9188end.lib

1. Constant defined in 9188e.h

```
#define    IN_BUF_SIZE       1024
#define    NoError             0
#define    InitPinIsOpen       0
#define    InitPinIsNotOpen    1
#define    QueueIsEmpty        0
#define    QueueIsNotEmpty     1
#define    PortError           -1
#define    DataError           -2
#define    ParityError         -3
#define    StopError           -4
#define    TimeOut             -5
#define    QueueEmpty          -6
#define    QueueOverflow       -7
#define    PosError            -8
#define    AddrError           -9
#define    BlockError          -10
#define    WriteError          -11
#define    SegmentError        -12
#define    BaudRateError       -13
#define    CheckSumError       -14
#define    ChannelError        -15
#define    TimeIsUp            1
```

2 Function Calls

Install COM port driver

Function Name	Description
InstallCom	Install driver for COM ports

Uninstall COM port driver

Function Name	Description
RestoreCom	Uninstall driver of COM port

Check if there is data in the input buffer of COM port

Function Name	Description
IsCom	Check for COM ports input data

Read one byte of data from input buffer of COM port

Function Name	Description
ReadCom	Read COM port data

Send one byte of data to COM port

Function Name	Description
ToCom	Send data to COM ports

Clear input buffer in COM ports

Function Name	Description
ClearCom	Clear input buffer COM ports

Check if the transmission is finished

Function Name	Description
WaitTransmitOver	Check for COM ports transmission

Check if output buffer is empty

Fnction Name	Description
IsCom1OutBufEmpt	Check for COM1 output buffer
IsCom2OutBufEmpty	Check for COM2 output buffer

Get the number of data in input buffer

Function Name	Description
DataSizeInCom	Get number of data of COM port

Set flow control Active of com1

Function Name	Description
SetFlowControlActive	Set flow control Active of com1

Set flow control Inactive of com1

Function Name	Description
SetFlowControlInactive	Set flow control Inactive of com1

Send command to 9000 module

Function Name	Description
SendCmdTo9000	Sent command to 9000 module to com port

Receive response from 9000 module

Function Name	Description
ReceiveResponseFrom9000	Receive response from 9000 module from com port

Switch on/off Red LED

Function Name	Description
LedOff	Red Led light Off
LedOn	Red Led light On

For 5-digit LED

Function Name	Description
Init5DigitLed	Initialize the hardware of 5-digit LED & blank all digits
Show5DigitLed	Show digital number (0-F),blank(space),’-‘&’.’(dot)
Show5DigitLedSeg	Show any segment
Show5DigitLedWithDot	Show digital number and dot’.’
Set5DigitLedTestMode	Set to test mode,all segment will be turn on
Set5DigitLedIntensity	Set the intensity of 5-Digit LED
Disable5DigitLed	Disable 5-Digit LED,all segment will br OFF
Enable5DigitLed	Enable 5-Digit LED,all segment will br ON

Time delay function

Fnction Name	Description
DelayTimeMs	Delay time unit,unit is 1ms(use software)
DelayMs	Delay time unit,unit is 1ms(use hardware)
Delay_1	Delay time unit,unit is 0.1ms(use hardware)

For NVRAM

Function Name	Description
ReadNVRAM	Read one byte of data from NVRAM
WriteNVRAM	Write one byte of data to NVRAM

For EEPROM

Fnction Name	Description
WriteEEP	Write one byte of data to EEPROM
ReadEEP	Read one byte of data from EEPROM
EnableEEP	Enable EEPROM for writing
ProtectEEP	Set to write-protect

Watch dog timer function

Function Name	Description

EnableWDT	Enable watch dog timer
RefreshWDT	Refresh watch dog timer
DisableWDT	Disable watch dog timer
IsResetByWatchDogTimer	check 9188e is reset by watchdog timer
IsResetByPowerOff	check 9188e is reset by power off

For flash memory

Function Name	Description
FlashReadId	Read the flash memory type
FlashWrite	Write one byte of data to flash
FlashErase	Erase one block of data
FlashRead	Read one byte of data

Standard IO

Function Name	Description
getch4	Instead of getch
kbhit4	Instead pf kbhit
ungetch4	Instead of ungetch
putch4	Instead of putchar

MISC functions

Function Name	Description
GetLibVersion	Get the version of library
ReadInitPin	Get the status of INIT pin
_MK_FP	Make a far pointer

