



**HOCHSCHULE OSNABRÜCK**  
UNIVERSITY OF APPLIED SCIENCES

# **Praktikum im Fach**

## **Internet of Things (IoT) / Industrie 4.0**

### **Basisinformationen zur Praktikums-Hardware und -Software**

**Autoren:** Marco Schaarschmidt (m.schaarschmidt@hs-osnabrueck.de)  
Nicolas Lampe (n.lampe@hs-osnabrueck.de)  
**Version:** 2.1

#### **Wichtige Hinweise:**

- Bei Fragen oder Rückmeldungen zum Praktikumsversuch stellen Sie diese bitte primär zu den Praktikumszeiten (dienstags 14:30 bis 16:00). Bei dringenden Fällen schreiben Sie eine E-Mail an Nicolas Lampe (n.lampe@hs-osnabrueck.de) oder Sebastian Böhm (sebastian.boehm@hs-osnabrueck.de).

# I Inhaltsverzeichnis

<b>I</b>	<b>INHALTSVERZEICHNIS .....</b>	<b>I</b>
<b>II</b>	<b>ABBILDUNGSVERZEICHNIS .....</b>	<b>I</b>
<b>III</b>	<b>TABELLENVERZEICHNIS .....</b>	<b>1</b>
<b>1</b>	<b>EINLEITUNG .....</b>	<b>2</b>
<b>2</b>	<b>HARDWAREPLATTFORM .....</b>	<b>3</b>
2.1	AUFBAU DES M5STACKS .....	3
2.2	BEDIENUNG DES M5STACKS .....	4
<b>3</b>	<b>TOOLCHAIN .....</b>	<b>6</b>
<b>4</b>	<b>SENSOREN UND AKTOREN .....</b>	<b>9</b>
4.1	RGB LED BAR M5Go BOTTOM .....	9
4.2	9-DEGREES-OF-FREEDOM SENSOR .....	10
4.3	ENV SENSOR .....	10
4.4	EARTH MOISTURE SENSOR .....	11
4.5	GPS MODUL .....	11
4.6	HEART RATE SENSOR .....	12
4.7	LIGHT SENSOR .....	12
4.8	M5STACK TIMER CAMERA .....	13
<b>5</b>	<b>WIFI UND HTTP-SERVER .....</b>	<b>14</b>
<b>6</b>	<b>ALLGEMEINES ZUR DURCHFÜHRUNG UND ABGABE DER VERSUCHE .....</b>	<b>15</b>
<b>7</b>	<b>FAQ .....</b>	<b>16</b>
<b>8</b>	<b>METHODENLISTE DER M5STACK-BIBLIOTHEK .....</b>	<b>17</b>
<b>9</b>	<b>LITERATURVERZEICHNIS .....</b>	<b>19</b>

# II Abbildungsverzeichnis

ABBILDUNG 1: AUFBAU DES M5 Go BOTTOM MODULS .....	3
ABBILDUNG 2: M5 Go BOTTOM MODUL UND M5 Go CORE MODUL .....	4
ABBILDUNG 3: ESP 32 DER FIRMA ESPRESSIF SYSTEMS .....	4
ABBILDUNG 4: ESP 32 PIN LAYOUT [Esp19] .....	4
ABBILDUNG 5: EINFÜGEN DER BOARDVERWALTER-URL .....	6
ABBILDUNG 6: WAHL DES BOARDMANAGERS ESP32 BY ESPRESSIF SYSTEMS .....	7
ABBILDUNG 7: AUFBAU DER ARDUINO SOFTWARE IDE .....	7
ABBILDUNG 8: ERGEBNIS DES BEISPIELS "HELLOWORLD" .....	7
ABBILDUNG 9: ENV SENSOR [M5S21] .....	9
ABBILDUNG 10: EARTH MOISTURE SENSOR [M5S21] .....	9
ABBILDUNG 11: GPS MODUL [M5S21] .....	9
ABBILDUNG 12: LIGHT SENSOR [M5S21] .....	9
ABBILDUNG 13: REIHENFOLGE DER LEDs AM M5STACK .....	10
ABBILDUNG 14: FUNKTIONSWEISE EINES GPS-EMPFÄNGERS [Chi18] .....	12
ABBILDUNG 15: VENN-DIAGRAMM ZU WIRELESS, WLAN UND WI-FI [TEC21] .....	14

### III Tabellenverzeichnis

TABELLE 1: ARTEN VON HTTP-STATUSCODES [WIK21] .....	14
TABELLE 2: WICHTIGE STATUSCODES UND DEREN BEDEUTUNG [WIK21] .....	14

## 1 Einleitung

Das Praktikum zur Veranstaltung Internet of Things / Industrie 4.0 soll die Vorlesungsinhalte durch praktische Anwendungsbeispiele ergänzen. Dazu wird für die Praktikumsversuche ein M5Stack verwendet und mittels der Arduino IDE über C++ programmiert. Dieses Handbuch vermittelt die Grundlagen für den korrekten Umgang mit der Hardware und für die richtige Programmierung. Beginnend wird in Kapitel 2 die Hardware und darauf aufbauend in Kapitel 3 die Toolchain zur Programmierung des M5Stacks vorgestellt. Es folgt eine kurze Vorstellung der im Praktikum verwendeten Sensoren und Aktoren in Kapitel 4 und eine kurze Einführung in die Kommunikation über das WLAN in Kapitel 5.

Eine Einführung in die C++ Programmierung wird im Folgenden nicht gegeben. Grundlegende C++-Kenntnisse sind Voraussetzung für das Praktikum und können gegebenenfalls durch Literatur ergänzt werden. Da C++ eine der am meisten verwendeten Programmiersprachen ist, stehen online unzählige Lernvideos zur Verfügung.

