

## Arduino: pa ne še en računalnik!

Računalniki so vdrlji na področje fizikalnih merjenj zato, ker ponujajo avtomatizacijo in obdelavo podatkov. Na razpolago je veliko pripomočkov, ki naredijo merjenja in vrednotenje rezultatov enostavna, osebni računalnik je le en od njih. Zakaj bi torej potrebovali še en računalnik?

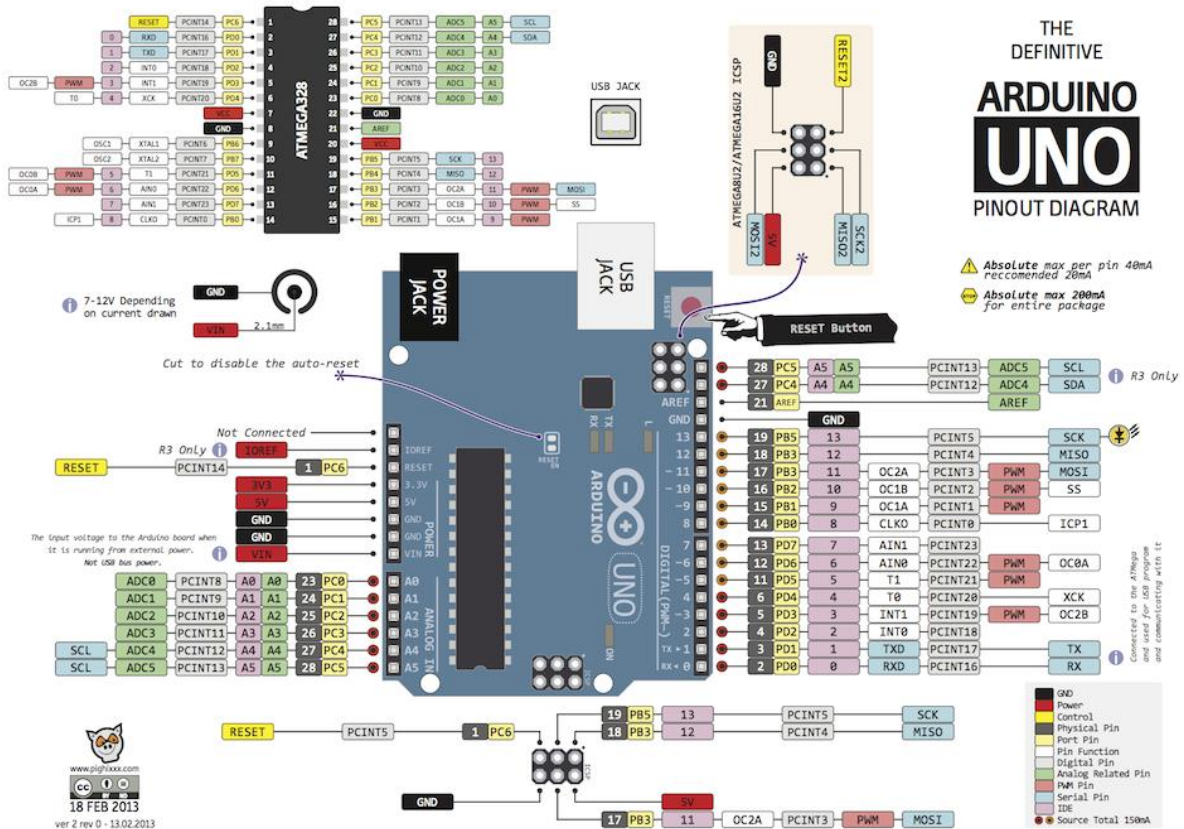
Osebnemu računalniku se zalomi takrat, ko je treba meritve in rezultate obdelovati sproti, temu pravimo »v realnem času«. Osebni računalnik je namreč otovorjen z operacijskim sistemom, ki občasno zahteva strojno opremo računalnika zase in takrat nam računalnikove sposobnosti niso na razpolago. Spremljanje fizikalnih pojavov lahko implementirajo v vmesniku, računalnik pa služi samo za prikazovanje in obdelavo podatkov. Uporabniku je pač vseeno, če rezultate meritev dobi z nekaj milisekundno kasnitvijo. Kadar pa je treba na podlagi izmerjenih rezultatov poseči v fizikalni sistem, je osebni računalnik zaradi občasne zasedenosti z operacijskim sistemom popolnoma neuporaben; milisekunde so predolge. Takrat potrebujemo druga sredstva za merjenja in obdelavo podatkov ter poseganje v fizikalni sistem.

