IOT Schulung
Arduino
2. Teil

Tagesablauf

- Vormittag
  - Besprechung mit Teilnehmer
    - Erfahrungen mit Thingworx
    - Eigene Projekte?
    - Hat für die Realisierung eines Projektes etwas gefehlt?
  - Ausgabe neue Hardware
  - Kurzer Ausblick über zukünftige Entwicklungen seitens der Vortragenden
  - Wiederholung des ersten Seminarteiles Teil 1
    - Anschließen der Hardware
    - HTTP Request
    - Neue Bibliothek für Arduino MKR Wifi 1010
    - Arduino IDE Installation für das Inbetriebnehmen der Schulungshardware

Tagesablauf

- Vormittag
  - Neuer Thingworx Server
    - Aufbau
    - Wichtige Einstellungen für die Inbetriebnahme

Tagesablauf

- Nachmittag
  - Wiederholung des ersten Seminarteiles Teil 2
    - Inbetriebnahme des neuen Arduinos am neuen Server
  - Einbindung eines neuen Sensor (hier der Ultraschallabstandssensor HC SR04)
    - Wie bindet man einen neuen (unbekannten) Sensor ein?
    - Funktionsweise des HC SR04
    - Einbindung des HC SR04 in den Arduino Code
    - Ausgabe im seriellen Monitor
  - Hausübung Drehdecoder KY-040 bis zum 3. Teil
    - Aufgabenstellung

Besprechung mit Teilnehmer

- Welche Erfahrungen habt ihr gemacht?
- Wurden eigene Projekte realisiert?
- Wurden Diplomarbeiten realisiert?
- Hat für die Realisierung eines Projektes etwas gefehlt?

Neue Hardware

- Arduino MKR Wifi 1010
- Ultraschall Abstandssensor HC SR04
- Kodierter Drehschalter KY-040
- Steckboard
- Benötige Jumper Kabel zum Verbinden

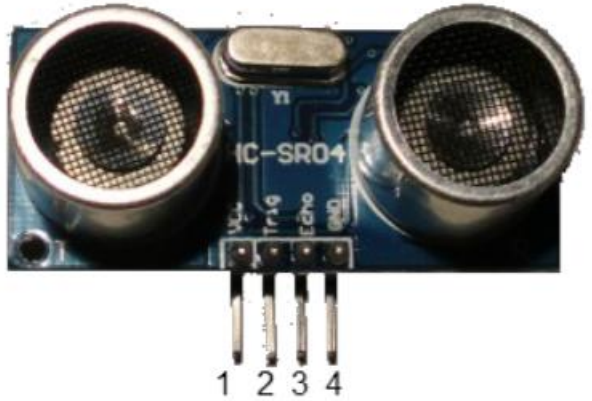
Neue Hardware

- Arduino MKR Wifi 1010
  - Link: https://store.arduino.cc/mkr-wifi-1010
  - ESP32 basierter Mikrocontroller mit Bluetooth und Wifi- Anbindung
  - Günstiger als MKR 1000 Modul
  - Entwicklung einer GUI (Graphical User Interface) auf ESP32 Modul im Gange

Quelle: https://store.arduino.cc/mkr-wifi-1010 (19.10.2019)

Neue Hardware

- Ultraschall Abstandssensor HC SR04
  - Datenblatt: https://www.mikrocontroller.net/attachment/218122/HC-SR04_ultraschallmodul_beschreibung_3.pdf
  - Abstand wird basierend auf einer Ultraschall Laufzeitmessung ermittelt



Quelle: https://www.mikrocontroller.net/attachment/218122/HC-SR04_ultraschallmodul_beschreibung_3.pdf(19.10.2019)

Neue Hardware

- Kodierter Drehschalter KY-040
  - Datenblatt: https://www.handsontec.com/dataspecs/module/Rotary%20Encoder.pdf
  - Ermittlung zur Stellung eines rotierenden Körpers inkl. Drehrichtungsangabe

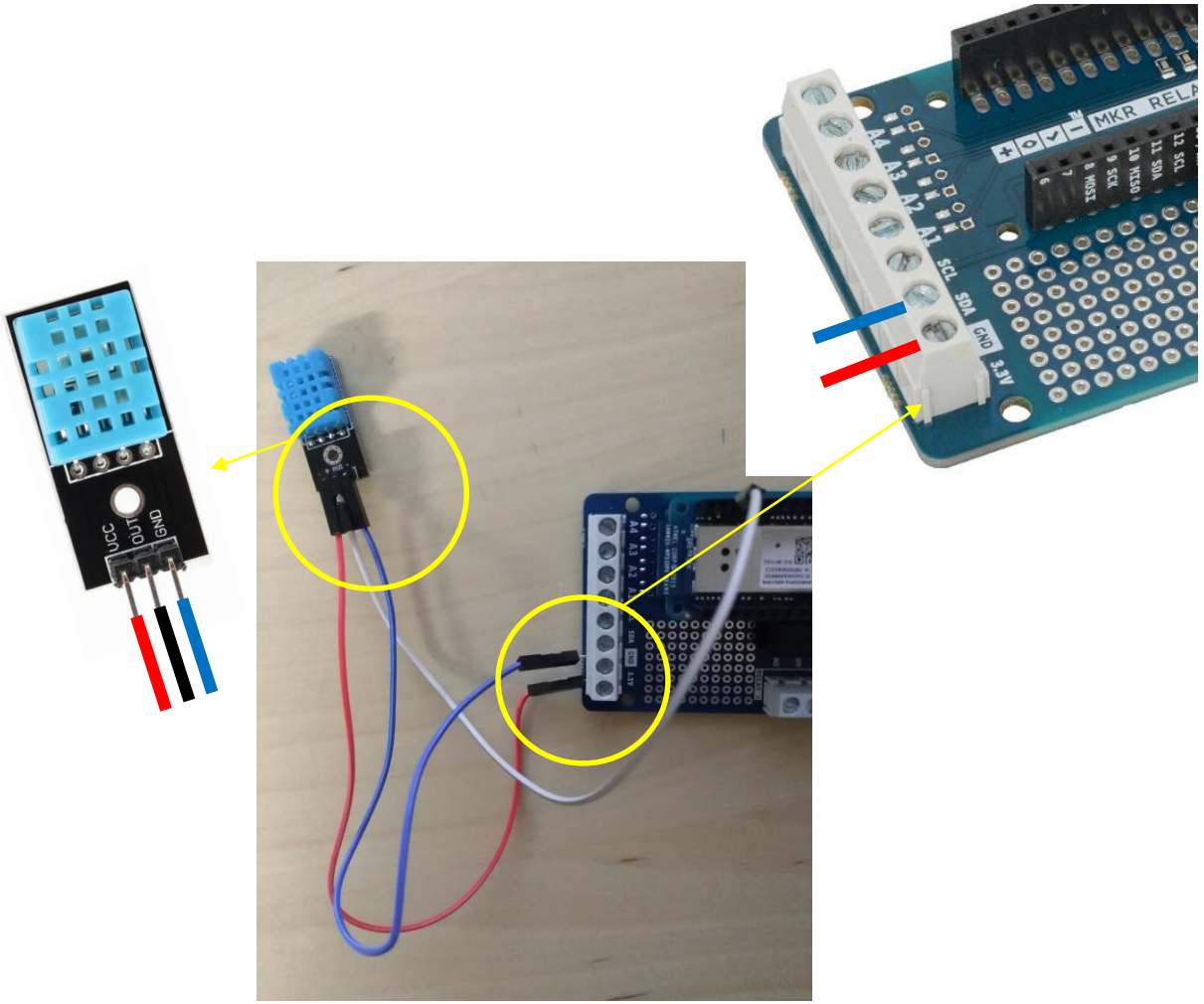
Quelle: https://www.handsontec.com/dataspecs/module/Rotary%20Encoder.pdf(19.10.2019)

Zukünftige Entwicklungen

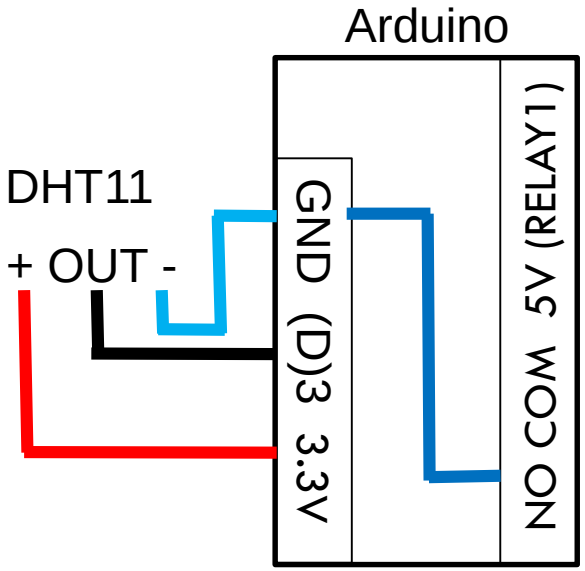
ESP32 Webserver

- Erstellung einer GUI (Graphical User Interface) für das Handling von Thingworx
- Verbindung mittels Laptop/PC/Handy mit Arduino MKR Wifi 1010
- Einstellungen des Servers, Things, Passwörter, usw. per Drop Down Menü
- Ebenfalls werden oft verwendete Sensoren im Bereich Maschinenbau eingebunden
- Idee dahinter: Schnelles schicken der Daten von oft verwendeten Sensoren im Bereich Maschinenbau ohne Erstellung eines Codes (ohne Programmierkenntnisse) an die Thingworx Cloud.
- IN ENTWICKLUNG!!! von Martin Schubert und Armin Fischer (HTL Wiener Neustadt, Abteilung Automatisierungstechnik) → https://youtu.be/Iytl2_7vRSE

