# A brief introduction in setting up a Raspberry Pi in a Windows IoT- Visual Studio environment

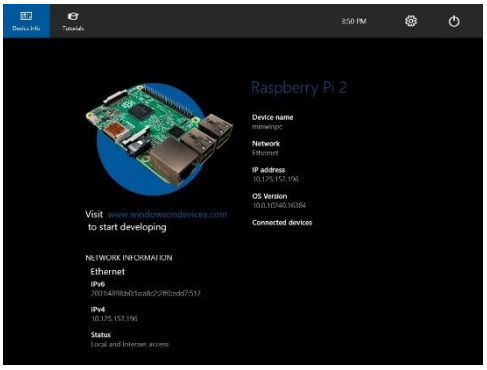
## 0 0 Prologue

The purpose of this document is to give information to the team which may or may not be helpful and reduce the time to get started. All explanations and definitions are to be read in the context of the AMOS Project “Sivantos Raspberry Pi (Team 1, Project 2”, thus, there is no pretensions to give a detailed and complete overview in the sections. For further information, please do not hesitate to use www.google.de for instance.

## 1 Windows IoT

### 1.1 What is it?

It is an operation system for devices with very limited resources, which is why it will be installed on the Raspberry Pi.



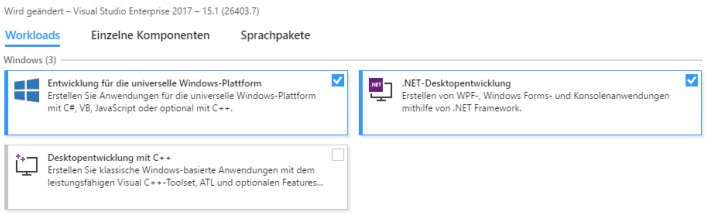

### 1.2 Features

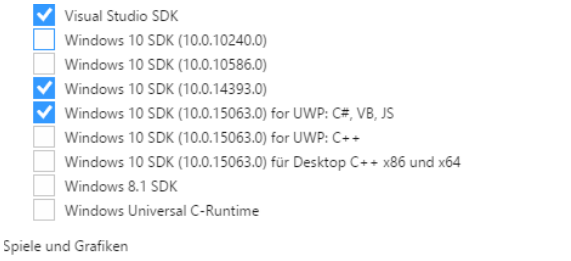
- The OS has a GUI where you can access various Hardware Features like Wi-Fi or read some HW-Information
- It also has a “Command Prompt”

# 2 Visual Studio

## 2.1 Installing

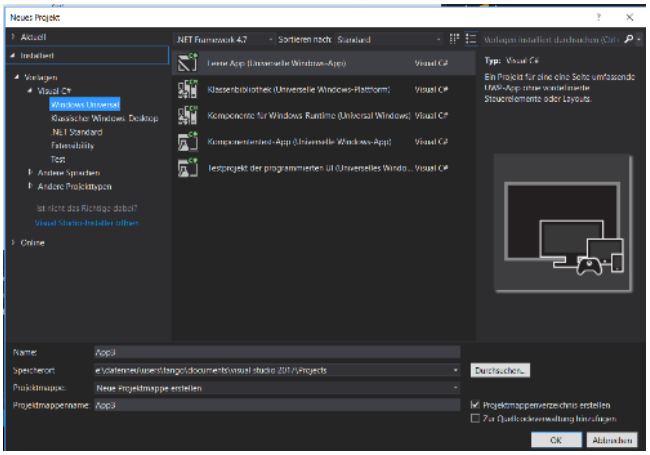
You can download Visual Studio Professional or Enterprise for free from our university through Microsoft Imagine, just log in with your student credentials. I use the Enterprise version where you can configure your version to your needs like complete Workloads or even single features like Windows 10 SDKs.
You should install the SDK of Version 14396 (10.0.14393.0) additionally since your Windows 10 Version will probably be lower as version 15063 (In my case it was not an official release candidate yet).




