

# PIONIERKRAFT



STM32 H7 43 ZI  
Ethernet Implementierung

Bewerbung Ing. Marinerro

# Ethernet Implementierung

- Allgemeines:
  - Zu verwendende Entwicklungsumgebung: STM32CubeIDE
  - Implementierung in neuem STM32-Projekt
  - Implementierung **ohne** Operation System (RTOS, o.ä.), falls möglich
- Aufgabenstellung:
  - Implementierung von Ethernet auf einem STM32H743ZIT6 Microcontroller
    - Entweder TCP oder UDP. Please implement the safer one.
  - Senden von Daten (25 Werte (float 32 bit)) zu einer Adresse
    - Werte vor dem Senden in JSON-Format umwandeln (besser für Backend, Web)
  - Empfangen von Daten (25 Werte (float 32 bit)) von einer Adresse
    - Werte wieder in float zurück wandeln
- Zusätzliche Fragestellungen:
  - Unterschied TCP und UDP erklären
  - Aufbau JSON-Nachricht
  - Wie können die im JSON-Format gesendeten Daten in einem Backend abgelegt werden (Schrittfolge aufzeigen, kein Code)?
- Präsentation (in Powerpoint):
  - Vorstellung der Aufgabenstellung in STM32CubeIDE
  - Vorstellung der zusätzlichen Fragestellungen

# STM32 H7 43 ZI T6

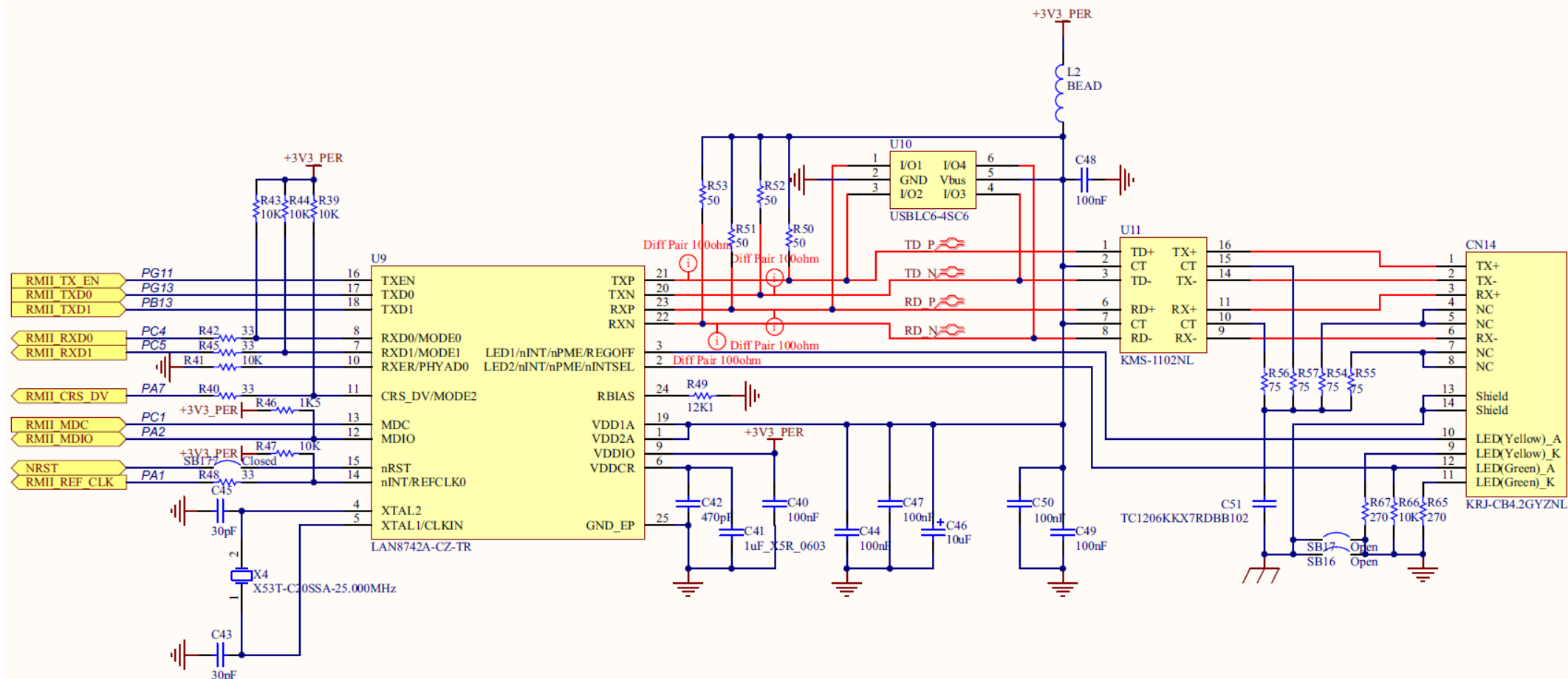
| Order code     | Board reference | User manual | Target STM32   | Differentiating features   |
|----------------|-----------------|-------------|----------------|--|
| NUCLEO-F767ZI  | MB1137          | UM1974      | STM32F767ZIT6U | <ul style="list-style-type: none"><li>• Arm® Mbed Enabled™</li><li>• Ethernet</li><li>• On-board USB OTG</li><li>• USB OTG FS on Micro-AB connector</li><li>• ST-LINK/V2-1</li></ul> |
| NUCLEO-H743ZI  |                 |             | STM32H743ZIT6U | <ul style="list-style-type: none"><li>• Arm® Mbed Enabled™</li><li>• Ethernet</li><li>• USB OTG FS on Micro-AB connector</li><li>• ST-LINK/V2-1</li></ul>                            |
| NUCLEO-H743ZI2 | MB1364          | UM2407      | STM32H743ZIT6U | <ul style="list-style-type: none"><li>• Ethernet</li><li>• USB OTG FS on Micro-AB connector</li><li>• STLINK-V3E</li></ul>   |
| NUCLEO-H753ZI  |                 |             | STM32H753ZIT6U | <ul style="list-style-type: none"><li>• Ethernet</li><li>• USB OTG FS on Micro-AB connector</li><li>• STLINK-V3E</li><li>• Cryptography</li></ul>                                    |

[https://www.st.com/resource/en/data\\_brief/nucleo-h743zi.pdf](https://www.st.com/resource/en/data_brief/nucleo-h743zi.pdf)

# STM32 H7 43 ZI T6

Board MB1137

Figure 19. Ethernet PHY with RJ45 connector



# STM32 H7 43 ZI

New Project from a MCU/MPU

MCU/MPU SelectorBoard SelectorCross Selector

MCU/MPU Filters

★

Part Number Search

STM32H743ZI

Core

Series

Line

Package

Other

Price = 8.289

IO = 114

Eeprom = 0 (Bytes)

Flash = 2048 (kBytes)

Ram = 1024 (kBytes)

Freq. = 400 (MHz)

FeaturesBlock DiagramDocs & ResourcesDatasheetBuyStart Project

★ STM32H743ZI

High-performance and DSP with DP-FPU, Arm Cortex-M7 MCU with 2MBytes of Flash memory, 1MB RAM, 480 MHz CPU, Art Accelerator, L1 cache, external memory interface, large set of peripherals

ACTIVE

Active  
Product is in mass production

Unit Price for 10kU (US\$) : **8.289**

Boards: [NUCLEO-H743ZI](#) - [NUCLEO-H743ZI2](#)

