	Mikrocontroller-Labor	Name: Rahm Datum: 06.11.2017 6_3_1_FES_Datenlogger.docx
	Datenlogger	6.3.1

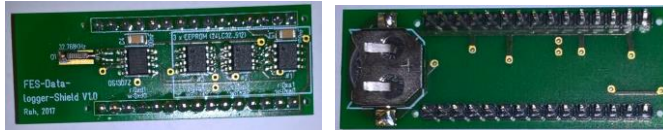
Projekt: Datenlogger zur Langzeit-Temperaturaufzeichnung

Die Hardware des Datenloggers besteht neben dem Mikrocontroller aus einem Temperatursensor, einer Echtzeituhr (Real-Time-Clock) und einem EEPROM zur Speicherung der Datensätze (Records).

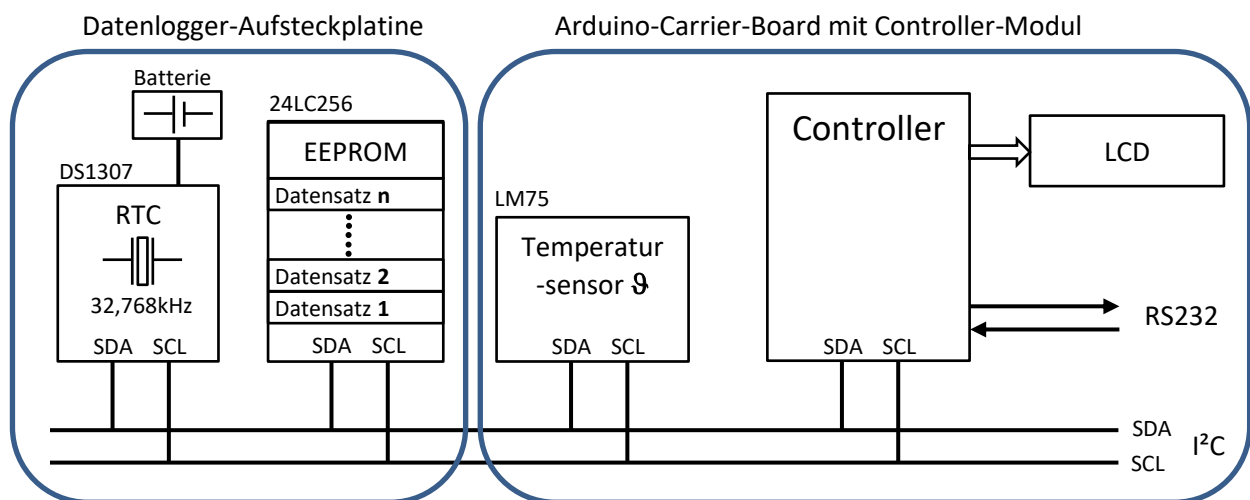
Die Temperatur wird alle 2min erfasst und als Datensatz mit Datum und Uhrzeit im EEPROM gespeichert. Der aktuelle Temperaturwert wird im Display angezeigt.

Der Datenlogger wird mit einem Terminalprogramm ausgelesen, welches die Daten über die RS232-Schnittstelle erhält. Auswertung und Darstellung der Daten erfolgt mit Excel.

Aufsteckplatine mit RTC, EEPROM und Pufferbatterie



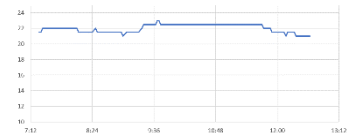
Übersichtsschema



Arbeitsauftrag

- 🔧 Erstellen Sie ein neues Projekt und binden Sie die Bibliotheken datalogger.c, eeprom.c und rtc.c ein.
- 🔧 Erstellen Sie den C-Code mit Hilfe des Vorgabeblatts, oder öffnen Sie die Vorlagedatei.
- 🔧 Um den Datenlogger über ein Terminal zu steuern, wird der serielle Interrupt verwendet. Wie alle Interrupts gibt es dafür vier Funktionen: **serial_interrupt_init()**, **serial_interrupt_enable()**, **serial_interrupt_disable()** und die Interrupt-Service-Routine (ISR) **serial_interrupt_isr()**. Die ISR wird ausgelöst, wenn ein Zeichen von der seriellen Schnittstelle empfangen wird. Erstellen Sie den Code für die Kommandos: 't' bzw. 'T': Uhrzeit und Datum stellen. 'p' bzw. 'P': Alle Datensätze über RS232 ausdrucken. 'n' bzw. 'N': Neuaufzeichnung starten.

Verwenden Sie die fertigen Bibliotheksfunktionen (siehe Datenlogger-Bibliotheken).



