

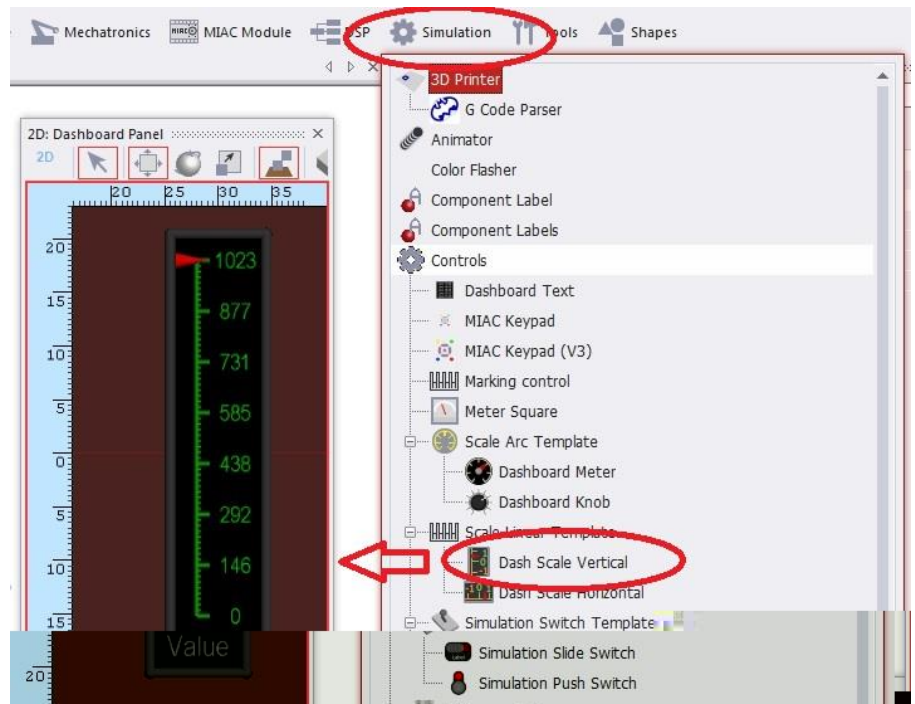
Example of creating a Flowcode 7 component for an I²C based device, such as a digital temperature sensor.

Create a new Flowcode project. It does not matter what the target is, although it helps if you choose a target that supports I²C.

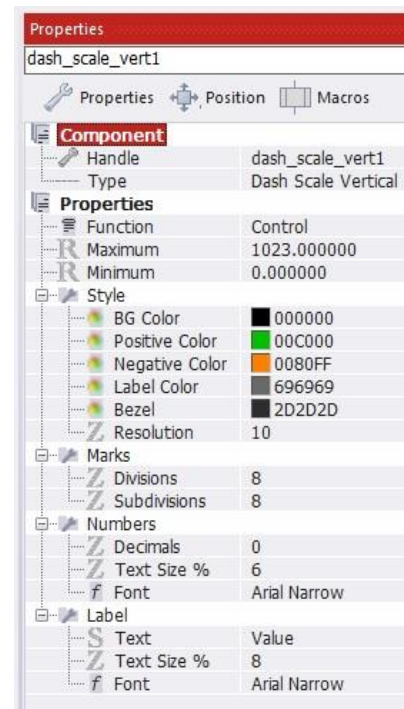
The main function of our chosen device is to provide us with a value for the measured temperature.

So we need a means to set this value when we are running the simulation of an application that uses our new component.

