

DMM Start

Reset (void)

```
{ Init_Ext_Registers();
  Panel_Reset();
  Init_Int_Registers();
  QuickSelfTest();
```

```
QuickSelfTest:                                CODE XREF: Reset$Gcip
mov     rax, rdi                               ; -5A
mov     rax, rax                               ; -5B
mov     rax, rax                               ; -5C
mov     rax, rax                               ; -5D

loc_29B70:
mov     rax, rdi                                CODE XREF: QuickSelfTest$5A
mov     rax, rax                               ; 5A: Keyboard leavestate array addr
jsr     Kbd_Is_HOME_pressed                   ; checking if HOME is pressed (DIAG mode)
mov     rax, dr[DIAG_Mode_Indx]
cmp     rax, # (DIAG_Mode_Indx)
beq     rax, loc_29B8E                       ; skip all tests if DIAG_flag = 1
pea     var_1(a6)
mov     rax, loc_29B80000 - (sp)
mov     rax, rax                               ; 29B80000 - (sp)
mov     rax, rax                               ; 29B80000 - (sp)
jsr     Mem_Check                             ; DRAM U104 checking
```

```
// No tests is performed in DIAG mode!
```

```
VRTX_INIT();  
VRTX_GO(Main_Task); // Start Application Execution
```

Main_Task(void)

```

Line_Freq_Measure();
if ( !Config_Memory_Check() ) Mem_Init();
Hardware_Init();
Copy_ISR_Addresses();
GPIB_Init();
 // not use in DEBUG mode
Menu_Init() // not use in DEBUG mode
VRTX_SC_SCREATE(...); // Create Semaphores
...
VRTX_SC_TCREATE(...); // Create Tasks
...
VRTX_SC_TCREATE(DIAG_Task); // Create Diagnostic Task
DIAG_Menu_Init();
VRTX_SC_TDELETE(); // Delete Current Task
}

```

```

DIAG_Task:                                     : DATA XREF: Main_Task+28E.o
                                                : DIAG_eub_6F5E8_Display_Check1_2_3+4C.o
                                                : DIAG_eub_6F79E+72.o
                                                : DIAG_eub_6F802+38.o
                                                : DIAG_Task_Create+46.o

var_4                                          : -4

link a6, #-4
movem.l d0/a5, -(sp)

loc_20454:                                     : CODE XREF: DIAG_Task+38.j

pea      var_4(a6)
clr.l   -(sp)
pea      (off_a0), w
jsr      URTX_SC_OPENED
lea      C4(sp), sp
movem.l d7, a5
move.w   Z(a5), d7
ext.l    d7
movem.l d7, (sp)
move.w   (a5), d7
ext.l    d7
asl.l    #3, d7
movem.l d7, a1
adda.l   #table_DIAG_checks_eubs, a1
movem.l (a1), a1
jsr      (a1)
bra.s    loc_20454

-----
movem.l (sp)+, d0/a5
unlk    a6
rts

: End of function DIAG_Task

```

**Message
(DIAG subfunction)**

```
table DIAG_Checks.subsdc {DIAG_Reset                                     DATA_XREF: DIAG_Task*2E;o
dc:1 DIAG_Rstb_Check
dc:1 DIAG_Rstb_Check
dc:1 DIAG_Display_Checks
dc:1 DIAG_Kay_Check
dc:1 DIAG_Transfer_Check
dc:1 DIAG_Rstb_Check
dc:1 DIAG_Int_Check
0 : 10u Zero (FAST mode)
0 : 10u Zero (PRECISION mode)
2 : 1u Zero
3 : 100uA Zero
4 : 7.2tref
5 : -100tref
6 : -10tref
7 : -0.1tref
8 : Internal temperature

dc:1 DIAG_Current
dc:1 DIAG_ac_Check
dc:1 DIAG_Option_Check
```

