

Calliope Mini V3

DATALOGGER

DIE ERWEITERUNG

GERD RÜHLE

MIT RAT & TAT DIGKOMPE.V.



ÄNDERUNGSHISTORIE

Version	Datum	Änderer	Änderung
00	26.04.2025	Gerd Rühle	Initiale Version
00-1	08.05.2025	Gerd Rühle	Hinzufügen Kapitel Event ,bei voller Logdatei‘

INHALTSVERZEICHNIS

Änderungshistorie.....	2
Inhaltsverzeichnis.....	3
1 Über dieses Dokument.....	4
1.1 Motivation.....	4
1.2 Nutzungsrechte.....	5
2 Der DataLogger.....	6
2.1 Die Blocks des DataLoggers.....	6
2.2 Anzeige der Einträge im DataLogger.....	8
2.3 Data Simulator.....	9
2.4 Reihenfolge der Datenelemente.....	10
3 Einfaches Szenario mit DataLogger.....	16
3.1 Das Szenario.....	16
3.2 Der Sensor.....	16
3.3 Das Programm.....	16
3.4 Die Daten.....	19
3.5 Auswertung der Daten.....	22
4 DataLogger - Die Grenzen austesten.....	25
4.1 Motivation.....	25
4.2 Testszenarien.....	25
4.3 Achtung: deutsche Umlaute.....	27
4.4 Szenario 1: 100 Zeichen in einem Text mit 100 Zeichen.....	28
4.5 Szenario 2: 100 Zeichen in zwei Texte mit je 50 Zeichen.....	28
4.6 Szenario 3: 100 Zeichen in fünf Texte mit je 20 Zeichen.....	30
4.7 Szenario 4: 5 Zahlen, zwei Integer, drei Nachkommazahlen.....	31
5 Event ,bei voller Logdtei'.....	32
6 Details des Calliope mini:.....	38

1 ÜBER DIESES DOKUMENT

1.1 Motivation

Mit Version 3 des Calliope mini können Daten einfach und mit Bordmitteln aufgezeichnet, als csv-Datei heruntergeladen und weiterverarbeitet werden. Damit werden Projekte möglich bei denen Beobachtungswerte der Sensoren zeitgenau erhoben werden ohne stets den Calliope mini beobachten zu müssen, um die angezeigten Werte zu notieren.

Das Dokument besteht aus drei Teilen. Im ersten Teil werden die Elemente des DataLoggers untersucht und in Beispielprogrammen das Ergebnis aufgezeigt. Zuerst werden die Blöcke so genutzt wie vorgesehen. Dann kommt die Neugier zu Tage und es werden Szenarien aufbereitet in denen von der angedachten Vorgehensweise abgewichen wird. Auch hier liefert der Datalogger die zu erwartenden Ergebnisse.

Nachdem jetzt bekannt ist, dass der DataLogger zuverlässig arbeitet kam das Verlangen auf den DataLogger im Einsatz zu testen. Gewählt wurde ein Szenario in dem die Datenaufzeichnung im Mittelpunkt steht. Das eigentliche Experiment beschränkt sich darauf einen Sensor anzuschliessen, die gelieferten Daten über einen längeren Zeitraum aufzuzeichnen.

Und dann kam die Frage auf: Wo sind die Grenzen des DataLoggers? Wieviele Daten kann er aufnehmen und was kann man daraus für Szenarien / Projekte ableiten. Anhand des abgebildeten Szenarios mit vier aufgezeichneten numerischen Variablen je Beobachtung ergeben sich 3.800 gespeicherten Beobachtungen/Sätze. Daraus ergeben sich je nach der Zeit zwischen zwei Beobachtungen ein Experimentzeitraum von 26 bis 158 Tagen.

Anzahl Beobachtungen insgesamt	3800	3800	3800	3800
Zeitraum zwischen Beobachtungen in Minuten	10	15	30	60
Anzahl Beobachtungen je Tag	144	96	48	24
Anzahl Tage	26	40	79	158

Diese langen Zeiträume lenken den Blick auch auf die Energieversorgung und -verbrauch während des Experiments. Für die Energieversorgung ist eine ausreichende, zuverlässige Quelle notwendig. Für die Reduzierung des Energieverbrauchs kommt die Erweiterung „Power“ in Betracht. Beides wird in einem gesonderten Dokument untersucht.

Untersucht werden

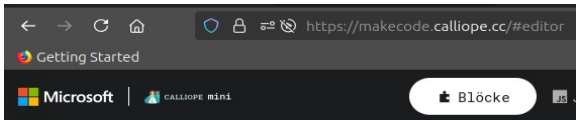
- die Blöcke der Erweiterung „DataLogger“, der Schreibvorgang mit und ohne des Blocks „set column“
- ein einfaches Beispiel unter Nutzung des SD-40: Temperatur, Luftfeuchtigkeit, CO2-Gehalt
- die Masse der Daten, die der DataLogger aufnehmen kann.

1.2 Nutzungsrechte



Die Texte, Bilder und Grafiken dieser Website stehen, soweit nicht anders angegeben, unter der Creative Commons-Lizenz CC BY-SA 4.0 (Creative-Commons-Lizenz Namensnennung, Weitergabe unter gleichen Bedingungen, Version 4.0), zu finden unter <https://creativecommons.org/licenses/by-sa/4.0/deed.de>.

Autor: Mit Rat & Tat DigKomp e.V. - G. Rühle- Juni 2025 <http://www.mrut.online>



Die Screenshots der verwendeten Programme sowie die Befehlsblöcke des DataLoggers sind dem Editor MakeCode entnommen.



Die Screenshots der Datensichten sind dem DataLogger des Calliope mini entnommen.



Auswertungen sind mit Libre Office Calc erstellt.

2 DER DATALOGGER

Informationen zum DataLogger sind auf der Calliope Webseite hier zu erreichen:
<https://makecode.calliope.cc/reference/datalogger>

2.1 Die Blocks des DataLoggers

Blocks, die typischerweise am Anfang festgelegt werden

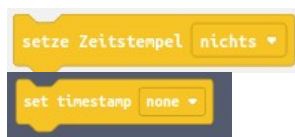


Definiert Namen und Reihenfolge der Datenelemente in der Log-Datei für die zukünftige Datenprotokollierung.

Datenwerte, die in das Datenprotokoll geschrieben werden, werden einer *Spalte* zugewiesen, um zu ermitteln, mit welchem Wert sie in Verbindung stehen. Bevor ein Datenwert protokolliert wird, wird er als “CV” oder “Spaltenwert” formatiert. Auf diese Weise wird ein Spaltenname an einen Datenwert angehängt.

Ein Datenprotokolleintrag wird als ein Array mit einem oder mehreren “Spaltenwert”-Daten in das Datenprotokoll geschrieben. Diese Funktion erzeugt das Datenelement für den Wert, der in einen Logeintrag eingefügt werden soll.

Es können bis zu 10 Spalten definiert werden.



Format für den Zeitstempel festlegen: keiner, Millisekunden, Sekunden, Minuten, Stunden, Tage

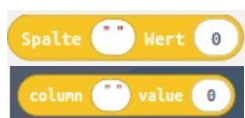
Ein Zeitstempelwert wird als eines der Datenelemente in einem Logeintrag berücksichtigt, wenn der Zeitstempel nicht „keiner“ ist. Standardmäßig wird der Wert „Sekunden“ gesetzt.



Legt fest, ob die Daten zusätzlich seriell gespiegelt werden.

Standardmäßig werden die gleichen Daten, die in das Datenprotokoll geschrieben werden, auch an den seriellen Ausgabe-Port gesendet.

Blocks, die typischerweise im Programmablauf mehrfach durchlaufen werden



Legt den Wert für die ausgewählte Spalte fest, die beim nächsten Schreibvorgang im DataLogger gespeichert werden soll.



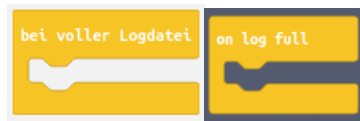
Daten im DataLogger speichern

Schreibt den Satz mit den angegebenen Elementen in die Log-Datei. Ist das Element in der Log-Datei nicht angelegt wird es implizit definiert.

Datenprotokolleinträge werden als Array Datenelementen von “Spalte” und “Wert” in das Log geschrieben. Jeder Datenwert, der im Log aufgenommen werden soll, hat eine Spalte, zu der dieser gehört. Der Datenwert wird an die Spalte angehängt, indem ein “Spaltenwert” Objekt mit dem Datenwert mit einem Spaltennamen erstellt wird.

Ein Logeintrag besteht aus einem oder mehreren “Spaltenwert”-Objekten, die in ein Array eingefügt werden. Dieses Array wird an das Datenprotokoll gesendet, um es als nächsten Logeintrag zu schreiben.

Blocks, der mitteilt, dass der Speicher des DataLoggers vollgeschrieben ist.



Beschreibt was ausgeführt werden soll, wenn der Speicher voll ist. Wenn das Ereignis „Log voll“ ausgelöst wird, wird dieser Code ausgeführt.

Block zum Löschen des DataLoggers



Löschen aller Einträge des DataLoggers inklusive der Spaltenüberschriften.

Wenn das Datenprotokoll voll ist oder du dich entscheidest, die Aufzeichnung von vorne zu beginnen, kannst du den Inhalt des Protokolls löschen. Es gibt zwei Methoden das Log zu löschen.

schnell Methode, um einfach am Anfang des Logs erneut zu starten und die vorhandenen Logeinträge zu überschreiben. Dies entspricht dem aus der IT bekannten logischen Löschen: die „alten Daten“ sind noch vorhanden, lediglich als gelöscht gekennzeichnet und man kann mit den Mitteln des DataLoggers nicht mehr auf sie zugreifen.

voll Methode, um das Log zuerst zu löschen, bevor erneut in das Log geschrieben wird. Dies entspricht dem aus der IT bekannten physischen Löschen: die „alten Daten“ werden mit Initialwerten überschrieben, sind nicht mehr vorhanden.

Standardmäßig ist die schnell Methode eingestellt.

2.2 Anzeige der Einträge im DataLogger

Die geschriebenen Sätze werden auf dem Calliope mini in der Datei „MY_DATA.HTM“ gespeichert. Um die Daten zu lesen, muss der Calliope mini mit dem Rechner verbunden sein. Er erscheint dort als eigenes Laufwerk mit einem USB-Symbol. Durch Doppelklick auf das Laufwerk werden die Dateien im Datenmanager (Explorer, Files, Finder) angezeigt:

 CALLIOPE.HTM	260 bytes	22 Mär 2016
 DETAILS.TXT	550 bytes	22 Mär 2016
 MY_DATA.HTM	127,0 kB	22 Mär 2016

Die Daten können im Datenmanager angesehen werden. Durch Doppelklick auf die Datei MY_DATA.HTM öffnet sich ein Browserfenster, das die Daten präsentiert.

In der Datei befinden sich neben den Daten Steuerungsinformationen sowie die HTML-Daten zur Darstellung und Funktion der Webseite:



The screenshot shows the 'Calliope mini Datalog' web interface. At the top is the Calliope logo. Below it are five buttons: 'Herunterladen', 'Kopieren', 'Aktualisieren...', 'Log leeren...', and 'Visuelle Vorschau'. A paragraph of text explains that the data is on the Calliope mini and can be analyzed or downloaded as a CSV file. Below the text is a table with three columns: 'Time (seconds)', 'Zeile', and 'Testtext'.

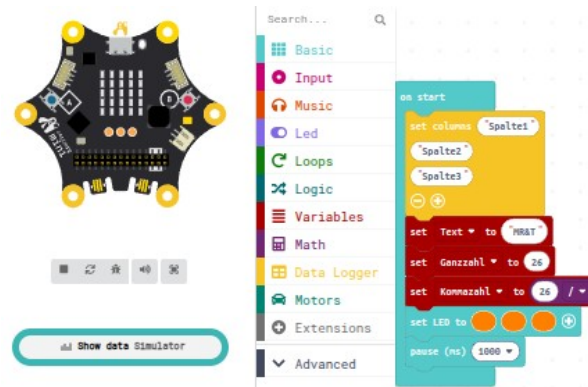
Hier die Seite mit einem leeren Log. Es werden nur die mit „set columns“ definierten Datenelemente angezeigt.

Die Buttons:

- **Herunterladen:** die Daten werden in eine csv-Datei mit Namen „Calliope mini.csv“ ins Downloadverzeichnis des Rechners heruntergeladen mit dem der Calliope mini verbunden ist. Mit diesen Daten können weitere Verarbeitungen/Auswertungen vorgenommen werden z.B. in einem Spreadsheet wie Libre Office Calc, numbers oder Excel.
- **Kopieren:** kopiert die Daten in die Zwischenablage.
- **Aktualisieren:** werden während des Betrachtens der Daten die Daten fortgeschrieben, können diese mit diesem Knopf neu geladen werden. Dafür ist ein abziehen und Wiedereinstecken des USB-Anschlusses des Calliope mini erforderlich.
Achtung: um das Programm weiterlaufen zu lassen muss die Batterie angeschlossen und an sein.
- **Log leeren:** die Sätze in der Log-Datei werden gelöscht. Die Datei ist leer.
- **Visuelle Vorschau:** es erfolgt eine Aufbereitung der gespeicherten Daten

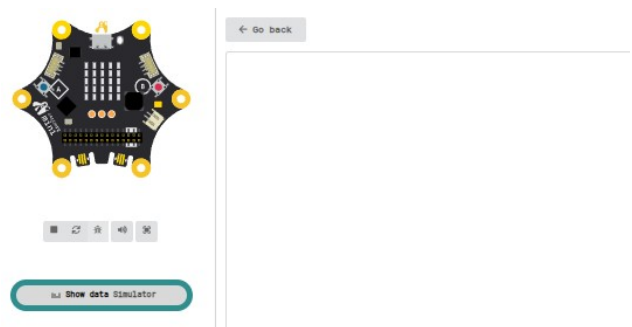
2.3 Data Simulator

Die vom Programm geschriebenen Daten lassen sich im Data Simulator ansehen. Dabei kann man auch sehen wie die Daten sukzessive in die Log-Datei geschrieben werden. Der Data Simulator wird über den Button „Show data Simulator“ aufgerufen.



