

Der ESP32 im Kontext MINT-Bereich

Kristina Holmann, Angela Fösel

Friedrich-Alexander-Universität Erlangen-Nürnberg, Department Physik, Didaktik der Physik
Stadttstraße 7, 91058 Erlangen
angela.foesel@fau.de

Kurzfassung

Der ESP32 ist ein beliebter Mikrocontroller in der Welt der Elektronik und des Internet of Things (IoT). Für den Einsatz im Physik- und Technikunterricht ist er hervorragend geeignet: Er verfügt über eine Vielzahl an intern verbauten Sensoren, und bei fast allen Modellen bietet der ESP32 Bluetooth /Bluetooth Low Energy (BLE) Konnektivität. Im Beitrag wird der ESP32 mit Blick auf den Einsatz im MINT-Unterricht charakterisiert. Der Fokus liegt hierbei auf der Nutzung der Sensoren sowie auf dem Einsatz eines modernen, modular aufgebauten, computergestützten Messwerterfassungssystems mit Option auf BLE-Konnektivität. Für die Praxis relevante Unterschiede gegenüber dem Mikrocontroller-Board „Arduino“ sowie dem Mini-PC „Raspberry Pi“ werden kurz charakterisiert und diskutiert. Exemplarisch werden zwei kreative schülerrelevante Projekte vorgestellt, die auf einem ESP32 basieren und eindrücklich zeigen, dass der ESP32 den MINT-Unterricht um digitale Medien bereichert und zugleich spielerisch zu einem kompetenzorientierten Unterricht einlädt. In einem ersten Projekt werden mit dem ESP32-CAM-Modul Objekte in einem Live-Stream erkannt und identifiziert, indem die Bilddaten analysiert und interpretiert werden. Im zweiten Projekt wird ein Anemometer in Kombination mit dem im ESP32 integrierten Hall-Sensor eingesetzt, um die Windgeschwindigkeit zu messen.

1. Motivation für den Einsatz des ESP32 im MINT-Bereich

Der Mikrocontroller ESP32 ist ein vielseitiges digitales Werkzeug, das eine fundierte Basis für einen praxisnahen, interaktiven und sinnstiftenden MINT-Unterricht (Mathematik, Informatik, Naturwissenschaften und Technik) bietet [1]. Laut der Sinus-Studie 2024 [2] fördert ein solcher Unterricht das nachhaltige Interesse an Technik und Naturwissenschaften bei Schülerinnen und Schülern – eine der zentralen Ziele des MINT-Unterrichts.

Der Mikrocontroller beweist sich durch seine interdisziplinären Einsatzmöglichkeiten und unterstützt damit gleichzeitig den Erwerb fachlicher Kompetenzen sowie die Entwicklung eines vernetzten Denkens bei den Lernenden.

Beim Arbeiten mit dem Mikrocontroller im naturwissenschaftlichen Unterricht ermöglicht der ESP32 die Vermittlung grundlegender Programmier- und Sensorikkenntnisse sowie den Erwerb von Kompetenzen im Bereich der Erkenntnisgewinnung. Beispielsweise wird im Physikunterricht die Erkenntniscompetenz durch das eigenständige Planen, Durchführen und Auswerten von Experimenten gestärkt, wodurch ein forschender Lernansatz bei Schülerinnen und Schülern gefördert wird.

Ein weiterer Vorteil des ESP32 ist seine Funktion als kosteneffiziente Alternative zu traditionellen Messwerterfassungssystemen wie CASSY. Aufgrund seiner geringen Anschaffungskosten ist eine

flächendeckende Ausstattung ganzer Klassen möglich, was den Einsatz im Unterricht auf breiter Basis unterstützt. Dies erleichtert den Zugang zu digitalen Mess- und Steuerungstechnologien und schafft gleiche Lernbedingungen für alle Schülerinnen und Schüler.

2. Charakterisierung des ESP32 - ein Überblick

Der Mikrocontroller ESP32 ist in verschiedenen Versionen erhältlich. Die hier verwendete Variante, der ESP32-Wroom-32D, sticht hervor durch bereits integrierte Sensoren. Dazu zählen ein Hall-Sensor sowie kapazitive Touch-Sensoren [3]. Zusätzlich zu den integrierten Sensoren können externe Sensoren an den ESP32 angeschlossen werden wie zum Beispiel ein Temperatursensor. Diese Sensoren ermöglichen eine Erfassung physikalischer Größen oder Zustände der Umgebung und erweitern die Einsatzmöglichkeiten des Mikrocontrollers erheblich.

