

Project Documentation

File: EMS_Bachelor_2.0.ecp

Date: 11/28/2022

Profile: e!COCKPIT

Table of contents

1	DUT: Load	3
2	DUT: Temp	3
3	Global Variable List: GVL	3
4	POU: Capacity_Estimator_100Ah	4
5	POU: Capacity_Estimator_23Ah	5
6	POU: Capacity_Selector	7
7	POU: Conversion	8
8	POU: Hovedprogram	10
9	POU: Power_Time	16
10	POU: Volt_Average	18
11	POU: V_DC_Selector	19
12	POU: Wh_In_Out	20

1 DUT: Load

```

1     TYPE Load :
2     (
3         Standby , Low , Medium , High
4     ) ;
5     END_TYPE
6

```

2 DUT: Temp

```

1     TYPE temp :
2     STRUCT
3         decimal : REAL ;
4         integer : WORD ;
5     END_STRUCT
6     END_TYPE
7

```

3 Global Variable List: GVL

```

1     //{attribute 'qualified_only'}
2     VAR_GLOBAL
3         V_DC_RAW AT %IW13 : INT ;           // Spenningsmåler
4
5         A_DC_BAT_RAW AT %IW1 : INT ;       // Strømtrafo 5 batteri ut
6         A_DC_EMS_RAW AT %IW2 : INT ;      // Strømtrafo 4 EMS supply
7         A_AC_IN_RAW AT %IW3 : INT ;       // Strømtrafo 2 AC inn inverter
8         A_AC_OUT_RAW AT %IW4 : INT ;      // Strømtrafo 1 ut til forbruker
9         A_DC_SUN_RAW AT %IW5 : INT ;      // Strømtrafo 3 solceller
10        A_DC_BAT_RAW_IN AT %IW8 : INT ;    //Strømtrafo 6 batteri inn
11
12        ShorePower AT %IW6 : INT ;         // Relé land.
13        GenPower AT %IW7 : INT ;          // Relé generator
14
15        K2_Shore AT %QX0.0 : BOOL ;        //Landstrøm kontaktor
16        K1_Gen AT %QX0.1 : BOOL ;         //Generator kontaktor
17        K3_Sol AT %QX0.4 : BOOL ;         //Solcelle kontaktor
18        K4_User AT %QX0.5 : BOOL ;        //Forbruker 1 kontaktor
19        K5_User AT %QX0.6 : BOOL ;        //Forbruker 2 kontaktor
20        K6_User AT %QX0.7 : BOOL ;        //Forbruker 3 kontaktor
21
22        Servo AT %QX0.2 : BOOL ;           // Relé choke
23        GenStop AT %QX1.6 : BOOL ;         // Relé stopp
24        GenStart AT %QX1.7 : BOOL ;       // Relé start
25
26        GenTemp : temp ;                   // PT1000 på generator
27        BatTemp : temp ;                   // PT1000 på batteri
28
29        Volt_Max : REAL := 12.75 ;         //Max spenning man kan beregne EOT
30        Volt_Min : REAL := 11.0 ;         //Laveste batterispenning før man

```

```

31      kobler inn generator
32      Battery100 : BOOL ;
33      Battery235 : BOOL ;
34
35      END_VAR
36

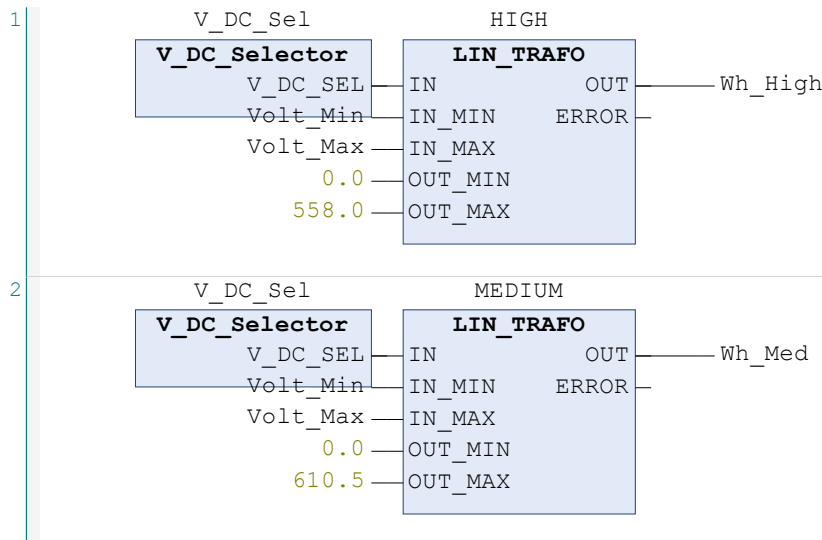
```

