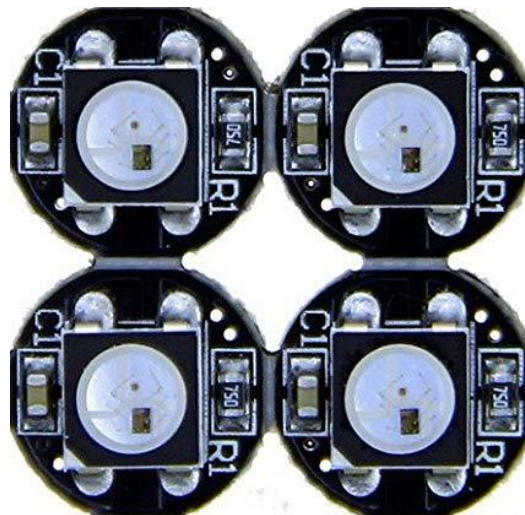


Wort Uhr

Labor Mikrocontroller der Studiengänge
Elektrotechnik/Informationstechnik
Technische Informatik
Mechatronik
Medizintechnik



1 Programmierparadigmen

1.1 Paradigma

Eine grundsätzliche Denkweise.

Ein zusammenhängendes Bündel aus theoretischen Leitsätzen und Methoden.

Ein Programmierparadigma ist ein fundamentaler Programmierstil.

Statisch Programmelemente: Methoden, Variablen (zum Speichern von Daten).

Dynamische Programmelemente: Zuweisungen, Kontrollfluss, Datenfluss

1.2 Paradigmen der Programmiersprache C

1.2.1 Variable

Ein Behälter für eine Größe(Datum) die während eines Rechenprozesses auftritt. Im Normalfall erhält eine Variable einen Namen und wird im Speicher der Maschine abgelegt.

```
//aktuelle Uhrzeit
uint8_t Second;
uint8_t Minute;
uint8_t Hour;
```

1.2.2 Imperative Programmierung

Ein Programm ist eine Folge von Anweisungen. Sie orientiert sich an der Ausführung von Maschinen-code auf Prozessoren mit einer Von-Neumann-Architektur.

```
//Variablen initialisieren C-Code
Second=3;
Minute=2;
Hour=1;

//Kompilierter(übersetzter) Code
//Adresse Maschinenbefehle Assemblercode (ARM)
0x00000E66 2003 MOVS r0, #0x03
0x00000E68 9020 STR r0, [sp, #0x80]
0x00000E6A 2502 MOVS r5, #0x02
0x00000E6C 2701 MOVS r7, #0x01
```

1.2.3 Prozedurale Programmierung

Eine Aufgabe (Programm) wird in überschaubare Teile (Unterprogramm) zerlegt. In C wird ein Unterprogramm Funktion genannt. Es können Parameter übergeben und Ergebnisse zurückgegeben werden.

```
//Zeit einlesen
Second=Time_getSecond();
wait_ms(1);
Minute=Time_getMinute();
wait_ms(1);
Hour=Time_getHour();
wait_ms(1);
```

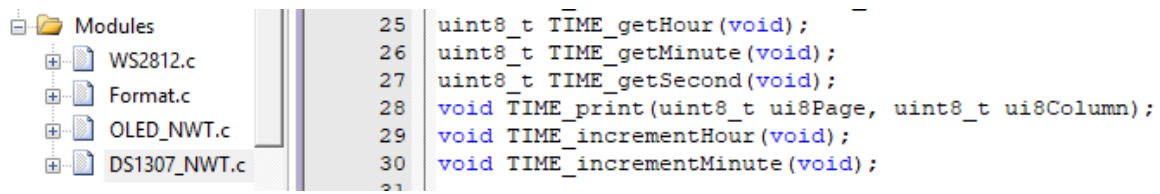
1.2.4 Strukturierte Programmierung

Es wird auf absolute Sprünge verzichtet und stattdessen mit Kontrollstrukturen gearbeitet.

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

1.2.5 Modulare Programmierung

Prozeduren werden in logischen Einheiten zusammengefasst. Sie werden Module genannt.



```
25 uint8_t TIME_getHour(void);  
26 uint8_t TIME_getMinute(void);  
27 uint8_t TIME_getSecond(void);  
28 void TIME_print(uint8_t ui8Page, uint8_t ui8Column);  
29 void TIME_incrementHour(void);  
30 void TIME_incrementMinute(void);  
31
```

