Configuring LabJack T Series Devices in MatDeck for Multiple Roles - Automatic Reading of Results

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

Configuring 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 are using the linear temperature sensor, MCP9701A. Here, the output is the 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:

$$V_{OUT} = T_C \cdot T_A + V_{0^\circ C}$$

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

$$T_A = V_{OUT}/T_C - V_{0^\circ C}/T_C$$

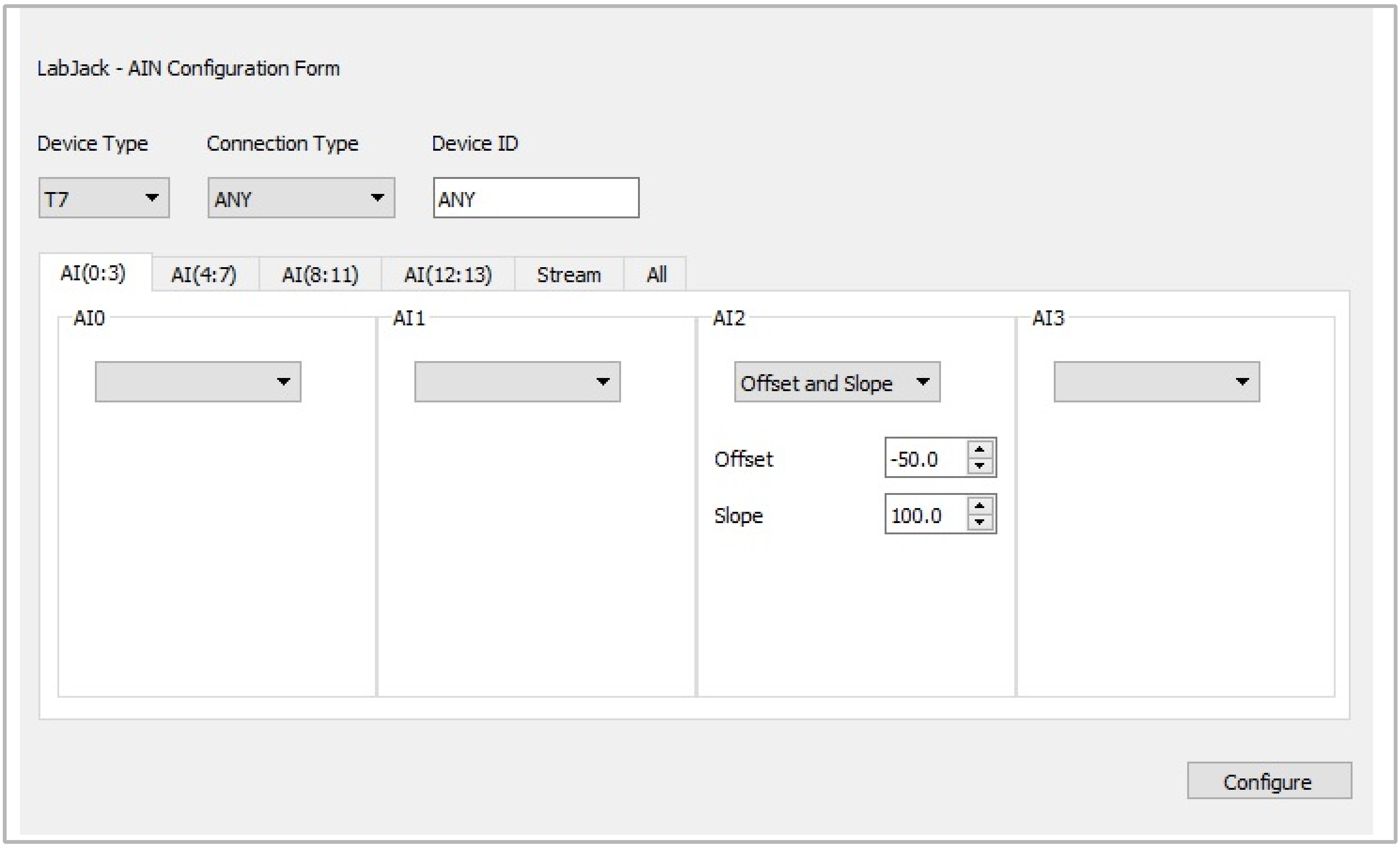
Slope and offset can be determined as follows:

```
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 501")

ljainT7_config_form_configure(form)
```



