

# RAM Array analysieren

1. Programmieren Sie nach und ergänzen Sie die Kommentare

**main:**

```
bl      startLCD
ldr     R0,=hadc      //AD-Wandler starten
bl      HAL_ADC_Start

ldr     R5,=Messwerte //Array Basisadresse
mov     R6,#0         //Array Offset
```

**schleife:** //Endlosschleife

```
mov     R0,#0
bl      LCD_i2c_cursorpos //cursor auf Position 0
(1.Zeile Anfang)
```

//AD-Wandler abfragen Ergebnis in R0

```
ldr     r0,=hadc
```

```
bl      HAL_ADC_GetValue //
```

```
str     R0,[R5,R6] //
```

```
add     R6,#4      //
```

```
cmp     R6,#80     //
```

```
bne     weiter     //
```

```
mov     R6,#0      //
```

**weiter:**

```
bl      LCD_i2c_dezaus //
```

```
ldr     R0,=leerstring
```

```
bl      LCD_i2c_textaus
```

```
ldr     R0,=1000    //
```

```
bl      HAL_Delay
```

```
b       schleife
```

**leerstring:**

```
.asciz " "
```

**.data** //

**Messwerte:**

```
.space 80,0 //
```

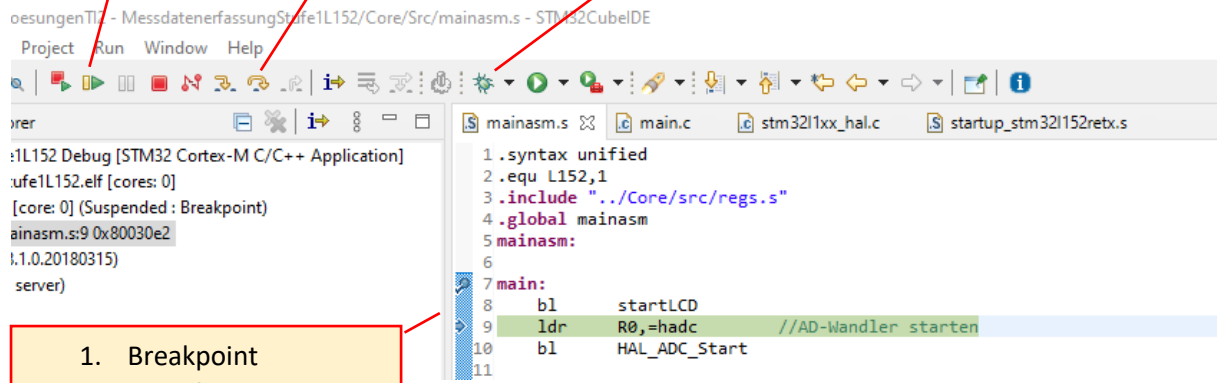
```
.end
```

## 2. Analysieren Sie das Programm im Einzelschrittmodus

3. Programm ausführen

4. Fortsetzen mit Step Over

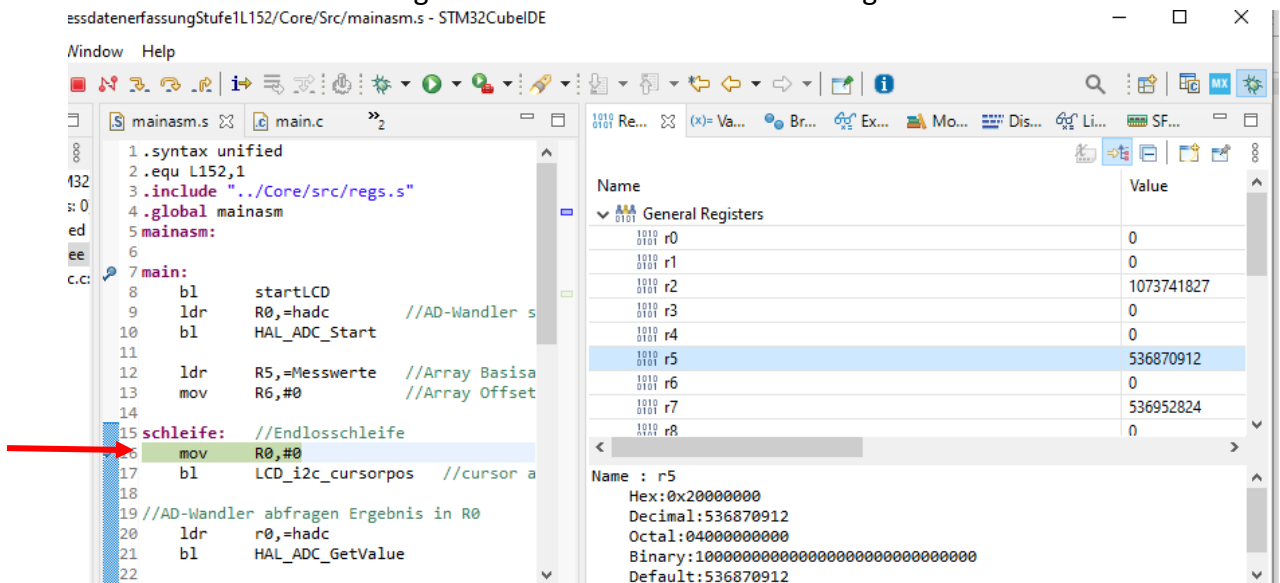
2. Debug starten



1. Breakpoint hinzufügen mit Rechtsklick und Toggle Breakpoint

Dazu muss:

1. Ein Breakpoint am Anfang des Programms gesetzt werden
2. Das Debugging gestartet werden
3. Das Programm bis zum Breakpoint ausgeführt werden
4. Anschließend wird das Programm im Einzelschrittbetrieb fortgesetzt.



Name	Value
General Registers	
r0	0
r1	0
r2	1073741827
r3	0
r4	0
r5	536870912
r6	0
r7	536952824
r8	0

Name : r5  
 Hex: 0x20000000  
 Decimal: 536870912  
 Octal: 0400000000  
 Binary: 10000000000000000000000000000000  
 Default: 536870912

5. Welchen Wert nehmen die Register R5 und R6 an, sobald Sie bis zur Programmzeile 16 gekommen sind? (Hexadezimal)  
 R5= 0x\_\_\_\_\_, R6= 0x\_\_\_\_\_
6. Wo im RAM-Speicher des Mikrocontrollers befindet sich das Array?  
 Adresse: 0x\_\_\_\_\_

7. Lassen Sie sich das RAM anzeigen:

7.1 Mit + einen neuen Bereich ab der Adresse von 6. hinzufügen

7.2 Bei New Renderings: Unsigned Integer einstellen

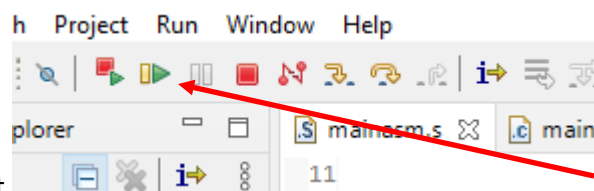
Address	0 - 3	4 - 7	8 - B	C - F
20000000	0	0	0	0
20000010	0	0	0	0
20000020	0	0	0	0
20000030	0	0	0	0
20000040	0	0	0	0
20000050	0	39	2	0
20000060	1296629760	2038266185	9	0
20000070	4194171209	1497976137	7	0
20000080	424234313	1566120221	6	0
20000090	757717469	151849257	1	0
200000A0	151849257	151849257	1	0

8. Weiter im Einzelschrittbetrieb bis Zeile 26

Welche Veränderung im RAM stellen Sie fest? Woher kommt der Wert?

Address	0 - 3	4 - 7	8 - B	C - F
20000000	2139	0	0	0
20000010	0	0	0	0
20000020	0	0	0	0
20000030	0	0	0	0
20000040	0	0	0	0
20000050	0	39	2828	0
20000060	2357723136	135006344	962157913	1296644413

9. Setzen Sie einen Breakpoint auf Zeile 26 (Rechtsklick auf 26 und „Toggle Breakpoint“)



10. Setzen Sie das Programm mit Resume fort und füllen Sie die folgende Tabelle aus. Verändern Sie dabei die Stellung des Potis.

R6	0-3	4-7	8-B	C-f

11. Beantworten Sie folgende Fragen

- a. Übersetzen Sie in Assembler:

Messwerte[R6]=R0;

\_\_\_\_\_

- b. Warum wird in Zeile 24 zu R6 die Zahl 4 hinzugezählt?

\_\_\_\_\_  
 \_\_\_\_\_

- c. Wozu dient die Compare-anweisung in Zeile 25?

\_\_\_\_\_  
 \_\_\_\_\_

- d. Wie wird ein Array mit 20 Integerwerten im RAM angelegt?

\_\_\_\_\_  
 \_\_\_\_\_  
 \_\_\_\_\_

- e. Wie viele Bytes umfasst ein Integerwert?

```

7 main:
8     bl    startLCD
9     ldr   R0,=hadc      //AD-Wandler starten
10    bl    HAL_ADC_Start
11
12    ldr   R5,=Messwerte //Array Basisadresse
13    mov   R6,#0         //Array Offset
14
15 schleife: //Endlosschleife
16    mov   R0,#0
17    bl    LCD_i2c_cursorpos //cursor auf Posit:
18
19 //AD-Wandler abfragen Ergebnis in R0
20    ldr   r0,=hadc
21    bl    HAL_ADC_GetValue
22
23    str   R0,[R5,R6]
24    add   R6,#4
25    cmp   R6,#80
26    bne   weiter
27    mov   R6,#0
28    mov   R7,#0         //Auslesezeige
29 weiter:
30

```