

Getting Started with STM32CubeIDE



Die Entwicklungsumgebung

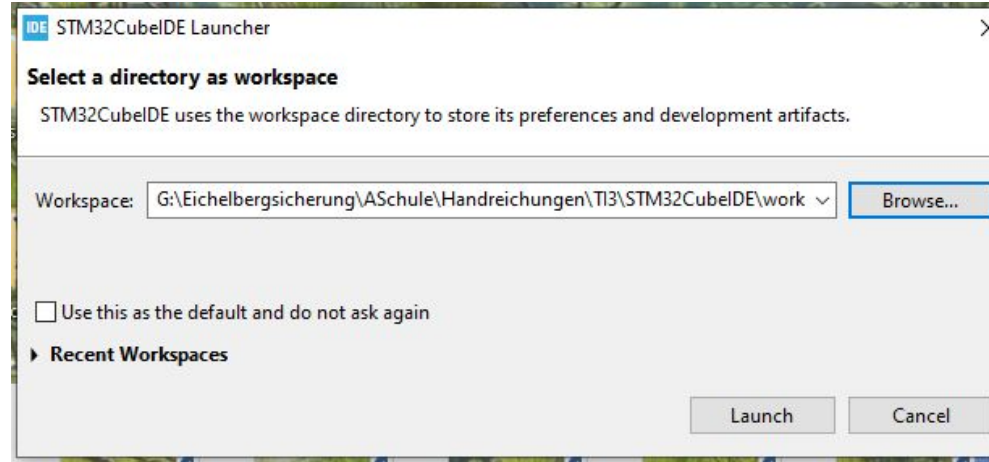
Ich bin Mik, Dein Mikrocontroller

Getting Started with STM32CubeIDE



STM32CubeIDE starten

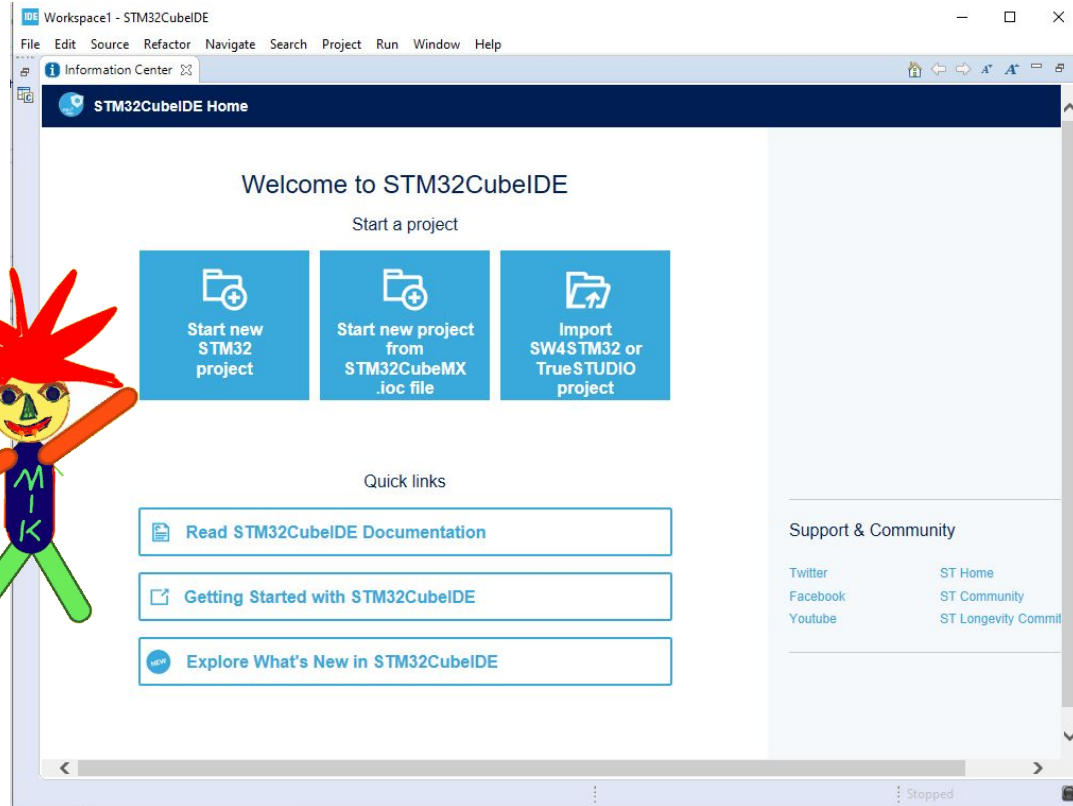
Getting Started with STM32CubeIDE

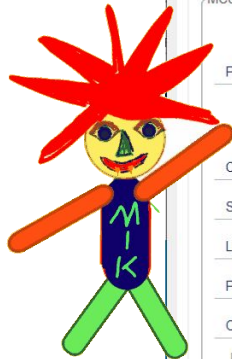


Ein Workspace ist ein Ordner in dem Projekte gespeichert werden. Beim ersten Starten am besten einen neuen Ordner anlegen. Dann "Launch"

Getting Started with STM32CubeIDE

Wir starten ein neues
STM32 project





Im Target Selector wählen
wird unseren
Mikrocontroller aus:
STM32L152RE
bei Part Number Search

STM32 Project

Target Selection
Select STM32 target

MCU/MPU Selector Board Selector Cross Selector

MCU/MPU Filters

Part Number Search

Core

Series

Line

Package

Other

Price From 0.0 to 9.488

IO From 11 to 176

Eeprom From 0 to 16384 (Bytes)

Flash From 0 to 2048 (kBytes)

Ram From 2 to 1184 (kBytes)

Freq. From 24 to 650 (MHz)

Peripheral

ADC 12-bit 0 42

ADC 16-bit 0 36

Features Block Diagram Docs & Resources Datasheet Buy

STM32L5 MCUs with Arm® Cortex®-M33
Ultra-low power and security for IoT

STM32L5

MCUs/MPUs List: 1616 items

	Part No.	Reference	Marketing Status	Unit Price for 10kU (US\$)	Board	Package	Flash	RAM	ID	Freq.
☆	STM32F030C6	STM32F030C6Tx	Active	0.597		LQFP48	32 kBytes	4 kBytes	39	48 MHz
☆	STM32F030C8	STM32F030C8Tx	Active	0.722		LQFP48	64 kBytes	8 kBytes	39	48 MHz
☆	STM32F030CC	STM32F030CCTx	Active	1.1		LQFP48	256 kBytes	32 kBytes	37	48 MHz
☆	STM32F030F4	STM32F030F4Px	Active	0.424		TSSOP20	16 kBytes	4 kBytes	15	48 MHz
☆	STM32F030K6	STM32F030K6Tx	Active	0.518		LQFP32	32 kBytes	4 kBytes	25	48 MHz
☆	STM32F030R8	STM32F030R8Tx	Active	0.754	N. 32	LQFP64	64 kBytes	8 kBytes	55	48 MHz
☆	STM32F030RC	STM32F030RCTx	Active	1.21		LQFP64	256 kBytes	32 kBytes	51	48 MHz
☆	STM32F031C4	STM32F031C4Tx	Active	0.97		LQFP48	16 kBytes	4 kBytes	39	48 MHz
☆	STM32F031C6	STM32F031C6Tx	Active	1.013		LQFP48	32 kBytes	4 kBytes	39	48 MHz
☆	STM32F031E6	STM32F031E6Yx	Active	0.776		WLCS25	32 kBytes	4 kBytes	20	48 MHz
☆	STM32F031F4	STM32F031F4Px	Active	0.711		TSSOP20	16 kBytes	4 kBytes	15	48 MHz
☆	STM32F031F6	STM32F031F6Px	Active	0.755		TSSOP20	32 kBytes	4 kBytes	15	48 MHz
☆	STM32F031G4	STM32F031G4Ux	Active	0.733		UFQFPN28	16 kBytes	4 kBytes	23	48 MHz

< Back Next > Finish Cancel

Target Selection

Select STM32 target



Im Target Selector wählen
wird unseren
Mikrocontroller aus:
STM32L152RE
bei Part Number Search

MCU/MPU Selector Board Selector Cross Selector

MCU/MPU Filters

Part Number Search

STM32L152RE

Series

Line

Package

Other

Price = 3.159

IO = 51

Eeprom = 16384 (Bytes)

Flash = 512 (kBytes)

Ram = 80 (kBytes)

Freq. = 32 (MHz)

Peripheral

ADC 12-bit 0 21

ADC 16-bit 0 0

Features Block Diagram Docs & Resources Datasheet Buy

STM32L5 MCUs with Arm® Cortex®-M33
Ultra-low power and security for IoT

STM32L5

MCUs/MPUs List: 1 item

Display similar items

Export

*	Part No	Reference	Marketing Status	Unit Price for 10kU (US\$)	Board	Package	Flash	RAM	ID	Freq
☆	STM32L152RE	STM32L152RETx	Active	3.159	NUCLEO	LQFP64	512 kBytes	80 kBytes	51	32 MHz

Target Selection

Select STM32 target

MCU/MPU Selector Board Selector Cross Selector

MCU/MPU Filters

Part Number Search

STM32L152RE

Core

Series

Line

Package

Other

Price = 3.159

IO = 51

Eeprom = 16384 (bytes)

Flash = 512 (Kbytes)

Ram = 80 (Kbytes)

Freq. = 32 (MHz)

Peripheral

ADC 12-bit 0

ADC 16-bit 0 0

STM32L152RE

Ultra-low-power ARM Cortex-M3 MCU with 512 Kbytes Flash, 32 MHz CPU, USB, 2xOp-amp

STM32 L1

ACTIVE Active

Product is in mass production

Unit Price for 10kU (US\$) : 3.159

Board: NUCLEO-L152RE

LQFP64

The ultra-low-power STM32L151xE and STM32L152xE devices incorporate the connectivity power of the universal serial bus (USB) with the high-performance ARM Cortex-M3 32-bit RISC core operating at a frequency of 32 MHz (33.3 DMIPS), a memory protection unit (MPU), high-speed embedded memories (Flash memory up to 512 Kbytes and RAM up to 80 Kbytes), and an extensive range of enhanced I/Os and peripherals connected to two APB buses.

The STM32L151xE and STM32L152xE devices offer two operational amplifiers, one 12-bit ADC, two DACs, two ultra-low-power comparators, one general-purpose 32-bit timer, six general-purpose 16-bit timers and two basic timers, which can be used as time bases. Moreover, the STM32L151xE and STM32L152xE devices contain standard and advanced communication interfaces: up to two I2Cs, three SPIs, two I2S, three USARTs, two UARTs and an USB. The STM32L151xE and STM32L152xE devices offer up to 34 capacitive sensing

MCUs/MPUs List: 1 item

Display similar items

Export

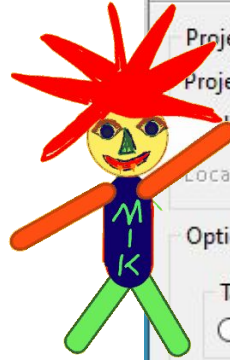
	Part No	Reference	Marketing Status	Unit Price for 10kU (US\$)	Board	Package	Flash	RAM	IO	Freq
☆	STM32L152RE	STM32L152RETx	Active	3.159	NUCLEO	LQFP64	512 Kbytes	80 Kbytes	51	32 MHz

< Back Next > Finish Cancel

Den Mikrocontroller auswählen, weiter mit Next



Gib Deinem ersten Projekt
einen Namen.
Targeted Language: C++
=> Finish



IDE STM32 Project

Setup STM32 project

Project

Project Name:

Project default location

Location:

Options

Targeted Language

☐ C ☒ C++

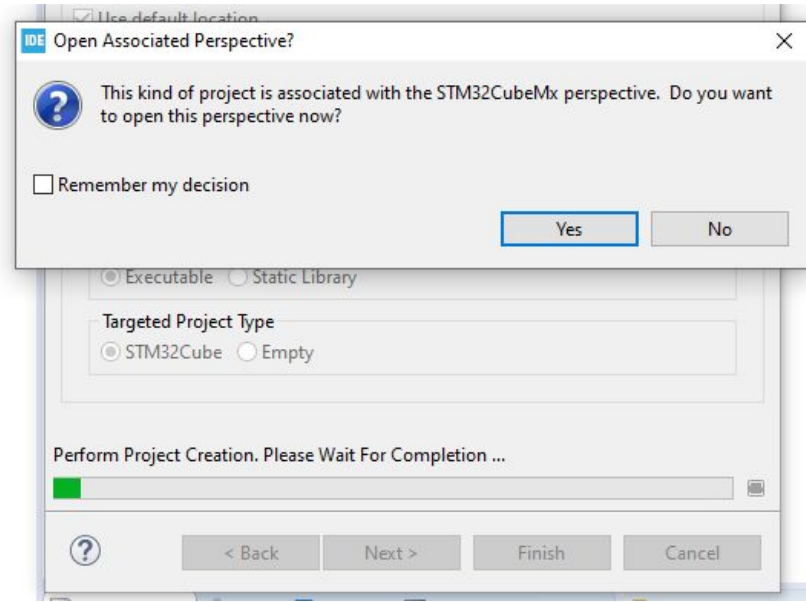
Targeted Binary Type

☒ Executable ☐ Static Library

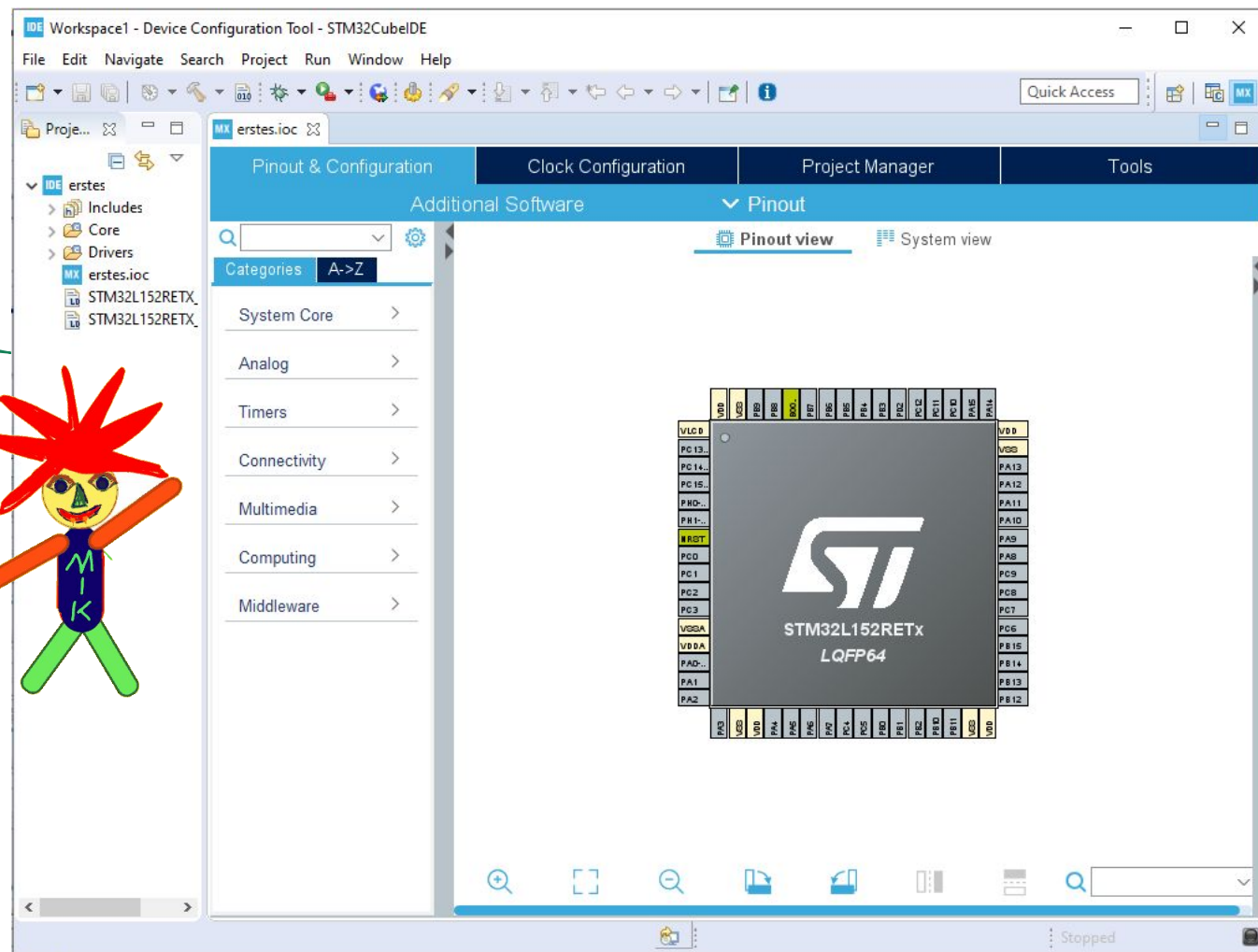
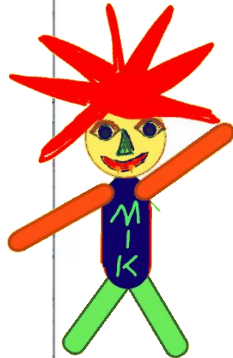
Targeted Project Type

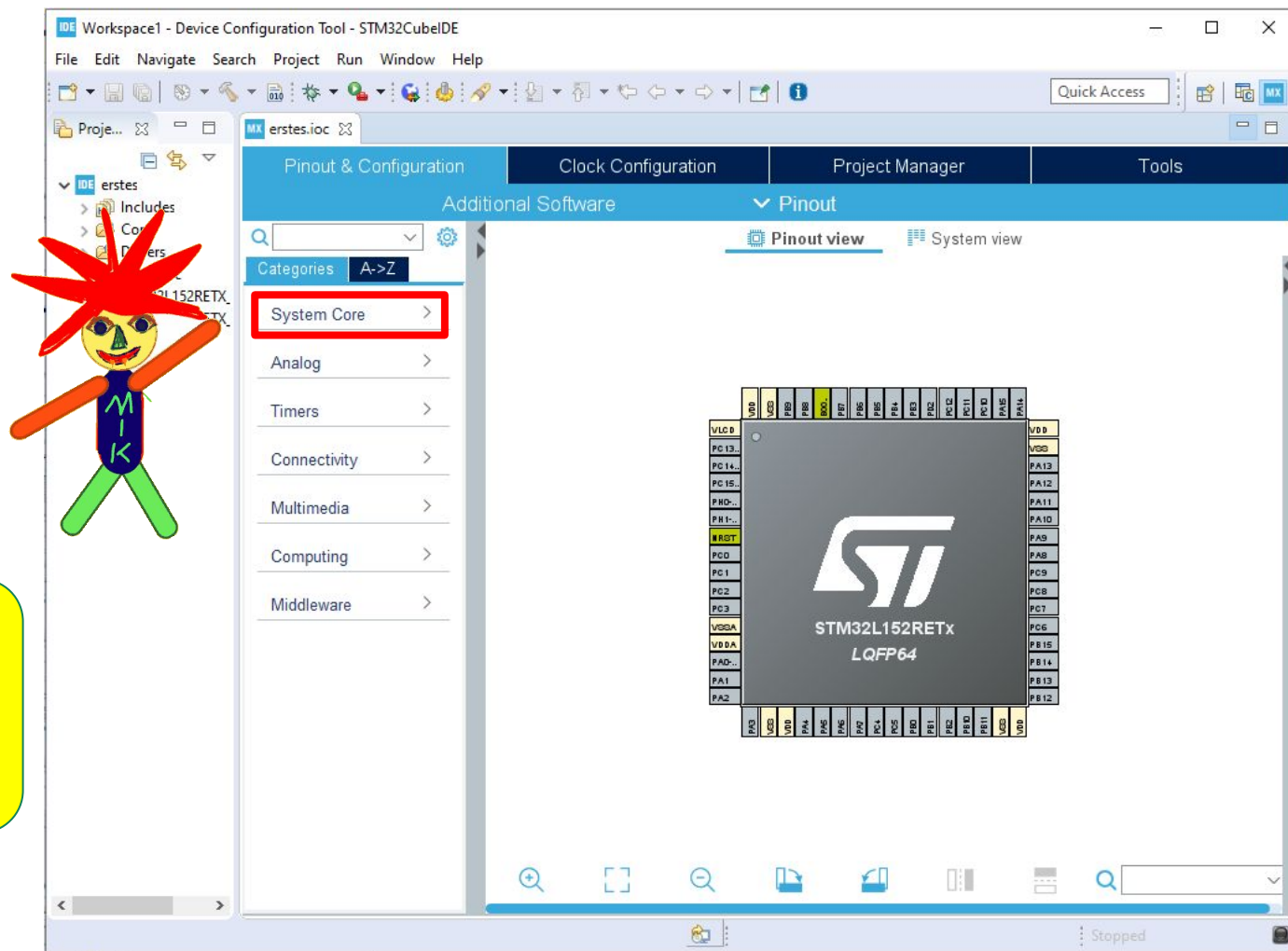
☒ STM32Cube ☐ Empty

Wir wechseln natürlich in
die STM32CubeMX
Ansicht
also Yes



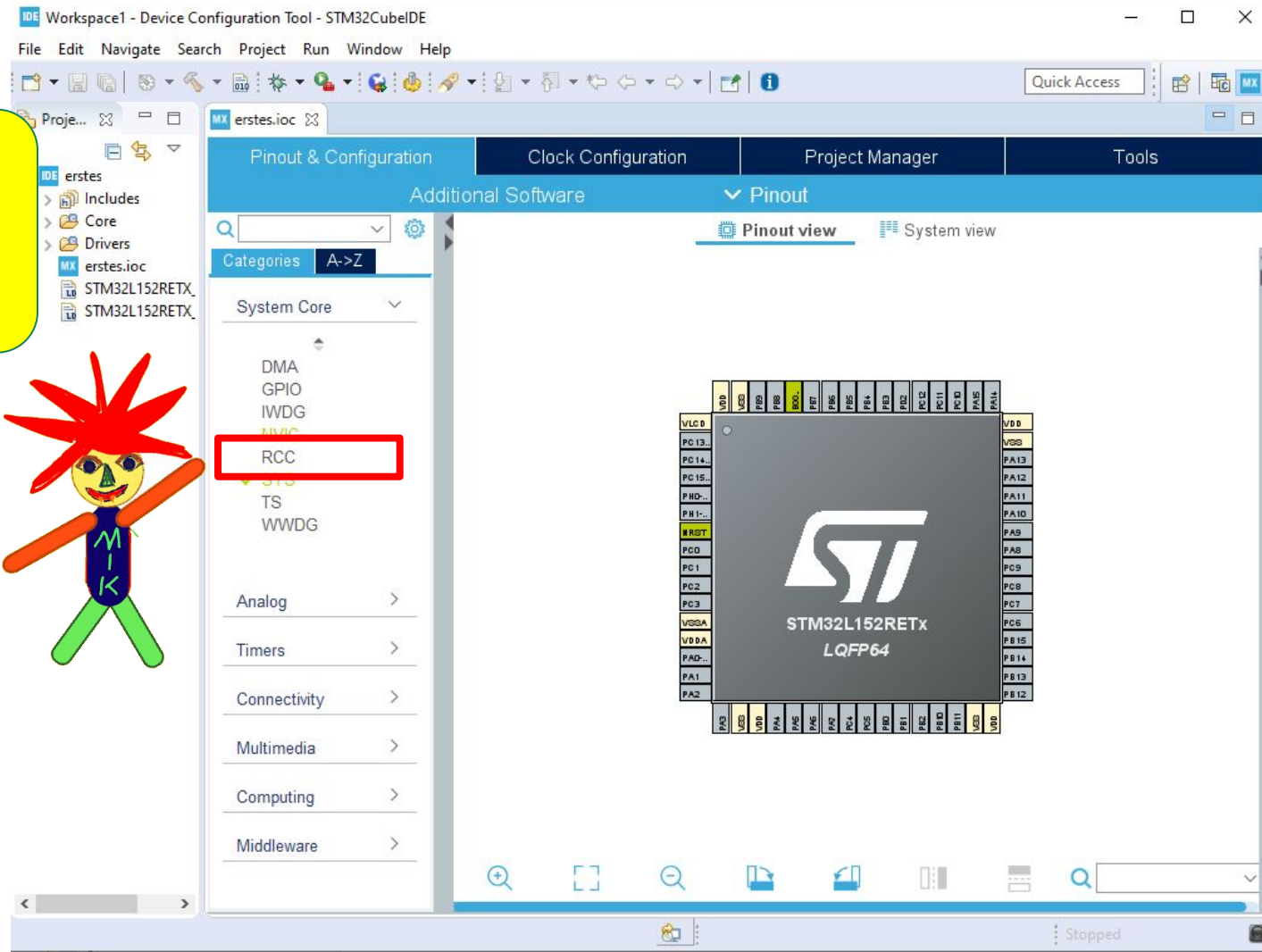
Die Datei erstes.ioc wird grafisch dargestellt und zeigt die Hardwarekonfiguration des Mikrocontrollers