LQFP144

STM32H742x/G and STM32H743x/G devices are based on the high-performance Arm® Cortex®-M7 32-bit RISC core operating at up to 480 MHz. The Cortex®-M7 core features a floating point unit (FPU) which supports Arm® double-precision (IEEE 754 compliant) and single-precision data-processing instructions and data types. STM32H742x/G and STM32H743x/G devices support a full set of DSP

MCUs/MPUs List: 1 item

+ Display similar items

Export

| ★ | Part No     | Ref... | Marketing St... | Unit Price for 10kU... | Board                         | Package                        | Flash   | RAM        | IO         | Freq. | GFX Sc...    |
|---|-------------|--------|-----------------|------------------------|-------------------------------|--------------------------------|---------|------------|------------|-------|--------------|
| ★ | STM32H743ZI | ST...  | Active          | 8.289                  | <a href="#">NUCLEO-H743ZI</a> | <a href="#">NUCLEO-H743ZI2</a> | LQFP144 | 2048 kB... | 1024 kB... | 114   | 400 M... 0.0 |

# Ethernet – main bibliography



## **RM0433 Reference manual**

---

STM32H742, STM32H743/753 and STM32H750 Value line  
advanced Arm<sup>®</sup>-based 32-bit MCUs

---

[https://www.st.com/content/ccc/resource/technical/document/reference\\_manual/group0/c9/a3/76/fa/55/46/45/fa/DM00314099/files/DM00314099.pdf/jcr:content/translations/en.DM00314099.pdf](https://www.st.com/content/ccc/resource/technical/document/reference_manual/group0/c9/a3/76/fa/55/46/45/fa/DM00314099/files/DM00314099.pdf/jcr:content/translations/en.DM00314099.pdf)

**Chapter 57:**      **Ethernet (ETH): media access control (MAC) with DMA controller**

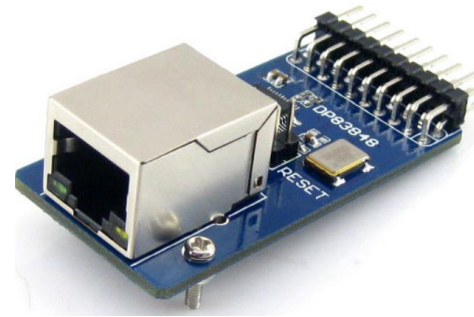
# Ethernet vocabulary

ETH: Ethernet

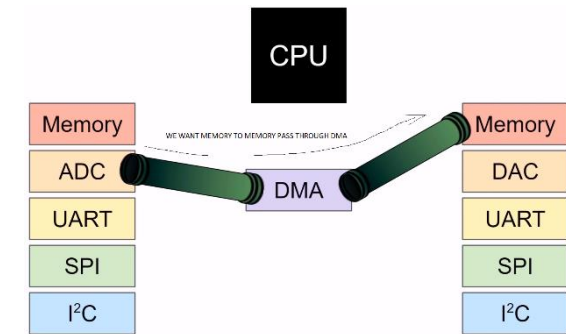
IP: volatile address

MAC: fixed address

PHY: physical device port



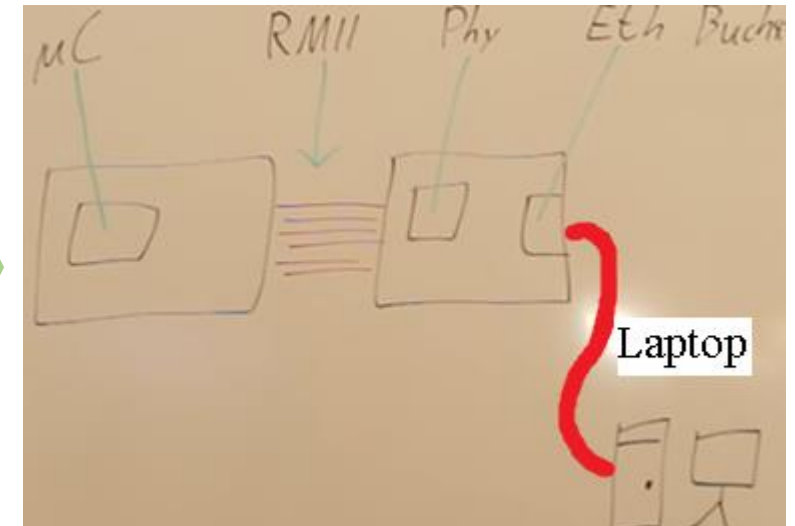
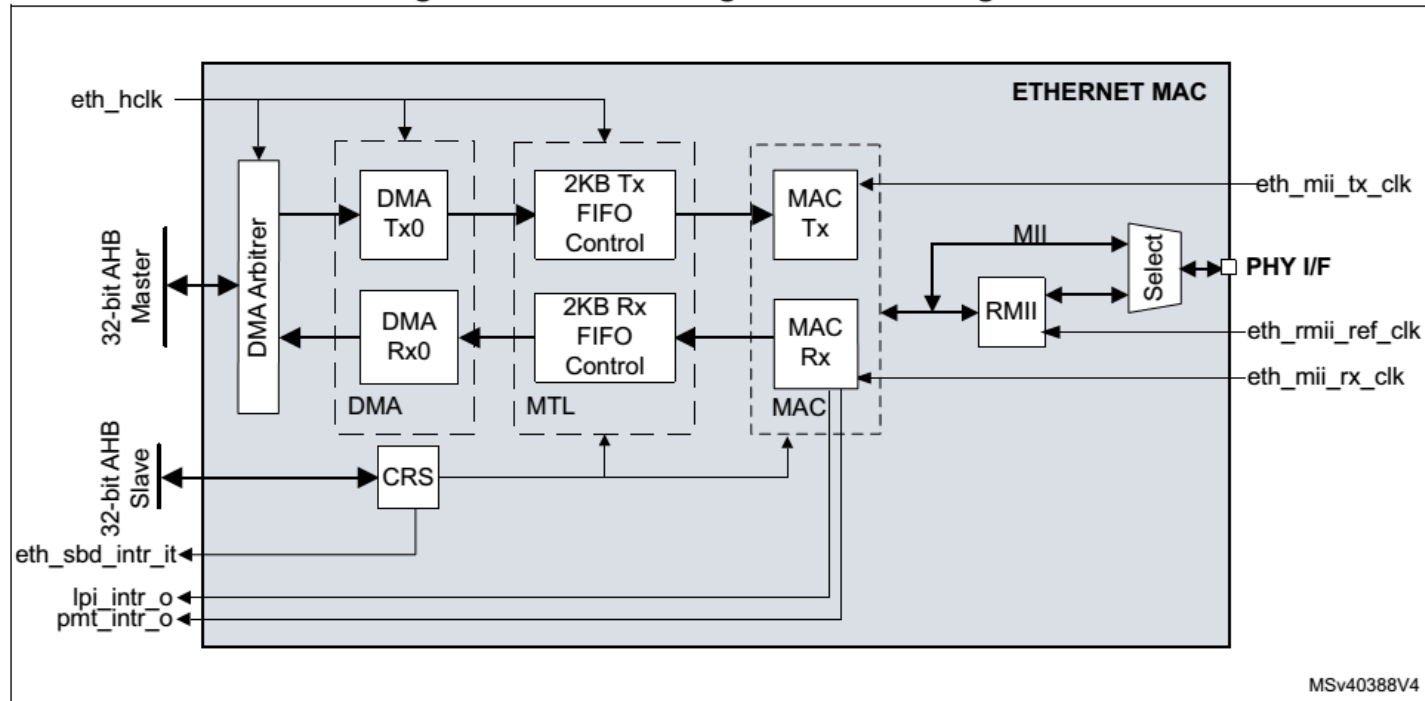
PHY





# Ethernet

Figure 776. Ethernet high-level block diagram



MAC - media access control modul: to implement Eth protocol

MTL - mac transaction layer: to control data flow between application and MAC

RMII: Reduced media independent interface

DMA: direct memory access interface, to communicate without bothering the CPU



# Ethernet pins

## 57.3 Ethernet pins and internal signals

Table 506 lists the Ethernet inputs and output signals connected to package pins or balls, while Table 507 shows the internal Ethernet signals.

