



ЛАБОРАТОРНАЯ РАБОТА №4

Создание приложения дополненной реальности. Разработка AR- Приложения для совместной работы с сервером IoT/IIoT (часть III из III)

Оглавление

1. Введение.....	2
2. Разработка Приложения ДР для совместной работы с сервером IoT/IIoT ..	4
3. Практическое задание.....	33

Дата	Автор	Версия	Описание
09.12.2020	Пирогова М.А.	1	Разработка ЛР № 4
09.12.2021	Пирогова М.А.	2	Коррекция ЛР № 4
24.05.2026	Пирогова М.А.	3	Редактирование для курса ТДР

1. Введение

Применение Vuforia Studio Enterprise Suite в качестве платформы разработки **AR**-приложений Интернета Вещей (**IoT**) и промышленного Интернета Вещей (**IIoT**) значительно упрощает и сокращает сам процесс разработки **AR**-приложения, а за счёт облачного размещения результата разработки – облегчает доступ к самому **AR**-приложению в любом месте при наличии интернет-соединения.

AR-приложения, разработанные в **Vuforia Studio**, размещаются в «облаке» и загружаются на устройство пользователя только в момент использования. Однако реальная промышленная потребность в **AR**-приложениях часто связана с необходимостью взаимодействия с пространством данных интернета вещей. В этом случае Приложение **AR** должно быть настроено на получение данных от сервера **IoT/IIoT**, а физический объект, отслеживаемый приложением ДР, должен являться источником данных для интернета вещей. Иными словам и, на него должен существовать т.н. **Цифровой двойник (ЦД)** на сервере **IoT/IIoT**.

В данной ЛР №4 необходимо модифицировать ранее разработанный **AR**-Проект, использующий способ таргетирования **ThingMark**. Средствами **Vuforia Studio** требуется обеспечить связь с сервером **IoT/IIoT**; на сервере **IoT/IIoT** необходимо выбрать значения соответствующих свойств реального объекта (в нашем случае – это **3D-Принтер MakerBot Replicator 2**) и передать эти данные в заранее подготовленные в **AR**-Проекте виджеты для их отображения.

Предполагаемый внешний вид разработанного в рамках ЛР №4 Приложения **AR** – см. ниже.



2. Разработка Приложения ДР для совместной работы с сервером IoT/IIoT

Как и в предыдущих Лабораторных работах, будем считать, что тренировочный сервер ThingWorx Experience Service (TWx ES) находится по адресу (актуальный URL уточнять у преподавателя):

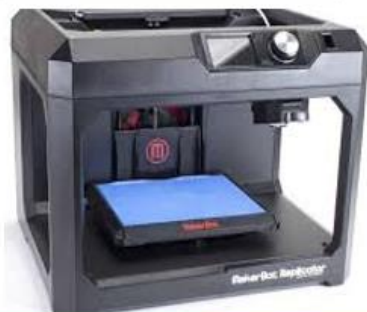
<http://rrr.kkk.com:2019>

В данной ЛР будет использован тренировочный сервер IoT/IIoT – Thing Worx Foundation (TWx Fndtn), доступный по адресу (актуальный URL уточнять у преподавателя):

<http://rrr.kkk.com:8085/Thingworx>

На сервере заранее [администратором] создаются цифровые двойники умных подключаемых вещей (SCT) – **Thing Model**, каждая из которых имеет уникальное имя (свойство **name**). В данном описании ЛР будем использовать **Thing Model** с именем **Replicator01**. Для выполнения индивидуальных заданий обучающимся будут предоставлены уникальные имена ЦД для их работы. Для данного описания:

☐	# HeadTemperature
☐	l23 HUMIDITY
☐	-T- Model
☐	l23 Model3DLoad
☐	l23 PLA_Filament
☐	# PRESSURE
☐	# Price
☐	# TEMPERATURE



При разработке AR-приложения считаем, что Физический объект (3D-принтер) - содержит в своем составе такие обязательные сенсоры и датчики, как :

- Измеритель температуры сопла в экструдере (в градусах Цельсия) ; # [HeadTemperature](#)
- Индикатор готовности выполнения работы (в процентах готовности); l23 [Model3DLoad](#)
- Указатель запаса рабочего тела на бобине (в сантиметры); l23 [PLA_Filament](#)
- Идентификатор устройства – уникальное имя для распознавания при сетевом опросе (строка символов) -T- [Model](#)

Кроме того, для контроля параметров окружающей среды в составе сенсоров устройства также есть:

- Термометр воздуха рядом с принтером (в градусах Цельсия); # [TEMPERATURE](#)
- Датчик атмосферного давления (в гектопаскалях) # [PRESSURE](#)
- Сенсор определения влажности воздуха рядом с рабочей открытой зоной (в процентах). l23 [HUMIDITY](#)

Кроме того в список свойств добавлена цены # [Price](#)

Перечисленные свойства определены на сервере **Thing Worx Foundation (TWx Fndtn)** [администратором] →

← → ↻ 🏠 ⚠ Не защищено | web0.pts-russia.com:8085/Thingworx/Composer/Apps/Composer/#/modeler/details/Thing~Replicator01/properties ☆ ⚙ 👤 ⋮

Сервисы 🌐 Новая вкладка 🖼 Викторович Алекс... hi Hi.ru

thingworx 🔍 SEARCH + NEW 📄 Import/Export ▾ User08 ▾ ? Help ▾ 📄

Set Project Context +

Recent 12 🕒

- ExcavatorDigitalTwin
 - Replicator01
 - PatientCard
 - Blue background - 2
 - Blue background
 - patient-human-gorba...
 - led-display-gorbanev
 - gorbanev-text-style
- A06m
 - A06m-19
- MakerBot_001
- Gorbanev002
- MediCare-001
- Logotype MPEI Trans...
- Gorbanev001
- NewPhoto
- Ris001
- Excavator-DT
- User08
- User07
- User05
- User04
- User03

Thing: Replicator01 ? To Do ▾ **Save** Cancel More ▾

📄 General Information 📄 **Properties and Alerts** <> Services ⚡ Events 📡 Subscriptions 🔒 Permissions 🕒 Change History 👤 View Relationships

Properties | **Alerts** 🔍 Choose category ▾

▼ **My Properties** + Add Duplicate Delete Manage Bindings Refresh

☐	Name	Actions	Source	Default Value	Value	Alerts	Category	Additional Info	📄	🔒	☰
☐	# <u>HeadTemperature</u>	📄			58.12126534	+ 0	5 to 150 Celsius	✓	✓		
☐	123 <u>HUMIDITY</u>	📄			24	+ 0	0 to 100 %	✓	✓		
☐	-T- <u>Model</u>	📄			MaterBot-02	+ 0		✓	✓		
☐	123 <u>Model3DLoad</u>	📄			53	+ 0	0 to 100 %	✓	✓		
☐	123 <u>PLA Filamet</u>	📄			430	+ 0	5 to 512 cm	✓	✓		
☐	# <u>PRESSURE</u>	📄			1024	+ 0	100 to 1600 HectoPascal	✓	✓		
☐	# <u>Price</u>	📄			2000	+ 0	200 to 20000 \$	✓	✓		
☐	# <u>TEMPERATURE</u>	📄			10	+ 0	-20 to 80 Celsius	✓	✓		

▼ **Generic**

Name	Actions	Source	Default Value	Value	Alerts	Category	Additional Info	📄	🔒	☰
-T- description	📄				+ 0	Metadata		✓		
-T- name	📄			Replicator01	+ 0	Metadata		✓		
tags	📄					Metadata		✓		
thingTemplate	📄			GenericThing		Metadata		✓		

Шаг 1.

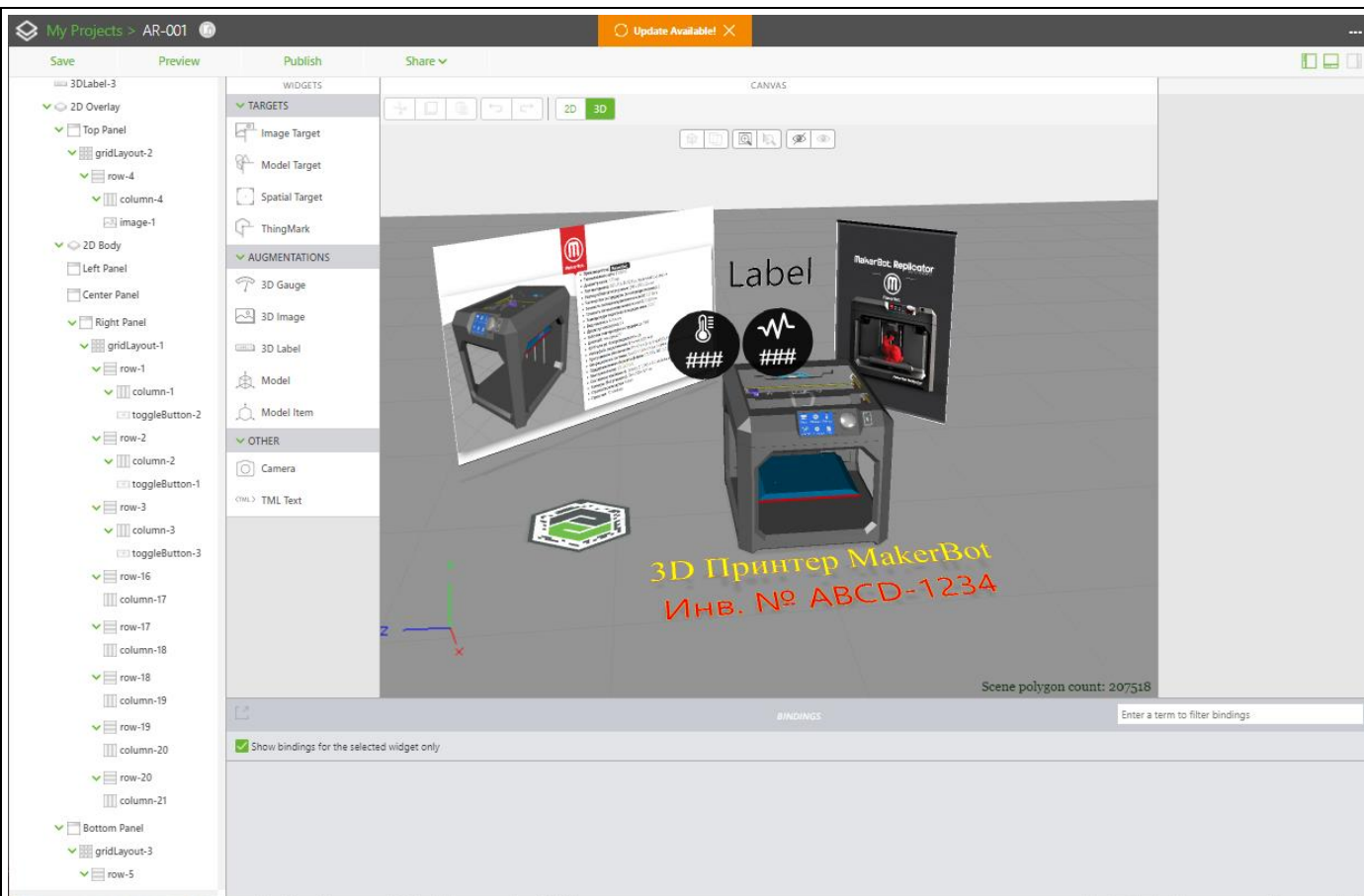
Стартуем известным вам способом **Vuforia Studio** (см. ЛР №2) и открываем Проект ДР с методом таргетирования – **ThingMark**.

ВАЖНО!! В открывшемся Проекте д.б. следующие виджеты из набора **3D**:

- как минимум один **3D Gauge**;
- как минимум один **3D Label**.

К ним, по ходу выполнения **ЛР** будет добавлен как минимум один **2D Label** и **2D Gauge**.