Wiederholung des ersten Seminarteiles



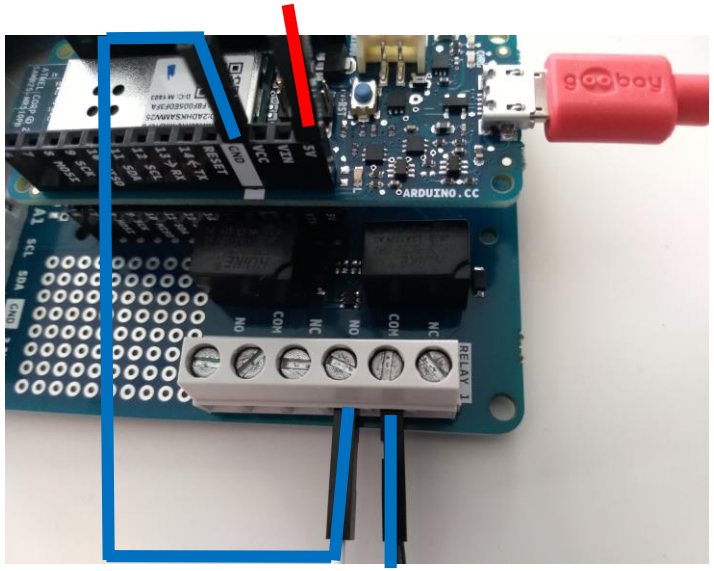
Anschluss DHT11

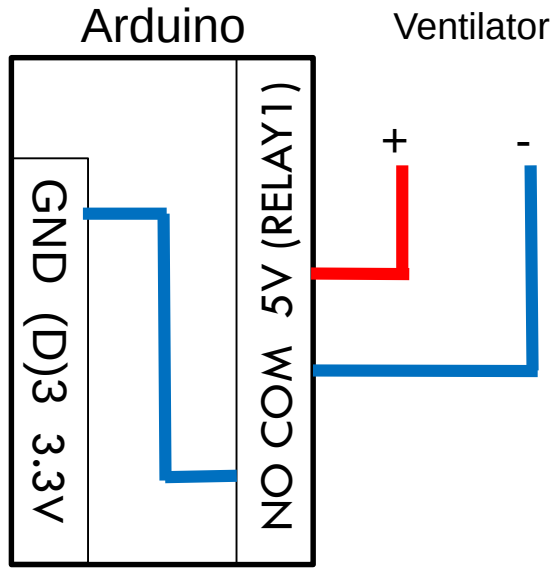


Quelle: Seminarunterlagen IOT Schulung Teil 1, Armin Fischer

Wiederholung des ersten Seminarteiles

Anschluss Ventilator

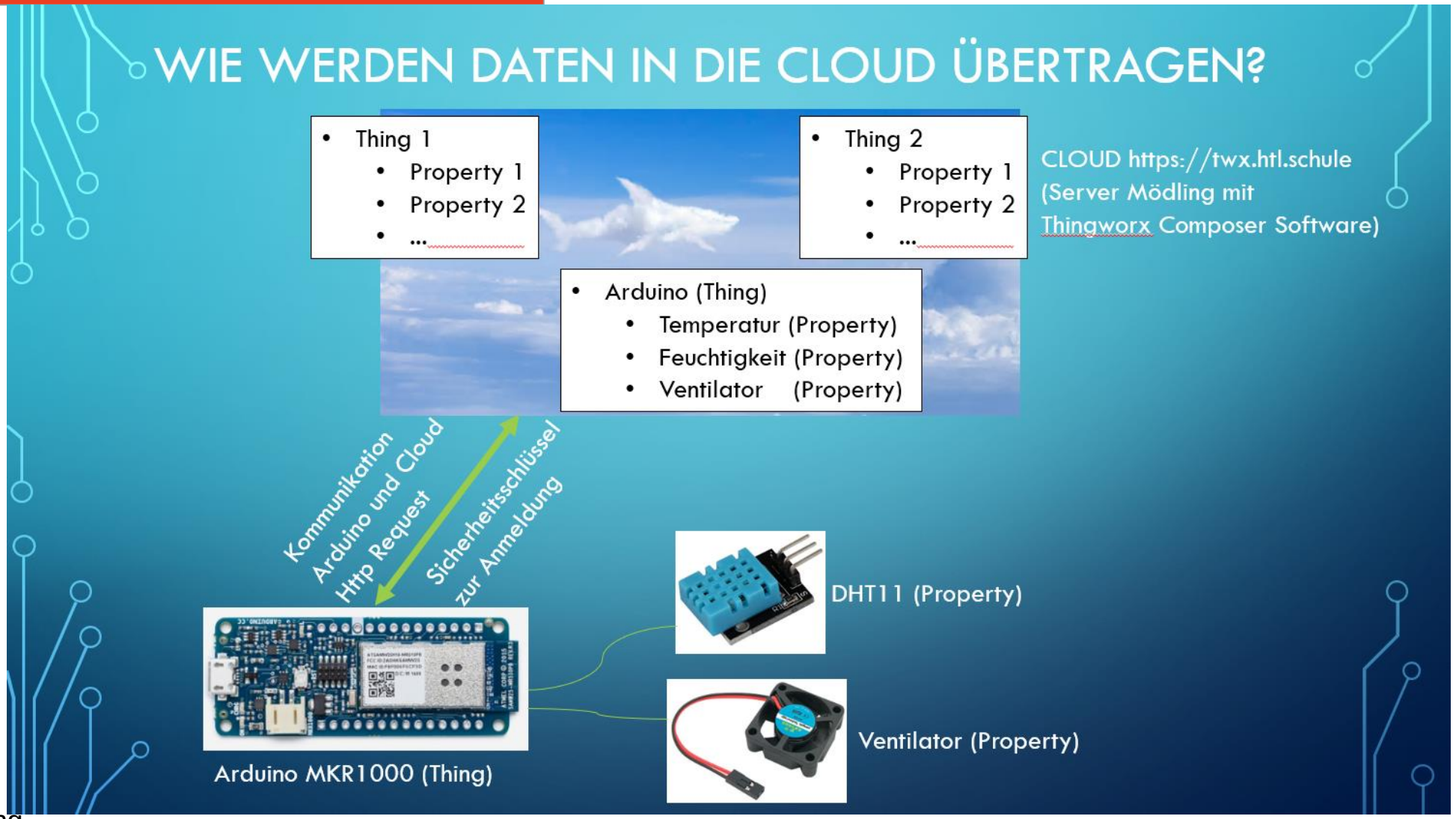




Quelle: Seminarunterlagen IOT Schulung Teil 1, Armin Fischer

Wiederholung des ersten Seminarteiles

Datenübertragung



Quelle: Seminarunterlagen IOT Schulung Teil 1, Armin Fischer

Wiederholung des ersten Seminarteiles

HTTP Request

HTTP REQUEST

- Protokoll zum Übertragen von Daten (genauso wie im Browser)
- 1 HTTP Request um 1 Sensorwert (Property) vom Arduino (Thing) auf die Cloud zu übertragen
- HTTP Request enthält Sicherheitsschlüssel um eine automatische Authentifizierung/Anmeldung zur Cloud zu erhalten
- Beispiel

Thingworx Server Adresse | Thing | Property

https://twx.htl.schule/Thingworx/Things/920417_18FISA_00_Dummy/Properties/Sensorwert_1

?appKey=de5d03ab-6cac-4e13-a4a9-9b5bfe9f7eda

Sicherheitsschlüssel

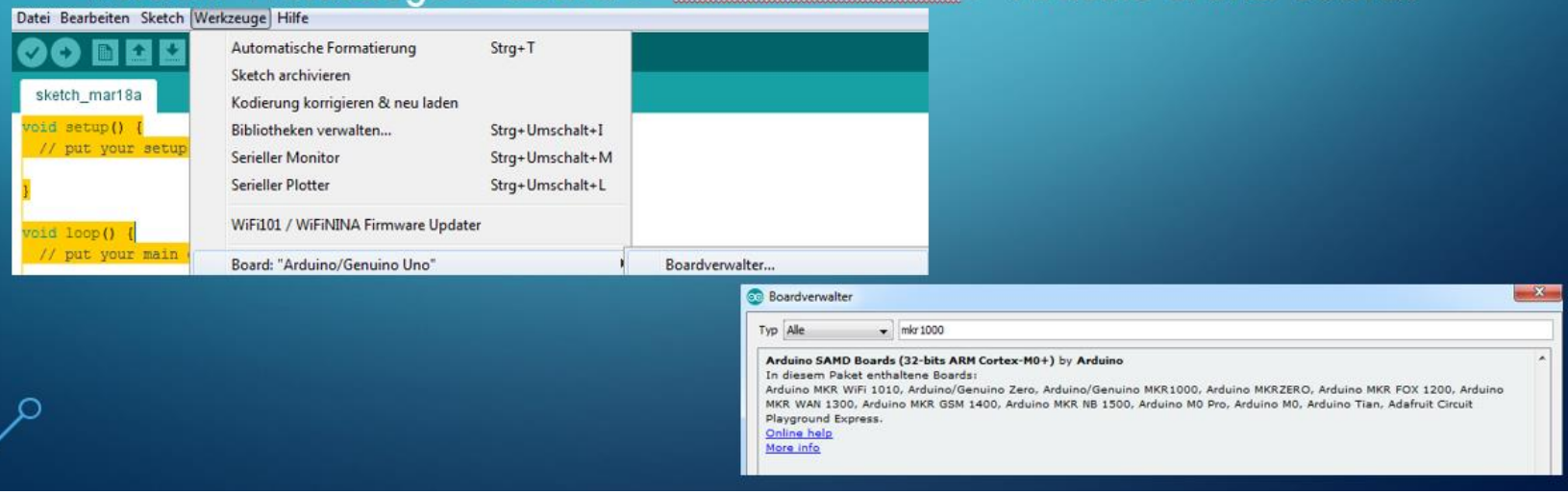
Quelle: Seminarunterlagen IOT Schulung Teil 1, Armin Fischer

