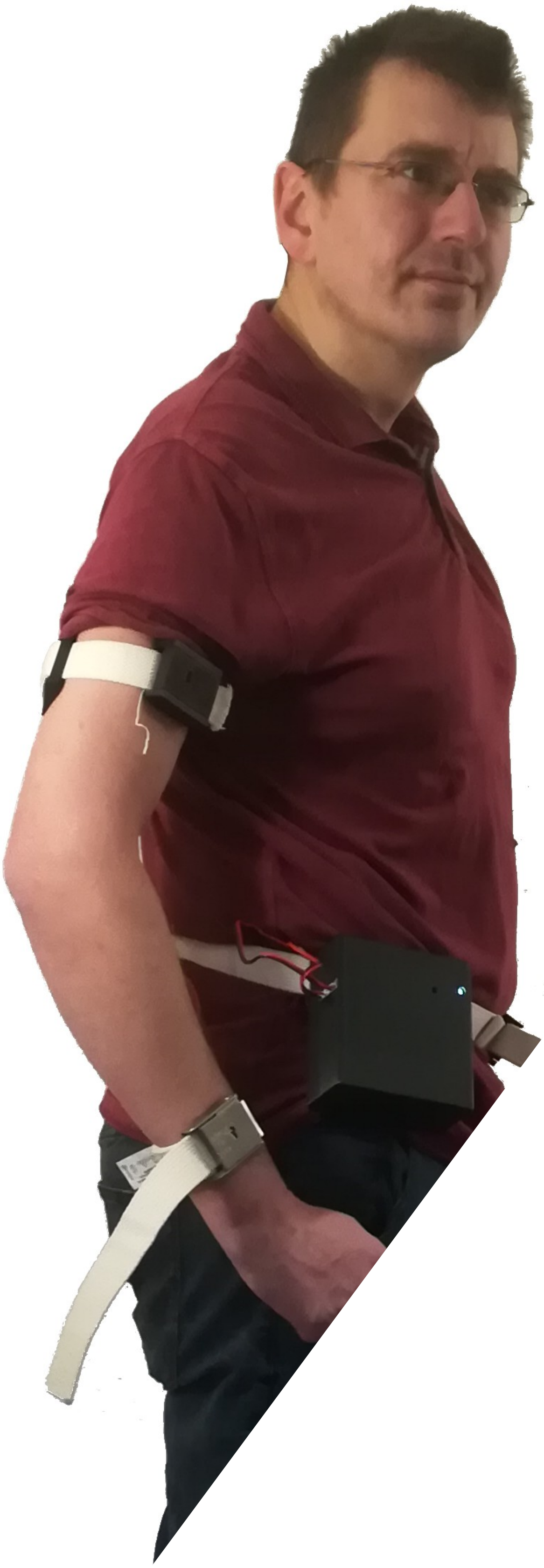
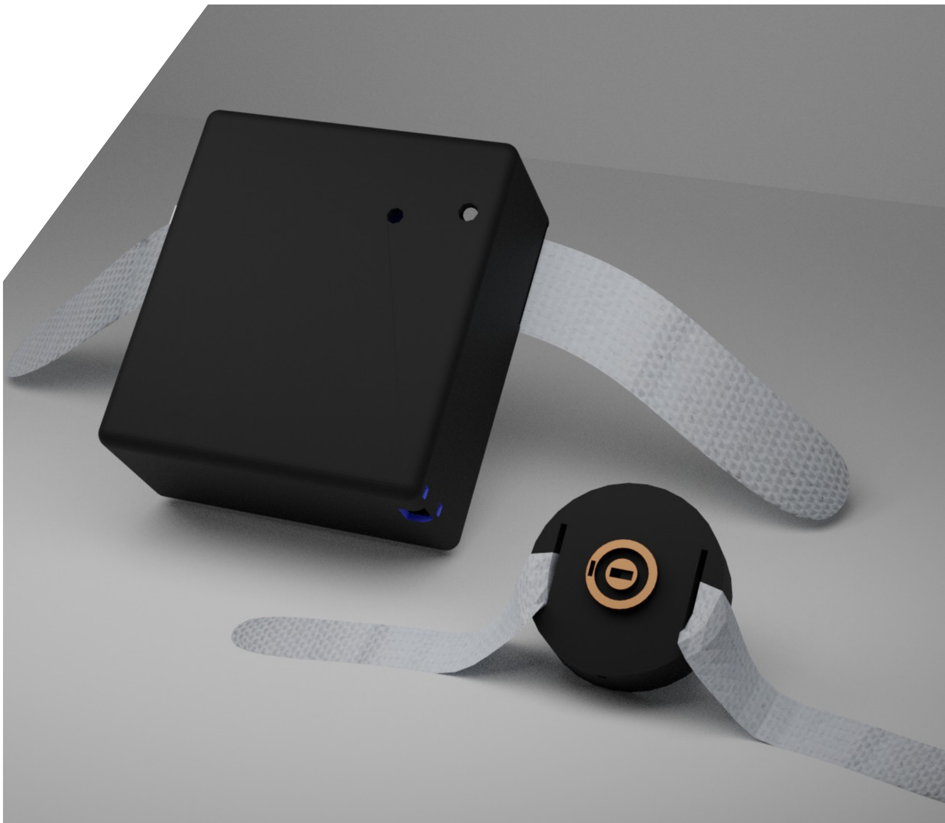


# GED

(Graduated Electronic Decelerator)
electronic behavior correction device



# GED (Graduated Electronic Decelerator)
# electronic behavior correction device

# Manual

This is not the Original GED from the Judge Rotenberg Center (JRC).
It's a improved recreation based on the specifications from JRC.

