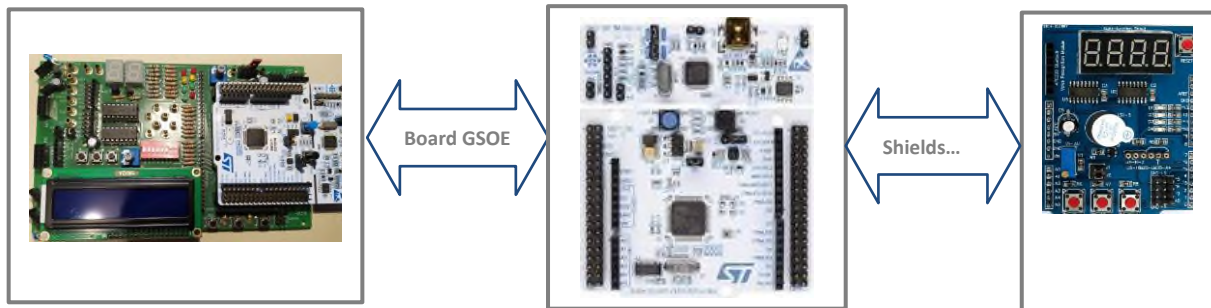


Präzisierung zu Modulbereich Mikrocontrollertechnik [E62]

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

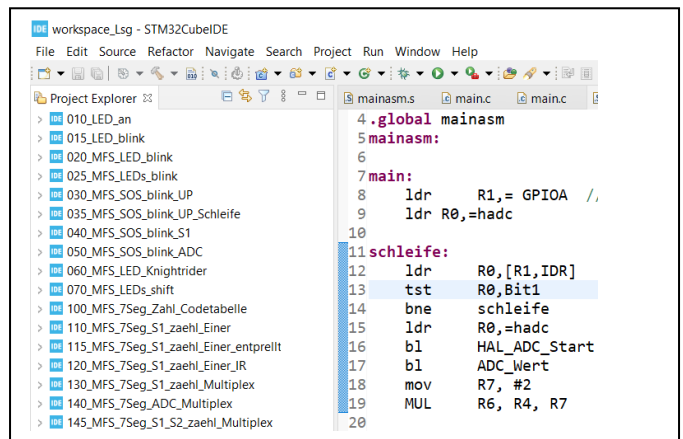
Hardware wie µC-Board, Evaluation-Board, Arduino-Shields

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



```

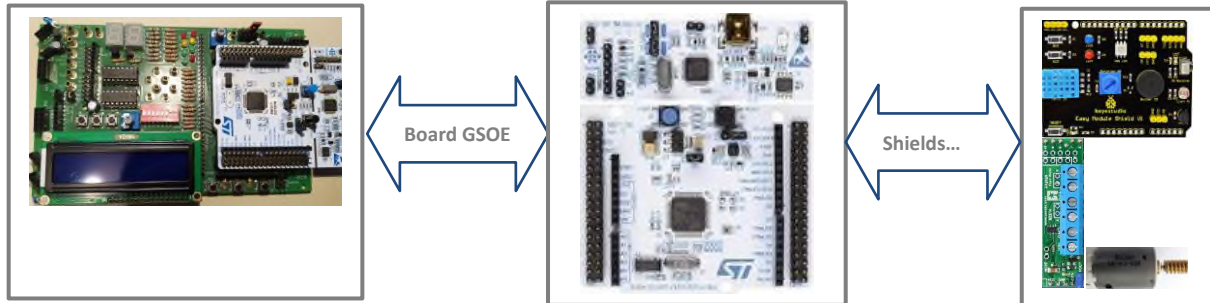
104 workspace_Lsg - STM32CubeIDE
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer
> 010_LED_an
> 015_LED_blink
> 020_MFS_LED_blink
> 025_MFS_LEDs_blink
> 030_MFS_SOS_blink_UP
> 035_MFS_SOS_blink_UP_Schleife
> 040_MFS_SOS_blink_S1
> 050_MFS_SOS_blink_ADC
> 060_MFS_LED_KnightRider
> 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_IR
> 130_MFS_7Seg_S1_zaehl_Multiplex
> 140_MFS_7Seg_ADC_Multiplex
> 145_MFS_7Seg_S1_S2_zaehl_Multiplex

mainasm.s | 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
16b1 HAL_ADC_Start
17b1 ADC_Wert
18mov R7, #2
19MUL R6, R4, R7
20

```



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



Evaluation-Board GSOE...

STM32 Nucleo => mbed ⇔ Arduino-IDE

Arduino-Shields...

