

ARDUINO – THINGWORX COMPOSER SCHULUNG

OKTOBER 2021

VORTRAGENDE:

DIPL. ING. MARTIN SCHUBERT (HTL WIENER NEUSTADT, SUM@HTLWRN.AC.AC)

ING. ARMIN FISCHER, M.SC. (HTL WIENER NEUSTADT, FIA@HTLWRN.AC.AC)

VERSION 4.0

EINFÜHRUNG ARDUINO

WAS IST ARDUINO?

- Elektronische Open Source Plattform (Hardware und Software)
- Mikrocontroller mit In- und Outputs
- Verschiedene Typen erhältlich (Inputs, Outputs, Schnittstellen, Stromverbrauch, ...)
- Eigene Arduino Programmiersprache (ähnlich C/C++)
- Einsteigerfreundlich
- Riesige Arduinocommunity mit Projekten erhältlich
- Offizielle Homepage <https://www.arduino.cc/>

WAS IST ARDUINO?

Arduino Uno



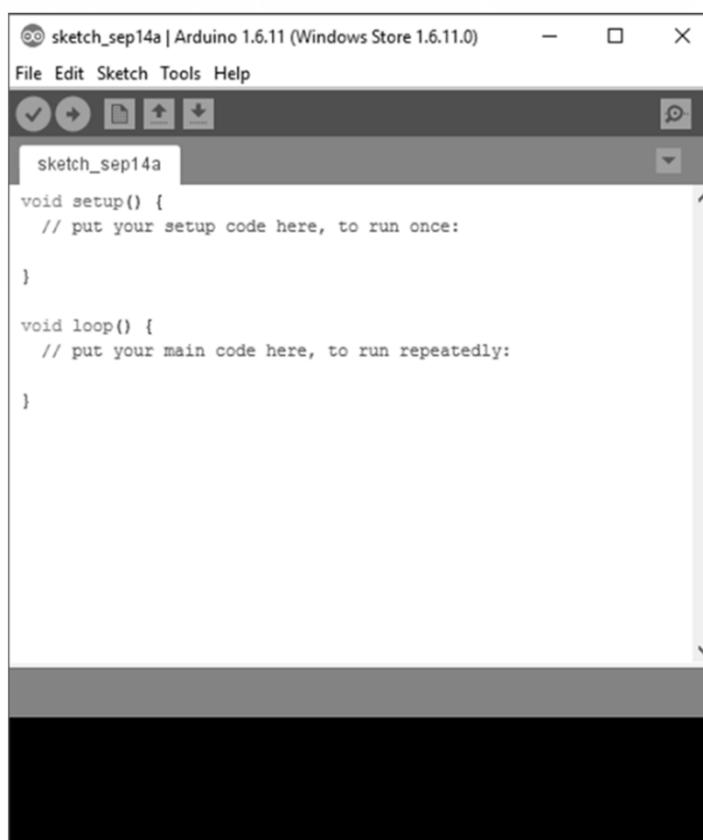
Arduino Mega 2560



Arduino Pro Mini



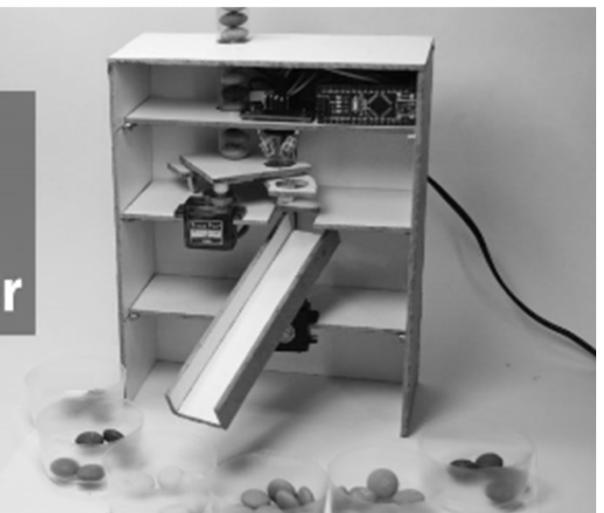
Arduino Leonardo with Headers



The image shows a screenshot of the Arduino IDE interface. The title bar reads "sketch_sep14a | Arduino 1.6.11 (Windows Store 1.6.11.0)". The menu bar includes "File", "Edit", "Sketch", "Tools", and "Help". The main area displays the following C++ code:

```
sketch_sep14a
void setup() {
  // put your setup code here, to run once:
}

void loop() {
  // put your main code here, to run repeatedly:
}
```



SCHULUNGSMATERIAL

ARDUINO MKR1010



- Stromsparender Mikrocontroller für IOT Anwendungen
- Stromversorgung 5V (USB)
- Wifi Anbindung
- 7 Analoge (0-3.3V) und 8 digitale Pins (LOW-0V & HIGH-3.3V)
- ACHTUNG: Pins dürfen nur bis 3.3V belastet werden
- Weitere Informationen siehe <https://store-usa.arduino.cc/products/arduino-mkr-wifi-1010>

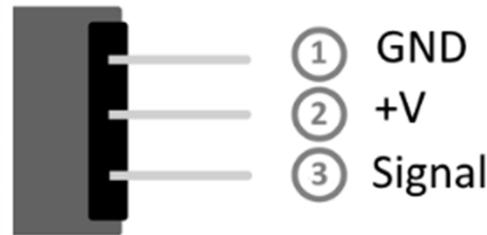
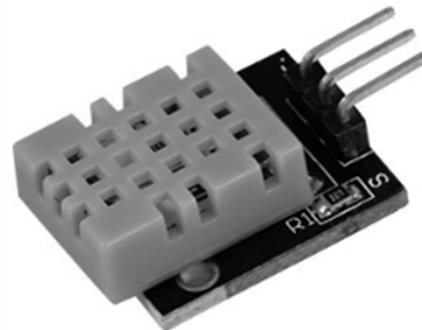
ARDUINO MKR RELAY PROTO SHIELD



- MKR1010 aufsteckbar
- 2 Relays mit digitalem Pin 1 und 2 verbunden (D1 & D2)
- Klemmen um A1-A4 leicht zu verbinden
- Eigener Bereich um Sensoren, Schalter, LED's usw. selbst zu applizieren.
- Weitere Informationen siehe <https://store.arduino.cc/mkr-relay-proto-shield>

DHT11 TEMPERATUR- UND FEUCHTIGKEITSSENSOR

- Messung von Temperatur 0-50°C
- Messung von Luftfeuchtigkeit 20-90%
- Betriebsspannung 3.3-5V
- Eingebauter 1kOhm Widerstand
- Abtastrate maximal alle 2 Sekunden
- Pinbelegung
- Weitere Informationen siehe <https://www.conrad.at/de/p/makerfactory-mf-6402159-sensor-modul-1-st-2134053.html>



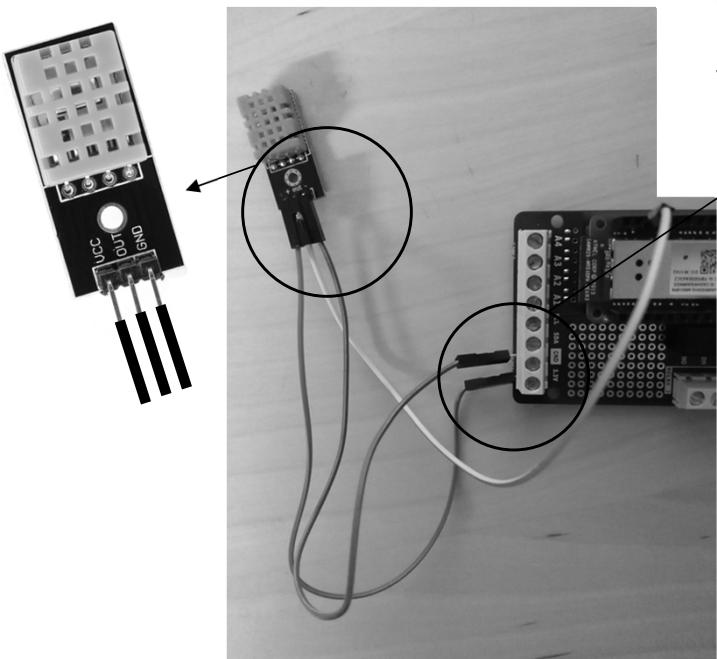
VENTILATOR



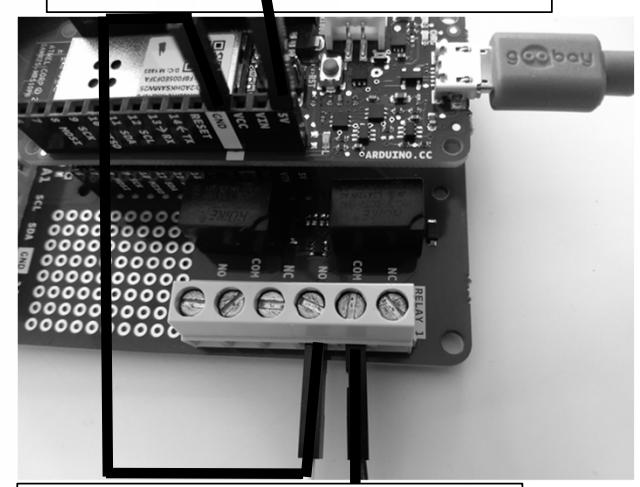
- Betriebsspannung 5VDC
- Weitere Informationen siehe <https://www.conrad.at/de/p/sbc-fan-303010-aktiver-luefter-passend-fuer-entwicklungskits-raspberry-pi-rock-pi-banana-pi-schwarz-1720600.html>

