



ARDUINO – THINGWORX COMPOSER SCHULUNG

MÄRZ 2019

VORTRAGENDER: ING. ARMIN FISCHER, M.SC. (TGM WIEN)

E-MAIL: AFISCHER2@TGM.AC.AT

VERSION 1.0

A faint, light-gray circuit board pattern serves as the background for the title, featuring various tracks and component pads.

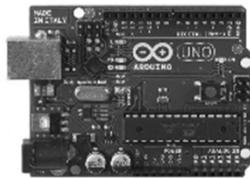
EINFÜHRUNG ARDUINO

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

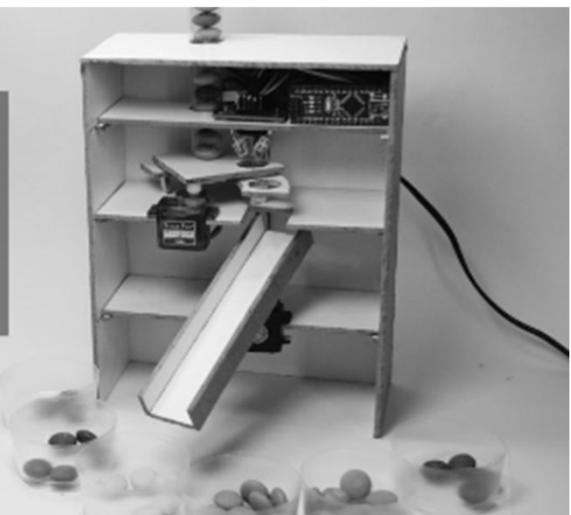


Arduino Leonardo with Headers

A screenshot of the Arduino IDE interface. The title bar reads "sketch_sep14a | Arduino 1.6.11 (Windows Store 1.6.11.0)". The code editor displays the following sketch:

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

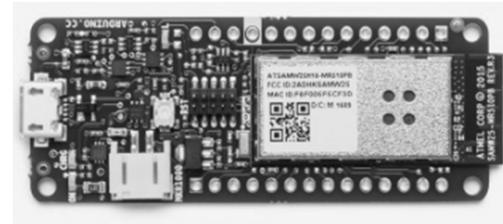
void loop() {
  // put your main code here, to run repeatedly:
}
```





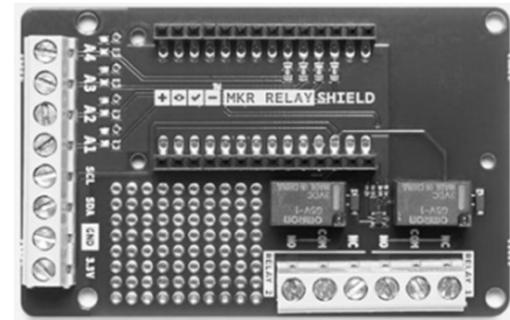
SCHULUNGSMATERIAL

ARDUINO MKR1000



- Stromsparender Mikrocontroller für IOT Anwendungen
- Stromversorgung 5V (USB)
- Wifi Anbindung
- 7 Analoge (0-3.3V) und 14 digitale Pins (LOW-0V & HIGH-3.3V)
- ACHTUNG: Pins dürfen nur bis 3.3V belastet werden
- Weitere Informationen siehe <https://store.arduino.cc/arduino-mkr1000>

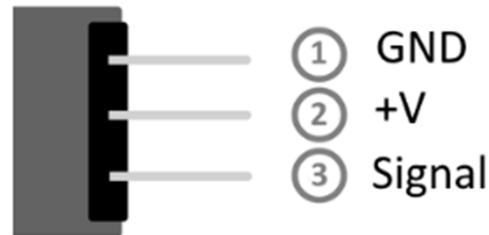
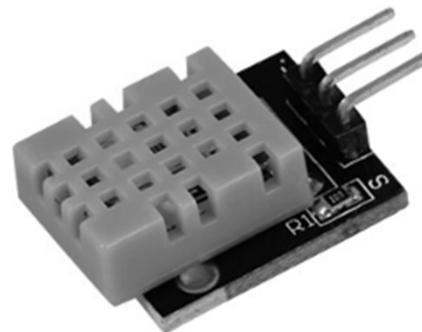
ARDUINO MKR RELAY PROTO SHIELD



- MKR1000 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 http://sensorkit.en.joy-it.net/index.php?title=KY-015_Combi-Sensor_Temperature%2BHumidity



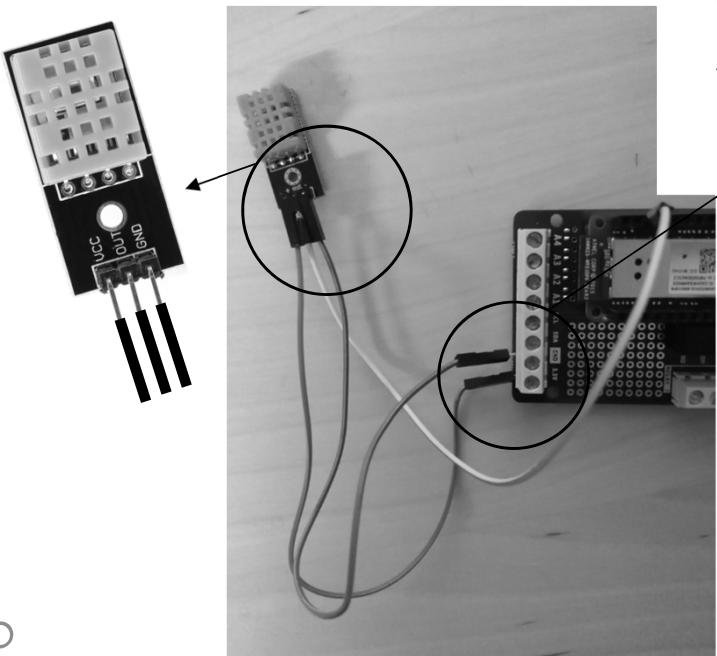
VENTILATOR



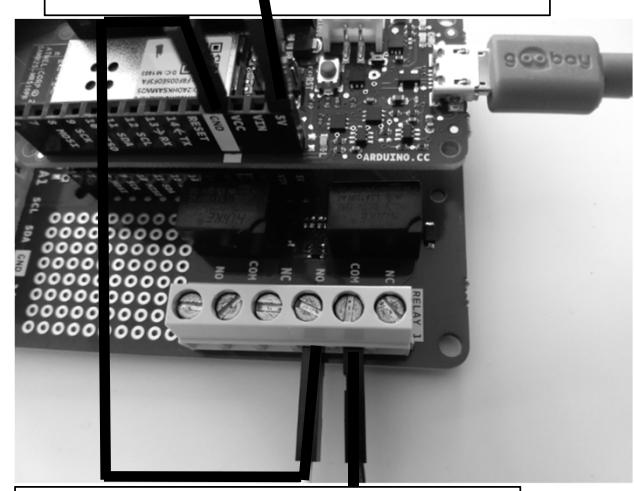
- Betriebsspannung 5VDC
- Weitere Informationen siehe www.reichelt.at/ Prduktnummer: RPI FAN 30X30

VERBINDEN DER EINZELTEILE

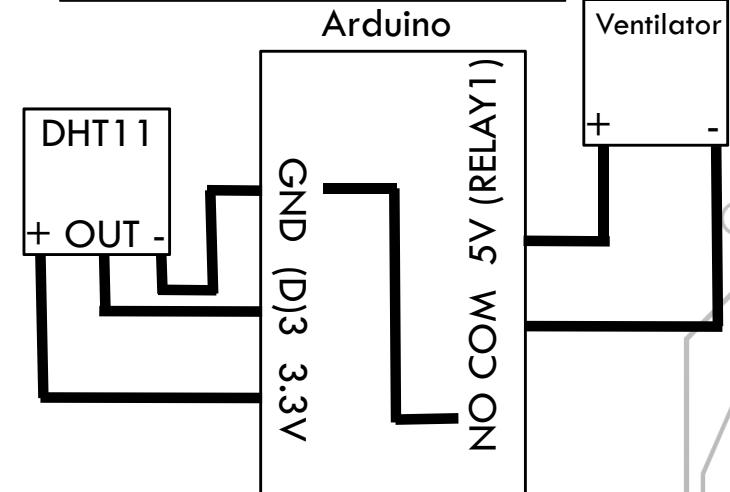
DHT11 mit Arduino Shield



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

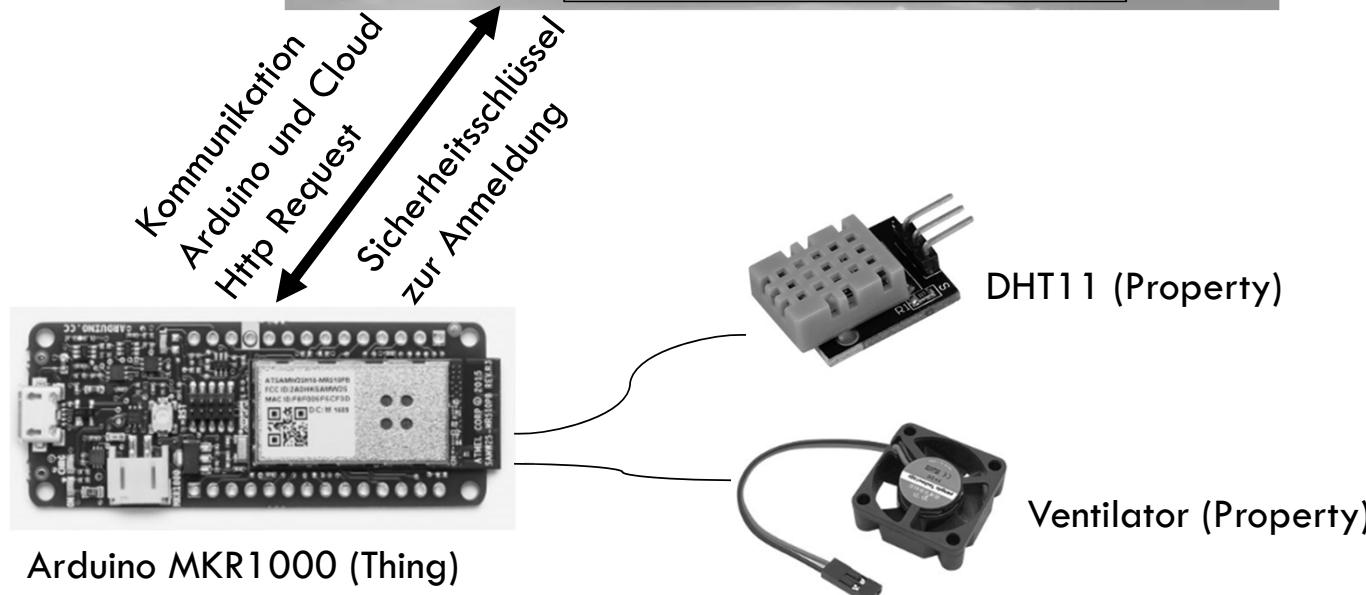
WIE WERDEN DATEN IN DIE CLOUD ÜBERTRAGEN?

- Thing 1
 - Property 1
 - Property 2
 - ...

- Thing 2
 - Property 1
 - Property 2
 - ...

CLOUD <https://twx.htl.schule>
(Server Mödling mit
Thingworx Composer Software)

- Arduino (Thing)
 - Temperatur (Property)
 - Feuchtigkeit (Property)
 - Ventilator (Property)



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

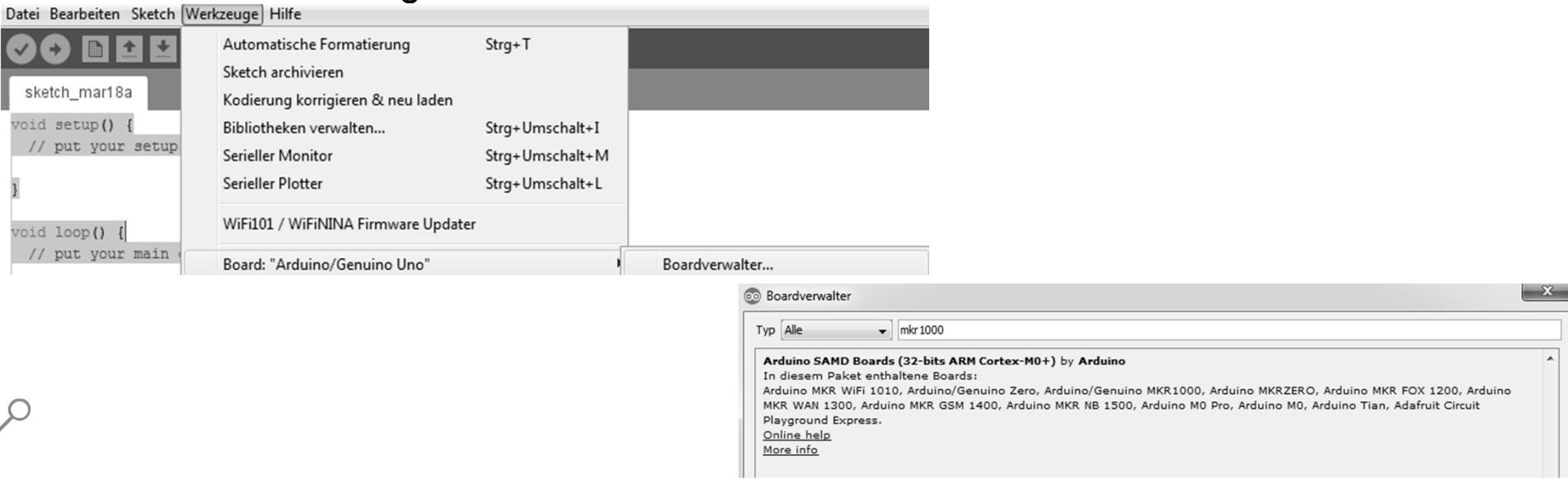
Thingworx Server Adresse	Thing	Property
<u>https://twx.hth.schule/Thingworx/Things/920417_18FISA_00_Dummy</u>	<u>Properties</u>	<u>Sensorwert_1</u>
<u>?appKey=de5d03ab-6cac-4e13-a4a9-9b5bfe9f7eda</u>		
Sicherheitsschlüssel		



REALISIERUNG DER DATENÜBERTRAGUNG MIT ARDUINO

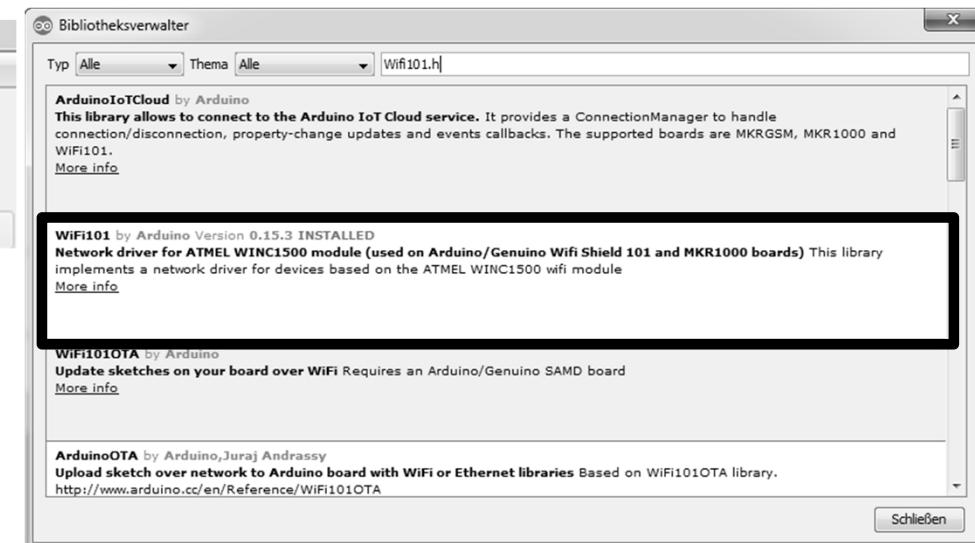
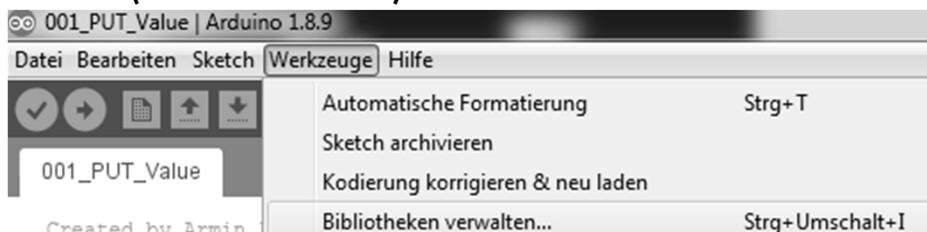
ARDUINO IDE

- Software zum Programmieren von Arduino Mikrocontroller
- Download unter <https://www.arduino.cc/en/main/software>
- Um das Arduino MKR1000 Board verwenden zu können muss dies installiert werden: Werkzeuge > Board > Boardverwalter > Arduino SAMD Boards



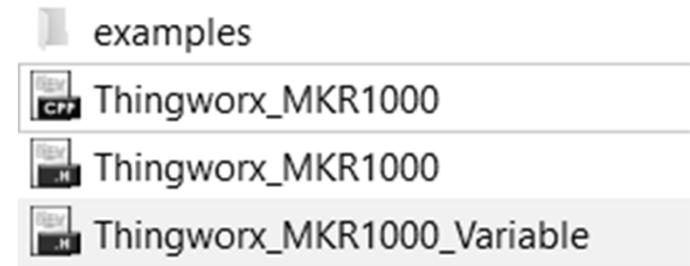
INSTALLATION NÖTIGER LIBRARIES

- Libraries = Vorprogrammierte Codes, welche mit Befehlen aufgerufen werden
- Benötige Libraries für den HTTP Request
 - Thingworx MKR1000 (von USB Stick)
 - Wifi101.h (Werkzeuge > Bibliotheken verwalten > Wifi101.h)
 - Dht.h (von USB Stick)



THINGWORX MKR1000 LIBRARY

- Library von Armin Fischer geschrieben
- Auf Schulungs USB Stick gespeichert (Thingworx_MKR1000)
- 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 MKR1000 LIBRARY - EINSTELLUNGEN

- Die folgenden Informationen müssen in der Datei
Thingworx_MKR1000_Variable.h eingestellt werden

