

State Machine Exercise

Tässä tulee ohjeita harjoitustyön suorittamiseksi.

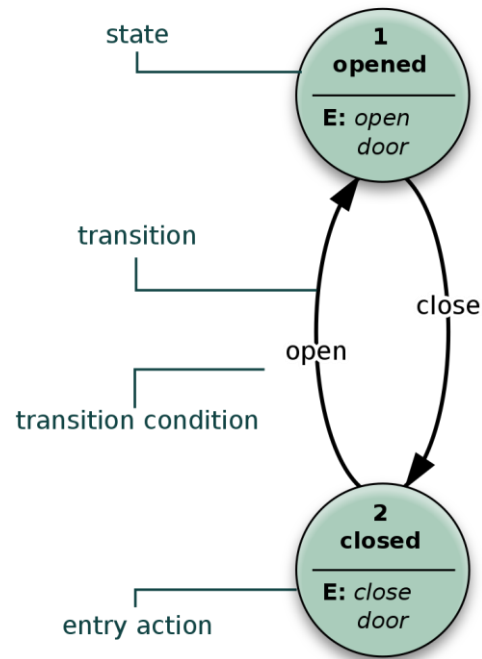
State Machine eli tilakone on ohjelmistoarkkitehtuuri, jota käytetään hyvin paljon automaatiojärjestelmissä ja robotiikassa.

1. State Machine Exercise

State Machine eli tilakone

https://en.wikipedia.org/wiki/Finite-state_machine

- Finite-state machine

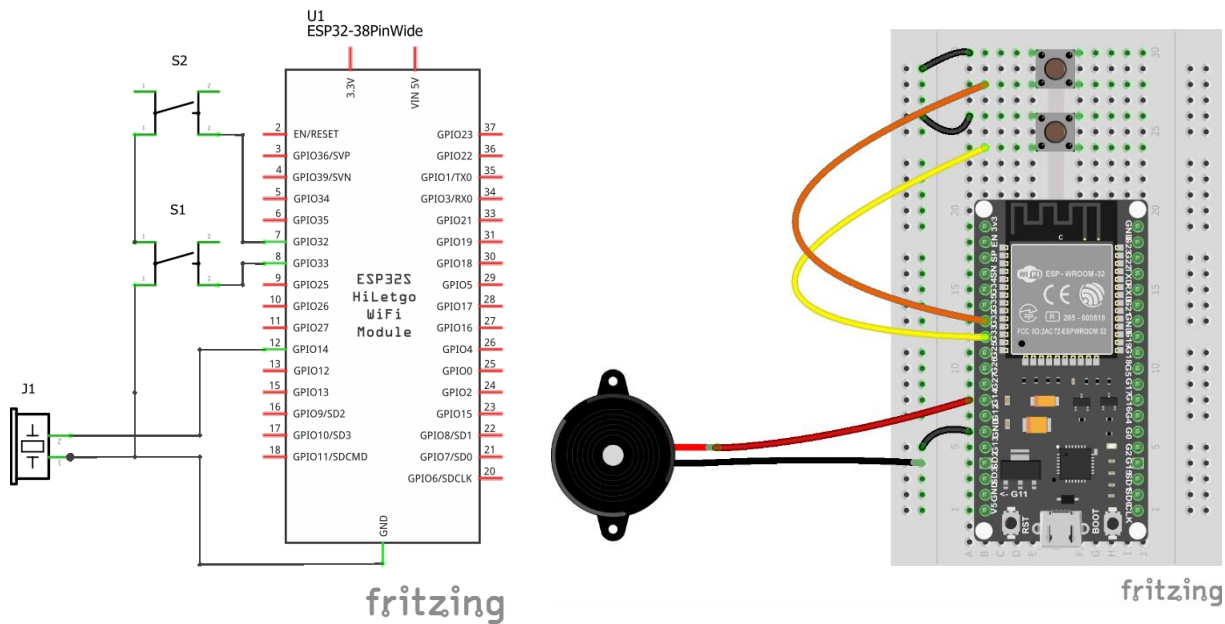


Vastaavatko
nämä toisiaan?
No ei....

```
/*State-Machine pseudokoodia  
 * 24th Aug. 2019 by Mr T  
 */
```

```
void setup() {  
    // Alkuasetukset  
  
}  
void loop() {  
    if (open == TRUE) {  
        closed()  
    }  
    if (close == TRUE) {  
        opened()  
    }  
}  
void opened() {  
    // Sulkee oven  
}  
  
void closed() {  
    // Avaa oven  
}  
}
```