VERBINDEN DER EINZELTEILE

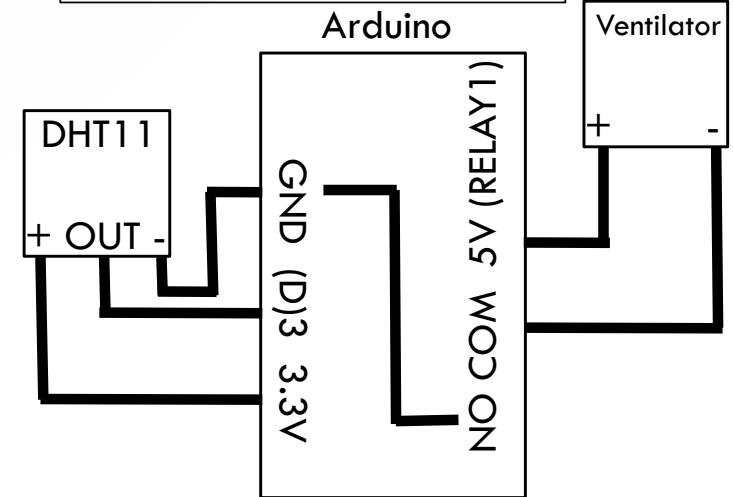
DHT11 mit Arduino Shield



Ventilator



Verbindungsplan



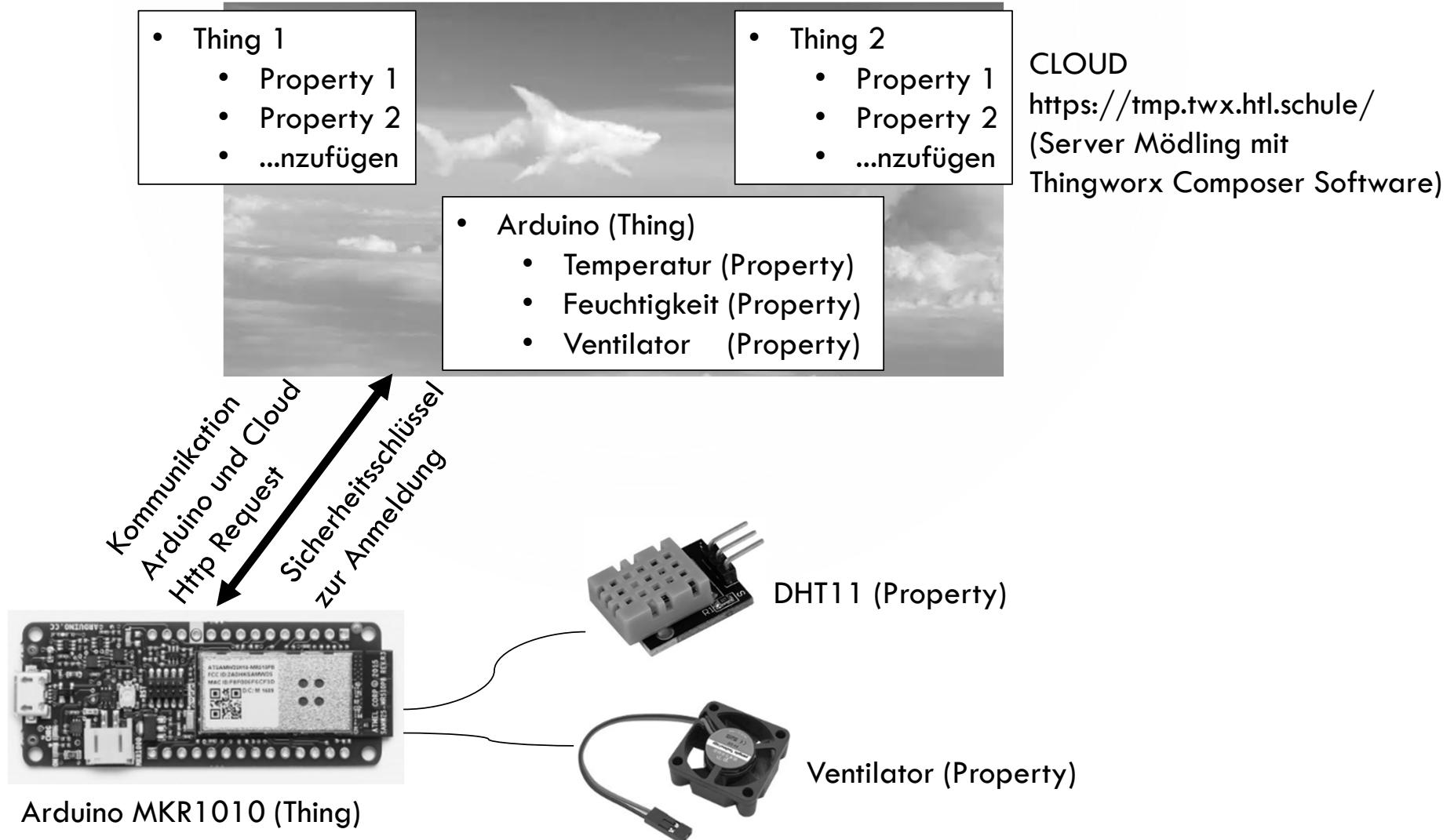
AUFGABENSTELLUNG

AUFGABENSTELLUNG

- Messen der Raumtemperatur und -feuchtigkeit mittels DHT11 Sensor
- Übertragen der Daten in die Cloud
- In der Cloud befindet sich die Software Thingworx Composer
- Zuordnen der übertragenen Daten in Thingworx Composer
- Einschalten des Ventilators über das Internet
- Einbindung der Sensordaten in Vuforia Studio

DATENÜBERTRAGUNG

WIE WERDEN DATEN IN DIE CLOUD ÜBERTRAGEN?



HTTP REQUEST

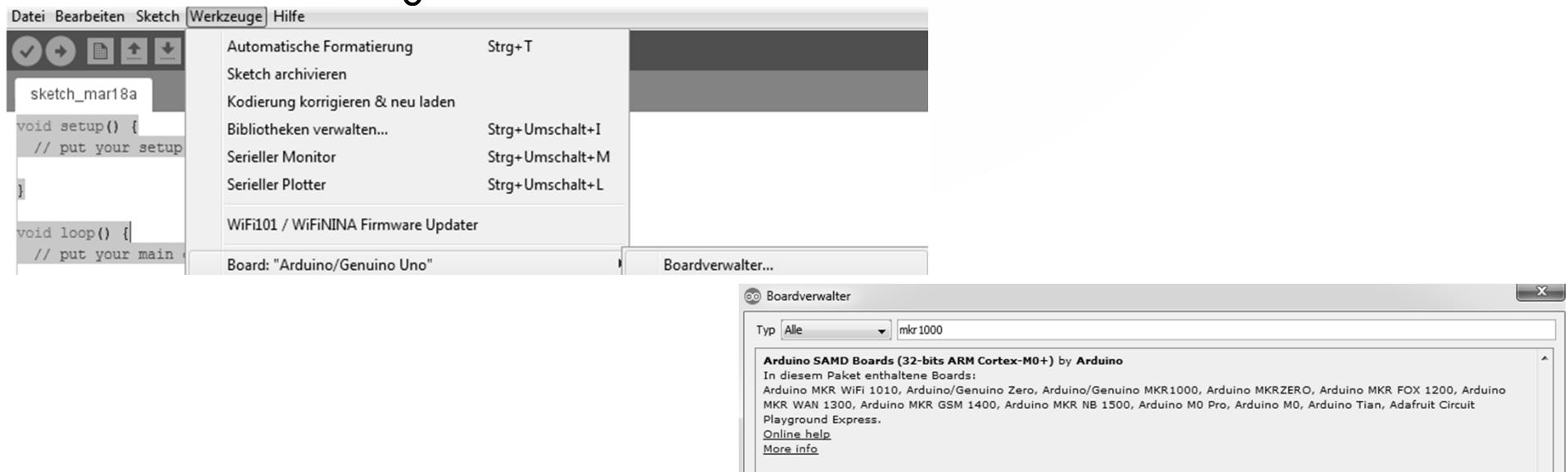
- Protokoll zum Übertragen von Daten (genauso wie im Browser)
- 1 HTTP Request um 1 Sensorwert (Property) vom Arduino (Thing) auf die Cloud zu übertragen
- HTTP Request enthält Sicherheitsschlüssel um eine automatische Authentifizierung/Anmeldung zur Cloud zu erhalten
- Beispiel

Thingworx Server Adresse	Thing	Property
<u>https://tmp.twx.htl.schule/Thingworx/Things/304417_FIA_IOTSchulung_Dummy/Properties/Sensorwert1</u>		

REALISIERUNG DER DATENÜBERTRAGUNG MIT ARDUINO

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



INSTALLATION NÖTIGER LIBRARIES

- Libraries = Vorprogrammierte Codes, welche mit Befehlen aufgerufen werden
- Benötige Libraries siehe
https://github.com/Snorp84/DigiPro_IOT_Arduino_Schulung/tree/main/libraries

THINGWORX MKR1010 LIBRARY

 examples
 Thingworx_MKRWifi1010
 Thingworx_MKRWifi1010
 Thingworx_MKRWifi1010_Variable
 version

- Library von Armin Fischer geschrieben
- Auf Schulungs USB Stick gespeichert (Thingworx_MKRWifi1010)
- 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

THINGWORX MKR1010 LIBRARY - EINSTELLUNGEN

- Die folgenden Informationen müssen in der Datei Thingworx_MKR1010_Variable.h eingestellt werden

```
////////////////////////////////////////////////////////////////////////  
//*****TIMING VARIABLES*****  
//*****  
const unsigned long TPOST = 2000; //Time between requests to TWX server (every 2 sec)  
unsigned long lastConnectionTime = 0; //Last connection ms time between server requests  
  
////////////////////////////////////////////////////////////////////////  
//*****WIFI CONNECTION*****  
//*****  
  
char* ssid = ""; //WiFi SSID  
char* password = ""; //WiFi Pass  
  
////////////////////////////////////////////////////////////////////////  
//*****HOST THINGWORX*****  
//*****  
char* host = "xxxxx.twx.htl.schule"; //TWX Host for HTL Austria twx.htl.schule (without http at beginning)  
unsigned int port = 443; //TWX host port for https  
  
////////////////////////////////////////////////////////////////////////  
//*****THINGWORX VARIABLES*****  
//*****  
char appKey[] = "";  
char thingName[] = "RC-Car"; //Thing name from TWX  
String property_list[]={ "Temperatur", "Feuchtigkeit", "Drehzahl", "Strom", "Distanz", "Pitch", "Roll", "Beschleunigung_x"
```

THINGWORX MKR1010 LIBRARY - EINSTELLUNGEN

- Für die ersten Beispiele werden folgende Einstellungen verwendet
 - SSID und Passwort von Schulnetzwerk (oder HOTSPOT von Handy)
 - Voreingestellten Sicherheitsschlüssel (kann sich durch Serverumstellung verändern)
 - Thingname 304417_FIA_IOTSchulung_Dummy

THINGWORX MKR1010 LIBRARY – BEISPIELE

- Um die Benutzung der Library besser zu verstehen

werden vorgefertigte Programme verwendet:

- 001_PUT_Value
- 002_PUT_DHT11_Value
- 003_GET_Value
- 004_GET_Relay



THINGWORX MKR1010 LIBRARY

003_GET_VALUE – ABFRAGEN EINES WERTES

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

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

//Variable for Sensor Values
float sensor_1;

void setup() {
    Serial.begin(9600);           //Serial communications with computer at 9600 bauds for debug purposes
    myThing.Wifi();               //Start the Wifi Connection
}

void loop() {
    if (millis() - lastConnectionTime > TPOST)      //Send request to server every TPOST seconds
    {
        sensor_1=myThing.getjson("Sensorwert_1");      //Get data with GET Request from Thingworx