4 POU: Capacity_Estimator_100Ah

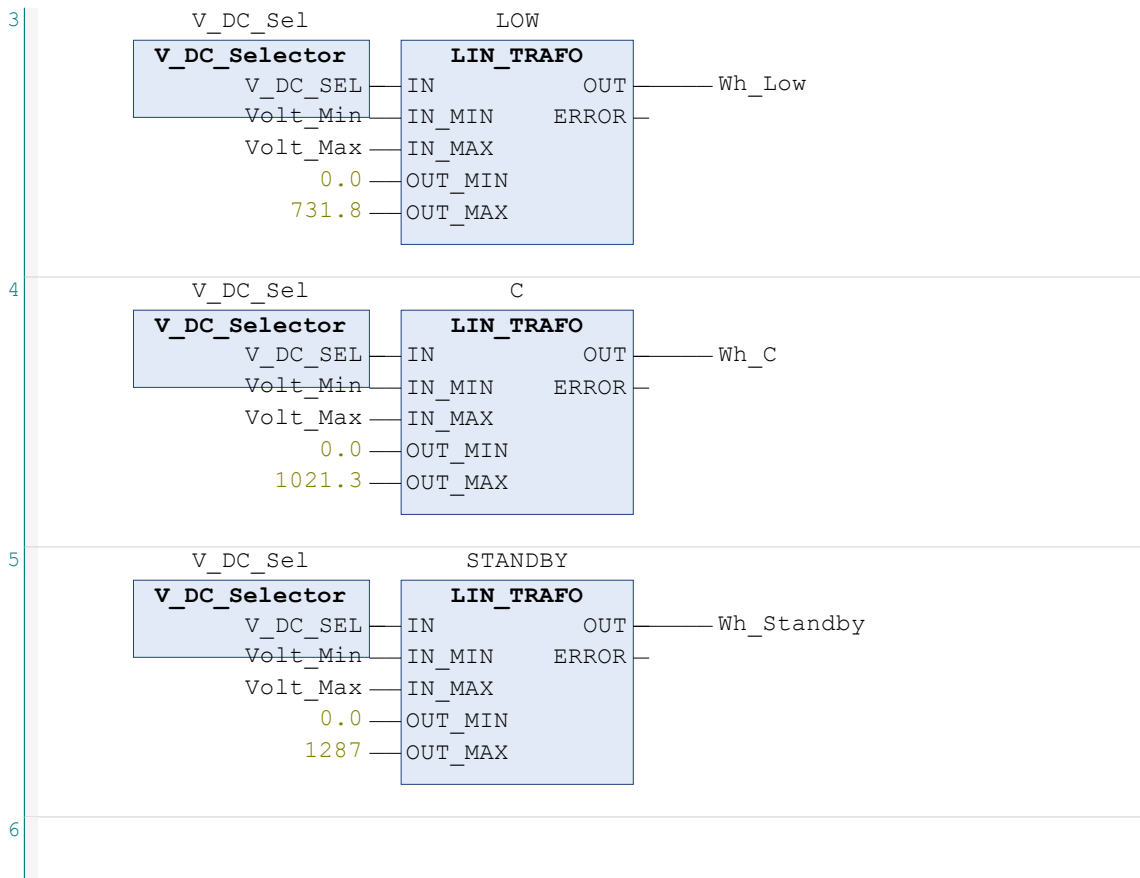
```

1      //FB for estimering av Wg på 100 Ah batteri
2      FUNCTION_BLOCK Capacity_Estimator_100Ah
3      VAR_INPUT
4      END_VAR
5      VAR_OUTPUT
6      //Wh ved de ulike driftstilfellene
7      Wh_High : REAL ;
8      Wh_Med : REAL ;
9      Wh_Low : REAL ;
10     Wh_C : REAL ;
11     Wh_Standby : REAL ;
12     END_VAR
13     VAR
14     //Linær transformering av volt til Wh
15     HIGH : LIN_TRAFO ;
16     MEDIUM : LIN_TRAFO ;
17     LOW : LIN_TRAFO ;
18     C : LIN_TRAFO ;
19     STANDBY : LIN_TRAFO ;
20     V_DC_Sel : V_DC_Selector ;
21
22
23     END_VAR
24

