

---

# Pic18F Abstraction Layer (MCAL) Documentation

---

*Revision: A*

---

*Date: 03.01.2020*

---

<b>Title</b>	GScrubber Class Documentation	<b>Revision</b>	A
		<b>Date</b>	03.01.2020

## Table of Contents

<b>TABLE OF CONTENTS.....</b>	<b>1</b>
<b>1 – MCAL_PIC18F FILE ELEMENTS :TYPES &amp; GLOBAL VARIABLES.....</b>	<b>2</b>
I. PUBLIC (GLOBAL) TYPES : .....	2
II. PUBLIC(GLOBAL) VARIABLES:.....	2
<b>2 – DETAILED DESCRIPTION FUNCTIONS.....</b>	<b>3</b>
<b>3 – MEMBER DOCUMENTATION.....</b>	<b>5</b>
-VOLATILE GPIOXX* MCAL_PIC_GETGPIO(U8_T PORTXX).....	5
-VOID MCAL_PIC_SetBitDir( U8_T PORTXX , U8_T BITX , U8_T DIR).....	5
-VOID MCAL_PIC_SetBitsDir( U8_T PORTXX , U8_T BITX , U8_T BIT_COUNUTER , U8_T DIR).....	5
-VOID MCAL_PIC_SetPortDir(U8_T PORTXX,U8_T DIR);.....	5
-VOID MCAL_PIC_SetBitVal( U8_T PORTXX , U8_T BITX , U8_T VAL);.....	5
-VOID MCAL_PIC_SetBitsVal( U8_T PORTXX , U8_T BITX , U8_T VAL,U8_T BIT_COUNUTER);.....	6
-VOID MCAL_PIC_SetPortVal( U8_T PORTXX , U8_T VAL );.....	6
-U8_T MCAL_PIC_GetBitVal ( U8_T PORTXX , U8_T BITX);.....	6
-U8_T MCAL_PIC_GetPinVal( U8_T PORTXX );.....	7
-VOID MCAL_PIC_Init_Port(U8_T PORTXX);.....	7
-U8_T MCAL_PIC_GetIntSource(VOID);.....	7
-VOID MCAL_PIC_INTPEN (U8_T ID, U8_T SOURCE ,ENUM _MODE MODE);.....	7
-VOID MCAL_PIC_ENINTPx(U8_T SOURCE , ENUM _MODE MODE);.....	7
-VOID MCAL_PIC_ENGINTP(VOID);.....	8
-VOID MCAL_PIC_DisGINT(VOID);.....	8
-VOID MCAL_PIC_INTPSERVE(U8_T SOURCE);.....	8
-VOID MCAL_PIC_DIRECTHIGHISR(VOID);.....	9
-VOID MCAL_PIC_DisINT(U8_T INTID);.....	9
-VOID MCAL_PIC_XXXXREDIRECT(VOID (*PTR) (VOID));.....	9
-VOID MCAL_PIC_InitTimer0(U8_T MODE,U16_T PRESCALER);.....	9
-VOID MCAL_PIC_TIMER0SetPS(U16_T PRESCALER);.....	10
-VOID MCAL_PIC_TIMER0_GetTick(VOID);.....	10
-VOID MCAL_PIC_TIMER0SysClkHandler(VOID);.....	10
-Time* MCAL_PIC_GetTime(VOID);.....	10
-VOID MCAL_PIC_ResetTime(VOID);.....	11
-VOID MCAL_PIC_SysClkDef(VOID);.....	11
<b>REVISION HISTORY.....</b>	<b>11</b>

<b>Title</b>	GScrubber Class Documentation	<b>Revision</b>	A
		<b>Date</b>	03.01.2020

# 1 – MCAL\_PIC18F File Elements :Types & Global Variables

---

## I. Public (Global) Types:

Type	Name	Description	Accessed by
enum _Mode	_Mode/Mode	{OnChange,Ris_edge,Fal_edge}	Used by InterruptManger.C to initialize interrupt in a certain mode
Struct GPIOx	GPIOXX	Elements :volatile u8_t* (PORTX,TRISX,LATX)	Used by DIO to resolve real port address
Struct time	Time	Elements: float ticks u32_t minutes; u16_t milliseconds; u8_t seconds;	Ticks hold the needed number of interrupt to trigger 1 millisecond increase, struct is used to introduce systime simplification .
StdTypes		Refer to compiler documentation	
extern void (* MCAL_PIC_XXHandler) (void)	MCAL_PIC_XXHandler	A group of pointers to functions that takes void and returns void to act as interrupt handlers , each pointer has a redirection function mentioned blow	InterruptManager.C and any file that needs to set an interrupt handler.

