

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 Webbased 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 <u>www.inicotech.com</u>.

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 <u>www.inicotech.com</u>.

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:

Expansion I/O

Module type: E10 Energy Analyzer 💌		
Friendly name:	E10 module	Add

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

Configuration: ID0

		E 4 0 0 0 0
Friendiv	name	E10 module
riciary	name	L IV mouulo

Expansion slot: 1 -

E10 ENERGY ANALYZER PARAMETERS			
Current transformer ratio:	50	/5 A	
Voltage range:	600 -		
Voltage transformer ratio:	1.0	(primary turns / secondary turns)	
Line frequency:	50 🔹 Hz		
Noise filter:	Soft 🔹		
Publish all values via DPWS points every:		seconds (optional)	

	POINT ALIASES
Active energy A:	AWATTHR
Active energy B:	BWATTHR
Active energy C:	CWATTHR
Active power A:	AWATT
Active power B:	BWATT
Active power C.	CWATT

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.

```
3 (* Power factor for each phase. *)
4 APF : real;
5 BPF : real;
6 CPF : real;
7
8 APF_low : bool;
9 BPF_low : bool;
10 CPF_low : bool;
```



• Add an ST program to calculate the values

```
1 PROGRAM power_factor
2
3 (* Calculate each power factor. *)
4 APF := AWATT / AVA;
5 BPF := BWATT / BVA;
6 CPF := CWATT / CVA;
7
8 (* Detect any low power factors. *)
9 IF APF < 0.5 THEN
   APF_LOW := true;
10
11 ELSE
   APF LOW := false;
12
13 END_IF
14
15 IF BPF < 0.5 THEN
   BPF_LOW := true;
16
17 ELSE
18
   BPF_LOW := false;
19 END IF
20
21 IF CPF < 0.5 THEN
22
   CPF_LOW := true;
23 ELSE
   CPF LOW := false;
24
25 END_IF
26
27 END_PROGRAM
20
```

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.