```
#ifndef Thingworx_MKR1000_Variable_H  
#define Thingworx_MKR1000_Variable_H
```

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

```
//Wifi - Variables
```

```
char* ssid = ""; //WiFi SSID  
char* password = ""; //WiFi Pass
```

Benutzername und Passwort von WLAN

```
//Host Thingworx
```

```
char* host = "twx.htl.schule"; //TwX Host for HTL Austria twx.htl.schule (without http at beginning)  
unsigned int port = 8080; //TwX host port
```

```
//Thingworx Variables
```

```
char appKey[] = "4db65fd0-58bb-49f1-8670-8837b4f24bc0"; //API Key from TwX  
char thingName[] = ""; //Thing name from TwX
```

```
//->Timing vars
```

```
unsigned long lastConnectionTime = 0; //Last connection ms time between server requests
```

Sicherheitsschlüssel und
Thingname

THINGWORX MKR1000 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 920417_18FISA_00_Dummy

THINGWORX MKR1000 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 MKR1000 LIBRARY

003_GET_VALUE – ABFRAGEN EINES WERTES

```
//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);

//Variable for Sensor Values
float sensor_1;

void setup() {
    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
    {
        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 MKR1000 LIBRARY

003_GET_VALUE – ABFRAGEN EINES WERTES

```
//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);

//Variable for Sensor Values
float sensor_1;

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

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
    }
}
//
```

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 und thingname von der Datei Thingworx_MKR1000_Variable

THINGWORX MKR1000 LIBRARY

003_GET_VALUE – ABFRAGEN EINES WERTES

```
//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);

//Variable for Sensor Values
float sensor_1;

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

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
  }
}
```

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.

s for debug purposes

THINGWORX MKR1000 LIBRARY

003_GET_VALUE – ABFRAGEN EINES WERTES

```
//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);

//Variable for Sensor Values
float sensor_1;

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

void loop() {
  if (millis() - lastConnectionTime > TPOST)
  {
    sensor_1=myThing.getjson("Sensorwert_1");
    lastConnectionTime = millis();
  }
}

//Serial
//Start the Thingworx connection
//Send request to Thingworx
//Get data with GET Request from Thingworx
//Refresh last connection time for if
```

Die void loop() Schleife wird endlos ausgeführt.

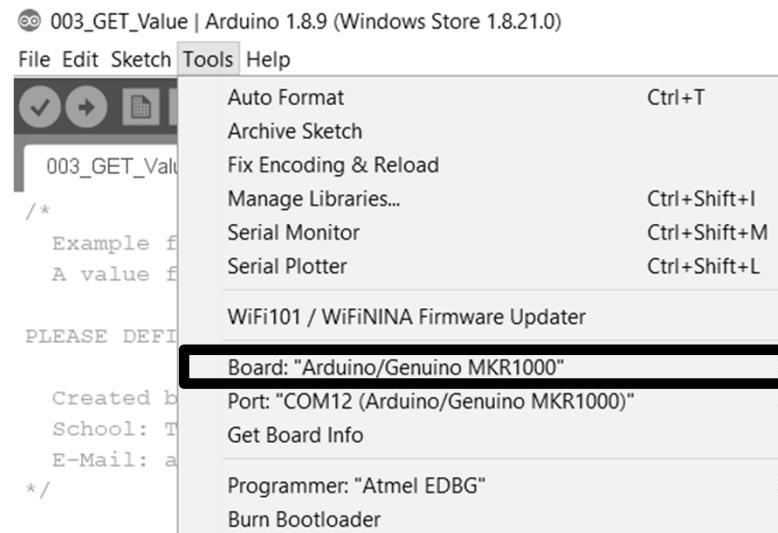
If-Abfrage: Es wird alle TPOST (Variable in der Datei Thingworx_MKR1000_Variable.h) Sekunden ein Wert abgefragt.

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 MKR1000 LIBRARY

003_GET_VALUE – ABFRAGEN EINES WERTES

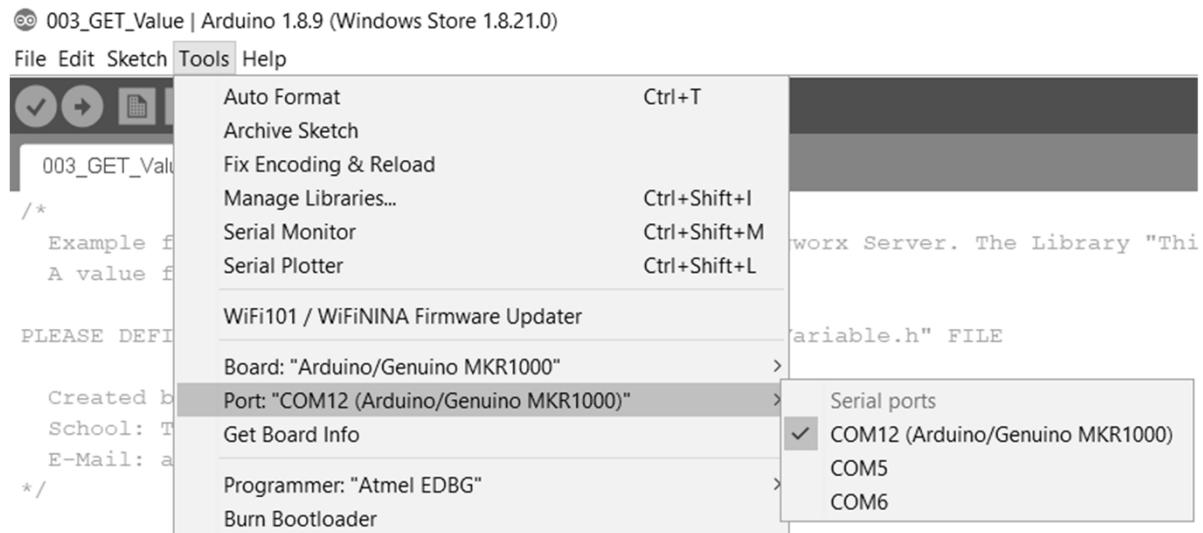
- Einstellung des MKR1000 Mikrocontrollers



THINGWORX MKR1000 LIBRARY

003_GET_VALUE – ABFRAGEN EINES WERTES

- Einstellung des richtigen Port (Darstellung kann abweichen!)



