# PCM-9375 GPIO Program note

```
SMBus Base I/O Address ------------------------------6000h
          Input Register Offset-------------------------00h
          Output Register Offset------------------------01h
          Invertion Register Offset---------------------02h
          Configure Register Offset---------------------03h
```

**SMB Native Registers Map**

| SMB I/O Offset | Name | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|
| 00h | SMBSDA | SMBSDA | | | | | | | |
| 01h | SMBST | SLVSTP | SDAST | BER | NEGACK | STASTR | NMATCH | MASTER | XMIT |
| 02h | SMBCST | SVD | | GSCL | TSDA | CMTCH | MATCH | BB | BUSY |
| 03h | SMBCTL1 | STASTRE | NMINTE | GCMEN | ACK | SVD | INTEN | STOP | START |

```
GPIO Chip ID ----------------------------------------4Eh
```

```
;Chack PCM-9375 GPIO Test function.
;=====================================================================

     NEWIODELAY Macro
                out  0ebh,al
                ENDM

;=====================================================================
                .model small
                .486p
                .stack 256
                .data
;=====================================================================
;                   Data Area
;=====================================================================
      SMBUS_BASE_IO_ADDR    EQU     6000h
      PCA9554_ID            EQU     4Eh
      Input_Reg             EQU     00h
      Output_Reg            EQU     01h
      Inversion_Reg         EQU     02h
      Configure_Reg         EQU     03h
      Manufact0_Str         db  'Company Copyright (C) 2005
Design by Duncan',0AH,0Dh,'$'
      GX3_Fun_Str           db  'Check PCM-9375 GPIO
function.',0Ah,0Dh,'$'
      Note1_Str             db  'Test Method: GPIO 0 connect to GPIO
1.',0Ah,0Dh,'$'
      Note2_Str             db  '             GPIO 2 connect to GPIO
3.',0Ah,0Dh,'$'
      Note3_Str             db  '             GPIO 4 connect to GPIO
5.',0Ah,0Dh,'$'
```

```
      Note4_Str             db  '             GPIO 6 connect to GPIO
7.',0Ah,0Dh,'$'
      Success_Str           db  'GPIO Test success !! ',0Ah,0Dh,'$'
      Fail_Str              db  'GPIO Test Fail !! ',0Ah,0Dh,'$'
      GX3_Fun_ISA_Str           db  'Check PCM-9375 ISA
function.',0Ah,0Dh,'$'
      ISA_Success_Str           db  'ISA Test success !! ',0Ah,0Dh,'$'
      ISA_Fail_Str              db  'ISA Test Fail !! ',0Ah,0Dh,'$'



;=====================================================================
;                 Main Program Start
;=====================================================================
.code
                  org     100h
 .STARTup
                ;Clear Screen
                 pusha



            ;1.Set GPIO 0,2,4,6 as output, GPI 1,3,5,7 as input
                  mov     ch,PCA9554_ID
                  mov     cl,Configure_Reg
                  mov     al,0AAh
                  call    Gx3SmbWriteByte


            ;2. Set GPIO 0,2,4,6 Output High
                  mov     ch,PCA9554_ID
                  mov     cl,Output_Reg
                  mov     al, 0h
                  call    Gx3SmbWriteByte

                  mov     ch,PCA9554_ID
                  mov     cl,Input_Reg
                  call     Gx3SmbReadByte
                  out     80h,al




                  popa
                  .exit
;=====================================================================
;SA02 - start
;[]============================================================[]
;Input  : CH - SMB address
;       CL - Index
;Output : AL - Value Read
;[]============================================================[]
            public      Gx3SmbReadByte
Gx3SmbReadByte    PROC NEAR
            push  dx
;1) START
            call  Gx3SmbStart
```

```
            jc    ExitGx3SmbReadByte
;2) Device Address (Write)
;3) Acknowledge
            mov   al, ch
            and   al, NOT 01h ; write
            call  WriteSDA
            jc    ExitGx3SmbReadByte
;4) Word Address(n)
;5) Acknowledge
            mov   dx, SMBUS_BASE_IO_ADDR + 03h
            in    al, dx
            or    al, (1 shl 4)
            out   dx, al

            mov   al, cl
            call  WriteSDA
            jc    ExitGx3SmbReadByte
                call    Delay5ms
;6) START
            call  Gx3SmbStart
            jc    ExitGx3SmbReadByte
;7) Device Address (Read)
;8) Acknowledge
            mov   al, ch
            or    al, 01h           ; read
            call  WriteSDA
            jc    ExitGx3SmbReadByte
;9) Data(n)
;10) No Acknowledge
            mov   dx, SMBUS_BASE_IO_ADDR + 00h
            in    al, dx
            newiodelay
;11) STOP
            call  Gx3SmbStop

ExitGx3SmbReadByte:
            pop   dx
            ret
Gx3SmbReadByte    ENDP
;[]============================================================[]
;Input  : CH - SMB address
;       CL - Index
;       AL - Value to Write
;Output     : none
;[]============================================================[]

            public      Gx3SmbWriteByte
Gx3SmbWriteByte   PROC NEAR
            pusha
            push  ax
;1) START
            call  Gx3SmbStart
            jc    ExitGx3SmbWriteByte
;2) Device Address (Write)
;3) Acknowledge
            mov   al, ch
            and   al, NOT 01h ; write
```

```
            call  WriteSDA
            jc    ExitGx3SmbWriteByte
;4) Word Address(n)
;5) Acknowledge
            mov   al, cl
            call  WriteSDA
            jc    ExitGx3SmbWriteByte
;6) Data
;7) Acknowledge
            pop   ax
            push  ax
            call  WriteSDA
;8) STOP
            call  Gx3SmbStop
ExitGx3SmbWriteByte:
            pop   ax
            popa
            ret
Gx3SmbWriteByte   ENDP

Gx3SmbReset PROC NEAR
            pusha
      ; 5535_A3 A2 SMB controller workaround.
