



Timer

Basic Timer TIM6 und TIM7



Ich bin Mik, Dein Mikrocontroller



Timer



Solange die Zitrone im
Körbchen sichtbar ist:
Zähle auf hundert.
Zähle genau jede
Sekunde um 1 weiter.
Wenn Du dort
angekommen bist,
beginne von vorne und
schalte das Licht ein.

Der Auftrag



Timer

Solange die Zitrone im
Körbchen sichtbar ist:
Zähle auf hundert.
Zähle genau jede
Sekunde um 1 weiter.
Wenn Du dort
angekommen bist,
beginne von vorne und
schalte das Licht ein.

Obwohl ich viel weiter
zählen kann soll ich nur bis
100 zählen. Und welche
Zitrone?

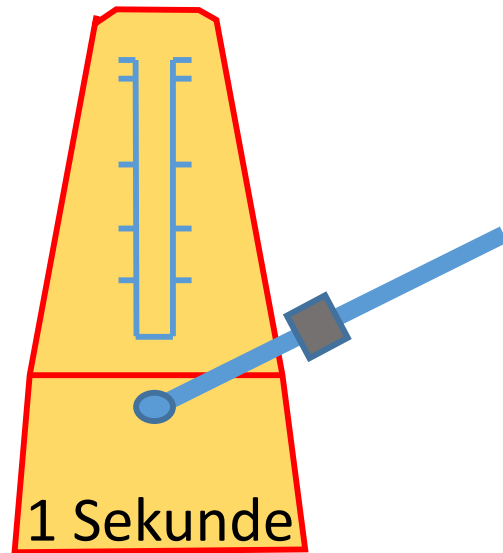


Timer

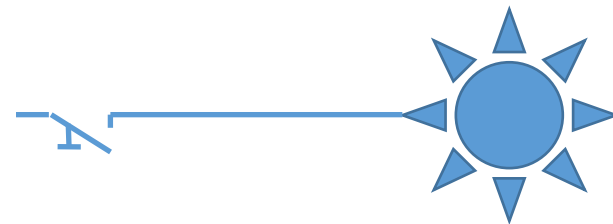


100

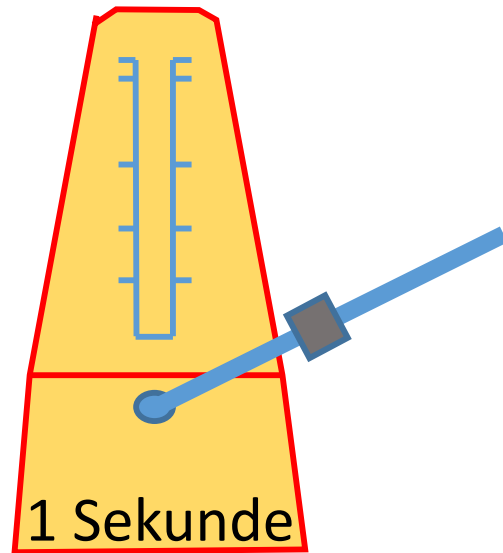
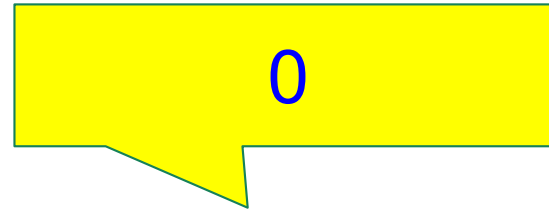
0



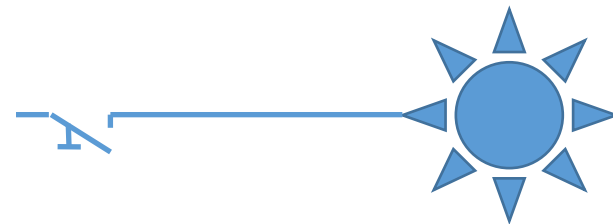
Licht



Timer



Licht

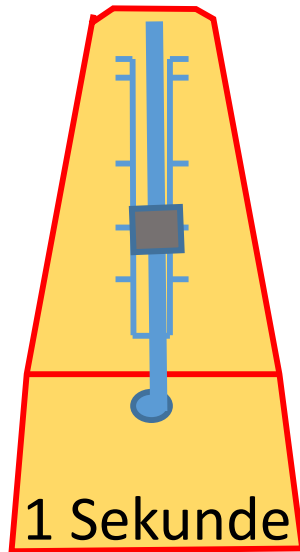


Timer

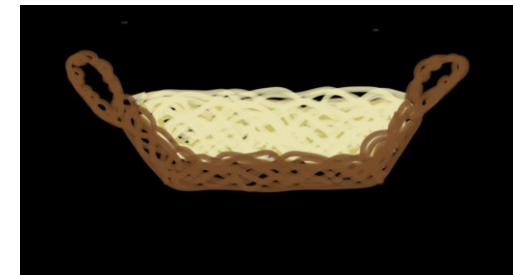
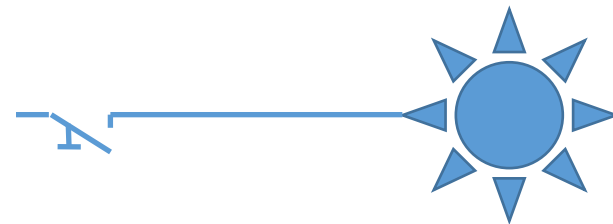


100

0



Licht

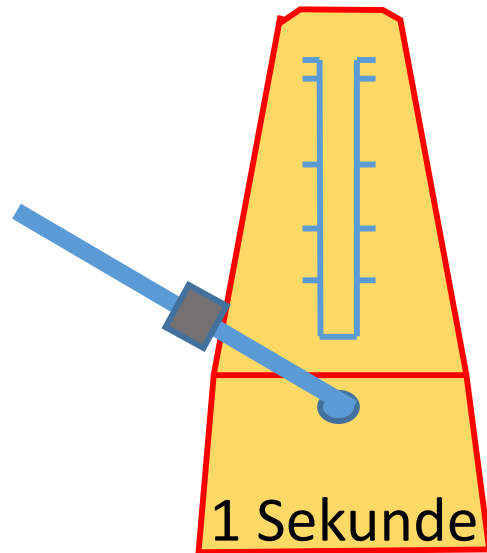


Timer

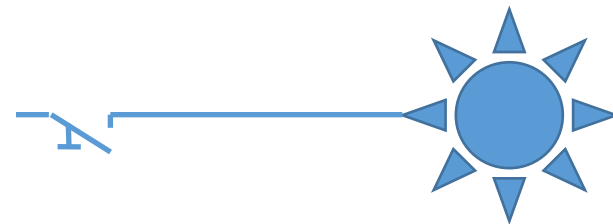


100

0



Licht

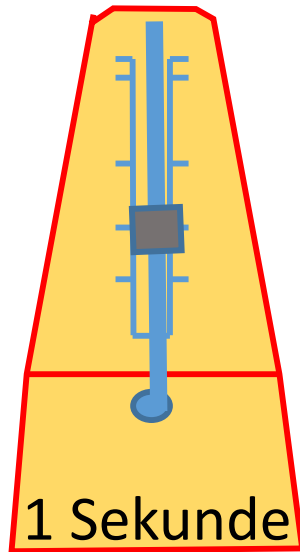


Timer

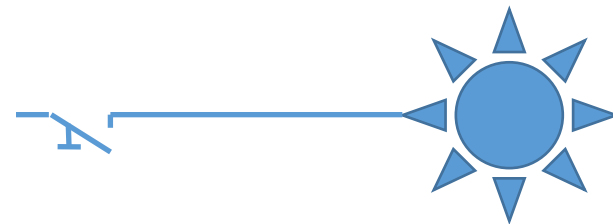


100

0



Licht

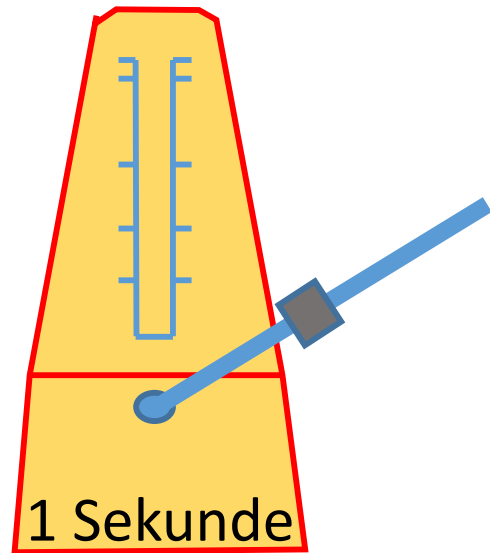


Timer

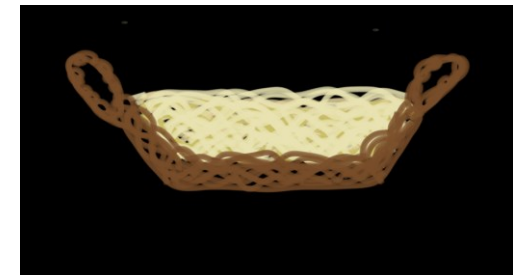
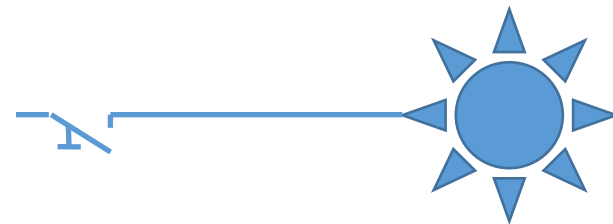


