



ZSL

Zentrum für Schulqualität
und Lehrerbildung
Baden-Württemberg

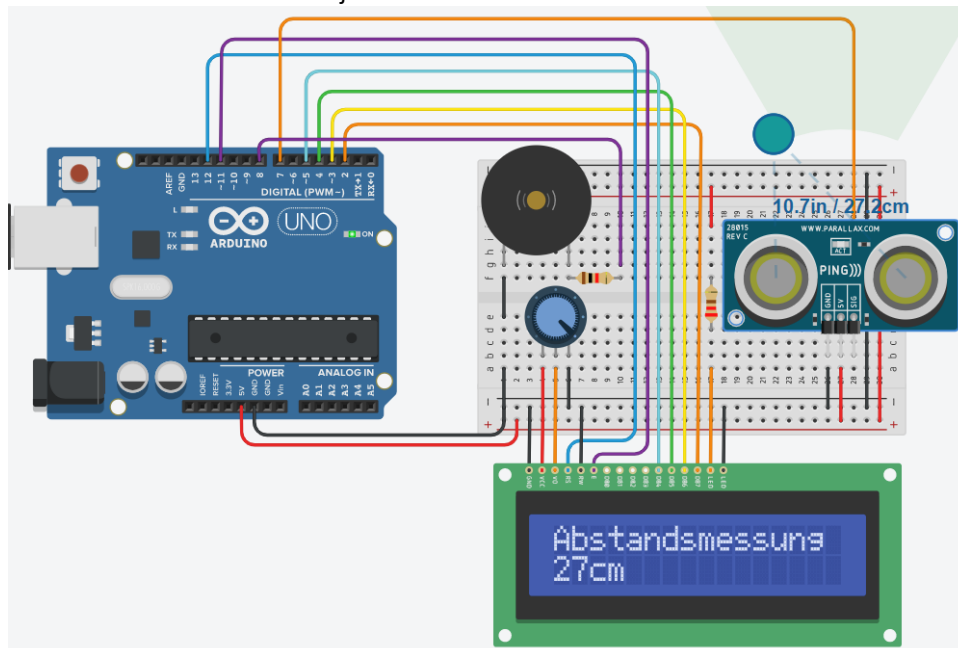
Modulfortbildungen >> **Elektrotechnik**

Präzisierung zu Modulbereich Mikrocontrollertechnik [E61]

Teilmodul E611 | Mikrocontrollerprogrammierung aus dem Homeoffice (mit Tinkercad)

Beschreibung:

Mit Tinkercad können viele interessante Projekte aus dem Homeoffice realisiert werden.



Hierfür muss keine Software auf dem PC installiert werden, da die Anwendung im Internet Browser läuft.

Zielgruppe:

Zielgruppe sind technische und wissenschaftliche KollegInnen unterschiedlicher Schularten welche Elektrotechnische Grundlagen im Fernunterricht unterrichten oder unterrichtet werden.

Ziele des Moduls:

Anmeldung als Lehrer in Tinkercad
Anlegen von Klassen und Anmeldung der Schüler unter Einhaltung der DSGVO
Gas-, Temperatur- und Licht-Messung
Abstands-, Kraft- und Bewegungs-Sensoren
Neo-Pixel, LCD, 7-Segment

Voraussetzungen:

PC-Kenntnisse
Grundwissen in Elektrotechnik

Mindestausstattung für den eigenen Unterricht:

PC-Raum mit Internetzugang

Weiterführende Fortbildungen:

Dieses Teilmodul ist die Grundlage für die weiteren Mikrocontroller-Module E61x. Die Teilmodule E612 oder E613 bauen direkt auf diesem Teilmodul E611 auf.

zuständige Modulkoordinatoren:

Stuttgart und Schwäbisch Gmünd: Rolf Rahm | Karlsruhe und Mannheim: Artur Busch | RPF: Gero Albrecht | RPT: nn



ZSL

Zentrum für Schulqualität
und Lehrerbildung
Baden-Württemberg

Teilmodul E612 | Einstieg in Arduino mit Carrier Board und Multifunction Shields

Beschreibung:

Dieses Modul ist das Einstiegsmodul in die Welt der Mikrocontrollertechnik. Es vermittelt die notwendigen Grundkenntnisse in die Anwendung und Programmierung von Mikrocontrollern am Beispiel des Atmel ATmega328. Die Programmiersprache ist C und als Entwicklungsumgebung wird Arduino eingesetzt. Es werden Kaufempfehlungen für die Hardware geben, damit die notwendige Vertiefung nach der Fortbildung stattfinden kann.

Zielgruppe:

Zielgruppe sind technische und wissenschaftliche KollegInnen unterschiedlicher Schularten welche Mikrocontrollertechnik schon unterrichten oder unterrichten werden.

Ziele des Moduls:

- Anwendungen von Mikrocontrollern
- Prinzipieller Aufbau eines Mikrocontrollers
- Einzelne Komponenten des Mikrocontrollers genauer besprechen
- Einführung in eine Entwicklungssoftware
- Vorgehensweise bei der Programmierung von Mikrocontrollern
- Einfache Beispielprogramme in der Programmiersprache C
- Beschaffung der Entwicklungsumgebung (HW und SW)
- Umsetzungshilfen für den Online-Unterricht.

Voraussetzungen:

- PC-Kenntnisse
- Grundwissen in Elektrotechnik

Mindestausstattung für den eigenen Unterricht:

- PC-Raum mit Internetzugang
- Passende Entwicklungsumgebung (HW und SW)
- Sinnvoll: Multimeter, Labornetzgerät, Oszilloskop,

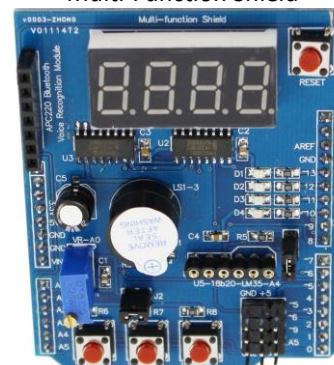
Weiterführende Fortbildungen:

Dieses Teilmodul ist die Grundlage für die weiteren Mikrocontroller-Module E61x. Das Teilmodul E613 baut direkt auf diesem Teilmodul E612 auf.

