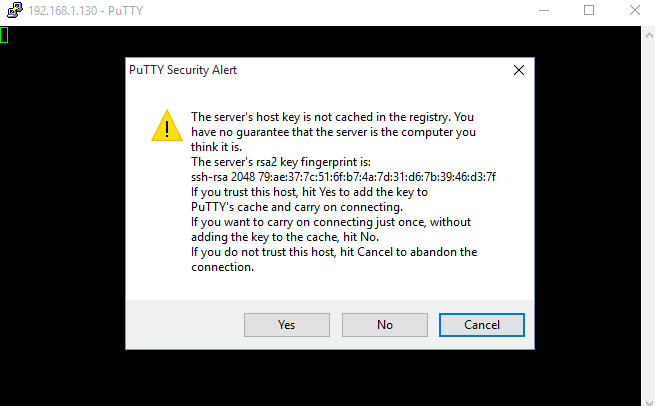


# Loading the desktop mode and changing boot mode

If your Raspberry Pi boots into the console mode, you can load the desktop by typing **“startx”**. To configure the start-up mode of your Raspberry Pi, do the following: -

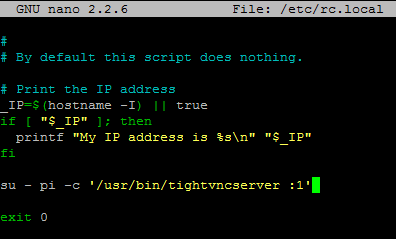
1. On the console enter **“sudo raspi-config”**.
2. Go to option **“3 Boot Options”**.
3. Select the **boot option** you want and press enter.
4. Press **“Esc key”** to exit.

# Connecting to your Pi with SSH

1. To connect to your Pi from another machine using SSH you will require and SSH client. Open **“PuTTY”** (or your own choice of SSH tool) on your computer. If you do not have PuTTy already installed it can be downloaded from <http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html>
2. Enter the address of your RaspberryPi in the host name and make sure the port is set to 22.
3. Click the **“open”** button.
4. PuTTY will now connect to your Pi. You may receive a PuTTY Security Alert. If you do press **“Yes”** to continue.   
   
5. Once connected you will be prompted for the username and password. The username is the same as the username and password you used to login directly on the Pi. If you have changed these then the SSH login details will have also changed.
6. 
7. You are now logged in and can type any commands in the SSH client that you can type directly on the Pi’s command line.

# Setting up your Pi for remote desktop

Sometimes you will want to connect to your Pi remotely and get the full desktop experience. To do this we can install VNC (Virtual Network Computing) software. One of the most popular VNC programs for Raspbian is TightVNC.

1. Open an SSH session and type **“sudo apt-get update”** on the command line.
2. When this has finished type “sudo apt-get tightvncserver”. You will be prompted to set a password for TightVNC. Enter a password.
3. Next you will be asked if you want to set a view only password for TightVNC. This isn’t required so for now type **“N”**.
4. Once TightVNC has finished installing type **“sudo nano /etc/rc.local”**. This opens up the SSH text editor. We will modify the rc.local file so that TightVNC server starts automatically when the Pi boots up.
5. Above “exit 0” enter the following text **“su – pi -c ‘/usr/bin/tightvncserver :1’”**.  
   
6. Press **“Ctrl + O”** to save the changes (write out).
7. Press **“Ctrl + X”** to exit.
8. Type **“sudo reboot”**.
9. While the Raspberry Pi reboots you will need to download a VNC client onto your computer. Go to <http://www.tightvnc.com/download.php> and download TightVNC.
10. Install the downloaded file. If you don’t want to install TightVNC server on your computer, select the “Custom” install option and deselect the TightVNC server option.
11. Once installed open TightVNC on your computer and type the IP address of the Raspberry Pi into the remote host textbox followed by **“:1”.** E.g. 192.168.1.130:1.
12. Click the **“Connect”** button.
13. You will now be prompted to enter the password. This is the password you set when installing TightVNC on your Pi. Upon entering the correct password, you will now see the Raspbian desktop in the viewer.

