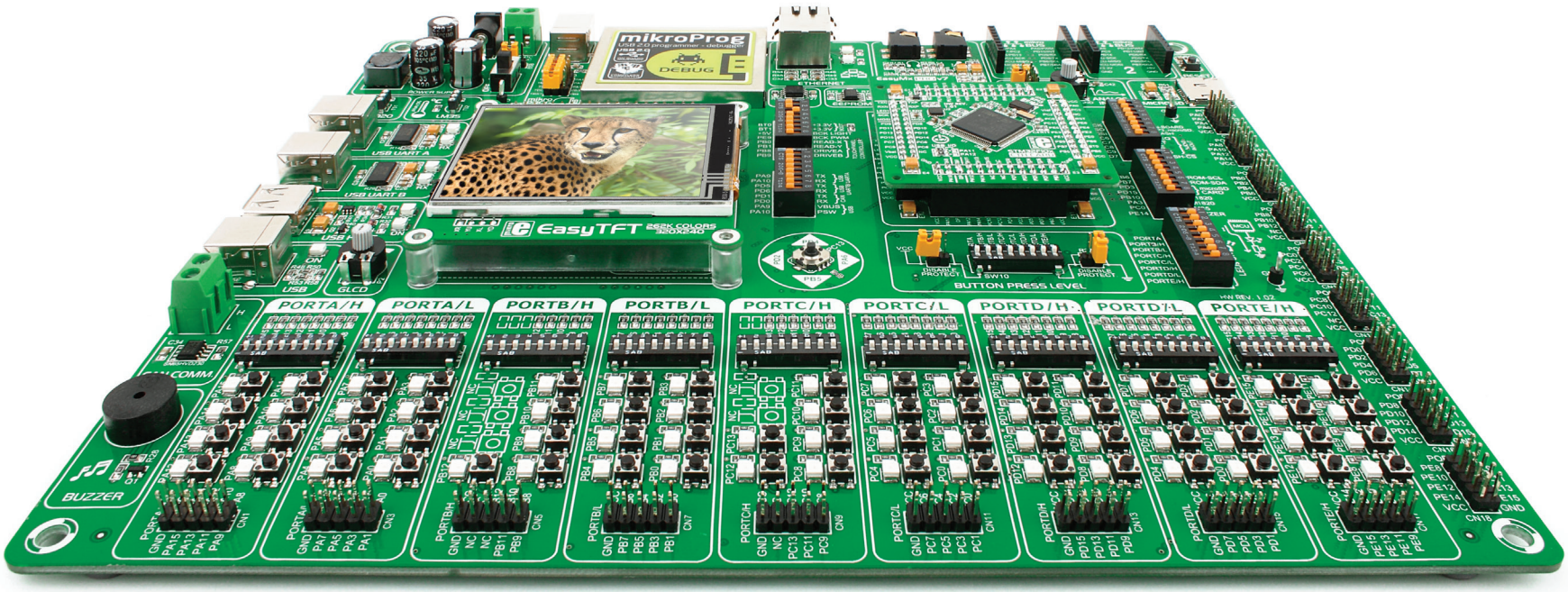
EasyMx PRO™ v7
for STM32 ARM®

USER'S GUIDE



186

microcontrollers supported
The ultimate STM32 board

Many on-board modules
Multimedia peripherals

Easy-add extra boards
mikroBUS™ sockets

Two connectors for each port
Amazing Connectivity

Fast USB 2.0 programmer and
In-Circuit Debugger

MikroElektronika
DEVELOPMENT TOOLS | COMPILERS | BOOKS

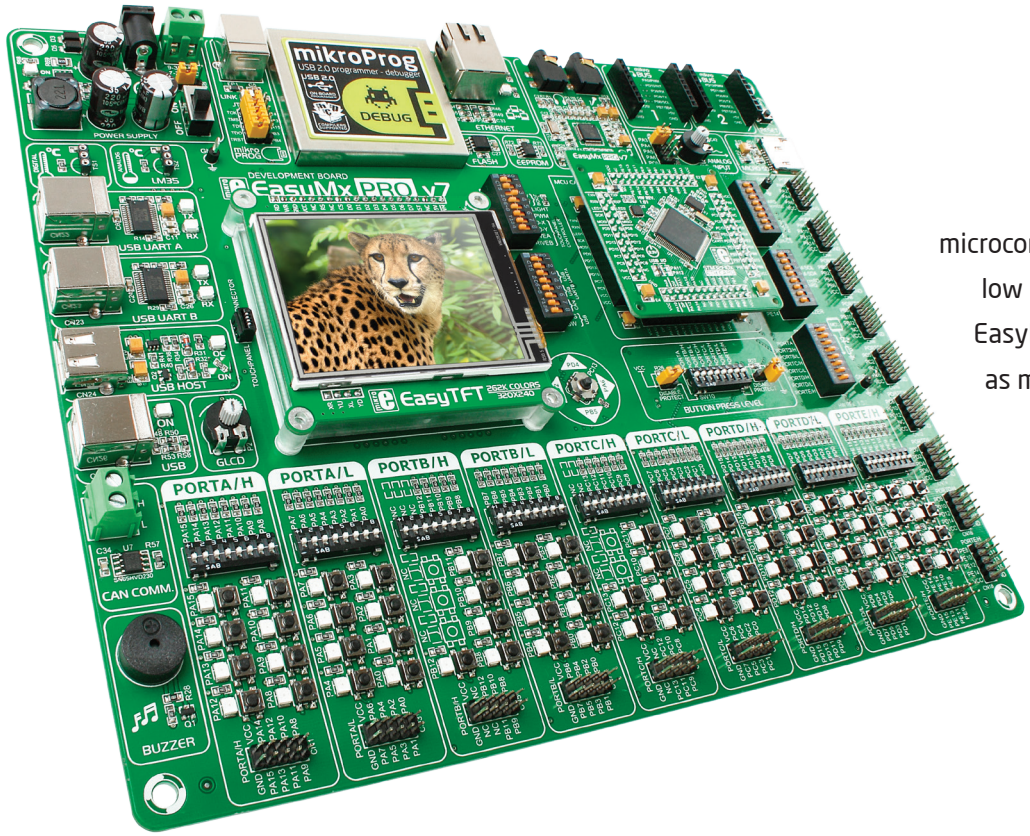
To our valued customers

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

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

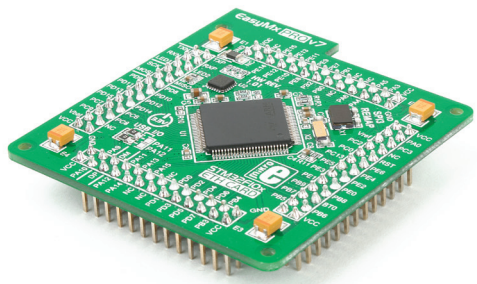
Just plug in your Click™ board, and it's ready to work. We picked up a set of the most useful pins you need for development and made a pinout standard you will enjoy using.

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)

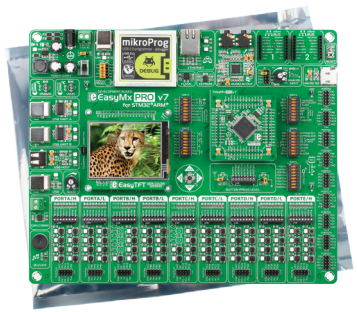
power consumption
~76mA when all peripheral modules are disconnected

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

weight
~500g (1.1 lbs)

Package contains

1 Damage resistant protective box

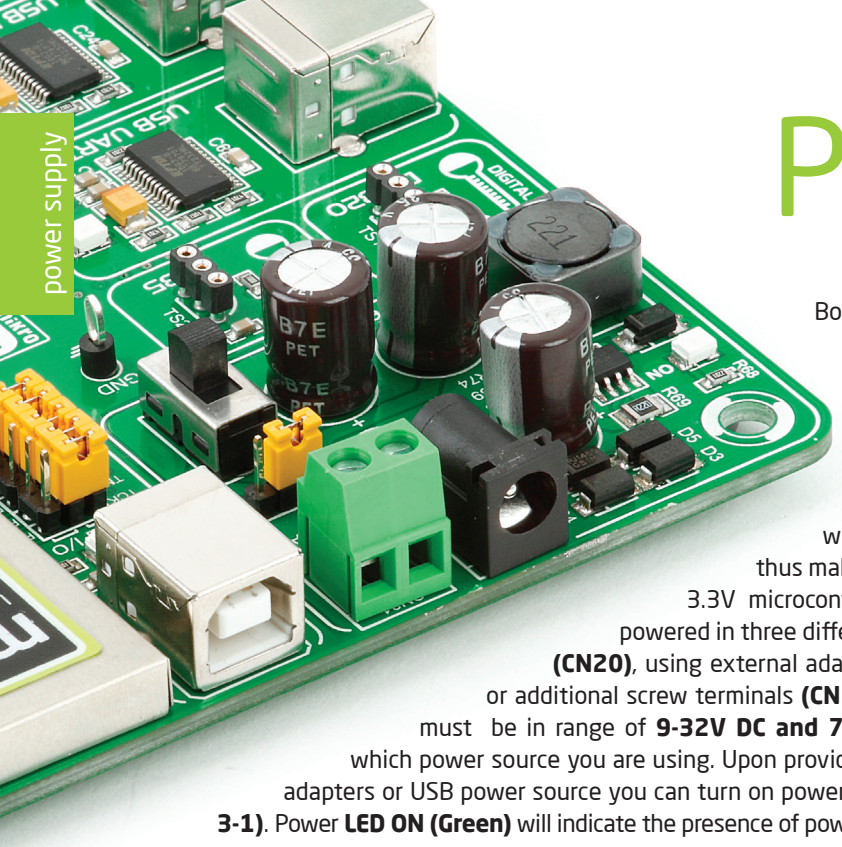


2 EasyMx PRO™ v7 for STM32 board in antistatic bag

3 USB cable

4 User Manuals and Board schematics

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.

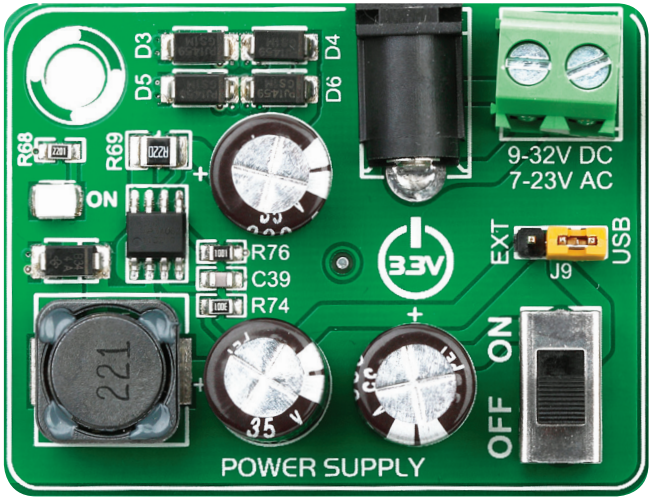
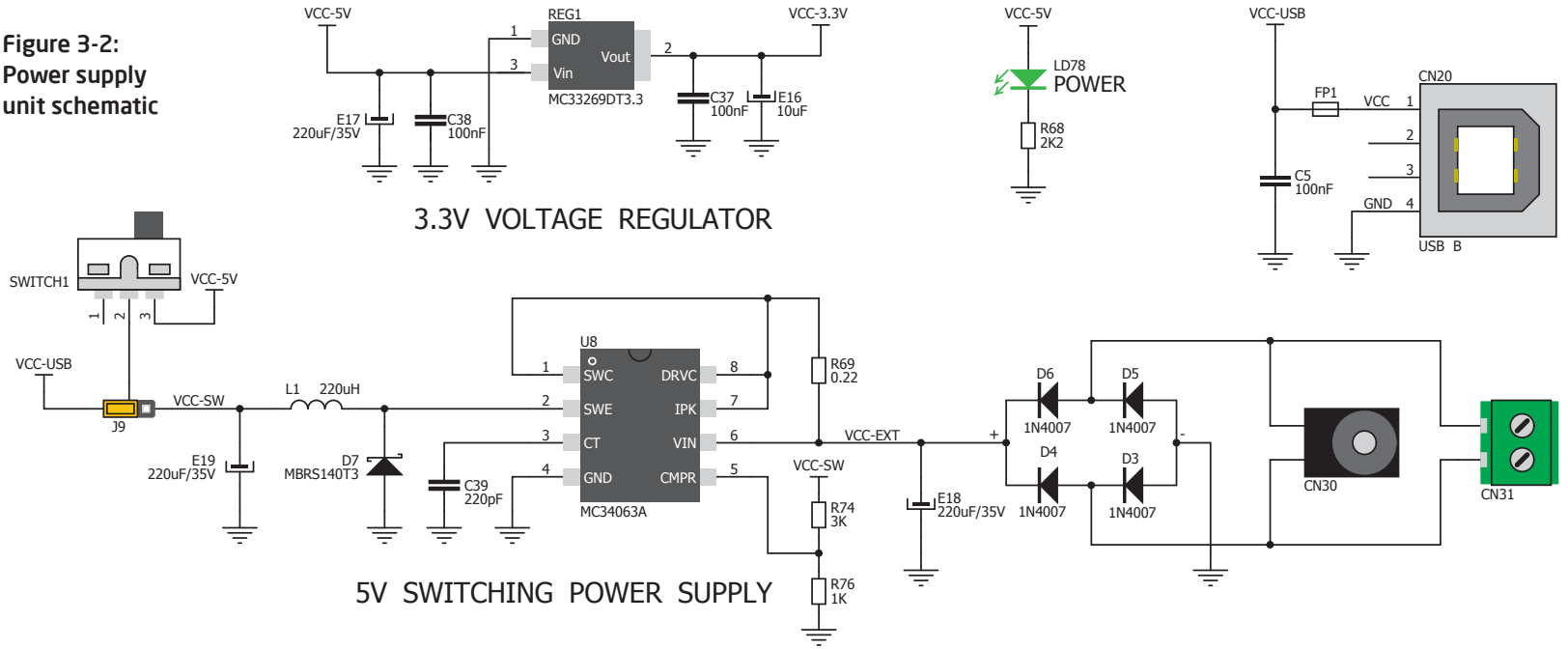


Figure 3-1: Power supply unit of EasyMx PRO™ v7 STM32

Figure 3-2: Power supply unit schematic



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)

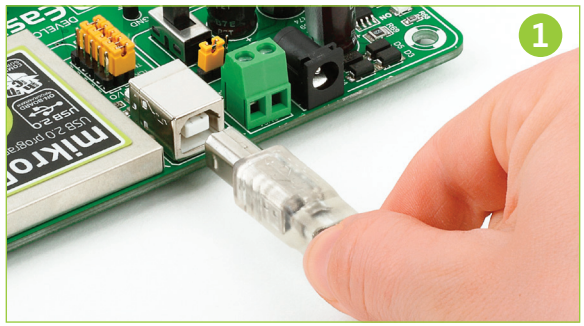
Power capacity: up to 500mA with USB, and up to 600mA with external power supply

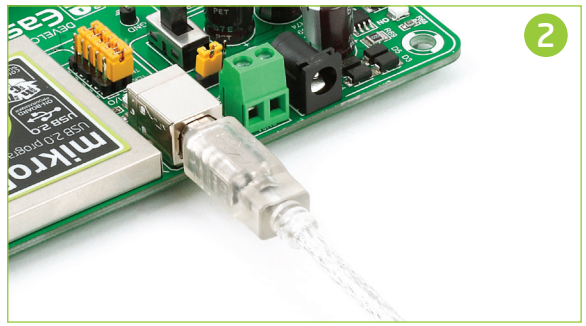
How to power the board?

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.

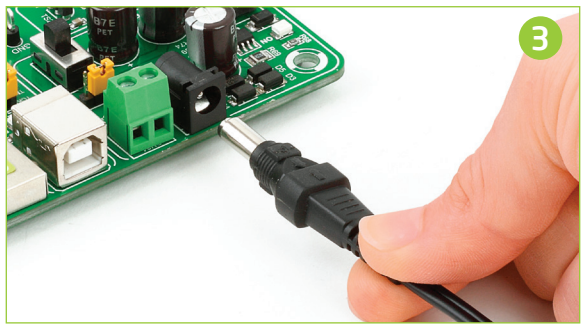


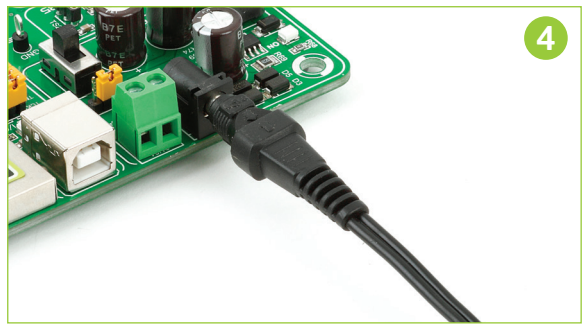


2. Using adapter

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.

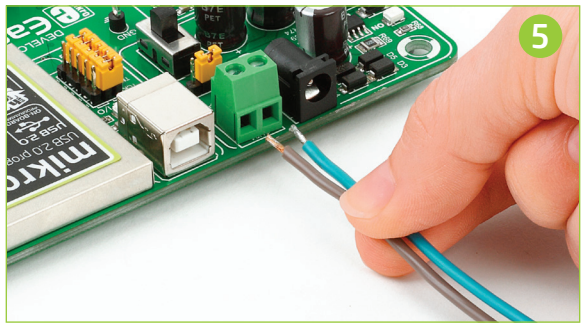


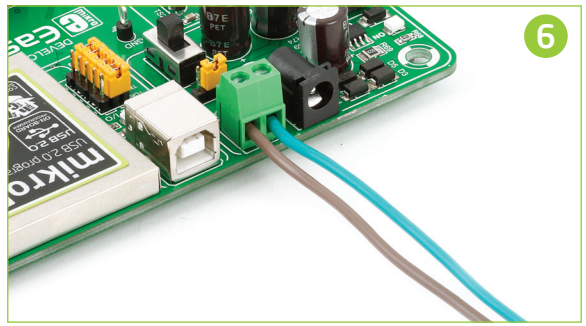


3. With laboratory power supply

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.

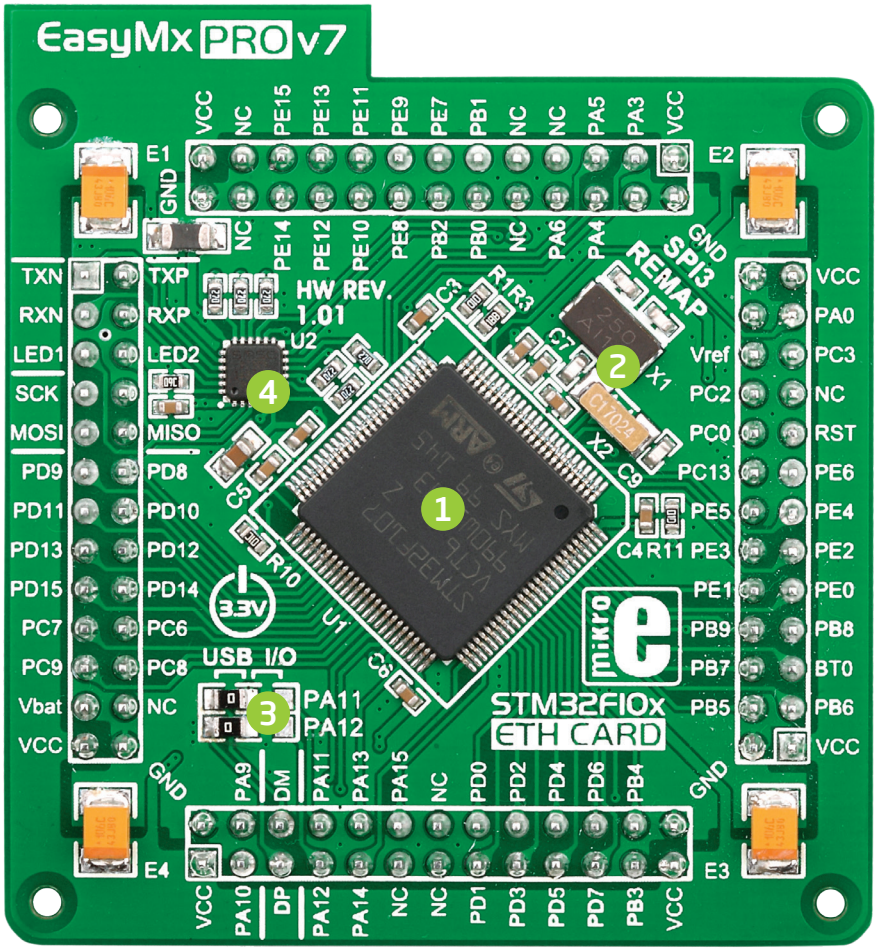


Figure 4-1: Default MCU card with STM32F107VCT6

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.