DIAG subfunction LUT

```

0 : 10u Zero (FAST mode)
1 : 10u Zero (PRECISION mode)
2 : 10 Zero
3 : 1000u Zero
4 : 7.20u ref
5 : -10u ref
6 : -10u ref
7 : -0.10u ref
8 : Internal temperature
Attributes: nonreturn hps-based frame

DING_Int_Check:                                     : DATA XREF: EPRON:0092D43C+0
IC_idx                                             : $8
d5_IC_idx <= d5
link a6, #1
move.w d0/d5_IC_idx, (sp)
move.w (a6), d5_IC_idx
clr.w (Measurement_Mode).l : 0:DCU 1:ACU 2:DCl 3:ACI 4:2M 5:4M 6:FRO 7:PER
move.w (Measurement_Mode_5).l : (Measurement_Mode_Store).l : 0:DCU 1:ACU 2:DCl 3:ACI
move.w (Measurement_Mode).l : (Measurement_Mode_Store).l : 0:DCU 1:ACU 2:DCl 3:ACI
(ZERO_AUTO_Flag).l
move.w (Measurement_Mode).l : d7 : 0:DCU 1:ACU 2:DCl 3:ACI 4:2M 5:4M 6:FRO 7:PER
ext.l d7
move.l d7, d0
asl.l #1, d7 : x4
add.l d0, d7 : x5
asl.l #1, d7 : x20
add.l d0, d7 : x21
add.l d7, d7 : x42
add.l #DCU_CONFIG.d7 : add 0
move.l d7, (Config_data_ptr).l
move.w (Config_data_ptr).l
move.w d5_IC_idx, d7
ext.l d7
asl.l #1, d7
move.w d7, d2 : x4
add.l #Table_DING_DCU_range, a2
move.w (a2), (a1) : RANGE Variable loading
move.l (Config_data_ptr).l, a1
move.w d5_IC_idx, d7
ext.l d7
asl.l #1, d7
move.w d7, d2
add.l #Table_DING_Int, a2
move.w (a2), (a1)
move.w (Config_data_ptr).l, a1
clr.w (a1)
move.w (Config_data_ptr).l, a1
move.w #1, (a1) : DCl 3:4:5
move.w (Config_data_ptr).l, (sp)
clr.l -(sp) : Flag=0
jsr DING_Int_Check_Init
addq.l #1, sp
move.w d5_IC_idx, d7
ext.l d7
add.l d7, d0
asl.l #1, d7
sub.l d0, d7
add.l d7, d0
move.l d7, -(sp)
add.l #Table_DING_IC_Init, (sp)
jsr FPGA_Send_List : (0-terminated list of words)
                                : $FFXX - delay index, $XXXX - FPGA command
addq.l #1, sp
capi.w #d5_IC_idx
loc_2D9A8:
loc_2D9A8:
jsr sub_7C654
jsr sub_7E498
jsr sub_544CC
clr.l (sp)
jsr URAM_Dispatch
jsr URAM_Select_Dispatch
move.w (Config_data_ptr).l, a1
capi.w #16, (a1) : Fint <= 0.0001 s ?
bnc s loc_2D9A8
move.l db1_2D958(pc), (ADC_ZERO_offset).l : 614400.0
move.l db1_2D958+4(pc), (ADC_ZERO_offset+4).l : 614400.0
bra s loc_2D9A8
loc_2D9A8:
loc_2D9A8:
                                : CODE XREF: DING_Int_Check+114;
move.l db1_2D958(pc), (ADC_ZERO_offset).l : 6.144E8
move.l db1_2D958+4(pc), (ADC_ZERO_offset+4).l : 6.144E8
loc_2D9A8:
loc_2D9A8:
                                : CODE XREF: DING_Int_Check+126;
move.l (Config_data_ptr).l, (sp)
move.w (Measurement_Mode).l : d7 : 0:DCU 1:ACU 2:DCl 3:ACI 4:2M 5:4M 6:FRO 7:PER
ext.l d7
move.l d7, -(sp)
move.w d5_IC_idx, d7
ext.l d7
asl.l #1, d7
move.l d7, a1
add.l #Table_DING_IC_math_subs, a1
move.l (a1), -(sp)
jsr DING_IC_math_result
addq.l #1, sp
move.l (sp)+, d0/d5
unlk a6
rts
End of function DING_Int_Check

