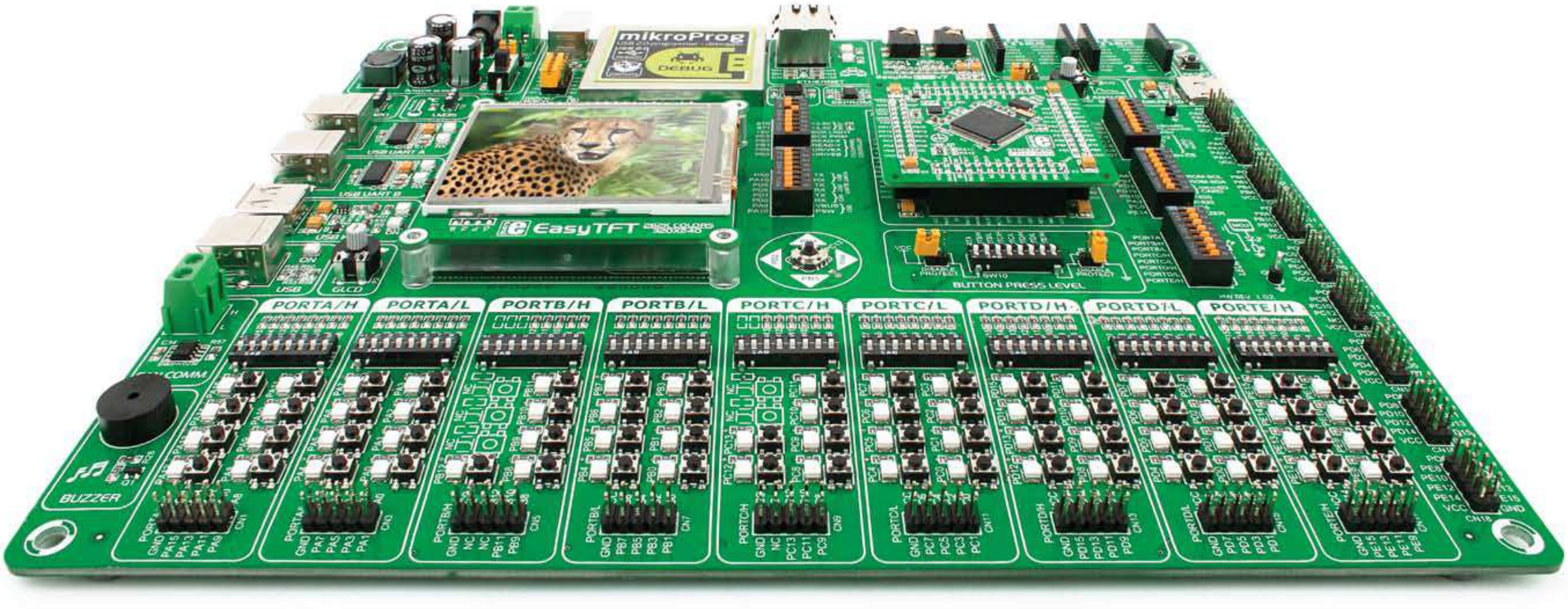
# EasyMx PRO™ v7 for STM32 ARM®

USER'S GUIDE



**186**

microcontrollers supported

**The ultimate STM32 board**

Many on-board modules

**Multimedia peripherals**

mikro BUS

Easy-add extra boards

**mikroBUS™ sockets**

connectivity

Two connectors for each port

**Amazing Connectivity**

Fast USB 2.0 programmer and

**In-Circuit Debugger**

MikroElektronika
DEVELOPMENT TOOLS | COMPILERS | BOOKS

# To our valued customers

EasyMx PRO™ v7 for STM32 is our first development board for STM32 devices. We have put all of our knowledge that we gained in the past 10 years of developing embedded systems into it's design, functionality and quality. It may be our first STM32 development board, but it sure looks and feels like it's our 7th.

You made the right choice. But the fun has only just begun!

Nebojsa Matic,  
Owner and General Manager  
of mikroElektronika

# Table of contents

## Introduction

## Power Supply

## Supported MCUs

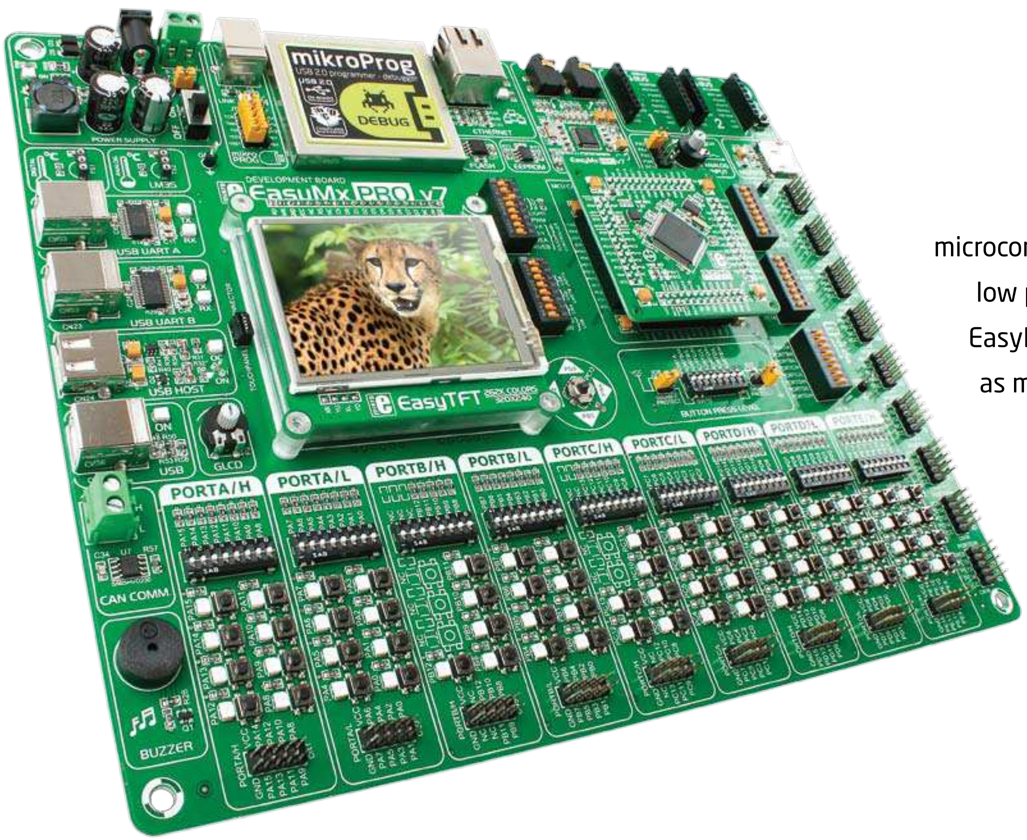
## Programmer/debugger

## Connectivity

## Communication

## Multimedia

## Other Modules



# Introduction

ARM® Cortex™-M3 and Cortex™-M4 are increasingly popular microcontrollers. They are rich with modules, with high performance and low power consumption, so creating a development board the size of EasyMx PRO™ v7 for STM32 was really a challenge. We wanted to put as many peripherals on the board as possible, to cover many internal modules. We have gone through a process of fine tuning the board's performance, and used 4-layer PCB to achieve maximum efficiency. Finally, it had met all of our expectations, and even exceeded in some. We present you the board which is powerful, well organized, with on-board programmer and debugger and is ready to be your strong ally in development.

*EasyMx PRO™ v7 development Team*

Two connectors for each port

## Amazing connectivity

EasyMx PRO™ v7 for STM32 is all about connectivity. Having two different connectors for each port, you can connect accessory boards, sensors and your custom electronics easier then ever before.

Everything is already here

## mikroProg™ on board

Powerful on-board mikroProg™ programmer and hardware debugger can program and debug over 180 STM32 ARM® microcontrollers. You will need it, whether you are a professional or a beginner.

Ready for all kinds of development

## Multimedia peripherals

TFT 320x240 with touch panel, stereo mp3 codec, audio input and output, navigation switch and microSD card slot make a perfect set of peripherals for multimedia development.

For easier connections

## mikroBUS™ support

Just plug in your Click™ board, and it's ready to work. We picked up a set of the most useful pins you need for development and made a pinout standard you will enjoy using.

# It’s good to know

## STM32F107VCT6 is the default microcontroller

**STM32F107VCT6** is the default chip of EasyMx PRO™ v7 for STM32. It belongs to ARM® Cortex™-M3 family. It has **72MHz** frequency, **256K bytes** of Flash memory, **64K bytes** of general purpose SRAM, integrated Ethernet controller, **USB 2.0** (OTG, Host, Device), 80 General purpose **I/O pins** (mappable on 16 external interrupt), 4x16-bit **timers**, 2x12-bit **A/D** (16 channels), 2x12-bit **D/A** , 5x**UARTs**, internal Real time clock (**RTC**), 2x**I2C,** 3x**SPI** and 2x**CAN** controllers. It has Serial wire debug (SWD) and JTAG interfaces for programming and debugging.

- *Great choice for both beginners and professionals*
- *Rich with modules*
- *Comes with examples for mikroC, mikroBasic and mikroPascal compilers*

## System Specification

**power supply**
7-23V AC or 9-32V DC
or via USB cable (5V DC)

**power consumption**
~76mA when all peripheral modules are disconnected

**board dimensions**
266 x 220mm (10.47 x 8.66 inch)

**weight**
~500g (1.1 lbs)

# Package contains

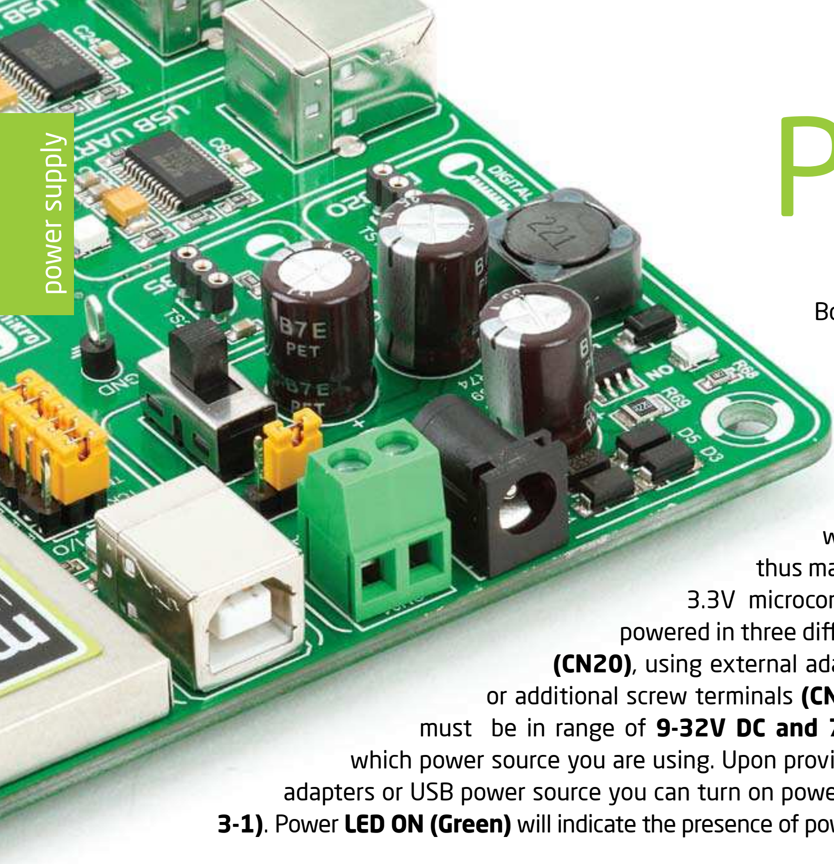
1 Damage resistant protective box

2 EasyMx PRO™ v7 for STM32 board in antistatic bag

3 USB cable

4 User Manuals and Board schematics

5 DVD with examples and documentation



# Power supply

Board contains switching power supply that creates stable voltage and current levels necessary for powering each part of the board. Power supply section contains specialized **MC33269DT3.3** power regulator which creates VCC-3.3V power supply, thus making the board capable of supporting 3.3V microcontrollers. Power supply unit can be powered in three different ways: with **USB power supply (CN20)**, using external adapters via adapter connector **(CN30)** or additional screw terminals **(CN31)**. External adapter voltage levels must be in range of **9-32V DC and 7-23V AC**. Use jumper **J9** to specify which power source you are using. Upon providing the power using either external adapters or USB power source you can turn on power supply by using **SWITCH 1 (Figure 3-1)**. Power **LED ON (Green)** will indicate the presence of power supply.

