

Drehbuch IoT Workshop

Bernhard.Disselhoff@microsoft.com

Version 1 vom 14.03.2018

13.06.2017

Update 30.06.2017



Inhalt

1. Notwendige Hardware
2. Notwendige Software
3. Zusätzlich
4. Zielsetzung
5. Ablauf des IoT Workshops

1. Notwendige Hardware

NodeMcu Lua ESP8266

Amazon

[NodeMcu Lua ESP8266](#)



Breadboard

[Amazon](#)



DHT22

[Amazon](#)



Steckbrücken

[Amazon](#)



Wiedererstand 10K

USB Kabel USB-MicroUSB

2. Notwendige Software

- a. [Arduino Ide](#) 1.8.x oder später



- b. Zusätzliche Treiber einbinden über Einstellungen:

```
http://arduino.esp8266.com/stable/package_esp8266com_index.json
```

- c. Zusätzliche Libs laden über Library Manager

AzureIoTHub

AzureIoTUtility

AzureIoTProtocol_MQTT

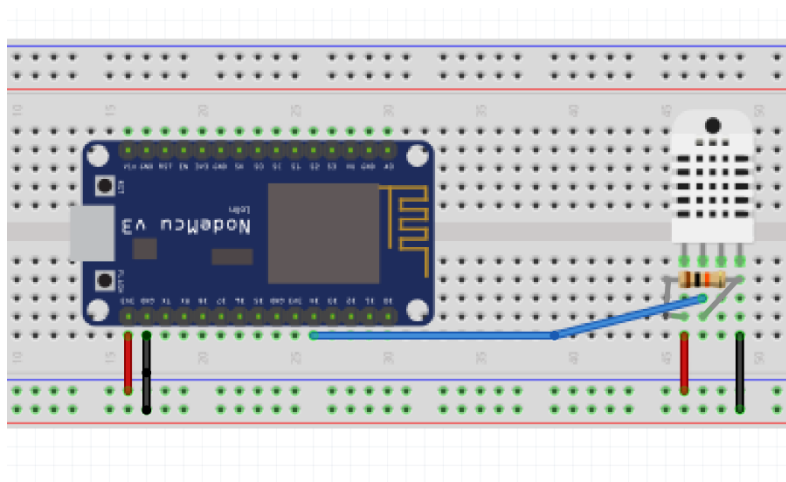
DHT sensor library

Adafruit Unified Sensor

- d. [Examples Code](#)

- e. <https://github.com/Azure-Samples/iot-hub-c-huzzah-getstartedkit>

- f. Aufbau im Workshop



op

3. Zusätzlich

- a. Microsoft Azure Subscription ☺

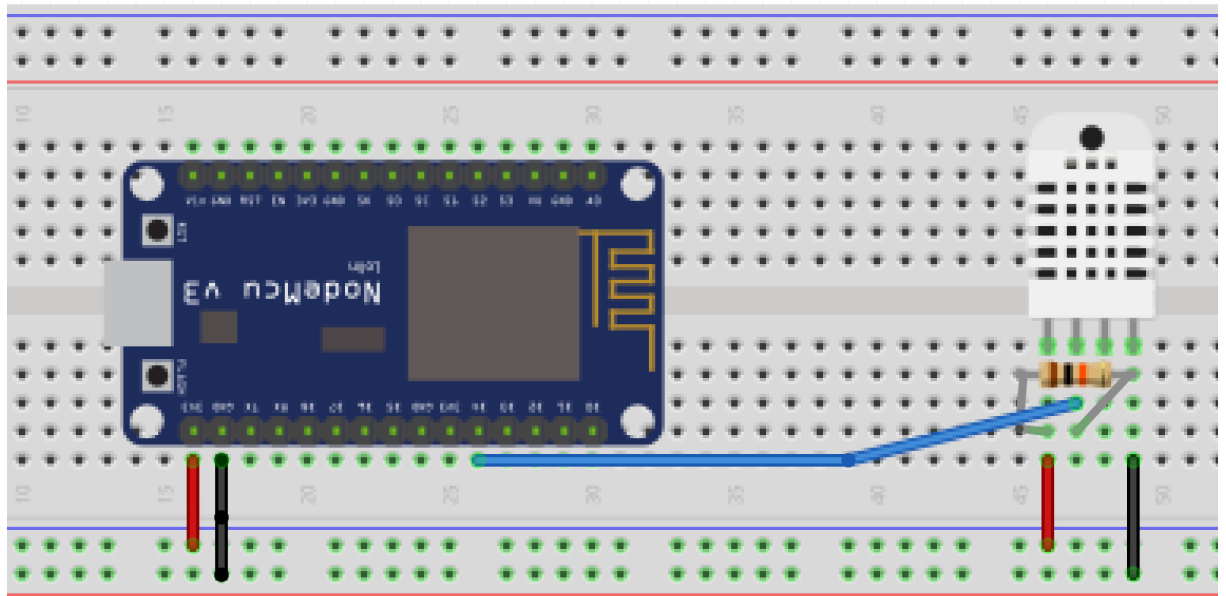
4. Zielsetzung

Zielsetzung des Workshops ist in einer gegebenen Zeit von 2 Stunden einen IoT Device mit Sensor zu bauen, in Betrieb zu nehmen und alle Daten in Microsoft Azure abzulegen. Innerhalb von Azure wird eine Anwendung zur Auswertung und Visualisierung benutzt.

