

Globals

```
#define A2_G 10
#define A2_R 9
#define A1_R 13
#define A1_Y 12
#define A1_G 11
#define LIGHT_SENSOR 19
#define NUM_WARN_BLINKS 4
```

```
char pins[5] = {A2_G, A2_R,
A1_R, A1_Y, A1_G};
```

```
const byte interruptPin = 2;
volatile bool pressed = false;
volatile bool night = false;
```

```
int sensorValue = 0;
```

```
int sensorMin = 1023;
```

```
int sensorMax = 0;
```

End

the sensor value

minimum sensor value

maximum sensor value

```
attachInterrupt(digitalPinToInterrupt(interruptPin), handleRequest_,
RISING);
digitalWrite(A1_G, HIGH);
digitalWrite(A1_Y, LOW);
digitalWrite(A1_R, LOW);
digitalWrite(A2_G, LOW);
digitalWrite(A2_R, HIGH);
```

calibrate during the first five seconds

```
calibrate_();
Serial.begin(9600);
```

End

Setup

```
void setup()
```

```
size_t i = 0;
```

```
i <= sizeof(pins) ?
```

```
pinMode(pins[i], OUTPUT);
```

```
i++
```

Loop

```
void loop()
```

```
sensorValue =
analogRead(LIGHT_SENSOR);
```

in case the sensor value is outside the range seen during calibration

```
sensorValue = constrain(sensorValue,
sensorMin, sensorMax);
```

apply the calibration to the sensor reading

```
sensorValue = map(sensorValue, sensorMin, sensorMax, 0, 255);
Serial.println(sensorValue);
```

Switch to Night mode

```
sensorValue < 100 && !night ?
```

```
digitalWrite(A1_G, LOW);
digitalWrite(A1_R, LOW);
digitalWrite(A2_G, LOW);
digitalWrite(A2_R, LOW);
night = true;
```

Switch to Day mode

```
sensorValue > 150 && night ?
```

```
digitalWrite(A1_G, HIGH);
digitalWrite(A1_R, LOW);
digitalWrite(A2_G, LOW);
digitalWrite(A2_R, HIGH);
night = false;
pressed = false;
```

Night mode

```
night ?
```

```
pressed ?
```

```
digitalWrite(A1_Y, HIGH);
delay(1000);
digitalWrite(A1_Y, LOW);
delay(1000);
```

```
schaltSequenz_0;
```

End

Schaltseq.

```
void schaltSequenz_()
```

Green Blinking phase

```
size_t i = 0;
```

```
i < NUM_WARN_BLINKS ?
```

```
digitalWrite(A1_G, LOW);
delay(1000);
digitalWrite(A1_G, HIGH);
delay(1000);
```

```
i++
```

```
digitalWrite(A1_G, LOW);
```

Yellow phase

```
digitalWrite(A1_Y, HIGH);
delay(1000);
```

GO phase

```
digitalWrite(A1_Y, LOW);
digitalWrite(A1_R, HIGH);
digitalWrite(A2_R, LOW);
digitalWrite(A2_G, HIGH);
delay(5000);
```

Green Blinking phase for walkers

```
size_t i = 0;
```

```
i < NUM_WARN_BLINKS ?
```

```
digitalWrite(A2_G, LOW);
delay(1000);
digitalWrite(A2_G, HIGH);
delay(1000);
```

```
i++
```

STOP for walkers and Transition Car

```
digitalWrite(A2_R, HIGH);
digitalWrite(A1_Y, HIGH);
delay(1000);
```

Car go space

```
digitalWrite(A1_Y, LOW);
digitalWrite(A1_R, LOW);
digitalWrite(A1_G, HIGH);
pressed = false;
```

End

intpr.

```
void handleRequest_()
```

```
!pressed ?
```

```
pressed = true;
```

End

Calibration

```
void calibrate_()
```

```
millis() < 5000 ?
```

```
sensorValue =
analogRead(LIGHT_SENSOR);
```

record the maximum sensor value

```
sensorValue > sensorMax ?
```

```
sensorMax = sensorValue;
```

record the minimum sensor value

```
sensorValue < sensorMin ?
```

```
sensorMin = sensorValue;
```

End