THINGWORX MKR1000 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"

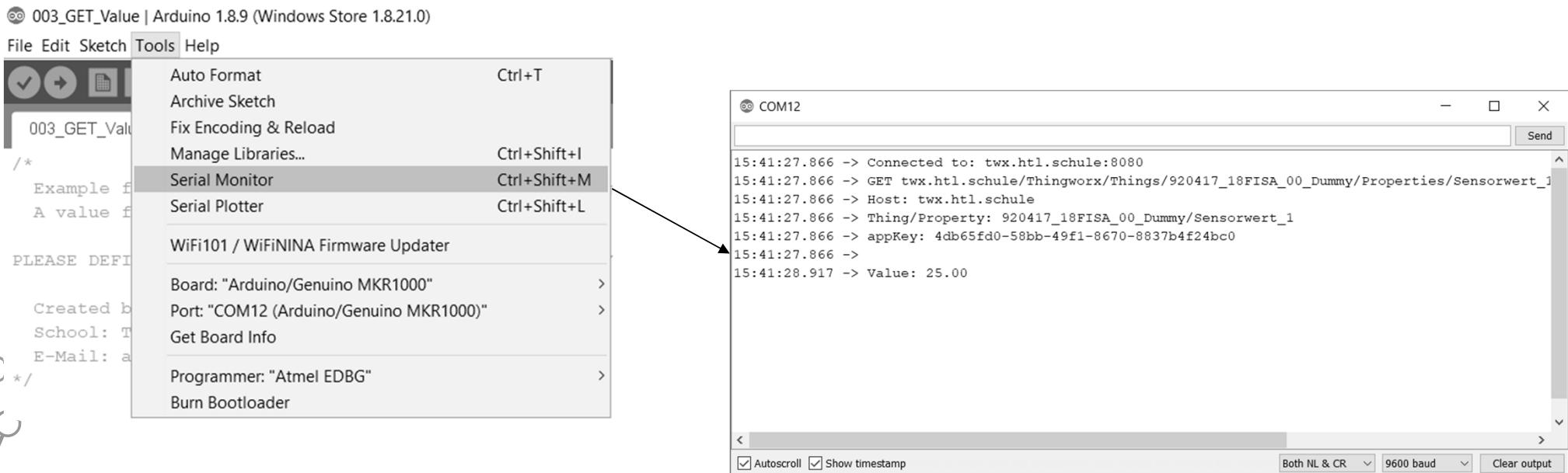
Created by Armin Fischer, Jan 2019.
School: TGM Vienna
E-Mail: afischer2@tgm.ac.at
*/
//Definition of used Libraries
#include "Thingworx_MKR1000.h"
#include "Thingworx_MKR1000_Variable.h"

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

THINGWORX MKR1000 LIBRARY

003_GET_VALUE – ABFRAGEN EINES WERTES

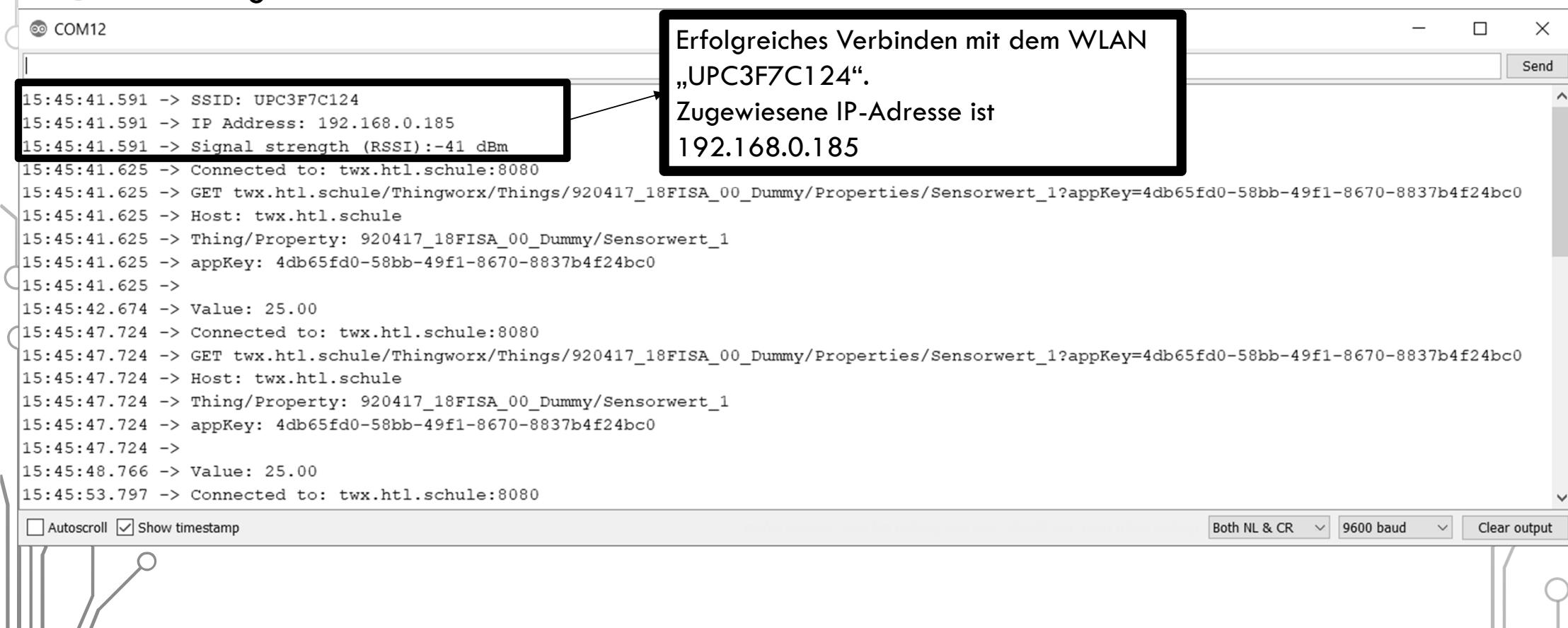
- Einschalten des Serial Monitor um den Ablauf des Programmes nachzuvollziehen



THINGWORX MKR1000 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
|  
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/Properties/Sensorwert_1?appKey=4db65fd0-58bb-49f1-8670-8837b4f24bc0  
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
```

Erfolgreiches Verbinden mit dem WLAN
„UPC3F7C124“.
Zugewiesene IP-Adresse ist
192.168.0.185

Autoscroll Show timestamp Both NL & CR 9600 baud Clear output

THINGWORX MKR1000 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/Properties/Sensorwert_1?appKey=4db65fd0-58bb-49f1-8670-8837b4f24bc0
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
```

Autoscroll Show timestamp

Mit dem Server twx.html.schule am Port 8080 verbunden.
Es wird ein GET Request geschickt mit der Adresse
twx.html.schule/Thingworx/... geschickt.
Es wird das Thing 920417_18FISA_00_Dummy angesprochen. Die Property
Sensorwert_1 wird abgefragt.
Der Sicherheitsschlüssel ist 4db...
Der abgefragt Wert beträgt 25