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

Einfügen der gebrauchten Libraries

THINGWORX MKR1010 LIBRARY

003_GET_VALUE – ABFRAGEN EINES WERTES

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

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

//Variable for Sensor Values
float sensor_1;

void setup() {
    Serial.begin(9600);                                //Serial communications with computer at 9600
    myThing.Wifi();                                     //Start the Wifi Connection
}

void loop() {
    if (millis() - lastConnectionTime > TPOST)          //Send request to server every TPO
    {
        sensor_1=myThing.getjson("Se Sensorwert_1");      //Get data with GET Request from T
        lastConnectionTime = millis();                      //Refresh last connection time for if
    }
}
```

Erstellen einer Thingworx Klasse:

- Die Klasse verhält sich wie eine Variable
- In der Klasse sind die verschiedenen Library Funktionen enthalten (Verbinden mit WLAN, schicken und abfragen eines Wertes)
- Die Klasse heißt hier „myThing“ und kann beliebig verändert werden
- Die Klasse benötigt die Variablen host, port, appKey, thingname, property_list, ssid und password von der Datei Thingworx_MKR1010_Variable

THINGWORX MKR1010 LIBRARY

003_GET_VALUE – ABFRAGEN EINES WERTES

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

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

//Variable for Sensor Values
float sensor_1;

void setup() {
    Serial.begin(9600);
    myThing.Wifi();
}

void loop() {
    if (millis() - lastConnectionTime > TPOST)          //Send request to server every TPOST seconds
    {
        sensor_1=myThing.getjson("Se Sensorwert_1");      //Get data with GET Request from Thingworx

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

Die void `setup()` Schleife ist Arduino spezifisch. Die darin enthaltenen Befehle werden einmalig beim Starten des Arduinos aufgerufen.

`Serial.begin(9600)`: Es wird der serielle Monitor (Ausgabefenster am PC) aktiviert.

`myThing.Wifi`: Es wird eine Kommunikation mit dem WLAN aufgebaut.

THINGWORX MKR1010 LIBRARY

003_GET_VALUE – ABFRAGEN EINES WERTES

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

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

//Variable for Sensor Values
float sensor_1;

void setup() {
    Serial.begin(9600);                                //Serial communications with Thingworx
    myThing.Wifi();                                     //Start the Wifi Connection
}

void loop() {
    if (millis() - lastConnectionTime > TPOST)          //Send request
    {
        sensor_1=myThing.getjson("Sensorwert_1");         //Get data with GET Request from Thingworx
        lastConnectionTime = millis();                     //Refresh last connection time for if
    }
}
```

Die void loop() Schleife wird endlos ausgeführt.

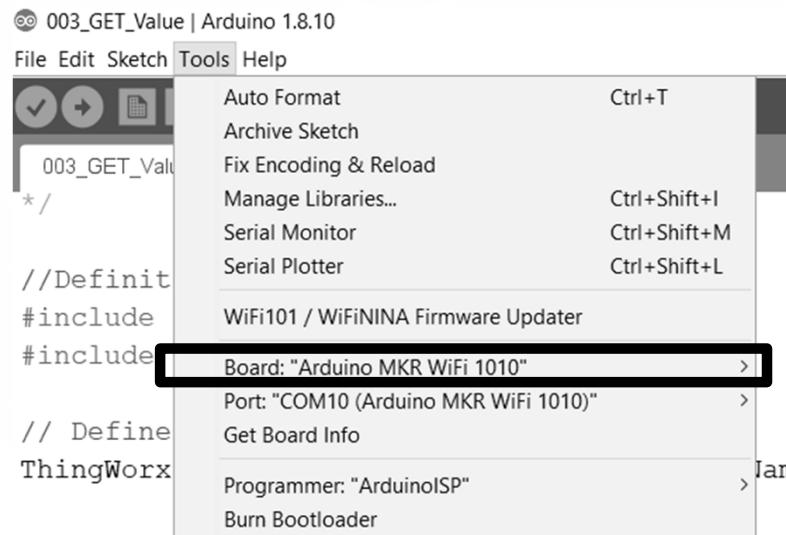
If-Abfrage: Es wird alle TPOST (Variable in der Datei Thingworx_MKR1000_Variable.h) Sekunden ein Wert abgefragt.

Sensor_1=myThing.getjson();
Es wird in die Variable sensor_1 mit dem Befehl myThing.getjson (in myThing ist schon der Thingname enthalten!) die Property „Sensorwert_1“ abgerufen

THINGWORX MKR1010 LIBRARY

003_GET_VALUE – ABFRAGEN EINES WERTES

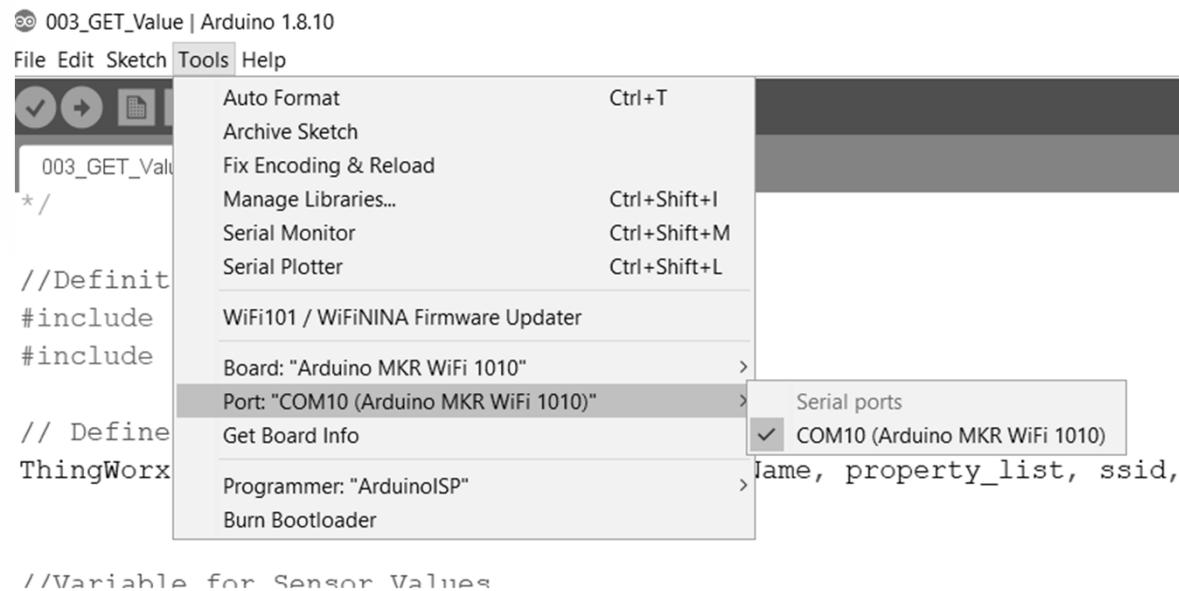
- Einstellung des MKR1010 Mikrocontrollers



THINGWORX MKR1010 LIBRARY

003_GET_VALUE – ABFRAGEN EINES WERTES

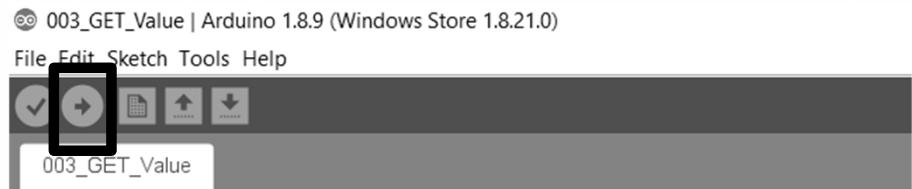
- Einstellung des richtigen Port (Darstellung kann abweichen!)



THINGWORX MKR1010 LIBRARY

003_GET_VALUE – ABFRAGEN EINES WERTES

- Kompilierung (Übersetzung des Programmes in die Sprache des Mikrocontrollers) und Upload des Programmes



```
003_GET_Value | Arduino 1.8.9 (Windows Store 1.8.21.0)
File Edit Sketch Tools Help
  ↗  ↘  ⌂  ⌃  ⌄  ⌅
003_GET_Value

/*
  Example for using an Arduino MKR1000 with a PTC Thingworx Server.
  A value from the Thingworx Server is obtained.

  PLEASE DEFINE ALL VARIABLES IN THE "Thingworx_MKR1000_Variable.h"

  Created by Armin Fischer, Jan 2019.
  School: TGM Vienna
  E-Mail: afischer2@tgm.ac.at
*/

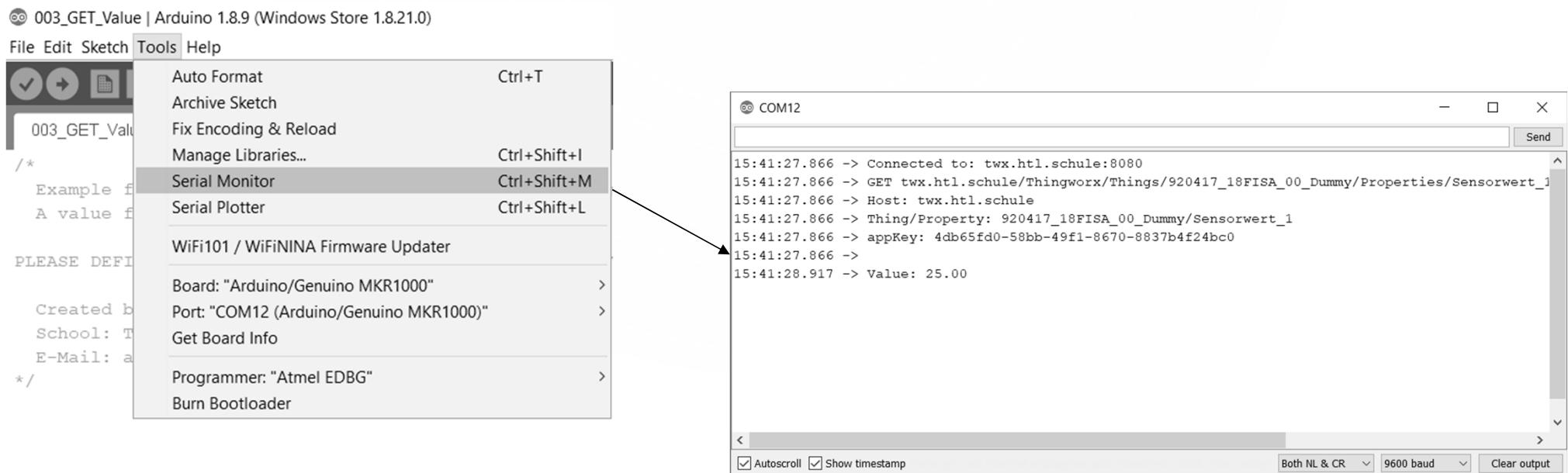
//Definition of used Libraries
#include "Thingworx_MKR1000.h"
#include "Thingworx_MKR1000_Variable.h"

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

THINGWORX MKR1010 LIBRARY

003_GET_VALUE – ABFRAGEN EINES WERTES

- Einschalten des Serial Monitor um den Ablauf des Programmes nachzuvollziehen



THINGWORX MKR1010 LIBRARY

003_GET_VALUE – ABFRAGEN EINES WERTES

- Beim Empfangen der Werte werden folgende Zeilen ausgegeben. Die Ausgabe kann zum Finden von Fehlern verwendet werden

