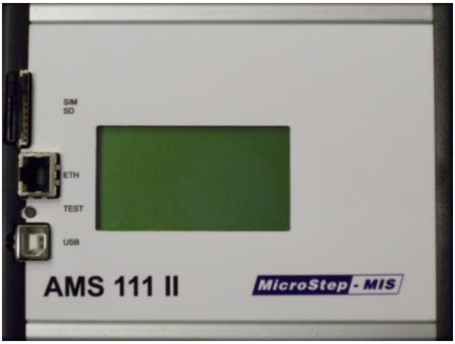
AWS Setup



Configurations

Version 1

Quick Guide

April 2015

Trademarks

Windows is a registered trademark of the Microsoft Corporation.
All other trademarks or registered trademarks mentioned belong to their respective owners.

Version of edition

Version of edition: 1
Date of the edition: 2015-04-17

Authors

Editorial team: Mr. Csaba Ruman, MicroStep-MIS

Address

MicroStep-MIS
Monitoring and Information Systems
Čavojského 1
841 04 Bratislava 4
Slovakia
Tel.: +421 2 602 00 100
Fax: +421 2 602 00 180
info@microstep-mis.com

MicroStep-MIS develops, manufactures and markets monitoring and information systems. The key fields of our activities are meteorology, aviation, environmental sciences, seismology, power engineering, civil defense and tourism.

MicroStep-MIS operates worldwide. Our core customer groups are airports, meteorological and seismological institutes, environmental authorities, industry, power stations and electricity distribution companies.

Table of Contents

1. Open the project file

The project file for a station is stored in a file with `.DLX` file extension. To open a project file, double click on that file and it should be opened by the **AWS Setup** program automatically. If the AWS Setup is not associated with the DLX file extension, then it is possible to open this file from the AWS Setup program.
The program is capable to open even an older file with `.CBU` file extension, but the changes are saved in newer DLX file.

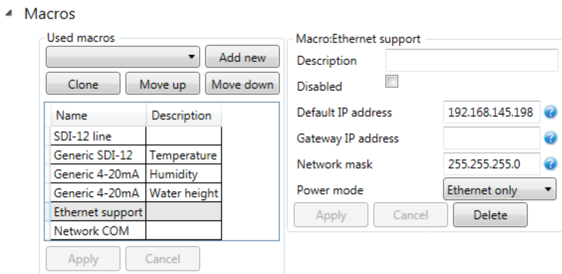
2. Make changes

When the project file is opened you can make changes in the configuration. You can add support for a new sensor, or change something in existing configuration.

2.1 Using macros

The macro feature of AWS Setup can simplify the creation of configuration tasks. For example if you want to add a sensor, you have to define a variable, an analog channel, calculations and events. These tasks may be confusing and error prone.

Macros simplify the configuration, because they execute these steps at once. You just have to change some parameters and the macros will do the process instead of you.



MicroStep-MIS can create various types of macros, for example:

- support for generic or one specific type of sensor
- communication protocol
- algorithms
- network settings
- much more

If the preinstalled macros are not enough, you can contact MicroStep-MIS and we will create the requested macros. Once you have a macro for a sensor, you can add multiple sensors of the same type in the same project or use them in other projects as well.

Every macro may have parameters, which can be used to specify the details. For example this parameter may be the channel number of sensor and the variable. You can create copies of this macro and then change the channel and the variable. By doing this you have added a new sensor to configuration. The algorithm and the settings are the same, but the difference is only in the channel and the variable (more generally: in parameters).

3. Example configurations

In this chapter you can find some example methods of sensor configurations. There may be other methods, and these are just for reference. If you don't find the sensor in this chapter, you can define one from the generic way (Chapter 3.1). If you are in doubt, feel free to consult with MicroStep-MIS or see the full documentation about using AWS Setup.

3.1 Generic analog sensor

These steps are universal for any kind of analog sensors which have voltage output.

From the sensor's manual you have to know the measuring range and the corresponding value to these values.

Voltage	Quantity value	Example voltage [mV]	Example value [°C]
UL	VL	0	-40
UH	VH	1000	60

For example voltage UL = 0 V corresponds to temperature VL= -40 °C and voltage UH = 1 V corresponds to temperature VH = 60 °C.

In analog section you have to create analog channel and select a differential channel which is not used in the project. You have to create a variable, where the result will be stored. Enter a name in variable section (unique in project, no spaces, and maximum 8 characters) then click **Create new** button.
By convention we use a prefix "**a**" for actual value, then capital abbreviation of the measured quantity (e.g **T**emperature of **A**ir) then a number (e.g. **1**) which make a difference between multiple sensors (e.g. full name of variable is **aTA1**).

