Configuring LabJack T Series Devices in MatDeck for Multiple Roles

In this document, we will illustrate how a single LabJack T7 or T4 device is configured and used for multiple purposes. The same unit can have multiple settings. This means that the function of the unit pins can be changed while your using it. MatDeck provides LabJack functions, and configuration is possible directly in MatDeck's Script - C++ style. However, the most effective and intuitive way for the configuring LabJack devices is by using the MatDeck GUI.

Configuration of Analog Inputs for Temperature Measurement using MatDeck GUIs

In this example, the T7 is configured to measure the temperature by collecting the signal at the AIN2 channel. In this example, we presume that the low power linear active thermistor circuit MCP9701A is used as a temperature sensor. Here, output is voltage that depends on the ambient temperature, which should be converted into the temperature by 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 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 the inverse function.

$$T_A = V_{OUT}/T_C - V_{0°C}/T_C$$

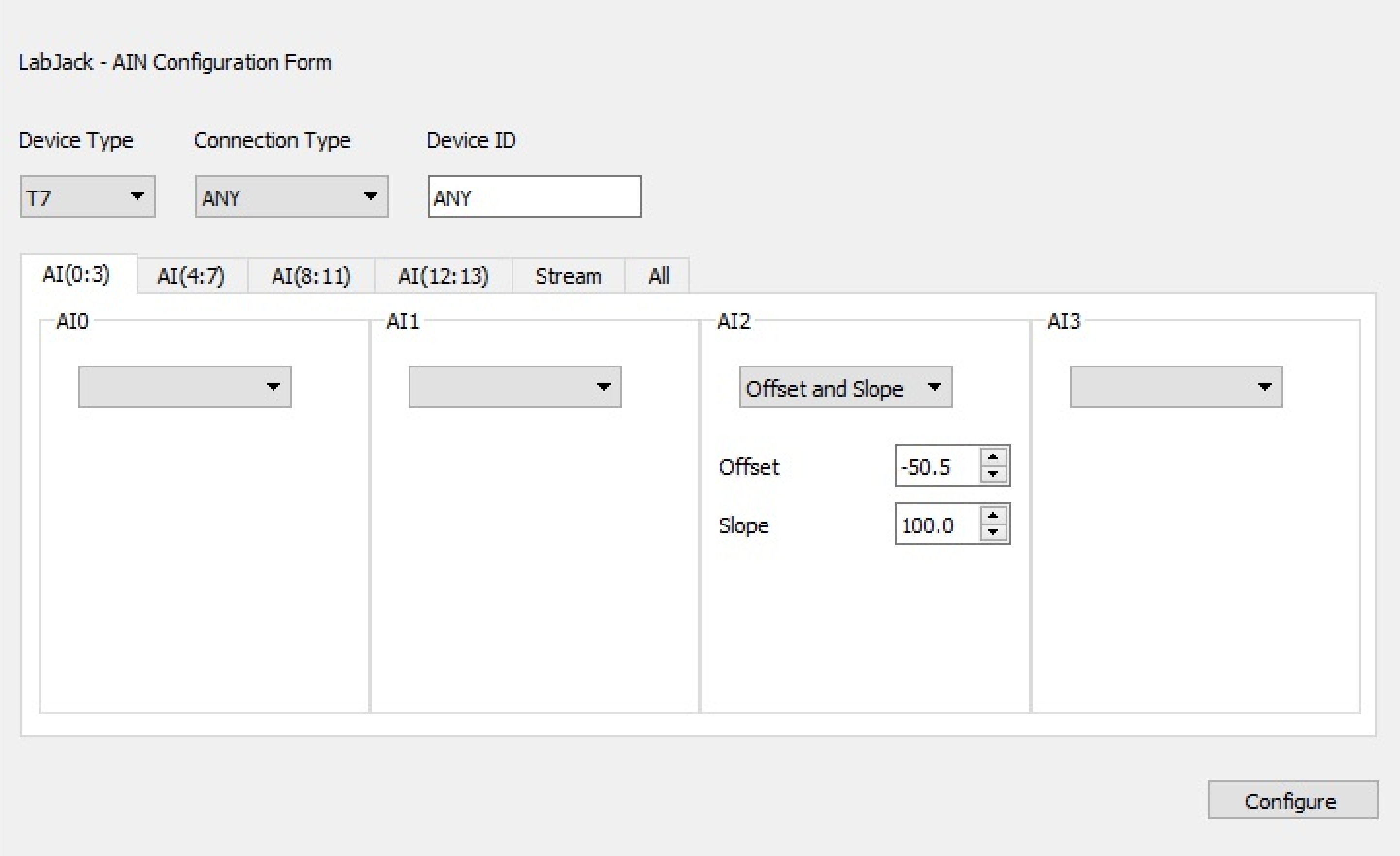
Slope and offset can be determined as follows:

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

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. The tab AI(0:3) is used to select AIN2, where the extended feature Offset and Slope which is appropriate for this type of temperature measurement is selected. Slope and Offset are entered in the appropriate field.

```
form := ljainT7_config_form(0 , "Form")

ljainT7_config_form_configure(form)
```



LabJack - AIN Configuration Form

Device Type: T7

Connection Type: ANY

Device ID: ANY

AI2: Offset and Slope

Offset: -50.5

Slope: 100.0

Before use, the device should be opened first, and afterwards should be closed again. The temperature is automatically read using:

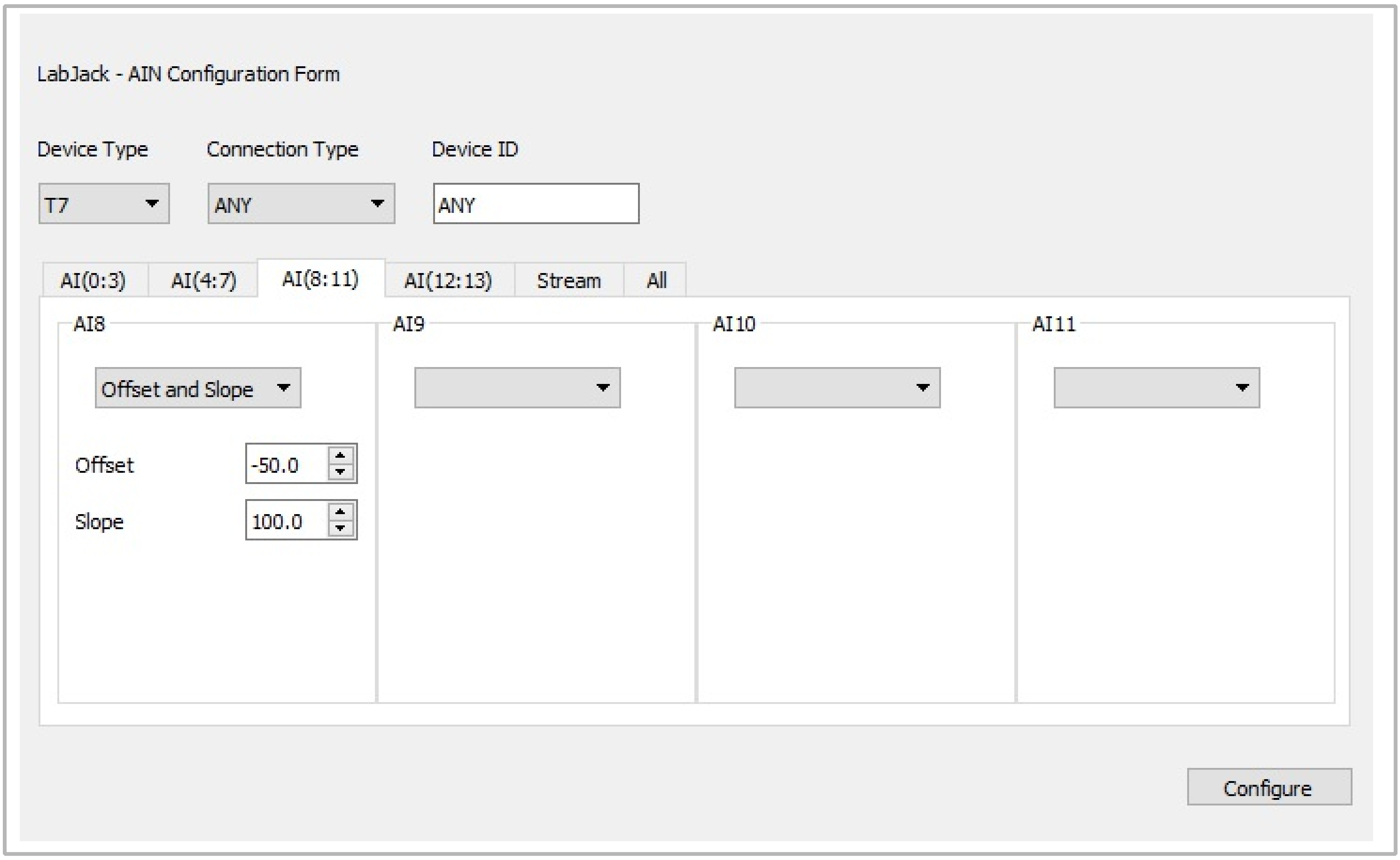
```
dev1 := ljdevice_open("any", "any", "any")
Ta :=ljdevice_read(dev1, "AIN2_EF_READ_A")
ljdevice_close(dev1)
```

```
Ta = 21.773    C
```