Für die Durchführung des Praktikums wird ein sogenannter „IoT/I40-Koffer“ benötigt, der alle Komponenten und Bauteile enthält. Dieser Koffer kann im Rahmen der ersten Praktikumsveranstaltung oder beim TI-Support (Raum: SI0026) ausgeliehen werden und unterliegt für die Leihdauer Ihrer Obhut. Die Zeiten zur Ausgabe der Hardware wird im Rahmen der Vorlesung und ebenfalls per Mail bekannt gegeben. Es ist ausdrücklich erlaubt und erwünscht, dass Sie diesen Koffer auch außerhalb der Praktikumstermine einsetzen können, beispielsweise begleitend zur Vorlesung für eigene Experimente.

## 2 Hardwareplattform

Im Rahmen des Praktikums wird als Entwicklungsplattform der in Abbildung 1 dargestellte M5Stack der Firma M5Stack Technology Co. Ltd. verwendet, welcher mit einem ESP32 der Firma Espressif Systems ausgestattet ist. Der M5 Stack besteht in der grundlegenden Ausstattung aus einem M5 Go CORE und dem M5 Go Bottom und beinhaltet somit bereits einen Grafikdisplay, Taster, MicroSD-Kartensteckplatz, USB-C Anschluss, Lautsprecher und drei Erweiterungsstecker. Über letztere können mittels der Grove Kabel Sensoren angeschlossen werden. Weiterhin lassen sich an das CORE Modul weitere Module anschließen wie beispielsweise das GPS-Modul. Eine Einführung zu den jeweiligen Sensoren kann dem Kapitel 4 entnommen werden. Im Folgenden wird der M5 Stack sowie die grundlegende Bedienung vorgestellt.

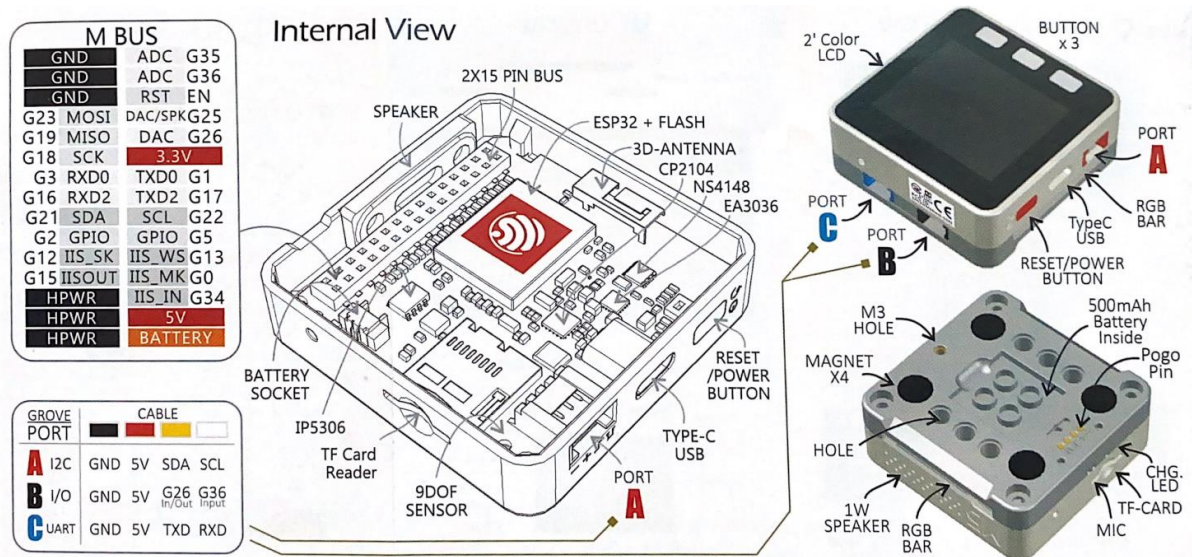


Abbildung 1: Aufbau des M5 Go Bottom Moduls

### 2.1 Aufbau des M5Stacks

Zentraler Bestandteil des M5Stacks ist das M5 Go Core Modul mit dem ESP 32 System-on-a-Chip (SoC). In Abbildung 1 ist der M5Stack mit den einzelnen Bestandteilen dargestellt. Der ESP 32 ist bereits im Core Modul integriert. Abbildung 2 zeigt den aufgeschraubten M5Stack. Mittels eines Inbusschlüssels lässt sich der M5Stack aufschrauben und somit durch weitere Module ergänzen. Links dargestellt ist das Bottom Modul mit der Batterie. Rechts dargestellt ist das M5 Go Core Modul.

In Abbildung 3 ist ein ESP 32 dargestellt. Über die einzelnen Pins lassen sich beispielsweise Sensoren und Aktoren anschließen. Dabei haben die einzelnen Pins unterschiedliche Funktionen, die im Pin-Layout in Abbildung 4 dargestellt sind. Da der ESP 32 bereits im M5Stack integriert ist, vereinfacht dies das Anschließen einzelner Sensoren und Aktoren.

Wie in Abbildung 1 dargestellt, sind bereits Sensoren und Aktoren im Bottom Modul integriert. Dies enthält einen 9 Degrees of Freedom (DOF) Sensor, einen Lautsprecher und eine 3D-Antenne. Weiterhin sind die Grove Ports A, B und C für die Anbindung weiterer Sensoren enthalten. Beispielsweise lässt sich an Port A mittels des Grove Kabels der ENV Sensor zur Messung von Temperatur, Luftfeuchte und Luftdruck anschließen. Dieser benötigt eine I2C Kommunikation. Dies bedeutet wiederum in Bezug auf den ESP 32, dass im Bottom Modul von Port A aus die Anschlüsse GND (Ground = Boden/Erdung) als Bezugspotential, 5V, sowie SCL (= Serial Clock) als Taktleitung und SDA (= Serial Data) als Datenleitung verwendet werden.