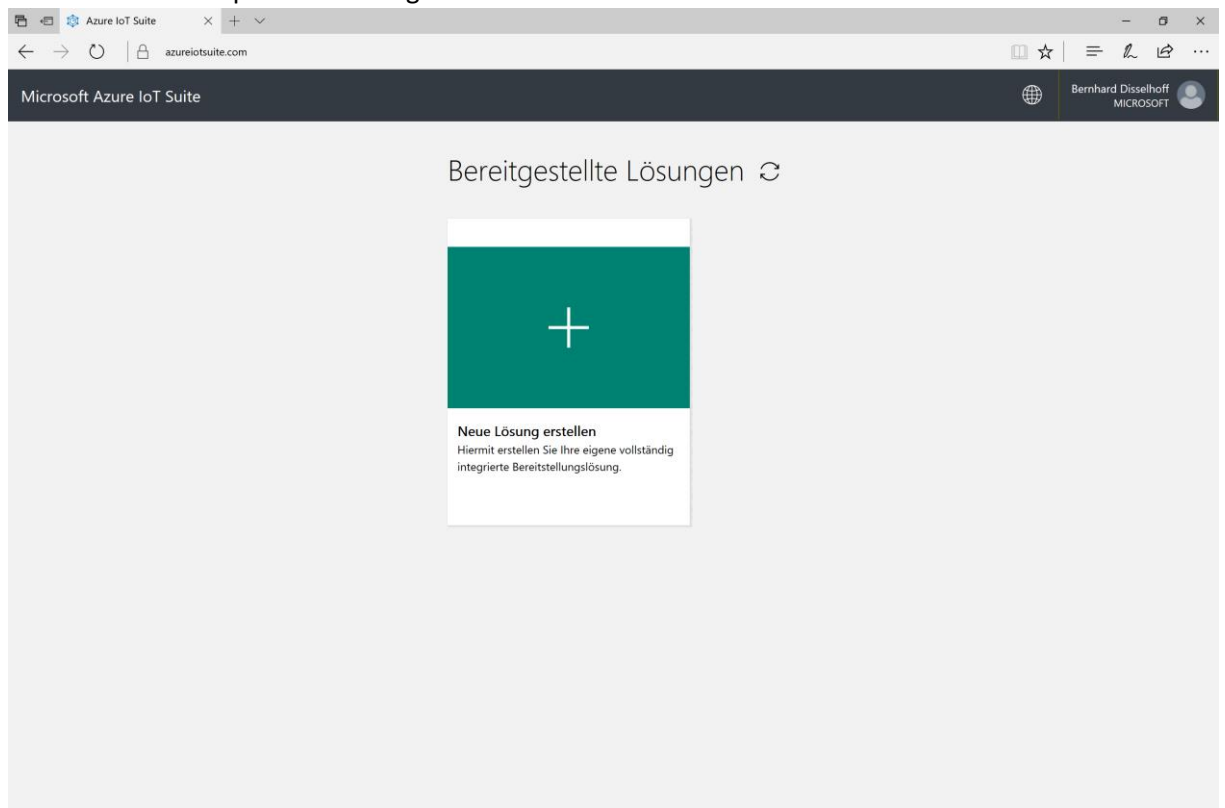
Nach dem Workshop sollen alle Komponenten aus dem Workshop angewandt werden und eine End-to-End Lösung laufen.

5. Ablauf des IoT Workshops

1. Erklären des Vorgehens und zusammensetzen des IoT Devices mit Sensor.



2. Einrichten der Sample Anwendung



4.


Azure IoT Suite

https://www.azureiotsuite.com/#solutions/types

Microsoft Azure IoT Suite

Bernhard Disselhoff
MICROSOFT

Lösungstypen



New

Geräte, um ungenutzte Daten zu analysieren und die Geschäftsergebnisse durch die Automatisierung von Prozessen zu verbessern.


Diese aktualisierte Version der Remoteüberwachung bietet erweiterte Funktionen und Operatorszenarien basierend auf einer neuen Architektur, die Anpassungen ermöglicht.

Die neue Version umfasst Folgendes:

- Neu gestaltete Benutzeroberfläche
- Auf Microservices basierende Architektur
- Verfügbarkeit sowohl in .NET als auch in Java
- [Interaktive Demo anzeigen](#)

Rufen Sie die vorherige Version ab.


Auswählen



Verbundene Factory

Beschleunigen Sie Ihren Weg zu Industrie 4.0 – verbinden, überwachen und steuern Sie Industriegeräte über die OPC UA, um Einblicke zu erhalten und so die Produktivität und Rentabilität des Betriebs zu fördern.


Auswählen



Vorbeugende Wartung

Sehen Sie Wartungsbedarf voraus, und vermeiden Sie ungeplante Ausfallzeiten durch das Verbinden und Überwachen Ihrer Geräte für die vorbeugende Wartung.

Auswählen



Gerätesimulation

Optimieren Sie die Entwicklung Ihrer IoT-Lösung durch den Einsatz von simulierten IoT-Geräten für das Erstellen und Testen Ihrer Lösung im gesamten Softwareentwicklungs-Lebenszyklus.

Auswählen

<https://www.azureiotsuite.com/#solutions/types/RM>

Feedback

Unten, runterscrollen und die vorherige Version auswählen.


Azure IoT Suite

azureiotsuite.com/#solutions/types

Microsoft Azure IoT Suite

Bernhard Disselhoff
MICROSOFT


Lösungstypen



Remoteüberwachung

Verbinden und überwachen Sie Ihre Geräte, um ungenutzte Daten zu analysieren und Geschäftsergebnisse durch die Automatisierung von Prozessen zu verbessern.


Auswählen



Verbundene Factory

Beschleunigen Sie Ihren Weg zu Industrie 4.0 – verbinden, überwachen und steuern Sie Industriegeräte über die OPC UA, um Einblicke zu erhalten und so die Produktivität und Rentabilität des Betriebs zu fördern.