THINGWORX MKR1000 LIBRARY

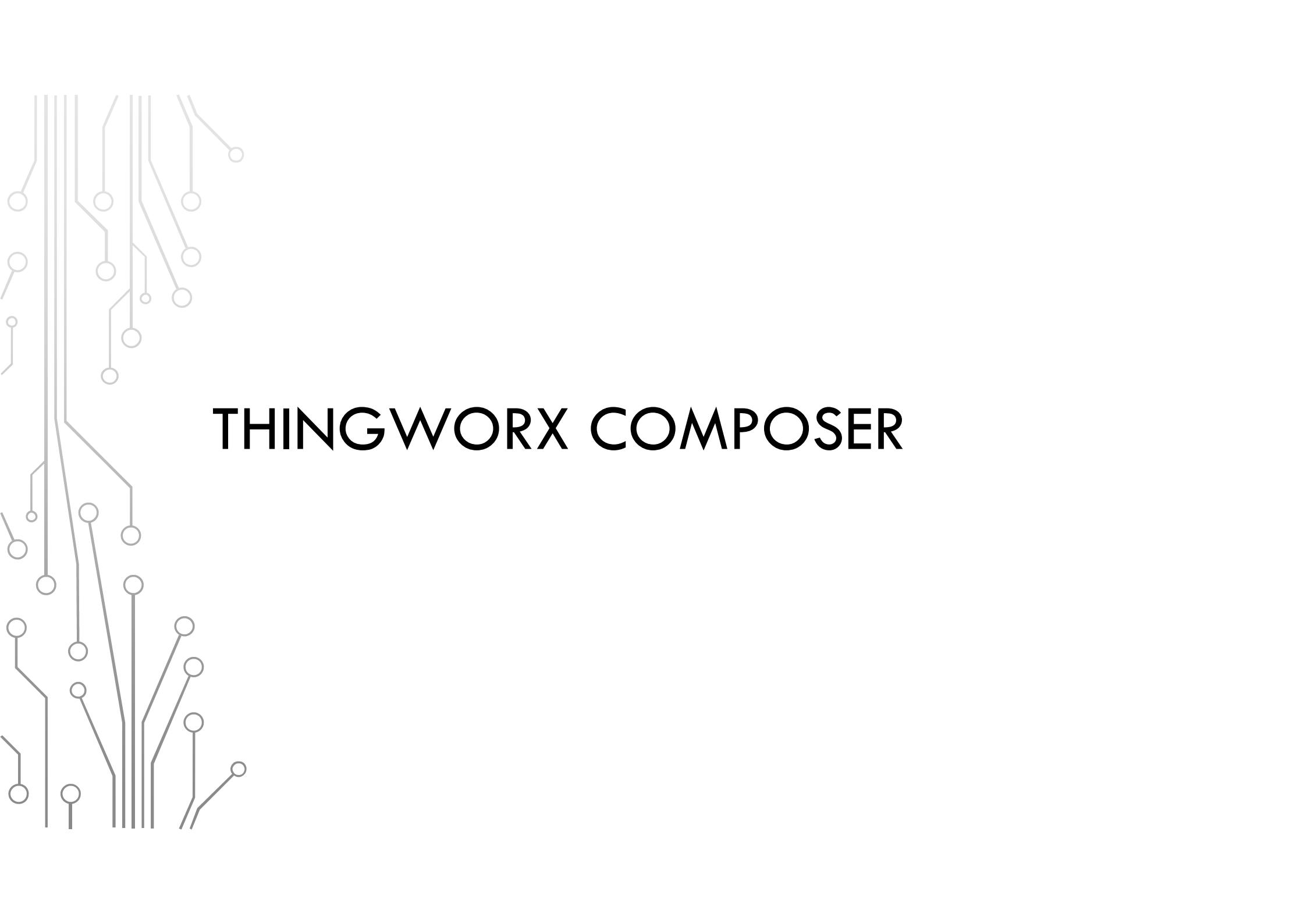
003_GET_VALUE – ABFRAGEN EINES WERTES

- 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
```

Autoscroll Show timestamp Both NL & CR 9600 baud Clear output

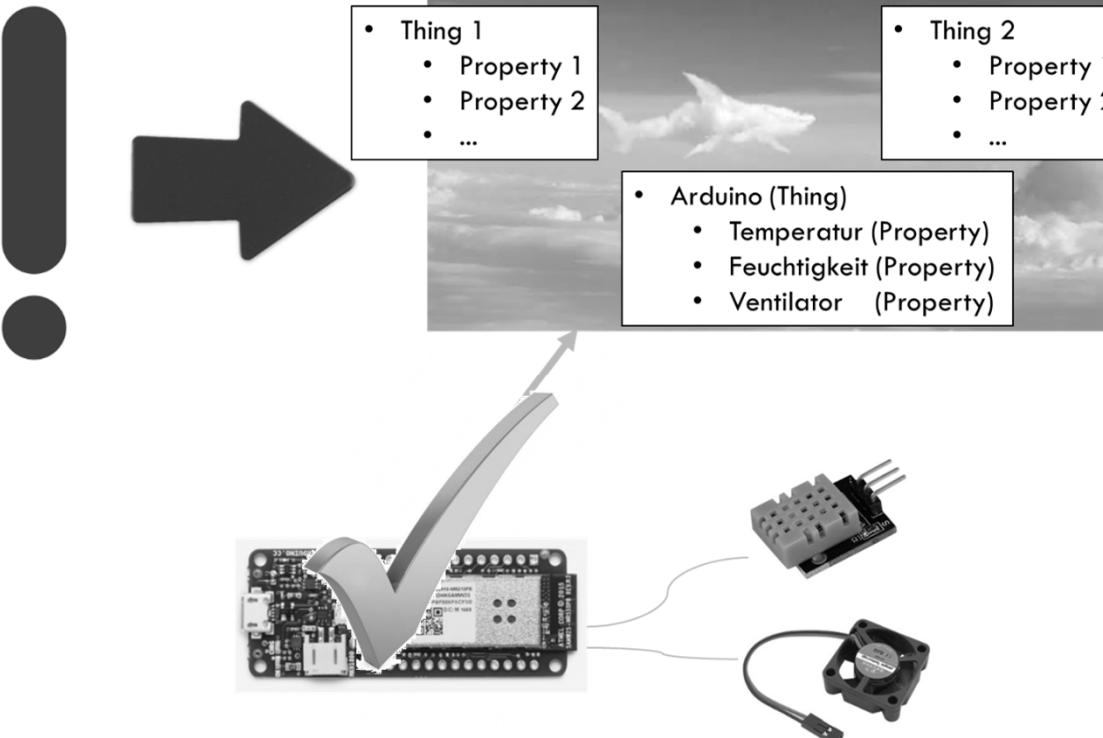
5 Sekunden vergehen und es wird ein weiteres Mal abgefragt.

The background features a complex network of thin, light-gray lines forming a grid-like structure. Small, open circles are placed at various intersections and endpoints of these lines, creating a sense of a circuit board or a molecular lattice.

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://twx.htl.schule/>
- Zugriffsdaten
 - Benutzer: iot-seminar16
 - Passwort: LiTec23-25



OBERFLÄCHE

The screenshot shows the ThingWorx Composer interface. On the left is a sidebar with a navigation tree:

- 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**

The main area displays a list of "Things" with the following columns: View, Name, Description, and Modified. A search bar and filter buttons are at the top of the list table.

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		
<input type="checkbox"/>	Jenbach-Bonapace_Gruber		
<input type="checkbox"/>	Jenbach-Delic-Fanki		
<input type="checkbox"/>	dibse_raspi1		
<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
<input type="checkbox"/>			2019-03-01

A callout box highlights the "Things" section of the sidebar and lists examples of functions:

- Auswahl der verschiedenen Funktionen.
- Beispiele:
- Erstellung von Things
- Erstellung von Benutzeroberflächen „Mash-Ups“
- ...

OBERFLÄCHE

The screenshot shows the ThingWorx Composer interface. On the left, there is a sidebar with categories like MODELING, ANALYTICS, and VISUALIZATION. A large arrow points from the 'Things' section in the sidebar to a callout box labeled 'Bearbeitungsfenster für die ausgewählten Funktionen'. The main area displays a table titled 'Things' with columns for View, Name, Description, and Modified. The table lists various objects, including 'dibse-raspi3', 'die_thing', and several entries starting with '920417_18FISA_00_Dummy'. A detailed description for '910417_Raspberry' is shown: 'Daten des SensorHAT des Raspberry PI'. Another entry, '920417_18FISA_00_Lucky_Shield', is described as 'TGM Lucky Shield'.

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"/>			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

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 left sidebar under 'MODELING' has 'Things' selected, with a sub-menu for 'New'. A callout points from the 'New' button to the 'Save' button in the 'New Thing' dialog. The 'General Information' tab is active in the dialog. The 'Name' field contains 'Schule_JahrKlasse_KatNr_Thingname'. A note 'Namenskonvention siehe nächste Folie' is overlaid on the dialog. The 'Thing Template' dropdown is set to 'GenericThing'. The right side of the screen shows a list of existing Things, including 'dibse-raspi3' and 'die thing', with a note 'Showing: 148 items'.

General Information

Namenskonvention siehe nächste Folie

Name: Schule_JahrKlasse_KatNr_Thingname

Thing Template: GenericThing

Save

Entity Information

- General Information
- Properties
- Services
- Events
- Subscriptions
- Home Mashup

Permissions

- Visibility
- Design Time
- Run Time

Active

- Home Mashup
- Avatar
- Published
- Identifier
- Last Modified Date
- Value Stream

SECURITY

2019-03-01

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, there's a sidebar with categories like MODELING, ANALYTICS, VISUALIZATION, DATA STORAGE, COLLABORATION, and SECURITY. The main area is titled 'Things' and shows a list of entities. A callout box with the text 'Klick' points to the row for '920417_18FISA_00_Dummy'. Another callout box with the text 'Nächste Folie' points to the row for '910417_Raspberry'.

<input type="checkbox"/>	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 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 illustrates the process of creating a new property for an entity named "Schule_JahrKlasse_KatNr_Thingname". The interface is divided into two main sections: "General Information" and the "Properties" editor.

General Information: Shows the entity name "Schule_JahrKlasse_KatNr_Thingname", a description field, and status indicators like "Active" and "Home Mashup".

Properties Editor: This section is highlighted with a large black border.

- Toolbar:** Includes "Save", "Cancel Edit", "To Do 2", "Manage Bindings", "Edit", "Delete", and "Duplicate" buttons.
- Left Sidebar:** Lists "ENTITY INFORMATION", "Properties" (selected), "Services", "Events", "Subscriptions", and "Home Mashup".
- Properties Table:** A grid showing columns for "Edit", "Name", "Type", "Alerts", "Additional Info", "Default Value", "Value", and "DataChange". A new row is being added with the placeholder "New Property".
- New Property Form:** Contains fields for "Name" (with a dropdown menu showing "Ganze Zahl" and "Kommazahl"), "Description", "Category", and "Alerts".
- Type Selection:** A modal or dropdown menu titled "BaseType Info" shows the selected type "STRING" and a list of other options:
 - Has Default Value: 123 INTEGER, 123 LONG, JSON, LOCATION, MASHUPNAME, MENUNAME, NOTIFICATIONCONTENTNAME, NOTIFICATIONDEFINITIONNAME, NUMBER, PASSWORD
- Aspect Options:** Includes checkboxes for "Persistent", "Read-only", and "Logged".
- Data Change Info:** A section for setting data change types.
- Buttons:** "Cancel", "Done", and "Done and Add".

PROPERTY – WERT EINSTELLEN ODER AKTUALISIEREN

The screenshot shows the 'Properties' screen for an entity named 'Schule_JahrKlasse_KatNr_Thingname'. The left sidebar includes sections for Entity Information, Permissions, Change History, and Dependencies. The main area displays a table of properties under 'My Properties'. One property, 'Sensorwert_1', is selected. A callout box labeled 'Wert aktualisieren' points to the 'Set' button in the 'Value' column. Another callout box labeled 'Wert einstellen.' points to the 'Value' input field in the same row.

Edit	Name	Type	Alerts	Additional Info	Default Value	Value
	Sensorwert_1		0 Alerts			DataChange Value

- 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_MKR1000_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 MKR1000 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_MKR1000.h"
#include "Thingworx_MKR1000_Variable.h"

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

//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(ssid, password);
    pinMode(RELAY_1,OUTPUT);
    pinMode(RELAY_2,OUTPUT);
}

void loop() {
    if (millis() - lastConnectionTime > TPOST)
    {
        //Logic for the relay
        if( myThing.getjson("REL1") == 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.

//Serial communications with computer at 9600 bauds for debug purposes
//Start the Wifi Connection
//Digital pin 1 is an output pin
//Digital pin 2 is an output pin

Der Digitalpin 1 und 2 wird als ein Ausgang gesetzt.

004_GET_RELAYS