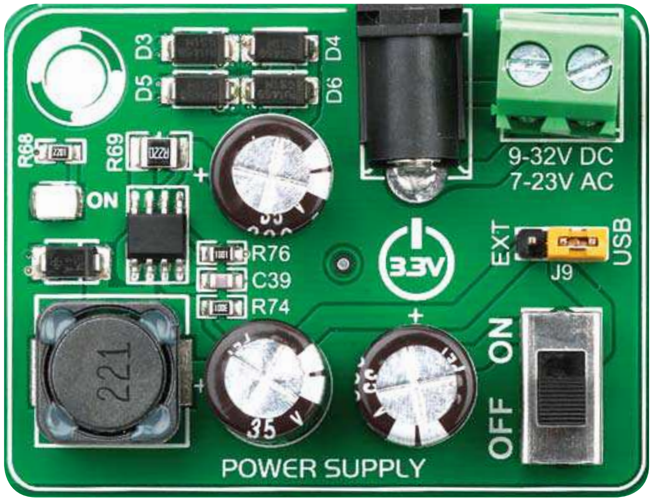
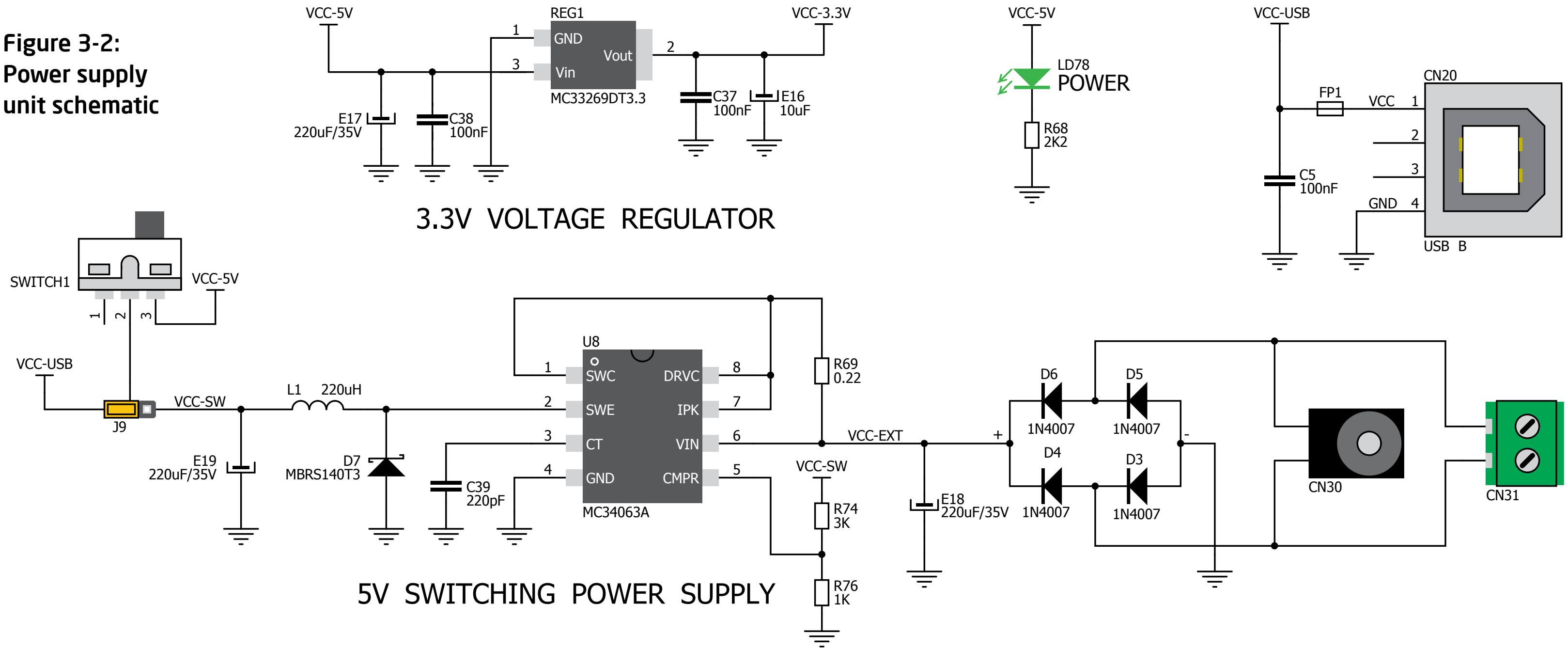


**Figure 3-1: Power supply unit of EasyMx PRO™ v7 STM32**

**Figure 3-2: Power supply unit schematic**



3.3V

Board power supply creates stable 3.3V necessary for operation of the microcontroller and all on-board modules.

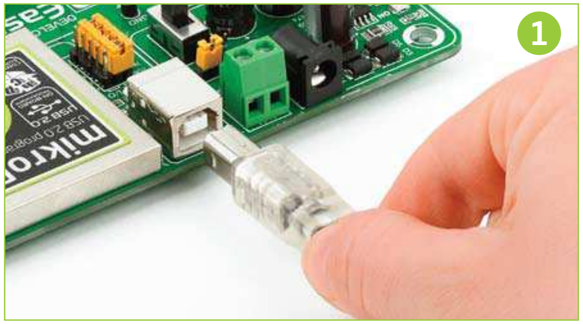
| | |
|---|---|
| **Power supply:** | via DC connector or screw terminals (7V to 23V AC or 9V to 32V DC), or via USB cable (5V DC) |
| **Power capacity:** | up to 500mA with USB, and up to 600mA with external power supply |

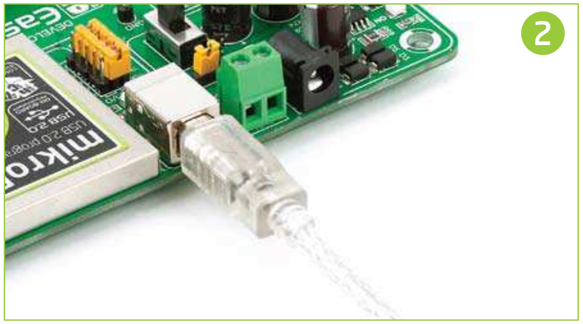
# How to power the board?

## 1. With USB cable

**Set J9 jumper to USB position**

To power the board with USB cable, place jumper **J9** in USB position. You can then plug in the USB cable as shown on images 1 and 2, and turn the power switch ON.

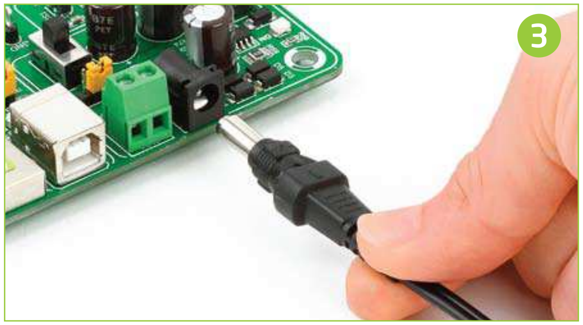


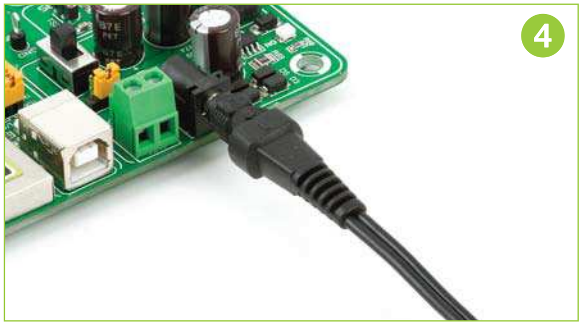


## 2. Using adapter

**Set J9 jumper to EXT position**

To power the board via adapter connector, place jumper **J9** in EXT position. You can then plug in the adapter cable as shown on images 3 and 4, and turn the power switch ON.

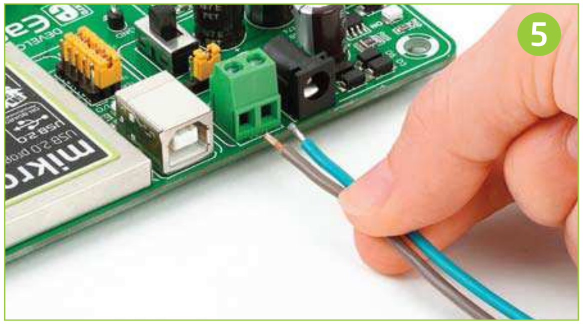


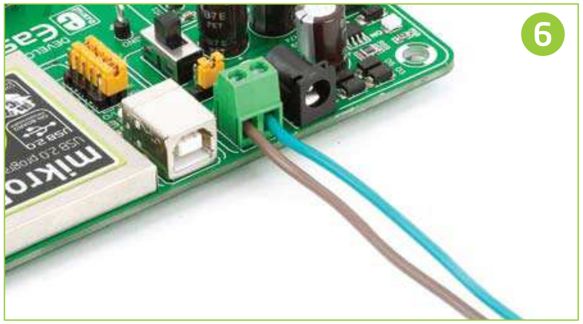


## 3. With laboratory power supply

**Set J9 jumper to EXT position**

To power the board using screw terminals, place jumper **J9** in EXT position. You can then screw-on the cables in the screw terminals as shown on images 5 and 6, and turn the power switch ON.





# Default MCU card

Microcontrollers are supported using specialized MCU cards containing 104 pins, which can be placed into the on-board female MCU socket. There are several types of cards which cover all microcontroller families of STM32 Cortex™-M3, as well as Cortex™-M4. The **Default MCU card** that comes with the EasyMx PRO™ v7 for STM32 package is shown on **Figure 4-1**. It contains **STM32F107VCT6** microcontroller with on-chip peripherals and is a great choice for both beginners and professionals. After testing and building the final program, this card can also be taken out of the board socket and used in your final device.

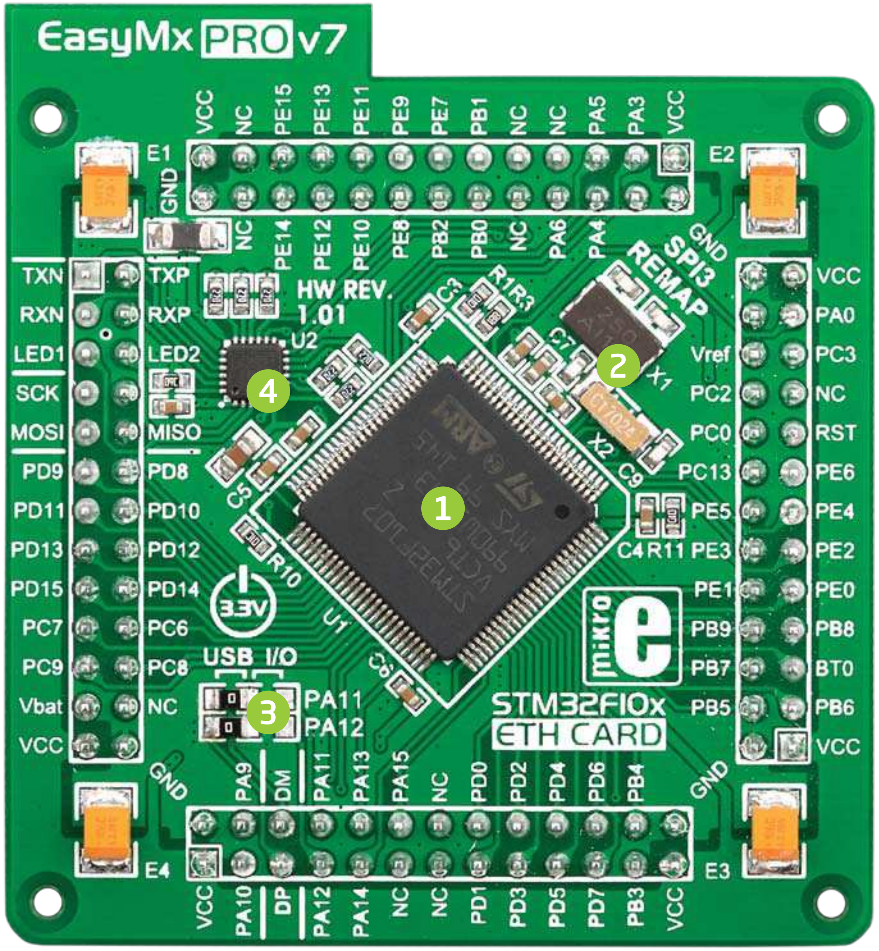

EasyMx PRO v7
VCC NC PE15 PE13 PE11 PE9 PE7 PB1 NC NC PA5 PA3 VCC
GND NC PE14 PE12 PE10 PE8 PB2 PB0 NC PA6 PA4 GND
E1
E2
E3
E4
TXN TXP
RXN RXP
LED1 LED2
SCK
MOSI MISO
PD9 PD8
PD11 PD10
PD13 PD12
PD15 PD14
PC7 PC6
PC9 PC8
Vbat NC
VCC
HW REV. 1.01
U2
U1
SPI3 REMAP
X1
X2
R1R3
C3
C5
C7
C9
R10
C4 R11
3.3V
USB I/O
PA11
PA12
STM32F10x ETH CARD
mikro e
Vref PC2 PC0 PC13 PE5 PE3 PE1 PB9 PB7 PB5
VCC PA0 PC3 NC RST PE6 PE4 PE2 PE0 PB8 BT0 PB6 VCC
GND PA9 DM PA11 PA13 PA15 NC PD0 PD2 PD4 PD6 PB4 GND
VCC PA10 DP PA12 PA14 NC NC PD1 PD3 PD5 PD7 PB3 VCC
1
2
3
4

Figure 4-1: Default MCU card with **STM32F107VCT6**

1 **STM32F107VCT6** is the default chip of EasyMx PRO™ v7. It has **72MHz** frequency, **256K bytes** of Flash memory, **64K bytes** of general-purpose SRAM, integrated Ethernet controller, **USB 2.0** (OTG, Host, Device), 80 General purpose **I/O pins** (mappable on 16 external interrupt), 4x16-bit **timers**, 2x12-bit **A/D** (16 channels), 2x12-bit **D/A** , 5x**UARTs**, internal Real time clock (**RTC**), 2x**I2C,** 3x**SPI** and 2x**CAN** controllers.

2 **25MHz crystal oscillator.** We carefully chose the most convenient crystal value that provides clock frequency which can be used directly, or with the PLL multipliers to create higher MCU clock value. MCU card also contains 32.768 kHz crystal oscillator which provides external clock for RTCC module.

3 **USB communications lines.** These two jumpers, when in USB position, connect D+ and D- lines of the on-board USB connector with PA11 and PA12 microcontroller pins. Since **STM32F107VCT6** supports USB, jumpers are in USB position.

4 **Ethernet transceiver.** Default MCU card contains single-chip Ethernet physical (PHY) layer transceiver which provides additional Ethernet functionality to **STM32F107VCT6** controller

5 With STM32 Cortex™-M3 and Cortex™-M4 microcontrollers you have the ability to select specific boot space (User flash memory, system memory or embedded SRAM), depending on the boot pins value (BT0, PB2). Boot pins are set to ground (0) through 100K resistors. In order to set BT0 and PB2 pins to VCC (1), you must push **SW11.1** and **SW11.2** DIP switches to **ON** position, **Figure 4-2**. The values on the BOOT pins are latched on the fourth rising edge of system clock after a reset.