Danes se bomo igrali z enostavnim mikroprocesorskim sistemom, ki je zelo razširjen in odično dokumentiran. Zanj se dobi na internetu mnogo zgledov, programov, dodatkov, senzorjev in aktuatorjev; vse to se dobi tudi pri cenovno ugodnejših virih. In programiranje je tako zabavno...

Arduino sistem so razvili v Italiji. Programiramo ga v programskem jeziku »C«, ki je obogaten z dodatnimi ukazi; ti pomagajo uporabljati vsebino mikroprocesorja. Za vabo: cena sistema Arduino je manj kot 10€, senzori so naprodaj za nekaj deset € (set 37 senzorjev, ki jih bomo na primer uporabljali v tej delavnici). Na razpolago so tudi moduli za Bluetooth komunikacijo, dostop do interneta, poganjanje koračnih in običajnih motorjev... Cena torej ni ovira za uporabo, ki jo olajšuje še velika količina zgledov na internetu. Sredstva za programiranje so v javni rabi in zastonj. Čas je za igranje!

Danes bomo začeli prav tam, pri igranju z mikroprocesorjem.

Na sliki spodaj je tloris Arduino Uno ploščice z označenimi priključki. Vsak digitalni priključek razume logično 0 (približno nič voltov) ali logično 1 (približno pet voltov). Največji tok, ki ga lahko črpamo iz izhoda, je 20 mA. Vsak analogni priključek razume katerokoli napetost med nič in pet voltov. Napetost preko teh meja lahko trajno poškoduje ploščico. Arduino ploščico priključimo na osebni računalnik s priloženim USB kablom. Ta povezava služi programiranju in hkrati napajanju ploščice. Ko je program enkrat v Arduinu, lahko ploščico odklopimo od osebnega računalnika in napajamo iz baterij.



Podrobnosti, navodila, zglede in komercialne ponudbe najdete na primer na straneh:

- <http://arduino.cc/>
- <http://en.wikipedia.org/wiki/Arduino>
- <http://arduinox.si/>

Cenovno ugodnejše ploščice Arduino in dodatne module najdete na straneh:

- <http://www.dx.com/c/electrical-tools-499/arduino-scm-supplies-436?pageSize=200&tc=EUR&gclid=CIOetp3OssQCFUcUwwodB4IA6g>
- [http://www.aliexpress.com/premium/arduino.html?ltype=wholesale&SearchText=arduino&isPremium=y&d=y&origin=y&initiative\\_id=SB\\_20150318111547&isViewCP=y&catId=0](http://www.aliexpress.com/premium/arduino.html?ltype=wholesale&SearchText=arduino&isPremium=y&d=y&origin=y&initiative_id=SB_20150318111547&isViewCP=y&catId=0)

Zadoščalo tudi, da v Google iskalnik vtipekate »Arduino« in zasule vas bodo strani o Arduino mikroprocesorskem sistemu in izvedenkah. Zglede programov in opise najlaže najdete iz okolja Arduino, rubrika »Zgledi« (»Examples«).