100

0



Licht

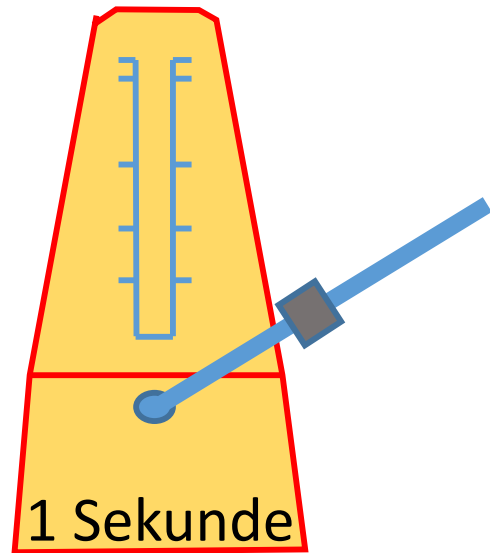


Timer

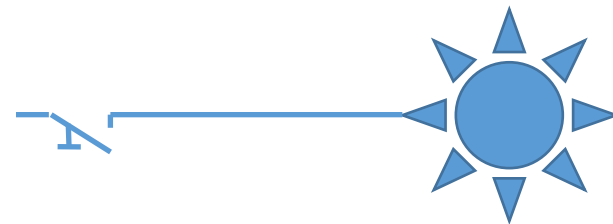


100

0



Licht

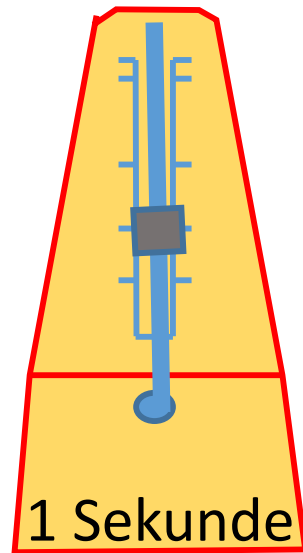


Timer

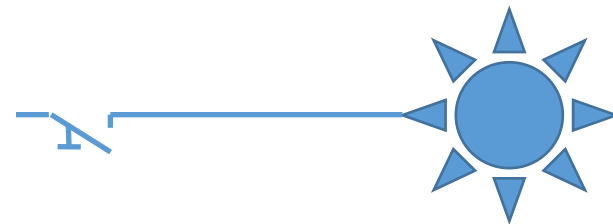


100

1



Licht

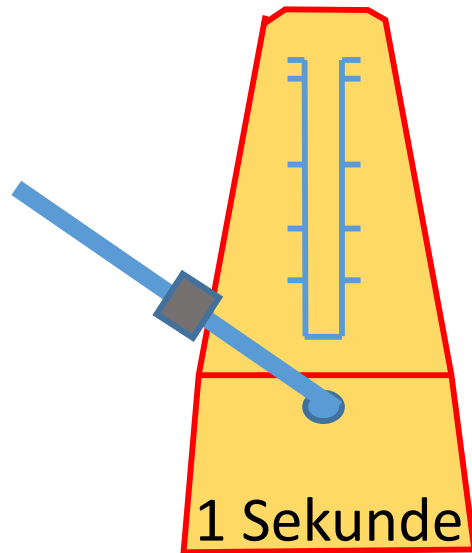


Timer

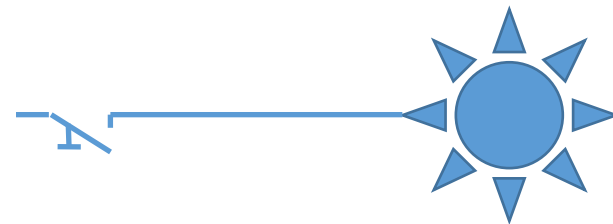


100

1



Licht

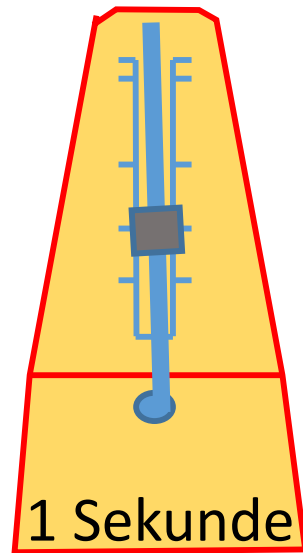


Timer

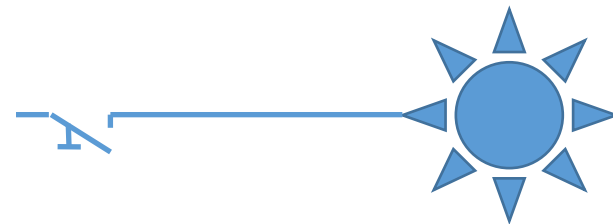


100

1



Licht

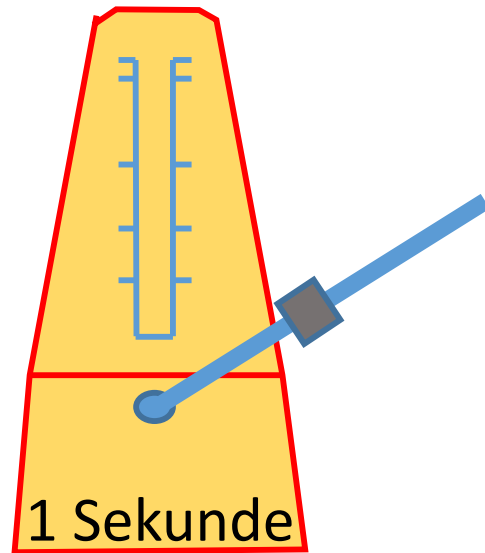


Timer

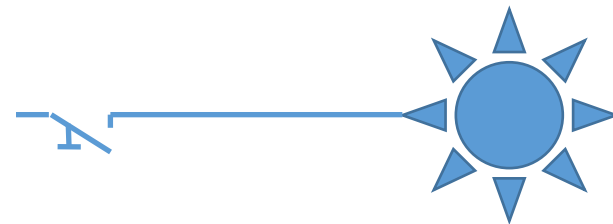


100

1



Licht

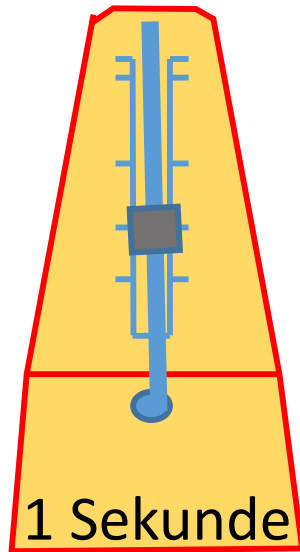


Timer

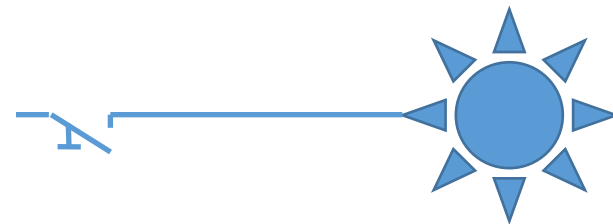


100

2



Licht

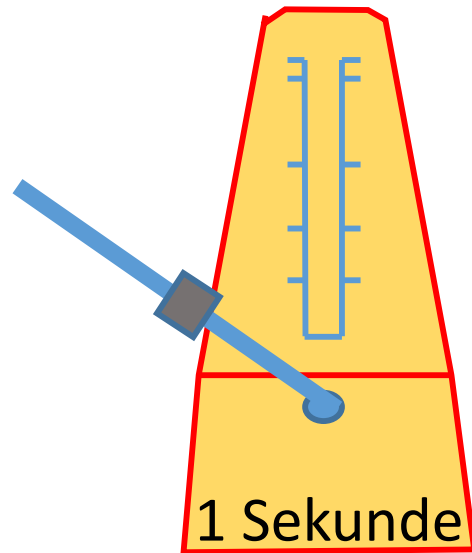


Timer

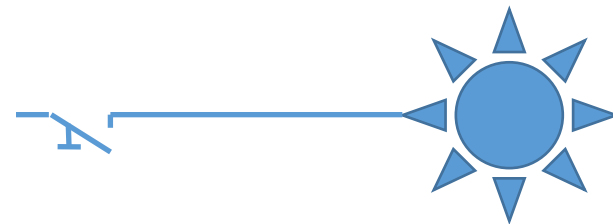


100

2



Licht

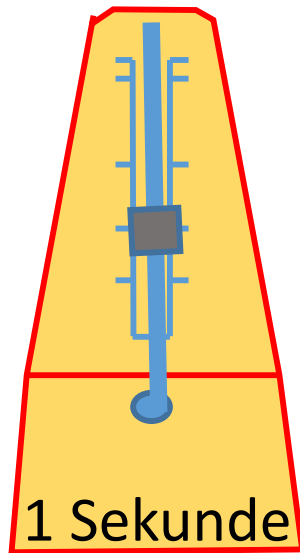


Timer

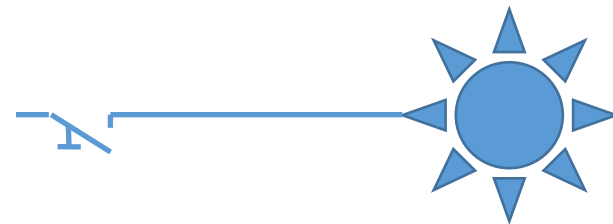


100

2



Licht

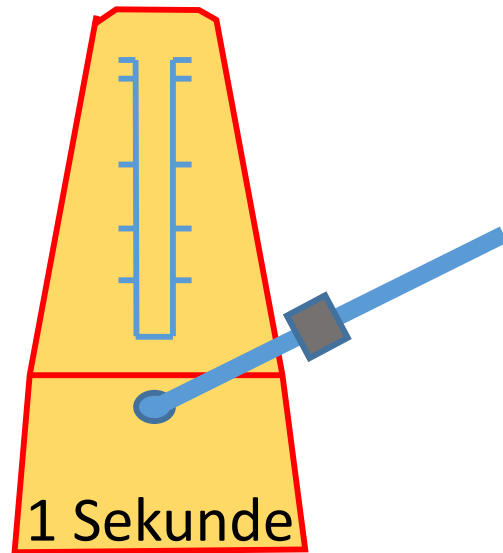


Timer

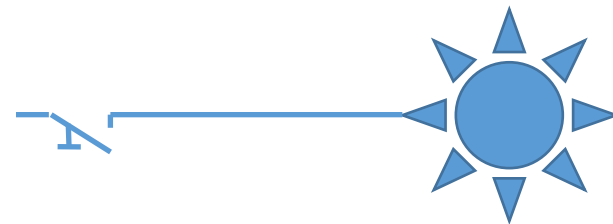


100

2



Licht

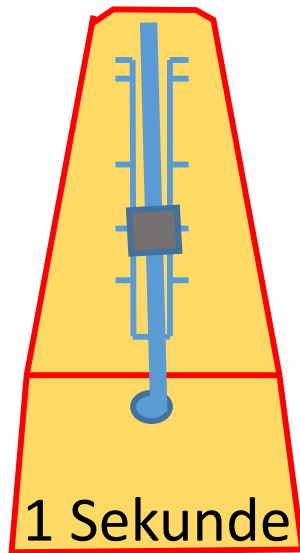


Timer

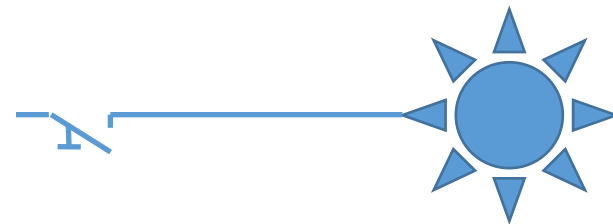


100

3



Licht