```



4 POU: Capacity_Estimator_100Ah

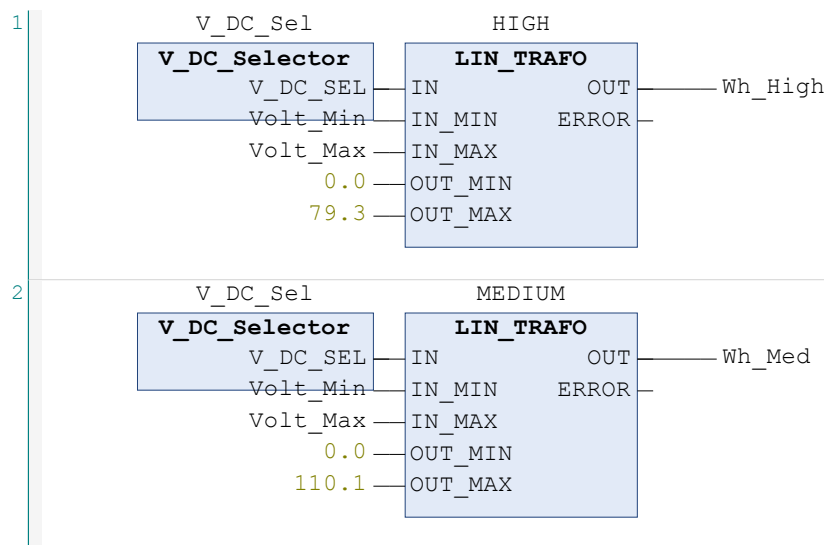


5 POU: Capacity_Estimator_23Ah

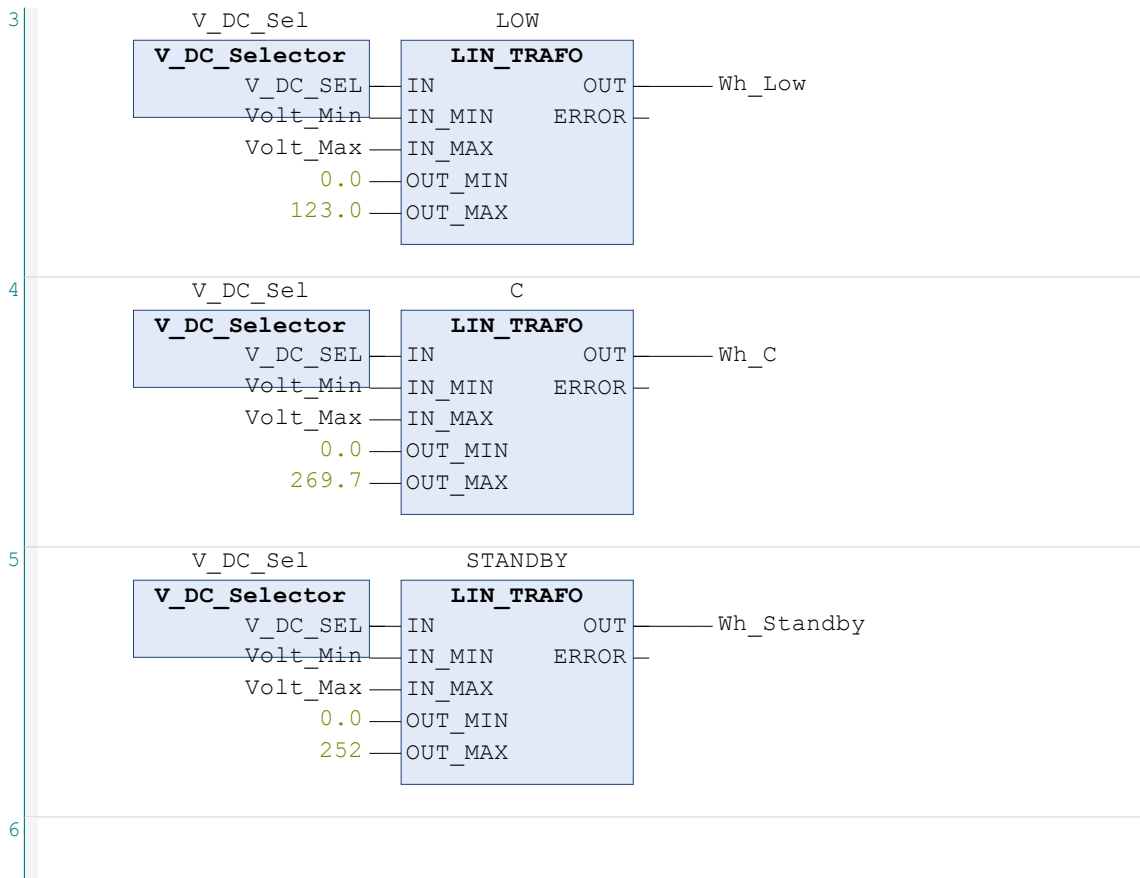
```

1 //FB for estimering av Wg på 23 Ah batteri
2 FUNCTION_BLOCK Capacity_Estimator_23Ah
3 VAR_INPUT
4 END_VAR
5 VAR_OUTPUT
6 //Wh ved de ulike driftstilfellene
7 Wh_High : REAL ;
8 Wh_Med : REAL ;
9 Wh_Low : REAL ;
10 Wh_C : REAL ;
11 Wh_Standby : REAL ;
12 END_VAR
13 VAR
14 //Linær transformering av volt til Wh
15 HIGH : LIN_TRAFO ;
16 MEDIUM : LIN_TRAFO ;
17 LOW : LIN_TRAFO ;
18 C : LIN_TRAFO ;
19 STANDBY : LIN_TRAFO ;
20 V_DC_Sel : V_DC_Selector ;
21 END_VAR
22

```



5 POU: Capacity_Estimator_23Ah



6 POU: Capacity_Selector

```

1 //FB for å velge hvilket batteri man bruker
2 FUNCTION_BLOCK Capacity_Selector
3 VAR_INPUT
4 END_VAR
5 VAR_OUTPUT
6 //Output for kapasitet avhengig av om man bruker 23 Ah eller 100 Ah
7 Cap_High : REAL ;
8 Cap_Med : REAL ;
9 Cap_Low : REAL ;
10 Cap_C : REAL ;
11 Cap_Standby : REAL ;
12 END_VAR
13 VAR
14 Cap23Ah : Capacity_Estimator_23Ah ;
15 Cap100Ah : Capacity_Estimator_100Ah ;
16 END_VAR
17

```

```

1 Cap23Ah () ;
2 Cap100Ah () ;
3
4 IF Battery100 THEN // Hvis knappen = TRUE så velger man 100Ah
5 Cap_High := Cap100Ah . Wh_High ;
6 Cap_Med := Cap100Ah . Wh_Med ;
7 Cap_Low := Cap100Ah . Wh_Low ;
8 Cap_C := Cap100Ah . Wh_C ;
9 Cap_Standby := Cap100Ah . Wh_Standby ;
10 ELSIF Battery235 THEN // Hvis knappen = TRUE så velger man 23Ah
11 Cap_High := Cap23Ah . Wh_High ;
12 Cap_Med := Cap23Ah . Wh_Med ;
13 Cap_Low := Cap23Ah . Wh_Low ;
14 Cap_C := Cap23Ah . Wh_C ;
15 Cap_Standby := Cap23Ah . Wh_Standby ;
16 END_IF
17

