Zwei Sensordaten eines Mikrocontrollers drahtlos zur Smartphone-App Phyphox übertragen und gegeneinander auftragen

Ein einfaches Beispiel mit einem Magnetfeld- und Time-of-Flight-Abstandssensor

Alexander Pusch – Max Loch – Sebastian Staacks

Der Beitrag zeigt, wie Messdaten zweier Sensoren von einem bluetoothfähigen Mikrocontroller in Echtzeit an die Smartphone-App Phyphox übertragen und dort gegeneinander geplottet werden können. Dazu kombinieren wir einen Magnetfeldsensor mit einem Time of Flight Abstandssensor, erläutern die zugrunde liegenden Messprinzipien und stellen ein einfaches Experiment vor.

1 Einleitung

Wenn Messwerte mit Mikrocontrollern aufgenommen werden, trifft man oft auf das Problem der Darstellung. In dem Artikel von Dorsel et al. (2023) haben wir dazu eine sehr einfache und universelle Möglichkeit vorgestellt, einzelne Messwerte eines beliebigen Sensors von einem bluetoothfähigen Mikrocontroller wie bspw. dem ESP32 in Echtzeit an die App Phyphox zu übertragen und dort grafisch darzustellen und auszuwerten.

In diesem Artikel möchten wir darauf aufbauend demonstrieren, wie die Messwerte zweier Sensoren übertragen und in der App Phyphox gegeneinander aufgetragen werden können. Dazu stellen wir die technischen und physikalischen Hintergründe eines Magnetfeldsensors (Anisotroper Magnetowiderstand) und Abstandssensors (Time-of-Flight) vor, die in einem einfachen Experiment kombiniert werden. Weiterhin wird anhand von Codebeispielen erläutert, wie Messwerte mit der Phyphox-Bibliothek übertragen und in der App angezeigt werden können.

Es gibt eine große Anzahl an (Low-Cost-)Sensoren, die man mit Mikrocontrollern auslesen kann. Die Art der Anbindung an den Mikrocontroller kann sich dabei ganz grundsätzlich unterscheiden. Manche Sensoren stellen sogar mehrere Möglichkeiten zur Verfügung. Nachfolgend werden einige geläufige Varianten vorgestellt:

- *Variable Spannung:* Der *Temperatursensor TMP36* ist im Grunde ein Transistor, der eine von der Temperatur linear abhängige Ausgangsspannung ausgibt. Diese Spannung kann mit dem Analog-Digital Wandler des Mikrocontrollers eingelesen und anschließend in einen Wert für die Temperatur umgerechnet werden.
- *Zeitdauer eines Signals:* Der *Ultraschallsensor HC-SR04* gibt die Laufzeit einer reflektierten Schallwelle aus (wie in Dorsel et al., 2023 näher behandelt). Dafür wird ein PIN am Sensor für diese Zeitdauer auf HIGH gesetzt. Der Mikrocontroller misst, wie lange dort ein HIGH-Signal anliegt und berechnet anschließend aus der Laufzeit den Abstand.
- *SPI-Bus:* Der *barometrische Sensor GY-BMP280* lässt sich u.a. mit dem SPI-Bus (Serial Peripheral Interface) auslesen. Um den SPI-Bus zu nutzen, werden vier Kabel an den Mikrocontroller angeschlossen. Die Sensoren müssen zusätzlich auch mit Spannung (VCC und GND) versorgt werden, weswegen sechs Kabel benötigt werden. Anders als bei den ersten beiden Möglichkeiten werden bei Bus-Systemen die Daten kodiert übertragen. Die dazu notwendigen Befehle für die Kommunikation übernehmen Bibliotheken. Mehrere Geräte können im SPI-Bus parallel angeschlossen werden, aber jedes einzelne Gerät braucht ein weiteres separates Kabel für Chip Select (CS), da keine Adressierung im Bus vorgenommen wird.
- *1-Wire-Bus:* Der *Temperatursensor DS18B20* besitzt drei Anschlüsse. Neben GND, VCC gibt es einen Kanal für Daten. Das Signal ist kodiert und liegt zwischen dem Datenkanal und dem GND an. Mehrere Sensoren können parallel geschaltet werden, diese besitzen alle eine eindeutige, einzigartige Adresse. Die Kommunikation erfolgt asynchron.
- *I²C-Bus:* Gegenüber dem 1-Wire-Protokoll ist das I²C-Protokoll (Inter-Integrated Circuit) u.a. Zeit-synchronisiert und erlaubt höhere Datenraten. Die Adressen von I²C-Geräten sind nicht einzigartig, können aber meist durch das Setzen von Brücken geändert werden. Neben GND und VCC werden zwei weitere Anschlüsse für Daten (SDA) und Takt zur Synchronisation (SCL) verwendet. Die nachfolgend verwendeten Sensoren können mit dem I²C-Bus angesteuert werden. Sie besitzen von Werk aus jeweils unterschiedlichen Adressen und können daher an einem I²C-Bus ausgelesen werden.

