XDK Extension Board Overview

4DI-Extension

Document revision	0.5
Document release date	**December 19th, 2019**
Workbench version	3.0.0 and above
Document number	BCDS - XDK 4DI-Extension
Technical reference code(s)	
Notes	The data in this document is for XDK 4DI-Extension version 1 and version 2. Data in this document is subject to change without notice. Product photos and pictures are for illustration purposes only and may differ from the real product's appearance. **Advance information – Subject to change without notice**

Table of Contents

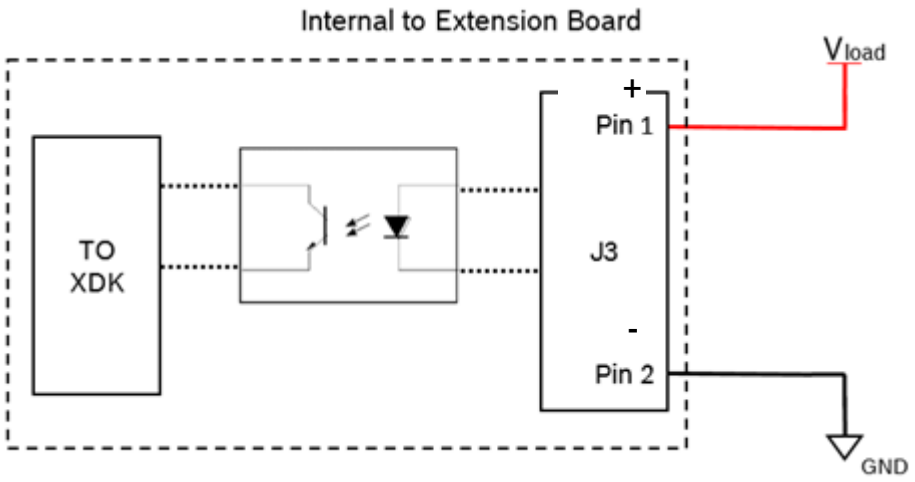
1 Introduction

The XDK 4DI-Extension board consists of two main elements:
-four digital inputs rated up to 40V (in both versions, 1 and 2)
-and one 12VDC output (only in version 2)

This document will go over how to use each element of the board and describe the maximum rating of each element.

2 Digital Inputs

The board, version 1 and version 2, is equipped with four digital inputs rated up to 40V. These inputs can be wired in at terminal block J3. The positive and negative terminals are clearly marked on the board and Figure 3 shows a typical wiring diagram. To isolate the digital input from the XDK an optocoupler was used. The board also makes use of voltage regulator diodes to prevent overvoltage and reverse polarity. A bipolar transistor in constant current mode is used to manage the power consumption on the input.



Basic Wiring Diagram of a Digital Input

The digital input channels work with active-low logic. Active-low means that XDK reads logical HIGH if low voltage signal is applied to the input. If the voltage level is above the threshold, XDK gets logical LOW.

Logic level read by XDK	Voltage level on J3
Logical HIGH	0V to 3.5V typical
Logical LOW	3.5V typical to 40V

Chanel of Extension	Pin on Micro Controller (Extension Bus)
Chanel 1	PC1 (A3)
Chanel 2	PC2 (A4)
Chanel 3	PC3 (A5)
Chanel 4	PC4 (A6)

