

Vorstellung des ARM-MC

Mit Evaluationsboard

Vorstellung des ARM-MC

**Warum benötigen wir
einen Mikrocontroller?**

Vorstellung des ARM-MC

Programmieren lernen?

Geht auch mit Visual Studio

Vorstellung des ARM-MC

Steuerungsaufgaben?

Geht auch mit EA-Karte am PC

Vorstellung des ARM-MC

IoT?

Geht auch mit ESP01 am USB des PC

Vorstellung des ARM-MC

Mobile Anwendungen?

Geht auch mit Raspberry Pi

Vorstellung des ARM-MC

Was sonst?

Spaß an der Technik vermitteln!

Funktionsweise der digitalen

Welt vermitteln!

Vorstellung des ARM-MC

- ARM Advanced Risc Machine
- STM32L152RET Mikrocontroller
- STM32 Nucleo
- Evaluationsboard
- Tools

Vorstellung des ARM-MC

- ARM Advanced Risc Machine
 - Britisches Unternehmen, seit 1990
 - entwickelt das Design von RISC-Prozessoren
 - Fertigung von den Lizenznehmern

AMD



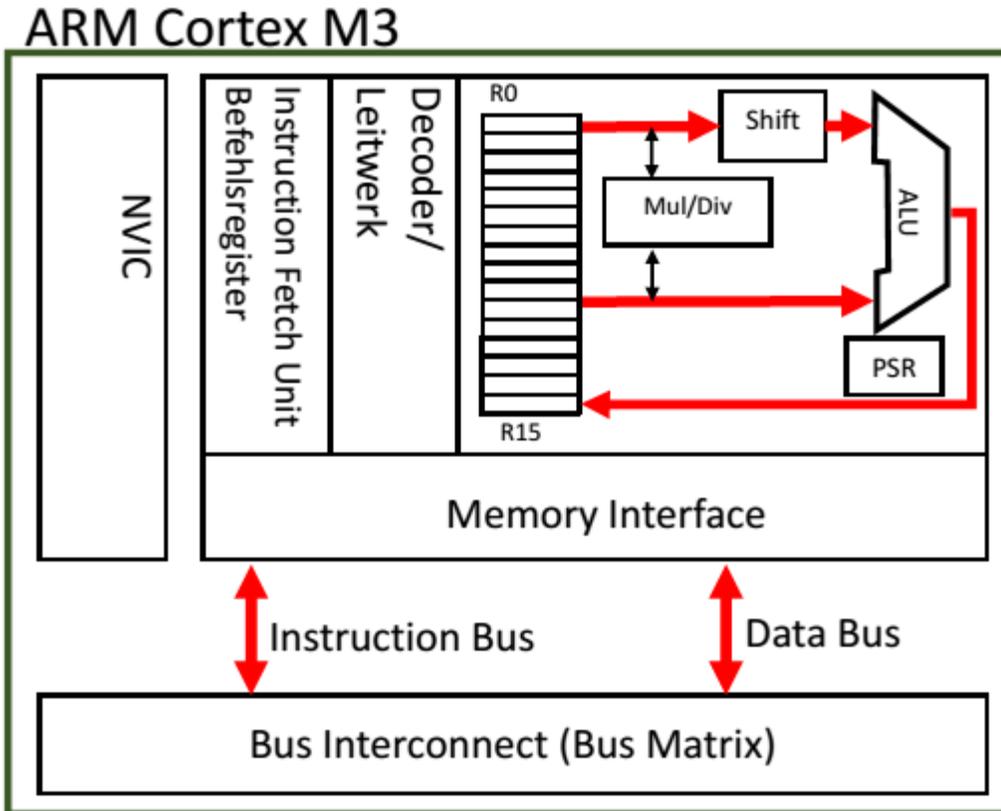
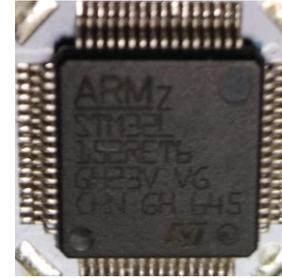
MICROCHIP



Atmel

Vorstellung des ARM-MC

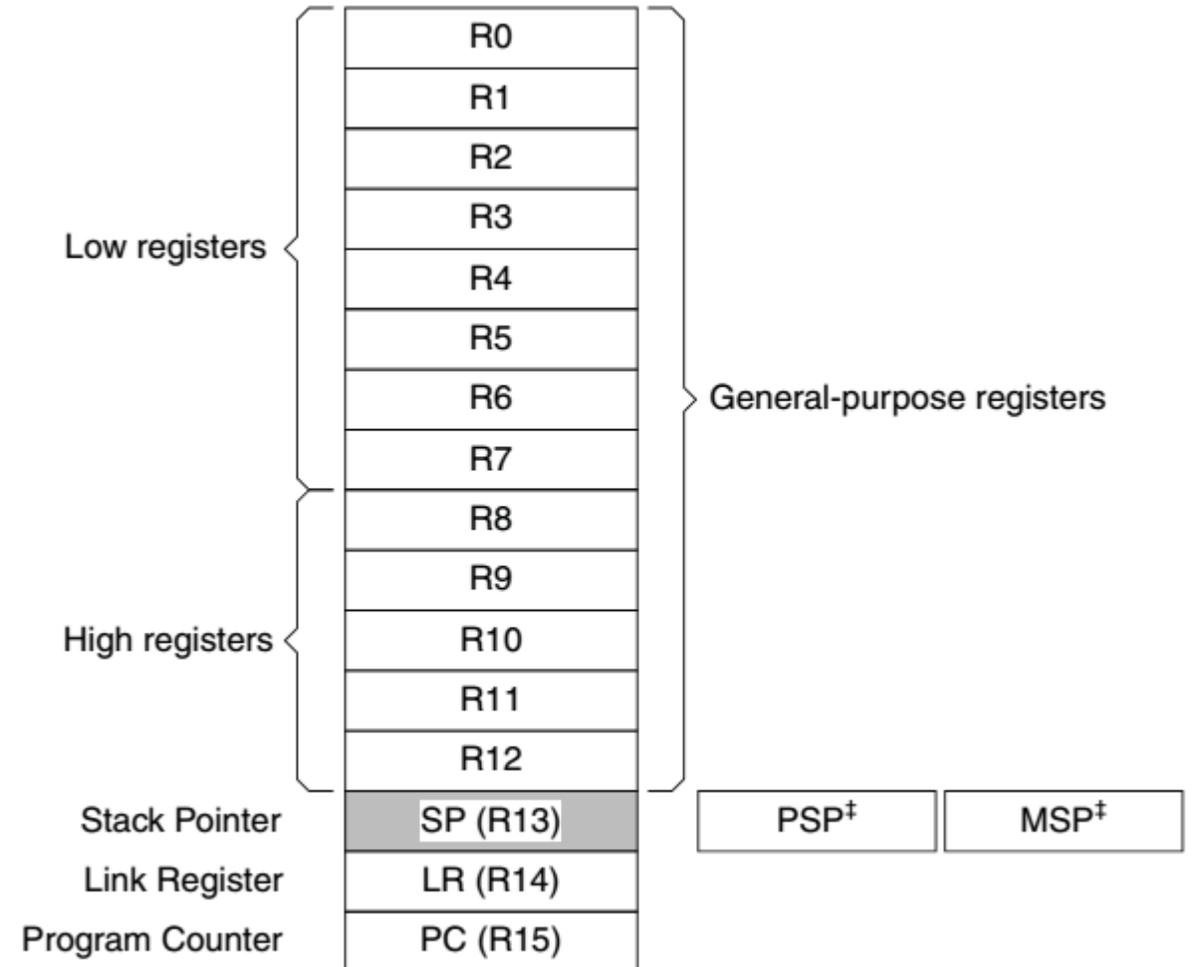
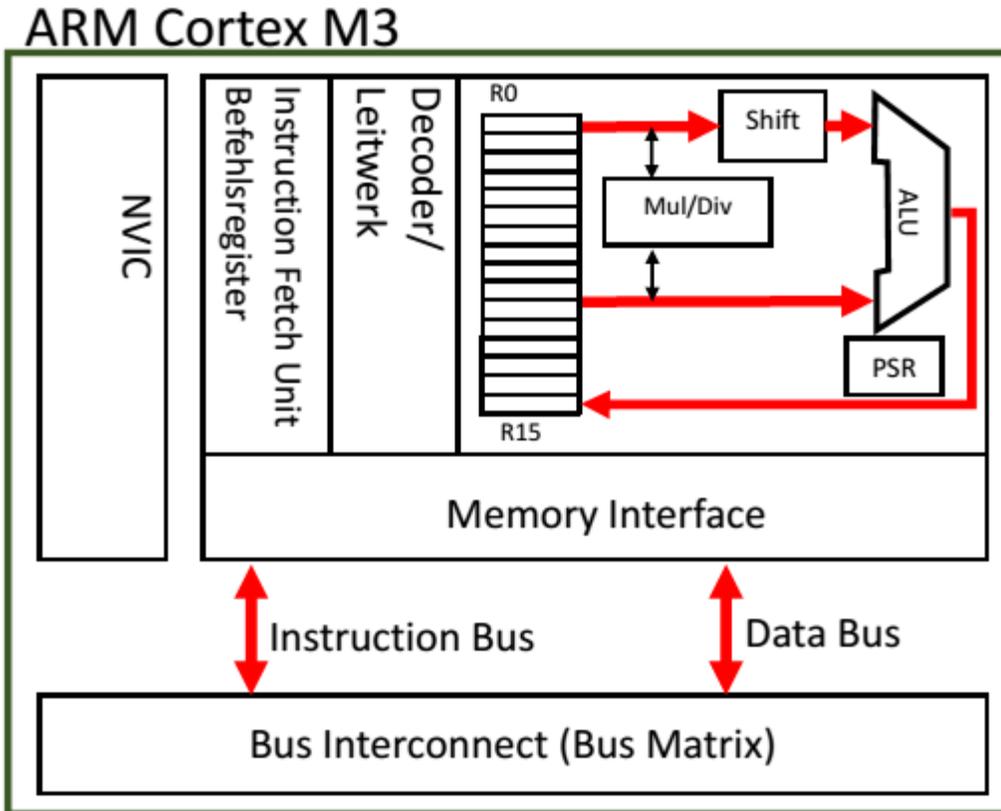
- STM32L152RET Mikrocontroller



- 32 Bit Architektur
- 32 MHz
- RISC: 1 Befehl pro Takt
- Harvard-Architektur
- 3-stufige Pipeline
- 32 x 32Bit in einem Takt
- Load-Store-Architektur