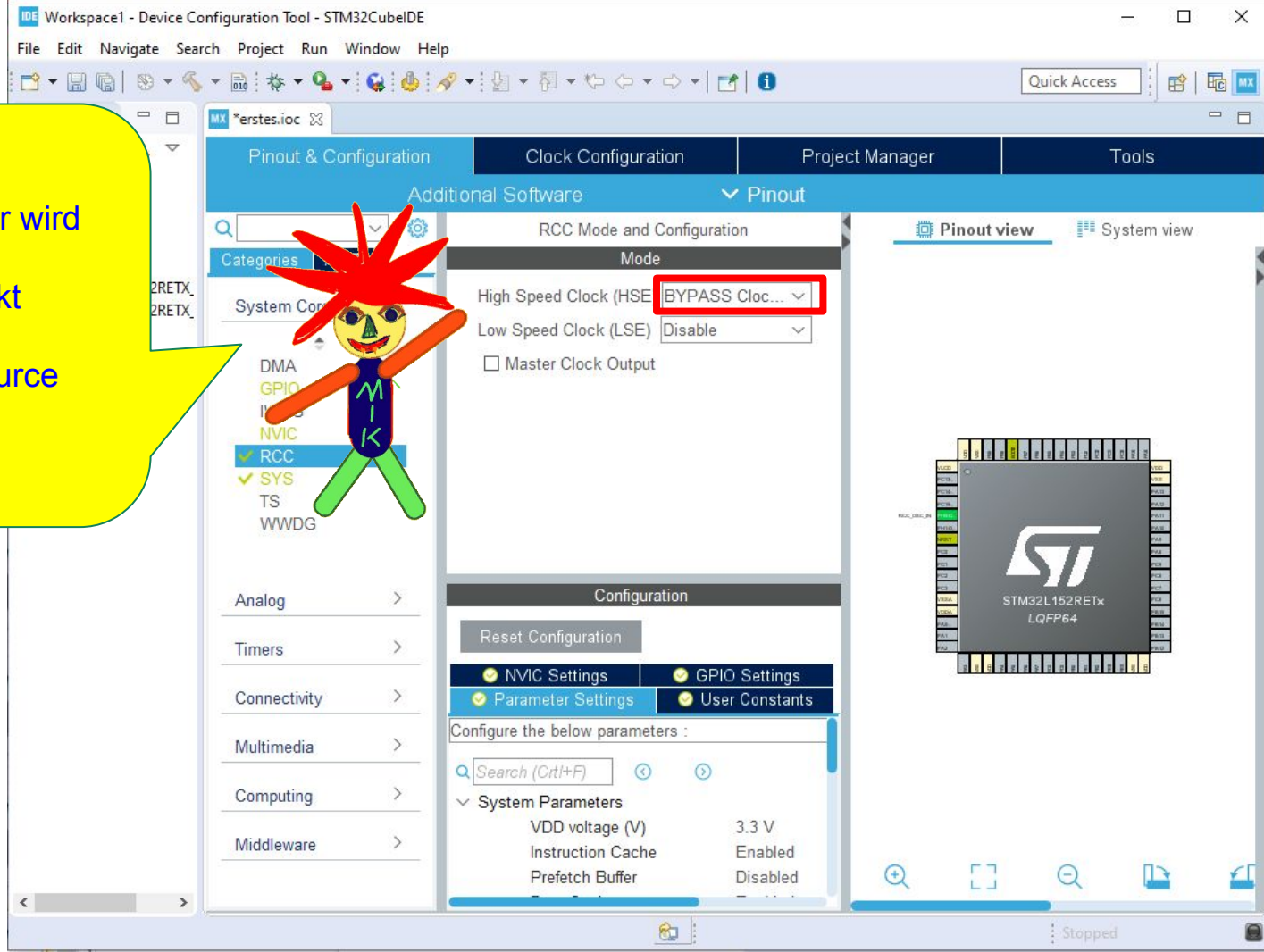


Wir müssen ein paar
Grundeinstellungen
vornehmen.
System Core anklicken

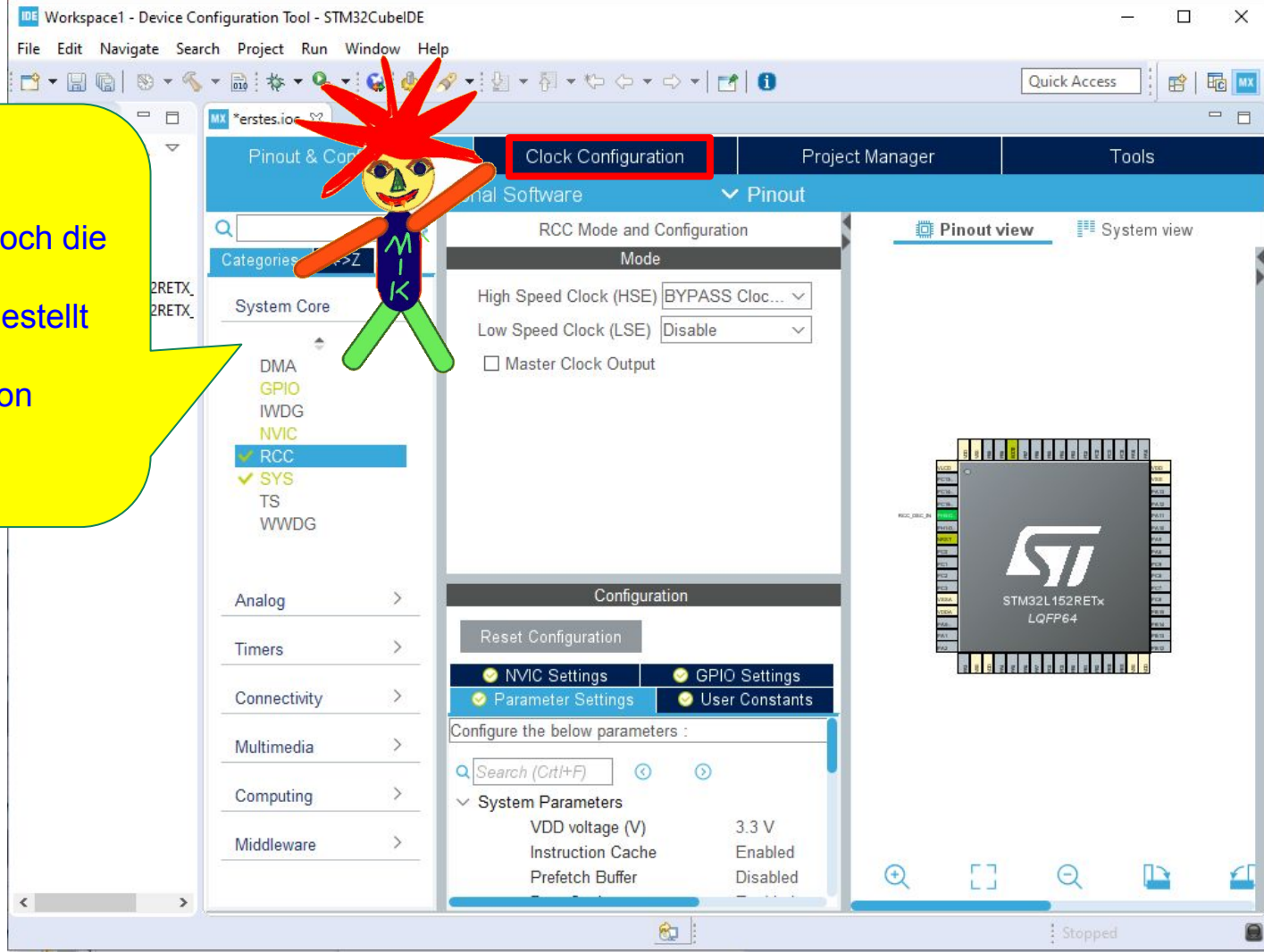
Im Hardwaremodul RCC
stehen
Taktvoreinstellungen
=>anklicken



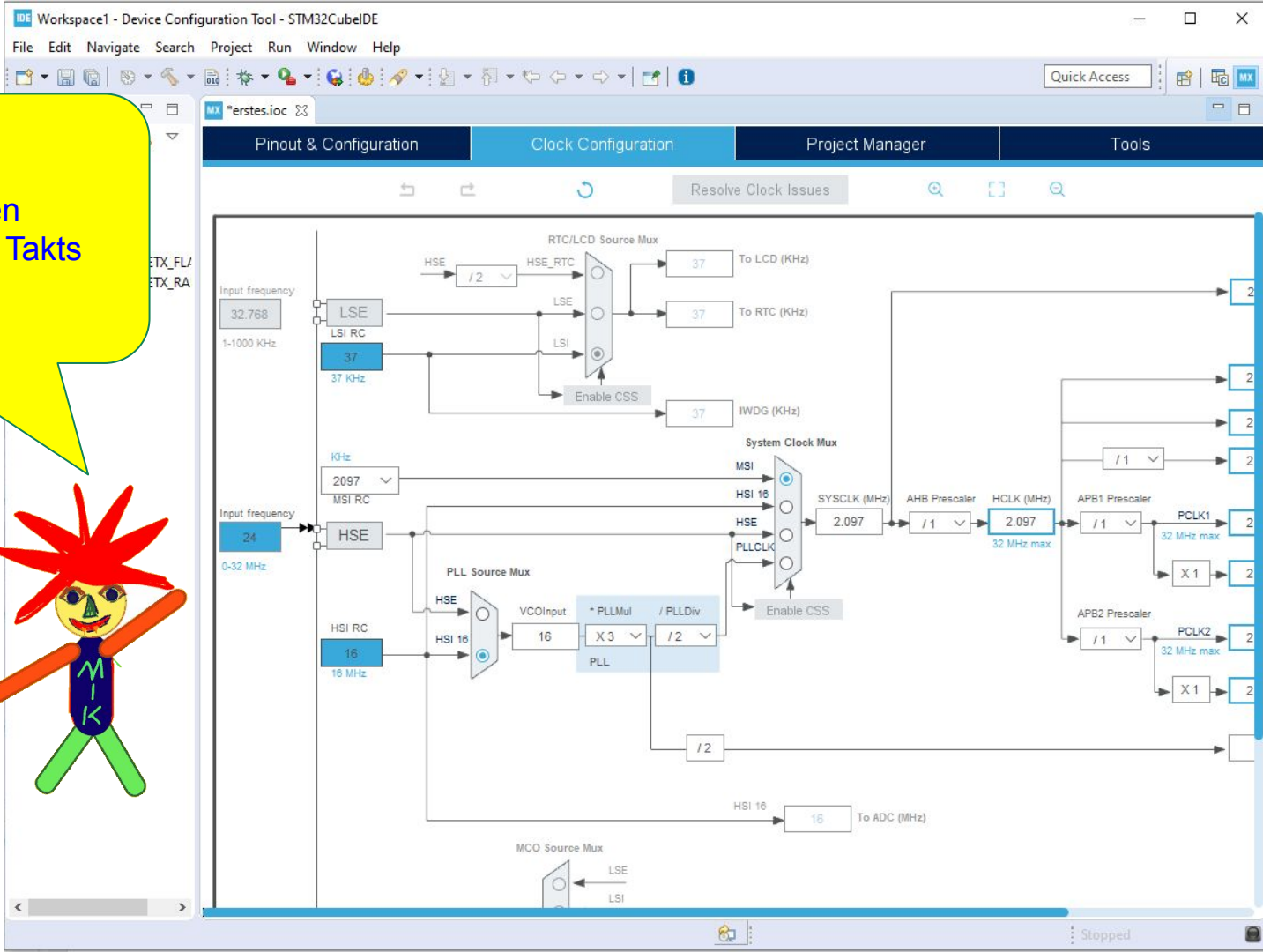
Unser Mikrocontroller wird
von einem externen
Taktgenerator mit Takt
versorgt.
=> Bypass Clock Source
auswählen



