Application Note

ÚTIA Akademie věd České republiky
Ústav teorie informace a automatizace AV ČR, v.v.i.

Support for STM32H573I-DK web server

Jiří Kadlec, Zdeněk Pohl, Lukáš Kohout
kadlec@utia.cas.cz, zdenek.pohl@utia.cas.cz, kohoutl@utia.cas.cz

Revision history

Rev.	Date	Author	Description
v01	9.2.2024	J.K.	Initial release
v02	14.2.2024	J.K.	Fixed typos
v03	16.2.2025	J.K.	Updated references

Contents

Acknowledgement

The EECONE project is supported by the Chips Joint Undertaking and its members, including the top-up funding by National Funding Authorities from involved countries under grant agreement no. 101112065.

https://zs.utia.cas.cz/index.php?ids=projects/eecone

https://eecone.com/eecone/home/

1 Introduction

EECONE project *https://eecone.com/eecone/home/* work package 4, task 4.3 is investigating measures to support second life of electronics due to modular design.

Work package 4 task 4.4 is investigating measures to support extension of life of electronics due to methodology of support used custom platform to adapt for the in-time-evolving design tools and embedded Linux PetaLinux operating system.

UTIA AV CR, v.v.i. (Institute of Information Theory and Automation of the Czech Academy of Sciences, in short UTIA) is not-for profit research institute located in Prague, Czech Republic. UTIA is involved as partner in both tasks, T4.3 and T4.4.

Both EECONE tasks require specification of comparable reference systems which are based on modular HW with potential for “second life” by reuse of modules or use cost optimized PCB HW without modularity.

Systems (with HW modularity or low cost single PCB) should be capable to perform similar challenging tasks. Systems have to be capable to accelerate in HW AI inference algorithms with video camera input for edge application like person detection, face detection, car-make or car-type detection and graphical output to on X11 desktop of an remote PC connected by wired Ethernet in a local network.

Systems should also support remote monitoring and remote user control from a PC connected by wired Ethernet in a local network.

The investigated measures and methodologies to support “second life” of electronic modules (T4.3) and measures to support extension of life of electronics (T4.4) due to methodology of support used custom platform to adapt for the in-time-evolving design tools and embedded Linux PetaLinux operating system. We target developers designing the final commercial, AI inference based edge applications, mainly in the area of home automation.

Based on these requirements UTIA have selected two types of systems:
- Low cost systems. See [2], [3]
- Modul based systems. See [4] and [5].

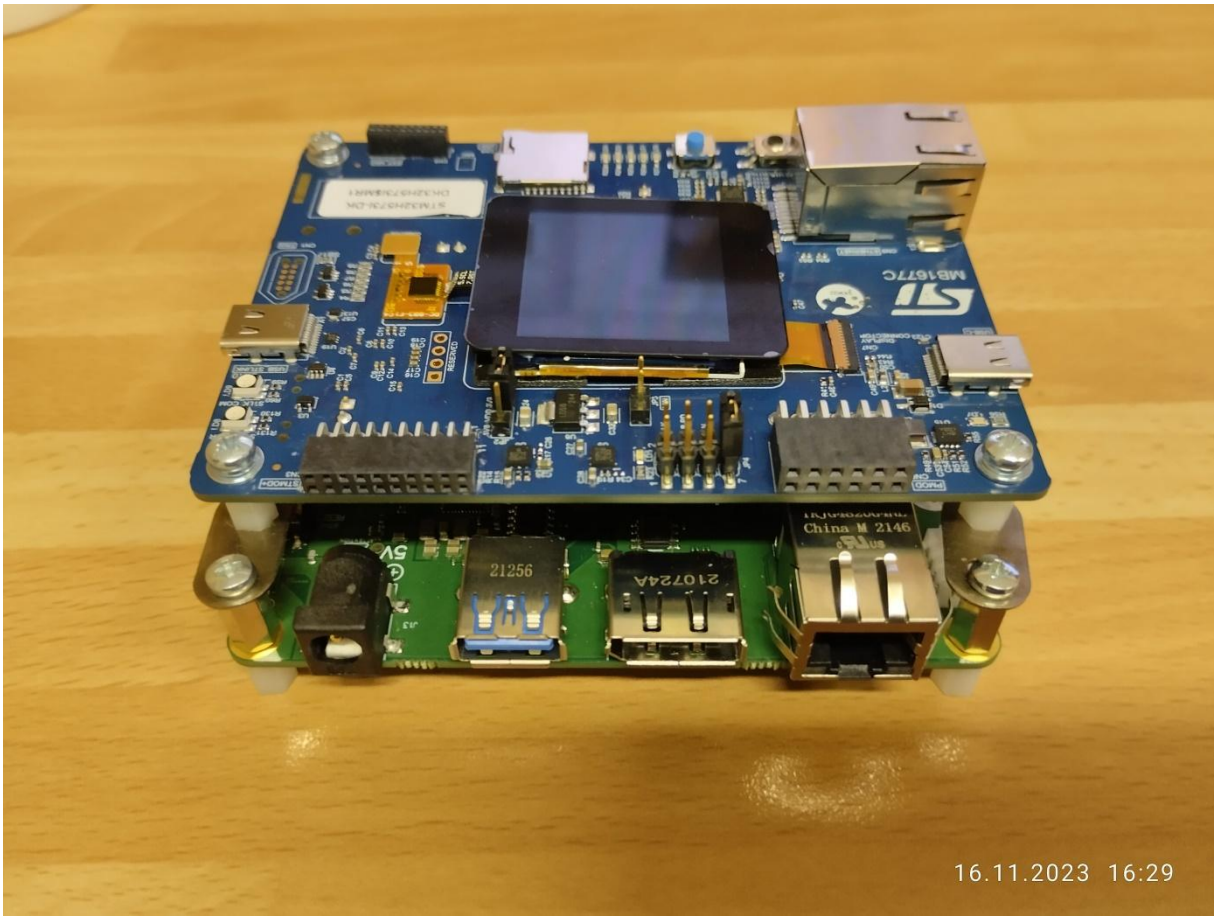
Both compared types of systems use STMicroelectronic STM32H573I-DK board for:
- local system control on small graphical touch screen display
- remote system control from www browser based on www-server or secure communication based on mqtt client. Board is supported by STMicroelectronic CubeMX SW framework and also by NetXDuo SW framework on top of ThreadX OS and FileX SW package.

The MCU used on STM32H573I-DK board is a 40nm chip with 32 bit ARM M33 MCU operating with 250 MHz clock, 2 MBytes of program flash memory and 640 KBytes of RAM.

Compared systems use 16nm AMD ZynqUltrascale+ device with 64 bit ARM A53 Microprocessor and programmable logic in the same device and Petalinux OS.
- Low-cost systems have an AMD ZynqUltrascale+ device and DDR4 with all peripheral interfaces soldered on a single, low cost PCB
- Module-based systems have an AMD ZynqUltrascale+ device and DDR4 soldered on an 4x5 cm module connected by connectors to a carrier board with all peripheral interfaces

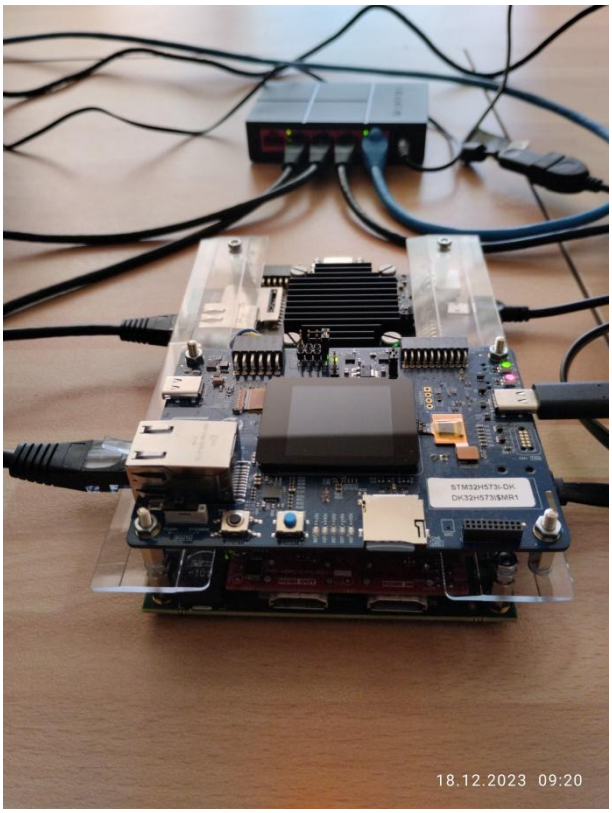
1.1 Low cost systems used by UTIA in EECONE T4.3 and T4.4