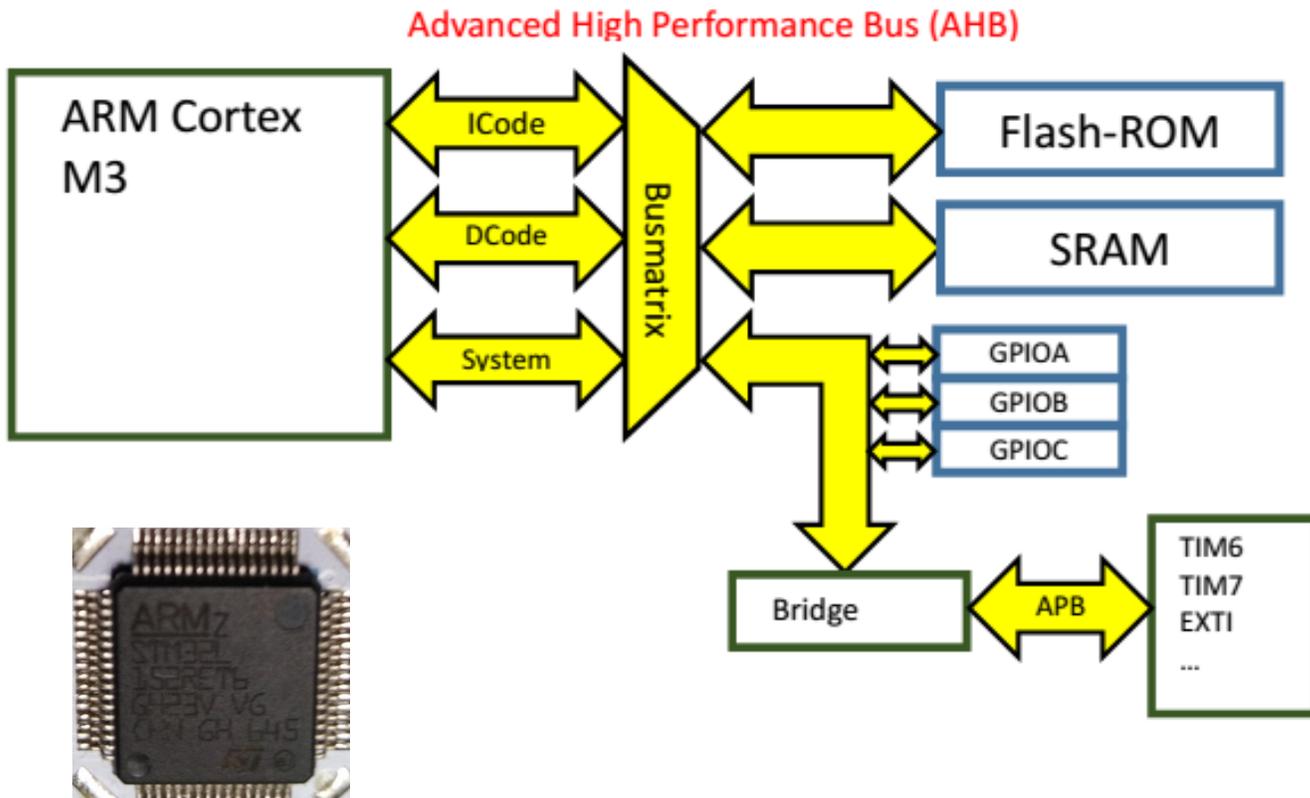
Vorstellung des ARM-MC

- STM32L152RET Mikrocontroller

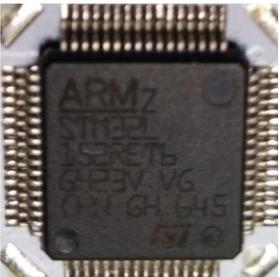


Vorstellung des ARM-MC

- STM32L152RET Mikrocontroller



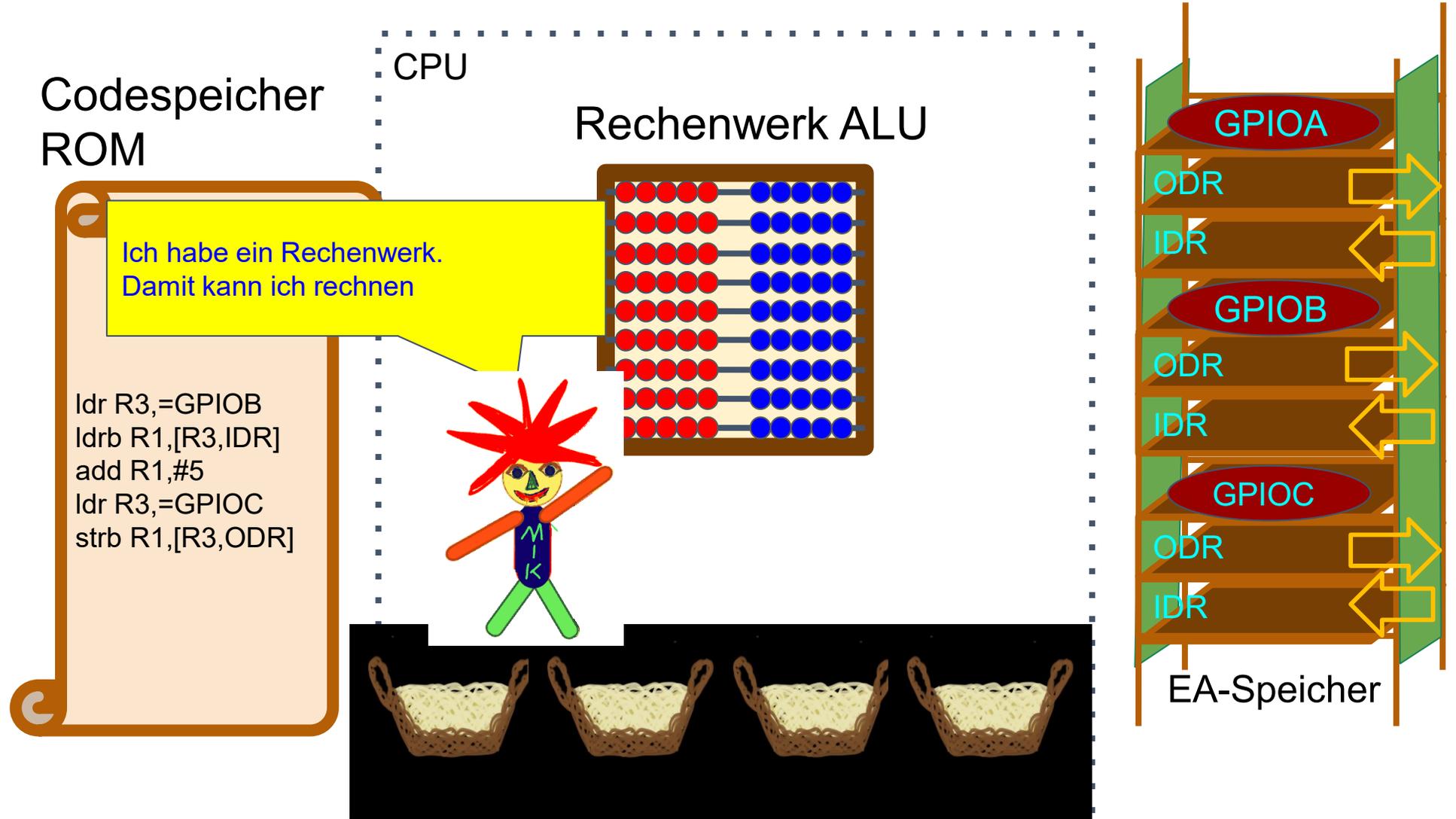
- 512kB Flash
- 80kB SRAM
- 3 x 16Bit Digital-IO
- 9 Timer
- 3 SPI
- 2 I2C
- 5 USART
- 1 12-Bit ADC mit 21 Kanälen
- 2 DAC



Vorstellung des ARM-MC

- STM32L152RET Mikrocontroller

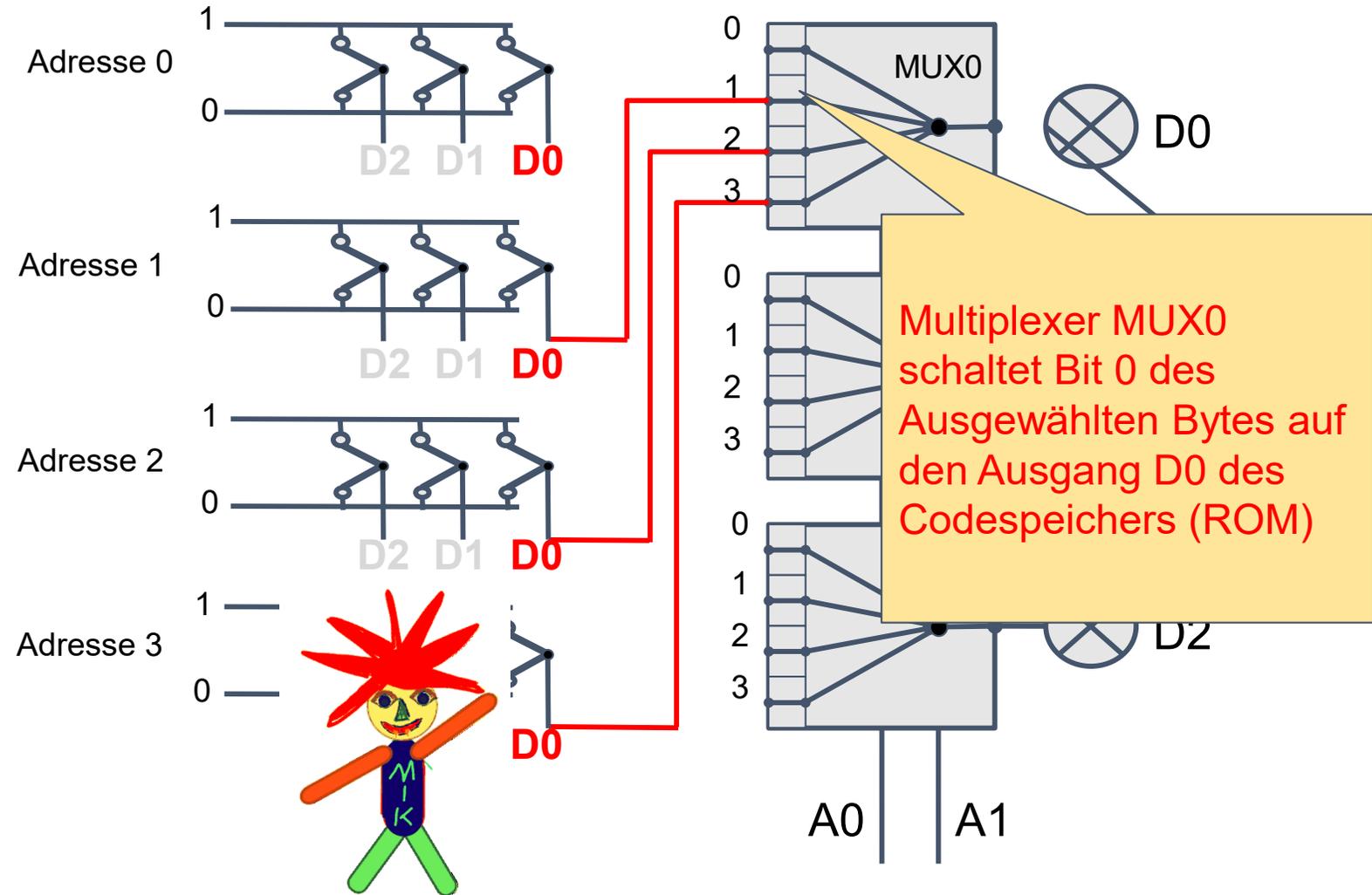
Körbchen-
modell:



Vorstellung des ARM-MC

- STM32L152RET Mikrocontroller

Codespeicher

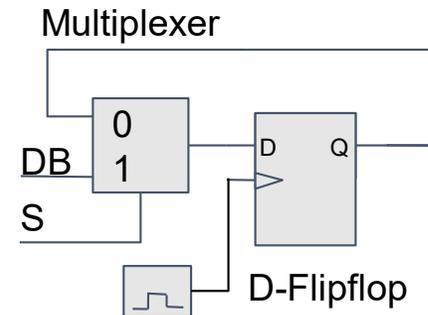


Vorstellung des ARM-MC

- STM32L152RET Mikrocontroller

Register

DB	Q	S	Takt	Q
X	0	0		0
X	1	0		1
0	X	1		0
1	X	1		1



Entweder soll das Register bei einem Takt einen neuen Wert vom Datenbus DB einspeichern, oder den alten Wert behalten.



Vorstellung des ARM-MC

- STM32L152RET Mikrocontroller

Register

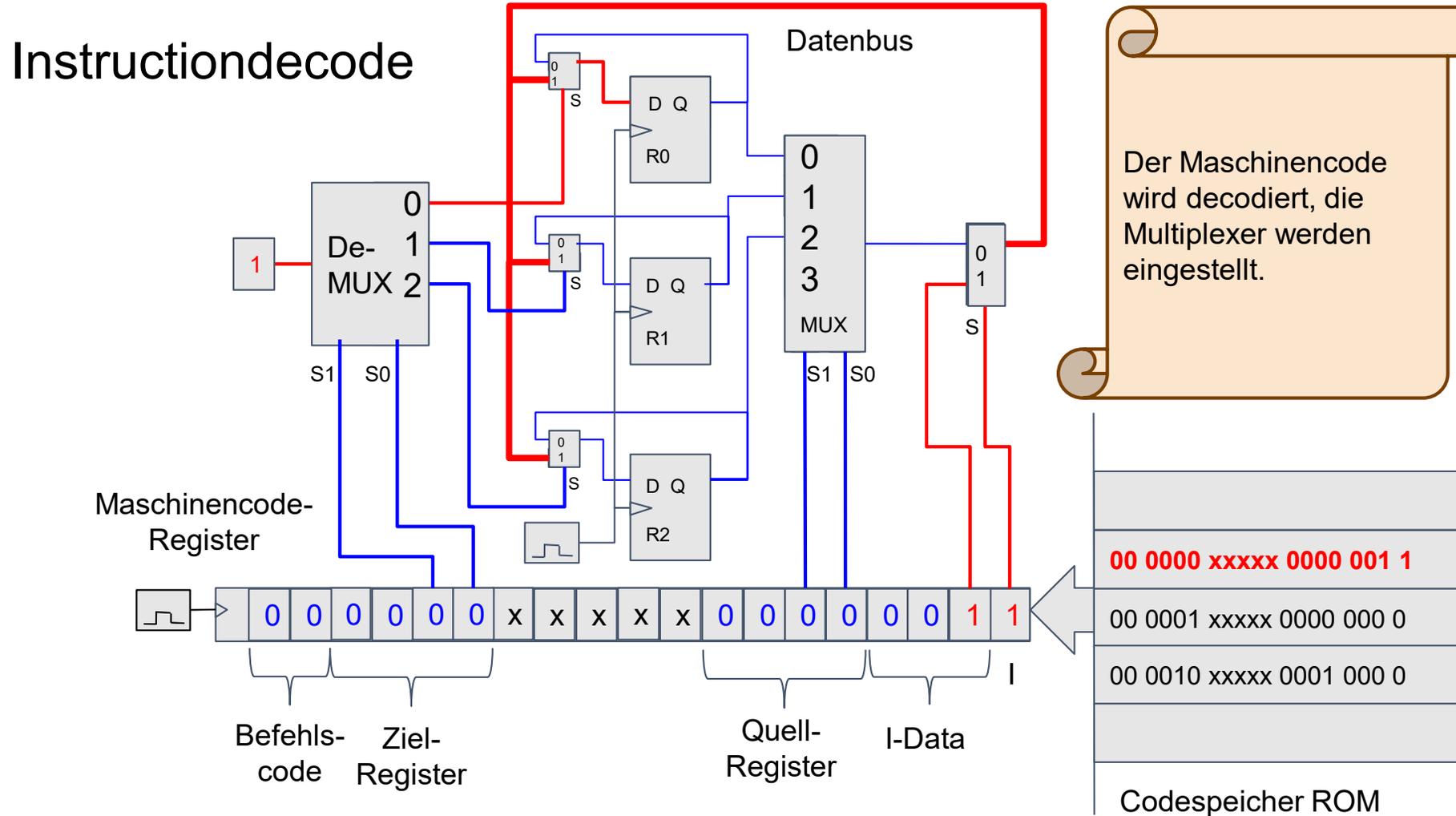
32 Flipflops bilden ein Register in der ARM CPU