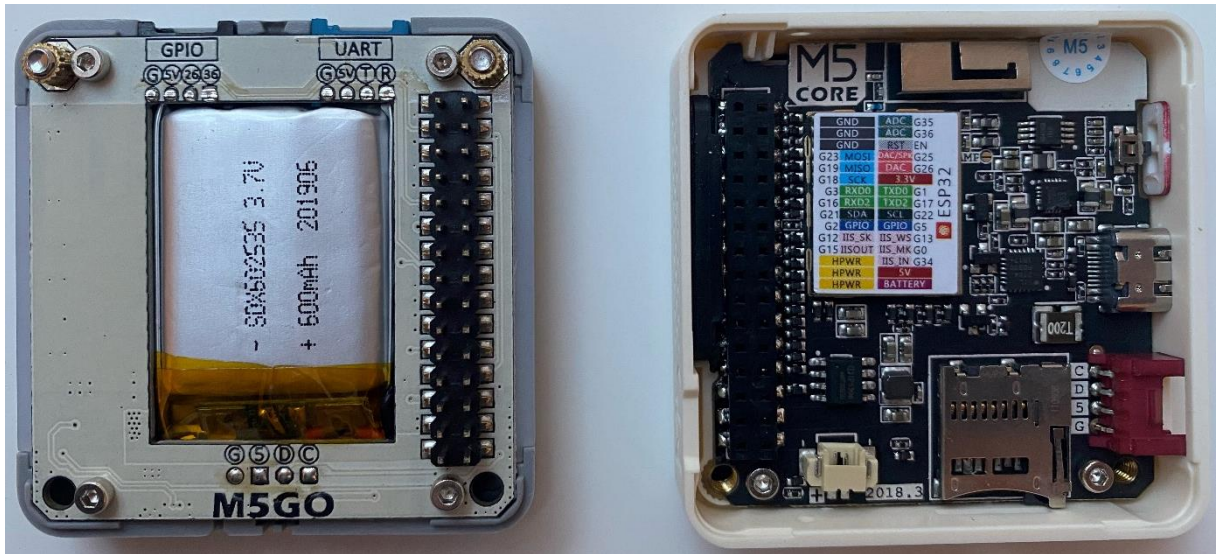


Abbildung 2: M5 Go Bottom Modul und M5 Go Core Modul



Abbildung 3: ESP 32 der Firma Espressif Systems

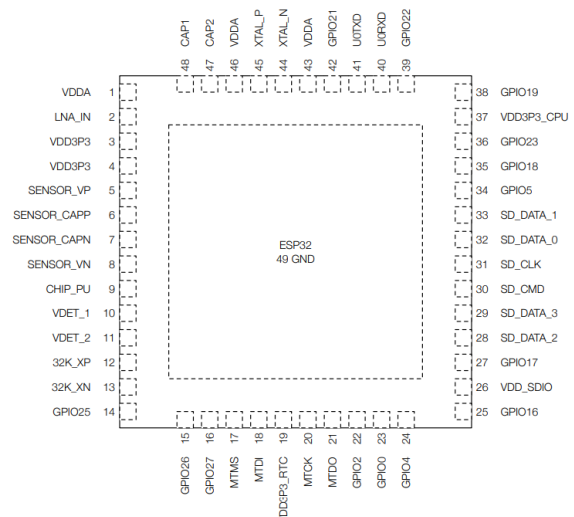


Abbildung 4: ESP 32 Pin Layout [Esp19]

## 2.2 Bedienung des M5Stacks

Programmiert wird der M5Stack im Rahmen des Praktikums mittels der Arduino IDE über C++. Arduino ist bereits auf den Pool-Rechnern vorinstalliert. Dazu ist es notwendig Skripte,

in Arduino auch Sketches genannt, auf den M5Stack zu spielen. Dies erfolgt entweder über das Wifi oder über den USB-C Anschluss. Die Toolchain für eine erfolgreiche Programmierung des M5 Stacks mit Arduino Sketches wird in Kapitel 3 vorgestellt.

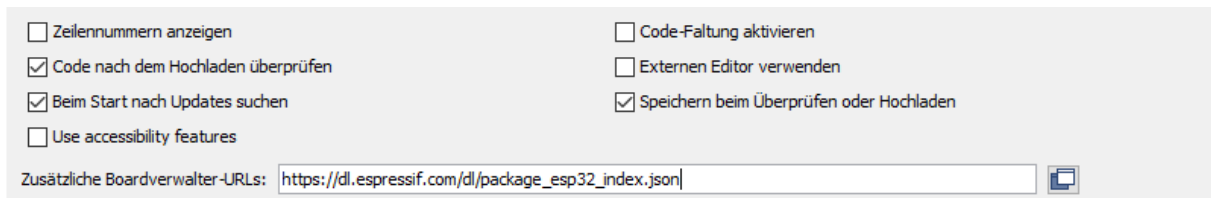
Zu Beginn muss der M5Stack über das USB-C Kabel geladen werden. Über den Power Button lässt sich der M5Stack an und ausstellen. Nach dem Starten des M5Stacks gelangt man in ein Hauptmenü bei dem man die Optionen Upload, App List und Setup hat. Über Setup lässt sich auswählen, ob der M5Stack über Wifi oder über die USB-C Schnittstelle mit Programmen gespielt werden soll. Hier soll die Verbindung über den COM Port ausgewählt werden.



### 3 Toolchain

Der M5Stack soll bei den Praktikumsversuchen in der Entwicklungsumgebung Arduino mittels C++ programmiert werden. Im Folgenden ist die Toolchain für eine Programmierung des M5Stacks mit der Arduino IDE dargestellt.

