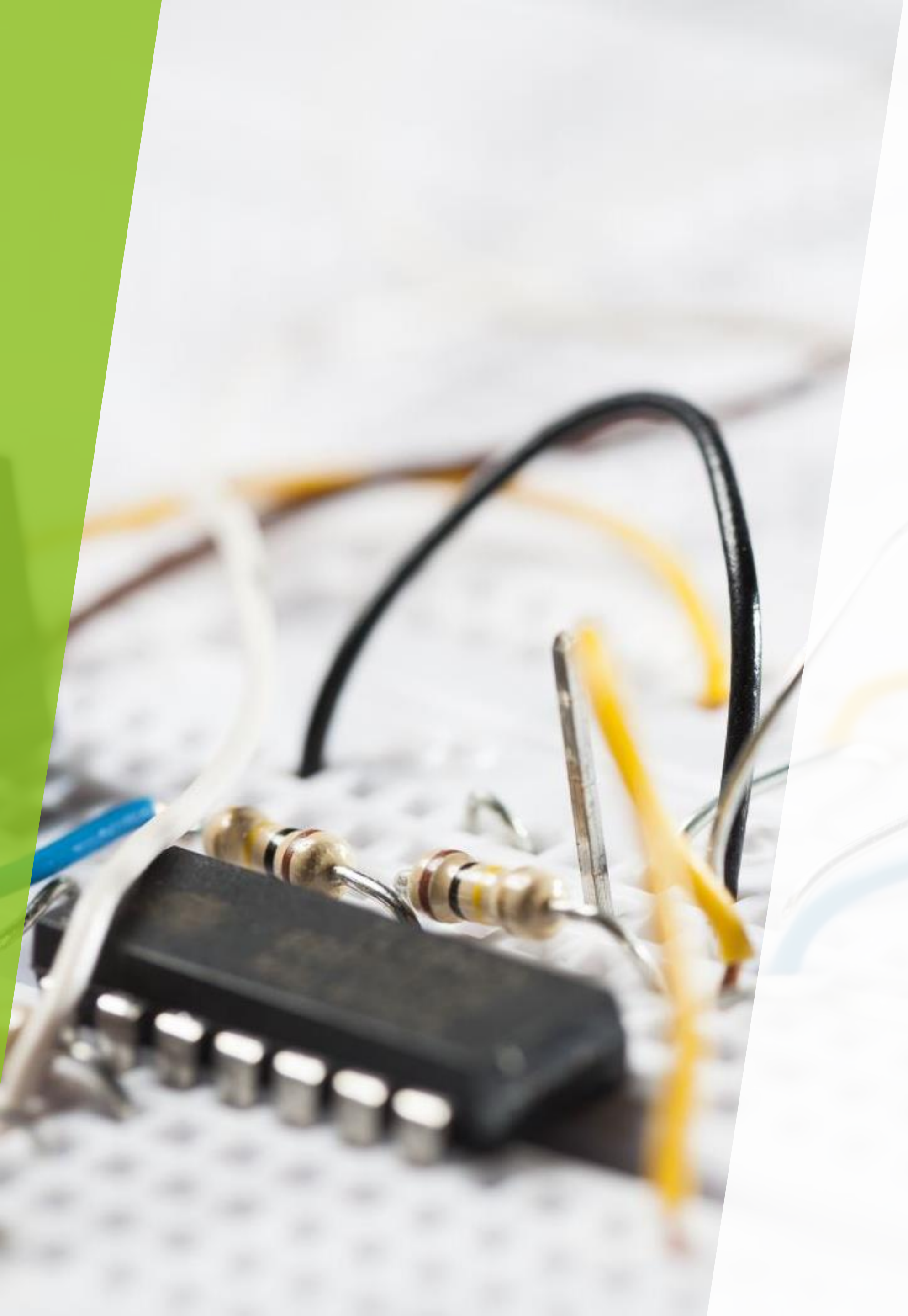
WELCOME TO

ARDUINO CURIOSITY

CRAFTING ELECTRONICAL SENSORS

IOSITY
AL SENSORS



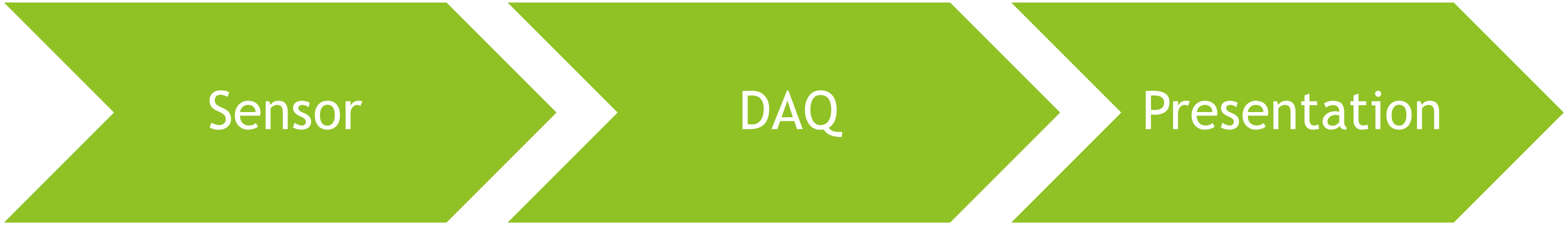
Sensores Data Transmissio et Machina Humana Interface

Engr. Voltaire B. Dupo, ECE

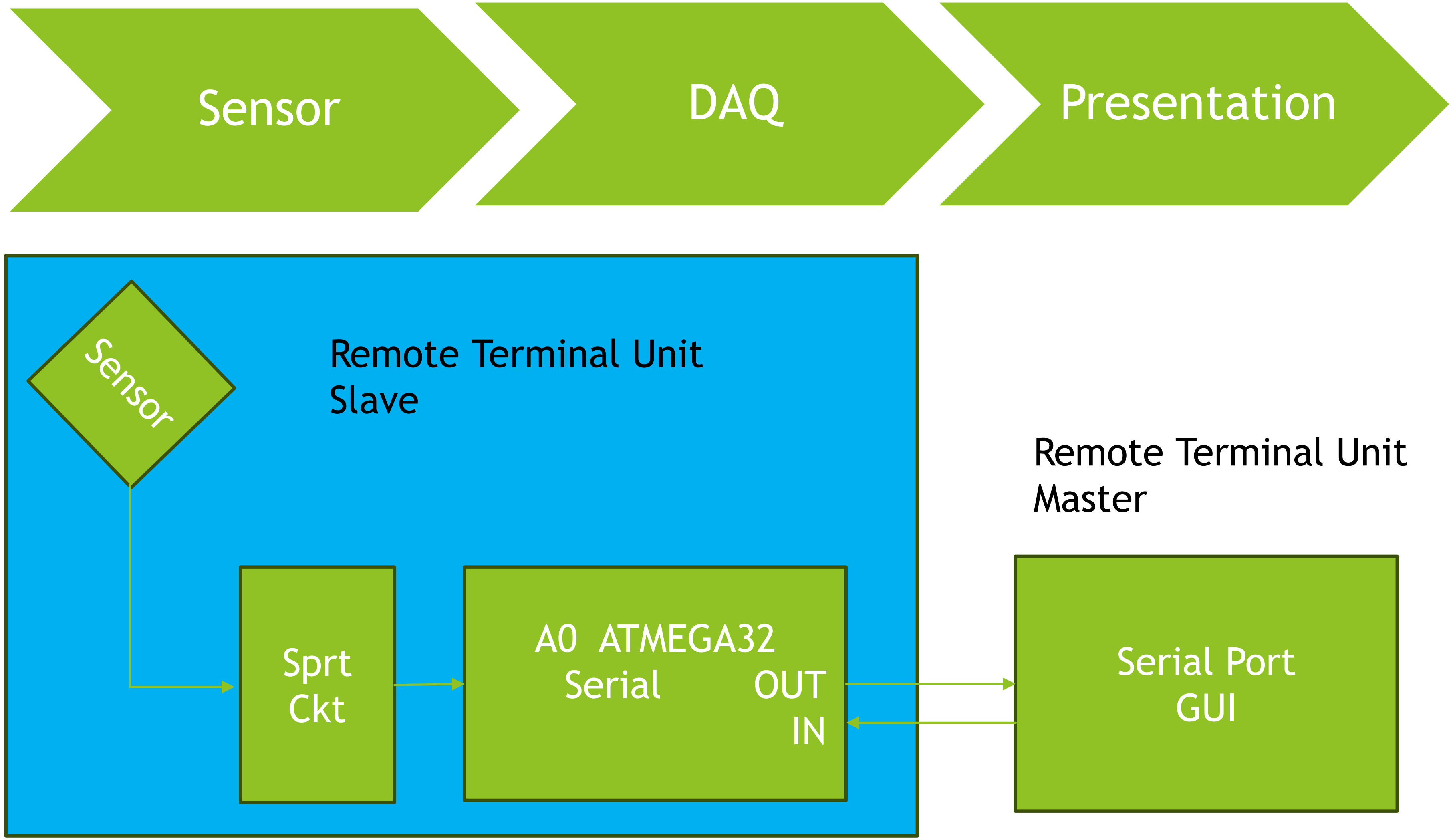
Topic List

- Sensors Primer
- Modbus as Data Transmission Protocol
- Human Machine Interface

Overview

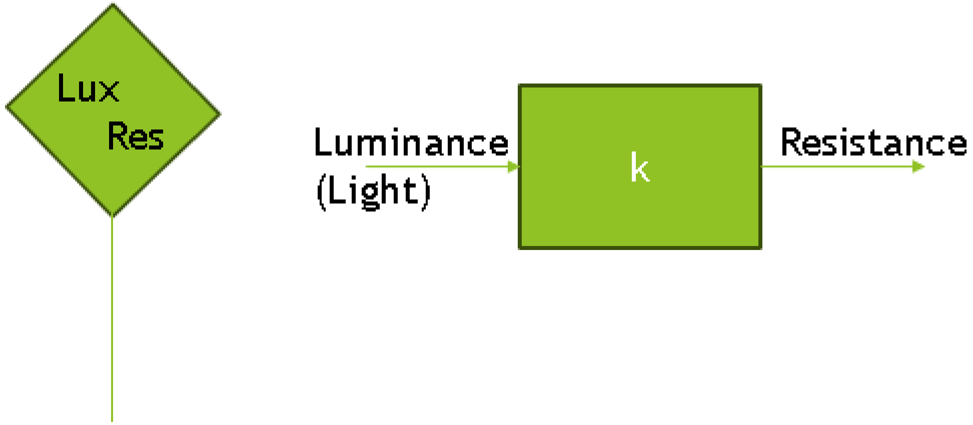


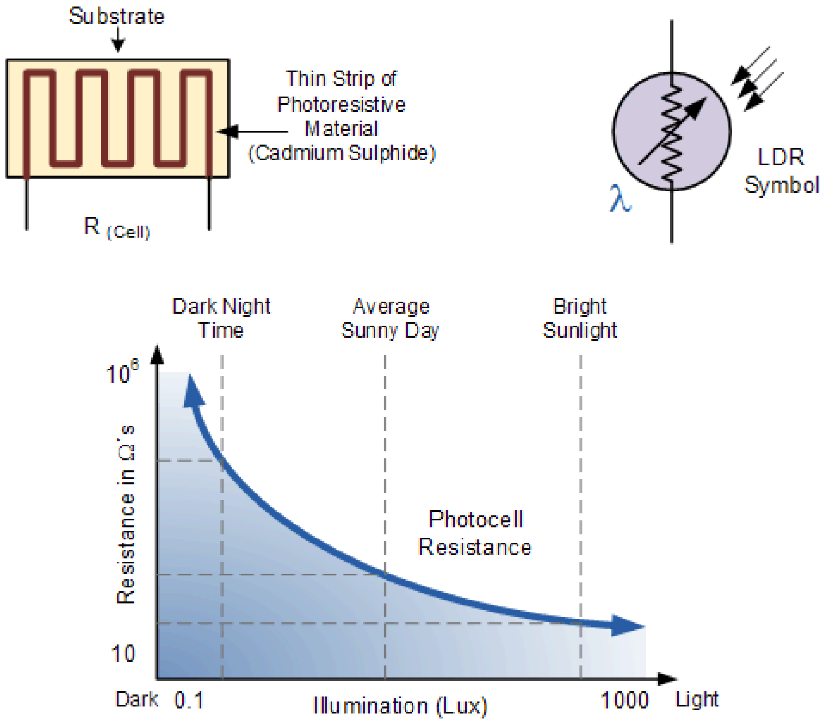
Overview


Sensor
DAQ
Presentation
Sensor
Remote Terminal Unit
Slave
Sprt
Ckt
A0 ATMEGA32
Serial
OUT
IN
Remote Terminal Unit
Master
Serial Port
GUI

LDR Sensor

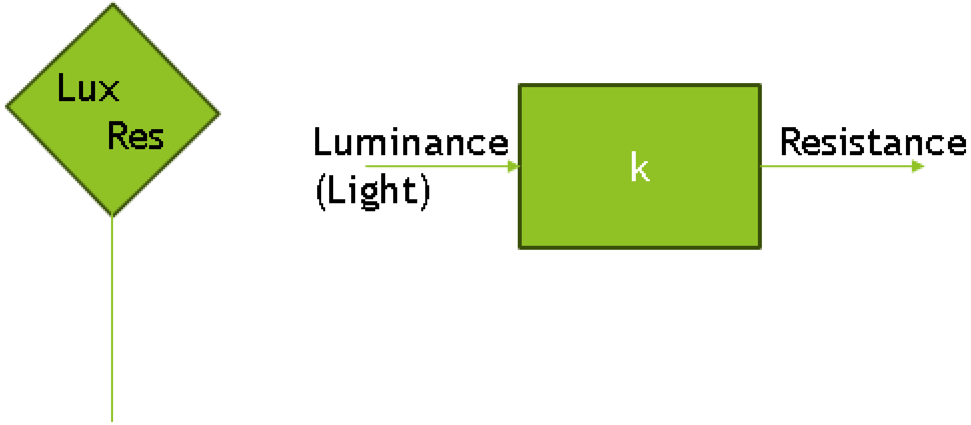
- A device that converts a light intensity into resistance invented by John N Shive and supported by Bell Laboratories in 1949

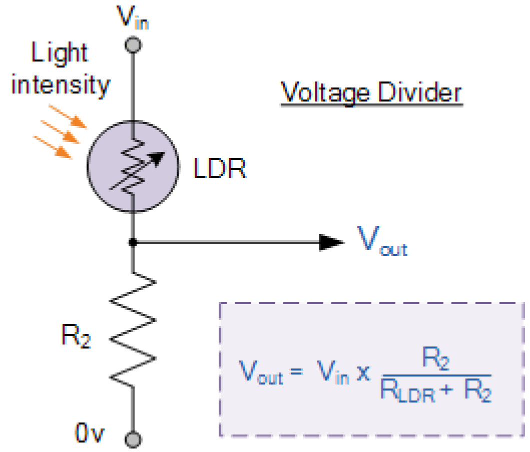




LDR Sensor

- A device that converts a light intensity into resistance invented by John N Shive and supported by Bell Laboratories in 1949

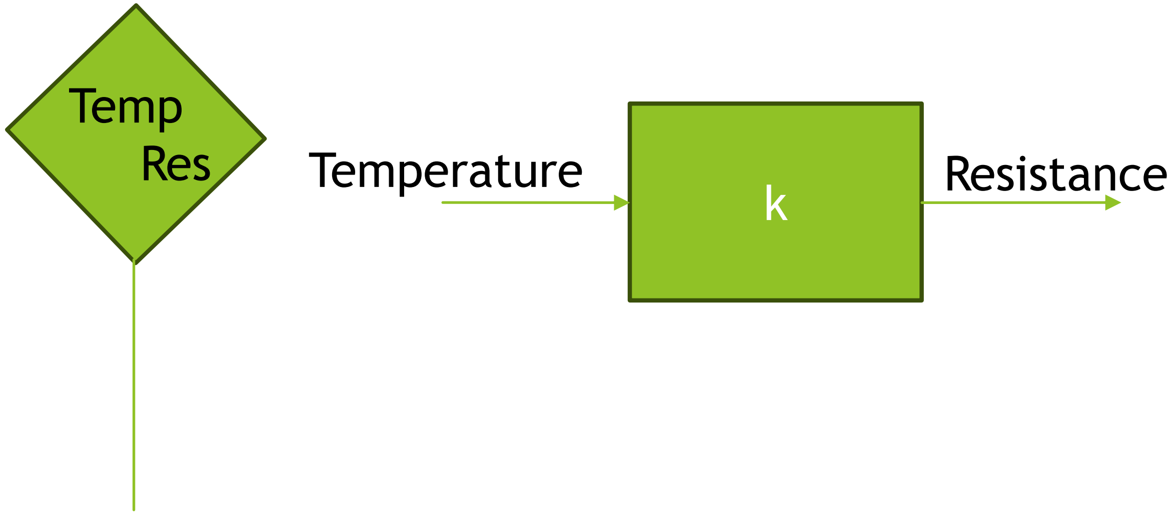




$$V_{out} = V_{in} \times \frac{R_2}{R_{LDR} + R_2}$$

Thermistor Sensor

- A device that converts a Temperature into Resistance invented by Michael Faraday in 1833. This was improved by Samuel Ruben in 1930 into the commercially viable thermistor we all use

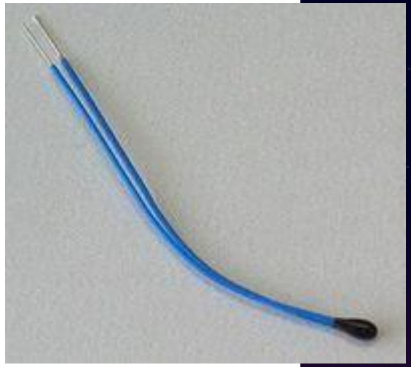


Temp
Res

Temperature

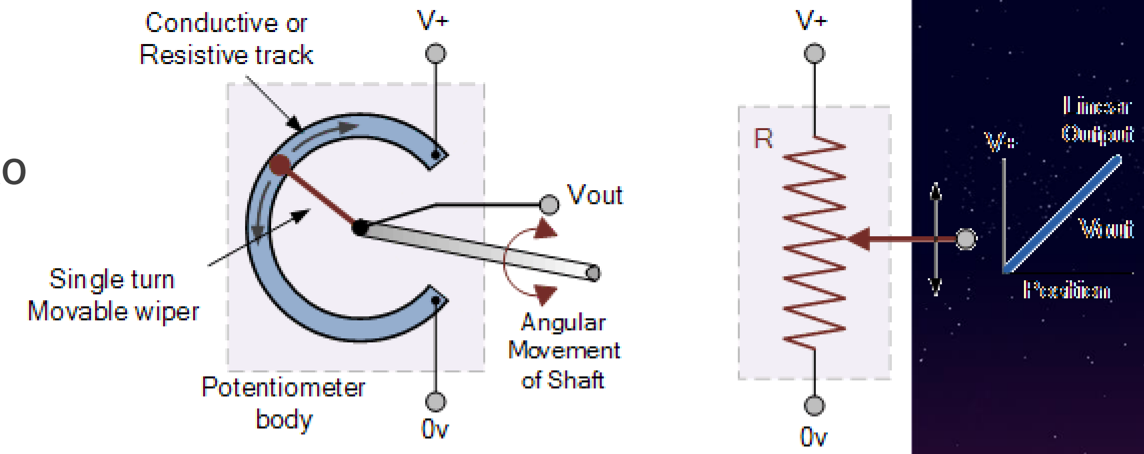
k

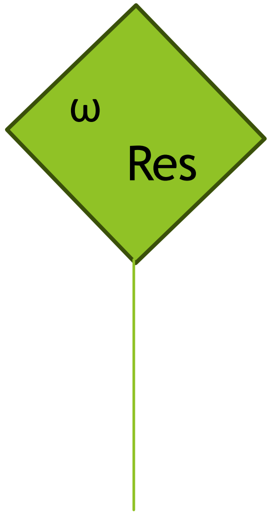
Resistance

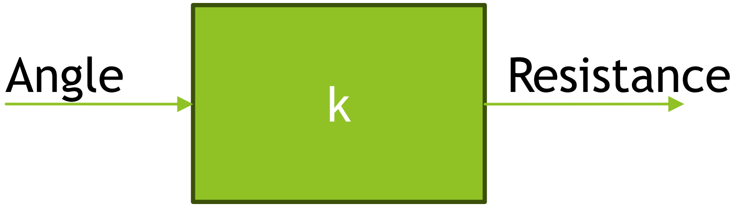


Potentiometer as Angular Sensor

- A device that converts angular displacement into Resistance

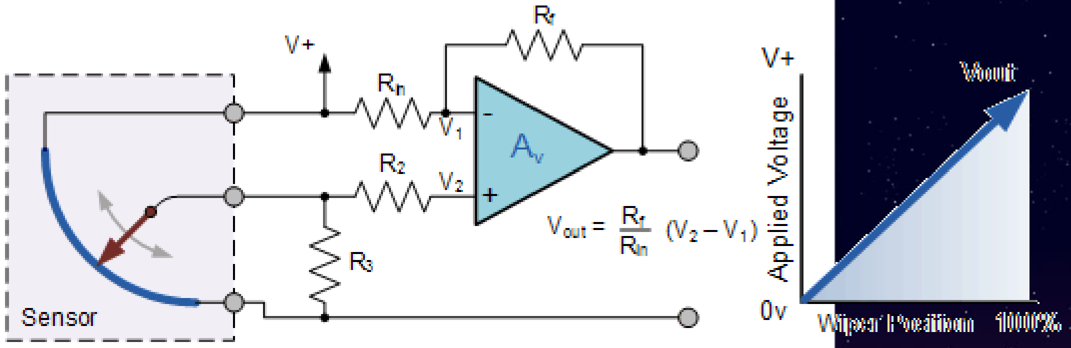

Conductive or Resistive track
V+
Vout
Single turn Movable wiper
Angular Movement of Shaft
Potentiometer body
0v
V+
R
Vs
Linear Output
Vout
Position
0v