<pre>(* Scaled power and energy values. *) AKWATT : real; BKWATT : real; CKWATT : real; AKWATTHR : real; CKWATTHR : real; CKWATTHR : real; AKWATHR : real; PROGRAM scale PROGRAM scale AKWATT := AWATT / 1000.0; BKWATT := CWATT / 1000.0; AKWATTHR := AWATTHR / 1000.0; BKWATT := CWATTHR / 1000.0; AKWATTHR := AWATTHR / 1000.0; AKWATTHR := AWATTHR / 1000.0; AKWATTHR := AWATHR / 1000.0; AKWATTHR := AWATHR / 1000.0; AKWATTHR := AWARHR / 1000.0; AKWATHR := AWARHR / 1000.0; AKWATTHR := AWARHR / 1000.0; AKWARTHR :</pre>		
<pre>13 14 AKWATT : real; 15 BKWATT : real; 16 CKWATT : real; 17 18 AKWATTHR : real; 20 CKWATTHR : real; 21 22 AKVARHR : real; 24 CKVARHR : real; 24 CKVARHR : real; 24 24 CKVARHR : real; 25 3 (* Divide applicable values by 1000 to convert to Kilo-unit AKWATT := AWATT / 1000.0; BKWATT := BWATT / 1000.0; 6 CKWATT := CWATT / 1000.0; 7 8 AKWATTHR := AWATTHR / 1000.0; 9 BKWATTHR := BWATTHR / 1000.0; 10 CKWATTHR := AVARHR / 1000.0; 11 12 AKVARHR := AVARHR / 1000.0; 13 24 25 25 25 25 25 25 25 25 25 25 25 25 25</pre>	12	(* Scaled power and energy values. *)
<pre>14 AKWATT : real; 15 BKWATT : real; 16 CKWATT : real; 17 18 AKWATTHR : real; 19 BKWATTHR : real; 20 CKWATTHR : real; 21 22 AKVARHR : real; 23 BKVARHR : real; 24 CKVARHR : real; 24 CKVARHR : real; 25 BKWATT := AWATT / 1000.0; 36 BKWATT := BWATT / 1000.0; 37 BKWATT := CWATT / 1000.0; 37 CKWATTHR := AWATTHR / 1000.0; 37 BKWATTHR := BWATTHR / 1000.0; 38 AKWATTHR := BWATTHR / 1000.0; 39 BKWATTHR := AVARHR / 1000.0; 31 AKWARTHR := AVARHR / 1000.0; 31 AKWARTHR := AVARHR / 1000.0; 31 AKWARTHR := AVARHR / 1000.0; 32 AKWARTHR := AVARHR / 1000.0; 33 AKWARTHR := AVARHR / 1000.0; 34 AKWARTHR := AVARHR / 1000.0; 35 BKWARTHR := AVARHR / 1000.0; 36 BKWARTHR := AVARHR / 1000.0; 37 AKWARTHR := AVARHR / 1000.0; 37 AKWARTHR := AVARHR / 1000.0; 38 AKWARTHR := AVARHR / 1000.0; 39 BKWARTHR := AVARHR / 1000.0; 30 AKWARTHR := AVARHR / 100</pre>	13	
<pre>16 BKWATT : real; 16 CKWATT : real; 17 18 AKWATTHR : real; 19 BKWATTHR : real; 20 CKWATTHR : real; 21 22 AKVARHR : real; 23 BKVARHR : real; 24 CKVARHR : real; 24 CKVARHR : real; 25 CKVARHR : real; 26 CKVARHR : real; 27 AKWATT := AWATT / 1000.0; 28 BKWATT := BWATT / 1000.0; 30 CKWATT := CWATT / 1000.0; 31 AKWATTHR := AWATTHR / 1000.0; 32 BKWATTHR := AVARHR / 1000.0; 33 AKWATTHR := AVARHR / 1000.0; 34 AKWATHR := AVARHR / 1000.0; 35 BKWATTHR := AVARHR / 1000.0; 36 BKWATHR := AVARHR / 1000.0; 37 AKWARHR := AVARHR / 1000.0; 37 AKWARHR := AVARHR / 1000.0; 38 AKWATHR := AVARHR / 1000.0; 39 BKWARTHR := AVARHR / 1000.0; 30 CKWATTHR := AVARHR / 1000.0; 31 AKVARHR := AVARHR / 1000.0; 32 BKVARHR := AVARHR / 1000.0; 33 AKWARHR := AVARHR / 1000.0; 34 AKWARHR := AVARHR / 1000.0; 35 AKWARHR := AVARHR / 1000.0; 36 AKWARHR := AVARHR / 1000.0; 37 AKVARHR := AVARHR / 1000.0; 37 AKVARHR := AVARHR / 1000.0; 37 AKVARHR := AVARHR / 1000.0; 38 AKWARHR := AVARHR / 1000.0; 39 AKWARHR := AVARHR / 1000.0; 30 AKWARHR := AVARHR /</pre>	14	AKWATT : real;
