

Übung 1

Leguan Entwicklungsumgebung

Aufgabe 1.1 Installation STM32CubeIDE

Im Unterricht des Fachbereichs Elektrotechnik und Informationstechnologie verwenden wir für die Programmierung von Embedded Systems die Entwicklungsumgebung STM32CubeIDE von ST Microelectronics. Dies ist eine professionelle Entwicklungsumgebung (IDE) für Microcontroller der Firma ST Microelectronics und darf auch für kommerzielle Projekte ohne Lizenzkosten verwendet werden.

Installieren Sie STM32CubeIDE auf Ihrem Laptop. Die Download-Webseite finden Sie hier:

<https://www.st.com/en/development-tools/stm32cubeide.html>

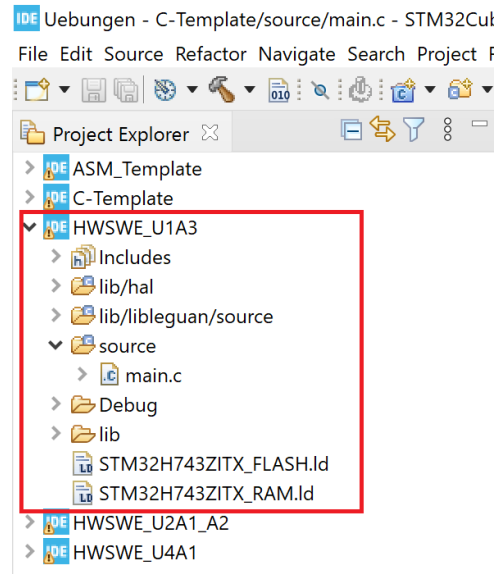
Wählen Sie am Anfang der Seite "Get Software" oder scrollen Sie auf der Seite nach unten, bis Sie beim Thema "Get Software" sind. Wählen Sie anschliessend die korrekte Version für Ihr Zielsystem, z. Bsp. STM32CubeIDE-Win falls Sie ein Windows-OS haben. Mit "Get Latest" erhalten Sie die aktuellste Version. Diese können Sie später auch aus STM32CubeIDE aktualisieren, wenn neue Versionen erhältlich sind. Anschliessend müssen Sie die Lizenz-Bestimmungen akzeptieren. Nun müssen Sie noch Name und e-Mail angeben. Falls Sie über neuere Version informiert werden möchten, können Sie dies per Checkbox angeben. Default ist, dass Sie keine Infos erhalten. Sie erhalten nun eine e-Mail zur Validierung Ihrer Mail-Adresse. Danach können Sie die Software herunterladen und diese installieren.

Aufgabe 1.2 Starten STM32CubeIDE und Laden C-Template

In diesem Unterrichtsmodul werden Sie diverse Übungen programmieren, wobei jede Übung aus Sicht von STM32CubeIDE ein eigenes Projekt ist. Um die Übersicht zu behalten, ist es sinnvoll, wenn alle diese Übungen in einem gemeinsamen Workspace abgelegt werden. Erzeugen Sie dazu auf Ihrem Laptop einen Ordner für alle Übungen dieses Unterrichtsmoduls und vergeben Sie einen sinnvollen Namen, z.Bsp, "BTE5053" oder "HWSWE". Legen Sie diesen Ordner unbedingt lokal ist, ein Netzlaufwerk ist wegen des langsamen Zugriffs nicht ideal. Wenn Sie STM32CubeIDE starten, wird der zu verwendende Workspace abgefragt. Geben Sie in dieser Dialogbox den Pfad zum Ordner an, den Sie gerade erzeugt haben. Wenn Sie STM32CubeIDE in anderen Unterrichtsmodulen wieder brauchen, können Sie für diese je einen eigenen Workspace erstellen.