```

7 POU: Conversion

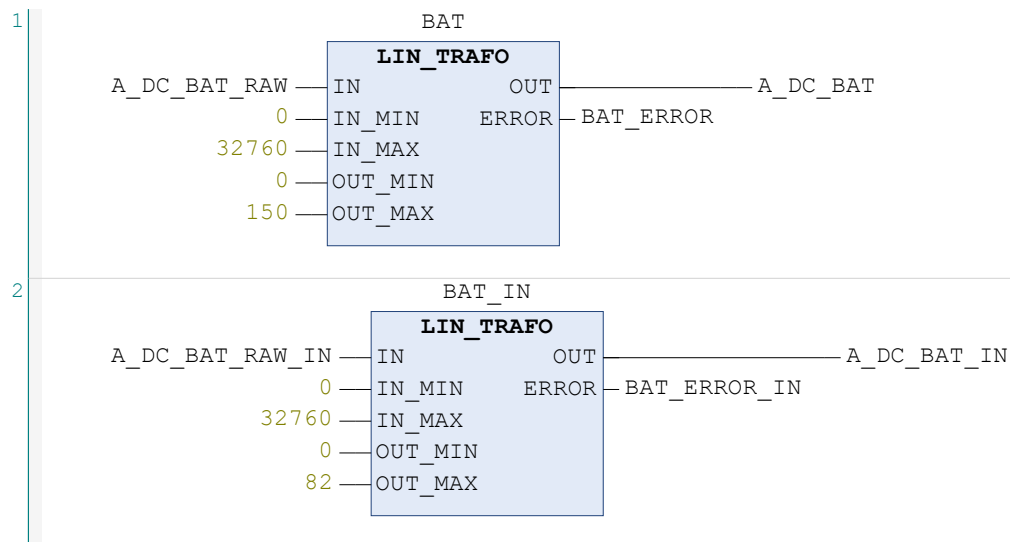
```

1 //FB for konvertering av input verdiene til strømtrafoene og
2 spenningsmålingen av batteri
3 FUNCTION_BLOCK Conversion
4 VAR_INPUT
5 END_VAR
6 VAR_OUTPUT
7 A_DC_BAT : REAL ; //Ampere ut av batteri
8 BAT_ERROR : BOOL ;
9
10 A_DC_EMS : REAL ; //Ampere til PLS/EMS
11 EMS_ERROR : BOOL ;
12
13 A_AC_IN : REAL ; //Ampere AC inn

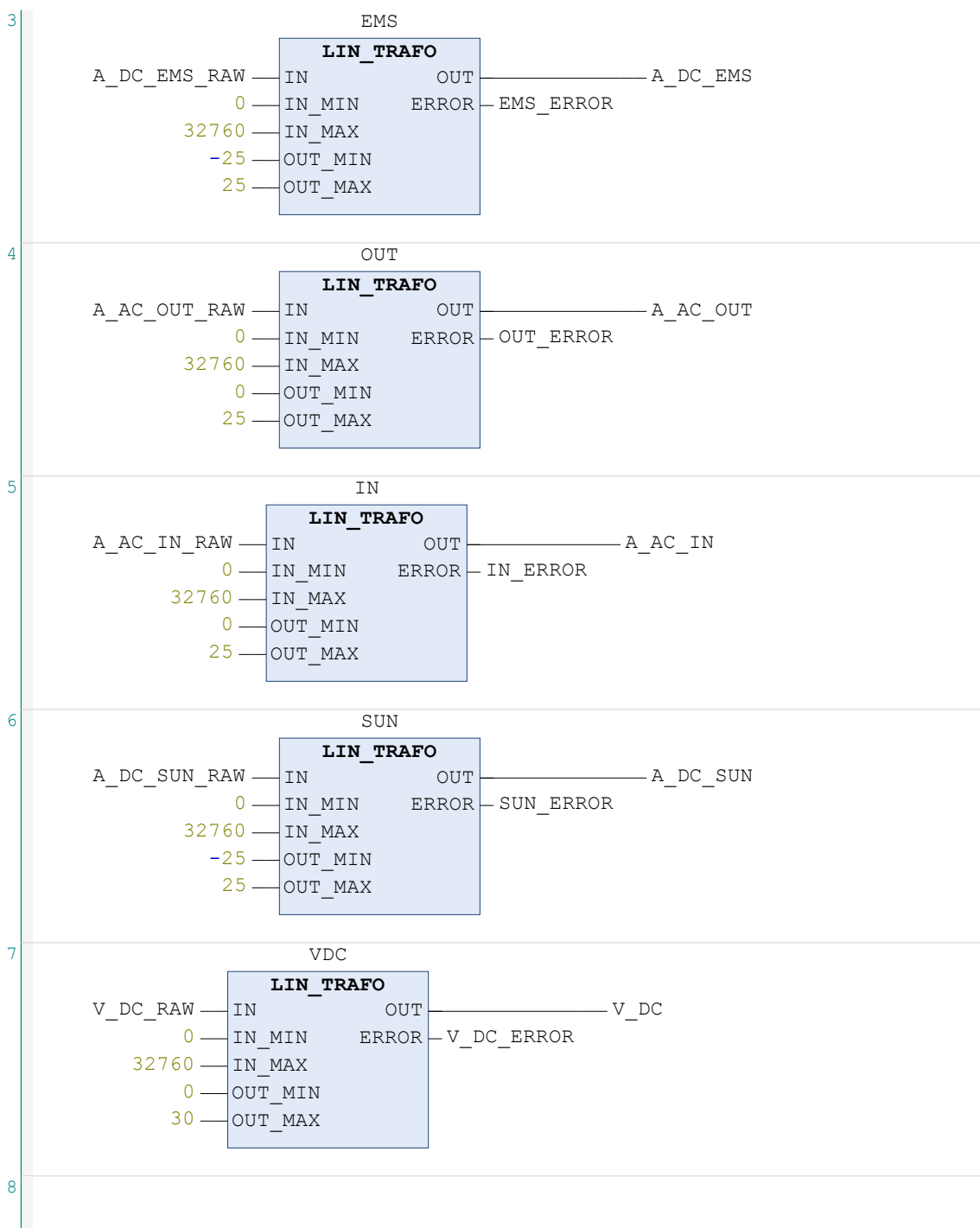
```


