



DEUTSCHE
GESELLSCHAFT FÜR
ZERSTÖRUNGSFREIE
PRÜFUNG E.V.

ZfP-Sonderpreis der DGZfP beim Landeswettbewerb Jugend forscht

SAARLAND



Multifunktions-Datenlogger

Dominik Bastian

Schule:

Peter-Wust-Gymnasium
Merzig

Jugend forscht 2018

Multifunktions-Datenlogger



Kurzfassung

Oft ist es nicht einfach, verschiedene physikalische Größen zu messen. Man benötigt viele Geräte, meist eines für jede zu messende Größe. Diese Daten dann auch noch über längere Zeiträume zentral aufzuzeichnen, ist dann noch schwieriger. Für einzelne Fachbereiche gibt es solche Geräte, möchte man aber eigene Systeme überwachen, sind passende Geräte eher selten.

Der „Multifunktions-Datenlogger“ oder kurz „MultiLogger“ bietet eine Möglichkeit, verschiedenste Größen kompakt auf einem Display darzustellen sowie diese auf eine SD-Karte abzuspeichern und an einen PC zu senden. Zudem kann man die Messwerte auch per WLAN auf dem Smartphone oder Tablet anzeigen lassen. Auswerten lassen sich die Datensätze dann mithilfe von Microsoft Excel.

Inhaltsverzeichnis

Kurzfassung.....	2
1. Einleitung.....	3
2. Der MultiLogger.....	3
2.1 Funktionen.....	3
2.2 Das Gerät MultiLogger.....	4
2.3 Einsatzmöglichkeiten.....	5
2.3.1 Heizungsbauer.....	5
2.3.2 Physikunterricht in Schulen.....	6
2.3.3 Privater Einsatz.....	6
3. Technische Umsetzung.....	6
3.1 Mikrocontroller ATmega1284P.....	6
3.2 I2C-Bus.....	6
3.3 WLAN.....	6
3.4 Echtzeituhr.....	7
3.5 Analogeingänge.....	7
3.6 Digitaleingänge.....	8
3.7 Temperatursensoren.....	8
3.8 Externe Sensoren im I2C-Bus.....	8
3.8.1 PeakTech 4000.....	9
3.8.2 Feinstaubsensor.....	9
4. Programmierung.....	9
4.1 Boards.....	9
4.2 Libraries.....	9
4.3 PC.....	10

5. Bewertung.....	10
6. Danksagung.....	10
7. Quellennachweis.....	10

1. Einleitung

Die ursprüngliche Idee zu diesem Projekt war es, die Prüflöstemperaturen an einer Prüfbank zur Fehlersuche aufzuzeichnen. Daher sollten sich die Messgrößen zunächst auf Temperaturen beschränken. Doch schnell merkte ich, dass in der Technik wesentlich mehr Potential steckt. Daher entschied ich mich, das Gerät um zusätzliche Messgrößen zu erweitern.

Mir fiel dabei auch auf, dass es auf dem Markt bereits einige Produkte gibt, die das Messen und Aufzeichnen von Daten ermöglichen. Doch bei diesen Geräten fehlte mir die Erweiterbarkeit und Flexibilität, die mein MultiLogger ermöglicht. Oft ist es auch nicht möglich, gleichzeitig viele Messwerte zu erfassen, da die Produkte meist nur wenige Kanäle unterstützen.

Dabei stellte ich mir die folgenden Fragen:

- Wie flexibel, anpassbar und erweiterbar ist ein Mess- und Aufzeichnungssystem, das mit einfachen Mitteln erstellt wurde?
- Wie ist es möglich, ohne das Grundgerät zu verändern, weitere Messbereiche hinzuzufügen?