2.1 **Arrangement of the inputs**

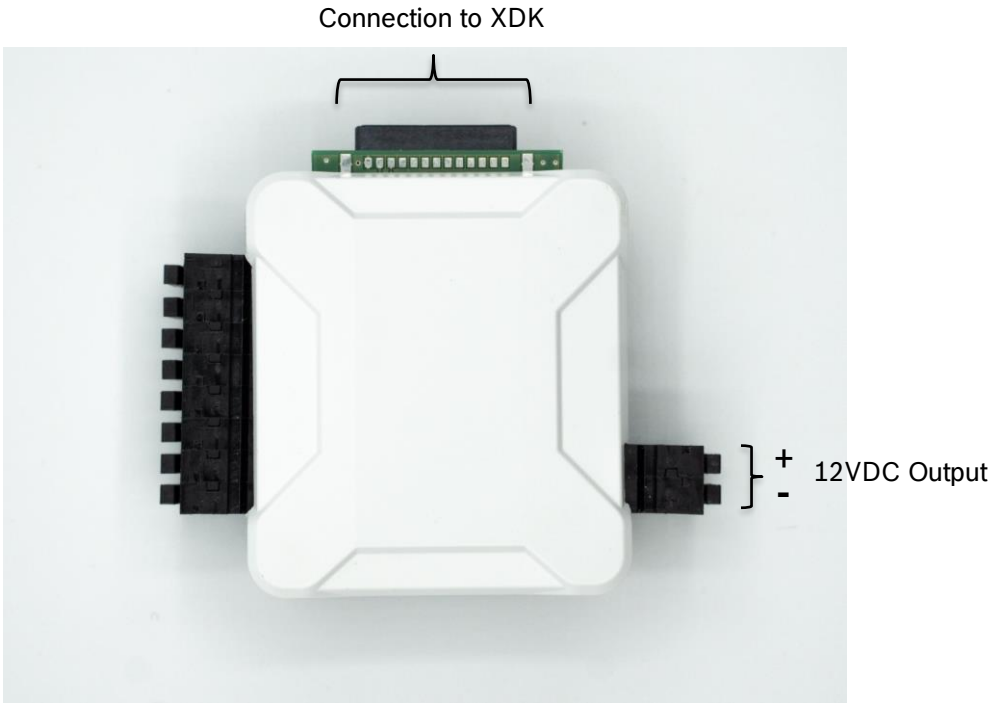
Attention: Check polarity!

3 12VDC Output

The board, in version 2, is furthermore equipped with a 12VDC output. This output can be wired in at terminal block J4. The positive and negative terminals are clearly marked. This output allows connecting “dry contacts” without the need for a separate DC-Powersupply.

As this 9V-Output uses the XDK as power supply, the output is intended only to drive the Opto coupler. The output is not intended to supply power to external Sensors!