Все эти виджеты будем использовать для отображения свойств **ЦД** целевого устройства (**3D Принтера**), получаемых от сервера **IoT/IIoT**

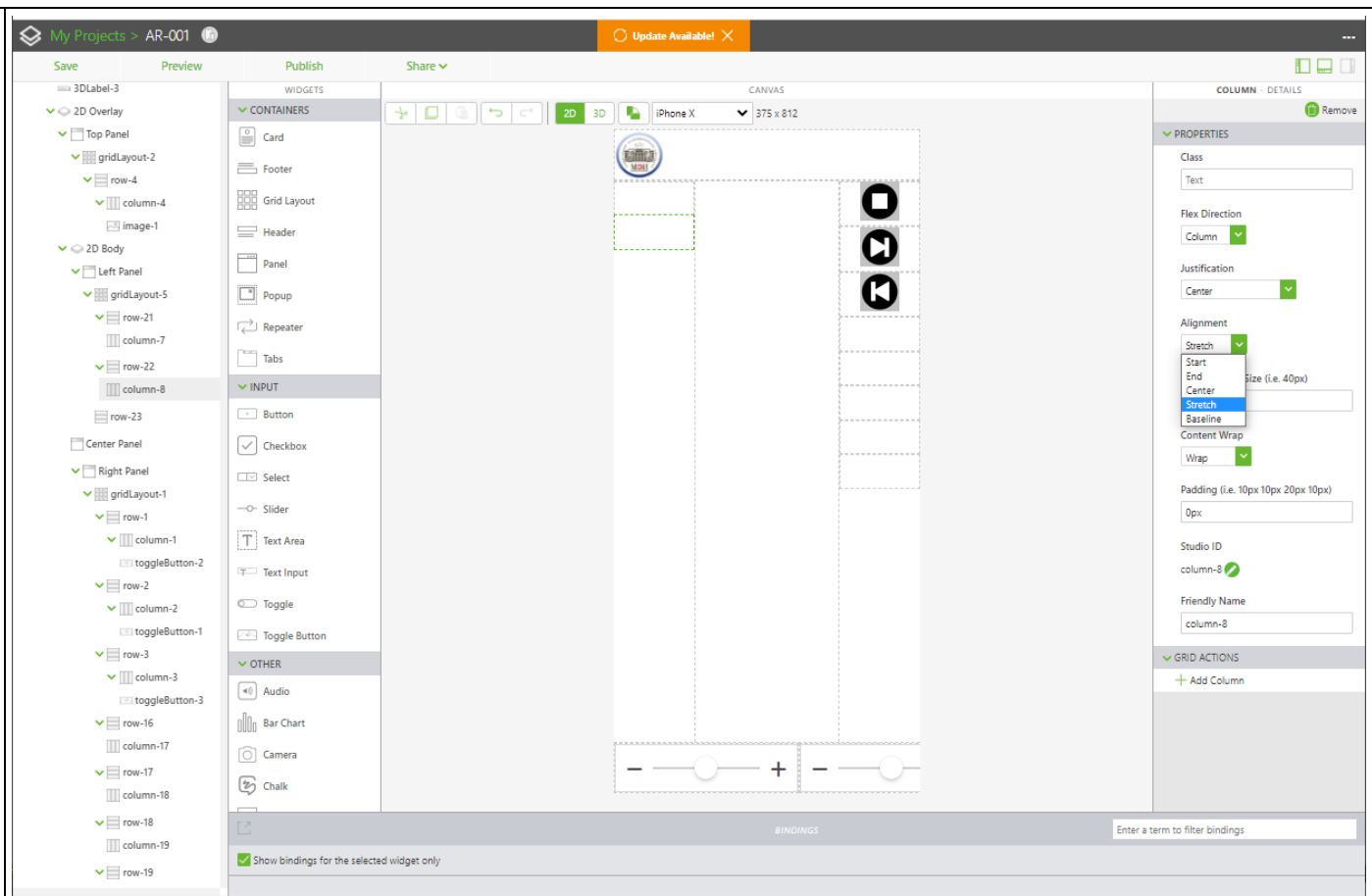


Шаг 2.

Подготовим заранее **2D**-Виджеты типа **Label** и **Gauge** для отображения на «стекле» значений свойств **ЦД**, получаемых с сервера **TWx Fndtn**.

Для этого на левую панель канвы **2D** переносим **Grid Layout**, формируем в нем минимум 2 ряда, один из которых будет предназначен для **Label**, а второй – для **Gauge**.

Колонку для **Label** предлагается выравнять методом **Center/Center**, а колонку для **2D Gauge** – методом **Center/Stretch**.

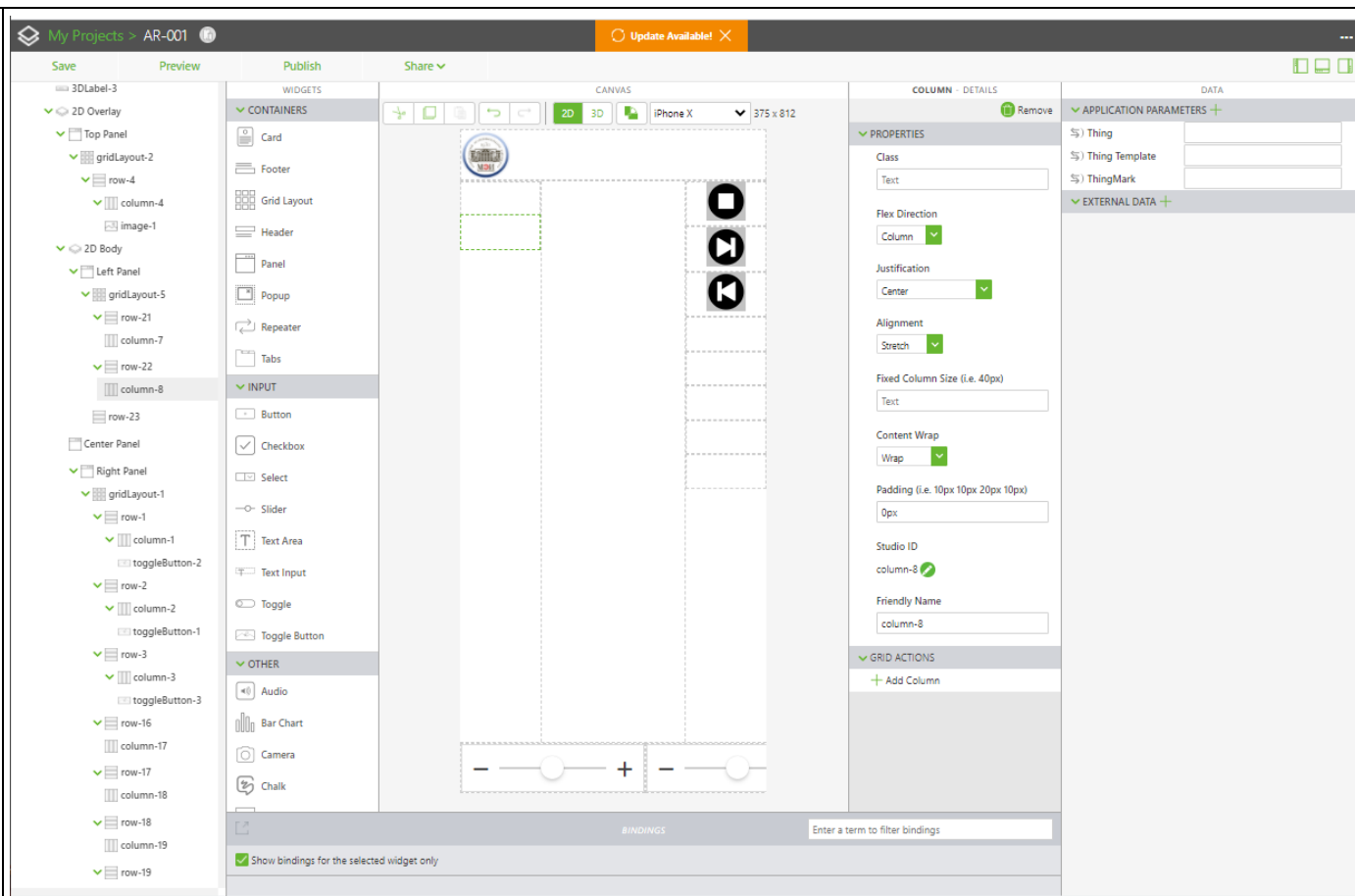


Шаг 3.

Начинаем подключать сервер **TWx Fndtn**.
Для этого в редакторе **Vuforia Studio**
открываем правую панель →



В открывшейся правой панели для связи с
внешними данными сервера **TWx Fndtn**
выбираем опцию **EXTERNAL DATA +**



Шаг 4.

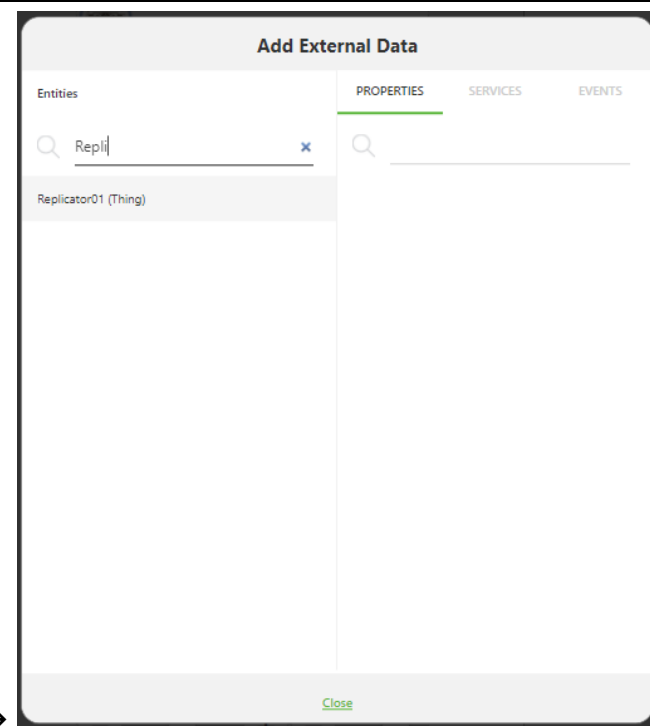
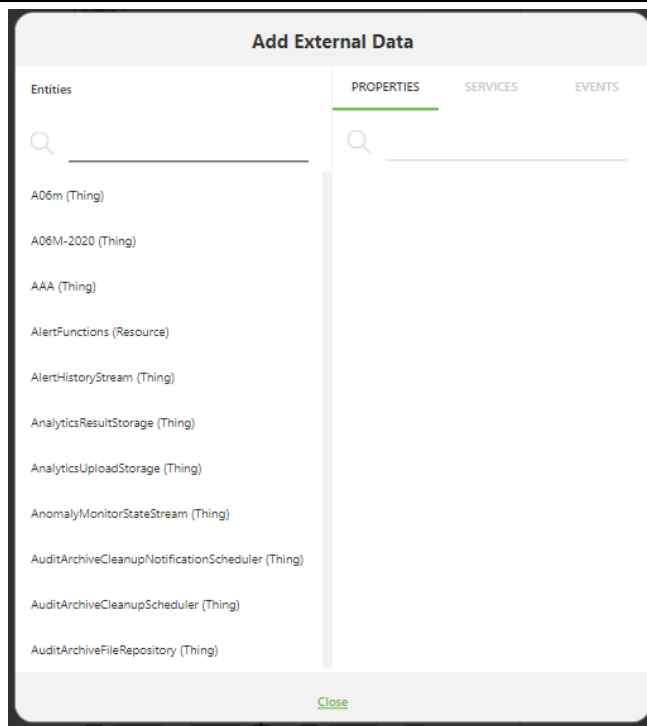
В открывающемся окне, в левой части, находим имя используемого в данном описании ЦД → **Replicator01**.