Table 506. Ethernet peripheral pins

| Pin name | Alternate function name (mapped on AF11) |
|----------|--|
| PA0      | ETH_MII_CRS                              |
| PA1      | ETH_MII_RX_CLK/ETH_RMII_REF_CLK          |
| PA2      | ETH_MDIO                                 |
| PA3      | ETH_MII_COL                              |
| PA7      | ETH_MII_RX_DV/ETH_RMII_CRS_DV            |
| PA9      | ETH_TX_ER                                |
| PB0      | ETH_MII_RXD2                             |
| PB1      | ETH_MII_RXD3                             |
| PB2      | ETH_TX_ER                                |
| PB5      | ETH_PPS_OUT                              |
| PB8      | ETH_MII_TXD3                             |
| PB10     | ETH_MII_RX_ER                            |
| PB11     | ETH_MII_TX_EN/ETH_RMII_TX_EN             |
| PB12     | ETH_MII_TXD0/ETH_RMII_TXD0               |
| PB13     | ETH_MII_TXD1/ETH_RMII_TXD1               |
| PC1      | ETH_MDC                                  |
| PC2      | ETH_MII_TXD2                             |
| PC3      | ETH_MII_TX_CLK                           |
| PC4      | ETH_MII_RXD0/ETH_RMII_RXD0               |
| PC5      | ETH_MII_RXD1/ETH_RMII_RXD1               |
| PE2      | ETH_MII_TXD3                             |
| PG8      | ETH_PPS_OUT                              |
| PG11     | ETH_MII_TX_EN/ETH_RMII_TX_EN             |

Table 506. Ethernet peripheral pins (continued)

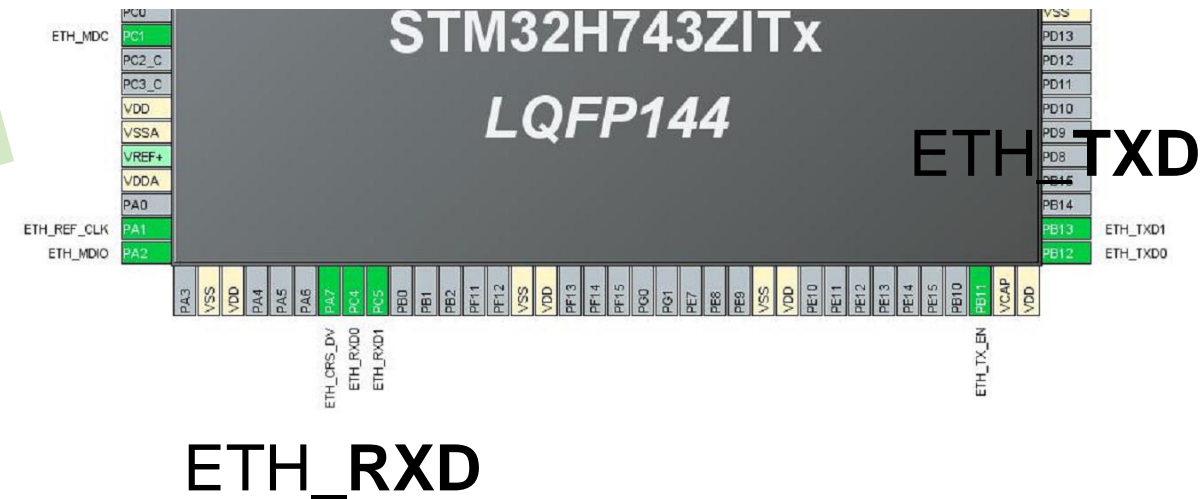
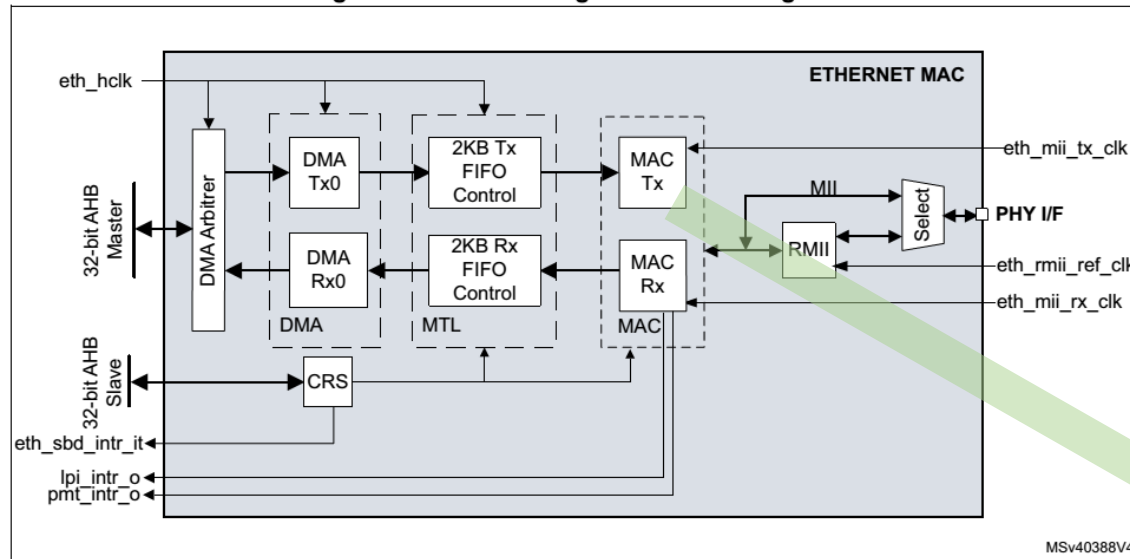
| Pin name | Alternate function name (mapped on AF11) |
|----------|--|
| PG12     | ETH_MII_TXD1/ETH_RMII_TXD1               |
| PG13     | ETH_MII_TXD0/ETH_RMII_TXD0               |
| PG14     | ETH_MII_TXD1/ETH_RMII_TXD1               |
| PH2      | ETH_MII_CRS                              |
| PH3      | ETH_MII_COL                              |
| PH6      | ETH_MII_RXD2                             |
| PH7      | ETH_MII_RXD3                             |
| PI10     | ETH_MII_RX_ER                            |
| PI12     | ETH_TX_ER                                |