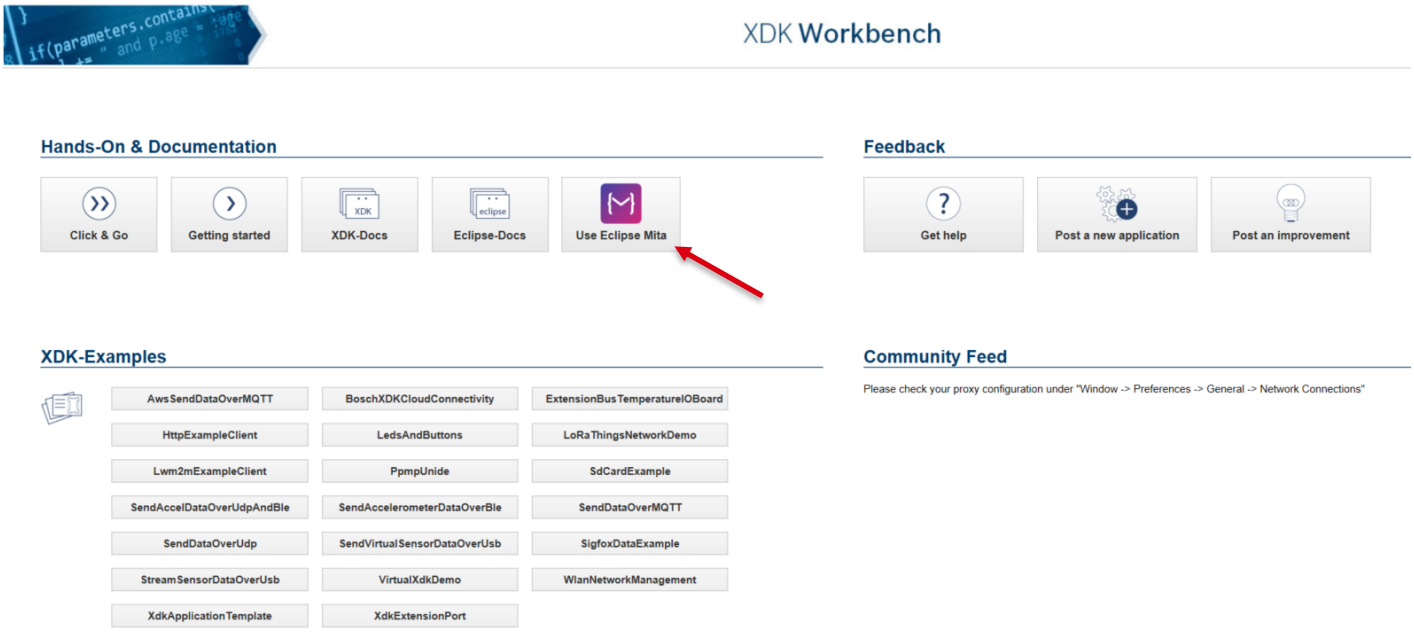
3.1 Arrangement of the output



Attention: Check polarity!

4 Getting Started

1. To get started, simply download and open the “XDK workbench” available for download on www.xdk.io



XDK Workbench start-up screen with example selection

2. Select “Use Eclipse Mita” on the startscreen

3. Copy Mita-Code-Example into file “application.mita”

application.mita

```
/**
 * Welcome to Eclipse Mita.
 *
 * Not sure what to do now?
 * Check out the "Getting started" guide on https://mita.io.
 */

package main;
import platforms.xdk110;

setup PIN : GPIO {
    var input_PC1 = digitalIn(PC1,NoPull);
    var input_PC2 = digitalIn(PC2,NoPull);
    var input_PC3 = digitalIn(PC3,NoPull);
    var input_PC4 = digitalIn(PC4,NoPull);
}


every 1 second {
    println(`state PC1: ${PIN.input_PC1.read()}`);
    println(`state PC2: ${PIN.input_PC2.read()}`);
    println(`state PC3: ${PIN.input_PC3.read()}`);
    println(`state PC4: ${PIN.input_PC4.read()}`);
}

```

Mita source code example:

```
/**
 * Welcome to Eclipse Mita.
 *
 * Not sure what to do now?
 * Check out the "Getting started" guide on https://mita.io.
 */

package main;
import platforms.xdk110;

setup PIN : GPIO {
    var input_PC1 = digitalIn(PC1,NoPull);
    var input_PC2 = digitalIn(PC2,NoPull);
    var input_PC3 = digitalIn(PC3,NoPull);
    var input_PC4 = digitalIn(PC4,NoPull);
}


every 1 second {
    println(`state PC1: ${PIN.input_PC1.read()}`);
    println(`state PC2: ${PIN.input_PC2.read()}`);
    println(`state PC3: ${PIN.input_PC3.read()}`);
    println(`state PC4: ${PIN.input_PC4.read()}`);
}
```

4. Connect your XDK via a USB cable to your computer

5. Flash “EclipseMitaApplication” on the XDK.
   Flashing: Select “EclipseMitaApplication” then press Flash!

5 Change history

	Changes:	**Date:**	**Responsable:**
0.1	Initial creation		M. Röhrig
0.2	Updated Layout and 4 Getting started	06.05.2019	S. Baumann
0.3	Updated Layout	29.11.2019	S. Baumann
0.4	New pictures	12.12.2019	J. Schweigerer
0.5	version 1 and version 2	13.12.2019	S. Baumann

Bosch Connected Devices and Solutions GmbH
Ludwig-Erhard-Straße 2
72760 Reutlingen
Germany

support@bosch-connectivity.com