Proiect MPSIT 2016

Raspberry Pi 3 Windows IoT Core

Nastase Liviu SSA

https://github.com/liviunst/mpsit-raspberrypi

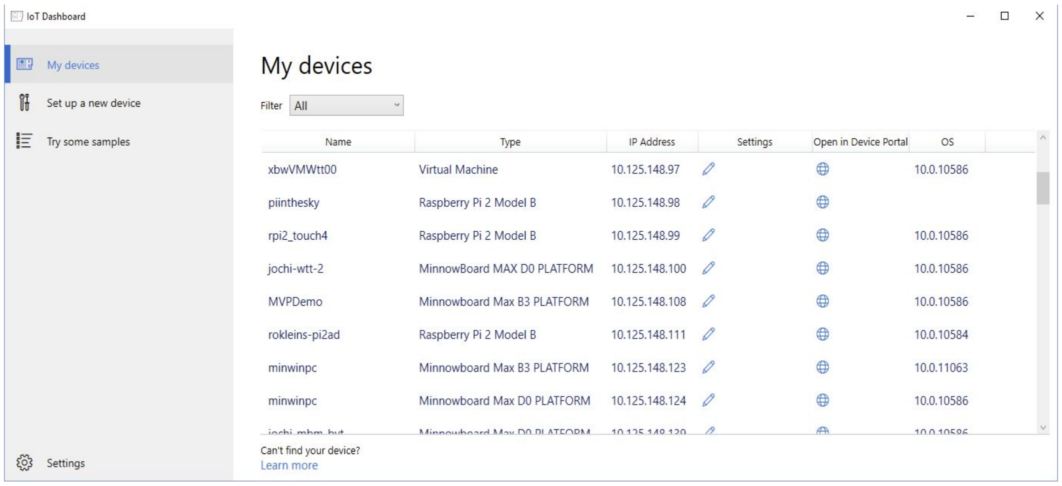
Overview:

- Windows IoT Core
- Developing solutions for Raspberry Pi 3
- Temperature Sensor
- WebCam
- Project architecture
- Future features
- Usecases

Windows IoT Core

Windows 10 IoT Core is a version of Windows 10 that is optimized for smaller devices with or without a display, and that runs on the Raspberry Pi 2 and 3. Windows 10 IoT Core utilizes the rich, extensible Universal Windows Platform (UWP) API for building solutions.

The fastest way to install Windows 10 IoT Core on a device is through Windows IoT Core Dashboard. The IoT Dashboard will be used through the development of the solution for interaction with the device. You will need a microSD card, then from the dashboard select your device type, choose a name and the Windows 10 IoT Core will be flashed to the microSD. You can also set up a wi-fi network to which the device will connect after boot.