Table 507. Ethernet internal input/output signals

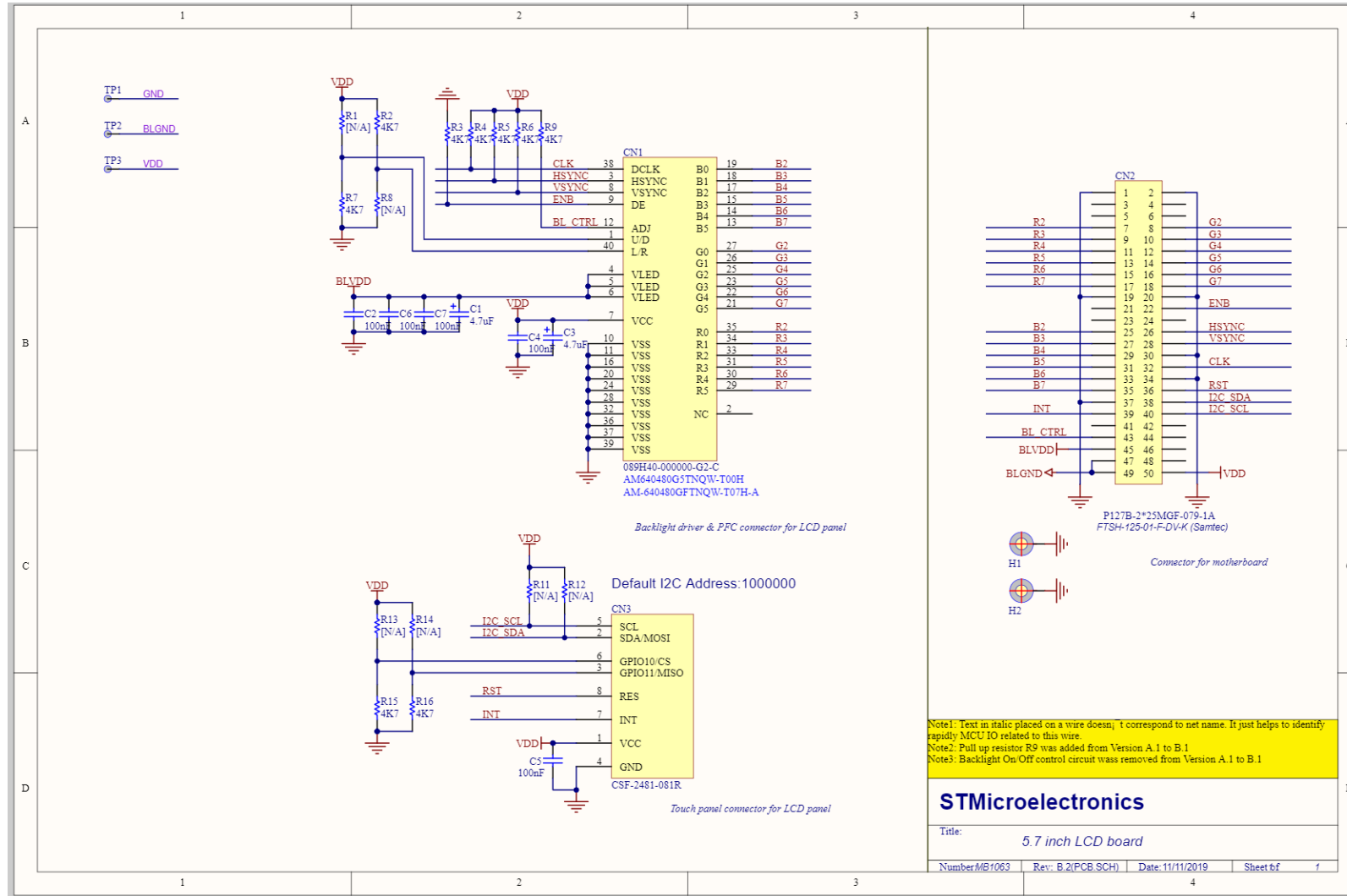
| Signal name      | Signal type    | Description   |
|------------------|----------------|---|
| eth_hclk         | Digital input  | AHB clock   |
| eth_sbd_intr_it  | Digital output | Main Ethernet interrupt   |
| lpi_intr_o       | Digital output | Sideband signal generated when the transmitter or receiver enters or exits the LPI state. |
| pmt_intr_o       | Digital output | Sideband signal generated when a valid remote wakeup packet is received                   |
| eth_mii_tx_clk   | Digital input  | MII Tx kernel clock   |
| eth_mii_rx_clk   | Digital input  | MII Rx kernel clock   |
| eth_rmii_ref_clk | Digital input  | RMII reference kernel clock   |

# Ethernet pins

Figure 776. Ethernet high-level block diagram



# Ethernet pins



# DMA

Figure 777. DMA transmission flow (standard mode)

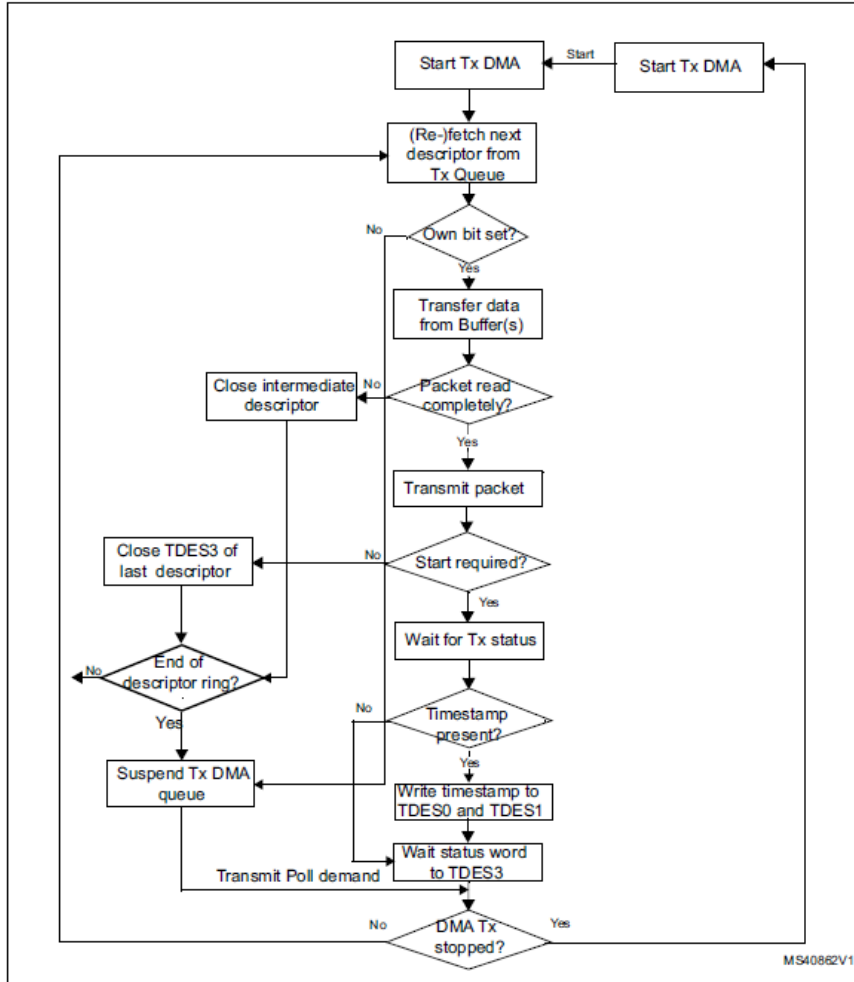
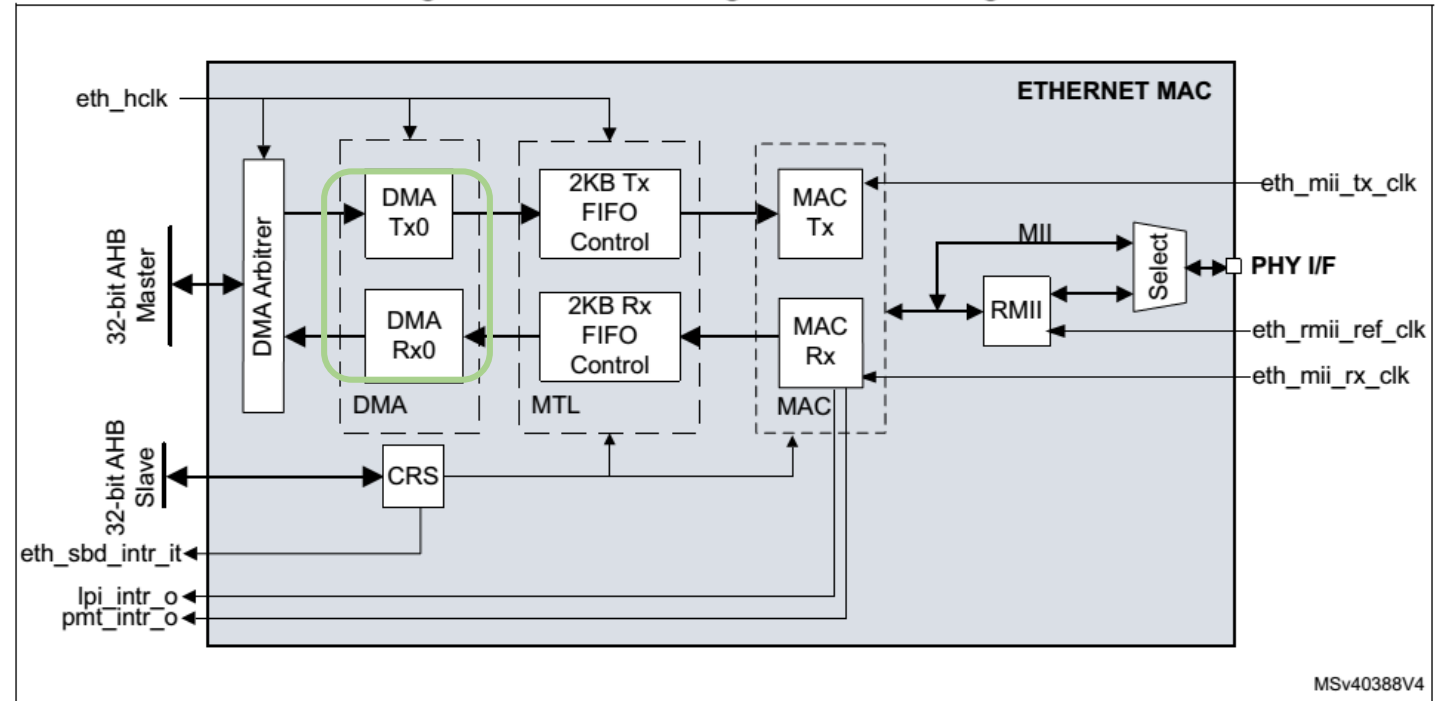