You should define how often the value is measured in **Period** section.
In **minimum** field use the minimal value (VL) of the measuring range of the sensor (e.g. -40 °C) and in the **maximum** field use the maximal value (VH) of the measuring range of the sensor (e.g. 60 °C)

Set the **polarity** to **unipolar** if the voltage is always positive for all measuring range and **bipolar** if the voltage has negative parts as well.

Select a gain, for which the voltage range is greater than the output range of the sensor. (E.g. **Gain 2** with range up to 1.25 V for sensor with output from 0 to 1 V)

Use calculation **Scaling.** This will convert the measured voltage to the measured quantity value. You have to define a polynomial (coefficients **k_1, k_2, k_3, k_4**) which converts this value. To calculate the values for linear sensors use these formulas:

$$k_1 = \frac{V_H - V_L}{U_H - U_L}$$

$$k_0 = V_L - k_1 \cdot U_L$$

$$k_2 = 0$$

$$k_3 = 0$$

Please note that UH and UL are in **millivolts for ADF** (differential) channels and in **volts for A** (single ended) channels.

Example:

$U_L = 0mV \qquad V_L = -40°C$

$U_H = 1000mV \qquad V_H = +60°C$

$$k_1 = \frac{V_H - V_L}{U_H - U_L} = \frac{60-(-40)}{1000-0} = \frac{100}{1000} = 0.1$$

$$k_0 = V_L - k_1 \cdot U_L = -40 - 0.1 \cdot 0 = -40$$

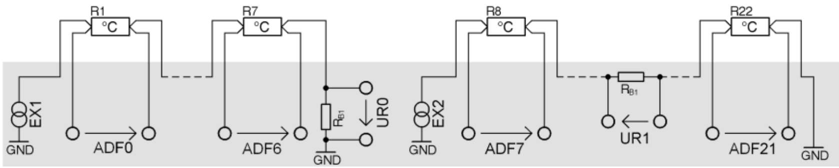
$$k_2 = 0$$

$$k_3 = 0$$

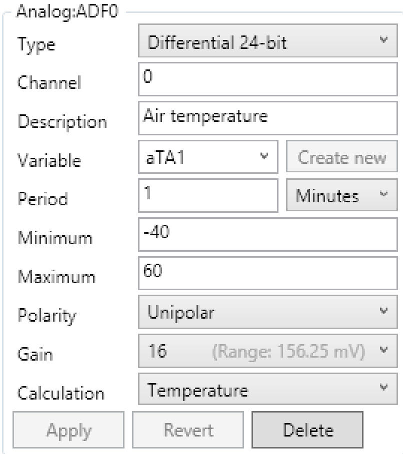
3.2 Temperature

3.2.1. Pt100

Temperature is measured using Pt100 thermometers. The data logger supports measuring the resistance, then convert this resistance to temperature using the **Temperature** calculation.



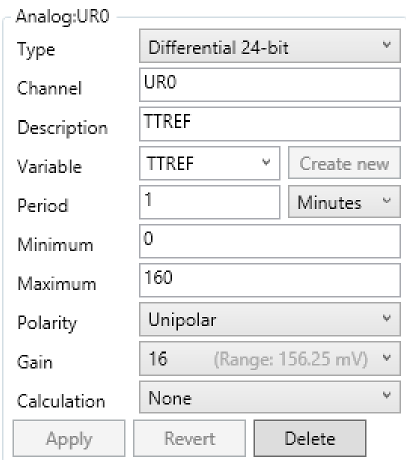
Connect the Pt100 to a free differential channel ADF.



Use **Gain 16** to ensure that voltage will remain in measurement range for temperatures in required range. Use **Calculation Temperature**. Save the measured value to **variable** aTA1 (**a**ctual **T**emperature of **A**ir no. **1**)

If you choose in range ADF0 – ADF6, include a measurement of channel UR0 into configuration. For higher channels include channel UR1.

This is required for measuring the electric current over the reference resistor. Save that voltage for example to **variable** TTREF (temperature reference) every minute.



3.2.2. Analog output

The temperature channel for sensors with analog output was described in chapter 3.1 (Generic analog sensor).