Wiederholung des ersten Seminarteiles

Konfigurieren der Arbeitsumgebung

ARDUINO IDE

- Software zum Programmieren von Arduino Mikrocontroller
- Download unter https://www.arduino.cc/en/main/software
- Um das Arduino MKR1010 Board verwenden zu können muss dies installiert werden: Werkzeuge > Board > Boardverwalter > Arduino SAMD Boards



Quelle: Seminarunterlagen IOT Schulung Teil 1, Armin Fischer

Wiederholung des ersten Seminarteiles

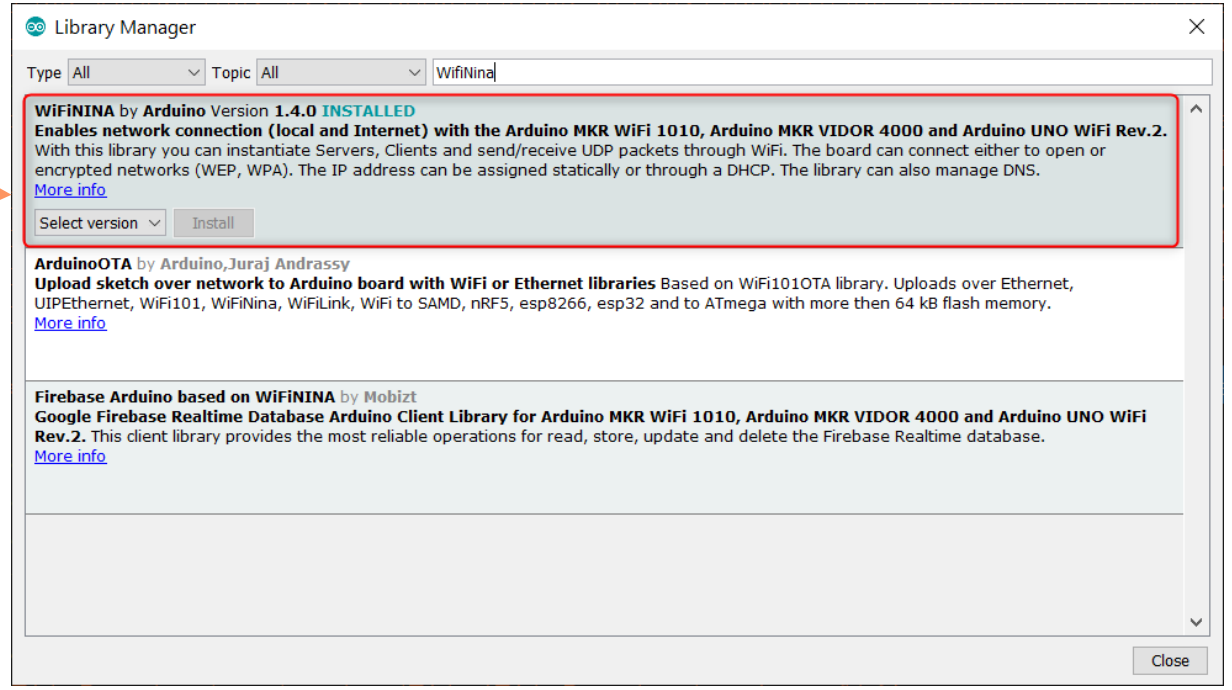
Neue Bibliotheken für Arduino MKR 1010

- Der neue Prozessor (jetzt ESP32 vorher SAMD21 Cortex-M0+) bedingt neue Befehle. Darum muss eine neue Library eingebunden werden.

- Libraries = Vorprogrammierte Codes, welche mit Befehlen aufgerufen werden.

- Benötige Libraries für den HTTP Request
  - WifiNINA.h

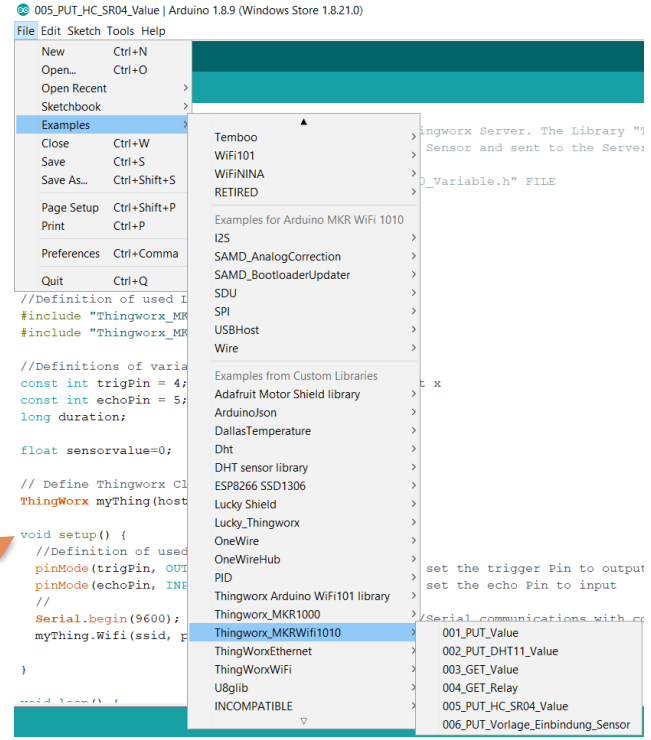
  (Werkzeuge > Bibliotheken verwalten > WifiNINA.h
  - Thingworx MKR1010
  - Dht.h



Wiederholung des ersten Seminarteiles

Neue Bibliotheken für Arduino MKR 1010

- Libraryname: Thingworx_MKRWifi1010
- Library von Armin Fischer geschrieben
- Muss in folgenden Ordner kopiert werden \Documents\Arduino\libraries
- Mit dieser Library sind folgende Operationen möglich
  - Verbinden mit WLAN
  - Abfragen eines Wertes am Thingworx Server
  - Schicken eines Wertes zum Thingworx Server
- In der Arduino IDE unter Beispiele befinden sich auch vorprogrammierte Beispiele.



Neuer Thingworx Server

- Neue Serverarchitektur (weltweit einzigartig) an der HTL Mödling
- Jede Schule hat einen Thingworxadministrator, welcher berechtigt ist, bis zu derzeit 2 Instanzen zu erstellen.
- Eine Instanz besteht aus einem Vuforia Studio Experience und einem Thingworx Composer Server.
- Vorteile zum Vorgänger:
  - Jede Schule arbeitet unabhängig der anderen Schulen
  - Für Laborübungen können Instanzen einfach gelöscht und wieder erstellt werden.

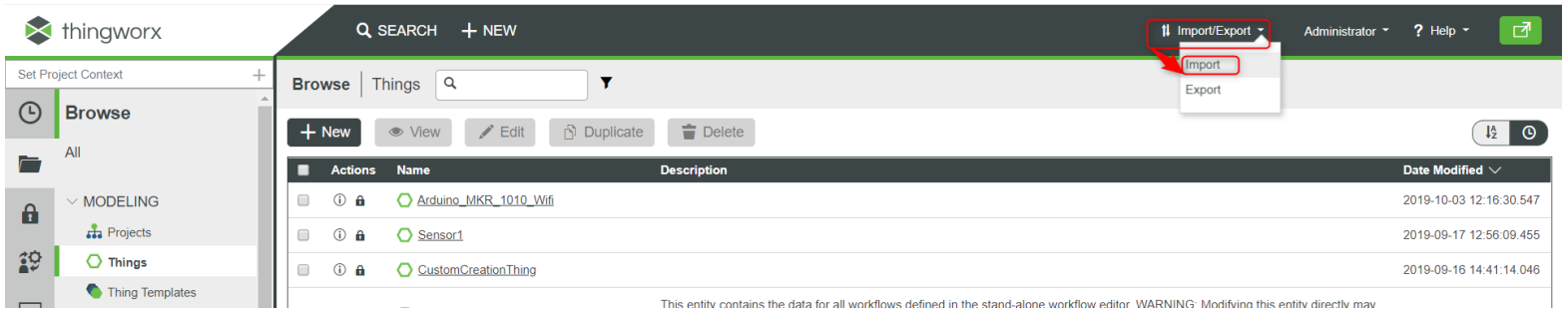
Neuer Thingworx Server

Einstellungen

- Es müssen folgende Einstellungen am Server gemacht werden um Daten mit einem Thing auszutauschen

