

```
; Procedure library for XXXX company Base
; List of RFlags being used:
; 108 - used to send meter and setpoint data to an RTU
; 115 - controls request for quick or complete download from an RTU
```

```
proc startup ; complete start0 functions
```

```
    echo INITIAL STARTUP OF SCADA/RTU PROGRAM
    set dir PCX ..\SCREENS\
    set dir font ..\screens\
    SET DIR db ..\db\
; set vga 39 ; 800x600x256 Default Vga Mode
; set chip 6 81 ; Trident chip 9400
; set share SERVER ; Database and Image files
    set warble on
    set color on ; Show data-only points in color when abnormal (no
blink)
    monitor off
; IMAGE file base.img server ; this has read write ability
; image load
```

```
    RTG AUTO ON ; tell RTG system to pick up text from chan
descripts
    map load 1 Modbus $.mbr ; read in Modbus mapping definition
```

```
; if @image(0) = 0
; msg Image Loaded
; else
; Msg Image not loaded. Processing individual RTU setup files
; load RTU ; load all RTUs in one command
; msg RTU setup processing complete
; gosub $.lin
; endif
```

```
task 1 ID Radio
task radio prior 4
; task radio start
```

```
task 2 id GE_PLC
; task ge_plc start
```

```
; task 3 id Net10
; task Net10 start
```

```
; task 4 id Net11
; task Net11 start
```

```
task 3 id util
task util start
```

```
set UPTASK util Lib ; use Network on all RTU updates/timestamps
; task 6 ID driver
; task driver prior 1
```

```

; task driver delay 18
; task driver start

task sec start
task scan start
task scan delay 2
> ack ; acknowledge all current alarms for all RTUs

image on 120 ; save image every 2 minutes
gosub agenda
log every 60 ; save logged data to disk every 60 secs
sele MTU
set horn HORNOUT
calc pollrtus = @high(pollrtus) ; start local timer for automatic
polling

set menu user
set tick off
; user

proc start1 ; TSP Multidrop Task Start File
msg Starting task 1
set port 1 echo 8 ; echo to port 8 for testing
set media multidrop
set fail 8 ; terminate comm link if no commands entered in
this time
set comm 600 ; process BYE automatically if no activity for
this long
set trys 4 ; number of tries to send a block and receive an
ack
set wait 3 ; time to wait for response from other unit
during callout
set first 3 ; number of secs to wait for 1st cmd after going
online
set md delay 5 4 ; allow 5 secs for CD to clear, key 9 ticks
before xmitting
set md dwell on ; keep RTS on for 1 additional tick after
transmitting
set dcd off ; read everything coming in, don't worry about
dcd
sele mtu
set md id MTU ; assume we are a go
; set alert on ; allow processing of incoming ALERT commands

proc start2 ; Modbus Multidrop Task Start File
; msg Starting task 2
; msg Setting port for task 2 to 8
; set port 8 share 1 ; get echo from port 8, real xmit on port 1
; set port 1
set media multidrop
set fail 8 ; terminate comm link if no commands entered in
this time
set comm 600 ; process BYE automatically if no activity for
this long

```

```

    set trys 4          ; number of tries to send a block and receive an
ack
    set wait 3         ; time to wait for response from other unit
during callout
    set first 3        ; number of secs to wait for 1st cmd after going
online
    set md delay 5 4   ; allow 5 secs for CD to clear, key 9 ticks
before xmitting
    set md dwell on    ; keep RTS on for 1 additional tick after
transmitting
    set dcd off        ; read everything coming in, don't worry about
dcd
    sele mtu
    set md id MTU      ; assume we are a go
; set alert on        ; allow processing of incoming ALERT commands
; map sele 1          ; set modbus map
msg task 2 started
sleep 2

```

```

proc bye2 ; disconnect for Mosbus port
; force 1 set port 1 echo 8 ; tell task 1 to send us characters
; sleep 1 ; let task 1 get setup
; set port 8 share 1 ; return to echo input from port 1
; msg Task 2 returned to echo input mode from port 1

```

```

; Network Ports

```

```

proc start9 ; NETWORK TASK
msg Starting Network Task
set port 10
msg Net10 Port set
set md id NETHOST
set first 2
set wait 2
set trys 2
set fail 10
msg Net Task startup complete

```

```

proc start10
msg Starting Network Task
set port 11
msg Net11 Port set
set md id NETREM
set first 2
set wait 2
set trys 2
set fail 10
msg Net Task startup complete

```

```

; UTILITY TASK

```

```

proc start3 ; util task
set port -1

```

```

; DO_ALL
; This is a general purpose shell routine that will repeat commands
; for all RTUS. It can be used in place of the > function for
; file reads. You cannot use > READ XXXX. The > can only be used
; for direct commands like ACK or CALC
; The usage for this is:
;     read do_all, Filename with params
; Note that a comma is used after do_all so that all that follows is
one param

```

```

proc do_all      ; repeat command for all RTUS
  local initial_rtu
  calc initial_Rtu = @getrtu(0)
  sele mtu
  gosub $1 $2 $3 $4 $5
  sele PC
  gosub $1 $2 $3 $4
  sele PF
  gosub $1 $2 $3 $4
  sele PG
  calc initial_rtu = @setrtu(initial_Rtu) ; restore original rtu
  return

```

```

; This runs when the poll timer runs down
proc pollrtus      ; automatic poll of all RTUs
  > poll now

```

```

proc set_polltime ; change timer channel's high setpoint
  cls
  if @pass(0) < 3
    beep 4
    cursor 1,5,"Password level 3 required for this function"
    sleep 3
    return
  endif

  sele mtu
  cursor 1, 4

  local W
  calc w = @high(pollrtus)

  entry clear
  entry, title, AUTOMATIC POLL TIMEER CONTROL
  entry text, 4,1, Enter the number of minutes between automatic
polls.
  entry, add, 8, 5, Poll Time in Minutes, W, 30, , 0, 180
  config entry
  if @no(0) = 1 ; if hit ESC to exit the entry screen
    release w x y z
    return

```

```
endif  
  
sethigh pollrtus $(W) ; if hit F2 to exit the entry screen  
if pollrtus > w ; if now more than total time then set lower now  
  calc pollrtus = w  
endif  
save ; save new values
```

```
proc hornout  
  horn off
```

```
; ----- Definition of Function Keys -----
```

```
proc sf1  
  msg Watch 1 turned on  
  watch 1
```

```
proc sf2  
  msg All watches turned off  
  watch off
```

```
proc sf10 ; help  
  cls  
  help
```

```
→
```