



Elvis Beispielbeschreibung zu Projekt „IoT-Geräte“

INF - Beschreibung (.pdf)

Produkt: Elvis 3

Version: ab 3.3.238

Stand: 2019-26-02

Autor: Niko Paßmann

Ziel des Dokuments

Dieses Dokument soll als Beschreibung für das Beispielprojekt „IoT-Geräte“ dienen und die darin enthaltenen Projektschritte verdeutlichen.

Aufgabenstellung

In diesem Beispiel wird gezeigt, wie IoT-Geräte in Elvis integriert werden können. Konkret wird das Beispiel anhand von shelly1, 2 und H&T durchgeführt. Die Projektierung bleibt für Elvis Terminal, -Web,-Viewer und Viewer direct gleich. Ein allgemeines Beispiel für Elvis Viewer direct finden Sie [hier](#). Mit Elvis sind Sie frei in der Gestaltung Ihrer Bedienoberflächen. Dieses Projekt veranschaulicht die reine Funktionalität möglicher Steuerelemente. Über den Button „Example Room“ finden Sie beispielhaft eine Bedienseite, wie sie für einen einzelnen Raum aussehen könnte.

Vorgehensweise

Nach Einrichtung des MQTT Port, wie im verlinkten [Beispiel](#) beschrieben, muss dieser um einen weiteren Anschluss erweitert werden. Für diesen Anschluss wird eine xml (Steuerdatei) hinterlegt, welche zur Steuerung einer IoT-Lampe benötigt wird.

Eigenschaften	
▼ Allgemein	
Beschreibung	
Name	shellyBulb
Text	
▼ Prozessanschluss	
▼ Konfiguration	Type="Itgmbh.Elvis.MQTT.MQ...
Abfrageintervall beim Hochlauf (ms)	200
Certificate Store	LocalMachine
Client Certificate Name	
Host Name	192.168.1.83
Leseabfragen beim Hochfahren puf...	<input type="checkbox"/> False
Lesen-Abstand (ms)	0
Password	
Port	1883
Prefix	
Secure	<input type="checkbox"/> False
Service Configuration	bulb.xml
User Name	
Warten auf Lesebestätigung (ms)	500
Warten auf Schreibbestätigung (ms)	0
Startabfrage überspringen	<input type="checkbox"/> False
Treibertyp	MQTTDriver

Knowledge Base

Projektbeispiel "IoT-Geräte"



Die xml enthält neben der Information Contenttyp Json noch den dazugehörigen String, welcher alle einstellbare Parameter beinhaltet. Hier werden die zu setzenden **Parameter maskiert**. Z.B. {..."mode":{"mode"}, ...} original {..."mode":color, ...}

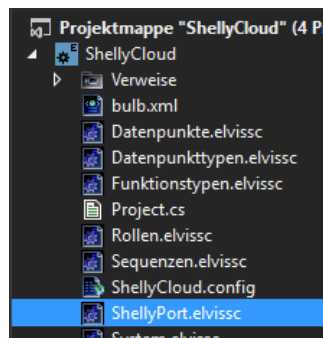
```
ShellyPort.elvissc
{<?xml version="1.0" encoding="utf-8" ?>
<!-- Configuration file for Elvis3 MQTTDriverConfiguration -->
<MQTTDriverConfiguration
PayloadPattern="{\"ison\":false,\"mode\":\"{mode}\",\"red\":\"{red}\",\"green\":\"{green}\",\"blue\":\"{blue}\",\"white\":0,\"gain\":\"{gain}\",\"temp\":\"{temp}\",\"brightness\":\"{brightness}\",\"effect\":0}'
ContentType='json'
```

Die Prozessadresse für diesen Datenpunkt sieht nun wie folgt aus:

`shellies/shellybulb-3CC597/color/0/set/mode -W---`; `shellies/shellybulb-3CC597/color/0/status/mode --I--`
Der erste Teil ist das **Topic**, auf welches das Gerät hört. „**Mode**“ ist der zu ändernde Parameter im Json String.

Für die Rolladensteuerung mit shelly2, die Lichtsteuerung der Bulb und die Anzeige der Sensordaten wurden jeweils Parts erstellt. Dies ermöglicht dieselbe und einfache Darstellung und Bedienung über alle Bedienseiten hinweg. Des Weiteren muss bei Änderung nur der Part angepasst werden. Weiter Infos zur Erstellung von Parts finden Sie in der Elvis Hilfe oder Anleitung.

Die Datenpunkte zu den einzelnen Geräten finden Sie in der Datei Shellyport.elvissc in der Projektmappe.