2. Der MultiLogger

2.1 Funktionen

Der MultiLogger selbst kann vier Temperaturen mit den Sensoren DS18B20 messen. Außerdem können acht Spannungen im Bereich von -12V bis 12V und acht Digitaleingänge gemessen werden. Zudem soll die Möglichkeit gegeben werden, weitere Sensoren anzuschließen, die über einen Adapter verbunden werden (vgl. 3.8 Externe Sensoren im I2C-Bus). Jeder Sensor hat einen Namen, den man vom PC aus einstellen kann. Dadurch wird die spätere Zuordnung der Sensoren leichter. Der Name kann eine Länge von maximal 15 Zeichen haben. Er bleibt auch nach dem Ausschalten des MultiLoggers erhalten, weshalb zum Betrieb ein PC nicht unbedingt erforderlich ist.

Die Messwerte werden auf einem Display angezeigt und sind somit direkt ohne ein weiteres Gerät ablesbar. Zudem sind die Messwerte auf einem Smartphone mittels eines WLAN-Netzes ablesbar. Dadurch kann man sich einfach vom Gerät wegbewegen und weiterhin problemlos Messungen durchführen. Es wird keine App benötigt, ein normaler Internetbrowser genügt. Das macht das System einfach einzurichten und plattformunabhängig. Ein Handy ist wesentlich einfacher mitzunehmen als beispielsweise ein Laptop. Zuschaltbar ist die Funktion, die Messwerte auf eine SD-Karte, die in das Gerät eingelegt wird, zu schreiben. Die Datensätze werden im CSV-Format abgelegt, sodass man sie leicht

in Excel weiter bearbeiten kann. Sie werden mit Datum und Uhrzeit abgespeichert. Dadurch kann man die Daten einfach auswerten. Außerdem können die Messwerte über eine serielle Schnittstelle an einen PC übertragen werden. Dieser schreibt die Werte in eine Excel-Tabelle und generiert daraus Diagramme, deren Aufbau bereits während des Aufzeichnens erfolgt. Das dazu benötigte Programm kann außerdem Einstellungen des MultiLoggers bearbeiten, wie z.B. die Sensornamen ändern oder die Uhrzeit einstellen.

Die Messwerte können entweder automatisch aufgenommen werden. Dann werden in einem vom PC aus einstellbaren Zeitintervall automatisch Messwerte aufgezeichnet. Dabei ist das kleinste Intervall eine Sekunde, da das Auslesen der Sensoren einige Zeit in Anspruch nimmt. Eine Alternative dazu bietet das manuelle Aufzeichnen, bezeichnet mit „Snapshot“. In diesem Fall werden Daten abgespeichert, wenn ein Knopf am Gerät betätigt wird.

Der MultiLogger wird mittels eines Steckernetzteils mit Strom versorgt. Es wird also nur eine normale Steckdose zum Betrieb benötigt. Alternativ dazu wäre aber auch die Versorgung mithilfe eines Akkus, zum Beispiel ein LiPo mit drei Zellen, denkbar, wird derzeit aber noch nicht unterstützt.

Basierend auf einem früheren Prototyp wurde der MultiLogger um die grafische Ausgabe mit einem Grafikdisplay (240 x 128 Pixel) erweitert. Das ermöglicht eine übersichtlichere Darstellung der Daten, da mehr auf den Bildschirm passt. Dadurch werden alle Messwerte einer Größe auf einer Seite angezeigt, wie auch die Namen der Sensoren. Man muss weniger durchblättern. Des Weiteren bietet das Display die Möglichkeit, bereits auf dem MultiLogger Diagramme anzuzeigen.

2.2 Das Gerät MultiLogger

Zunächst habe ich einen Prototyp des Loggers gebaut. Dieser ist auf Lochrasterplatine aufgebaut, da dieses System leicht zu bauen und auch nachträglich erweiterbar ist. Er verwendet noch ein zweizeiliges Text-Display mit 16 Zeichen pro Zeile. Als ich dann mit diesem Gerät zufrieden war, entschied ich mich, mit einem neuen, besseren Display einen Neubau zu beginnen. Ich habe dazu zunächst am Computer einen Schaltplan gezeichnet. Der Plan ist aus dem Prototyp entstanden, die Geräte sind elektrisch größtenteils identisch. Dann habe ich aus dem Schaltplan selbst eine Platine erstellt. Diese wurde dann zuhause mit der Hilfe meines Vaters geätzt.



