

# FAQs

## What makes XDK so great?

XDK is a toolkit with state-of-the-art sensor technology and ready-to-use software packages. There is no need for component selection, hardware assembly, or deployment of a real-time operating system, or a development environment

## What are the included deliveries?

In addition to the device in its housing, with built-in lithium ion-battery, a break-outboard for easy access to additional MCU functionality, incl. connector and USB cable, a mounting plate and screws, access to the community and to useful documents (getting started guide and data sheet) are also delivered.

## What are the dimensions of XDK?

The product dimensions are: 60 mm x 42 mm x 22 mm. Weight: 43g

## Which sensors are included?

The device includes a full array of Micro-Electromechanical sensors (MEMS). These are the following: Accelerometer, Magnetometer, Gyroscope, Humidity/Temperature/Pressure Sensor, Acoustic noise sensor, Digital light sensor.

## What hardware interfaces does it have?

The XDK comes with a USB interface to program via boot loader. It also provides a JTAG debug interface for in-depth debugging of the MCU via JTAG debug probe. The extension bus has 23 GPIO pins, including UART, I2C and SPI in addition to 2.5V and 3.3V power pins (power pins limited to 100mA)

## What SD card sizes does it support?

XDK supports Micro SD cards up to version 2.0 of the SD specification (card capacity up to 32 GB) with FAT (File Allocation Table) filesystems. Note: SDXC is NOT supported! We recommend anything up from Class 4 performance wise.

## What communication technologies are on the XDK?

Wireless LAN (IEEE 802.11 b/g/n) and Bluetooth LowEnergy (IEEE 802.15.1) are implemented on the device. Antennas are integrated in to the XDK.

## How long is the battery life?

Depends heavily on the usage. It can range from two weeks up to several months. To increase battery life, make sure to disable sensors you don't need.

## **Does XDK have any certifications?**

XDK is certified for Europe (CE), US (FCC) and Canada (IC).

## **Do I need any other components?**

The XDK was designed to get started quickly, therefore the only thing you need is the included starter kit and your PC. Drivers for all system components are included. Functional extendibility also is possible via the included extension board and ribbon cable. A debugger is not required but can be used if desired.

## **Which operating systems are supported?**

XDK works with Windows 7 and higher.

## **What are the supported programming languages?**

The official programming language of the XDK is "C".

## **Will I get support after buying?**

You can get help in different ways:

The user guides on [xdk.io](http://xdk.io) offers detailed information about the device, its function and features, and specific technical information

## **How do I power and charge the device?**

The XDK device is powered with a lithium-ion rechargeable battery and can be charged via standard USB charging. You will find a USB cable in our starter kit.

## **What do the different colours of the status LEDs on XDK mean?**

The LEDs can have different states. If the green LED is continuous, the XDK is charging. If it's flashing, the safety timers have expired. If it's off, charging is complete and the device is fully charged. The yellow LED means that the boot loader mode is active, and USB connected. If the red LED is on, the XDK is in boot loader mode.

## **What are the functions of the two mechanical push buttons?**

The two push buttons present on the XDK device are programmable or can be defined by the user. Interrupts are supported for both buttons.

## **How to connect XDK with the Extension Board?**

The extension board offers a simple way to implement additional functions and can be connected to XDK via ribbon cable (included in delivery) with the extension connector of the XDK

## What is the difference between accelerometer, gyroscope and magnetometer?

**Accelerometer:** An accelerometer measures the acceleration forces. These forces may be static (like force of gravity) or dynamic (moving or vibrating the accelerometer). For example, a dynamic accelerometer measures gravitational pull to determine the angle at which a device is tilted with respect to the Earth. By sensing the amount of acceleration, users can analyze how the device is moving.

**Gyroscope:** A gyroscope is a device for measuring or maintaining orientation, based on the principles of angular momentum. It is used in navigation on aerial vehicles, compasses, machinery, hardware monitoring etc. A Gyroscope is used in an aircraft to aid in indication of the rate of rotation around the aircraft roll axis. The main difference between the two devices is simple: A gyroscope can sense the rotation, whereas the other cannot.

**Magnetometer:** A magnetometer is a device that measures magnetic fields, by sensing and measuring the magnetic flux density. Many smart phones contain magnetometer; apps exist that serve as compasses.

## How to load a program on XDK?

Loading a program is very simple: Just execute the following steps:

1. Connect your XDK device to the system via USB cable that was included in the delivery.
2. Open the XDK workbench
3. Load/import the project
4. Click on the "Flash" button

## Flashing does not work. Why?

If you are having troubles flashing the XDK or getting errors while doing so, try to put the XDK into the bootloader mode manually. For this, turn the XDK off. Press button 1 and hold it, turn the XDK on again while still pressing the button. Once the red LED turns on, release the button.

## How to enable/disable sensors in MITA application

Please use the interface functions of the respective sensor module from for example `xdk110\Common\source\Sensor\NoiseSensor.c`. From application, for example the noise sensor can be used by sequentially calling the functions,

```
NoiseSensor_Setup()
```

```
NoiseSensor_Enable()
```

```
NoiseSensor_ReadRmsValue()
```

You can refer VirtualXdkDemo application on usage of noise sensor.

## **Is it possible to connect multiple XDK connection to an IoT Platform?**

Yes, though the XDK's must each be registered as a unique "thing" (for example in AzureIoT Hub) and use that as the Client ID to connect. Otherwise, the XDK's will fail to maintain connection due to conflicting Device ID's.

