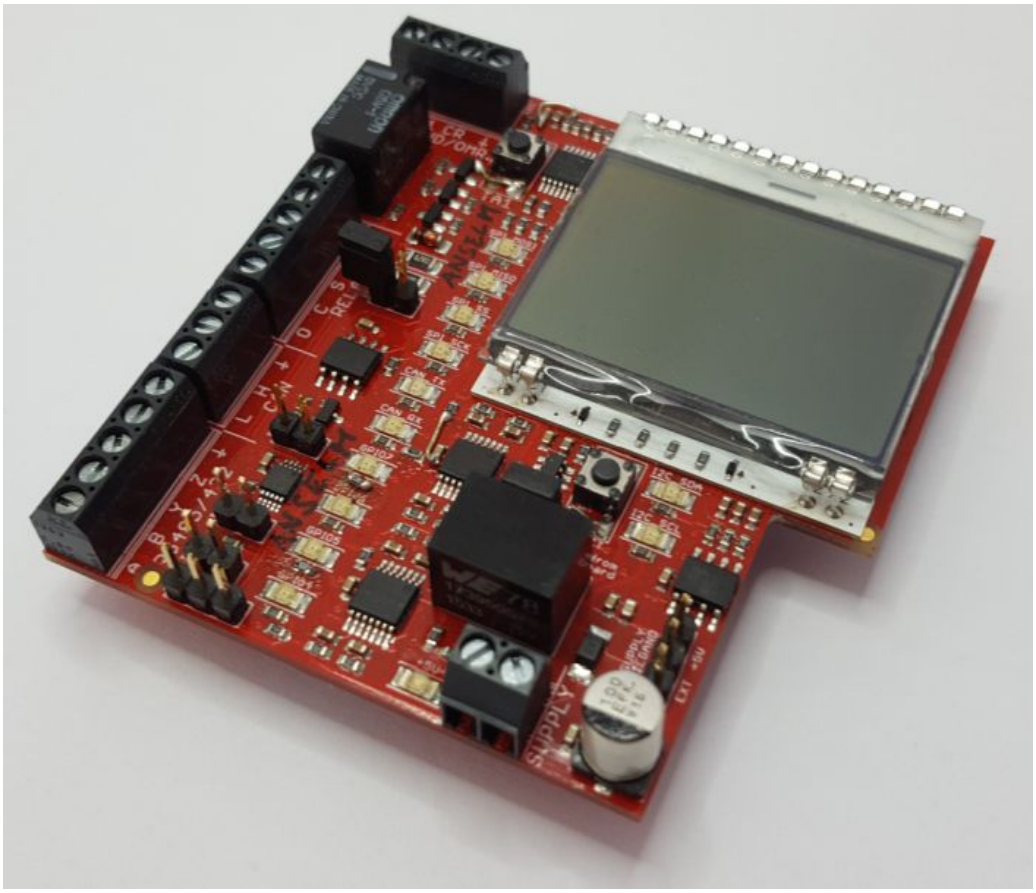
TWN4 MultiTech Nano DevBoard I/O Extender

Technical Handbook

DocRev1, January 31, 2017



Elatec GmbH

Contents

1. Introduction

The TWN4 MultiTech Nano module Development Board provides ready-to-use interfaces (USB and UART) and peripherals (Low-/High-frequency antennas, SAM Card slots) needed to test the core functionalities of the Nano module. The purpose of the I/O Extender is to provide access to the other Module interfaces including CAN, RS-422/485 and Wiegand. An LCD Graphic Display (over SPI interface) is also available for applications requiring greater degree of User-interaction as well as for debug purposes. The I/O Extender also offers a generic Clock/Data port, driven through the GPIO of the TWN4 MultiTech Nano Module. This generic port can be used to send/receive data in OMRON, Wiegand and other custom formats.

The I/O Extender connects to the Expansion Slot of the Development Board, the pinout of which also matches exactly the pinout of the Nano Module itself. The I/O Extender can be supplied from either the Development Board (5V through the Expansion Slot) or via an external variable DC power source. Constant 5V-output Regulator/Converter is present on the I/O Extender for this purpose.

Please note that the Development Board always needs its own power source, supplied through either USB, DF11 or COM port. This is the case even when an external power source is made available to the I/O Extender.

2. Getting Started

2.1. Functional Overview

The functional layout of the I/O Extender is shown on Figure 2.1. Main features are:

- RS-422/485 port with optional on-board 120-Ohm terminations. A Jumper is available to select between Half- and Full-Duplex operating modes.
- CAN port with optional 120-Ohm termination.
- Configurable Clock/Data port driven through GPIO of Nano Module that can be used for Wiegand/OMRON/Custom standards. This port can be powered through an external DC Power Source.
- Generic Relay port for testing.
- LCD Display accessible through SPI. Jumper turns on Optional Backlight.
- I2C-accessible Temperature Sensor.