Vorstellung des ARM-MC

- STM32L152RET Mikrocontroller

CPU als Informationslogik



Vorstellung des ARM-MC

- STM32L152RET Mikrocontroller

Pipeline

Nächster Takt:

4. Befehl Holen

3. Befehl Dekodieren

2. Befehl Ausführen

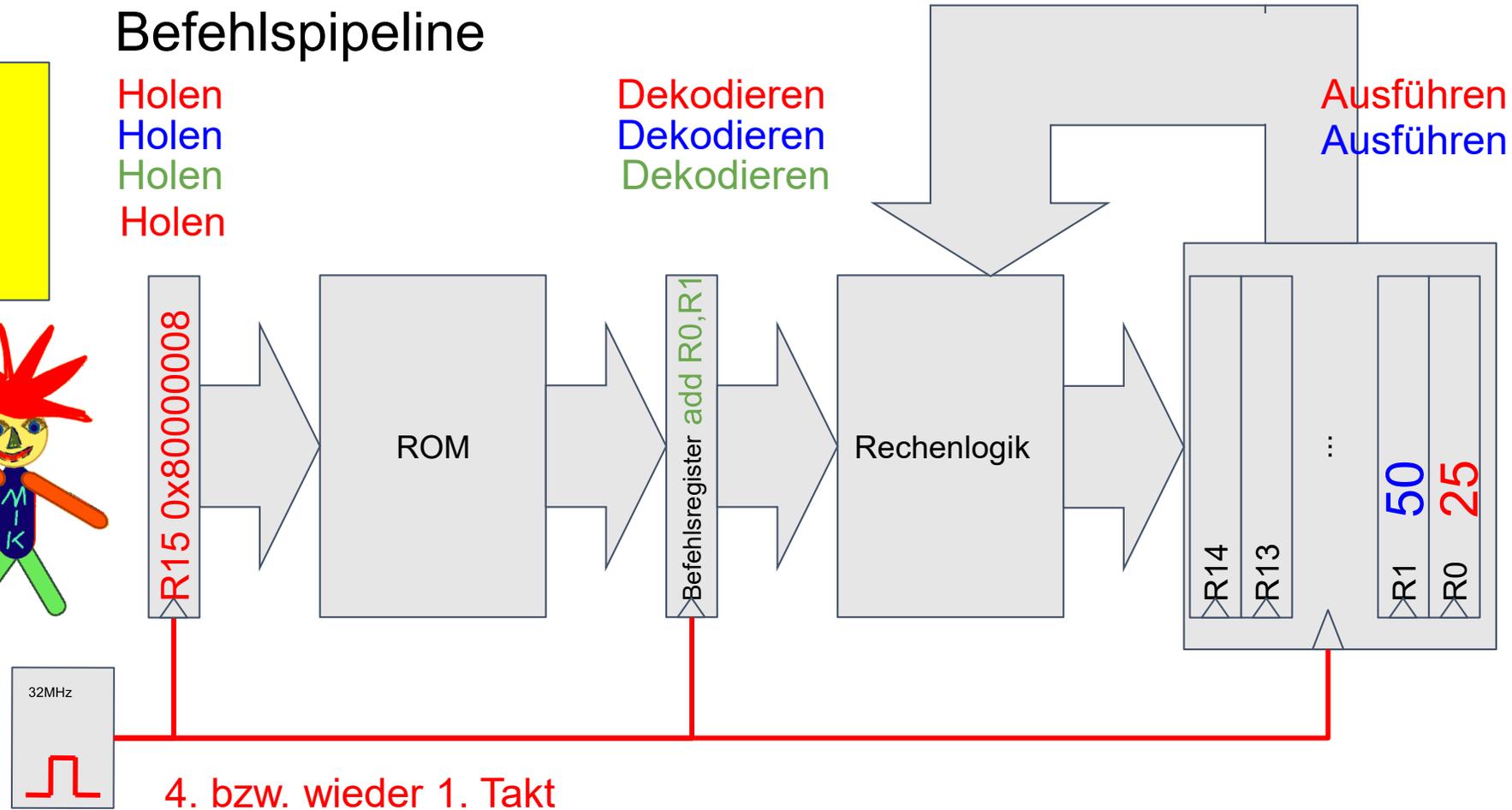


Befehlspipeline

Holen
Holen
Holen
Holen

Dekodieren
Dekodieren
Dekodieren

Ausführen
Ausführen

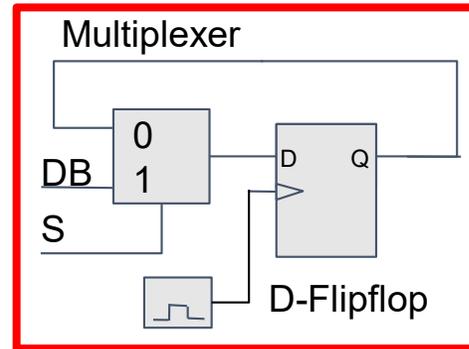


Vorstellung des ARM-MC

- STM32L152RET Mikrocontroller

RAM

8 Flipflops bilden ein Byte im RAM der ARM CPU



D7

Byte

D0



Vorstellung des ARM-MC

- STM32L152RET Mikrocontroller

RAM

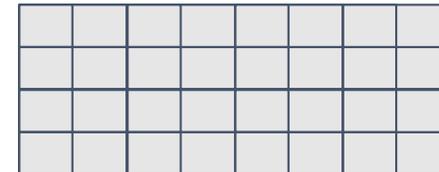
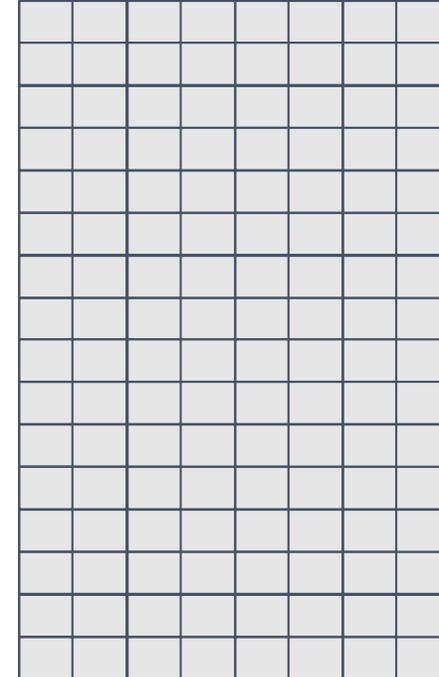
80 kByte bilden das RAM des
STM32L152RET Mikrocontrollers



0x20000000
0x20000001
0x20000002
0x20000003
0x20000004
0x20000005
0x20000006
0x20000007
0x20000008
0x20000009
0x2000000A
0x2000000B
0x2000000C
0x2000000D
0x2000000E
0x2000000F

...

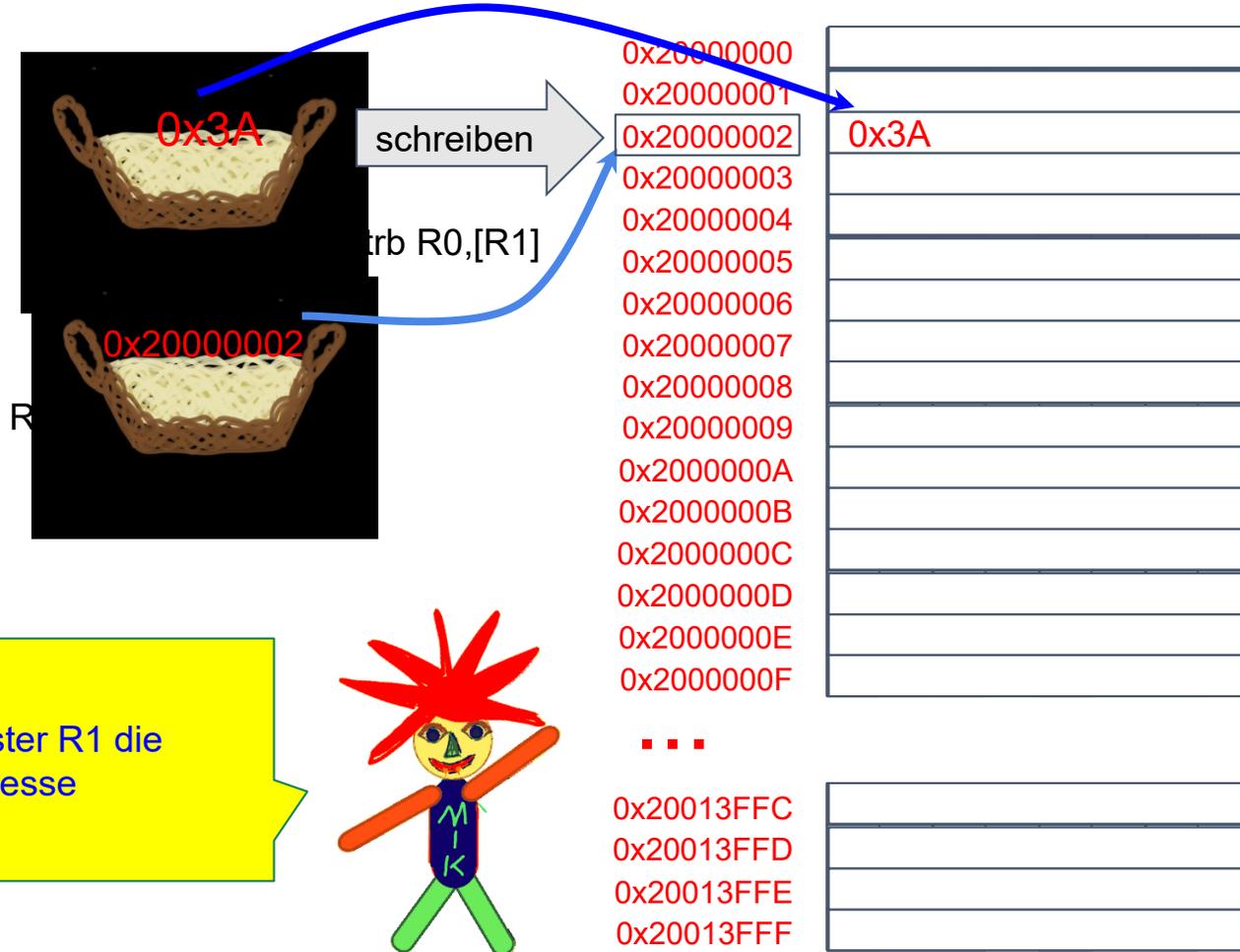
0x20013FFC
0x20013FFD
0x20013FFE
0x20013FFF



