

## Projektmappe Arduino Personenzähler

BBS Verden	LG 12.6	Name:
	Projekt Arduino	Klasse:
		Datum:

## Projekt Arduino

Als Aufgabe soll mit Hilfe des Arduinos ein Messwert aufgenommen, ausgewertet und angezeigt werden. Hierfür müssen Recherchen zu den Messmethoden und –möglichkeiten des Messwertes durchgeführt werden. Auch Alternativen zu den Standardverfahren sollen berücksichtigt werden. Die gemessenen Größen sollen durch den Arduino auf darstellbare Werte umgerechnet werden und dann mit Hilfe geeigneter einfacher Lösungen visualisiert werden.

Auch Arduino - Projekte aus anderen Bereichen können nach Absprache mit dem Lehrer durchgeführt werden. Hierbei muss auf den Umfang besonders geachtet werden!

Sämtliche Arbeitsschritte sind in einer Dokumentation schriftlich festzuhalten. Die Ergebnisse sind zum Abschluss des Projektes vorzustellen, wobei die Art der Vorstellung frei gewählt werden kann. Dieses Blatt dient auch als Bewertungsblatt und wird am Ende zu den Arbeiten geheftet.

Der Projektabschluss ist am **18.06.2021**.

Projekttitel: |

Bearbeiter: |

Bis zu diesem Termin sind folgende Zwischenergebnisse zu liefern:

	Aufgabe	Bemerkungen	Abgabe	Zeit	%	Punkte
1.	Zeitplanung	Erstellen einer Zeitplanung mit Hilfe des vorgegebenen Zeitstrahls	22.01.2021	<input type="checkbox"/>	4%	
2.	Gliederung	Inhaltsangabe, Aufbau der Dokumentation	19.02.2021	<input type="checkbox"/>	4%	
3.	Einleitung	Was will ich mit und bei diesem Projekt erreichen?	05.03.2021	<input type="checkbox"/>	4%	
4.	Hardwarekonzept	Schaltbild de Projektes mit Ein- und Ausgängen. Welche Bauteile werden benötigt?	26.03.2021	<input type="checkbox"/>	4%	
5.	Softwarekonzept	Strukturgramm des Programms, Bibliotheken, Variablen Unterfunktionen	24.04.2021	<input type="checkbox"/>	4%	
6.	Abgabe	Dokumentation, Hardware, Vorstellungsart	18.06.2021	<input type="checkbox"/>		
					Σ	20%

Die Projektnote setzt sich dann wie folgt zusammen:

	Ergebnisse	%	Punkte (max.)	erreicht	
1.	Zwischenergebnisse	20	100		
2.	Dokumentation	50	100		
3.	Vorstellung	30	100		
		Note:	Σ		

Die Note setzt sich aus den erzielten Ergebnissen laut obigen Eintragungen, der Eigenbewertung und der Bewertung durch den Lehrer zusammen. Eine Gewichtung der drei Einzelergebnisse wird nach Absprache gewichtet.

Arbeitsleistung laut Projekttagbuch (* xx%)	
Arbeitsleistung laut Bearbeiter (* xx%)	
Arbeitsleistung laut Lehrer (* xx%)	
Individualnote LG 12.6	

<b>Inhaltsverzeichnis</b>	
<b>Thema</b>	<b>Seitenzahl</b>
1. Umfeld	1
1.1. Projektbeschreibung und Zielsetzung	1
2. SOLL-Zustand	2
2.1. Beschreibung	2
3. Planung	2
3.1. Idee	2
3.2. Zeitstrahl	2
4. IST-Zustand	3-5
4.1. Beschreibung	3-4
4.2. Ablaufbeschreibung	4
4.2.1. Präposition	4
4.2.2. Reingehen	4
4.2.3. Rausgehen	5
4.2.4. Stehenbleiben	5
5. Kostenkalkulation	6
5.1. Präposition	6
5.2. Budget	6
5.3. Tabellarische Rechnung	6
6. Implementierung	7-8
6.1. Programm	7
6.2. Eingabe	7
6.3. Zeitstempel	7
6.4. Verarbeitung	8
6.5. Speichern	8
6.6. Ausgabe	8
7. SOLL/IST Vergleich	9
8. Nutzungsanalyse	10-16
8.1. Beschreibung	10
8.2. Maximum erreicht	10-13
8.2.1. Beschreibung	10
8.2.2. Erwartung	10-11
8.2.3. Auswertung der Messergebnisse	11-13
8.2.4. Interpretation der Ergebnisse	13
8.3. Minimum erreicht	14-15
8.3.1. Beschreibung	14
8.3.2. Erwartung	14
8.3.3. Auswertung der Messergebnisse	14-15
8.3.4. Interpretation der Ergebnisse	15
8.4. Durchgang blockiert	16
8.4.1. Beschreibung	16
8.4.2. Erwartung	16
8.4.3. Auswertung der Messergebnisse	16
8.4.4. Interpretation der Ergebnisse sowie Interpretation	16
8.5. SD Fehlerhaft	16
8.5.1. Beschreibung	16
8.5.2. Erwartung	16
8.5.3. Auswertung der Ergebnisse sowie Interpretation	16
9. Fazit	17-18
9.1. Reflektion	17

9.2. Probleme	17-18
9.3. Geplante Features	18
<b>Anlagen</b>	
<b>I. Projekttagbuch</b>	<b>I-II</b>
I.I. Organisatorisches	I
I.II. Recherche	I
I.III. Tätigkeit	II
I.IV. Dokumentation	II
<b>II. Glossar</b>	<b>III</b>
<b>III. Verwendete Werkzeuge</b>	<b>IV</b>
III.I. Beschreibung	IV
III.II. Software	IV
III.III. Hardware	IV
<b>IV. Konzept</b>	<b>V-X</b>
IV.I. Bibliotheken	V
IV.II. Variablen	V
IV.III. Unterfunktionen	VI
IV.III.I. Beschreibung	VI
IV.III.II. Speichern	VI
IV.III.III. Zeitstempel erhöhen	VI
IV.IV. Struktogramm	VII-VIII
IV.V. Pinbelegung	IX
IV.VI. Bauteilliste	X
IV.VII. Skizze	X
<b>V. Kundendokumentation</b>	<b>XI-XLVI</b>
V.I. Bedienungsanleitung	XI-XIV
V.II. Rechnungen der verbauten Komponenten	XV-XVI
V.II.I. Arduino	XV
V.II.II. Infrarotsensoren	XV
V.II.III. Netzteil und Mikro SD-Karte	XV
V.II.IV. SD Shield	XVI
V.II.V. LCD-Display	XVI
V.II.VI. Jumperkabel	XVI
V.III. Datenblätter	XVII-XLVI
V.III.I. Arduino Uno Rev 3	XVII
V.III.II. Infrarotsensor	XVIII-XXXII
V.III.III. Micro SD Shield	XXXIII-XXXVII
V.III.IV. LCD-Display	XXXVIII-XLVI
<b>VI. Quelltextdokumentation</b>	<b>XLVII-LI</b>
VI.I. Programmcode	XLVII-XLIX
VI.II. Quellen	L-LI

## **1. Umfeld**

### **1.1. Projektbeschreibung und Zielsetzung**

Der Technikunterricht an der BBS Verden<sup>1</sup> ist in verschiedene Lerngebiete aufgeteilt.

Einer dieser Lerngebiete ist Lerngebiet 12.3, welcher u.A. zur Realisierung Auftragsbezogener Planungen für ein technisches System mitsamt der Präsentation der daraus resultierenden Ergebnisse dient.

Hierfür arbeiten die Schüler an mit dem Lehrer Herr Düren, welcher gleichzeitig als Ansprechpartner zur Verfügung steht vereinbarten Projekte. Die resultierende Projektdokumentation sowie das Aufnehmen, Auswerten und Anzeigen der Messwerte dienen als Leistungsnachweis von Punkt IT 2 des Lerngebietes 12.3. Als Projekt habe ich, Rojat Kadah, mit Herr Düren einen Arduino basierten Personenzähler als Projekt vereinbart.

Im Verlauf der Dokumentation wird der Personenzähler als Projekt, Arduino und Personenzähler bezeichnet.

Bei diesem werden die vorbeigehenden Personen anhand der geplanten Eingabemöglichkeiten als Messwerte aufgenommen. Anschließend werden diese vom Arduino verarbeitet. Zu guter Letzt werden die Ergebnisse Visuell ausgegeben und als Datensatz abgespeichert.

Aufgrund einer nicht genau definierten Vorgehensweise steht es mir frei, auszuwählen, wie ich vorgehe. Die Ergebnisse sind spätestens am 18.06.2021 zu präsentieren. Zu den nicht Unterrichtsbezogenen Zielen gehören vor allem die Vertiefung im Umgang mit Projekten, das verbessern bzw. vertiefen der Hard- sowie Softwarekenntnisse sowie der Umgang mit Kompromissen und einem Begrenzten Budget.

## 2. SOLL-Zustand

### 2.1. Beschreibung

Geplant ist ein Arduino basierter Personenzähler<sup>2</sup>. Jedes Mal, wenn jemand vorbeigeht, wird ein Datensatz<sup>3</sup> erstellt, welcher zusätzlich zur Personenzählung einen Zeitstempel abspeichert, welcher für weitere Zwecke genutzt werden kann. Anhand Visueller Ausgabemethoden wird der Nutzer auf die Zählung hingewiesen.

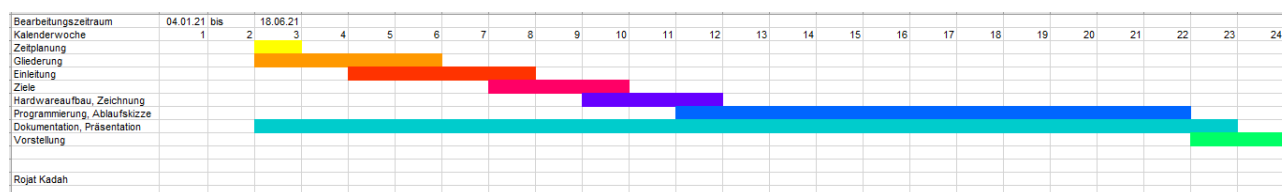
## 3. Planung

### 3.1. Idee

Die Messwerte sollen mithilfe einer Sensorik<sup>4</sup> aufgenommen werden. Knöpfe haben den Nachteil, dass sie berührt werden müssen, was vor allem in Zeiten von Corona<sup>5</sup> eine Zumutung ist. Ein Sensor kann nicht in Frage kommen, da dieser nicht unterscheiden kann, ob jemand rein- oder rausgeht. Zwei Sensoren bieten bereits die Möglichkeit, zwischen rein- und rausgehenden Personen zu unterscheiden, doch können viele Fälle, wie z.B. das umentscheiden der Gehrichtung beim Vorbeigehen oder mehrere vorbeigehende Personen, nicht berücksichtigen. Die größte Beschränkung besteht darin, dass der Arduino nur alle x Sekunden eine vorbeigehende Person zählen kann. Ein Großteil dieser Probleme kann durch Hinzufügung eines dritten Sensors dazwischen, einem Bestätiger, ausgemerzt werden. Der Mittlere Sensor wird im Verlauf der Dokumentation unter den Begriffen Bestätiger sowie Kennzeichnungen, dass er zwischen den anderen Sensoren ist, genannt. Bei dieser Vorgehensweise besteht der Vorteil, dass alles in Echtzeit ablaufen kann und es keine Verzögerungen gibt, da nur gezählt wird, sobald das Vorbeigehen an einer der äußeren Sensoren durch den mittleren Sensor bestätigt wird. Sobald der Bestätiger ausgelöst wird, erhält der Arduino einen Datensatz inklusive des Zeitstempels zur weiteren Verarbeitung. Zur Verarbeitung gehören zum einen die Speicherung der Daten auf einem Speichermedium und zum anderen die Visuelle Ausgabe der Daten für den Vorbeigehenden.

### 3.2. Zeitstrahl

Grafische Darstellung der benötigten Zeit.



Siehe „Dateien/Zeitplan/Zeitplanung.xlsx“ auf Folie 1.

## **4. IST-Zustand**

### **4.1. Beschreibung**

Basierend auf der Planung, Siehe hierfür Kapitel 3, sowie des Sollzustandes, Siehe hierfür Kapitel 2, ist der Personenzähler in einem Zustand, welcher der vereinbarten Funktionalität entspricht.

Für die Speicherung der Datensätze fiel die Entscheidung auf ein Micro SD Modul. Für die Ausgabe der Daten wird ein 20x4 Zeichen I2C LCD-Display genutzt.

Die im 3. Kapitel unter „3.1 Planung“ genannte drei Sensoren Taktik wird beim Personenzähler übernommen. Dank dieser kann der Personenzähler je nach Konfiguration in Echtzeit arbeiten und zwischen der Bewegungsrichtung unterscheiden. Jegliches vorbeigehen wird erst registriert, sobald der Bestätiger aktiviert wird. Die Vorteile sind in Kapitel 3, Planung, dargestellt. Zeitbedingt ist des Weiteren hinzugefügt worden, dass der Personenzähler erkennt, wenn jemand den Durchgang blockiert. Dies erfolgt durch ununterbrochenes Aktivieren des Bestätigers.

Sobald der Bestätiger aktiviert ist, sei es nach vorbeigehen an einem der äußeren Sensoren oder durch Blockierung, erstellt der Arduino auf der angeschlossenen SD-Karte einen Datensatz. Dieser enthält:

- Raumnummer: Gibt wieder, wo der Personenzähler aufgebaut ist.
- Zeitstempel: Standardmäßig 20 Millisekunden genauer Zeitstempel. Gibt an, wann der Datensatz nach Start eingetragen worden ist. Die Angabe erfolgt In Sekunden.
- Maximale Anzahl an Personen: Maximal erlaubte Anzahl an Personen im Raum.
- Aktuelle Anzahl an Personen: Aktuell im Raum befindliche Anzahl an Personen.
- Anzahländerung: Gibt wieder, ob jemand rein bzw. rausgegangen ist oder ob der Bestätiger blockiert wird.
- Gesamtzahl: Gibt an, wie viele Personen insgesamt vorbeigegangen sind.
- Gesamt raus: Insgesamt rausgegangene Anzahl an Personen.
- Gesamt rein: Insgesamt reingegangene Anzahl an Personen.
- Status: Gibt wieder, ob der Raum voll ist, nicht voll ist oder ob der Bestätiger blockiert oder wieder frei ist.

Die Daten werden im CSV Format abgespeichert, doch da der Arduino dieses nicht kennt, werden die Daten als Textdatei abgespeichert, sodass die Dateiendung nur noch abgeändert werden muss. Der Grund hierfür besteht darin, dass Excel das CSV kennt, sodass über Excel die Möglichkeit zur Speicherung im XLSX Format besteht um Diagramme auszugeben.

Sollte die SD nicht erkannt werden oder fehlen, so gibt der Arduino dies beim Start über den LCD-Display wieder.

Anschließend erfolgt die Ausgabe der Personenzahl über das vierzeilige I2C Display.

In der ersten Zeile wird die Raumnummer ausgegeben.

Bei der zweiten Zeile werden die aktuell im Raum befindliche sowie die maximal erlaubte Anzahl an Personen im Raum wiedergegeben.

Der Übersicht halber wird die dritte Zeile lediglich gefüllt, sobald der Bestätiger blockiert wird. Diese weist den Nutzer darauf hin, dass er den Sensor blockiert. Die letzte Zeile, Zeile vier, gibt wieder, ob der Raum voll oder nicht voll ist. „Nicht voll“ wird auch bei keiner Person im Raum ausgegeben, da der Nutzer lediglich darauf hingewiesen wird, dass der Raum nicht voll ist. Die Ausgabe erfolgt Dynamisch je nach Anzahl an Personen gegenüber der Maximal erlaubten Anzahl an Personen im Raum. Die drei Möglichen Programmabläufe sind unter Kapitel „4.2 Ablaufbeschreibung“ abrufbar

## **4.2. Ablaufbeschreibung**

### **4.2.1. Präposition<sub>6</sub>**

Zur Vermeidung komplexer Erklärungen wird bei den folgenden Programmabläufen davon ausgegangen, dass die Sensoren Horizontal in Reihe an den Türrahmen eines Raumes befestigt worden sind.

### **4.2.2. Reingehen**

Wenn jemand in den Raum geht, wird erst der Außensensor aktiviert. Solange die Person ab jetzt nicht am Bestätiger vorbeigeht, wird die Anzahl an Personen im Raum nicht erhöht.

Wenn die Person nun aber am mittleren Sensor vorbeigeht, so wird die Zählung bestätigt, ein Datensatz auf der SD-Karte erstellt und die Anzahl an im Raum befindlichen Personen auf dem LCD-Display aktualisiert. Des Weiteren wird, solange die Anzahl an Personen im Raum kleiner als die maximal erlaubte Anzahl ist, in der vierten Zeile der Status „Nicht Voll“ ausgegeben. Ansonsten wird der Status auf Voll gesetzt.

Wenn die Person nun weiter in den Raum und damit am Innensensor vorbeiläuft, so wird die Anzahl an Personen im Raum nicht um eins verringert, da hierfür eine Betätigung des Bestätigers notwendig ist.



#### **4.2.3. Rausgehen**

Wenn jemand aus dem Raum geht, wird erst der Innensensor aktiviert. Solange die Person ab jetzt nicht am Bestätiger vorbeigeht, wird die Anzahl an Personen im Raum nicht verringert.

Wenn die Person nun am mittleren Sensor vorbeigeht, so wird die Zählung bestätigt, ein Datensatz auf der SD-Karte erstellt und die Anzahl an im Raum befindlichen Personen auf dem LCD-Display aktualisiert. Aufgrund der Programmierung ist es nicht möglich, dass die Anzahl an Personen im Raum kleiner als Null ist. Entsprechend wird standardmäßig keine Personenzahl kleiner als Null auf dem LCD-Display ausgegeben. Wenn die Person nun weiter aus dem Raum und damit am Außensensor vorbeiläuft, so wird die Anzahl an Personen im Raum nicht um eins erhöht, da hierfür eine Bestätigung des Bestätigers notwendig ist.

#### **4.2.4. Stehenbleiben**

Sollte eine Person den Durchgang blockieren, so wird dies entsprechend solange auf dem LCD-Display ausgegeben wie die Blockierung dauert. Des Weiteren wird dies als entsprechender Datensatz abgespeichert.

Ein weiterer, entsprechender Datensatz wird erstellt, sobald der Durchgang wieder frei ist.

## **5. Kostenkalkulation**

### **5.1. Präposition**

Aufgrund mangelnder Vorkenntnisse mit dem Arduino sowie Fein- und Grobmechanischer, gesundheitlicher Probleme wurde jede Komponente mindestens zweimal erworben, um selbst für den Fall, dass der Personenzähler kaputt geht, noch ein Backup zu haben.

### **5.2. Budget**

Wegen mangelnder Wirtschaftlicher Mittel besteht nicht die Möglichkeit, die besten verfügbaren Komponenten zu erwerben. Vor allem auch deshalb, dass aufgrund Möglicher Hardwareausfälle noch ein ausreichendes Budget für den Erwerb von Ersatzkomponenten erhalten bleiben muss. Entsprechend ist das Budget auf 300€ beschränkt.

### **5.3. Tabellarische Rechnung**

Tabellarische Aufstellung der Kosten pro Komponente sowie des Gesamtpreises

<b>Komponente</b>	<b>Anzahl</b>	<b>Einzelpreis</b>	<b>Gesamtpreis</b>
Arduino Uno Revision 3	2	22,47€	22,47€+80€
Jumperkabel 20cm	2 Packungen a 40 Kabel	7,99€	15,98€
KY-004 Taster	1 Packung a 3 Taster	5,99€	5,99€
20x4 Zeichen 2004 LCD-Display	2	9,99€	19,98€
Lesezeichen Sticker	1 Packung a 840 Sticker	6,99€	6,99€
Lötkolben 60W	1	18,99€	18,99€
Netzteil 9V 2A	1	11,49€	11,49€
32GB Micro SD	1	4,79€	4,79€
Micro SD Shield	1	5,29€	5,29€
TRCT 5000	2 Packungen a 3 Sensoren	4,99€	9,98€
<b>Gesamt</b>		<b>98,98€</b>	<b>277,96 €</b>

Siehe Digitalen Anhang unter „Dateien/Rechnungen/Rechnung.xlsx“

## **6. Implementierung**

### **6.1. Programm**

Das Programm besteht aus mehreren Teilen, welche allesamt zum Großteil im Setup sowie Loop Teil des Programms implementiert worden sind.

