



# HoloLens Vuforia Studio Creo Illustrate

P.Fürli, M.Schubert

# Inhalt

m.schubert

Vorbereitung	
Installation von Vuforia Studio	
Einrichten der Hololens	4
Arbeiten mit Vuforia Studio	7
Erstes Projekt "Wagenheber"	7
Grundlagen	7
Animation hinzufügen	13
Export und Import	15
Projekt "Richtvorrichtung"	16
Vorbereitungen	16
Objekteigenschaften ändern	17
Zusatzinformationen einfügen	22
Sprachsteuerung	23
Arbeiten mit Illustrate	24
Installation	24
Vorbereitung der Daten in Creo	25
Beispiel Getriebemontage	25
Verwendung in Studio	28
Animationen aus Creo Mechanismus importieren	29
Zusatzinfo "Scharniere"	30
Beispiele	31
Modellflieger (Dreidecker – Fokker DR1)	31
Vorbereitung in TWX-Composer	32
Vuforia Studio	33
JavaScript für Luftschraube	
Loggen von Werten	40
Timer verwenden	41
Funktionen verwenden	41
Bedingungen verwenden	42
Gesamter Code zum Drehen des Propellers	43
Drehzahl und Statusanzeige	44
Steigflug und Looping ausführen	47

# Vorbereitung

Installation von Vuforia Studio

Anlegen eines PTC-Accounts:

https://support.ptc.com/appserver/common/acco	unt/createConfirm.jsp	
📚 ptc	:	Tools Resources Martin Schubert
Home > PTC.com-Online-A	Accounts > Neues Online-Konto einrichten	
Neues Onli	ine-Konto einrichten	
Neukunde Part	iner Standard Academic	
Alle mit einem * geken	nzeichneten Felder sind Pflichtfelder.	Academic-Konto: Dieses Konto wird zum Herunterladen vor
* Vorname:	Martin	Software für Schulen sowie für den Zugriff auf die Systeme des Academic Program und auf die PTC Communities verwendet. PTC nutzt die E-Mail-Adressen von Schülern und
* Nachname:	Schubert	Studenten generell nicht für Marketingzwecke.
* E-Mail:	Dieser Benutzername ist bereits vorhanden. Falls dies Ihre E-Mail-Adresse ist: Passwort zurü	Hinweis: PTC nimmt wissentlich keine persönlichen Daten von Minderjährigen ohne Zustimmung der Eltern an. Interessenten, die jünger als 13 Jahre sind, geben bitte auch
* E-Mail bestätigen:	Dieser Benutzername ist bereits vorhanden. Falls dies Ihre E-Mail-Adresse ist: Passwort zurü	die E-Mail-Adresse ihrer Eltern an, damit wir für dieses Konto die Zustimmung der Eltern einholen können. Sollten Sie als Erziehungsberechtigter das PTC.com-Konto entfernen oder korrigieren wollen, wenden Sie sich bitte an den
E-Mail-Adresse der El	tern:	PTC.com Account Support.
* Academic Program:	Other ~	
* Titel:	Professor ~	
* Schule:	HTL-Ottkring	
Bundesland/Kanton:	~	
* Land:	Österreich v	
* Passwort:	••••	

# Download und Installation der Software via https://studio.vuforia.com

📚 vuforia <sup>.</sup>	studio portal		Startseite Suppor	rt + Kaufon	📚 vuforia <sup>,</sup> 🛛	studio portal		Startseite Suppr	rt + Kaufen Konto +
Ihre Anfrage wurde ge	Trouen Sie sich! enehmigt! Jetzt müssen Sie nur r damit	Klicken Sie auf d noch auf "Test starten" klicken Sie Ihre Telebolien höten kör	ie Schaltfläche und wir bringen den Experi nnen.	ence Service zu	Jetzt ka Der Experience Service	nn der Spaß begir ist bereit, Ihre Erlebnisse zu hos Finden Sie met	nnen! Erstellen Si sten! Laden Sie Vuforia Studio hr heraus in den nachstehend	e noch heute e herunter, und fangen Sie ar len Ressourcen!	in Erlebnis. AR-Erlebnisse zu erstellen.
		Test starten	<b>)</b>	r C	Server	30 Verbleibende Tage	o 0 Anzahl der Ansichten	0 Veröffentlichungen	Projekte
	)		$\bigcirc$			VL	uforia Studio Ressourc	en	
				Projek	(i) COMMON	n 🔯 Milano	IDATEMBANK	YOUTUBE-KANAL	🕼 HUT-CINTR
	Vu	iforia Studio Ressource	n				Vuic la Studio		
а станалит	n. 🔘 Musasea	SATENBIANK	YOUTURE KANAL	HUF-CINT		Für Wind herunteri	lows aden Eur arritte Handbuch (PDF)+	Mac herunterladen	

## Einrichten der Hololens

Die Betriebssystemsoftware sollte auf dem neuesten Stand sein.

Die Hololens muss bereits mit einem WLAN mit Internetanbindung verbunden sein.

im Store nach Vuforia suchen und die App installieren



Nach erfolgreicher Installation unter Settings den Eintrag <u>For developers</u> anwählen und <u>developer features</u> und <u>device portal</u> aktivieren:



Ermittlung der IP-Adresse (dieser Schritt ist für die Verwendung mit Vuforia Studio nicht erforderlich)



Beim aktiven Netzwerk <u>Advanced options</u> anklicken und im sich öffnenden Fenster nach unten scrollen.

Hier findet man die IP-Adresse der HoloLens:

← Settings		¢ ×
බ Schubi		
SSID: Protocol	Schubi 802.11ac	
IPv4 address:	192168.0.94	
IPv4 DIVS Servers	1742-1003-0142	
DNS suffix search list:	schubi	
Manufacturer:	Broadcom	
Description	Broadcom 802 that Wireless PCIE Full Dongle Adapter	
Driver version:	1.558.53.36	
Physical address	94-9A-A9-1F-F1-C8	
100		

Nun kann man auf einem Rechner (der im selben Netzwerk sein muss) diese IP in den Browser eintragen. Warnungen bezüglich der Sicherheit ignorieren – bzw. bestätigen, dass es OK ist.



Nun werden ein Name und Passwort abgefragt. Wenn man das nicht weiß, dann mindestens drei Mal Name und Passwort leer lassen und Anmelden drücken.

	Anmelden https://192.168.0.94	
	Nutzername	
	Passwort	
	Abbrechen es öffnet sich diese Sei	ite
	Hicrosoft Set up access	
ĺ		
	$\pi^{k}$ Credentials reset	
	Request a PIN to set your user name and password	
	Request pin	
	Security certificate	
	Seeing a "certificate error" in your browser? You can fix that and get a secure connection	
	1. Make sure you are on a secure network (USB or a Wi-Fi network you trust)	
	2. Download this device's certificate	
	3. Install the certificate in the "Trusted Root Certification Authorities" store	
	4. Restart your browser	
	For more information, see the documentation.	
	De	~

Request PIN anklicken und den in der

HoloLens angezeigten Pin im folgenden Fenster eintragen:

6 In schubert		Dokumentation Hololens Seminar 2018&2019 Mödling
Microsoft HoloLens	Set up access	
	م <sup>لا</sup> Credentials reset	
	PIN displayed on your device: 4011076	
	New user name: Martin	
	New password:	
	Confirm password:	
	Pair	

Nun kann man sich mit dem angegeben Namen und Passwort in das WEBIF der HoloLens einloggen. Hier kann man zum Beispiel den Livestream im Browser anzeigen lassen, oder auch Screenshots herunterladen:



# Arbeiten mit Vuforia Studio

# Erstes Projekt "Wagenheber"

# Grundlagen

n.schubert

Vuforia Studio läuft als Webapplikation am lokalen Rechner und wird daher in einem Browser dargestellt. Es ist ratsam **ausschließlich Google Chrome zu verwenden!** 



Hier mit dem PTC-Account anmelden.

