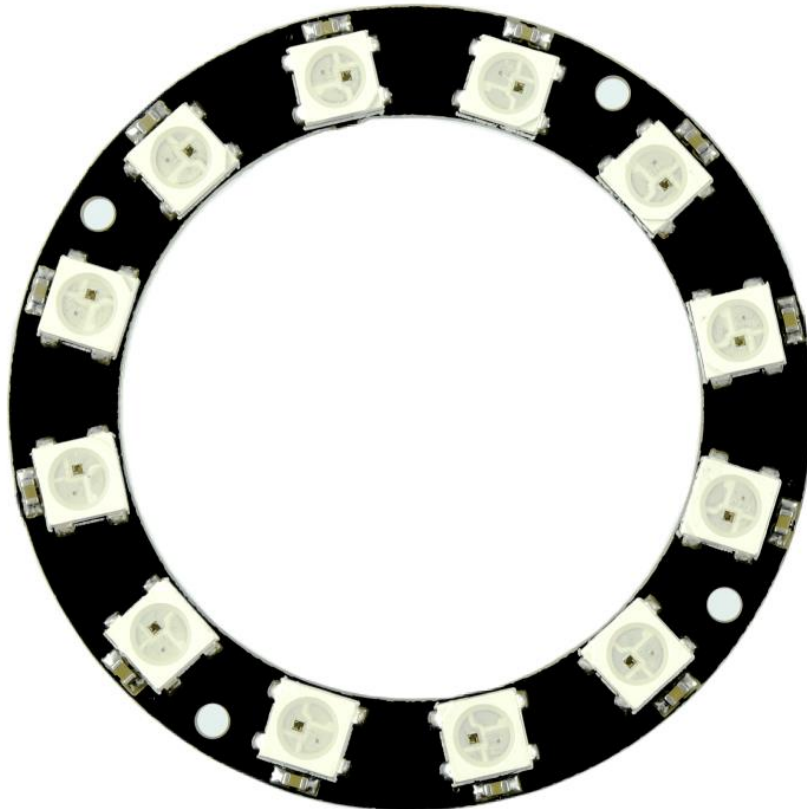


Ring Uhr

Labor Mikrocontroller der Studiengänge
Elektrotechnik/Informationstechnik
Technische Informatik
Mechatronik
Medizintechnik



1 Die Zeiteinheit

Es steht eine externe Echtzeituhr zur Verfügung. Sie stellt Sekunden, Minuten und auch Stunden bereit.

1.1 Funktionen

Die Initialisierung der Zeiteinheit wird bereits bei der Initialisierung des Boards vorgenommen:

```
void setup()
```

Zeitanzeige in der angegebenen Zeile und Spalten

```
TIME_print(Zeile, Spalte);
```

Die nachfolgenden Variablen sind bereits deklariert

```
uint8_t Second;  
uint8_t Minute;  
uint8_t Hour;
```

Zeiten einlesen

```
Minute=Time_getMinute();  
Hour=Time_getHour();
```

Zeiten verändern

```
Minute++; //Minute=Minute+1  
Hour++; //Hour=Hour+1
```

Zeiten setzen

```
Time_setMinute(Minute);  
Time_setHour(Hour);
```

Minuten um 1 erhöhen

Hier wird automatisch auch die Sekunde auf 0 gesetzt.

```
Time_incrementMinute();
```

Stunden um 1 erhöhen

```
Time_incrementHour();
```

1.2 Beispiel

Die Uhrzeit wird ausgegeben und die Minuten können mit der Taste SW2 gestellt werden.

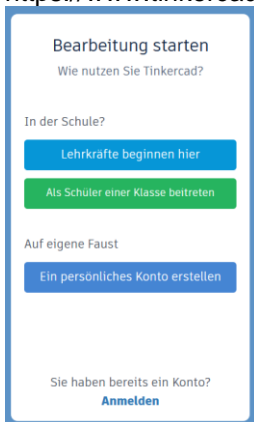
```
//Minuten mit Taster SW2 stellen  
if(SW2_get()==1)  
{  
    Time_incrementMinute(); //Minute erhoehen  
    Time_print(1,0); //Zeit anzeigen  
    DisplayLEDS=1; //Ausgabeanforderung(Flag) fuer LEDs setzen  
}
```