For timers

Function Name	Description
TimerOpen	Install timer driver
TimerClose	Uninstall timer driver
TimerResetValue	Reset the value of timer ticks to 0
TimerReadValue	Read the value of timer ticks
StopWatchReset	Reset stopwatch timer to 0
StopWatchStart	Start stopwatch
StopWatchStop	Stop stopwatch
StopWatchPause	Pause stopwatch
StopWatchContinue	Continue stopwatch
StopWatchReadValue	Read the value of stopwatch timer
CountDownTimerStart	Start countdown timer
CountDownTimerReadValue	Read the value of countdown timer
InstallUserTimer	Install the user’s timer function
InstallUserTimer1C	Install the user’s timer function on INT 0x1C

3 Declaration and Input Parameter description

3.1 InstallCom

- Declaration:

  InstallCom(int port, unsigned long baud, int data, int parity, int stop);

- Input parameter

Argument	Description
Port	1/2/3…/8 for com1/2/3…./8
Baud	Baudrate, 1200,2400…9600..57600,115200
Data	Data bits, 6 or 7 or 8 (com1/com2 only 7 or 8)
Parity	Parity bit, 0=none, 1=evn, 2=odd
Stop	Stop bit, 1 or 2 (com1/com2 only for stop bit=1)

3.2 RestallCom

- Declaration:

  RestoreCom(int port)；

- Input parameter

Argument	Description
Port	1/2/3…/8 for com1/2/3…./8

3.3 IsCom

- Declaration:

  IsCom(int port);

- Input parameter

Argument	Description
Port	1/2/3…/8 for com1/2/3…./8

- Return value:

  0: Queue is empty

  1: Queue is not empty

3.4 ReadCom

- Declaration:

  ReadCom(int port);

- Input parameter

Argument	Description
Port	1/2/3…/8 for com1/2/3…./8

- Return value:

  Character read in。

3.5 ToCom

- Declaration:

  ToCom(int port, int data);

- Input parameter

Argument	Description
Port	1/2/3…/8 for com1/2/3…./8
Data	8-bit character to transmit

3.6 ClearCom

- Declaration:

  ClearCom(int port);

- Input parameter

Argument	Description
Port	1/2/3…/8 for com1/2/3…./8

3.7 WaitTransmitOver

- Declaration:

  WaitTransmitOver(int port);

- Input parameter

Argument	Description
Port	1/2/3…/8 for com1/2/3…./8

- Return value:

  NoError：com data transmit over

  Others：Negative error code

3.8 IsCom1OutBufEmpty

- Declaration:

  IsCom1OutBufEmpty(void);

- Input parameter

  None

- Return value:

  When com1 output buffer is empty return 1，else return 0

3.9 IsCom2OutBufEmpty

- Declaration:

  IsCom2OutBufEmpty(void);

- Input parameter

  None

- Return value:

When com2 output buffer is empty return 1， else return 0

3.10 DataSizeInCom

- Declaration:

  DataSizeInCom(int port);

- Input parameter

Argument	Description
Port	1/2/3…/8 for com1/2/3…./8

- Return value:

  Data number in comport input buffer

3.11 SetFlowControlActive

- Declaration:

  SetFlowControlActive(void);

  (Set com1 hardware flow control active)

- Input parameter

  None,

- Return value:

  None

3.12 SetFlowControlInactive

- Declaration:

  SetFlowControlInactive(void);

  (Set com1 hardware flow control Inactive)

- Input parameter

  None,

- Return value:

  None

3.1 3 SendCmdTo9000

- Declaration:

  SendCmdTo9000(int iPort, unsigned char *cCmd, int iChksum);

- Input parameter

Argument	Description
Port	1/2/3…/8 for com1/2/3…./8
cCmd	The address of the command want to be sent。 The command need not end with the command terminator 0x0d, this function will auto send 0x0d out after send the command and check

	sum (if check sum is enable) Ref:9000 module command set
iChksum	0：check sum disable, 1：check sum enable。 If check is set to enable, the function will add 2 bytes check sum after the command

- Return value:

  When success return NoError

  Others：return error code

3.14 ReceiveResponseFrom9000

- Declaration:

  ReceiveResponseFrom9000(int iPort, unsigned char *cCmd, long lTimeout, int iChksum);

- Input parameter

Argument	Description
Port	1/2/3…/8 for com1/2/3…./8
cCmd	The address of buffer to save the received message。

lTimeout	This function will check the comport for receive message, if it check lTimeout times, and do not get a response message (end with 0x0d), it will return timeout error。
iChksum	0：check sum disable, 1：check sum enable。

