

# Projekt Schrittmotor mit Siebensegmentanzeige und analoger Geschwindigkeitsvorgabe

Multifunctionshield + Schrittmotorshield

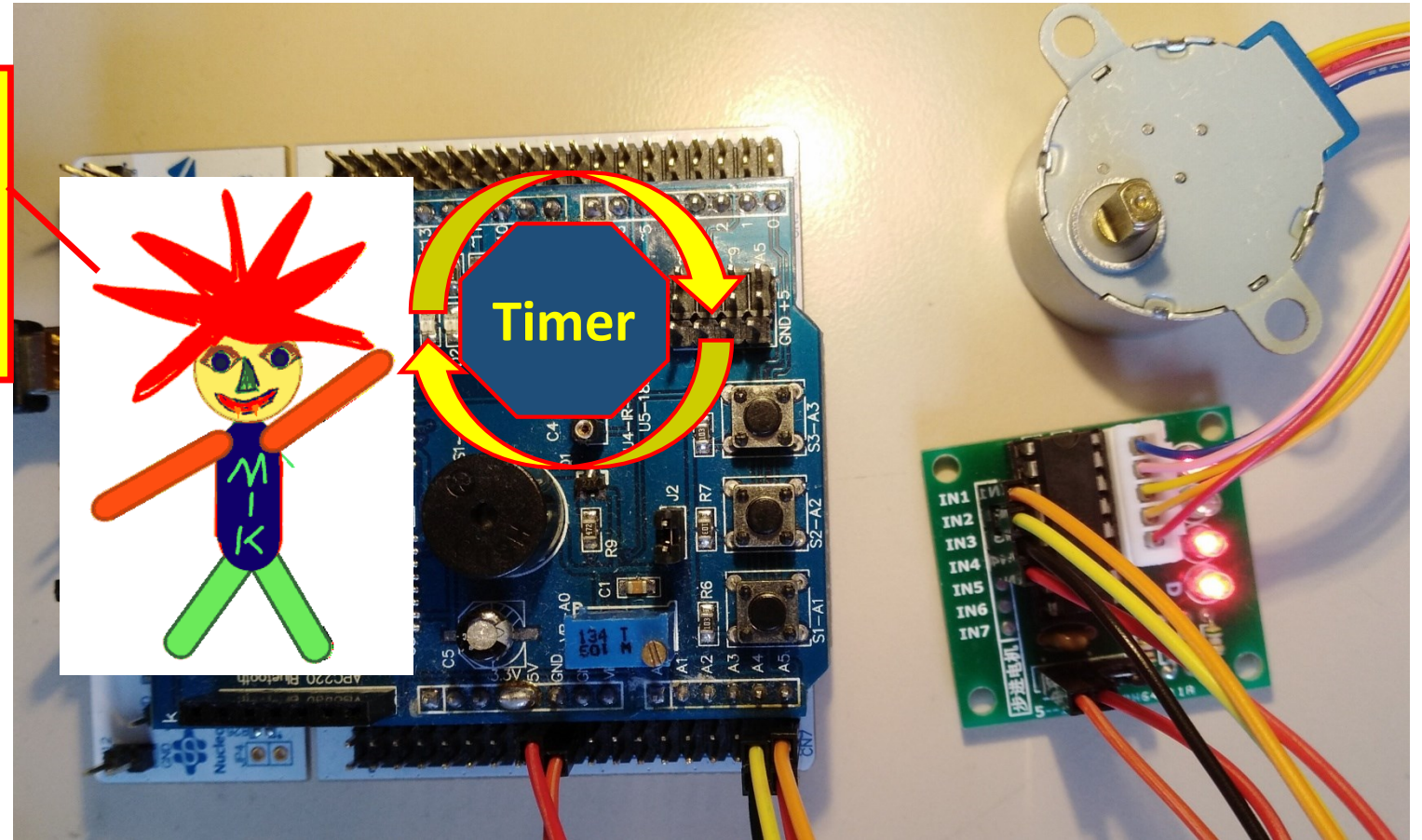


# Projekt Schrittmotor mit Siebensegmentanzeige und analoger Geschwindigkeitsvorgabe

## Projekthighlights:

- Schrittmotor
- Multifunktionsdisplay
  - Analog Geschwindigkeit mit Poti
  - 4-stellige Geschwindigkeitsanzeige auf Siebensegmentanzeige
  - Interruptgesteuerte Richtungsumkehr mit Taste A1