7 POU: Conversion

```
13     IN_ERROR : BOOL ;
14
15     A_AC_OUT : REAL ; //Ampere AC ut
16     OUT_ERROR : BOOL ;
17
18     A_DC_SUN : REAL ; //Ampere fra solcelle
19     SUN_ERROR : BOOL ;
20
21     A_DC_BAT_IN : REAL ; //Ampere inn til batteri
22     BAT_ERROR_IN : BOOL ;
23
24     V_DC : REAL ; //Volt over batteri
25     V_DC_ERROR : BOOL ;
26 END_VAR
27 VAR
28     //Linær transformering av input verdiene til strømtrafoene til riktig
    ampere verdi
29     BAT : LIN_TRAFO ;
30     BAT_IN : LIN_TRAFO ;
31     EMS : LIN_TRAFO ;
32     OUT : LIN_TRAFO ;
33     IN : LIN_TRAFO ;
34     SUN : LIN_TRAFO ;
35     VDC : LIN_TRAFO ;
36 END_VAR
37
```



7 POU: Conversion



8 POU: Hovedprogram

```
1  PROGRAM Hovedprogram
2  VAR
3      //
4      States : Load := Standby ;
5
6      //Skrur på/av 0, 1, 2 eller 3 forbrukere
7      Consumer_0 : BOOL ;
8      Consumer_1 : BOOL ;
9      Consumer_2 : BOOL ;
10     Consumer_3 : BOOL ;
11
12     //Brytere for å bytte mellom de ulike energikildene
13     Shore_Switch : BOOL ;
14     Generator_Switch : BOOL ;
15     Battery_Switch : BOOL ;
16     Battery_State : BOOL := TRUE ;
17
18
19     //Funksjonsblokker
20     Wh_Count : Wh_In_Out ;
21     Convert : Conversion ;
22     Power_And_Time : Power_Time ;
23     Volt_DC_Average : Volt_Average ;
24     V_DC_Select : V_DC_Selector ;
25     Cap_Selector : Capacity_Selector ;
26     Capacity_23Ah : Capacity_Estimator_23Ah ;
27     Capacity_100Ah : Capacity_Estimator_100Ah ;
28
29
30     Wh_Count_Timer : TON := ( PT := T#15h ) ;
31     Temp_Time : TIME ;
32     Wh_Total_Bat : REAL ;
33
34     //Alarm variabler
35     GenTemp_High : BOOL ;
36     BatTemp_High : BOOL ;
37     Shore_Off : BOOL ;
38     Reset_All : BOOL ;
39     Reset_GenTemp_High : BOOL ;
40     Reset_BatTemp_High : BOOL ;
41     Low_Voltage : BOOL ;
42     Reset_Low_Voltage : BOOL ;
43     High_Load : BOOL ;
44     Reset_High_Load : BOOL ;
45     Shore_Not_Connected : BOOL ;
46     Reset_Shore_Not_Connected : BOOL ;
47     Critical_Low_Voltage : BOOL ;
48     Reset_Critical_Low_Voltage : BOOL ;
49     Battery_Temp : REAL ;
50     Generator_Temp : REAL ;
51
52
```

8 POU: Hovedprogram

```
53
54     //Styring genrator
55     IngenStrom : TON := ( PT := T#500MS );
56     Vent : TON := ( PT := T#500MS );
57     GenStart_TP : TP := ( PT := T#1S );
58     Generator_starter : BOOL ;
59
60     //Styring land
61     LandTimer : TON := ( PT := T#2S );
62
63     //Sjekkliste
64     Check1 : BOOL ;
65     Check2 : BOOL ;
66     Check3 : BOOL ;
67     Check4 : BOOL ;
68 END_VAR
69
```

```
1     //Gjør om temperaturmålinger til Celsius.
2     GenTemp . integer := %IW9 ;
3     GenTemp . decimal := GenTemp . integer / 10.0 ; //Temperatur til generator
4     Generator_Temp := GenTemp . decimal ;
5     BatTemp . integer := %IW10 ;
6     BatTemp . decimal := BatTemp . integer / 10.0 ; //Temperatur til batteri
7     Battery_Temp := BatTemp . decimal ;
8
9     Wh_Count ( ) ;           //Oppdaterer hvor mye energi som går inn/ut av
    batteri
10    Convert ( ) ;           //Konverterer alle strøm og spenningsmålingene til
    riktige verdier
11    Power_And_Time ( ) ; //Regner ut EOT basert ved de ulike driftstilfellene
12    Capacity_23Ah ( ) ; //Estimering av batterikapasitet til 23 Ah batteri
13    Capacity_100Ah ( ) ; //Estimering av batterikapasitet til 100 Ah batteri
14    Cap_Selector ( ) ;     //FB for å velge om man bruker 23 Ah eller 100 Ah
    batteri
15    Volt_DC_Average ( ) ; //Regner ut en gjennomsnittsverdi for
    spenningsmålingen
16    V_DC_Select ( ) ;     //Velger om man skal bruke gjennomsnitt eller nåtid
    verdi spenning
17
18    CASE States OF           //Case struktur på om man går med
    Standby, Low, Med eller High last
19        Standby :           //I Standby eller alle forbrukerne av
20            K4_User := FALSE ;           //alle kontaktorene til forbrukerne
    er av i standby
21            K5_User := FALSE ;
22            K6_User := FALSE ;
23
24            Wh_Total_Bat := Cap_Selector . Cap_Standby ; // Verdi for Wh som
    skal benyttes hentes basert på batteri og last
25
26            IF Consumer_1 THEN           //Hvis man skrur på en forbruker går man
    over til low
27                States := Low ;
28                Consumer_1 := FALSE ; //Resetter knapp
```