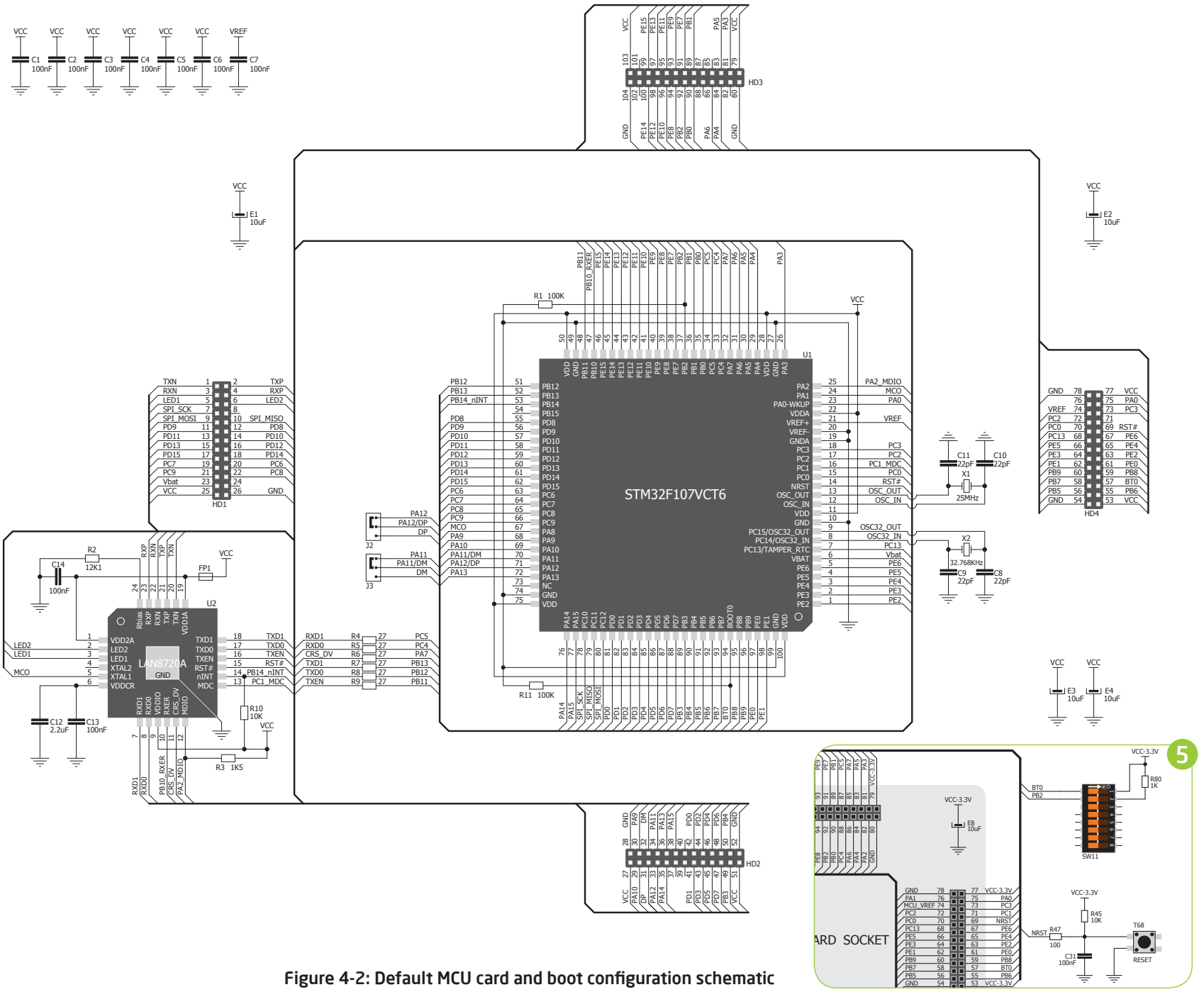


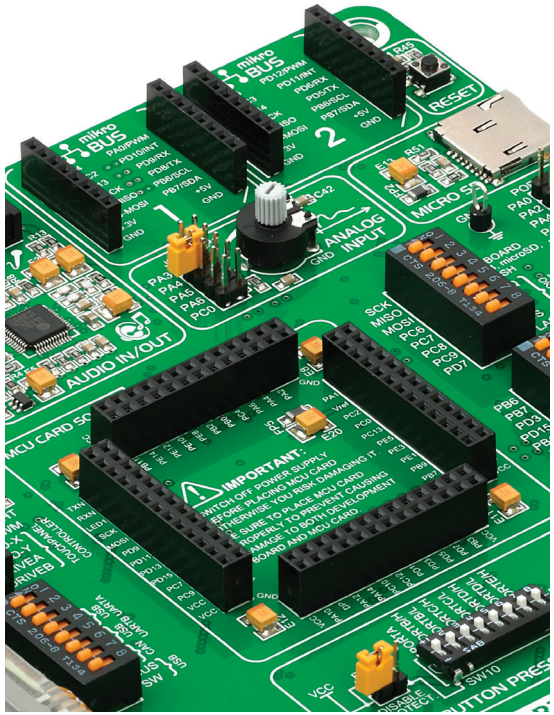
Figure 4-2: Default MCU card and boot configuration schematic

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.

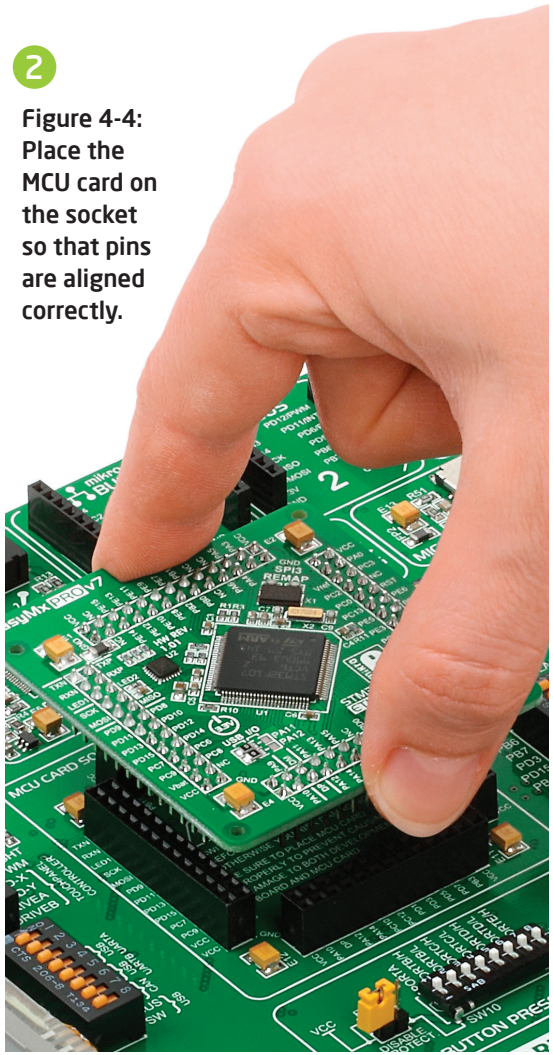
1

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



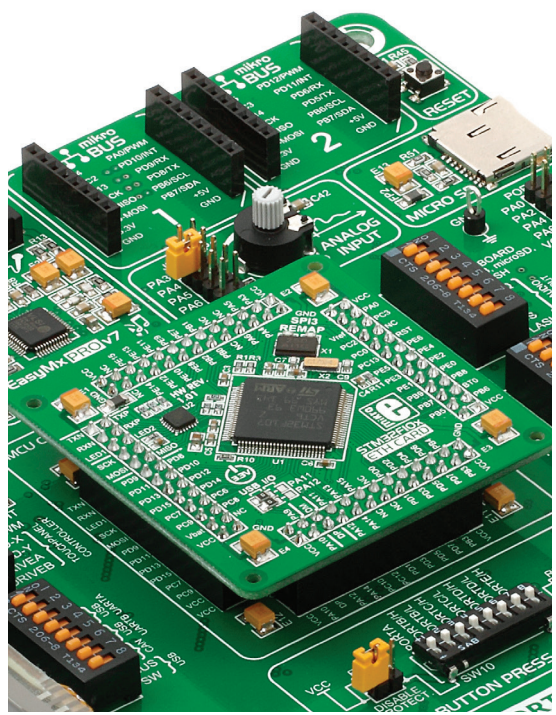
2

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



3

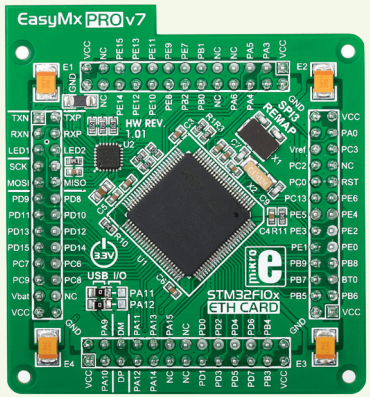
Figure 4-5 Properly placed MCU card.



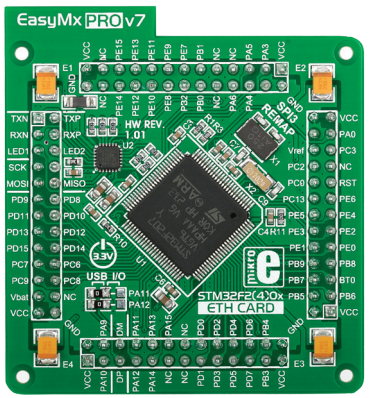
Other supported MCU cards

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:

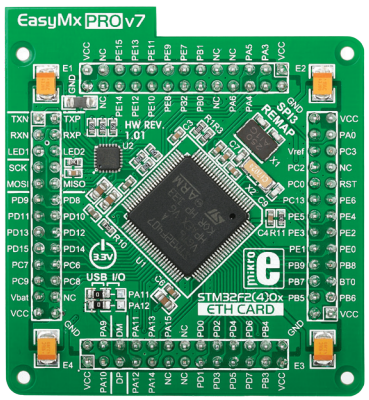
http://www.mikroe.com/easymx-pro/stm32/



Default 100-pin ETH MCU card with STM32F107VCT6



Standard 100-pin HP ETH MCU card with STM32F207VGT6

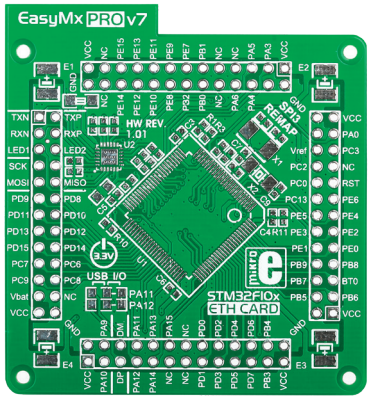


Standard 100-pin HP ETH MCU card with STM32F407VGT6

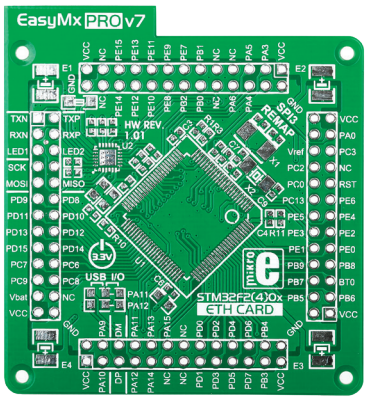
NOTE:

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

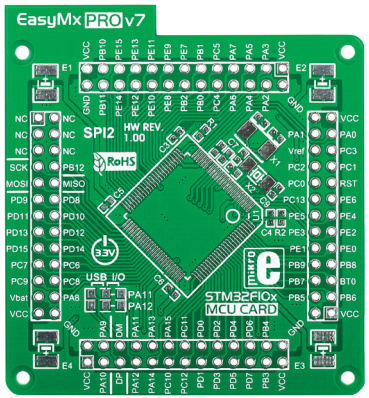
"ETH" (Ethernet) - Empty MCU cards with single-chip Ethernet PHY layer transceiver which provides additional Ethernet functionality to microcontrollers



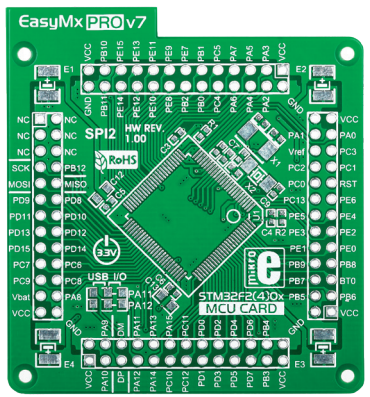
Empty ETH MCU card for 100-pin STM32F10x MCUs



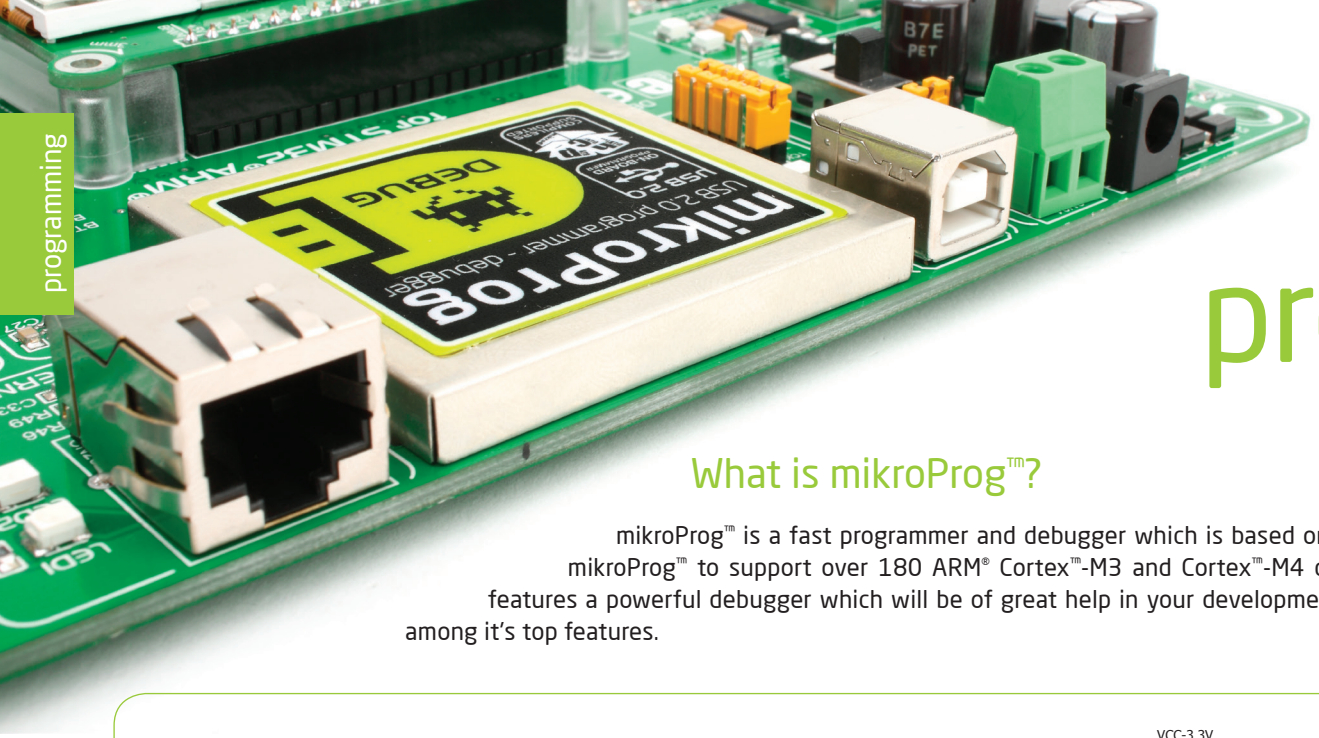
Empty ETH HP MCU card for 100-pin STM32F2(4)0x MCUs



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.

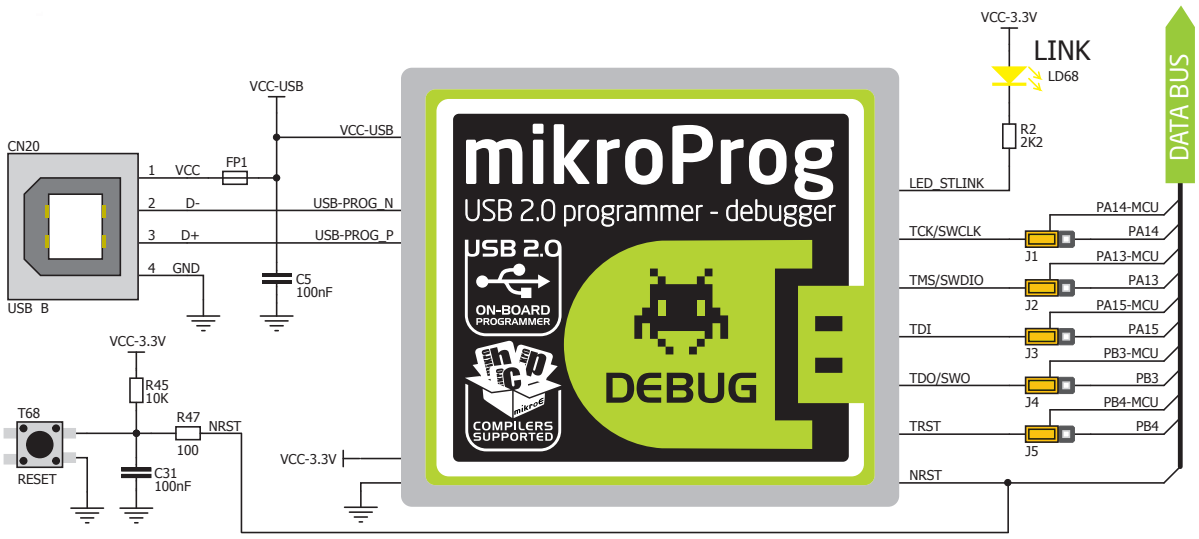
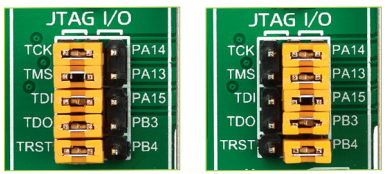


Figure 5-1: mikroProg™ block schematics

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	STM32F101R6	STM32F102C6	STM32F103V8	STM32F205RE	STM32F215RE	STM32L151VC
STM32F100C6	STM32F101R8	STM32F102C8	STM32F103VB	STM32F205RF	STM32F215RG	STM32L151VD
STM32F100C8	STM32F101RB	STM32F102CB	STM32F103VC	STM32F205RG	STM32F215VE	STM32L151ZC
STM32F100CB	STM32F101RC	STM32F102R4	STM32F103VD	STM32F205VB	STM32F215VG	STM32L151ZD
STM32F100R4	STM32F101RD	STM32F102R6	STM32F103VE	STM32F205VC	STM32F215ZE	STM32L152C6
STM32F100R6	STM32F101RE	STM32F102R8	STM32F103VF	STM32F205VE	STM32F215ZG	STM32L152C8
STM32F100R8	STM32F101RF	STM32F102RB	STM32F103VG	STM32F205VF	STM32F217IE	STM32L152CB
STM32F100RB	STM32F101RG	STM32F103C4	STM32F103ZC	STM32F205VG	STM32F217IG	STM32L152QC
STM32F100RC	STM32F101T4	STM32F103C6	STM32F103ZD	STM32F205ZC	STM32F217VE	STM32L152QD
STM32F100RD	STM32F101T6	STM32F103C8	STM32F103ZE	STM32F205ZE	STM32F217VG	STM32L152R6
STM32F100RE	STM32F101T8	STM32F103CB	STM32F103ZF	STM32F205ZF	STM32F217ZE	STM32L152R8
STM32F100V8	STM32F101TB	STM32F103R4	STM32F103ZG	STM32F205ZG	STM32F217ZG	STM32L152RB
STM32F100VB	STM32F101V8	STM32F103R6	STM32F105R8	STM32F207IC	STM32L151C6	STM32L152RC
STM32F100VC	STM32F101VB	STM32F103R8	STM32F105RB	STM32F207IE	STM32L151C8	STM32L152RD
STM32F100VD	STM32F101VC	STM32F103RB	STM32F105RC	STM32F207IF	STM32L151CB	STM32L152V8
STM32F100VE	STM32F101VD	STM32F103RC	STM32F105V8	STM32F207IG	STM32L151QC	STM32L152VB
STM32F100ZC	STM32F101VE	STM32F103RD	STM32F105VB	STM32F207VC	STM32L151QD	STM32L152VC
STM32F100ZD	STM32F101VF	STM32F103RE	STM32F105VC	STM32F207VE	STM32L151R6	STM32L152VD
STM32F100ZE	STM32F101VG	STM32F103RF	STM32F107RB	STM32F207VF	STM32L151R8	STM32L152ZC
STM32F101C4	STM32F101ZC	STM32F103RG	STM32F107RC	STM32F207VG	STM32L151RB	STM32L152ZD
STM32F101C6	STM32F101ZD	STM32F103T4	STM32F107VB	STM32F207ZC	STM32L151RC	STM32L162QD
STM32F101C8	STM32F101ZE	STM32F103T6	STM32F107VC	STM32F207ZE	STM32L151RD	STM32L162RD
STM32F101CB	STM32F101ZG	STM32F103T8	STM32F205RB	STM32F207ZF	STM32L151V8	STM32L162VD
STM32F101R4	STM32F102C4	STM32F103TB	STM32F205RC	STM32F207ZG	STM32L151VB	STM32L162ZD

STM32 Cortex™-M4 microcontrollers supported with mikroProg™

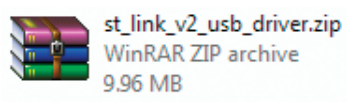
STM32F405RG	STM32F407IE	STM32F407VG	STM32F415RG	STM32F417IE	STM32F417VG
STM32F405VG	STM32F407IG	STM32F407ZE	STM32F415VG	STM32F417IG	STM32F417ZE
STM32F405ZG	STM32F407VE	STM32F407ZG	STM32F415ZG	STM32F417VE	STM32F417ZG

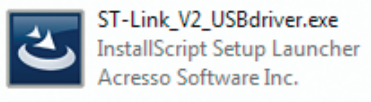
Installing programmer drivers

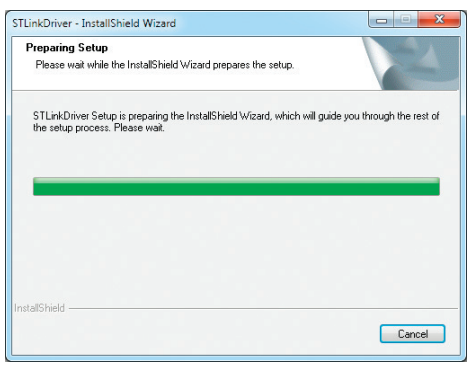
On-board mikroProg™ requires drivers in order to work. Drivers are located on the link below:

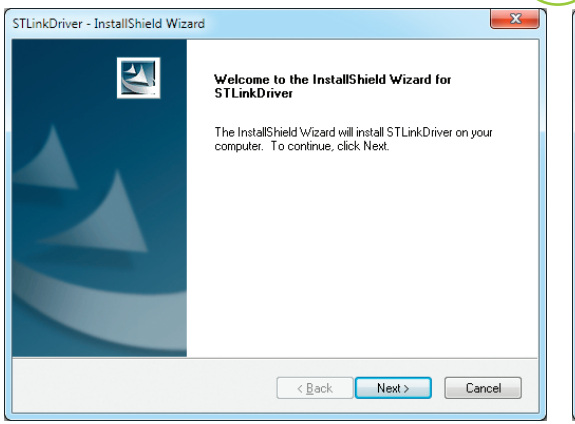
http://www.mikroe.com/downloads/get/1838/st_link_v2_usb_driver.zip

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.