Abb. 1: Der Prototyp

Das Gerät wird mittels fünf Tasten bedient. Mit „Bild auf“ und „Bild ab“ kann man verschiedene Ansichten von verschiedenen Messwerten wählen. Die Taste „SD-

„Logging“ wird benötigt, um festzulegen, ob die Daten auf die SD-Karte geschrieben werden. Die LED daneben zeigt an, ob die Funktion aktiviert ist. Mit „Live-Logging“ wird aktiviert, dass Daten an einen PC übertragen werden. Auch hier signalisiert die LED die Aktivität der Funktion. Mit der Taste „Snapshot“ kann eine einzelne Messung aufgenommen werden.

Der MultiLogger verfügt über Anschlüsse für die einzelnen Messgrößen. Am Gerät selbst befinden sich Anschlüsse für acht Spannungen und acht digitale Signale. Die Buchsen für die Temperatursensoren und die externen Geräte sind doppelt vorhanden, obwohl je eine genügt hätte. Zudem finden sich dort Spannungsausgänge für Masse, sowie 5V, 15V und -15V. Benötigt ein externes Gerät oder ein System, an dem gemessen werden soll, eine Spannungsversorgung, so kann dieses bei geringen Strömen auch vom MultiLogger versorgt werden. Damit benötigt es keine weitere Spannungsversorgung. An der Rückseite befindet sich der Stromstecker sowie der SD-Kartenslot und die serielle Schnittstelle.



Abb. 2: Das Bedienfeld des MultiLoggers (links) und Rückseite (rechts)



Abb. 3: Die Anschlüsse für Messeingänge und Ausgänge der Spannungen.

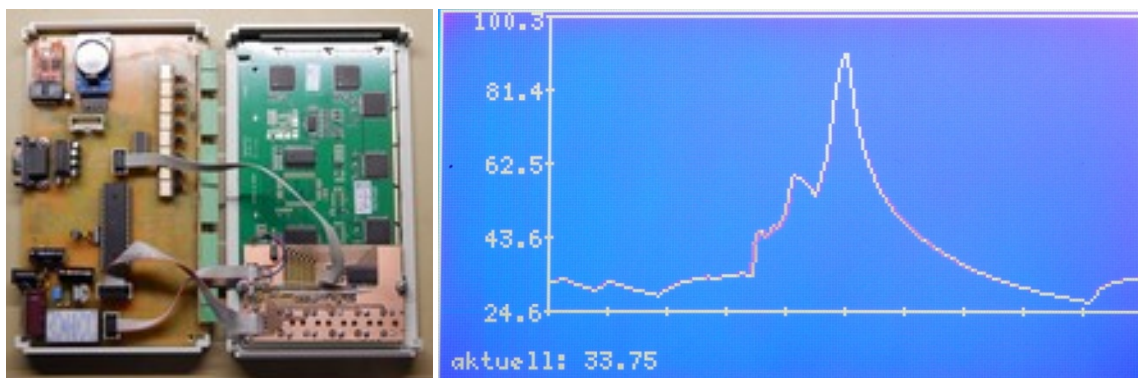


Abb. 4: Blick in das geöffnete Gerät (links) und ein Beispiel für ein Diagramm (rechts)

2.3 Einsatzmöglichkeiten

Durch die hohe Flexibilität des MultiLoggers sind seine Einsatzmöglichkeiten nahezu unbegrenzt. Ich möchte im Folgenden einige davon kurz vorstellen.