Die Willkommensseite öffnet sich – nun die Sprache auf Englisch umstellen:



n.schubert

Mit dem Plus-Symbol rechts oben ein neues Projekt erstellen:



... für die HoloLens 3D Eyeware auswählen.



Viele Funktionen in der Oberfläche lassen sich per "drag and drop" ausführen.

Zuerst überprüfen ob wir mit dem richtigen Experience Service arbeiten:

# Project→Info:

	Info Editor
Description	Max. 100 characters
Experience Service	<ul> <li>cp://233.00 **********************************</li></ul>
Access	Public         •           If set to Public, users can access Experiences without authentication. Authentication may be required to view Thir information, see Configuring Public Access to ThingWorx in the Help Center.           Image: Allow download for offline viewing
Minimum Screen Width	Holographic •
Project Thumbnail	Drop image or click to upload

- Mittels Validate den Status überprüfen (Anmeldung mit PTC-Account erforderlich) gegeben Falls richtigen Server auswählen.
- Der Angezeigte QR-Code enthält den Link zum Server, der von der View-App auf der HoloLens benötigt wird!
- Hier kann auch gleich ein Vorschaubild für das Projekt hochgeladen werden.
- Access sollte auf Public stehen.

Model hinzufügen – mittels "drag&drop":

- Das Model ist vorerst nur ein Platzhalter – bzw. Hülle. Es muss erst mit Daten "befüllt" werden:



Die eingespielten Daten (3D-Daten, Bilder, …) werden über Resources hinzugefügt. Dies kann entweder vor der Erstellung des Model geschehen (linker Menübereich) oder während der Model Generierung (rechts oben)

Add Resources:





PVZ hinzufügen..

CAD Optimizer für große Dateien – es werden 4 Größen erzeugt.



Es wird immer das oberste Koordinatensystem in der Baugruppe für die Einbaubedingungen verwendet.

Die X, Y, Z Koordinaten sind in Meter angegeben.

Zurzeit gibt es Probleme, wenn die Baugruppe Unterbaugruppen enthält. Es ist sinnvoll die Baugruppen in Einzelteile zu verschmelzen.

Nun muss man noch ein Target einfügen (mittels drag&drop) – wir verwenden vorerst "Spatial Target", dies lässt ein freies Platzieren der AR im Raum zu.

Die gewünschten Properties im rechten Bereich einstellen:

SPATIAL TARGET - DETAILS	
(i) Remove	
✓ PROPERTIES	
$\rightarrow$ ) X Coordinate	
0	Vor dem Publizieren der AR kann man noch unter Preview die
$\rightarrow$ ) Z Coordinate	Funktion im Browser testen:
0	TOTIKTOTTTTT DIOWSET TESTETT.
←) □ Tracked	Letzter Schritt ist das veröffentlichen der AR mittels Publish:
→) ✔ Enable Pan Gesture	Save     Proview     Publish     Share ×
→) <mark>✓</mark> Enable Rotate Gesture	
→)   Enable Scale Gesture	
Studio ID	ACHTUNG Popups zulassen:
spatialTarget-1 🔗	
Friendly Name	e a construction of the second s
spatialTarget-1	Pop-ups blockiert:
	nttp://iocainostsuouerineodxneight=underineo
←) Tracking Acquired IS	localhost:3000 immer zulassen
←) Tracking Lost JS	O Weiterhin blockieren
	Verwalten Fertig
	⇒1 Besourc

# OPTIONAL:

1.schubert

Damit die AR in der HoloLens auch ohne Sprachsteuerung aufgerufen werden kann ist es gut unter Configuration  $\rightarrow$  Experiences noch ein Thingmark anzugeben:

Solution My Projects	> 01_Wagenheb	ber 🔵		
Save	Preview	Publish	Share 🗸	
PROJECT	r			
<ul> <li>CONFIGURATION</li> </ul>				Edit Experiences
Themes				
Experiences				
Info				
My ThingMarks				New Experience
✓ VIEWS +				
► Home	⇒)			
Home.js				
✓	r i i	ThingMark Association	ThingMarks	
魚 model-1				
spatialTarget	-1	litie	01_Wagenheber	
✓ STYLES		ThingMark		
Application		Thingwidtk	Enter or select a ThingMark	
✓ RESOURCES +		Initial View	28417:1	
✓ Uploaded +		Initial View	28417:2 28417:3	

Wenn ein Thingmark angegeben wurde wird diese AR nicht in der library angezeigt, sondern kann nur durch Scannen das Thingmarks geladen werden!

Letzter Schritt ist das veröffentlichen der AR mittels Publish:

😣 My Projec	ts > 01_Wagenheber			heber 🔵
Save	Preview	Publish	Share 🗸	Uploading project
PRO.	JECT			(1.9 MB of 4.2 MB completed)
<ul> <li>CONFIGURATION</li> </ul>				N
Themes				フ_

Nachdem der Upload abgeschlossen wurde kann man in der HoloLens die Vuforia View App aufrufen.

Erster Schritt ist dabei das Scannen des QR-Codes (enthält die Server-URL)

Nun kann mit den Sprachkommandos "Hey View" … warten auf Signalton …"Show library" und der Auswahl der gewünschten AR das Teil am Boden platzieren.

### Animation hinzufügen

Dafür muss im pvz-File bereits eine Animation vorhanden sein.

Im 3D-Container das entsprechende Model anwählen und bei den Properties (rechts) nach unten scrollen und bei Sequence die Animation auswählen: ACHTUNG – Programm genehmigt sich eine kurze Watezeit 😊



Gleichzeitig kann man auch noch die Einträge Studio ID und Friendly Name anpassen (das ist nicht zwingend erforderlich)

Die Animation soll mit einem Doppelklick gestartet werden und mit Swiperight wieder in den Ausgangszustand gebracht werden.

MODEL - DETAILS 📋 Remove > APPLICATION PARAMETERS + ✓ PROPERTIES ✓ APPLICATION EVENTS +  $\rightarrow$ ) Resource + ←) > doubletap JS ~ ←) > hold JS Wagenheber.pvz ←) > swipeback □S →) Scale ←) > swipeforward JS 1.0 ←) > swipeleft JS →) X Coordinate ←) > swiperight JS 0 ←) > swipeup JS ✓ EXTERNAL DATA + →) Y Coordinate

Dazu das Menü Data einblenden:

# Dann den doubletab nach Wagenheber ziehen:

Design - Vuforia Studi	• × +						-
← → C () loc	alhost:3000/design/0	1_Wagenheber					🖬 🕶 🗟 🚖 😁 🗄
S My Projects >	01_Wagenheber	•					
Save	Preview	Publish	Share 🗸				
PROJECT		WIDGETS		CANVAS	→)  Occluding	DATA	
CONFIGURATION	✓ TA	RGETS	* 🛛 🖻 🖕 🗠			> APPLICATION PARAMETERS +	
Themes	6 <sup>2</sup>	Image Target			→) Always on top	✓ APPLICATION EVENTS +	
Experiences	84-	Model Target			→) Onarity (LOnarity)	←) ❤ doubletap 15	
Info	-				c) open (	Voice Alias	
My ThingMarks	· · ·	Spatial Target					
✓ VIEWS +	G.	ThingMark			→) Sequence	Voice Response	
<ul> <li>Startseite</li> </ul>	→) 🗸 AL	JGMENTATIONS			Choose a resource		
Startseite.js	P	3D Gauge			Animation		
Solution States					(a) Steer	Voice Help	
.m wagenneber		5D Image		>. $>$			
spatialTarget-1		3D Label			S) Current Step	←) > hold IS	
✓ STYLES	a.	Model			←) Playing	←) > swipeback IS	
Application		Model Item	<			←) > swipeforward JS	
✓ RESOURCES +		modernem			Studio ID	←) > swipeleft _JS	
Vienekskesien	V 01	THER			model-1	←) > swiperight IS	
Wagenheber.jpg	(IML)	TML Text			Friendly Name	←) > swipeup 15	
> Default					Wagenheber	✓ EXTERNAL DATA +	
,					✓ SERVICES		
					→) Forward		
					→) Play		
					→) Play All		
					→) Reset		
					→) Rewind		
					→) Stop		
					✓ EVENTS		
					←) Play Started IS		
					←) Play Stopped IS		
					←) Model Loaded US		
			Z X		←) Click JS		
					<ul> <li>Acknowledge Reguested 35</li> </ul>		