The same thermistor circuit ,MCP9701A, is used as a temperature sensor at AIN8 , as well. Thus, we repeat the configuration procedure.

```
form2:= ljainT7_config_form(0 , "Form_AIN8")

ljainT7_config_form_configure(form2)
```



LabJack - AIN Configuration Form

Device Type: T7

Connection Type: ANY

Device ID: ANY

AI8: Offset and Slope

Offset: -50.0

Slope: 100.0

The temperature at AIN8 is read using:

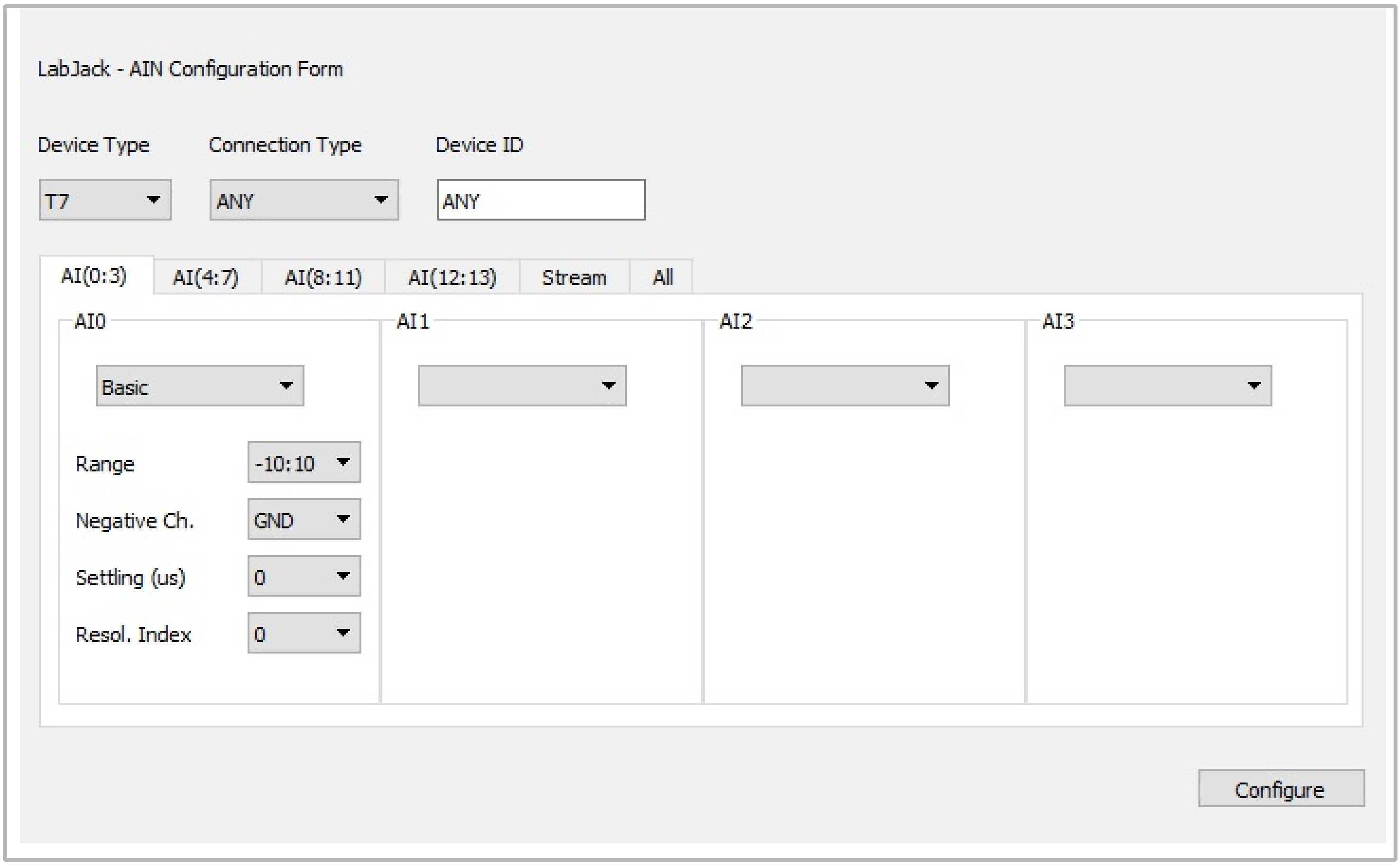
```
dev2 := ljdevice_open("any", "any", "any")
Ta8 :=ljdevice_read(dev2, "AIN8_EF_READ_A")
ljdevice_close(dev2)
```