Global werden vor Beginn der Funktionen die Organisatorischen Variablen wie die Raumnummer und der Zeitstempel, Bibliotheken zur Verwendung der verbauten Komponenten und zu guter Letzt die Sensoren deklariert.

Im Setupteil werden die Sensoren sowie der LCD konfiguriert. Abschließend wird der Fall, dass keine SD-Karte eingefügt worden ist, ins Setup integriert.

Sollte keine SD-Karte eingelegt sein, so gibt der Arduino dies beim Start wieder und stoppt die Arbeit durch eine leere Dauerschleife.

Im Setupteil wird erst die LCD-Ausgabe durchgeführt und anschließend die Sensorabfragen. Siehe hierfür „6.2. Eingabe“.

Anschließend wird die Funktion zur Erhöhung des Zeitstempels aufgerufen. Siehe hierfür „6.3. Zeitstempel“.

Diese erhöht den Zeitstempel um die jeweils beim delay<sup>9</sup> vergangene Anzahl an Millisekunden. Siehe hierfür „6.3. Zeitstempel“. Im Verlauf dieses Kapitels wird davon ausgegangen, dass der Personenzähler Horizontal im Türrahmen aufgebaut worden ist.

Siehe Kapitel „V.I. Bibliotheken“ für die jeweils genutzten Bibliotheken.

### **6.2. Eingabe**

Die Eingabe erfolgt innerhalb des Loop Teils. Dabei wird erstmal abgefragt, ob der Bestätiger aktiviert worden ist. Wenn, dann wird über das Display ausgegeben, dass der Durchgang frei gemacht werden soll.

Folgend findet dann die Betätigung der äußeren Sensoren statt. Wenn jemand in den Raum geht wird erstmal der Außensensor betätigt. Wenn nun der Bestätiger betätigt wird, so wird die Personenzahl um eins erhöht. Wenn stattdessen der Innensensor aktiviert wird, dann wird stattdessen der Rausgehprozess in Gang gesetzt, welcher genauso funktioniert, außer dass das anschließend folgende betätigen des Bestätigers zur Reduzierung der Personenzahl um eins führt. Des Weiteren führt die Betätigung des Bestätigers zum Aufruf der Speicherfunktion. Siehe Hierfür „6.5. Speichern“

### **6.3. Zeitstempel**

Der Zeitstempel wird genutzt, da es Budgetbedingt nicht möglich ist, eine RTC<sup>10</sup>, Real Time Clock, zu verwenden. Die Erhöhung von diesem findet in einer Funktion vom Typ Void statt und wird vor jedem Delay aufgerufen. In der Funktion haben wir zwei Werte. Einmal den Zeitstempel und den Wert, welche die Anzahl an Millisekunden pro Delay angibt. Folgend als Zeitsprung bezeichnet. Zeitstempel wird bei jedem Aufruf um Zeitsprung erhöht. Da Zeitsprung aber in Millisekunden ist, wird er dabei jedes Mal durch 1000 geteilt.

#### 6.4. Verarbeitung

Bei jeder Eingabe wird immer der Bestätiger betätigt. Mit Aufruf von diesem wird immer ein Datensatz auf dem Arduino erstellt, bestehend aus dem Zeitstempel in Sekunden, der maximalen sowie aktuellen Anzahl an Personen, der Anzahländerung, der Gesamten Personenzahl sowie der Gesamten rein- und rausgehenden Anzahl an Personen sowie dem Status. Der Status gibt an, ob der Durchgang blockiert ist, wieder frei ist oder ob der Raum voll bzw. nicht voll ist. Die Anzahländerung gibt je nachdem, ob jemand rein- oder rausgeht oder den Durchgang blockiert, +, – oder 0 wieder.

#### 6.5. Speichern

Bei der Speicherfunktion werden der Zeitstempel, die maximale sowie aktuelle Anzahl an Personen im Raum, die Anzahländerung sowie die Gesamte Anzahl vorbeigehender Personen sowie rein und rausgehend gespeichert. Zuvor wird mithilfe von SD.open der Zugriff auf die Datei gewährt, deren Dateiname am Anfang des Programms vom Nutzer festgelegt werden kann. Am Ende wird dieser Zugriff wieder beendet, sodass die SD-Karte während der Personenzählung sicher entfernt und wieder eingefügt werden kann. Gespeichert wird die Datei mit der Endung einer Textdatei. Das verwendete Format ist das CSV<sup>11</sup> Format, was an der Semikolontrennung zwischen jeder Variablen und dem Zeilenumbruch am Ende jeder Zeile erkennbar ist. Die Wahl fiel auf diese Lösung, da der Arduino standardmäßig keine Möglichkeit zur Speicherung im XLSX<sup>12</sup> Format hat. Dennoch besteht bei Excel die Möglichkeit, CSV Dateien im XLSX Format abzuspeichern.

Diese Vorgehensweise wurde aufgrund ihrer Einfachheit gegenüber anderen Möglichkeiten gewählt und weil das Erneute aufrufen der CSV Datei zur erneuten Erstellung der Diagramme führt.

#### 6.6. Ausgabe

Zur Ausgabe dient ein I2C<sup>13</sup> Display. Die Wahl ist auf dieses gefallen, da es gegenüber herkömmlichen Displays nur vier Anschlusspins besitzt und so mehr Pins für andere Zwecke zur Verfügung stehen.

Über das Display findet die Gesamte Ausgabe statt. Hierzu zählen die Ausgabe der SD Fehlermeldung am Anfang und die Datenausgabe im Verlauf des Loop Teils.

Die SD Fehlermeldung gibt aus, dass die SD entweder falsch formatiert ist oder dass keine angeschlossen ist. Eine Unterscheidung zwischen diesen beiden Fällen kennt der Personenzähler in seiner aktuellen Version nicht.

Teil der Ausgabe im Loop Teil sind die Ausgabe der Raumnummer in der ersten Zeile, die Angabe der aktuellen sowie maximalen Anzahl an Personen im Raum in der zweiten Zeile, bei Blockierung des Durchgangs die zugehörige Meldung in der dritten Zeile und die Ausgabe des Status in der vierten Zeile.

## **7. SOLL/IST Vergleich**

Im Sollzustand geht es vor allem darum, wie das Programm im groben aussehen soll. Als erster Punkt ist das Erstellen eines Datensatzes genannt worden, sobald jemand vorbeigeht. Dies ist im Programm entsprechend umgesetzt worden und wurde insofern erweitert, dass der Arduino zwischen der Bewegungsrichtung unterscheiden kann. Der zweite Punkt des Soll-Zustandes ist der Zeitstempel. Die Umsetzung ist ohne eine RTC erfolgt und wird mithilfe eines auf drei Kommastellen genau eingetragenen Floats gezählt.

Punkt drei ist, dass der Zeitstempel für weitere Zwecke genutzt werden soll. Beim Projekt wird er aktuell lediglich mitgespeichert um zu erfahren, wann jemand vorbeigegangen ist bzw. wann der Durchgang blockiert worden ist. Dies kann sich mit zukünftigen Versionen ändern.

Der Letzte Punkt im Soll-Zustand ist die Visuelle Ausgabe, mit welcher der Nutzer auf die Zählung hingewiesen wird.

Hierfür wird ein I2C Display verwendet, welches dank seiner vier Zeilen mit jeweils zwanzig Zeichen ausreichend Platz hat, um die aktuelle sowie maximal erlaubte Anzahl an Personen wiederzugeben.

Des Weiteren wird das Display genutzt, um die Raumnummer sowie den Status wiederzugeben.

Der Soll-Zustand wurde erfüllt und um weitere Funktionalitäten erweitert.

Dank der modularen Programmierung besteht weiterhin die Möglichkeit, das Programm zu erweitern.

## 8. Nutzungsanalyse

### 8.1. Beschreibung

Um sicherzustellen, dass der Personenzähler korrekt funktioniert, wurde dieser mithilfe einer Testumgebung getestet. Hierbei sind die Sensoren statt in einem Türrahmen an eine Platte befestigt worden. Die Maximal erlaubte Anzahl an Personen wird auf zwei und der Zeitsprung auf 20ms gesetzt. Aufgrund der geringen Reichweite der Sensoren wird statt einer vorbeilaufenden Person eine Hand genutzt, um mehrere Szenarien zu testen. Um Komplexe Erklärungen zu vermeiden, wird dennoch davon ausgegangen, dass der Personenzähler am Türrahmen des Raumes B11 befestigt worden ist und statt Händen Personen sich daran vorbeibewegen.

Diese sind:

- Es gehen solange Personen in den Raum, bis der Raum voll ist. Anschließend gehen diese Personen wieder raus.  
Siehe hierfür „I.II. Maximum erreicht“.
- Es gehen mehr Personen raus als rein.  
Siehe hierfür „I.III. Minimum erreicht“.
- Eine Person stellt sich vor den Bestätiger und blockiert so den Durchgang.  
Siehe hierfür „I.IV. Durchgang blockiert“.
- Die SD Fehlt bzw. ist fehlerhaft.  
Siehe hierfür „I.V. SD Fehlerhaft“

### 8.2. Maximum erreicht

#### 8.2.1. Beschreibung

Bei diesem Testdurchlauf soll geprüft werden, wie der Personenzähler bei Erreichen einer Personenzahl reagiert, welche höher als die maximal erlaubte ist. In diesem Fall fünf. Anschließend soll überprüft werden, wie er beim Rückgang zurück auf null Personen reagiert.

Dieser Test dient zum einen der Überprüfung der Auf- bzw. Abwärtszählung mitsamt der dazugehörigen Statusänderung und zum anderen der Überprüfung der Richtigkeit der erstellten Datensätze.

#### 8.2.2. Erwartung

Voraussichtlich wird der Arduino bis zum Erreichen von zwei Personen im Raum ausgeben, dass der Raum nicht voll ist. Anschließend, von zwei bis hin zu fünf Personen im Raum, wird der Personenzähler voraussichtlich wiedergeben, dass der Raum voll ist. Dies wird sich ggf. bis zur Rückkehr auf zwei nicht ändern. Bei Erreichen von weniger als zwei Personen im Raum wird mit großer Wahrscheinlichkeit ausgegeben, dass der Raum nicht voll ist.

Nun zum Datensatz.

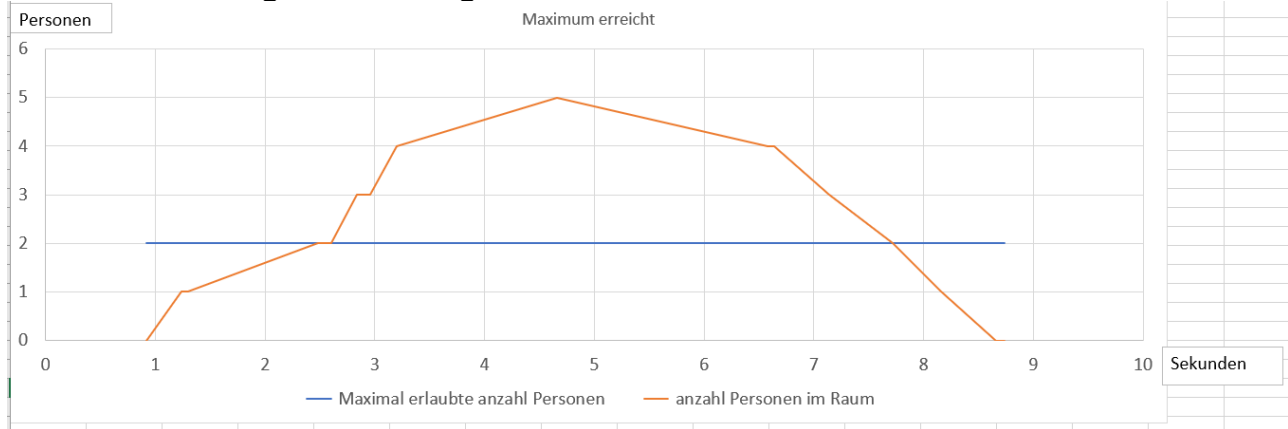
Während der Aufwärtszählung werden die Spalten C, die Personenanzahl, E sowie F, die Globalen Personenzählungen, wahrscheinlich um eins erhöht. Dabei wird ggf. Spalte D, welche die Anzahländerung pro Datensatz angibt, auf + gesetzt, um die Erhöhung darzustellen.

Beim anschließenden herunterzählen wird Spalte C voraussichtlich bis zu einem Wert

von null heruntergezählt, während die Globalen Personenzählungen in Spalte E und F um eins erhöht werden.

Der Status wird mit großer Wahrscheinlichkeit ab zwei Personen im Raum auf „voll“ gesetzt und sonst auf „nicht voll“

### 8.2.3. Auswertung der Messergebnisse



Vorschaubild. Siehe „Dateien/Testdurchlauf/maximum erreicht/Diagramm Maximum Durchlauf.png“ aus Seite 2 im Digitalen Anhang für eine vergrößerte Ansicht.

Am Diagramm ist an der blauen Linie die maximal erlaubte Anzahl an Personen ersichtlich. Die Orange Linie gibt die aktuelle Anzahl an Personen im Raum wieder. Hier muss nun, solange die Orange Linie unter der blauen ist, beim LCD-Display „nicht voll“ als Status wiedergegeben werden. In diesem Fall mit null sowie einer Person im Raum.



Vorschaubild. Siehe „Dateien/Fotos/LCD/einePerson.png“ im Digitalen Anhang für eine vergrößerte Ansicht.

Wie an dem Foto erkennbar, wird dies erfüllt. Anschließend, ab Erreichen von zwei oder mehr Personen im Raum, wird als Status „voll“ ausgegeben.



Vorschaubild. Siehe „Dateien/Fotos/LCD/zweiPersonen.png“ im Digitalen Anhang für eine vergrößerte Ansicht des linken Bildes.

Siehe „Dateien/Fotos/LCD/fuenfPersonen.png“ im Digitalen Anhang für eine vergrößerte Ansicht des rechten Bildes.

Wie erkennbar ist, wird dies ebenso erfüllt. So ist erkennbar, dass der Personenzähler die Zählung samt der Statussynchronisierung draufhat. Am Diagramm ist des Weiteren erkennbar, dass die Personenzahl im Raum gegen Ende wieder unter zwei, auf eins, geht. Entsprechend hat der Status sich auf „nicht voll“ zu stellen.



Vorschaubild. Siehe „Dateien/Fotos/LCD/einePerson.png“ im Digitalen Anhang für eine vergrößerte Ansicht.

Erkennbar ist, dass die Personenzahl unter zwei ist und der Status entsprechend auf „nicht voll“ steht. Entsprechend funktioniert die LCD-Ausgabe.



## Datensatz:

Raumnummer:	B11						
Zeitstempel [s]	Maximal erlaubte anzahl Personen	anzahl Personen im Raum	Anzahländerung	Personenzahl Gesamt	Personenzahl Gesamt raus	Personenzahl Gesamt rein	Status
0.92	2	0	0	0	0	0	0 Durchgang wird blockiert
0.92	2	0	0	0	0	0	0 Durchgang wieder frei
1.24	2	1 +	1	1	0	0	1 nicht voll
1.3	2	1	0	1	0	0	1 Durchgang wird blockiert
1.3	2	1	0	1	0	0	1 Durchgang wieder frei
2.48	2	2 +	2	2	0	0	2 nicht voll
2.54	2	2	0	2	0	0	2 Durchgang wird blockiert
2.56	2	2	0	2	0	0	2 Durchgang wird blockiert
2.58	2	2	0	2	0	0	2 Durchgang wird blockiert
2.6	2	2	0	2	0	0	2 Durchgang wird blockiert
2.6	2	2	0	2	0	0	2 Durchgang wieder frei
2.84	2	3 +	3	3	0	0	3 voll
2.9	2	3	0	3	0	0	3 Durchgang wird blockiert
2.92	2	3	0	3	0	0	3 Durchgang wird blockiert
2.94	2	3	0	3	0	0	3 Durchgang wird blockiert
2.96	2	3	0	3	0	0	3 Durchgang wird blockiert
2.96	2	3	0	3	0	0	3 Durchgang wieder frei
3.2	2	4 +	4	4	0	0	4 voll
4.66	2	5 +	5	5	0	0	5 voll
6.58	2	4 -	4	6	1	1	5 voll
6.64	2	4	0	6	1	1	5 Durchgang wird blockiert
6.64	2	4	0	6	1	1	5 Durchgang wieder frei
7.14	2	3 -	7	7	2	2	5 voll
7.72	2	2 -	8	8	3	3	5 voll
8.16	2	1 -	9	9	4	4	5 voll
8.66	2	0 -	10	10	5	5	5 nicht voll
8.72	2	0	0	10	5	5	5 Durchgang wird blockiert
8.74	2	0	0	10	5	5	5 Durchgang wird blockiert
8.74	2	0	0	10	5	5	5 Durchgang wieder frei

Vorschau. Siehe „Dateien/Testdurchlauf/maximum erreicht/Tabelle.xlsx“ im Digitalen Anhang für eine vergrößerte Ansicht.

Wie an der Tabelle an u.A. Zeile drei mit einem Zeitstempel von 1,24 Sekunden erkenntlich ist, werden mit der Erhöhung der Personenzahl im Raum auch die Globale Personenzahl sowie die Globale Personenzahl welche reingeht um eins erhöht. Weiterhin wird bei jeder Erhöhung Anzahländerung auf plus gesetzt.

Ebenso ist an u.A. Zeile zwanzig mit dem Zeitstempel 6,58 Sekunden ersichtlich, dass es bei der Verringerung der Personenzahl im Raum andersherum mit demselben Prinzip funktioniert. Die Globale Personenzahl sowie die Globale rausgehende Personenzahl wird um eins erhöht. Die Personenzahl im Raum wird um eins verringert. Dabei wird die Anzahländerung auf minus gesetzt.

Sobald der Durchgang blockiert wird, ist Anzahländerung auf 0 gesetzt, da weder jemand rein- noch rausgeht.

Rechts beim Status wird ausgegeben, ob der Raum voll, nicht voll oder ob der Durchgang blockiert ist. Bei Zeile zwanzig mit dem Zeitstempel 6,58 z.B. ist der Status voll, was widerspiegelt, dass mehr Personen im Raum sind als erlaubt.

### 8.2.4. Interpretation der Ergebnisse

Unter Heranziehung der Erwartungen wird ersichtlich, dass die Programmierung samt der Ausgabe und der Speicherung des Personenzählers stimmen.

Der Personenzähler wurde nicht in gleichen Zeitabständen betätigt, um diesen auf veränderte Zeitabstände zu überprüfen.

Dennoch wird ersichtlich, dass die unter Punkt „9.3. Geplante Features“ genannte Veränderung der Sensorreihenfolge in zukünftigen Updates übernommen wird, da es zeitlich nicht mehr möglich ist, die Anpassung samt Testung durchzuführen.

### 8.3. Minimum erreicht

#### 8.3.1. Beschreibung

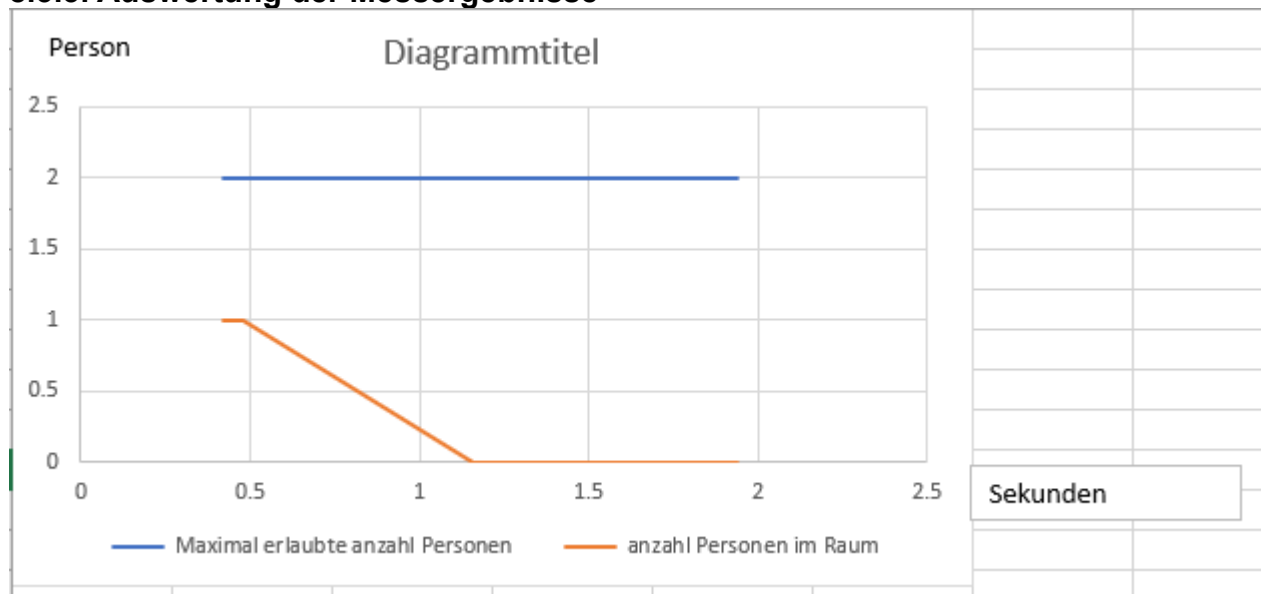
Ziel des Testdurchlaufes soll es sein, zu überprüfen wie der Personenzähler damit umgeht, wenn mehr Personen rausgehen als reingehen. Hierzu soll die Personenzahl erstmal auf eins erhöht werden. Anschließend sollen zwei Personen rausgehen, sodass die Reaktion überprüft werden kann.

