



Internet stvari s RPi-om i Javom Internet of things with RPi and Java

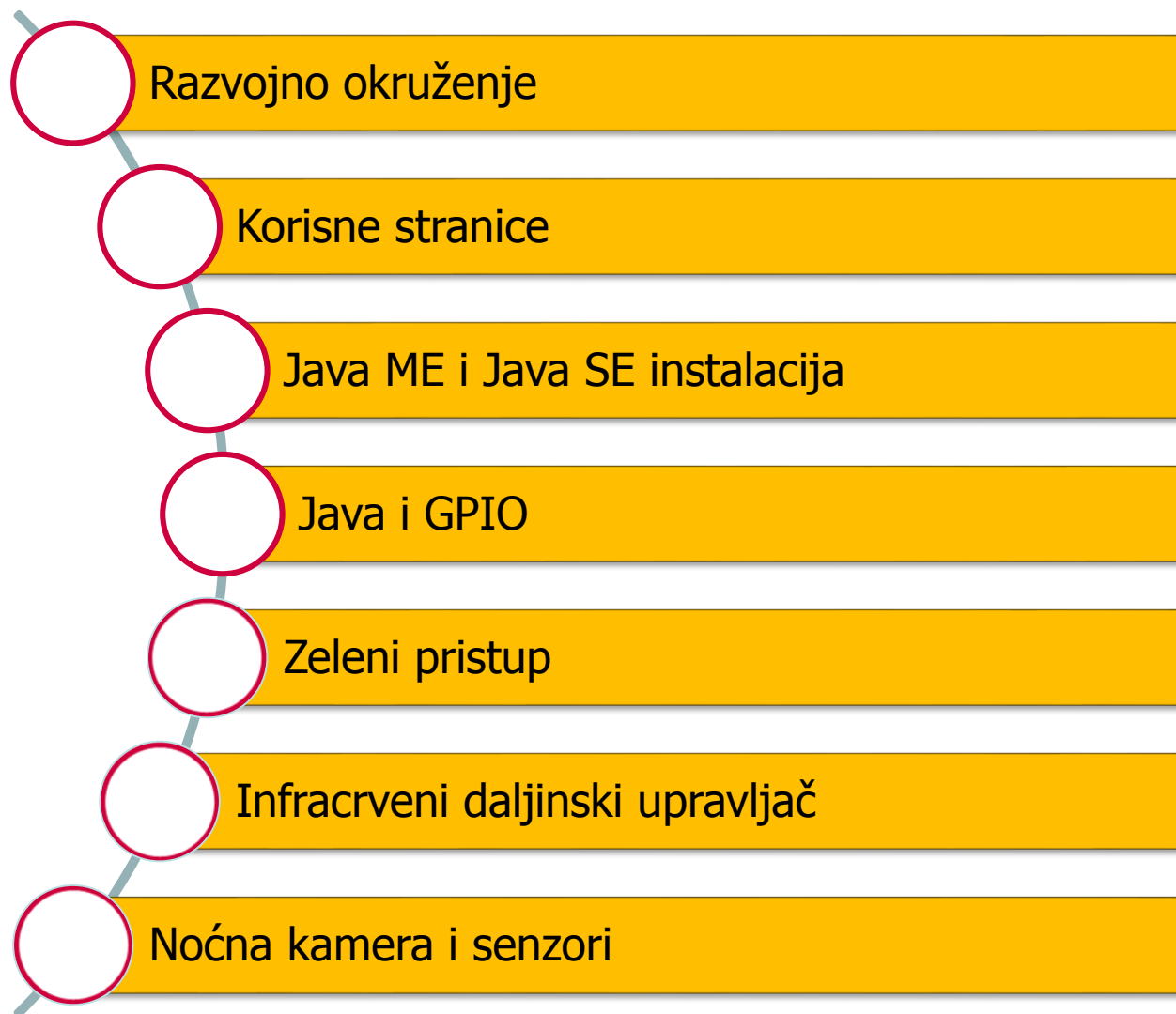
Matija Novak, mag. Inf.
prof. dr. sc. Dragutin Kermek

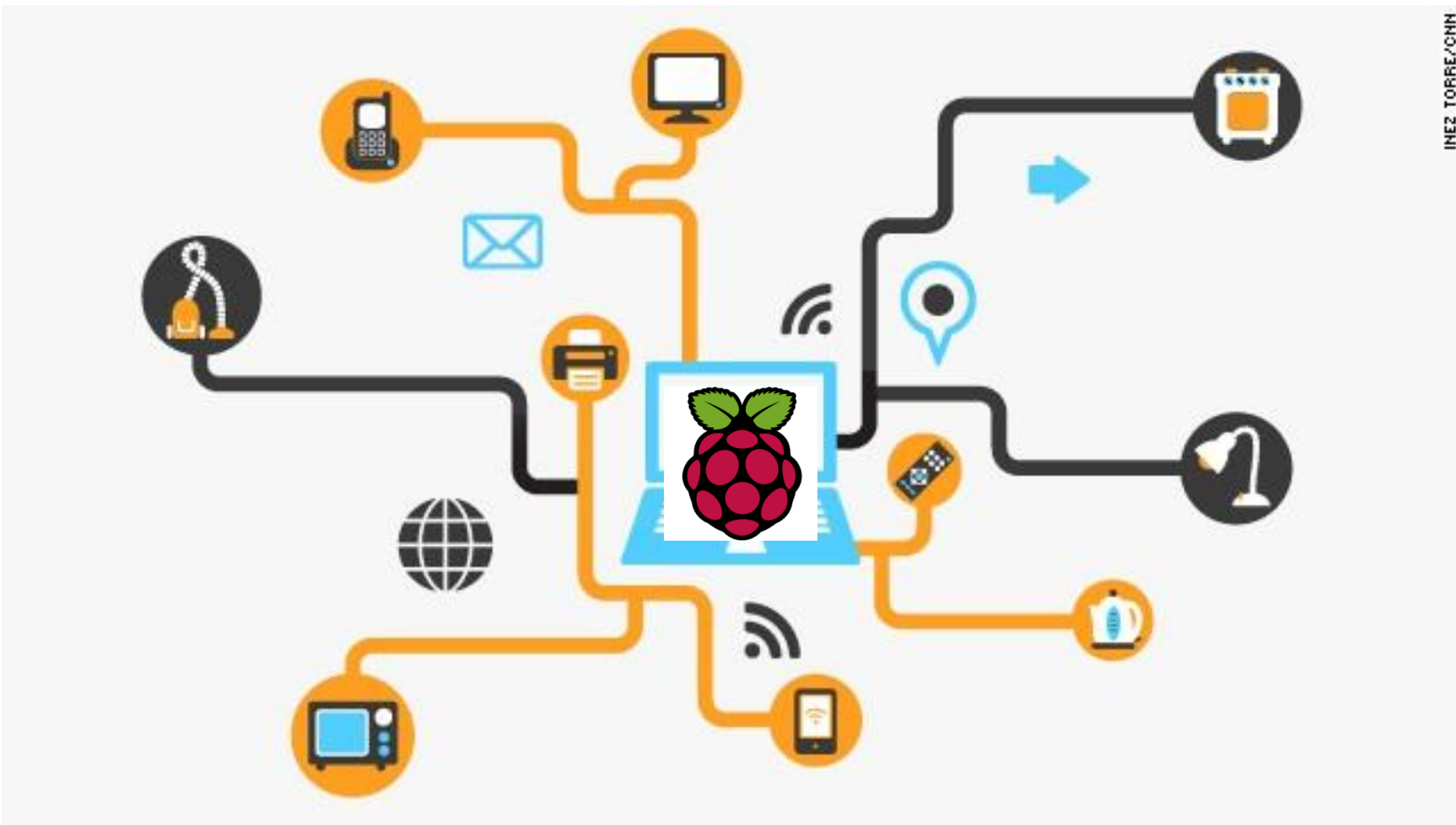


University of Zagreb
Faculty of Organization and Informatics
Pavlinska 2, 42000 Varaždin, Croatia
<http://www.foi.unizg.hr/>

