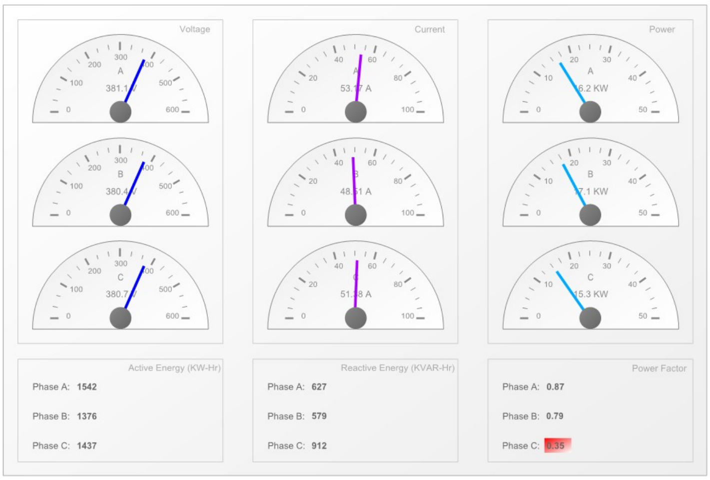
Application Note 002:

"Energy monitoring using the E10 analyzer module and S1000 Web HMI"

1 Introduction

This application note describes how to utilize the E10 energy analyzer, and how to create a Web-based HMI to display energy parameters.



Notice

You may download the project described in this application note, with all I/O, ST logic, and HMI parameters pre-configured, from www.inicotech.com.

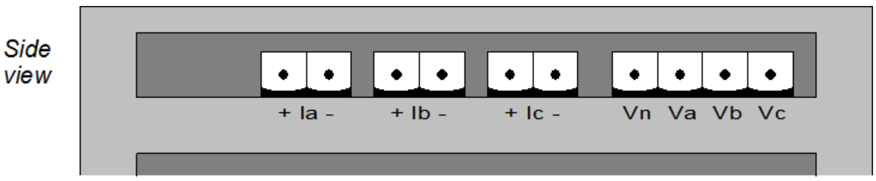
1.1 S1000 Smart RTU

This application note uses an S1000 Smart RTU equipped with an E10 Energy Analyzer expansion module. For ordering information, please visit www.inicotech.com.

1.2 Electrical setup

The energy consumption of a three-phase AC motor is measured by using:

- Direct connection of the 3 phases and neutral to the Vn, Va, Vb and Vc terminals of the E10 expansion module.
- Connection of 3 current transformers (100A/5A) to the +Ia-, +Ib- and +Ic- terminals



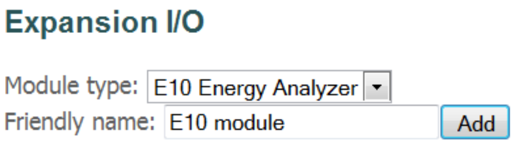
Warning

Never disconnect a current transformer from the +/- terminals unless the primary side has been disengaged. Connecting a current transformer with the secondary side left open will cause arching, and can cause injury and permanent damage to the equipment.

