

27.9.2021

Tilakone

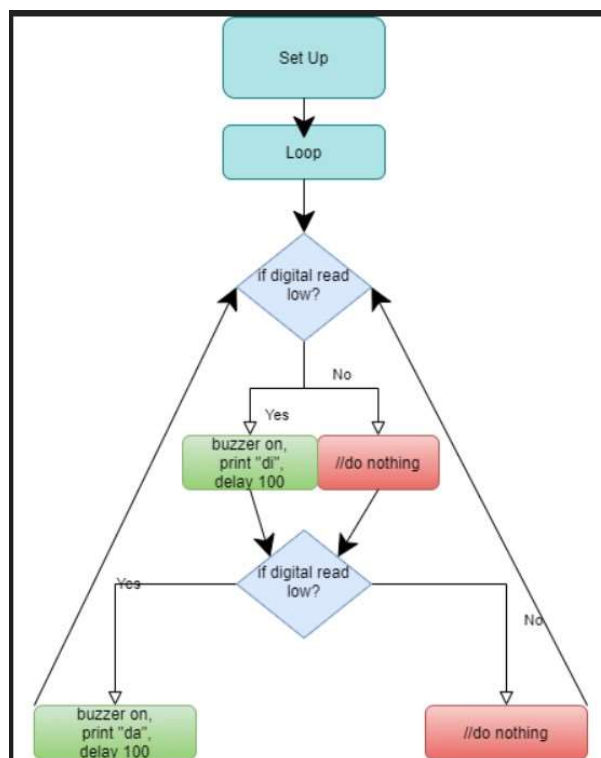
Tässä raportissa käsitellään Automaatiotekniikan ensimmäisen kysymyspatteriston vastauksia.

Kysymys 1

Ylösvetovastus on vastus, joka varmistaa, että elektronisen komponentin tuloliitäntä on ylätilassa eli looginen ykkönen silloin, kun liitäntään ei ole kytketty muuta signaalia.

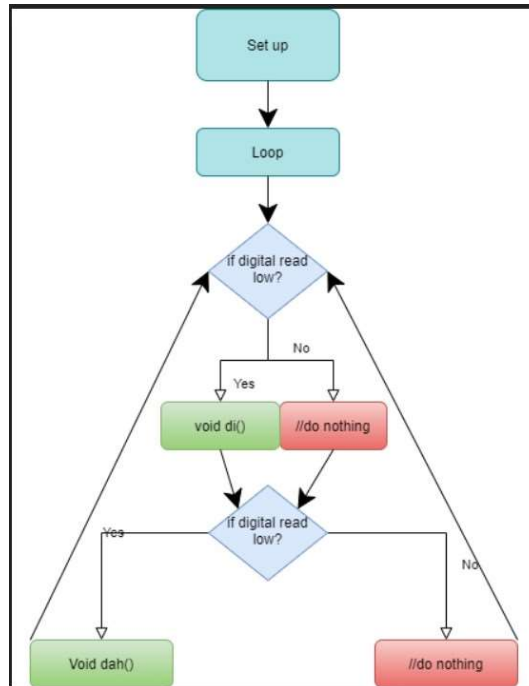
Avokollektorilähtöinen kytkentä ei ykköstilassa ollessaan anna ollenkaan jännitettä, mutta ylösvetovastuksella seuraavan komponentin tuloon saadaan oikea ykköstilaa merkitsevä jännite. Kun avokollektorilähtö siirtyy nollatilaan, transistori alkaa johtaa, jolloin ylösvetovastuksen kautta tuleva sähkövirta ei riitä pitämään lähtöä ykköstilassa ja jännite putoaa nollatilaan vastaavaksi. Transistorin tilalla voi olla myös vaikkapa kytkin, jolla tulon tila saadaan tarvittaessa pakotettua nol-laksi. Mikäli transistorin tilalle kytketään kondensaattori, asettuu jän-nite virran kytkemishetkellä nollatasolle, mutta kondensaattorin varau-duttua jännite nousee ykköstasoa vastaavaksi. Tällaista vastuksen ja kondensaattorin muodostamaa RC-piiriä käytetään, kun halutaan esi-merkiksi nollata mikropiiri virran kytkemishetkellä.

Kysymys 2



27.9.2021

Kysymys 3



Kysymys 4

Koodi suorittaa (ikuista) silmukkaa, josta se käy tarkastamassa ovatko tietyn koodin suoritusehdot täyttyneet, jos ovat niin ohjelma suorittaa ko. ohjelman, jos ei, niin ko. koodia ei suoriteta, vaan tarkastetaan välittömästi seuraava ehto.

```
automaatio6
void di() {
  tone(Buzzer, NOTE_D6, 60, BUZZER_CHANNEL);
  //digitalWrite(Buzzer, HIGH);
  delay(60); // ...for 0.06 sec
  //noTone(Buzzer); // Stop sound...
  noTone(Buzzer, BUZZER_CHANNEL);
  //digitalWrite(Buzzer, LOW);
  delay(100); // ...for 0.1 sec
}

void dah() {
  tone(Buzzer, NOTE_D6, 300, BUZZER_CHANNEL);
  delay(60); // ...for 0.06 sec
  noTone(Buzzer, BUZZER_CHANNEL);
  delay(100); // ...for 0.1 sec
}

void setup() {
  pinMode(sw1, INPUT);
  pinMode(sw2, INPUT);
  pinMode(LED, OUTPUT);
  Serial.begin(115200);
}

void loop() {
  // Yhtälö joka muuttaa painalluksen luvuksi
  // (0 tai 1) * 32 + (0 tai 1) * 33 = val
  val = ((digitalRead(sw1)*sw1) + (digitalRead(sw2)*sw2));
  switch(val) {
    case 32 :
      di();
      break;
    case 33 :
      dah();
      break;
  }
}
```