1. Installation der Arduino Software IDE (Anleitung unter [Ard20]).
2. Installation des CP2104 Treibers, wahlweise für Windows<sup>1</sup>, Linux<sup>2</sup> oder MacOS<sup>3</sup>
3. Schließe den M5Stack über das USB-C Kabel an deinen Computer an.
4. Öffne Sie die Arduino Software IDE auf Ihrem Computer.
5. Einfügen der Boardverwalter-URL zur Nutzung der Arduino Software:
  - a. Gehe auf *Datei* → *Voreinstellungen*
  - b. Gebe bei zusätzliche Boardverwalter-URLs folgende URL ein:  
`https://dl.espressif.com/dl/package_esp32_index.json`
  - c. Mit *Ok* bestätigen (siehe Abbildung 5)



**Abbildung 5: Einfügen der Boardverwalter-URL**

6. Wahl des Boardmanagers:
  - a. Gehe auf *Werkzeuge* → *Board: „Arduino Uno“* → *Boardverwalter...*
  - b. Warten bis alle Boardverwalter geladen sind.
  - c. Eingabe von „esp“ in die Suchleiste.
  - d. Installieren des Boardverwalters esp32 von Espressif Systems und im Anschluss schließen (siehe Abbildung 6)
  - e. Über *Werkzeuge* → *Board: „Arduino Uno“* → *ESP 32 Arduino* → *M5Stack-Core-ESP32* auswählen.
7. Manuelles Installieren von Paketen:
  - a. Gehe auf *Sketch* → *Bibliothek Einbinden* → *.ZIP-Bibliothek hinzufügen*
  - b. Wähle die mit den Unterlagen bereitgestellte Datei aus *M5Stack.zip*.
  - c. Für den dritten Praktikumsversuch werden außerdem die beiden Bibliotheken *TinyGPSPlus-1.0.2.zip* und *InfluxDB-Client-for-Arduino-master.zip* benötigt, die auf gleiche Art und Weise hinzuzufügen sind.

<sup>1</sup> CP2104 Windows-Treiber: [https://m5stack.oss-cn-shenzhen.aliyuncs.com/resource/drivers/CP210x\\_VCP\\_Windows.zip](https://m5stack.oss-cn-shenzhen.aliyuncs.com/resource/drivers/CP210x_VCP_Windows.zip)

<sup>2</sup> CP2104 Linux-Treiber: [https://m5stack.oss-cn-shenzhen.aliyuncs.com/resource/drivers/CP210x\\_VCP\\_Linux.zip](https://m5stack.oss-cn-shenzhen.aliyuncs.com/resource/drivers/CP210x_VCP_Linux.zip)

<sup>3</sup> CP2104 MacOS-Treiber: [https://m5stack.oss-cn-shenzhen.aliyuncs.com/resource/drivers/CP210x\\_VCP\\_MacOS.zip](https://m5stack.oss-cn-shenzhen.aliyuncs.com/resource/drivers/CP210x_VCP_MacOS.zip)



8. Wahl des Ports über *Werkzeuge* → *Port* → Auswahl des Ports über den der M5Stack angeschlossen ist. Dieser kann über den Gerätemanager identifiziert werden.



Abbildung 6: Wahl des Boardmanagers esp32 by Espressif Systems



Abbildung 7: Aufbau der Arduino Software IDE



Abbildung 8: Ergebnis des Beispiels "HelloWorld"

Nach erfolgreichem Absolvieren der Punkte 1 bis 7 können nun Programme (bzw. in Arduino Sketches) auf den M5Stack geladen werden. Um dies vorerst an einem einfachen Beispiel zu testen, soll im Folgenden das Standardbeispiel „HelloWorld“ getestet werden:

1. Stelle den M5Stack auf USB-Mode.
2. Geben Sie den in Abbildung 7 dargestellten Code in der Arduino IDE ein und speichern Sie die Datei unter „HelloWorld.ino“.
3. Klicke auf Hochladen (siehe Abbildung 7).

Es sollte das Ergebnis aus Abbildung 8 erscheinen. Der grundlegende Aufbau der Arduino IDE ist in Abbildung 7 dargestellt. Für weitere Informationen zu der Arduino IDE sei auf [Bru19] verwiesen.

Für die im Praktikum eingesetzten Sensoren werden weitere Bibliotheken benötigt, die wie unter Punkt 7 beschrieben, entsprechend eingebunden werden müssen. Dazu zählt beispielsweise die Bibliotheken *Adafruit NeoPixel* zur Ansteuerung der LEDs.

## 4 Sensoren und Aktoren

In den folgenden Versuchen wird der M5Stack jeweils um weitere Sensorik ergänzt. Im M5Stack sind bereits ein 9-Degrees-of-Freedom (9-DoF) Sensor und ein Mikrophon verbaut. Es gibt hingegen weitere Sensoren, die über die I2C-, I/O und UART-Schnittstelle an den M5Stack angeschlossen und ausgewertet werden können. Im Folgenden werden die in den jeweiligen Versuchen verwendete Sensorik und Aktorik vorgestellt.



Abbildung 9: ENV Sensor [M5S21]



Abbildung 10: Earth Moisture Sensor [M5S21]



Abbildung 11: GPS Modul [M5S21]



Abbildung 12: Light Sensor [M5S21]

### 4.1 RGB LED Bar M5Go Bottom

Innerhalb des M5Go Bottom ist auf jeder Seite eine NeoPixel RGB LED Bar vom Typ SK6812 eingebaut, die auf jeder Seite jeweils 5 RGB LEDs bereitstellt. Die LEDs beginnen dabei von der Zählung her von rechts oben (Sichtweise siehe Abbildung 13) nach rechts unten und dann von links unten nach links oben. LED9 sitzt also links oben und LED4 sitzt rechts unten.