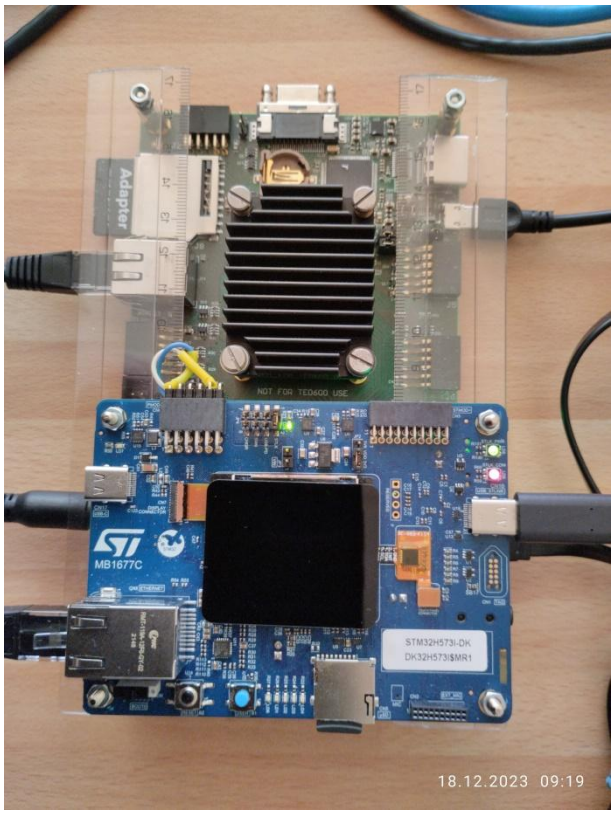
[1]	STM32H573I-DK	https://www.st.com/en/evaluation-tools/stm32h573i-dk.html	Local or remote system control (www-server or secure mqtt client) for [2], [3]
[2]	TE0802-02-1BEV2-A	https://shop.trenz-electronic.de/en/TE0802-02-1BEV2-A-MPSoC-Development-Board-with-AMD-Zynq-UltraScale-ZU1EG-and-1-GB-LPDDR4?c=474	AMD Vitis AI 3.0 AMD DPU in PL USB camera, remote X11 desktop
[3]	TE0802-02-2AEV2-A	MPSoC Development Board mit AMD Zynq™ UltraScale+™ ZU2 und 1 GB LPDDR4 | Trenz Electronic GmbH Online Shop (EN) (trenz-electronic.de)	AMD Vitis AI 3.0 AMD DPU in PL USB camera, remote X11 desktop



1.2 Module based systems used by UTIA in EECONE T4.3 and T4.4

[1]	STM32H573I-DK TE0701-06 Carrier Board for Trenz Electronic 4 x 5 Modules TE0821 or TE0820	https://www.st.com/en/evaluation-tools/stm32h573i-dk.html https://shop.trenz-electronic.de/en/TE0701-06-Carrier-Board-for-Trenz-Electronic-4-x-5-Modules?c=261	Local or remote system control (www-server or secure mqtt client) for 2-1, 2-2 Carrier Board for range of 4x5 cm modules [3], [4].
[4]	TE0821 Module: **17 module types** (to be supported)	https://shop.trenz-electronic.de/en/Products/Trenz-Electronic/TE08XX-Zynq-UltraScale/TE0821-Zynq-UltraScale/	AMD Vitis AI 3.0 AMD DPU in PL USB camera FULL HD HDMI display or remote X11 desktop
[5]	TE0820 Module: **100 module types** (to be supported)	https://shop.trenz-electronic.de/en/Products/Trenz-Electronic/TE08XX-Zynq-UltraScale/TE0821-Zynq-UltraScale/	AMD Vitis AI 3.0 AMD DPU in PL USB camera FULL HD HDMI display or remote X11 desktop





This application note [1] and the accompanying evaluation package describe web server running on STM32H573I-DK board. It serves for remote control and remote monitoring of systems [2], [3], [4] and [5]. It is available for free public download from UTIA server dedicated to UTIA contributions to EECONE project:
https://zs.utia.cas.cz/index.php?ids=projects/eecone

This application note [1] and the accompanying evaluation package will be also available for free public download in format of wiki tutorial on Trenz-Electronic wiki server:
https://wiki.trenz-electronic.de/display/PD/Vitis+AI+and+Vitis+Acceleration+Tutorials+with+Trenz+Electronic+Modules

1.3 Objective of this Application Note and Evaluation Package

This application note [1] and the accompanying evaluation package describe and support:

- Installation and use of STM32CubeIDE framework on Win 11 Pro PC.
- Power supply options for the STM32H573I-DK board.
- Description of the reference web server. Web server is using NetX Duo and FileX frameworks and ThredX OS. Web content is stored on STM32H573I-DK SD card. It is managed by FileX framework. It is using PC compatible FAT32 file system.
  It is using DHCP server of the local network to get the automatic IP address resolution.
- Description of modifications of the reference web server to implement simple HW reset for systems [2], [3], [4] and [5].
- Support for HW remote reset. It can be initiated by user of a web-page displayed on a web-browser client running on a remote PC. PC is connected to the same local Ethernet network. This PC is using same DHCP server to get IP address resolution.

2 Installation of STM Development Tools and Packages

2.1 Installation of STM32CubeIDE

Go to:

https://www.st.com/en/development-tools/stm32cubeide.html

Install the latest release of the Integrated Development Environment for STM32:

STM32CubeIDE
Version: 1.14.1
Build: 20064_20240111_1413 (UTC)
(C) 2024 STMicroelectronics ALL RIGHTS RESERVED

2.2 Installation of STM32CubeMX

Go to:

https://www.st.com/content/st_com/en/stm32cubemx.html

Install the latest release of the STM32Cube initialization code generator

STM32CubeMX
Version: 6.10.0

2.3 Installation of STM32CubeH5 Firmware Package

In STM32CubeMX, install STM32CubeH5 Firmware Package V1.1.1

STM32CubeH5 Firmware Package
Version: 1.1.1

2.4 STM32H573I-DK Discovery Kit Documentation

Go to:

https://www.st.com/en/evaluation-tools/stm32h573i-dk.html

and download STM documentation for STM32H573I-DK Discovery kit with STM32H573IIK3Q MCU.

Links to the most useful documents:

UM3065: Getting started with STM32CubeH5 for STM32H5 Series

UM3132: Description of STM32H5 HAL and low-layer driver

UM2298: STM32Cube BSP drivers development guidelines

Getting started with STM32H5 security

Security with STM32H5

3 Reference Web Server based on NetX Duo and FileX

The NetX Duo Web Server Appliction provides an example of Azure RTOS NetX Duo stack usage on STM32H573G-DK board. It shows how to develop Web HTTP server based application.

3.1 Nx_WebServer

The `Nx_WebServer` application is designed to load files and dynamic web pages stored in SD card using a Web HTTP server.

The reference code provides all required features to build a compliant Web HTTP Server.

The main entry function `tx_application_define()` is called by `ThreadX` during kernel start, at this stage, all `NetX` and `FileX` resources are created.

- A `NX_PACKET_POOL` **NxAppPool** is allocated
- A `NX_IP` instance **NetXDuoEthIpInstance** using that pool is initialized
- A `NX_PACKET_POOL` **WebServerPool** is allocated
- A `NX_WEB_HTTP_SERVER` **HTTPServer** instance using that pool is initialized
- The `ARP`, `ICMP` and protocols (`TCP` and `UDP`) are enabled for the `NX_IP` instance
- A `DHCP` client is created.

The application then creates 2 threads with different priorities:

- **NxAppThread**
  (priority 10, PreemtionThreashold 10) :
  created with the `TX_AUTO_START` flag to start automatically.

- **AppServerThread**
  (priority 5, PreemtionThreashold 5) :
  created with the `TX_DONT_START` flag to be started later.
- **LedThread**
  (priority 15, PreemtionThreashold 15) :
  created with the `TX_DONT_START` flag to be started later.

The **NxAppThread** starts and perform the following actions:
- Starts the DHCP client
- Waits for the IP address resolution
- Resumes the **AppServerThread**

The **AppServerThread**, once started:
- Fx_media_open.
- Starts `HTTP` server.
- Each command coming from client (browser) is treated on the callback `webserver_request_notify_callback`.