Diese Sensoren müssen über eine integrierte Entwicklungsumgebung (IDE) angesteuert werden, wobei die spezifischen Messparameter (wie Messbereich oder Abtastrate) direkt im Quellcode definiert werden. Dieser Code wird in Programmiersprachen wie C++ oder MicroPython verfasst und auf den Mikrocontroller hochgeladen. Details zu Hard- und Software des ESP32 sowie ein Einstieg in die Programmierung mit MicroPython und C++ bietet Brands Werk Mikrocontroller ESP32 – Das umfassende Handbuch [4].



Abb. 1: ESP32-Wroom-32D mit integriertem Hall-Sensor (schwarzes Kreuz) unter der Metallbeschichtung und den integrierten Touch-Sensoren, die sich unter folgenden Pinbelegungen befinden: 0, 2, 4, 13, 12, 14, 27, 32, 33. (Pins als kleine, graue Metallkontakte sichtbar, die auf der Seite oder Unterseite des Mikrocontrollers angeordnet sind). Keine externen Sensoren angeschlossen. Eigene Darstellung.

Das Auslesen der Sensoren bezieht sich auf den Vorgang, bei dem die von den Sensoren erfassten Daten in eine lesbare und weiter verarbeitbare Form überführt werden. Die erfassten Messwerte können in Echtzeit angezeigt, aufgezeichnet oder für weitere Analysen verarbeitet werden.

Neben den Sensormöglichkeiten bietet der ESP32 weitere zentrale Funktionen wie integrierte WLAN- und Bluetooth-Schnittstellen. Insbesondere die Bluetooth-Funktion ermöglicht eine drahtlose Kommunikation mit externen Geräten, was den Anwendungsbereich des Mikrocontrollers erheblich erweitert. Für einen einsteigerfreundlichen und praxisnahen Zugang zum ESP32 ist das Werk von Bartmann [5] empfehlenswert.

3. ESP32 in Abgrenzung zu Arduino & Raspberry Pi

Die größte Stärke des ESP32 im Vergleich zu Arduino oder Raspberry Pi liegt in seiner flexiblen Programmierbarkeit (vgl. Tabelle 1).

Tab. 1: Drei Mikrocontroller im Vergleich hinsichtlich ihrer Programmiermöglichkeiten zur Ansteuerung der an sie angeschlossenen bzw. integrierten Sensoren.

Programmierung		
ESP32	Arduino	Raspberry PI
C++ MicroPython	C++	Python Linux OS

Nutzerinnen und Nutzer, die bereits mit Arduino oder Raspberry Pi vertraut sind, finden schnell Zugang zum ESP32, da die Sensoren sowohl mit C++ als auch mit MicroPython angesteuert und ausgelesen werden können (vgl. Tabelle 1). MicroPython ist dabei eine speziell für Mikrocontroller optimierte Version der Programmiersprache Python und deshalb von der Syntax sehr ähnlich [4]. Für Einsteiger und Einsteigerinnen bietet die Arduino IDE eine benutzerfreundliche grafische Oberfläche, die durch zahlreiche

Codebeispiele einen besonders leichten Einstieg ermöglicht. Im Gegensatz dazu setzt die Nutzung eines Raspberry Pi häufig komplexere Setups voraus, beispielsweise die Installation eines Betriebssystems oder spezifischer Treiber. Der ESP32 bietet darüber hinaus eine steile Lernkurve: Lernende können sich von blockbasierter Programmierung über C++ bis hin zur skriptbasierten Entwicklung in MicroPython weiterentwickeln und dabei ihre Programmierfähigkeiten schrittweise ausbauen. Dank dieser Vielseitigkeit bietet der ESP32 deutlich mehr Gestaltungsspielraum für differenzierte Aufgabenstellungen im MINT-Unterricht – unabhängig vom Erfahrungsniveau der Lernenden.