- Return value:

  When success return NoError

  Others：return error code

3.15 ReadInitPin

- Declaration:

  ReadInitPin(void);

- Input parameter

  None,

- Return value:

  1：Init pin touch to ground

  0：Init pin open

3.16 LedOff

- Declaration:

  LedOff(void);

  (light red led off)

- Input parameter

  None

- Return value:

  None

3.17 LedOn

- Declaration:

  LedOn(void);

  (light red led on)

- Input parameter

  None

- Return value:

  None

3.18 Init5DigitLed

- Declaration:

  Init5DigitLed(void);

- Input parameter

  None

- Return value:

  None

3.19 Show5DigitLed

- Declaration:

  Show5DigitLed(int pos, int value);

- Input parameter

Argument	Description
pos	1~5 FOR POSITION 1~5 (FROM LEFT TO RIGHT)
VALUE	0~9 FOR SHOW ‘0’~’9’, 10 SHOW ‘A’ 11 SHOW ‘b’ 12 SHOW ‘c’ 13 SHOW ‘d’ 14 SHOW ‘e’ 15 SHOW ‘F’ 16 SHOW ‘ ’ blank or space

	17 SHOW ‘- ’ 18 SHOW ‘.’(dot)

- Return value:

When success return NoError

Others：return error code

3.20 Show5DigitLedWithDot

- Declaration:

Show5DigitLedWithDot (int pos, int value);

- Input parameter

Argument	Description
pos	1~5 FOR POSITION 1~5 (FROM LEFT TO RIGHT)
VALUE	0~9 FOR SHOW ‘0’~’9’, 10 SHOW ‘A’ 11 SHOW ‘b’ 12 SHOW ‘c’ 13 SHOW ‘d’ 14 SHOW ‘e’ 15 SHOW ‘F’ 16 SHOW ‘ ’ blank or space 17 SHOW ‘- ’ 18 SHOW ‘.’(dot) 0~17 will also show the dot(‘.’)

- Return value:

When success return NoError

Others：return error code

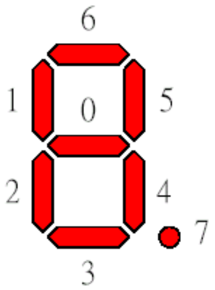
3.21 Show5DigitLedSeg

- Declaration:

Show5DigitLedSeg(int pos,int value);

- Input parameter

Argument	Description
pos	1~5 FOR POSITION 1~5 (FROM LEFT TO RIGHT)
value	Each bit of value control one segment of 7-segment led and the dot. The bit is 1 will let the segment light，The bit is 0 will let the segment light off。(EX：in following picture，if we want to show “2”， it must light segment 6、5、0、2 、3 and 7, the value must be 0xed



3.22 Set5DigitLedTestMode

- Declaration:

  Set5DigitLedTestMode(int mode);

- Input parameter

Argument	Description
mode	0：normal mode。 1：set to test mode，all the segment will light。

3.23 Set5DigitLedIntensity

- Declaration:

  Set5DigitLedIntensity(int mode);

- Input parameter

Argument	Description
mode	0~15，mode=0 is the most dark；mode=1 is the most bright。 When call Init5DigitLed will set mode=7

- Return value:

  When success return NoError

  Others：return error code

3.24 Disable5DigitLed

- Declaration:

  Disable5DigitLed(void);

  (Call Disable5DigitLed will blank all 5-digit LED)

- Input parameter

  None

- Return value:

  None

3.25 Enable5DigitLed

- Declaration:

  Enable5DigitLed(void);

  (Call Enable5DigitLed after call Disable5DigitLed will set the 5-digit LED back to normal mode)

- Input parameter

  None

- Return value:

  None

3.26 ReadNVRAM

- Declaration:

  ReadNVRAM(int addr);

- Input parameter

Argument	Description
addr	0~30 (total size of NVRAM is 31 bytes)

- Return value:

  Positive：8-bit NVRAM data

  Others: negative err code

3.27 WriteNVRAM

- Declaration:

  WriteNVRAM(int addr, int data);

- Input parameter

Argument	Description
addr	0~30
data	0~255(8-bit data)

3.28 WriteEEP

- Declaration:

  WriteEEP(int block, int addr, int data);