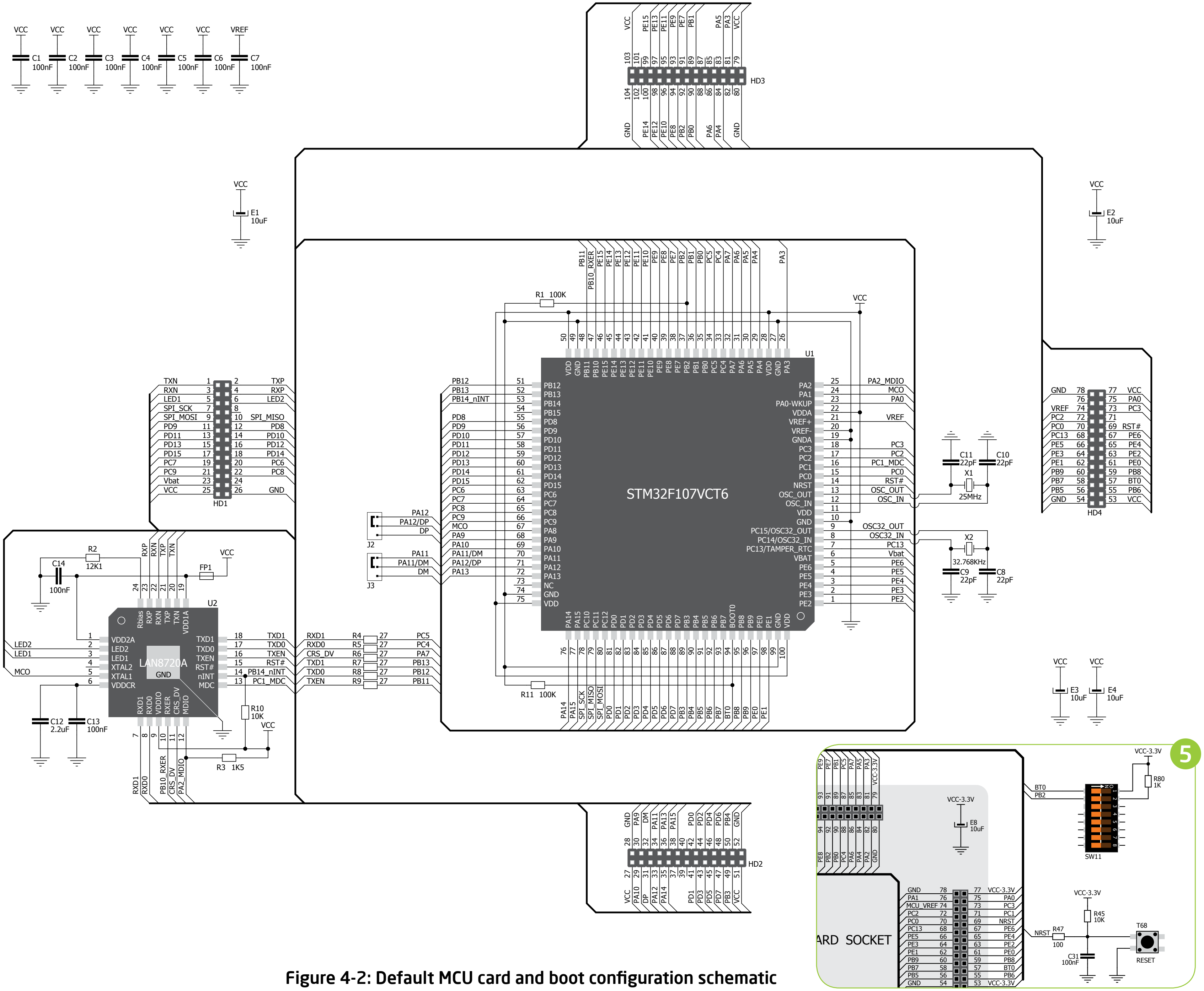


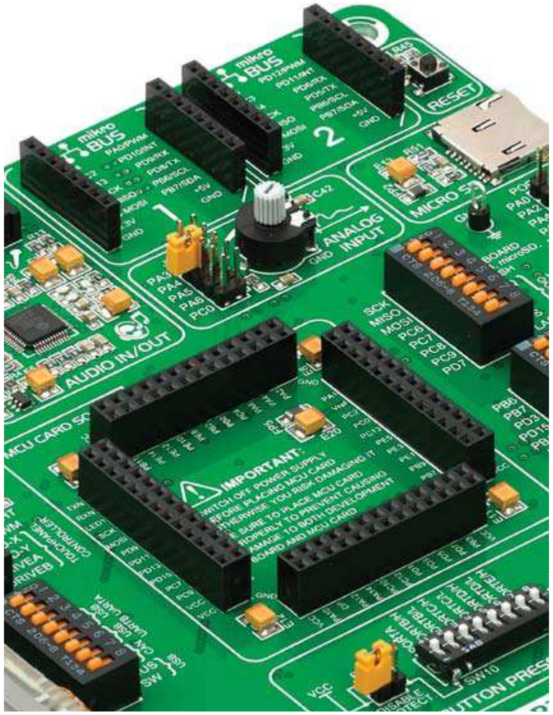
Figure 4-2: Default MCU card and boot configuration schematic

# How to properly place your MCU card into the socket?

Before you plug the microcontroller card into the socket, make sure that the **power supply is turned off.** Images below show how to correctly plug the card. First make sure that MCU card orientation matches the silkscreen outline on the EasyMx PRO™ v7 STM32 board MCU socket. Place the MCU card over the socket, so each male header is properly aligned with the female socket, as shown in **Figure 4-4**. Then put the MCU card slowly down until all the pins match the socket. Check again if everything is placed correctly and press the MCU card until it is completely plugged into the socket as shown in **Figure 4-5**. If done correctly, all pins should be fully inserted. Only now you can turn on the power supply.

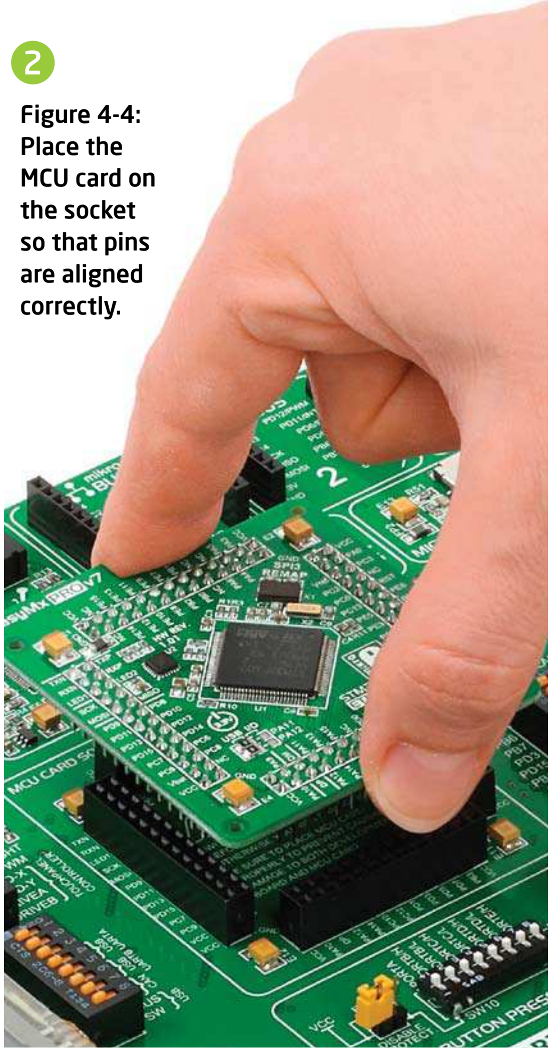
1

**Figure 4-3: On-board MCU socket has silkscreen markings which will help you to correctly orient the MCU card before inserting.**



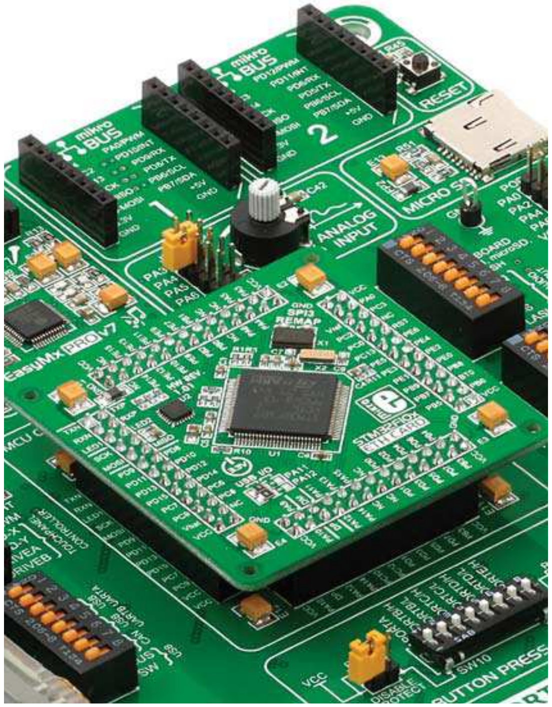
2

**Figure 4-4: Place the MCU card on the socket so that pins are aligned correctly.**



3

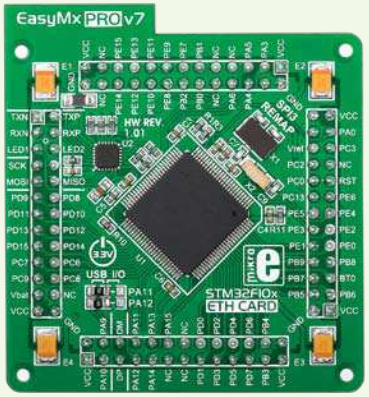
**Figure 4-5 Properly placed MCU card.**



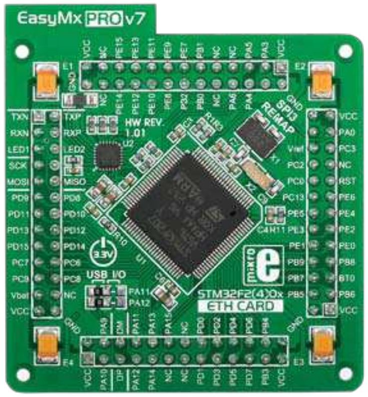
# Other supported MCU cards

mikroElektronika currently offers total of three populated MCU cards. Two with Cortex™-M3: **STM32F107VCT6** microcontroller (default), **STM32F207VGT6** microcontroller and one with Cortex™-M4: **STM32F407VGT6** microcontroller. You can also purchase empty PCB cards that you can populate on your own and solder any supported microcontroller you need in your development. There are total of four empty PCB cards available. This way your EasyMx PRO™ v7 for STM32 board becomes truly flexible and reliable tool for almost any of your ARM® projects. MCU cards can also be used in your final devices. For complete list of currently available MCU cards, please visit the board webpage:

*http://www.mikroe.com/eng/products/view/852/easymx-pro-v7-for-stm32/*



**Default 100-pin ETH MCU card with STM32F107VCT6**



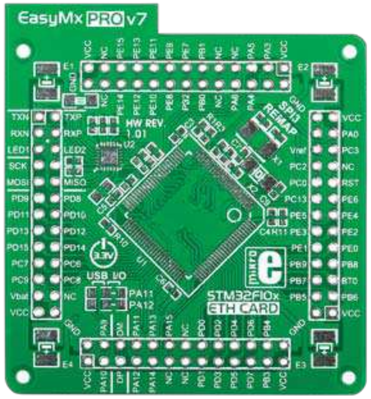
**Standard 100-pin HP ETH MCU card with STM32F207VGT6**

**Standard 100-pin HP ETH MCU card with STM32F407VGT6**

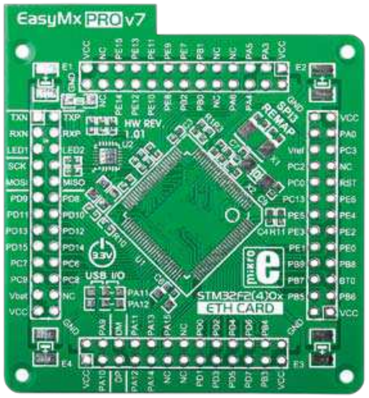
*NOTE:*

*"HP" (High performance) - Empty MCU cards that support only high performance STM32F20x and STM32F40x microcontrollers family.*

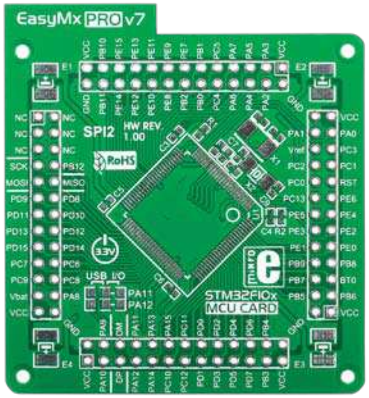
*"ETH" (Ethernet) - Empty MCU cards with single-chip Ethernet PHY layer transceiver which provides additional Ethernet functionality to microcontrollers*



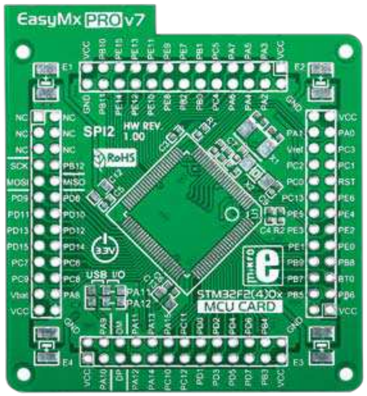
**Empty ETH MCU card for 100-pin STM32F10x MCUs**



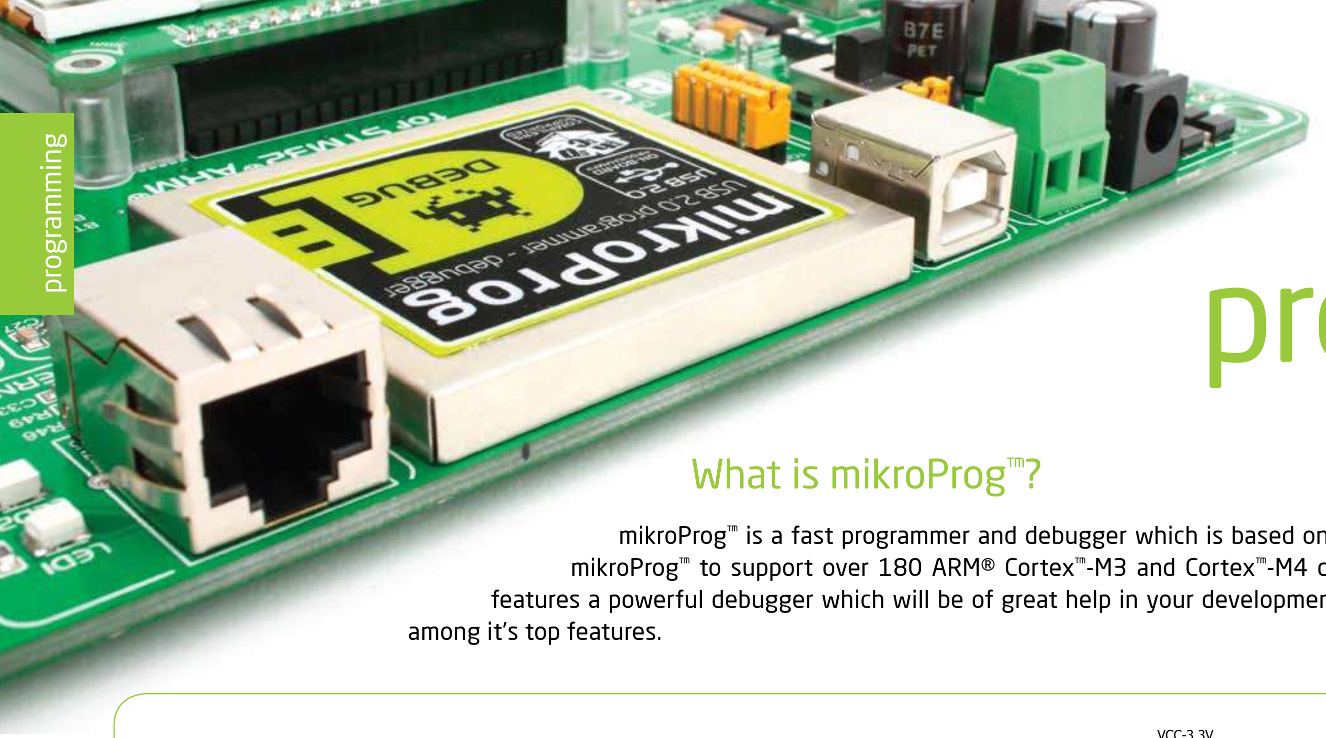
**Empty ETH HP MCU card for 100-pin STM32F2(4)0x MCUs**



**Empty MCU card for 100-pin STM32F10x MCUs**



**Empty HP MCU card for 100-pin STM32F2(4)0x MCUs**



# On-board programmer

## What is mikroProg™?

mikroProg™ is a fast programmer and debugger which is based on ST-LINK V2 programmer. Smart engineering allows mikroProg™ to support over 180 ARM® Cortex™-M3 and Cortex™-M4 devices from STM32 in a single programmer. It also features a powerful debugger which will be of great help in your development. Outstanding performance and easy operation are among it's top features.