Arduino



Multi-Function Shield



zuständige Modulkoordinatoren:

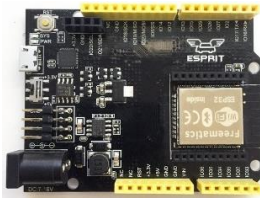
Stuttgart und Schwäbisch Gmünd: Rolf Rahm | Karlsruhe und Mannheim: Artur Busch | RPF: Gero Albrecht | RPT: nn

**ZSL**

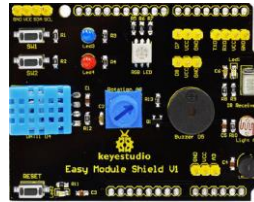
Zentrum für Schulqualität
und Lehrerbildung
Baden-Württemberg

Modulfortbildungen >> **Elektrotechnik**

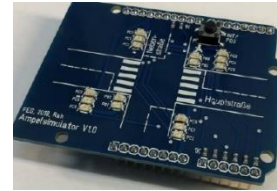
Teilmodul E613 | Einstieg in die ESP32-Programmierung mit Arduino IDE



ESP32 freemotics mit Arduino-IDE



Easy Modul Shield



Ampel Shield

Beschreibung:

Dieses Modul dient zum Einstieg in die ESP32 Mikrocontrollerprogrammierung mit der Entwicklungsumgebung Arduino. Von der Installation, Einbinden des Entwicklungsboards bzw. Bibliotheken bis zur Programmierung mit Programmstrukturen und Unterprogrammtechnik sollen alle Voraussetzungen für eine erfolgreiche Umsetzung im Unterricht gegeben werden.

Ebenso wollen wir die Mikrocontroller spezifischen Hardwarekomponenten wie Interrupt's und Timer konfigurieren und mit praktischen Beispielen anwenden.

Zielgruppe:

Zielgruppe sind Teilnehmer (techn. und wissenschaftliche KollegInnen) unterschiedlicher Schularten die Mikrocontrollertechnik etc. unterrichten.

Ziele des Moduls:

Grundlagenprogrammierung des ESP32 mit didaktischen Umsetzungshilfen von LED_an bis Timer-IR
Programmstrukturen in C/CPP anwenden und üben

Voraussetzungen:

PC-Kenntnisse

Controllerkenntnisse wie Programmierung μ C in C/C++ (evtl. aus Modul E611 + E612)

Mindestausstattung für die LFB:

PC-Raum mit Entwicklungssoftware, PC
Controllersystem

Weiterführende Fortbildungen:

Dieses Teilmodul hat den Umfang von 2 Tagen und setzt auf die Module E611 bis E612 auf.

Einbindung ergänzender Hardwarekomponenten, die Unterrichtsprojekte ermöglichen.

- 010_LED_an
- 015_LEDs_Blink
- 020_LEDs_Blink_Serial
- 030_LED_SOS
- 035_LED_SOS_Unterprogramm
- 040_LED_SOS_Unterprogramm_Strukturen
- 045_LED_SOS_Unterprogramm_Strukturen_Schalter
- 050_LED_SOS_EXT_IR
- 055_LED_SOS_EXT_IR_ADC
- 060_Temp_LM35
- 065_Light_LDR
- 070_Temp_Feuchte_DHT11
- 075_RGB_LED_PWM
- 080_RGB_LED_PWM_IR_ADC_stateMachine
- 085_LED_Blink_Timer_IR
- 090_Timer_Input_Measure_S2_reaction
- 095_Timer_Input_Measure_Infraret_Pin
- 100_RC5Code_Infraret_Pin

zuständige Modulkoordinatoren:

Stuttgart und Schwäbisch Gmünd: Rolf Rahm | Karlsruhe und Mannheim: Artur Busch | RPF: Gero Albrecht | RPT: nn

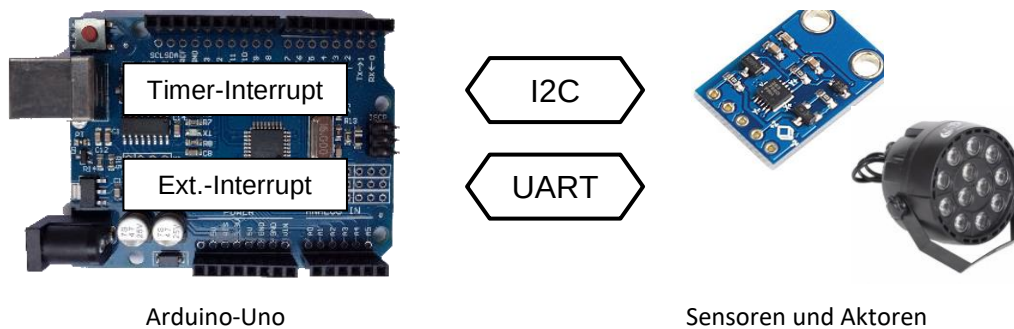


Teilmodul E614 | Aufbaukurs μ C-Programmierung: Interrupts & Schnittstellen (Arduino Uno, ESP32)

Beschreibung:

Aufbauend auf das Grundlagenmodul 611 des vergangenen Jahres, wird hier der Fokus auf die Anwendung von externen Interrupts, vom Timer-Interrupt und auf den Einsatz von Schnittstellen (I2C und UART) gelegt.

Leitlinie ist immer die Vermittlung der theoretischen Hintergründe gepaart mit praktischen Beispielen für den Unterricht.



Zielgruppe:

Zielgruppe sind technische und wissenschaftliche KollegInnen unterschiedlicher Schularten, die aktuelle IoT-Projekte in ihrem Unterricht umsetzen wollen.

Ziele des Moduls:

- Vorteile der Verwendung von externen Interrupts aufzeigen und in einem Schülerprojekt (z.B. Ampelsteuerung) realisieren.
- Realisierung einer Anwendung mit Timer-Interrupts zur ‚genauen‘ Intervallmessung.
- Grundlegende Prinzipien der UART-Schnittstelle verstehen und Datenübertragung zwischen zwei Endgeräten realisieren.
- I2C-Grundlagen anwenden; Sensordaten mit Hilfe dieser Schnittstelle erfassen und auswerten.

Voraussetzungen:

Grundlagen in der Mikrocontrollerprogrammierung!