- Input parameter

Argument	Description
block	0 to 6 (total 7 blocks)
addr	0 to 255 (every block has 256 bytes)
Data	0 to 255(8-bit data)

3.29 ReadEEP

- Declaration:

  ReadEEP(int block, int addr);

- Input parameter

Argument	Description
block	0 to 6 (total 7 blocks)
addr	0 to 255 (every block has 256 bytes)

- Return value:

  Positive： 8-bit data from EEPROM

  Others: negative err code

3.30 EnableEEP

- Declaration:

  EnableEEP(void);

  (The EEPROM is in the write-protect mode when first power on, it should enable the EEPROM before any write operation)

- Input parameter

  None

- Return value:

  None

3.31 ProtectEEP

- Declaration:

  ProtecEEP(void);

  (The EEPROM is in the write-protect mode when call ProtecEEP(void), it will inhabit the write operation。)

- Input parameter

  None

- Return value:

  None

3.32 EnableWDT

- Declaration:

  EnableWDT(void);

  (The watchdog timer is fixed in 1.6 sec, if the watchdog is enable, the application program should refresh the watchdog timer before the 1.6 sec timer up。If the watchdog timer is up, the 9188E will be reset by the hardware circuit。)

- Input parameter

  None

- Return value:

  None

3.33 RefreshWDT

- Declaration:

  RefreshWDT(void);

- Input parameter

  None

- Return value:

  None

3.34 DisableWDT

- Declaration:

  DisableWDT(void);

  (This function can disable the watchdog timer。)

- Input parameter

  None

- Return value:

  None

3.35 IsResetByWatchDogTimer

- Declaration:

  IsResetByWatchDogTimer(void);

- Input parameter

  None

- Return value:

  0：It is not reset by watchdog timer since last time call IsResetByWatchDogTimer。

  1：It is reset by watchdog timer since last time call IsResetByWatchDogTimer。If call IsResetByWatchDogTimer again，it will return 0。

3.36 IsResetByPowerOff

- Declaration:

  IsResetByPowerOff(void);

  To check 9188e is reset by power on or other reasons (reset by watchdog timers or just the program execute a far jump to 0xffff:0000 or call int 19h)

- Input parameter

  None

- Return value:

  0：It is not reset by power off since last time call IsResetByPowerOff。

1：It is reset by power off since last time call IsResetByPowerOff。If call IsResetByPowerOff again，it will return 0。

3.37 FlashReadId(void);

- Declaration:

  FlashReadId(void);

- Input paramete

  None

- Return value:

  High byte return the flash types,and low byte returns the manufacture number。

3.38 FlashWrite

- Declaration:

  FlashWrite(unsigned int seg, unsigned int offset, char data);

- Input parameter

Argument	Description
seg	Seg can be 0xc000 or 0xd000 to 0xe000 for 256k type， and can be 0x8000 to 0xe000 for 512k types
offset	Range is 0 to 0xffff
data	Range is 0 to 0xff

- Return value:

When success return NoError

Others： return error code

3.39 FlashErase

- Declaration:

FlashErase(unsigned int seg);

- Input parameter

Argument	Description
seg	Seg can be 0xc000 or 0xd000 to 0xe000 for 256k type， and can be 0x8000 to 0xe000 for 512k types

- Return value:

When success return NoError

Others： return error code

3.40 FlashRead

- Declaration:

FlashRead(unsigned int seg, unsigned int offset);

- Input parameter

Argument	Description
seg	Segment of flash memory
offset	Offset of flash memory

- Return value:

Return the bytes at the memory location address by seg offset。

3.41 getch4

- Declaration:

getch4(void);

- Input parameter

None

- Return value:

The next key input value。

3.42 kbhit4

- Declaration:

  kbhit4 (void);

- Input parameter

  None

- Return value:

  Return 0 when keyboard buffer is empty, others when keyboard is not empty。

3.43 ungetch4

- Declaration:

  ungetch4(int key);

- Input parameter

Argument	Description
key	The key value want to putback to keyboard buffer。

- Return value:

  When success return NoError

  Others：keyboard buffer is full

3.44 putch4

- Declaration:

  putch4(int data);

- Input parameter

Argument	Description
data	The value want to send out from com1(only the low byte send out)

- Return value:

  None

3.45 GetLibVersion

- Declaration:

  GetLibVersion(void);

- Input parameter

  None

- Return value:

  Return the library version

3.46 _MK_FP

- Declaration:

  *_MK_FP(unsigned segment,unsigned offset);