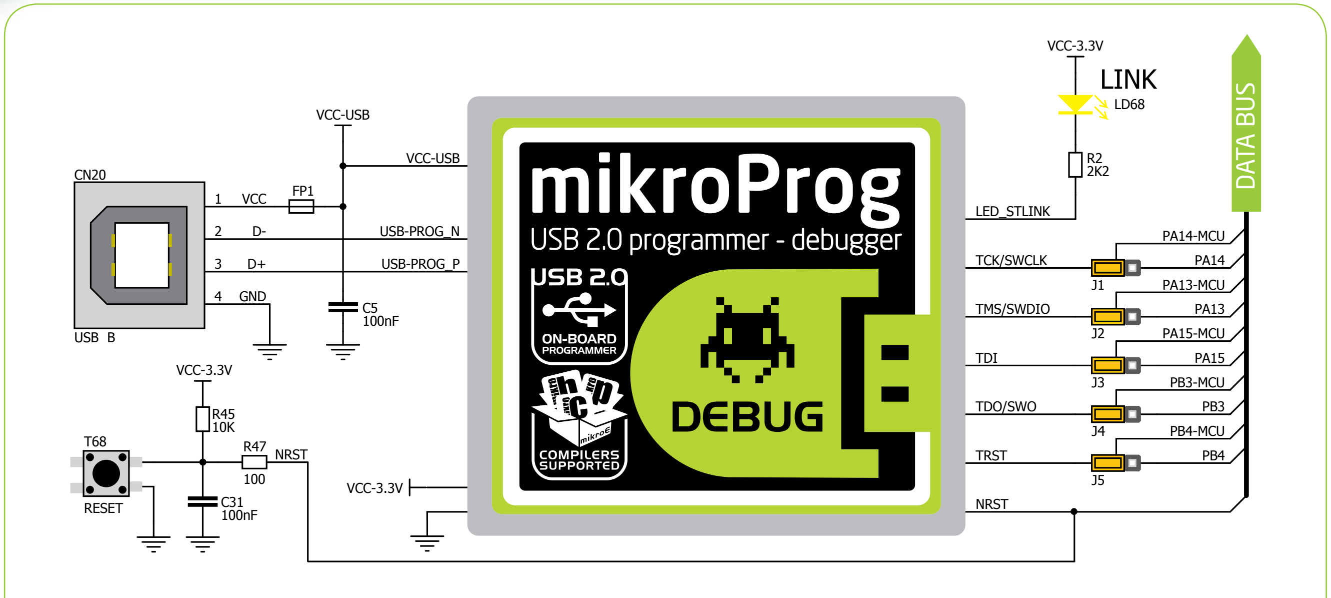
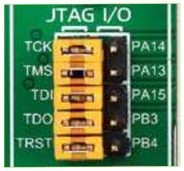
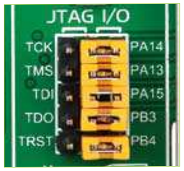


**Figure 5-1: mikroProg™ block schematics**

## Enabling mikroProg™





Five jumpers below the programmer USB connector are used to specify whether programming lines should be connected to programmer, or used as general purpose I/Os. If placed in **JTAG/SWD position**, jumpers connect **PA13-PA15** pins to TMS/SWDIO, TCK/SWCLK , TDI, and **PB3-PB4** pins to TDO/SWO and TRST programming lines respectively and are cut off from the rest of the board.

## How do I start?

In order to start using mikroProg™, and program your microcontroller, you just have to follow two simple steps:

**1. Install the necessary software**

- Install programmer drivers
- Install mikroProg Suite™ for ARM® software

**2. Power up the board, and you are ready to go.**

- Plug in the programmer USB cable
- LINK LED should light up.

## STM32 Cortex™-M3 microcontrollers supported with mikroProg™

STM32F100C4
STM32F100C6
STM32F100C8
STM32F100CB
STM32F100R4
STM32F100R6
STM32F100R8
STM32F100RB
STM32F100RC
STM32F100RD
STM32F100RE
STM32F100V8
STM32F100VB
STM32F100VC
STM32F100VD
STM32F100VE
STM32F100ZC
STM32F100ZD
STM32F100ZE
STM32F101C4
STM32F101C6
STM32F101C8
STM32F101CB
STM32F101R4

STM32F101R6
STM32F101R8
STM32F101RB
STM32F101RC
STM32F101RD
STM32F101RE
STM32F101RF
STM32F101RG
STM32F101T4
STM32F101T6
STM32F101T8
STM32F101TB
STM32F101V8
STM32F101VB
STM32F101VC
STM32F101VD
STM32F101VE
STM32F101VF
STM32F101VG
STM32F101ZC
STM32F101ZD
STM32F101ZE
STM32F101ZG
STM32F102C4

STM32F102C6
STM32F102C8
STM32F102CB
STM32F102R4
STM32F102R6
STM32F102R8
STM32F102RB
STM32F103C4
STM32F103C6
STM32F103C8
STM32F103CB
STM32F103R4
STM32F103R6
STM32F103R8
STM32F103RB
STM32F103RC
STM32F103RD
STM32F103RE
STM32F103RF
STM32F103RG
STM32F103T4
STM32F103T6
STM32F103T8
STM32F103TB

STM32F103V8
STM32F103VB
STM32F103VC
STM32F103VD
STM32F103VE
STM32F103VF
STM32F103VG
STM32F103ZC
STM32F103ZD
STM32F103ZE
STM32F103ZF
STM32F103ZG
STM32F105R8
STM32F105RB
STM32F105RC
STM32F105V8
STM32F105VB
STM32F105VC
STM32F107RB
STM32F107RC
STM32F107VB
STM32F107VC
STM32F205RB
STM32F205RC

STM32F205RE
STM32F205RF
STM32F205RG
STM32F205VB
STM32F205VC
STM32F205VE
STM32F205VF
STM32F205VG
STM32F205ZC
STM32F205ZE
STM32F205ZF
STM32F205ZG
STM32F207IC
STM32F207IE
STM32F207IF
STM32F207IG
STM32F207VC
STM32F207VE
STM32F207VF
STM32F207VG
STM32F207ZC
STM32F207ZE
STM32F207ZF
STM32F207ZG

STM32F215RE
STM32F215RG
STM32F215VE
STM32F215VG
STM32F215ZE
STM32F215ZG
STM32F217IE
STM32F217IG
STM32F217VE
STM32F217VG
STM32F217ZE
STM32F217ZG
STM32L151C6
STM32L151C8
STM32L151CB
STM32L151QC
STM32L151QD
STM32L151R6
STM32L151R8
STM32L151RB
STM32L151RC
STM32L151RD
STM32L151V8
STM32L151VB

STM32L151VC
STM32L151VD
STM32L151ZC
STM32L151ZD
STM32L152C6
STM32L152C8
STM32L152CB
STM32L152QC
STM32L152QD
STM32L152R6
STM32L152R8
STM32L152RB
STM32L152RC
STM32L152RD
STM32L152V8
STM32L152VB
STM32L152VC
STM32L152VD
STM32L152ZC
STM32L152ZD
STM32L162QD
STM32L162RD
STM32L162VD
STM32L162ZD

## STM32 Cortex™-M4 microcontrollers supported with mikroProg™

STM32F405RG
STM32F405VG
STM32F405ZG

STM32F407IE
STM32F407IG
STM32F407VE

STM32F407VG
STM32F407ZE
STM32F407ZG

STM32F415RG
STM32F415VG
STM32F415ZG

STM32F417IE
STM32F417IG
STM32F417VE

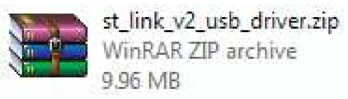
STM32F417VG
STM32F417ZE
STM32F417ZG

# Installing programmer drivers

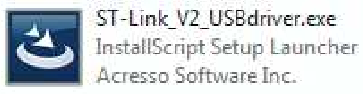
On-board mikroProg™ requires drivers in order to work. Drivers are located on the **Product DVD** that you received with the EasyMx PRO™ v7 for STM32 package:

*DVD://download/eng/software/development-tools/arm/stm32/mikroprog/st_link_v2_usb_driver.zip*

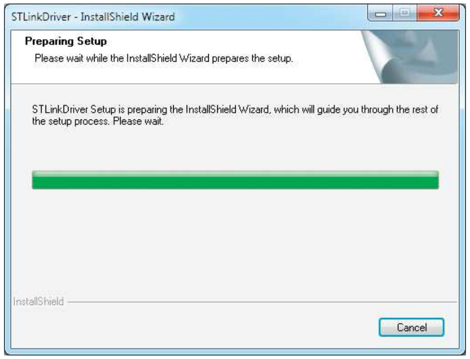
When you locate the drivers, please extract the setup file from the ZIP archive. You should be able to locate the driver setup file. Double click the setup file to begin installation of the programmer drivers.



st_link_v2_usb_driver.zip
WinRAR ZIP archive
9.96 MB



ST-Link_V2_USBdriver.exe
InstallScript Setup Launcher
Acresso Software Inc.

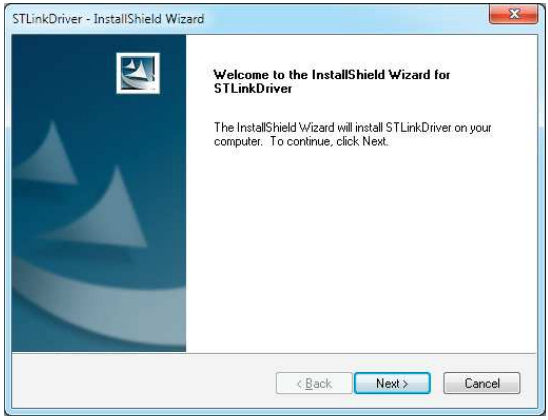


STLinkDriver - InstallShield Wizard

**Preparing Setup**
Please wait while the InstallShield Wizard prepares the setup.

STLinkDriver Setup is preparing the InstallShield Wizard, which will guide you through the rest of the setup process. Please wait.

InstallShield



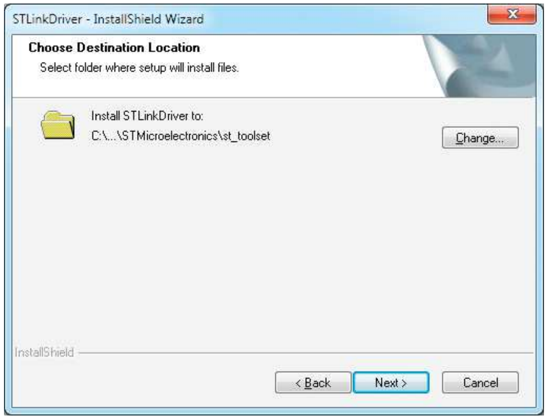
STLinkDriver - InstallShield Wizard

**Welcome to the InstallShield Wizard for STLinkDriver**

The InstallShield Wizard will install STLinkDriver on your computer. To continue, click Next.

## Step 1 - Start Installation

Welcome screen of the installation. Just click on **Next** button to proceed.



STLinkDriver - InstallShield Wizard

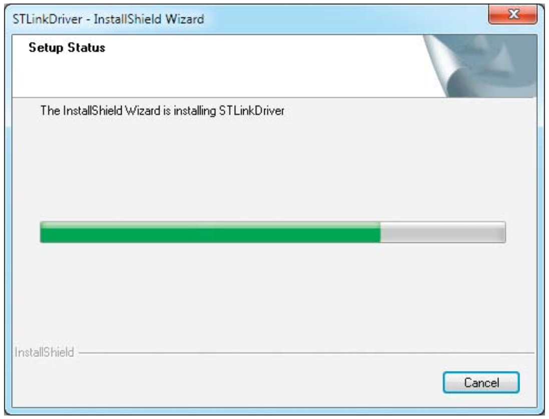
**Choose Destination Location**
Select folder where setup will install files.

Install STLinkDriver to:
C:\...\STMicroelectronics\st_toolset

InstallShield

## Step 2 - Select Destination

Click **Change** button to select new destination folder or use the suggested installation path.



STLinkDriver - InstallShield Wizard

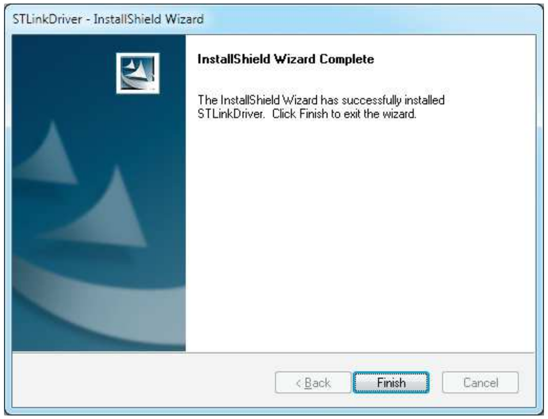
**Setup Status**

The InstallShield Wizard is installing STLinkDriver

InstallShield

## Step 3 - Installing drivers

Drivers are installed automatically in a matter of seconds.



STLinkDriver - InstallShield Wizard

**InstallShield Wizard Complete**

The InstallShield Wizard has successfully installed STLinkDriver. Click Finish to exit the wizard.