1.3 Ring Uhr mit tinkercad.com testen und weiter entwickeln

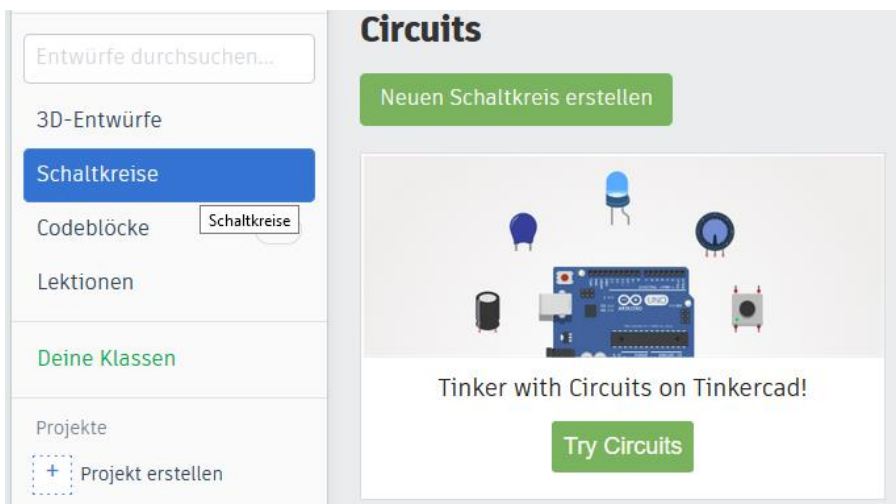
Es gibt zwei Möglichkeiten mit tinkercad zu arbeiten.
Ihr erstellt ein persönliches Konto oder ihr tretet einer Klasse bei. Dies muss die Schule vorgeben.

Webseite:

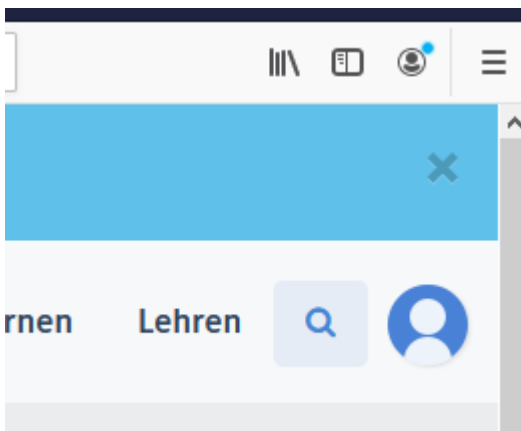
<https://www.tinkercad.com/>



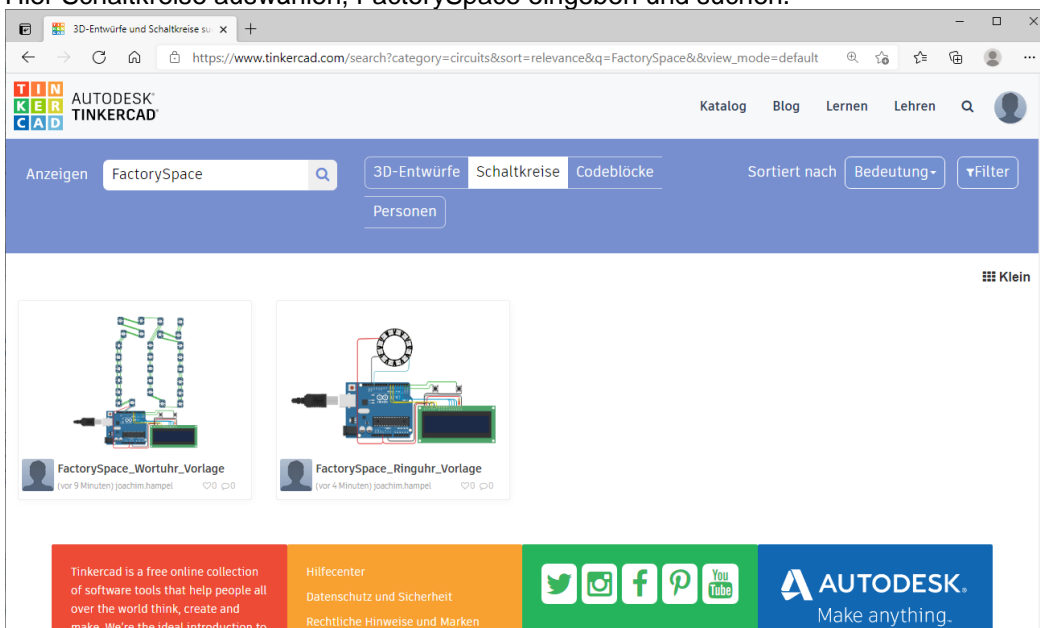
Nach der Anmeldung auf Schaltkreise wechseln.