JavaCro'15 10-12 Svibnja 2015

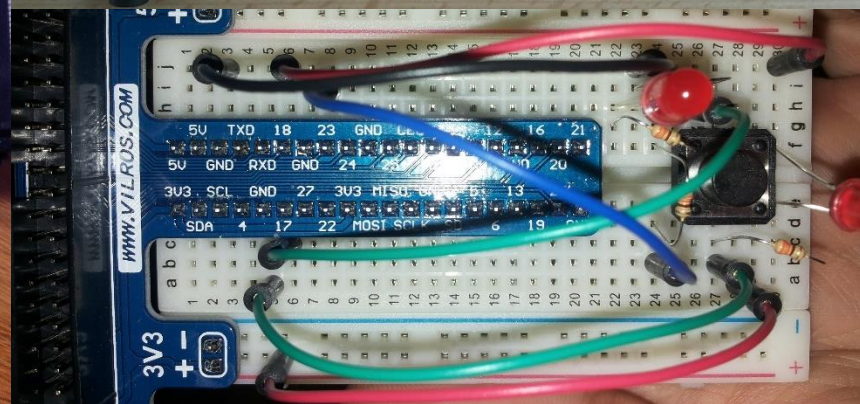
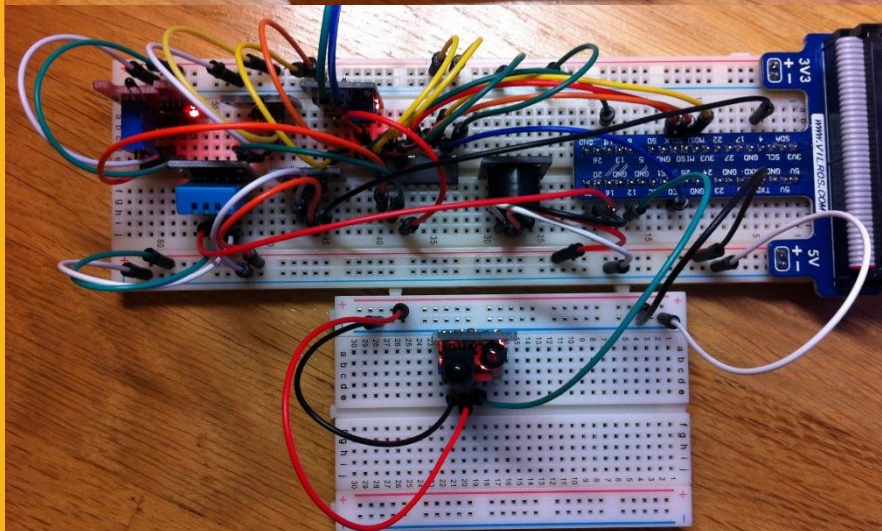
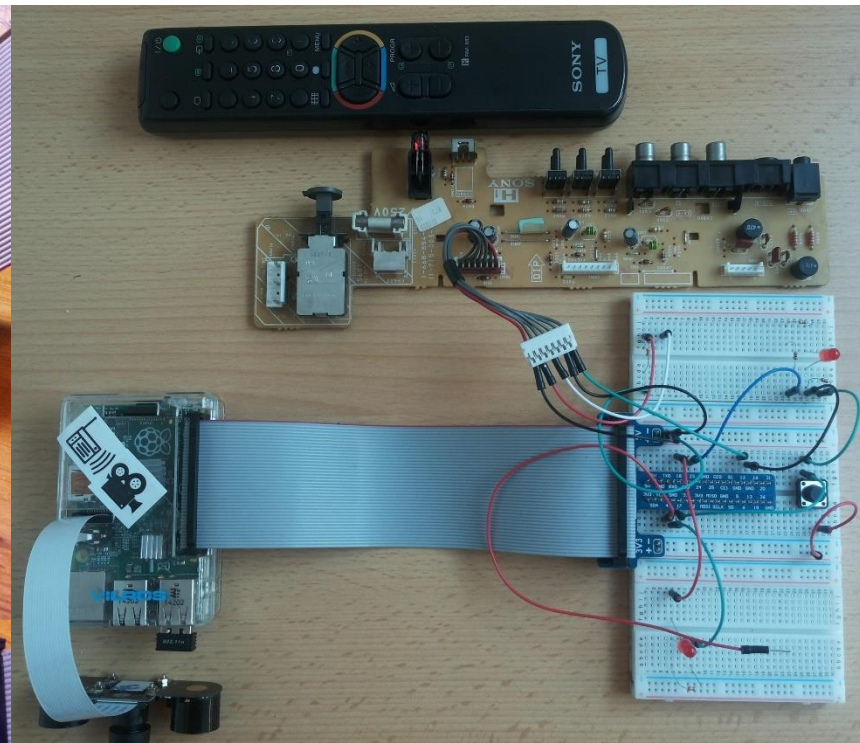
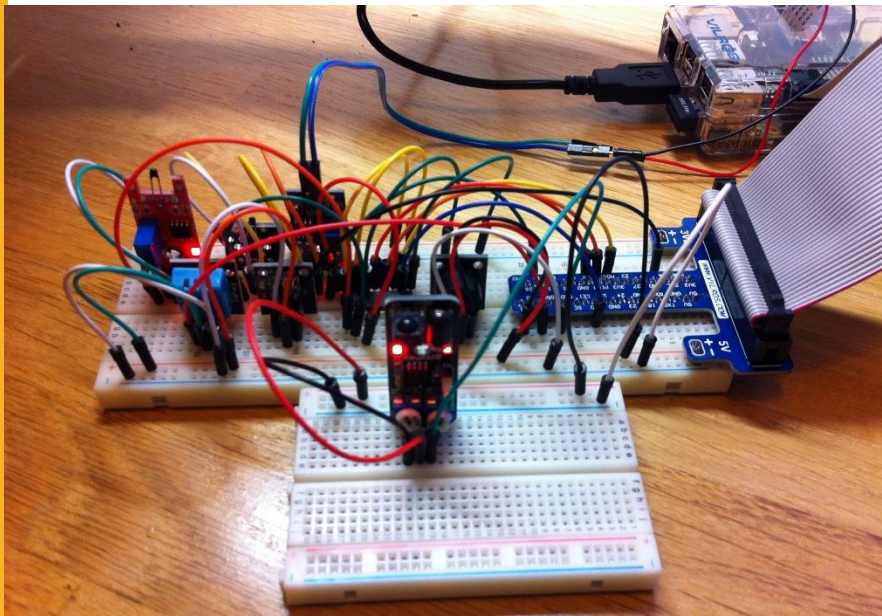
- Za provjeru koncepta Interneta stvari primjenom RPi-om kreirana je Java SE aplikacija koja omogućuje upravljanje RPi kamera tako da se uključuje putem raznih senzora (svjetlost, kontakt i sl). Prikazana je priprema razvojne okoline, razvoj i testiranje aplikacije. Osim kupljenih elemenata posebnost razvijenog rješenja je tzv. zeleni pristup.
- This paper shows the proof of concept for Internet of Things using the RPi. It consists of Java SE app that allows controlling RPi camera using a variety of sensors (light, touch, etc.). It shows the preparation of the environment, development and testing. In addition to the purchased elements specialty of this solution is so called green approach.