The **LedThread**, once resumed from the dashboard:
- Green Led is toggling.
- Messages are printed on PC `PuTTY` terminal.

3.2 Create and Compile Reference Nx_WebServer project

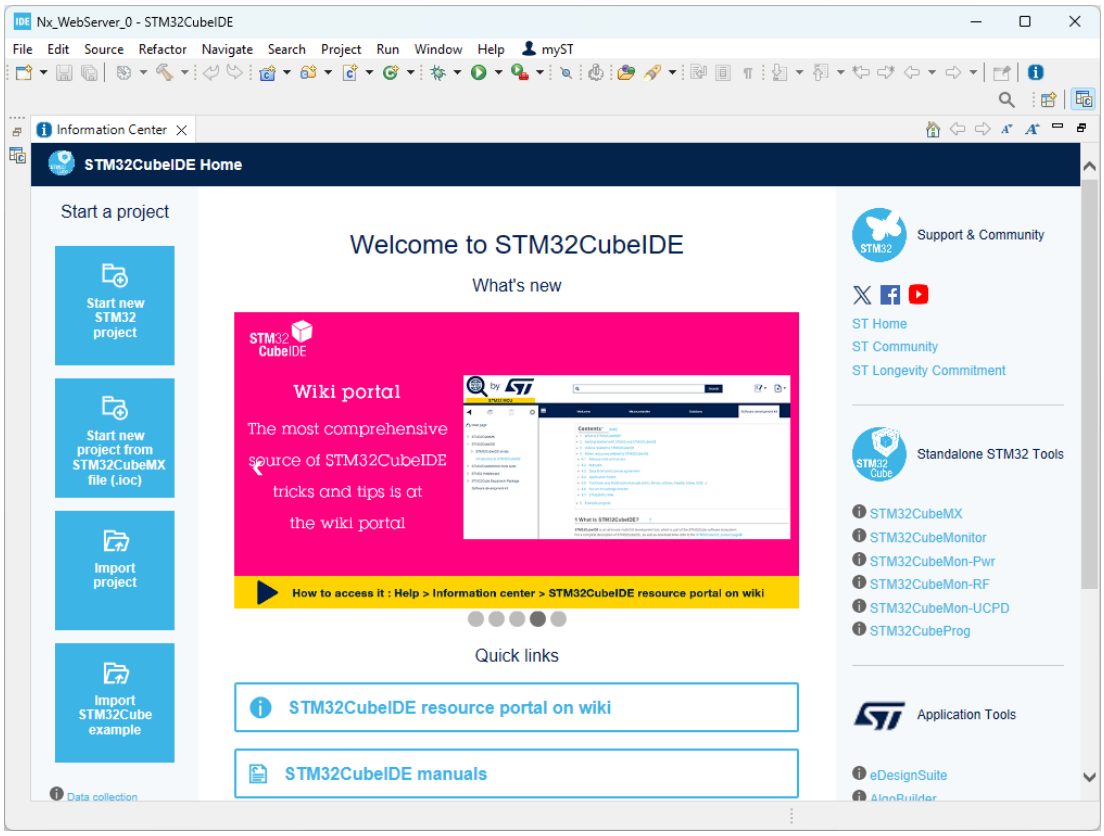
Copy reference Nx_WebServer project

```
<install_path>\STM32Cube\Repository\STM32Cube_FW_H5_V1.1.1\Projects\STM32H573I-DK\Applications\NetXDuo\Nx_WebServer\
```

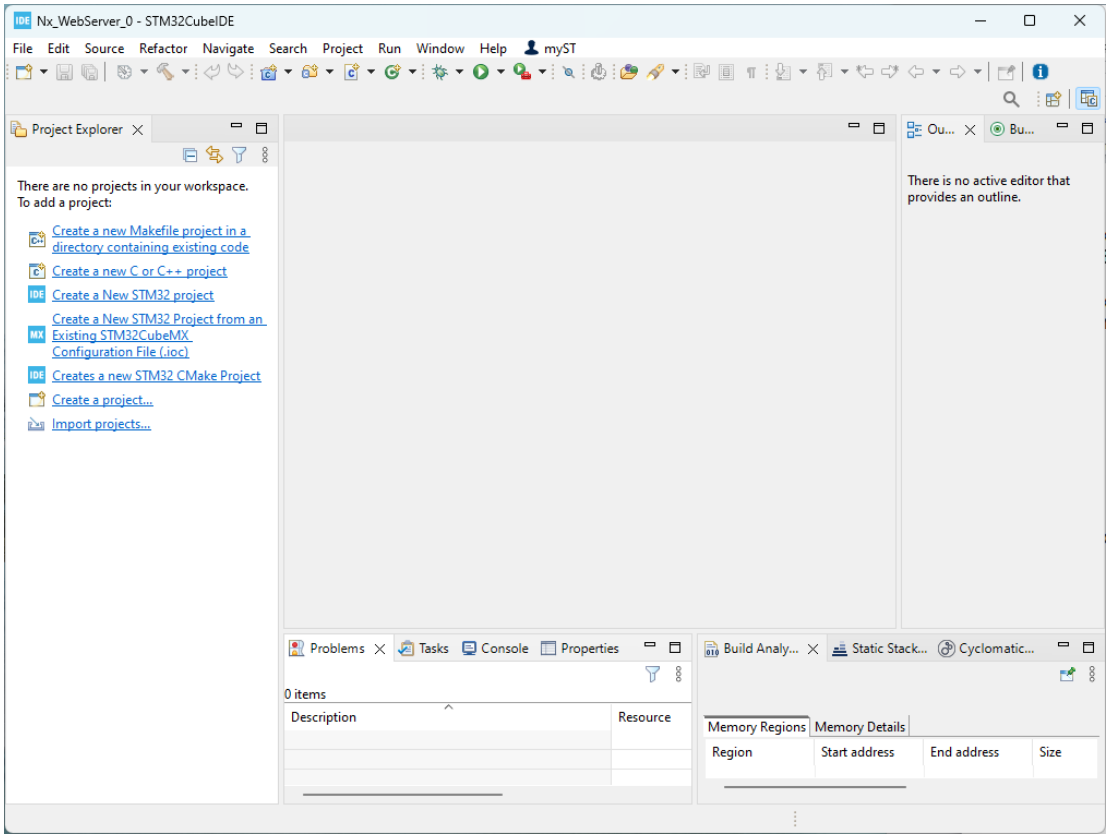
to new directory `Nx_WebServer_0` created at the same level of the directory structure:

```
<install_path>\STM32Cube\Repository\STM32Cube_FW_H5_V1.1.1\Projects\STM32H573I-DK\Applications\NetXDuo\Nx_WebServer_0\
```

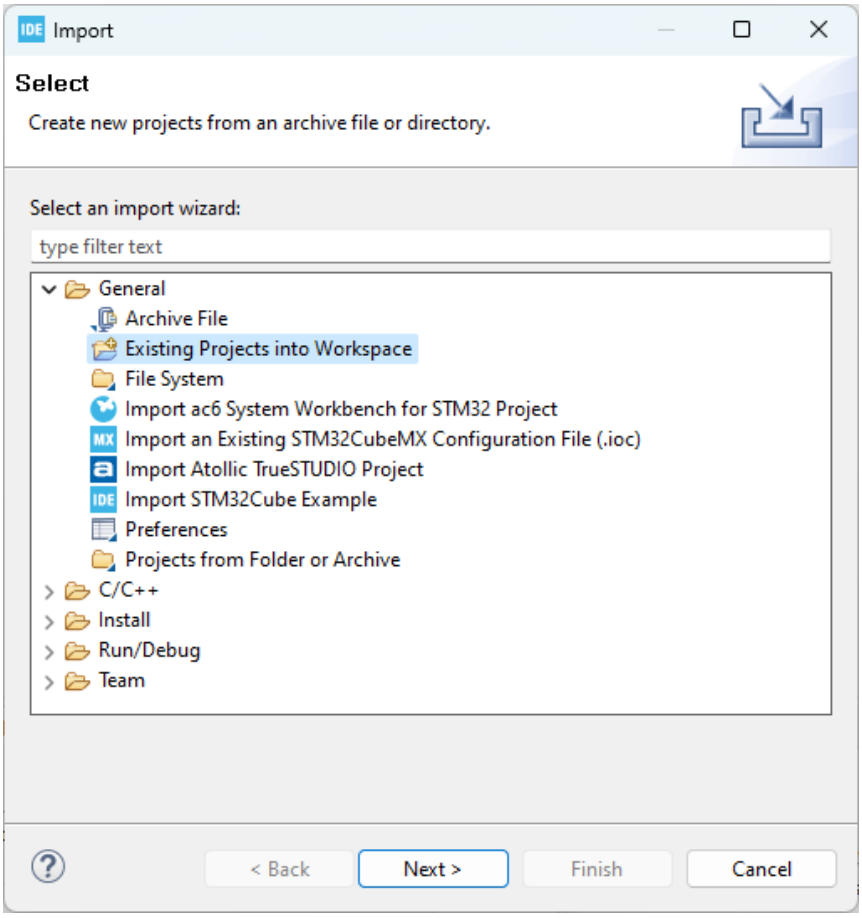
Open STM32CubeIDE tool:



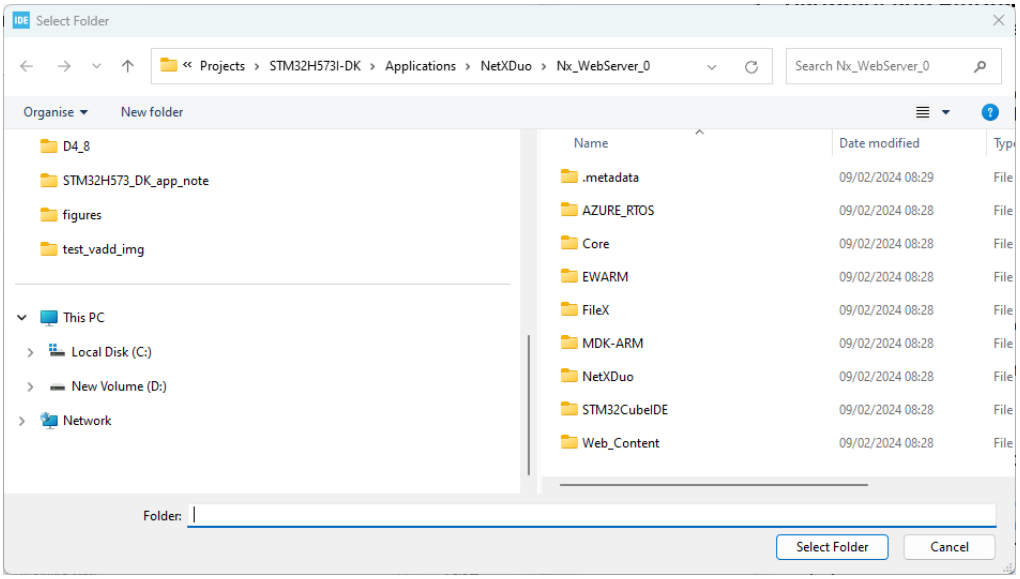
Close velcome page.



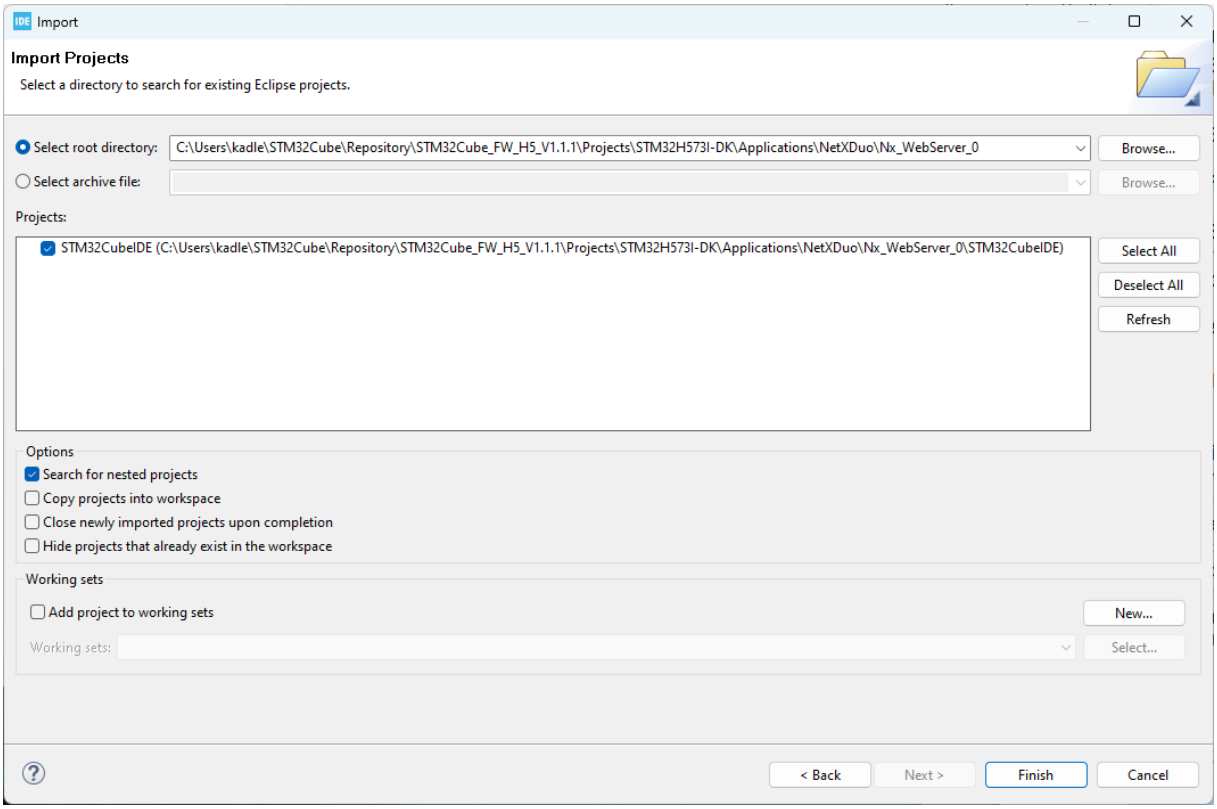
Select: `File` → `Import` → `Existing Projects into Workspace`
Click on `Next`



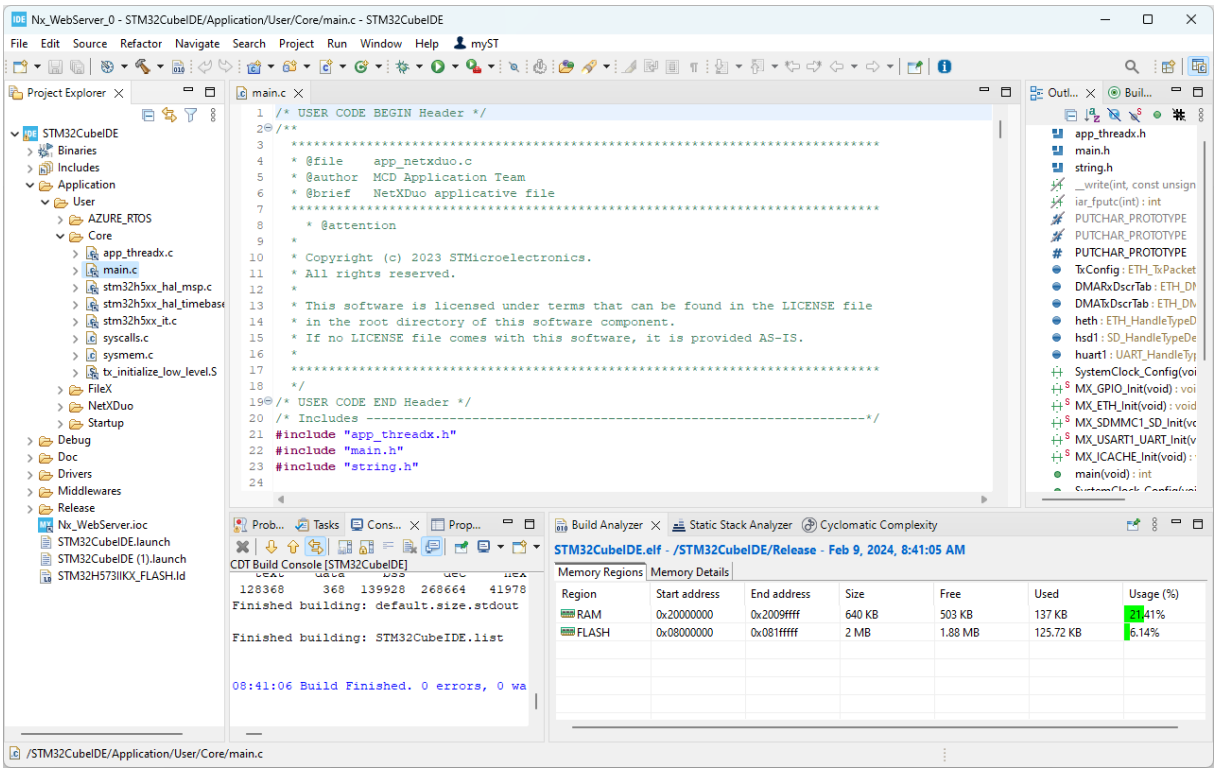
Click on `Select Folder`



Click on `Finish`



Reference Nx_WebServer project is created. Select `release` target. Compile.



