

Tipkam na banane

Prejšnjič smo veliko programirali. Pisali smo zapletene programe, ki so si morali zapomniti, do kje so prišli (recimo pri prižiganju diod) in čez koliko časa morajo prižgati ali ugasniti naslednjo.

Danes bomo veliko manj programirali in se veliko več igrali.

Izpisovanje z Arduinom

Ko kak program ne deluje, se ga lahko lotimo z razhroščevalnikom. Če ne veste, kaj je to, nič hudega, saj ga za Arduino tako ali tako ni (vsaj ne v brezplačnih orodjih, kolikor vem). Sicer pa tudi napake v programih v Pythonu pogosto iščem kar tako, da na kakem strateškem mestu kaj izpišem — recimo trenutno vrednost kake poredne spremenljivke..

Na Arduinu danes ne bomo izpisovali le zato, ker bi na ta način iskali napake (no, lahko tudi), temveč da bomo lažje opazovali, kaj se dogaja.

Hm, izpisovali - na kaj? Lahko na Arduino mar priključimo monitor? Težka bo, saj ima - poleg kupa za to neuporabnih pinov - le USB. V resnici ga ne moremo. Pač pa imamo na Arduino že priključen pravi računalnik, ne?

V programu, s katerim programiramo Arduino, v meniju Tools izberi Serial Monitor. Kar bomo izpisali z Arduinom, se bo izpisalo v to okno. Preskusi tale program.

```
void setup() {
  Serial.begin(9600);
  Serial.println("Jaz sem Arduino...");
}

void loop() {
  Serial.println("... ki mu je dolgčas");
  delay(2000);
}
```

S `Serial.begin(9600);` povemo, da bomo uporabljali USB za "pogovor" med Arduinom in računalnikom. Številka 9600 pove hitrost, s katero se bosta pogovarjala. Ne spreminjaj je; 9600 bo ravno prav.

Nato z ukazom `Serial.println` izpisujemo besedila in številke. Če izpišemo besedilo, ga moramo dati v dvojne narekovaje. Številke pa izpisujemo brez njih, recimo

```
Serial.println(42);
```

Izpisujemo lahko tudi vrednosti spremenljivk, na primer

```
int nekaj = 15;  
Serial.println(nekaj);
```

Ker je `nekaj` spremenljivka, je ne damo v narekovaje. Če bi namreč napisali

```
Serial.println("nekaj");
```

bi se izpisalo besedilo `nekaj` namesto vrednosti spremenljivke `nekaj`.

Ukaz `Serial.println` izpiše besedilo tako, da se naslednji izpis nadaljuje v novi vrstici. Če želimo, da bo naslednji izpis v isti vrstici, namesto njega uporabimo `Serial.print`. Če imamo, recimo, temperaturo, shranjeno v spremenljivki `temp`, bi lahko napisali

```
Serial.print("Danes je ");  
Serial.print(temp);  
Serial.println(" stopinj.");
```

Upori in napetosti

Zdaj potrebujemo malo fizike oziroma elektronike. Predstavljajte si, da enega za drugim (zaporedno, kot bi učeno rekli) vežemo pet uporov za, recimo, 1 kiloohm. En konec "verige" priključimo na pozitivni, drugega na negativni pol 1,5 voltne baterije.

- Vzamemo voltmeter in pomerimo napetost. Recimo, da res namerimo 1,5 volta.
- Zdaj pomerimo napetost na enem polu baterije. Vem, da se sliši neumno. Pravzaprav celo upam, da se vam ne bo zdelo niti vredno preskusiti. Seveda bomo namerili 0 voltov.
- Kaj pa, če poskusimo nekje vmes? Če je med enim in drugim koncem 1,5 volta, med enim in istim koncem pa 0 voltov, koliko je na dveh petinah poti? Koliko bomo namerili, če merimo napetost prek dveh uporov? Če se zdi komu logično, da dve petini 1,5 volta, se mu zdi prav. Namerili bomo 0,6 volta.

