

SIOC-PROSIM

```
// *****
// * Opencockpits MCP - By Manolo VÈlez - www.opencockpits.com
// * Adapted by Marty Bochane - Prosim
// *****
// * FileName : prosim_mcp.txt
// * Date : 2012-01-18
//
//
// This SIOC script supports One MCP device, two EFIS devices and one CDU device
// For this script to work, the following device numbers are expected:
//
// Device 0: MCP
// Device 1: Efis Captain
// Device 2: Efis CO
//
// Devices do not need to be present, but if they are, they should be configured
for the above device numbers.
// Configuration is done in the sioc.ini file with "master=" lines.
// *****
// * Config_SIOC ver 3.6 - By Manolo VÈlez - www.opencockpits.com
// *****
// * FileName : modulo_COM.txt
// * Date : 01/12/2008
var 0,value 0
{
&ent_com1 = 118
&dec_com1 = 0
&act_com1 = 118000
&enc_com1 = 1
}
// ***** COM1 *****
var 2, name enc_com1
{
if &enc_com1 = 1
{
&io_pt_com1 = 1
&io_pt2_com1 = 1
call &sub_d_com1
&id2_com1 = &act_com1
}
else
{
&io_pt_com1 = 0
&io_pt2_com1 = 0
&id_com1 = -999999
&id2_com1 = -999999
}
}
var 4, name ent_com1
var 6, name dec_com1
var 8, name ie_dec_com1, link IOCARD_ENCODER, input 4, type 2, aceleration 1,
device 4
```

## SIOC-PROSIM

```

{
if &enc_com1 = 1
{
l0 = &ie_dec_com1 * 25 // oprindeligt 25 aendret til 10 men nu tilbage til 25
l1 = &dec_com1 + l0
if l1 < 0
{
l1 = 975
}
if l1 > 975
{
l1 = 0
}
&dec_com1 = l1
call &sub_d_com1
}
}
var 10, name ie_ent_com1, link IOCARD_ENCODER, input 0, type 2, aceleration 1,
device 4
{
if &enc_com1 = 1
{
l0 = &ie_ent_com1 * -1
&ent_com1 = rotate 118,135,l0
call &sub_d_com1
}
}
var 12, name id_com1,link IOCARD_DISPLAY, digit 6, numbers 6, device 4
var 14, name sub_d_com1, link SUBROUTINE
{
l0 = &ent_com1 * 1000
&id_com1 = l0 + &dec_com1
}
var 16, name io_pt_com1, link IOCARD_OUT, output 20, device 4
var 18, name ii_sw_com1, link IOCARD_SW, input 6, device 4
{
if &ii_sw_com1 = 1
{
if &enc_com1 = 1
{
l0 = &ent_com1 * 1000
l0 = l0 + &dec_com1
l1 = div &act_com1 1000
l2 = mod &act_com1 1000
&act_com1 = l0
&ent_com1 = l1
&dec_com1 = l2
call &sub_d_com1
&id2_com1 = &act_com1
}
}
}
}

```

SIOC-PROSIM

```

var 20, name act_com1
var 22, name id2_com1, link IOCARD_DISPLAY, digit 0, numbers 6, device 4
{
if &id2_com1 > 0
{
&xp_com1 = div &id2_com1 10
}
}
var 24, name io_pt2_com1, link IOCARD_OUT, output 21, device 4
var 26, name xp_com1, value 11800
{
l0 = &xp_com1 - 10000
&fcom1 = tobcd l0
}
// ***** OFFSETS *****
Var 27, Link FSUIPC_OUT, name fcom1, Offset $034E, Length 2 // COM1 Active
// *****
// * Config_SIOC ver 3.6 - By Manolo VÈlez - www.opencockpits.com
// *****
// * FileName : modulo_NAV.txt
// * Date : 01/12/2008
var 53, value 0
{
&ent_nav1 = 108
&dec_nav1 = 0
&act_nav1 = 10800
&enc_nav1 = 1
}
// ***** NAV1 *****
var 54, name enc_nav1
{
if &enc_nav1 = 1
{
&io_pt_nav1 = 1
&io_pt2_nav1 = 1
call &sub_d_nav1
&id2_nav1 = &act_nav1
}
else
{
&io_pt_nav1 = 0
&io_pt2_nav1 = 0
&id_nav1 = -999999
&id2_nav1 = -999999
}
}
var 56, name ent_nav1
var 58, name dec_nav1
var 60, name ie_dec_nav1, link IOCARD_ENCODER, input 4, type 2, aceleration 1,
device 5
{
if &enc_nav1 = 1

