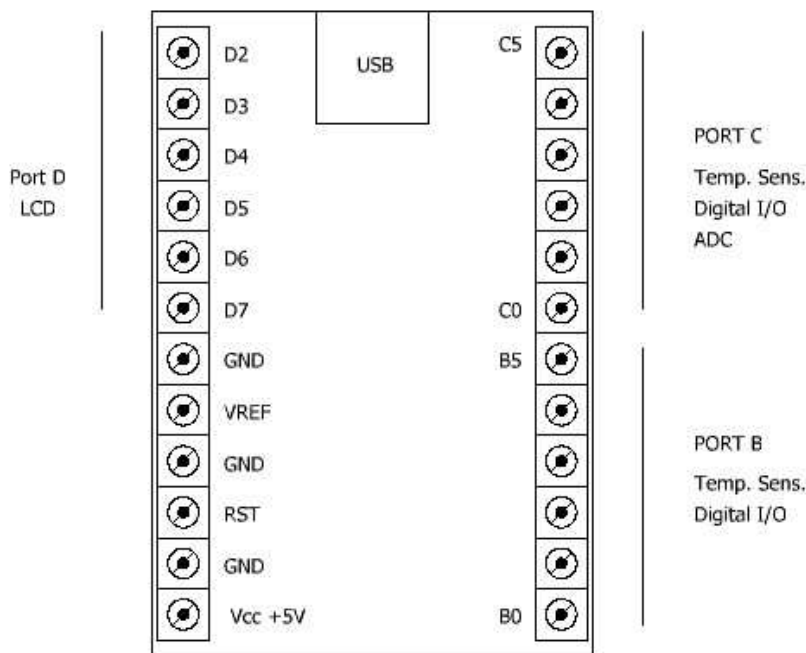


Firmware /TEMP12

The /TEMP12 firmware allows you to connect up to twelve digital temperature sensors (type Dallas DS18B20). Data from twelve channels is transferred to your PC via USB. ONE temperature sensor can be connected to each of the screw terminals B0...B5 and C0...C5.

Channels that are not occupied by temperature sensors, can take over alternative functions:

- DS18B20 temperature sensor
- Digital input
- Digital output
- Alarm output
- Sensor input (ADC) (channels C0...C5 only)



This document shows wiring diagrams using the USB-μPIO module. The can be found as well on the netPIO network module, with same functionality and arrangement.

The device is delivered with a configuration software, which allows you to setup each channel. The configuration is transferred to the device and stored temporarily or permanently. So the device can work as PC interface as well as in a stand-alone application.

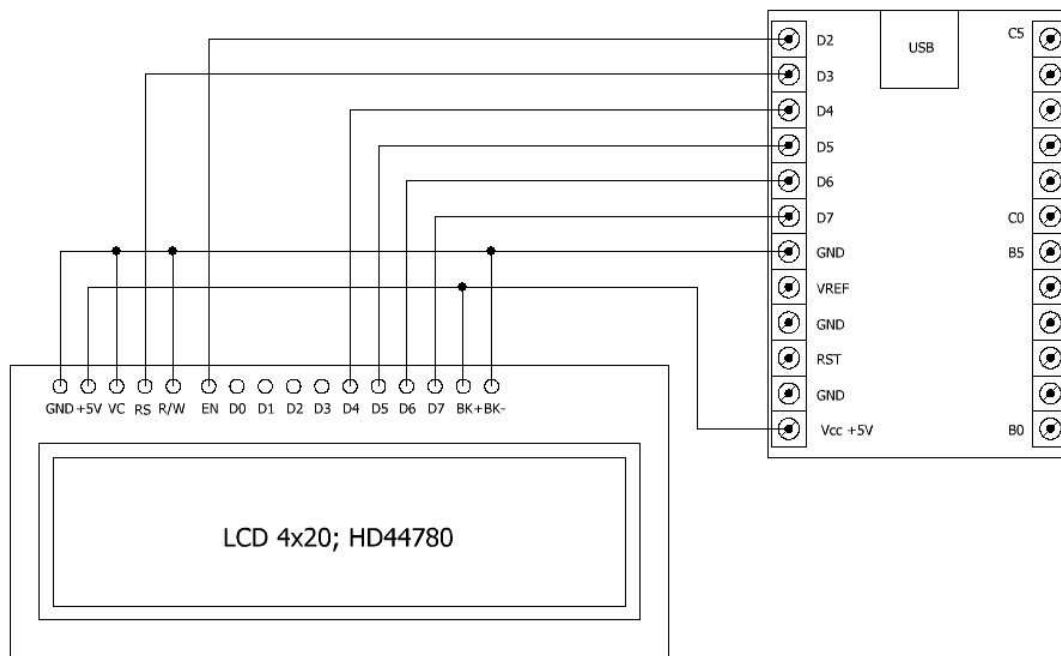
A “HIGH” sense limit and a “LOW” sense limit can be configured for each analogue channel (temperature or sensor). If a channel value gets classified as HIGH or LOW, this can trigger an depending alarm output. For example a cooler could be activated if a temperature channel indicates “HIGH”. Channels that exceed its limits are also indicated on an optional LCD display.