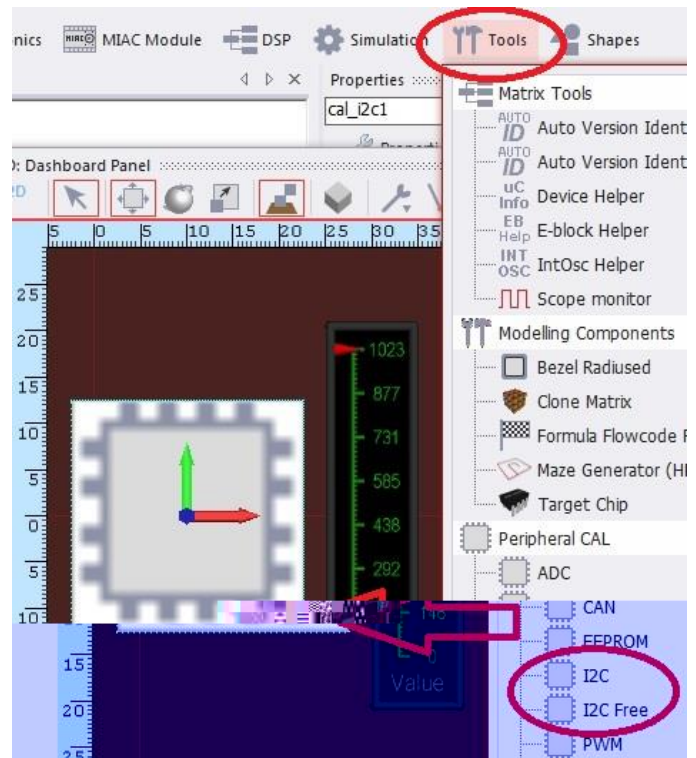
A linear scale is a good way to provide it, so let's add one. Select the "Dash Scale Vertical" from the "Simulation" list and drag it onto the panel.



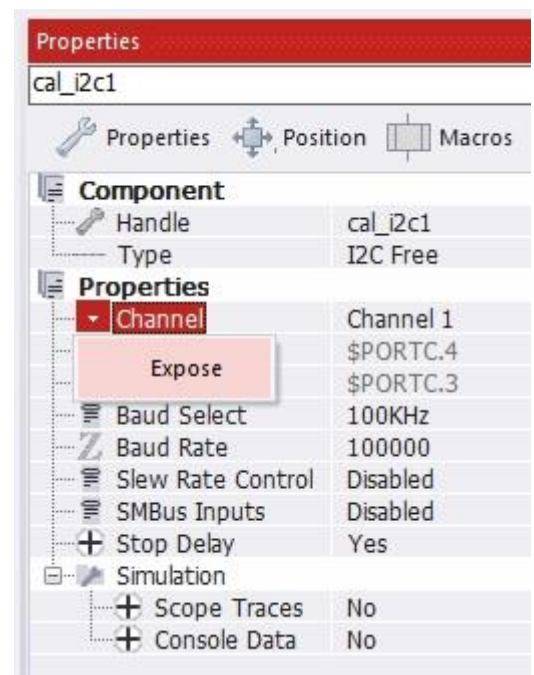
Set the scale component properties, such as the minimum, maximum and resolution values.



The device interface is I²C, so the Flowcode I²C CAL component will help us, drag one to the panel from Tools->CAL-I2C



Set the I²C properties, such as the Baud Rate, to match the device. The properties that are going to be user configurable in our new component, such as Channel, SDA and SCL, need to be set to “Expose”



From the device datasheet decide the API (macros) that you require to access the device functionality. For this example we are assuming a device that provides a digital value of temperature, but it could just as easily be a voltage, current or some other sensor.

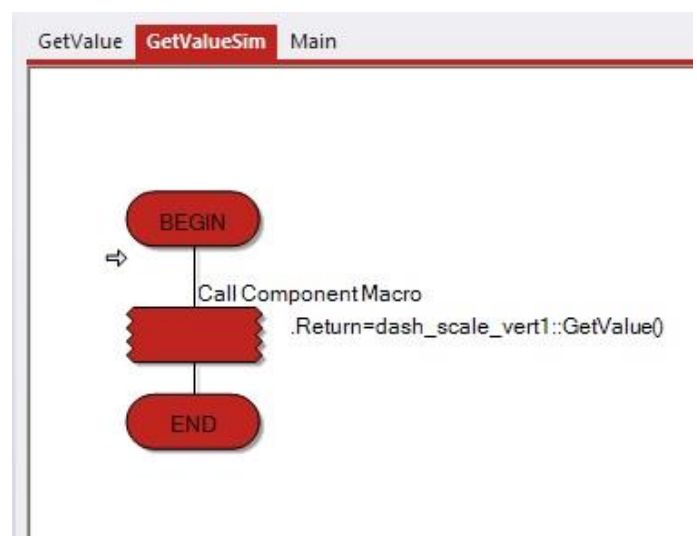
Create a new macro, called say GetValue, to do this go to menu MACRO->New
Enter the name of the macro, "GetValue", and set the Return type to UINT



This macro "GetValue" will be the macro that is part of the API we provide for the downloaded target component.

