INDUSTRIAL CONTROL DIRECT
A B&D Industrial Company

TRAINING

i^3 ...Display, Control, Communicate

Tutorial

Mathematics

IMO

Table of Contents

Introduction

The purpose of this tutorial is to demonstrate the ***i***^{3} mathematical functions and graphical representation capabilities with bar graphs, trends and meters.

The ***i***^{3} has basic mathematical functions such as add, multiply, subtract and divide but it has can perform floating-point maths. Therefore the ***i***^{3} has sine, cosine and tan functions amongst others in special Advanced Maths operations.

Programming Maths Functions

Design a program where the user can enter the year they were born and the ***i***^{3} will calculate how many days they have been alive for (approximately). The program will then have another screen where the user will enter the radius of a circle and the ***i***^{3} will calculate the area, circumference and diameter. Finally we will set up the trend function to display the result of a sine calculation.

Memory requirements of mathematical functions

The ***i***^{3} can handle floating point maths as well as standard integer maths. These calculations require different data types and the different data types require different sizes of memory to store them.

Data Types

BOOL - Boolean; A single bit. It can contain only the values '0' or '1', a.k.a 'FALSE' or 'TRUE'

BYTE - Byte; A string of 8 consecutive bits. Byte format is used more where the value of the data is not as important as the bit patterns (shifts and rotates).

WORD - Word; A string of 16 consecutive bits. Word format is used more where the value of the data is not as important as the bit patterns (shifts and rotates).

DWORD - Double Word; A string of 32 consecutive bits. DWORD format is used where the value of the data is not as important as the bit patterns (shifts and rotates).

INT - Integer; A 16-bit signed value. Integers are used where the value of the data is expected to be in the range of -32,768 to +32,767

SINT - Short Integer; An 8-bit signed value. Short Integers are used where the value of the data is expected to be in the range of -128 to +127.

DINT - Double Integer; A 32-bit signed value. Double Integers are used where the value of the data is expected to be in the range of -2,147,483,648 to +2,147,483,647.

UINT - Unsigned Integer; A 16-bit unsigned value. Unsigned Integers are used where the value of the data is expected to be in the range of -0 (zero) to 65,535.

USINT - Unsigned Short Integer; An 8-bit unsigned value. Unsigned Short Integers are used where the value of the data is expected to be in the range of 0 (zero) to 255

UDINT - Unsigned Double Integer; A 32-bit unsigned value. Unsigned Double Integers are used where the value of the data is expected to be in the range of 0 (zero) to 4,294,967,296.

REAL - Floating Point; A 32-bit value. Values are stored and operated on in IEEE single precision (six digit) format. Values range from -3.40282E+38 to +3.40282E+38.

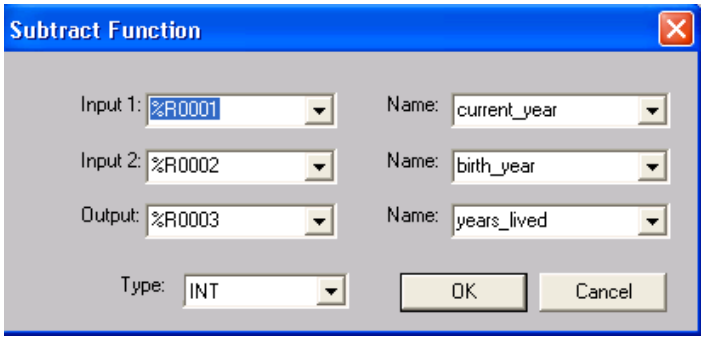
STRING - String; A variable-length succession of characters. Each character is represented by one byte.

Programming the ladder logic

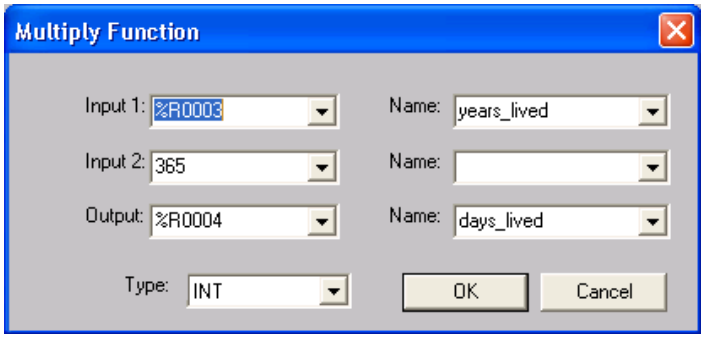
We will begin by writing the code for the days lived calculation. First we will subtract the year of birth from the current year to find the years lived. Then multiply the years lived by 365 (days in a year) to find the approximate days lived.