Для ускорения поиска в очень обширном списке можно воспользоваться средствами фильтрации поиска →



Набрав в строке первые буквы искомого имени.

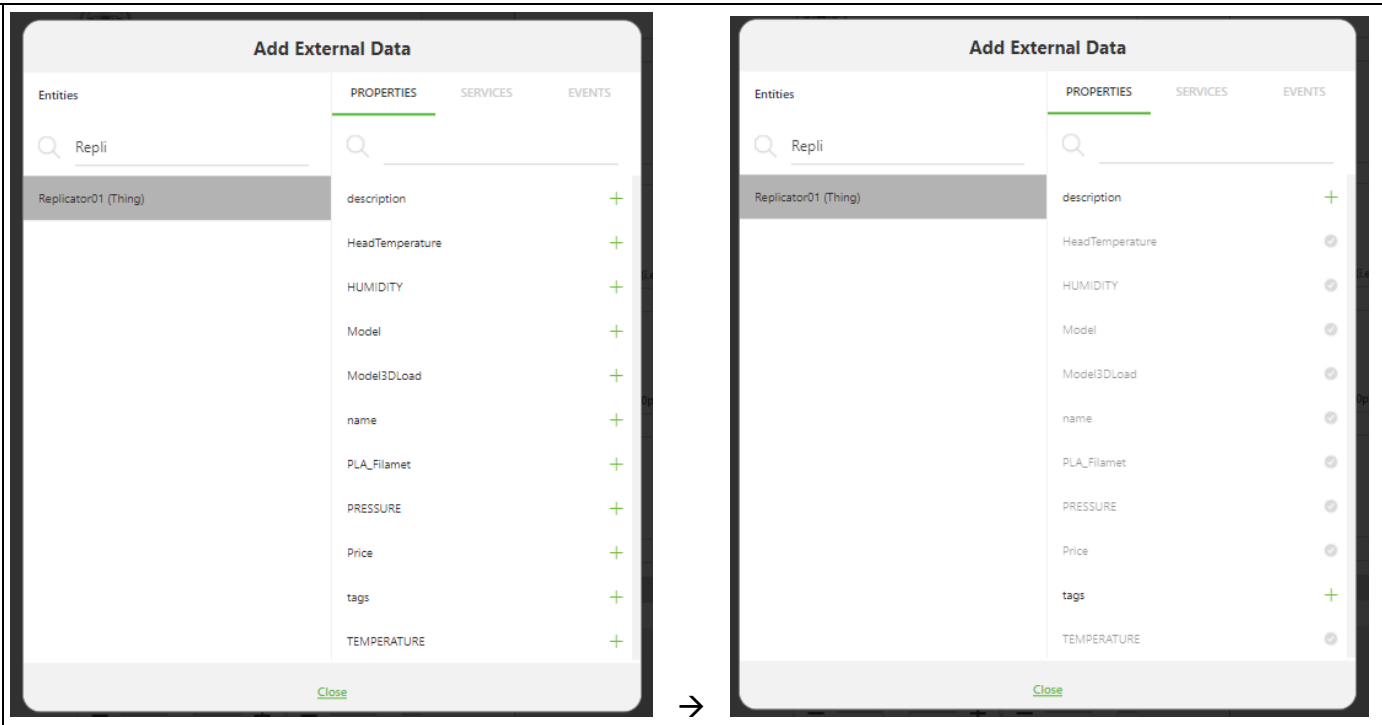
Найдя нужное имя (**Replicator01**), выбираем его →



Шаг 5.

В правой части экрана появился список свойств **ЦД**. Для выбора свойств **3D Принтера**, которые мы хотим отображать в

Приложении ДР, нажимаем на символ **+** справа от каждого из выбранных свойств (см. стр. 4). По окончании процедуры – клавиша **Close**.



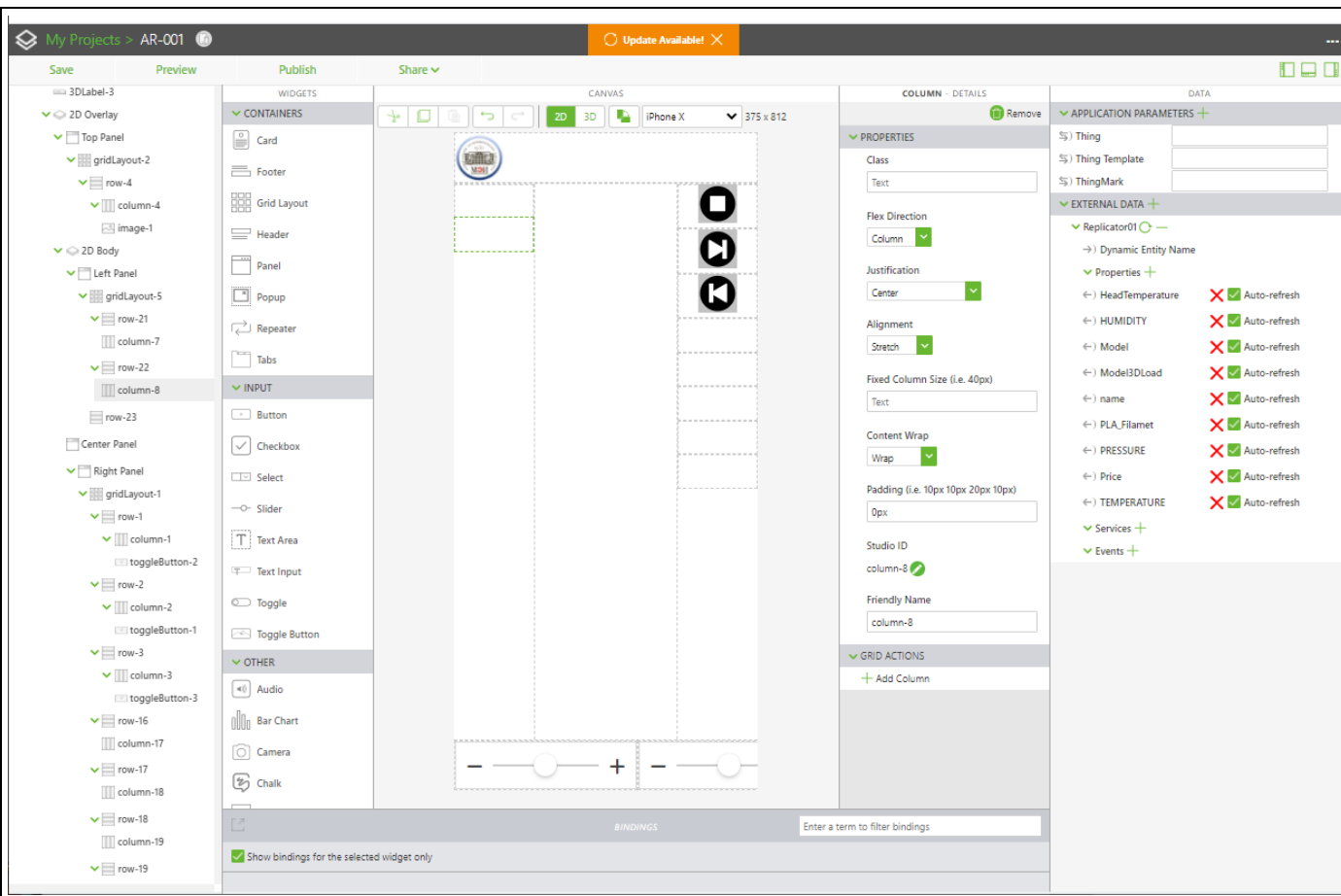
Шаг 6.

В результате сформировался список передаваемых от сервера **TWx Fndtn** свойств **ЦД (Replicator01)** подключаемой умной вещи – **3D Принтера**.

Для визуализации этих свойств в Приложении ДР нам достаточно только считывать их значения. Сервер **TWx Fndtn** предоставляет нам для этого **соответствующий сервис - GetPropertyValues**. Его надо выбрать из списка доступных сервисов.

Для этого вновь выбираем опцию

EXTERNAL DATA +



Шаг 7.

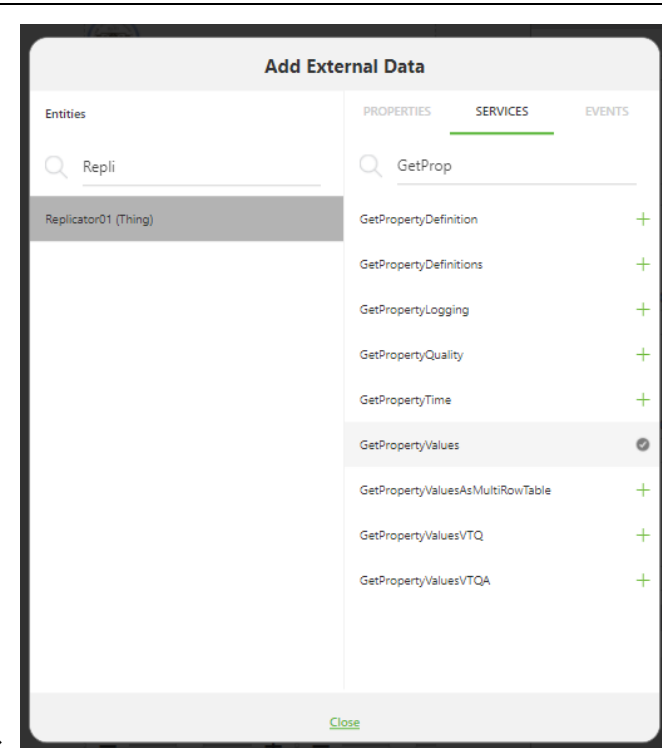
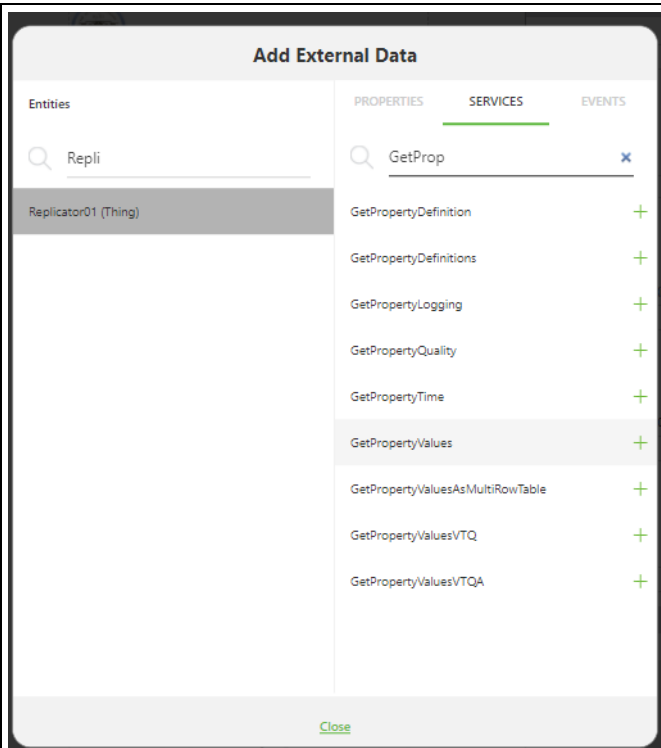
В открывающемся окне, в левой части, находим имя используемого в данном описании ЦД → **Replicator01**.

Для ускорения поиска в очень обширном списке можно воспользоваться средствами фильтрации поиска →



набрав в строке первые буквы искомого имени.

