



TouchGFX

Создание проекта в TouchGFX Designer



Программы для лабораторной работы

- STM32 CubeMX 6.2.1
- H7 Cube Firmware package 1.9.0
- X-CUBE-TOUCHGFX 4.17
- STM32 CubeIDE 1.6.1
- STM32 Cube Programmer 2.6.0



STM32H7B3I-DK Discovery kit

- STM32H7B
 - 1.4 Мбайт RAM
 - 2 Мбайт flash
 - **DPI интерфейс** (LTDC)
- Дисплей
 - **DPI интерфейс**
 - Сенсорная панель
 - RGB888 (24-бит) и RGB565 (16-бит)
 - 480x272 WQVGA разрешение
- 16 Мбайт SDRAM



Этап 1: Макет UI



Этап 1 - цели

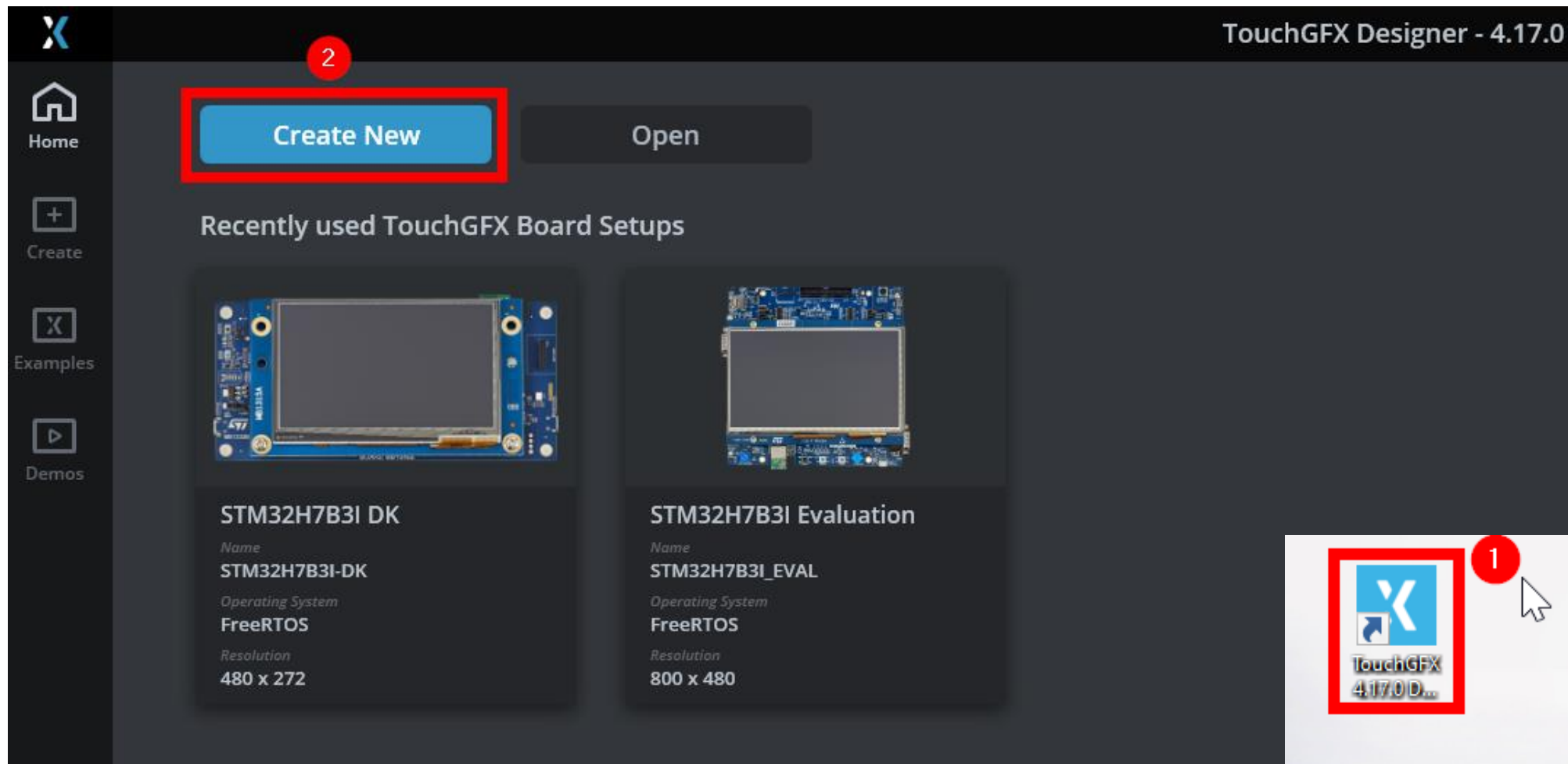
- Реализация базового GUI
- Виджеты
 - Изображение
 - Анимированное изображение





Начало работы в подготовленном проекте

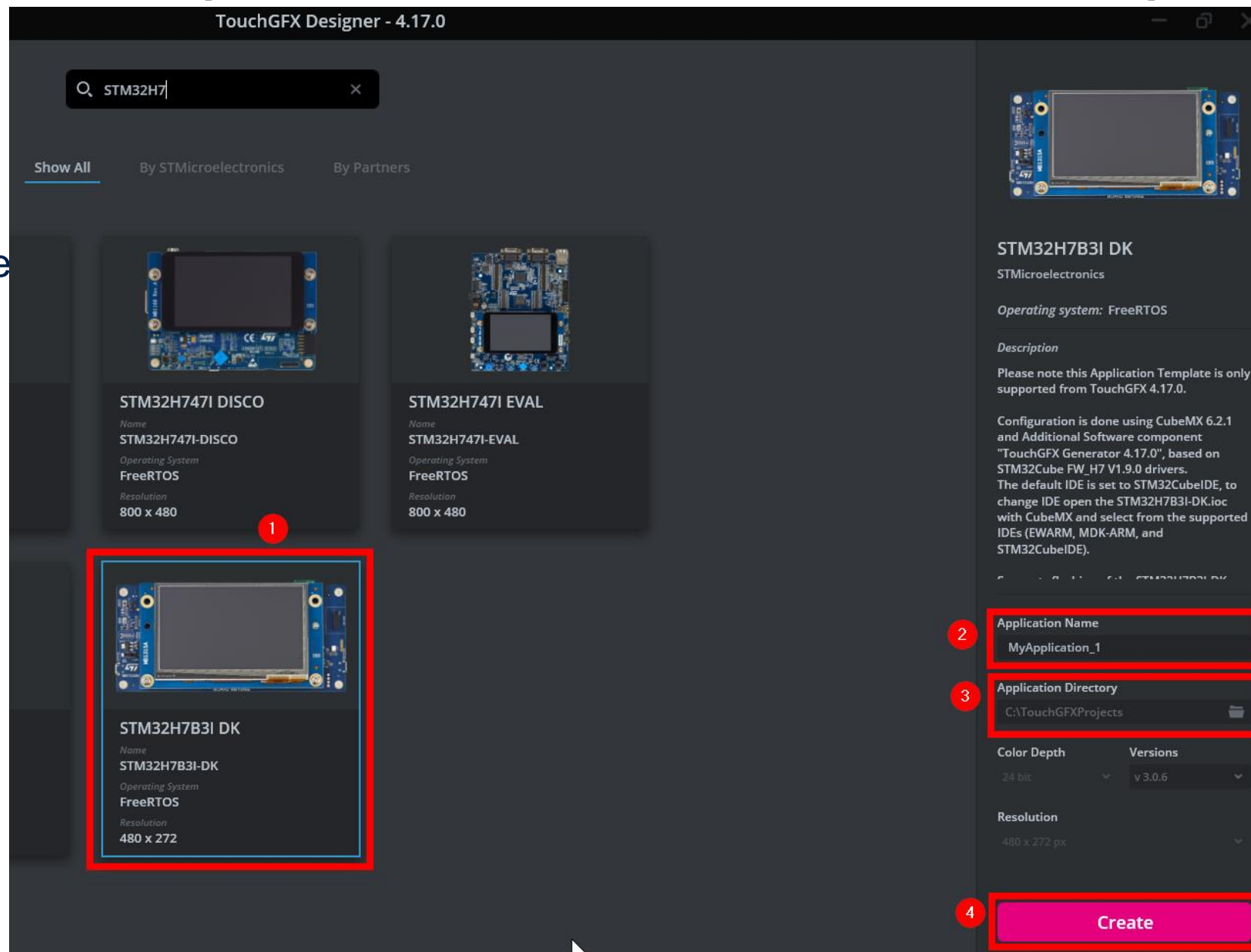
1. Запустите TouchGFX 4.17
2. Создайте новый проект





Начало работы в подготовленном проекте

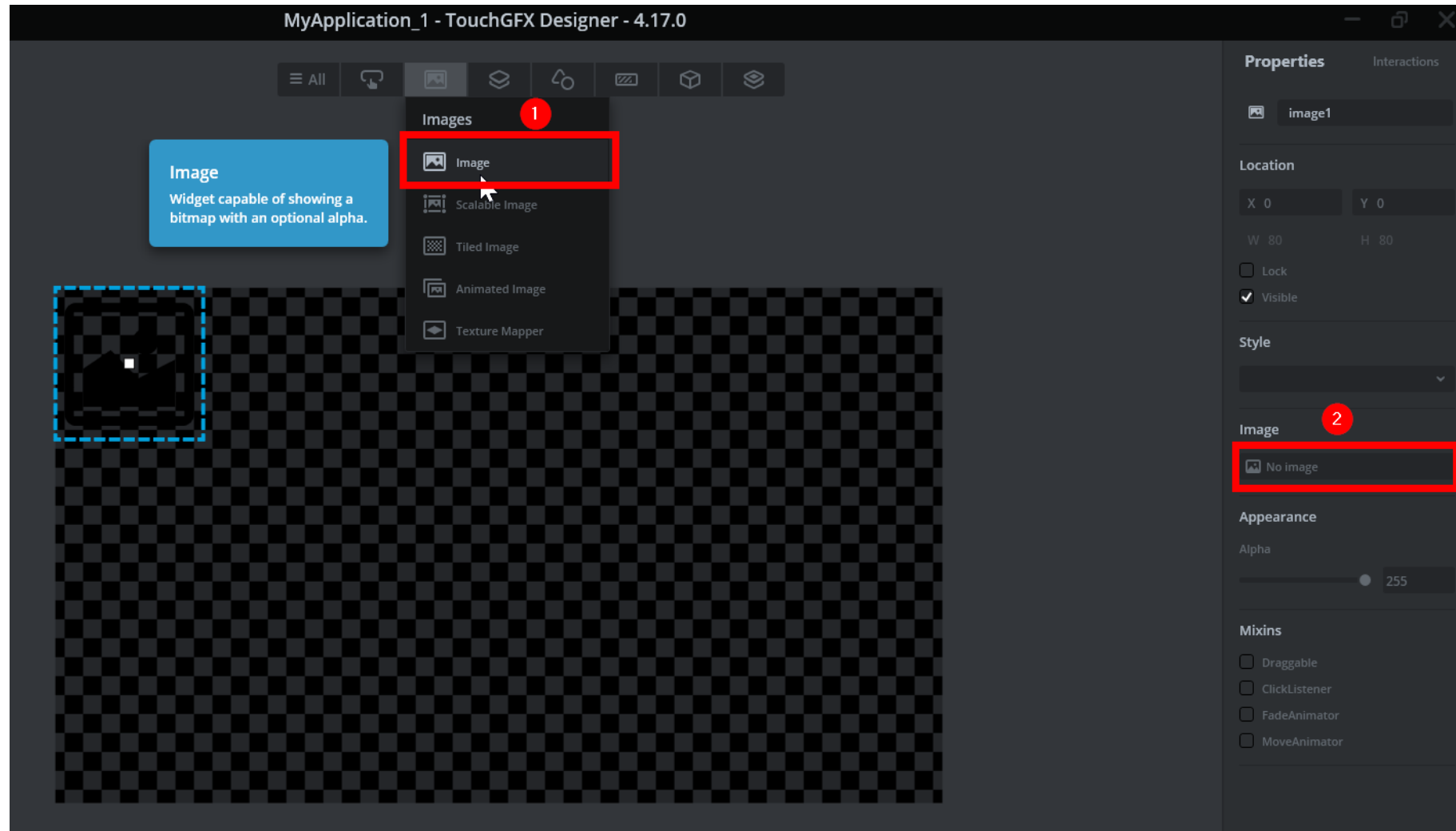
1. Выберите отладочную плату STM32H7B3I-DK
2. Выберите название проекта и расположение проекта
3. Нажмите кнопку "Create"





Добавление элементов GUI: фон

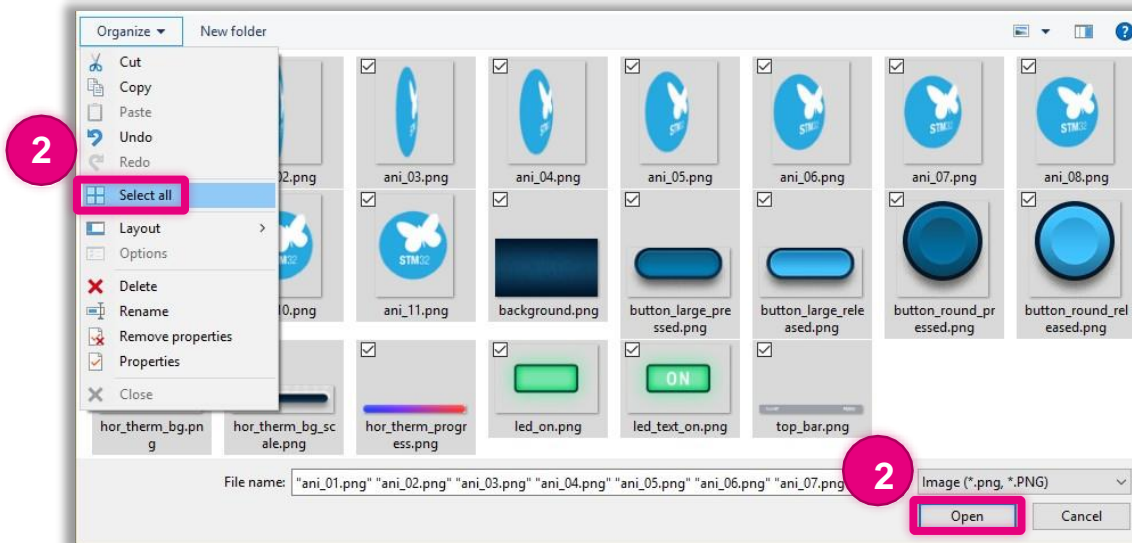
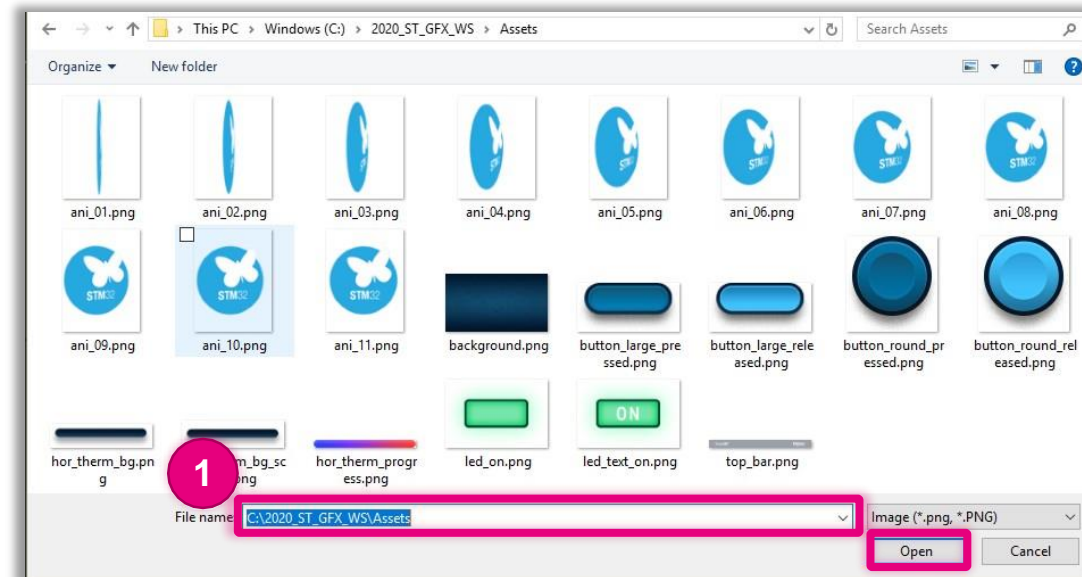
1. Выберите виджет “Image”
2. Справа, нажмите на появившуюся строку “Image”
3. Нажмите “+” для добавления изображений



Добавление элементов GUI: фон

В диалоговом окне выбора файлов:

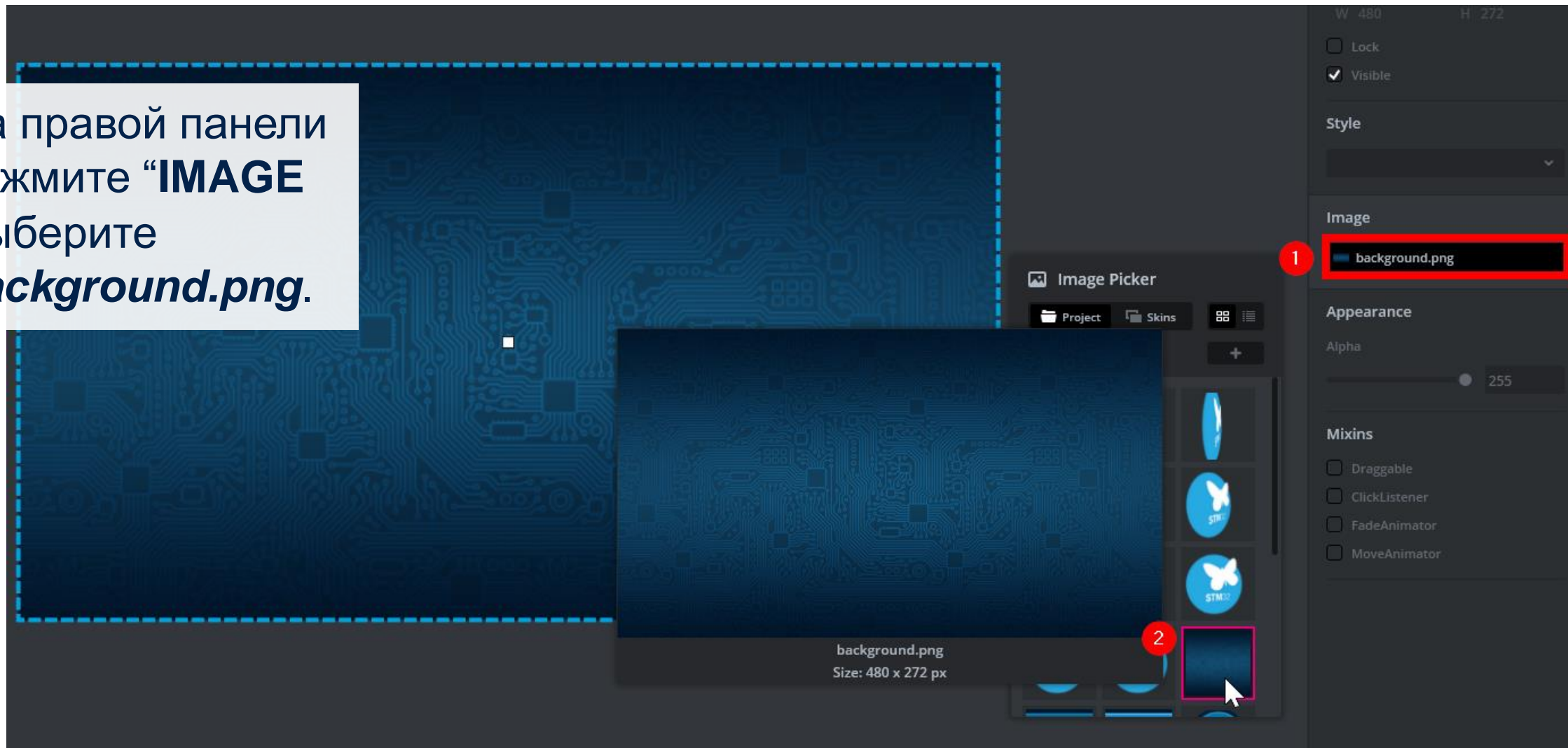
- 1 Выберите папку с изображениями
... \Assets
нажмите "Open"
- 2 Выберите все файлы
(*Ctrl+A* или "*Organize->Select all*"),
далее нажмите "Open"