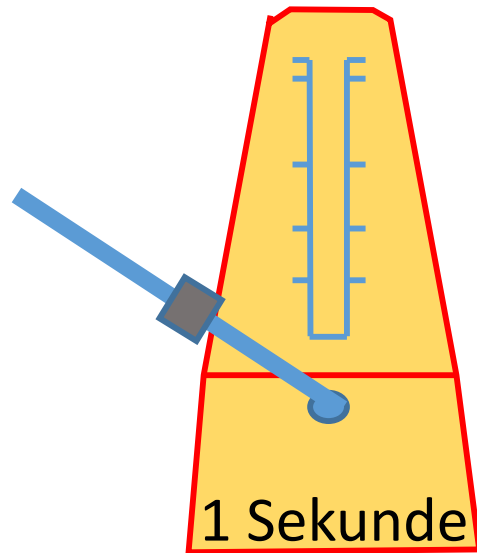


Timer

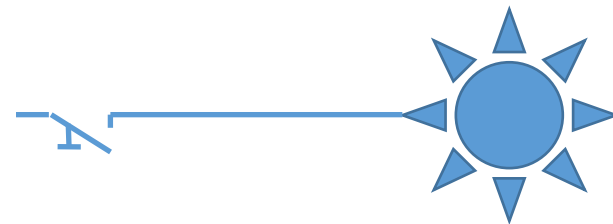


100

3



Licht

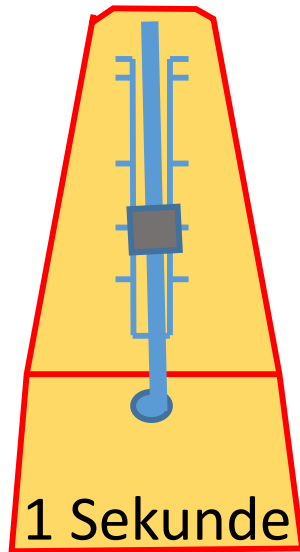


Timer

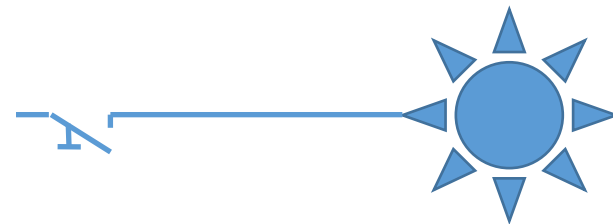


100

3



Licht

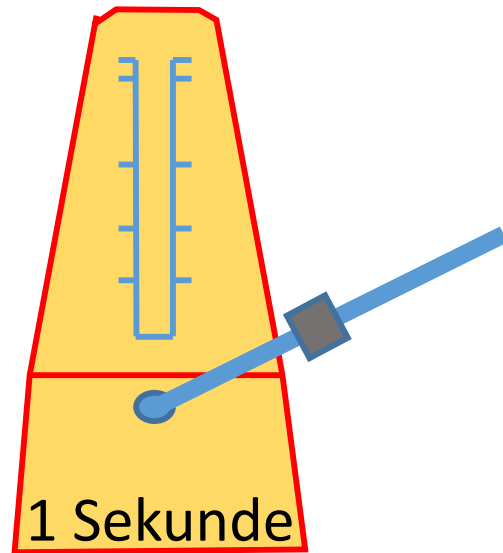


Timer

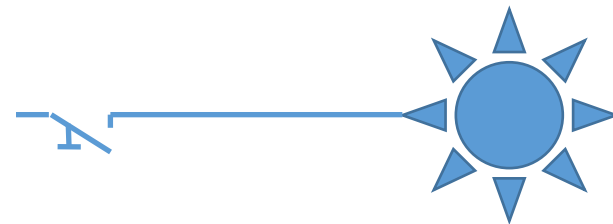


100

3



Licht

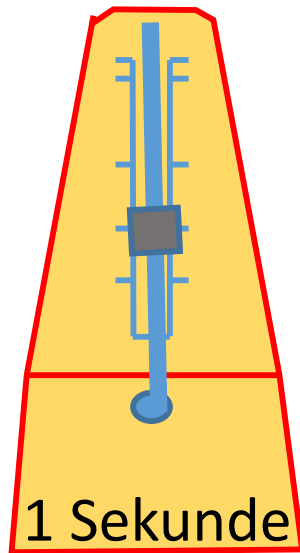


Timer

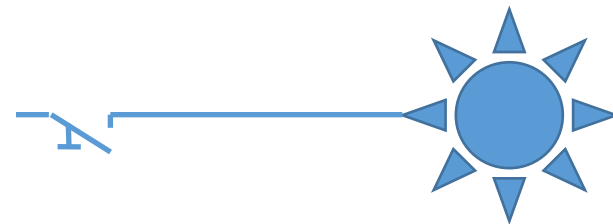


100

4



Licht



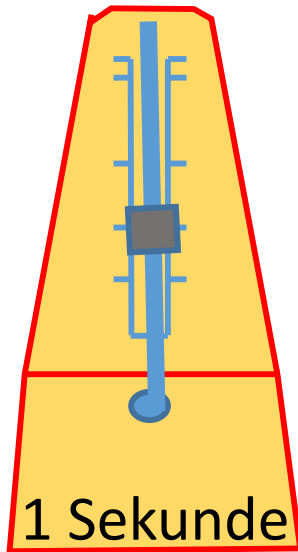
Timer



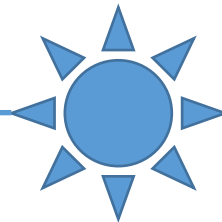
95 Sekunden später

100

99



Licht

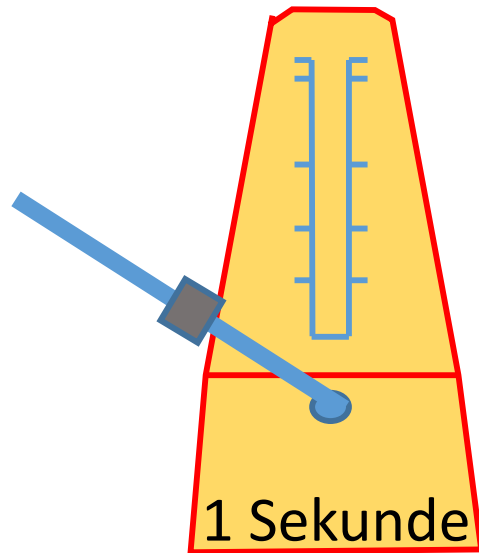


Timer

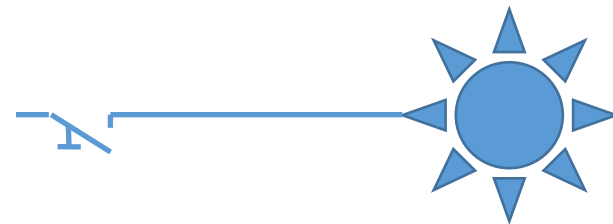


100

99



Licht

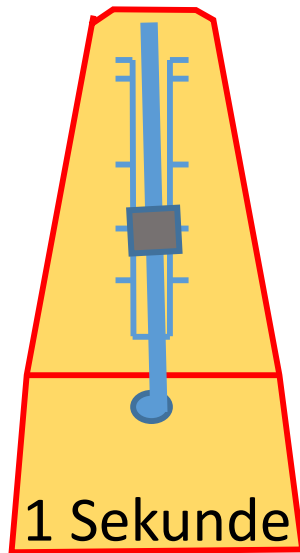


Timer

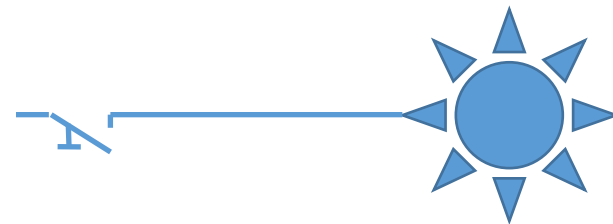


100

99



Licht

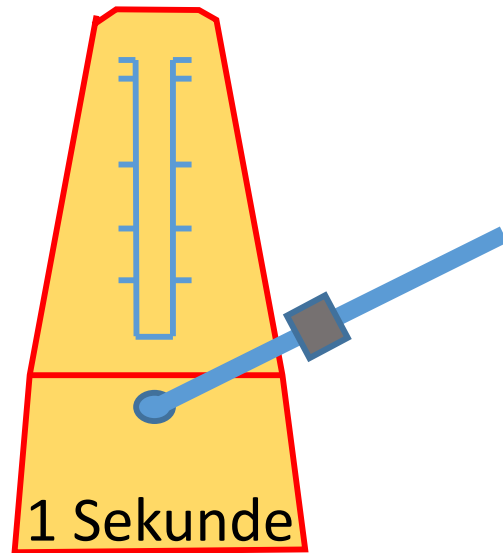


Timer

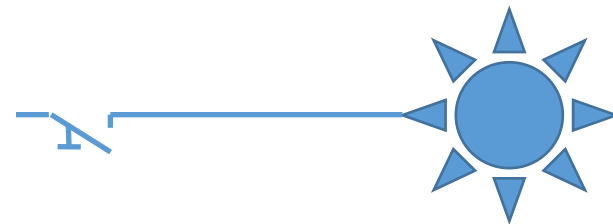


100

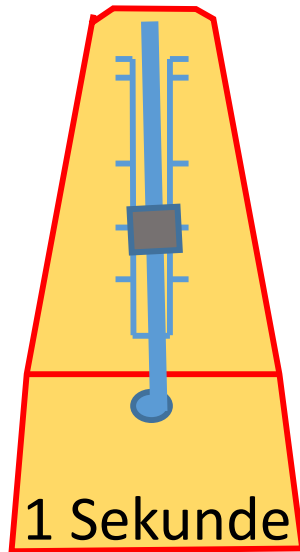
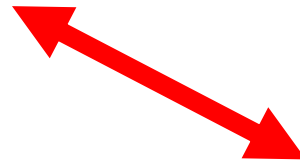
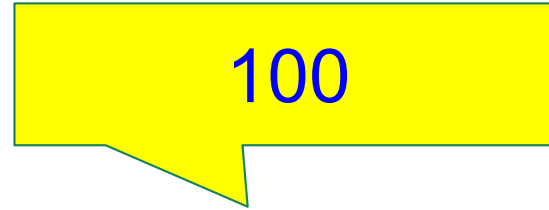
99



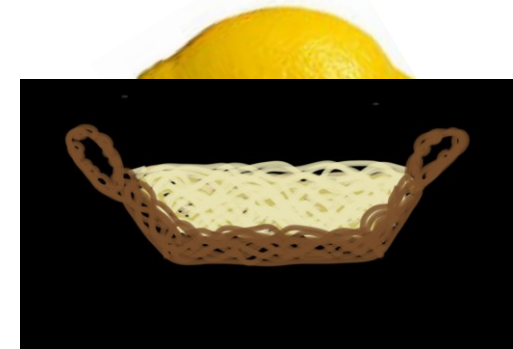
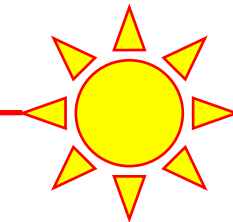
Licht