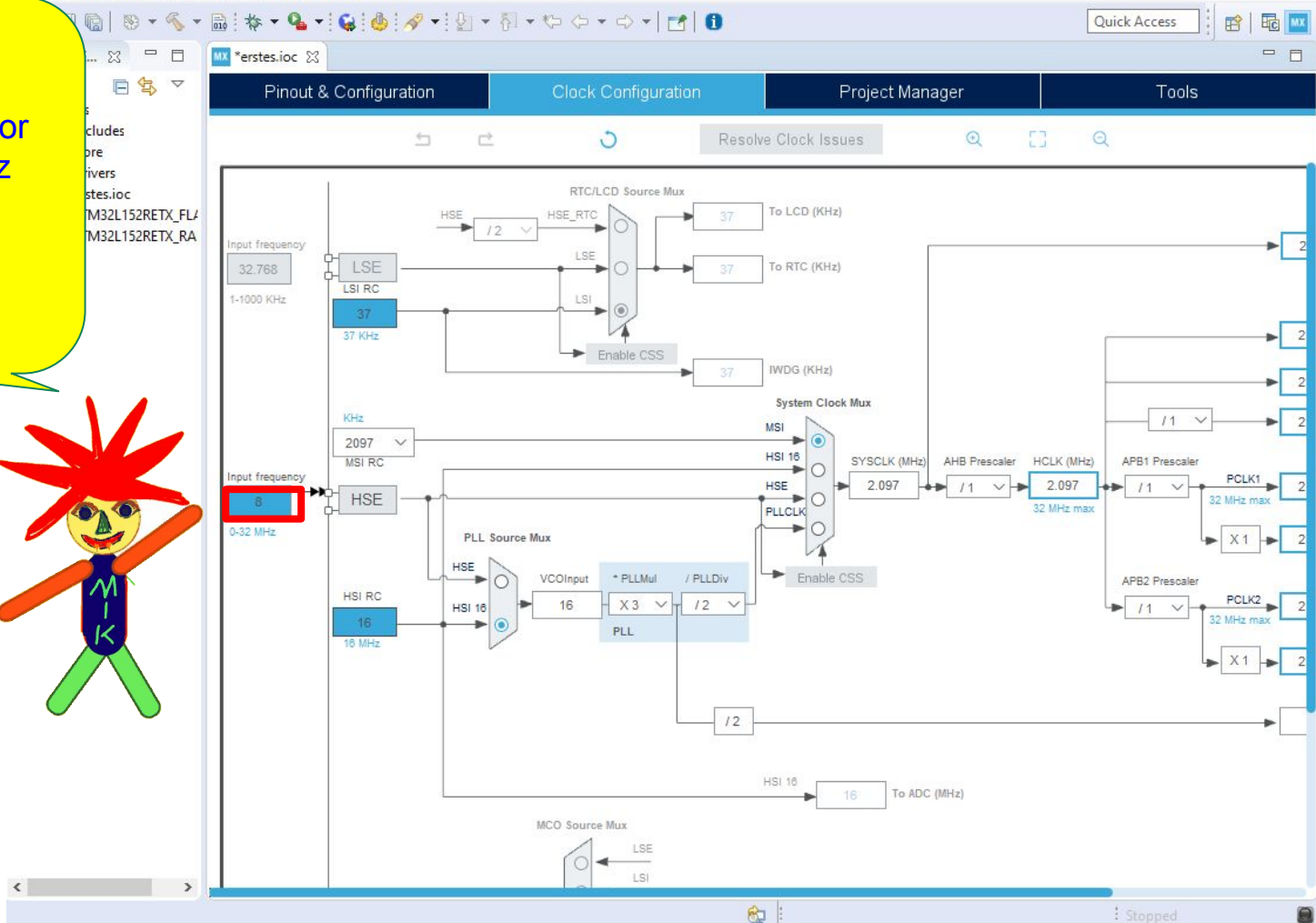
Als nächstes muss noch die
Taktversorgung des
Mikrocontrollers eingestellt
werden.
=> Clock Configuration



Kompliziertes Bild mit allen
Einstellmöglichkeiten des Takts

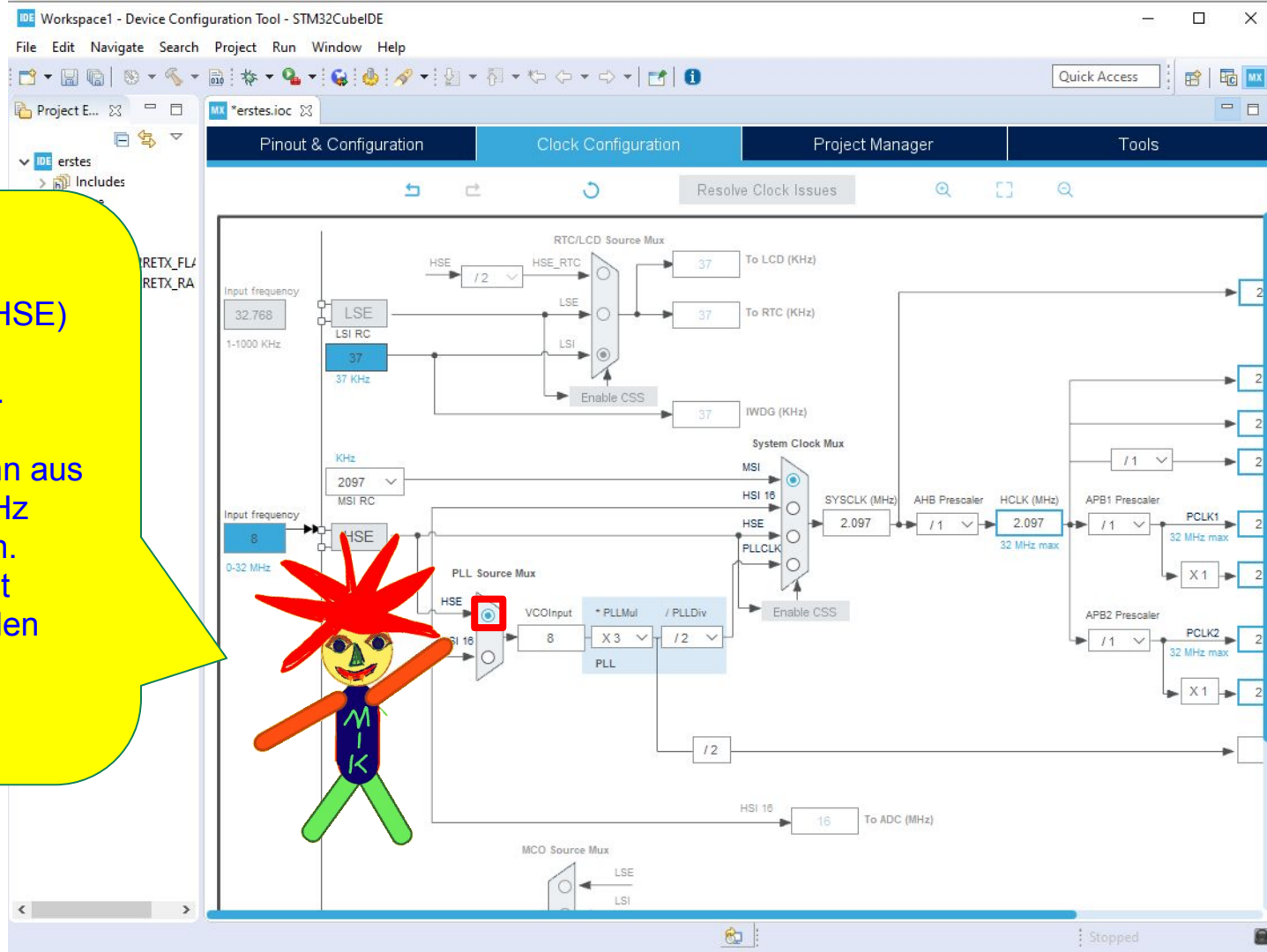


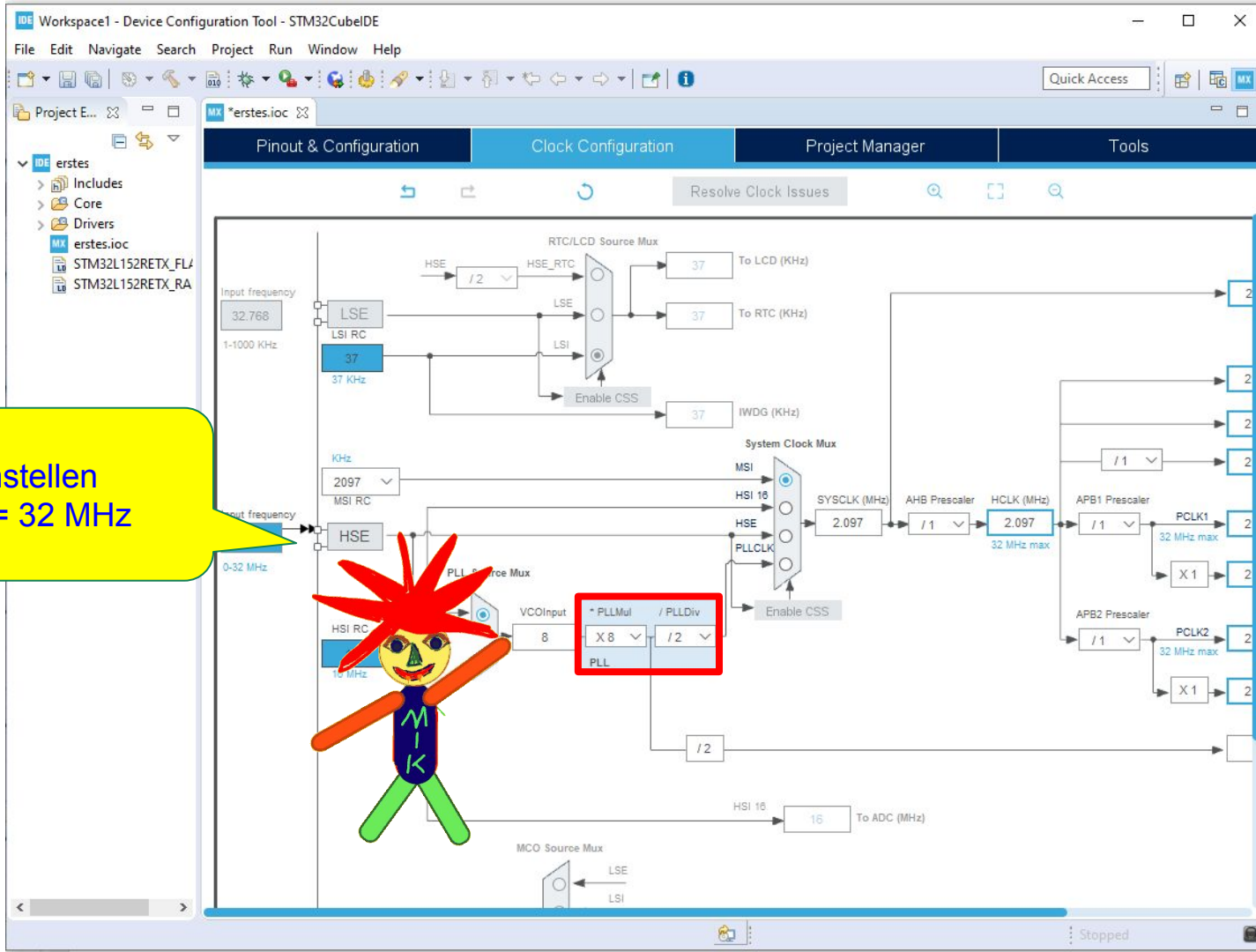
Das Nucleo-Board
versorgt den Prozessor
mit einer Taktfrequenz
 $f=8\text{MHz}$
 \Rightarrow Input Frequency 8
eintragen



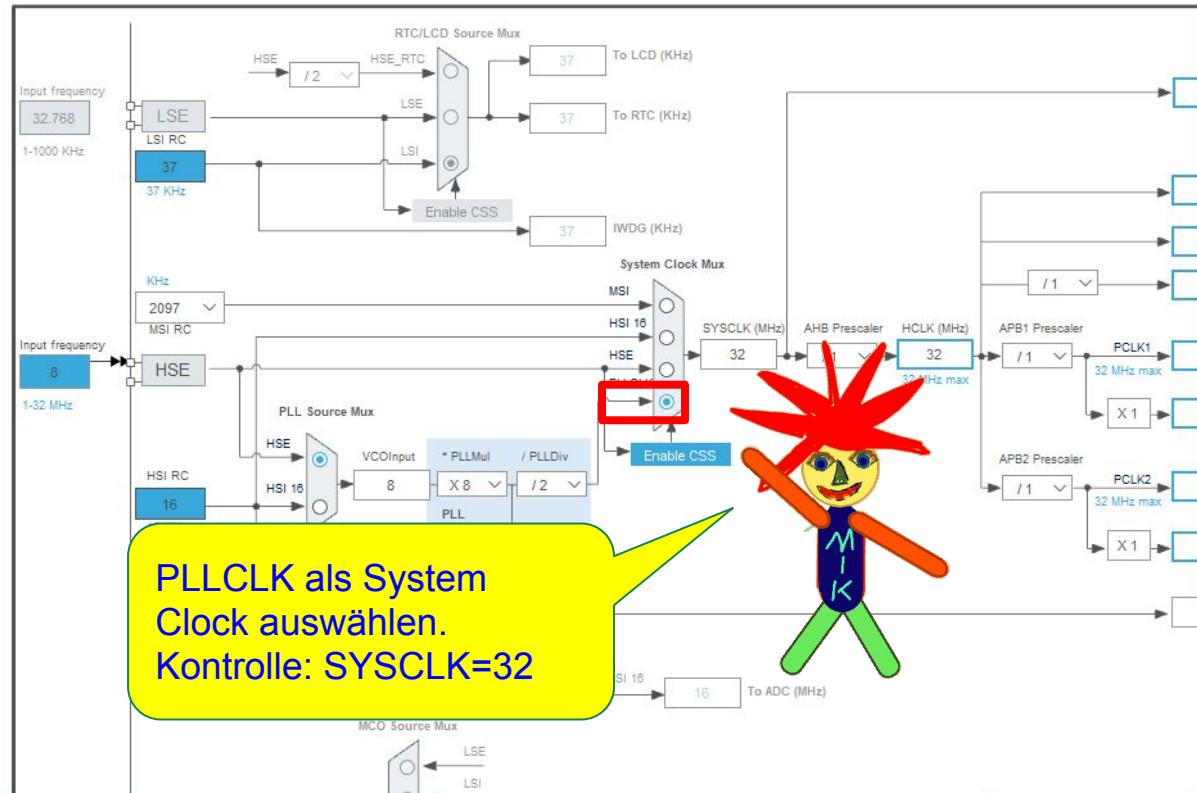
High Speed External (HSE)
als Quelle für die
PLL-Schaltung wählen.