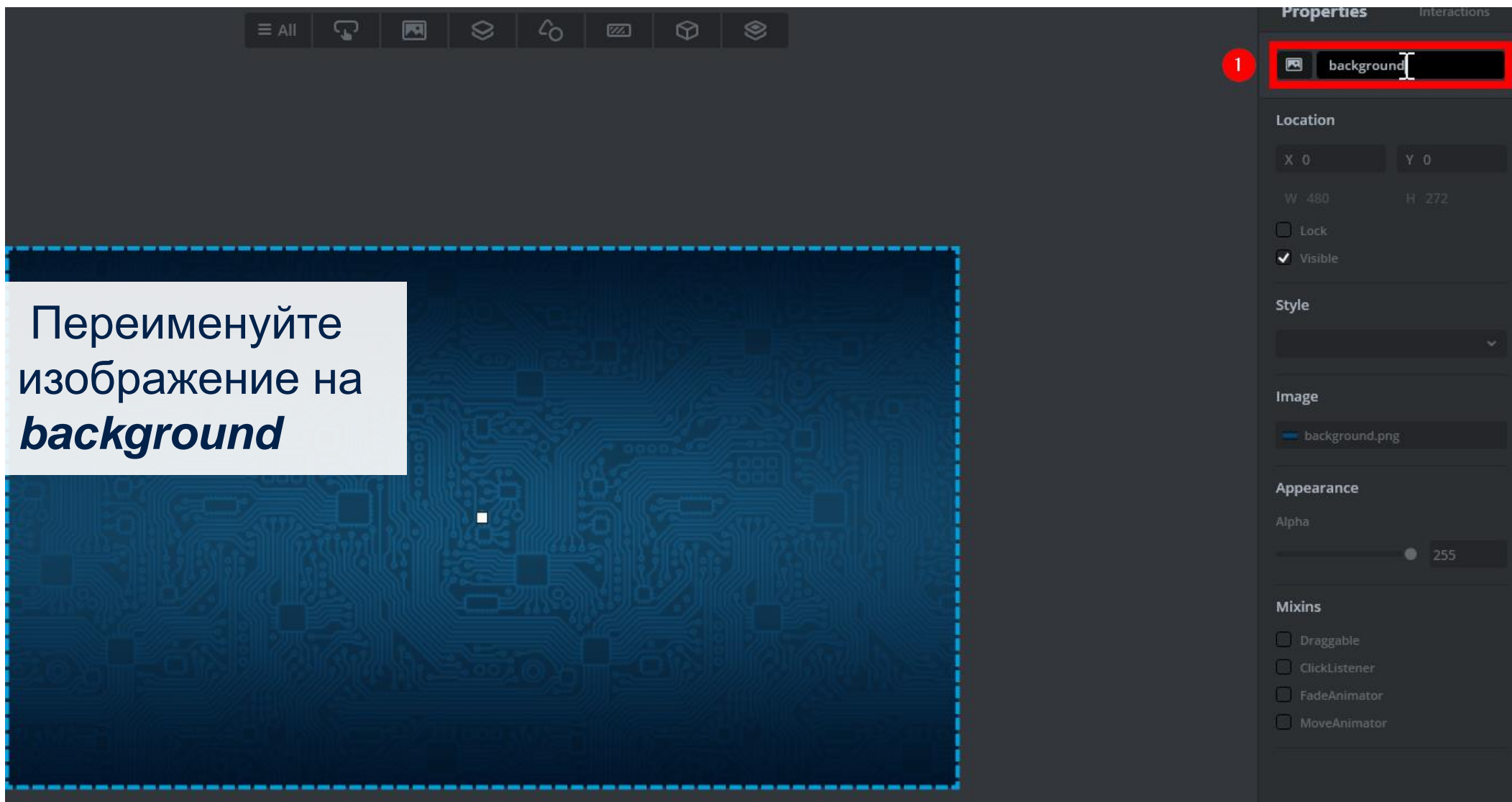
Добавление элементов GUI: фон

1. На правой панели нажмите “**IMAGE**”
2. Выберите *background.png*.



Добавление элементов GUI: фон

1. Переименуйте изображение на ***background***



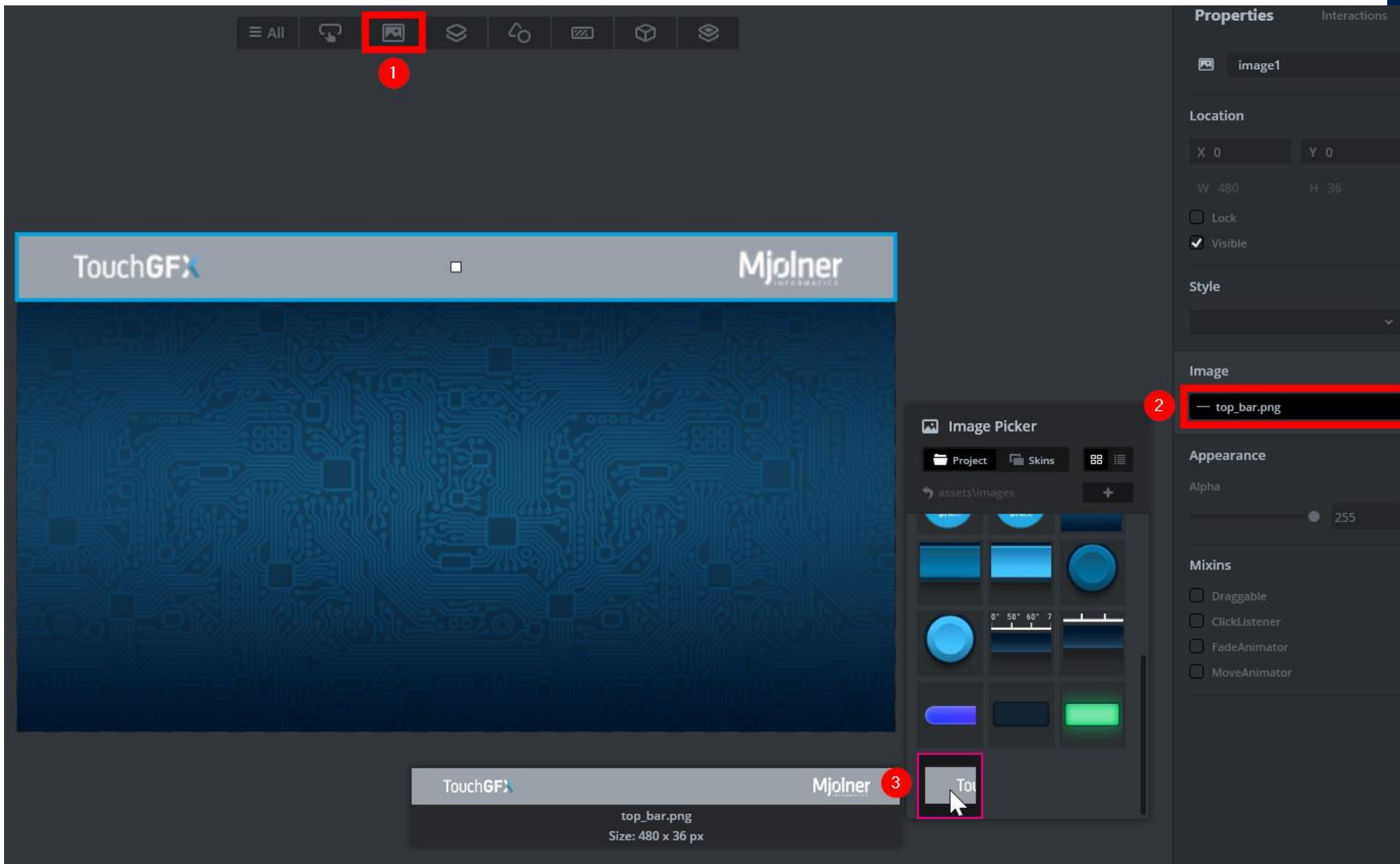


Добавление элементов GUI: Верхняя панель

1. Нажмите на **Image**

2. На правой панели нажмите на **IMAGE**

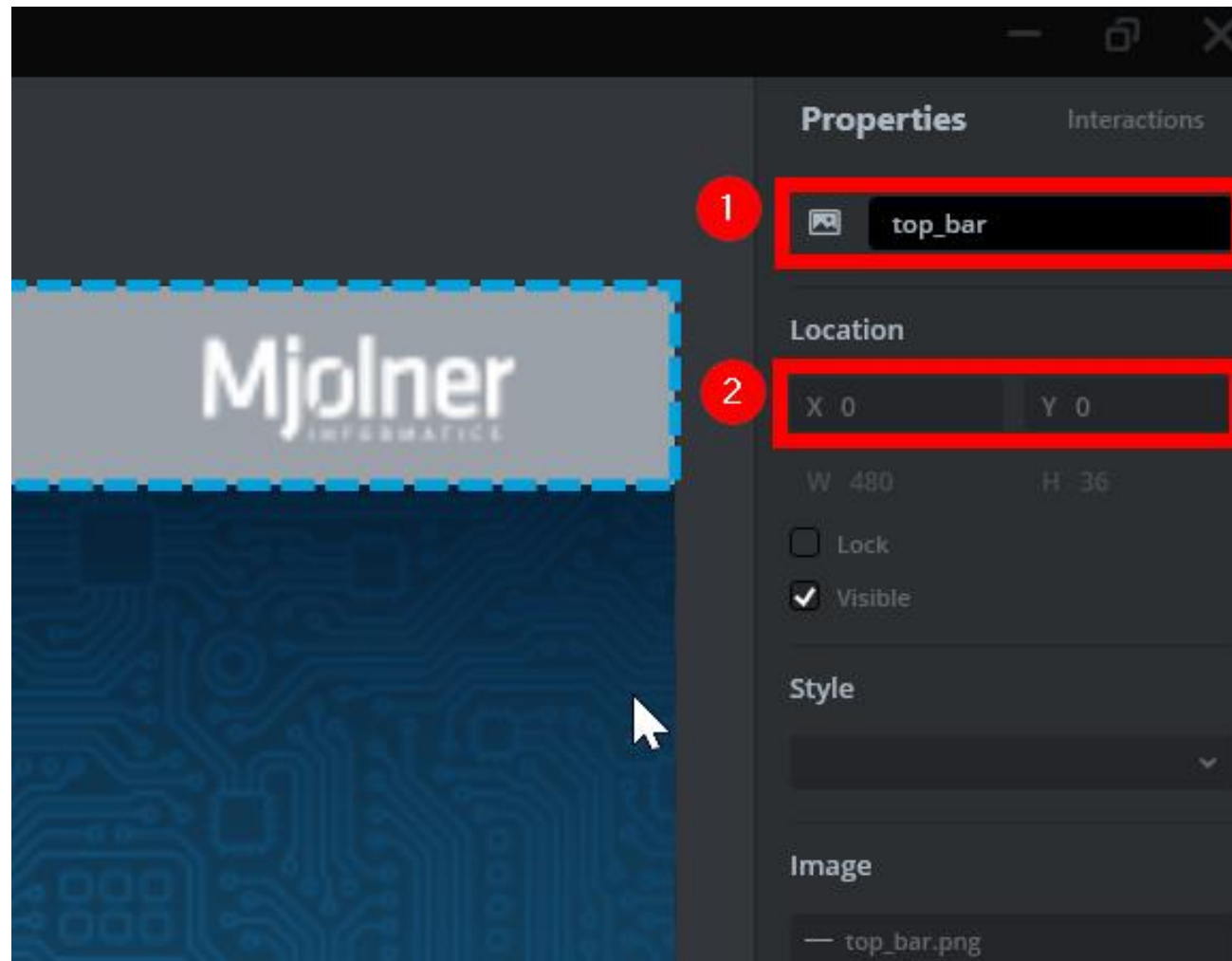
3. Выберите **top_bar.png** из списка





Добавление элементов GUI: Верхняя панель

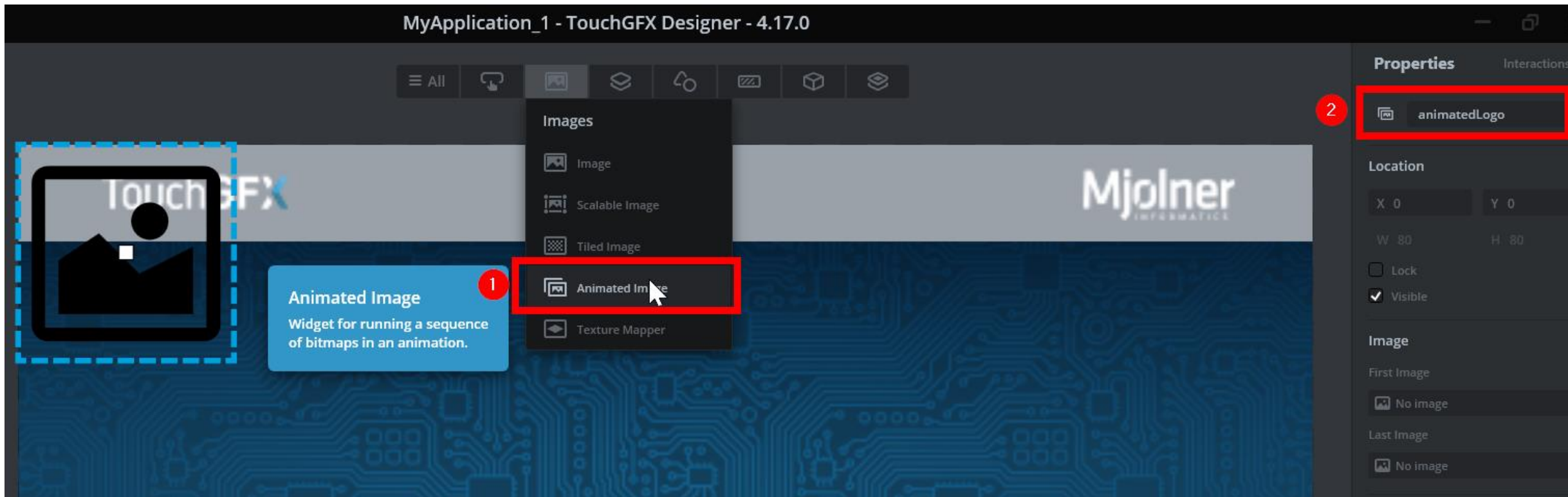
1. Переименуйте изображение в ***top_bar***
2. На правой панели можно изменить положение виджета [0, 0]





Добавление элементов GUI: анимированное изображение

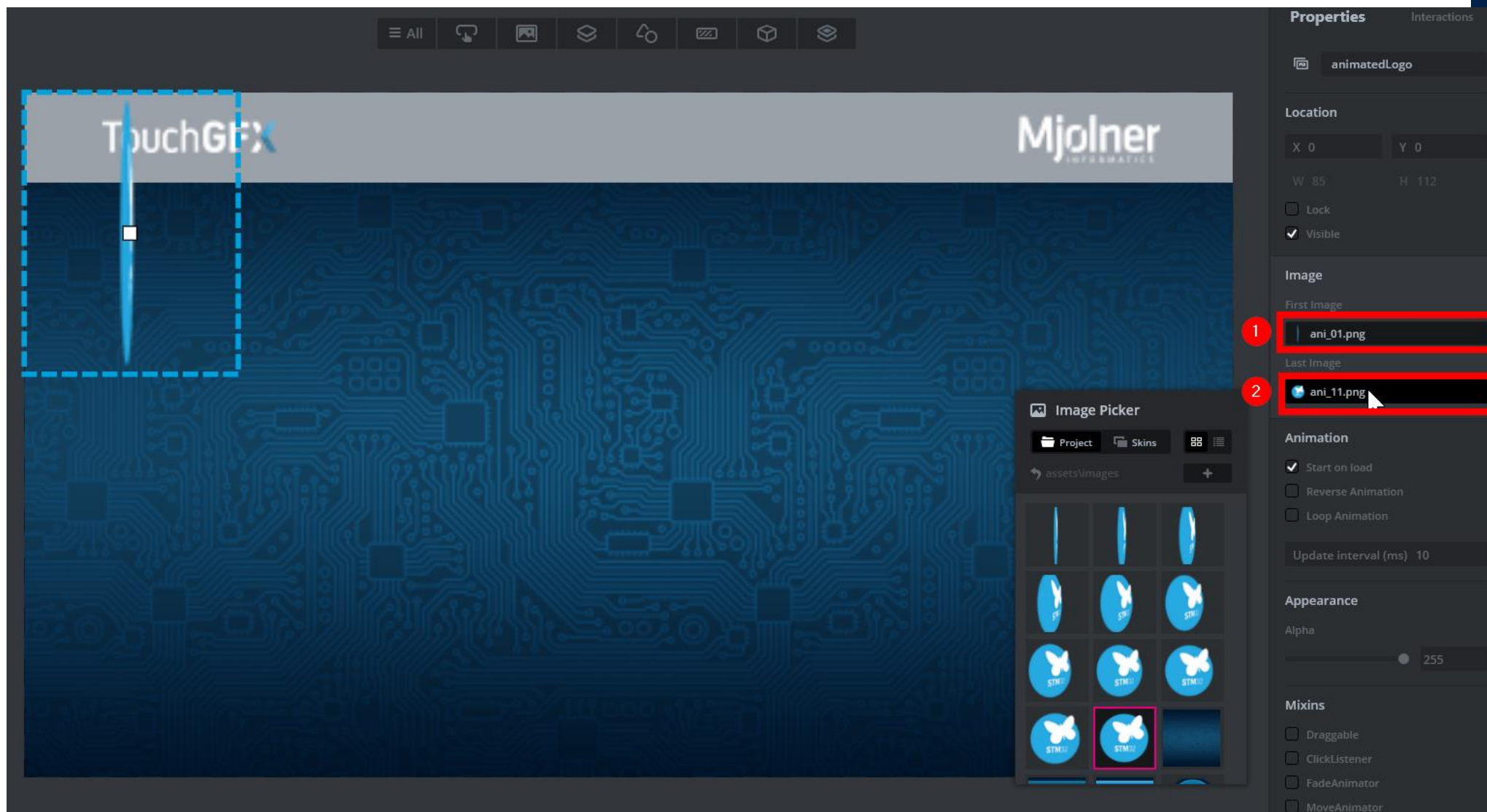
1. Выберите виджет **Animated Image**
2. Переименуйте виджет на *animatedLogo*





Добавление элементов GUI: анимированное изображение

1. Установите на **First Image** *ani_01.png*
2. Установите на **Last Image** *ani_11.png*