Datum	Zeit
16.11.17	23:26
14:	20,5°C
Aktueller Datensatz	Temperatur

```
void serial_interrupt_isr(void)
{
    uint8_t c;


    c=rs232_get();

    switch (c)
    {
        case 't':
        case 'T': // Zeit einstellen

            break;

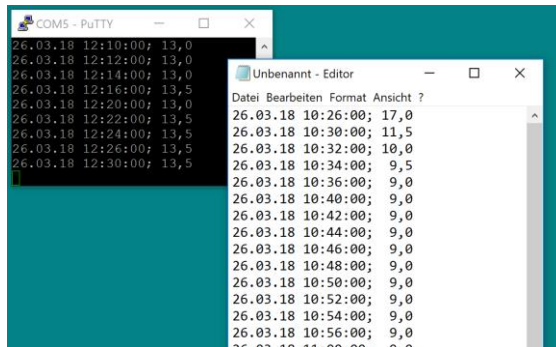
        .....

        default: break;
    }
}
```


 Friedrich-Ebert-Schule Esslingen FES	Mikrocontroller-Labor		Name: Rahm Datum: 06.11.2017 6_3_1_FES_Datenlogger.docx
	Datenlogger		6.3.2

Darstellen des gespeicherten Temperaturverlaufs mit Excel

1. Starten Sie putty.exe (oder ein vergleichbares Terminal-Programm) und öffnen Sie die Verbindung zum Datenlogger.

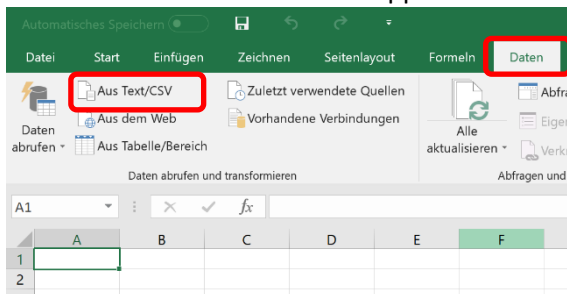


Durch Eingabe von 'p' werden alle gespeicherten Daten ausgelesen und im Terminalfenster angezeigt.

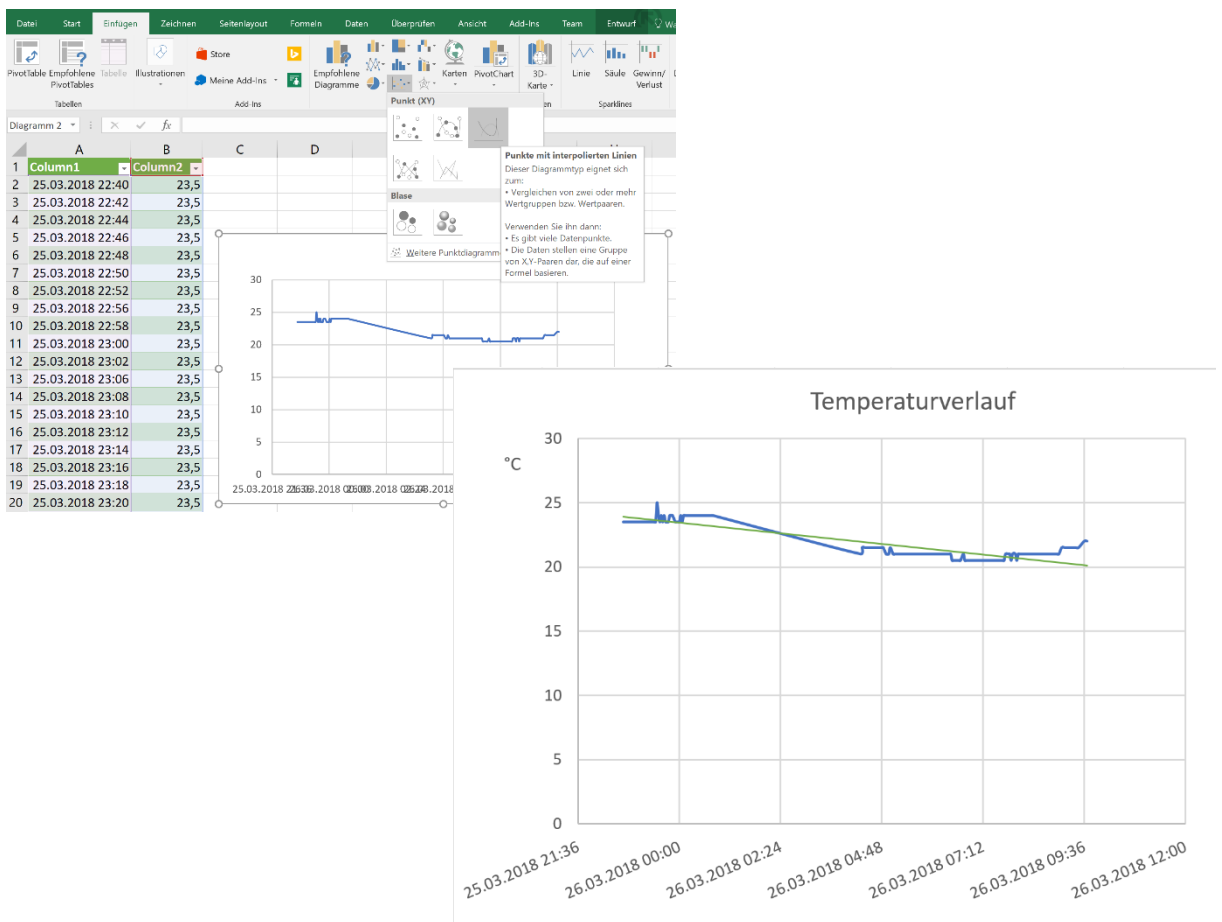
Mit Rechtsklick auf das Fenstericon  von PuTTY öffnet sich ein Menü, wo Sie 'Copy All to Clipboard' auswählen. Alternativ können Sie auch die gewünschten Datensätze markieren und mit 'Strg+c' in die Zwischenablage kopieren.