Dann suchen auswählen



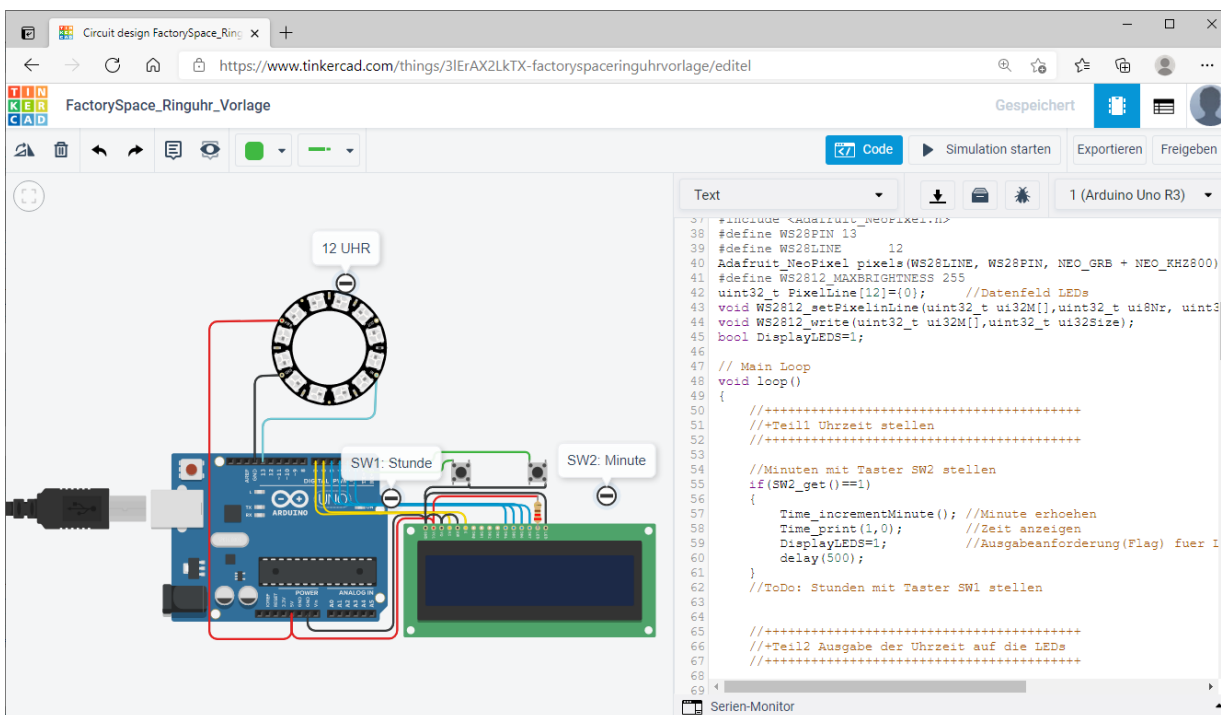
Hier Schaltkreise auswählen, FactorySpace eingeben und suchen.



Das angezeigte Projekt FactorySpace_Ringuhr_Vorlage auswählen, **Kopieren und bearbeiten** anklicken und los geht's. Rechts werden die verfügbaren Komponenten angezeigt.

Mit dem Button **Code** kann der Sourcecode ein- und ausgeblendet werden.

Der Button **Simulation starten** startet und beendet die Simulation. Der Sourcecode kann nur verändert werden, wenn die Simulation nicht gestartet ist.



1.4 Programmieraufgabe Stunde stellen

Das Stellen der Stunden 1-12 soll mit Taster SW1 vorgenommen werden können.

//ToDo: Stunden mit Taster SW1 stellen

2 RGB Leuchtdiode WS2812B

Die WS2812B ist eine „smarte“ RGB Leuchtdiode der Firma Worldsemi. Über eine serielle Datenleitung werden für jede angeschlossene Diode 24Bit übertragen. Es stehen also für jede Farbe 8Bit und damit 256 Farbewerte (0-255) zur Verfügung.



2.1 Farbwerte

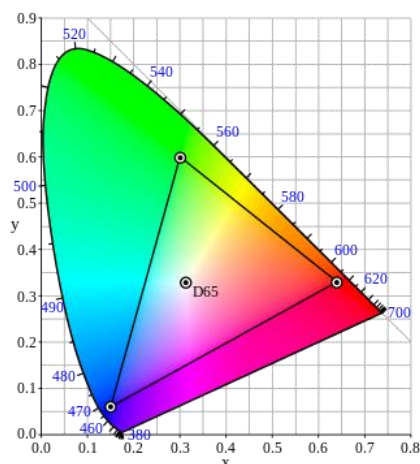
Die Farbwerte für die 27 LEDs werden in einem Datenfeld gespeichert.

```
uint32_t PixelLine[27]={0};
```

Es steht eine Funktion zur Verfügung, mit der die Farbwerte gesetzt werden können.

```
void WS2812_setPixelinLine(uint32_t ui32M[],uint32_t ui8Nr, uint32_t uiR,  
uint32_t uiG, uint32_t uiB );
```

Dieser Funktion wird das zu beschreibende Datenfeld, die Pixelnummer (Diode) und die drei Farbewerte für Rot, Grün und Blau (RGB) übergeben. Die Pixelnummer startet bei 0 und endet mit 26.



