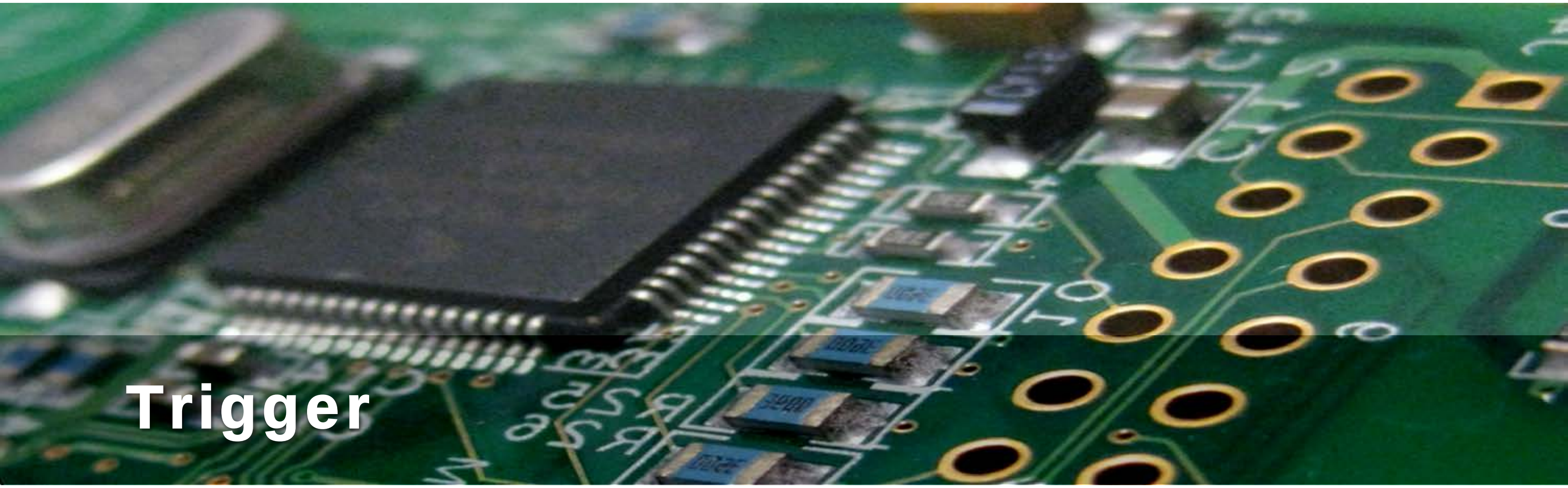
Lucerne University of Applied Sciences and Arts

HOCHSCHULE LUZERN

Technik & Architektur



Trigger

“It would be good if we could get notifications in the future. Back to the future would be an excellent thing.”

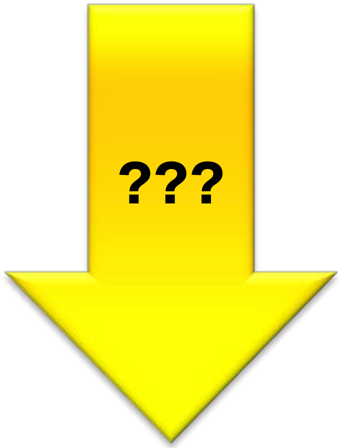
Prof. Erich Styger
erich.styger@hslu.ch
+41 41 349 33 01

Scriptum: Triggers

Learning Goals

- Problem: we have a periodic timer, need now to cause events in the future

- Creation of Trigger Module
  - Timer usage
  - Adding trigger in the future
  - Callback methods
- Trigger Usage
  - Interrupt synchronization
  - Flashing LED every 500 ms
  - Buzzer on key press