- Input parameter

Argument	Description
segment	The segment of the far pointer
offset	The offset of the far pointer

- Return value:

  The far pointer segment offset。

3.47 DelayTimeMs

- Declaration:

  DelayTimeMs(unsigned int time);

- Input parameter

Argument	Description
time	The time interval want to delay, unit is 1 ms

- Return value:

  None

3.48 DelayMs

- Declaration:

  DelayMs(unsigned t);

- Input parameter

Argument	Description
t	The time interval want to delay, unit is 1 ms

- Return value:

  None

3.49 Delay_1

- Declaration:

  Delay_1(unsigned ms);

- Input parameter

Argument	Description
ms	The time interval want to delay, unit is 0.1 ms

- Return value:

  None

3.50 TimerOpen

- Declaration:

  TimerOpen(void);

- Input parameter

  None

- Return value:

  When install success return NoError

  Return 1 if timer function is already installed。

3.51 TimerClose

- Declaration:

  TimerClose (void);

- Input parameter

  None

- Return value:

  Always return NoError

3.52 TimerResetValue(void);

- Declaration:

  TimerResetValue (void);

  (reset the timer tick value of timer to 0)

- Input parameter

  None

- **Return value:**

  None

3.53 TimerReadValue(void);

- Declaration:

  TimerReadValue (void);

- Input parameter

  None

- **Return value:**

  Return the current value of timer ticks。

3.54 StopWatchReset

- Declaration:

  StopWatchReset(int channel);

(Reset the stop watch value)

- Input parameter

Argument	Description
channel	The range is 0~7。

- Return value:

When success return NoError

Others： return error code

3.55 StopWatchStart

- Declaration:

StopWatchStart(int channel);

(Set the StopWatch start to count time from 0. after this setting,the StopWatch timer will increase 1 every 1 ms)

- Input parameter

Argument	Description
channel	The range is 0~7。

- Return value:

When success return NoError

Others： return error code

3.56 StopWatchStop

- Declaration:

  StopWatchStart(int channel);

  (This function will stop use the StopWatch timer)

- Input parameter

Argument	Description
channel	The range is 0~7。

- Return value:

  When success return NoError

  Others： return error code

3.57 StopWatchPause

- Declaration:

  StopWatchPause(int channel);

  (This function will pause the StopWatch timer)

- Input parameter

Argument	Description
channel	The range is 0~7。

- Return value:

When success return NoError

Others： return error code

3.58 StopWatchContinue

- Declaration:

StopWatchContinue (int channel);

(This function will continue the StopWatch timer)

- Input parameter

Argument	Description
channel	The range is 0~7。

- Return value:

When success return NoError

Others： return error code

3.59 StopWatchReadValue

- Declaration:

StopWatchReadValue (int channel,unsigned long *value);

(This function will read the current value of StopWatch, the unit is 1 ms)

- Input parameter

Argument	Description
channel	The range is 0~7。
value	A pointer to a unsigned long variable, to store the timer value

- Return value:

When success return NoError

Others： return error code

3.60 CountDownTimerStart

- Declaration:

CountDownTimerStart(int channel,unsigned long count);

(Call this function must set a start value for the CountDown timer to count down until it reach 0, that is “time is up”)

- Input parameter

Argument	Description
channel	The range is 0~7。
count	The start value of CountDown timer

- Return value:

When success return NoError

Others： return error code

3.61 CountDownTimerReadValue

- Declaration:

  CountDownTimerReadValue(int channel,unsigned long *value);

- Input parameter

Argument	Description
channel	The range is 0~7。
value	A pointer to a unsigned long variable, to store the timer value

- Return value:

  When success return NoError

  Others： return error code

3.62 InstallUserTimer

- Declaration:

  InstallUserTimer(void (*fun)(void));

  (The timer will generate interrupt every 1 ms, user can install a timer function, then the timer interrupt will call the user function, but the user function must finished before 1 ms)

- Input parameter

Argument	Description
fun	The function pointer of user Timer function

- Return value:

None

3.63 InstallUserTimer1C

- Declaration:

InstallUserTimer1C(void (*fun)(void));

(The system timer will generate interrupt 8 every 1/18.2 second, and interrupt 8 service routine will call int 0x1c ， user can install a timer1C function, when int 0x1c is called， it will call user timer1C function)

- Input parameter

Argument	Description

fun	The function pointer of user Timer1C function

- Return value:

  None