

Beispielprogramme

Beispielprogramm Button

button.c

/*

Von 8 Bit Daten sind 5 Bit für die 5 Buttons bzw. 5 Leds

Wird ein Button gedrückt, leuchtet die jeweilige Led

*/

#define reg_leds (*(volatile uint8_t*)0xFFFFFFF0) //Adresse Leds

#define reg_button (*(volatile uint8_t*)0xFFFFFFE0) //Adresse Buttons

```
void main() {
    uint8_t btn0 = reg_button;
    while (1) {
        btn0 = reg_button; //Einlesen der Buttons Bit0=Button1, Bit1=Button2 , etc.
        reg_leds = btn0; //Ausgabe an Leds Bit0=Led1, Bit1=Led2, etc.
    }
}
```

Beispielprogramm Binärzähler

led.c

//Die Leds zeigen einen Zähler, der binär hochzählt. Die Leds haben die Adresse 0xFFFFFFF0

#include <stdint.h>

#include <stdbool.h>

#define reg_leds (*(volatile uint8_t*)0xFFFFFFF0)

```
void main() {
```

```
    uint32_t led = 0;
```

```
    while (1) {
```

```
        //32 Bit werden hochgezählt, Bit 17-21 werden an den Leds angezeigt
```

```
        reg_leds = (led >> 16) & 0xFF; led = led + 1;
```

```
    }
```

```
}
```

Beispielprogramm Binärzähler in Assembler

devices.h:

```
#ifndef DEVICES_H
#define DEVICES_H
#define DEV_LED 0xFFFFFFFF //Adresse für die Leds

#endif
```

blink-test.s:

```
/*
Register t0 wird auf 0 gesetzt. In einer Schleife wird dem Wert 1 dazu addiert und um 16 Stellen
nach rechts geschiftet in Register t1 geschrieben. Danach wird der Wert an der Speicheradresse der
Leds gespeichert.
*/
#include "devices.h"

.section .text

.global _start

main:
    li t0,0

loop:
    addi t0,t0,1
    srli t1,t0,16
    sb t1,DEV_LED(zero)

    j loop

.size main, .-main
```