Auswählen



Vorbeugende Wartung

Sehen Sie Wartungsbedarf voraus, und vermeiden Sie ungeplante Ausfallzeiten durch das Verbinden und Überwachen Ihrer Geräte für die vorbeugende Wartung.

Auswählen

<https://www.azureiotsuite.com/#solutions/types/RM>

Azure IoT SuiteDashboard - Microsoft Azureazureiotsuite.com/#solutions/types/RM

Microsoft Azure IoT SuiteBernhard DisselhoffMICROSOFT

Remoteüberwachung-Lösung erstellen

Lösungsdetails

Durch das Erstellen einer Lösung werden folgende Azure-Dienste kostenpflichtig in Ihrem Azure-Abonnement bereitgestellt:

- 1 Azure Active Directory application
- 1 IoT Hub (S2 - Standard tier)
- 1 DocumentDB Account (S1)
- 2 Event Hubs (Basic throughput unit)
- 1 Storage account (Standard-GRS)
- 3 Stream Analytics jobs (1 streaming unit per job)
- 1 Azure App Service Web App for Website (P1 - Premium: 2 small)
- 1 Azure App Service Web App for Web jobs (S1 - Standard: 2 small) running 25 simulated devices by default

Preisinformationen für diese Dienste finden Sie [hier](#). Verbrauchsbeträge und Abrechnungsinformationen für Ihr Abonnement finden Sie im [Azure-Portal](#).

Zusätzlich zu den oben genannten Azure-Diensten registrieren Sie sich durch das Erstellen einer Lösung auch für ein Abonnement für die folgenden Azure Marketplace-Angebote, die den folgenden Bedingungen unterliegen:

[Bing Karten-API für Unternehmen \(interne Websitetransaktionen der Stufe 1\); Nutzungsbedingungen und Datenschutzbestimmungen.](#)

Lösungsname

ESP-BED

Abonnement

Microsoft Azure Internal Consumption

Um mit der Erstellung fortzufahren, klicken Sie unten auf "Ich stimme zu". Ihre Lösung wird mit einer statischen Karte erstellt. Um eine interaktive Karte hinzuzufügen, befolgen Sie die Anweisungen unter [Häufig gestellte Fragen](#).

☒ Ich stimme zu

Region

Europa, Westen

Lösung erstellen

Abbrechen

Dashboard - Microsoft AzureAzure IoT Suiteazureiotsuite.com

Microsoft Azure IoT SuiteBernhard DisselhoffMICROSOFT

Bereitgestellte Lösungen ↻

Neue Lösung erstellen
Hiermit erstellen Sie Ihre eigene vollständig integrierte Bereitstellungslösung.

Wird bereitgestellt...
ESP-BED
Überwachen Sie Ereignisse und Bedingungen mit Ihren Mobilgeräten.
[Details](#)

Dashboard - Microsoft Azure

Azure IoT Suite

azureiotsuite.com/#

Microsoft Azure IoT Suite


Bernhard Disselhoff
MICROSOFT

Bereitgestellte Lösungen ↻

Neue Lösung erstellen

Hiermit erstellen Sie Ihre eigene vollständig integrierte Bereitstellungslösung.

Wird bereitgestellt...



ESP-BED

Überwachen Sie Ereignisse und Bedingungen mit Ihren Mobilgeräten.

Details

ESP-BED

Wird bereitgestellt

Ihre Lösung Remoteüberwachung wird in der Region Europa, Westen bereitgestellt.

Bereitstellungszustände

IoTPortal

Microsoft.Resources/Deployments

ESP-BED-DeviceInfo

Microsoft.StreamAnalytics/streamingjobs

ESP-BED-Telemetry

Microsoft.StreamAnalytics/streamingjobs

ESP-BED-Rules

Microsoft.StreamAnalytics/streamingjobs

ESP-BED/MSDeploy

Microsoft.Web/deployments

ESP-BED

Microsoft.Azure.Devices/iot-hubs

espsbed

Microsoft.Storage/storageAccounts

esp-bede3c05

Microsoft.Devices/IoTHubs

esp-bed

Microsoft.EventHub/namespaces

esp-bed7a9f

Microsoft.DocumentDb/databaseAccounts

esp-bed/esp-bed-ehout

Microsoft.EventHub/namespaces/eventhubs

esp-bed/esp-bed-ehruleout

Microsoft.EventHub/namespaces/eventhubs

esp-bed/RootManageSharedAccessKey

Microsoft.EventHub/namespaces/eventhubs

esp-bede3c05/iotHubowner

Microsoft.Devices/IoTHubs/IoTHubKeys

esp-bede3c05/events/telemetrycg

Microsoft.Devices/IoTHubs/eventhubs/eventhubs/ConsumerGroups

Dashboard - Microsoft Azure

Azure IoT Suite

azureiotsuite.com/#

Microsoft Azure IoT Suite


Bernhard Disselhoff
MICROSOFT

Bereitgestellte Lösungen ↻

Neue Lösung erstellen

Hiermit erstellen Sie Ihre eigene vollständig integrierte Bereitstellungslösung.

Bereit



ESP-BED

Überwachen Sie Ereignisse und Bedingungen mit Ihren Mobilgeräten.

Starten

ESP-BED

Bereit

Hier können Sie die Ausführung Ihrer vorkonfigurierten Lösung anzeigen:

Lösungsdashboard

Region

Europa, Westen

Abonnement-ID

4be6abc1-fc3d-4e6d-8297-87af9dfb2c6c

Lösung ändern

Zeigen Sie hier die bereitgestellten Azure-Ressourcen an, aus denen Ihre vorkonfigurierte Lösung besteht: [Azure Management Portal](#).

Zeigen Sie hier den Quellcode für diese vorkonfigurierte Lösung an: [GitHub](#).

