Fiksētie slēdži

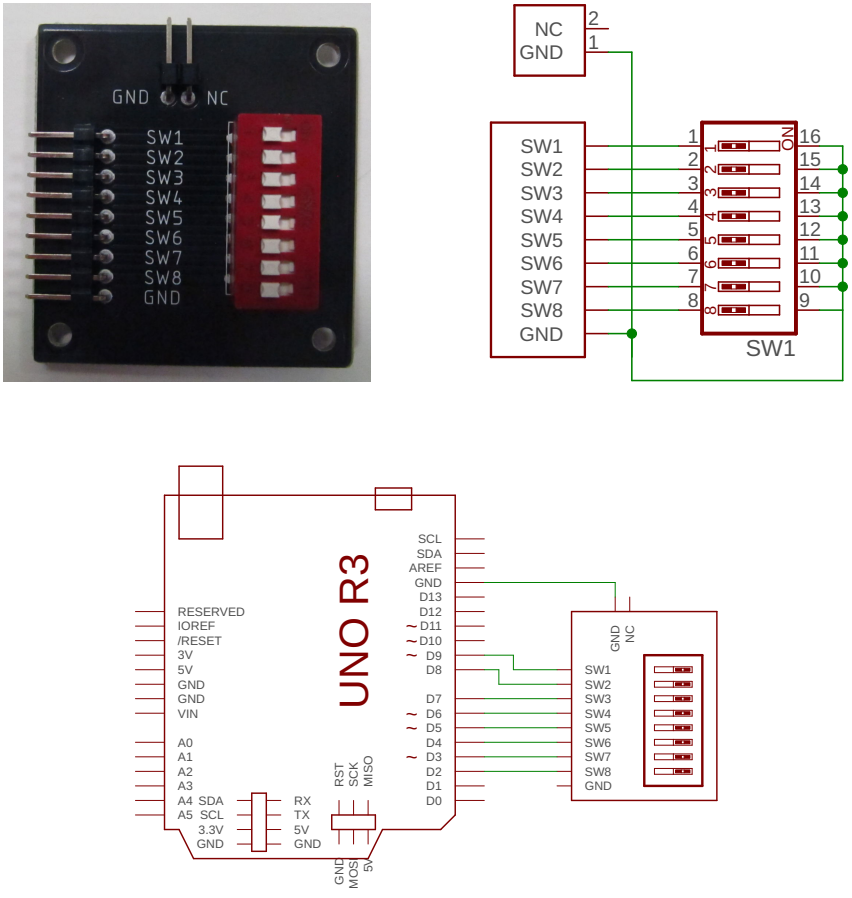
Ivars Driķis

2019. gada 4. septembrī

1 Dip slēdžu bloks

Daudzos gadījumos, izstrādājot Arduino balstītas iekārtas, ir nepieciešami fiksēta stāvokļa slēdži, pie kam bieži vien daudz. Piemēram, lai pārslēgtu cīņas stratēģiju Sumo robotam tieši pirms cīņas pēc pretinieka izvērtēšanas. Parasti izmanto DIP slēdžus. Ta darīju arī es, bet pēc divām studentu nodarbību sezonām mani DIP slēdži sāka pa vienam nojukt. Meklēju citus ilglaicīgākus, bet pietiekami lētus risinājumus un nonācu …pie DIP slēdžiem ar resursu 10000 pārslēgumu [1]. Panelis iznāca vienkāršs, kā tas redzams 1 zīmējumā.

Att. 1: Fiksēto slēdžu panelis, tā shēma un savienojums ar Arduino, kā tas lietots tālākajos piemēros.



2 Kāpņu telpas slēdži

Lai ekonomētu enerģiju, kāpņu telpās katrā stāvā izvieto slēdžus apgaismojuma ieslēgšanai. Diennakts tumšajā laikā izejot no dzīvokļa, stāva vai ieejas kāpņu telpā pārslēdzam tuvāko slēdzi un gaisma iedegās. Pēc tam, izejot no kāpņu telpas, atkal pārslēdzam tuvāko slēdzi (citu) un gaisma kāpņu telpā nodziest.

