Mit Sensor auf Kollisionskurs – ein Schüler:innenexperiment zur DART-Mission

In Bezug auf den Beitrag: „Raumsonde DART verändert Asteroiden stärker als erwartet“ in der Zeitschrift „Sterne und Weltraum“ 10/25, Rubrik: Nachrichten, S. 12, WIS-ID: 1571300, Zielgruppe: Mittel- und Oberstufe.

Katharina Supp

Die NASA-Mission **DART** (**D**ouble **A**steroid **R**edirection **T**est) hatte 2022 das Ziel, durch einen gezielten Einschlag die Umlaufbahn des Asteroidenmondes Dimorphos minimal zu verändern. Die 570 kg schwere Sonde traf mit einer effektiven Geschwindigkeit von 6,6 km/s auf ihr Ziel und verkürzte die Umlaufbahn von Dimorphos um etwa 33 Minuten – ein Meilenstein für die Planetary Defense.
In der hier vorgestellten Unterrichtseinheit simulieren Schüler:innen die physikalische Wirkung einer gezielten Kollision und analysieren diese mithilfe eines Beschleunigungssensors (MPU6050) und eines Arduinos. Im Modellversuch rekonstruieren die Lernenden vergleichbare Stoßszenarien: Ein Projektil wird aus definierter Höhe auf eine sensorbestückte Plattform fallen gelassen. Der Verlauf der Beschleunigung beim Stoß wird dabei mithilfe eines selbstgeschriebenen Programms automatisch erfasst. Ziel ist es, Stärke und Dauer der Stoßbelastung zu analysieren, den Einfluss von Fallhöhe und Material zu untersuchen und physikalische Erklärungsmodelle zu entwickeln. Die Einheit verknüpft Mechanik mit aktueller Sensorik und eröffnet einen experimentellen Zugang zur modernen Raumfahrtforschung.

Übersicht der Bezüge im WiS-Beitrag		
Astronomie	**Kleinkörper, Raumfahrt**	**Asteroid, planetare Verteidigung, NASA, DART-Mission**
Physik	**Mechanik Experiment**	**Impuls, kinetische Energie, Stoß, freier Fall, Beschleunigung Datenerfassung, digitale Messsysteme, Auswertung von Messdaten, Fehlerdiskussion, Diagramme, Messgrenzen**
Fächerverknüpfung	**Astro-Informatik Astro-Technik Naturwissenschaft und Technik**	**Mikrocontroller (Arduino), Programmieren, Sensoren, MPU6050, Aufbau von Sensorsystemen**
Lehrer:innen allgemein	**Kompetenzen Unterrichtsmittel Lehr-/Sozialformen**	**Arbeitsblätter, Aufgaben, Video, Lesetexte, Programmiercodes, Projektarbeit, Teamarbeit, Binnendifferenzierung**

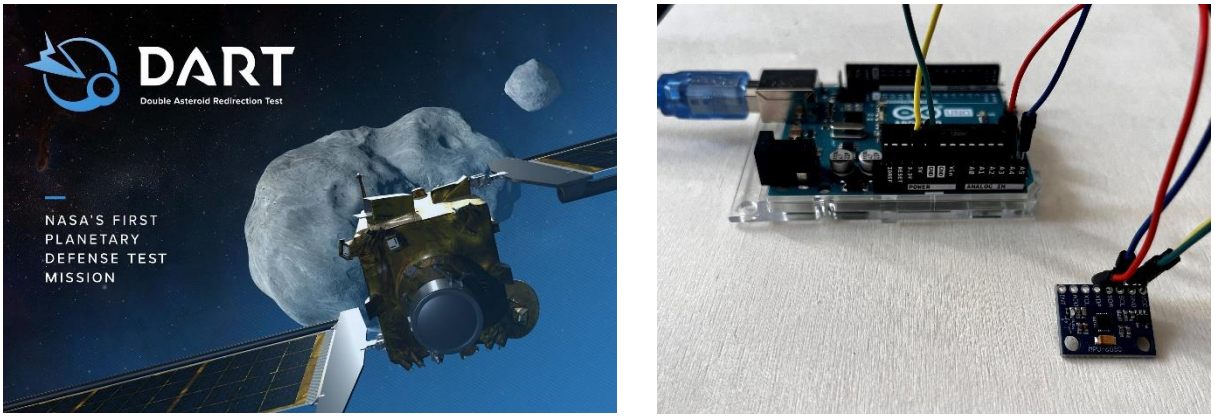


Abb. 1: Ein Schüler:innenexperiment zur DART-Mission.
Bild (links): NASA/Johns Hopkins APL, https://dart.jhuapl.edu/Gallery/media/graphics/lg/DART-poster-graphic_web.jpg (Public Domain, USA)
Bild (rechts): Katharina Supp

Hinweise für Lehrkräfte

1. Einsatz des Materials im Unterricht

Das Material ist für die Mittel- oder Oberstufe konzipiert und eignet sich als Projektarbeit im Physik- und NwT-Unterricht. Grundlegende Vorkenntnisse im Umgang mit Arduino (z. B. Anschluss eines Sensors, Arbeiten mit der Arduino-IDE, einfache Programmstruktur) sind dafür erforderlich. Die beigefügten Missionshilfen ermöglichen differenziertes Arbeiten in heterogenen Lerngruppen. Der Zeitumfang beträgt je nach Klassenniveau und Vertiefung etwa sechs bis acht Unterrichtsstunden.

2. Aufbau des Materials

Tab. 1: Überblick über die einzelnen Arbeitsblätter des Schüler:innenprojekts

AB 1: Die DART-Mission	**Einführung:** Einstiegsimpuls, z.B. Live-Stream der letzten Minuten vor dem Aufprall der Sonde[1] Lesetext mit Aufgaben zur DART-Mission Projektvorstellung
AB 2: Physikalischer Hintergrund: Stöße	**Wiederholung/physikalischer Hintergrund:** grundlegende Formeln zum Thema „Stöße“ Beispielaufgaben
AB 3: Der MPU6050-Sensor	**Kennenlernen des Sensors:** Lesetext mit Aufgabe zum Funktionsprinzip des MPU6050-Sensors Schaltskizze zum Anschließen des Sensors an Arduino Testprogramm (Basiscode) und Aufgaben zum Testen und Kennenlernen des Sensors
AB 4: Ein Programm zur Impact-Messung entwickeln	**Sensor programmieren:** Aufgaben zur schrittweisen Erweiterung des Basiscodes Erweiterung um Gesamtbeschleunigung, Trigger und Messdauer
AB 5: Teststation bauen, Messung durchführen und auswerten	**Modellversuch durchführen:** Bauanleitung Sensorplattform Versuchsdurchführung mit automatischer Messwerterfassung (MPU6050-Sensor + Arduino) Tabelle zur Auswertung des Versuchs Aufgaben zur Verknüpfung des Modellversuchs mit der DART-Mission/Ausblick

3. Vorbereitung durch die Lehrkraft

Technische Einrichtung

- Arduino-IDE auf allen Schüler:innen-PCs installieren
- folgende Bibliotheken einbinden:
  - Adafruit_MPU6050
  - Adafruit_Sensor

[1] Video verfügbar unter: https://dart.jhuapl.edu/Gallery/media/videos/dart_impact_replay.mp4 (zuletzt aufgerufen am 12.08.2025)

  - Wire (standardmäßig in der IDE enthalten)
- Basiscode auf allen Rechnern bereitstellen oder zum Abtippen ausdrucken

Materialbereitstellung

- Technik und Sensorik:
  - Arduino
  - MPU6050-Beschleunigungssensor
  - Jumperkabel
  - Computer mit Arduino-IDE und Programm zur Tabellenkalkulation
- Messaufbau:
  - Sensorplattform (z. B. dünne Holzplatte)
  - Kugeln (Holz o. ä.)
  - verschiedene Aufprallunterlagen: Moosgummi, Schaumstoff
  - Klebeband
  - Maßband

Unterrichtsmaterialien

- Arbeitsblätter
- Missionshilfen (beide enthalten im Material)

4. Hinweise zur Durchführung

Sensorbefestigung:

Die Sensorplattform muss stabil fixiert werden (z. B. mit Klebeband). Ein zentraler, senkrechter Treffer der Kugel über dem Sensor ist für verwertbare Messwerte entscheidend.

Messqualität:

Die Ergebnisse sind stark abhängig von:

- Fallhöhe
- Material des Untergrunds
- Art und Masse der Kugel
- Genauigkeit der Durchführung

Differenzierung:

Erfahrungsgemäß fällt einigen Schüler:innen das Programmieren schwer. Um Frust zu vermeiden, wird den Schüler:innen ein Basiscode zur Verfügung gestellt, den sie angeleitet um eine Triggergrenze und eine definierte Messdauer erweitern. Dabei bieten die Missionshilfen zusätzliche Unterstützung im Sinne der Binnendifferenzierung.
Mit dem fertigen Programm sind die Schüler:innen dann in der Lage, die Messwerte des Stoßversuchs automatisch zu erfassen.