```

SIOC-PROSIM

```

{
l0 = &ie_dec_nav1 * 25
l1 = &dec_nav1 + l0
if l1 < 0
{
l1 = 975
}
if l1 > 975
{
l1 = 0
}
&dec_nav1 = l1
call &sub_d_nav1
}
}
var 62, name ie_ent_nav1, link IOCARD_ENCODER, input 0, type 2, aceleration 1,
device 5
{
if &enc_nav1 = 1
{
l0 = &ie_ent_nav1 * -1
&ent_nav1 = rotate 108,117,l0
call &sub_d_nav1
}
}
var 64, name id_nav1,link IOCARD_DISPLAY, digit 5, numbers 5, device 5
var 66, name sub_d_nav1, link SUBROUTINE
{
l0 = &ent_nav1 * 100
l1 = div &dec_nav1 10
&id_nav1 = l0 + l1
}
var 68, name io_pt_nav1, link IOCARD_OUT, output 20, device 5
var 70, name ii_sw_nav1, link IOCARD_SW, input 6, device 5
{
if &ii_sw_nav1 = 1
{
if &enc_nav1 = 1
{
l0 = &ent_nav1 * 100
l1 = div &dec_nav1 10
l0 = l0 + l1
l1 = div &act_nav1 100
l2 = mod &act_nav1 100
&act_nav1 = l0
&ent_nav1 = l1
l0 = mod l2 10
l2 = l2 * 10
if l0 > 0
{
l2 = l2 + 5
}
}
}
}

```

## SIOC-PROSIM

```

&dec_nav1 = 12
call &sub_d_nav1
&id2_nav1 = &act_nav1
}
}
}
var 72, name act_nav1
var 74, name id2_nav1, link IOCARD_DISPLAY, digit 0, numbers 5, device 5
{
if &id2_nav1 > 0
{
&xp_nav1 = &id2_nav1
}
}
var 76, name io_pt2_nav1, link IOCARD_OUT, output 21, device 5
var 78, name xp_nav1, value 10800
{
l0 = &xp_nav1 - 10000
&fnav1 = tobcd l0
}
// ***** OFFSETS ****
Var 79, Link FSUIPC_OUT, name fnav1, Offset $0350, Length 2 // NAV1 Active

// NAV 2

// *****
// End NAV1
// *****

// *****
// Start NAV2
// *****

var 7300, value 0
{
&nav2_ent = 108
&nav2_dec = 0
&nav2_act = 10800
}

var 7301, name nav2_ent

var 7302, name nav2_dec

var 7303, name nav2_dec_enc, link IOCARD_ENCODER, input 4, type 2, aceleration
1, device 9
{
l0 = &nav2_dec_enc * 50
l1 = &nav2_dec + l0
if l1 < 0
{

```

SIOC-PROSIM

```

    l1 = 950
}
if l1 > 950
{
    l1 = 0
}
&nav2_dec = l1
call &sub_d_nav2
}

var 7304, name nav2_ent_enc, link IOCARD_ENCODER, input 0, type 2, aceleration
1, device 9
{
    l0 = &nav2_ent_enc * -1
    &nav2_ent = rotate 108,117,l0
    call &sub_d_nav2
}

var 7305, name nav2_id,link IOCARD_DISPLAY, digit 5, numbers 5, device 9

var 7306, name sub_d_nav2, link SUBROUTINE
{
    l0 = &nav2_ent * 100
    l1 = div &nav2_dec 10
    &nav2_id = l0 + l1
}

var 7307, name nav2_act_pt, link IOCARD_OUT, output 20, device 9
var 7308, name nav2_stb_pt, link IOCARD_OUT, output 21, device 9

var 7309, name nav2_tfr_sw, link IOCARD_SW, input 6, device 9
{
    if &nav2_tfr_sw = 1
    {
        l0 = &nav2_ent * 100
        l1 = div &nav2_dec 10
        l0 = l0 + l1
        l1 = div &nav2_act 100
        l2 = mod &nav2_act 100
        &nav2_act = l0
        &nav2_ent = l1
        l0 = mod l2 10
        l2 = l2 * 10
        if l0 > 0
        {
            l2 = l2 + 5
        }
        &nav2_dec = l2
        call &sub_d_nav2
        &nav2_id2 = &nav2_act
    }
}