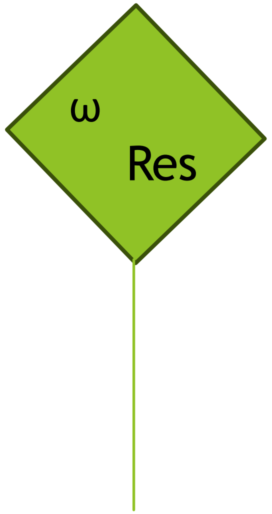


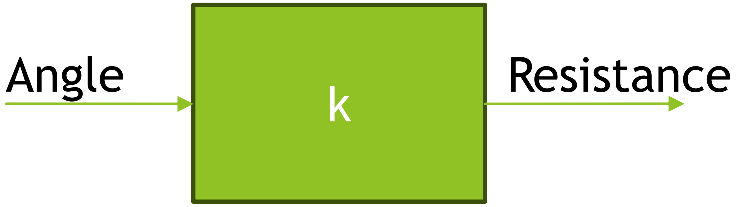


Potentiometer as Angular Sensor

- A device that converts angular displacement into Voltage

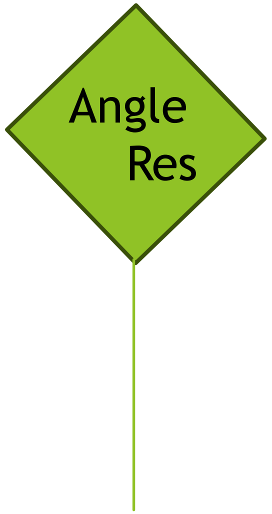


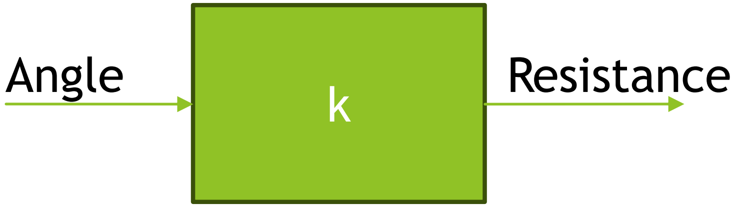


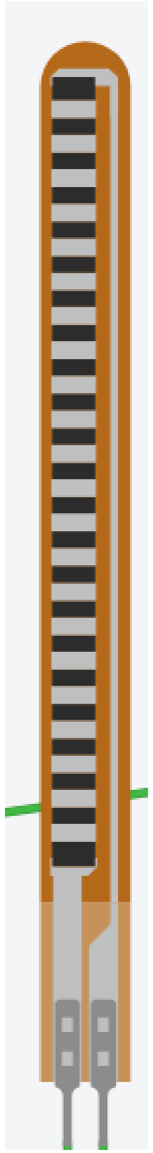


Flex Sensor

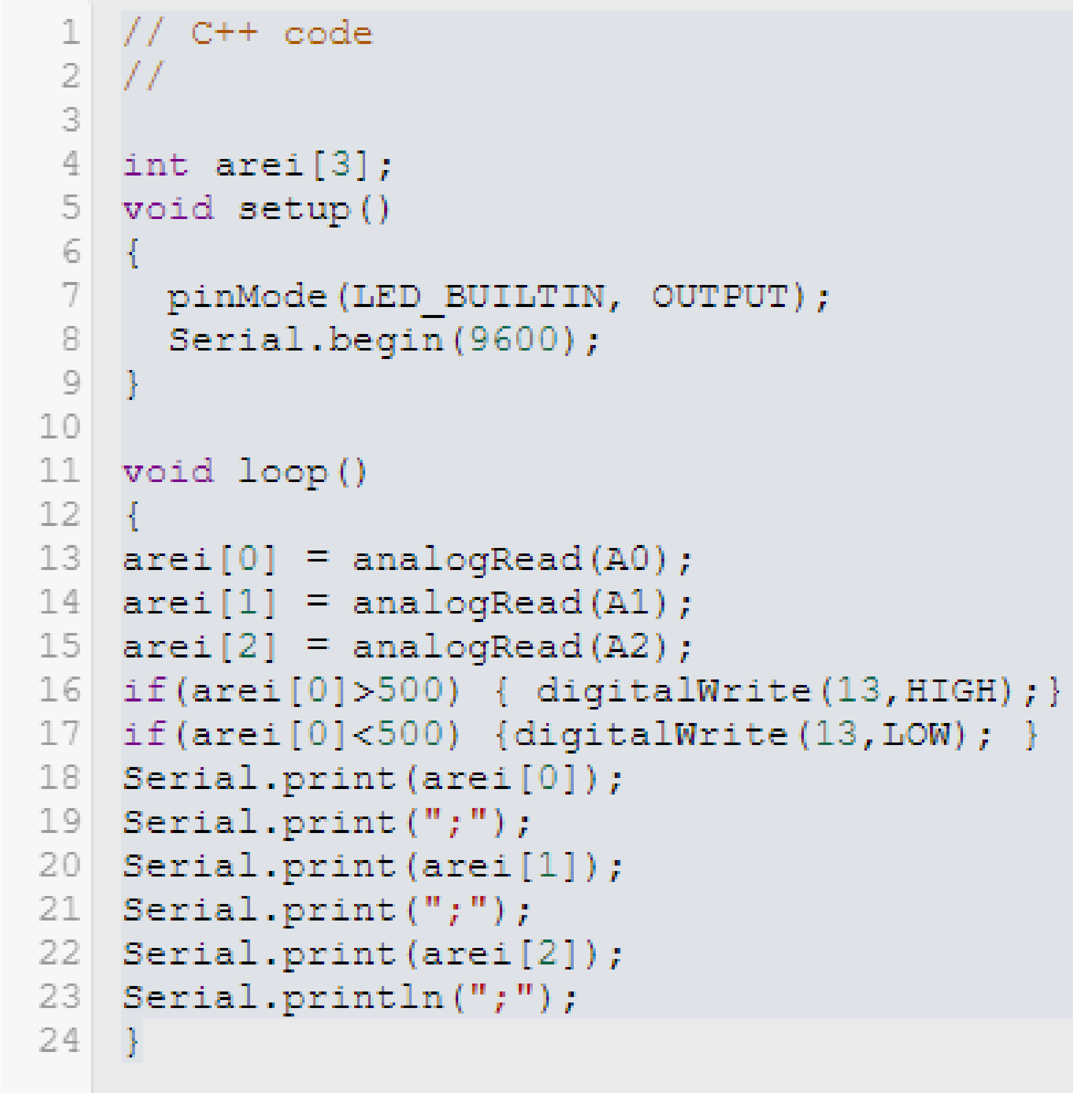
- A device that converts bending angle to Electrical Resistance



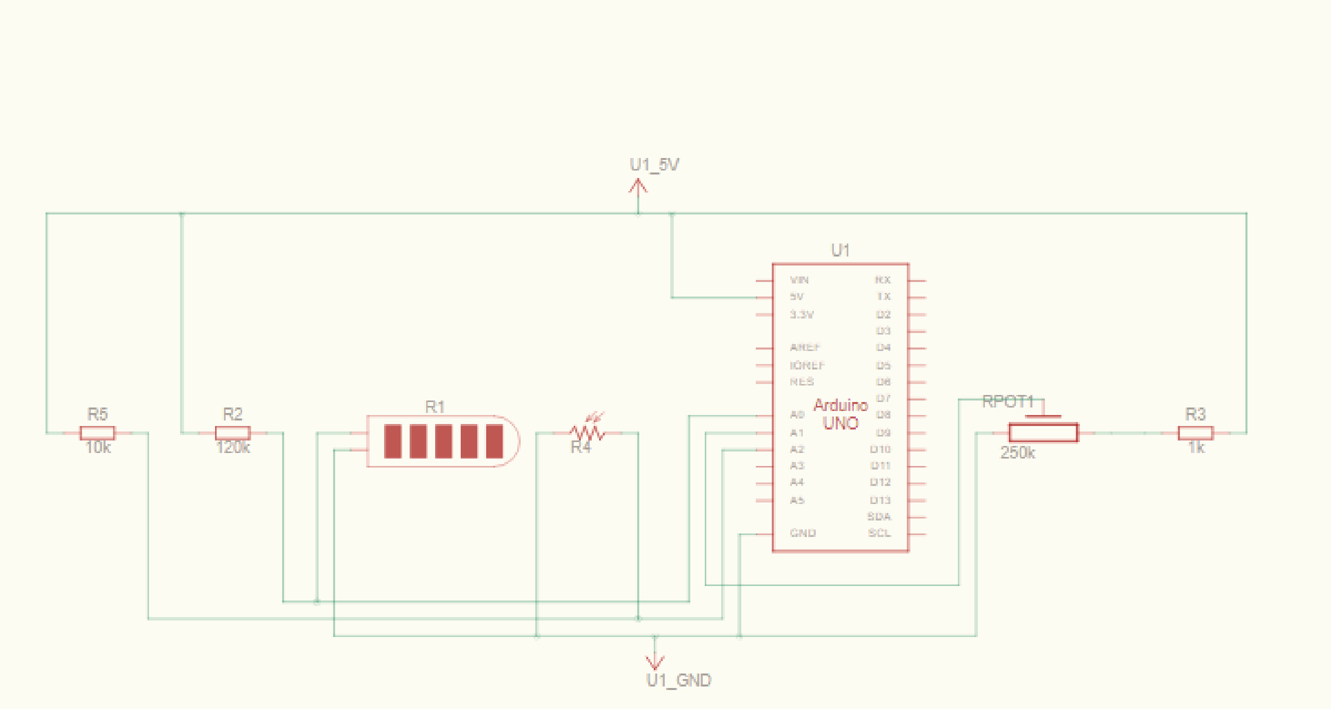




Code

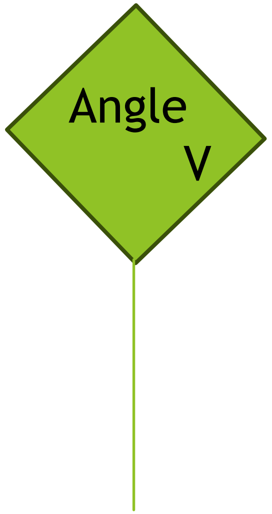


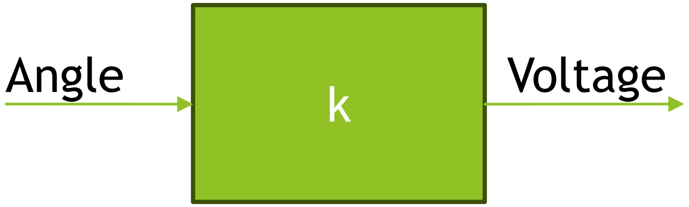
```
// C++ code
//

int arei[3];
void setup()
{
  pinMode(LED_BUILTIN, OUTPUT);
  Serial.begin(9600);
}

void loop()
{
arei[0] = analogRead(A0);
arei[1] = analogRead(A1);
arei[2] = analogRead(A2);
if(arei[0]>500) { digitalWrite(13,HIGH);}
if(arei[0]<500) {digitalWrite(13,LOW); }
Serial.print(arei[0]);
Serial.print(";");
Serial.print(arei[1]);
Serial.print(";");
Serial.print(arei[2]);
Serial.println(";");
}
```



Tilt Sensor

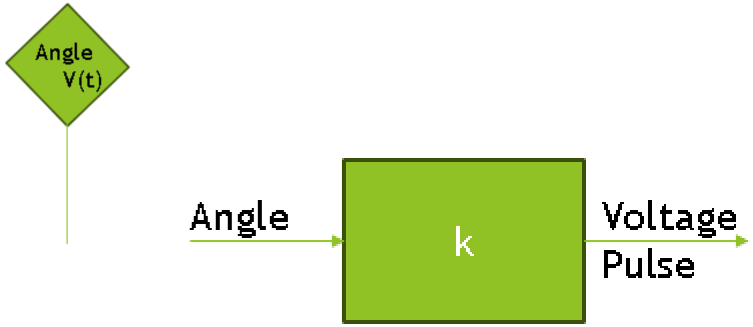
- A device that converts a physical phenomenon into Electrical Signals





Rotational Encoder as a Sensor

- A device that converts angle displacement into Electrical Pulses

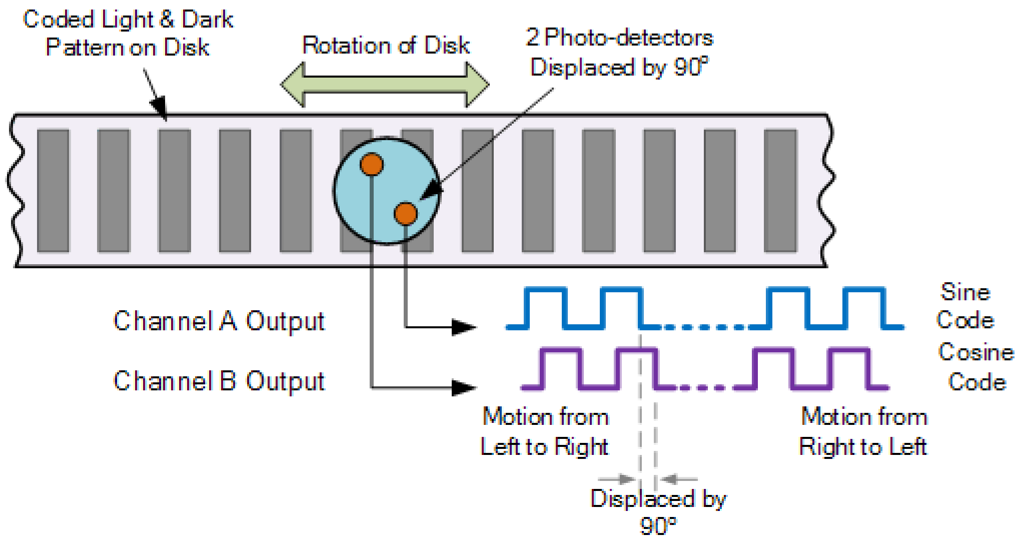


Angle V(t)

Angle

k

Voltage Pulse



Coded Light & Dark Pattern on Disk

Rotation of Disk

2 Photo-detectors Displaced by 90°

Channel A Output

Channel B Output

Sine Code

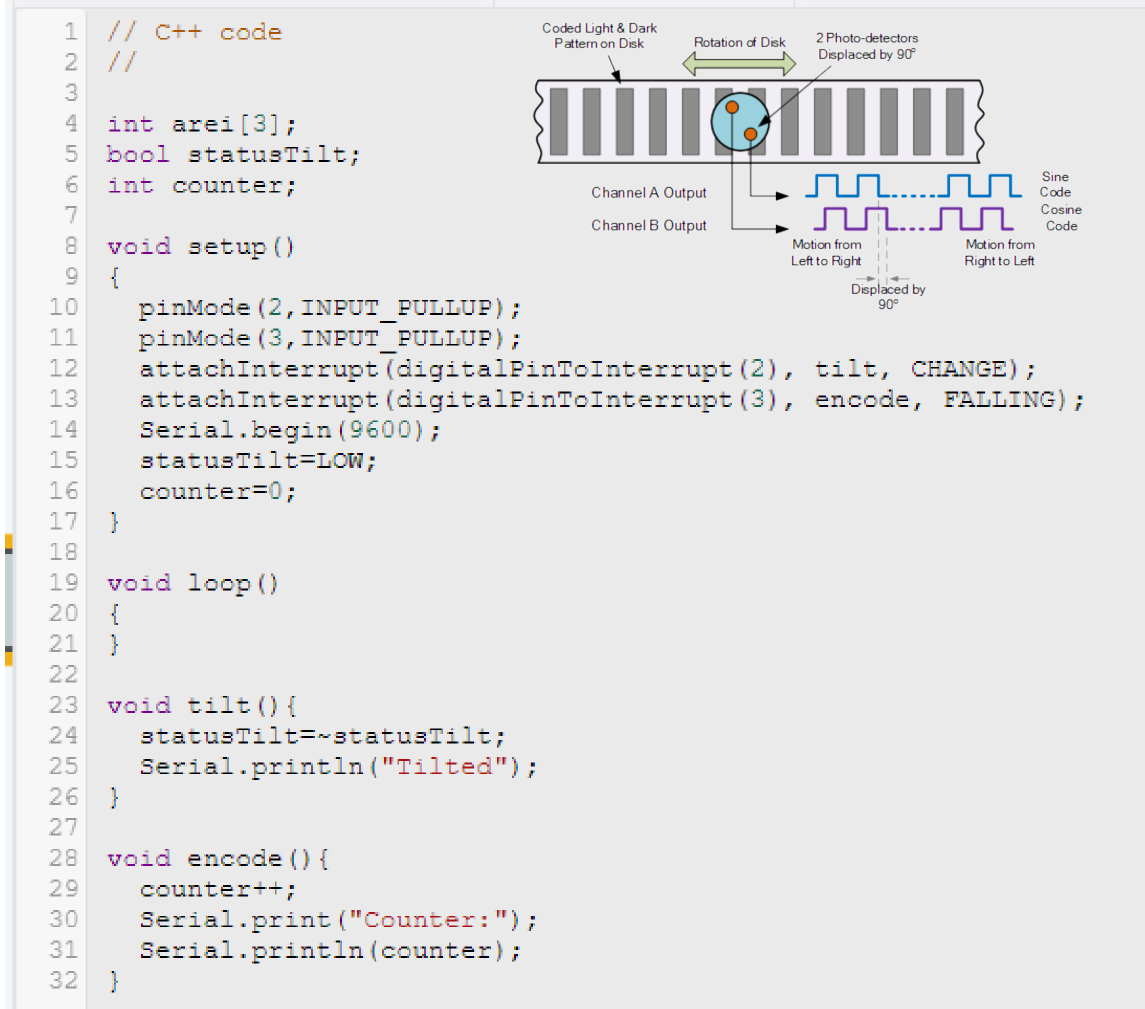
Cosine Code

Motion from Left to Right

Motion from Right to Left

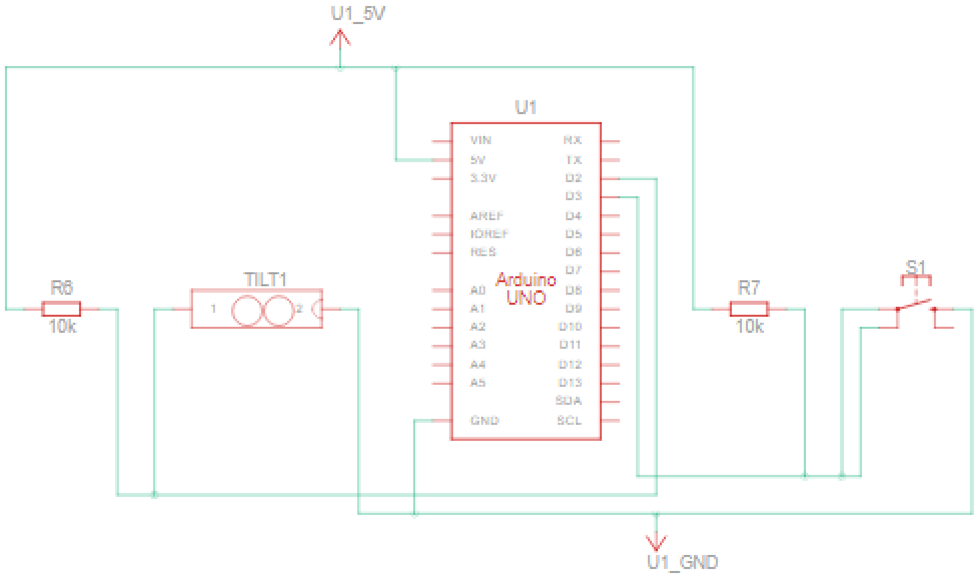
Displaced by 90°

Code Example



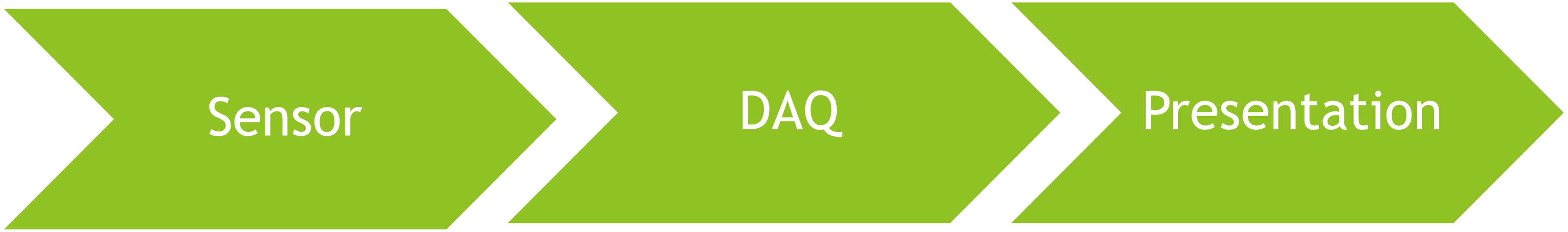
```
// C++ code
//

int arei[3];
bool statusTilt;
int counter;

void setup()
{
  pinMode(2,INPUT_PULLUP);
  pinMode(3,INPUT_PULLUP);
  attachInterrupt(digitalPinToInterrupt(2), tilt, CHANGE);
  attachInterrupt(digitalPinToInterrupt(3), encode, FALLING);
  Serial.begin(9600);
  statusTilt=LOW;
  counter=0;
}

void loop()
{
}

void tilt(){
  statusTilt=~statusTilt;
  Serial.println("Tilted");
}

void encode(){
  counter++;
  Serial.print("Counter:");
  Serial.println(counter);
}
```

Coded Light & Dark Pattern on Disk
Rotation of Disk
2 Photo-detectors Displaced by 90°
Channel A Output
Channel B Output
Sine Code
Cosine Code
Motion from Left to Right
Motion from Right to Left
Displaced by 90°



