

1)

```

const unsigned long t_measuring=30000;
const unsigned long t_debounce=50;
const unsigned long interval=10;
const int measures_len=(int) (t_measuring/interval); //3000;
const int unmeasured=-1;

const int light_sensor= A0;
const int push_button= 53;
const int measuring_led=13;
int state=0;
int i=0;
unsigned long t=0,t_aux=0;
int is_button_pressed=0;

int minimum=5000,maximum=-1;
long sum=0;
int n=0;

int measures[measures_len];
unsigned long deltat();
int verify_button();

void setup() {
  // put your setup code here, to run once:
  pinMode(push_button, INPUT);
  pinMode(measuring_led, OUTPUT);
  digitalWrite(measuring_led,LOW);
  Serial.begin(9600);
  for(int i=0;i<measures_len;i++){
    measures[i]=unmeasured;
  }
}

void loop() {
  // put your main code here, to run repeatedly:
  switch (state) {
    case 0:
      is_button_pressed=digitalRead(push_button);
      if(is_button_pressed==HIGH) {
        t=millis();
        state=1;
      }
      break;
    case 1:

```

```

is_button_pressed=digitalRead(push_button);
if(deltat()>=t_debounce && is_button_pressed==HIGH) {
    t=millis();
    state=2;
} else if(deltat()>=t_debounce && is_button_pressed==LOW) {
    state=0;
}
break;
case 2:
is_button_pressed=digitalRead(push_button);
if(is_button_pressed==LOW) {
    state=3;
    Serial.println("Start measures");
    i=0;
}
break;
case 3:
    digitalWrite(measuring_led,HIGH);
is_button_pressed=digitalRead(push_button);
if(deltat()>=(i+1)*10 && i<measures_len) {
    measures[i]=analogRead(light_sensor);
    i++;
}
if(deltat()>=t_measuring || is_button_pressed==HIGH) {
    state=4;
    Serial.println("Stop measures");
}
break;
case 4:
    digitalWrite(measuring_led, LOW);
    Serial.println("Printing results");
    for(int i=0;i<measures_len;i++) {
        if(measures[i]>0) {
            if(measures[i]>maximum) {
                maximum=measures[i];
            } else if(measures[i]<minimum) {
                minimum=measures[i];
            }
            sum+=measures[i];
            n++;
        } else {
            break;
        }
    }
    Serial.print("max: ");
    Serial.println(maximum);
    Serial.print("min: ");
    Serial.println(minimum);
    Serial.print("mean: ");

```

```

Serial.println((float) sum/ (float) n);
Serial.print("number of points: ");
Serial.println(n);

if(is_button_pressed==HIGH && deltat()>t_debounce) {
    state=5;
} else if(is_button_pressed==LOW) {
    state=6;
}
break;
case 5:
    is_button_pressed=digitalRead(push_button);
    if(is_button_pressed==LOW) {
        state=6;
    }
    break;
case 6:
    for(int i=0;i<measures_len;i++){
        measures[i]=unmeasured;
    }
    minimum=5000;
    maximum=unmeasured;
    sum=0;
    n=0;
    state=0;
    break;
default:
    Serial.println("reseting FSM");
    state=6;
}

```

```

unsigned long deltat() {
    return millis()-t;
}

```

6/6

2.1)

```

int buttonblau=4; // taster 1
int buttonrot=5; // taster 2
int buttongruen=6; // taster 3
int buttongelb=7; // taster 4

int ledblau=8;
int ledrot=9;
int ledgruen=10;
int ledgelb=11;

```

```

int lautsprecher=3;

unsigned long t=0;
int abspielenUser=750;
int abspielenArduino=500;
int breaktime=100;

int speicher[100] = {0};
int counter=0;
int countersequence=0;
int state = 0;
int led = 0;

int blaupushed=0;
int rotpushed=0;
int gruenpushed=0;
int gelbpushed=0;

unsigned long delta();
void initial();

void setup(){
    pinMode(buttonblau, INPUT);
    pinMode(buttonrot, INPUT);
    pinMode(buttongruen, INPUT);
    pinMode(buttongelb, INPUT);

    pinMode(ledgruen, OUTPUT);
    pinMode(ledgelb, OUTPUT);
    pinMode(ledblau, OUTPUT);
    pinMode(ledrot, OUTPUT);
    pinMode(lautsprecher, OUTPUT);

    Serial.begin(9600);
}

void loop(){
    switch(state){
        case 0:
            //Take input
            if(digitalRead(buttonblau)==HIGH){
                Serial.println("Blau gepushed!");
                speicher[counter] = 1;
                state = 1;
                led = 1;
            }else if (digitalRead(buttonrot)==HIGH){
                Serial.println("Rot gepushed!");
                speicher[counter] = 2;
                state = 1;
            }
        }
    }