Reading Voltage at Analog Inputs

At AIN0, the voltage should be read. Thus, AIN0 is set for basic voltage reading.

```
form3:= ljainT7_config_form(0 , "Form_AIN0")

ljainT7_config_form_configure(form3)
```



LabJack - AIN Configuration Form

Device Type: T7

Connection Type: ANY

Device ID: ANY

AI0: Basic

Range: -10:10

Negative Ch.: GND

Settling (us): 0

Resol. Index: 0

The voltage at AIN0 is read:

```
dev3 := ljdevice_open("any", "any", "any")
V0 :=ljdevice_read(dev3, "AIN0")
ljdevice_write(dev3, "IO_CONFIG_SET_CURRENT_TO_DEFAULT", 1)
ljdevice_close(dev3)
```

Results

The results can be displayed in a table simultaneously. Because of the simplicity, the table is generated manually. This means all the text: 1. 2. 3.,Temperature at AIN2, AIN8, and Voltage at AIN0 is written directly into the table cells. Variable names: Ta, Ta8, and Vo are also written into cells, but their value is automatically generated in the cell. There is another example where the table is generated by table commands instead of manually.

1.	Temperature at AIN2	$Ta = 21.773$ C
2.	Temperature at AIN8	$Ta8 = -12.078$ C
3.	Voltage at AIN0	$Vo = 3.311$ V

Exporting Values to Excel

Here, the data obtained by three measurements will be exported to Excel file at appropriate positions. The variables are exported manually.

```
excel_write("measurements.xlsx", "Sheet1", "A1", "Temperature at AIN2")

excel_write("measurements.xlsx", "Sheet1", "B1", Ta)
excel_write("measurements.xlsx", "Sheet1", "A2", "Temperature at AIN8")
excel_write("measurements.xlsx", "Sheet1", "B2", Ta8)
excel_write("measurements.xlsx", "Sheet1", "A3", "Voltage at AIN0")
excel_write("measurements.xlsx", "Sheet1", "B3", V0)
dat := date("d/m/y")
excel_write("measurements.xlsx", "Sheet1", "A4", "Date")
excel_write("measurements.xlsx", "Sheet1", "A5", dat)
tim := time(":")
excel_write("measurements.xlsx", "Sheet1", "B5", "Time")
excel_write("measurements.xlsx", "Sheet1", "B5", tim)
```

Exporting to Text File

Measurements data can be exported to txt file, as well. For example, the txt file will be made if Temperature at AIN2 is above 20°C. The sentence "The measurement is successful" will be written to the test.txt file.

```
if(Ta >= 20)
{
  file_name := date("d-m-y") + " " + time("-")
  F := file_create(file_name, "text", true)
  file_write(F, "The measurement is successful\n")
  file_write(F, Ta)
  file_write(F, "°C\nDate and Time")
  file_write(F, "\n" + dat + " " + tim)
  file_close(F)
}
```