Ressourcen

[Dokumentation für Entwickler](#)

Aktionen

Lösung löschen

Geräte-Explorer - Microsoft Azure IoT Suite IoT Device Portal - Dash

esp-bed.azurewebsites.net/Dashboard/Index

Microsoft Azure IoT Suite - Lösung für die Remoteüberwachung

bedissel@microsoft.com ADMINISTRATOR

DASHBOARD

GERÄTE

REGELN

AKTIONEN

VERWALTUNGSAU...

ERWEITERT

GERÄT HINZUFÜGEN

Anzuzeigendes Gerät:

Telemetrieverlauf

Alarmverlauf

ORTSZEIT	GERÄTE-ID	REGELAUSGABE	WERT
No matching records found			

Aktive Aufträge: 0

Geplante Aufträge: 0

Auftragsfehler in den letzten 24 Stunden: 0

In den letzten 24 Stunden abgeschlossene Aufträge: 0

Max. Luftfeuchtigkeit für Gerät: 000

Min. Luftfeuchtigkeit für Gerät: 000

Durchschnittliche Luftfeuchtigkeit für Gerät: 000

5. Installation der lokalen Entwicklungsumgebung

Geräte-Explorer - Microsoft Azure IoT Suite IoT Device Portal - Dashbo Arduino - Home

arduino.cc

ARDUINO

HOME BUY SOFTWARE PRODUCTS LEARNING FORUM SUPPORT BLOG

WHAT IS ARDUINO?

BUY AN ARDUINO

LEARN ARDUINO

SUPPORT ARDUINO

ARDUINO PROJECT HUB

INSPIRATION FOR YOUR PROJECTS!

ARDUINO UNO THE CLASSIC ARDUINO TO GET STARTED SHOP NOW

THIS GIANT ELECTROMECHANICAL CLOCK IS A WORK OF ART


https://www.arduino.cc/en/Main/Software

Geräte-Explorer - Microsoft Azure IoT Suite IoT Device Portal - Dashboard Arduino - Software

arduino.cc/en/Main/Software

HOME BUY SOFTWARE PRODUCTS LEARNING FORUM SUPPORT BLOG

Download the Arduino IDE



ARDUINO 1.8.3
The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software. This software can be used with any Arduino board. Refer to the [Getting Started](#) page for installation instructions.

Windows installer
Windows ZIP file for non admin install

Windows app [Get it](#)

Mac OS X 10.7 Lion or newer

Linux 32 bits
Linux 64 bits
Linux ARM

[Release Notes](#)
[Source Code](#)
[Checksums \(sha512\)](#)

ARDUINO SOFTWARE
HOURLY BUILDS

LAST UPDATE
23 June 2017 17:13:9 GMT

Download a preview of the incoming release with the most updated features and bugfixes.

[Windows](#)
[Mac OS X](#) (Mac OSX Lion or later)
[Linux](#) 32 bit, [Linux](#) 64 bit, [Linux](#) ARM

ARDUINO 1.0.6 / 1.5.x / 1.6.x
PREVIOUS RELEASES

Download the previous version of the current release, the classic Arduino 1.0.x, or the Arduino 1.5.x Beta version.

All the Arduino 0xxx versions are also available for download. The Arduino IDE can be used on Windows, Linux (both 32 and 64 bits), and Mac OS X.

Source Code

Active development of the Arduino software is [hosted by GitHub](#). See the instructions for [building the code](#).

https://www.arduino.cc/download_handler.php?fl=https://www.microsoft.com/store/apps/9nblggh4rsd8?ocid=badge are available [here](#). The archives are PGP-signed so they can be verified using


Oder

Geräte-Explorer - Microsoft Azure IoT Suite IoT Device Portal - Dashboard Arduino

Store

Startseite Apps Spiele Musik Filme & TV

Suchen



Arduino IDE
Arduino LLC
★★★★★ 36
Dieses Produkt ist installiert.

[Starten](#) [Teilen](#)

Bildschirmfotos

Beschreibung

Arduino is an open-source electronics platform based on easy-to-use hardware and software. It's intended for anyone making interactive projects.

IMPORTANT: This app performs with core functionality on Windows 10 S but some limited plugins do not work. We are working with Microsoft on a fix.

Verfügbar auf

PC

USK ab 0 Jahren

Anderen gefällt auch

[Alle anzeigen](#)

slack, MD EDITOR, etc.

The source code archives of the latest release are available [here](#)

Seite 11 von 12 213 Wörter

6. Einrichten des Devices

Voreinstellungen

Einstellungen

Netzwerk

Sketchbook-Speicherort:

C:\Users\bedissel\Documents\Arduino

Durchsuchen

Editor-Sprache:

Systemstandard

(erfordert Neustart von Arduino)

Editor-Textgröße:

12

Oberflächen-Zoomstufe:

☒Automatisch

100%

(erfordert Neustart von Arduino)

Ausführliche Ausgabe während:

☐Kompilierung

☐Hochladen

Compiler-Warnungen:

Keine

☐Zeilennummern anzeigen

☐Code-Faltung aktivieren

☒Code nach dem Hochladen überprüfen

☐Externen Editor verwenden

☒Aggressively cache compiled core

☒Beim Start nach Updates suchen

☒Sketche beim Speichern auf die neue Dateierweiterung aktualisieren (.pde -> .ino)

☒Speichern beim Überprüfen oder Hochladen

Zusätzliche Boardverwalter-URLs:

esp8266.com/stable/package_esp8266com_index.json

Mehr Voreinstellungen können direkt in der Datei bearbeitet werden

C:\Users\bedissel\Documents\ArduinoData\preferences.txt

(nur bearbeiten, wenn Arduino nicht läuft)

OK

Abbruch

Boardverwalter

Typ

Alle

Grenzen Sie Ihre Suche ein...