#### Nun das Binding auf Play setzen:

Select Binding Target	
app Doublebo	Selben Vorgang noch für swiperight mit dem Binding Reset ausführen.
Services Forward Play	Die vorhanden Bindings im Projekt können im unteren Fensterbereich angezeigt werden:
Play All Reset Rewind Stop	
Bind	VAMETERS + NTS +
Cancel	Folgende Bindings sollten nun vorhanden sein:



m,schubert

DATA DATA > APPLICATION PARAMETERS + ~ APPLICATION EVENTS + (-) ~ doubletap ]25 Voice Alias

15

play		
Voice Response		
playing now		
Voice Help		

Dokumentation Hololens Seminar 2018&2019 Mödling

Mit den Einstellungen bei Voice kann man noch einen Sprachbefehl bzw. eine Sprachausgabe festlegen.

Also z.B. Sprachbefehl bei doubletap "play" und bei swiperight "reset".

Das Projekt wie gewohnt sichern bzw. publizieren.

# Export und Import

←) > swipeback JS

Für die spätere Verwendung kann das gesamte Projekt aus der Übersichtsseite heraus exportiert werden:

completing the different	Studio Sample proje	cts below!
Q		
Exposition	a	) ھ
	OPEN PROJECT	

Das so erstellte Zip-File kann auch wieder importiert werden:



# Projekt "Richtvorrichtung"

## Vorbereitungen

Das vorbereitet pvz-File wieder wie im 1. Beispiel einfügen und die Animation wieder wie vorher einstellen.

Das Ergebnis sollte so aussehen:



## Wichtig!

Immer zwischendurch unter Preview die Funktion kontrollieren. Wenn es hier nicht funktioniert, dann geht es auch sicher nicht mit der Hololens!

Für die weiteren Schritte gibt es mehrere Ansätze:

- 1.) Anlegen von Application Parameters
- 2.) Direktes Ansprechen der Objekte in JS

#### Objekteigenschaften ändern

Damit ist es möglich die Eigenschaften einzelne Elemente während der Simulation gezielt zu verändern. Dazu gehören z.B. Verschiebung, Transparenz, Rotation,...

Durch den Einsatz von JavaScript (JS) ist es erreichbar beliebige Abhängigkeiten zu schaffen.

Über sogenannte Model-Items kann man einzelne Bauteile "ansprechen"

Dazu wird ein Modelltem auf das gewünschte Bauteil gezogen:



Sofort mit einem Namen versehen:

→) X Rotation
0
$\rightarrow$ ) Y Rotation
0
→) Z Rotation
0
→) ✓ Visible
→) □ Occluding
→)
→) Opacity (1 Opaque - 0 Transparent)
1
→) Color
×
studio ID
mdlit_deckel 🔗
Friendly Name
mdlit_deckel

Auch das Gehäuse einem Modelltem zuweisen:

✓ VIEWS +		
✓ → Home	$\rightarrow)$	
Home.js		
✓		
✓ ≜ mdl_biegevorrichtung		
i mdlit_deckel		
indlit_gehaeuse		

n.schubert

Weiters werden Application Paramter erzeugt. Sie ermöglichen den gezielten Zugriff auf die Eigenschaften der vorher angelegten Modelltems.

DATA	
$\checkmark$ Application parameters $+$	
\$) Thing	
与) Thing Template	
与) ThingMark	
🖘 trans_mdlit_deckel 🔗	×
🖘 trans_mdlit_gehaeuse 🔗	×

Die Application Parameter nun auf das gewünschte Moelltem ziehen (kann im 3D-Bereich **oder** in der Auflistung geschehen:

Solution My Projects	s > 02_Richtvor	richtung 🔵						
Save	Preview	Publish	Share 🗸					
PROJE	ст	WIDGETS		CANVAS		- A COOTONIANC	DATA	
✓ CONFIGURATION		✓ TARGETS	-> D @ ->			0	✓ APPLICATION PARAMETERS +	
Themes		Image Target			→	Y Coordinate	S) Thing	
Experiences		90				0	(S) Thing Template	
Info		Model Target				7 Coordinate	S) ThingMark	
My ThingMarks		<ul> <li>Spatial Target</li> </ul>				0	📕 🖘) trans_mdlit_deckel ⊘	×
✓ VIEWS +							≲) trans_mdlit_gehaeuse ⊘	×
✓ 🕒 Home	$\rightarrow$ )	1 3			-	X Rotation	✓ APPLICATION EVENTS +	
Home.js		✓ AUGMENTATIONS					←) > doubletap JS	
✓ 💮 3D Contain	ner	℃7 3D Gauge			->	Y Rotation	←) > hold JS	
✓ 魚 mdl_bie	gevorrichtung	2D1				0	←) > swipeback JS	
🚊 mdlit_de	eck 🕘 🔶	3D Label			→	Z Rotation	←) > swipeforward JS	
☆ mdlit_ge	ehaeuse				>	0	←) > swipeleft _JS	
× STVLES		A Model				Vicible	←) > swiperight _JS	
Application		Model Item				Visible	←) > swipeup JS	
✓ RESOURCES +		✓ OTHER	$\langle \rangle$		$\rightarrow$	0 Cccluding	✓ EXTERNAL DATA +	
V Uploaded +		(THL) TML Text			7			

Danach wird das entsprechende Binding (also auf welche Eigenschaft sich der Parameter beziehen soll) angegeben:

Select Binding Tar	get
trans_mdlit_deckel mdlit_	get deckel
Property	
Scale	-
X Coordinate	
Y Coordinate	
Z Coordinate	
X Rotation	
Y Rotation	
Z Rotation	
Visible	
Force Hide	
Occluding	
Always on top	
Opacity (1 Opaque - 0 Transp	parent)
Item Translucence	
Bind	
Cancel	

Jetzt muss nur noch der Code geschrieben werden, der den Zusammenhang zwischen dem Anklicken des Teils und dem dadurch zuzuweisenden Wert herstellt:

Das entsprechende Event findet man im Modelltem "mdlit\_deckel":

Dort fügt man einen Funktionsaufruf ein:

	mdlit_deckel
	✓ EVENTS
	←) Click JS
$\checkmark$	klick <u>mdlit.deckel</u> 0

Man könnte den Code auch direkt hier hineinschreiben, der Vorteil einer Funktion liegt aber darin, dass der Code aller Funktionen zentral in einem Fenster bearbeitet werden kann.