8 POU: Hovedprogram

```
29         END_IF
30
31         IF Consumer_2 THEN           //Hvis man skrur på en forbruker går man
over til med
32             States := Medium ;
33             Consumer_2 := FALSE ; //Resetter knapp
34         END_IF
35
36         IF Consumer_3 THEN           //Hvis man skrur på en forbruker går man
over til high
37             States := High ;
38             Consumer_3 := FALSE ; //Resetter knapp
39         END_IF
40
41
42     Low :
43         K4_User := TRUE ;           //K4 er på i low
44         K5_User := FALSE ;
45         K6_User := FALSE ;
46
47         Wh_Total_Bat := Cap_Selector . Cap_Low ; // Verdi for Wh som skal
benyttes hentes basert på batteri og last
48
49         IF Consumer_0 THEN           //Hvis man skrur på en forbruker går man
over til standby
50             States := Standby ;
51             Consumer_0 := FALSE ; //Resetter knapp
52         END_IF
53
54         IF Consumer_2 THEN           //Hvis man skrur på en forbruker går man
over til med
55             States := Medium ;
56             Consumer_2 := FALSE ; //Resetter knapp
57         END_IF
58
59         IF Consumer_3 THEN           //Hvis man skrur på en forbruker går man
over til high
60             States := High ;
61             Consumer_3 := FALSE ; //Resetter knapp
62         END_IF
63
64     Medium :
65         K4_User := TRUE ;           //K4 og K5 er på i med
66         K5_User := TRUE ;
67         K6_User := FALSE ;
68
69         Wh_Total_Bat := Cap_Selector . Cap_Med ; // Verdi for Wh som skal
benyttes hentes basert på batteri og last
70
71         IF Consumer_0 THEN           //Hvis man skrur på en forbruker går man
over til standby
72             States := Standby ;
73             Consumer_0 := FALSE ; //Resetter knapp
74         END_IF
75
76         IF Consumer_1 THEN           //Hvis man skrur på en forbruker går man
```

8 POU: Hovedprogram

```
over til low
77     States := Low ;
78     Consumer_1 := FALSE ; //Resetter knapp
79     END_IF
80
81     IF Consumer_3 THEN //Hvis man skrur på en forbruker går man
over til high
82     States := High ;
83     Consumer_3 := FALSE ; //Resetter knapp
84     END_IF
85
86
87     High :
88     K4_User := TRUE ; //K4, K5 og K6 er på i high
89     K5_User := TRUE ;
90     K6_User := TRUE ;
91
92     Wh_Total_Bat := Cap_Selector . Cap_High ; // Verdi for Wh som skal
benyttes hentes basert på batteri og last
93
94     IF Consumer_0 THEN //Hvis man skrur på en forbruker går man
over til standby
95     States := Standby ;
96     Consumer_0 := FALSE ; //Resetter knapp
97     END_IF
98
99     IF Consumer_1 THEN //Hvis man skrur på en forbruker går man
over til low
100    States := Low ;
101    Consumer_1 := FALSE ; //Resetter knapp
102    END_IF
103
104    IF Consumer_2 THEN //Hvis man skrur på en forbruker går man
over til med
105    States := Medium ;
106    Consumer_2 := FALSE ; //Resetter knapp
107    END_IF
108
109    END_CASE
110
111    //Landstrøm som energikilde. Dersom landstrømsbryter trykkes eller
spenningen blir for lav
112    IF ( Shore_Switch OR ( V_DC_Select . V_DC_SEL <= ( 11.2 ) ) ) THEN
113    K2_Shore := TRUE ; //Landstrømkontaktor
114    Battery_State := FALSE ; //Brukes til å vise at batteri ikke brukes
som energikilde i HMI
115    K1_Gen := FALSE ; //Generatorkontaktor
116    GenStop := TRUE ; //Rele Stopp generator
117    GenStart := FALSE ; //Rele Start generator
118    Shore_Switch := FALSE ; //Resetter landstrømknappen
119    Wh_Count ( Start := TRUE ) ; //FB for beregning av siste sekunds
watttimer oppdateres
120
121    END_IF
122
123    Wh_Count_Timer ( IN := K2_Shore ) ; //Timer for å summere opp wattimer med
forrige estimat oppdateres
```