INEZ TORRE/CHN

<https://flipboard.com/@stogner/ieee-internet-of-things-rsqdr8tjy>

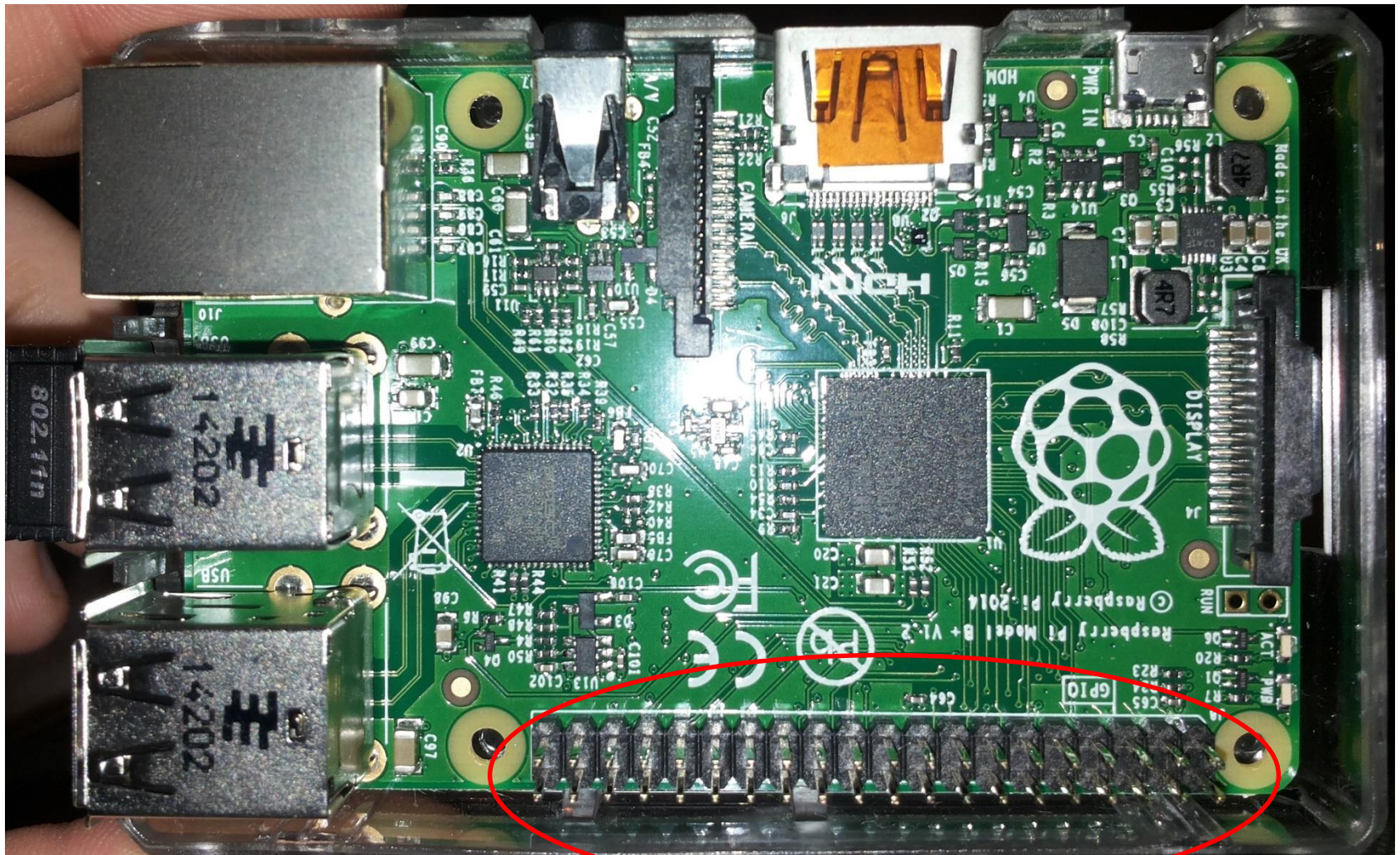


- Windows 8.1
 - Netbeans 8.0.2
 - Udaljena platforma (eng. Remote platform)
 - Java SE 8u40 – [jdk1.8.0_40](#)
 - Java ME SDK 8.1 – [jmesdk-8-1](#)
- Raspberry Pi – Model B+
 - Oracle Java ME Embedded 8.1 for Raspberry Pi Model B (ARM11/Linux) – [jmee-8-1](#)
 - Linux ARM v6/v7 Hard Float ABI – [jdk1.8.0_06](#)
 - NoIR camera
- Java SE Embedded vs Java ME Embedded
 - Runtime -> exec()
- Pločica za prototipiranje, kratkospojnici, razni senzori, Sony TV-H1 ploča

- Java ME Embedded MOOC: Installing the Java ME 8 Embedded SDK and NetBeans – [YouTube](#)
- Running Imlets on Raspberry Pi using the Command shell - [doc](#)
- Raspberry Pi – Installing Oracle Java SE 8 – Developer Prview for ARM – [page](#)
- Using Device I/O with Java Embedded Suite on a Raspberry Pi – [blog](#)
- Using Oracle Java SE Embedded Support in NetBeans IDE – [page](#)
- Working with GPIO by Using Java ME Embedded and a Raspberry Pi – [page](#)
- JavaMagazine
 - „JavaFX and Near Field Communication on the Raspberry Pi" Java Magazine, March/April 2014
 - "The Device I/O API," Java Magazine, November/December 2014
 - "Code Java on the Raspberry Pi," Java Magazine, November/December 2014

- Skinuti [javaME](#) i kopirati na Raspberry uređaj
- Ekstrakcija:
 - `sudo unzip oracle-jmee-8-1-rr-raspberrypi-linux-bin.zip -d /usr/lib/jvm/javame8`
- Test:
 - `cd /usr/lib/jvm/javame8`
 - `sudo ./usertest.sh`

- Skinuti [jdk1.8.0_06](#) i kopirati na Raspberry uređaj
- Ekstrakcija:
 - `sudo tar zxvf jdk-8u6-linux-arm-vfp-hflt.tar.gz -C /usr/lib/jvm`
- Instalacija alternativa:
 - `sudo update-alternatives --install /usr/bin/javac javac /usr/lib/jvm/jdk1.8.0_06/bin/javac 1`
 - `sudo update-alternatives --install /usr/bin/java java /usr/lib/jvm/jdk1.8.0_06/bin/java 1`
- Ažuriranje alternativa ručno:
 - `sudo update-alternatives --config java`
 - `sudo update-alternatives --config javac`
- JAVA_HOME varijabla okruženja (iskoristi željeni editor poput „nano”)
 - Dodaj u „/etc/environment”: `JAVA_HOME="/usr/lib/jvm/jdk1.8.0_06,,`
 - Dodaj u "`~/ .bashrc,,`:
 - `export JAVA_HOME="/usr/lib/jvm/jdk1.8.0_06"`
 - `export PATH=$PATH:$JAVA_HOME/bin`