Figure 776. Ethernet high-level block diagram



# DMA

DMA Mode and Configuration

Configuration

☑ DMA1, DMA2    ☑ MemToMem

| DMA Request | Stream        | Direction        | Priority |
|-------------|---------------|------------------|----------|
| MEMTOMEM    | DMA1 Stream 0 | Memory To Memory | Low      |

Add    Delete

DMA Request Settings

Mode:

Use Fifo: ☒    Threshold:

| Src Memory   |  | Dst Memory   |  |
|--|--|--|--|
| Increment Address: <input checked="" type="checkbox"/> |  | Increment Address: <input checked="" type="checkbox"/> |  |
| Data Width: <input type="text" value="Byte"/>          | <input type="text" value="Byte"/>        | Data Width: <input type="text" value="Byte"/>          | <input type="text" value="Byte"/>        |
| Burst Size: <input type="text" value="4 Increment"/>   | <input type="text" value="4 Increment"/> | Burst Size: <input type="text" value="4 Increment"/>   | <input type="text" value="4 Increment"/> |

## FIFO: Threshold & Burst mode 7

- Threshold level triggers the data transfers to/from Memory.
- Each stream has independent configurable threshold levels:
  - Four threshold levels available: 1/4 FIFO Full, 1/2 FIFO Full, 3/4 FIFO Full, and FIFO Full
- Burst mode:
  - Burst mode is available only when FIFO mode is enabled (Direct mode disabled)
  - Available Burst modes:
    - INC4: 1 burst = 4 beats (4 Words, 8 Half-Words or 16 Bytes)
    - INC8: 1 burst = 8 beats (8 Half-Words or 16 Bytes)
    - INC16: 1 burst = 16 beats (16 Bytes)

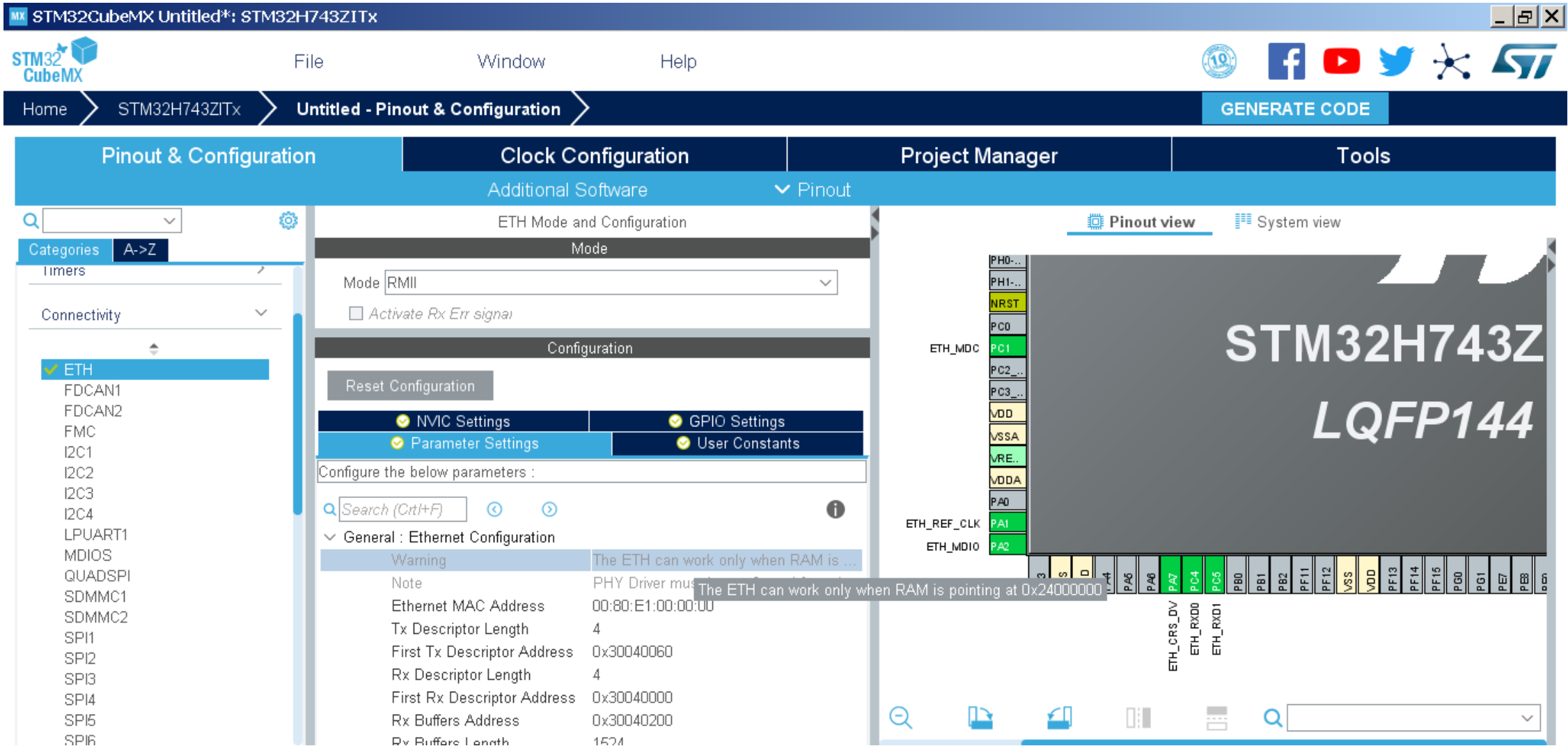
| Memory data size | Burst Size       | Allowed threshold levels |
|------------------|------------------|--------------------------|
| Byte             | 4 beats (INC4)   | 1/4, 1/2, 3/4 and Full   |
|                  | 8 beats (INC8)   | 1/2 & Full               |
|                  | 16 beats (INC16) | Full                     |
| Half-Word        | 4 beats (INC4)   | 1/2 & Full               |
|                  | 8 beats (INC8)   | Full                     |
| Word             | 4 beats (INC4)   | Full                     |