Za poskus v resnici ne potrebujemo baterije. Vzamemo lahko kar Arduino in upore povežemo tako, da prek njih teče tok od +5 V do GND. Napetost prek dveh uporov bo zdaj 2 V.

Vemo, kaj se dogaja z zaporedno vezanimi upori? Njihova upornost se sešteva. Tudi to je preprosto izmeriti z merilnim inštrumentom. Hitro se lahko prepričamo, da je upornost dveh zaporedno vezanih 1-kiloohmskih uporov 2 kiloohma, upornost treh pa 3 kiloohme.

Z učenci lahko to naredimo še zabavneje. Izmerimo lahko upornost učenca: eno elektrodo mu "zabodemo" (v resnici je dobro, če malo pritisne, da bo boljši stik) v prst na eni roko, drugo v prst na drugi. Potem pa lahko izmerimo upornost dveh ali treh ali štirih učencev, ki se držijo za roke. Na poletni šoli so skupine tekmovala, katera bo imela najmanjšo upornost prek vseh

štirih otrok.

Ta poskus niti ni nepomemben. Rezultat nam bo prišel prav kasneje. Ugotovili bomo, da je upornost človeka približno 1 megaohm. Čisto približno. Nismo povedali ali med rokama ali med nosom in popkom. Prav tako je navadno manj, recimo pol megaohma. Predvsem pa ljudje kar dobro prevajamo, ne samo španščine temveč tudi elektriko, edini del, ki ne prevaja, je koža. Zato je na koncu vse odvisno od tega, kako suho kožo imamo. (Kadar zamenjujem stikalo, ne da bi se mi dalo izključiti varovalko, je to zelo slaba ideja tudi zato, ker me je malo strah in se še toliko bolj znojim.) 1 megaohm je vseeno dovolj natančno, da nam bo služilo za orientacijo malo kasneje.

Vrnimo se k temi. Namerili smo, da je padec napetosti (ja, tak je uraden strokovni termin) prek dveh od petih eno-kiloohmskih uporov enak 2 volta (od 5 V, ki jih dobimo iz Arduina). Poleg tega smo namerili, da je skupna upornost dveh 1-kiloohmskih uporov dva kiloohma. Potemtakem bi ju lahko zamenjali z enim samim 2-kiloohmskim uporom. One tri na drugi strani pa z enim tri kiloohmskim.

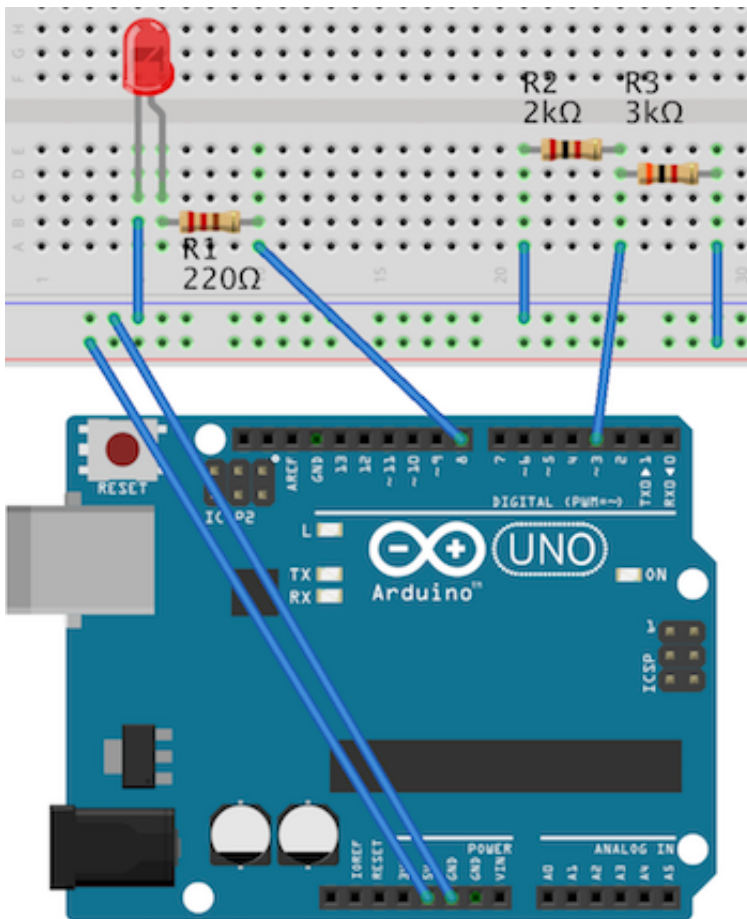
Pa razmislimo, kaj to pomeni. Če imamo 2-kiloohmski in 3-kiloohmski upor in ju priključimo na 5 V (recimo tako, da je 3-kiloohmski na tisti strani, kjer je 5 V, ne GND), bo napetost, ki jo izmerimo med prek 2-kiloohmskega upora enaka 5 V. Na eni strani kiloohmskega upora je GND. Torej lahko rečemo, da je napetost med GND in drugo stranjo 2-kiloohmskega upora enaka 2 V. Ker pa tako ali tako vse napetosti primerjamo z GND, bomo rekli, da je napetost na točki med uporoma enaka 2 V.