Für die Übungen auf dem Leguan-Board gibt es diverse Templates, welche Ihnen die Programmierung erleichtern. Für diese erste Aufgabe und viele weitere Aufgaben in diesem Unterrichtsmodul, welche in der Programmiersprache C geschrieben sind, können Sie das C-Template verwenden. Dieses Template finden Sie auf Moodle für diesen Kurs unter "Weiterführende Unterlagen" (ganz am Schluss der Seite) und "Unterlagen Leguan". Zusätzlich finden Sie hier auch eine Anleitung, wie Sie das Template in die Entwicklungsumgebung laden können. Gehen Sie diese Anleitung Schritt für Schritt durch und benennen Sie das Projekt in HWSWE_U1A3 um. Dieses Vorgehen können Sie für alle folgenden C-Übungen ebenfalls anwenden.

In STM32CubeIDE wird nun ein neues Projekt erstellt, im aktuellen Fall HWSWE_U1A3. Im Fenster links (Project Explorer) finden Sie diverse Ordner für die Libraries und im Ordner "source" die Datei "main.c".



Das Hauptprogramm in der Datei “main.c” enthält folgenden Code:

```

/**
*****
* @file      main.c
* @brief     BTE 5053 HWSWE, C Template
*
* This is a template for C-Projects running on the Leguan board
* and using STM32CubeIDE.
*
* @version   1.0
* @date      2022-01-07
* @author    ssn1, WBR1
*
* @remark    Last Modification
*            \li wbr1 2022-01-07 created for BTE 5053
*****
*/
#include <leguan.h>
#include <cube.h>

/**
* @brief      main function
*
* This main function initializes the Leguan board. There is no additional
* functionality in the main loop.
*
* @param      none
* @return     always 0
*/
int main(void)
{
    /* Initialize Hardware */
    CUBEMX_Init();
    /* Initialize Leguan board */
    LEGUAN_Init();

    /* Set logging output destination to be the LCD */

```

```
LOG_SetDestination(LCD_Stream);  
  
/* Main loop */  
for (;;) {  
  
}  
  
return (0);  
}
```

Studieren Sie diesen Code. Passen Sie für jedes neue Projekt den Dateihheader an (Kurzbeschreibung des Projektes respektive der Übung) und beginnen Sie anschliessend mit der Programmierung.

Aufgabe 1.3 Das erste Projekt auf Leguan

Ziel dieser ersten Übung ist es, dass Sie für das Leguan-Board einige Zeilen Code programmieren können, zu Testzwecken Ausgaben auf dem Display und/oder auf der seriellen Schnittstelle (UART) machen können, und dass Sie den Code im Debugger ausführen sowie Variablen überwachen können.

Aufgaben:

- Die Library von Leguan unterstützt die Ausgabe von Log-Messages. Dies ist sehr praktisch um verschiedene Zustände und Informationen des Boards auszugeben. Die Ausgabe kann wahlweise auf dem Display oder über die serielle Schnittstelle erfolgen:

```
LOG_SetDestination(LCD_Stream); // Ausgabe Log-Infos auf dem Display  
LOG_SetDestination(SERIAL_Stream); // Ausgabe Log-Infos ueber die UART
```