Mindestausstattung für den eigenen Unterricht:

PC-Raum, Passende Entwicklungsumgebung (Arduino IDE)

Weiterführende Fortbildungen:

Für die Umsetzung der Technischen Richtlinien im EGS-Bereich dient E615.

zuständige Modulkoordinatoren:

Stuttgart und Schwäbisch Gmünd: Rolf Rahm | Karlsruhe und Mannheim: Artur Busch | RPF: Gero Albrecht | RPT: nn

Teilmodul E615 | μ C-Programmierung für EGS mit Technischen Richtlinien (Arduino/ ESP32)

Beschreibung:

Dieses Modul zeigt die Intentionen und den Einsatz der Bibliotheksfunktionen aus den neu erstellten Technischen Richtlinien FA205 für den Beruf Elektroniker für Geräte und Systeme. Die Technischen Richtlinien FA205 werden anhand verschiedener Übungseinheiten vorgestellt und deren Auswirkung auf die zentrale Abschlussprüfung aufgezeigt. Die Bibliotheksfunktionen kommen an zwei Mikrocontroller-Systemen zum praktischen Einsatz:

1. Arduino mit IDE Arduino
2. AVR ATmega328 mit IDE Atmel Studio 7 und erweiterten Debugging-Fähigkeiten

Zielgruppe:

Zielgruppe sind technische und wissenschaftliche KollegInnen, welche die Mikrocontrollertechnik vorzugsweise im Beruf Elektroniker für Geräte und Systeme unterrichten.

Ziele des Moduls:

Programmierung in der Sprache C mit der Technischen Richtlinie FA205

Unterrichtsbeispiele und Hinweise zur Prüfungserstellung

Umsetzung komplexer Lernsituationen (z.B. Lageregelung; einfache IOT-Systeme mit μC und Node-RED, ...)

Umsetzungshilfen für den Online-Unterricht.

Voraussetzungen:

PC-Kenntnisse

Grundwissen in Elektrotechnik

Grundwissen in der Programmiersprache C (siehe Modul E612)

Grundwissen in der Funktionsweise eines Mikrocontrollers
(siehe Modul E612)

Mindestausstattung für den eigenen Unterricht:

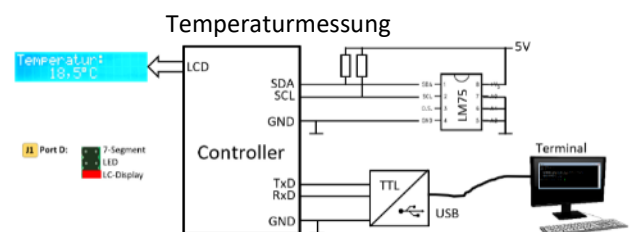
PC-Raum mit Internetzugang

Passende Entwicklungsumgebung (HW und SW)

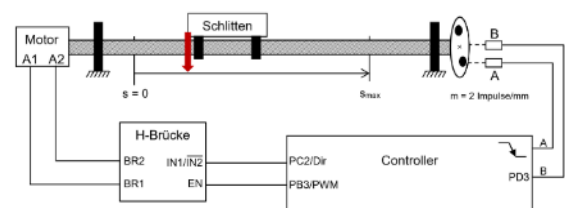
Sinnvoll: Multimeter, Labornetzgerät, Oszilloskop,



Analyse I2C-Busprotokoll (Nunchuk)



Lageregelung mit μC

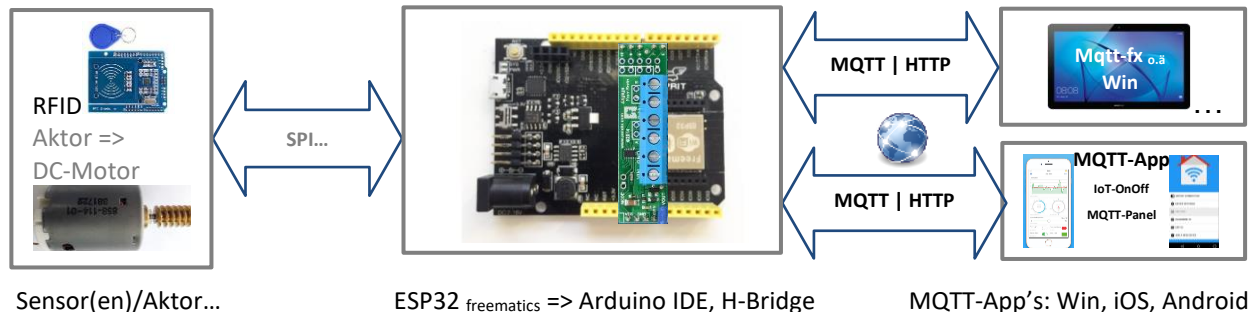


zuständige Modulkoordinatoren:

Stuttgart und Schwäbisch Gmünd: Rolf Rahm | Karlsruhe und Mannheim: Artur Busch | RPF: Gero Albrecht | RPT: nn

**ZSL**Zentrum für Schulqualität
und Lehrerbildung
Baden-WürttembergModulfortbildungen >> **Elektrotechnik**

Teilmodul E616 | IoT mit ESP32: Lernsituation Garagentor, MQTT, HTTP, RFID

**Beschreibung:**

Das IoT ist ein Sammelbegriff für Technologien, die es ermöglichen, physische und virtuelle Gegenstände miteinander zu vernetzen und sie durch Informations- und Kommunikationstechniken zusammenarbeiten zu lassen. Das IoT ist also ein Trend, der sich nicht mehr aufhalten lässt. Wir möchten alles im Haus mit unserem Mobilgerät steuern, fernsehen, die Beleuchtung bedienen und die Heizungs-/Klimaanlage einstellen.... Gleiche Prinzipien werden ebenso im Industrial Internet of Things => IIoT eingesetzt; was Industrie 4.0 entspricht.

Zur Datenübertragung im Ethernet benutzen wir das MQTT bzw. http Protokoll und visualisieren somit die Daten mit MQTT-App's bzw. konfigurieren das *IoT-Produkt* mit dem Browser.

Wir beschreiben exemplarisch das Prinzip einer IoT=IIoT=I4.0-Anwendung und zeigen die konkrete Umsetzung im Unterricht anhand der Lernsituation Garagentor.

Zielgruppe:

Zielgruppe sind Teilnehmer (techn. und wissenschaftliche KollegInnen) unterschiedlicher Schularten die Mikrocontrollertechnik etc. unterrichten.

Ziele des Moduls:

µC-Kommunikation mit Bausteinen wie z. B. RFID-Transponder und H-Brücke zur Ansteuerung von Motoren

Grundlagen Netzwerktechnik (IP, MAC, DHCP...)