## II. Public(global) Variables:

Type	Name	InFile	Usage
Struct time	SysTime	MCAL_PIC18F.h	Keep track of time since sysclk is enabled
u8_t	<b>global_prescaler</b>	MCAL_PIC18F-Timer0Module.c	Keep track of the most recent prescaler set
u8_t	<b>current_source</b>	MCAL_PIC18F-InterruptManager.c	Keep track of the most recent interrupt source
void (*) (void)	MCAL_PIC_XXHandler	MCAL_PIC18F-InterruptManager.c	<b>Explained above</b>

<b>Title</b>	GScrubber Class Documentation	<b>Revision</b>	A
		<b>Date</b>	03.01.2020

## 2 – Detailed Description Functions

Detail Description in member documentation .

#	Return Type	Input Type	Name	Calls	Called by
1	volatile GPIOXX*	u8_t PORTxx	MCAL_PIC_getgpio	N/A	MCAL_PIC18F_DIO.c functions
2	void	u8_t PORTxx , u8_t bitx , u8_t dir	MCAL_PIC_SetBitDir	Predefined function,MCAL_PIC_getgpio(PORTxx )	HAL_picgenios.h members
3	void	u8_t PORTxx , u8_t bitx , u8_t bit_counuter , u8_t dir	MCAL_PIC_SetBitsDir	Predefined function,MCAL_PIC_getgpio(PORTxx )	HAL_picgenios.h members
4	void	u8_t PORTxx ,u8_t dir	MCAL_PIC_SetPORTDir	Predefined function,MCAL_PIC_getgpio(PORTxx )	HAL_picgenios.h members
5	void	u8_t PORTxx , u8_t bitx , u8_t val	MCAL_PIC_SetBitVal	Predefined function,MCAL_PIC_getgpio(PORTxx )	HAL_picgenios.h members
6	void	u8_t PORTxx , u8_t bitx , u8_t val,u8_t bit_counuter	MCAL_PIC_SetBitSVal	Predefined function,MCAL_PIC_getgpio(PORTxx )	HAL_picgenios.h members
7	void	u8_t PORTxx , u8_t val	MCAL_PIC_SetPORTVal	Predefined function,MCAL_PIC_getgpio(PORTxx )	HAL_picgenios.h members
8	void	u8_t PORTxx	MCAL_PIC_Init_Port	Predefined function,MCAL_PIC_getgpio(PORTxx )	MCAL_PIC18F_DIO.c functions
9	u8_t	u8_t PORTxx , u8_t bitx	MCAL_PIC_GetBitVal	Predefined function,MCAL_PIC_getgpio(PORTxx )	HAL_picgenios.h members
10	u8_t	u8_t PORTxx	MCAL_PIC_GetPINVal	Predefined function,MCAL_PIC_getgpio(PORTxx )	HAL_picgenios.h members
11	void	u8_t id, u8_t source ,enum_Mode Mode	MCAL_PIC_InitNTPEn	MCAL_PIC_EnableNTP,MCAL_PIC_EnableNTPx	HAL_picgenios.h members
12	void	u8_t source , enum_Mode mode	MCAL_PIC_EnableNTPx	MCAL_PIC_EnableNTP	HAL_picgenios.h members
13	void	void	MCAL_PIC_EnableGINTP		HAL_picgenios.h members & and others

<b>Title</b>	GScrubber Class Documentation	<b>Revision</b>	A
		<b>Date</b>	03.01.2020