Sensorgrenze:

Der MPU6050-Sensor misst maximal ±16g. Diese Grenze wird bei senkrechtem Fall (v. a. bei harten Unterlagen) schon nach wenigen Zentimetern erreicht. Das kann zur Sättigung führen (Werte werden nicht mehr korrekt erfasst).
Der MPU6050 wurde trotz dieser Einschränkung bewusst gewählt, da er sehr kostengünstig, leicht verfügbar und gut dokumentiert ist.

Arbeitsblätter

1 Die DART-Mission

Stell dir vor, ein Asteroid befindet sich auf Kollisionskurs mit der Erde. Was tun?
Um auf diesen Fall vorbereitet zu sein, haben die NASA, die ESA und andere Raumfahrtorganisationen spezielle Programme zur ***planetaren Verteidigung***. Diese umfassen alle Maßnahmen und Pläne, um die Erde vor potenziellen Bedrohungen durch extraterrestrische Objekte wie Asteroiden oder Kometen zu schützen.
Das ***Planetary Defense Coordination Office* (*PDCO*)** der NASA wurde 2016 gegründet. Es überwacht sogenannte ***Near-Earth Objects* (*NEOs*)**, also Asteroiden und Kometen, die sich der Erde nähern können. Ziel ist es, mittels Himmelsüberwachung möglichst früh zu erkennen, ob ein Objekt der Erde gefährlich werden könnte, um es gegebenenfalls abzuwehren.
Der Bedarf an planetarer Verteidigung ist real: Wissenschaftler:innen schätzen, dass es rund 25.000 Asteroiden im Sonnensystem gibt, die groß genug wären, um eine Stadt zu zerstören. Bisher sind etwa 8.000 davon bekannt. Zwar droht aktuell kein direkter Einschlag, doch die Vergangenheit zeigt, dass Einschläge möglich sind: So geht man davon aus, dass das Massenaussterben der Dinosaurier vor etwa 66 Millionen Jahren durch den Einschlag eines etwa 10 km großen Asteroiden nahe der heutigen Halbinsel Yucatán verursacht wurde (***Chicxulub-Krater***). Auch Ereignisse wie das ***Tunguska-Ereignis*** 1908 in Sibirien oder das ***Chelyabinsk-Ereignis*** 2013 zeigen, dass kosmische Einschläge reale Gefahren darstellen.
Zurzeit werden verschiedene Abwehrmethoden diskutiert und erforscht, unter anderem:

- ***kinetische Ablenkung***: Ein Objekt (z. B. Raumsonde) kollidiert mit dem Asteroiden, um seine Bahn leicht zu verändern.
- ***Gravitational Tractor***: Ein Raumfahrzeug fliegt in der Nähe des Asteroiden und zieht ihn durch seine Gravitation über längere Zeit aus der Bahn.
- ***nukleare Sprengung***: Eine gezielte Explosion nahe oder auf dem Asteroiden könnte seine Flugbahn verändern, jedoch ist dies aufgrund der Verwendung von nuklearen Sprengkörpern technisch und politisch sehr umstritten.

Die ***DART***-Mission (***D****ouble* *A****s****teroid* ***R****edirection* ***T****est*) der NASA war der erste praktische Versuch, mit einem gezielten Einschlag die Bahn eines Asteroiden messbar zu verändern.
Ziel von DART war das Asteroiden-Doppelsystem ***Didymos–Dimorphos***, das sich in sicherer Entfernung von der Erde befindet.

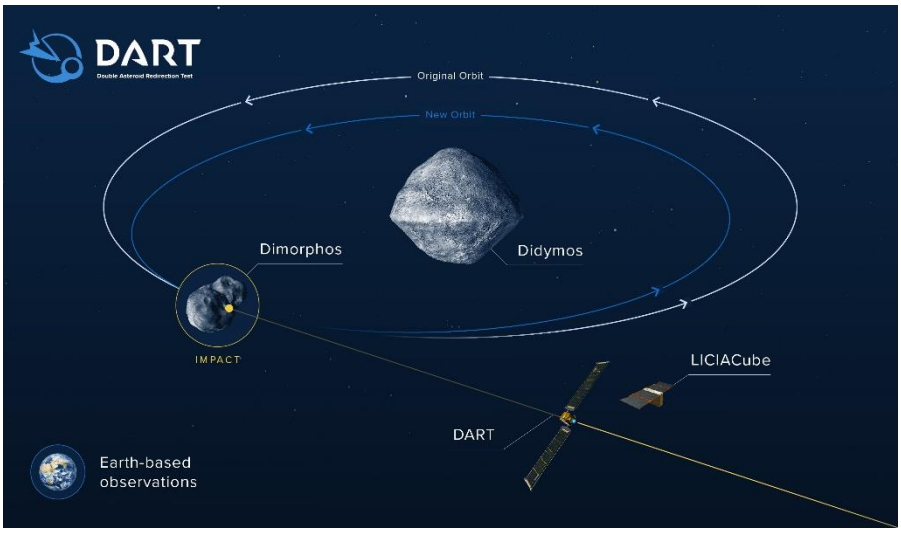


Abb. 2: DART-Mission. Einfluss der Kollision auf die Umlaufbahn von Dimorphos.
Bild: NASA/John Hopkins, https://dart.jhuapl.edu/Gallery/media/graphics/lg/DART-infographic_v4.jpg (Public Domain, USA)

Didymos ist ein Asteroid mit einem Durchmesser von etwa 780 m. Er besitzt einen kleineren Begleiter: Dimorphos, ein rund 160 m großer Mond, der Didymos in etwa 12 Stunden umkreist.
Am 26. September 2022 prallte die 570 kg schwere Raumsonde DART mit effektiv etwa 6,6 km/s (23.700 km/h) auf Dimorphos und verkürzte dessen Umlaufzeit um ganze 33 Minuten. Die Bahnablenkung war ein historischer Erfolg. Zum ersten Mal überhaupt wurde die Bahn eines Himmelskörpers durch menschliches Eingreifen messbar verändert (vgl. Abb. 2).
Die detaillierte Analyse zeigte jedoch, dass der Einschlag komplexer verlief als erwartet. Statt eines einfachen Kraters kam es zu starken Verformungen, massiver Materialverlagerung und einem erheblichen Auswurf von Trümmern. Diese herausgeschleuderten Brocken verstärkten die Ablenkung des Asteroiden deutlich: Der Impuls des Auswurfs war mehr als dreimal so groß wie der der Sonde selbst. Insgesamt konnten über hundert Felsbrocken mit Geschwindigkeiten bis zu 52 m/s beobachtet werden, die sich in klar abgegrenzten Clustern verteilten. Vermutlich kollidierten dabei bereits die Sonnenkollektoren der Sonde mit größeren Oberflächenbrocken. Solche Erkenntnisse verdeutlichen, wie stark die Oberflächenbeschaffenheit die Dynamik eines Einschlags beeinflusst und liefern wichtige Grundlagen für künftige Abwehrstrategien – Erkenntnisse, die mit der ESA-Mission ***Hera*** ab 2026 weiter vertieft werden sollen.

Abb. 3: Video der letzten fünfeinhalb Minuten vor der Kollision der DART-Sonde mit Dimorphos. Die DART-Sonde übertrug die Bilder von seiner DRACO-Kamera in Echtzeit zur Erde, während er sich dem Asteroiden näherte.
Video: NASA/John Hopkins APL, https://dart.jhuapl.edu/Gallery/media/videos/dart_impact_replay.mp4 (Public Domain, USA)

Aufgaben:

1.1) Beschreibe, welche Ziele die DART-Mission verfolgt und ordne sie in den Zusammenhang der Planetary-Defense ein.
1.2) Erkläre, warum Dimorphos als Zielobjekt besonders geeignet war.
1.3) Erläutere, welche Rolle die Oberflächenstruktur des Asteroiden bei der Wirkung des Einschlags spielt.
1.4) Informiere dich über das ***Tunguska-Ereignis*** und das ***Chelyabinsk-Ereignis*** und fasse die Ergebnisse deiner Recherche stichpunktartig zusammen.

Von der Raumfahrt ins Klassenzimmer – dein eigener DART-Test
Die DART-Mission war ein aufwendiges Weltraumprojekt mit Millionenbudget, internationaler Zusammenarbeit und jahrelanger Vorbereitung. Doch die dahinterstehende physikalische Frage ist erstaunlich grundlegend:
Was passiert, wenn ein Objekt mit hoher Geschwindigkeit auf ein anderes trifft?
In einem eigenen Versuch gehst du dieser Frage selbst nach. Du programmierst einen MPU6050-Sensor, um damit die Beschleunigung beim Aufprall zu messen. Dann lässt du eine Kugel aus unterschiedlichen Höhen auf verschiedene Unterlage fallen und nimmst die gemessenen Beschleunigungen automatisch auf. Die Art des Untergrunds – z. B. Holz, Moosgummi oder Schaumstoff – beeinflusst, wie stark und wie schnell die Kugel abgebremst wird.
Wie bei DART weiß man im Voraus nicht genau, wie sich das Material verhält. Auch hier hängt die Wirkung des Stoßes von der Oberflächenbeschaffenheit ab. Mit deiner Messung erfährst du also mehr über grundlegende physikalische Zusammenhänge und bekommst gleichzeitig ein Gefühl dafür, wie komplex reale Raumfahrtmissionen sind.

WICHTIG: Bei deiner eigenen Mission bist du nicht auf dich allein gestellt. Bei vielen Aufgaben stehen dir **Missionshilfen** zur Verfügung.

2 Physikalischer Hintergrund: Stöße

Bevor du mit dem Experiment beginnst, lohnt sich ein Blick auf die Physik hinter dem Einschlag. Auch wenn unser Versuch kleiner ist, gelten dieselben Prinzipien wie bei der DART-Mission. Physikalisch betrachtet funktioniert die DART-Mission nach den Gesetzen des ***Impulses p***: Wenn ein Objekt (hier: die Sonde) mit hoher Geschwindigkeit auf einen Asteroiden trifft, überträgt es seinen Impuls auf diesen. Der Asteroid wird dadurch ganz leicht beschleunigt, also in seiner Bewegung verändert. Der Stoß ist dabei nicht einfach elastisch, wie bei Billardkugeln, sondern ein komplexer Wechsel aus Verformung, Materialauswurf und Energieverlust.

1. Impuls und Stoß

Der ***Impuls p*** eines Körpers ist definiert als Produkt aus *Masse m* und *Geschwindigkeit v*:

$$p = m \cdot v$$

Beim Stoß ändert sich der Impuls in sehr kurzer Zeit. Daraus folgt die mittlere ***Kraft F***, die auftritt:

$$F = \frac{\Delta p}{\Delta t} = m \cdot \frac{\Delta v}{\Delta t}$$

2. Freier Fall/Geschwindigkeit beim Aufprall

Wenn ein Körper aus der *Höhe h* fallen gelassen wird (ohne Luftwiderstand), beschleunigt er mit der ***Erdbeschleunigung g*** = *9,81* m/s².
Die ***Geschwindigkeit v*** beim Aufprall ist:

$$v = \sqrt{2 \cdot g \cdot h}$$

3. Mittlere Abbremsbeschleunigung

Beim Aufprall wird die Kugel in sehr kurzer Zeit abgebremst. Die mittlere ***Abbremsbeschleunigung a*** hängt von der *Geschwindigkeit Δv* und der *Zeit Δt* ab, in der die Abbremsung stattfindet:

$$a = \frac{\Delta v}{\Delta t}$$

Je nach Untergrund variiert die Abbremszeit und damit die mittlere Abbremsbeschleunigung. Kurze Abbremszeiten führen zu großen Beschleunigungen, längere Abbremszeit zu geringeren Beschleunigungen.

4. Energieübertragung

Die ***kinetische Energie E_{kin}*** beim Aufprall ist:

$$E_{kin} = \frac{1}{2} \cdot m \cdot v^2$$

Diese Energie wird beim Stoß in Verformung, Wärme und Schwingung umgewandelt.

Aufgaben:

2.1) Berechne die Geschwindigkeit eines Objekts beim Aufprall für Fallhöhen von 1 cm, 3 cm und 50 cm.

2.2) Berechne, mit welcher kinetischen Energie die DART-Sonde beim Aufprall auf Dimorphos gewirkt hat. Entnimm alle benötigten Informationen aus den Infotexten.

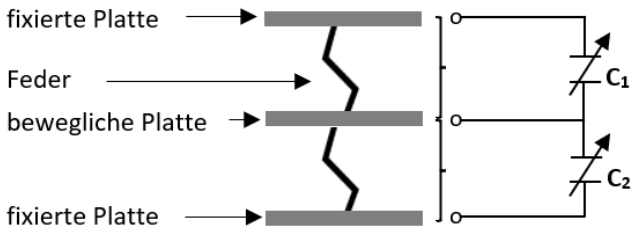
Vergleiche das Ergebnis mit der Sprengkraft von TNT (1 t TNT $\approx 4{,}2 \cdot 10^9$ J).

3 Der MPU6050-Sensor

In der DART-Mission war es entscheidend, den Einschlag genau zu analysieren: Wie heftig war die Kollision? Wie hat der Asteroid darauf reagiert? Solche Informationen erhält man über **Beschleunigungssensoren**. Auch in deinem DART-Test arbeitest du mit einem solchen System: dem **MPU6050**.
Der MPU6050 ist ein moderner Sensor, der Beschleunigungen und Drehbewegungen in drei Raumrichtungen messen kann. Er basiert auf einem sogenannten ***MEMS-System*** (***mikro-elektromechanisches System***) und wird über ein Mikrocontroller-Board (z. B. ***Arduino***) ausgelesen. Auch in Smartphones, Spielkonsolen oder Fahrassistenzsystemen steckt diese Technik und in der Raumfahrt hilft sie, Einschläge zu erfassen oder Fluglagen zu stabilisieren.
In deinem Versuch nutzt du den Sensor, um die maximale Beschleunigung beim Aufprall zu erfassen und daraus Rückschlüsse auf Abbremszeit, Impulsübertragung und Materialeigenschaften zu ziehen.

1. Beschleunigungsmessung des MPU6050-Sensors

Die Beschleunigungsmessung erfolgt mit einem dreiachsigen ***MEMS-Beschleunigungssensor***. Im Inneren dieses Sensors befindet sich für jede Raumrichtung eine winzige bewegliche Platte, die mit feinen Federstrukturen an fixierten Platten aufgehängt ist (vgl. Abb. 4). Wenn der Sensor beschleunigt wird, verschiebt sich diese Platte aufgrund des *Trägheitsprinzips* minimal. Diese Verschiebung wird durch kapazitive Messung erkannt – ähnlich wie bei einem kleinen Kondensator, dessen Abstand sich verändert. Die Elektronik misst diese Kapazitätsänderung und berechnet daraus die Beschleunigung. Mithilfe des Sensors kann man also nicht nur erkennen, ob etwas gefallen ist, sondern auch wie stark der Aufprall war.



keine Beschleunigung

$C_1 = C_2$

Abb. 4: Funktionsprinzip eines MEMS-Beschleunigungssensors.
Bild: Katharina Supp

Aufgabe:

3.1) Zeichne in die Skizze unten ein, wie sich die bewegliche Platte in z-Richtung im Beschleunigungssensor verschiebt, wenn…

a) … eine Kugel auf den Sensor nach unten einschlägt.
b) … die Kugel auf dem Sensor nach oben zurückfedert.

Wie verändert sich dabei jeweils die Kapazität C?

2. Anschließen und Testen des MPU6050-Sensors

- Arduino
- MPU6050-Beschleunigungssensor
- Jumper-Kabel
- Computer mit Arduino IDE (erforderliche Bibliotheken: Adafruit_MPU6050, Adafruit_Sensor, Wire)

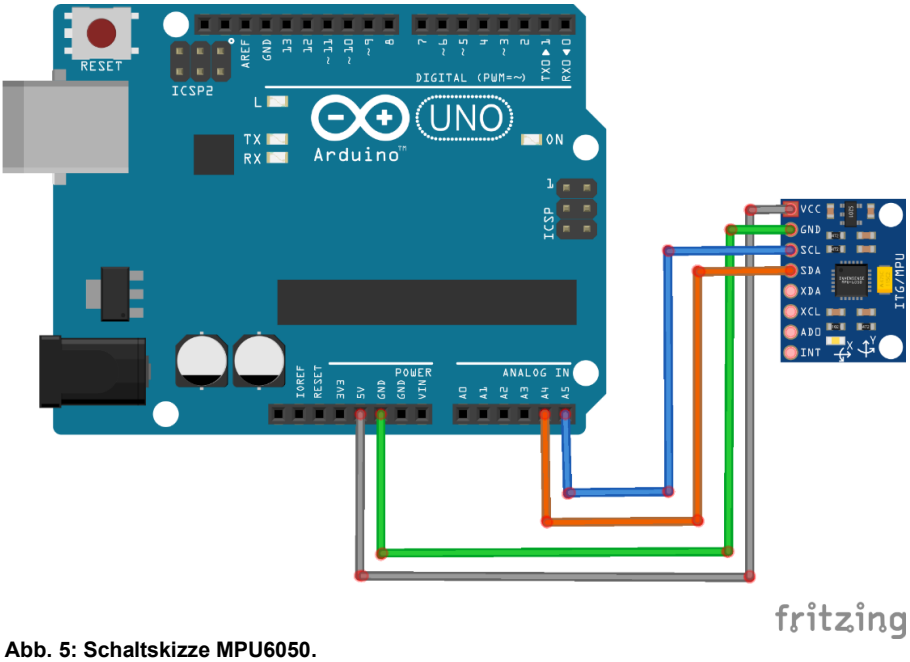


Abb. 5: Schaltskizze MPU6050.
Bild: Katharina Supp. Diese Abbildung wurde mit Fritzing erstellt.

Tab. 2: Pin-Funktionen der benutzen Pins des MPU6050-Sensors.

Pin	Pin-Funktion	Arduino Uno
VCC	Stromversorgung mit 5V	5V
GND	Masse/0V	GND
SCL	Serial clock: Gibt den Rhythmus vor, in dem die Daten zwischen Sensor und Mikrocontroller übertragen werden	A5
SDA	Serial data: Überträgt die Messwerte zwischen Sensor und Arduino Uno	A4

Aufgaben:

3.2) Schließe den MPU6050-Sensor so an, wie in Abb. 5 und Tab. 2 dargestellt.

3.3) Übertrage den Basiscode zum Testen des MPU6050-Sensors in deine IDE und öffne den seriellen Monitor.

a) Überprüfe, ob der Sensor erkannt wird. Falls nicht, überprüfe die Verkabelung.
b) Erkläre, warum eine Achse im Ruhezustand 1g anzeigt?
c) Was passiert, wenn der Sensor „kopfüber“ liegt?
d) Kippe den Sensor und beobachte, welche Achsen sich verändern.
e) Drehe den Sensor um 90°. Wo liegt jetzt die 1g?
f) Bewege den Sensor ruckartig. Beschreibe, wie die Werte jetzt aussehen.

4 Ein Programm zur Impact-Messung entwickeln

Du bist jetzt bereit, ein Programm zur Impact-Messung zu entwickeln.
Als Grundlage dient dir das zuvor getestete Basisprogramm, das du nun schrittweise erweitern und an deine Messaufgabe anpassen wirst. Dabei wirst du unter anderem die **Gesamtbeschleunigung** berechnen, einen **Triggermechanismus** einbauen und die **Messdauer** steuern.
Wenn du bei einzelnen Schritten Unterstützung benötigst, helfen dir die *Missionshilfen* weiter.

1. Die Gesamtbeschleunigung berechnen

Aufgabe:
4.1) Für die folgenden Messungen genügt es, die Gesamtbeschleunigung beim Stoß zu kennen. Berechne die Gesamtbeschleunigung und ergänze damit das Programm.

→ *Missionshilfe 1*

2. Programm um Trigger erweitern

Die DART-Sonde startete am 24. November 2021 von der Erde und schlug am 26. September 2022 auf dem Asteroiden Dimorphos ein. Sie war also etwa 10 Monate unterwegs, bevor sie ihr Ziel erreichte und dort gezielt einschlug. Damit Wissenschaftler:innen bei der DART-Mission den Einschlag und seine Folgen untersuchen konnten, mussten die Sensoren der Sonde genau im richtigen Moment messen. Hier kommt das Prinzip des ***Triggers*** ins Spiel. Ein Trigger ist ein Mechanismus, der eine Messung automatisch startet, wenn ein bestimmter Grenzwert überschritten wird, zum Beispiel eine bestimmte Beschleunigung. Dadurch werden nur wichtige Ereignisse aufgezeichnet, was Speicherplatz spart, und die Analyse vereinfacht.

Aufgaben:
4.2) Passe das bestehende Programm so an, dass es nur dann misst, wenn ein Stoß erkannt wird (Trigger). Der Trigger soll ausgelöst werden, wenn die Gesamtbeschleunigung größer als 3g ist.
Nutze hierfür folgende Variablen:

```
const float TRIGGER_GRENZE = 3.0;
bool triggered;
```

→ *Missionshilfen 2 + 3*

4.3) Verändere die Variable TRIGGER_GRENZE und beobachte, was sich bei der Messung verändert.

3. Programm um Messdauer erweitern

Bei DART war nicht nur der Moment des Aufpralls entscheidend, sondern auch das Verhalten des Asteroiden nach dem Aufprall musste genau analysiert werden. Um solche Stoßverläufe zu untersuchen, mussten die Sensoren nicht nur den Triggerpunkt des Einschlags erkennen, sondern auch für eine bestimmte Zeit danach Daten aufzeichnen – nur so konnte man sehen, wie sich der Stoß über die Zeit entwickelt hat.

Aufgabe:

4.4) Erweitere dein Programm so, dass nach dem Einschlag für eine bestimmte Zeitspanne weiter gemessen wird (z. B. 500 ms). Danach soll die Messung automatisch stoppen und der Sensor auf den nächsten Stoß warten.
Nutze hierfür folgende Variable:

```
const unsigned long MESSDAUER = 500;
unsigned long startZeit;
```

Die Steuerung der Messung erfolgt mit:

```
millis() - startZeit <= MESSDAUER
```

→ *Missionshilfen 2 + 4*

4.5) Verändere die Variable MESSDAUER und beobachte, wie sich die Messdaten verändern.

5 Teststation bauen, Messung durchführen und auswerten

Mit deinem erweiterten Programm kannst du nun automatisch die Beschleunigung bei verschiedenen Stößen messen.
Dabei erfasst der Sensor nur dann Daten, wenn ein Stoß erkannt wird und zeichnet anschließend für eine definierte Dauer den zeitlichen Verlauf der Gesamtbeschleunigung auf.
So kannst du systematisch untersuchen, wie sich unterschiedliche Materialien und Fallhöhen auf die Stärke und Dauer des Stoßes auswirken.

1. Materialien:

- MPU6050-Beschleunigungssensor mit Arduino
- Holzplattform mit Filzfüßen
- Kugel (Holz o. ä.)
- Verschiedene Unterlagen: Holz, Moosgummi (1,5 cm), Schaumstoff (3 cm)
- Klebeband
- Maßband oder Lineal zur Bestimmung der Fallhöhe
- Computer mit Arduino IDE und Tabellenkalkulation (z. B. Excel)

2. Aufbau:

Befestige den (mit dem Arduino verbundenen) Sensor mithilfe von Klebeband unter der Holzplattform. Fixiere die Holzplattform z. B. mit Klebeband fest am Boden. Platziere zunächst das 1,5 cm dicke Moosgummi (später Schaumstoff, Holz) auf der Plattform, sodass der gesamte Sensor bedeckt ist. Markiere auf dem Moosgummi (später Schaumstoff, Holz) die Stelle, unter der sich der Sensor befindet (vgl. Abb. 6).
Verbinde den Arduino mit dem PC, öffne die IDE, lade das Programm auf den Arduino und öffne den seriellen Monitor.

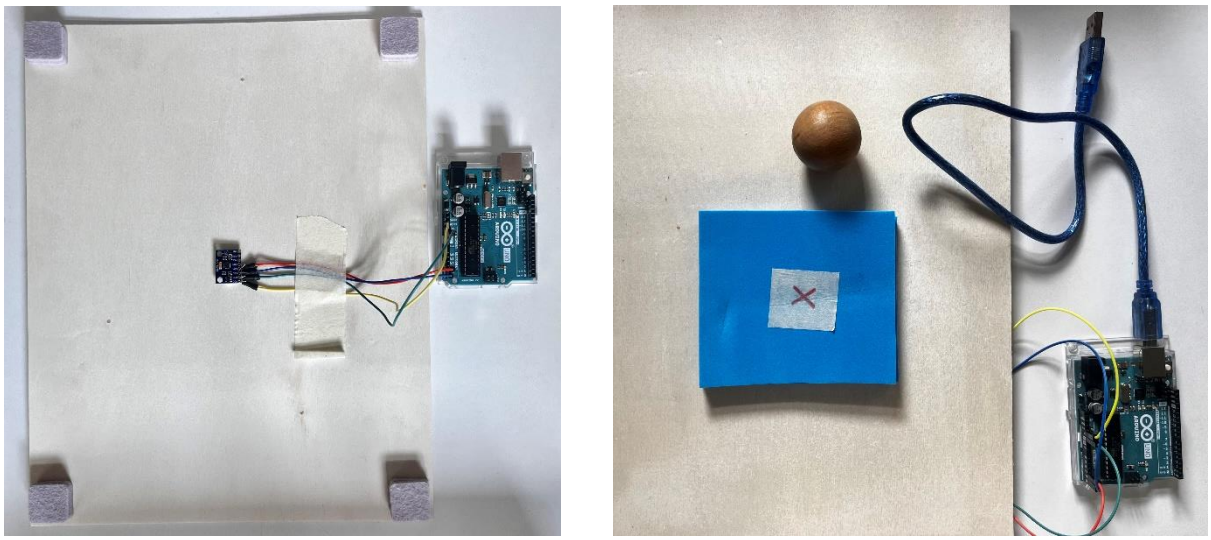


Abb. 6: Versuchsaufbau Messplattform mit MPU6050 und Arduino von unten (links) und oben (rechts).
Bild: Katharina Supp

3. Durchführung:
Lass die Kugel aus einer definierten Höhe (1 cm, 2 cm, 3 cm, 4 cm, 5 cm) senkrecht auf die markierte Stelle der Unterlage fallen. Achte dabei darauf, dass du den Sensor immer zentral triffst. Deine Messwerte werden automatisch auf dem seriellen Monitor angezeigt.
Wiederhole den Versuch anschließend für die anderen Untergrundmaterialien (Holz und Schaumstoff).

⊕ → *Falls du zu viel oder zu wenige Daten erhältst, kannst du bei Bedarf die Triggergrenze (TRIGGER_GRENZE) oder die Messdauer (MESSDAUER) im Code anpassen.*

4. Auswertung:
Datenübertragung
Kopiere die Werte aus dem seriellen Monitor in einen Texteditor (jede Fallhöhe für jedes Material einzeln). Ersetze die Dezimalpunkte durch Kommata (Strg+H → „.“ durch „,“ ersetzen).
Speichere die Datei als .txt ab.

Import in Tabellenkalkulation
Öffne anschließend ein Tabellenkalkulationsprogramm (z. B. Excel) und importiere deine gespeicherten Messdaten.

Diagramm erstellen
Erstelle für jede Fallhöhe bei jedem Untergrund ein Punktdiagramm:
- x-Achse: Zeit in Millisekunden
- y-Achse: Beschleunigung (in Vielfachen von g)
- Diagrammtitel und Achsenbeschriftung klar und einheitlich.

Daten analysieren
Trage alle deine Ergebnisse der Datenanalyse in Tabelle 3 ein.
A: Form des Peaks beurteilen:
Schau dir den Peak im Diagramm an und notiere in Spalte **„Form des Peaks“**, ob der Peak eher spitz oder breit ist und welche anderen Besonderheiten er aufweist.

B: Abbremszeit bestimmen:
Miss im Diagramm die Breite des Peaks (Zeit zwischen Beginn des Anstiegs und Ende des Abfalls der Beschleunigung) und trage diesen Wert in die Spalte **„Abbremszeit Δt [ms]“** ein.

C: Peak-Beschleunigung messen:
Lies für jede Fallhöhe und jedes Material den höchsten Peak (Maximalbeschleunigung) aus deinem Beschleunigungs-Zeit-Diagramm ab und trage ihn in die Spalte „**Maximalbeschleunigung a/g (Messwert)“** ein.

D: Theoretischen Wert berechnen:
Berechne die theoretische mittlere Beschleunigung in Vielfachen von g.
Trage den berechneten Wert in die Spalte **„Maximalbeschleunigung a/g (berechneter Wert)“** ein.

E: Prozentuale Abweichung berechnen:
Berechne die prozentuale Abweichung zwischen Theoriewert und experimentell bestimmtem Wert.
Trage die Werte in die Spalte **„Abweichung (%)“** ein.

Tab. 3: Übersicht zur Auswertung der Impact-Versuche.

Material	Fallhöhe [m]	Form des Peaks	Abbremszeit Δt [ms]	Maximalbeschleunigung a/g (Messwert)	Maximalbeschleunigung a/g (berechneter Wert)	Abweichung [%]
Holz	0,01					
	0,02					
	0,03					
	0,04					
	0,05					
Moosgummi	0,01					
	0,02					
	0,03					
	0,04					
	0,05					
Schaumstoff	0,01					
	0,02					
	0,03					
	0,04					
	0,05					

Aufgaben:

5.1) Vergleiche die Kurvenform (Peak-Höhe und -Breite) bei den drei Untergründen. Was kannst du daraus über die Härte und Dämpfungseigenschaften des Materials ableiten?

5.2) Begründe, warum bei Holz die Werte irgendwann nicht weiter steigen.

5.3) Erkläre, weshalb weiche Materialien (z. B. Schaumstoff) zu niedrigeren, aber breiteren Peaks führen.

5.4) Beurteile auf Grundlage deiner Untersuchungen, welcher Untergrund für eine Raumkapsel-Landung am besten geeignet wäre.

5.5) Stelle dar, warum die genaue Kenntnis der Aufprallbeschleunigung für Raumfahrtmissionen wie DART entscheidend ist.

5.6) Dein Impact-Versuch mit dem MPU6050-Sensor liefert viele interessante Daten zur Beschleunigung beim Aufprall. Doch wie bei jedem Experiment entstehen dabei Messfehler.

a) Nenne und erkläre drei *systematische* Fehlerquellen und drei *zufällige* Fehlerquellen, die bei deinem Versuch auftreten können.

> ***Systematische Fehler*** sind Abweichungen, die bei jeder Messung in gleicher Weise auftreten. Sie führen dazu, dass alle Messergebnisse in eine bestimmte Richtung verfälscht werden (z. B. immer zu hoch).
> Sie entstehen durch falsche Einstellungen, falsch programmierte Software, unpräzise Aufbauweise, …
>
> ***Zufällige Fehler*** sind nicht vorhersehbare Schwankungen, die bei jeder Messung unterschiedlich auftreten. Sie entstehen durch kleine, unkontrollierbare Einflüsse während der Durchführung.

b) Schlage für zwei genannte Fehlerquellen konkrete Maßnahmen zur Verbesserung der Messgenauigkeit vor.

Anhang

Missionshilfe 1

Nutze den Satz des Pythagoras!

Missionshilfe 2

SCHRITT 1: VARIABLEN HINZUFÜGEN

- eine Variable *triggered*, um zu merken, ob die Messung läuft.
- eine Variable *startZeit*, um den Startzeitpunkt der Messung in Millisekunden zu speichern.
- eine Variable *TRIGGER_GRENZE*

Überlege dir genau, welcher Variablentyp jeweils sinnvoll ist. Hilfe findest du in der Tabelle unten.

Tab. 3: Übersicht über Variablentypen

Variablentyp	Bedeutung/Verwendung im Code	Beispiel
int	ganze Zahl (mit Vorzeichen), typischerweise -32.765 bis 32.767	int zaehler = 42;
float	Gleitkommazahl (Nachkommastellen möglich)	float temperatur = 23.5
long	sehr große Ganzzahlen (mit Vorzeichen), -2.147.483.648 bis 2.147.483.647	long distanz = 100000;
bool	logischer Wert: nur true oder false	bool ledAn = true;
const	macht Werte fest, schützt vor versehentlicher Änderung	const float PI = 3.14159;
unsigned	Ganze Zahl ohne Vorzeichen, Wertebereich nur positiv	unsigned int anzahl = 500;

Missionshilfe 3

SCHRITT 2: TRIGGER EINBAUEN

Das Flussdiagramm in Abb. 7 hilft dir, die Abläufe besser zu verstehen.

Falls du noch nicht weiterkommst: Verbinde die Programmbausteine rechts mit dem richtigen Prozess im Flussdiagramm.

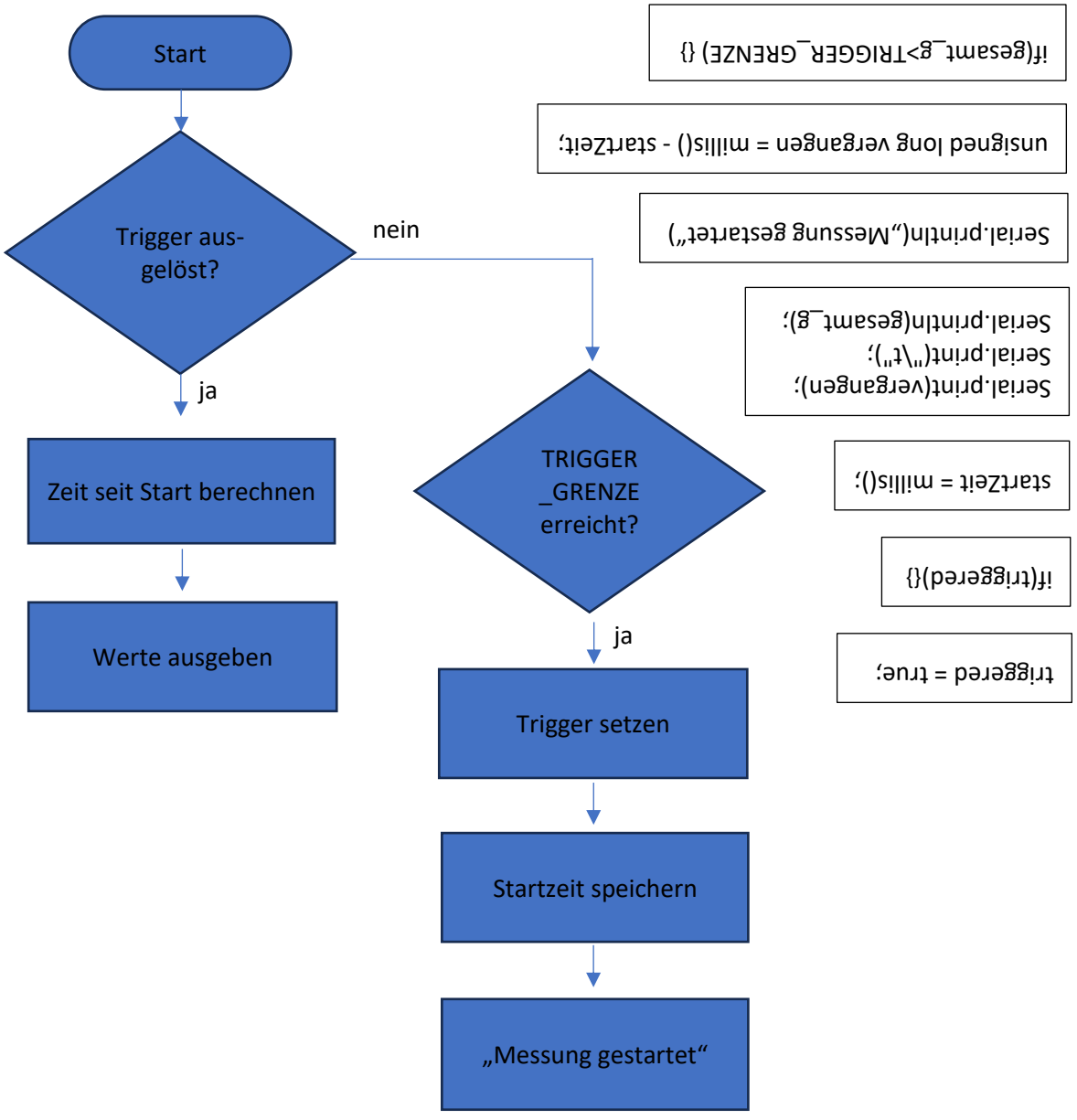


Abb. 7: Flussdiagramm zur Veranschaulichung des Triggers.

Missionshilfe 4

SCHRITT 3: VARIABLEN HINZUFÜGEN

Variable *MESSDAUER* einfügen

Überlege dir genau, welcher Variablentyp sinnvoll ist. Hilfe findest du bei Missionshilfe 2.

SCHRITT 4: MESSUNG MIT MESSDAUER STEUERN

Die Messung läuft und gibt Daten aus, falls die Messdauer noch nicht überschritten ist. Sobald die maximale Messdauer erreicht ist, wird die Messung beendet und der Trigger zurückgesetzt. Die Informationen „Messung beendet“ und „Warte auf nächsten Stoß…“ werden auf dem seriellen Monitor angezeigt.

Das Flussdiagramm in Abb. 8 hilft dir, die Abläufe besser zu verstehen.

Falls du noch nicht weiterkommst: Verbinde die Programmbausteine rechts mit dem richtigen Prozess im Flussdiagramm.

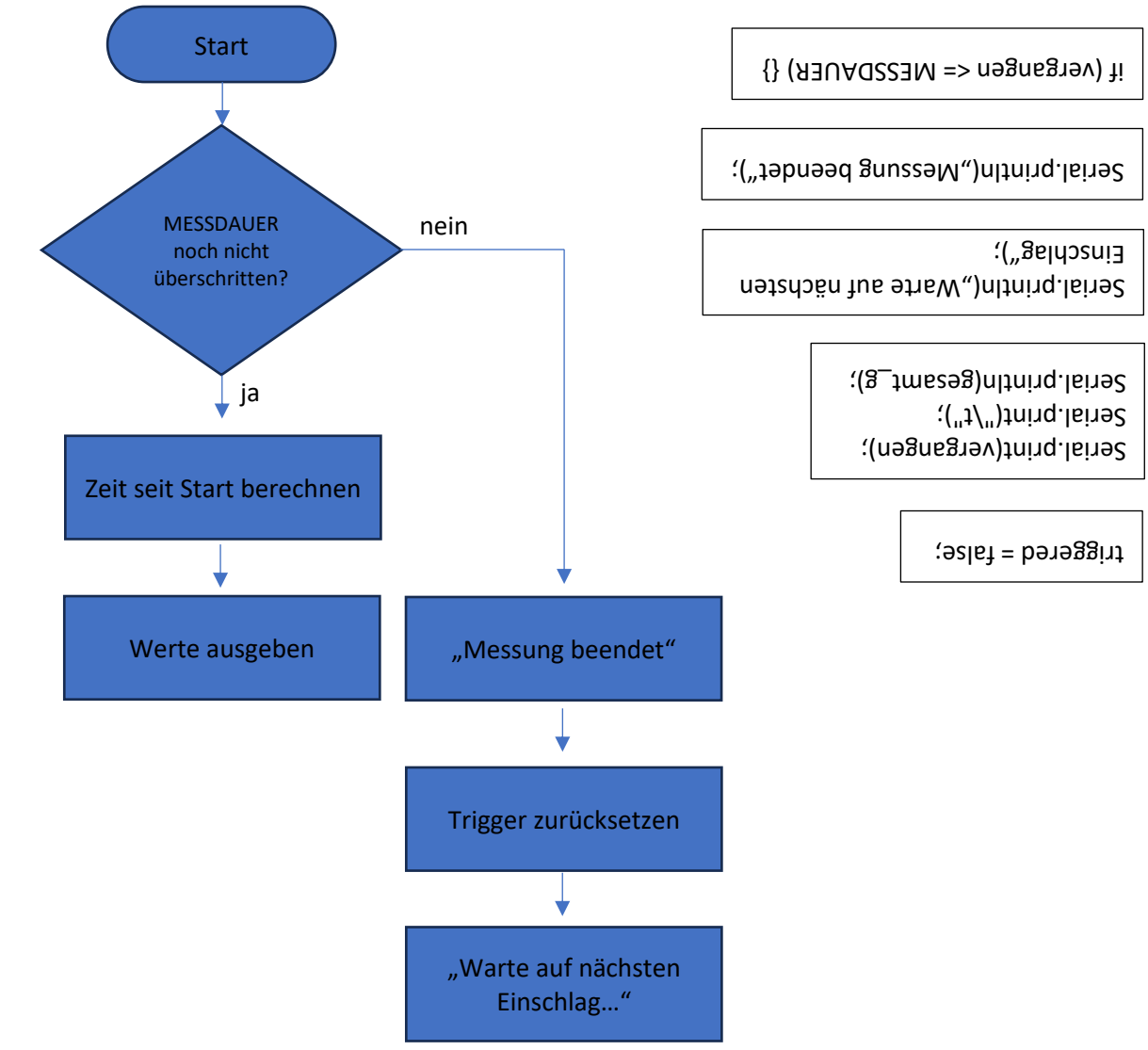


Abb. 8: Flussdiagramm zur Veranschaulichung der Steuerung des Programms mit der Messdauer.