Dieser Test dient zum einen der Überprüfung der Programmierung und zum anderen der Überprüfung der Richtigkeit der erstellten Datensätze.

#### 8.3.2. Erwartung

Es wird davon ausgegangen, dass die Personenzahl um eins erhöht und anschließend um eins verringert wird, da die Programmierung eine Verringerung unter null nicht zulässt. Beim zweiten, der rausgeht, wird die Globale Personenzahlbewegung voraussichtlich nicht um diesen erhöht.

#### 8.3.3. Auswertung der Messergebnisse



Vorschau bild. Siehe „Dateien/Testdurchlauf/minimum erreicht/Tabelle.xlsx“ Auf Folie 2 im Digitalen Anhang für eine vergrößerte Ansicht.

Am Diagramm ist erkennbar, dass die maximale Anzahl an Personen, die blaue Linie, nicht von der aktuellen Personenzahl, der Orangen Linie, erreicht wird. Stattdessen bleibt die aktuelle Anzahl nach den zwei verlassenden Personen bei null.

Dies ist auch am LCD-Display ersichtlich



Vorschaubild. Siehe „Dateien/Fotos/LCD/einePerson.png“ im Digitalen Anhang für eine vergrößerte Ansicht vom linken Bild.

Vorschaubild. Siehe „Dateien/Fotos/LCD/nullPersonen.png“ im Digitalen Anhang für eine vergrößerte Ansicht vom rechten Bild.

Wie am Anfang erkenntlich, ist der Status bei „nicht Voll“. Weiterhin ist die Personenzahl im Raum unter der Maximal erlaubten Anzahl.

Raumnummer	B11							
Zeitstempel	Maximal erla	anzahl Perso	Anzahlaende	Personenzah	Personenzah	Personenzah	Status	
0.42	2	1 +		1	0	1	nicht voll	
0.48	2	1	0	1	0	1	Durchgang wird blockiert	
0.48	2	1	0	1	0	1	Durchgang wieder frei	
1.16	2	0 -		2	1	1	nicht voll	
1.94	2	0	0	3	1	1	Durchgang wird blockiert	
1.94	2	0	0	3	1	1	Durchgang wieder frei	

Vorschaubild. Siehe „Dateien/Testdurchlauf/minimum erreicht/Tabelle.xlsx“ auf Folie 1 im Digitalen Anhang für eine vergrößerte Ansicht.

Wie ersichtlich ist, wird die Globale Personenzahl sowie die sich Global reinbewegenden beim Reingehen um eins erhöht. Beim Verlassen des Raumes wird die Globale Personenzahl um insgesamt zwei erhöht, während die Global rausgehenden Zahl um eins erhöht wird. Dies wird in einem Update ausgebessert, da es zeitlich nicht mehr mit der Testung machbar ist.

#### 8.3.4. Interpretation der Ergebnisse

Unter Heranbeziehung der Erwartungen wird ersichtlich, dass die Programmierung mitsamt der Ausgabe und Speicherung zum Großteil funktionieren, aber es bei der Eintragung der Globalen Personenzahlwerte stockt.

Wie bereits bei Kapitel „8.3.3. Auswertung der Ergebnisse“ ersichtlich, wird dies in einer zukünftigen Version behoben.

## 8.4. Durchgang blockiert

### 8.4.1. Beschreibung

Bei diesem Testdurchlauf wird der Bestätiger blockiert. Es wird überprüft, ob der Status in der dritten LCD<sub>14</sub>-Zeile sowie der Status im Datensatz übereinstimmen.

### 8.4.2. Erwartung

Es wird erwartet, dass in der dritten Zeile beim blockieren des Bestätigers „beweg dich“ ausgegeben wird.

Weiterhin, dass Anzahländerung im Datensatz 0 und Status bis zum Ende der Blockierung „Durchgang wird blockiert“ wiedergibt. Anschließend „Durchgang wieder frei“.

### 8.4.3. Auswertung der Messergebnisse

Raumnummer: B11							
Zeitstempel [s]	Maximal erlaubte anzahl Personen	anzahl Personen im Raum	Anzahlaenderung	Personenzahl Gesamt	Personenzahl Gesamt raus	Personenzahl Gesamt rein	Status
1.64	2	0	0	0	0	0	0 Durchgang wird blockiert
1.66	2	0	0	0	0	0	0 Durchgang wird blockiert
1.68	2	0	0	0	0	0	0 Durchgang wird blockiert
1.7	2	0	0	0	0	0	0 Durchgang wird blockiert
1.72	2	0	0	0	0	0	0 Durchgang wird blockiert
1.72	2	0	0	0	0	0	0 Durchgang wieder frei

Vorschaubild. Siehe „Dateien/Testdurchlauf/Durchgang blockiert/Tabelle.xlsx“ auf Folie 1 im Digitalen Anhang für eine vergrößerte Ansicht.

Wie an der Tabelle ersichtlich wird solange der Durchgang blockiert wird unter Anzahländerung „0“ und unter Status „Durchgang wird blockiert“ ausgegeben. Beim freiwerden „Durchgang wieder frei“.

### 8.4.4. Interpretation der Ergebnisse

Unter Heranziehung der Erwartung wird ersichtlich, dass es funktioniert.

## 8.5. SD Fehlerhaft

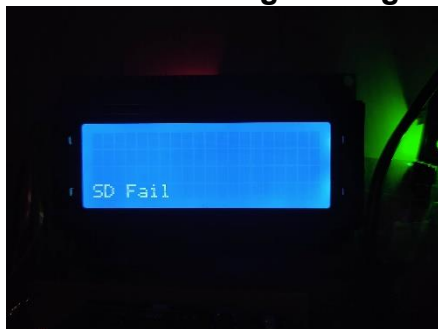
### 8.5.1. Beschreibung

Wenn beim Start die SD Karte fehlt soll eine Fehlermeldung ausgegeben werden.

### 8.5.2. Erwartung

Der Personenzähler gibt bei fehlen eine Fehlermeldung in der letzten Zeile wieder.

### 8.5.3. Auswertung der Ergebnisse sowie Interpretation



Die SD wurde nicht angeschlossen. Dies hat der Personenzähler beim Start erkannt und die Fehlermeldung „SD Fail“ ausgegeben.

Unter Heranziehung der Erwartungen wird ersichtlich, dass die Ausgabe der Fehlermeldung funktioniert.

Vorschaubild. Siehe „Dateien/Fotos/LCD/sd fail.png“ im Digitalen Anhang für eine vergrößerte Ansicht.

## **9. Fazit**

### **9.1. Reflektion**

Anhand von Kapitel 7, dem Soll/Ist Vergleich, ist ersichtlich, dass das Projekt in fast vollem Umfang erfüllt worden ist.

Es wurde geplant, dass der Arduino Datensätze erstellt, einen Zeitstempel erstellt und das Ergebnis Visuell wiedergibt.

Lediglich beim Zeitstempel gibt es Probleme. Diese bestehen nicht in Soft- bzw. Hardwareproblemen, sondern darin, dass der Nutzer die Startuhrzeit Manuell Nachtragen muss, da keine RTC eingebaut ist und entsprechend der Delay für den Zeitstempel genutzt wird

Zum Erreichen des Projektes trugen vor allem meine Vorkenntnisse in der Programmierung sowie Hilfe Seitens der Lehrkräfte und einiger Mitschüler bei.

Zu guter Letzt habe ich das Ziel, meine Kenntnisse mit dem Hard- und Softwareumgang sowie mit dem Arduino umzugehen, insofern vertieft, dass ich sicher im belegen der Pins sowie besser im Umgang mit der Programmierung bin. Beim Softwareumgang vor allem das vertiefen der genutzten Programmierstruktur sowie das verbessern meiner Programmierkenntnisse.

### **9.2. Probleme**

Während des Projektes ist es zu vielen Problemen gekommen. Hauptsächlich im Umgang mit der Hardware sowie des Budgets.

- Angefangen bei der Eingabemöglichkeiten. Es hatte relativ lange gedauert, bis eine Festlegung der Messwerteingabe festgelegt worden ist. Anfangs unter Nutzung von Buttons, doch dann mit der Überlegung zwischen PIRs sowie Infrarotsensoren.
  - Ein PIR hat eine größere Reichweite, aber dafür eine gewisse Wartezeit, bis er wieder für die Zählung genutzt werden kann. Dies würde zwischen jeder Aktivierung zu unnötigen Wartezeiten führen. Weiterhin hat der PIR bei Testdurchläufen nicht funktioniert, weshalb er nicht genutzt worden ist.
  - Ein Infrarotsensor hat in derselben Preisklasse eine geringere Reichweite, arbeitet dafür aber in Echtzeit und ohne Wartezeit. Weiterhin ist die Sonne eine Infrarotquelle, welche zu falschen Aktivierungen der Sensoren führt. Dies kann jedoch durch Abschirmung sowie Umpositionierung der Sensoren vermeiden werden.

Die Entscheidung ist am Ende auf den Infrarotsensor gefallen, da er Preislicher ist, ohne Wartezeiten auskommt und bei den Testdurchläufen eine höhere Zuverlässigkeit vorgewiesen hat.

- Budgetbedingt hat es keine Möglichkeit gegeben, eine RTC sowie höherwertige Komponenten zu verbauen, sodass ein Zeitstempel zur zeitlichen Orientierung dient sowie eine Praktische Umsetzung des Projektes nicht möglich ist.

- Wenn sich jemand vor dem mittleren Sensor aufhält wird eine Meldung ausgegeben, dass diese Person weitergehen soll. Hierbei besteht das Problem, dass durchgehend Datensätze erstellt werden solange sich der Sensor blockiert wird. Aufgrund der übrigen Projektzeit besteht nicht die Möglichkeit, dies zu korrigieren, doch wird in zukünftigen Versionen ausgebessert.

### 9.3. Geplante Features

Für das Projekt sind noch weitere Updates sowie Upgrades geplant, für deren Umsetzung die Zeit bzw. das Budget nicht gereicht hatte.

Geplant sind:

- Verbauen von Infrarotsensoren mit höherer Reichweite.
- Ein Display mit Touchscreen, um die Konfiguration unabhängig von der IDE zu machen.
- Eine Menüführung, sodass die IDE lediglich für Update sowie Wartungszwecke verwendet werden muss.
- Abschirmung für die Infrarotsensoren.
- Gehäuse für den Personenzähler.
- Button, um bei Fehlzählungen die Personenzahl im Raum zurückzusetzen.
- Die Programmierung wird insofern verändert, dass der mittlere Sensor nicht mehr der Bestätiger ist, sondern der Ausgangspunkt, von welchem aus der innere oder äußere Sensor aktiviert werden können. Zeitlich konnte dies aufgrund der dazugehörigen Testphase nicht mehr umgesetzt werden.

Geplant ist:

- Wenn jemand reingeht, wird erstmal der mittlere und anschließend der innere Sensor betätigt. Der äußere Sensor macht solange nichts
- Wenn jemand rausgeht, wird erstmal der mittlere und anschließend der äußere Sensor betätigt. Der innere Sensor macht solange nichts.
- Die Aufgabe der Blockierungserkennung werden entsprechend der innere sowie der äußere Sensor machen, vorausgesetzt diese werden vor dem mittleren Aktiviert.

## Anlagen

### I. Projekttagbuch

#### I.I. Organisatorisches

Datum	Tätigkeit	Erreicht	Nächstes Ziel	gelernt	Dauer in Stunden
03.01.2021	Arduino IDE installiert	Kann anfangen zu Programmieren	-Arduino erwerben	-Arduino IDE ist Grundlage für Programmierung mit Arduino	0,1
21.01.2021	Zeitplan erstellt und abgegeben	Zeitplan erstellt	-Inhaltsverzeichnis erstellen	-	0,5
18.02.2021	Inhaltsverzeichnis angefertigt und abgegeben	Inhaltsverzeichnis erstellt	-Einleitung erstellen	-	0,5
05.03.2021	Einleitung angefertigt und abgegeben	Einleitung erstellt und abgegeben	-Hardwarekonzept erstellen	-	0,3
26.03.2021	Hardwarekonzept angefertigt und abgegeben	Hardwarekonzept erstellt und abgegeben	-Softwarekonzept anfertigen	-	1
29.04.2021	Softwarekonzept angefertigt	Softwarekonzept erstellen	-	-	1

**Gesamtzeit**

**3,4 Stunden**

#### I.II. Recherche

Datum	Tätigkeit	Erreicht	Nächstes Ziel	gelernt	Dauer in Stunden
01.03.2021	Über Verkabelung recherchiert	Jumperkabel erworben	-Über Arduino recherchieren	-Komponenten werden mit Kabeln verbunden	1
08.03.2021	Über Arduino recherchiert	Arduino Starter Kit erworben	-Über benötigte Komponenten recherchieren	-Arduino ist wichtig für	1
01.04.2021	Über Speichermöglichkeiten recherchiert	SD Shield Erworben	-Ein- und Ausgabegeräte recherchieren, -Speichermöglichkeit durchgehen	-Arduino hat nur RAM	2
01.04.2021	Über Ein- und Ausgabe recherchiert	Taster sowie I2C-Display Erworben	-Lötkolben für LCD Pins erwerben	-Ein- und Ausgabe ist nicht im Arduino integriert	2
01.04.2021	Über Lötkolben recherchiert	Lötkolben erworben	-LCD Display löten	-Nicht alle Komponenten sind Einbaufertig	1
14.04.2021	Über alternative Eingabemöglichkeiten recherchiert	TCRT5000 Sensoren erworben	-TRCT5000 Sensoren einbauen	-Sensoren sind für das Projekt besser geeignet	2
24.04.2021	Über Stromversorgung recherchiert	Netzteil erworben	-Über Backup recherchieren	-Falsches Netzteil führt zu Schäden	1
21.05.2021	Über Fall eines Spontanen Ausfalls recherchiert	Backupplan erstellt	-	-Immer auf Nummer sicher gehen	2

**Gesamt**

**12 Stunden**

### I.III. Tätigkeit

Datum	Tätigkeit	Erreicht	Nächstes Ziel	gelernt	Dauer in Stunden
01.04.2021	Programmiert	Mit Taster funktionierender Personenzähler	LCD löten	-	4
01.04.2021	gelötet	LCD verlötet und verbaut	Programmierung auf LCD anpassen	Löten	1
02.04.2021	Programmierung angepasst (LCD)	Ausgabe via LCD	Über alternative Programmierweisen recherchieren	Werteausgabe per LCD	2
14.04.2021	Programmierung angepasst (Sensoren)	Personenzähler mit Sensoren	Testen	Umgang mit Sensoren	4
14.04.2021	Programm getestet	Programm getestet	Programm auf drei Sensor Taktik umformen	Programmtestung	1
18.04.2021	Programmierung angepasst (Drei Sensor Taktik)	Personenzähler mit drei Sensor Taktik	Testen auf verschiedene Szenarien	Umgang mit Sensoren	8
20.04.2021	Programm getestet	Programm auf Herz und Nieren geprüft	Dokumentation vollenden	Programmtestung	3

**Gesamt**

**23 Stunden**

### I.IV. Dokumentation

Datum Anfang	Datum Ende	Kapitel	Geplante Zeit	Tatsächliche Zeit
05.03.2021	17.06.2021	1. Umfeld	1	1
05.03.2021	17.06.2021	2. SOLL-Zustand	2	1
21.01.2021	17.06.2021	3. Planung	1	1
20.04.2021	17.06.2021	4. IST-Zustand	1	2
01.03.2021	17.06.2021	5. Kostenkalkulation	3	3
01.04.2021	17.06.2021	6. Implementierung	2	3
20.04.2021	17.06.2021	7. SOLL/IST Vergleich	1	1
20.04.2021	17.06.2021	8. Nutzungsanalyse	4	3
24.04.2021	17.06.2021	9. Fazit	1	1
17.06.2021	17.06.2021	I. Projekttagbuch	1	2
17.06.2021	17.06.2021	II. Glossar	1	1
17.06.2021	17.06.2021	III. Verwendete Werkzeuge	1	1
26.03.2021	17.06.2021	IV. Konzept	2	2
16.06.2021	17.06.2021	V. Kundendokumentation	1	1
05.03.2021	17.06.2021	VI. Quelltextdokumentation	4	3

**Gesamt**

**26 Stunden**

**26 Stunden**

**Zeit Gesamt**

**64,4 Stunden**



## II. Glossar

Nummer	Begriff	Bedeutung
1	BBS Verden	Schule in Verden
2	Personenzähler	Meist mechanisches Gerät zur Messung auf Knopfdruck
3	Datensatz	Eine Gruppe inhaltlich zusammenhängender Daten
4	Sensorik	Anwendung von Sensoren zur Messung und Kontrolle von Veränderungen
5	Corona	Meldepflichtige Infektionskrankheit
6	Präposition	Zeigt, wie etwas sich zum anderen verhält
7	Budget	Gegenüberstellung von Einnahmen und Ausgaben
8	Integriert	Zusammenschluss einzelner Elemente zu einem System
9	delay	Zeitintervall, um welchen ein Ereignis verzögert wird
10	RTC	Uhr, welche die Physikalische Zeit misst
11	CSV	Format mit Aufbau einer Textdatei zur Speicherung einfach strukturierter Daten
12	XLSX	Open Office XML basiertes Dateiformat
13	LCD	Auf Flüssigkristallen basierte Anzeige
14	int	Ganzzahliger Datentyp
15	float	Datentyp zur angenäherten Darstellung Reeler Zahlen
16	string	Datentyp für endlichen Reihenfolge von Zeichen
17	char	Datentyp zur Wiedergabe einzelner Zeichen
18	file	Objekt der Arduino SD.h Bibliothek zum beschreiben und Lesen von Dateien
19	void	Datentyp ohne Rückgabewert
20	Struktogramm	Diagrammtyp zur Darstellung von Programmentwürfen
21	NSD	Dateiendung für Dateien vom Structorizer
22	Infrarot	Elektromagnetische Strahlung im Spektralbereich zwischen sichtbarem Licht und Terahertzstrahlung

Vorschau. Siehe „Dateien/Quellen/Quellenverzeichnis“ für Quellen.

### III. Verwendete Werkzeuge

#### III.I. Beschreibung

Für die Umsetzung des Projektes sind verschiedene Tools eingesetzt worden.  
Im Folgenden eine Liste der verwendeten Werkzeuge nach Kategorie.  
Nach Gesprächen mit Herr Düren am 17.06.2021 bleibt die Nennung derer Quellen aus.

#### III.II. Software

Zweck	Programm
Programmierungsumgebung	Arduino IDE
Struktogrammerstellung	Structorizer
Anfertigen der Dokumentation	Microsoft Office Suite
Erstellen der Schaltskizze	Fritzing

#### III.III. Hardware

Zweck	Programm
Festigen der LCD Pins	Lötkolben
Konfigurieren der Potentiometer mehrerer Komponenten	Kreuzschraubenzieher

## IV. KONZEPT

### IV.I. Bibliotheken

Name	Funktion
Wire.h	Für die Kommunikation mit I2C Geräten
LiquidCrystal_I2C.h	Um über i2c mit dazugehörigem LCD zu kommunizieren
SPI.h	Um mit SPI Geräten zu kommunizieren
SD.h	Um auf die SD zu schreiben bzw. von dieser zu lesen

Sämtliche Bibliotheken sind, vorausgesetzt sie sind nicht in der IDE integriert, im zugehörigen Ordner unter „Dateien/Programm/Bibliotheken/“ im digitalen Anhang zu finden.

