

# Stackanalyse

```
main: //Hauptprogramm
    bl    startup
    ldr    R1,=GPIOB
    bl    unterprogramm
    ldrb    R0,[R1,IDR]        //Eingabe von GPIOB.IDR
    ldr    R1,=GPIOC
    bl    unterprogramm
    strb    R0,[R1,ODR]        //Ausgabe auf GPIOC.ODR
schleife:
    b      schleife
unterprogramm: //Wichtiges Unterprogramm verwendet unbedingt Register R1
    mov    R1,#0x110
    bx     lr
.end
```

Obiges Programm muss unbedingt die Zahl von GPIOB.IDR einlesen und auf die an GPIOC.ODR angeschlossenen LEDs ausgeben. Sonst passiert ein großes Unglück.

Aufgabe 1: Analysieren Sie ob das so funktionieren kann.

Aus unerfindlichen Gründen gelten folgende Randbedingungen:

- Das darf Hauptprogramm nicht verändert werden.
- Der Entwickler des Unterprogramms besteht auf die Verwendung von Register R1 und er ist der Sohn des Firmeninhabers

Aufgabe 2: Ein Entwickler schlägt folgende Lösung vor. Prüfen Sie die Lösung mit dem Debugger, füllen Sie hierzu die Tabelle unten aus.

Leicht verändertes Programm

```
main: //Hauptprogramm
    bl    startup
    ldr    R1,=GPIOB
    bl    unterprogramm
    ldrb    R0,[R1,IDR]        //Eingabe von GPIOB.IDR
    ldr    R1,=GPIOC
    bl    unterprogramm
    strb    R0,[R1,ODR]        //Ausgabe auf GPIOC.ODR
schleife:
    b      schleife
unterprogramm: //Wichtiges Unterprogramm verwendet unbedingt Register R1
    push    {R1}
    mov    R1,#0x110
    pop     {R1}
    bx     lr
.end
```

Programm	Weiter mit	=>	R13 SP	R1	0x20013ff4
ldr R1,=GPIOB	Step in F5	=>	0x20013ff8	-	
bl unterprogramm	Step in F5	=>			
push {R1}	Step in F5	=>			
mov R1,#0x110	Step in F5	=>			
pop {R1}	Step in F5	=>			
bx lr	Step in F5	=>			
ldrb R0,[R1,IDR]	Step in F5	=>			
ldr R1,=GPIOC	Step in F5	=>			
bl unterprogramm	Step in F5	=>			
push {R1}	Step in F5	=>			
mov R1,#0x110	Step in F5	=>			
pop {R1}	Step in F5	=>			
bx lr	Step in F5	=>			

Anleitung:

1. Programmieren Sie obiges Programm
2. Setzen Sie auf **bl startup** einen Breakpoint

Rechtsklick und  
Toggle Breakpoint

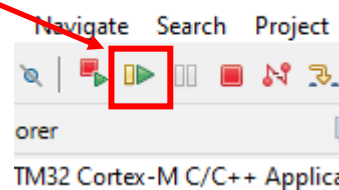
```

> mainasm:
6
7 main: //Hauptprogramm
8 bl startup
9 ldr R1,=GPIOB
10 bl unterprogramm
11 ldr R0,[R1,IDR] //Eingabe von GPIOB.IDR
12 ldr R1,=GPIOC
13 bl unterprogramm
14 ldr R0,[R1,ODR] //Ausgabe auf GPIOC.ODR
15 .end
16
17 unterprogramm: //Wichtiges Unterprogramm verwendet
18 push {R1}
19 mov R1,#0x110
20 pop {R1}
21 bx lr
22 .end
23

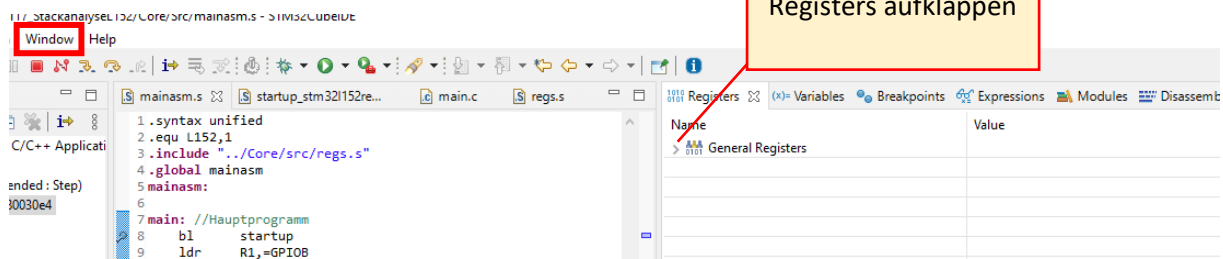
```

