

Akceleracija slobodnog pada

Saša Ilijić (FER/ZPF)

6. listopada 2015.

Sažetak

U ovoj laboratorijskoj vježbi mjerimo trajanje slobodnog pada tijela sa zadane visine te određujemo akceleraciju tijela u slobodnom padu. Koristimo uređaj u kojem elektromagnet otpušta čeličnu kuglicu koja, nakon što slobodno padajući prevali zadanu visinu, udara u prekidač. Elektromagnet (aktuator) i prekidač (senzor) povezani su s upravljačkim modulom (programabilnim mikrokontrolerom) *Arduino Uno* koji mjeri proteklo vrijeme. Vježba se sastoji od sastavljanja elektroničkog sklopa, programiranja mikrokontrolera te od prikupljanja i obrade rezultata mjerenja. Ove upute pretpostavljaju da je riječ o prvom susretu studenta s upravljačkim modulom iz porodice *Arduino*.

1 Uvod i najvažniji pojmovi

- Slobodni pad. Ispustimo li tijelo koje miruje, ono započinje slobodni pad. Zanimarimo li učinak sila otpora zraka, u intervalu vremena t ono prevlađuje visinsku razliku

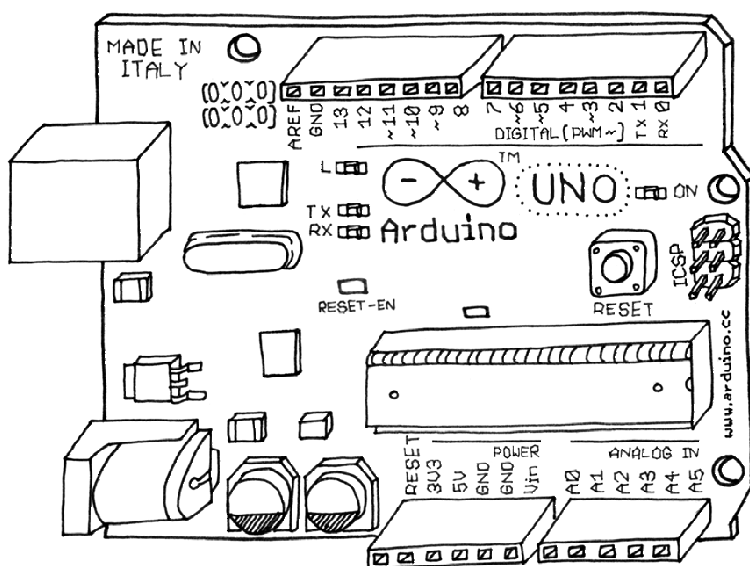
$$h = \frac{g}{2}t^2, \quad (1)$$

gdje je g akceleracija tijela. Izmjerimo li h i t , možemo odrediti vrijednost g . S obzirom da je približna vrijednost akceleracije 10 m s^{-2} , visinskoj razlici od 1 m ili manje odgovaraju trajanja pada kraća od pola sekunde. U ovoj vježbi upoznat ćemo uređaj kojime se takva vremena mogu precizno izmjeriti.

- Senzori i aktuatori. Senzori pretvaraju fizikalne događaje (npr. gibanje) u električne impulse, a aktuatori pretvaraju električne impulse u fizikalne događaje (npr. gibanje).
- Programabilni mikrokontroler je elektronički sklop koji može, u skladu s programom koji napišemo, primati i obrađivati podatke iz senzora, upravljati aktuatorima, te komunicirati s računalima, računalnim mrežama i drugim uređajima (npr. s ekranom). U ovoj vježbi ćemo upoznati upravljački modul *Arduino Uno* unutar kojeg se nalazi programabilni mikrokontroler.
- Elektromagnet se sastoji od zavojnice i jezgre. Kada zavojnicom teče električna struja, u okolini jezgre se stvara magnetsko polje koje privlači predmete načinjene od materijala poput željeza. Prekine li se električna struja, privlačna sila nestaje. U ovoj vježbi elektromagnet ima ulogu aktuatora.

2 Upravljački modul Arduino Uno

Arduino Uno je jednostavan upravljački modul namijenjen izradi prototipova interaktivnih uređaja te uporabi u nastavi. Unutar modula se nalazi programabilni mikrokontroler te niz elektroničkih komponenata koji olakšavaju njegovo korištenje. Sva tehnička dokumentacija, materijali za učenje o modulima iz porodice Arduino te programska potpora potrebna za rad s njima (Arduino razvojno okruženje, engl. Arduino Integrated Development Environment, IDE), nalaze se na adresi <http://www.arduino.cc>.



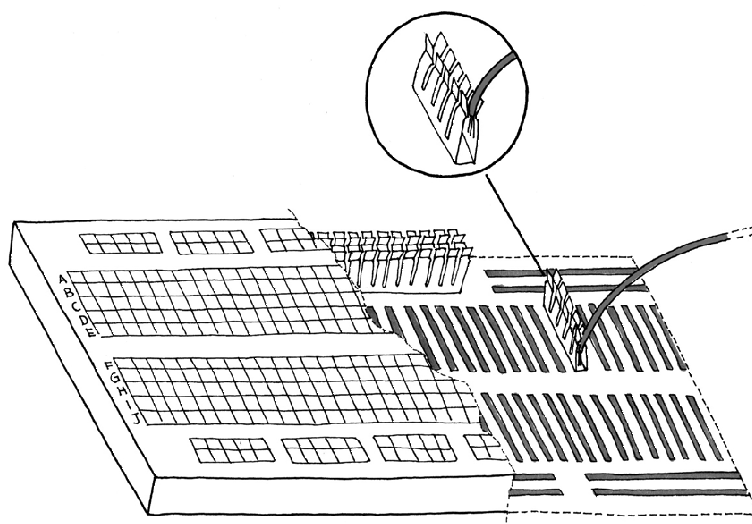
Slika 1: Upravljački modul Arduino Uno. Ovisno o proizvođaču i generaciji, izgled modula može u manje važnim pojedinostima odstupati od ovdje prikazanog. (Ilustracija je preuzeta iz [1]).