Hardwaretimer für die Geschwindigkeitseinstellung



# Projekt Schrittmotor mit Siebensegmentanzeige und analoger Geschwindigkeitsvorgabe

Der Mikrocontroller  
hat viele Timer



Pinout & Configuration

Search

A-Z

Categories

System ... >

Analog >

Timers >

RTC

⚠ TIM2

TIM3

TIM4

⚠ TIM5

✓ TIM6

TIM7

TIM9

TIM10

TIM11

Connect... >

Multime... >

TIM2 Mode and Configuration

Mode

Slave Mode Disable

Trigger Source Disable

Clock Source Internal Clock

Channel1 Disable

Channel2 Disable

Configuration

Reset Configuration

✓ NVIC Settings

✓ DMA Settings

✓ Parameter Settings

✓ User Constants

Configure the below parameters :

Search (Ctrl+F)

Counter Settings

Prescaler (PSC -... 319

Counter Mode Up

Counter Period (... 4999

Internal Clock Div... No Division

auto-reload preload Disable

Trigger Output (TRGO) ...

Master/Slave Mo... Disable (Trigger input effect...

Trigger Event Sel Reset (UG bit from TIMx F



# Projekt Schrittmotor mit Siebensegmentanzeige und analoger Geschwindigkeitsvorgabe

Mit vielen  
Einstellungen



Pinout & Configuration

Search

A-Z

Categories

System ... >

Analog >

Timers >

Connect... >

Multime... >

### TIM2 Mode and Configuration

#### Mode

Slave Mode

Trigger Source

Clock Source

Channel1

Channel2

#### Configuration

Reset Configuration

☒ NVIC Settings ☒ DMA Settings

☒ Parameter Settings ☒ User Constants

Configure the below parameters :

Search (Ctrl+F) < > i

Counter Settings

Prescaler (PSC -...	319
Counter Mode	Up
Counter Period (...)	4999
Internal Clock Div...	No Division
auto-reload preload	Disable

Trigger Output (TRGO) ...

Master/Slave Mo...	Disable (Trigger input effect...
Trigger Event Sel	Reset (UG bit from TIMx F



# Projekt Schrittmotor mit Siebensegmentanzeige und analoger Geschwindigkeitsvorgabe

Wir aktivieren Timer  
TIM2



Pinout & Configuration

Search

A-Z

Categories

System ... >

Analog >

Timers ▾

RTC

⚠ TIM2

TIM3

TIM4

⚠ TIM5

✓ TIM6

TIM7

TIM9

TIM10

TIM11

Connect... >

Multime... >

TIM2 Mode and Configuration

Mode

Slave Mode Disable ▾

Trigger Source Disable ▾

Clock Source Internal Clock ▾

Channel1 Disable ▾

Channel2 Disable ▾

Configuration

Reset Configuration

✓ NVIC Settings

✓ DMA Settings

✓ Parameter Settings

✓ User Constants

Configure the below parameters :

Search (Ctrl+F)

Counter Settings

Prescaler (PSC -... 319

Counter Mode Up

Counter Period (... 4999

Internal Clock Div... No Division

auto-reload preload Disable

Trigger Output (TRGO) ...


Master/Slave Mo... Disable (Trigger input effect...

Trigger Event Sel Reset (UG bit from TIMx F



# Projekt Schrittmotor mit Siebensegmentanzeige und analoger Geschwindigkeitsvorgabe

Mit Taktquelle:  
Internal Clock



Pinout & Configuration

Search

TIM2 Mode and Configuration

Mode

Slave Mode

Trigger Source

**Clock Source**

Channel1

Channel2

Configuration

Reset Configuration

☒ NVIC Settings ☒ DMA Settings

☒ Parameter Settings ☒ User Constants

Configure the below parameters :

Search (Ctrl+F)

Counter Settings

Prescaler (PSC -...)	319
Counter Mode	Up
Counter Period (...)	4999
Internal Clock Div...	No Division
auto-reload preload	Disable

Trigger Output (TRGO) ...