## 2.2 Creating a new UWP Project

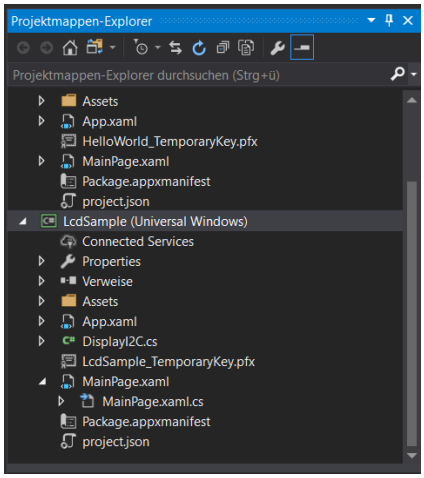
In VS you can make your life easy if you start your project with a given Template. There you have different options e.g. on which platform you wish to develop. In the case of the Sample-Demo the industry partner used a UWP (Universal Windows Platform) – Project. So let us take a look on the project folder structure of the Demo by looking at the Project Explorer on the very right hand side of the IDE.




## 2.3 Project Explorer

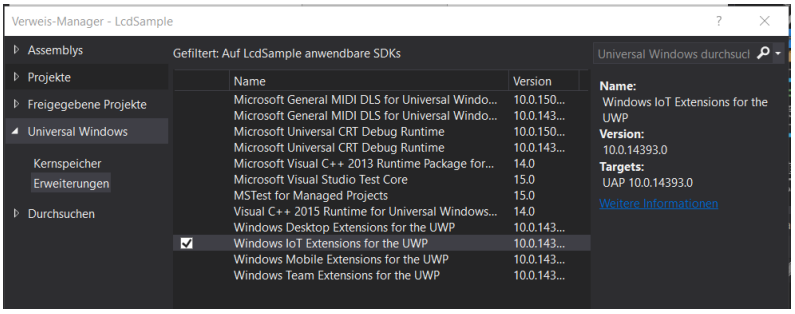
### 2.3.1 Code Files

The code-relevant-files have the ending ".cs" for c#. The Main-Program-Code is a **Code-Behind-File** ".xaml.cs" files which is behind the "MainPage.xaml" file.



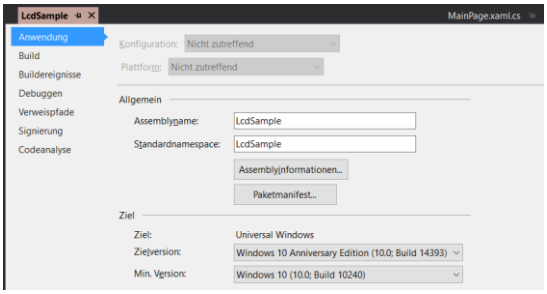

### 2.3.2 References (Verweise ("library's"))

The cool stuff like library's can be added by right-clicking "References (Verweise)". There you should make sure you install the "Windows IoT Extensions for the UWP" in the right version. How to determine the right version will discussed in the next section "Properties".



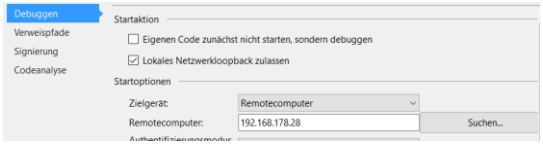
### 2.3.3 Properties

#### *2.3.3.1 Version*




The target version of your application should be compatible with the version which is installed on the Raspberry Pi. It can be lower but not higher. The Pi configured by me has version 15063 installed. If you can choice a version you wish to install, then you should install the proper **Windows 10 SDK** through the **Visual Studio Installer.**