Ovo poglavlje u grubim crtama opisuje samo one funkcije modula Arduino Uno koje koristimo u ovoj laboratorijskoj vježbi.

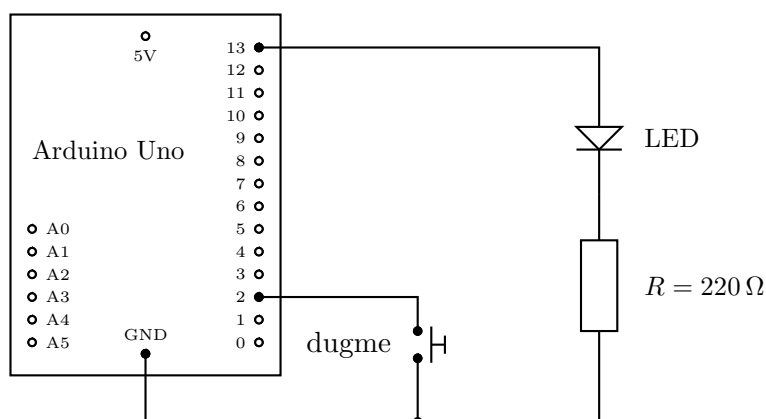
2.1 Osnovne karakteristike upravljačkog modula Arduino Uno

Arduino Uno je pločica veličine približno 5×7 cm na kojoj se pored niza elektroničkih komponenata nalazi i niz utičnica, tzv. pinova, koje omogućuju njegovo korištenje bez potrebe za lemljenjem. Arduino Uno je prikazan slici 1 gdje uočavamo sljedeće:

- Veliki kvadar lijevo–gore na slici je USB-utičnica koja omogućuje povezivanje Arduina s računalom. Arduino povezujemo s računalom kako bismo u njegovu memoriju pohranili program koji će on izvršavati te kako bi on tokom svog rada mogao slati i primiti podatke. Arduino može izvršavati svoj program i bez da je spojen na računalo. U tom slučaju USB utičnica može služiti za napajanje Arduina električnom energijom.
- U grupi utičnica (pinova) označenoj s POWER nalaze se dva pina označena s GND (engl. ground, odn. zemlja ili masa). Ti se pinovi nalaze na potencijalu, odn. naponu 0 V što još smatramo niskom logičkom razinom (LOW). U istoj se grupi nalaze pinovi označeni s 5V i 3V3 koji se nalaze na potencijalu, odn. naponu +5 V i +3.3 V u odnosu na GND. Njih možemo koristiti za napajanje elektroničkog sklopa koji priključujemo na Arduino. Potencijal 5 V smatramo visokom logičkom razinom (HIGH).
- Utičnice (pinovi) iz grupe DIGITAL označene brojevima 0 do 13 mogu, ovisno o potrebi, služiti u dvije različite svrhe. U INPUT načinu rada (modu) pin se koristi kako bi Arduino detektirao je li na njega doveden napon od približno 5 V (visoka logička razina, HIGH) ili je na njega doveden napon od približno 0 V (niska logička razina, LOW). U OUTPUT načinu rada (modu) Arduino će pin postaviti na napon 5 V (HIGH) ili na napon 0 V (LOW), ili će pak prema nekom obrascu brzo izmijenjivati ta dva stanja. Mod u kojem pojedini pin djeluje određuje se programom koji će Arduino izvršavati.
- Utičnice (pinovi) u grupi ANALOG IN označene s A0 do A5 omogućuju korištenje “analog-to-digital” (ADC) konvertera kojima Arduino može izmjeriti napon u rasponu 0–5 V i pretvoriti ga



Slika 2: Vanjski izgled i pogled na kontakte u unutrašnjosti eksperimentalne pločice. (Ilustracija preuzeta iz [1]).



Slika 3: Shema elektroničkog kruga s modulom Arduino Uno u kojem pritiskom na dugme palimo svijetleću diodu (LED). U seriju s LED postavljamo otpor $220\ \Omega$ kako iz priključnice (pina) 13, koja ovdje radi u OUTPUT modu, ne bi potekla struja veća od najveće dopuštene $I_{\max} = 20\ \text{mA}$.

u cijeli broj u rasponu od 0 do 1023. Pinove iz ove grupe moguće je koristiti i u druge svrhe kao npr. za komunikaciju Arduina s vanjskim ekranom.

- Na Arduinu se nalazi i dugme označeno s RESET. Pritiskom na to dugme Arduino prekida rad te započinje izvršavanje svog programa od njegova početka. Naravno, isto se postiže i isključenjem i ponovnim uključenjem napajanja.

Pri radu s Arduinom od velike je pomoći tzv. eksperimentalna pločica (slika 2) koja omogućuje sastavljanje elektroničkih krugova jednostavnim ubađanjem elektroničkih elemenata i spojnih žica, bez potrebe za lemljenjem. Eksperimentalne pločice se razlikuju po veličini i rasporedu kontakata u njihovoj unutrašnjosti.