ESP32 kytkentä



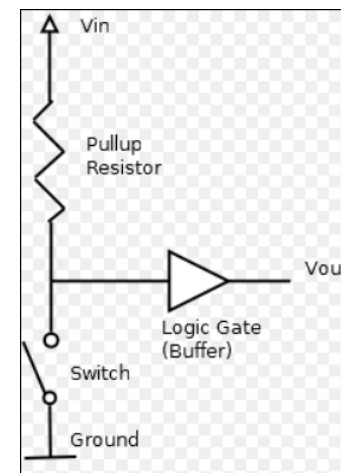
Esimerkiksi tällainen kytkentä

Piezo buzzer (GPIO14). Voi siihen kytkeä LEDin

Jos piipitys käy hermoille

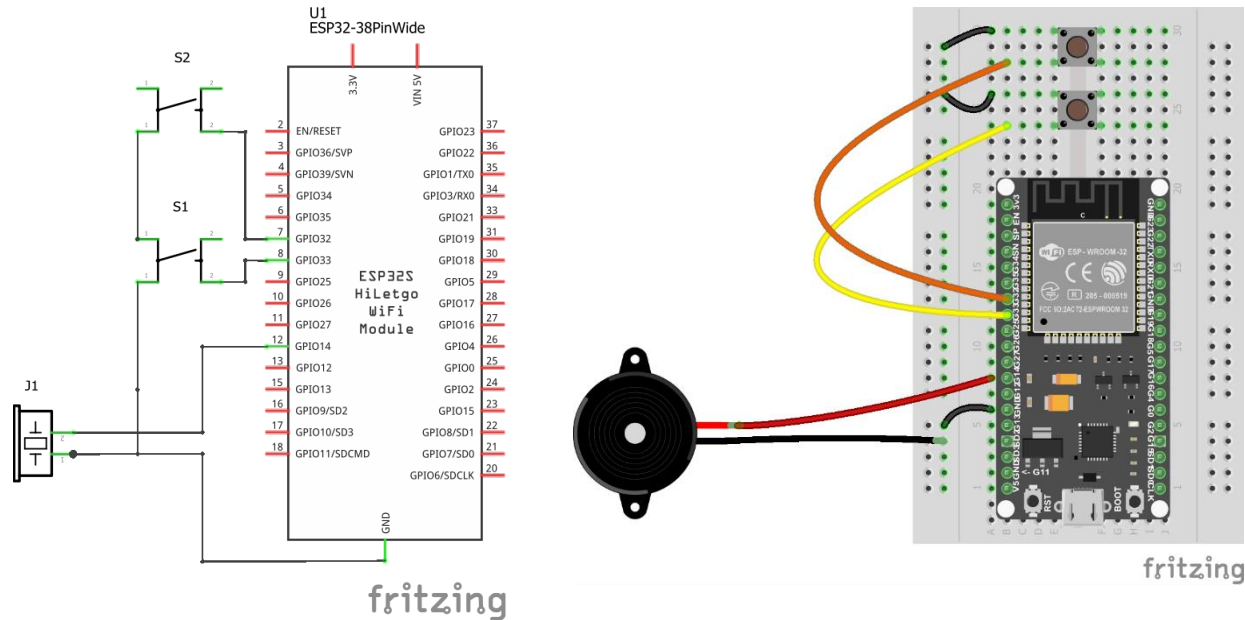
Kytkin 1 (GPIO32), kytkin2(GPIO33)

Käytetään sisäistä ylösvetovastusta!



Kysymys 1: Selitä ylösvetovastuksen (pullup) periaate! Miksi sitä tarvitaan?