Добавление элементов GUI: анимированное изображение

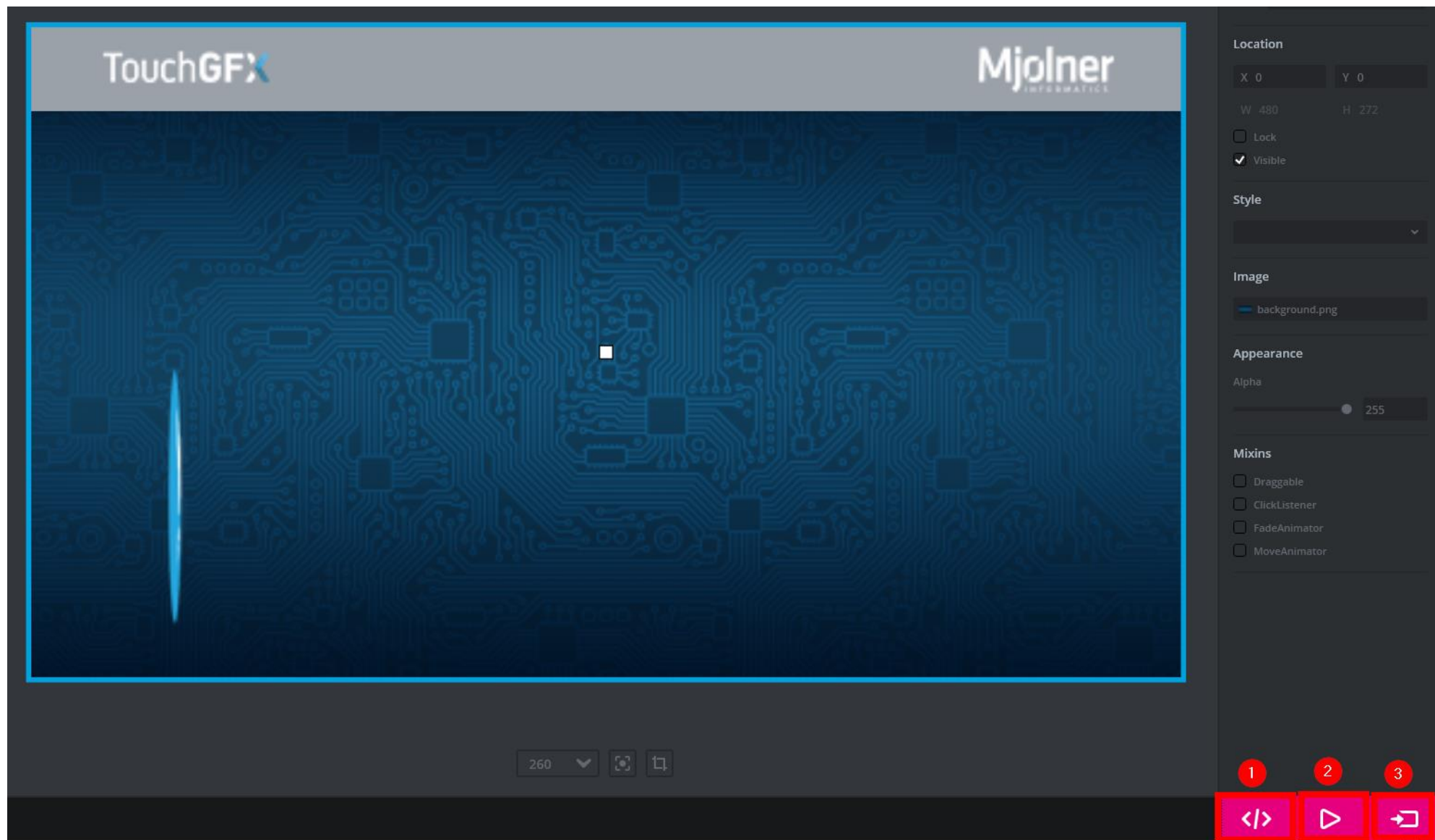
1. Установите **Location** на $X = 20$ и $Y = 140$
2. Активируйте **loop animation**
3. Установите **Update interval** на 100ms

The screenshot displays the TouchGFX IDE interface. The main canvas shows a blue circuit board pattern with a dashed blue rectangle indicating the position of an animated logo. The right-hand 'Properties' panel is open, showing the configuration for the 'animatedLogo' element. The 'Location' section is highlighted with a red box and a red circle '1', showing 'X' set to 20 and 'Y' set to 140. The 'Image' section shows 'First Image' as 'ani_01.png' and 'Last Image' as 'ani_11.png'. The 'Animation' section is highlighted with a red box and a red circle '2', showing 'Start on load' checked, 'Reverse Animation' unchecked, and 'Loop Animation' checked. The 'Update interval (ms)' is set to 100, highlighted with a red box and a red circle '3'. The 'Appearance' section shows 'Alpha' set to 255. The 'Mixins' section at the bottom lists 'Draggable', 'ClickListener', 'FadeAnimator', and 'MoveAnimator', all of which are unchecked.



Сборка и проверка

1. Нажмите кнопку **Generate Code** button
2. Нажмите кнопку **Run Simulator**
3. Подключите отладочную плату и загрузите ПО, нажав на кнопку **Run Target**

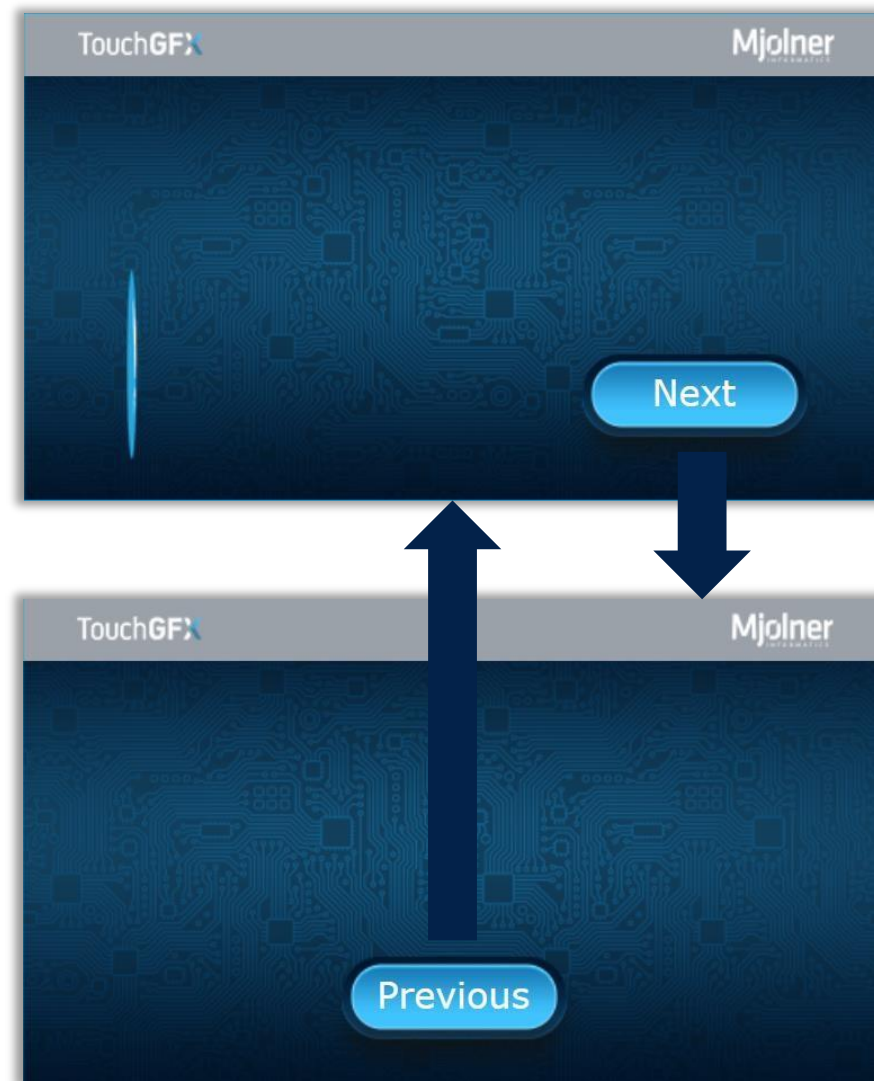


Этап 2: Переключения



- Использование общих виджетов
 - Flex buttons
- Добавление взаимодействий
- Использование нескольких экранов

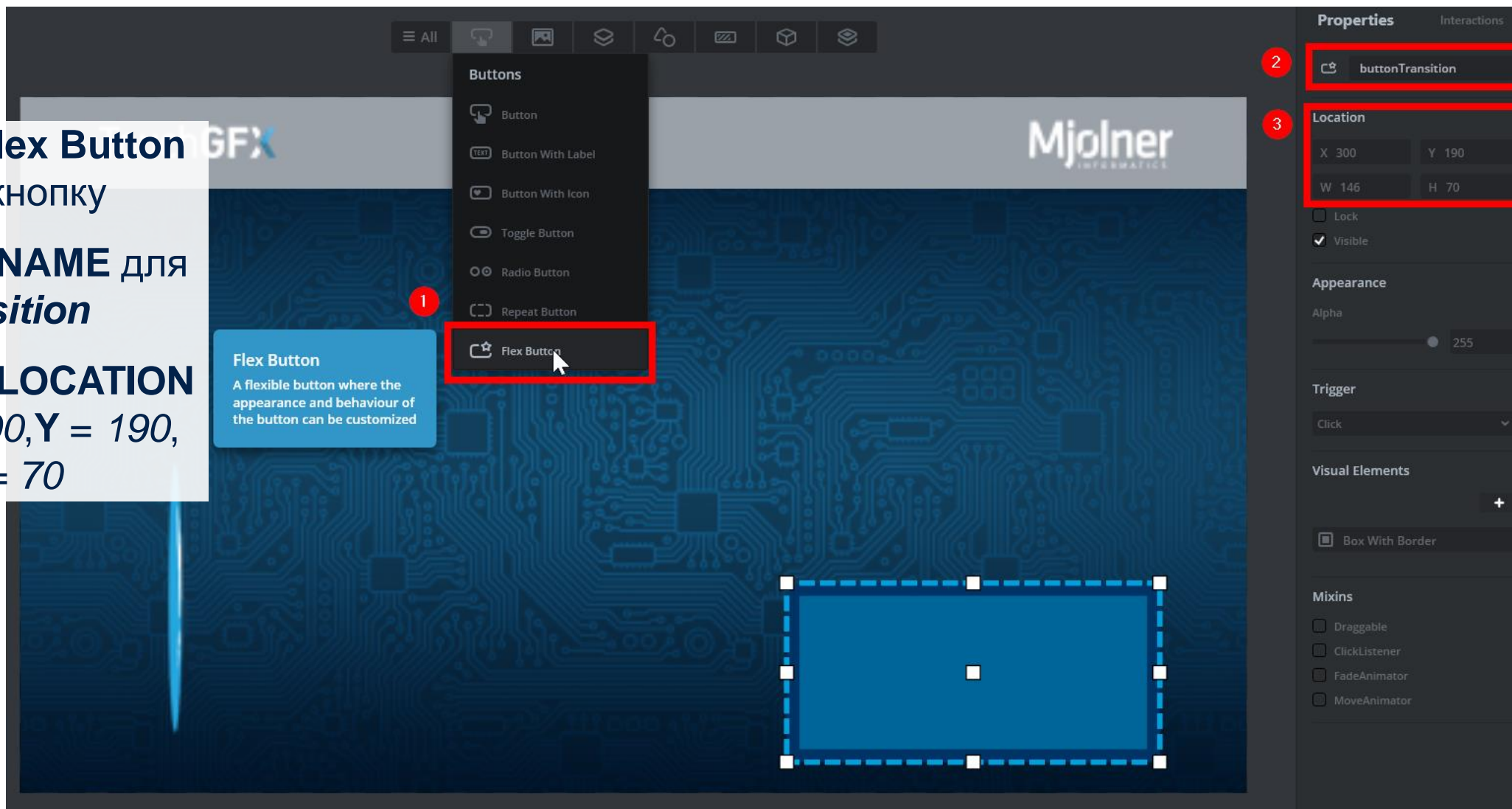
Этап 2 - цели





Добавление элементов GUI: flex button

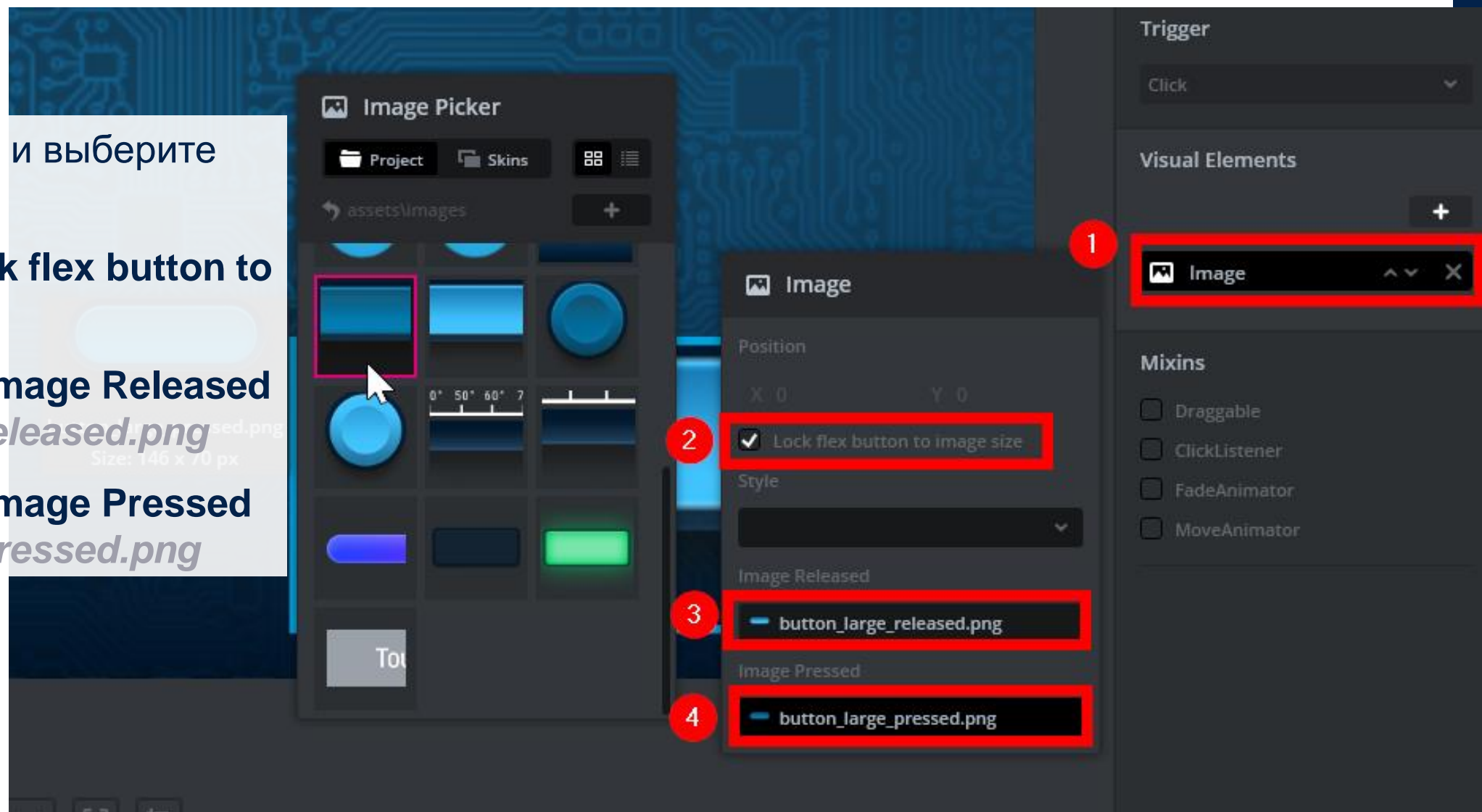
1. Выберите **Flex Button** и добавьте кнопку
2. Установите **NAME** для *buttonTransition*
3. Установите **LOCATION** на $X = 300, Y = 190, W = 146, H = 70$





Добавление элементов GUI: flex button

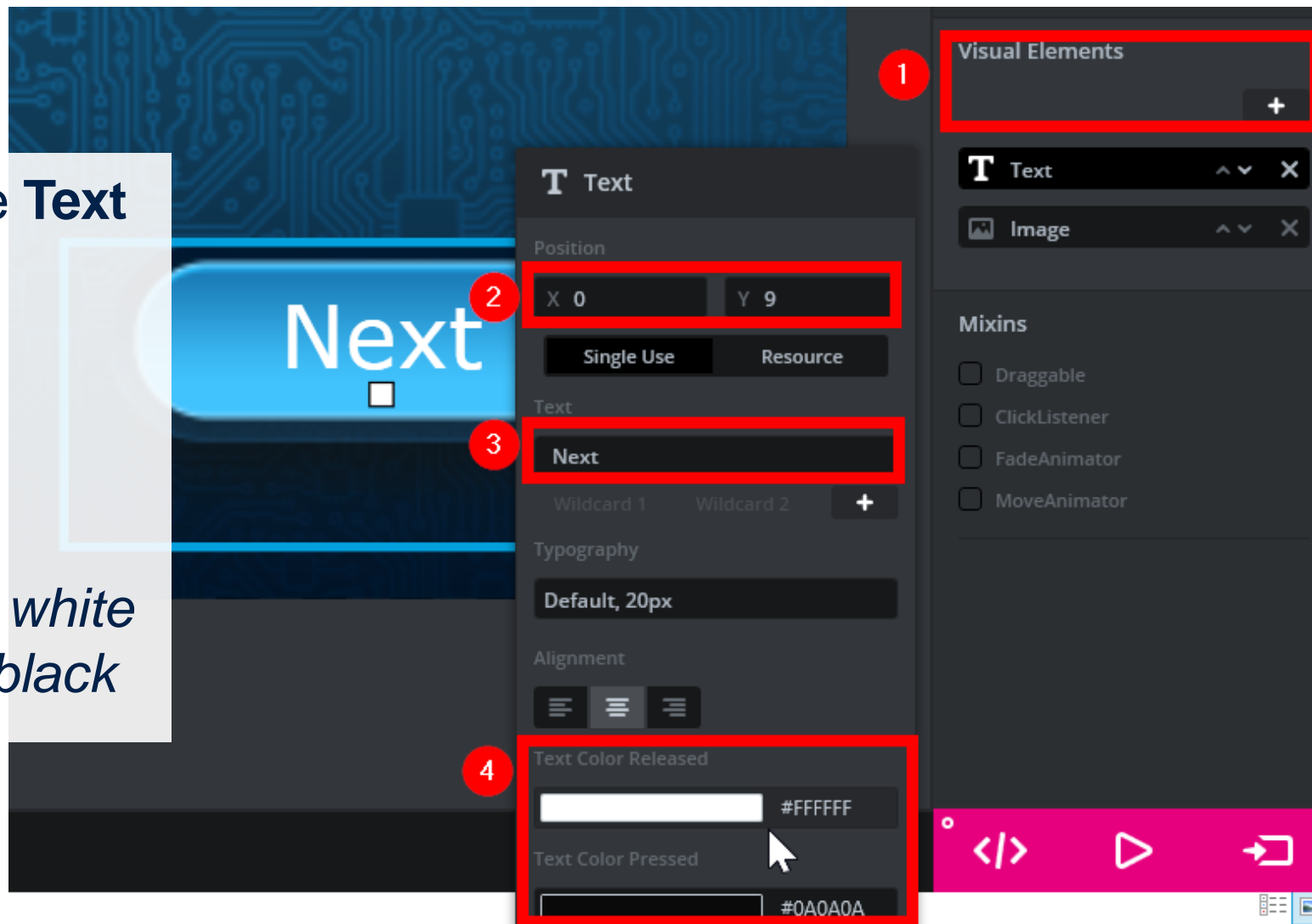
1. Нажмите на “+” и выберите **Image**
2. Установите **Lock flex button to image size**
3. Выберите для **Image Released** *button_large_released.png*
4. Выберите для **Image Pressed** *button_large_pressed.png*