2.2 Programiranje mikrokontrolera (primjer s LED)

Osnovne elemente programiranja mikrokontrolera prikazat ćemo na jednostavnom primjeru u kojem pritiskom na dugme palimo svijetleću diodu (eng. light-emitting diode, LED).¹ Shema elektroničkog sklopa prikazana je na slici 3. Dugme je spojeno na priključnicu (pin) 2, a LED je spojen na priključnicu (pin) 13, obje iz grupe DIGITAL. Pin 2 koristimo u INPUT modu jer želimo da Arduino prepozna je li dugme pritisnuto ili nije. Pin 13 koristimo u OUTPUT modu jer Arduino njime napaja ili ne napaja LED.

Programiranje Arduina izvodi se u jeziku koji sintaksom nalikuje programskom jeziku C. Svaki Arduino program sastoji se od funkcije `setup()` koju Arduino izvršava samo jednom neposredno nakon njegova pokretanja te od funkcije `loop()` koju nakon toga Arduino izvršava opetovano. Pri čitanju programa od pomažu nam komentari koji se nalaze između znakova `/* i */` ili iza znaka `//`.

```
1 /* arduino primjer: LED svijetli dok je dugme pritisnuto */
2
3 #define pinLED 13                // LED spajamo na pin 13
4 #define pinButton 2              // dugme spajamo na pin 2
5
6 void setup() {                   // izvršava se samo jednom
7     pinMode(pinLED, OUTPUT);      // pinLED radi u OUTPUT modu
8     pinMode(pinButton, INPUT_PULLUP); // pinButton radi u INPUT modu
9 }
10
11 void loop() {                    // izvršava se opetovano
12     if ( LOW == digitalRead(pinButton) ) { // ako je dugme pritisnuto
13         digitalWrite(pinLED, HIGH);        // upali LED
14     } else {                               // u protivnom
15         digitalWrite(pinLED, LOW);          // ugasi LED
16     }
17 }
```

Prepoznamo sljedeće elemente programa:

- U linijama 3 i 4 definiramo oznake `pinLED` i `pinButton` za brojeve priključnica (pinova) na koje smo odlučili spojiti LED i dugme. Korištenje takvih oznaka umjesto samih brojeva olakšava pisanje i čitanje programa te prilagodbu programa odlučimo li se za drugačiji odabir pinova.
- Linije 6–9 definiraju funkciju `setup()` unutar koje smo, pozivajući funkciju `pinMode()`, postavili pinove u željeni način rada (mod). Oznaka moda `INPUT_PULLUP` naznačava da se potencijal pina privlači na visoku logičku razinu (5 V) preko unutarnjeg priteznog otpornika ugrađenog u mikrokontroler. Kada se dugme pritisne, pin se privlači na nisku razinu jer se pritiskom zatvara kontakt između mase i pina, a cijeli pad napona od 5 V je na priteznom otporniku.
- Linije 11–17 definiraju funkciju `loop()`. Unutar te funkcije provjeravamo je li dugme pritisnuto, palimo LED ako jest, dok u protivnom gasimo LED. Poziv funkciji `digitalRead()` vraća vrijednosti `HIGH` ako je pin pri potencijalu 5 V, odnosno `LOW` ako je pin pri potencijalu 0 V. Poziv funkciji `digitalWrite()` postavlja pin na potencijal 5 V ili na potencijal 0 V.

2.3 Komunikacija Arduina s računalom (voltmetar kao primjer)

Tokom svog rada Arduino može serijskom vezom, tj. putem USB-utičnice, slati podatke računalu. Kao primjer prikazujemo program kojime Arduino, jednom u sekundi, računalu šalje vrijednost napona koju

¹LED je poluvodički element koji struju propušta samo u jednom smjeru i pritom svijetli. Za ispravan rad LED-a obično je potreban napon od oko 3 V. Prevelik napon, odn. struja, oštećuje LED.

mjeri na svom pinu A0. Računalo može, na primjer, te vrijednosti ispisivati na ekranu. Arduino IDE u tu svrhu ima tzv. *serial monitor tool*, što je prozor u kojem se u realnom vremenu prikazuje sve što Arduino serijskom vezom šalje računalu.

```

1 /* arduino primjer: voltmetar 0-5 V, serijska komunikacija */
2
3 void setup() {
4     Serial.begin ( 9600 );           // otvaramo serijsku vezu brzine 9600 bps
5 }
6
7 void loop() {
8     int a0val;                       // varijabla za vrijednost adc
9     a0val = analogRead ( A0 );       // ocitanje pina A0
10    Serial.print ( "Napon na A0: " ); // zapocinjemo redak
11    Serial.print ( 5. * a0val / 1023. ); // racunamo i ispisujemo napon
12    Serial.println ( "V" );           // završavamo redak
13    delay ( 1000 );                  // čekamo 1000 ms
14 }
```

Mjerenje se izvodi pozivajući funkciju `analogRead()` nad pinom A0 iz grupe ANALOG IN. Ta funkcija vraća cjelobrojnu vrijednost u rasponu od 0 do 1023 koja odgovara naponu od 0 V do 5 V. Napon V računamo prema formuli