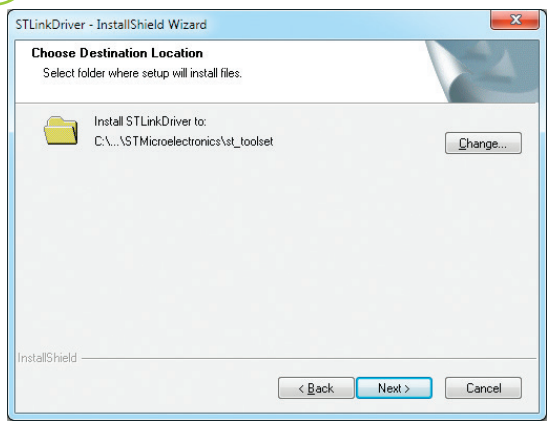






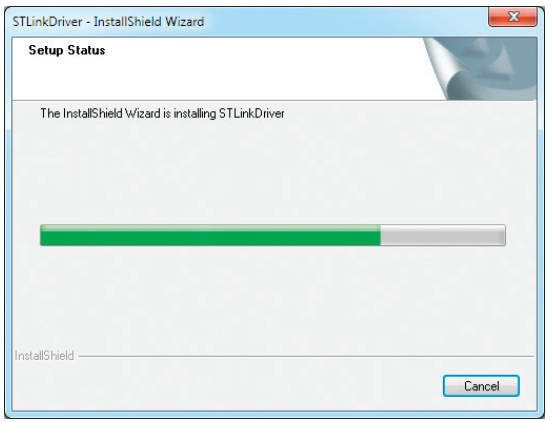
Step 1 - Start Installation

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



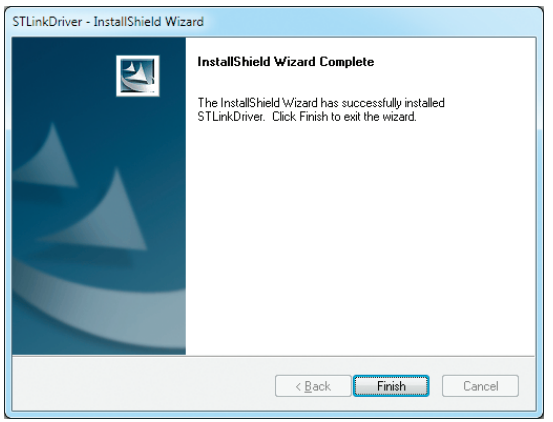
Step 2 - Select Destination

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



Step 3 - Installing drivers

Drivers are installed automatically in a matter of seconds.



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 our web site:

http://www.mikroe.com/downloads/get/1809/mikroprog_suite_for_arm.zip

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

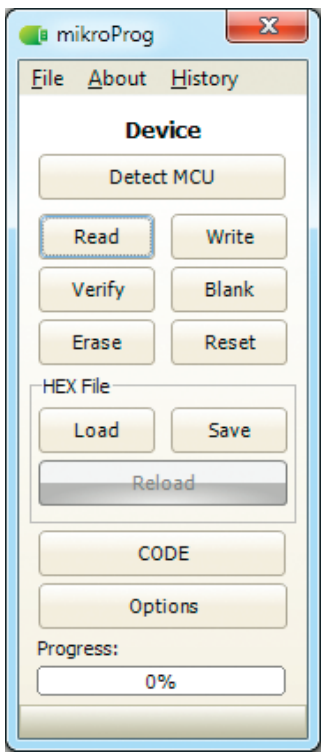


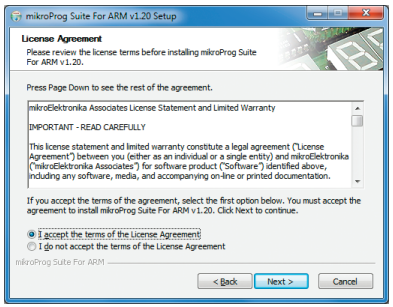
Figure 5-2: mikroProg Suite™ for ARM® window

Quick Guide

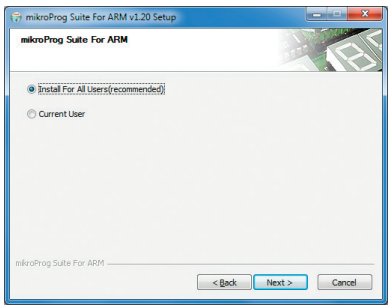
1. Click the **Detect MCU** button in order to recognize the device ID.
2. Click the **Read** button to read the entire microcontroller memory. You can click the **Save** button to save it to target HEX file.
3. If you want to write the HEX file to the microcontroller, first make sure to load the target 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

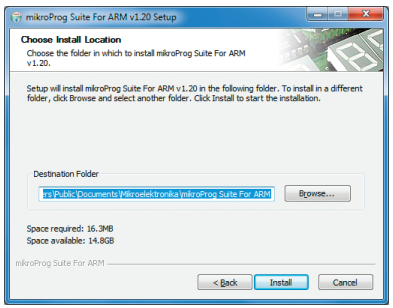
Step 1 - Start Installation



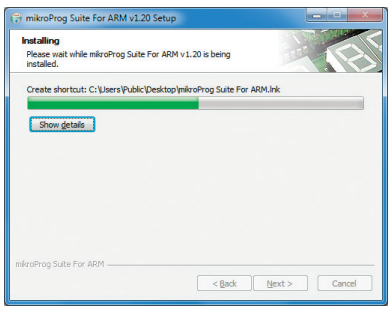
Step 2 - Accept EULA and continue



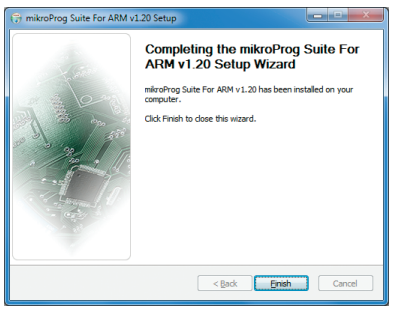
Step 3 - Install for All users or current user



Step 4 - Choose destination folder



Step 5 - Installation in progress



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.

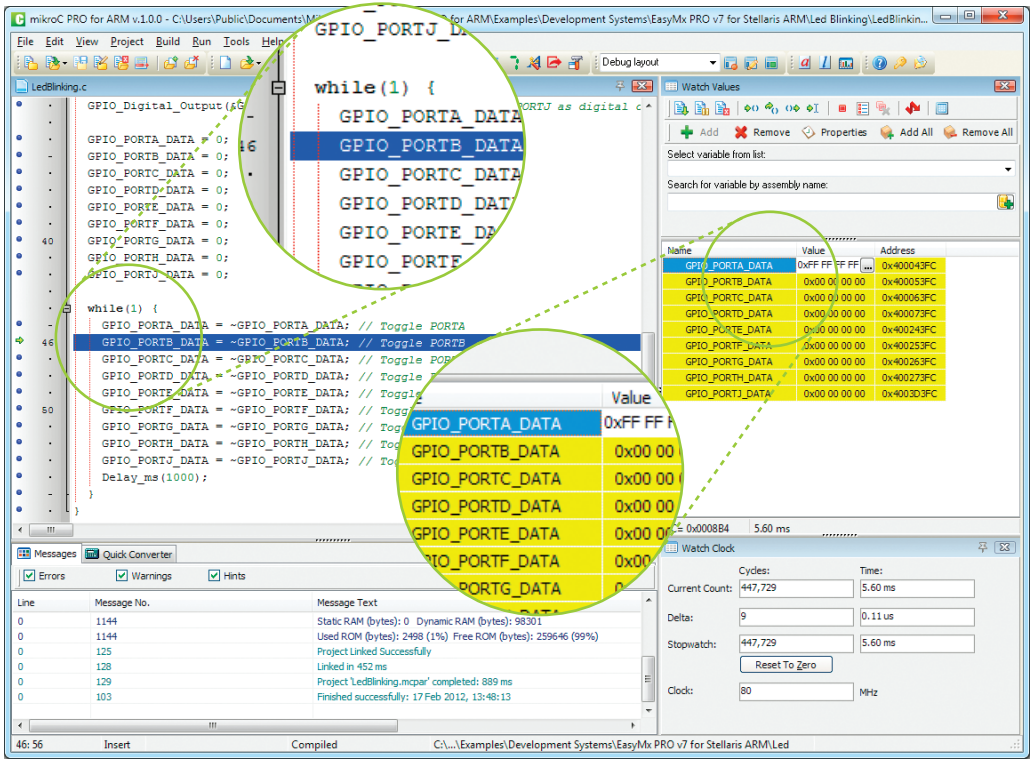


Figure 5-3: mikroC PRO for ARM® compiler in debugging view, with SFR registers in Watch Window

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.
	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.

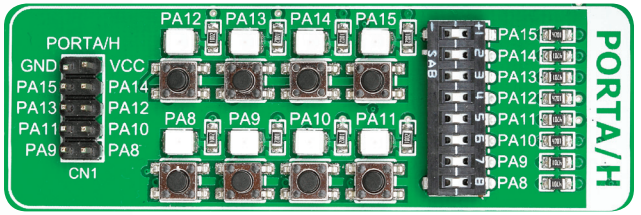


Figure 6-1: I/O group contains PORT header, tri-state pull up/down DIP switch, buttons and LEDs all in one place

Everything is grouped together

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

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.

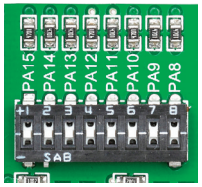
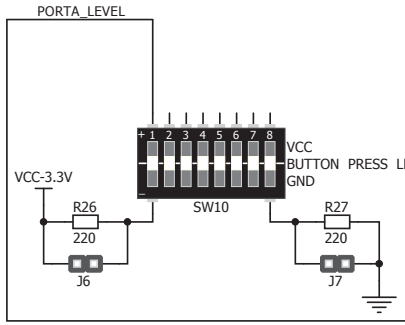


Figure 6-2: Tri-state DIP switch on PORTA/H



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

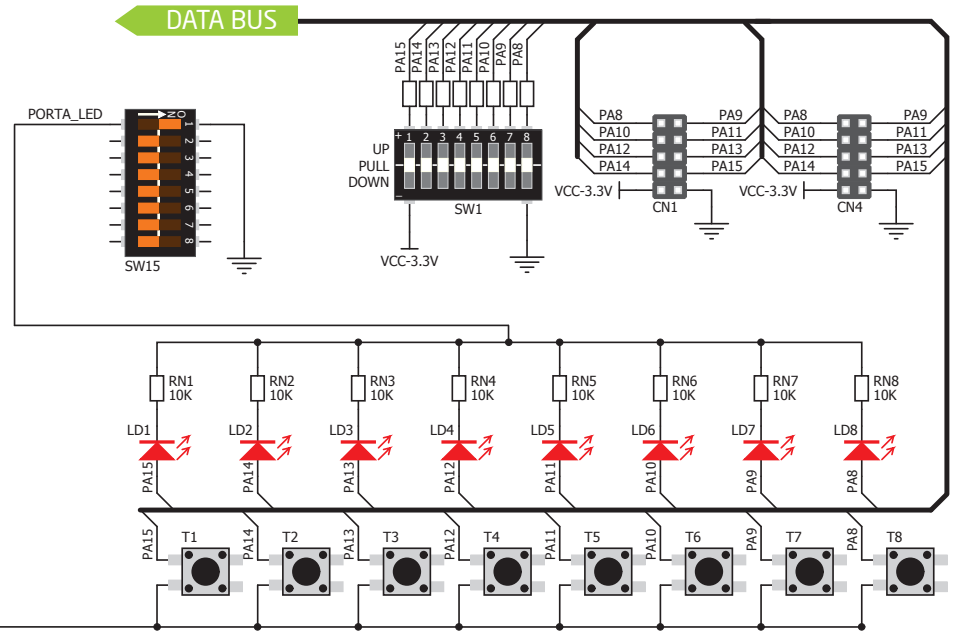
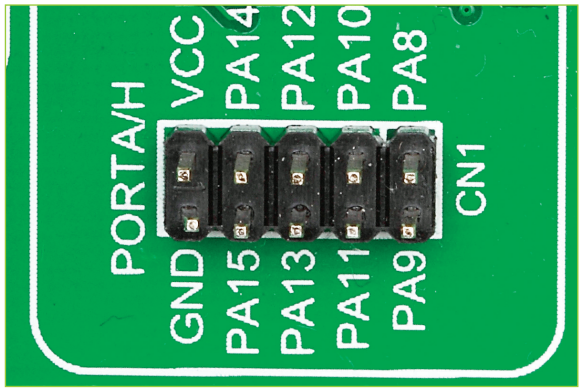
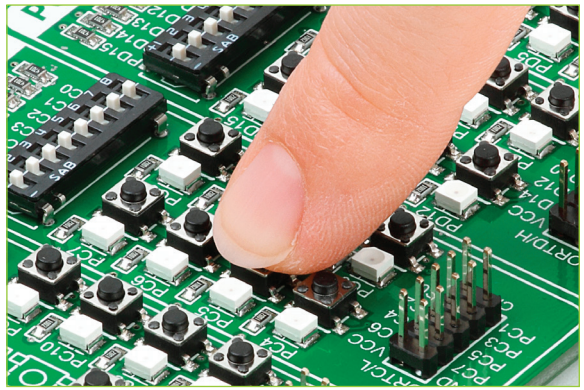
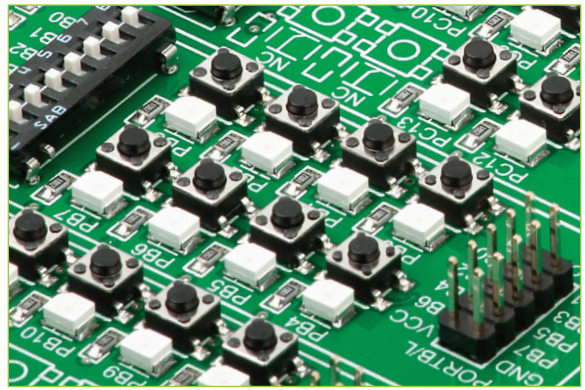


Figure 6-3: Schematic of the single I/O group connected to microcontroller PORTA/H







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.

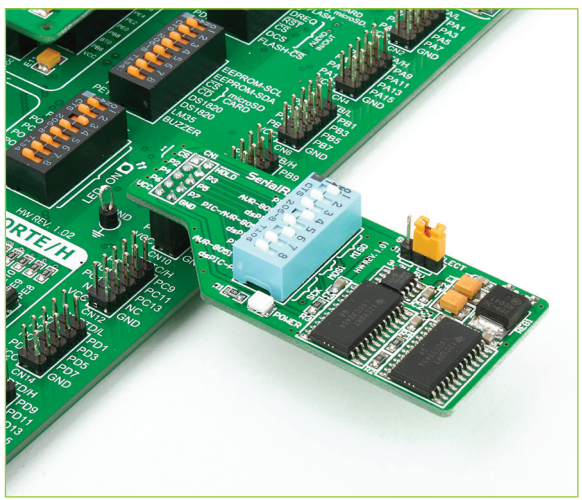


Figure 6-4: IDC10 male headers enable easy connection with MikroElektronika accessory boards

Buttons

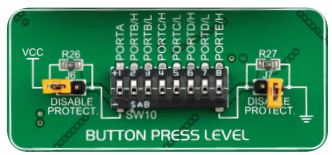
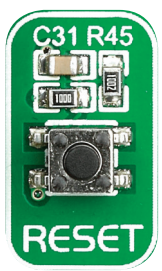


Figure 6-5: Button press level DIP switch (tri-state)

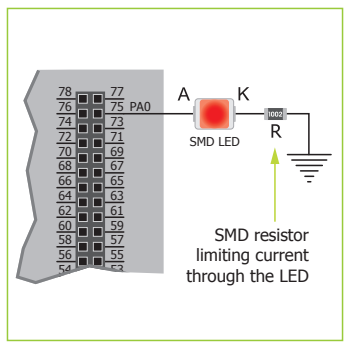
The logic state of all microcontroller digital inputs may be changed using **push buttons**. Tri-state DIP switch **SW10** is available for selecting which logic state will be applied to corresponding MCU pin when button is pressed, for each I/O port separately. If you, for example, place **SW10.1** 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**).

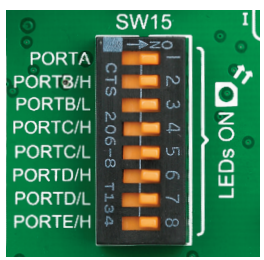


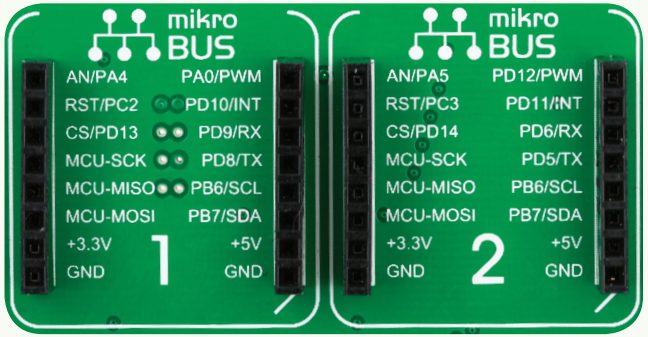
Figure 6-6: SW15.1 through SW15.8 switches are used to enable PORT LEDs

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.

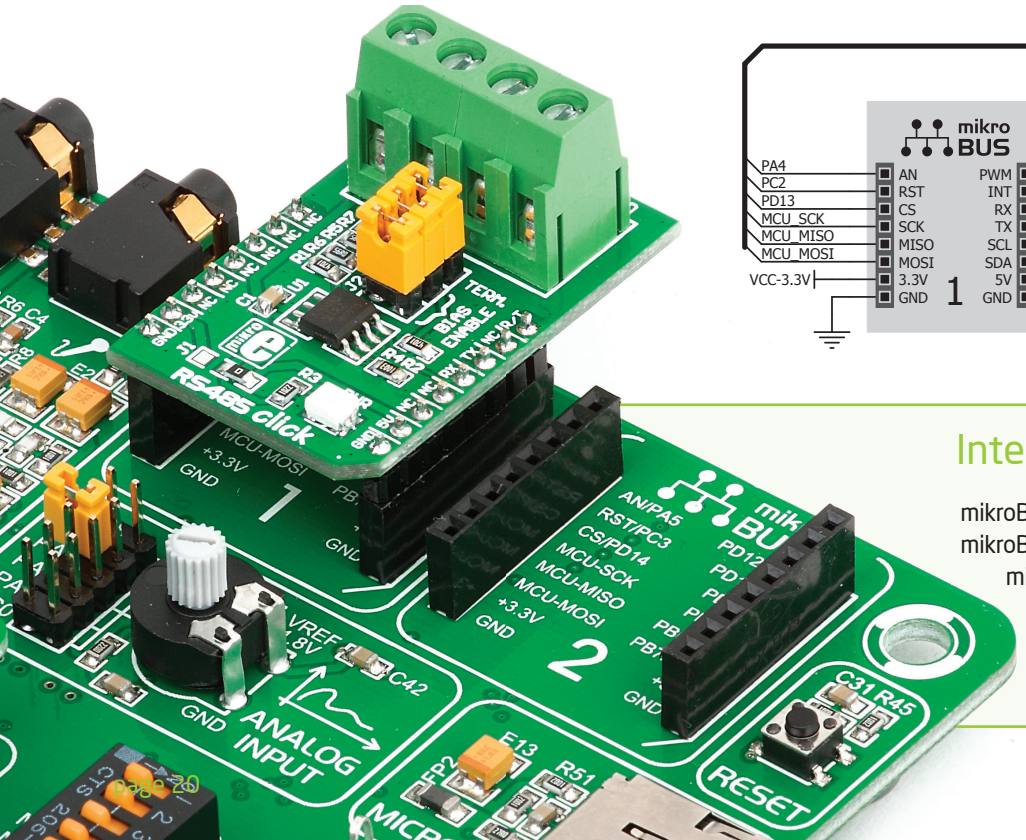
mikroBUS™ host connector

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



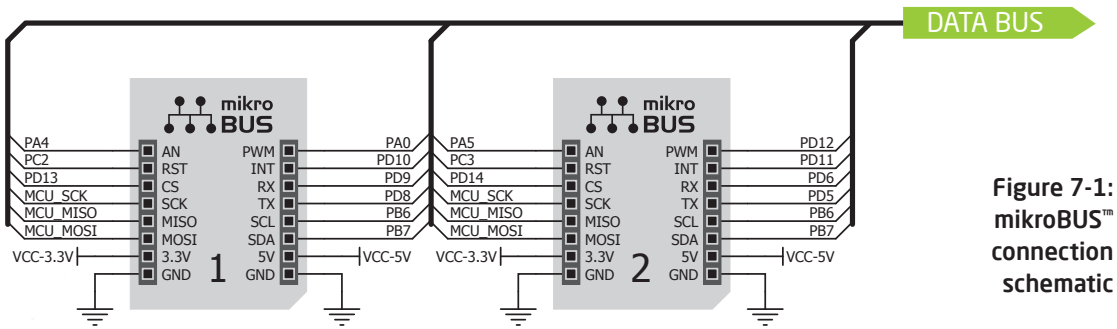
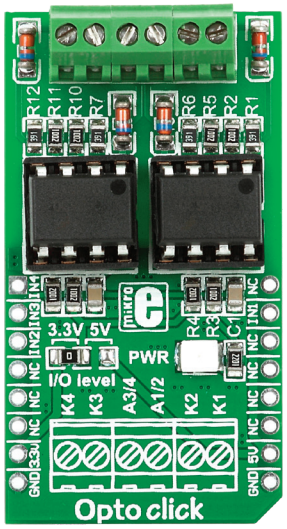


Figure 7-1: mikroBUS™ connection schematic

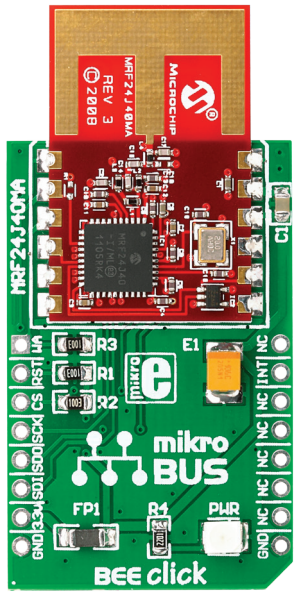
Integrate mikroBUS™ in your design

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/



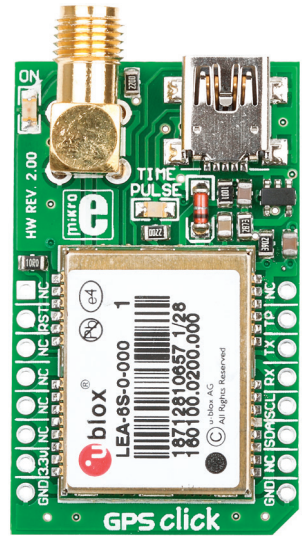
Opto click™



BEE click™

BlueTooth click™

WiFi PLUS click™

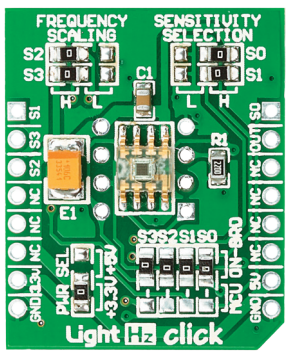


GPS click™

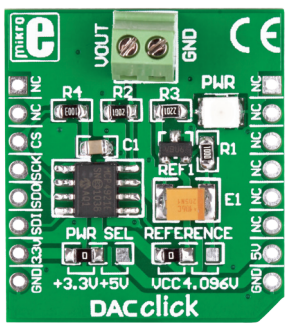
click Boards™ are plug-n-play!