Master/Slave Mo...	Disable (Trigger input effect...
Trigger Event Sel	Reset (UG bit from TIMx F

TIM4

⚠ TIM5

✓ TIM6

TIM7

TIM9

TIM10

TIM11

Connect... >


Multime... >





# Projekt Schrittmotor mit Siebensegmentanzeige und analoger Geschwindigkeitsvorgabe

Alles andere bleibt disabled



Pinout & Configuration

TIM2 Mode and Configuration

Mode

- Slave Mode: Disable
- Trigger Source: Disable
- Clock Source: Internal Clock
- Channel1: Disable
- Channel2: Disable

Configuration

Reset Configuration

- ✓ NVIC Settings
- ✓ DMA Settings
- ✓ Parameter Settings
- ✓ User Constants

Configure the below parameters :

Search (Ctrl+F)

Counter Settings

- Prescaler (PSC -...): 319
- Counter Mode: Up
- Counter Period (...): 4999
- Internal Clock Div...: No Division
- auto-reload preload: Disable

Trigger Output (TRGO) ...

- Master/Slave Mo...: Disable (Trigger input effect...)
- Trigger Event Sel: Reset (UG bit from TIMx F

TIM4

⚠ TIM5

✓ TIM6

TIM7

TIM9

TIM10

TIM11

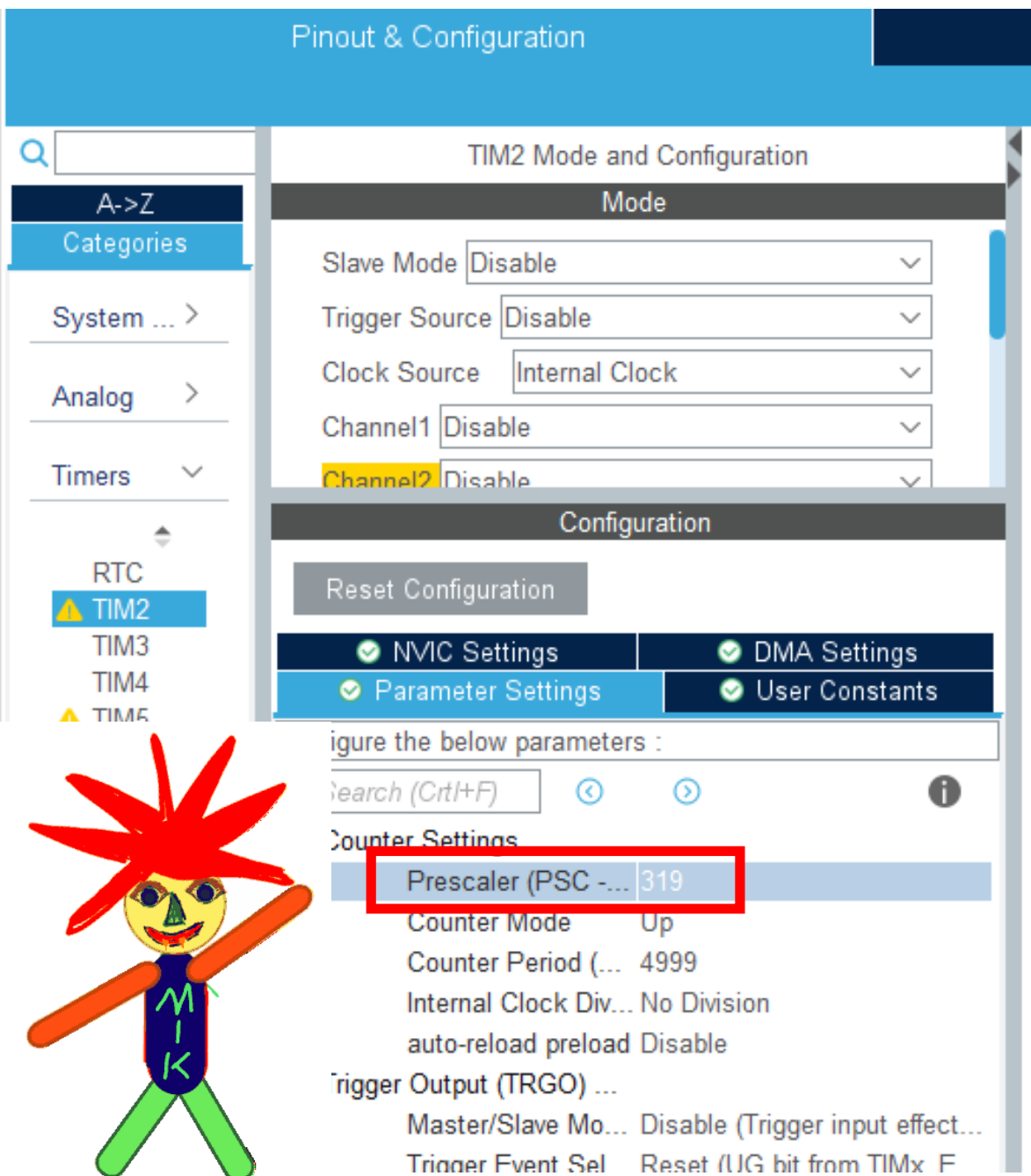
Connect... >

Multime... >



# Projekt Schrittmotor mit Siebensegmentanzeige und analoger Geschwindigkeitsvorgabe

Der Zähltakt wird mit  
dem Prescalerwert  
319 auf 10  $\mu$ s  
eingestellt



Pinout & Configuration

Search

A-Z  
Categories  
System ... >  
Analog >  
Timers

RTC  
TIM2  
TIM3  
TIM4  