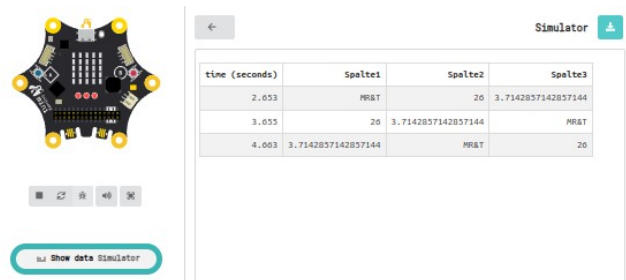
Danach öffnet sich rechts ein Fenster das die vorhandenen Daten zeigt.

Im Beispiel hier wurden bisher keine Daten geschrieben. Deshalb ist die Anzeige des Simulators zunächst leer.



Beim Starten des Programms im Simulator wird im Data Simulator die geschriebenen Datensätze angezeigt.

Die Datensätze können über den grünen Download-Button rechts oben heruntergeladen werden.



2.4 Reihenfolge der Datenelemente

Das angedachte Vorgehen zur Nutzung des DataLoggers ist es einmalig beim Start im „on start“-Block die im weiteren Verlauf verwendeten Datenelemente durch Festlegung des Names sowie deren Reihenfolgen zu definieren. Im weiteren Programmablauf werden die Datenelemente mit dem „column » « value“-Block wie erforderlich gefüllt und anschliessend mit dem „log data“-Block in den DataLogger geschrieben.

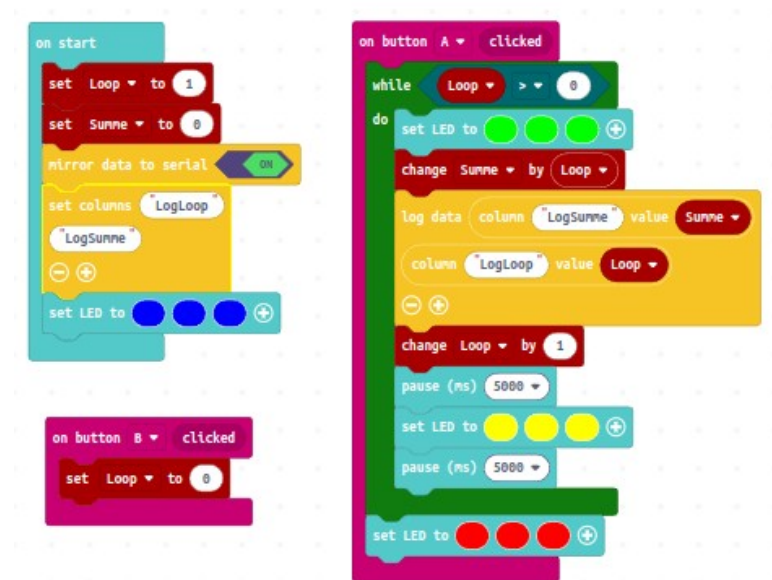
Und natürlich reizt es zu wissen was passiert, wenn vom gewohnten Pfad abgewichen wird. Die typisch denkbaren Fälle sind nachfolgend ausgetestet.

2.4.1 So sollte es sein: Mit dem „set column“-Block die Datenelemente festlegen

Das angedachte Verfahren ist hier umgesetzt: Im „On start“-Block werden in der Column-Definition zwei Datenelemente in der Reihenfolge „LogLoop“ und „LogSumme“ festgelegt. Beim Schreibvorgang werden die Datenelemente in der umgekehrten Reihenfolge angegeben. Die Daten werden im Satz in der Reihenfolge wie in der Definition festgelegt angelegt.

Ergebnis: die Datenelemente werden gemäss der im „set columns“-Block vorgegebenen Reihenfolge geschrieben, egal wie die Reihenfolge im „log data“-Block angegeben ist.

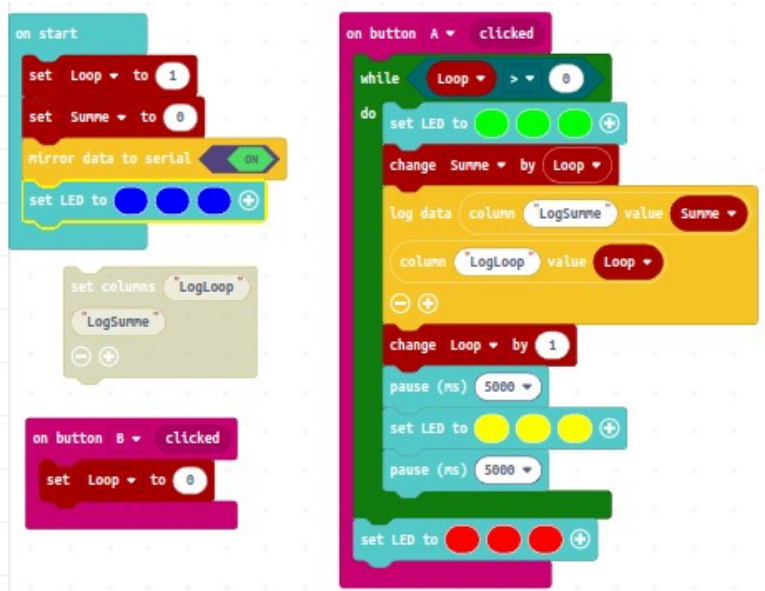
Test-Programm und Output:



time (seconds)	LogLoop	LogSumme
1.452	1	1
11.455	2	3
21.457	3	6
31.459	4	10
41.461	5	15
51.464	6	21

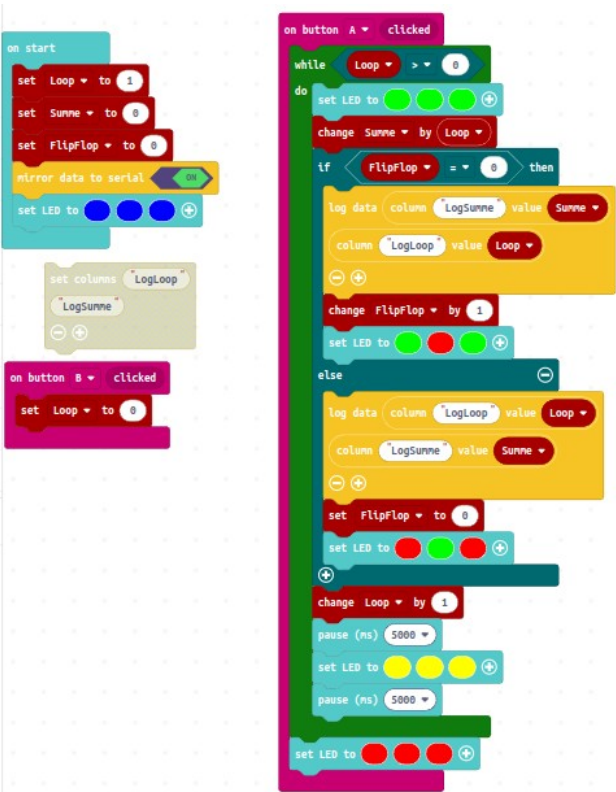
2.4.2 Test: Daten schreiben ohne Nutzen des „set column“-Blocks

Fehlt die Festlegung der Namen und Reihenfolge der Datenelemente mittels des „set column“-Blocks, werden die Elemente gemäss der Reihenfolge im DataLogger implizit angelegt in der sie im „log data“-Block geschrieben werden. In diesem Beispiel wird in einem Schreibvorgang zuerst das Element „LogSumme“ geschrieben, danach das Element „LogLoop“.



time (seconds)	LogSumme	LogLoop
1.278	1	1
11.281	3	2
21.283	6	3
31.285	10	4
41.288	15	5
51.29	21	6

Dies gilt auch wenn in unterschiedlichen Reihenfolge die Datensätze geschrieben werden. Die Reihenfolge der Elemente im DataLogger wird durch den ersten „log data“-Block festgelegt

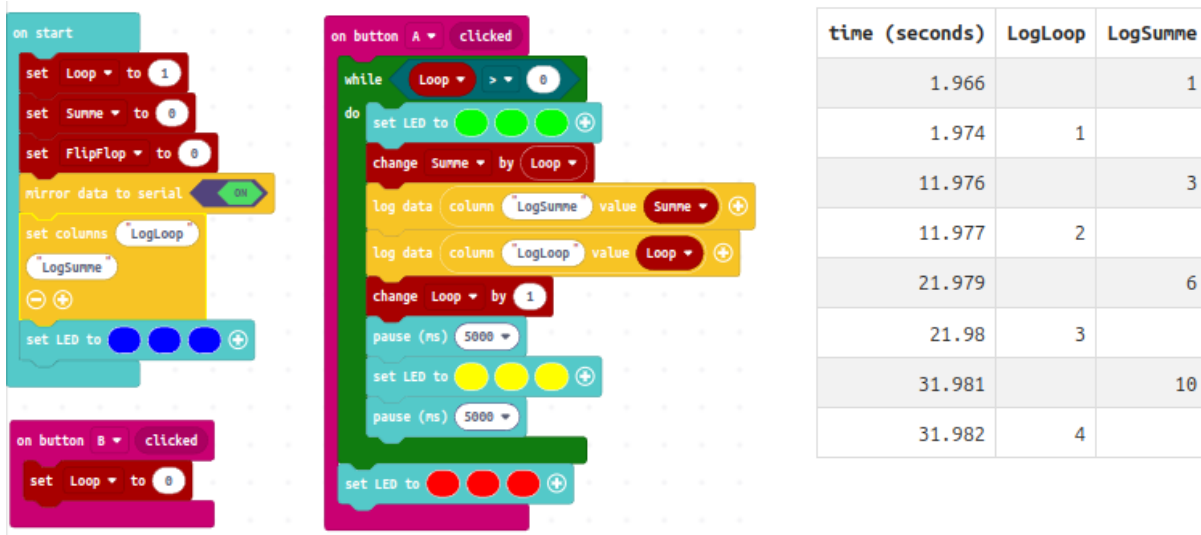


time (seconds)	LogSumme	LogLoop
1.17	1	1
11.173	3	2
21.175	6	3
31.178	10	4
41.18	15	5
51.182	21	6

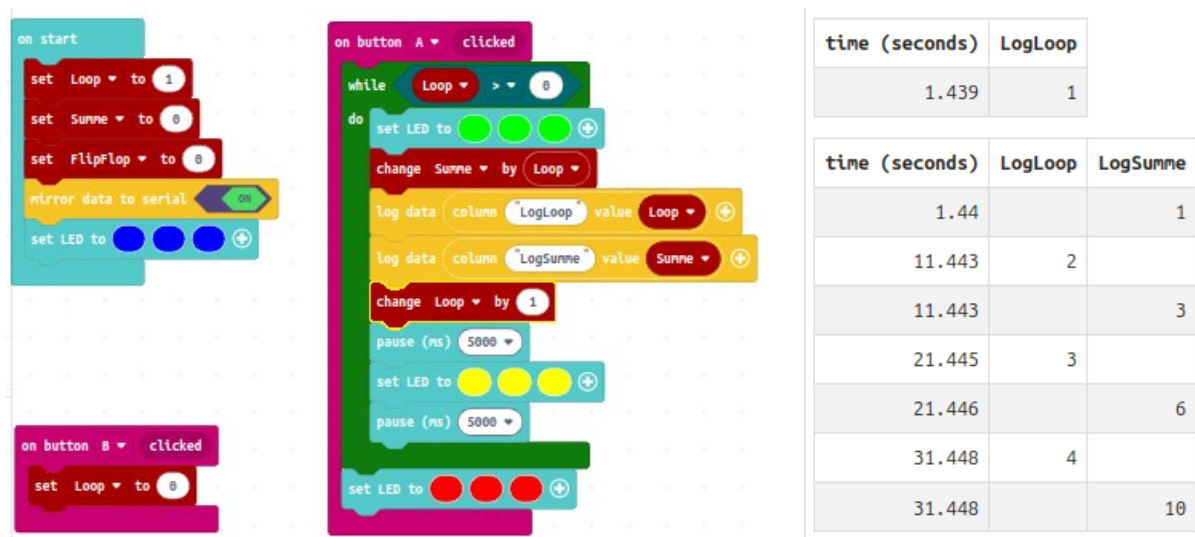
2.4.3 Test: Schreiben einzelner Datenelemente

Werden nur ausgewählte Datenelemente mit einem „log data“-Block geschrieben, werden auch nur diese angegebenen Elemente in einen Datensatz geschrieben. Die nicht verwendeten Datenelemente bleiben leer.

Die Datenelemente werden im „on start“-Block über „set column“-Block explizit definiert und die Reihenfolge festgelegt. Danach wird je eines der beiden Datenelemente mittels „log data“-Block geschrieben. Beide Datenelemente sind beim ersten Schreibvorgang angelegt und werden ausgewiesen.



Werden die Datenelemente indirekt über einen Schreibvorgang mittels „log data“ definiert, ist im ersten Schreibvorgang nur ein Datenelement angezeigt. Das zweite ergänzt sich beim ersten Schreibvorgang des Datenelements.



Im Beispiel werden Sätze mit abwechselnd gefüllten Daten beginnend mit dem Datenelement „LogLoop“. Deshalb ist in der ersten Zeile nur ein Element ausgewiesen. Im zweiten Satz wird die Datensatzdefinition, um den Wert „LogSumme“ erweitert und danach alternierend geschrieben.

Für diesen Output wurden die beiden „log data“-Blöcke vertauscht.

Im ersten Schreibvorgang wird das Datenelement „LogSumme“ implizit angelegt und der erste Datensatz mit einem Datenelement geschrieben.