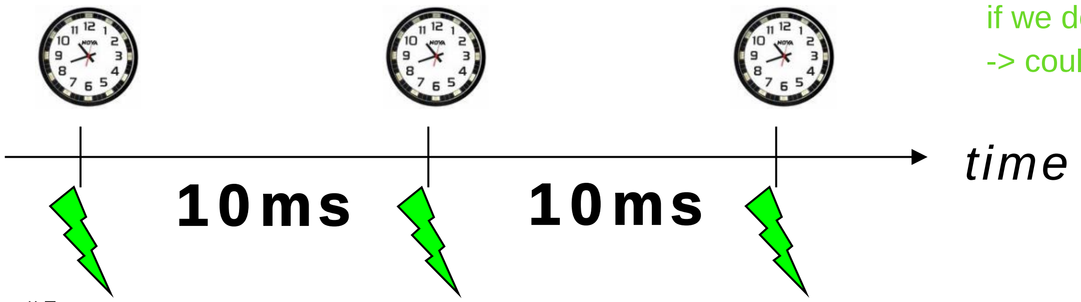
Problem 1

- Assumption
  - System with 10ms periodic Timer

- Requirement
  - Flashing LED
  - Every 500 ms

```
void ISR_On10ms(void) {
  static uint8_t i = 0;
  i++;
  if (i==50) {
    LED0_Neg();
    i = 0;
  }
}
```

if we do all things together from next slide in this code
-> could be difficult



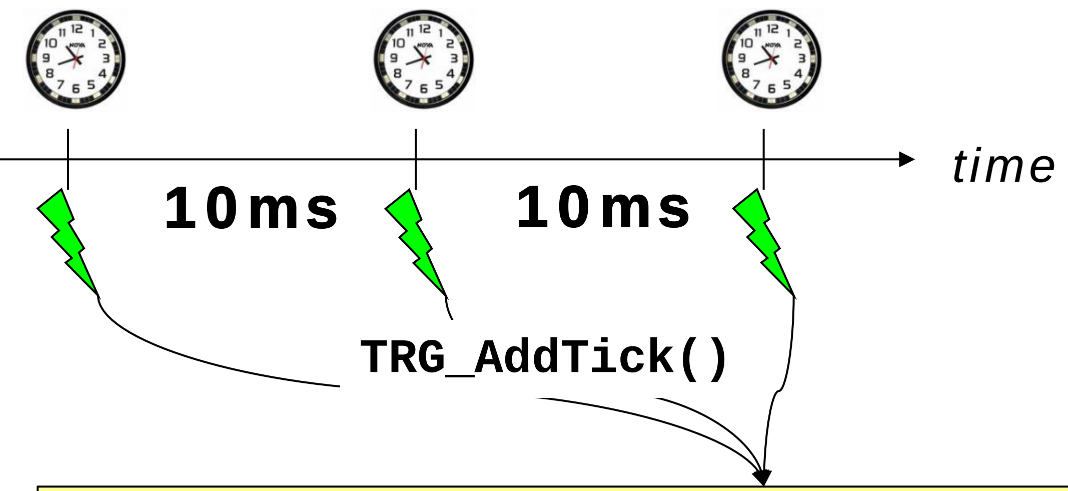
Even more complicated things to do?!?

a) Flash LED every 500 ms
b) Button pressed
   ➔ Turn on LED1 for 200 ms, then turn off
c) Button pressed
   ➔ Start sounder for 500 ms, then turn off
d) Combine a), b) and c) (!!!)

Need for a **common** infrastructure?

Minimal memory usage
1 timer/reuse
Universal interface

Idea



```
void TMR_On10ms(void)
{
  …
  TRG_AddTick();
  …
}
```

```
void TRG_AddTick(void) {
  Increment Tick Counter;
  if HasTriggerForThisTickCount then
    removeTrigger;
    callback();
  end if
}
```

counts the number of timer events or ticks

Reentrancy!