Kasten 1. Übersicht über Varianten zum Auslesen von Sensoren

2 Untersuchung der magnetischen Feldstärke in Abhängigkeit zum Abstand

In einem einfachen Experiment soll nun die Größe der magnetischen Feldstärke eines Neodym-Magneten in Abhängigkeit vom Abstand bestimmt werden. Nachfolgend werden die verwendeten Sensoren, die Schaltung, die Übertragung in Phyphox sowie ein mögliches Messergebnis vorgestellt. Beide Sensoren verwenden den I²C-Bus und sind daher vergleichsweise einfach anzuschließen und durch den Programmcode anzusteuern. Der Kasten 1 beschreibt diese und weitere geläufige Möglichkeiten für das Auslesen von Low-Cost-Sensoren genauer.

2.1 Time-of-Flight Abstandssensor VL530X

Der *VL530X* misst den Abstand optisch und nicht, wie bspw. der *HC-SR04*, akustisch (zur Funktionsweise siehe Dorsel et al, 2023). Der Sensor erzeugt dafür mit einem Klasse 1 Laser ein Lichtsignal mit einer Wellenlänge von 940 nm im nahen Infrarotbereich (Datenblatt VL530, o.D.), welches für das menschliche Auge nicht mehr sichtbar ist. Da die Lichtgeschwindigkeit in Luft bekannt ist, kann der Sensor aus der gemessenen Laufzeit einen Abstand bis zu zwei Meter bestimmen.
Der Betrieb erfolgt über die Einbindung der zugehörigen Bibliothek in den Programmcode. Dabei können auch verschiedene Einstellungen für den Sensor (z.B. höhere Genauigkeit, Auslesерate) gewählt werden.
Interessant aus physikalischer Sicht ist u.a. der Einfluss verschiedener Oberflächen, Farben und Materialien auf die Messung. Während der ultraschallbasierte *HC-SR04* bspw. Schwierigkeiten mit stark schallabsorbierenden Oberflächen (bspw. einen Schwamm) hat, kann es bei der Verwendung von Licht im Infrarotbereich zu Schwierigkeiten bei Rauch und Nebel, zusätzlicher IR-Einstrahlung (z.B. durch die Sonne) sowie auch stark IR-absorbierenden Oberflächen geben. Lernende können diese Einflüsse in Experimenten untersuchen.

2.2 Magnetfeldsensor QMC5883L

Der *QMC5883L* ist ein Drei-Achsen-Magnetsensor, der u.a. auf dem Breakoutboard *GY-271* verbaut ist. Er kann einen Bereich von ±0,8 mT abdecken (Datenblatt QMC5883L, o.D.).