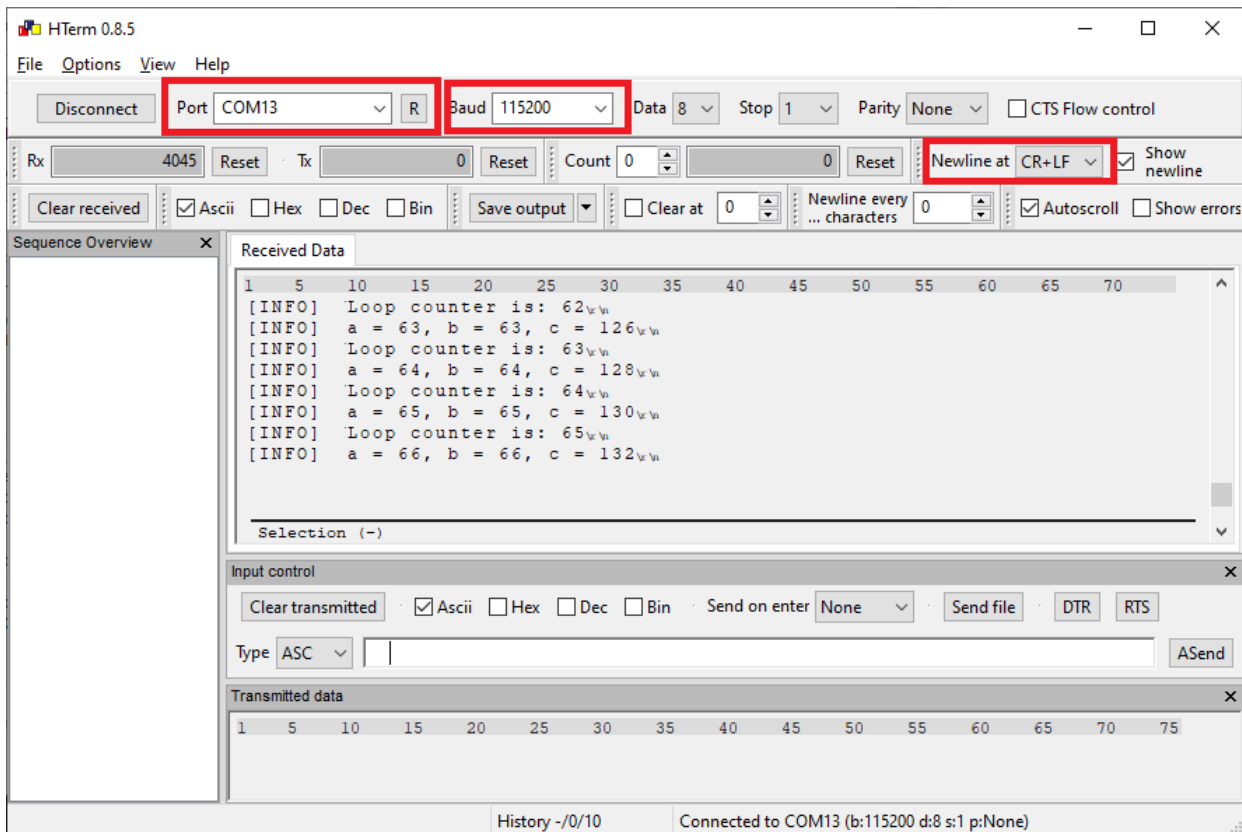
Informationen zu den Library-Funktionen finden Sie auf der Leguan-Webseite: <https://leguan.ti.bfh.ch/> Für weitere Angaben zu den Log-Funktionen wählen Sie auf der Leguan-Webseite im Menü libleguan → Common Features → Logger Module.

Mit Hilfe der Funktion

```
LOG_Info("Ausgabetext");
```

können Sie Texte ausgeben. Geben Sie den Text "Welcome to exercise U1A3" aus, wahlweise auf das Display oder die serielle Schnittstelle.

Um die Ausgabe der UART darzustellen, brauchen Sie noch ein Terminal-Programm. In der STM32CubeIDE ist defaultmässig kein solches integriert. Wenn Sie auf Ihrem Laptop noch kein Terminalprogramm installiert haben, dann installieren Sie am besten HTerm, Link: <https://www.heise.de/download/product/hterm-53283>. In HTerm müssen Sie mindestens den korrekten COM-Port wählen und die korrekte Baudrate einstellen.



Um den korrekten COM-Port herauszufinden, öffnen Sie unter Windows den Geräte-Manager (Device Manager) und gehen auf "Anschlüsse (COM & LPT)". Sobald Sie das Leguan-Board mit Ihrem Laptop verbunden haben, sollten Sie mindestens zwei Einträge finden: "JLink CDC UART Port (COMxx)" sowie "Seriellles USB-Gerät (COMyy)". In HTerm müssen Sie auf den COM-Port des seriellen USB-Gerätes verbinden.