Die PLL-Schaltung kann aus
den eingespeisten 8MHz
intern 32MHz erzeugen.
Der Prozessor kann mit
32MHz laufen. Wir wollen
nichts verschenken



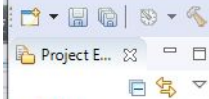


X8 und /2 einstellen
 $8\text{MHz} \cdot 8 / 2 = 32\text{ MHz}$



PLLCLK als System
Clock auswählen.
Kontrolle: SYSCLK=32





*erstes.ioc

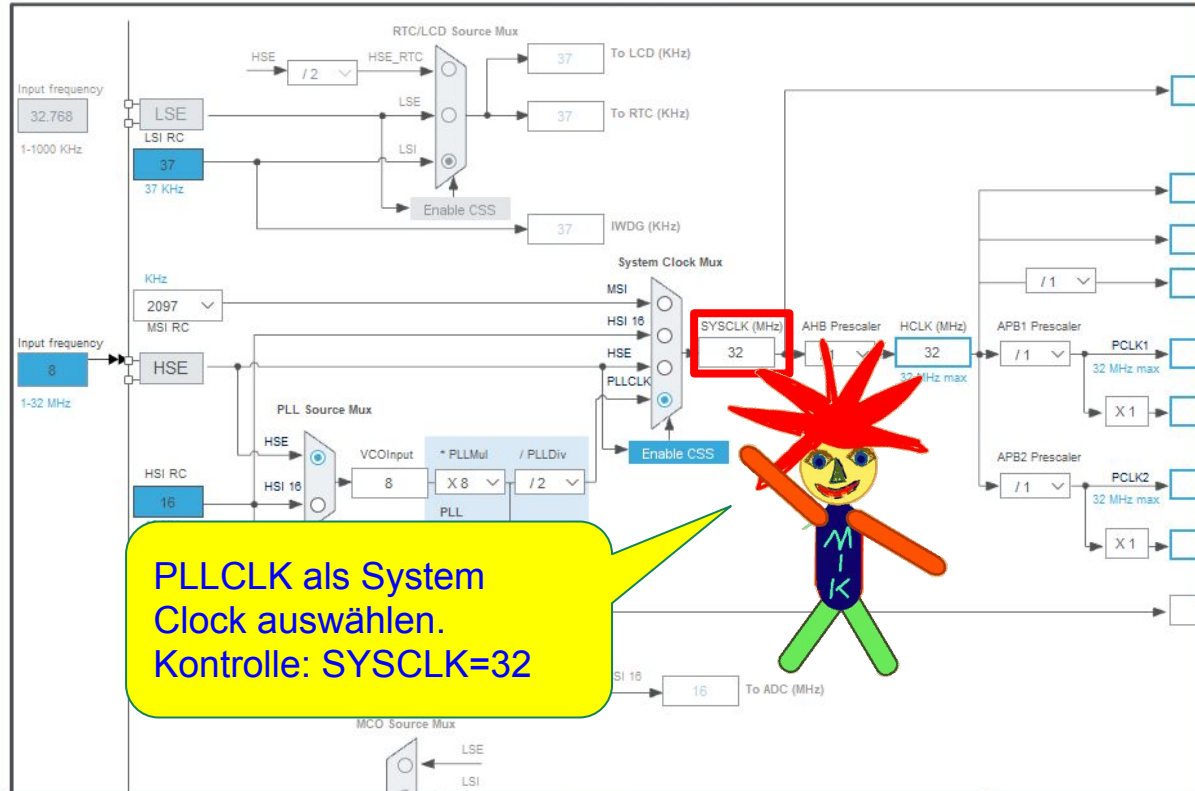
Pinout & Configuration

Clock Configuration

Project Manager

Tools

Resolve Clock Issues



PLLCLK als System
Clock auswählen.
Kontrolle: SYSCLK=32



Quick Access

Workspace1 - Device Configuration Tool - STM32CubeIDE

File Edit Navigate Search Project Run Window Help

Quick Access

Project Explorer

- erstes
- Includes
- Core
- Drivers
- erstes.ioc
- STM32L152RETX_FL
- STM32L152RETX_RA

Pinout & Configuration

Resolve Clock Issues

Zurück zu Pinout & Configuration

The diagram illustrates the clock configuration for the STM32L152RETX. It shows the following components and connections:

- Input frequencies:** 32.768 KHz (LSE), 1-1000 KHz (LSI RC), 2097 KHz (MSI RC), 1-32 MHz (HSE), 16 MHz (HSI RC).
- Clock sources:** LSE, LSI, HSE, MSI, HSI, PLL.
- System clocks:** SYSCLK (32 MHz), HCLK (32 MHz max), PCLK1 (32 MHz max), PCLK2 (32 MHz max).
- Peripherals:** To LCD (KHz), To ADC (MHz), To ADC (MHz), To ADC (MHz).
- Configuration:** The PLL Source Mux is set to HSE. The PLL is configured with a multiplier of 8 and a divider of 2. The HSI is configured with a multiplier of 8 and a divider of 2. The HSE is configured with a multiplier of 1 and a divider of 1. The HSI is configured with a multiplier of 1 and a divider of 1.

Workspace1 - Device Configuration Tool - STM32CubeIDE

File Edit Navigate Search Project Run Window Help

Project Explorer

- erstes
- Includes
- Core
- Drivers
- erstes.ioc
- STM32L152RETX_FLASH.I
- STM32L152RETX_RAM.I

Pinout & Configuration

Additional Software

Pinout

RCC Mode and Configuration

Mode

High Speed Clock (HSE) BYPASS Clock Sou...

Low Speed Clock (LSE) Disable

☐ Master Clock Output

Categories A-Z

- System Core
- Analog
- Timers
- Connectivity
- Multimedia
- Computing
- Middleware

Reset Configuration

- NVIC Settings
- Parameter Settings
- User Co...

Configure the below parameters :

Search (Ctrl+H)

System Parameters

- VDD voltage (V) 3.3 V
- Instruction Cache Enabled
- Prefetch Buffer Disabled
- Data Cache Enabled
- Flash Latency (WS) 1 WS (2 CPU cycle

Pinout view

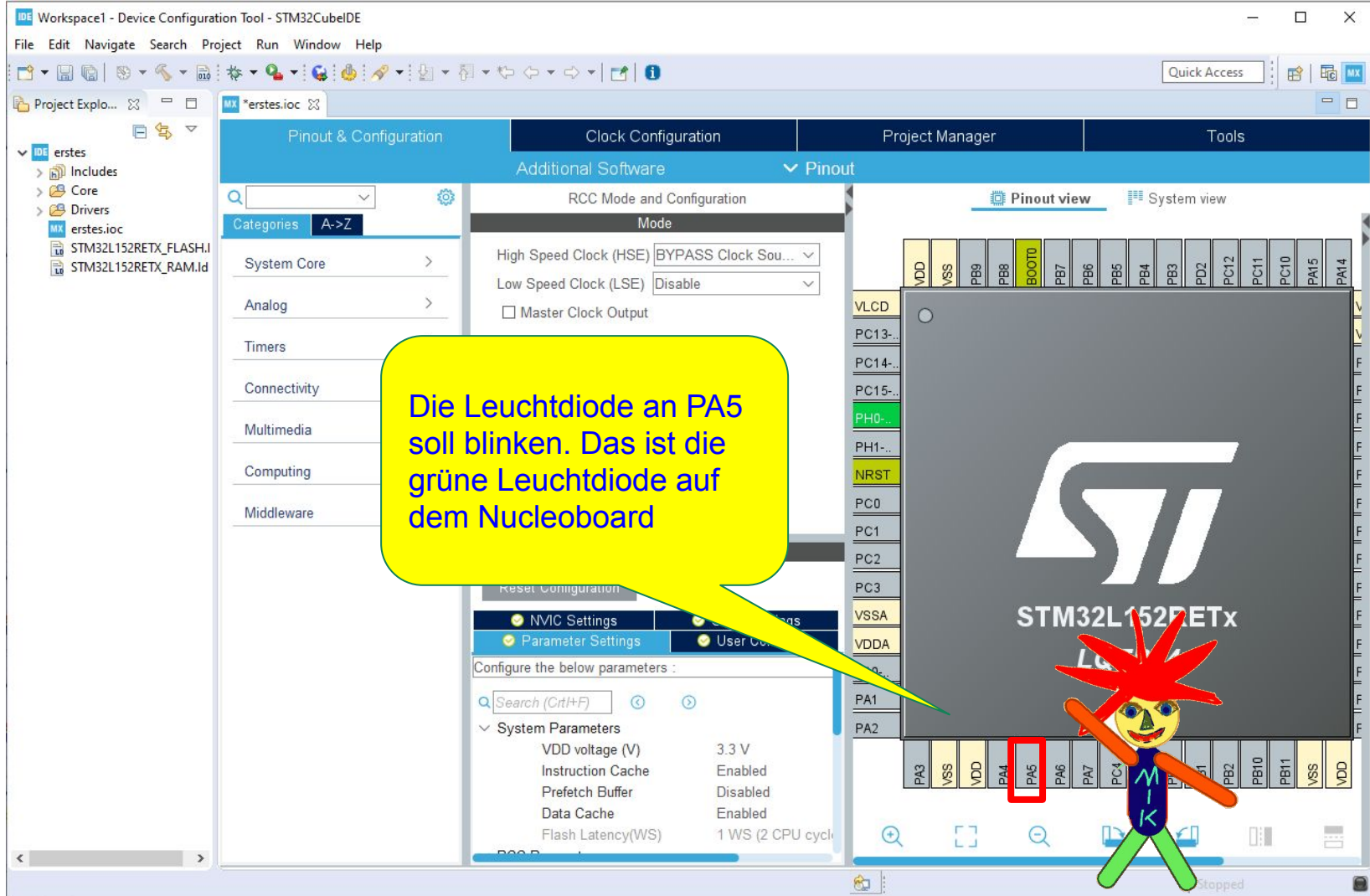
System view

Pinout view

STM32L152RETx

PA5

Die Leuchtdiode an PA5 soll blinken. Das ist die grüne Leuchtdiode auf dem Nucleoboard



Workspace1 - Device Configuration Tool - STM32CubeIDE

File Edit Navigate Search Project Run Window Help

Project Explorer

- erstes
- Includes
- Core
- Drivers
- erstes.ioc
- STM32L152RETx_FLASH.I
- STM32L152RETx_RAM.Id

Pinout & Configuration

Additional Software

Pinout

RCC Mode and Configuration

Mode

High Speed Clock (HSE) BYPASS Clock Sou...