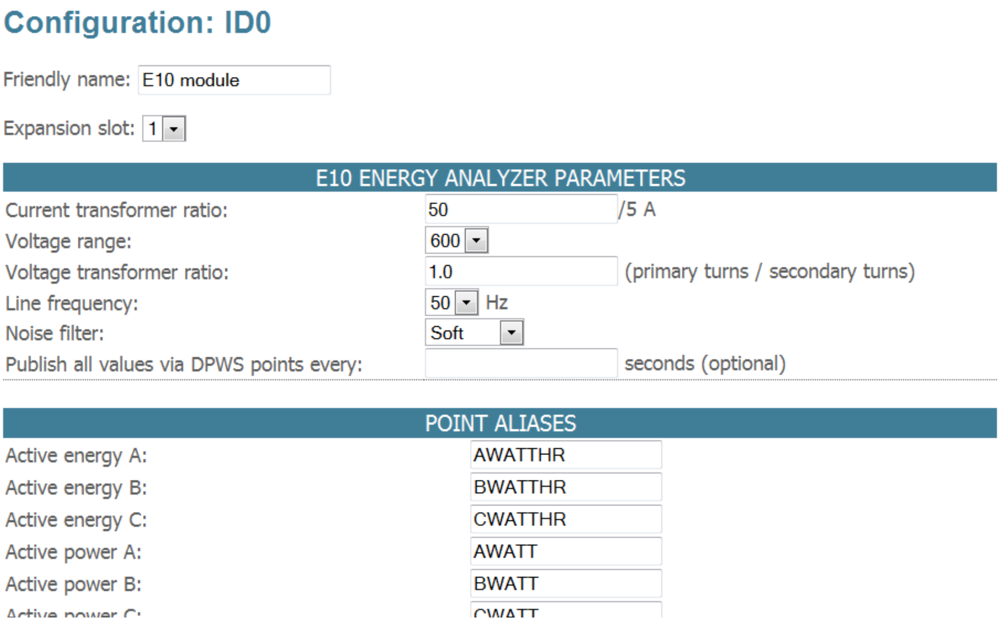
2 Energy measurement and analysis

2.1 I/O configuration

The first step is to setup the E10 expansion I/O module. To do this, add a new E10 module to the configuration:



Next, configure the applicable parameters, and give each measurement a meaningful alias. The default aliases are suitable for most applications.



2.2 Power factor calculation

The next step is to calculate the ***power factor*** for each phase. The power factor is defined as the ratio of active power to apparent power.

In order to perform these calculations, the steps to follow are:

- Add global variables to hold these values. Also, add variables to signal when the power factor is too low.

```
(* Power factor for each phase. *)
APF : real;
BPF : real;
CPF : real;

APF_low : bool;
BPF_low : bool;
CPF_low : bool;
```

- Add an ST program to calculate the values

```
PROGRAM power_factor

(* Calculate each power factor. *)
APF := AWATT / AVA;
BPF := BWATT / BVA;
CPF := CWATT / CVA;

(* Detect any low power factors. *)
IF APF < 0.5 THEN
  APF_LOW := true;
ELSE
  APF_LOW := false;
END_IF

IF BPF < 0.5 THEN
  BPF_LOW := true;
ELSE
  BPF_LOW := false;
END_IF

IF CPF < 0.5 THEN
  CPF_LOW := true;
ELSE
  CPF_LOW := false;
END_IF

END_PROGRAM
```

2.3 Value scaling

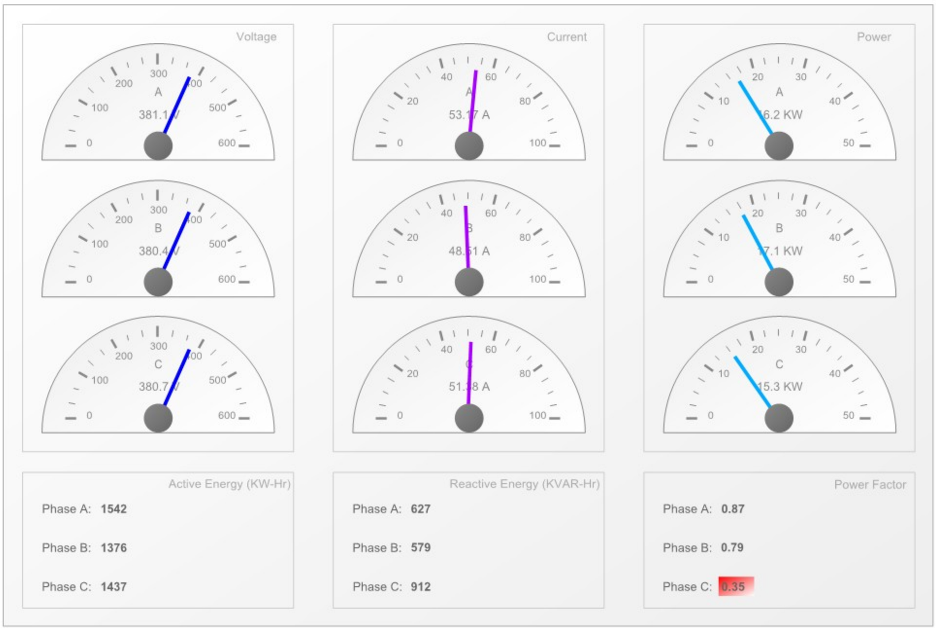
The next step is to scale the measured values to useful units. Often, values expressed in Watts are too large to be manageable, therefore it is more useful to express them as Kilowatts. The following Variables declaration and ST program takes care of this scaling.

```
(* Scaled power and energy values. *)

AKWATT : real;
BKWATT : real;
CKWATT : real;

AKWATTHR : real;
BKWATTHR : real;
CKWATTHR : real;

AKVARHR : real;
BKVARHR : real;
CKVARHR : real;
```

```
PROGRAM scale

(* Divide applicable values by 1000 to convert to Kilo-units. *)
AKWATT := AWATT / 1000.0;
BKWATT := BWATT / 1000.0;
CKWATT := CWATT / 1000.0;

AKWATTHR := AWATTHR / 1000.0;
BKWATTHR := BWATTHR / 1000.0;
CKWATTHR := CWATTHR / 1000.0;

AKVARHR := AVARHR / 1000.0;
BKVARHR := BVARHR / 1000.0;
CKVARHR := CVARHR / 1000.0;

END_PROGRAM
```