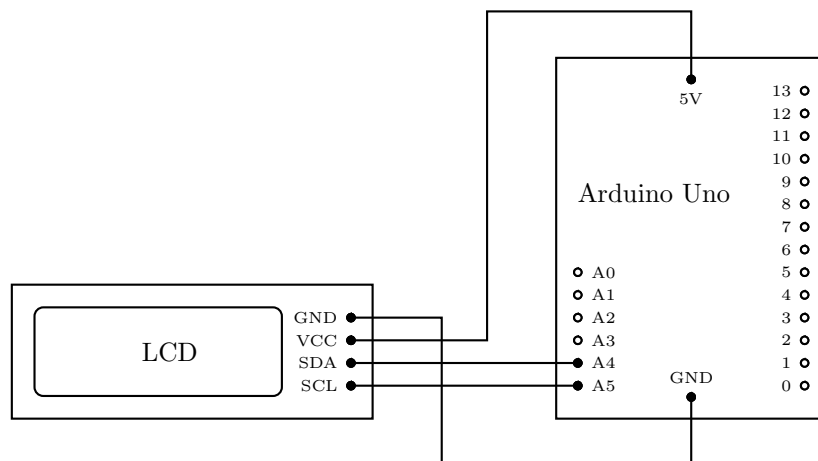
$$V = (5.0 \text{ V}) \times \frac{\text{AnalogRead}(A0)}{1023}.$$

2.4 Priključivanje LCD-ekrana (primjer “3, 2, 1, kreni!”)

Na modul Arduino Uno se vrlo jednostavno može priključiti maleni ekran s tekućim kristalima (liquid crystal display, LCD) na kojem Arduino, tokom svog rada, može ispisivati nama važne podatke. Shema na slici 4 prikazuje spoj Arduina i LCD-a, a jednostavan program kojim će se na LCD-u napisati “3, 2, 1, kreni”, pritom paleći i gasći pozadinsko svjetlo LCD-a, dan je u nastavku:

```

1 /* arduino primjer: korištenje lcd-ekrana, ispisuje "3, 2, 1, kreni" */
2
3 // naredne tri linije pozivaju biblioteke i pripremaju lcd-ekran
4 #include <Wire.h>
5 #include <LiquidCrystal_I2C.h>
6 LiquidCrystal_I2C lcd(0x3F,2,1,0,4,5,6,7,3,POSITIVE);
7
8 void setup() {
9     lcd.begin ( 16, 2 );              // lcd ima 2 retka sa 16 znakova
10    lcd.setBacklight ( HIGH );        // ukljuci pozadinsko svjetlo
11 }
12
13 void loop() {
14     lcd.clear ();                    // ocisti ekran
15     for ( int i = 3 ; i > 0 ; i-- ) { // i = 3, 2, 1
16         lcd.print ( i );             // napisi i u prvi redak
17         delay ( 500 );                // cekaj 500 ms
18         lcd.setBacklight ( LOW );     // iskljuci pozadinsko svjetlo
19         delay ( 500 );                // cekaj 500 ms
20         lcd.setBacklight ( HIGH );    // ukljuci pozadinsko svjetlo
21     }
22     lcd.setCursor ( 0, 1 );           // idi na pocetak drugog retka
23     lcd.print ( "kreni!" );           // napisi "kreni!" u drugi redak
24     delay ( 5000 );                  // cekaj 5 s pa ponovo
25 }
```



Slika 4: Priključivanje LCD-ekrana na modul Arduino Uno. Arduino koristi pinove A4 i A5 kako bi slao podatke LCD-u.

Arduino komunicira s LCD-om koristeći protokol poznat kao I²C čijim se potankostima ovdje ne bavimo. Primijetimo samo da su pinovi A4 i A5, čija je osnovna namjena očitavanje napona ADC-om, ovdje iskorišteni u svrhu komunikacije Arduinoa s LCD-om.

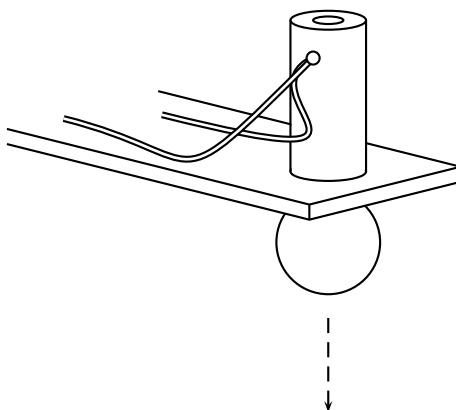
3 Eksperimentalni uređaj

Eksperimentalni uređaj se sastoji od sljedećeg:

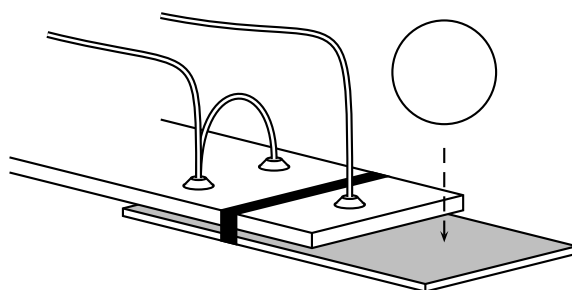
- Elektromehanički dio: stup visine oko 1 m na koji možemo, na željenim visinama, pričvrstiti dvije platforme. Na gornjoj platformi se nalazi elektromagnet, a na donjoj se nalazi prekidač. Tu je i kuglica koja ima ulogu padajućeg tijela te sitni pribor i alat.
- Elektronički dio: upravljački modul Arduino Uno, eksperimentalna pločica i LCD-ekran te potrebne elektroničke komponente (tranzistor, dugme, spojne žice).
- Programski dio: jednostavan Arduino program za upravljanje eksperimentalnim uređajem opisan je niže u ovom poglavlju. Vježba može uključivati modificiranje tog programa ili pisanje drugačijeg.
- Računalo (nije obavezno). Ako vježba se izvodi tako da uključuje programiranje mikrokontrolera, potrebno je i osobno računalo s instaliranim Arduino razvojnim okruženjem (Arduino IDE).