Low Speed Clock (LSE) Disable

☐ Master Clock Output

Categories A-Z

- System Core
- Analog
- Timers
- Connectivity
- Multimedia
- Computing
- Middleware

Reset Configuration

- NVIC Settings
- GPIO Settings
- Parameter Settings
- User Constants

Configure the below parameters :

Search (Ctrl+F)

System Parameters

- VDD voltage (V) 3.3 V
- Instruction Cache Enabled
- Prefetch Buffer Disabled
- Data Cache Enabled
- Flash Latency(WS) 1 WS (2 CPU cycle)


Pinout view

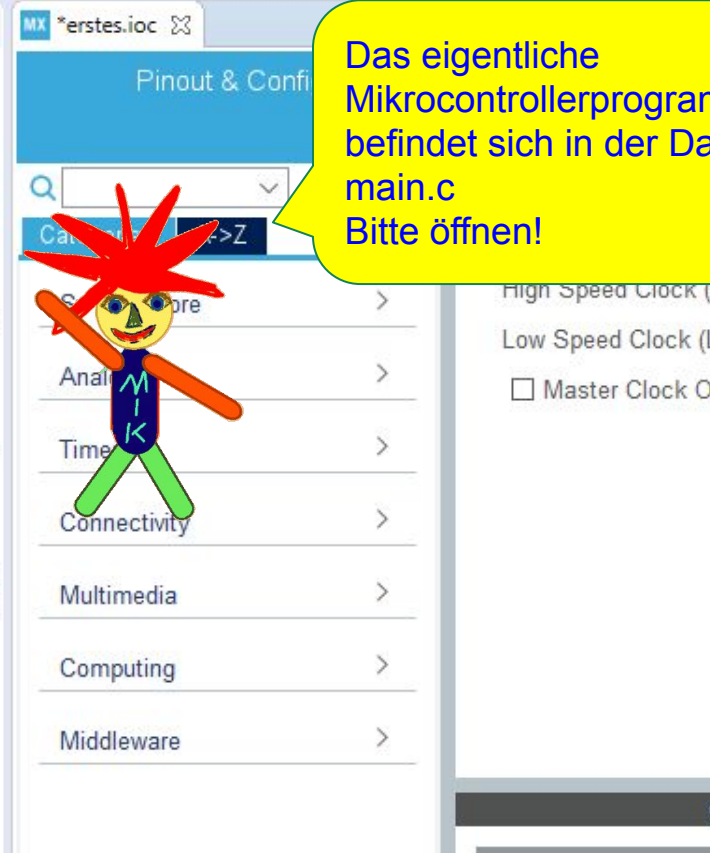
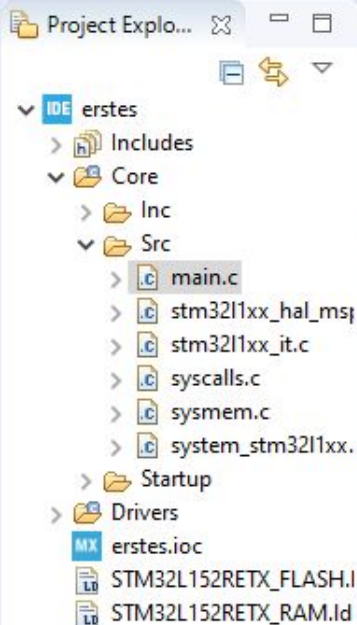
System view

STM32L152RETx LQFP64

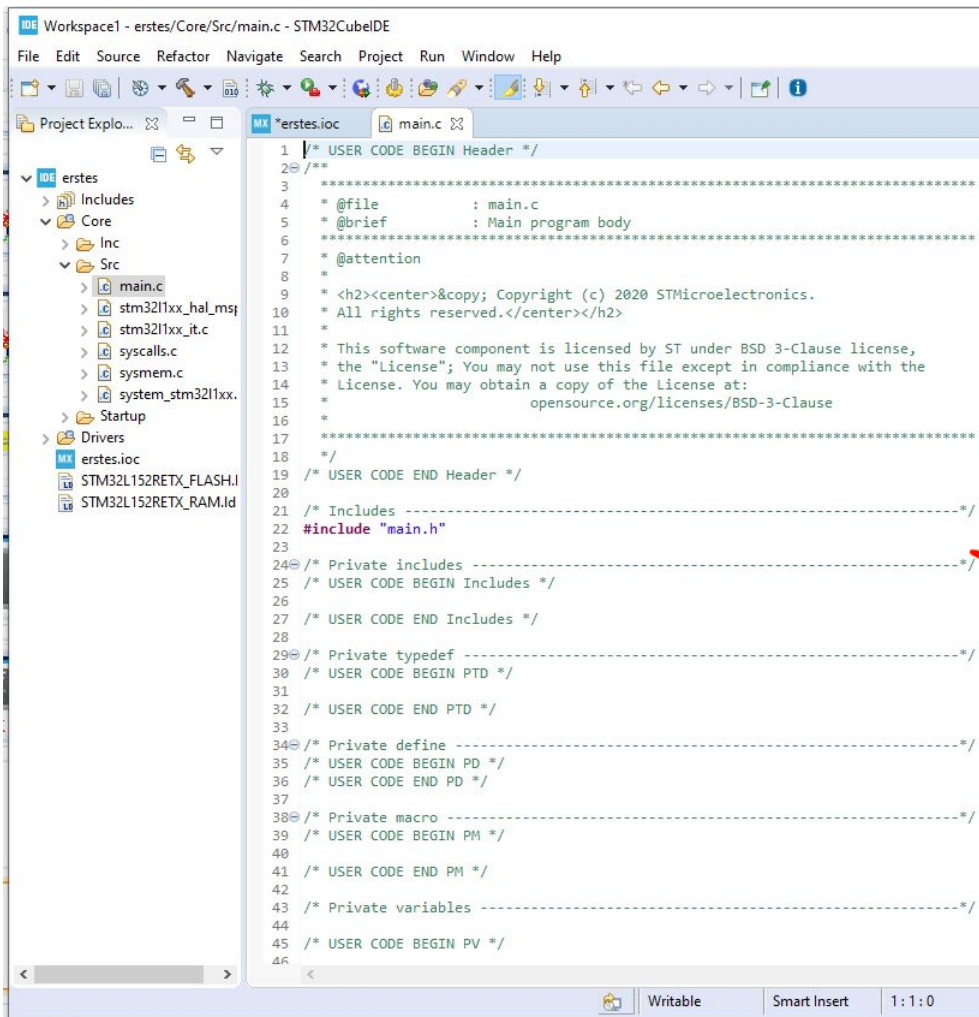
GPIO_Output

Als nächstes kann das Blinkprogramm geschrieben werden. Aber wo?



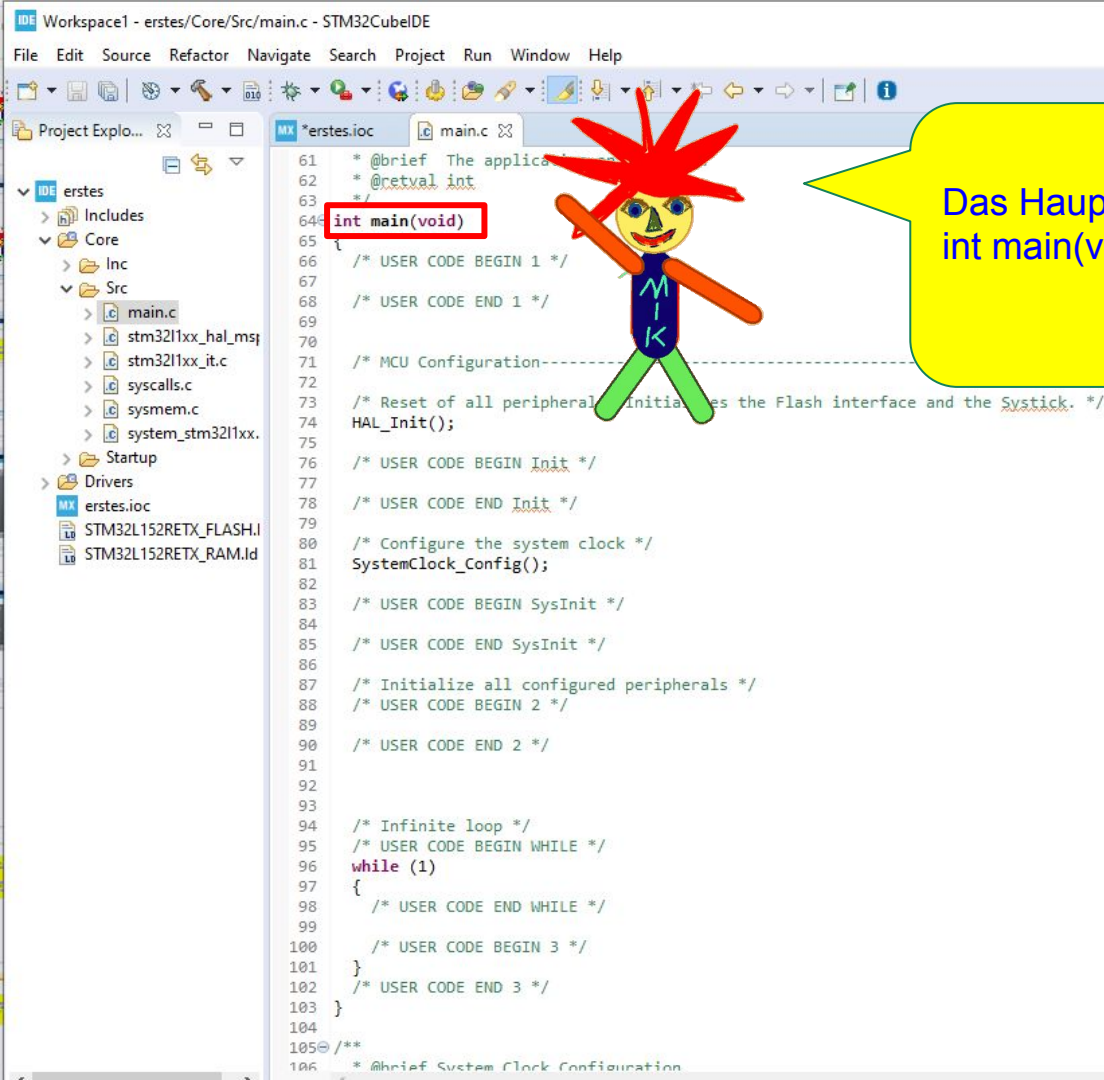


Das eigentliche
Mikrocontrollerprogramm
befindet sich in der Datei
main.c
Bitte öffnen!

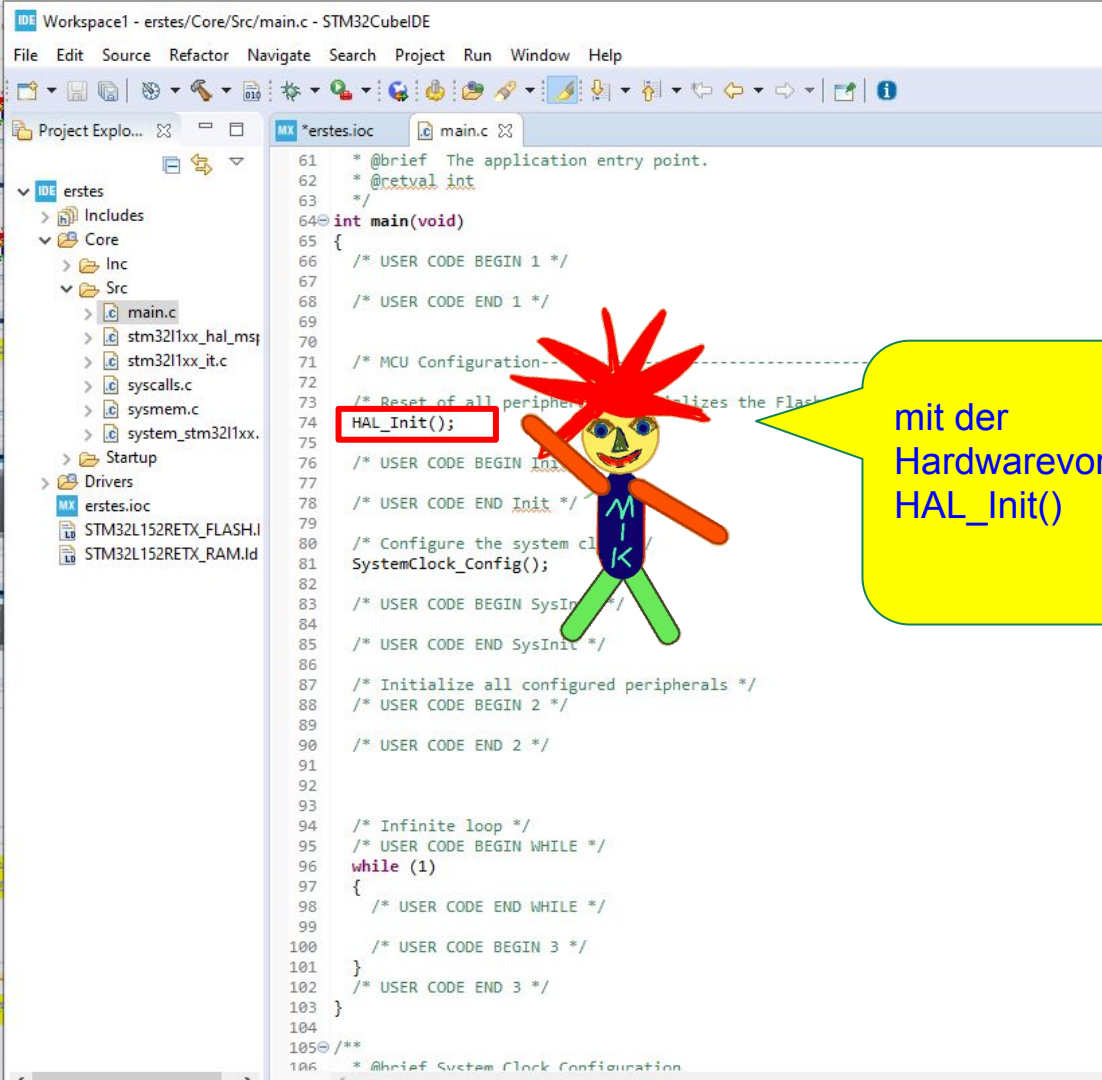


Hier finden sich
allerhand Kommentare
und einzelne
Codezeilen.
Das Programmgerüst,
erzeugt von der IDE
Herunterscrollen!