Pa začnimo s programiranjem. Prvi zglede bo program »Hello world«; ker Arduino ploščica nima zaslona, bo delovanje potrdilo utripanje LE diode na ploščici. Ta je prispajkana na priključek »13«, najdete ga na zgornjem tlorsisu. In kako pripravimo LE diodo do utripanja? Najprej postavimo priključek »13« v stanje 1, čez nekaj časa ga postavite v stanje 0 in po nekaj časa spet v stanje 1... Upornik, ki bi ga bilo treba vezati zaporedno z LE diodo do priključka »13« je že na ploščici Arduino.

Program je sestavljen iz dveh delov, slika spodaj. Prvi del se izvede le enkrat, takoj po zagonu programa. V tem delu konfiguriramo Arduino ploščico: definiramo delovanje priključkov mikroprocesorja, inicializiramo kose programa... Drugi del se ponavlja; ko izvajanje drugega dela pride do konca, se ponovi od začetka.

Za utripanje LE diode je zato treba:

- Prvi del programa: konfigurirati priključek »13« tako, da bo Arduino tam generiral digitalni signal
- Drugi del programa: postaviti priključek »13« na logično ena, malo počakati, nato postaviti priključek »13« na logično 0 in spet malo počakati.

Besedilo programa je na sliki spodaj. Čakanje definira argument funkcije »delay(1000)«; 1000 pomeni 1000 ms. Signal na mikroprocesorjevem priključku spreminjata ukaza »digitalWrite(13, HIGH)« in »digitalWrite(13, LOW)«, 13 seveda pomeni priključek »13«.

```
void setup() {           // prvi del
  // put your setup code here, to run once:
  pinMode(13, OUTPUT);
}

void loop() {           // drugi del
  // put your main code here, to run repeatedly:
  digitalWrite(13, HIGH);
  delay(1000);
  digitalWrite(13, LOW);
  delay(1000);
}
```

Z osebnega računalnika lahko Arduino pošiljamo črke in številke preko serijskega kanala. Poskusimo poslati dve števili, Arduino naj vrne vsoto. Števili sta ločeni s presledkom, program je tule:

```
void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
}

int a, b, rez;

void loop() {
  // put your main code here, to run repeatedly:
  if (Serial.available()) {
    a = Serial.parseInt();
    b = Serial.parseInt();
    rez = a + b;
    Serial.println(rez);
  };
}
```

Komunikacijski kanal je treba najprej konfigurirati. To storimo samo enkrat v prvem delu programa z ukazom »Serial.begin(9600)«, ki nastavi hitrost prenosa podatkov med osebnim računalnikom in Arduinom na 9600 bitov na sekundo.

Ukaz »Serial.available()« preveri, ali je bilo z osebnega računalnika poslano število. Če je temu tako, program prebere število z ukazom »Serial.parseInt()«, ki prejete številke interpretira kot celo število, v našem programu to storimo dvakrat zapored, nato števili seštejemo in vrnemo po serijskem kanalu z ukazom »Serial.print(rez)«; dvojček tega ukaza »Serial.println(rez)« še pomakne kurzor v novo vrstico.

Zdaj lahko že poskusimo z osebnega računalnika prižigati in ugašati LE diodo. Domenimo se, da z osebnega računalnika poslana črka »h« vklopi LED, črka »l« naj LED ugasne. V drugem delu programa najprej preberemo znak (če je ta na razpolago) v spremenljivko »znak«. Vrednost spremenljivke »znak« preverjamo z zaporednimi stavki »if ()«; če je odgovor na katero vprašanje trdilen, se izvede ustrezen del programa. Drugi del programa je spodaj, prvi del je enak tistemu zgoraj.

```
char znak;

void loop() {
  // put your main code here, to run repeatedly:
  if (Serial.available()){
    znak = Serial.read();
    if (znak == 'h') digitalWrite(13, HIGH);
    if (znak == 'l') digitalWrite(13, LOW);
  };
}
```

Z osebnega računalnika bi lahko pošiljali več različnih ukazov, v program na Arduinu bi morali le dodati več vprašalnih stavkov in ustreznih akcij na posamezne ukaze.