```
Attempting to connect to SSID: test-iot
Attempting to connect to SSID: test-iot
Attempting to connect to SSID: test-iot
SSID: test-iot
IP Address: 192.168.43.245
Signal strength (RSSI):-39 dBm
Connected to: tmp.twx.htl.schule:443
GET tmp.twx.htl.schule/Thingworx/Things/304417_FIA_IOTSchulung_Dummy/Properties/Sensorwert_1?appKey=d8d3e9d3-ba85-4da6-be7a-35a31ab3640e
Host: tmp.twx.htl.schule
Thing/Property: 304417_FIA_IOTSchulung_Dummy/Sensorwert_1
appKey: d8d3e9d3-ba85-4da6-be7a-35a31ab3640e
```

The diagram illustrates a callout from the log output on the left to the successful connection message on the right. A line with an arrow points from the 'Connected to:' line in the log to the 'Erfolgreiches Verbinden' message in the callout box.

Erfolgreiches Verbinden mit dem WLAN
„test-iot“.
Zugewiesene IP-Adresse ist
192.168.43.245

THINGWORX MKR1010 LIBRARY

003_GET_VALUE – ABFRAGEN EINES WERTES

```
Connected to: tmp.twx.htm.schule:443
GET tmp.twx.htm.schule/Thingworx/Things/304417_FIA_IOTSchulung_Dummy/Properties/Sensorwert_1?appKey=d8d3e9d3-ba85-4da6-be7a-35a31ab3640e
Host: tmp.twx.htm.schule
Thing/Property: 304417_FIA_IOTSchulung_Dummy/Sensorwert_1
appKey: d8d[REDACTED]
```

```
***ANTWORT VOM CLIENT:***
HTTP/1.1 200
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
Content-Security-Policy: frame-ancestors 'self'
X-Frame-Options: SAMEORIGIN
Expires: 0
Cache-Control: no-store, no-cache
Cache-Control: post-check=0, pre-check=0
Pragma: no-cache
Content-Type: application/json; charset=UTF-8
Transfer-Encoding: chunked
Date: Mon, 04 Oct 2021 13:40:34 GMT
Keep-Alive: timeout=20
Connection: keep-alive
d0
{"dataShape":{"fieldDefinitions":{"Sensorwert_0":
```

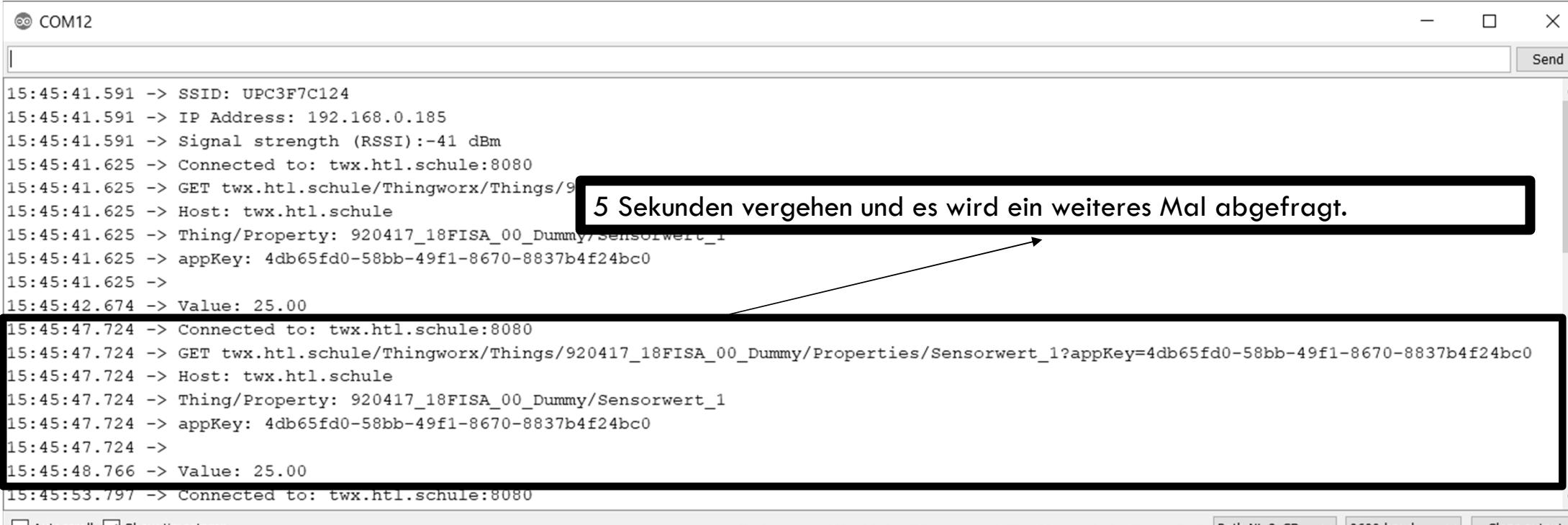
Value: 25.00

Mit dem Server tmp.twx.htm.schule am Port 443 verbunden.
Es wird ein GET Request geschickt mit der Adresse
tmp.twx.htm.schule/Thingworx/... geschickt.
Es wird das Thing 304417_FIA_IOTSchulung_Dummy angesprochen. Die
Property Sensorwert_1 wird abgefragt.
Der Sicherheitsschlüssel ist d8d...
Der abgefragt Wert beträgt 25

THINGWORX MKR1010 LIBRARY

003_GET_VALUE – ABFRAGEN EINES WERTES

- Beim Empfangen der Werte werden folgende Zeilen ausgegeben. Die Ausgabe kann zum Finden von Fehlern verwendet werden