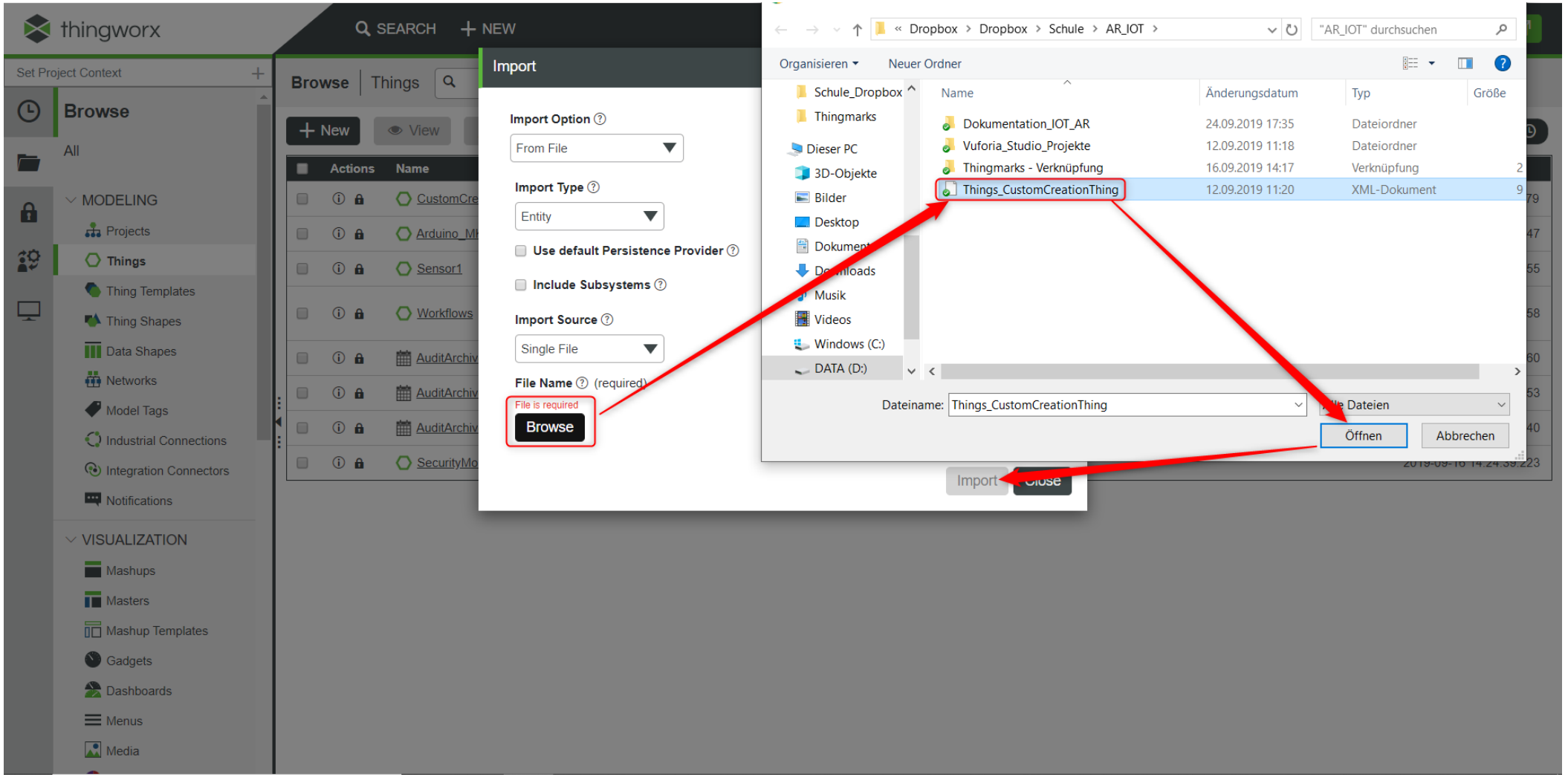
1. Auf den Server eine limitierte Admingruppe erstellen
   - Es wird dazu ein File namens “Things_CustomCreationThing.xml“ benötigt. ACHTUNG: Dieses File ist gerade in Bearbeitung und wird verbessert.