Timer



Licht

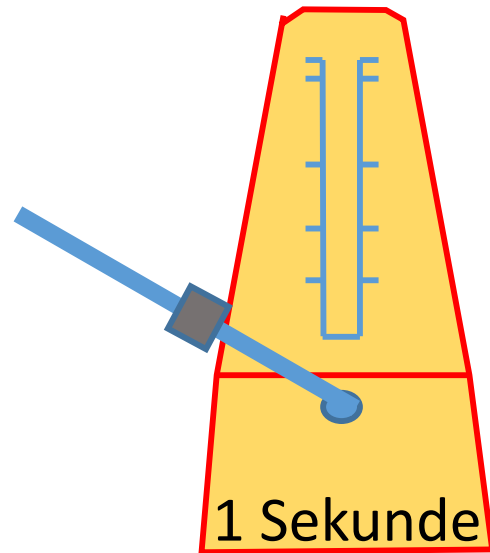


Timer

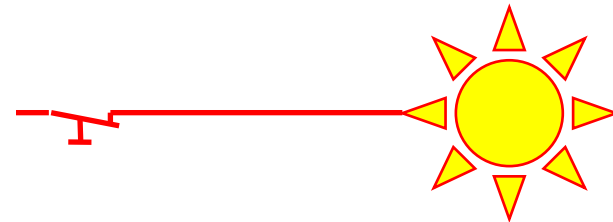


100

0



Licht

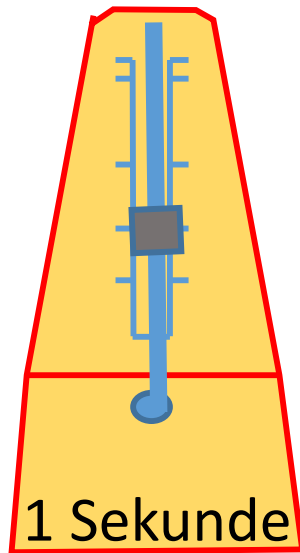


Timer

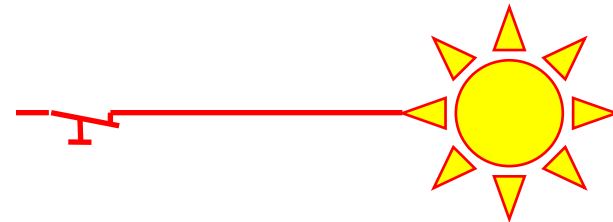


100

0



Licht

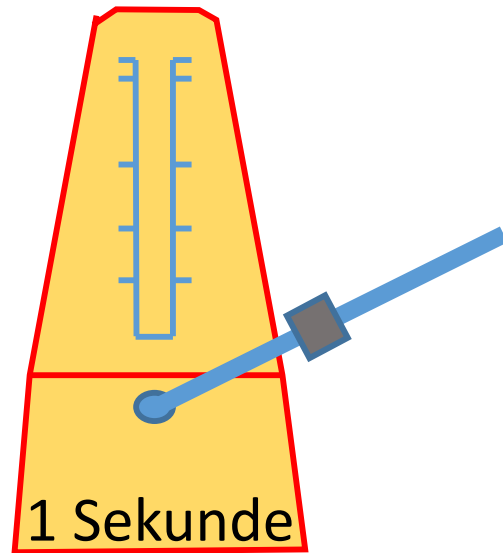


Timer

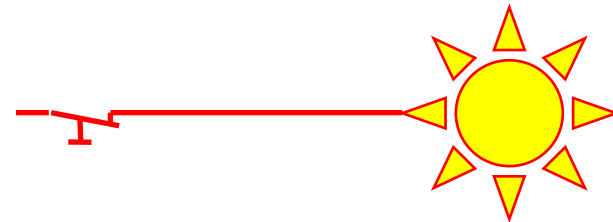


100

0



Licht

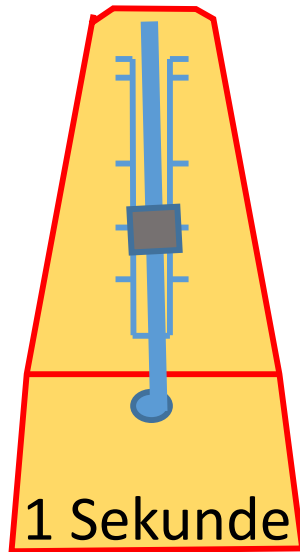


Timer

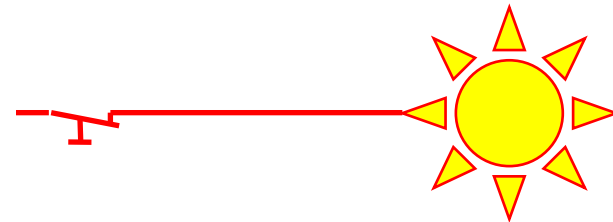


100

1



Licht

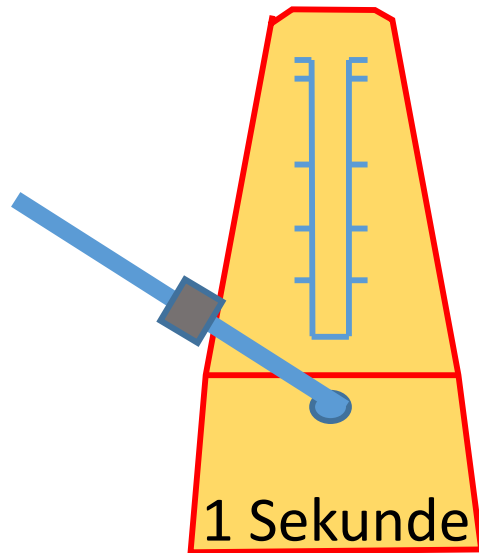


Timer

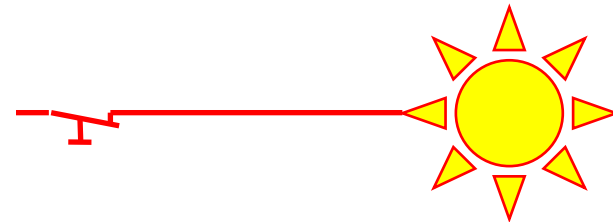


100

1



Licht

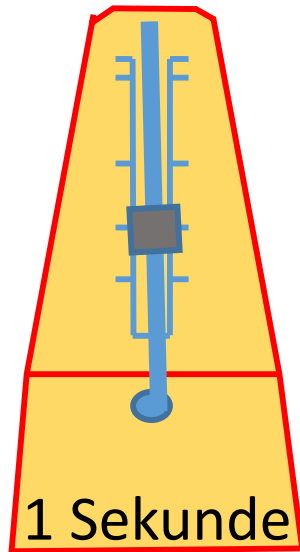


Timer

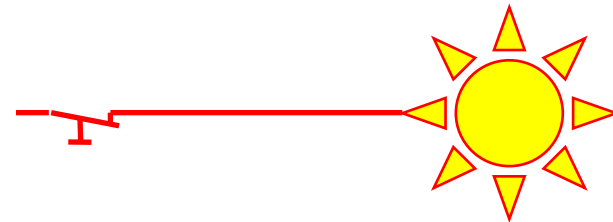


100

1



Licht

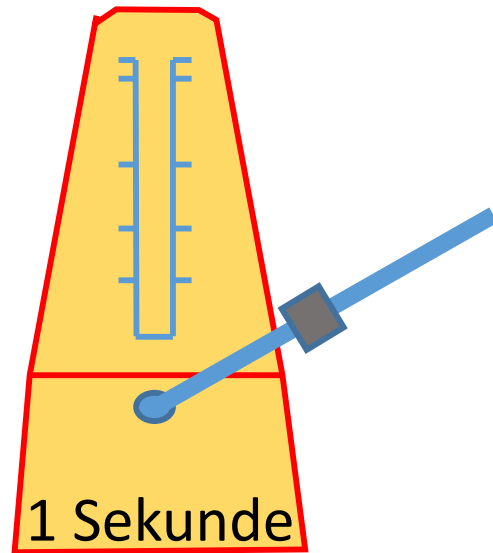


Timer

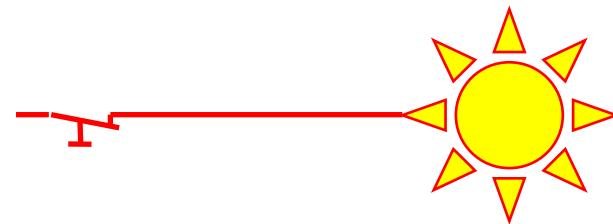


100

1



Licht

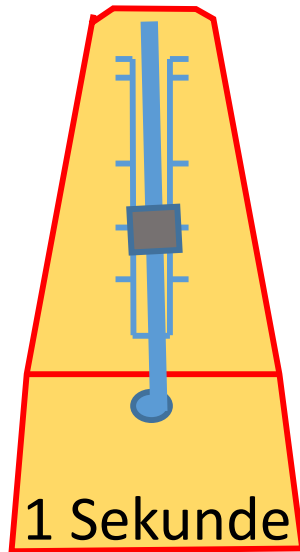


Timer

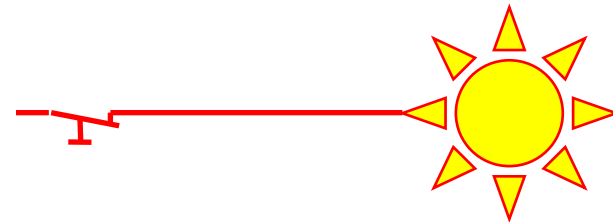


100

2



Licht

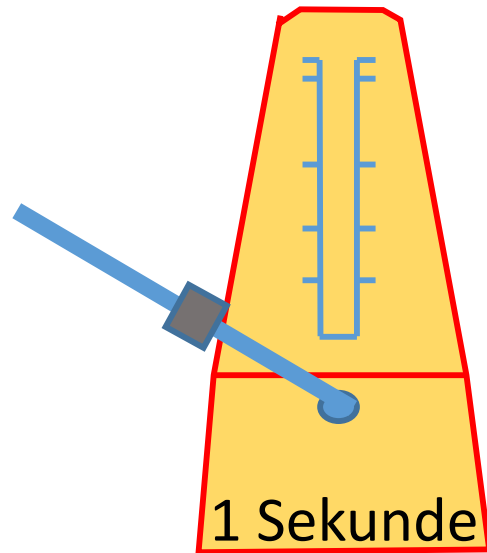


Timer

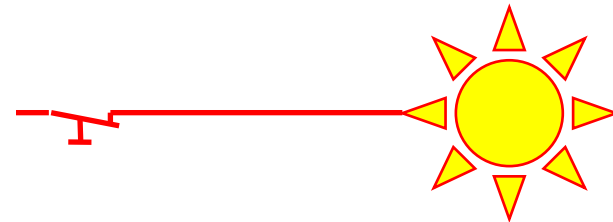


100

2



Licht

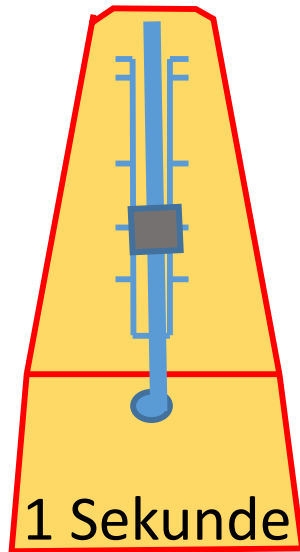


Timer

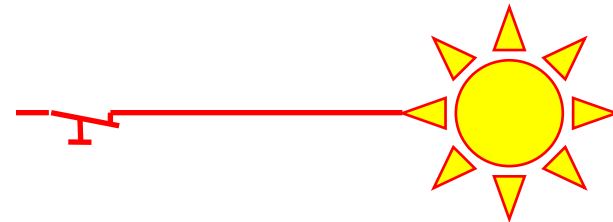


100

2



Licht

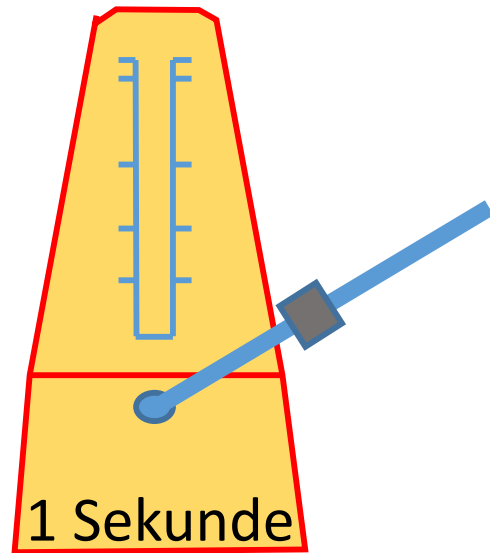


Timer

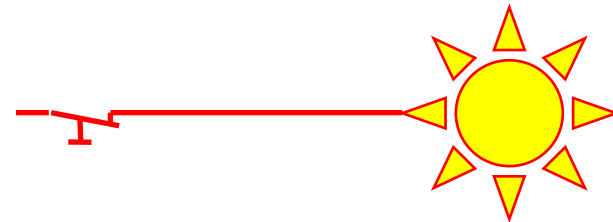


100

2



Licht

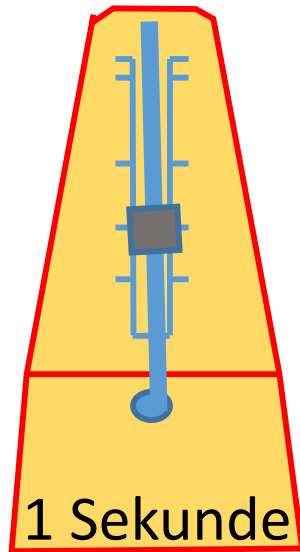


Timer

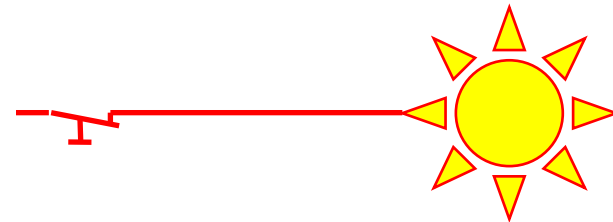


100

3



Licht

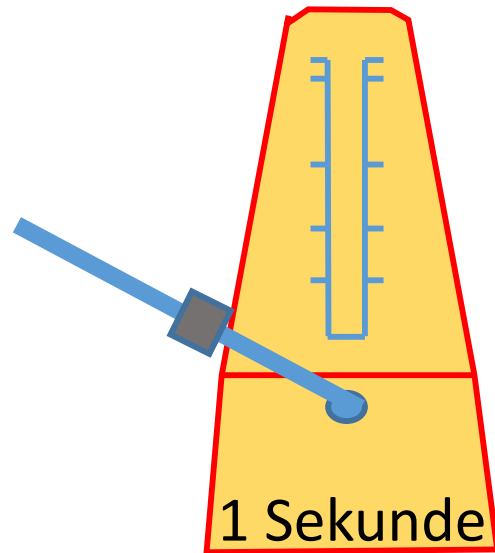


Timer

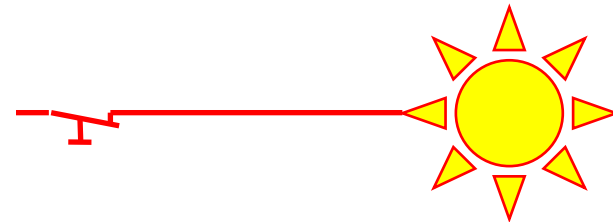


100

3



Licht

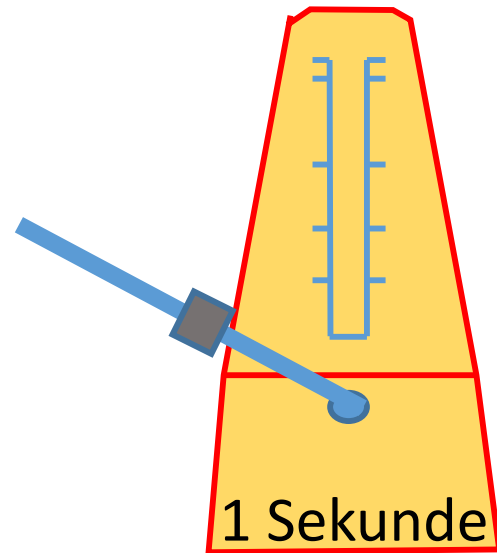


Timer

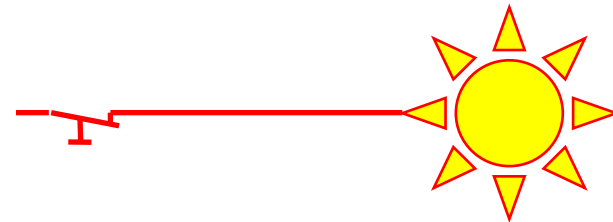


100

3



Licht

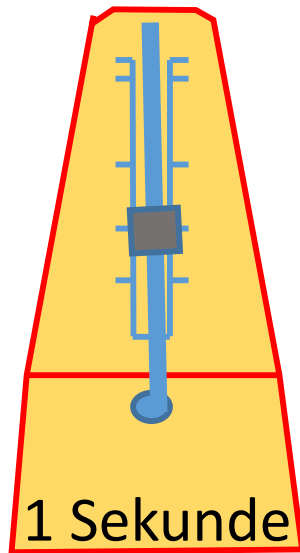


Timer

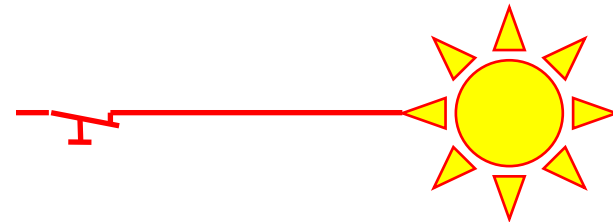


100

3



Licht

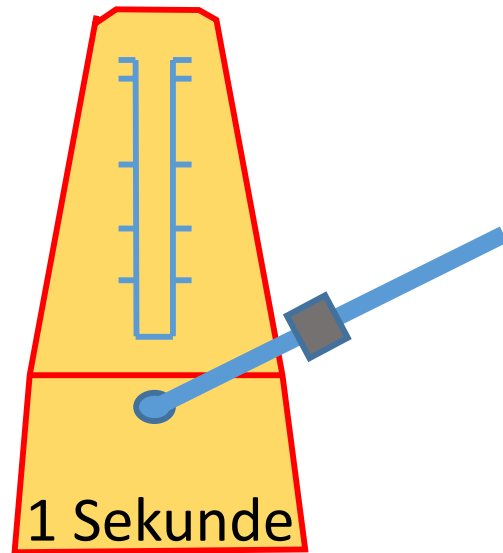


Timer

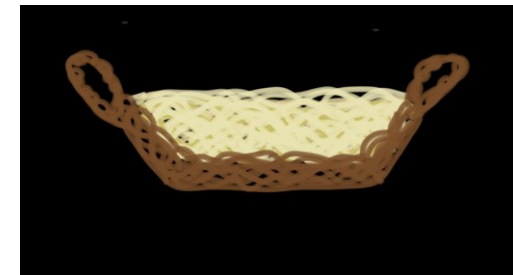
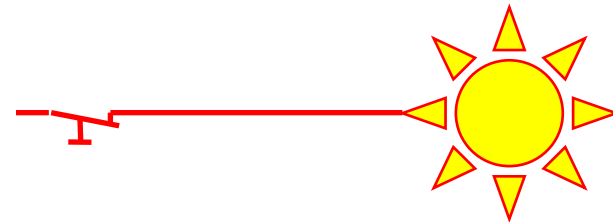