Der Code wird unter Home→ Home.js eingefügt:

PROJECT	
<ul> <li>CONFIGURATION</li> </ul>	
Themes Experiences Info My ThingMarks VUEWS + Homese Homese	<pre>// Scope.slick_mdlit Scope.klick_mdlit Scope.apo.par } </pre>
<pre>\$scope.klick_mdlit</pre>	t_deckel = function() // Funktionsaufruf DECKEL
{	
<pre>\$scope.app.pa</pre>	<pre>rams.trans_mdlit_deckel = 0.3; // Transparenz auf 0.3 setzen</pre>
};	

Wenn man will, dass bei jedem Klick die Transparenz wechselt, kann man folgenden Code verwenden:

```
$scope.klick_mdlit_deckel = function() // Funktionsaufruf DECKEL
{
    if ($scope.app.params.trans_mdlit_deckel < 1)
    {
        $scope.app.params.trans_mdlit_deckel = 1; // Transparenz auf 1 setzen
    }
    else
    {
        $scope.app.params.trans_mdlit_deckel = 0.3; // Transparenz auf 0.3 setzen
    };
};</pre>
```

Dieselben Schritte sind sinngemäß beim Gehäuse durchzuführen.

Weiters kann man beim swipright noch einen JS-Code hinzufügen, der die Transparenz unabhängig vom Ist-Wert auf zurücksetzt (Das Binding mit dem reset von mdl\_biegevorrichtung wir dadurch nicht aufgehoben).

VIEWS +	ThingMark			1.0	토) trans_mdlit_gehaeuse 🖉 🛛 🗙
✓ Home →)	✓ AUGMENTATIONS			a) V Caerdinate	
Home.js 🗸 💮 3D Container	n 3D Gauge			0	<ul> <li>←) &gt; doubletap 15</li> <li>←) &gt; hold 15</li> </ul>
✓ 魚 mdl_biegevorrichtung	3D Image			→) Y Coordinate	←) > swipeback JS
indlit_deckel	3D Label			1	←) > swipeforward JS
္လံ mdlit_gehaeuse	📩 Model			→) Z Coordinate	<ul> <li>↔ &gt; swipeleft Js</li> <li>↔ &gt; swiperight Js</li> </ul>
Application	📩 Model Item			→) X Rotation	viewStrLawiperight_apl0:
RESOURCES +	✓ OTHER	z ×		90	
✓ Uploaded +	CIMES THE Text				
Richtvorrichtung_Sequenz.pvz			INGS	Enter a term to filter bindings	←) > swipeup JS
> Default	Show bindings for the selecte	ed widget only			✓ EXTERNAL DATA +
	Source		Handler		
	Label: Application Event Type: event Name: app Event: doubletap		Service: playAll Name: mdl_biegevorrichtung Type: ui	Û	
	Source		Handler		
	Label: Application Event Type: event Name: app Event: swiperight		Service: reset Name: mdl_biegevorrichtung Type: ui	Ć	

Bei diesem Funktionsaufruf muss noch viewCtrl vorangesetzt werden da es aus dem app-Objekt aufgerufen wird

#### viewCtrl.swiperight\_apl();

In home.js muss man folgendes ergänzen:

```
$scope.swiperight_apl = function() // Funktionsaufruf SWIPERIGHT
{
    $scope.app.params.trans_mdlit_deckel = 1; // Transparenz auf 1 setzen
    $scope.app.params.trans_mdlit_gehaeuse = 1; // Transparenz auf 1 setzen
};
```

Alternative Methode - direktes ansprechen von Objekten:

Man kann auf das erzeugen Application Parametern verzichten und stattdessen folgende Syntax im JS-Code verwenden:



Dieser Code erreicht die selbe Funktionalität wie auf Seite17, ohne dass man dafür einen Application Parameter anlegen muss.

Ein mischen beider Varianten innerhalb einer Funktion funktioniert nur bedingt, da der Application Parameter bei einer Änderung der Eingenschaft nicht aktualisiert wird.

...schubert

#### Zusatzinformationen einfügen

Im Folgenden wird ein Infoblatt eingefügt, auf dem z.B. die Bedienung und die Funktionen der AR beschrieben sind.

Ein 3D Image wird in den Modellbereich gezogen, dabei ist es praktisch die Funktion "mate to surface" zu aktivieren, sie ermöglicht das gezielte platzieren auf einer Oberfläche des Modells.

Zuerst fügen wir das Symbol für die Schaltfläche ein:



Nun noch über Resource+ die gewünschte Datei hinzufügen:



Dann das eigentliche Infoblatt: (Hier die Funktion "mate to surface" nicht einschalten)

Folgende Parameter einstellen:



...schubert

Im JS-Code müssen noch folgende Funktionen eingebaut werden:

- Ausblenden Info Button wenn Animation gestartet
- Transparenz setzen wenn Deckel Transparent

Der gesamte Code sieht nun so aus: (Deckel wird über Applicationparameter gesteuert, der Rest direkt)

```
$scope.klick_mdlit_deckel = function()
                                                // Funktionsaufruf KLICK DECKEL
{
  if ($scope.app.params.trans_mdlit_deckel < 1)</pre>
    {
      $scope.app.params.trans_mdlit_deckel = 1;
                                                      // Transparenz auf 1 setzen
      $scope.view.wdg['img_ibutton'].opacity = 1;
    }
  else
    {
      $scope.app.params.trans_mdlit_deckel = 0.3;
                                                        // Transparenz auf 0.3 setzen
      $scope.view.wdg['img_ibutton'].opacity = 0.3;
    }
};
                                                // Funktionsaufruf KLICK GEHAEUSE
$scope.klick_mdlit_gehaeuse = function()
{
    if ($scope.view.wdg['mdlit_gehaeuse'].opacity < 1)</pre>
    {
      $scope.view.wdg['mdlit_gehaeuse'].opacity = 1;
    }
    else
    {
      $scope.view.wdg['mdlit_gehaeuse'].opacity = 0.3;
    }
};
$scope.swiperight_apl = function()
                                              // Funktionsaufruf SWIPERIGHT = RESET
{
                                                     // Transparenz auf 1 setzen
      $scope.app.params.trans_mdlit_deckel = 1;
                                                     // Transparenz auf 1 setzen
      $scope.app.params.trans_mdlit_gehaeuse = 1;
      $scope.view.wdg['img_ibutton'].visible = true; // Infobutton anzeigen
};
$scope.klick_img_ibutton = function()
{
  $scope.view.wdg['img_info'].visible =! $scope.view.wdg['img_info'].visible; // Toggle Infotext
};
$scope.doubletap_apl =function ()
                                           // Aufruf wenn Animation gestartet wird
{
  $scope.view.wdg['img_ibutton'].visible = false; // Infobutton ausblenden
};
```

## Sprachsteuerung

Für die Sprachsteuerung werden noch weitere Application Events angelegt und mit entsprechendem Code in Home.js erzeugen

	←) ✔ hideinfo JS	×
Event	viewCtrl.hideinfo_apl();	
hideinfo		
Allow voice commands to trigger the event	Voice Alias	
	hide info	
Add	Voice Response	
Cancel	hide info	

# Arbeiten mit Illustrate

# Installation

Nach der Installation kann es erforderlich sein, die Umgebungsvariable <u>ptc\_d\_license\_file</u> zu erstellen und als Wert den Pfad zum Lizenzfile anzugeben:

Systemeigenschaft	ten	×	Umgebungsvariablen					
Computername H	ardware Enweitert Computerschutz Remote		Benutzervariablen für Martin					
Sie müssen als A durchführen zu k	dministrator angemeldet sein, um diese Änderunge önnen	n	Variable	Wert				
Leistung			HOME	%HOMEDRIVE%%HOMEPA	ATH%			
Viguelle Effekte	Prozessorzeitolapung, Speichemutzung und vitu	allar	Deth	C:\Users\Martin\OneDrive	in (OneDrive			
Speicher	, riozessorzeipianang, speicheinatzang and vita	andi	TEMP	%USERDROEILE%\AppData\	Local/Microsoft/WindowsAp	ocal\Temp		
	Einstellun	gen	TMP	%USERPROFILE%\AppData	\Local\Temp			
Benutzerprofile Desktopeinstellu	ungen bezüglich der Anmeldung				Neu Bearbeiten	Löschen		
	Einstellun	jen	Systemvariablen					
			Variable	Wert		^		
- Starten und Wie	derherstellen		asl.log	Destination=file				
Suntamatart Sur	tamfablar und Dabugiafarmationan		ComSpec	C:\WINDOWS\system32\cmd.exe				
Systemstart, Sys	stemener und Debuginionnationen		Creo4_Install_Path C:\Program Files\PTC\Creo 4.0\M060\Parametric\bin					
	D		DISPLAY	:0.0				
	Einstellun	jen	MAN_CHM_INDEX	C:/PROGRA~1/PTC/MKSTC	00~1/etc/chm/tkutil.idx;C:/P	ROGRA~1/		
			MAN_HIM_PATHS	C:/PROGRA~1/PTC/MKSTC	JO~1/etc/htm/perl;C:/PROG	(A~1/PTC/		
	Umgebungsvari	ablen			Neu Bearbeiten	Löschen		
	OK Abbrechen (	bernehmen			OK	Abbrechen		
eue Systemvariab	le			×				
Name der								
ariabien:	ptc_d_license_file							
Vert der 'ariablen:	c:\ptc\Im21354879456.dat							
Verzeichnis durchs	suchen Datei durchsuchen		OK Abb	rechen				

Wenn man einen Lizenzserver verwendet ist dieser Schritt nicht notwendig.

Beim Starten, noch als Edition "Creo Illustrate Standard" wählen!

	Creo Illustrate Start	×				
No such	No such feature exists.					
Lizen	nzoptionen beim Start wählen					
Creo Illu	strate Lizenzoptionen					
Edition:	Creo Illustrate Professional 💌					
	Creo Illustrate Essentials					
Schemat	Creo Illustrate Standard					
	Creo Illustrate Professional					
Lizenzin	formationen					
Lizenzserver: 7788@localhost						
Lizenzser	ver: 7788@localhost					
Lizenzser Vorkonfi	igurierte Lizenzserver					
Lizenzser Vorkonfi PTC_D_L	igurierte Lizenzserver CENSE_FILE: 7788@localhost					
Lizenzser Vorkonfi PTC_D_LI LM_LICE	ver: 7788@localhost igurierte Lizenzserver ICENSE, FILE: 7788@localhost NSE_FILE:					

# Vorbereitung der Daten in Creo

Da zurzeit die Verwendung von Unterbaugruppen in Vuforia Studio Probleme verursacht, sollten Unterbaugruppen als Einzelteil eingebaut werden.

# Ablauf:

1.schubert

Unterbaugruppen als IGES speichern und als Teil wieder in die Hauptbaugruppe einbauen. Die originale Unterbaugruppe löschen.

# Beispiel Getriebemontage

In Illustrate über die Schaltfläche Importieren die vorbereitete <u>Getriebemontage Demontage.pvz</u> einbetten. Wenn man Verbinden wählt, dann werden Änderungen im Creomodell übernommen. Für die Aufbereitung mit Thingworx Studio ist allerdings das Einbetten sinnvoller.

Nach dem Öffnen den Oberkasten deaktivieren:





Im Menüband Animation die Schaltfläche Sequenz auswählen:

Danach Im Menüband Startseite Abbildungsansicht – Aktuelle Ansicht verwenden





Dann unter Animation den ersten Schritt erzeugen:

Zwischenwelle anklicken und einblenden, da wir später den Effekt umkehren. D.h. aus dem Einblenden wird dann ein Ausblenden!



Nun die Welle in die Ablageposition bewegen – hier muss man sich vorher die genauen Koordinatenverschiebungen überlegen:

Die Zeit für die einzelnen Bewegungen können durch das Verschieben der Pfeile in der Zeitleiste angepasst werden. Dies ist vor allem bei sehr langen oder kurzen Wegen sinnvoll.

entliste	A Notizen	🔒 Ressourcen	a Lokalisi
		· · · ·	

Die Schaltfläche Transformieren anwählen und folgende Verschiebungen nacheinander durchführen:

X: +70; Y: -105; Z: -98; X: -83

Dann die Zeit eventuell wieder etwas länger stellen und den Effekt "pulsieren" auswählen.

Zuletzt den gesamten Schritt mittels RMT auf die Zeitleiste umkehren:

Charles C	<ul> <li>Pfad in 3D anzeigen</li> <li>Schlüsselbild einfügen</li> </ul>	×
	Umkehren Spiegeln	
► [ (0:00.00/ 0:04.95) Schritt 1	Teilen Zusammenführen	ählen
Schlüsselbild-Editor	Löschen Kopieren Einfügen	Entf Strg+C Strg+V

Dann einen neuen Schritt hinzufügen und den Vorgang sinngemäß für das nächste Bauteil erzeugen.

Auf diese Weise erhält man für jeden Arbeitsschritt eine eigene Sequenz, die dann in Vuforia Studio abgespielt werden kann.

Servicestückli:	<caption> Abbildungen</caption>	Contemporation Schritt-Editor
<b>S</b>	•	
Montageablauf		
5°	ĥ	
Zwischenwelle		
<b>S</b>	e.	
Eingangswelle		

Das fertige Projekt soll dann immer nur die aktuellen Arbeitsschritte in die Realität eiblenden, daher ist es sinnvoll, wenn man zu Beginn keines der Teile sieht. Daher die Abbildung (Montageablauf) im Schritteditor anklicken, dann auf Servicestückliste alle Teile abwählen und Abbildungsansicht – Aktuelle Ansicht verwenden:



Abschließend als PVZ exportieren:

In PVZ-Datei publizieren				×	
🔹 💌 👻 🚞 « Drive )	• • • • Suchen				
🗇 Organisieren 🗸 📗	🗇 Organisieren 🗸 🏢 Ansichten 🗸 🏹 Werkzeuge 🗸				
Gemeinsame Ordner	🛅 05_Getriebe				
📃 Desktop					
🖪 Eigene Dokumente					
📃 desktop-62d3vh8					
🔺 Arbeitsverzeichnis	Dateiname: Änderungsdatum:				
×	Dateiname: Montage_Getriebe		Typ PVZ (*.pvz)	-	
▶ Ordnerstruktur		Speichern	- Abbrechen		

...schubert

## Verwendung in Studio

n.schubert

Wie schon vorher wird in dem neuen Projekt ein Model erzeugt und die soeben generierte PVZ-Datei importiert.

CANVAS	Montage_Getriebe.pvz
	Scale .
	1.0
⇒)	) X Coordinate
	0
⇒)	) Y Coordinate
	-0,006
→)	) Z Coordinate
	0,204
→)	) X Rotation
	0
	) Y Rotation
	0
→)	) Z Rotation
	0
→)	) 🔽 Visible
→)	)  Occluding
)	) Always on top
→)	) Opacity (1 Opaque - 0 Transparent)
	1
→	Sequence
	Montageablauf - erzeugte Sequenz auswählen.

Damit die Sequenz gestartet wird, muss noch ein Binding hergestellt werden. Zum Beispiel mit einem Doppelklick:



Durch Auswahl von Play wird immer nur ein Schritt abgespielt.

# Animationen aus Creo Mechanismus importieren

Die Mechanismus-Analyse muss mit der Wiedergabesteuerung in CREO als fra-Datei exportiert werden:



Die Datei wird im Arbeitsverzeichnis gespeichert.

In Creo Illustrate wieder die folgenden Schritte durchführen:

Daten eingebettet importieren



Abbildung der Ausgangssituation erstellen:



Dann Animation – Sequenz auswählen:



Nun in der Registrierkarte Animation auf "Animation importieren" klicken und die soeben erzeugte fra-Datei laden:

Datei	Sta	artseite	Abbildung	Tools
		+	A Notizenseite	5
Sequenz Anim	ation	Schritt hinzufüge	Animation imp	ortieren 🗙
Neu				Aufgaben

Der weiter Vorgang ist gleich wie vorher: - Publizieren und in Studio als Resource laden.