Vorstellung des ARM-MC

- STM32L152RET Mikrocontroller

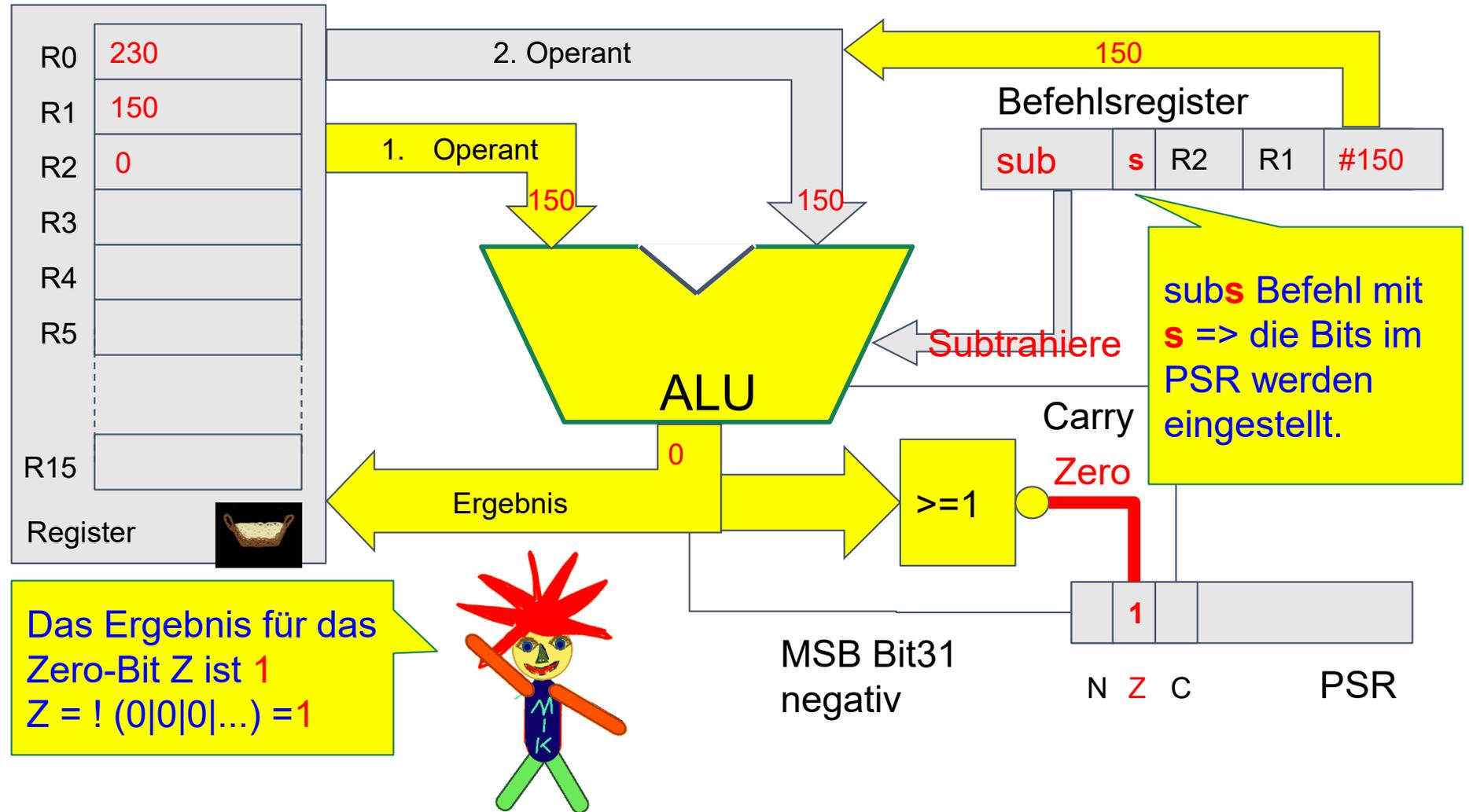
RAM



Vorstellung des ARM-MC

- STM32L152RET Mikrocontroller

ALU

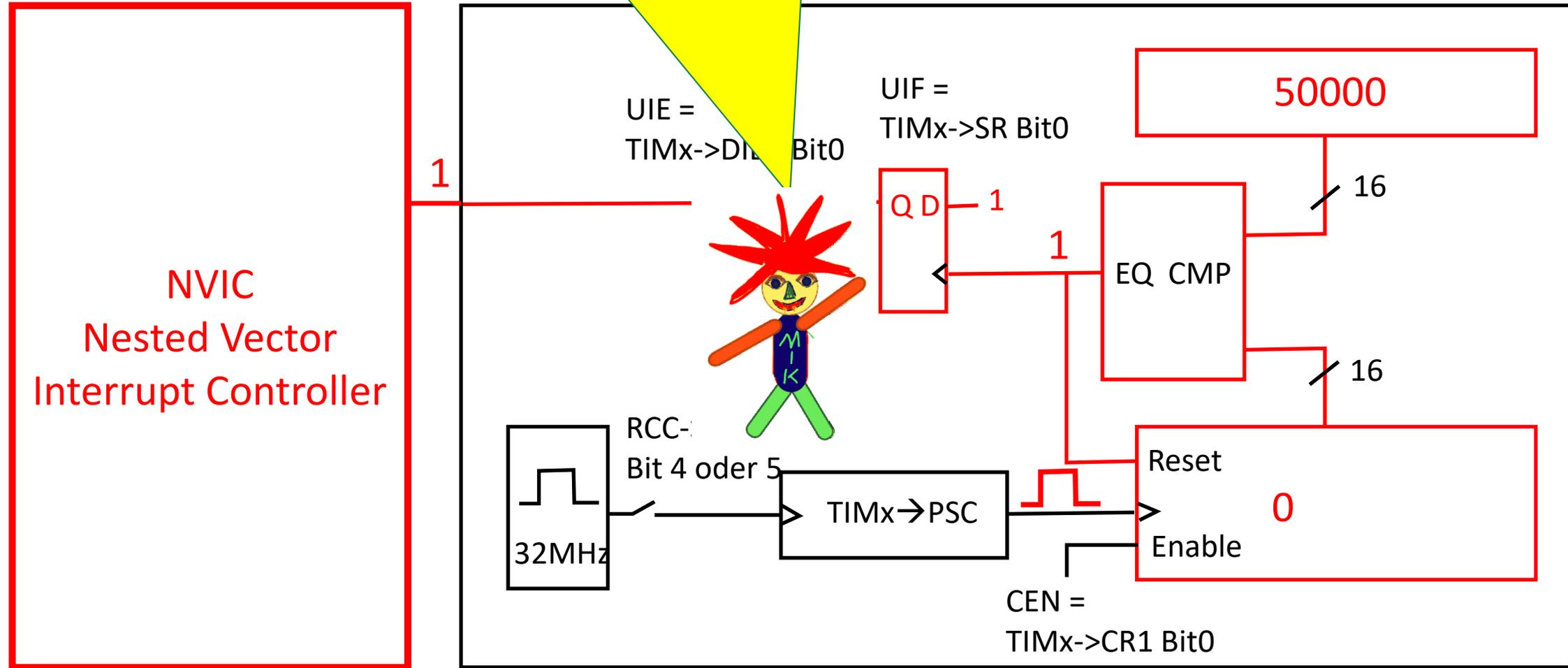


Vorstellung des ARM-MC

- STM32L152RET Mikrocontroller

Timer und Interrupt

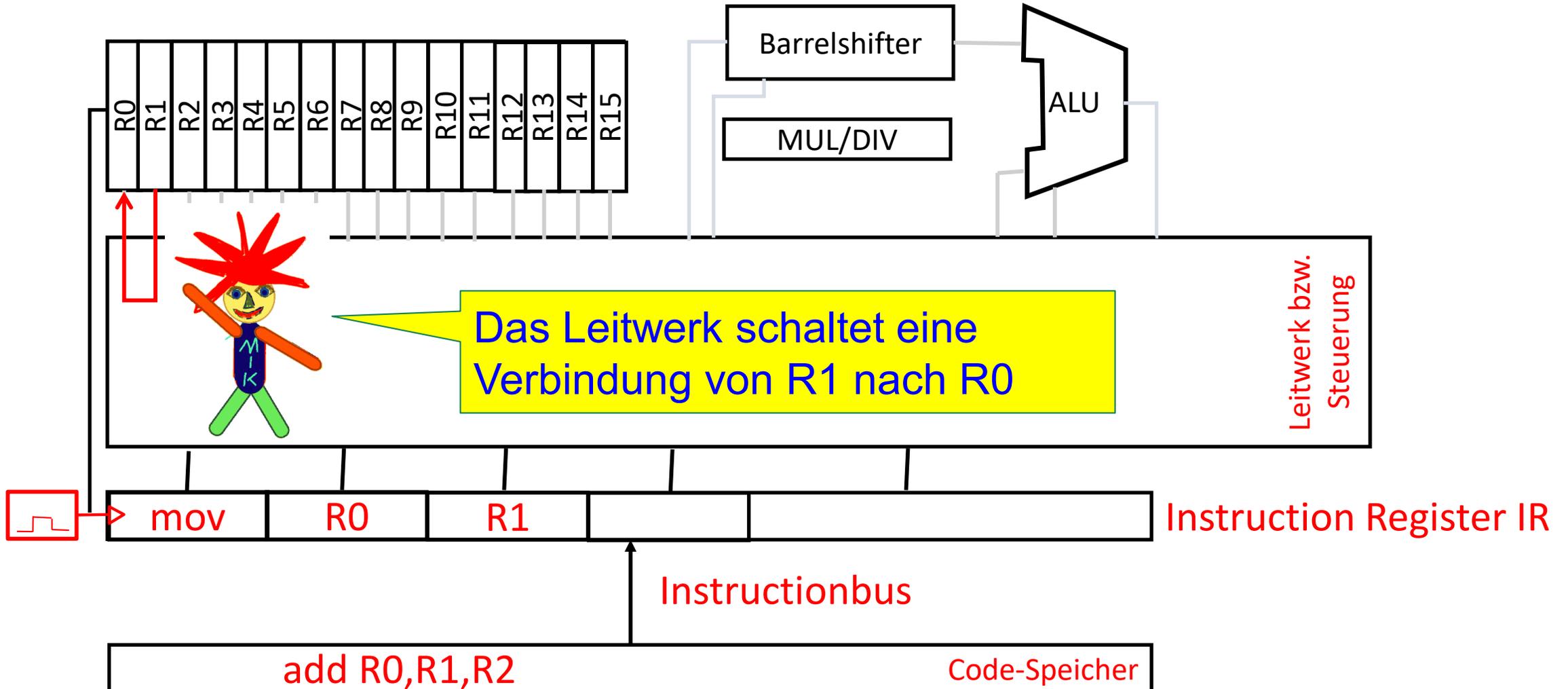
Gleichzeitig wird das UIF Flipflop (Update-Interrupt-Flag) auf 1 gesetzt