LabJack - AIN Configuration Form

Device Type: T7

Connection Type: ANY

Device ID: ANY

AI2: Offset and Slope

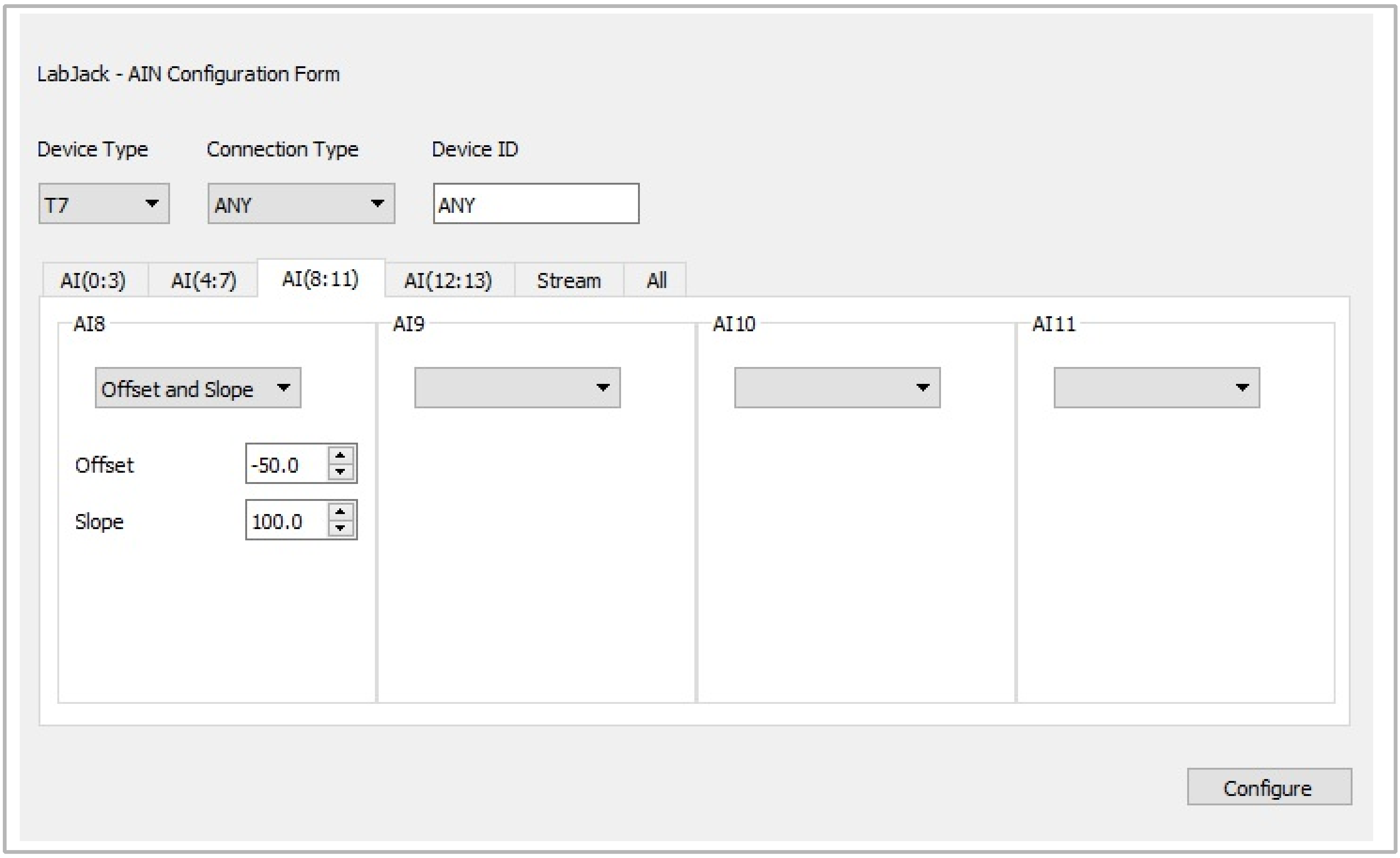
Offset: -50.0

Slope: 100.0

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_AIN 502")

ljainT7_config_form_configure(form2)
```



LabJack - AIN Configuration Form

Device Type: T7