3.3 Relative Humidity

3.3.1. Analog sensor

For analog sensors set the connected analog channel according to sensors' parameters. For example if your analog humidity sensor has an output range 0-1 V which means 0-100 %RH, adjust the measured value by calculation **Scaling**. This is a polynomial adjustment. Change the coefficient **k1** (gain) to **100** for single ended analog channels or **0.1** for differential channels. The reason for discrepancy is that single ended values are measured in volts and differential channels in millivolts.

Set channel **minimum** to 0 and **maximum** to 100. If a value is out of this range, the value becomes invalid, which may be represented differently in messages. See messages for details.

3.3.2. Sdi-12 digital sensor

Sdi-12 digital bus is located on serial line **COM5**. To create a measurement, create an **event** with **action Send** (`requestRH1`) and an **event** with **action Receive** (`responseRH1`). For send event set the **interval** to desired sample interval and for receive event set the **validity** to sample interval + 1 second, which ensures that the measurement will not expire if the response is delayed. The measurement will expire if the sensor is not responding or the value cannot be parsed.

<u>requestRH1</u>
Set the output message with text for example "`1M!\r1D0!\r`" which means the following: start measurement in device with address 1, then get the result from the device with address 1.

<u>responseRH1</u>
The device returns "`1+34.45+24.14\r\n`" for this command. Data logger combines the command with the response and from the serial line COM5 is returned: "`1D0!1+34.45+24.14\r\n`"

This result is parsed in an input message (of event `responseRH1`), which has to be defined for example with start "`1D0!1`" and end "`\r\n`".

Between these strings is the response from the sensor. To parse these values create a message in Cells section, where you can assign the values to variables. Create two cells:

- variable aRH1 with format **`%f`** to store +34.45
- variable aTA1 with format **`%f`** to store +24.14

3.4 Wind direction

3.4.1. Analog

Example sensor: Thies Wind direction sensor

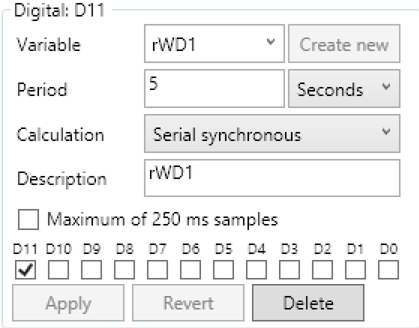
Voltage	Direction value
0 V	0 °
5 V	359 °

Connect the sensor to a **Single ended** analog channel.
Set variable to **aWD1** and click Create new button.
Set the **minimum** value to **0** and **maximum** to **359**.
Set **calculation** to **Scaling**. Use the following scaling coefficients:

k_0	0
k_1	71.8
k_2	0
k_3	0

3.4.2. Thies first class

Connect the `Data` pin of the sensor to data loggers' DIN11 pin and the `Clock` pin of the sensor to DOUT3. In AWS Setup create a **digital** on pin D11 with **Calculation Serial synchronous**.

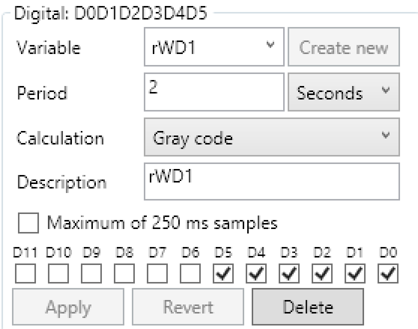


3.4.3. Gray code

Connect the sensor to free digital inputs which are side by side. (e.g. D2, D3, D4, D5, D6, D7)

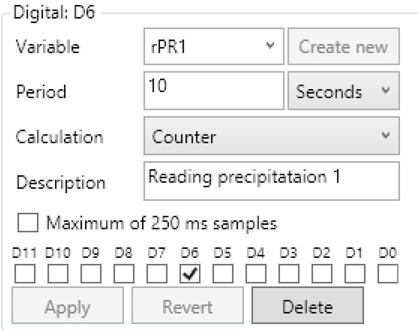
Connect the lowest bit to the lower bit number in the data logger. (E.g. sensor’s G0 to data logger’s D2, then sensor’s G1 to data logger’s D3 etc.)

In AWS Setup add a digital and select the connected pins. Use **Calculation Gray code**.