## Step 4 - Finish installation

You will be informed if the drivers are installed correctly. Click on **Finish** button to end installation process.

# Programming software

## mikroProg Suite™ for ARM®

On-board **mikroProg™** programmer requires special programming software called **mikroProg Suite™ for ARM®**. This software is used for programming of all supported microcontroller families with ARM® Cortex™-M3 and Cortex™-M4 cores. Software has intuitive interface and **SingleClick™** programming technology. To begin, first locate the installation archive on the Product DVD:

*DVD://download/eng/software/development-tools/arm/mikroprog/mikroprog_suite_for_arm_v110.zip*

After downloading, extract the package and double click the executable setup file, to start installation.

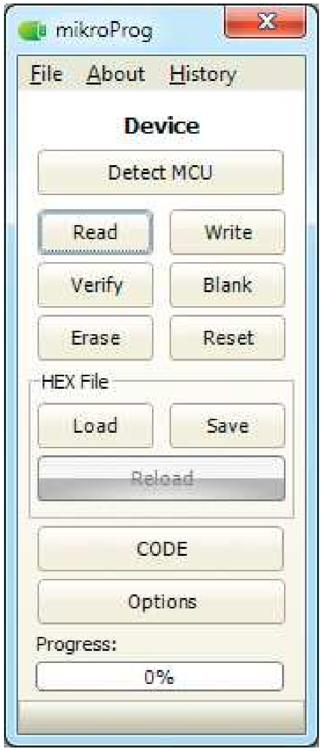


Figure 5-2: mikroProg Suite™ for ARM® window

## Quick Guide

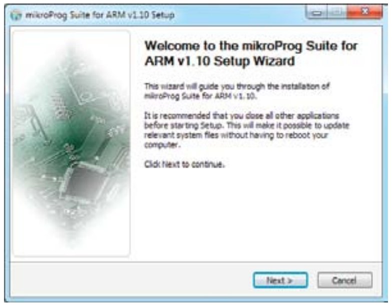
1. Click the **Detect MCU** button in order to recognize the device ID.

2. Click the **Read** button to read the entire microcontroller memory. You can click the **Save** button to save it to target HEX file.

3. If you want to write the HEX file to the microcontroller, first make sure to load the target HEX file. You can drag-n-drop the file onto the software window, or use the **Load** button to open **Browse dialog** and point to the HEX file location. Then click the **Write** button to begin programming.

4. Click the **Erase** button to wipe out the microcontroller memory.

## Installation wizard - 6 simple steps



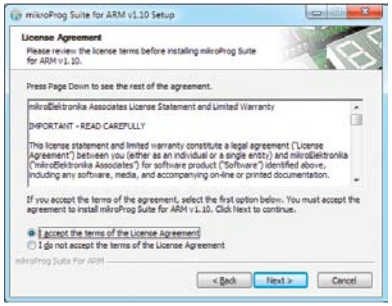
**Welcome to the mikroProg Suite for ARM v1.10 Setup Wizard**

This wizard will guide you through the installation of mikroProg Suite for ARM v1.10.

It is recommended that you close all other applications before starting Setup. This will make it possible to update relevant system files without having to reboot your computer.

Click Next to continue.

### Step 1 - Start Installation



**License Agreement**

Please review the license terms before installing mikroProg Suite for ARM v1.10.

Press Page Down to see the rest of the agreement.

mikroElektronika Associates License Statement and Limited Warranty

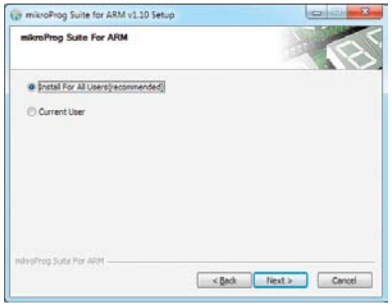
IMPORTANT - READ CAREFULLY

This license statement and limited warranty constitute a legal agreement ("License Agreement") between you (either as an individual or a single entity) and mikroElektronika ("mikroElektronika Associates") for software product ("Software") identified above, including any software, media, and accompanying on-line or printed documentation.

If you accept the terms of the agreement, select the first option below. You must accept the agreement to install mikroProg Suite for ARM v1.10. Click Next to continue.

● I accept the terms of the License Agreement
○ I do not accept the terms of the License Agreement

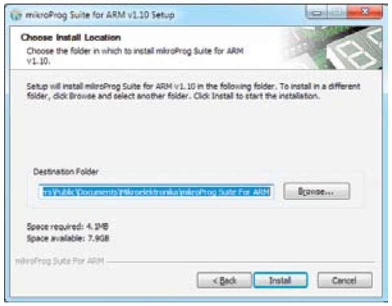
### Step 2 - Accept EULA and continue



**mikroProg Suite For ARM**

● Install For All Users(recommended)
○ Current User

### Step 3 - Install for All users or current user



**Choose Install Location**

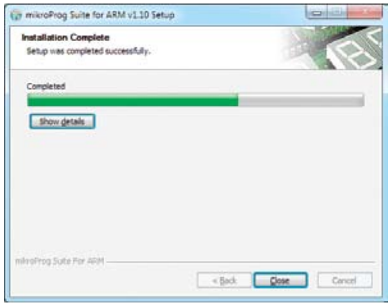
Choose the folder in which to install mikroProg Suite for ARM v1.10.

Setup will install mikroProg Suite for ARM v1.10 in the following folder. To install in a different folder, click Browse and select another folder. Click Install to start the installation.

Destination Folder: rs\Public\Documents\Mikroelektronika\mikroProg Suite For ARM

Space required: 4.1MB
Space available: 7.9GB

### Step 4 - Choose destination folder

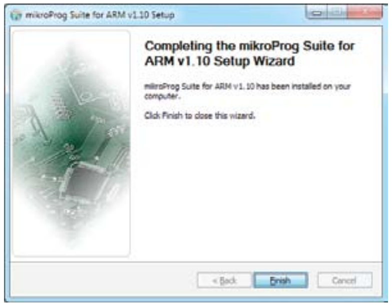


**Installation Complete**

Setup was completed successfully.

Completed

### Step 5 - Installation in progress



**Completing the mikroProg Suite for ARM v1.10 Setup Wizard**

mikroProg Suite for ARM v1.10 has been installed on your computer.

Click Finish to close this wizard.

### Step 6 - Finish Installation

# Hardware Debugger

## What is Debugging?

Every developer comes to a point where he has to monitor the code execution in order to find errors in the code, or simply to see if everything is going as planed. This hunt for bugs, or errors in the code is called **debugging**. There are two ways to do this: one is **the software simulation**, which enables you to simulate what is supposed to be happening on the microcontroller as your code lines are executed, and the other, most reliable one, is monitoring the code execution on the chip itself. And this latter one is called **hardware debugging**. “hardware” means that it is the real deal - code executes right on the target device.

## What is hardware debugger?

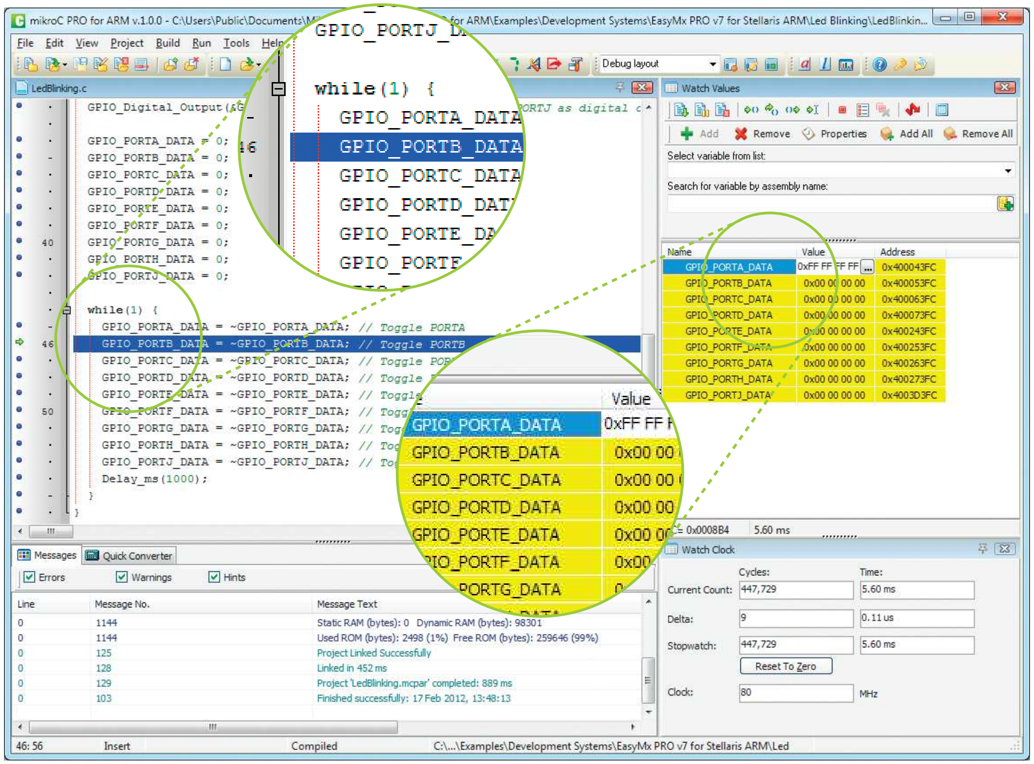
The on-board **mikroProg**™ programmer supports **hardware debugger** - a highly effective tool for a **Real-Time debugging** on hardware level. The debugger enables you to execute your program on the host STM32 microcontroller and view variable values, Special Function Registers (SFR), RAM, CODE and EEPROM memory along with the code execution on hardware. Whether you are a beginner, or a professional, this powerful tool, with intuitive interface and convenient set of commands will enable you to track down bugs quickly. mikroProg debugger is one of the fastest, and most reliable debugging tools on the market.

## Supported Compilers

All MikroElektronika compilers, **mikroC**™, **mikroBasic**™ and **mikroPascal**™ for ARM® natively support mikroProg™ for STM32, as well as other compilers, including KEIL®, IAR®. Specialized DLL module allows compilers to exploit the full potential of fast hardware debugging. Along with compilers, make sure to install the appropriate **programmer drivers** and **mikroProg Suite**™ **for ARM®** programming software, as described on **pages 14** and **15**.

## How do I use the debugger?

When you build your project for debugging, and program the microcontroller with this HEX file, you can start the debugger using **[F9]** command. Compiler will change layout to debugging view, and a blue line will mark where code execution is currently paused. Use **debugging toolbar** in the **Watch Window** to guide the program execution, and stop anytime. Add the desired variables to Watch Window and monitor their values.


mikroC PRO for ARM v.1.0.0 - C:\Users\Public\Documents\Mi... for ARM\Examples\Development Systems\EasyMx PRO v7 for Stellaris ARM\Led Blinking\LedBlinkin...
File Edit View Project Build Run Tools Help
Debug layout
LedBlinking.c
Watch Values
Add
Remove
Properties
Add All
Remove All
Select variable from list:
Search for variable by assembly name:
Watch Clock
Reset To Zero
Messages
Quick Converter
Errors
Warnings
Hints
46: 56
Insert
Compiled
C:\...\Examples\Development Systems\EasyMx PRO v7 for Stellaris ARM\Led

```
  GPIO_Digital_Output(&G                        ORTJ as digital c

  GPIO_PORTA_DATA = 0;
  GPIO_PORTB_DATA = 0;
  GPIO_PORTC_DATA = 0;
  GPIO_PORTD_DATA = 0;
  GPIO_PORTE_DATA = 0;
  GPIO_PORTF_DATA = 0;
  GPIO_PORTG_DATA = 0;
  GPIO_PORTH_DATA = 0;
  GPIO_PORTJ_DATA = 0;

  while(1) {
    GPIO_PORTA_DATA = ~GPIO_PORTA_DATA; // Toggle PORTA
    GPIO_PORTB_DATA = ~GPIO_PORTB_DATA; // Toggle PORTB
    GPIO_PORTC_DATA = ~GPIO_PORTC_DATA; // Toggle POR
    GPIO_PORTD_DATA = ~GPIO_PORTD_DATA; // Toggle
    GPIO_PORTE_DATA = ~GPIO_PORTE_DATA; // Toggle
    GPIO_PORTF_DATA = ~GPIO_PORTF_DATA; // Togg
    GPIO_PORTG_DATA = ~GPIO_PORTG_DATA; // Tog
    GPIO_PORTH_DATA = ~GPIO_PORTH_DATA; // To
    GPIO_PORTJ_DATA = ~GPIO_PORTJ_DATA; // To
    Delay_ms(1000);
  }
}
```

| Name | Value | Address |
|---|---|---|
| GPIO_PORTA_DATA | 0xFF FF FF FF | 0x400043FC |
| GPIO_PORTB_DATA | 0x00 00 00 00 | 0x400053FC |
| GPIO_PORTC_DATA | 0x00 00 00 00 | 0x400063FC |
| GPIO_PORTD_DATA | 0x00 00 00 00 | 0x400073FC |
| GPIO_PORTE_DATA | 0x00 00 00 00 | 0x400243FC |
| GPIO_PORTF_DATA | 0x00 00 00 00 | 0x400253FC |
| GPIO_PORTG_DATA | 0x00 00 00 00 | 0x400263FC |
| GPIO_PORTH_DATA | 0x00 00 00 00 | 0x400273FC |
| GPIO_PORTJ_DATA | 0x00 00 00 00 | 0x4003D3FC |

= 0x0008B4 5.60 ms

| | Cycles: | Time: |
|---|---|---|
| Current Count: | 447,729 | 5.60 ms |
| Delta: | 9 | 0.11 us |
| Stopwatch: | 447,729 | 5.60 ms |
| Clock: | 80 | MHz |