TIM5

TIM2 Mode and Configuration

Mode

Slave Mode Disable  
Trigger Source Disable  
Clock Source Internal Clock  
Channel1 Disable  
Channel2 Disable

Configuration

Reset Configuration

✓ NVIC Settings  
✓ Parameter Settings  
✓ DMA Settings  
✓ User Constants

Configure the below parameters :

Search (Ctrl+F)

Counter Settings

Prescaler (PSC) 319

Counter Mode Up  
Counter Period (...) 4999  
Internal Clock Div... No Division  
auto-reload preload Disable  
Trigger Output (TRGO) ...  
Master/Slave Mo... Disable (Trigger input effect...  
Trigger Event Sel Reset (UG bit from TIMx F





# Projekt Schrittmotor mit Siebensegmentanzeige und analoger Geschwindigkeitsvorgabe

Hinter Counter Period verbirgt sich der Autoreloadwert

Pinout & Configuration

Search

A-Z

Categories

System ... >

Analog >

Timers >

RTC

TIM2

TIM3

TIM4

TIM5

TIM6

TIM7

TIM2 Mode and Configuration

Mode

Slave Mode Disable

Trigger Source Disable

Clock Source Internal Clock

Channel1 Disable

Channel2 Disable

Configuration

Reset Configuration

✓ NVIC Settings

✓ DMA Settings

✓ Parameter Settings

✓ User Constants

Configure the below parameters :

Search (Ctrl+F)

er Settings

Prescaler (PSC -... 319

Counter Mode Up

Counter Period (... 4999


Internal Clock Div... No Division

auto-reload preload Disable

r Output (TRGO) ...