Kā piemēru izveidosim datorprogrammu, kas darbojas tieši šādā veidā: pārslēdzot jebkuru no fiksētajiem slēdžiem pārslēdzās iebūvētā spīddiode. Algoritms ir sekojošs: nolasām slēdzu stāvokli vienu pēc otra un izveidojam mainīgo, kurā iekodēts *visu* slēdžu šo stāvoklis. Pēc tam šo jauniegūto vērtību salīdzina ar iepriekšējo un, ja atšķirās, tad pārslēdz iebūvēto mirdzdiodi.

Vispirms norādīsim, pie kurām konkrēti Arduino ieejām ir pieslēgti slēdži. Šī programmas daļa ir vienlaikus arī dokumentācija, kas palīdzēs atkārtoti izmantot šo programmu pareizi pieslēdzot slēdžu paneli.

Listing 1: StearSwitch.ino

```
1 const int pinSwi1 = 9;
2 const int pinSwi2 = 8;
3 const int pinSwi3 = 7;
4 const int pinSwi4 = 6;
5 const int pinSwi5 = 5;
6 const int pinSwi6 = 4;
7 const int pinSwi7 = 3;
8 const int pinSwi8 = 2;
```

Tālāk definējam nepieciešamos globālos mainīgos un piešķiram tiem sakuma vērtības.

```
10 int prewSwiState = -1;
11 int ledState = HIGH;
```

Šeit

prewSwiState - mainīgais, kurā tiek iekodēts *iepriekšējais* slēdžu stāvoklis. Piešķiram tam vērtību, kuru *nav* iespējams iegūt iekodējot slēdžu stāvokli.

ledState - norāda spīddiodes vadošā signāla vērtību. Kāpēc HIGH, redzēsim vēlāk.

Tālāk konfigurējam Arduino ieejas/izejas līnijas un uzstādām spīddiodes līnijas izejā mainīgajam ledState atbilstošu vērtību.

```
13 void setup() {
14   pinMode(pinSwi1, INPUT_PULLUP);
15   pinMode(pinSwi2, INPUT_PULLUP);
16   pinMode(pinSwi3, INPUT_PULLUP);
17   pinMode(pinSwi4, INPUT_PULLUP);
18   pinMode(pinSwi5, INPUT_PULLUP);
19   pinMode(pinSwi6, INPUT_PULLUP);
20   pinMode(pinSwi7, INPUT_PULLUP);
21   pinMode(pinSwi8, INPUT_PULLUP);
22
23   pinMode(LED_BUILTIN, OUTPUT);
24   digitalWrite(LED_BUILTIN, ledState);
25 }
```

Šeit izmantota funkcijas pinMode(pin, mode) un digitalWrite(pin, value)

pinMode(pin, mode) - konfigurē norādīto digitālo līniju vai nu par ieeju vai arī par izeju, [2]:

pin: digitālās līnijas numurs

mode: ja OUTPUT, tad izeja, ja INPUT vai INPUT_PULLUP tad ieeja.

digitalWrite(pin, value) - norādīto pinu uzstāda zemā vai augstā līmenī, [3]

pin: digitālās līnijas numurs

value: spriegums digitālās līnijas izejā, LOW vai HIGH

Neaizmirstam, ka gadījumos, kad ieejām pieslēgti slēdži, ir jālieto INPUT_PULLUP opcija, kas ieejai pieslēdz iebūvētos rezistoru un ar tā palīdzību signāla līmeni paceļ līdz HIGH. Ieslēdzot slēdzi ieeja tiek savienota ar GND, tādējādi signāls ieejā kļūst LOW.

Tālāk programmas galvenā daļa. Vispirms pa vienam nolasām slēdžu stāvokļus, katru no tiem ierakstot vienā no mainīgā state *bitiem* izmantojot funkciju bitWrite. Tālāk salīdzinām tik tiko nolasīto slēdzu stāvokli ar stāvokli, kāds ir noglabāts mainīgajā prewSwiState. Ja atšķirās, tad pārslēdz LED neaizmirstot noglabāt jauno vērtību mainīgajā ledState, gan arī saglabājot mainīgajā prewSwiState jauno slēdžu stāvokli. Pēc tam pagaidām 0.1 sekundi un visu atkārtojam vēlreiz.