2.3.1 Heizungsbauer

Ein gutes Einsatzbeispiel für den MultiLogger sind Heizungsbauer. Beklagt sich ein Kunde über ein Fehlverhalten seiner Heizung und der zu Hilfe gerufene Handwerker kann den Fehler nicht vor Ort finden, so bietet es sich an, den Logger zu installieren und einige wichtige Temperaturen zu beobachten. Ein längerer Zeitraum kann erfasst und auf diese Weise ein selten auftretender Fehler doch nachvollzogen werden. So ist es im Nachhinein möglich, genau zu erkennen, bei welchen Ereignissen eine Störung auftritt. Das erleichtert die Fehlersuche enorm.

2.3.2 Physikunterricht in Schulen

Auch in Schulen kann der MultiLogger zum Einsatz kommen. Er kann zum Messen an Versuchen, die von der Lehrkraft durchgeführt werden, eingesetzt werden. Gegenüber anderen Produkten für den Schulunterricht hat der Logger den Vorteil, dass das verbaute WLAN auch mehrere Smartphones unterstützt. Somit ist es selbst in der letzten Bankreihe möglich, mehrere Messwerte komfortabel ablesen zu können, ohne das Pult mit Anzeigen vollstehen zu haben. Des Weiteren ist eine Möglichkeit geplant, aufgezeichnete Dateien auf das Smartphone herunterzuladen. Schüler haben dann digital Messwerte, müssen also nichts abschreiben. Eine Hausaufgabe könnte dann sein, die Daten selbstständig auszuwerten und Gesetze herzuleiten.

2.3.3 Privater Einsatz

Natürlich ist der MultiLogger auch für den privaten Gebrauch geeignet. Man kann zum Beispiel sein Haus überwachen, etwa hier eine Solaranlage, deren Temperaturen kontrolliert werden sollen. Verbindet man dann das WLAN des Loggers mit einem Router, so ist es möglich, bei einer bestehenden VPN-Verbindung auch von unterwegs Daten auslesen zu können.

Man kann den MultiLogger aber natürlich auch als Universalmessgerät verwenden.

3. Technische Umsetzung

3.1 Mikrocontroller ATmega1284P

Der Mikrocontroller des MultiLoggers ist ein ATmega1284P von Atmel. Das ist ein 8-bit AVR mit 40 Pins. Er ist mit 16 MHz getaktet. In seinem Flashspeicher steht das Programm, das die Messwerte sammelt, anzeigt und aufzeichnet. Ich verwende keinen weiteren Controller, da alle anfallenden Aufgaben von diesem vorhandenen erledigt werden.

3.2 I2C-Bus

Im Gerät verwende ich den sogenannten I2C-Bus. Das ist ein System, das es erlaubt, bis zu 128 verschiedene Busteilnehmer anzusteuern. Der eigentliche

Vorteil des Busses liegt darin, dass man dafür immer nur zwei Datenleitungen benötigt, egal wie viele Geräte man verwendet. Viele gängige Chips benutzen den I2C-Bus, so zum Beispiel Porterweiterungsbausteine oder welche, die Spannungen messen.

Ich verwende diesen Bus, da ich viele Eingänge erfassen möchte. Würde man diese alle an den Controller selbst anschließen, wäre dieser schnell zu klein. Durch die Verwendung des Busses benötige ich so nur zwei Pins.

3.3 WLAN

Zur Anbindung eines Smartphones oder Tablets verwendet der MultiLogger WLAN. Dazu verbaue ich den vom chinesischen Hersteller Espressif hergestellten Chip ESP8266. Dieser ist auf einem Modul von Olimex, dem MOD-WIFI-ESP8266 verbaut.

Der ESP8266 baut ein WLAN-Netzwerk namens „MultiLogger“ auf. Man kann sich dann mit einem beliebigen Gerät, das WLAN unterstützt, darauf einwählen. Im Internetbrowser wird die IP-Adresse des MultiLoggers eingegeben. Dann bekommt man eine Internet-Seite, auf der alle Messwerte des Loggers angezeigt werden. Man kann aber auch Darstellungen wählen, in denen alle Messwerte einer Größe oder nur ein Wert angezeigt werden.