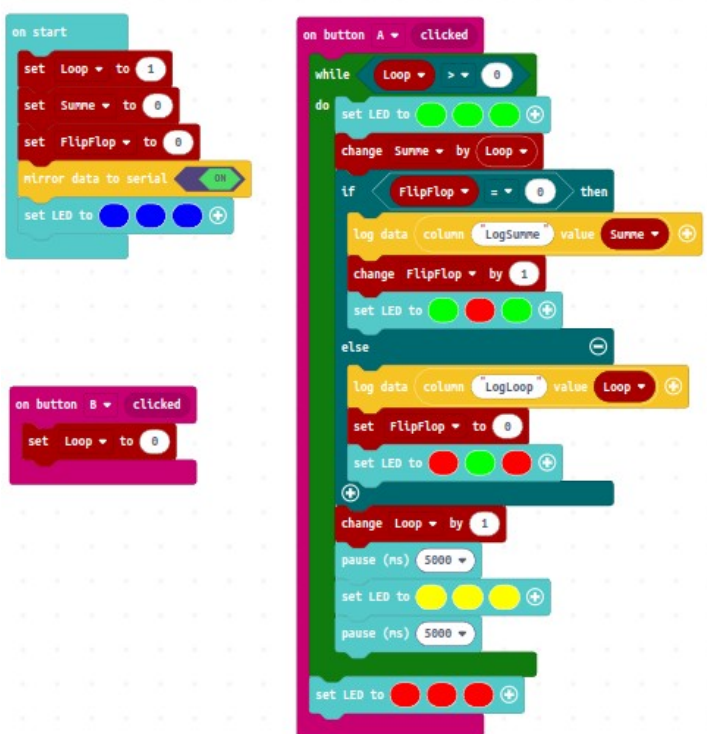
Im zweiten Schreibvorgang unmittelbar nach dem ersten wird das Datenelement „LogLoop“ implizit angelegt und der zweite Datensatz mit einem leeren Eintrag in „LogSumme“ und dem Wert in „LogLoop“ geschrieben.

In dem weiteren Durchlauf der Schleife erfolgen die Schreibvorgänge alternierend.

time (seconds)	LogSumme
1.488	1

time (seconds)	LogSumme	LogLoop
1.489		1
11.491	3	
11.492		2
21.493	6	
21.494		3
31.496	10	
31.497		4

Das selbe Ergebnis ergibt sich, wenn die Schreibvorgänge nicht unmittelbar hintereinander erfolgen sondern zeitlich getrennt sind.



Hier das Ergebnis mit der Variablen FlipFlop = 0

time (seconds)	LogSumme
1.121	1

time (seconds)	LogSumme	LogLoop
11.124		2
21.127	6	
31.13		4
41.132	15	
51.135		6

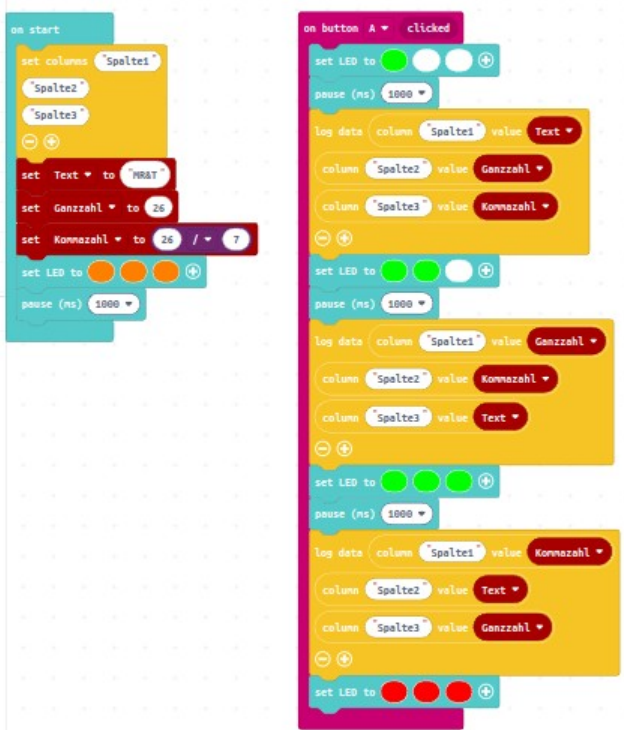
Hier das Ergebnis mit der Variablen FlipFlop = 1

time (seconds)	LogLoop
1.012	1

time (seconds)	LogLoop	LogSumme
11.015		3
21.018	3	
31.021		10
41.023	5	
51.025		21
61.028	7	

2.4.4 Test: Datentypen

Mit definieren eines Datenelemente oder erstes Schreiben eines Datenelements wird kein Datentyp festgelegt. In das selbe Datenelement können bei verschiedenen Schreibvorgängen unterschiedliche Datentypen geschrieben werden.

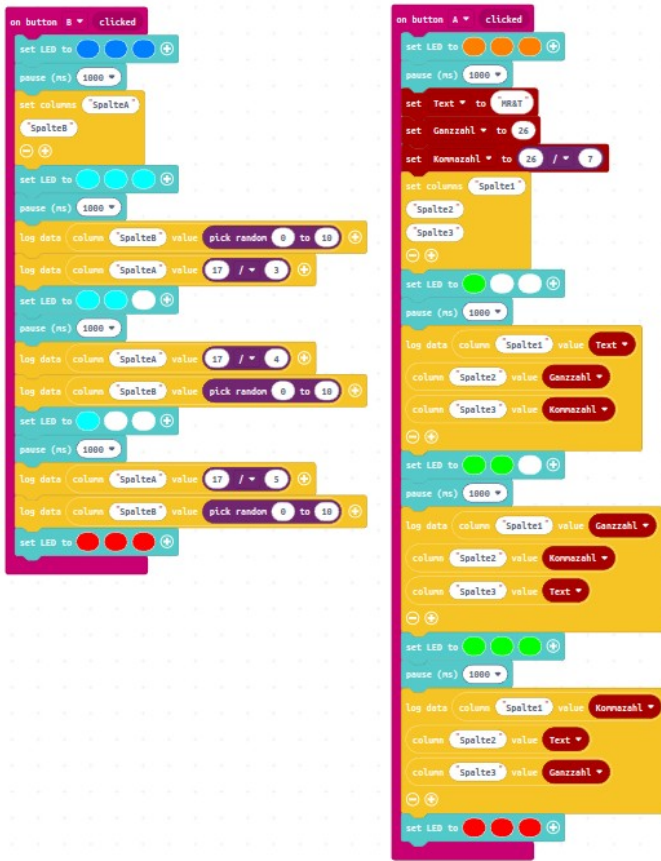


Hier werden drei Datenelemente explizit festgelegt sowie drei Variablen, je eine als Text, Ganzzahl oder Kommazahl.

Anschliessend werden Sätze geschrieben in denen die Datentypen in den einzelnen Datenelementen unterschiedlich sind.

time (seconds)	Spalte1	Spalte2	Spalte3
3.297	MR&T	26	3.7142857142857144
4.299	26	3.7142857142857144	MR&T
5.381	3.7142857142857144	MR&T	26

2.4.5 Test: mehrere explizite Datendefinition



Werden mehrere „set column“-Blöcke verwendet werden die neu angelegten Definitionen an die bereits bestehenden „angehängt“ und beim Schreiben angezeigt.

Zuerst Drücken Button A, danach Button B

time (seconds)	Spalte1	Spalte2	Spalte3
5.102	MR&T	26	3.7142857142857144
6.103	26	3.7142857142857144	MR&T
7.100	3.7142857142857144	MR&T	26

time (seconds)	Spalte1	Spalte2	Spalte3	SpalteA	SpalteB
13.132					0
13.133			3.0000000000000007		
14.141				4.25	
14.141					5
15.143				3.4	
15.143					7

Zuerst Drücken Button B, danach Button A

time (seconds)	SpalteA	SpalteB
4.044		3
4.045	3.0000000000000007	
5.047	4.25	
5.047		8
6.053	3.4	
6.054		8

time (seconds)	SpalteA	SpalteB	Spalte1	Spalte2	Spalte3
26.324			MR&T	26	3.7142857142857144
27.320			26	3.7142857142857144	MR&T
28.33			3.7142857142857144	MR&T	26

3 EINFACHES SZENARIO MIT DATALOGGER

Das gewählte Szenario wurde ausgewählt, weil es ohne grosse technische Vorbereitung durchgeführt werden kann. Es bedarf lediglich eines externen Sensors und das unter aufgeführten Programms.

3.1 Das Szenario

Ermittle die Werte Luftfeuchtigkeit, Temperatur, CO₂-Gehalt in einem festen Zeitintervall über einen längeren Zeitraum und halte die gemessenen Werte im DataLogger fest, um sie später auswerten zu können.

3.2 Der Sensor

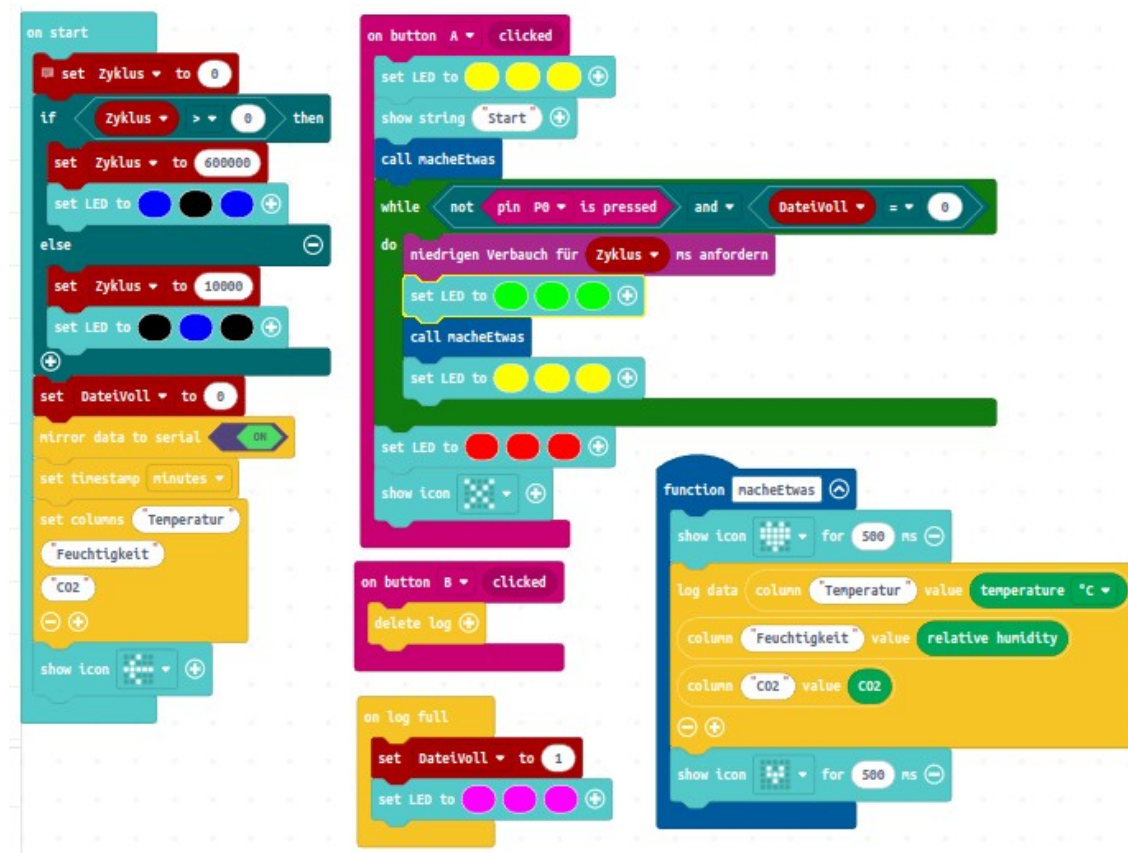
Zum Ermitteln der Werte Luftfeuchtigkeit, Temperatur und CO₂ wird ein SCD40-Sensor genutzt, der über die A0-Schnittstelle angeschlossen werden kann. Das Auslesen Werte erfolgt über eine eigene Erweiterung mit Namen „CO₂-Sensor SCD40“ eingebunden. In der Erweiterung werden drei Reporter „relative Luftfeuchtigkeit“, „Temperatur“, „CO₂“ bereitgestellt mit denen die gemessenen Werte im Calliope mini abgefragt werden können. Die Erweiterung ist aktuell nur für MakeCode Editor verfügbar.

Der SCD40 kann z.B. direkt bei Calliope.cc bestellt werden: <https://webshop.calliope.cc/calliope-mini-co2-sensor>

Weitere Infos zum SCD40 gibt es hier: <https://calliope.cc/calliope-mini/erweiterungen/calliope-co2-sensor>

3.3 Das Programm

Das hierfür verwendete Programm in der Version1:



Über die Variable „Zyklus“ kann die Länge des Zeitintervalls zwischen der Erhebung und Speicherung der Messdaten festgelegt werden.

Hier ist zusätzlich die Erweiterung „Power“ eingebunden. Mit ihr kann der Calliope mini für einen vorgegebenes Zeitintervall – hier die Zeit zwischen den Messungen – in einen Zustand versetzt werden, der Energie spart.

Der SCD40-Sensor ist dauerhaft mit Strom versorgt. Dies ist an der weissen LED auf dem SCD40 erkennbar.