```
COM12 - X
Send
15:45:41.591 -> SSID: UPC3F7C124
15:45:41.591 -> IP Address: 192.168.0.185
15:45:41.591 -> Signal strength (RSSI):-41 dBm
15:45:41.625 -> Connected to: twx.html.schule:8080
15:45:41.625 -> GET twx.html.schule/Thingworx/Things/920417_18FISA_00_Dummy/Sensorwert_1
15:45:41.625 -> Host: twx.html.schule
15:45:41.625 -> Thing/Property: 920417_18FISA_00_Dummy/Sensorwert_1
15:45:41.625 -> appKey: 4db65fd0-58bb-49f1-8670-8837b4f24bc0
15:45:41.625 ->
15:45:42.674 -> Value: 25.00
15:45:47.724 -> Connected to: twx.html.schule:8080
15:45:47.724 -> GET twx.html.schule/Thingworx/Things/920417_18FISA_00_Dummy/Properties/Sensorwert_1?appKey=4db65fd0-58bb-49f1-8670-8837b4f24bc0
15:45:47.724 -> Host: twx.html.schule
15:45:47.724 -> Thing/Property: 920417_18FISA_00_Dummy/Sensorwert_1
15:45:47.724 -> appKey: 4db65fd0-58bb-49f1-8670-8837b4f24bc0
15:45:47.724 ->
15:45:48.766 -> Value: 25.00
15:45:53.797 -> Connected to: twx.html.schule:8080
```

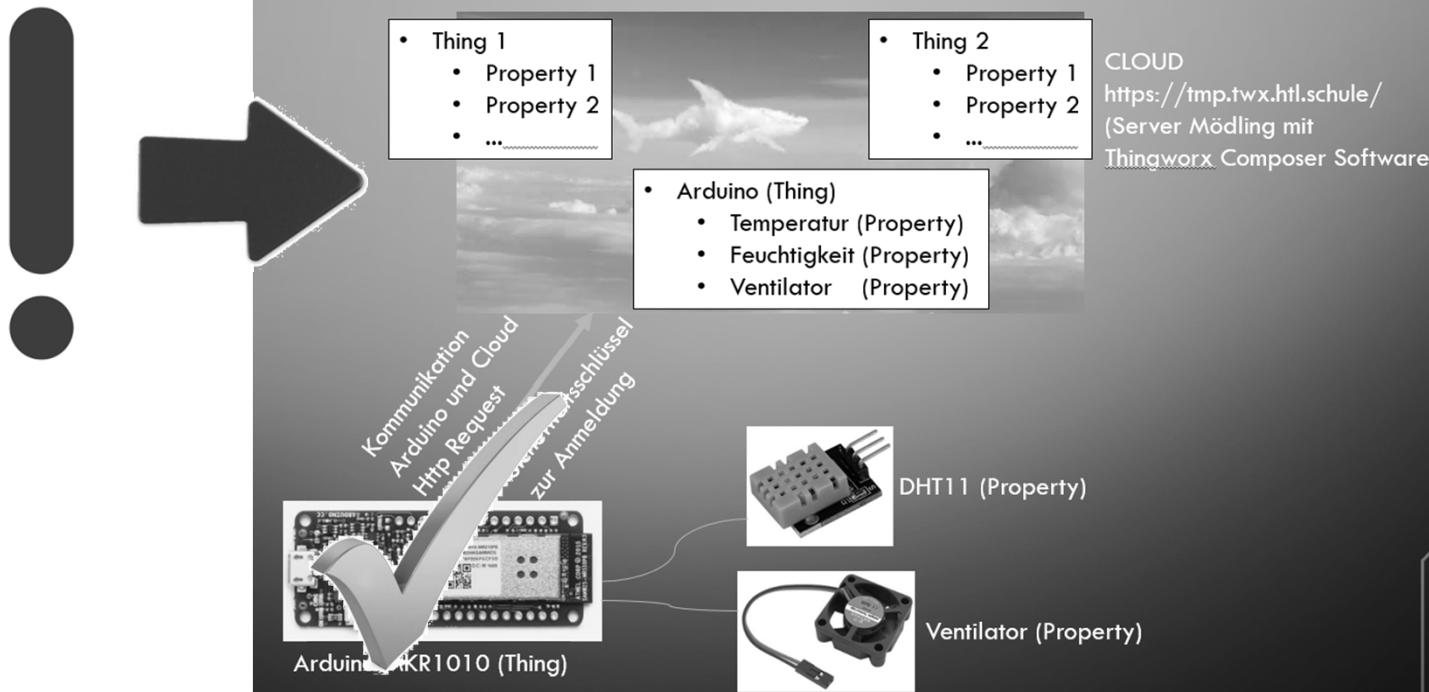
5 Sekunden vergehen und es wird ein weiteres Mal abgefragt.

Autoscroll Show timestamp Both NL & CR 9600 baud Clear output

THINGWORX COMPOSER

EINFÜHRUNG

- Bis jetzt → Zugriff auf ein vorgegebenes Thing.
- Für weitere Projekte/Anwendungen muss man jedoch eigene Things und eigene Properties dazu erstellen.
- Dies ist mit der Software Thingworx Composer am Server möglich.



THINGWORX COMPOSER

- Software zur Verwaltung/Steuerung von „Things“
- Zugriff unter <https://tmp.twx.h1.schule/>
- Zugriffsdaten
 - Benutzer: xxx
 - Passwort: yyy



OBERFLÄCHE

The screenshot shows the ThingWorx Composer interface. On the left is a sidebar with categories like MODELING, ANALYTICS, VISUALIZATION, etc. The central area shows a list of 'Things' with columns for View, Name, Description, and Modified. A callout box highlights the 'Things' list with the following text:

Auswahl der verschiedenen Funktionen.
Beispiele:

- Erstellung von Things
- Erstellung von Benutzeroberflächen „Mash-Ups“
- ...

View	Name	Description	Modified
<input type="checkbox"/>	dibse-raspi3		2019-03-19 16:38:07.780
<input type="checkbox"/>	die_thing		2019-03-19
<input type="checkbox"/>	920417_18FISA_00_Dummy		2019-03-19
<input type="checkbox"/>	Jenbach-Bonapace_Gruber		
<input type="checkbox"/>	Jenbach-Delic-Fanki		
<input type="checkbox"/>	dibse_raspi1		2019-03-13 10:50:43.113
<input type="checkbox"/>	cd-thing2		2019-03-13 10:15:27.540
<input type="checkbox"/>	dibse_raspi2		2019-03-11 08:54:45.692
<input type="checkbox"/>	910417_Waterbike		2019-03-08 11:32:58.015
<input type="checkbox"/>	910417_Raspberry	Daten des SensorHAT des Raspberry PI	2019-03-05 17:51:05.258
<input type="checkbox"/>	920417_18FISA_00_Library_Test	Test Thing from Armin Fischer Testing Library	2019-03-05 16:58:08.820
<input type="checkbox"/>	920417_18FISA_00_Lucky_Shield	TGM Lucky Shield	2019-03-01

OBERFLÄCHE

The screenshot shows the ThingWorx Composer interface. On the left, there is a sidebar with categories like MODELING, ANALYTICS, and VISUALIZATION. A callout box labeled "Bearbeitungsfenster für die ausgewählten Funktionen" points to the main content area. The main area displays a list of "Things" with columns for View, Name, Description, and Modified. The list includes various items such as "dibse-raspi3", "die_thing", and "920417_18FISA_00_Dummy".

View	Name	Description	Modified
<input type="checkbox"/>	dibse-raspi3		2019-03-19 16:38:07.780
<input type="checkbox"/>	die_thing		2019-03-19 13:00:13.018
<input type="checkbox"/>	920417_18FISA_00_Dummy		2019-03-18 13:20:12.654
<input type="checkbox"/>	Jenbach-Bonapace_Gruber		2019-03-15 11:10:19.828
<input type="checkbox"/>	Jenbach-Delic-Fanki		2019-03-15 08:58:32.754
<input type="checkbox"/>	dibse_raspi1		2019-03-13 14:17:40.059
<input type="checkbox"/>			2019-03-13 10:50:43.113
<input type="checkbox"/>	dibse_raspi2		2019-03-13 10:15:27.540
<input type="checkbox"/>	910417_Waterbike		2019-03-11 08:54:45.692
<input type="checkbox"/>	910417_Raspberry	Daten des SensorHAT des Raspberry PI	2019-03-08 11:32:58.015
<input type="checkbox"/>	920417_18FISA_00_Library_Test	Test Thing from Armin Fischer Testing Library	2019-03-05 17:51:05.258
<input type="checkbox"/>	920417_18FISA_00_Lucky_Shield	TGM Lucky Shield	2019-03-05 16:58:08.820
<input type="checkbox"/>			2019-02-01

ERSTELLUNG THING

The screenshot shows the ThingWorx Composer interface. In the top navigation bar, the URL is https://ttx.htl.schule/Thingworx/Composer/index.html. The main menu includes New Entity, Import/Export, Monitoring, Help, Learning Connector, and a user account (iot-seminar16). On the left sidebar under MODELING, the 'Things' option is selected and highlighted with a black box. A callout arrow points from this selection to the 'New' button in the toolbar above the list of existing Things. The list shows two items: 'dibse-raspi3' and 'die thing'. The 'die thing' entry has a timestamp of 2019-03-19 16:38:07.780. Below the list is a note: 'Showing: 148 items Modified 2019-03-19 16:38:07.780 2019-03-19'.

The bottom half of the screen displays the 'New Thing' creation dialog. The title bar says 'New Thing Thing'. The left sidebar lists ENTITY INFORMATION (Properties, Services, Events, Subscriptions, Home Mashup) and PERMISSIONS (Visibility, Design Time, Run Time). The main form is titled 'General Information'. It contains fields for Name (Schule_JahrKlasse_KatNr_Thingname), Description, Project (Search Projects), Tags (Search Model Vocabulary), Thing Template (GenericThing), and Implemented Shapes (Search Thing Shapes). A note 'Namenskonvention siehe nächste Folie' is overlaid on the form. The right side of the dialog shows various properties like Active (checked), Home Mashup (Search Mashups), Avatar (Change), Published (unchecked), Identifier (Browse...), Last Modified Date (No date and time selected), and Value Stream (Search Thing).

NAMENSKONVENTION VON THINGS

- Warum? : Viele verschiedene Schulen erstellen verschiedene Things. Zur Zeit sehen alle Benutzer alle Things. Wenn keine Konvention vorhanden ist, dann wird eine Zuordnung der vielen Things nicht möglich sein.
- Namensaufbau:
 - Schulkennzahl_SchuljahrJahrgangsnummer_Katalognummer_Thingname
- Beispiel:

Schüler des TGM im Jahr 2018/19 in der 5C mit der Katalognummer 10. Es wird eine Temperatursteuerung erstellt.

920417_185C_10_Temperatursteuerung

Lehrer des TGM im Jahr 2018/19. Das Kürzel in der Schule beträgt FISA. Es wird eine Temperatursteuerung erstellt.

920417_18FISA_00_Temperatursteuerung

ERSTELLUNG PROPERTY

The screenshot shows the ThingWorx Composer interface. On the left is a sidebar with categories like MODELING, ANALYTICS, VISUALIZATION, DATA STORAGE, COLLABORATION, and SECURITY. The central area is titled 'Things' and shows a list of entities. A callout box labeled 'Klick' points to the row for '920417_18FISA_00_Dummy'. Another callout box labeled 'Nächste Folie' points to the bottom right corner of the screen.

<input type="checkbox"/>	<input type="checkbox"/> View	Name	Description	Modified	<input type="checkbox"/>
	<input type="checkbox"/>	dibse-raspi3		2019-03-19 16:38:07.780	
	<input type="checkbox"/>	die_thing		2019-03-19 13:00:13.018	
	<input checked="" type="checkbox"/>	920417_18FISA_00_Dummy		2019-03-18 13:20:12.654	
	<input type="checkbox"/>	Jenbach-Bonapace_Gruber		2019-03-15 11:10:19.828	
	<input type="checkbox"/>	Jenbach-Delic-Fanki		2019-03-15 08:58:32.754	
	<input type="checkbox"/>	dibse_raspi1		2019-03-13 14:17:40.059	
	<input type="checkbox"/>	cd-thing2		2019-03-13 10:50:43.113	
	<input type="checkbox"/>	dibse_raspi2		2019-03-13 10:15:27.540	
	<input type="checkbox"/>	910417_Waterbike		2019-03-11 08:54:45.692	
	<input type="checkbox"/>	910417_Raspberry	Daten des SensorHAT des Raspberry PI	2019-03-08 11:32:58.015	
	<input type="checkbox"/>	920417_18FISA_00_Library_Test	Test Thing from Armin Fischer Testing Library	2019-03-05 17:51:05.258	
	<input type="checkbox"/>	920417_18FISA_00_Lucky_Shield	TGM Lucky Shield	2019-03-05 16:58:08.820	
				2019-03-01	

ERSTELLUNG PROPERTY

The screenshot shows a software interface for managing entity properties. At the top, there's a navigation bar with tabs for Entity Information, General Information, Properties, Services, Events, Subscriptions, Home Mashup, and a permission section. Below this is a 'General Information' panel for an entity named 'Schule_JahrKlasse_KatNr_Thingname'. The 'Properties' tab is selected in the sidebar.

In the main area, there's a 'Properties' section with a table titled 'My Properties'. A new row is being added, indicated by a 'New Property' button. An arrow points from this button to a 'General Property Info' form. This form includes fields for Name, Description, Category, and Alerts, along with buttons for Enabled?, Type, Config, Name, and Del.

Another arrow points from the 'Name' field in the 'General Property Info' form to a 'BaseType Info' section. This section shows a dropdown menu labeled 'STRING' and a list of other base types: 123 INTEGER, 123 LONG, JSON, LOCATION, MASHUPNAME, MENUNAME, NOTIFICATIONCONTENTNAME, NOTIFICATIONDEFINITIONNAME, NUMBER, and PASSWORD. A box labeled 'Ganze Zahl Kommazahl' is overlaid on the 'BaseType Info' section.

At the bottom right of the 'BaseType Info' section are buttons for Cancel, Done, and Done and Add.

PROPERTY – WERT EINSTELLEN ODER AKTUALISIEREN

The screenshot shows the 'Properties' tab for an entity named 'Schule_JahrKlasse_KatNr_Thingname'. The left sidebar contains sections for Entity Information, Properties (selected), Services, Events, Subscriptions, Home Mashup, Permissions, Change History, and Dependencies. The main area displays a table of properties under 'My Properties'. One row is selected for 'Sensorwert_1', which has a 'Type' of 'DataChange'. The 'Value' column contains a 'Set' button, which is highlighted with a black box and connected by arrows to two callout boxes: one labeled 'Wert aktualisieren' and another labeled 'Wert einstellen.'

Edit	Name	Type	Alerts	Additional Info	Default Value	Value
	Sensorwert_1	DataChange	0 Alerts			Wert aktualisieren Wert einstellen.

- **Wert einstellen:** Wenn am Arduino ein Wert abgefragt wird, kann dieser hier eingestellt werden. Am Arduino wird sich im Serial Monitor die Zahl ändern.
- **Wert aktualisieren:** Wenn Sensorwerte von Arduino geschickt werden, dann muss bei einer Wertänderung hier händisch aktualisiert werden (sonst wird alter Wert angezeigt)

ÜBUNG

- Erstelle Dein eigenes Thing mit der Namenskonvention in Thingworx Composer
- Erstelle eine beliebige Property mit dem Datentyp Number
- Stelle bei der Property einen beliebigen Wert ein
- Konfiguriere die Datei Thingworx_MKR1010_Variable.h
- Frage den Wert mit dem Arduino ab und verändere diesen (Der Wert wird im Serial Monitor gesehen!)

STEUERUNG DES RELAIS MITTELS THINGWORX COMPOSER

ÜBUNG

- Mittels eines Things im Thingworx Composer soll das Relais am MKR1010 Proto Shield gesteuert werden.
- Die zugehörige Property ist vom Typ Integer. Wenn eine „1“ eingestellt ist, dann soll das Relais schalten. Bei einer „0“ wird das Relais geöffnet.
- Am Relais ist der Ventilator verbunden. Dieser wird ein- und ausgeschalten.
- Es wird dazu das Beispiel „004_GET_Relay“ verwendet
 - Bemerkung: Der Aufbau ist dem Programm „003_GET_Value“ ähnlich. Die gleichen Programmteile werden nicht nochmals beschrieben.

004_GET_RELAYS

```
//Definition of used Libraries
#include "Thingworx_MKRWifil010.h"
#include "Thingworx_MKRWifil010_Variable.h"

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

//Relay 1 is on pin 1, relay 2 is on pin 2
#define RELAY_1 1
#define RELAY_2 2

void setup() {
    Serial.begin(9600);
    myThing.Wifi(); //Serial communications with computer at 9600 bauds for debug purposes
    //Start the Wifi Connection
    pinMode(RELAY_1,OUTPUT); //Digital pin 1 is an output pin
    pinMode(RELAY_2,OUTPUT); //Digital pin 2 is an output pin
}

void loop() {
    if (millis() - lastConnectionTime > TPOST) //Send a POST request to the Thingworx API
    {
        //Logic for the relay
        if( myThing.getjson("Ventilator") == 1.0)
        {
            digitalWrite(RELAY_1,HIGH);
        }
        else
        {
            digitalWrite(RELAY_1,LOW);
        }

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

Das Relais 1/2 ist standardmäßig mit dem digitalen Pin 1/2 verbunden.
Hier wird eine Konstante für die Pinzuweisung erstellt.