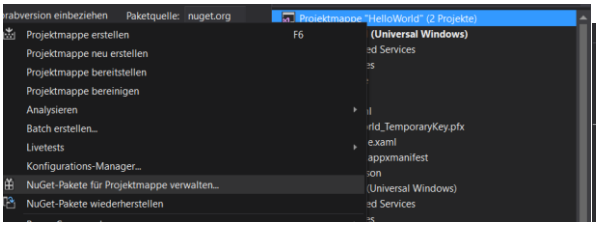
#### *2.3.3.2 Debugging*

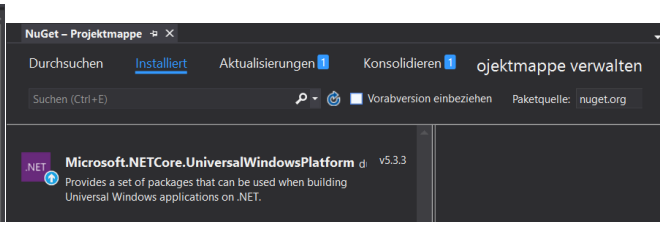



In this submenu, you can set up the communication between Visual Studio and the Raspberry Pi by entering the IP-Adress of it in the textBlock “Remotecomputer”. You could enter the network-name of the Pi but I ran into issues then.

### 2.3.4 .NuGet-Packages

The very cool stuff you find in the right-click-context-menu of your project-folder. If you open the “NuGet-Package-Manager” then it will open a menu where you at first see all your installed Packages. In the submenu “search”(Durchsuchen) you have access to online available packages.




I also have researched a bit and found the following project relevant top-list, which might be also a start for further research:

- http://nugetmusthaves.com/Tag/Raspberry

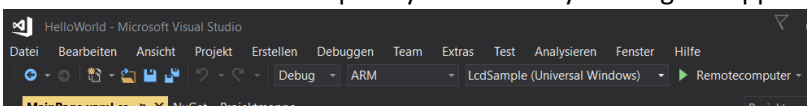
### 2.3.5 Helpful Extensions (Tipps and Tricks)

If you have worked with Eclipse you will probably miss some features like One-Click-Auto-Formating the code or Strg-Clicking on a method and jumping to the method implementation. So I did a bit of research and came up with the following nice extension which focus actually on keeping your hands on the keyboard instead on the mouse thus it has some very nice features:

- Productivity Power Tools 2017

## 2.4 Visual Studio and Raspberry Pi

If you considered the information on section 3.3 than the communication between the Raspberry PI and Visual Studio is very simple. In your upper “Menu-Bar” you choice the target platform (ARM) and the project you like to deploy. Afterwards, you just click on “Remotecomputer” which first “build” the program then transfers the files to the Raspberry Pi and finally starting the application on the pi.




If you don’t have the Pi connected to a monitor, then you won’t see anything special but the bar on the bottom of your IDE will change from blue to orange. Alternatively, you can remotely connect to the Raspberry Pi’s desktop which will be further discussed in the next chapter.


Interesting to know is, after you stop the Remote-Connection the program-files are still on the Pi and can be executed in the Raspberry Pi-Environment without Visual Studio

# 3 Raspberry Pi

## 3.1 Relevant Features

As for the Raspberry Pi 3 it has:

- Wi-Fi
- Bluetooth
- Virtual Keyboard

## 3.2 Setting up the Pi from scratch (Windows IoT-Dashboard)

There is quite good step-by step Tutorial on the Windows Dev Center for installing Win IoT on the Pi with the App "Windows IoT-Dashboard":

- https://developer.microsoft.com/de-de/windows/iot/getstarted

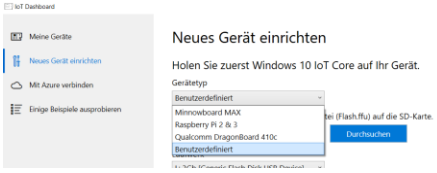
### 3.2.1 Need to knows

- The SD-Card for the Pi and Windows IoT cannot be chosen randomly -not all of them are supported.
- During the setup, you will be asked to set an administrator password, which is "amos" for the Pi I have configurated.