We also require a macro that provides the same functionality for the simulation.
So click on the GetValue tab, then go to menu MACRO->Duplicate, enter "GetValueSim" as the name of this new macro and click OK.

In GetValueSim macro add a call to
dash_scale_vert1::GetValue and put the value into
.Return



The I²C device will have an address that might be configurable.

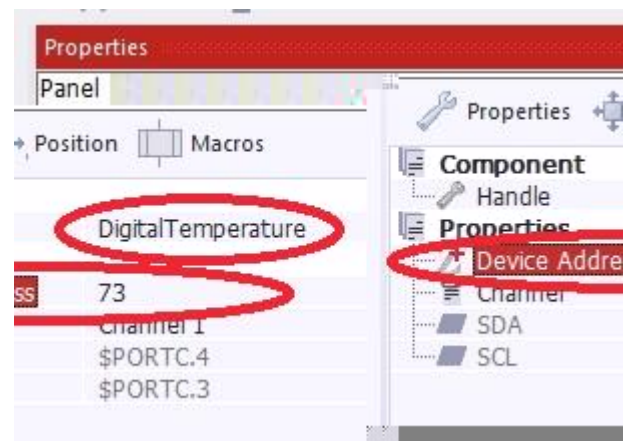
So here we add a property “Device Address” to our component, a panel property, and give it a default value.

We also give the property a variable name of “DEVICE_ADDRESS”



This is also a good point to give our component a name, such as “DigitalTemperture”

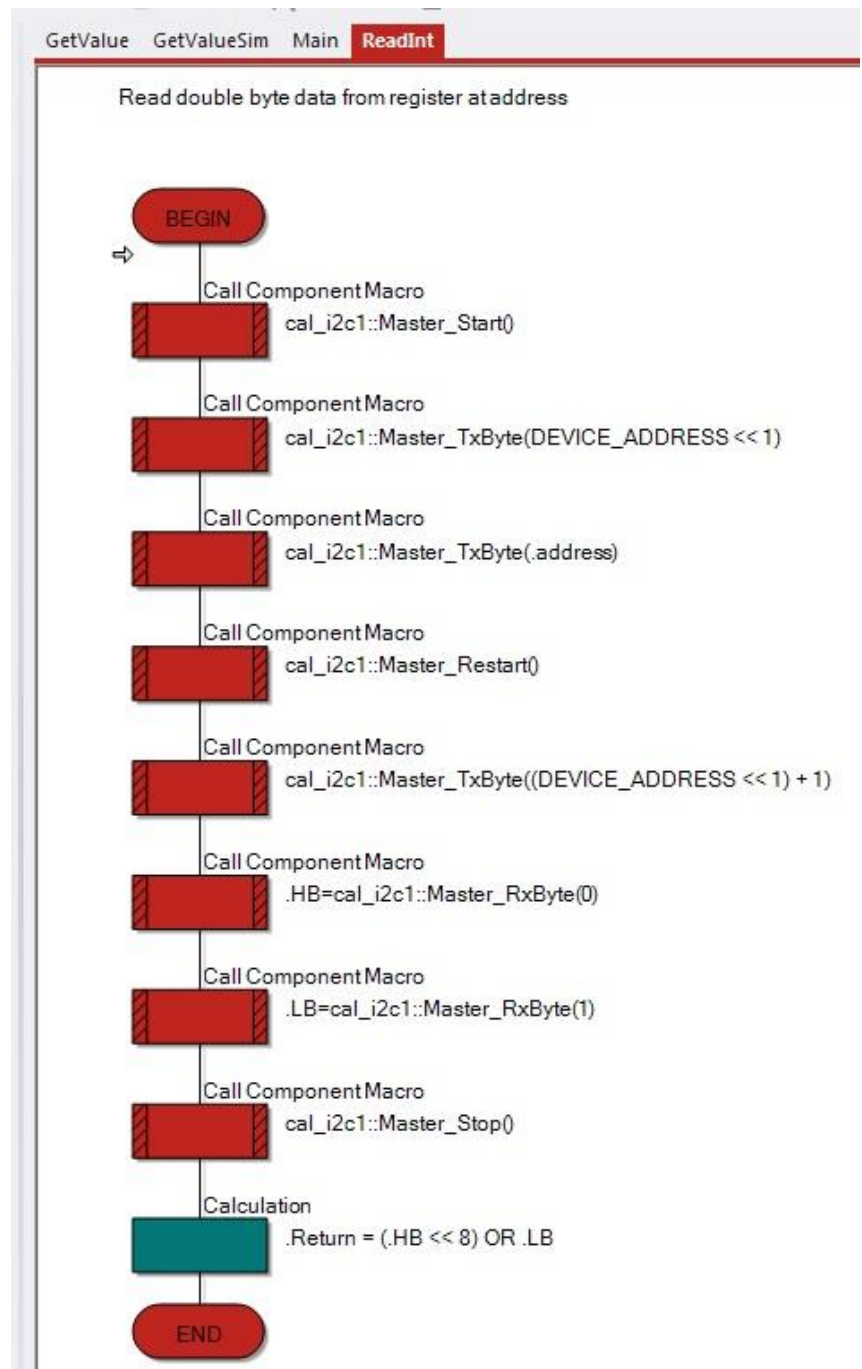
Note that we can also see our “exposed” properties from the I²C CAL.