```

## SIOC-PROSIM

```
var 7310, name nav2_test_sw, link IOCARD_SW, input 2, device 9
{
  if &nav2_test_sw = 1
  {
    &nav2_id = 88888
    &nav2_id2 = 88888
  }
  else
  {
    CALL &sub_d_nav2
    &nav2_id2 = &nav2_act_pro
  }
}

var 7311, name nav2_act

var 7312, name nav2_id2, link IOCARD_DISPLAY, digit 0, numbers 5, device 9
{
  if &nav2_id2 > 0
  {
    &xp_nav2 = &nav2_id2
  }
}

var 7313, name xp_nav2, value 10800
{
  l0 = &xp_nav2 - 10000
  &fnav2 = tobcd l0
}

var 7314, name nav2_act_pro
{
  if &nav2_id <> &nav2_act_pro
  {
    &nav2_id2 = &nav2_act_pro
    &nav2_act = &nav2_act_pro
  }
}

Var 7315, Link FSUIPC_OUT, name fnav2, Offset $0352, Length 2 // NAV1 Active

// END of RADIOS
// CDU
Var 80, name CDUKey, Link IOCARD_KEYS, device 3 // Keyboard reading
Var 81, name CDUExec, Link IOCARD_OUT, device 3, Output 2 // device
Var 82, name CDUFail, Link IOCARD_OUT, device 3, Output 4
Var 83, name CDUMSG, Link IOCARD_OUT, device 3, Output 1
// Start of MCP Device
// DIGITS
```

## SIOC-PROSIM

```
// These are the actual MCP values that are read and written by Prosim. They
need to be whole values.
Var 100, name D_COURSE1, Link IOCARD_DISPLAY, Digit 0, Numbers 3
Var 102, name D_IAS, Link IOCARD_DISPLAY, Digit 3, Numbers 3
Var 104, name D_HDG, Link IOCARD_DISPLAY, Digit 6, Numbers 3
Var 106, name D_ALT, Link IOCARD_DISPLAY, Digit 9, Numbers 5
Var 108, name D_VS, Link IOCARD_DISPLAY, Digit 14, Numbers 5
Var 110, name D_COURSE2, Link IOCARD_DISPLAY, Digit 19, Numbers 3
// Vertical Speed Trigger
// This variable is set to 1 when the VS wheel is rotated while the VS display
is blanked.
// This can trigger a V/S mode if MCP altitude is different than current
altitude.
Var 112, Name VS_Trigger
// OUTPUTS
// These are the lights on the MCP, written by Prosim
Var 200, name DECIMAL
{
// Do not place the decimal if the display is blanked
If &Inhibit_IAS = 0
{
V201 = &DECIMAL
}
}
Var 201, Link IOCARD_OUT, Output 20
Var 202, name O_N1, Link IOCARD_OUT, Output 21
Var 204, name O_AT, Link IOCARD_OUT, Output 22
Var 206, name O_SPEED, Link IOCARD_OUT, Output 23
Var 208, name O_LVLCHG, Link IOCARD_OUT, Output 24
Var 210, name O_VNAV, Link IOCARD_OUT, Output 25
Var 212, name O_HDGSEL, Link IOCARD_OUT, Output 26
Var 214, name O_APP, Link IOCARD_OUT, Output 27
Var 216, name O_VORLOC, Link IOCARD_OUT, Output 28
Var 218, name O_LNAV, Link IOCARD_OUT, Output 29
Var 220, name O_ALTHLD, Link IOCARD_OUT, Output 30
Var 222, name O_VS, Link IOCARD_OUT, Output 31
Var 224, name O_CWSA, Link IOCARD_OUT, Output 32
Var 226, name O_CMDA, Link IOCARD_OUT, Output 33
Var 228, name O_CWSB, Link IOCARD_OUT, Output 34
Var 230, name O_CMDB, Link IOCARD_OUT, Output 35
Var 232, name O_FD1 // Not available on OC MCP
Var 234, name O_FD2 // Not available on OC MCP
Var 236, name O_IASSYMB // Flashing A or B in front of speed in IAS window, not
available in OC MCP: 0 = No warning, 1 = Flashing A, 2 = Flashing B
Var 238, name O_BACKLIGHT // Backlight: 1 = on, 0 = off
// ROTARY ENCODERS
// The rotary encoders control the MCP values. The actual rotary encoder
variables are not read or written by Prosim.
// The rotary encoders change the MCP values. Before doing this, they will check
the a sociated inhibit variable, to
// see if the display is not blanked. If the display is blanked, the rotary
encoder does nothing.
```