```

**Forced ADC ZERO to fixed value
(middle of the scale)**

Simple A/D conversion without AUTO ZERO, delays & linearization

```

DIAG_Get_ADC_result:                                ; CODE XREF: DIAG_Int_CheckF8!sp
                                                    ; DIAG_CurrentF8.p
                                                    ; DIAG_AC_CheckF8.p
var_4        = -4
diag_32_math_sub = 8
Measurement_Mode = $E
Config_data_ptr = $10

d5_Measurement_Mode = d5
a5_Config_data_ptr = a5
        link
        moven.1 d0/d3-d5_Measurement_Mode/a4=a5_Config_data_ptr,-(sp)
        move.w Measurement_Mode(a5),d5_Measurement_Mode
        move.w Config_data_ptr(a5),a5_Config_data_ptr

loc_2D0D6:                                ; CODE XREF: DIAG_Get_ADC_result+3Cj
                                                    ; DIAG_Get_ADC_result+52j
                                                    ; DIAG_Get_ADC_result+D4j
        move.w # (AD_conversion_flag),l
        move.l (addr_PPGA_Reg1_Tag),l.at : Write 0FFh to trigger
        move.b $FF,(a1)
        jbr VRAM_Send_Mail
        jbr l5_FPGA_Ready
        move.l d7,a4
        move.l (addr_PPGA_Rx_Buffer),l.at
        moveq.w #0,l
        move.w (a1),d4
        move.l a4,d7
        beq.s loc_2D0D6
        jbr (FPGA_Wait_Ready),l
        move.l (addr_PPGA_Tx_Buffer),l.at
        move.w #0,l.at : read lower 16 bit of AD result
        jbr l5_FPGA_Ready
        move.l d7,a4
        move.l (addr_PPGA_Rx_Buffer),l.at
        moveq.w #0,l.at
        move.w (a1),d3
        move.l a4,d7
        beq.s loc_2D0D6
        moveq.w #10,d7
        asl.l d7,d4
        move.l d4,d7
        asl.l d7,d4
        move.l d4,d7
        or.l d3,d7
        move.l d7,d0
        jbr math_ulong_to_double : convert from 32-bit unsigned integer to double
        move.l d5 (ADC_result),l
        move.l d7 (ADC_result+4),l
        move.l DIAG_32_math_sub(a5),al
        lsr (a1)
        ; Measurement result calculation
        move.l (Measurement_result+1,Measurement_result_store),l
        move.l (Measurement_result+4),l.(Measurement_result_store+4),l
        clr.w (Overflow_flag_store),l
        clr.w (AD_conversion_flag),l
        move.l a5_Config_data_ptr,(sp)
        move.w d5_Measurement_Mode,d5
        ext.l d7
        move.l d7,-(sp)
        move.l lword_2324Ch,-(sp)
        jbr sub_T0C05H
        addq.l #1,sp
        pea var_4(a5)
        move.l (VRAM_Disp_sub_addr),l,-(sp)
        pea ($46),w
        jbr UNTIL_E0_DP05T
        lea $1(sp),sp
        bra.w loc_2D0D6
;-----
        move.l (sp)+,d0/d3-d5_Measurement_Mode/a4=a5_Config_data_ptr
        unlk. a5
        rts
; End of function DIAG_Get_ADC_result

```

Post message to Display Task

```

DIAG_Int_Check_Volts_math:      DATA XREF EPROM:table_DIAG_IC_math_sub0ro
                                EPROM_0002D930+0
                                EPROM_0002D930+1
                                EPROM_0002D930+2
                                EPROM_0002D930+3
                                EPROM_0002D930+4
                                EPROM_0002D930+5
                                EPROM_0002D940+0
                                EPROM_0002D940+1
                                EPROM_0002D940+2
                                EPROM_0002D940+3
                                EPROM_0002D940+4
                                EPROM_0002D940+5
                                DIAG_Current*F2+0
                                DIAG_AC_Check*F8+0

    move.l   (ADC_result),d6
    move.l   (ADC_result+4),d7
    lea     (ADC_ZERO_offset),a0 ; ADC zero point (forced to 6.144E8)
    jsr     (d6-d7 * (a0)) or (d6-d7 * d1.d2[a0=0]) -> d6
    lea     (ADC_scale),a0
    jsr     math_mul ; (d6-d7 * (a0)) or (d6-d7 * d1.d2[a0=0]) -> d6
    lea     (INPUT_ZERO_offset),a0
    jsr     math_sub ; (d6-d7 * (a0)) or (d6-d7 * d1.d2[a0=0]) -> d6
    move.l   d6,(Measurement_result)+1
    move.l   d7,(Measurement_result+4),1
    rts

; End of function DIAG_Int_Check_Volts_math

```

$$\text{Measurement_result} = (\text{ADC_result} - \text{ADC_ZERO_offset}) * \text{ADC_scale} - \text{INPUT_ZERO_offset}$$