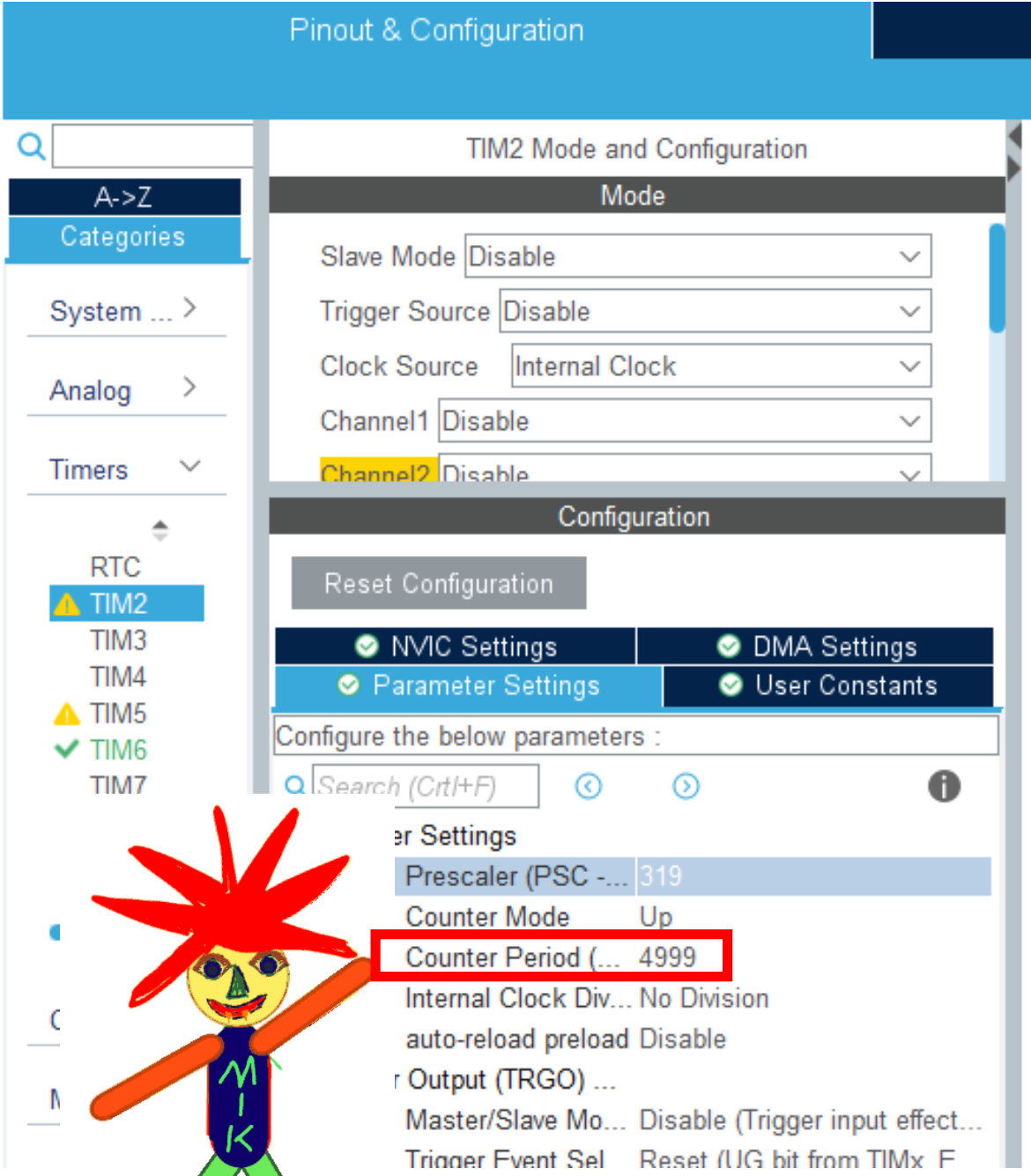
Master/Slave Mo... Disable (Trigger input effect...

Trigger Event Sel Reset (UG bit from TIMx F



# Projekt Schrittmotor mit Siebensegmentanzeige und analoger Geschwindigkeitsvorgabe

4999 bedeutet eine  
Periode von  $5000 * 10\mu s = 50ms$



The screenshot shows the STM32CubeMX Pinout & Configuration window. The left sidebar lists various components, with TIM2 selected under the Timers category. The main panel displays the TIM2 Mode and Configuration settings. The Mode section includes dropdown menus for Slave Mode (Disable), Trigger Source (Disable), Clock Source (Internal Clock), Channel1 (Disable), and Channel2 (Disable). The Configuration section includes a Reset Configuration button and checkboxes for NVIC Settings, DMA Settings, Parameter Settings, and User Constants. Below these, a search bar and navigation arrows are present. The Counter Settings section shows the Prescaler (PSC) set to 319, Counter Mode set to Up, and Counter Period set to 4999. Other settings like Internal Clock Division (No Division), auto-reload preload (Disable), Output (TRGO), Master/Slave Mode (Disable), and Trigger Event Selection (Reset) are also visible.