On PC, copy web content from directory `Web_Content`

```
<install_path>\STM32Cube\Repository\STM32Cube_FW_H5_V1.1.1\Projects\STM32H573I-DK\Applications\NetXDuo\Nx_WebServer_0\Web_Content\
```

to SD card formated as FAT32. It will be two files and one directory. These files define the www page content.

```
about.html
dashboard.html
[assets]
```

Inset SD card to STM32H573G-DK board.

3.3 Test of Reference Nx_WebServer on STM32H573G-DK board

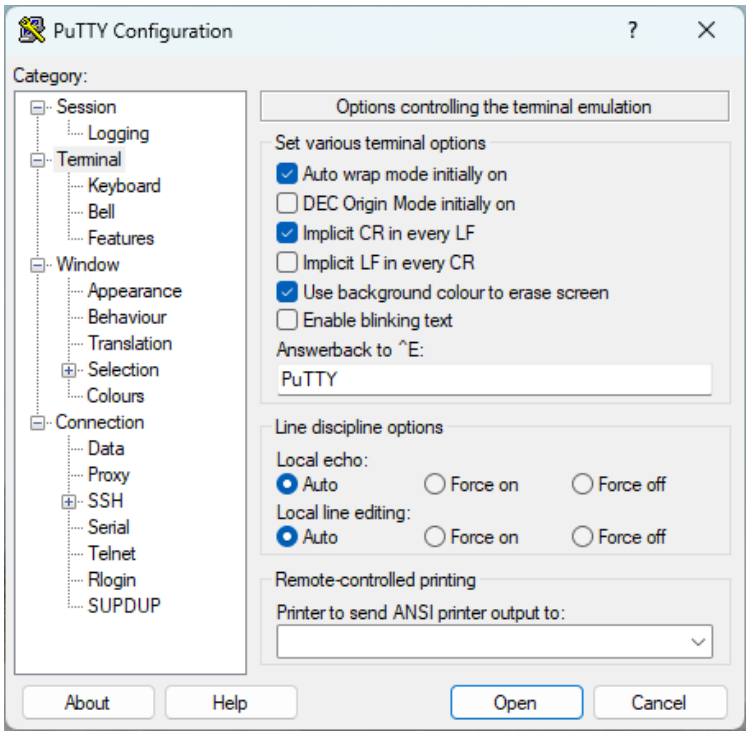
Connect STM32H573G-DK board to PC by USB C cable to CN10 USB STLINK connector.

Nx_WebServer runs on STMicroelectronics STM32H573G-DK board Revision: MB1520-H573I-B02.

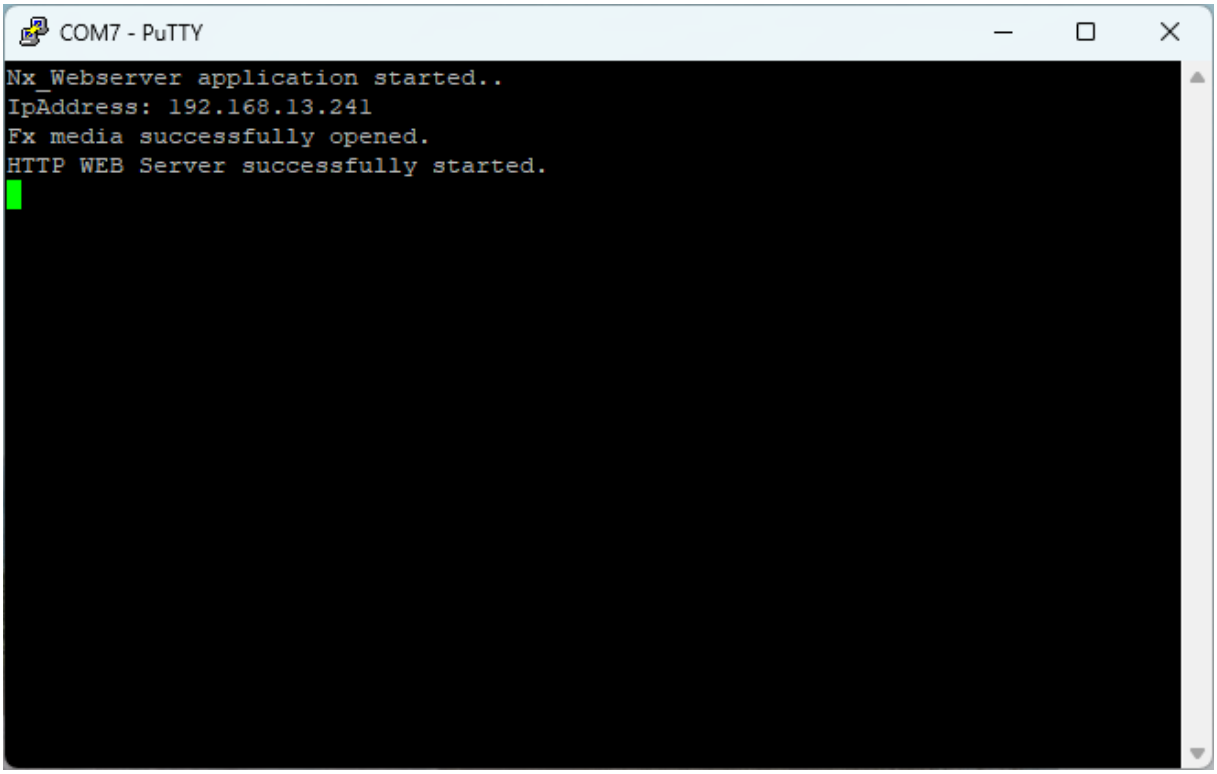
Nx_WebServer uses USART1 to display logs.
The PC `PuTTY` terminal configuration:

- BaudRate = 115200 baud
- Word Length = 8 Bits
- Stop Bit = 1
- Parity = None
- Flow control = None

