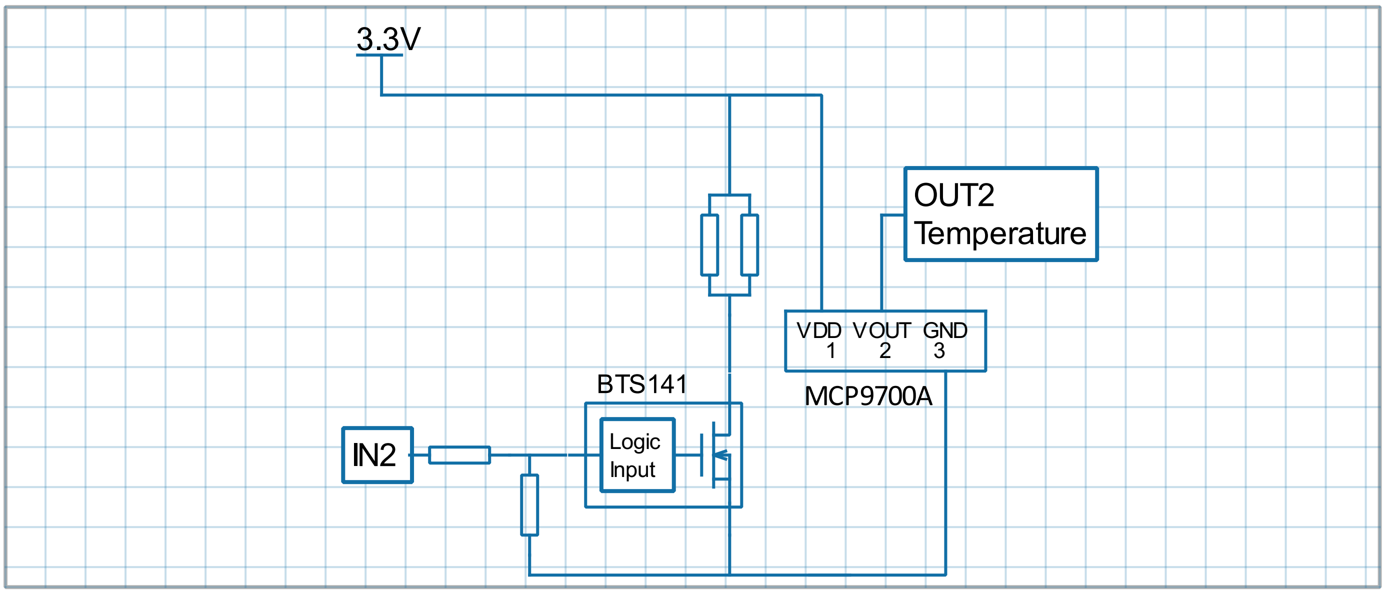
Temperature Control using LabJack and Virtument- GUIs for LabJack Configuration

This document illustrates how the LabJack T7 device can be used to control the ambient temperature by switching a electronic circuit on and off. When the circuit is switched on, the current trough the resistors cause a temperature change. Temperature measurement is performed by using a MCP9701A temperature sensor. The LabJack T7 unit is used to switch the circuit on and off using digital output (DIO) and to measure the temperature using the analog input, AIN. Virtument, a complementary software part of MatDeck, is used to set up the electronic circuit and to read the temperature using a virtual thermometer.

Schematics of the electronic circuits

The schematics of the described system are displayed below. It should be pointed out that the schematics are created in MatDeck, which is suitable for various professional drawings.