Das Funktionsprinzip entspricht dabei aber nicht dem einer Hallsonde, wo ein Strom durch eine leitende Platte im Magnetfeld seitlich durch die Lorentzkraft abgelenkt wird und eine resultierende Hallspannung gemessen werden kann. Stattdessen wird der Effekt des Anisotropen Magnetowiderstands (AMR) ausgenutzt. Bei diesem ist der elektrische Widerstand aufgrund von Spin-Bahn-gekoppelten Streuprozessen raumrichtungsabhängig vom äußeren Magnetfeld. D.h. je nach Stärke und Richtung des äußeren Magnetfeldes ändert sich der Widerstand. Der Sensor enthält vier solcher dünnschichtigen AMR-Widerstände, die mit einer Wheatstone-Brücke ausgelesen werden. Eine Wheatstone-Brücke ist vom Funktionsprinzip ein Spannungsteiler und wird eingesetzt, um kleine ohmsche Widerstandsänderungen zu messen.

2.3 Schaltung

Das benötigte Material für die Schaltung ist im Kasten 2 aufgelistet. Beide Sensoren, VL530X und QMC5883L, werden über einen I²C-Bus ausgelesen. Um im Experiment den Abstand und

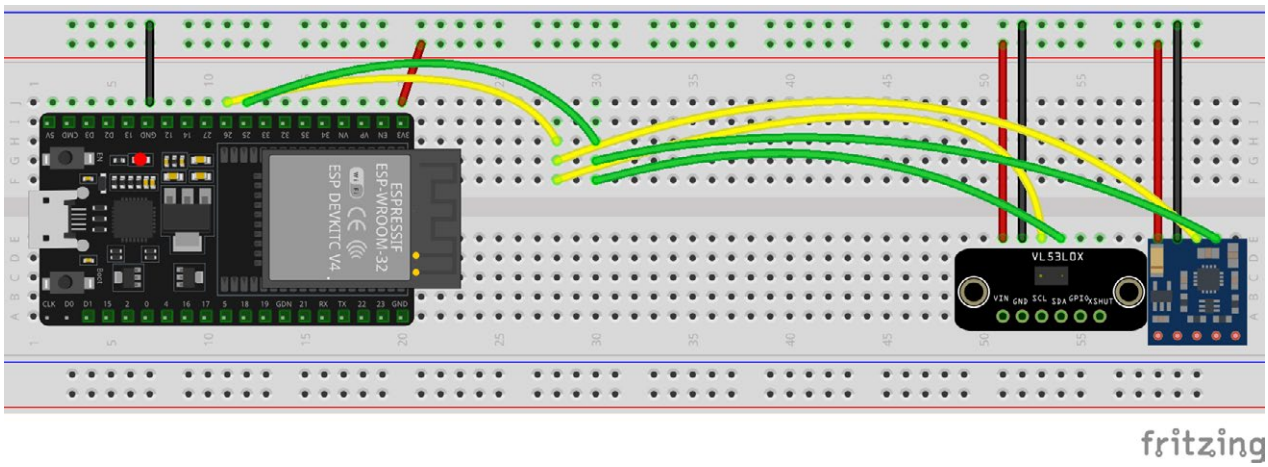


Abb. 1. Schaltung des ESP32 mit dem VL530X und dem QMC5883L. Achten Sie auf die übereinstimmende Beschriftung der Ausgänge beim Aufbau der Schaltung.

die magnetische Feldstärke möglichst gut erfassen zu können, werden beide Sensoren so dicht wie möglich zusammen und an den Rand des Boards gesteckt.

Die zu verwendenden Anschlüsse an den Sensoren heißen beim I²C-Bus SDA und SCL und werden am ESP32 an den Pins 25 (SDA) sowie 26 (SCL) verbunden (Abb.1). Bei manchen Mikrocontrollern, wie dem Arduino UNO, sind die Ports für bestimmte Bussysteme vorgegeben. Bei anderen Mikrocontrollern, wie dem ESP32, können verschiedene Pins hierfür im Programmcode definiert und verwendet werden. Die Sensoren werden außerdem jeweils mit 3,3 V und GND versorgt.