### IV.II. Variablen

Name	Typ	Funktion
maximaleAnzahlPersonen	int <sub>15</sub>	Maximal im Raum erlaubte Anzahl an Personen
aktuelleAnzahlPersonen	Int	Aktuelle Personenzahl im Raum
zeitMesser	float <sub>16</sub>	Wird als Zeitstempel genutzt
zeitSprung	int	Zeitabstand zwischen Durchläufen. In Millisekunden
raumNummer	string <sub>17</sub>	Gibt Raumnummer wieder
Status	string	Gibt aus, ob noch Platz ist oder sich jemand vor den mittleren Sensoren stellt.
anzahlAenderung	char <sub>18</sub>	Gibt im Datensatz an, ob jemand hinzugekommen ist oder rausgegangen ist
personenZahlGesamt	int	Insgesamt vorbeigegangene Personen
personenZahlRein	int	Insgesamt reingegangene Personen
personenZahlRaus	int	Insgesamt rausgegangene Personen
modulSDPin	int	Variable für SD Datenpin
myFile	file <sub>19</sub>	Für Zugriff auf Dateien
Dateiname	string	Um auf bestimmte Datei zuzugreifen
pinSensorRein	int	Pin für Sensor fürs reingehen
pinSensorRaus	int	Pin für Sensor fürs rausgehen
pinSensorMitte	int	Pin für Sensor in der Mitte
fertig	int	Variable um Änderung der Personenzahl zu bestätigen

Typ	Beschreibung
int	Dient zur Speicherung ganzer Zahlen.
float	Dient zur Speicherung von Gleitkommazahlen
string	Dient zur Speicherung von endlichen Zeichenketten
char	Dient zur Speicherung einzelner Zeichen
file	Dient zur Speicherung einer Datei als Variable
void	Funktionstyp ohne Rückgabewert

### IV.III. Unterfunktionen

#### IV.III.I. Beschreibung

Neben den Setup- sowie Loopteil Funktionen hat das Programm insgesamt zwei Unterfunktionen. Die erste, vom Typ void<sub>20</sub> und mit dem Namen Speichern(), dient der Speicherung der Datensätze. Eine zweite Unterfunktion, vom Typ void und mit dem Namen zeitMesserErhoehung(), dient der Erhöhung des Zeitstempels.

#### IV.III.II. Speichern

Die Funktion dient zum Speichern der Datensätze und wird bei jeder Betätigung des Bestätigers aufgerufen. Für eine Erklärung der Variablenfunktion siehe Kapitel „IV.II. Variablen“.

```
void Speichern()                //Speicherfunktion
{
  myFile = SD.open(Dateiname, FILE_WRITE); //SD objekt zuweisen
  myFile.print(zeitMesser, 3);
  myFile.print(";");
  myFile.print(maximaleAnzahlPersonen);
  myFile.print(";");
  myFile.print(aktuelleAnzahlPersonen);
  myFile.print(";");
  myFile.print(anzahlAenderung);
  myFile.print(";");
  myFile.print(personenZahlGesamt);
  myFile.print(";");
  myFile.print(personenZahlRaus);
  myFile.print(";");
  myFile.print(personenZahlRein);
  myFile.print(";");
  myFile.print(Status);
  myFile.println(";");
  myFile.close();
}
```

Siehe „Dateien/Programmcode/Personenzaehler/Personenzaehler.ino“ für Vollständigen Programmcode

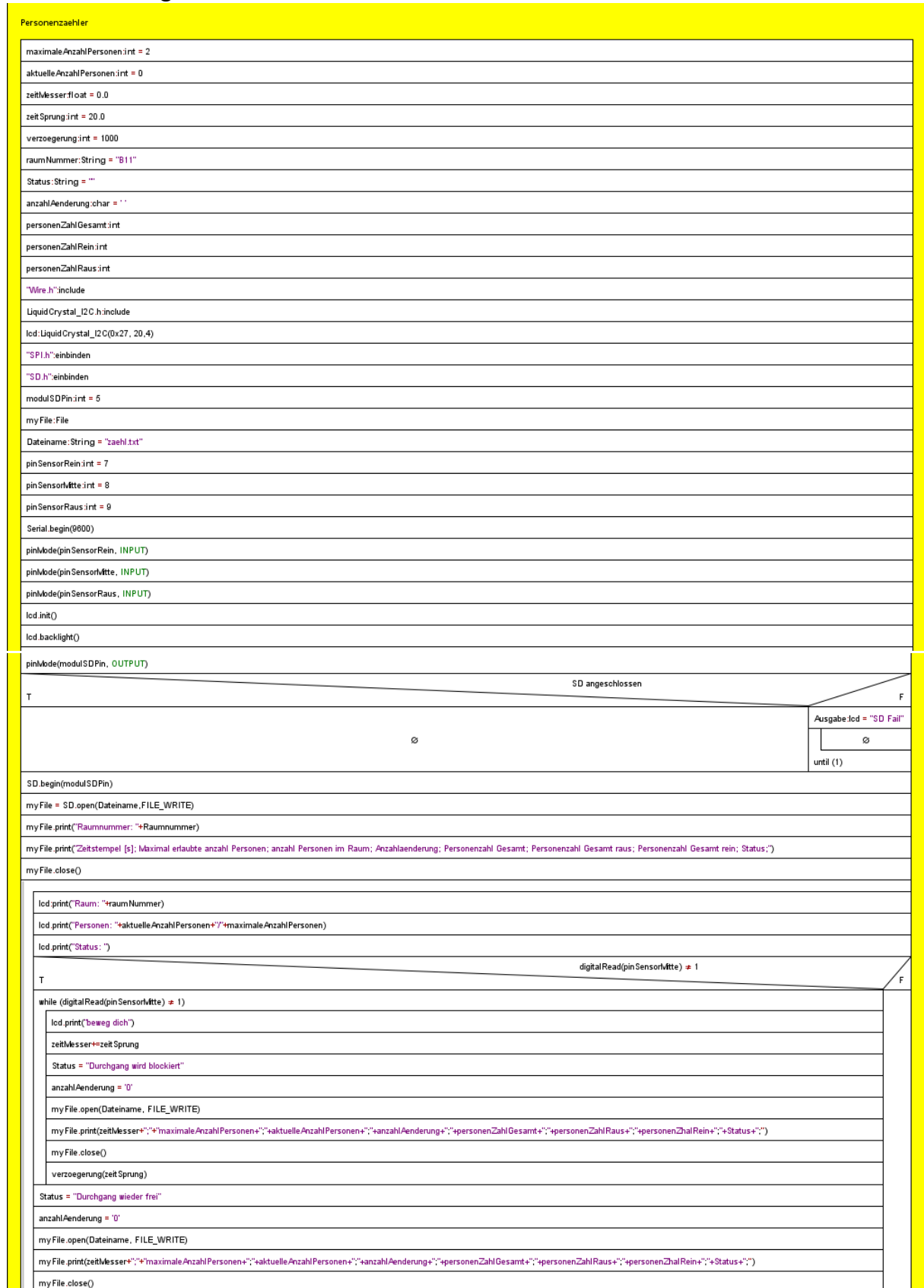
#### IV.III.III. Zeitstempel erhöhen

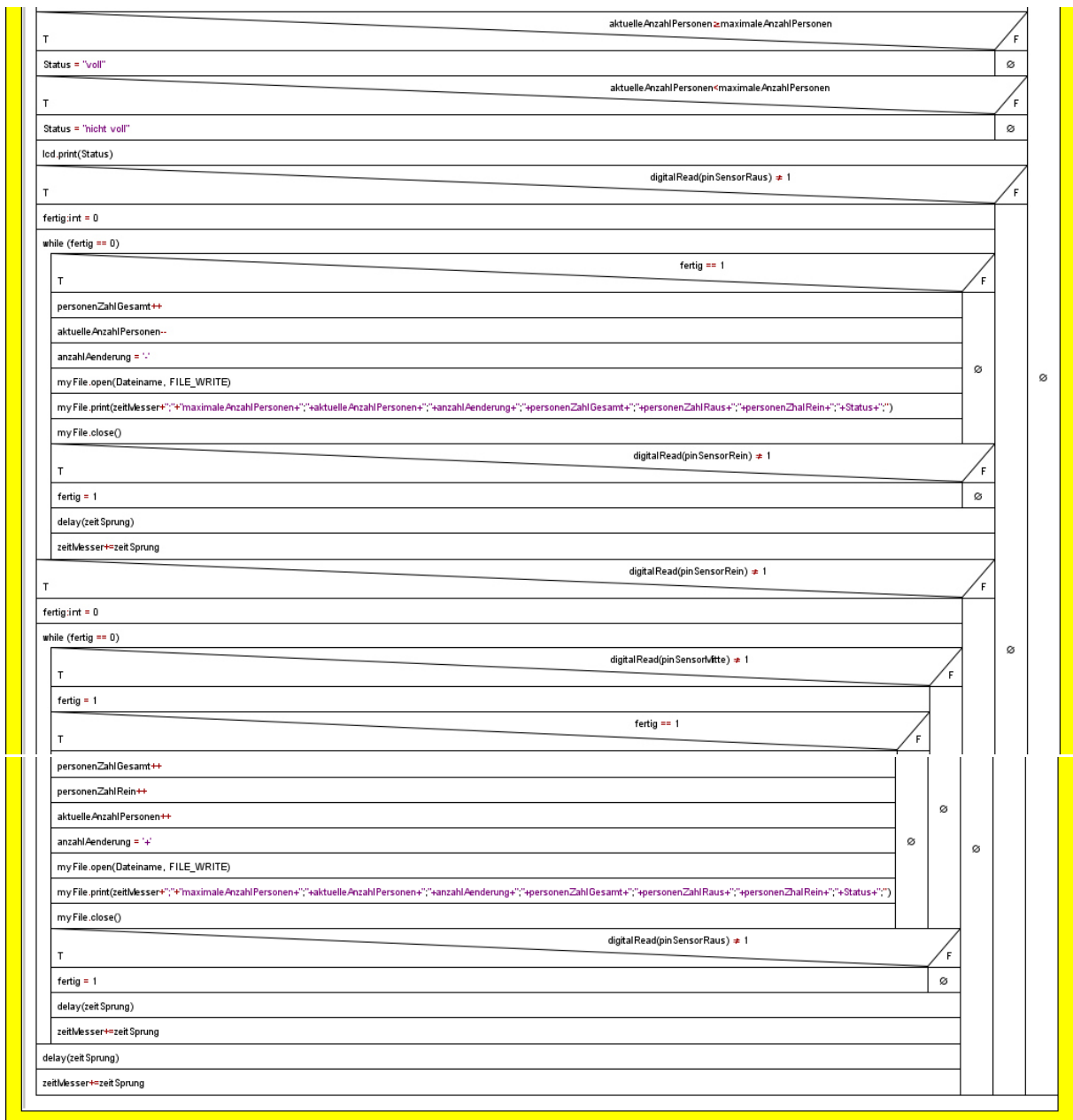
Die Funktion dient zum erhöhen des Zeitstempels um die Dauer des Delays  
Für eine Erklärung der Variablenfunktion siehe Kapitel „IV.II. Variablen“.

```
void zeitMesserErhoehung() //Funktion um zeitMesser um zeitSprung Sekunden erhoeuen
{
  zeitMesser = zeitMesser+((float)zeitSprung/1000); //zeitMesser um zeitSprung Sekunden
}
```

Siehe „Dateien/Programmcode/Personenzaehler/Personenzaehler.ino“ für Vollständigen Programmcode

## IV.IV. Struktogramm<sup>21</sup>





Siehe „Dateien/Struktogramm/struktogrammVoll.png“ sowie „Dateien/Struktogramm/arduino projekt.nsd“ im digitalen Anhang für Vollständiges Struktogramm. Dabei ist zu beachten, dass alle Einträge drin sind, aber in einer anderen Reihenfolge

Für .nsd<sub>22</sub> Dateien ist der Structorizer notwendig.  
Verfügbar unter [https://www.chip.de/downloads/Structorizer\\_64884440.html](https://www.chip.de/downloads/Structorizer_64884440.html)

#### IV.V. Pinbelegung

Zweck	Bauteilname	Shield Pin	Arduino Pin
Sensor Innen	TCRT5000	+	5V
		-	GND
		s	D9
Sensor Mitte	TCRT5000	+	5V
		-	GND
		s	D8
Sensor Außen	TCRT5000	+	5V
		-	GND
		s	D7
Speicherung	MicroSD	GND	GND
		VCC	5V
		CS	D5
		SCK	D13
		MOSI	D11
		MISO	D12
Ausgabe	20x4 2004 LCD	GND	GND
		VCC	5V
		SDA	A4
		SCL	A5
Stromversorgung	Netzteil 9V 2A	-	Netzteilanschluss

Siehe „Dateien/Hardware/Pinbelegung.xlsx“ im digitalen Anhang für Vollständiges Auflistung.

Da der Arduino nicht über ausreichende Pins für die Stromversorgung der Komponenten besitzt wird diese Funktionalität über das Breadboard erweitert. VCC wird mit der + Reihe und GND mit der – Reihe verbunden.  
Für eine Grafische Darstellung siehe Kapitel „IV.VII. Skizze“

#### IV.VI. Bauteilliste

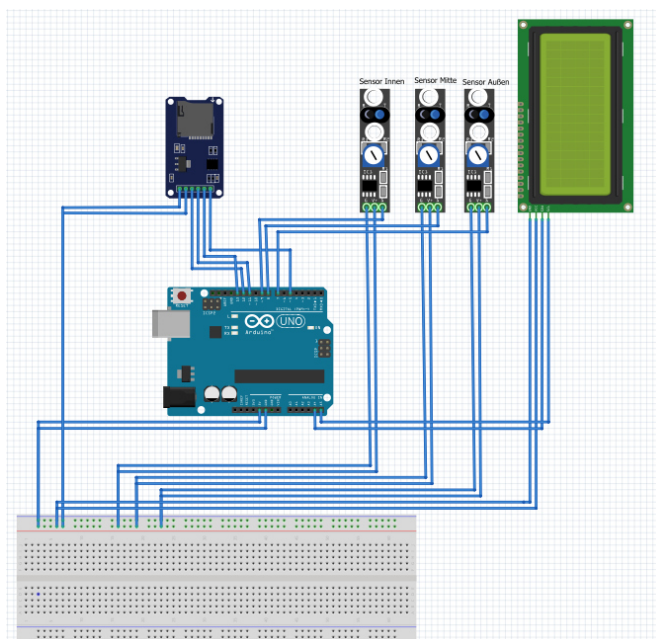
Für das Projekt wird eine Infrarot<sup>23</sup> Sensorik bestehend aus drei Sensoren für die Eingabe genutzt, ein SD Kartenmodul für die Speicherung der Datensätze und ein LCD-Display für die Ausgabe der aktuellen Personenzahl verwendet.

##### Bauteilliste

Anzahl	Bauteilbezeichnung	Funktion
3	TCRT5000	Sollen dem Arduino ein Signal abschicken, um je nach Sensorkombination den Zählwert zu erhöhen oder zu verringern.
1	Micro SD-Reader	Erhält vom Arduino die Datensätze, um diese dann zu speichern.
1	20x4 2004 LCD-Display	Gibt die errechnete Personenzahl aus
1	Arduino Uno Revision 3	Erhält von der Sensorik die Signale zur Veränderung des Zählwertes und gibt die Veränderung mitsamt eines Zeitstempels als Datensatz an die SD-Karte bzw. das Display.
	Jumper Kabel	Zur Verbindung des Arduinos mit den Komponenten

Siehe „Dateien/Hardware/Bauteilliste.xlsx“ im digitalen Anhang für Vollständiges Auflistung.

#### IV.VII. Skizze



Vorschau bild. Siehe „Dateien/skizze/Steckplatine.png“ sowie „Steckplatine.fzz“ im selben Verzeichnis im digitalen Anhang.



## V. Kundendokumentation

### V.I. Bedienungsanleitung

Benötigtes Werkzeug: Kreuzschraubenzieher

1. Bauen Sie den Arduino entsprechend der Pinbelegung Kapitel „IV.V. Pinbelegung“ auf.

Eine grafische Hilfestellung wird durch Kapitel „IV.VII. Skizze“ gestellt.

2. Befestigen Sie die Sensoren Horizontal in Reihe an einer Oberfläche. Achten Sie dabei darauf, dass der innere und der äußere Sensor jeweils rechts und links vom mittleren Sensor sind.

Beachten Sie dabei, dass der Datenpin des inneren Sensors mit dem Digitalen Pin D9, der des mittleren Sensors mit dem Digitalen Pin D8 und der des äußeren Sensors mit dem Digitalen Pin D7 verbunden sind.

3. Legen Sie eine Mikro SD-Karte ein

4. Schließend Sie den Arduino per USB-A auf USB-B Stecker an einen Computer. Vorteilhafter ist ein entsprechendes Netzteil.

Empfohlen wird ein 9V 2A Netzteil.

5. Der LCD leuchtet auf und sollte in etwa folgendermaßen aussehen



6. Sobald Sie am äußeren und anschließend am mittleren Sensor vorbeigehen wird die Personenzahl, links im Bild 0, um eins erhöht.

Solange weniger als die maximale Anzahl an Personen, in diesem Fall zwei, im Raum sind, wird in der vierten Zeile der Status „nicht voll“ ausgegeben. Zu erwähnen ist, dass die Personenzahl im Raum kann nicht unter 0 sinken



Wenn die Aktuelle Anzahl an Personen so groß wie die Maximal erlaubte Anzahl ist oder diese übersteigt, so wird der Status auf Voll gesetzt, und bleibt solange so, bis die Personenzahl im Raum unter die maximal erlaubte Anzahl sinkt.

7. Um den Personenzähler auszuschalten trennen Sie diesen von der Stromversorgung.

## Datensätze

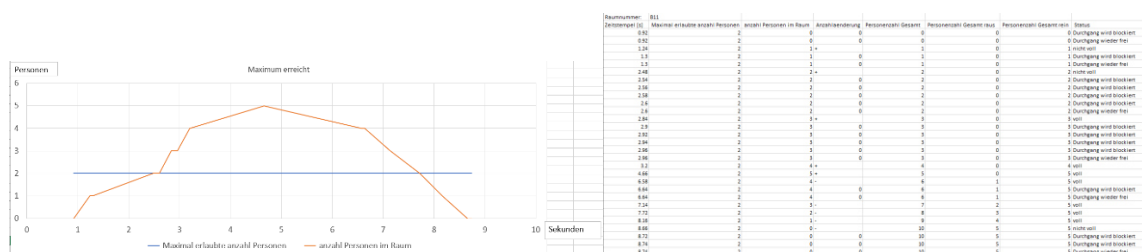
Sobald der mittlere Sensor betätigt wird, egal ob alleine oder nach Betätigung einer der äußeren Sensoren, wird ein Datensatz erstellt. Der Datensatz enthält:

- Raumnummer: Gibt wieder, wo der Personenzähler aufgebaut ist. Dieser Wert kann vom Nutzer im Konfigurationsteil der Programmierung verändert werden
- Zeitstempel: Standardmäßig 20 Millisekunden genauer Zeitstempel. Gibt an, wann der Datensatz nach Start eingetragen worden ist. Die Angabe erfolgt In Sekunden.
- Maximale Anzahl an Personen: Maximal erlaubte Anzahl an Personen im Raum. Dieser Wert kann vom Nutzer im Konfigurationsteil der Programmierung
- Aktuelle Anzahl an Personen: Aktuell im Raum befindliche Anzahl an Personen.
- Anzahländerung: Gibt wieder, ob jemand rein bzw. rausgegangen ist oder ob der Bestätiger blockiert wird.
- Gesamtzahl: Gibt an, wie viele Personen insgesamt vorbeigegangen sind.
- Gesamt raus: Insgesamt rausgegangene Anzahl an Personen.
- Gesamt rein: Insgesamt reingegangene Anzahl an Personen.
- Status: Gibt wieder, ob der Raum voll ist, nicht voll ist oder ob der Bestätiger blockiert oder wieder frei ist.

Weiterhin kann der Zeitsprung, die „Sensibilität“ der Sensoren auf Bewegung, vom Nutzer im Konfigurationsteil der Programmierung verändert werden. Da es sich beim Personenzähler um Sensible Hardware handelt, ist es besonders wichtig, auf eine entsprechende Kühlung zu achten.

Der Arduino legt die Datensätze im CSV<sub>1</sub> Format ab. Da er dieses aber nicht kennt, speichert er diese mit der Endung einer Textdatei ab. Der Vorteil besteht darin, dass Excel durch diese eine Möglichkeit zur Darstellung mit Diagrammen hat.