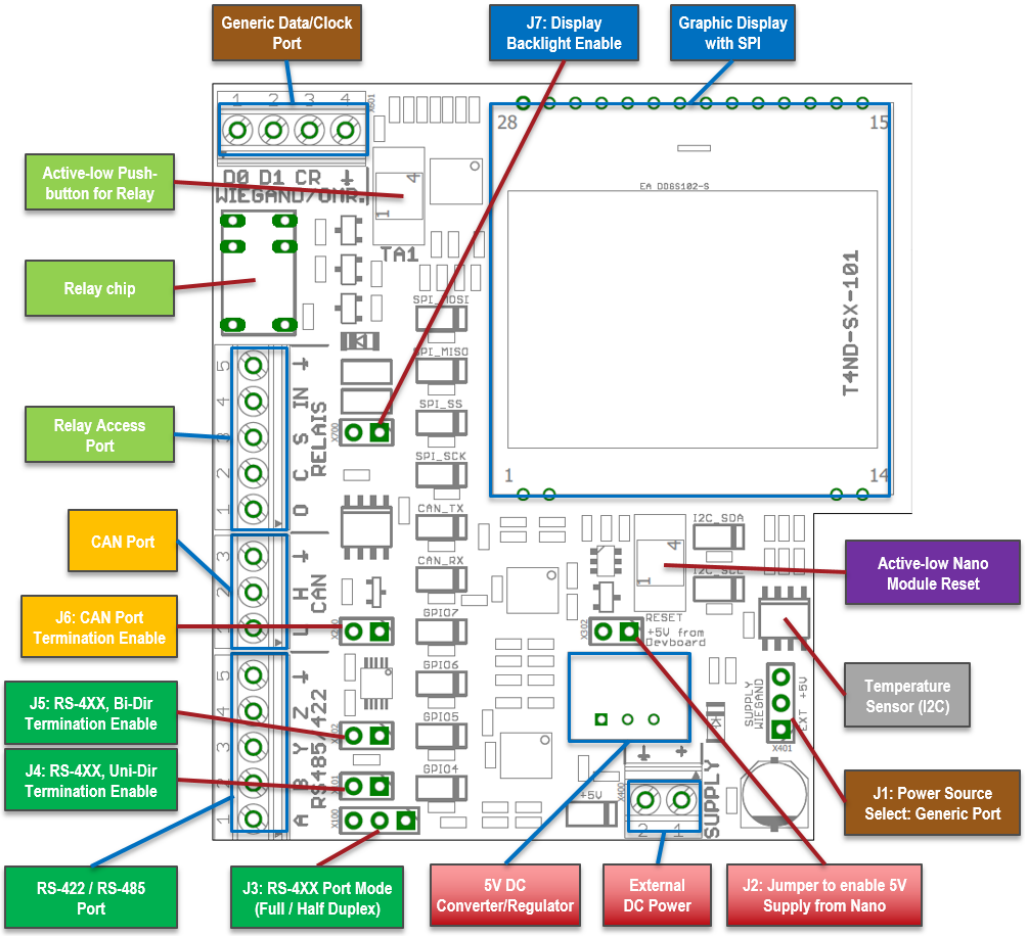


Figure 2.1.: I/O Extender Functional Overview

2.2. I/O Extender Bring-Up

Ensure both the Development Board and the I/O Extender are disconnected from Power. Attach the I/O Extender to the Expansion Slot of the Development Board as shown on Figure 2.2. Ensure correct alignment of the I/O Extender; LCD Display should be partly overshadowing the SAM2 Slot of the Development Board.