Z osebnega računalnika lahko pošiljamo tudi kombinirane ukaze, na primer črko za vrsto ukaza in številko za argument tega ukaza. Vzemimo, da želimo vklopiti/izklopiti več LE diod, ki so priključene na različne priključke Arduina. Takrat bi ukaz sestavljali črka »h« ali »l«, ki bi ji sledile številka priključka; »h13« vklopi LED na Arduino ploščici. Takrat mora program interpretirati kombiniral ukaz, to storimo s spodnjimi programskimi vrsticami.

```
char znak;
int pin;

void loop() {
  // put your main code here, to run repeatedly:
  if (Serial.available()){
    znak = Serial.read();
    pin = Serial.parseInt();
    if (znak == 'h') digitalWrite(pin, HIGH);
    if (znak == 'l') digitalWrite(pin, LOW);
  };
}
```

V prvi vrstici čakamo na znake z osebnega računalnika. Ko ti pridejo, najprej preberemo eno črko v spremenljivko »znak«, preostanek pa preberemo v spremenljivko »pin«. Nazadnje pravi priključek mikroprocesorja postavimo na željeno vrednost.

Arduinu lahko pošiljamo ukaze preko serijskega priključka, kot smo to pravkar storili. Namesto tega lahko uporabimo tudi na primer Bluetooth modul, ki ga povežemo na Arduino. Takrat se Bluetooth modul poveže na primer s pametnim telefonim (na njem teče operacijski sistem Android in program za terminal) in že lahko brezžično vklopimo in izklopimo LE diode (ali pa karkoli drugega, kar je navezano na Arduino).

Arduino ploščica lahko meri analogne napetosti. Poskusimo tako, da z osebnega računalnika pošljemo ukaz »m«, Arduino pa v odzivu vrne izmerjeno vrednost s priključka A1. Program je spodaj.

```
char znak;  
  
void loop() {  
  // put your main code here, to run repeatedly:  
  if (Serial.available()){  
    znak = Serial.read();  
    if (znak == 'm') Serial.println(analogRead(A1));  
  };  
}
```

Arduino priključkov za merjenje analognih napetosti ni treba posebej konfigurirati. Dovolj je, da analogno vrednost preberemo z ukazom »analogRead(A1)« ter vrnjeno vrednost pošljemo po serijskem kanalu do osebnega računalnika (ali preko Bluetooth modula do pametnega telefona).

Arduino ne zna generirati analogne napetosti, lahko pa generira digitalni signal, ki skače med 0 in 1 tako, da je povprečna vrednost generiranega signala ena od 256 možnih diskretnih vrednosti. Metodi rečemo PWM (»Pulse Width Modulation«, pulzno-širinska modulacija). Poskusimo tako, da z osebnega računalnika pošljemo ukaz »p###«, pri tem predstavlja »###« katerokoli število med 0 in 255. Ukaz za generiranje PWM signala je »analogWrite(9,###)«, pri tem je na primer 9 oznaka priključka, kjer želimo generirati PWM signal; primerni so priključki, ki so na ploščici Arduino označeni z '~'. Seveda gre tudi preko Bluetootha, če pa imamo na razpolago modul za povezavo z internetom, lahko to počnemo od koderkoli na svetu.

```
char znak;  
int val;  
  
void loop() {  
  // put your main code here, to run repeatedly:  
  if (Serial.available()){  
    znak = Serial.read();  
    val = Serial.parseInt();  
    if (znak == 'p') analogWrite(9, val);  
  };  
}
```

Bluetooth modul in modul za povezavo z internetom staneta vsak posebej med pet in deset €.

Isto tehniko uporabimo takrat, kadar želimo krmiliti servo motor. Taki so na primer namenjeni uravnavanju krmila v daljinsko vodenih igračkah. Tukaj trajanje pulzov na kontrolnem vhodu servo motorja definira položaj motorja. V okolju Arduino je na razpolago zgled za tak motor, položaj spet lahko nadziramo iz osebnega računalnika, ali pa ga določi Arduino na podlagi merjenja analogne vrednosti.