SmartEverything Fox, SmartEverything Fox3, SmartEverything Lion, SmartEverything Dragonfly, Analog ADI, SmartTutto.

[Online help](#)

[More info](#)

EMORO 2560 by Inovatic-ICT

In diesem Paket enthaltene Boards:

EMoRo 2560. Board based on ATmega 2560 MCU.

[Online help](#)

[More info](#)

AMEL-Tech Boards by replaced by Arrow Boards

In diesem Paket enthaltene Boards:

SmartEverything Fox.

[Online help](#)

[More info](#)

esp8266 by ESP8266 Community Version 2.3.0 INSTALLED

In diesem Paket enthaltene Boards:

Generic ESP8266 Module, Olimex MOD-WIFI-ESP8266(-DEV), NodeMCU 0.9 (ESP-12 Module), NodeMCU 1.0 (ESP-12E Module), Adafruit HUZZAH ESP8266 (ESP-12), ESPresso Lite 1.0, ESPresso Lite 2.0, SparkFun Thing, SweetPea ESP-210, WeMos D1, WeMos D1 mini, ESPino (ESP-12 Module), ESPino (WROOM-02 Module), WifiInfo, ESPDuino.

[Online help](#)

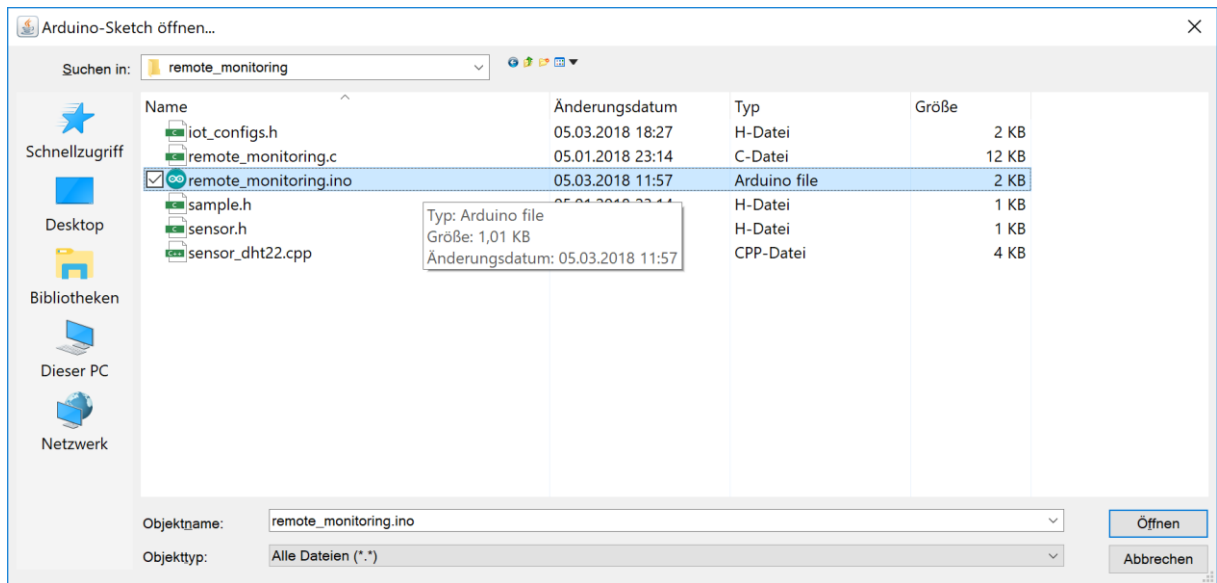
[More info](#)