<pre>16 CKWATT : real; 17 18 AKWATTHR : real; 19 BKWATTHR : real; 20 CKWATTHR : real; 21 22 AKVARHR : real; 23 BKVARHR : real; 24 CKVARHR : real; 24 CKVARHR : real; 25 WAATT := AWATT / 1000.0; 26 BKWATT := BWATT / 1000.0; 37 BKWATTHR := AWATTHR / 1000.0; 38 AKWATTHR := AWATTHR / 1000.0; 39 BKWATTHR := BWATTHR / 1000.0; 30 CKWATT := CWATT / 1000.0; 31 CKWATTHR := AVARHR / 1000.0; 31 AKVARHR := AVARHR / 1000.0; 32 BKVARHR := AVARHR / 1000.0; 33 AKVARHR := AVARHR / 1000.0; 34 AKVARHR := AVARHR / 1000.0; 35 BKVARHR := AVARHR / 1000.0; 36 BKVARHR := AVARHR / 1000.0; 37 AKVARHR := AVARHR / 1000.0; 37 AKVARHR := AVARHR / 1000.0; 38 AKVARHR := AVARHR / 1000.0; 39 BKVARHR := AVARHR / 1000.0; 30 CKWATTHR := AVARHR / 1000.0; 30 CKWATTH</pre>	15	BKWATT : real;
<pre>17 18 AKWATTHR : real; 19 BKWATTHR : real; 20 CKWATTHR : real; 21 22 AKVARHR : real; 23 BKVARHR : real; 24 CKVARHR : real; 24 CKVARHR : real; 2 3 (* Divide applicable values by 1000 to convert to Kilo-unit 4 AKWATT := AWATT / 1000.0; 6 BKWATT := BWATT / 1000.0; 7 8 AKWATTHR := AWATTHR / 1000.0; 9 BKWATTHR := BWATTHR / 1000.0; 10 CKWATTHR := BWATTHR / 1000.0; 11 12 AKVARHR := AVARHR / 1000.0; 13 14 15 26 27 27 27 28 28 29 29 29 20 20 20 20 20 20 20 20 20 20 20 20 20</pre>	16	CKWATT : real;
<pre>18 AKWATTHR : real; 19 BKWATTHR : real; 20 CKWATTHR : real; 21 22 AKVARHR : real; 23 BKVARHR : real; 24 CKVARHR : real; 24 CKVARHR : real; 25 CKVARHR : real; 26 CKVARHR : real; 27 AKWATT := AWATT / 1000.0; 28 BKWATT := BWATT / 1000.0; 30 CKWATT := CWATT / 1000.0; 31 AKWATTHR := AWATTHR / 1000.0; 32 BKWATTHR := AVARHR / 1000.0; 33 AKWATTHR := AVARHR / 1000.0; 34 AKWATHR := AVARHR / 1000.0; 35 BKWATHR := AVARHR / 1000.0; 36 BKWATHR := AVARHR / 1000.0; 37 AKWARHR := AVARHR / 1000.0; 37 AKWARHR := AVARHR / 1000.0; 38 AKWATHR := AVARHR / 1000.0; 39 BKWARHP := AVARHR / 1000.0; 30 CKWATTHR := AVARHR / 1000.0; 30 CKWATHR := AVARHR / 1000.0; 30 CKWATHR := AVARHR / 1000.0; 31 AKVARHR := AVARHR / 1000.0; 32 BKVARHR := AVARHR / 1000.0; 33 AKWARHR := AVARHR / 1000.0; 34 AKWARHR := AVARHR / 1000.0; 35 AKWARHR := AVARHR / 1000.0; 36 AKWARHR := AVARHR / 1000.0; 37 AKVARHR := AVARHR / 1000.0; 38 AKWARHR := AVARHR / 1000.0; 39 AKWARHR := AVARHR / 1000.0; 30 AK</pre>	17	
<pre>19 BKWATTHR : real; 20 CKWATTHR : real; 21 22 AKVARHR : real; 23 BKVARHR : real; 24 CKVARHR : real; 24 CKVARHR : real; 3 (* Divide applicable values by 1000 to convert to Kilo-unit 4 AKWATT := AWATT / 1000.0; 6 BKWATT := BWATT / 1000.0; 7 BKWATTHR := AWATTHR / 1000.0; 9 BKWATTHR := BWATTHR / 1000.0; 10 CKWATTHR := CWATTHR / 1000.0; 11 12 AKVARHR := AVARHR / 1000.0; 13 BKWATHR := AVARHR / 1000.0; 14 DKVARHR := AVARHR / 1000.0; 15 DKVARHR := AVARHR / 1000.0; 16 DKVARHR := AVARHR / 1000.0; 17 DKVARHR := AVARHR / 1000.0; 18 DKVARHR := AVARHR / 1000.0; 19 DKVARHR := AVARHR / 1000.0; 10 DKVARHR := AVARHR / 1000.0; 11 DKVARHR := AVARHR / 1000.0; 12 DKVARHR := AVARHR / 1000.0; 13 DKVARHR := AVARHR / 1000.0; 14 DKVARHR := AVARHR / 1000.0; 15 DKVARHR := AVARHR / 1000.0; 16 DKVARHR := AVARHR / 1000.0; 17 DKVARHR := AVARHR / 1000.0; 18 DKVARHR := AVARHR / 1000.0; 19 