FIFO threshold should be compatible with Burst size

**25 DWORDS = 25\*4bytes**  
(imposed specification)

# Ethernet + RMII



# TCP vs UDP

- UDP: no guarantee you are getting all the packets.
- TCP: guarantees the recipient will receive the packets in order by numbering them.



## TCP is safer → LWIP

Middleware

Configure the below parameters :

Search (Ctrl+F)

General : Ethernet Configuration

Warning The ETH can work only when RAM is ...

LWIP: **TCP**

Lightweight TCP/IP stack

Status:

Not available:

Active only if: ETH IP configured / FREERTOS is enabled when MBEDTLS is enabled.

For STM32H7, CORTEX\_M7 CPU DCache parameter must be Enabled (Configuration tab) to allow LwIP activation.

ETH\_REF\_CLK

ETH\_MDIO

System Core

BDMA

CORTEX\_M7

DMA

GPIO

IWDG1

MDMA

NVIC

RCC

✓ SYS

WWDG1

Reset Configuration

Parameter Settings

User Constants

Configure the below parameters :

Search (Ctrl+F)

Cortex Interface Settings

CPU ICache Disabled

CPU DCache Disabled

Cortex Memory Protection Unit Co... Enabled

MPU Control Mode Disabled

Enabled

Enable LWIP



# TCP vs UDP

In LwIP General Settings...

LWIP Mode and Configuration

Configuration

Reset Configuration

|            |                  |                |                   |
|------------|------------------|----------------|-------------------|
| Checksum   | Debug            | User Constants | Platform Settings |
| SNMP       | SNTP             | MDNS/TFTP      | Perf/Checks       |
| Statistics | General Settings | Key Options    | PPP               |
| IPv6       | HTTPD            |                |                   |

Configure the below parameters :

Search (Ctrl+F) ⏪ ⏩ i

- > LwIP Version
- > IPv4 - DHCP Options
- > IP Address Settings
- > RTOS Dependency
- > Platform Settings
- ▼ Protocols Options
  - LWIP\_ICMP (ICMP Module Activation) Enabled
  - LWIP\_IGMP (IGMP Module) Disabled
  - LWIP\_DNS (DNS Module) Disabled
  - LWIP\_UDP (UDP Module) Disabled
  - MEMP\_NUM\_UDP\_PCB (Number of UDP Connections) 4
  - LWIP\_TCP (TCP Module) Enabled
  - MEMP\_NUM\_TCP\_PCB (Number of TCP Connections) 5

Disable: UDP



Enable: TCP



# PHY configure

Mode RMII

☐ Activate Rx Err signal

Configuration

Reset Configuration

☒ Parameter Settings ☒ User Constants ☒ NVIC Settings ☒ GPIO Settings

Configure the below parameters :

General : Ethernet Configuration

Warning

Note

Ethernet MAC Address 00:80:E1:00:00:00

Tx Descriptor Length 4

The ETH can work only when PAM is point...

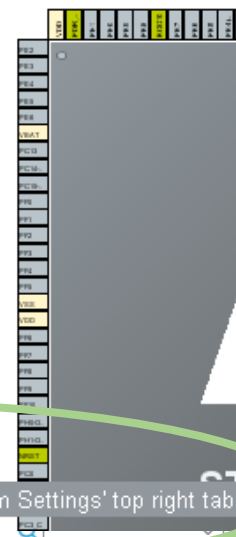
PHY Driver must be configured from the LwIP...

PHY Driver must be configured from the LwIP 'Platform Settings' top right tab

## Board MB1137

| U9 |              |                        | D11 P2 |  |
|----|--------------|------------------------|--------|--|
| 16 | TXEN         | TXP                    | 21     |  |
| 17 | TXD0         | TXN                    | 20     |  |
| 18 | TXD1         | RXP                    | 23     |  |
|    |              | RXN                    | 22     |  |
| 8  | RXD0/MODE0   |                        | 3      |  |
| 7  | RXD1/MODE1   | LED1/nINT/nPME/REGOFF  | 2      |  |
| 10 | RXER/PHYAD0  | LED2/nINT/nPME/nINTSEL | 2      |  |
| 11 | CRS_DV/MODE2 | RBIAS                  | 24     |  |
| 13 | MDC          | VDD1A                  | 19     |  |
| 12 | MDIO         | VDD2A                  | 1      |  |
|    |              | VDDIO                  | 9      |  |
| 15 | nRST         | VDDCR                  | 6      |  |
| 14 | nINT/REFCLK0 |                        |        |  |
| 4  | XTAL2        |                        |        |  |
| 5  | XTAL1/CLKIN  | GND_EP                 | 25     |  |

LAN8742A-CZ-TR



Categories A-Z

USB\_OTG\_FS

USB\_OTG\_HS

Multimedia >

Security >

Computing >

Middleware >

FATFS

FREERTOS

LIBJPEG

☒ LWIP

LWIP Mode and Configuration

Mode

☒ Enabled

Configuration

Reset Configuration

☒ Checksum ☒ Debug ☒ User Constants ☒ Platform Settings

☒ SNMP ☒ SNTP ☒ MDNS/TFTP ☒ Perf/Checks ☒ Statistics

☒ General Settings ☒ Key Options ☒ PPP ☒ IPv6 ☒ HTTPD

Platform proposal

BSP

| Name       | IPs or Components | Found Solutions | BSP API              |
|------------|-------------------|-----------------|----------------------|
| Driver_PHY | LAN8742           | LAN8742         | BSP_COMPONENT_DRIVER |

# IP

DCHP disabled → insert IP address

LWIP Mode and Configuration

Mode

☒ Enabled

Configuration

Reset Configuration

|  |  |  |   |
|--|--|--|---|
| <input checked="" type="checkbox"/> Checksum   | <input checked="" type="checkbox"/> Debug            | <input checked="" type="checkbox"/> User Constants | <input checked="" type="checkbox"/> Platform Settings |
| <input checked="" type="checkbox"/> SNMP       | <input checked="" type="checkbox"/> SNTP             | <input checked="" type="checkbox"/> MDNS/TFTP      | <input checked="" type="checkbox"/> Perf/Checks       |
| <input checked="" type="checkbox"/> Statistics | <input checked="" type="checkbox"/> General Settings | <input checked="" type="checkbox"/> Key Options    | <input checked="" type="checkbox"/> PPP               |
| <input checked="" type="checkbox"/> IPv6       | <input checked="" type="checkbox"/> HTTPD            |  |   |

Configure the below parameters :

Search (Ctrl+F) ⏪ ⏩ ⓘ

▼ LwIP Version  
LwIP Version (Version of LwIP su... 2.0.3)

▼ IPv4 - DHCP Options  
LWIP\_DHCP (DHCP Module) Disabled

▼ IP Address Settings

|                               |                 |
|-------------------------------|-----------------|
| * IP_ADDRESS (IP Address)     | 000.000.000.000 |
| * NETMASK_ADDRESS (Netmask... | 000.000.000.000 |
| * GATEWAY_ADDRESS (Gateway... | 000.000.000.000 |

customize !!