100

3



Licht

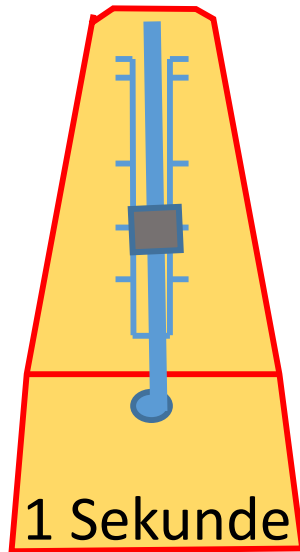


Timer

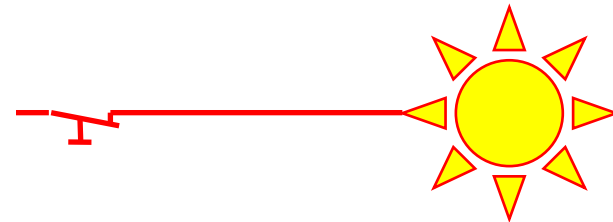


100

3



Licht

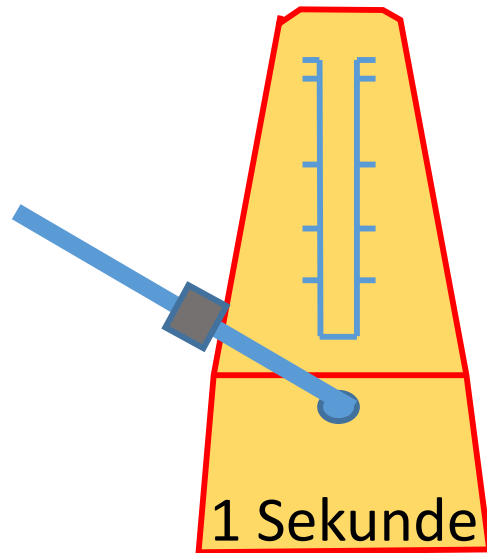


Timer

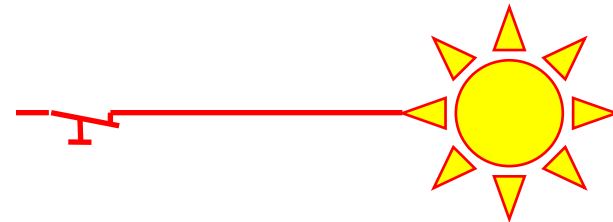


100

3



Licht

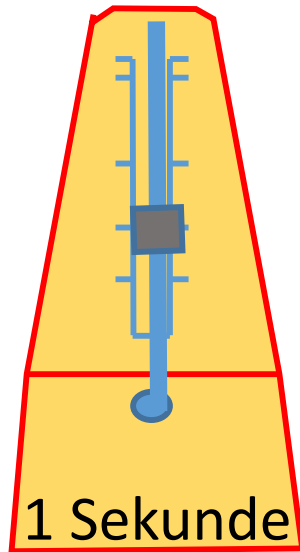


Timer

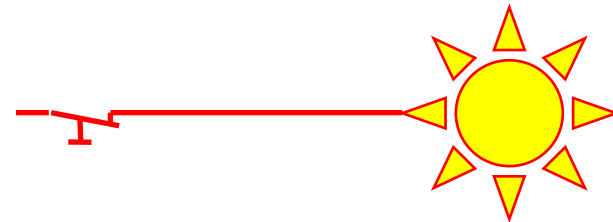


100

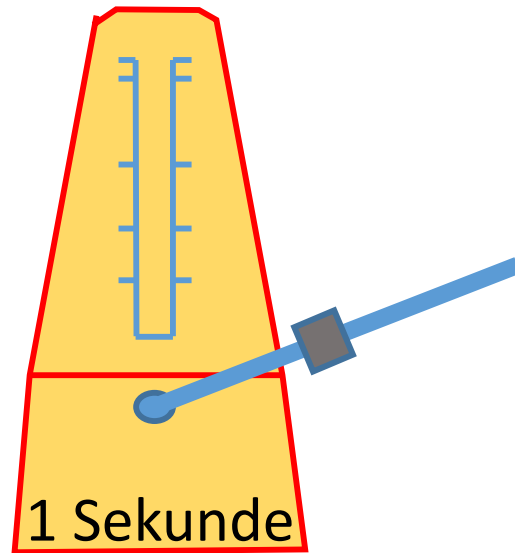
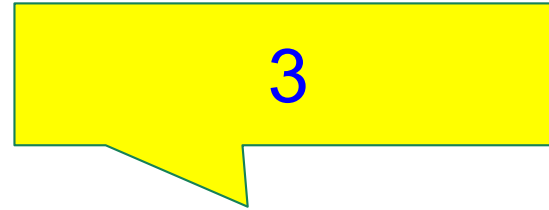
3