If-Else rakenne...



```
#include < pitches.h> // Library for notes and frequencies
#include < Tone32.h> // Library to generate tones
#define Buzzer 14
#define BUZZER_CHANNEL 0
```

```
int Sw1 = 32;
int Sw2 = 33;
```

```
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
```

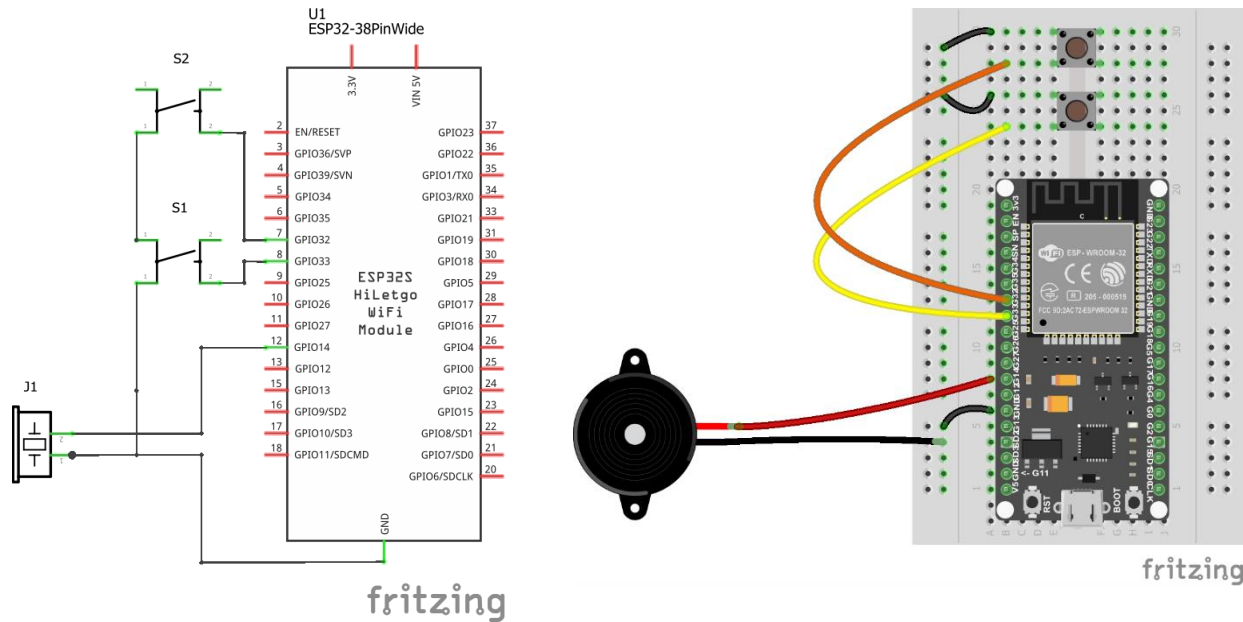
```
  }
  void loop(){
    if (digitalRead(Sw1) == LOW) {
      // If Sw1 pushed, play Di
      tone(Buzzer, NOTE_C6, 100, BUZZER_CHANNEL); // 100ms frequency 1047Hz
      Serial.println("di ");
      noTone(Buzzer, BUZZER_CHANNEL); // Silence...
      delay(100); // ...100ms

    }
    else {
      //Do nothing
    }
  }
```

```
  if (digitalRead(Sw2) == LOW) {
    // If Sw2 pushed, play Dah
    tone(Buzzer, NOTE_C6, 300, BUZZER_CHANNEL); // 100ms frequency 1047Hz
    Serial.println("dah ");
    noTone(Buzzer, BUZZER_CHANNEL); // Silence...
    delay(100); // ...100ms
  }
  else {
    //Do nothing
  }
}
```

Kysymys 2: Reverse Engineering, piirrä koodista tilakaavio. Ei ole helppo, ei ☹

Rakenne selkiytyy, kun käyttää funktio -kutsuja



Kysymys 3: Piirrä koodista

Tilakaavio!

```
/* Di_Dah_functionCall_ESP32.ino
 * 13th Sept. 2020 by Mr T
 */
#include <Tone32.h>
#define BUZZER_CHANNEL 0
```

```
int Buzzer = 14;
int Sw1 = 32;
int Sw2 = 33;
```

```
void setup() {
  pinMode(Buzzer, OUTPUT); // Buzzer to pin 10
  pinMode(Sw1, INPUT_PULLUP); // Switch1 to pin 5
  pinMode(Sw2, INPUT_PULLUP); // Switch2 to pin7
}
```

```
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
}
```