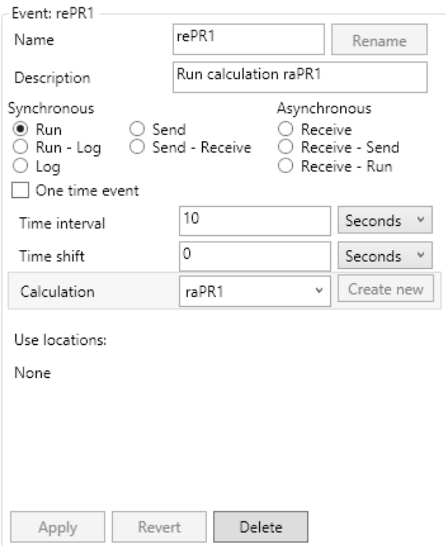


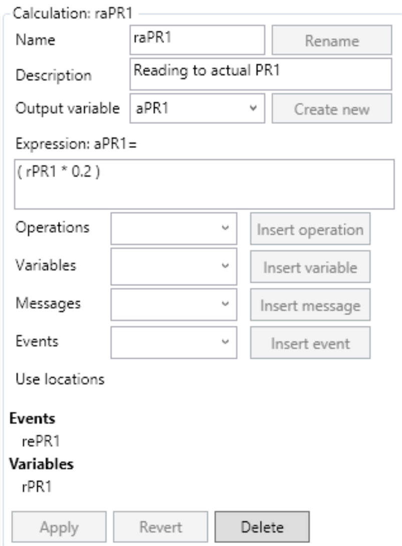
3.5 Precipitation

Create a **digital** measurement, which will count pulses on digital pin.



Add an event, which runs a calculation in the same interval as the digital is sampled. From this event create a calculation which will adjust the measured count value to mm units.

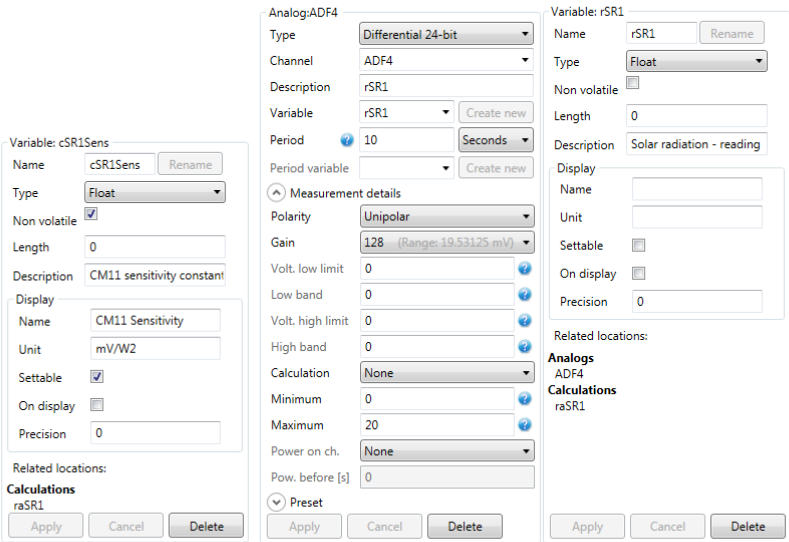




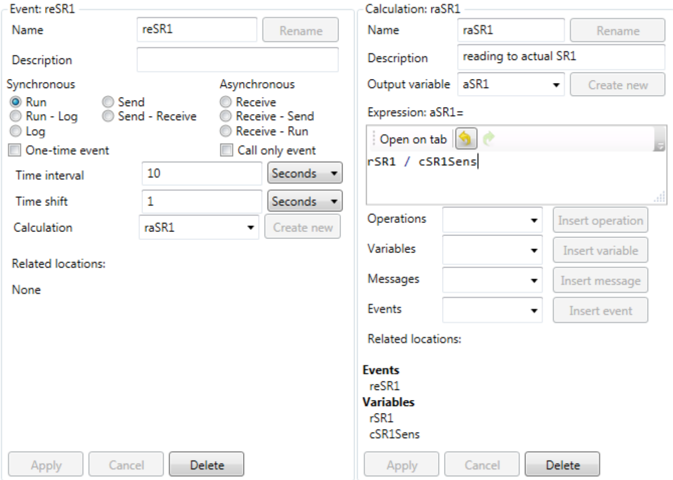
3.6 Solar radiation

Example sensor: Kipp & Zonen CM11
This sensor has a sensitivity coefficient, which differs from sensor to sensor. That's why we have to define 3 variables:
rSR1 – **S**olar **R**adiation **r**eading in millivolts
aSR1 – **a**ctual **S**olar **R**adiation in W/m^2
cSR1Sens – **S**olar **R**adiation **sens**itivity **c**onstant in $mV/(W/m^2)$

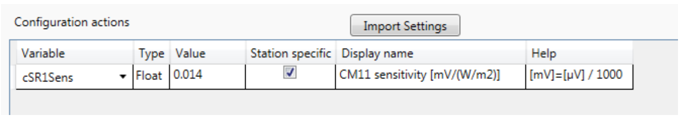
The sensitivity constant has to be **non-volatile** to store the value even after disconnected power.