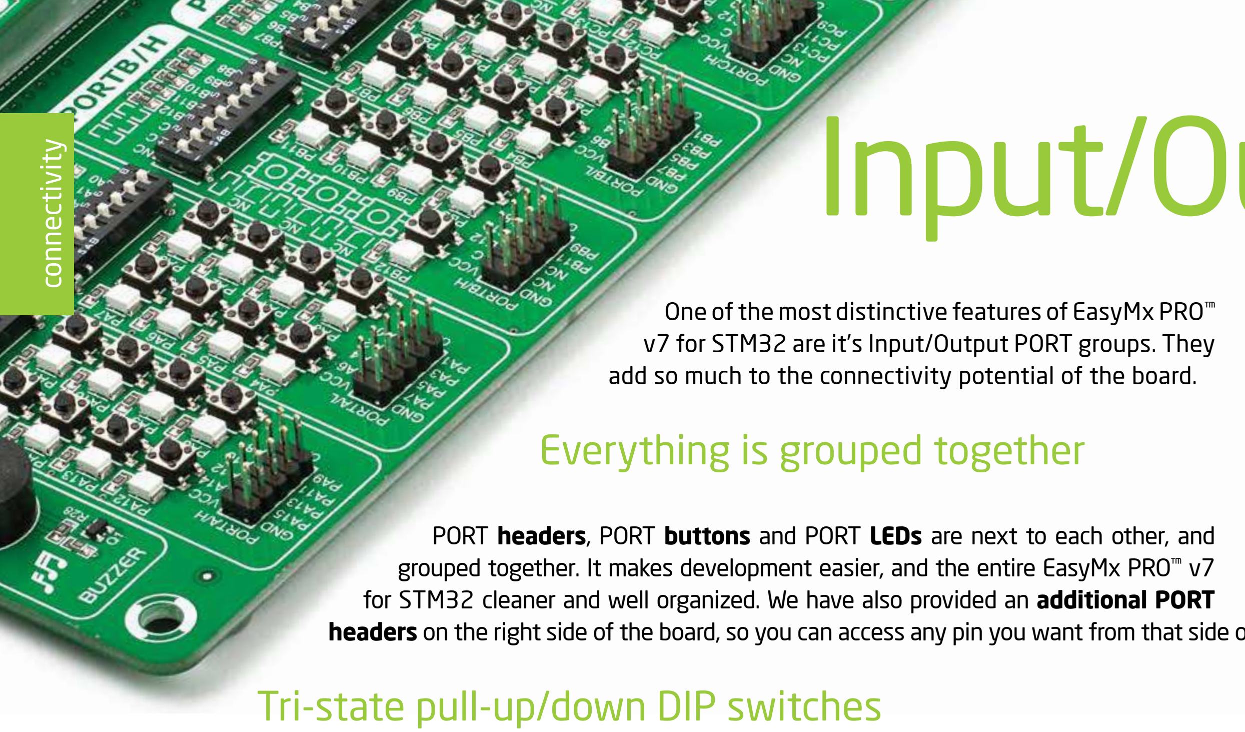
| Line | Message No. | Message Text |
|---|---|---|
| 0 | 1144 | Static RAM (bytes): 0 Dynamic RAM (bytes): 98301 |
| 0 | 1144 | Used ROM (bytes): 2498 (1%) Free ROM (bytes): 259646 (99%) |
| 0 | 125 | Project Linked Successfully |
| 0 | 128 | Linked in 452 ms |
| 0 | 129 | Project 'LedBlinking.mcpar' completed: 889 ms |
| 0 | 103 | Finished successfully: 17 Feb 2012, 13:48:13 |

**Figure 5-3: mikroC PRO for ARM® compiler in debugging view, with SFR registers in Watch Window**

# Debugger commands

Here is a short overview of which debugging commands are supported in mikroElektronika compilers. You can see what each command does, and what are their shortcuts when you are in debugging mode. It will give you some general picture of what your debugger can do.

| Toolbar Icon | Command Name | Shortcut | Description |
|---|---|---|---|
|  | Start Debugger | [F9] | Starts Debugger. |
|  | Run/Pause Debugger | [F6] | Run/Pause Debugger. |
|  | Stop Debugger | [Ctrl + F2] | Stops Debugger. |
| ⇨() | Step Into | [F7] | Executes the current program line, then halts. If the executed program line calls another routine, the debugger steps into the routine and halts after executing the first instruction within it. |
|  | Step Over | [F8] | Executes the current program line, then halts. If the executed program line calls another routine, the debugger will not step into it. The whole routine will be executed and the debugger halts at the first instruction following the call. |
| ()⇨ | Step Out | [Ctrl + F8] | Executes all remaining program lines within the subroutine. The debugger halts immediately upon exiting the subroutine. |
| ⇨I | Run To Cursor | [F4] | Executes the program until reaching the cursor position. |
| ▣ | Toggle Breakpoints | [F5] | Toggle breakpoints option sets new breakpoints or removes those already set at the current cursor position. |
|  | Show/Hide breakpoints | [Shift+F4] | Shows/Hides window with all breakpoints |
|  | Clears breakpoints | [Shift+Ctrl+F5] | Delete selected breakpoints |
|  | Jump to interrupt | [F2] | Opens window with available interrupts (doesn’t work in hardware debug mode) |



# Input/Output Group

One of the most distinctive features of EasyMx PRO™ v7 for STM32 are it's Input/Output PORT groups. They add so much to the connectivity potential of the board.

## Everything is grouped together

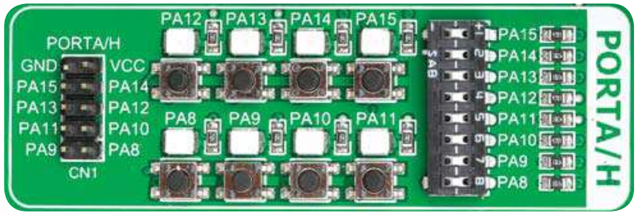


Figure 6-1: I/O group contains PORT header, tri-state pull up/down DIP switch, buttons and LEDs all in one place

PORT **headers**, PORT **buttons** and PORT **LEDs** are next to each other, and grouped together. It makes development easier, and the entire EasyMx PRO™ v7 for STM32 cleaner and well organized. We have also provided an **additional PORT headers** on the right side of the board, so you can access any pin you want from that side of the board too.

## Tri-state pull-up/down DIP switches

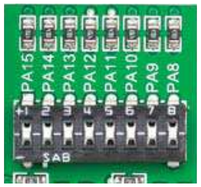
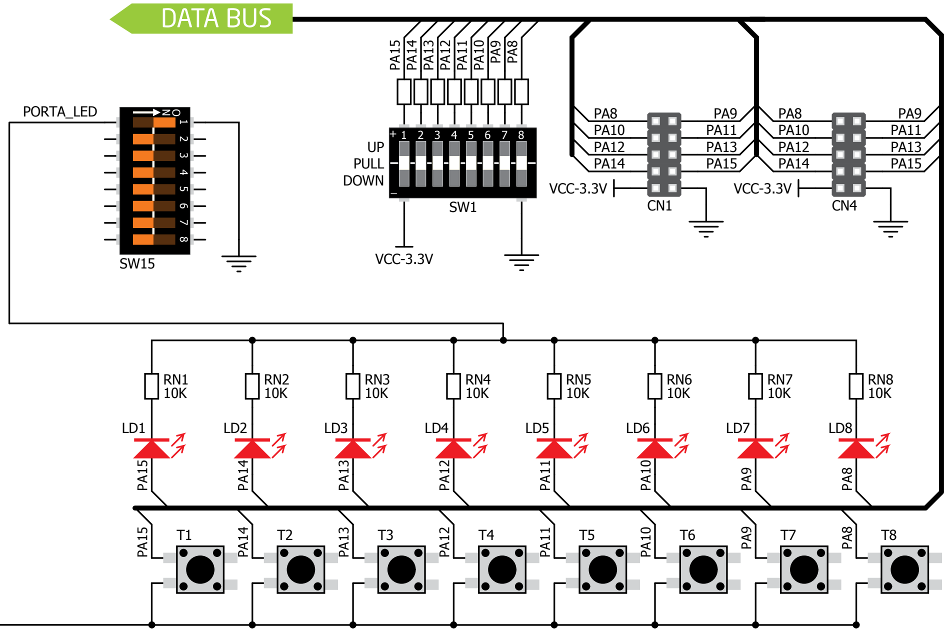
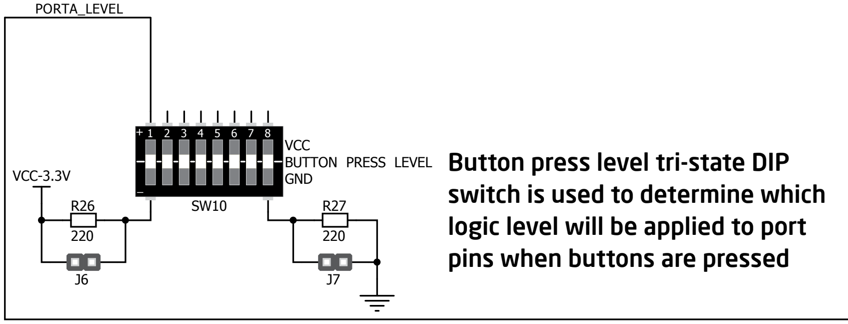


Figure 6-2: Tri-state DIP switch on PORTA/H

Tri-state DIP switches, like **SW1** on **Figure 6-3**, are used to enable 4K7 pull-up or pull-down resistor on any desired port pin. Each of these switches has three states:

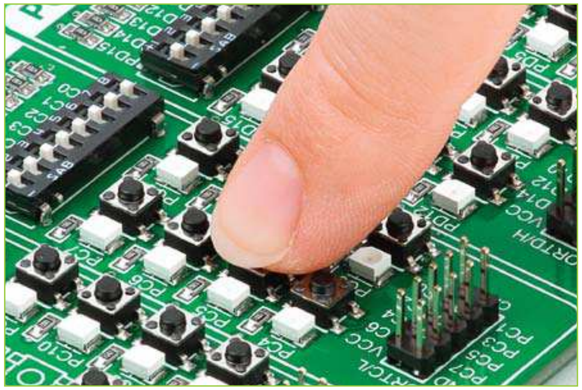
1. **middle position** disables both pull-up and pull-down feature from the PORT pin
2. **up position** connects the resistor in pull-up state to the selected pin
3. **down position** connects the resistor in pull-down state to the selected PORT pin.

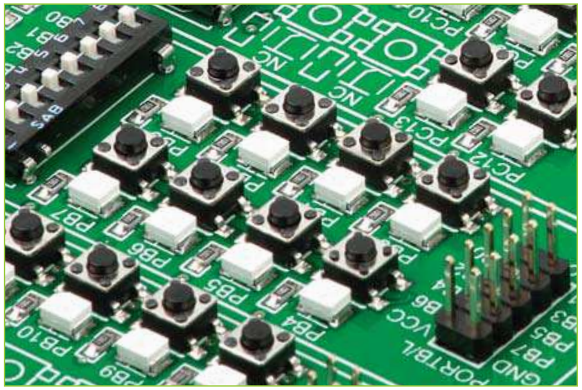




Button press level tri-state DIP switch is used to determine which logic level will be applied to port pins when buttons are pressed

Figure 6-3: Schematic of the single I/O group connected to microcontroller PORTA/H





# Headers

With enhanced connectivity as one of the key features of EasyMx PRO™ v7 for STM32, we have provided **two connection headers for each PORT**. I/O PORT group contains one male IDC10 header (like **CN1 Figure 6-3**). There is **one more IDC10 header** available on the right side of the board, next to DIP switches (like **CN4** on **Figure 6-3**). These headers can be used to connect accessory boards with IDC10 female sockets.

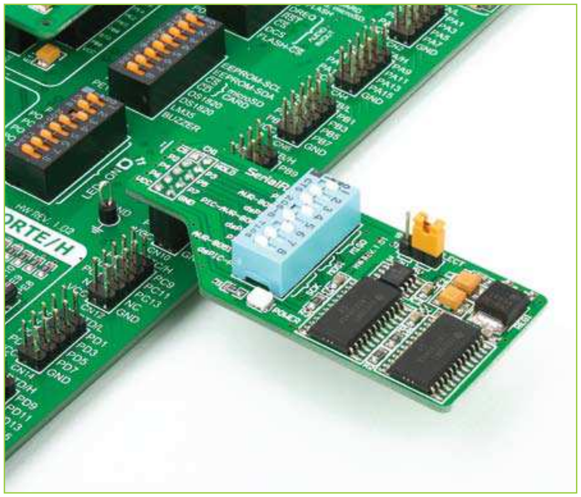


**Figure 6-4: IDC10 male headers enable easy connection with mikroElektronika accessory boards**

# Buttons

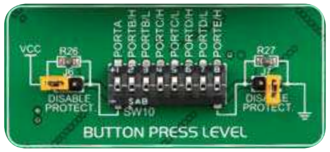


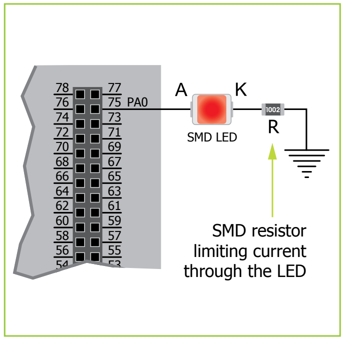
**Figure 6-5: Button press level DIP switch (tri-state)**

The logic state of all microcontroller digital inputs may be changed using **push buttons**. Tri-state DIP switch **SW10** is available for selecting which logic state will be applied to corresponding MCU pin when button is pressed, for each I/O port separately. If you, for example, place **SW10.1** in **VCC** position, then pressing of any push button in PORTA/H I/O group will apply logic one to the appropriate microcontroller pin. The same goes for **GND**. If DIP switch is in the middle position neither of two logic states will be applied to the appropriate microcontroller pin. You can disable pin protection 220ohm resistors by placing jumpers **J6** and **J7**, which will connect your push buttons directly to VCC or GND. Be aware that doing so you may accidentally damage MCU in case of wrong usage.

## Reset Button

In the far upper right section of the board, there is a **RESET button**, which can be used to manually reset the microcontroller.

# LEDs



**LED (Light-Emitting Diode)** is a highly efficient electronic light source. When connecting LEDs, it is necessary to place a current limiting resistor in series so that LEDs are provided with the current value specified by the manufacturer. The current varies from 0.2mA to 20mA, depending on the type of the LED and the manufacturer. The EasyMx PRO™ v7 for STM32 board uses low-current LEDs with typical current consumption of 0.2mA or 0.3mA. Board contains 67 LEDs which can be used for visual indication of the logic state on PORT pins. An active LED indicates that a logic high (1) is present on the pin. In order to enable PORT LEDs, it is necessary to enable the corresponding DIP switch on **SW15** (**Figure 6-6**).