   - Importieren dieses Files



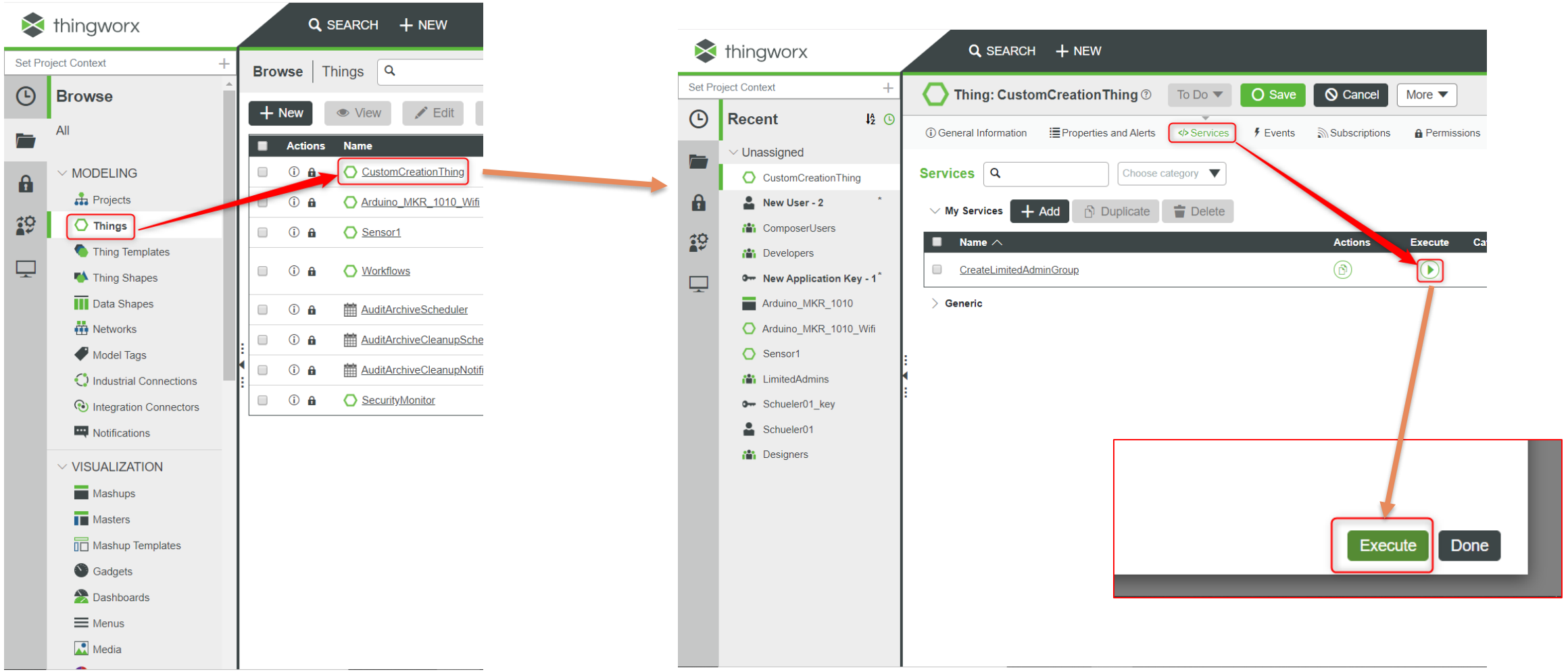
Neuer Thingworx Server

Importieren des Files



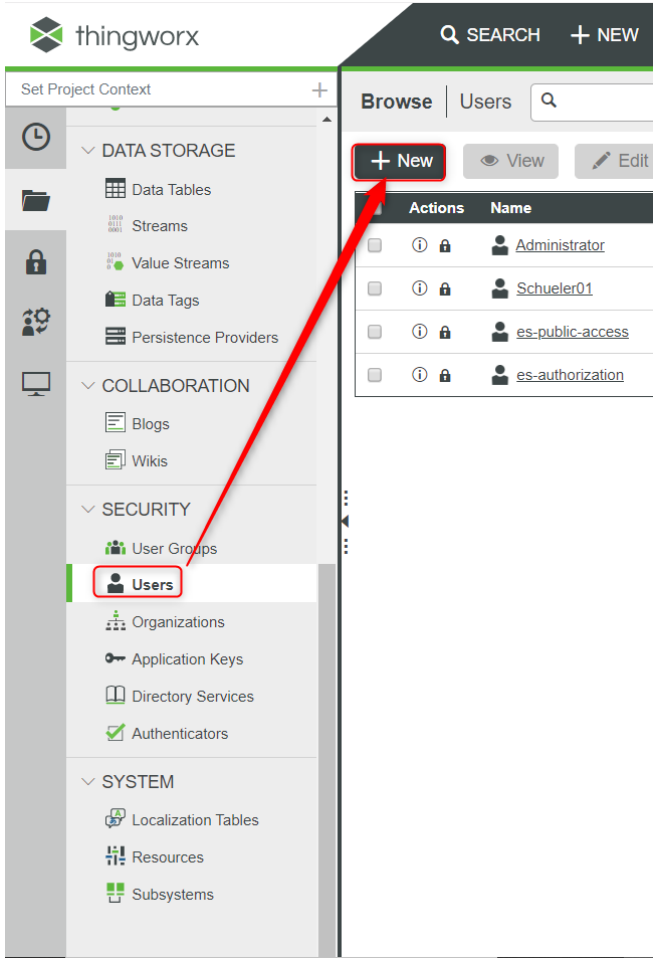
Neuer Thingworx Server

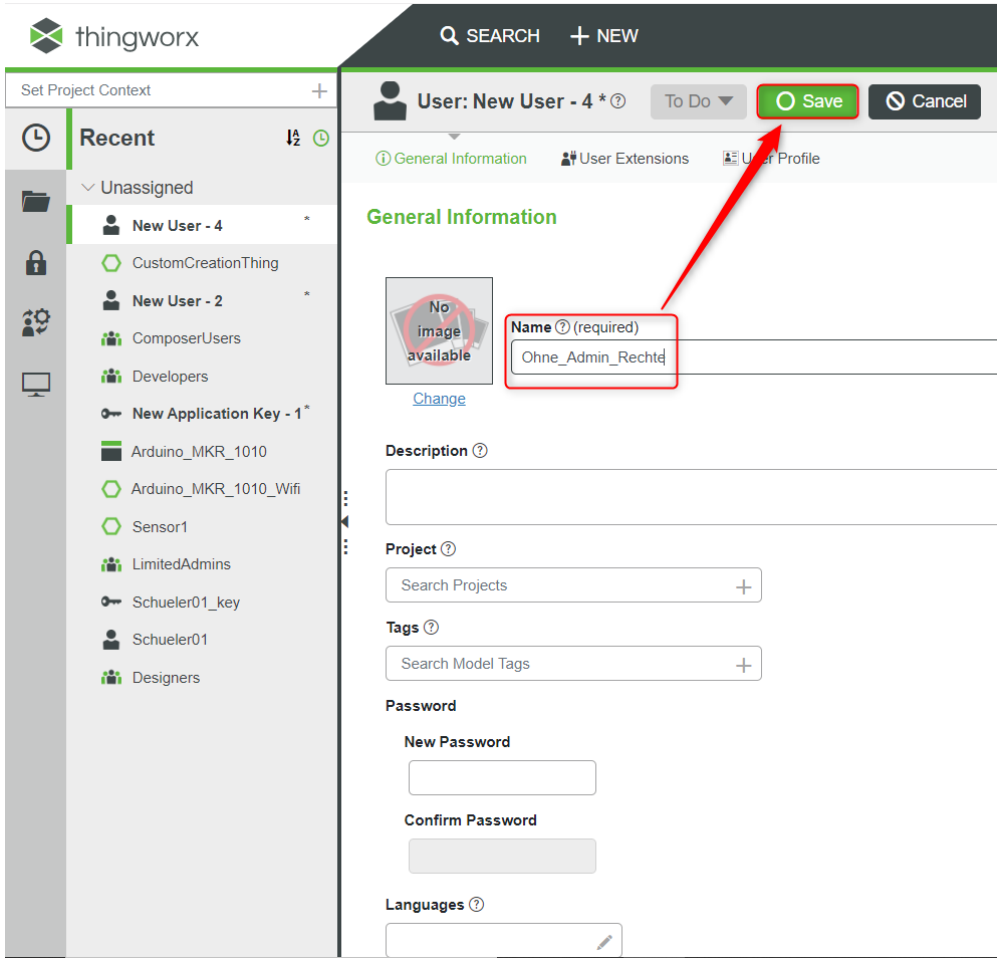
Service des Custom Creation Thing ausführen



Neuer Thingworx Server

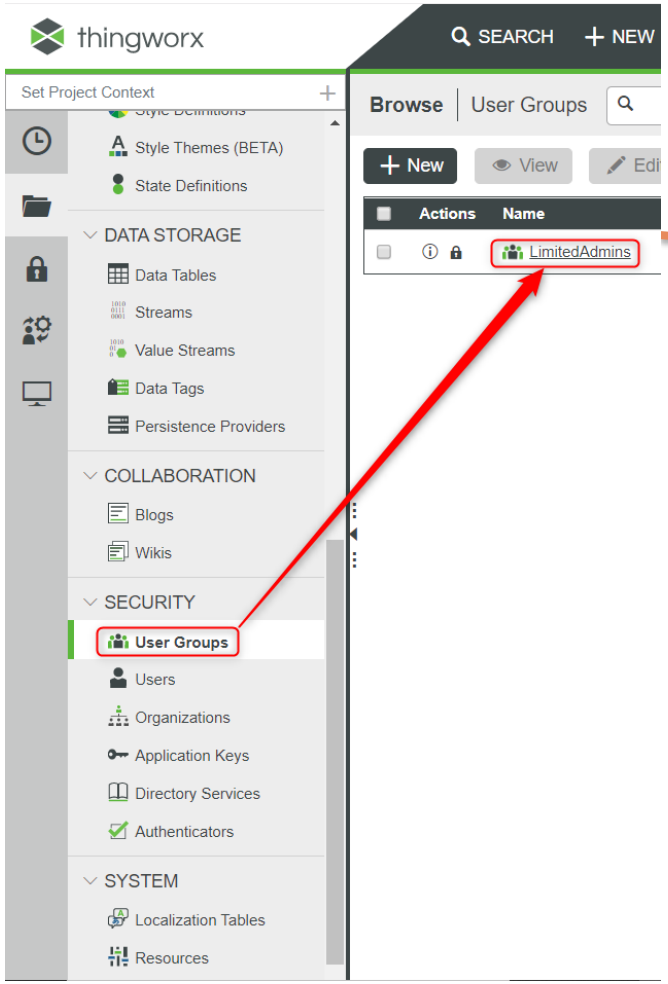
Neuen User „ohne Admin Rechte“ erstellen

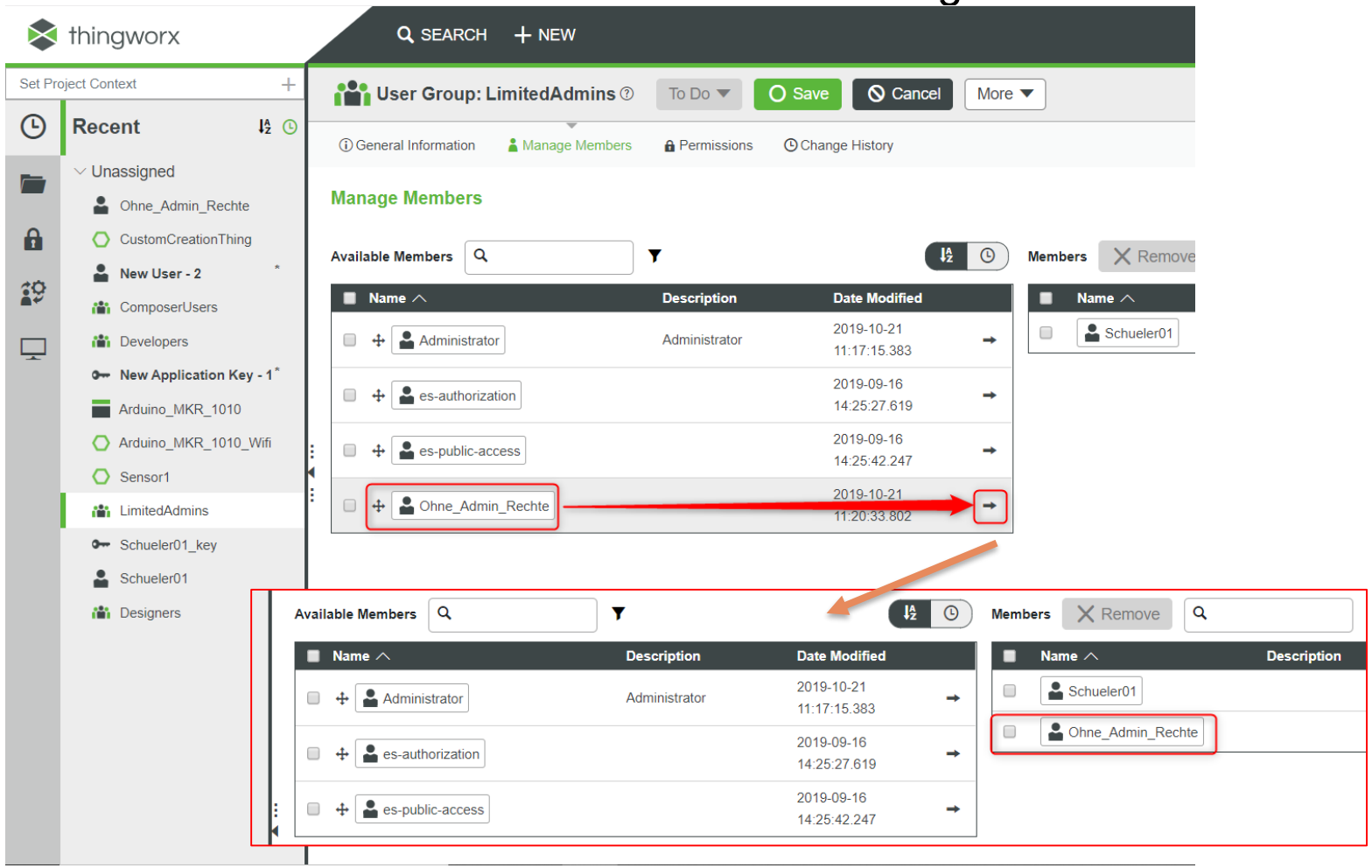




Neuer Thingworx Server

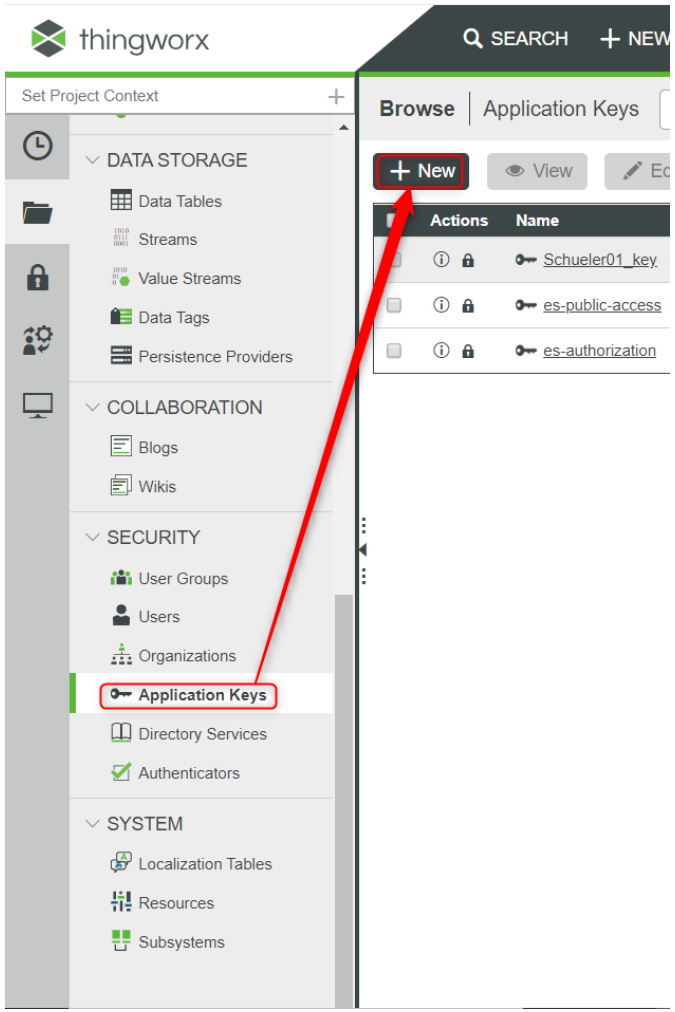
Neuen User „ohne Admin Rechte“ zur Gruppe LimitedAdmins hinzufügen