DKVARHR := AVARHR / 1000.0; 10 DKVARHR := AVARHR / 1000.0; 11 DKVARHR := AVARHR / 1000.0; 12 DKVARHR := AVARHR / 1000.0; 13 DKVARHR := AVARHR / 1000.0; 14 DKVARHR := AVARHR / 1000.0; 15 DKVARHR := AVARHR / 1000.0; 16 DKVARHR := AVARHR / 1000.0; 17 DKVARHR := AVARHR / 1000.0; 18 DKVARHR := AVARHR / 1000.0; 19 DKVARHR := AVARHR / 1000.0; 10 DKVARHR := AVARHR / 1000.0; 11 DKVARHR := AVARHR / 1000.0; 12 DKVARHR := AVARHR / 1000.0; 13 DKVARHR := AVARHR / 1000.0; 14 DKVARHR := AVARHR / 1000.0; 15 DKVARHR := AVARHR / 1000.0; 17 DKVARHR := AVARHR / 1000.0; 18 DKVARHR := AVARHR / 1000.0; 19 DKVARHR := AVARHR / 1000.0; 10 DKVARHR := AVARHR / 1000.0;</pre>	18	AKWATTHR : real;
<pre>20 CKWATTHR : real; 21 22 AKVARHR : real; 23 BKVARHR : real; 24 CKVARHR : real; 24 CKVARHR : real; 3 (* Divide applicable values by 1000 to convert to Kilo-unit 4 AKWATT := AWATT / 1000.0; 5 BKWATT := BWATT / 1000.0; 6 CKWATT := CWATT / 1000.0; 7 AKWATTHR := AWATTHR / 1000.0; 9 BKWATTHR := BWATTHR / 1000.0; 10 CKWATTHR := AVARHR / 1000.0; 11 12 AKVARHR := AVARHR / 1000.0; 13 BKWATHR := AVARHR / 1000.0; 14 AKVARHR := AVARHR / 1000.0; 15 BKWAPHP := PWAPHP (1000.0; 16 BKWAPHP := PWAPHP (1000.0; 17 BKWAPHP := PWAPHP (1000.0; 18 BKWAPHP := PWAPHP (1000.0; 19 BKWAPHP := PWAPHP (1000.0; 10 BKWAPHP := PWAPHP (1000.0; 10 BKWAPHP := PWAPHP (1000.0; 10 BKWAPHP := PWAPHP (1000.0; 11 BKWAPHP := BWAPHP (1000.0; 12 BKWAPHP := PWAPHP (1000.0; 13 BKWAPHP := BWAPHP (1000.0; 14 BKWAPHP := BWAPHP (1000.0; 15 BKWAPHP := BWAPHP (1000.0; 16 BKWAPHP := BWAPHP (1000.0; 17 BKWAPHP := BWAPHP (1000.0; 18 BKWAPHP := BWAPHP (1000.0; 19 BKWAPHP := BWAPHP (1000.0; 10 BKWAPHP := BWAPHP (1000.0; 10 BKWAPHP := BWAPHP (1000.0; 11 BKWAPHP := BWAPHP (1000.0; 12 BKWAPHP := BWAPHP (1000.0; 13 BKWAPHP := BWAPHP (1000.0; 14 BKWAPHP := BWAPHP (1000.0; 15 BKWAPHP := BWAPHP (1000.0; 16 BKWAPHP := BWAPHP (1000.0; 17 BKWAPHP := BWAPHP (1000.0; 18 BKWAPHP := BWAPHP (1000.0; 19 BKWAPHP := BWAPHP (1000.0; 10 BKWAPHP := BWAPHP (</pre>	19	BKWATTHR : real;
<pre>21 22 AKVARHR : real; 23 BKVARHR : real; 24 CKVARHR : real; 24 CKVARHR : real; 1 PROGRAM scale 2 3 (* Divide applicable values by 1000 to convert to Kilo-unit 4 AKWATT := AWATT / 1000.0; 5 BKWATT := BWATT / 1000.0; 6 CKWATT := CWATT / 1000.0; 7 8 AKWATTHR := AWATTHR / 1000.0; 9 BKWATTHR := BWATTHR / 1000.0; 10 CKWATTHR := CWATTHR / 1000.0; 11 12 AKVARHR := AVARHR / 1000.0; 13 AKWARHR := AVARHR / 1000.0; 14 AKWARHR := AVARHR / 1000.0; 15 BKWARHR := AVARHR / 1000.0; 16 AKWARHR := AVARHR / 1000.0; 17 AKVARHR := AVARHR / 1000.0; 18 AKWARHR := AVARHR / 1000.0; 19 AKWARHR := AVARHR / 1000.0; 10 AKWARHR := AVARHR / 1000.0; 11 AKWARHR := AVARHR / 1000.0; 12 AKWARHR := AVARHR / 1000.0; 13 AKWARHR := AVARHR / 1000.0; 14 AKWARHR := AVARHR / 1000.0; 15 AKWARHR := AVARHR / 1000.0; 16 AKWARHR := AVARHR / 1000.0; 17 AKWARHR := AVARHR / 1000.0; 18 AKWARHR := AVARHR / 1000.0; 19 AKWARHR := AVARHR / 1000.0; 10 AKWARHR := AVARHR / 1000.0; 10 AKWARHR := AVARHR / 1000.0; 11 AKWARHR := AVARHR / 1000.0; 12 AKWARHR := AVARHR / 1000.0; 13 AKWARHR := AVARHR / 1000.0; 14 AKWARHR := AVARHR / 1000.0; 15 AKWARHR := AVARHR / 1000.0; 16 AKWARHR := AVARHR / 1000.0; 17 AKWARHR := AVARHR / 1000.0; 18 AKWARHR := AVARHR / 1000.0; 19 AKWARHR := AVARHR / 1000.0; 10 AKWARHR := AVARHR</pre>	20	CKWATTHR : real;