Triggers and Callbacks

- Service Module
  - Counting ticks: Called by periodic timer
  - Adding Triggers (by when, do what)
  - Checks if #ticks reached trigger ➔ Do Action

- 'Do Action'
  - Callback
  - Function Pointer

recursive function

```
void (*trg)(void);

void test(void) {
  trg = test;
  trg();
}
```

```
void (*trg)(uint8_t);

void test(uint8_t ch) {
  trg = trg;
  trg(ch++);
}
```

Trigger Descriptor

- Need
  - Trigger time
  - What to do
- Pointer to void: ‘generic’/‘opaque’ data pointer

```
typedef void (*TRG_Callback)(void*);
```

void-pointer -> pointer to nothing

```
typedef struct {
  uint16_t triggerTick;
  TRG_Callback callback;
  void *data;
} TriggerDesc;
```

```
#ticks0
LED_Neg()
&data0
```

Lucerne University of
Applied Sciences and Arts

HOCHSCHULE
LUZERN

Technik & Architektur

Trigger Interface (Trigger.h)

```
#define TRG_TICKS_MS  TMR_TICK_MS
```

we have a makro telling what the frequency is

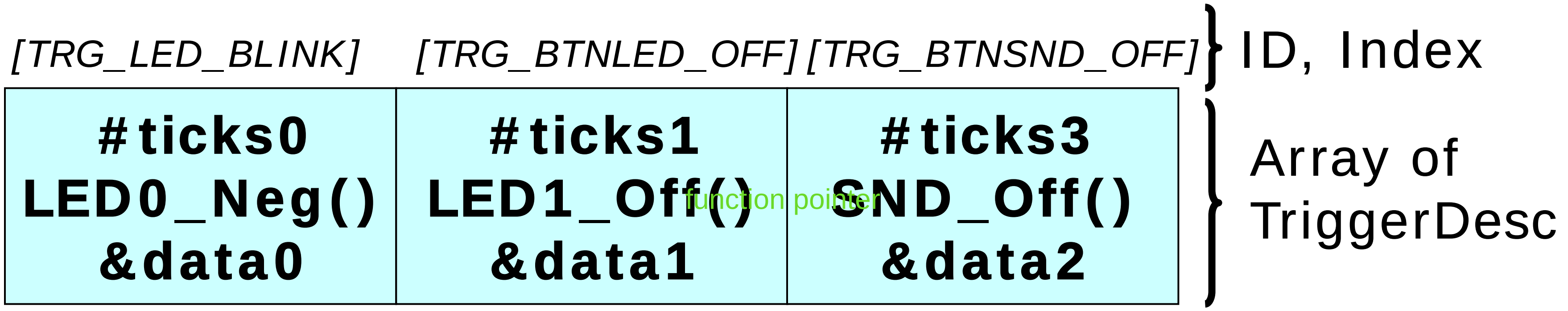
```
typedef enum {
  TRG_BUZ_BEEP, /*!< Buzzer beep */
  TRG_NOF_TRIGGERS /*!< Must be last! */
} TRG_TriggerKind;
typedef void *TRG_CallBackDataPtr;
typedef void (*TRG_Callback)(TRG_CallBackDataPtr);
typedef uint16_t TRG_TriggerTime;

uint8_t TRG_SetTrigger(TRG_TriggerKind trigger,
TRG_TriggerTime ticks, TRG_Callback callback,
TRG_CallBackDataPtr data);

void TRG_AddTick(void);
```