Datenaustausch Ethernet | Wifi

Inbetriebnahme einer IoT-Anwendung für den Unterricht:

µC + WiFi-Client => MQTT-Client => MQTT.Broker <=> MQTT-App's => Steuerung des DC-Motors

RFID über SPI <=> µC + WiFi-AP => WebServer <=> Webseite zur Konfiguration von WiFi / MQTT...

Voraussetzungen:

PC-Kenntnisse

Controllerkenntnisse wie Programmierung µC in C (evtl. aus Modul E611 + E612, E613)

Grundkenntnisse Bussysteme (SPI, Ethernet...)

Mindestausstattung für den eigenen Unterricht:

PC-Raum mit Entwicklungssoftware, WLAN/PC

Controllersystem mit busfähigen Komponenten

Weiterführende Fortbildungen:

Dieses Teilmodul hat den Umfang von 2 Tagen und setzt auf die Module E611 bis E614 auf.

Einbindung ergänzender Hardwarekomponenten, die Unterrichtsprojekte ermöglichen.

zuständige Modulkoordinatoren:

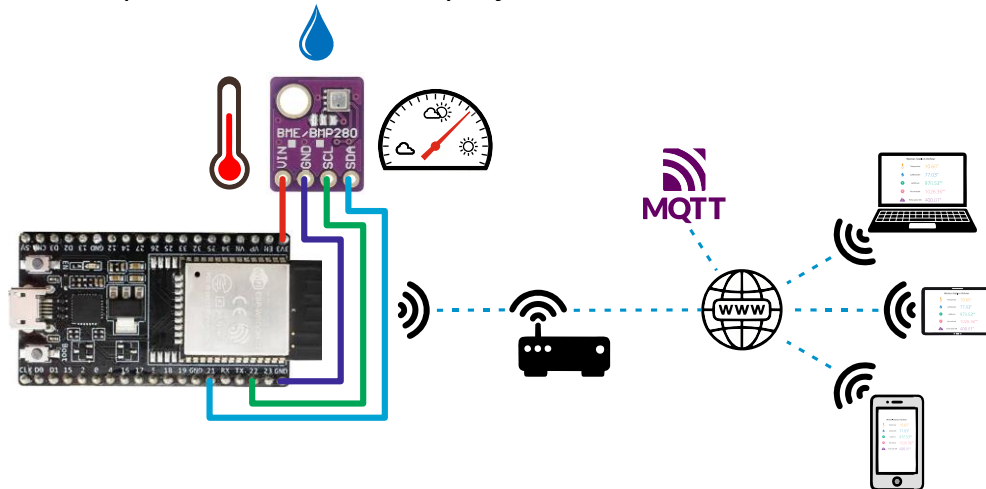
Stuttgart und Schwäbisch Gmünd: Rolf Rahm | Karlsruhe und Mannheim: Artur Busch | RPF: Gero Albrecht | RPT: nn

**ZSL**

Zentrum für Schulqualität
und Lehrerbildung
Baden-Württemberg

Modulfortbildungen >> **Elektrotechnik**

Teilmodul E617 | IoT mit ESP32: Kleinprojekte und embedded Webserver



Beschreibung:

Nach einer kurzen Einführung in die Gestaltung von Internetseiten mit HTML und CSS wird der ESP32-Mikrocontroller als WLAN-Access Point bzw. WLAN-Client in Betrieb genommen und die Funktion mittels eines einfachen Webserverns getestet. Der ESP32 kann nun auf Anfrage eines Clients (Browser auf einem beliebigen PC oder Mobilgerät) zunächst statischen HTML-Code ausgeben. Durch Anbinden von Sensoren und Aktoren an den ESP32 können dann auch dynamische Webseiten mit asynchron aktualisierten Inhalten (z.B. Messdaten von Umweltmesstechnik-Sensoren) ausgeliefert und clientseitig visualisiert werden. Auch die Ansteuerung von angebotenen Aktoren über HTML5-Elemente im Browser (Button, Slider, ...) wird geübt. Mit einer Vorstellung verschiedener Unterrichtsprojekte wird die Fortbildung abgerundet.

Zielgruppe:

Zielgruppe sind technische und wissenschaftliche KollegInnen unterschiedlicher Schularten, die aktuelle IoT-Projekte in ihrem Unterricht umsetzen wollen. Besonders geeignet ist die Fortbildung auch für KollegInnen, die das neue Fach Informatik am TG Mechatronik mit den BPEs 2, 3, 19, 20, 37, 38, 39 und 40 unterrichten.

Ziele des Moduls:

Grundaufbau einer Website mit HTML und CSS

Betrieb des ESP32-Mikrocontrollers als WLAN-Client und WLAN-AccessPoint.

Einrichten eines einfachen Webserverns auf dem ESP32 und Zugriff vom Browser auf dem Laptop/Tablet/Smartphone aus per WLAN, Upload der Webserverinhalte per SPIFFS

Client- und Serverseitige Dynamik durch Anbinden von Sensoren und Aktoren an den ESP32 und Visualisierung bzw. Ansteuerung über den Webserver/Browser

Übertragen von Sensordaten bzw. Ansteuerung von Aktoren per MQTT und Websockets

Vorstellung diverser Unterrichtsprojekte mit dem ESP32 als embedded Webserver (z.B. Wetterstation)

Voraussetzungen:

Grundlagen in der Mikrocontrollerprogrammierung sollten vertraut sein (z.B. aus Teilmodul E613 o. ä.)

Grundkenntnisse in HTML und CSS sind von Vorteil aber nicht zwingend erforderlich.