```
//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);

//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(ssid, password);                      //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("REL1") == 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.

003_PUT_DHT11_VALUE

```
//Definition of used Libraries
#include "Thingworx_MKR1000.h"
#include "Thingworx_MKR1000_Variable.h"
#include <dht.h>

//Definition for DHT11 sensor
dht DHT;
#define DHT11_PIN 3

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

void setup() {
    pinMode(DHT11_PIN, INPUT);
    Serial.begin(9600);
    myThing.Wifi(ssid, password);
}

void loop() {
    if (millis() - lastConnectionTime > TPOST)
    {
        //Collect DHT11 Data
        int chk = DHT.read11(DHT11_PIN);
        //Send data with PUT Request to Thingworx
        myThing.put("BME280_TEMP2",DHT.temperature); //Send temperature to server platform
        myThing.put("BME280_HUM",DHT.humidity); //Send humidity to server platform

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

Einfügen der Library dht.h

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

//DHT11 Sensor is on Pin 3

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

Die Signalleitung des DHT11 ist am Pin 3 angeschlossen

Der Pin 3 ist ein Input Pin.

003_PUT_DHT11_VALUE

```
//Definition of used Libraries
#include "Thingworx_MKR1000.h"
#include "Thingworx_MKR1000_Variable.h"
#include <dht.h>

//Definition for DHT11 sensor
dht DHT;
#define DHT11_PIN 3

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

void setup() {
    pinMode(DHT11_PIN, INPUT);
    Serial.begin(9600);
    myThing.Wifi(ssid, password);
}

void loop() {
    if (millis() - lastConnectionTime > TPOST)           //Send request to server every TPOST seconds
    {
        //Collect DHT11 Data
        int chk = DHT.read11(DHT11_PIN);
        //Send data with PUT Request to Thingworx
        myThing.put("BME280_TEMP2",DHT.temperature); //Send temperature to server platform
        myThing.put("BME280_HUM",DHT.humidity);       //Send humidity to server platform

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

Es wird mit dem Befehl `DHT.read11` Befehl der Bibliothek `dht.h` die Temperatur und Feuchtigkeitswerte abgefragt. Danach wird mit dem Befehl `myThing.put(...,DHT.temperature)` die Temperatur auf das Thing mit der Property „`BME280_TEMP2`“ geschickt. Danach wird mit dem Befehl `myThing.put(...,DHT.humidity)` die Feuchtigkeit auf das Thing mit der Property „`BME280_HUM`“ geschickt.

// set DHT11_PIN to input
//Serial communications with computer at 9600 bauds for debug purposes
//Start the Wifi Connection

003_PUT_DHT11_VALUE – THINGWORX COMPOSER

In Thingworx Composer das Thing auwählen

Bei neuem Wert muss aktualisiert werden

Wert ablesen

Edit	Name	Type	Alerts	Additional Info	Default Value	Value
	# BME280_TEMP		0 Alerts	°C	20.5	Value: 0
	# BME280_HUM		0 Alerts	%	36.0	Value: 0
	# BME280_PRESSURE		0 Alerts	hPa	0.0	Value: 0
	# BME280_HEIGHT		0 Alerts	m	0.0	Value: 0
	123 LED1		0 Alerts		0	Value: 0
	123 REL1		0 Alerts		0	Value: 0

GenericThing (ThingTemplate) - Properties

Generic Properties



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

Mashups

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. At the top, there's a navigation bar with 'thingworx' and a search bar. Below it is a tab bar with 'Widgets', 'Mashups', and 'Workspace' (which is currently selected). A central workspace area contains a 'Mashup' component. To the left is a 'Widgets' palette with various categories like Auto Refresh, Blog, Bubble Chart, etc. On the right, there are three panels: 'Data', 'Session', and 'User'. A bottom panel shows 'Connections' and a 'To-Do' list.

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

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.

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 from the 'Widgets' category is selected and highlighted with a black box. A callout bubble labeled 'Drag&Drop' points to the 'Label' icon in the list. The 'label-3' properties panel is open, showing the following configuration:

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"/>

The main workspace shows a placeholder for the 'label-3' component. The right side of the interface displays the 'Available Space' and 'Default Language' settings, along with tabs for 'Data', 'Session', and 'User'. A preview window is also visible.