14	void	<b>void</b>	MCAL_PIC_D isGINT		HAL_picgenios.h members & and others
15	void	<b>u8_t source</b>	MCAL_PIC_In tpServe	Interrupt_handler()	HAL_picgenios.h members & and others
16	void	<b>void</b>	MCAL_PIC_D irectHighISR	Interrupt_handler()	HAL_picgenios.h members & and others
17	void	<b>void (*ptr) (void</b>	MCAL_PIC_I NTP0redirect		HAL_picgenios.h members & and others
18	void	<b>void (*ptr) (void</b>	MCAL_PIC_I NTP1redirect		HAL_picgenios.h members & and others
19	void	<b>void (*ptr) (void</b>	MCAL_PIC_I NTP2redirect		HAL_picgenios.h members & and others
20	void	<b>void (*ptr) (void</b>	MCAL_PIC_I NTPRBredirect		HAL_picgenios.h members & and others
21	void	<b>void (*ptr) (void</b>	MCAL_PIC_T imer0OVredire ct		HAL_picgenios.h members & and others
22	void	<b>u8_t INTID</b>	MCAL_PIC_D isINT	MCAL_PIC_DisGIN T	HAL_picgenios.h members & and others
23	u8_t	<b>void</b>	MCAL_PIC_G etINTSource		HAL_picgenios.h members & and others
24	void	<b>u8_t mode,u16_t prescaler</b>	MCAL_PIC_In itTimer0	MCAL_PIC_Timer0 SetPS,MCAL_PIC_T imer0_GetTick	HAL_picgenios.h members & and others
25	void	<b>u16_t prescaler</b>	MCAL_PIC_T imer0SetPS	MCAL_PIC_Timer0 _GetTick	HAL_picgenios.h members & and others
26	void	<b>void</b>	MCAL_PIC_T imer0_GetTick	MCAL_PIC_Timer0 SetPS	HAL_picgenios.h members & and others
27	void	<b>void</b>	MCAL_PIC_T imer0SysClkH andler		HAL_picgenios.h members & and others
28	void	<b>u8_t OVT0</b>	MCAL_PIC_T imer0_SetCM		HAL_picgenios.h members & and others
29	Struct Time*	<b>void</b>	MCAL_PIC_G etTime		HAL_picgenios.h members & and others
30	void	<b>void</b>	MCAL_PIC_R esetTime		HAL_picgenios.h members & and others
31	void	<b>void (*ptr) (void)</b>	MCAL_PIC_S ysClkredirect	MCAL_PIC_Timer0 SysClkHandler	HAL_picgenios.h members & and others
32	void	<b>void</b>	MCAL_PIC_S ysClkDef		No one

Find detailed description blow...

Title	GScrubber Class Documentation	Revision	A
		Date	03.01.2020

## 3 – Member Documentation

---

**-volatile GPIOXX\* MCAL\_PIC\_getgpio(u8\_t PORTxx).**

Description: resolves dummy port names to an actual address for further control

Input Value: unsigned 8bit char for the predefined dummy port names

Return Value: a volatile pointer to structure of type GPIOation

**-void MCAL\_PIC\_SetBitDir( u8\_t PORTxx , u8\_t bitx , u8\_t dir).**

Description: sets a PINx (bitx) in PORTx as INPUT/OUTPUT (dir)

Input Value: unsigned 8bit char for the predefined dummy port names ,

unsigned 8bit char for bitx location

unsigned 8bit char for direction (0 or 1)

Return Value: void simple debugging method for the scrubber functionality

**-void MCAL\_PIC\_SetBitsDir( u8\_t PORTxx , u8\_t bitx , u8\_t bit\_counuter , u8\_t dir).**

Description: sets a PINx (bitx)s in PORTx as INPUT/OUTPUT (dir)

Input Value: unsigned 8bit char for the predefined dummy port names

unsigned 8bit char for bitx start location ,

unsigned 8bit char for number of bits starting bitx location ,

unsigned 8bit char for direction (0 or 1)

Return Value: void

**-void MCAL\_PIC\_SetPORTDir(u8\_t PORTxx ,u8\_t dir);**

Description: sets PORTx as INPUT/OUTPUT (dir)

Input Value: unsigned 8bit char for the predefined dummy port names ,

unsigned 8bit char for direction (0 or 1)

Return Value: void

**-void MCAL\_PIC\_SetBitVal( u8\_t PORTxx , u8\_t bitx , u8\_t val);**

Description: sets a PINx (bitx) in PORTx value.