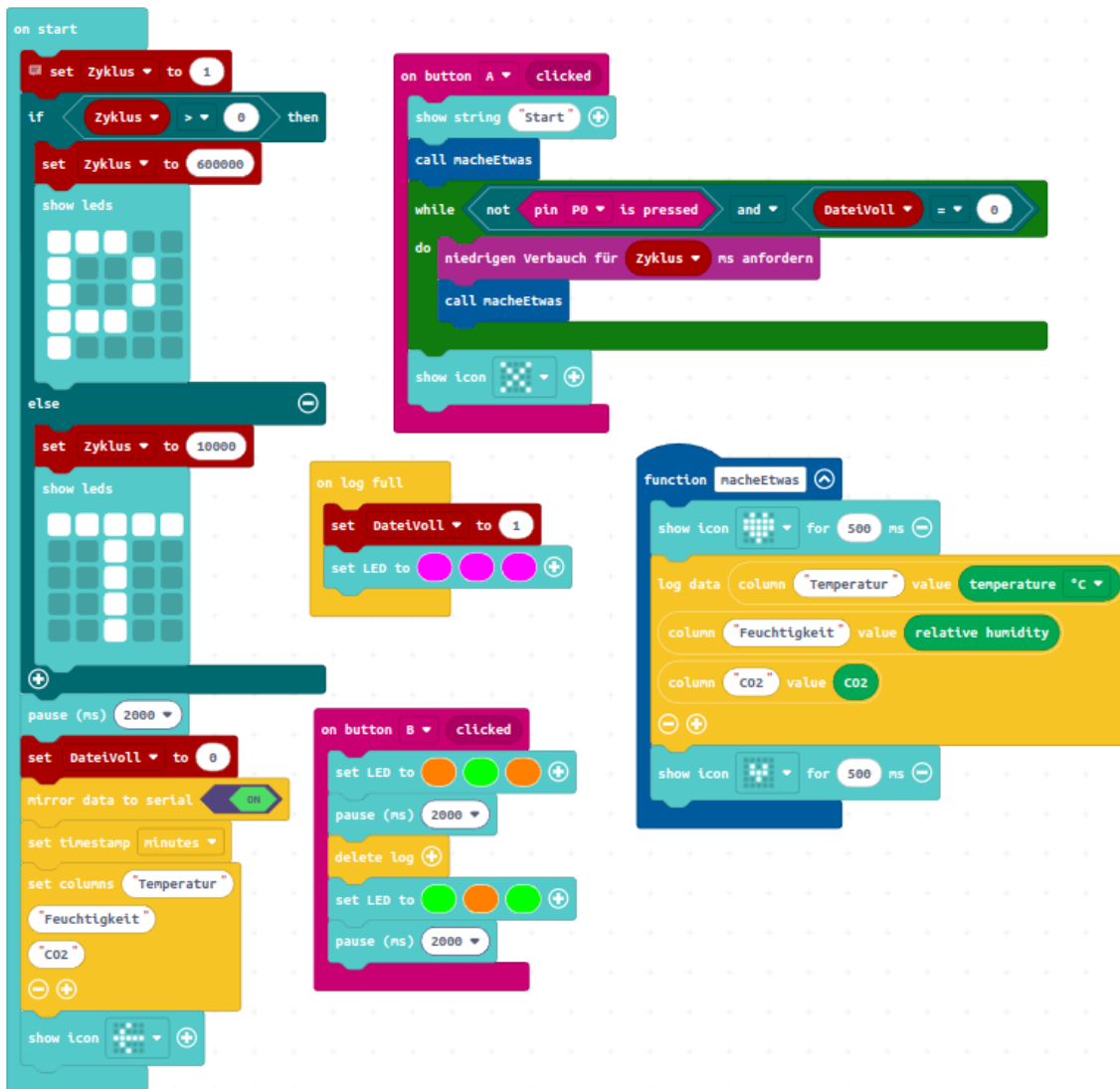
Bleibt die Frage wie kann man dem Programm mitteilen, dass die Verarbeitungen beendet werden soll. Solange das Programm im Schlafmodus ist, nimmt es keine Events wie z.B. Button A oder B gedrückt wahr. Deshalb wurde hier der Weg über „pin P0 ist gedrückt“ gewählt. Dies kann man erzeugen indem man ein Kabel zwischen Pin 0 und dem Minus-Pol klemmt. Der Zustand „ist Pin P0 gedrückt“ wird in der Verarbeitungsschleife als Abbruchkriterium verwendet.

Das Programm funktioniert, allerdings hat sich das ständig leuchtende LED hat sich als störend erwiesen, wenn nachts im Raum gemessen wird in dem auch geschlafen wird.

Deshalb diese neue Version ohne Nutzung der LEDs. Dass das Programm arbeitet sieht man jetzt nur noch, wenn die Daten geschrieben werden durch das kurzzeitige Aufleuchten des grossen und kleinen Herzens während des Schreibvorgangs. , nur Anzeige über Display.

Mit der Variable „Zyklus“ kann festgelegt werden, ob das Programm in einem Testmodus (kürzere Zeiten – wird beim Programmstart mit einem T auf dem 5x5-Display angezeigt) oder Produktionsmodus (längere Zeiten – wird beim Programmstart mit einem P auf dem 5x5-Display angezeigt) läuft. Das Messintervall zwischen der Erhebung wird durch die Variable „Zyklus“ bestimmt.

Programm: mini-ProdPower-Finale.hex



Bei einem ersten Versuchslauf wurde ein Powerpack über den USB-Anschluss angeschlossen und als Energiequelle genutzt. Hier wurden die Werte wie erwartet aufgezeichnet

Bei einem zweiten Versuchslauf wurde der Batterieanschluss mit einer neuen/vollen Batterie genutzt. Hier wurden die Messwerte nur anfänglich aufgezeichnet. Nach einiger Zeit ca. 15 Minuten wurden im Log stets die selben Werte eingestellt. Es ist hier wohl die selbe Problematik wie bei anderen Geräten auch, dass die Batterie zu ihrer dauerhaften Stromversorgung zu schwach ist. Optisch war dies daran zu erkennen, dass die weiße Diode auf dem SCD40 weniger hell leuchtete.

1	Time (minutes)	Temperatur	Feuchtigkeit	CO2
2	0.18	19.7	51	1189
3	0.20	18.7	41.4	767
4	10.22	17.5	49.4	696
5	20.24	17.9	49.2	760
6	30.26	18.1	50	833
7	40.28	18.2	50.8	901
8	50.30	18.2	50.6	929
9	60.31	18.3	51.1	954
10	70.33	17.8	46.8	815

1	Time (minutes)	Temperatur	Feuchtigkeit	CO2
2	0.18	19.7	51	1189
3	0.2	18.7	41.4	767
4	10.22	17.5	49.4	696
5	20.24	17.9	49.2	760
6	30.26	18.1	50	833
7	40.28	18.2	50.8	901
8	50.3	18.2	50.6	929
9	60.31	18.3	51.1	954
10	70.33	17.8	46.8	815

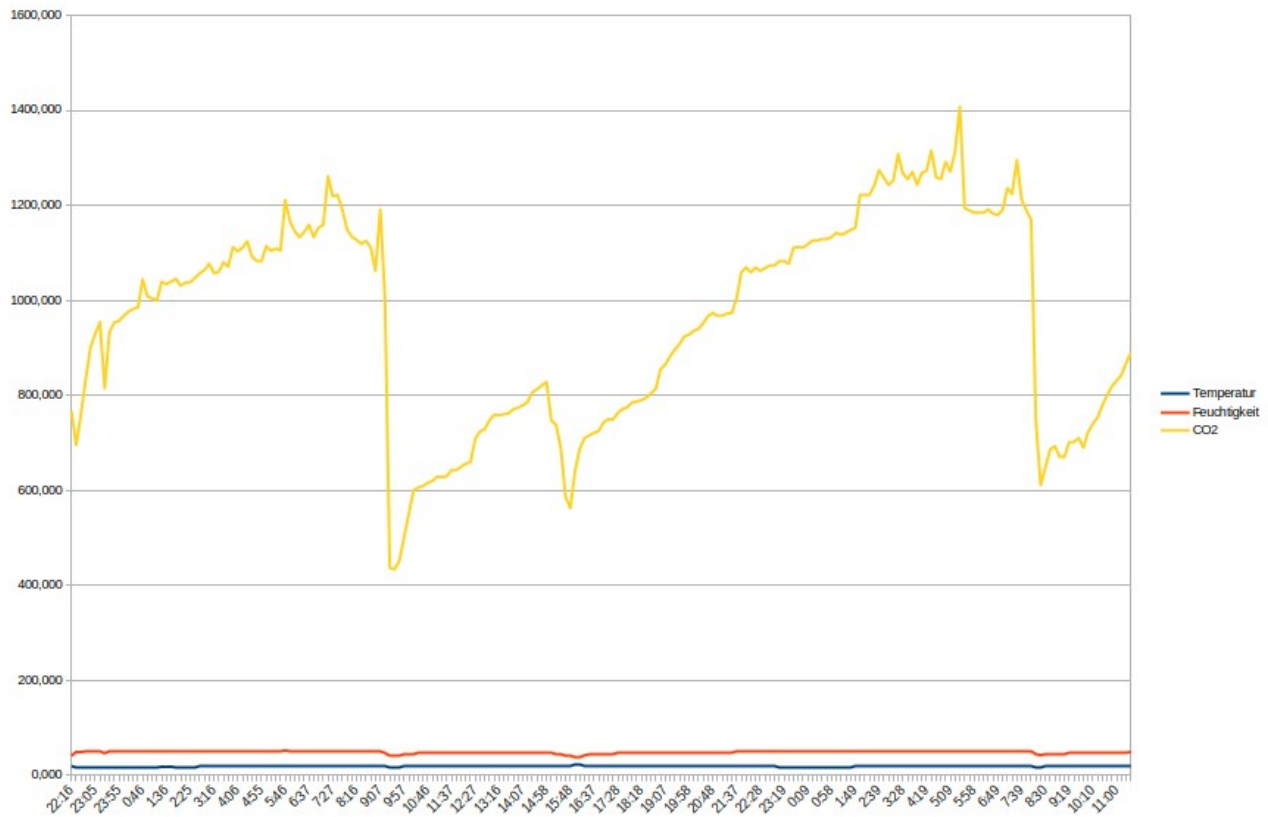
3.4.3 Daten Ergänzen - Uhrzeit

Die **Uhrzeit auf der X-Achse** wird aus den Minutenangaben zurückgerechnet von dem Zeitpunkt zu dem das Programm beendet wurde und so die jeweiligen Uhrzeiten der Beobachtungen ermittelt.

Das Experiment wurde um 11:30 Uhr beendet. Der Beginn wird mit 22:16 Uhr ermittelt, was mit der notierten Anfangszeit übereinstimmt. Das Experiment begann um 22:16 Uhr an Tag 1, sammelte ab 22:16 Uhr an Tag 1, den ganzen Tag 2 sowie Tag 3 bis 11:30 Uhr Messwerte im 10 Minuten-Takt.

Hier das aufbereitete Ergebnis mit den Originaldaten versehen mit den ermittelten Uhrzeiten auf der X-Achse. Die Messwerte können mit Hilfe der Y-Achse abgelesen werden.

Das Schaubild sieht so aus wie das vom Data Simulator erzeugte.



Was kann man aus den Daten ablesen?

Auswertbar sind nur die Daten zu CO2. Die Messreihen zu Temperatur und Feuchtigkeit sind zu gleichförmig.

Die CO₂-Werte nehmen im Laufe der Nacht zu, erreichen gegen 6 Uhr einen Höchstwert, fallen gegen 8 Uhr stark ab um von an im Laufe des Tages unterbrochen durch einen Abfall gegen 14:30 Uhr bis in den nächsten Morgen an um dann wieder steil abzufallen.

3.4.4 Daten Ergänzung: Messwerte normieren

Zu jeder Messreihen zu Temperatur, Feuchtigkeit und CO2 werden folgende Werte ermittelt:

- Maximum: den grössten Wert in der Messreihe
- Minimum: den kleinsten Wert in der Messreihe
- Spannweite: als Differenz von Maximum und Minimum
- arithmetischen Mittelwert: Summe aller Messwerte dividiert durch die Anzahl der Messwerte

	Temperatur	Feuchtigkeit	CO2
Maximum	23,1	52,7	1407
Minimum	16,5	38,9	434
Spannweite	6,6	13,8	973
Mittelwert	18,8	49,3	962

Diese zusätzlichen Werte können mittels Funktionen der Spreadsheet-Anwendung ermittelt werden.

Nun werden die Messwerte in ein neues Werteschema eingepasst in dem die Werte aller drei Messreihen zwischen 0 und 1 normiert werden:

- **Schritt 1: auf das Minimum als Nullwert reduzieren** indem der neue Wert als Messwert – Minimum ermittelt wird
Damit wird erreicht, dass die Messwerte jeder der drei Werte bei Null beginnen (dort wo die Messwerte das Minimum annehmen) und als höchsten Wert die Spannweite annimmt (dort wo die Messwerte das Maximum annimmt, beim Maximum wird dann das Minimum abgezogen).

Das Ergebnis:

- alle drei Messreihen beginnen bei Null
- die Y-Achse zeigt nicht mehr die absoluten Messwerte an, sondern die um das Minimum reduzierten.

	Temperatur	Feuchtigkeit	CO2
Minimum	0,0	0,0	0,0
Maximum	6,6	13,8	973,0

Formel: neuer Messwert = alter Messwert – Minimum

- **Schritt 2: Auf die Spannweiten normieren.** Um die Messreihen der einzelnen Beobachtungsgrößen vergleichbar zu machen werden die in Schritt 1 auf das Minimum reduzierten Messwerte durch die jeweilige Spannweite dividiert. Da die neuen Messwerte nur Werte zwischen 0 und Spannweite annehmen können, ergibt sich Division durch Spannweite in allen drei Messreihen eine Obergrenze von 1, die einzelnen Messwerte sind in das Raster von 0 bis 1 eingepasst.