Takle razmislek seveda ne velja le za 2- in 3-kiloohmske upore. Napetost se vedno razdeli v razmerju uporov. Če bi vzeli, recimo, upor za 1 megaohm in 10 megaohmov ter ju povezali na enak način, bi na prvem padlo $1 / 11$ napetosti ali slabe pol volta (od petih).

Napetosti na vhidih

Kot vemo, lahko s klicem funkcije `digitalRead(3)` izvemo, ali je pin 3 "HIGH" ali "LOW". Pin je HIGH, kadar mu pripeljemo 5 V in LOW, če je na zemlji (GND, 0 V). Kaj pa, če mu pripeljemo nekaj vmes, recimo 2 V?

Poskusimo. Točko med uporoma povežimo s pinom, recimo, 3. Na pin 8 dajmo diodo.



Napišimo program, ki določi vhodne in izhodne pine. Nato prebere stanje pina 3 (HIGH ali LOW) in postavi pin 8 na enako stanje.

```
void setup() {
  pinMode(3, INPUT);
  pinMode(8, OUTPUT);
}

void loop() {
  int stanje = digitalRead(3);
  digitalWrite(8, stanje);
}
```

Naloga 1. Stori tako. Dioda sveti ali ne?

Naloga 2. Na pin 3 pripelji različne napetosti - 0 V, 1 V, 2 V, 3 V, 4 V, 5 V. Pomagaj si z verigo uporov, kot smo jo sestavljali prej. Kaj se dogaja z diodo?

Dougcajt. Pripeljemo mu različne napetosti, Arduino pa v njih vidi le dve vrednosti, LOW ali HIGH. Ne obupajmo. Moj Arduino je lahko tudi voltmeter.

Analogni vhodi

Spremenimo vezje tako, da točke med uporoma ne povežemo z vhodom 3, temveč z enim od

vhodov, ki jih doslej še nismo uporabljali: A0 (ali katerikoli drugim, ki se začne s črko A; če niso označeni z A, pa nad skupino pinov piše ANALOG).

Stanja analognih pinov ne beremo z `digitalRead` temveč `analogRead`. Kot argument podamo oznako pina, recimo `A0`, `A1`, `A2` in tako naprej. Rezultat klica funkcije `analogRead` ni HIGH ali LOW temveč število med 0 in 1024. Vrednost 0 pomeni 0 V, 1024 pa 5 V.

Če vzamemo upora z 2 in 3 kilohme ter točko med njima povežemo z A0, bi moral `analogRead(A0)` vrniti dobrih 400.

