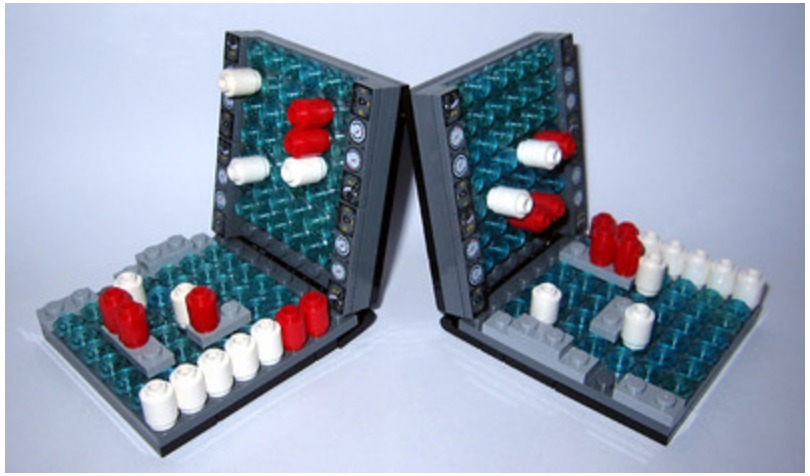
# Lab 3 - Treffer versenkt



# Infos

- wird über UART gespielt
- Eure KI gegen die anderer Studenten
- großes Abschlussturnier mit Siegerehrung

## Wie wird gespielt

- 2 STM32 Boards werden via UART verbunden (RX,TX,GND)
- Austausch von Nachrichten ("Protokoll")
- auf einem Board wird der Startbutton gedrückt
- beide Boards spielen interaktiv eine Runde gegeneinander
- Gewinn / Niederlage / Protokollfehler werden via LED angezeigt
- Bonusidee: grafisches Display

## Regeln

- https://de.wikipedia.org/wiki/Schiffe_versenken
- 10x10 Raster bestehend aus
  - 10 Reihen, nummeriert von 0 bis 9
  - 10 Spalten, nummeriert von 0 bis 9
- Schiffe: 1x5, 2x4, 3x3 und 4x2

## Protokoll

1. Startnachricht von Spieler 1 (S1) an S2
2. S2 generiert Spielfeld und übermittelt Spielfeldchecksummennachricht
3. S1 generiert Spielfeld und übermittelt Spielfeldchecksummennachricht
4. S2 schickt Startnachricht an S1
5. Gameloop (maximum 100 Iterationen)
    i. S1 übermittelt Schussnachricht
    ii. S2 übermittelt Schussbericht
    iii. prüfen auf Sieg / Niederlage
    iv. S2 übermittelt Schussnachricht
    v. S1 übermittelt Schussbericht
    vi. prüfen auf Sieg / Niederlage

## Protokoll cont'd

5. Anzeige Sieg / Niederlage / Unentschieden / Protokollfehler
6. S1 übermittelt Spielfeldnachrichten (Spalten 0-9)
7. S2 übermittelt Spielfeldnachrichten (Spalten 0-9)
8. Anzeige Schummler / Protokollfehler

**Jederzeit** ein Spieler kann Protokollfehler begehen und dadurch verlieren. Wenn der Spieler der eine Nachricht senden soll 1 Sekunde nichts sendet gilt das als Protokollfehler.

Die Spielfeldnachricht wird bei Protokollfehlern nicht übermittelt.

## Protokoll: Nachrichtenframe

- Nachrichten sind ASCII codiert
- sie enden mit eine newline Zeichen ('\n', 0x0A)

Die einzelnen Nachrichten und ihre Bedeutungen sind in den folgenden Folien erklärt, hierbei gilt die Konvention: 'X' steht für den einzelnen ASCII Character X, [0-9] steht für einen einzelnen ASCII Character auf der Menge '0', '1', '2', '3', '4', '5', '6', '7', '8' sowie '9', [w|t|v] steht für einen einzelnen ASCII Character aus der Menge 'w' oder 't' oder 'v'. '\n' steht für den einzelnen ASCII Character newline. '*' bedeutet das vorhergehende Zeichen wird wiederholt. In den Beispielen werden ASCII strings mit " umschlossen.