### 3.2.2 Issues with installing Windows OS

I had troubles to install the OS: It downloaded it but during the transfer to the Pi it broke with a failure massage (Failed to unpack the Windows 10 IoT Core installation package.). The solution consists of manually installing the OS:

1. You need the "flash.ffu" file. You can download it from the web or extract it from the file the Dashboard downloaded for you (I dont remember where it was, but it was an executable, which you install and it returns two files – one of it is the "flash.ffu").
2. You select customized on the device drop-down-menu.




3. Fill the elements which are left and install it

## 3.3 How to interact with the Pi

First step is to connect the Raspberry Pi to your LAN by network cable or, as far as it is configured, WLAN

### 3.3.1 Directly

You can connect it through HDMI on a Monitor. Then you should connect at least an USB-Mouse in order to interact with, for instance, in order to set up the WLAN-Adapter.

### 3.3.2 Remotely

#### *3.3.2.1 Windows IoT Dashboard – Windows Device Portal (Browser-App)*

Through the Dashboard you can directly go to the Browser-App called Windows Device Portal. Therefore, you need to go in the submenu "My Devices" and then right-click on the network-connected raspberry pi IP-address.


https://developer.microsoft.com/de-de/windows/iot/docs/deviceportal

#### *3.3.2.2 Desktop-Remote Control*

It is an App which should be simple to use but do not work in my Network-Environment.

Here you can find a Tutorial: *https://developer.microsoft.com/de-de/windows/iot/docs/remotedisplay*

#### *3.3.2.3 Other Methods (PowerShell, FTP, SSH...)*

Look at the Navigation Bar on the left-hand side when you go to the following page:
https://developer.microsoft.com/de-de/windows/iot/docs/powershell#collapseSection3

# 4 You-Tube Videos & other Weblinks

Here are some Videos and Links I found helpful

## 4.1 C# the programming language

- https://www.youtube.com/watch?v=lisiwUZJXqQ ( Learning C# in one Video (1,5 h Overview) )

- Weblink: Die Programmiersprache C# für Java-Entwickler

### 4.1.1 Asynchronous Programming (Threads, Tasks, Lamda-Expressions)

- (part 1) https://www.youtube.com/watch?v=mkkcqQlIpeQ
- (part 2) https://www.youtube.com/watch?v=IONqMWGn9-w

## 4.2 Visual Studio (Tipps & Tricks)

- https://www.youtube.com/watch?v=JhxC-K-Eehg
- https://www.youtube.com/watch?v=YQ4c_Vd6Mmo

## 4.3 Developing an UWP in c# and Visual Studio

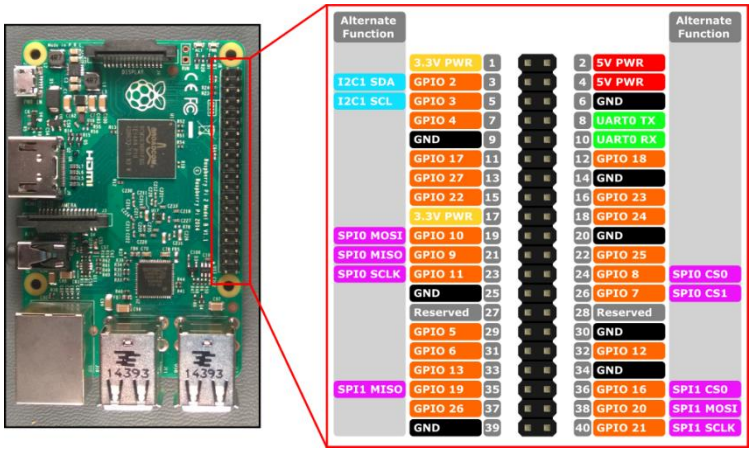
### 4.3.1 Desktop UWP – Calculator

- (part 1) https://www.youtube.com/watch?v=yCRxbnwORIM&t=313s
- (part 2) https://www.youtube.com/watch?v=GQg_niOGIRw

# 5 Programming the Pi in C# and Visual Studio

## 5.1 GPIO

- General-purpose input/output: https://en.wikipedia.org/wiki/General-purpose_input/output




The target GPIO needs to be initialized and programmed to Input / Output before.

(to be continued).....