```
// Получаем уникальный идентификатор модуля и шины при запуске
 IR.AddListener(IR.EVENT_MODULE_START, 0, function(moduleID, busID){
   // Получение копии модуля
   // moduleID - уникальный идентификатор модуля
   // Присваивание переменной module
   var module = B.getModule(moduleID);
   // Формируем имя подсети и присваиваем переменной netWorkName
   var netWorkName = "HDL-BUS Pro Network (UDP)";
   // Берем существующее устройство и присваиваем переменной device
   var device = module.getDevice(netWorkName);
   // Получение адреса подсети
   var subnetID = module.FromSubnetID;
   // Получение номера устройства
   var deviceID = module.FromDeviceID;
   // Формируем имя устройства
   var deviceName = "HDL-MC48IPDMX.231";
   // Формируем имя канала
   var statusOnStartName = 'deviceName + ":" + "statusOnStart"';
   // Разделитель, используемый в командах
   var separator = 0x0;
   // Таблица HDL кодов команд
   var HDLCodes = {
      singleChannelLigtning: 0x31,
      singleChannelReadTarget: 0x33,
      singleSceneControl: 0x02,
      singleSequenceControl: 0x1A
   };
   // Неполное имя каналов для простоты использования
   var drivers = "Drivers." + netWorkName + "." + deviceName + ":" +
"channel";
   // Формируем параметры канала
   var Parametrs = [subnetID, deviceID, HDLCodes.singleChannelReadTarget,
separator, 0x0, 0x0, 0x0, 0x0, 0x1, 0x0, 0x1, 0xE8, 0x3, 0x0, 0x0]
   // Если устройство подключено то
   if(device) {
      // Добавляем канал StatusOnStart
      device.AddChannel(statusOnStartName, Parametrs);
      // Очистка кеша
      B.clearEmulatorCahce(true);
      // Цикл для создания виджетов
      for (var i = 0; i<3; i++){
         // Обращаемся к виджету который собираемся скопировать
         var popup = module.GetPopup("Dimmer");
         // Присваиваем оригинальное имя
         var name = "Dimmer" + i;
         // Создаем новое под-устройство с параметрами:
         // Device - объект драйвера устройства iRidium
         // Имя создаваемого под-устройства, через которое будем к нему
обращаться и это же имя увидит пользователь
         var NewSubDevice = module.addSubDevice({
                Device: device,
```

```
                DeviceName: name
        });
         // Создание виджета
         var widget = module.ClonePopup(popup, "Dimmer" + i)
         // Добавляем виджет на экран
         NewSubDevice.addWidget(widget);
         // Объявления переменных необходимых для создания виджета
         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;
         // Добавление каналов
         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]);
         // Вызов функции привязки слайдера
         var UserSlider_1 = new UserSlider(Dimmer_level,
Dimmer.GetItem("Slider"));
         // Отправка команды с уровня на HDL канал диммера
         IR.AddListener(IR.EVENT_ITEM_RELEASE, Dimmer_level, function ()
         {
            var data = this;
            device.Set(data.channel, data.level.Value);
         },{level: Dimmer_level, channel: channelName});
         // Кнопка для включения и отключения (0 или 100)
         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});
         // Привязка графических элементов к фидбекам
         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");
      }
   };
   // Пользовательский слайдер для уровня
   function UserSlider(Level, Slider)
   {
      Property = "X";
      Len = "Width";
      // Функция вычисления позиции слайдера относительно уровня
      function Move(){
         Slider[Property] = Level.Value * (Level[Len] -50) / 100;
      }
```

```
      // Подписка на события
      IR.AddListener(IR.EVENT_ITEM_PRESS, Level, Move); // нажатие на уровень
      IR.AddListener(IR.EVENT_MOUSE_MOVE, Level, Move); // движение мыши по
уровню
      IR.AddListener(IR.EVENT_TOUCH_MOVE, Level, Move); // движение пальца по
уровню
      IR.SetInterval(600, Move);  // автообновление через 600 мс
   };
 });
```

Результатом данного модуля должны появиться 3 диммера с разными каналами, как показано на рисунке:

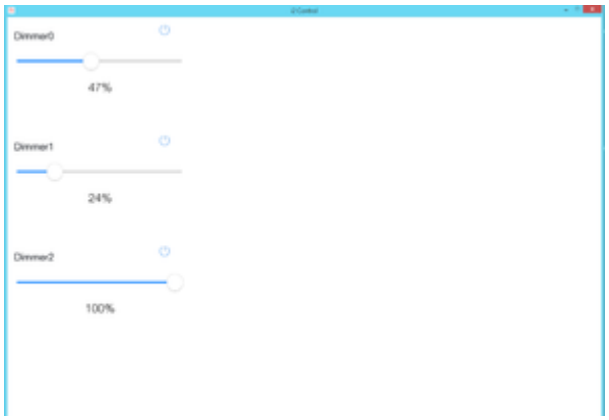

Рис. Результат