GPIO

- Java SE Embedded treba dio.jar i Raspberry mora imati „dio dev“ direktorij:

```
hg clone http://hg.openjdk.java.net/dio/dev  
(JavaMagazine, November/December 2014 - The Device I/O API)
```

- Kreirati novi Java projekt
- Kreirati gpio.policy u korijenskom direktoriju projekta

```
grant {  
    permission jdk.dio.gpio.GPIOPinPermission "*:*";  
    permission jdk.dio.DeviceMgmtPermission "*:*",  
                                                "open";  
};
```

- Import dio.jar preuzeti iz /dev/build/jar

```
import jdk.dio.gpio.GPIOPin;  
import jdk.dio.gpio.GPIOPinConfig;  
  
GPIOPinConfig pinConfig = new  
GPIOPinConfig(  
    DeviceConfig.DEFAULT,  
    18,  
    GPIOPinConfig.DIR_OUTPUT_ONLY,  
    GPIOPinConfig.MODE_OUTPUT_PUSH_PULL,  
    GPIOPinConfig.TRIGGER_NONE, false);
```

```
GPIOPin pin = null;
    pin = (GPIOPin)
    DriverManager.open(GPIOPin.class,
    pinConfig);
//GPIOPin pin = (GPIOPin)
DeviceManager.open(18);
System.out.println("Blinking LED");
pin.setValue(true);
Thread.sleep(500);
pin.setValue(false);
pin.close();
```

- Pokrenuti kao root korisnik sa VM opcijama:
 - Djdk.dio.registry=/home/pi/dev/config/dio.properties-raspberrypi
 - Djava.security.policy=/home/pi/JavaProjects/Pi/gpio.policy
 - Djava.library.path=/home/pi/dev/build/so

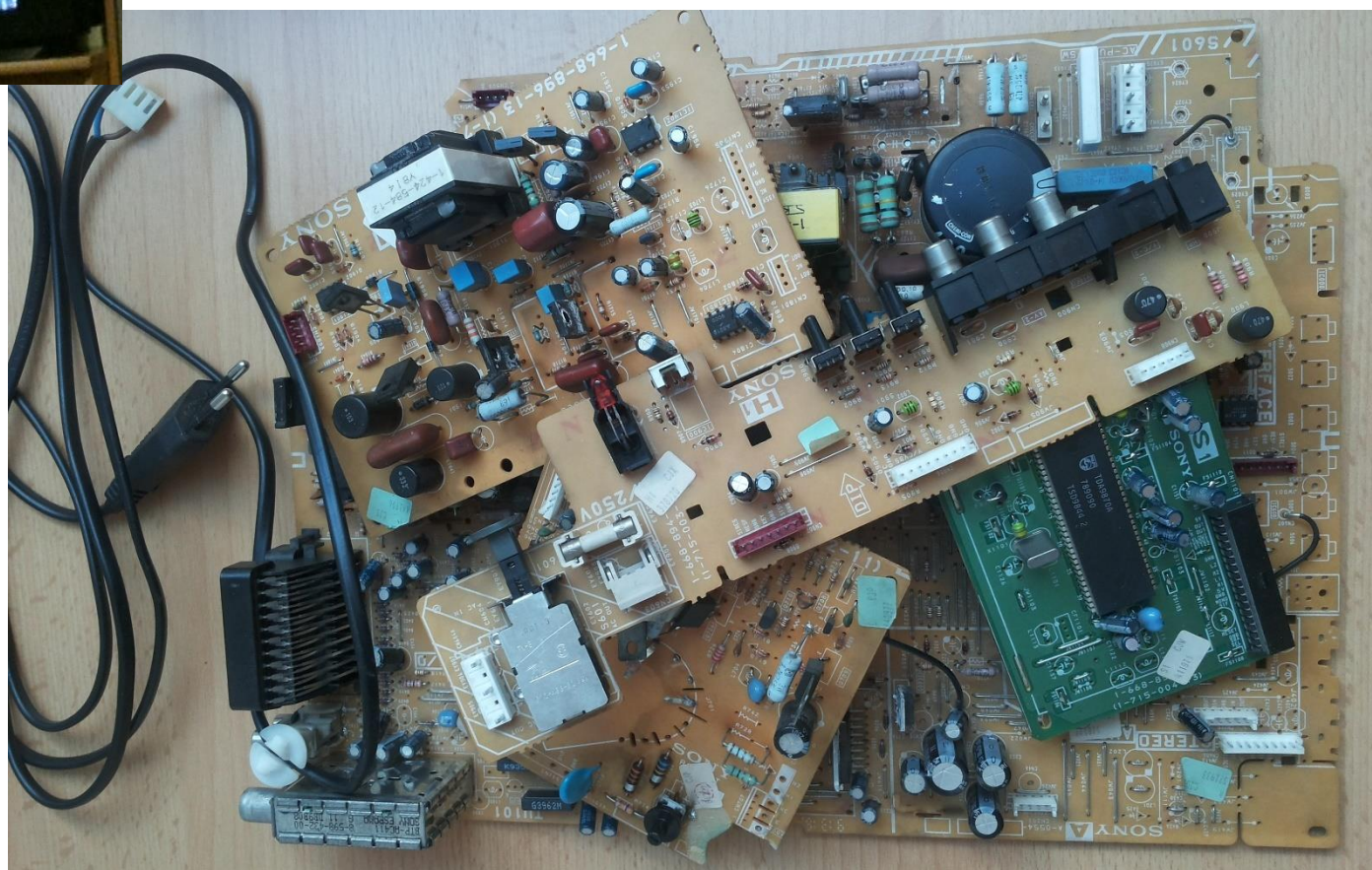
- Kod postavljanja i pokretanja preko NetBeans-a GPIO može se desiti da se ne napravi unexport GPIO
 - U Java programu uhvatiti grešku i zatvoriti korišteni GPIO pin sa `pin.close()`;
 - ```
#include <stdio.h> #include <fcntl.h>
int main(void){
 int pin=18; char buffer[4]; ssize_t bytes_written;
 int fd;
 fd = open("/sys/class/gpio/unexport", O_WRONLY);
 if (-1 == fd) {
 fprintf(stderr, "Failed to open unexport!");
 return(-1);
 }
 bytes_written = snprintf(buffer,4, "%d", pin);
 if(-1 == write(fd, buffer, bytes_written)){
 printf(„ALREADY UNEXPORTED!!„);
 }
 close(fd);
 return(0);}
```