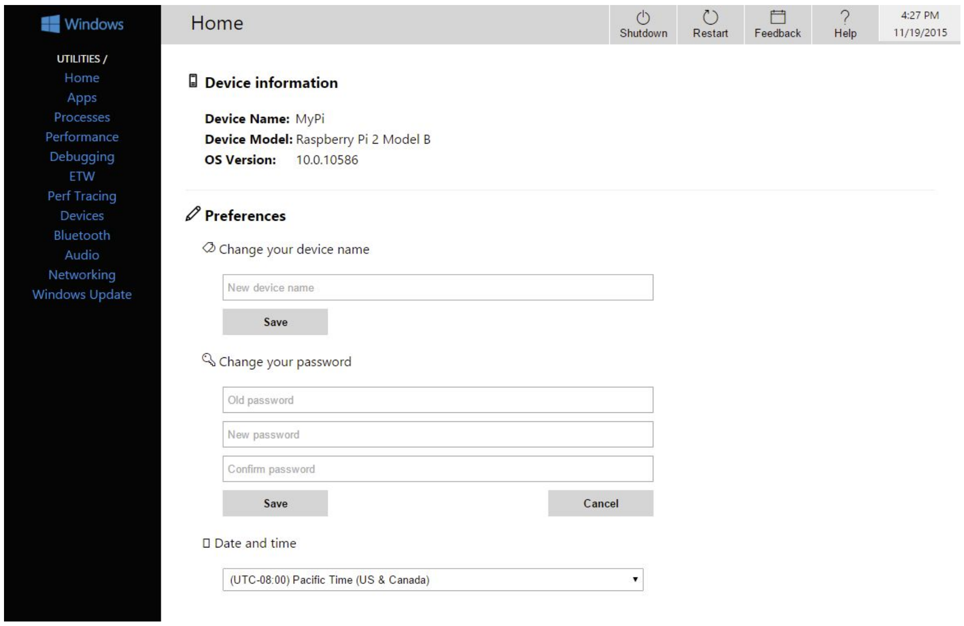


My devices

Filter: All

Name	Type	IP Address	Settings	Open in Device Portal	OS
xbwVMWtt00	Virtual Machine	10.125.148.97	✎	🌐	10.0.10586
piinthesky	Raspberry Pi 2 Model B	10.125.148.98	✎	🌐	
rpi2_touch4	Raspberry Pi 2 Model B	10.125.148.99	✎	🌐	10.0.10586
jochi-wtt-2	MinnowBoard MAX D0 PLATFORM	10.125.148.100	✎	🌐	10.0.10586
MVPDemo	Minnowboard Max B3 PLATFORM	10.125.148.108	✎	🌐	10.0.10586
rokleins-pi2ad	Raspberry Pi 2 Model B	10.125.148.111	✎	🌐	10.0.10584
minwinpc	Minnowboard Max B3 PLATFORM	10.125.148.123	✎	🌐	10.0.11063
minwinpc	Minnowboard Max D0 PLATFORM	10.125.148.124	✎	🌐	10.0.10586

Can't find your device?
Learn more



Home

4:27 PM
11/19/2015

Device information

Device Name: MyPi
Device Model: Raspberry Pi 2 Model B
OS Version: 10.0.10586

Preferences

Change your device name

New device name

Save

Change your password

Old password

New password

Confirm password

Save | Cancel

Date and time

(UTC-08:00) Pacific Time (US & Canada)

Developing solutions for Raspberry Pi 3

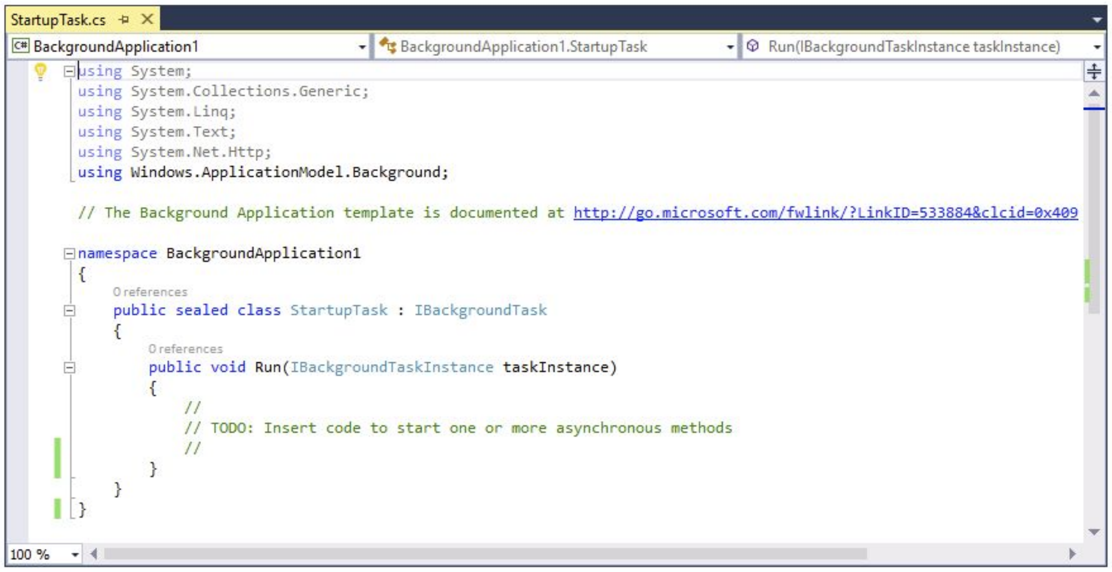
Developers who want to create solution for Windows 10 IoT Core can choose from several language as C# / C++ / Python / JavaScript.

There are 2 types of solutions that can be created:

- Headed
- Headless

- Headed solutions are UWP applications that can display an UI for the user, they can be set as default application to start on the device

- Headless solutions are applications based on BackgroundTask template that can be run in the background and set to start when the device booted

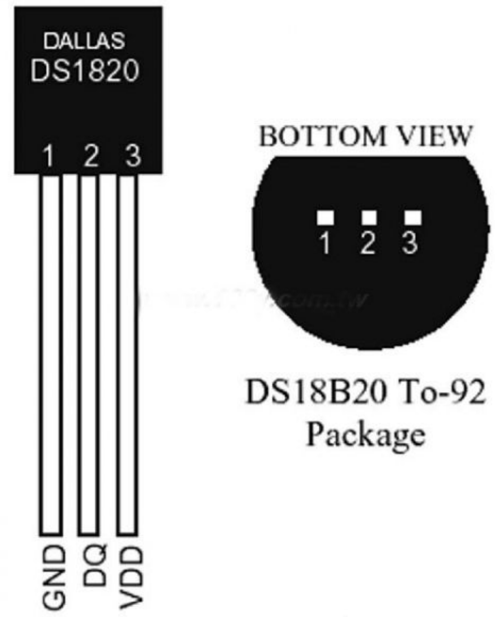

StartupTask.cs
BackgroundApplication1
BackgroundApplication1.StartupTask
Run(IBackgroundTaskInstance taskInstance)
100 %