Vorstellung des ARM-MC

- STM32L152RET Mikrocontroller

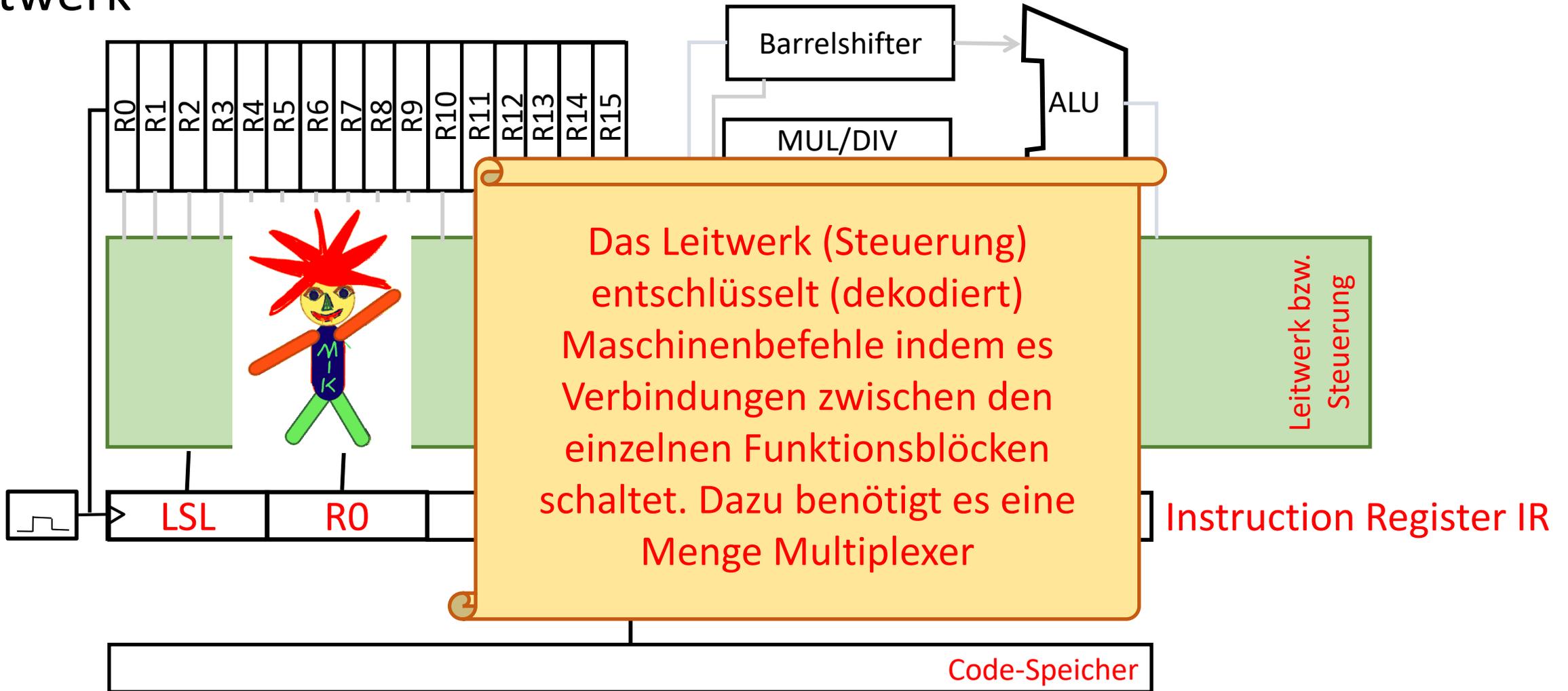
Leitwerk



Vorstellung des ARM-MC

- STM32L152RET Mikrocontroller

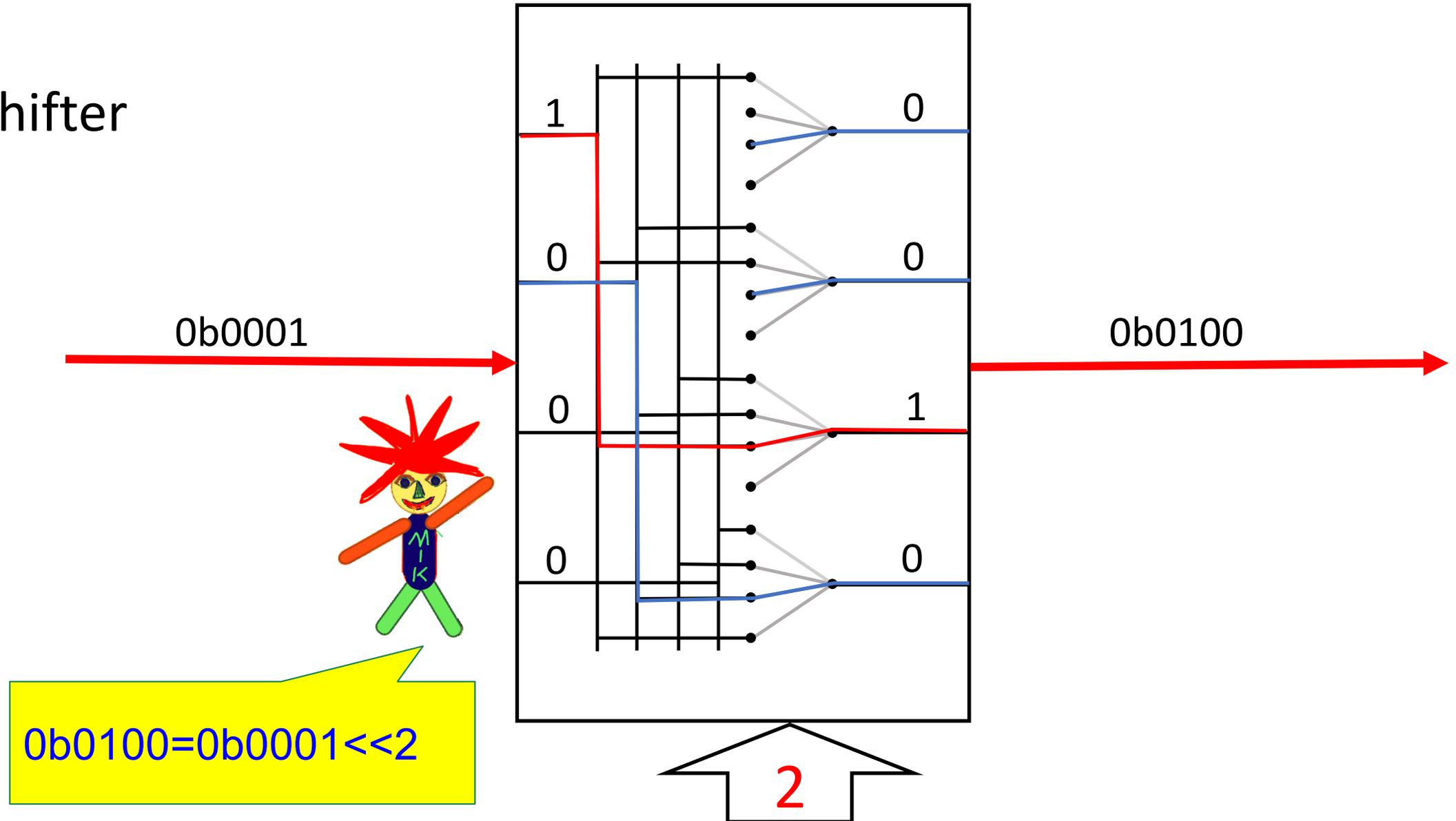
Leitwerk



Vorstellung des ARM-MC

- STM32L152RET Mikrocontroller

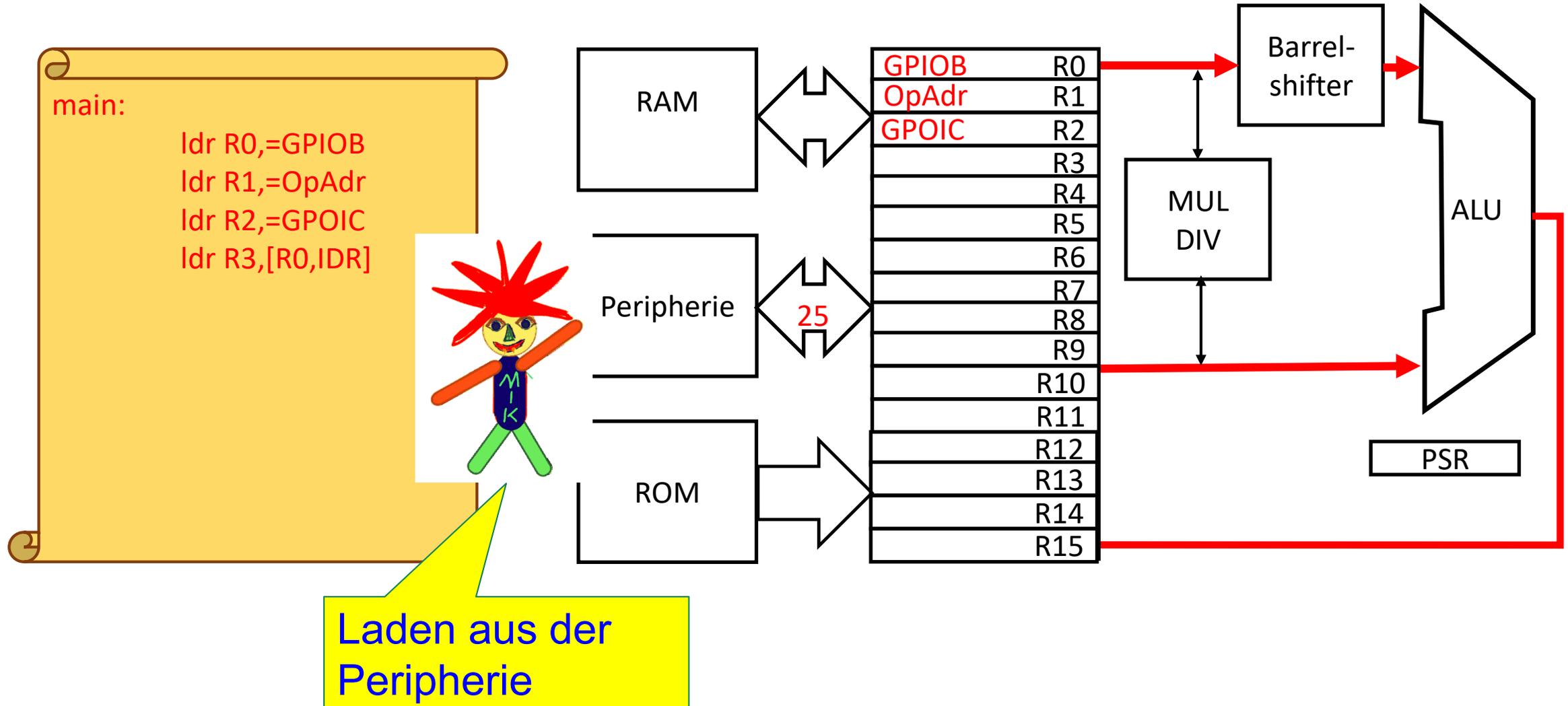
Barrelshifter



Vorstellung des ARM-MC

- STM32L152RET Mikrocontroller

STM32 Load-Store-Architektur

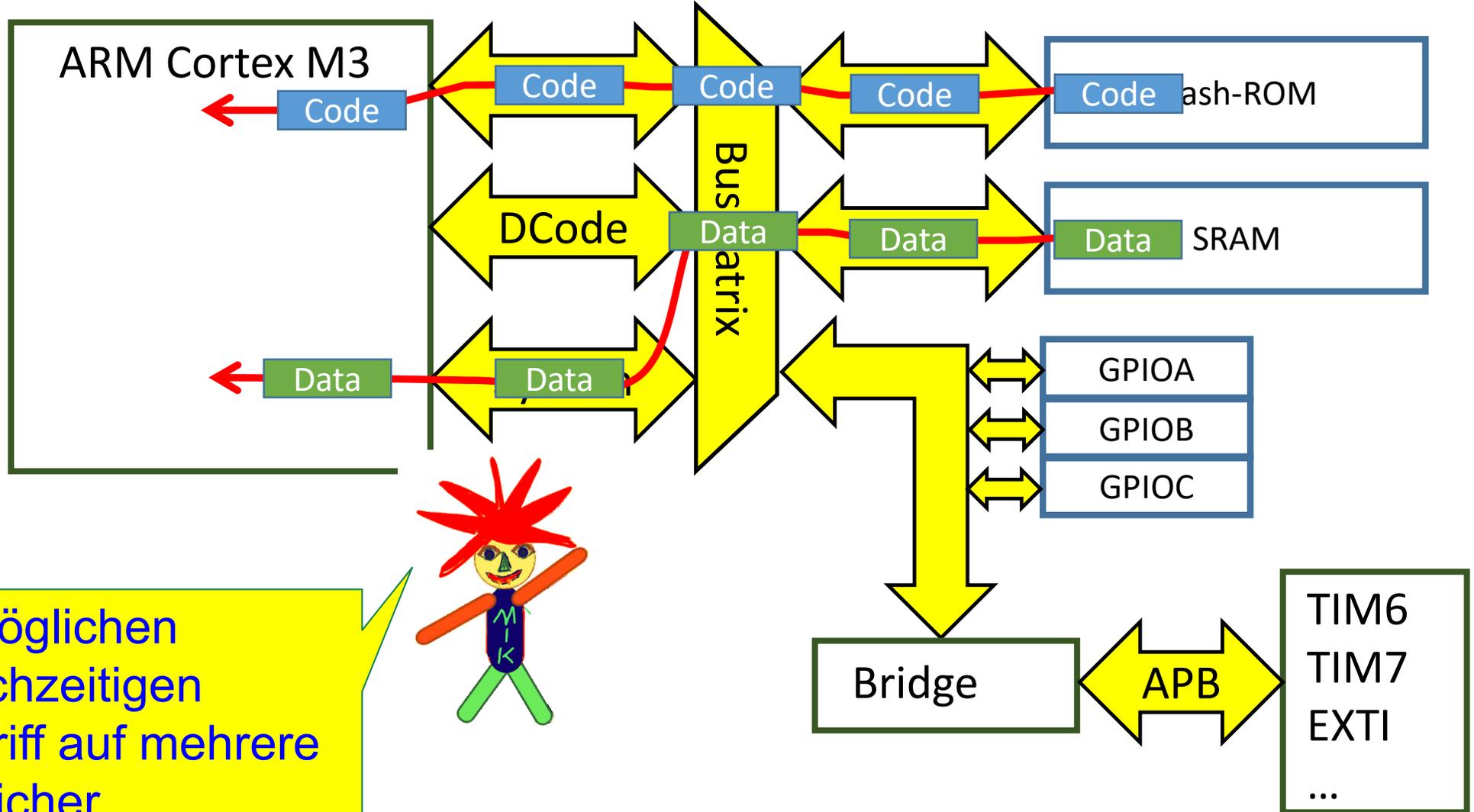


Vorstellung des ARM-MC

- STM32L152RET Mikrocontroller

Cortex-M3
Speicher-
architektur

Advanced High Performance Bus (AHB)