In case of PuTTY terminal, select option: `Implicit CR in every LF`

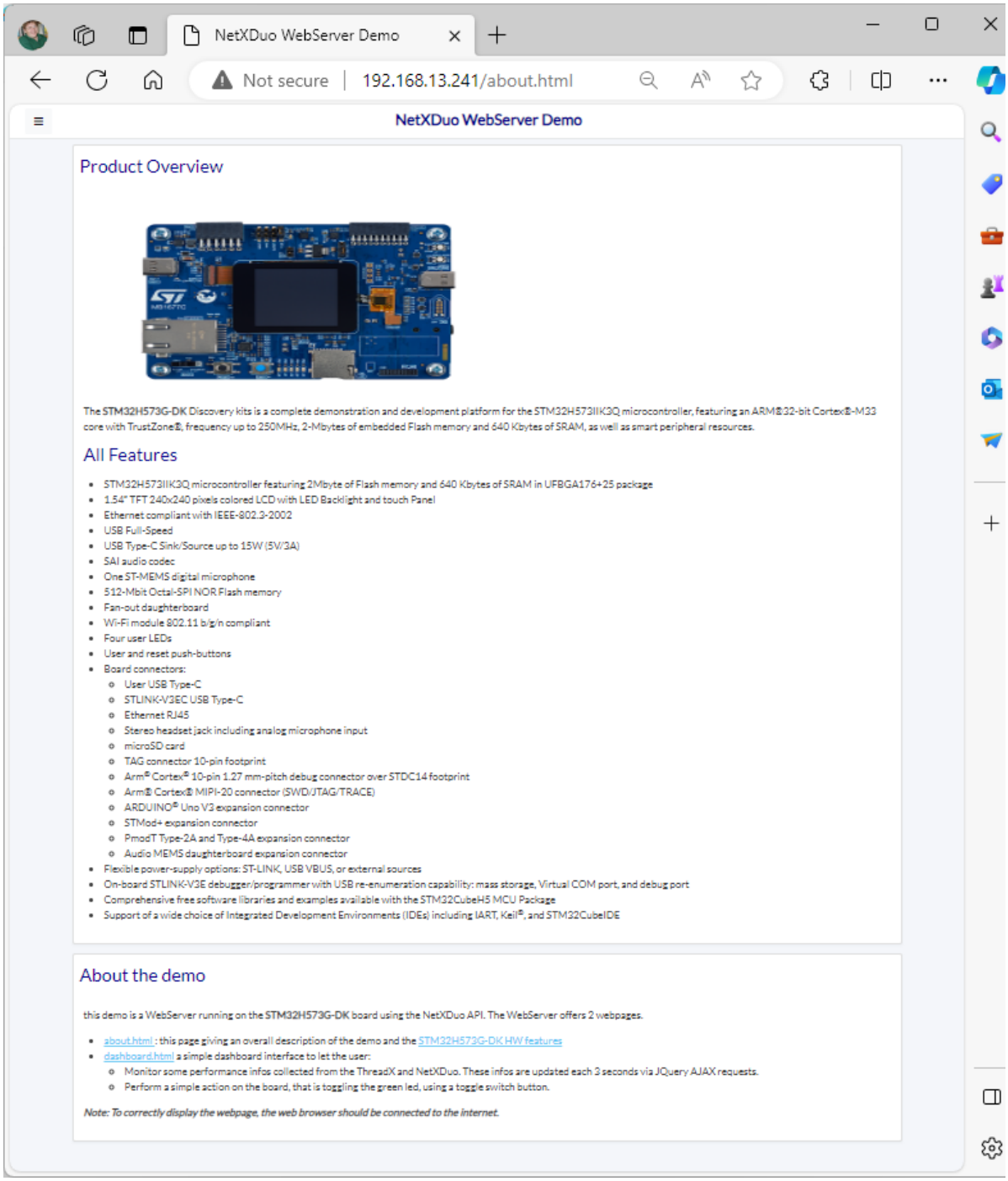


- Open reference Nx_WebServer project in STM32CubeIDE tool.
- Rebuild project for release and load compiled executable into target memory
- Reset STM32H573G-DK board.
- The board IP address “IP@” (assigned by the DNS server of the local network) printed on the PC PuTTY terminal.



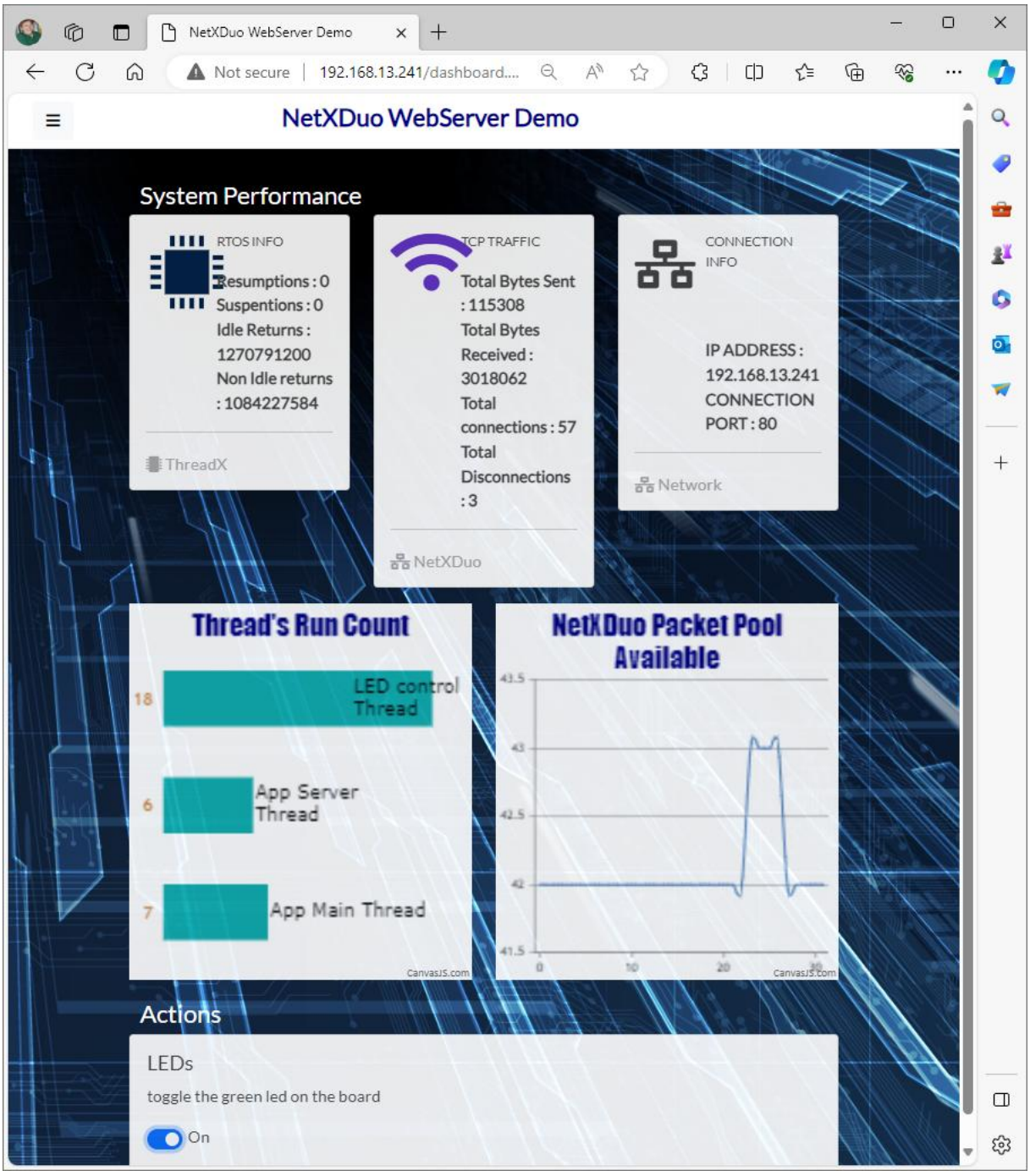
- Home page is displayed on PC browser after entering the url:

```
http://IP@/about.html
```



- Dashboard is displayed on PC browser after entering the url:

```
http://IP@/dashboard.html
```



- On dashboard, you can use mouse to control if Green LED blinking or not.
- On dashboard, you can also watch dynamically updated text information and graphical. information about some `threadx` and `netxduo` middleware paramemeters.

Error status is indicated by Red LED in these cases:

- SD card is not inserted
- www content files are not present on the SD card
- Ethernet cable is not connected
- IP address assignement is not resolved by the DNS server

4 Remote Reset Web Server

The remote reset Web Server application is based on the reference Web HTTP server based application decribed above. It serves for remote reset of the Vitis AI 3.0 inference system [2], [3], [4] or [5].

Copy reference Nx_WebServer project

```
< install_path>\STM32H573_DK_eval_package\sw\Nx_WebServer_3\
```

to new directory `Nx_WebServer_2` created at the same level of the directory structure:

```
<install_path>\STM32Cube\Repository\STM32Cube_FW_H5_V1.1.1\Projects\STM32H573I-DK\Applications\NetXDuo\Nx_WebServer_2\
```

Import project located in directory `Nx_WebServer_2` to STM32CubeIDE tool and compile.

On PC, copy web content from directory Web_Content

```
< install_path>\STM32H573_DK_eval_package\sw\Nx_WebServer_3\Web_Content\
```

to SD card formated as FAT32. It will be two files and one directory.
These files and files in the `assets` directory define the www page content:

```
about.html
dashboard.html
[assets]
```

Inset SD card to STM32H573G-DK board.