Mindestausstattung für den eigenen Unterricht:

PC-Raum oder Laptops mit Internetzugang, geeignete Entwicklungsumgebung

HW: ESP32 Dev Kit, div. Sensoren und Aktoren, Breadboards

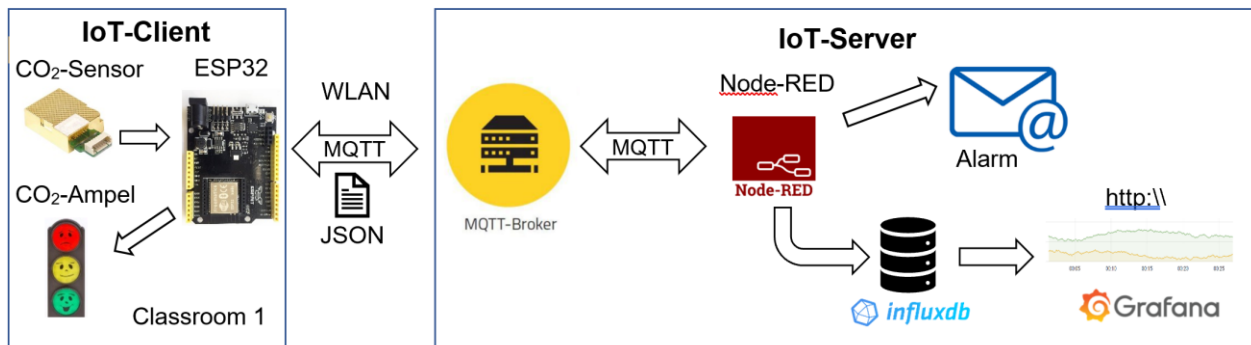
SW: Arduino IDE oder VSCode Editor mit Platform IO Extension

Weiterführende Fortbildungen:

Diese Modulfortbildung E617 umfasst einen Web-Termin (1 Nachmittag) sowie zwei Präsenztage. Weiterführende Fortbildungen sind die Module E616, E618 und E619.

zuständige Modulkoordinatoren:

Stuttgart und Schwäbisch Gmünd: Rolf Rahm | Karlsruhe und Mannheim: Artur Busch | RPF: Gero Albrecht | RPT: nn

**ZSL**Zentrum für Schulqualität
und Lehrerbildung
Baden-Württemberg**Teilmodul E618 | IoT: Messwerterfassung mit Datenbankanbindung und Dashboard****Beschreibung:**

Wie wäre es, wenn der Hausmeister morgens eine E-Mail erhält, dass im Klassenraum der CO₂-Wert zu hoch ist. Ist die Lüftung defekt, oder muss einfach nur das Fenster geöffnet werden? Ein Blick auf das Handy zeigt ihm, dass die CO₂-Konzentration schon seit 2 Tagen durchgängig über dem Grenzwert liegt. Also doch den Lüftungsmonteure bestellen; Oder hat das System das auch schon selbst erledigt?

Am Beispiel einer CO₂-Messung werden wichtige Techniken und Prinzipien des Internet of Things (IoT) vermittelt. Die Sensorwerte werden vom µController (ESP8266/ESP32) per WLAN übertragen. Als zentrales Tool kommt Node-RED zum Einsatz. Es beherrscht die verschiedensten, im IoT verwendeten Übertragungsprotokolle und kann Daten durch grafische Funktionsblöcke oder mittels JavaScript aufbereiten, darstellen und weiterversenden. Speziell auf die Bedürfnisse von verteilten Sensor-/ Aktor-Systemen zugeschnitten, wird hier das MQTT-Protokoll verwendet. Die Sensordaten werden persistent in einer Datenbank gespeichert. Ein Zeitstempel und die SensorID gewährleisten die zeitliche und räumliche Zuordnung der Daten. Mit dem Visualisierungs- und Analysetool Grafana werden die Sensordaten grafisch aufbereitet und in einem browserbasierten dynamischen Dashboard dargestellt. Alle verwendeten Tools lassen sich als Serverdienste auf beliebigen Systemen (Windows, Raspberry, ...) installieren.

Zielgruppe:

Zielgruppe sind technische und wissenschaftliche Kolleg*innen, aus unterschiedlichen Schularten die IoT-Themen unterrichten oder unterrichtet werden. Im neuen Pflichtfach Informatik am TG werden die Wahlthemen: BPE 39, 40 (TGM), bzw. BPE 61, 62 (TGU) abgedeckt.

Ziele des Moduls:

Sensordaten sind per MQTT über IP-basierte Netze verschickt.

Eigene Node-RED-Flows zur Kommunikation und Weiterverarbeitung der Sensordaten sind erstellt.

Eine InfluxDB-Datenbank ist angelegt und die Daten persistent gespeichert.

Ein Dashboard zur Visualisierung der Sensordaten ist in Grafana erstellt.

Möglichkeiten und Gefahren der Daten-Auswertung sind aufgezeigt (Warn- und Alarmschwellen, Langzeitauswertung „Wer vergisst immer zu lüften?!“, ...).

Voraussetzungen:

PC-Kenntnisse

Mindestausstattung für den eigenen Unterricht:

PC-Raum

Controllersystem

Ergänzende Fortbildungen:

Dieses Teilmodul hat den Umfang von 2 Tagen. Die weiteren Module E616, E617 und E618 ergänzen diese Fortbildung.

zuständige Modulkoordinatoren:

Stuttgart und Schwäbisch Gmünd: Rolf Rahm | Karlsruhe und Mannheim: Artur Busch | RPF: Gero Albrecht | RPT: nn



ZSL

Zentrum für Schulqualität
und Lehrerbildung
Baden-Württemberg

Teilmodul E619 | IoT-Funkstandards anwenden: LoRaWan, WLAN, Bluetooth (BLE)

Beschreibung:

In diesem Modul werden mit der Arduino-Entwicklungsumgebung kleine Projekte zur Übertragung von Sensordaten via LoRaWAN, WLAN und Bluetooth/BLE realisiert. Hierzu werden die charakteristischen Eigenschaften der einzelnen Funkstandards thematisiert und Möglichkeiten für verschiedene Schülerprojekte aufgezeigt.



Arduino Uno mit Dragino-LoRa-Shield



ESP32 – freemotics-board für WLAN und Bluetooth

Zielgruppe:

Zielgruppe sind technische und wissenschaftliche KollegInnen unterschiedlicher Schularten, die aktuelle IoT-Projekte in ihrem Unterricht umsetzen wollen.

Ziele des Moduls:

- a) Nutzung der Arduino-Umgebung unter Verwendung eines LoRa-Shields
- b) Erfassen von Sensorwerten unter Verwendung von Bibliotheken
- c) Grundlegende Kenntnisse der LoRa-Übertragungstechnik anwenden
- d) Grundlagen des MQTT-Protokolls verstehen und zielführend einsetzen
- e) Bluetooth/BLE zur Sendung von Daten an den ESP32 anwenden
- f) Sensordaten mit Hilfe eines Dashboards am Smartphone visualisieren.
- g) ESP32 als WLAN-Client und als Accesspoint zur Datenerfassung einsetzen

Voraussetzungen:

Grundlagen in der Mikrocontrollerprogrammierung!