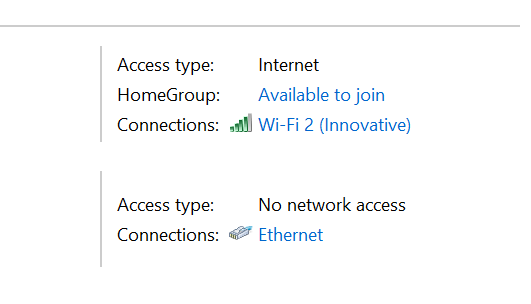
# Setting up your Pi with a static IP address

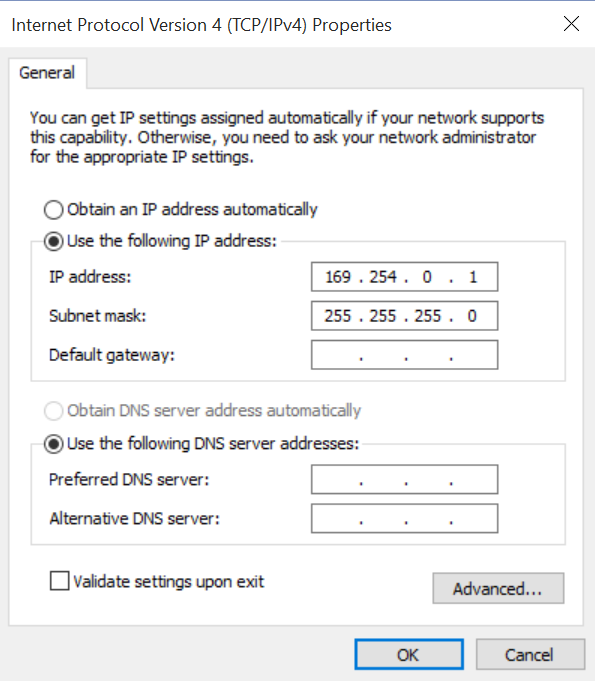
If you want to assign a static IP address or use a direct network connection to your Raspberry Pi and won’t have a display and keyboard and mouse available to you it pays to do some basic configuration work first. There a several ways of setting a static IP address for your Raspberry Pi. Below I’ve detailed the simplest method I’ve found.

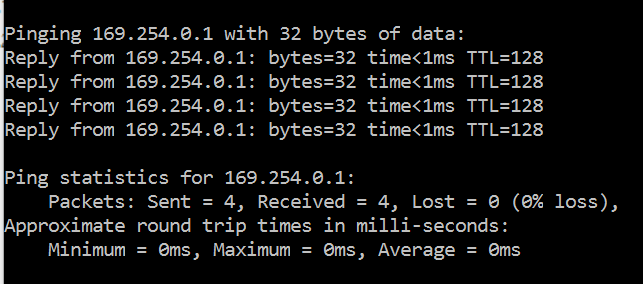
1. Open an SSH session or directly on the Raspberry Pi command line type “**sudo cp /boot/cmdline.txt /boot/cmdline.normal”**. This copies your cmdline.txt file so you can easily restore the file if required.
2. Next type **“sudo nano /boot/cmdline.txt”** to open the nano editor. At the end of the long line shown in nano type **“ip=<IP address>”** where <IP address> is the IP address you want to assign to your Raspberry Pi, for example 169.254.0.2. Make sure you have a space before “ip”.
3. Press **“Ctrl + O”** to save changes (write out).
4. Press **“Ctrl + X”** to exit.
5. Copy this file as well by typing **“sudo cp /boot/cmdline.txt /boot/cmdline.direct”.**
6. Type “sudo reboot” to reboot your Raspberry Pi with your new IP address.
7. To switch between IP addresses type **“sudo cp /boot/cmdline.direct /boot/cmdline.txt”** or **“sudo cp /boot/cmdline.normal /boot/cmdline.txt”.**
8. Using a standard Ethernet cable plug one end into your Raspberry Pi and the other end into your computers Ethernet port. The Raspberry Pi and most modern computers Ethernet ports now automatically switch the signal when a direct connection is detected. Using an Ethernet cross over cable might not work.
9. Next follow the **“Configuring your Windows 10 PC for a direct network connection”** tutorial.

# Configuring your Windows 10 PC for direct network connection

You might need to make changes to your Windows network settings in order to make sure that your Raspberry Pi and PC are on the same network range.