Kako pa bomo preverili, ali je res? Ha, zato pa smo se naučili izpisovati! Čez sedem minut res vse prav pride!

```
void setup() {
  Serial.begin(9600);
  int v = analogRead(A0);
  Serial.println(v);
}

void loop() {
}
```

Ta program je nekoliko dolgočasen. V začetku pomeri napetost in jo izpiše, potem pa navdušeno goni prazen `loop`. Naj izpisuje stalno.

```
void setup() {
  Serial.begin(9600);
}

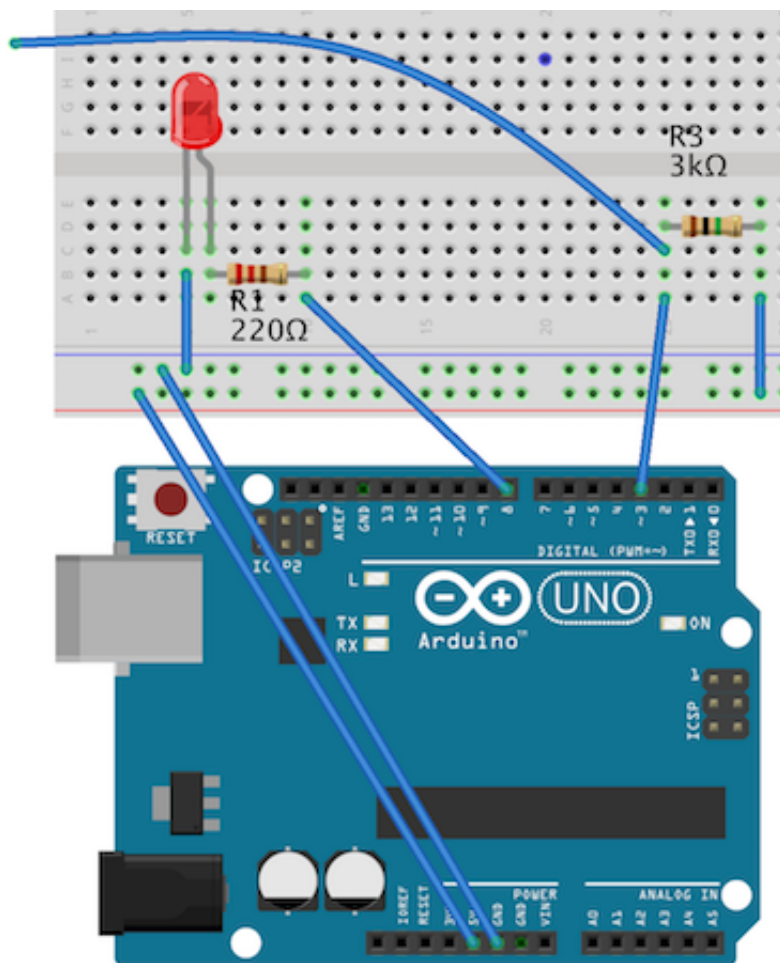
void loop() {
  int v = analogRead(A0);
  println(v);
  delay(300);
}
```

Še vedno je nekoliko dolgočasen. Prej je izpisal in konec, zdaj pa stalno izpisuje eno in isto. Kot da je to kaj boljše...

Kdo je močnejši?

Smo že skoraj na cilju. 3-kilohmski upor, se pravi *tisti upor, ki je vezan na 5 V* zamenjajmo z ogromnim uporom. Najbolj pravšnji bo takšen za 10 mega-ohmov. Drugi upor preprosto potegnemo iz poskusne plošče, pa žico, ki je peljala do njega, prav tako. Pač pa dodajmo na drugi konec preostalega upora še eno žico, ki ne pelje nikamor. Ker ... zakaj pa ne bi?

Da bo bolj zanimivo, preklopimo žico, ki vodi iz točke med uporoma (kjer enega upora pravzaprav ni več) iz A0 nazaj v tretji digitalni pin, D3.



In potem spet poskusimo prvi program, ki smo ga napisali danes.

```
void setup() {  
  pinMode(3, INPUT);  
  pinMode(8, OUTPUT);  
}  
  
void loop() {  
  int stanje = digitalRead(3);  
  digitalWrite(8, stanje);  
}
```

Bo dioda svetiła? Preskusi, preden bereš naprej.

Prepozno, si že prebral. :) Seveda bo. Pin 3 je povezan s 5V - sicer prek upora, a vseeno. Kaj pa tista žica, ki visi kar tako? Tista nič ne vpliva. Ne, ker ... zakaj pa bi?

Zdaj pa primi drugi konec tiste žice. Ne izoliranega dela temveč iglico na koncu žice. Z golo roko, brez strahu. :)

Z diodo se lahko zgodi dvoje. Ker so reči pač digitalne. :) Dioda bo mora svetiła še naprej,

morda pa bo ugasnila. To je odvisno predvsem od tebe.

Da bo poučnejše, se delajmo, da sveti še naprej. Stori tole: z eno roko še naprej drži žico, z drugo pa se dotakni konektorja USB. Ne plastičnega dela, temveč njegovega kovinskega ohišja. Zdaj pa dioda zagotovo ugasne.

Kaj se dogaja?

Spet imamo dva upornika! Povezana sta natančno tako, kot sta bila 2- in 3-kiloohmski upor. Prvi upornik smo mi. Povezani smo z ohišjem USB. To pa je vedno povezano z zemljo, GND. Povezava gre torej prek nas do drugega upora in potem na 5 V. Enako kot prej. D3 nameri, kakšna je napetost med njima. Točneje, kakšna napetost pade na prvem uporniku - na nas.

Rekli smo, da je upornost človeka (govorimo o zrelih osebah, ne pubertetnikih, se razume) približno 1 megaohm. Drugi upor ima 10 megaohmov. Hej, saj to smo pa že izračunali! Na D3 bomo namerili slabega pol volta. Za D3 je to definitivno LOW in dioda ugasne.

Zakaj pa včasih dioda ugasne tudi, če se ne dotikamo kovinskega ohišja USBja? Svet je po definiciji ozemljen. Vse je sicer malo odvisno od tega, kako in kam je priključen Arduino, kakšna je soba, v kateri se to igramo, morda vpliva celo vlažnost zraka, gotovo pa vpliva, kaj smo *šlatali* preden smo prijeli žico. Če se dotaknemo, recimo, radiatorja, se bomo dobro naužili zemlje.

Se spomnimo stikal s "pull up uporom"? Stikal, ki smo jih vezali tako, da smo jim dodali upor? To je zelo podobno. 10 megaohmski upor vleče gor, na 5 voltov. Ker imamo ljudje samo 1 megaohm upora, to brez težav potegnemo dovolj dol, proti zemlji.

Ko smo se šli tole z otroki, smo jim naredili zapestnice iz neizolirane žice in jih z izolirano žico priključili na GND.

Limona na analognem vhodu

Najprimernejše sadje za nadaljevanje te vaje je limona. Je primerno trda, predvsem pa ji s tem, kar jo čaka, ne bomo prišli do živega, sploh če so nam podtaknili takšne s 3 cm debelo lupino.

Vzamemo torej limono in zabodimo žico, ki smo je prej prijemali vanjo. To je to. Namesto da bi prijeli žico, zdaj primemo limono. Učinek je isti.

Ali pa tudi ne. Limona doda nekaj svoje upornosti. Poleg tega ni vseeno, ali se je dotaknemo z mezinco ali jo pošteno zgrabimo. Če je limona suha (recimo posušena) ali pa imamo slabo povezavo iz kakega drugega razloga, dioda ne bo ugasnila, saj napetost ne bo padla pod tri volte. To bi sicer pomenilo, da je naša upornost tako velika, da ne premagamo niti 10 megaohmskega upora, vendar - se zgodi. (V resnici je problem navadno v limoni, ne v nas.)

Napetost ne pade pod 3 volte... In zato digitalni pin 3 ne prebere LOW. Ha, danes nam pa res vse prav pride! Saj imamo analogne pine. In tudi napetosti znamo izpisovati!

Brž preklopimo žico iz D3 na A0 in poženemo izpisovanje napetosti.

```
void setup() {
  Serial.begin(9600);
}

void loop() {
  int v = analogRead(A0);
  println(v);
  delay(300);
}
```

Tako “umerimo” limono. Videli bomo, da je napetost, ko se je ne dotikamo, zelo blizu 5 V; na A0 bomo dobili številke okrog 1000. Ko se je dotaknemo, pa bo meritev na A0 gotovo padla pod 900 ali, za vsak slučaj 800. Pa sprogramirajmo, naj bo dioda prižgana, ko se dotikamo limone.

```
void setup() {
  pinMode(8, OUTPUT)
}

void loop() {
  if (analogRead(A0) > 800) {
    digitalWrite(3, LOW);
  }
  else {
    digitalWrite(3, HIGH);
  }
}
```

Limona, ki piska, tipka...

Možnosti odtod so brezmejne. Na Arduino lahko priključimo zvočnik, da bo piskal, ko se dotaknemo limone. Ali pa vzamemo pet limon, ki piskajo z različnimi toni.

Namesto limon lahko damo na stopnice aluminijske folije (in jih priključimo na digitalne pine, saj je analognih le šest, mi pa jih potrebujemo več), stopniščno ograjo — seveda to deluje le, če je kovinska — pa priključimo na GND. Vsaka folija naj piska z drugim tonom iz lestvice. Pa smo spremenili stopnice v klavir. Le otroci morajo hoditi po njih bosih in se držati ograje.

Še imenitnejše je, če nimamo Arduina Uno temveč Arduina Leonardo, ki ga je mogoče brez posebnih komedij spremeniti v tipkovnico. Sprogramiramo ga tako, da dotik limone povzroči,

da računalnik misli, da smo pritisnili kako tipko. Mlajši otroci lahko te pritiske lovijo v Scratchu.

Na poletni šoli smo na tla prilepili folije v obliki smernih tipk in otroci so morali igrati igrico tako, da so skakali s folije na folijo.

Vse tole je čisto preprosto. Naredimo lahko, kar nam da domišljija.

Banana se upira

In to sploh še ni vse — kot rečejo v top shopovih reklamah.

Za tole lahko vzamemo banano. Ali kakšno manj eksotično sadje, samo da je dolgo in nekoliko vlažno. Tudi trak mokrega papirja bo čisto primeren, če se znamo izogniti potencialni poplavi. Kakorkoli, tule se bomo delali, da imamo banano.

Kakšna je upornost banane? V času pisanja tega besedila nisem imel pri roki nobene, Google pa mi vrne kup znanstvenih člankov o tem, kako z merjenjem upornosti določiti zrelost banane in celo njeno vrsto, nekdo pa je reč tudi uspešno patentiral. Le nesrečne upornosti nisem izvedel. Pa predpostavimo, da je 1 megaohm (v resnici mora biti precej manj, bi rekel).

Vzamemo torej banano in se delamo, da je upornik. Na eni strani vanjo zabodemo žico, ki vodi v GND, na drugi strani žico, ki vodi v 5 V. Nekam vmes pa žico, ki vodi na A0. (Slednje ni potrebno zabosti. Lahko je tudi le pritisnemo, da bomo banano potem z mirnejšim srcem pojedli. Naši poskusi je ne bodo poškodovali, pač pa bo zaradi vsega dotikanja hitro počrnela.)

Če nimamo banan ali pa se nočemo igrati s hrano (zelo veljaven pomislek!), vzamemo vlažen papirnat trak in žice pripenjamo nanj. Uporabimo lahko “krokodilčke”, če jih imamo, sicer pa ščipalke za perilo ali, če boljše, sponke za papir.