Platforma s elektromagnetom koji ima ulogu aktuatora prikazana je na slici 5. Jezgra elektromagneta prolazi kroz platformu te s njene donje strane na nju možemo prisloniti čeličnu kuglicu. Ako elektromagnetom teče struja, kuglica ostaje visjeti. U trenutku u kojem struja prestane teći, kuglica započinje slobodni pad.

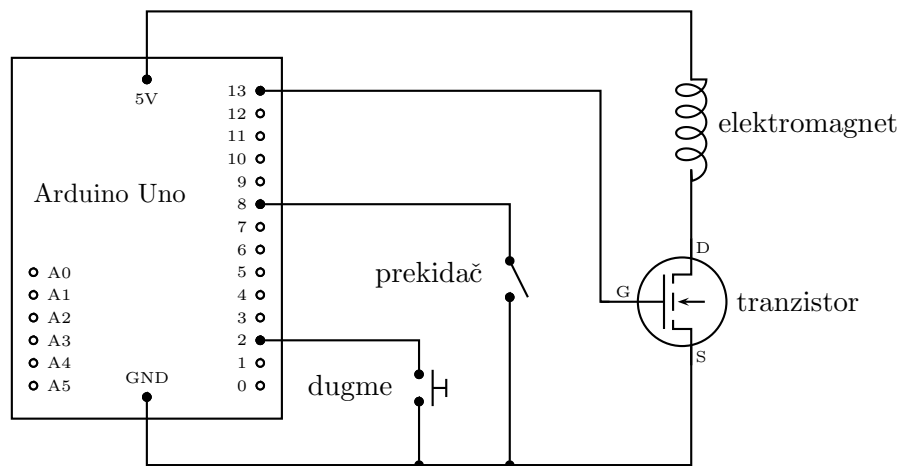
Platforma s prekidačem koji ima ulogu senzora prikazana je na slici 6. Na platformi se nalazi nekoliko bakrenih zakovica na koje odozdo prislanjamo pločicu koja sa svoje gornje strane ima sloj bakra. Kada je pločica dobro prislonjena, ona ostvaruje električni kontakt među zakovicama. Osim što dodiruje zakovice, pločica s prednje strane viri izvan same platforme. Kada padajuća kuglica stigne do pločice i udari u nju, pločica se pomakne i kontakt među zakovicama se prekida. Kako bi kuglica dovoljno dobro prijanjala na zakovice sve do udara kuglice, a kako bi se što je moguće lakše pomakla prilikom udara kuglice, pričvršćujemo ju elastičnom vezicom (gumicom).



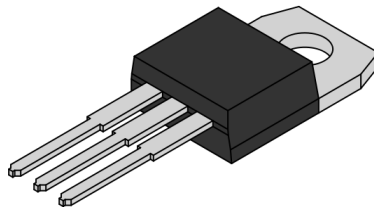
Slika 5: Aktuator: Na gornjoj platformi eksperimentalnog uređaja nalazi se elektromagnet. Kada elektromagnetom teče električna struja, on “drži” kuglicu prislonjenu s donje strane. Kada struja prestane teći, kuglica započinje slobodni pad.



Slika 6: Senzor (prekidač): Na donjoj platformi eksperimentalnog uređaja nalaze se zakovice. Na zakovice je elastičnom vezicom (crno) odozdo pričvršćena vodljiva pločica (sivo) čime se ostvaruje kontakt među zakovicama. Pri udaru kuglice u pločicu ona se pomiče i kontakt među zakovicama se prekida.



Slika 7: Shema sklopa za upravljanje padostrojem s pomoću modula Arduino Uno. Elektromagnet se napaja iz priključnice 5V. Pin 13 upravlja tranzistorom koji u krugu elektromagneta ima ulogu prekidača.



Slika 8: Izgled elektroničke komponente u kućištu TO220. Kada je riječ tranzistoru tipa MOSFET, njegovi izvodi su redom, idući s lijeva na desno, G (gate), D (drain) i S (source).

Elektronički sklop koji uključuje modul Arduino Uno, elektromagnet, prekidač i dugme prikazan je na slici 7. Dodatno, na Arduino treba spojiti LCD prema shemi na slici 4. Na pin 2 spojeno je dugme kojime ćemo Arduinou dati znak da prekine struju u elektromagnetu te da zabilježi trenutak u kojem je započeo pad kuglice. Prekidač u koji kuglica udara nakon što prevali postavljenu visinsku razliku spojen je na pin 8. U trenutku u kojem Arduino detektira da je taj prekidač otvoren, odn. da je kuglica u njega udarila, zabilježit će trenutak u kojem je pad završio. Razlika dvaju zabilježenih trenutaka jest trajanje pada kuglice.