Tudi poganjanje koračnega motorja je za Arduino »mala mal'ca«. Naprodaj so krmilniki koračnih motorjev in motorji, vse skupaj spet stane okrog pet €. Koračni motor ima štiri ločene priključke za štiri elektromagnete in en skupni priključek. En od elektromagnetov je pod napetostjo, ostali ne. Elektromagnet, ki je pod napetostjo, zadržuje koračni motor v izbranem položaju. Ko postavimo pod napetost sosednji elektromagnet, se koračni motor zasuka za majhen kot in tam obstane. Z zaporednim vklapljanjem elektromagnetov dosežemo, da se koračni motor suče. Ker vemo, kako smo krmilili motor, vemo njegovo trenutno orientacijo, kar omogoča natančno pozicioniranje.

Kadar želimo samostojno delovanje Arduino ploščice in ne maramo povezave do drugega računalnika, lahko rezultate kažemo na LCD zaslončku z dvema vrsticama po šestnajst znakov. Enoti rečejo »LCD shield«, stane med pet in osem € in ima za povrh še nekaj tipk. Natakne se na Arduino tako, da so povezave zanesljive, prosti priključki Arduina pa so zdaj na razpolago na ploščici z LCD zaslonom. Za poganjanje takega zaslončka je na razpolago knjižnica s standardnimi ukazi. Zgled programa je spodaj.

```
#include <LiquidCrystal.h>

// initialize the library with the numbers of the interface pins
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

void setup() {
  lcd.begin(16, 2);    // set up the LCD's number of columns and rows:
  lcd.print("hello, world!");    // Print a message to the LCD.
}

void loop() {
  lcd.setCursor(0, 1);    // set the cursor to column 0, line 1
  lcd.print(millis() / 1000);    // number of seconds since reset
}
```

LCD zaslon zaseda precej priključkov Arduino ploščice. Če nam to ne ustreza, lahko uporabimo drugačno verzijo LCD zaslončka, ki prejema informacije preko samo dveh priključkov (I2C vodilo, 7€), za tako verzijo je potrebna druga knjižnica. Tudi ta je, kot prejšnja, prosto dostopna na internetu.

Dokupimo lahko tudi senzor nagiba in rotacij. Cena? Malenkost, okrog 10€. Na Arduino ga priključimo z dvema žicama, komunikacija pa poteka po I2C protokolu. Brez skrbi, komunikacijo pokriva knjižnica, ki je prosto dostopna, zgled uporabe pa je na razpolago v rubriku »zglede«.

Mikroprocesorski sistem je uporaben takrat, ko lahko z njim merimo in nadziramo fizikalne sisteme. Naprodaj je več različnih setov senzorjev in aktuatorjev, ki jih enostavno priklopimo na Arduino, spišemo program in že (upam) deluje. Za delavnico imamo na razpolago set 37 senzorjev. Z njimi lahko merimo temperaturo, osvetljenost (vidna in IR svetloba), vlažnost, magnetno polje, zvok, tresljaje... V setu je tudi nekaj aktuatorjev (LE diode različnih vrst, rele, zvočnik, ...). Setu smo dodali še senzorje za merjenje pospeškov v treh dimenzijah ter žiroskop. Imamo tudi močnostno izhodno stopnjo za pogon motorjev in MOSFET tranzistor za močnejša bremena. Set dopolnjujejo povezovalne žice (3€), pa še preskusna ploščica (»protoboard«) bo prišla prav. Zunanje napajanje potrebujemo šele takrat, ko so potrebne višje napetosti ali pa več toka, kot ga daje USB priključek osebnega računalnika.

Eden bolj ambicioznih projektov za Arduino je 3D tiskalnik, opis projekta najdete na:

- <http://store.arduino.cc/category/41>

Na tem naslovu pa najdete prispevke uporabnikov in navdušencev Arduina:

- <http://playground.arduino.cc/>