Security sub-system

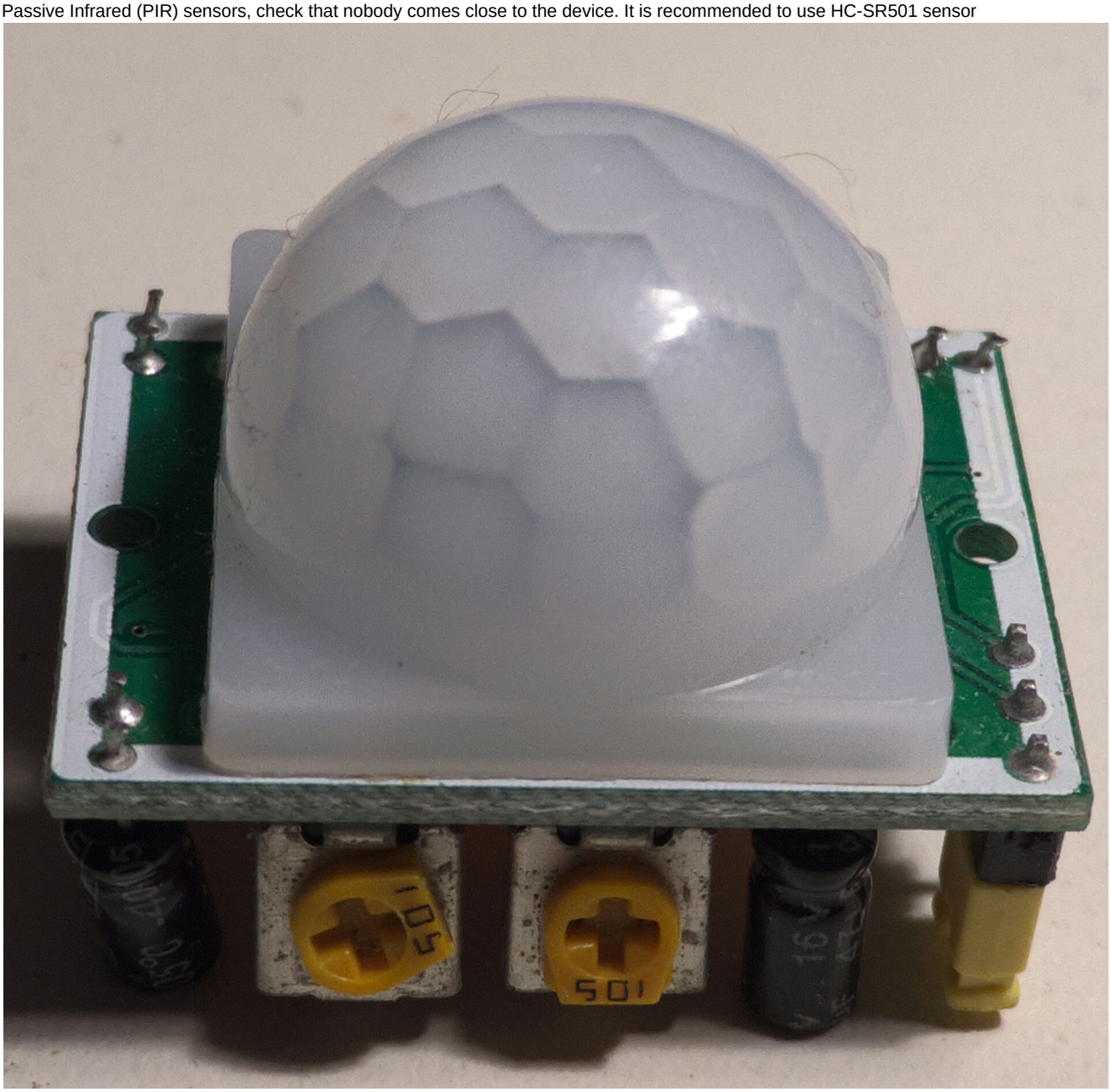
The security sub-system allows control of the environment where the Kraken device is working.

Hardware components

(you can use the simulator to avoid any hardware components. see section ‘Configuration’)

The typical system contains the following components

1. “Reed Switch” on the box cover which checks that the lid is not opened, Any kind of **normally closed** reed switch can be used.
2. Passive Infrared (PIR) sensors, check that nobody comes close to the device. It is recommended to use HC-SR501 sensor



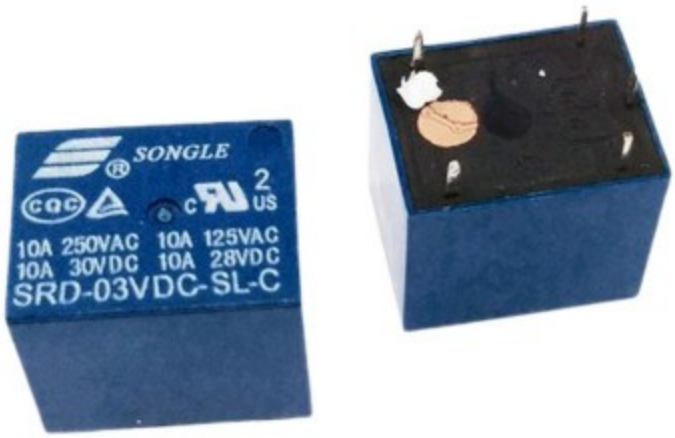
Settings:

- Jumper is recommended to switch to H position (but it is not obligatory)
- The left switch is responsible for time detection adjustment. It is recommended to turn it fully counterclockwise
- The right switch is for sensitivity adjustment. You should adjust it for specific environmental conditions

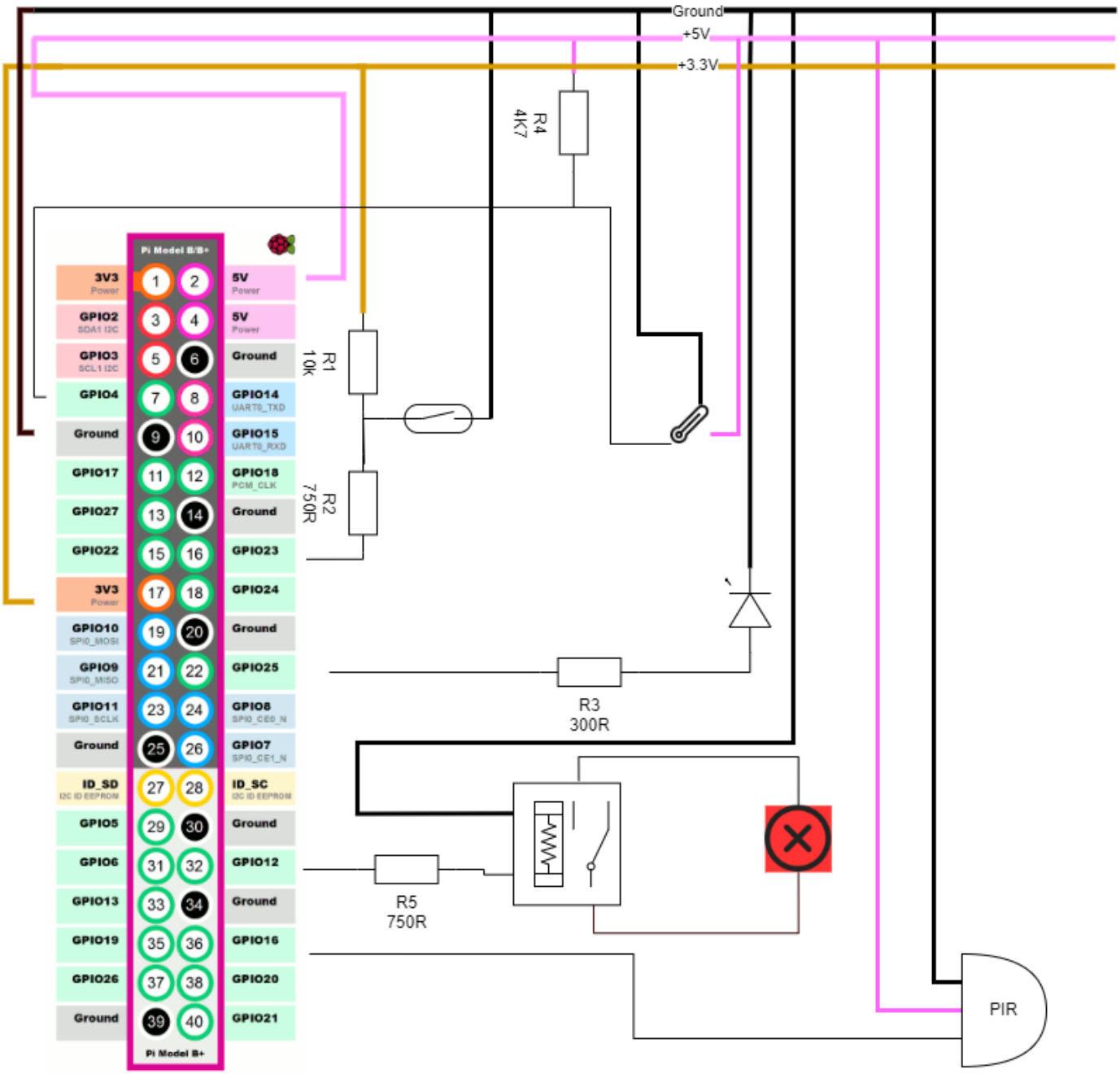
3. Thermometer sensor DS18B20. You can use any kind of thermosensor:

4. LED indicator - any kind of LED for 3.3V

5. Relay for an emergency switch - any kind of relay driven by 3.3v. I.e. SRD-03VDC-SL-C



The schema of commutation:



All resistor values are approximate and may vary. No requirements for the precision and power of the resistors.