4. Interner Hall-Sensor

4.1. Funktionsüberprüfung des internen Hall-Sensors mit einem experimentellen Setup.

Der interne Hall-Sensor des ESP32 ist für die Messung der magnetischen Flussdichte ausgelegt [6] und befindet sich unter der Metallbeschichtung des Mikrocontrollers (vgl. Abbildung 1). Es wurde ein Experiment konzipiert, bei dem ein Neodym-Magnet ($B = 437 \text{ mT}$) direkt auf den Hall-Sensor unter der Metallbeschichtung des ESP32 positioniert wurde. Der Magnet wurde dabei einmal mit dem Nordpol und einmal mit dem Südpol auf den Sensor gelegt – jeweils unter denselben Bedingungen und an derselben Position. Ziel des Experiments war es, die Funktionalität des integrierten Hall-Sensors zu überprüfen und festzustellen, ob er wie erwartet in der Lage ist, die magnetische Flussdichte des Neodym-Magneten in mT zu erfassen.

4.2. Auswertung der Funktionsüberprüfung

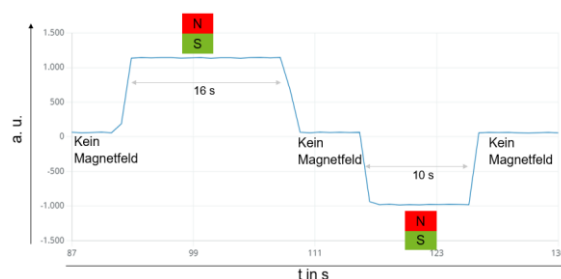


Abb. 2: „Magnetische Flussdichte“ in Abhängigkeit von der Zeit. Eigene Darstellung.

Die aufgezeichneten Messwerte sind in einem Graphen (vgl. Abbildung 2) dargestellt. Während die x-Achse die Zeit in Sekunden (s) angibt, wurde die y-Achse mit willkürlichen Einheiten (arbitrary units, a.u.) versehen. Die Wahl dieser Einheiten wird dadurch begründet, dass keine eindeutige Kalibrierung des Hall-Sensors zur Messung der magnetischen Flussdichte in Millitesla (mT) vorlag.

Wird der Neodym-Magnet mit dem Südpol nach unten auf den Hall-Sensor platziert, registriert der Sensor Messwerte im Bereich von +1000 bis +1100 a.u. Wird der Magnet umgekehrt (Nordpol nach unten) positioniert, liegen die Messwerte bei etwa -1000 a.u.

Ohne Magnet erkennt der Sensor kein magnetisches Feld, wobei ein Offset von 100 a.u. sich einstellt.

Zusammenfassend zeigt die Funktionsüberprüfung, dass der Hall-Sensor des ESP32 in der Lage ist, Nord- und Südpol eines Magneten zu unterscheiden. Er registriert Magnetfelder zuverlässig und kann somit als logischer Schaltmechanismus verwendet werden.

4.3. Didaktische Erkenntnis für den MINT-Unterricht

Der Hall-Sensor des ESP32 ist für eine präzise Messung der magnetischen Flussdichte ungeeignet und kann hinsichtlich Genauigkeit und Messbereich nicht mit professionellen Sensoren wie denen des CASSY-Systems konkurrieren. Statt exakter Messwerte liefert der Sensor lediglich eine qualitative Aussage: Er zeigt an, ob ein Magnetfeld erkannt wird (Ja/Nein).

Darüber hinaus muss das Magnetfeld eine ausreichende Stärke aufweisen, um vom Sensor überhaupt erfasst zu werden [4]. Diese Eigenschaften begrenzen den Einsatz des Hall-Sensors auf qualitative Experimente, eignen sich jedoch gut zur Vermittlung grundlegender Prinzipien der Magnetfeldmessung und als Schaltmechanismus.

5. Bluetooth Funktion des ESP32 in Verbindung mit der Phyphox-App

Die Bluetooth-Funktion des ESP32 bietet im schulischen Kontext erhebliche Vorteile, insbesondere an Schulen mit begrenzter Ausstattung an Laptops oder PCs. Sie ermöglicht eine drahtlose Verbindung zu mobilen Endgeräten wie Smartphones oder Tablets und erweitert damit die Möglichkeiten für interaktive und praxisnahe Experimente.