LCD option

A standard LCD text display (HD44780) with 4x20 characters can be connected on Port D2..D7. Other display formats are NOT supported. The display will show value and status of call channels automatically. Connections between the board and the LCD display must be made as follows:

BOARD	LCD 4x20 characters
PORT D	HD44780
D2	EN
D3	RS
D4	D4
D5	D5
D6	D6
D7	D7

LCD input R/W must be connected to ground (GND). The remaining data lines **of the LCD** D0...D3 are not connected.



This document shows wiring diagrams using the USB- μ PIO module. The can be found as well on the netPIO network module, with same functionality and arrangement.

Channel configuration – DS18B20 temperature sensor

The picture below shows how to connect temperature sensors to the board.

Temperature sensors can be operated on the channel B0...B5 and C0...C5. Use the configuration software to setup the mode for each channel. The DS18B20 can measure temperatures from -55 °C und 125° C. In case you need to change the unit (for example from °C to °F) you can enter a linear sensor scale factor and offset.

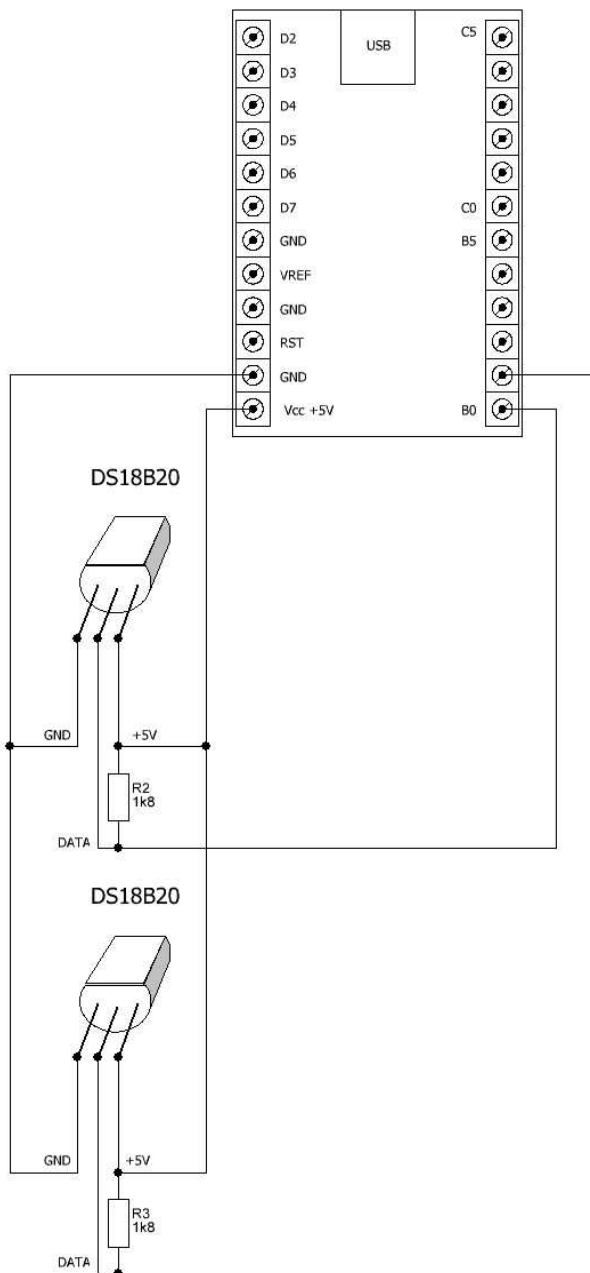
Example – Scale °C to °F:

Unit = „°F“

Factor = 1,8

Offset = 32

Understand that scaling a sensor does NOT change its physical range (-55 °C to 125° C), but only changes its representation. Limits for „HIGH“ and „LOW“ detection must be entered in SCALED units.



This document shows wiring diagrams using the USB-μPIO module. The can be found as well on the netPIO network module, with same functionality and arrangement.

Channel configuration – ADC sensor input

Channels C0...C5 can be configured as ADC sensor inputs. In this channel mode an analogue-to-digital-converter measures an input voltage, which may be from a light sensor for example..

The ADC uses a reference voltage VREF to measure the input voltage in volt units. There are three possible sources for the reference voltage. First is the supply voltage (5V USB), second is a internal voltage source (1.1V) and third is an external reference voltage supplied at REF input. The reference voltage source can be selected for each ADC channel input separately. The reference voltage VREF must never exceed the supply voltage! The input voltage must be between 0V and VREF, which defines the input range. Voltages at all pins must never go below 0 Volt (GND) or above supply voltage (VCC)!

With a correct VREF setting the ADC will report the input status in volt units. ADC channels can be scaled to different units with linear scale factor and offset.

Example – Scale voltage (with VREF = 5V) into percentage

Unit = „%“
Factor = 20
Offset = 0

The input channel is now represented as percentage, as $5(V) * 20 = 100\%$. Understand that scaling a sensor does NOT change its physical range (0V ... VREF), but only changes its representation. Different ranges of sensors must be adapted with suitable input circuits (voltage divider, amplifier or similar)

Limits for „HIGH“ and „LOW“ detection must be entered in SCALED units.

Channel configuration – Alarm output

In this channel mode a channel works as digital output, that outputs high level (5V) if the channel is ON or low level (0V) if the channel is OFF. The output can drive up to 20 mA loads. The alarm output will turn ON and OFF automatically, in case a certain CONDITION is true or false.

Configurable conditions are the “HIGH” and “LOW” states of other analogue channels (temperature or ADC inputs). To avoid flickering of alarm outputs in case the inputs value swings slightly around a limit, a fixed hysteresis of +/-2 LSB was built in.

Channel configuration – Digital output

In this channel mode a channel works as digital output, that outputs high level (5V) if the channel is ON or low level (0V) if the channel is OFF. The output can drive up to 20 mA loads. The output is controlled by the PC.

Channel configuration – Digital input

This channel mode sets the channel direction to input. A high level (5V) is interpreted as “ON”. A low level (0V) represents “OFF”. Open (not connected) inputs are internally pulled high. This makes it possible to sample the status of a switch, which connects the input to ground (GND) if contacts a closed.

Using the LCD option, the LCD will show the status of digital channels (digital inputs, digital outputs, alarm outputs) as “ON” or “OFF”

Measuring differences between sensor values

In practice you may need to measure differences between sensors, like indoor temperature – outdoor temperature. For that purpose the /TEMP12 firmware offers four additional (virtual) channels. The values for these channels are calculated from the difference of two sensors as follows:

- DIFF1 = Channel C0 – Channel C1
- DIFF2 = Channel C0 – Channel C2

- DIFF3 = Channel C0 – Channel C3
- DIFF4 = Channel C4 – Channel C5

Both channels of a pair must be configured identically, to enable the differential channels. (Otherwise the difference would not make sense.) The only configurable parameters for the differential channels are the “HIGH” and “LOW” limits, which can be used as conditions for alarm outputs as any other sensor channel.