8 POU: Hovedprogram

```
124
125 IF ( Wh_Count_Timer . ET >= Temp_Time + T#1S ) THEN //Endring i Wh
summerer med forrige estimerte nivå på batteriet
126 Wh_Total_Bat := Wh_Total_Bat + Wh_Count . Wh_Bat ;
127 Temp_Time := Wh_Count_Timer . ET ;
128 END_IF
129
130 LandTimer ( IN := K2_Shore ) ; //Starter en 2 sek
timer når K2_shore går på
131 IF ( LandTimer . Q = TRUE AND ( A_AC_IN_RAW < 200.0 ) ) THEN //Hvis det
etter 2 sek ikke er strøm inn fra land så går den over til batteri
132 Shore_Off := TRUE ; //Varsel til HMI
hvis det ikke gikk å koble til land
133 Battery_Switch := TRUE ; //Går over til
batteri
134 END_IF
135
136 IF ( A_AC_IN_RAW > 200 ) AND NOT K1_Gen THEN
137 Shore_Off := FALSE ;
138 END_IF
139
140 //Batteri som energikilde
141 IF Battery_Switch THEN
142 Battery_State := TRUE ; //Til HMI for å si man går i batteridrift
143 K2_Shore := FALSE ; //Landstrømkontaktor
144 K1_Gen := FALSE ; //Generatorkontaktor
145 GenStop := TRUE ; //Rele stopp generator
146 GenStart := FALSE ; //Rele start generator
147 Battery_Switch := FALSE ; //Resetter batteriknapp
148
149 //Nullstillingen av timer til Wh inn og ut
150 Wh_Count ( Start := FALSE , Time_Var := T#0S ) ;
151 Wh_Count_Timer ( IN := FALSE ) ;
152 Temp_Time := T#0S ;
153
154 END_IF
155
156 // Generator som energikilde.
157 // Dersom generatorbryter trykkes eller spenningen blir for lav OG
landstrøm ikke er tilkoblet
158 IF ( Generator_Switch OR ( ( V_DC_Select . V_DC_SEL <= Volt_Min ) AND
A_AC_IN_RAW < 130 ) ) THEN
159 K2_Shore := FALSE ; //Landstrømkontaktor
160 K1_Gen := TRUE ; //Generatorkontaktor
161 GenStop := FALSE ; //Rele stopp generator
162 Generator_Switch := FALSE ; //Resetter generatorknapp
163 Generator_starter := TRUE ; //Rele start generator
164 Battery_State := FALSE ; //Til HMI for å si man går i
batteridrift
165
166 //Oppdatering av timere til Wh
167 Wh_Count ( Start := TRUE ) ;
168 Wh_Count_Timer ( IN := TRUE , PT := T#15S ) ;
169 IF ( Wh_Count_Timer . ET >= Temp_Time + T#1S ) THEN //Endring i Wh
summerer med forrige estimerte nivå på batteriet
170 Wh_Total_Bat := Wh_Total_Bat + Wh_Count . Wh_Bat ;
```

8 POU: Hovedprogram

```
171         Temp_Time := Wh_Count_Timer . ET ;
172     END_IF
173
174 END_IF
175
176 //Styring av generator
177 GenStart_TP ( IN := Generator_starter ) ; //Startrelet
178 pulserer høy i 1 sek til generatoren starter
179 IF ( ( A_AC_IN_RAW > 175 ) AND NOT GenStart_TP . Q ) THEN //Hvis det
180     kommer strøm fra generator så slutter start relet å pulsere
181     Generator_starter := FALSE ;
182 END_IF
183
184 IF Generator_starter = TRUE THEN //Sikkerhet for at K1_Gen blir på
185     GenStop := FALSE ;
186     K1_Gen := TRUE ;
187     IF GenTemp . decimal <= 42.0 THEN
188         Servo := TRUE ; //Gir signal til å operere choke
189     END_IF
190     GenStart := GenStart_TP . Q ;
191 ELSE
192     GenStart := FALSE ;
193     Servo := FALSE ;
194 END_IF
195
196 //Alarm styring
197 Shore_Not_Connected S= Shore_Off ; //Landstrøm ikke tilkoblet
198
199 Shore_Not_Connected R= ( Reset_All OR Reset_Shore_Not_Connected ) ;
200
201 High_Load S= ( A_DC_BAT_RAW > 20000 ) ; //Trekkes høy last
202 High_Load R= ( Reset_All OR Reset_High_Load OR ( A_DC_BAT_RAW < 20000 ) ) ;
203
204 Low_Voltage S= ( V_DC_Select . V_DC_SEL <= ( 11.3 ) ) ; //Spenning på
205 batteri er lav
206 Low_Voltage R= ( Reset_All OR Reset_Low_Voltage OR ( V_DC_Select . V_DC_SEL
207     > ( 11.3 ) ) ) ;
208
209 Critical_Low_Voltage S= ( V_DC_Select . V_DC_SEL <= ( 10.8 ) ) ; //Spenning
210 på batteri er veldig lav
211 Critical_Low_Voltage R= ( Reset_All OR Reset_Critical_Low_Voltage OR (
212     V_DC_Select . V_DC_SEL > ( 10.8 ) ) ) ;
213
214 GenTemp_High S= ( GenTemp . decimal > 250 ) ; //Generator temp er høy
215 GenTemp_High R= ( Reset_All OR Reset_GenTemp_High OR ( GenTemp . decimal <
216     250 ) ) ;
217
218 BatTemp_High S= ( BatTemp . decimal > 40 ) ; //Batteri temp er høy
219 BatTemp_High R= ( Reset_All OR Reset_BatTemp_High OR ( BatTemp . decimal <
220     40 ) ) ;
221
222
```


9 POU: Power_Time

```

1 //FB for å regne ut effektflyt og estimering av EOT (Estimated operating
2 times)
3 FUNCTION_BLOCK Power_Time
4 VAR_INPUT
5 END_VAR
6 VAR_OUTPUT
7 //Watt inn/ut komponenter
8 W_Battery : REAL;
9 W_EMS : REAL;
10 W_Inverter : REAL;
11 W_AC_IN : REAL;
12 W_AC_OUT : REAL;
13
14 //Estimated operating times
15 EOT_High : TIME;
16 EOT_Medium : TIME;
17 EOT_Low : TIME;
18 EOT_C : TIME;
19 EOT_Standby : TIME;
20
21 Wait_Estimates : BOOL;
22 END_VAR
23 VAR
24 Wh_Count : Wh_In_Out;
25 Convert : Conversion;
26 Cap_Selector : Capacity_Selector;
27 V_DC_Select : V_DC_Selector;
28 Cap23Ah : Capacity_Estimator_23Ah;
29 Cap100Ah : Capacity_Estimator_100Ah;
30 END_VAR