Beschreibung:

Gemäß der Bildungspläneinheiten 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.

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

08S_LED_Blink_Timer_IR
#define Oszi_Bit D7
#define LED_rt D12

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

volatile bool toggle;

void ISR_Timer(void)
{
    toggle = !toggle;
    digitalWrite(Oszi_Bit, toggle);
    digitalWrite(LED_rt, toggle);
}

// the setup function runs once when you press reset or power
void setup()
{
    Serial.begin(115200);
    pinMode(Oszi_Bit, OUTPUT);
    pinMode(LED_rt, OUTPUT);
    mytimer.setOverflow(2000, MICROSEC_FORMAT); //0-65535 =>
    mytimer.setPrescaleFactor(100); // 0-65535
    mytimer.attachInterrupt(ISR_Timer); //Timer IR aktiviere
    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

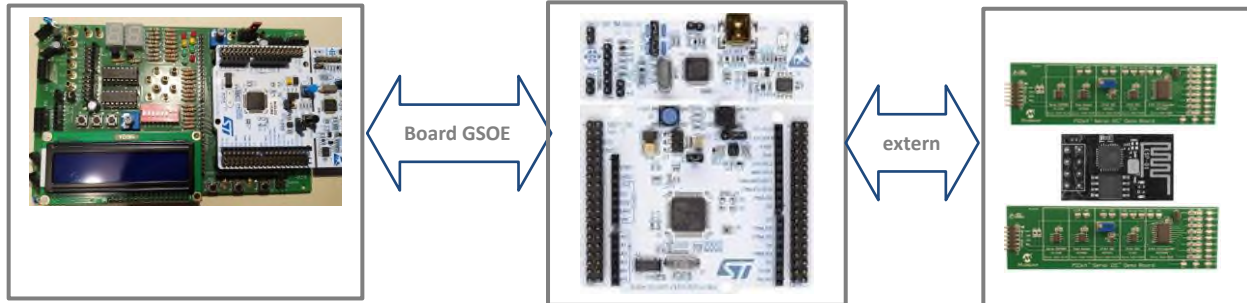


ZSL

Zentrum für Schulqualität
und Lehrerbildung
Baden-Württemberg

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



```
#define LED_Blink_Timer_IR
#define Oszil_Bit D7
#define LED_rt D12

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

volatile bool toggle;

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

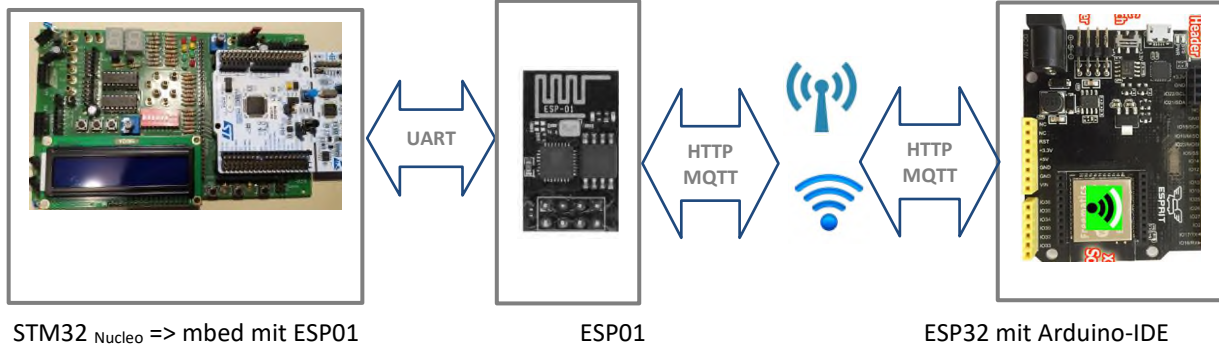
// the setup function runs once when you press reset or power
void setup()
{
    Serial.begin(115200);
    pinMode(Oszil_Bit, OUTPUT);
    pinMode(LED_rt, OUTPUT);
    mytimer.setOverflow(2000, MICROSEC_FORMAT); //0-65535 =>
    mytimer.setPrescaleFactor(100); // 0-65535
    mytimer.attachInterrupt(ISR_Timer); //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



Teilmodul E624 | TG-INFT: Internet der Dinge [IoT] mit ESP-Varianten



Beschreibung:

Gemäß der Bildungsplaneinheiten der Jahrgangsstufen 2 des TG-Informationstechnik gibt es hier eine Einführung in Internet of Things (IoT) mit ESP-Derivaten; je nach Hardware an der Schule.

Hierzu wird die Entwicklungsumgebung mbed genutzt, um mit dem GSOE-Board und dem STM32 Nucleo über AT-Kommandos (UART) den ESP01 für eine Internetverbindung anzusteuern.

Alternativ kann der ESP32 inkl. WiFi mit der Arduino-IDE genutzt werden, der im Uno-Format das Nutzen diverser Shields möglich macht, um individuelle IoT-Projekte umzusetzen.

Dabei folgen wir einer didaktischen Vorgehensweise zur Einführung der Mikrocontrollerprogrammierung aus der Jahrgangsstufe 1 des TG-INFT.

Als IoT-Protokolle werden HTTP und MQTT genutzt, welche durch praktische Programmierbeispiele auf den jeweiligen Hardwarevarianten angewendet werden.

Zielgruppe:

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

Ziele des Moduls:

- Einführung in IoT
- HTTP-Anwendungen
- MQTT-Anwendungen
- Didaktische Vorgehensweise bei der Einführung im Unterricht

Voraussetzungen:

- PC-Kenntnisse
- Grundkenntnisse Mikrocontroller
- Grundkenntnisse Hochsprache C/C++
- Grundkenntnisse Netzwerktechnik

Mindestausstattung für die LFB:

- PC-Raum mit Entwicklungssoftware
- Hardware wie µC-Board, WiFi-Zugang, Shields...