Ermöglichen
gleichzeitigen
Zugriff auf mehrere
Speicher



Vorstellung des ARM-MC

• STM32L152RET Mikrocontroller

Cortex-M3
Speicher-
architektur

ARM



Adressbus HADDR mit 32 Adressleitungen

Datenbus HWDATA mit 32 Datenleitungen

Datenbus HRDATA mit 32 Datenleitungen

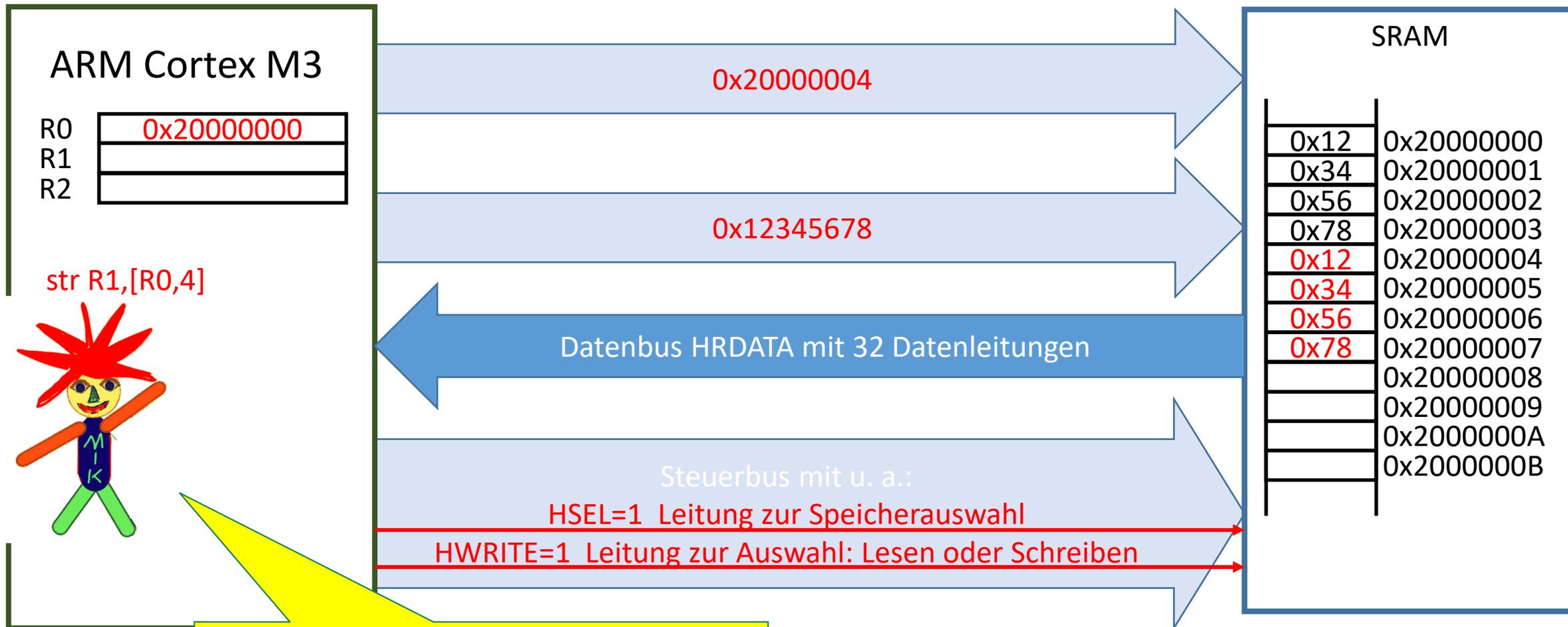
Steuerbus mit u. a.:
HSEL Leitung zur Speicherauswahl
HWRITE Leitung zur Auswahl: Lesen oder Schreiben

Jeder Kommunikationspfad verfügt über Adress-, Daten-, und Steuerbusse

M6
TIM7
EXTI
...

Vorstellung des ARM-MC • STM32L152RET Mikrocontroller

Cortex-M3 Speicherarchitektur



Beim str-Befehl kommuniziert die CPU mit dem Speicher schreibend

Vorstellung des ARM-MC

- STM32L152RET Mikrocontroller

RISC - CISC

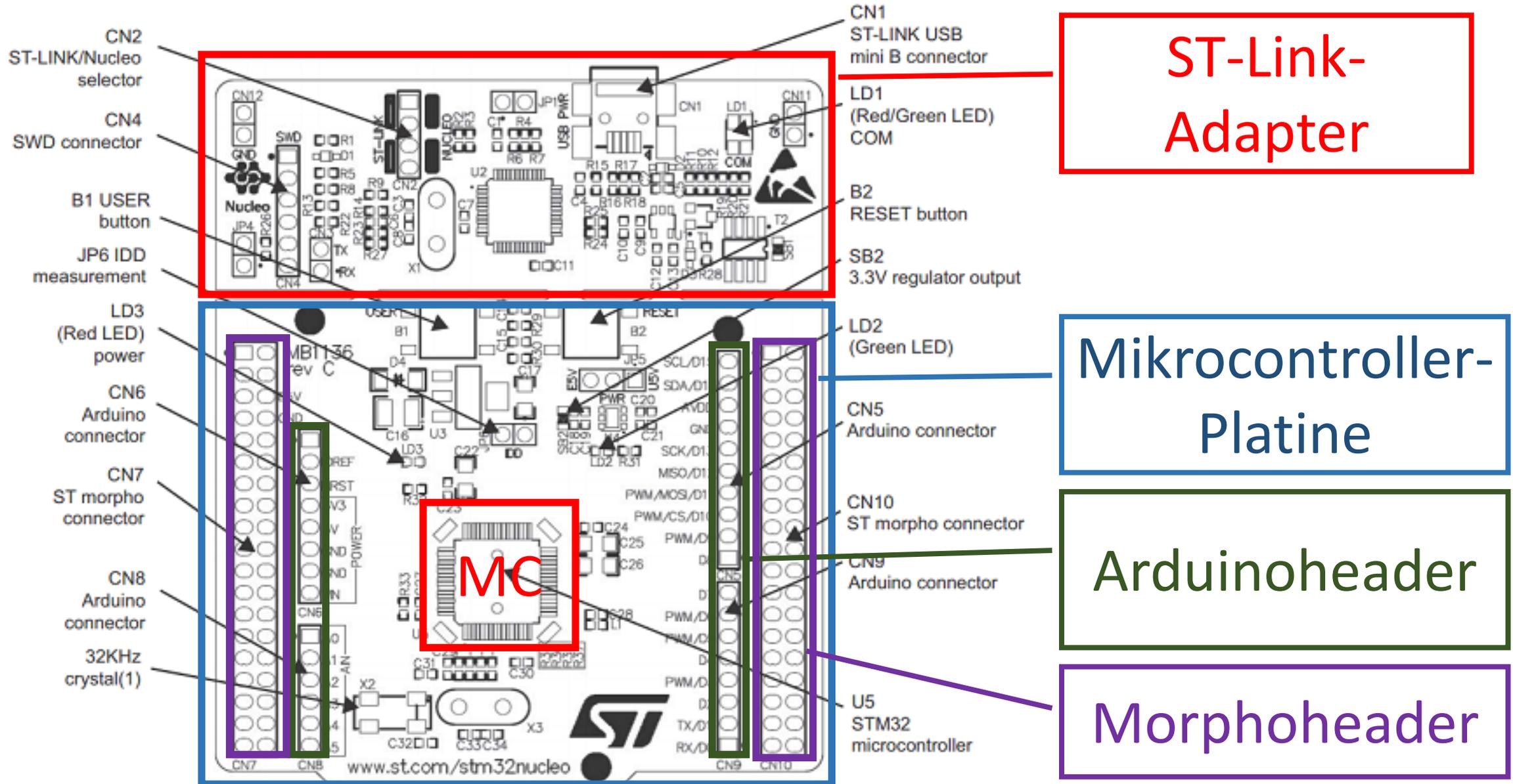
	8051 CISC	ARM RISC
Addition einer Zahl aus dem RAM zu einem Register	add A,Adresse (12 Takte) 2 Byte	ldr R1,=Adresse (1 Takt) ldr R2,[R1,0] (1 Takt) add R0,R0,R2 (1 Takt)
Eine Speicherstellen im RAM um 1 verringern, bei ungleich 0 verzweige zur Sprungmarke	djnz Adresse,Sprungmarke 4 Byte	ldr R1,=Adresse ldr R2,[R1,0] subs R2,#1 str R2,[R1,0] bne Sprungmarke 5 Takte
Ein Register erhöhen	inc A (12 Takte) 1 Byte	add R0,#1 (1 Takt) Alle Befehle 32 Bit



CISC arbeitet oft Befehl für Befehl ab
RISC verwendet eine Pipeline

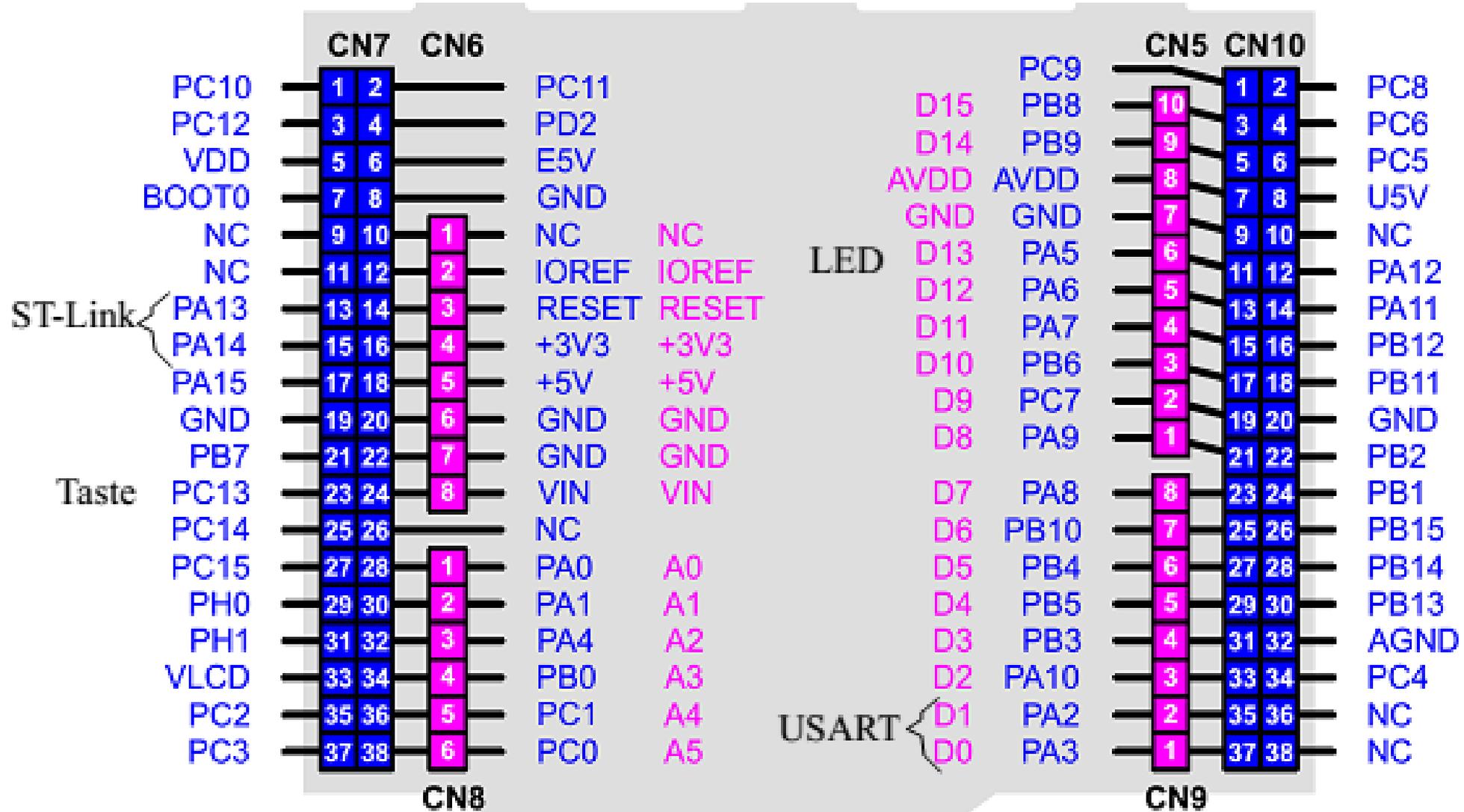
Vorstellung des ARM-MC

- STM32 Nucleo



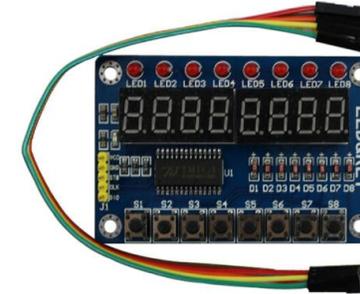
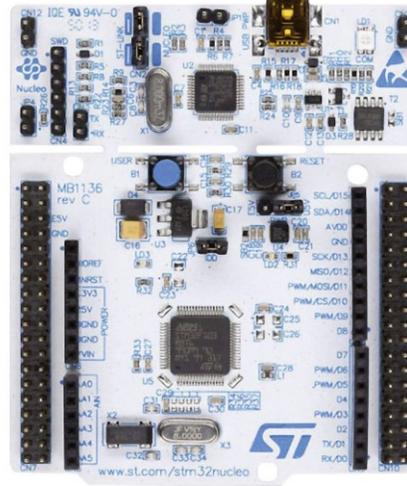
Vorstellung des ARM-MC