Folgend eine Darstellung einer ins Excel Format umgespeicherten Tabelle aus Datensätzen:



## Konfigurationsteil.

```
//organisatorisches
int maximaleAnzahlPersonen = 2; //Konfigurierbar
int aktuelleAnzahlPersonen = 0;

float zeitMesser = 0.0;
int zeitSprung = 20; //in ms. Konfigurierbar.
int verzoegerung = 1000;

String raumNummer = "B11"; //Konfigurierbar
String status = "";
char anzahlAenderung = ' ';

//Globale Messwertaenderungen
int personenZahlGesamt;
int personenZahlRein;
int personenZahlRaus;

//Einrichtung des 4 zeilen lods
#include <Wire.h> // Library for I2C communication
#include <LiquidCrystal_I2C.h> // bibliothek für i2c lcd
LiquidCrystal_I2C lcd = LiquidCrystal_I2C(0x27, 20, 4); //lcd objekt erstellen
//

//SD Bezogen
#include "SPI.h"
#include "SD.h"
int modulSDPin = 5;
File myFile;
String Dateiname = "zaehl3.txt"; //konfigurierbar
//
```

Vorschaubild. Siehe „Dateien/Programm/Konfigurierbare Variablen.png“ sowie „Dateien/Programm/Personenzähler/Personenzaehler.ino“ im digitalen Anhang.

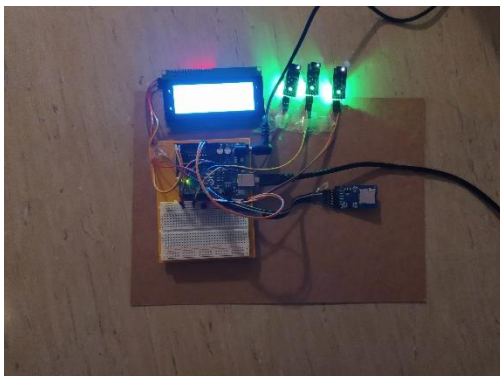
Bei den farblich markierten Feldern handelt es sich um durch den Benutzer konfigurierbare Variablen.

Die rot markierte Variable, maximaleAnzahlPersonen, dient zur Festlegung der maximal erlaubten Anzahl an Personen.

Die grün markierte Variable, zeitSprung, dient zur Festlegung der Sensibilität des Arduinos.

Die blau markierte Variable, raumNummer, dient zur Festlegung des Raumes, in welchem der Personenzähler aufgebaut worden ist.

Die lila markierte Variable, Dateiname, dient zur Festlegung des Dateinamens mitsamt der Dateiendung, unter welchem die Datei abgespeichert wird. Empfohlen wird, den Dateinamen samt Dateiendung nicht länger als zehn Zeichen zu machen.



Hier ein Beispiel eines fertig aufgebauten

Vorschaubild. Siehe „Dateien/Fotos/Personenzähler.jpg“ für vergrößerte Ansicht

## Hinweise

-Nutzen Sie den Schraubenzieher, um die Potentiometer an den Sensoren sowie dem LCD-Display einzurichten.

Dieser Schritt ist zur Einstellung der Empfindlichkeit der Sensoren sowie des Kontrastes des LCD-Displays notwendig.

-Ziehen Sie den Stecker, um die Stromversorgung zu unterbrechen.

-Sollte die SD-Karte nicht eingebaut sein, so wird eine Fehlermeldung über den LCD-Display ausgegeben.

-Sollten Sie den Personenzähler ohne SD-Karte betreiben wollen, so ist es dennoch nötig, diesen beim Start einzufügen. Nach Ausgabe der Raumnummer, Personenzahl sowie des Status kann die SD-Karte jederzeit entfernt werden.

-Sollte der Potentiometer des Displays nicht korrekt eingestellt sein, so wird dieser nichts ausgegeben.

-Der Potentiometer der Sensoren ist je nach Lichtbestrahlung zu drehen, um durch Sonnenstrahlen verursachte Fehlmesswerte zu minimieren.

## V.II. Rechnungen der verbauten Komponenten

Auflistung der Rechnungen der im Projekt verwendeten Komponenten.

### V.II.I. Arduino

**Zugestellt: 10.03.2021**



**Arduino Starter Kit für Anfänger K040007 [Projektbuch auf Deutsch]**

Verkauf durch: **ARDUINO** | Haben Sie eine Frage zum Produkt? [Frage an den Verkäufer stellen](#)

Zeitraum für Rückgabe endet am 09.04.2021

**EUR 79,87**

**Zustand: Neu**



**Nochmals kaufen**

Vorschaubild. Siehe „Dateien/Rechnungen/Arduino Uno/rechnungAmazonArduino.png“ für vergrößerte Ansicht.

### V.II.II. Infrarotsensoren

**Zugestellt: 26.04.2021**

Die Sendung wurde einem Hausbewohner übergeben.



**AZDelivery 3 x TRCT5000 IR Infrarot Linien Folger Hindernis Vermeidung Modul kompatibel mit Arduino**

Verkauf durch: **AZ-Delivery Vertriebs GmbH** | Haben Sie eine Frage zum Produkt? [Frage an den Verkäufer stellen](#)

Zeitraum für Rückgabe endet am 26.05.2021

**EUR 4,99**

**Zustand: Neu**



**Nochmals kaufen**

Vorschaubild. Siehe „Dateien/Rechnungen/Sensoren/rechnungAmazonSensor.png“ für vergrößerte Ansicht.

### V.II.III. Netzteil und Mikro SD-Karte

**Zugestellt: 26.04.2021**

Die Sendung wurde einem Hausbewohner übergeben.



**Leicke ULL-Netzteil 9V 2A 18W, 5,5 x 2,5mm Stecker, Ladegerät, Für Arduino Systeme, UNO R3 REV 3, Mega 2560 R3, Elegoo UNO R3, IEIK UNO R3 etc.**

Verkauf durch: **LEICKE - Leipzig**

Zeitraum für Rückgabe endet am 26.05.2021

**EUR 11,49**

**Zustand: Neu**



**Nochmals kaufen**



**Intenso Micro SDHC 32GB Class 10 Speicherkarte inkl. SD-Adapter**

Verkauf durch: **Amazon EU S.a.r.L.**

Zeitraum für Rückgabe endet am 26.05.2021

**EUR 4,79**

**Zustand: Neu**



**Nochmals kaufen**

Vorschaubild. Siehe „Dateien/Rechnungen/Netzteil und SD/rechnungAmazonNetzteilSD.png“ für vergrößerte Ansicht. Die SD-Karte wird Preislich nicht mit zum Projekt gezählt da sie nur nebenbei für das Projekt genutzt wird.

## V.II.IV. SD Shield

**Zugestellt: 03.04.2021**

Die Sendung wurde einem Hausbewohner übergeben.



AZDelivery 3 x Set SPI Reader Micro Speicher SD TF Karte Memory Card Shield  
Modul kompatibel mit Arduino inklusive E-Book!

Verkauf durch: [AZ-Delivery Vertriebs GmbH](#) | Haben Sie eine Frage zum Produkt? [Frage an den Verkäufer stellen](#)

Zeitraum für Rückgabe endet am 03.05.2021

EUR 5,29

Zustand: Neu



Nochmals kaufen

VorschauBild. Siehe „Dateien/Rechnungen/SD Shield/rechnungAmazonNetzteilSDShield.png“ für vergrößerte Ansicht.

## V.II.V. LCD-Display



WayinTop 20x4 2004 LCD Display Zeichen Bildschirm mit TWI IIC I2C LCD  
Schnittstelle Adapter für Arduino für Mega 2560 (Blau/2004)

Verkauf durch: [WayinTop](#)

Zeitraum für Rückgabe endet am 03.05.2021

EUR 9,99

Zustand: Neu



Nochmals kaufen

VorschauBild. Siehe „Dateien/Rechnungen/LCD und Taster/rechnungAmazonLCD.png“ für vergrößerte Ansicht.

## V.II.VI. Jumperkabel

**Zugestellt: 22.05.2021**

Die Sendung wurde im Briefkasten hinterlegt



Elegoo Jumper Wire 40x 20cm Female-Female, Male-Female, Male-Male Kabel  
Steckbrücken 28AWG Drahtbrücken für Arduino Raspberry Pi (3er Set)

Verkauf durch: [ELEGOO Official - DE](#)

Rücksendung bis zum 21.06.2021 möglich.

EUR 7,99

Zustand: Neu

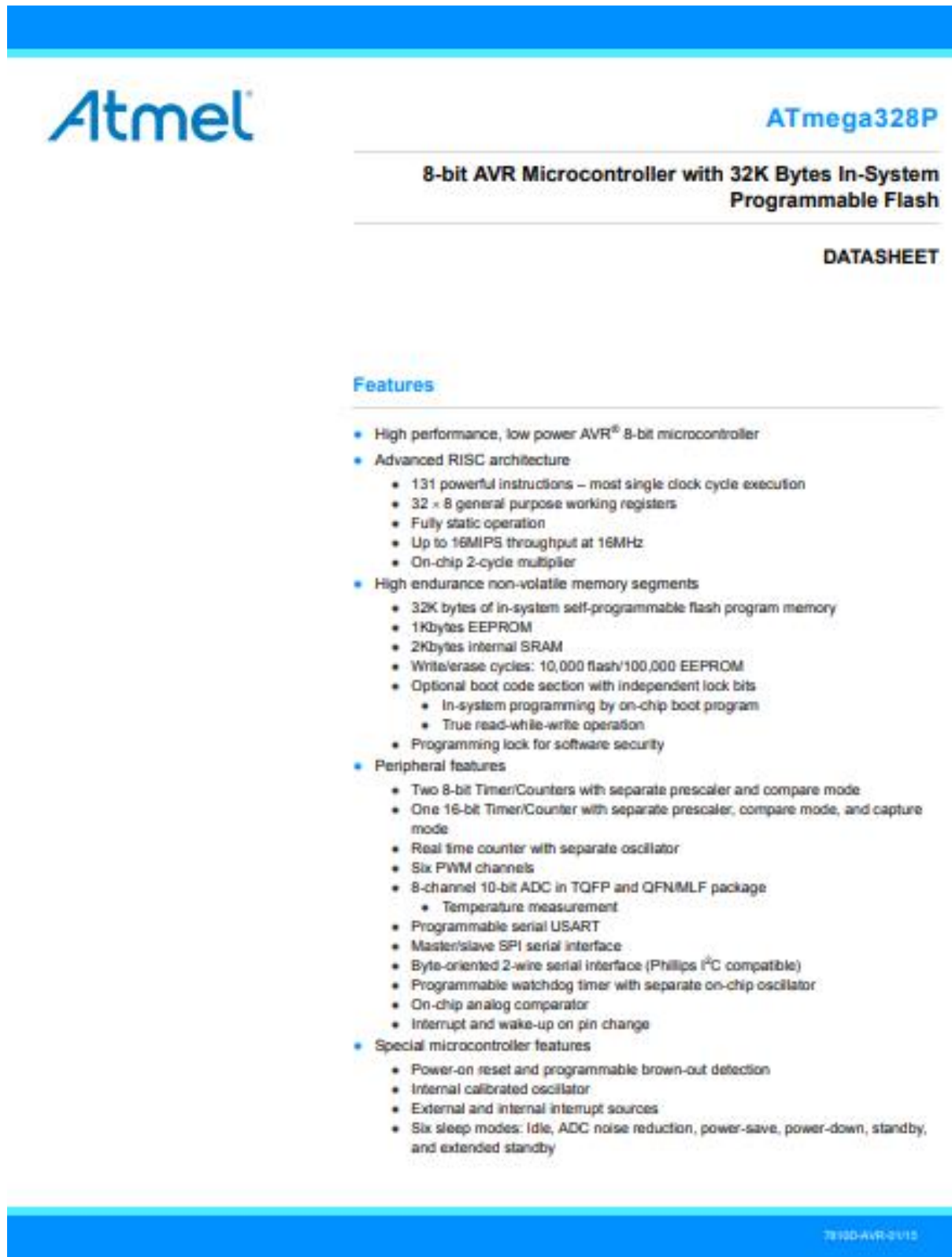


Nochmals kaufen

VorschauBild. Siehe „Dateien/Rechnungen/Jumperkabel/rechnungAmazonJumperkabel.png“ für vergrößerte Ansicht.

## V.III. Datenblätter

### V.III.I. Arduino Uno Rev 3



Vorschau bild. Siehe „Dateien/Datenblätter/Arduino/ATmega328P\_Datasheet.pdf“ für 294 Seiten langes Datenblatt.



## VI.III.II. Infrarotsensor

**Az-Delivery**  
Ihr Experte für Mikroelektronik

# Tracker Sensor TCRT5000 Linienfolger Modul Datenblatt







**Tracker Sensor  
TCRT5000 Linienfolger Modul  
Datenblatt**

**Contents:**

- 1. Features**
- 2. Description**
- 3. Applications**
- 4. Product Summary**
- 5. Absolute Maximum Ratings**
- 6. Basic Characteristics**
- 7. Dimensions**
- 8. Tube Specification Figures**

## 1. Features

- Package type: leaded
- Detector type: phototransistor
- Dimensions (L x W x H in mm): 10.2 x 5.8 x 7
- Peak operating distance: 2.5 mm
- Operating range within  $\pm 20\%$  relative collector current: 0.2 mm to 15 mm
- Typical output current under test:  $I_C = 1\text{ mA}$
- Daylight blocking filter
- Emitter wavelength: 950 nm
- Lead (Pb)-free soldering released
- Compliant to RoHS directive 2002/95/EC and in accordance to WEEE 2002/96/EC



## 2. Description

The TCRT5000 and TCRT5000L are reflective sensors which include an infrared emitter and phototransistor in a leaded package which blocks visible light.

## 3. Applications

- Position sensor for shaft encoder
- Detection of reflective material such as paper, IBM cards, magnetic tapes etc.
- Limit switch for mechanical motions in VCR
- General purpose - wherever the space is limited

## 4. Product Summary

PART NUMBER	DISTANCE FOR MAXIMUM CTR <sub>rel</sub> <sup>(1)</sup> (mm)	DISTANCE RANGE FOR RELATIVE I <sub>out</sub> > 20 % (mm)	TYPICAL OUTPUT CURRENT UNDER TEST <sup>(2)</sup> (mA)	DAYLIGHT BLOCKING FILTER INTEGRATED
TCRT5000	2.5	0.2 to 15	1	Yes
TCRT5000L	2.5	0.2 to 15	1	Yes

### Notes

(1) CTR: current transfere ratio, I<sub>out</sub>/I<sub>in</sub>

(2) Conditions like in table basic charactristics/sensors

## 5. Absolute Maximum Ratings

### Absoulte Maximum Ratings <sup>(1)</sup>

PARAMETER	TEST CONDITION	SYMBOL	VALUE	UNIT
<b>INPUT (EMITTER)</b>				
Reverse voltage		V <sub>in</sub>	5	V
Forward current		I <sub>F</sub>	60	mA
Forward surge current	I <sub>F</sub> ≤ 10 μs	I <sub>FSM</sub>	3	A
Power dissipation	T <sub>amb</sub> ≤ 25 °C	P <sub>F</sub>	100	mW
Junction temperature		T <sub>J</sub>	100	°C
Collector emitter voltage		V <sub>CEO</sub>	70	V
Emitter collector voltage		V <sub>ECO</sub>	5	V
Collector current		I <sub>C</sub>	100	mA
Power dissipation	T <sub>amb</sub> ≤ 55 °C	P <sub>F</sub>	100	mW
Junction temperature		T <sub>J</sub>	100	°C
<b>SENSOR</b>				
Total power dissipation	T <sub>amb</sub> ≤ 25 °C	P <sub>tot</sub>	200	mW
Ambient temperature range		T <sub>amb</sub>	- 25 to + 85	°C
Storage temperature range		T <sub>stg</sub>	- 25 to + 100	°C
Soldering temperature	2 mm from case, t ≤ 10 s	T <sub>sd</sub>	260	°C

### Note

(1) T<sub>amb</sub> = 25 °C, unless otherwise specified

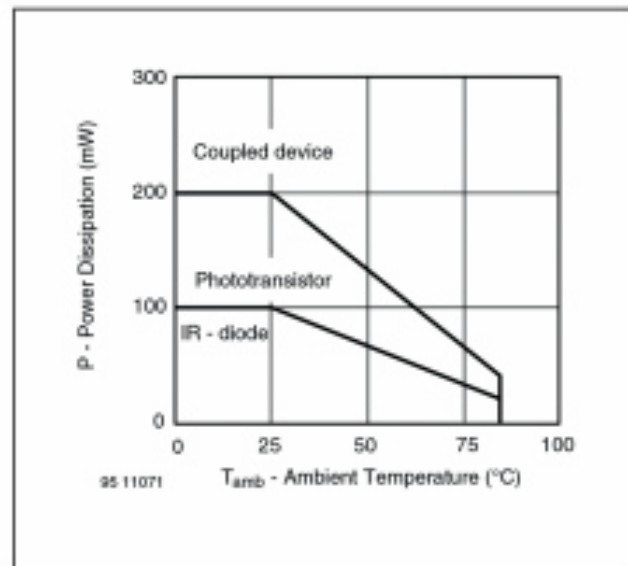


Figure 1 - Power Dissipation Limit vs. Ambient Temperature

## 6. Basic Characteristics

### Basic Characteristics <sup>(1)</sup>

PARAMETER	TEST CONDITION	SYMBOL	MIN.	TYP.	MAX.	UNIT
<b>INPUT (EMITTER)</b>						
Forward voltage	$I_F = 60 \text{ mA}$	$V_F$		1.25	1.5	V
Junction capacitance	$V_D = 0 \text{ V}, f = 1 \text{ MHz}$	$C_j$		17		pF
Radiant intensity	$I_F = 60 \text{ mA}, t_p = 20 \text{ ms}$	$I_a$			21	mW/sr
Peak wavelength	$I_F = 100 \text{ mA}$	$\lambda_p$	940			nm
Virtual source diameter	Method: 63 % encircled energy	$d$		2.1		mm
<b>OUTPUT (DETECTOR)</b>						
Collector emitter voltage	$I_C = 1 \text{ mA}$	$V_{CE0}$	70			V
Emitter collector voltage	$I_E = 100 \text{ }\mu\text{A}$	$V_{EC0}$	7			V
Collector dark current	$V_{CE} = 20 \text{ V}, I_F = 0 \text{ A}, E = 0 \text{ lx}$	$I_{CE0}$		10	200	nA
<b>SENSOR</b>						
Collector current	$V_{CE} = 5 \text{ V}, I_F = 10 \text{ mA}, D = 12 \text{ mm}$	$I_C (2) (3)$	0.5	1	2.1	mA
Collector emitter saturation voltage	$I_F = 10 \text{ mA}, I_C = 0.1 \text{ mA}, D = 12 \text{ mm}$	$V_{CEsat} (2) (3)$			0.4	V

#### Note

- (1)  $T_{amb} = 25^\circ\text{C}$ , unless otherwise specified
- (2) See figure 3
- (3) Test surface: mirror (Mfr. Spindler a. Hoyer, Part No. 340005)