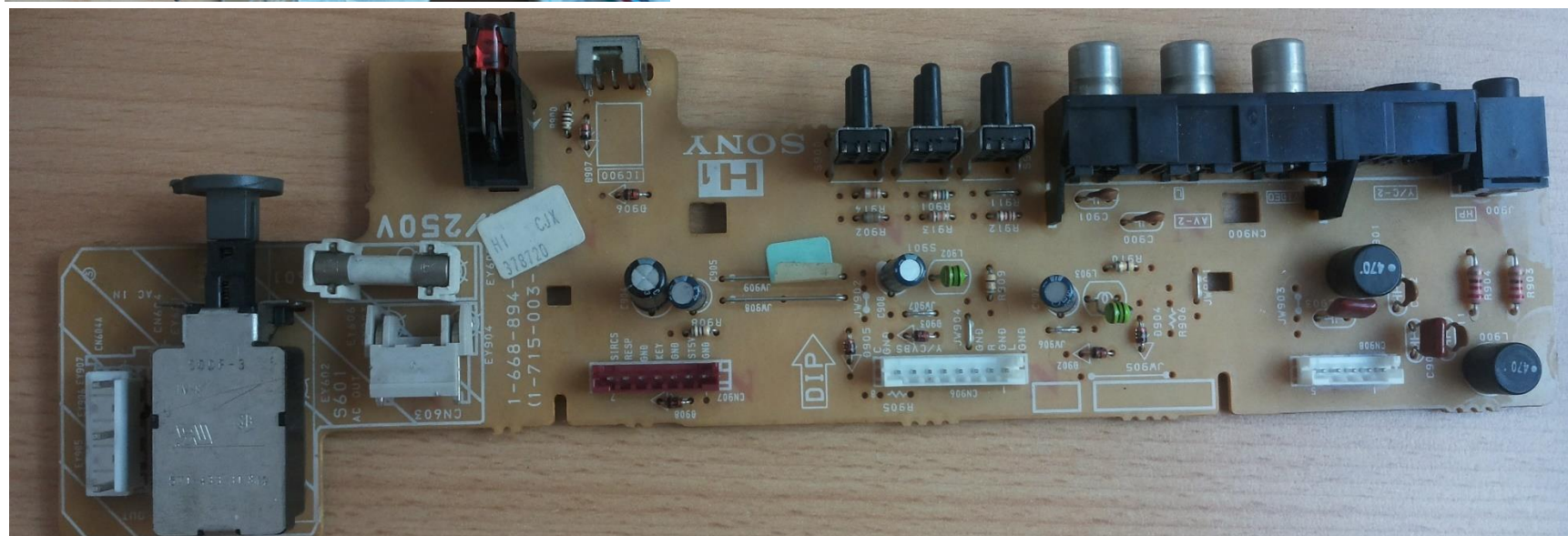
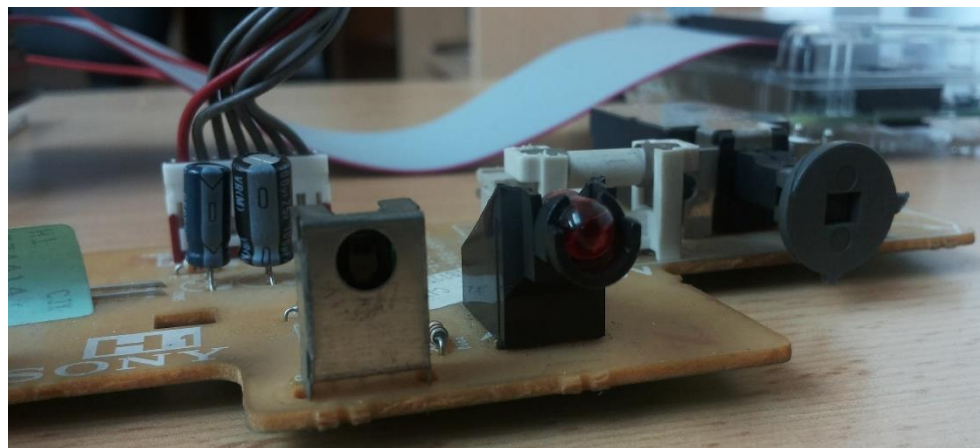
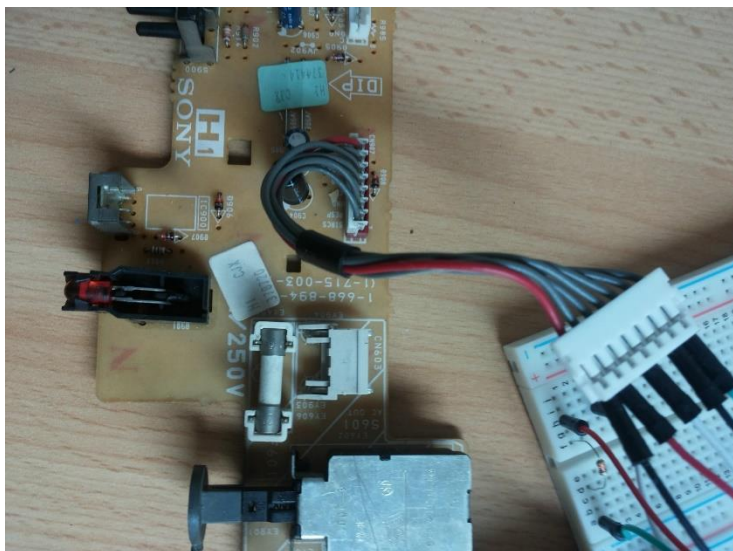


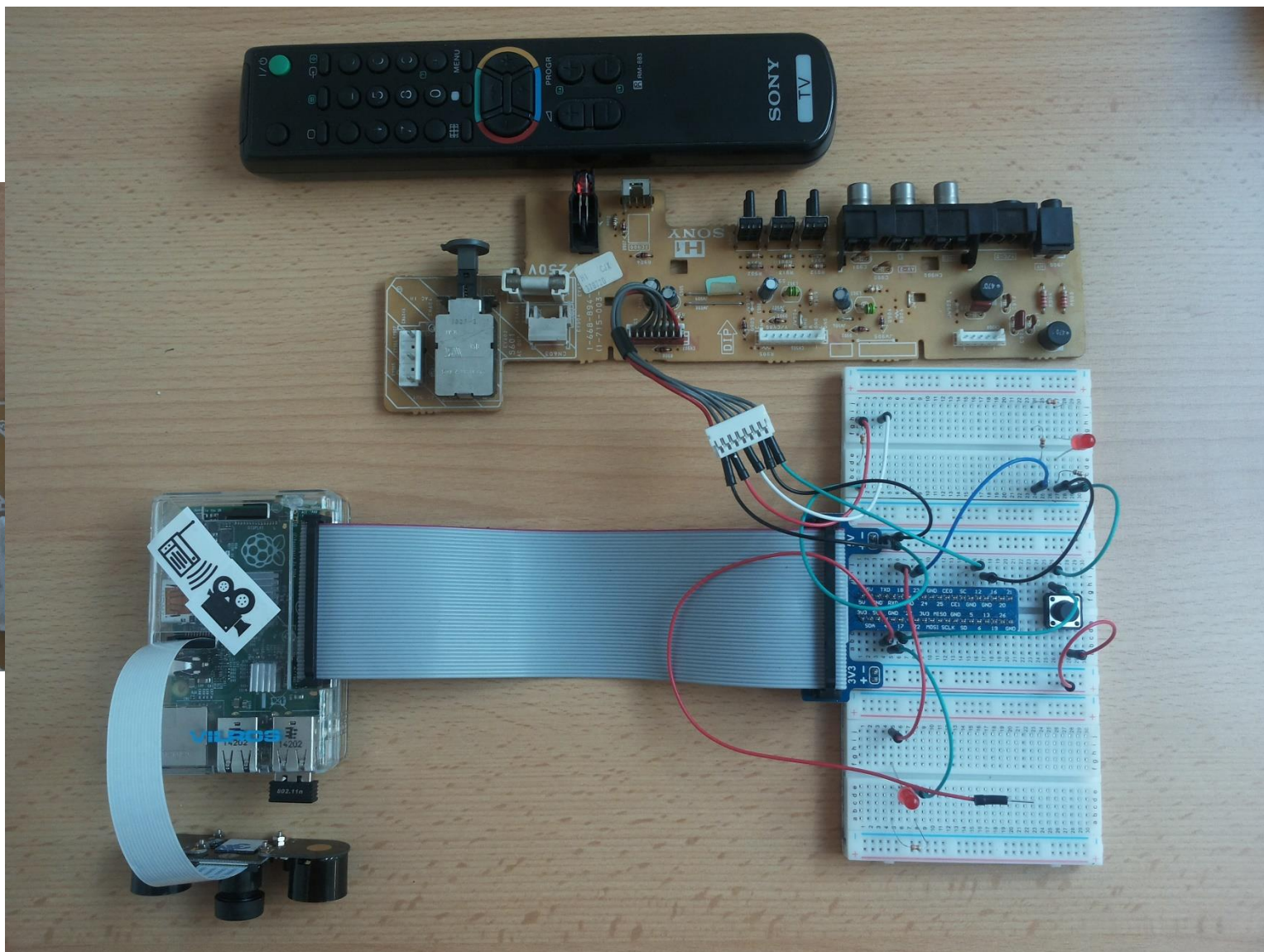
The „green” way to dump electronic junk - [članak](#)