Der Digitalpin 1 und 2 wird als ein Ausgang gesetzt.

004_GET_RELAYS

```
//Definition of used Libraries
#include "Thingworx_MKRWifil010.h"
#include "Thingworx_MKRWifil010_Variable.h"

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

//Relay 1 is on pin 1, relay 2 is on pin 2
#define RELAY_1 1
#define RELAY_2 2

void setup() {
    Serial.begin(9600);                                //Serial communications with computer at 9600 bauds for debug purposes
    myThing.Wifi();                                     //Start the Wifi Connection
    pinMode(RELAY_1,OUTPUT);                            //Digital pin 1 is an output pin
    pinMode(RELAY_2,OUTPUT);                            //Digital pin 2 is an output pin
}

void loop() {
    if (millis() - lastConnectionTime > TPOST)          //Send request to server every TPOST seconds
    {
        //Logic for the relay
        if( myThing.getjson("Ventilator") == 1.0
        {
            digitalWrite(RELAY_1,HIGH);
        }
        else
        {
            digitalWrite(RELAY_1,LOW);
        }

        lastConnectionTime = millis();
    }
}
```

If-Abfrage:

Die Property REL1 wird abgefragt. Ist der Wert 1 → der Pin wird auf HIGH gesetzt (das Relais wird geschalten). Sonst (else) wird das Relais geöffnet.

TEMPERATUR- UND FEUCHTIGKEITSMESSUNG

ÜBUNG

- Es wird mittels dem angeschlossenen DHT11 Sensor Temperatur und Feuchtigkeit gemessen.
- Diese Werte werden an 2 Properties eines Things zum Thingworx Composer Server geschickt.
- Es wird dazu das Beispiel „002_PUT_DHT11_Value“ verwendet
 - Bemerkung: Die Thing- und Propertynamen müssen angepasst werden.

002_PUT_DHT11_VALUE

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

//Definition for DHT11 sensor
#define DHT11_PIN 3
DHT sens_dht( DHT11_PIN ,DHT11);

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

void setup() {
    binMode(DHT11_PIN, INPUT);
    Serial.begin(9600);
    myThing.Wifi();
    sens_dht.begin();
}

void loop() {
    if (millis() - lastConnectionTime > TPOST) //Send request to server every T
    {
        //Send data with PUT Request to Thingworx
        myThing.put("DHT11_Temp",sens_dht.readTemperature()); //Send temperature to server
        myThing.put("DHT11_Moist",sens_dht.readHumidity()); //Send humidity to server

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

Einfügen der Library DHT.h

Dadurch können vorgefertigte Befehle für das Messen von Temperatur und Feuchtigkeit verwendet werden.

Die Signalleitung des DHT11 ist am Pin 3 angeschlossen

Es wird wieder (wie bei ThingWorx) eine Klasse für den Sensor erstellt. Die Klasse heißt sens_dht.

Der Pin 3 ist ein Input Pin.

002_PUT_DHT11_VALUE

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

//Definition for DHT11 sensor
#define DHT11_PIN 3
DHT sens_dht( DHT11_PIN ,DHT11);

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

void setup() {
    pinMode(DHT11_PIN, INPUT);                                // set DHT11_PIN to input
    Serial.begin(9600);                                       //Serial communications with co
    myThing.Wifi();                                         //start the Wifi Connection
    sens_dht.begin();
}

void loop() {
    if (millis() - lastConnectionTime > TPOST)           //Send request to server every T
    {
        //Send data with PUT Request to Thingworx
        myThing.put("DHT11_Temp",sens_dht.readTemperature()); //Send temperature to s
        myThing.put("DHT11_Moist",sens_dht.readHumidity());    //Send humidity to se

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

Es wird mit dem Befehl `sens_dht.readTemperature` bzw.
`sens_dht.readHumidity` der Bibliothek `DHT.h` die
Temperatur bzw. Feuchtigkeitswerte abgefragt.
Danach wird mit dem Befehl `myThing.put(...,...)` die
Temperatur auf das Thing mit der Property
„`DHT11_Temp`“ bzw. „`DHT11_Moist`“ geschickt.

002_PUT_DHT11_VALUE – THINGWORX COMPOSER

The screenshot shows the ThingWorx Composer interface. On the left is a sidebar with various icons and a list of open projects. The main area displays a project titled "Thing:304417_FIA_IOTSchulung_Dummy". The "Properties" tab is selected, showing a table of properties. A callout box with the text "Bei neuem Wert muss aktualisiert werden" points to the "Refresh" button at the top of the table. Another callout box with the text "Wert ablesen" points to the value column of the first row in the table.

Name	Actions	Source	Default Value	Value	Alerts	Category	Additional Info
-T_DHT11_Moist	(i)			45.0	(+)	0	
# DHT11_Temp	(i)			25.6	(+)	0	
# Sensorwert_1	(i)			25	(+)	0	
# Sensorwert1	(i)			0	(+)	0	
# Ventilator	(i)			0	(+)	0	

ERSTELLUNG BEDIENPANEL („MASH UP“) IN THINGWORX COMPOSER

MASH UP

- In Thingworx Composer können Bedienpanele oder sogenannte Mash Ups erstellt werden.
- Diese erlauben das Erstellen einer GUI (Graphical User Interface)
- Es können somit Sensorwerte visualisiert und getrackt werden.
- Das Mash Up ist per digitalem Endgerät (Tablet, Smart Phone, PC,...) abrufbar.

ERSTELLUNG MASH UP

thingworx Search + New Entity Import/Export Monitoring Help Learning Connector iot-seminar16

All MODELING Things Thing Templates Thing Shapes Data Shapes Networks Projects Model Tags Integration Connectors ANALYTICS Data Analysis Definitions VISUALIZATION Mashups Masters Gadgets Dashboards Menus Media Style Definitions State Definitions DATA STORAGE COLLABORATION SECURITY

Schule_JahrKlasse_KatNr_Thingname 920417_18FISA_00_Lucky_Shield

Mashups type to filter list... Advanced Clear

+ New View Edit Duplicate Delete Permissions

Filtering by: Exclude System Objects

New Mashup

Mashup Type

Page Thing Template Thing Shape

Layout Options

Responsive Static

Showing: 76 items

View	Name	Description	Modified
	diemash		2019-03-20 04:29:48.879
	dibse-raspi3-mashup		2019-03-19 17:51:21.428
	Delic_Fanki_Mashup		2019-03-15 11:15:09.946
	Gruber_Dummer_Mashup		2019-03-15 11:09:47.892
	DummaBua		2019-03-15 11:00:45.399
	910417_Raspberry		2019-03-11 09:20:49.627
	910417_Waterbike		2019-03-11 09:14:37.832
	920417_185C_10_Lucky-Shield-Mashup		2019-03-11 01:23:10.354
	RFID_11_5CHMBZ_1819		2019-03-06 11:49:48.969
	920417_185B_10_Lucky_Shield_Mashup		2019-03-05 01:16:29.164
	TopiLolloh_mashup		2019-03-01 10:41:39.527
	Bildauswertung	Bildauswertung der Fundboxüberwachung	2019-02-22 14:10:21.647
	910417_SensorkitX40	Anzeige der Daten des Sensorkit X40	2019-02-22 13:51:37.752

MASH UP OBERFLÄCHE

The screenshot shows the Thingworx Mashup interface. On the left, there's a sidebar with categories like Widgets, Mashups, and Workspace. A callout box points to the 'Widgets' tab with the text: "Auswahl der zu verwendeten „Widgets“(grafische Funktionen). Beispiel: Anzeigen, Bilder, Diagramme,...". In the center, there's a main workspace for designing the mashup. A callout box points to this workspace with the text: "Grafische Benutzeroberfläche. Hier können die Widgets reingezogen und positioniert werden.". On the right, there are tabs for Data, Session, and User, and a table for managing connections. A callout box points to the 'Mashup' tab in the bottom-left corner with the text: "Hier können Things und Properties ausgewählt werden. Im späteren Verlauf werden die Properties dann mit den Widgets verknüpft. Werte können so in Anzeigen dargestellt werden." Arrows from the callout boxes point to their respective target areas in the interface.

Hier können Things und Properties ausgewählt werden. Im späteren Verlauf werden die Properties dann mit den Widgets verknüpft. Werte können so in Anzeigen dargestellt werden.

Auswahl der zu verwendeten „Widgets“(grafische Funktionen). Beispiel: Anzeigen, Bilder, Diagramme,...

Grafische Benutzeroberfläche.
Hier können die Widgets reingezogen und positioniert werden.

Eigenschaften des ausgewählten Widgets

TEXT ERSTELLEN

The screenshot shows the thingworx interface for creating a new Mashup. A 'Label' widget has been selected from the 'Widgets' category and is being drag-and-dropped onto the workspace. The workspace contains a single label with the text 'Temperatur'. The properties panel on the left is open for the selected 'label-3' widget, showing its configuration. A callout box labeled 'Drag&Drop' points to the action of dragging the widget.

Widgets **Mashups** **Workspace**

Category All

Drag&Drop

label-3

Name	Value
-T- Id	label-3
-T- Type	Label
-T- DisplayName	
-T- Description	
-T- Text	Temperatur
Style	
-T- Alignment	Left Aligned
-T- ToolTipField	
ShowDataLoading	<input checked="" type="checkbox"/>

Connections 0 To-Do

label-3

New Entity Import/Export Monitoring Help Learning Connector iot-seminar16

Available Space Default Language

Data Session User

ANZEIGE ERSTELLEN

The screenshot shows the thingworx interface for creating a new Mashup. A 'Gauge' widget has been selected from the left sidebar and is being drag-and-dropped onto the main workspace. The workspace is titled 'Temperatur'. The properties panel on the left shows the following configuration for the 'gauge-4' widget:

Name	Value
- Id	gauge-4
- Type	Gauge
- DisplayName	gauge-4
- Description	
# Data ↵	
# MinValue ↵	0
# MaxValue ↵	100
ValueFormatter	State Formatting
FormatNeedle	<input checked="" type="checkbox"/>

A large black callout box labeled 'Drag&Drop' points to the 'Gauge' icon in the sidebar. Another callout box labeled 'gauge-4' points to the widget instance in the workspace.