Ein besonders praxisnaher Anwendungsfall ist die Nutzung der Phyphox-Bibliothek [7]. Über die Bluetooth-Schnittstelle des ESP32 können Messdaten externer Sensoren in Echtzeit auf ein Smartphone übertragen und dort direkt in der Phyphox-App visualisiert werden. Zur Veranschaulichung wurde ein Abstandssensor mit dem ESP32 verbunden, dessen Messwerte über die Phyphox-App grafisch dargestellt wurden.



Abb. 3: Abstand in Abhängigkeit der Zeit; Weg-Zeit-Diagramm. Eigene Darstellung. Eigene Darstellung.

Der resultierende Graph zeigt den Abstand in Zentimetern (cm) in Abhängigkeit von der Zeit (s) und ermöglicht eine unmittelbare Analyse des Messverlaufs.

Darüber hinaus bietet die Phyphox-App die Möglichkeit, die aufgezeichneten Messdaten zu exportieren und in Programmen wie Excel weiterzuverarbeiten. Dadurch wird der ESP32 zu einem vielseitigen Werkzeug für den Unterricht, das nicht nur die Datenerfassung, sondern auch die Datenanalyse und -interpretation fördert.

6. Projekte

Im Folgenden werden exemplarisch zwei Projekte vorgestellt, die den Einsatz des ESP32 im MINT-Unterricht praxisnah veranschaulichen. Das erste Projekt konzentriert sich auf die vorgestellten Funktionen des Mikrocontrollers und hebt dessen zentrale Eigenschaften hervor. Das zweite Projekt demonstriert ergänzend die vielfältigen Anwendungsmöglichkeiten des ESP32 und zeigt, wie dieser flexibel in unterschiedlichen Kontexten [8] eingesetzt werden kann.

6.1. Messung der Windgeschwindigkeit mit einem Anemometer und dem ESP32

Ziel dieses Projekts war es, die Windgeschwindigkeit [9] indirekt mithilfe des ESP32 zu bestimmen. Anstelle einer direkten Messung diente der interne Hall-Sensor des ESP32 als Schaltmechanismus, da dieser – wie im Kapitel 4 diskutiert – zuverlässig starke Magnetfelder detektiert. Für das Projekt kam ein selbst 3D-gedrucktes Anemometer (vgl. Abbildung 4) zum Einsatz. Der ausführliche physikalische und technische Hintergrund des Anemometers ist vor allem in Quelle [10] sowie ergänzend in den Quellen [11] und [12] beschrieben.



Abb.4: Zusammengesetztes Anemometer - 3D gedruckt. Eigene Darstellung.

An der rotierenden Dachkonstruktion dieses Anemometers war ein Neodym-Magnet befestigt (vgl. Abbildung 5).

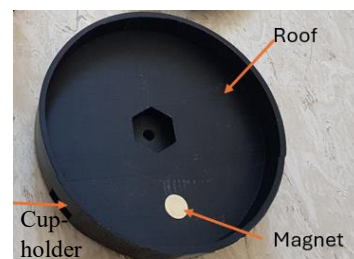


Abb. 5: Befestigter Neodym-Magnet im rotierenden Dach/Roof. Eigene Darstellung.

Bei jeder Umdrehung der Dachkonstruktion passierte der Magnet einen festgelegten Punkt, an dem der Hall-Sensor des ESP32 im kleinsten möglichen Abstand zu dem Magneten unter dem Dach positioniert war. Jedes Mal, wenn der Magnet diesen Punkt passierte, erfasste der Hall-Sensor das Magnetfeld und übermittelte ein Signal an den ESP32.

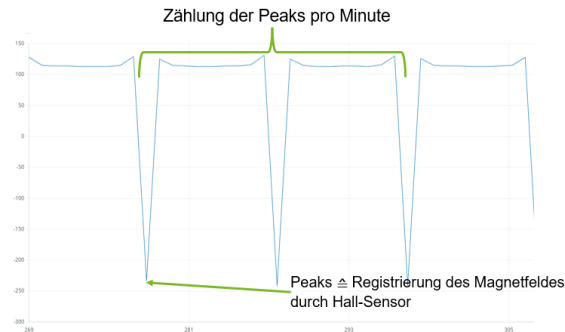


Abb. 6: Auswertung der ausgelesenen Daten durch den internen Hall-Sensor des ESP32. Eigene Darstellung.