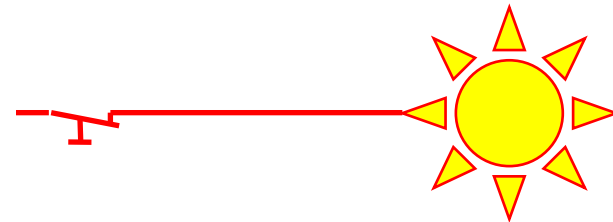
Licht



Timer



Licht

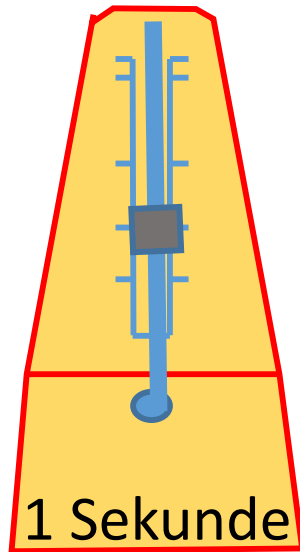


Timer

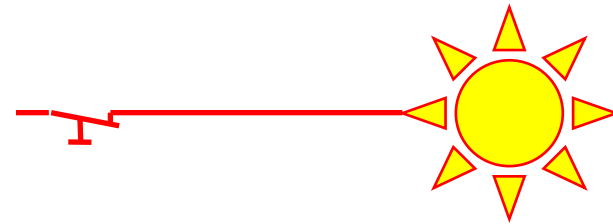


100

3



Licht

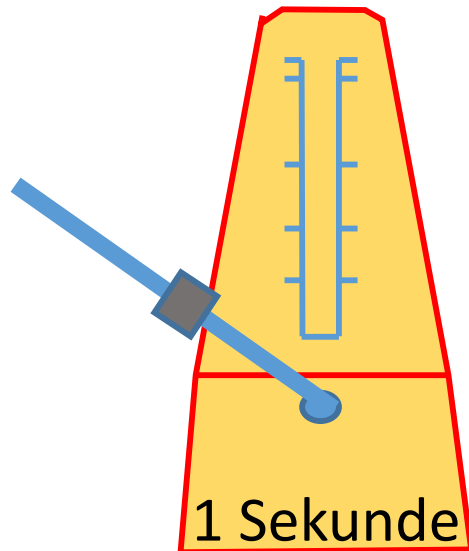


Timer

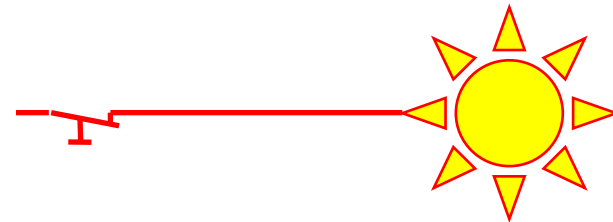


100

3



Licht

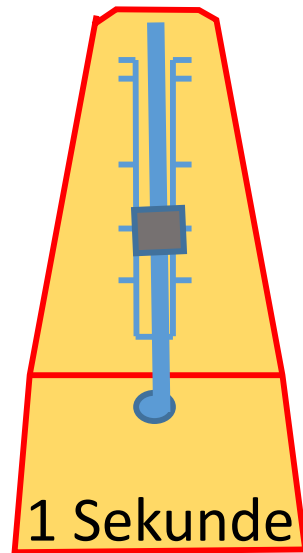


Timer

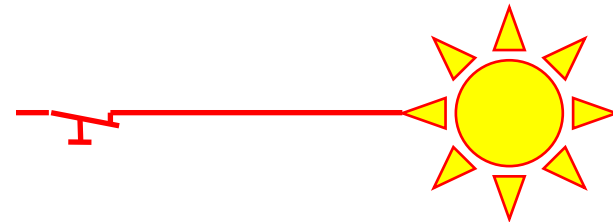


100

3



Licht

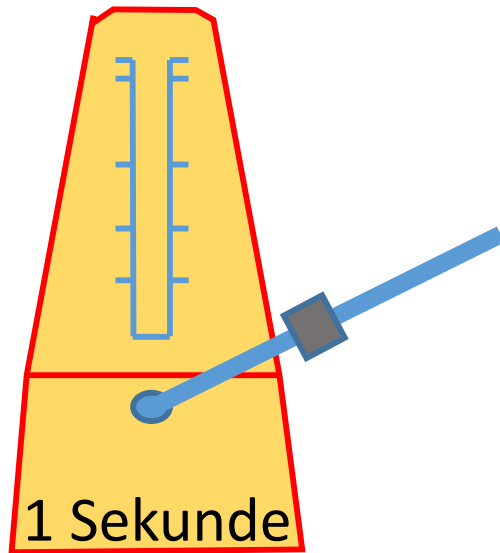


Timer

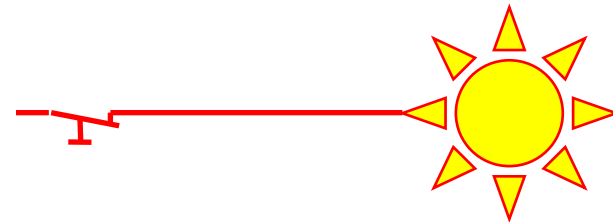


100

3



Licht

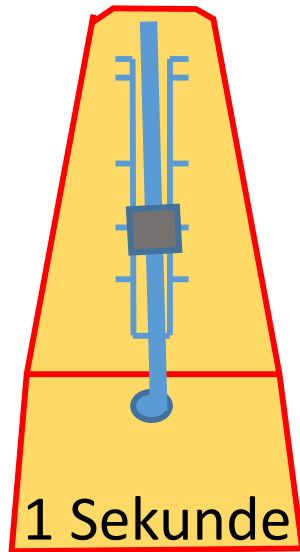


Timer

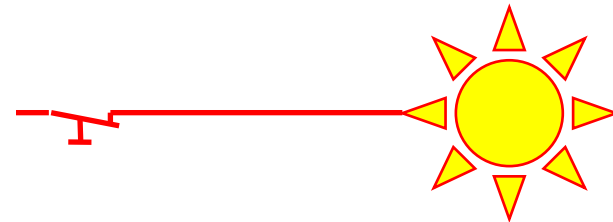


100

4



Licht

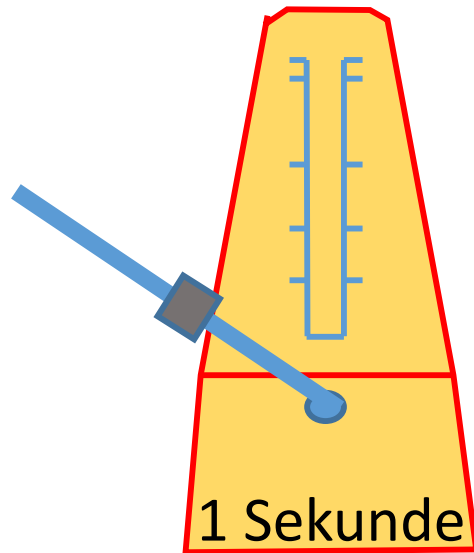


Timer

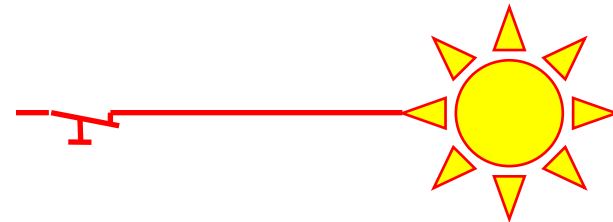


100

4



Licht

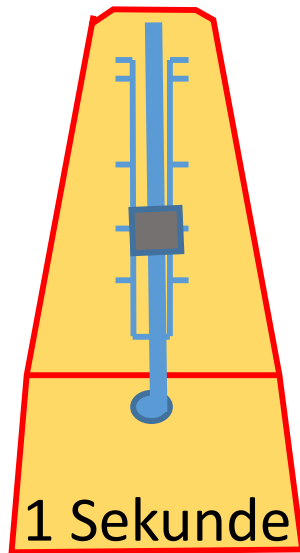


Timer

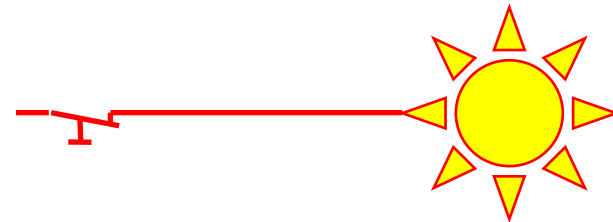


100

4



Licht

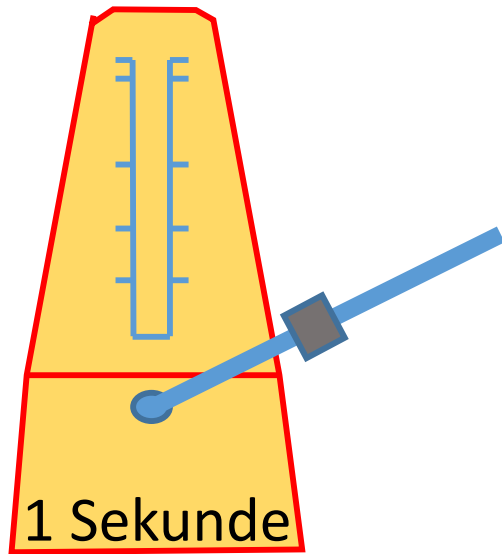


Timer

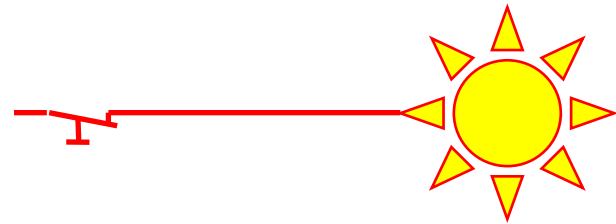


100

4



Licht

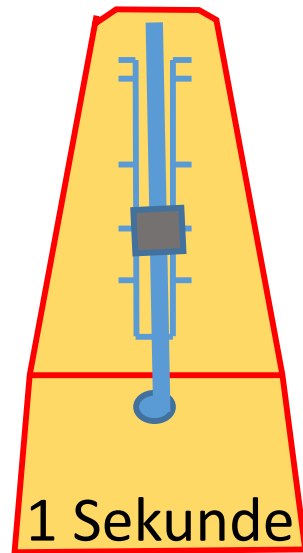


Timer

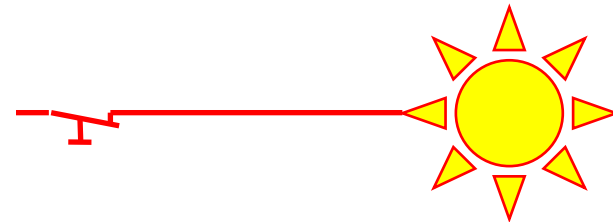


100

5



Licht

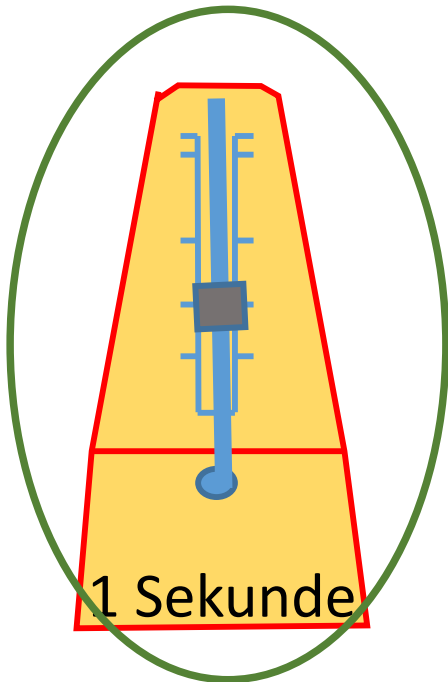


Timer

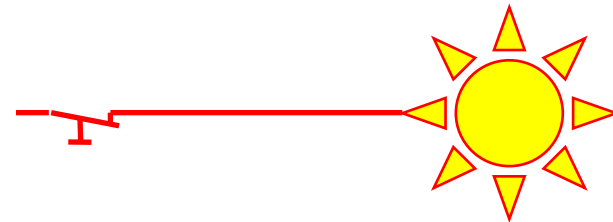


100

5



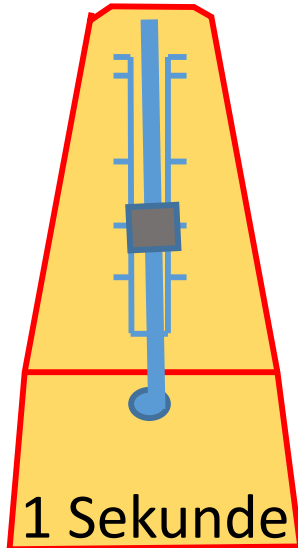
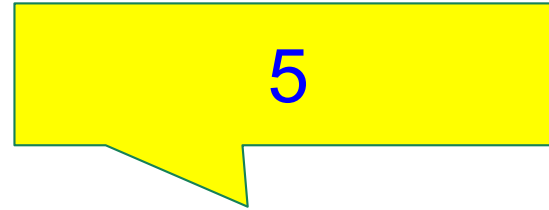
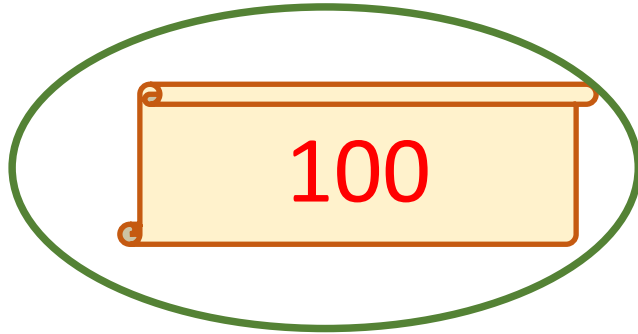
Licht



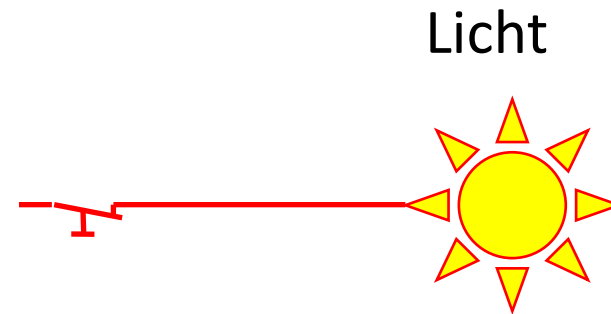
Prescaler PSC => Zählgeschwindigkeit

Timer

Autoreload Register ARR: So weit zählen



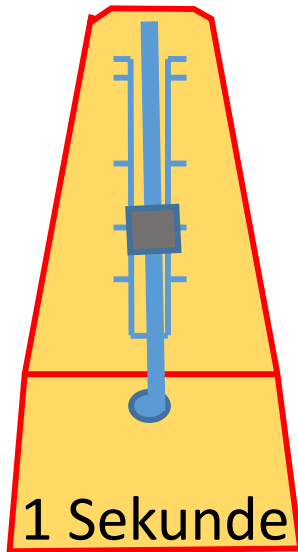
Prescaler PSC



Timer

Autoreload Register ARR

Counter CNT: Der Zähler

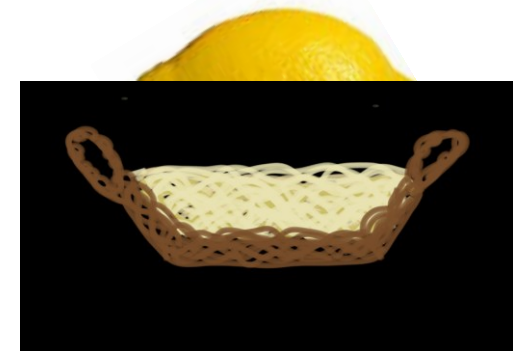
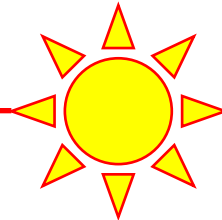


1 Sekunde

Prescaler PSC



Licht

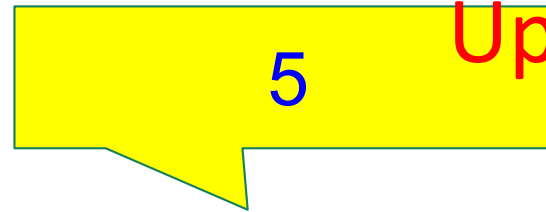


Timer

Autoreload Register ARR

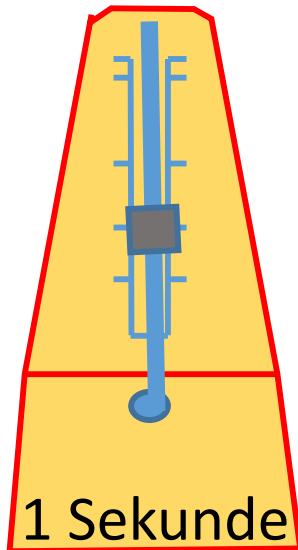


Counter CNT

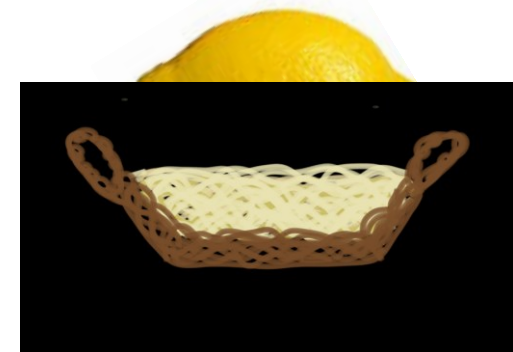
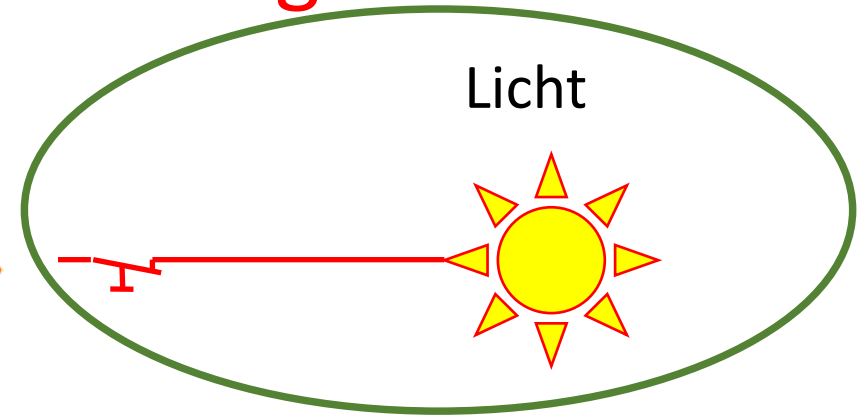


Update Interrupt Flag UIF

Zeigt Überlauf an



Prescaler PSC

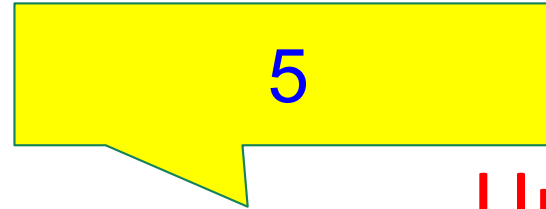


Timer

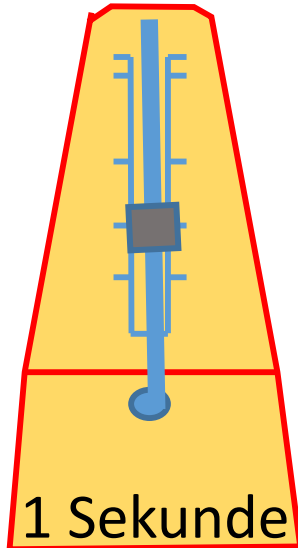
Autoreload Register ARR



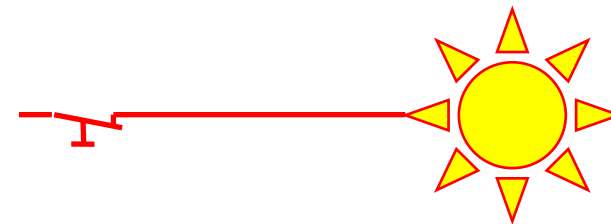
Counter CNT



Update Interrupt Flag UIF
Licht



Prescaler PSC



Counter Enable CEN:



Schaltet zählen ein

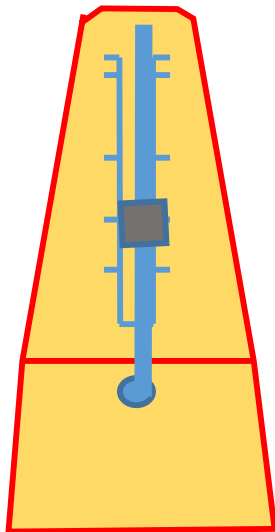


Timer

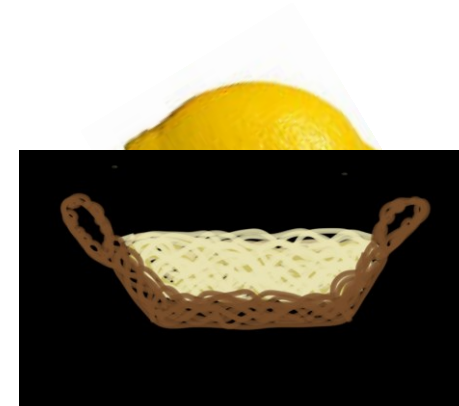
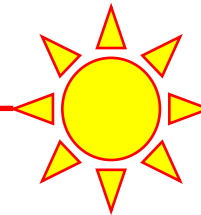