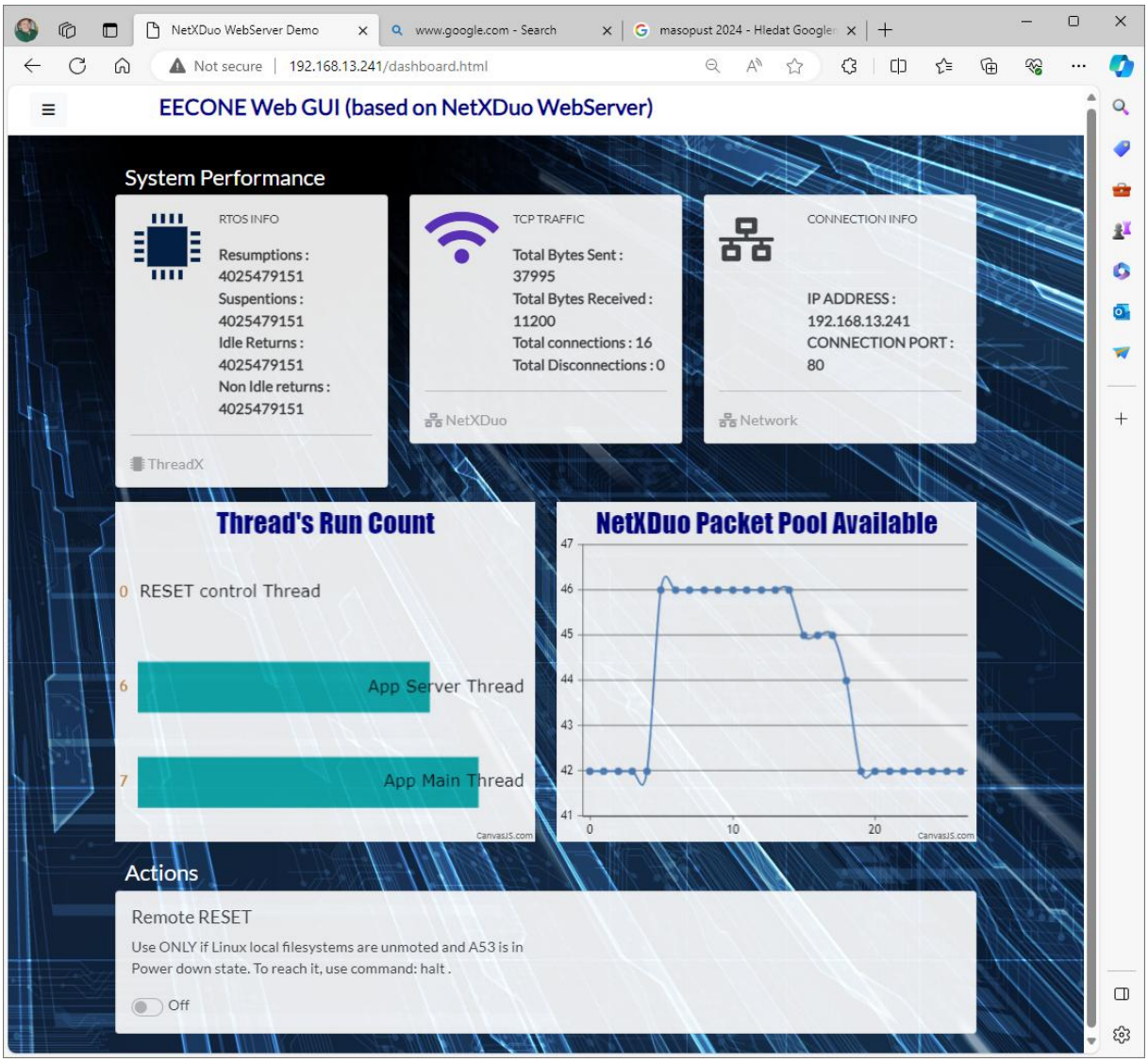
- Open the remote-reset Nx_WebServer project in STM32CubeIDE tool in directory

```
<install_path>\STM32Cube\Repository\STM32Cube_FW_H5_V1.1.1\Projects\STM32H573I-DK\Applications\NetXDuo\Nx_WebServer_2\
```

- Rebuild project for release and load compiled executable into target memory.
- Reset STM32H573G-DK board.
- The board IP address "`IP@`" (assigned by the DNS server of the local network) printed on the PC `PuTTY` terminal.
- On dashboard, you can use mouse to control if Remote RESET is pressed or not.
  If pressed, Green LED is lighting (without blinking).
- On dashboard, you can also watch dynamically updated text information and graphical information about some `threadx` and `netxduo` middleware paramemeters.

Error status is indicated by Red LED in these cases:

- SD card is not inserted
- www content files are not present on the SD card
- Ethernet cable is not connected
- IP address assignement is not resolved by the DNS server

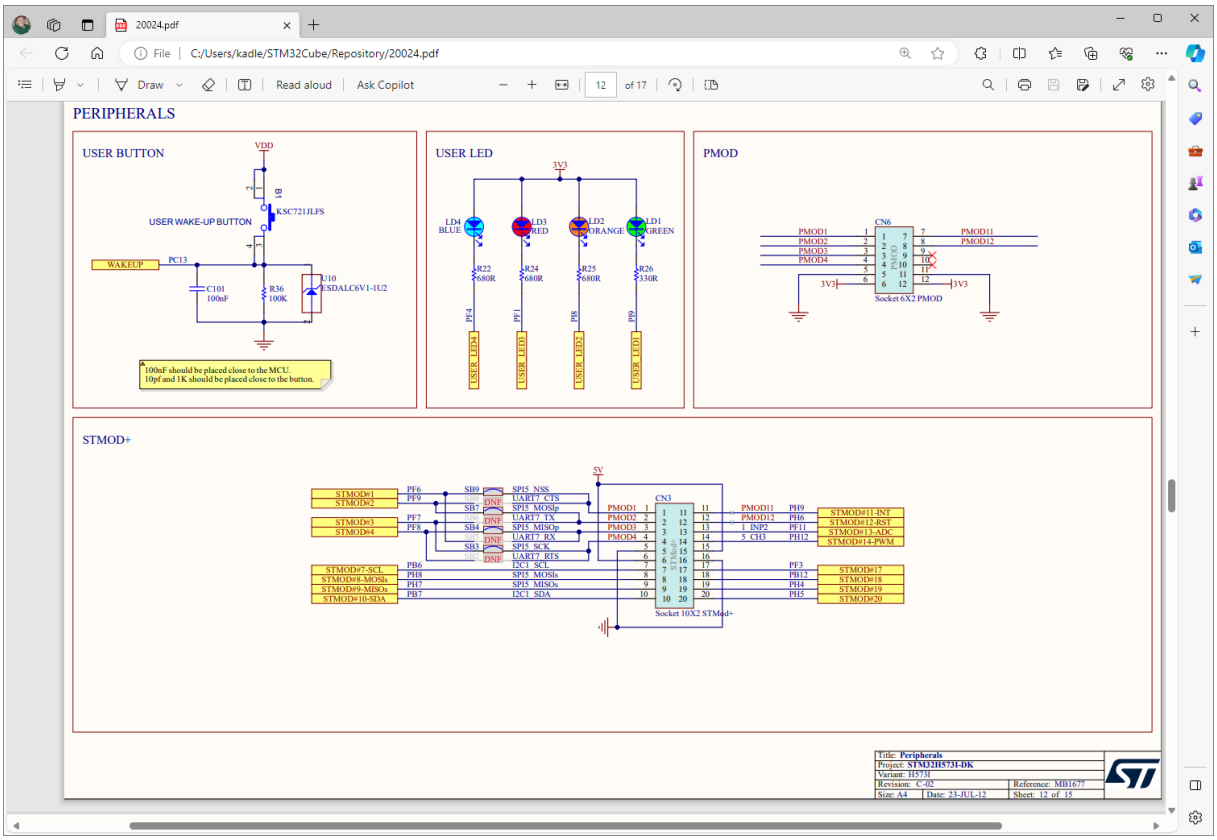


4.1 Modifications of the reference Nx_WebServer

Definition of `RESET_TE_Pin`.

Reset wire is connected to pin 7 of PMOD connector with ground connection pin 11 of the PMOD connector CN6. See shematic sheet 12 of 15 available from:

https://www.st.com/en/evaluation-tools/stm32h573i-dk.html#cad-resources



Reset wire connected to pin 7 of PMOD connector (PMOD11) is driven by pin PH9 of the STM32H5 device.

The `RESET_TE_Pin` is configured to STM32H5 device pin PH9 to control pin 7 of the PMOD connector `CN6`.