Insert a N/O contact at A1 and assign to %S07, Always ON (ALW_ON). This is so that the function blocks on this rung are always enabled.

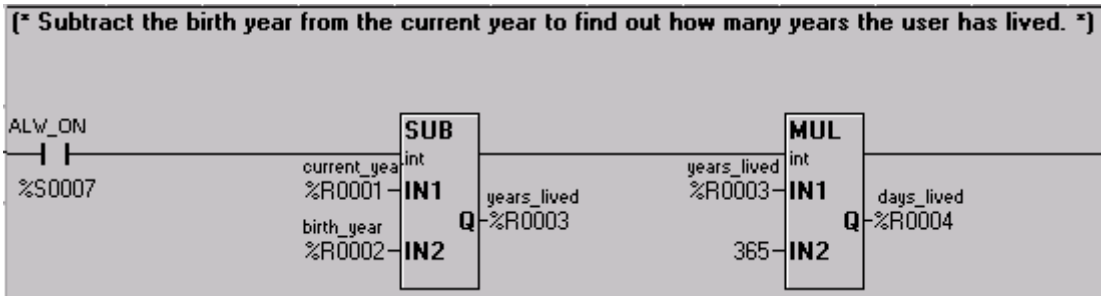
Select the Subtract function from the maths operations menu and insert it into the rung and enter the following detail.



Next insert a Multiply function to the right of the Subtract function on the same rung. Enter the details as shown below.



The line of ladder logic should now look like the logic below.