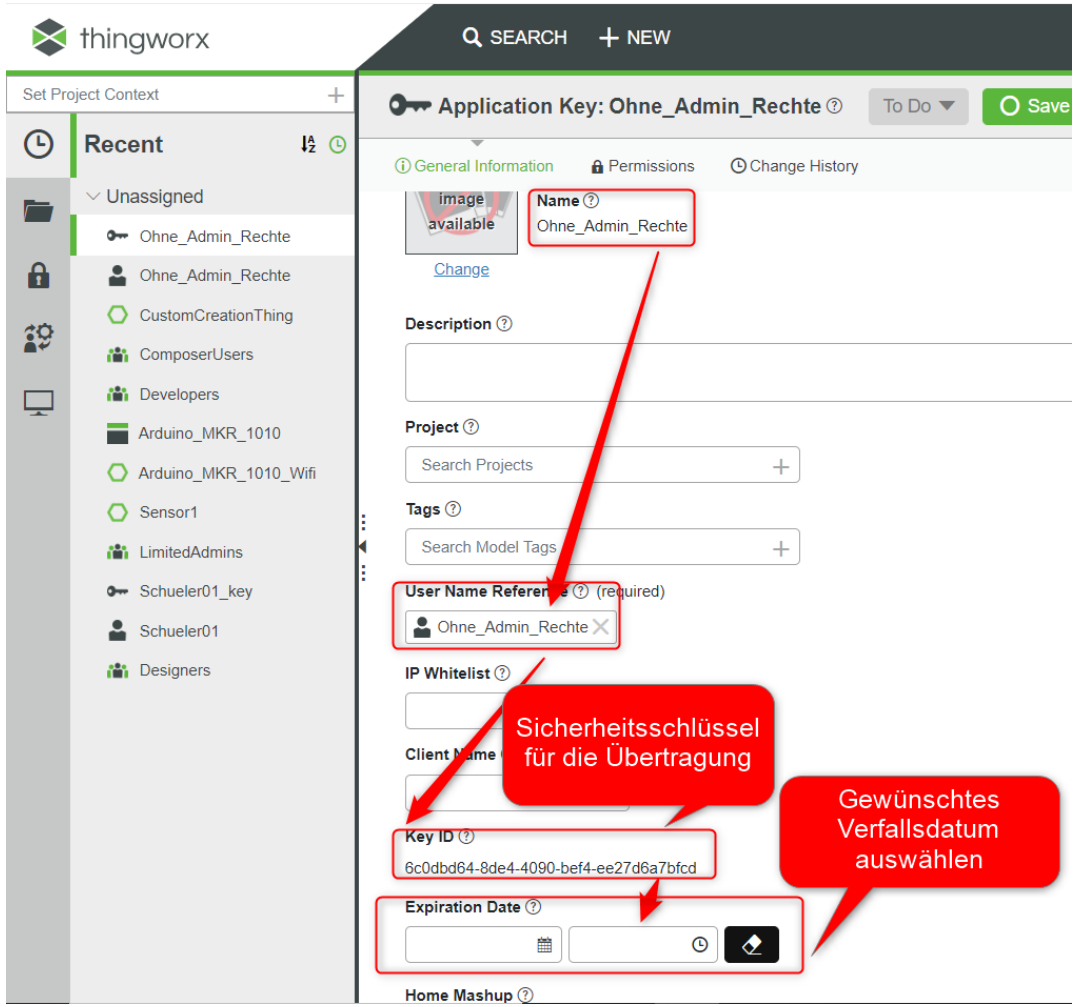




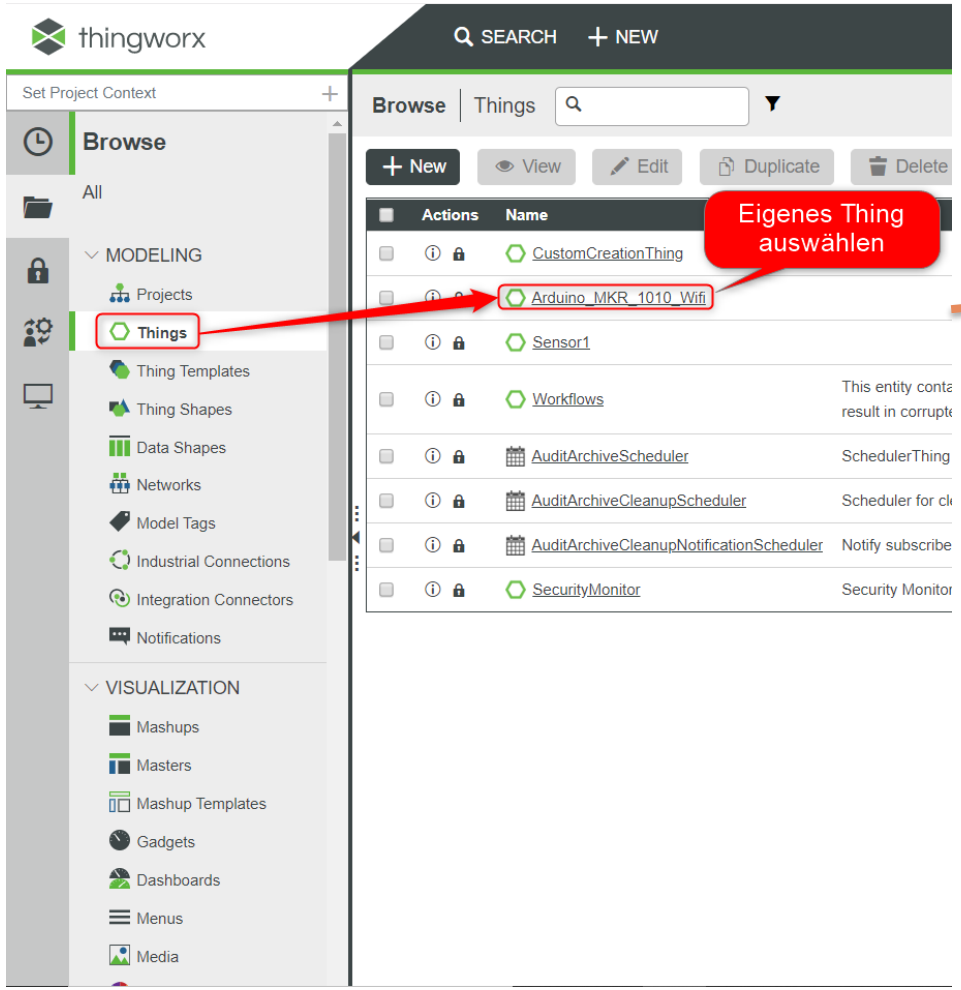
Neuer Thingworx Server

Sicherheitsschlüssel für den neuen User „ohne Admin Rechte“ erstellen

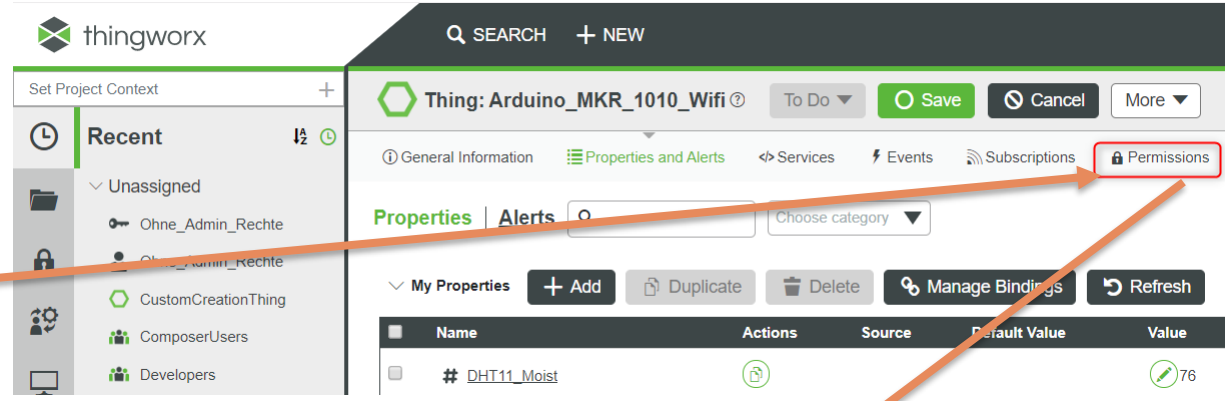


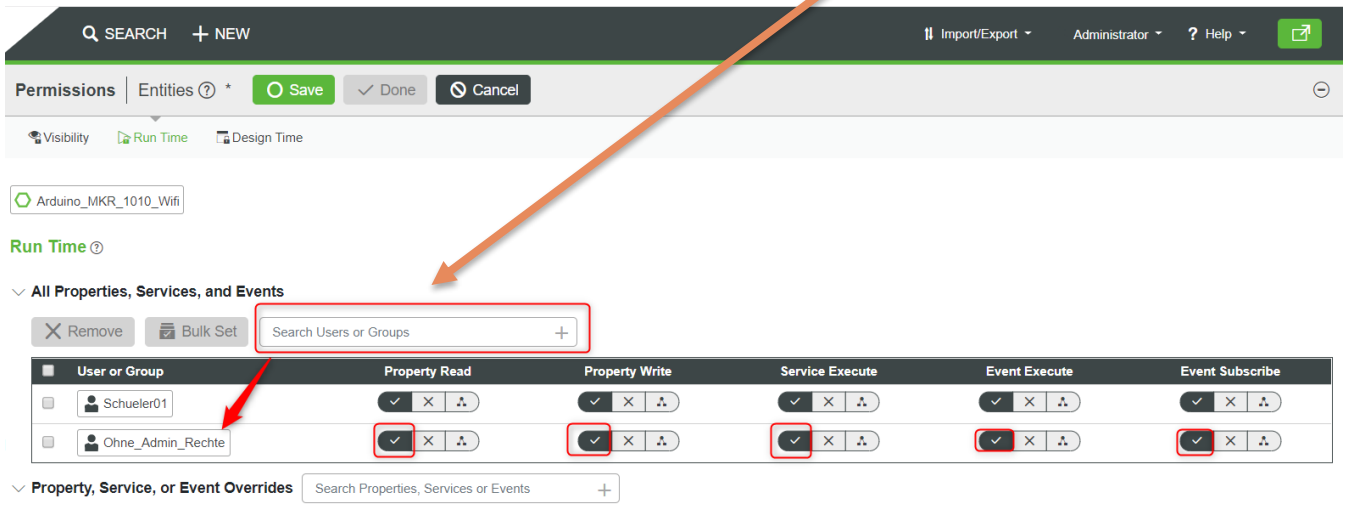


Neuer Thingworx Server



Dem User die erforderlichen Befugnisse für den Zugriff auf ein Thing erteilen





Inbetriebnahme alte Hardware

Einstellungen in der Thingworx_MKRWifi1010_Variable.h Library

```
const unsigned long TPOST = 5000;  //Time between requests to TWX server (every 5 secs)

//Wifi - Variables
//char* ssid =  //WiFi SSID
//char* password = ; //WiFi Pass
char* ssid = "[redacted]"; //WiFi SSID
char* password = "[redacted]"; //WiFi Pass

//Host Thingworx
char* host = "[redacted].twx.htl.schule";  //TWX Host for [redacted]hout http at beginning)
unsigned int port = 443; //TWX host port for https

//Thingworx Variables
//char appKey[] = "c311cc04-b617-47ca-b2cd-584610ee5cbc"; //API Key from TWX
char appKey[] = "[redacted]";
char thingName[] = "Arduino_MKR_1010_Wifi";[redacted] TWX
//->Timing Vars
unsigned long lastConnectionTime = 0; //Last connection ms time between server requests

#endif
```

Wifi Einstellungen

Server Einstellungen

Sicherheitsschlüssel

Thing- Name

Inbetriebnahme alte Hardware

- Selbstständige Inbetriebnahme der alten Schulungshardware auf neuem Server und mit MKR Wifi 1010 Controller
- Präsentation von letzter Schulung zur Hilfe nehmen!
- Schritte:
  - Richtiges Anschließen der Komponenten (siehe Folien vorher)
  - Arduino IDE konfigurieren (siehe vorher)
  - Server konfigurieren (siehe vorher)
  - Ein Thing und die zugehörigen Properties erstellen (siehe .ppt letzte Schulung)
  - In der Arduino IDE die Examples der Bibliothek durchprobieren

Einbindung von neuem Sensor

- Um gängige Sensoren am Arduino anzuschließen gibt es 2 Varianten:
  - „Wissenschaftliche“ Methode: Durchlesen des Datenblattes des Sensors und ausprogrammieren des benötigten Algorithmus.
  - „Praktische“ Methode: Google Suche verwenden.
- Da von den Teilnehmern keine Programmierkenntnisse gefordert sind, wird hier auf die praktische Methode eingegangen.
- Hier sollte nochmals erwähnt werden: Für das Realisieren eigener Projekte ist das Erlernen der Grundkenntnisse der Arduino Programmiersprache (C++) unumgänglich!

Einbindung von neuem Sensor

Praktische Methode

- Arduino ist ein oft verwendeter Mikrocontroller. Viele Millionen Menschen hantieren damit weltweit.
- Man findet daher im Internet viele Bedienungsanleitungen, Algorithmen und Schaltpläne zu oft verwendeten Sensoren.
- Der Abstands-Ultraschallsensor HC SR04 ist ein oft verwendetet Sensor.
- Eine gute Anleitung befindet sich unter:

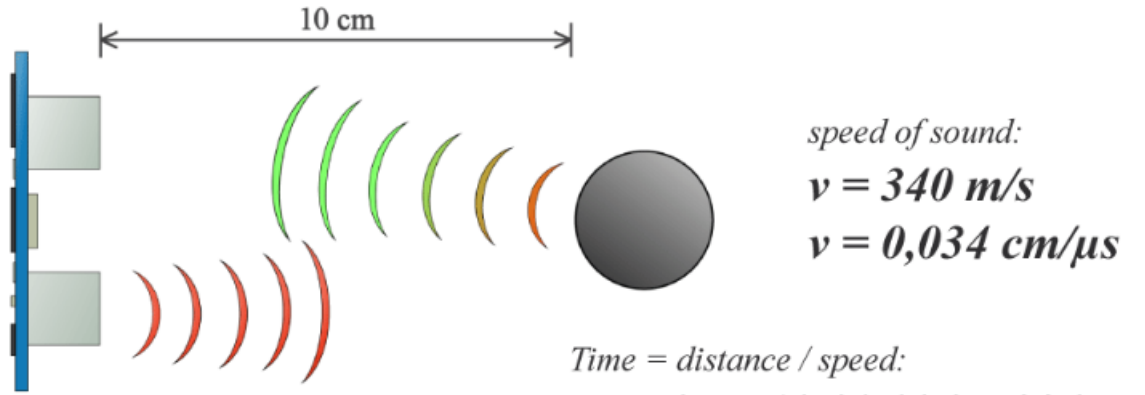
https://howtomechatronics.com/tutorials/arduino/ultrasonic-sensor-hc-sr04/

Einbindung von neuem Sensor

Wirkungsweise HC SR04

- Mit Hilfe des Wissens der Schallgeschwindigkeit und dem Messen der Zeit, kann eine Distanz s mit Hilfe der Formel s= v*t ermittelt werden.

- Sehen wir uns folgendes Beispiel an:



speed of sound:
$v = 340\ m/s$
$v = 0{,}034\ cm/\mu s$

Time = distance / speed:
$t = s / v = 10 / 0{,}034 = 294\ \mu s$

Distance:
$s = t \cdot 0{,}034 / 2$

Quelle: https://howtomechatronics.com/tutorials/arduino/ultrasonic-sensor-hc-sr04/

Einbindung von neuem Sensor

Verschaltung

- Auf das Steckbrett sollen 5V („+“) und GND („-“) mit Hilfe der Jumper Kabel geleitet werden.
- Der Port VCC des HC SR04 wird auf die 5V des Steckbrettes geleitet.
- Der Port GND des HC SR04 wird auf das GND des Steckbrettes geleitet.
- Der Port Trig des HC SR04 wird auf den digitalen Port D4 des Arduino MKR 1010 geleitet.
- Der Port Echo des HC SR04 wird auf den digitalen Port D5 des Arduino MKR 1010 geleitet.