Getting the value from the actual device involves reading a pair of register values from the device, so it makes sense for us to first write a generic macro that reads a pair of bytes from a given address in the device and returning it as an integer.

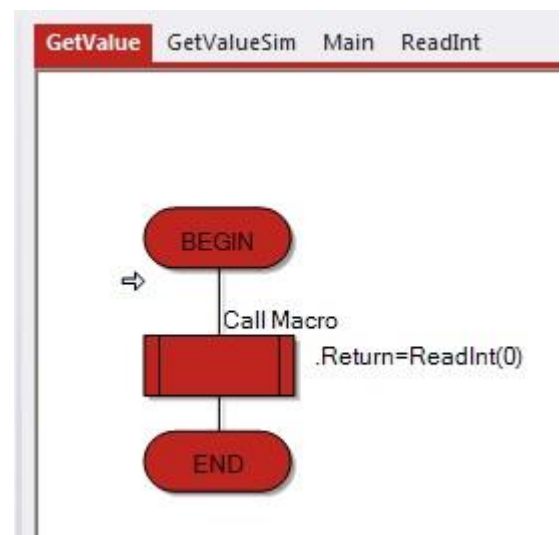
So here is a new macro called ReadInt that has a parameter “Address” and returns a UINT.

The devices’ I²C bus address is determined from our property DEVICE_ADDRESS



For this particular device the digital temperature is read from the first two locations of its internal registers.

