

```
int ledHaR = 22;
int ledHaGe = 24;
int ledHaGr = 26;
int ledNaR = 28;
int ledNaGe = 30;
int ledNaGr = 32;
int ledFuR = 34;
int ledFuGr = 36;
int sensorfuss = 38;
int sensor = 40;

int ms = 1000;
long lastswitch = 0;
long lastNa = 0;
long lastbewegung = 0;
bool bewegung = false;
bool fussPress = false;
int phase = 1;
```

```
void setup() {
    pinMode(ledHaGr, OUTPUT);
    pinMode(ledHaGe, OUTPUT);
    pinMode(ledHaR, OUTPUT);
    pinMode(ledNaGr, OUTPUT);
    pinMode(ledNaGe, OUTPUT);
    pinMode(ledNaR, OUTPUT);
    pinMode(ledFuR, OUTPUT);
    pinMode(ledFuGr, OUTPUT);
    pinMode(sensor, INPUT);
    pinMode(sensorfuss, INPUT);
    digitalWrite(ledHaGr, HIGH);
    digitalWrite(ledNaR, HIGH);
    digitalWrite(ledFuR, HIGH);
}
```

```
void loop(){
    if(digitalRead(sensor)==HIGH || bewegung)
        if(millis() - lastswitch >= (20 * ms))
            NebenAmpelTrigger();
        else
            bewegung = true;

    if(digitalRead(sensorfuss)==HIGH || fussPress)
        if(millis() - lastswitch >= (20 * ms))
            FussGaengerAmpelTrigger();
        else
            fussPress = true;
}
```

```
void FussGaengerAmpelTrigger(){
    bewegung = false;
    lastswitch = millis();
    HauptAmpelGelb();
    waitAndCheck(1);
}
```

```

    HauptAmpelRot();
    waitAndCheck(1);
    FussAmpelGruen();
    waitAndCheck(5);
    FussAmpelRot();
    // now it makes sense to press fuss again!
    fussPress = false;
    waitAndCheck(1);
    HauptAmpelGelbRot();
    waitAndCheck(1);
    HauptAmpelGruen();
    waitAndCheck(1);
}

void NebenAmpelTrigger(){
    fussPress = false;
    lastswitch = millis();
    HauptAmpelGelb();
    waitAndCheck(1);
    HauptAmpelRot();
    waitAndCheck(1);
    NebenAmpelGelbRot();
    waitAndCheck(1);
    NebenAmpelGruen();
    waitAndCheck(5);
    NebenAmpelGelb();
    // now it makes sense to trigger bewegung again
    bewegung = false;
    waitAndCheck(1);
    NebenAmpelRot();
    waitAndCheck(1);
    HauptAmpelGelbRot();
    waitAndCheck(1);
    HauptAmpelGruen();
    waitAndCheck(1);

}

void waitAndCheck(int x){
    lastswitch = millis();
    while(millis()-lastswitch < (x * ms)){
        if(digitalRead(sensor)==HIGH)
            bewegung=true;
        else if(digitalRead(sensorfuss)==HIGH)
            fussPress=true;
    }
}

void FussAmpelRot(){
    digitalWrite(ledFuGr, LOW);
    digitalWrite(ledFuR, HIGH);
}

void FussAmpelGruen(){

```

```
    digitalWrite(ledFuGr, HIGH);
    digitalWrite(ledFuR, LOW);
}

void NebenAmpelGruen(){
    digitalWrite(ledNaGr, HIGH);
    digitalWrite(ledNaGe, LOW);
    digitalWrite(ledNaR, LOW);
}

void NebenAmpelGelb(){
    digitalWrite(ledNaGr, LOW);
    digitalWrite(ledNaGe, HIGH);
    digitalWrite(ledNaR, LOW);
}

void NebenAmpelRot(){
    digitalWrite(ledNaGr, LOW);
    digitalWrite(ledNaGe, LOW);
    digitalWrite(ledNaR, HIGH);
}

void NebenAmpelGelbRot(){
    digitalWrite(ledNaGr, LOW);
    digitalWrite(ledNaGe, HIGH);
    digitalWrite(ledNaR, HIGH);
}

void HauptAmpelGruen(){
    digitalWrite(ledHaGr, HIGH);
    digitalWrite(ledHaGe, LOW);
    digitalWrite(ledHaR, LOW);
}

void HauptAmpelGelb(){
    digitalWrite(ledHaGr, LOW);
    digitalWrite(ledHaGe, HIGH);
    digitalWrite(ledHaR, LOW);
}

void HauptAmpelRot(){
    digitalWrite(ledHaGr, LOW);
    digitalWrite(ledHaGe, LOW);
    digitalWrite(ledHaR, HIGH);
}

void HauptAmpelGelbRot(){
    digitalWrite(ledHaGr, LOW);
    digitalWrite(ledHaGe, HIGH);
    digitalWrite(ledHaR, HIGH);
}
```