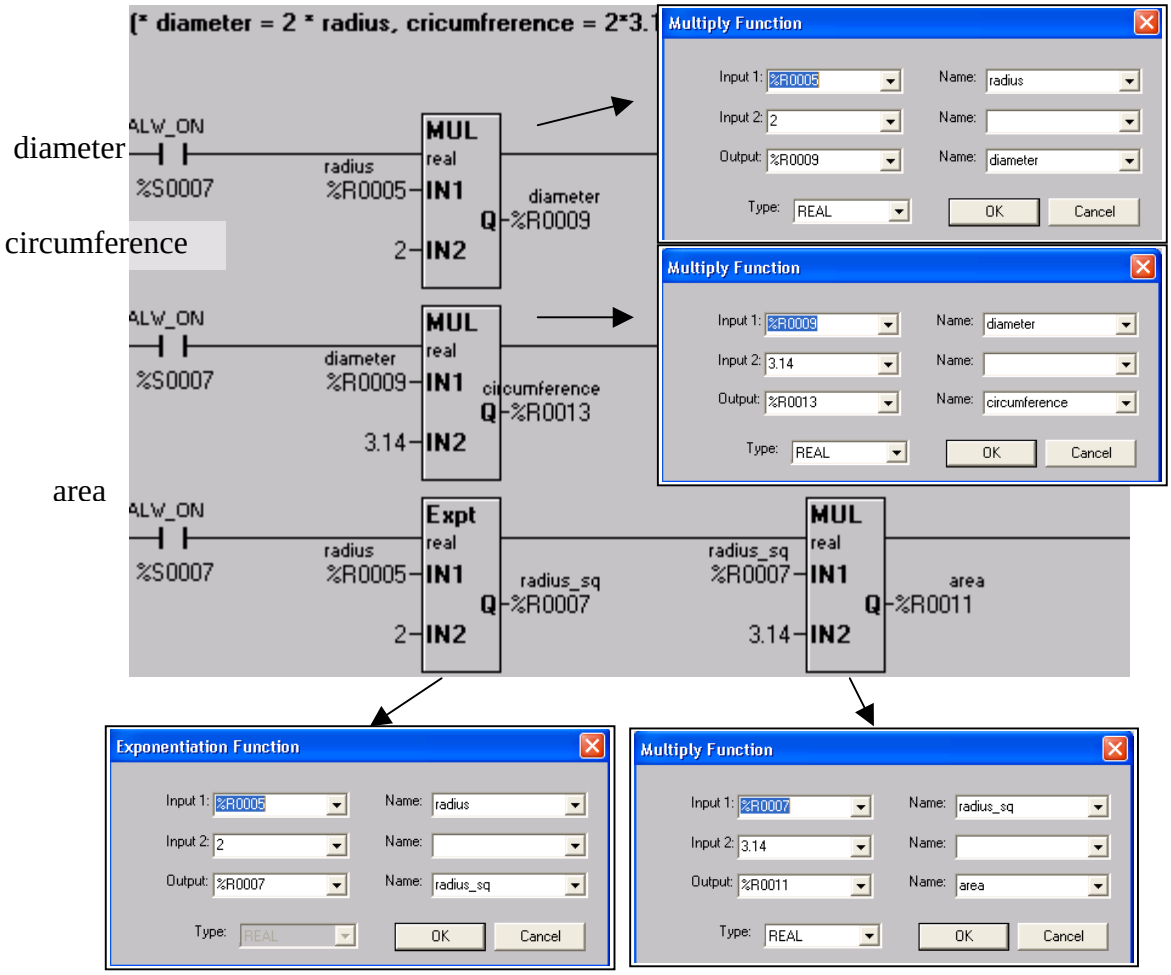


The formulas to calculate the diameter, circumference and area of a circle from the radius are as follows:

$d = 2r$
$circ = 2\pi r$
$area = 2\pi r^2$

d = diameter
r = radius
circ = circumference
π = 3.14

We are going to perform each calculation on a separate line, this is just so that it is easy to view. Each line will begin with an Always On (ALW_ON) contact, so that the functions are always enabled.



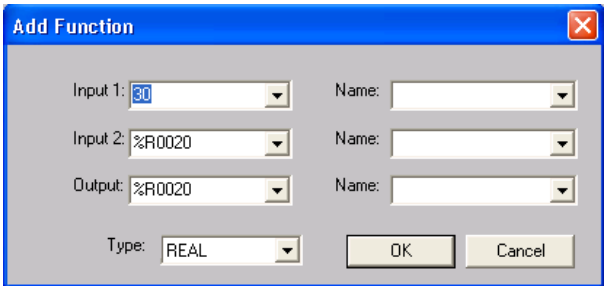
All of the calculations performed are using the data format Real. These require two registers, please note that the register addresses are at least 2 consecutive numbers apart. i.e. the variable “radius” is stored in %R05 and %R06 but it is only addressed to %R05.

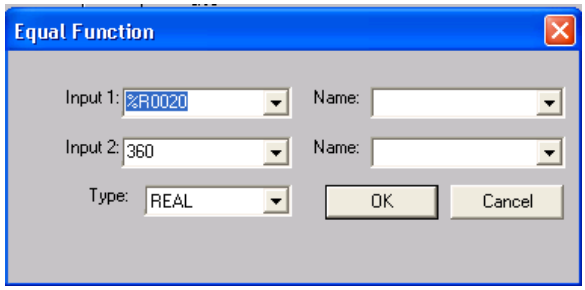
Lastly we are going to write the logic for the trend.

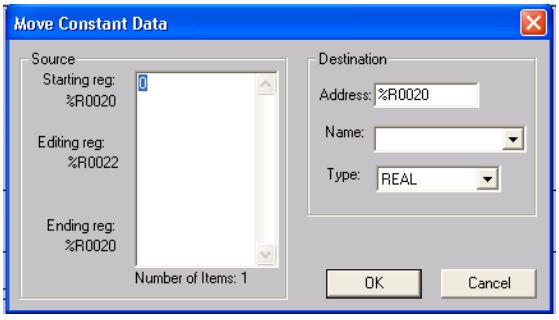
The sine function is going to operate on a set of values; 0-360 in steps of 30. To achieve this we are going to use a ten-millisecond pulse to add 30 to a register until it equals a set value.

First insert a N/O contact and address it to %S03 10mS system pulse. This will act as the trigger for the addition function, then on the same rung add a N/O coils that operates on a Pulse, addressing it to %M01. Then on the Rung below insert a N/O contact in the first column addressing it to %M01. This gives us an action that operates only on the rising edge of a transition.

Next add the addition function from the maths operations and insert it in the same rung as the N/O contact. Next insert an Equal to function from the Compare operations to the right of the addition block. Finally insert a Move function (MOV) to move 0 into the register when it's equal to 360. Insert the details into the functions as shown below.