1. To configure your network settings, launch the network and sharing centre. You can either do this from the control panel or by typing **“Network”** into the Cortana search box.
2. The network centre will show the network connections you have available. Select the Ethernet connection by clicking on the **“Ethernet”** text.
3. This will bring up the details for the connection. Next click on the **“Properties”** button.
4. Select **“Internet Protocol Version 4 (TCP/IPv4)”** in the list and click the **“Properties”** button.
5. Select “Use the following IP address:” option and type in the IP address you want to use. In this example I’m using **“169.254.0.1”**. Also in the subnet mask box if it is empty type **“255.255.255.0”**.

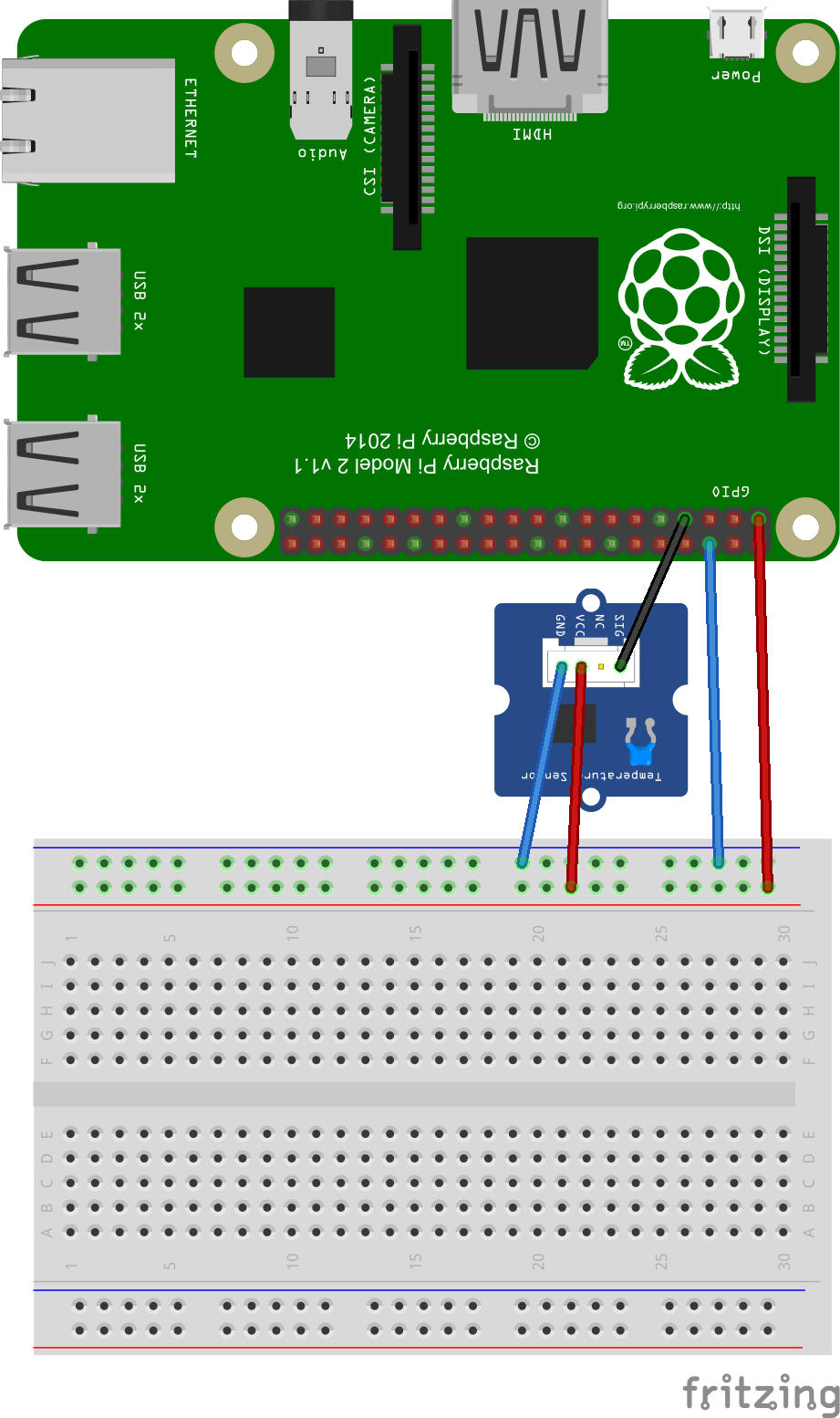


1. Click the **“Ok”** button, and close the remaining windows you opened.
2. Your Raspberry Pi and Windows PC should now be using the same network range. To test this, open a command prompt by typing into the Cortana search box **“cmd”**.