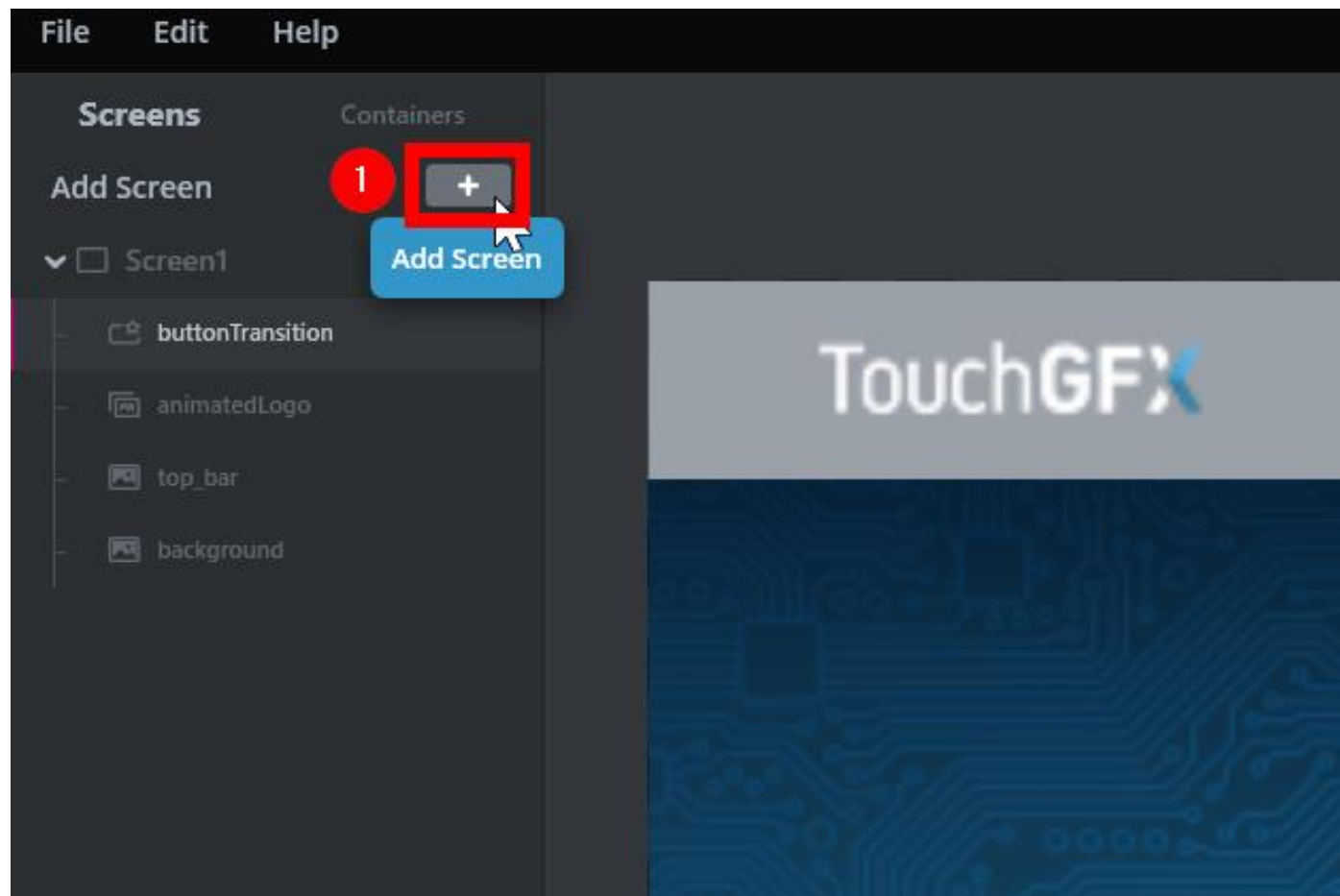
Добавление элементов GUI: flex button

1. Нажмите “+” и выберите **Text**
2. Установите **POSITION**
 $X = 0$, $Y = 9$
3. Измените **Text** на **Next**
4. Установите цвета:
Text color Released to *white*
Text Color Pressed to *black*



Добавление экрана

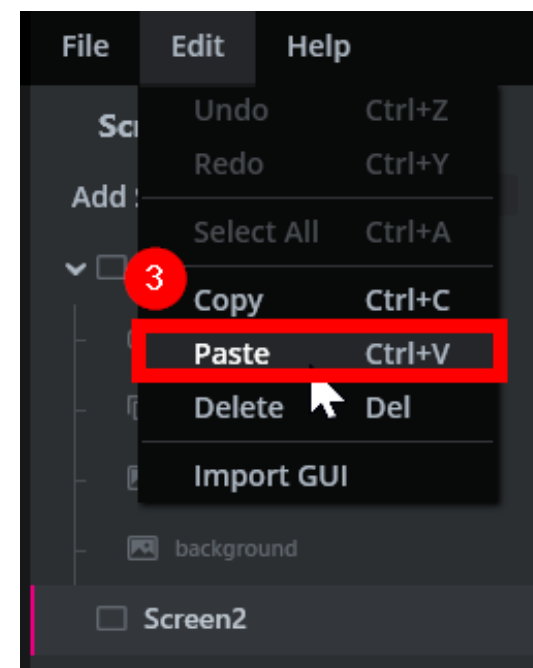
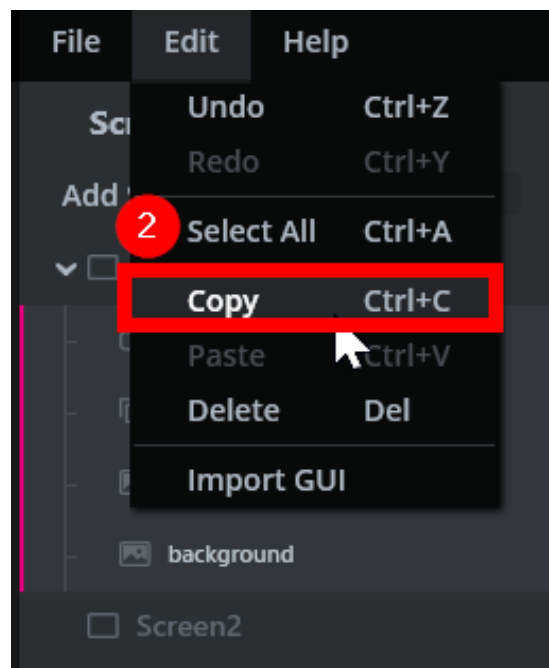
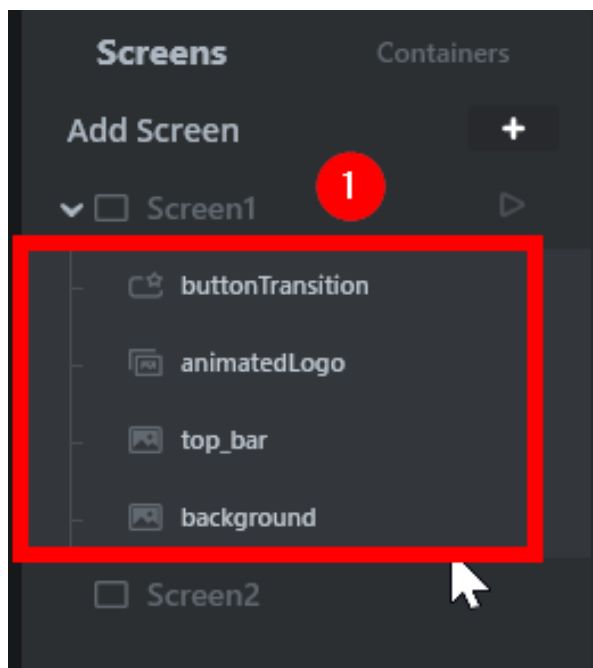
1. Создайте новый экран, нажав на “+”





Добавление элементов GUI: копирование виджетов

1. Нажмите на **buttonTransition** (Screen1) потом удерживая Ctrl выделите виджеты **top_bar** и **background**
2. В меню выберите Edit->Copy или <Ctrl+C> для копии выделенных элементов
3. Нажмите на Screen2 и выберите в меню Edit->Paste или <Ctrl+V>



Переименование кнопки на Screen2



1. Выберите **buttonTransition** на **Screen**

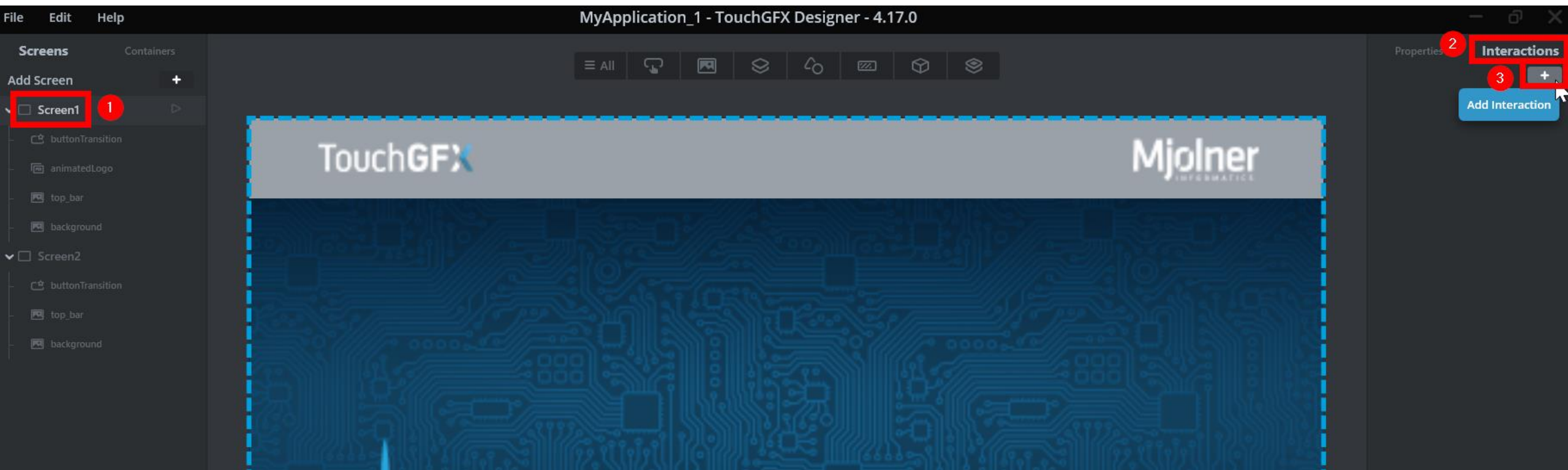
2. Установите **LOCATION** в **X=167, Y=200**

3. Измените **Text** на **Previous**



Создание перехода: с Screen1 на Screen2

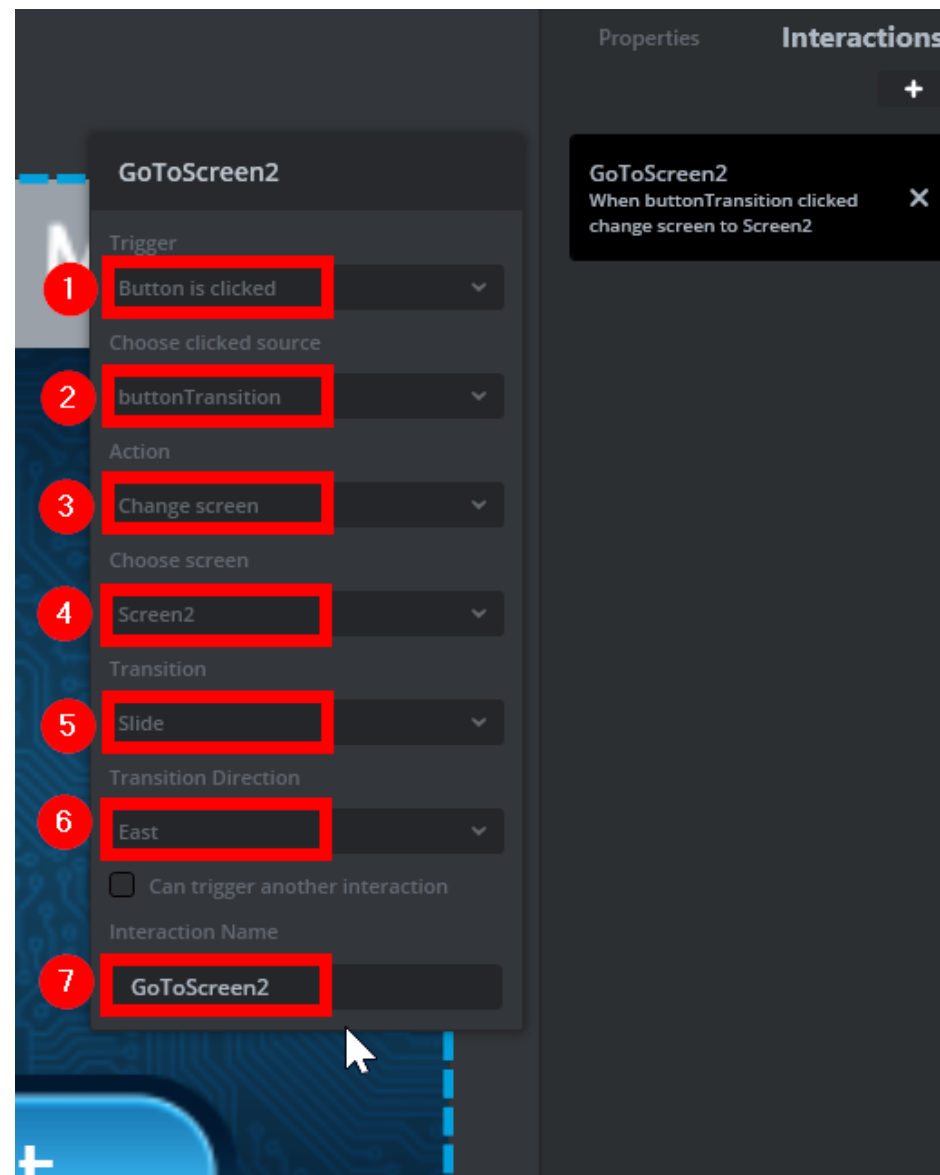
1. Выберите **Screen1**
2. Нажмите на **Interactions**
3. Нажмите на “+” для добавления нового взаимодействия





Настройка перехода: с Screen1 на Screen2

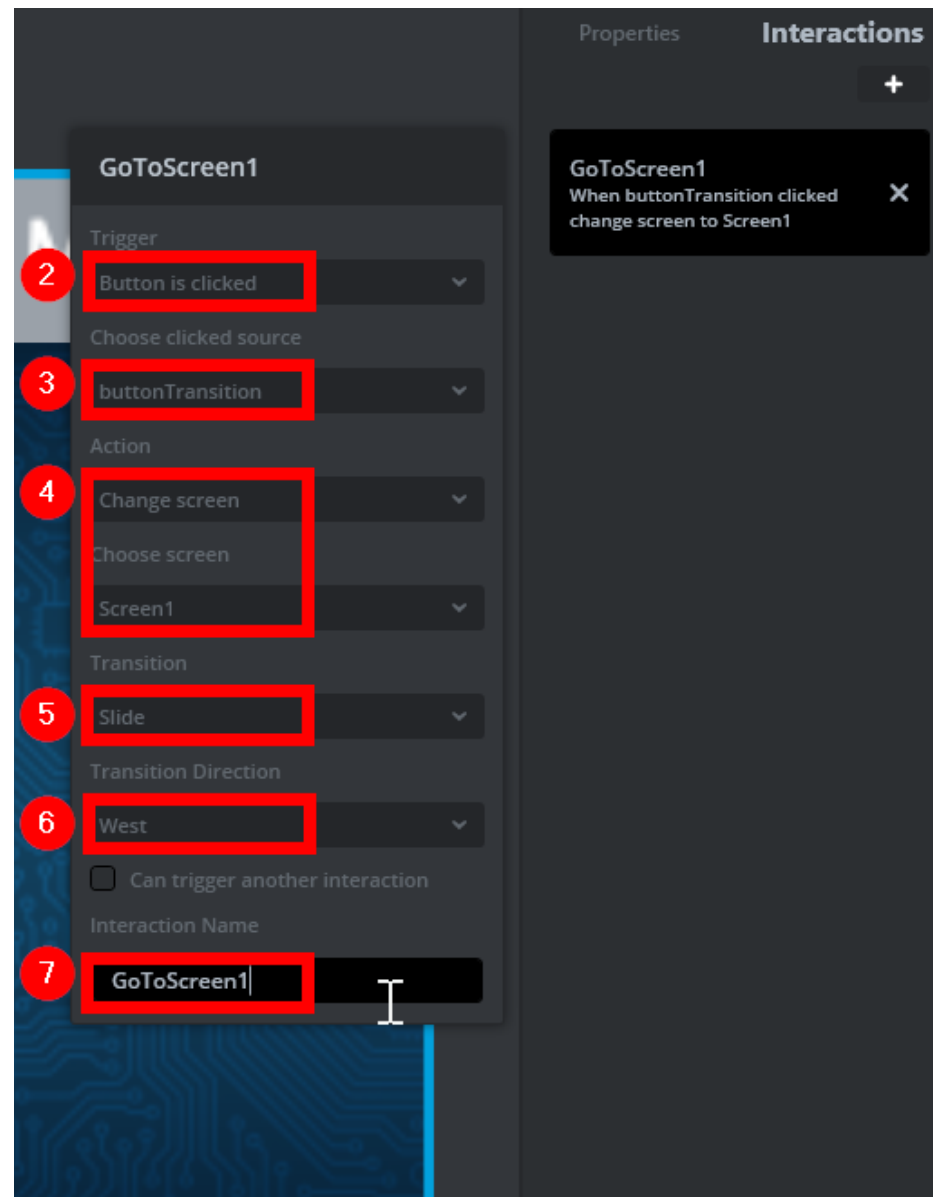
1. Переключите **Trigger** в *Button is clicked*
2. Переключите **Choose clicked source** в *buttonTransition*
3. Переключите **Action** в *Change screen*
4. Переключите **Choose screen** в *Screen2*
5. Переключите **Transition** в *Slide*
6. Переключите **Transition Direction** в *East*
7. Переключите **Interaction Name** в *GoToScreen2*





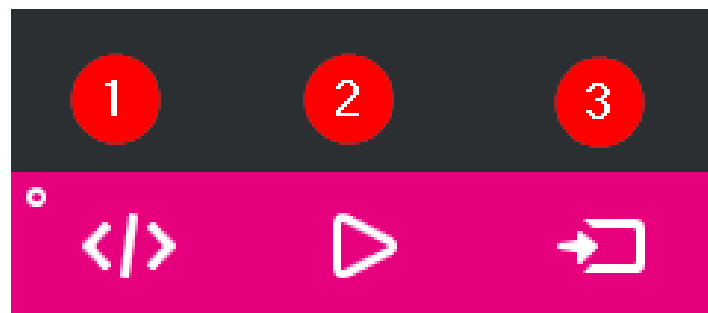
Настройка перехода: с Screen2 на Screen1

1. На экране screen2 нажмите “+”
2. Переключите **Trigger** в *Button is clicked*
3. Переключите **Choose clicked source** в *buttonTransition*
4. Переключите **Action** в *Change screen*.
Переключите **Choose screen** в *Screen1*
5. Переключите **Transition** в *Slide*
6. Переключите **Transition Direction** в *West*
7. Переключите **Interaction Name** в *GoToScreen1*





Run and test



1. Нажмите кнопку **Generate Code** button
2. Нажмите кнопку **Run Simulator**
3. Подключите отладочную плату и загрузите ПО, нажав на кнопку **Run Target**