```

```

    led = 2;
} else if (digitalRead(buttongruen) == HIGH) {
    Serial.println("Gruen gepushed!");
    speicher[counter] = 3;
    state = 1;
    led = 3;
} else if (digitalRead(buttongelb) == HIGH) {
    Serial.println("Rot gepushed!");
    speicher[counter] = 4;
    state = 1;
    led = 4;
}
break;
case 1:
    //Repeat input
    t = millis();
    switch(led) {
        case 1:
            digitalWrite(ledblau, HIGH);
            tone(lautsprecher, 220);
            break;
        case 2:
            digitalWrite(ledrot, HIGH);
            tone(lautsprecher, 440);
            break;
        case 3:
            digitalWrite(ledgruen, HIGH);
            tone(lautsprecher, 660);
            break;
        case 4:
            digitalWrite(ledgelb, HIGH);
            tone(lautsprecher, 880);
            break;
    }
    state = 2;
    break;
case 2:
    //Stop repeating input
    if(delta() >= abspielenUser) {
        switch(led) {
            case 1:
                digitalWrite(ledblau, LOW);
                break;
            case 2:
                digitalWrite(ledrot, LOW);
                break;
            case 3:
                digitalWrite(ledgruen, LOW);
                break;

```

← Wenn 2mal die selbe
 LED hintereinander blinken
 soll keine sichtbare
 LOW phase

(schöner aber nicht
 explizit gefordert)

```

        case 4:
            digitalWrite(ledgelb,LOW);
            break;
    }
    noTone(lautsprecher);
    state = 3;
    counter++;
    t=millis();
}
break;
case 3:
    //break
    if(delta() >= breaktime){
        state=4;
    }
    break;
case 4:
    //Start repeating sequence
    led = speicher[countersequence];
    countersequence++;
    t =millis();
    switch(led){
        case 1:
            digitalWrite(ledblau, HIGH);
            tone(lautsprecher,220);
            break;
        case 2:
            digitalWrite(ledrot,HIGH);
            tone(lautsprecher,440);
            break;
        case 3:
            digitalWrite(ledgruen,HIGH);
            tone(lautsprecher,660);
            break;
        case 4:
            digitalWrite(ledgelb,HIGH);
            tone(lautsprecher,880);
            break;
    }
    state = 5;
    break;
case 5:
    if(delta())>=abspielenArduino){
        switch(led){
            case 1:
                digitalWrite(ledblau, LOW);
                break;
            case 2:
                digitalWrite(ledrot,LOW);

```

```

        break;
    case 3:
        digitalWrite(ledgruen,LOW);
        break;
    case 4:
        digitalWrite(ledgelb,LOW);
        break;
    }
    noTone(lautsprecher);
    t=millis();
    state=6;
}
break;
case 6:
    //Break or end of sequence
    if(delta() >= breaktime){
        if(countersequence == counter){
            countersequence=0;
            state=0;
        }else{
            state=4;
        }
    }
    break;
}
}

```

```

unsigned long delta(){
    return millis()-t;
}

```

✓ 5/5

2.2)

So wie es hier geschrieben ist, kann man nur 100 Eingaben machen. Danach kann
Das Problem hierbei ist, dass wir das array mit fest anlegen müssen und so können

2.3) (bonus)

```

int buttonblau=4; // taster 1
int buttonrot=5; // taster 2
int buttongruen=6; // taster 3
int buttongelb=7; // taster 4

int ledblau=8;