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:

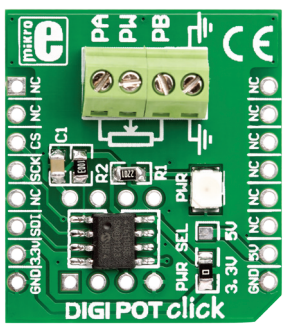
http://www.mikroe.com/click/



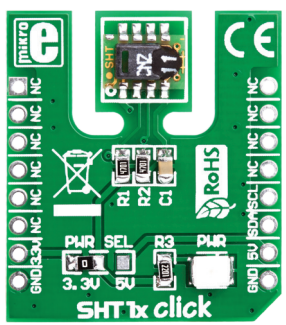
LightHz click™



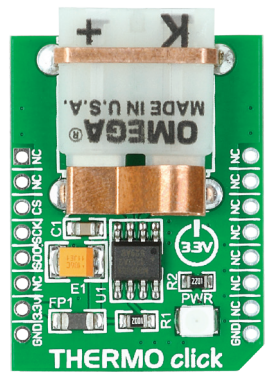
DAC click™



DIGIPOT click™

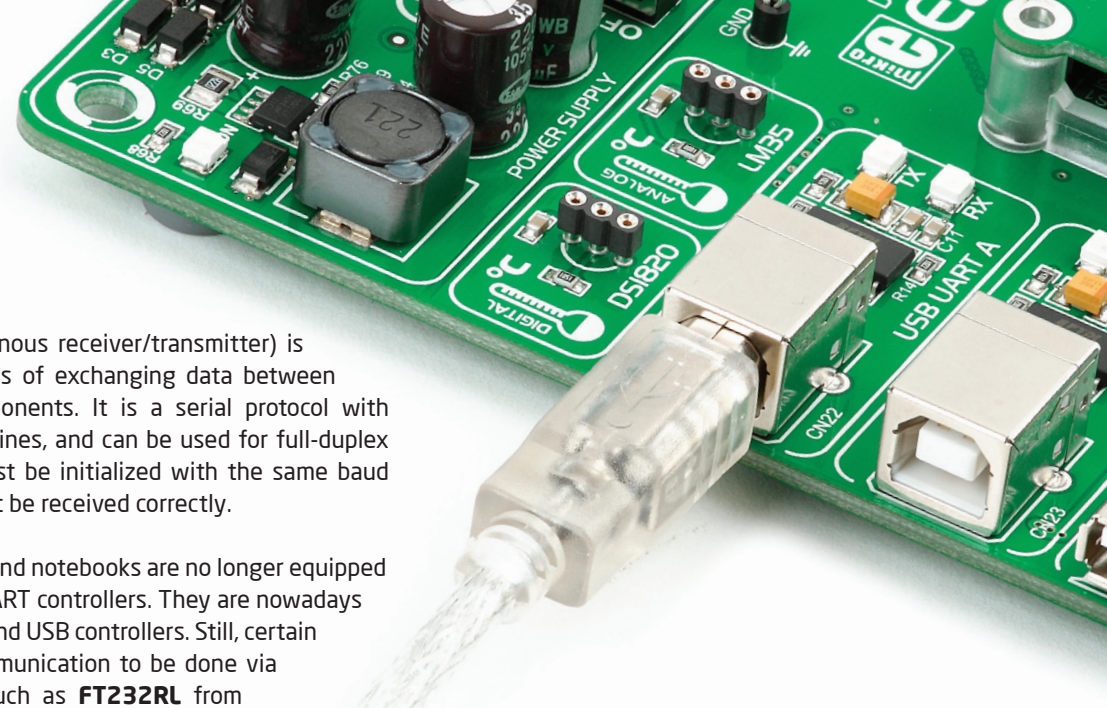


SHT1x click™

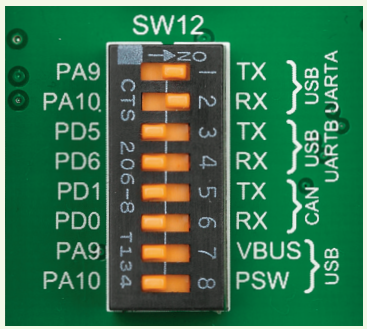


THERMO click™

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 link below:

http://www.ftdichip.com/Drivers/VCP.htm

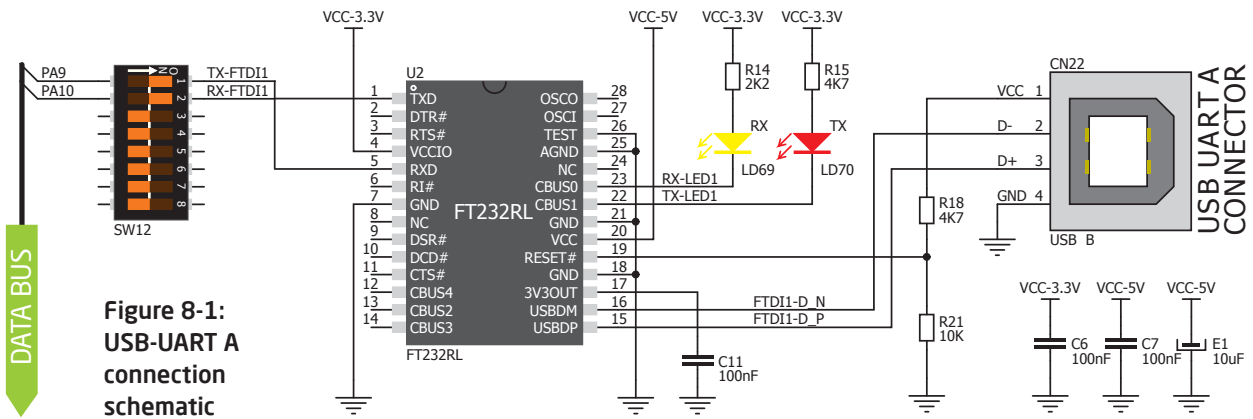
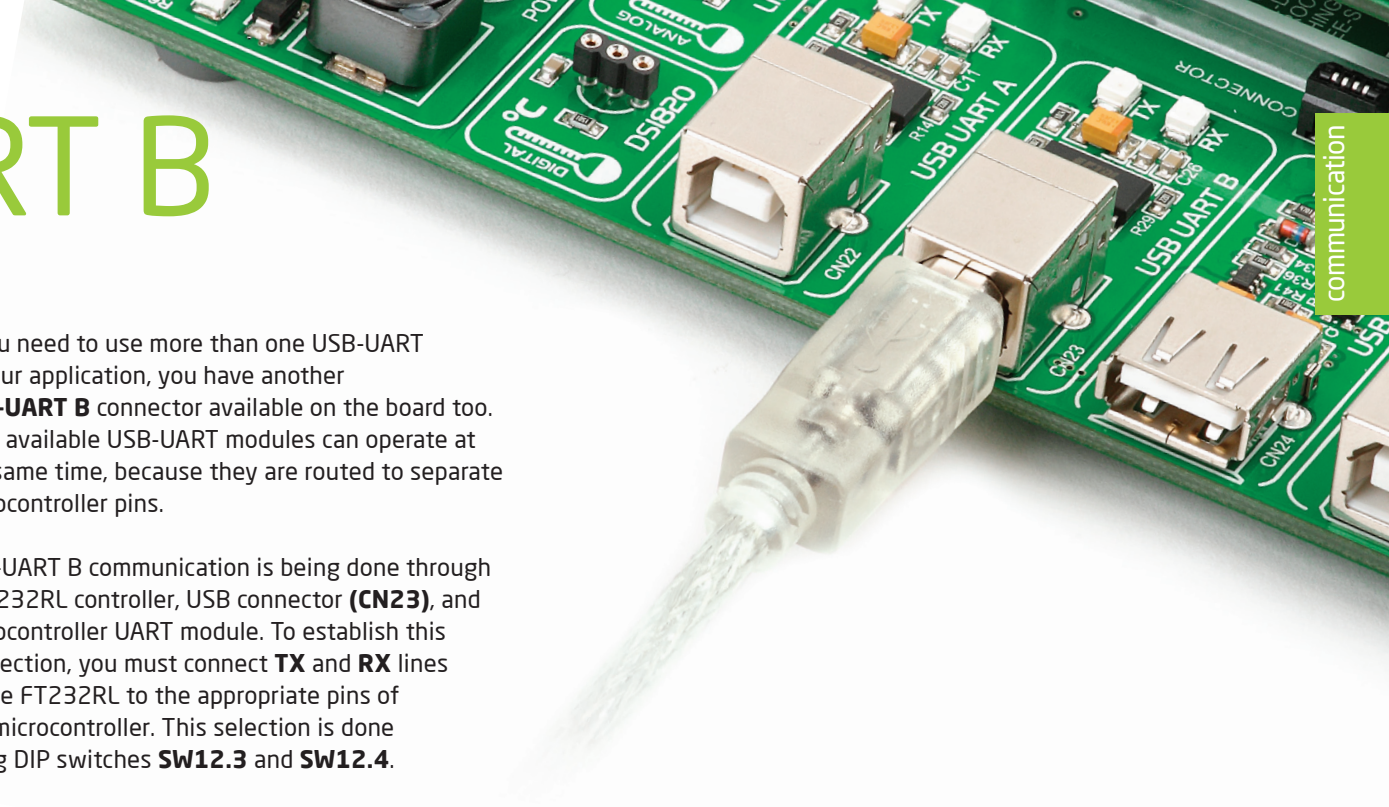
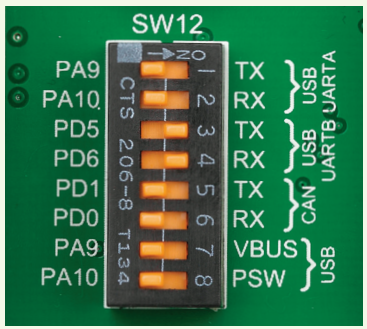


Figure 8-1: USB-UART A connection schematic

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 for STM32, you must first install FTDI drivers on your computer. Drivers can be found on the link bellow:

http://www.ftdichip.com/Drivers/VCP.htm

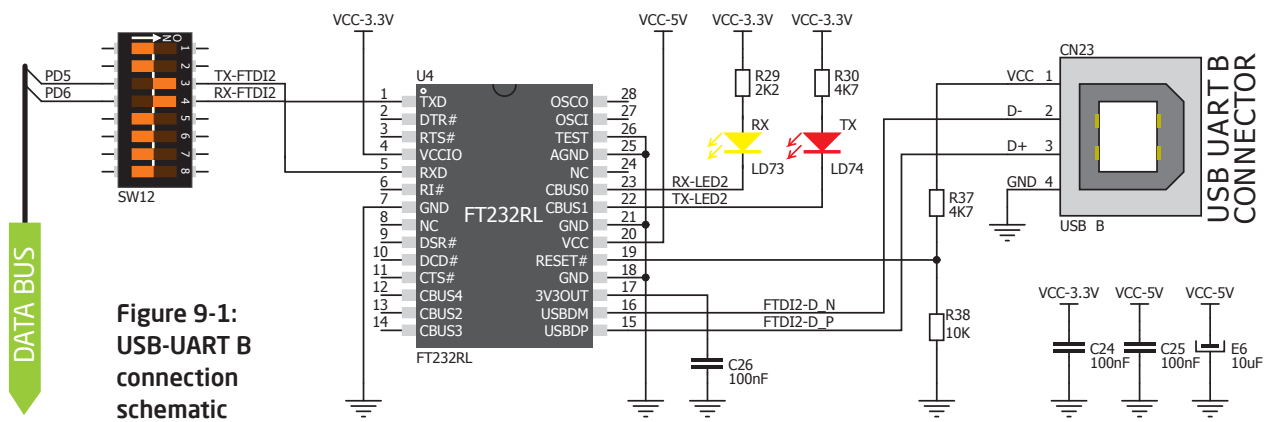
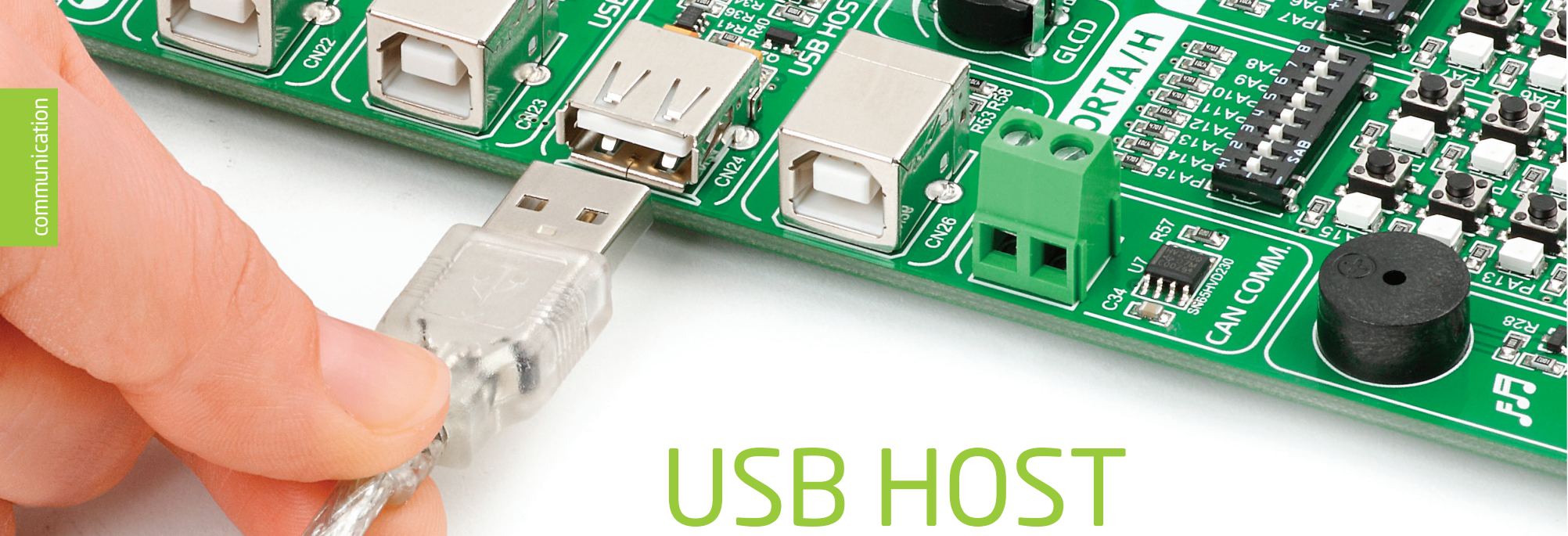


Figure 9-1: USB-UART B connection schematic



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.

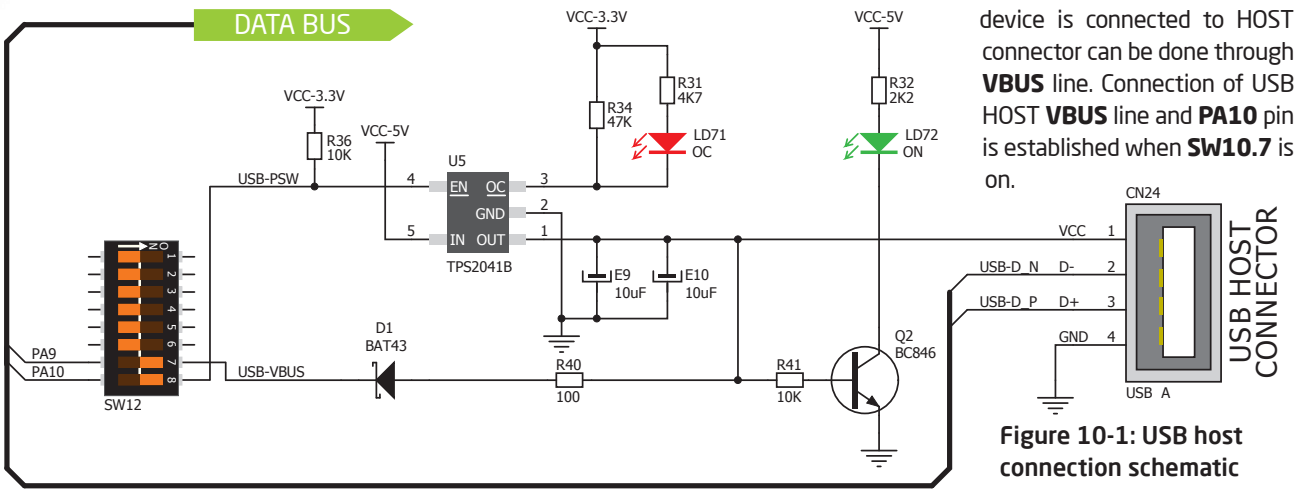


Figure 10-1: USB host connection schematic

Powering USB device

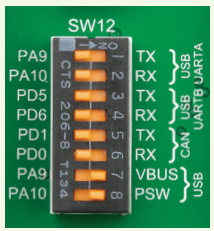
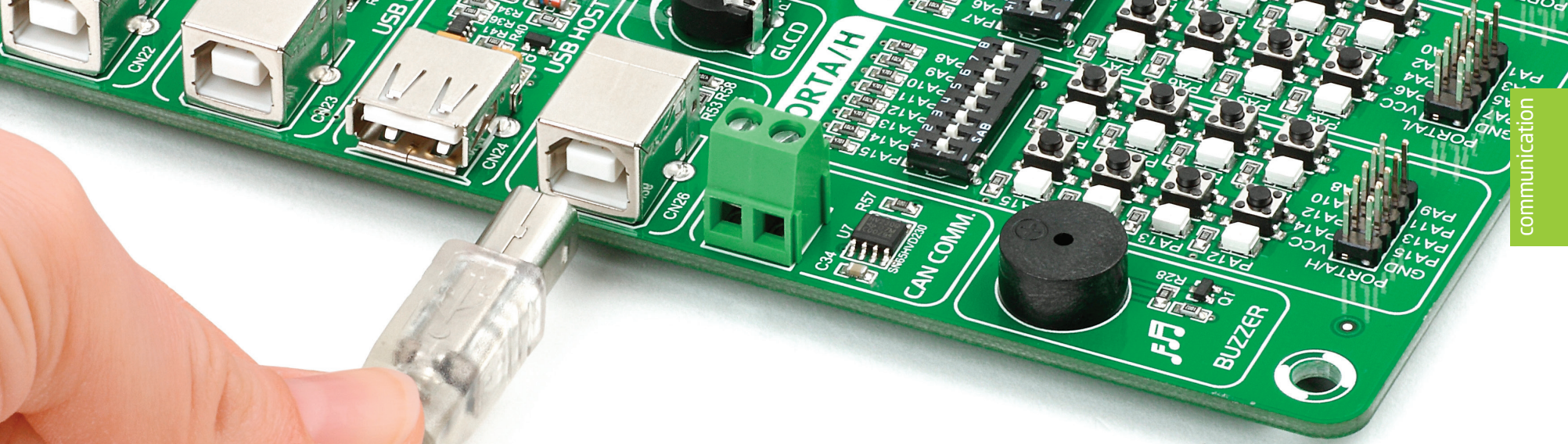


Figure 10-2: Powering USB device through PSW line

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.