Connection Type: ANY

Device ID: ANY

AI8: Offset and Slope

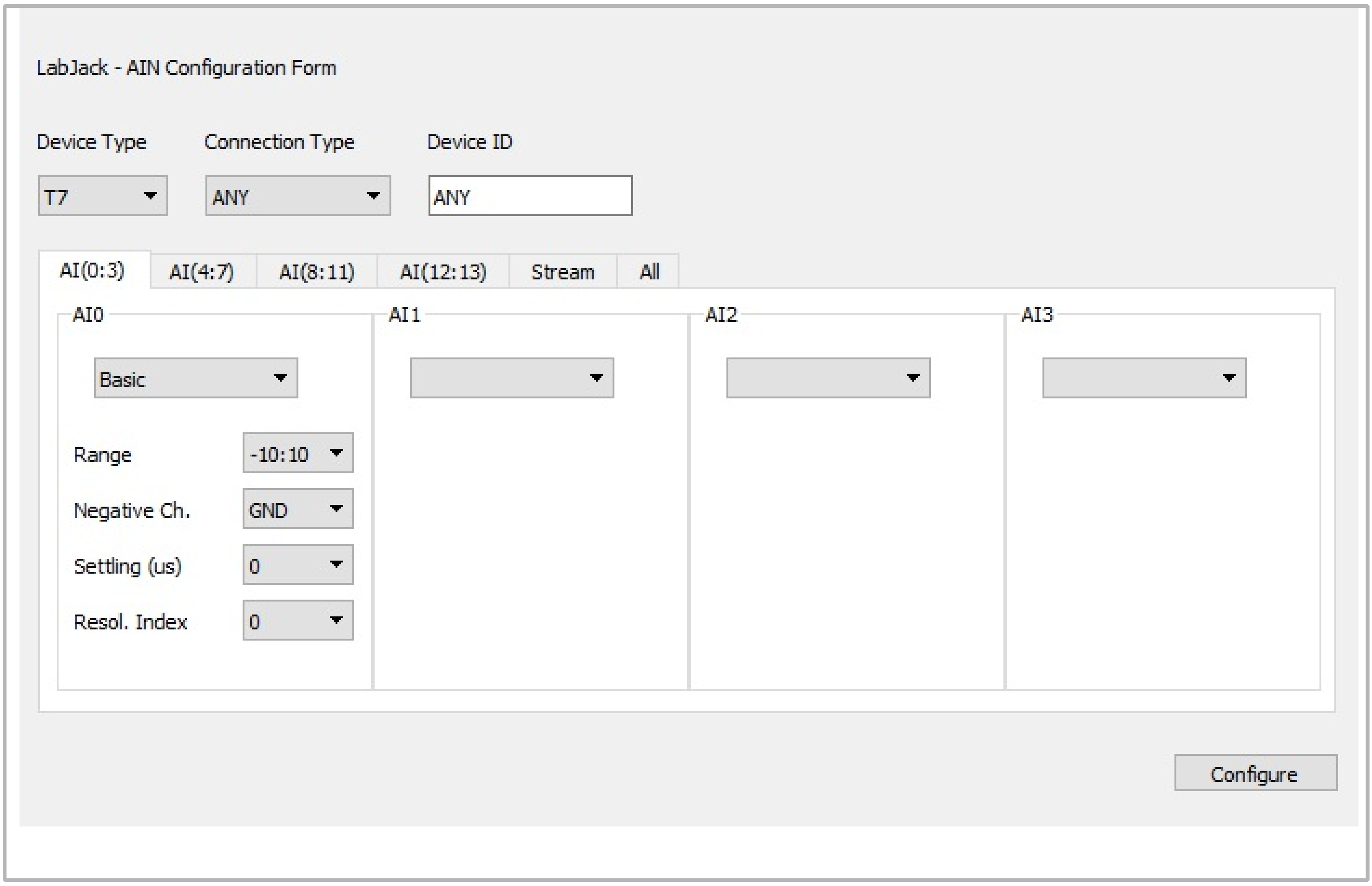
Offset: -50.0

Slope: 100.0

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_AIN 503")
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

Results

Before use, the device should be opened first, and afterwards should be closed again.