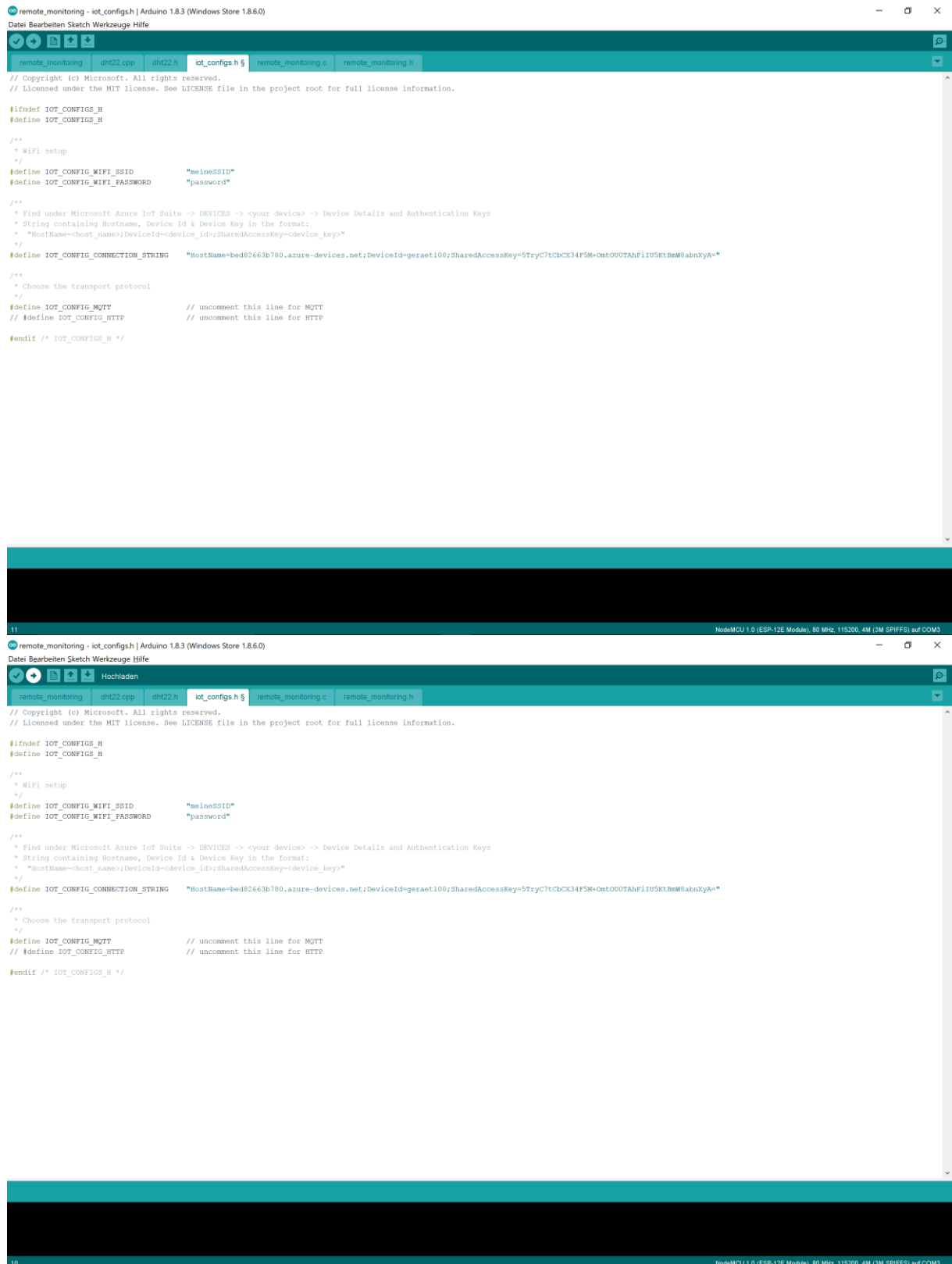
Version auswählen

Installieren

Entfernen

Schließen





8. Starten und Testen und bei Bedarf debuggen des Devices

remote_monitoring | Arduino 1.8.3 (Windows Store 1.8.6.0)

Datei Bearbeiten Sketch Werkzeuge Hilfe

remote_monitoring dht22.cpp dht22.h iot_configs.h remote_monitoring.c remote_monitoring.h

```
// Copyright (c) Microsoft. All rights reserved.
// Licensed under the MIT license. See LICENSE file in the project root for full license information.

// Please use an Arduino IDE 1.6.8 or greater

#include "iot_configs.h"

#include <ESP8266WiFi.h>
#include <WiFiClientSecure.h>
#include <WiFiUDP.h>

#include <AzureIoTHub.h>
#if defined(IOT_CONFIG_MQTT)
    #include <AzureIoTProtocol_MQTT.h>
#elif defined(IOT_CONFIG_HTTP)
    #include <AzureIoTProtocol_HTTP.h>
#endif

#include "remote_monitoring.h"

#define LED D7

static char ssid[] = IOT_CONFIG_WIFI_SSID;
static char pass[] = IOT_CONFIG_WIFI_PASSWORD;

void setup() {
    initSerial();
    initWifi();
    initTime();
}

void loop() {
    // Run the Remote Monitoring from the Azure IoT Hub C SDK
    // You must set the device id, device key, IoT Hub name and IoT Hub suffix in
    // remote_monitoring.c
}
```

Hochladen...

Archiving built core (loading) in C:\Users\bediesel\AppData\Local\Temp\arduino_cache_131457\core\core_esp8266_esp8266_nodemcuvar2_cpufrequency_80_uploadspeed_115200_flashsize_4M3M_wifi00002171ad7a372a886ef72ec931..a
Der Sketch verwendet 428129 Bytes (40%) des Programmspeicherplatzes. Das Maximum sind 1044464 Bytes.
Globale Variablen verwenden 51868 Bytes (63%) des dynamischen Speichers, 30052 Bytes für lokale Variablen verbleiben. Das Maximum sind 81920 Bytes.
Uploading 432272 bytes from C:\Users\bediesel\AppData\Local\Temp\arduino_build_229443\remote_monitoring.ino.bin to flash at 0x00000000
..... [18%]
..... [37%]
..... [56%]
.....

NodeMCU 1.0 (ESP-12E Module), 80 MHz, 115200, 4M (3M SPIFFS) auf COM3

remote_monitoring | Arduino 1.8.3 (Windows Store 1.8.6.0)

Datei Bearbeiten Sketch Werkzeuge Hilfe

remote_monitoring c remote_monitoring.h

Automatische Formatierung Strg+T
Sketch archivieren
Kodierung korrigieren & neu laden
Serieller Monitor Strg+Umschalt+M
Serieller Plotter Strg+Umschalt+L
WiFi101 Firmware Updater
Board: "NodeMCU 1.0 (ESP-12E Module)"
CPU Frequency: "80 MHz"
Flash Size: "4M (3M SPIFFS)"
Upload Speed: "115200"
Port: "COM3"
Boardinformationen holen
Programmer: "AVRISP mkII"
Bootloader brennen

```
// Copyright (c) Microsoft. All rights reserved.
// Licensed under the MIT license. See LICENSE file in the project root for full license information.

// Please use an Arduino IDE 1.6.8 or greater

#include "iot_configs.h"

#include <ESP8266WiFi.h>
#include <WiFiClientSecure.h>
#include <WiFiUDP.h>

#include <AzureIoTHub.h>
#if defined(IOT_CONFIG_MQTT)
    #include <AzureIoTProtocol_MQTT.h>
#elif defined(IOT_CONFIG_HTTP)
    #include <AzureIoTProtocol_HTTP.h>
#endif

#include "remote_monitoring.h"

#define LED D7

static char ssid[] = IOT_CONFIG_WIFI_SSID;
static char pass[] = IOT_CONFIG_WIFI_PASSWORD;

void setup() {
    initSerial();
    initWifi();
    initTime();
}

void loop() {
    // Run the Remote Monitoring from the Azure IoT Hub C SDK
    // You must set the device id, device key, IoT Hub name and IoT Hub suffix in
    // remote_monitoring.c
}
```

Hochladen abgeschlossen.