SIOC-PROSIM

```

Var 300, name E_IAS, Link IOCARD_ENCODER, Input 0, Aceleration 4, Type 2
{
If &Inhibit_IAS = 0
{
L0 = -1 * V300
If &DECIMAL = 1
{
V102 = LIMIT 50, 90, L0
}
}
Else
{
V102 = LIMIT 110, 400, L0
}
}
}
Var 302, name E_ALT, Link IOCARD_ENCODER, Input 2, Aceleration 4, Type 2
{
If &Inhibit_Alt = 0
{
L0 = -100 * V302
V106 = LIMIT 0 ,50000, L0
}
}
}
Var 304, name E_COURSE1, Link IOCARD_ENCODER, Input 4, Aceleration 4, Type 2
{
If &Inhibit_OBS1 = 0
{
L0 = V304 * -1
V100 = ROTATE 0, 359, L0
}
}
}
Var 306, name E_HDG, Link IOCARD_ENCODER, Input 6, Aceleration 4, Type 2
{
If &Inhibit_HDG = 0
{
L0 = V306 * -1
V104 = ROTATE 0, 359, L0
}
}
}
Var 308, name E_COURSE2, Link IOCARD_ENCODER, Input 9, Aceleration 4, Type 2
{
If &Inhibit_OBS2 = 0
{
L0 = V308 * -1
V110 = ROTATE 0, 359, L0
}
}
}
Var 310, name E_VS, Link IOCARD_ENCODER, Input 11, Aceleration 4, Type 2
{
If &Inhibit_VS = 0
{
L0 = V310 * -50

```

## SIOC-PROSIM

```
V108 = LIMIT -7000, 7000, L0
}
Else
{
&VS_Trigger = 1
}
}
// Display Inhibit
// Display inhibit variables signal the blanking of the displays. They are
written by Prosim
Var 320, name Inhibit_IAS
{
If V320 = 1
{
&D_IAS = -999999
V201 = 0
}
}
Else
{
V201 = &DECIMAL
}
}
Var 322, name Inhibit_Alt
{
If V322 = 1
{
&D_ALT = -999999
}
}
}
Var 324, name Inhibit_OBS1
{
If V324 = 1
{
&D_COURSE1 = -999999
}
}
}
Var 326, name Inhibit_HDG
{
If V326 = 1
{
&D_HDG = -999999
}
}
}
Var 328, name Inhibit_OBS2
{
If V328 = 1
{
&D_COURSE2 = -999999
}
}
}
Var 330, name Inhibit_VS
{
```

SIOC-PROSIM

```

If V330 = 1
{
&D_VS = -999999
}
}
// SWITCHES
// The MCP switches are read by Prosim
Var 400, name I_CO, Link IOCARD_SW, Input 13
Var 402, name I_FD2, Link IOCARD_SW, Input 14
Var 404, name I_DISENGAGE, Link IOCARD_SW, Input 15
Var 406, name I_FD1, Link IOCARD_SW, Input 16
Var 408, name I_CWSB, Link IOCARD_SW, Input 18
Var 410, name I_CMDB, Link IOCARD_SW, Input 19
Var 412, name I_CWSA, Link IOCARD_SW, Input 20
Var 414, name I_CMDA, Link IOCARD_SW, Input 21
Var 416, name I_VS, Link IOCARD_SW, Input 22
Var 418, name I_ALTHLD, Link IOCARD_SW, Input 23
Var 420, name I_APP, Link IOCARD_SW, Input 24
Var 422, name I_VORLOC, Link IOCARD_SW, Input 25
Var 424, name I_LNAV, Link IOCARD_SW, Input 27
Var 426, name I_HDGSEL, Link IOCARD_SW, Input 28
Var 428, name I_LVLCHG, Link IOCARD_SW, Input 29
Var 430, name I_VNAV, Link IOCARD_SW, Input 30
Var 432, name I_SPEED, Link IOCARD_SW, Input 31
Var 434, name I_N1, Link IOCARD_SW, Input 32
Var 436, name I_AT, Link IOCARD_SW, Input 33
Var 438, name I_SPD_INTER // Speed intervention, not available on OC MCP
Var 440, name I_ALT_INTER // Alt intervention, not available on OC MCP
Var 442, name I_BANK_LIMIT // Bank limit, not available on MCP, values are 10,
15, 20, 25 or 30
//=====
// End Of MCP device
//=====

// *****
// Begin Transponder
// *****
Var 6100 Link FSUIPC_OUT Offset $0354 Length 2 // SetATC
Var 6101 Link FSUIPC_OUT Offset $53fe Length 2 // ProSIM
// Global var
Var 6102 name XPDRFreqLow Value 00
Var 6103 name XPDRFreqHigh value 12
Var 6104 name XPDRFreq Value 1200
//DIGITS
Var 6105 name ATC_D_ACTIVE0, Link IOCARD_DISPLAY, Device 6, Digit 0, Numbers 1
Var 6106 name ATC_D_ACTIVE1, Link IOCARD_DISPLAY, Device 6, Digit 1, Numbers 1
Var 6107 name ATC_D_ACTIVE2, Link IOCARD_DISPLAY, Device 6, Digit 2, Numbers 1
Var 6108 name ATC_D_ACTIVE3, Link IOCARD_DISPLAY, Device 6, Digit 3, Numbers 1
Var 6109, name ATC_D_FAIL, Link IOCARD_DISPLAY, Device 6, Digit 4, Numbers 1
// -999997 for 'F' letter in D_FAIL display
Var 6110, name ATC_D_BRIGHT, Link IOCARD_DISPLAY, Device 6, Digit 16, Numbers 3
// OUTPUTS