2 Die Zeiteinheit

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

2.1 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();
```

2.2 Beispiel

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

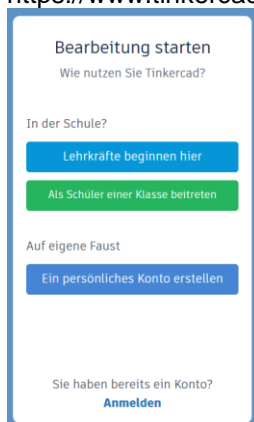
```
//Minuten mit Taster SW2 stellen  
if(SW2_get()==1)  
{  
    Time_incrementMinute(); //Minute erhoehen  
    Time_print(1,0); //Zeit anzeigen  
    DisplayLEDS=1; //Ausgabeanforderung(Flag) fuer LEDs setzen  
}
```

2.3 Wort Uhr mit tinkercad.com testen und weiter entwickeln

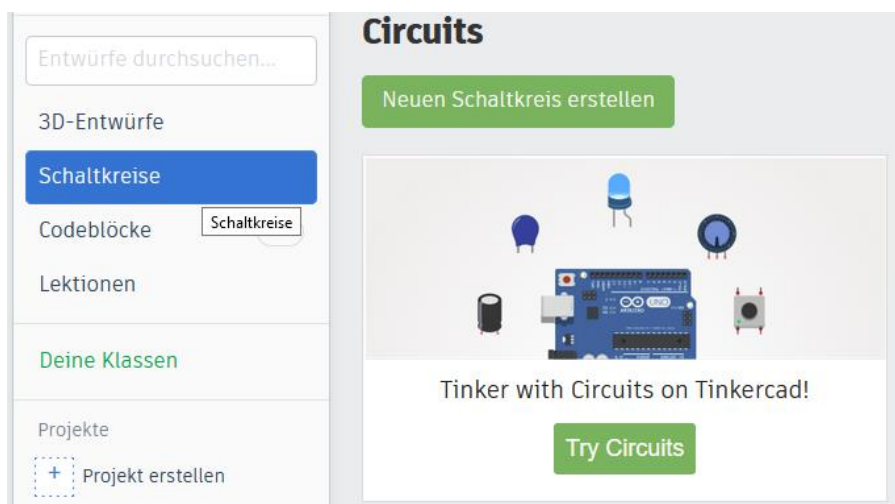
Es gibt zwei Möglichkeiten mit tinkercad zu arbeiten.
Ihr erstellt ein persönliches Konto oder ihr tretet einer Klasse bei. Dies muss die Schule vorgeben.

Webseite:

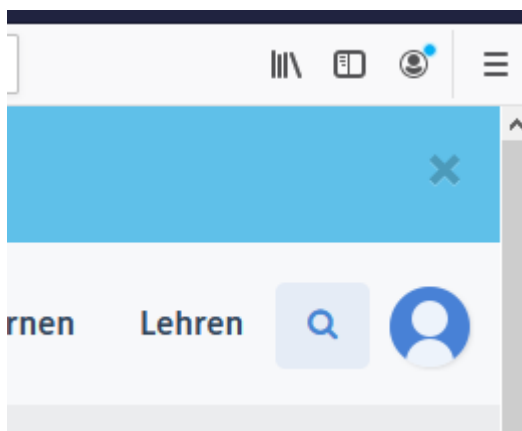
<https://www.tinkercad.com/>



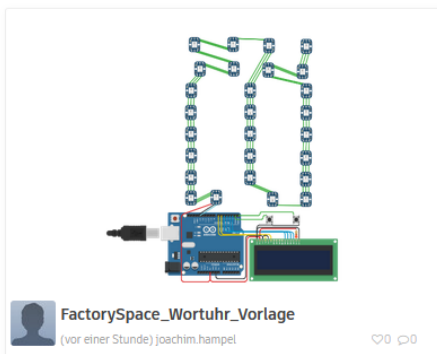
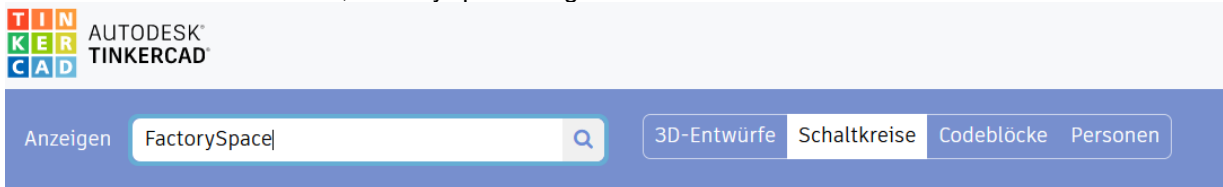
Nach der Anmeldung auf Schaltkreise wechseln.



Dann suchen auswählen



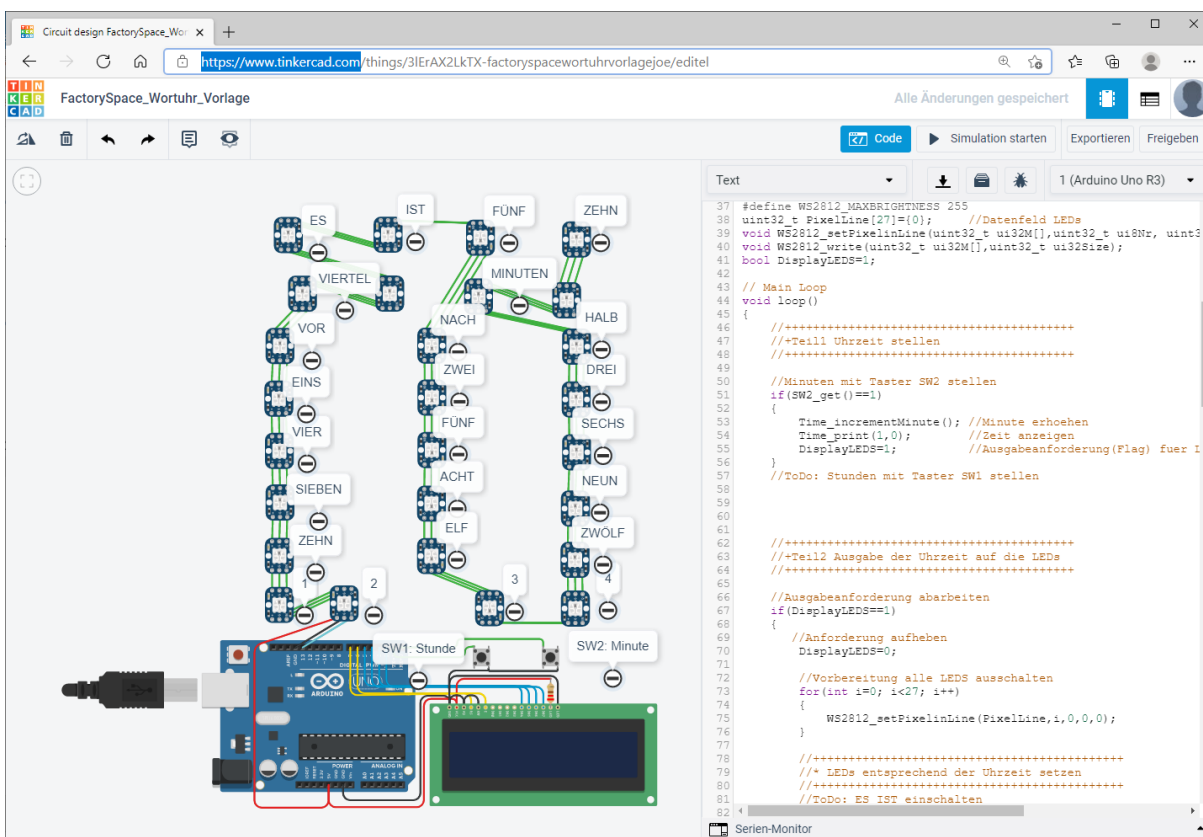
Hier Schaltkreise auswählen, FactorySpace eingeben und suchen.



Das angezeigte Projekt FactorySpace_Wortuhr_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.



2.4 Programmieraufgabe Stunde stellen

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

//ToDo: Stunden mit Taster SW1 stellen

3 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.



3.1 Farbwerte

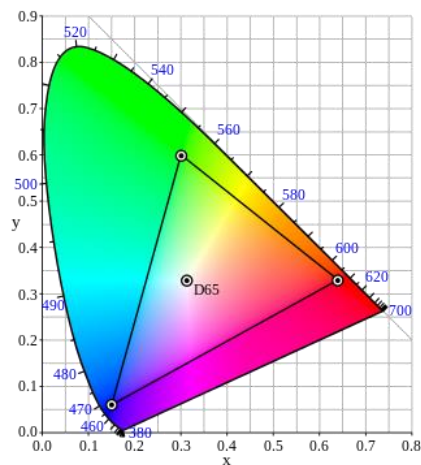
Die Farbwerte für die 27 LEDs werden in einem Datenfeld gespeichert.