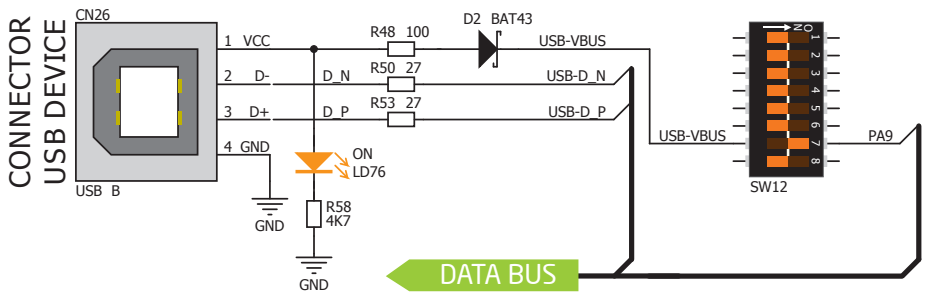


Figure 11-1: USB device connection schematic

Detecting connection

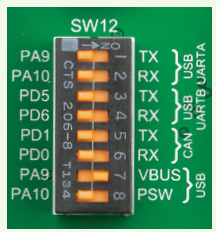
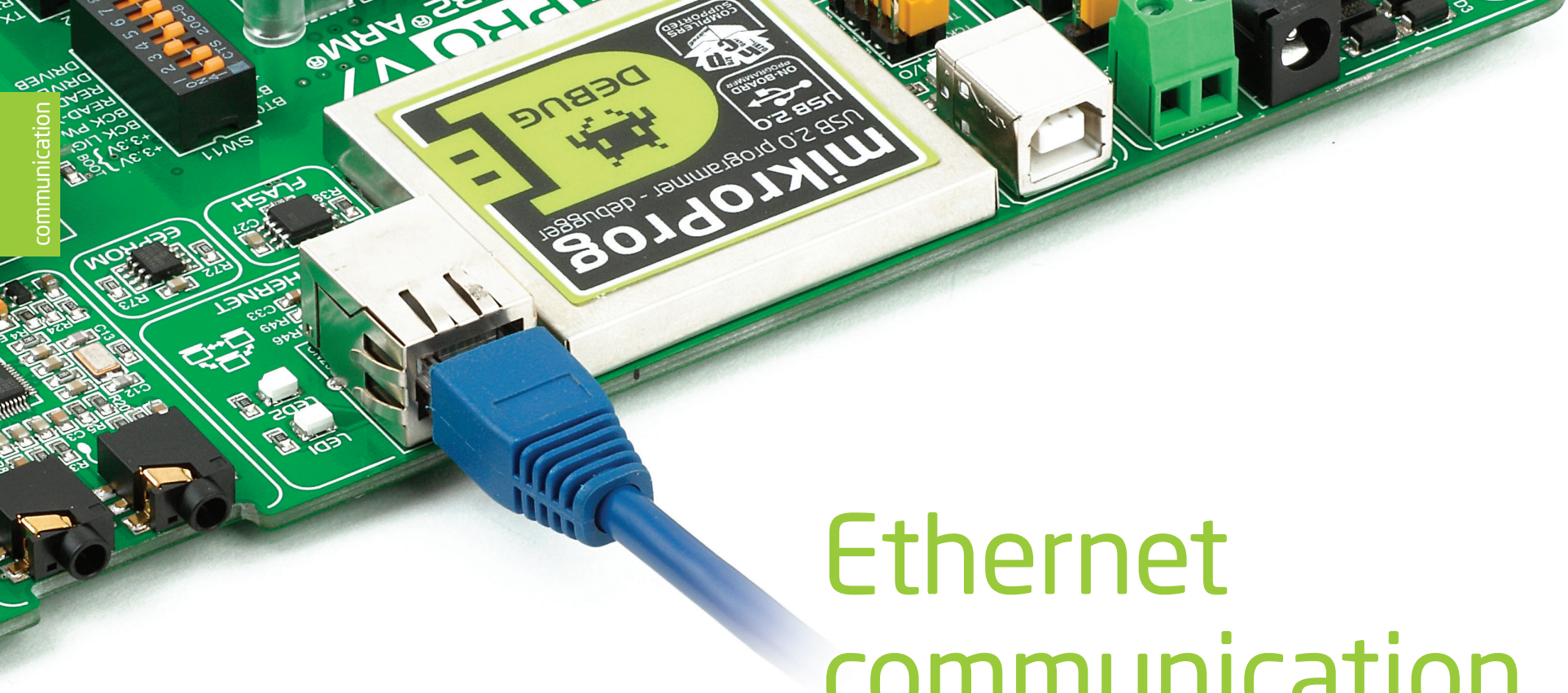


Figure 11-2: Enabling USB DEVICE detection via VBUS line

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.



Ethernet communication

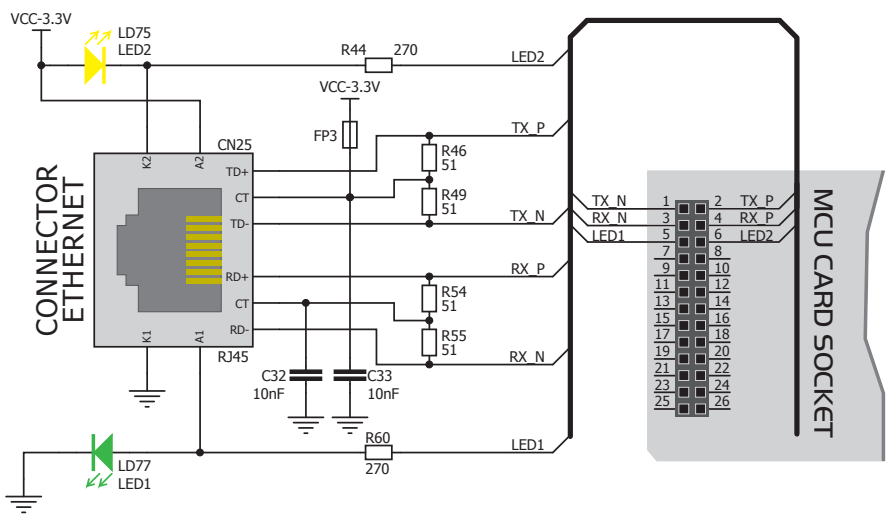
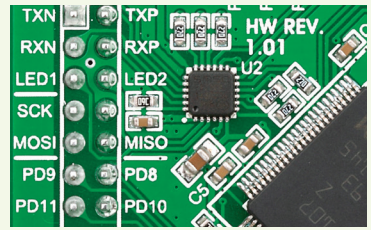


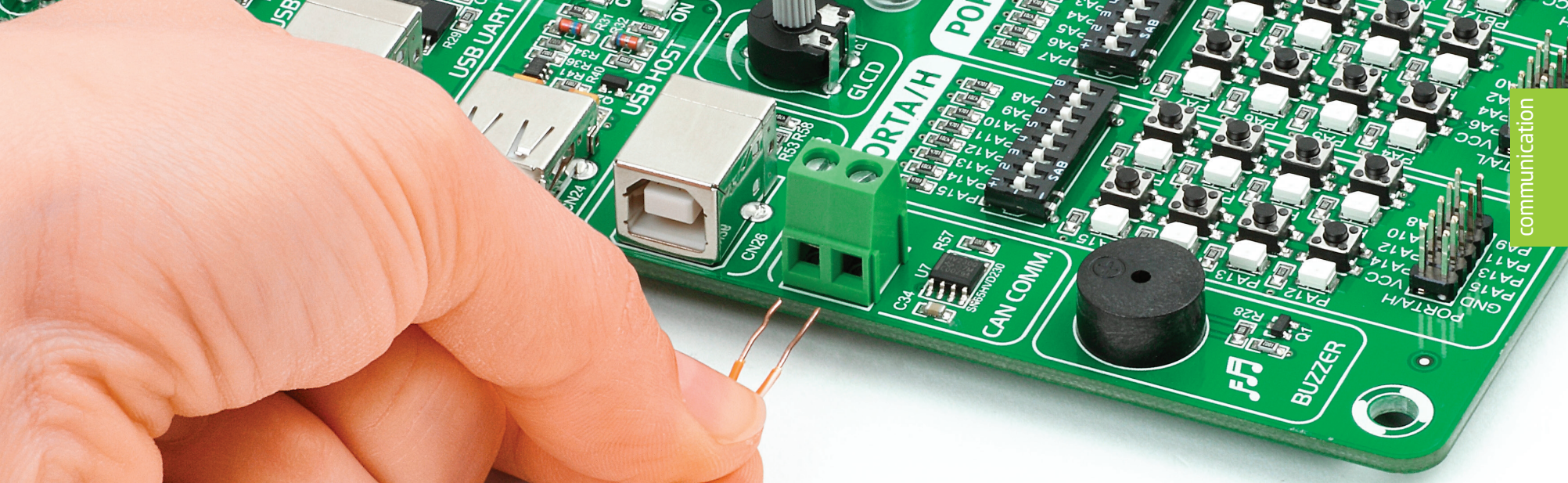
Figure 12-1: Ethernet connection schematic

Ethernet is a popular computer networking technology for local area networks (LAN). Systems communicating over Ethernet divide a stream of data into individual packets called frames. Each frame contains source and destination addresses and error-checking data so that damaged data can be detected and re-transmitted. EasyMx PRO™ v7 for STM32 features standard **RJ-45** connector which enables microcontrollers that support Ethernet communication to establish a connection with a computer, router or other devices. All four Ethernet lines (TPOUT+, TPOUT-, TPIN+ and TPIN-) are routed directly to the MCU card socket and cannot be accessed via PORT headers. Additional signalization LEDs (green and yellow) are provided on the Board next to **RJ-45** connector.

Ethernet MCU cards



Ethernet communication (TPOUT+, TPOUT-, TPIN+ and TPIN-) and signalization lines (LED1, LED2) are routed directly to the MCU card socket and can be used only with a Ethernet MCU cards (ETH MCU, HP ETH MCU, **Page 11**).



CAN communication

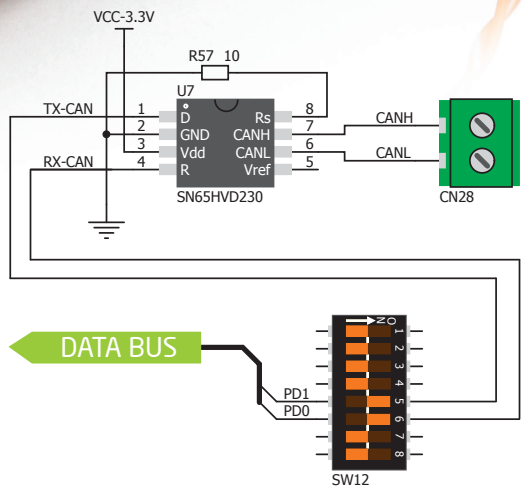


Figure 13-1: CAN connection schematic

Controller Area Network (CAN or CAN bus) is a vehicle bus standard designed to allow microcontrollers and devices to communicate with each other within a vehicle without a host computer. CAN is a message-based protocol, designed specifically for automotive applications but now also used in other areas such as industrial automation and medical equipment. EasyMx PRO™ v7 for STM32 is equipped with **SN65HVD230** - a 3.3V CAN Transceiver and a pair of screw terminals which provide microcontrollers with integrated CAN controller with the necessary physical interface for CAN communication. Make sure to correctly connect negative and positive differential communication lines before using this module.

Enabling CAN

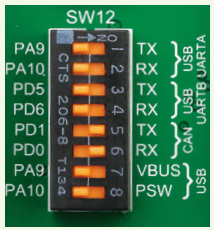


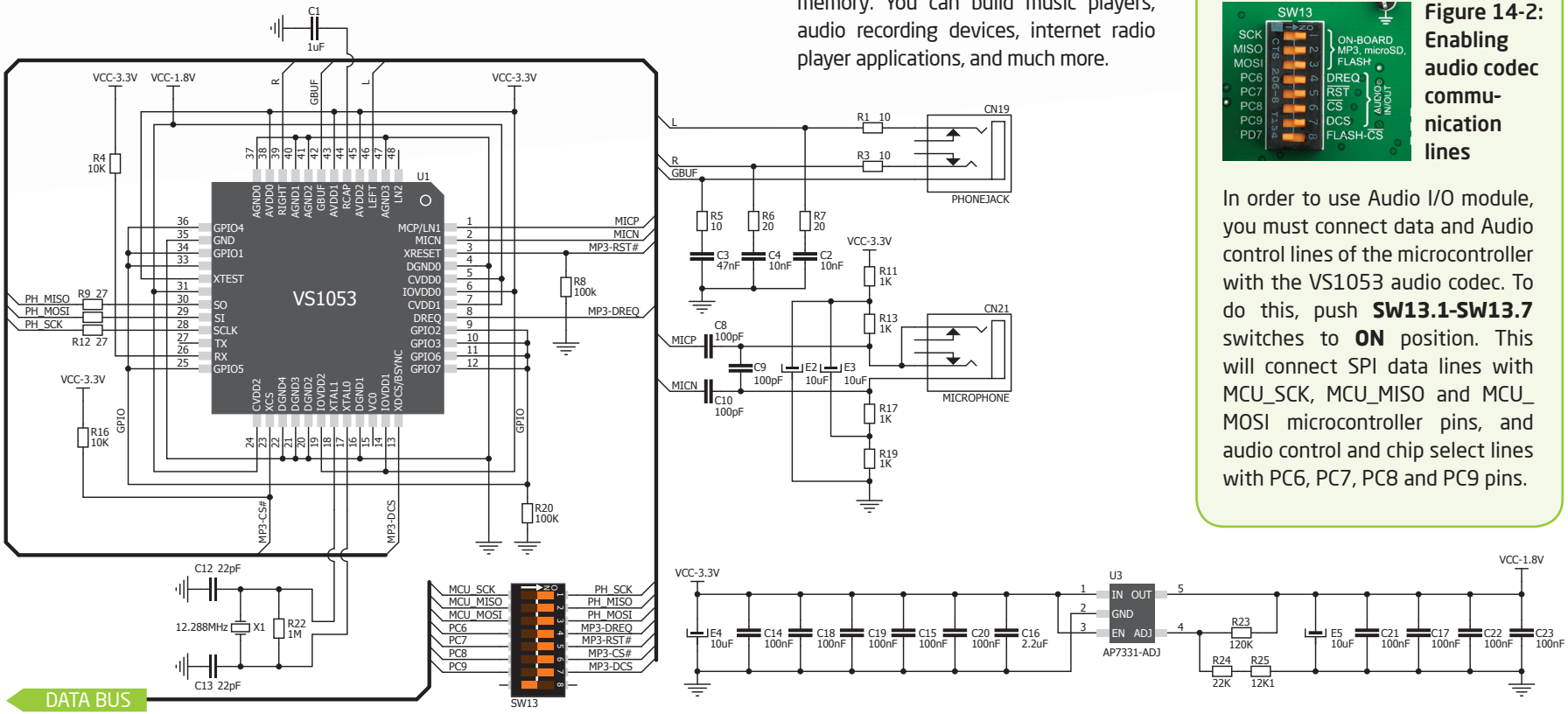
Figure 13-2: enabling CAN communication

In order to enable CAN communication, you must push **SW12.5** (PD1) and **SW12.6** (PD0) to **ON** position. This connects the **TX** and **RX** lines to appropriate microcontroller pins and its CAN module.

Audio I/O

It's hard to imagine modern multimedia devices without high quality audio reproduction modules. Sounds and music are almost as important as graphical user interfaces. Along with other multimedia modules, EasyMx PRO™ v7 for STM32 contains high end stereo **VS1053** audio codec. It features Ogg Vorbis/MP3/AAC/WMA/FLAC/WAV/MIDI **audio decoder**, as well as an PCM/IMA ADPCM/Ogg Vorbis **encoder** on a single chip. Board also contains **two stereo audio connectors** for interfacing with standard 3.5mm stereo audio jacks. VS1053 receives the input bit stream through a serial input bus, which it listens to as a system slave. The input stream is decoded and passed through a digital volume control to an 18-bit oversampling, multi-bit, sigma-delta Digital to Analog Converter (DAC). The decoding is controlled via a serial control bus. In addition to the basic decoding, it is possible to add application specific features like DSP effects to the user RAM memory. You can build music players, audio recording devices, internet radio player applications, and much more.

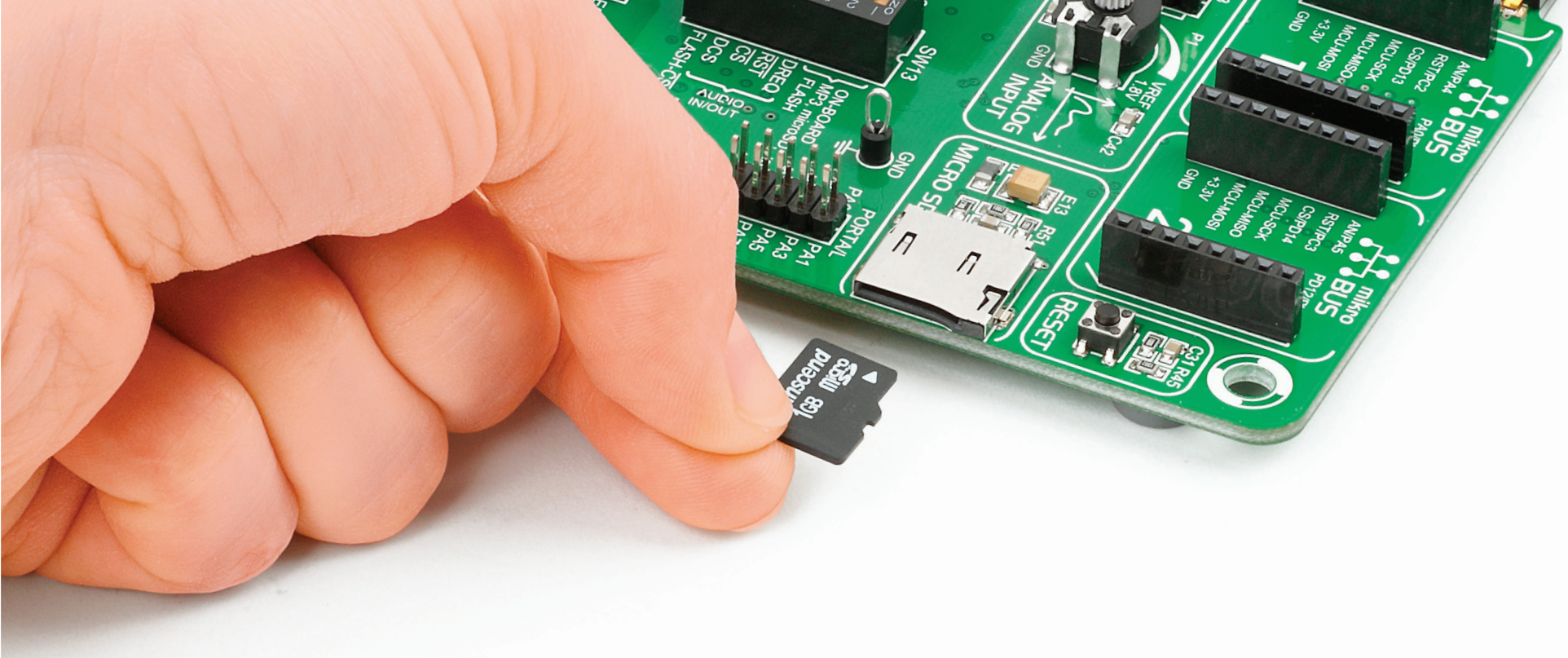
Figure 14-1: Audio IN/OUT connection schematics



Enabling Audio I/O

Figure 14-2: Enabling audio codec communication lines

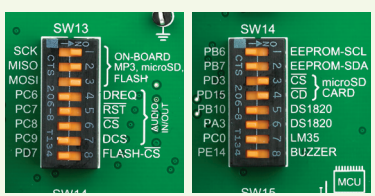
In order to use Audio I/O module, you must connect data and Audio control lines of the microcontroller with the VS1053 audio codec. To do this, push **SW13.1-SW13.7** switches to **ON** position. This will connect SPI data lines with MCU_SCK, MCU_MISO and MCU_MOSI microcontroller pins, and audio control and chip select lines with PC6, PC7, PC8 and PC9 pins.



microSD card slot

Secure Digital (SD) is a non-volatile memory card format developed for use in portable devices. It comes in different packages and memory capacities. It is mostly used for storing large amounts of data. EasyMx PRO™ v7 for STM32 features the **microSD card slot**. The microSD form factor is the smallest card format currently available. It uses standard SPI user interface with minimum additional electronics, mainly used for stabilizing communication lines which can be significantly distorted at high transfer rates. Special ferrite is also provided to compensate the voltage and current glitch that can occur when pushing-in and pushing-out microSD card into the socket.

Enabling microSD



In order to access microSD card, you must enable SPI communication lines using **SW13.1** - **SW13.3** DIP switches, as well as Chip Select ($\overline{CS}$) and Card Detect ($\overline{CD}$) lines using **SW14.3** and **SW14.4** switches.

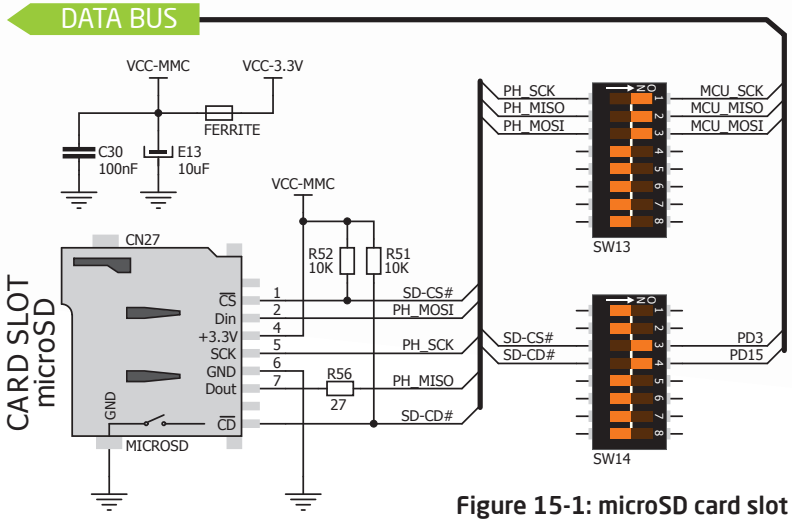


Figure 15-1: microSD card slot connection schematics

TFT display 320x240 pixels

One of the most powerful ways of presenting data and interacting with users is through color displays and touch panel inputs. This is a crucial element of any multimedia device. EasyMx PRO™ v7 for STM32 features EasyTFT board carrying 320x240 pixel 2.83" color TFT display with LED back-light and **HX8347D** controller.