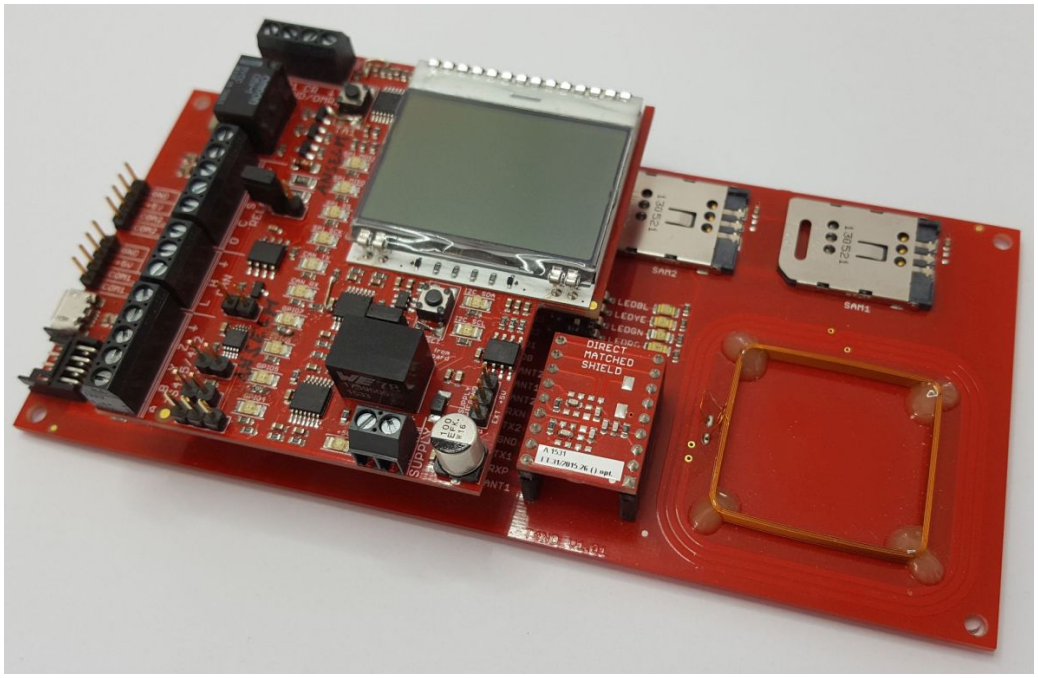


Figure 2.2.: Connecting I/O Extender to Development Board

Establish connection between the PC and the Development Board via either USB or UART protocol. Ensure 5V power is supplied to the Development Board. Please see Development Kit Technical Handbook for instructions. The I/O Extender can be powered from either the Development Board, or via an external DC Source. Depending on Application, please choose one of the 2 options listed in Table 2.1. Refer to Figure 2.1 for assistance locating the components on the board.

Power Option	**Instructions**
DevBoard	Short Jumper J2. Make sure no external supply is connected to the I/O Extender. If using Generic port on I/O Extender, connect a Jumper on J1 (2 pins closest to LCD Display). Otherwise leave it open.
External	Leave Jumper J2 open. Connect external supply to the I/O Extender (DC Range: 7V - 28V). If using Generic port on I/O Extender, connect a Jumper on J1 (2 pins farthest from LCD Display). Otherwise leave it open.

Table 2.1.: Supplying Power to I/O Extender

3. Using Peripherals

A demo "IO Extender Test" App is available in the Development Pack. Please feel free to use it as a starting point in order to expedite the development of Your Application. The demo App demonstrates the use of the peripherals described in this section.

3.1. RS-485/422 Port

Connected to the COM2 port of TWN4 MultiTech Nano Module through a MAX1485CUB Transceiver. The interface facing the Nano Module operates at 3.3V supply. There are 2 single-ended signals to transmit and receive data (COM2_TX and COM2_RX) and an active-high TX_Data_Enable (driven by GPIO 4).

The outward-facing Transceiver Interface operates at 5V. This interface can be used in Half-Duplex or Full-Duplex mode. Use Jumper J3 to switch between the 2 modes. Short the 2 pins closest to the connector to operate in Full-Duplex mode. Short the 2 pins farthest from the connector to operate in Half-Duplex mode.

In Full-Duplex mode, use the Bi-Directional port (Y/Z) to transmit data from the Nano Module, and the Uni-Directional port (A/B) to receive data. In Half-Duplex mode, use the Bi-Directional port (Y/Z) to both transmit and receive data. Please see Table A.1 for more information.

Both the Bi-Directional and the Uni-Directional ports have optional 120-Ohm on-board terminations. Short Jumper J4 to enable the termination for the Uni-Directional port. Short Jumper J5 to enable the termination for the Bi-Directional port.

The maximum Baud Rate supported by the port is 115200. It is up to the User to decide the actual operating Baud Rate, based on desired cable reach and application requirements.

Please see TWN4 API Reference document for instructions on configuring/using the COM2 port.

3.2. CAN Port

Connected to the CAN port of TWN4 MultiTech Nano Module through a SN65HVD230 Transceiver. The interface facing the Nano Module operates at 3.3V supply. There are 2 single-ended signals to transmit and receive data (COM2_TX and COM2_RX).