Normierter Messwerte = neuer Messwert aus Schritt 1 / Spannweite

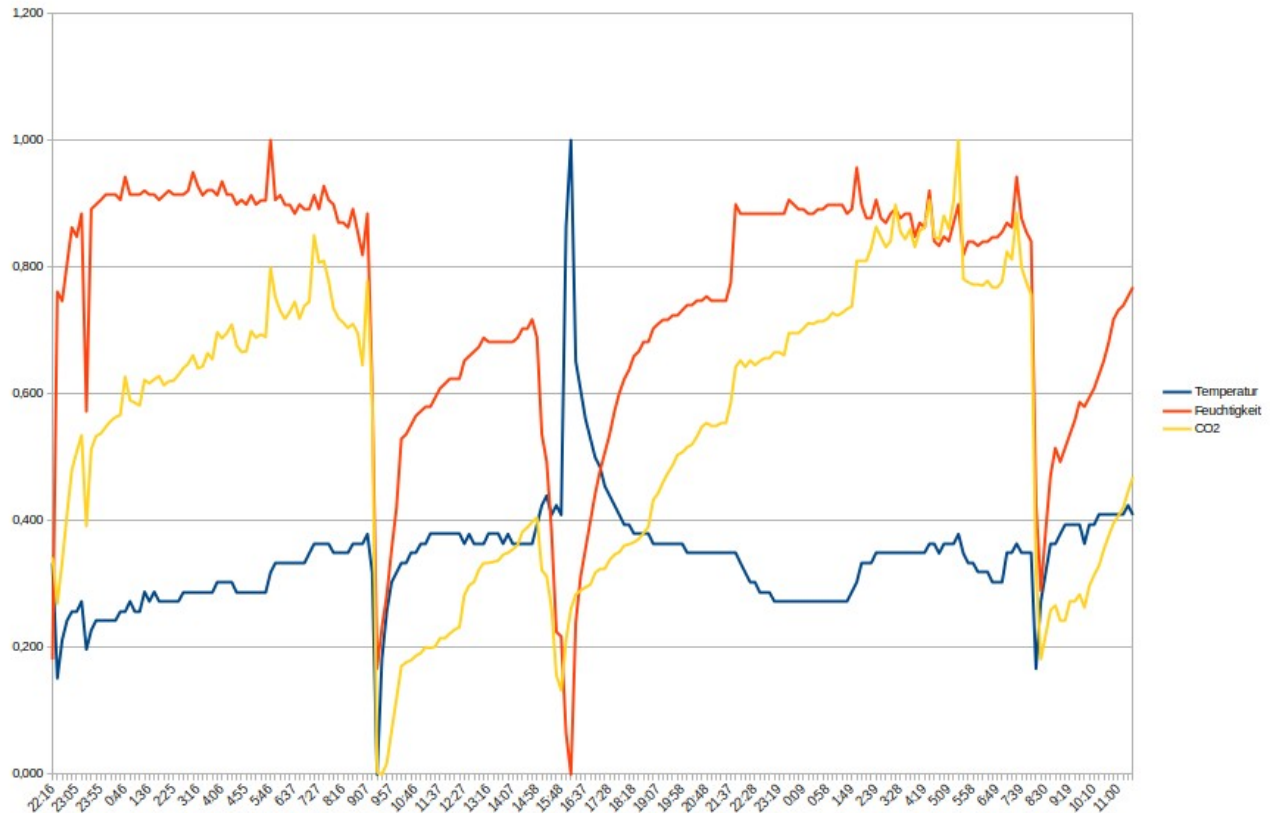
Beide Schritte in einer Formel vereint: (OriginalMesswert – Minimum) / Spannweite

Wendet man dies auf alle Messwerte der einzelnen Messreihen an erhält man Messwerte, die bei Null nicht unterschreiten (dort wo die Daten der Messreihe das Minimum annehmen) und höchstens den Wert 1 annehmen (dort wo die Daten der Messreihe das Maximum annehmen).

Ergebnis:

- die Werte aller drei Messreihen liegen zwischen Null und Eins
- die Y-Achse / Vertikale zeigt nur noch den relativen Wert des originalen Messwertes bezüglich der Spannweite an
- die Bewegungen der einzelnen Messreihen bleiben erhalten
- Bewegungen der verschiedenen Messreihen werden vergleichbar

Wenn wir aus den normierten Daten ein neues Schaubild zeichnen sieht es so aus:



3.5 Auswertung der Daten

3.5.1 Interpretation der Daten

- Grösstenteils zeigen die Messreihen abweichende Verhalten auf, sie sind weitgehend unabhängig voneinander
- es gibt Bereiche in denen die Messreihen sich gleich entwickeln: zu Beginn, ca. 9 Uhr am zweiten Tag, ca. 8 Uhr am dritten Tag
- es gibt einen Bereich in dem sich das Verhalten der Temperatur genau umgekehrt verhält wie das von Feuchtigkeit und CO₂:zweiter Tag ca. 14 Uhr.

3.5.2 Die Geschichte zu den Daten

Die Messwerte wurden in einem Schlafzimmer aufgenommen.

- Person 1 geht am ersten Tag gegen 22 Uhr zu Bett. Luftfeuchtigkeit und CO₂-Gehalt steigen an, die Temperatur erhöht sich mässig.
- Person 2 geht gegen 23 Uhr zu Bett: durch die Türbewegung wird Luft bewegt was zur Reduzierung von Feuchtigkeit und CO₂-Gehalt führt. Danach steigen die Werte von Feuchtigkeit und CO₂ wieder an.
- Gegen 7 Uhr am zweiten Tag stehen beide auf. Feuchtigkeit und CO₂-Gehalt nehmen leicht ab.
- Gegen 9 Uhr wird gelüftet:
 - Da es draussen kalt ist fällt die Temperatur auf das Minimum
 - Die hinzukommende frische Luft senkt den CO₂-Gehalt.
 - Da kalte hinzukommende Luft beim Erwärmen mehr Feuchtigkeit aufnehmen kann, fällt die relative Feuchtigkeit.

- Gegen 15 Uhr Lüften bei massiver Sonneneinstrahlung:
 - Die Luft erwärmt sich, die Temperatur steigt auf das Maximum.
 - Die beobachteten Auswirkungen auf CO₂ und Feuchtigkeit sind eine Vermutung. Vielleicht kann jemand dies profund erklären.
 - hinzukommende frische Luft senkt den CO₂-Gehalt
 - erwärmte Luft kann mehr Feuchtigkeit aufnehmen kann, deshalb fällt auch die Feuchtigkeit.
- Person 1 geht gegen 22 Uhr zu Bett
- Person 2 geht gegen Mitternacht zu Bett
- Person 1 steht am dritten Tag gegen 5:30 Uhr auf
- Person 2 steht am dritten Tag gegen 6:30 Uhr auf
- Lüften gegen 8 Uhr, draussen ist es kalt.

3.5.3 Kleiner Ausflug: Bedeutung der CO₂-Werte

Quelle für dieses Kapitel: <https://www.dein-heizungsbauer.de/ratgeber/raumklima/co2-raumluft/>

Folgen von zu viel CO₂ in Raumluf

Die Folgen einer zu hohen CO₂-Konzentration sind vielfältig und wurden beispielsweise in einer umfangreichen Studie von Max Josef von Pettenkofer untersucht. Die Studie zeigt, dass sich Personen in einem Raum mit einer Konzentration von weniger als 1.000 ppm behaglich fühlen, bei mehr als 2.000 ppm jedoch unbehaglich.

Diese gesundheitlichen Probleme können durch eine zu hohe CO₂-Konzentration auftreten:

- Kopfschmerzen
- Konzentrationsschwäche
- Leistungsabfall
- erhöhtes Ansteckungsrisiko
- erhöhte Atemfrequenz

Klassifizierung der Atemluftqualität gemäß Umweltbundesamt

CO ₂ -Gehalt im Raum	Hygienische Bewertung	Empfehlungen
< 1.000 ppm	hygienisch unbedenklich	keine weiteren Maßnahmen
1.000 bis 2.000 ppm	hygienisch auffällig	Lüftungsmaßnahme, Lüftungsverhalten überprüfen und verbessern
> 2.000 ppm	hygienisch inakzeptabel	Belüftbarkeit des Raums prüfen, gegebenenfalls weitergehende Maßnahmen

Interpretation der Daten

Um die Messdaten gegen diese Werte abzugleichen, gehen wir zurück zum ersten Schaubild der Originaldaten. Es zeigt, dass die gemessenen Werte in einigen Zeitabschnitten oberhalb der 1.000 ppm Linie liegen und bis ca. 1.400 ppm ins Maximum gehen.

Deshalb sollte man hier die Möglichkeit einer Lüftung in den frühen Stunden des Tages vornehmen bzw. eine dauerhafte Lüftung vorzunehmen.

3.5.4 Achtung mit Datenweitergabe

Bereits aus diesen wenigen Daten kann bei bekanntem Umfeld (Schlafzimmer, zwei Personen, Aussentemperatur) der Tagesablauf von Personen abgelesen werden. Verknüpft mit weiteren Daten wie z.B. aktueller Höhe des Stromverbrauchs oder wann wieviel Wasser verbraucht wird, Transportvolumen über den Internetanschluss können Profile von Hausbewohnern erstellt werden.

Deshalb vorsichtig bei der Weitergabe von vermeintlich unwichtigen Daten.

4 DATALOGGER - DIE GRENZEN AUSTESTEN

4.1 Motivation

Mit dem DataLogger ist es möglich Daten über einen längeren Zeitraum aufzuzeichnen ohne diese laufend ablesen zu müssen. Da wird es interessant zu wissen welches Datenvolumen / Anzahl Sätze das Log aufnehmen kann.

4.2 Testszenarien

Ein Programm schreibt Sätze mit festen Inhalten solange in den zuvor geleerten Speicher des Calliope mini bis die Aufnahmekapazität erschöpft ist.

Vier Szenarien wurden untersucht. In allen Szenarien wird die Zeit festgehalten (Block „Setze Zeitstempel auf Sekunden“ aus der DataLogger-Erweiterung) sowie die Anzahl des Satzes (eigene Variable, die mit jedem Durchgang hochgezählt wird), die als Zahl hochgezählt und mit abgespeichert wird. Die Szenarien unterscheiden sich in den weiteren Werten:

- Szenario 1: es wird ein Text mit 100 Zeichen geschrieben.
- Szenario 2: es werden zwei Texte mit je 50 Zeichen geschrieben.
- Szenario 3: es werden fünf Texte mit je 20 Zeichen geschrieben.
- Szenario 4: es werden fünf Zahlen geschrieben, zwei Ganzzahlig, drei Dezimalzahlen

Jedes Szenario ist in zwei Varianten ausgeführt worden: mit und ohne bedienen der seriellen Schnittstelle (Block „spiegle Daten seriell“).

Zahlen - Hier sind die Ergebnisse aus den Szenarien zusammengefasst.

Szenario	1 Text 100 Zeichen					
	mit Schreiben seriell			ohne Schreiben seriell		
	Schreibdauer In Sekunden	Anzahl Sätze In Stück	Löschzeit in Millisek.	Schreibdauer In Sekunden	Anzahl Sätze In Stück	Löschzeit in Millisek.
Versuch1	33,17	1070	492	22,24	1071	475
Versuch2	32,97	1071	484	21,93	1072	484
Versuch3	32,97	1070	484	21,95	1072	484
Szenario	2 Texte je 50 Zeichen					
	mit Schreiben seriell			ohne Schreiben seriell		
	Schreibdauer In Sekunden	Anzahl Sätze In Stück	Löschzeit in Millisek.	Schreibdauer In Sekunden	Anzahl Sätze In Stück	Löschzeit in Millisek.
Versuch1	33,00	1061	484	24,57	1063	475
Versuch2	32,75	1060	484	25,41	1060	484
Szenario	5 Texte je 20 Zeichen					
	mit Schreiben seriell			ohne Schreiben seriell		
	Schreibdauer In Sekunden	Anzahl Sätze In Stück	Löschzeit in Millisek.	Schreibdauer In Sekunden	Anzahl Sätze In Stück	Löschzeit in Millisek.
Versuch1	34,35	1041	484	24,28	1043	475
Versuch2	32,69	1040	484	24,1	1044	475
Szenario	5 Zahlen					
	mit Schreiben seriell			ohne Schreiben seriell		
	Schreibdauer In Sekunden	Anzahl Sätze In Stück	Löschzeit in Millisek.	Schreibdauer In Sekunden	Anzahl Sätze In Stück	Löschzeit in Millisek.
Versuch1	42,91	3744	473	29,01	3865	475
Versuch2	43,14	3881	484	28,41	3872	484
Versuch3	43,30	3879	475	28,51	3899	475
Versuch4	42,96	3857	484	28,50	3876	475
Versuch5	43,43	3872	484	28,50	3897	484

Zusammenfassungen

- Beobachtungen sind bei allen Szenarien identisch
- Bei vollem Log wird in der Browser-Ansicht die Meldung „LOG VOLL“ angezeigt. Der Block „bei voller Logdatei“ reagiert nicht, wenn die Log-Datei voll ist.
- Das Programm löst den Event „bei voller Logdatei“ nicht aus. Eine Erklärung sowie Lösungen hierfür werden im nachfolgenden Kapitel aufgeführt.
- Darstellungsproblem im Browser bei Schreiben von Text, der deutsche Umlaute verwendet.

Beobachtungen: bei allen Szenarien identisch

- Bei **vollem Log**: - Programm hängt – Event wird nicht gefeuert
 - Die im Block „Bei voller Logdatei“ hinterlegten Blocks werden nie ausgeführt:
 - Kreuz wird in der 5x5-LED nicht angezeigt
 - die LEDs werden nicht rot angezeigt.
 - Nach Vollschieben der Datei, läuft das Programm weiter – erkennbar am Wechsel der grünen-und-weissen LEDs in schneller Folge – es wird nur noch prozessiert, Schreibvorgang findet nicht stattfindet.
 - Das Programm reagiert auf keine weitere Aktionen:
 - Block „bei voller Logdatei“ wird nicht ausgeführt
Falls doch, müsste die While-Schleife beendet werden
 - Reagiert nicht auf Tastendruck B sowie Betätigen Pin 0
 - Weiter geht nur mit Reset / USB entfernen, wenn keine Batterie an / Batterie aus, wenn kein USB an.
 - Nach Restart (Reset / USB anschliessen) wird die volle Log-Datei nicht erkannt.
Das Programm läuft nach Betätigen der Taste A in der While-Schleife.
- Die **Zeit**, die **zum Löschen** des Logs benötigt wird, liefert genau zwei Ergebnisse: 475 ms und 484 ms. Die Zeiten sind unterschiedlich in Abhängigkeit, davon ob die Logs nach Reset (475 ms) oder Wiederverbinden über USB (484 ms) gelöscht werden.
- Das **Schreiben mit serieller Schnittstelle** dauert ca. 1,5-mal solange wie das **Schreiben ohne serielle Schnittstelle** zu bedienen.
- Sinnvoll: **Neuer Reporter „Log voll“** - teilt mit, dass der Log voll ist
=> kann direkt in eine Logikabfrage eingebunden werden, statt erst im Block „bei voller Logdatei“ über eine Variable verfügbar gemacht zu werden.