Basiscode

```
// -------- Bibliotheken einbinden --------
// Bibliothek für MPU6050-Sensor von Adafruit
#include <Adafruit_MPU6050.h>
// Einheitliche Sensor-Schnittstelle für Adafruit-Sensoren
#include <Adafruit_Sensor.h>
// Bibliothek für I²C-Kommunikation
#include <Wire.h>

// Erzeugt ein Sensor-Objekt "mpu"
Adafruit_MPU6050 mpu;

// -------- Einstellungen (an Versuchsaufbau anpassen) --------
// Pause zwischen Messpunkten in Millisekunden
const int WARTE_MS = 1;

void setup()
{
    // Serielle Verbindung mit 115200 Baud starten
    Serial.begin(115200);
    // MPU6050-Sensor initialisieren
    initialisiereSensor();

}

void loop()
{
    float x_g, y_g, z_g;
    // Beschleunigungswerte in g auslesen
    leseBeschleunigung(x_g, y_g, z_g);

    // Werte seriell ausgeben, getrennt durch Tabulator (\t) für bessere Lesbarkeit
    druckeBeschleunigung(x_g, y_g, z_g);

    // kleine Pause für stabile Abtastrate
    // (sehr hohe Abtastrate für höhere Genauigkeit)
    delay(WARTE_MS);
}



// -------- Unterfunktionen --------

// Initialisiert den MPU6050-Sensor
void initialisiereSensor()
{
    // Falls Sensor erkannt wird,
    if (mpu.begin())
    {
        // Messbereich und Filterbandbreite des Sensors festlegen:
        // Messbereich ±16 g (höchster Bereich → nötig für Stöße)
```

```
        mpu.setAccelerometerRange(MPU6050_RANGE_16_G);
        // Filterbandbreite 260 Hz (schnelle Reaktion → weniger Glättung)
        mpu.setFilterBandwidth(MPU6050_BAND_260_HZ);
        // Ausgabe in seriellem Monitor strukturieren:
        Serial.println("MPU6050 gestartet");
        // Spaltenüberschriften
        Serial.println("X (g)\tY (g)\tZ (g)");
    }
    // sonst (Sensor nicht erkannt)
    else
    {
        // "MPU6050 nicht gefunden!" ausgeben
        Serial.println("MPU6050 nicht gefunden!");
        // Endlosschleife – Programm stoppt hier
           while (true) {delay(10);}

    }
}

// Liest die Beschleunigungswerte in g
void leseBeschleunigung(float &x_g, float &y_g, float &z_g)
{
    // Beschleunigungsdaten abrufen
    sensors_event_t a;
    mpu.getAccelerometerSensor()->getEvent(&a);
    // Beschleunigung der Einzelachsen von m/s² in Vielfache von g umrechnen
    // (1 g = 9,81 m/s²)
    x_g = a.acceleration.x / 9.81;
    y_g = a.acceleration.y / 9.81;
    z_g = a.acceleration.z / 9.81;
}


void druckeBeschleunigung(float &x_g, float &y_g, float &z_g)
{
    // Werte seriell ausgeben, getrennt durch Tabulator (\t) für bessere Lesbarkeit
    Serial.print(x_g);
    Serial.print("\t");
    Serial.print(y_g);
    Serial.print("\t");
    Serial.println(z_g);
}
```

Code (fertig)

```
#include <Adafruit_MPU6050.h>
#include <Adafruit_Sensor.h>
#include <Wire.h>

Adafruit_MPU6050 mpu;

// -------- Einstellungen (an Versuchsaufbau anpassen) --------
// ab dieser Gesamt-Beschleunigung (in g) startet die Messung
const float TRIGGER_GRENZE = 3.0;
// Messdauer nach dem Trigger in Millisekunden
const unsigned long MESSDAUER = 500;
// Pause zwischen Messpunkten in Millisekunden
const int WARTE_MS = 1;

// true: Messung läuft; false: wartet
bool triggered = false;
// Startzeit der Messung
unsigned long startZeit;

void setup()
{
    // Serielle Verbindung mit 115200 Baud starten
    Serial.begin(115200);
    // MPU6050-Sensor initialisieren
    initialisiereSensor();
}

void loop()
{
    float x_g, y_g, z_g;
    // Beschleunigungswerte in g auslesen
    leseBeschleunigung(x_g, y_g, z_g);

    // Gesamtbeschleunigung berechnen (Pythagoras für 3D):
    float gesamt_g = sqrt(x_g * x_g + y_g * y_g + z_g * z_g);

    // Falls der Trigger ausgelöst wurde,
    if (triggered)
    {
        // dann Messung durchführen.
        fuehreMessungDurch(gesamt_g);
    }
    // Sonst
    else
    {
        // prüfen, ob der Triggerwert überschritten wurde, und Messung starten.
        pruefeAufTrigger(gesamt_g);
    }
    // kleine Pause für stabile Abtastrat
    delay(WARTE_MS);
}
```

```
// -------- Unterfunktionen --------

// Initialisiert den MPU6050-Sensor
void initialisiereSensor()
{
    // Falls Sensor erkannt wird,
    if (mpu.begin())
    {
        // Messbereich und Filterbandbreite des Sensors festlegen:
        // Messbereich ±16 g (höchster Bereich → nötig für Stöße)
        mpu.setAccelerometerRange(MPU6050_RANGE_16_G);
        // Filterbandbreite 260 Hz (schnelle Reaktion → weniger Glättung)
        mpu.setFilterBandwidth(MPU6050_BAND_260_HZ);
        // Ausgabe in seriellem Monitor strukturieren:
        Serial.println("MPU6050 gestartet");
        // Spaltenüberschriften
        Serial.println("Zeit (ms)\tGesamtbeschleunigung (g)");
    }
    // sonst (Sensor nicht erkannt)
    else
    {
        // "MPU6050 nicht gefunden!" ausgeben

        Serial.println("MPU6050 nicht gefunden!");
        // Endlosschleife – Programm stoppt hier
           while (true) {delay(10);}

    }
}

// Liest die Beschleunigungswerte in g
void leseBeschleunigung(float &x_g, float &y_g, float &z_g)
{
    // Beschleunigungsdaten abrufen
    sensors_event_t a;
    mpu.getAccelerometerSensor()->getEvent(&a);
    // Beschleunigung der Einzelachsen von m/s² in Vielfache von g umrechnen
    // (1 g = 9,81 m/s²)
    x_g = a.acceleration.x / 9.81;
    y_g = a.acceleration.y / 9.81;
    z_g = a.acceleration.z / 9.81;
}

// Führt eine laufende Messung durch
void fuehreMessungDurch(float gesamt_g)
{
    //Zeit seit Start berechnen
    unsigned long vergangen = millis() - startZeit;

    // falls die Messdauer nicht überschritten ist,
    if (vergangen <= MESSDAUER)
```

```
    {
        // dann Zeit und Gesamtbeschleunigung ausgeben (tabgetrennt)
        Serial.print(vergangen);
        Serial.print("\t");
        Serial.println(gesamt_g);
    }
    // sonst (d.h. die Messdauer überschritten ist)
    else
    {
        // Ausgabe "Messung beendet" auf seriellem Monitor
        Serial.println("Messung beendet");
        // Messung beenden, Trigger zurücksetzen
        triggered = false;
        // Ausgabe "Warten auf nächsten Stoß..." auf seriellem Monitor
        Serial.println("Warte auf nächsten Stoß...");
    }
}


// Prüft, ob Triggerwert überschritten wurde und startet Messung
void pruefeAufTrigger(float gesamt_g)
{
// falls gesamt_g > TRIGGER_GRENZE,
        if (gesamt_g > TRIGGER_GRENZE)
        {
            // dann läuft die Messung,
            triggered = true;
            // Startzeit speichern,
            startZeit = millis();
            // Ausgabe "Messung gestartet" auf seriellem Monitor
            Serial.println("Messung gestartet");
        }
}
```

Lösungsvorschläge

1.1)

- Ziel: Nachweis, dass ein gezielter kinetischer Einschlag die Bahn eines Asteroiden messbar verändern kann. Bei DART wurde der Asteroidenmond Dimorphos in seiner Umlaufzeit um ca. 33 min verkürzt – erster menschengemachter, beobachtbarer Bahneingriff an einem Himmelskörper.
- Einordnung: Teil der Planetary Defense-Strategien. DART testet die Methode der kinetischen Ablenkung und liefert reale Daten zur Wirksamkeit.

1.2)

- Binärsystem Didymos-Dimorphos: Änderung der Mond-Umlaufzeit lässt sich von der Erde aus präzise messen.
- Sicherheitsaspekt: Kein Risiko für Erdkurs.
- Größe/Bahndynamik: Erreichbar, Änderung messbar.

1.3)

- Wirkung hängt von Porosität, Festigkeit, Regolith ab.
- Materialauswurf erzeugte zusätzlichen Rückstoß → größere Bahnänderung als erwartet.

1.4)

- Tunguska (1908): Luftdetonation, großflächige Waldzerstörung, keine größeren Krater.
- Chelyabinsk (2013): Luftexplosion, Druckwelle beschädigte Gebäude, mehr als 1000 Verletzte.

2.1)

Aufprallgeschwindigkeiten ($g=9{,}81\ m/s^2$):

- h=0,01 m → v=0,443 m/s
- h=0,03 m → v=0,767 m/s
- h=0,50 m → v=3,132 m/s

2.2)

geg.: m=570 kg, v=6600 m/s

$E_{kin} = 1{,}24 \times 10^{10}\ J \approx 3\ t\ TNT$

3.1)

a) Einschlag nach unten: Platte nach unten ausgelenkt, Kapazität auf einer Seite sinkt, auf anderer steigt.

b) Rückfederung nach oben: umgekehrte Auslenkung, Kapazitätsänderung invers.

3.3)

b) 1g im Ruhezustand: Schwerebeschleunigung.

c) kopfüber: Vorzeichenwechsel.

d) kippen: 1g wandert auf andere Achse.

e) 90°: 1g "springt" auf die Achse, die jetzt vertikal steht.

f) ruckartig: Peaks >1g.

5.1)

Peakformen:

- Holz: hoch, schmal, kurze Δt → große a
- Moosgummi: mittel, breiter

- Schaumstoff: niedrig, breit, lange Δt

5.2)
Holzwerte steigen nicht weiter: Sensorlimit ±16 g

5.3)
Weiche Unterlagen: längere Δt → kleinere a, breiterer Peak.