Das Signal äußerte sich in einem negativen Peak wie in der Abbildung 6 zu sehen ist. Ein Peak repräsentiert eine Umdrehung der Dachkonstruktion bzw. des Anemometers und somit die Registrierung des Magnetfeldes ausgehend von Neodym-Magneten durch den internen Hall-Sensor des ESP32. Somit konnte der Microcontroller ESP32 die Umdrehungen des pro Minute erfassen. Die Auswertung und Zeitmessung erfolgten mithilfe eines selbst entwickelten Python-Skripts, das die eingehenden Signale analysierte.

Der Zusammenhang zwischen Windgeschwindigkeit und den registrierten Signalspitzen (Peaks) ist direkt: Je stärker der Wind, desto höher die Drehzahl des Anemometers und damit die Anzahl der erfassten Magnetfeldänderungen (Peaks) pro Minute.

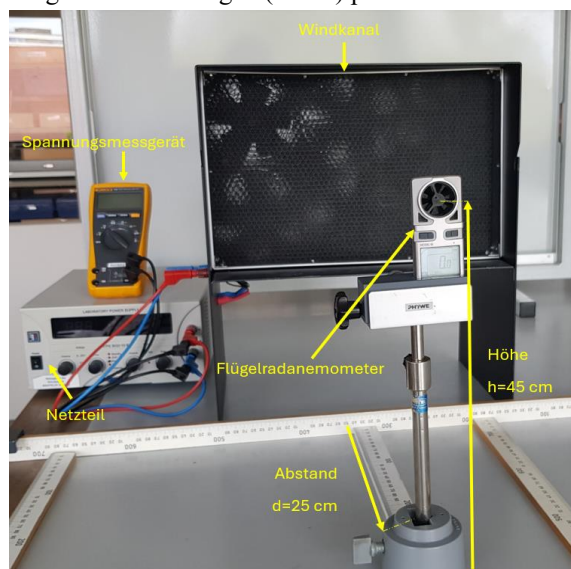


Abb. 7: Kalibrierungsprozess des Anemometers. Eigene Darstellung.

Zur Kalibrierung des selbstgebauten Anemometers wurde ein Windkanal verwendet [11], dessen Windgeschwindigkeit variiert werden konnte durch das

Einstellen verschiedener Spannungswerte am Netzteil. Zunächst wurde die vom Windkanal produzierte Windgeschwindigkeit mithilfe eines professionellen Flügelradanemometers (vgl. Abbildung 7) gemessen, um eine exakte Referenz zu erhalten.

Im nächsten Schritt wurde das selbstgebaute Anemometer vor Windkanal positioniert. Das experimentelle Setup ist identisch geblieben. Der ESP32 erfasste die Anzahl der Umdrehungen pro Minute (RPM) des Anemometers in Abhängigkeit von der eingestellten Spannung. Diese Daten ermöglichten eine Zuordnung der Windgeschwindigkeit zur gemessenen Drehzahl.

Auf Basis der erhobenen Messwerte wurde ein Kalibrierungsdiagramm (vgl. Abbildung 8) erstellt, das die Windgeschwindigkeit als Funktion der Drehzahl darstellt. Der resultierende Graph zeigt einen nahezu linearen Verlauf, was auf eine direkte Proportionalität zwischen Drehzahl und Windgeschwindigkeit hinweist [12]. Diese Kalibrierung ermöglichte es, aus der gemessenen Drehzahl des Anemometers präzise Rückschlüsse auf die tatsächliche Windgeschwindigkeit zu ziehen.

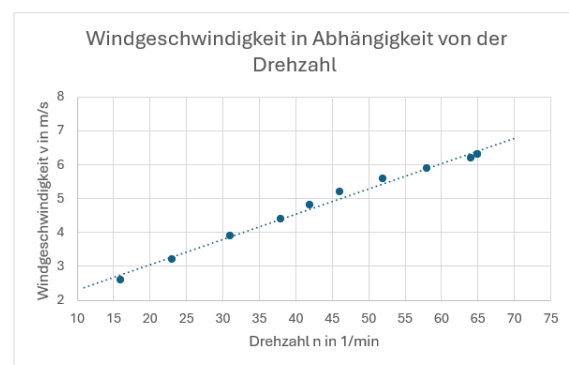


Abb. 8: Ergebnis der Kalibrierung des Anemometers und die damit indirekt Bestimmung der Windgeschwindigkeit mit dem Anemometer und ESP32. Eigene Darstellung.