Device setup

All configured parameters must be transferred to the device and are stored on the device for normal operation. The configuration software must be used for this configuration upload. The configuration can be stored temporary in the RAM memory or permanently in the EEPROM memory of the device. A device reset will read the configuration stored in the EEPROM memory. A configuration stored temporary in the RAM memory will be overwritten in this case. A reset occurs when the device is powered up or is triggered by a low pulse at the reset input (RST). If jumper J1 (SOFT-RESET) is set, a reset also occurs when PC software establishes a data connection, opening the COM port. Avoid this behaviour removing Jumper J1 if necessary.

During upload process a internal clock is synchronized with the PC time and date setting, Allowing to display time and date information on the optional LCD display. Time information gets lost for technical reasons, when a device reset occurs. Time and date will not be displayed in that case. A permanent power supply is needed to display time information.

Protocol description

Baud 57600
 8 data bits
 No parity
 1 stop bit

Request / Response communication is used. A request consists of 21 data bytes. After receiving 21 bytes, the device responses with a response data block in any case. A new request must not be send, before the response on the previous request was received completely.

Request: 21 bytes

The request data block is 21 bytes long and transfers all control data.

Byte 1: Chr(66) = „B“ in any case
 Byte 2...Byte 13: Digital output states for channels B0...B5, C0...C5
 Byte 14...21: Reserved

A request must start with a character „B“. The following characters (Byte 2 ... 13) control the output state of channels that have been configured as digital outputs. Channels that are not configured as digital output will not be affected. A character “0” = chr(48) turns the corresponding channel OFF. A character “1” = chr(49) turn the corresponding channel ON.

Any other character will be ignored.

Programming example (with PRINT command on serial port):

```
Print „B11111111111111111111-----“;
```

Above command will turn all (configured) outputs ON. Note the semicolon at the end of each line. A chr(13) chr(10) must not be sent after the 21 bytes of the request!

```

Print „B000000000000-----“; // Turn off all digital outputs (low)
Print „B100000000000-----“; // Set B0 HIGH; all others LOW
Print „Bxxxxx11xxxxx-----“; // Set B5 and C0 HIGH; do not change others
Print „Bxxxxx01xxxxx-----“; // Set B5 LOW; Set C0 HIGH; do not change others
Print „Bxxxxx01xxxxx-----“; // Set B5 LOW; Set C0 HIGH; do not change others
Print „Bxxxxxxxxxxxxx-----“; // No change of outputs (request only)

```

The device will send a response after each of above PRINT commands.

Response

After receiving 21 request bytes the device will send a response to the PC. The response is a readable character chain (String). The response string consists of several so-called “cells”. Each cell is six characters long, followed by a semicolon as separation character. The first cell contains the channel name of the first channel (B0). The second cell contains the channel value for channel B0. Two cells are sent for each channel. A pair of two cells (Name and value) is sent for each channel. Channels B0...B5, C0 ... C5 and DIFF1 ...4 are sent one after each other.

Example response:

```

„ ALARM; EIN ;B1-OUT; EIN ;B2-OUT; EIN ;B3-OUT; EIN ;B4-OUT; EIN ; PC-B3; EIN
; INNEN; 20.9; TANK ;96.77%;C2-ADC; 1.07-;C3-ADC;4.24;C4-ADC; 4.29 ;C5-ADC; 4.38 ;
-- ; -- ; -- ; -- ; -- ; -- ; -- ; -- ;” +CHR(13)CHR(10)

```

Finally the response is terminated with chr(13)chr(10). Altogether 32 cells (á 6 characters + Semicolon) and two terminating character are sent. This makes $32 * 7 + 2 = 226$ characters.

The cell content is used to display information on the optional LCD display as well. Therefore cells contain some special characters. To get a readable representation on the PC, the character must be replaced as follows:

```

Chr(1) =  μ
Chr(2) =  °
Chr(3) =  °C
Chr(4) =  °F
Chr(5) =  °K
Chr(6) =  €

```