Der ESP8266 wird nur für WLAN verwendet. Leider wird I2C nicht als Slave unterstützt, sondern nur als Master, weshalb ich das Modul nicht an den bereits vorhandenen I2C-Bus anschließen konnte. Daher musste ich auf eine serielle Schnittstelle für die Kommunikation mit dem ESP8266 zurückgreifen.

Auf dem ESP8266 läuft ein Webserver auf dem Port 80. Daher sendet ein Webbrowser eine Anfrage für eine Webseite an den MultiLogger, wenn man einfach nur die IP-Adresse verwendet. Die Webseiten schreibe ich am Computer in HTML von Hand mit Notepad++. Anschließend wird der Quelltext für eine Seite mit einem selbst geschriebenen Java-Programm in einen Text konvertiert, den ich in die Arduino IDE kopieren kann. Um Werte in die Webseite einzusetzen, verwende ich Platzhalter, die später mit Textersetzung durch die realen Werte ersetzt werden.

3.4 Echtzeituhr

Im MultiLogger ist eine Echtzeituhr (eng. real time clock, kurz RTC) verbaut. Sie ermöglicht es, die Messwerte mit einem Zeitstempel zu versehen. Das macht die Datensätze nachvollziehbar und gibt die Möglichkeit, die Daten zu einem bestimmten Zeitpunkt einzusehen.

Die Uhr wird ebenfalls am I2C-Bus betrieben. Sie verfügt über einen Akku. Dadurch wird die Zeit weitergezählt, auch wenn der MultiLogger nicht an das Stromnetz angeschlossen ist. Ich habe mich bewusst gegen die Verwendung einer Funkuhr entschieden. Erfahrungen mit der RTC haben gezeigt, dass diese Uhr recht genau geht, also selten nachgestellt werden muss. Zudem kann die Zeit aus ihr sofort beim Start ausgelesen werden. Eine Funkuhr benötigt einige Minuten, um sich zu synchronisieren. Manchmal gelingt dies aber nicht. Da die

Zeit auf einfache Weise vom PC aus gestellt werden kann, bietet die RTC hier eine bessere Alternative zur Funkuhr.

3.5 Analogeingänge

Zum Messen von acht analogen Spannungen verwende ich die ADCs (Analog Digital Converter) MCP3428 von Microchip. Diese mittels I2C auslesbaren Chips haben vier Kanäle, die jeweils 0V bis 2,048V messen können. Da ich aber Spannungen von $\pm 12V$ messen möchte, brauche ich noch eine Schaltung, die die gemessenen Spannungen entsprechend herabsetzt.

Dazu verwende ich eine Schaltung, die zwei Operationsverstärker pro Kanal benötigt. Der erste ist lediglich ein Impedanzwandler, der dafür sorgt, dass das Messergebnis nicht durch die eigentliche Spannungsquelle verfälscht wird. Es folgt ein invertierender Summierverstärker. Dieser addiert zu der gemessenen Spannung -15V. Damit wären Ausgangsspannungen von -30V bis 0V möglich. Der Aufbau der Schaltung bewirkt eine Invertierung und passende Widerstände teilen den Ausgang auf 0V bis 2,048V herunter. Diese Spannung kann nun von den ADCs gemessen werden.

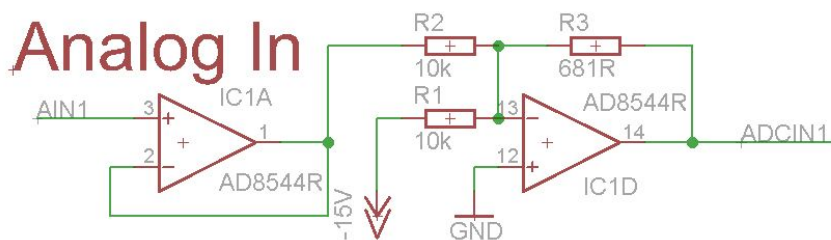


Abb. 5: Die Schaltung für einen Kanal. AIN1 ist Analogeingang, ACDIN1 Eingang des ADC

Theoretisch könnte der Logger sogar $\pm 15V$ messen. Die Operationsverstärker haben nur eine Betriebsspannung von $\pm 15V$. Bauteilbedingt ist aber eine Regelung der Spannung an den Rändern der Betriebsspannung nicht sauber möglich. Reale Messwerte sind daher nur bis etwa $\pm 12V$ möglich.