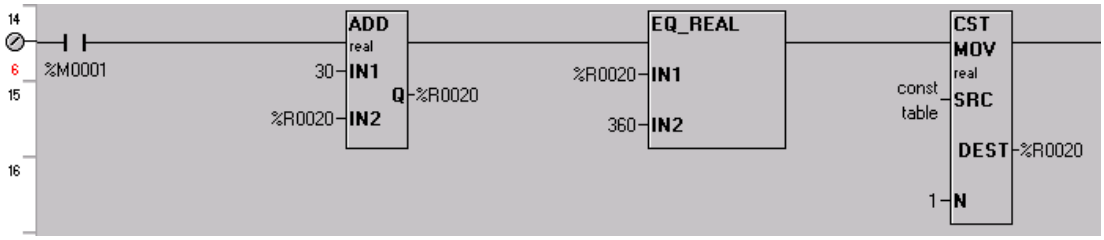






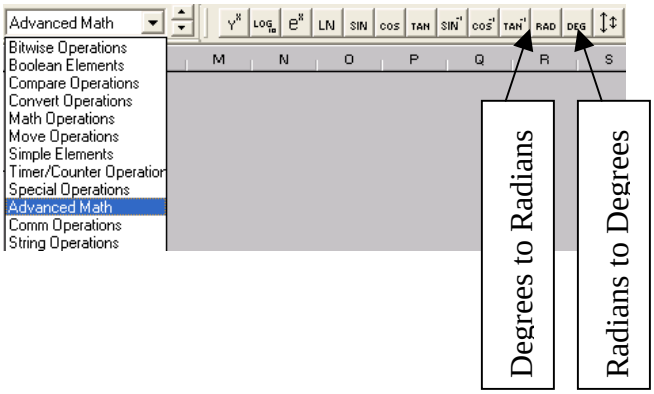
A constant move function was use as it allows for the movement of real values and keeps the calculation constant in Real data.

The rung of code should now like the ladder logic below.

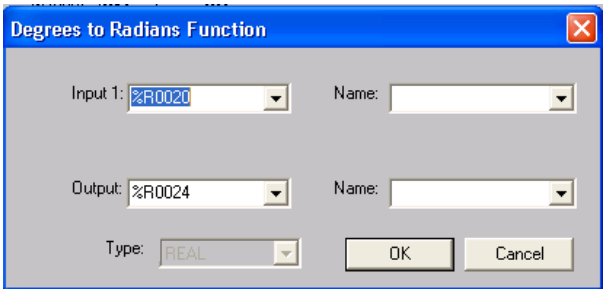


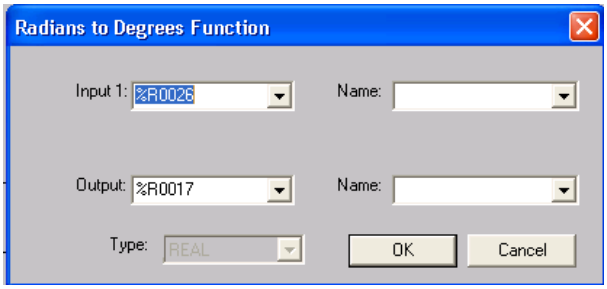
Now that the data to be operated on has been configured, insert a N/O contact on the Run below, in the first rung and assign to the Always On (ALW_ON) system bit. On this rung we are going to insert two conversion of data type functions and the sine function.

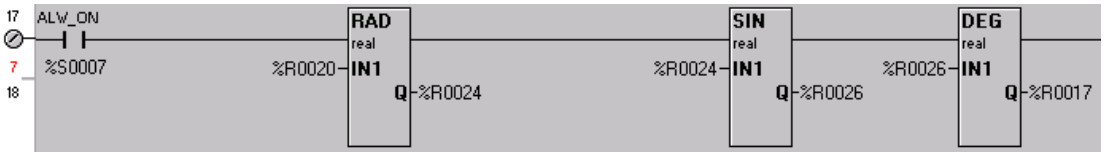
The sin, cos and tan functions all operate in radians. Thus the input to the function must be in radian and so the first function to insert on this rung will be a degree to radian function. This function is in the advanced math operations.