Das CIE-1931-xy-Chromatizitätsdiagramm mit sRGB-Farbraum und D65-Weißpunkt
Quelle: Wikipedia

Die Farbwerte, die im Datenfeld gespeichert sind können mit der folgenden Funktion zu den RGB LEDs übertragen werden. Der Funktion wird das Datenfeld und die Länge übergeben.

```
void WS2812_write(uint32_t ui32M[],uint32_t ui32Size);
```

2.2 Beispiel

Pixel 0,1 und 2 werden Farben Rot, Grün und Blau angeschaltet.

```
WS2812_setPixelinLine(PixelLine,0,100,0,0); //Rot 100 von 255
WS2812_setPixelinLine(PixelLine,1,0,100,0); //Gruen 100 von 255
WS2812_setPixelinLine(PixelLine,2,0,0,100); //Blau 100 von 255
WS2812_write(PixelLine,27); //Werte an LEDs uebertragen
```

2.3 Weiter Möglichkeiten die Farben einzustellen

Die Farben können auch direkt für jede LED im Datenfeld gesetzt werden.

Einzelne Farben setzen:

```
PixelLine[0]=100*ROT + 0*GRUEN + 0*BLAU; //LED1
PixelLine[1]=0*ROT + 100*GRUEN + 0*BLAU;
PixelLine[2]=0*ROT + 0*GRUEN + 100*BLAU;
```

Mischfarben setzen:

```
PixelLine[3]=100*ROT + 0*GRUEN + 100*BLAU;
```

Weitere Farbinformation zu einer bestehenden Farbinformation hinzufügen:

Grundfarbe setzen:

```
PixelLine[4]=100*ROT + 0*GRUEN + 0*BLAU;
```

Weiter Farbinformation hinzufügen:

```
PixelLine[4] += 100*GRUEN;
```

3 Programmieraufgaben Ring Uhr LEDs

In der Simulation sind die LEDs wenige Grad nach Links verschoben dies ist bei der Darstellung zu beachten.

3.1 Rahmen

Die Ausgabeanforderung wird gesetzt, wenn eine Minute vergangen ist oder wenn die Uhrzeit mit den Tastern gestellt wird. Der Wert 1 in der Variable *DisplayLEDS* signalisiert, dass sich die Uhrzeit geändert hat und eine Ausgabe nötig ist. Als erste Aktion wird der Wert auf 0 gesetzt, da ja mit der Abarbeitung begonnen wird. Danach werden alle LEDs ausgeschaltet – also alle Farbwerte auf 0 gesetzt.

Eure Aufgabe ist es nun Farbwerte für die zur Anzeige der Uhrzeit benötigten LEDs zu setzen. Ist diese Aufgabe erledigt so werden die Farbwerte an die LEDs übertragen.

```
//+++++
//+Teil2 Ausgabe der Uhrzeit auf die LEDs
//+++++

//Ausgabeanforderung abarbeiten
if(DisplayLEDS==1)
{
    //Anforderung aufheben
    DisplayLEDS=0;

    //Vorbereitung alle LEDES ausschalten
    for(int i=0; i<12; i++)
    {
        WS2812_setPixelinLine(PixelLine,i,0,0,0);
    }

    //+++++
    /* LEDs entsprechend der Uhrzeit setzen
    //+++++

    // T O D O

    //+++++
    /* Farbwerte an LEDs uebertragen
    //+++++
    WS2812_write(PixelLine,12);
}
```

3.2 Test der LED Positionen

Verwenden Sie nun die neue Methode um die Nummern der einzelnen Leds heraus zu finden und fertigen Sie eine Skizze an.

```
PixelLine[0]=100*ROT + 0*GRUEN + 0*BLAU;
PixelLine[0]+=100*GRUEN;
```

3.3 Stunden LEDs anschalten

Hier reicht eine einfache Abfrage der Stunde aus.

```
//To Do: Stunden Woerter anschalten
if(Hour==1)
{
    //TO DO: passendes Wort anschalten
}
```

3.4 Minuten LEDs anschalten

Nun ist ihre Kreativität gefragt. Entwickeln Sie das bestehende Programm weiter um auch die Minuten ausreichend genau auf dem Ring darstellen zu können

Vorgehensweise:

1. Erarbeiten Sie einen Vorschlag wie die 12LEDs zur Darstellung von Stunden und Minuten verwendet werden können.
2. Entwickeln Ihr Programm weiter.
3. Führen Sie gründliche Test der Software durch.

4 Fragen oder Anregungen

Unter folgender E-Mail-Adresse könnt ihr das FactorySpace Team der Hochschule Pforzheim erreichen.

factory-space@hs-pforzheim.de

Meldet euch bitte, wenn ihr Fragen oder auch Verbesserungsvorschläge habt.