```

SIOC-PROSIM

```

Var 6111, name ATC_DIS_ATC, Link IOCARD_OUT, Device 6, Output 20
Var 6112, name ATC_DIS_1, Link IOCARD_OUT, Device 6, Output 21
Var 6113, name ATC_DIS_2, Link IOCARD_OUT, Device 6, Output 22
Var 6114, name ATC_LED_FAIL, Link IOCARD_OUT, Device 6, Output 23
Var 6115, Value 0 // INIT
{
&ATC_E_RIGHT = 00
&ATC_E_LEFT = 00
CALL &outXPDRFreq
&ATC_D_BRIGHT = 40
&ATC_DIS_ATC = 1
if &I_XPNDR = 0
{
&ATC_DIS_1 = 1
&ATC_DIS_2 = 0
}
}
// ROTARY ENCODERS
Var 6116, name ATC_E_RIGHT, Link IOCARD_ENCODER, Device 6, Input 10, Aceleration
1, Type 2
{
L0 = &ATC_E_RIGHT * -1
&XPDRFreqLow = ROTATE 0 77 L0
CALL &CalcXPDRFreqL L0
}
Var 6117, name ATC_E_LEFT, Link IOCARD_ENCODER, Device 6, Input 5, Aceleration
1, Type 2
{
L0 = &ATC_E_LEFT * -1
&XPDRFreqHigh = ROTATE 0 77 L0
CALL &CalcXPDRFreqH L0
}
// SWITCHES
Var 6118, name I_IDENT, Link IOCARD_SW, Device 6, Input 12
{
if &I_IDENT = 1
{
&sb3_ident = 1
}
ELSE
{
&sb3_ident = 0
}
}
Var 6119, name sb3_ident, LINK FSUIPC_INOUT, Offset $7b93, Length 1 // Squawkbox
Ident push
{
if &sb3_ident = 1
{
&ATC_LED_FAIL = 1
}
ELSE

```

SIOC-PROSIM

```

{
&ATC_LED_FAIL = 0
}
}
Var 6120, name I_ALTSRC, Link IOCARD_SW, Device 6, Input 9
{
if &I_ALTSRC = 1
{
&ATC_D_FAIL = -999997
&ATC_D_ACTIVE0 = 8
&ATC_D_ACTIVE1 = 8
&ATC_D_ACTIVE2 = 8
&ATC_D_ACTIVE3 = 8
&ATC_LED_FAIL = 1
&ATC_DIS_ATC = 1
&ATC_DIS_1 = 1
&ATC_DIS_2 = 1
}
ELSE
{
&ATC_D_FAIL = -999999
&ATC_LED_FAIL = 0
&ATC_DIS_ATC = 1
if &I_XPNDR = 0
{
&ATC_DIS_1 = 1
&ATC_DIS_2 = 0
}
else
{
&ATC_DIS_1 = 0
&ATC_DIS_2 = 1
}
CALL &OutXPDRFreq
}
}
Var 6121, name I_XPNDR, Link IOCARD_SW, Device 6, Input 7
{
if &I_XPNDR = 0
{
&ATC_DIS_1 = 1
&ATC_DIS_2 = 0
}
ELSE
{
&ATC_DIS_1 = 0
&ATC_DIS_2 = 1
}
}
Var 6122, name I_R_STB, Link IOCARD_SW, Device 6, Input 0
{
if &I_R_STB = 1

```

SIOC-PROSIM

```
{
V6101 = 50
}
}
Var 6123, name I_R_ALT, Link IOCARD_SW, Device 6, Input 1
{
if &I_R_ALT = 1
{
V6101 = 51
}
}
Var 6124, name I_R_XPN, Link IOCARD_SW, Device 6, Input 2
{
if &I_R_XPN = 1
{
V6101 = 52
}
}
Var 6125, name I_R_TAO, Link IOCARD_SW, Device 6, Input 3
{
if &I_R_TAO = 1
{
V6101 = 53
}
}
Var 6126, name I_R_TAR, Link IOCARD_SW, Device 6, Input 4
{
if &I_R_TAR = 1
{
V6101 = 54
}
}
var 6127, name sb_connect, LINK FSUIPC_INOUT, Offset $7b81, Length 1 //
Squawkbox connected
{
if &sb_connect = 1
{
&ATC_DIS_2 = 1
}
ELSE
{
&ATC_DIS_2 = 0
}
}
var 6128, name pbrake, LINK FSUIPC_INOUT, Offset $0bc8, Length 2 // Parking
Brake status
{
if &pbrake = 0
{
&ATC_LED_FAIL = 0
}
ELSE
```