- Steckbrett (groß)
- 12 Jumperkabel Male-Male
- 3-Axis Compass QMC5883L (auf Breakout-Board GY-271)
- VL53L0X Time-of-Flight (ToF)
- ESP32 z.B. Dev Kit V4 oder Node MCU
- Neodym-Stabmagnet Ø 4 mm, Höhe 12,5 mm
- 3D-gedruckte Halterung

Kasten 2. Material

2.4 Nutzung der Phyphox-Bibliothek

In Dorsel et al. (2023) wurde bereits die grundlegende Funktionsweise der Phyphox-Bibliothek erklärt und ein einfaches Beispiel mit dem Ultraschallabstandssensor HC-SR04 vorgestellt. In diesem Beitrag erklären wir in kurzer Form das Übertragen von mehreren Messwerten. Mehr über die Verwendung von Bluetooth Sensoren in Phyphox findet sich außerdem in Staacks et al. (2025).

Der Programmcode 1 zeigt die grundlegenden Befehle, um die Phyphox-Bibliothek zu initialisieren und ein Experiment anzulegen.

Im Programmcode des Mikrocontrollers wird außerdem vorab definiert, wie die zu übertragenen Werte von der App Phyphox dargestellt werden sollen. Neben der Anzeige von Messwerten, können diese auch in einem Graphen visualisiert werden. Programmcode 2 zeigt einen Ausschnitt, in dem u.a. ein Graph, wie z.B. in der Abbildung 3 gezeigt, angelegt sowie die Beschriftungen und Skalierungen der Achsen definiert werden. Es wird in der letzten Zeile im Programmcode 2 ebenfalls schon festgelegt, welche der später übertragenen Daten in dem Graph dargestellt werden sollen. In diesem Fall sind es die “Kanäle” 1 und 2. Mit dem Programmbefehl aus Programmcode 3 werden

```
//Starten der PhyphoxBLE Umgebung
PhyphoxBLE::start("ESP32 Magnetfeldstärke");
//Erstellung einer Instanz „Experiment".
PhyphoxBleExperiment experiment;
//Setzen von Titel etc. für die Darstellung in der Phyphox App.
experiment.setTitle("Magnetische Feldstärke mit Abstandsmessung");
experiment.setCategory("Externer Sensor");
experiment.setDescription("In diesem Experiment wird die Magnetische Feldstärke gegen den Abstand aufgetragen.");
```

Programmcode 1. Initialisierung der phyphox-Bibliothek

```
//Graph 1 X-Achse
//Setzen von Eigenschaften des Graphen
PhyphoxBleExperiment::Graph firstGraph;
//Setzen der Minima und Maxima des Graphen und ob dieser bei Bedarf expandieren soll oder nicht.
firstGraph.setMinY(-500, LAYOUT_FIXED);
firstGraph.setMaxY(6500, LAYOUT_EXTEND);
firstGraph.setMinX(0, LAYOUT_FIXED);
firstGraph.setMaxX(400, LAYOUT_FIXED);

firstGraph.setLabel("x-Komponente");
firstGraph.setUnitX("mm");
firstGraph.setUnitY("mG");
firstGraph.setLabelX("Distanz");
firstGraph.setLabelY("Mag. Feldstärke");
firstGraph.setXPrecision(1);          //Anzahl Nachkommastellen der Achsenbeschriftung
firstGraph.setYPrecision(1);
firstGraph.setStyle(STYLE_DOTS);      //Datenpunkte werden als Punkte eingetragen.
firstGraph.setChannel(1, 2);          //Die Daten der Kanäle 1 und 2 werden mit diesem Graphen assoziiert.
```

Programmcode 2. Anlegen der Graphen für die Ansicht in der App phyphox

```
//Die Werte werden an Phyphox übergeben.
PhyphoxBLE::write(distance, intensityX, intensityY, intensityZ);
```

Programmcode 3. Übertragen der Werte an die App phyphox

an Phyphox vier Variablen (distance, intensityX, intensityY, intensityZ) in "Kanälen" übermittelt. Dadurch werden in dem zuvor definierten Graphen die Variablen distance (Messwert des Abstandssensors) und intensityX (Feldstärke des Magnetfelds entlang der x-Achse) gegeneinander aufgetragen. Soll etwas gegen die Zeit aufgetragen werden, ist "Kanal" 0 dafür zu wählen.