<pre>22 AKVARHR : real; 23 BKVARHR : real; 24 CKVARHR : real; 25 PROGRAM scale 2 3 (* Divide applicable values by 1000 to convert to Kilo-unit 4 AKWATT := AWATT / 1000.0; 6 BKWATT := BWATT / 1000.0; 7 BKWATTHR := AWATTHR / 1000.0; 9 BKWATTHR := BWATTHR / 1000.0; 10 CKWATTHR := CWATTHR / 1000.0; 11 AKVARHR := AVARHR / 1000.0; 12 AKVARHR := AVARHR / 1000.0; 13 BKWATHR := AVARHR / 1000.0; 14 AKVARHR := AVARHR / 1000.0; 15 BKWARHR := AVARHR / 1000.0; 16 BKWARHR := AVARHR / 1000.0; 17 AKVARHR := AVARHR / 1000.0; 18 AKWARHR := AVARHR / 1000.0; 19 BKWARHR := AVARHR / 1000.0; 10 BKWARHR := AVARHR / 1000.0; 10 BKWARHR := AVARHR / 1000.0; 11 AKVARHR := AVARHR / 1000.0; 12 AKVARHR := AVARHR / 1000.0; 13 AKWARHR := AVARHR / 1000.0; 14 AKWARHR := AVARHR / 1000.0; 15 BKWARHR := AVARHR / 1000.0; 16 BKWARHR := AVARHR / 1000.0; 17 AKVARHR := AVARHR / 1000.0; 18 AKWARHR := AVARHR / 1000.0; 19 BKWARHR := AVARHR / 1000.0; 10 BKWARHR := AVARHR / 1000.0; 10 BKWARHR := AVARHR / 1000.0; 11 AKWARHR := AVARHR / 1000.0; 12 AKVARHR := AVARHR / 1000.0; 13 AKWARHR := AVARHR / 1000.0; 14 AKWARHR := AVARHR / 1000.0; 15 BKWARHR := AVARHR / 1000.0; 16 BKWARHR := AVARHR / 1000.0; 17 AKVARHR := AVARHR / 1000.0; 18 AKWARHR := AVARHR / 1000.0; 19 BKWARHR := AVARHR / 1000.0; 10 BKWARHR :=</pre>	21	
<pre>23 BKVARHR : real; 24 CKVARHR : real; 1 PROGRAM scale 2 (* Divide applicable values by 1000 to convert to Kilo-unit 4 AKWATT := AWATT / 1000.0; 6 BKWATT := BWATT / 1000.0; 7 CKWATT := CWATT / 1000.0; 9 BKWATTHR := AWATTHR / 1000.0; 10 CKWATTHR := CWATTHR / 1000.0; 11 AKVARHR := AVARHR / 1000.0; 12 AKVARHR := AVARHR / 1000.0; 13 BKWATHR := AVARHR / 1000.0; 14 AKWARHR := AVARHR / 1000.0; 15 BKWARHR := AVARHR / 1000.0; 16 BKWARHR := AVARHR / 1000.0; 17 BKWARHR := AVARHR / 1000.0; 18 BKWARHR := AVARHR / 1000.0; 19 BKWARHR := AVARHR / 1000.0; 10 BKWARHR := AVARHR / 1000.0; 11 BKWARHR := AVARHR / 1000.0; 12 AKVARHR := AVARHR / 1000.0; 13 BKWARHR := AVARHR / 1000.0; 14 BKWARHR := AVARHR / 1000.0; 15 BKWARHR := AVARHR / 1000.0; 16 BKWARHR := AVARHR / 1000.0; 17 BKWARHR := AVARHR / 1000.0; 18 BKWARHR := AVARHR / 1000.0; 19 BKWARHR := AVARHR / 1000.0; 10 BKWARHR := AVARHR / 1000.0; 10 BKWARHR := AVARHR / 1000.0; 11 BKWARHR := AVARHR / 1000.0; 12 BKWARHR := AVARHR / 1000.0; 13 BKWARHR := AVARHR / 1000.0; 14 BKWARHR := AVARHR / 1000.0; 15 BKWARHR := AVARHR / 1000.0; 16 BKWARHR := AVARHR / 1000.0; 17 BKWARHR := AVARHR / 1000.0; 18 BKWARHR := AVARHR / 1000.0; 19 BKWARHR := AVARHR / 1000.0; 10 BKWAR</pre>	22	AKVARHR : real;