Um die LEDs innerhalb der Arduino IDE nutzen zu können, muss über den Reiter „Werkzeuge“ und dann „Bibliothek verwalten“ die „Adafruit NeoPixel“ Bibliothek installiert werden. Innerhalb des „Dokumente“ Ordners auf dem Computer kann unter „Arduino“ und anschließend im Unterordner „libraries“ die „Adafruit\_NeoPixel“ Bibliothek gefunden werden. Schauen Sie

sich bezüglich der Verwendung dieser die „Adafruit\_NeoPixel“ cpp- und h-Datei an, um die Funktionen und den Konstruktor innerhalb Ihres Programms verwenden zu können.



Abbildung 13: Reihenfolge der LEDs am M5Stack

## 4.2 9-Degrees-of-Freedom Sensor

Der 9-DoF Sensor ist bereits im M5Stack integriert und kann die Drehraten um x-, y- und z-Achse, die Beschleunigungen in x-, y- und z-Richtung, sowie den Roll-, Nick- und Gierwinkel des M5Stacks messen. Bei dem im M5Core verbauten 9-DoF Sensor handelt es sich um einen MPU9250 9-Achsen Beschleunigungssensor der Firma InvenSense. Dieser ist hingegen selbst nach dem Aufschrauben nicht sichtbar und liegt zwischen Display. Zur Messung der Beschleunigungen werden mikromechanische Beschleunigungssensoren verwendet, die durch ein Masse-Feder-Dämpfer-System die Auslenkung einer Masse durch die Beschleunigungen messen. In diesem Versuch werden lediglich die gemessenen Beschleunigungen des 9-DoF Sensors benötigt. Der MPU9250 kommuniziert mit dem ESP32 des M5Core über I2C. Als Taktleitung wird GPIO22 und als Datenleitung GPIO21 verwendet [M5S21].

Im Rahmen des Versuches *Road Condition Monitoring* soll der Beschleunigungssensor dazu verwendet werden, die Vertikalbeschleunigung des M5Stacks und ebenfalls des Fahrrads zu messen. Anhand der Beschleunigung kann der Straßentyp und -zustand ermittelt werden. Folglich muss ebenfalls gewährleistet sein, dass der Beschleunigungssensor korrekt und statisch fest am Fahrrad montiert ist und ebenfalls korrekt ausgerichtet ist.

Anmerkung: Je nach Erscheinungsdatum des jeweiligen M5Stacks kann anstatt des MPU9250 ein MPU6886 verbaut sein. Dies ändert hingegen im Folgenden lediglich die anfängliche Definition des Sensors.

## 4.3 ENV Sensor

Weiterhin kann der M5Stack um den ENV Sensor erweitert werden. Der ENV Sensor setzt sich zusammen aus einem DHT12 und einem BMP280. Der DHT12 ist ein Temperatur- und Luftfeuchtigkeitssensor. Bei dem BMP280 handelt es sich um einen Drucksensor, sodass der ENV Sensor in Summe die Temperatur in °C, die relative Luftfeuchte in % und den Luftdruck in bar

messen kann. Der ENV Sensor kommuniziert über I2C mit dem M5Stack, wobei wiederum als Taktleitung der GPIO22 und als Datenleitung der GPIO21 verwendet wird [M5S21]. Der ENV Sensor ist in Abbildung 9 dargestellt.

#### 4.4 Earth Moisture Sensor

Als weitere Einheit kann der in Abbildung 10 dargestellte Earth Moisture Sensor verwendet werden. Dieser misst die Bodenfeuchte über zwei Elektroden, die in den Boden gesteckt werden müssen. Je höher die Feuchtigkeit im Boden, desto besser ist die Leitfähigkeit zwischen den beiden Elektroden. Es resultiert ein geringerer Widerstand zwischen den beiden Elektroden, der in die Bodenfeuchte umgerechnet werden kann. Der Earth Moisture Sensor kommuniziert über die I/O Schnittstelle des M5Stacks [M5S21].

Durch das Potentiometer kann mittels Schraubendreher der Bereich, in dem der Sensor arbeitet, variiert werden. Zur Kalibrierung vom maximalen und minimalen Wert, muss der Sensorwert zunächst in trockener Luft und anschließend in einem Glas mit Wasser aufgenommen werden. Diese so ermittelten Werte dienen für die spätere Programmierung als Grundwert, um einen relativen Anteil in Prozent angeben zu können.

Anmerkung: Um Beschädigungen an dem Sensor zu vermeiden, sollte dieser nur maximal bis zur schwarzen Markierung an den Stäben in den Boden gesteckt werden.

#### 4.5 GPS Modul

Das GPS (Global Positioning System) Modul ist ein Zusatzmodul der Firma M5Stack Technology Co. Ltd. Das GPS bzw. NAVSTAR GPS ist ein globales Navigationssystem basierend auf Satelliten zur Positionsbestimmung. Das GPS-Modul ist ein GPS-Empfänger der wie in Abbildung 14 dargestellt gleichzeitig den Abstand zu mehreren Satelliten misst. Die Positionen der Satelliten können vorhergesagt werden und sind bekannt, sodass durch die Kombination aus dem Abstand zu den Satelliten und deren Position ebenfalls die Position des GPS-Empfängers ermittelt werden kann. Über die Positionsänderung kann ebenfalls die Geschwindigkeit des GPS Empfängers bestimmt werden [XX16].