```
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 Farbwerte 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);
```

3.2 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
```

4 Programmieraufgaben Wort Uhr LEDs

4.1 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<27; i++)
    {
        WS2812_setPixelinLine(PixelLine,i,0,0,0);
    }

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

    // T O D O

    //+++++
    /* Farbwerte an LEDs uebertragen
    //+++++
    WS2812_write(PixelLine,27);
}
```


4.2 Testen der LEDs und einschalten der Wörter *ES IST*

Es kommt leider immer wieder mal vor, dass Unterlagen aus anderen Abteilungen fehlen. So auch hier. Es gilt heraus zu finden wo sich LED0 bis LED26 befinden.

Unter dem Punkt //ToDo: *ES IST* einschalten kann nun mit der Funktion *WS2812_setPixelinLine* getestet werden.

```
//To Do: ES IST einschalten  
WS2812_setPixelinLine(PixelLine,7,50,10,20); //ES
```

Fertigt eine Skizze an. Testet mit verschiedenen Nummern und nummeriert die LEDs. Wegen der Farben solltet ihr frühzeitig Kontakt mit der Design-Abteilung aufnehmen denn die haben meistens so ihre ganz eigenen Vorstellungen.

Nach den Tests müssen an der Stelle //ES *IST* einschalten die Leuchtdioden für *ES* und *IST* dauerhaft eingeschaltet werden.

4.3 Einschalten der Zusatz Minuten

Die Uhrzeit wird in fünf Minuten Schritten in Wörtern angezeigt.

ES IST FÜNF NACH DREI

Um die Ungenauigkeit auszugleichen wird ganzen unten ein Offset von 1-4 angezeigt. Die Minuten können so ganz genau abgelesen werden

Natürlich könnte man nun mit der if-Abfrage alle Minutenwerte, die kein Vielfaches von 5, sind testen.

```
If (Minute==1)      -> 1 setzen  
If (Minute==2)      -> 2 setzen  
If (Minute==3)      -> 3 setzen  
If (Minute==4)      -> 4 setzen  
If (Minute==6)      -> 1 setzen  
If (Minute==7)      -> 2 setzen  
If (Minute==8)      -> 3 setzen  
If (Minute==9)      -> 4 setzen
```

Da kommen vielen Anweisungen zusammen. Eigentlich muss aber nur der Rest einer Ganzzahldivision mit 5 ausgewertet werden. Dafür wird der Modulo Operator % verwendet. Er ermittelt den Rest einer Division.

```
//To Do: Minuten 1 bis 4 anschalten.  
if (Minute%5==1)  
{  
    //To Do    LED Zusatzminute 1 setzen  
}
```

Jetzt funktioniert die Zusatzminute 1. Die Abfrage muss jetzt noch für 2,3 und 4 wiederholt werden. Und unbedingt ausführlich testen.

4.4 Minuten Wörter anschalten

Auch für die nächste Aufgabe sind die Vorüberlegungen sehr wichtig und Dauern unter Umständen länger wie das Schreiben des eigentlichen Programmcodes.

Bei der Uhrzeit 12:05 gilt es folgendes anzuzeigen: ES IST FÜNF MINUTEN NACH ZWÖLF

Am besten erstellt man eine Tabelle in der die Wörter und die entsprechenden Nummern gelistet sind.

Achtung: Die Wörter VIERTEL und MINUTEN sind so breit, dass 2 LEDs zur Hinterleuchtung benötigt werden.

Uhrzeit	Wörter	LED Nummern
XX:00	--	--
XX:05	FÜNF MINUTEN NACH	
XX:10		
XX:15		
XX:20		
XX:25		
XX:30		
XX:35		
XX:40		
XX:45		
XX:50		
XX:55		

Die Wörter FÜNF MINUTEN NACH müssen für die Minuten :05, :06, :07, :08 und :09 angezeigt werden.

Die naheliegende Lösung wären fünf Abfragen

```
if(Minuten==5)..
```

```
if(Minuten==6)..
```

```
if(Minuten==7)..
```

```
if(Minuten==8)..
```

```
if(Minuten==9)..
```

Dies lässt sich aber auch zu einer Abfrage verbinden. Dazu müssen zwei Bedingungen eingehalten werden.

Minuten muss größer oder genau 5 sein UND Minuten muss kleiner als 10 sein.

```
//To Do: Minuten Woerter anschalten
if(Minute>=5 && Minute<10)
{
    //TO DO: passende Wörter anschalten
}
```

Auch wenn jetzt erst 5-9 Minuten nach funktionieren unbedingt testen.

Nun ist die Vorgehensweise geklärt. Jetzt muss der Code für alle Spalten in der Tabelle erstellt und ausführlich getestet werden.

4.5 Stunden Wörter anschalten

Hier reicht eine einfache Abfrage der Stunde aus.

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

Auch hier gilt es muss unbedingt ausführlich getestet werden. Im Sprachgebrauch wird 1:30 zum Beispiel HALB ZWEI genannt.

Es reicht also nicht aus nur direkt auf die Stunde zu reagieren. Findet eine Lösung.

Vorgehensweise wenn ihr keine Lösung findet:

1. Im Team diskutieren (ihr kennt eure Probleme am Besten)
2. Mit den Vorgesetzten diskutieren (z.B. Lehrer)
3. Externe Berater einbinden (das Factory Space Team der Hochschule Pforzheim)

5 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.