events

typedef for de void-pointer

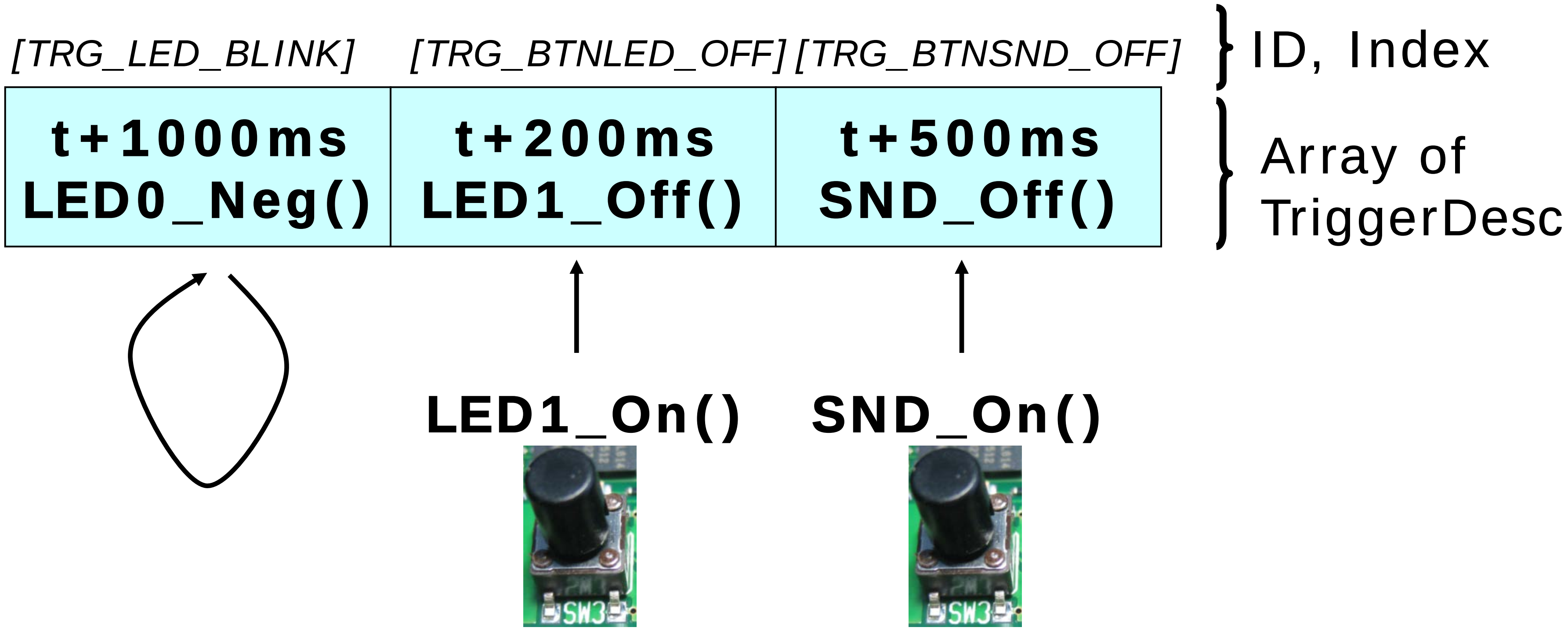
Trigger Descriptor

```
typedef enum {
  TRG_LED_BLINK,
  TRG_BTNLED_OFF,
  TRG_BTNSND_OFF,
  TRG_NOF_TRIGGERS  /*!< Must be last! */
} TRG_TriggerKind;

static TRG_TriggerDesc TriggerList[TRG_NOF_TRIGGERS];
```



like the eventbits, multiple trigger in an array

Trigger Examples



absolute & relative time

#Ticks options:
- **a) absolute: tick count at which to trigger compare with actual tick count**
- **b) relative: how many ticks to go decrement, check on zero**

Example Implementation

Example 1: Blinking LED

```
static void LED_HeartBeat(void *p) {
  (void)p;
  LED1_Neg();
  TRG_SetTrigger(TRG_LED_BLINK,
    1000/TRG_TICKS_MS, LED_HeartBeat, NULL);
}
```

1000ms, the operating system counts ticks not a time

how to set the trigger

```
TRG_SetTrigger(TRG_LED_BLINK,
      1, LED_HeartBeat, NULL);
```

blink count from now, by the next timerinterrupt it will call the LED_HeartBeat function

Example 2: Blinking 2 LEDs

```
static void LED_Blink (void *p) {
  if (*((uint8_t*)p)==0) {
    LED1_Neg();
    (*(uint8_t*)p)++;
  } else if (*((uint8_t*)p)==1) {
    LED2_Neg();
    (*(uint8_t*)p)=0;
  }
  TRG_SetTrigger(TRG_LED_BLINK,
    1000/TRG_TICKS_MS, LED_Blink, p);
}
```

alternating blinking LED1 or LED2

```
uint8_t led = 0;
TRG_SetTrigger(TRG_LED_BLINK, 1, LED_Blink, &led);
```