- STM32 Nucleo



Vorstellung des ARM-MC

- Evaluationsboard



Vorstellung des ARM-MC

• Evaluationsboard

Stromversorgung
für Versuche

Temperatursensor

Analog Poti

8 LED

2mm Buchsen

LED-Display

8 Hochstromausgänge

LAN-Buchse ENC28J60

ESP01 WLAN Buchse

2 Ultraschall

USB

6 Taster

Alle Ports auf Stift- oder
Buchsenleiste

8 Schalter

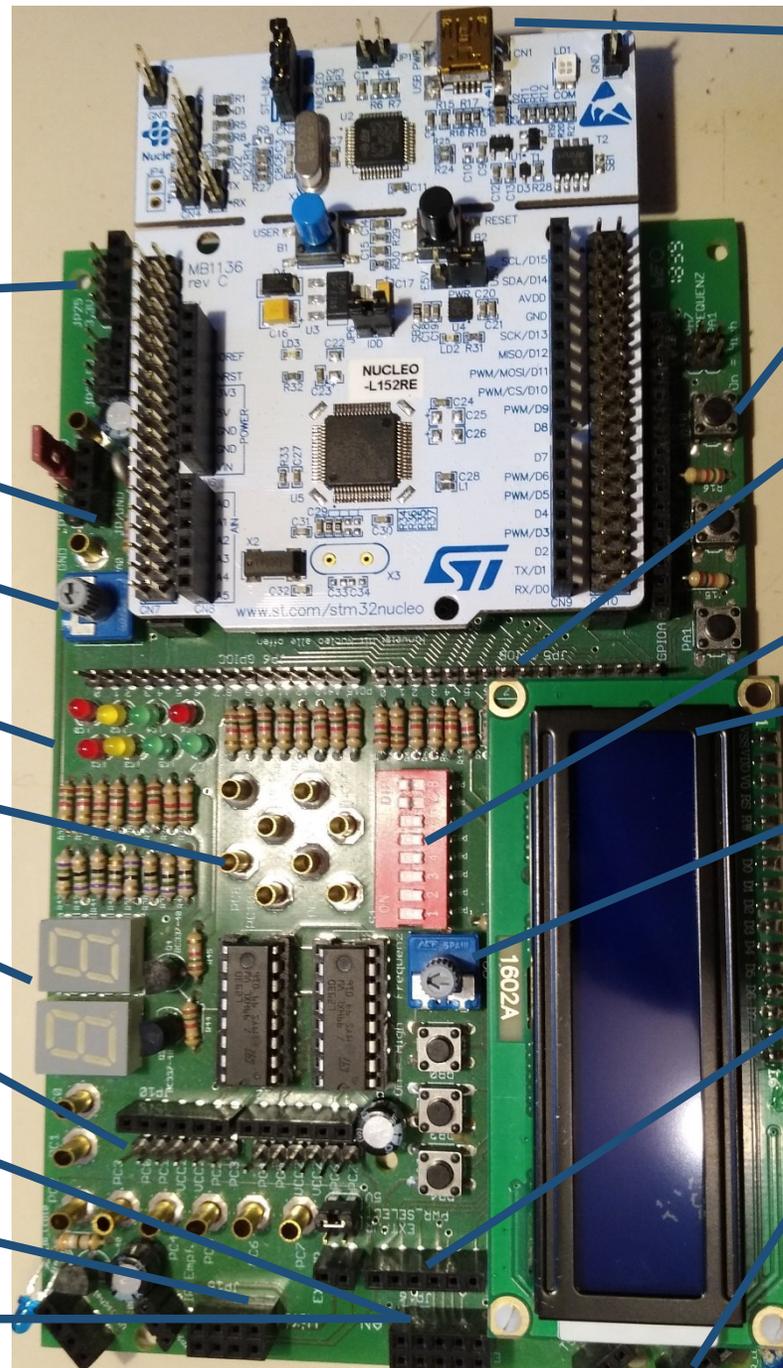
LCD

Frequenzgenerator 555

Anschluss für
HC05 Bluetooth

2 IR-Empfänger

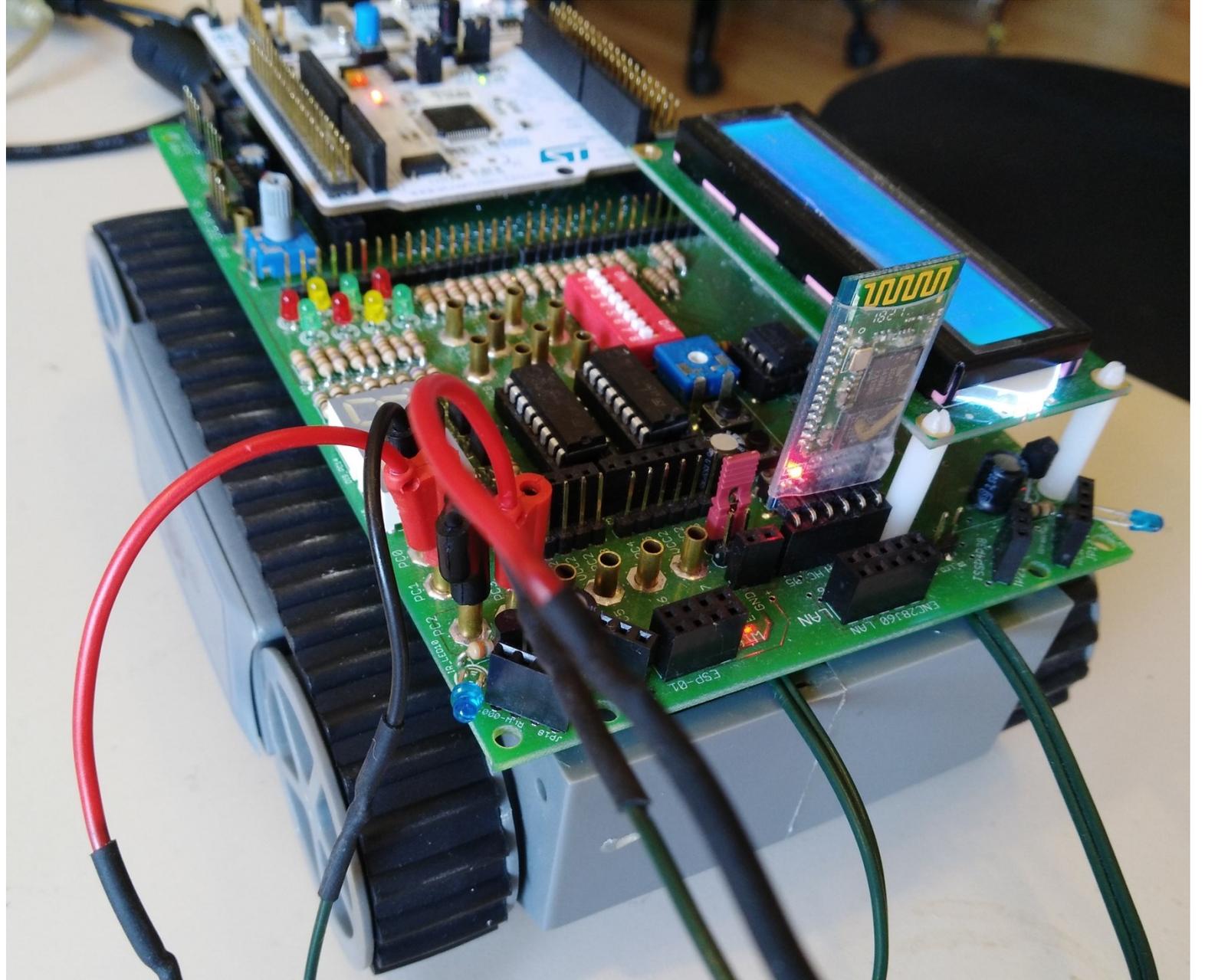
2 IR-
Sendediode



Vorstellung des ARM-MC

- Evaluationsboard

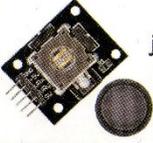
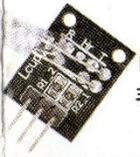
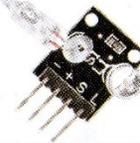
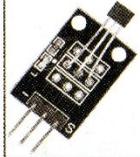
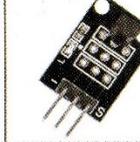
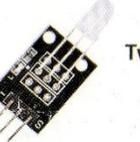
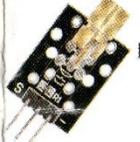
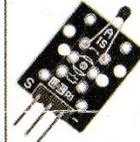
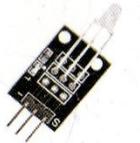
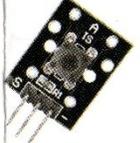
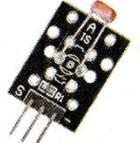
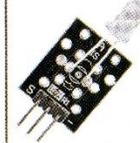
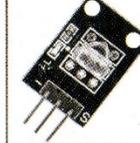
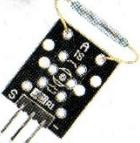
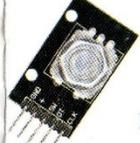
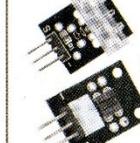
Beispiel:
Steuerung des Arexx
Roboters RP5 mit
Bluetooth



Vorstellung des ARM-MC

- Evaluationsboard

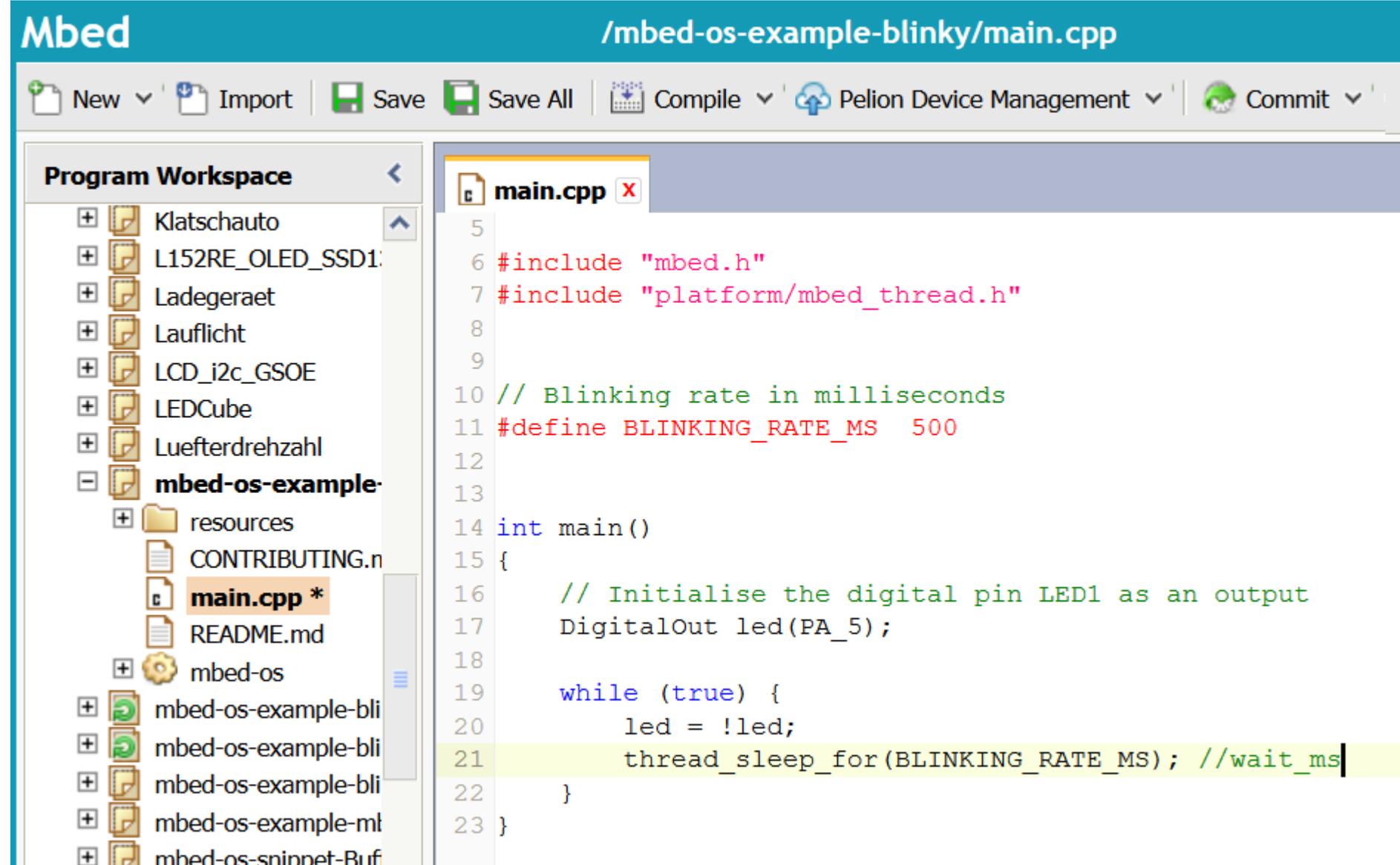
Sensor Kit for Arduino

 <p>joystick XY摇杆</p>	 <p>Flame 火焰传感器</p>	 <p>RGB LED 3色灯模块</p>	 <p>Heartbrat 手指测心跳</p>	 <p>Light Cup 魔术光杯</p>	 <p>Hall magnatic 霍尔磁力</p>
 <p>Relay 继电器</p>	 <p>Linear Hall 线性霍尔</p>	 <p>SMD RGB 3色RGB</p>	 <p>7 color flash 7彩闪烁</p>	 <p>Tilt switch 水银开关</p>	 <p>TEMP 18B20</p>
 <p>Big sound 高感度声音</p>	 <p>Touch 人体触摸</p>	 <p>Two-color LED 双色LED</p>	 <p>Laser emit 激光发射</p>	 <p>Ball switch 倾斜开关</p>	 <p>Analog temp 模拟温度</p>
 <p>Small sound 小麦克风</p>	 <p>Digital temp 数字温度</p>	 <p>Two-color 小双色</p>	 <p>Button 按键开关</p>	 <p>Photoresistor 光线</p>	 <p>IR emission 红外发射</p>
 <p>Tracking 循迹</p>	 <p>buzzer 有源蜂鸣器</p>	 <p>Reed switch 磁簧开关</p>	 <p>Shock 震动开关</p>	 <p>Temp and humldty 温湿度</p>	 <p>IR receiver 红外接收</p>
 <p>Avoid 避障碍</p>	 <p>passive buzzer 无源蜂鸣器</p>	 <p>Mini Reed 迷你磁簧</p>	 <p>Rotary encoders 旋转编码器</p>	 <p>Analog Hall 模拟霍尔</p>	 <p>Tap module 敲击模块 Light blocking 光遮断</p>