100

5

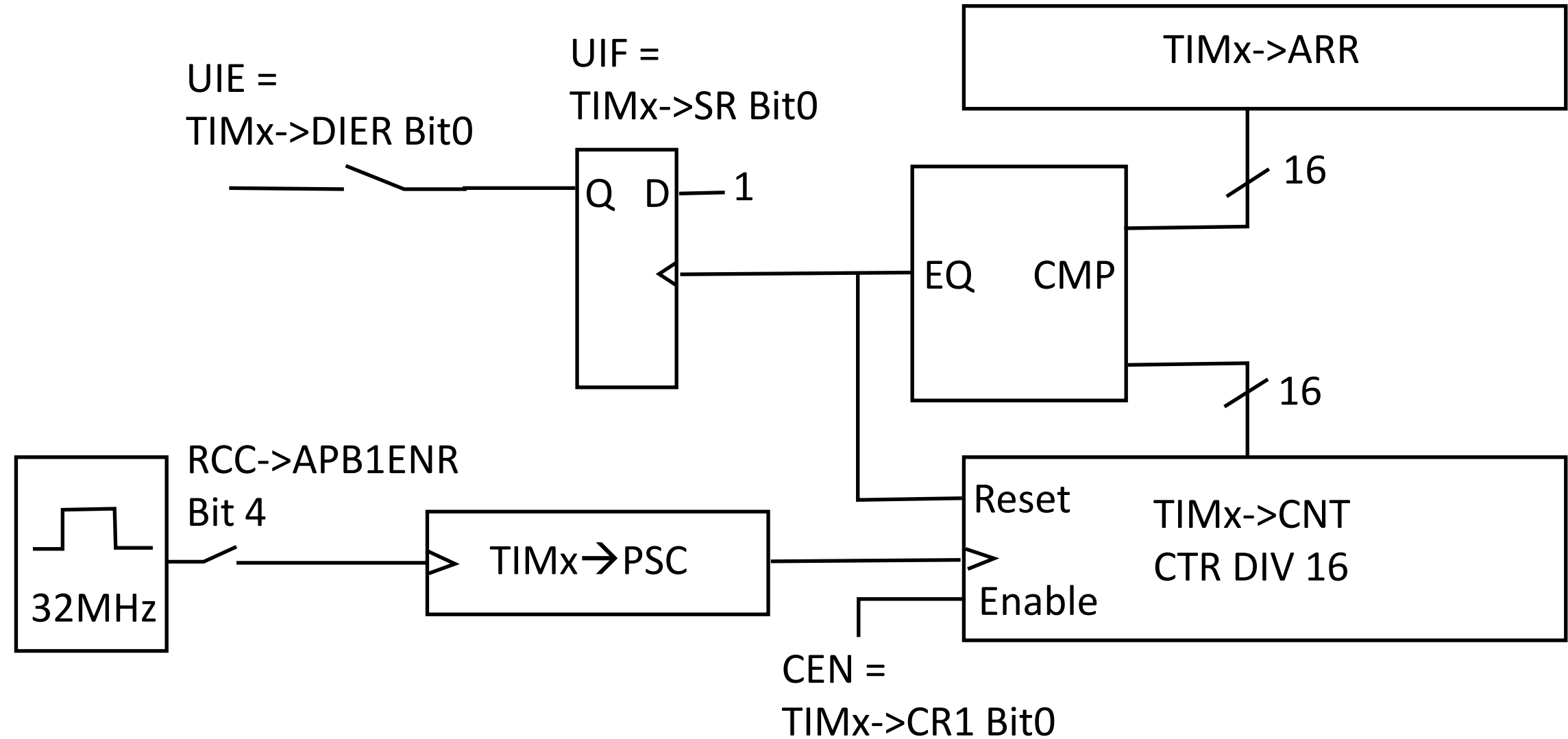


Licht



Basic Timer TIM6 oder TIM7

Timer



A vintage-style alarm clock with a white face and Roman numerals, set against a dark background. The clock has a dark metal frame and two bells on top. The time shown is approximately 10:10. The clock is positioned on the left side of the page, with a dark rectangular shape partially visible below it.



Timer



UIE =
TIMx->DIER Bit0

UIF =
TIMx->SR Bit0

TIMx->ARR

16

EQ

CMP

Autoreloadregister
Für den maximalen Zählwert

RCC->APB1ENR
Bit 4

TIMx->PSC

Reset

TIMx->CNT
CTR DIV 16

Enable

CEN =
TIMx->CR1 Bit0

32MHz

A vintage-style alarm clock with a white face and Roman numerals, set against a dark background. The clock has a dark metal casing and two bells on top. The time shown is approximately 10:10. The clock is positioned on the right side of the page, with its shadow cast onto the dark background.