4.3 Achtung: deutsche Umlaute

Szenario 6: Szenario4 mit geändertem Text: in den bisher verwendeten Szenarien wird das Wort „Behalter“ verwenden. Hier ist es durch das Wort „Behälter“ ersetzt.

Beobachtungen:

- Das Programm auf dem Calliope mini läuft ab wie in Szenario 4 = Text ohne Umlaut.
- Die Anzeige der grün-weißen-LEDs wechselt, zeigt damit an, dass das Programm läuft. Die Dauer des Wechsels zwischen den LED-Anzeigen ist die selbe wie in Szenario 4.
- In der Anzeige der DataLogger-Daten wird nur der Teil bis zum ersten Umlaut angezeigt, in der Darstellung wird die Meldung „LOG VOLL“ angezeigt.

Folgerungen:

- Die Daten sind auf dem Calliope mini vorhanden = geschrieben worden. Dieser Zustand wird im Browser angezeigt mit der Meldung „LOG VOLL“, Die Anzeige der Daten wird ab dem ersten Umlaut ignoriert.
- Es liegt ein Darstellungsproblem des DataLogger vor.

LOG VOLL

Time (seconds)	Zeile	Testtext	Testtext2	Testtext3	Testtext4	Testtext5
5.60	1	Fur das Experiment m	it dem Calliope mini	und den transparent	en Beh	

Beispiel:

Anzeige im Browser – Firefox 137.0.2 (64-bit):

LOG VOLL

Time (seconds)	Zeile	Testtext	Testtext2	Testtext3	Testtext4	Testtext5
23.92	1	Fur das Experiment m	it dem Calliope mini	und den transparent	en Beh	

Ausschnitte aus der MY.DATA.HTM-Datei – geöffnet mit TXT-Editor:

Anfang und das Ende der MY.DATA.HTM-Datei. Der deutsche Umlaut „ä“ ist mit „\E4“ eingefügt.

Ende: zeigt an, dass im Log sind 1041 Sätze vorhanden sind.

[illegible]

4.4 Szenario 1: 100 Zeichen in einem Text mit 100 Zeichen

Text:

Für das Experiment mit dem Calliope mini und den transparenten Behältern eignen sich kleine Pflanzen

Programm



Ergebnisse:

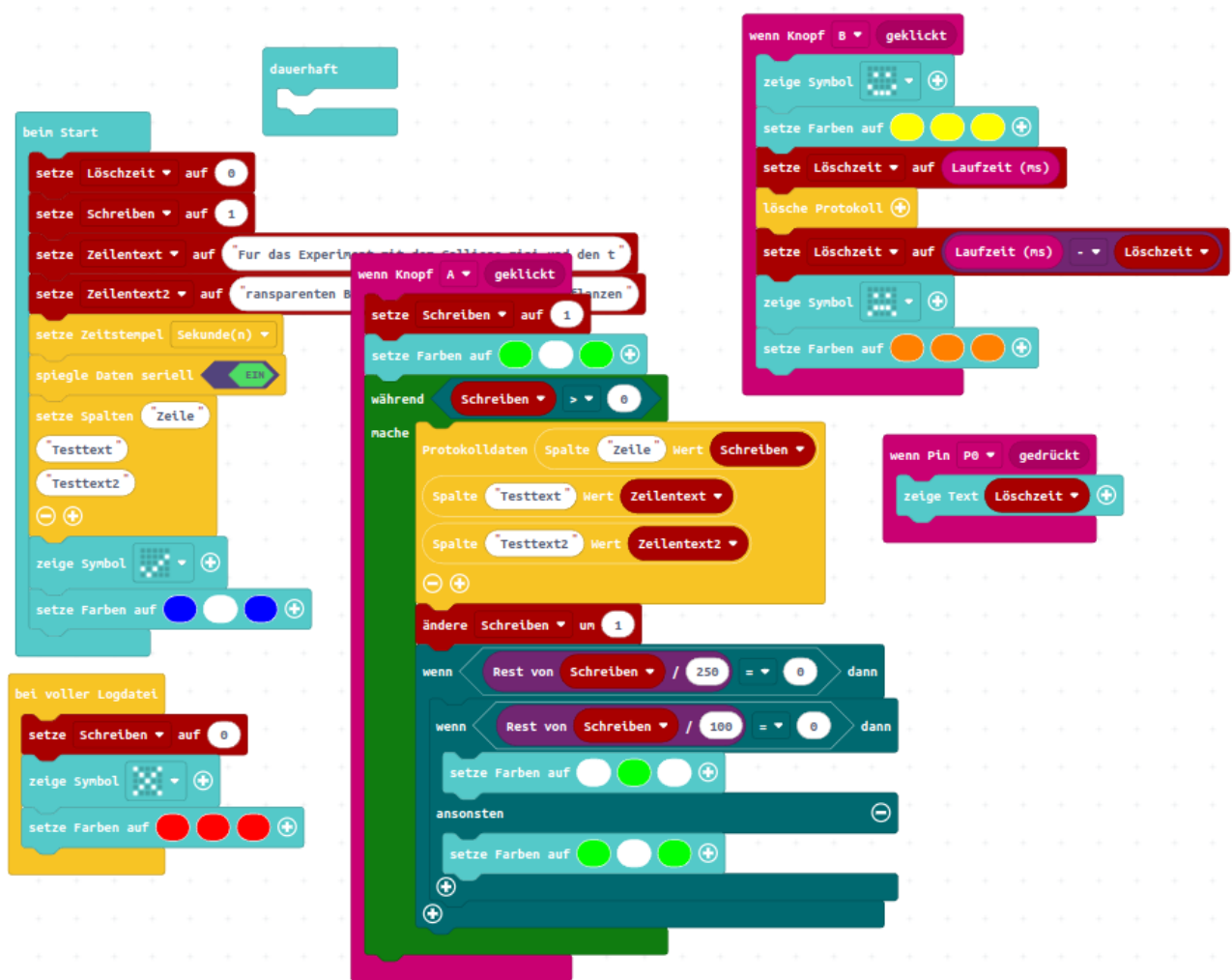
- Mit Schreiben seriell:
 - Anzahl geschriebener Sätze in Stück: 1070 bzw. 1071 bzw. 1070
 - Schreibzeit in Sekunden: 13.90 bis 47.07 = 33,17 Sekunden bzw. 4.97 bis 37.94 = 32,97 Sekunden bzw. 19.13 bis 52.10 = 32,97 Sekunden
 - Löschzeit in ms: 492 bzw. 484 bzw. 484
- Ohne Schreiben seriell:
 - Anzahl geschriebener Sätze in Stück: 1071 bzw. 1072 bzw. 1072
 - Schreibzeit in Sekunden: 7.99 bis 30.23 = 22.24,17 Sekunden bzw. 3.99 bis 25.92 = 21.93 Sekunden bzw. 4.34 bis 26.29 = 21,95 Sekunden
 - Löschzeit in ms: 475 bzw. 484 bzw. 484

4.5 Szenario 2: 100 Zeichen in zwei Texte mit je 50 Zeichen

Text

Für das Experiment mit dem Calliope mini und den transparenten Behältern eignen sich kleine Pflanzen

Programm



Ergebnis

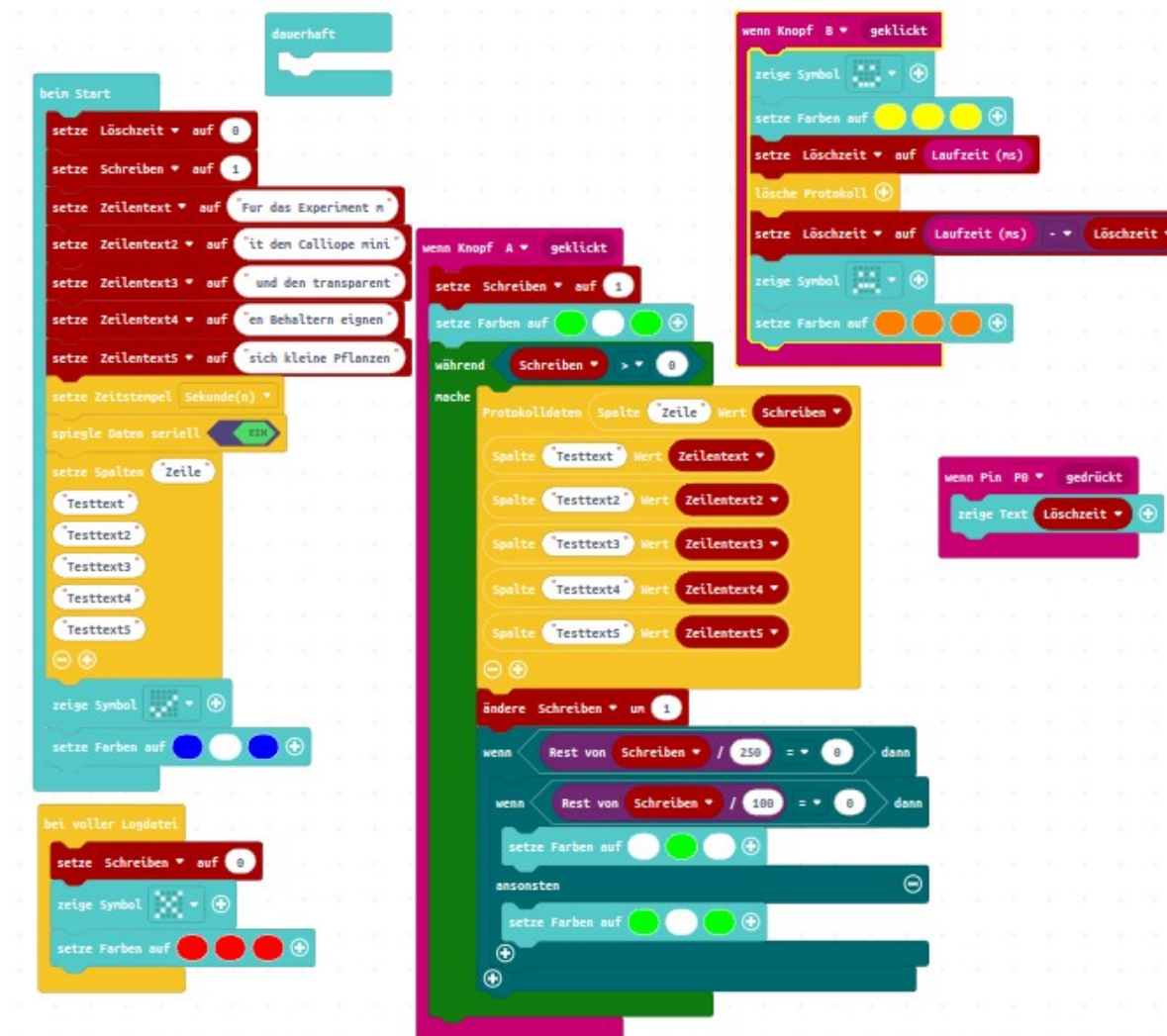
- mit Schreiben serieller Schnittstelle:
 - Anzahl geschriebener Sätze in Stück: 1061 bzw. 1060
 - Schreibzeit in Sekunden: 6.53 bis 39.53 = 33,00 Sekunden bzw. 42.74 bis 75.49 = 32,75 Sekunden
 - Löschozeit in ms: 484 bzw. 484
- ohne Schreiben serieller Schnittstelle:
 - Anzahl geschriebener Sätze in Stück: 1063 bzw. 1060
 - Schreibzeit in Sekunden: 3.13 bis 27.70 = 24,57 Sekunden bzw. 3.46 bis 25.41 = 21,95 Sekunden
 - Löschozeit in ms: 475 bzw. 484

4.6 Szenario 3: 100 Zeichen in fünf Texte mit je 20 Zeichen

Text

Für das Experiment mit dem Calliope mini und den transparenten Behältern eignen sich kleine Pflanzen

