

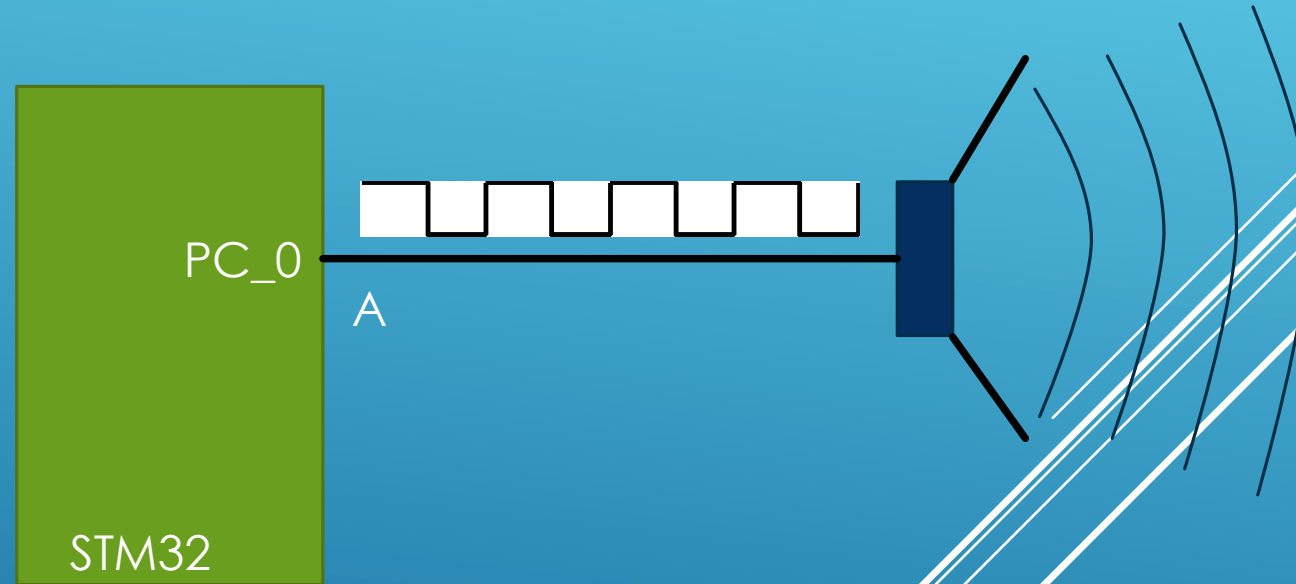
TÖNE ERZEUGEN

Kammerton A 440Hz



TÖNE ERZEUGEN

Prinzip:



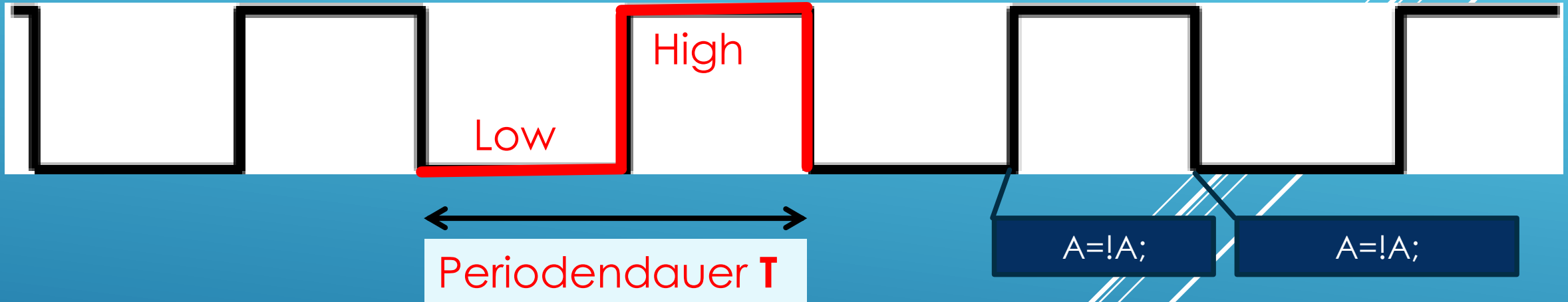
Der Mikrocontroller erzeugt rechteckförmige Spannung an PC_0. Diese wird im Lautsprecher in Schall umgesetzt.



TÖNE ERZEUGEN

Im Mikrocontroller:

Periode



Als Periode bezeichnet man eine Low- und eine High-Phase

Die Rechteckspannung an PC_0 wird erzeugt, indem PC_0 zyklisch komplementiert wird:
1 -> 0 -> 1 -> 0 -> ...