3.6 Digitaleingänge

Der MultiLogger kann acht digitale Signale messen. Dazu verwende ich den mittels I2C-Bus angebotenen Chip PCF8574. Die Schaltschwelle der Eingänge liegt bei etwa 4V, das heißt, ab einer anliegenden Spannung von 4V wird der Eingang als High erkannt. Die Eingänge sind vor Verpolung geschützt und bis 15V spannungsfest.

Die Eingänge sind nicht direkt mit dem PCF8574 verbunden, sondern durch einen Optokoppler getrennt. Optokoppler verwenden eine LED und einen Lichtsensor, um ein Signal weiterzugeben. Damit ist es unmöglich, durch einen falschen Eingang den Rest des MultiLoggers zu schädigen. Die eingesetzten Feldeffekttransistoren dienen als Konstantstromquelle für die LEDs. Hinzu kommen einige Dioden zur Verpolungssicherheit und zum Anheben der Schaltschwelle.

3.7 Temperatursensoren

Zur Messung der Temperatur verwende ich die von Dallas hergestellten Sensoren DS18B20. Diese verfügen über eine 1-Wire-Schnittstelle. Das ist ebenfalls ein Bussystem. Dadurch können alle Sensoren parallel geschaltet werden, weshalb auch nur ein Anschluss am Gerät benötigt wird.

Jeder Sensor hat eine eigene Adresse, die eine Bauteilkonstante ist und bei jedem Sensor unterschiedlich ist. Über diese Adresse kann dann ein bestimmter Sensor angesprochen werden.

Zurzeit sucht der MultiLogger noch nach den ersten vier Sensoren, die er auf dem Bus findet und liest die Temperaturen aus. Sind mehr als vier Sensoren angeschlossen, ist die Auswahl der ausgelesenen Sensoren nicht wählbar. Geplant ist jedoch, die Sensoren vom PC auswählen zu können. Die Einstellungen sollen ebenfalls gespeichert werden, sodass sie nicht nach jedem Neustart des Gerätes neu übertragen werden müssen.

3.8 Externe Sensoren im I2C-Bus

Der I2C-Bus, der bereits im Inneren des MultiLoggers verwendet wird, ist über Buchsen nach außen geführt. Dadurch ist es möglich, weitere Sensoren anzuschließen, die ursprünglich nicht vorgesehen waren.

Die Funktion ist noch nicht in den MultiLogger integriert. Im Folgenden erläutere ich das geplante Konzept, das auf Testaufbauten bereits erste Erfolge erzielt hat.

Für jeden Sensor, der angeschlossen wird, wird ein Mikrocontroller benötigt. Dieser übernimmt dann komplett das Auslesen des Sensors. Via I2C wird dann an den MultiLogger nur der Wert inklusive der Einheit und des Sensornamens weitergegeben. Dieser ist dann nur für das Anzeigen und Aufzeichnen verantwortlich. Das macht das System extrem flexibel, da jeder beliebige Sensor, der mit einem Mikrocontroller ausgelesen werden kann, so an den MultiLogger angeschlossen werden kann.

An den externen Geräten soll ein Schalter angebracht werden, mit denen die Adresse des Gerätes eingestellt werden kann. Der MultiLogger fragt dann vier Adressen für vier verschiedene Sensoren ab. Dabei ist egal, welche Sensoren das sind. Es können identische sein, aber auch vier verschiedene Sensoren für unterschiedliche Größen.