- <http://www.lirc.org/> i <http://jlirc.sourceforge.net/>
- Vodiči
  - How to Control the GPIO on a Raspberry Pi with an IR Remote – [page](#)
  - Raspberry Pis, Remotes & IR Receivers! – [blog](#)
- Potrebno je napraviti build na Raspberry Pi uređaju da bi se mogao koristiti jlirc:
  - Promijeniti „makefile”
    - Change JDK\_HOME to your JDK dir
    - Remove in gcc line: „-marsh=1386”
  - Promijeniti u UnixSocketImpl.java (ukoliko nije dostupna standardna putanja)
    - `libpath = UnixSocketImpl.class.getResource(libname).getPath();`
    - To
    - `libpath = System.getProperty("user.dir")+File.separator+libname;`
- Kopirati libunixsocket.so u korijenski direktorij projekta

- `sudo apt-get install lirc`
- **Test:**
  - `irrecord --list-namespace | grep KEY`
  - `sudo kill $(pidof lircd)`
  - `mode2 -d /dev/lirc0`
- **Konfiguriranje daljinskog upravljača:**
  - `sudo kill $(pidof lircd)`
  - `irrecord -d /dev/lirc0 ~/lircd2.conf`
  - `sudo mv lircd2.conf /etc/lirc/lircd.conf`
  - `sudo /etc/init.d/lirc restart`
  - `irw`
- <http://lirc.sourceforge.net/remotes/sony/>

- Kreirati command.lirc dokument u korijenskom direktoriju projekta:

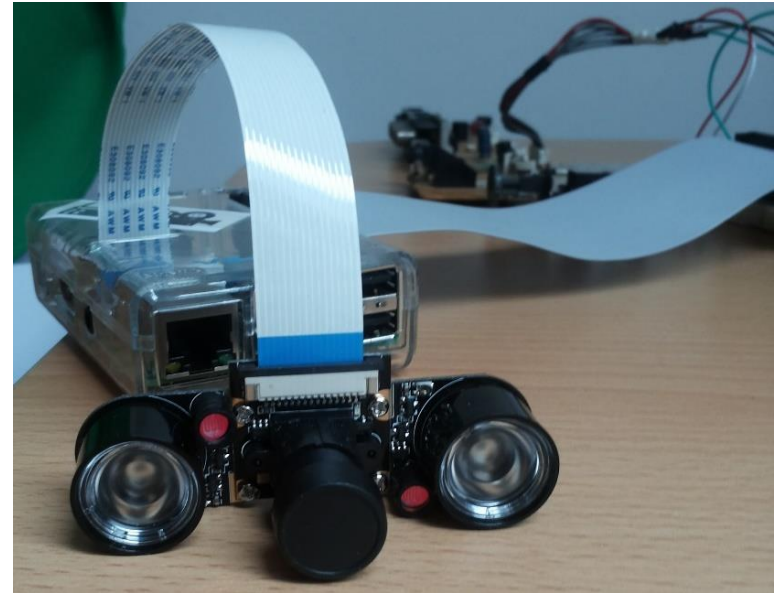
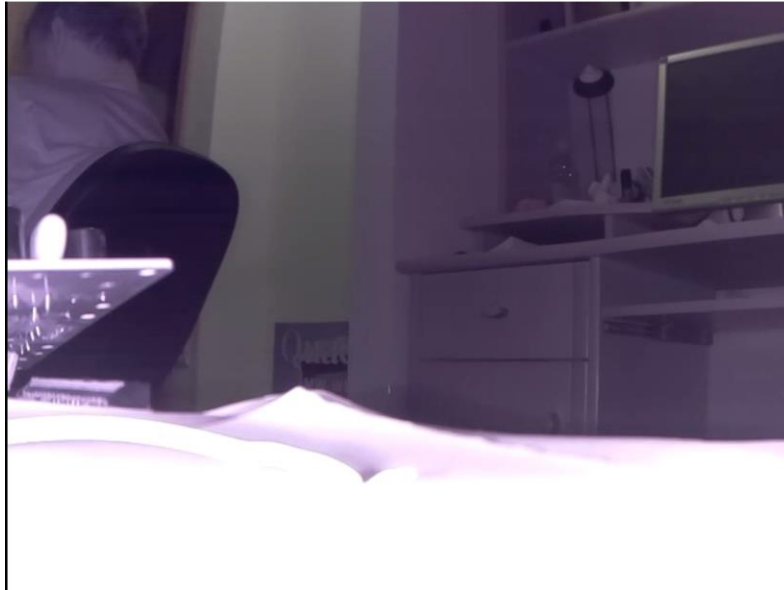
```
begin
 button = KEY_MUTE
 config = mute
 repeat = 1
End
```

```
private static SimpleLIRCClient client;
client = new
SimpleLIRCClient("/home/pi/JavaProjects/Pi/command.
lirc");
client.addIRActionListener(new ButtonListener());
```

```
private static class ButtonListener
implements IRActionListener {
 ...
 public void action(String command) {
 System.out.println(command); //npr. mute
 Date now = new Date();
 long interval = now.getTime()-
 last_time.getTime();
 last_time=now;
 System.out.println(interval);
 if(interval<(long)3000)
 { return; }
 ... GPIO dio kako bi zasvijetlila LED dioda kad se
 pritisne tipka ...
 if(command.equalsIgnoreCase(„mute“))
 { ... }
 }
}
```



- Kućni (noćni) nadzor



- Building an action camera using a Raspberry Pi and Java by Franco Raimondi – [blog](#)

