

In Part 3 we are going to show you how to create a PC program using Zeus to talk to the Slave Proc we created in Part 2.

We have already done all the work on the Slave Proc so it is a simple matter to create a Zeus program to take advantage of our work.

### Step A: ReadPort

This Zeus program will read the state of port 0 on slave procs with an IDof 10, 11, ad 12.

```
'Chip to Chip Program 3A
func main()

    dim Chip10,Chip11,Chip12
    dim x
    Chip10=0
    Chip11=0
    Chip12=0

    InitSlaves(1,8)

loop:

    print "Chip10","Chip11","Chip12"
    print "-----","-----","-----"
    for x = 1 to 10
        Chip10=ReadPort(10,0)
        Chip11=ReadPort(11,0)
        Chip12=ReadPort(12,0)

        print Chip10,Chip11,Chip12
    next
    print
endfunc

'-----
'Set up Zeus Master to Slave Interface Demo
'Pass
' Channel,Port
func InitSlaves(x as integer,y as integer) as integer
    global SlavesChannel as integer

    SlavesChannel = x
    if ComOpen(SlavesChannel,baud=9600,port = y) = 0 then
        if MsgBox("Error Opening Com Port.  Quit?",3) = 6 then
            end
        endif
        exit 0
    endif
    ComSettings SlavesChannel,Timeout = 1,Priority = 500
    ComBGSuspend SlavesChannel,0
```

```

'Do this incase you forget to remove DTR Enable
ComDTR(SlavesChannel,0)
Pause 100 'When enabled chip needs time to recover
ComPurge SlavesChannel
exit 1
endfunc

'-----
'Return the state of a port
'Pass
' ID, PORT
func ReadPort(y as integer ,z as integer) as integer
    ComPurge SlavesChannel

    '          Channel      ID      CMD      Data
    ComOutput SlavesChannel,chr(y)+chr(2)+chr(z)
    exit ComWaitForByte(SlavesChannel,500)
endfunc

```

The heart of the program is the InitSlaves function. This function opens the comport and sets DTR so that the Nemesis is turned on. If you dont set the DTR port as shown the Nemesis will be held in the reset state

We also need to wait for 100ms to allow the Nemesis to go through its reset cycle. After that it is simple matter to send the request port state command (2).

## Step B: Output Port

In this program we set port 1 as output and toggle it high and low every 200ms.

```

'Chip to Chip Program 3B
func main()

    InitSlaves(1,8)
    PortOutput(12,1)

Loop:
    PortHigh(12,1)
    Pause 200
    PortLow(12,1)
    Pause 200
    Goto Loop

endfunc

'-----
'Set up Zeus Master to Slave Interface Demo
'Pass
' Channel,Port
func InitSlaves(x as integer,y as integer) as integer
    global SlavesChannel as integer

```

```

SlavesChannel = x
if ComOpen(SlavesChannel,baud=9600,port = y) = 0 then
    if MsgBox("Error Opening Com Port.  Quit?",3) = 6 then
        end
    endif
    exit 0
endif
ComSettings SlavesChannel,Timeout = 1,Priority = 500
ComBGSuspend SlavesChannel,0

'Do this incase you forget to remove DTR Enable
ComDTR(SlavesChannel,0)
Pause 100 'When enabled chip needs time to recover
ComPurge SlavesChannel
exit 1
endfunc

'-----
'Set a port to output
'Pass
' ID, PORT
func PortOutput(y as integer,z as integer)
    ComOutput SlavesChannel,chr(y)+chr(1)+chr(z)
endfunc

'-----
'Set a port to input
'Pass
' ID, PORT
func PortInput(y as integer,z as integer)
    ComOutput SlavesChannel,chr(y)+chr(0)+chr(z)
endfunc

'-----
'Set a port High
'Pass
' ID, PORT
func PortHigh(y as integer,z as integer)
    ComOutput SlavesChannel,chr(y)+chr(5)+chr(z)
endfunc

'-----
'Set a port Low
'Pass
' ID, PORT
func PortLow(y as integer,z as integer)
    ComOutput SlavesChannel,chr(y)+chr(4)+chr(z)
endfunc

