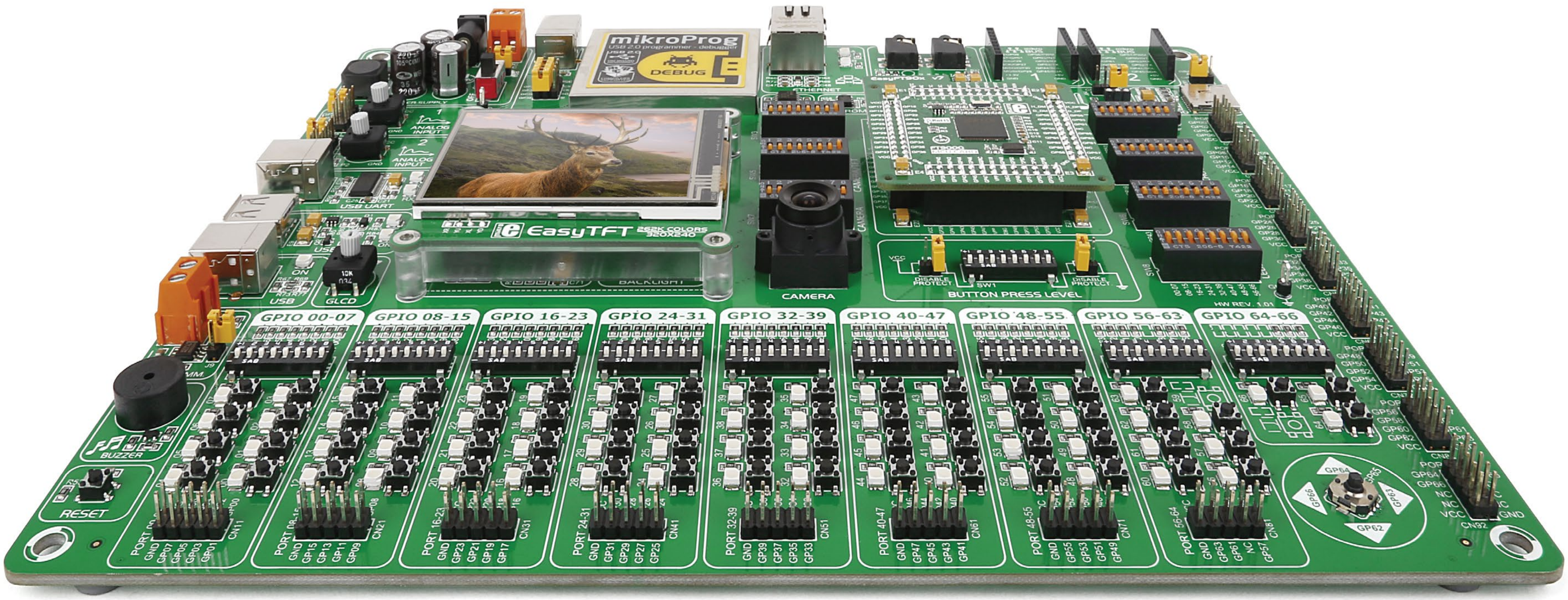
EasyFT90x v7



CMOS image sensor
Onboard camera

Many on-board modules
Multimedia peripherals

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

EasyFT90x v7 is our first development board for FT90x, a new and exciting 32-bit MCU architecture from FTDI Chip. We put all our expertise into designing it. Our team relied on tried and tested concepts that define our range of "Easy" boards, but we also incorporated innovations that utilize some standout features specific to FT90x.

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

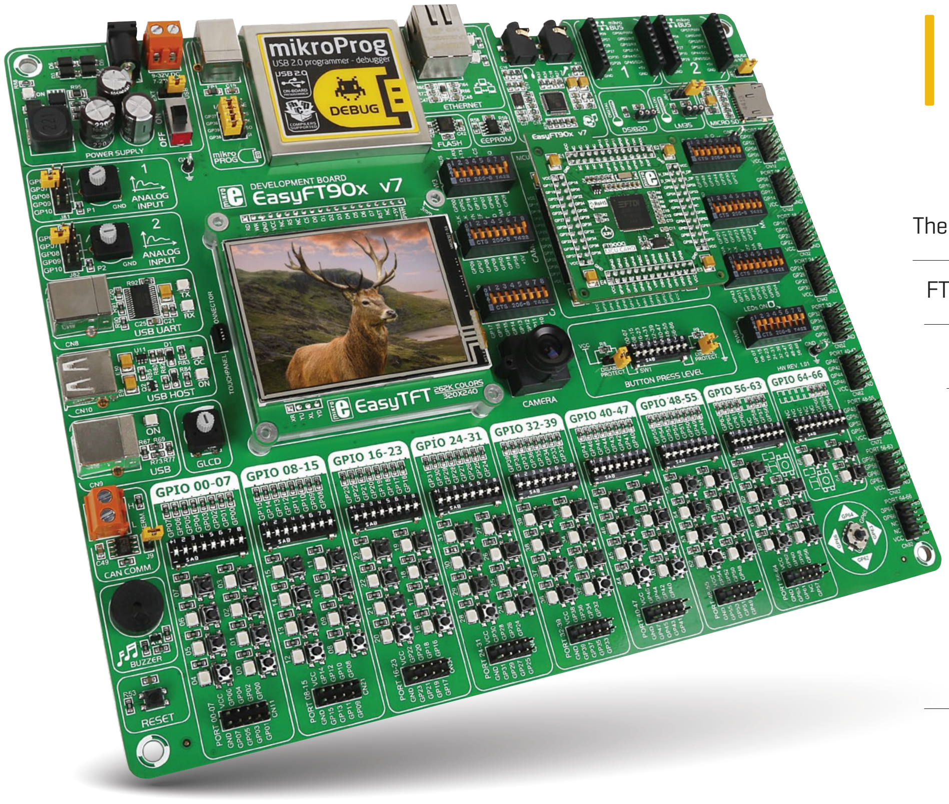
Programmer/debugger

Connectivity

Communication

Multimedia

Other modules



Introduction

The FT90x family of MCUs is based on FTDI Chip’s proprietary 32-bit RISC core called FT32. The most exciting feature of this new architecture is its ability to execute instructions from Shadow RAM. This enables FT90x MCUs to operate at zero wait states at up to 100Mhz, which translates to 310DMIPS. Performance-wise, this places FT90x high above many MCUs with similar clock rates. The high speed also made it possible to integrate a camera interface into FT90x. Taking advantage of that feature, EasyFT90x is our first “Easy” boards that incorporates a camera along with all other modules and connectors.

EasyFT90x v7 development Team

Supporting video

Onboard camera

EasyFT90x is the first of our development boards with a camera, taking advantage of the camera interface of FT90x MCUs. The board also boasts a multitude of other multimedia features.

Everything is already here

mikroProg™ on board

Powerful on-board mikroProg™ programmer and In-Circuit debugger supports all existing FT90x MCUs. It features fast enhanced programming and rich set of debugging instructions.

Two connectors for each port

Amazing connectivity

EasyFT90x v7 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.

For easier connections

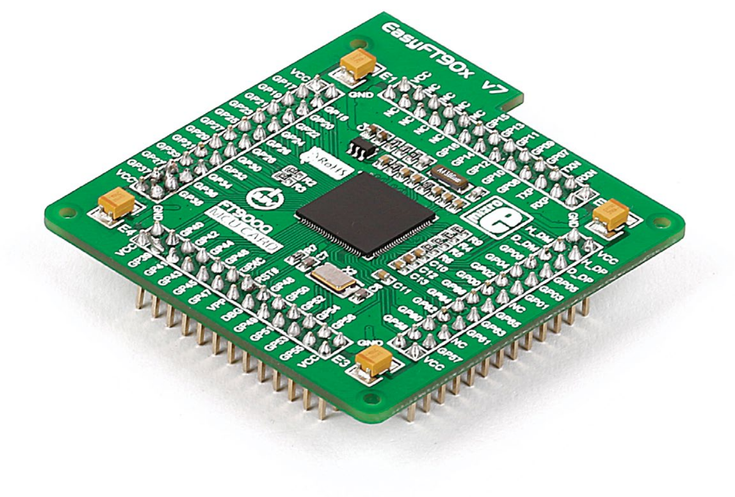
mikroBUS™ support

This innovative socket allows you to use dozens of click add-on boards with almost no hardware adjustments. Adding new functionality to your device was never so easy.

It's good to know

FT900 - default microcontroller

FT900 is the default chip of EasyFT90 v7. It operates at a 100MHz, has 256K bytes of program/shadow memory, 64K bytes of data memory, a parallel camera interface, USB2.0 Hi-Speed (480Mbps) host controller with BCD emulation, I^{2}S master/slave interface (24.57/22.57MHz), 65 General purpose I/O pins, 4x16-bit timers with prescale and watchdog funtion, two 10 bit DACs, 2 programmable UARTs, 7xpwm, master/ slave I^{2}C, one master and two slave SPI, and 2xCAN controllers.



- *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
~72-80mA when all peripheral modules are disconnected

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

weight
~500g
(1.1 lbs)

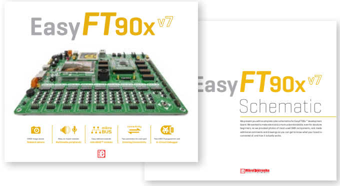
Package contains

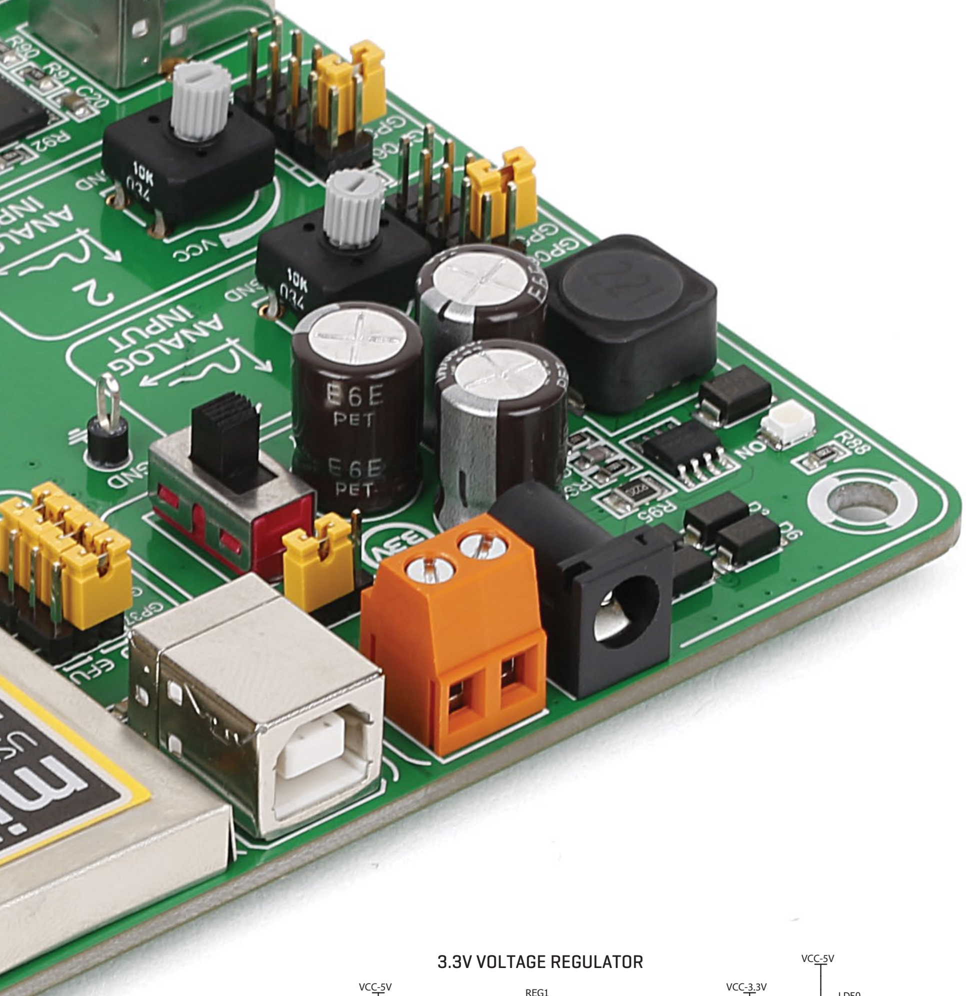
1. Damage resistant protective box

2. EasyFT90x v7 board in antistatic bag

3. USB cable

4. User manual and board schematics





Power supply

The board contains a 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 (CN7)**, using external adapters via adapter connector **(CN34)** or additional screw terminals **(CN35)**. External adapter voltage levels must be in range of **9-32V DC and 7-23V AC**. Use jumper **J1** to specify which power source you are using. Upon providing the power using either external adapters or USB power source you can turn on the power with **SWITCH 1 (Figure 3-1)**. Power **LED ON** indicates the presence of a power supply.

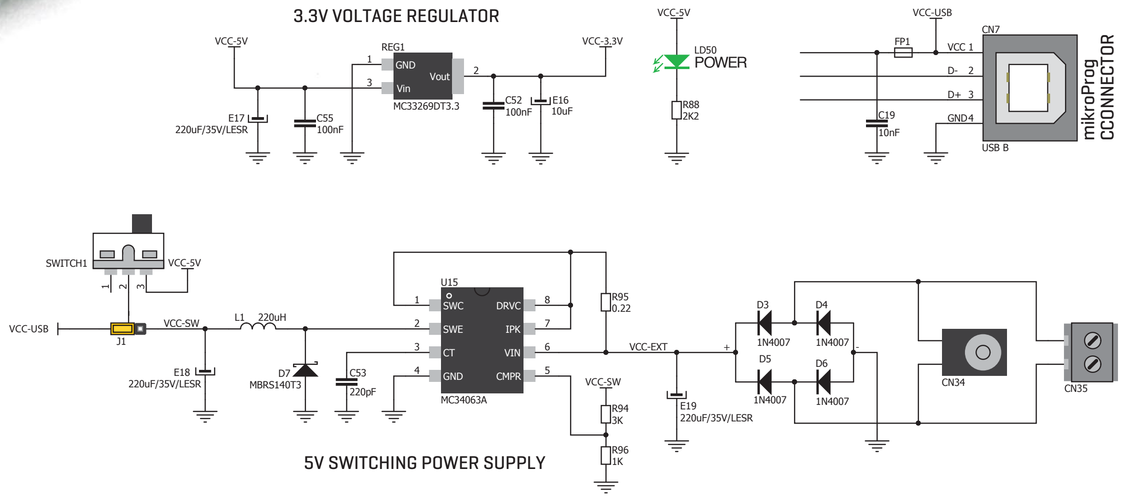


Figure 3-2: Power supply unit schematic

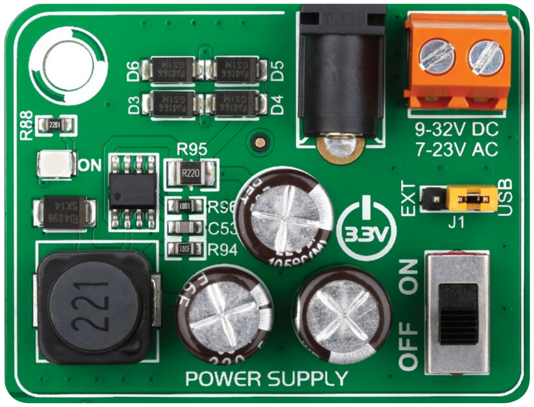


Figure 3-1: Power supply unit of EasyFT90x v7

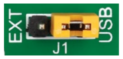
3.3V

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 1500mA with external power supply

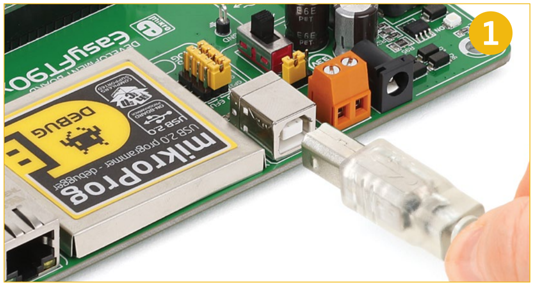
How to power the board?

1. With USB cable

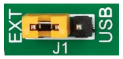


Set J1 jumper to USB position

To power the board with a USB cable, place jumper **J1** in USB position. You can then plug in the USB cable as shown on images 1 and 2, and turn the power switch ON.

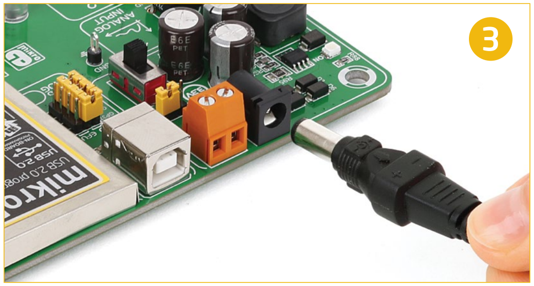


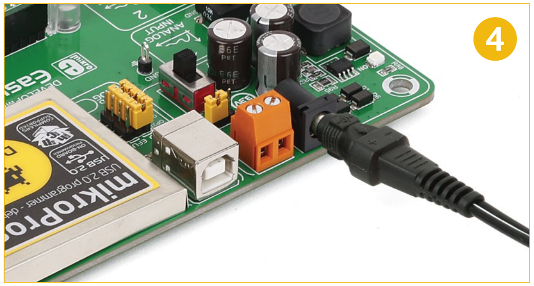
2. Using adapter



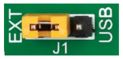
Set J1 jumper to EXT position

To power the board via adapter connector, place jumper **J1** in EXT position. You can then plug in the adapter cable as shown on images 3 and 4, and turn the power switch ON.



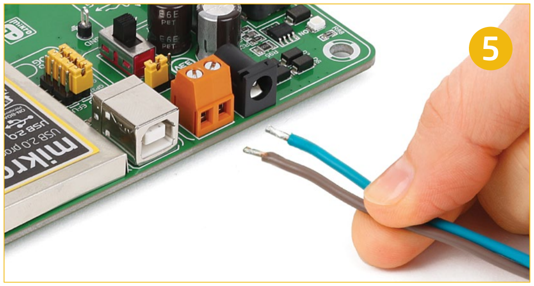


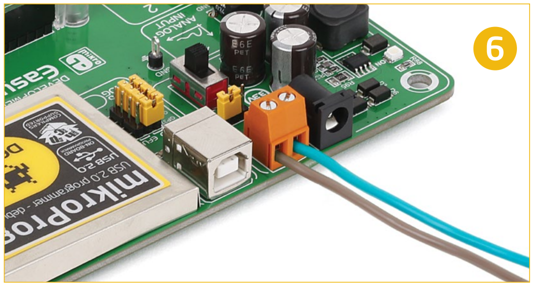
3. With laboratory power supply



Set J1 jumper to EXT position

To power the board using screw terminals, place jumper **J1** 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.