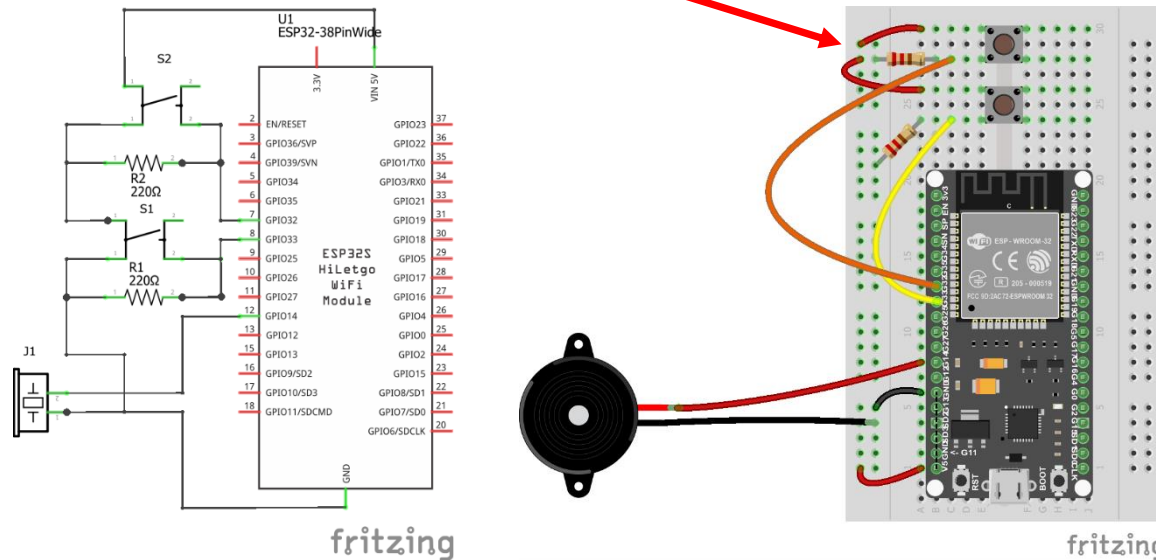
```
void loop() {
```

```
  if (digitalRead(Sw1) == LOW) {
    di();
  }
  else {
    //Do nothing
  }
}
```

```
  if (digitalRead(Sw2) == LOW) {
    dah();
  }
  else {
    //Do nothing
  }
}
```


Koodi rupeaa näyttämään tilakoneen koodilta, kun käytetään Switch – Case rakennetta

Huom! Kytkentä on muuttunut! Nyt siellä on PullDown vastukset!
Ja kytkimet menee +5Vdc



Kysymys 4: Ohjelma on edelleen pollaava (polling). Selitä pollaus!

```
/*Di_Dah_switchCase_ESP32.ino  
* 13th Sept. 2020 Mr T  
*/
```

```
const int sw1 = 32;  
const int sw2 = 33;  
const int LED = 27;
```

```
int val = 0;
```

```
void setup() {  
  
  pinMode(sw1, INPUT);  
  pinMode(sw2, INPUT);  
  pinMode(LED, OUTPUT);
```

