1)

```
const int haupt_amp_rot=22;
const int haupt_amp_gel=24;
const int haupt_amp_grun=26;
const int t_ms=1000;
void setup() {
  // put your setup code here, to run once:
  pinMode(haupt_amp_grun, OUTPUT);
  pinMode(haupt_amp_gel, OUTPUT);
  pinMode(haupt_amp_rot, OUTPUT);
}

void loop() {
  // put your main code here, to run repeatedly:
  //1st state rot
  digitalWrite(haupt_amp_rot, HIGH);
  digitalWrite(haupt_amp_grun, LOW);
  digitalWrite(haupt_amp_gel, LOW);
  delay(t_ms);
  //2nd state rot-gelb
  digitalWrite(haupt_amp_rot, HIGH);
  digitalWrite(haupt_amp_grun, LOW);
  digitalWrite(haupt_amp_gel, HIGH);
  delay(t_ms);
  //3rd state grun
  digitalWrite(haupt_amp_rot, LOW);
  digitalWrite(haupt_amp_grun, HIGH);
  digitalWrite(haupt_amp_gel, LOW);
  delay(t_ms);
  //4th state gelb
  digitalWrite(haupt_amp_rot, LOW);
  digitalWrite(haupt_amp_grun, LOW);
  digitalWrite(haupt_amp_gel, HIGH);
  delay(t_ms);

}
```

√ 3/3

2)

```
const unsigned int haupt_amp_rot=22;
const unsigned int haupt_amp_gel=24;
const unsigned int haupt_amp_grun=26;

const unsigned int neben_amp_rot=28;
const unsigned int neben_amp_gel=30;
const unsigned int neben_amp_grun=32;
const unsigned int bewegung_sensor=40;

unsigned int state=1;
unsigned int bewegung=0;
unsigned long t=0;
```

```
void grun_h();
void gelb_h();
void rot_h();
void rot_gelb_h();

void grun_n();
void gelb_n();
void rot_n();
void rot_gelb_n();

unsigned long deltat();
void update_bewegung();


void setup() {
  // put your setup code here, to run once:
  pinMode(haupt_amp_grun, OUTPUT);
  pinMode(haupt_amp_gel, OUTPUT);
  pinMode(haupt_amp_rot, OUTPUT);

  pinMode(neben_amp_grun, OUTPUT);
  pinMode(neben_amp_gel, OUTPUT);
  pinMode(neben_amp_rot, OUTPUT);

  pinMode(bewegung_sensor, INPUT);
  Serial.begin(9600);
}

void loop() {
  // put your main code here, to run repeatedly:
  Serial.print("STATE: ");
  Serial.println(state);
  Serial.print("BEWEGUNG?: ");
  Serial.println(bewegung);
  Serial.print("delta t: ");
  Serial.println(deltat());
  Serial.println("\n");
  switch (state){
    case 1:
      grun_h();
      rot_n();
      update_bewegung();
      if (bewegung==1){
        state=2;
        bewegung=0;
        t=millis();
      }
      break;

    case 2:
      gelb_h();
```

```
  if (deltat()>=1000){
    state=3;
  }
  break;

case 3:
  rot_h();
  if(deltat()>=2000){
    state=4;
  }
  break;

case 4:
  rot_gelb_n();
  if(deltat()>=3000){
    state=5;
  }
  break;

case 5:
  grun_n();
  if(deltat()>=8000){
    state=6;
  }
  break;
case 6:
  gelb_n();
  if(deltat()>=9000){
    state=7;
  }
  break;

case 7:
  rot_n();
  update_bewegung();
  if (deltat()>=10000){
    state=8;
  }
  break;
case 8:
  rot_gelb_h();
  update_bewegung();
  if(deltat()>=11000){
    t=millis();
    state=9;
  }
  break;
case 9:
  grun_h();
  update_bewegung();
  if(deltat()>=20000){
    state=1;
```

```
      }
      break;
    default:
      state=1;
      Serial.println("RESETING FSM");
  }
}

unsigned long deltat(){
  return millis()-t;
}
void grun_h(){
  digitalWrite(haupt_amp_rot, LOW);
  digitalWrite(haupt_amp_gel, LOW);
  digitalWrite(haupt_amp_grun, HIGH);
}
void gelb_h(){
  digitalWrite(haupt_amp_rot, LOW);
  digitalWrite(haupt_amp_gel, HIGH);
  digitalWrite(haupt_amp_grun, LOW);
}
void rot_h(){
  digitalWrite(haupt_amp_rot, HIGH);
  digitalWrite(haupt_amp_gel, LOW);
  digitalWrite(haupt_amp_grun, LOW);
}
void rot_gelb_h(){
  digitalWrite(haupt_amp_rot, HIGH);
  digitalWrite(haupt_amp_gel, HIGH);
  digitalWrite(haupt_amp_grun, LOW);
}

void grun_n(){
  digitalWrite(neben_amp_rot, LOW);
  digitalWrite(neben_amp_gel, LOW);
  digitalWrite(neben_amp_grun, HIGH);
}
void gelb_n(){
  digitalWrite(neben_amp_rot, LOW);
  digitalWrite(neben_amp_gel, HIGH);
  digitalWrite(neben_amp_grun, LOW);
}
void rot_n(){
  digitalWrite(neben_amp_rot, HIGH);
  digitalWrite(neben_amp_gel, LOW);
  digitalWrite(neben_amp_grun, LOW);
}
void rot_gelb_n(){
  digitalWrite(neben_amp_rot, HIGH);
  digitalWrite(neben_amp_gel, HIGH);
  digitalWrite(neben_amp_grun, LOW);
}
```

```
void update_bewegung(){
  if(bewegung!=1){
    bewegung=digitalRead(bewegung_sensor);
  }

}
```

