

Топпер КСО

1. Определение в системе.

Топпер подключается к целевой машине по USB, определяться как USB-COM устройство и имеет соответствующую ноду устройства в /dev.

Пример:

```
/dev/ttyACM33
```

Пример:

```
ACTION=="add"          KERNEL=="ttyACM*",          ATTRS{product}=="SuperTopper",  
ATTRS{idVendor}==0403,          ATTRS{idProduct}==6001,  
SYMLINK+="ttyACM33_SuperTopper", MODE="0666"
```

(Значения 0403 и 6001 могут изменяться, лучше выносить их в конфиг).

2. Программное взаимодействие и набор команд

Устройство принимает на вход код цвета в hex формате и меняет свой цвет соответствующим образом. Скорость установки цвета: не более чем за 1 секунду (для корректной работы режима "моргания"). Вызывающая сторона конвертирует String с hex кодом в byte[] и отправляет на устройство вызовом метода OutputStream.write(byte[] b).

Цвет	Hex
Черный	000000000
Желтый	255190000
Синий	000000255
Красный	255000000
Зеленый	000128000

Таблица 1. Распределение hex кодов по цветам

Пример программного взаимодействия с устройством:

```

public class SuperTopper {
    private CommPort commPort;
    private CountingOutputStream countingOutputStream;

    public void connect() {
        System.out.println("Connecting to topper device");

        try {
            CommPortIdentifier portIdentifier = CommPortIdentifier.getPortIdentifier("/dev/ttyACM33");
            if (portIdentifier.isCurrentlyOwned()) {
                throw new Exception(String.format("Topper's port (%s) is currently in use", portName));
            }

            commPort = portIdentifier.open(this.getClass().getName(), 2000);
            if (!(commPort instanceof SerialPort)) {
                throw new Exception(String.format("Topper's port (%s) is not serial...", portName));
            }

            SerialPort serialPort = (SerialPort) commPort;
            serialPort.setSerialPortParams(9600, SerialPort.DATABITS_8, SerialPort.STOPBITS_1, SerialPort.PARITY_NONE);
            countingOutputStream = new CountingOutputStream(serialPort.getOutputStream());

            System.out.println("Successfully connected to topper device");
        } catch (Error | Exception e) {
            throw new Exception(String.format("%s occurred when connecting to topper (%s): %s", e.getClass(), portName,
e.getMessage()));
        }
    }

    public void disconnect() {
        if (countingOutputStream != null) countingOutputStream.close();
        if (commPort != null) commPort.close();

        System.out.println("Successfully disconnected from topper device");
    }

    protected void send(String colorHex) {
        try {
            countingOutputStream.write(colorHex.getBytes());
            Awaitility.await().atMost(1000 TimeUnit.MILLISECONDS)
                .until(() -> countingOutputStream.getCount() == colorHex.getBytes().length);
        } catch (IOException | ConditionTimeoutException e) {
            System.out.println("Cannot send color {} with blink to topper ({}): {}", getColorByHex(colorHex), portName,
e.getMessage());
        } finally {
            System.out.println("Color changed to {}", getColorByHex(colorHex));
            System.out.println("Bytes to write: {}, bytes wrote: {}", colorHex.getBytes().length, countingOutputStream.getCount());
            countingOutputStream.resetCount();
        }
    }
}

```