

---

## Abstand Tacho/Sensor

### **Benötigt Hardware:**

- \* MasterBrick
- \* Distanz Sensor
- \* Speaker

### **Benötigt Funktionen:**

- \* Bedingungen
- \* timePassed

### **Schritt 01:**

- \* Nutze die "timePassed" Funktion, um den Speaker nach 1000 Millisekunden piepen zu lassen
- \* *Hinweis: der Piepton sollte nicht länger als die erwarteten 1000 Millisekunden dauern*

### **Schritt 02:**

- \* Ersetze die 1000 Millisekunden durch den Distanzwert
- \* Teile die länge des Pieptons durch die hälfte des Distanzwertes

### **Zusatz 01 (Anzeige auf Display):**

- \* Schließe das Segmentdisplay an
- \* Gebe den distanzwert / 10 mit "cm" aus
- \* *Hinweis: CM wird nicht richtig dargestellt, da "M" auf dem Segmentdisplay nicht möglich ist*

### **Zusatz 02 (Anzeige auf Display):**

- \* Schließe das Segmentdisplay an
  - \* Gebe "PLAY" aus Bei > 250 mm
  - \* Gebe "CLOSE" aus Bei > 150 mm
  - \* Gebe "STOP" aus Bei > 50 mm
-

---

# Lärmampel

## Benötigt Hardware:

- \* MasterBrick
- \* 16IO
- \* Sound Intensity

## Benötigt Funktionen:

- \* Bedingungen
- \* timePassed
- \* Sleep

## Schritt 01:

- \* Nutze die "timePassed" Funktion, um ein angeschlossenes LEDs nach 1000 Millisekunden blinken zu lassen
- \* Hinweis: LEDs sind durchnummeriert einschalten mit 1 bis 16
- \* Hinweis: LEDs abschalten ist von -1 bis -16
- \* Hinweis: Nach einschalten einer LED sollte sleep() angewendet werden um kurz zu warten bis sie wieder aus gehen soll

## Schritt 02:

- \* Sage welche LEDs wann an und aus gehen - time passed kann angepasst werden
- \* LED 1 wenn Lautstärke > 250
- \* LED 2 wenn Lautstärke > 500
- \* LED 3 wenn Lautstärke > 1000

## Zusatz 01 (dynamische Berechnung):

- \* Definiere maximale Lautstärke als variable z.B. maxVolume
- \* Tausche die fest geschriebenen Werte aus Schritt 02 (250, 500, 100) gegen aus
  - \* LED 1 wenn Lautstärke > (maxVolume / 4)
  - \* LED 2 wenn Lautstärke > (maxVolume / 3)
  - \* LED 3 wenn Lautstärke > (maxVolume / 2)

## Zusatz 02 (Einstellen per Drehkopf):

- \* Schließe den Rotary Sensor an
- \* Setze maxVolume aus Zusatz 02 auf (wert von Rotary \* 100)

## Zusatz 03 (Anzeige auf Display):

- \* Schließe das Segmentdiplay an
  - \* Zeige Lautstärke / 10 mit "dB" oder Lautstärke % an
-

---

## Led Lautstärke anzeigen (Equalizer)

### Benötigt Hardware:

- \* MasterBrick
- \* 16IO
- \* Sound Intensity

### Benötigt Funktionen:

- \* Bedingungen
- \* Loops/Schleifen
- \* timePassed
- \* Sleep

### Schritt 01:

- \* Nutze die "timePassed" Funktion, um ein angeschlossenes LEDs nach 1000 Millisekunden blinken zu lassen
- \* Hinweis: LEDs sind durchnummeriert einschalten mit 1 bis 16
- \* Hinweis: LEDs abschalten ist von -1 bis -16
- \* Hinweis: Nach einschalten einer LED sollte sleep() angewendet werden um kurz zu warten bis sie wieder aus gehen soll

### Schritt 02:

- \* Setze timePassed auf 10 Millisekunden
- \* Berechne wieviele LEDs an geschaltet werden
  - \* `ledAnzahl = aktuelle Lautstärke / (LautstärkeMax / 16 maxLEDs)`
- \* Schreibe ein Loop, der die LEDs einschaltet
  - \* `for (int led = 1; led < ledAnzahl; led++)`
- \* Schreibe ein weiteren Loop, der die LEDs abschaltet
  - \* `for (int led = ledAnzahl; ledAnzahl < 16; led++)`

### Zusatz 01 (Anzeige auf Display):

- \* Schließe das Segmentdisplay an
- \* Zeige Lautstärke / 10 mit "dB" oder ledAnzahl oder Lautstärke % an

### Zusatz 02 (Anzeige auf Display):

- \* Schließe das Rotary Sensor an
- \* Verändere die Sensibilität z.B. `ledMax = 16 - RotaryWert`

---

## Polizei Sirene

### Benötigt Hardware:

- \* MasterBrick
- \* Dual Button
- \* Speaker

### Benötigt Funktionen:

- \* Loop

### Schritt 01:

- \* Schreibe am ende der „main“ (Start Funktion) eine neue „loop“ Funktion
  - \* **loop("Police LED", run -> { //yourCode }**
- \* Schreibe in der neuen „loop“ Funktion weiter
- \* Schalte ein LED am DualButton ein
- \* Nutze die „sleep“ Funktion, um das Programm vor der nächsten Aktion zu pausieren
- \* Schalte das LED am DualButton aus
- \* Schalte nun abwechselnd LEDs an und aus

### Schritt 02:

- \* Sende zum Speaker Piep Töne gleichzeitig zu den LEDs
  - \* Wechsle die Frequenz von den Piep Tönen je nach dem welche LED gerade an ist
-

---

## Mini Music Player

### **Benötigt Hardware:**

- \* MasterBrick
- \* Dual Button
- \* Rotary
- \* Computer

### **Benötigt Funktionen:**

- \* Bedingungen

### **Schritt 01:**

- \* Suche im Internet nach eine Ton oder Musik Datei. Bitte nur „wav“ Dateien keine „mp3“.

### **Schritt 02:**

- \* Schreibe eine Bedingung, dass die Musik abspielt, wenn ein DualButton gedrückt ist
  - \* Siehe virtuellen Sensor „LocalAudio“

### **Schritt 03:**

- \* Schreibe eine Bedingung, dass die Musik stoppt, wenn der zweite DualButton gedrückt ist

### **Schritt 04:**

- \* Schreibe eine Bedingung, die bei einem Rotary event die Lautstärke regelt.

### **Zusatz:**

- \* Nutze die „loop“ Funktion, um die DualButton LED blinken zu lassen während die Musik spielt
-