3.8.1 PeakTech 4000

Das PeakTech 4000 ist ein Tischmultimeter, das alle gängigen Größen der Elektronik messen kann, also Spannung, Strom, Widerstand, Kondensatoren etc. Zudem verfügt das Messgerät über eine serielle Schnittstelle. Damit ist es möglich, eine Verbindung zu anderen Geräten herzustellen.

Auf einem Testaufbau ist es bereits möglich, das Multimeter mit einem Mikrocontroller auszulesen und auf einem Display anzuzeigen. Es muss noch die I2C-Schnittstelle integriert werden.

Das Messgerät kann jedoch nicht direkt mit einem Controller verbunden werden, da beide Geräte mit verschiedenen Pegeln arbeiten. Das PeakTech verwendet

eine RS232, die $\pm 12V$ -Pegel verwendet, während der Controller 0V bzw. 5V verwendet. Daher wird ein weiterer Chip benötigt, der die Pegel angleicht. Ich verwende hier den MAX232.

3.8.2 Feinstaubsensor

Geplant ist auch die Anbindung eines Feinstaubensors. Dieser Sensor wird voraussichtlich das von Nova hergestellte Modul SDS011 sein. Dieser „arbeitet nach dem Prinzip des Streulichtverfahrens und kann die Partikelkonzentration zwischen 0,3 und 10 μ m in der Luft messen.“ (<http://www.watterott.com/de/Nova-SDS011-Feinstaub-Sensor>, 26.01.18, Watterott electronic GmbH, Nova SDS011 Feinstaub-Sensor-Modul). Das Modul verfügt genauso wie das Tischmultimeter über eine serielle Schnittstelle, jedoch wird hier keine Pegelanpassung benötigt.

Ich habe mich für einen Feinstaubsensor entschieden, da zurzeit das Thema Feinstaub relativ aktuell ist. Das Gerät ermöglicht dann, auf einfache Weise selbst Feinstaubmessungen durchzuführen. Man kann so offizielle Angaben kontrollieren. Ist gerade am Wohnort keine Messstation in der Nähe, so kann man dennoch die Feinstaubbelastung ermitteln.

4. Programmierung

Zur Programmierung des MultiLoggers verwende ich die Arduino IDE. Ich musste einige Erweiterungen für die Umgebung installieren, dazu zählen Boards und Libraries.

4.1 Boards

Zur Programmierung des ATmega1284 wird ein Boardpaket benötigt, welches viele gängige AVR's von Atmel hinzufügt. Ich verwende hier das Paket „Platino“ von Elektor Labs. Es kann einfach mit dem Boards Manager installiert werden. Wie das genau geht, ist auf der Internetseite beschrieben.

Außerdem verwende ich keinen Bootloader, sodass ich den Controller über die ISP-Schnittstelle programmieren muss. Dazu verwende ich das von Ulrich Radig vorgestellte Programmiergerät „USBasp“. Dieses ist bereits in der Arduino IDE installiert. Da Platino eigentlich einen Bootloader verwendet, muss unter „Sketch“ → „Hochladen mit Programmer“ ausgewählt werden.

Auch das WLAN-Modul ESP8266 habe ich mit der Arduino IDE programmiert. Eine Anleitung, wie das Board in der Umgebung installiert wird, habe ich im PDF-Dokument des IoT-Hackathons des IT-Gipfels 2016 gefunden. Ich verwende zwar ein anderes Modul, ein entsprechendes Board ist aber trotzdem verfügbar.

4.2 Libraries

Zur Programmierung werden auch einige Libraries benötigt. Diese vereinfachen es, bestimmte Sensoren auszulesen. Ich verwende zum Einen Libraries, die bereits in der Arduino IDE eingebunden sind, aber auch solche, die man aus dem Internet herunterladen kann. Libraries werden unter anderem benötigt für die Temperatursensoren und die Echtzeituhr. Ein genauer Nachweis der Libraries findet sich im Abschnitt „5. Quellennachweis“. Verschiedene Libraries habe ich selbst geschrieben oder aus Beispielen anderer Libraries kopiert.