Modbus Standard

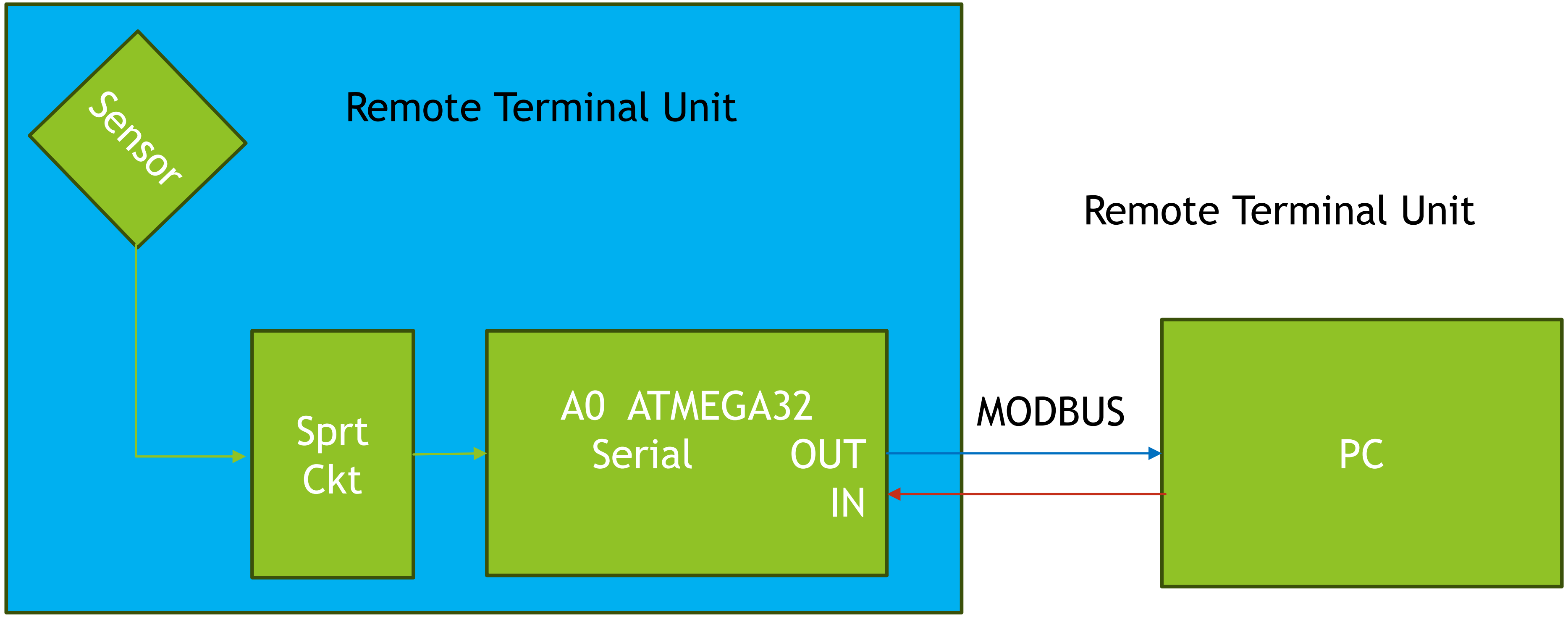
Overview



Sensor

DAQ

Presentation



Remote Terminal Unit

Remote Terminal Unit

History of Modbus

- Modicon 1968
  - Programmable Logic Controller thru the direction of Dick Morley

- Modicon 1979
  - Modbus Communication Protocol
- Modicon was sold to Gould Electronics in 1977 and resold to Schnieder Electric in 2014 when Gould Electronics was dissolved by JX Holdings

- Modbus is hailed as the de facto standard for PLC Communication commonly done between a Remote Terminal Units and a Central Monitoring and Processing Terminal
- Modbus is managed now by the Modbus Organization (Modbus.org)

- *Drury, Bill (2009). Control Techniques Drives and Controls Handbook (PDF) (2nd ed.). Institution of Engineering and Technology. pp. 508–.*

What is Modbus

- Modbus is a memory management Standard and a Transmission System

As a Memory Management Standard

- PLC Storage System
  - I/O Pins
  - Memory
- I/O Pins
  - Coils
    - 1 or 0
- Memory
  - Register
    - 2^^{32} or 2^^{16} Length Containers

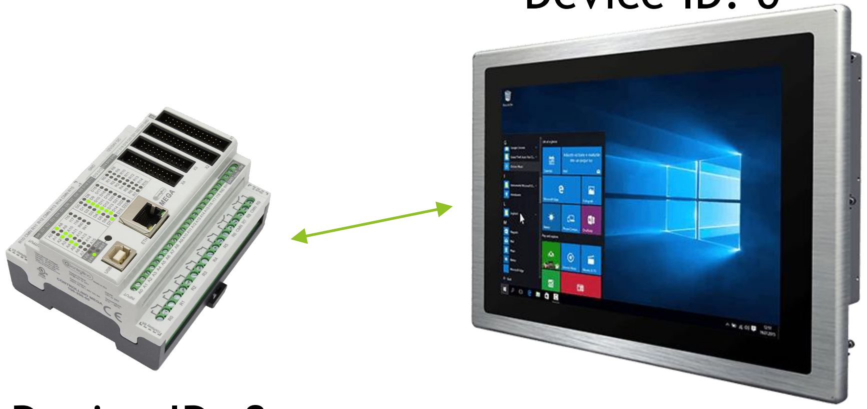
What is Modbus

- Modbus is a memory management Standard and a Transmission System

Type	Function Code	Object Type
Input Pin	2	Discrete Input
Output Pin	1	Coil Status
Input Digitized Analog	4	Input Register
Internal Register	3	Holding Register

What is Modbus

- Modbus is a memory management Standard and a Transmission System



Device ID: 0

Device ID: 8

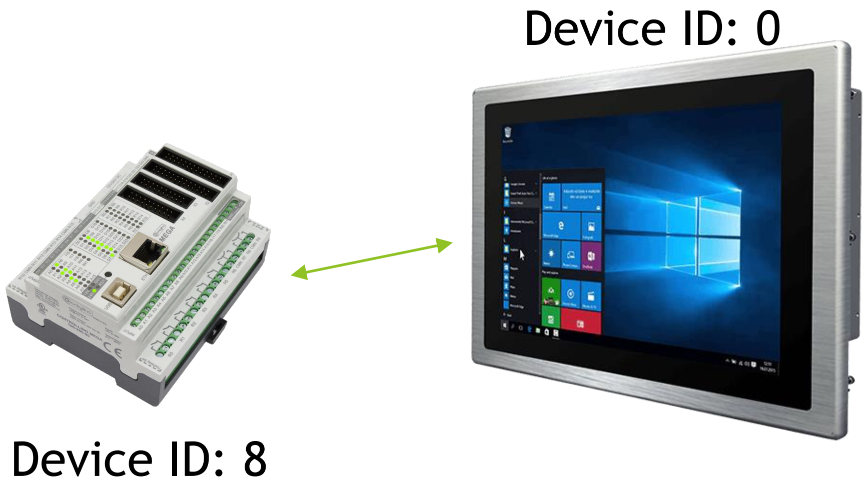
As a Transmission Standard

- Device ID
  - Slave
    - ID numbers 1 – 127
  - Master
    - ID number 0
- Function Code to Request/Transmit

Type	Function Code	Object Type
Input Pin	2	Discrete Input
Output Pin	1	Coil Status
Input Digitized Analog	4	Input Register
Internal Register	3	Holding Register

What is Modbus

- Modbus is a memory management Standard and a Transmission System



As a Transmission Standard Mobus Org stated in Feb 2020 that all Language be changed from old System RTU / new System TCP/IP to avoid confusion

- Device ID
  - Server
    - ID numbers 1 – 127
  - Client
    - ID number 0
- Function Code to Request/Transmit

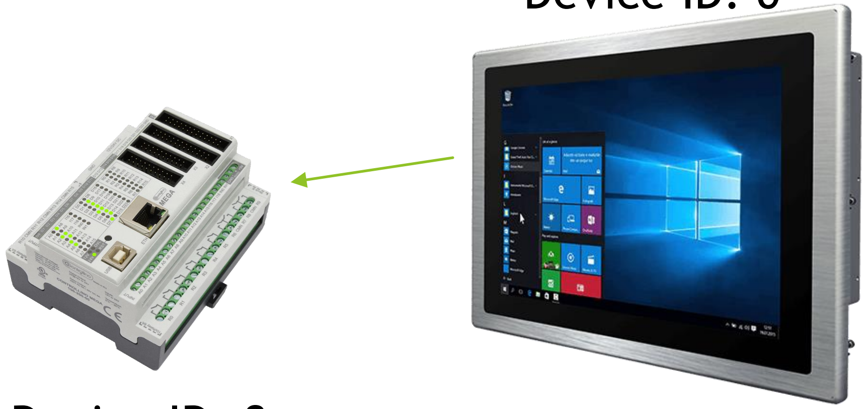
Type	Function Code	Object Type
Input Pin	2	Discrete Input
Output Pin	1	Coil Status
Input Digitized Analog	4	Input Register
Internal Register	3	Holding Register

What is Modbus

- Modbus is a memory management Standard and a Transmission System

8:0	2	FFFF	CRC

Device ID: 0



Device ID: 8

Address	Func. C.	Data	CRC

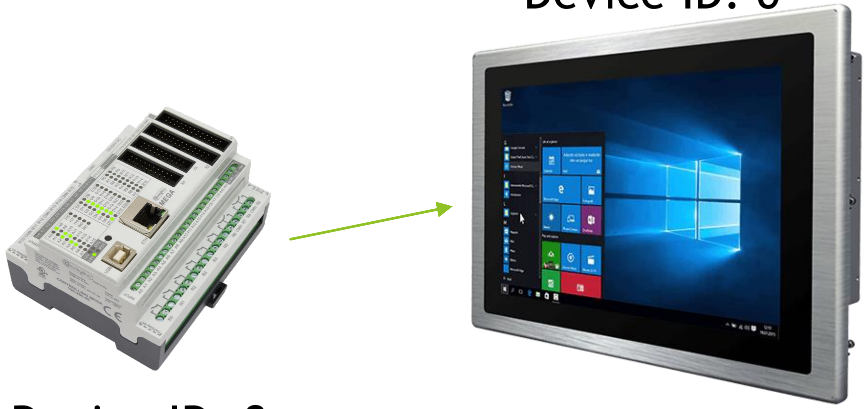
Type	Function Code	Object Type
Input Pin	2	Read Discrete Input
Output Pin	1	Write-Read Coil Status
Input Digitized Analog	4	Read Input Register
Internal Register	3	Write-Read Holding Register

What is Modbus

- Modbus is a memory management Standard and a Transmission System

Address	Func. C.	Data	CRC

0:8	2	1	CRC



Type	Function Code	Object Type
Input Pin	2	Read Discrete Input
Output Pin	1	Write-Read Coil Status
Input Digitized Analog	4	Read Input Register
Internal Register	3	Write-Read Holding Register

What is Modbus

- Modbus is a memory management Standard and a Transmission System

難しい

Address	Func. C.	Data	CRC

0:8	2	1	CRC

Device ID: 0

Device ID: 8

Type	Function Code	Object Type
Input Pin	2	Read Discrete Input
Output Pin	1	Write-Read Coil Status
Input Digitized Analog	4	Read Input Register
Internal Register	3	Write-Read Holding Register

Translating Modbus to Arduino

Modbus RTU Library

- https://github.com/smarmengol/Modbus-Master-Slave-for-Arduino/blob/master/ModbusRtu.h

SCAN ME

Elements

- Library Declaration
  - #include <ModbusRtu.h>
- Container Declaration
  - uint16_t <name>[n] = {values list};
- Initialization
  - Modbus slave(<id>,<port>,<txpin>);
- Activation
  - Inside Setup
    - slave.begin(baudrate);
- Updating Values
  - Inside Loop
    - slave.poll(<arrayname>,<#elements>);

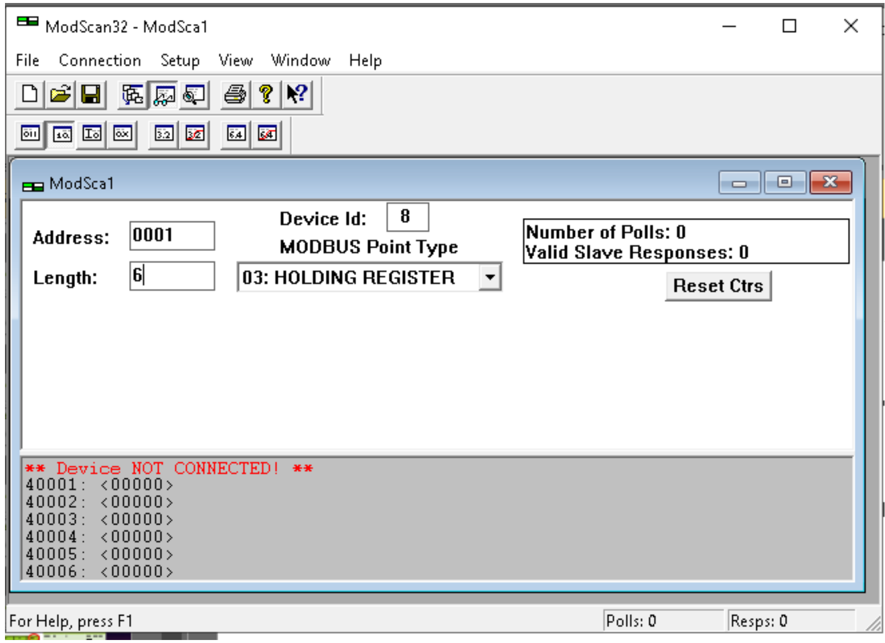
Sample Code Modbus 1:

```
#include <ModbusRtu.h>


// registers in the slave Do not remove this unless
you want the Modbus to not work

uint16_t au16data[6] = {103, 0, 0, 0, 0, 0};


int trigger=4;

int commandline;

int i,k;


Modbus slave(8,0,0); // this is slave @1 and RS-232
or USB-FTDI


void setup() {

  slave.begin( 9600 ); // baud-rate at 9600

  pinMode(13, OUTPUT);

  digitalWrite(13,HIGH);

}
```

```
void loop()
{
slave.poll( au16data, 6 );
commandline=au16data[0];
if(commandline==103 && trigger==6) {
trigger=4; au16data[5]=4;}
if(commandline!=103 && trigger==4)
{switch(commandline)
{
case 0: k=0;       break;
case 11: k=11;       break;
case 22: k=22;       break;
case 33: k=33;      break;
case 44: k=44;    break;
case 55: k=55;     break;
case 66: k=66;       break;
case 77: k=77;    break;
case 88: k=88;       break;
case 99: k=99;      break;
case 100: k=100;      break;
case 111: k=111;      break;
case 112: k=112;     break;
case 69: k=69;       break;
case 79: k=79;     break;
case 89: k=89;       break;
case 97: k=97;       break;
case 59: k=59;       break;
case 49: k=49;      break;
case 39: k=39;      break;
case 29: k=29;      break;
default: k=8;      break;
}
```

```
trigger=6;        // resets trigger
commandline=134   // resets trigger
au16data[4]=6;    // assigns 4 to au16data[4]
au16data[1]=k;        // assigns the value of k to au16data[1]
}

}
```

Terminal Communication Program 1
ModScan



Address: 0001

Device Id: 8

MODBUS Point Type

Length: 6

03: HOLDING REGISTER

Number of Polls: 0
Valid Slave Responses: 0

```
** Device NOT CONNECTED! **
40001: <00000>
40002: <00000>
40003: <00000>
40004: <00000>
40005: <00000>
40006: <00000>
```

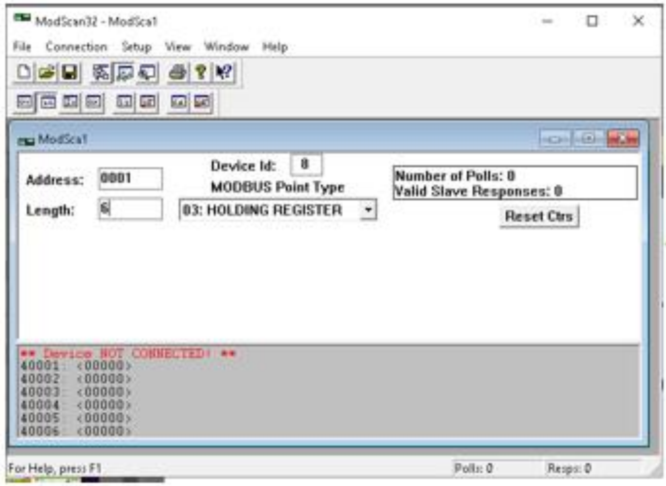
For Help, press F1 | Polls: 0 | Resps: 0

USB

Device: 0
Client

Device: 8
Server

ModScan Download Link



Address: 0001

Length: 6

Device Id: 8

MODBUS Point Type

03: HOLDING REGISTER

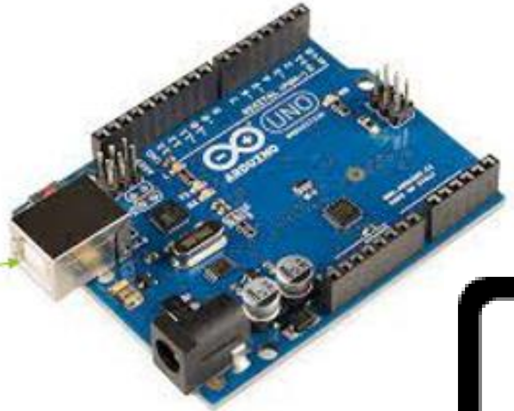
Number of Polls: 0
Valid Slave Responses: 0

```
** Device NOT CONNECTED! **
40001: <00000>
40002: <00000>
40003: <00000>
40004: <00000>
40005: <00000>
40006: <00000>
```

For Help, press F1 Polls: 0 Resps: 0

Device: 0
Client

USB



Device: 8
Server

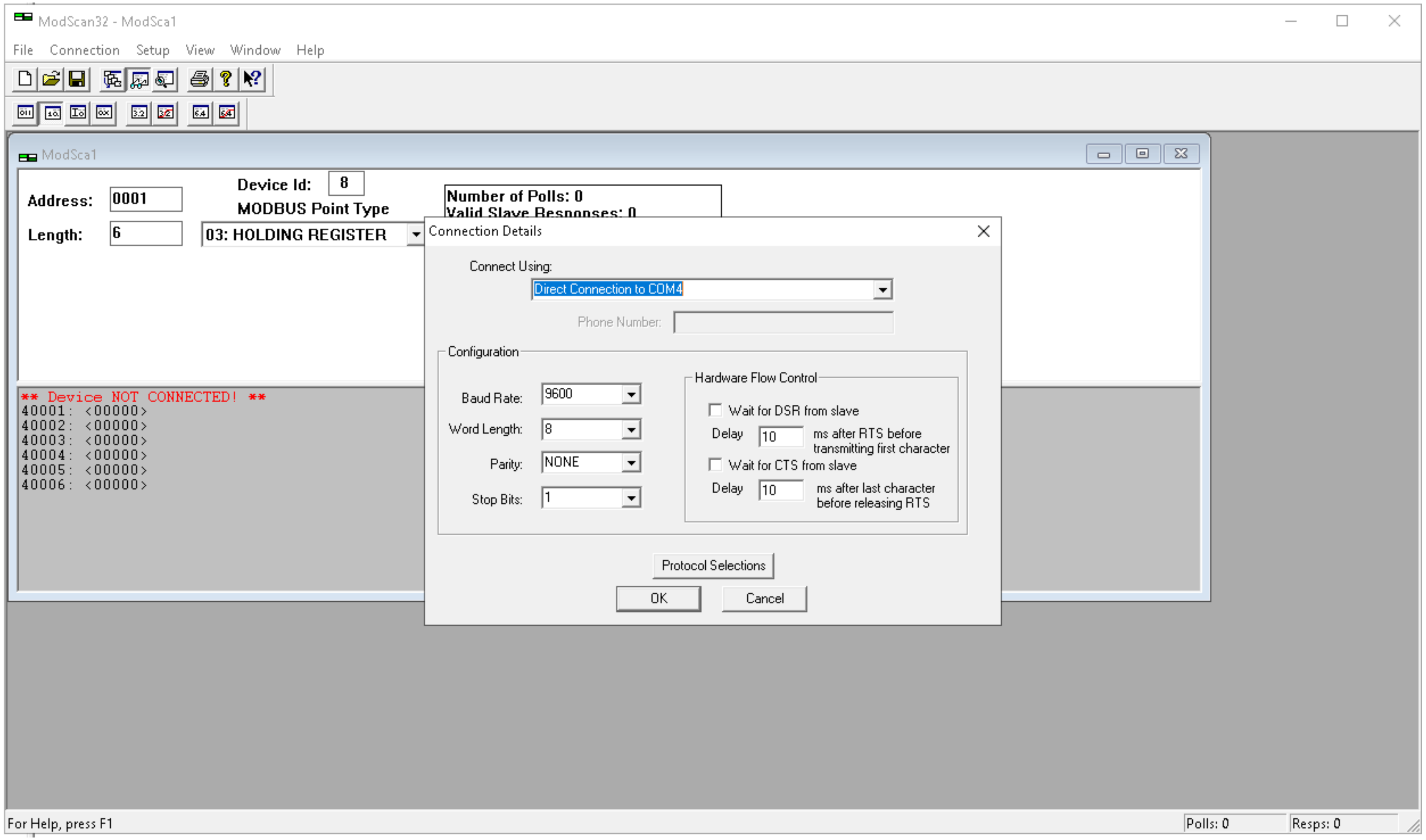
- https://drive.google.com/drive/folders/1tOXJwciSB7_lcpMgATGnggSyML6vbXKo?usp=sharing

SCAN ME

Sample Code Modbus 2:

```
#include <ModbusRtu.h>
// registers in the slave Do not remove this unless you want the Modbus to not work
uint16_t au16data[6] = {103, 0, 0, 0, 0, 0};
int cmd;
Modbus slave(8,0,0); // this is slave @1 and RS-232 or USB-FTDI
void setup() {
  slave.begin( 9600 ); // baud-rate at 9600
  pinMode(13, OUTPUT);
  digitalWrite(13,HIGH);
}
void loop() {
slave.poll( au16data, 6 );
cmd = au16data[0];
if(cmd==64) { digitalWrite(13,LOW);  }
if(cmd!=64) { digitalWrite(13,HIGH); }
au16data[2] = analogRead(A0);
}
```