You can add an extra sensors the same type.

Configuration

Prerequisites:

- node.js is installed
- Node modules should be installed (npm install):
  - onoff
  - ws
- The hardware schema is configured and attached to the gpio (the minimal configuration is: thermometer DS18B20)
- sudo modprobe w1-gpio && sudo modprobe w1_therm && ls -l /sys/bus/w1/devices/ You should see something, like
- hyppo@hyppo:/ $ ls -l /sys/bus/w1/devices/
  total 0
  lrwxrwxrwx 1 root root 0 Jan 22 17:09 28-0516b4211eff -> ../../../devices/w1_bus_master1/28-0516b4211eff

lrwxrwxrwx 1 root root 0 Jan 22 16:17 w1_bus_master1 -> ../../../devices/w1_bus_master1
Please, copy the string marked red and green. The temperature sensor is connected incorrectly when you don’t see this device. You can connect as many devices to the same bus as you needed or you can add another 1-wire bus.

- Add a line at the end of /boot/config

```
[all]

dtoverlay=w1-gpio,gpiopin=4
```

- reboot the board

Configuration

Download **sensors.js** and **config.json** from the repository.

config.json describes all components of the security sub-system. The simulator can be configured there:

```
{
        "sensors": [{
                "name":"ReedSwitch",
                "type": "contactor",
                "port" : "23",
                "levelOn" : "low",
                "debounceTimeout" : "50",
                "simulator" : "0"
        },
        {
                "name": "LED Blink",
                "type" : "LED_Blink",
                "port" : "25",
                "period": "1000",
                "simulator" : "0"
        },
        {
                "name": "PIR 1",
                "type" : "PIR",
                "port" : "16",
                "timeCalibration": "10000",
                "timeRestore" : "6000",
                "simulator" : "0"
        },
        {
                "name" : "T1",
                "type" : "ds1820",
                "bus" : "w1",
                "address" : "28-0516b4211eff",
                "interval" : "1000",
                "simulator" : "0"
        },
        {
                "name" : "Emergency Switch",
                "type" : "relay",
                "port" : "12",
                "levelOn" : "high",
                "simulator" : "0"
        }
        ],
        "logFile" : "/share/0/RPi_Sensors/sensors_log.txt",
        "wsUrl" : "ws://3.64.54.137:4003/",
        "apiKey" : "220769CL-Sigma"

}
```

Common (root parameters)

logFile : path to the log file. Can be skipped. /var/log/sensors.log is used by default

wsUrl: URL of websocket server

apiKey: your API key

sensors - the list of used sensors:

Parameter	Description	Default	Required for Real	Required for simulator
name	The name of sensor should be unique.	NoName	Yes	No
simulator	Is this sensor real (0) or simulated (1)	0	No	Yes
type	Type of sensor, The following types are supported: ‘contactor’ - some contact, like reed switch, button ‘relay’ - any kind of relay ‘LED_Blink’ - LED which blinks with period ‘PIR’ - Passive Infrared sensor ‘ds1820’ - thermosensor DS18B20		Yes	Yes
port	(for all, except ds1820) Port number, where this sensor is connected. Please, pay attention - it is GPIO number, it is not a pin number. I.e. the contactor on schema upper, is connected to the GPIO23 on pin 16. You should set “port”:”23”		Yes (for all, except ds1820)	No
levelOn	(for contactor and relay) The level for switch unit to “On” state. Onle ‘high’ and ‘low’ values are allowed	‘low’	No	No
debounceTimeout	The debounce time (in msec), to avoid a false positive reactions	100	No	No
period	Blink period for LED (in msec)	1000	No	No
timeCalibration	(For PIR only) Calibration time for PIR sensor. The sensor require this time for warm up	10000	No	No
timeRestore	(For PIR only) The PIR sensor require some time to restore after motion detected	6000	No	No
bus	(For ds1820 only) The 1-wire bus where the thermosensor is connected. See green highlights in pre-requisites.	w1	No	No
address	(For ds1820 only) The address of thermo. See red highlights in pre-requisites.		Yes	No
interval	(For ds1820 only) The interval of polling the thermosensor (in msec)	1000	No	No

Start

run

$ node sensors.js

API

The system sends to the server (via websocket) the state of sensor when the value is changed:

Example

```
{
        "apikey": "220769CL-Sigma",
        "sensors": {
                "tStamp": 1674408301123,
                "T1": "17.5"
        }
}
```

The main payload is `"T1": "17.5"` - the sensor, named “T1” send the message “17.5”

The server can send a command to the security subsystem:

```
{
... Any elements here...
        "sensors": {
                "Emergency Switch": "off",
                "second switch": 1
        }
        ... Any elements here...
}
```

These commands will be applied to the ‘relay’ unit only. The following commands are allowed:

- 0
- 1
- on
- off