The description of the circuit is as follows:
Functionality

- IN2 and OUT2 demonstrate the PID temperature control

Demo board schematic pin descriptions

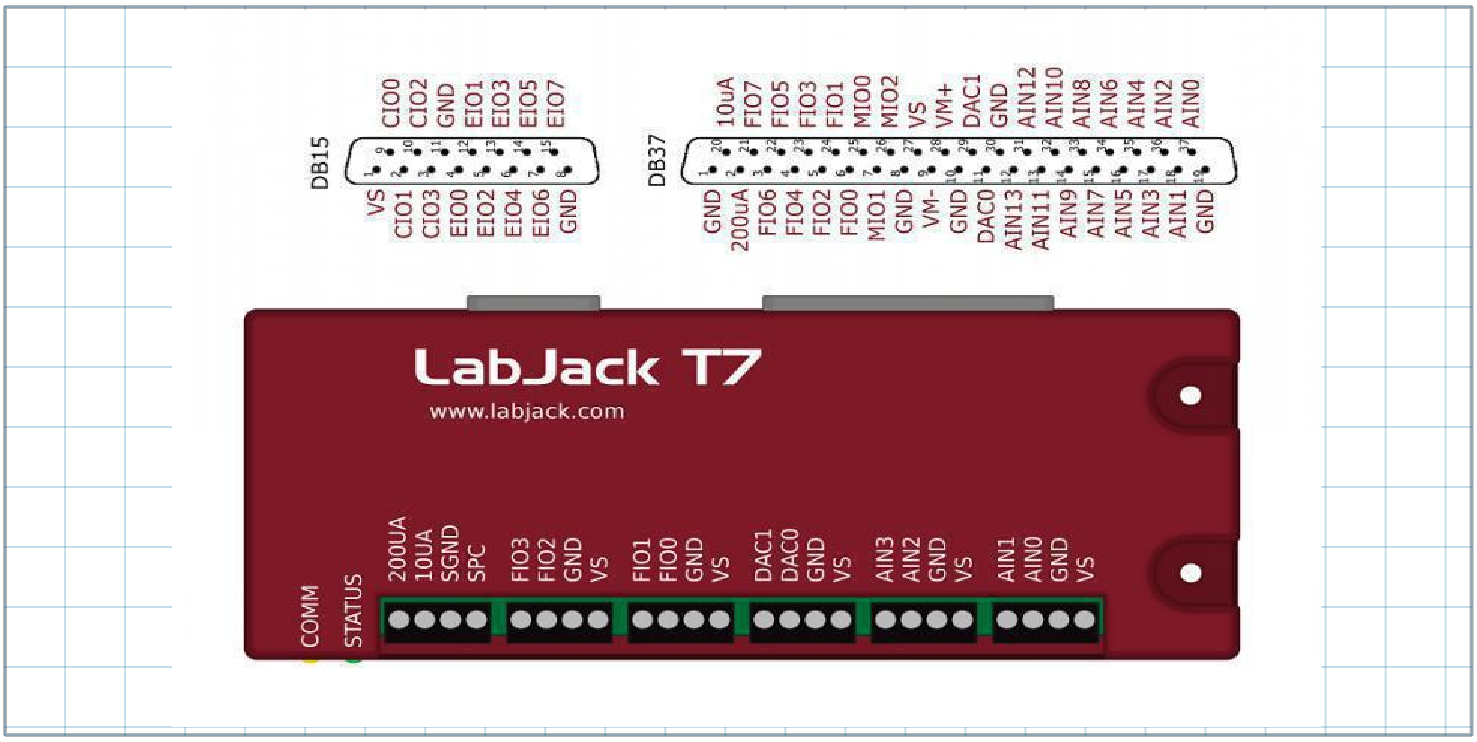
- IN2 input is connected to the PWM driver output in order to heat up the resistor
- OUT2 - output from the temperature sensor in mV

Connection to ADC unit

- IN2 is connected to the PWM output.
- OUT2 is connected to the Analog inputs

Parts:

- For the power driver, BTS 141 is used which has a logic level low side driver
- The temperature sensor used is a MCP9700A-E/TO



LabJack T7 uses in MatDeck

In this experiment, a LabJack T7 unit is used to produce the IN2 signal, as a digital PWM output out from the DIO0 channel. At the same time, the T7 is used to measure the temperature by recording the OUT2 signal at the AIN2 channel. MatDeck supports LabJack functions which can be used directly inside MatDeck's script to configure LabJack devices, and to generate and acquire signals from the electronic circuits as described above. MatDeck also provides Graphical User Interface (GUI) plug-ins for simple, effective configuration. There are three plug-ins for three different groups of pins: ljdio_config_form() is used to configure DIOs, ljaio_config_form() is used to configure AINs and ljdac_config_form() is used to configure DACs. Here, details about the configuration of the selected features in this experiment are explained.

GUI Configuration of DIO EF PWM out

Here, DIO is used to produce the IN1 signal. When DIO0 is high, the transistor is switched on and the current flowing through the resistors heats the temperature sensor. If DIO0 is low, the transistor is switched off and there is no current, thus the temperature falls.