```
dev1 := ljdevice_open("any", "any", "any")
```

There will be N measurements, which is set below. The first 10 measurements will be automatically displayed in the table. After all N measurements, the data is exported to a xlsx file. We read the temperature at AIN2 and at AIN8. The voltage is read at AIN0.

```
N := 20
Ta := vector_create(N, false, 0)
Ta8 := vector_create(N, false, 0)
V0 :=vector_create(N, false, 0)
ID := vector_create(N, false, 0)
TableH := ["ID", "Temp. at AIN2", "Temp. AIN8", "Voltage at AIN0"]
tt :=ljdevice_read(dev1, "AIN2_EF_READ_A")
Ta[0] = ljdevice_read(dev1, "AIN2_EF_READ_A")
Ta8[0] = ljdevice_read(dev1, "AIN8_EF_READ_A")
V0[0] = ljdevice_read(dev1, "AIN0")
ID[0] = 0 + 1
 Temp := [ID[0], Ta[0], Ta8[0], V0[0]]
Table := table_create(Temp, TableH)
Tablef := table_create(Temp, TableH)
for(i := 1; i < N; i += 1)
{
  Ta[i] = ljdevice_read(dev1, "AIN2_EF_READ_A")
  Ta8[i] = ljdevice_read(dev1, "AIN8_EF_READ_A")


 V0[i] = ljdevice_read(dev1, "AIN0")
 ID[i] = i + 1
  Temp = [ID[i], Ta[i], Ta8[i], V0[i]]
 TT := table_create(Temp, TableH)
 Tablef = join_mat_rows(Tablef, TT)
 if(i < 10)
 {
  Table = join_mat_rows(Table, TT)
 }
 sleep(2000)
}
```

Table =

ID	Temp. at AIN2	Temp. AIN8	Voltage at AIN0
1	23.781	-62.211	2.814
2	23.434	-66.23	2.814
3	23.418	-75.349	2.814
4	23.426	-76.312	2.813
5	23.41	-76.257	2.813
6	23.426	-76.075	2.814
7	23.426	-76.264	2.814
8	23.434	-76.028	2.814
9	23.434	-75.996	2.814
10	23.41	-76.241	2.814

```
ljdevice_write(dev1, "IO_CONFIG_SET_CURRENT_TO_DEFAULT", 1)
ljdevice_close(dev1)
```

The results can be displayed in a table simultaneously. The table is generated automatically, and 10 measurements will be displayed.

Exporting Values to Excel

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

```
excel_write("measurements1.xlsx", "Sheet1", "A3", Tablef)
dat := date("d/m/y")
excel_write("measurements1.xlsx", "Sheet1", "A1", "Date")
excel_write("measurements1.xlsx", "Sheet1", "A2", dat)
tim := time(":")
excel_write("measurements1.xlsx", "Sheet1", "B1", "Time")
excel_write("measurements1.xlsx", "Sheet1", "B2", tim)
```