Lucerne University of Applied Sciences and Arts

HOCHSCHULE LUZERN

Technik & Architektur

Sounder Example 1

```
static void Sounder(void *data) {
 uint16_t duration = *((uint16_t*)data);
 if (duration==0) { /* off */
    BUZZER_Off();
  } else {
    BUZZER_On();
    *((uint16_t*)data) = 0;
    TRG_SetTrigger(TRG_SOUNDER, duration, Sounder, data);
  }
}
```

pointer to void: pointer to something, could be int, ...

cast the value ot time

problem: it doesn't work:

```
void foo(void) {
  uint16_t time = 200/TRG_TICK_MS;
  Sounder(&time);
}
```

turns i on for 200ms

Lucerne University of Applied Sciences and Arts
HOCHSCHULE LUZERN
Technik & Architektur

Sounder Example 2

```
static void Sounder(void *data) {
 uint16_t duration = *((uint16_t*)data);
 if (duration==0) { /* off */
     BUZZER_Off();
   } else {
     BUZZER_On();
     *((uint16_t*)data) = 0;
     TRG_SetTrigger(TRG_SOUNDER, duration, Sounder, data);
   }
}

void foo(void) {
   static uint16_t time = 200/TRG_TICK_MS;
   Sounder(&time);
}
```

Sounder Example 3

```
static void Sounder(void *data) {
  /* sizeof(int)==sizeof(void*) */
  uint16_t duration = (uint16_t)data;
  if (duration==0) { /* off */
    BUZZER_Off();
  } else {
    BUZZER_On();
    TRG_SetTrigger(TRG_SOUNDER, duration, Sounder, 0);
  }
}

void foo(void) {
  Sounder((void*)200/TRG_TICK_MS);
}
```

Lucerne University of Applied Sciences and Arts
HOCHSCHULE LUZERN
Technik & Architektur

Sounder Example 4

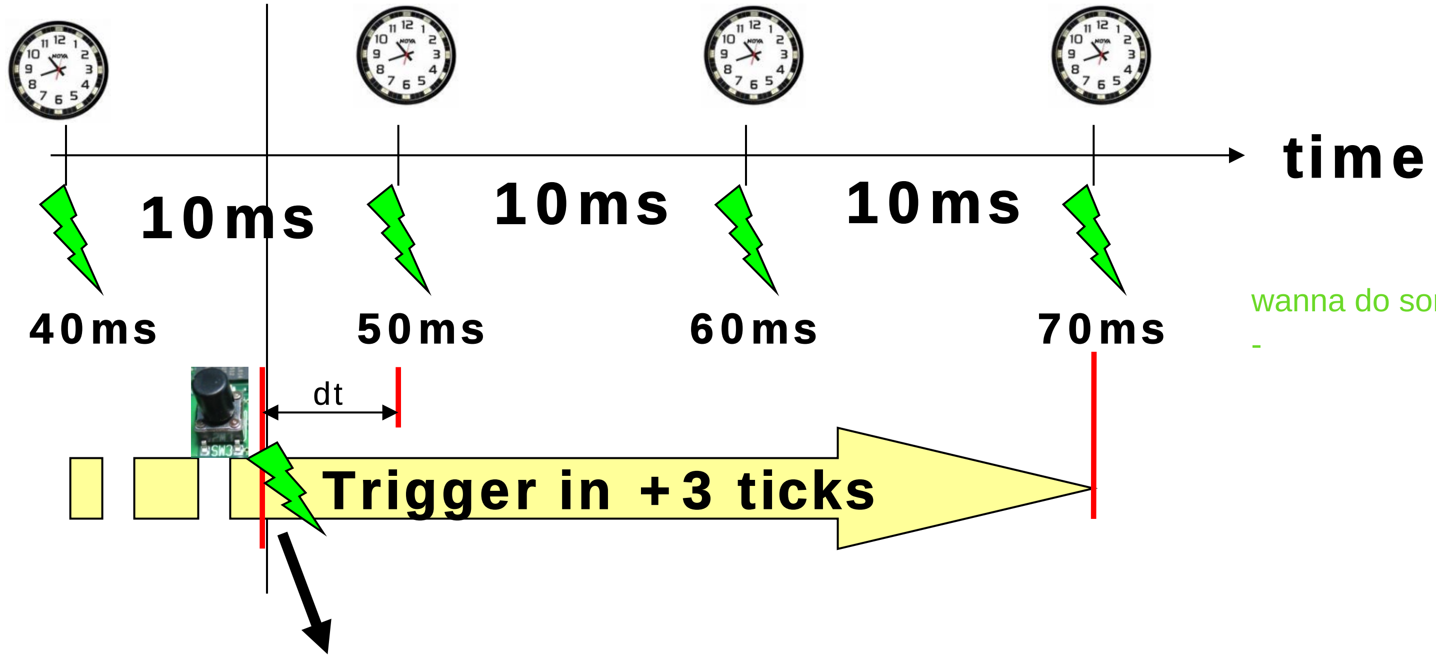
```
static void SoundOff(void *p) {
  BUZZER_Off(); /* turn buzzer off */
}

void Beep(uint16_t ms) {
  BUZZER_On(); /* turn buzzer on */
  TRG_SetTrigger(TRG_BTNSND_OFF,
       ms/TRG_TICKS_MS, SoundOff, 0);
}
```