Der weitere Programmcode definiert zusätzliche Ansichten und Graphen. Es wird an dieser Stelle u.a. nicht auf die Befehle zum Einbinden von Bibliotheken und Anlegen von Variablen eingegangen. Diese sind im vollständigen Programmcode unter https://phyphox.org/p/mnu-arduino zu finden. Das Initialisieren der Sensoren sowie auch das Auslösen der Messungen geschieht vergleichsweise „unspektakulär" über die in Programmcode 4 gezeigten Befehle.

2.5 Durchführung und Ergebnis

Das Experiment kann als Freihandexperiment durchgeführt werden. Eine 3D-gedruckte Halterung für einen Magneten (rot, Abb. 2), die an der Tischkante entlang auf die beiden Sensoren auf dem Breadboard zu bewegt werden kann, hilft bei der gleichbleibenden räumlichen Positionierung des Magneten und ermöglicht eine Reflexion des IR-Signals an einer Ebene.

```
distance = distSensor.readRangeSingleMillimeters();
magSensor.read();
intensityX = abs(magSensor.getX());
intensityY = abs(magSensor.getY());
intensityZ = abs(magSensor.getZ());
```

Programmcode 4. Aufnahme der Messwerte

In der App Phyphox lässt sich mit “+” und “Neues Experiment für Bluetoothgerät” die Verbindung mit dem ESP (in dem Fall mit dem Namen “ESP32 Magnetfeldstärke”) herstellen. Nach dem Starten der Messwertaufzeichnung über “▶“ lässt sich durch Bewegung des Magnetes auf die Sensoren eine Kurve wie in Abbildung 3 erzeugen.

Erkennbar ist ein inverser Potenz-Zusammenhang zwischen der magnetischen Feldstärke B und dem Abstand in x-Richtung. Dieser Zusammenhang ist in der Theorie $B \propto 1/x^3$. Bei sehr geringen Abständen vom Magneten zum Sensor (in unserem Fall < 5 mm) kommt es zu Effekten der Sättigung und dadurch zu ungültigen Messwerten.

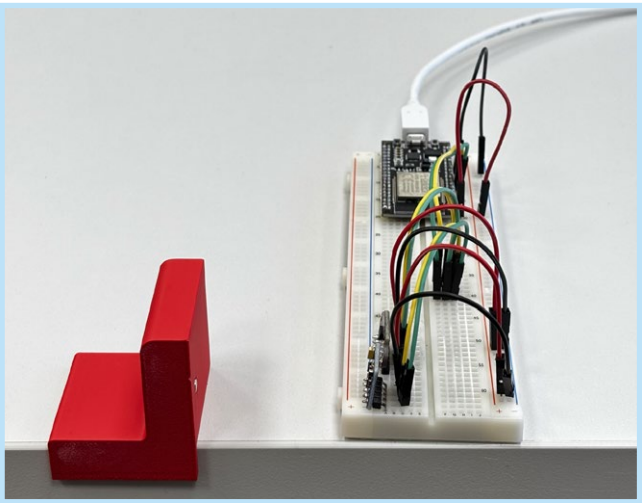


Abb. 2. Aufbau des Experiments

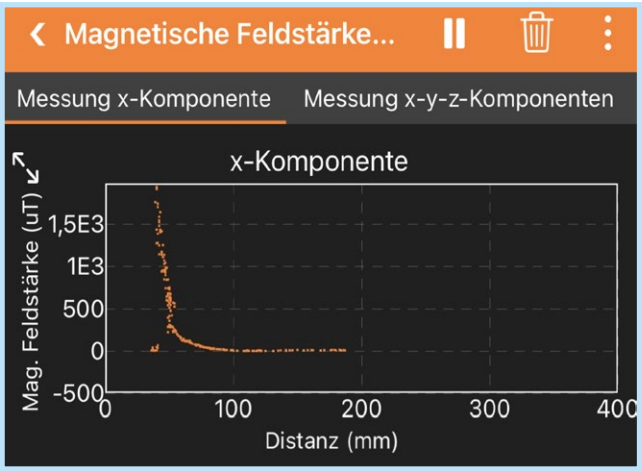


Abb. 3. Screenshot der Messung aus Phyphox. Zu sehen ist der Auftrag der magnetischen Feldstärke in x-Richtung gegen die Distanz in mm.

