

```
-----  
-- add 20210813Monitor the switches on the Pendant.  
-----
```

```
SigLib = {  
  [mc.ISIG_INPUT0] = function (state)  
    PendantAxisChange()      --X軸呼び出し関数  
  end,  
  
  [mc.ISIG_INPUT1] = function (state)  
    PendantAxisChange()      --Y軸呼び出し関数  
  end,  
  
  [mc.ISIG_INPUT2] = function (state)  
    PendantAxisChange()      --Z軸呼び出し関数  
  end,  
  
  [mc.ISIG_INPUT3] = function (state)  
    PendantSpeedChange()     -- × 1 呼び出し関数  
  end,  
  
  [mc.ISIG_INPUT4] = function (state)  
    PendantSpeedChange()     -- × 10呼び出し関数  
  end,  
  
  [mc.ISIG_INPUT5] = function (state)  
    PendantSpeedChange()     -- × 1 0 0 呼び出し関数  
  end,  
}
```

```
-----  
#NAME?  
-----
```

```
  [mc.ISIG_INPUT6] = function (state)  --Performs a Cycle Stop  
    InputCycleStop()  
  end,
```

```
  [mc.ISIG_INPUT7] = function (state)  --Zero All  
    InputZeroAll()  
  end,
```

```
}
```

```
-----  
#NAME?  
-----
```

```
function InputZeroAll() --This will do Zero All button push
```

```
  mc.mcAxisSetPos(inst, mc.X_AXIS,0.0);  
  mc.mcAxisSetPos(inst, mc.Y_AXIS,0.0);  
  mc.mcAxisSetPos(inst, mc.Z_AXIS,0.0);  
  mc.mcCntlSetLastError(inst, "Input Zero All")
```

```
end
```

```
-----  
function InputCycleStop() --This will do a cycle stop because of a button push
```

```
    CycleStop()
```

```
    mc.mcCntlSetLastError(inst, "Input Cycle Stop")
```

```
end
```

```
--for the pendant assigned to MPG#7:
```

```
local MpgNo = 7
```

```
#NAME?
```

```
local UnmapMPG = -1
```

```
-----  
--This will process the Pendant's Rate/Speed Selection Switch
```

```
-----  
function PendantSpeedChange()
```

```
    local hX1, rc = mc.mcSignalGetHandle(inst, mc.ISIG_INPUT3)    --INPUT6 for the X1 Selection
```

```
    local Step1, rc = mc.mcSignalGetState(hX1)
```

```
    local hX10, rc = mc.mcSignalGetHandle(inst, mc.ISIG_INPUT4)    --INPUT7 for the X10 Selection
```

```
    local Step10, rc = mc.mcSignalGetState(hX10)
```

```
    local hX100, rc = mc.mcSignalGetHandle(inst, mc.ISIG_INPUT5)    --INPUT8 for the X100 Selection
```

```
    local Step100, rc = mc.mcSignalGetState(hX100)
```

```
    local msg
```

```
    if (Step1 == 1) then
```

```
        mc.mcMpgSetInc(inst, MpgNo, 1.0)
```

```
        msg = "Pendant Distance X1 = " .. 1.0
```

```
        mc.mcCntlSetLastError(inst, msg) --Show a message
```

```
    elseif (Step10 == 1) then
```

```
        mc.mcMpgSetInc(inst, MpgNo, 0.1)
```

```
        msg = "Pendant Distance X10 = " .. 0.1
```

```
        mc.mcCntlSetLastError(inst, msg) --Show a message
```

```
    elseif (Step100 == 1) then
```

```
        mc.mcMpgSetInc(inst, MpgNo, 0.01)
```

```
        msg = "Pendant Distance X100 = " .. 0.01
```

```
        mc.mcCntlSetLastError(inst, msg) --Show a message
```

```
    end
```

```
end
```

```
-----  
-- The Pendant's Axis switch changed... turn the Pendant LED ON/OFF
```

```
-----  
function PendantAxisChange()
```

```
    --Variable prep first for handles
```

```
local hAxis_X, rc = mc.mcSignalGetHandle(inst, mc.ISIG_INPUT0)
```

```
local SelectAxis_X, rc = mc.mcSignalGetState(hAxis_X)
```

```
local hAxis_Y, rc = mc.mcSignalGetHandle(inst, mc.ISIG_INPUT1)
```

```
local SelectAxis_Y, rc = mc.mcSignalGetState(hAxis_Y)
```

```
local hAxis_Z, rc = mc.mcSignalGetHandle(inst, mc.ISIG_INPUT2)
```

```
local SelectAxis_Z, rc = mc.mcSignalGetState(hAxis_Z)
```

```
-----  
--This is the Pendant's LED
```

```
    local hLedAxis, rc = mc.mcSignalGetHandle(inst, mc.OSIG_OUTPUT0)-- Pendant LED Port 3 Pin 1
```

```
    local hLedYAxis, rc = mc.mcSignalGetHandle(inst, mc.OSIG_OUTPUT1)-- Pendant LED Port 3 Pin 5
```

```
    local hLedZAxis, rc = mc.mcSignalGetHandle(inst, mc.OSIG_OUTPUT2)-- Pendant LED Port 3 Pin 6
```

```
--軸選択 code
```

```
if (SelectAxis_X == 1) then
```

```
    mc.mcMpgSetAxis(inst, MpgNo, 0)  --Map the MPG to control the X Axis
```

```
    mc.mcCntlSetLastError(inst, "Pendant Axis X Selected") --Show a message in the Screen Set
```

```
    mc.mcSignalSetState(hLedAxis, 1)  --X Axis LED ON
```

```
        mc.mcSigr #NAME?
```

```
        mc.mcSigr #NAME?
```

```
elseif (SelectAxis_Y== 1) then
```

```
    mc.mcMpgSetAxis(inst, MpgNo, 1)  --Map the MPG to control the Y Axis
```

```
    mc.mcCntlSetLastError(inst, "Pendant Axis Y Selected") --Show a message in the Screen Set
```

```
    mc.mcSignalSetState(hLedYAxis, 1)  --Y Axis LED ON
```

```
        mc.mcSignalSetState(hLedAxis, 0)  --X Axis LED OFF
```

```
        mc.mcSigr #NAME?
```

```
elseif (SelectAxis_Z== 1) then
```

```
    mc.mcMpgSetAxis(inst, MpgNo,2)  --Map the MPG to control the Z Axis
```

```
    mc.mcCntlSetLastError(inst, "Pendant Axis Z Selected") --Show a message in the Screen Set
```

```
    mc.mcSignalSetState(hLedZAxis, 1)  --Z Axis LED ON
```

```
        mc.mcSignalSetState(hLedAxis, 0)  --X Axis LED OFF
```

```
        mc.mcSigr #NAME?
```

```
else
```

```
    --None of the Axis Switch inputs are active, so unmap the MPG from all axes (by assigning an invalid axis -1)
```

```
    mc.mcMpgSetAxis(inst, MpgNo, UnmapMPG )  --Unmap the MPG, so it won't control any axes
```

```
    mc.mcCntlSetLastError(inst, "NO Pendant Axis Selected") --Show a message in the Screen Set
```

```
    mc.mcSignalSetState(hLedAxis, 0)  --X Axis LED OFF
```

```
        mc.mcSigr #NAME?
```

```
        mc.mcSigr #NAME?
```

```
end
```

```
end
```

PendantSpeedChange()

PendantAxisChange()