



## Busylight for Chrome SDK + Sample

The Busylight for Chrome Sample is a Chrome App using the Chrome HID interface to communicate with the Busylight Device.

This sample is intended for developers who want to integrate the Busylight into their own app. It contains an SDK file which can be used for own apps and a chrome app as a working sample.

### To install the App, follow these steps:

- Store the Busylight App, which is the complete content of the ZIP file somewhere on your computer.
- In Chrome, navigate to `chrome://extensions`
- In the extensions window, activate the development mode. (upper right corner)
- Click Load unpacked.
- Find and select the app folder you have selected in the first step. Do not be astonished not to see the content of the folder in the folder selector. Just select the folder and press OK. The app will be installed immediately.
- Open a new tab in Chrome click Apps and click the app in the app menu. If you don't have an Apps button, navigate to <chrome://apps/>. The app looks like this:



### How to use the Busylight in your own chrome app or extension:

Make sure you have included the Busylight Device list in the permissions part of your manifest.json:

```
{
  "name": "Busylight Tester",
  "manifest_version": 2,
  "version": "1.0.0",
  "minimum_chrome_version": "45.0.2439.3",
  "icons": {
    "16": "Busylight16.png"
  },
  "app": {
    "background": {
      "scripts": [ "main.js" ]
    }
  },
  "permissions": [
    "hid",
    {
      "usbDevices": [
        {
          "vendorId": 1240,
```



```
        "productId": 63560
      },
      {
        "vendorId": 10171,
        "productId": 15307
      },
      {
        "vendorId": 10171,
        "productId": 15306
      },
      {
        "vendorId": 10171,
        "productId": 15309
      }
    ]
  }
}
```

Include the BusylightSDK.js in your App:

```
<script src="BusylightSDK.js"></script>
```

The Busylight is initialized by calling the BusylightSDK function:

```
var busylightSDK = new BusylightSDK(function() {
  // insert your code to run after successful connection here
});
```

Now, you can use the busylightSDK object in your code like

```
busylightSDK.ColorRGB(0,100,0);
```

## BusylightSDK.js Reference

### Available Functions

Function	Description
Color(BusylightColor)	Busylight will show the color defined in the BusylightColor object.
ColorRGB(red, green, blue)	Busylight will show the color defined by the RGB color parts. The Range for the color is 0..100.
Alert(BusylightColor, BusylightSoundclip, BusylightVolume)	Busylight will alert with BusylightColor object, Busylight SoundClip



Blink(BusylightColor,ontime,offtime)	Busylight will blink.OnTime and OffTime are in 0.1 second steps.
Pulse(BusylightPulseSequence)	Busylight will Pulse, defined in the BusylightPulse Object.
ColorWithFlash(BusylightColor, BusylightColor)	Busylight will Show the first color with short flashes in the second color.

### Available Ringtones

BusylightSoundclips	ID
OpenOffice	1
Quiet	2
Funky	3
FairyTale	4
KuandoTrain	5
TelephoneNordic	6
TelephoneOriginal	7
TelephonePickMeUp	8
IM1	9
IM2	10

### Loudness

BusylightVolume	ID
Max	100
High	75
Middle	50
Low	25
Mute	0

You can find definitions and declarations as well in the BusylightSDK.js file.

### Known issues to be fixed in the next release:

- Adding / Removing the Busylight Device during running app
- Only a single Busylight can be used at one PC.