ANZEIGE ERSTELLEN

thingworx

Search

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

Schule_JahrKlasse_KatNr_Thingname 910417_18FISA_00_Lucky_Shield New Mashup - 2 910417_Waterbike New Mashup - 3

New Mashup Mashup Design Info Save Cancel Edit More

Widgets Mashups Workspace

Category All Filter Widgets

Expression Fieldset File Upload Folding Panel Gauge GeoTag Google Location Picker Google Map Grid

gauge-4

Drag&Drop

Temperatur

Gauge

Filter Properties

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

Connections To-Do gauge-4

Data Session User

Name Value

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 'Widgets', 'Mashups', 'Workspace', and a 'More' dropdown. A central workspace shows a 'Temperatur' (Temperature) gauge. On the left, a sidebar lists various widget categories like Expression, Fieldset, File Upload, Gauge, GeoTag, Google Location Picker, Google Map, Grid, and a selected 'gauge-4' entry. The 'gauge-4' entry has properties: Name (gauge-4), Type (Gauge), DisplayName (gauge-4), Description, Data, MinValue (0), MaxValue (100), ValueFormatter (State Formatting), and FormatNeedle (checked). A modal window titled 'Add Data' is open, containing a 'Select Entity' dropdown set to 'Things', a search bar, and a 'Search Results' panel. The 'Search Results' panel shows 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 box highlights the 'Search Results' panel. In the top right corner of the workspace, there is a large black square button with a white plus sign (+).

HINZUFÜGEN VON DATEN

Add Data

Select Entity 920417_18FISA_00_Dummy  Dynamic  

Select Services

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

Dynamic  

Selected Services

Entity Type Entity Name Service Mashup Loaded? Remove

Things 920417_18FISA_00_Dummy GetPropertyValues 

GetPropertyValues  

GetPropertyValuesAsMultiRowTable  

GetPropertyValuesVTQ  

GetPropertyValuesVTQA  

Cancel Done

VERBINDEN DER DATEN MIT WIDGET

The screenshot shows the thingworx interface for creating a new mashup. A 'Drag&Drop' box highlights the action of dragging a data source to a target. In the center, a 'Select Binding target' dropdown menu is open, showing options like '# Data', '# MinValue', '# MaxValue', '-T Legend', and '-T ToolTipField'. A callout box points to the '# Data' option. On the right, the 'Data' tab of the properties panel is selected, showing a list of connected data sources under 'Returned Data'. One item is highlighted: '123 Sensorwert_1'. Below this, a 'Connections' window displays a flow from a 'GetPropertyValues' block to a 'gauge-4' block, with '123 Sensorwert_1' as the data source. A callout box at the bottom right states: 'In diesem Fenster werden alle verbundenen Daten zu einem Widget angezeigt'.

thingworx

New Mashup

Widgets, Mashups, Workspace

Category: All

Filter Widgets

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

panel-2

Name Value

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

Temperatur

Select Binding target

- # Data
- # MinValue
- # MaxValue
- T Legend
- T ToolTipField

Connections

- Things_920417_18FISA_00_Dummy
- GetPropertyValues
 - All Data
 - 123 Sensorwert_1
- # Data
- gauge-4

In diesem Fenster werden alle verbundenen Daten zu einem Widget angezeigt

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

Data, Session, User

Things_920417_18FISA_00_Dummy

- GetPropertyParams
 - Parameters
 - Returned Data
 - All Data
 - 123 Sensorwert_1
 - name
 - description
 - thingTemplate
 - tags
 - Selected Row(s)

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. A callout box points to the 'Refresh Now' button in the center of the screen, which is highlighted with a black border. Another callout box points to the '# RefreshInterval' field in the properties panel on the left, where the value '30' is entered. The central workspace displays a gauge widget titled 'Temperatur'. The right side of the interface shows the 'Data' tab with a 'GetPropertyValue' block connected to a 'Things_920417_18FISA_00_Dummy' data source.

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 Designer interface with the following components and annotations:

- Top Bar:** thingworx, Search, New Entity, Import/Export, Monitoring, Help, Learning Connector, User iot-seminar16.
- Mashup Tabs:** Schule_JahrKlasse_KatNr_Thingname, 920417_18FISA_00_Lucky_Shield, 123, 910417_Waterbike.
- Toolbars:** Widgets, Mashups, Workspace; View Mashup, Save, Cancel Edit; Preview, Available Space, Default Language.
- Left Sidebar:** Category All, Filter Widgets, Auto Refresh, Blog, Bubble Chart, Button, Checkbox, Contained Mashup, Dashboard, Data Export, Data filter, Date Time Picker.
- Properties Panel:** autorefresh-5, Filter Properties, Name, Value:
 - T- Id: autorefresh-5
 - T- Type: Auto Refresh
 - T- DisplayName: autorefresh-5
 - T- Description:
 - # RefreshInterval: 30
 - AutoRefresh: checked
 - ShowControls: checked
 - # AutoRefreshTabSe...: 0
 - T- Label: Refresh Now
- Main Area:** A dashboard titled "Temperatur" containing a "Gauge" widget. A "Drag&Drop" callout points to the gauge's configuration area.
 - Configuration Area:** Shows a "Widget" dropdown, "Configure Bindings", and a "Refresh" button with a lightning bolt icon.
 - Connections:** An "autorefresh-5" entity is connected to a "Things_920417_18FISA_00_Dummy" entity, which in turn connects to a "GetPropertyValues" action and a "Data" entity named "gauge-4".
 - Data Panel:** A table showing Name and Value columns.
- Annotations:** A large curved arrow points from the "autorefresh-5" properties panel to the "Widget" dropdown in the gauge's configuration area, indicating the flow of control from the refresh entity to the gauge's refresh button.

HINZUFÜGEN EINES DIAGRAMMES

The screenshot shows the thingworx interface for creating a mashup. A 'Drag&Drop' box highlights the 'Time Series Chart' widget from the left sidebar. An arrow points from this box to the chart area in the center. Another arrow points from the 'Time Series Chart' icon in the center to the 'GetPropertyValues' node in the right sidebar. The right sidebar also displays a table of event properties for 'Things_920417_18FISA_00_Dummy'.

Widgets | **Mashups** | **Workspace**

Category: All

Filter Widgets

- Tag Picker
- TagCloud
- TextArea
- TextBox
- Time Selector
- Time Series Chart**
- Tree
- Validator
- Value Display
- Vertical Slider

timeserieschart-6

Filter Properties

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	

123 Mashup | **Design** | **Info**

View Mashup | **Save** | **Cancel Edit**

Preview | Available Space | Default Language

Data | **Session** | **User**

Things_920417_18FISA_00_Dummy

GetPropertyValues

- Parameters
- Returned Data
- All Data
- description
- name
- Sensorwert_1
- tags
- thingTemplate
- Selected Row(s)

ServiceInvokeCompleted ↳

AllDataChanged ↳

SelectedRowsChanged ↳

HINZUFÜGEN EINES DIAGRAMMES

Add Data

Select Entity  920417_18FISA_00_Dummy x

Dynamic ? □

Select Services

querypropertyhistory x

All

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



Selected Services

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

Cancel

Done

HINZUFÜGEN EINES DIAGRAMMES

The screenshot shows the thingworx interface for creating a Mashup. A 'Select Binding Target' dialog box is open, centered over a panel titled 'Temperatur'. The dialog lists four options: 'Data', 'DataSource1', 'DataSource2', and 'DataSource3'. The 'Data' option is highlighted with a black border. A large black arrow points from the text 'Drag&Drop' to the 'Data' button. The interface includes a left sidebar with various widget categories like 'Widgets', 'Mashups', and 'Workspace'. On the right, there's a 'Returned Data' pane listing items such as 'All Data', 'description', 'name', and 'Sensorwert_1'. Below it is a 'Things' pane showing a table of properties for 'Things_920417_18FISA_00_Dummy1'. The main workspace contains a panel titled 'panel-2' which is currently empty.

Drag&Drop

Select Binding Target

- Data
- DataSource1
- DataSource2
- DataSource3

Temperatur

On Refresh Now

panel-2

panel 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

Widgets Mashups Workspace

Category All

Filter Widgets

Tag Picker TagCloud TextArea TextBox Time Selector Time Series Chart Tree Validator Value Display Vertical Slider

123 Mashup Design Info View Mashup Save Cancel Edit Preview Available Space Default Language

Returned Data

- All Data
- description
- name
- Sensorwert_1
- tags
- thingTemplate

Selected Row(s)

All Data

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 main area displays a panel titled "Temperatur" containing a "Gauge" widget and a "Time Series Chart". A context menu is open over the "Time Series Chart" with options "On", "Refresh Now", "Widget", "Configure Bindings", and "Refresh". A large black callout box with the text "Drag&Drop" has an arrow pointing from the "Time Series Chart" towards the "Selected Row(s)" section in the right sidebar.