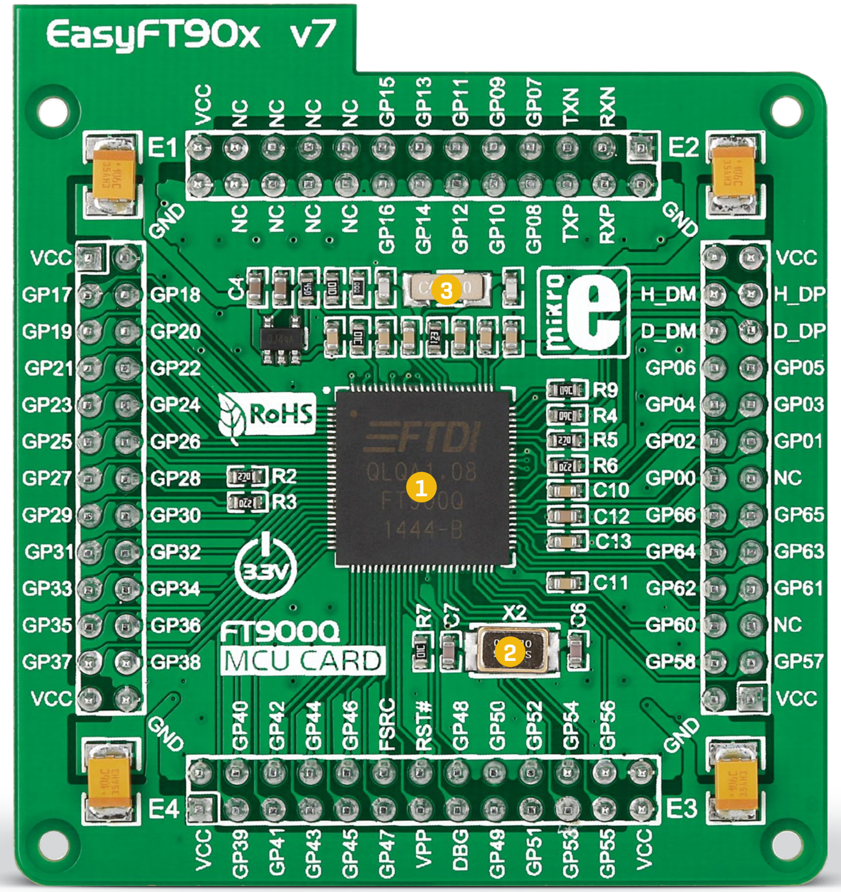




MCU cards

Microcontrollers are supported using specialized MCU cards containing 104 pin, which are placed into the on-board female MCU socket (Figure 4.1). The default card contains the FT900 microcontroller with on-chip peripherals. After testing and building the final program on the development board, this card can also be taken out of the socket and used in your final device.

Figure 4-1:
MCU card
with FT900



1 FT900 has 100MHz maximum frequency, 256KB of Flash, 64KB of on-chip data memory and 256KB of Shadow program memory. It has an integrated Parallel Camera Interface, Ethernet, USB (Host, Device), 65 General purpose I/O pins, 4 x 16-bit timers with 32-bit watchdog function, 7 Analog Input pins (ADC), 2 UARTs, I^{2}S master/slave interface, internal slow clock oscillator, 2 x I^{2}C (master/slave), SPI (master/slave) and 2xCAN controllers, and a debug interface.

2 12MHz crystal oscillator. We carefully chose the most convenient crystal value that provides clock frequency for the PLL multipliers.

3 MCU card also contains 32.768 kHz crystal oscillator which provides external clock waveform for the internal RTCC module.

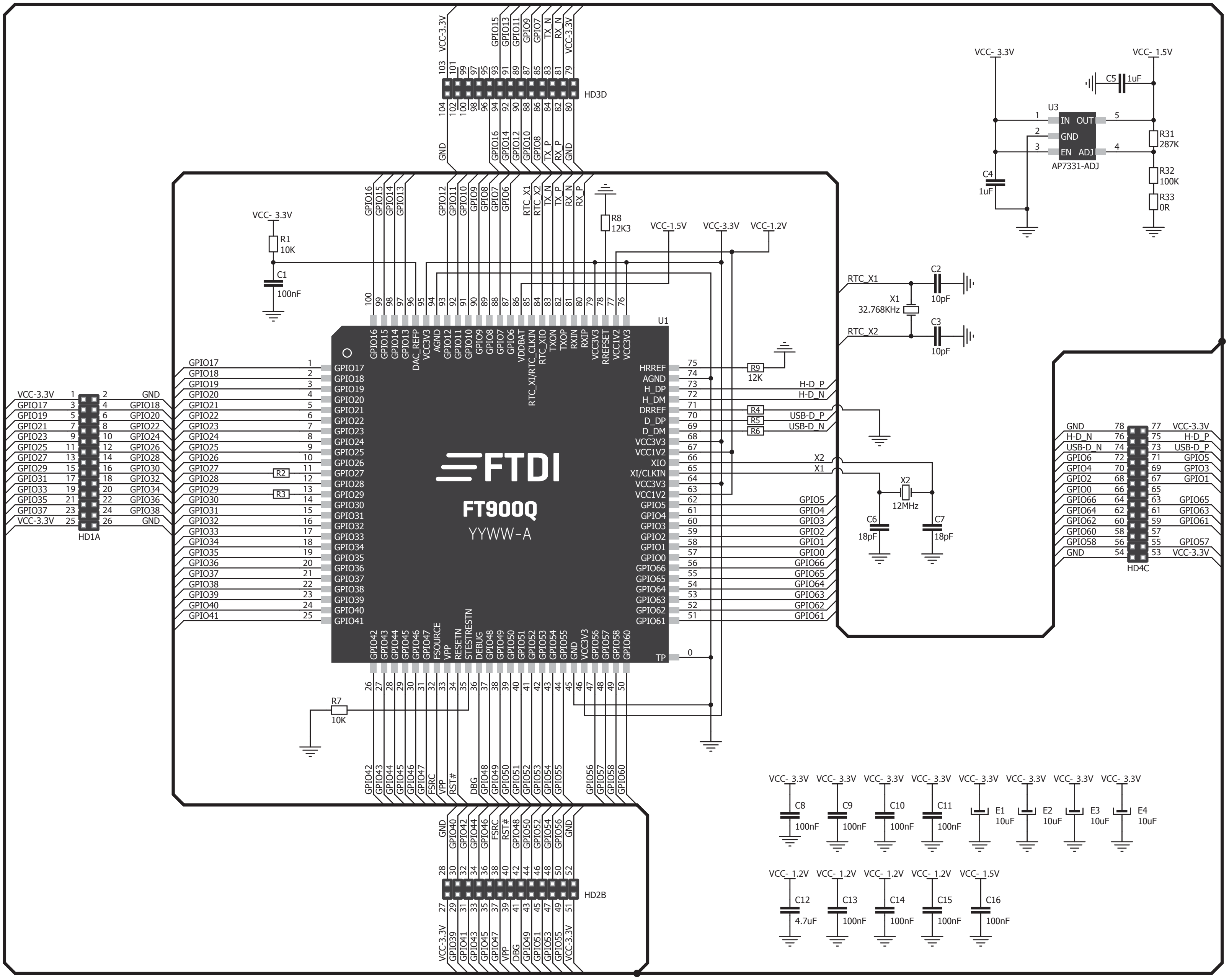


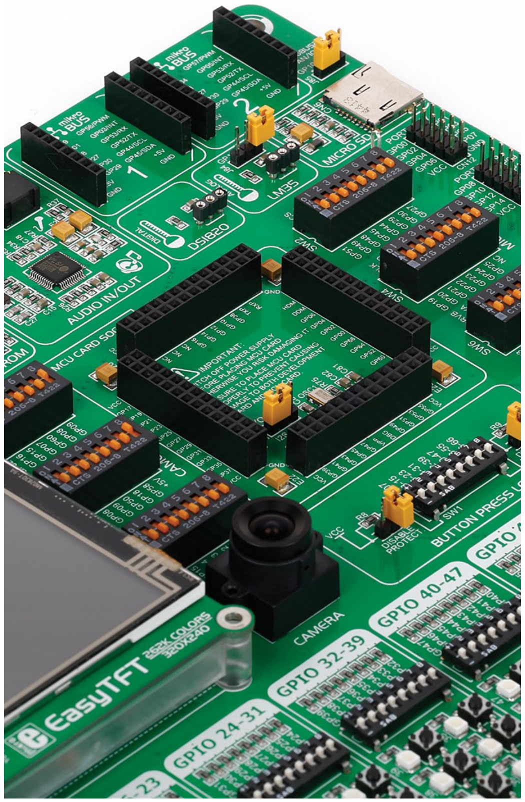
Figure 4-2: MCU card schematic

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 MCU card. First make sure that MCU card orientation matches the silkscreen outline on the EasyFT90x v7 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 can you turn on the power supply.

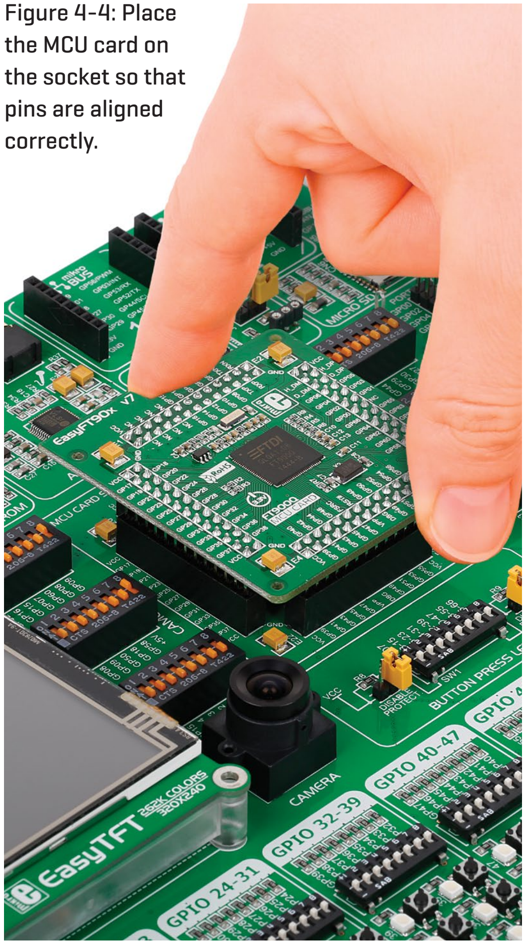
1

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



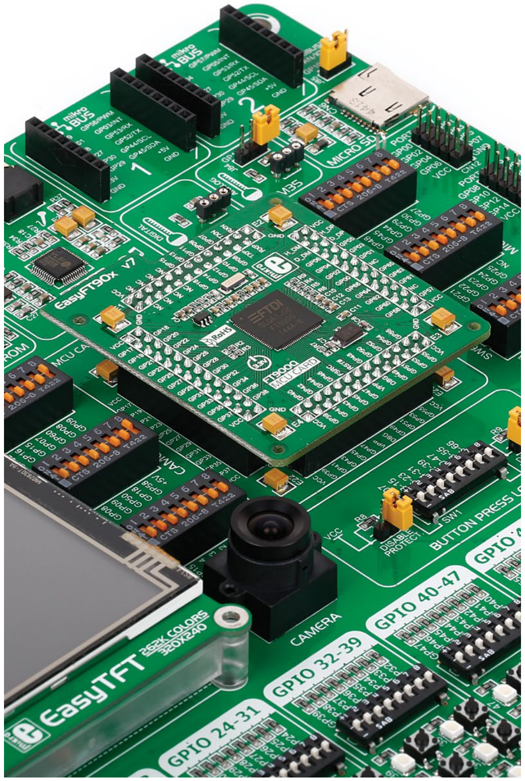
2

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



3

Figure 4-5 Properly placed MCU card.



On-board programmer

What is mikroProg™?

mikroProg™ is a fast programmer and debugger for FT90x devices, the first of its kind. The programmer supports all currently available FT90x devices. 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.

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

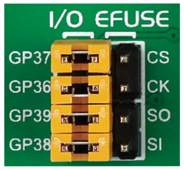
- Install programmer drivers
- Install mikroProg Suite™ for FT90x software

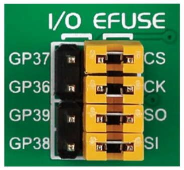
2. Power up the board, and you are ready to go.

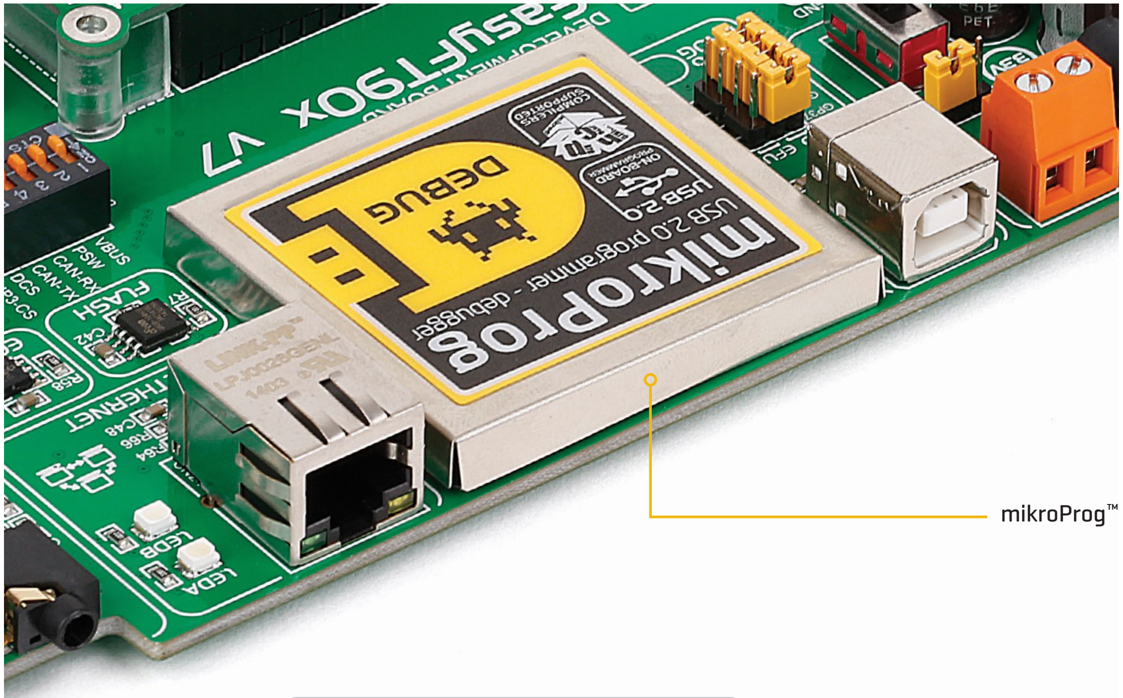
- Plug in the programmer USB cable
- LINK LED should light up.

mikroProg™ or eFUSE

In addition to mikroProg™, EasyFT90x allows eFUSE programming. eFUSE permanently programs the chip, disabling further altercations. eFUSE is chiefly used when the microcontroller is to be installed in a final device, for security reasons. To enable this method of programming, switch the four jumpers under the mikroPROG™ USB port into the EFUSE position.







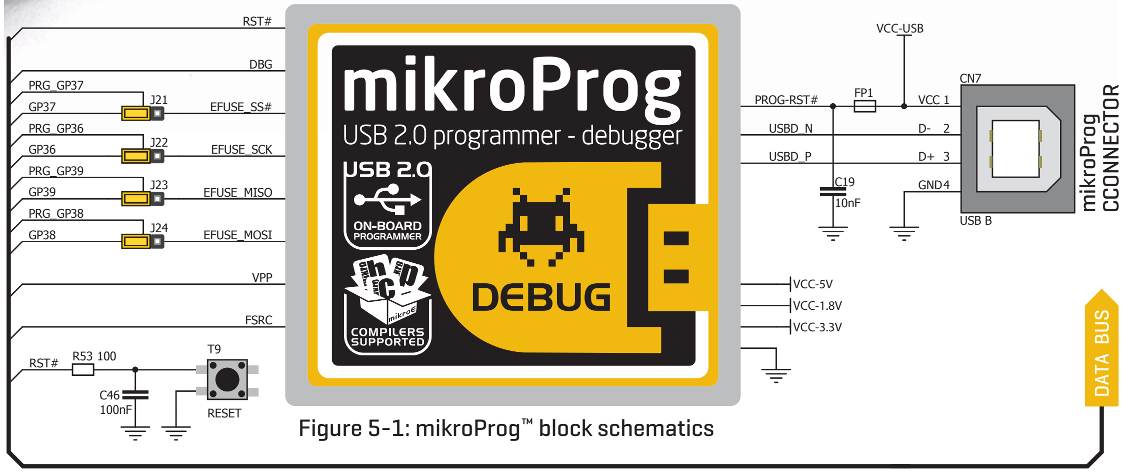


Figure 5-1: mikroProg™ block schematics

Installing programmer drivers

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

www.mikroe.com/downloads/get/2216/mikroprog_suite_for_ft90x_drivers.zip

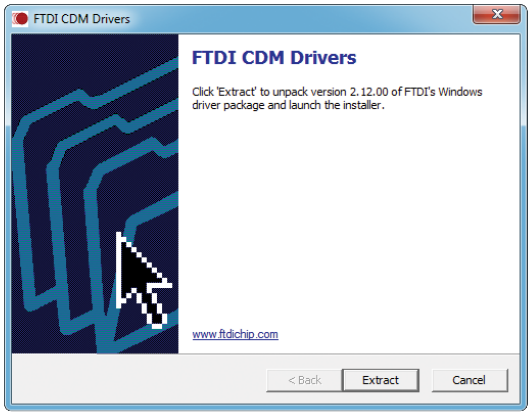
When you locate the drivers, please extract the setup file from the ZIP archive. You should be able to locate the driver setup file. Double click the setup file to begin installation of the programmer drivers.

mikroprog_suite_for_ft90x_drivers.zip
WinRAR ZIP archive

▼

CDM v2.12.00 WHQL Certified.exe
Type: Application

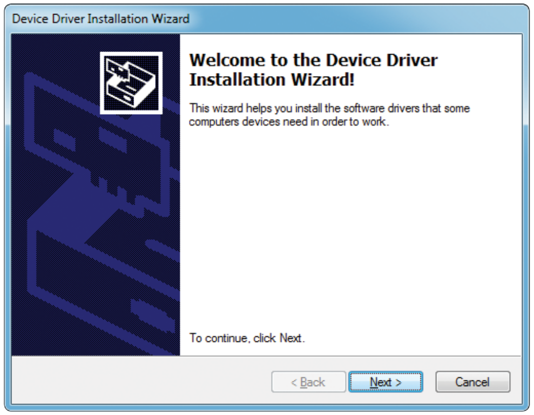
▼


FTDI CDM Drivers

FTDI CDM Drivers

Click 'Extract' to unpack version 2.12.00 of FTDI's Windows driver package and launch the installer.

www.ftdichip.com


Device Driver Installation Wizard

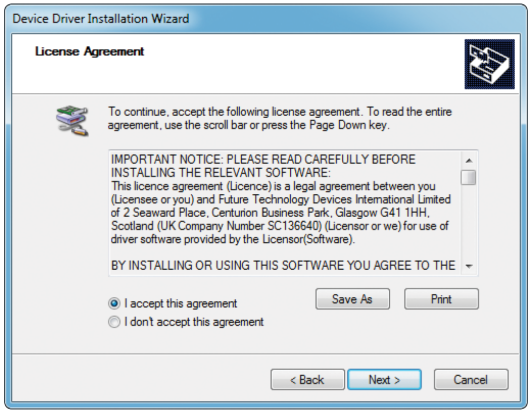
Welcome to the Device Driver Installation Wizard!

This wizard helps you install the software drivers that some computers devices need in order to work.

To continue, click Next.

Step 1 - Start Installation

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


Device Driver Installation Wizard

License Agreement

To continue, accept the following license agreement. To read the entire agreement, use the scroll bar or press the Page Down key.

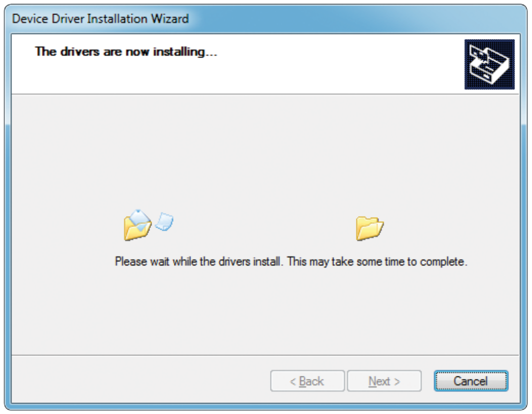
IMPORTANT NOTICE: PLEASE READ CAREFULLY BEFORE INSTALLING THE RELEVANT SOFTWARE:
This licence agreement (Licence) is a legal agreement between you (Licensee or you) and Future Technology Devices International Limited of 2 Seaward Place, Centurion Business Park, Glasgow G41 1HH, Scotland (UK Company Number SC136640) (Licensor or we) for use of driver software provided by the Licensor(Software).

BY INSTALLING OR USING THIS SOFTWARE YOU AGREE TO THE

● I accept this agreement
○ I don't accept this agreement

Step 2 - Select Destination

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

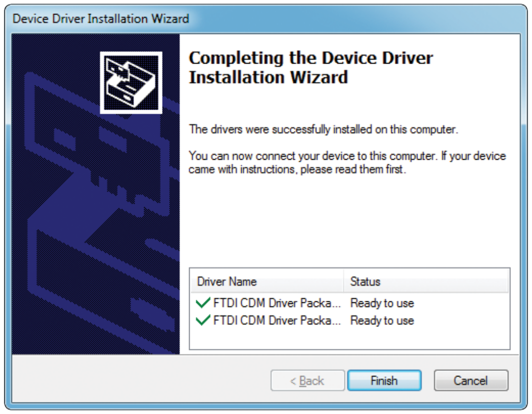

Device Driver Installation Wizard

The drivers are now installing...

Please wait while the drivers install. This may take some time to complete.

Step 3 - Installing drivers

Drivers are installed automatically in a matter of seconds.


Device Driver Installation Wizard

Completing the Device Driver Installation Wizard

The drivers were successfully installed on this computer.

You can now connect your device to this computer. If your device came with instructions, please read them first.

Driver Name	Status
✓ FTDI CDM Driver Packa...	Ready to use
✓ FTDI CDM Driver Packa...	Ready to use

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 FT90x

A standalone programming software utility called **mikroProg Suite™ for FT90x** is available as an alternative to programming the MCU directly from the FT90x compiler. This software is used for programming of all supported FT90x microcontrollers. The software has an intuitive interface and **SingleClick™** programming technology. To begin, first locate the installation archive on our web site:

www.mikroe.com/downloads/get/2215/mikroprog_suite_ft90x_v100.zip

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

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 and drop the file onto the software window, or use the **Load** button to open the **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.

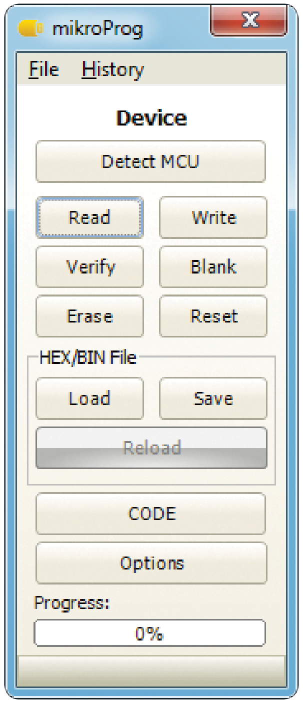
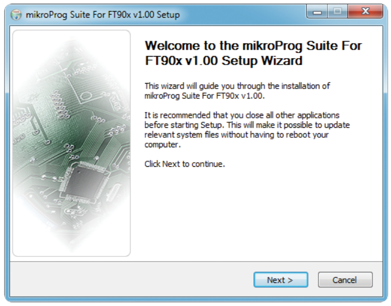


Figure 5-2: mikroProg Suite™ window

Installation wizard - 6 simple steps



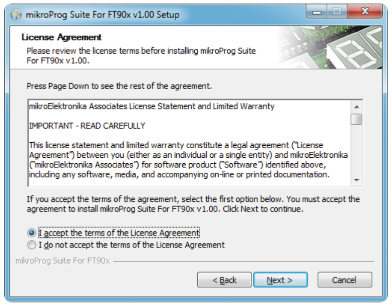
Welcome to the mikroProg Suite For FT90x v1.00 Setup Wizard

This wizard will guide you through the installation of mikroProg Suite For FT90x v1.00.

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


mikroProg Suite For FT90x v1.00 Setup
mikroProg Suite For FT90x
< Back
Next >
Cancel

License Agreement

Please review the license terms before installing mikroProg Suite For FT90x v1.00.

Press Page Down to see the rest of the agreement.

mikroElektronika Associates License Statement and Limited Warranty

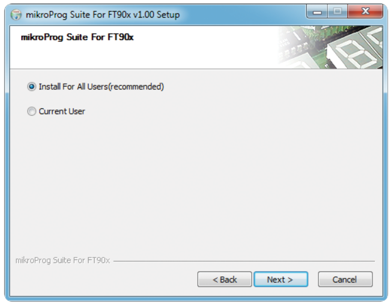
IMPORTANT - READ CAREFULLY

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 FT90x v1.00. Click Next to continue.

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

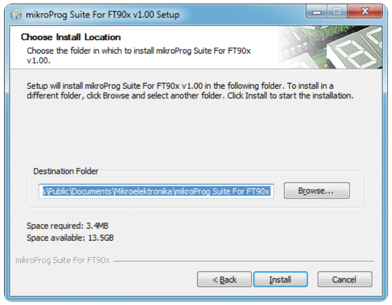
Step 2 - Accept EULA and continue


mikroProg Suite For FT90x v1.00 Setup
mikroProg Suite For FT90x
< Back
Next >
Cancel

mikroProg Suite For FT90x

● Install For All Users(recommended)
○ Current User

Step 3 - Install for All users or current user


mikroProg Suite For FT90x v1.00 Setup
Browse...
mikroProg Suite For FT90x
< Back
Install
Cancel

Choose Install Location

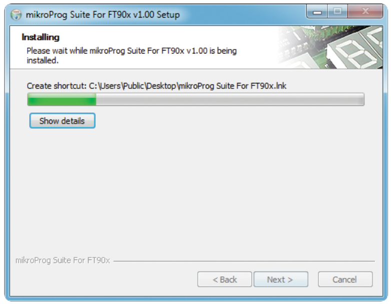
Choose the folder in which to install mikroProg Suite For FT90x v1.00.

Setup will install mikroProg Suite For FT90x v1.00 in the following folder. To install in a different folder, click Browse and select another folder. Click Install to start the installation.

Destination Folder: s\Public\Documents\Mikroelektronika\mikroProg Suite For FT90x

Space required: 3.4MB
Space available: 13.5GB

Step 4 - Choose destination folder

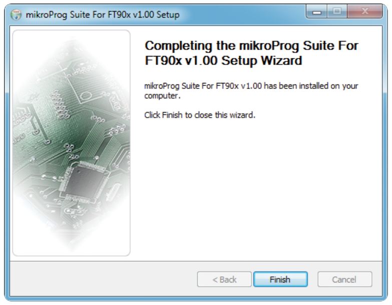

mikroProg Suite For FT90x v1.00 Setup
Show details
mikroProg Suite For FT90x
< Back
Next >
Cancel

Installing

Please wait while mikroProg Suite For FT90x v1.00 is being installed.

Create shortcut: C:\Users\Public\Desktop\mikroProg Suite For FT90x.lnk

Step 5 - Installation in progress



Completing the mikroProg Suite For FT90x v1.00 Setup Wizard

mikroProg Suite For FT90x v1.00 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 planned. 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?

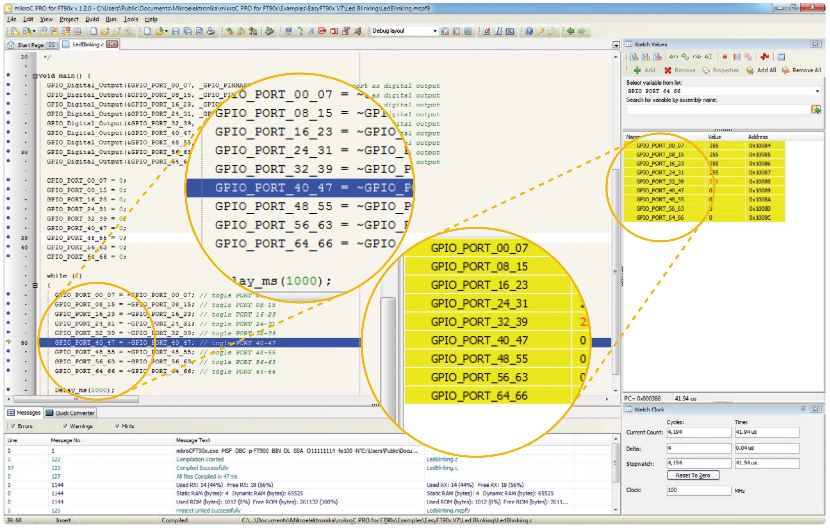
The on-board **mikroProg**™ programmer supports a **hardware debugger** — a highly effective tool for **Real-Time debugging** on hardware level. The debugger enables you to execute your program on the host FT90x microcontroller and view variable values, Special Function Registers (SFR), RAM, CODE along with the code execution on hardware. Whether you are a beginner, or a professional, this powerful tool, with an 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 FT90x natively support mikroProg™ for FT90x. 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 FT90x** 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 FT90x v.1.2.0 - C:\Users\Public\Documents\Mikroelektronika\mikroC PRO for FT90x\Examples\EasyFT90x V7\Led Blinking\LedBlinking.mcpf9
File Edit View Project Build Run Tools Help
Debug layout
Start Page
LedBlinking.c
Watch Values
Add
Remove
Properties
Add All
Remove All
Select variable from list:
GPIO PORT 64 66
Search for variable by assembly name:
Messages
Quick Converter
☑ Errors
☑ Warnings
☑ Hints
Watch Clock
Reset To Zero

```
*/

void main() {
  GPIO_Digital_Output(&GPIO_PORT_00_07, _GPIO_PINMASK
  GPIO_Digital_Output(&GPIO_PORT_08_15, _GPIO_PI
  GPIO_Digital_Output(&GPIO_PORT_16_23, _GPIO
  GPIO_Digital_Output(&GPIO_PORT_24_31, _GP
  GPIO_Digital_Output(&GPIO_PORT_32_39,
  GPIO_Digital_Output(&GPIO_PORT_40_47,
  GPIO_Digital_Output(&GPIO_PORT_48_55,
  GPIO_Digital_Output(&GPIO_PORT_56_63
  GPIO_Digital_Output(&GPIO_PORT_64_6

  GPIO_PORT_00_07 = 0;
  GPIO_PORT_08_15 = 0;
  GPIO_PORT_16_23 = 0;
  GPIO_PORT_24_31 = 0;
  GPIO_PORT_32_39 = 0;
  GPIO_PORT_40_47 = 0;
  GPIO_PORT_48_55 = 0;
  GPIO_PORT_56_63 = 0;
  GPIO_PORT_64_66 = 0;

  while (1)
  {
    GPIO_PORT_00_07 = ~GPIO_PORT_00_07; // togle PORT 0
    GPIO_PORT_08_15 = ~GPIO_PORT_08_15; // togle PORT 08-15
    GPIO_PORT_16_23 = ~GPIO_PORT_16_23; // togle PORT 16-23
    GPIO_PORT_24_31 = ~GPIO_PORT_24_31; // togle PORT 24-31
    GPIO_PORT_32_39 = ~GPIO_PORT_32_39; // togle PORT 32-39
    GPIO_PORT_40_47 = ~GPIO_PORT_40_47; // togle PORT 40-47
    GPIO_PORT_48_55 = ~GPIO_PORT_48_55; // togle PORT 48-55
    GPIO_PORT_56_63 = ~GPIO_PORT_56_63; // togle PORT 56-63
    GPIO_PORT_64_66 = ~GPIO_PORT_64_66; // togle PORT 64-66

    Delay_ms(1000);
```

Name	Value	Address
GPIO_PORT_00_07	255	0x10084
GPIO_PORT_08_15	255	0x10085
GPIO_PORT_16_23	255	0x10086
GPIO_PORT_24_31	255	0x10087
GPIO_PORT_32_39	255	0x10088
GPIO_PORT_40_47	0	0x10089
GPIO_PORT_48_55	0	0x1008A
GPIO_PORT_56_63	0	0x1008B
GPIO_PORT_64_66	0	0x1008C

PC= 0x000388 41.94 us

	Cycles:	Time:
Current Count:	4,194	41.94 us
Delta:	4	0.04 us
Stopwatch:	4,194	41.94 us
Clock:	100	MHz

Line	Message No.	Message Text	
0	1	mikroCFT90x.exe -MSF -DBG -p FT900 -BIN -DL -SSA -O11111114 -fo100 -N"C:\Users\Public\Docu...	
0	122	Compilation Started	LedBlinking.c
57	123	Compiled Successfully	LedBlinking.c
0	127	All files Compiled in 47 ms	
0	1144	Used RX: 14 (44%) Free RX: 18 (56%)	Used RX: 14 (44%) Free RX: 18 (56%)
0	1144	Static RAM (bytes): 4 Dynamic RAM (bytes): 65525	Static RAM (bytes): 4 Dynamic RAM (bytes): 65525
0	1144	Used ROM (bytes): 1012 (0%) Free ROM (bytes): 261132 (100%)	Used ROM (bytes): 1012 (0%) Free ROM (bytes): 2611...
0	125	Project Linked Successfully	LedBlinking.mcpf9

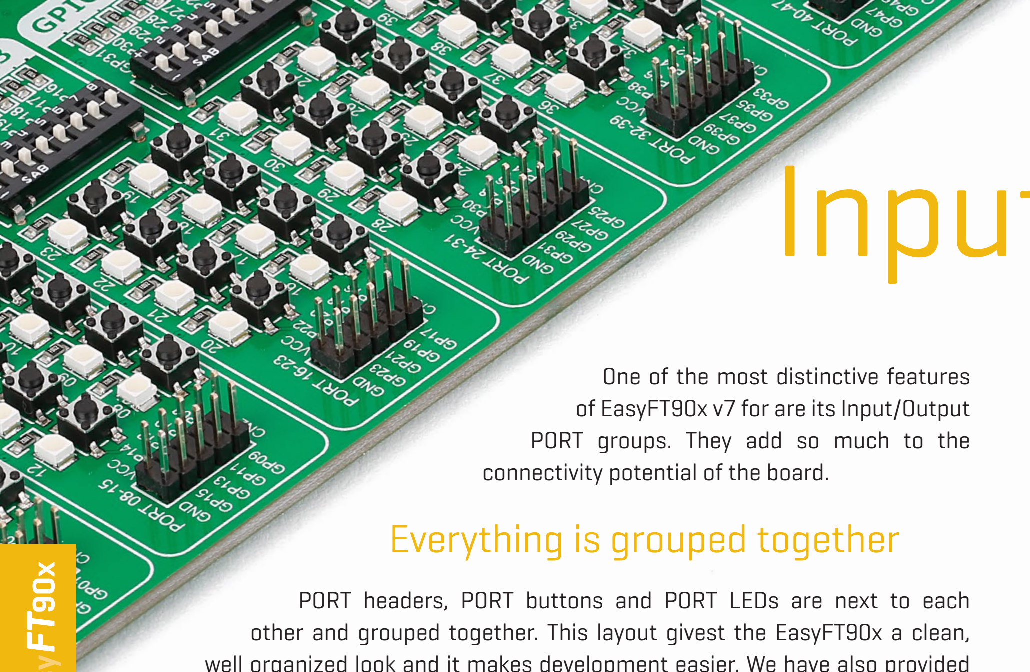
39: 68 Insert Compiled C:\...\Documents\Mikroelektronika\mikroC PRO for FT90x\Examples\EasyFT90x V7\Led Blinking\LedBlinking.c

Figure 5-3:
mikroC PRO for FT90x 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 EasyFT90x v7 for are its Input/Output PORT groups. They add so much to the connectivity potential of the board.

Everything is grouped together

PORT headers, PORT buttons and PORT LEDs are next to each other and grouped together. This layout givest the EasyFT90x a clean, well organized look and it makes development easier. We have also provided 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 **SW11** 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.

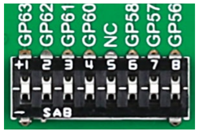


Figure 6-3: Tri-state DIP switch on PORT00-07

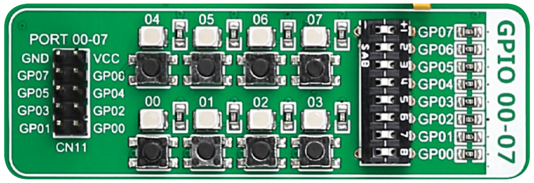
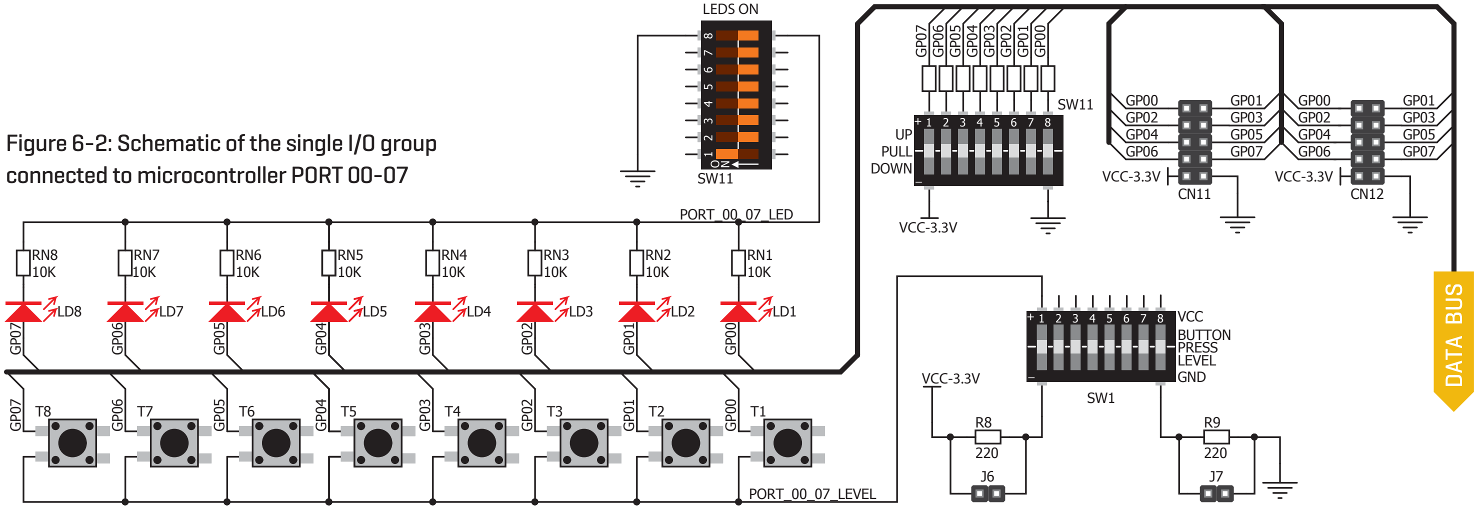


Figure 6-1: I/O group 00-07 contains PORT header, tri-state pull up/down DIP switch, buttons and LEDs all in one place

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 6-2: Schematic of the single I/O group connected to microcontroller PORT 00-07



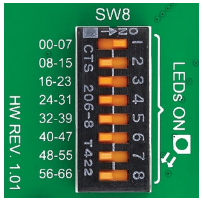
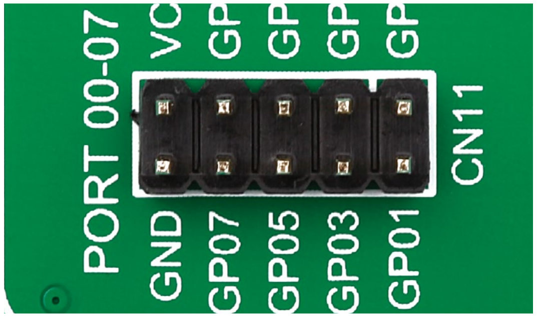


Figure 6-4: Switch SW8.1 enables LEDs on PORT group 00-07

Headers



With enhanced connectivity as one of the key features of EasyFT90x v7, we have provided **two connection headers for each PORT**. I/O PORT group contains one male IDC10 header (like **CN11 shown above**). There is **one more IDC10 header** available on the right side of the board, next to DIP switches (like **CN12** on **Figure 6-4**). These headers can be used to connect accessory boards with IDC10 female sockets.

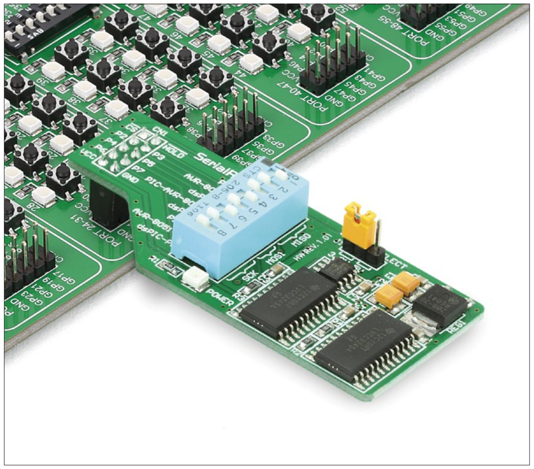
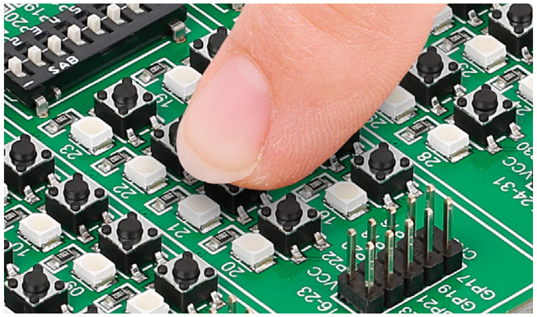


Figure 6-4: IDC10 male headers enable easy connection with MikroElektronika accessory boards

Buttons



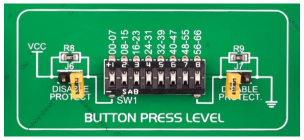


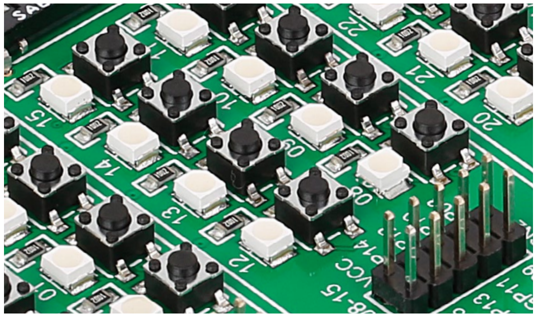
Figure 6-5: Button press level DIP switch (tri-state)

The logic state of all microcontroller digital inputs may be changed using **push buttons**. Tri-state DIP switch **SW11** 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 **SW11.1** in **VCC** position, then pressing any push button on PORT 00-07 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 the MCU in case of wrong usage.

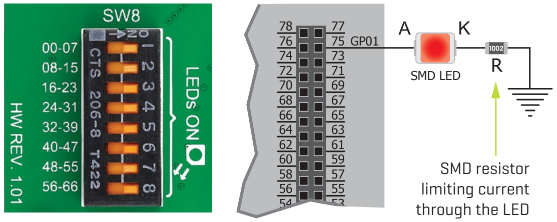
Reset Button

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 EasyFT90x v7 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 **SW8** (**Figure 6-6**).



SMD resistor limiting current through the LED

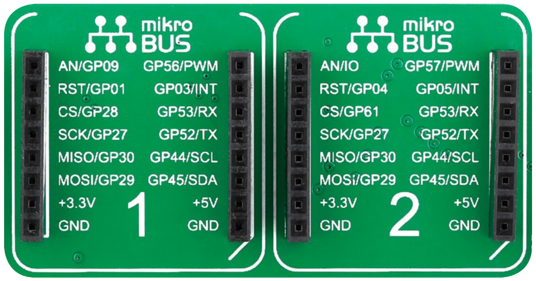
Figure 6-6: SW8.1 through SW8.8 switches are used to enable PORT LEDs

mikroBUS™ sockets

Easier connectivity and simple configuration are imperative in modern electronic devices. Success of the USB standard comes from its 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 the 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 for additional hardware settings. We called this new standard the **mikroBUS™.** EasyFT90x v7 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 **I^{2}C** communications. 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 - I^{2}C Clock line
SDA - I^{2}C Data line
+5V - VCC-5V power line
GND - Reference Ground

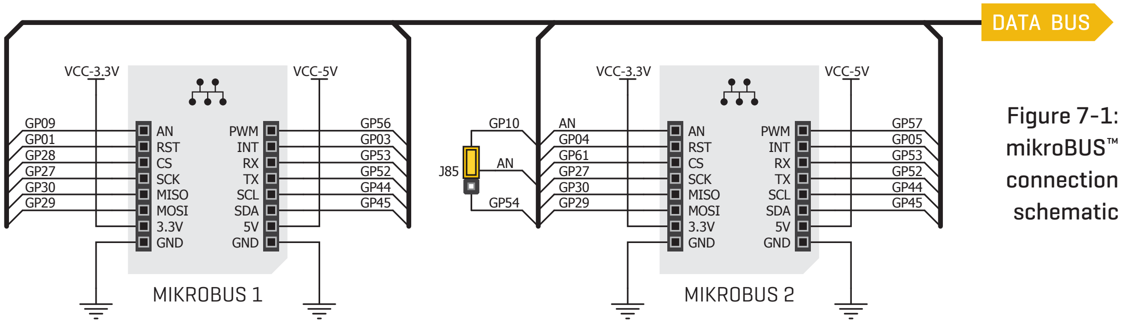
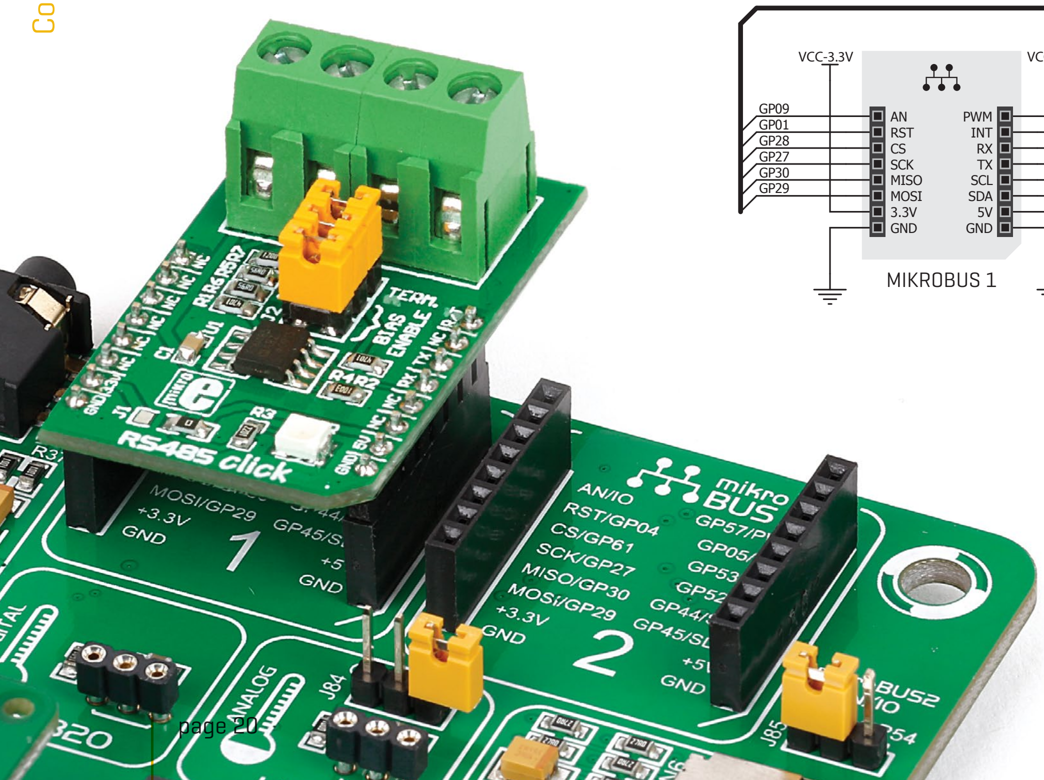


Figure 7-1: mikroBUS™ connection schematic



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:

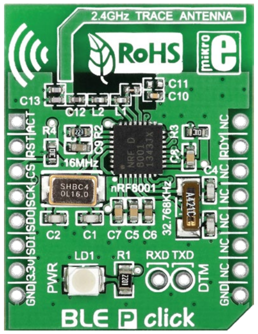
www.mikroe.com/mikrobus

click™ boards are plug-and-play!

For a few years now, we have been expanding our range of click™ boards. Almost **each month several new click™ boards are released, carrying all types of sensors and communication modules.** There are over a 100 click™ boards to choose from. You'll be able to **expand your EasyFT90x v7 board with additional functionality with literally zero hardware configuration**. Just plug and play.

For the complete list of available click™ boards, please visit:

www.mikroe.com/click



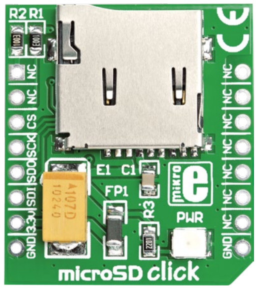
BLE P click™

BlueTooth click™

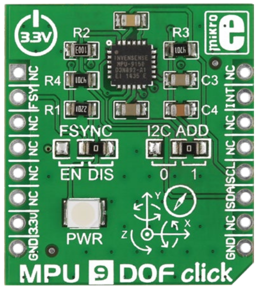
GPS click™

WiFi *PLUS* click™

GSM click™

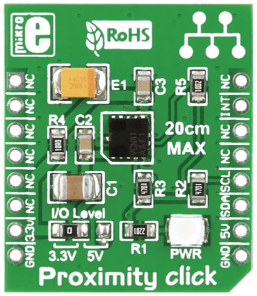


microSD click™

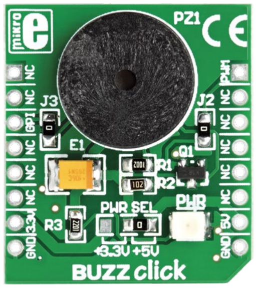


MPU 9DOF click™

nRF C click™

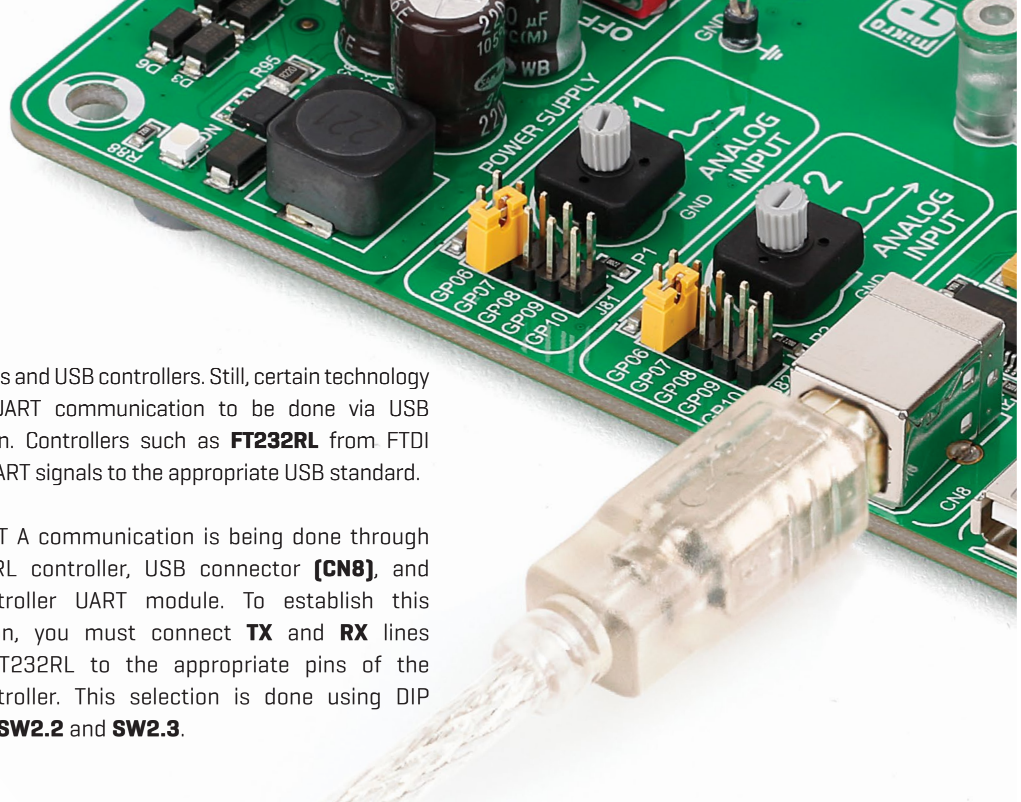


Proximity click™



BUZZ click™

USB-UART



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 that 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 **(CN8)**, 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 **SW2.2** and **SW2.3**.

In order to use the USB-UART module on EasyFT90x v7, you must first install FTDI drivers on your computer. Drivers can be found on link below:

www.ftdichip.com/Drivers/VCP.htm

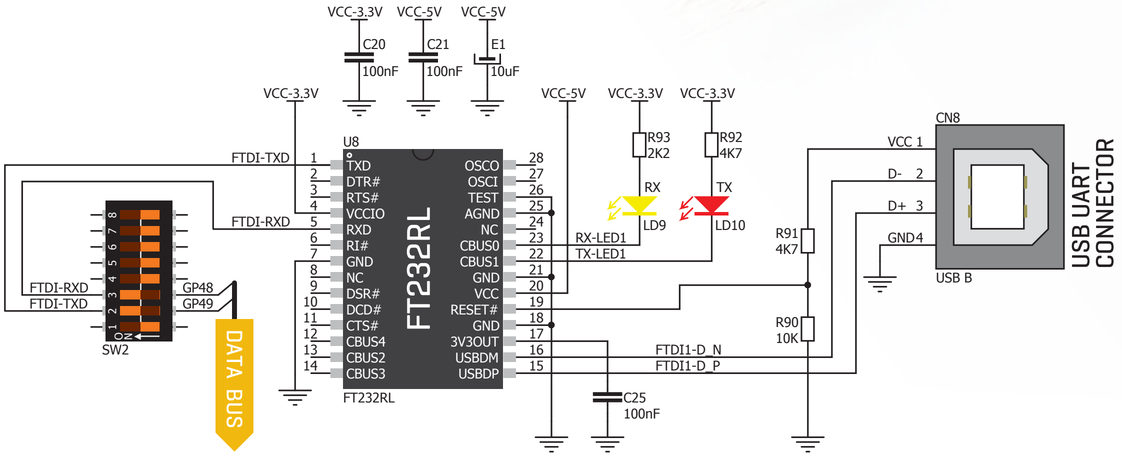
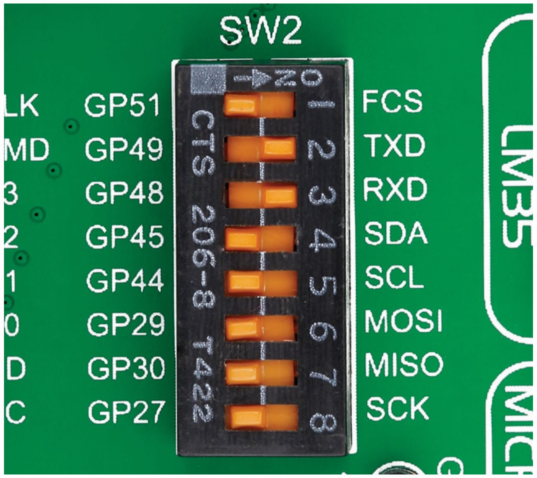
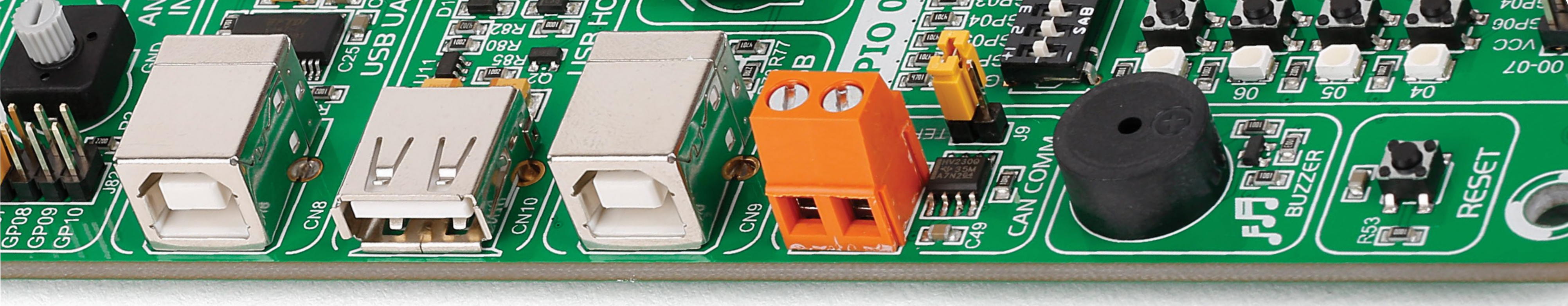


Figure 8-1: USB-UART connection schematic

Enabling USB-UART



In order to enable USB-UART communications, you must push **SW2.2** (GP49) and **SW2.3** (GP48) to **ON** position. This connects the **RX** and **TX** lines to **GP48** and **GP49** microcontroller pins.



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. EasyFT90x v7 contains a USB HOST connector **(CN10)** for a 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 **GP00** pin is established when **SW3.1** is on.

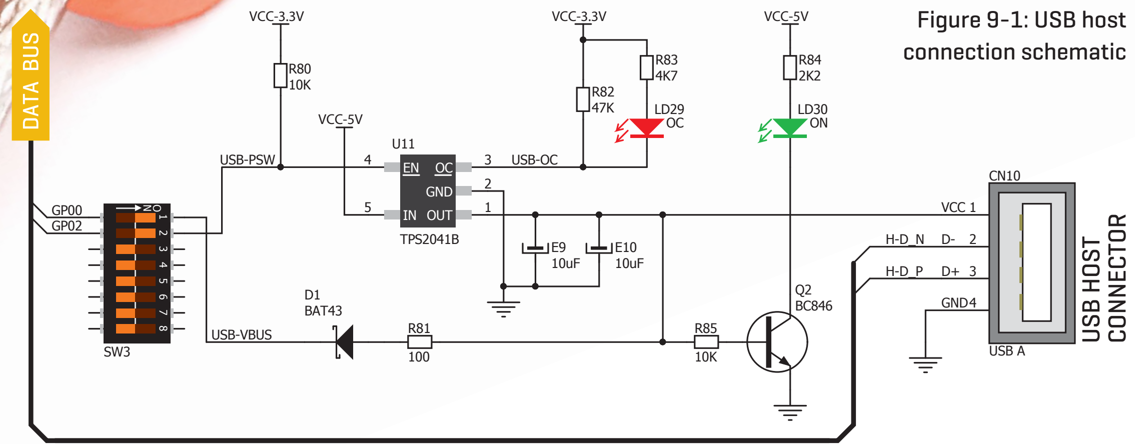


Figure 9-1: USB host connection schematic

Powering USB device

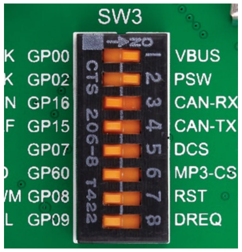
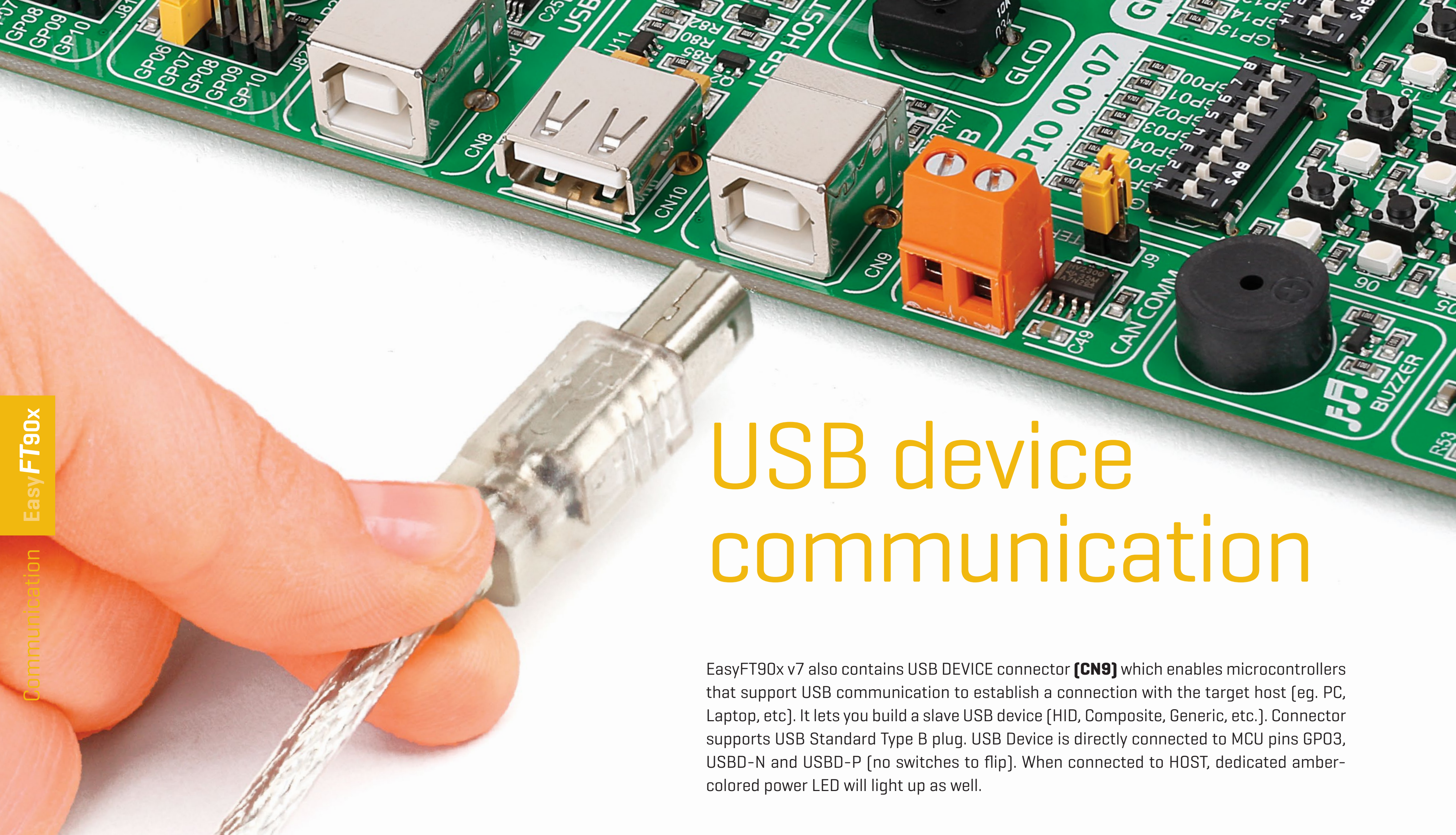


Figure 9-2: Powering USB device through PSW line

You can enable or disable power supply to USB device connected to HOST, through microcontroller **GP02** pin. In order to connect EN TPS2041B IC pin to microcontroller, you must push **SW3.2** to **ON** position.



USB device communication

EasyFT90x v7 also contains USB DEVICE connector **(CN9)** 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. USB Device is directly connected to MCU pins GP03, USBD-N and USBD-P (no switches to flip). When connected to HOST, dedicated amber-colored power LED will light up as well.

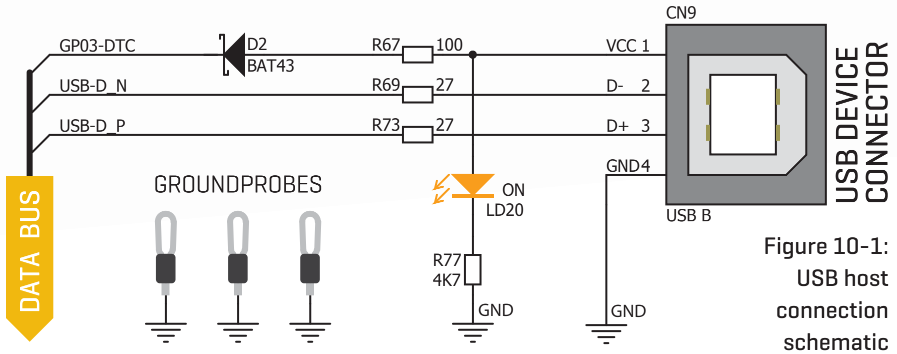


Figure 10-1: USB host connection schematic

Ethernet communication

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. EasyFT90x v7 features a 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.

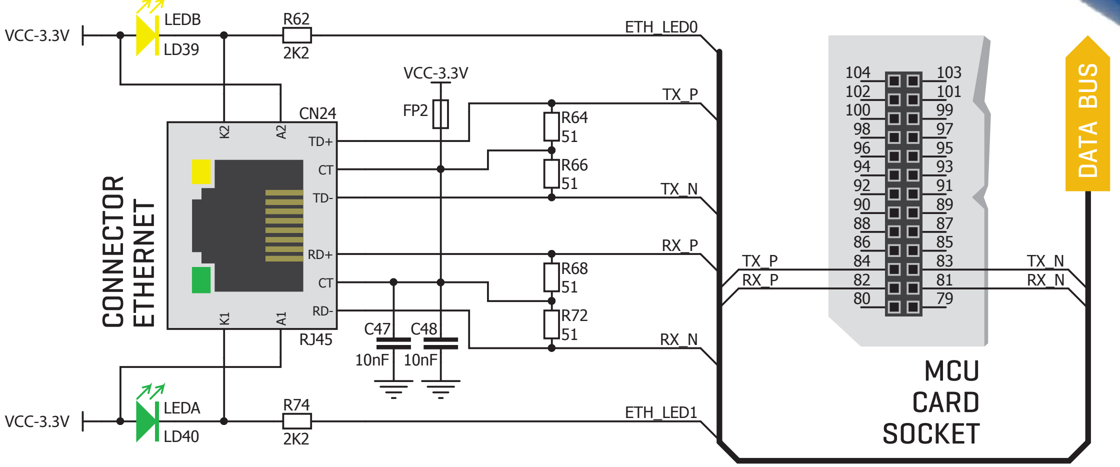
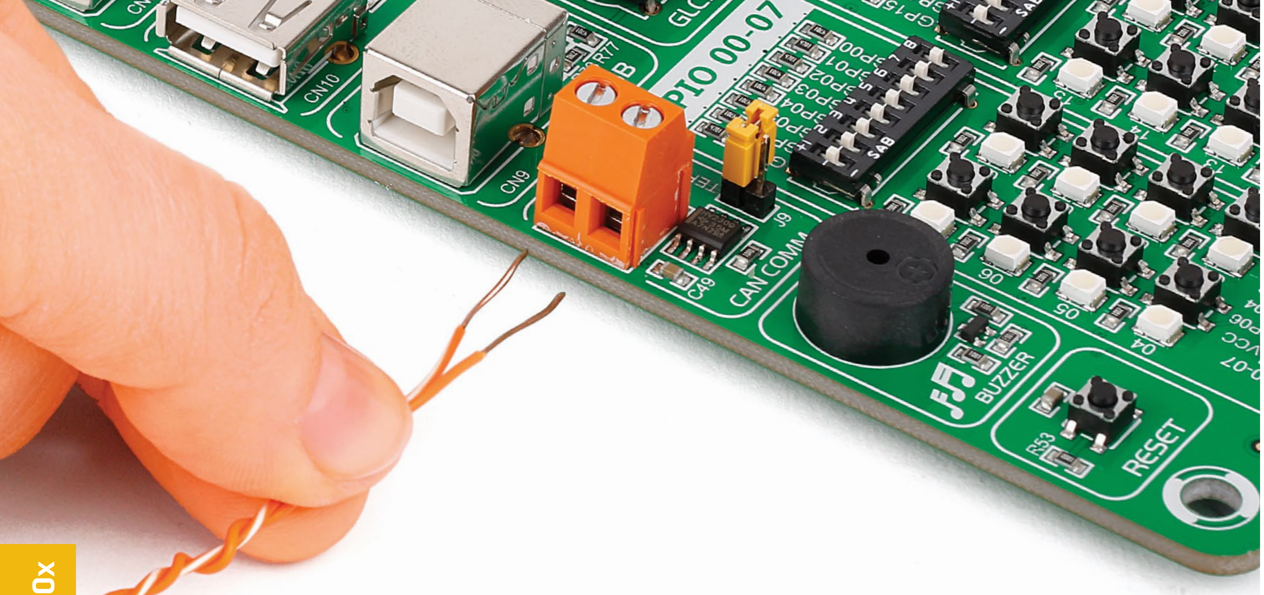


Figure 11-1: Ethernet connection schematic



CAN communication

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. EasyFT90x v7 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

In order to enable CAN communication, you must push **SW3.3** (GP16) and **SW3.4** (GP15) to **ON** position. This connects the **TX** and **RX** lines to appropriate microcontroller pins and its CAN module.

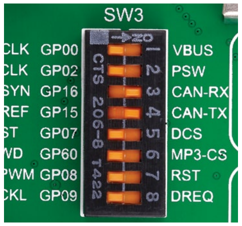


Figure 12-2: enabling CAN communication

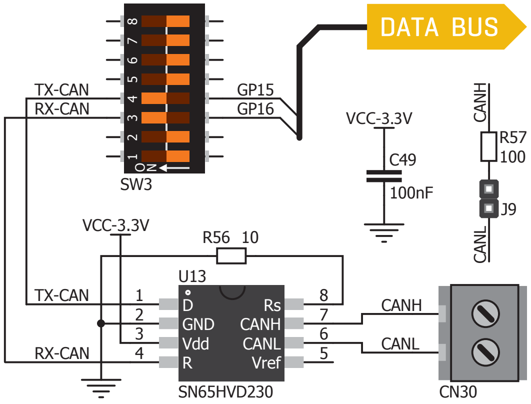
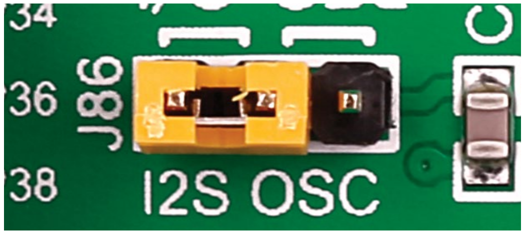


Figure 12-1: CAN connection schematic

I^{2}S

FT90x microcontrollers have an integrated I^{2}S master/slave interface. I^{2}S is a serial bus standard used for carrying audio data between various devices (hence, the protocol is sometimes called Inter-IC Sound). The interface comprises a Bit clock line, a Left/Right clock line (**LRCLK**), a Master clock (**MCLK**) and two data lines (**SDAI**, **SDAO**). The bit clock (**BCLK**) regulates data transmission (one pulse for each bit), while the pulsing of the LR clock specifies which data channel is currently being transferred (by convention, low level is for the left audio channel, high level for the right). The master clock is used by an I^{2}S slave module (if required). The frequencies for these clocks are derived from the external 24.576 MHz crystal oscillator which is tucked under the MCU card on the EasyFT90x board. In case you won't be using I^{2}S, you can free up the extra pin (GP66), by placing Jumper J86 in the GPIO position.



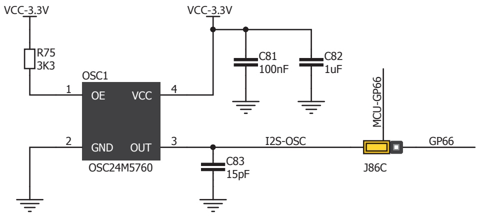
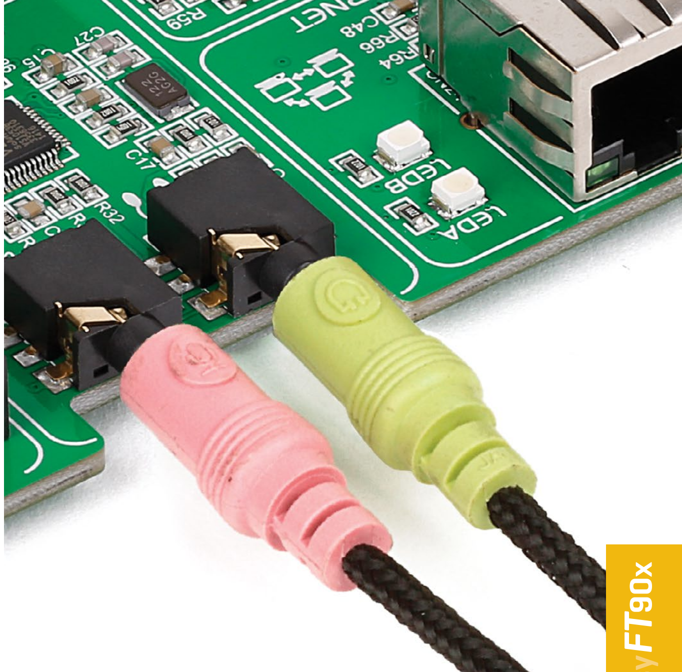


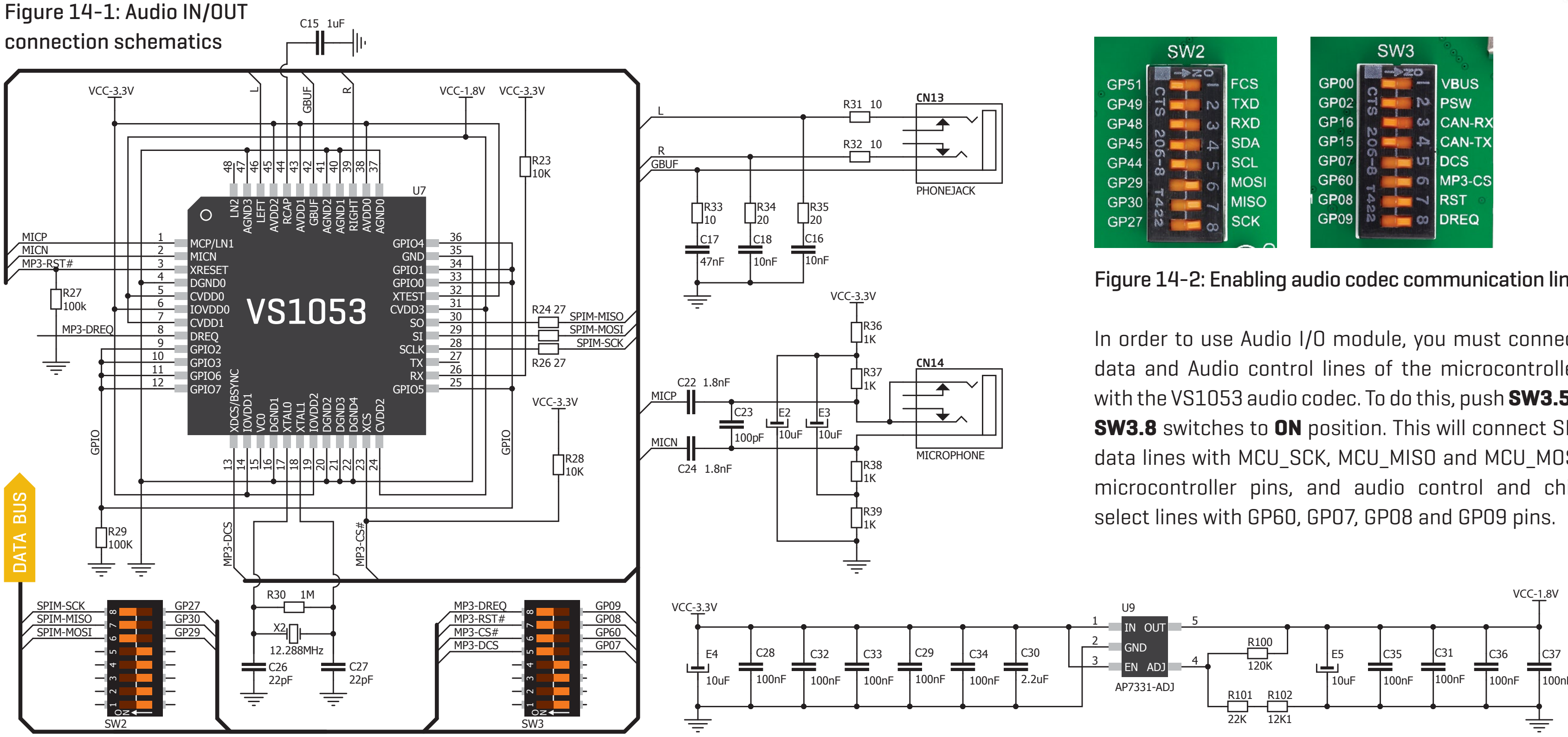
Figure 13-1: Schematic of I^{2}S module

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, EasyFT90x v7 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

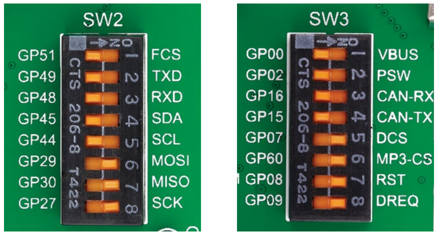
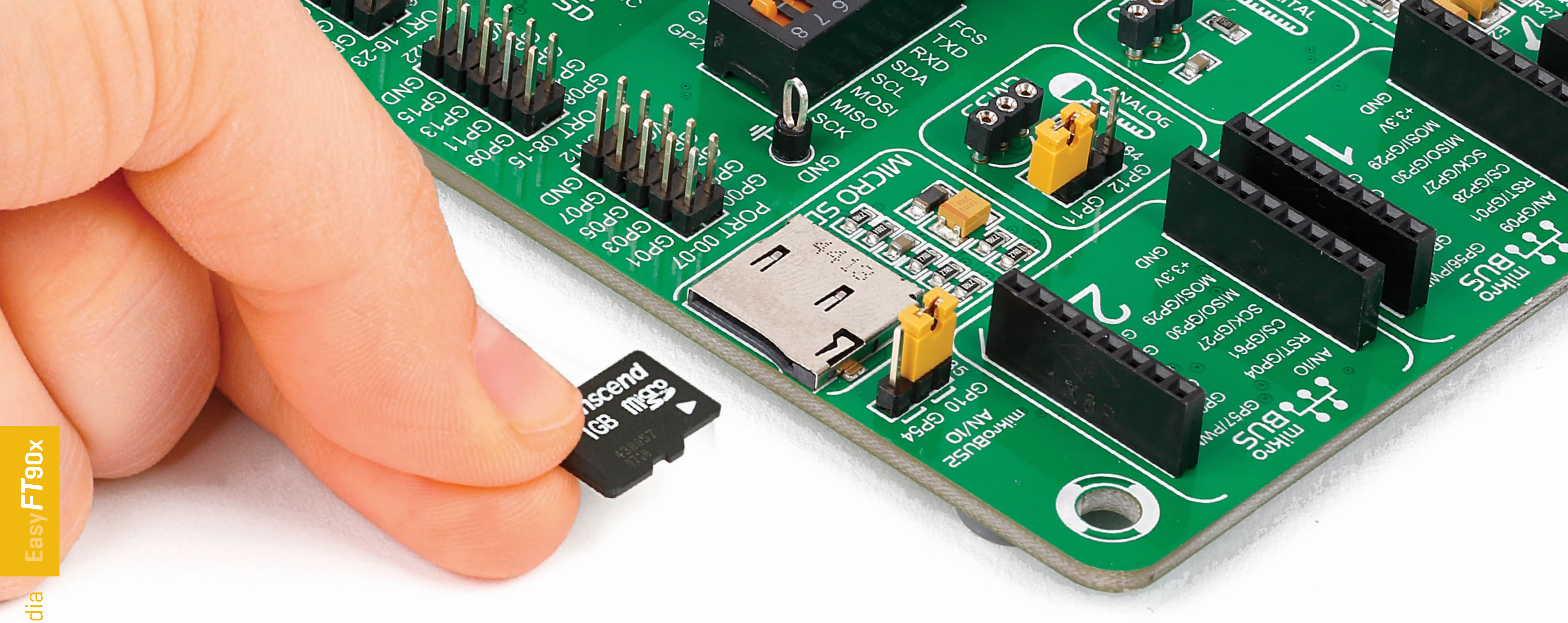


Figure 14-2: Enabling audio codec communication lines

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 **SW3.5–SW3.8** 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 GP60, GP07, GP08 and GP09 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. EasyFT90x v7 features a **microSD card slot**. The microSD form factor is the smallest card format currently available. It uses a 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.

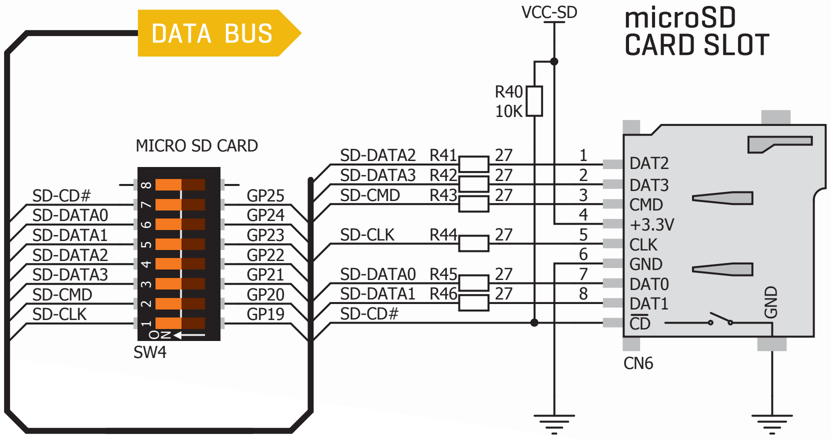
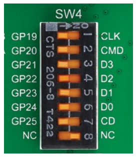


Figure 15-1: microSD card slot connection schematics



Enabling microSD

In order to access the microSD card, you must enable communication lines with switches **SW4.1** through **SW4.8**

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. EasyFT90x v7 features an EasyTFT board carrying a 320x240 pixel 2.83" color TFT display with LED back-light and a **HX8347D** controller.

Each pixel is capable of showing 262.144 different colors. TFT display is connected to microcontroller PORT 32-39 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.

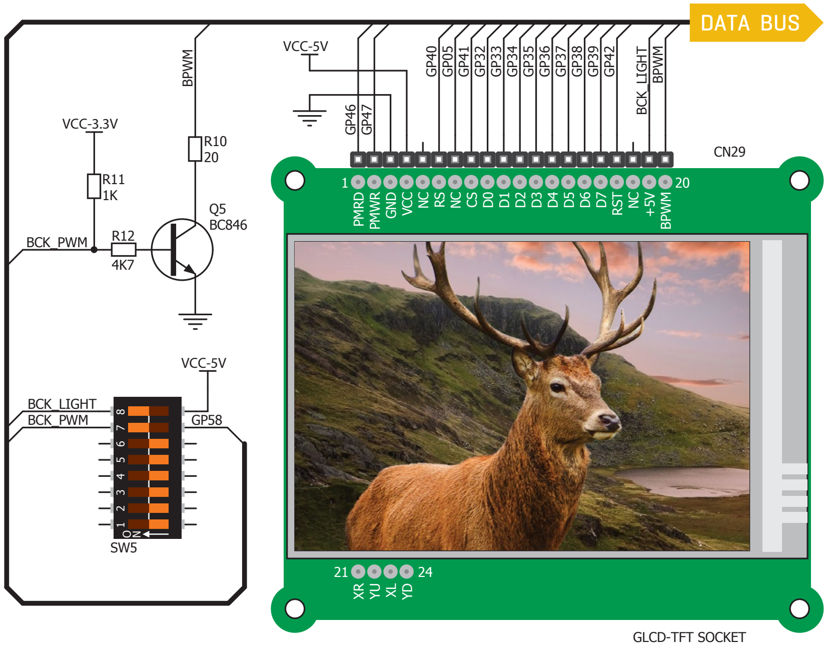
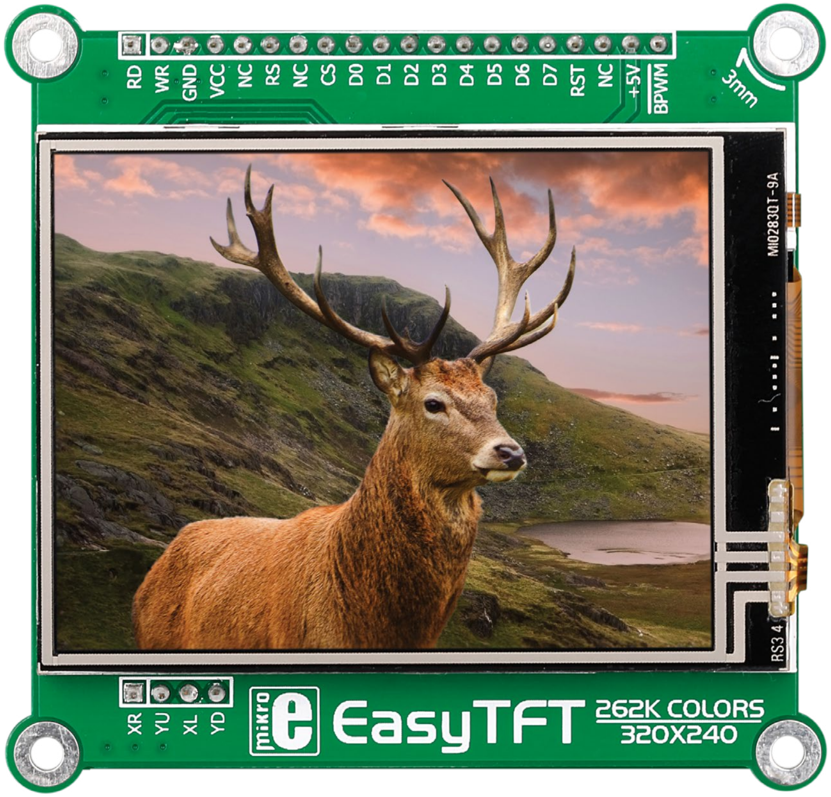


Figure 16-1: TFT display connection schematic



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

Driving Display Back-light

TFT display is enabled using **SW5.7–SW5.8** DIP switches. Back-light can be enabled in two different ways:

1. It can be **turned on with full brightness** using **SW5.8** 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 **SW5.7** and **SW5.8** switches are in ON position.

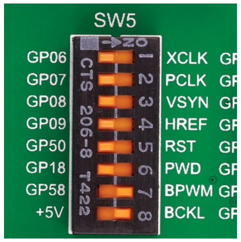


Figure 16-2: Turn on switches **SW5.7** and **SW5.8** to enable back-light

Touch Panel controller

A 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. The pressure at the specific spot is measured by appropriate controllers. This is how touch panels can be used as an input devices. EasyFT90x v7 is equipped with a touch panel controller and a 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

Touch panel is enabled using **SW6.1**, **SW6.2**, **SW6.3** and **SW6.4** switches. They connect **BOTTOM** and **LEFT** lines of the touch panel with **GP10** and **GP11** analog inputs, and **DRIVEA** and **DRIVEB** with **GP31** and **GP43** 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.

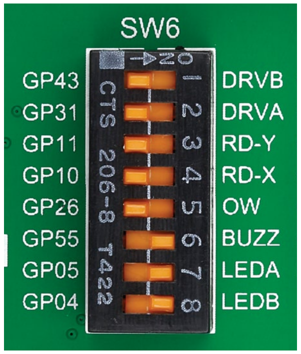
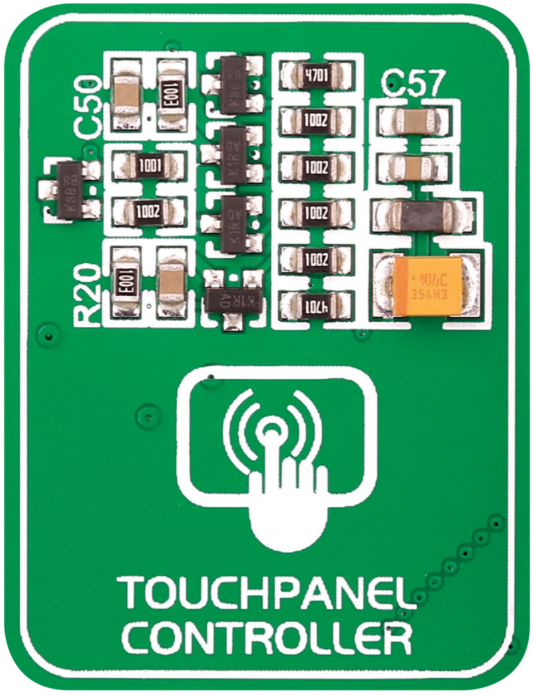


Figure 17-2: Turn on switches **SW6.1, SW6.2, SW6.3** and **SW6.4** to enable Touch panel controller



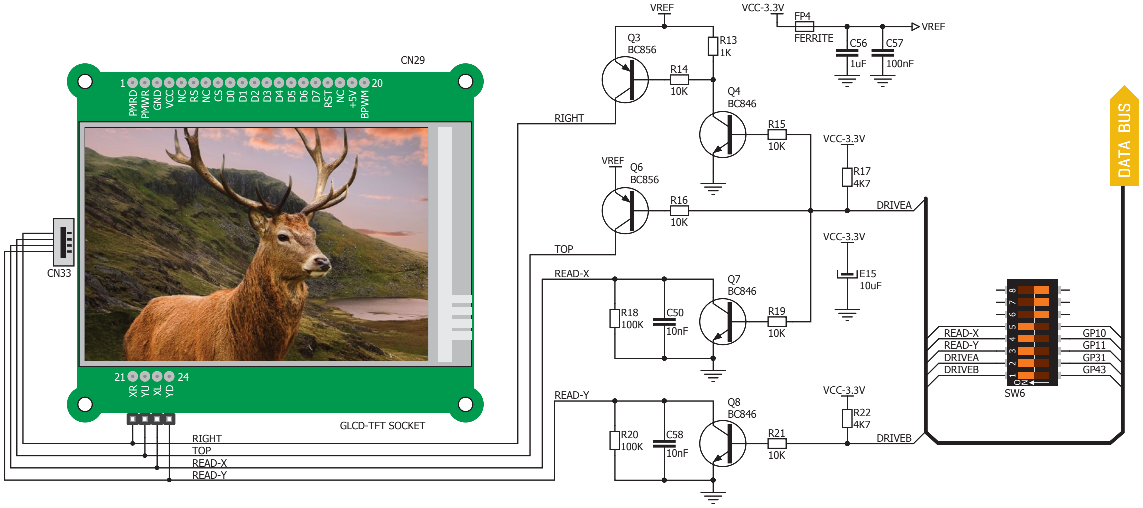
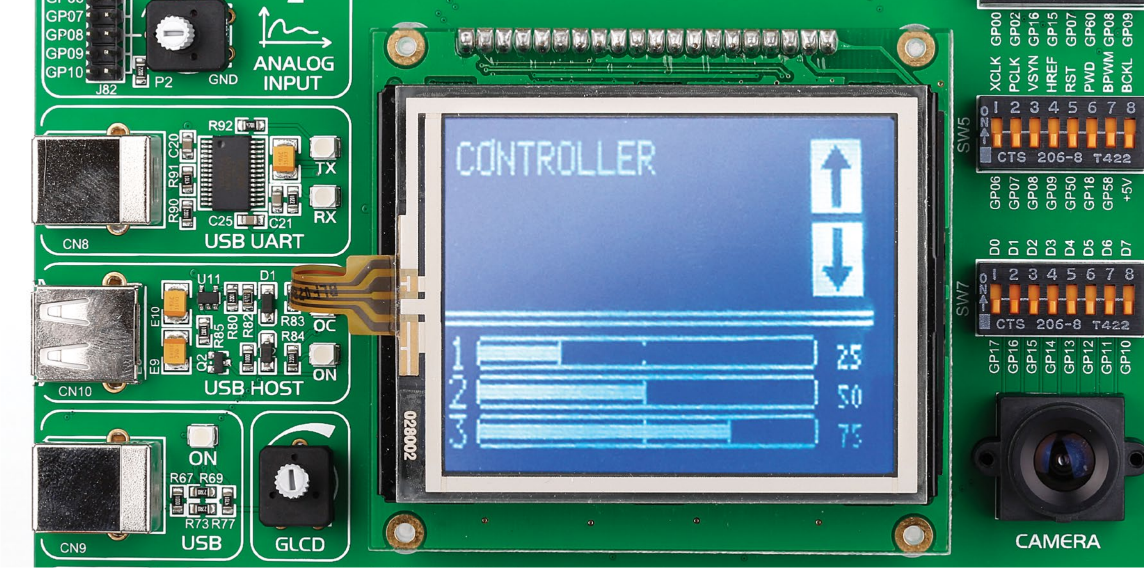


Figure 17-1: Touch Panel controller and connection schematic



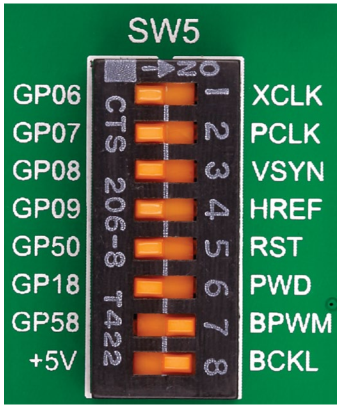


Figure 18-1: To activate full brightness display back-light, flip SW5.8 to ON. For PWM backlight, both SW5.7 and 5.8 should be turned on

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. EasyFT90x v7 provides the connector and the necessary interface for supporting GLCD with a resolution of 128x64 pixels, driven by the KS108 or compatible display controller. Communication with the display module is done through **CN33** display connector. Board is fitted with a plastic display distancer, which allows the GLCD module to perfectly and firmly fit into place.

Display connector is routed to PORT 32-39 (control and data lines) of the microcontroller sockets. The same port is also used by the TFT display. You can control the display contrast using the dedicated potentiometer **P3**. Full brightness display back-light can be enabled with **SW5.8** switch and PWM-driven back-light with **SW5.7** switch.

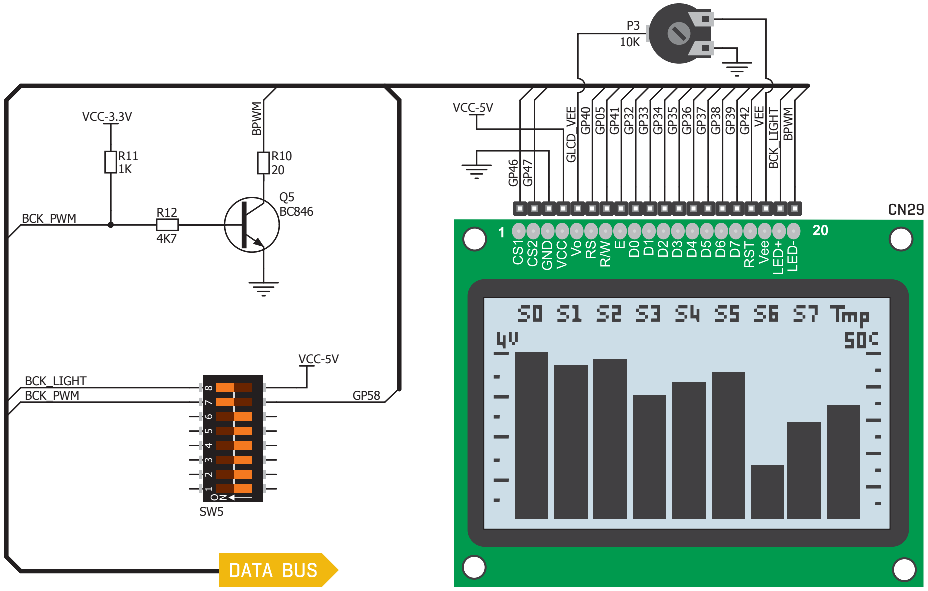


Figure 18-2: GLCD 128x64 connection schematic

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

Onboard camera

One of the most distinguishing features of FT90x microcontrollers is the integrated Camera Parallel interface (CPI). To make full use of it, and to allow you to develop multimedia applications that combine the camera and TFT, EasyFT90x incorporates a OV07670-VL2A CMOS image sensor. The sensor has a resolution of up to 640x480px (but keep in mind that the EasyTFT screen has 320x240px). CPI is an 8-bit interface. Camera control signals are VSYNC, HREF, and PCLK. Therefore, to enable the camera, activate DIP switches from 7.1 through 7.8, as well as SW5.1 through 5.6. SW2.4 and SW2.5 should also be on.

Enabling the camera

To use the onboard camera, flip the following switches to the ON position: SW7.1, SW7.2, SW7.3, SW7.4, SW7.5, SW7.6, SW7.7, SW7.8, SW5.1, SW 5.2, SW5.3, SW5.4, SW5.5, SW5.6, SW2.4 and SW2.5

A jumper to help you stream video over WiFi

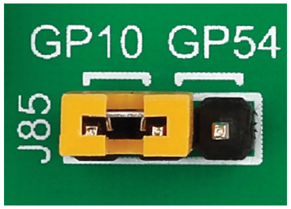
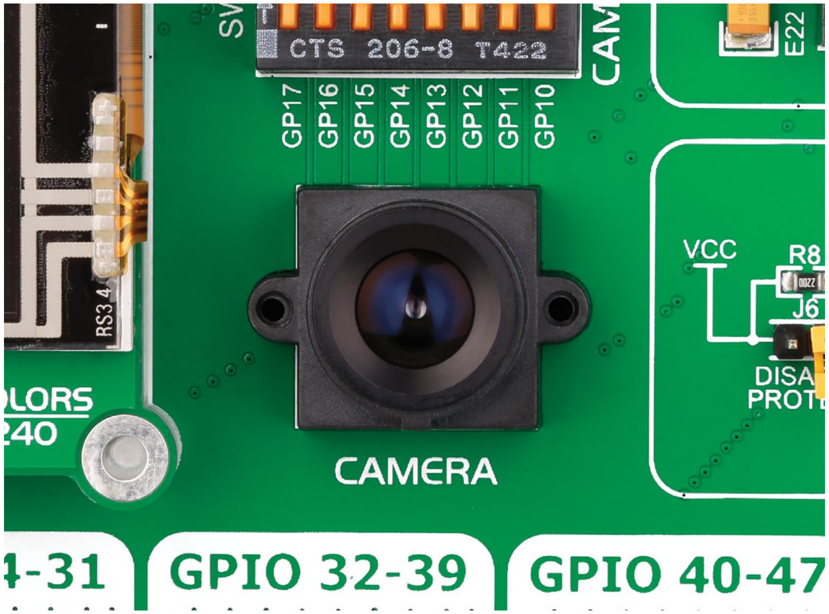


Figure 19-1: Jumper J85 reroutes the AN pin on the second mikroBUS™ socket from GP10 to GP54

The first upper left pin of the mikroBUS™ socket is usually reserved for an analog line, but in some click™ boards that pin has a different function. For example, WiFi3 click uses that pin for a PWD line. Since the camera on EasyFT90x takes up all the the analog lines, we added a jumper (J85) that enables you to reroute the AN pin on the second mikroBUS™ socket (GP10) to a General Purpose IO (GP54). This allows you to use some of those click boards like WiFi3 click together with the camera, to stream video over WiFi, for example.



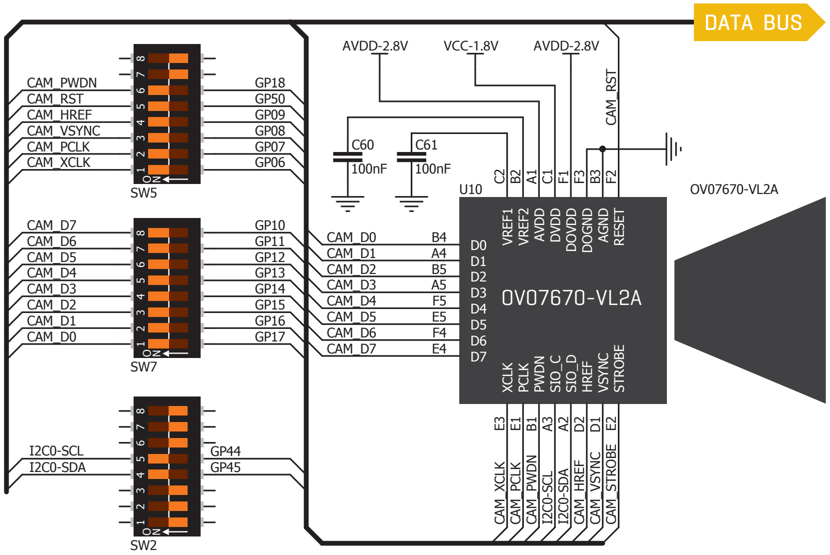


Figure 19-2: Onboard camera connection schematic.

Navigation switch

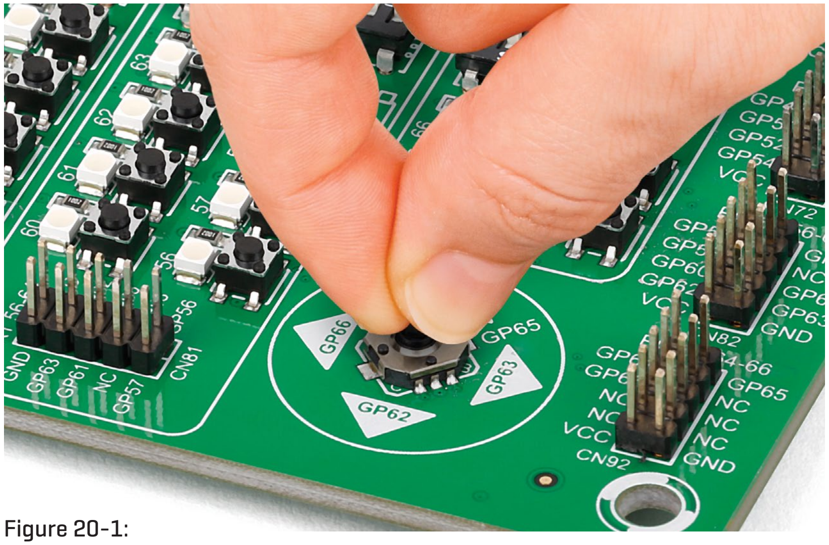
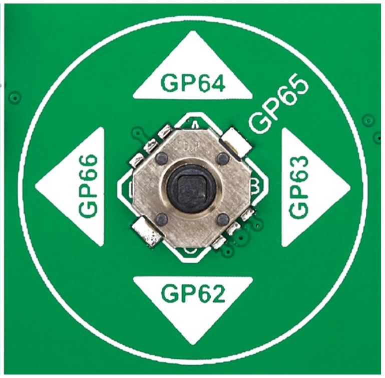


Figure 20-1:
Navigation switch is an intuitive solution for browsing through on-screen menus.



When working with multimedia applications it is far more intuitive to use a single joystick than several different push buttons that are far apart. This is more natural for users as they can browse through on-screen menus, or even play games much easier. EasyFT90x v7 features a 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: **GP64**, **GP62**, **GP66**, **GP63**, **GP65** (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.

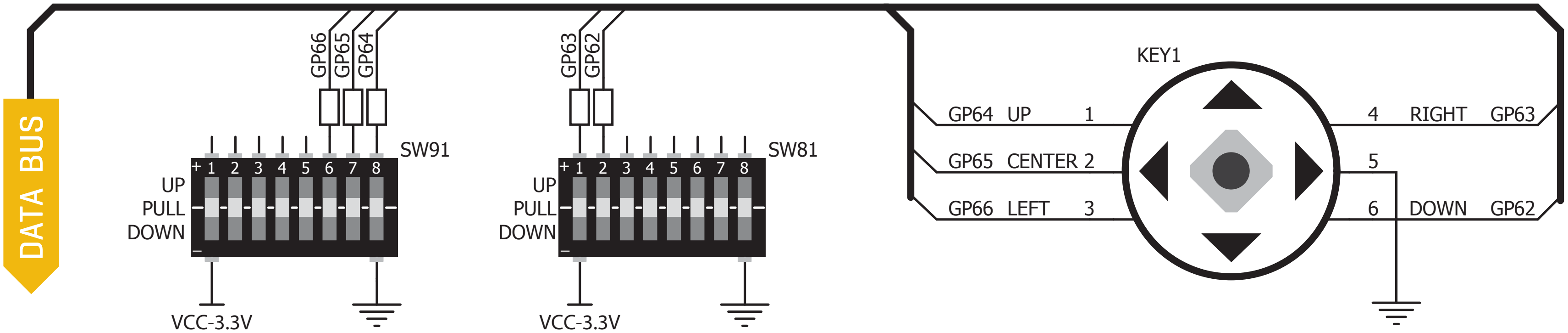


Figure 20-2: Navigation switch connection schematic. Pull-up resistors should be enabled during operation

Piezo Buzzer

Piezoelectricity is the charge that accumulates in certain solid materials in response to mechanical pressure. It works in reverse too: providing a charge to piezoelectric materials causes them to phisically deform. One of the most widely used applications of piezoelectricity is the production of sound generators, called piezo buzzers. EasyFT90x v7 for comes with a piezo buzzer connected to **GP55** microcontroller pin. Connection is established using **SW6.6** 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. You can generate a PWM signal using a 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 frequenciesof 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.

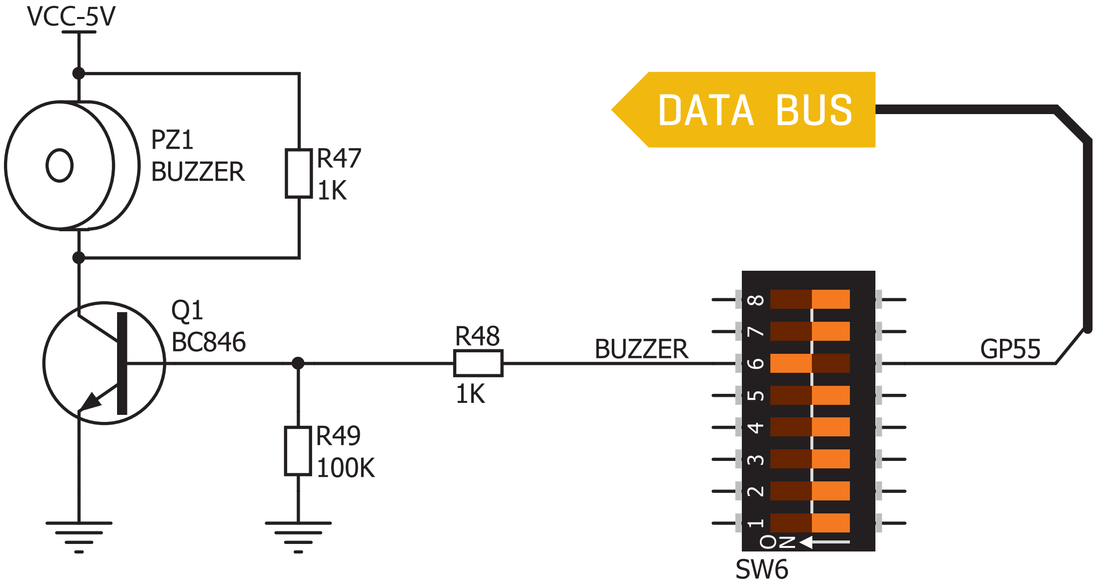


Figure 21-1: Piezo buzzer connected to GP55 microcontroller pin

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 the **SW6.6** DIP switch which connects it to **GP55** pin.

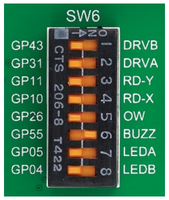
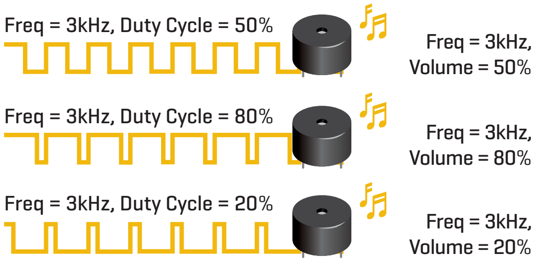


Figure 21-2:
push SW6.6 to ON position to
connect Piezo buzzer to GP55

Supported sound frequencies

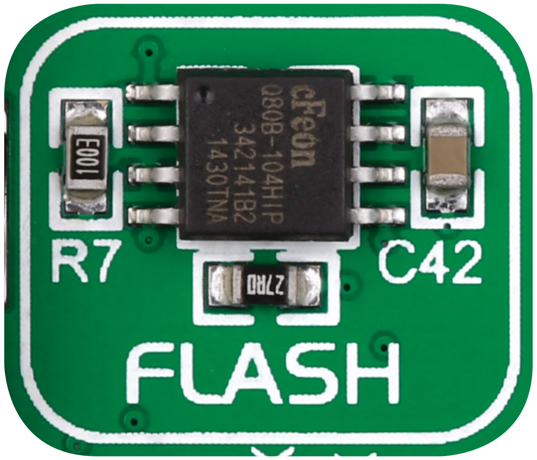
Piezo buzzer's resonant frequency (where you can expect it's best performance) is **3.8kHz**, but you can also use it to create sound in the range between **2kHz** and **4kHz**.

How to make it sing?


Freq = 3kHz, Duty Cycle = 50%
Freq = 3kHz,
Volume = 50%
Freq = 3kHz, Duty Cycle = 80%
Freq = 3kHz,
Volume = 80%
Freq = 3kHz, Duty Cycle = 20%
Freq = 3kHz,
Volume = 20%

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.

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.

EasyFT90x 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.

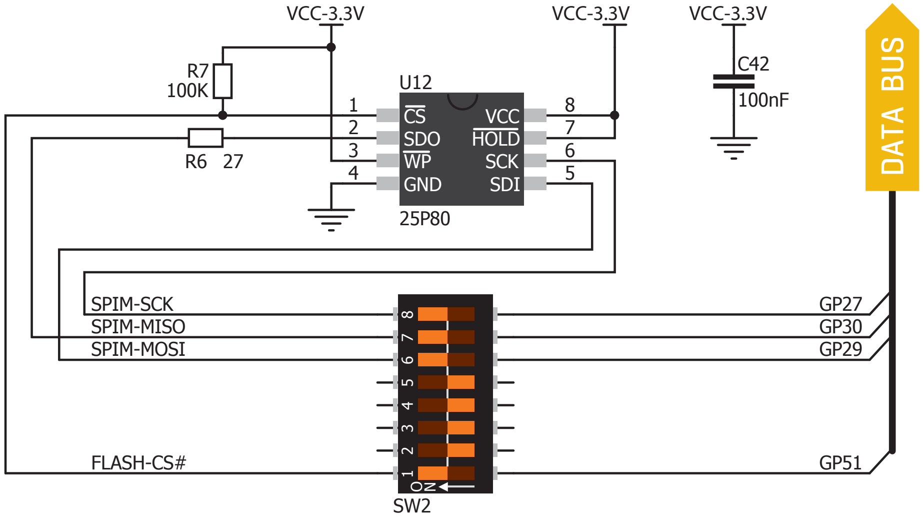
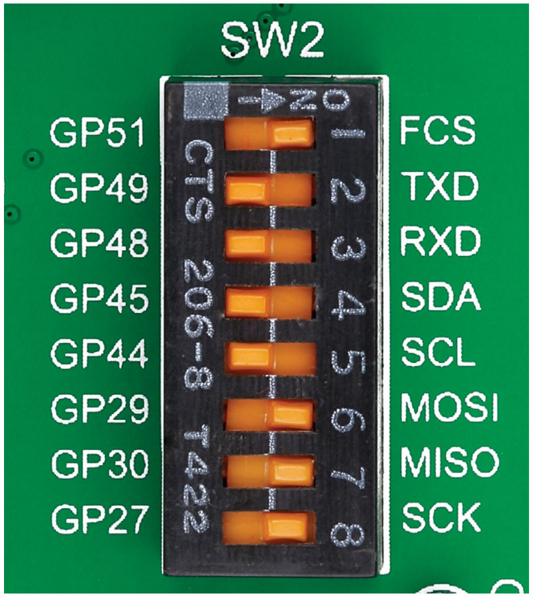


Figure 22-1: Schematic of Serial Flash Memory module

Enabling Serial Flash



In order to connect Serial Flash Memory to the microcontroller you must enable **SW2.1, SW2.6, SW2.7** and **SW2.8** switches. This connects SPI lines to **GP51 (CS), MOSI, MISO AND SCK** microcontroller pins.

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.

Digital Temperature Sensor DS1820

DS1820 is a digital temperature sensor that uses **1-wire® interface** for its 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.

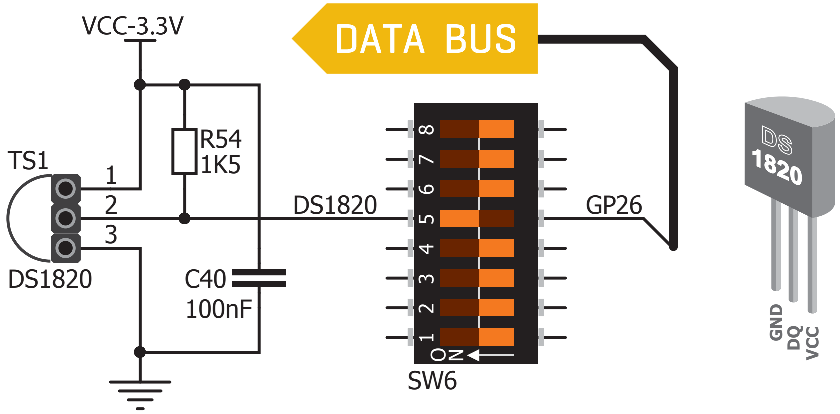
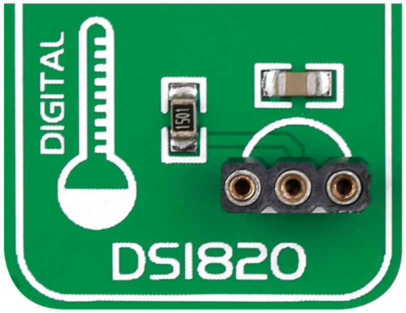


Figure 23-1: DS1820 connected to PB10 pin



Board provides a separate socket (**TS1**) for the DS1820. Communication line with the microcontroller is established through the **SW6.5** DIP switch (ON position).

Enabling DS1820 Sensor

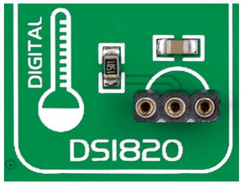


Figure 23-2: DS1820 socket

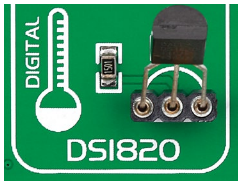


Figure 23-3: DS1820 correctly placed in socket

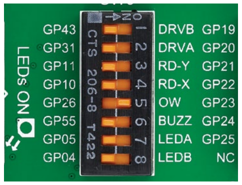

GP43 GP31 GP11 GP10 GP26 GP55 GP05 GP04
DRVB DRVA RD-Y RD-X OW BUZZ LEDA LEDB
GP19 GP20 GP21 GP22 GP23 GP24 GP25 NC
LEDs ON
1 2 3 4 5 6 7 8

Figure 23-4: Enabled SW6.5 DIP switch

EasyFT90x v7 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 **SW6.5** 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.

Analog Temperature Sensor LM35

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 degrees 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. EasyFT90x v7 enables you to get analog readings from the LM35 sensor in restricted temperature range from +2°C to +150°C. The board has a socket (**TS2**) for the LM35 sensor in TO-92 plastic packaging. The microcontroller reads of the temperature data of a single analog input line, which is selected with jumper J84.

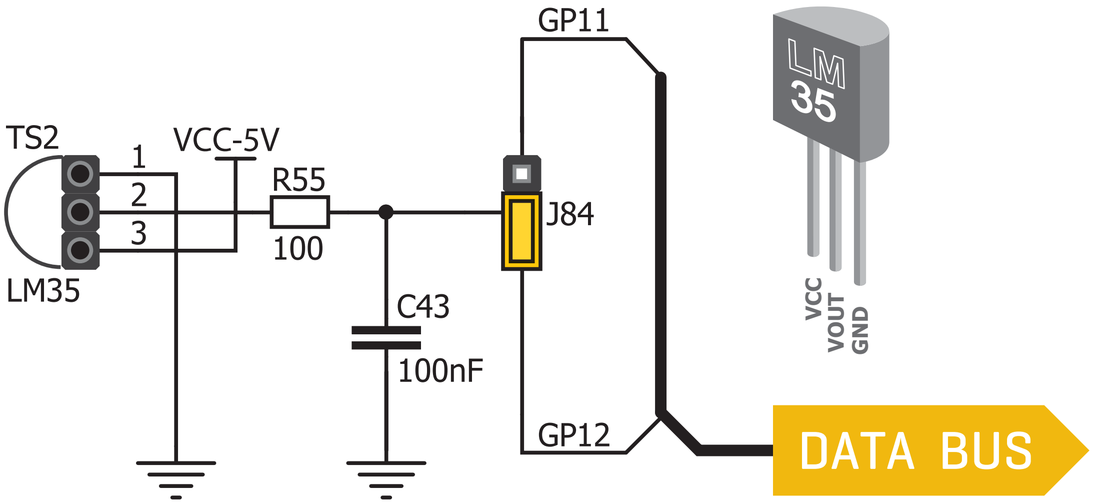


Figure 24-1: LM35 connected to PC0 pin

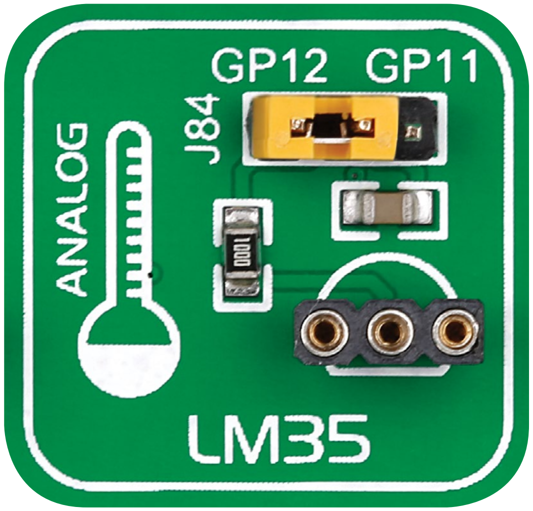


Figure 24-2: Choose between two analog input lines to use with LM35, with jumper J84

Enabling LM35 Sensor

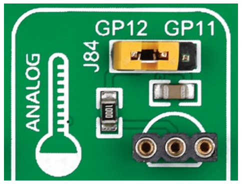


Figure 24-3: LM35 socket

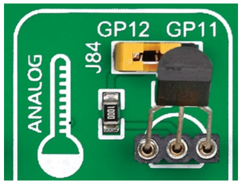


Figure 24-4: LM35 correctly placed in socket

Use the jumper J84, right next to the temperature sensor, to connect it to either GP11 or GP12. Having two pins to choose from gives you more flexibility to use the temperature sensor with other board modules. 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.

I²C EEPROM

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 its 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.

EasyFT90x v7 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 depend on the power supply voltage, and go up to **400 kHz** for a 3.3V power supply.

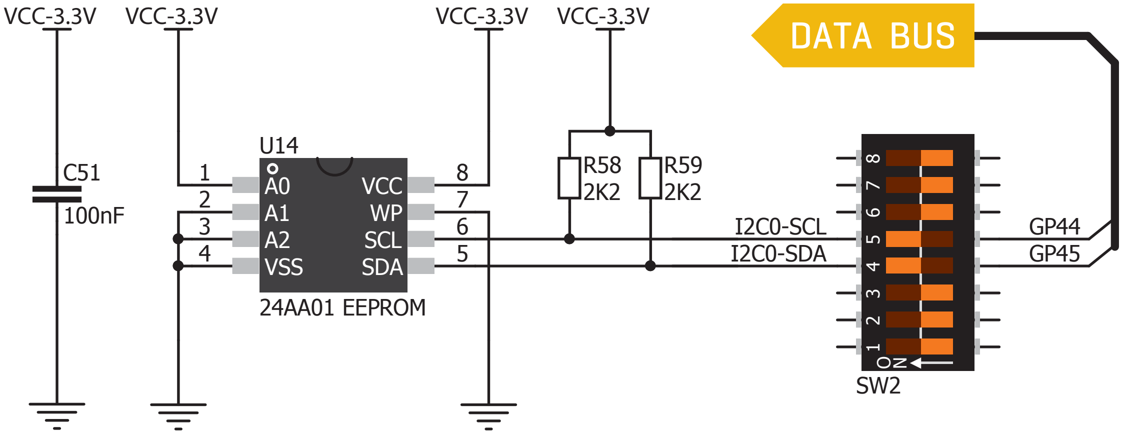


Figure 25-1: Schematic of I²C EEPROM module

Enabling I²C EEPROM

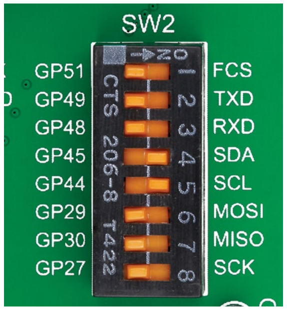


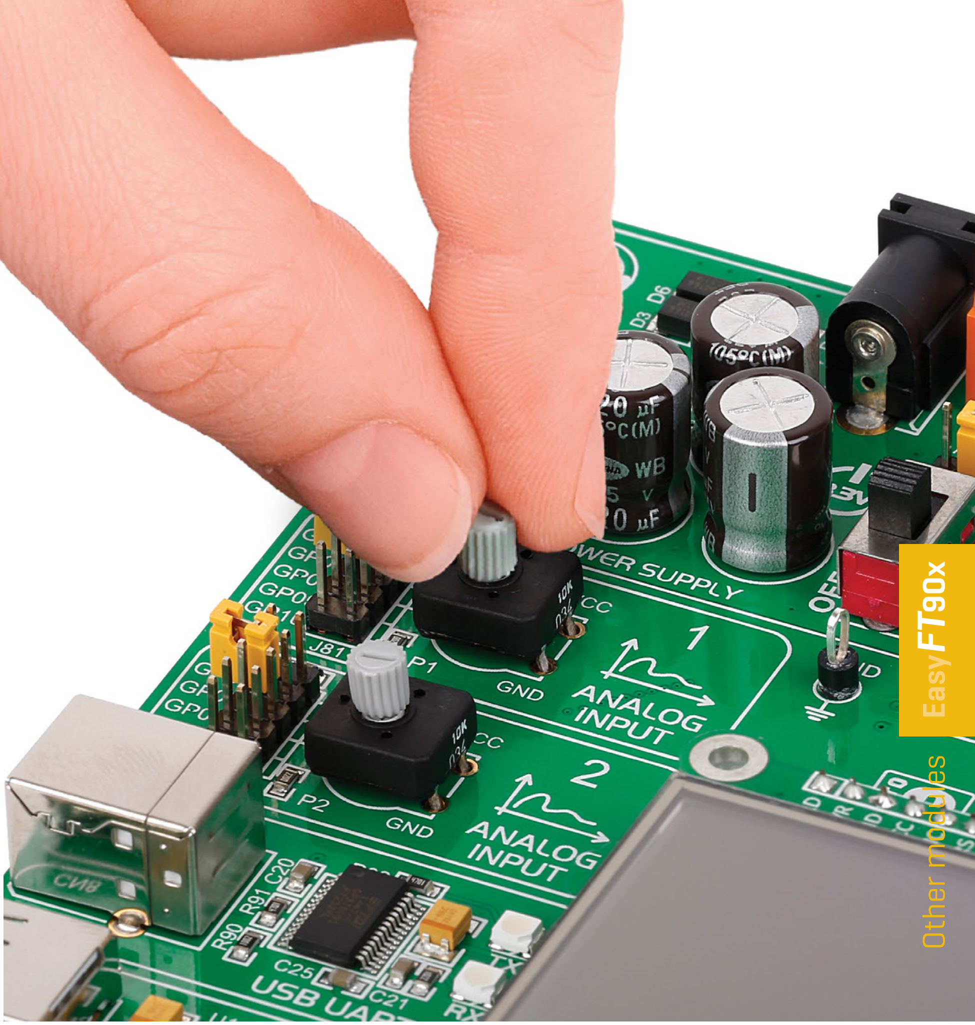
Figure 25-2: Turn on switches **SW2.4** and **SW2.5** to connect EEPROM lines to MCU

In order to connect I²C EEPROM to the microcontroller you must enable **SW2.4** and **SW2.5** 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 GP45 and **GP44** communication lines that could interfere with the data signals and cause data corruption.

What is I²C?

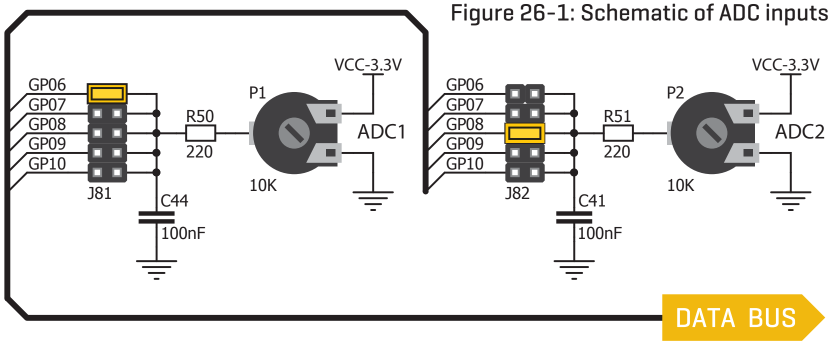
I²C is a multi-master serial single-ended bus that is used to attach low-speed peripherals to computer or embedded systems. I²C uses only two open-drain lines, **Serial Data Line (SDA)** and **Serial Clock (SCL)**, pulled up with resistors. **SCL** line is driven by a master, while **SDA** is used as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

ADC inputs



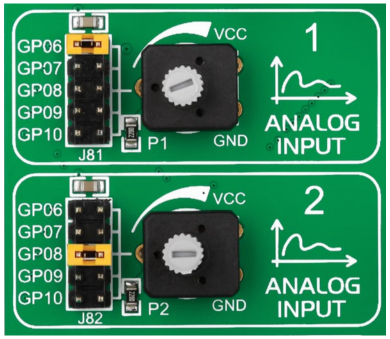
Digital signals have two discrete states, which are decoded as high and low, and interpreted as logic 1 and logic 0. Analog signals, on the other hand, are continuous, and can have any value within defined range. A/D converters are specialized circuits which can convert analog signals (voltages) into 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 the 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 210=1024. This means that the supported voltage range, for example from 0-1.8V, can be divided into 1024 discrete steps of about 1.758mV. EasyFT90x provides two interfaces in the form of potentiometers for simulating analog input voltages that can be routed to any of the 5 supported analog input pins.

Figure 26-1: Schematic of ADC inputs



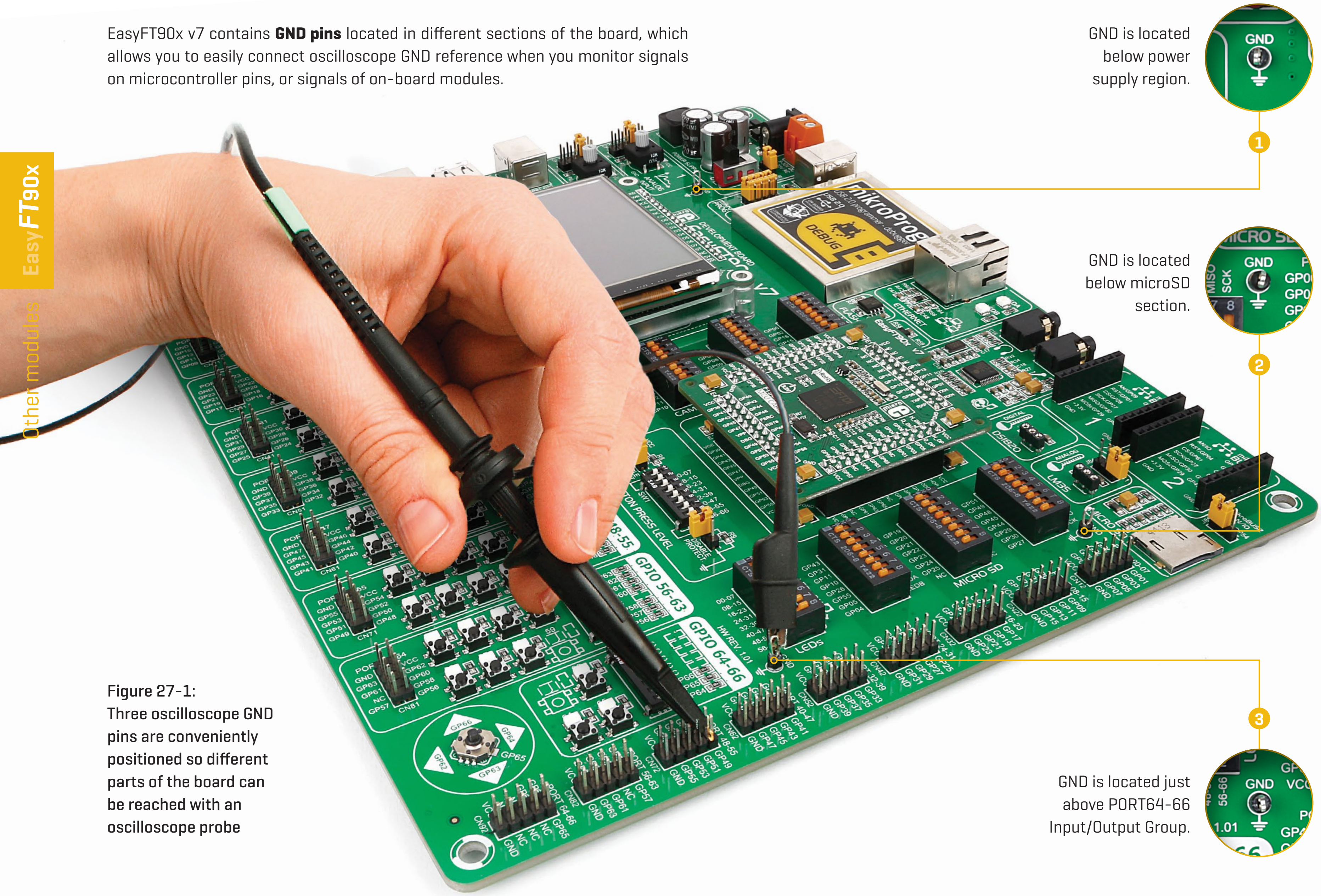
Enabling ADC inputs

In order to connect the output of the potentiometer P1 to GP06, GP07, GP08, GP09 or GP10 analog microcontroller input, you have to place the jumper J81 in the desired position. By moving the potentiometer knob, you can create voltages in range from GND to VCC. The same applies for potentiometer P2 and its corresponding jumper J82 (just make sure not to put both potentiometers to the same input)



Additional GNDs

EasyFT90x v7 contains **GND pins** located in different sections of the board, which allows you to easily connect oscilloscope GND reference when you monitor signals on microcontroller pins, or signals of on-board modules.



GND is located below power supply region.

GND is located below microSD section.

GND is located just above PORT64-66 Input/Output Group.

Figure 27-1: 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 **EasyFT90x v7**. You got to know its 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 the EasyFT90x 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 **the FT90x compiler** that suits you best on our website:

www.mikroe.com/ft90x/compilers

Choose between **mikroC™**, **mikroBasic™** and **mikroPascal™** and download fully functional demo version, so you can begin building your FT90x 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 EasyFT90x v7 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:

www.mikroe.com/easyft90x

3 Community

If you want to find answers to your questions on many interesting topics we invite you to visit our forum at **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.

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!

www.mikroe.com/support

www.mikroe.com

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, may 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, mikroC™, mikroBasic™, mikroPascal™, Visual TFT™, Visual GLCD™, mikroProg™, Ready™, MINI™, mikroBUS™, EasyPIC™, EasyAVR™, Easy8051™, click™ boards and mikromedia™ 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**

EasyFT90x v7 manual
ver 1.01a

Designed by
MikroElektronika Ltd.
www.mikroe.com