27.9.2021

Kysymys 5

Esimerkkikoodi toimii samalla tavalla kuin Switch Debouncing, eli ottaa vain yhden painalluksen. Switch Case ottaa ns. useamman painalluksen.

```
automaatio2  
bool invert = true;  
  
//lampulla jatkuva funktion kutsu  
//EasyButton diButton(sw1,debounce,pullup,invert);  
//EasyButton dahButton(sw2,debounce,pullup,invert);  
  
//toimiva maaritys  
EasyButton diButton(sw1);  
EasyButton dahButton(sw2);  
  
void setup() {  
  //maaritetaan mita tapahtuu kun nappia painetaan.  
  Serial.begin(115200);  
  diButton.begin();  
  dahButton.begin();  
  diButton.onPressed(di);  
  dahButton.onPressed(dah);  
}  
  
void di(){  
  // Aani paalle ja aani pois. Printtaa myös debugin.  
  Serial.println(" di");  
  tone(Buzzer, NOTE_G6, 60, BUZZER_CHANNEL);  
  noTone(Buzzer, BUZZER_CHANNEL);  
}  
  
void dah(){  
  // Aani paalle ja aani pois. Printtaa myös debugin.  
  Serial.println(" dah");  
  tone(Buzzer, NOTE_G6, 200, BUZZER_CHANNEL);  
  noTone(Buzzer, BUZZER_CHANNEL);  
}  
  
void loop() {  
  //lukee painetaanko nappia  
  diButton.read();  
  dahButton.read();  
}
```

Kysymys 6

ISR ei voi ottaa mitään parametrejä eikä voi palauttaa mitään jotta millis(), joka palauttaa ja delay(), joka tarvitsee parametrin eivät sovellu tähän.

Kysymys 7

Switch Debouncing toimii samalla tavalla kuin jos löisit vasaralla metallilevyyn. Mikäli tätä suoritusta katsottaisiin hidastetusti, nähtäisiin kuinka vasara osuisi useaan otteeseen metallilevyyn eli "pomp-pisi". Debouncilla poistetaan nämä useammat "pomput", eli mikäli tämä toiminta siirretään nappien painalluksiin, se poistaa "ylimääräiset" painallukset.

27.9.2021

```
automaatio5

pressTime = millis();
button_pressed = true;
soi = 60;
}

void IRAM_ATTR dah(){
pressTime = millis();
button_pressed = true;
soi = 200;
}

void setup(){
pinMode(Buzzer, OUTPUT); // Buzzer to pin 14
pinMode(Sw1, INPUT_PULLUP); // Switch1 to pin 32
pinMode(Sw2, INPUT_PULLUP); // Switch2 to pin33
Serial.begin(115200); // Serial port is open just for debugging purpose
attachInterrupt(32,di,FALLING);
attachInterrupt(33,dah,FALLING);
}

void loop(){
// Nappia painetaan, niin alkaa valobileet.
//Debounce varmistaa ettei klikata useampaa kertaa
if (button_pressed){
if ((millis() - pressTime) >deBounceDelay){
tone(Buzzer, NOTE_B5, 500, BUZZER_CHANNEL);
noTone(Buzzer, BUZZER_CHANNEL);
button_pressed = false;
}
else {
button_pressed = true;
}
}
else {
//DoNothing
}
}
}
```

Event – Listener

```
automaatio3

pinMode(14, OUTPUT);
pinMode(32, INPUT_PULLUP);
pinMode(33, INPUT_PULLUP);
Serial.begin(115200);

//Luotiin kuuntelijat
mgr.addListener(new EvtPinListener(32, (EvtAction)sw1pressed));
mgr.addListener(new EvtPinListener(33, (EvtAction)sw2pressed));

}

USE_EVENTUALLY_LOOP(mgr)

bool sw1pressed(){
// Maaritetaan piipaika, debugataan ja lopetataan.
mgr.resetContext();
beebtime = 60;
Serial.println("Di");
mgr.addListener(new EvtPinListener(32, (EvtAction)sw1pressed));
mgr.addListener(new EvtPinListener(33, (EvtAction)sw2pressed));

noTone(14, 0);
return true;
}

bool sw2pressed(){
// Maaritetaan piipaika, debugataan ja lopetataan.
mgr.resetContext();
beebtime = 200;
Serial.println("Dah");
mgr.addListener(new EvtPinListener(32, (EvtAction)sw1pressed));
mgr.addListener(new EvtPinListener(33, (EvtAction)sw2pressed));

noTone(14, 0);
return true;
}
}
```

Noora Angelva
Sanna Inkerö
R54T19S

Raportti
Automaatiotekniikka

27.9.2021

Noora Angelva
Noora Angelva

Sanna Inkerö
Sanna Inkerö