- Definieren Sie in Ihrem Programm einige Variablen und führen Sie mit diesen einfache Berechnung durch. Inkrementieren Sie auch eine der Variablen in jedem Loop-Durchgang der Endlosschleife. Beispiel:

```
int main(void)
{
    uint32_t loop_count = 0;           /* counter, incremented each loop */
    uint32_t a = 0, b = 5, c;         /* variables to do some calculation */

    /* Initialize Hardware */
    CUBEMX_Init();
    /* Initialize Leguan board */
    LEGUAN_Init();

    ...

    /* Main loop */
    for (;;) {
        ...
        /* calculate something very important, log actual values */
        loop_count++;                /* increment counter each loop */
        a++;
        b = loop_count;
        c = a + b;
        ...
    }
}
```

```

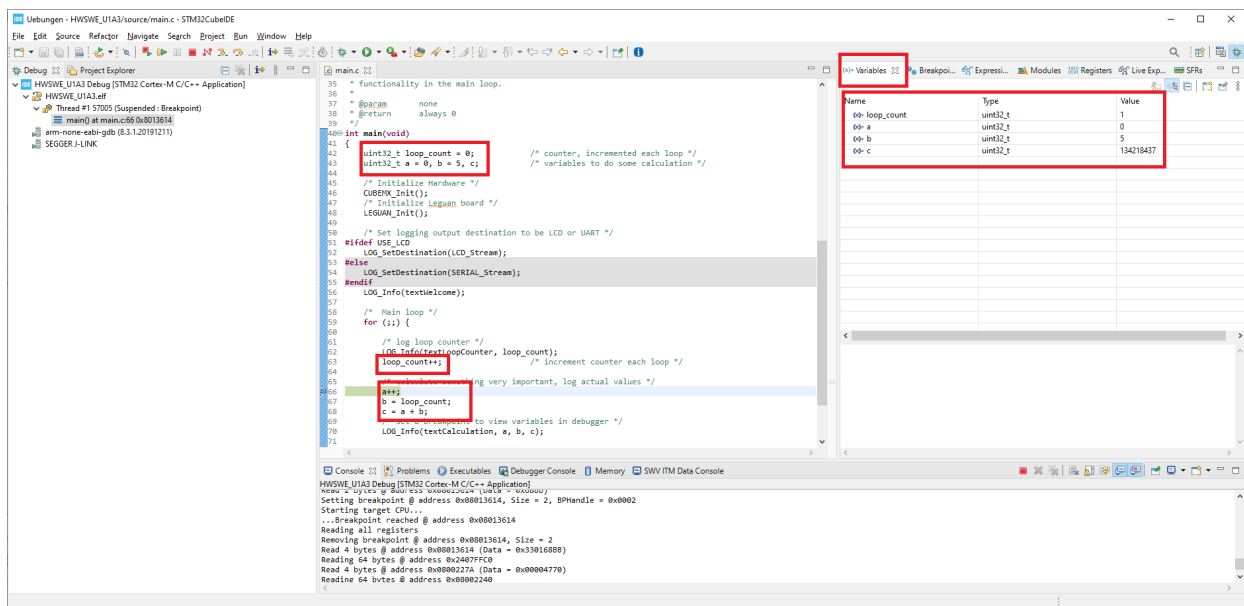
    /* do some dealy (in ms) and start again */
    CORE_Delays(1000);
}

return (0);
}

```

Geben Sie diese Variablen im Log aus.

- Starten Sie nun die Applikation erneut im Debugger. Sie können schrittweise durch den Code "step-ten" oder mit Breakpoints arbeiten. Zeigen Sie die aktuellen Werte der Variablen im Debugger an. Um die Werte jeder Variablen zu prüfen, können Sie entweder mit der Maus im Code auf die Variable zeigen, oder Sie wählen das Tab "Variables" rechts oben in STM32CubeIDE:



Beachten Sie, dass die Werte der Variablen nur aktualisiert werden, wenn Sie durch den Code "steppen" oder wenn Sie einen Breakpoint erreichen.