# Real-time optimization (RTO)

LWIP Mode and Configuration

Mode

☒ Enabled

Configuration

Reset Configuration

|  |   |  |   |  |
|--|---|--|---|--|
| <input checked="" type="checkbox"/> Checksum         | <input checked="" type="checkbox"/> Debug       | <input checked="" type="checkbox"/> User Constants | <input checked="" type="checkbox"/> Platform Settings |  |
| <input checked="" type="checkbox"/> SNMP             | <input checked="" type="checkbox"/> SNTP        | <input checked="" type="checkbox"/> MDNS/TFTP      | <input checked="" type="checkbox"/> Perf/Checks       | <input checked="" type="checkbox"/> Statistics |
| <input checked="" type="checkbox"/> General Settings | <input checked="" type="checkbox"/> Key Options | <input checked="" type="checkbox"/> PPP            | <input checked="" type="checkbox"/> IPv6              | <input checked="" type="checkbox"/> HTTPD      |

Configure the below parameters :

- > LwIP Version
- > IPv4 - DHCP Options
- > IP Address Settings
- ▼ RTOS Dependency
  - WITH\_RTOS (Use FREERTOS \*\*... Disabled

**Disabled (imposed specification)**

# C code generator → STM32CubeIDE

Follow STM32CubeMX “User Manual” (tutorial 1):

## Tutorial 1: From pinout to project C code generation using an MCU of the STM32F4 Series

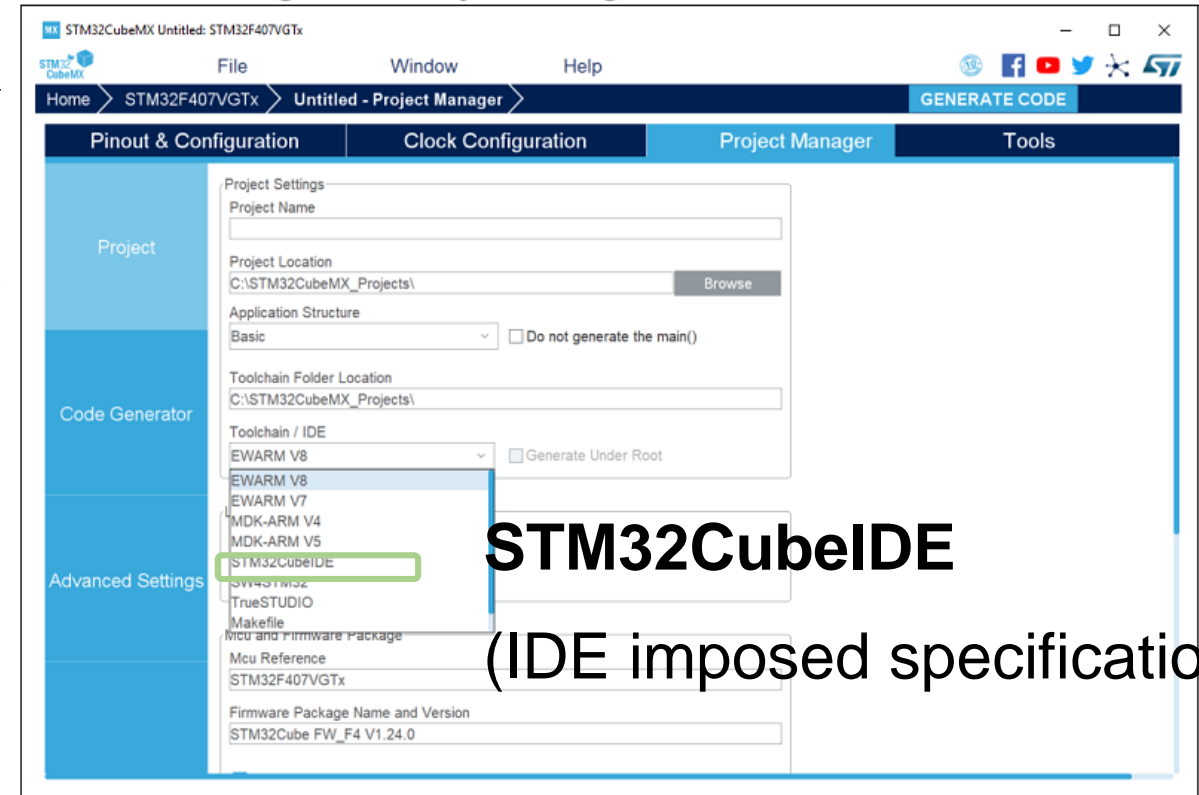
### 10.7 Generating a complete C project

#### 10.7.1 Setting project options

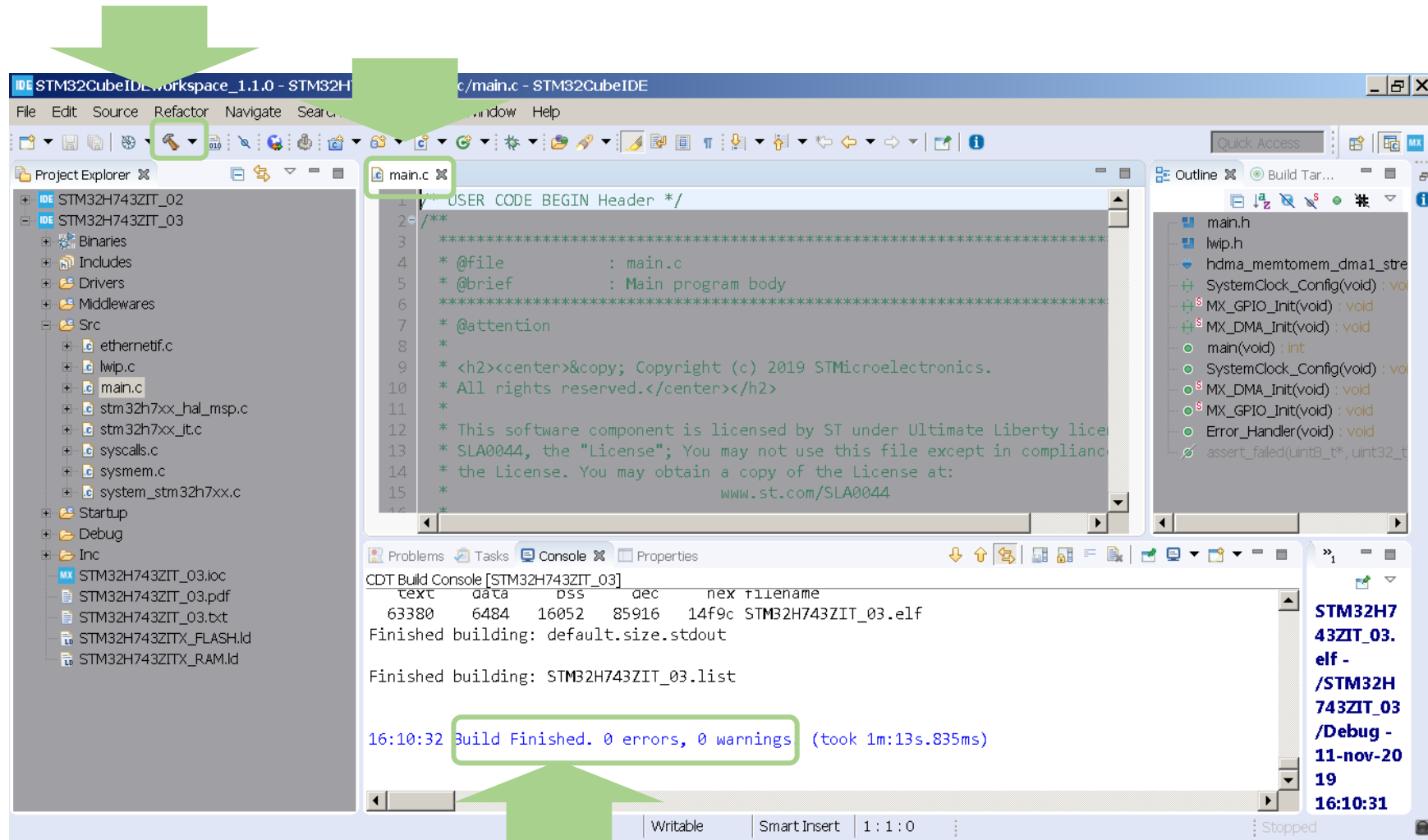
Default project settings can be adjusted prior to C code generation as shown in [Figure 216](#).