# Zusatzinfo "Scharniere"

Standardmäßig wird das Koordinatensystem des Bauteils als Drehzentrum verwendet. Um eine gezielte Drehbewegung um eine bestimmte Achse auszuführen ist es möglich ein anders KSYS auszuwählen:



Hier kann z.B. die Kante einer Bohrung ausgewählt werden:



So ist es möglich eine Drehung um diese Achse zu erzeugen.

...schubert

# Beispiele

# Modellflieger (Dreidecker - Fokker DR1)

Die vorliegenden CAD-Daten des Modellfliegers basieren auf einem Holzmodell von Conrad.



Folgende Messwerte werden erfasst und an den TWX-Server übermittelt:

- Drehzahl (IR-Reflektionslichtschranke Interrupt im ESP32)
- Temperatur und Feuchtigkeit (DHT22)
- Spannung der Solarzelle (AD-Wandler im ESP32)

Zum Testen werden die entsprechenden Datenpunkte im TWX-Composer angelegt, aber nicht mit realen Werten versorgt.

Die erste Aufgabe besteht darin, den Propeller des Zwillings drehen zu lassen, wenn die Drehzahl > 0 ist.

# Vorbereitung in TWX-Composer

# Projekt und Thing anlegen:

×	thingworx	Q SEARCH + NEW
Set Pr	oject Context	+ Project: 19U01_99_SUM_Dreidecker_Pr ⑦ To Do ▼ O Save O Cancel More ▼
(L)	Recent         I₂           ∨ 19U01_99_SUM_Dreid	© ① General Information ● Entities <> Services ▲ Permissions ④ Change History
<b>₽</b>	<ul> <li>19U01_99_SUM_Dreide</li> <li>Pro_Flugzeug</li> <li>Flugzeug</li> <li>Flugzeug</li> <li>Flugzeug</li> <li>MQTT-Test</li> <li>MQTT-Test</li> <li>MQTT_loBroker_SUM</li> <li>Unassigned</li> <li>ApplicationKey</li> <li>es-public-access</li> <li>Entfeuchter_MQTT</li> <li>SUM_Test_MQTT</li> <li>TEST</li> <li>AnyDataChangeEvent</li> </ul>	Central mormation
Set Pro	thingworx           yect Context         +           Recent         I2           > 19U01_99_SUM_Dreide           19U01_99_SUM_Dreide           > 19U01_99_SUM_Dreide	Q SEARCH + NEW         Thing: 19U01_99_SUM_Dreidecker_Th ⑦         To Do ▼       O Save         O General Information         Eproperties and Alerts       ↔ Services         f Events       Subscriptions         A Permissions       O Change History         A View Relationships
<b>1</b> 9 	Ho_Hogeug     Filogeug     Filogeug     Filogeug     Filogeug     MQTT-Test     NQTT_IoBroker_SUM	No mage vame () available 19001_99_SUM_Dreidecker_Th Change Description ()
	<ul> <li>Unassigned</li> <li>ApplicationKey</li> <li>es-public-access</li> <li>Entfeuchter_MQTT</li> <li>SUM_Tes_MQTT</li> <li>TEST</li> <li>AnyDataChangeEvent</li> </ul>	Project ⑦
		<ul> <li>✓ Active ⑦</li> <li>□ Published ⑦</li> </ul>

#### Property Drehzahl hinzufügen:



E New Property 4
Name (required)
Drehzahl
Description
Pose type (?)
# NUMBER V
Units @
rpm
Min Value 🕐
Max Value (?)
Has Default Value ⑦
Persistent ⑦
Read Only 🕐
Logged ⑦
Binding 🕐
None 🔻
Die angegeben Einheit hat keinen

Die angegeben Einheit hat keinen Einfluss auf die Funktion.

Das Thing danach speichern!

Um Daten an TWX senden zu können müsste auch noch ein Application Key erzeugt werden.

## Vuforia Studio

Anlegen eines neuen Projekts (3D-Eyewear)

Model in den Arbeitsbereich ziehen und Resource laden:





Koordinaten ggf. auf 0 setzen und Namen vergeben:

mdl_Flugzeug 💋
Friendly Name
mdl_Flugzeug
✓ SERVICES

34

n.schubert

Als Target ein Spatial Target oder Thingmark wählen und in den Arbeitsbereich ziehen.

Nun muss die Luftschraube einem Modelitem zugewiesen werden:



m.schubert

# Benennung ändern:

	$\rightarrow$ ) $\square$ Always on top
	→) Opacity (1 Opaque - 0 Transparent)
TRANS I	→) Color
	Studio ID mdlit_Luftschraube ⊘
	Friendly Name mdlit_Luftschraube
	✓ EVENTS
$\mathbf{X}$	←) Click JS

Die Achsen sind:



Für die Drehung der Luftschraube ist daher die Z-Achse passend.

Nun wird ein Application Parameter erzeugt und Z Rotation zugewiesen:



Dann den Parameter direkt auf Z Rotation ziehen, **oder** auf das Modelitem (Auswahldialog frägt dann gewünschte Property ab)

PROJECT	wibders	(4)		UAIA	
CONFIGURATION	✓ TARGETS		/0/0/22	✓ APPLICATION PARAMETERS +	
Themes	Möglichkeiten		→) Scale	5) rz_mdlit_Luftschraube ⊘	×
Experiences	. Mogierikeitett		1000	15) Thing	
Info	d woder rarget	and the second sec		気) Thing Template	
My ThingMarks	<ul> <li>Spatial Target</li> </ul>		->) X Coordinate	\$) ThingMark	
V VIEWS T	ThingMark			✓ APPLICATION EVENTS +	
✓ 🔄 Home			→) Y Coordinate	←) > doubletap 15	
Home.js			0,01	←) > hold IS	
V 💮 3D Container	SD Button		→) Z Coordinate	(-) > swipeback	
✓	~ AUGMENTAR		0,006	(-) > swipeforward 25	
🚊 mdlit_Luftschraube	T <sup>o</sup> 3D Gauge		(a) X Rotation	←) > swipeleft 🖂	
spatialTarget-1	3D Image	L. K.		(-) > swiperight 15	
✓ STYLES				←) > swipeup 15	
Application	3D Label		->) Y Rot on	V EXTERNAL DATA +	
✓ RESOURCES +	A Model				
> Default	A Model Item		→) Z Rotation		
✓ Uploaded +	N OTHER	~	0		
Dreidecker.pvz	z		->) 🗹 Visible		
	Cost.) TML Text				
	12			-	
			Enter a term to filter bindings		
	Show bindings for the selected widget only				

36

#### Folgendes Binding sollte angezeigt werden:

ЕЗ		Enter a term to filter bindings		
Show bindings for the selected widget only				
Source	Target			
Binding Expression: app.params['rz_mdlit_Luftschraube']	Widget: mdlit_Luftschraube Property: rz		+ Add Filter	Ū

#### Einbinden der externen Daten:

Add External Data				
Entities	PROPERTIES SERVICES	EVENTS		
Q	Q getprop			
19U01_99_SUM_Dreidecker_Th (Thing)	GetPropertyDefinition	+		
AlertFunctions (Resource)	GetPropertyDefinitions	+		
AlertHistoryStream (Thing)	GetPropertyLogging	+		
AlertProcessingSubsystem (Subsystem)	GetPropertyQuality	+		
AlwaysOnReporting (Thing)	GetPropertyTime	+		
AnalyticsGateway (ThingTemplate)	GetPropertyValues	+		
AnalyticsJobServer (ThingTemplate)	GetPropertyValuesAsMultiRowTable	+		
AnalyticsResultStorage (Thing)	GetPropertyValuesVTQ	+		
<u>c</u>	lose			

Den Service "GetPropertyValue" hinzufügen.

Die Drehzahl sollte nun verfügbar sein. Wichtig ist das setzen der angezeichneten Optionen. Auto-refresh Rate ist das Aktualisierungsintervall in Sekunden.

#### ✓ Services +

- $\rightarrow$ ) V GetPropertyValues
  - $\leftarrow) \text{ All Items}$
- ←) All Selected Items ✓ Current Selected Item
- ←) description
- ←) Drehzahl
- ←) name
- ←) tags
- ←) thingTemplate > Parameters
- Configuration
- 🗸 Invoke On Startup
- Invoke On Entity Change
  Auto-select first row
- 🗸 Auto-refresh
- Auto-refresh Rate

5

Für den Drehzahlwert wird auch noch ein Application Parameter erzeugt und ein Binding mit den externen Daten hergestellt:

		DATA	
	- Application parameters $+$		
	与) Drehzahl 💋		×
1	) rz_mdlit_Luftschraube 💋		×
	5) Thing		
	3) Thing Template		
	) ThingMark		
	APPLICATION EVENTS +		
	🗧 🔪 doubletap 🛛 S		
	← > hold JS		
	← > swipeback JS		
	← > swipeforward JS		
	←) > swipeleft JS		
	←) ⇒ swiperight JS		
	←) → swipeup JS		
	✓ E TERNAL DATA +		
	✓ I9U01_99_SUM_Dreidecker_Th 🔿	_	
	⇒) Dynamic Entity Name		
	Properties +		
	Services +		
	$\rightarrow$ ) V GetPropertyValues —		
	←) All Items		
	←) All Selected Items		
	✓ Current Selected Item		
	$\leftarrow$ ) description		
	←) Drehzahl		

Zur Anzeige der Drehzahl kann diese auch mit einem 3DGauge verbunden werden.

г.	- 1- 1		-1: -	f = 1 = . = . = . = .	
ES	gipt	aann	ale	tolgende	n binaings:

Source	Target		
Binding Expression: app.params['rz_mdlit_Luftschraube']	Widget: mdlit_Luftschraube Property: rz	+ Add Filter	Ů
Source	Target		
Binding Expression: app.params['Drehzahl']	Widget: 3DGauge-1 Property: text	+ Add Filter	Û
Source	Target		
Binding Expression: app.mdl['19U01_99_SUM_Dreidecker_Th'].svc['GetPropertyValues'].data.current['Drehzahl']	Property: app.params['Drehzahl']	+ Add Filter	Û



# JavaScript für Luftschraube

Zum Steuern des Winkels der Luftschraube in Abhängigkeit des Drehzahlwerts ist es erforderlich auf Änderung des Drehzahlwertes zu reagieren. Wir öffnen daher den integrierten Editor in dem wir auf Home.js clicken.

Mit folgendem Code wird auf eine Änderung reagiert:

```
$scope.$watch("app.params['Drehzahl']",function()
{
 **** MEIN CODE ***
})
```

Als ersten Test versuchen wir den Winkel der Luftschraube bei jeder Werteänderung um 10 zu erhöhen.

Der Code zur Addition von 10 lautet:

\$scope.app.params.rz\_mdlit\_Luftschraube = \$scope.app.params.rz\_mdlit\_Luftschraube +10;

Oder in kurzer Schreibweise:

\$scope.app.params.rz\_mdlit\_Luftschraube += 10;

# ACHTUNG!

...schubert

Verwendet man diesen Code ohne weiter Maßnahmen funktioniert es nicht, da die Application Parameter noch nicht initialisiert sind und keinen definierten Wert haben.

Man muss daher prüfen ob die Variable bereits "gültig" ist und einen Startwert zuweisen:

```
if (!$scope.app.params.rz_mdlit_Luftschraube){$scope.app.params.
rz_mdlit_Luftschraube = 0}
```

Der fertige Code hat dann folgende Form:

CODE  $\rightarrow$  Schritt1.txt

Mit Preview nun die Funktionalität testen – bei Wertänderungen der Drehzahl im Composer sollte sich spätestens 5 Sekunden später die Luftschraube um 10 Grad weiter drehen.

#### Loggen von Werten

n.schubert

Bei der Entwicklung von neuem Code ist es oft hilfreich Werte oder Hinweise aus dem Programcode anzeigen zu lassen.

Der Eintrag wird im Code mit der folgenden Zeile ausgelöst:

console.log("HALLO");

Unter Chrome müssen die Entwicklertools aktiviert werden [Strg]+[Shift]+[i]

Dann Registrierkarte "Console" umschalten:



Für unser Beispiel "Flugzeug" sieht das dann wie folgt aus:



Damit sich die Luftschraube kontinuierlich dreht, muss der Drehwinkel kontinuierlich erhöht werden.

Hierfür verwenden wir einen Timer:

Timer verwenden Grundaufbau:

...schubert

setInterval	
clearInterval	

Wenn der Timer gestartet wird, bekommt man als Rückgabewert einen sogenannten "timer identifier". Damit ist es später möglich diesen Timer auch wieder zu löschen.



weitere Infos:

https://javascript.info/settimeout-setinterval

#### Funktionen verwenden

Um die Übersichtlichkeit zu erhöhen kann man eigene Funktionen definieren und im Programmablauf aufrufen.

Wir definieren die Funktion rotate:



Diese Funktion erzeugt einen Eintrag im Log und zeigt den Übergabewert an.

Der Funktionsaufruf erfolgt mit:

rotate(true);

bzw.:

rotate(false);

## Bedingungen verwenden

Um den Programmablauf steuern zu können braucht man noch die if und else Anweisungen:

```
if ($scope.app.params.Drehzahl > 0){
  rotate(true);
}
else {
  rotate (false);
}
```

Vergleichsoperatoren:

==	ISTGLEIC
!=	UNGLEICH
>	GRÖßER
>=	GRÖßERGLEICH
<	KLEINER
<=	KLEINERGLEICH

#### Weitere Infos:

https://wiki.selfhtml.org/wiki/JavaScript/Operatoren/Vergleichsoperatoren

Hat man mehrere Bedingungen zu verknüpfen, so können die <u>logischen Operatoren</u> dazu verwendet werden:

&&	UND
	ODER
!	NICHT

#### Weitere Infos:

https://wiki.selfhtml.org/wiki/JavaScript/Operatoren/Logische Operatoren

43

#### Gesamter Code zum Drehen des Propellers

```
var timerId = -1;
                     //timer identifier
var timingIntervall = 300; //miliseconds for timer
 if (!$scope.app.params.rz_mdlit_Luftschraube){$scope.app.params.rz_mdlit_Luftschraube = 0} // set
to startvalue
$scope.$watch("app.params['Drehzahl']",function() // call function when value changes
{
 if ($scope.app.params.Drehzahl > 0){ //check propeller rotation
    rotate(true);
  }
  else {
    rotate (false);
  }
})
function rotate (active){
console.log('function call -rotate- '+ active);
if (active && timerId == -1){
timerId = setInterval(function() {
   $scope.$apply(function(){
                                         // code called every timingIntervall
       $scope.app.params.rz_mdlit_Luftschraube += 10;
 })
}, timingIntervall)
}
if(!active){
 clearInterval(timerId);
   timerId=-1;
}
```

CODE → Schritt2.txt

#### schubert

#### Drehzahl und Statusanzeige

Wir fügen noch eine Anzeige für den Drehzahlwert und einen Infotext über den Zustand des Motors ein.

3D Gauge in den Arbeitsbereich ziehen und mit dem Applicationparameter "Drehzahl" ein Binding herstellen.