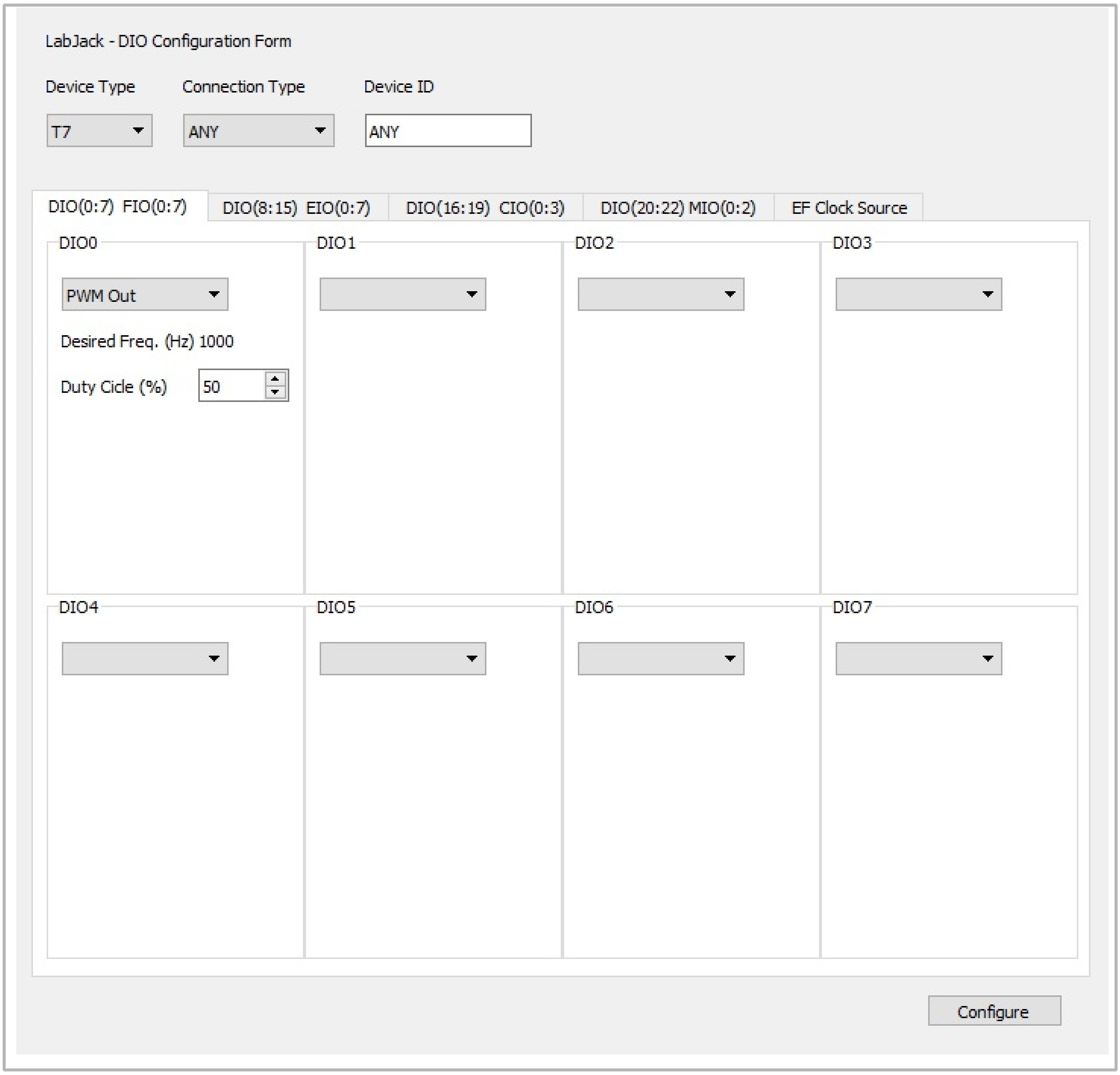
The GUI form used for configuring DIO can be started by using ljdio_config_form(), as follows. The form is embedded within the canvas and used for the configuration of DIOs.

```
f := ljdioT7_config_form(0 , "DIO form")
ljdioT4_config_form_configure(f)
```