Mode
Slave Mode: Disable
Trigger Source: Disable
Clock Source: Internal Clock
Channel1: Disable
Channel2: Disable

Configuration
Reset Configuration
NVIC Settings: <input checked="" type="checkbox"/>
DMA Settings: <input checked="" type="checkbox"/>
Parameter Settings: <input checked="" type="checkbox"/>
User Constants: <input checked="" type="checkbox"/>

Configure the below parameters :

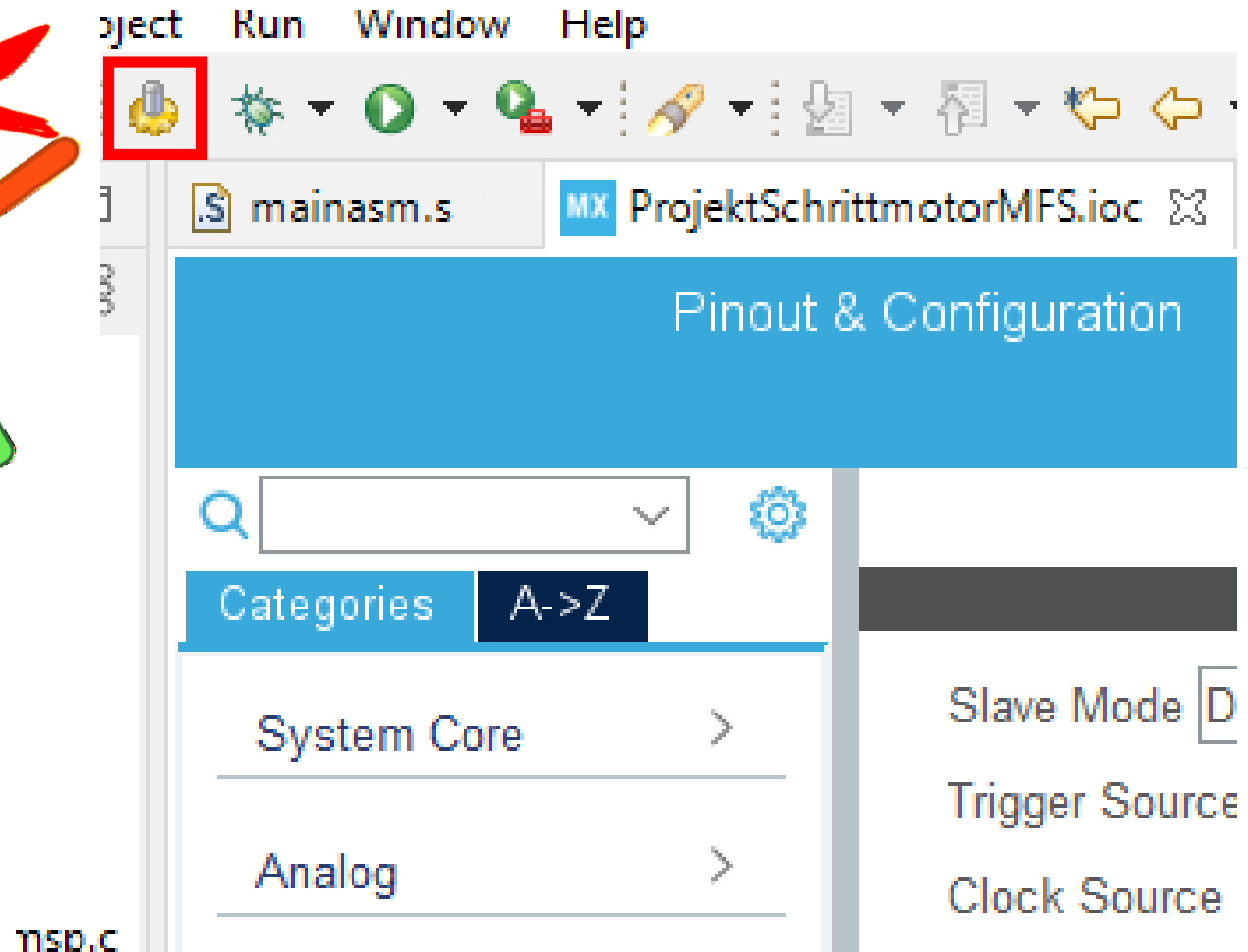
Counter Settings
Prescaler (PSC): 319
Counter Mode: Up
Counter Period (...): 4999
Internal Clock Div...: No Division
auto-reload preload: Disable
Output (TRGO): ...
Master/Slave Mo...: Disable (Trigger input effect...)
Trigger Event Sel: Reset (UG bit from TIMx F