Das GPS Modul enthält ein NEO-M8N der Firma u-Blox Holding AG [UB21], welches in der Lage ist Daten von bis zu drei globalen Navigationssatellitensystemen (GNSS) zu empfangen. Die Datenverarbeitung ist ebenfalls im GPS Modul integriert, sodass der M5Core über UART die Daten des GPS Moduls empfangen kann. Standardmäßig wird die UART2 Schnittstelle des M5Cores verwendet. Die Baudrate, also die Anzahl der übertragenen Symbole pro Zeitspanne, des GPS Moduls beträgt 9600 pro Sekunde [M5S21b]. Zur Auswertung der vom GPS Modul gesendeten Daten eignen sich verschiedene Arduino-Bibliotheken wie beispielsweise TinyGPS oder TinyGPSPlus. Über diese lässt sich der vom GPS Modul gesendete Character auswerten,

sodass die Anzahl der Satelliten, der HDOP-Wert, der Breitengrad, der Längengrad, das Datum, die Zeit und die Geschwindigkeit des Empfängers ermittelt werden kann [Har21]. Der HDOP (Horizontal Dilution of Precision)-Wert dient der Kontrolle der Genauigkeit des GPS Sensors und gibt einen Wert für die Horizontalgenauigkeit an. Ein sehr guter HDOP-Wert liegt unter 1.0. Dieser Wert muss hingegen im Rahmen des Praktikumsversuchs nicht erreicht werden.

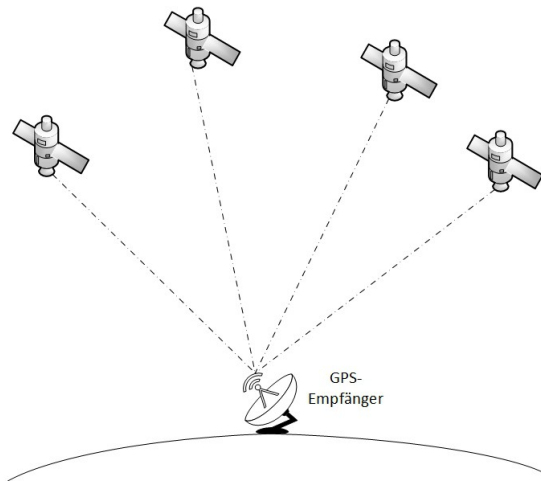


Abbildung 14: Funktionsweise eines GPS-Empfängers [Chi18]

## 4.6 Heart Rate Sensor

Der Heart Rate Sensor MAX 30100 misst die Herzfrequenz und die Sauerstoffsättigung im Blut. Der MAX 30100 funktioniert basierend auf den Absorptionseigenschaften des Hämoglobins im Blut. Sauerstoffhaltiges Hämoglobin absorbiert mehr Infrarotlicht als rotes Licht, während sauerstoffarmes Blut mehr rotes Licht absorbiert. Über Photodetektoren lässt sich das Verhältnis von rotem und infrarotem absorbiertem Licht detektieren und so auf den Sauerstoffgehalt im Blut zurückschließen. Aufgrund des Pulses lässt sich ebenfalls die Pulsfrequenz und die Intensität bestimmen [Rob21]. Der Heart Rate Sensor kommuniziert über I2C mit dem M5Stack und ist in Abbildung 12 dargestellt.

## 4.7 Light Sensor

Der Light Sensor basiert auf einem Photowiderstand (auch als Light Dependent Resistor (LDR) bezeichnet) und ist in Abbildung 12 dargestellt. Dabei handelt es sich um einen lichtgesteuerten variablen Widerstand basierend auf dem fotoelektrischen Effekt. Bei zunehmender Lichtintensität nimmt der Widerstand des Sensors ab und umgekehrt. Zur Herstellung eines Photowiderstands wird eine dünne Schicht eines fotosensitiven Halbleitermaterials (oft Cadmiumsulfid (CdS) oder Cadmiumselenid (CdSe)) mäanderförmig auf einer Keramik-Unterlage aufgebracht. Diese mäanderförmige Struktur ist ebenfalls bei dem Light Sensor von M5Stack zu erkennen. Der Light Sensor kommuniziert ebenfalls wie der Earth Moisture Sensor über die I/O Schnittstelle des M5Stacks [M5S21].

## 4.8 M5Stack Timer Camera

Die M5Stack Timer Camera ist ein Entwicklungsboard für die Bilderkennung. Es verfügt über einen ESP32 Chip und eine Kamera (OV3660) mit einer maximalen Auflösung von 2048x1536. Das Modul bietet 8 MB Speicherplatz durch einen zusätzlichen PSRAM. Es unterstützt auch die Bildübertragung über Wi-Fi und Debugging über den USB-Typ-C-Anschluss.

Im Rahmen des ersten Versuches wird beschrieben, wie die benötigte Software auf der M5Stack Timer Camera installiert wird. Hier bieten sich zahlreiche Einstellungsmöglichkeiten, die für die Durchführung des Versuchs allerdings nicht angepasst werden müssen.



## 5 WiFi und http-Server

Während der Begriff „WLAN“ (Wireless Local Area Network) ein kabelloses Funknetzwerk beschreibt ist „Wi-Fi“ (Wireless Fidelity) eine Zertifizierung die einen IEEE Standard für WLAN-Netze beschreibt (vgl. Abbildung 15). Beim WLAN werden der Netzwerkname als „SSID“ (Service Set Identifier) bezeichnet. Das Protokoll „http“ (Hypertext Transfer Protocol) wird dabei zum Übertragen von Webseiten, Webservices und Weiterem verwendet. Eine Verbindung benötigt einen Server und einen Client. Über sogenannte Statuscodes kann der Status der Verbindung abgefragt werden. Die Arten der Statuscodes sind in Tabelle 1 aufgelistet. Dabei sind die relevantesten Statuscodes in Tabelle 2 dargestellt.