6/6

3)

```
const unsigned int haupt_amp_rot=22;
const unsigned int haupt_amp_gel=24;
const unsigned int haupt_amp_grun=26;

const unsigned int neben_amp_rot=28;
const unsigned int neben_amp_gel=30;
const unsigned int neben_amp_grun=32;

const unsigned int fussgaenger_amp_rot=34;
const unsigned int fussgaenger_amp_grun=36;

const unsigned int button=38;

const unsigned int bewegung_sensor=40;

unsigned int state=1;
unsigned int bewegung=0;
unsigned int ispressed=0;
unsigned long t=0;

void grun_h();
void gelb_h();
void rot_h();
void rot_gelb_h();

void grun_n();
void gelb_n();
void rot_n();
void rot_gelb_n();

void rot_f();
void grun_f();

unsigned long deltat();
void update_bewegung();
void update_isButtonPressed();


void setup() {
  // put your setup code here, to run once:
  pinMode(haupt_amp_grun, OUTPUT);
  pinMode(haupt_amp_gel, OUTPUT);
```

```
  pinMode(haupt_amp_rot, OUTPUT);

  pinMode(neben_amp_grun, OUTPUT);
  pinMode(neben_amp_gel, OUTPUT);
  pinMode(neben_amp_rot, OUTPUT);

  pinMode(fussgaenger_amp_grun, OUTPUT);
  pinMode(fussgaenger_amp_rot, OUTPUT);

  pinMode(button, INPUT);
  pinMode(bewegung_sensor, INPUT);
  Serial.begin(9600);
}

void loop() {
  // put your main code here, to run repeatedly:
  Serial.print("STATE: ");
  Serial.println(state);
  Serial.print("BEWEGUNG?: ");
  Serial.println(bewegung);
  Serial.print("delta t: ");
  Serial.println(deltat());
  Serial.println("\n");
  switch (state){
    case 1:
      grun_h();
      rot_n();
      rot_f();
      update_bewegung();
      update_isButtonPressed();
      if (bewegung==1 || ispressed==1){
        state=2;
        t=millis();
      }
      break;

    case 2:
      gelb_h();
      update_bewegung();
      update_isButtonPressed();
      if (deltat()>=1000){
        state=3;
      }
      break;

    case 3:
      rot_h();
      update_bewegung();
      update_isButtonPressed();
      if(deltat()>=2000){
        if (bewegung==1){
          bewegung=0;
```

```
      state=4;
     }else{
      ispressed=0;
      state=10;
     }
   }
   break;

 case 4:
   rot_gelb_n();
   update_isButtonPressed();
   if(deltat()>=3000){
    state=5;
   }
   break;

 case 5:
   grun_n();
   update_isButtonPressed();
   if(deltat()>=8000){
    state=6;
   }
   break;
 case 6:
   gelb_n();
   update_isButtonPressed();
   if(deltat()>=9000){
    state=7;
   }
   break;

 case 7:
   rot_n();