Setup of Modscan32 for Server Interface



Address: 0001

Length: 6

Device Id: 8

MODBUS Point Type: 03: HOLDING REGISTER

Number of Polls: 0
Valid Slave Responses: 0

```
** Device NOT CONNECTED! **
40001: <00000>
40002: <00000>
40003: <00000>
40004: <00000>
40005: <00000>
40006: <00000>
```

Connect Using: Direct Connection to COM4

Phone Number:

Configuration

Baud Rate: 9600

Word Length: 8

Parity: NONE

Stop Bits: 1

Hardware Flow Control

□ Wait for DSR from slave

Delay 10 ms after RTS before transmitting first character

□ Wait for CTS from slave

Delay 10 ms after last character before releasing RTS

For Help, press F1

Polls: 0

Resps: 0

Human Machine Interface

Human Machine Interface

- Also known as User Interface (UI)
- Means for people to interact with Machines without knowledge of programming or inner workings
  - Abstractions are hidden

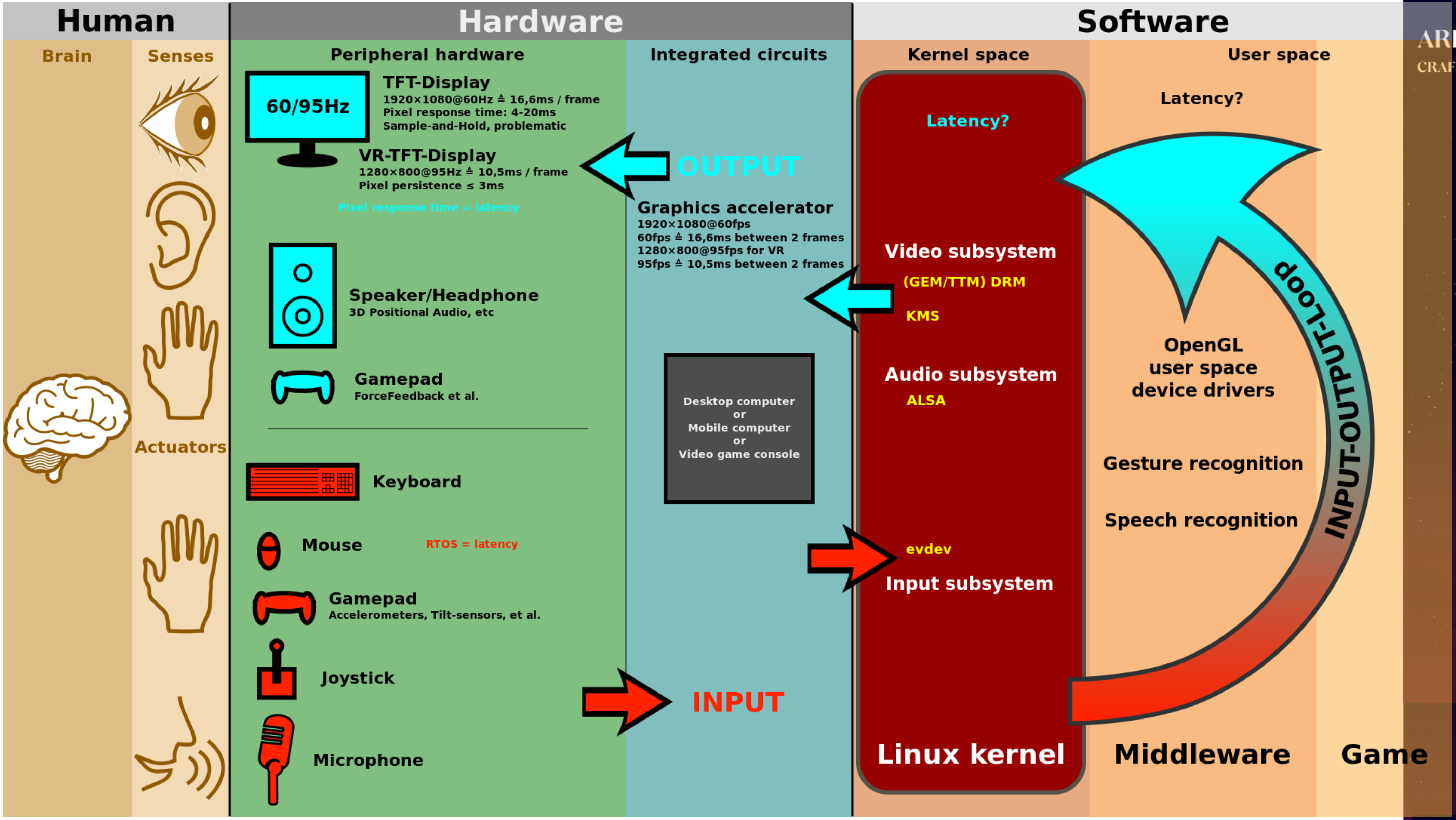
Human Machine Interface

Batch Interface
1945

Command Line Interface
1968

Text User Interface
1985

Graphics User Interface
1968 (SRI)



Human

Brain

Senses

Actuators

Hardware

Peripheral hardware

60/95Hz

TFT-Display
1920×1080@60Hz ≙ 16,6ms / frame
Pixel response time: 4-20ms
Sample-and-Hold, problematic

VR-TFT-Display
1280×800@95Hz ≙ 10,5ms / frame
Pixel persistence ≤ 3ms

Pixel response time = latency

Speaker/Headphone
3D Positional Audio, etc

Gamepad
ForceFeedback et al.

Keyboard

Mouse

RTOS = latency

Gamepad
Accelerometers, Tilt-sensors, et al.

Joystick

Microphone

Integrated circuits

OUTPUT

Graphics accelerator
1920×1080@60fps
60fps ≙ 16,6ms between 2 frames
1280×800@95fps for VR
95fps ≙ 10,5ms between 2 frames

Desktop computer
or
Mobile computer
or
Video game console

INPUT

Software

Kernel space

Latency?

Video subsystem
(GEM/TTM) DRM
KMS

Audio subsystem
ALSA

evdev
Input subsystem

Linux kernel

User space

Latency?

OpenGL
user space
device drivers

Gesture recognition

Speech recognition

INPUT-OUTPUT-Loop

Middleware

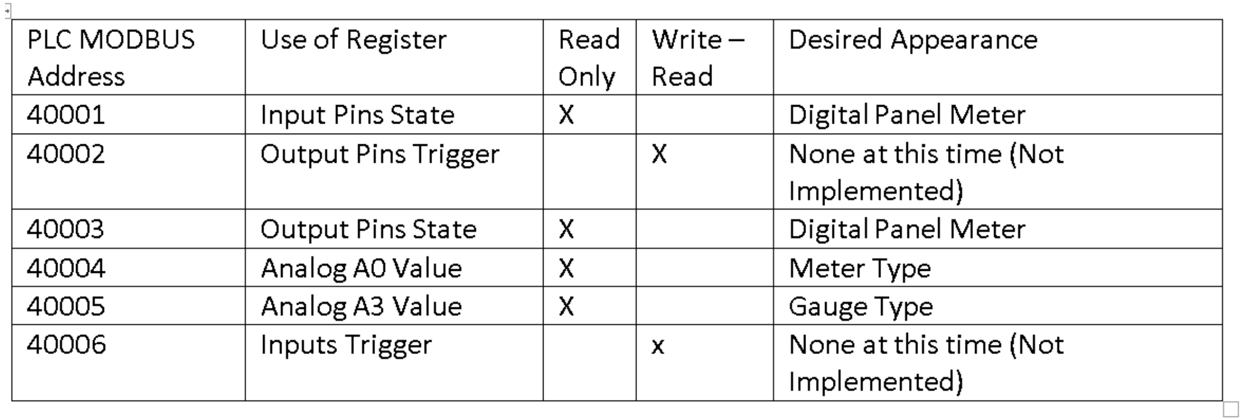
Game

Rules on Creating HMI

- Before you start with any HMI Creation work make sure to figure out and state what the parameters you want to see, manipulate and how you want to see these parameters.

1. Prepare your Modbus Communications Table
    1. Contains all essential elements to view or write to
2. Write down the respective interface you want to implement as an input or output

Table 6.Planning your HMI layout based on table 5.



PLC MODBUS Address	Use of Register	Read Only	Write – Read	Desired Appearance
40001	Input Pins State	X		Digital Panel Meter
40002	Output Pins Trigger		X	None at this time (Not Implemented)
40003	Output Pins State	X		Digital Panel Meter
40004	Analog A0 Value	X		Meter Type
40005	Analog A3 Value	X		Gauge Type
40006	Inputs Trigger		x	None at this time (Not Implemented)

V.Dupo, Utilization of HMI, MODBUS RTU for Applications in Robotics and Control Systems, [Online] https://github.com/VoltsBD/AutomationSystemsIntegrationManual, Last accessed 12 July 2023

WinLog Lite as an HMI Build Tool

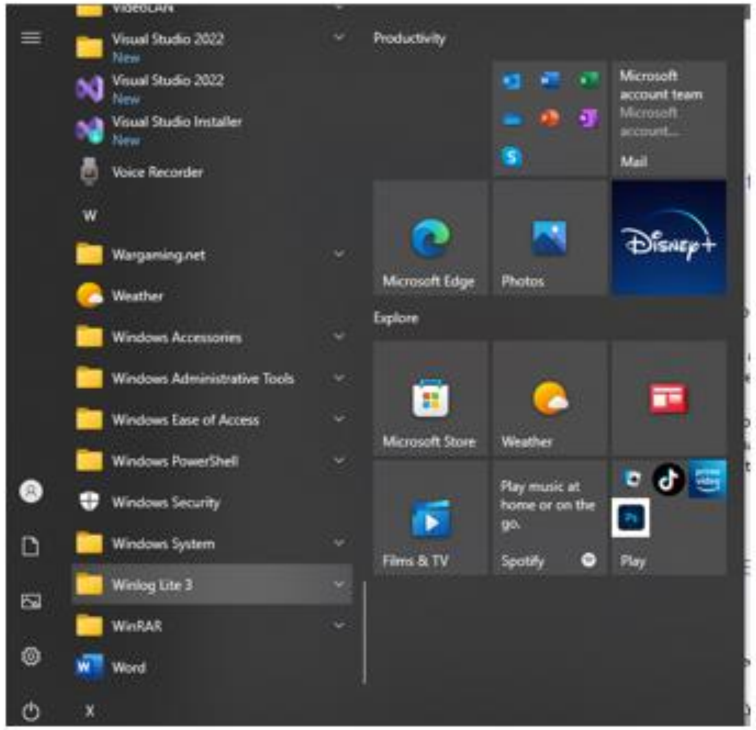
This was a recommended software by my instructor since its PC based but the end product if you have the pro edition can be ported to a java applications. You may download this software thru

- https://www.sielcosistemi.com/en/download/public/winlog_lite.html#:~:text=Winlog%20Lite%20is%20the%20free,60%20minutes%20of%20full%20operation.

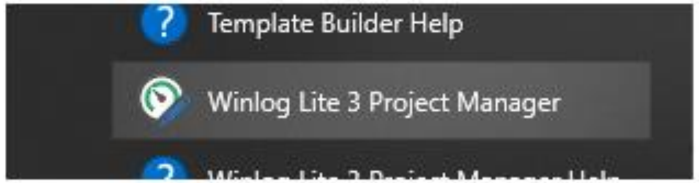
Launching

Preparation for Use and Opening the Program

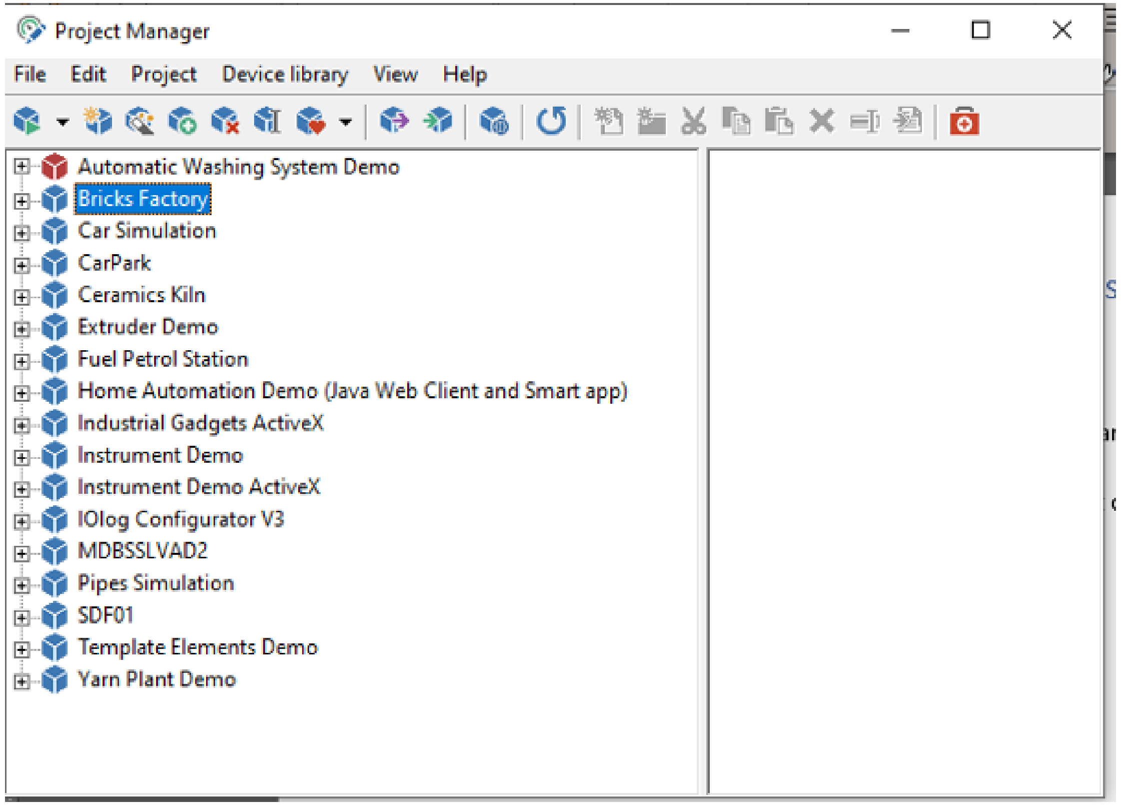
Sielco Sistemi emailed me that Winlog Lite 3.0 as this July 2022 is now not covered by a paid edition which is a shame instead they sell the Winlog Pro edition. This program (Winlog Lite) needs to be installed. After installation you can find it on the W section of the applications menu area.



Click on the Winlog Lite 3 folder and find and click on the Project Manager Application

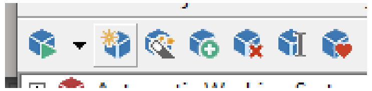

Template Builder Help
Winlog Lite 3 Project Manager
Winlog Lite 3 Project Manager Help

Creating New Projects

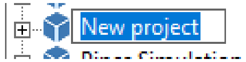


- Automatic Washing System Demo
- Bricks Factory
- Car Simulation
- CarPark
- Ceramics Kiln
- Extruder Demo
- Fuel Petrol Station
- Home Automation Demo (Java Web Client and Smart app)
- Industrial Gadgets ActiveX
- Instrument Demo
- Instrument Demo ActiveX
- IOlog Configurator V3
- MDBSSLVAD2
- Pipes Simulation
- SDF01
- Template Elements Demo
- Yarn Plant Demo

To create a new file click on the box icon with the sun.



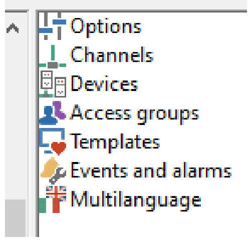
This should lead to a new project being declared.



You can rename this to your liking.

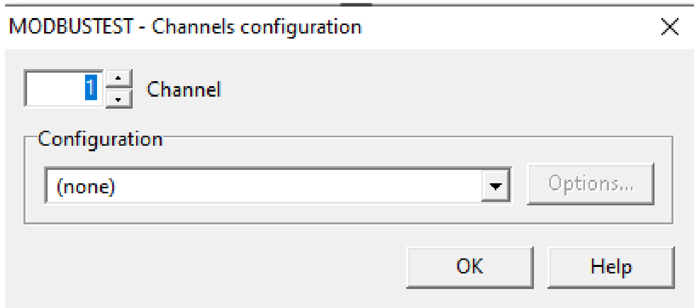
Configuration

Click on the configuration folder it should show you the following



Setting up Modbus communication

Double Click on the Channels. It should open up a new popup window

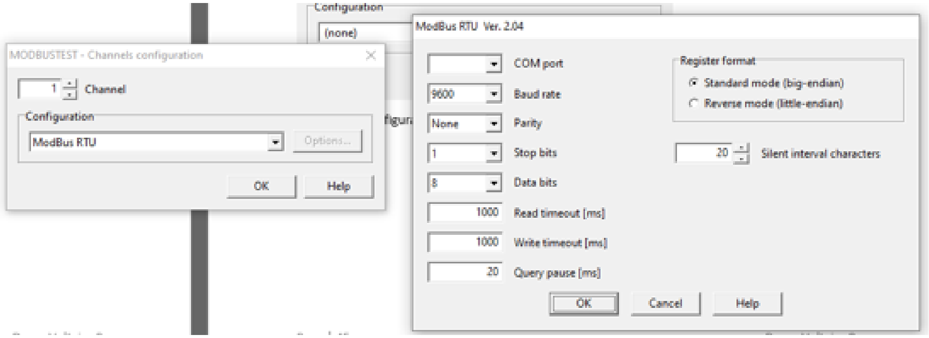


MODBUSTEST - Channels configuration

Channel: 1

Configuration: (none)

Click on the configuration, find MODBUS RTU and select it from the menu



MODBUSTEST - Channels configuration

Channel: 1

Configuration: ModBus RTU

ModBus RTU Ver. 2.04

COM port:

Baud rate: 9600

Parity: None

Stop bits: 1

Data bits: 8

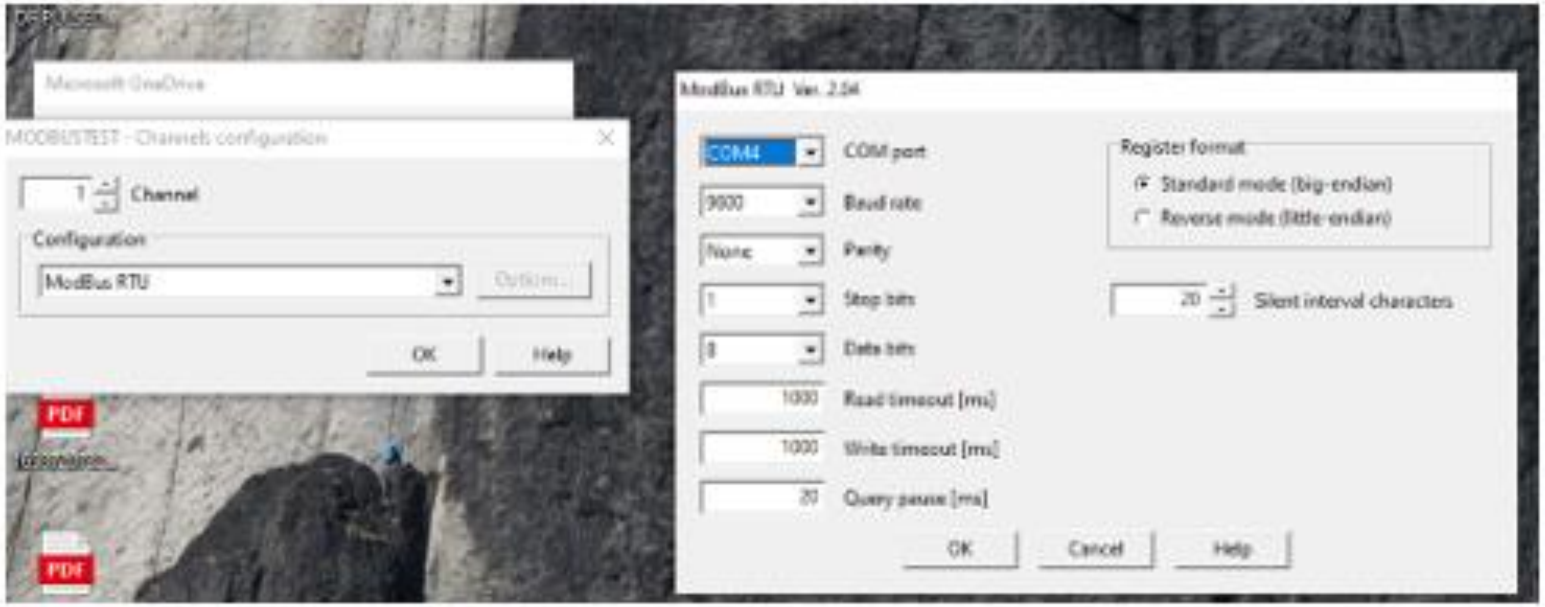
Read timeout [ms]: 1000

Write timeout [ms]: 1000

Query pause [ms]: 20

Register format: ● Standard mode (big-endian) ○ Reverse mode (little-endian)

Silent interval characters: 20



MODBUSTEST - Channels configuration

Channel: 1

Configuration: ModBus RTU

ModBus RTU Ver. 2.04

COM port: COM4

Baud rate: 9600

Parity: None

Stop bits: 1

Data bits: 8

Read timeout [ms]: 1000

Write timeout [ms]: 1000

Query pause [ms]: 20

Register format: ● Standard mode (big-endian) ○ Reverse mode (little-endian)

Silent interval characters: 20

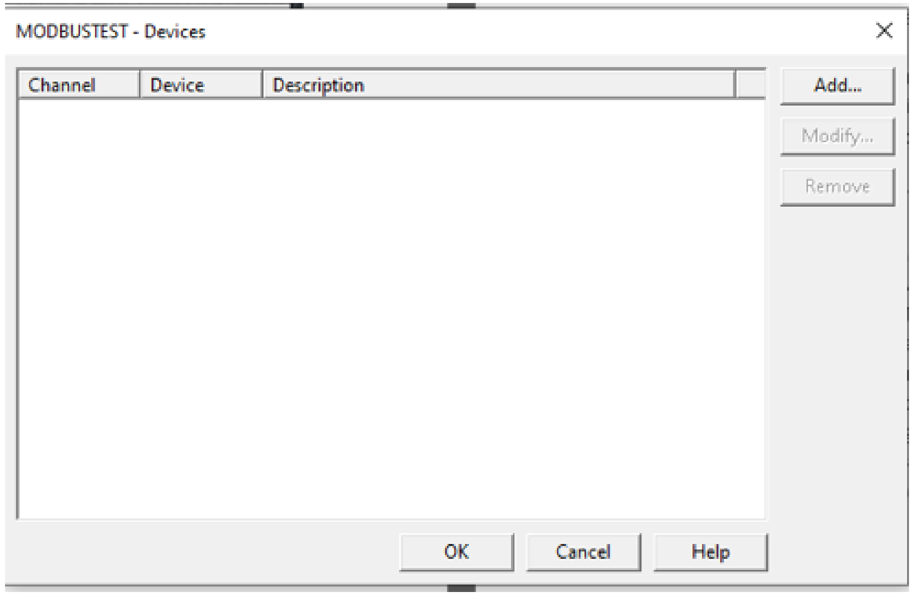
Set the COMPORT, Parity, Stop Bits and Data Bits

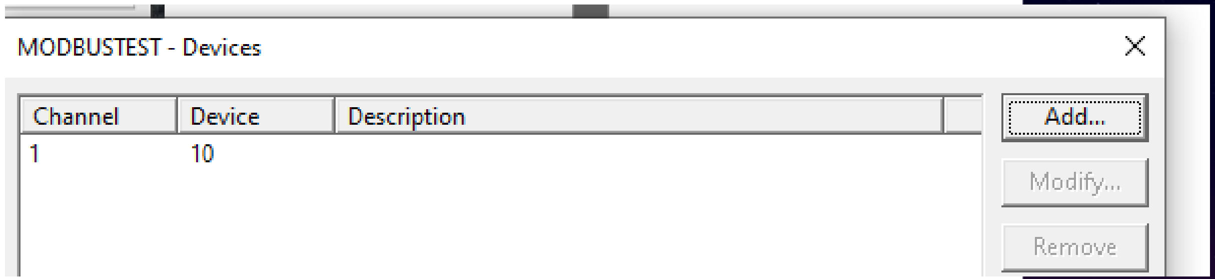
Configuration

Press OK on the MODBUS RTU Configuration Pop – up

Press OK on the Channels Configuration this will bring you back to the Project Manager Window

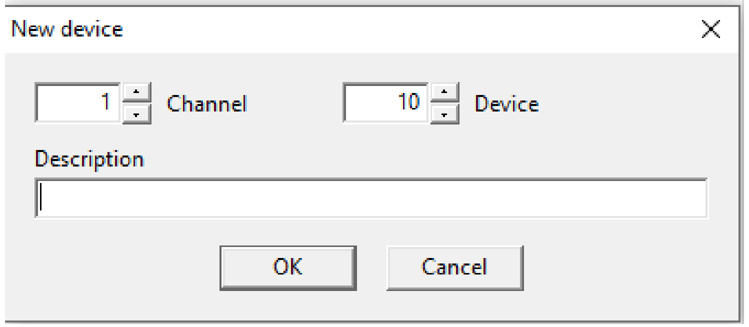
Proceed to double click on the device.