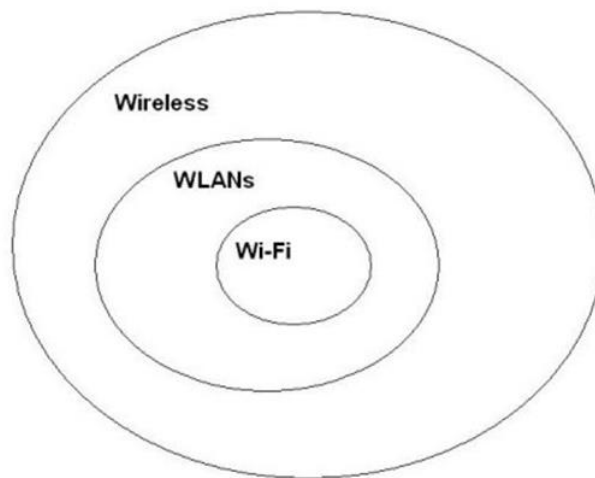


Abbildung 15: Venn-Diagramm zu Wireless, WLAN und Wi-Fi [Tec21]

Tabelle 1: Arten von http-Statuscodes [Wik21]

http-Statuscode	Erklärung
1xx	Informationen
2xx	Erfolgreiche Operationen
3xx	Umleitung
4xx	Client-Fehler
5xx	Server-Fehler

Tabelle 2: Wichtige Statuscodes und deren Bedeutung [Wik21]

http-Statuscode	Nachricht	Bedeutung
200	OK	Die Anfrage wurde erfolgreich bearbeitet
404	Not Found	Die angeforderte Ressource wurde nicht gefunden bzw. die Anfrage wurde ohne Grund abgewiesen

## 6 Allgemeines zur Durchführung und Abgabe der Versuche

Im Rahmen des Praktikums sollen Sie die in der Vorlesung vermittelten Grundlagen durch praktisch orientierte Anwendungsbeispiele vertieft werden. Dazu werden Sie mit der zur Verfügung gestellten Hardware vier verschiedene Versuche durchführen. Bei den Praktikumsterminen werden die Versuche jeweils kurz eingeführt, sodass Sie im Anschluss an den jeweiligen Praktikumstermin mit dem Versuch starten können. Lesen Sie sich dazu die Basisinformationen und ebenfalls die Versuchsbeschreibungen genau durch. Wenn Sie nicht weiterkommen oder Fragen haben, dann können Sie diese bei dem darauffolgenden Praktikumstermin stellen.

Zu jeder Versuchsbeschreibung sind entsprechende Fragen formuliert, die Sie in einem Versuchsprotokoll beantworten müssen. Weiterhin sollen Sie Ihre Eindrücke zu der Vorgehensweise, Schwierigkeit der Versuche und ebenfalls die Probleme, die bei der Bearbeitung der Versuche aufgetreten sind, kurz dokumentieren. Dies dient der Verbesserung des Praktikums für die kommenden Jahre.

Vor dem Testates laden Sie Ihren finalen Sourcecode für den jeweiligen Versuch sowie das entsprechende Versuchsprotokoll im Abgabebereich des OSCA mit der Bezeichnung

`Name_Vorname_Versuch_X.ino`

hoch (X = Versuchsnummer).

Viel Erfolg und Spaß bei der Bearbeitung der Versuche!

## 7 FAQ

Lösungsansätze für folgende Fehlersituationen:

1. Das Board *M5Stack-TimerCam* oder *M5Stack-Core-ESP32* ist unter *Werkzeuge* → *Board*: „*Arduino Uno*“ nicht gelistet.

Wenn das Board nicht aufgelistet ist, können die Einstellungen manuell vorgenommen werden. Dazu wird das Board *ESP32 Wrover Module* ausgewählt und die folgenden Einstellungen vorgenommen:

- Upload Speed: 115200
- Flash Frequency: 40Mhz
- Flash Mode: DIO
- Partition Scheme: Huge APP (3MB No OTA/1MB SPIFFS)
- Code Debug Level: Keine

## 8 Methodenliste der M5Stack-Bibliothek

Die folgende Methodenliste gibt einen Überblick über die Methoden der M5Stack-Bibliothek. Für weitere Methoden, vor allem auch für die Nutzung und Einbindung der jeweiligen Sensoren, sei auf [Git21] verwiesen.