## Protokoll: Startnachricht

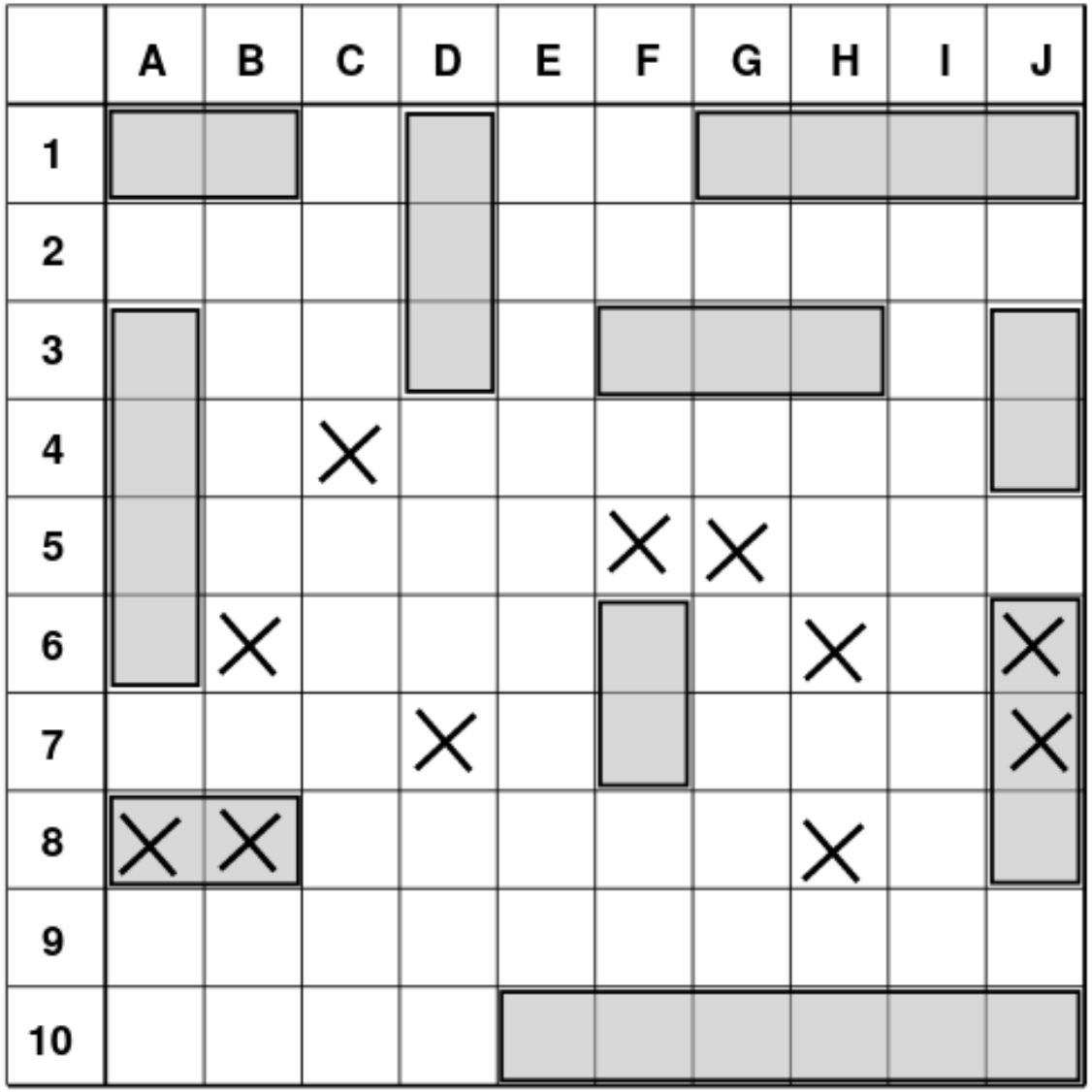
'S' 'T' 'A' 'R' 'T' [0-9]*'\n'

also der String "START<Ziffernfolge>\n" bestehend aus 6 ASCII Characters und einer Ziffernfolge. Die Ziffernfolge ist deine Matrikelnummer (bzw. die des Gegner). Wir vom Startspieler gesendet, ausgelöst durch den Druck auf den Startknopf des entsprechenden Boards bzw. von S2 wenn bereit beschossen zu werden.