SIOC-PROSIM

```

{
&ATC_LED_FAIL = 1
}
}
//sub routines code
Var 6129 name OutXPDRFreq Link SUBROUTINE
{
L0 = TOBCD &XPDRFreq
V6100 = L0
L1 = MOD L0 16
&ATC_D_ACTIVE0 = L1
L0 = DIV L0 16
&ATC_D_ACTIVE1 = MOD L0 16
L0 = DIV L0 16
&ATC_D_ACTIVE2 = MOD L0 16
L0 = DIV L0 16
&ATC_D_ACTIVE3 = MOD L0 16
}
Var 6130 name CalcXPDRFreqL Link SUBROUTINE
{
L0 = &XPDRFreqLow
L2 = DIV L0 10
L1 = MOD L0 10
IF L1 = 8
{
IF &CalcXpdrFreqL > 0
{
L2 = L2 + 1
L1 = 0
}
ELSE
{
L1 = 6
}
}
ELSE
{
IF L1 = 9
{
IF &CalcXpdrFreqL > 0
{
L2 = L2 + 1
L1 = 1
}
ELSE
{
L1 = 7
}
}
}
L0 = L2 * 10
&XPDRFreqLow = L0 + L1

```

SIOC-PROSIM

```

L2 = &XPDRFreqHigh * 100
L1 = &XPDRFreqLow
&XPDRFreq = L1 + L2
CALL &OutXPDRFreq
}
Var 6131 name CalcXPDRFreqH Link SUBROUTINE
{
L0 = &XPDRFreqHigh
L2 = DIV L0 10
L1 = MOD L0 10
IF L1 = 8
{
IF &CalcXpdrFreqH > 0
{
L2 = L2 + 1
L1 = 0
}
ELSE
{
L1 = 6
}
}
ELSE
{
IF L1 = 9
{
IF &CalcXpdrFreqH > 0
{
L2 = L2 + 1
L1 = 1
}
ELSE
{
L1 = 7
}
}
}
L0 = L2 * 10
&XPDRFreqHigh = L0 + L1
L2 = &XPDRFreqHigh * 100
L1 = &XPDRFreqLow
&XPDRFreq = L1 + L2
CALL &OutXPDRFreq
}
//
*****
// End Transponder
//
*****
// Engine Start AUTO-OFF
Var 2000, name iniciaeng1, static, Value 0 // Set engine 1 rotary switch to OFF
{

```



SIOC-PROSIM

```

&eng1 = 0
&servo1 = 1023
}
Var 2001, name servo1, static, Link USB_SERVOS, Device 2, Output 1, PosL 854,
PosC 900, PosR 1023
{
&servo1 = DELAY 854 ,50
&eng1 = 0
}
Var 2002, name eng1, static // var 2000
{
IF &eng1 = 1
{
&servo1 = DELAY 1023 ,5500
}
}
Var 2003, name iniciaeng2, static, Value 0 // Set engine 2 rotary switch to OFF
{
&eng2 = 0
&servo2 = 1023
}
Var 2004, name servo2, static, Link USB_SERVOS, Device 2, Output 1, PosL 854,
PosC 900, PosR 1023
{
&servo2 = DELAY 854 ,50
&eng2 = 0
}
Var 2005, name eng2, static
{
IF &eng2 = 1
{
&servo2 = DELAY 1023 ,5500 // var 2500 foer
}
}
Var 801, name fueltemp, Link USB_SERVOS, Device 2, Output 2, PosL 281, PosC 511,
PosR 1023
//Var 802, name duct, Link USB_SERVOS, Device 2, Output 5, PosL 180, PosC 511,
PosR 1023
//Var 803, name fueltemp, Link USB_SERVOS, Device 2, Output 1, PosL 180, PosC
511, PosR 1023
//Var 804, name engstart, Link USB_SERVOS, Device 2, Output 6, PosL 854, PosC
900, PosR 1023
//Var 805, name diffpress, Link USB_SERVOS, Device 2, Output 2, PosL 290, PosC
900, PosR 1023
//Var 806, name cabinalt, Link USB_SERVOS, Device 2, Output 3, PosL 290, PosC
900, PosR 1023

// USB Output card
Var 8012, Link IOCARD_OUT, Device 1, Output 0 // Electric 2 Low Pressure
Var 8013, Link IOCARD_OUT, Device 1, Output 1 // Hyd Pump Eng 2 Low Pressure
Var 8014, Link IOCARD_OUT, Device 1, Output 2 // L Aplha Valve