TÖNE ERZEUGEN

Im Mikrocontroller:

1 Sekunde

440 Perioden = Frequenz 440Hz

Die Anzahl der Perioden pro Sekunde bezeichnet man als Frequenz f in Hz (Hertz)

1000 Hz = 1kHz (Kilohertz)

1000000 Hz = 1 MHz (Megahertz)

Der STM32L152 arbeitet mit 32 MHz = 32000000 Hz = 32000000 Perioden pro Sekunde



TÖNE ERZEUGEN

Im Mikrocontroller:

1 Sekunde



440 Perioden = Frequenz 440Hz

Dreisatz:

440 Perioden \equiv 1s

1 Periode \equiv 1s / 440 = 0,00227272s

Formel:

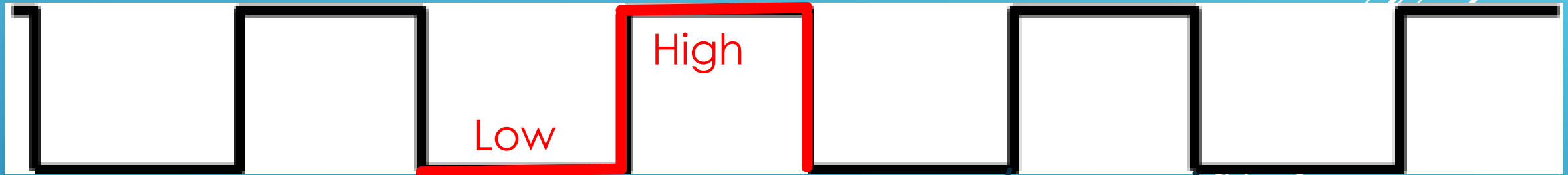
Periodendauer $T = 1/f$



TÖNE ERZEUGEN

Im Mikrocontroller:

Periode



Periodendauer
 $T = 0,00227272s$

A=!A

A=!A

Kurze Zeiten werden oft in ms (Millisekunden)
oder μs (Mikrosekunden) angegeben
 $0,00227272s = 2,27272 ms = 2272,72 \mu s$

$1 s = 1000ms = 1000000\mu s$

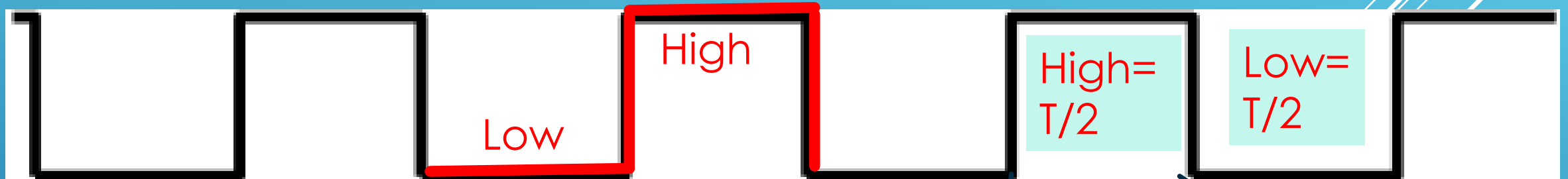
Die Rechteckspannung an PC_0
wird erzeugt, indem PC_0 zyklisch
komplementiert wird:
1 -> 0 -> 1 -> 0 -> ...



TÖNE ERZEUGEN

Im Mikrocontroller:

Periode



Periodendauer
 $T = 0,00227272s$

Kurze Zeiten werden oft in ms (Millisekunden)
oder μs (Mikrosekunden) angegeben
 $0,00227272s = 2,27272 ms = 2272,72 \mu s$
 $1 s = 1000ms = 1000000\mu s$

$A = !A$

$A = !A$

PC_0 muss 2 mal in einer Periode
umgeschaltet (komplementiert)
werden.