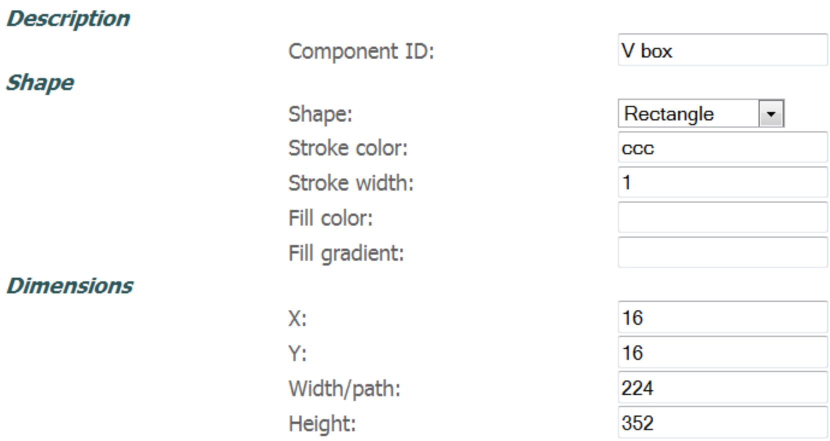
3 Web HMI Interface

The final stage in this application is to create a Web-based HMI page that will display the energy measurements. This section will describe step by step how to set up the following HMI page:



3.1 Display groups

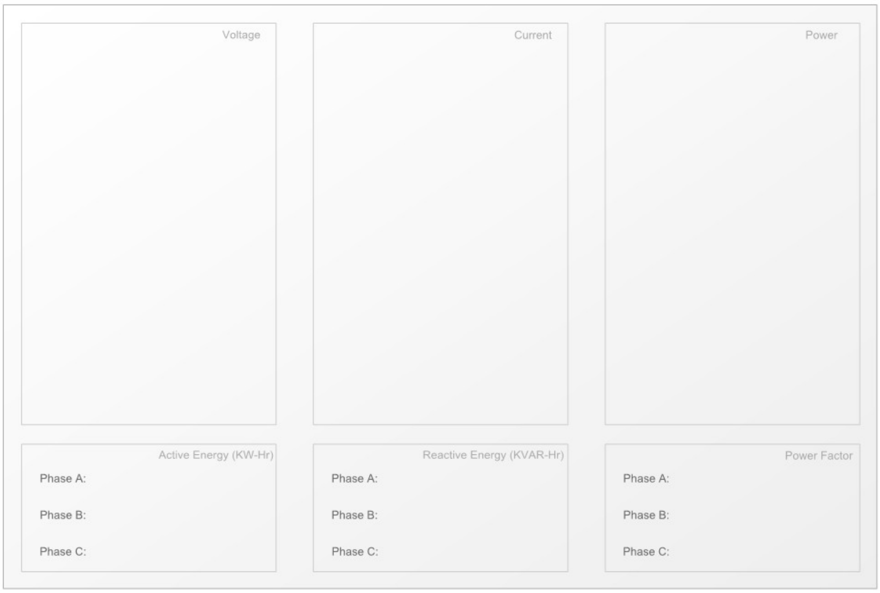
The first step is to display the sections where we will display the graphic components. Each group shall be bounded by a box and will have an identification label. The following screen capture shows suggested parameters for these boxes, which are implemented as *Shape* components of type *Rectangle*:



Likewise, the following capture shows suggested parameters for the *Label* components.

Description		
	Component ID:	V Label
Caption		
	Text:	Voltage
Dimensions		
	X:	192
	Y:	28
Font		
	Font:	Arial
	Font size:	10
	Font weight:	Normal
	Color:	aaa

Once the boxes and labels have been added, we may preview the design, which will show as follows.