3. Starten Sie das Programm im Debugmode mit Debug As
4. Programm starten mit **resume**

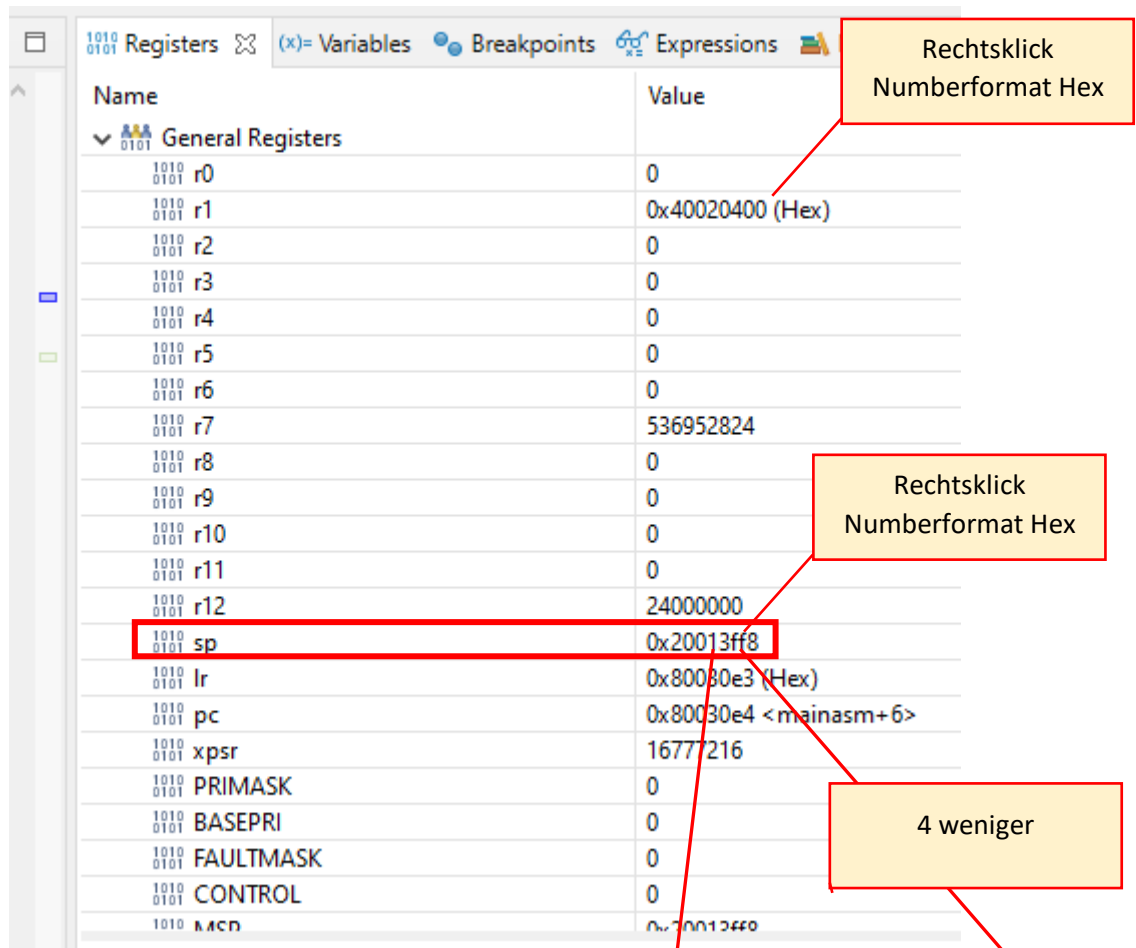
LoesungenT12 - T17\_Stackanaly:



## 5. Window -> Show View -> Registers



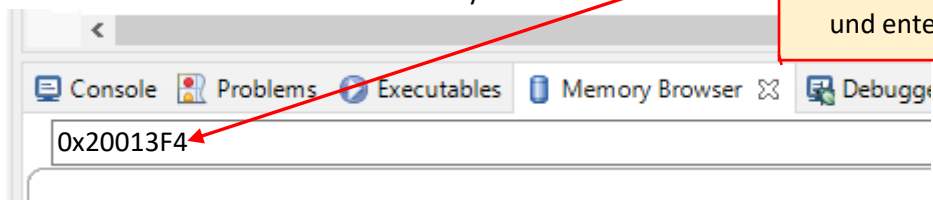
## 6. Auf Hexadezimale Anzeige umstellen

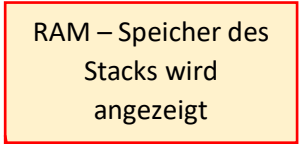


## 7. Wert des Stackpointers SP (R13) (hier 0x20013ff8) ablesen und in Tabelle eintragen

Programm	Weiter mit =>	R13 SP	R1	0x20013F4
<u>ldr</u> R1,=GPIOB	<u>Step in F5</u> =>	0x20013ff8	-	
bl unterprogramm	Step in F5 =>			

## 8. Window -> Show View -> Memory Browser





Aufgabe 5: Untersuchen Sie in gleicher Weise die Instruktionen: push {R0,R1,R2} und pop {R0,R1,R2}  
Hinweis: Dazu muss im Memory Browser eine Adresse ausgewählt werden, die um nochmals 8 niedriger ist.