Programm



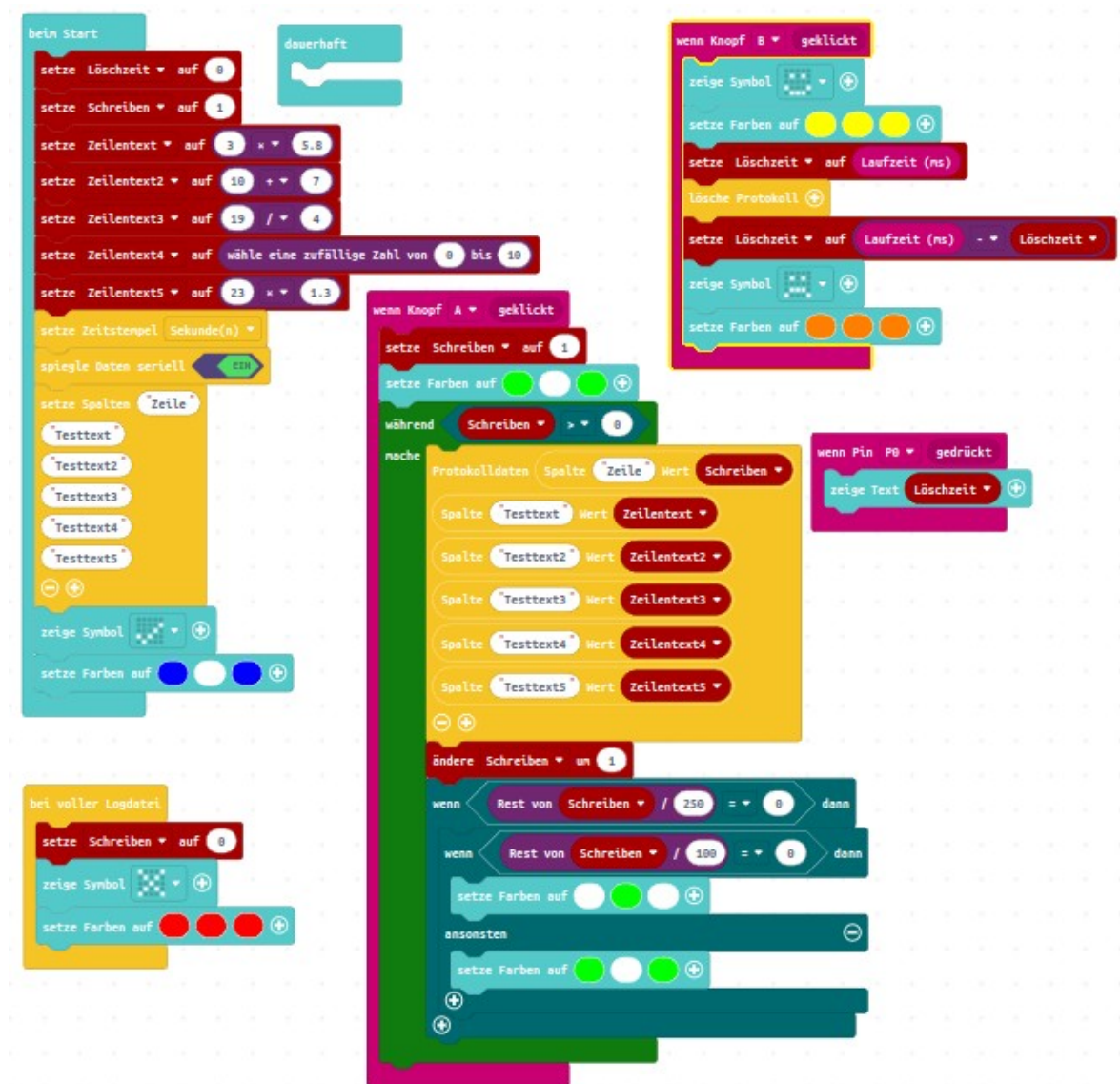
Ergebnis

- mit Schreiben serieller Schnittstelle:
 - Anzahl geschriebener Sätze in Stück: 1041 bzw. 1041
 - Schreibzeit in Sekunden: 12.68 bis 47.03 = 34,35 Sekunden
bzw. 34.90 bis 67.59 = 32,69 Sekunden
 - Löschzeit in ms: 484 bzw. 484
- ohne Schreiben serieller Schnittstelle:
 - Anzahl geschriebener Sätze in Stück: 1043 bzw. 1044
 - Schreibzeit in Sekunden: 5.94 bis 30.32 = 24,28 Sekunden
bzw. 4.20 bis 28.30 = 24,10 Sekunden
 - Löschzeit in ms: 475 bzw. 475

4.7 Szenario 4: 5 Zahlen, zwei Integer, drei Nachkommazahlen

Text Daten: fünf fest vorgegebene Zahlen, die in jeden Satz geschrieben werden

Programm



Ergebnis:

- mit Schreiben serieller Schnittstelle:
 - Anzahl geschriebener Sätze in Stück: 3744 bzw. 3881 bzw. 3879 bzw. 3857
 - Schreibzeit in Sekunden: 12.02 bis 54.93 = 42,91 Sekunden bzw. 4.61 bis 47,75 = 43,14 Sekunden 5,07 bis 48,37 = 43,30 Sekunden bzw. 44,54 bis 87,50 = 42,96 Sekunden 59,21 bis 102,64 = 43,43 Sekunden
 - Löschzeit in ms: 473 bzw. 484 bzw. 475 bzw. 484 bzw. 484
- ohne Schreiben serieller Schnittstelle:
 - Anzahl geschriebener Sätze in Stück: 3865 bzw. 3872 bzw. 3899 bzw. 3876 bzw. 3897
 - Schreibzeit in Sekunden: 14.31 bis 43.40 = 29.01 Sekunden bzw. 8.47 bis 36.88 = 28,41 Sekunden 2.84 bis 31.35 = 28,51 Sekunden bzw. 2.46 bis 30.96 = 28,50 Sekunden 2.84 bis 31.34 = 28,50 Sekunden
 - Löschzeit in ms: 475 bzw. 484 bzw. 475 bzw. 475 bzw. 484

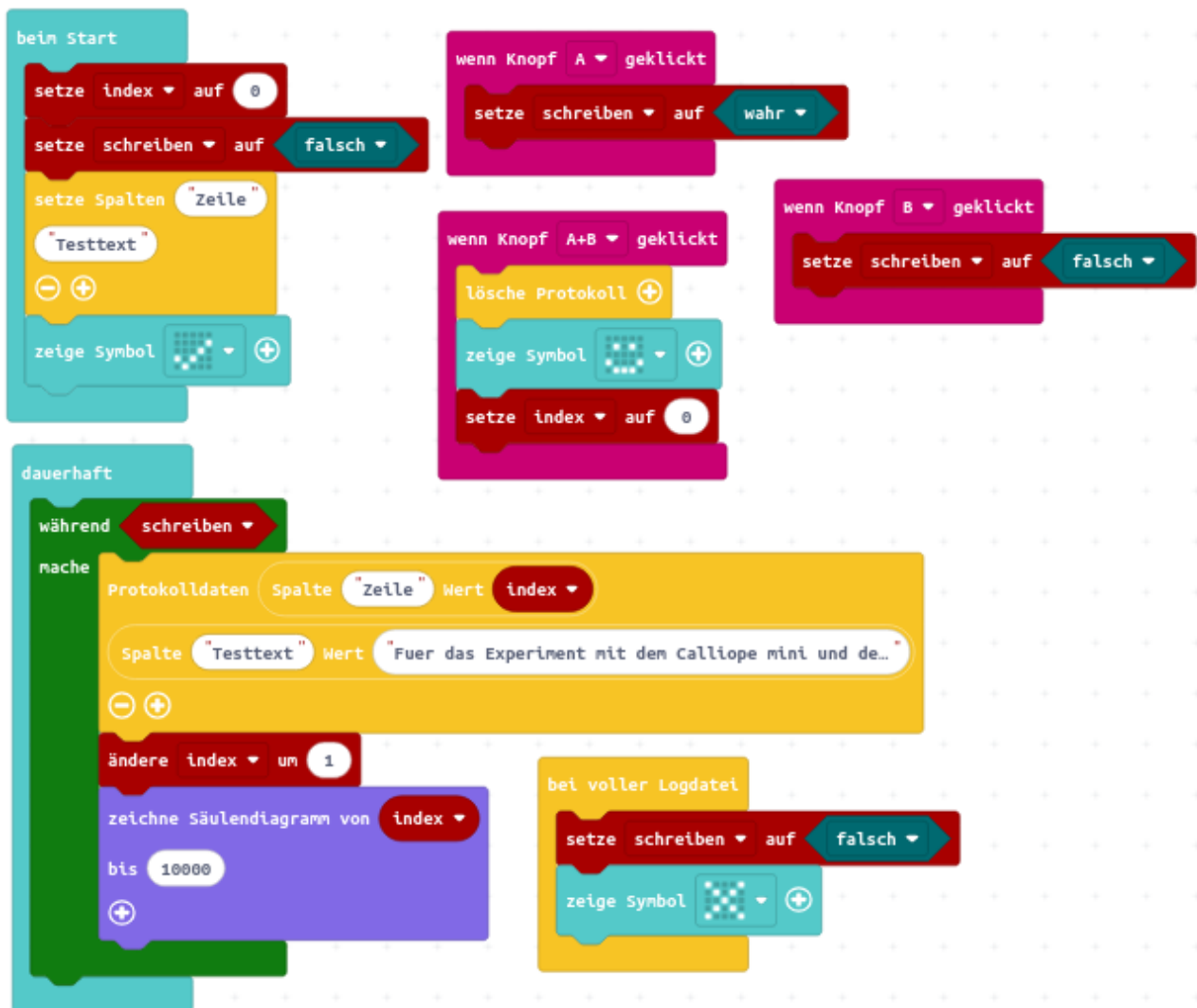
5 EVENT ‚BEI VOLLER LOGDATEI‘

Das Testprogramm löst den Event „bei voller Logdatei“ nicht aus. Wenn das DataLog vollgeschrieben ist, wird die Schleife weiter durchlaufen. Erkennbar daran, dass die Anzeige der LEDs viel schneller wechselt, da kein Schreibvorgang mehr erfolgt. In der Datenanzeige im Calliope mini wird die Nachricht „LOG VOLL“ ausgegeben.

Hier Lösungsmöglichkeiten aus dem Labor der Calliope gGmbH:

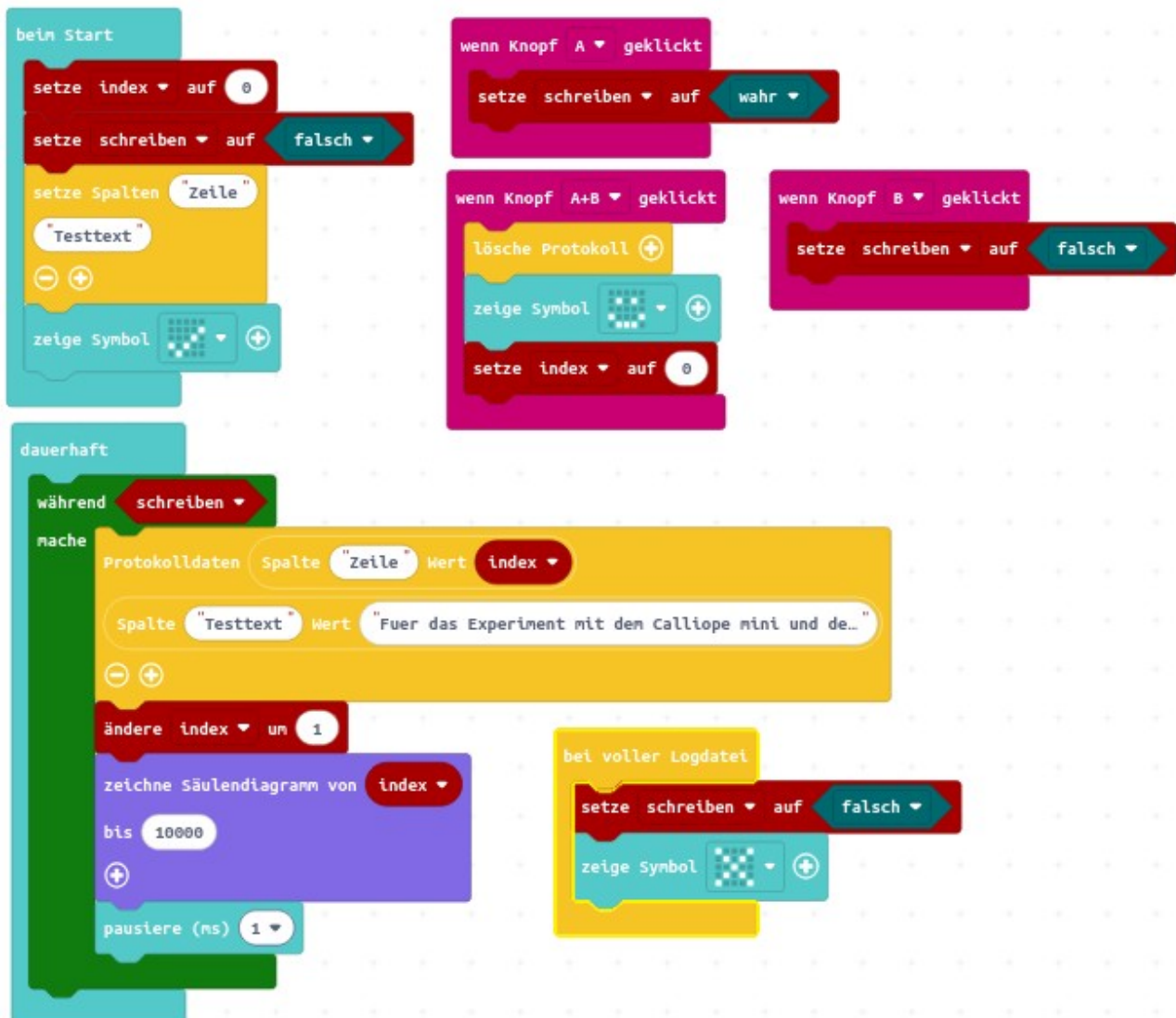
DataLoggerTest01 – kein stopp

Dies ist das Äquivalent zum Testprogramm und verhält sich genau so. Der Event „bei voller Logdatei“ wird nicht ausgelöst.