Протестируйте переключение с одного экрана на другой через нажатие кнопки

```
File download complete
Time elapsed during download operation: 00:00:10.431

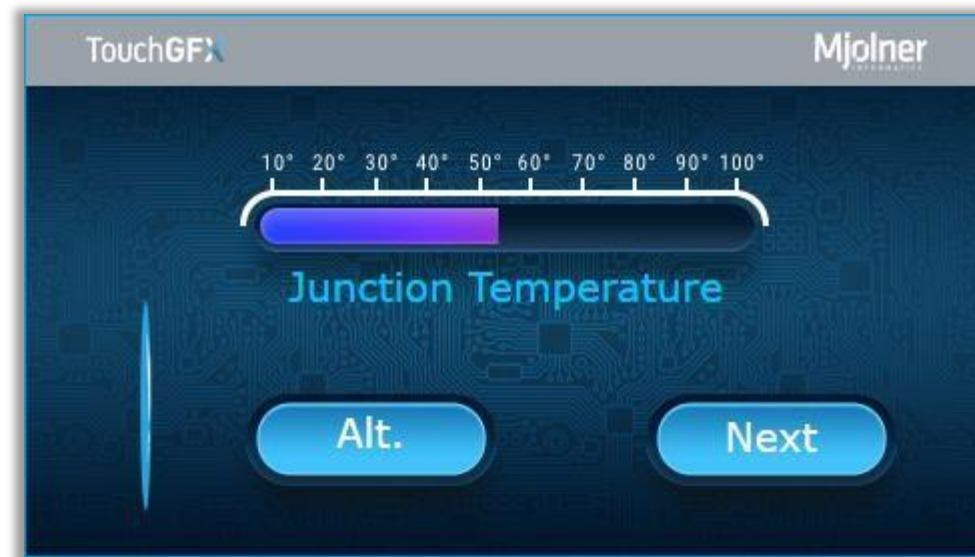
Hard reset is performed
Done
Done
```

Этап 3: взаимодействие с событиями системы: датчик температуры



Этап 3 - цели

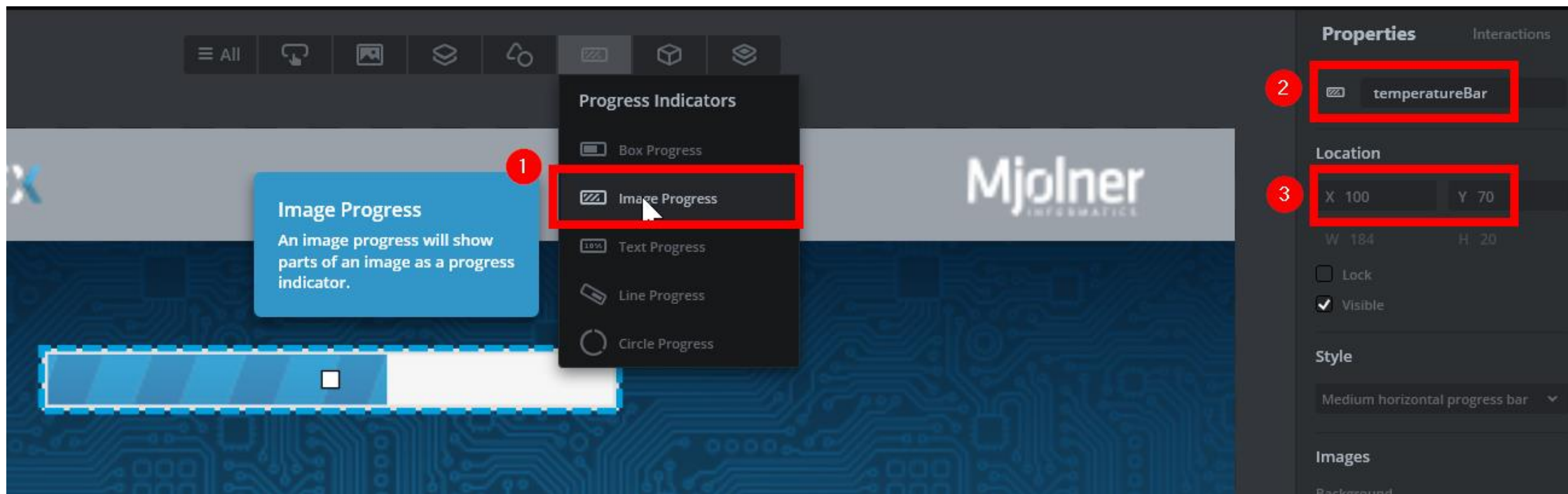
- Использование виджета Progress Image
- Использование CubeMX для конфигурации АЦП и получения внутренней температуры
- Работа с архитектурой model view presenter (MVP) и взаимодействие GUI с системой





Добавление элементов GUI : Image progress

1. Выберите Screen1 и нажмите на виджет **Image Progress**
2. Измените **NAME** на *temperatureBar*
3. Измените **LOCATION** на x=100 и Y = 70





Добавление элементов GUI : Image progress

1. Измените **Style** на **No style**
2. Выберите изображения для шкалы:
background – *hor_therm_bg_scale.png*
progress – *hor_therm_progress.png*





Добавление элементов GUI : Image progress

1. Установите **Progress Position** на **X=18, Y=27, W=244 и H=18**

2. Установите значения **Range** на **Min=0 и Max=100**

3. Установите значение **Initial** на **49**

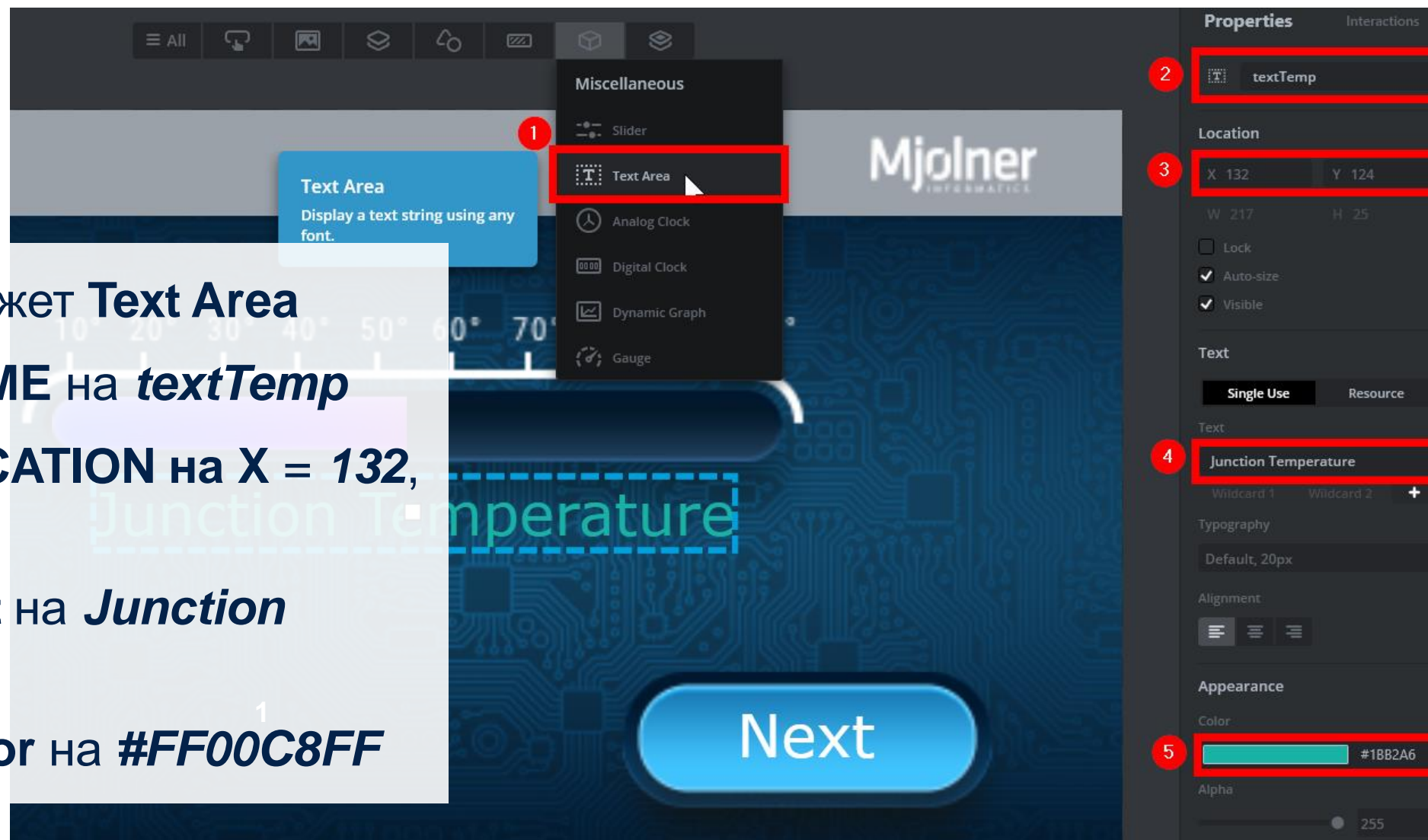
4. Установите **Direction** в **Right**

5. Отметьте – **Anchor progress image at zero**



Добавление надписи под шкалой прогресса

1. Выберите виджет **Text Area**
2. Измените **NAME** на *textTemp*
3. Измените **LOCATION** на **X = 132, Y = 124**
4. Измените **Text** на *Junction Temperature*
5. Измените **Color** на **#FF00C8FF**





Добавление надписи под шкалой прогресса

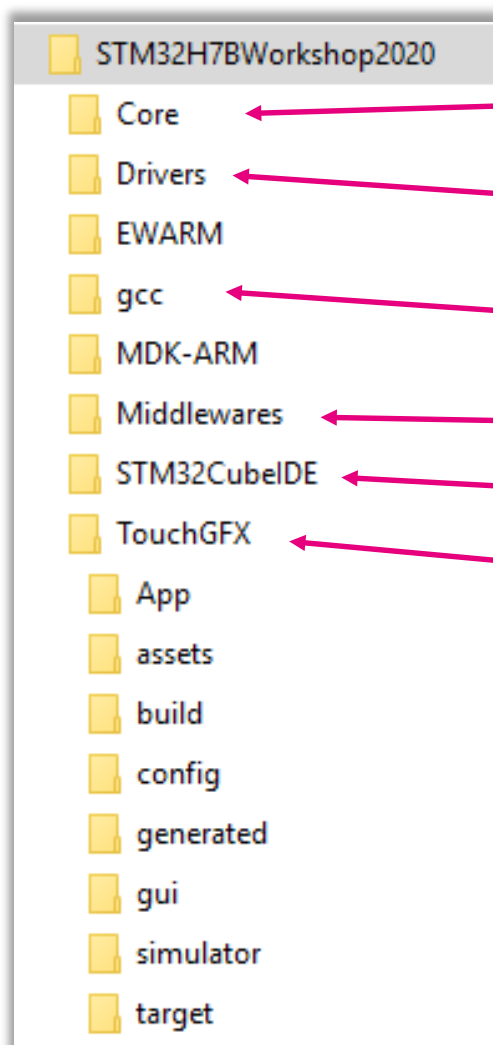
The screenshot displays the TouchGFX IDE interface. The main canvas shows a progress bar with a scale from 10° to 100°. Below the progress bar, the text "Junction Temperature" is highlighted with a dashed blue border, indicating it is being edited or added. A blue "Next" button is visible at the bottom right of the canvas. The right sidebar contains the configuration for the selected text label:

- Location:** X 132, Y 124, W 217, H 25
- ☐ Lock
- ☒ Auto-size
- ☒ Visible
- Text:** Single Use, Resource
- Text:** Junction Temperature
- Wildcard 1:** Wildcard 2 +
- Typography:** Default, 20px
- Alignment:** Left, Center, Right
- Appearance:** Color #1BB2A6, Alpha 255, Line spacing 0, Text rotation 0
- Generate Code:** F4
- ClickListener:** (empty)

At the bottom of the IDE, there is a button labeled "Сгенерируйте код" (Generate code) and a status bar with icons for zooming and other functions.



Сгенерированный код



Application common config
(main.cpp, HAL and FreeRTOS conf)

ST MCU Drivers (BSP, HAL)

gcc toolchain files

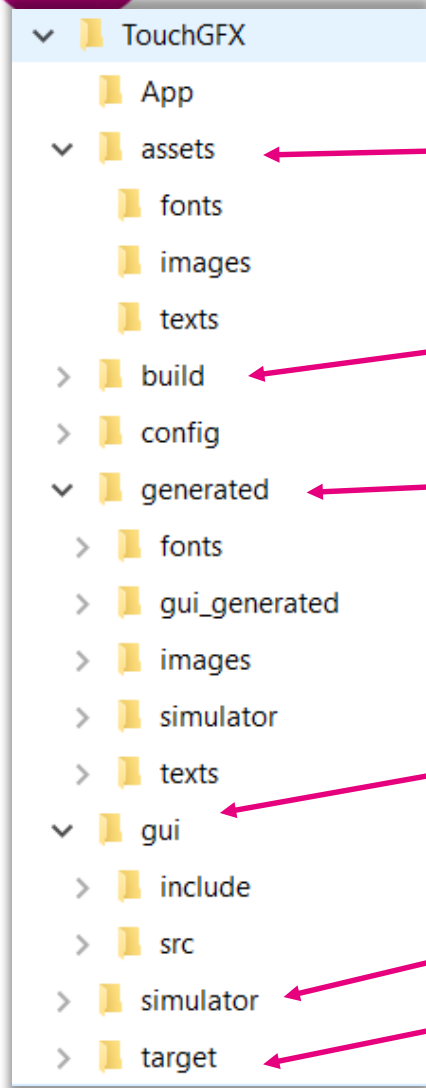
FreeRTOS, TouchGFX framework

STM32CubeIDE configuration files

TouchGFX Designer common files



Код сгенерированный TouchGFX



GUI binary inputs (images, texts, fonts)

Working directory for simulator and target compilation

All the cpp files generated from the assets directory + GUI files that should not be modified (in gui_generated)

GUI files that can be modified to implement interactions and customize

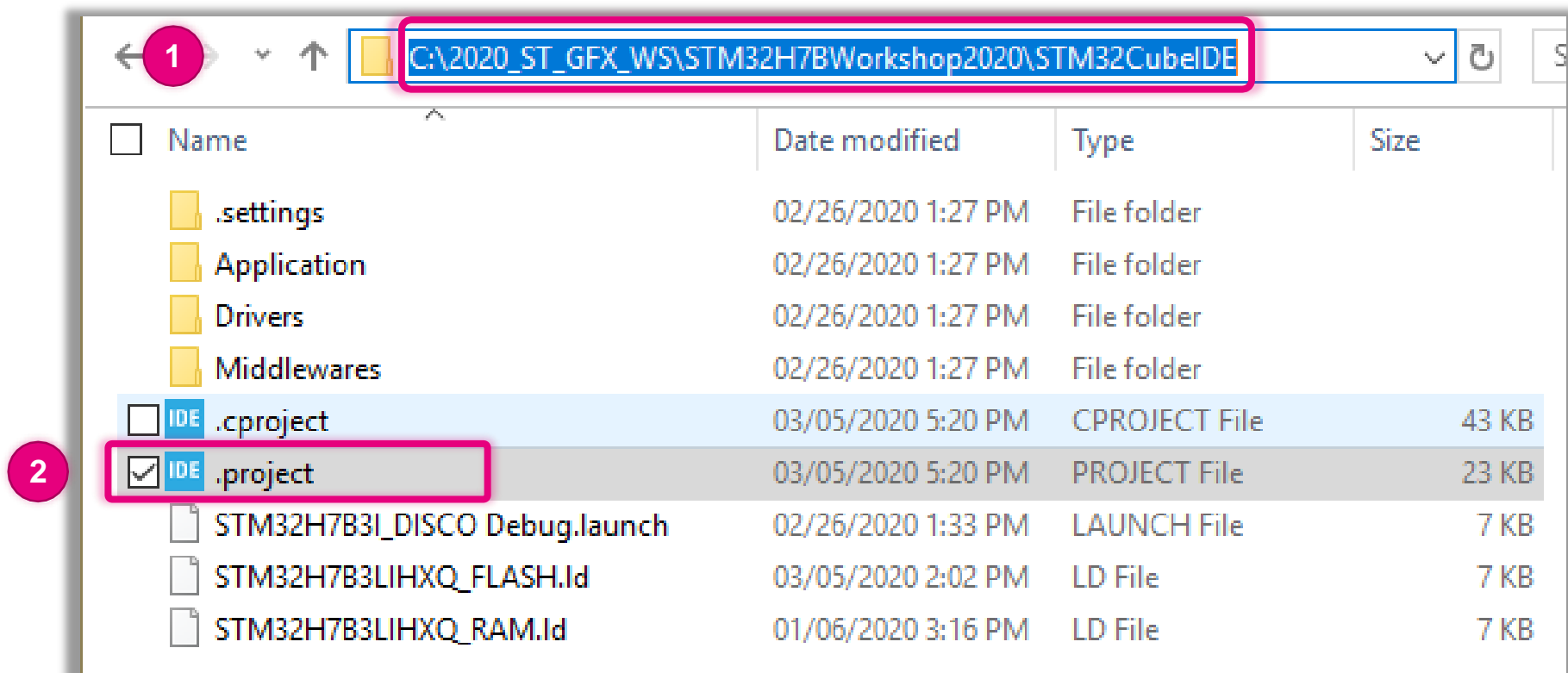
Simulator compilation scripts

Display HW initialisation (LTDC, DMA2D, ...)