Mindestausstattung für den eigenen Unterricht:

PC-Raum, Passende Entwicklungsumgebung (Arduino-IDE)

Passendes LoRa-Shield und LoRa-Gateway

ESP32-Board

Weiterführende Fortbildungen:

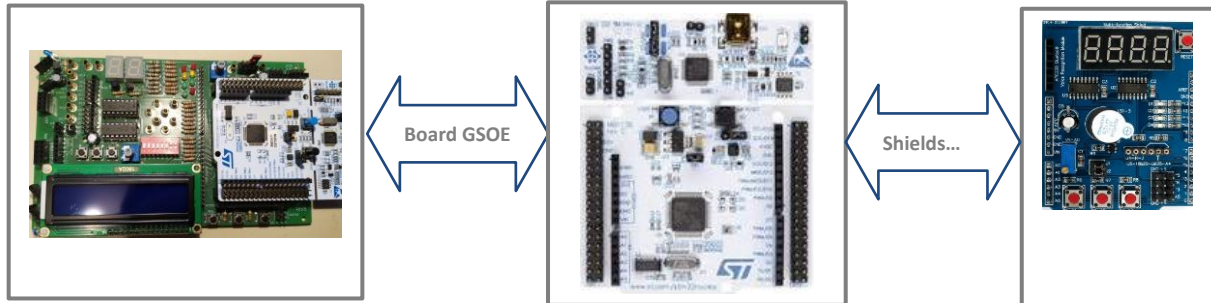
Dem Thema IoT widmen sich auch die Fortbildungen E616, E617 und E618.

zuständige Modulkoordinatoren:

Stuttgart und Schwäbisch Gmünd: Rolf Rahm | Karlsruhe und Mannheim: Artur Busch | RPF: Gero Albrecht | RPT: nn

**ZSL**Zentrum für Schulqualität
und Lehrerbildung
Baden-WürttembergModulfortbildungen >> **Elektrotechnik**

Teilmodul E621 | TG-INFT: STM32 Assembler



Evaluation-Board GSOE...

STM32 Nucleo => STM32 Cube-IDE

Arduino-Shields...

Beschreibung:

Gemäß der Bildungsplaneinheit BPE II des TG-Informationstechnik gibt es hier eine Einführung in die ARM-Cortex-Mikrocontrollerfamilie, speziell des STM32 Nucleo L152 RE sowie eine Einführung in die Assemblerprogrammierung des genannten Controllers. Hierzu wird die STM-Entwicklungsumgebung CUBE genutzt, um eine gemeinsame „einfache“ Einführung in die Assemblerprogrammierung zu erhalten.

Für Programmieranwendungen kommt das Evaluationsboard der GS-Östringen zum Einsatz bzw. diverse Arduino-Shields, die ebenso auf den STM32 Nucleo aufgesteckt werden können.

Dabei folgen wir einer didaktischen Vorgehensweise zur Einführung der Mikrocontrollerprogrammierung in der Eingangsklasse des TG-INFT.

Zielgruppe:

Zielgruppe sind Teilnehmer (techn. und wissenschaftliche KollegInnen) die im TG-Profilfach Informationstechnik bzw. Informatik die Mikrocontrollertechnik unterrichten.

Ziele des Moduls:

Einführung in die Entwicklungsumgebung STM32 CUBE
Grundlagen der Mikrocontrollerprogrammierung mit Assembler

Anwendungsprogrammierung mit dem Evaluation-Board bzw. Arduino-Shields

Didaktische Vorgehensweise bei der Einführung im Unterricht

Voraussetzungen:

PC-Kenntnisse

Grundkenntnisse Mikrocontroller

Mindestausstattung für die LFB:

PC-Raum mit Entwicklungssoftware

Hardware wie µC-Board, Evaluation-Board, Arduino-Shields

Weiterführende Fortbildungen:

Dieses Teilmodul hat den Umfang von 1 Tag. Das Modul E622 ergänzt diese Fortbildung.

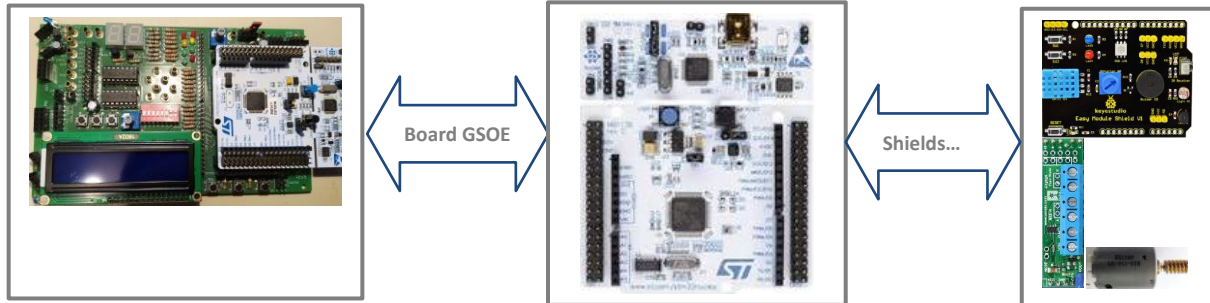
zuständige Modulkoordinatoren:

Stuttgart und Schwäbisch Gmünd: Rolf Rahm | Karlsruhe und Mannheim: Artur Busch | RPF: Gero Albrecht | RPT: nn

```
workspace_Lsg - STM32CubeIDE
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer
> 010_LED_an
> 015_LED_blink
> 020_MFS_LED_blink
> 025_MFS_LEDs_blink
> 030_MFS_SOS_blink_UP
> 035_MFS_SOS_blink_UP_Schleife
> 040_MFS_SOS_blink_S1
> 050_MFS_SOS_blink_ADC
> 060_MFS_LED_KnightRider
> 070_MFS_LEDs_shift
> 100_MFS_7Seg_Zahl_Codetabelle
> 110_MFS_7Seg_S1_zaehl_Einer
> 115_MFS_7Seg_S1_zaehl_Einer_entprellt
> 120_MFS_7Seg_S1_zaehl_Einer_JR
> 130_MFS_7Seg_S1_zaehl_Multiplex
> 140_MFS_7Seg_ADC_Multiplex
> 145_MFS_7Seg_S1_S2_zaehl_Multiplex
mainasm main.c main.c
4.global mainasm
5mainasm:
6
7main:
8ldr R1,= GPIOA //
9ldr R0,=hadc
10
11schleife:
12ldr R0,[R1, IDR]
13tst R0, Bit1
14bne schleife
15ldr R0,=hadc
16bl HAL_ADC_Start
17bl ADC_Wert
18mov R7, #2
19MUL R6, R4, R7
20
```

**ZSL**Zentrum für Schulqualität
und Lehrerbildung
Baden-WürttembergModulfortbildungen >> **Elektrotechnik**

Teilmodul E622 | TG-INFT: STM32 mbed | Arduino IDE und UML



Evaluation-Board GSOE...