Das Hauptprogramm
int main(void)



mit der
Hardwarevoreinstellung
HAL_Init()

Workspace1 - erstes/Core/Src/main.c - STM32CubeIDE

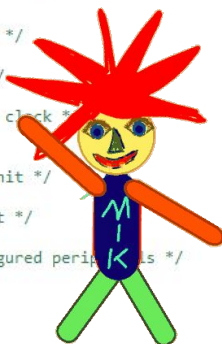
File Edit Source Refactor Navigate Search Project Run Window Help

Project Explorer

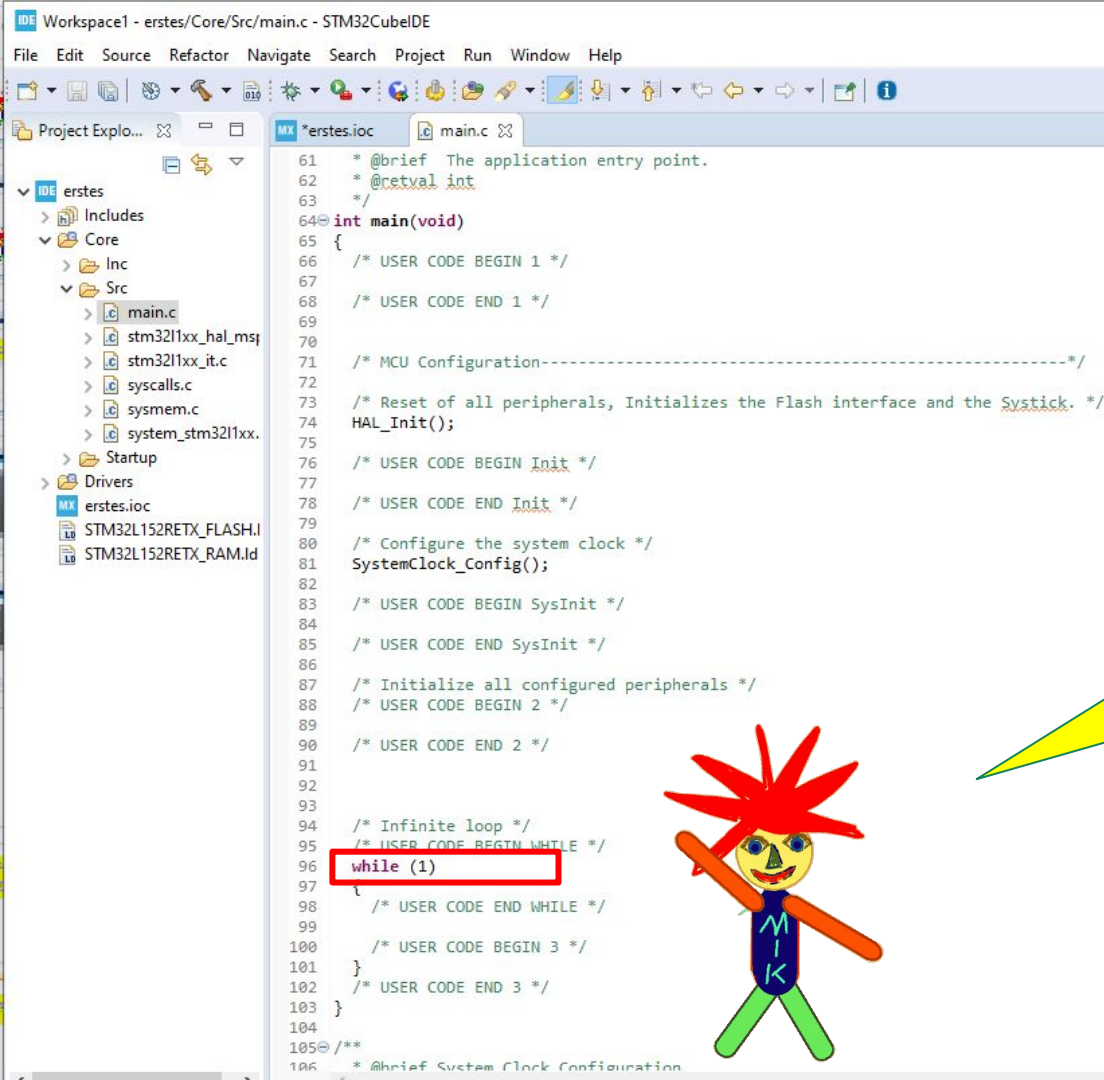
- IDE erstes
 - Includes
 - Core
 - Inc
 - Src
 - main.c
 - stm32l1xx_hal_msp
 - stm32l1xx_it.c
 - syscalls.c
 - sysmem.c
 - system_stm32l1xx
 - Startup
 - Drivers
 - erstes.ioc
 - STM32L152RETX_FLASH.I
 - STM32L152RETX_RAM.Id

main.c

```
61  * @brief The application entry point.
62  * @retval int
63  */
64  int main(void)
65  {
66      /* USER CODE BEGIN 1 */
67
68      /* USER CODE END 1 */
69
70
71      /* MCU Configuration-----*/
72
73      /* Reset of all peripherals, Initializes the Flash interface and the Systick. */
74      HAL_Init();
75
76      /* USER CODE BEGIN Init */
77
78      /* USER CODE END Init */
79
80      /* Configure the system clock */
81      SystemClock_Config();
82
83      /* USER CODE BEGIN SysInit */
84
85      /* USER CODE END SysInit */
86
87      /* Initialize all configured peripherals */
88      /* USER CODE BEGIN 2 */
89
90      /* USER CODE END 2 */
91
92
93
94      /* Infinite loop */
95      /* USER CODE BEGIN WHILE */
96      while (1)
97      {
98          /* USER CODE END WHILE */
99
100         /* USER CODE BEGIN 3 */
101     }
102     /* USER CODE END 3 */
103 }
104
105 /**
106  * @brief System Clock Configuration
```



mit unserer
Takteinstellung
SystemClock_Config()



Der Endlosschleife
while(1)

Workspace1 - erstes/Core/Src/main.c - STM32CubeIDE

File Edit Source Refactor Navigate Search Project Run Window Help

Project Explorer

- IDE erstes
 - Includes
 - Core
 - Inc
 - Src
 - main.c
 - stm32l1xx_hal_msp
 - stm32l1xx_it.c
 - sycalls.c
 - sysmem.c
 - system_stm32l1xx
 - Startup
 - Drivers
 - erstes.ioc
 - STM32L152RETX_FLASH.Id
 - STM32L152RETX_RAM.Id

main.c

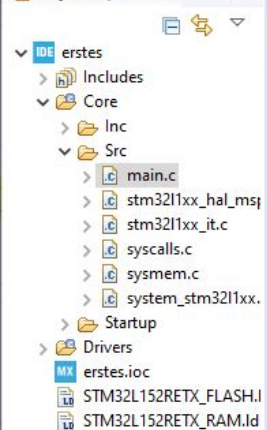
```
61  * @brief The application entry point.
62  * @retval int
63  */
64  int main(void)
65  {
66      /* USER CODE BEGIN 1 */
67
68      /* USER CODE END 1 */
69
70
71      /* MCU Configuration-----*/
72
73      /* Reset of all peripherals, Initializes the Flash interface and the Systick. */
74      HAL_Init();
75
76      /* USER CODE BEGIN Init */
77
78      /* USER CODE END Init */
79
80      /* Configure the system clock */
81      SystemClock_Config();
82
83      /* USER CODE BEGIN SysInit */
84
85      /* USER CODE END SysInit */
86
87      /* Initialize all configured peripherals */
88      /* USER CODE BEGIN 2 */
89
90      /* USER CODE END 2 */
91
92
93
94      /* Infinite loop */
95      /* USER CODE BEGIN WHILE */
96      while (1)
97      {
98          /* USER CODE END WHILE */
99
100         /* USER CODE BEGIN 3 */
101     }
102     /* USER CODE END 3 */
103 }
104
105 /**
106  * @brief System Clock Configuration
```

Wir sollen die Bereiche mit USER CODE BEGIN USER CODE END für unsere Eingaben verwenden. Diese Bereiche werden vom Codegenerator, falls z.B. Die Konfiguration nochmals geändert werden muss, nicht angerührt.





Project Explorer



```

61  * @brief The application entry point.
62  * @retval int
63  */
64  int main(void)
65  {
66      /* USER CODE BEGIN 1 */
67
68      /* USER CODE END 1 */
69
70
71      /* MCU Configuration-----*/
72
73      /* Reset of all peripherals, Initializes the Flash interface and the Systick. */
74      HAL_Init();
75
76      /* USER CODE BEGIN Init */
77
78      /* USER CODE END Init */
79
80      /* Configure the system clock */
81      SystemClock_Config();
82
83      /* USER CODE BEGIN SysInit */
84
85      /* USER CODE END SysInit */
86
87      /* Initialize all configured peripherals */
88      /* USER CODE BEGIN 2 */
89
90      /* USER CODE END 2 */
91
92
93
94      /* Infinite loop */
95      /* USER CODE BEGIN WHILE */
96      while (1)
97      {
98          GPIOA->ODR=GPIOA->ODR^GPIO_PIN_5;
99
100         /* USER CODE END WHILE */
101
102         /* USER CODE BEGIN 3 */
103     }
104     /* USER CODE END 3 */
105 }

```



Bit 5 von GPIOA->ODR
wird negiert (^ = XOR)

Workspace1 - erstes/Core/Src/main.c - STM32CubeIDE

File Edit Source Refactor Navigate Search Project Run Window Help

Project Explorer

- erstes
 - Includes
 - Core
 - Inc
 - Src
 - main.c
 - stm32l1xx_hal_msp
 - stm32l1xx_it.c
 - syscalls.c
 - sysmem.c
 - system_stm32l1xx.
 - Startup
 - Drivers
 - erstes.ioc
 - STM32L152RETX_FLASH.ld
 - STM32L152RETX_RAM.ld