Te nepieciešams izsekot tam, kas notiek pēc pirmās slēdžu pārbaudes. Tā kā nolasot no slēdžiem un iekodējot vienmēr iegūstam pozitīvu skaitli, bet sākotnēji prewSwiState ierakstīts negatīvs skaitlis, tad pirmajā reizē noteikti tiks mainīts LED stāvoklis un noglabāta jaunā tekošā vērtība. Tā kā pēc iekārtas ieslēgšanas apgaismojumam ir jābūt izslēgtam, tad sākumā ledState sākumā ir jābūt HIGH lai pēc pirmā pārbaudes cikla apgaismojums tiktu izslēgts.

```
27 void loop()
28 {
29   int state = 0;
30   bitWrite(state, 0, digitalRead(pinSwi1));
31   bitWrite(state, 1, digitalRead(pinSwi2));
32   bitWrite(state, 2, digitalRead(pinSwi3));
33   bitWrite(state, 3, digitalRead(pinSwi4));
34   bitWrite(state, 4, digitalRead(pinSwi5));
35   bitWrite(state, 5, digitalRead(pinSwi6));
36   bitWrite(state, 6, digitalRead(pinSwi7));
37   bitWrite(state, 7, digitalRead(pinSwi8));
38
39   if( state != prewSwiState ) {
40     ledState = !ledState;
41     digitalWrite(LED_BUILTIN, ledState);
42     prewSwiState = state;
43   }
44
45   delay( 100 );
46 }
```

Šeit izmantotas funkcijas bitWrite(x, n, b) un delay(ms).

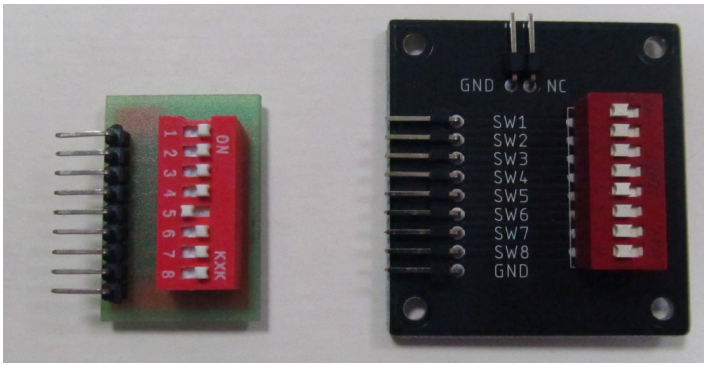
bitWrite(x, n, b) - mainīgajam x bitā n ieraksta vērtību b, kas ir 0 vai 1, [4]:

delay(ms) - aptur programmas izpildi uz noteiktu laiku [5];

ms: laiks milisekundēs no 0 līdz $2^{32} - 1$. Tātad ilgākais laiks ir apmēram 50 diennaktis!

Pateicības

Platīte izgatavota Latvijas Universitātes Fonda projekta "Programmējam ar prieku" ietvaros un to finansē AS Mikrotīkls.



Att. 2: Fiksēto slēdžu panelis tā vēsturiskajā attīstībā.

Literatūras saraksts

[1] C&K, BD Series standard profile DIP. https://www.ckswitches.com/products/switches/product-details/DIP/BD/. [Online; accessed 2018.09.24].

[2] Arduino Reference. pinMode(). https://www.arduino.cc/en/Reference/PinMode. [Online; accessed 2016.09.18].

[3] Arduino Reference. digitalWrite(). https://www.arduino.cc/en/Reference/DigitalWrite. [Online; accessed 2016.09.18].

[4] Arduino Reference. BitWrite(). https://www.arduino.cc/en/Reference/BitWrite. [Online; accessed 2016.10.18].

[5] Arduino Reference. delay(). https://www.arduino.cc/en/Reference/Delay. [Online; accessed 2016.09.18].