4.3 PC

Das Programm für den PC habe ich in der Programmierumgebung Delphi geschrieben. Auf dem PC wird eine Installation von Microsoft Excel benötigt. Das Programm öffnet selbstständig Excel und legt eine Datei an, in der die Messwerte gespeichert werden. Außerdem werden die Daten in Diagrammen grafisch dargestellt. Die Diagramme bauen sich auf dem Bildschirm live auf, sodass man den Verlauf einer Größe mit verfolgen kann, ohne die Messung unterbrechen zu müssen. Dazu greift das Programm direkt auf einzelne Zellen zu und verändert deren Wert. Zudem werden die Datenbereiche der Diagramme angepasst.

5. Bewertung

Ein Kritikpunkt an dem System ist jedoch, dass das Aufnehmen von Messwerten einige Zeit in Anspruch nimmt. Eine höhere Geschwindigkeit als ein Messwert pro Sekunde ist damit nicht zu erreichen. Ein damit verbundenes Problem ist, dass man nur Langzeitbeobachtungen durchführen kann, nicht jedoch eine schnelle Messung, die eine sich schnell verändernde Größe protokolliert. Das verkleinert gerade im Bereich der Schule das Einsatzgebiet etwas.

Dennoch bin ich der Meinung, es ist gelungen, ein System zu entwerfen, mit dem es möglich ist, auf einfache Weise viele Messgrößen gleichzeitig zu messen und aufzuzeichnen. Vor allem die hohe Flexibilität und einfache Erweiterbarkeit des Systems mithilfe der I2C-Geräte machen den Logger zu einem guten Hilfsmittel zum Aufnehmen und Auswerten von Daten über längere Zeiträume. Die Einsatzmöglichkeiten des Systems sind damit nahezu unbegrenzt.

Die zu Anfang gestellten Fragen lassen sich also nun beantworten: Ja, es ist möglich, mit einfachen Mitteln ein gut funktionierendes, leicht anzupassendes und zu erweiterndes Mess- und Aufzeichnungssystem zu entwickeln. Auch ist es gelungen, ein Konzept zu entwickeln, bei dem es möglich ist, einfach neue Sensoren hinzuzufügen, ohne das Hauptgerät verändern zu müssen. Damit habe ich meine Ziele größtenteils erreicht.

6. Danksagung

An dieser Stelle möchte ich mich ganz herzlich bei meinem Vater, Andreas Bastian, bedanken. Er hat mich im Bereich der Hardware beim Entwickeln der Grundsaltungen und dem Ätzen der Platine unterstützt. Zudem wirkte er beim Bau der Gehäuse mit. Außerdem ist zu sagen, dass er viele gute Ideen für weitere Messbereiche beigesteuert hat.

7. Quellennachweis

- 1) <https://playground.arduino.cc/Learning/OneWire>, 30.1.2018, GitHub Inc., Library für Temperatursensoren
- 2) <https://github.com/Christensen/DS3232RTC>, 30.1.2018, GitHub Inc., Library für die RTC
- 3) <https://github.com/uChip/MCP342X>, 30.1.2018, GitHub Inc., Library für die Analog Digital Konverter

- 4) <http://www.watterott.com/de/Nova-SDS011-Feinstaub-Sensor>, 30.1.2018,
Watterott electronic GmbH, Feinstaub-Sensor-Modul
- 5) http://deutschland-intelligent-vernetzt.org/app/uploads/sites/4/2016/09/loT_Hackathon_C_Code_Version.pdf,
30.01.2018, Nationaler IT-Gipfel, Beschreibung ESP8266
- 6) <https://www.arduino.cc/en/Main/Software>, 30.1.2018, Arduino,
Programmierumgebung des MultiLoggers
- 7) <https://github.com/ElektorLabs/Arduino>, 30.1.2018, GitHub Inc., Boardpaket
für den MultiLogger