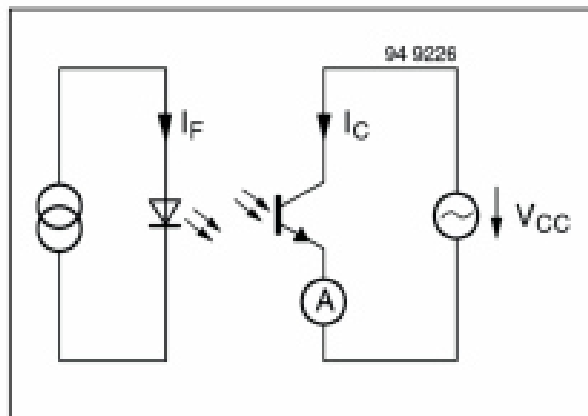


Figure 2 - Test Circuit

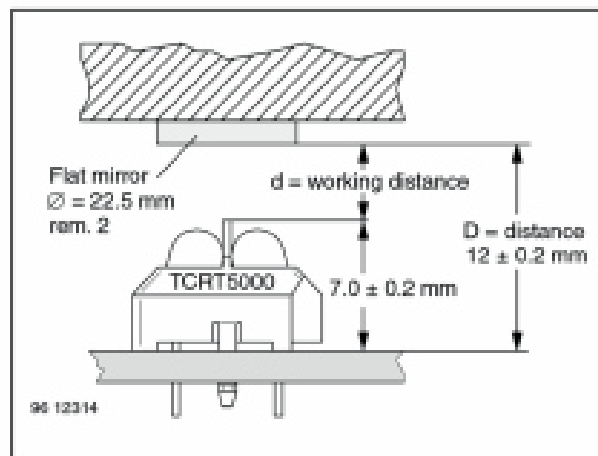


Figure 3 - Test Circuit



# Tracker Sensor TCRT5000 Linienfolger Modul Datenblatt

## Basic Characteristics

$T_{amb} = 25^{\circ}\text{C}$ , unless otherwise specified

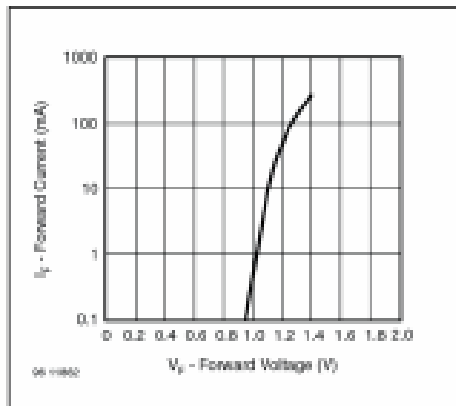


Figure 4 - Forward Current vs. Forward Voltage

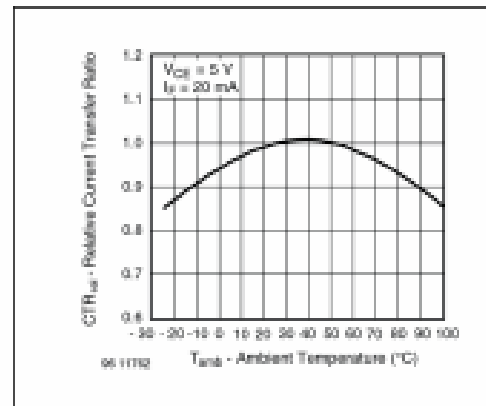


Figure 5 - Relative Current Transfer Ratio vs. Ambient Temperature

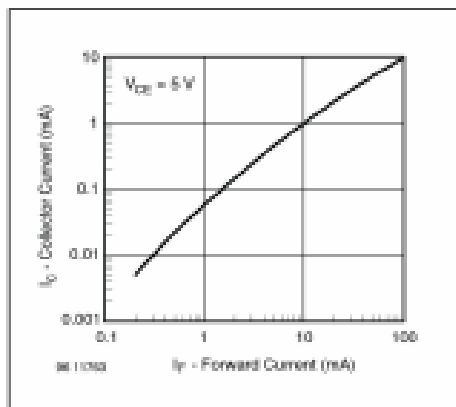


Figure 6 - Collector Current vs. Forward Current

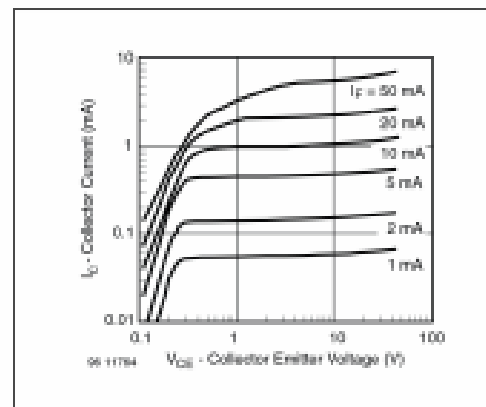


Figure 7 - Collector Emitter Saturation Voltage vs. Collector Current



## Tracker Sensor TCRT5000 Linienfolger Modul Datenblatt

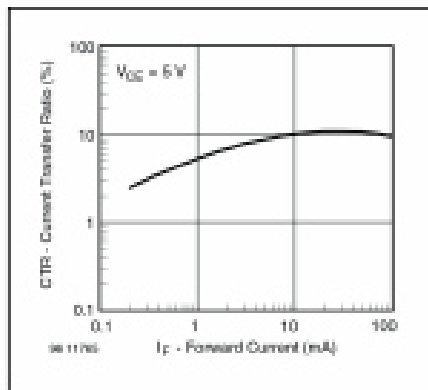


Figure 8 - Current Transfer Ratio vs.  
Forward Current

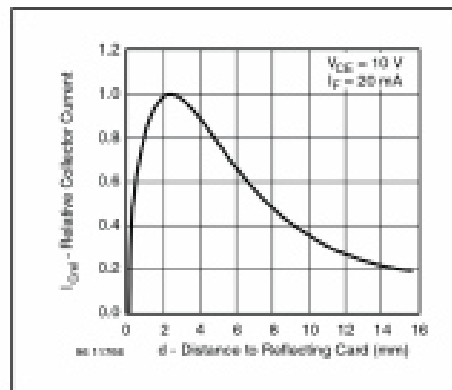
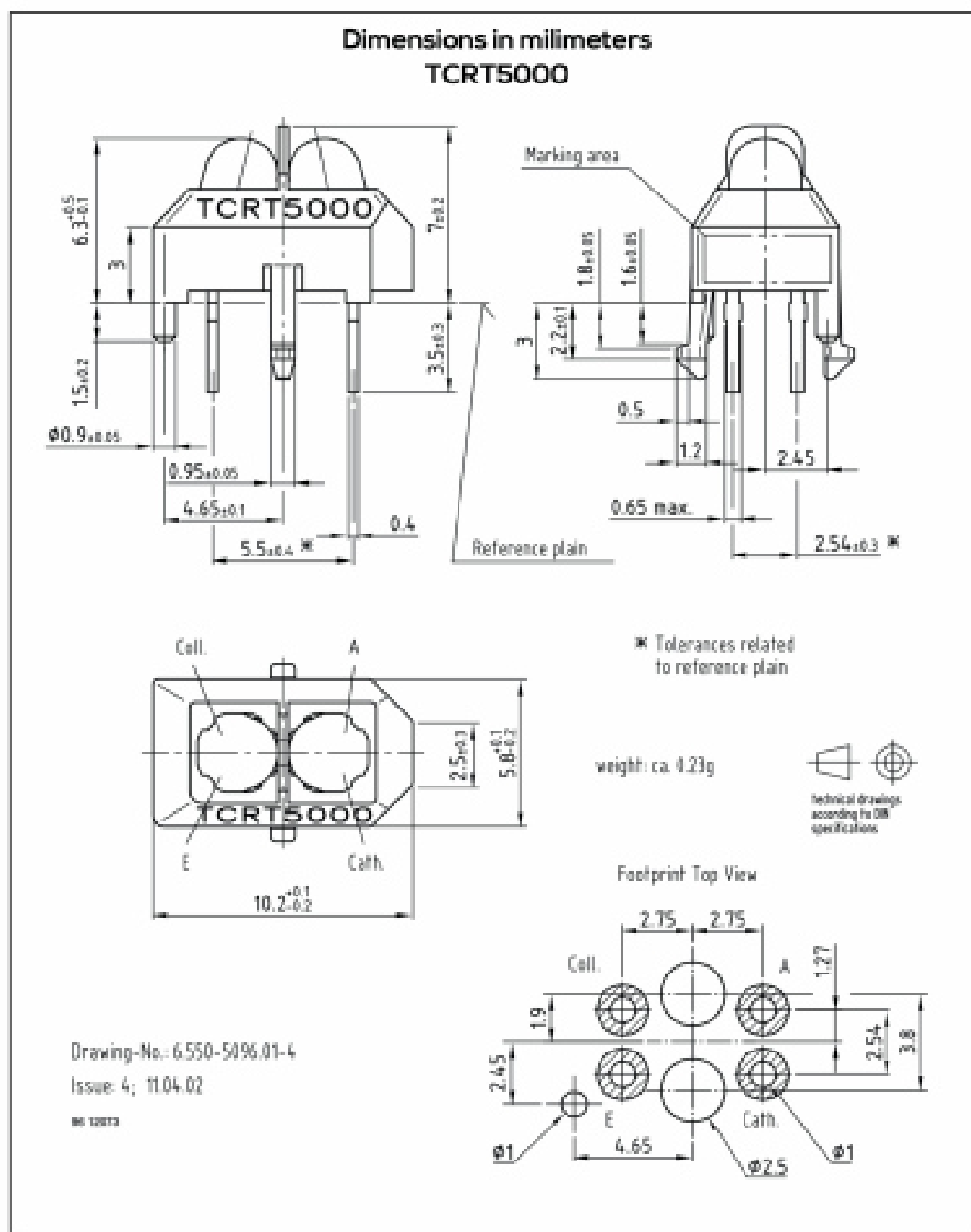
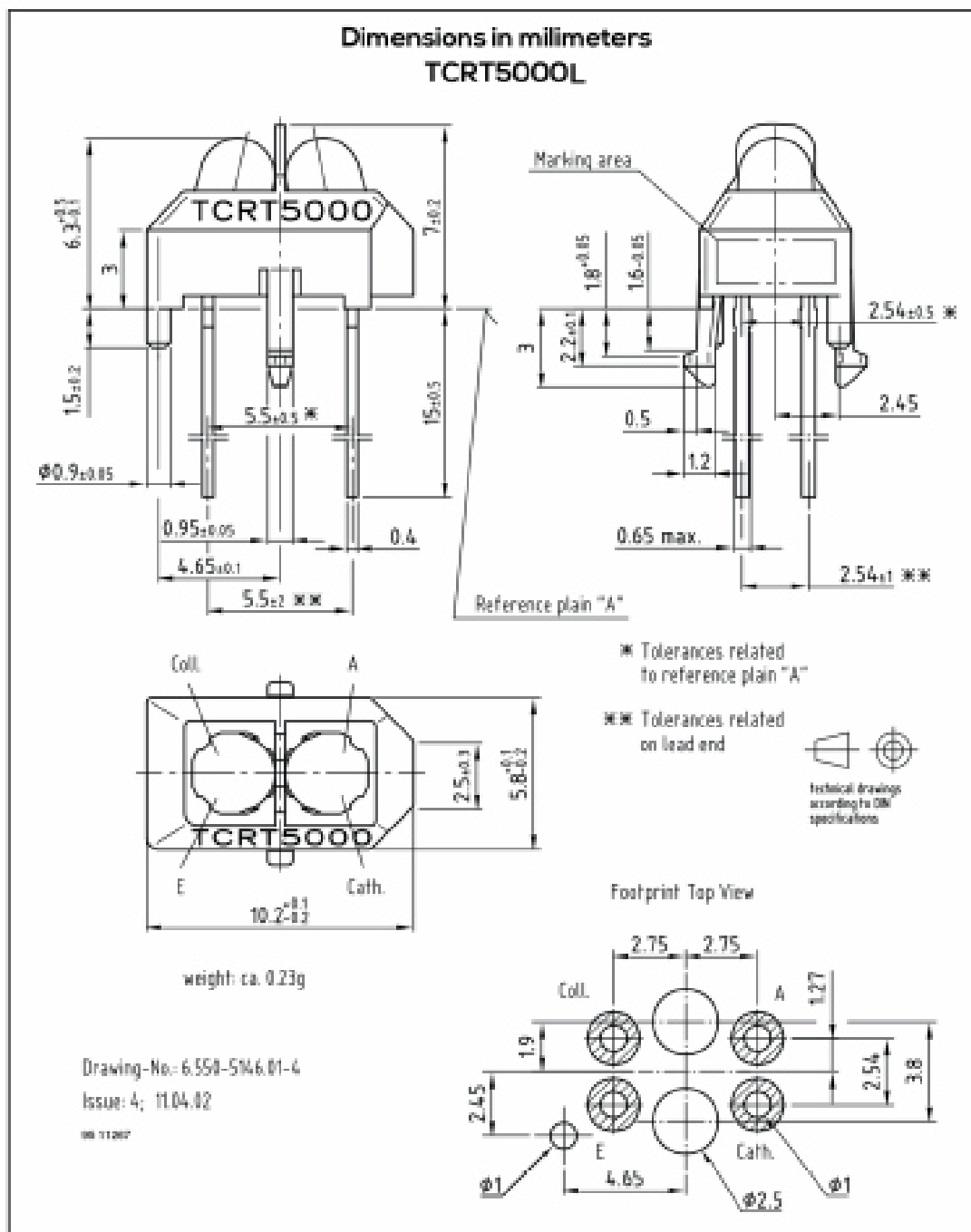


Figure 9 - Relative Collector Current vs.  
Distance

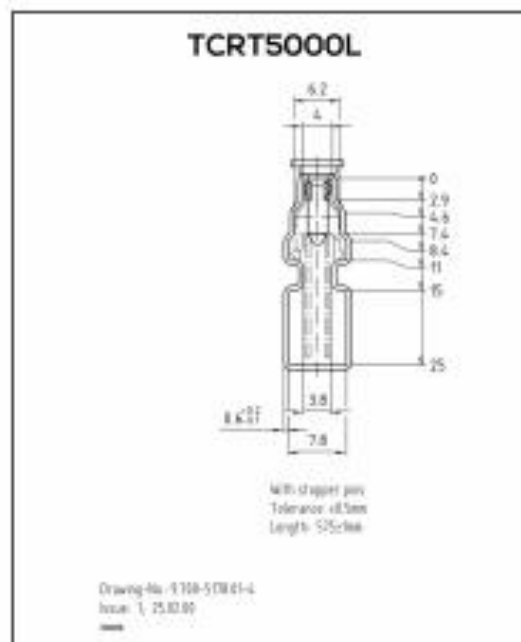
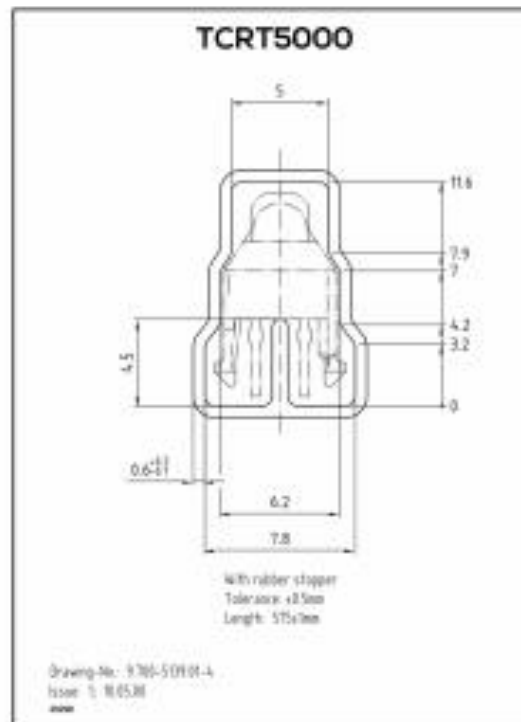
## 7. Dimensions