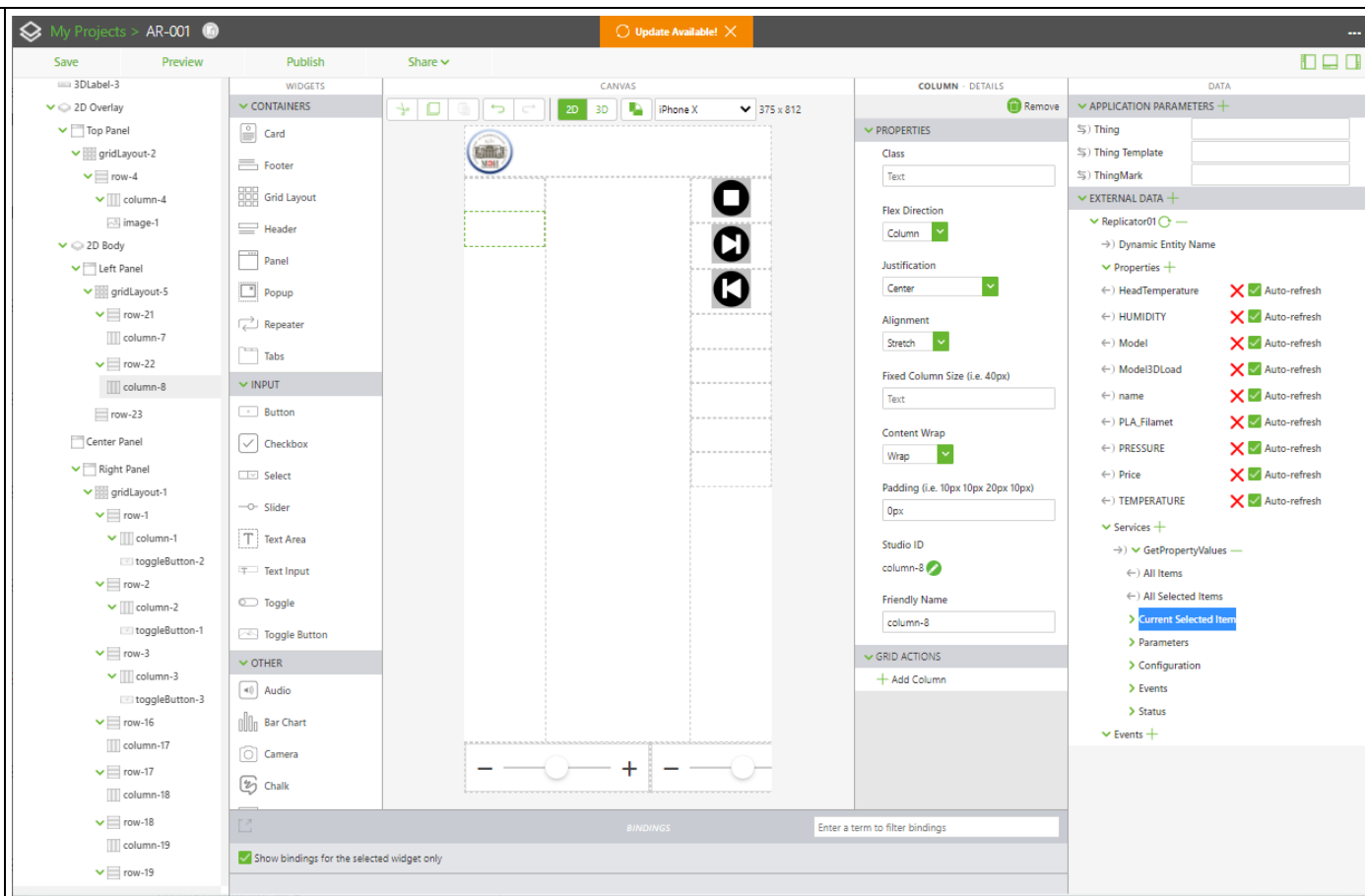
Найдя нужное имя (**Replicator01**), выбираем в правой части опцию **SERVICES** и находим – тоже с использованием фильтрации – сервис **GetPropertyValues**. Выбираем напротив имени - **+** → **Close**



Шаг 8.

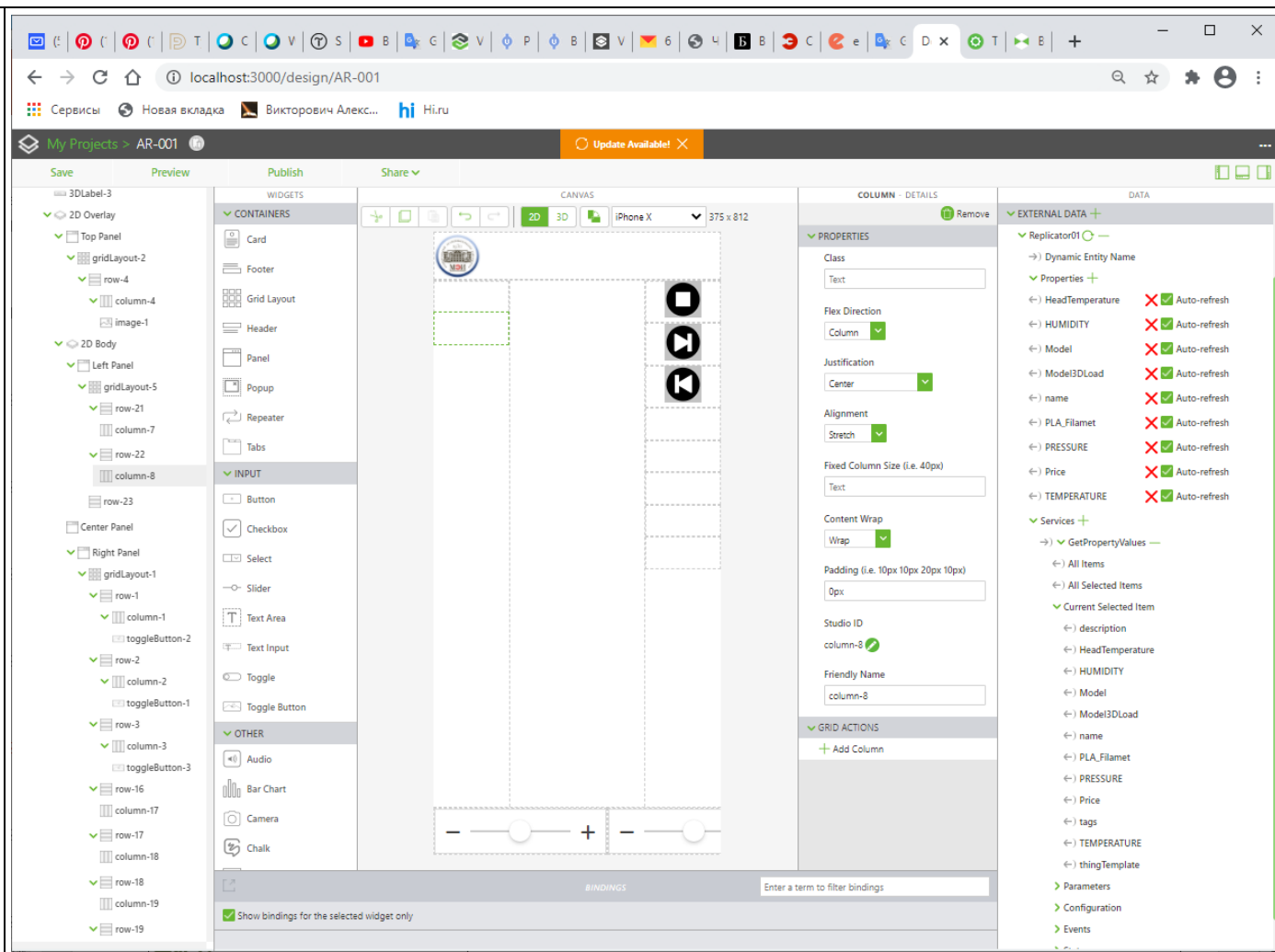
В результате в правой части интерфейса редактора **Vuforia Studio** появляется информация о подключенном сервисе сервера **TWx Fndtn**.

Для того, чтобы убедиться в корректности подключения, раскроем список **Current Selected Item** →



Шаг 9.

Убеждаемся в том, что раскрывшийся список содержит все интересующие нас свойства.



Шаг 10.

Все наши свойства объявлены обновляющимися (  Auto-refresh).

Для отработки этого автообновления сервисом **GetPropertyValues**, сконфигурируем его → в опции **Configuration** нашего сервиса необходимо отметить **Auto-refresh**, и задать величину обновления в секундах – **Auto-refresh Rate**:

Services +

→  **GetPropertyValues** —

← All Items

← All Selected Items

> Current Selected Item

> Parameters

 **Configuration**

☐ Invoke On Startup

☐ Invoke On Entity Change

 Auto-select first row

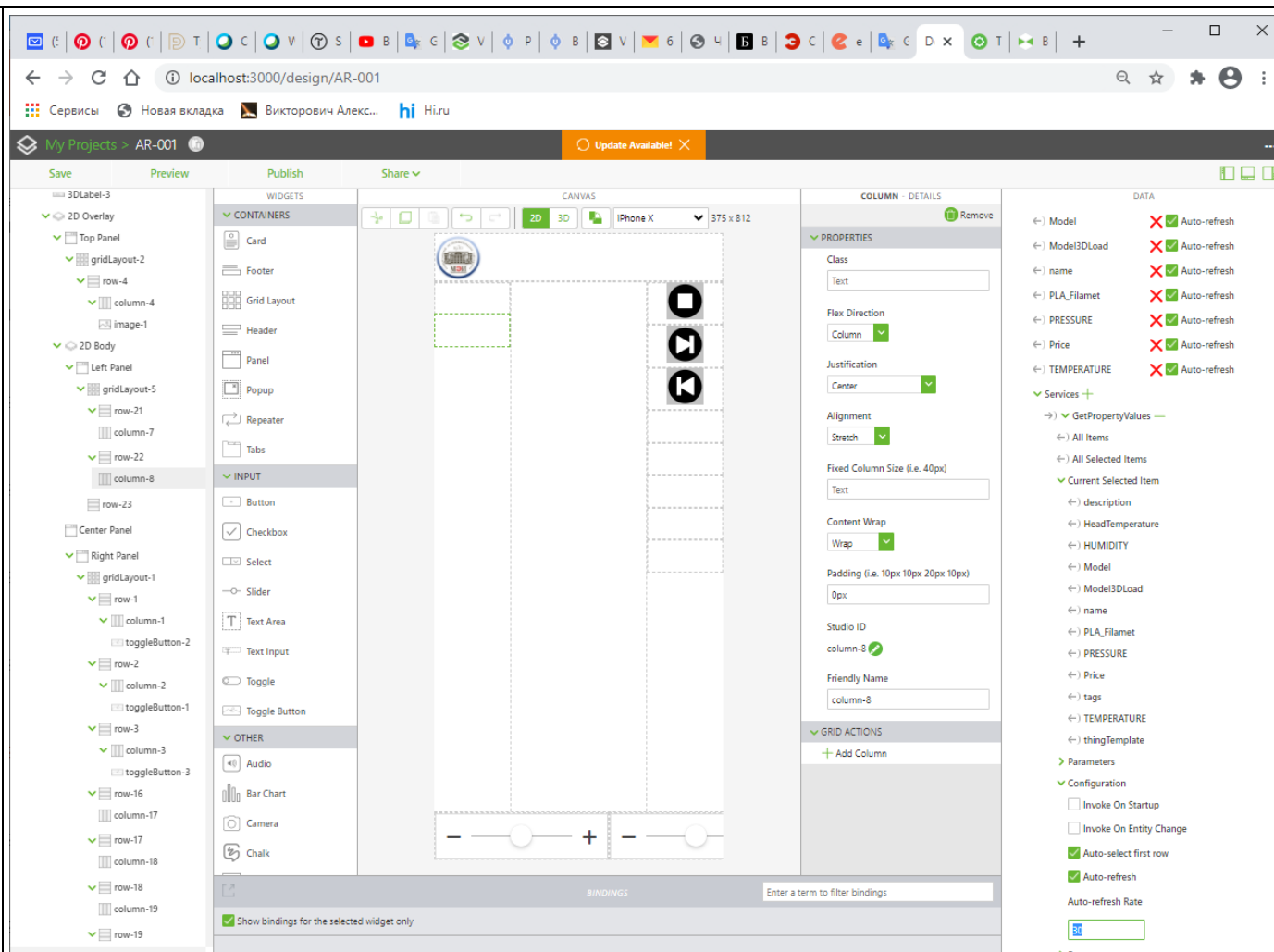
 Auto-refresh

Auto-refresh Rate

30

> Events

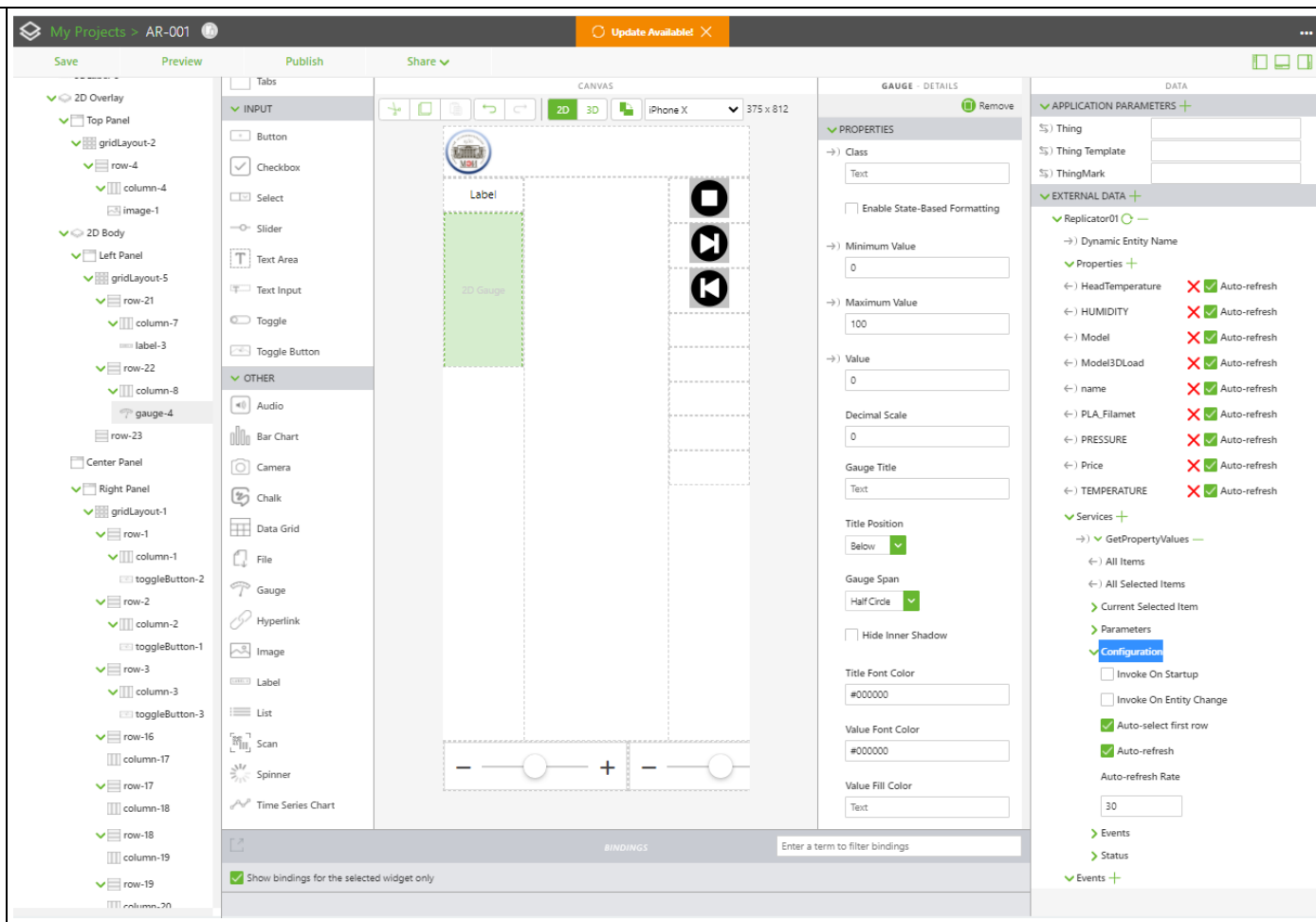
> Status



Шаг 11.

Возвращаемся в канву **2D**, где мы начали подготавливать виджеты для плоского экрана («стекла»), отображающие значения выбранных свойств **ССТ**.

Расположим в первой строке левой панели виджет **2D Label**, а во второй – **2D Gauge**.



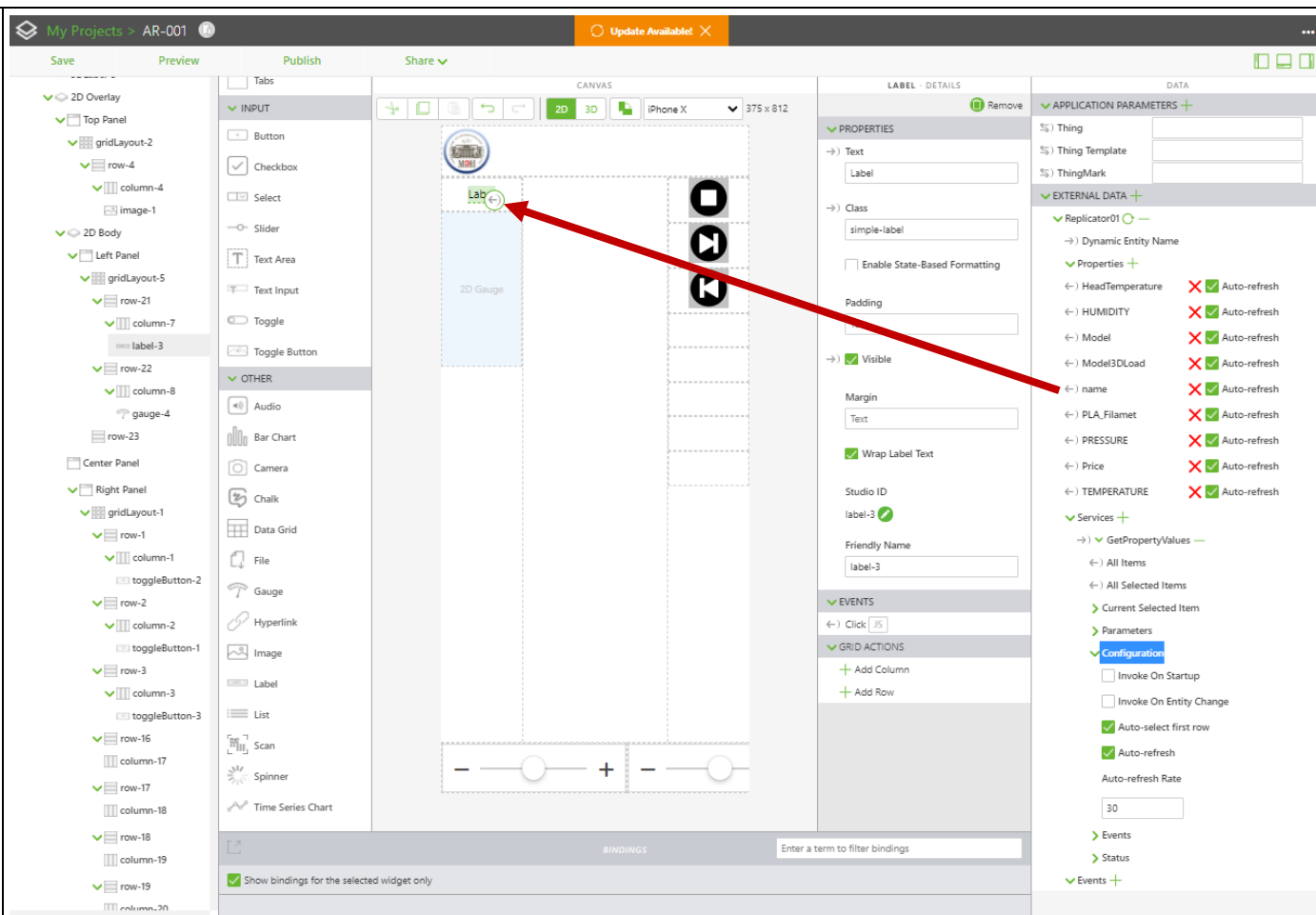
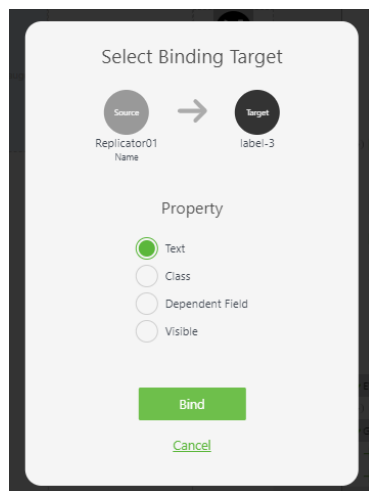
Шаг 12.

Предлагается на «стекле» отображать следующие свойства 3D Принтера:

- **Label** → поместим наименование ЦД – свойство **name**;
- **Gauge** → поместим значение температуры «за бортом».

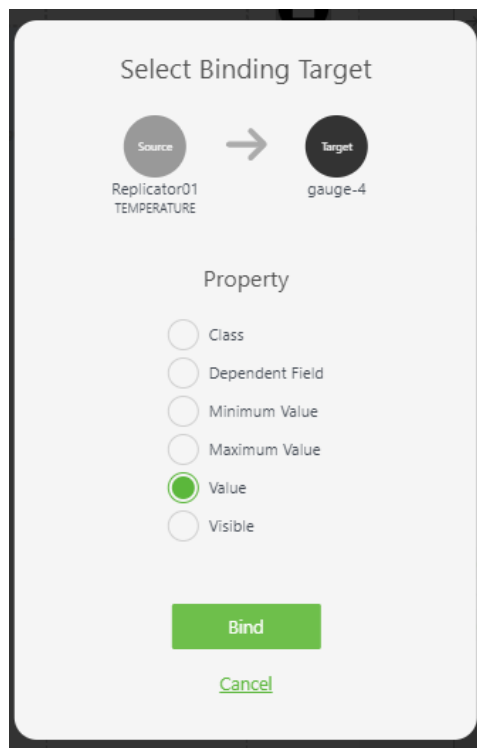
Начинаем с **Label** →

Предлагается в качестве свойства, отображаемого в этом виджете выбрать свойство, относящееся к классу базовых (**Generic** в **TWx Fndtn**). В нашем случае это свойство **name**. См. распределение свойств на стр. 5 данного описания.