HINZUFÜGEN VON DATEN

The screenshot shows the thingworx interface for creating a new mashup. The top navigation bar includes 'Search', 'New Entity', 'Import/Export', 'Monitoring', 'Help', 'Learning Connector', and a user account 'iot-seminar16'. Below the navigation is a toolbar with various icons for 'Widgets', 'Mashups', 'Workspace', and a 'More' dropdown. A central workspace shows a 'Temperatur' (Temperature) gauge. On the left, a sidebar lists available widgets like 'Expression', 'Fieldset', 'File Upload', 'Folding Panel', 'Gauge', 'GeoTag', 'Google Location Picker', 'Google Map', and 'Grid'. A specific 'gauge-4' component is selected, showing its properties: Name (gauge-4), Type (Gauge), DisplayName (gauge-4), Description, Data, MinValue (0), MaxValue (100), ValueFormatter, and FormatNeedle. A modal dialog titled 'Add Data' is open, containing a 'Select Entity' dropdown set to 'Things', a search bar, and a 'Search Results' section. The 'Search Results' section has an 'Actions' button with a '+' icon and a 'Thing' label. It displays a list of entities under 'Recent' (44) and 'TYPES' (Things, 223). One entity, '920417_18FISA_00_Dummy', is highlighted. A callout box points to this entity with the text 'Thing auswählen' (Select Thing). A large black rectangle highlights the entire 'Add Data' dialog.

Add Data

Select Entity

Search Results

Actions: + Thing

Recent 44

TYPES Things 223

920417_18FISA_00_Dummy

- 920417_18FISA_00_Lucky_Shield
- 920417_18FISA_00_Library_Test
- 920417_185C_10_Lucky_Shield
- 920417_185B_10_Lucky_Shield
- 910417_Raspberry
- RFID_11_5CHMBZ_1819

HINZUFÜGEN VON DATEN

Add Data

Select Entity  920417_18FISA_00_Dummy  Dynamic  

Select Services

All 

- Alerts
- Bindings
- Configuration
- Data
- DataLogging
- Dependencies
- Editing
- Federation
- Identifier
- Lifecycle
- Maintenance
- Mashups
- Metadata
- Networks
- Permissions 

getpropertyvalues 

GetPropertyValues  

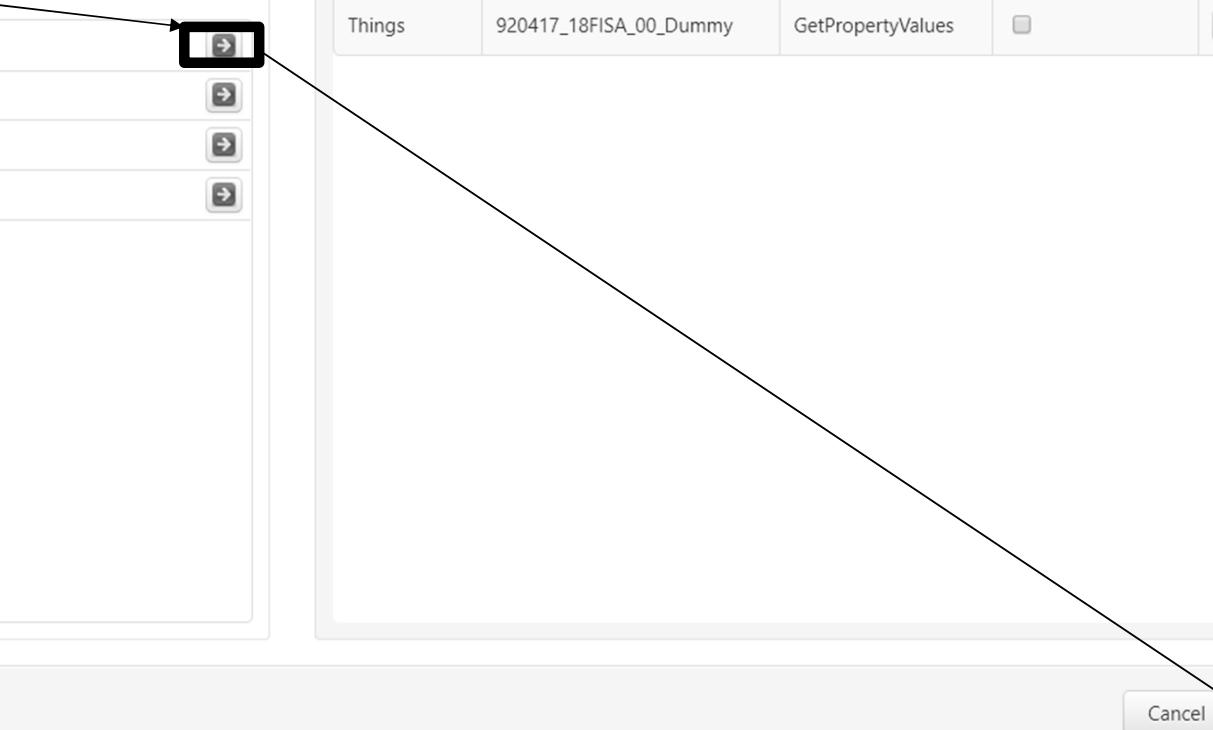
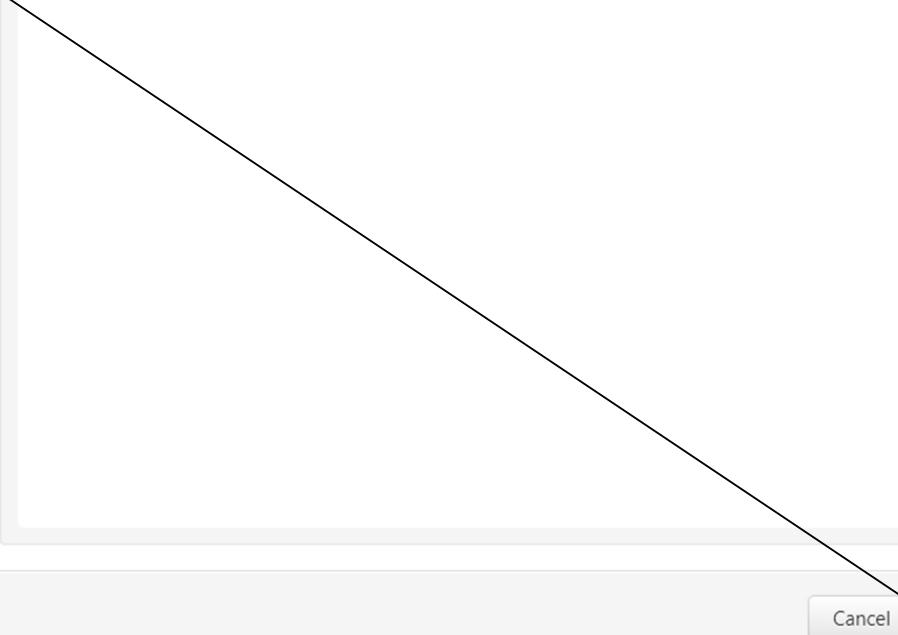
GetPropertyValuesAsMultiRowTable  

GetPropertyValuesVTQ  

GetPropertyValuesVTQA  

Selected Services

Entity Type	Entity Name	Service	Mashup Loaded?	Remove
Things	920417_18FISA_00_Dummy	GetPropertyValues	<input type="checkbox"/>	

Cancel 

VERBINDEN DER DATEN MIT WIDGET

The screenshot shows the thingworx interface for creating a new mashup. The main area displays a gauge widget titled "Temperatur". A callout box labeled "Drag&Drop" points from the gauge's binding target area to the "Data" section of the right-hand sidebar. Another callout box labeled "Select Binding target" points to the dropdown menu where "# Data" is selected. A third callout box contains the German text: "In diesem Fenster werden alle verbundenen Daten zu einem Widget angezeigt". The sidebar also lists other binding options like "# MinValue", "# MaxValue", "-T- Legend", and "-T- ToolTipField". Below the gauge, the "Connections" panel shows a flow from a "GetPropertyValues" block to a "gauge-4" block, with "123 Sensorwert_1" connected to the "Data" input of the gauge.

thingworx

New Mashup **Mashup** ? Design Info Save Cancel Edit

Widgets Mashups Workspace

Category All

Filter Widgets

Entity Picker Event Chart Events Router Expression Fieldset File Upload Folding Panel Gauge GeoTag

panel-2

Filter Properties

Name	Value
-T- Id	panel-2
-T- Type	Panel
-T- DisplayName	panel-2
-T- Description	
Style	
HideScrollbars	
ShowDataLoading	
ResetInputsToDefaultValue	
# Z-index	10

+

New Entity Import/Export Monitoring Help Learning Connector iot-seminar16

Temperatur

Select Binding target

Data

MinValue

MaxValue

-T- Legend

-T- ToolTipField

Drag&Drop

Things_920417_18FISA_00_Dummy

GetPropertyValues

Parameters

Returned Data

All Data

123 Sensorwert_1

name

description

thingTemplate

tags

Selected Row(s)

In diesem Fenster werden alle verbundenen Daten zu einem Widget angezeigt

HINZUFÜGEN EINES REFRESH BUTTONS

The screenshot shows the Thingworx interface for creating a Mashup. On the left, the 'Widgets' panel is open, with 'Auto Refresh' selected and highlighted with a black box. A callout arrow points from this selection to a 'Refresh Now' button located in the center of the workspace. Another callout arrow points from the 'Refresh Interval' field in the properties panel below to the 'Refresh Now' button. The workspace contains a 'Temperature' gauge. To the right, the 'Data' panel shows a connection to a 'Things_920417_18FISA_00_Dummy' entity with a 'GetPropertyValues' step.

Dieser Button ist notwendig um die Anzeige in einem bestimmten Zeitintervall upzudaten.

Name	Value
-T- Id	autorefresh-5
-T- Type	Auto Refresh
-T- DisplayName	autorefresh-5
-T- Description	
# RefreshInterval	30
AutoRefresh	<input checked="" type="checkbox"/>
ShowControls	<input checked="" type="checkbox"/>
# AutoRefreshTabSe...	0
Label	Refresh Now

HINZUFÜGEN EINES REFRESH BUTTONS

The screenshot shows the Thingworx interface for creating a mashup. A central workspace displays a gauge widget titled "Temperatur". A "Refresh Now" button is highlighted with a callout bubble containing the text "Drag&Drop". Below the workspace, the "Connections" panel shows a flow from an "autorefresh-5" entity to a "GetPropertyValues" step, which then connects to a "gauge-4" data source. On the left, the "Widgets" palette lists various components like Auto Refresh, Button, Checkbox, and Gauge. A modal window titled "Things 920417_18FISA_00_Dummy" is open, showing a "GetPropertiesValues" step. The bottom right corner features a table for managing data values.