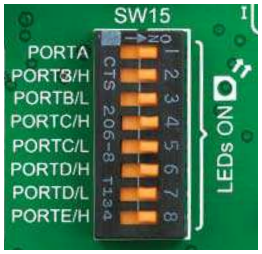


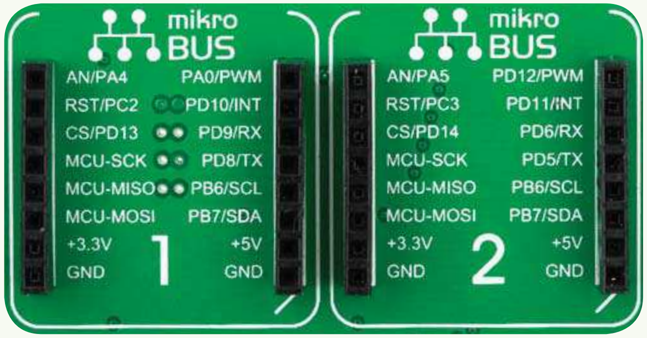
**Figure 6-6: SW15.1 through SW15.8 switches are used to enable PORT LEDs**

# mikroBUS™ sockets

Easier connectivity and simple configuration are imperative in modern electronic devices. Success of the USB standard comes from it's simplicity of usage and high and reliable data transfer rates. As we in mikroElektronika see it, Plug-and-Play devices with minimum settings are the future in embedded world too. This is why our engineers have come up with a simple, but brilliant pinout with lines that most of today's accessory boards require, which almost completely eliminates the need of additional hardware settings. We called this new standard the **mikroBUS™.** EasyMx PRO™ v7 for STM32 supports mikroBUS™ with two on-board sockets. As you can see, there are no additional DIP switches, or jumper selections. Everything is already routed to the most appropriate pins of the microcontroller sockets.

## mikroBUS™ host connector

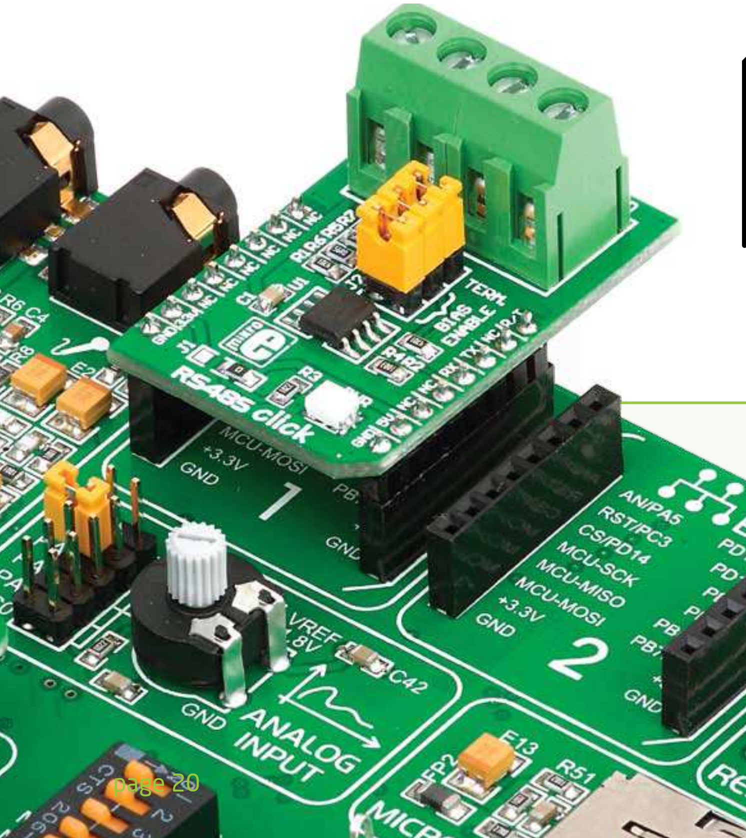
Each mikroBUS™ host connector consists of two 1x8 female headers containing pins that are most likely to be used in the target accessory board. There are three groups of communication pins: **SPI**, **UART** and **I²C** communication. There are also single pins for **PWM**, **Interrupt**, **Analog input**, **Reset** and **Chip Select**. Pinout contains two power groups: **+5V and GND** on one header and **+3.3V and GND** on the other 1x8 header.



## mikroBUS™ pinout explained

**AN** - Analog pin
**RST** - Reset pin
**CS** - SPI Chip Select line
**SCK** - SPI Clock line
**MISO** - SPI Slave Output line
**MOSI** - SPI Slave Input line
**+3.3V** - VCC-3.3V power line
**GND** - Reference Ground

**PWM** - PWM output line
**INT** - Hardware Interrupt line
**RX** - UART Receive line
**TX** - UART Transmit line
**SCL** - I²C Clock line
**SDA** - I²C Data line
**+5V** - VCC-5V power line
**GND** - Reference Ground



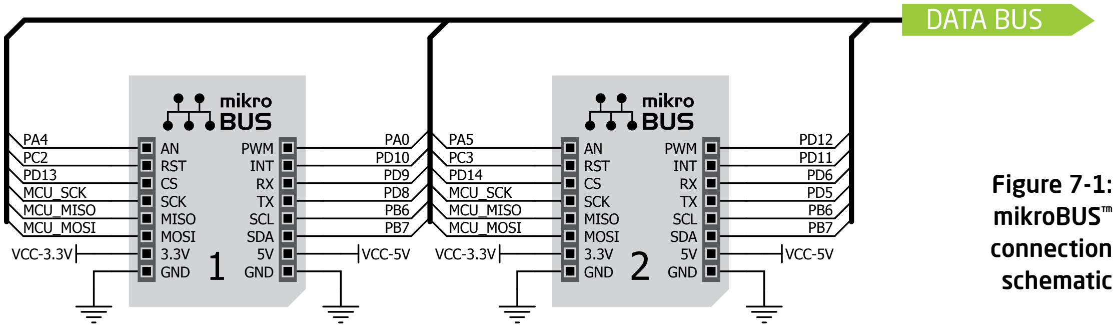
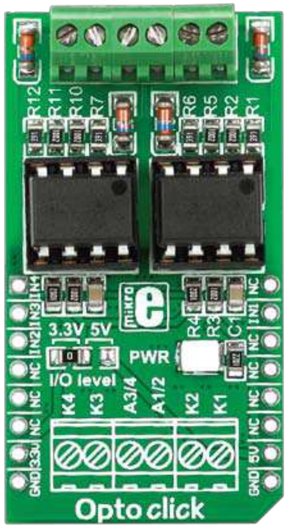


**Figure 7-1: mikroBUS™ connection schematic**

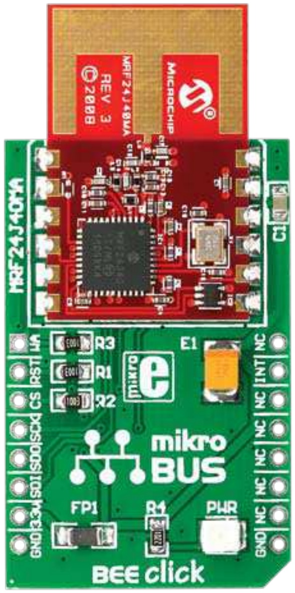
## Integrate mikroBUS™ in your design

mikroBUS™ is not made only to be a part of our development boards. You can freely place mikroBUS™ host connectors in your final PCB designs, as long as you clearly mark them with mikroBUS™ logo and footprint specifications. For more information, logo artwork and PCB files visit our website:

*http://www.mikroe.com/mikrobus*



Opto click™



BEE click™

BlueTooth click™

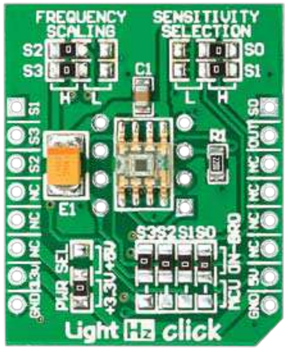
WiFi PLUS click™

GPS click™

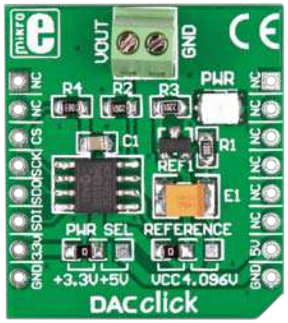
# Click Boards™ are plug-n-play!

mikroElektronika's portfolio of over 200 accessory boards is now enriched by an additional set of mikroBUS™ compatible **Click Boards™**. Almost each month several new Click boards™ are released. It is our intention to provide the community with as much of these boards as possible, so you will be able to expand your EasyMx PRO™ v7 for STM32 with additional functionality with literally zero hardware configuration. Just plug and play. Visit the Click boards™ webpage for the complete list of available boards:

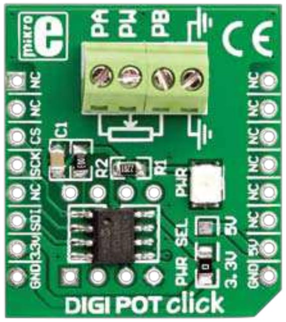
*http://www.mikroe.com/click/*



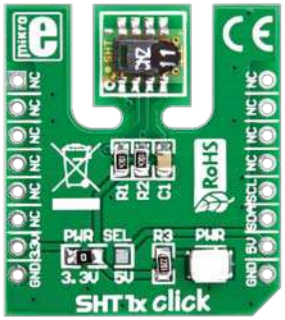
LightHz click™



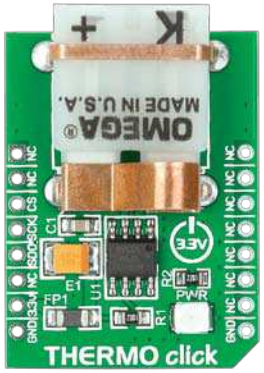
DAC click™



DIGIPOT click™

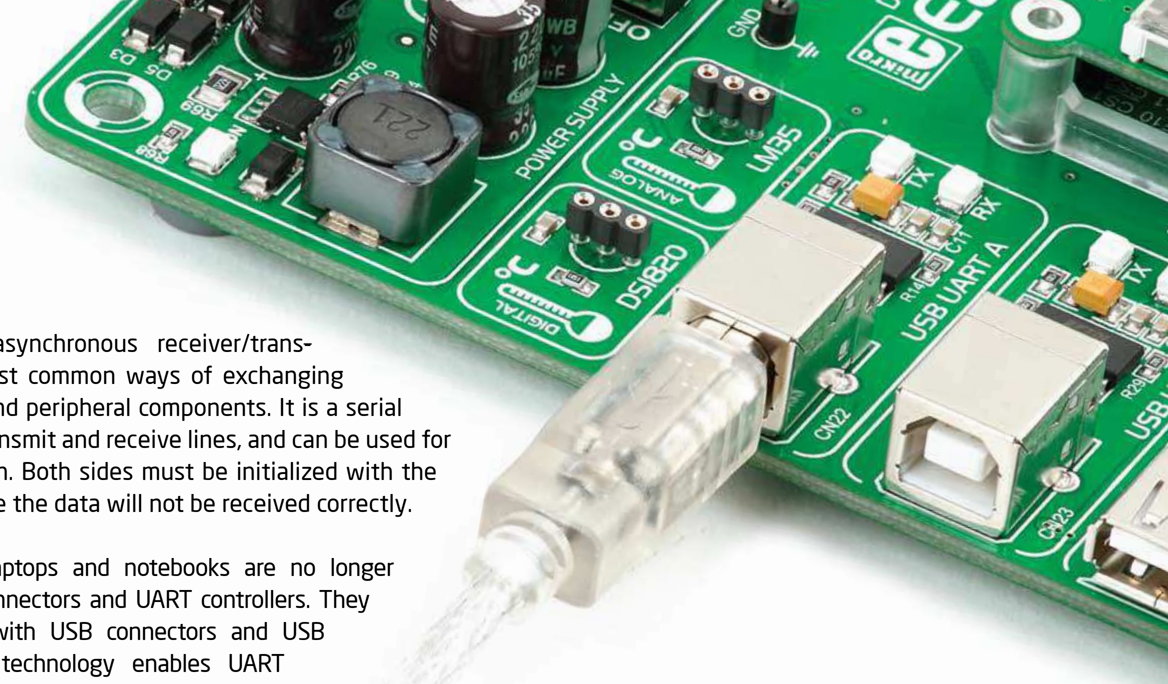


SHT1x click™

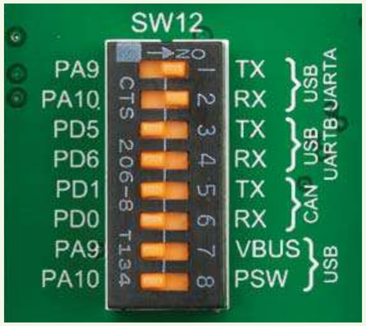


THERMO click™

# USB-UART A



## Enabling USB-UART A



In order to enable USB-UART A communication, you must push **SW12.1** (PA9) and **SW12.2** (PA10) to **ON** position. This connects the **RX** and **TX** lines to **PA9** and **PA10** microcontroller pins.

The **UART** (universal asynchronous receiver/transmitter) is one of the most common ways of exchanging data between the MCU and peripheral components. It is a serial protocol with separate transmit and receive lines, and can be used for full-duplex communication. Both sides must be initialized with the same baud rate, otherwise the data will not be received correctly.

Modern PC computers, laptops and notebooks are no longer equipped with RS-232 connectors and UART controllers. They are nowadays replaced with USB connectors and USB controllers. Still, certain technology enables UART communication to be done via USB connection. Controllers such as **FT232RL** from FTDI® convert UART signals to the appropriate USB standard.

USB-UART A communication is being done through a FT232RL controller, USB connector **(CN22)**, and microcontroller UART module. To establish this connection, you must connect **TX** and **RX** lines of the FT232RL to the appropriate pins of the microcontroller. This selection is done using DIP switches **SW12.1** and **SW12.2**.

