```
// Get the unique identifiers of the module and the bus at launch
 IR.AddListener(IR.EVENT_MODULE_START, 0, function(moduleID, busID){
   // Receiving the module copy
   // moduleID - the unique module identifier
   // Assigning of the variable module
   var module = B.getModule(moduleID);
   // Form the subnet name and assign it to the variable netWorkName
   var netWorkName = "HDL-BUS Pro Network (UDP)";
   // Take the existing device and assign it to the variable device
   var device = module.getDevice(netWorkName);
   // Receiving the subnet address
   var subnetID = module.FromSubnetID;
   // Receiving the device number
   var deviceID = module.FromDeviceID;
   // Form the device name
   var deviceName = "HDL-MC48IPDMX.231";
   // Form the channel name
   var statusOnStartName = 'deviceName + ":" + "statusOnStart";
   // The separator used in commands
   var separator = 0x0;
   // The table of HDL command codes
   var HDLCodes = {
      singleChannelLigtning: 0x31,
      singleChannelReadTarget: 0x33,
      singleSceneControl: 0x02,
      singleSequenceControl: 0x1A
   };
   // Short name of channels for simplicity
   var drivers = "Drivers." + netWorkName + "." + deviceName + ":" +
"channel";
   // Form the channel parameters
   var Parametrs = [subnetID, deviceID, HDLCodes.singleChannelReadTarget,
separator, 0x0, 0x0, 0x0, 0x0, 0x1, 0x0, 0x1, 0xE8, 0x3, 0x0, 0x0]
   // If the device is connected then
   if(device) {
      // Add the channel StatusOnStart
      device.AddChannel(statusOnStartName, Parametrs);
      // Cache clearing
      B.clearEmulatorCahce(true);
      // The cycle for creating wigets
      for (var i = 0; i<3; i++){
         // Refer to the wiget you want to copy
         var popup = module.GetPopup("Dimmer" + moduleID);
         // Give a unique name
         var name = "Dimmer" + i;
         // Create a new subdevice with properties:
         // Device - the driver object of the iRidium device
         // The name of the created SubDevice which will be used to referring
to it. It will be seen by the user
         var NewSubDevice = module.addSubDevice({
                Device: device,
                DeviceName: name
        });
         // Wiget creation
         var widget = module.ClonePopup(popup,"Dimmer" + i)
         // Add the wiget on the screen
         NewSubDevice.addWidget(widget);
         // Declaring the variables required for wiget creation
         var channelName = deviceName + ":" + "channel" + i;
         var Dimmer = module.GetPopup("Dimmer" + i);
         var Dimmer_level = Dimmer.GetItem("Level");
         var Dimmer_button = Dimmer.GetItem("Button");
         var uiDimmer = "UI.Dimmer" + i;
         // Adding channels
         device.AddChannel(channelName, [subnetID, deviceID,
HDLCodes.singleChannelLigtning, separator, parseInt(i+1, 16), 0x0, 0x0, 0x0,
0x01, 0x0, 0x0]);
         device.AddTag(channelName, [subnetID, deviceID,
HDLCodes.singleChannelLigtning, separator, parseInt(i+1, 16), 0x0, 0x0]);
         // Activation of the function for assigning the slider
         var UserSlider_1 = new UserSlider(Dimmer_level,
Dimmer.GetItem("Slider"));
         // Sending the command from the level to the HDL dimmer channel
         IR.AddListener(IR.EVENT_ITEM_RELEASE, Dimmer_level, function ()
         {
            var data = this;
            device.Set(data.channel, data.level.Value);
         },{level: Dimmer_level, channel: channelName});
         // The switching on/off (0 or 100) buton
         IR.AddListener(IR.EVENT_ITEM_PRESS, Dimmer_button, function()
         {
            var data = this;
            device.Set (data.channel, data.button.Value * 100);
         }, {button: Dimmer_button, channel: channelName});
         // Assigning graphic items to feedback channels
         Dimmer.GetItem("Name").Text = "Dimmer" + i;
         module.AddRelation(drivers + i , uiDimmer + ".Visible Level.Value");
         module.AddRelation(drivers + i , uiDimmer + ".Level.Value");
         module.AddRelation(drivers + i , uiDimmer + ".label main title
640x88.Value");
         module.AddRelation(drivers + i , uiDimmer + ".Button.Value");
      }
   };
   // The user slider for the level
   function UserSlider(Level, Slider)
   {
      Property = "X";
      Len = "Width";
      // The function for calculating the slider position in relation to the
livel
      function Move(){
         Slider[Property] = Level.Value * (Level[Len] -50) / 100;
      }
      // Subscription to events
      IR.AddListener(IR.EVENT_ITEM_PRESS, Level, Move); // pressing on the
level
      IR.AddListener(IR.EVENT_MOUSE_MOVE, Level, Move); // moving the mouse
on the level
      IR.AddListener(IR.EVENT_TOUCH_MOVE, Level, Move); // moving a finger on
the level
      IR.SetInterval(600, Move);  // auto-update in 600 ms
   };
 });
```

As a result of the module you should have 3 dimmers with different channels as it is shown in the image:

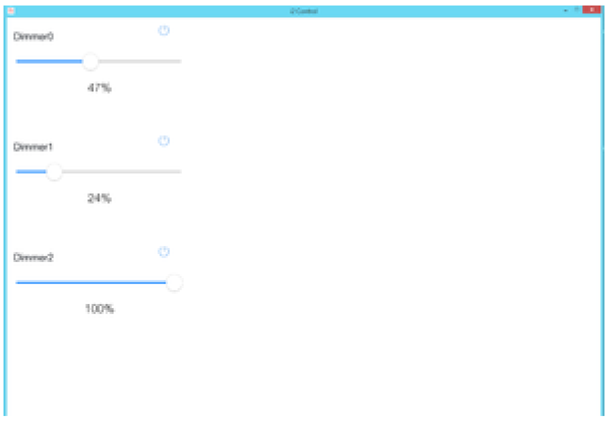

Рис. Result