Открытие STM32CubeIDE

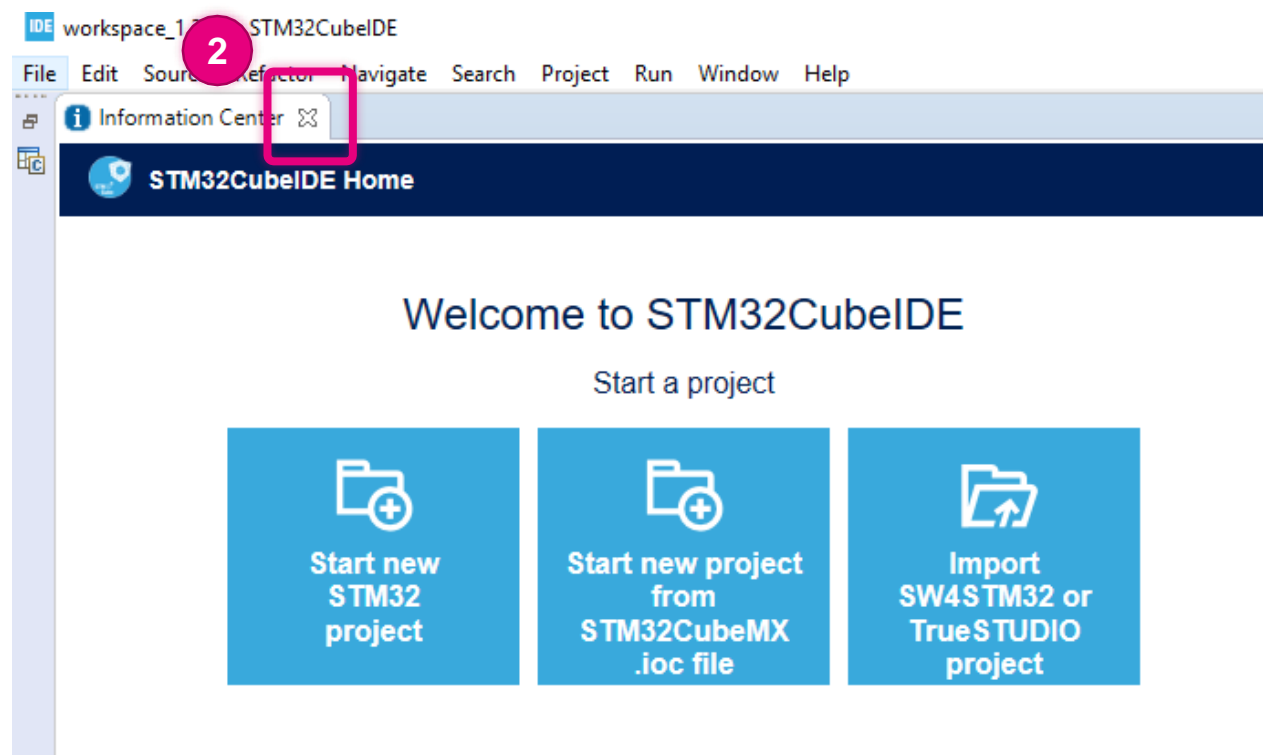
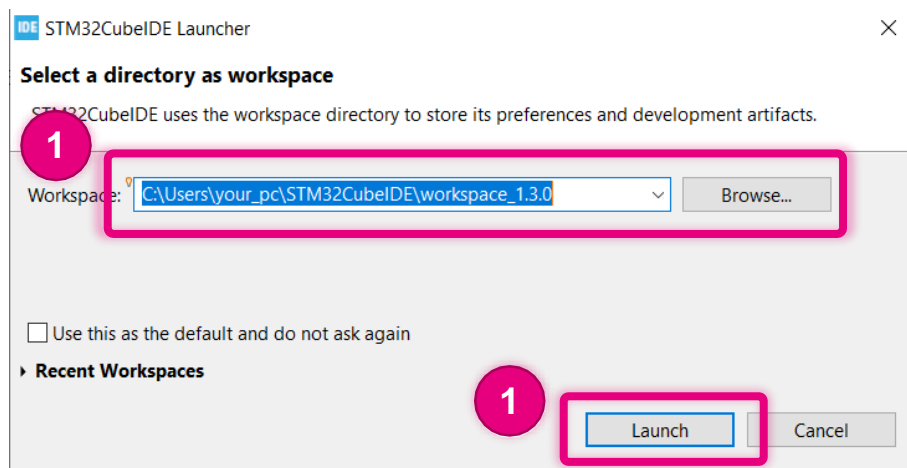
- 1 Перейдите в [папка проекта]\STM32CubeIDE
- 2 Двойным нажатием на **.project**, включите STM32CubeIDE





Запустите проект вCubeIDE

- 1 Выберите workspace для CubeIDE
- 2 Закройте информационное окно





Внутри CubeIDE

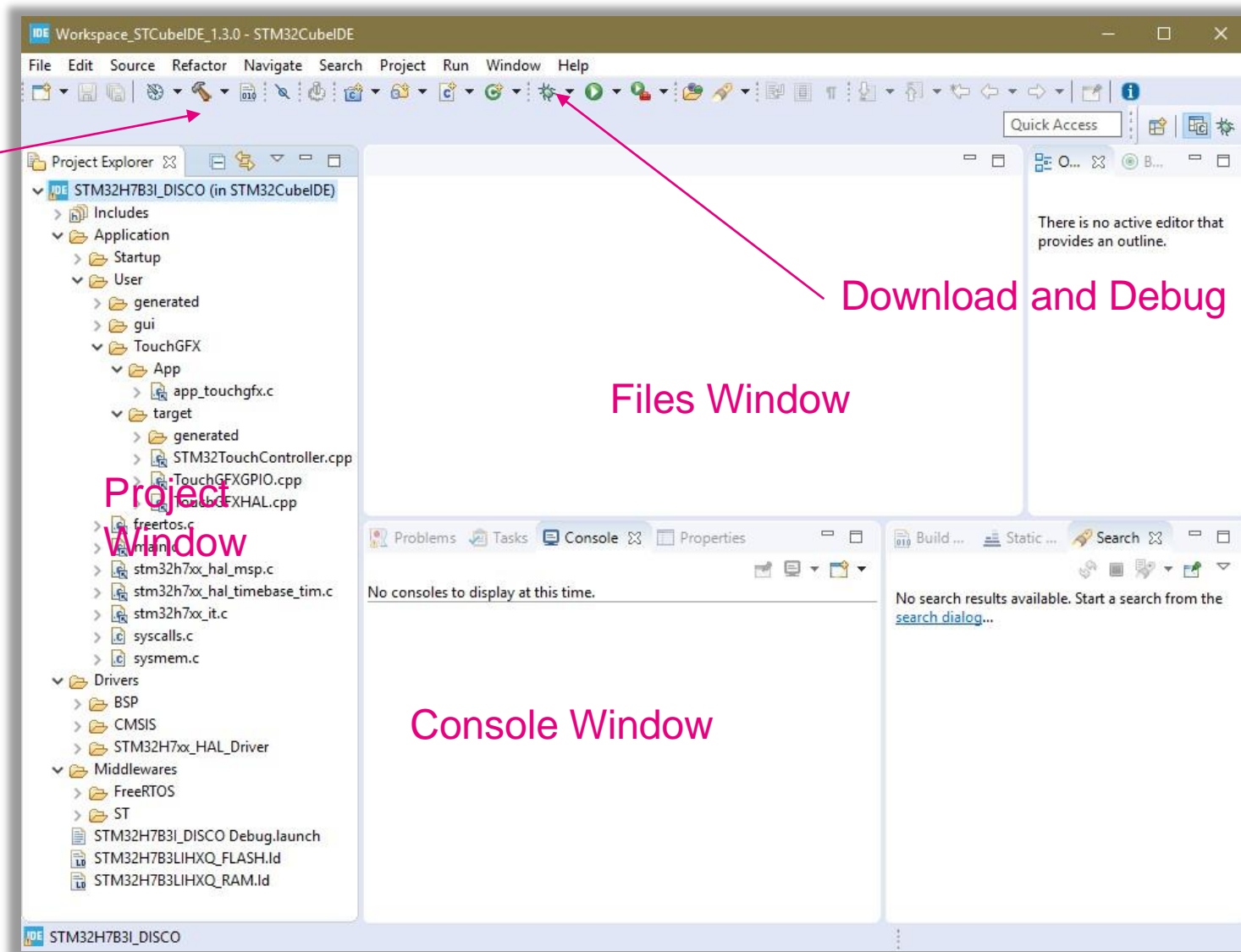
Build Button

Download and Debug Button

Project Window

Files Window

Console Window

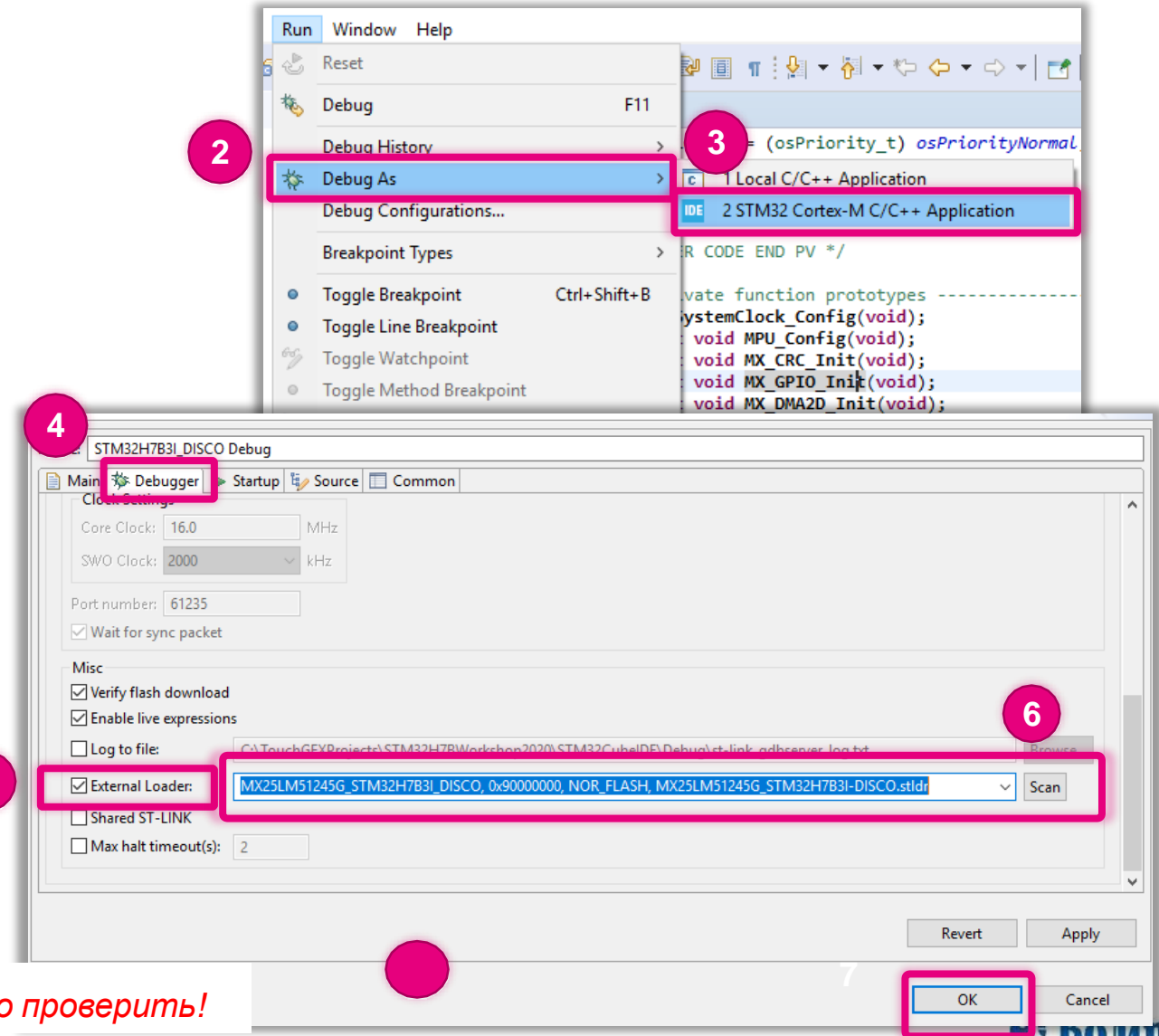




Настройте внешний загрузчик H7B

> IDE STM32H7B3I_DISCO (in STM32CubeIDE)

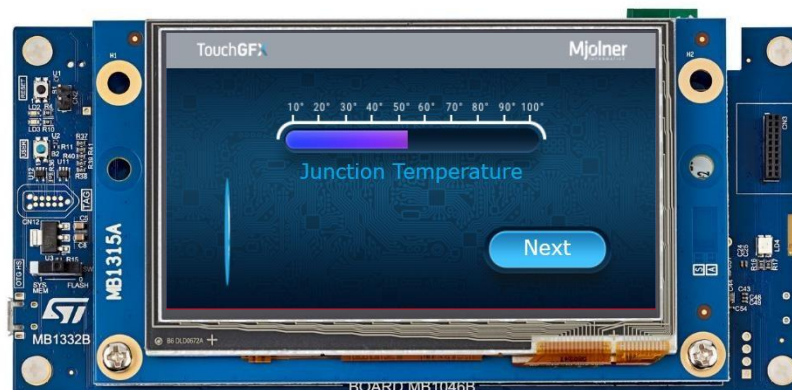
1. Нажмите на проект
2. Нажмите **Run, Debug As**
3. Выберите **STM32 Cortex-M C/C++ Application**
4. Выберите **Debugger**
5. Нажмите **External loader**, далее **Scan** и выберите отладку_STM32H7B3I-DISCO.stldr
6. Нажмите **Ok**



- Запустите приложение, нажав на зеленый треугольник



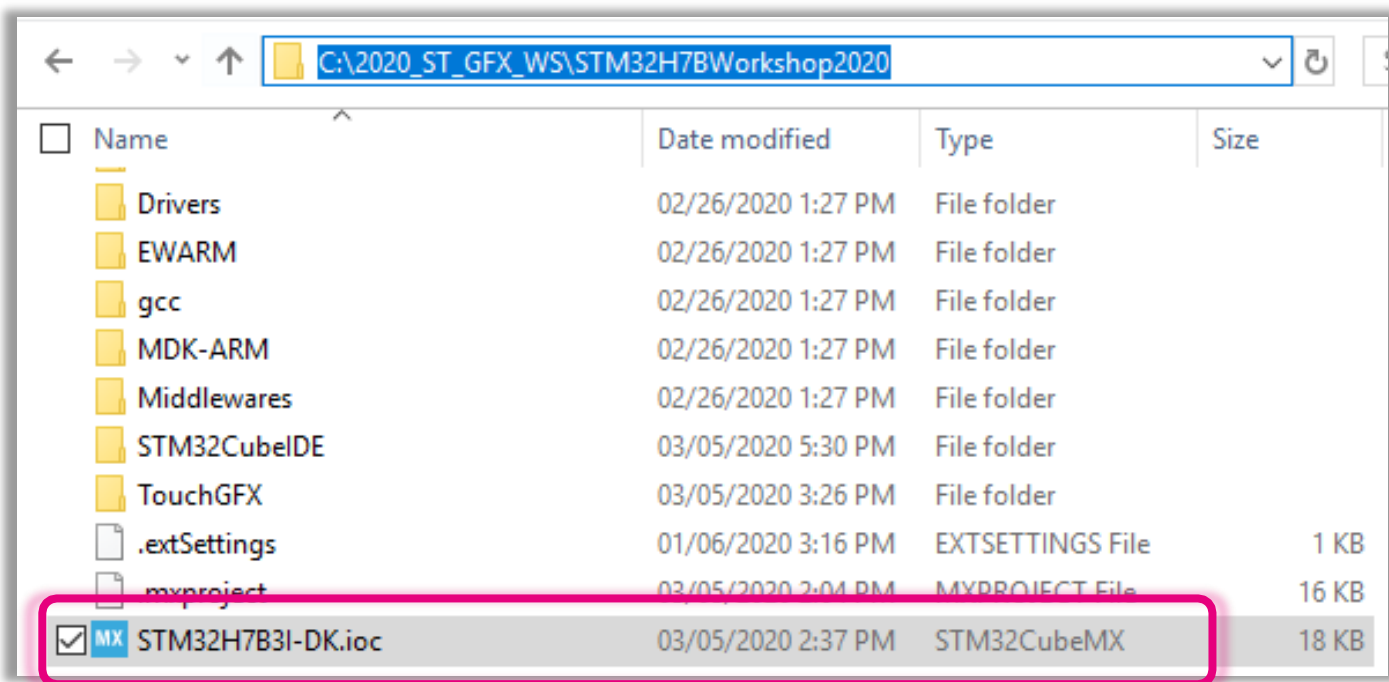
- Проверьте результат на отладке:





Инициализация АЦП в CubeMX

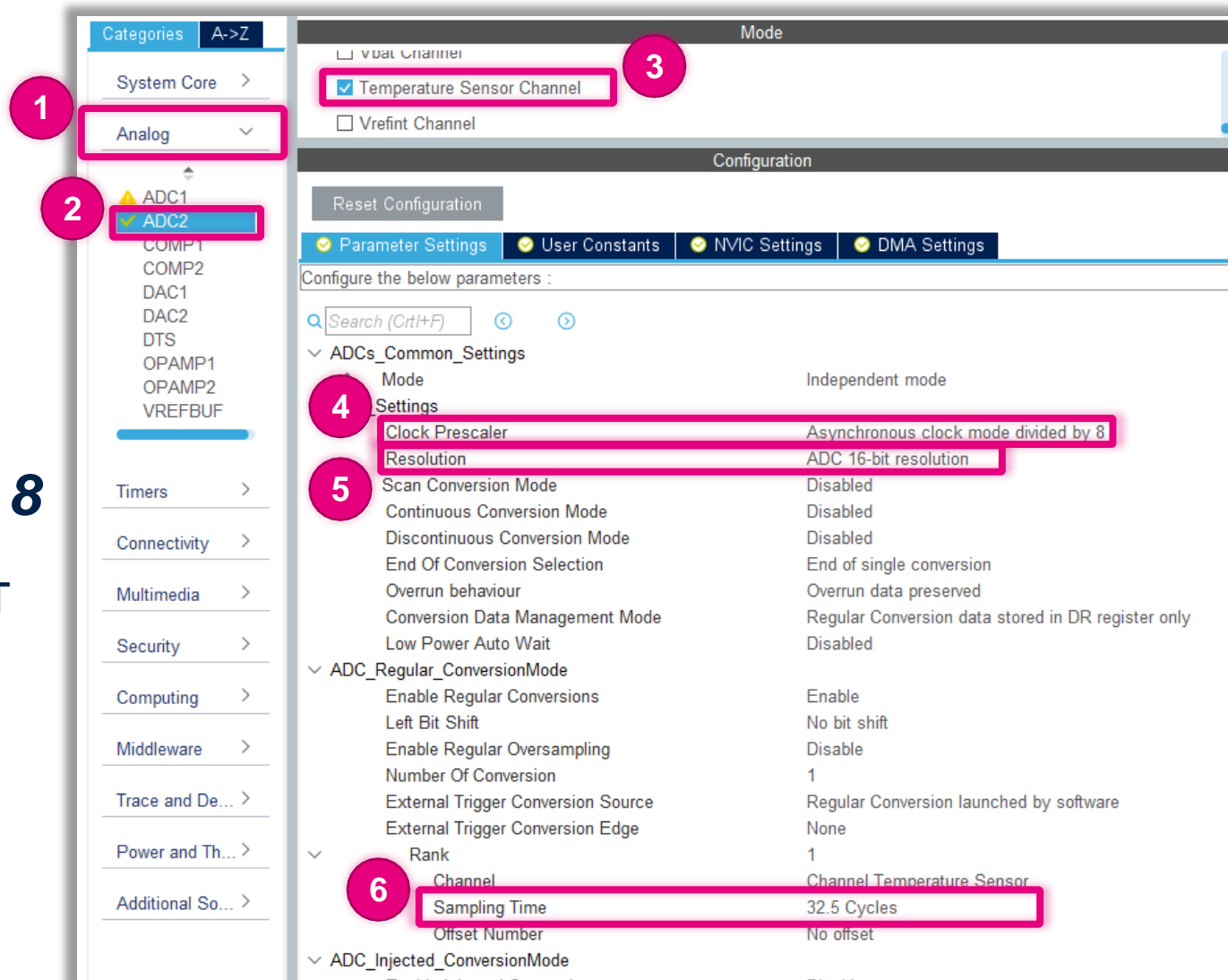
Откройте папку [название проекта] \ **STM32H7B3I-DK.ioc**





Настройка ADC2 в CubeMX

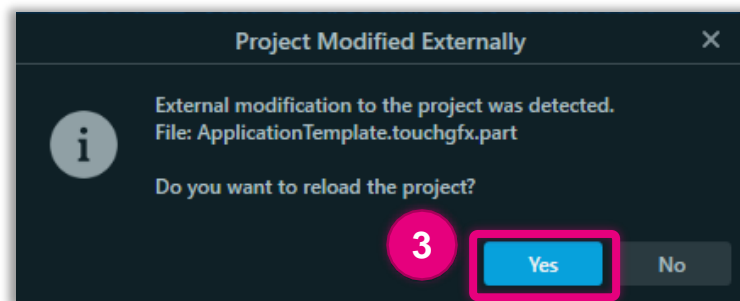
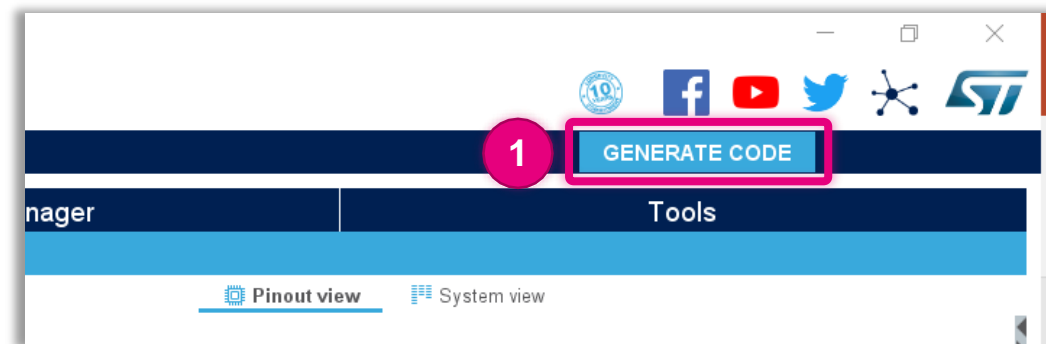
1. Откройте секцию **Analog**
2. Выберите **ADC2**
3. Разрешите **Temperature Sensor Channel**
4. Установите предделитель ***Asynchronous clock divided by 8***
5. Установите разрешение в 16-бит
6. В секции **Rank** установите **Sampling time** в 32.5 cycles