The SN65HVD230 Transceiver is configured to operate in high-speed mode (Rs pulled to GND) with no limitation on slew rate.

For the Connector pinout, please see Table A.2.

An on-board termination of 120-Ohm is available by shorting Jumper J6.

3.3. Generic Output Clock/Data Port

This port is an output-only port consisting of 3 single-ended signals driven through the Nano Module GPIO. Due to its generality, an OMRON, Wiegand, or a custom interface can be implemented here. A pin can take on a data or a clock functionality, depending on the User's requirements. There is an inverse relationship between the GPIOs and these 3 output pins:

- CIR (OMRON) = ~GPIO 4
- D0 (Wiegand) / Data (OMRON) = ~GPIO 5
- D1 (Wiegand) / Clock (OMRON) = ~GPIO 6

As described in Section 2.2, the I/O Extender can powered through either the Development Board itself, or an external supply. If no external supply is used, then the Generic port will be operated at 5V. Please connect a Jumper on J1 (2 pins closest to LCD Display). The User is also free to provide an external supply source (7V - 28V). In this case, please connect a Jumper on J1 (2 pins farthest from LCD Display).

2 out of the 3 output ports (D0 and D1) can be operated from the variable external supply. The CIR output is operated from a fixed 5V supply.

A Relay port is available for debug and demonstration purposes. The relay switch is controlled on the board by the CIR pin of the Generic port. When CIR is at ground level, the coil within the relay is under voltage, causing pins 2 and 3 of the Relay port to be shorted. When CIR is brought up to 5V, the voltage across the relay coil is marginal, causing pins 1 and 2 of the Relay port to be shorted.

The Relay port also has a signal "IN1" that can be configured as either input or output with respect to GPIO 7. A push-button is also present on the board to drive this net to ground, regardless of the state of the GPIO.

For the pinouts of Generic port and Relay port, please see Tables A.3 and A.4 respectively.

3.4. Temperature Sensor via I2C

The Temperature Sensor device used is SE95D. Its I2C Slave address is 0x48 (address pins A0, A1, A2 are grounded on the board). It is connected directly to the I2C port of the Nano Module. Please see the Nano Module API Guide for further information.

3.5. Graphic LCD Display via SPI

The LCD Display device used is EA-DOGS102-S. It interacts with the Nano Module via the SPI interface. The backlight for the Display can be enabled/disabled using Jumper J7.

An API is available in the Development Pack to facilitate basic drawing operations on the LCD Display. It currently supports only black-and-white mode of operation. The supported commands are described in the following subsections of this chapter. Important: input parameter type **byte** is defined in TWN4 Nano API as **unsigned char**.

Include file: `lcd_spi_ea.c`

3.5.1. SPI Display Init

This method initializes the SPI interface to the LCD Display and specifies its operating parameters.

```
void SPIDisplayInit(void);
```

Parameters:	None.
Return:	None.

3.5.2. SPI Display Clear

This method fills (clears) the entire LCD Display, depending on argument given. This argument can be used as the background color argument for all other functions.

```
void SPIDisplayClear(bool bg_color);
```

Parameters:	
`bool bg_color`	Background Color. Specifies color to fill Display with: 0=full clear, 1=full black.
Return:	None.

3.5.3. SPI Display String 6x8 XY

This method prints a string of ASCII characters. Display of ASCII characters is made possible through a symbol look-up table. Every character is represented by a 6x8 matrix of pixel data; the matrix incorporates 1-pixel separation between characters. Given that resolution of the Display is 102x64 pixels, a 17x8 character array is supported.

```
bool SPIDisplayString6x8XY
        (
        byte x,
        byte y,
        char text[ ],
        byte len,
        bool bg_color
        );
```

Parameters:

`byte x`	Column Position (X) of character, not pixel. (0 <= x <= 17-len)
`byte y`	Row Position (Y) of character, not pixel. (0 <= y <= 7)
`char text[ ]`	Text string to print in ASCII format.
`byte len`	Number of characters from "text" to print, counting from index 0. This is intentionally provided in case the User does not wish to print the entire "text" string. (1 <= len <= 17)
`bool bg_color`	Background Color chosen for Display (0:clear, 1:black). Usually matches the argument given to "SPI Display Clear" command.
Return:	If the operation was successful, the return value is `true`, otherwise it is `false`.

Important: This method does not support newline characters nor row overflows. Care should be taken to ensure the text printed does not go over screen boundaries. The maximum width of Display is 17 characters. The User needs to verify the start position (X) and length of text printed (len).