3. On the command prompt type “**ping <IP address>**” where the <IP address> is the address you assigned to your Raspberry Pi. You should see something similar to the screen shot below: -  
   

You are now ready to establish a SSH or Remote desktop session with your Raspberry Pi.

# Your first program

Using a DHT22 temperature and humidity sensor wire up the Raspberry Pi as per the diagram.

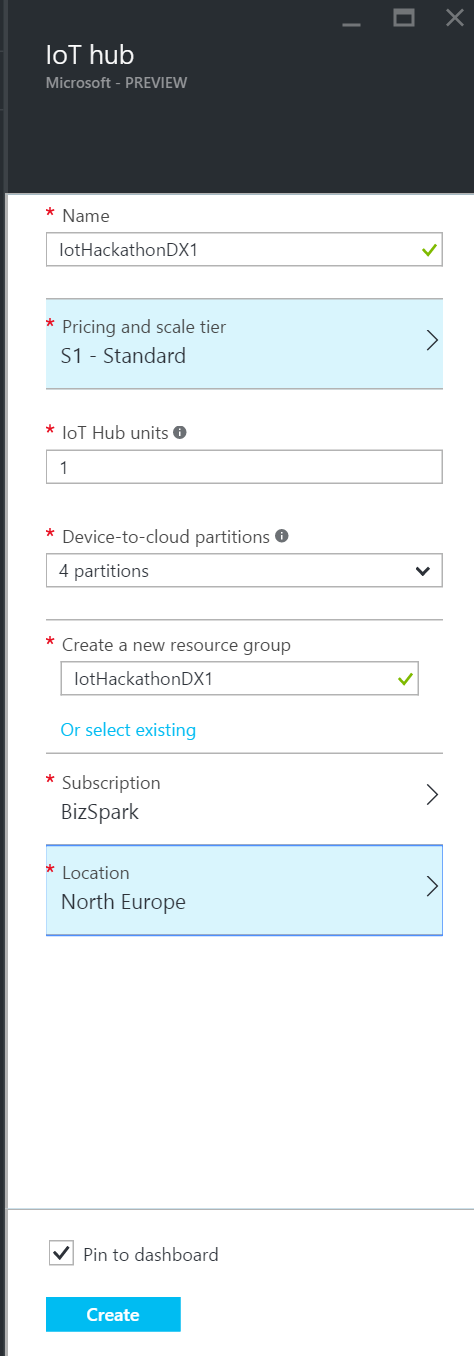


# Login to Azure

Azure has two management portals – the classic portal (<http://manage.windowsazure.com>) and a new portal that is in preview at <http://portal.azure.com>. You will be using the new portal for this challenge.

1. **Open a browser** and go to [**http://portal.azure.com**](http://portal.azure.com)
2. Enter your **Microsoft Account email address** and **password** for the Microsoft Account associated with your subscription.
3. You will now be in your Azure subscription (see opposite) and from here you can create and manage Azure services.