Archiving built core (loading) in C:\Users\bediesel\AppData\Local\Temp\arduino_cache_131457\core\core_esp8266_esp8266_nodemcuvar2_cpufrequency_80_uploadspeed_115200_flashsize_4M3M_wifi00002171ad7a372a886ef72ec931..a
Der Sketch verwendet 428129 Bytes (40%) des Programmspeicherplatzes. Das Maximum sind 1044464 Bytes.
Globale Variablen verwenden 51868 Bytes (63%) des dynamischen Speichers, 30052 Bytes für lokale Variablen verbleiben. Das Maximum sind 81920 Bytes.
Uploading 432272 bytes from C:\Users\bediesel\AppData\Local\Temp\arduino_build_229443\remote_monitoring.ino.bin to flash at 0x00000000
..... [18%]
..... [37%]
..... [56%]
..... [75%]
..... [94%]
..... [100%]

NodeMCU 1.0 (ESP-12E Module), 80 MHz, 115200, 4M (3M SPIFFS) auf COM3

^

```
COM3
Please start shop first.
Fetching NTP epoch time failed! Waiting 2 seconds to retry.
Fri Jun 30 09:36:15 2017

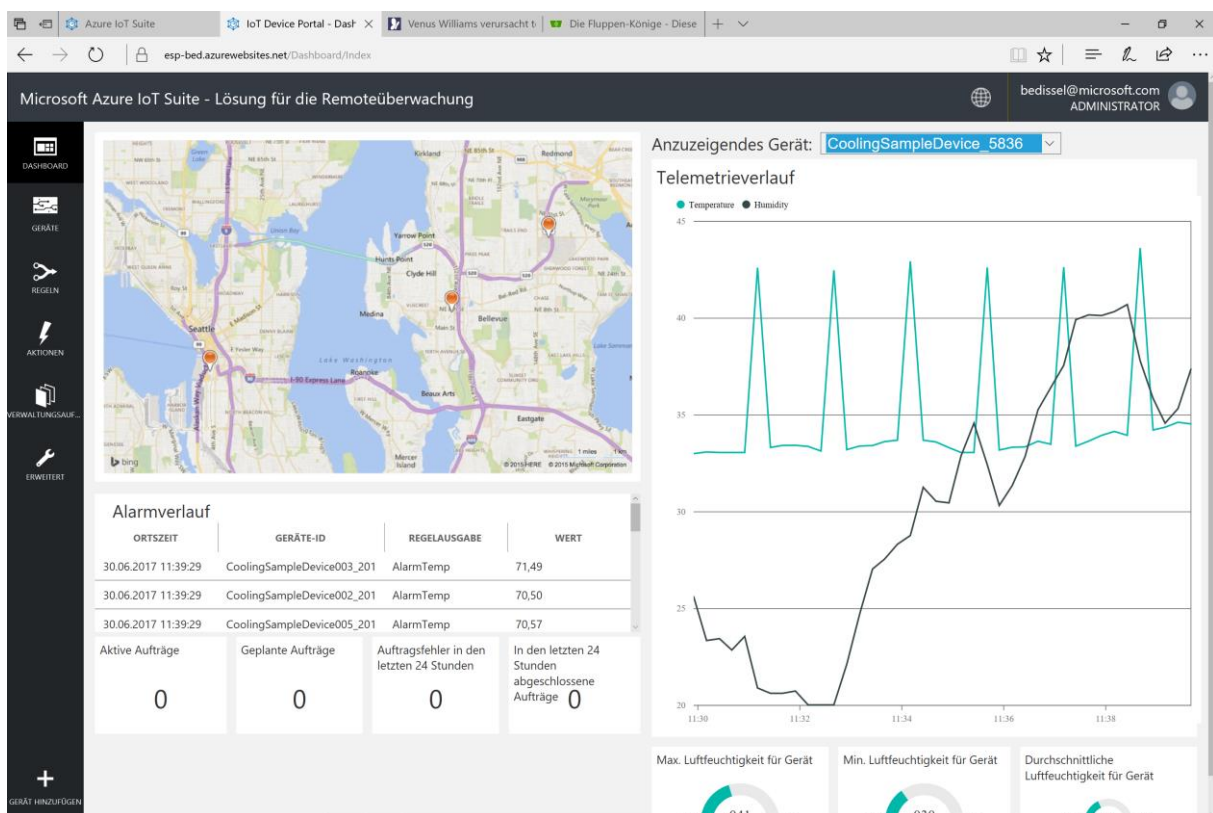
Fetched NTP epoch time is: 1498815376
DHTxx Unified Sensor Example
-----
Temperature
Sensor:      DHT21
Driver Ver:  1
Unique ID:   -1
Max Value:   80.00 *C
Min Value:   -40.00 *C
Resolution:  0.10 *C
-----

Humidity
Sensor:      DHT21
Driver Ver:  1
Unique ID:   -1
Max Value:   100.00%
Min Value:   0.00%
Resolution:  0.10%
-----

IoT Hub SDK for C, version 1.1.17
deviceId=geraet100
IoTHubClient accepted the message for delivery

Error reading temperature!
Error reading humidity!
Sending sensor value Temperature = 2, Humidity = 3

IoTHubClient accepted the message for delivery
```



9. Innerhalb von Microsoft Azure betrachten der eingesetzten Komponenten

Microsoft Azure Dashboard view showing resources and quickstart programs.