Channel	Device	Description
1	10	

Click on the Add Button and set your Slave Device ID here



Channel: 1

Device: 10

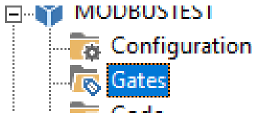
You can be able to verify the configuration settings will be shown to you after you confirm the new device.

ARDUINO CURIOSITY

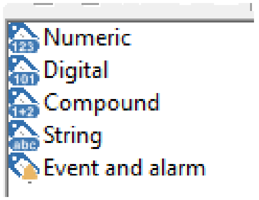
CRAFTING ELECTRONICAL SENSORS

Creation of Memory Containers

Proceed to click on the Gates folder inside the project your editing.



This will show you the following data types



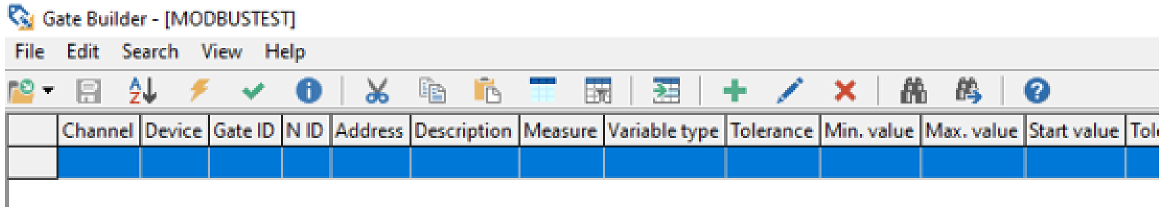
Gates are also known as tags in other general language used in PLC

At this point open up your program or au16data table to know what is the nature of the inputs for this example its actually

Table 7. au16data Containers

0	1	2	3	4	5	6	7	8
Prime	Command	Value	referencetime	count	Elapsed time		Motor On / Off	SML Value

Click on the Numeric this should open up the Gate Builder Screen



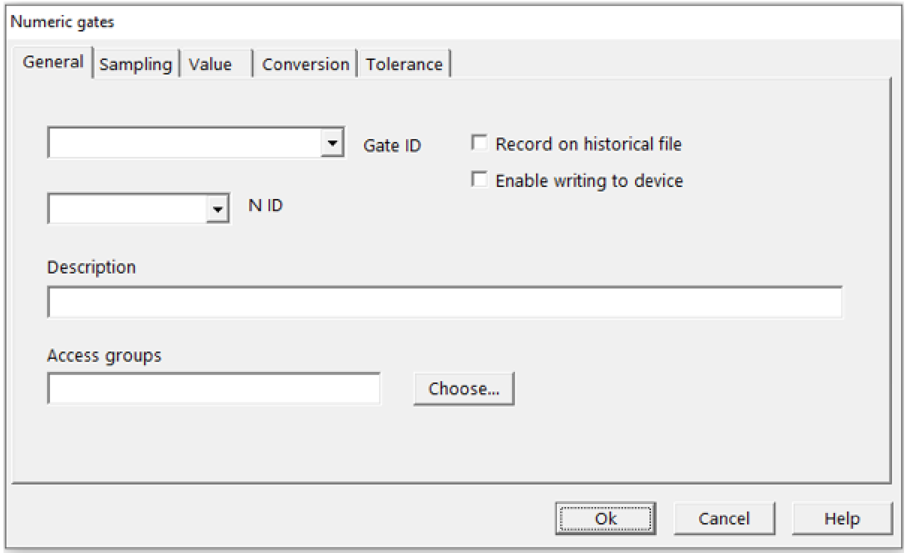
Press the green plus sign to open listing of new tags.

Use your version of Table 7 to create the table.

Table 6.Planning your HMI layout based on table 5.

PLC MODBUS Address	Use of Register	Read Only	Write – Read	Desired Appearance
40001	Input Pins State	X		Digital Panel Meter
40002	Output Pins Trigger		X	None at this time (Not Implemented)
40003	Output Pins State	X		Digital Panel Meter
40004	Analog A0 Value	X		Meter Type
40005	Analog A3 Value	X		Gauge Type
40006	Inputs Trigger		x	None at this time (Not Implemented)

Creation of Memory Containers



Gate ID:

N ID:

☐ Record on historical file

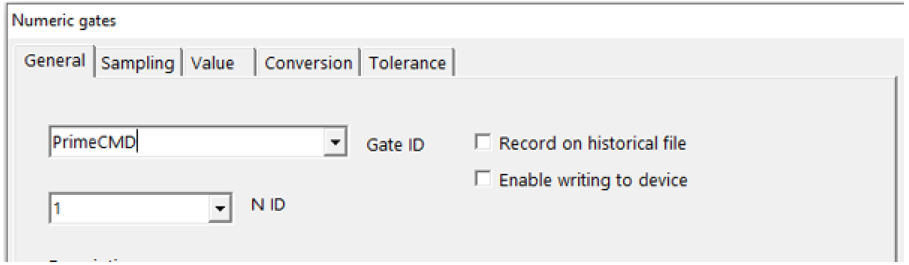
☐ Enable writing to device

Description:

Access groups:

Add in the Gate ID this is a text value at assigns a more easier to remember name for the type of data.

Add in the N ID which is just a number that denotes number in a series of similar numerical tags there are you should start at 1.



Gate ID: PrimeCMD

N ID: 1

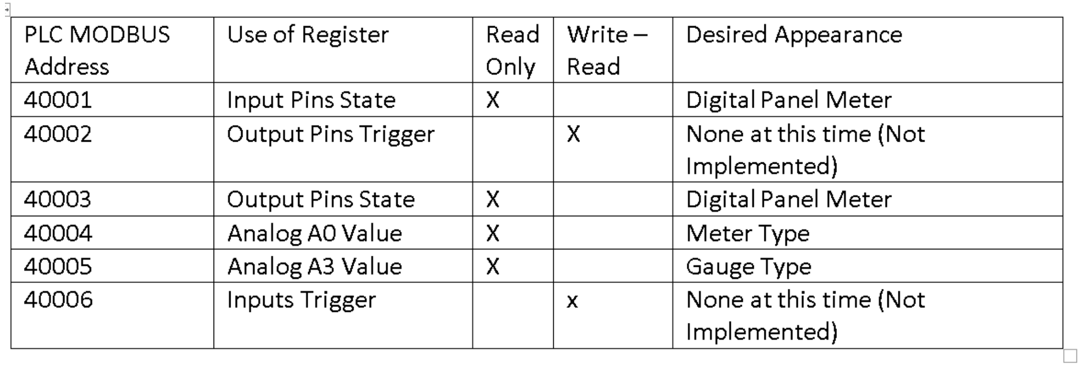
☐ Record on historical file

☐ Enable writing to device

Press the green plus sign to open listing of new tags.

Use your version of Table 7 to create the table.

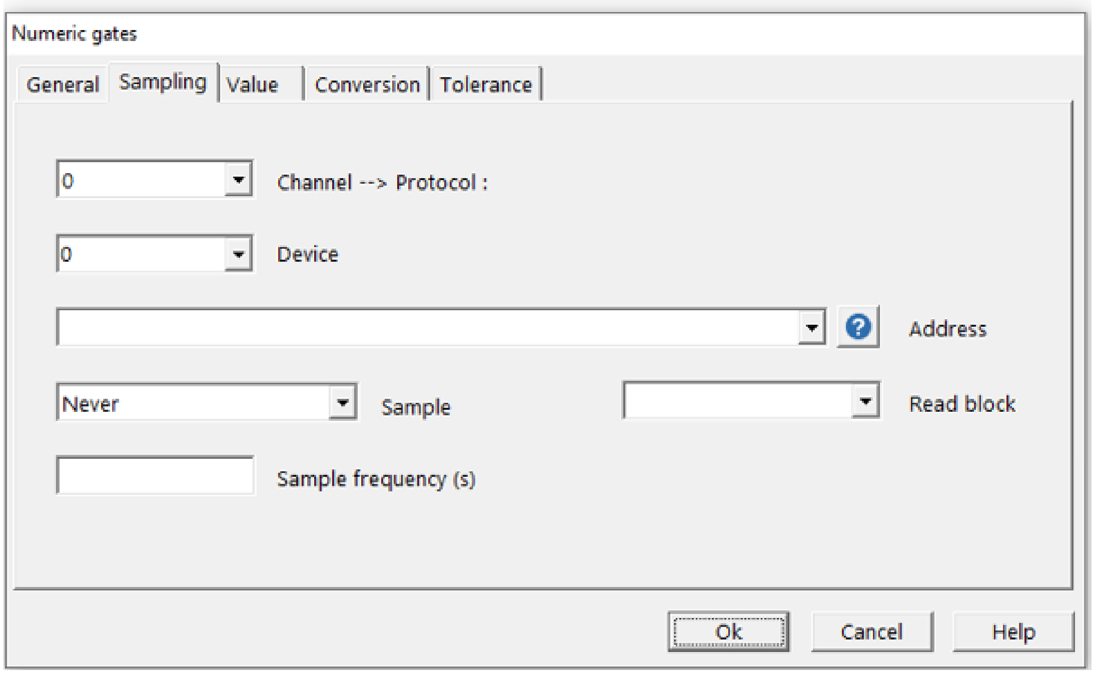
Table 6.Planning your HMI layout based on table 5.



PLC MODBUS Address	Use of Register	Read Only	Write – Read	Desired Appearance
40001	Input Pins State	X		Digital Panel Meter
40002	Output Pins Trigger		X	None at this time (Not Implemented)
40003	Output Pins State	X		Digital Panel Meter
40004	Analog A0 Value	X		Meter Type
40005	Analog A3 Value	X		Gauge Type
40006	Inputs Trigger		x	None at this time (Not Implemented)

Creation of Memory Containers

Click on the Sampling Tab these open configuration items should be shown.



Numeric gates

Channel --> Protocol : 0

Device: 0

Address:

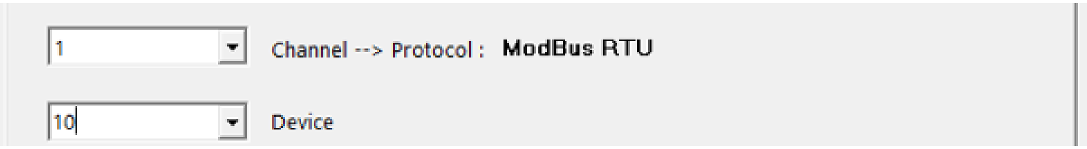
Sample: Never

Read block:

Sample frequency (s):

Select Channel you configured previously

Select the device ID number of your Arduino.



Channel --> Protocol : 1 ModBus RTU

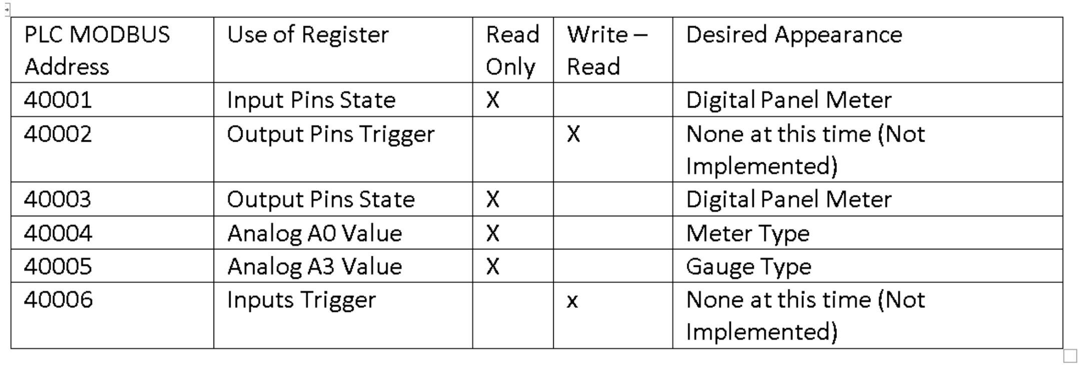
Device: 10

You will notice upon selecting 1 Ch the name of the protocol is shown associated with this channel.

Press the green plus sign to open listing of new tags.

Use your version of Table 7 to create the table.

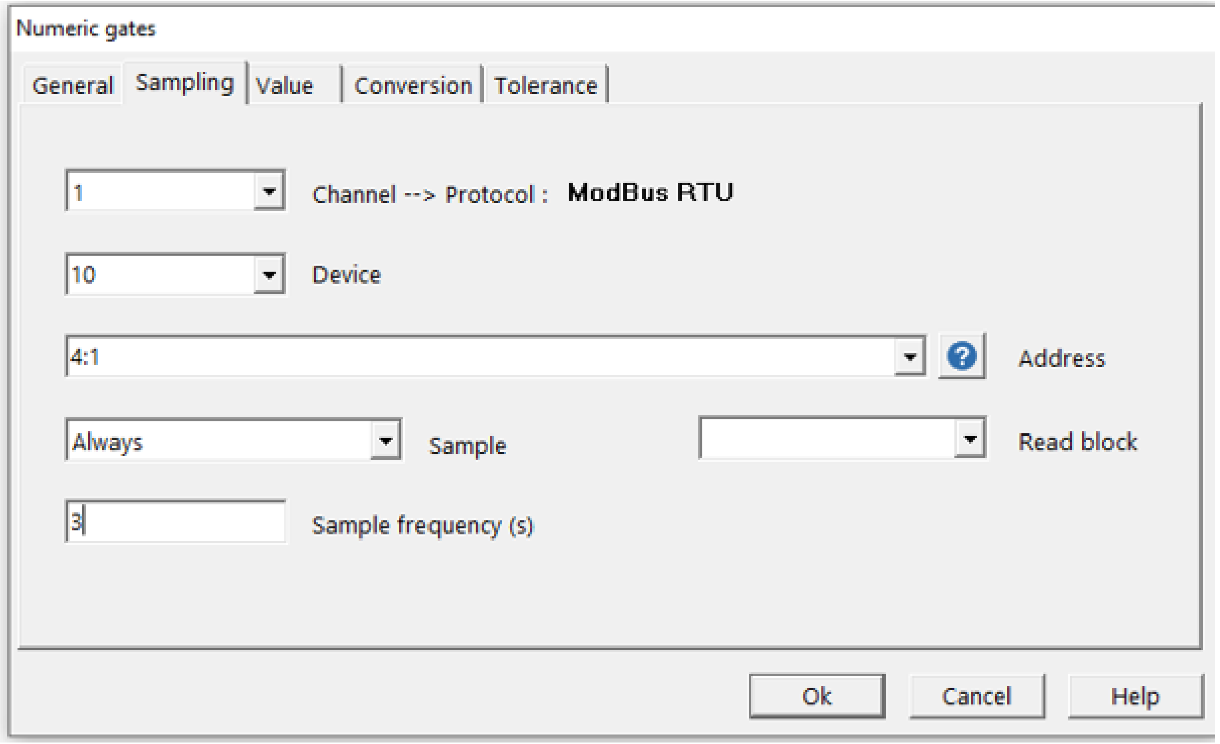
Table 6.Planning your HMI layout based on table 5.



PLC MODBUS Address	Use of Register	Read Only	Write – Read	Desired Appearance
40001	Input Pins State	X		Digital Panel Meter
40002	Output Pins Trigger		X	None at this time (Not Implemented)
40003	Output Pins State	X		Digital Panel Meter
40004	Analog A0 Value	X		Meter Type
40005	Analog A3 Value	X		Gauge Type
40006	Inputs Trigger		x	None at this time (Not Implemented)

Creation of Memory Containers

Let the program sample the register with the setting Always



Numeric gates

1 Channel --> Protocol : **ModBus RTU**

10 Device

4:1 Address

Always Sample

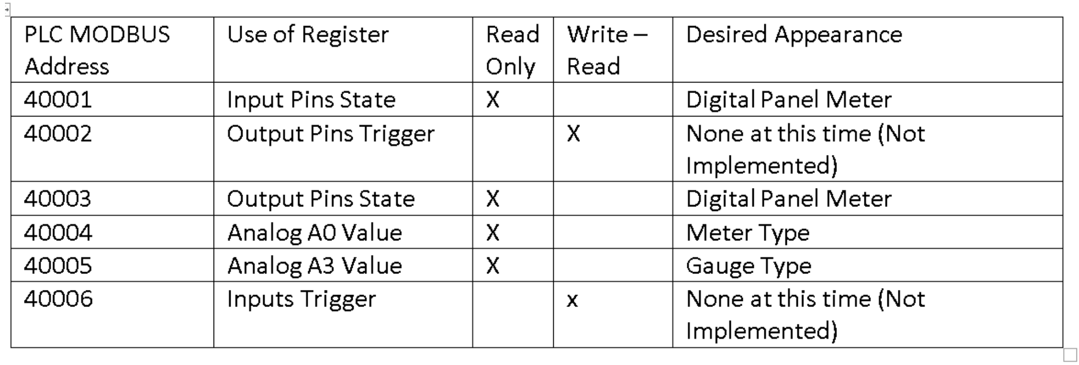
Read block

3 Sample frequency (s)

The sample frequency is up to you to determine and press ok

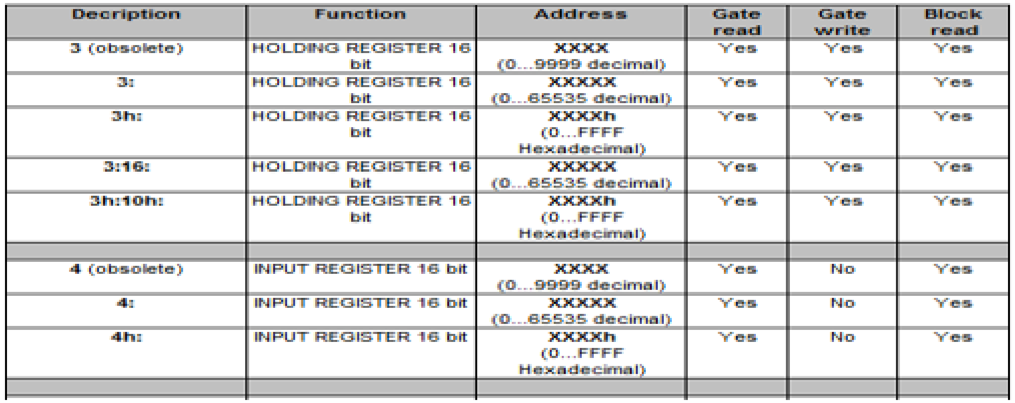
Populate the Gate Builder with more numerictags based on Table 7.

Table 6.Planning your HMI layout based on table 5.