# Provision the IoT Hub

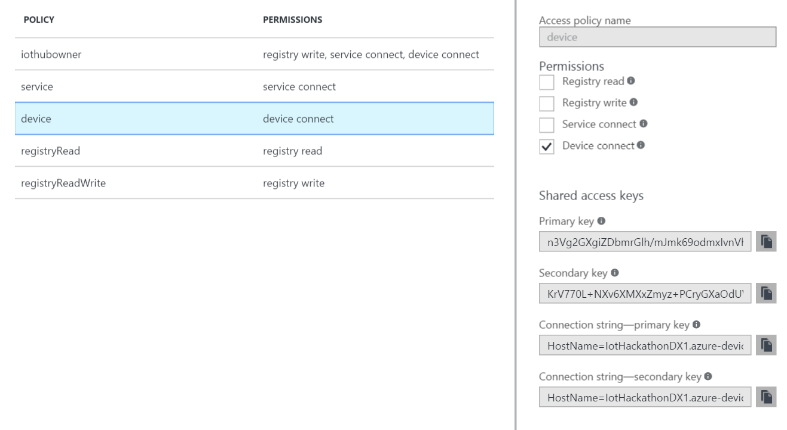
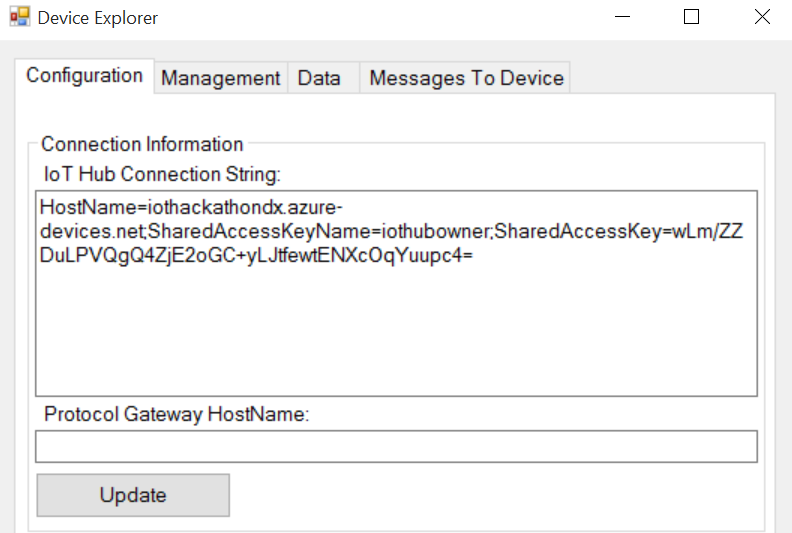
1. On the Preview Portal (remember you need to be in the Preview Portal), click **“+ NEW”** (top left) and select **Internet of Things.**
2. Select **IoT Hub** from the featured apps shown.
3. Enter a name for your hub. In this example I am using “IoTHackathonDX1”.
4. The IoT Hub is charged based on the number of messages per day you want to send to the hub. There is a **“free”** tier in this service that gives you 8,000 messages/day at no charge – select this option unless you are already using the **“free”** tier, in which case select “**S1 – Standard**”.

**Each Azure subscription may only have one free tier IoT Hub provisioned**

1. Leave the IoT Hub units as 1 and the number of partitions as 4.
2. Create a new resource group by clicking **“or create new”** and enter the name you want to call this resource group. We will re-use this resource group for provisioning additional services later on in this lab.
3. For the location select a location local to you where possible.
4. Click the **create button**. You will now be taken back to your Azure dashboard where you will see a new dashboard tile appear showing the status of the provision. IoT Hubs can take a few minutes to provision.

# Using Device Explorer

You can’t configure your IoT Hub devices from the Preview Portal however as part of the Azure IoT SDK there is an application called DeviceExplorer which you can use to manage devices, view messages being sent to your hub and send cloud to device messages. For this part of the lab you will need to have the Azure IoT SDK.

1. **Download** or clone the GitHub repo for the **Azure IoT SDK** (<https://github.com/Azure/azure-iot-sdks>). If downloading extract, the zip files to your machine.
2. Navigate to \tools\DeviceExplorer\ in your local repo and **open the DeviceExplorer solution** file in Visual Studio.
3. Run the project.
4. To use Device Explorer, the first thing you need to do is enter the connection string for your IoT Hub. In the Preview Portal click your IoT Hub tile that was created for your earlier.
5. Two panes should now appear for your IoT Hub. The right-hand side pane should be titled **“settings”**
6. Select the **“Shared access policies”** menu option and select the **“iothubowner”** policy in the shared access policies tile.
7. From the **“iothubowner”** tile copy the **“Connection string – primary key”.**
8. Back in Device Explorer paste into the **“IoT Hub connection string”** textbox in Device Explorer.
9. Click the **“Update”** button and a dialog confirming the update of the settings should appear.
10. Select the **management tab** and click the **“Create”** button.
11. Enter a name for your device in the **“device id”** textbox. In this example I’ve called my device “raspberrypidevice”, copy the primary key and click **“create”**. You should now see the device you created shown in the data grid. You can verify it was created by clicking the “refresh” button.