System:

```
M5.begin();  
M5.update();
```

Power:

```
M5.Power.setPowerBoostKeepOn()  
M5.Power.setCharge(uint8_t mode);  
M5.Power.setPowerBoostKeepOn(bool en);  
M5.Power.isChargeFull();  
M5.Power.setWakeupButton(uint8_t button);  
M5.Power.powerOFF();  
bool setPowerBoostOnOff(bool en);  
bool setPowerBoostSet(bool en);  
bool setPowerVin(bool en);  
bool setPowerWLEDSet(bool en); LCD:  
M5.Lcd.setBrightness(uint8_t brightness);  
M5.Lcd.drawPixel(int16_t x, int16_t y, uint16_t color);  
M5.Lcd.drawLine(int16_t x0, int16_t y0, int16_t x1, int16_t y1, uint16_t color);  
M5.Lcd.fillRect(int16_t x, int16_t y, int16_t w, int16_t h, uint16_t color);  
M5.Lcd.fillScreen(uint16_t color);  
M5.Lcd.drawCircle(int16_t x0, int16_t y0, int16_t r, uint16_t color);  
M5.Lcd.drawCircleHelper(int16_t x0, int16_t y0, int16_t r, uint8_t cornername, uint16_t color);  
M5.Lcd.fillCircle(int16_t x0, int16_t y0, int16_t r, uint16_t color);  
M5.Lcd.fillCircleHelper(int16_t x0, int16_t y0, int16_t r, uint8_t cornername, ...  
    int16_t delta, uint16_t color);  
M5.Lcd.drawTriangle(int16_t x0, int16_t y0, int16_t x1, int16_t y1, ...  
    int16_t x2, int16_t y2, uint16_t color);  
M5.Lcd.fillTriangle(int16_t x0, int16_t y0, int16_t x1, int16_t y1, int16_t x2, ...  
    int16_t y2, uint16_t color);  
M5.Lcd.drawRoundRect(int16_t x0, int16_t y0, int16_t w, int16_t h, ...  
    int16_t radius, uint16_t color);  
M5.Lcd.fillRoundRect(int16_t x0, int16_t y0, int16_t w, int16_t h, int16_t radius, ...  
    uint16_t color);  
M5.Lcd.drawBitmap(int16_t x0, int16_t y0, int16_t w, int16_t h, const uint16_t *data, ...  
    uint16_t transparent);  
M5.Lcd.drawBmpFile(fs::FS &fs, const char *path, uint16_t x, uint16_t y);
```

```
M5.Lcd.drawChar(uint16_t x, uint16_t y, char c, uint16_t color, uint16_t bg, uint8_t size);
M5.Lcd.setCursor(uint16_t x0, uint16_t y0);
M5.Lcd.setTextColor(uint16_t color);
M5.Lcd.setTextColor(uint16_t color, uint16_t backgroundColor);
M5.Lcd.setTextSize(uint8_t size);
M5.Lcd.setTextWrap(boolean w);
M5.Lcd.printf();
M5.Lcd.print();
M5.Lcd.println();
M5.Lcd.drawCentreString(const char *string, int dX, int poY, int font);
M5.Lcd.drawRightString(const char *string, int dX, int poY, int font);
M5.Lcd.drawJpg(const uint8_t *jpg_data, size_t jpg_len, uint16_t x, uint16_t y);
M5.Lcd.drawJpgFile(fs::FS &fs, const char *path, uint16_t x, uint16_t y);
M5.Lcd.drawBmpFile(fs::FS &fs, const char *path, uint16_t x, uint16_t y);
```

**Button:**

```
M5.BtnA/B/C.read();
M5.BtnA/B/C.isPressed();
M5.BtnA/B/C.isReleased();
M5.BtnA/B/C.wasPressed();
M5.BtnA/B/C.wasReleased();
M5.BtnA/B/C.wasreleasedFor()
M5.BtnA/B/C.pressedFor(uint32_t ms);
M5.BtnA/B/C.releasedFor(uint32_t ms);
M5.BtnA/B/C.lastChange(); Speaker:
M5.Speaker.tone(uint32_t freq);
M5.Speaker.tone(freq, time);
M5.Speaker.beep();
M5.Speaker.setBeep(uint16_t frequency, uint16_t duration);
M5.Speaker.mute();
```

## 9 Literaturverzeichnis

- Rob21     anzado GmbH: Herzfrequenz Sensor Modul MAX30100 2021. URL <https://www.roboter-bausatz.de/p/herzfrequenz-sensor-modul-gy-max30100>. Zuletzt geprüft: 19.03.2021.
- Ard20     Arduino LLC.: Getting Started with Arduino products 2020. URL <https://www.arduino.cc/en/Guide>. Zuletzt geprüft: 09.03.2022.
- Bru19     Brühlmann, Thomas: Arduino: *Praxiseinstieg*. 4. Auflage, 2019
- Esp19     Espressif Systems Inc.: ESP32 Series Datasheet 2019. Zuletzt geprüft: 09.03.2022.
- Chi18     Flasskamp, M.: Wie funktioniert GPS? Einfach erklärt 2018. URL [https://praxistipps.chip.de/wie-funktioniert-gps-einfach-erklaert\\_41414](https://praxistipps.chip.de/wie-funktioniert-gps-einfach-erklaert_41414). Zuletzt geprüft: 18.02.2021.
- Har21     Hart, Mikal: TinyGPSPlus 2021. URL <https://github.com/mikalhart/TinyGPSPlus>. Zuletzt geprüft: 18.02.2021.
- M5S21b   M5Stack Technology Co. Ltd.: GPS Module with Internal & External Antenna 2021. URL <https://m5stack.com/products/gps-module>. Zuletzt geprüft: 18.02.2021.
- Git21     M5Stack Technology Co. Ltd.: M5 Docs 2022. URL <https://github.com/m5stack/m5-docs/tree/master/docs/en/api>. Zuletzt geprüft: 09.03.2022.
- M5S21     M5Stack Technology Co. Ltd.: M5Stack Units 2022. URL <https://m5stack.com/collections/m5-unit>. Zuletzt geprüft: 09.03.2022.
- Wik21     Sigler, Marian: HTTP-Statuscode 2022. URL <https://de.wikipedia.org/wiki/HTTP-Statuscode>. Zuletzt geprüft: 09.03.2022.
- Tec21     TechTarget: ComputerWeekly 2022. URL <https://www.computer-weekly.com/de/antwort/Was-ist-der-grundsatzliche-Unterschied-zwischen-Wi-Fi-und-WLAN#:~:text=Im%20Grunde%20genom-men%20ist%20Wi,eine%20bestimmte%20Art%20von%20WLAN.&text=Wi%2DFi%20ist%20die%20Abk%C3%BCrzung,Protokoll%2DFamilie%20aus%20>. Zuletzt geprüft: 09.03.2022.
- UB21     U-Blox: NEO-M8 series 2021. URL <https://www.u-blox.com/en/product/neo-m8-series?lang=de>. Zuletzt geprüft: 18.02.2021.
- XX16     Xu, Guochang; Xu, Yan: GPS: *Theory, algorithms and applications*. Third edition. Berlin, Heidelberg: Springer, 2016