$T/2 = 0,001136s = 1,136ms$
 $= 1136 \mu s$

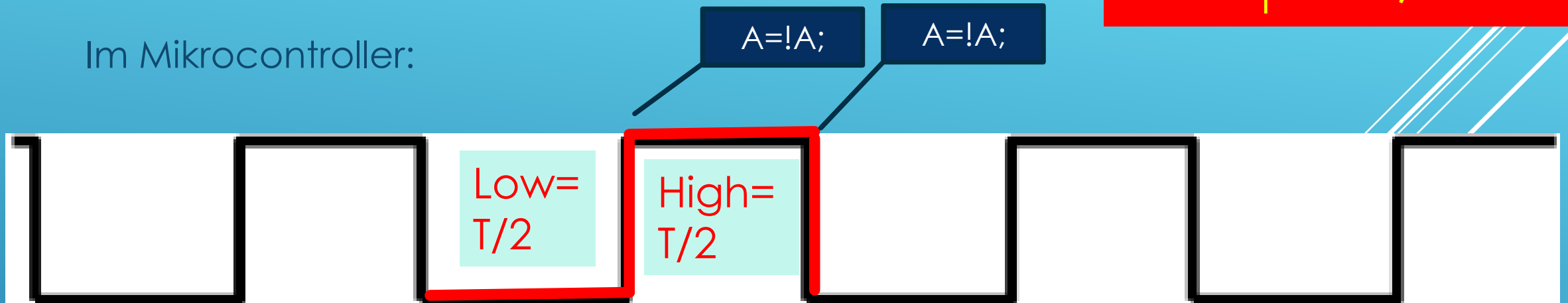


TÖNE ERZEUGEN

Im Mikrocontroller:

$$T/2 = 0,001136s = 1,136ms = 1136 \mu s$$

Prozessortakt:
 $T_p = 31,25ns$



UIE =
TIMx->DIER Bit0

UIF =
TIMx->SR Bit0

TIMx->ARR



16

16

RCC->APB1ENR
Bit 4

TIMx->PSC

Reset
Enable

TIMx->CNT
CTR DIV 16

CEN =
TIMx->CR1 Bit0

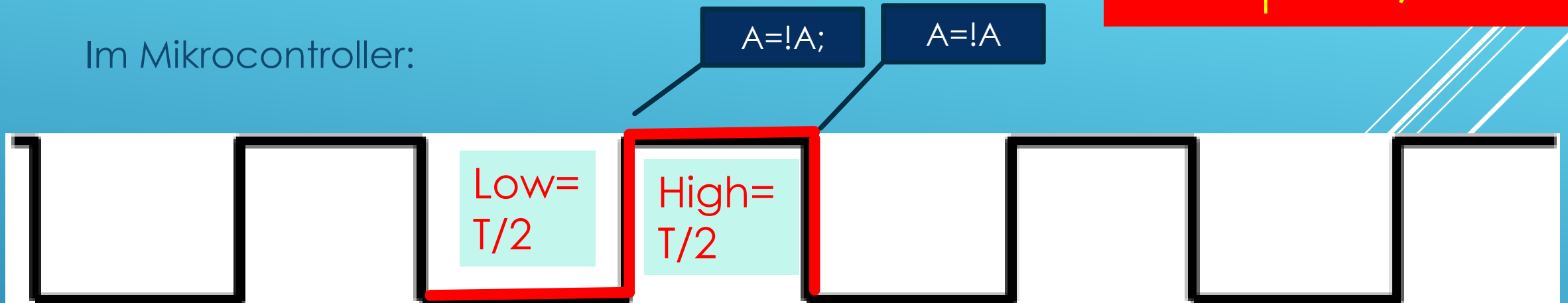
Prozessortakt:
 $T_p = 1/32MHz =$
 $1/32000000Hz =$
 $31,25 ns$



TÖNE ERZEUGEN

Im Mikrocontroller:

Prozessortakt:
 $T_p = 31,25\text{ns}$



$$T/2 = 0,001136\text{s} = 1,136\text{ms} = 1136\text{ }\mu\text{s}$$

TIMx->ARR

16

EQ CM

Reset
Enable

CEN =
TIMx->CR1 Bit0

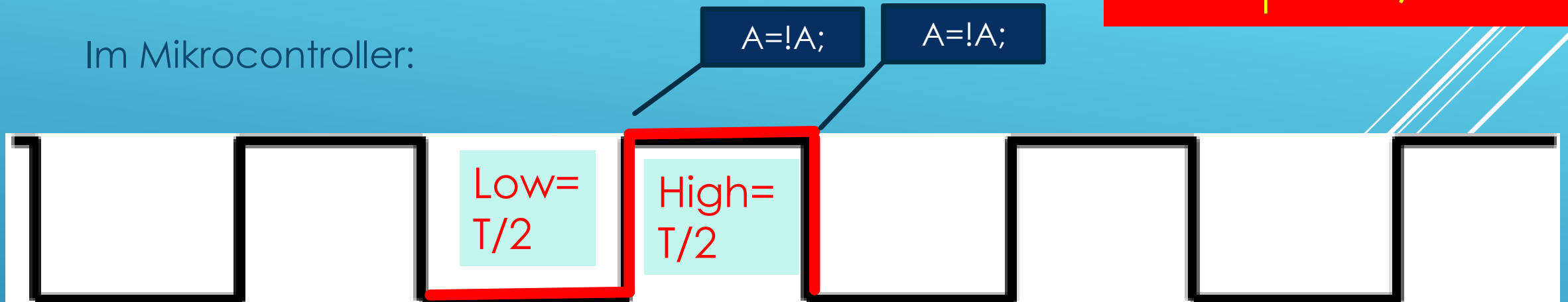
Wie kann der Timer
eine Zeit von $1136\text{ }\mu\text{s}$
erzeugen?