File: main.c
Add this additional code, starting from line 386:

```
 /*Configure GPIO pin Output Level */
 HAL_GPIO_WritePin(RESET_TE_GPIO_Port, RESET_TE_Pin, GPIO_PIN_SET);
 /*Configure GPIO pin: RESET_TE_Pin */
 GPIO_InitStruct.Pin = RESET_TE_Pin;
 GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP;
//  GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_OD;
//  GPIO_InitStruct.Pull = GPIO_PULLUP;
 GPIO_InitStruct.Pull = GPIO_NOPULL;
 GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;
 HAL_GPIO_Init(RESET_TE_GPIO_Port, &GPIO_InitStruct);
```

Please notice, that `RESET_TE_Pin` is configured as pin capable to drive the signal to “1” and to “0”. This helps to reduce sensitivity to electronic noice, but it also means, that the reset

button on systems [2], [3], [4] and [5] cannot be manually pressed (connected to zero), if the `RESET_TE_Pin` is set to “1”. This could demage the STM32H5 device.

Alternatively, `RESET_TE_Pin` can be configured as open collector pin with internal pullup resistor by this alternative specification:

```
 /*Configure GPIO pin Output Level */
 HAL_GPIO_WritePin(RESET_TE_GPIO_Port, RESET_TE_Pin, GPIO_PIN_SET);


 /*Configure GPIO pin: RESET_TE_Pin */
 GPIO_InitStruct.Pin = RESET_TE_Pin;
//  GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP;
 GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_OD;
 GPIO_InitStruct.Pull = GPIO_PULLUP;
// GPIO_InitStruct.Pull = GPIO_NOPULL;
 GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;
```

It drives the `RESET_TE_Pin` signal to “1” only by the with internal pullup resistor. This makes it more sensitive to electronic noice, but the reset button on systems [2], [3], [4] and [5] can be manually pressed (connected to zero), even if the `RESET_TE_Pin` is set to “1” by the internal pullup resistor.

Danger of demage for the STM32H5 device is resolved, but the reset wire must be shiled agains the electromagnetic noise in this case.

Change of defines for `RESET_TE_Pin`
File: main.h
Modify text starting from line 60 from:

```
/* Private defines -----------------------------------------------------------*/
#define LED_GREEN_Pin GPIO_PIN_9
#define LED_GREEN_GPIO_Port GPIOI


#define LED_RED_Pin GPIO_PIN_1
#define LED_RED_GPIO_Port GPIOF
```

To:

```
/* Private defines -----------------------------------------------------------*/
#define LED_GREEN_Pin GPIO_PIN_9
#define LED_GREEN_GPIO_Port GPIOI


#define LED_RED_Pin GPIO_PIN_1
```

```
#define LED_RED_GPIO_Port GPIOF


#define RESET_TE_Pin GPIO_PIN_9
#define RESET_TE_GPIO_Port GPIOH
```

`RESET_TE_Pin` is configured to STM32H5 device GPIO pin 9 (`GPIO_PIN_9`) in Bank H (`GPIOH`) to control pin 7 of the PMOD connector `CN6`.

Modification of messages on terminal
File: app_netxduo.c
Change text starting from line 462:

```
else if (strcmp(resource, "/LedOn") == 0)
{
  printf(" Loggling Green Led On \n");
  tx_thread_resume(&LedThread);
}
else if (strcmp(resource, "/LedOff") == 0)
{
  printf(" Loggling Green Led Off \n");
  HAL_GPIO_WritePin(LED_GREEN_GPIO_Port, LED_GREEN_Pin, GPIO_PIN_SET);


  tx_thread_suspend(&LedThread);
}
```

To:

```
else if (strcmp(resource, "/LedOn") == 0)
{
  printf(" Remote A53 RESET and Green Led: On \n");
  tx_thread_resume(&LedThread);
}
else if (strcmp(resource, "/LedOff") == 0)
{
  printf(" Remote A53 RESET and Green Led: Off \n");
  HAL_GPIO_WritePin(LED_GREEN_GPIO_Port, LED_GREEN_Pin, GPIO_PIN_SET);
  HAL_GPIO_WritePin (RESET_TE_GPIO_Port, RESET_TE_Pin, GPIO_PIN_SET);
  tx_thread_suspend(&LedThread);
}
```

Reset signal has to have constant level. No toggling.
Change from function call `HAL_GPIO_TogglePin` to function call to `HAL_GPIO_WritePin`
File: app_netxduo.c
Change function `LedThread_Entry` starting from line 561:

```
void LedThread_Entry(ULONG thread_input)
{
 (void) thread_input;
 /* Infinite loop */
 while (1)
 {
  HAL_GPIO_TogglePin(LED_GREEN_GPIO_Port, LED_GREEN_Pin);
  /* Delay for 500ms (App_Delay is used to avoid context change). */
  tx_thread_sleep(50);
 }
}
```

To:

```
void LedThread_Entry(ULONG thread_input)
{
 (void) thread_input;
 /* Infinite loop */
 while (1)
 {
        HAL_GPIO_WritePin(LED_GREEN_GPIO_Port, LED_GREEN_Pin, GPIO_PIN_RESET);
        HAL_GPIO_WritePin(RESET_TE_GPIO_Port, RESET_TE_Pin, GPIO_PIN_RESET);

  /* Delay for 500ms (App_Delay is used to avoid context change). */
  tx_thread_sleep(50);
 }
}
```

Finally, increase maximum number of times the DNS Client will query the current DNS server. The default value 3 is changed to safer value 10 to support also slow Ethernet networks.

File: nx_user.h
Change from line 1470:

```
/* The maximum number of times the DNS Client will query the current DNS server
```

```
  before trying another server or aborting the DNS query.
  The default value is 3. */
/*
#define NX_DNS_MAX_RETRIES                  3
*/
```

To:

```
/* The maximum number of times the DNS Client will query the current DNS server
  before trying another server or aborting the DNS query.
  The default value is 3. */
#define NX_DNS_MAX_RETRIES                  10
```

4.2 Modifications of www page definition files

Changes made to www page definition files are minimal.

Changes demonstrate simplicity of modifications in text editor. Modified www page definition files are stored on the SD card with FAT32 file system.

5 References

[1]
Jiří Kadlec, Zdeněk Pohl, Lukáš Kohout: Support for STM32H573I-DK web server. (Application note, with evaluation package, UTIA). Published for public access from: https://zs.utia.cas.cz/index.php?ids=results&id=1_STM32H573_DK
This application and evaluation package will be based on the STM32CubeH5 Firmware Examples for STM32H5xx Series Application based on NetXDuo: **Nx_WebServer.**
This STM application provides an example of Azure RTOS NetX Duo stack usage on STM32H573G-DK board, it shows how to develop Web HTTP server based application.
https://htmlpreview.github.io/?https://raw.githubusercontent.com/STMicroelectronics/STM32CubeH5/master/Projects/STM32CubeProjectsList.html

[2]
Lukáš Kohout, Jiří Kadlec, Zdeněk Pohl: Support for TE0802-02-1BEV2-A board with Vitis AI 3.0 DPU and VGA display (Application note with evaluation package, UTIA). Published for public free access from:
https://zs.utia.cas.cz/index.php?ids=results&id=2_TE0802-02-1BEV2-A_AI_3_0_VGA

[3]
Lukáš Kohout, Jiří Kadlec, Zdeněk Pohl: Support for TE0802-02-2AEV2-A board with Vitis AI 3.0 DPU and VGA display (Application note, with evaluation package, UTIA). Published for public access from:
https://zs.utia.cas.cz/index.php?ids=results&id=3_TE0802-02-2AEV2-A_AI_3_0_VGA

[4]
Jiří Kadlec, Zdeněk Pohl, Lukáš Kohout: Support for module-based systems with TE0821 modules on TE0701 carrier board with Vitis AI 3.0 DPU (Application note, with evaluation package, UTIA). Published for free public access from:
https://zs.utia.cas.cz/index.php?ids=results&id=4_TE0821_AI_3_0

[5]
Jiří Kadlec, Zdeněk Pohl, Lukáš Kohout: Support for module-based systems with TE0820 modules on TE0701 carrier board with Vitis AI 3.0 DPU (Application note, with evaluation package, UTIA). Published for free public access from:
https://zs.utia.cas.cz/index.php?ids=results&id=5_TE0820_AI_3_0

[6]
Jiří Kadlec, Zdeněk Pohl, Lukáš Kohout, Raissa Likhonina: Description of compilation of Vitis AI 3.0 models for different configurations of AMD DPUs, (Application note, with evaluation package, UTIA). Published for free public access from:
https://zs.utia.cas.cz/index.php?ids=results&id=6_TE_AI_3_0