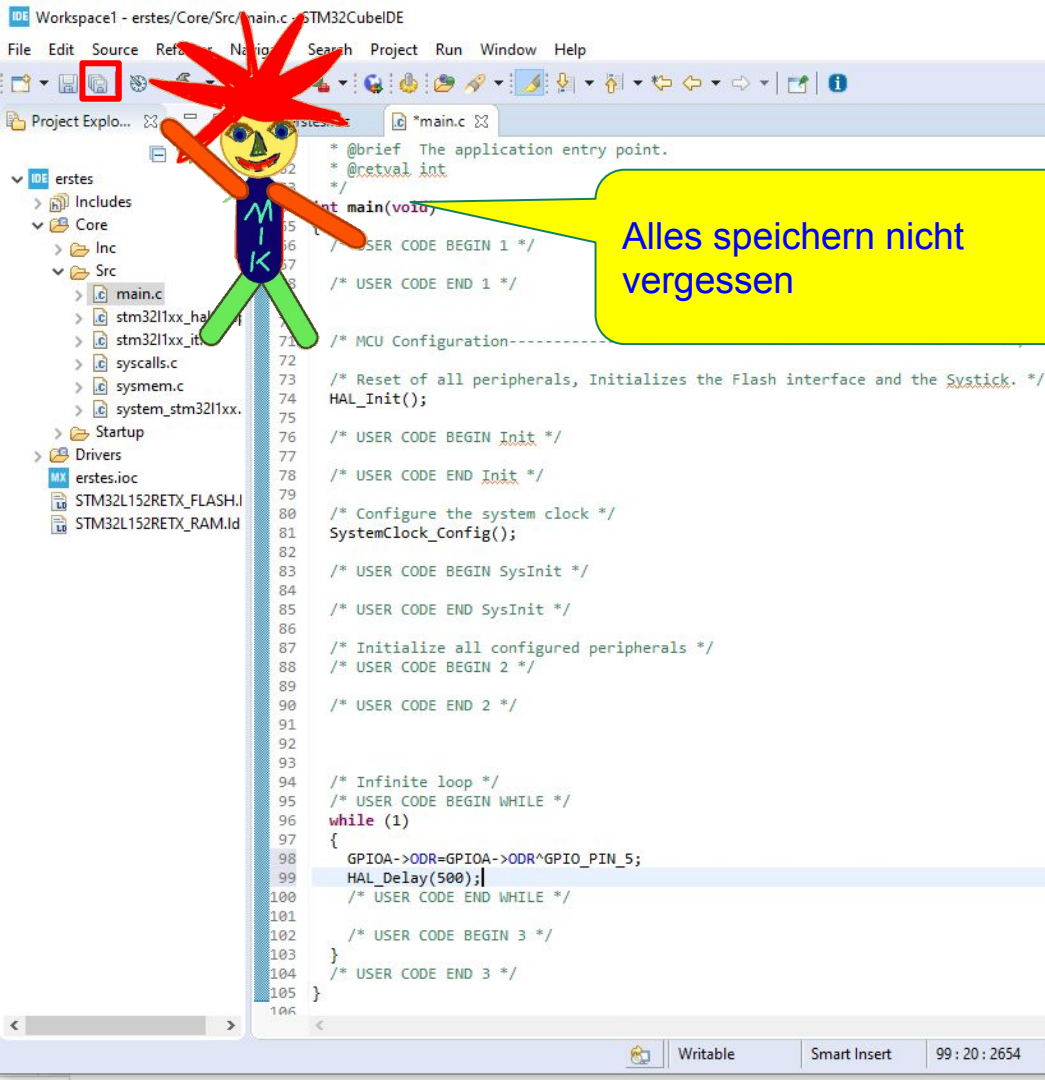
erstes.ioc

```
61  * @brief The application entry point.
62  * @retval int
63  */
64  int main(void)
65  {
66      /* USER CODE BEGIN 1 */
67
68      /* USER CODE END 1 */
69
70      /* MCU Configuration-----*/
71
72      /* Reset of all peripherals, Initializes the Flash interface and the Systick. */
73      HAL_Init();
74
75      /* USER CODE BEGIN Init */
76
77      /* USER CODE END Init */
78
79      /* Configure the system clock */
80      SystemClock_Config();
81
82      /* USER CODE BEGIN SysInit */
83
84      /* USER CODE END SysInit */
85
86      /* Initialize all configured peripherals */
87      /* USER CODE BEGIN 2 */
88
89      /* USER CODE END 2 */
90
91
92
93      /* Infinite loop */
94      /* USER CODE BEGIN WHILE */
95      while (1)
96      {
97          /* USER CODE BEGIN WHILE */
98          HAL_Delay(500);
99          /* USER CODE END WHILE */
100
101      /* USER CODE BEGIN 3 */
102
103      }
104      /* USER CODE END 3 */
105
106  }
```

Writable Smart Inspector 99:20:2654



500ms warten =>
Blinkfrequenz 1Hz
Programm starten



Workspace1 - erstes/Core/Src/main.c - STM32CubeIDE

File Edit Source Refactor Navigate Search Project Run Window

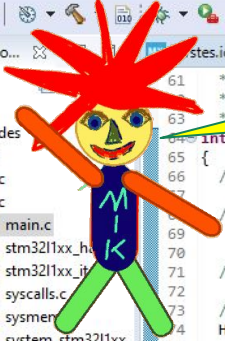
Project Explorer

- ▼ IDE erstes
 - > Includes
 - ▼ Core
 - > Inc
 - ▼ Src
 - main.c
 - stm32l1xx_h...
 - stm32l1xx_it...
 - syscalls.c
 - sysmem
 - system_stm32l1xx.
 - > Startup
 - > Drivers
 - erstes.ioc
 - STM32L152RETX_FLASH.I
 - STM32L152RETX_RAM.I

Rechtsklick auf das Projekt z.B. hier erstes

```
61  * @brief The appli
62  * @retval int
63  */
64  int main(void)
65  {
66      /* USER CODE BEGIN 1 */
67
68      /* USER CODE END 1 */
69
70      /* MCU Configuration-----*/
71
72      /* Reset of all peripherals, Initializes the Flash interface and the Systick. */
73      HAL_Init();
74
75      /* USER CODE BEGIN Init */
76
77      /* USER CODE END Init */
78
79      /* Configure the system clock */
80      SystemClock_Config();
81
82      /* USER CODE BEGIN SysInit */
83
84      /* USER CODE END SysInit */
85
86      /* Initialize all configured peripherals */
87      /* USER CODE BEGIN 2 */
88
89      /* USER CODE END 2 */
90
91
92
93      /* Infinite loop */
94      /* USER CODE BEGIN WHILE */
95      while (1)
96      {
97          GPIOA->ODR=GPIOA->ODR^GPIO_PIN_5;
98          HAL_Delay(500);
99          /* USER CODE END WHILE */
100
101          /* USER CODE BEGIN 3 */
102      }
103      /* USER CODE END 3 */
104
105  }
```

Writable Smart Insert 99 : 20 : 2654



Debug As
STM32 Cortex-M C/C++ Application

Project Explorer

- erstes
 - Includes
 - Core
 - Inc
 - Src
 - main.c
 - stm32l1xx_hal
 - stm32l1xx_it
 - syscalls.c
 - system
 - system_stm32l1xx
 - Startup
 - Drivers
 - erstes.ioc
 - STM32L152RETx_FLASH.ld
 - STM32L152RETx_RAM.ld

main.c

```
61  * @brief The application entry point.
62  * @retval None
63  */
64  int main(void)
65  {
66      /* USER CODE BEGIN 1 */
67
68      /* USER CODE END 1 */
69
70      /* MCU Configuration-----*/
71
72      /* Reset of all peripherals, Initializes the Flash interface and the Systick. */
73      HAL_Init();
74
75      /* USER CODE BEGIN Init */
76
77      /* USER CODE END Init */
78
79      /* Configure the system clock */
80      SystemClock_Config();
81
82      /* USER CODE BEGIN SysInit */
83
84      /* USER CODE END SysInit */
85
86      /* Initialize all configured peripherals */
87      /* USER CODE BEGIN 2 */
88
89      /* USER CODE END 2 */
90
91
92
93      /* Infinite loop */
94      /* USER CODE BEGIN WHILE */
95      while (1)
96      {
97          GPIOA->ODR=GPIOA->ODR^GPIO_PIN_5;
98          HAL_Delay(500);
99      }
100      /* USER CODE END WHILE */
101
102      /* USER CODE BEGIN 3 */
103  }
104  /* USER CODE END 3 */
105
106
```

Writable Smart Insert 99 : 20 : 2654

Edit launch configuration properties



Name: erstes Debug

Main Debugger Startup Source Common

C/C++ Application:

Debug\erstes.elf

Search Project...

Browse...

Project:

erstes

Browse...

Build (if required) before launching

[Build Configuration:](#) Select Automatically☐ Enable auto build☐ Disable auto build☒ Use workspace settings[Configure Workspace Settings...](#)

Revert

Apply

OK

Cancel

OK



```
f The application entry point.
```

```
al int
```

```
void)
```

```
CODE BEGIN
```

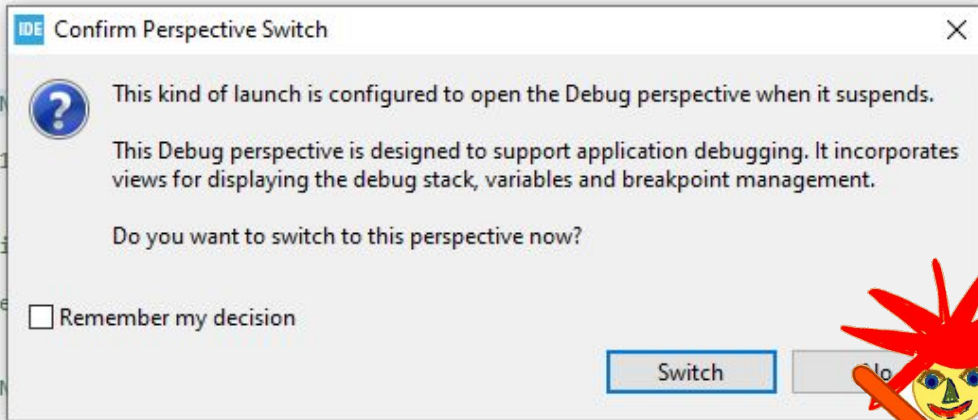
```
CODE END 1
```

```
Configurati
```

```
t of all pe  
t());
```

```
CODE BEGIN
```

```
CODE END Init */
```

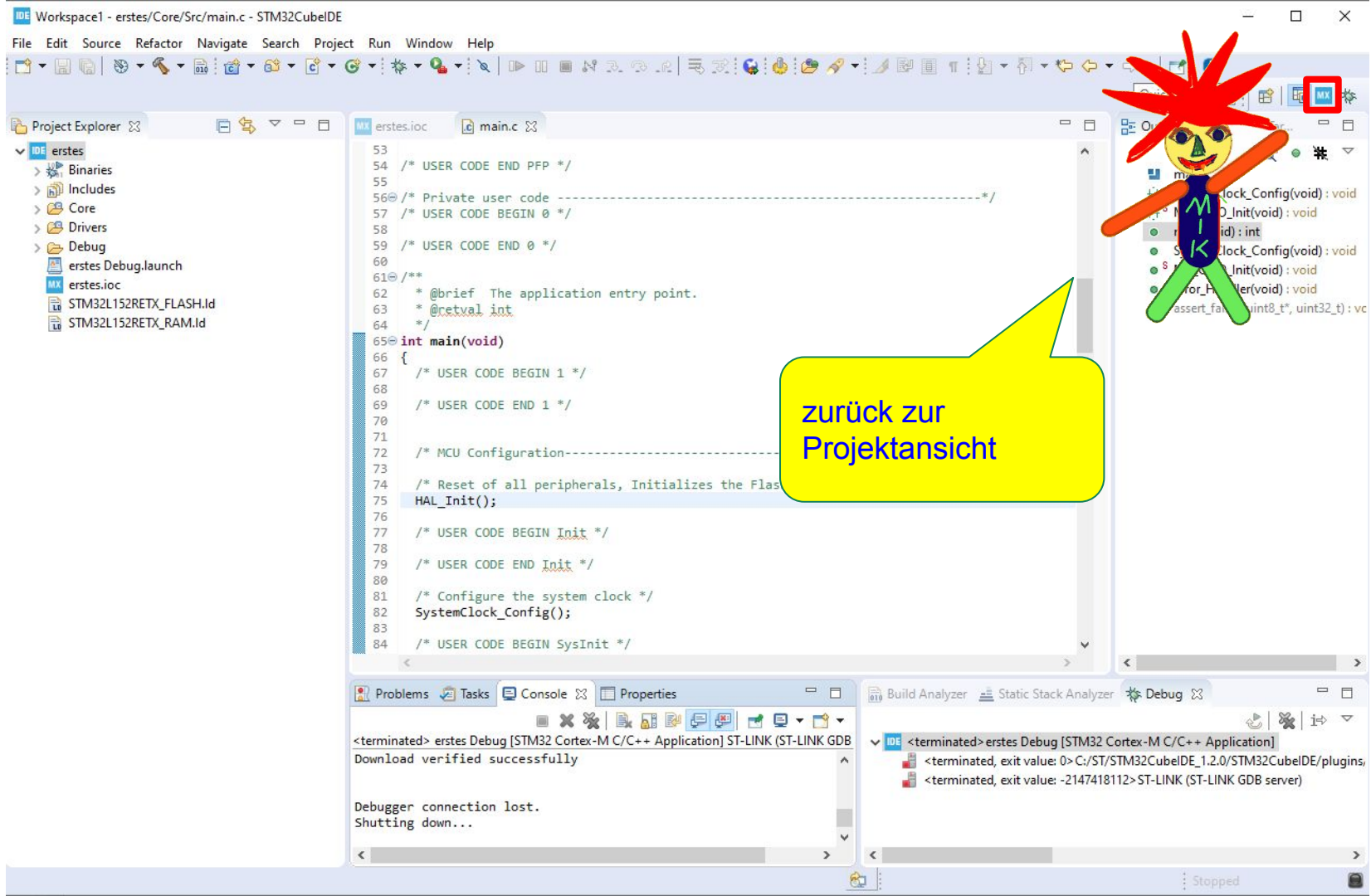


Wir wechseln in die
Debug-Ansicht => Switch

[illegible]

Download verified successfully





Getting Started with STM32CubeIDE



Die Entwicklungsumgebung

Ich bin Mik, Dein Mikrocontroller