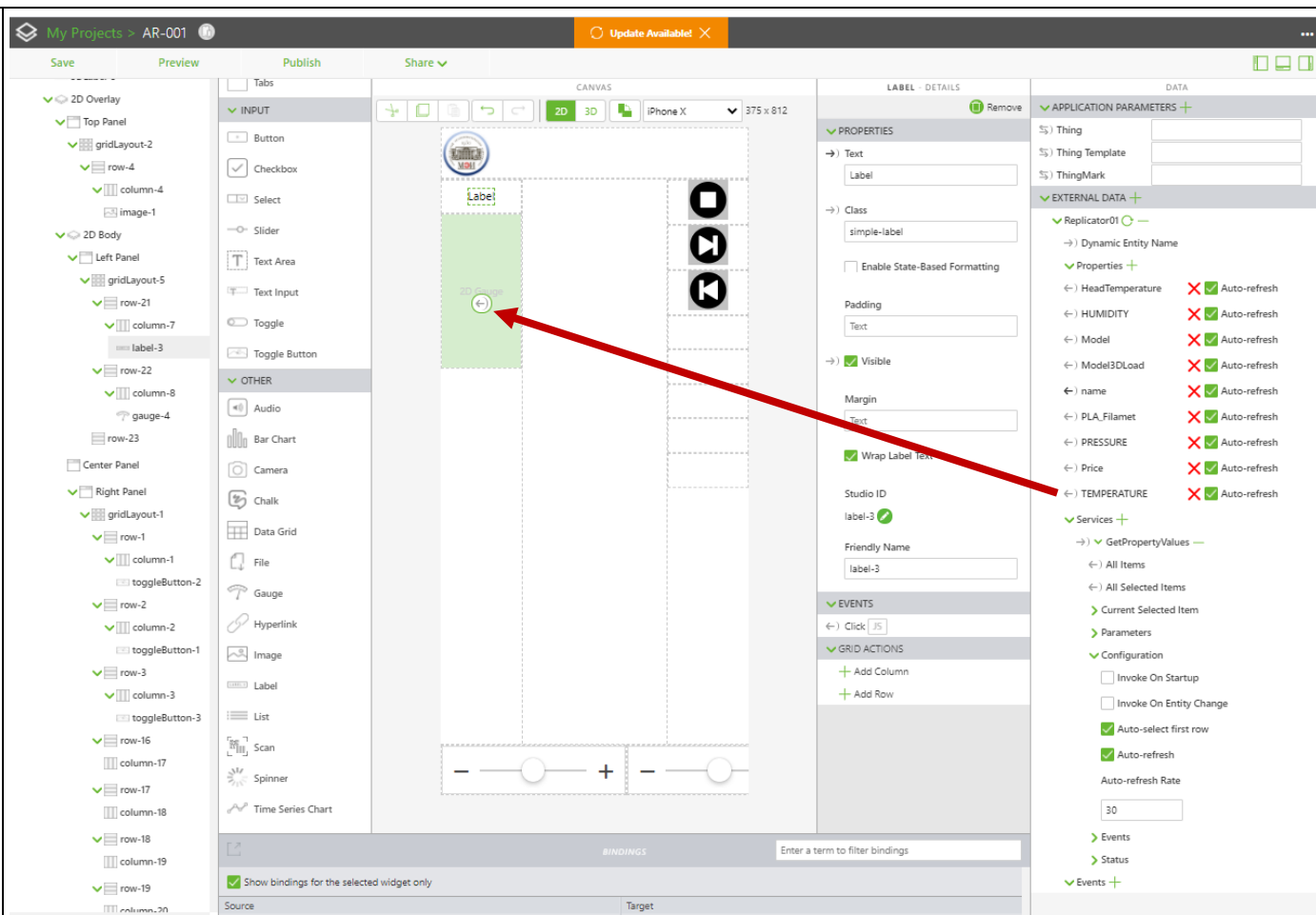


Шаг 13.

Связываем свойство **ЦД TEMPERATURE** с подготовленным **2D Gauge** →

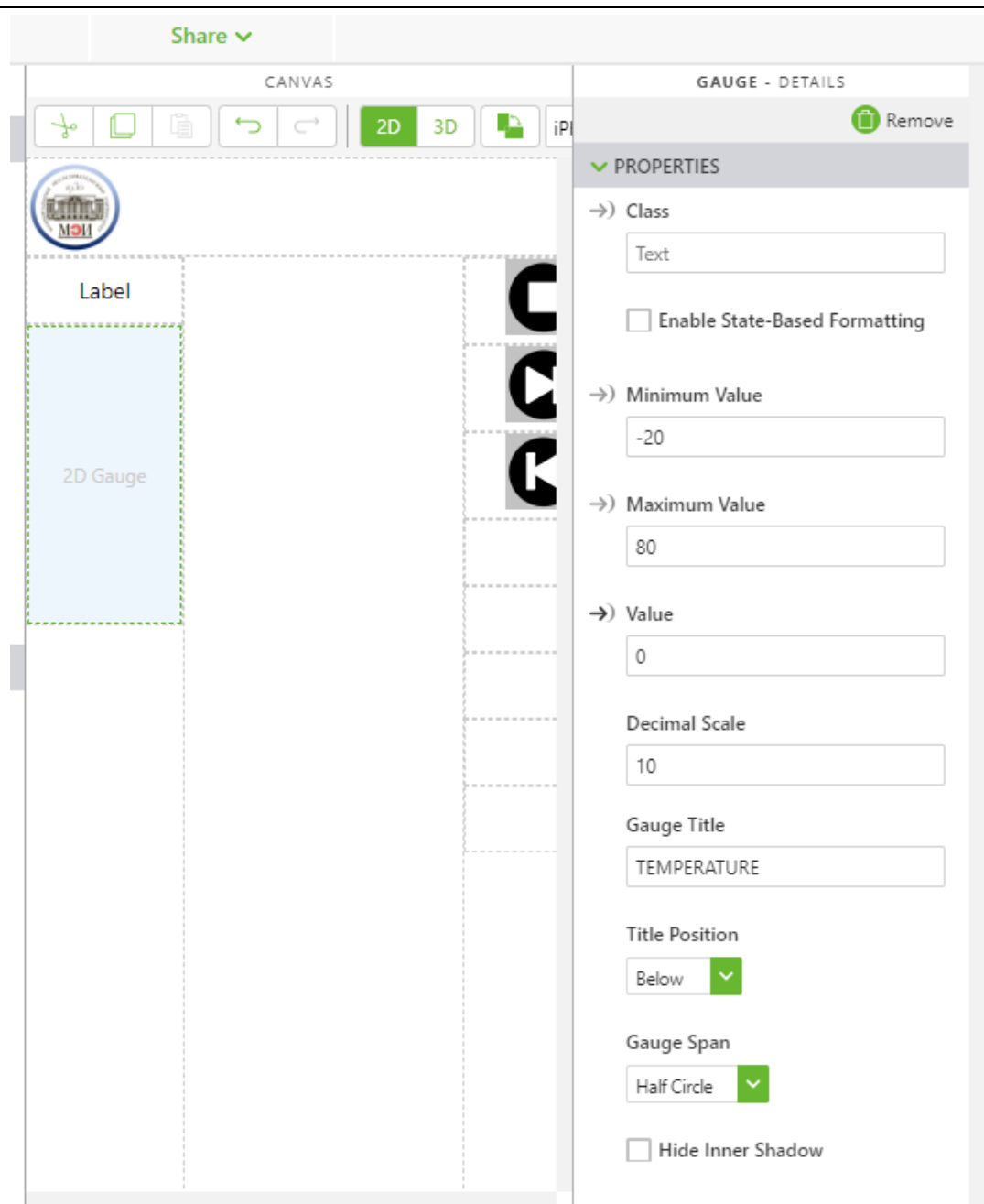


Обратите внимание, что для Виджета типа **Gauge** передача информации осуществляется в поле **Value**.



Шаг 14.

Настраиваем отображение **2D Gauge** → «циферблат» сделаем **полукруглым**, подпись расположим **под ним**, и сам «циферблат» будет размечен на цветовые зоны, которые будут отображать приближение к критическим зонам считываемого свойства. Сами критические зоны определены в **ЦД** на сервере **IoT/IIoT**. Как определено выше (см. начало раздела 2.) датчик **TEMPERATURE** дает значения в диапазоне **от -20 (нижняя критическая зона) до + 80 (верхняя критическая зона) градусов Цельсия**.



Шаг 15.

Промежуточный результат можно наблюдать в режиме **Preview**



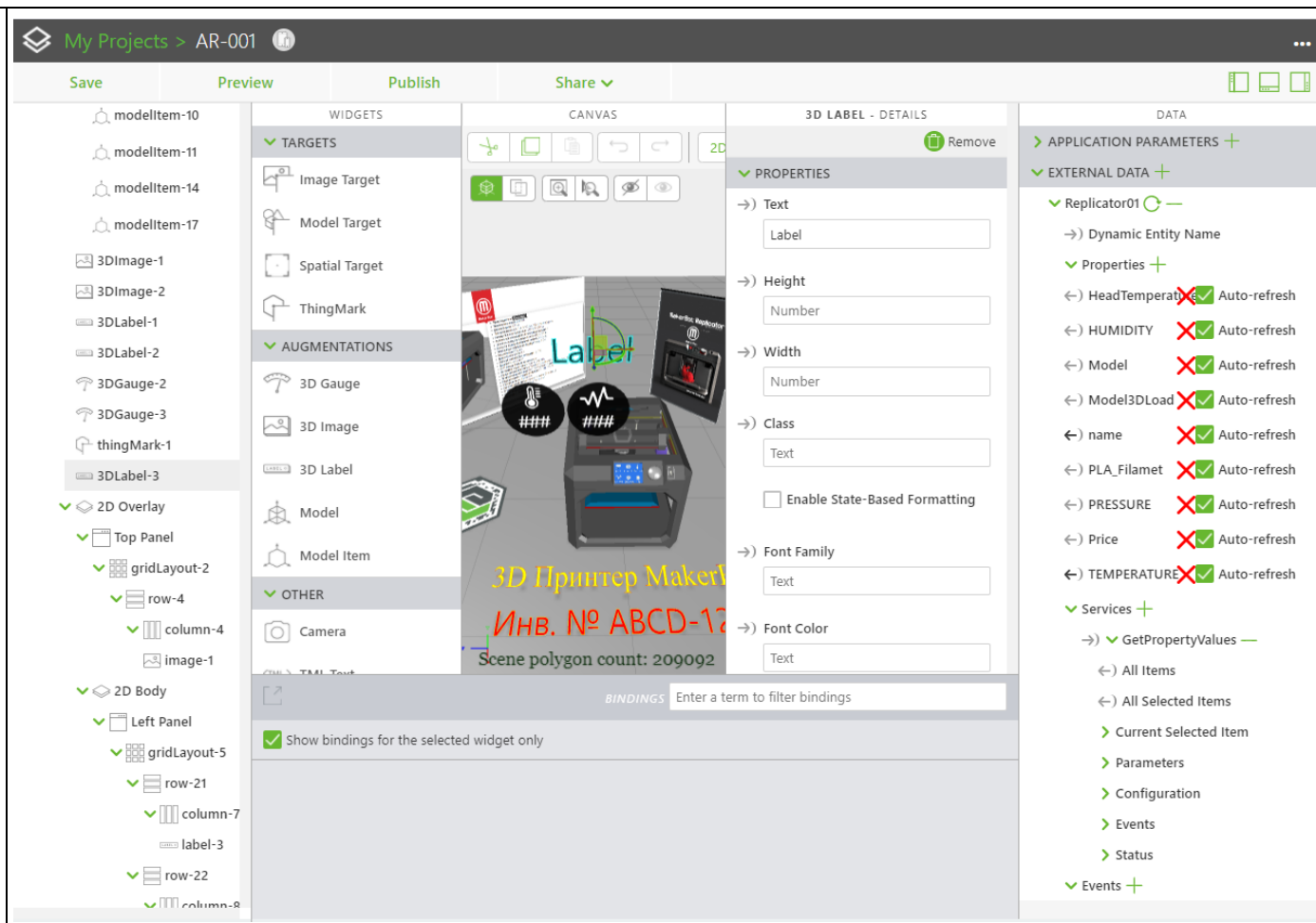
Шаг 16.

Продолжаем оформлять **Приложение AR** для связи его с сервером **IoT/IIoT → TWx Fndtn.**

Переходим в **3D-Канву**. В Проекте, который мы модифицируем, уже предусмотрены **3D-Виджеты** для отображения значений свойств целевого **ЦД** с сервера **TWx Fndtn**: **3D Label** и **3D Gauge**. Работа с этими виджетами была рассмотрена в **ЛР №2**. При желании вы можете добавить новые **3D-Виджеты**.