PLC MODBUS Address	Use of Register	Read Only	Write – Read	Desired Appearance
40001	Input Pins State	X		Digital Panel Meter
40002	Output Pins Trigger		X	None at this time (Not Implemented)
40003	Output Pins State	X		Digital Panel Meter
40004	Analog A0 Value	X		Meter Type
40005	Analog A3 Value	X		Gauge Type
40006	Inputs Trigger		x	None at this time (Not Implemented)

Table 8. Address Prefixes for Numerical Input and Holding Register

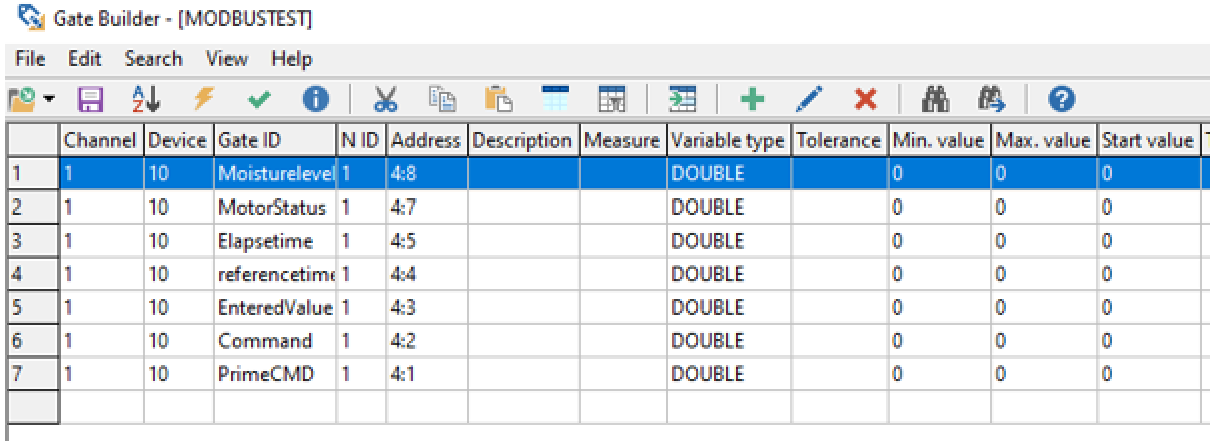


Decription	Function	Address	Gate read	Gate write	Block read
3 (obsolete)	HOLDING REGISTER 16 bit	XXXX (0...9999 decimal)	Yes	Yes	Yes
3:	HOLDING REGISTER 16 bit	XXXXX (0...65535 decimal)	Yes	Yes	Yes
3h:	HOLDING REGISTER 16 bit	XXXXh (0...FFFF Hexadecimal)	Yes	Yes	Yes
3:16:	HOLDING REGISTER 16 bit	XXXXX (0...65535 decimal)	Yes	Yes	Yes
3h:10h:	HOLDING REGISTER 16 bit	XXXXh (0...FFFF Hexadecimal)	Yes	Yes	Yes
4 (obsolete)	INPUT REGISTER 16 bit	XXXX (0...9999 decimal)	Yes	No	Yes
4:	INPUT REGISTER 16 bit	XXXXX (0...65535 decimal)	Yes	No	Yes
4h:	INPUT REGISTER 16 bit	XXXXh (0...FFFF Hexadecimal)	Yes	No	Yes

Creation of Memory Containers

The sample frequency is up to you to determine and press ok

Populate the Gate Builder with more numerictags based on Table 7.



	Channel	Device	Gate ID	N ID	Address	Description	Measure	Variable type	Tolerance	Min. value	Max. value	Start value
1	1	10	Moisturelevel	1	4:8			DOUBLE		0	0	0
2	1	10	MotorStatus	1	4:7			DOUBLE		0	0	0
3	1	10	Elapsetime	1	4:5			DOUBLE		0	0	0
4	1	10	referencetime	1	4:4			DOUBLE		0	0	0
5	1	10	EnteredValue	1	4:3			DOUBLE		0	0	0
6	1	10	Command	1	4:2			DOUBLE		0	0	0
7	1	10	PrimeCMD	1	4:1			DOUBLE		0	0	0

Now we need to edit some values like the Maximum Values

Press the Save Button [Diskette Icon]

Leave the Gate Builder Window.

Press the green plus sign to open listing of new tags.

Use your version of Table 7 to create the table.

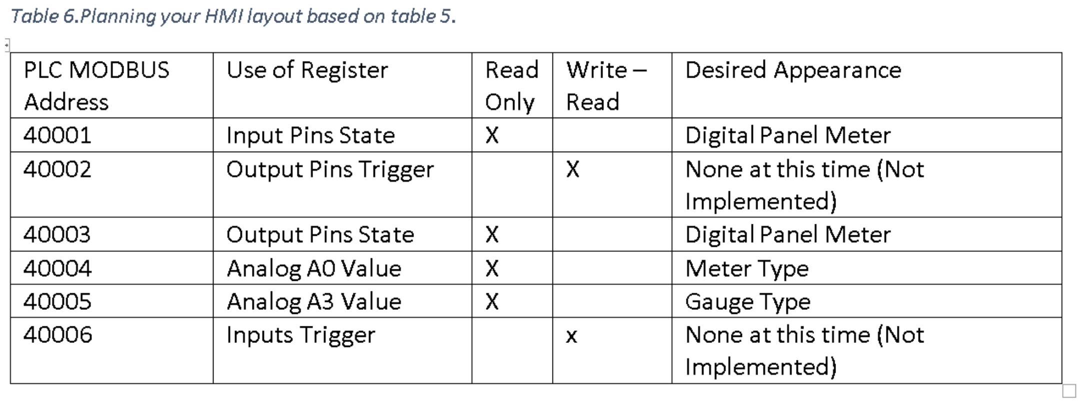
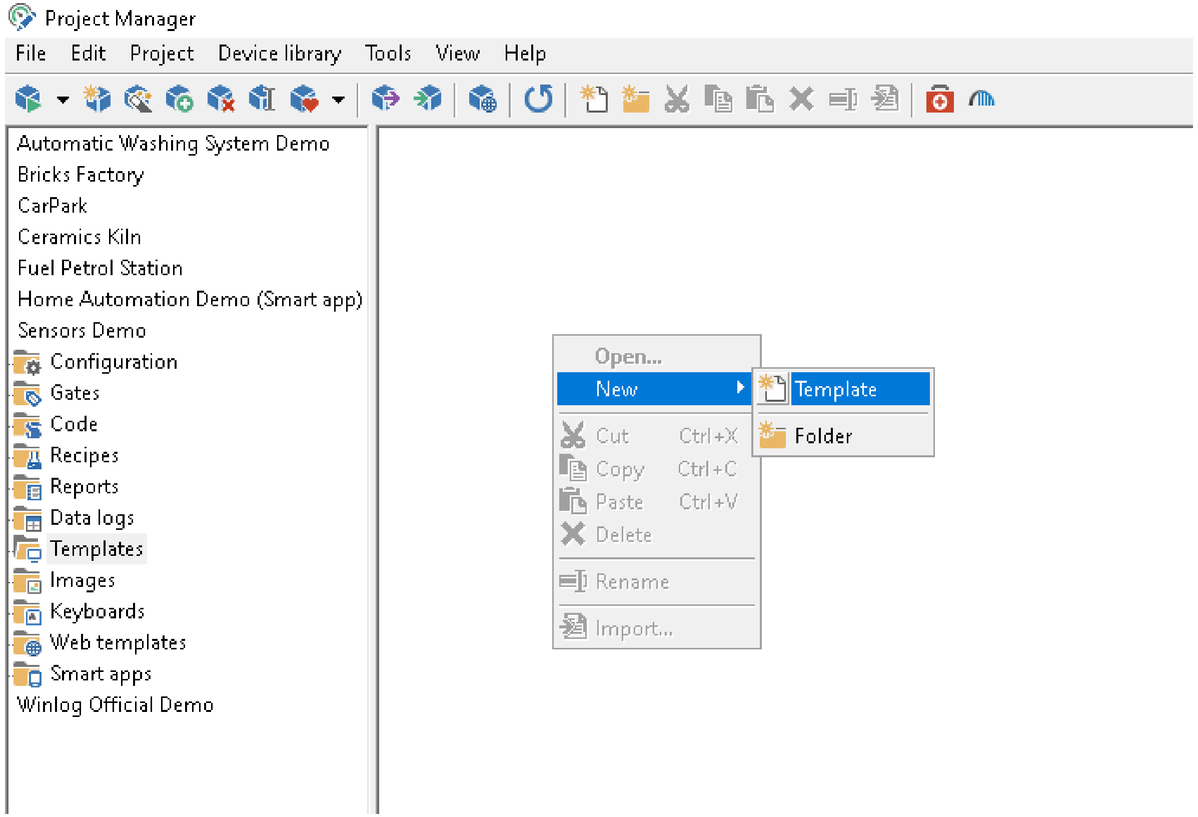


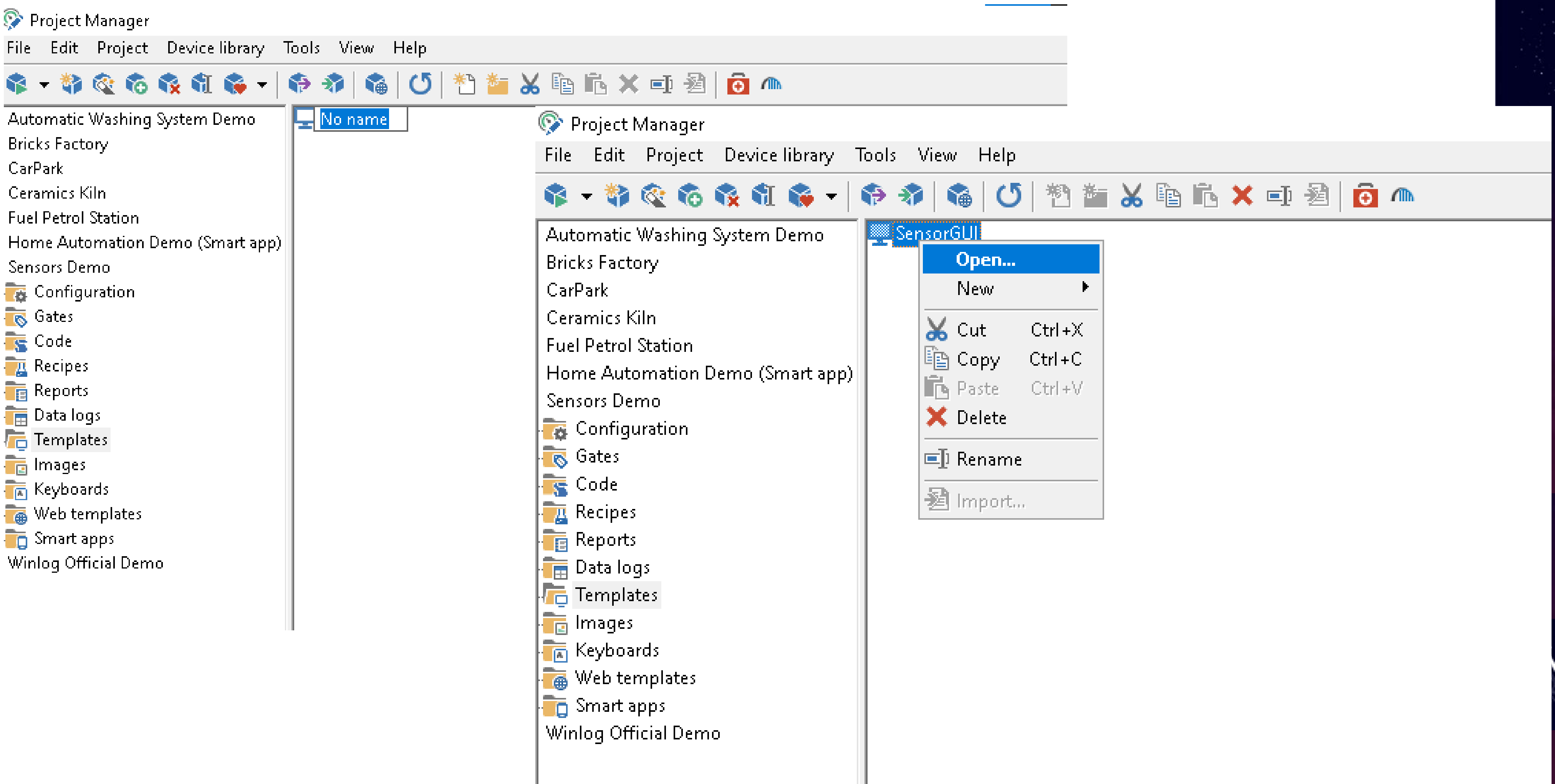
Table 6.Planning your HMI layout based on table 5.

PLC MODBUS Address	Use of Register	Read Only	Write – Read	Desired Appearance
40001	Input Pins State	X		Digital Panel Meter
40002	Output Pins Trigger		X	None at this time (Not Implemented)
40003	Output Pins State	X		Digital Panel Meter
40004	Analog A0 Value	X		Meter Type
40005	Analog A3 Value	X		Gauge Type
40006	Inputs Trigger		x	None at this time (Not Implemented)

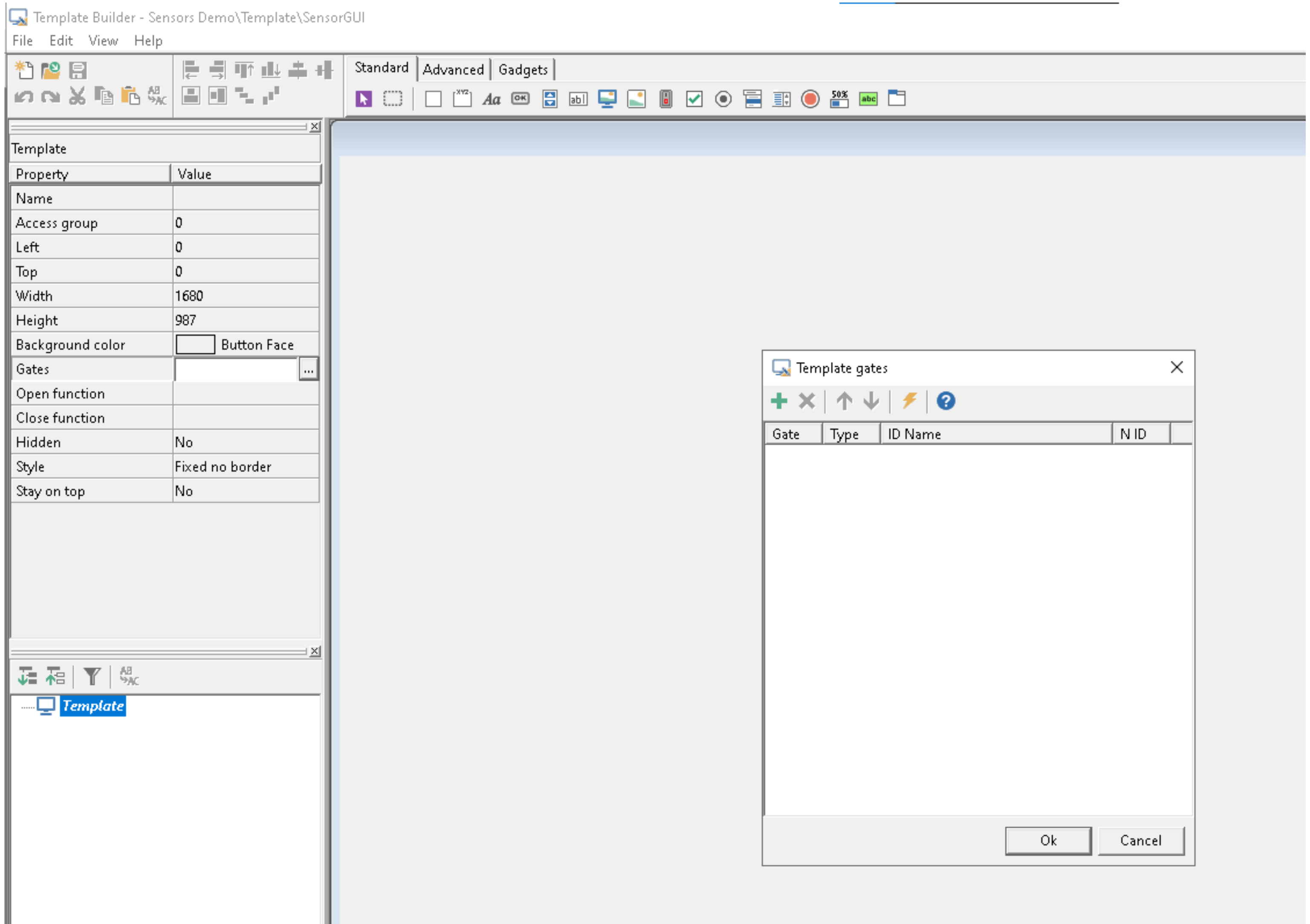
Template
Global Values Inheritance



Template
Global Values Inheritance



Template
Global Values Inheritance



Template Builder - Sensors Demo\Template\SensorGUI

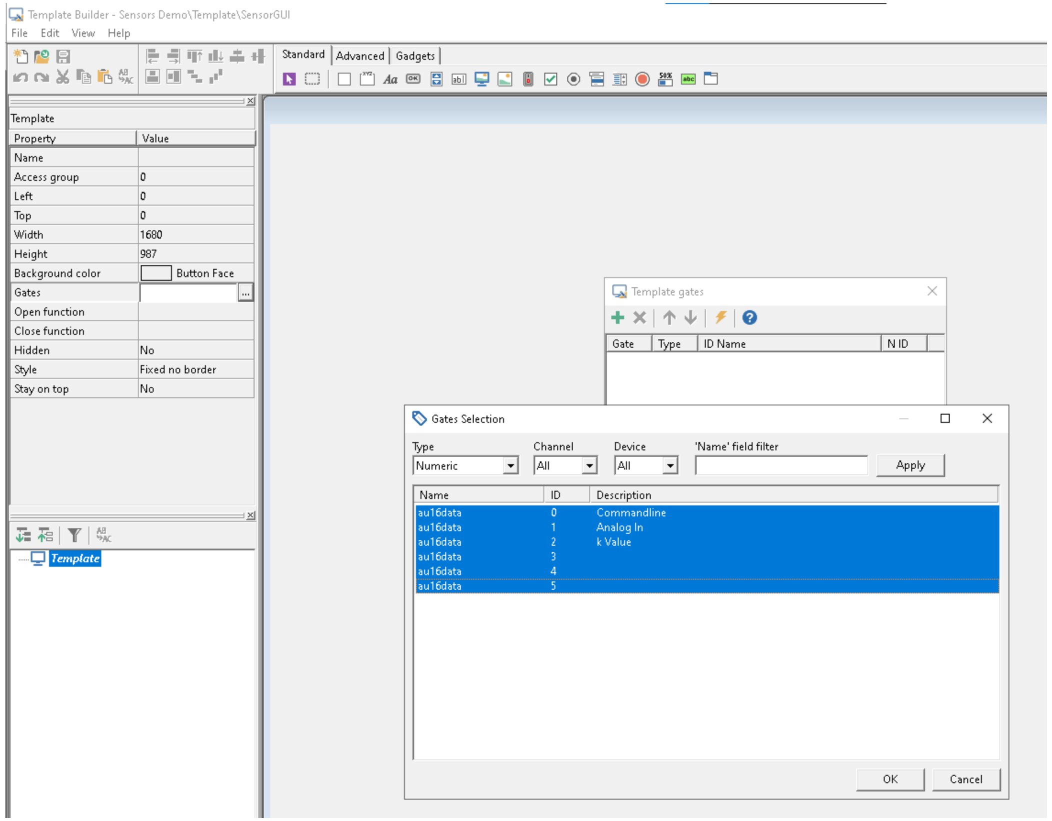
Template

Property	Value
Name	
Access group	0
Left	0
Top	0
Width	1680
Height	987
Background color	Button Face
Gates	...
Open function	
Close function	
Hidden	No
Style	Fixed no border
Stay on top	No

Template gates

Gate	Type	ID Name	N ID

Template
Global Values Inheritance



Template Builder - Sensors Demo\Template\SensorGUI

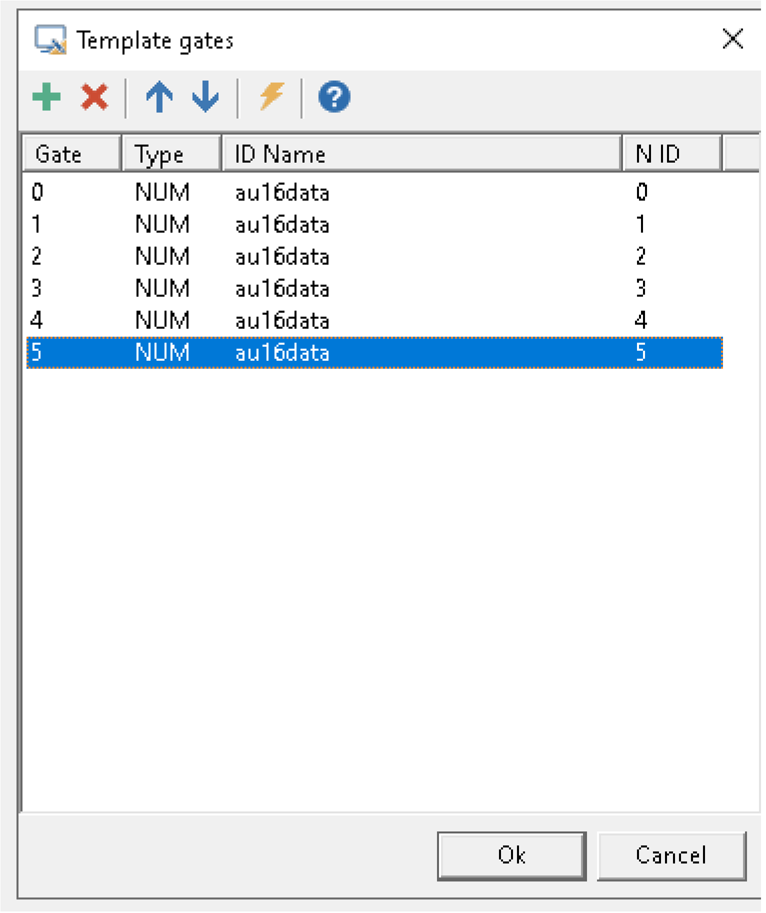
Template

Property	Value
Name	
Access group	0
Left	0
Top	0
Width	1680
Height	987
Background color	Button Face
Gates	...
Open function	
Close function	
Hidden	No
Style	Fixed no border
Stay on top	No