1. Select the **Project Manager** view to update project settings and generation options.
2. Select the **Project Tab** and choose a Project **name**, **location** and a **toolchain** to generate the project (see [Figure 216](#)).
3. Select the **Code Generator** tab to choose various C code generation options:
  - The library files copied to *Projects* folder.
  - C code regeneration (e.g. what is kept or backed up during C code regeneration).
  - HAL specific action (e.g. set all free pins as analog I/Os to reduce MCU power consumption).

Figure 216. Project Settings and toolchain selection

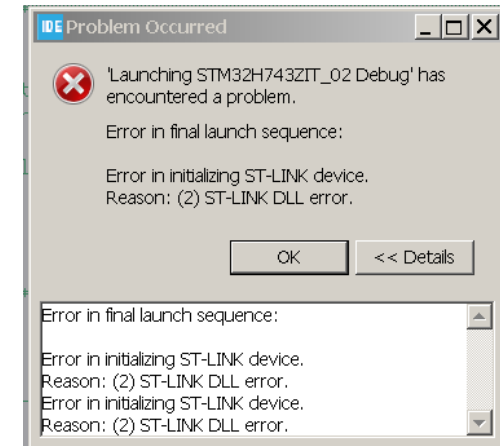
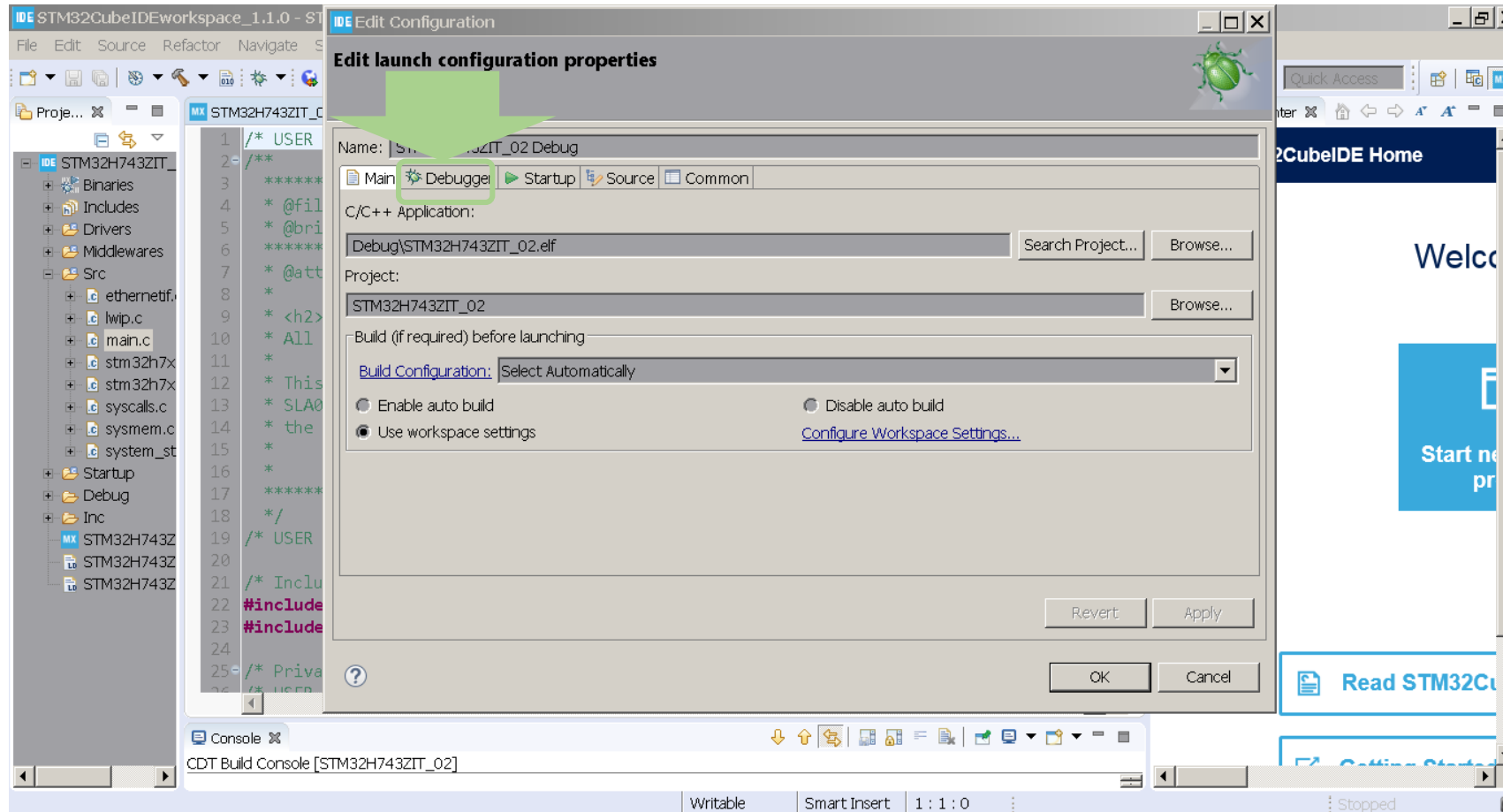


# code build



# code debug

## Run/Debug





# Es fehlt noch

- Allgemeines:
  - Zu verwendende Entwicklungsumgebung: STM32CubeIDE
  - Implementierung in neuem STM32-Projekt
  - Implementierung **ohne** Operation System (RTOS, o.ä.), falls möglich
- Aufgabenstellung:
  - Implementierung von Ethernet auf einem STM32H743ZIT6 Microcontroller
    - Entweder TCP oder UDP. Please implement the safer one.
  - Senden von Daten (25 Werte (float 32 bit)) zu einer Adresse
    - Werte vor dem Senden in JSON-Format umwandeln (besser für Backend, Web)
  - Empfangen von Daten (25 Werte (float 32 bit)) von einer Adresse
    - Werte wieder in float zurück wandeln
- Zusätzliche Fragestellungen:
  - Unterschied TCP und UDP erklären
  - Aufbau JSON-Nachricht
  - Wie können die im JSON-Format gesendeten Daten in einem Backend abgelegt werden (Schrittfolge aufzeigen, kein Code)?
- Präsentation (in Powerpoint):
  - Vorstellung der Aufgabenstellung in STM32CubeIDE
  - Vorstellung der zusätzlichen Fragestellungen