TÖNE ERZEUGEN

Im Mikrocontroller:

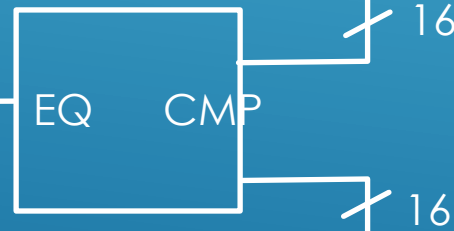
Prozessortakt:
 $T_p = 31,25\text{ns}$



$$T/2 = 0,001136\text{s} = 1,136\text{ms} = 1136\text{ }\mu\text{s}$$

TIMx->ARR

ARR = 1135



$$f_C = f_P / (PSC + 1) = 1\text{MHz}$$
$$T_C = 1 / f_C = 1\text{ }\mu\text{s}$$

$$T_C = 1\text{ }\mu\text{s}$$



RCC->APB1ENR
Bit 4

TIMx->PSC

Prescaler: 31

Reset
Enable

TIMx->CNT
CTR DIV 16

CEN =
TIMx->CR1 Bit0



TÖNE ERZEUGEN

- Zusammenfassung:
- Alle **1136 μ s** muss Port PC_0 komplementiert werden, damit die richtige Frequenz erzeugt wird.
- Dazu muss der Timer alle 1136 μ s einen Interrupt auslösen.



TÖNE ERZEUGEN

Autoreload-Register ARR mit 1135

65535

1136

=470h

0

A=!A;

A=!A;

A=!A;

A=!A;

A=!A;

Interrupt

Interrupt

Interrupt

Interrupt

Interrupt

1136 Takte =
 $1136 * 1\mu s =$
 $1136\mu s$

1136 Takte =
 $1136 * 1\mu s =$
 $1136\mu s$



Periodendauer $T = 2272\mu s = 2,272ms$ $f = 1/T = 440Hz$



TÖNE ERZEUGEN

- Zusammenfassung:
- Alle $1136\mu\text{s}$ muss Port PC_0 komplementiert werden, damit die richtige Frequenz erzeugt wird.
- Dazu muss der Timer alle $1136\mu\text{s}$ einen Interrupt auslösen.
- Deshalb verkürzen wir die Zeit, indem wir das Autoreload-Register ARR von TIM6 mit dem Wert 1135 laden



TÖNE ERZEUGEN

- Rechenbeispiel:
- Gewünschtes Intervall zwischen 2 Überläufen = 25ms
- 25ms sind bei Zählerperiode=1µs: $25000\mu\text{s} / 1\mu\text{s} = 25000$
- Autoreloadwert: ARR= 24999
- TIM6->ARR=24999;
- TIM6->PSC=31;



TÖNE ERZEUGEN

```
void TIM6_Init(void)
{
    RCC->APB1ENR |= 0b10000; //Clock Enable
    TIM6->PSC=31;           //Prescaler 32MHz/32=1MHz
    TIM6->ARR=1135;         //Autoreload 1136* 1µs = 1136µs
    TIM6->DIER=1;           //UIE = 1 (Update Interrupt Enable)
    TIM6->SR=0;             //UIF =0 (Update Interrupt Flag)
    TIM6->CR1=1;            //CEN=1 (Counter Enable)
    /* TIM6_IRQn interrupt configuration */
    NVIC_SetVector(TIM6_IRQn, (uint32_t)&TIM6_IRQHandler);
    HAL_NVIC_EnableIRQ(TIM6_IRQn);
}
```

Freigabe Timer0
Interrupt
(Einzelfreigabe)

ISR für TIM6



TÖNE ERZEUGEN

```
DigitalOut A(PC_0);

void TIM6_IRQHandler(void)
{
    A=!A;
    TIM6->SR=0; //clear Update Interrupt Flag
    HAL_NVIC_ClearPendingIRQ(TIM6_IRQn);
}
```

Aufgabe: Programmieren Sie das Beispiel nach. Testen Sie verschiedene Töne.