3 Ausblick

Die Phyphox-Bibliothek ermöglicht es, die Messdaten nahezu beliebiger Sensoren einfach über Bluetooth Low Energy (BLE) am Smartphone in Phyphox darzustellen. Dadurch, dass Phyphox den Lernenden oftmals bereits bekannt ist, stellt sie ein ideales Visualisierungswerkzeug für externe Sensordaten dar. Benötigt wird für die Übertragung ein bluetoothfähiger Mikrocontroller wie z.B. ein ESP32 oder ein Arduino Nano 33 BLE.

In diesem Artikel wurde gezeigt, wie mehrere Sensorwerte aufgenommen und in Phyphox dargestellt werden können. Weitere Ideen für Experimente mit zwei Sensoren sind z.B. die Untersuchung des Licht-Abstand-Gesetztes, Temperatur-Abstand-Gesetztes oder die Bestimmung des Maximum-Power-Points (MPP) beim Betrieb einer Solarzelle (s.a. Pusch et al., 2021).

Der ausführlich kommentierte Programmcode sowie die STL-Datei der Halterung sind unter https://phyphox.org/p/mnu-arduino zu finden.

Literatur

Datenblatt VL530 (o.D.) *https://www.st.com/resource/en/datasheet/vl53l0x.pdf*.

Datenblatt QMC5883L (o.D.). *http://wiki.sunfounder.cc/images/7/72/QMC5883L-Datasheet-1.0.pdf*.

Dorsel, D., Staacks, S., Loch, M. & Pusch, A. (2023). Sensordaten drahtlos zur Smartphone-App phyphox übertragen und grafisch auswerten – ein einfaches Beispiel mit dem ESP32 und dem Ultraschallsensor HC-SR04. *MNU-Journal, 76*, 36–43.

Pusch, A., Ubben, M., Laumann, D., Heinicke, S. & Heusler, S. (2021). Real-time data acquisition using Arduino and phyphox: measuring the electrical power of solar panels in contexts of exposure to light in physics classroom. *Physics Education, 56*, 1–13. *https://doi.org/10.1088/1361-6552/abe993*.

Staacks, S., Dorsel, D., Krampe, A., Hagedorn, M., Leier, E., Heinke H. & Stampfer, C. (2025). Bluetooth sensors in phyphox with Arduino and MicroPython: paving the way from an idea to an experiment for teachers and learners. *Physics Education, 60*(3), 035014, *https://doi.org/10.1088/1361-6552/adbf5f*.

Staacks, S. & Pusch, A. (2025). Experimentieren mit Phyphox. Einfache Alltagsexperimente und „klassische“ Experimente aus dem Physikunterricht mit dem Smartphone. *MNU-Journal, 78*, 320–329. *https://physikkommunizieren.de/wp-content/uploads/2025/07/Staacks-Pusch-2025-Experimentieren-mit-Phyphox.pdf*.

Dr. Alexander Pusch, alexander.pusch@uni-muenster.de, lehrt und forscht am Institut für Didaktik der Physik an der Westfälischen Wilhelms-Universität Münster.

Maximilian Loch war wissenschaftlicher Mitarbeiter am Institut für Didaktik der Physik an der Westfälischen Wilhelms-Universität Münster und wechselte 2026 in den Schuldienst.

Dr. Sebastian Staacks, staacks@physik.rwth-aachen.de, entwickelt die App „Phyphox“ am 2. Physikalischen Institut A der RWTH Aachen.