

Bau einer Zählstation mit Hilfe der senseBox

Seit mindesten 50000 Jahren zählen wir Menschen. Wir zählen unsere eignen Sachen, Menschen, Treppenstufen, den Verkehr und, und und. Wir erhoffen uns dadurch Informationen über uns und unsere Umwelt.

Fragen die wir uns stellen könnten zum Beispiel sein:

- Wie viele Menschen leben in unserer Stadt/Dorf?
- Wie viele Menschen sind auf einer Veranstaltung?
- Wie viele Autos fahren auf einer Straße
- Wie viele Autos fahren auf den Schulhof?
- Wie oft wird ein Vogelnest verlassen und wieder betreten?

Bei all diesen Fragen ist nicht nur der Wert selber interessant, sondern auch, wie er sich über die Zeit hinweg verändert. So kann man Rückschlüsse über das Verhalten des Objektes und Einflüsse auf es untersuchen.

Auf diesem Arbeitsblatt werden wir im besonderen beachten, wie man einen solchen Zähler mit Hilfe der sensebox bauen kann und wie man ihn so baut, dass man nicht selber jedes einzelne zu zählende Objekt aufschreiben muss.

- ① Suche dir eine Person mit der du zusammenarbeiten möchtest!
 - Schaut euch gemeinsam die nächsten Seiten an und überlegt dann, was genau ihr zählen möchtet.

Die Projektidee:

Das wollen wir mit Hilfe der senseBox messen:

○

Wir wollen gerade dieses Phänomen messen, da :

○

Los geht's!



Diese Teile braucht ihr:

- Ultraschall-Distanzsensor
- senseBox MCU
- senseBox JST-Adapterkabel
- Display

③ Sucht euch die oben genannten Komponenten heraus.

- Ihr nutzt den Ultraschall-Distanzsensor um mit Hilfe des Schalls die Entfernung von Objekten zu messen. Der Sensor sendet einen Impuls aus und misst die Zeit, bis er das Echo des Impulses wieder empfängt. Aus dieser Zeit wird mit Hilfe der Schallgeschwindigkeit die Entfernung des Objektes berechnet.



Verkabelung

Das rote Kabel und das schwarze Kabel sind für die Stromversorgung zuständig(5V)

Das grüne Kabel und das gelbe Kabel sind für die Datenübertragung zuständig.

④ Baut nun die Komponenten zusammen!

- Verbindet den Ultraschallsensor mit einem JST-Kabel mit der senseBox MCU.
- Sucht auf der senseBox MCU den Steckplatz Digital A und steckt das weiße Ende des Kabels hinein.
- Steckt nun den Ultraschallsensor vertikal in die Spalte e auf das weiße Steckbrett der senseBox MCU. Achtet hierbei darauf, dass die Beschriftung HC-SR04 nicht in Richtung der MCU zeigt.
- Verbindet nun die Ultraschallsensor mit dem Digital A Port indem ihr das rote Kabel in der Spalte d an die Stelle steckt, wo auf dem Ultraschallsensor Vcc steht, das grüne Kabel in die Spalte c steckt, wo beim Ultraschallsensor Trig steht, das gelbe Kabel in die Spalte c steckt, wo beim Ultraschallsensor echo steht und das schwarze Kabel in die Spalte d steckt, wo beim Ultraschallsensor Gnd steht.

Wenn der Anschluss erfolgt ist, werdet ihr nun überlegen, wie ihr die Daten visualisieren könnt.

- ⑤ Um die Daten zu visualisieren muss die **sensebox** programmiert werden. Dazu nutzt die Programmierumgebung blockly:
(<https://blockly.sensebox.de/ardublockly/?lang=de&board=sensebox-mcu>)
- Macht euch mit blockly vertraut. Welche Blöcke könnt ihr finden? Was machen diese Blöcke und welche Blöcke braucht ihr für euren Abstandsmesser?
 - Sucht als erstes alle Blöcke heraus, die ihr benötigt um das Display programmieren zu können.



Diese Bausteine braucht ihr

```
{ } Arduino Quellcode

#include <SPI.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#include <senseBoxIO.h>

#define OLED_RESET 4
Adafruit_SSD1306 display(OLED_RESET);

void setup() {
  senseBoxIO.powerI2C(true);
  delay(2000);
  display.begin(SSD1306_SWITCHCAPVCC, 128, 32);
  display.display();
  delay(100);
  display.clearDisplay();
}

void loop() {
  display.display();
  display.clearDisplay();
}
```

Bausteine für das Display



Diese Bausteine braucht ihr

The screenshot shows the Arduino IDE interface. On the left, the block-based code editor contains the following logic:

- Arduino run first:** A block to increase the 'Element' variable by 1.
- Arduino loop forever:** A loop containing:
 - A 'wenn mache' (if) block with a 'falsch' (false) condition, containing a 'Schreibe Element' (write) block.
 - An 'und' (and) block with a '$0 >= 0$' condition, containing a 'Schreibe Element' block.
 - A 'wenn mache' block with a 'wahr' (true) condition, containing an 'und' block with a '<math>0 <= 0</math>' condition, which then leads to a 'Schreibe Element' block.

On the right, the C++ source code view shows the following code:

```
{ } Arduino Quellcode
int Element;
void setup() {
}
void loop() {
  if (false) {
  }
  Element += 1;
  Element = (char)(0);
  0 >= 0;
  false;
  if (false) {
  }
  0 <= 0;
  Element = 0;
  true;
  (char)(0);
}
XML Blöcke
```

Bausteine für den Code

⑦ Passt nun euren Pseudocode an diese Bausteine an.

- Achtet darauf, dass ihr sinnvolle Variablennamen wählt
- Ist der Abstand des Sensors richtig eingestellt? (der Sensor misst in cm und seine Reichweite beträgt ca. 3m)

💡 So könnte euer Programm aussehen:

The screenshot shows the senseBox IDE interface. The main workspace contains a Scratch-style block-based program for an Arduino. The program is structured as follows:

- Arduino run first:**
 - Display initialisieren
- Arduino loop forever:**
 - wenn** (Ultraschall Abstandssensor an Port A \geq 40):
 - Trigger D1 Echo D2
 - mache** Schreibe Spurfrei **wahr** als Boolean
 - wenn** (Spurfrei $=$ wahr **und** Ultraschall Abstandssensor an Port A \leq 40):
 - Trigger D1 Echo D2
 - mache** erhöhe Fahrzeuge um 1
 - Schreibe Spurfrei **falsch** als Boolean
 - Zeige auf dem Display:
 - Schriftfarbe Weiß
 - Schriftgröße 1
 - x 0
 - y 0
 - Wert Fahrzeuge
 - Display löschen

The right sidebar shows the Arduino source code:

```
{ } Arduino Quellcode

#include <SPI.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#include <senseBoxIO.h>
#include "SenseBoxMCU.h"

boolean Spurfrei;
int Fahrzeuge;

#define OLED_RESET 4
Adafruit_SSD1306 display(OLED_RES

Ultrasonic UltrasonicA(1,2);

void setup() {
  senseBoxIO.powerI2C(true);
  delay(2000);
  display.begin(SSD1306_SWITCHCAPVCC
  display.display();
  delay(100);
  display.clearDisplay();
}

void loop() {
  if (UltrasonicA.getDistance() >
```

Beispielhafte Programmlösung