Važno je uočiti da se elektromagnet napaja iz priključnice 5V, a ne izravno iz pina 13 kao što se u ranijem primjeru napajao LED. Razlog ovakvom rješenju je činjenica da je struja koja mora teći elektromagnetom veća od najveće dopuštene struje koju smijemo “povući” iz bilo kojeg od pinova iz grupe DIGITAL modula Arduino Uno. (Najveća dopuštena struja po pinu iznosi oko 40 mA, a našim elektromagnetom, kada je priključen na napon 5 V, teče struja jakosti približno 130 mA.) Pin 13 ovdje koristimo isključivo za upravljanje tranzistorom koji se nalazi u električnom krugu elektromagneta i ima ulogu prekidača. Koristimo tranzistor tipa MOSFET (engl. metal oxide field effect transistor). Njegove izvode D (drain) i S (source) možemo shvatiti kao izvode otpornika čiji otpor ovisi o razlici potencijala između njegovog G (gate) izvoda i njegovog S izvoda, V_{GS} . Kada je G na istom potencijalu kao i S, odn. $V_{GS} = 0$ (G je na niskoj logičkoj razini), otpor između D i S je velik te tranzistorom od D prema S ne teče značajna struja (prekidač je otvoren). Kada je $V_{GS} = 5V$ (G je na visokoj logičkoj razini), otpor između D i S postaje malen te tranzistorom od D prema S može teći značajna struja (prekidač je zatvoren). U ovoj vježbi koristimo MOSFET IRF540 koji se proizvodi u kućištu TO220. Fizički izgled tog tranzistora prikazan je na slici 8.

U nastavku se nalazi jednostavan program za upravljanje eksperimentalnim uređajem:

```

1  /* arduino: osnovni program za padostroj s lcd-om (6. srpnja 2015.) */
2
3  // naredne tri linije pozivaju potrebne biblioteke i pripremaju lcd-ekran:
4  #include <Wire.h>
5  #include <LiquidCrystal_I2C.h>
6  LiquidCrystal_I2C lcd(0x3F,2,1,0,4,5,6,7,3,POSITIVE);
7
8  // pinovi koje koristimo:
9  #define pinMagnet 13 // upravljanje elektromagnetom preko tranzistora
10 #define pinSwitch 8  // prekidač pri dnu stupa
11 #define pinButton 2  // dugme na eksperimentalnoj pločici
12
13 void setup() {
14     lcd.begin ( 16, 2 );           // lcd ima 2 retka sa 16 znakova
15     lcd.setBacklight( HIGH );      // ukljuci pozadinsko svjetlo
16     lcd.print ( "Trajanje_pada:" ); // napisi u gornji redak
17     pinMode ( pinMagnet, OUTPUT );
18     pinMode ( pinSwitch, INPUT_PULLUP );
19     pinMode ( pinButton, INPUT_PULLUP );
20 }
21
22 void loop() {
23     unsigned long t0, t1;          // trenuci pocetka i zavrsetka pada
24     digitalWrite ( pinMagnet, HIGH ); // ukljuci elektromagnet
25     while ( HIGH == digitalRead ( pinButton ) ); // cekaj do pritiska dugmeta
26     digitalWrite ( pinMagnet, LOW ); // prekini struju u elektromagnetu
27     t0 = micros ();                // zabiljezi trenutak pocetka pada
28     while ( LOW == digitalRead ( pinSwitch ) ); // cekaj do prekida kontakta
29     t1 = micros();                 // zabiljezi trenutak zavrsetka pada
30     lcd.setCursor( 0, 1 );          // idi na pocetak donjeg retka lcd-a
31     lcd.print ( t1 - t0 );           // napisi trajanje pada
32     lcd.print ( "µmicrosecµ" );     // napisi mjernu jedinicu
33 }

```

Većinu elemenata gornjeg programa već smo susreli u primjerima u poglavljima 2.2 i 2.4, a posebnu pažnju ovdje zahtijevaju linije 25 i 28. Kada izvršavanje programa dođe do tih linija, u njima se opetovano očitava stanje pina sve dok funkcija `digitalRead()` ne “vrati” traženo stanje, odnosno, sve dok se pin ne dovede na potencijal koji odgovara traženoj logičkoj razini. Čim funkcija `digitalRead()` vrati traženu vrijednost, izvršavanje te linije je završeno i prelazi se na sljedeću. Može se reći da te linije zaustavljaju izvršavanje programa, odnosno da Arduino “čeka” da pin dođe na željenu logičku razinu. U liniji 25 čekamo da eksperimentator pritisne dugme nakon čega Arduino gotovo trenutno prekida struju u elektromagnetu te počinje slobodan pad. U liniji 28 čekamo da kuglica udarcem u pločicu otvori prekidač što označava da je kuglica prevalila postavljenu visinsku razliku. Važno je napomenuti da ovakvo jednostavno rješenje ne predstavlja primjer dobre programerske prakse jer je tok izvršavanja programa na neko vrijeme zaustavljen te mikrokontroler ne može obavljati niti jednu drugu korisnu aktivnost. Naprednije rješenje moguće je uz korištenje tzv. prekidnog potprograma (engl. interrupt service routine, ISR) što ujedno otvara prostor za unapređenje ovog eksperimentalnog uređaja.

4 Zadaci

U ovisnosti o raspoloživom vremenu i o nastavnom programu zadaci se mogu izvoditi svi ili samo neki od njih. Prva dva zadatka bave se sastavljanjem eksperimentalnog uređaja, dok u preostala dva zadatka njime obavljam mjerena i analiziramo ih. Izostavi li se zadatak 4.1 u kojem student samostalno sastavlja elektronički dio uređaja, studenta mora dočekati ispravno sastavljen uređaj. Zadatak 4.2 koji se bavi programiranjem mikrokontrolera može se izvoditi jedino ako je studentu na raspolaganju i računalo s instaliranim Arduino razvojnim okruženjem (Arduino IDE). Izostavi li se taj zadatak, studenta mora dočekati Arduino u kojega je upisan program prikazan u prethodnom poglavlju.