Gates Selection

Type: Numeric
Channel: All
Device: All
'Name' field filter:
Apply

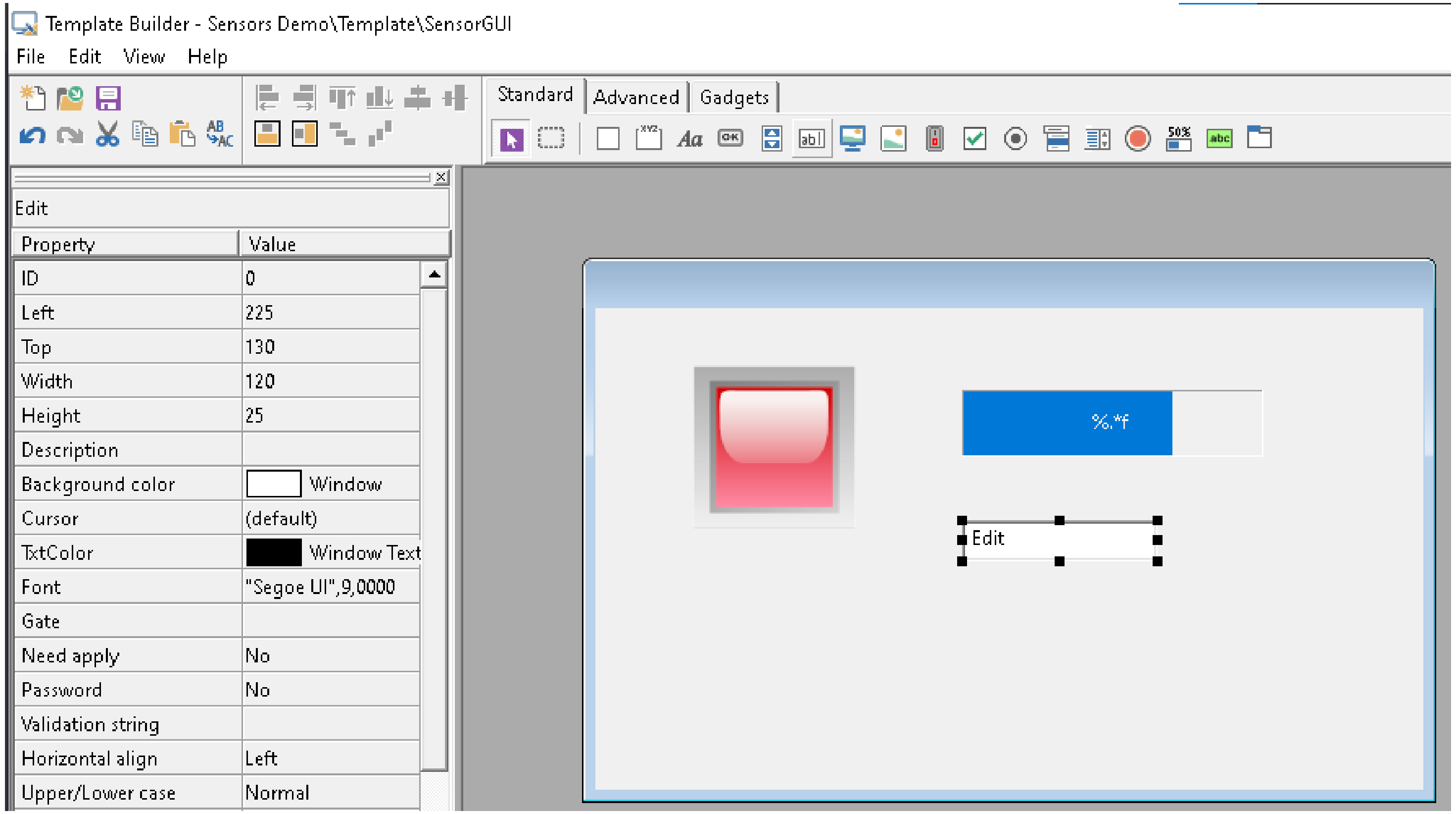
Name	ID	Description
au16data	0	Commandline
au16data	1	Analog In
au16data	2	k Value
au16data	3	
au16data	4	
au16data	5	



Template gates

Gate	Type	ID Name	N ID
0	NUM	au16data	0
1	NUM	au16data	1
2	NUM	au16data	2
3	NUM	au16data	3
4	NUM	au16data	4
5	NUM	au16data	5

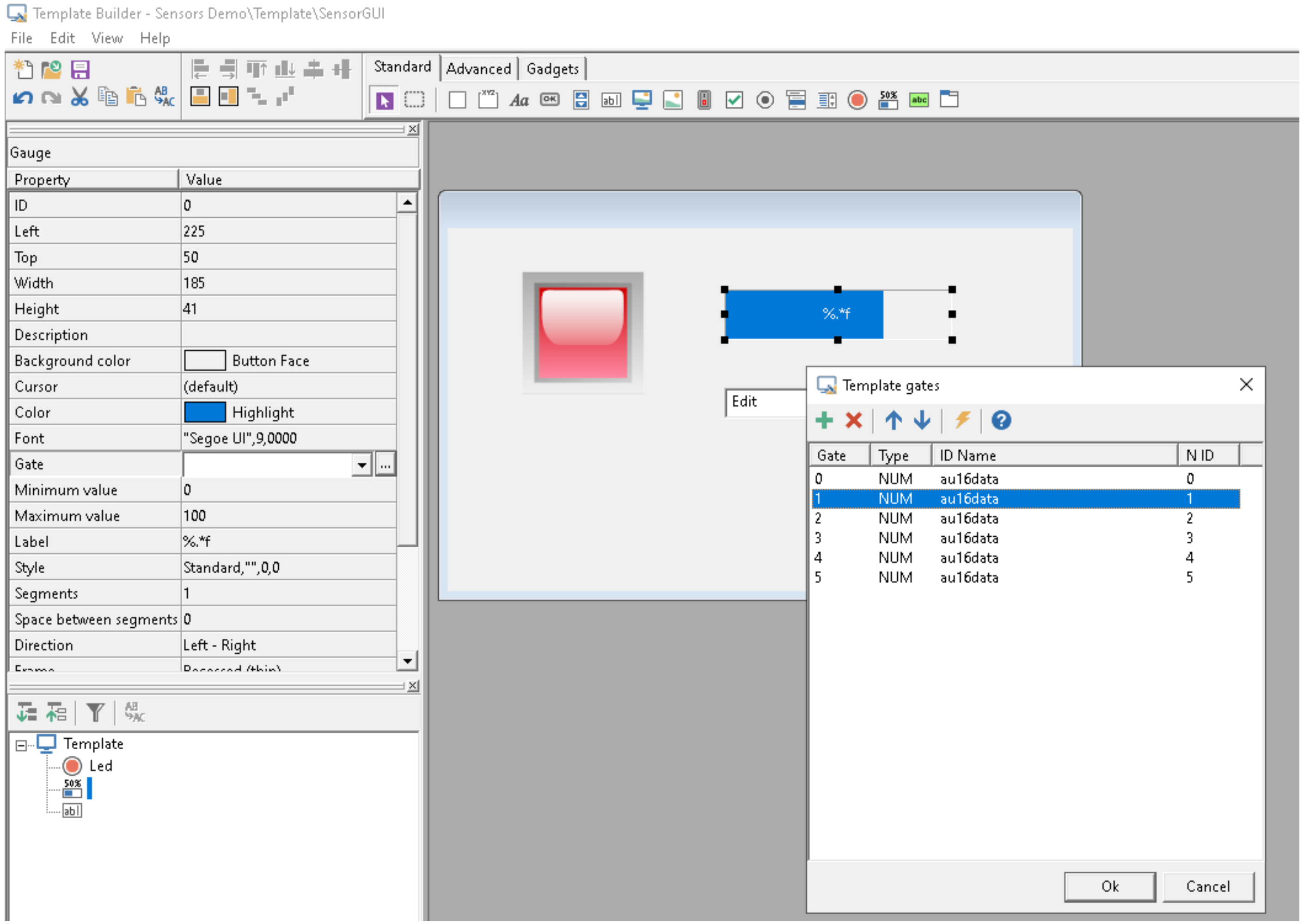
Template
Global Values Inheritance



Edit

Property	Value
ID	0
Left	225
Top	130
Width	120
Height	25
Description	
Background color	Window
Cursor	(default)
TxtColor	Window Text
Font	"Segoe UI",9,0000
Gate	
Need apply	No
Password	No
Validation string	
Horizontal align	Left
Upper/Lower case	Normal

Template
Assigning Numeric Values

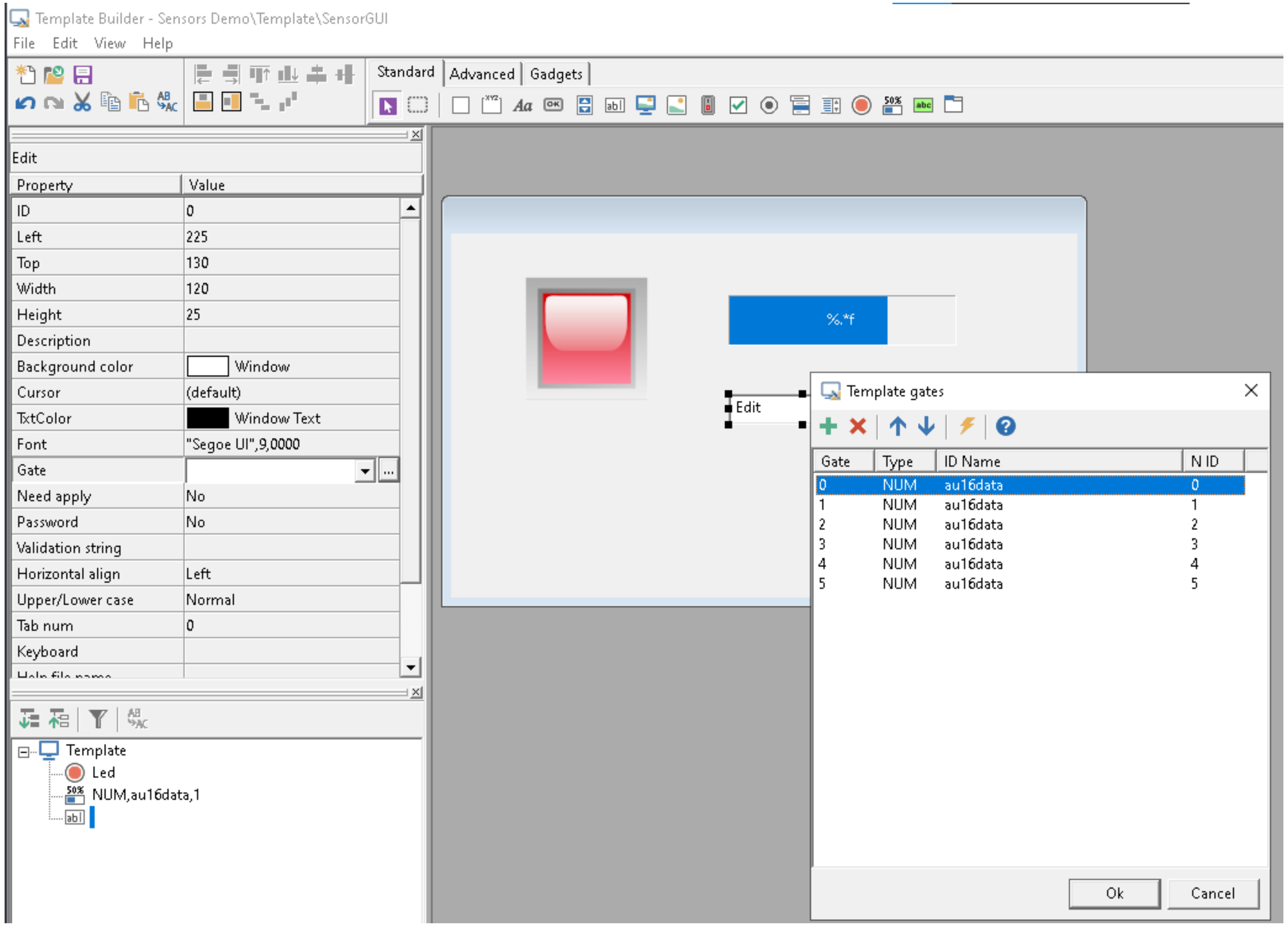


Gauge

Property	Value
ID	0
Left	225
Top	50
Width	185
Height	41
Description	
Background color	Button Face
Cursor	(default)
Color	Highlight
Font	"Segoe UI",9,0000
Gate	
Minimum value	0
Maximum value	100
Label	%.*f
Style	Standard,"",0,0
Segments	1
Space between segments	0
Direction	Left - Right
Frame	Recessed (thin)

Gate	Type	ID Name	N ID
0	NUM	au16data	0
1	NUM	au16data	1
2	NUM	au16data	2
3	NUM	au16data	3
4	NUM	au16data	4
5	NUM	au16data	5

Template
Assigning Writing Surface



Edit

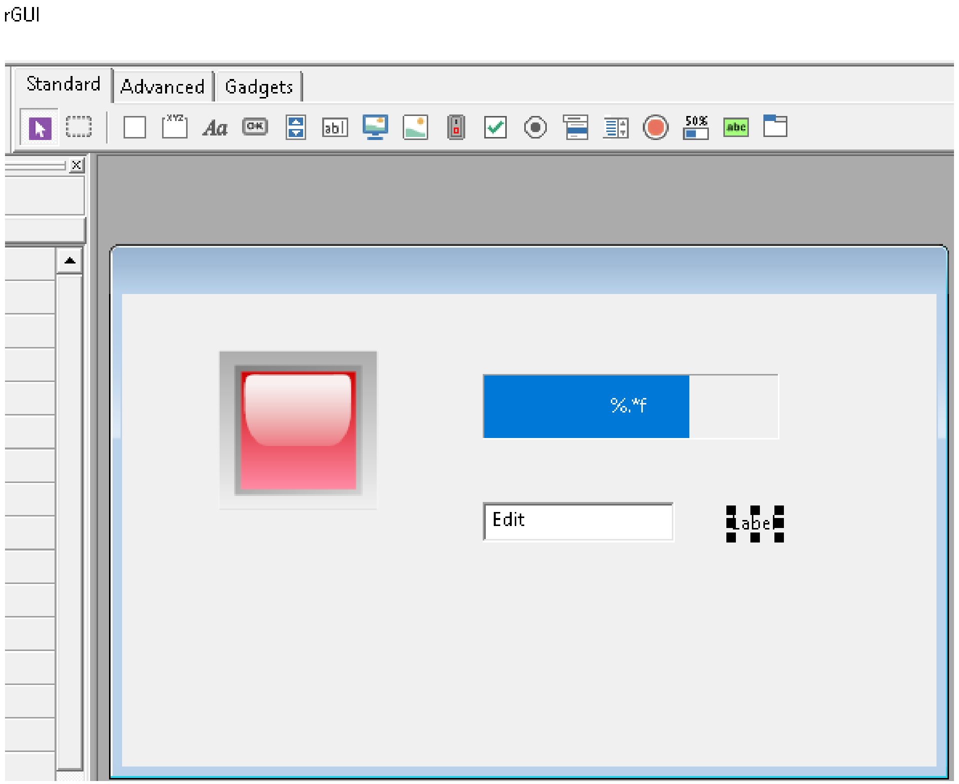
Property	Value
ID	0
Left	225
Top	130
Width	120
Height	25
Description	
Background color	Window
Cursor	(default)
TxtColor	Window Text
Font	"Segoe UI",9,0000
Gate	
Need apply	No
Password	No
Validation string	
Horizontal align	Left
Upper/Lower case	Normal
Tab num	0
Keyboard	

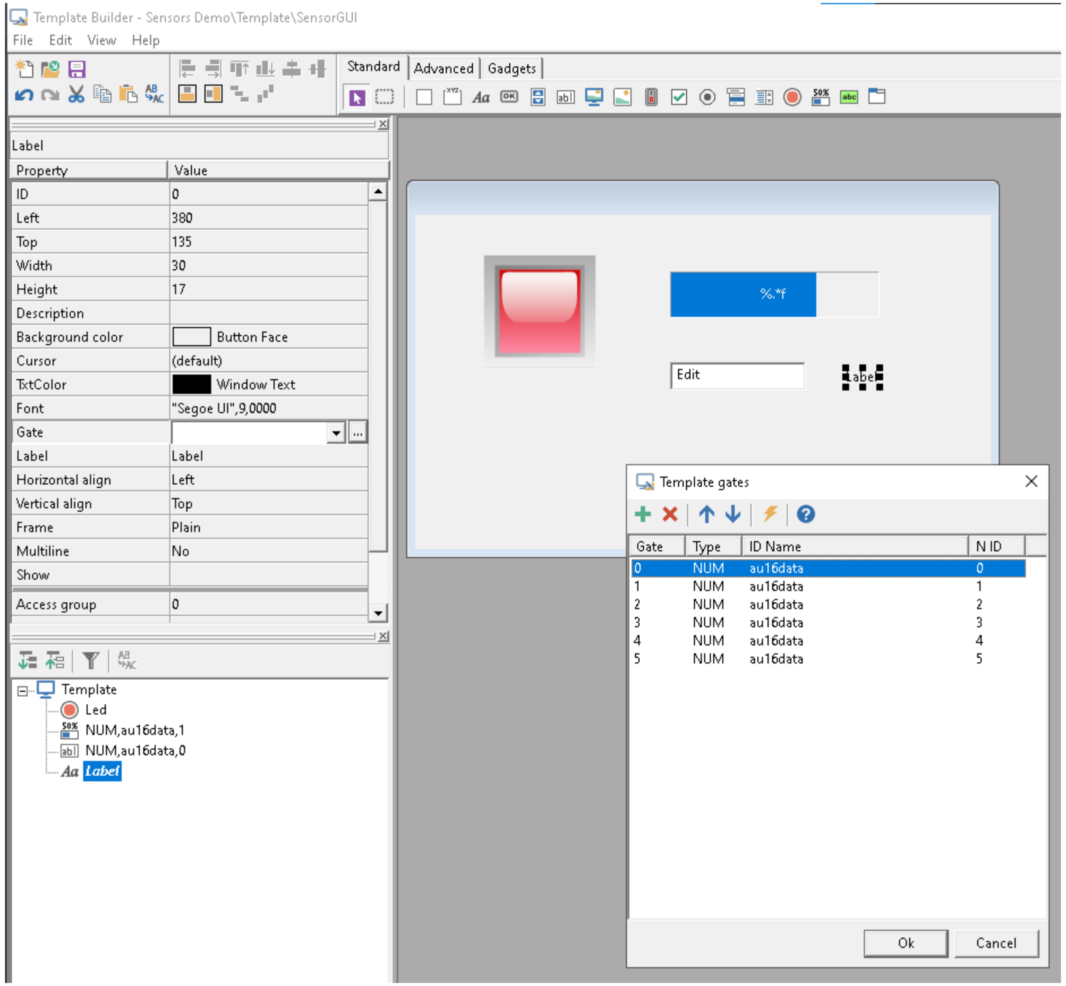
- Template
  - Led
  - NUM,au16data,1

Template gates

Gate	Type	ID Name	N ID
0	NUM	au16data	0
1	NUM	au16data	1
2	NUM	au16data	2
3	NUM	au16data	3
4	NUM	au16data	4
5	NUM	au16data	5

Template
Assigning Write Value Monitor





Label

Property	Value
ID	0
Left	380
Top	135
Width	30
Height	17
Description	
Background color	Button Face
Cursor	(default)
TxtColor	Window Text
Font	"Segoe UI",9,0000
Gate	
Label	Label
Horizontal align	Left
Vertical align	Top
Frame	Plain
Multiline	No
Show	
Access group	0

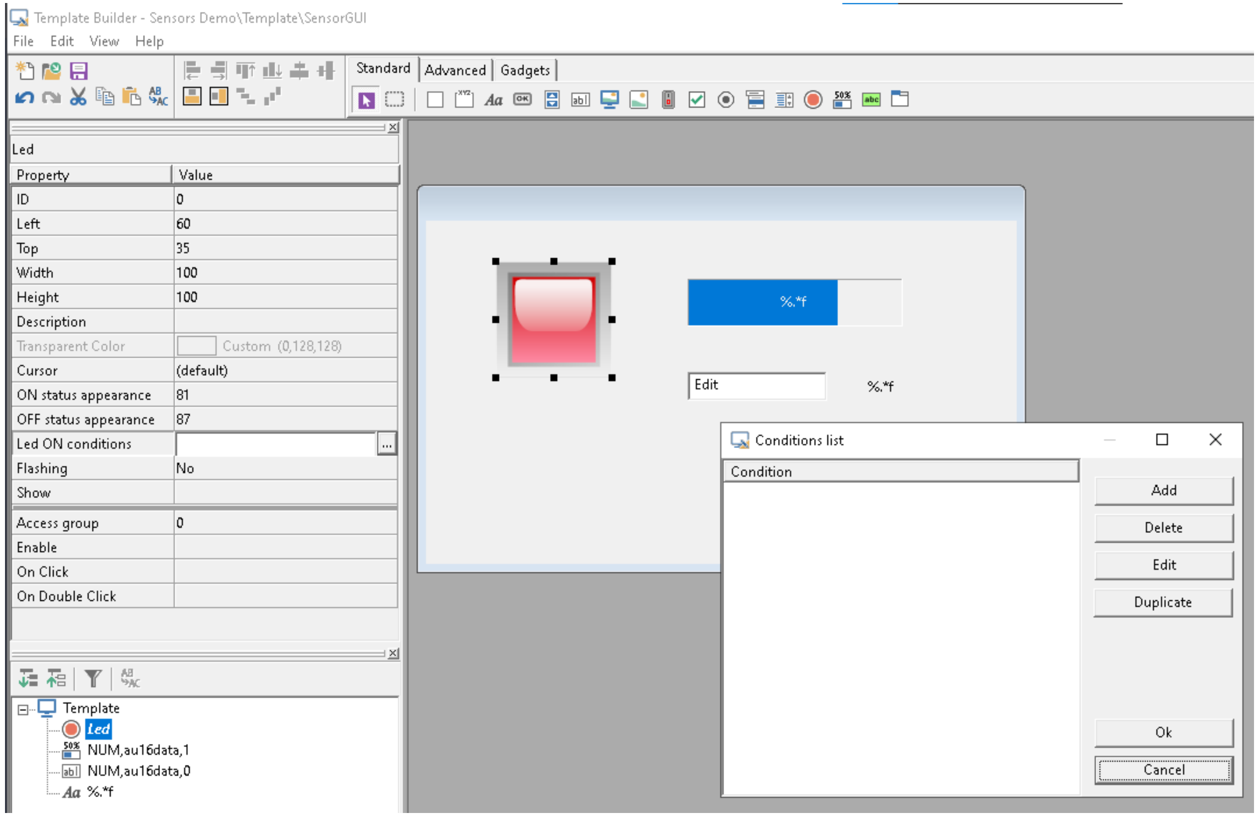
- Template
  - Led
  - NUM,au16data,1
  - NUM,au16data,0
  - Label