```

↗
?
Scheint aber
richtig zu sein
1/1

```

int ledrot=9;
int ledgruen=10;
int ledgelb=11;

int lautsprecher=3;

unsigned long t=0;
int abspielenUser=750;
int abspielenArduino=500;

int speicher[25] = {0};
int speicherbegin=0;
int counter=0;
int countersequence=0;
int state = 0;

int blaupushed=0;
int rotpushed=0;
int gruenpushed=0;
int gelbpushed=0;

unsigned long delta();
void initial();

void setup(){
    pinMode(buttonblau, INPUT);
    pinMode(buttonrot, INPUT);
    pinMode(buttongruen, INPUT);
    pinMode(buttongelb, INPUT);

    pinMode(ledgruen, OUTPUT);
    pinMode(ledgelb, OUTPUT);
    pinMode(ledblau, OUTPUT);
    pinMode(ledrot, OUTPUT);
    pinMode(lautsprecher, OUTPUT);

    Serial.begin(9600);
}

void loop(){
    switch(state){
    case 0:
        t = millis();
        if (digitalRead(buttonblau)== HIGH)
        {
            Serial.println("Blau gepushed!");
            speicher[counter] = 1;
            t = millis();
            state = 1;

```



```

}else if (digitalRead(buttonrot) == HIGH)
{
    Serial.println("Rot gepushed!");
    speicher[counter] = 2;
    t = millis();
    state = 2;
}else if (digitalRead(buttongruen) == HIGH)
{
    Serial.println("Gruen gepushed!");
    speicher[counter] = 3;
    t = millis();
    state = 3;
}else if (digitalRead(buttongelb) == HIGH)
{
    Serial.println("Gelb gepushed!");
    speicher[counter] = 4;
    t = millis();
    state= 4;
}
break;
case 1:
    if (delta() < abspielenUser)
    {
        digitalWrite(ledblau, HIGH);
        tone(lautsprecher, 220);
    }else{
        digitalWrite(ledblau, LOW);
        noTone(lautsprecher);
        state = 0;
    }
break;
case 2:
    if (delta() < abspielenUser)
    {
        digitalWrite(ledrot, HIGH);
        tone(lautsprecher, 440);
    }else{
        digitalWrite(ledrot, LOW);
        noTone(lautsprecher);
        state = 0;
    }
break;
case 3:
    if (delta() < abspielenUser)
    {
        digitalWrite(ledgruen, HIGH);
        tone(lautsprecher, 660);
    }else{
        digitalWrite(ledgruen, LOW);

```

```

        noTone(lautsprecher);
        state = 0;
    }
    break;
    case 4:
        if (delta() < abspielenUser)
        {
            digitalWrite(ledgelb, HIGH);
            tone(lautsprecher, 880);
        } else {
            digitalWrite(ledgelb, LOW);
            noTone(lautsprecher);
            state = 0;
        }
        break;
    default:
        break;
}
counter++;
if (counter > 24)
{
    counter = 0;
    speicherbegin = (speicherbegin + 1) % 25;
}
while (countersequence < 25) {
    switch (speicher[(speicherbegin + countersequence) % 25]) {
        case 0:
            state = 0;
            countersequence = 25;
            break;
        case 1:
            if (delta() < abspielenArduino)
            {
                digitalWrite(ledblau, HIGH);
                tone(lautsprecher, 220);
            } else {
                digitalWrite(ledblau, LOW);
                noTone(lautsprecher);
            }
            break;
        case 2:
            if (delta() < abspielenArduino)
            {
                digitalWrite(ledrot, HIGH);
                tone(lautsprecher, 440);
            } else {
                digitalWrite(ledrot, LOW);
                noTone(lautsprecher);
            }
    }
}

```

```

        break;
    case 3:
        if (delta() < abspielenArduino)
        {
            digitalWrite(ledgruen, HIGH);
            tone(lautsprecher, 660);
        } else {
            digitalWrite(ledgruen, LOW);
            noTone(lautsprecher);
        }
        break;
    case 4:
        if (delta() < abspielenArduino)
        {
            digitalWrite(ledgelb, HIGH);
            tone(lautsprecher, 880);
        } else {
            digitalWrite(ledgelb, LOW);
            noTone(lautsprecher);
        }
        break;
    }
    countersequence++;
}
countersequence = 0;
state = 0;
}

unsigned long delta() {
    return millis() - t;
}

```

Programm reagiert nicht
 auf Tastendrücker, macht
 also gar nichts
 aber Idee richtig
 + 1 Bonuspunkt

13/12