The PWM output at FIO0(DIO0) requires the clock source, thus the clock is first configured. There are three parameters to select for the configuration: the clock source, the clock divisor and the roll value for the given clock. There are three different clocks supported by the T7, the most common is clock0 whose frequency is 80MHz. The clock divisor can be any power of two from 1, 2, up to 256. In this example, we use a divisor value of one. The roll value is determined according to the desired frequency of the PWM Out signal. For example, if the desired frequency is 1kHz, the roll value is 80Hz/Divisor/1kHz=80000. In the GUI, it is possible to choose and set the desired frequency or desired roll value. The PWM output at FIO0 (DIO0) is configured by selecting the appropriate option from the drop down menu. In the GUI, it is also possible to set the desired value of the duty cycle directly to 50%.

```
a := ljdevice_last_error("s") //Check for errors in configuration
```

```
a = "LJ_SUCCESS"
```


LabJack - DIO Configuration Form

Device Type: T7
Connection Type: ANY
Device ID: ANY

DIO0: PWM Out
Desired Freq. (Hz) 1000
Duty Cicle (%) 50

Configuring Temperature Measurement via GUIs

The temperature is measured by using the AIN2 channel of the T7, where the OUT2 signal is connected. The low power linear active thermistor circuit ,MCP9701A, is used as the temperature sensor. Here, OUT2 is the voltage that depends on the ambient temperature, which should be converted into temperature using the linear function given in the data sheet. The sensor transfer function is:

$$V_{OUT} = T_C \cdot T_A + V_{0°C}$$

Here, V_{OUT} is the sensor output voltage, T_A is the ambient temperature, T_C is the temperature coefficient, and $V_{0°C}$ is the sensor output voltage at 0°C. From the MCP9701A datasheet, T_C=19.5 mV/°C and $V_{0°C}$=400mV. In order to determine the temperature from the voltage, we need to inverse the function.

$$T_A = V_{OUT}/T_C - V_{0°C}/T_C$$

Slope and offset can be determined as follows:

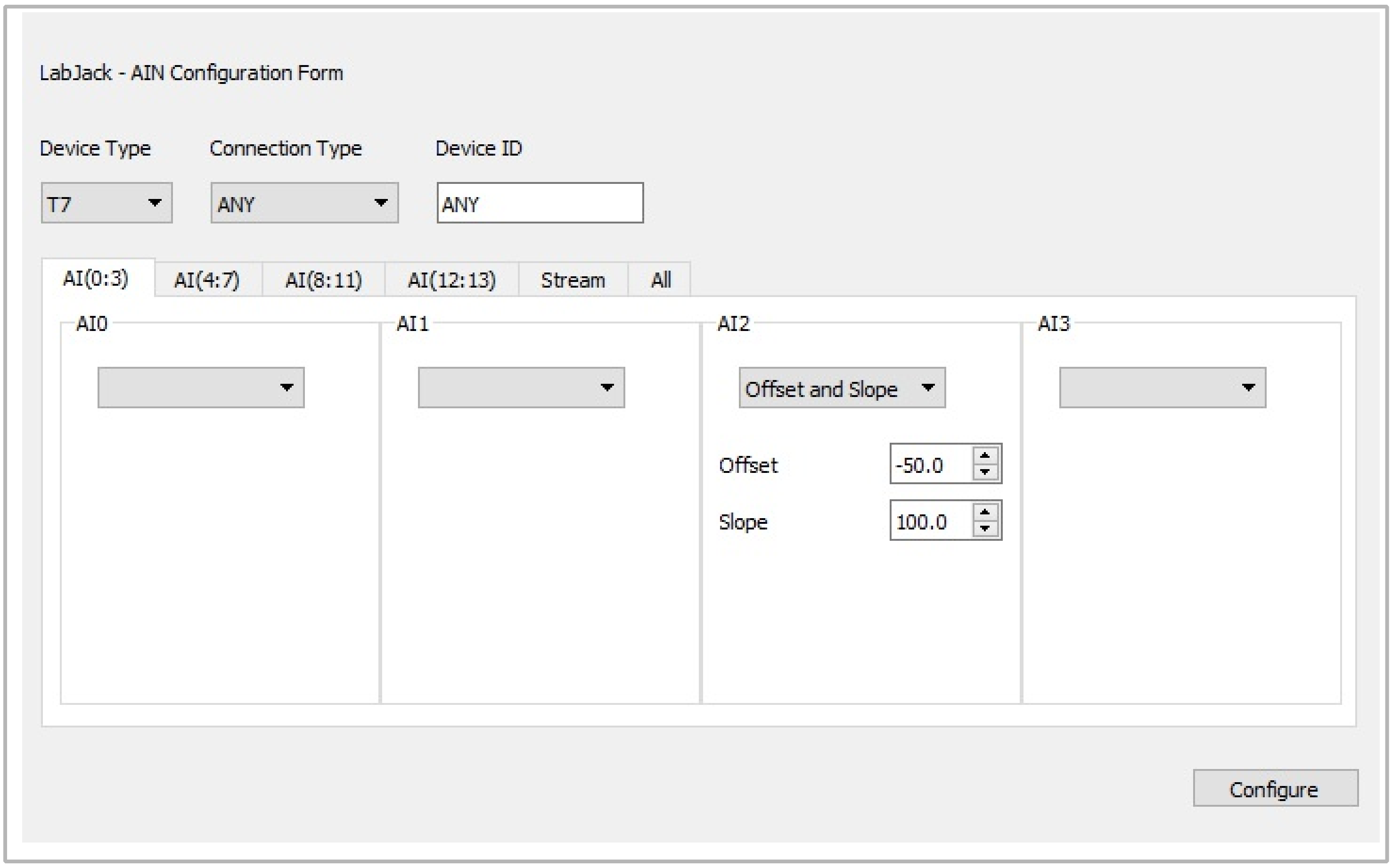
```
Tc := 0.0100
V0 := 0.5
Slope := 1 / Tc
Offset := -V0 / Tc
```