DataLoggerTest02 – pause

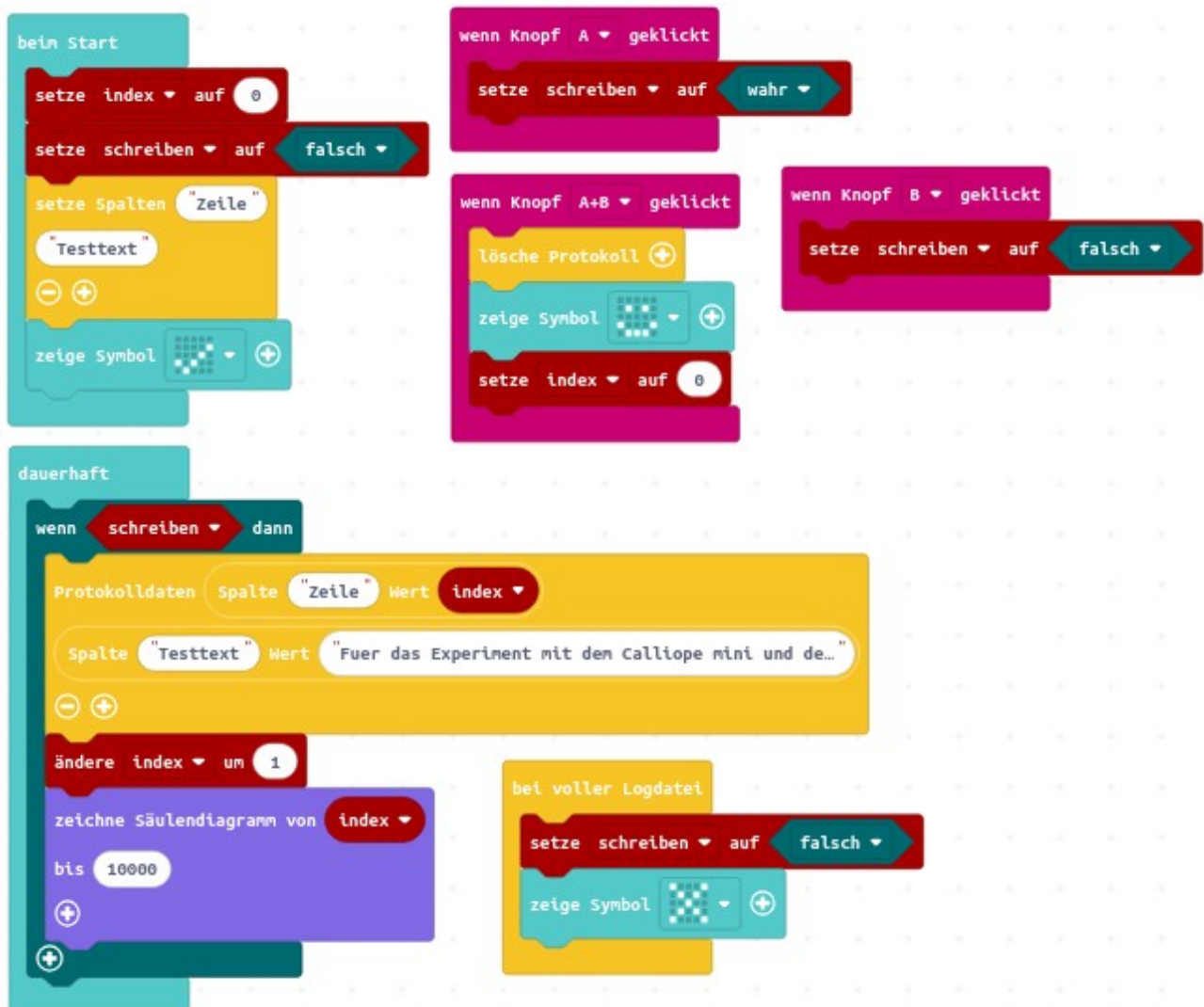
Das selbe Programm mit der zusätzlichen Zeile „pausiere 1ms“. Hier wird der Event ausgelöst.
Erklärung: durch das Warten bekommt der Event-Handler die Möglichkeit aktiv zu werden.

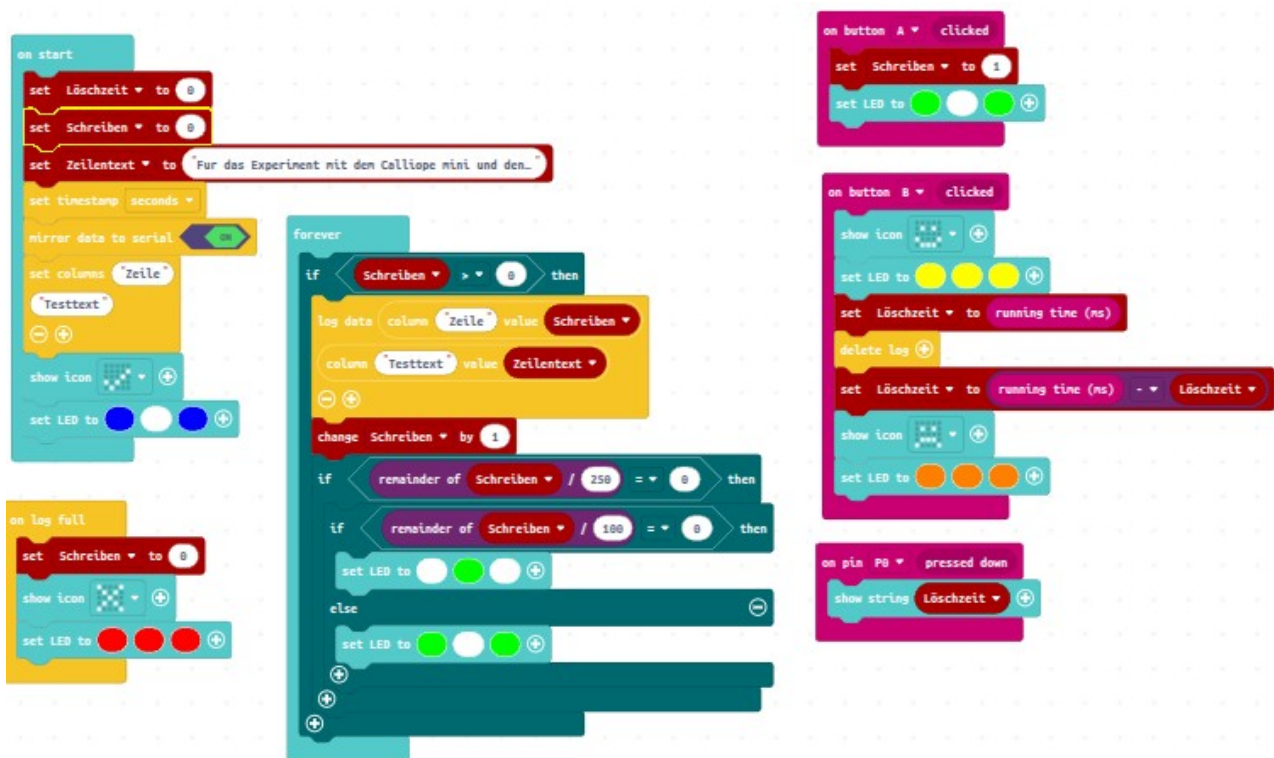


DataLoggerTest03 – if -forever

Das selbe Programm mit „wenn – dann“ statt „während – mache“. Hier wird der Event ausgelöst.

Erklärung: der „wenn – dann“-Block ermöglicht dem Event-Handler aktiv zu werden. Der „während – mache“-Block unterdrückt dies.

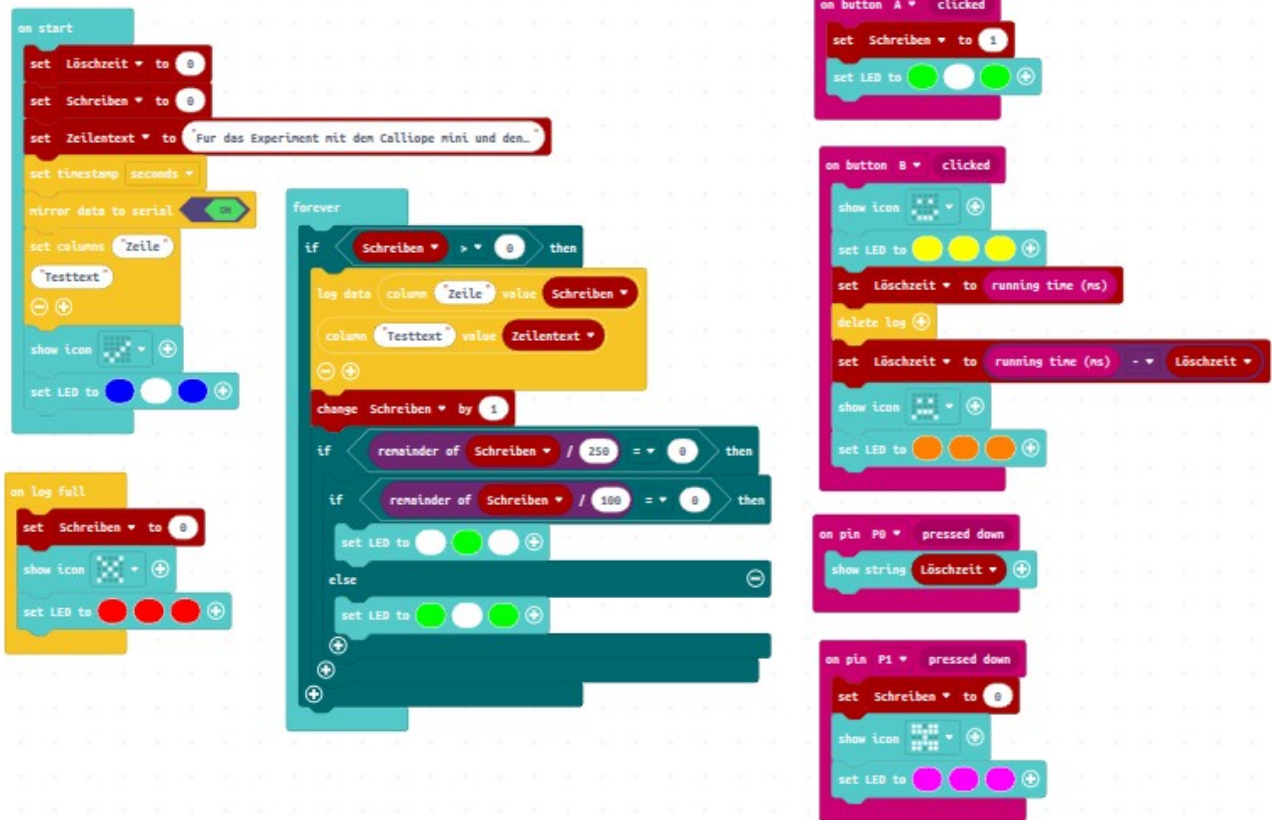




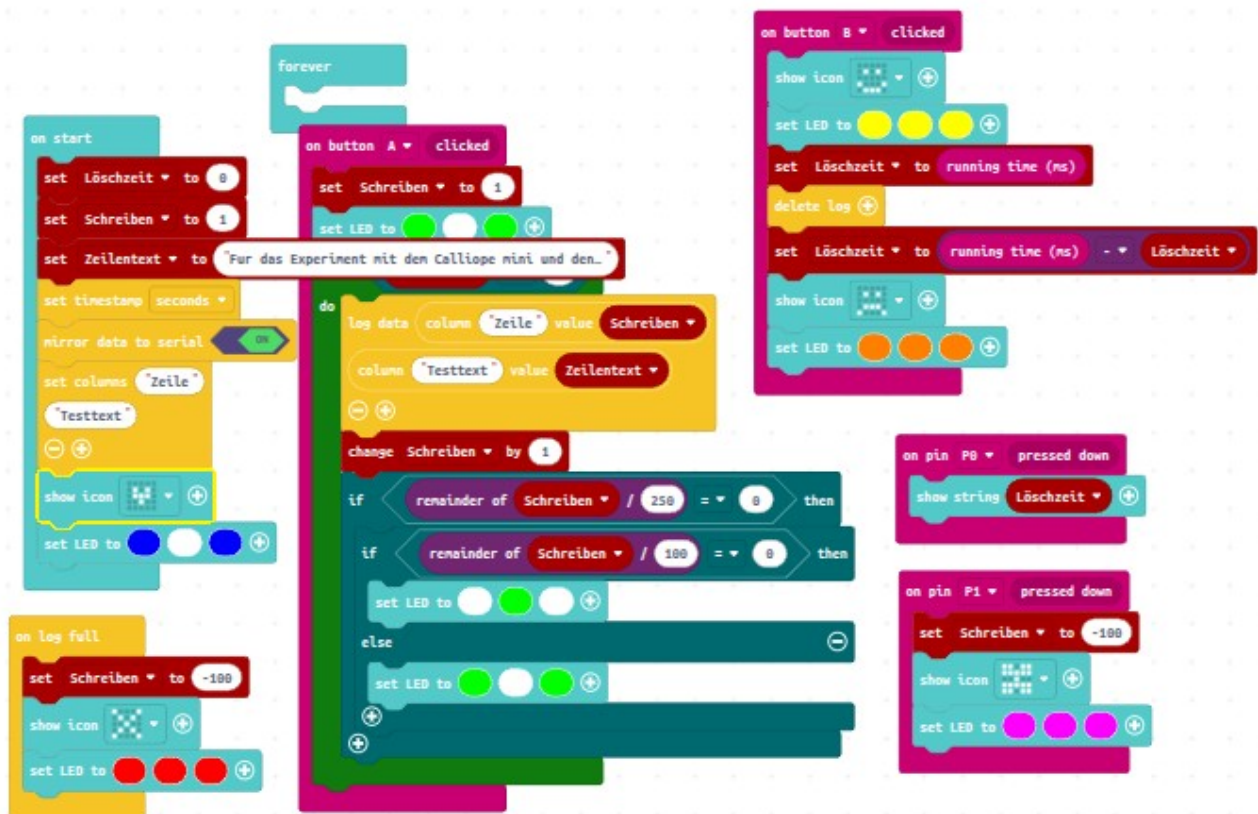
Prozessdauer: 67,83 Sekunden – 13,47 Sekunden = 54,36 Sekunden

Vergleich Variante mit „während »Variable« Prozessdauer 33 Sekunden.

Mit Pin1



im on log full -block Schreiben auf -100 gesetzt



6 DETAILS DES CALLIOPE MINI:

```
# DAPLink Firmware - see https://daplink.io
Build ID: v0258-2-gfe87dbba (gcc)
Unique ID: 9906360200052820235078dd139bf30f000000006e052820
HIC ID: 6e052820
Auto Reset: 1
Automation allowed: 1
Overflow detection: 1
Incompatible image detection: 0
Page erasing: 0
Daplink Mode: Interface
Interface Version: 0258
Bootloader Version: 0258
Git SHA: fe87dbba9c2c4dd9c46a328dbd7322922c0ea863
Local Mods: 0
USB Interfaces: MSD, CDC, HID, WebUSB
Bootloader CRC: 0x362c4d42
Interface CRC: 0x615c2f6b
Remount count: 0
URL: https://calliope.cc
```