Name	Datenpunktyp	Prozessanschl...	Prozess-Adressen
Bulb Red	String	shellyBulb	shellies/shellybulb-3CC597/color/0/set/red -W---; she
Bulb Green	String	shellyBulb	shellies/shellybulb-3CC597/color/0/set/green -W---; s
Bulb Blue	String	shellyBulb	shellies/shellybulb-3CC597/color/0/set/blue -W---; sh
Bulb OnOff	String	shelly	shellies/shellybulb-3CC597/color/0/command -WI---; !
Bulb Mode	String	shellyBulb	shellies/shellybulb-3CC597/color/0/set/mode -W---; s
Bulb Gain	String	shellyBulb	shellies/shellybulb-3CC597/color/0/set/gain -W---; sh
Bulb temp	String	shellyBulb	shellies/shellybulb-3CC597/color/0/set/temp -W---; sl
Bulb brightness	String	shellyBulb	shellies/shellybulb-3CC597/color/0/set/brightness -W

Über das grüne Kreuz lassen sich weitere Datenpunkte hinzufügen.

In dem Ordner Funktionen sind die Datenpunkte nun an die passende Funktion gebunden, welche an den entsprechenden Part gebunden wird.



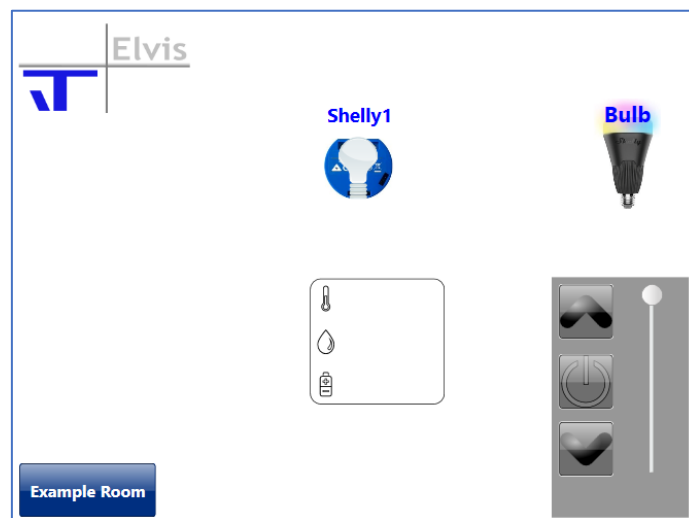
Name	Text	Funktionstyp	Rollen
ShellyBulb Color		ShellyColor	▼
ShellyBulb White		ShellyWhite	▼
ShellyHandT	Shelly H&T	ShellyHandT	▼
ShellyShutter	Shelly2 Roller	ShellyShutter	▲

Name	Datenpunkt
MoveStop	MoveStop=[Datenpunkte\shelly2\Shutter OpenClose].Value
Position	Position=[Datenpunkte\shelly2\Shutter Position].Value

Bedienseiten

Startseite:

- OnOff-Button: shelly1
- CommandButton: Bulb -> Seitenwechsel zur Lichtsteuerung
- CommandButton: „Example Room“ Seitenwechsel
- Parts für Sensor und Rollo -> ClientLib -> Parts: HandT und shutter



Bulb-Seite:

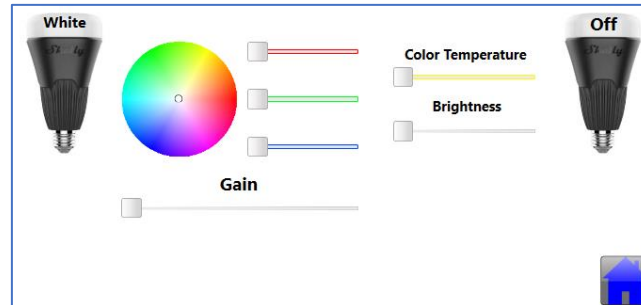
- OnOff-Button (links): Wechsel Modus White und Color
- OnOff-Button (rechts): Lampe Ein Aus

Knowledge Base

Projektbeispiel “IoT-Geräte”



- Part (links): Steuerung RGB und Gain
- Part (rechts): Steuerung Farbtemperatur und Helligkeit



„Example Room“

- durchsichtiger OnOff-Button über Lampe mit Statusanzeige darunter
- CommandButton über Fenster: Öffnet Popup mit Rollo-part
- CommandButton: Durchsichtig über Sensor Part öffnet Popup für Heizungsregelung



Nachdem Sie in der Projektmappe die zu startenden Projekte ausgewählt haben (klick mit rechts auf die Projektmappe -> Startprojekte festlegen), können Sie mit STRG und F5 oder unter Debuggen → starten ohne Debugging das Projekt im Testmodus starten.