Ne smemo pa uporabiti alu-folije! Ta prevaja predobro in bi naredili kratek stik.

Banana ali trak, vseeno: kaj bo izpisal Arduino?

Če ima cela banana upornost 1 megaohm, ima pol banane upornost pol megaohma. Pač, pol banane je tudi upornik; upornosti se seštevajo in upornost dveh polovic banane je pač dvakrat večja od upornosti ene polovice.

Kaj pa, če žico, ki vodi na A0 pomikamo po banani? Če se bližamo žici, ki gre na GND, po napetost padala, če žici, ki gre na 5 V, pa naraščala.

Naredimo lahko tole: na Arduina priključimo zvočnik in frekvenca zvoka naj bo odvisna od izmerjene napetosti.

Še boljše. Namesto, da bi bo banani pomikali žico, naredimo “prstan” in privežimo žico nanj. Zdaj lahko igramo na banano tako, da po njej drsimo s prstom.

Banana bo “fušala”. Uglasimo jo lahko tako, da nanjo narišemo klaviaturo. S prstom se dotikamo tipk in zapisujemo izpisane napetosti. Potem popravimo program, ki igra tone tako, da ne igra poljubnega tona, zvezno odvisnega od napetosti. Ker vemo, kakšna (približno) je napetost, ki jo dobimo pri pritisku določene tipke, lahko zaigramo ustreznega tona. Pri tem ne moremo pričakovati, da bo napetost vedno enaka prej namerjeni; zaigramo pač ton, ki je najbližji namerjeni frekvenci.

Kaj ne deluje?

Tipkovnico naj bi bilo mogoče narisati z dovolj mehkim svinčnikom, saj je grafit dober prevodnik. Vsaj tako je videti v reklami za Makey Makey, ki je navdahnil vse tole. To se mi ni obneslo. Če je svinčnik pretrd, na papirju ne pusti dovolj grafita in upornost je tako prevelika. Če je svinčnik mehak, pa bo grafita dovolj, vendar bo zelo hitro samo na naših prstih. Očitno je reklama za Arduino nekoliko zrežirana.

Prav tako ne deluje plastelin. V teoriji naj bi odlično prevajal. Morda obstajajo različne vrste in sem imel smolo, da sem šel v Mercator in ne Lidl. Vendar: ni delovalo.

Banane so zabavne, vendar postanejo zelo hitro črne in imaš slabo vest, saj tisto ni več (videti) okusno. Tudi kot klavir ne delujejo dolgo, saj se zaradi dotikanja zmehčajo in se jim prevodnost spreminja.

Limone so v resnici super. Ker sem pravi moški, ki si kavo, ki mu jo prinese žena v delavnico, brez pomislekov (in ženinega zgražanja, saj se je že navadila), pomeša s *šraufcigerjem*, ki ga ima ravno v roki (za kar imam čisto dober izgovor, ključ številka 28 pač ne bi šel v skodelico), tudi limone, ki so jih malo prebadali z iglicami, smatram za popolnoma uporabne. Dobro pa jih je čimprej utekočiniti, saj sicer na mestih, kjer so bile prebodene, porjavijo, poveča pa se jim tudi nagnjenost k gnitju.

Vlažen papir lepo deluje, je pa malo packarije.

Namesto limon smo uporabljali tudi lončke z vodo, v katere potopimo žice. Deluje, vendar z očitnim tveganjem.

Idej, kaj vse se še da priključiti, je na spletu še veliko. Vsekakor pa jih prej preskusite, saj sem naletel na dovolj imenitnih poskusov, ki mi jih ni uspelo ponoviti.

Rešitve

Naloga 1 in 2 Arduino bo napetosti manjše od 3 V šteje za LOW, večje pa za HIGH. V bližini

praga 3 V pa ne bo prepričan, kaj naj si misli, in bo skakal visoko in nizko.