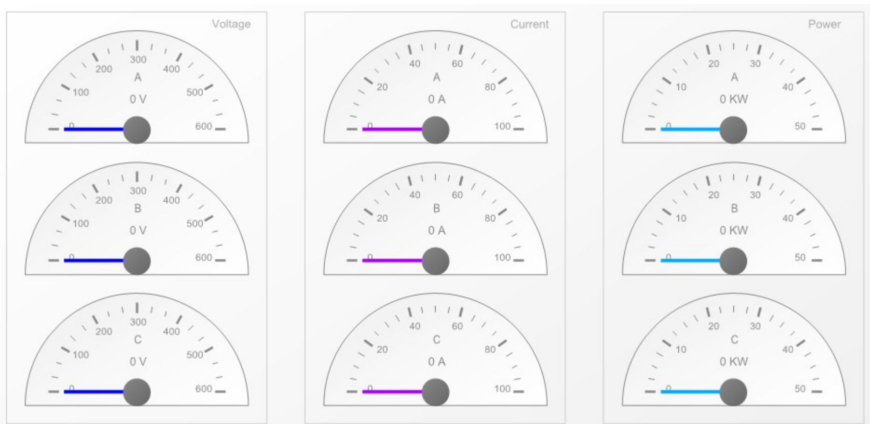


3.2 Gauges

The next step is to add the *Gauge* components which will display real-time current, voltage, and power measurements. The following capture shows suggested parameters for these gauges.

Description		
	Component ID:	AV gauge
Data source:		
	Alias:	AVRMS
	Data type:	Real
	Min value:	0
	Max value:	600
	Major tick:	100
	Minor tick:	33.5
Caption		
	Text:	A
	Units:	V
Shape		
	Shape:	Semi-circular gauge
	Border color:	888
	Border width:	1
	Fill color:	eee
	Fill gradient:	fff
	Needle color:	00f
	Needle width:	3
	Needle cap fill color:	888
	Needle cap gradient:	666
Dimensions		
	X:	32
	Y:	32
	Width:	192
	Height:	96
Font		
	Font:	Arial
	Font size:	10
	Font weight:	Normal
	Color:	888

Once all gauges have been added, we may preview the HMI Page:



3.3 Energy counters

The next step is to display counters for the accumulated active and reactive energy. These counters are implemented with *Value Text* components. The following screen capture shows suggested parameters for these components:

Description		
	Component ID:	AWH val
Data source:		
	Alias:	AKWATTHR
	Data type:	Integer
Dimensions		
	X:	80
	Y:	416
Font		
	Font:	Arial
	Font size:	10
	Font weight:	Bold
	Color:	666

After adding all counters, we may preview the page to ensure that the location on screen is correct:

3.4 Power factor display

The final step is to display the power factors for each phase. We shall use *Value Text* components to display the calculated power factors. We shall also use *Toggle Shapes* to display a warning indication when a power factor is too low. The Toggle Shapes are added first so that they are rendered *behind* the text. The following shows suggested parameters:

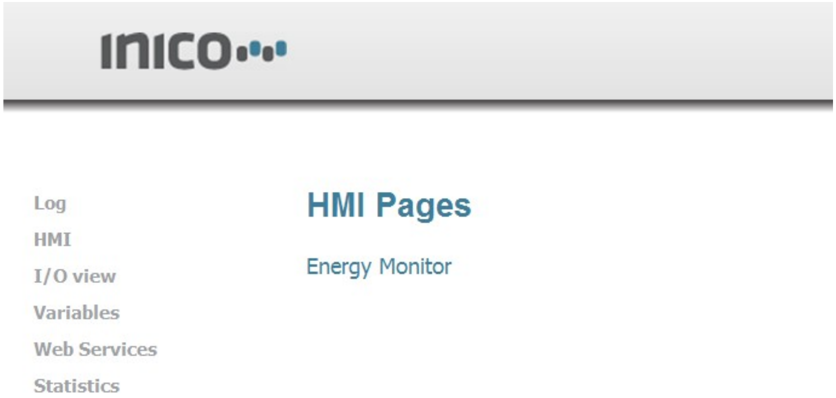
Description		
	Component ID:	APF box
Data source:		
	Alias:	APF_low
	Data type:	Boolean
Shape		
	Shape:	Rectangle
	Stroke color:	
	Stroke width:	
	Fill color:	f00
	Fill gradient:	fff
	Stroke color (true):	
	Stroke width (true):	
	Fill color (true):	
	Fill gradient (true):	
	Rotation (true):	
	Translation (true):	
Dimensions		
	X:	590
	Y:	406
	Width/path:	30
	Height:	16

Once the toggle shapes have been added, we may preview the display to ensure correct placement:

Once the placement is correct, the fill colors can be changed so that they are activated on the **true** state of the control variable.

3.5 Runtime display

Once the configuration is completed, the controller can be started in Run mode. The HMI display can be located in the list of HMI pages.



The final rendering will look as follows.