Template gates

Gate	Type	ID Name	N ID
0	NUM	au16data	0
1	NUM	au16data	1
2	NUM	au16data	2
3	NUM	au16data	3
4	NUM	au16data	4
5	NUM	au16data	5

ARDUINO CURIOSITY
CRAFTING ELECTRONICAL SENSORS

Template
Global Values Inheritance



Template Builder - Sensors Demo\Template\SensorGUI

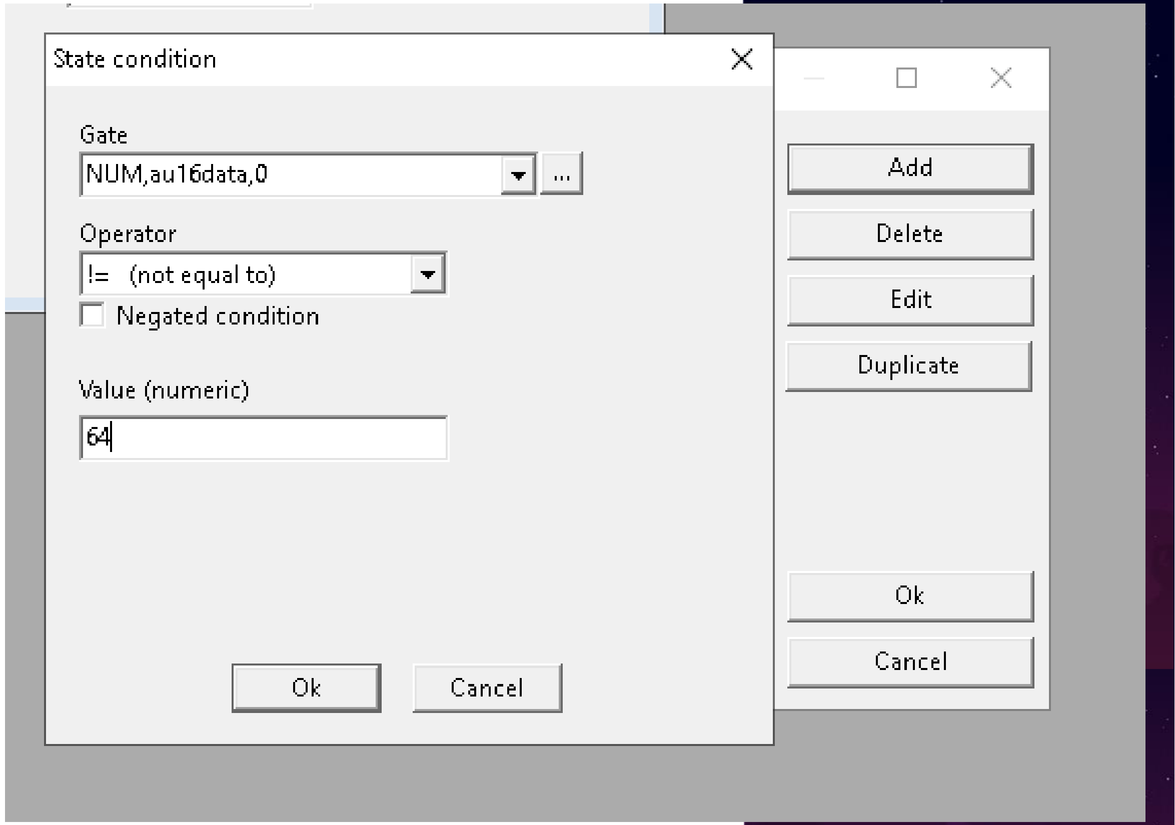
Led

Property	Value
ID	0
Left	60
Top	35
Width	100
Height	100
Description	
Transparent Color	Custom (0,128,128)
Cursor	(default)
ON status appearance	81
OFF status appearance	87
Led ON conditions	
Flashing	No
Show	
Access group	0
Enable	
On Click	
On Double Click	

Template
- Led
- NUM,au16data,1
- NUM,au16data,0
- %.*f

Conditions list

Condition



State condition

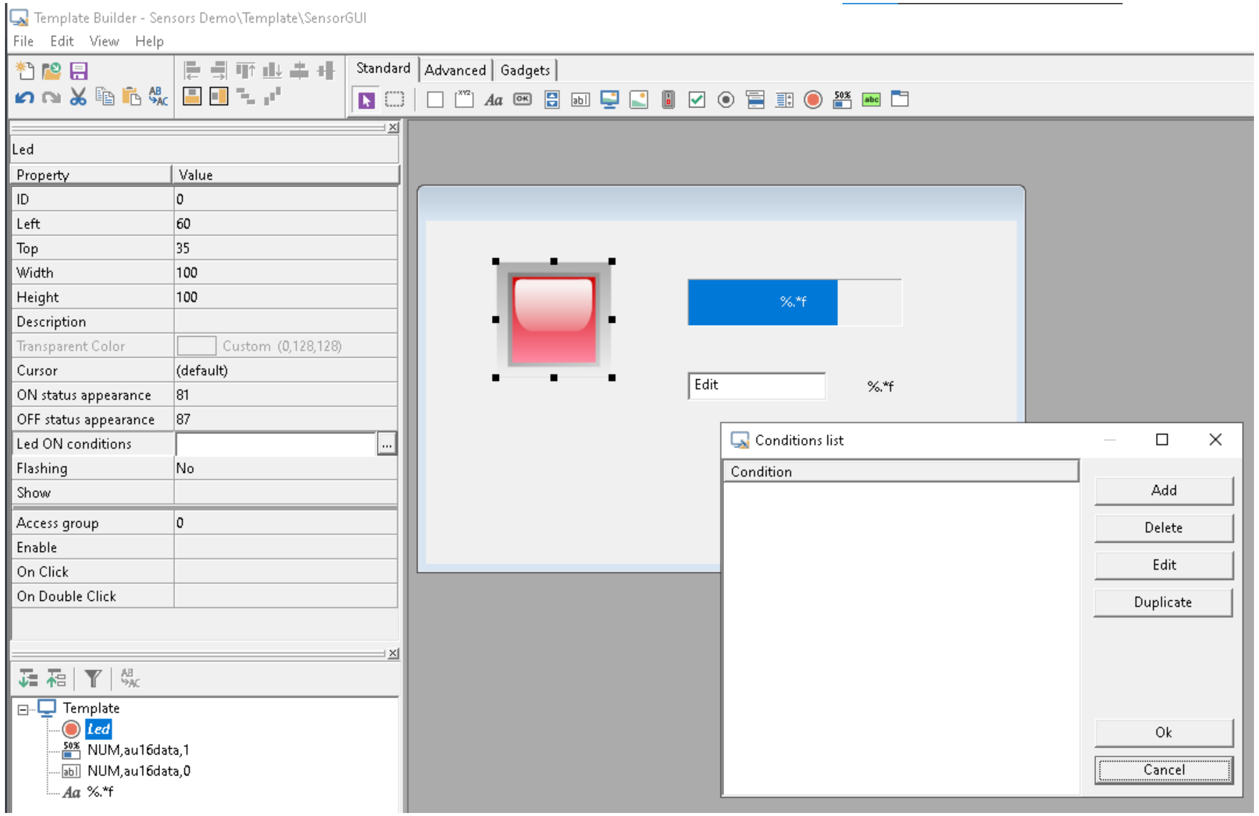
Gate: NUM,au16data,0

Operator: != (not equal to)

□ Negated condition

Value (numeric): 64

Template
Global Values Inheritance

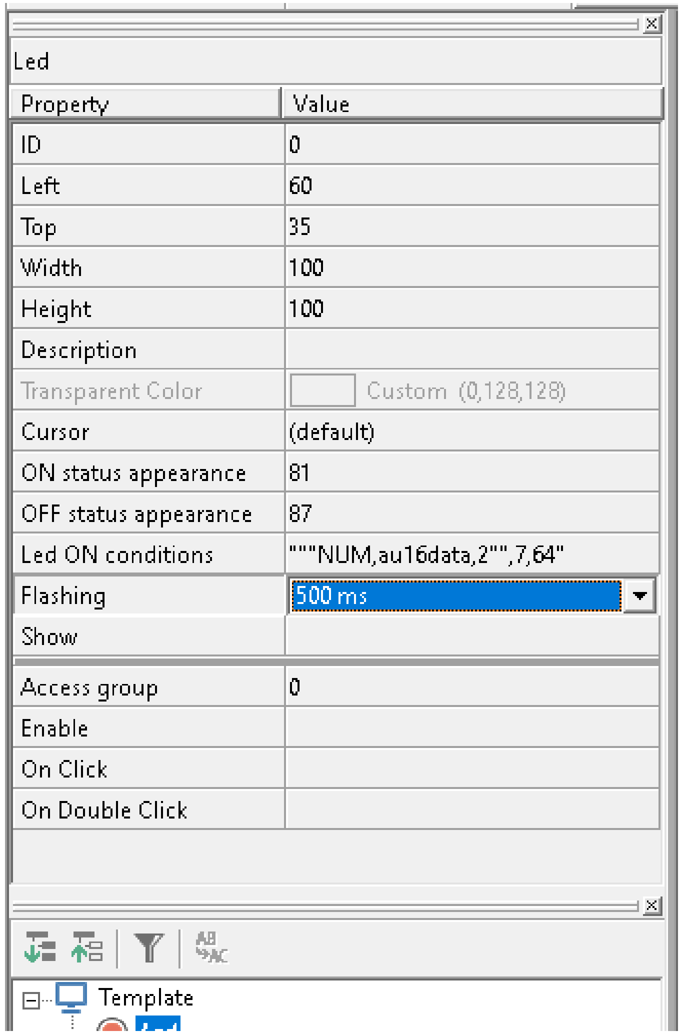


Led

Property	Value
ID	0
Left	60
Top	35
Width	100
Height	100
Description	
Transparent Color	Custom (0,128,128)
Cursor	(default)
ON status appearance	81
OFF status appearance	87
Led ON conditions	...
Flashing	No
Show	
Access group	0
Enable	
On Click	
On Double Click	

- Template
  - Led
  - NUM,au16data,1
  - NUM,au16data,0
  - %.*f

Condition

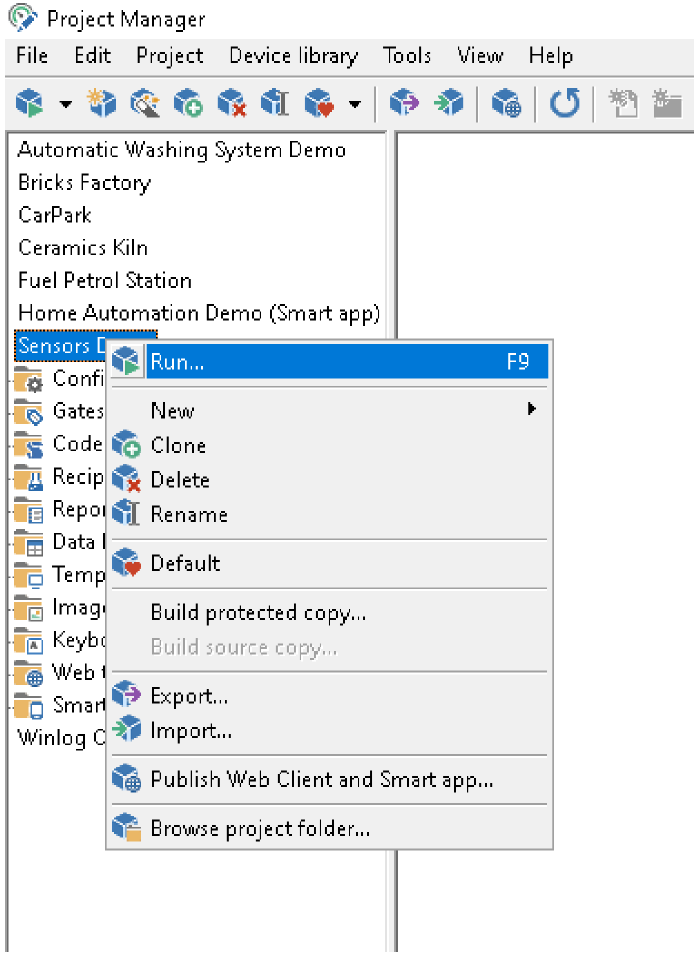


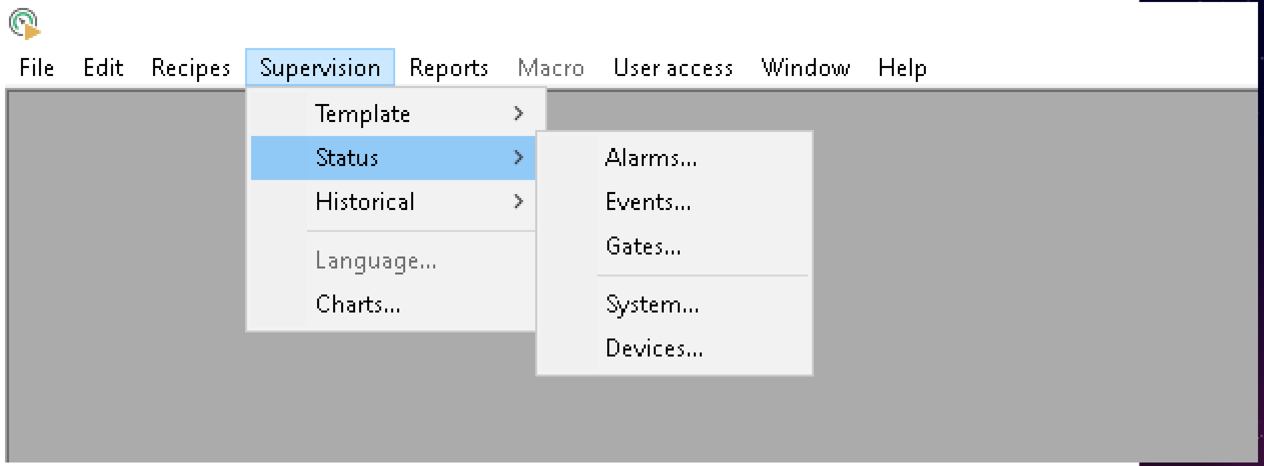
Led

Property	Value
ID	0
Left	60
Top	35
Width	100
Height	100
Description	
Transparent Color	Custom (0,128,128)
Cursor	(default)
ON status appearance	81
OFF status appearance	87
Led ON conditions	"""NUM,au16data,2""",7,64"
Flashing	500 ms
Show	
Access group	0
Enable	
On Click	
On Double Click	

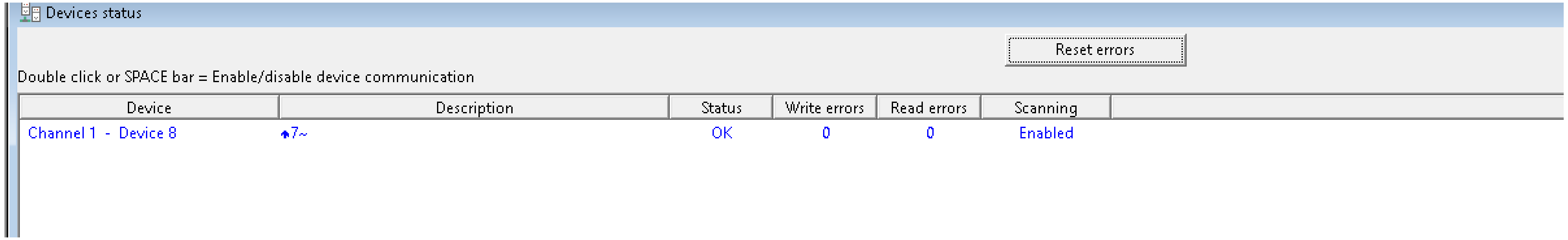
- Template

Template
Running your GUI





Template
Running your GUI

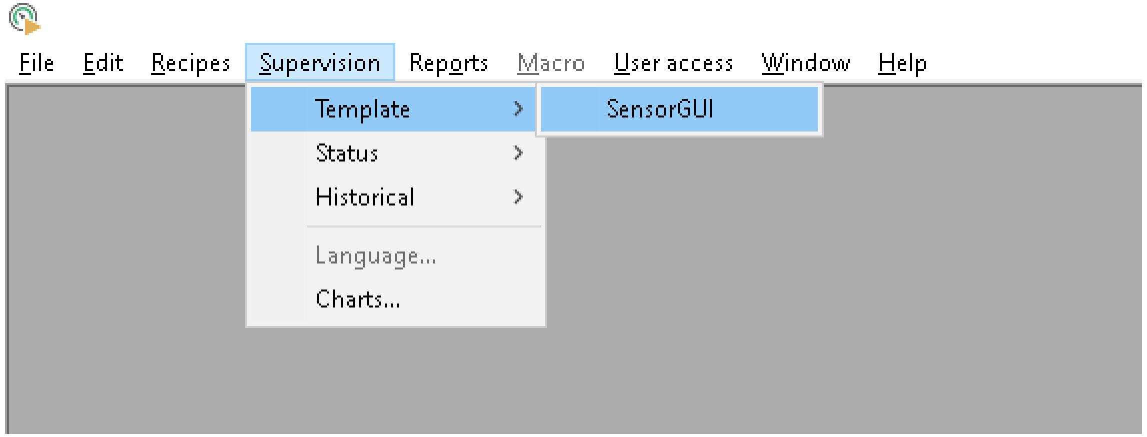


Devices status

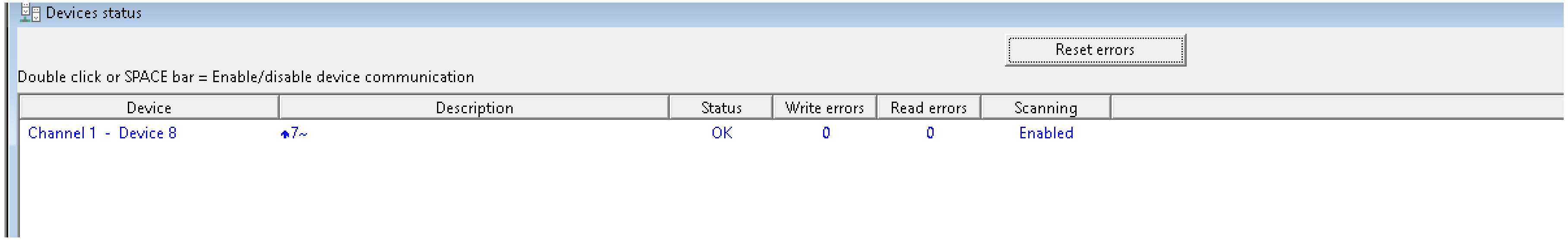
Reset errors

Double click or SPACE bar = Enable/disable device communication

Device	Description	Status	Write errors	Read errors	Scanning
Channel 1 - Device 8	↑7~	OK	0	0	Enabled



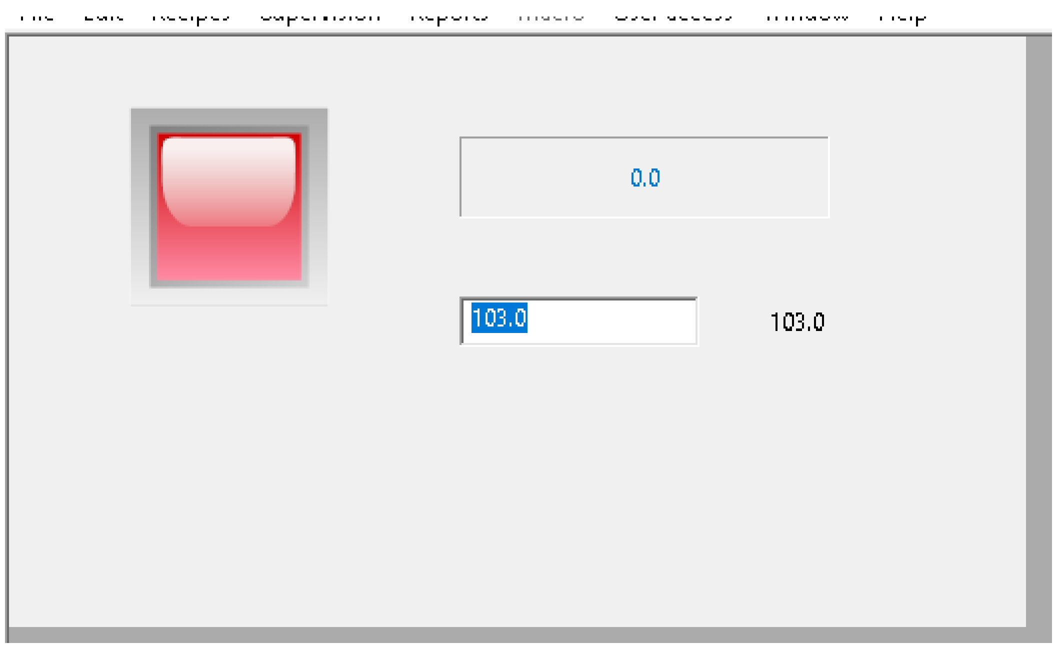
Template
Running your GUI



Devices status

Double click or SPACE bar = Enable/disable device communication

Device	Description	Status	Write errors	Read errors	Scanning
Channel 1 - Device 8	↑7~	OK	0	0	Enabled



File Edit Recipes Supervision Reports Macro User access Window Help
0.0
103.0
103.0