4.1 Sastavljanje elektroničkog dijela uređaja

Na radnom stolu ćete prepoznati stup s platformama, ploču na kojoj su učvršćeni Arduino, eksperimentalna pločica i LCD te kutiju s elektroničkim komponentama i priborom. Upoznajte se sa sadržajem kutije (prepoznajte spojne žice, tranzistore, LED u raznim bojama, otpornike $220\ \Omega$, dugme). Upoznajte se sa eksperimentalnom pločicom te rasporedom kontakata unutar nje. U tome će vam pomoći papiri s dodatnim uputama i pojašnjenjima koje ćete također pronaći na radnom stolu. Zatim započnite sastavljanje elektroničkog dijela uređaja prema shemama na slikama 4 i 7. Možete postupati ovim redom:

1. Priključivanje napajanja na eksperimentalnu pločicu. Uz rub eksperimentalne pločice nalazi se stupac međusobno povezanih utičnica označen plavim “-” (minus) znakom koji spojn timer žicom priključujemo na jednu od GND utičnica Arduina. Do njega se na eksperimentalnoj pločici nalazi stupac označen crvenim “+” (plus) znakom koji drugom spojn timer žicom priključujemo na 5V utičnicu Arduina. Time smo omogućili jednostavan pristup napajanju pri izgradnji sklopova na eksperimentalnoj pločici.
2. Spajanje dugmeta na pin 2. Izvode dugmeta ubadamo u dva različita retka na eksperimentalnoj pločici. Jedan od tih redaka zatim spojn timer žicom povezujemo sa stupcem spojenim na GND pin, dok drugi redak spojn timer žicom izravno priključujemo na pin 2 Arduina. (Ako je na raspolaganju računalo, ispravan rad dugmeta može se provjeriti programom opisanim u primjeru 2.2.)
3. Spajanje tranzistora. Tri izvoda tranzistora treba utaknuti u tri susjedna retka eksperimentalne pločice. Redak u kojem se nalazi G (gate) izvod tranzistora spojn timer žicom povezujemo izravno s pinom 13 Arduina. Redak u kojem se nalazi S (source) izvod spojn timer žicom povezujemo sa stupcem GND. Redak u kojem se nalazi D (drain) za sada ne priključujemo ni na što.
4. Spajanje LCD-a prema shemi na slici 4. Ispravan rad LCD-a može se provjeriti programom iz primjera 2.4.
5. Priključivanje elektromagneta (aktuatora). Jedan od vodova elektromagneta spajamo u redak eksperimentalne pločice u kojem se nalazi D (drain) tranzistora, a drugi vod spajamo u stupac spojen na 5V pin Arduina.
6. Priključivanje prekidača (senzora). Jedan od vodova prekidača spajamo u stupac eksperimentalne pločice priključen na GND pin, dok drugi vod spajamo izravno na pin 8 Arduina.

Ovime je elektronički dio eksperimentalnog uređaja sastavljen.

4.2 Unos i prilagodba programa

Prvi dio zadatka oslanja se na jednostavan program opisan u prethodnom poglavlju:

1. Koristeći Arduino razvojno okruženje (Arduino IDE) utipkajte ili učitajte program iz prethodnog poglavlja.
2. Najprije provjerite formalnu ispravnost programa pritiskom na “kvačicu”, a zatim ga pritiskom na “strelicu” snimate na Arduino.
3. Provjerite ispravnost rada čitavog eksperimentalnog uređaja. Napravite nekoliko mjerenja na raznim visinama kako biste stekli osjećaj za podešavanje prekidača na donjoj platformi.

Zatim samostalno provedite sljedeća unapređenja programa:

4. Uvedite vremenski razmak od 3 s između trenutka u kojem je pritisnuto dugme i trenutka u kojem program prekida struju u elektromagnetu. Taj vremenski interval vam dopušta da se, nakon što pritisnete dugme, usredotočite na pad kuglice te njen udarac u prekidač (ponekad je potrebno uhvatiti kuglicu da se ne otkotrlja na pod). Možete također uvesti odbrojavanje i bljeskanje ekrana kao u primjeru 2.4.
5. Omogućite ispisivanje rezultata mjerenja, osim na LCD-u, također i na ekranu računala. Rukovodite se primjerom 2.3.
6. Uvedite cjelobrojnu varijablu koja bilježi redni broj mjerenja počevši od pokretanja mikrokontrolera te omogućite da se taj broj ispisuje na LCD-u kao i na ekranu računala uz svako mjerenje. (Uvijek možete krenuti od nule pritiskom na dugme RESET koje se nalazi na samom modulu Arduino Uno.) Taj redni broj može značajno olakšati prikupljanje i obradu mjerenja.

Uz ova unapređenja s uređajem će biti jednostavnije raditi.

4.3 Određivanje akceleracije na osnovu trajanja pada s odabrane visine

Cilj ovog zadatka je odrediti akceleraciju slobodnog pada na osnovu mjerenja trajanja pada t s odabrane visine h .