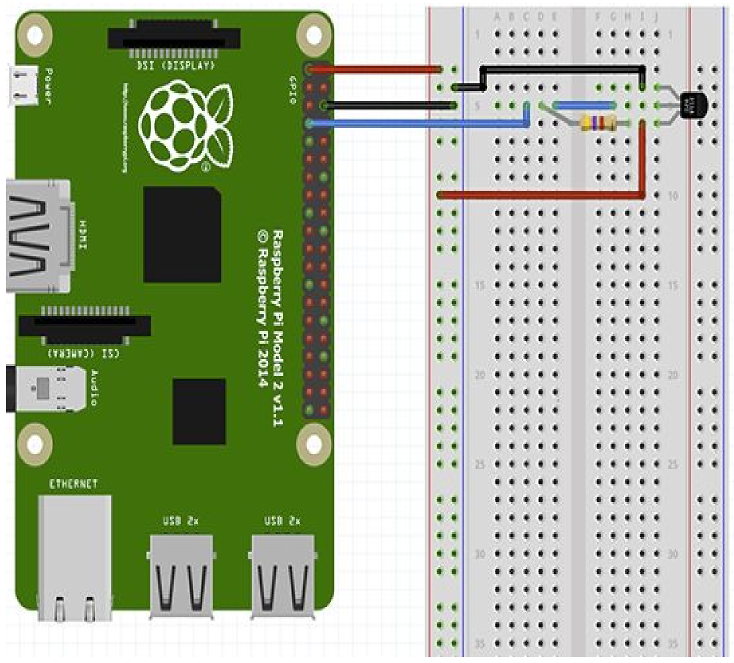
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Net.Http;
using Windows.ApplicationModel.Background;

// The Background Application template is documented at http://go.microsoft.com/fwlink/?LinkID=533884&clcid=0x409