<pre>24 CKVARHR : real; 1 PROGRAM scale 2 (* Divide applicable values by 1000 to convert to Kilo-unit 4 AKWATT := AWATT / 1000.0; 6 BKWATT := BWATT / 1000.0; 7 CKWATT := CWATT / 1000.0; 9 BKWATTHR := AWATTHR / 1000.0; 10 CKWATTHR := CWATTHR / 1000.0; 11 AKVARHR := AVARHR / 1000.0; 12 AKVARHR := AVARHR / 1000.0; 13 BKWATHR := AVARHR / 1000.0; 14 AKWARHR := AVARHR / 1000.0; 15 BKWARHR := AVARHR / 1000.0; 16 BKWARHR := AVARHR / 1000.0; 17 BKWARHR := AVARHR / 1000.0; 18 BKWARHR := AVARHR / 1000.0; 19 BKWARHR := AVARHR / 1000.0; 10 BKWARHR := AVARHR / 1000.0; 11 BKWARHR := AVARHR / 1000.0; 12 AKWARHR := AVARHR / 1000.0; 13 BKWARHR := AVARHR / 1000.0; 14 BKWARHR := AVARHR / 1000.0; 15 BKWARHR := AVARHR / 1000.0; 16 BKWARHR := AVARHR / 1000.0; 17 BKWARHR := AVARHR / 1000.0; 18 BKWARHR := AVARHR / 1000.0; 19 BKWARHR := AVARHR / 1000.0; 10 BKWARHR := AVARHR / 1000.0; 10 BKWARHR := AVARHR / 1000.0; 11 BKWARHR := AVARHR / 1000.0; 12 BKWARHR := AVARHR / 1000.0; 13 BKWARHR := AVARHR / 1000.0; 14 BKWARHR := AVARHR / 1000.0; 15 BKWARHR := AVARHR / 1000.0; 16 BKWARHR := AVARHR / 1000.0; 17 BKWARHR := AVARHR / 1000.0; 18 BKWARHR := AVARHR / 1000.0; 19 BKWARHR := AVARHR / 1000.0; 10 BKWARHR := AVARHR / 1000</pre>	23	BKVARHR : real;
<pre> 1 PROGRAM scale 2 3 (* Divide applicable values by 1000 to convert to Kilo-unit 4 AKWATT := AWATT / 1000.0; 6 BKWATT := BWATT / 1000.0; 7 CKWATT := CWATT / 1000.0; 9 BKWATTHR := AWATTHR / 1000.0; 10 CKWATTHR := CWATTHR / 1000.0; 11 2 AKVARHR := AVARHR / 1000.0; 12 BKVAPHP := BVAPHP / 1000.0; 13 BKVAPHP := BVAPHP / 1000.0; 14 BKVAPHP := BVAPHP / 1000.0; 15 BKVAPHP := BVAPHP / 1000.0; 16 BKVAPHP := BVAPHP / 1000.0; 17 BKVAPHP := BVAPHP / 1000.0; 18 BKVAPHP := BVAPHP / 1000.0; 19 BKVAPHP := BVAPHP / 1000.0; 10 BKVAPHP := BVAPHP / 1000.0; 11 BKVAPHP := BVAPHP / 1000.0; 12 BKVAPHP := BVAPHP / 1000.0; 13 BKVAPHP := BVAPHP / 1000.0; 14 BKVAPHP := BVAPHP / 1000.0; 15 BKVAPHP := BVAPHP / 1000.0; 16 BKVAPHP := BVAPHP / 1000.0; 17 BKVAPHP := BVAPHP / 1000.0; 18 BKVAPHP := BVAPHP / 1000.0; 19 BKVAPHP := BVAPHP / 1000.0; 10 BKVAPHP := BVAPHP / 1000.0; 10 BKVAPHP := BVAPHP / 1000.0; 10 BKVAPHP := BVAPHP / 1000.0; 11 BKVAPHP := BVAPHP / 1000.0; 12 BKVAPHP := BVAPHP / 1000.0; 13 BKVAPHP := BVAPHP / 1000.0; 14 BKVAPHP := BVAPHP / 1000.0; 15 BKVAPHP := BVAPHP / 1000.0; 16 BKVAPHP := BVAPHP / 1000.0; 17 BKVAPHP := BVAPHP / 1000.0; 18 BKVAPHP := BVAPHP / 1000.0; 19 BKVAPHP := BVAPHP / 1000.0; 10 BKVAPHP</pre>	24	CKVARHR : real;
<pre>PROGRAM scale (* Divide applicable values by 1000 to convert to Kilo-unit AKWATT := AWATT / 1000.0; BKWATT := BWATT / 1000.0; CKWATT := CWATT / 1000.0; BKWATTHR := AWATTHR / 1000.0; BKWATTHR := BWATTHR / 1000.0; AKWATTHR := AVARHR / 1000.0; AKVARHR := AVARHR / 1000.0; BKVARHR := AVARHR / 1000.0; AKVARHR := AVARHR / 1000.0; AKVARHR</pre>		-
<pre> 2 2 3 (* Divide applicable values by 1000 to convert to Kilo-unit 4 AKWATT := AWATT / 1000.0; 5 BKWATT := BWATT / 1000.0; 6 CKWATT := CWATT / 1000.0; 7 8 AKWATTHR := AWATTHR / 1000.0; 9 BKWATTHR := BWATTHR / 1000.0; 10 CKWATTHR := CWATTHR / 1000.0; 11 2 AKVARHR := AVARHR / 1000.0; 12 BKWAPHP := BWAPHP / 1000.0; 13 BKWAPHP := BWAPHP / 1000.0; 14 BKWAPHP := BWAPHP / 1000.0; 15 BKWAPHP := BWAPHP / 1000.0; 16 BKWAPHP := BWAPHP / 1000.0; 17 BKWAPHP := BWAPHP / 1000.0; 18 BKWAPHP := BWAPHP / 1000.0; 19 BKWAPHP := BWAPHP / 1000.0; 10 BKWAPHP := BWAPHP / 1000.0; 11 BKWAPHP := BWAPHP / 1000.0; 12 BKWAPHP := BWAPHP / 1000.0; 13 BKWAPHP := BWAPHP / 1000.0; 14 BKWAPHP := BWAPHP / 1000.0; 15 BKWAPHP := BWAPHP / 1000.0; 16 BKWAPHP := BWAPHP / 1000.0; 17 BKWAPHP := BWAPHP / 1000.0; 18 BKWAPHP := BWAPHP / 1000.0; 19 BKWAPHP := BWAPHP / 1000.0; 10 BKWAPHP := BWAPHP / 1000.0; 10 BKWAPHP := BWAPHP / 1000.0; 11 BKWAPHP := BWAPHP / 1000.0; 12 BKWAPHP := BWAPHP / 1000.0; 13 BKWAPHP := BWAPHP / 1000.0; 14 BKWAPHP := BWAPHP / 1000.0; 14 BKWAPHP := BWAPHP / 1000.0; 15 BKWAPHP := BWAPHP / 1000.0; 16 BKWAPHP := BWAPHP / 1000.0; 17 BKWAPHP := BWAPHP / 1000.0; 18 BKWAPHP := BWAPHP / 1000.0; 19 BKWAPHP := BWAPHP / 1000.0; 10 BKWAPHP := BKWAPHP / 1000.0; 10 BKWAPHP := BKWAPHP / 1000.0; 10 BKWAPHP := BKWAPHP / 1000.0; 10 BKWAPHP := BKWPHP / 1000.0; 10 BKWAPHP := BKWPHP / 1000.0; 10 BKWAPHP := BKWPHP / 1000.0; 10 BKWAPHP := BKWPH</pre>		DROCHAN and a
<pre> (* Divide applicable values by 1000 to convert to Kilo-unit 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; AKVARHR := AVARHR / 1000.0; BKWATHR := AVARHR / 1000.0; BKWATHR := AVARHR / 1000.0; AKVARHR := AVARHR / 1000.0; AKVARHR := AVARHR / 1000.0; AKVARHR := AVARHR / 1000.0; BKWATHR := AVARHR / 1000.0; </pre>	1	PROGRAM SCALE
<pre>A 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; BKWATHR := AVARHR / 1000.0; BKWAPHP := BWAPHP (1000.0; BKWAPHP (1000.0; BKWAPHP</pre>	4	(* Divide applicable values by 1000 to convert to Kilo-units *)
<pre>4 AKWATI := AWATI / 1000.0; 6 BKWATT := BWATT / 1000.0; 7 CKWATT := CWATT / 1000.0; 9 BKWATTHR := AWATTHR / 1000.0; 10 CKWATTHR := CWATTHR / 1000.0; 11 12 AKVARHR := AVARHR / 1000.0; 13 14 15 16 17 18 19 10 10 10 10 10 10 10 10 10 10</pre>	3	AKWAMM - AWAMM / 1000 0.
<pre>6 CKWATT := CWATT / 1000.0; 6 CKWATT := CWATT / 1000.0; 7 8 AKWATTHR := AWATTHR / 1000.0; 9 BKWATTHR := BWATTHR / 1000.0; 10 CKWATTHR := CWATTHR / 1000.0; 11 12 AKVARHR := AVARHR / 1000.0; 13 BKWAPHP := BWAPHP / 1000.0;</pre>	5	PKWATT - PWATT / 1000.0,
<pre>AKWATTHR := AWATTHR / 1000.0; BKWATTHR := BWATTHR / 1000.0; CKWATTHR := CWATTHR / 1000.0; AKVARHR := AVARHR / 1000.0; AKVARHR := AVARHR / 1000.0; BKWAPHP := BWAPHP (1000.0;</pre>	8	CKWATT := CWATT / 1000.0;
<pre>8 AKWATTHR := AWATTHR / 1000.0; 9 BKWATTHR := BWATTHR / 1000.0; 10 CKWATTHR := CWATTHR / 1000.0; 11 12 AKVARHR := AVARHR / 1000.0; 12 BKVARHR := BVARHR / 1000.0;</pre>	7	enwhit .= ewhit / 1000.0/
9 BKWATTHR := BWATTHR / 1000.0; 10 CKWATTHR := CWATTHR / 1000.0; 11 12 AKVARHR := AVARHR / 1000.0; 12 BKWAPHP := BWAPHP (1000.0;	é	AKWATTHR := AWATTHR / 1000.0:
10 CKWATTHR := CWATTHR / 1000.0; 11 12 AKVARHR := AVARHR / 1000.0; 13 BKVARHR := BVARHR / 1000.0;	a	BKWATTHR := BWATTHR / 1000.0:
11 12 AKVARHR := AVARHR / 1000.0; 13 BKVARHR := BVARHR / 1000.0;	ň	CKWATTHR := CWATTHR / 1000.0;
2 AKVARHR := AVARHR / 1000.0;	11	
(2) REVARUE - RVARUE (1000 0.	2	AKVARHR := AVARHR / 1000.0;
15 DAVARAR DVARAR / 1000.0;	з	BKVARHR := BVARHR / 1000.0;

