

```

// Example sketch for interfacing with the DS1302 timekeeping chip.
// http://quadpoint.org/projects/arduino-ds1302
// https://github.com/msparks/arduino-ds1302 Lib download
/*Pinbelegung für Arduino
*
*LCD Display:
*GND auf GND
*Vcc auf 5V
*SLC auf A5 Arduino
*SDA auf A4 Arduino
*
*DS1302
*Vcc auf 5V
*GND auf GND
*CLK auf GPIO 12
*DAT auf GPIO 11
*RST auf GPIO 10
*
*LED GPIO 7
*Schalter für Uhr GPIO 4
*
*/
#include <stdio.h> //Standard Lib für Formatierung
#include <DS1302.h>
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27,20,4); //LCD initialisieren

// Create a DS1302 object.
DS1302 rtc(10, 11, 12); //Pins für RST,DAT,CLK

//Zuweisung der Wochentage
String dayAsString(const Time::Day day) {
    switch (day) {
        case Time::kSunday: return "Sonntag";
        case Time::kMonday: return "Montag";
        case Time::kTuesday: return "Dienstag";
        case Time::kWednesday: return "Mittwoch";
        case Time::kThursday: return "Donnerstag";
    }
}

```

```

    case Time::kFriday: return "Freitag";
    case Time::kSaturday: return "Samstag";
}
return "(unknown day)";
}

void setup() { //-----
Serial.begin(9600);
lcd.init();
lcd.setBacklight(HIGH);
pinMode(7,OUTPUT);
digitalWrite(7,LOW);

// Initialisieren Sie einen neuen Chip, indem Sie den Schreibschutz deaktivieren und den löschen
// Uhr Stopp Flag. Diese Methoden müssen nicht immer aufgerufen werden. Siehe DS1302
// Datenblatt für Details.

rtc.writeProtect(false);
rtc.halt(false);

// Erstelle ein neues Zeitobjekt, um Datum und Uhrzeit einzustellen. nach einmal laden auskommentieren
//Jahr, Monat,Tag,Stunde,Minute,Sekunde::Wochentag

//Time t(2020, 05, 23, 15, 44, 50, Time::kSaturday); //Eingabe aktuelles Datum u. Zeit nach 1. Durchlauf
auskommentieren

// Uhrzeit und Datum auf dem Chip speichern nach 1. Durchlauf auskommentieren
//rtc.time(t);
}
//-----Ende Setup-----

void printTime() { //Funktion zum Ausgeben Datum u. Zeit-----

// Aktuelle Uhrzeit und Datum vom Chip abrufen.

Time t = rtc.time();

// Nenne den Wochentag.
const String day = dayAsString(t.day);

```

```

// Uhrzeit und Datum formatieren und in den temporären Puffer einfügen.
char buf[50];

//Formatierung der Ausgabe
//sprintf(buf, sizeof(buf), "%s %04d-%02d-%02d %02d:%02d:%02d",day.c_str(),t.yr, t.mon, t.date,t.hr, t.min, t.sec);
//org
sprintf(buf, sizeof(buf), "%s %02d.%02d.%04d %02d:%02d:%02d",day.c_str(), t.date, t.mon,t.yr,t.hr, t.min, t.sec);
//deutsche Schreibweise
//ergibt Samstag 23.05.2020 16:55:20

// Drucke die formatierte Zeichenfolge auf seriell Ausgabe
Serial.println(buf);

//auf LCD
lcd.clear();
lcd.setCursor(1,0); //2.Stelle 1.Zeile
lcd.print(t.date); //Tag
lcd.print("."); //Trennzeichen
lcd.print(t.mon); //Monat
lcd.print(".");
lcd.print(t.yr); //Jahr
lcd.setCursor(1,1); //2.Stelle 2.Zeile
lcd.print(t.hr); //Stunde
lcd.print(":");
lcd.print(t.min); //Minute
lcd.print(":");
lcd.print(t.sec); //Sekunde
lcd.setCursor(1,2); // 2.Stelle 3.Zeile
lcd.print(day.c_str()); //Wochentag

//geht auch über Umwandeln in String und zusammenführen
String ges = String(t.date) + "." + String(t.mon) + "." + String(t.yr);
Serial.println(ges); //aber ohne führende 0

//oder komplett für ganzen string und mit führender 0
char s[20];
sprintf(s,"%02d.%02d.%04d",t.date, t.mon, t.yr);
Serial.println(s);
Serial.println("Test:");

```

```

//Berechnung der akt Stunde
float aktstunde = (t.min / 60.0) + t.hr ;
Serial.println("Ausgangswerte");
Serial.println(t.min);
Serial.println(t.hr);
Serial.println("akt.Stunde");
Serial.println(aktstunde);

//Zeituhr1-----

int resultat1 =0; //Zustand Uhr1
int resultat2 =0; //Zustand Uhr2

int anfh = 10; //Anfangsstunde
int anfmin = 30; //Anfangsminute
int endh =10; //Endstunde
int endmin = 40; //Endminute

float anfstunde1=(anfmin / 60.0) + anfh ; //Berechnung Anfangsstunde
float endstunde1=(endmin / 60.0) + endh ; //Berechnung Endstunde
Serial.println ("anfstunde1");
Serial.println (anfstunde1);
Serial.println ("endstunde1");
Serial.println (endstunde1);

if( aktstunde >= anfstunde1 && aktstunde <=endstunde1 ){ //unter Anfangszeit
  Serial.println("an1");
  resultat1=1;
}
else{
  resultat1=0;
}

//Zeituhr2-----

int anfh2 = 15; //Anfangsstunde
int anfmin2 = 01; //Anfangsminute
int endh2 =17; //Endstunde
int endmin2 = 05; //Endminute

float anfstunde2=(anfmin2 / 60.0) + anfh2 ; //Berechnung Anfangsstunde

```

```

float endstunde2=(endmin2 / 60.0) + endh2 ; //Berechnung Endstunde

Serial.println ("anfstunde2");
Serial.println (anfstunde2);
Serial.println ("endstunde2");
Serial.println (endstunde2);

if( aktstunde >= anfstunde2 && aktstunde <=endstunde2 ){ //unter Anfangszeit
  Serial.println("an2");
  resultat2=1;
}
else{
  resultat2=0;
}

// Zeituhren auswertung-----

if (digitalRead(4)==HIGH){ //Schaltuhr nur aktiv wenn Pin 4 auf HIGH gesetzt ist

  if (resultat1 == 1 or resultat2 == 1){ //resultat1 und resultat2 werden ausgewertet
    Serial.print("LED an");
    lcd.setCursor(1,3); //2.Stelle 4.Zeile
    lcd.print("LED an ");
    digitalWrite(7,HIGH); //Pin Schalten
  }
  else{
    lcd.setCursor(1,3); //2.Stelle 4.Zeile
    lcd.print("LED aus ");
    digitalWrite(7,LOW); //PinSchalten
  }
}
else{
  lcd.setCursor(1,3); //2.Stelle 4.Zeile
  lcd.print("Schaltuhr aus");
}
}
//-----Ende Funktion-----

void loop() {
  printTime(); //Funktionsaufruf
}

```

```
delay(5000);  
}
```