# Windows 10 IoT Core App

1. Open Visual Studio 2015 and go to **File -> New -> Project**. From the Templates select **Windows -> Universal -> Blank app (Universal Windows)**. Select a location to save the project to and click the **“Ok”** button.
2. Download the source code for DHT22 from [https://github.com/porrey/dht/tree/master/source/Windows%2010%20IoT%20Core/DHT%20Solution/Sensors.Dht and add the Sensors.DHT](https://github.com/porrey/dht/tree/master/source/Windows%2010%20IoT%20Core/DHT%20Solution/Sensors.Dht%20and%20add%20the%20Sensors.DHT) project to your solution.
3. Add a reference to the Sensors.DHT project to the project you just created.
4. Next add a reference to your project for “Windows 10 IoT Extensions for UWP”. To do this right click the references node in the solution explorer and select add. From the left hand tree select **“Universal Windows -> Extensions”** and then select **“Windows 10 IoT Extensions for UWP”** and click the **“Ok”** button.
5. Next using **Nuget** **add** the **“Microsoft.Azure.Devices.Client”** package to your project.
6. At the top of your MainPage.xaml.cs file add the following imports: -  
   using Sensors.Dht;

using Microsoft.Azure.Devices.Client;  
using Windows.Devices.Gpio;  
using System.Text;

1. Next below the class declaration add a constant string to store your IoT Hub connection string in: -  
   private const string IOTHUBCONNECTIONSTRING = "HostName=<IoTHubName>.azure-devices.net;DeviceId=<DeviceID>;SharedAccessKey=<SharedAccessKey>";  
   Replace the <IoTHubName> with the name of the hub you created, replace <DeviceId> with the name of the device you created and replace <SharedAccesKey> with the shared access key you copied earlier from device manager.
2. Below the IoT Hub Connection string add the following constants and variables: -  
   private const int DHTPIN = 4;

private IDht dht = null;

private GpioPin dhtPin = null;  
private DispatcherTimer sensorTimer = new DispatcherTimer();

1. In the main page constructor add the following code: -  
   dhtPin = GpioController.GetDefault().OpenPin(DHTPIN, GpioSharingMode.Exclusive);

dht = new Dht11(dhtPin, GpioPinDriveMode.Input);

sensorTimer.Interval = TimeSpan.FromSeconds(10);

sensorTimer.Tick += sensorTimer\_Tick;

sensorTimer.Start();

1. In the class add the following methods: -  
   private void sensorTimer\_Tick(object sender, object e)

{

readSensor();

}

private async void readSensor()

{

DhtReading reading = await dht.GetReadingAsync().AsTask();

if (reading.IsValid)

{

// Send reading to IoT Hub

string message = "{\"temperature\":" + reading.Temperature.ToString() + ", \"humidity\":" + reading.Humidity.ToString() + "}";

Message eventMessage = new Message(Encoding.UTF8.GetBytes(message));

DeviceClient deviceClient = DeviceClient.CreateFromConnectionString(IOTHUBCONNECTIONSTRING);

await deviceClient.SendEventAsync(eventMessage);

}

}

1. Next you need to deploy the solution to your Raspberry Pi. In Visual Studio from the tool bar set the solution platform to **“ARM”** and the target as **“Remote Machine”**.



1. Open the project properties by double clicking the **“Properties”** node in solution explorer and navigate to the **“debug”** settings.
2. In the **“Remote machine:”** textbox you need to enter the IP Address/ Name of your Raspberry Pi. To do this click the “Find” button, wait a few moments for Visual Studio to search for your devices and your Rasbperry Pi should show in the “Auto Detected” section. Click your Raspberry Pi and click the **“Select”** button. If your device doesn’t show your device should be called **“minwinpc”** so enter this value.
3. Press the **“green play”** button next “Remote Machine” or press **F5** to build and deploy the solution to your Raspberry Pi. Your device should now start sending temperature and humidity readings to the IoT Hub.