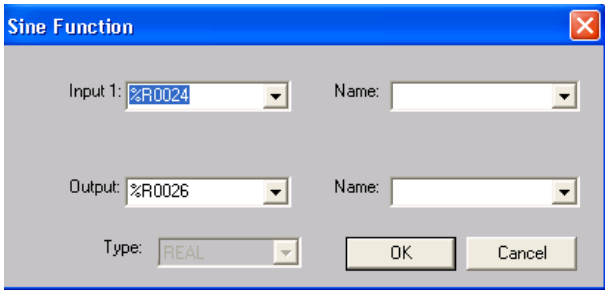


To the right of the degree to radian function we are going to insert the sine function. Select the sine function from the advanced maths operations. Lastly we are going to insert a conversion function to convert the function back from radian to degrees. Which will be more understandable to us.









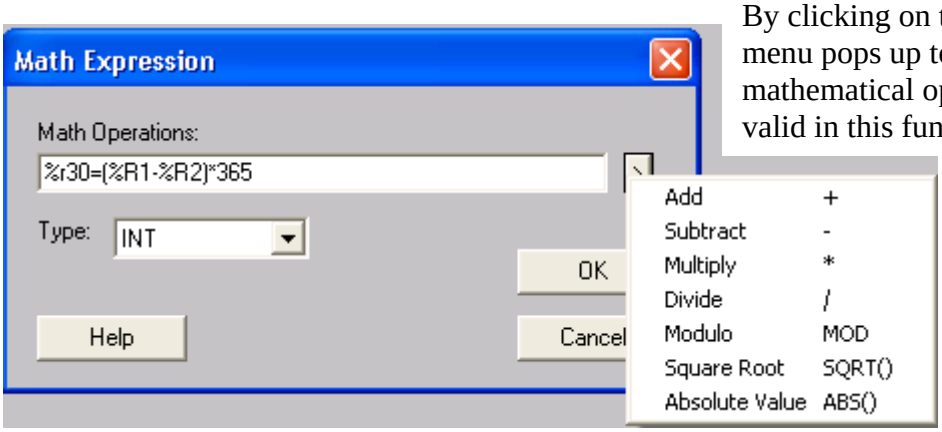
Equation Function

Remember that Real values require 2 consecutive registers

Calculations could also be performed in the equation function using only one function block and saving on memory.

The equation function allows the user to enter an equation into a single function block. To select the function block click on the icon  in the maths operations menu.

Double click on the function block to open up the editor.



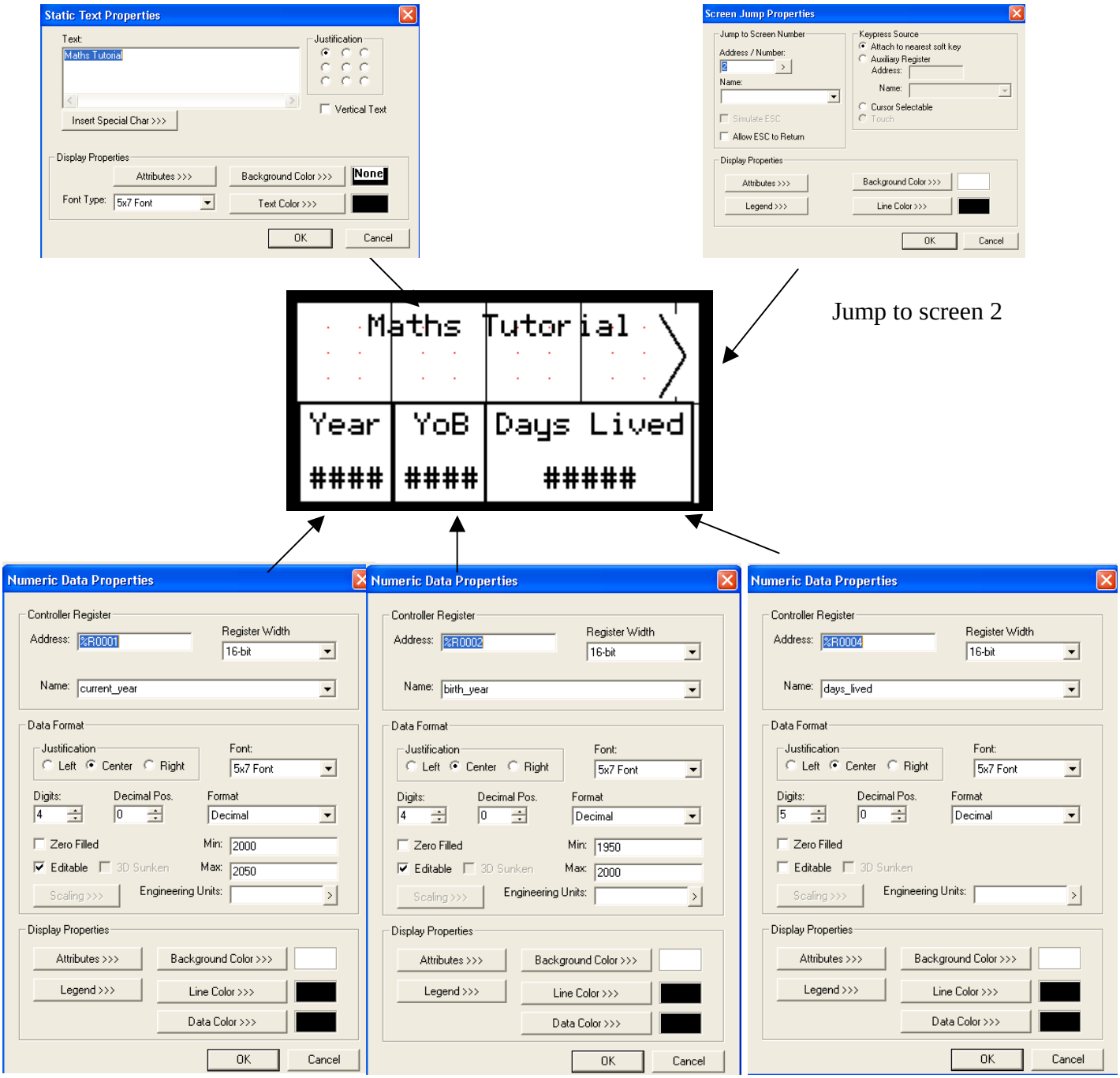
By clicking on the arrow a menu pops up to display what mathematical operations are valid in this function block