STM32 Nucleo => mbed ⇔ Arduino-IDE

Arduino-Shields...

Beschreibung:

Gemäß der Bildungsplaneinheiten der Jahrgangsstufen 1 des TG-Informationstechnik gibt es hier eine Einführung in die Hochsprache C/C++, speziell zur Programmierung des STM32 Nucleo L152 RE.

Hierzu wird die Online Entwicklungsumgebung mbed bzw. Arduino-IDE genutzt, um eine gemeinsame Einführung in die Mikrocontrollerprogrammierung der ARM-Cortex-Familie zu erhalten.

Für Programmieranwendungen kommt das Evaluationsboard der GS-Östringen zum Einsatz bzw. diverse Arduino-Shields, die ebenso auf den STM32 Nucleo aufgesteckt werden können inkl. I²C LC-Display.

Dabei folgen wir einer didaktischen Vorgehensweise zur Einführung der Mikrocontrollerprogrammierung in der Jahrgangsstufe des TG-INFT.

Zielgruppe:

Zielgruppe sind Teilnehmer (techn. und wissenschaftliche KollegInnen) die im TG-Profilfach Informationstechnik bzw. Informatik die Mikrocontrollertechnik unterrichten.

Ziele des Moduls:

Einführung in die Entwicklungsumgebungen

mbed Online ⇔ Offline bzw. Arduino-IDE

Grundlagen der Mikrocontrollerprogrammierung mit C/C++

Anwendungsprogrammierung mit dem Evaluation-Board bzw.

Arduino-Shields inkl. I²C-LC-Display

Didaktische Vorgehensweise bei der Einführung im Unterricht

Voraussetzungen:

PC-Kenntnisse

Grundkenntnisse Mikrocontroller

Grundkenntnisse Hochsprache C/C++

Mindestausstattung für die LFB:

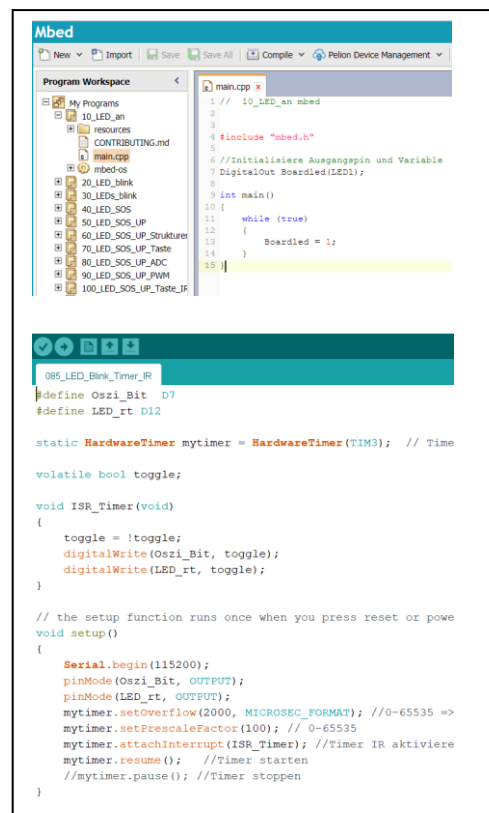
PC-Raum mit Entwicklungssoftware

Hardware wie µC-Board, Evaluation-Board, Arduino-Shields,

I²C-LC-Display

Weiterführende Fortbildungen:

Dieses Teilmodul hat den Umfang von 2 Tage. Das Modul E623 ergänzt diese Fortbildung.

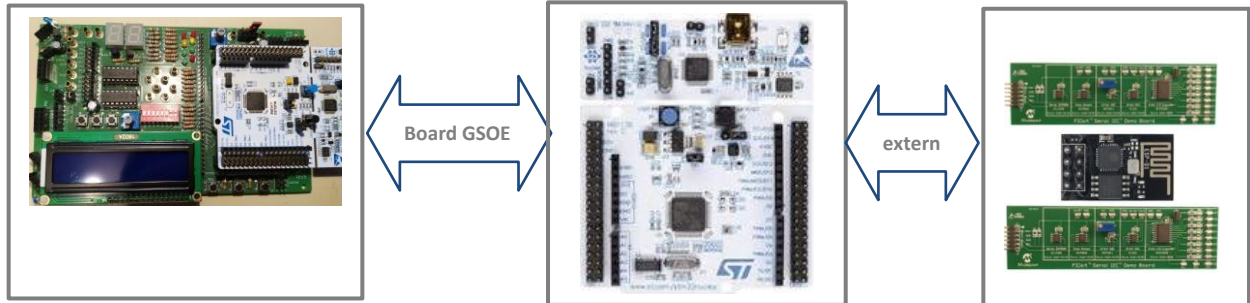


zuständige Modulkoordinatoren:

Stuttgart und Schwäbisch Gmünd: Rolf Rahm | Karlsruhe und Mannheim: Artur Busch | RPF: Gero Albrecht | RPT: nn

**ZSL**Zentrum für Schulqualität
und Lehrerbildung
Baden-WürttembergModulfortbildungen >> **Elektrotechnik**

Teilmodul E623 | TG-INFT: STM32, Schnittstellen und Bussysteme I2C, SPI, UART



Evaluation-Board GSOE...

STM32 Nucleo => mbed bzw. Arduino-IDE

I²C, SPI, UART...

Beschreibung:

Gemäß der Bildungsplaneinheiten der Jahrgangsstufen 2 des TG-Informationstechnik gibt es hier eine Einführung in Schnittstellenprogrammierung mit UART, SPI und I2C speziell zur Programmierung des STM32 Nucleo L152 RE. Hierzu wird die Online Entwicklungsumgebung mbed genutzt, um eine gemeinsame Einführung in die Mikrocontrollerprogrammierung der ARM-Cortex-Familie zu erhalten. Diese IDE gibt kann auch als Offline-Tool verwendet werden.

Für Programmieranwendungen kommt das Evaluationsboard der GS-Östringen zum Einsatz bzw. diverse Busshields, die ebenso auf den STM32 Nucleo integriert werden können inkl. I²C LC-Display.