Widget Properties for "autorefresh-5":

Name	Value
-T- Id	autorefresh-5
-T- Type	Auto Refresh
-T- DisplayName	autorefresh-5
-T- Description	
# RefreshInterval	30
AutoRefresh	<input checked="" type="checkbox"/>
ShowControls	<input checked="" type="checkbox"/>
# AutoRefreshTabSe...	0
-T- Label	Refresh Now

HINZUFÜGEN EINES DIAGRAMMES

The screenshot shows the thingworx interface for creating a Mashup. The main area displays a dashboard titled "Temperatur" with a "Time Series Chart" and a "Drag&Drop" placeholder. A tooltip "Time Series Chart" is visible near the chart icon. On the left, a sidebar lists various widgets, with "Time Series Chart" selected and highlighted. Below it, the properties for "timeserieschart-6" are shown in a table:

Name	Value
-T- Id	timeserieschart-6
-T- Type	Time Series Chart
-T- DisplayName	timeserieschart-6
-T- Description	
<input checked="" type="checkbox"/> SingleDataSource	
# NumberOfSeries	8
Data ↴	
-T- ChartType	Line
ChartStyle	

On the right, the "Data" panel shows a list of entities and their properties. A new entity "Things.920417.18FISA.00.Dummy" is selected, and its "GetPropertyValues" method is expanded, showing parameters like "description", "name", and "tags". A tooltip "Selected Row(s)" points to the "Selected Row(s)" entry. At the bottom right, a preview window shows the "timeserieschart-6" component.

HINZUFÜGEN EINES DIAGRAMMES

Add Data

Select Entity  920417_18FISA_00_Dummy 

Select Services  querypropertyhistory 

Dynamic  

All  Alerts Bindings Configuration Data DataLogging Dependencies Editing Federation Identifier Lifecycle Maintenance Mashups Metadata Networks Permissions

QueryPropertyHistory 

Selected Services

Entity Type	Entity Name	Service	Mashup Loaded?	Remove
Things	920417_18FISA_00_Dummy	QueryPropertyHistory	<input type="checkbox"/>	

HINZUFÜGEN EINES DIAGRAMMES

The screenshot shows the thingworx interface for creating a Mashup. A 'Select Binding Target' dialog box is open in the center, listing 'Data' as the selected target. A 'Drag&Drop' callout points from the 'All Data' section in the right-hand sidebar to the 'Data' button in the dialog. The sidebar also lists other data sources: 'DataSource1', 'DataSource2', and 'DataSource3'. The main workspace contains a panel titled 'Temperatur' with a gauge widget. The left sidebar shows various widget categories like Widgets, Mashups, and Workspace, along with a list of available components. The bottom right corner displays a table of properties for a selected entity.

Select Binding Target

- Data
- DataSource1
- DataSource2
- DataSource3

Drag&Drop

Things_920417_18FISA_00_Dummy1:Qu...

Name	Value
oldestFirst	☐
# maxItems	☐
endDate	☐
query	Edit Query
startDate	☐
ServiceInvokeCompleted	☐
AllDataChanged	☐
SelectedRowsChanged	☐

DIAGRAMM AKTUALISIEREN

The screenshot shows the thingworx interface for editing a dashboard. The top navigation bar includes options like New Entity, Import/Export, Monitoring, Help, Learning Connector, and a user account. The main workspace displays a panel titled "panel-2" containing a "Temperatur" section with a Gauge widget and a Time Series Chart. A context menu is open over the Gauge widget, showing options like "On", "Refresh Now", "Widget", "Configure Bindings", and "Refresh". A large black callout box with the text "Drag&Drop" is positioned over the central workspace area. On the left, a sidebar lists various widget categories such as Widgets, Mashups, and Workspace. On the right, there are several panes: one for "Data" showing "Returned Data" and "All Data" with fields like "description", "name", and "tags"; another for "Session" and "User"; and a third for "ThingTemplate" showing a list of things like "Things_920417_18FISA_00_Dummy1" and its properties. At the bottom, a detailed view of the "QueryPropertyHistory" property for "Things_920417_18FISA_00_Dummy1" is shown, listing parameters like "oldestFirst", "maxItems", "endDate", "query", "startDate", and event handlers for "ServiceInvokeCompleted", "AllDataChanged", and "SelectedRowsChanged".

Panel Properties:

Name	Value
-T- Id	panel-2
-T- Type	Panel
-T- DisplayName	panel-2
-T- Description	
Style	
HideScrollbars	
ShowDataLoading	
ResetInputsToDefaultValue	
# Z-index	10

Properties for Things_920417_18FISA_00_Dummy1:QueryPropertyHistory:

Name	Value
oldestFirst	
# maxItems	
endDate	
query	Edit Query
startDate	
ServiceInvokeCompleted	
AllDataChanged	
SelectedRowsChanged	

EINSTELLUNGEN AM THING FÜR DAS SPEICHERN VON DATEN – ERSTELLUNG VALUE STREAM

The screenshot shows the Thingworx interface with the following details:

- Left Sidebar:** Contains categories like Networks, Projects, Model Tags, Integration Connectors, ANALYTICS, VISUALIZATION, DATA STORAGE, and a highlighted **Value Streams** category.
- Top Bar:** Includes a search bar, navigation links for New Entity, Import/Export, Monitoring, Help, Learning Connector, and a user account (iot-seminar16).
- Current View:** The "Value Streams" view, showing a list of existing streams and a "Choose Template" dialog.
- "Value Streams" Dialog:** A modal window titled "Choose Template" with the instruction "Choose from the following Thing Templates." It lists two options:

Name	Description
RemoteValueStream	Remote Value Stream
ValueStream	Value Stream
- Table on the Right:** Shows a list of items with columns for Name, Type, and Modified date.
- Bottom Navigation:** Includes a URL (https://txw.htl.schule/Thingworx/Composer/index.html#) and a footer note (ValueStream).

EINSTELLUNGEN AM THING FÜR DAS SPEICHERN VON DATEN – ERSTELLUNG VALUE STREAM

The screenshot shows the Thingworx interface for creating a new entity. The top navigation bar includes 'Search', 'New Entity', 'Import/Export', 'Monitoring', 'Help', 'Learning Connector', and a user account 'iot-seminar16'. The main workspace shows several tabs: 'Schule_JahrKlasse_KatNr_Thingname', '920417_18FISA_00_Lucky_Shield', '920417_18FISA_00_Dummy', '123', and '920417_18FISA_00_Dummy_ValueStream'. The current tab is '920417_18FISA_00_Dummy_ValueStream'. The interface has a left sidebar with sections for 'ENTITY INFORMATION', 'PERMISSIONS', 'CHANGE HISTORY', and 'DEPENDENCIES'. The 'ENTITY INFORMATION' section is expanded, showing 'General Information' with fields for Name (set to '920417_18FISA_00_Dummy_ValueStream'), Description, Project (set to '920417_TGM'), Tags, Thing Template (set to 'ValueStream'), Implemented Shapes, Persistence Provider (set to 'ThingworxPersistenceProvider'), and Documentation. A large text area for Documentation is present. The 'General Information' field is highlighted with a red box. A red arrow points from the 'Save' button at the top to the 'Active' checkbox in the 'General Information' section. The 'Active' checkbox is checked.

EINSTELLUNGEN AM THING FÜR DAS SPEICHERN VON DATEN

The screenshot shows the Thingworx application interface. At the top, there is a navigation bar with links for 'New Entity', 'Import/Export', 'Monitoring', 'Help', 'Learning Connector', and a user account 'iot-seminar16'. Below the navigation bar, a toolbar includes 'Save' and 'Cancel Edit' buttons, along with a 'To Do' dropdown and a 'More' button.

The main content area displays a 'General Information' card for a 'Thing' named '920417_18FISA_00_Dummy'. The card contains the following fields:

- Name: 920417_18FISA_00_Dummy
- Description: (empty)
- Project: Search Projects
- Tags: Search Model Vocabulary
- Thing Template: GenericThing
- Implemented Shapes: Search Thing Shapes
- Active: checked
- Home Mashup: Search Mashups
- Avatar: Change
- Published: unchecked
- Identifier: (empty) Browse...
- Last Modified Date: 2019-03-20 10:25:55.248
- Value Stream: 920417_18FISA_00_Dummy_ValueStream

A large black arrow points from the 'General Information' link in the left sidebar to the 'Value Stream' field in the card. The sidebar also lists other sections: ENTITY INFORMATION, PERMISSIONS, CHANGE HISTORY, and DEPENDENCIES.

EINSTELLUNGEN AM THING FÜR DAS SPEICHERN VON DATEN

The screenshot shows the thingworx interface for managing entities. The top navigation bar includes 'Search', 'New Entity', 'Import/Export', 'Monitoring', 'Help', 'Learning Connector', and a user dropdown for 'iot-seminar16'. Below the navigation is a toolbar with icons for Home, Entity, Thing, Save, Cancel Edit, To Do (1), and More.

The main workspace displays the properties of a 'Thing' named '920417_18FISA_00_Dummy'. The left sidebar contains sections for Entity Information (Properties selected), Permissions (Visibility, Design Time, Run Time), Change History (Change History), and Dependencies (Entity Depends On, Uses This Entity).

The central area shows the 'Properties' tab for 'My Properties'. A table lists a single property named '123 Sensorwert_1'. The table columns include Name, Type, Alerts, Additional Info, Default Value, Value, and DataChange. The 'Value' field is set to 25, and the 'DataChange' section shows 'Value: 0'.

Below the table, detailed property information is shown in three panels:

- General Property Info:** Name: Sensorwert_1, Description: (empty), Category: (empty), Alerts: Manage Alerts, Enabled?, Type, Config, Name, Desc.
- BaseType Info:** Base Type: 123 INTEGER, Units: (empty), Min Value: (empty), Max Value: (empty), Has Default Value: (checkbox).
- Aspects:** Persistent: (checkbox), Read-only: (checkbox) (selected), Logged: (checkbox) (selected).

At the bottom right of the properties panel are 'Cancel' and 'Done' buttons.

At the very bottom, a footer bar shows 'GenericThing (ThingTemplate) - Properties' and 'Generic Properties'.

LETZTE EINSTELLUNGEN AM DIAGRAMM

# LegendWidth	0
-T- LegendLocation	Right ▾
-T- LegendOrientation	Vertical ▾
# MarkerSize	3
-T- MarkerType	Circle ▾
Smoothing	
XAxisField	timestamp ▾
ShowXAxis	<input checked="" type="checkbox"/>
XAxisStyle	
-T- XAxisFormat	yyyy-MM-dd HH:mm:
SecondaryYAxisZe...	
AllowSelection	<input checked="" type="checkbox"/>
EnableHover	<input checked="" type="checkbox"/>
ShowXAxisGrid	<input checked="" type="checkbox"/>
ShowYAxisGrid	<input checked="" type="checkbox"/>
GridStyle	
DataField1	Sensorwert_1 ▾
-T- DataLabel1	Wert1
-T- SeriesType1	Use Chart Setting ▾
-T- SeriesMarkerType1	Use Chart Setting ▾

FERTIGES MASHUP IM BROWSER