Enter the expression to calculate the days lived equation into the function as shown, starting with the result register equal to the expression. If you want to use the value in a register then enter the register address as shown.

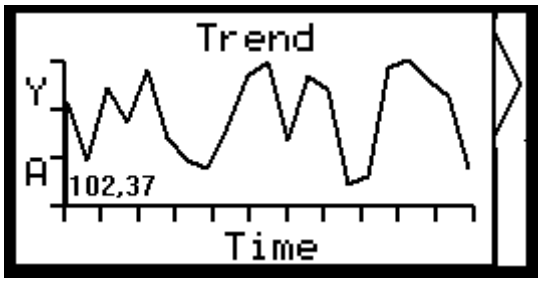
Screen Editor Programming

With all the ladder logic configured, we now need to edit the screens to display the information appropriately.

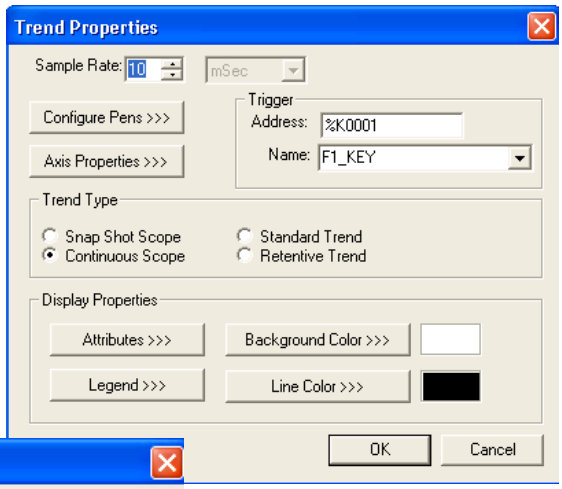
On the first screen we are going to have the user able to enter the current year and their year of birth. The screen will then display the days lived calculation result. There will also be a screen jump button so that the user can scroll through the screen.

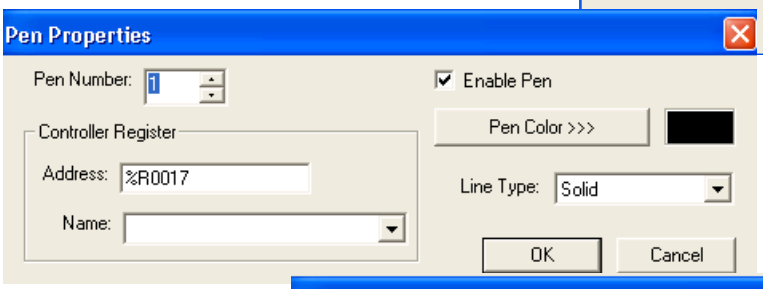


On the second screen we are going to display the trend of the sine function and have a screen jump to the next screen.