# Projekt Schrittmotor mit Siebensegmentanzeige und analoger Geschwindigkeitsvorgabe



Übersetzen nicht vergessen



# Projekt Schrittmotor mit Siebensegmentanzeige und analoger Geschwindigkeitsvorgabe

Timer TIM2  
auswählen



main:

```
ldr    R0,=hadc
bl     HAL_ADC_Start
ldr    R4,=GPIOC
mov    R5,#10      //Geschwindigkeit
ldr    R6,=GPIOA
ldr    R8,=TIM2    //Timerauswahl
mov    R0,#1
str    R0,[R8,CR1]//R0=1 setzt CEN auf 1
str    R0,[R8,CNT]//Counter startet bei 1
ife:
ldr    R0,=hadc
bl     HAL_ADC_GetValue
strh   R0,[R8,ARR]//AD-Wandelwert => Autoreload
mov    R5,R0
```



# Projekt Schrittmotor mit Siebensegmentanzeige und analoger Geschwindigkeitsvorgabe

Eine 1 im Controlregister 1 (CR1) setzt das Counter Enable Bit (CEN) auf 1 und startet den Timer



**main:**

```
ldr    R0,=hadc
bl     HAL_ADC_Start
ldr    R4,=GPIOC
mov    R5,#10      //Geschwindigkeit
ldr    R6,=GPIOA
ldr    R8,=TIM2    //Timerauswahl
mov    R0,#1
str    R0,[R8,CR1]//R0=1 setzt CEN auf 1
str    R0,[R8,CNT]//Counter startet bei 1
ife:
ldr    R0,=hadc
bl     HAL_ADC_GetValue
strh   R0,[R8,ARR]//AD-Wandelwert => Autoreload
mov    R5,R0
```



# Projekt Schrittmotor mit Siebensegmentanzeige und analoger Geschwindigkeitsvorgabe

**main:**

```
ldr    R0,=hadc
bl     HAL_ADC_Start
ldr    R4,=GPIOC
mov    R5,#10      //Geschwindigkeit
ldr    R6,=GPIOA
ldr    R8,=TIM2    //Timerauswahl
mov    R0,#1
str    R0,[R8,CR1]//R0=1 setzt CEN auf 1
str    R0,[R8,CNT]//Counter startet bei 1
ife:
ldr    R0,=hadc
bl     HAL_ADC_GetValue
strh   R0,[R8,ARR]//AD-Wandelwert => Autoreload
mov    R5,R0
```

Der Counter  
startet bei 1



# Projekt Schrittmotor mit Siebensegmentanzeige und analoger Geschwindigkeitsvorgabe

In der Endlosschleife speichern wir den jeweiligen Wandelwert im Autoreloadregister (ARR). Damit legen wir fest, wie weit der Counter zählt und dann wieder bei 0 beginnt

**main:**

```
ldr    R0,=hadc
bl     HAL_ADC_Start
ldr    R4,=GPIOC
mov    R5,#10      //Geschwindigkeit
ldr    R6,=GPIOA
ldr    R8,=TIM2    //Timerauswahl
mov    R0,#1
str    R0,[R8,CR1]//R0=1 setzt CEN auf 1
str    R0,[R8,CNT]//Counter startet bei 1
```