The voltage is sampled directly into variable rSR1 without calculation. The value of aSR1 is updated in a calculation raSR1 (**r**eading to **a**ctual **SR1**). This calculation contains the following equation:
rSR1 / cSR1Sens
and stores the value into variable **aSR1**. The calculation is started by a synchronous event with the same running **time interval** as the sample time of analog channel. The **time shift** is 1 second, to ensure that the analog voltage value rSR1 is sampled.

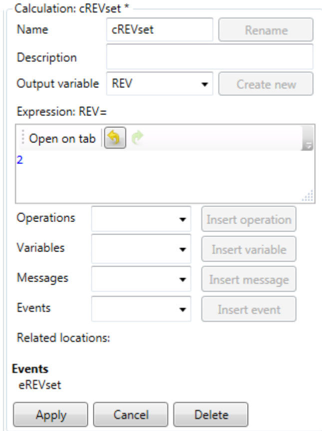


Add a row into the **Configuration actions** section on tab **Project**. Use the sensitivity constant's name in **variable** column and check the **Station specific** checkbox. This will ensure that after configuration in AWS Service, the user will be prompted to enter the sensitivity according to the used sensor. Set a default **value** to 0.014 for setting an approximate value as a default. (14 μV/(W/m^2) = 0.014 mV/(W/m^2))

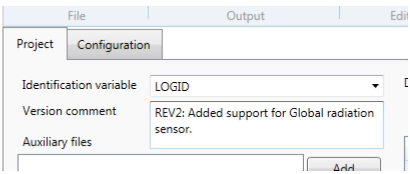


4. Compile the configuration

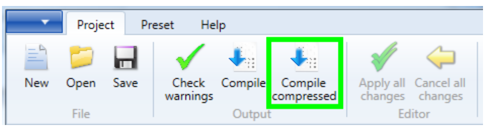
After changes in the configuration don't forget to change the revision number, so you can easily identify which revision is your configuration.
Usually the revision is stored in **REV** variable. The value is set in **cREVset** calculation. You can increment the number by one, to indicate that the revision is newer than the previous.

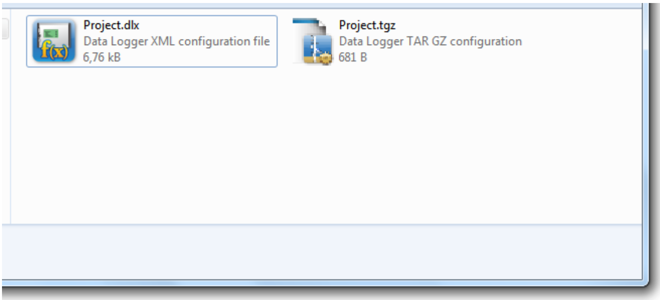


It is also possible to enter a comment in the **Project** tab. This **Version comment** will be shown in AWS Service.



When you finished the changes, click on **Compile compressed** button on the toolbar. This will generate a TGZ file next to the DLX file. If the TGZ file already exists, it will overwrite it with newly generated version.





5. Configure data logger

Use **AWS Service** program to write configuration to the data logger. Double click on the created `TGZ` file. If you have the AWS Service installed, it will begin the configuration process. If the AWS Service is installed, but another program opens the `TGZ` file, then probably you have changed the association of `TGZ` file extension. In this case select the file and open its properties in Windows. Then select the AWS Service as the default program for opening this `TGZ` file extension.

AWS Service can maintain multiple versions of configurations for each station. This helps to prevent confusion between configurations and stations. When you have tested that the changes you made are working correctly (e.g. the newly connected sensor works), then you can add that configuration version to Station List of the AWS Service.

Open **AWS Service** and select the **Station List** icon. Click on **Edit**, then **edit types** button. In this list find the configuration type, for which you changed the configuration. Then click **Add version** button. Click on the rectangle and select the newly created TGZ configuration file. Click **Add version** button to add the configuration to the list of configuration versions for the selected configuration type.

Once the configuration is added to AWS Service, you can go to the stations with that AWS Service, or connect to them remotely and change the configuration of the stations. Don't forget to make backups before updating the configuration.

See the AWS Service User's Guide for more details.