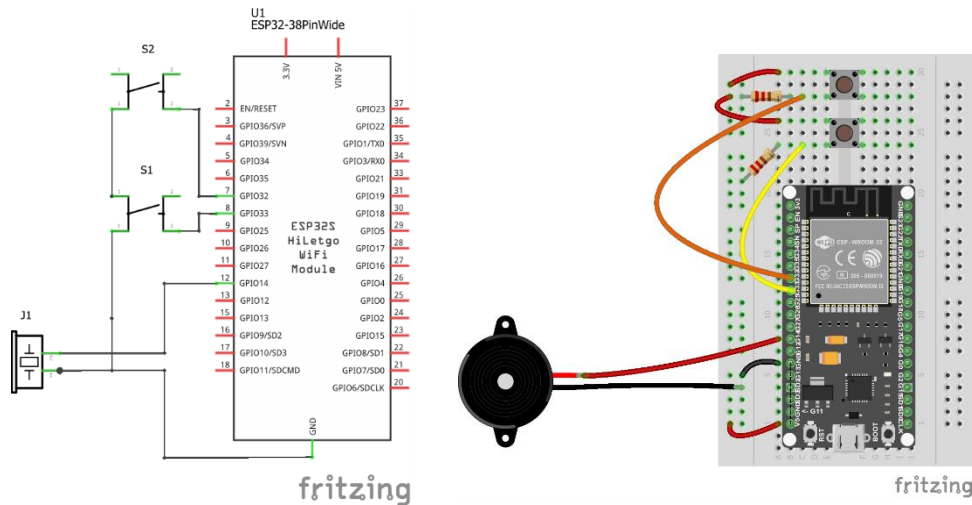
```
  Serial.begin(115200);  
}
```

```
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 :  
      //TEE TÄHÄN di()  
      Serial.println("di"); // tämän voi poistaa kun koodi toimii  
      delay(100); // tämän voi poistaa kun koodi toimii  
      break;
```

```
    case 33 :  
      //TEE TÄHÄN dah()  
      Serial.println("dah"); // tämän voi poistaa kun koodi toimii  
      delay(100); // tämän voi poistaa kun koodi toimii  
      break;  
  }  
}
```

Fiksuin tapa toteuttaa keskeytys on käyttää tapahtumapohjaista (Event Driven) koodia. En nyt ehtinyt koodata esimerkkiä, mutta tässä ns. Callback esimerkki, jonka tein.



```
/*Di_Dah_Callback_Driven_ESP32
 * 13th Sept. 2020 Mr T
 */
```

```
#include < pitches.h>
#include < Tone32.h>

#include <EasyButton.h>
#include <EasyButtonBase.h>
#include <EasyButtonTouch.h>
#include <EasyButtonVirtual.h>
#include <Sequence.h>
```

```
#define Buzzer 14
#define BUZZER_CHANNEL 0

#define sw1 32
#define sw2 33
```

```
int debounce = 35;
bool pullup = false;
bool invert = true;
```

```
EasyButton diButton(sw1,debounce,pullup,invert);
EasyButton dahButton(sw2,debounce,pullup,invert);
```

```
void setup() {
  Serial.begin(115200);

  diButton.begin();
  dahButton.begin();

  diButton.onPressed(di);
  dahButton.onPressed(dah);
}
```

```
void di(){
  Serial.println("    di");
  tone(Buzzer, NOTE_G6, 60, BUZZER_CHANNEL);
  noTone(Buzzer, BUZZER_CHANNEL);
}

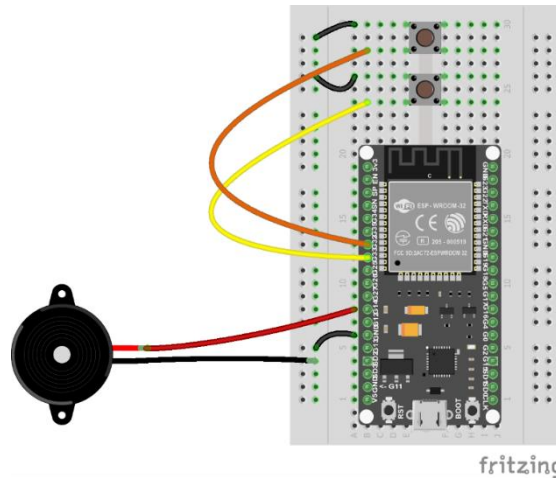
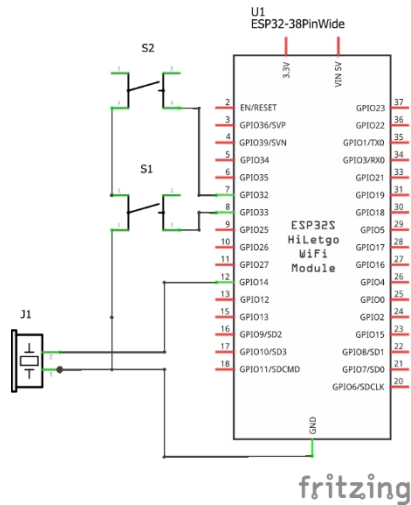
void dah(){
  Serial.println("    dah");
  tone(Buzzer, NOTE_G6, 200, BUZZER_CHANNEL);
  noTone(Buzzer, BUZZER_CHANNEL);
}
```

```
void loop() {
  diButton.read();
  dahButton.read();
  delay(1); //
}
```