```

SIOC-PROSIM

Var 8015, Link IOCARD\_OUT, Device 1, Output 3 // Aft Entry  
 Var 8016, Link IOCARD\_OUT, Device 1, Output 4 // EPUIQ  
 Var 8017, Link IOCARD\_OUT, Device 1, Output 5 // Aft Service  
 Var 8018, Link IOCARD\_OUT, Device 1, Output 6 // R F Overving  
 Var 8019, Link IOCARD\_OUT, Device 1, Output 7 // L F Overving  
 Var 8020, Link IOCARD\_OUT, Device 1, Output 8 // FWD Service  
 Var 8021, Link IOCARD\_OUT, Device 1, Output 9 // Aft Cargo  
 Var 8022, Link IOCARD\_OUT, Device 1, Output 10 // FWD Cargo  
 Var 8023, Link IOCARD\_OUT, Device 1, Output 11 // R Aft Overving  
 Var 8026, Link IOCARD\_OUT, Device 1, Output 12 // L Aft Overving  
 Var 8027, Link IOCARD\_OUT, Device 1, Output 13 // FWD Entry  
 Var 8032, Link IOCARD\_OUT, Device 1, Output 14 // Wing Anti ice L R Valve open  
 Var 8035, Link IOCARD\_OUT, Device 1, Output 15 // ENG Anti Ice COWL R Open  
 Var 8036, Link IOCARD\_OUT, Device 1, Output 16 // ENG Anti Ice COWL L Open  
 Var 8038, Link IOCARD\_OUT, Device 1, Output 17 // ENG Anti Ice L COWL  
 Var 8039, Link IOCARD\_OUT, Device 1, Output 18 // ENG Anti Ice R COWL  
 Var 8040, Link IOCARD\_OUT, Device 1, Output 19 // Cont Cab Zone  
 Var 8041, Link IOCARD\_OUT, Device 1, Output 20 // FWD Cab Zone  
 Var 8042, Link IOCARD\_OUT, Device 1, Output 21 // Aft Cab Zone  
 Var 8044, LINK IOCARD\_OUT, Device 1, Output 22 // L Ram Dor  
 Var 8045, LINK IOCARD\_OUT, Device 1, Output 24 // Rram Door  
 Var 8047, LINK IOCARD\_OUT, Device 1, Output 23 // Dual Bleed  
 Var 8048, LINK IOCARD\_OUT, Device 1, Output 25 // L Pack  
 Var 8049, LINK IOCARD\_OUT, Device 1, Output 26 // R Pack  
 Var 8051, LINK IOCARD\_OUT, Device 1, Output 27 // L Wing Body OVH  
 Var 8052, LINK IOCARD\_OUT, Device 1, Output 28 // R Wing Body OVH  
 Var 8053, LINK IOCARD\_OUT, Device 1, Output 29 // Auto Fail  
 Var 8054, LINK IOCARD\_OUT, Device 1, Output 30 // Off Sch Decent  
 Var 8055, LINK IOCARD\_OUT, Device 1, Output 31 // ALTN  
 Var 8056, LINK IOCARD\_OUT, Device 1, Output 32 // MANUAL  
 Var 8057, LINK IOCARD\_OUT, Device 1, Output 33 // Capt Pitot  
 Var 8058, LINK IOCARD\_OUT, Device 1, Output 34 //  
 Var 8059, LINK IOCARD\_OUT, Device 1, Output 35 // AUX Pitot  
 Var 8060, LINK IOCARD\_OUT, Device 1, Output 36 // Temp Probe  
 Var 8061, LINK IOCARD\_OUT, Device 1, Output 37 // R Alpha Valve  
 Var 8062, LINK IOCARD\_OUT, Device 1, Output 38 // R Elev Pitot  
 Var 8063, LINK IOCARD\_OUT, Device 1, Output 39 // L elev Pitot  
 Var 8064, LINK IOCARD\_OUT, Device 1, Output 40 // Hyd Pump Elec2 OVH  
 Var 8065, LINK IOCARD\_OUT, Device 1, Output 41 // Hyd Pump Elec1 Low P  
 Var 8066, LINK IOCARD\_OUT, Device 1, Output 42 // Hyd Pump Elec1 OVH  
 Var 8067, LINK IOCARD\_OUT, Device 1, Output 43 // GEN 1 Bus Off  
 Var 8068, LINK IOCARD\_OUT, Device 1, Output 44 // Gen1 Source Off  
 Var 8069, LINK IOCARD\_OUT, Device 1, Output 45 // Gen2 Source Off  
 Var 8070, LINK IOCARD\_OUT, Device 1, Output 46 // Gen2 Bus Off  
 Var 8071, LINK IOCARD\_OUT, Device 1, Output 47 // APU Gen Bus Off  
 Var 8072, LINK IOCARD\_OUT, Device 1, Output 48 // Apu Maint  
 Var 8073, LINK IOCARD\_OUT, Device 1, Output 49 // Apu Low Pressure  
 Var 8074, LINK IOCARD\_OUT, Device 1, Output 50 // Apu Fault  
 Var 8075, LINK IOCARD\_OUT, Device 1, Output 51 // Apu Overspeed  
 Var 8076, LINK IOCARD\_OUT, Device 1, Output 52 // Fuel L Filter Bypass  
 Var 8077, LINK IOCARD\_OUT, Device 1, Output 53 // Fuel C L Off  
 Var 8078, LINK IOCARD\_OUT, Device 1, Output 54 // Fuel C R Off