Einbindung von neuem Sensor

- Es ist auch folgender Algorithmus angegeben:

- Bei den Code Beispielen der Thingworx_MKRWifi1010 Library befindet sich 006_PUT_Vorlage_Einbindung_Sensor

- Diese Vorlage kann für das Anbinden eines Sensors verwendet werden. Mit einem HTTP-Request wird der Wert zum Server geschickt.

- Die hier gezeigten Code Teile müssen nun an den richtigen Platz des Beispieles kopiert werden!

```
10.  const int trigPin = 9;
11.  const int echoPin = 10;
12.
13.  // defines variables
14.  long duration;
15.  int distance;
16.
17.  void setup() {
18.  pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output
19.  pinMode(echoPin, INPUT); // Sets the echoPin as an Input
20.  Serial.begin(9600); // Starts the serial communication
21.  }
22.
23.  void loop() {
24.  // Clears the trigPin
25.  digitalWrite(trigPin, LOW);
26.  delayMicroseconds(2);
27.
28.  // Sets the trigPin on HIGH state for 10 micro seconds
29.  digitalWrite(trigPin, HIGH);
30.  delayMicroseconds(10);
31.  digitalWrite(trigPin, LOW);
32.
33.  // Reads the echoPin, returns the sound wave travel time in microseconds
34.  duration = pulseIn(echoPin, HIGH);
35.
36.  // Calculating the distance
37.  distance= duration*0.034/2;
38.
39.  // Prints the distance on the Serial Monitor
40.  Serial.print("Distance: ");
41.  Serial.println(distance);
42.  }
```

Quelle: https://howtomechatronics.com/tutorials/arduino/ultrasonic-sensor-hc-sr04/

Einbindung von neuem Sensor

VORLAGE 006_PUT_Vorlage_Einbindung_Sensor

006_PUT_Vorlage_Einbindung_Sensor

```
//Definition of used Libraries
#include "Thingworx_MKRWifi1010.h"
#include "Thingworx_MKRWifi1010_Variable.h"

//Definitions of variables for the sensor

float sensorvalue=0;

// Define Thingworx Class (1 per Thing)
ThingWorx myThing(host, port, appKey, thingName);

void setup() {
  //Definition of used Pins by the Sensor


  //
  Serial.begin(9600);                    //Serial communications with computer at 9600 bauds for debug purposes
  myThing.Wifi(ssid, password);          //Start the Wifi Connection

}

void loop() {
  if (millis() - lastConnectionTime > TPOST)      //Send request to server every TPOST seconds
  {
    //Code for the sensor, the value should be stored in the variable "sensorvalue"


    //Send data with PUT Request to Thingworx
    myThing.put("Sensorvalue_Thingworx",sensorvalue);  //Send distance to server platform

    lastConnectionTime = millis();               //Refresh last connection time for if
  }
}
```

Variablen für den Code

Einstellungen für den Code

Code für den Sensor

```
10.  const int trigPin = 9;
11.  const int echoPin = 10;
12.
13.  // defines variables
14.  long duration;
15.  int distance;
16.
17.  void setup() {
18.  pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output
19.  pinMode(echoPin, INPUT); // Sets the echoPin as an Input
20.  Serial.begin(9600); // Starts the serial communication
21.  }
22.
23.  void loop() {
24.  // Clears the trigPin
25.  digitalWrite(trigPin, LOW);
26.  delayMicroseconds(2);
27.
28.  // Sets the trigPin on HIGH state for 10 micro seconds
29.  digitalWrite(trigPin, HIGH);
30.  delayMicroseconds(10);
31.  digitalWrite(trigPin, LOW);
32.
33.  // Reads the echoPin, returns the sound wave travel time in microseconds
34.  duration = pulseIn(echoPin, HIGH);
35.
36.  // Calculating the distance
37.  distance= duration*0.034/2;
38.
39.  // Prints the distance on the Serial Monitor
40.  Serial.print("Distance: ");
41.  Serial.println(distance);
42.  }
```

Quelle: https://howtomechatronics.com/tutorials/arduino/ultrasonic-sensor-hc-sr04/

Einbindung von neuem Sensor

- Es entsteht folgender Code:
- Dieser Code ist als Nr. 005 In der Bibliothek vorhanden.
- Der Algorithmus wird hier beschrieben.