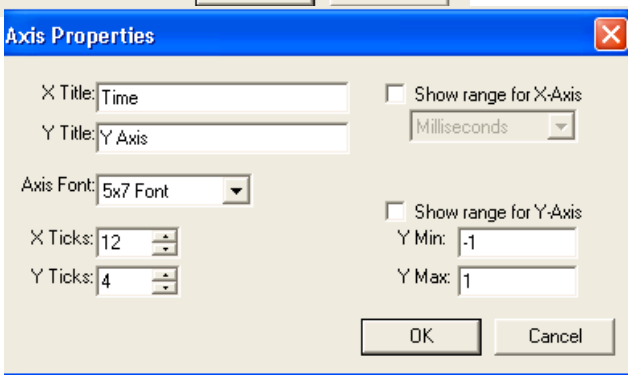
Click on the trend icon  and insert it into the middle of the screen. Resize the box to get a bigger trend, covering almost the full screen.
Setting the trigger to key 1 and having the trend type as a continuous scope, means that when the key is pressed there will be a trend, when it is release the trend stops.



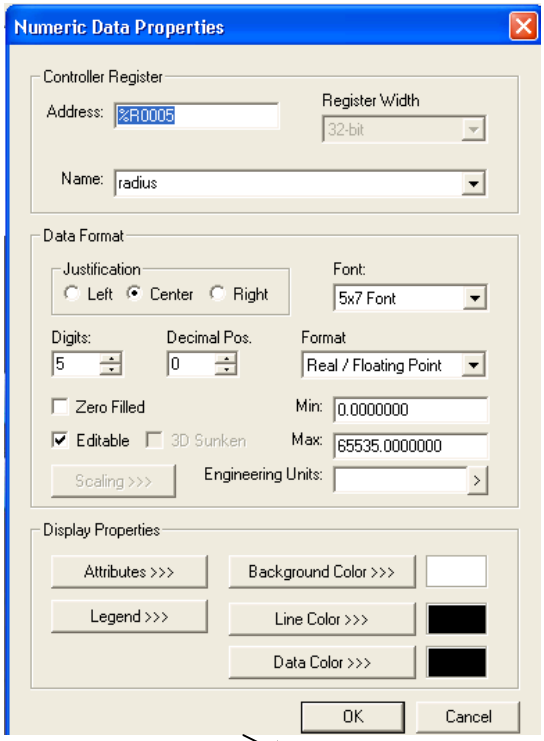


Set the pen to match the output register of the sin function

Set up the axis properties as shown opposite.



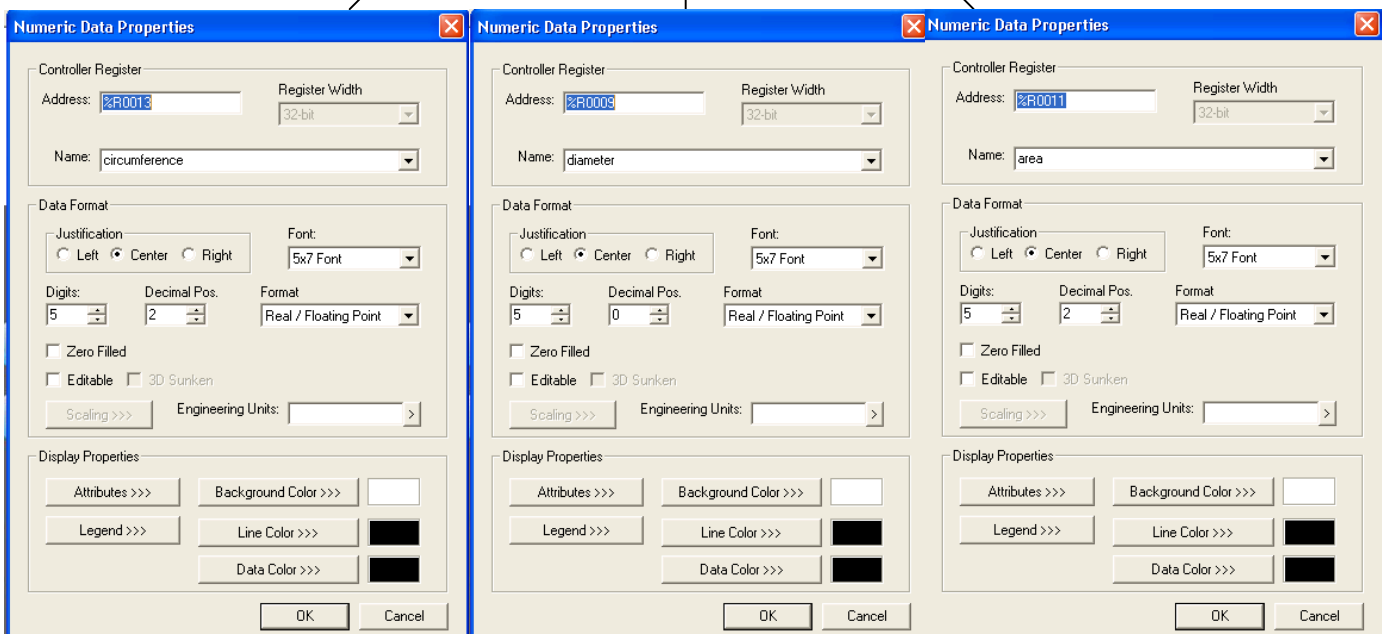
On the last screen we are going to have a screen jump back to the first screen so the user can cycle through. We are also going to display the circle calculations. The user will only be able to enter into one numeric box.