SIOC-PROSIM

```

Var 8079, LINK IOCARD_OUT, Device 1, Output 55 // Fuel R Filter Bypass
Var 8080, LINK IOCARD_OUT, Device 1, Output 56 // R ENG Valve Closed
Var 8081, LINK IOCARD_OUT, Device 1, Output 57 // Fuel L Aft
Var 8082, LINK IOCARD_OUT, Device 1, Output 58 // Fuel R Af
Var 8083, LINK IOCARD_OUT, Device 1, Output 59 // Fuel L FWD
Var 8084, LINK IOCARD_OUT, Device 1, Output 60 // L ENG Valve Closed
Var 8085, LINK IOCARD_OUT, Device 1, Output 61 // Fuel Crossfeed
Var 8086, LINK IOCARD_OUT, Device 1, Output 62 // Fuel R FWD
// ****
// ADF1
// ****
Var 7600 Value 0
{
&A1StbLow = 5
&A1StbHigh = 385
CALL &CaA1StbFreq
}
Var 7601 name X_A1ActHigh Link FSUIPC_INOUT Offset $034C Length 2
{
L0 = FROMBCD &X_A1ActHigh
&A1ActHigh = L0
CALL &CaA1ActFreq
}
Var 7602 name X_A1ActLow Link FSUIPC_INOUT Offset $0356 Length 2
{
&A1ActLow = &X_A1ActLow
CALL &CaA1ActFreq
}
Var 7603 name A1ActHigh // FSUIPC active ADF1 value high: 3 digits
Var 7604 name A1ActLow // FSUIPC active ADF1 value: 1 digit
Var 7605 name A1ActFreq // total (needed for display): 4 digits
Var 7606 name A1StbHigh // FSUIPC standby ADF1 value high: 3 digits
Var 7607 name A1StbLow // FSUIPC standby ADF1 value: 1 digit
Var 7608 name A1StbFreq // total (needed for display): 4 digits
Var 7609 name CaA1ActFreq Link Subroutine
{
L0 = &A1ActHigh * 10 // high * 10
&A1ActFreq = L0 + &A1ActLow // + decimal
CALL &OutA1Act
}
Var 7610 name CaA1StbFreq Link Subroutine
{
L0 = &A1StbHigh * 10 // high * 10
&A1StbFreq = L0 + &A1StbLow // + decimal
CALL &OutA1Stb
}
Var 7611 name OutA1Act Link SUBROUTINE
{
&D_A1Act = &A1ActFreq
&O_A1ActDp = 1
}
Var 7612 name OutA1Stb Link SUBROUTINE

```

SIOC-PROSIM

```

{
&D_A1Stb = &A1StbFreq
&O_A1StbDp = 1
}
Var 7613 name RO_A1StbHigh, Link IOCARD_ENCODER, device 7, Input 0, Aceleration
4, Type 2
{
L0 = &RO_A1StbHigh * -1 // change direction (turning right should be plus)
&A1StbHigh = ROTATE 100 999 L0
CALL &CaA1StbFreq
}
Var 7614 name RO_A1StbLow, Link IOCARD_ENCODER, device 7 Input 4, Aceleration 1,
Type 2
{
L0 = &RO_A1StbLow * 1 // in steps of 5
&A1StbLow = ROTATE 0 9 L0
CALL &CaA1StbFreq
}
Var 7615 name FreqSwap1, Link IOCARD_SW, device 7, Input 6, Type P
{
L0 = &A1ActHigh
L1 = &A1ActLow
&X_A1ActLow = &A1StbLow // decimal digit to fsuipc
&X_A1ActHigh = TOBCD &A1StbHigh // higher 3 Digits in bcd to fsuipc
&A1StbHigh = L0
&A1StbLow = L1
CALL &CaA1StbFreq
}
Var 7616 name D_A1Act, Link IOCARD_DISPLAY, device 7, Digit 0, Numbers 5
Var 7617 name O_A1ActDp, Link IOCARD_OUT, device 7, Output 20
Var 7618 name D_A1Stb, Link IOCARD_DISPLAY, device 7, Digit 5, Numbers 5
Var 7619 name O_A1StbDp, Link IOCARD_OUT, device 7, Output 21

```