relative time in ticks, not in ms (we are counting the ticks)

NULL -> void(*) 0, would be better

Relative Time Triggers: Delay / Accuracy

- Action for the future
- Relative, delta to current time (#ticks)
- Simplicity vs. Timer Resolution vs. Accuracy



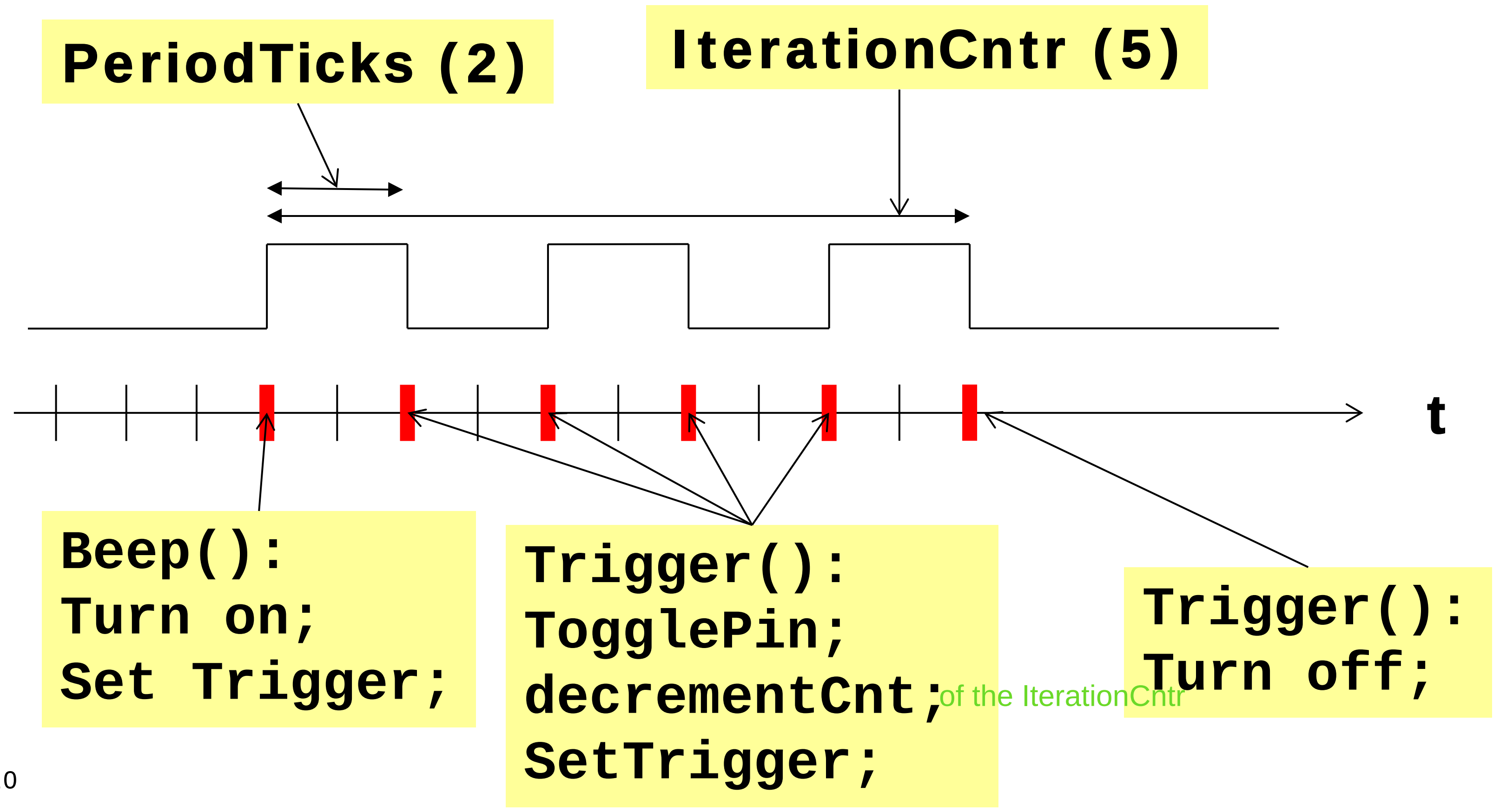
wanna do something at 64ms:
-

```
TRG_SetTrigger(TRG_BTNLED_OFF, 3, BlinkLED, NULL);
```