5.4)
Von den drei getesteten Untergründen ist Schaumstoff am besten geeignet.
Begründung (physikalisch):

- Beim Aufsetzen wird die Kapsel aus der vertikalen Sinkgeschwindigkeit v in sehr kurzer Zeit abgebremst. Die mittlere Abbremsbeschleunigung ist $a=\Delta v / \Delta t$
- Weiche, dämpfende Materialien (Schaumstoff) vergrößern Δt → senken die Spitzen-g-Belastung. In den a(t)-Kurven zeigt sich das durch breitere, niedrigere Peaks.
- Harte Unterlagen (Holz) führen zu kurzer Δt → hohe Peak-g, mehr Strukturbelastung und oft Rückprall (Instabilität). Zudem droht Sensorsättigung schon bei kleinen Höhen – ein Indiz für die hohen Lasten.
- Moosgummi liegt dazwischen: brauchbare Dämpfung, aber höhere Peaks als Schaumstoff.

Technische Einordnung:

- Für echte Landungen kombiniert man oft mehrlagige Dämpfer (z. B. Schaum/Crush-Core/Öldämpfer) oder verformbare Strukturen, um Δt sicher zu verlängern und Lastspitzen zu begrenzen.
- Zu weicher Untergrund darf aber nicht „durchschlagen“: Schaumstoff muss genug Dicke und tragfähige Stütze darunter haben, damit die Verzögerung gleichmäßig und kontrolliert bleibt.

Fazit:
Schaumstoff (ggf. als mehrlagiges Dämpfungssystem) minimiert die Spitzenbelastungen und erhöht die Landetoleranz – daher die beste Wahl für eine (modellhafte) Raumkapsel-Landung.

5.5)
Die exakte Kenntnis der Aufprallbeschleunigung ist für Missionen wie DART aus mehreren Gründen entscheidend:

- Berechnung der Impulsübertragung
  - Der Impuls $p=m \cdot v$ wird beim Einschlag teilweise auf den Asteroiden übertragen.
  - Aus dem zeitlichen Verlauf der Beschleunigung kann man auf die tatsächliche Kraftwirkung und damit auf den übertragenen Impuls schließen.
  - Nur so lässt sich quantitativ bestimmen, wie stark die Bahn des Asteroiden verändert wurde.
- Analyse des Materialverhaltens
  - Unterschiedliche Oberflächen (locker, fest, porös) reagieren sehr verschieden auf den Einschlag.
  - Die Beschleunigungskurve gibt Hinweise auf Härte, Dichte und Porosität des Zielobjekts.
  - Dies ist wichtig, um zu verstehen, ob der Asteroid Material ausstößt und damit zusätzlichen Rückstoß erzeugt.

Beispielanalyse (AB 5)

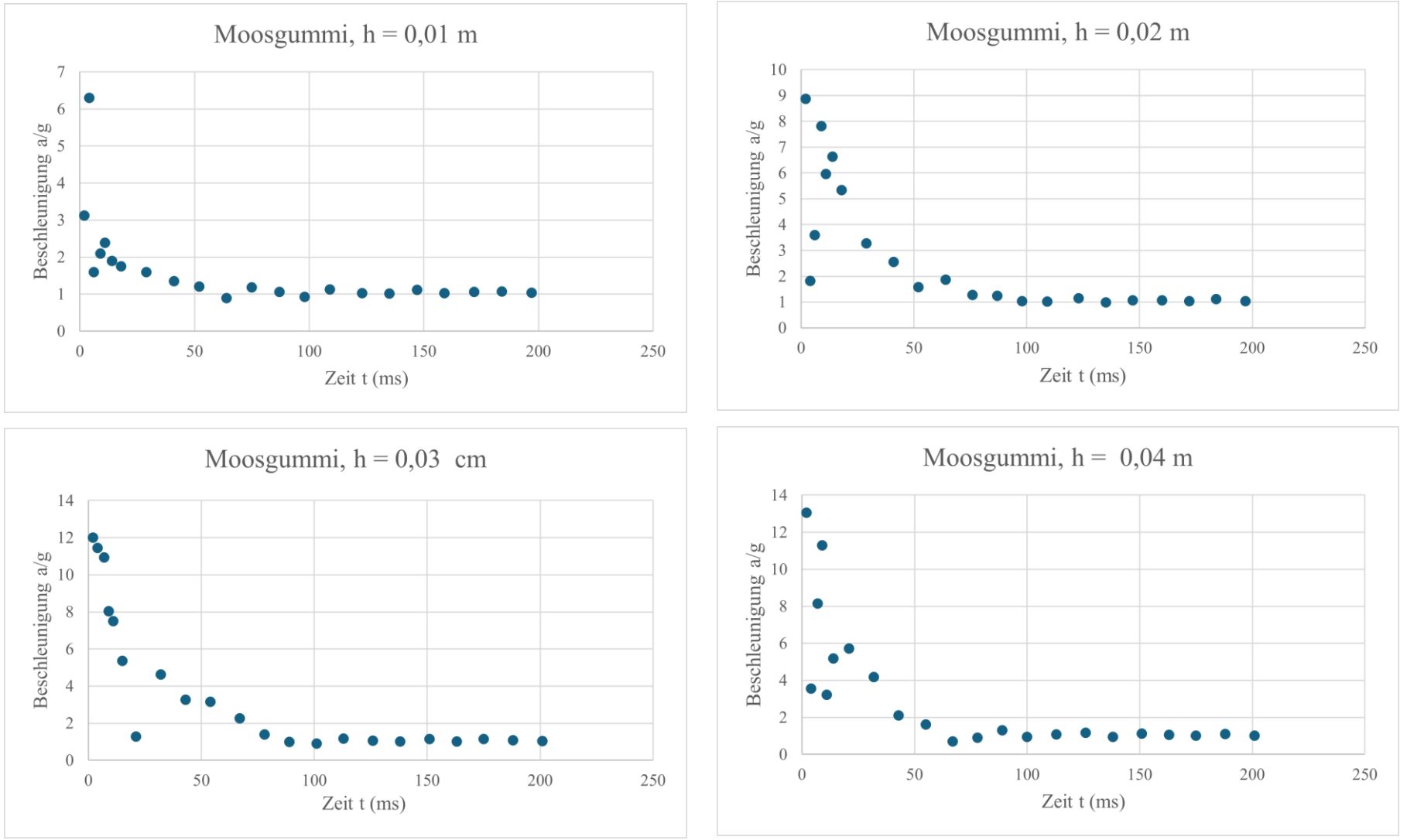


Abb. 9: Diagramme zur Beispielauswertung der Impact-Versuche mit Moosgummi. Bild: Katharina Supp

Material	Fallhöhe [m]	Form des Peaks	Abbremszeit Δt [ms]	Maximalbeschleunigung a/g (Messwert)	Maximalbeschleunigung a/g (berechneter Wert*)	Abweichung (%)
Moosgummi	0,01	spitzer, unten breit	7,0	6,3	6,4	1,6
	0,02	mittelbreit	8,0	9,0	7,9	12,2
	0,03	mittelbreit, unten breiter	7,0	12,0	11,2	7,1
	0,04	mittelbreit	7,0	13,3	12,9	3,0

Tab. 4: Beispielauswertung der Impact-Versuche mit Moosgummi

*Die Aufprallgeschwindigkeit berechnet sich für die verschiedenen Höhen h mit: $v = \sqrt{2 \cdot g \cdot h}$

Die Abbremszeit Δt liest man im Diagramm ab.

Für theoretische Beschleunigung ergibt sich dann aus: $a = \frac{\Delta v}{\Delta t}$

Quellen

Grieger, Christian: MPU6050: 3-Achsen Beschleunigungs- und Lagesensor. In: elektro.turanis.de, https://elektro.turanis.de/html/prj075/index.html (zuletzt aufgerufen am 06.08.2025).

NASA: Planetary Defense at NASA: In: nasa.gov, https://science.nasa.gov/planetary-defense/ (zuletzt aufgerufen am 07.08.2025).

Schnabel, Patrick: MEMS - Micro-Electro-Mechanical Systems. In: Elektronik-Kompendium.de, https://www.elektronik-kompendium.de/sites/bau/1503041.htm (zuletzt aufgerufen am 06.08.2025).

The Johns Hopkins University Applied Physics Laboratory LLC: DART. In: https://dart.jhuapl.edu/ (zuletzt aufgerufen am 06.08.2025).

Software/Bibliotheken

Adafruit MPU6050 Bibliothek: https://github.com/adafruit/Adafruit_MPU6050 (zuletzt aufgerufen am 06.08.2025)

Arduino IDE: https://www.arduino.cc/en/software/ (zuletzt aufgerufen am 06.08.2025)