## **What is the performance of the microcontroller (which evaluates signal from the sensor)?**

The performance of our 32-bit microcontroller ARM Cortex M3 offers a performance of 1.25 DMIPS/MHz

## **How do I use the temperature sensor properly?**

The hardware placements of the XDK are that way, that the WiFi module is on the other side of the PCB than the sensors. Therefore, if accurate temperature sensor measurement is required, the recommendation is to communicate via Blue Tooth Low Energy, BLE to avoid self-heating effect of the WiFi Module. The recommendation mounting orientation is that the sensor are down and the jumper for the extension board is up.

## **XDK Temperature measurement is not accurate. Why?**

The temperature sensor reading is correct. But this measures the board (PCB) temperature as sensed by the BME280. Components like the battery and the WLAN module tend to heat-up the board by up to 5°C over the ambient temperature. In steady state, this offset will be static, and as you noticed, it will be similar amongst devices of the same production year, so my recommendation is to calibrate this offset using your program/use-case with an external thermometer.  $T\_Offset = T\_reference - T\_BME280$ . So, every time you read out the temperature you can add this offset to obtain the real temperature  $T\_real = T\_BME280 + T\_Offset$

## **Is it possible to flash the XDK over the air?**

Yes, FOTA feature is available on the XDK. You can refer below documentation, which will be useful to implement the FOTA feature:

[https://rbdevportal.secure.footprint.net/Documentation/html/\\_x\\_d\\_k\\_bootloader.html](https://rbdevportal.secure.footprint.net/Documentation/html/_x_d_k_bootloader.html)

[https://rbdevportal.secure.footprint.net/Documentation/html/group\\_b\\_c\\_d\\_s\\_f\\_o\\_t\\_a.html#details](https://rbdevportal.secure.footprint.net/Documentation/html/group_b_c_d_s_f_o_t_a.html#details)

## **Is it possible to enhance the WIFI transmit range of the XDK?**

In order to maintain the CE/FCC certification and regulations, the XDK110 Radio-Frequency settings cannot be changed. If your solution demands higher transmit range, the XDK-Lora-extension would be the perfect solution.

## What Debugger models are compatible with the XDK?

Any Cortex-M compatible debugger with a SWD interface is able to flash and debug XDK. Though only Segger jLink powered devices are supported out-of-the-box by the XDK workbench, meaning any other solution will require some extra work by the user to get the flashing and debugging work. Just be aware to use properly licensed Segger Tools.

## Is it possible to charge the XDK through a powerbank?

It's important to take the maximum of 500mA at 5V into account when selecting the power supply or power bank. Exceeding this could damage the device permanently. Therefore, it is safe to connect the XDK permanently to a power bank or any other kind of power source.

## Connecting to my enterprise WiFi does not work. Why?

The WiFi chip in the XDK has to be upgraded. In order to do that, please flash the 2 applications found here: <https://developer.bosch.com/web/xdk/wi-fi-enterprise#>

1. WiFiHost Programming: This application will update your WiFi chip
2. Enterprise WiFi: This application will connect with the Enterprise network configured

## Is the upgraded Wifi-Chip version backwards compatible with projects, that ran on lower Wifi-Chip version?

The new Wifi-chip version will work with older projects/applications as well.

## Why does my XDK stop charging?

Usually, the XDK never stops charging when used with the external power supply. However, when you use some powerbanks or USB ports, the XDK will switch off after 1-2 hours of inactivity. This leads to XDK stop charging.

## Can you reboot the XDK without the XDK workbench?

You can not reboot the XDK with only the device itself. But it is possible to reboot the XDK by using any serial terminal such as HTerm, TeraTerm software, etc. When XDK is in Application mode, you can send the following commands in any serial terminal:

#reBoot\$            Reboot XDK and go into bootloader mode

#reSet\$             Reboot XDK and go into application mode

When the XDK is in Bootloader mode, you can send the following command in any serial terminal:

b            Boot application            This command will start the uploaded application

**Why does my XDK has issues connecting to an IoT Hub (Azure, AWS, Bosch IoT Suite, etc.)?**

One common issue that get overseen often is the length of the namespace. If the DeviceID or an Instance has more than 14 characters, XDK device is not able to establish the connection with an IoT hub. Therefore, create a subscription on an IoT hub with DeviceID name, Instance name and/or Namespace name with less than 14 characters.

**Are C-Programming skills necessary to write any application?**

No. The XDK-Workbench supports the Eclipse Mita programming language. Eclipse Mita is a programming language that is focused on making Internet-Of-Things things easier to program, especially for developers without an embedded development background. It aims to increase programmer productivity, enable fast exploration of use-cases and tries to make you feel right at home on an embedded device.

**Is it possible to copy config-file credentials on projects/applications without using an SD-card?**

The XDK-Workbench always checks if an SD-card is inserted in the XDK. If so, it proceeds to check, if a config file is saved on the XDK. If yes, the workbench takes the credentials of the config file in the SD-card and overwrites it in the root config. If no SD-card is inserted, the XDK-Workbench proceeds to use the root config file. Therefore, it is possible to copy the config file on XDL without an SD-Card. You just need to edit the root config file in the XDL120 project with the needed credentials, such as WiFi-SSID and Wifi-Password. If done correctly, the XDK should be able to flash and the application should start with the root config file.