Öffnen Sie nun den Windows-Editor und fügen Sie die Datensätze mit 'Strg+v' in den Editor ein. Speichern Sie die Datei unter einem beliebigen Namen.

2. Öffnen Sie eine neue Arbeitsmappe in Excel und wählen Sie aus dem Reiter 'Daten' den Eintrag 'Aus Text/CSV'. Im Importfilter laden Sie die vorher erstellte Datei.



3. Markieren Sie die Spalten A und B mit den Daten und fügen Sie ein Punkt (XY)-Diagramm ein. Formatieren Sie anschließend das Diagramm. Bei Bedarf können Trendlinien eingefügt werden.



 Friedrich-Ebert-Schule Esslingen FES	Mikrocontroller-Labor	Name: Rahm Datum: 06.11.2017 6_3_1_FES_Datenlogger.docx
	Datenlogger	6.3.3

```

#include "controller.h"
#include "datalogger.h"


// Funktions-Prototypen
void lcd_print_timestamp(void);
// Globale Variablen
volatile uint16_t _SAMPLE_TIME_ = 120; // Ein Sample alle 120s (=2min)

//*****
void setup(void)
{ // Initialisierungen
  lcd_init();
  i2c_init();
  rtc_init();
  eeprom_init();
  rs232_init();
  serial_interrupt_init(serial_interrupt_isr);
  //Startbildschirm-----
  lcd_clear();
  lcd_setcursor(1,1);
  lcd_print("FES Datenlogger");
  lcd_setcursor(2,1);
  lcd_print("COM: 9600Bd");
  delay_ms(4000);
  lcd_clear();
  //-----
  aktueller_Datensatz = read_current_recordnumber_from_rtc();
  aktueller_Datensatz++; // Damit die letzten Daten nicht überschrieben werden.
}
//*****
int main(void)
{
  volatile uint16_t sample_sekunde;
  uint8_t n=1;

  setup();
  serial_interrupt_enable(); // Ab hier Steuerung über RS232 möglich.

  while (1) // Endlosschleife loop()
  {
    temperatur = lm75_read();
    rtc_get(); // Aktuelles Datum und Zeit holen.
    sample_sekunde = sekunde + minute * 60 + stunde * 60*60;
    if((sample_sekunde % _SAMPLE_TIME_) == 0)
    {
      if(aktueller_Datensatz >= MAX_RECORD) aktueller_Datensatz = 0; // Neubeginn
      write_current_recordnumber_to_rtc(aktueller_Datensatz);
      eeprom_set_record(EEPROM_1,aktueller_Datensatz++);
    }
    lcd_print_timestamp();
    lcd_setcursor(2,1);
    lcd_int(aktueller_Datensatz); lcd_char(':');
    lcd_setcursor(2,8);
    lcd_print_temperatur(temperatur); // Ausgabe
    rs232_print_temperatur(temperatur,2); // Ausgabe auf Terminal (z.B. 23,5°C)
    delay_ms(1000);
  }
}
//*****
void lcd_print_timestamp(void)
{
  lcd_setcursor(1,1);
  lcd_dd(tag); lcd_char('.'); lcd_dd(monat); lcd_char('.'); lcd_dd(jahr);
  lcd_print(" "); lcd_dd(stunde); lcd_char(':'); lcd_dd(minute);
}

```

 Friedrich-Ebert-Schule Esslingen FES	Mikrocontroller-Labor	Name: Rahm Datum: 06.11.2017 6_3_1_FES_Datenlogger.docx
	Datenlogger	6.3.4

Lösung

```

void serial_interrupt_isr(void)
{
    uint8_t c;

    c=rs232_get();

    switch (c)
    {
        case 't':
        case 'T': // Zeit einstellen
            rs232_set_time();
            rs232_put('\n');
            break;

        case 'p':
        case 'P': // alle Datensätze drucken
            serial_print_all_records();
            break;

        case 'n':
        case 'N': // Aufzeichnung neustarten
            rs232_print("Alle Datensätze werden gelöscht [j/n]: ");
            while((c=rs232_get())== 0);
            rs232_put(c);
            if ((c=='j')||(c=='J'))
            {
                aktueller_Datensatz = 0;
                write_current_recordnumber_to_rtc(aktueller_Datensatz);
            }
            rs232_put('\n');
            break;

        case 'h':
        case 'H': // Hilfe anzeigen
            rs232_print("\nHilfe:\n");
            rs232_print("t: Datum und Zeit einstellen\n");
            rs232_print("p: Alle Datensätze drucken\n");
            rs232_print("n: Aufzeichnung neustarten\n");
            rs232_print("s: Sample-Zeit einstellen [default 120s]\n");
            break;

        default: break;
    }
}

```