Bit Banged PWM Buzzer with Trigger

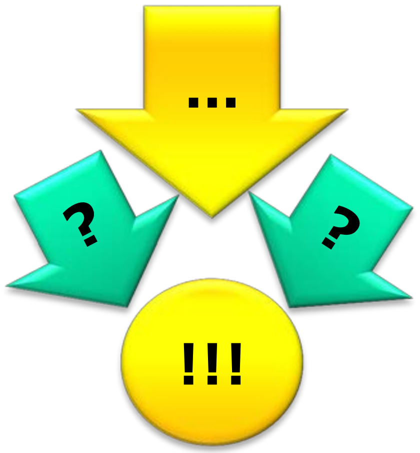
we are going to toggle a pin to generate a PWM signal

- uint8_t **BUZ_Beep(uint16_t freq, uint16_t durationMs)**
- Software PWM



Summary

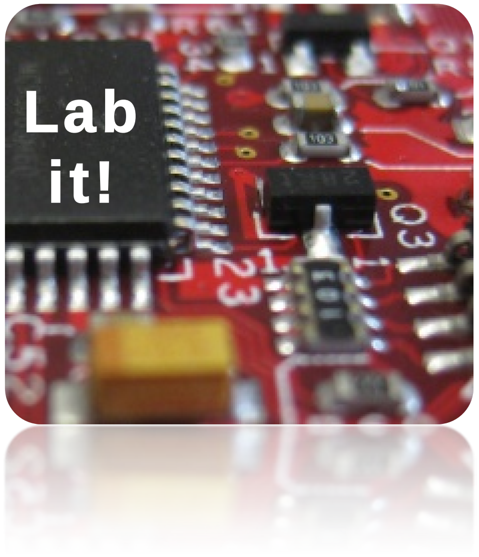
- Using Triggers for time relative callbacks
- Interrupt synchronization
- Function Pointers
- Callbacks
- void pointer arguments
- Data pointer vs. data size
- Pointer to data vs. immediate parameter



Lucerne University of Applied Sciences and Arts
HOCHSCHULE LUZERN
Technik & Architektur

Lab Task: Trigger

- Inspect/understand
  - Trigger.c and Trigger.h
- Implementation
  - Reentrancy
  - Extensibility
    - Adding new trigger(s)
- Buzzer
  - Buzzer.c/Buzzer.h
  - Buzzer usage



for a pause: frequency 0Hz