Slope = 100 Offset = -50

AIN2 is configured to use the Offset and Slope extended feature, EF_INDEX is 1, which automatically adds a slope and an offset to analog readings according to the linear function above.

MatDeck provides ljaio_config_form() that can be used to set all the parameters graphically, which is very convenient for the user. In the following segment, there is an illustration on how to use the form for AIN configuration. At the beginning, the form is evoked by calling ljaio_config_form(). The form is embedded within the canvas and used for the configuration of AINs.

```
f2 := ljainT7_config_form(0 , "AIN form1")
ljainT7_config_form_configure(f2)
```


LabJack - AIN Configuration Form

Device Type: T7
Connection Type: ANY
Device ID: ANY

AI2: Offset and Slope
Offset: -50.0
Slope: 100.0

Use of Configured LabJack T7

In order to use the configuration and use the device, the LabJack T7 device should be opened in the document:

```
6  dev := ljdevice_open("any", "any", "any")
```

The temperature measurement is automatically read by using:

```
7  Ta :=ljdevice_read(dev, "AIN2_EF_READ_A")
```

$$Ta = 29.325 \quad C$$

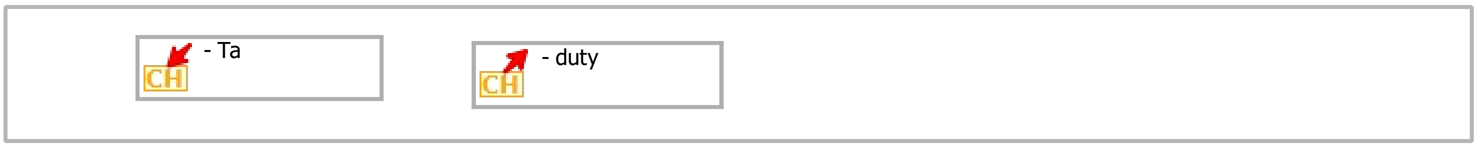
Communications with LabJack T7 from Virtument

The MatDeck software has an instruments panel otherwise known as a visual PC instrumentation called Virtument. The Virtument instrument panel can receive and transmit data to both the MatDeck software and documents. Both types of software can connect to IP address, and receive and transmit data.

Here, a vertical slider in Virtument is used to control the duty cycle of the PWM out signal. In this document, the channel is set and connected with the variable duty for the input from Virtument. The temperature variable Ta is connected to the channel in order to display its value in Virtument, using a virtual analog, digital thermometer, and a graph. The Virtument instrument panel ,labjackT.vir, contains the virtual instruments mentioned above.

```
8
```

```
duty := 50
roll_value := 80000
config_a := roll_value * duty / 100
```



In the following loop, the temperature Ta is controlled using the variable duty, and both variables are connected to the virtual instruments as explained above.

```
while(true)
{
  config_a = roll_value * (duty) / 100
  ljdevice_write(dev,"DIO0_EF_CONFIG_A", config_a)
  Ta = ljdevice_read(dev, "AIN2_EF_READ_A") //uncomment this
  sleep(1)
}
```

Finally, at the end, all extended features should be disabled and the device should be closed.

```
ljdevice_close(dev)
```