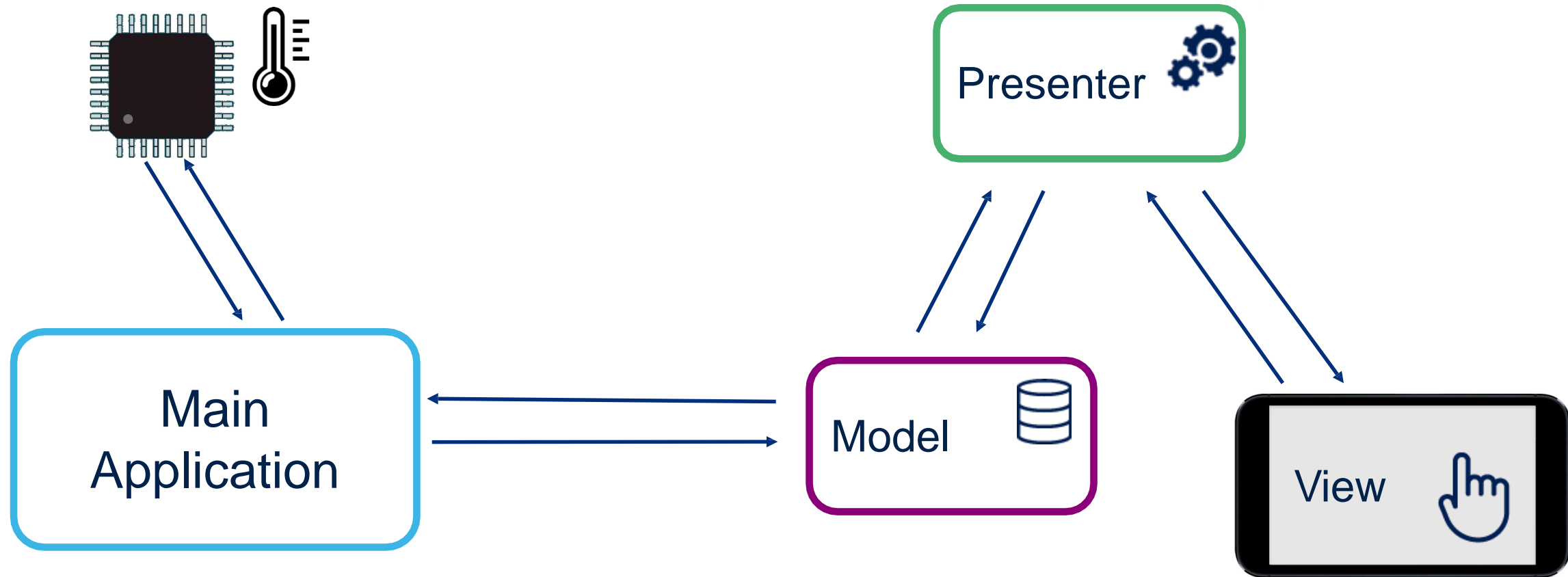
Сгенерируйте код в CubeMX

- Нажмите кнопку Generate Code
- Закройте всплывшее окно в CubeIDE
- Закройте окно в TouchGFX Designer через нажатие на **Yes**

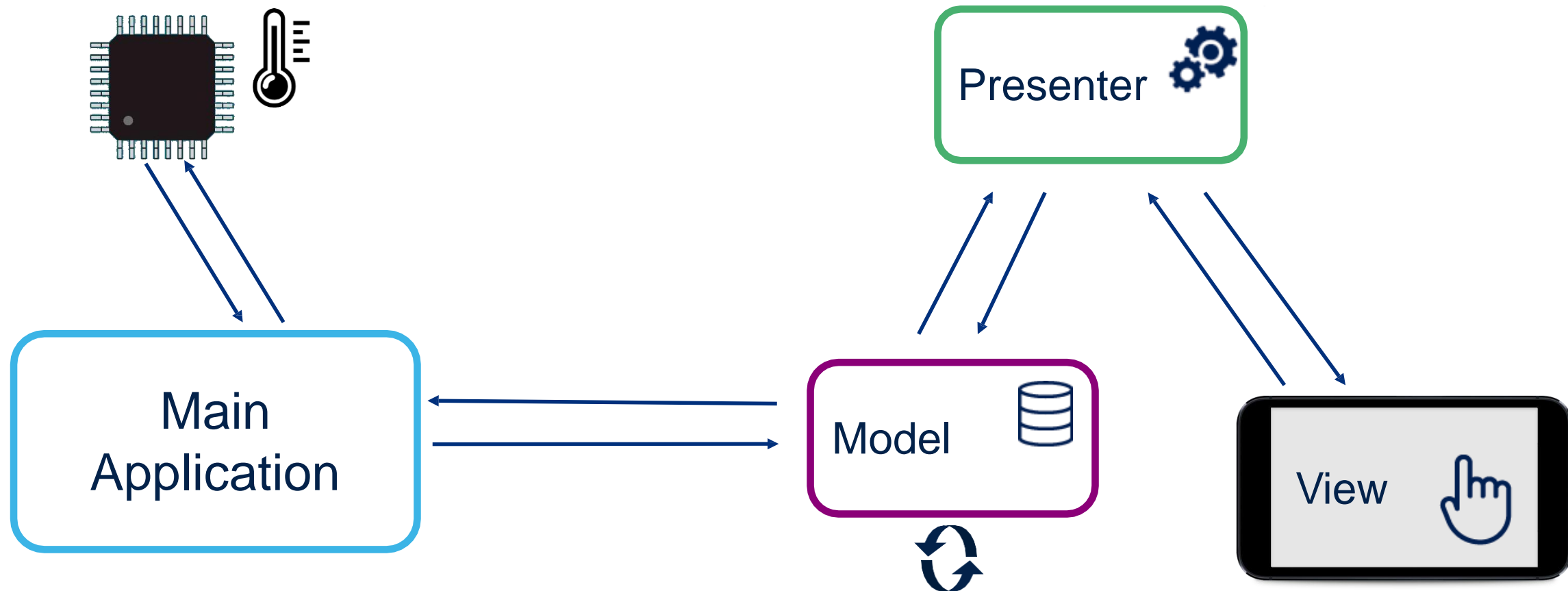




Model-View-Presenter

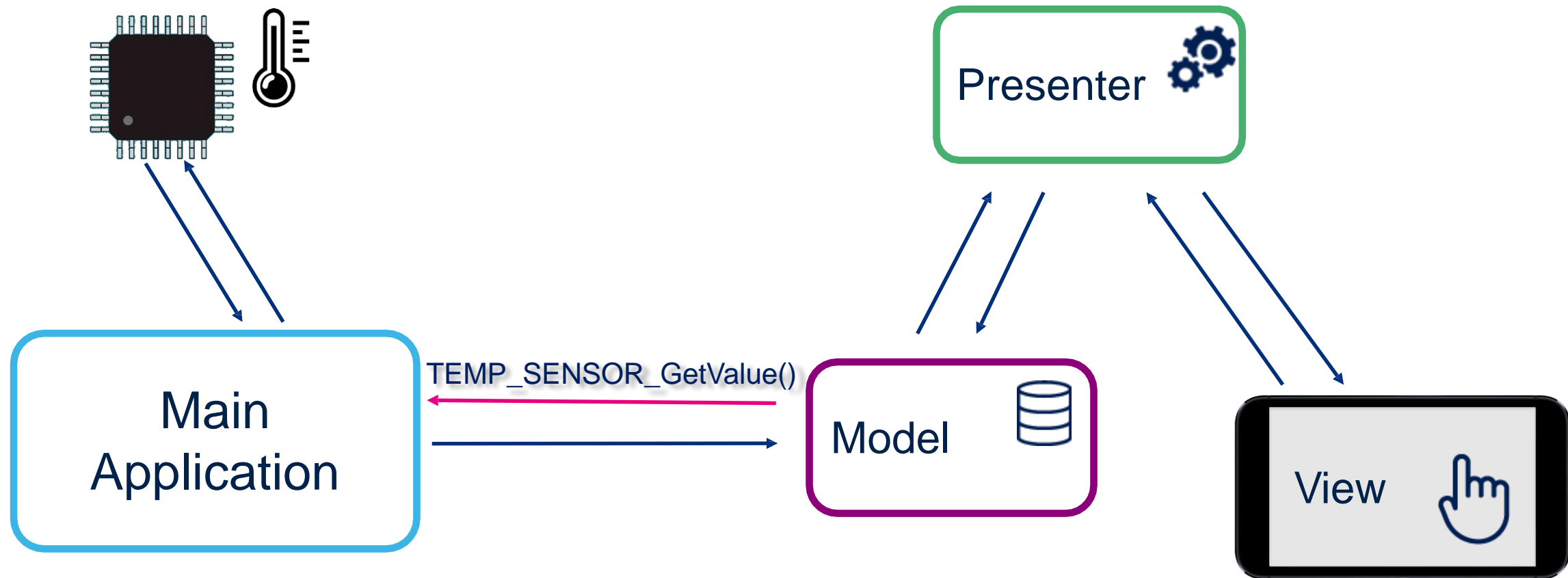


Model-View-Presenter



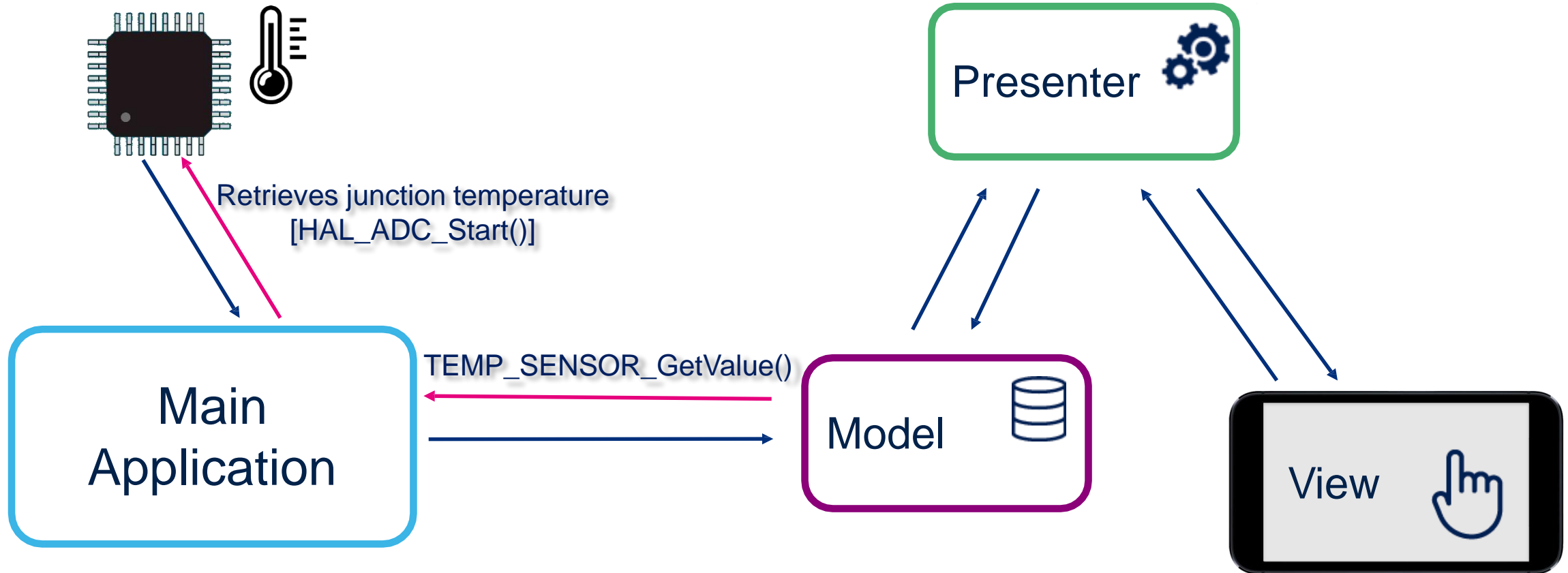
Model::tick() is called on each frame rendering (TGFX framework)