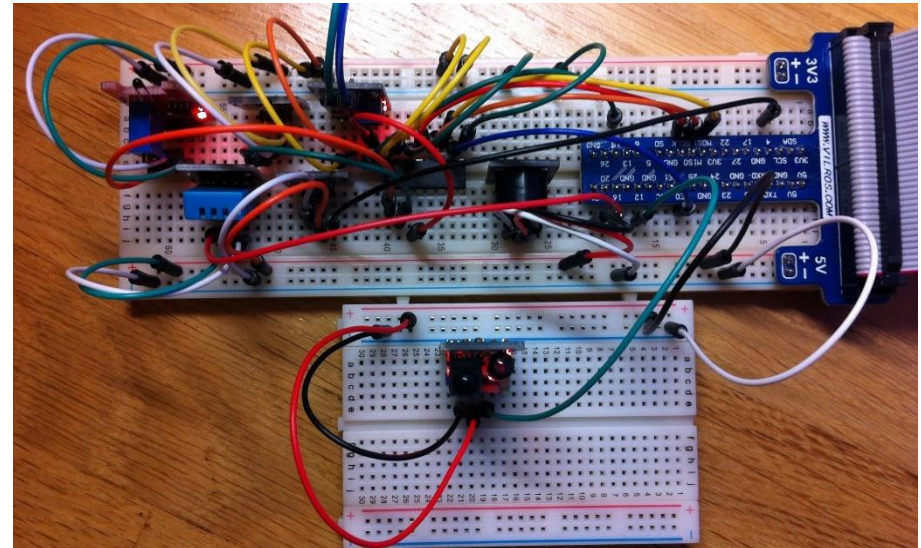
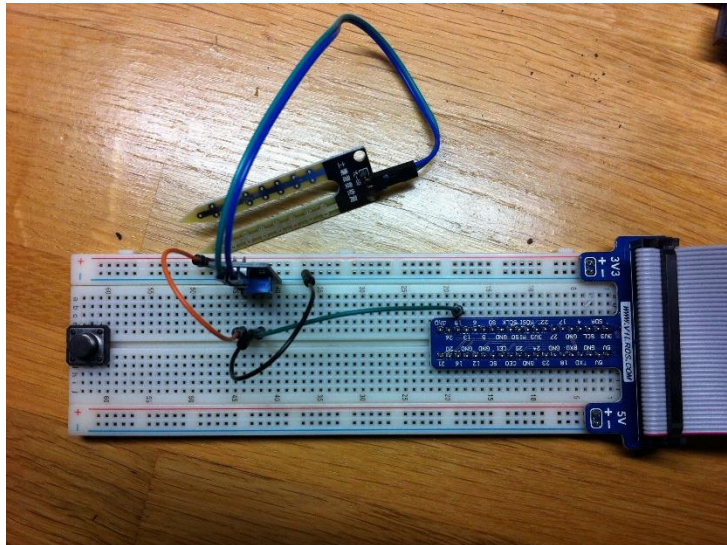
- Preuzeto sa: [blog](#)
- ```
raspivid -o - -t 99999999 -w 800 -h 600 --hflip --vflip | cvlc -vvv stream:///dev/stdin --sout '#standard{access=http,mux=ts,dst=:8080}':demux=h264
```
- ```
raspivid -o - -t 99999999 -w 800 -h 600 --hflip --vflip | tee test_video.h264 | cvlc -vvv stream:///dev/stdin --sout '#standard{access=http,mux=ts,dst=:8080}':demux=h264
```
- Uključivanje/isključivanje LED diode – koristiti GPIO 5

```
private void executeCommand(String cmd) {
 Runtime r = Runtime.getRuntime();
 Process p = null;
 p = r.exec(cmd);
 InputStream stdout = p.getErrorStream();

 /* Scanner scanner = new
 Scanner(stdout);
 while (scanner.hasNextLine()) {
 System.out.println(scanner.nextLine()+"\n");
 }*/
}
```

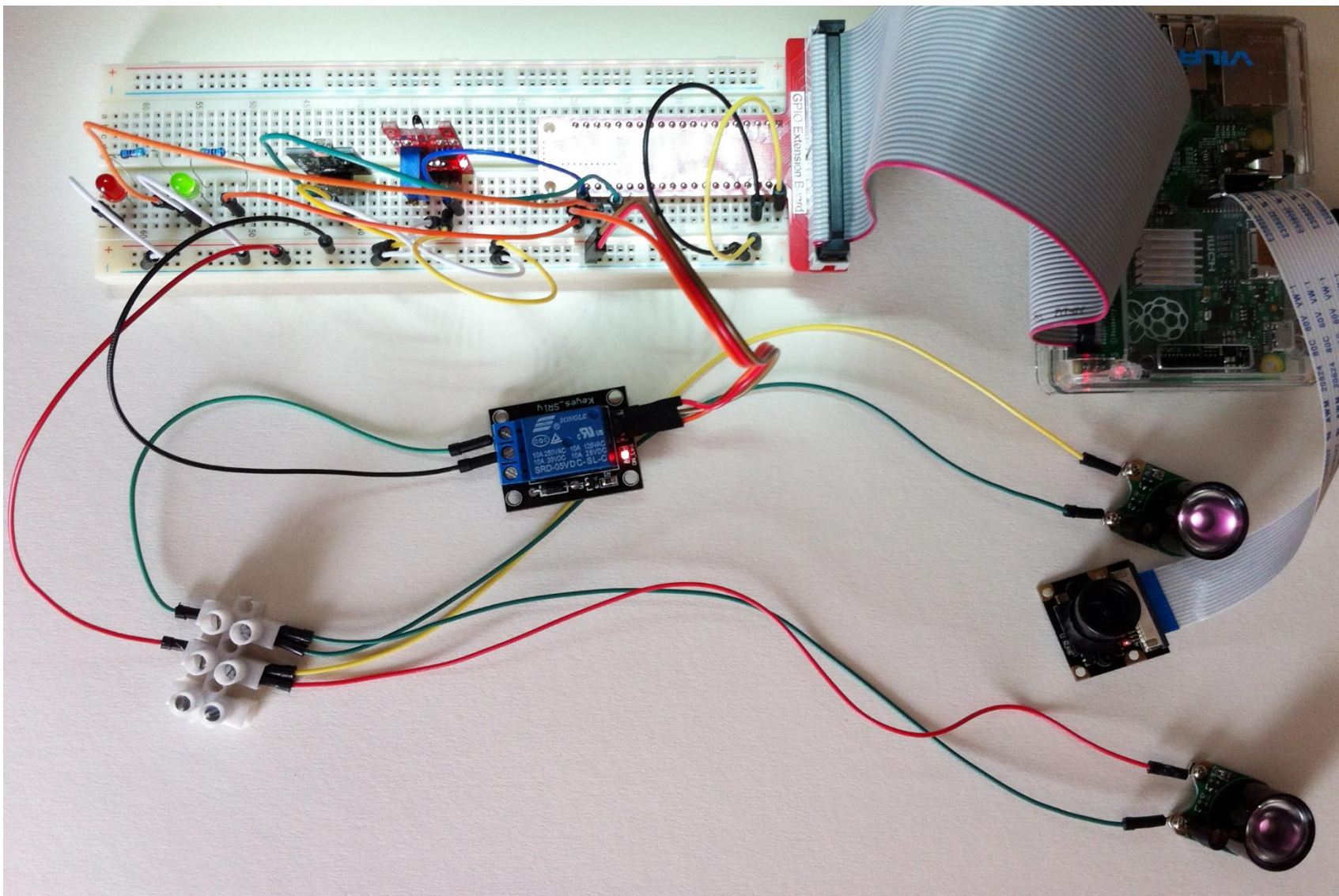
- Glavni razlog za korištenje Java SE Embedded