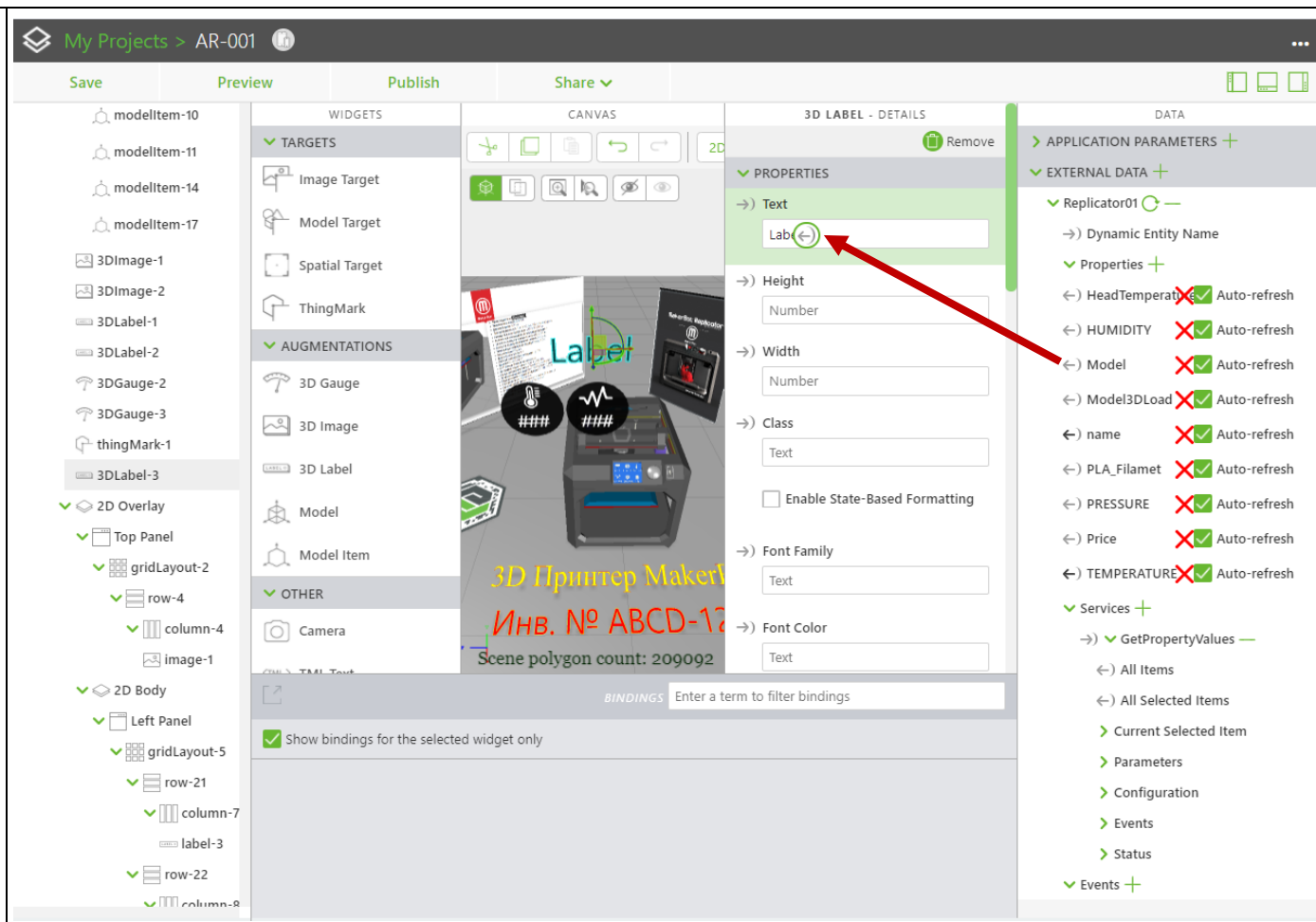
Начинаем со связывания **3D-Виджета 3D-Label** со значения одного из свойств **ЦД 3D Принтера**.

Предлагается в качестве такого свойства выбрать свойство **Model**.



Шаг 17.

Связываем **EXTERNAL DATA Model** с полем **Text** выбранного **3D Label**.



Шаг 18.

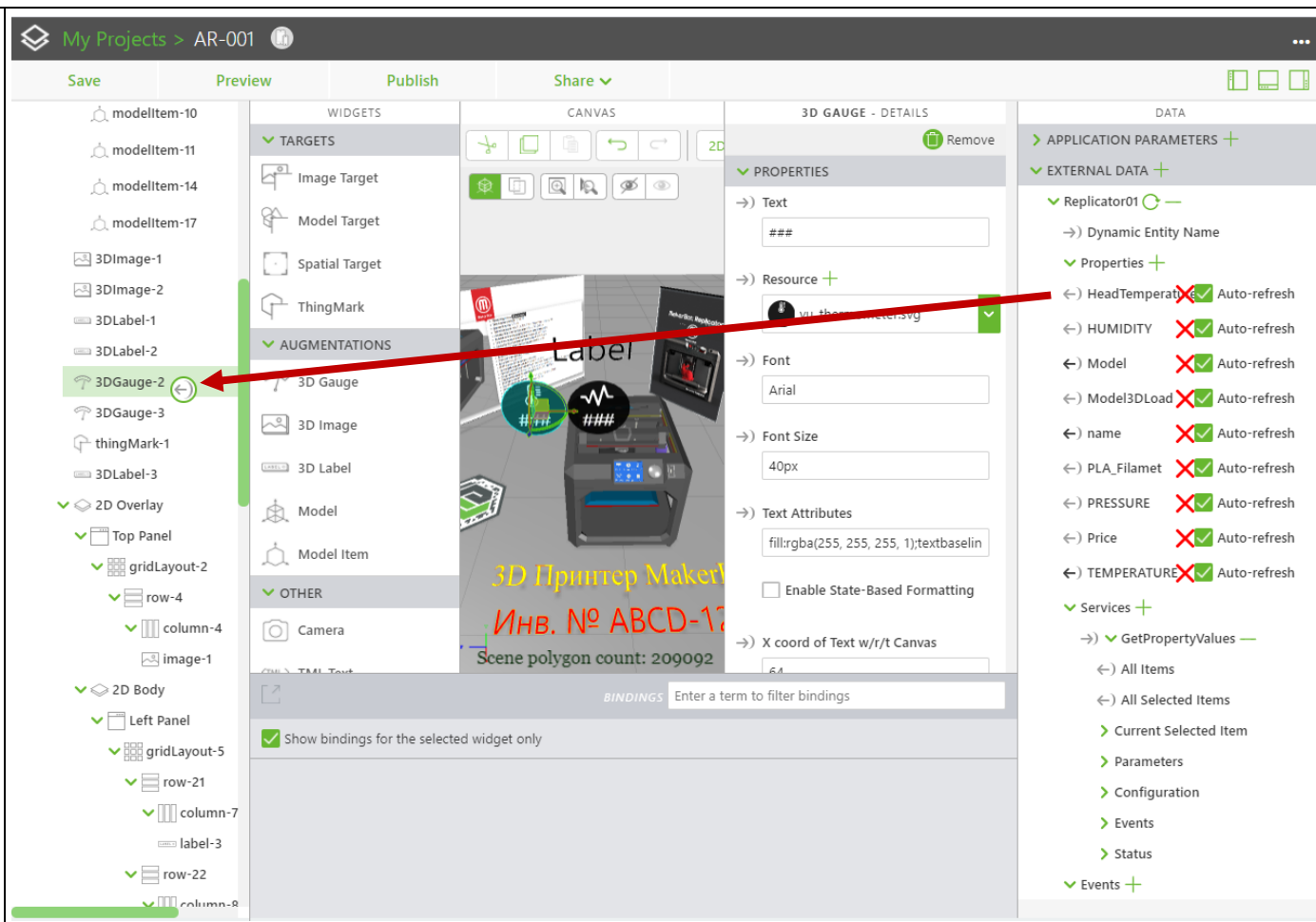
Результат → обратите внимание на изменение информации в поле связывания (**Bindings**), подтверждающее успешность операции связывания.

The screenshot displays the Vuforia Studio interface for project 'AR-001'. The left sidebar shows a hierarchy of widgets, with '3DLabel-3' selected under the '2D Overlay' group. The main canvas shows a 3D scene with a printer and a label. The '3D LABEL - DETAILS' panel on the right shows the 'PROPERTIES' section, where the 'Text' property is set to 'Label'. The 'BINDINGS' panel at the bottom shows a binding expression: 'app.mdl[\"Replicator01\"].properties[\"Model\"]' linked to the 'text' property of '3DLabel-3'. The 'DATA' panel on the right shows a list of application parameters, including 'Replicator01' and various sensor data like 'HUMIDITY', 'PRESSURE', and 'TEMPERATURE', each with an 'Auto-refresh' checkbox.

Шаг19.

Повторяем аналогичные действия для связывания **3D Gauge** со свойством **ЦД**, например, температурой сопла в экструдере (в градусах Цельсия) **HeadTemperature**;

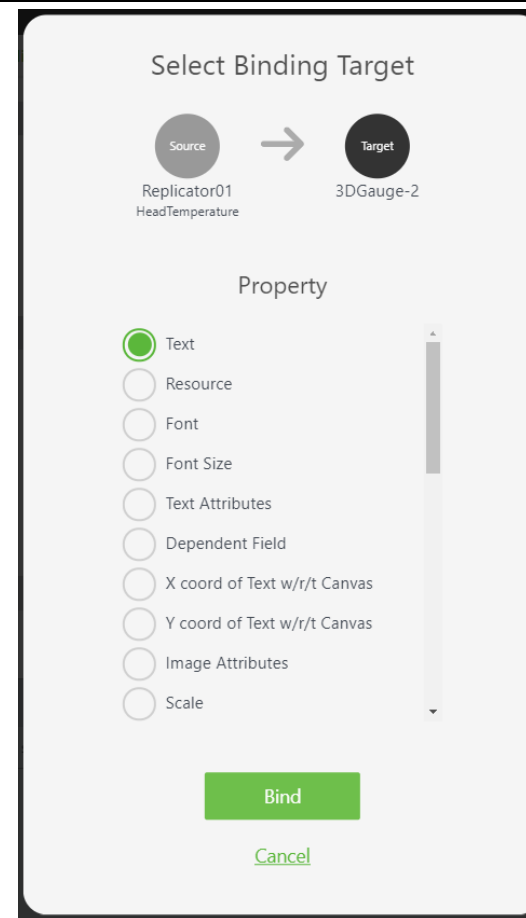
Обратите внимание!!! Здесь мы воспользовались еще одним способом выполнения связывания значения (из **EXTERNAL DATA HeadTemperature**) свойства с **3D-Виджетом**: **drag-n-drop** из поля **EXTERNAL DATA** в иерархию виджетов в поле **VIEW**.



Шаг 20.

В случае использования такого способа связывания, который был описан на предыдущем шаге, появляется промежуточный запрос в виде меню выбора **Property** (свойств), где нужно выбрать в нашем случае это **Text**.