6.2. Objekterkennung und -identifikation mit dem ESP32-CAM-Modul

Künstliche Intelligenz (KI) ist aus dem Alltag nicht mehr wegzudenken. Dieses Projekt zeigt, wie KI-basierte Bildverarbeitung auch auf kostengünstiger Hardware realisiert werden [13] kann und eröffnet spannende Möglichkeiten für den MINT-Unterricht.

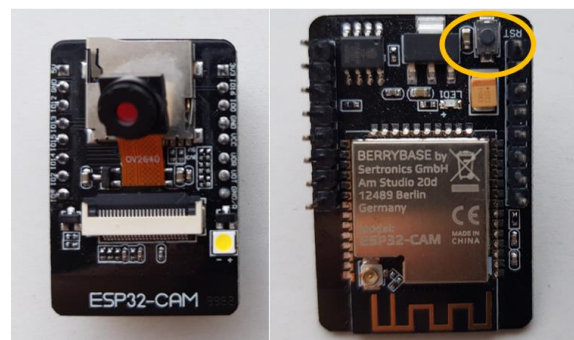


Abb. 9: ESP-32-CAM-Modul. Eigene Darstellung.

In diesem Projekt wurde die Objekterkennung und –identifikation mithilfe des ESP32-CAM-Moduls (vgl. Abbildung 9), einem weiteren Mitglied der ESP32-Familie, realisiert.

Ziel des Projekts war es, einen Livestream über das Kameramodul des ESP32 bereitzustellen, der auf einem integrierten Webserver gehostet wird. Während des Livestreams erkennt das System Objekte in Echtzeit, indem es sie mit einer begrenzenden Box (Bounding Box) umrahmt und mit einer Klassifikation samt Erkennungswahrscheinlichkeit versieht [14] – wie in der Abbildung 10 am Beispiel eines Buches zu sehen ist.

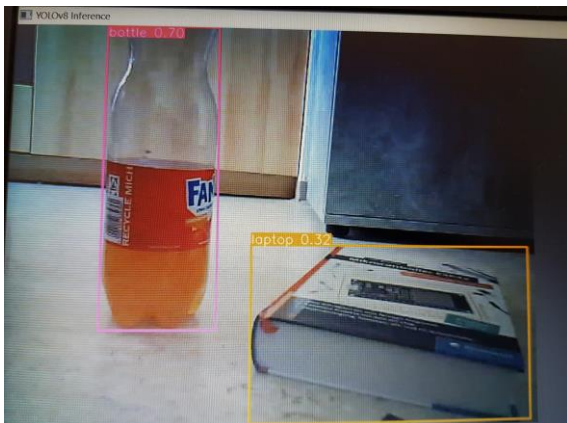


Abb. 10: Erkennung der Objekte durch Einrahmung und Benennung der eingerahmten Objekte, inklusive Wahrscheinlichkeitsanzeige für die Erkennungssicherheit. Eigene Darstellung.

Die Objekterkennung basiert auf YOLO (You Only Look Once), einem vortrainierten KI gestützten neuronalen Netzwerk, das für seine hohe Geschwindigkeit und Präzision bekannt ist.

7. Fazit: der ESP32 im MINT-Unterricht

Der ESP32 stellt ein wertvolles Werkzeug für einen kontextorientierten MINT-Unterricht dar und bietet einen klaren schulischen Mehrwert. Besonders hervorzuheben ist seine flexible Programmierung, die ihn im Vergleich zu anderen Mikrocontrollern auszeichnet. Während der interne Hall-Sensor lediglich qualitative Aussagen zulässt, ermöglicht die Bluetooth-Funktion den Einsatz mobiler Apps als Alternative zu traditionellen Computern. Zwei exemplarische Projekte veranschaulichen diese besonderen Eigenschaften des ESP32 und verdeutlichen sein Potenzial für den Einsatz im Unterricht.