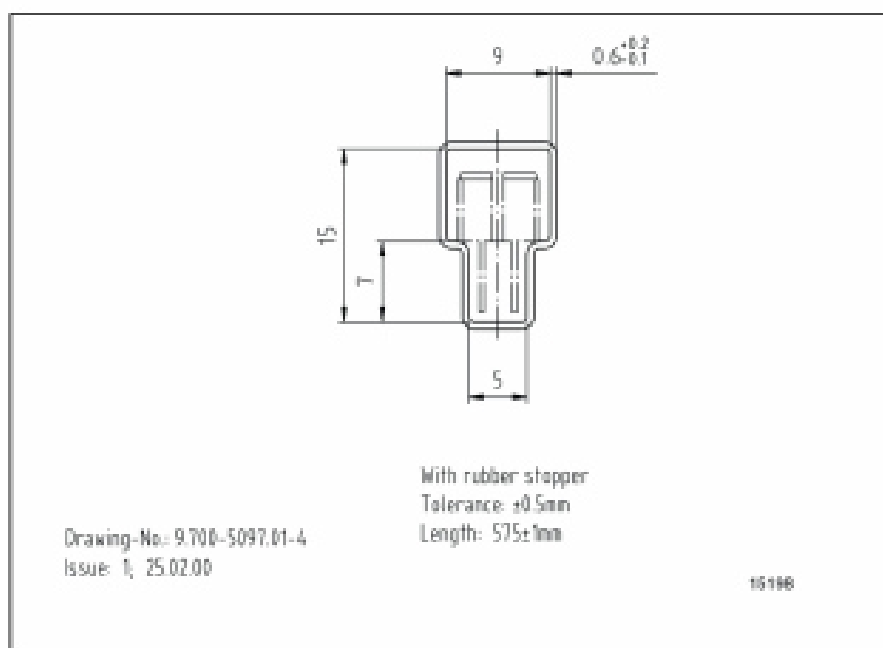




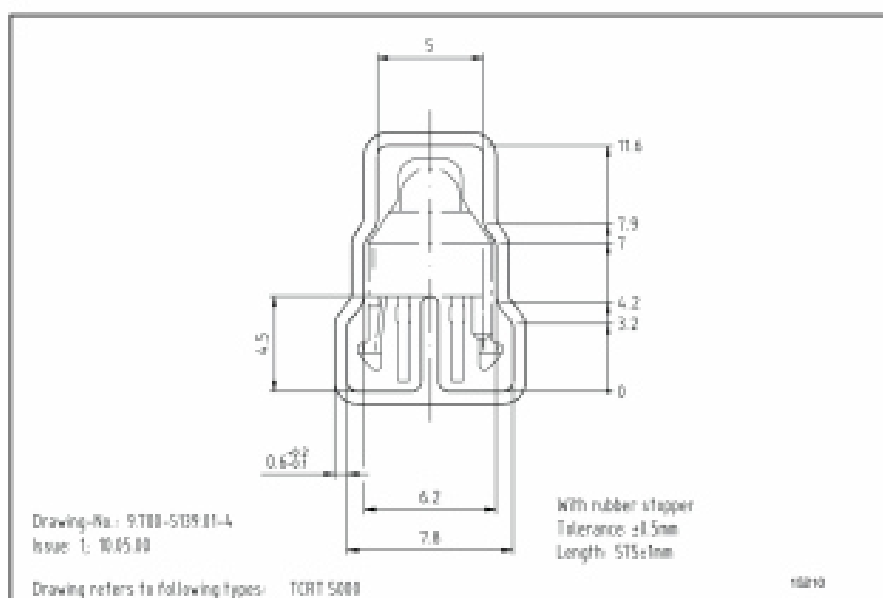
**Tube dimensions in millimeters**



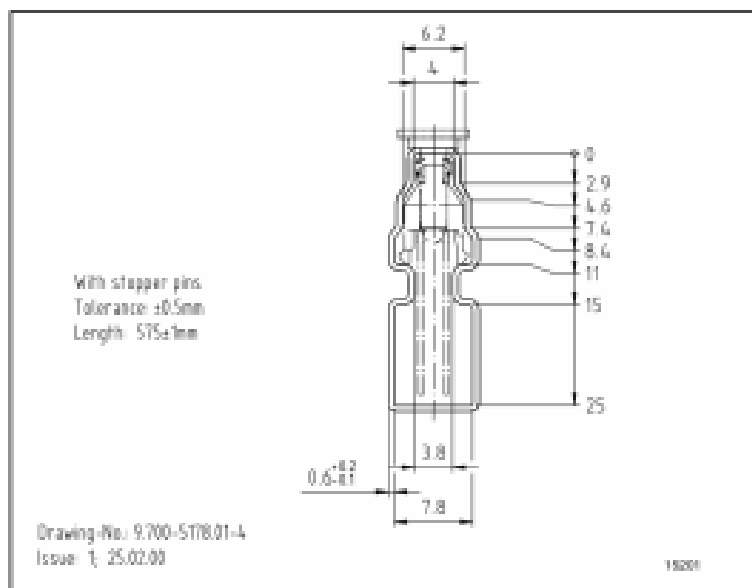
## 8. Tube Specification Figures



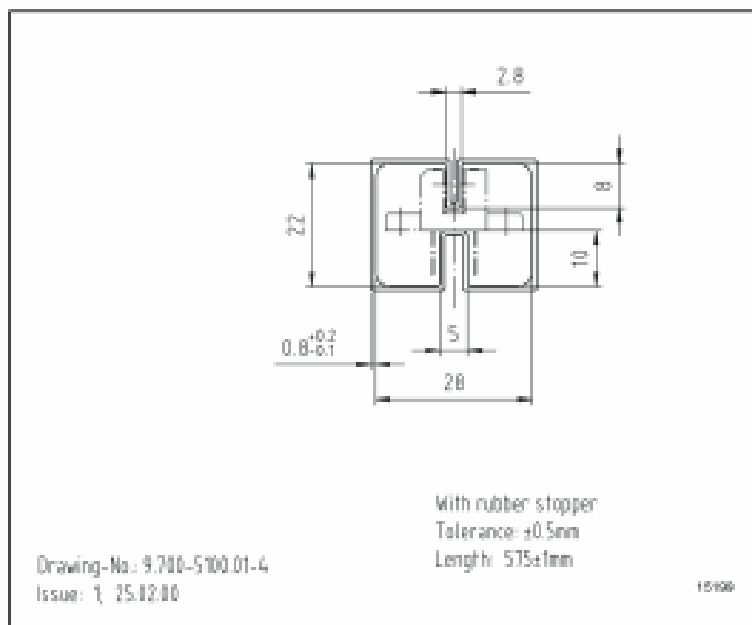
**Tube Specification Figure 1**



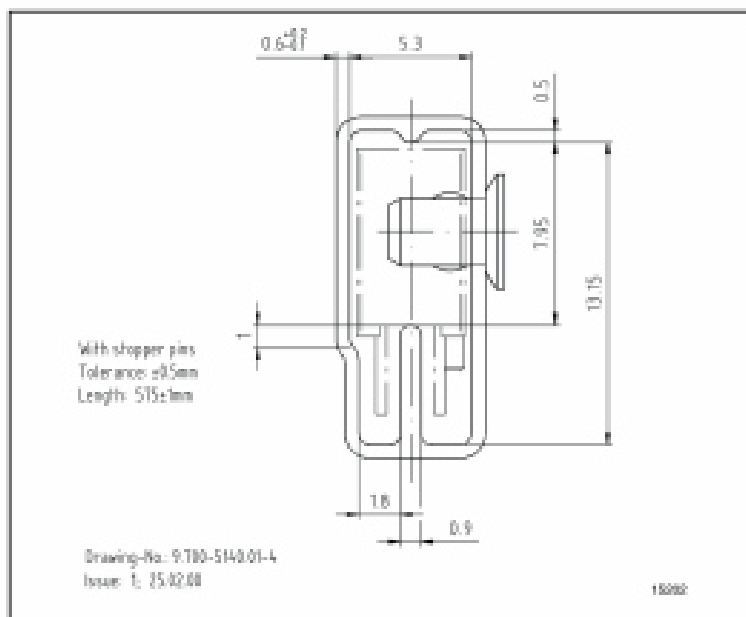
**Tube Specification Figure 2**



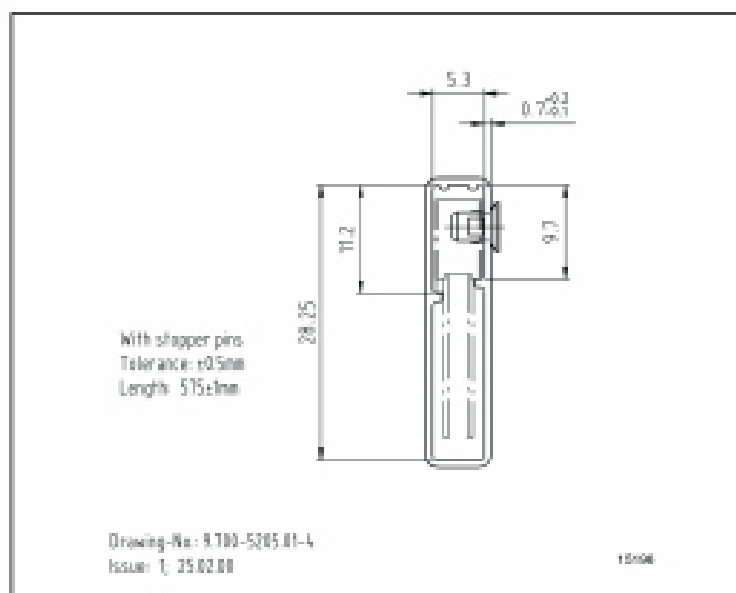
**Tube Specification Figure 3**



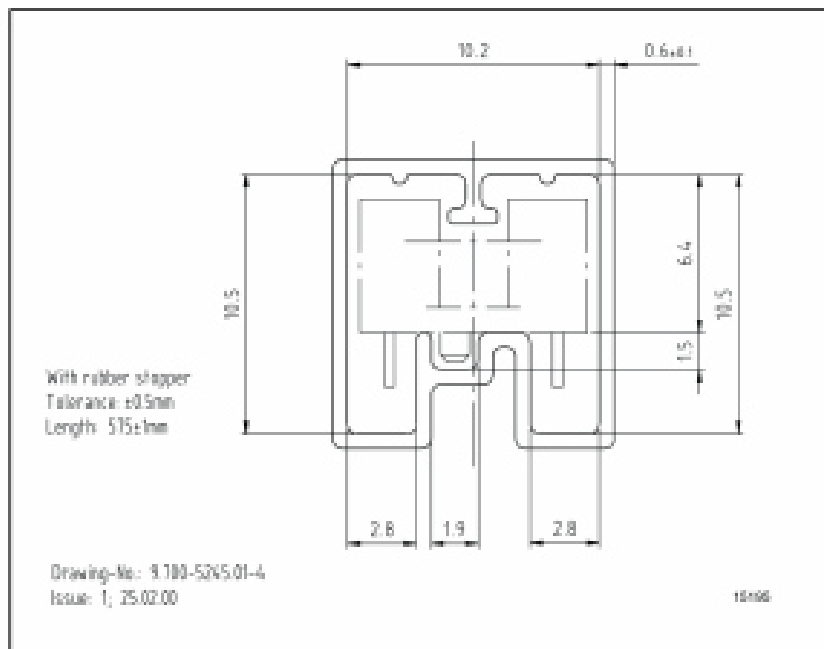
**Tube Specification Figure 4**



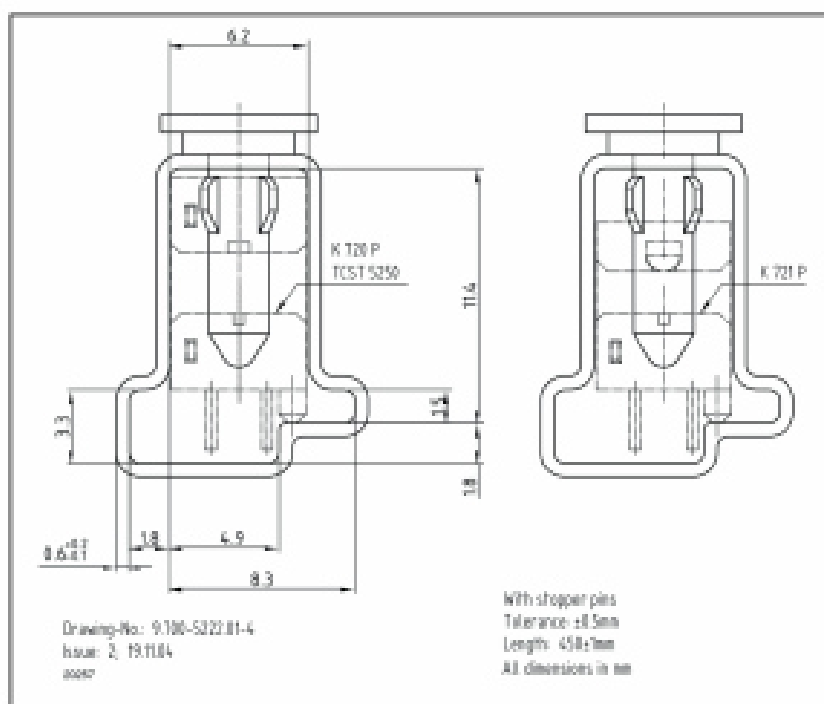
**Tube Specification Figure 5**



**Tube Specification Figure 6**



**Tube Specification Figure 7**



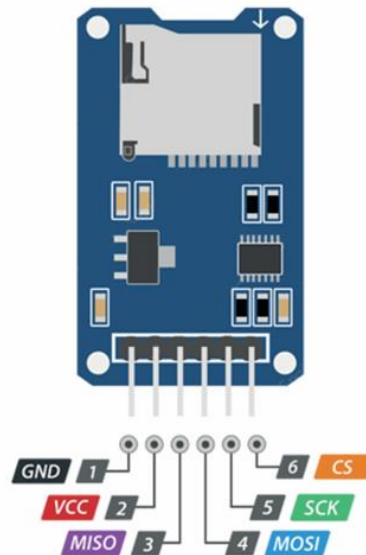
**Tube Specification Figure 8**

### VI.III.III. Micro SD Shield

#### Pinout



SPI Reader Micro Speicherkartenmodul  
Pinout

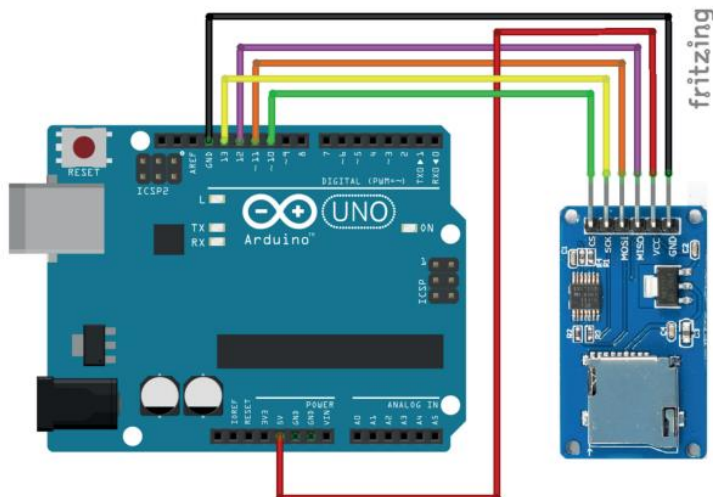


Vorschau. Siehe „Dateien/Datenblätter/SD Modul//SPI Reader Micro Speicherkartenmodul Pinout.pdf“ für Datenblatt.

#### Arduino Schematics



SPI Reader Micro Speicherkartenmodul  
Arduino Schematics



Vorschau. Siehe „Dateien/Datenblätter/SD Modul//SPI Reader Micro Speicherkartenmodul Arduino Schematics.pdf“ für Datenblatt.

## Datenblatt

**Az-Delivery**  
Ihr Experte für Mikroelektronik!

# SPI Reader Micro Speicherkartenmodul Datenblatt







**SPI Reader**  
**Micro Speicherkartenmodul**  
**Datenblatt**

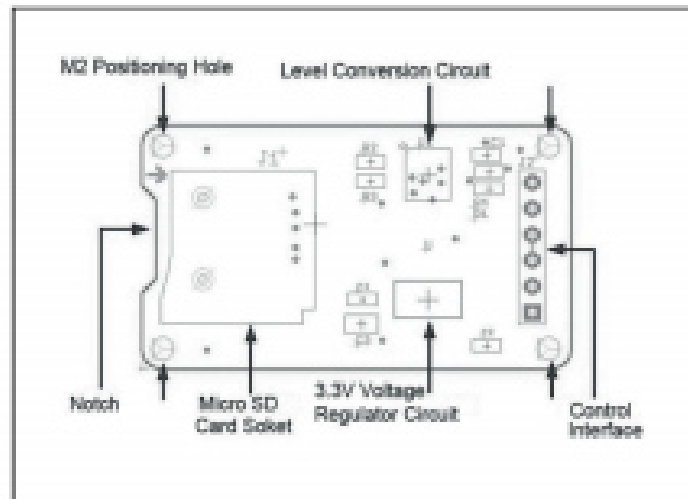
**Contents:**

- 1. Description**
- 2. Interface Parameters**
- 3. Micro SD Card Interface Module**

## 2. Interface Parameters

Items	Min	Typical	Max	Unit
Power Voltage VCC	4.5	5	5.5	V
Current	0.2	50	200	mA
Interface Electrical Potential	3.3 or 5			V
Support Card Type	Micro SD Card(<=2G), <u>Micro</u> SDHC Card(<=32G)			—
Size	42X24X12			mm
Weight	5			g

### 3. Micro SD Card Interface Module

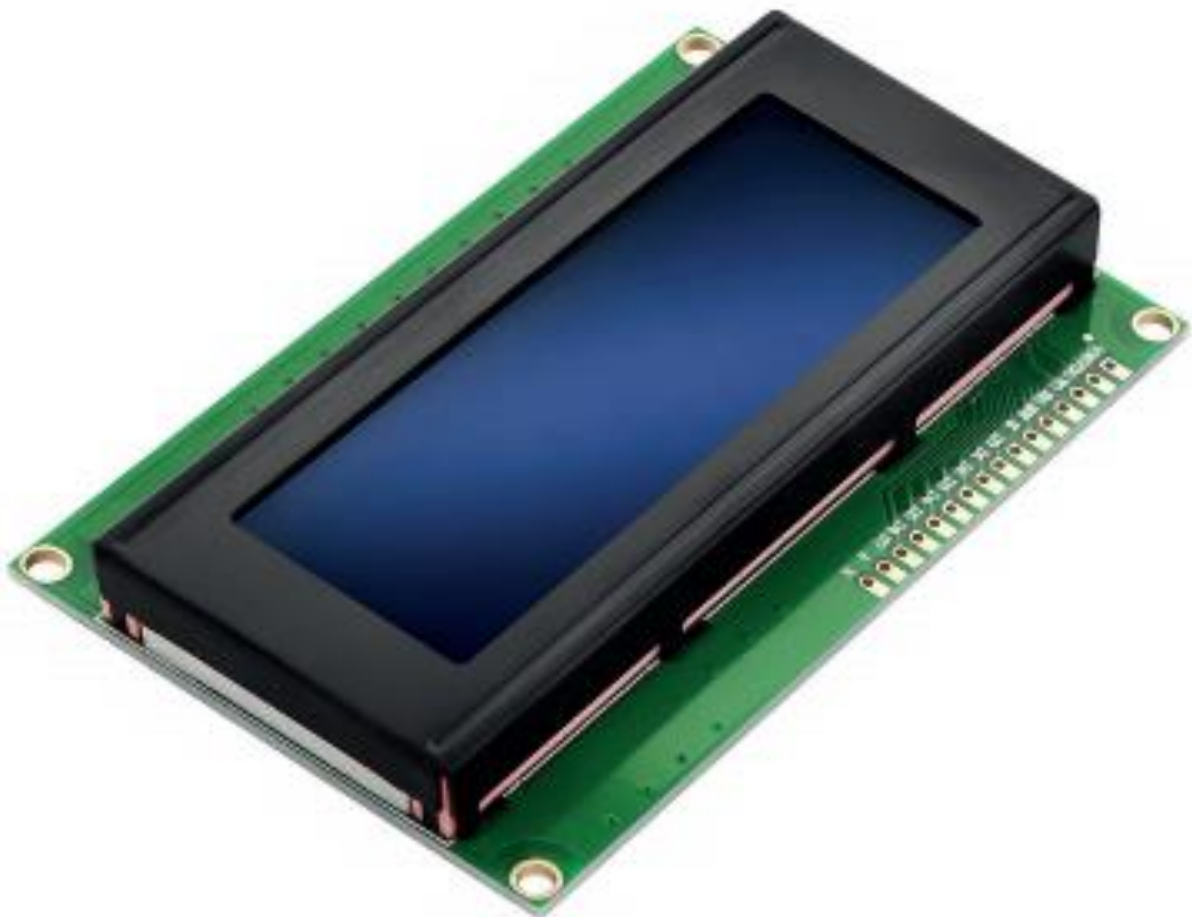


#### VI.III.IV. LCD-Display

##### Datenblatt

**Az-Delivery**  
Ihr Experte für Mikroelektronik!

## HD44780 20x04 Blaues Display Datenblatt



**Content:**

- [1. Basic Specifications](#)
- [2. Absolute Maximum Ratings](#)
- [3. Electrical Characteristics](#)
- [4. Pinout](#)
- [5. Connection Diagrams](#)
- [6. Precautions](#)

## 1. Basic Specifications

- Module dimensions: 98 mm x 60 mm x 14 mm
- Resolution: 20 Characters x 4 Lines
- LCD type: STN, Positive, Transflective, Blue
- Backlight: Blue
- View Angle: 180 degrees
- Modes: Parallel (8-bit and 4-bit)
- Operating Voltage: 3.3V and 5V
- Operating temperature: from -20°C to +70°C
- Storage temperature: from -30°C to +80°C

## 2. Absolute Maximum Ratings

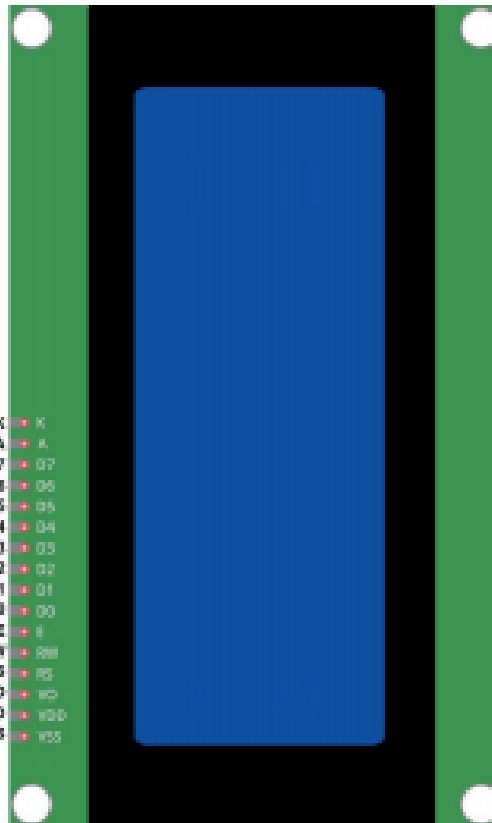
Item	Symbol	Min.	Typ.	Max.	Unit
Supply Voltage for Logic	VDD-VSS	-0.3	-	+7	V
Power Supply for LCD	VLCD	VDD-15	-	VDD+0.3	V
Input Voltage	VIN	-0.3	-	VDD+0.3	V
Supply Current for Backlight	I <sub>LED</sub>	-	-	75	mA

## 3. Electrical Characteristics

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
Power Supply for LCM	VDD-VSS	VDD=5V	4.8	5.0	5.2	V
		VDD=3.3V	3.0	3.3	3.6	V
Input Voltage	V <sub>IL</sub>	L Level	-0.2	-	1	V
	V <sub>IH</sub>	H Level	VDD-1.0	-	VDD	V
LCD Driving Voltage	VDD-V0	-	4.5	4.8	5.1	V
Supply Current for LCM	I <sub>DD</sub>	-	-	-	2000.0	μA
Supply Current for Backlight	I <sub>LED</sub>	-	-	45	-	mA

## 4. Pinout

Backlight cathode (-) - K	→ K
Backlight anode (+) - A	→ A
Data Input/Output Pin 7 - D7	→ D7
Data Input/Output Pin 6 - D6	→ D6
Data Input/Output Pin 5 - D5	→ D5
Data Input/Output Pin 4 - D4	→ D4
Data Input/Output Pin 3 - D3	→ D3
Data Input/Output Pin 2 - D2	→ D2
Data Input/Output Pin 1 - D1	→ D1
Data Input/Output Pin 0 - D0	→ D0
Enable - E	→ E
Read/Write - RW	→ RW
Register Select - RS	→ RS
LCD Contrast Pin - VO	→ VO
Power Supply - VDD	→ VDD
Ground - VSS	→ VSS



The VO pin uses analog voltage to set up the screen contrast level. Hardware contrast adjustment can be done by using voltage divider or potentiometer.

The anode pin A and cathode pin K are positive and negative terminals that are used to control the backlight of the screen.

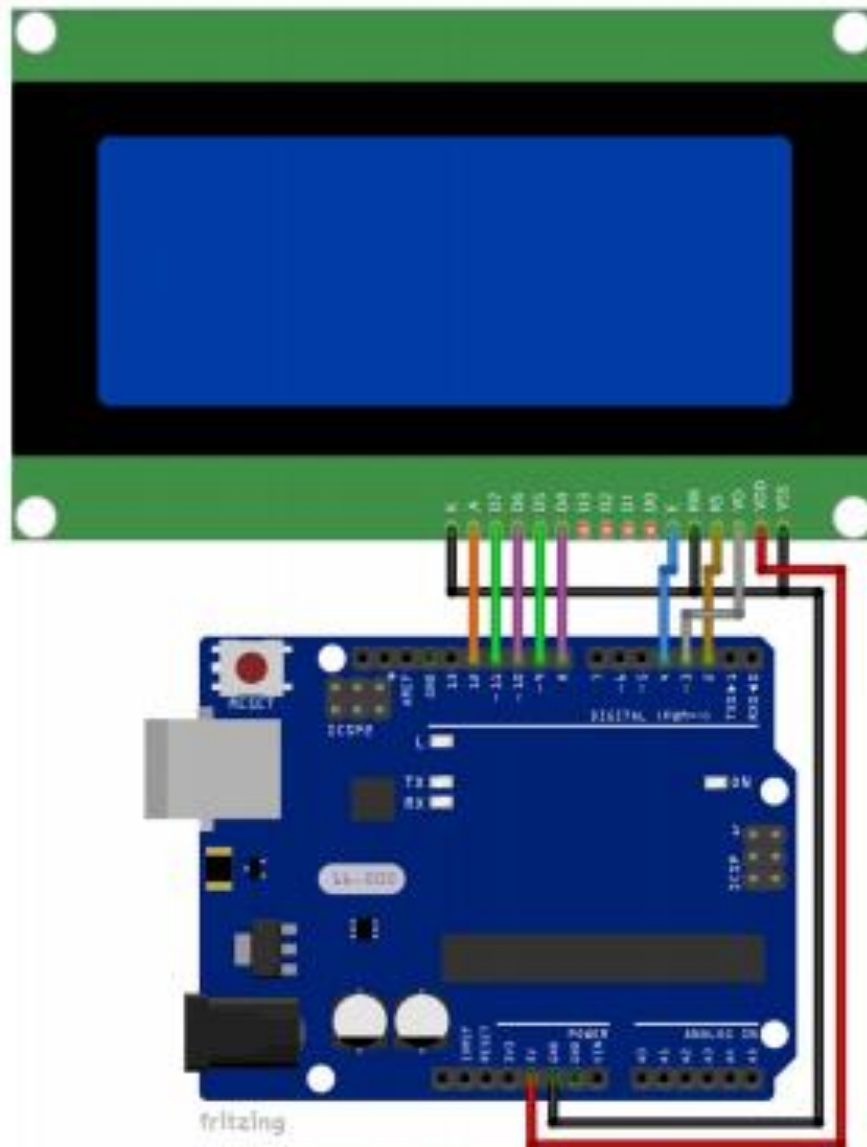
The RW pin, or Read/Write pin, is used to set the data direction, to read data from display-driver chip or to write data into the screen driver chip.