## Protokoll: Spielfeldchecksummen-nachricht

'C' 'S' [0-9] [0-9] [0-9] [0-9] [0-9] [0-9] [0-9] [0-9] [0-9] [0-9] '\n'

diese Nachricht dient dem Gegner als Prüfsumme damit man nicht schwindelt, man verrät ihm damit wie vielen Felder einer jeden **Spalte** des eigenen Spielfeldes Schiffe enthalten. Die Spalten werden in Reihenfolge 0-9 übermittelt. Für dieses Beispiel lautet die Nachricht "CS6203143327\n"



## Protokoll: Schussnachricht

'B' 'O' 'O' 'M' [0-9] [0-9] '\n'

die erste Ziffer ist die Spalte, gefolgt von der Reihe des Feldes das beschossen wird, um im vorherigen Bild auf A 10 zu schießen lautet die Nachricht "BOOM09\n"

## Protokoll: Schussbericht

[W|T|V] '\n' wobei 'W' für wasser (also nichts getroffen) und 'T' für einen Treffer eines Schiffes steht.
Im Beispiel würden "BOOM97\n" mit "T\n", "BOOM09\n" mit "W\n" und BOOM00\n" mit "T\n" beantwortet werden.
Mehrfacher Beschuss des gleichen Schifffeldes wird stets mit 'T' beanwortet.

## Protokoll: Spielfeldnachricht

'S' 'F' [0-9] 'D' [0-5] [0-5] [0-5] [0-5] [0-5] [0-5] [0-5] [0-5] [0-5] [0-5] '\n'

Die Ziffer nach "SF" steht für die Spalte, es werden 10 Nachrichten für die Spalten 0-9 in exakt dieser Reihenfolge übertragen. die 10 folgenden Ziffern stehen für die in den jeweiligen Reihen 0-9 befindlichen Schiffe, '0' für Wasser, '5' für das 5er Schiff, '2' für ein Teil eines der 2er Schiffe. Im Beispiel ist z.B.: "SF6D4030000006\n" (wobei 6 in unseren Regeln kein gültiges Schiff ist) oder "SF9D4022033306\n"

## Protokoll: Schummeln

- Schummeln ist unfair
    - **=> lass dich nicht erwischen**
- Die Spielfeldchecksummen ist schwach, und man kann Schiffe auf Positionen "ausweichen" lassen die der Checksumme entsprechen, man muss das aber geschickt machen
- man überträgt sein eigenes Spielfeld nach dem Spiel, der Gegner kann das dann prüfen und einen als Schummler überführen, man hat das Spiel dann verloren

## Tipps

- es gibt mehrere UARTS auf dem System, verwende den am Nucleo Board mit USB verbundenen für Debugging und interaktive Spiele
- lass dein Programm die Spiele auf dem Debug-UART mitloggen, das gibt dir die Beweise für das Schummeln der Gegner
- Starte mit Vereinfachungen
  - statt zufälliger Spielfelder, ein hart-codiertes
  - statt komplexer Schussstrategien Spalte für Spalte und Reihe für Reihe
- implementiere darauf aufbauen komplexen Code
- verwende 'git'
- verwende den debugger

## Bewertung

- Programm kann als S1 die Startnachricht senden (**10 Punkte**)
- Programm erfüllt Protokoll als S2 bis Startnachricht (**10 Punkte**)
- implementiert Gameloop und kann vollständige Spiele spielen (**20 Punkte**)
- Programm erzeugt sinnvolle Debugmeldungen (**10 Punkte**)
- Programm erzeugt zufälliges Spielfeld (**15 Punkte**)
- Programm implementiert intelligente Schußstrategie (**15 Punkte**)
- ein weiteres tolles Feature (**20 Punkte**)
  - schummelt ohne erwischt zu werden
  - Strategie des Gegners erkennen und kontern
  - grafisches Display, Soundeffekte, ...

**Name:_____ Datum:_____ Punkte:_____ von 100**