

# Contents

- [1 Gesture Types](#)
- [2 Activating Gestures](#)
- [3 Deactivating Gestures](#)
- [4 Processing Gestures](#)
- [5 Working with Gesture Areas](#)
- [6 DOWNLOAD: Example of a project](#)

## [DOWNLOAD: Example of a project](#)

## Gesture Types

There are 8 types of gestures:

- **IR.GESTURE\_SWIPE\_LEFT**
- **IR.GESTURE\_SWIPE\_RIGHT**
- **IR.GESTURE\_SWIPE\_UP**
- **IR.GESTURE\_SWIPE\_DOWN**
- **IR.GESTURE\_PINCH\_IN**
- **IR.GESTURE\_PINCH\_OUT**
- **IR.GESTURE\_LONG\_PRESS**
- **IR.GESTURE\_DOUBLE\_TAP**

## Activating Gestures

To use gestures in iRidium you should activate them. To activate gestures use the command:

**IR.AddRecognizer(Gesture\_Type);**

Example of activating gestures at the application launch:

```
IR.AddListener(IR.EVENT_START,0,function() // Event is activated at the
application launch
{
    IR.AddRecognizer(IR.GESTURE_SWIPE_LEFT); // Activate gesture SWIPE_LEFT
    IR.AddRecognizer(IR.GESTURE_SWIPE_RIGHT); // Activate gesture SWIPE_RIGHT
    IR.AddRecognizer(IR.GESTURE_SWIPE_UP);    // Activate gesture SWIPE_UP
    IR.AddRecognizer(IR.GESTURE_SWIPE_DOWN);  // Activate gesture SWIPE_DOWN
});
```

Activated gestures work until the application is working or until they are deactivated. If some gesture is already activated its reactivation is to the detriment of system resources.

## Deactivating Gestures

If the activated gesture is not required any more it can be deactivated. To deactivate gestures use the command:

## **IR.RemoveRecognizer(Gesture\_Type);**

Example of deactivating gestures when exiting the application:

```
IR.AddListener(IR.EVENT_EXIT,0,function() // Event is activated when exiting
the application
{
    IR.RemoveRecognizer(IR.GESTURE_SWIPE_LEFT); // Deactivate gesture
    SWIPE_LEFT
    IR.RemoveRecognizer(IR.GESTURE_SWIPE_RIGHT); // Deactivate gesture
    SWIPE_RIGHT
    IR.RemoveRecognizer(IR.GESTURE_SWIPE_UP); // Deactivate gesture
    SWIPE_UP
    IR.RemoveRecognizer(IR.GESTURE_SWIPE_DOWN); // Deactivate gesture
    SWIPE_DOWN
});
```

## **Processing Gestures**

The following event is responsible for gesture processing:

### **IR.EVENT\_GESTURE\_BEGIN**

To process gestures connect IR.AddListener, the event is sent to it:

```
IR.AddListener(IR.EVENT_GESTURE_BEGIN, Page_Name, function(gesture, x,  
y){Function_Body});
```

Gesturs are processed on pages only. Each page requires connection of a separate Listener To process gestures for a particular page you are required to send its name to Listener - **Page\_Name**. To receive data about gestures Listener receives the function with parameters:

- **gesture** - the variable stores the gesture type;
- **x** - the variable stores x-coordinate of gesture start;
- **y** - the variable stores y-coordinate of gesture start;

To describe commands for each gestures it is convenient to use the choice structure.

Example of gesture processing:

```
var button;
IR.AddListener(IR.EVENT_START, 0, function() // Event is activated at the
application launch
{
```

```

// Create a button for switching pages
var button = IR.CreateItem(IR.ITEM_BUTTON, "text", 10, 10);
button.Text = "";

// Activating gestures
IR.AddRecognizer(IR.GESTURE_SWIPE_LEFT); // Activate gesture SWIPE_LEFT
IR.AddRecognizer(IR.GESTURE_SWIPE_RIGHT); // Activate gesture SWIPE_RIGHT
IR.AddRecognizer(IR.GESTURE_SWIPE_UP); // Activate gesture SWIPE_UP
IR.AddRecognizer(IR.GESTURE_SWIPE_DOWN); // Activate gesture SWIPE_DOWN

// Connect Listener, send the current page and the function with
parameters.
IR.AddListener(IR.EVENT_GESTURE_BEGIN, IR.CurrentPage, function(gesture,
x, y)
{

    // Choice structure
    switch(gesture)
    {
        case IR.GESTURE_SWIPE_LEFT:
            button.Text = "Left"; // If you start a gesture SWIPE_LEFT you
see the sign "Left"
            break;
        case IR.GESTURE_SWIPE_RIGHT:
            button.Text = "Right"; // If you start a gesture SWIPE_RIGHT you
see the sign "Right"
            break;
        case IR.GESTURE_SWIPE_UP:
            button.Text = "Up"; // If you start a gesture SWIPE_UP you
see the sign "Up"
            break;
        case IR.GESTURE_SWIPE_DOWN:
            button.Text = "Down"; // If you start a gesture SWIPE_DOWN you
see the sign "Down"
            break;
    }
});
});

```

## Working with Gesture Areas

If it gestures are required to work only in a particular area on the screen use parameters x and y which store coordinates of gesture start. You can describe any area on the screen using these coordinates and calculations.

As an example let's describe a half of iPad screen and activate gestures in the described area only:

```

var button;
IR.AddListener(IR.EVENT_START, 0, function() // Event is activated at the
application launch
{
    // Create a button for switching pages
    var button = IR.CreateItem(IR.ITEM_BUTTON, "text", 10, 10);
    button.Text = "";

    // Activating gestures
    IR.AddRecognizer(IR.GESTURE_SWIPE_LEFT); // Activate gesture SWIPE_LEFT
    IR.AddRecognizer(IR.GESTURE_SWIPE_RIGHT); // Activate gesture SWIPE_RIGHT
    IR.AddRecognizer(IR.GESTURE_SWIPE_UP); // Activate gesture SWIPE_UP
    IR.AddRecognizer(IR.GESTURE_SWIPE_DOWN); // Activate gesture SWIPE_DOWN

    // Connect Listener, send the current page and the function with
parameters.
    IR.AddListener(IR.EVENT_GESTURE_BEGIN, IR.CurrentPage, function(gesture,
x, y)
    {
        // The width of iPad is 1024 pixels, 512 is a half
        if (x <= 512) // If the gesture start equals or is less than a half of
the screen
        {
            // Choice sructure
            switch(gesture)
            {
                case IR.GESTURE_SWIPE_LEFT:
                    button.Text = "Left"; // If you start a gesture SWIPE_LEFT you
see the sign "Left"
                    break;
                case IR.GESTURE_SWIPE_RIGHT:
                    button.Text = "Right"; // If you start a gesture SWIPE_RIGHT you
see the sign "Right"
                    break;
                case IR.GESTURE_SWIPE_UP:
                    button.Text = "Up"; // If you start a gesture SWIPE_UP you
see the sign "Up"
                    break;
                case IR.GESTURE_SWIPE_DOWN:
                    button.Text = "Down"; // If you start a gesture SWIPE_DOWN you
see the sign "Down"
                    break;
            }
        }
    });
});

```

**[DOWNLOAD: Example of a project](#)**