Alle Ressourcen

NAME	TYP
esp-bed7a9f	Azure Cosmos DB-Konto
ESP-BED-plan	App Service-Plan
ESP-BED-jobsp	App Service-Plan
ESP-BED	App Service
esp-bed	Event Hub
espbed	Speicherkonto
ESP-BED-plan	App Service
ESP-BED-DeviceInfo	Stream Analytics-Auftrag
esp-bede3c05	IoT Hub

Schnellstart-Lernprogramme

- Virtuelle Windows-Computer**: Hiermit stellen Sie Windows Server-, SQL Server- und SharePoint-VMs bereit.
- Virtuelle Linux-Computer**: Hiermit stellen Sie Ubuntu-, Red Hat-, CentOS-, SUSE- und CoreOS-VMs bereit.
- App Service**: Hiermit erstellen Sie Web-Apps mit .NET, Java, Node.js, Python und PHP.
- Functions**: Hiermit verarbeiten Sie Ereignisse mit einer serverlosen Codearchitektur.
- SQL-Datenbank**: Verwaltete relationale SQL-Datenbank als Database as a Service.

Azure Health

Personalized guidance and support when issues in Azure services affect you. [Learn more](#)

Marketplace

Microsoft Azure "Alle Ressourcen" view showing a table of resources.

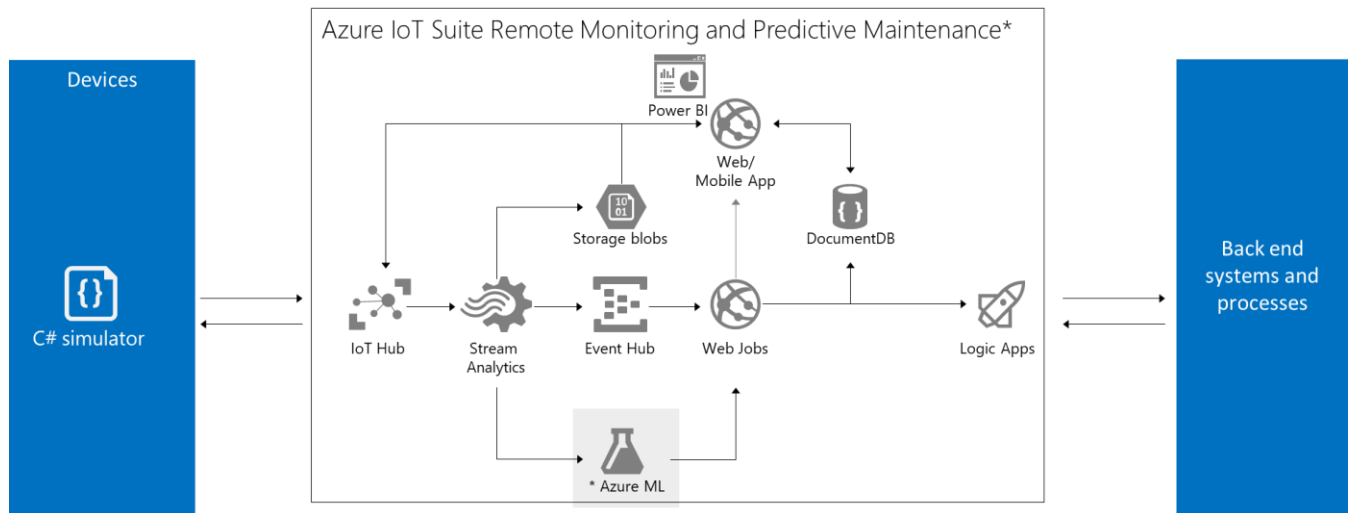
Abonnements: Microsoft Azure Internal Consumption – Wird kein Abonnement angezeigt? [Verzeichnis wechseln](#)

Nach Name filtern... Alle Typen Alle Standorte Keine Gruppierung

11 Elemente

NAME	TYP	RESSOURCENGRUPPE	STANDORT	ABONNEMENT
espbed	Speicherkonto	ESP-BED	Westeuropa	Microsoft Azure Internal Consumption ***
esp-bed	Event Hub	ESP-BED	Westeuropa	Microsoft Azure Internal Consumption ***
ESP-BED	App Service	ESP-BED	Westeuropa	Microsoft Azure Internal Consumption ***
ESP-BED-DeviceInfo	Stream Analytics-Auftrag	ESP-BED	Westeuropa	Microsoft Azure Internal Consumption ***
esp-bede3c05	IoT Hub	ESP-BED	Westeuropa	Microsoft Azure Internal Consumption ***
esp-bed7a9f	Azure Cosmos DB-Konto	ESP-BED	Westeuropa	Microsoft Azure Internal Consumption ***
ESP-BED-jobhost	App Service	ESP-BED	Westeuropa	Microsoft Azure Internal Consumption ***
ESP-BED-jobsp	App Service-Plan	ESP-BED	Westeuropa	Microsoft Azure Internal Consumption ***
ESP-BED-plan	App Service-Plan	ESP-BED	Westeuropa	Microsoft Azure Internal Consumption ***
ESP-BED-Rules	Stream Analytics-Auftrag	ESP-BED	Westeuropa	Microsoft Azure Internal Consumption ***
ESP-BED-Telemetry	Stream Analytics-Auftrag	ESP-BED	Westeuropa	Microsoft Azure Internal Consumption ***

<https://ms.portal.azure.com/#menu/account>



10. Abschluss
Wir haben Daten!!!