**schleife:**

```
ldr    R0,=hadc
bl     HAL_ADC_GetValue
strh   R0,[R8,ARR]//AD-Wandelwert => Autoreload
mov    R5,R0
```





# Projekt Schrittmotor mit Siebensegmentanzeige und analoger Geschwindigkeitsvorgabe



Das Warten mit  
HAL\_Delay wird jetzt  
überflüssig

```
mov    R0,#0b0110    //Schritt des Schrittmotors
strb   R0,[R4,ODR]    //Ausgeben
//lsl  R0,R5,8        //Wartezeit = R5/256
//bl   HAL_Delay      //warten
e2:
ldr    R0,[R8,SR]     //Statusregister abfragen
tst    R0,Bit0
beq    warte2
mov    R0,#0
strb   R0,[R8,SR]
```



# Projekt Schrittmotor mit Siebensegmentanzeige und analoger Geschwindigkeitsvorgabe



```
mov    R0,#0b0110    //Schritt des Schrittmotors
strb   R0,[R4,ODR]    //Ausgeben
//lsr   R0,R5,8        //Wartezeit = R5/256
//bl    HAL_Delay      //warten
:  
ldr    R0,[R8,SR]    //Statusregister abfragen  
tst    R0,Bit0  
beq    warte2  
mov    R0,#0  
strb   R0,[R8,SR]
```

In der Endlosschleife fragen wir in jedem Schritt des Schrittmotors das Statusregister des Timers ab



# Projekt Schrittmotor mit Siebensegmentanzeige und analoger Geschwindigkeitsvorgabe

```
mov    R0,#0b0110    //Schritt des Schrittmotors
strb   R0,[R4,ODR]    //Ausgeben
//lsl  R0,R5,8        //Wartezeit = R5/256
//bl   HAL_Delay      //warten
::
ldr    R0,[R8,SR]     //Statusregister abfragen
tst    R0,Bit0        //Bit0 = UIF
beq    warte2
mov    R0,#0
strb   R0,[R8,SR]
```



Bit0 ist das Updateinterruptflag  
(UIF)



# Projekt Schrittmotor mit Siebensegmentanzeige und analoger Geschwindigkeitsvorgabe

```
mov    R0,#0b0110    //Schritt des Schrittmotors
strb   R0,[R4,ODR]    //Ausgeben
//lsl  R0,R5,8         //Wartezeit = R5/256
//bl   HAL_Delay       //warten
```

**warte2:**

```
ldr    R0,[R8,SR]     //Statusregister abfragen
tst    R0,Bit0         //Bit0 = UIF
beq    warte2         //! $\neq 0 \Rightarrow$  weiter warten
mov    R0,#0
strb   R0,[R8,SR]
```



Solange das Updateinterruptflag (UIF) 0 ist gab es noch keinen Überlauf



# Projekt Schrittmotor mit Siebensegmentanzeige und analoger Geschwindigkeitsvorgabe

```
mov    R0,#0b0110    //Schritt des Schrittmotors
strb   R0,[R4,ODR]    //Ausgeben
//lsl  R0,R5,8        //Wartezeit = R5/256
//bl   HAL_Delay      //warten
```

**warte2:**

```
ldr    R0,[R8,SR]    //Statusregister abfragen
tst    R0,Bit0        //Bit0 = UIF
beq    warte2         //!
```



Nach jedem Überlauf muss das UIF  
wieder zurückgesetzt werden



# Projekt Schrittmotor mit Siebensegmentanzeige und analoger Geschwindigkeitsvorgabe

Beachte: Alle Labels  
müssen  
unterschiedlich  
sein:  
z.B. warte1  
warte2  
Usw.



```
mov    R0,#0b0110    //Schritt des Schrittmotors
strb   R0,[R4,ODR]    //Ausgeben
//lsr   R0,R5,8        //Wartezeit = R5/256
//bl    HAL_Delay      //warten

warte2:
ldr    R0,[R8,SR]     //Statusregister abfragen
tst    R0,Bit0         //Bit0 = UIF
beq    warte2          //!=0 => weiter warten
mov    R0,#0           //UIF zurücksetzen
strb   R0,[R8,SR]
```

//Nächster Schritt genauso