Timer



UIE =
TIMx->DIER Bit0

UIF =
TIMx->

Q D
<

TIMx->ARR

16

Der Überlauf wird im UIF
Update Interrupt Flag
angezeigt

RCC->APB1ENR
Bit 4

TIMx->PSC

32MHz

Reset

TIMx->CNT
CTR DIV 16

Enable

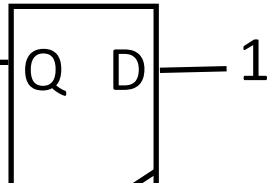
CEN =
TIMx->CR1 Bit0

Timer



UIE =
TIMx->DIER Bit0

UIF =
TIMx->SR Bit0



Mit Counter Enable
CEN=1 wird der Counter
freigegeben

RCC->APB1ENR

Bit 4

TIMx->



EQ

CMP

16

16

Reset

TIMx->CNT
CTR DIV 16

Enable

EN =
IMx->CR1 Bit0

32MHz

Timer



UIE =
TIMx->DIER Bit0

UIF =
TIMx->SR Bit0

Der Prescaler stellt die
Zählgeschwindigkeit ein:
 $31 = 1\mu s$
 $31999 = 1ms$

TIMx->ARR

EQ CMP

16

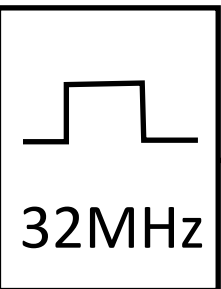
16

Reset

TIMx->CNT
CTR DIV 16

Enable

CEN =
TIMx->CR1 Bit0



Rec > Arr
Bit 1



TIMx->PSC

Timer



UIE =

TIMx->D

Der Timer muss mit 32MHz
Takt versorgt werden um
arbeiten zu können

TIMx->ARR

16

MP

16

RCC->APB1ENR

Bit

32MHz

TIMx->PSC

Reset

TIMx->CNT

CTR DIV 16

Enable

CEN =

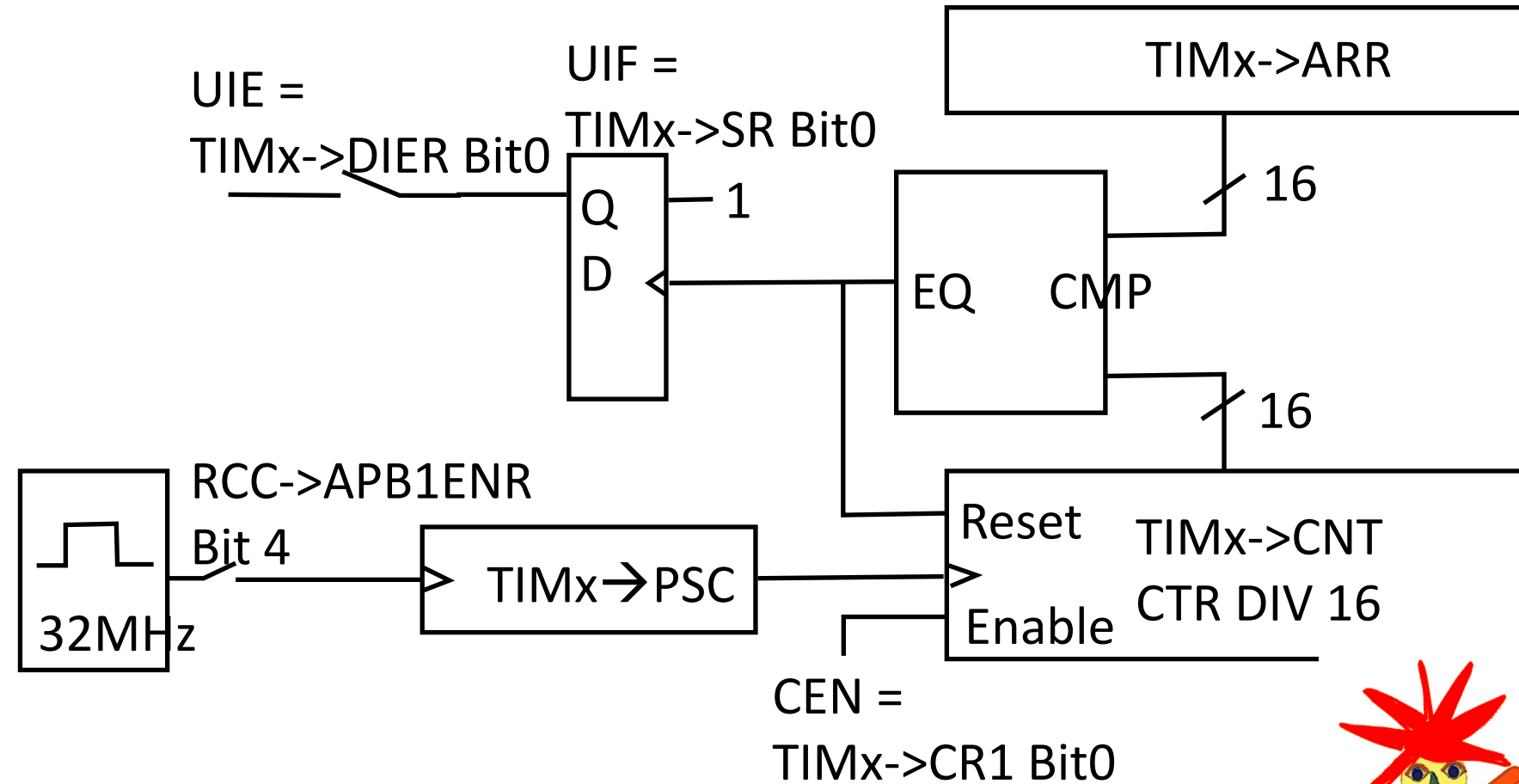
TIMx->CR1 Bit0



Timer

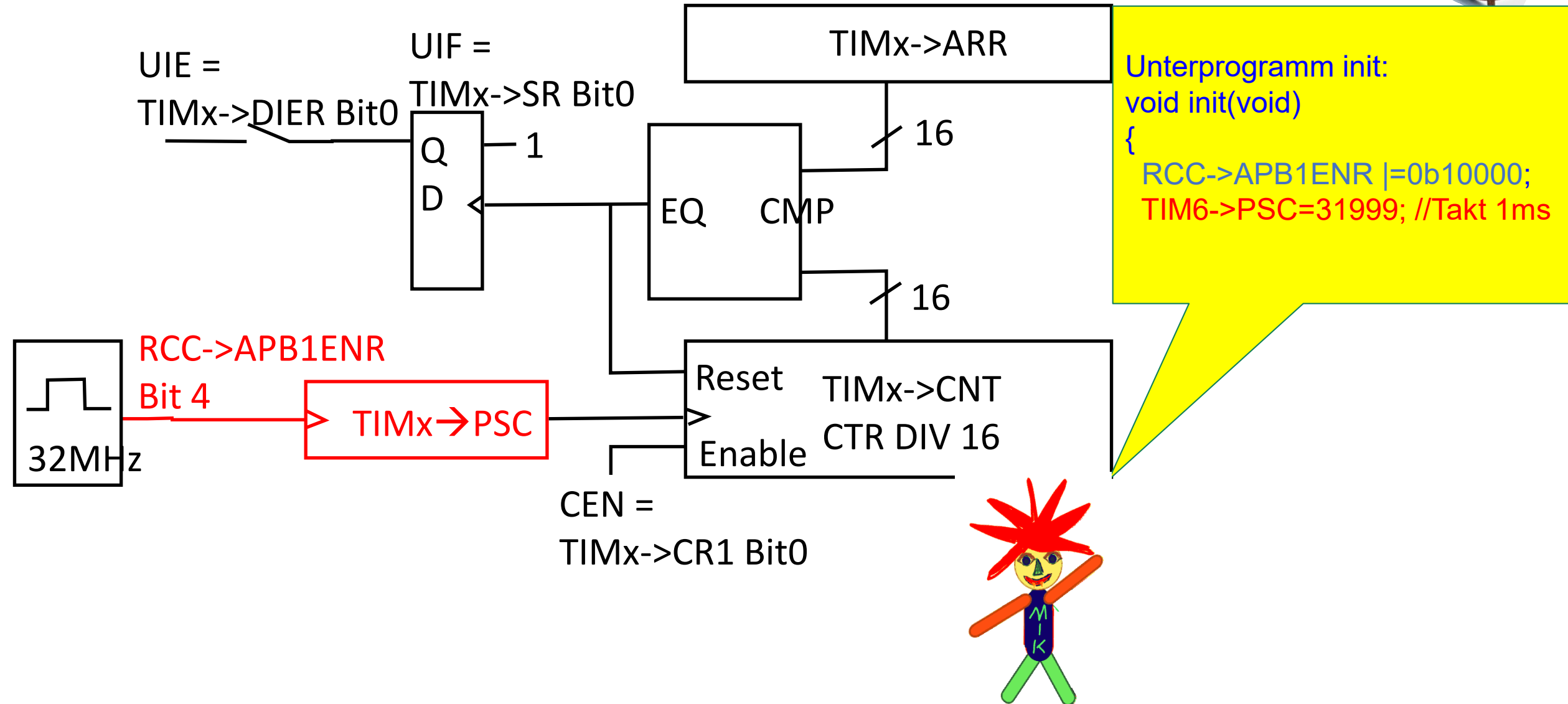


Beispiel: Eine LED
soll mit 1Hz blinken

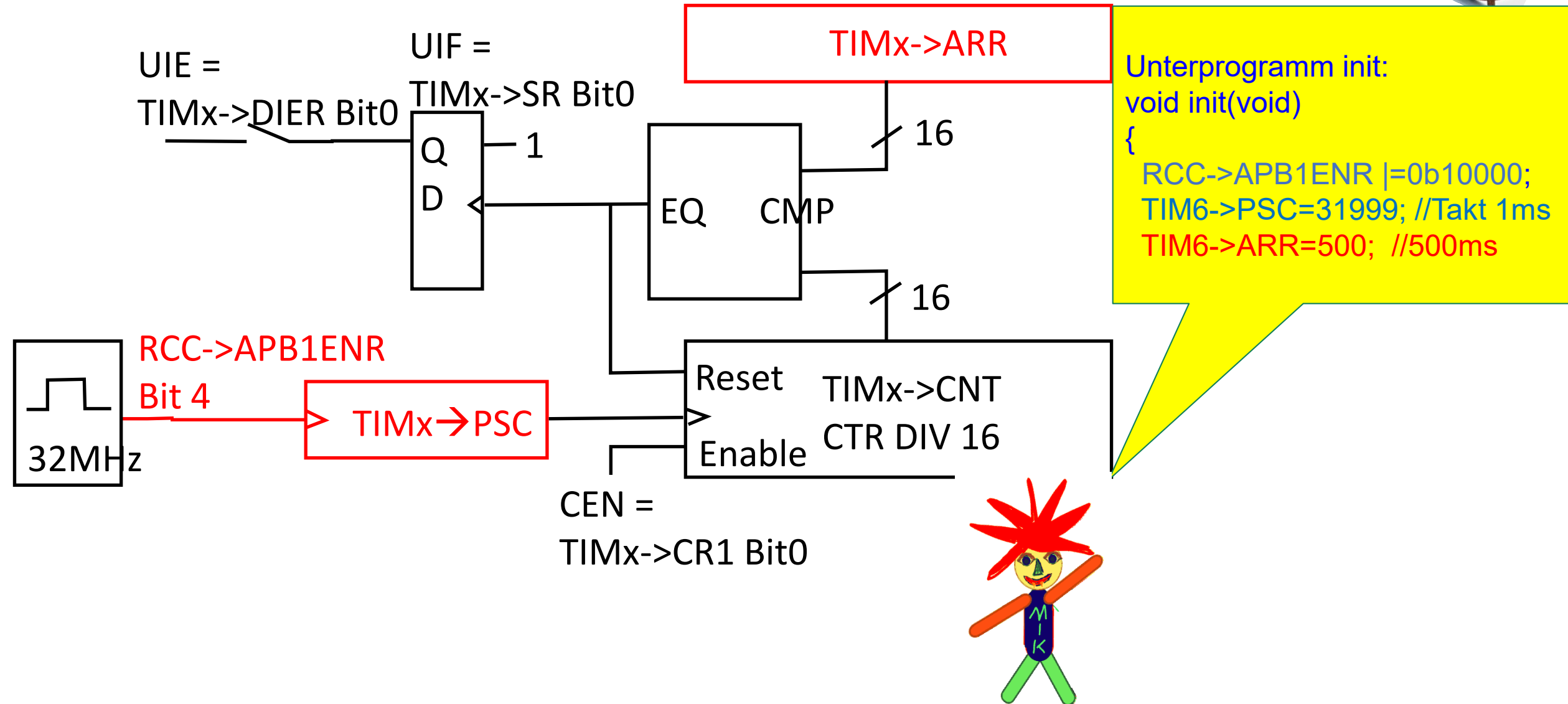




Timer



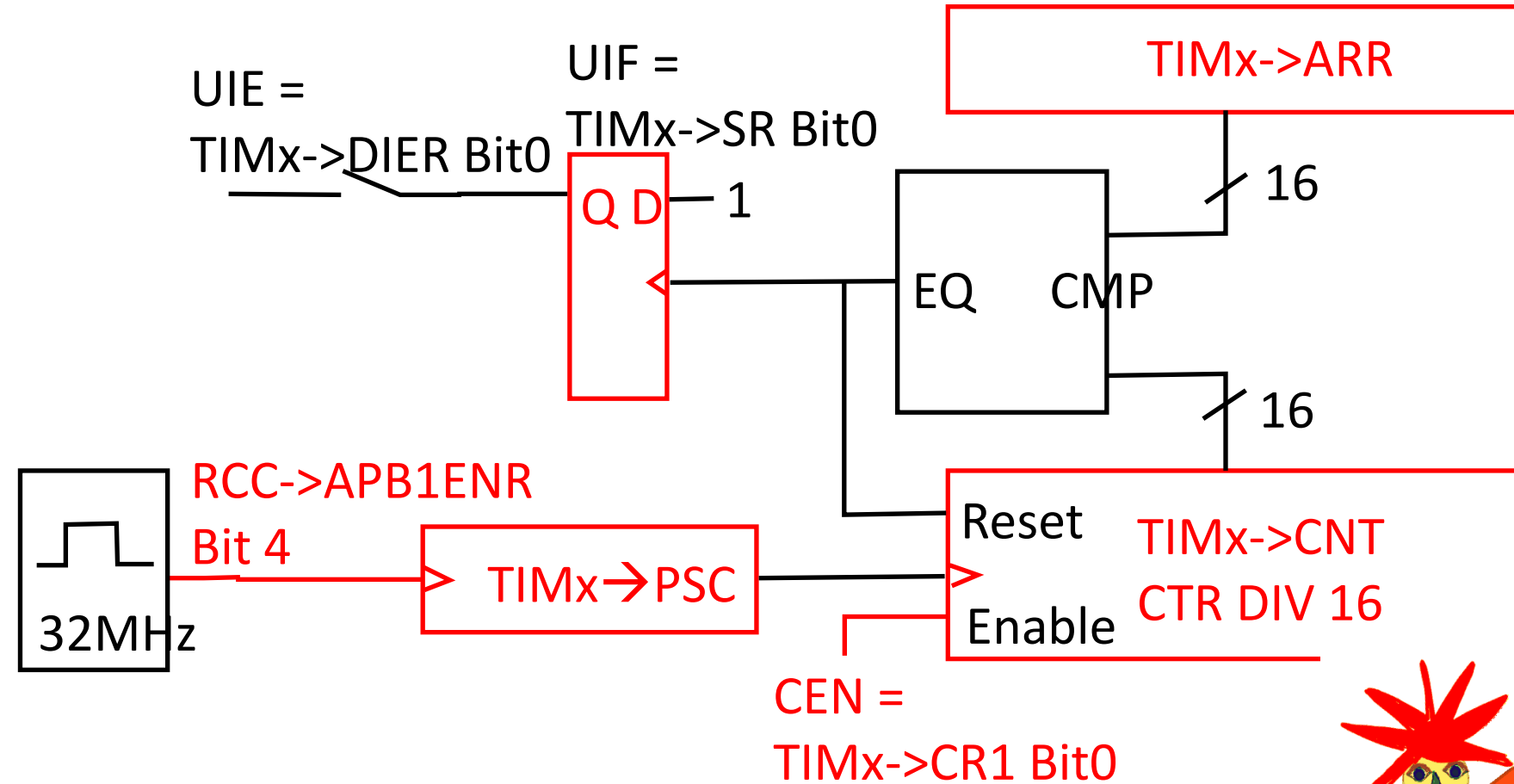
Timer







Timer

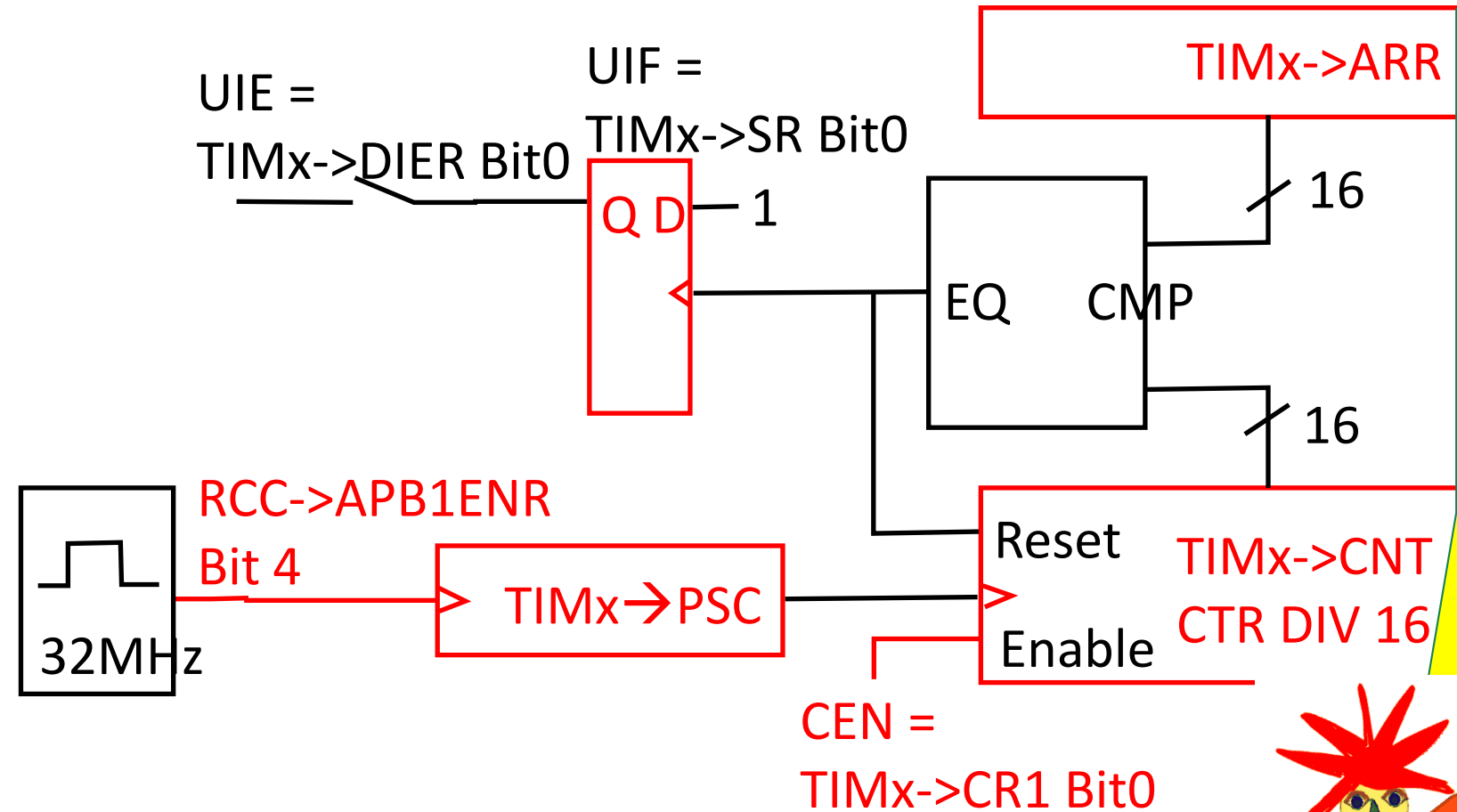


Unterprogramm init:
void init(void)

```
{  
  RCC->APB1ENR |= 0b10000;  
  TIM6->PSC=31999; //Takt 1ms  
  TIM6->ARR=500; //500ms  
  TIM6->CNT=0; //bei 0 beginnen  
  TIM6->SR=0; //UIF = 0  
  TIM6->CR1=1; //CEN=1 start  
}
```



Timer



Hauptprogramm:

```
int main()
```

```
{
```

```
    init(); //Timer initialisieren
```

```
    DigitalOut LED(PC_0);
```

```
    while(true)
```

```
    {
```

```
        if (TIM6->SR!=0) //Überlauf?
```

```
        {
```

```
            TIM6->SR=0; //Flag zurücksetzen
```

```
            LED=!LED; //LED umschalten
```

```
        }
```

```
    }
```

```
}
```