namespace BackgroundApplication1
{
    0 references
    public sealed class StartupTask : IBackgroundTask
    {
        0 references
        public void Run(IBackgroundTaskInstance taskInstance)
        {
            //
            // TODO: Insert code to start one or more asynchronous methods
            //
        }
    }
}
```

Both types of solution can be developed in Visual Studio, deployment to the device is also done through the IDE and solutions that run on the device can be debugged.

Temperature sensor

The DS18B20 digital thermometer provides 9-bit to 12-bit Celsius temperature measurements. The DS18B20 communicates over a 1-Wire bus that by definition requires only one data line (and ground) for communication with a central microprocessor. In addition, the DS18B20 can derive power directly from the data line, eliminating the need for an external power supply.

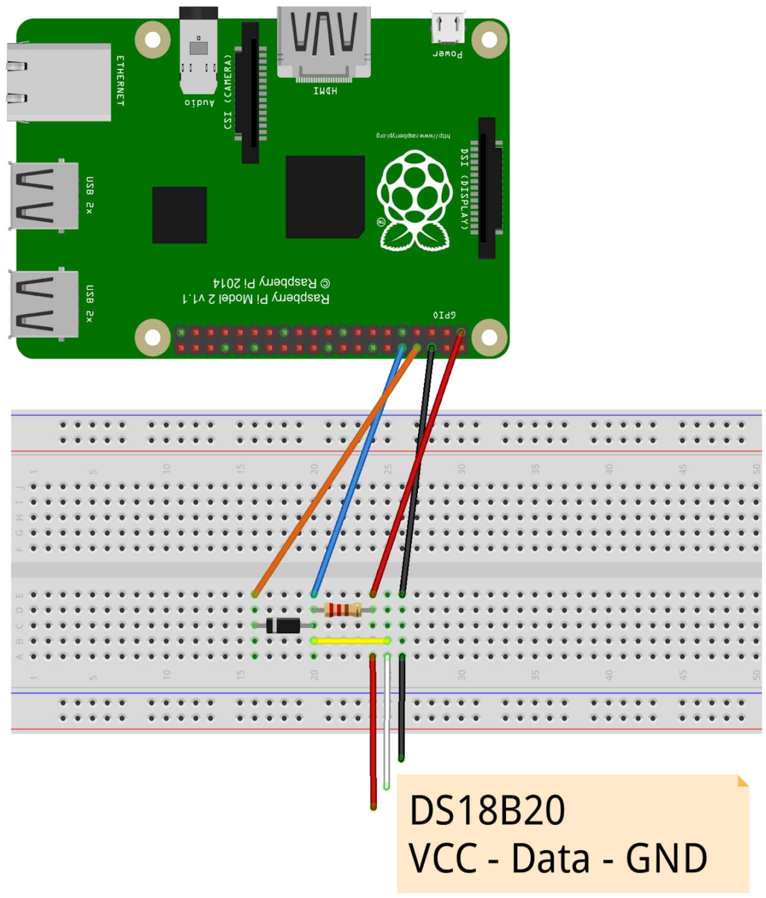




This is the schematics for the connection of DS18B20 to the Raspberry 3 using 1-wire connection.

Resistance - 4,7 k

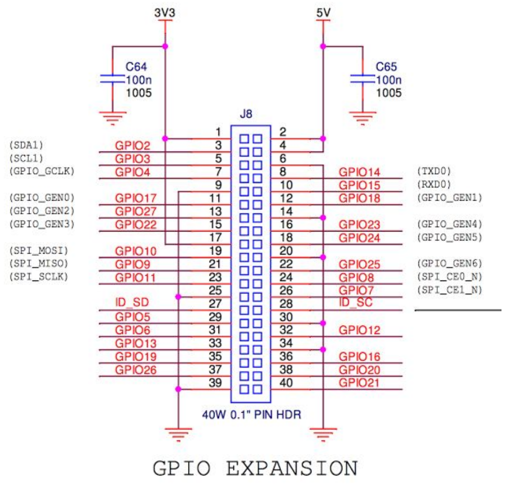
Unfortunately the Windows IoT Core does not support 1 wire communication, so we must use serial communication. In UWP apps it is required to declare the capability for serial communication



This is the schematics for the connection of DS18B20 to the Raspberry 3 using serial connection.

Resistance 4.7k
Used - 5.5k

1N5819 – 1A Schottky Barrier Rectifier

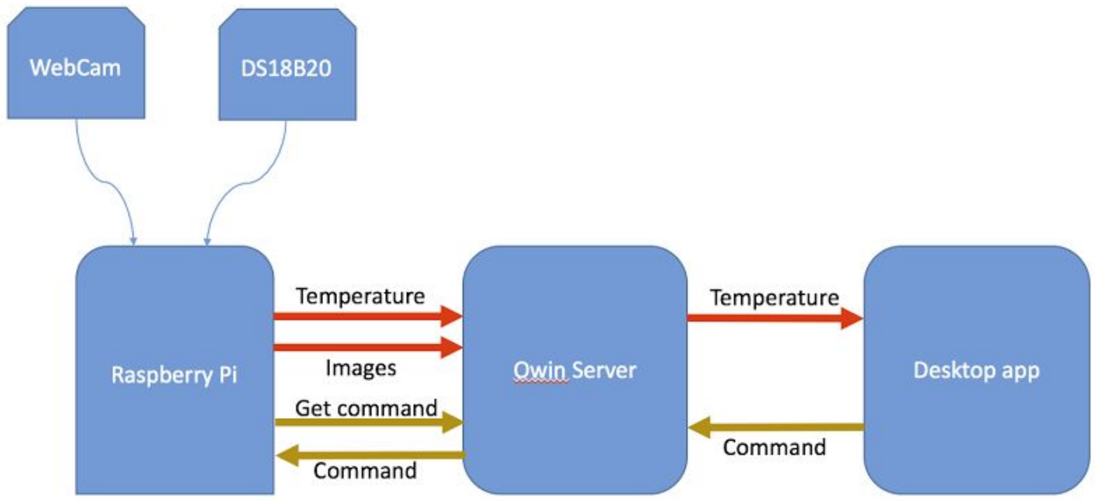


Webcam

Connecting an usb webcam to Raspberry Pi is straightforward in Windows IoT Core, it offers the MediaCapture class which can be used to initialize the webcam and get the data stream from it.

Applications that require to use the webcam need to declare the capabilities in appmanifest to have access to the webcam.

Project architecture



- Raspberry Pi
    - Webcam
    - Temperature sensor
- Owin Server
- Desktop app

All three components were written in C#.

On the raspberry pi an UWP application is installed, it can connect to the owin server and send temperature data and images from the webcam. The application also queries the server for new commands.

The Owin server is implemented using Owin self host webApi, it exposes a REST api through which the Raspberry pi can upload data and the Desktop app can read data. The desktop app can also upload commands for the Raspberry.

The desktop app is also an UWP application.

Future features

- Display images captured by the webcam in the desktop app, they are stored in the server
- Stream webcam data directly to a client, desktop app
- Add functionality for scheduled commands

Usecases

- Remote surveillance and monitoring (interior / exterior, servers / warehouses / old people & childs)