Model-View-Presenter



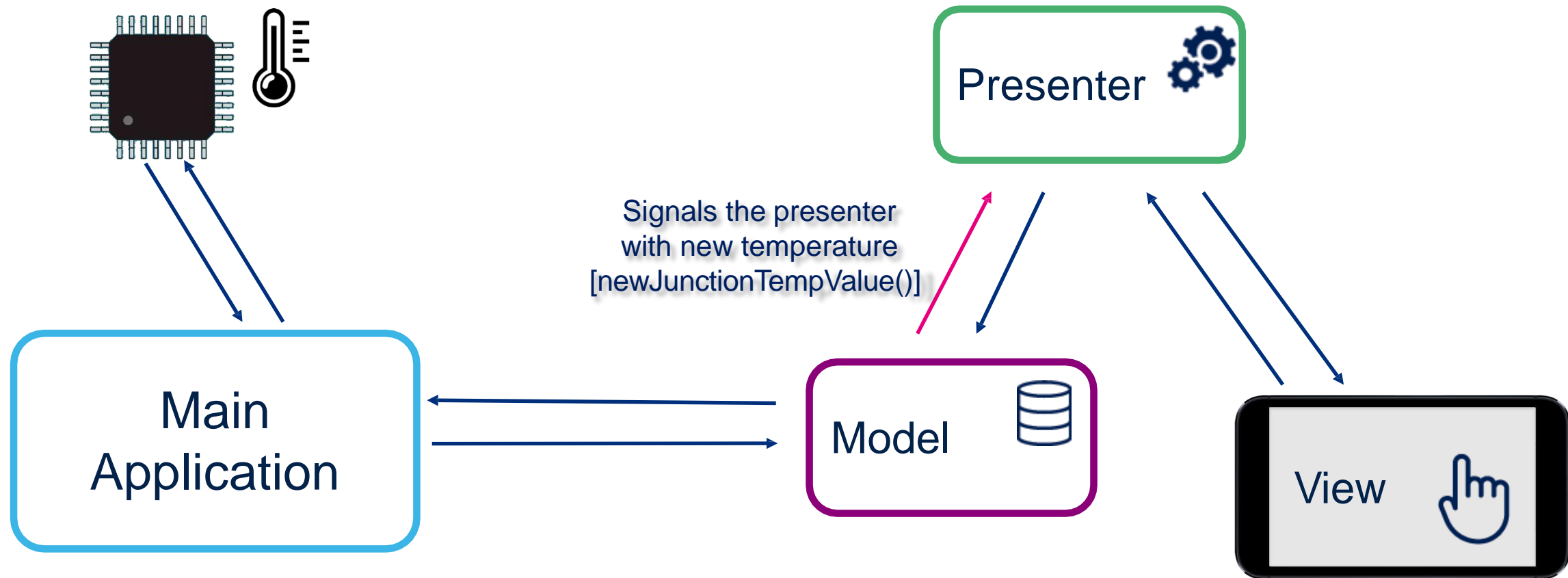


Model-View-Presenter

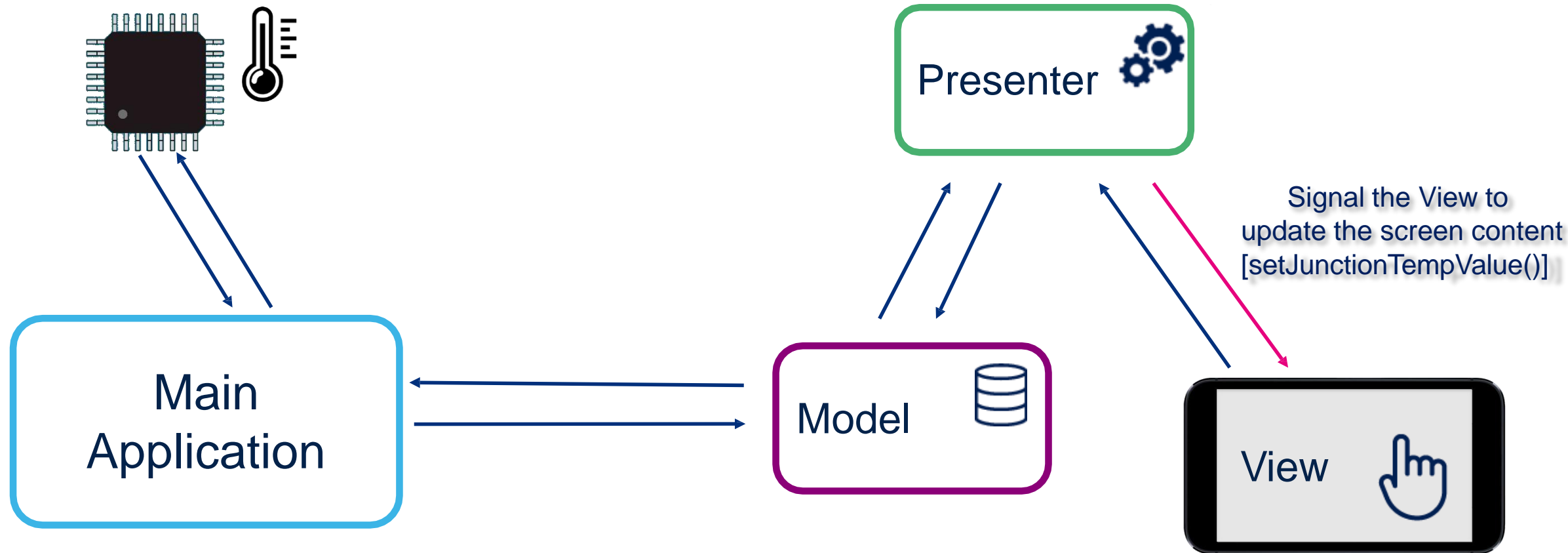




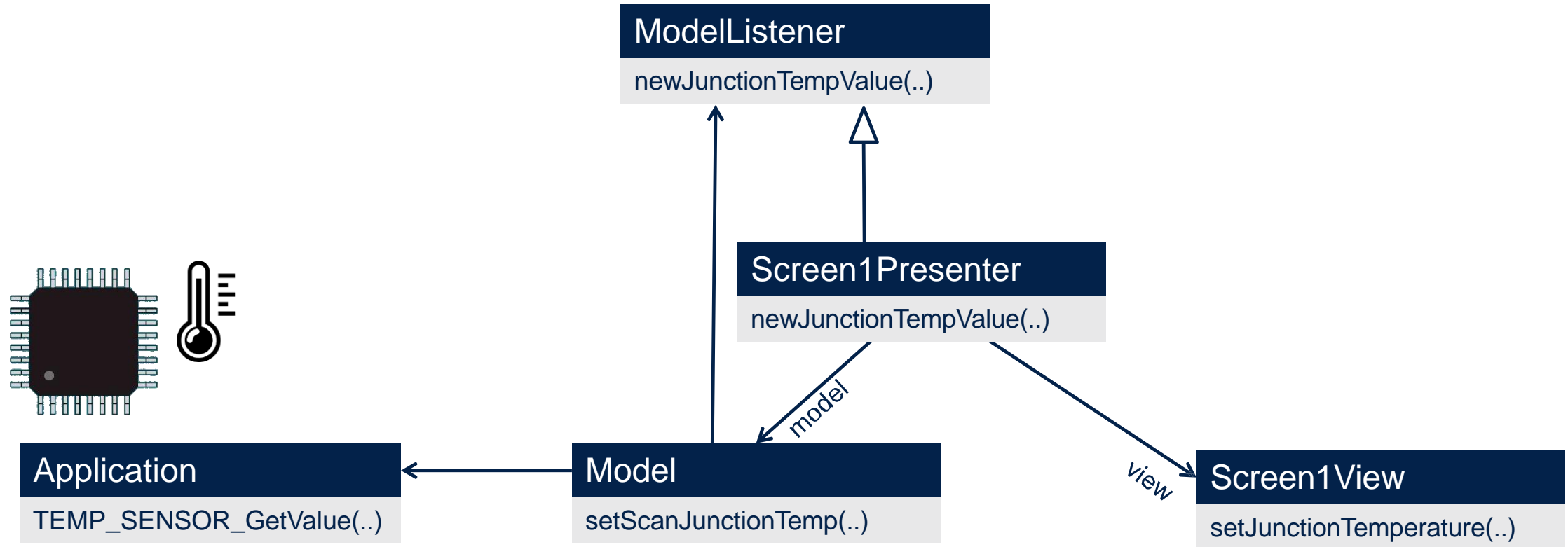
Model-View-Presenter



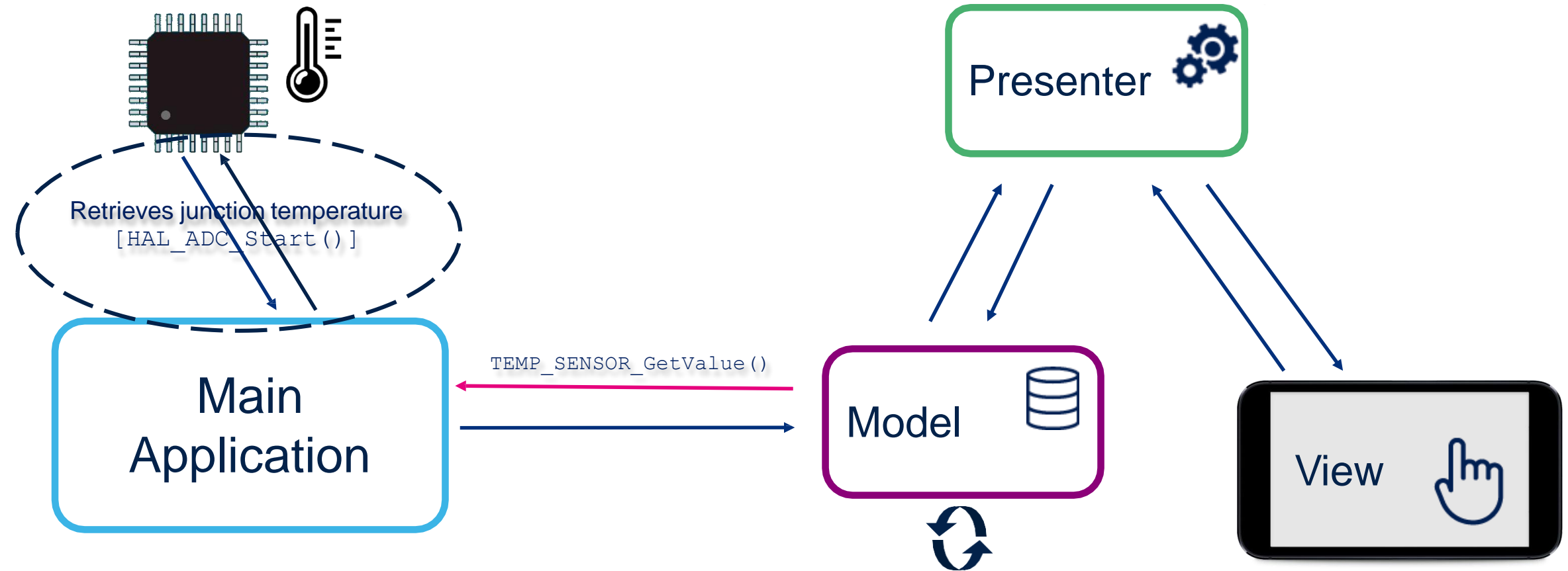
Model-View-Presenter



UML



Model-View-Presenter





Шаг 1: получение внутренней температуры

Core\Src\main.cpp (on disk)

Application\User\main.cpp (in CubeIDE)

- Новая функция

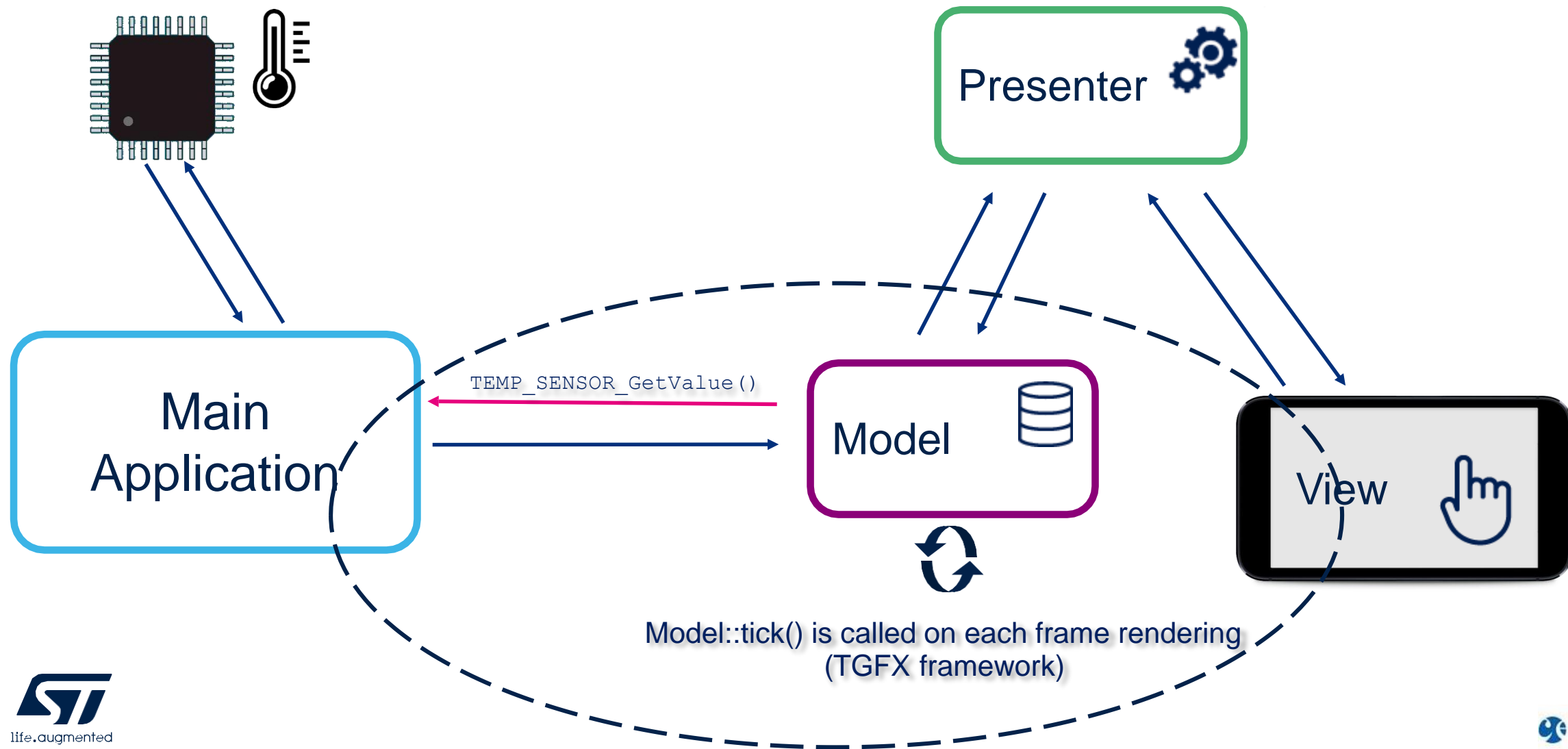
TEMP_SENSOR_GetValue(): start the ADC conversion and retrieve the current junction temperature value

```
uint32_t TEMP_SENSOR_GetValue(void)
{
    #define TEMP110_CAL_VALUE ((uint16_t*)((uint32_t)0x08FFF818))
    #define TEMP30_CAL_VALUE  ((uint16_t*)((uint32_t)0x08FFF814))
    #define TEMP110            110.0f
    #define TEMP30             30.0f

    if (HAL_ADC_Start(&hadc2) == HAL_OK)
    {
        if (HAL_ADC_PollForConversion(&hadc2, 1000) == HAL_OK)
        {
            return (int32_t)((((TEMP110 - TEMP30) *
                                (HAL_ADC_GetValue(&hadc2) - (*TEMP30_CAL_VALUE)))/
                                ((*TEMP110_CAL_VALUE) - (*TEMP30_CAL_VALUE))
                                + TEMP30);

        }
    }
    return 0xFFFFFFFF;
}
```

Шаг 2: Обновление модели





Шаг 2: Обновление модели

TouchGFX\gui\src\model\Model.cpp (on disk)
Application\User\gui\Model.cpp (in CubeIDE)

- **Добавление новых методов**

- `Model::getTempValue()`
Retrieve the temperature from the main application through the `TEMP_SENSOR_GetValue()` function
- `Model::setScanJunctionTemp()`
Enable scan of the temperature in the tick loop of the model

- **Обновление метода**

- `Model::tick()`
 - Get current temperature from application
 - Notify the presenter of a new temperature value

```
int Model::getTempValue()
{
    #ifndef SIMULATOR
    return TEMP_SENSOR_GetValue();
    #else
    // Implementation for simulator
    return 25;
    #endif /*SIMULATOR*/
}

extern "C" {
    extern uint32_t TEMP_SENSOR_GetValue(void);
}

void Model::setScanJunctionTemp(bool scanEnabled)
{
    scanJunctionTemp = scanEnabled;
}
```

```
void Model::tick()
{
    tickCounter++;
    if (((tickCounter % 20) == 0) &&(scanJunctionTemp))
    {
        if (modelListener != 0)
        {
            modelListener->newJunctionTempValue(getTempValue());
        }
    }
}
```



...\Model.hpp (in CubeIDE)

- Обновление класса

Шаг 2: Обновление модели

```
class Model
{
public:
    Model();
    void bind(ModelListener* listener)
    {
        modelListener = listener;
    }
    void tick();

    void setScanJunctionTemp(bool scanEnabled);
    int getTempValue();
protected:
    ModelListener* modelListener;

    bool scanJunctionTemp = false;
    int tickCounter;
};
#endif // MODEL_HPP
```



Шаг 2: Обновление модели

...\ModelListener.hpp (in CubeIDE)

- Обновление класса

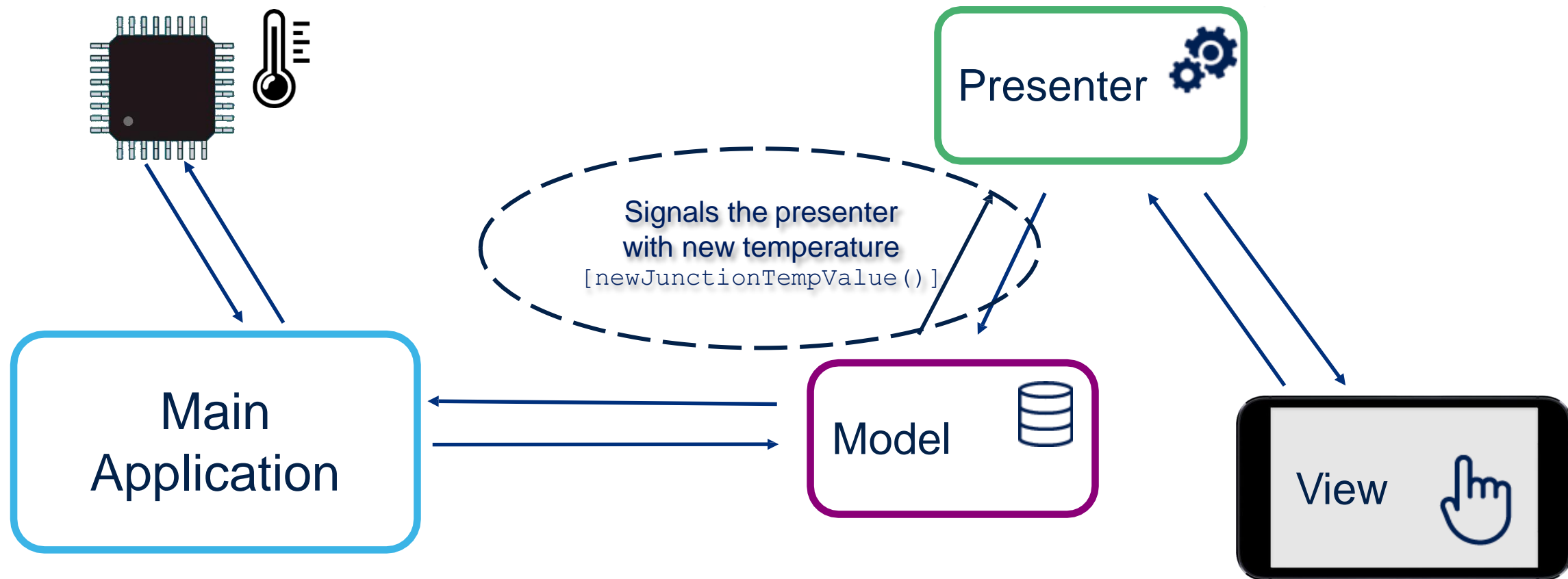
```
class ModelListener
{
public:
    ModelListener() : model(0) {}
    virtual ~ModelListener() {}
    void bind(Model* m)
    {
        model = m;
    }
    virtual void newJunctionTempValue(int value){};

protected:
    Model* model;
};

#endif // MODELLISTENER_HPP
```



Шаг 2: Обновление представителя





Шаг 2: Обновление представителя

TouchGFX\gui\src\screen1_screen\Screen1Presenter.cpp (on disk)

Application\User\gui\Screen1Presenter.cpp (in CubeIDE)

- Добавление нового метода

- `Screen1Presenter::newJunctionTempValue()`: calls the `setJunctionTempValue()` method of the `Screen1View` instance

```
void Screen1Presenter::newJunctionTempValue(int value)
{
    view.setJunctionTempValue(value);
}
```

- Обновление метода

- `Screen1Presenter::activate()`: enable the scan of the temperature in the `Model` when the presenter is activated

```
void Screen1Presenter::activate()
{
    model->setScanJunctionTemp(true);
}
```



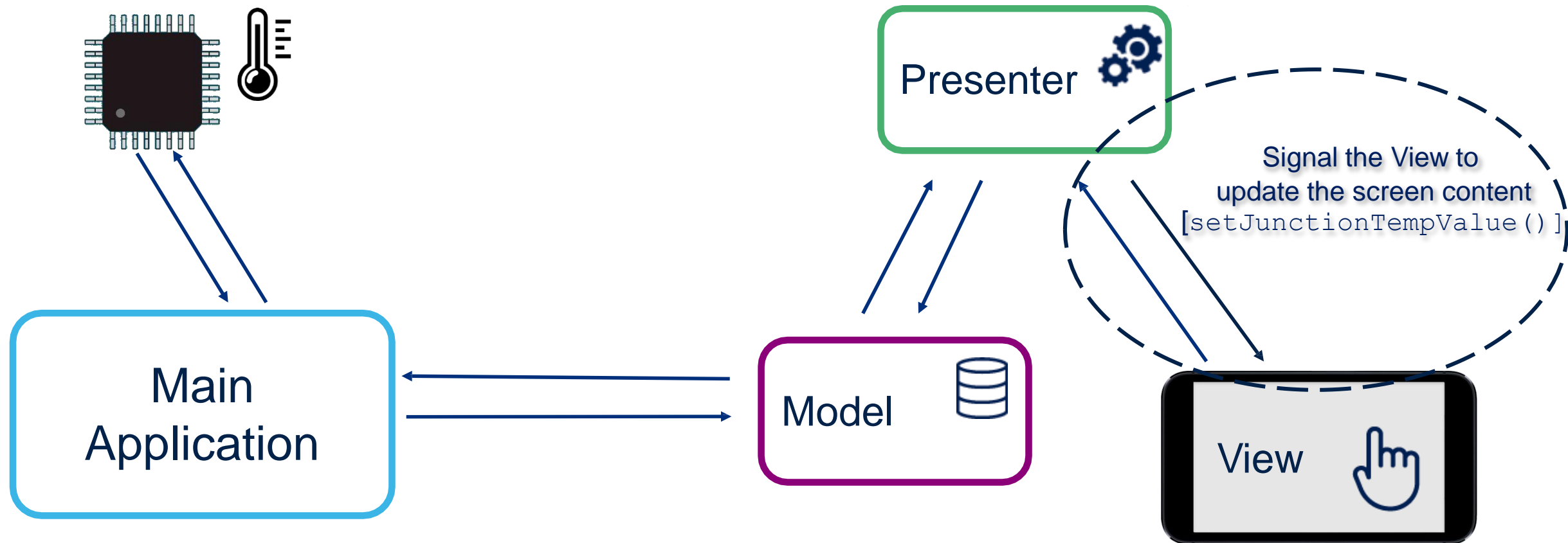

Шаг 2: Обновление представителя

...\Screen1Presenter.hpp (in CubeIDE)

- Обновление класса

```
class Screen1Presenter : public touchgfx::Presenter,  
public ModelListener  
{  
public:  
    Screen1Presenter(Screen1View& v);  
    virtual void activate();  
    virtual void deactivate();  
    virtual ~Screen1Presenter() {};  
  
    virtual void newJunctionTempValue(int value);  
  
private:  
    Screen1Presenter();  
    Screen1View& view;  
};  
  
#endif // SCREEN1PRESENTER_HPP
```

Обновление представления





Обновление представления

TouchGFX\gui\src\screen1_screen\Screen1View.cpp (on disk)

Application\User\gui\Screen1View.cpp (in CubeIDE)

- Добавление нового метода
 - `Screen1View::setJunctionTempValue()`:
 - Check if temperature as changed
 - Update the temperatureBar

```
void Screen1View::setJunctionTempValue(int value)
{
    if (lastTempValue != value)
    {
        lastTempValue = value;
        temperatureBar.setValue(value);
        temperatureBar.invalidate();
    }
}
```



Обновление представления

...\Screen1View.hpp (in CubeIDE)

- Обновление класса

```
class Screen1View : public Screen1ViewBase
{
public:
    Screen1View();
    virtual ~Screen1View() {}
    virtual void setupScreen();
    virtual void tearDownScreen();

    void setJunctionTempValue(int value);
protected:

    int lastTempValue;
};

#endif // SCREEN1VIEW_HPP
```

Спасибо!



Отладочные платы

NUCLEO-H7A3ZI-Q

<https://www.compel.ru/infosheet/ST/NUCLEO-H7A3ZI-Q>

STM32H735G-DK

<https://www.compel.ru/infosheet/ST/STM32H735G-DK>

STM32H750B-DK

<https://www.compel.ru/infosheet/ST/STM32H750B-DK>

STM32H747I-DISCO

<https://www.compel.ru/infosheet/ST/STM32H747I-DISCO>



Вопросы по техническим характеристикам и особенностям работы компонентов направляйте своему менеджеру Компэл или по адресу msk@compel.ru

Заказы на поставку компонентов от 1шт. оформляются здесь www.electronshtik.ru

Просчитать оптовую поставку или заказать образцы поможет ваш менеджер Компэл или специалист по адресу msk@compel.ru