Kysymys 5: Kokeile koodia. Mitä eroa toiminnallisuudessa huomaat esim. Switch-Case koodiin? Koodi ei toimi kunnolla osaako joku korjata? Se on 1extra piste!

Perusongelma eli pollaus voidaan kiertää käyttämällä keskeytystä (interrupt) Seuraava koodi tekee kovokeskeytyksen

Huom! Kytkentä on TAAS muuttunut! Alkuperäinen kytkentä! Tane on kyllä /c:stä! ☹️



Kysymys 6: Keskeytysrutiinin (ISR) sisällä ei voi käyttää delay(), tai millis(). Miksi?

Kysymys 7: Joskus koodi havaitsee useamman painalluksen. Syynä on Switch Debouncing. Selitä mikä se on

Tässä koodissa on alkeellinen debouncing toiminne!

```
/*  
 * InterruptDriven_ESP32.ino  
 * 13th Sept. 2020 by Mr T  
 */
```

```
#include < pitches.h> // Library for notes and  
frequencies  
#include < Tone32.h> // Library to generate tones  
#define Buzzer 14  
#define BUZZER_CHANNEL 0
```

```
int Sw1 = 32;  
int Sw2 = 33;  
int soi = 0;  
bool button_pressed = false;
```

```
long pressTime = 0;  
long debounceDelay = 35;
```

```
void IRAM_ATTR di(){  
  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(){  
  if (button_pressed){  
    if ((millis() - pressTime) > debounceDelay){  
      //Laita tähän soittamaan nuottia B5  
      button_pressed = false;  
    }  
    else {  
      button_pressed = true;  
    }  
  }  
  else {  
    //DoNothing  
  }  
}
```


Event - Listener

Toteuta event-pohjainen tilakone! Tässä on koodi hieman puutteellisena! Tutkikaa examplet tuosta *Eventually.h* kirjastosta!

Tee raportti, jossa käyt koodit läpi, kommentoit ne ja teeti niistä toimivat korjaamalla kohdat, jotka apitää korjata. Lisäksi vastaa kysymyksiin!

```
#include < pitches.h>
#include < Tone32.h>
#include < Eventually.h>

EvtManager mgr;
int beepTime = 0;
void setup() {
    // put your setup code here, to run once:

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

    mgr.laitaTähänKohtaanListener(new EvtPinListener(32, (EvtAction)swlpressed));
    mgr.laitaTähänKohtaanListener(new EvtPinListener(33, (EvtAction)sw2pressed));
}

USE_EVENTUALLY_LOOP(mgr)

bool swlpressed()
{
    mgr.resetContext();
    beepTime = 60;
    Serial.println("Di!");
    mgr.laitaTähänKohtaanListener(new EvtPinListener(32, (EvtAction)swlpressed));
    mgr.laitaTähänKohtaanListener(new EvtPinListener(33, (EvtAction)sw2pressed));
    //Soita nuottia C4
    noTone(14, 0);
    return true;
}

bool sw2pressed()
{
    mgr.resetContext();
    beepTime = 200;
    Serial.println("Dah!");
    mgr.laitaTähänKohtaanListener(new EvtPinListener(32, (EvtAction)swlpressed));
    mgr.laitaTähänKohtaanListener(new EvtPinListener(33, (EvtAction)sw2pressed));
    //Soita nuottia C4
    noTone(14, 0);
    return true;
}
```