Vorstellung des ARM-MC

- Tools

MBED Online
IDE für C/CPP



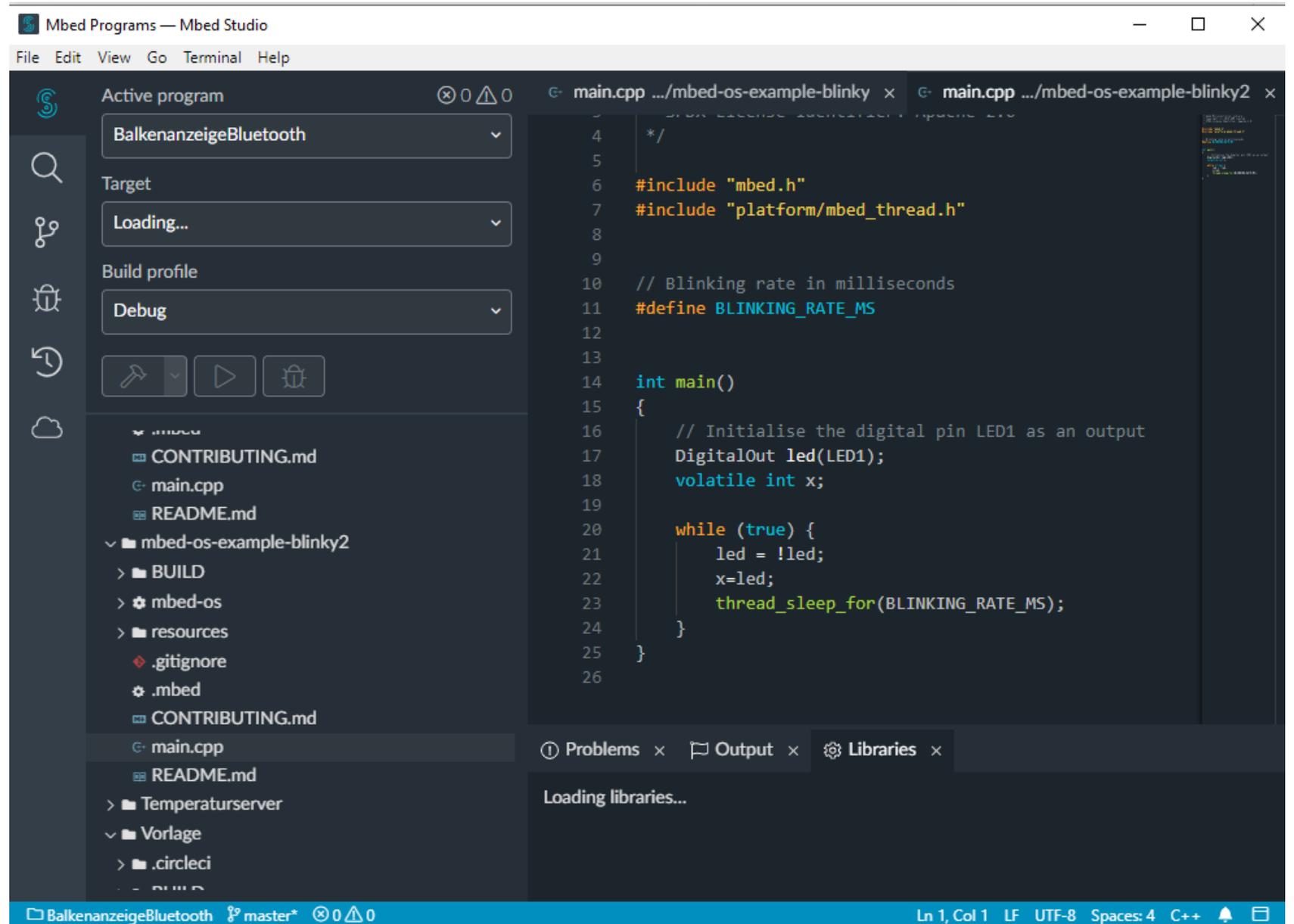
The screenshot shows the Mbed IDE interface. The title bar indicates the current file is `/mbed-os-example-blinky/main.cpp`. The menu bar includes options for New, Import, Save, Save All, Compile, Pelion Device Management, and Commit. The left sidebar displays the 'Program Workspace' with a tree view of project files, including a folder named 'mbed-os-example-' containing 'resources', 'CONTRIBUTING.n', 'main.cpp *', and 'README.md'. The main editor area shows the following C++ code:

```
5
6 #include "mbed.h"
7 #include "platform/mbed_thread.h"
8
9
10 // Blinking rate in milliseconds
11 #define BLINKING_RATE_MS 500
12
13
14 int main()
15 {
16     // Initialise the digital pin LED1 as an output
17     DigitalOut led(PA_5);
18
19     while (true) {
20         led = !led;
21         thread_sleep_for(BLINKING_RATE_MS); //wait_ms
22     }
23 }
```

Vorstellung des ARM-MC

- Tools

MBED Online
IDE für C/CPP

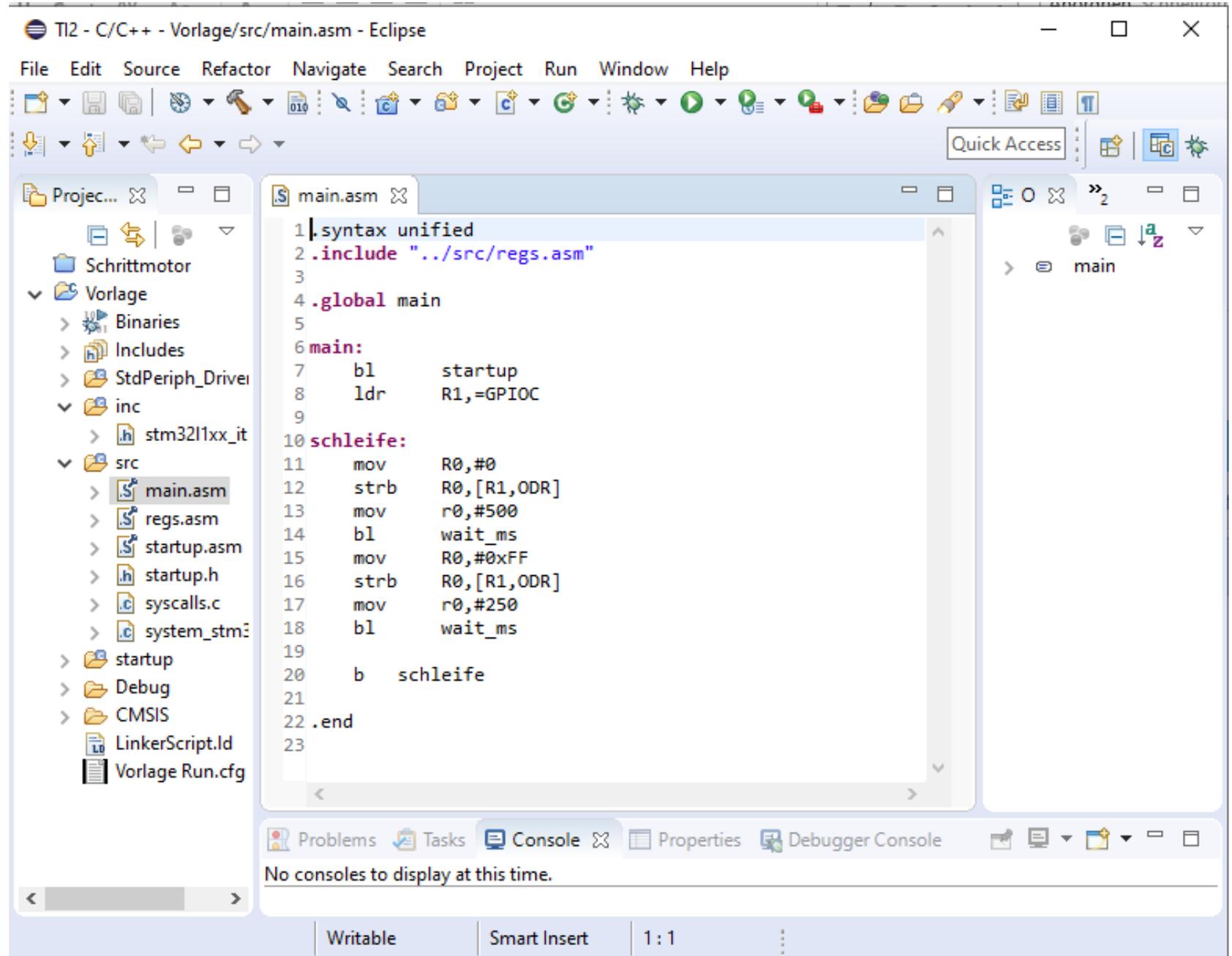


Vorstellung des ARM-MC

- Tools

SW4STM32 für
Assembler

Alternativ:
STM32CUBE IDE
(Achtung alle .asm
müssen in .s
umbenannt
werden)



```
T12 - C/C++ - Vorlage/src/main.asm - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Projec...
  Schrittmotor
  Vorlage
    Binaries
    Includes
    StdPeriph_Driver
    inc
      stm321xx_it
    src
      main.asm
      regs.asm
      startup.asm
      startup.h
      syscalls.c
      system_stm32
    startup
    Debug
    CMSIS
    LinkerScript.ld
    Vorlage Run.cfg
main.asm
1 |.syntax unified
2 |.include "../src/regs.asm"
3
4 |.global main
5
6 |main:
7 |    bl    startup
8 |    ldr   R1,=GPIOC
9
10 |schleife:
11 |    mov   R0,#0
12 |    strb  R0,[R1,ODR]
13 |    mov   r0,#500
14 |    bl   wait_ms
15 |    mov   R0,#0xFF
16 |    strb  R0,[R1,ODR]
17 |    mov   r0,#250
18 |    bl   wait_ms
19
20 |    b    schleife
21
22 |.end
23
Problems Tasks Console Properties Debugger Console
No consoles to display at this time.
Writable Smart Insert 1:1
```

Vorstellung des ARM-MC

- Tools

Weitere:

- Keil
- EnBitz
- CoIDE

Vorstellung des ARM-MC

- Tools

The screenshot shows a mobile application interface for configuring a microcontroller. The title bar reads "Mikrocontroller". Below the title bar, there is a search bar containing "b schleife" and navigation arrows. A list of registers is displayed on the left, with checkboxes for each. The registers are grouped into PSR, R0-R12, and PB0-PB3. The right side of the screen shows a track diagram for a robot, with various components labeled in pink text.

Register	Value	Checked
PSR	N	<input type="checkbox"/>
0	Z	<input type="checkbox"/>
R0	Cy	<input type="checkbox"/>
40020800	PC0	<input checked="" type="checkbox"/>
R1	PC1	<input type="checkbox"/>
40020400	PC2	<input checked="" type="checkbox"/>
R2	PC3	<input type="checkbox"/>
55	PC4	<input checked="" type="checkbox"/>
R3	PC5	<input type="checkbox"/>
0	PC6	<input checked="" type="checkbox"/>
R4	PC7	<input type="checkbox"/>
0	PC8	<input type="checkbox"/>
R5	PC9	<input type="checkbox"/>
0	PC10	<input type="checkbox"/>
R6	PC11	<input type="checkbox"/>
0	PC12	<input type="checkbox"/>
R7	PC13	<input type="checkbox"/>
0	PC14	<input type="checkbox"/>
R8	PC15	<input type="checkbox"/>
0	PC15	<input type="checkbox"/>
R9	PC15	<input type="checkbox"/>
0	PC15	<input type="checkbox"/>
R10	PB0	<input checked="" type="checkbox"/>
0	PB1	<input type="checkbox"/>
R11	PB2	<input checked="" type="checkbox"/>
0	PB3	<input type="checkbox"/>
R12	PB3	<input type="checkbox"/>

Track Diagram Labels:

- main-nasht
- LDR R0,=GPIO C
- ldr R1,=GPIO B
- schleife:
- ldr R2, [R1,DR]
- strb R2, [R0,DDR]
- b schleife