

Wie schreibt man eine Software für den Raspberry, der mit dem Betriebssystem Windows 10 IoT läuft? Der Raspberry ist ein kleiner PC. Soll er für uns tätig werden, müssen wir das gleiche tun wie bei seinem großen Bruder, wir müssen eine Windows-Anwendung (App) schreiben. Läuft diese auf unserem PC, dann läuft sie unverändert auch auf dem Raspberry, wenn wir uns auf die Hardware des Minicomputers einlassen.

In diesem Arbeitsblatt sollen die Vorarbeiten beschrieben werden, die zur Programmierung einer Anwendung auf dem Raspberry unter Windows einmalig notwendig sind. So muss das Betriebssystem des Raspberry auf eine SD-Karte geschrieben werden und die Programmierumgebung auf dem Laptop installiert werden.

1. Wir erstellen eine Windows 10 IoT Core-Boot-SD.

Als Festplatte verwendet der Raspberry eine SD-Karte. Wir verbinden diese SD-Karte zunächst mit einem Laptop. Auf ihr soll das Windows 10 IoT-System aufgespielt werden. Gleichzeitig soll diese SD-Karte bootfähig gemacht werden. Dazu benötigt der Laptop ein Windows 10 Betriebssystem. Auf der Seite <https://dev.windows.com/de-de/iot> suchen wir *Get started* → *Select hardware* → *Select your installations media* → *Select your OS-version* → *next*. Jetzt wird ein **Dashboard** (früher Core Watcher) geladen und gestartet. Die Software lässt sich auch aus dem Startmenü unter *Windows 10 IoT Core* → *Windows 10 IoT Core Dashboard* aufrufen. In dem Programm wählen wir: *Neues Gerät einrichten*.

Das Laufwerk, als das die SD-Karte erkannt, wird vom Dashboard automatisch gefunden. Ein Name ist für das Raspberry-Board, besser für das SD-Karten-Betriebssystem, einzugeben. Wir verwenden am GMG **rp_{xx}** mit der laufenden Nummer xx. Unter diesem Namen wird die Platine später vom Laptop erkannt. Das später gewünschte Loginpasswort für den Administrator-Zugang wird vorgegeben. Wir benutzen pauschal **gmgrp**. Ist das Betriebssystem heruntergeladen und auf die Karte übertragen, wird die SD Karte vom PC abgemeldet und in den Raspberry eingelegt. Der wird mit Spannung versorgt und fährt hoch.

Wenn das Betriebssystem installiert ist (dauert sehr lange), kann man den Raspberry mit seinem Namen im Dashboard unter *Meine Geräte* finden. Nach dem Rechtsklick auf das gewünschte Gerät klicken wir in dem sich öffnenden Fenster auf *Im Device Portal Öffnen*. Beim ersten Mal erscheint das Login von Windows 10 IoT Core. Es wird nach dem *Gerätenamen* (ist der eben festgelegte Name r_{pxx} für das Board) und dem *Benutzernamen* (ist auf **Administrator** mit großem A festgelegt und lässt sich nicht ändern) gefragt. Mit dem Link des Dashboards *Windows Device Portal im Browser öffnen* lässt sich der Raspberry betrachten (Benutzername *Administrator*, Passwort *gmgrp*).

Wenn wir schon auf dieser Seite <https://dev.windows.com/de-de/iot> sind, ist es ratsam das *Windows 10 IoT Core Creators Update* vorzunehmen. Das Visual Studio 2017, das wir in Schritt 2 einsetzen, verlangt die neueste Version.

Es gibt noch weitere Wege vom Laptop auf den Raspberry zugreifen zu können. Der eine ist der Zugriff über das **DOS-Fenster**. Windows nennt dieses Fenster jetzt Power-Shell. Man erreicht es z.B., wenn man in der Taskleiste auf das Windowsymbol rechtsklickt. Ein weiterer Weg ist die Benutzung des **FTP- (File Transfer Protocol) Servers**, den Windows 10 Core zur Verfügung stellt. Unter <http://winscp.net/eng/docs/lang:de> lässt sich der FTP Client *WinSCP* herunterladen. Mit ihm lässt sich eine Verbindung aufbauen. Weiter stellt Windows 10 auch eine Bedienoberfläche zur Verfügung, die über eine **http-Adresse mit einem Web Browser** aufgerufen wird. Die IP-Adresse entnimmt man dem Desktop des Rasperrys. Wichtig ist es, die Portnummer 8080 anzugeben (z.B. <http://192.168.0.103:8080>).

2. Wir installieren Visual Studio 2017 passend auf unserem Laptop.

Wir benötigen die kostenlose Programmierumgebung *Visual Studio 2017 Community*. Dazu laden wir die Bootstrapperdatei *vs_enterprise.exe* aus der folgenden Adresse https://www.visualstudio.com/de/downloads/?utm_source=mscom&utm_campaign=msdocs und führen sie aus. Wenn wir mehrere PCs mit der Software versehen wollen, ist die Offline-Installation zu empfehlen. Leider bietet Microsoft kein ISO-Image der Software an. Wir helfen uns, indem wir die Setupdatei *vs_enterprise.exe* speichern. In einem DOS-Fenster suchen wir das Verzeichnis der Setupdatei und führen sie mit folgenden Parametern aus: *vs_enterprise.exe --layout c:\vs2017offline*. (*Eingabeaufforderung* oder *Windows PowerShell* aus *Windows + X*) Es werden dann alle zur Installation notwendigen Dateien in das Verzeichnis *c:\vs2017offline* geladen (ca. 30 GB !). Wir führen das Setup benutzerdefiniert durch und aktivieren bei

der Funktionsauswahl die Option *Entwicklertools für universelle Windows-Apps*.

Wir benötigen noch die *IoT Core Projekt Templates für Visual Studio 2017*. Sie stellen in Visual Studio die für uns notwendigen Funktionen zur Verfügung. Wir laden sie aus folgender Seite herunter: <https://visualstudiogallery.msdn.microsoft.com> und führen sie aus.

Das Betriebssystem unseres PCs muss noch auf den *Entwicklermodus* gesetzt werden: *Systemsteuerung* → *Start/Einstellungen* → *Update und Sicherheit* → *Für Entwickler* → *Entwicklermodus*.

Als Literatur wird empfohlen:

Schäffer Florian: *Sensoren am Raspberry Pi 2*, Elektor Verlag, Aachen 2016

Kofler Michael: *Raspberry Pi*, Rheinwerk Verlag, Bonn 2016

Nachtrag: Diese Art der Windows 10 IoT Core Installation macht mit SD-Karten Probleme, die vorher mit einem Linux-Image beschrieben worden sind. Abhilfe schafft es, die SD-Karte mit https://www.sdcard.org/downloads/formatter_4/eula_windows zu formatieren.