- Svi senzori korišteni su preko GPIO sučelja
- Kontrolirani su preko daljinskog upravljača (paljenje/gašenje)
- Korišteni senzori/dijelovi:
  - Senzor za vlažnost tla
  - Senzor za temperaturu, Senzor za temperaturu i vlažnost zraka
  - Laser i senzor za laser (prekidom se uključuje buzzer)
  - Buzzer i A-D konverter (za dobivanje točnog postotka vlage tla)



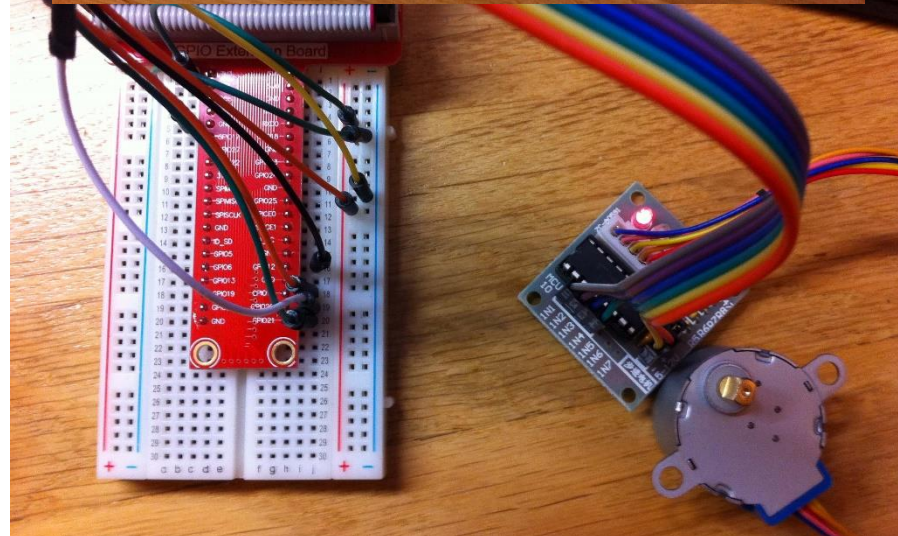
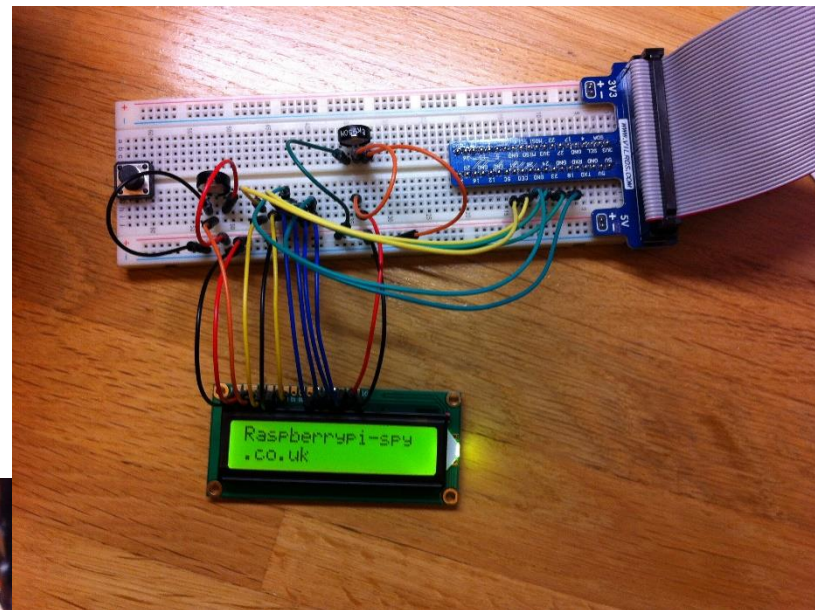
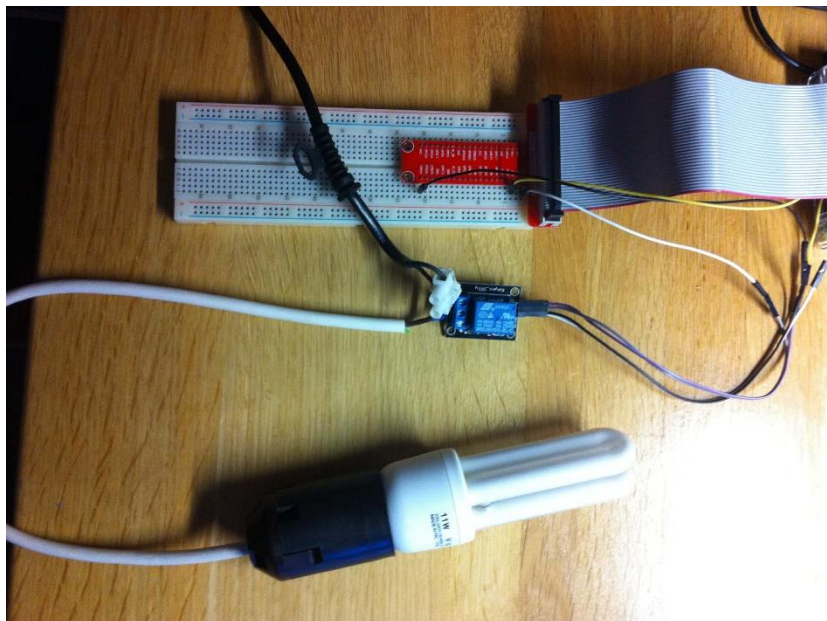


- Problem zagrijavanja dioda za osvjetljavanje kod noćne kamere
- Rješenje:
  - staviti hladnjake
  - upravljanje kamerom i IR osvjetljivačima na bazi releja, senzora svjetla, senzora temperature i LED dioda (zelena i crvena).
    - Senzoru temperature potenciometrom se podešava prag kada se daje signal na GPIO portu
    - Zelena LED svijetli kada su uključeni IR osvjetljivači
    - Crvena LED svijetli kada je temperatura previsoka pa su ugašeni IR osvjetljivači
    - Ni jedna LED ne svijetli ako je dovoljno svjetla pa nisu potrebni IR osvjetljivači
- Budući rad: napraviti kućište



- Opcija 1: Potrebno je startati java program kod paljenja Raspberry Pi uređaja
  - `cd /etc/init.d/`
  - `sudo nano autostartMyScript.sh`
    - `#!/bin/sh`
    - `sh /home/pi/my_autostart.sh > /home/pi/my_autostart.log`
  - `sudo chmod +x autostartMyScript.sh`
  - `sudo update-rc.d autostartMyScript.sh defaults`
- `my_autostart.sh` – sadrži java komandu za startanje java aplikacije
- Opcija 2: kreirati cron job





- Java ME
  - Bolji za početnike
  - Direktno korištenje GPIO
- Java SE napredniji
  - Prava pristupa GPIO
  - Veće mogućnosti (Runtime)
- Java + RPi + zeleni pristup = jeftina automatizacija doma

# Hvala vam na pažnji!

## Pitanja?

Matija Novak, mag. Inf. – [matnovak@foi.hr](mailto:matnovak@foi.hr)

prof. dr. sc. Dragutin Kermek – [dkermek@foi.hr](mailto:dkermek@foi.hr)



University of Zagreb

Faculty of Organization and Informatics

Pavlinka 2, 42000 Varaždin, Croatia

<http://www.foi.unizg.hr/>

---