```
main.cpp
1 // 10_LED_mbed
2
3 #include "mbed.h"
4
5 //Initialisiere Ausgangspinn und Variable
6 DigitalOut Boardled(LED1);
7
8 int main()
9 {
10 {
11     while (true)
12     {
13         Boardled = !;
14     }
15 }
16 }
```

```
085_LED_Blink_Timer_IR
#define Osz1_Bit D7
#define LED_rt D12

static HardwareTimer mytimer = HardwareTimer(TIM5); // Time

volatile bool toggle;

void ISR_Timer(void)
{
    toggle = !toggle;
    digitalWrite(Osz1_Bit, toggle);
    digitalWrite(LED_rt, toggle);
}

// the setup function runs once when you press reset or power
void setup()
{
    Serial.begin(115200);
    pinMode(Osz1_Bit, OUTPUT);
    pinMode(LED_rt, OUTPUT);
    mytimer.setOverflow(2000, MICROSEC_FORMAT); //0-65535 =>
    mytimer.setPrescaleFactor(100); // 0-65535
    mytimer.attachInterrupt(ISR_Timer); //Timer IR aktiviere
    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:

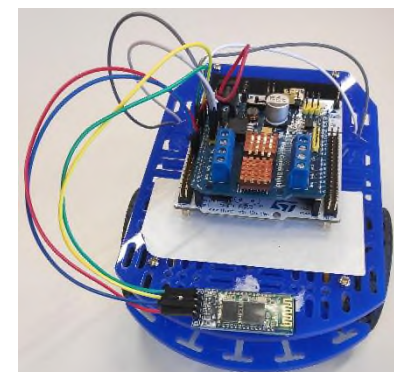
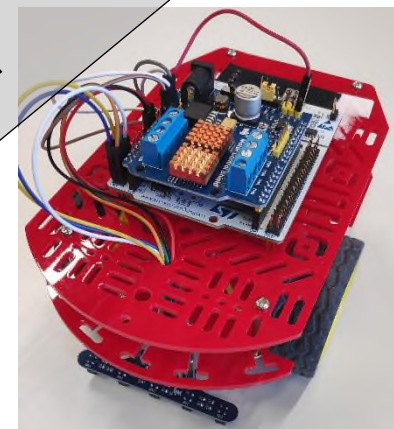
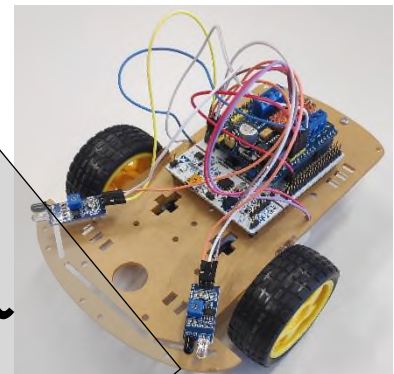
- Grundwissen in Elektrotechnik
- Grundwissen in der Programmiersprache C
- Grundwissen zu Funktionsweise von Mikrocontrollern

Mindestausstattung für den eigenen Unterricht:

- PC-Raum mit Internetzugang
- Passende Entwicklungsumgebung (HW und SW)

Weiterführende Fortbildungen:

aktuell keine



Wird im Schuljahr
2023/2024 nicht
angeboten!

zuständige Modulkoordinatoren:

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