The RS pin is Register Select pin which is used to shift between command or data registers of the driver chip.

The E pin, or Enable pin, is used to enable/disable communication between the main micro-controller and the driver chip of the screen.

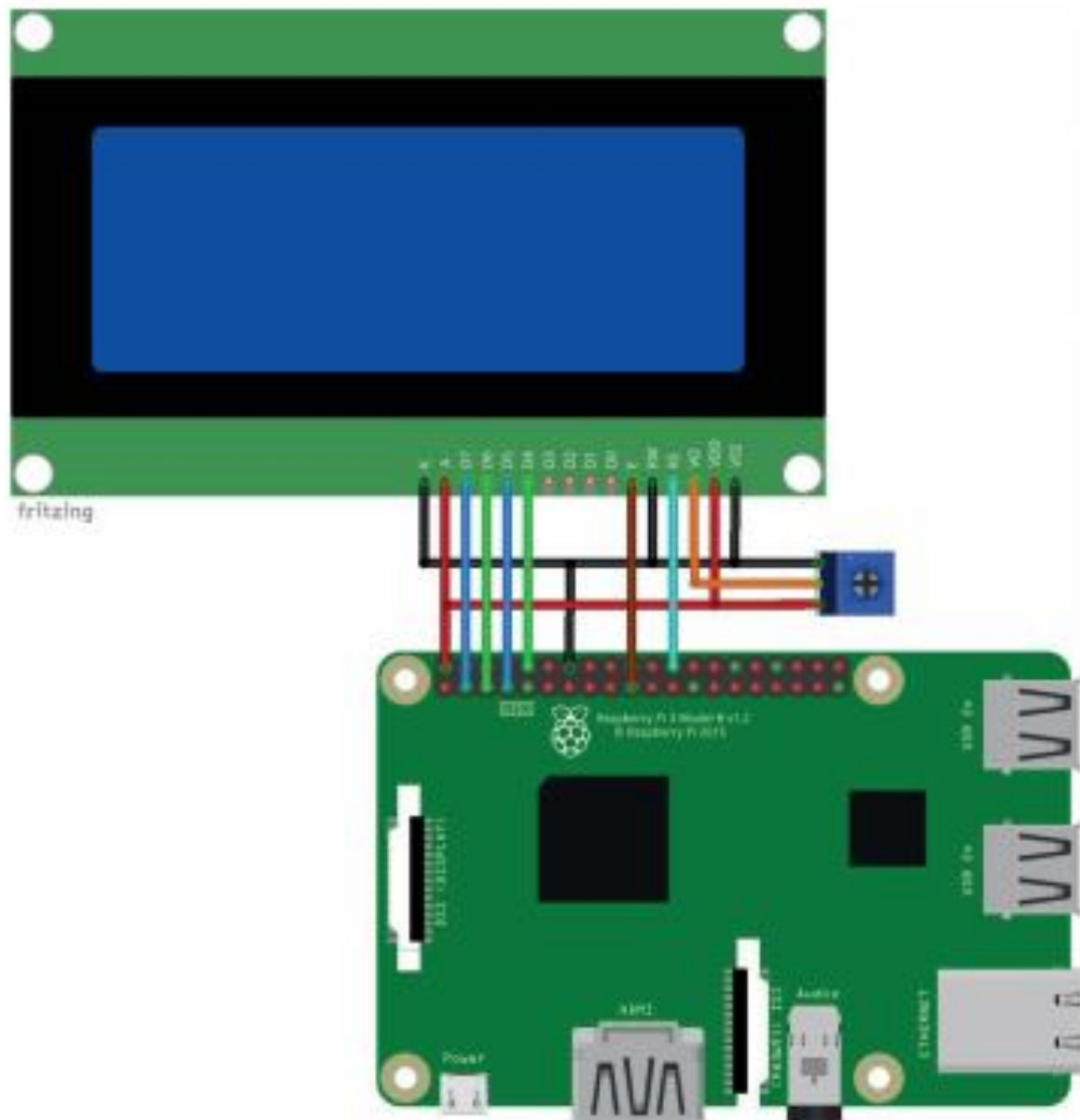
## 5. Connection Diagrams

Connect the HD44780 20x04 LCD Display with the Microcontroller Compatible with Arduino as shown on the following connection diagram:





Connect the HD44780 20x04 LCD Display with the Raspberry Pi as shown on the following connection diagram:



## 6. Precautions

- This device is susceptible to Electro-Static Discharge (ESD) damage. Observe Anti-Static precautions.
  - Do not apply excessive force to display surface or the adjoining areas since this may cause the color tone to vary.
  - The polarizer covering display surface of the LCD module is soft and easily scratched. Handle this polarizer carefully.
  - If display surface becomes contaminated, breathe on the surface and gently wipe it with a soft dry cloth. If it is heavily contaminated, moisten cloth with Isopropyl or alcohol.
  - Solvents other than those above-mentioned may damage the polarizer. Especially, do not use the Water.
  - Exercise care to minimize corrosion of the electrode. Corrosion of the electrodes is accelerated by water droplets, moisture condensation or a current flow in a high-humidity environment.
  - Install LCD Module by using the mounting holes. When mounting the LCD module make sure it is free of twisting, warping and distortion. In particular, do not forcibly pull or bend the cable or the backlight cable.
  - Do not attempt to disassemble or process LCD module.
  - NC terminal should be open. Do not connect anything.
  - If the logic circuit power is off, do not apply the input signals.
  - To prevent destruction of the elements by static electricity, be careful to maintain an optimum work environment.
  - Be sure to ground the body when handling LCD modules.
  - Tools required for assembling, such as soldering irons, must be properly grounded.
- To reduce the amount of static electricity generated, do not conduct assembling and other work under dry conditions.
- The LCD module is coated with a film to protect the display surface. Exercise care when peeling off this protective film since static electricity may be generated.

- Identify and, at all times, observe absolute maximum ratings for both logic and LC drivers.
- Prevent the application of reverse polarity to VDD and VSS, however briefly.
- Use a clean power source free from transients. Power-up conditions are occasionally jolting and may exceed the maximum ratings of these LCD modules.
- The VDD power of LCD module should also supply the power to all devices that may access the display. Do not allow the data bus to be driven when the logic supply to the module is turned off.
- Operate this LCD module within the limits of the modules temperature specifications.
- Surface of the LCD panel should not be touched or scratched. The display front surface is an easily scratched, plastic polarizer.
- Always employ anti-static procedure while handling LCD module.
- Do not store in direct sunlight.
- If leakage of the liquid crystal material should occur, avoid contact with this material, particularly ingestion. If the body or clothing becomes contaminated by the liquid crystal material, wash thoroughly with water and soap.
- When storing the LCD modules, avoid exposure to direct sunlight or to the light of fluorescent lamps.
- Keep LCD modules in bags (avoid high temperature / high humidity and low temperatures below 0°C.
- Liquid crystals solidify under low temperature (below the storage temperature range) leading to defective orientation or the generation of air bubbles (black or white). Air bubbles may also be generated if the module is subject to a low temperature.
- To minimize the performance degradation of the LCD modules resulting from destruction caused by static electricity etc., exercise care to avoid holding the following sections when handling the modules.
  - Exposed area of the printed circuit board.
  - Terminal electrode sections.
- Please keep the temperature within specified range for use and storage. Polarization degradation, bubble generation or polarizer peel-off may occur with high temperature and high humidity.
- Do not touch, push or rub the exposed polarizers with anything harder than an HB pencil lead (glass, tweezers, etc.).

# Az-Delivery

Ihr Experte für Mikroelektronik!

If you are looking for the high quality products for Arduino and Raspberry Pi, AZ-Delivery Vertriebs GmbH is the right company to get them from. You will be provided with numerous application examples, full installation guides, eBooks, libraries and assistance from our technical experts.

Page 9

Vorschau. Siehe „Dateien/Datenblätter/lcd/lcd datenblatt.pdf“ für Datenblatt

## VI. Quelltextdokumentation

### VI.I. Programmcode

---

```
//organisatorisches
int maximaleAnzahlPersonen = 2; //Konfigurierbar
int aktuelleAnzahlPersonen = 0;

float zeitMesser = 0.0;
int zeitSprung = 20; //in ms. Konfigurierbar.
int verzoeigerung = 1000;

String raumNummer = "B11"; //Konfigurierbar
String Status = "";
char anzahlAenderung = ' ';

//Globale Messwertaenderungen
int personenZahlGesamt;
int personenZahlRein;
int personenZahlRaus;

//Einrichtung des 4 zeilen lcds
#include <Wire.h> // Library for I2C communication
#include <LiquidCrystal_I2C.h> // bibliothek für i2c lcd
LiquidCrystal_I2C lcd = LiquidCrystal_I2C(0x27, 20, 4); //lcd objekt erstellen
// _____

//SD Bezogen
#include "SPI.h"
#include "SD.h"
int modulSDPin = 5;
File myFile;
String Dateiname = "zaehl3.txt"; //konfigurierbar
// _____

//Festlegen der sensorpins
int pinSensorRein = 7;
int pinSensorMitte = 8;
int pinSensorRaus = 9;
// _____

void setup() {

//baudrate
Serial.begin(9600); //baudrate festlegen
// _____

//sensoren als eingabe festlegen
pinMode(pinSensorRein, INPUT);
pinMode(pinSensorMitte, INPUT);
pinMode(pinSensorRaus, INPUT);
// _____

//LCD starten
lcd.init();
lcd.backlight();
// _____

//SD einrichtung und erste zeilen mit beschreibung fuellen
pinMode(modulSDPin, OUTPUT);

//wenn nicht an fehlermeldung ausgeben und dauerschleife
if(!SD.begin(modulSDPin))
{
Serial.print("SD Fail");
lcd.setCursor(0,3);
lcd.print("SD Fail");
while(1);
}

//Dateiobjekt starten und füllen
SD.begin(modulSDPin);
myFile = SD.open(Dateiname, FILE_WRITE);
myFile.print("Raumnummer: ");
myFile.print(raumNummer);
myFile.println("");
myFile.println("Zeitstempel [s]; Maximal erlaubte anzahl Personen; anzahl Personen im Raum; Anzahlaenderung; Personenzahl Gesamt; Personenzahl Gesamt raus;
Personenzahl Gesamt rein; Status;");
myFile.close();
}

void loop() {

//LCD ausgabeeinrichtung
//Raumnummer
lcd.setCursor(0, 0);
lcd.print("Raum: ");
lcd.setCursor(10, 0);
lcd.print(raumNummer);
```

```
// _____

//Personenanzahl
lcd.setCursor(0,1);
lcd.print("Personen: ");
lcd.setCursor(10,1);
lcd.print(" ");
lcd.setCursor(10,1);
lcd.print(aktuelleAnzahlPersonen);
lcd.setCursor(12,1);
lcd.print("/");
lcd.setCursor(13,1);
lcd.print(maximaleAnzahlPersonen);
// _____

//Ausgabe Status
lcd.setCursor(0,3);
lcd.print("Status: ");
// _____

//raumfuelle abfrage Sensor
//mitte, falls jemand sich einfach hinstellt
if(digitalRead(pinSensorMitte) !=1)                                //Wenn der mittleresensor ein hindernis hat
{
    while(digitalRead(pinSensorMitte) !=1)                        //solange der mittlere Sensor ein hindernis hat
    {
        lcd.setCursor(0,2);
        lcd.print("beweg dich");
        zeitMesserErhoehung();
        Status = "Durchgang wird blockiert";
        anzahlAenderung = '0';
        Speichern();
        delay(zeitSprung);
    }

    lcd.setCursor(0,2);
    lcd.print(" ");
    Status = "Durchgang wieder frei";
    anzahlAenderung = '0';
    Speichern();
}

//raumfuelle abfrage
//wenn mehr im raum sind als erlaubt
if(aktuelleAnzahlPersonen>=maximaleAnzahlPersonen)
{
    Status = "voll";
}

//wenn weniger im raum sind als erlaubt
if(aktuelleAnzahlPersonen<maximaleAnzahlPersonen)
{
    Status = "nicht voll";
}
// _____

//Status ausgeben
lcd.setCursor(10,3);
lcd.print(" ");
lcd.setCursor(10,3);
lcd.print(Status);
// _____

//aussen
if(digitalRead(pinSensorRaus) !=1) //wenn der aussensensor ausgeloeset wurde beginne prozess zur personenzahlreduzierung
{
    int fertig = 0; //schleifenbeender
    while(fertig == 0)
    {
        if(digitalRead(pinSensorMitte) != 1) //wenn bestaetigung mittlerer Sensor
        {
            fertig = 1;
            if(fertig == 1)
            {
                personenZahlGesamt++;
                if(aktuelleAnzahlPersonen>0)
                {
                    personenZahlRaus++;
                    aktuelleAnzahlPersonen--;
                    anzahlAenderung = '-';
                    Speichern();
                }
            }
        }
        if(digitalRead(pinSensorRein) != 1)//wenn statt dem mittleren sensor der reingehsensor betaetigt wird
        {
            fertig = 1;
        }
        delay(zeitSprung);
        zeitMesserErhoehung();
    }
}
```

```
//innen
if(digitalRead(pinSensorRein) !=1)    //wenn der innensensor ausgeloeset wurde beginne prozess zur personenzahlerhoehung
{
    int fertig = 0;                //schleifenbeender
    while(fertig == 0)
    {
        if(digitalRead(pinSensorMitte) != 1) //wenn bestaetigung mittlerer Sensor
        {
            fertig = 1;
            if(fertig == 1)
            {
                personenZahlGesamt++;
                personenZahlRein++;
                aktuelleAnzahlPersonen++;
                anzahlAenderung = '+';
                Speichern();
            }
        }
        if(digitalRead(pinSensorRaus) != 1)//wenn statt dem mittleren sensor der reingehsensor betaetigt wird
        {
            fertig = 1;
        }
        delay(zeitSprung);
        zeitMesserErhoehung();
    }
}
//_____

delay(zeitSprung);
zeitMesserErhoehung();
}
void zeitMesserErhoehung() //Funktion um zeitMesser um zeitSprung Sekunden erhoeuen
{
    zeitMesser = zeitMesser+((float)zeitSprung/1000);//zeitMesser um zeitSprung Sekunden erhoeuen
}

void Speichern()                //Speicherfunktion
{
    myFile = SD.open(Dateiname, FILE_WRITE); //SD objekt zuweisen
    myFile.print(zeitMesser, 3);
    myFile.print(";");
    myFile.print(maximaleAnzahlPersonen);
    myFile.print(";");
    myFile.print(aktuelleAnzahlPersonen);
    myFile.print(";");
    myFile.print(anzahlAenderung);
    myFile.print(";");
    myFile.print(personenZahlGesamt);
    myFile.print(";");
    myFile.print(personenZahlRaus);
    myFile.print(";");
    myFile.print(personenZahlRein);
    myFile.print(";");
    myFile.print(Status);
    myFile.println(";");
    myFile.close();
}
```

---

Vorschau. Siehe „Dateien/Programm/Personenzaehler/Personenzaehler.ino“ im digitalen Anhang für vollen Programmcode.

## VI.II. Quellen

1. BBS Verden  
<https://www.bbsverden.de/>
2. Personenzähler  
<https://de.wikipedia.org/wiki/Handz%C3%A4hler>
3. Datensatz  
<https://de.wikipedia.org/wiki/Datensatz>
4. Sensorik  
[https://de.wikipedia.org/wiki/Sensorik\\_\(Technik\)](https://de.wikipedia.org/wiki/Sensorik_(Technik))
5. Corona  
<https://de.wikipedia.org/wiki/COVID-19>
6. Präposition  
<https://de.wikipedia.org/wiki/Pr%C3%A4position>
7. Budget  
<https://de.wikipedia.org/wiki/Budget>
8. Integriert  
[https://de.wikipedia.org/wiki/Integration\\_\(Technik\)](https://de.wikipedia.org/wiki/Integration_(Technik))
9. delay  
[https://de.wikipedia.org/wiki/Verz%C3%B6gerung\\_\(Telekommunikation\)](https://de.wikipedia.org/wiki/Verz%C3%B6gerung_(Telekommunikation))
10. RTC  
<https://de.wikipedia.org/wiki/Echtzeituhr>
11. CSV  
[https://de.wikipedia.org/wiki/CSV\\_\(Dateiformat\)](https://de.wikipedia.org/wiki/CSV_(Dateiformat))
12. XLSX  
[https://en.wikipedia.org/wiki/Office\\_Open\\_XML](https://en.wikipedia.org/wiki/Office_Open_XML)
13. I2C
13. LCD  
<https://de.wikipedia.org/wiki/FI%C3%BCssigkristallanzeige>
14. int  
[https://de.wikipedia.org/wiki/Integer\\_\(Datentyp\)](https://de.wikipedia.org/wiki/Integer_(Datentyp))
15. float  
<https://de.wikipedia.org/wiki/Gleitkommazahl>
16. string  
<https://de.wikipedia.org/wiki/Zeichenkette>
17. char  
[https://de.wikipedia.org/wiki/Char\\_\(Datentyp\)](https://de.wikipedia.org/wiki/Char_(Datentyp))
18. file  
<https://www.arduino.cc/en/reference/SD>
19. void  
[https://de.wikipedia.org/wiki/Void\\_\(Schl%C3%BCsselwort\)](https://de.wikipedia.org/wiki/Void_(Schl%C3%BCsselwort))
20. Struktogramm  
<https://de.wikipedia.org/wiki/Nassi-Shneiderman-Diagramm>
21. nsd  
<https://help.structorizer.fisch.lu/index.php?menu=69>
22. Infrarot  
<https://de.wikipedia.org/wiki/Infrarotstrahlung>
- Fritzing LCD  
<https://forum.fritzing.org/uploads/default/original/2X/3/396ac45a9caa834358c9439a959ef2a6203c7a2d.fzip>
- Fritzing Infrarotsensor



<https://forum.fritzing.org/uploads/default/original/2X/3/3198993fad58be49eec4acad545dfe3d66d4beac.fzpz>

Arduino Uno

<https://www.amazon.de/gp/product/B01ELNJAUC/>

LCD

<https://www.amazon.com/WayinTop-Display-Interface-Adapter-Arduino/dp/B07TXGD3WS>

SD Modul

<https://www.amazon.de/AZDelivery-Reader-Speicher-Memory-Arduino/dp/B077MCQS9P/>

Kabel

<https://www.amazon.de/gp/product/B01EV70C78/>

Sensor

<https://www.amazon.de/gp/product/B07DRCKV3X/>

Fritzing

<https://fritzing.org/download/>

Structorizer

[https://www.chip.de/downloads/Structorizer\\_64884440.html](https://www.chip.de/downloads/Structorizer_64884440.html)

Word Suite

[www.microsoft.com](http://www.microsoft.com)

Arduino IDE

<https://www.arduino.cc/en/software>

Wire.h

<https://www.arduino.cc/en/Reference/Wire>

LiquidCrystal\_I2C.h

<https://www.arduino.cc/reference/en/libraries/liquidcrystal-i2c/>

SD.h

<https://www.arduino.cc/en/reference/SD>

SPI.h

<https://www.arduino.cc/en/reference/SPI>

Vorschau. Siehe "Dateien/Quellen/Quellenverzeichnis.docx" für vergrößerte Ansicht.