Widgets **Mashups** **Workspace**

Category: All

Filter Widgets

- Tag Picker
- TagCloud
- TextArea
- TextBox
- Time Selector
- Time Series Chart
- Tree
- Validator
- Value Display
- Vertical Slider

panel-2

Filter Properties

Name	Value
-T- Id	panel-2
-T- Type	Panel
-T- DisplayName	panel-2
-T- Description	
Style	<input type="color"/>
HideScrollbars	<input checked="" type="checkbox"/>
ShowDataLoading	<input checked="" type="checkbox"/>
ResetInputsToDefaultValue	<input type="checkbox"/>
# Z-index	10

Connections **To-Do**

Preview Available Space Default Language

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

Data **Session** **User**

Returned Data

- All Data
- description
- name
- Sensorwert_1
- tags
- thingTemplate
- Selected Row(s)

Things_920417_18FISA_00_Dummy1

QueryPropertyHistory

- Parameters
- Returned Data
- All Data
- Selected Row(s)

Things_920417_18FISA_00_Dummy1:Qu...

Name	Value
oldestFirst	<input type="checkbox"/>
# maxItems	<input type="checkbox"/>
endDate	<input type="checkbox"/>
query	<input type="checkbox"/> Edit Query
startDate	<input type="checkbox"/>
ServiceInvokeCompleted	<input type="checkbox"/>
AllDataChanged	<input type="checkbox"/>
SelectedRowsChanged	<input type="checkbox"/>

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

The screenshot shows the Thingworx interface with the following details:

- Header:** thingworx, Search, New Entity, Import/Export, Monitoring, Help, Learning Connector, User iot-seminar16.
- Left Sidebar:** Networks, Projects, Model Tags, Integration Connectors, ANALYTICS (Data Analysis Definitions), VISUALIZATION (Mashups, Masters, Gadgets, Dashboards, Menus, Media, Style Definitions, State Definitions), DATA STORAGE (Data Tables, Streams), Value Streams (selected), Data tags, Persistence Providers, COLLABORATION, SECURITY.
- Current View:** Value Streams. The "Value Streams" tab is selected in the sidebar. A modal dialog titled "Choose Template" is open, listing templates: RemoteValueStream (Remote Value Stream) and ValueStream (Value Stream). The "ValueStream" row is highlighted with a dark gray background.
- Toolbar:** + New, View, Edit, Duplicate, Delete, Permissions.
- Table:** Shows a list of items with columns: View, Name, Modified. The list includes various entries such as 920417_18FISA_00_Lucky_Shield, 920417_18FISA_00_Dummy, and TU-18W, all modified on 2019-03-20 at 10:35:30.972.
- Bottom:** URL https://twx.hlt.schule/Thingworx/Composer/index.html#.

A large callout arrow points from the "Value Streams" button in the sidebar to the "ValueStream" entry in the "Choose Template" dialog. Another callout arrow points from the "Choose" button in the dialog back to the "Value Stream" entry in the list on the right.

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 'New Entity', 'Import/Export', 'Monitoring', 'Help', 'Learning Connector', and a user account 'iot-seminar16'. The main workspace displays 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 'PERMISSIONS' section includes 'Visibility', 'Design Time', and 'Run Time'. The 'DEPENDENCIES' section lists 'Entity Depends On' and 'Uses This Entity'. At the top right, there are buttons for 'Save' (with a mouse cursor arrow pointing to it), 'Cancel Edit', and 'To Do'. A callout line points from the 'Save' button to the 'Active' checkbox in the 'General Information' section, which 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 is a toolbar with icons for home, search, and various entity types like 'Schule_JahrKlasse_KatNr_Thingname', '920417_18FISA_00_Lucky_Shield', '920417_18FISA_00_Dummy', '123', and '920417_18FISA_00_Dummy_ValueStream'. The main content area displays the 'General Information' tab for a 'Thing' named '920417_18FISA_00_Dummy'. The 'General Information' section contains fields for Name (920417_18FISA_00_Dummy), Description, 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 (Browse...), Last Modified Date (2019-03-20 10:25:55.248), and Value Stream (920417_18FISA_00_Dummy_ValueStream). A large black arrow points from the 'Value Stream' field to the 'Value Stream' section of the page. On the left side, there is a sidebar with sections for 'ENTITY INFORMATION' (General Information, Properties, Services, Events, Subscriptions, Home Mashup), 'PERMISSIONS' (Visibility, Design Time, Run Time), 'CHANGE HISTORY' (Change History), and 'DEPENDENCIES' (Entity Depends On, Uses This Entity). Below the sidebar is a rich text editor for 'Documentation'.

EINSTELLUNGEN AM THING FÜR DAS SPEICHERN VON DATEN

The screenshot shows the thingworx interface for managing entities. A specific entity, "920417_18FISA_00_Dummy", is selected and being edited. The "Properties" tab is active, displaying a table of properties. One property, "Sensorwert_1", is highlighted with a black box. The "Name" column for this property contains "123 Sensorwert_1". The "Type" column shows "123 INTEGER". In the "Aspects" section, the "Logged" checkbox is checked, also highlighted with a black box. Other aspects like "Persistent" and "Read-only" are shown but not checked.

Entity	Name	Type	Alerts	Additional Info	Default Value	Value	DataChange
Sensorwert_1	123 Sensorwert_1	123 INTEGER	0 Alerts		25	Value: 0	Set

Properties

General Information

Events

Subscriptions

Home Mashup

Permissions

Visibility

Design Time

Run Time

Change History

Change History

Dependencies

Entity Depends On

Uses This Entity

GenericThing (ThingTemplate) - Properties

Generic Properties

LETZTE EINSTELLUNGEN AM DIAGRAMM

thingworx

Schule_JahrKlasse_KatNr_Thingname

123 Mashup Design Info

Widgets Mashups Workspace

Category All

Filter Widgets

Tag Picker TagCloud TextArea TextBox Time Selector Time Series Chart Tree Validator Value Display Vertical Slider

panel-2

Filter Properties

Name	Value
-T- Id	panel-2
-T- Type	Panel
-T- DisplayName	panel-2
-T- Description	
Style	
<input checked="" type="checkbox"/> HideScrollbars	
<input checked="" type="checkbox"/> ShowDataLoading	
<input checked="" type="checkbox"/> ResetInputsToDefaultValue	
# Z-index	10

# 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...	<input type="checkbox"/>
AllowSelection	<input checked="" type="checkbox"/>
EnableHover	<input checked="" type="checkbox"/>
ShowXAxisGrid	<input checked="" type="checkbox"/>
ShowYAxisGrid	<input checked="" type="checkbox"/>
GridStyle	
DataField1	Sensorwert_1 ▾
DataLabel1	Wert1
-T- SeriesType1	Use Chart Setting ▾
-T- SeriesMarkerType1	Use Chart Setting ▾

thingworx

Schule_JahrKlasse_KatNr_Thingname

123 Mashup Design Info

View Mashup Save Cancel

Widgets Mashups Workspace

Category All

Filter Widgets

Tag Picker TagCloud TextArea TextBox Time Selector Time Series Chart Tree

Temperatur

Gauge

FERTIGES MASHUP IM BROWSER



Show/Hide Log

Show/Hide Debug Info

Reload

Default

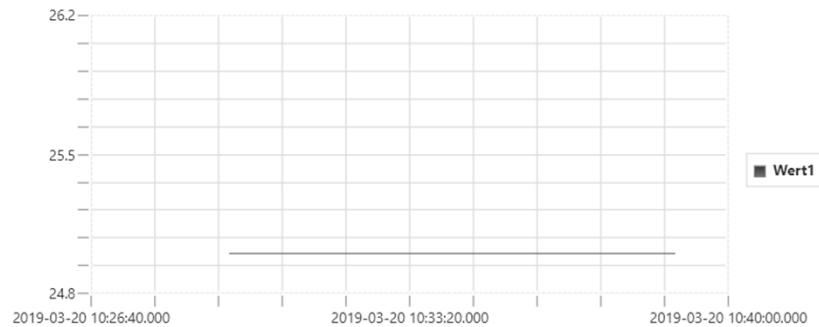
FullScreen

Temperatur



On

Refresh Now



Diese Adresse kann in einem
Browser eines digitalen
Endgerätes eingegeben werden.
Nach der Eingabe der
Logindaten ist diese Seite
sichtbar.