Each pixel is capable of showing 262.144 different colors. TFT display is connected to microcontroller PORTE using standard 8080 parallel 8-bit interface, with additional control lines. Board features back-light driver which besides standard mode can also be driven with PWM signal in order to regulate brightness in 0 to 100% range.

IMPORTANT: *In order to use PWM back-light both* **SW11.3** *and* **SW11.4** *switches must be enabled at the same time.*

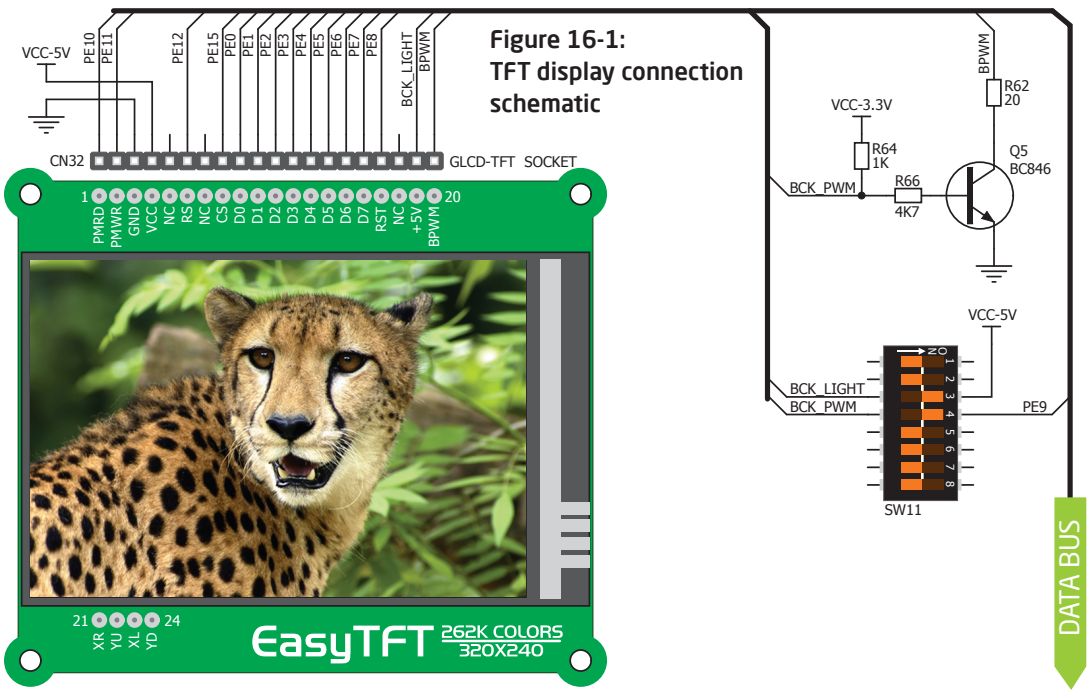


Figure 16-1: TFT display connection schematic

Driving Display Back-light

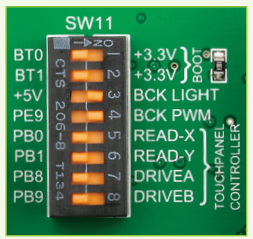


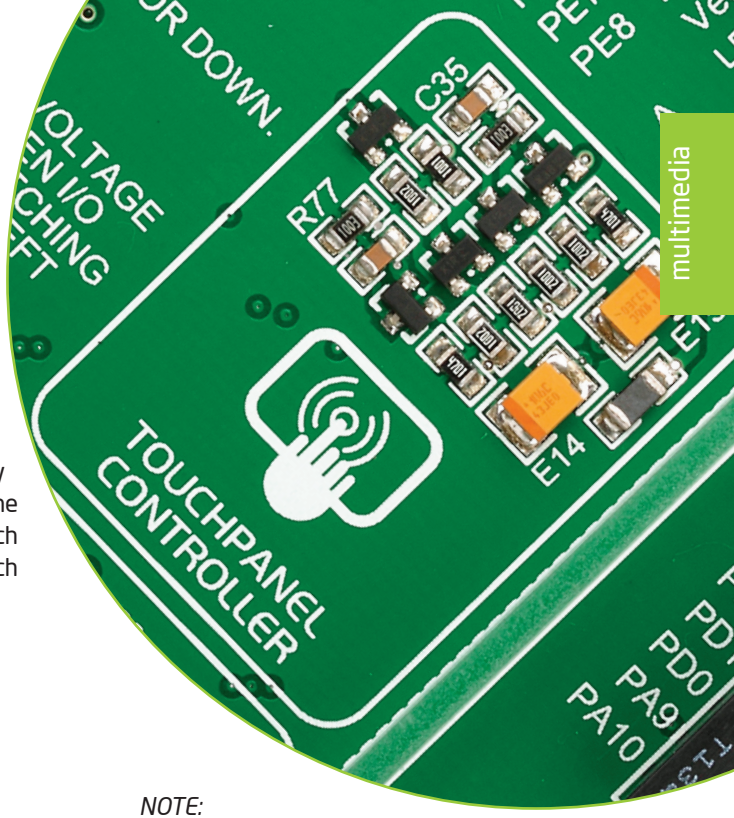
Figure 16-2: Turn on switches SW11.3 and SW11.4 to enable back-light

TFT display is enabled using **SW11.3-SW11.4** DIP switches. Back-light can be enabled in two different ways:

1. It can be **turned on with full brightness** using **SW11.3** switch.
2. Brightness level can be determined **with PWM signal** from the microcontroller, allowing you to write custom back-light controlling software. This back-light mode is enabled when both **SW11.3** and **SW11.4** switches are in ON position.

Touch Panel controller

Touch panel is a glass panel whose surface is covered with two layers of resistive material. When the screen is pressed, the outer layer is pushed onto the inner layer and appropriate controllers can measure that pressure and pinpoint its location. This is how touch panels can be used as an input devices. EasyMx PRO™ v7 for STM32 is equipped with touch panel controller and connector for **4-wire resistive touch panels**. It can very accurately register pressure at a specific point, representing the touch coordinates in the form of analog voltages, which can then be easily converted to X and Y values. Touch panel comes as a part of TFT 320x240 display.



Enabling Touch panel

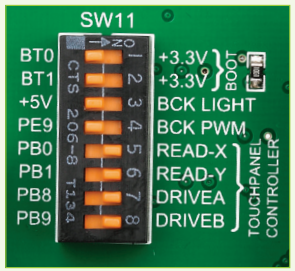
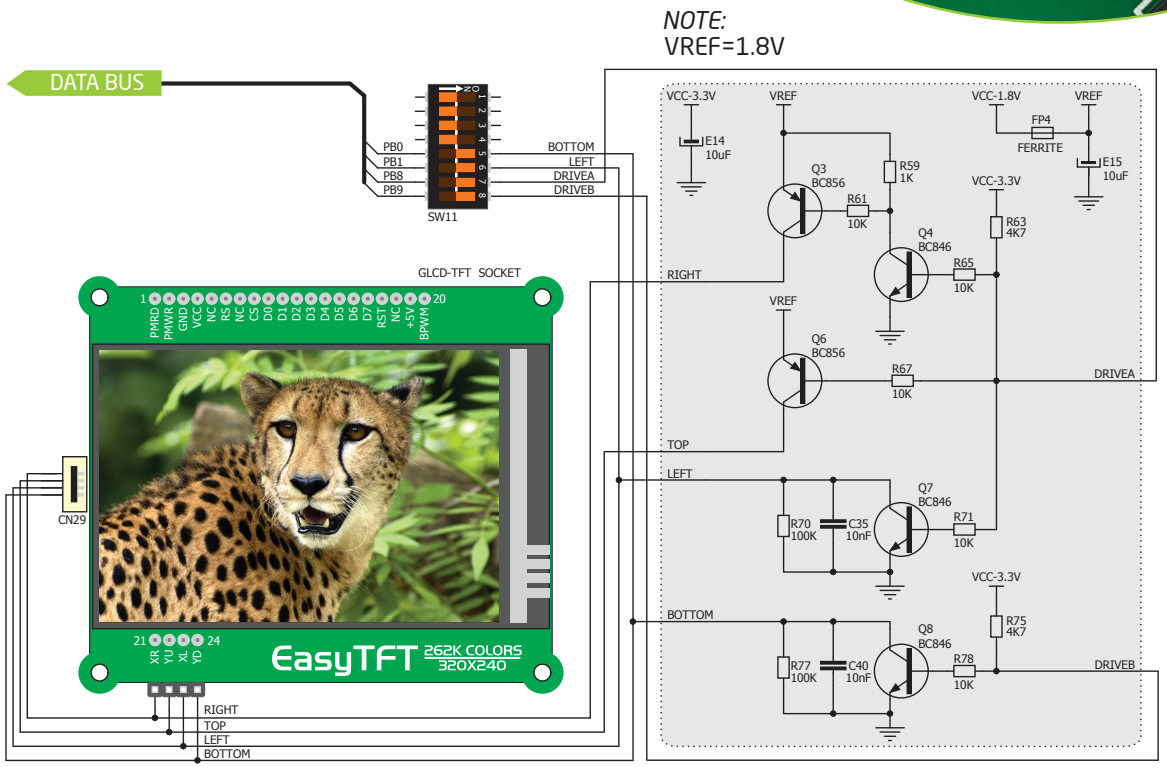
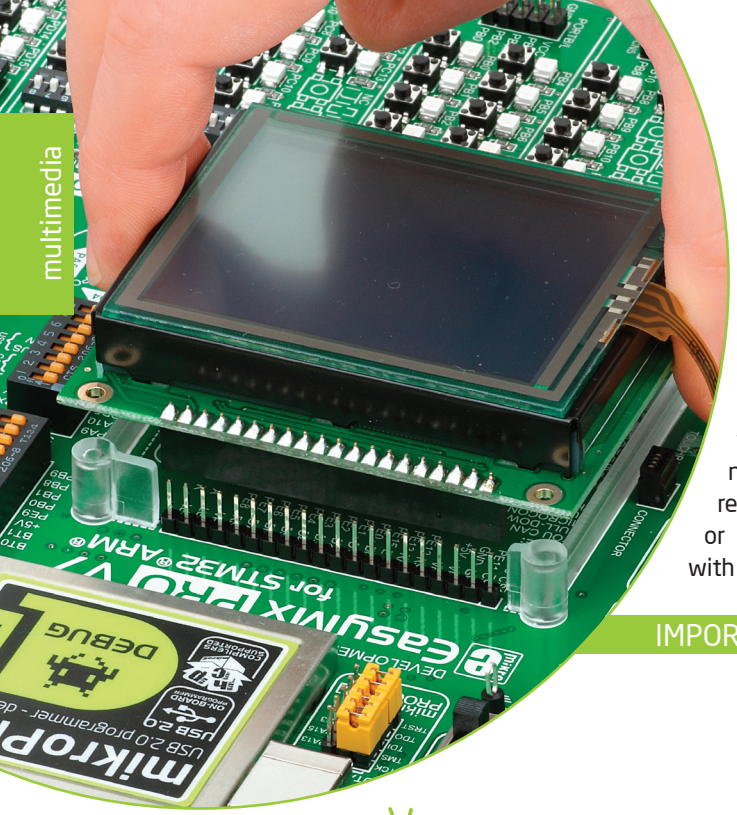


Figure 17-2: Turn on switches **SW11.5, SW11.6, SW11. 7** and **SW11. 8** to enable Touch panel controller

Touch panel is enabled using **SW11.5**, **SW11.6**, **SW11.7** and **SW11.8** switches. They connect **BOTTOM** and **LEFT** lines of the touch panel with **PB0** and **PB1** analog inputs, and **DRIVEA** and **DRIVEB** with **PB8** and **PB9** digital outputs on microcontroller sockets. Make sure to disconnect other peripherals, LEDs and additional pull-up or pull-down resistors from the interface lines so they do not interfere with signal/data integrity.

Figure 17-1: Touch Panel controller and connection schematic





GLCD 128x64

Graphical Liquid Crystal Displays, or **GLCDs** are used to display monochromatic graphical content, such as text, images, human-machine interfaces and other content. EasyMx PRO™ v7 for STM32 provides the connector and necessary interface for supporting GLCD with resolution of 128x64 pixels, driven by the KS108 or compatible display controller. Communication with the display module is done through **CN32** display connector. Board is fitted with uniquely designed plastic display distancer, which allows the GLCD module to perfectly and firmly fit into place.

Display connector is routed to PORTE (control and data lines) of the microcontroller sockets. PORTE is also used by TFT display. You can control the display contrast using dedicated potentiometer **P2**. Full brightness display back-light can be enabled with **SW13.3** switch, and PWM-driven back-light with **SW13.4** switch.

IMPORTANT: *In order to use PWM back-light both* **SW13.3** *and* **SW13.4** *switches must be enabled at the same time.*

Connector pinout explained

CS1 and **CS2** - Controller Chip Select lines
VCC - +5V display power supply
GND - Reference ground
Vo - GLCD contrast level from potentiometer P3
RS - Data (High), Instruction (Low) selection line
R/W - Determines whether display is in Read or Write mode.
E - Display Enable line
D0-D7 - Data lines
RST - Display reset line
Vee - Reference voltage for GLCD contrast potentiometer P3
LED+ - Connection with the back light LED anode
LED- - Connection with the back light LED cathode

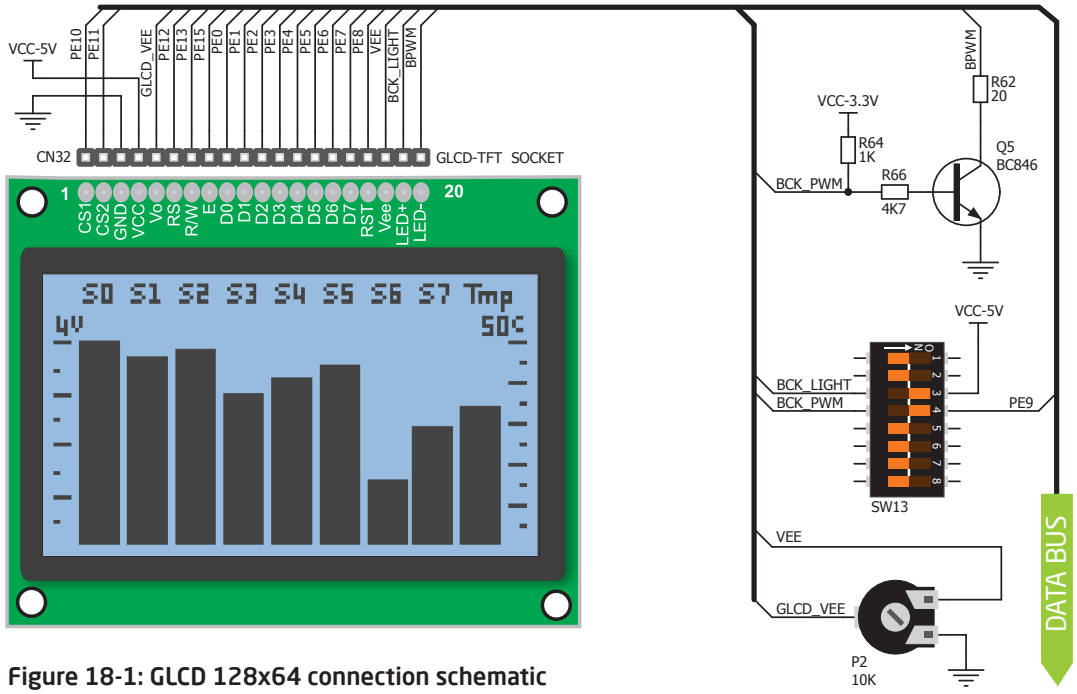
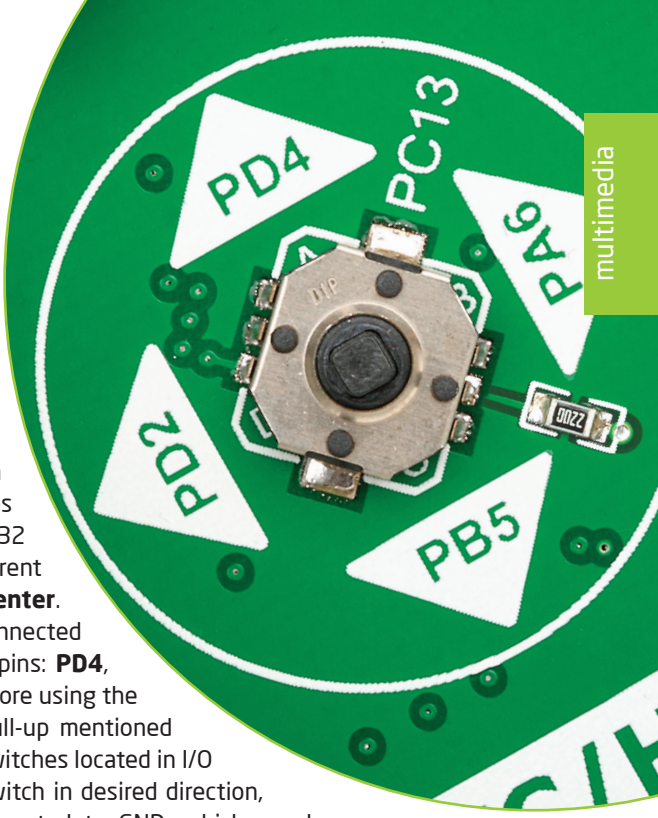


Figure 18-1: GLCD 128x64 connection schematic

Navigation switch



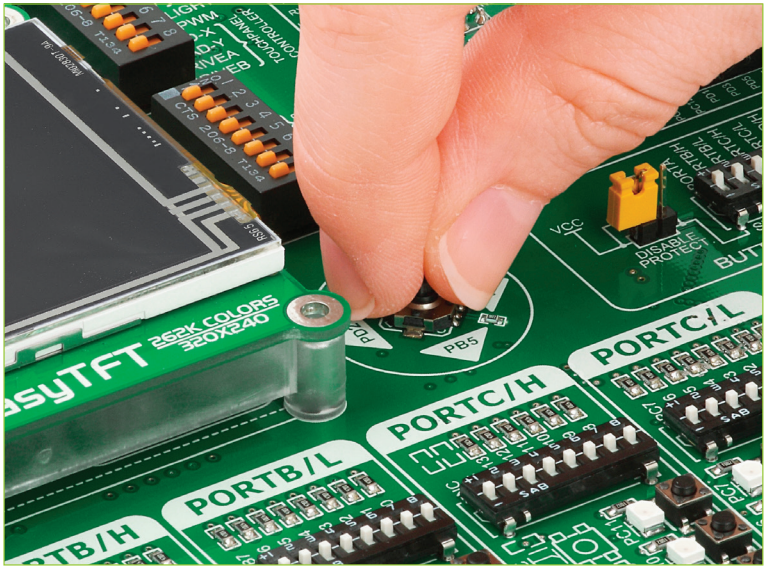


Figure 19-2: Navigation switch is an intuitive solution for browsing through on-screen menus.

When working with multimedia applications it is far more intuitive to use a single joystick than several different push buttons that are more far apart. This is more natural for users and they can browse through on-screen menus, or even play games much easier. EasyMx PRO™ v7 for STM32 features navigation switch with five different positions: **Up**, **Down**, **Left**, **Right** and **Center**. Each of those acts as a button, and is connected to one of the following microcontroller pins: **PD4**, **PB5**, **PD2**, **PA6**, **PC13** (respectively). Before using the navigation switch, it is necessary to pull-up mentioned microcontroller pins using tri-state DIP switches located in I/O groups. After pressing the navigation switch in desired direction, associated microcontroller pins are connected to GND, which can be detected in user software.

