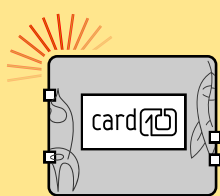




# state LED

# parts



- no contact, please!
- chaos
- communication
- camp

[https://card\[badge\].badge.events.ccc.de/personal-state](https://card[badge].badge.events.ccc.de/personal-state)

## Need any help? Get in touch with us!



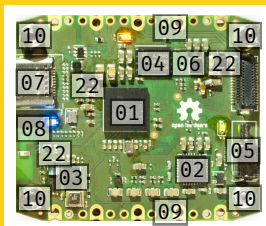
[https://card\[badge\].badge.events.ccc.de/](https://card[badge].badge.events.ccc.de/)

@card[badge]badge@chaos.social

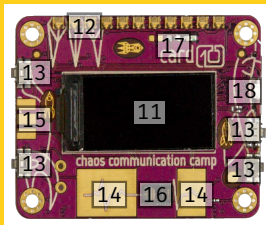
twitter.com/card[badge]badge

freenode.com#card[badge]badge

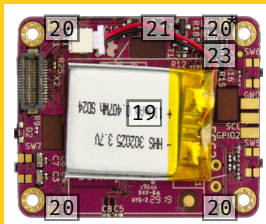
analog state indicator



fundamental



harmonic (front)

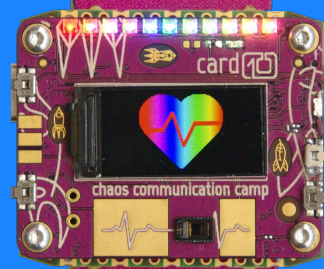


harmonic (back)

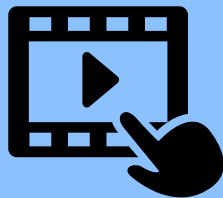
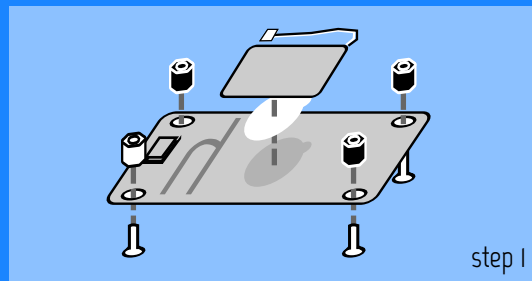
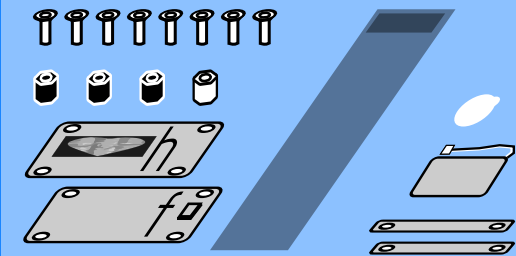
- CPU MAX32666 (dual core Cortex-M4F)
- ECG MAX30001
- Position sensor with magnetometer BHI160+BMM150, Accelerometer BMA400, Environmental sensor (temperature, humidity, air pressure) BME680
- External flash for file storage
- Vibraton motor
- Power Management (PMIC) MAX77651
- USB-C connector (with ECG contacts)
- BLE antenna
- Wristband connectors
- ECG wrist contacts
- 80x160 pixel display
- RGB LEDs
- Buttons
- ECG finger contacts
- SAO V1.69bis connector
- Pulse sensor MAX86150 (2nd ECG chan)
- Light sensor / 940 nm IR LED
- White side LED
- 3.7V, 200 mAh rechargeable LiPo battery
- RGB LEDs (\*:state LED)
- Transistors
- Coils
- I2C Portexpander PCAL6408



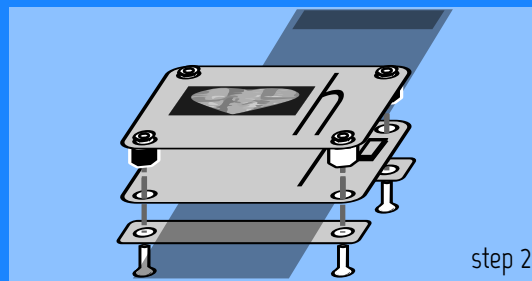
# card[badge] quick reference



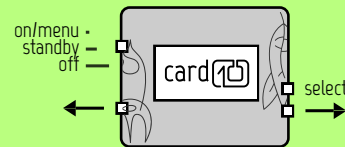
# assembly



[https://card\[robot\].badge.events.ccc.de/vid](https://card[robot].badge.events.ccc.de/vid)



# quick start



[https://card\[robot\].badge.events.ccc.de/app](https://card[robot].badge.events.ccc.de/app)

**main.py**

```
import display, light_sensor, color, leds
import utime
```

```
FADE_LIMIT = 15
HUE_STEP = 5
SLEEP_TIME = 0.2
hue = 0
light_sensor.start()
disp = display.open()
```

```
while True:
    sensor_value = light_sensor.get_reading()
    disp.clear()
    disp.print("sensor {}".format(sensor_value))
    disp.update()
    if sensor_value < FADE_LIMIT:
        hue = (hue + HUE_STEP)%360
        led_color = color.from_hsv(hue, 1, 1)
        leds.set_all([led_color]*15)
        utime.sleep(SLEEP_TIME)
```

save  
on USB  
drive:

