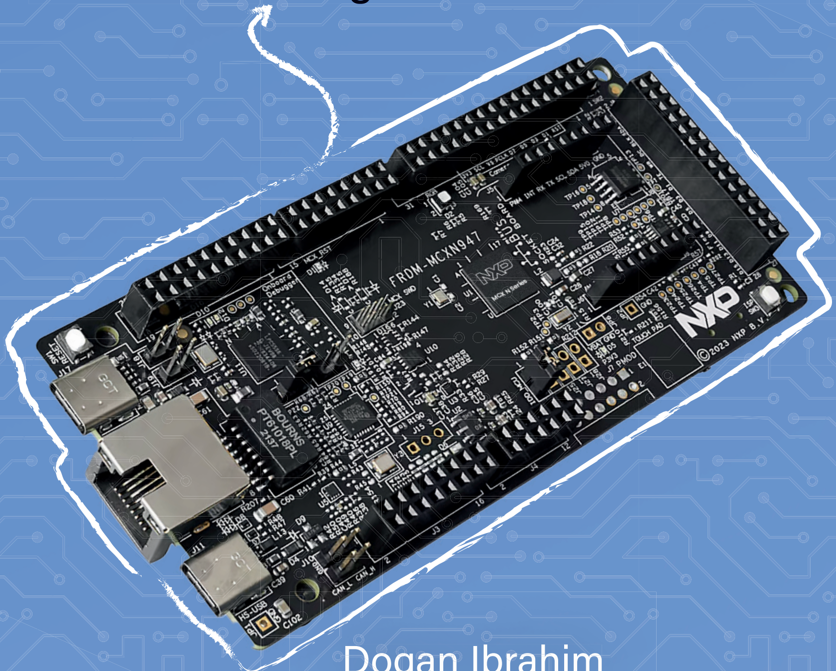
H0W2 Volume 4

Get Started with the NXP FRDM-MCXN947 Development Board

Develop projects on connectivity, graphics, machine learning, motor control, and sensors



Dogan Ibrahim

NXP

elektor knows how

Get Started with the

FRDM-MCXN947

Development Board

Dogan Ibrahim

elektor
design > share > earn

This is an Elektor Publication. Elektor is the media brand of
Elektor International Media B.V.
PO Box 11, NL-6114-ZG Susteren, The Netherlands
Phone: +31 46 4389444

Declaration

The authors and publisher have used their best efforts in ensuring the correctness of the information contained in this book. They do not assume, and hereby disclaim, any liability to any party for any loss or damage caused by errors or omissions in this book, whether such errors or omissions result from negligence, accident or any other cause.

ISBN 978-3-89576-634-3 Print
ISBN 978-3-89576-635-0 eBook

www.elektor.com
Prepress Production: D-Vision, Julian van den Berg
Printers: Ipskamp, Enschede, The Netherlands

Contents

Preface

It is becoming important for microcontroller users to adapt to new technologies quickly and learn the architecture and use of high performance 32-bit microcontrollers. Several manufacturers offer 32-bit microcontrollers as general-purpose processors in embedded applications. For example, Microchip Inc offers the 32-bit PIC family of microcontrollers and development tools in addition to their highly popular 8-bit and 24-bit family. Companies like NXP Semiconductors, STMicroelectronics and several others offer ARM based processors for high-speed professional applications.

ARM offers 32-bit and 64-bit processors for the embedded applications. Nowadays, majority of mobile devices such as mobile phones, tablets, and GPS receivers are based on the ARM processors. The low cost, low power consumption, and high performance of the ARM processors make them ideal candidates to be used in complex communication and mixed signal applications.

This book is about the use of the FRDM-MCXN947 Development Board, developed and manufactured by NXP Semiconductors. This is a complex low-cost through-hole USB-powered PCB. At its heart lies the MCXNx4x family MCU, featuring NXP's advanced implementation of the Arm Dual Cortex-M33 TrustZone microcontroller. This microcontroller operates at speeds of up to 150 MHz to provide high CPU performance and best real-time response. It is supported by Zephyr OS for developing Internet of Things with a free, open-source embedded operating system. Popular development IDE tools such as the MCUXpresso IDE, MCUXpresso for VS Code, IAR Embedded Workbench, or the Keil MDK can be used for program development. Additionally, a powerful SDK is provided which simplifies program development greatly. The board is shipped with 2 MB dual-bank on-chip flash, 512 KB RAM, 10 x LP Flexcomms each supporting SPI, I2C, and UART, 2 x FlexCAN, Ethernet, on-board MCU-Link debugger with CMSIS-DAP, ADC, DAC, RTC, digital MEMS microphone interface, LCD interface, capacitive touch sensor interface, OpAmp, analog comparators, many timers, etc.

One of the important features of the development board is that it supports N1-16 Neural Processing Unit (NPU), thus enabling users to develop AI based projects. The development board also supports Arduino UNO form factor header pins, making it compatible with many Arduino shields, mikroBUS connector for mikroElektronica Click Boards, and Pmod connector.

One of the nice things of the FRDM-MCXN947 development board is that several on-board debug probes are provided with so that programmers can debug their programs by communicating directly with the MCU. With the help of the debugger, programmers can single step through a program, insert breakpoints, view and modify variables and so on.

Many working and tested projects have been developed in the book using the popular MCUXpresso IDE and the SDK with various sensors and actuators. The project descriptions, block diagrams, circuit diagrams, complete program listings, and detailed descriptions of all the developed programs are given in the book for all the projects. Use of the popular CMSIS-DSP library is also explained with several commonly used matrix operations.

The projects provided in the book can be used without any modifications in many applications. Alternatively, readers can base their projects to the ones given in the book during the development of their own projects. The author hopes that readers use the FRDM-MCXN947 development board in their future projects.

Hope you enjoy reading the book.

Dr. Dogan Ibrahim

Chapter 1 • The FRDM-MCXN947 Development Board

1.1 Overview

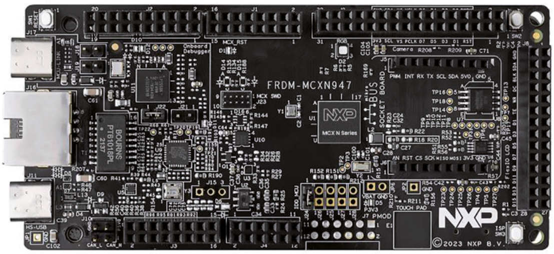
The FRDM-MCXN947 is a compact and scalable development board for rapid prototyping of the MCX N94 (and the N54) MCUs. The board offers industry standard headers for easy access to the MCU's I/Os, integrated open-standard serial interfaces, external flash memory and an on-board MCU-Link debugger.

This book provides detailed information and various projects on using this development board. In this chapter, you will get to know the most commonly used features of the FRDM-MCXN947 Development Board.

1.2 The FRDM-MCXN947 Development Board hardware

Figure 1.1 shows a close-up picture of the top and bottom views of the development board. A description of the major components on the board is shown in Figure 1.2.

Top view:



Bottom view:

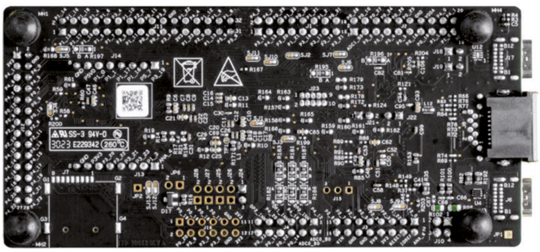


Figure 1.1: The development board.

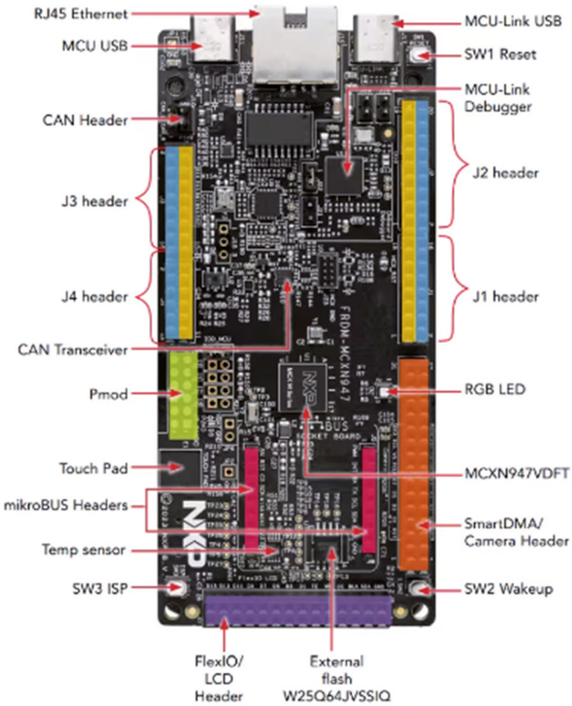


Figure 1.2: Components on the board.

A block diagram of the development board is shown in Figure 1.3. The basic features of the development board are:

- MCX-N947 Dual Arm 32-bit Cortex-M33 cores at 150 MHz each
- Up to 2 MB dual-bank flash memory
- Neural Processing Unit
- On-board MCU-Link debugger
- SPI/I2C/UART support
- CAN-FD transceiver
- Ethernet controller
- PMOD, mikroBUS, Arduino, DNP, FRDM connectors
- Temperature sensor
- Touchpad
- HS USB Type C connectors
- JTAG/SWD connector
- FlexIO/LCD connector
- SmartDMA/camera connector
- RGB LED
- Reset button
- Wakeup and ISP buttons

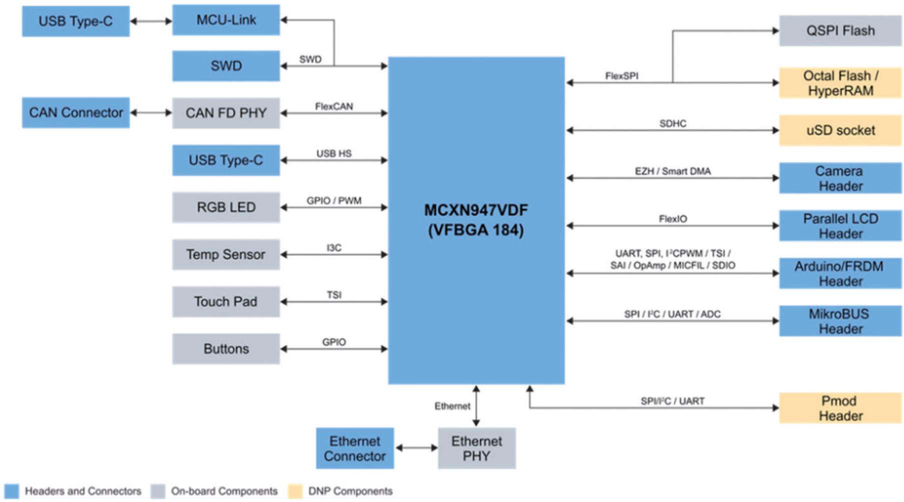


Figure 1.3: Block diagram of the development board.

A more detailed block diagram of the development board is shown in Figure 1.4.

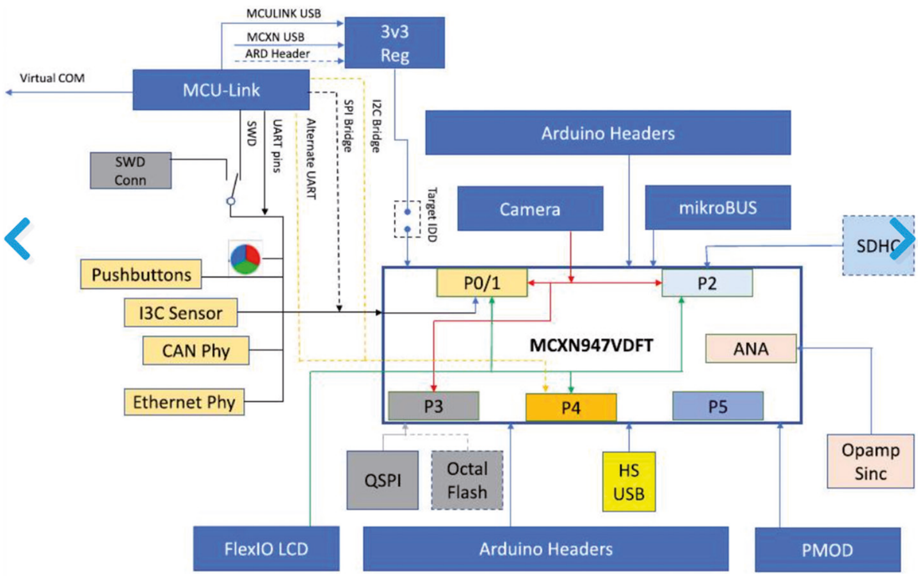


Figure 1.4: Detailed block diagram.

The MCX-N947 Dual Arm Cortex™ M33 microcontroller provides operating performance of 150 MHz. The board is equipped with Flash memory up to 2 MB, optional ECC RAM and external flash memory, which provides adequate capacity for storing data and programs. Additionally, it offers a range of advanced features such as a neural processing unit, PowerQuad, Smart DMA, Micro SD card slot, Ethernet PHY, and HS USB Type-C connectors.

The FRDM-MCXN947 board also has SPI/I2C/UART connectors, WIFI connectors, CAN-FD transceiver, a built-in MCU-Link debugger with **CMSIS-DAP** and JTAG/SWD connectors. The user interface includes RGB LEDs and Reset, ISP and Wakeup buttons, making it easy to control and monitor the device's operation. Additional tools, such as expansion cards and an Application Code Hub with software samples, are available in the **MCUXpresso Developer Experience** to support application development and system design using the FRDM-MCXN947 board.

1.2.1 On-board connectors

There are 9 connectors on the board with the names J1 to J9. Figure 1.5 shows the layout of the connectors on the board.

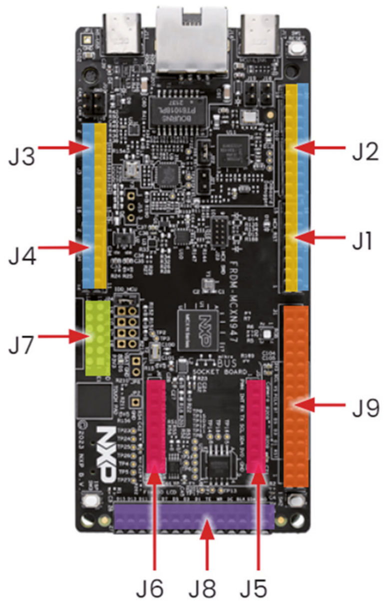


Figure 1.5: On-board connectors.

The pin layout of the connectors is shown in Figure 1.6, Figure 1.7, and Figure 1.8

J1

SAI	P3_21	15	16	P0_31	ARD_D7
SAI	P3_19	13	14	P1_2	ARD_D6
SAI	P3_17	11	12	P1_21	ARD_D5
SAI	P3_18	9	10	P0_30	ARD_D4
SAI	P1_21	7	8	P1_23	ARD_D3
SAI	P3_20	5	6	P0_29	ARD_D2
MC	P4_13	3	4	P4_2	ARD_D1
SAI	P3_16	1	2	P4_3	ARD_D0

J2

NC	NC	19	20	P4_1	ARD_D19
TSI	P1_0	17	18	P4_0	ARD_D18
TSI	P1_1	15	16	VDDA	ARD
TSI	P0_19	13	14	GND	ARD
TSI	P1_12	11	12	P0_25	ARD_D13
MC	P0_3	9	10	P0_26	ARD_D12
MC	P5_4	7	8	P0_24	ARD_D11
MC	P5_3	5	6	P0_27	ARD_D10
MC	P5_2	3	4	P0_10	ARD_D9
MC	ADC1_B0	1	2	P0_28	ARD_D8

J3

ARD	ADC0_A3	2	1	P2_0	MC
ARD	P3V3	4	3	P1_22	MC
ARD	RESET	6	5	P2_3	MC
ARD	P3V3	8	7	P2_2	MC
ARD	P5V0	10	9	P2_5	MC
ARD	GND	12	11	P2_4	MC
ARD	GND	14	13	P2_7	MC
ARD	VIN	16	15	P2_6	MC

Figure 1.6: Pin layout of connectors J1, J2 and J3.

J4

ARD_A0	ADC0_A0	2	1	P4_12	OPAMP
ARD_A1	ADC0_B0	4	3	OP0_INN	OPAMP
ARD_A2	P0_14	6	5	P4_16	OPAMP
ARD_A3	P0_22	8	7	OP1_INN	OPAMP
ARD_A4	P0_15	10	9	P4_20	OPAMP
ARD_A5	P0_23	12	11	OP2_INN	OPAMP

mikroBUS

J6		J5	
ANA_4	1	1	P3_19
P1_3	2	2	P5_7
P3_23	3	3	P1_16
P3_21	4	4	P1_17
P3_22	5	5	P1_1
P3_20	6	6	P1_0
P3V3	7	7	P5V0
GND	8	8	GND

J7

CS	P0_19	1	2	P0_20	
SDO	P0_16	3	4	P5_6	
SDI	P0_18	5	6	P3_3	SCL
SCK	P0_17	7	8	P3_2	SDA
	GND	9	10	GND	
	VDD	11	12	VDD	

Figure 1.7: Pin layout of connectors J4, J5, J6, and J7.

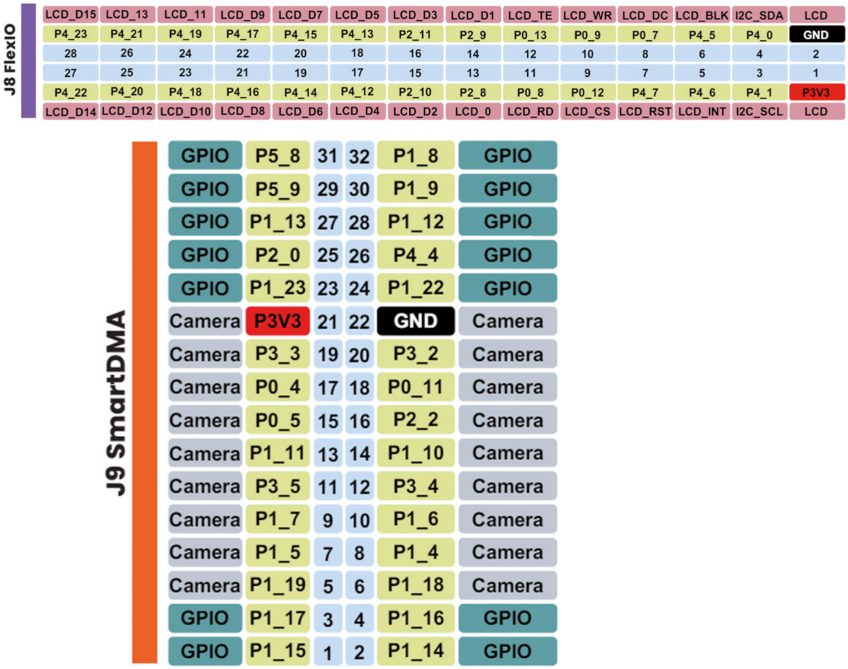


Figure 1.8: Pin layout of connectors J8, J9.

Figure 1.9 shows the development board and all the connectors in a single figure.

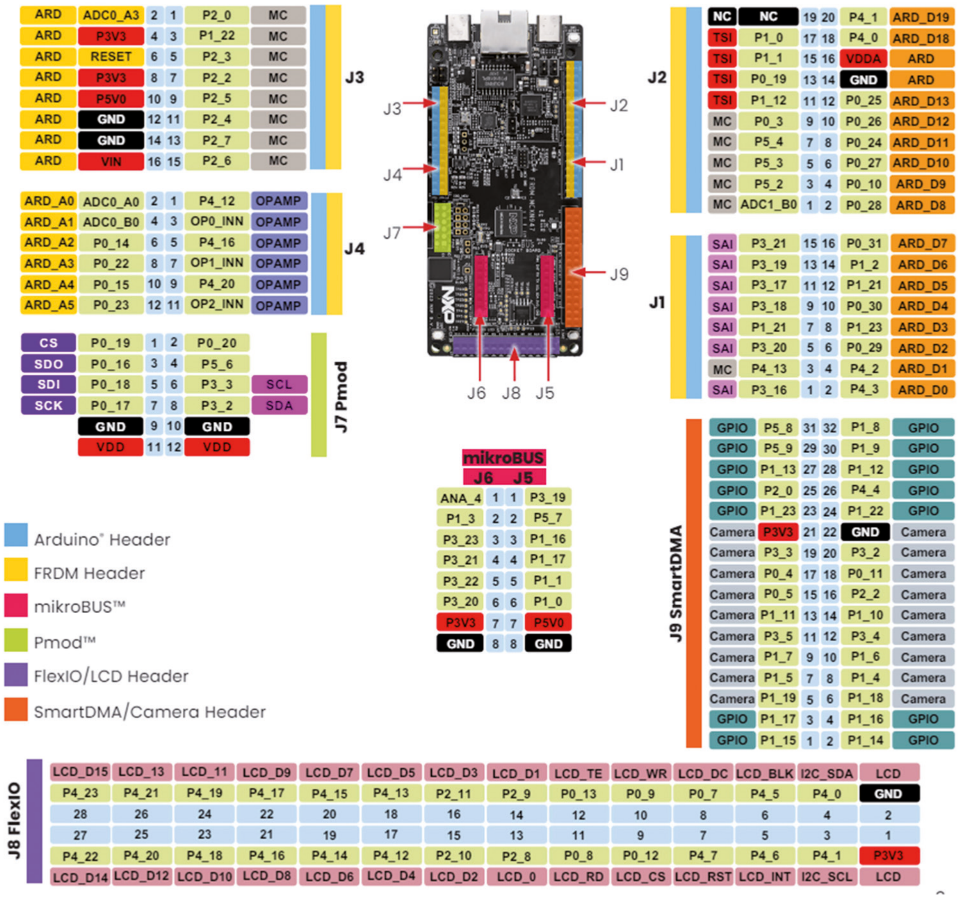


Figure 1.9: The development board and all the connectors.

1.2.2 The MCU

Some important MCU features are (refer to the **MCXNx4x Data Sheet** for full details):

- 150 MHz operation
- Up to 512 KB RAM
- 16 KB RAM cache
- 4 x 8 KB ECC RAM
- 256 KB ROM
- 2 x 16-bit ADC
- Up to 75 ADC channels
- One integrated temperature sensor per ADC
- 3 high speed comparators
- 2 x 12-bit DAC
- 1 x 14-bt DAC
- Accurate voltage reference
- 3 x operational amplifiers

- 5 x 32-bit general purpose timers
- USB high-speed communication
- DSP accelerator
- Tamper detect
- Neural Processing Unit
- SCTimer/PWM
- LPTimer
- RTC with calendar
- Watchdog timer
- Frequency measurement timer
- 2 x DMA modules
- 10 x LP Flexcomms each support SPI, I2C, UART
- 2 x FlexCAN with FD
- Programmable Logic Unit
- 2 x FlexPWM
- 2 x Quadature encoder
- 1 x Event generator
- SINC filter module
- Digital PDM microphone (connection of up to 4 MEMS microphones)
- Capacitive touch sensor interface
- Up to 124 GPIOs
- Support 1.71 V – 3.6 V I/O supply
- Operating voltage 1.71 V to 3.6 V
- Power-down, deep power-down, and deep sleep modes
- Max I/O current 3 mA
- Output HIGH/LOW current (total for all ports) 100 mA maximum

1.2.3 Jumpers on the board

Figure 1.10 shows the jumpers on the development board. The description of these jumpers is given in Table 1.1.

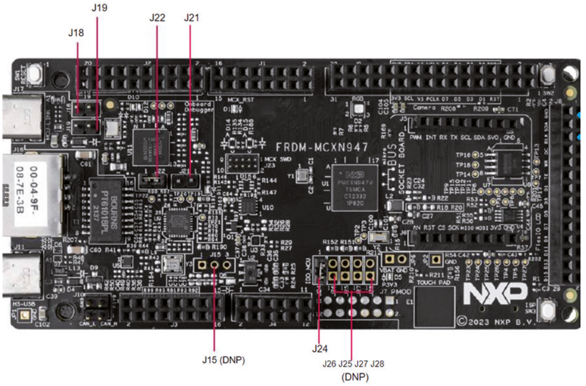


Figure 1.10: Jumpers on the board.

Part identifier	Jumper type	Description
J18	1x2 pin header	• Open (default setting): MCU-Link VCOM port is enabled. • Shorted: Sends a low signal on LPC_HW_VER_6 to disable MCU-Link VCOM port
J19	1x2 pin header	• Open (default setting): Enables the MCU-Link SWD feature • Shorted: Sends a low signal on LPC_HW_VER_7 to disable the onboard MCU-Link SWD feature *Note: This configuration is required to enable target MCU debug through an external debug probe.*
J21	1x2 pin jumper	MCU-Link (LPC55S69) force ISP mode jumper: • Open (default setting): MCU-Link follows the normal boot sequence (MCU-Link boots from internal flash if a boot image is found). With the internal flash erased, the MCU-Link normal boot sequence falls through to ISP boot mode. • Shorted: MCU-Link is forced to ISP mode (USB). Use this setting to reprogram the MCU-Link internal flash with a new image or use the MCUXpresso IDE with the CMSIS-DAP protocol. *Note: By default, MCU-Link flash is preprogrammed with a version of CMSIS-DAP firmware.*
J22	1x2 pin jumper	MCU-Link SWD clock enable jumper: • Open: MCU-Link SWD clock is disabled. • Shorted (default setting): MCU-Link SWD clock is enabled.
J24	1x2 pin header	Pin 1-2 shorted (default setting): P3V3_MCU is sourced from the P3V3 power supply
J25 (DNP)	1x2 pin jumper	DNP by default. If shorted, VDD_IO_USB is powered by P3V3_MCU through the jumper. *Note: BY default, VDD_IO_USB is powered from the P3 V3_MCU supply through a zero-ohm shunt resistor.*
J26 (DNP)	1x2 pin jumper	DNP by default.
J27 (DNP)	1x2 pin jumper	DNP by default. If shorted, VDD_BAT is powered by P3V3_MCU through the jumper. *Note: BY default, VDD_BAT is powered from the P3V3_ MCU supply through a zero-ohm shunt resistor.*
J28 (DNP)	1x2 pin jumper	DNP by default. If shorted, VDD_CORE_SYS is powered by P3V3_MCU through the jumper. *Note: BY default, VDD_CORE_SYS is powered from the P3V3_MCU supply through a zero-ohm shunt resistor.*

Table 1.1: Jumpers on the board.

1.2.4 Push buttons on the board

Tactile buttons are populated on the FRDM-MCXN947 development board for human machine interaction (HMI). Each of the buttons has a 0.1 µF bypass capacitor for debouncing and pads for external pull-up resistors. Table 1.2 gives a description of the buttons on the board.

Part identifier	Switch name	Description
SW1	Reset button (MCXN947 RST)	Pressing SW1 resets the target MCU that causes board peripherals to reset to their default states and execute the boot code. When SW1 is pressed, the reset LED D1 turns ON.
SW2	Wakeup button	SW2 is a general-purpose input and a low-power wake up unit (WUU) pin. Pressing SW2 gives a low level on P0_23/WAKEUP_B, otherwise, it is a high level on P0_23/WAKEUP_B.
SW3	In-system programming (ISP) mode switch	SW3 is an ISP mode switch and can also act as a general-purpose input. Pressing SW3 gives a low level on P0_6/ISPMODE_N-DEBUG, otherwise, it is a high level on P0_6/ISPMODE_N-DEBUG.

Table 1.2: Buttons on the board.

Reset (SW1): connected to the Reset input of the MCU
Wakeup (SW2): connected to port P0_23
ISP (SW3): connected to P0_6
Touch Pad: connected to P1_3

1.2.5 LEDs on the board

There are 3 LEDs on the board as described in Table 1.3.

Part identifier	LED color	LED name / function	Description
D1	Red	Reset LED	Indicates system reset activity. When board reset is initiated, for example, by pressing the SW1 reset button, the D1 LED turns ON.
D2	Red/green/blue	RGB LED	User application LEDs. Each of these LEDs can be controlled through a user application. • Red LED connects to target MCU pin P0_10 • Green LED connects to target MCU pin P0_27 • Blue LED connects to target MCU pin P1_2
D5	Green	P3V3 PWR ON	Indicates P3V3 power on status. When P3V3 is available on board, D5 turns ON.

Table 1.3: LEDs on the board.

RGB LED: Connected in common-cathode mode. i.e. An LED is ON when the corresponding port pin is LOW. Red LED connected to P0_10, Green LED to P0_27, and the Blue LED to P1_2

1.2.6 Ethernet interface

On the FRDM-MCXN947 board, the Ethernet controller connects to an RJ45 connector through an Ethernet PHY transceiver. The transmit, receive, and other Ethernet signals are on the P1 port pins. The FRDM-MCXN947 only supports RMII configuration. For this reason, the TXD3 and TXD2 pins of the Ethernet PHY (LAN8741A-EN) have been grounded through resistors R68 and R67, respectively.

1.2.7 FlexCAN interface

The controller area network (FlexCAN) is a full implementation of the CAN protocol specification, the CAN with flexible data rate (CAN FD) protocol, and the CAN 2.0 version B protocol, which supports both standard and extended message frames and long payloads. The target MCU (MCXN947) supports two CAN (w/wo FD) controllers (CAN0 to CAN1).

On FRDM-MCXN947, only the CAN0 controller is used. The CAN0 controller connects to a 4-pin CAN header through a CAN transceiver (TJA1057GTK/3Z). The CAN0_TXD and CAN0_RXD signals are through ports P1_10 and P1_11, respectively.

Jumper J10 is a 4-pin CAN header. It is connected to the CAN0 bus and allows external connection with the bus. Table 1.4 shows the CAN header pin layout.

Pin	Signal	Description
1	CAN1_H	CAN transceiver high signal
2	CAN1_L	CAN transceiver low signal
3	P5V0	5 V power supply
4	GND	Ground

Table 1.4: CAN header pin layout.

1.2.8 I³C sensor interface

The FRDM-MCXN947 board includes one P3T1755DP digital temperature sensor to demonstrate the I³C capabilities of the target MCU. This sensor device allows for 32 I3C provisional IDs, supports the full operating voltage of the board, programmable overtemperature alerts, 12b resolution, and has an accuracy of ±0.5 °C (maximum) from −20 °C to +85 °C.

The 7-bit I2C address of the sensor device is 0b1001000 (0x48). The sensor device connects to the I3C1 controller of the device through P1_[16:17] Port, where P1_16 is the SDA and P1_17 is the SCL pins. 4.7 kΩ internal pull-up resistors are provided for the I3C bus.

1.2.9 SD card interface

On the FRDM-MCXN947 board, the uSDHC controller connects to the SD card connector (J12) (not populated by default). The SD card detect pin is an open switch that shorts with GND when the card is inserted. The interface pins are:

DET:	P2_1
DAT1:	P2_2
DAT0:	P2_3
CLK:	P2_4
CMD:	P2_5
CD/DAT3:	P2_6
DAT2:	P2_7

1.3 Starting Up – demo program

A demo program has been pre-loaded into the memory of the development board for testing purposes. Connect a type-C USB cable from connector J17 (see Figure 1.11) to a host computer or power supply to power up the board and run the demo program. You should see the RGB LED blinking at a steady rhythm.

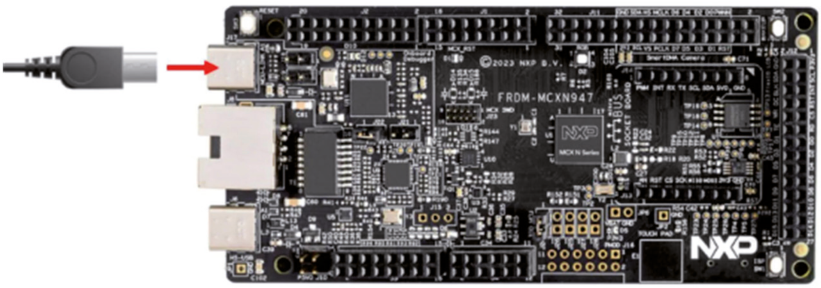


Figure 1.11: Running the demo program.

1.4 MCX N Series microcontrollers

MCX N series of microcontrollers are highly integrated, low-power MCUs designed with intelligent peripherals and on-chip accelerators that provide the ultimate balance of performance and energy consumption. They are based on the dual Arm Cortex-M33 core operating up to 150 MHz. Table 1.12 shows the basic features of the MCX N series of microcontrollers.

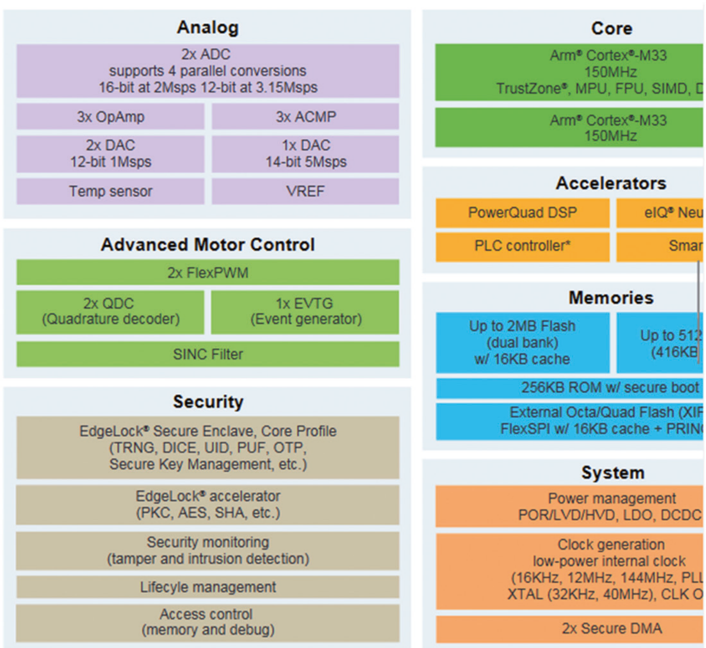


Figure 1.12: MCX N series features.

Table 1.5 shows the basic specifications of all the MCUs in the MCX N series of microcontrollers. The one used in this book is highlighted for comparison purposes.

Part Number	Flash (KB)	SRAM	NPU	FlexSPI	PLC Controller	USB HS	DAC	Op Amp	Flexcomm	CAN FD	Packages
MCXN235VDF	512	192 KB (160 KB w/ ECC)	No	No	No	Yes			8	2	VFBGA184
MCXN235VNL	512	192 KB (160 KB w/ ECC)	No	No	No	Yes			8	2	HLQFP100
MCXN236VDF	1024	352 KB (288 KB w/ ECC)	No	No	No	Yes			8	2	VFBGA184
MCXN236VNL	1024	352 KB (288 KB w/ ECC)	No	No	No	Yes			8	2	HLQFP100
MCXN546VDF	1024	352 KB (288 KB w/ ECC)	Yes	Yes	No	Yes	1 x 12b		10	1	VFBGA184
MCXN546VNL	1024	352 KB (288 KB w/ ECC)	Yes	Yes	No	Yes	1 x 12b		10	1	HLQFP100
MCXN547VDF	2048	512 KB (416 KB w/ ECC)	Yes	Yes	No	Yes	1 x 12b		10	1	VFBGA184
MCXN547VNL	2048	512 KB (416 KB w/ ECC)	Yes	Yes	No	Yes	1 x 12b		10	1	HLQFP100
MCXN946VDF	1024	352 KB (288 KB w/ ECC)	Yes	Yes	Yes	No	2 x 12b + 1 x 14b	3	10	2	VFBGA184
MCXN946VNL	1024	352 KB (288 KB w/ ECC)	Yes	Yes	Yes	No	2 x 12b + 1 x 14b	3	10	2	HLQFP100
MCXN947VDF	2048	512 KB (416 KB w/ ECC)	Yes	Yes	Yes	Yes	2 x 12b + 1 x 14b	3	10	2	VFBGA184
MCXN947VNL	2048	512 KB (416 KB w/ ECC)	Yes	Yes	Yes	Yes	2 x 12b + 1 x 14b	3	10	2	HLQFP100
MCX-N5xx-EVK	MCX N54x full evaluation kit										VFBGA184
MCX-N9xx-EVK	MCX N94x full evaluation kit										VFBGA184
FRDM-MCXN236	MCX N236 FRDM development board										VFBGA184
FRDM-MCXN947	MCX N947 FRDM development board										VFBGA184

Table 1.5: MCX N series of microcontrollers.

Chapter 2 • MCUXpresso and the Software Development Kit (SDK)

2.1 Overview

In this book we will be using the **MCUXpresso SDK** and the **IDE** for developing projects using the development board. **MCUXpresso IDE** is an Eclipse based development environment for NXP MCUs using Cortex-M cores. It supports many processors, including the MCXN947, the i.MX RT, LPC and Kinetis devices, devices from Cortex-M0+ to most of the Cortex family of processors. The SDK and the IDE must be installed before they can be used.

2.2 Installing the MCUXpresso IDE and SDK

The steps to install the **MCUXpresso SDK** and **IDE** are given below:

- Go to following web site: https://www.nxp.com/document/guide/getting-started-with-frdm-mcxn947:GS-FRDM-MCXNXX?section=get-software

- Click on **GET MCUXPRESSO IDE** (Figure 2.1)

Figure 2.1: Click on the link.

- Click **DOWNLOADS** (Figure 2.2)

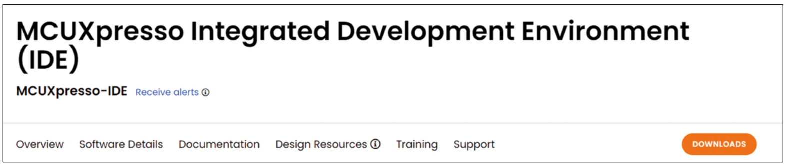


Figure 2.2: Click Downloads.

- Click to Download the **MCUXPRESSO IDE** (Figure 2.3)

Figure 2.3: Click to Download.

- You will have to create an account and login to the NXP site.
- Select your operating system and click to download the file (Figure 2.4). At the time of drafting the book the file was named: **MCUXpressoIDE_11.9.1_2170.exe**

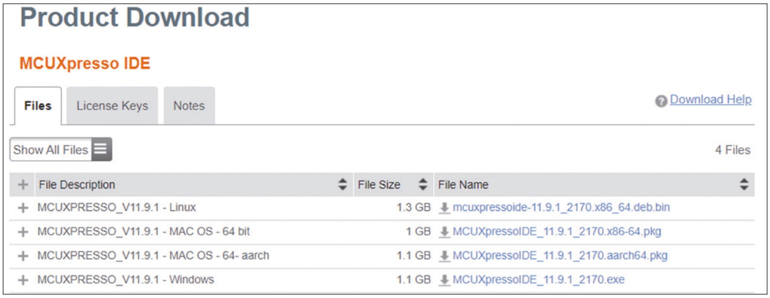


Figure 2.4: Select your operating system and download.

- Click on the file to install the IDE.
- Click to start the IDE and give a workspace name.
- Click **Launch.**
- Click Download and Install SDKs (Figure 2.5).

Figure 2.5: Install SDKs.

- Select board **frdmmcxn947** (Figure 2.6)

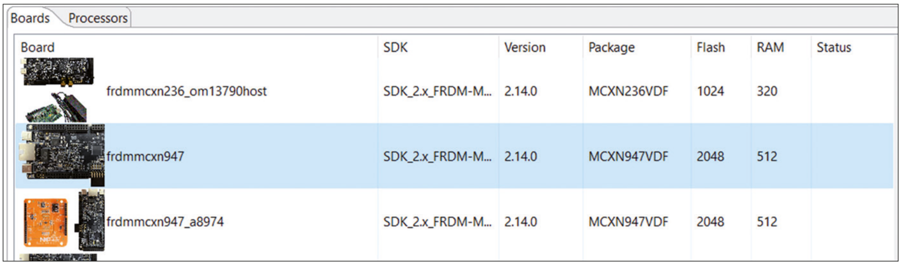


Figure 2.6: Select frdmmcxn947.

- Click Install and Import Examples

This completes the installation of the **MCUXpresso IDE**.

2.3 Testing the installation

Now that the installation is complete, you should check the installation by compiling and running one of the demo programs supplied. Here, you will compile and run the demo example called **led_blinky** which flashes the on-board LED. The steps are as follows:

- Start the **MCUXpresso IDE.**

- Create Workspace **ledtest** (Figure 2.7) and click **Launch.**

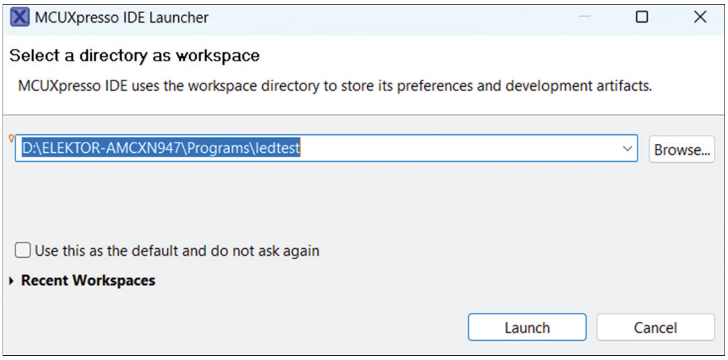


Figure 2.7: Create the ledtest workspace.

- Click **IDE** to be in IDE mode (Figure 2.8).

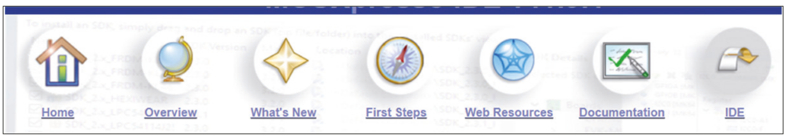


Figure 2.8: Click IDE.

- Click **Import SDK Example(s)...** under the **MCUXpresso IDE – Quickstart Panel.**

- Click to select board **frdmmcxn947** (Figure 2.9). Click **Next.**

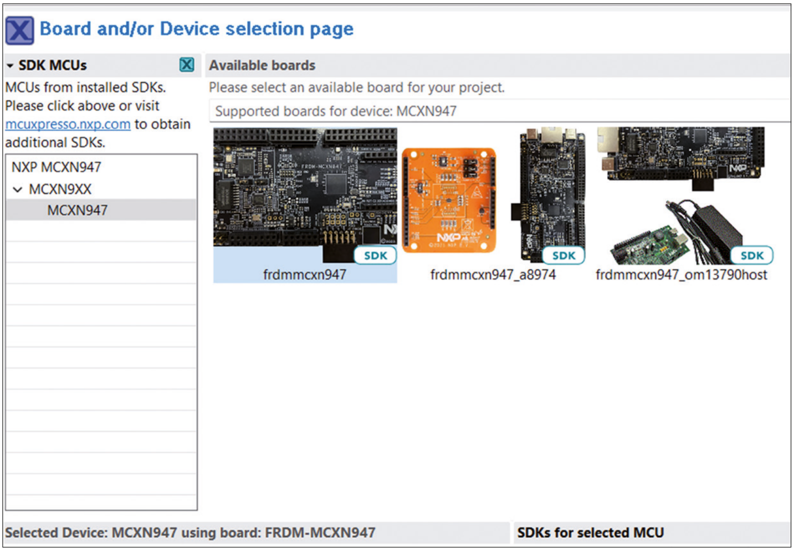


Figure 2.9: Select board frdmmcxn947.

- Click to expand **demo_apps** (Figure 2.10).

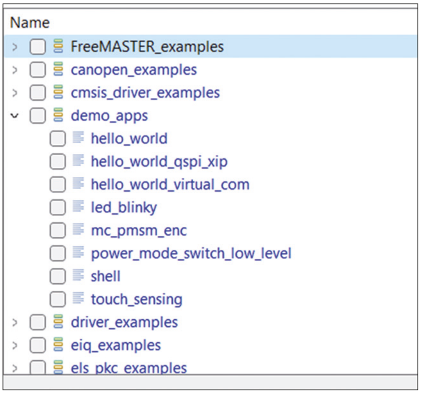


Figure 2.10: Expand demo_apps.

- Click to select **led_blinky** and click **Finish.**

- You should see the **File Explorer** at the left window, the program code in the middle window, the header filenames and functions used at the right-hand window, and the console terminal etc at the bottom of the window (Figure 2.11). Do not worry if you do not understand how the program works at this stage!

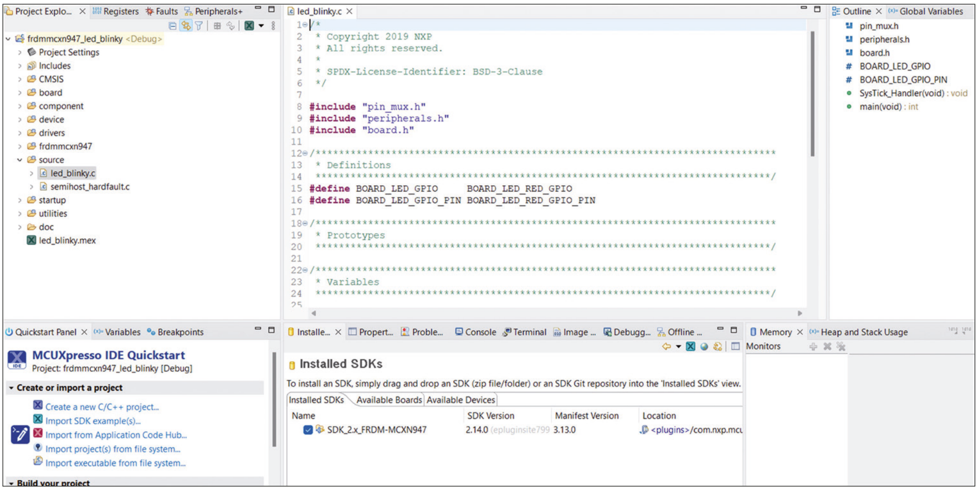


Figure 2.11: The IDE window.

- The program code listing is shown in Figure 2.12.

```
/*
 * Copyright 2019 NXP
 * All rights reserved.
 *
 * SPDX-License-Identifier: BSD-3-Clause
 */

#include "pin_mux.h"
#include "peripherals.h"
#include "board.h"

/*******************************************************************************
 * Definitions
 ******************************************************************************/
#define BOARD_LED_GPIO     BOARD_LED_RED_GPIO
#define BOARD_LED_GPIO_PIN BOARD_LED_RED_GPIO_PIN

/*******************************************************************************
 * Prototypes
 ******************************************************************************/

/*******************************************************************************
 * Variables
 ******************************************************************************/

/*******************************************************************************
 * Code
```