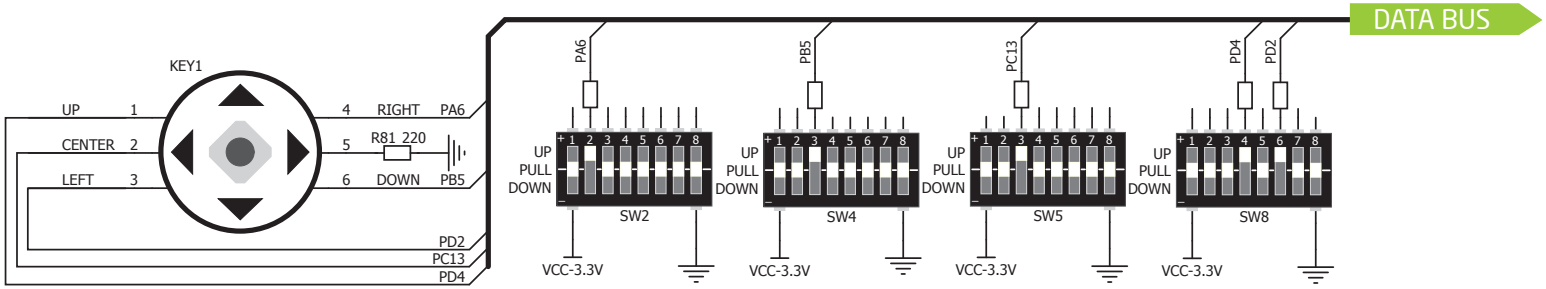
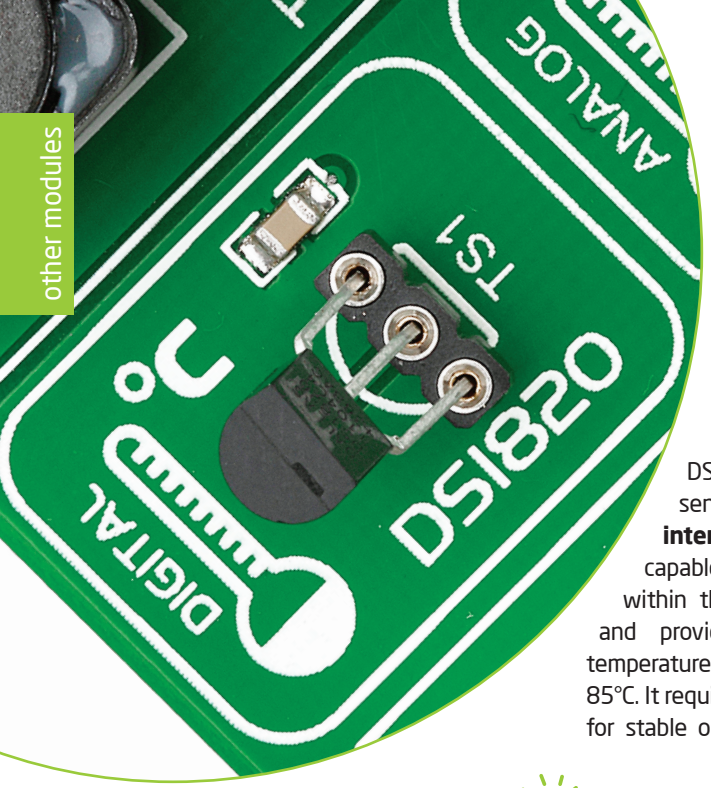


Figure 19-1: Navigation switch connection schematic. Pull-up resistors should be enabled during operation



DS1820 - Digital Temperature Sensor

DS1820 is a digital temperature sensor that uses **1-wire® interface** for it's operation. It is capable of measuring temperatures within the range of -55 to 128°C, and provides ±0.5°C accuracy for temperatures within the range of -10 to 85°C. It requires 3V to 5.5V power supply for stable operation. It takes maximum of 750ms for the DS1820 to calculate temperature with 9-bit resolution. **1-wire® serial communication** enables data to be transferred over a single communication line, while the process itself is under the control of the master microcontroller. The advantage of such communication is that only one microcontroller pin is used. Multiple sensors can be connected on the same line. All slave devices by default have a unique ID code, which enables the master device to easily identify all devices sharing the same interface.

Board provides a separate socket (**TS1**) for the DS1820. Communication line with the microcontroller is established using **SW14.5** or **SW14.6** DIP switch (ON position).

Enabling DS1820 Sensor

Figure 20-1: DS1820 socket

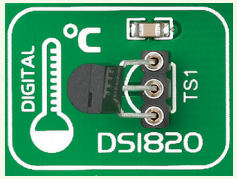


Figure 20-2: DS1820 correctly placed in socket

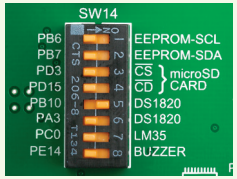


Figure 20-3: Enabled SW14.5 DIP switch

EasyMx PRO™ v7 for STM32 enables you to establish 1-wire® communication between **DS1820** and the microcontroller over **PB10** or **PA3** pin. The connection is done placing **SW14.5** or **SW14.6** DIP switch to ON position (**Figure 20-3**). When placing the sensor in the socket make sure that half-circle on the board's silkscreen markings matches the rounded part of the DS1820 sensor. If you accidentally connect the sensor the other way, it may be permanently damaged. Make sure to disconnect other peripherals, LEDs and additional pull-up or pull-down resistors from the interface lines in order not to interfere with signal/data integrity.

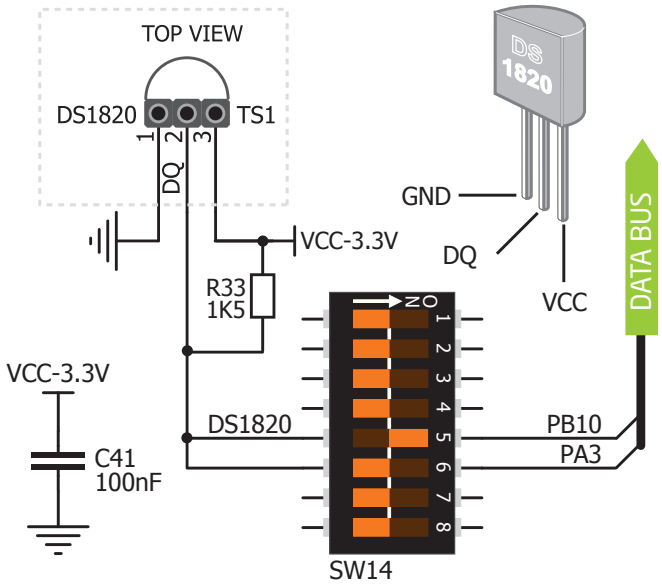
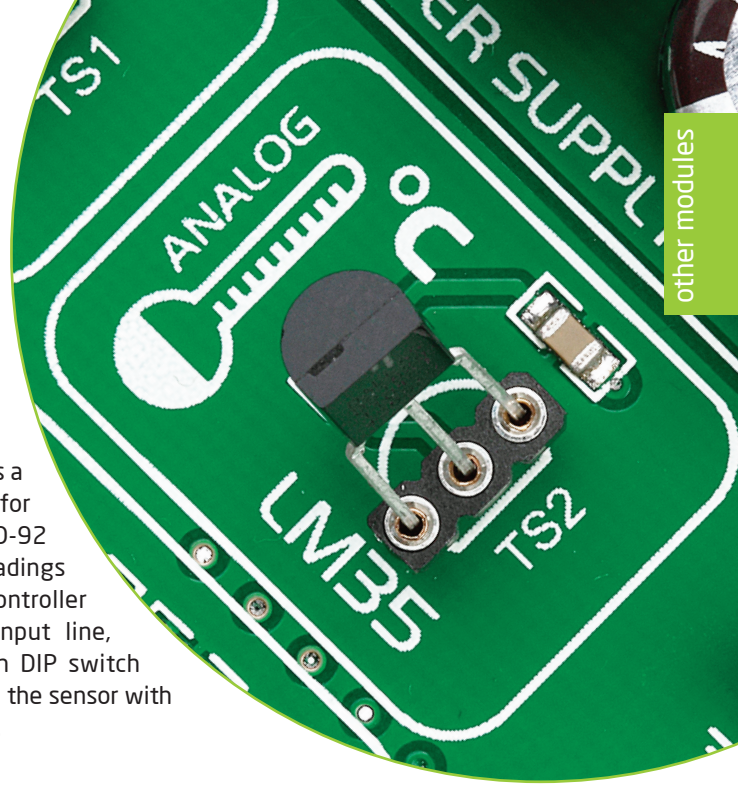


Figure 20-4: DS1820 connected to PB10 pin

LM35 - Analog Temperature Sensor



The **LM35** is a low-cost precision integrated-circuit temperature sensor, whose output voltage is linearly proportional to the Celsius (Centigrade) temperature. The LM35 thus has an advantage over linear temperature sensors calibrated in ° Kelvin, as the user is not required to subtract a large constant voltage from its output to obtain convenient Centigrade scaling. It has a linear +10.0 mV/°C scale factor and less than 60 µA current drain. As it draws only 60 µA from its supply, it has very low self-heating, less than 0.1°C in still air. EasyMx PRO™ v7 for STM32 enables you to get analog readings from the LM35 sensor in restricted temperature range from +2°C to +150°C. Board provides a separate socket (**TS2**) for the LM35 sensor in TO-92 plastic packaging. Readings are done with microcontroller using single analog input line, which is selected with DIP switch **SW14**. Switch connects the sensor with **PC0** microcontroller pin.

Enabling LM35 Sensor

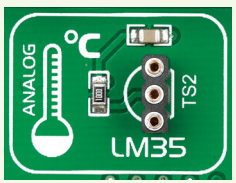


Figure 21-1: LM35 socket

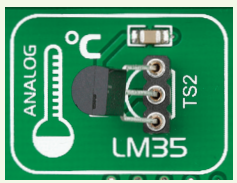


Figure 21-2: LM35 correctly placed in socket

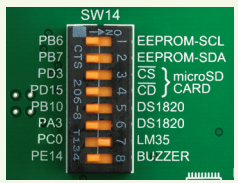


Figure 21-3: Enabled SW14.7 DIP switch

EasyMx PRO™ v7 for STM32 enables you to get analog readings from the LM35 sensor using **PC0** microcontroller pin. The connection is done placing **SW14.7** DIP switch to ON position (**Figure 21-3**). When placing the sensor in the socket make sure that half-circle on the board's silkscreen markings matches the rounded part of the LM35 sensor. If you accidentally connect the sensor the other way, it can be permanently damaged and you might need to replace it with another one. During the readings of the sensor, make sure that no other device uses the selected analog line, because it may interfere with the readings.

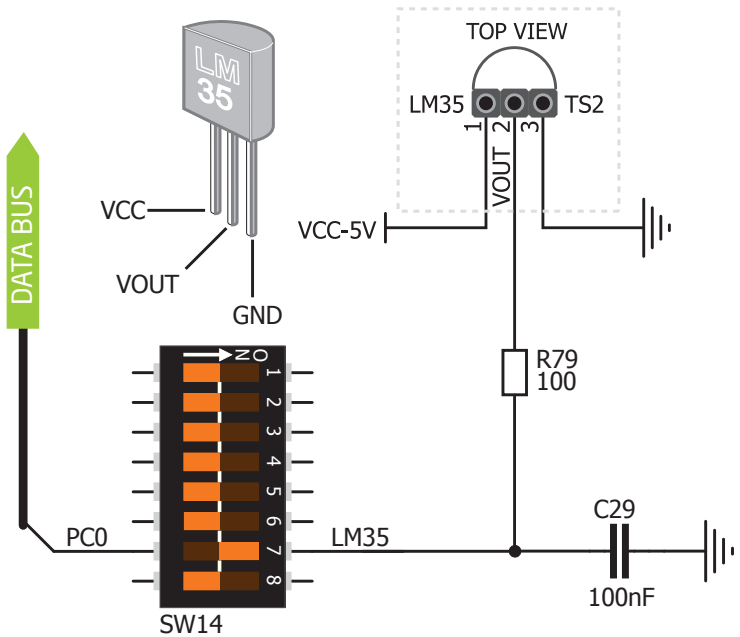
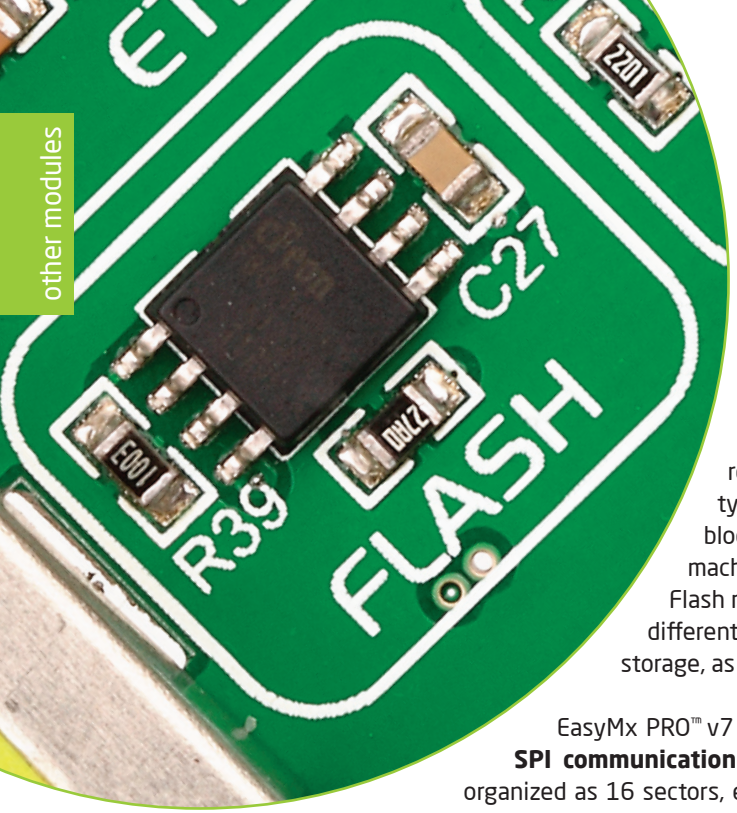


Figure 21-4: LM35 connected to PC0 pin



Serial Flash Memory

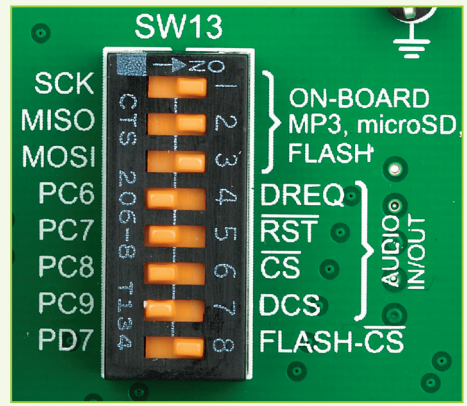
Flash memory is a non-volatile storage chip that can be electrically erased and reprogrammed. It was developed from EEPROM (electrically erasable programmable read-only memory) and must be erased in fairly large blocks before these can be rewritten with new data. The high density NAND type must also be programmed and read in (smaller) blocks, or pages, while the NOR type allows a single machine word (byte) to be written or read independently. Flash memories come in different sizes and supporting different clock speeds. They are mostly used for mass storage, as in USB Flash Drives, which are very popular today.

EasyMx PRO™ v7 features **M25P80** Serial Flash Memory which uses **SPI communication interface** and has **8 Mbits** of available memory, organized as 16 sectors, each containing 256 pages. Each page is 256 bytes wide. Thus, the whole memory can be viewed as consisting of 4096 pages, or 1,048,576 bytes. Maximum clock frequency for READ instructions is 40MHz.

What is SPI?

The **Serial Peripheral Interface Bus** or SPI bus is a synchronous serial data link standard that operates in full duplex mode. It consists of four lines **MISO** (Master Input Slave Output), **MOSI** (Master Output Slave Input), **SCK** (Clock) and **CS** (Chip Select). Devices communicate in master/slave mode where the master device initiates the data frame. Multiple slave devices are allowed with individual slave select (chip select) lines.

Enabling Serial Flash



In order to connect Serial Flash Memory to the microcontroller you must enable **SW13.1, SW13.2, SW13.3** and **SW13.8** switches. This connects SPI lines to **MCU_MOSI**, **MCU_MISO**, **MCU_SCK** and **PD7** (CS) microcontroller pins.

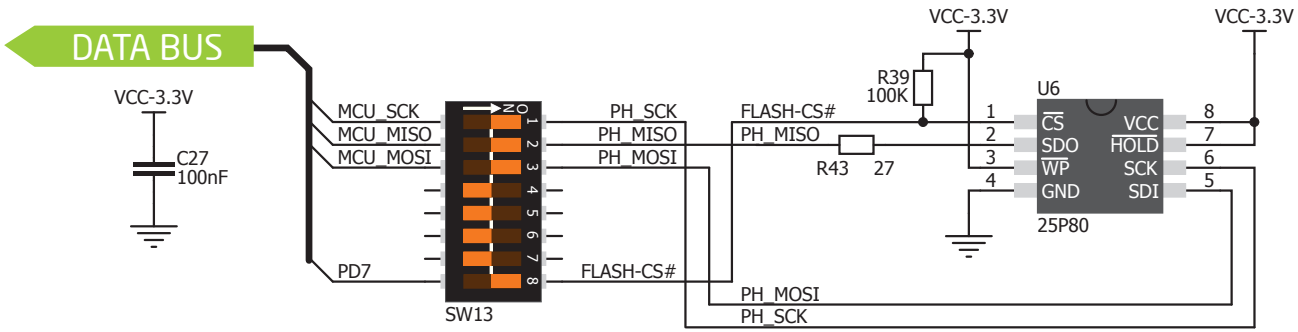
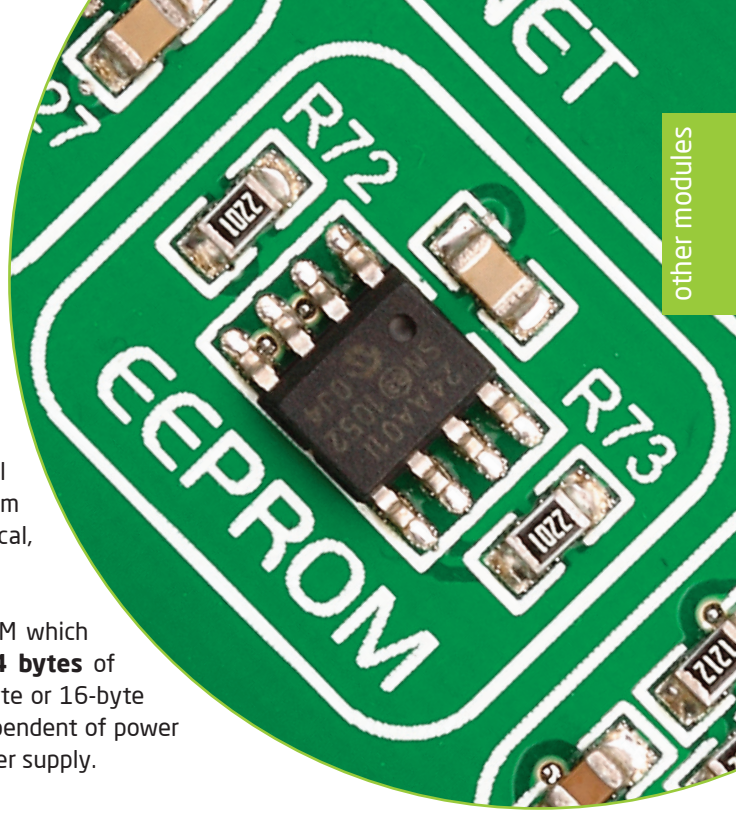


Figure 22-1: Schematic of Serial Flash Memory module

I²C EEPROM



Enabling I²C EEPROM

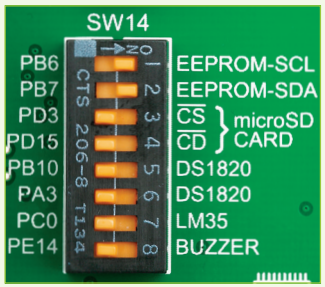


Figure 23-2: Turn on switches SW14.1 and SW14.2 to connect EEPROM lines to MCU

In order to connect I²C EEPROM to the microcontroller you must enable **SW14.1** and **SW14.2** switches, as shown on **Figure 23-2**. **2K2** pull-up resistors necessary for I²C communication are already provided on **SDA** and **SCL** lines once switches are turned on. Prior to using EEPROM in your application, make sure to disconnect other peripherals, LEDs and additional pull-up or pull-down resistors from the **PB6** and **PB7** communication lines that could interfere with the data signals and cause data corruption.

EEPROM is short for **Electrically Erasable Programmable Read Only Memory**. It is usually a secondary storage memory in devices containing data that is retained even if the device looses power supply. Because of the ability to alter single bytes of data, EEPROM devices are used to store personal preference and configuration data in a wide spectrum of consumer, automotive, telecommunication, medical, industrial, and PC applications.

EasyMx PRO™ v7 for STM32 supports serial EEPROM which uses **I²C communication interface** and has **1024 bytes** of available memory. EEPROM itself supports single byte or 16-byte (page) write and read operations. Data rates are dependent of power supply voltage, and go up to **400 kHz** for 3.3V power supply.

What is I²C?

I²C is a multi-master serial single-ended bus that is used to attach low-speed peripherals to computer or embedded systems. I²C uses only two open-drain lines, **Serial Data Line (SDA)** and **Serial Clock (SCL)**, pulled up with resistors. **SCL** line is driven by a master, while **SDA** is used as bidirectional line either by master or slave device. Up to 112 slave devices can be connected to the same bus. Each slave must have a unique address.