```

```

1 Cap23Ah ();
2 Cap100Ah ();
3 Convert ();
4 Wh_Count ();
5 Cap_Selector ();
6 V_DC_Select ();
7
8 Wait_Estimates := (V_DC_Select.V_DC_SEL <= 12.75); //Spenningen over
9 batteriet må være under 12,75 V
10
11 W_EMS := ABS (Convert.A_DC_EMS * V_DC_Select.V_DC_SEL); //Watt
12 EMS/PLS
13 W_Battery := ABS (Convert.A_DC_BAT * V_DC_Select.V_DC_SEL); //Watt
14 batteri
15 W_AC_IN := Convert.A_AC_IN * 230.0; //Watt AC
16 inn
17 W_AC_OUT := Convert.A_AC_OUT * 230; //Watt AC ut
18
19 IF K1_Gen OR K2_Shore THEN //Regner ut Watt til inverter fra Watt

```

9 POU: Power_Time

```
    til batteri og EMS
16     W_Inverter := W_Battery + W_EMS ;
17     ELSE
18     W_Inverter := W_Battery - W_EMS ;
19     END_IF
20
21     //Regner ut EOT verdier i de ulike driftstilfellene
22     EOT_High := REAL_TO_TIME ( 3600000 * Cap_Selector . Cap_High / W_Battery ) ;
23     EOT_Medium := REAL_TO_TIME ( 3600000 * Cap_Selector . Cap_Med / W_Battery ) ;
24     EOT_Low := REAL_TO_TIME ( 3600000 * Cap_Selector . Cap_Low / W_Battery ) ;
25     EOT_C := REAL_TO_TIME ( 3600000 * Cap_Selector . Cap_C / W_Battery ) ;
26     EOT_Standby := REAL_TO_TIME ( 3600000 * Cap_Selector . Cap_Standby / W_Battery )
    ;
27
```

10 POU: Volt_Average

```
1     //FB for å regne ut gjennomsnittsverdi av spenning
2     FUNCTION_BLOCK Volt_Average
3     VAR_INPUT
4     END_VAR
5     VAR_OUTPUT
6     V_DC_AVERAGE : REAL ;
7     END_VAR
8     VAR
9     AverageTimer : TON ;
10    TempTime : TIME ;
11    Values : ARRAY [ 1 .. 100 ] OF REAL ;
12    Convert : Conversion ;
13    n : INT ;
14    k : INT ;
15    ArraySum : REAL ;
16    i : INT ;
17    END_VAR
18
19
20    // Oppdaterer spenningsmåleren
21    Convert ( ) ;
22    // Starter timeren
23    AverageTimer ( IN := TRUE , PT := T#15H ) ;
24
25
26    IF ( AverageTimer . ET >= TempTime + T#3S270MS ) THEN // Skal ikke
gjennomføres oftere enn hvert 3. sek og 270 ms
27
28        // Flytter alle verdier ett hakk videre i "Values" og frigjør plass til
en ny verdi i den første.
29        FOR n := 1 TO 99 DO
30            Values [ n ] := Values [ n+1 ] ;
31        END_FOR
32
33        // Mater inn aktuell spenning i dne nye ledige plassen ved batteridriftm
ved land/generatur mates 0 inn.
34        IF NOT ( K1_Gen OR K2_Shore ) THEN
35            Values [ 100 ] := Convert . V_DC ;
36        ELSE
```

10 POU: Volt_Average

```
17         Values [ 100 ] := 0 ;
18     END_IF
19
20     // Summerer opp og teller alle verdier over 0.
21     FOR i := 1 TO 100 DO
22         IF Values [ i ] > 0 THEN
23             ArraySum := ArraySum + Values [ i ] ;
24             k := k + 1 ;
25         END_IF
26     END_FOR
27
28     // Deler summen på antall verdier over 0, deler på 1 dersom det ikke er
    noen verdier over 0.
29     IF k > 0 THEN
30         V_DC_AVERAGE := ArraySum / k ;
31     ELSE
32         k := 1 ;
33         V_DC_AVERAGE := ArraySum / k ;
34     END_IF
35
36     // Nullstiller antall verdier og summen. Gjør klar for å telle på nytt.
37     k := 0 ;
38     ArraySum := 0 ;
39     TempTime := AverageTimer . ET ;
40
41 END_IF
42
```

11 POU: V_DC_Selector

```
1     //FB for å velge om man skal bruke gjennomsnitt verdi eller nåtid verdi av
    spenningsmåling
2     FUNCTION_BLOCK V_DC_Selector
3     VAR_INPUT
4     END_VAR
5     VAR_OUTPUT
6         V_DC_SEL : REAL ;
7     END_VAR
8     VAR
9         DC_Realtime : Conversion ;
10        DC_Average : Volt_Average ;
11    END_VAR
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
```

```
9     END_IF
10
```

12 POU: Wh_In_Out

```
1     //FB for å regne ut Wh inn og ut av batteri
2     FUNCTION_BLOCK Wh_In_Out
3     VAR_INPUT
4         Start : BOOL ;
5         Time_var : TIME ;
6     END_VAR
7     VAR_OUTPUT
8         Wh_Bat : REAL ; //Endring av Wh inn/ut av batteri
9     END_VAR
10    VAR
11        Wh_Timer : TON ;
12        Convert : Conversion ;
13    END_VAR
14
```

```
1     //Regner ut Wh inn til batteri når man går med landstrøm eller generator
2     //Multipliserer tiden med spenning og strøm inn til batteri
3     Wh_Timer ( IN := Start , PT := T#15H ) ;
4     IF ( Wh_Timer . ET >= Time_var + T#1S ) THEN
5         Convert ( ) ;
6         Wh_Bat := ABS ( ( TIME_TO_REAL ( Wh_Timer . ET - Time_var ) / 3600000 ) * (
7         Convert . A_DC_BAT_IN ) * ( Convert . V_DC ) ) ;
8         Time_var := Wh_Timer . ET ;
9     END_IF
```