8. Ausblick- Perspektiven des ESP32 im MINT-Unterricht und darüber hinaus

Der ESP32 eröffnet vielfältige Perspektiven – insbesondere durch seine Bluetooth-Funktion und die Anbindung an das Internet of Things (IoT). Anwendungen wie Smart-Home-Systeme oder automatisierte Bewässerungssysteme verdeutlichen das Potenzial des Mikrocontrollers für praxisorientierte Projekte. Darüber hinaus schafft der ESP32 neue Möglichkeiten für den Einsatz von Künstlicher Intelligenz, was

ihn nicht nur für den regulären Unterricht, sondern auch für Forschungs-AGs und Wettbewerbe wie "Jugend forscht" von Interesse macht.

Durch die Arbeit mit dem ESP32 erwerben Schüler und Schülerinnen wertvolle Zukunftskompetenzen in den Bereichen Programmierung, Sensorik, KI und Automatisierung und vertiefen gleichzeitig ihre naturwissenschaftliche Arbeitsweise.

9. Literatur

- [1] Läufer, T.; Oehler, D.-X. K.; Wetzel, S.; Ludwig, M.: Mikrocontrollerprojekte: Unterrichtsrelevante Beispiele aus dem Digitechnikum. In: MNU Journal/Verband zur Förderung des MINT-Unterrichts, Bd. 76 (2023), Nr. 2, S. 117–124
- [2] SINUS-Institut: Was motiviert zum MINT-Lernen? Bonn: Deutsche Telekom Stiftung, 2024
- [3] Espressif Systems: ESP32-WROOM-32 Datasheet. URL: https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32_datasheet_en.pdf (Stand 05/2025)
- [4] Brandes, U.: Mikrocontroller ESP32 – Das umfassende Handbuch. 1. Aufl. Bonn: Rheinwerk, 2020
- [5] Bartmann, E.: Das ESP32-Praxisbuch: Programmieren mit der Arduino-IDE. 1. Aufl. München: Franzis, 2018
- [6] Schmidt, M.: Magnetfelder messen. In: Make Magazin, Bd. 14 (2019), S. 50–55.
- [7] Dorsel, D.: Sensordaten drahtlos zur Smartphone-App Phyphox übertragen und grafisch auswerten – ein einfaches Beispiel mit dem ESP32 und dem Ultraschallsensor HC-SR04. In: Schulpraxis. Januar 2023. URL: https://phyphox.org/p/mnu-arduino/mnu_journal_01_36-43.pdf (Stand 05/2025)
- [8] Parchmann, I.; Kuhn, J. (2018): Lernen im Kontext. In: Krüger, Dirk; Parchmann, Ilka; Schecker, Horst (Hrsg.): Theorien in der naturwissenschaftsdidaktischen Forschung. Berlin: Springer, S. 193–207
- [9] Kraus, H. (2004): Die Windgeschwindigkeit. In: Kraus, H. (Hrsg.): Die Atmosphäre der Erde. Springer, Berlin, Heidelberg, S. 33–44. URL: https://doi.org/10.1007/3-540-35017-9_6 (Stand 05/2025)
- [10] Emeis, S. (2022): Quellen für Winddaten. In: Windenergie Meteorologie. Springer Vieweg, Cham, S. 195–244. URL: https://doi.org/10.1007/978-3-031-22446-1_7 (Stand 05/2025)
- [11] Deiss, O. (2001): Turbulenzeinflüsse bei der Messung der Windgeschwindigkeit. In: Fachhochschule Düsseldorf, Labor für Strömungsmaschinen, Fachbereich Maschinenbau und Verfahrenstechnik

- [12] Manwell, J. F., McGowan, J. G., Rogers, A. L. (2009): Wind Energy Explained: Theory, Design and Application. 2. Aufl. New York: Wiley, S. 74–84
- [13] Mehendale, N.: Object Detection using ESP32 CAM. In: SSRN Electronic Journal. 2022. DOI: <https://doi.org/10.2139/ssrn.4152378>
- [14] C, P.; L, V.; K, P.; S, P.; Rahiman, S.: Enhanced IoT Devices with Intelligent System for Object Detection. In: 2023 9th International Conference On Smart Structures And Systems, 2023. DOI: <https://doi.org/10.1109/icsss58085.2023.10407058>