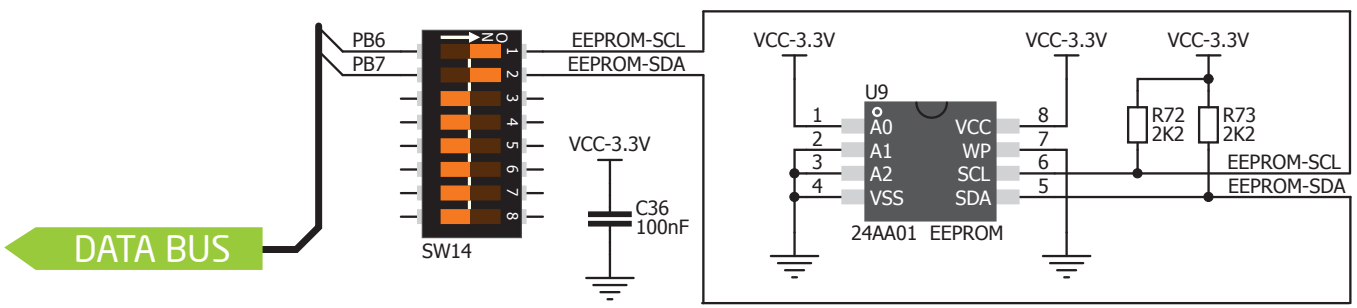
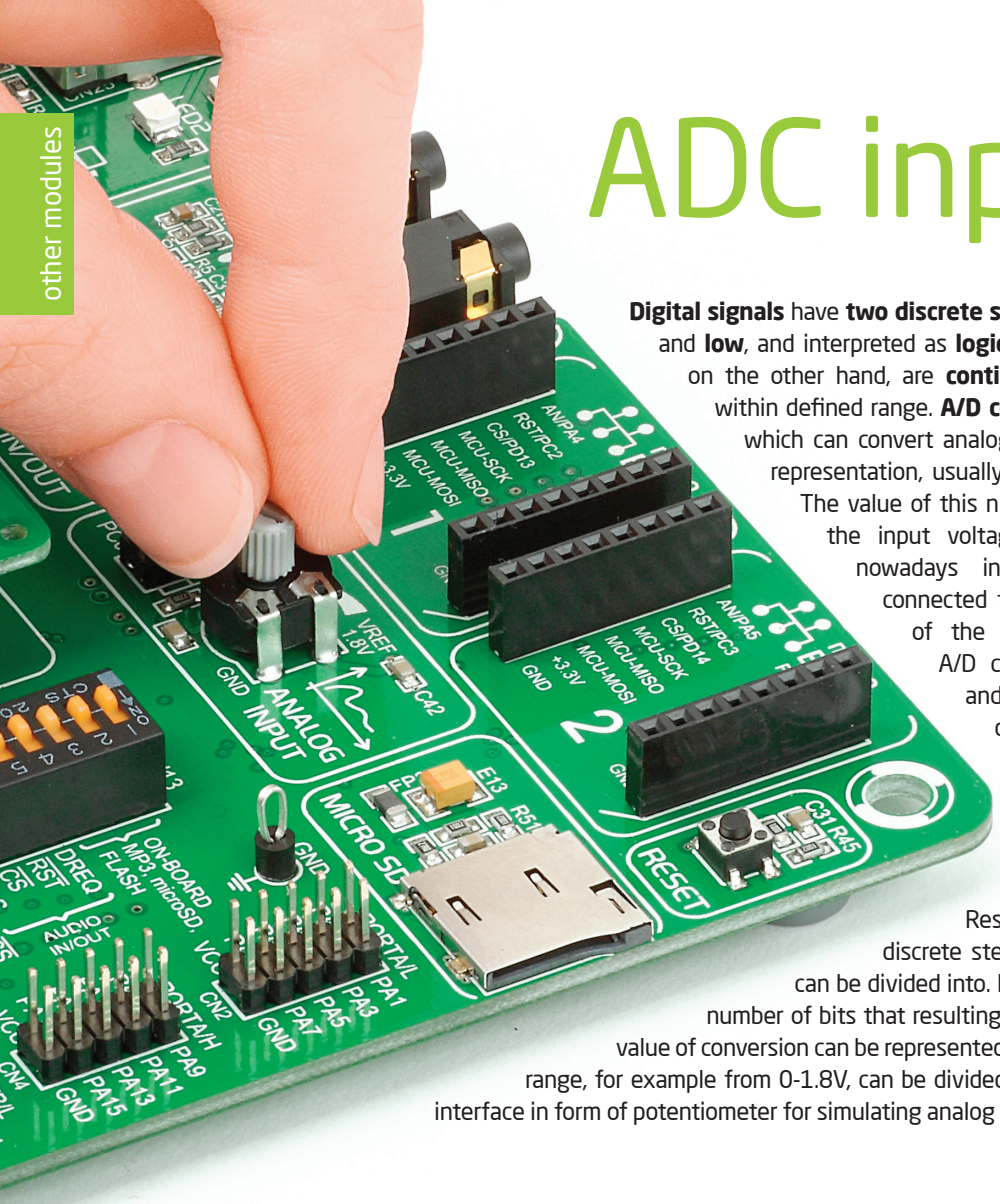


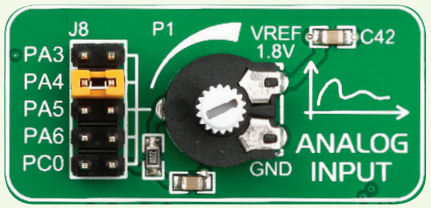
Figure 23-1: Schematic of I²C EEPROM module

ADC inputs



Digital signals have **two discrete states**, which are decoded as **high** and **low**, and interpreted as **logic 1** and **logic 0**. **Analog signals**, on the other hand, are **continuous**, and can have any value within defined range. **A/D converters** are specialized circuits which can convert analog signals (voltages) into a digital representation, usually in form of an **integer number**. The value of this number is **linearly dependent** on the input voltage value. Most microcontrollers nowadays internally have A/D converters connected to one or more input pins. Some of the most important parameters of A/D converters are **conversion time** and **resolution**. Conversion time determines how fast can an analog voltage be represented in form of a digital number. This is an important parameter if you need fast data acquisition. The other parameter is resolution. Resolution represents the number of discrete steps that supported voltage range can be divided into. It determines the sensitivity of the A/D converter. Resolution is represented in maximum number of bits that resulting number occupies. Most microcontrollers have 10-bit resolution, meaning that maximum value of conversion can be represented with 10 bits, which converted to integer is 2^{10}=1024. This means that supported voltage range, for example from 0-1.8V, can be divided into 1024 discrete steps of about 1.758mV. EasyMx PRO™ v7 for STM32 provides an interface in form of potentiometer for simulating analog input voltages that can be routed to any of the 5 supported analog input pins.

Enabling ADC inputs



In order to connect the output of the potentiometer **P1** to **PA3**, **PA4**, **PA5**, **PA6** or **PC0** analog microcontroller inputs, you have to place the jumper **J8** in the desired position. By moving the potentiometer knob, you can create voltages in range from **GND** to **VCC**.

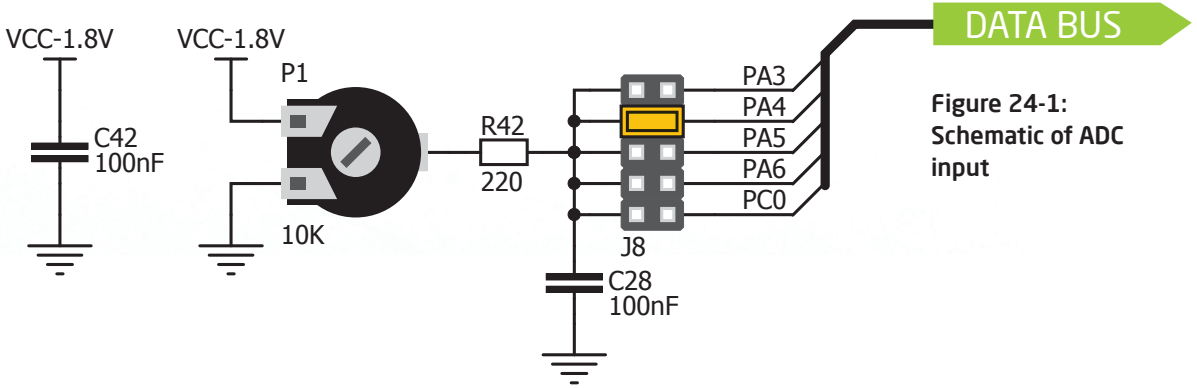
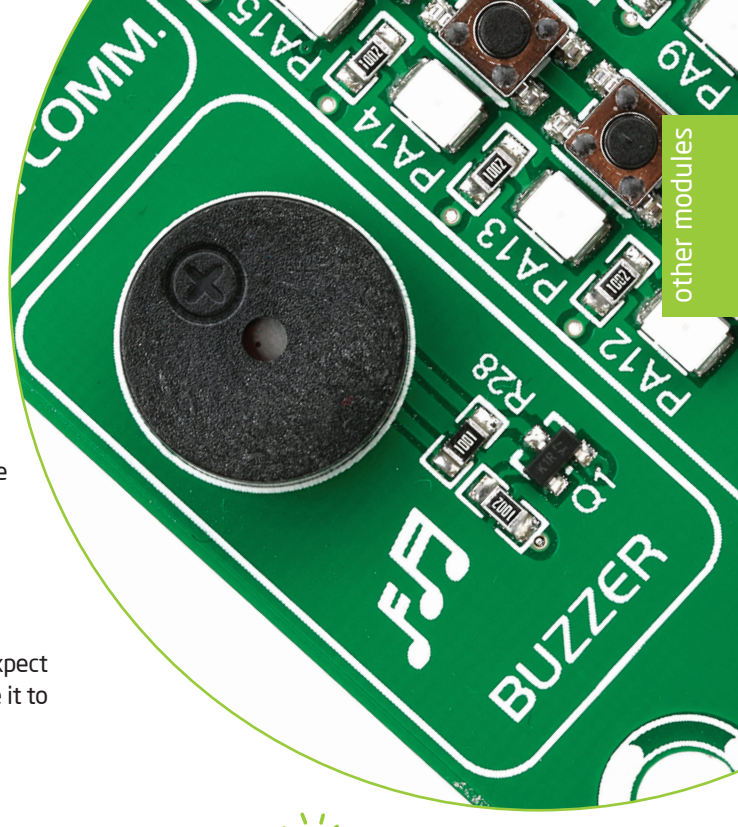


Figure 24-1: Schematic of ADC input

Piezo Buzzer



Piezo electricity is the charge which accumulates in certain solid materials in response to mechanical pressure, but also providing the charge to the piezo electric material causes it to physically deform. One of the most widely used applications of piezo electricity is the production of sound generators, called piezo buzzers. **Piezo buzzer** is an electric component that comes in different shapes and sizes, which can be used to create sound waves when provided with analog electrical signal. EasyMx PRO™ v7 for STM32 comes with piezo buzzer which can be connected to **PE14** microcontroller pin. Connection is established using **SW14.8** DIP switch. Buzzer is driven by transistor **Q1** (**Figure 25-1**). Microcontrollers can create sound by generating a PWM (Pulse Width Modulated) signal - a **square wave** signal, which is nothing more than a sequence of logic zeros and ones. Frequency of the square signal determines the pitch of the generated sound, and duty cycle of the signal can be used to increase or decrease the volume in the range from 0% to 100% of the duty cycle. You can generate PWM signal using hardware capture-compare module, which is usually available in most microcontrollers, or by writing a custom software which emulates the desired signal waveform.

Supported sound frequencies

Piezo buzzer's resonant frequency (where you can expect it's best performance) is **3.8kHz**, but you can also use it to create sound in the range between **2kHz** and **4kHz**.

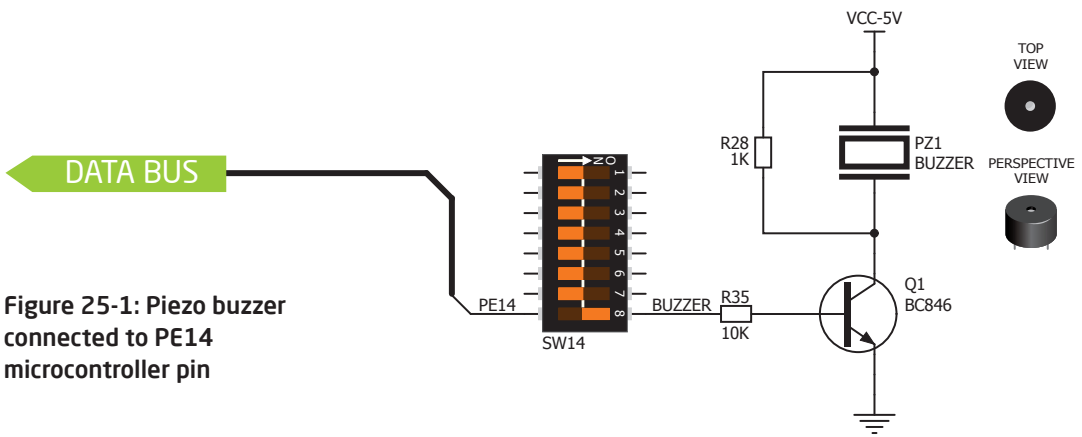
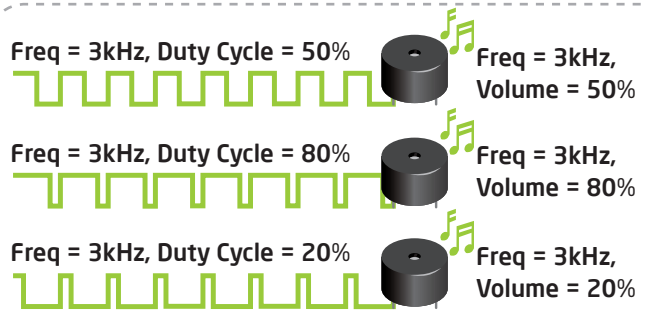


Figure 25-1: Piezo buzzer connected to PE14 microcontroller pin



How to make it sing?

Buzzer starts "singing" when you provide PWM signal from the microcontroller to the buzzer driver. The pitch of the sound is determined by the frequency, and amplitude is determined by the duty cycle of the PWM signal.

Enabling Piezo Buzzer

In order to use the on-board Piezo Buzzer in your application, you first have to connect the transistor driver of piezo buzzer to the appropriate microcontroller pin. This is done using **SW14.8** DIP switch which connects it to **PE14** pin.

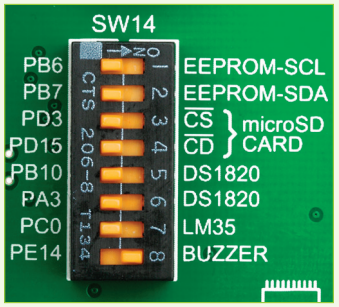


Figure 25-2: push SW14.8 to ON position to connect Piezo buzzer to PE14

Additional GNDs

EasyMx PRO™ v7 for STM32 contains **GND pins** located in different sections of the board, which allow you to easily connect oscilloscope GND reference when you monitor signals on microcontroller pins, or signals of on-board modules.

1. GND is located below microSD section.
2. GND is located just above PORTE/H Input/Output Group.
3. GND is located below power supply region.

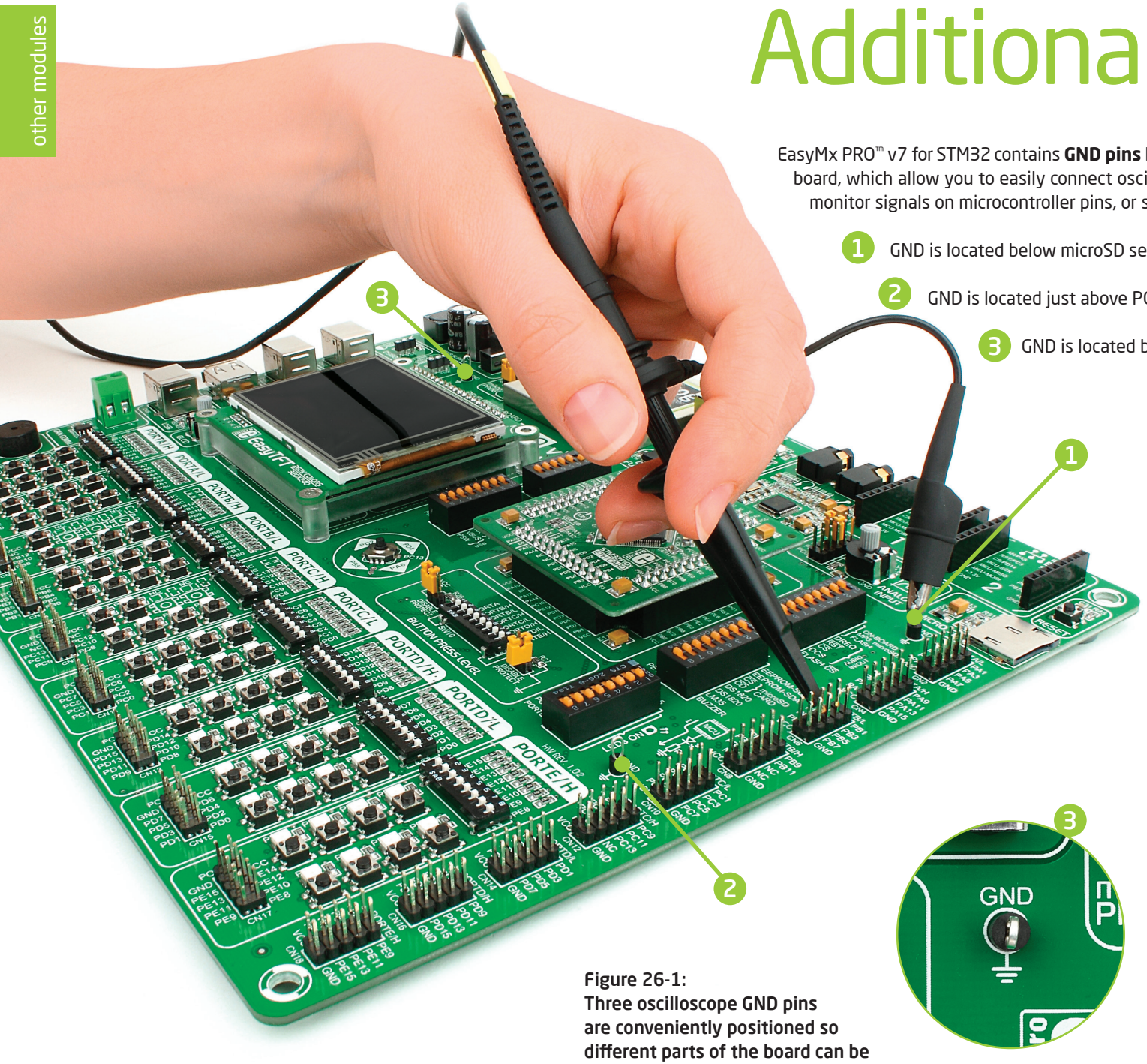


Figure 26-1:
Three oscilloscope GND pins are conveniently positioned so different parts of the board can be reached with an oscilloscope probe

What's Next?

You have now completed the journey through each and every feature of **EasyMx PRO™ v7 for STM32** board. You got to know it's modules, organization, supported microcontrollers, programmer and debugger. Now you are ready to start using your new board. We are suggesting several steps which are probably the best way to begin. We invite you to join the users of EasyMx PRO™ brand. You will find very useful projects and tutorials and can get help from a large ecosystem of users. Welcome!

1 Compiler

You still don't have an appropriate compiler? Locate **ARM® compiler** that suits you best on our website:

http://www.mikroe.com/arm/compilers/

Choose between **mikroC™**, **mikroBasic™** and **mikroPascal™** and download fully functional demo version, so you can begin building your ARM® Cortex™-M3 and Cortex™-M4 applications.

2 Projects

Once you have chosen your compiler, and since you already got the board, you are ready to start writing your first projects. We have equipped our compilers with dozens of examples that demonstrate the use of each and every feature of the **EasyMx PRO™ v7 for STM32** board, and all of our accessory boards as well. This makes an excellent starting point for your future projects. Just load the example, read well commented code, and see how it works on hardware. Browse through the compiler **Examples** on this link:

http://www.mikroe.com/easymx-pro/stm32/

3 Community

If you want to find answers to your questions on many interesting topics we invite you to visit our forum at **http://www.mikroe.com/forum** and browse through more than 150 thousand posts. You are likely to find just the right information for you. On the other hand, if you want to download free projects and libraries, or share your own code, please visit the **Libstock website**. With user profiles, you can get to know other programmers, and subscribe to receive notifications on their code.

http://www.libstock.com/

4 Support

We all know how important it is that we can rely on someone in moments when we are stuck with our projects, facing a deadline, or when we just want to ask a simple, basic question, that's pulling us back for a while. We do understand how important this is to people and therefore our Support Department is one of the pillars upon which our company is based. MikroElektronika offers **Free Tech Support** to the end of product lifetime, so if something goes wrong, we are ready and willing to help!

http://www.mikroe.com/support/

notes

DISCLAIMER

All the products owned by MikroElektronika are protected by copyright law and international copyright treaty. Therefore, this manual is to be treated as any other copyright material. No part of this manual, including product and software described herein, must be reproduced, stored in a retrieval system, translated or transmitted in any form or by any means, without the prior written permission of MikroElektronika. The manual PDF edition can be printed for private or local use, but not for distribution. Any modification of this manual is prohibited.

MikroElektronika provides this manual 'as is' without warranty of any kind, either expressed or implied, including, but not limited to, the implied warranties or conditions of merchantability or fitness for a particular purpose.

MikroElektronika shall assume no responsibility or liability for any errors, omissions and inaccuracies that may appear in this manual. In no event shall MikroElektronika, its directors, officers, employees or distributors be liable for any indirect, specific, incidental or consequential damages (including damages for loss of business profits and business information, business interruption or any other pecuniary loss) arising out of the use of this manual or product, even if MikroElektronika has been advised of the possibility of such damages. MikroElektronika reserves the right to change information contained in this manual at any time without prior notice, if necessary.

HIGH RISK ACTIVITIES

The products of MikroElektronika are not fault-tolerant nor designed, manufactured or intended for use or resale as on-line control equipment in hazardous environments requiring fail-safe performance, such as in the operation of nuclear facilities, aircraft navigation or communication systems, air traffic control, direct life support machines or weapons systems in which the failure of Software could lead directly to death, personal injury or severe physical or environmental damage ('High Risk Activities'). MikroElektronika and its suppliers specifically disclaim any expressed or implied warranty of fitness for High Risk Activities.

TRADEMARKS

The MikroElektronika name and logo, the MikroElektronika logo, mikroC™, mikroBasic™, mikroPascal™, mikroProg™, mikromedia™, EasyARM™, EasyMx PRO™, Click boards™ and mikroBUS™ are trademarks of MikroElektronika. All other trademarks mentioned herein are property of their respective companies.

All other product and corporate names appearing in this manual may or may not be registered trademarks or copyrights of their respective companies, and are only used for identification or explanation and to the owners' benefit, with no intent to infringe.

If you want to learn more about our products, please visit our website at www.mikroe.com
If you are experiencing some problems with any of our products or just need additional information, please place your ticket at www.mikroe.com/support
If you have any questions, comments or business proposals, do not hesitate to contact us at office@mikroe.com

EasyMx PRO v7 for STM32® Manual
ver. 1.02b
0 100000 024768