```

### Step C: AtoD Input

Here we put a port into AtoD mode and display the result.

```

'Chip to Chip Program 3C
func main()

    InitSlaves(1,8)
    PortAtoD(12,1)

Loop:
    print ReadAtoD(12,1)
    Goto Loop

endfunc

'-----
'Set up Zeus Master to Slave Interface Demo
'Pass
' Channel,Port
func InitSlaves(x as integer,y as integer) as integer
    global SlavesChannel as integer

    SlavesChannel = x
    if ComOpen(SlavesChannel,baud=9600,port = y) = 0 then
        if MsgBox("Error Opening Com Port. Quit?",3) = 6 then
            end
        endif
        exit 0
    endif
    ComSettings SlavesChannel,Timeout = 1,Priority = 500
    ComBGSuspend SlavesChannel,0

    'Do this incase you forget to remove DTR Enable
    ComDTR(SlavesChannel,0)
    Pause 100 'When enabled chip needs time to recover
    ComPurge SlavesChannel
    exit 1
endfunc

'-----
'Set a port to output
'Pass
' ID, PORT
func PortAtoD(y as integer,z as integer)
    ComOutput SlavesChannel,chr(y)+chr(7)+chr(z)
endfunc

'-----
'Return the state AtoD Value
'Pass
' ID, PORT
func ReadAtoD(y as integer ,z as integer) as integer
    ComPurge SlavesChannel

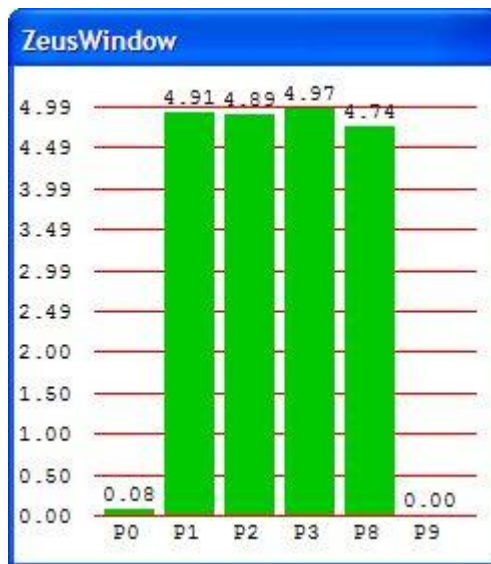
    '          Channel      ID      CMD      Data
    ComOutput SlavesChannel,chr(y)+chr(10)+chr(z)
    dim th
    dim tl
    dim tv
    th = ComWaitForByte(SlavesChannel,500)
    tl = ComWaitForByte(SlavesChannel,500)

```

```
    exit th * 256 + tl
endfunc
```

## Step D: More AtoD

In this program we are going to read 5 AtoD ports and display the data graphically.



```
'Chip to Chip Program 3D
func main()

    InitSlaves(1,8)

    PortAtoD(12,0)
    PortAtoD(12,1)
    PortAtoD(12,2)
    PortAtoD(12,3)
    PortAtoD(12,8)
    PortAtoD(12,9)
    dim x,z

    FormUpdateAutoOff

Loop:

    FormCls
    FormPen 255,0,0
    for x = 0 to 1024 step 102
        FormDrawText 0,204-(x/5)+13,Format(Float(x *.00489),"0.00")
        FormLine 40,204-(x/5)+20,230,204-(x/5)+20
    next

    DrawBar(45,ReadAtoD(12,0),"P0")
```

```

DrawBar(75,ReadAtoD(12,1),"P1")
DrawBar(105,ReadAtoD(12,2),"P2")
DrawBar(135,ReadAtoD(12,3),"P3")
DrawBar(165,ReadAtoD(12,8),"P8")
DrawBar(195,ReadAtoD(12,9),"P9")

FormUpdate
Goto Loop

endfunc

func DrawBar(Pos as integer,Tvalue as integer,Pvalue as string)
FormDrawText Pos-3,204-(Tvalue/5)+5,Format(Float(Tvalue *.00489),"0.00")
FormBrush 0,200,0
FormFillRectangle Pos,204-(Tvalue/5)+20,25,Tvalue/5
FormDrawText Pos+2,225,PValue
endfunc

'-----
'Set up Zeus Master to Slave Interface Demo
'Pass
' Channel,Port
func InitSlaves(x as integer,y as integer) as integer
global SlavesChannel as integer

SlavesChannel = x
if ComOpen(SlavesChannel,baud=9600,port = y) = 0 then
    if MsgBox("Error Opening Com Port. Quit?",3) = 6 then
        end
    endif
    exit 0
endif
ComSettings SlavesChannel,Timeout = 1,Priority = 500
ComBGSuspend SlavesChannel,0

'Do this incase you forget to remove DTR Enable
ComDTR(SlavesChannel,0)
Pause 100 'When enabled chip needs time to recover
ComPurge SlavesChannel
exit 1
endfunc

'-----
'Set a port to output
'Pass
' ID, PORT
func PortAtoD(y as integer,z as integer)
ComOutput SlavesChannel,chr(y)+chr(7)+chr(z)
endfunc

'-----
'Return the state AtoD Value
'Pass
' ID, PORT
func ReadAtoD(y as integer ,z as integer) as integer
ComPurge SlavesChannel

'          Channel      ID      CMD      Data
ComOutput SlavesChannel,chr(y)+chr(10)+chr(z)
dim th
dim tl

```

```
dim tv
th = ComWaitForByte(SlavesChannel,500)
tl = ComWaitForByte(SlavesChannel,500)

exit th * 256 + tl
endfunc
```

This program uses double buffering so you will have to have zeus pro in order to try this program. Also note that the Nemesis only has AtoD on Ports 0,1,2,3,8,9, and 10.

## Parts

Athena Microcontroller           Kronos Robotics #16276

Nemesis Microcontroller        Kronos Robotics #16406

Athena WorkBoard DeluxeKronos Robotics #16457

Athena Compiler                 <http://www.kronosrobotics.com/downloads/AthenaSetup.exe>

Zeus Pro                         <http://www.krmicros.com/Development/ZeusPro/ZeusPro.htm>