In order to use USB-UART A module on EasyMx PRO™ v7 for STM32, you must first install FTDI drivers on your computer. Drivers can be found on **Product DVD**:

*DVD://download/eng/software/development-tools/universal/ftdi/vcp_drivers.zip*

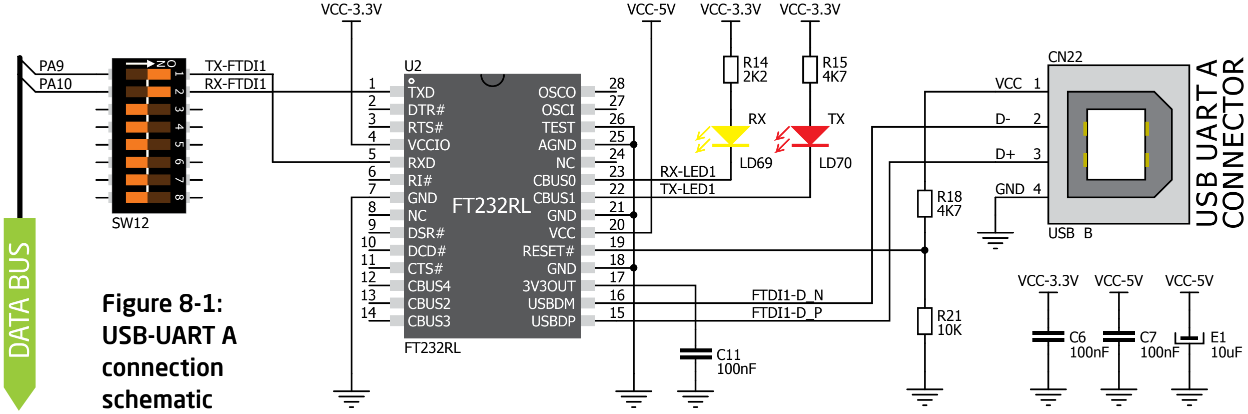
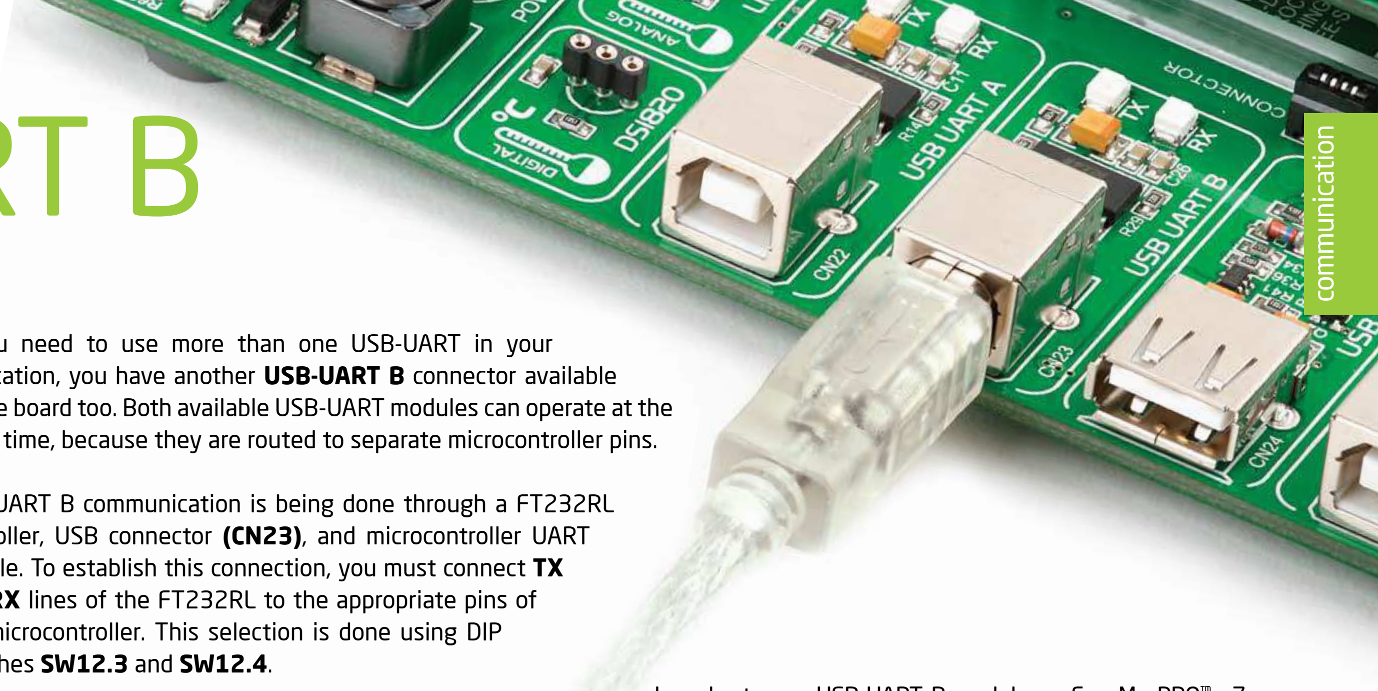
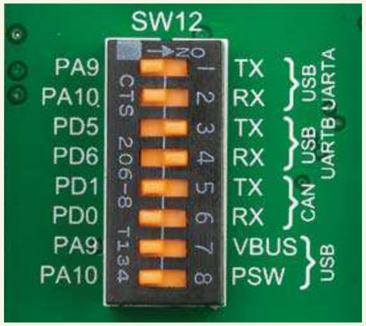


**Figure 8-1: USB-UART A connection schematic**

# USB-UART B



## Enabling USB-UART B



In order to enable USB-UART B communication, you must push **SW12.3** (PD5) and **SW12.4** (PD6) to **ON** position. This connects the **RX** and **TX** lines to **PD5** and **PD6** microcontroller pins.

If you need to use more than one USB-UART in your application, you have another **USB-UART B** connector available on the board too. Both available USB-UART modules can operate at the same time, because they are routed to separate microcontroller pins.

USB-UART B communication is being done through a FT232RL controller, USB connector **(CN23)**, and microcontroller UART module. To establish this connection, you must connect **TX** and **RX** lines of the FT232RL to the appropriate pins of the microcontroller. This selection is done using DIP switches **SW12.3** and **SW12.4**.

When using either USB-UART A or USB-UART B, make sure to disconnect all devices and additional boards that could interfere with the signals and possibly corrupt the sent or received data.

In order to use USB-UART B module on EasyMx PRO™ v7 STM32, you must first install FTDI drivers on your computer. Drivers can be found on **Product DVD**:

*DVD://download/eng/software/development-tools/universal/ftdi/vcp_drivers.zip*

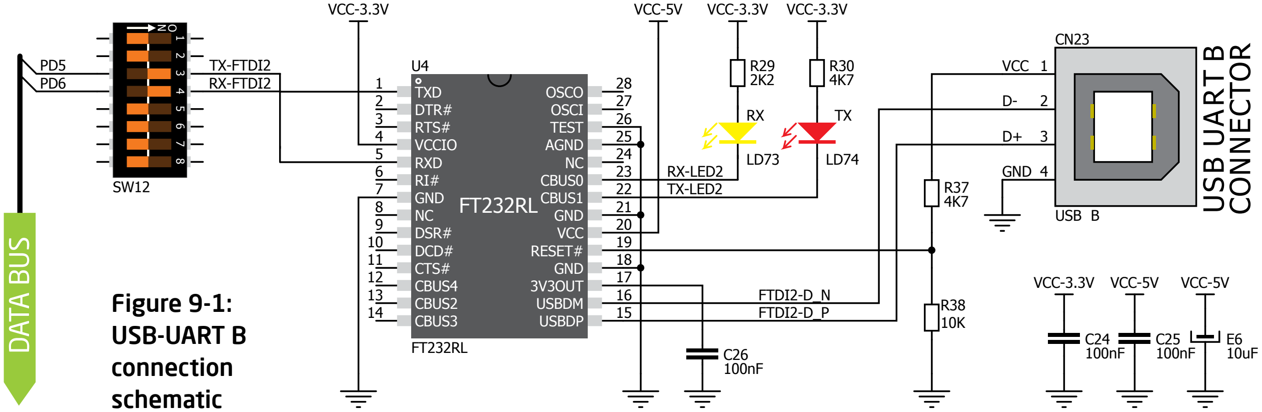
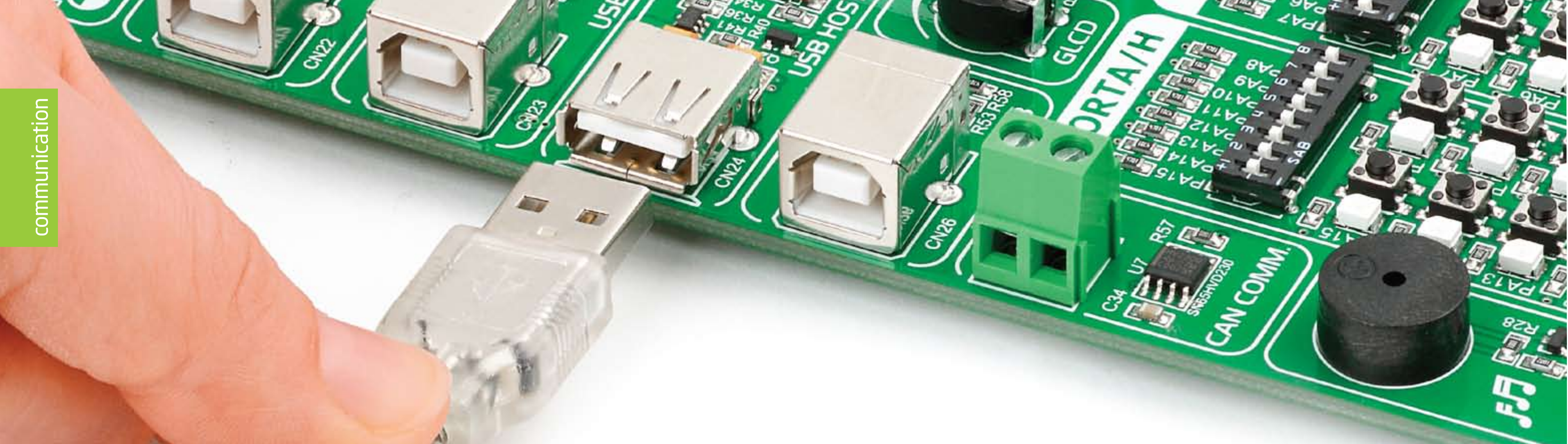


**Figure 9-1: USB-UART B connection schematic**



# USB HOST communication

USB is the acronym for **Universal Serial Bus**. This is a very popular industry standard that defines cables, connectors and protocols used for communication and power supply between computers and other devices. EasyMx PRO™ v7 for STM32 contains USB HOST connector **(CN24)** for USB Standard Type A plug, which enables microcontrollers that support USB communication to establish a connection with the target device (eg. USB Keyboard, USB Mouse, etc). USB host also provides the necessary 5V power supply to the target via **TPS2041B** IC. Detection whether USB device is connected to HOST connector can be done through **VBUS** line. Connection of USB HOST **VBUS** line and **PA10** pin is established when **SW10.7** is on.

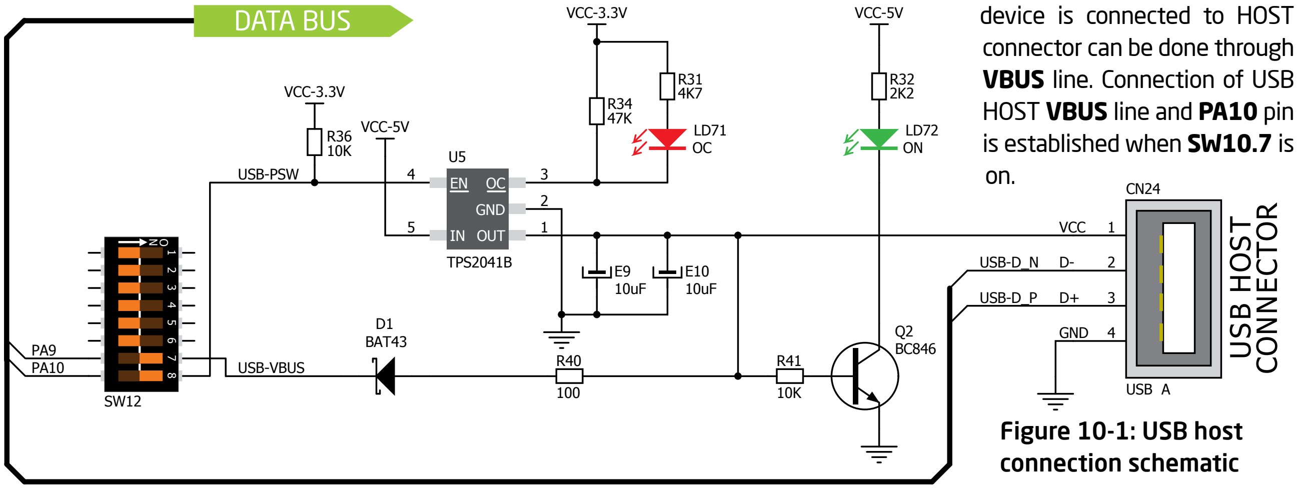


**Figure 10-1: USB host connection schematic**

## Powering USB device

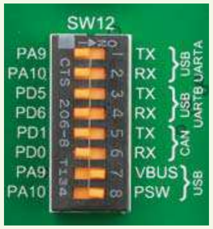


**Figure 10-2: Powering USB device through PSW line**

You can enable or disable power supply to USB device connected to HOST, through microcontroller **PA10** pin. In order to connect EN TPS2041B IC pin to microcontroller, you must push **SW10.8** to **ON** position.

# USB device communication

EasyMx PRO™ v7 for STM32 also contains USB DEVICE connector **(CN26)** which enables microcontrollers that support USB communication to establish a connection with the target host (eg. PC, Laptop, etc). It lets you build a slave USB device (HID, Composite, Generic, etc.). Connector supports USB Standard Type B plug. Detection whether USB device is connected to HOST can be done through **VBUS** line. This line is traced to microcontroller **PA9** pin. Connection of USB DEVICE **VCC line** and **PA9** pin is established when **SW12.7** DIP switch is in ON position. When connected to HOST, dedicated amber-colored power LED will light up as well. This VCC line cannot be used for powering the board. It's only used for detecting connection.

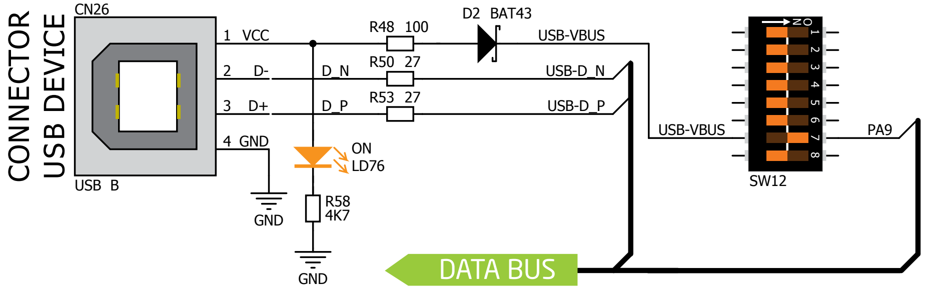


**Figure 11-1: USB device connection schematic**

## Detecting connection

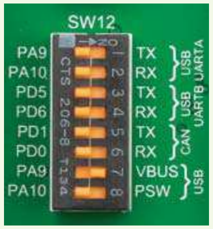


**Figure 11-2: Enabling USB DEVICE detection via VBUS line**

You can detect whether USB device is plugged into the USB device connector using **VBUS** power detection line **(PA9)**. Before using this feature, you must connect **PA9** pin to USB connector using **SW12.7** DIP switch.