

```

// Calling the array of forbidden symbols
var simboli = ["\\", "@", "*", "/", "|", "&", "%", "#", "'", '^', ':', ';',
'$', '<', '<<', '>', '>>', '!', '?', '{', '}', '[', ']', '(', ')', '=', '\,
'.', ',', '`'];

// Add the channel StatusOnStart
device.AddChannel(deviceName + ":" + "statusOnStart" + moduleID, [subnetID,
deviceID, HDLCodes.singleChannelReadTarget, separator, 0x0, 0x0, 0x0, 0x0,
0x1, 0x0, 0x1, 0xE8, 0x3, 0x0, 0x0]);
// If the device is connected, then
if(device) {
    var cache = settingPopup.getData();
    //B.clearEmulatorCache(true);        // Cache clearing
    // Downloading cache is it does not equal to zero. Otherwise, creating
the variables
    if (cache == null) var cache = [];
    if (cache[1] == null) {
        // Filling in the array of free channels, if the cache is empty
        var m = [];
        for (var j = 0; j<48; j++)
            m[j] = j+1;
    }
    else
        var m = cache[1];
    // Filling in the list of free channels
    fillListFreeChannel();
    if (cache[2] == null) {
        var id = 0;
    }
    else
        var id = cache[2];
    if (cache[0] == null)
        var md = []
    else
        var md = cache[0];
    for (var i = 0; i < md.length; i++) {
        // Refer to the existing subdevice with the name from the cache
        var create = module.getSubDevice(md[i].data);
        // Assign the existing subdevice to the array
        md[i].subDevice = create;
        // Activating the function for creating dimmer widgets
        new CreateDimmer (md[i]);
        // filling in the list of subdevices
        fillListSubDevice();
    };
};

```