**Features:**

- 2 Output Channels as H-Bridge (alternating current).
- RF433 Reciver for CarKey-Style Remote.
- WiFi for conficuration and control.
- Up to 3 WiFi SSID for automatic connection to any of them
- Output Voltage 70V – 350V.
- Base Configuration and Charge over Micro USB.
- 12V Li-Ion Akkupack (3-Cells)
- Up to 24 h Operation Time with one Charge.

**Table of Contents:**

**Device Status LED & Sound:**

| **LED Colors** | **Meening** |
|---|---|
| *-static-* | |
| Off | Device Off, Battery is empty or in Reset State. |
| White | Device is Starting ( < 5 Seconds) |
| Violet | Deploy shock to person. |
| | |
| *-cycling-* | A color Change Cycle of 2.5 seconds indicates connection state and Battery Level. |
| | *The First Color Shows the connection state.* |
| Green | Connected to WiFi and IoT-Server |
| Yellow | Connected to WiFi and IoT-Server and Deliver shock if out of WiFi-Area |
| Blue | Connected to WiFi-Only |
| Cyan | No WiFi |
| | *The Second Color shows the Battery Level by Fading* |
| Green | 100% |
| Yellow | 50% |
| Orange | 20% |
| Red | 0% |

**Sound:**

| | |
|---|---|
| 200 ms | Startup Buzzer-Test |
| ~2 sec. | Shock request from Key-Remote. |
| Short seq. | Battery empty |

**Startup:**
before you can use the GED,
you need to make some base configuration.
For this, the GED should be charged fully,
if it's not charged the cable communication could fail because of to low input Voltage as a result of the charging proces.

Connect the GED with a MicroUSB-Cable to your computer, and determinate which SerialPort it is.
On Ubuntu Linux you can do it like this:
**ls /dev/ttyUSB***

Then you connect over the determinated SerialPort with 115200 bit/sec; 8 Bit; no flow control
On Ubuntu Linux you can do it like this:

```
minicom -b 115200 -8 -D /dev/ttyUSB0 -s
```

Then you chose something like "**Serial Port Setup**",
where you **dissable flow-control** and chose **exit**.

You can get a short help from the GED when you send "**?**" followed by Return.
The command "**list**" shows you the current configuration.
A setup workflow could look like this:

```
initConfig

setAP_a_name darkIoT_AP
setAP_a_password darkIoT-trainingCenter_PW
IoTauth darkIoT-trainingCenter1234
IoTserver darkiot.top 8888
devName GED_JhonDoe
conMode_ap_iot
saveConfig
```

Now the WiFi and Network Setup is complete.
In the next step you need to configurate the Remote.
For this the GED need's to be switched on, because of powering the RF-Reciver, and be connected to the GED over the SerialPort.
Determinate the RF-Code is a bit tricky because there is a lot of "noise" on 433Mhz.
Just be near the GED with the Remote an Press the Key you want to use,
then you see some Hex Numbers, press the Key multiple times and have a lock which number repeat's this number's you put into the configuration look like that:

```
118 - 100011000
231E9 - 100011000111101001
-
```

If you think you have the right values, you type something like that, to configurate the "A" sequence:

```
Rfcode_A 118 231E9
```

To Configurate the shockPulse sequence "A" in this example with a total duration of 2000ms (2sec.) an Impulse of 20 microseconds and a Pause of 80 microseconds you Type:

```
cfgShock_A 2000 20 80
```

**Operation:**

- If Belt and Switch-Cover is in the Main-Device, remove them
- Turn the Power-Switch on the Rear-side (in the Belt-Guide) to I.
- Put the Power-Switch-Cover on the Power-Switch
- Feed the lockable Belt through the Belt-Slot of the Device
- Attach the Main-Unit to the Students waist
- Attach the Electrodes to the Students body, ensure that the wires are not limiting the movement.
- With the Key-Remote you can chose which electrodes should be used.
  A = Pair 1, B = Pair 2, 3 = Both

Operation over WiFi:
When you use the WiFi and IoT-Server, you are able to interconnect Environment- and/or Body-Sensors to deliver shocks automatically.

Examples:
Rocking and/or leaving the chair.
Not being in bed at bedtime, for a longer period of time.
Running/wild activity.
Leaving WiFi Area (leaving the Building)

**WiFi Communication Protocol:**
To be able to Address the target device in the right way,
It's not allowed that the deviceID contains Space
The Data Record for communication with an IoT-Server locks like the following:

*Send from GED:*

```
<device Type>,<device ID>,<Account ID>,<data>
```

Example:

```
“GED v3,GED_JhonDoe,darkIoT-trainingCenter1234,”
```

*Send to GED over IoT-Server:*

```
IoTControler,<senderID>,<AccountID>,sendto <Target DeviceID>
<command>
```

Example:

```
“IoTControler,Alexa_Interface,darkIoT-trainingCenter1234,sendto
GED_JhonDoe shockA”
```

Send to GED direct:
Get the DeviceIP. It is listen on Port 8888.
Transmit just the command sequence
*Linux Example*:

```
echo “<command>” > cmdfile.txt
netcat 192.168.1.250 8888 < cmdfile.txt
```