1. Gornju platformu učvrstite na odabranoj visini iznad donje platforme. Uključite elektromagnet i objesite kuglicu na njega. Metrom izmjerite visinsku razliku h između donjeg tjemena kuglice i gornje plohe pločice prekidača, odnosno visinsku razliku koju kuglica prevaljuje od početka pada do udara u pločicu prekidača. Procijenite nesigurnost tog mjerenja i konačni rezultat mjerenja prikažite u standardnom obliku.
2. Ne mijenjajući visinsku razliku h koju ste postavili i izmjerili u prethodnom koraku, izmjerite trajanje pada kuglice t barem 10 puta. Izračunajte aritmetičku sredinu mjerenja $\bar{t}$ i standardnu devijaciju aritmetičke sredine $\sigma_{\bar{t}}$ prema poznatim izrazima. Rezultat mjerenja trajanja pada kuglice prikažite u standardnom obliku.
3. Koristeći rezultate mjerenja visinske razlike h i trajanja pada t , odredite iznos akceleracije g prema izrazu

$$g = \frac{2h}{t^2}$$

te odgovarajuću nesigurnost. Rezultat prikažite u standardnom obliku.

4. Ispitajte koja od dviju mjerenih veličina svojom nesigurnošću ima veći utjecaj na nesigurnost pri određivanju iznosa akceleracije g .

4.4 Određivanje akceleracije mjerenjem trajanja pada s različitih visina

Za razliku od prethodnog zadatka u kojem ste akceleraciju slobodnog pada odredili na osnovu mjerenja samo jedne visine h i njoj odgovarajućeg trajanja pada t , ovdje ćete akceleraciju odrediti na osnovu mjerenja trajanja pada s niza različitih visina. Takvom kompleksnijom procedurom ćete uz jednak ili manji broj samih mjerenja iznos akceleracije odrediti uz manju mjernu nesigurnost.

1. Gornju platformu eksperimentalnog uređaja učvrstite pri vrhu, a donju platformu učvrstite pri dnu stupa. Na visinskoj skali ucrtanoj na samom stupu očitajte položaj x_1 gornje ivice klizača gornje platforme. (Zbog jednostavnosti, uputno je unaprijed postaviti gornju ivicu klizača na “okruglu vrijednost”, npr. $x_1 = 95$ cm.) Položaj klizača donje platforme dovoljno je grubo očitati. Izmjerite trajanje pada kuglice t_1 za postavljenu visinsku razliku. Zatim snizite gornju platformu za 10 cm i izmjerite trajanje pada t_2 za novu visinsku razliku. Nastavite snižavati gornju platformu u koracima od 10 cm i mjeriti vremena t_i , sve dok visinsku razliku ne smanjite na dvadesetak centimetara. Mjerenja upišite u tablicu kao

$$(x_i, t_i) \quad i = 1, \dots, N, \quad (2)$$

gdje je N ukupan broj mjerenja (za naš uređaj $N = 7$ ili 8). Važno je primijetiti da visina x_i ne odgovara izravno visinskoj razlici h_i koju je kuglica prevalila padajući, već vrijedi

$$h_i = x_i - a, \quad (3)$$

gdje je a za sada nepoznati parametar povezan s položajem donje platforme uređaja. (Pokazat će se da u postupku koji slijedi točan položaj donje platforme ne igra važniju ulogu u određivanju akceleracije slobodnog pada.)

2. Mjerenja prikupljena u prethodnom koraku prikažite grafički koristeći vodoravnu x -os i uspravnu t -os. Uočite nelinearnu ovisnost trajanja pada o visinskoj razlici. Ona nas ne iznenađuje jer na osnovu relacije (1) očekujemo $h_i = x_i - a = (g/2)t_i^2$, odnosno

$$t_i = \sqrt{\frac{2h_i}{g}} = \sqrt{\frac{2(x_i - a)}{g}}. \quad (4)$$

Uočite također da iz nacrtanog grafa nije jednostavno “očitati” vrijednosti za sada nepoznatih parametara a i g .

3. Kako biste dobili linearnu ovisnost među promatranim veličinama uvedite novu veličinu

$$y_i = \frac{t_i^2}{2}, \quad (5)$$

izračunajte njene vrijednosti za mjerenja $i = 1, \dots, N$ i unesite ih u tablicu. Parove (x_i, y_i) prikažite grafički koristeći vodoravnu x -os i uspravnu y -os. Uočite da mjerenja prikazana u x, y -ravnini “gotovo savršeno” leže na pravcu. To je očekivano jer na osnovu (1) imamo $h_i = x_i - a = gy_i$, odnosno

$$y_i = \frac{1}{g} x_i - \frac{a}{g}. \quad (6)$$

što govori da je koeficijent smjera pravca na kojem leže parovi (x_i, y_i) recipročna vrijednost akceleracije slobodnog pada $1/g$ te da je njegov odsječak na y -osi povezan s parametrom a , odnosno, s položajem donje platforme.

4. Na grafu koji ste nacrtali, ravnalom nacrtajte pravac koji po vašem “osjećaju” prolazi najbliže svim točkama. Procijenite koeficijent smjera nacrtanog pravca $1/g$ te izračunajte akceleraciju slobodnog pada g .
5. Akceleraciju slobodnog pada g koju ste u prethodnom koraku procijenili iz grafa sada odredite računski primjenom metode najmanjih kvadrata. Za provedbu računa možete koristiti mrežni alat <http://sail.zpf.fer.hr/sdmt>. Rezultat prikažite u standardnom obliku.

Literatura

- [1] M. Banzi, “Getting Started with Arduino”, 2. izdanje (2011), O’Reilly