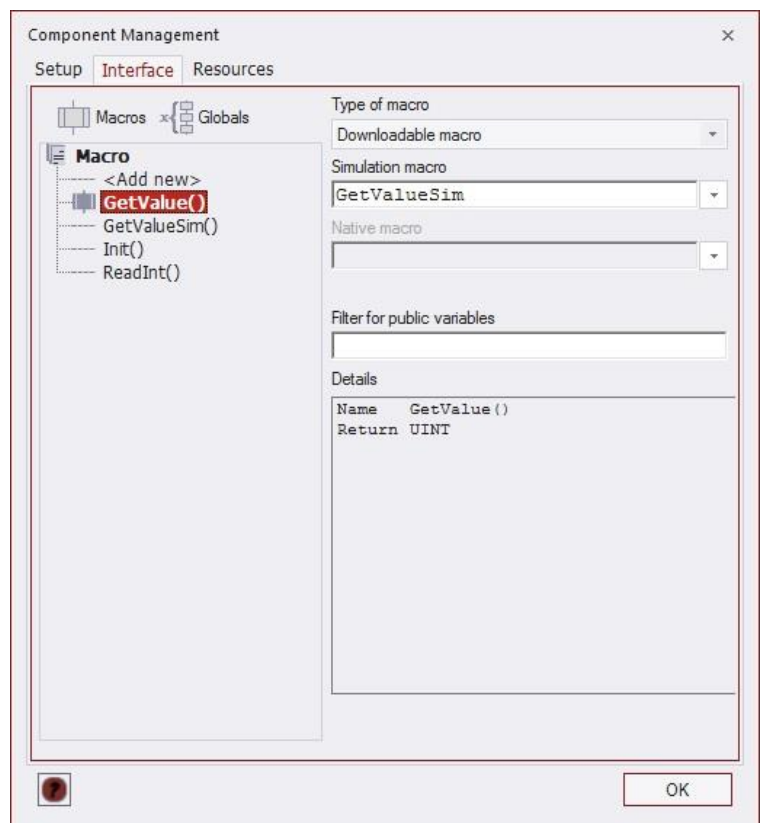
So our GetValue macro simply calls ReadInt with an address parameter of 0.



We are now ready to make our component.
Get the configuration dialog from
FILE->Component configuration

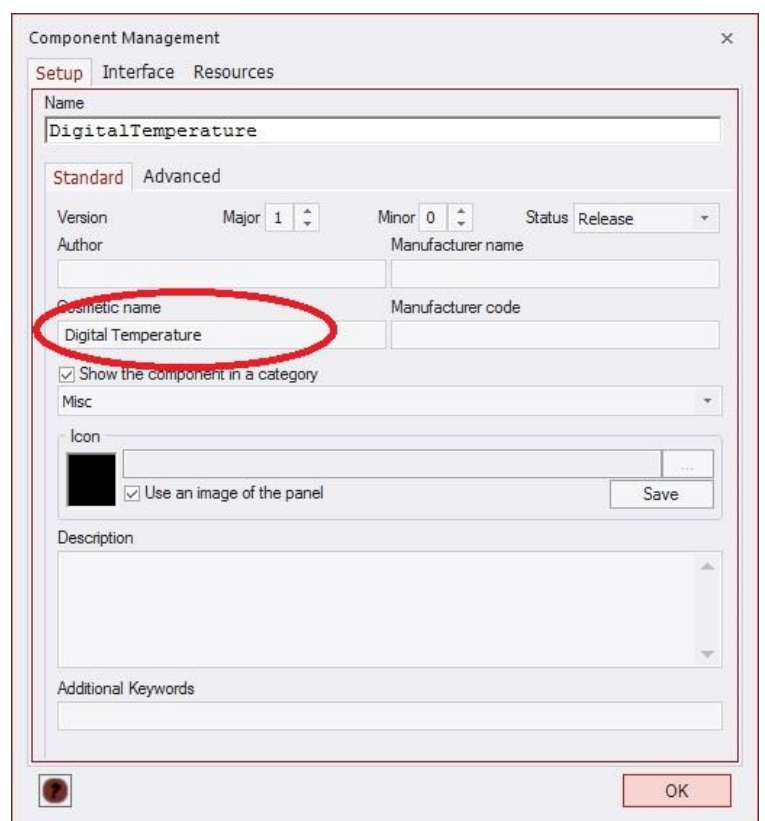
Go to the “Interface” tab and set the GetValue
macro as type “Downloadable macro” and set
the Simulation macro as “GetValueSim”

All others should be “Hidden macro”



Go to the “Setup” “Standard” tab and give the
component a Cosmetic name.

A description of the component and keywords
for the search can also be added.



Go to the "Setup" "Advanced" tab and click the GUID Edit button.

Click the "New" button and then "OK" to create a new GUID.

Then click the "OK" button to close the Component Management dialog.

If not already done, Save your project.

Now export this new component via the menu FILE->Export Component

Close Flowcode and re-open to create a test program. Your new component should appear in the default "Misc" list, if not search for it by name.