Die Anzeige kann über den Filter im Binding angepasst werden:

	Edit Data Filte	r
Filter Name		
Drehzahl		
Filter Body		
return(value + '	'rpm');	
return(value + '	'rpm');	
return(value + '	rpm');	
return(value + '	'rpm');	
return(value + '	ppm'); Done	

Wenn man Messwerte mit Nachkommastellen hat und diese runden möchte, dann kann man den Code erweitern: (funktioniert nur wenn Datentyp keine integer Wert ist)

Return (value.toFixed(2) + '°C');

Zum Testen Property "Temperatur" anlegen und mit neuem Gauge binden.

Für den Infotext ein 3DLabel anlegen und als ID "lab\_Info" angeben:



## Im Code folgende Zeilen einfügen:

	<pre>var timerId = -1;</pre>	paraı
) v   v	<pre>{     tropp:paramote cheaning of chea</pre>	<del>(</del>
) ) 	<pre>rotate (false); \$scope.view.wdg['lab_Info'].text = 'Motor aus'; }</pre>	÷
	<pre>function rotate (active){   console.log('function call -rotate- '+ active);   if (active &amp;&amp; timerId == -1){</pre>	

Eine Schaltfläche "Take off" soll nur sichtbar sein, wenn die Drehzahl > 0 ist.

3D Button einfügen und Namen vergeben. Dann noch ein Clickevent angeben:

Studio ID
Friendly Name but_Takeoff
✓ EVENTS
←) Click JS
click_but_Takeoff();

Im Code die Sichtbarkeit steuern:



Mit der Schaltfläche soll nun ein Bewegungsvorgang für das ganze Flugzeug gestartet werden. Dazu führen wir einen weiteren Timer ein. Zur Unterscheidung benötigen wir eine neue timerID. Daher wird die vorhandene umbenannt:



Für die Bewegung wird eine neue Funktion erstellt, die im Wesentlichen gleich jener ist, die die Rotation der Luftschraube ausführt.

...schubert

# Steigflug und Looping ausführen

Die Applicationparameter und die Bindings:



Mit dem Objekt Math können Berechnungen durchgeführt werden und Konstanten eingefügt werden:

Für den weiteren Ablauf benötigen wir

- Math.Pl
- Math.sin(x)
- Math.cos(x)

Weiter Infos:

https://wiki.selfhtml.org/wiki/JavaScript/Objekte/Math

Das Flugzeug soll nun eine bestimmte Höhe im Steigflug erreichen und dann Loopings fliegen.

Wir legen zunächst folgende Variablen an:

```
var timerId_mov = -1; //timer identifier
var timingIntervall_mov = 30; //miliseconds for timer (climb & looping)
var climb_mz = 0.01; // climb parameter
var climb_my = 0.005;
```

und erstellen eine Funktion, die die Position des Flugzeugs ständig verändert:

```
function move (active){
console.log('function call -move- '+ active);
if (active && timerId_mov == -1){
timerId_mov = setInterval(function() {
   $scope.$apply(function(){
        $scope.app.params.mz_mdl_Flugzeug += climb_mz;
        $scope.app.params.my_mdl_Flugzeug += climb_my;
 })
}, timingIntervall_mov)
}
if(!active){
                            // return home
clearInterval(timerId mov);
   timerId_mov=-1;
   $scope.app.params.mz_mdl_Flugzeug = 0;
  $scope.app.params.my_mdl_Flugzeug = 0;
}
```

```
CODE → Schritt3.txt
```

Der letzte Schritt ist das Verknüpfen der Funktion mit dem CLickereigniss des Buttons:

```
$scope.click_but_Takeoff = function () {
move(true);
}
```

Das Flugzeug sollte nun nach dem Betätigen des Buttons mit dem Steigflug beginnen.

Der weitere Code könnte folgendermaßen aussehen:

```
// $scope, $element, $attrs, $injector, $sce, $timeout, $http, $ionicPopup,
and $ionicPopover services are available
/*
    m.schubert 2019
    HTL Wiener Neustadt
*/
var timerId_rot = -1; //timer identifier
var timerId_mov = -1; //timer identifier
var timingIntervall_rot = 30; //miliseconds for timer
var timingIntervall_mov = 30; //miliseconds for timer (climb & looping)
var climb_mz = 0.01; // climb parameter
var climb_my = 0.005;
```

```
m.schubert
```

```
//looping rad/s
var omega_loop = 0.4;
var r_loop = 1; // radius looping
var starthight_loop = 0.3; // starting loop at hight
var mzstart_loop = 0;
var mystart_loop = 0;
var phi_loop = 0;
 if
(!$scope.app.params.rz_mdlit_Luftschraube){$scope.app.params.rz_mdlit_Luftschr
aube = 0} // set to startvalue
if (!$scope.app.params.mz_mdl_Flugzeug){$scope.app.params.mz_mdl_Flugzeug = 0}
// mdl_pos z
 if (!$scope.app.params.my_mdl_Flugzeug){$scope.app.params.my_mdl_Flugzeug =
0}
        // mdl_pos z
 if (!$scope.app.params.mz_mdl_Flugzeug){$scope.app.params.rx_mdl_Flugzeug =
0}
     // mdl_rot x
$scope.$watch("app.params['Drehzahl']",function()
                                                                       11
call function when value changes
{
  if ($scope.app.params.Drehzahl > 0){ //check propeller rotation
    rotate(true);
      $scope.view.wdg['lab_Info'].text = 'Motor läuft';
      $scope.view.wdg['but_Takeoff'].visible = true;
  }
  else {
    rotate (false);
      $scope.view.wdg['lab_Info'].text = 'Motor aus';
      $scope.view.wdg['but_Takeoff'].visible = false;
  }
})
$scope.click_but_Takeoff = function () {
move(true);
}
$scope.click but Home = function () {
move(false);
}
function rotate (active){
console.log('function call -rotate- '+ active);
if (active && timerId_rot == -1){
timerId_rot = setInterval(function() {
   $scope.$apply(function(){
                                         // code called every timingIntervall
     if ($scope.app.params.rz_mdlit_Luftschraube >=
360){$scope.app.params.rz_mdlit_Luftschraube =0}
    $scope.app.params.rz_mdlit_Luftschraube += 10;
  })
}, timingIntervall_rot)
```

```
aschubert
```

```
}
if(!active){
clearInterval(timerId_rot);
   timerId_rot=-1;
}
}
function move (active){
console.log('function call -move- '+ active);
if (active && timerId_mov == -1){
timerId_mov = setInterval(function() {
   $scope.$apply(function(){
 if ($scope.app.params.my_mdl_Flugzeug<starthight_loop){ // climb to desired
hight
        $scope.app.params.mz_mdl_Flugzeug += climb_mz;
        $scope.app.params.my_mdl_Flugzeug += climb_my;
        mzstart_loop = $scope.app.params.mz_mdl_Flugzeug;
       mystart_loop = $scope.app.params.my_mdl_Flugzeug;
      }
      else{
        phi_loop=phi_loop + (omega_loop*timingIntervall_rot/1000);
//looping
          $scope.app.params.mz mdl Flugzeug = mzstart loop +
r_loop*Math.cos(phi_loop + Math.PI*3/2);
          $scope.app.params.my_mdl_Flugzeug = mystart_loop +
r_loop*Math.sin(phi_loop + Math.PI*3/2) + r_loop;
          $scope.app.params.rx_mdl_Flugzeug = -1*phi_loop/Math.PI*180;
                                                                           11
rotate airplane
      }
 })
}, timingIntervall_mov)
}
if(!active){
                            // return home
clearInterval(timerId_mov);
   timerId mov=-1;
   $scope.app.params.mz_mdl_Flugzeug = 0;
 $scope.app.params.my_mdl_Flugzeug = 0;
 $scope.app.params.rx_mdl_Flugzeug = 0;
}
}
```