Выбрать **опцию Bind** →

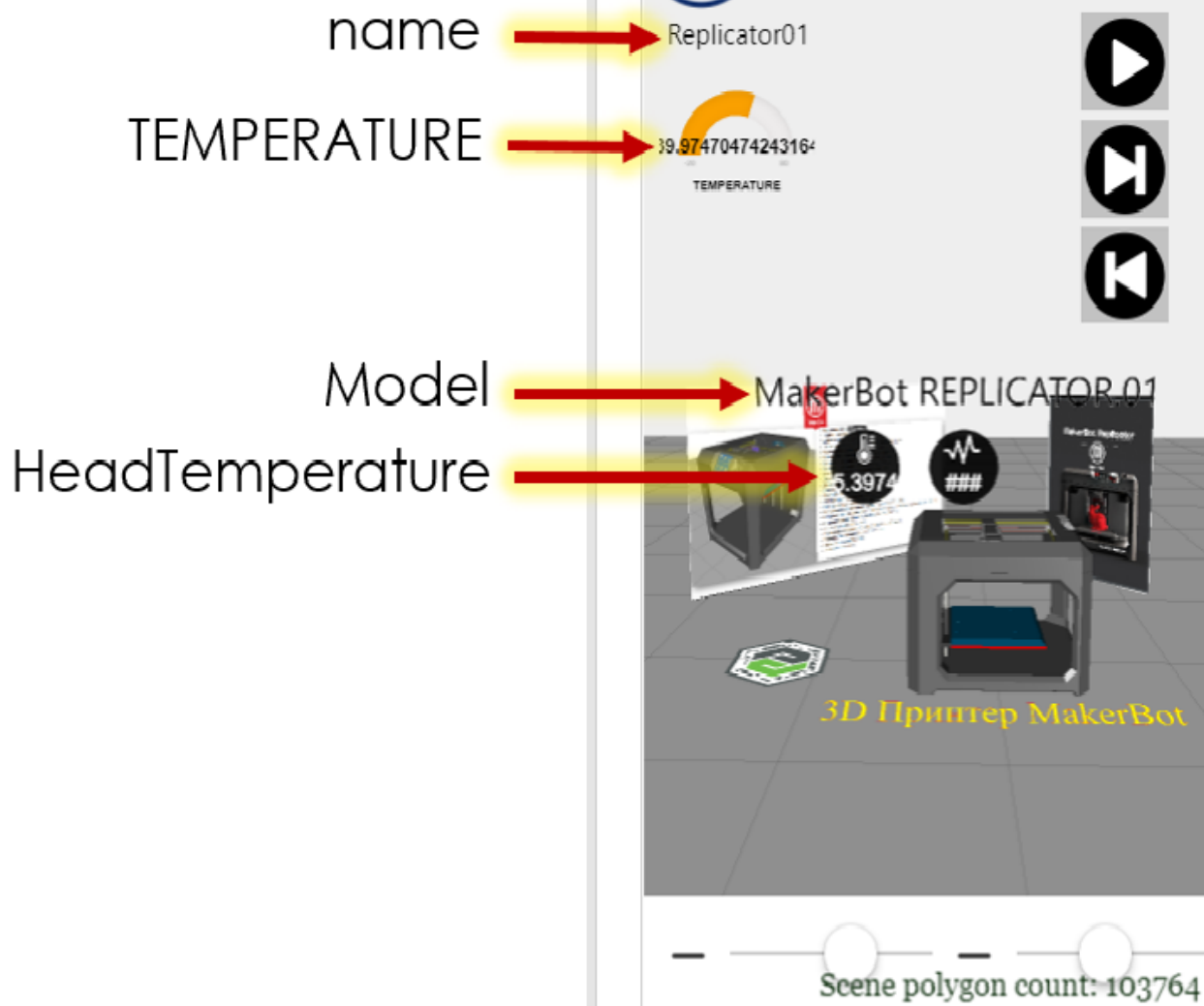


Шаг 21.

Промежуточный результат связывания **2D** - и **3D** – Виджетов со свойствами ЦД целевого объекта (**3D Принтера**) с сервера **IoT/IIoT** в Проекте ДР можно наблюдать в **Vuforia Studio** в режиме **Preview**.

Очевидно, что результат не может быть назван удовлетворительным, т.к. численные значения внешней температуры и температуры головки рабочего инструмента (Экструдера) выводятся с избыточным количеством разрядов, что не несет никакой значимой информации. Более того, важная информация – значимые разряды значений, могут выйти за пределы отображаемой разрядной сетки, как это произошло в данном примере.

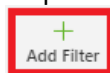
Для преодоления этой коллизии вручную пропишем дополнительный фильтр для исключения избыточной разрядности.



Шаг 22.

Начнем с **2D** канвы, виджета **2D Gauge**. В поле связей (**Binding**), для выбранного

виджета добавляем фильтр



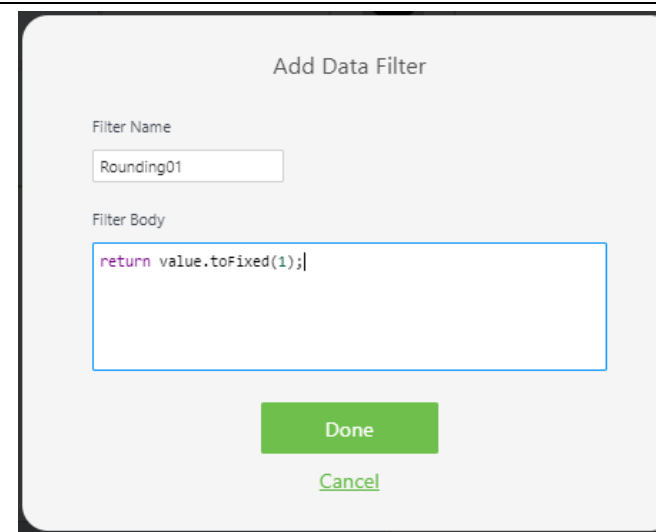
The screenshot shows the Vuforia Studio interface with the 'My Projects > AR-001' project selected. The 'WIDGETS' panel on the left lists various components, with '2D Gauge' selected under the '2D Overlay' category. The 'CANVAS' area displays a 2D layout with a 'Label' and a '2D Gauge' widget. The 'GAUGE - DETAILS' panel on the right shows the properties for the selected gauge, including 'Class', 'Minimum Value', 'Maximum Value', 'Value', 'Decimal Scale', 'Gauge Title', 'Title Position', and 'Gauge Span'. The 'BINDINGS' panel at the bottom is highlighted with a red box, showing a binding configuration for the 'gauge-4' widget. The binding is defined by the 'Source' 'app.mdl[Replicator01].properties[TEMPERATURE]' and the 'Target' 'Widget: gauge-4 Property: value'. An 'Add Filter' button is visible next to the binding.

Source	Target
Binding Expression: app.mdl[Replicator01].properties[TEMPERATURE]	Widget: gauge-4 Property: value

Шаг 23.

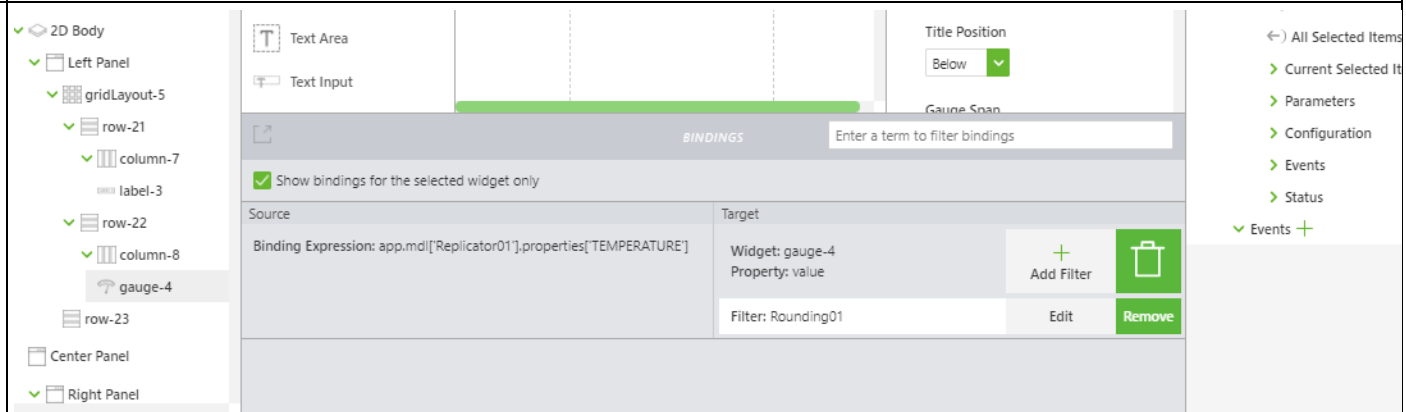
В открывшемся окне фильтра, задаем его имя (**Rounding01**) и в теле прописываем выражение **Java Script**:

```
return value.toFixed(1);
```



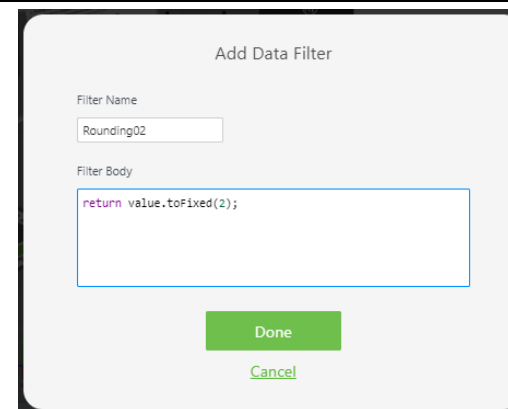
Шаг 24.

Контролируем создание фильтра, наблюдая изменение содержания поля связывания (**Binding**) для выбранного **Widget**'а.



Шаг 25.

Переходим в **3D** Канву и производим аналогичные действия по установке фильтра (задаем ему имя – **Rounding02**) для исключения избыточной разрядности температуры головки Экструдера. Для нас значимыми являются максимум 2 десятичных разряда после запятой.



Шаг 26.

Проверим результат в режиме **Preview**.



Шаг 27.

Теперь Проект можно опубликовать известным вам способом (см. ЛР №3) и проверить работоспособность получившегося упрощенного AR Приложения с помощью **Vuforia View** на вашем мобильном устройстве. Не забудьте, что данное Приложение ДР использует способ таргетирования – **ThingMark** для вашего аккаунта!





3. Практическое задание

В данной Лабораторной работе № 4 необходимо модифицировать ранее разработанный **AR**-Проект, использующий способ таргетирования **ThingMark**.

Средствами **Vuforia Studio** требуется обеспечить связь с сервером **IoT/IIoT - TWx Fndtn**;

На сервере Администратором (*) заведены **Цифровые Двойники (Things)** с именами, указанными в таблице:

Таблица1.

№	Студент-Аккаунт	Имя ЦД (Thing)	3D Label	3D Gauge	2D Label	2D Gauge
1	User01	Replicator01	name	HeaterTemperature	Price	TEMPERATURE
2	User02	Replicator02	Model	PLA_Filament	name	PRESSURE
3	User03	Replicator03	Price	Model3DLoad	PLA_Filament	HUMIDITY
4	User04	Replicator04	name	Price	Model	PRESSURE
5	User05	Replicator05	Model	Model3DLoad	name	TEMPERATURE
6	User06	Replicator06	Price	PLA_Filament	Model	PRESSURE
7	User07	Replicator07	name	HeaterTemperature	Model	HUMIDITY
8	User08	Replicator08	Model	Price	PLA_Filament	PRESSURE
9	User09	Replicator09	Price	HeaterTemperature	name	HUMIDITY
10	Useer10	Replicator10	name	Model3DLoad	Price	PRESSURE

Для выполнения задания на **ЛР №4** из таблицы **Таб.1** в соответствии с Аккаунтом обучающегося (разосланы для предыдущих частей ЛР№4) выбираются :

- имя **Thing** (цифрового двойника) в **ThingWorx Foundation**;
- свойства цифрового двойника для **2D** – виджетов;
- свойства цифрового двойника для **3D** – виджетов.

*) В случае, если физическое устройство отсутствует, если речь идет о разработке учебного или демонстрационного **AR-Приложения (Experience)**, работоспособность полученного решения проверяется с использованием **аппаратного или программного эмулятора устройства**. Данный эмулятор транслирует значения типового набора свойств **3D-Принтер MakerBot Replicator 2 (умного подключаемого устройства)** в состав свойств соответствующего **ЦД** на сервере – **Thingworx Foundation** - сервере интернета вещей. **Администратор TWx Fndtn** может иметь прямой доступ к данным в случае необходимости, или в случае отказа эмулятора. Проверить работоспособность опубликованных Приложений возможно только при включенном и корректно работающем сервере **Thingworx Foundation**.