   update_bewegung();
   update_isButtonPressed();
   if (deltat()>=10000){
    state=8;
   }
   break;
 case 8:
   rot_gelb_h();
   update_bewegung();
   update_isButtonPressed();
   if(deltat()>=11000){
    t=millis();
    state=9;
   }
   break;
 case 9:
   grun_h();
   update_bewegung();
   update_isButtonPressed();
```

```
      if(deltat()>=20000){
        state=1;
      }
      break;
    case 10:
      grun_f();
      update_bewegung();
      if (deltat()>=7000){
      state=11;
      }
      break;
    case 11:
      rot_f();
      update_bewegung();
      update_isButtonPressed();
      if (deltat()>=8000){
        state=12;
      }
      break;
    case 12:
      rot_gelb_h();
      update_bewegung();
      update_isButtonPressed();
       if (deltat()>=9000){
        state=9;
        t=millis();
      }
      break;
    default:
      state=1;
      Serial.println("RESETING FSM");
  }
}

unsigned long deltat(){
  return millis()-t;
}
void grun_h(){
  digitalWrite(haupt_amp_rot, LOW);
  digitalWrite(haupt_amp_gel, LOW);
  digitalWrite(haupt_amp_grun, HIGH);
}
void gelb_h(){
  digitalWrite(haupt_amp_rot, LOW);
  digitalWrite(haupt_amp_gel, HIGH);
  digitalWrite(haupt_amp_grun, LOW);
}
void rot_h(){
  digitalWrite(haupt_amp_rot, HIGH);
  digitalWrite(haupt_amp_gel, LOW);
  digitalWrite(haupt_amp_grun, LOW);
}
```

```
void rot_gelb_h(){
  digitalWrite(haupt_amp_rot, HIGH);
  digitalWrite(haupt_amp_gel, HIGH);
  digitalWrite(haupt_amp_grun, LOW);
}

void grun_n(){
  digitalWrite(neben_amp_rot, LOW);
  digitalWrite(neben_amp_gel, LOW);
  digitalWrite(neben_amp_grun, HIGH);
}
void gelb_n(){
  digitalWrite(neben_amp_rot, LOW);
  digitalWrite(neben_amp_gel, HIGH);
  digitalWrite(neben_amp_grun, LOW);
}
void rot_n(){
  digitalWrite(neben_amp_rot, HIGH);
  digitalWrite(neben_amp_gel, LOW);
  digitalWrite(neben_amp_grun, LOW);
}
void rot_gelb_n(){
  digitalWrite(neben_amp_rot, HIGH);
  digitalWrite(neben_amp_gel, HIGH);
  digitalWrite(neben_amp_grun, LOW);
}
void rot_f(){
  digitalWrite(fussgaenger_amp_rot, HIGH);
  digitalWrite(fussgaenger_amp_grun, LOW);
}
void  grun_f(){
  digitalWrite(fussgaenger_amp_rot, LOW);
  digitalWrite(fussgaenger_amp_grun, HIGH);
}

void update_bewegung(){
  if(bewegung!=1){
    bewegung=digitalRead(bewegung_sensor);
  }
}
void update_isButtonPressed(){
  if(ispressed==0){
    ispressed=digitalRead(button);
  }
}
```

J 3/3

12/12