3.5.4. SPI Display Draw Line

This method prints a straight horizontal or vertical line on the Display. Unlike the "SPI Display String 6x8 XY" method, the start and end coordinates are given in pixel format. A line length can effectively be a single dot on the screen.

```
bool SPIDisplayDrawLine
        (
        byte xs,
        byte ys,
        byte xe,
        byte ye,
        bool bg_color
        );
```

Parameters:

`byte xs`	Start Point X-Coordinate (0 <= x < 102)
`byte ys`	Start Point Y-Coordinate (0 <= x < 64)
`byte xe`	End Point X-Coordinate (0 <= x < 102)
`byte ye`	End Point Y-Coordinate (0 <= x < 64)
`bool bg_color`	Background Color chosen for Display (0:clear, 1:black). Usually matches the argument given to "SPI Display Clear" command.
Return:	If the operation was successful, the return value is `true`, otherwise it is `false`.

3.5.5. SPI Display Fill In Rectangle

This method fills in a rectangle on Display, overwriting everything within it. Can effectively be a single dot on the screen.

```
bool SPIDisplayFillInRectangle
        (
        byte xs,
        byte ys,
        byte xe,
        byte ye,
        bool bg_color
        );
```

Parameters:

`byte xs`	Start Point X-Coordinate (0 <= x < 102)
`byte ys`	Start Point Y-Coordinate (0 <= x < 64)
`byte xe`	End Point X-Coordinate (0 <= x < 102)
`byte ye`	End Point Y-Coordinate (0 <= x < 64)
`bool bg_color`	Filling color (0:clear, 1:black)
Return:	If the operation was successful, the return value is `true`, otherwise it is `false`.

4. Disclaimer

Elatec reserves the right to change any information or data in this document without prior notice. The distribution and the update of this document is not controlled. Elatec declines all responsibility for the use of product with any other specifications but the ones mentioned above. Any additional requirement for a specific custom application has to be validated by the customer himself at his own responsibility. Where application information is given, it is only advisory and does not form part of the specification.

All referenced brands, product names, service names and trademarks mentioned in this document are the property of their respective owners.

A. Appendix: Connector Pinout Reference

Pin	Pin Name	Function
1	A: Uni-Directional Port +	Non-Inverting RX Data Port. Used in Full-Duplex Mode only.
2	B: Uni-Directional Port -	Inverting RX Data Port. Used in Full-Duplex Mode only.
3	Y: Bi-Directional Port +	Non-Inverting TX/RX Data Port. In Full-Duplex Mode, used as TX Data Port only.
4	Z: Bi-Directional Port -	Inverting TX/RX Data Port. In Full-Duplex Mode, used as TX Data Port only.
5	GND	Ground shared with I/O Extender and Development Board.

Table A.1.: RS-485/422 Port (X103) Pin Configuration

Pin	Pin Name	Function
1	Data Lo	CAN Bus Low Port
2	Data Hi	CAN Bus High Port
3	GND	Ground shared with I/O Extender and Development Board.

Table A.2.: CAN Port (X201) Pin Configuration

The Relay port can be used together with Generic Clock/Data port for debug or demonstration purposes. "Card-In-Reader" pin of the Generic port switches the Relay on/off.

GPIO 4 = HIGH -> CIR = LOW -> Relay Port Short pins 2-3
GPIO 4 = LOW -> CIR = HIGH -> Relay Port Short pins 1-2

Pin	Pin Name	Function
1	D0 / Data	Wiegand D0 Port. OMRON Data. Driven by GPIO 5 of Nano Module. Inverse Relationship: D0 = ~GPIO 5
2	D1 / Clock	Wiegand D1 Port. OMRON Clock. Driven by GPIO 6 of Nano Module. Inverse Relationship: D1 = ~GPIO 6
3	CIR	Card-In-Reader. Driven by GPIO 4 of Nano Module. Inverse Relationship: CIR = ~GPIO 4
4	GND	Ground shared with I/O Extender and Development Board.

Table A.3.: Generic Output Clock/Data Port (X601) Pin Configuration

Pin	Pin Name	Function
1	O (Open)	Open when CIR=0V (LOW), shorted to Pin (2) when CIR=5V (HIGH).
2	C (Common)	Common pin within Relay, to be connected to either pin 1 or 3 depending on Generic Port pin CIR.
3	S (Short)	Shorted to Pin (2) when CIR=0V (LOW), open when CIR=5V (HIGH).
4	IN1	Connected to GPIO 7 of Nano Module with no inversion. Can be driven Low by on-board push-button.
5	GND	Ground shared with I/O Extender and Development Board.

Table A.4.: Relay Port (X600) Pin Configuration