Set the screen up with the 4 numeric data functions. Set only the Radius to be editable and the others are non-editable.

They should all be set to Real / Floating point to match the data type used in the ladder logic.

Enter suitable legends and if you wish enter the Engineering units as you want.



Please refer to the program: “maths_tut.csp” 

IMO Precision Controls Limited
1000 North Circular Road
Staples Corner, London
NW2 7JP United Kingdom
Tel: +44 (0)20 8452 6444
Fax: +44 (0)20 8450 2274
Email: imo@imopc.com
Web: www.imopc.com

IMO Jeambrun Automation SAS
165 Rue Jean Jaures,
94700 Maisons Alfort
Paris, France
Tel: +33 (0)1 45 13 47 05
Fax: +33 (0)1 45 13 47 37
Email: info@imopc.fr
Web: www.imopc.fr

IMO Deutschland
Für weitere Einzelheiten zu IMO Agenten und Distributoren in Ihrer Nähe schreiben. Sie bitte ein E-mail an folgende Adresse:
imo@imopc.com

IMO Italia
Viale A. Volta 127/a
50131 Firenze, Italia
Tel: +39 800 783281
Fax: +39 800 783282
Email: info@imopc.it
Web: www.imopc.it

IMO Canada
Unit 10, Whitmore Road
Woodbridge, Ontario.
L4L 8G4 Cananda
Tel: +1 905 265 9844
Fax: +1 905 265 1749
Email: imocanada@imopc.com

PANEL
- Cam Switches
- Din Terminals
- Drives
- Enclosures
- Fieldbus remote I/O
- Isolators & Switch Fuses
- MCB & RCD
- Motor Circuit Breakers
- Motor Control Gear
- Panel Meters
- Relays
- Signal Conditioning
- Sockets
- Timers
- Transformers & Power Supplies

MACHINE
- Drives
- Intelligent Terminals/HMI
- Limit Switches
- Photoelectric Switches
- PLCs
- Proximity Switches
- Temperature Controls

PROCESS
- Data Acquisition & Control
- Drives
- Intelligent Terminals/HMI
- Limit Switches
- Photoelectric Switches
- Proximity Switches
- PLCs
- Signal Conditioning
- Temperature Controls

SAFETY
- Lightguards
- Safety Limit Switches
- Safety Relays

IMO JAGUAR DRIVES
- Jaguar VXM 0.37-500kW
- Jaguar VXSM 0.37-7.5kW
- Jaguar CUB 0.37-2.2kW

PCB EMECH
- Audible devices
- Chip-on-Board
- Device programmers
- LEDs & 7 seg. displays
- PCB Terminal blocks
- Relays - automotive
- Relays - power
- Relays - signal
- Switches

All IMO products are tried, tested and approved to relevant international quality standards

nqa. ISO 9001 Registered UKAS QUALITY MANAGEMENT

3 YEAR GUARANTEE

5 YEAR GUARANTEE