005_PUT_HC_SR04_Value

```
#include "Thingworx_MKRWifi1010.h"
#include "Thingworx_MKRWifi1010_Variable.h"
```

Es werden die benötigten Libraries eingebunden

```
//Definitions of variables for the sensor
const int trigPin = 4;  //Trigger Pin on digital port x
const int echoPin = 5;  //Echo Pin on digital port y
long duration;

float sensorvalue=0;
```

Der Trigger Pin befindet sich auf D4 und der Echo Pin auf D5. Es wird eine Variable „Duration“ und „sensorvalue“ erstellt.

```
// Define Thingworx Class (1 per Thing)
ThingWorx myThing(host, port, appKey, thingName);

void setup() {
  //Definition of used Pins by the Sensor
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  //
  Serial.begin(9600);                    //Serial communications with computer at 9600 bauds for debug purposes
  myThing.Wifi(ssid, password);          //Start the Wifi Connection

}

void loop() {
  if (millis() - lastConnectionTime > TPOST)      //Send request to server every TPOST seconds
  {
    //Code for the sensor, the value should be stored in the variable "sensorvalue"
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  // Sets the trigPin on HIGH state for 10 micro seconds
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  // Reads the echoPin, returns the sound wave travel time in microseconds
  duration = pulseIn(echoPin, HIGH);
  sensorvalue= duration*0.034/2;

    //Send data with PUT Request to Thingworx
    myThing.put("HC_SR04",sensorvalue);  //Send distance to server platform

    lastConnectionTime = millis();              //Refresh last connection time for if
  }
}
```

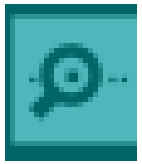
Der Pin 4 ist ein digitaler Output. Der Pin 5 ein digitaler Input

Benötigte Algorithmus für die Verwendung des HC SR04.

Der Wert der Variable „sensorvalue“ wird auf die Property „HC_SR04“ geschickt

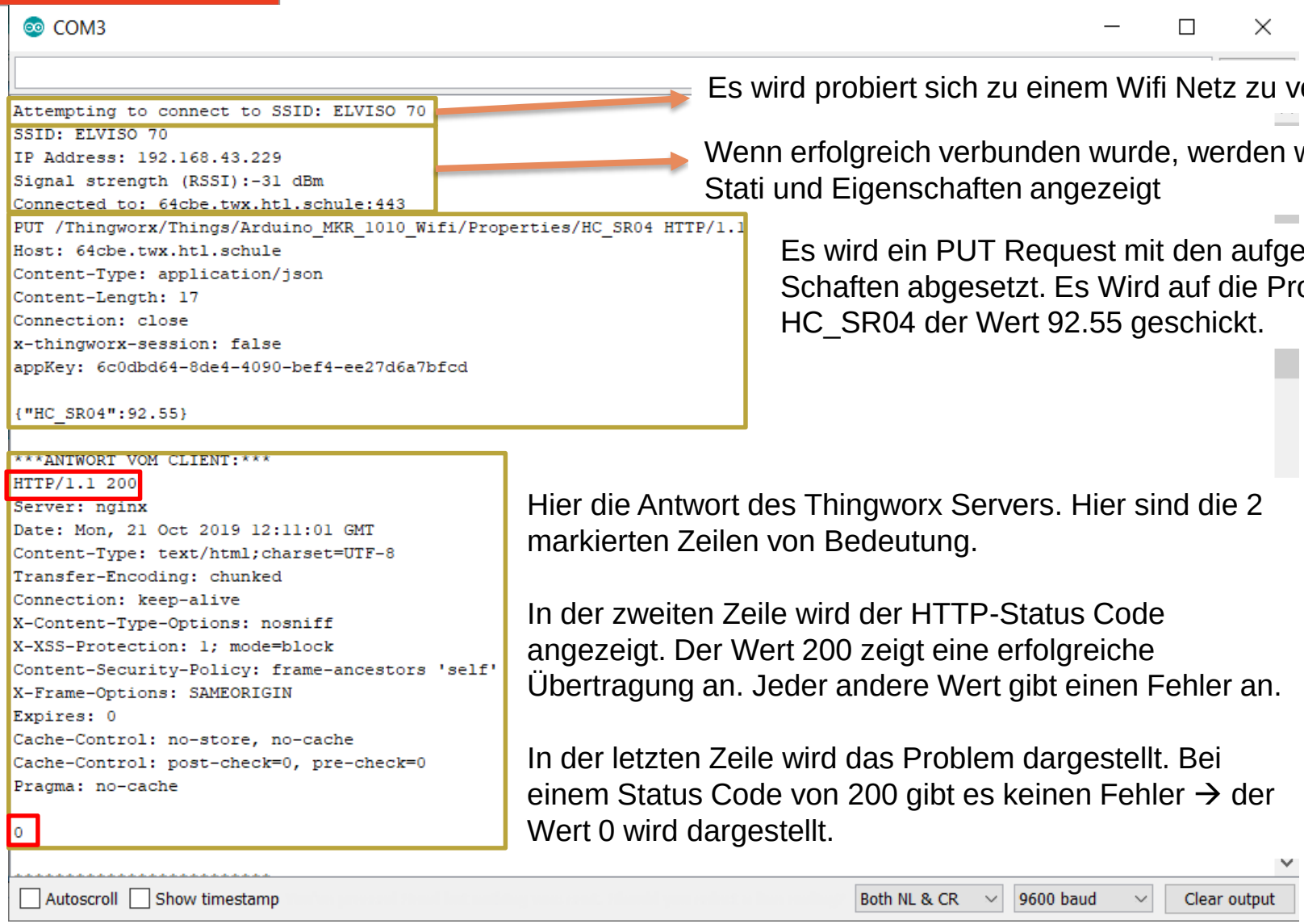
Einbindung von neuem Sensor

Ausgabe des seriellen Monitors

- Da der Arduino keine Anzeige hat, gibt es folgende Möglichkeit um den aktuellen Status auslesen zu können → der serielle Monitor.
- In der Arduino IDE wird rechts oben auf das Lupensymbol geklickt. 
- Wenn der Beispielcode 005 ausgeführt wird, dann ergibt sich folgende Ausgabe:

Einbindung von neuem Sensor

Ausgabe des seriellen Monitors



Es wird probiert sich zu einem Wifi Netz zu verbinden

Wenn erfolgreich verbunden wurde, werden wichtige Stati und Eigenschaften angezeigt

Es wird ein PUT Request mit den aufgeführten Eigen-Schaften abgesetzt. Es Wird auf die Property HC_SR04 der Wert 92.55 geschickt.

Hier die Antwort des Thingworx Servers. Hier sind die 2 markierten Zeilen von Bedeutung.

In der zweiten Zeile wird der HTTP-Status Code angezeigt. Der Wert 200 zeigt eine erfolgreiche Übertragung an. Jeder andere Wert gibt einen Fehler an.

In der letzten Zeile wird das Problem dargestellt. Bei einem Status Code von 200 gibt es keinen Fehler → der Wert 0 wird dargestellt.

Einbindung von neuem Sensor

Beispiele Fehler

- Falscher Propertyname

```
***ANTWORT VOM CLIENT:***
HTTP/1.1 404
```

```
6b
Unable To Write HC_SR05 on Arduino_MKR_1010_Wifi - Invalid Property &#x3a; HC_SR05 in Arduino_MKR_1010_Wifi
0
```

- Falscher Server

```
***ANTWORT VOM CLIENT:***
HTTP/1.1 503 Service Unavailable
```

```
<html><body><b>Http/1.1 Service Unavailable</b></body> </html>**************************
```

- Falscher Sicherheitsschlüssel

```
***ANTWORT VOM CLIENT:***
HTTP/1.1 401
```

- Link für HTTP Status-Tabelle: https://de.wikipedia.org/wiki/HTTP-Statuscode

Hausübung Drehdecoder

- Bis zum 3. Teil der Schulung (März 2020) soll der kodierte Drehschalter KY-040 selbst in Betrieb genommen werden.

- Auf der Thingworxplattform sollen folgende Properties erstellt und angezeigt werden:
  - Clockwise
  - Counterclockwise

    →Wird gegen oder in den Uhrzeigersinn gedreht?
  - Position

    →Anzeige der aktuellen Position (dabei wird in den Uhrzeigersinn ein Schritt als +1 positiv und gegen den Uhrzeigersinn als -1 negativ gewertet).

- Es wird Anfang Februar eine Erinnerungs- und Hilfestellungsmail von mir geschickt!