14 CKVARHR := CVARHR / 1000.0;

15

47

16 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

Description

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*:

2 Courperon		
	Component ID:	V box
Shape		
	Shape:	Rectangle -
	Stroke color:	CCC
	Stroke width:	1
	Fill color:	
	Fill gradient:	
Dimensions		
	X:	16
	Y:	16
	Width/path:	224
	Height:	352

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

INICO

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.

Voltage	Current	Power
Active Energy (KW-Hr)	Reactive Energy (KVAR-Hr)	Power Factor
Phase A:	Phase A:	Phase A:
Phase B:	Phase B:	Phase B:
Phase C:	Phase C:	Phase C:

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:	OOf
	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:

Active Energy (KW-Hr)	Reactive Energy (KVAR-Hr)
Phase A: (var value)	Phase A: (var value)
Phase B: (var value)	Phase B: (var value)
Phase C: (var value)	Phase C: (var value)

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:

Data source:APF boxAlias:APF_lowData type:Boolean •ShapeStroke color:Stroke color:Image: Stroke color:Stroke width:Image: Stroke width:Fill color:f00Fill gradient:fffStroke color (true):Image: Stroke width (true):Fill gradient (true):Image: Stroke width (true):Fill color (true):Image: Stroke width (true):Fill gradient (true):Image: Stroke (true):	Description		
Data source:Alias:APF_lowData type:Boolean •ShapeRectangle •Shape:Stroke color:Stroke color:5troke width:Fill color:f00Fill gradient:fffStroke color (true):1Stroke width (true):1Fill color (true):1Fill gradient (true):1Fill gradient (true):1Fill gradient (true):1Fill gradient (true):16		Component ID:	APF box
Alias: APF_low Data type: Boolean • Shape: Rectangle • Stroke color: • Stroke color: • Stroke width: • Fill color: f00 Fill gradient: fff Stroke color (true): • Stroke color (true): • Stroke width (true): • Fill gradient (true): • Translation (true): • V: \$90 Y: 406 Width/path: 30 Height: 16	Data source:		
Data type: Boolean Shape: Rectangle Shape: Rectangle Stroke color: Image: Stroke color: Stroke width: Image: Stroke width: Fill color: f00 Fill gradient: fff Stroke color (true): Image: Stroke width (true): Fill color (true): Image: Stroke width (true): Fill color (true): Image: Stroke width (true): Fill gradient (true): Image: Stroke width (true): Fill gradient (true): Image: Stroke width (true): Fill gradient (true): Image: Stroke width (true): Translation (true): Image: Stroke width (true): V: Stroke width (true): Y: 406 Width/path: 30 Height: 16		Alias:	APF_low
ShapeRectangleStroke color:Stroke width:Fill color:f00Fill gradient:fffStroke color (true):Stroke width (true):Fill color (true):Fill gradient (true):Fill gradient (true):Translation (true):DimensionsX:X:590Y:406Width/path:30Height:16		Data type:	Boolean -
Shape: Rectangle Stroke color:	Shape		
Stroke color:Stroke width:Fill color:f00Fill gradient:fffStroke color (true):Image: Stroke width (true):Stroke width (true):Image: Stroke width (true):Fill color (true):Image: Stroke width (true):Fill gradient (true):Image: Stroke width (true):Fill gradient (true):Image: Stroke width (true):Translation (true):Image: Stroke width (true):DimensionsX:S90Y:406Width/path:30Height:16		Shape:	Rectangle 💌
Stroke width:f00Fill color:f00Fill gradient:fffStroke color (true):		Stroke color:	
Fill color: f00 Fill gradient: fff Stroke color (true):		Stroke width:	
Fill gradient:fffStroke color (true):		Fill color:	f00
Stroke color (true):Image: Stroke width (true):Stroke width (true):Image: Stroke width (true):Fill color (true):Image: Stroke width (true):Fill gradient (true):Image: Stroke width (true):Translation (true):Image: Stroke width (true):DimensionsX:X:590Y:406Width/path:30Height:16		Fill gradient:	fff
Stroke width (true):Image: Stroke width (true):Fill color (true):Image: Stroke width (true):Fill gradient (true):Image: Stroke width (true):Translation (true):Image: Stroke width (true):DimensionsStroke width (true):X:590Y:406Width/path:30Height:16		Stroke color (true):	
Fill color (true):Image: Second s		Stroke width (true):	
Fill gradient (true): Image: Constraint of true): Dimensions Image: Constraint of true): X: 590 Y: 406 Width/path: 30 Height: 16		Fill color (true):	
Rotation (true): Image: Constraint of true Dimensions X: 590 X: 590 Y: 406 Width/path: 30 Height: 16		Fill gradient (true):	
Translation (true):		Rotation (true):	
Dimensions X: 590 Y: 406 Width/path: 30 Height: 16		Translation (true):	
X: 590 Y: 406 Width/path: 30 Height: 16	Dimensions		
Y: 406 Width/path: 30 Height: 16		Х:	590
Width/path:30Height:16		Y:	406
Height: 16		Width/path:	30
		Height:	16

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



	Power Factor
Phase A:	
Phase B:	
Phase C:	

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.