      ; This will reset the registers which forces scl/sda high
SMBreset1:
            mov   dx, SMBUS_BASE_IO_ADDR + 05h
            xor   al, al
            out   dx, al

      ; Set SMBus clock frequency + Enable SMBus
      ; This must be done before any other smb register setup
            mov   dx, SMBUS_BASE_IO_ADDR + 05h
            in    al, dx
            or    al, (07Fh SHL 1)        ; about 1Mhz?
            out   dx, al
            or    al, (1 SHL 0)
            out   dx, al

            mov   dx, SMBUS_BASE_IO_ADDR + 02h
            in    al, dx
            test  al, (1 SHL 4)
            jnz   SMB_OK1
            jmp   SMBreset1
SMB_OK1:
            popa
            ret
Gx3SmbReset ENDP

Gx3SmbStart PROC NEAR
            pusha
            mov   cx, 1000h

CheckStop:
      ; Make sure the stop bit is not set because if it is
      ; the sendAddress phase will not work correctly
            mov   dx, SMBUS_BASE_IO_ADDR + 03h
            in    al, dx
```

```
            test  al, (1 shl 1)
            jz    IssueStart
            loop    CheckStop
;For the device to become the bus master, the software
;should perform the following steps:
; 1) Configure SMB I/O Offset 03h[2] to the desired operation mode
; (Polling = 0 or Interrupt = 1) and set SMB I/O Offset 03h[0]. This
causes
; the SMB Controller to issue a START condition on the bus when the bus
becomes
; free (SMB I/O Offset 02h[1] is cleared, or other conditions that can
delay
; START). It then stalls the bus by holding SCL low.
IssueStart:
            and   al, 11111011b     ; polling mode
            or    al, 01h           ; set 03h[0]
            out   dx, al
; 2) If a bus conflict is detected (i.e., another device pulls
; down the SCL signal), SMB I/O Offset 01h[5] is set.
; 3) If there is no bus conflict, SMB I/O Offset 01h[6] and
; SMB I/O Offset 01h[1] are set.
; 4) If SMB I/O Offset 03h[2] is set and either SMB I/O Offset
; 01h[5] or SMB I/O Offset 01h[6] is set, an interrupt
; is issued.
            mov   cx, 1000h

            mov   dx, SMBUS_BASE_IO_ADDR + 01h
CheckSDAST:
            in    al, dx
            test  al, (1 shl 5)     ; 01h[5] set? BER
            jnz   ErrorExit
            test  al, (1 shl 6)     ; wait for 01h[6]
            jnz   StartOk
            loop  CheckSDAST
            jmp   ErrorExit
StartOk:
            popa
            clc
            ret
ErrorExit:
            ;DAXX start
            mov    al,01h
            out    80h,al
            ;DAXX end
            call  Gx3SmbStop
            popa
            stc

            ret
Gx3SmbStart ENDP

WriteSDA    PROC NEAR
            pusha
            mov   dx, SMBUS_BASE_IO_ADDR + 00h
            out   dx, al
            mov   dx, SMBUS_BASE_IO_ADDR + 01h
            mov   cx, 1000h
```

```
WriteSDACheckSDAST:
            in    al, dx
            test  al, (1 shl 4)     ; BER?
            jnz   WriteSDAErrorExit
            test  al, (1 shl 6)     ; wait for 01h[6]
            jnz   WriteSDAExitWithoutError
            loop  WriteSDACheckSDAST
            jmp   WriteSDAErrorExit
WriteSDAExitWithoutError:
            popa
            clc
            ret
WriteSDAErrorExit:
            ;DAXX start
            mov    al,02h
            out    80h,al
            ;DAXX end
            popa
            call  Gx3SmbStop
            stc
            ret
WriteSDA    ENDP

Gx3SmbStop  PROC NEAR
            pusha
            mov   dx, SMBUS_BASE_IO_ADDR + 03h
            mov   al, (1 shl 1)
            out   dx, al            ; set STOP
            mov   dx, SMBUS_BASE_IO_ADDR + 01h
            in    al, dx
            out   dx, al
            call    Delay5ms
            popa
            ret
Gx3SmbStop  ENDP
;SA02 - end
;============================================================================
Delay5ms    proc  near
            push  cx
            mov   cx, 1000
      @@:
            NEWIODELAY
            loop  short @B
            pop   cx
            ret
Delay5ms        ENDP
Phoenix_debuger  proc near
                 pushf
                 push cx
                 push offset PhdebugRetAddr
                 push cs
                 push cs
                 db 0EAh
                 dw 0013h
                 dw 0DA00h
PhdebugRetAddr:
```

```
                popf
Phoenix_debuger  endp
;=============================================================================
;             Program  END
;=============================================================================

        END
```