Dabei folgen wir einer didaktischen Vorgehensweise zur Einführung der Mikrocontrollerprogrammierung in der Jahrgangsstufe des TG-INFT.

Zielgruppe:

Zielgruppe sind Teilnehmer (techn. und wissenschaftliche KollegInnen) die im TG-Profilfach Informationstechnik bzw. Informatik die Mikrocontrollertechnik unterrichten.

Ziele des Moduls:

Einführung in die Entwicklungsumgebungen

mbed Online ⇌ Offline bzw. Arduino-IDE

Grundlagen der Mikrocontrollerprogrammierung mit C/C++

Anwendungsprogrammierung mit dem Evaluation-Board bzw.

Shields inkl. I²C-LC-Display

Didaktische Vorgehensweise bei der Einführung im Unterricht

Voraussetzungen:

PC-Kenntnisse

Grundkenntnisse Mikrocontroller

Grundkenntnisse Hochsprache C/C++

Mindestausstattung für die LFB:

PC-Raum mit Entwicklungssoftware

Hardware wie µC-Board, Evaluation-Board, Shields, I²C-LC-Display und weitere...

```
1 // 10_LED_an mbed
2
3
4 #include "mbed.h"
5
6 //Initialisiere Ausgangspin und Variable
7 DigitalOut BoardLed(LED1);
8
9 int main()
10 {
11     while (true)
12     {
13         BoardLed = !BoardLed;
14     }
15 }

// 10_LED_Blink_Timer_IR
#define Oszil_Bit D7
#define LED_rft D12

static HardwareTimer mytimer = HardwareTimer(TIM3); // Time

volatile bool toggle;

void ISR_Timer(void)
{
    toggle = !toggle;
    digitalWrite(Oszil_Bit, toggle);
    digitalWrite(LED_rft, toggle);
}

// the setup function runs once when you press reset or power
void setup()
{
    Serial.begin(115200);
    pinMode(Oszil_Bit, OUTPUT);
    pinMode(LED_rft, OUTPUT);
    mytimer.setOverflow(2000, MICROSEC_FORMAT); //0-65535 =>
    mytimer.setPrescaleFactor(100); // 0-65535
    mytimer.attachInterrupt(ISR_Timer); //Timer IR aktivieren
    mytimer.resume(); //Timer starten
    //mytimer.pause(); //Timer stoppen
}
```

zuständige Modulkoordinatoren:

Stuttgart und Schwäbisch Gmünd: Rolf Rahm | Karlsruhe und Mannheim: Artur Busch | RPF: Gero Albrecht | RPT: nn

**ZSL**

Zentrum für Schulqualität
und Lehrerbildung
Baden-Württemberg

Teilmodul E6xx | IoT: Kleinprojekte

Beschreibung:

Dieses Modul soll Möglichkeiten aufzeigen, wie ARM und ESP32 Mikrocontroller für Kleinprojekte genutzt werden können. Im Fokus steht dabei Funktion und Ansteuerung unterschiedlichster Motoren, die in mobilen Robotern genutzt werden. Die mobilen Roboter werden per Bluetooth bzw. Wifi mittels Smartphone-APP ferngesteuert. Zum Einsatz kommt dabei das ARM-MC-System NUCLEO-F103RB sowie ein ESP32-basiertes MC-System in Kombination mit einem Motortreiber-Shield. Die Programme für ARM basierten MC- werden auf Basis von MBed entwickelt.

Zielgruppe:

Zielgruppe sind technische und wissenschaftliche KollegInnen von unterschiedlichen Schularten, die Mikrocontrollertechnik unterrichten.

Ziele des Moduls:

Ansteuerung von unterschiedlichen Motoren

- DC Motor mit H-Brücke und PWM
- Schrittmotoren mit H-Brücke
- Servo Motor mit PWM
- Funktionsweise von Brushless DC Motoren
- Electronic Speed Control (ESC) steuert BLDC mobile Roboter (u.a. Hovercraft)
- Fahrt mit Hinderniserkennung
- Leitlinienfahrt

Fernsteuern der mobilen Roboter mittels Smartphone-APP

- Bluetooth
- Wifi

Voraussetzungen:

Grundwissen in Elektrotechnik

Grundwissen in der Programmiersprache C

Grundwissen zu Funktionsweise von Mikrocontrollern

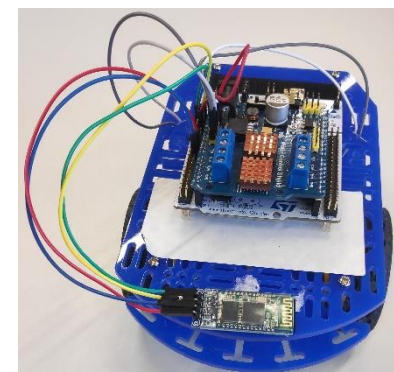
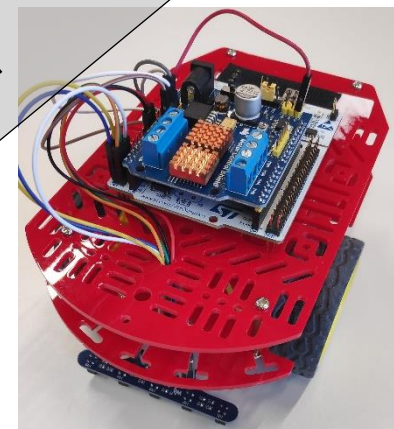
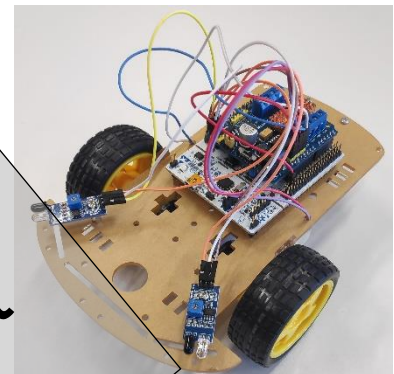
Mindestausstattung für den eigenen Unterricht:

PC-Raum mit Internetzugang

Passende Entwicklungsumgebung (HW und SW)

Weiterführende Fortbildungen:

aktuell keine



zuständige Modulkoordinatoren:

Stuttgart und Schwäbisch Gmünd: Rolf Rahm | Karlsruhe und Mannheim: Artur Busch | RPF: Gero Albrecht | RPT: nn