Input Value: unsigned 8bit char for the predefined dummy port names ,

<b>Title</b>	GScrubber Class Documentation	<b>Revision</b>	A
		<b>Date</b>	03.01.2020

unsigned 8bit char for bitx location ,

unsigned 8bit char for bit value

Return Value:void

**-void MCAL\_PIC\_SetBitSVal( u8\_t PORTxx , u8\_t bitx , u8\_t val,u8\_t bit\_counuter);**

Description: sets PINx (bitx)s in PORTx values.

Input Value: unsigned 8bit char for the predefined dummy port names ,

unsigned 8bit char for bitx location ,

unsigned 8bit char for bitx number of bits to be changed ,

unsigned 8bit char for bits common value

Return Value:void

**-void MCAL\_PIC\_SetPORTVal( u8\_t PORTxx , u8\_t val );**

Description: sets PORTx values

Input Value: unsigned 8bit char for the predefined dummy port names ,

unsigned 8bit char for bits common value

Return Value:void

**-u8\_t MCAL\_PIC\_GetBitVal ( u8\_t PORTxx , u8\_t bitx);**

Description: reads the value of PINx (bitx) in PORTx .

Input Value: unsigned 8bit char for the predefined dummy port names ,

unsigned 8bit char for bitx location .

Return Value:unsigned 8bit char holding the value of PORTx input register PINx bit

<b>Title</b>	GScrubber Class Documentation	<b>Revision</b>	A
		<b>Date</b>	03.01.2020

**-u8\_t MCAL\_PIC\_GetPINVal( u8\_t PORTxx );**

Description: reads the value of PORTx .

Input Value: unsigned 8bit char for the predefined dummy port names .

Return Value: unsigned 8bit char holding the value of PORTx input register

**-void MCAL\_PIC\_Init\_Port(u8\_t PORTxx);**

Description: sets a PINx (bitx) in PORTx as INPUT/OUTPUT (dir)

**-u8\_t MCAL\_PIC\_GetINTSource(void);**

Description: return interrupt source in case get & serve are used

Input Value: void

Return Value: a predefined interrupt source signal mentioned in the header file

**-void MCAL\_PIC\_INTPEn (u8\_t id, u8\_t source ,enum \_Mode Mode);**

Description: enables an interrupt source to send an interrupt signal and sets it's configuration mode

Input Value: Interrupt id , interrupt source if id have more then one interrupt source ,

interrupt mode , specific to interrupt source and system basis

Return Value: void

Note: function not complete due it's not needed for the current application yet it's importing for porting

**-void MCAL\_PIC\_EnINTPx(u8\_t source , enum \_Mode mode);**

Description: enables external interrupt sources on PORTB it also unmaskes the interrupt & configures it

Input Value: interrupt sources as unsigned 8bit char:

INTx0 : high priority external interrupt on PORTB RB0



<b>Title</b>	GScrubber Class Documentation	<b>Revision</b>	A
		<b>Date</b>	03.01.2020

INTx1 : low priority external interrupt on PORTB RB1

INTx2 : low priority external interrupt on PORTB RB2

INTx5..7 : low priority external interrupt on change PORTB RB6 RB6 RB5

interrupt modes as unsigned 8bit char :

OnChange: for RB7 through RB5 only

Rise\_edge: for INTx0..2 interrupt is triggered on rising edge of the signal

fall\_edge:for INTx0..2 interrupt is triggered on fall edge of the signal

Return Value: void

Note:calls MCAL\_PIC\_EnGINTP() :

**-void MCAL\_PIC\_EnGINTP(void);**

Description:rewraps general interrupt enable

Input Value: void

Return Value:void

**-void MCAL\_PIC\_DisGINT(void);**

Description:rewraps general interrupt disable

Input Value: void

Return Value:void

**-void MCAL\_PIC\_IntpServe(u8\_t source);**

Description:serve an interrupt for it's source by calling their default handler

Input Value: interrupt source as unsigned 8bit char

Return Value:void

<b>Title</b>	GScrubber Class Documentation	<b>Revision</b>	A
		<b>Date</b>	03.01.2020

Note: handlers are pointers to function's of type void and input void

Note2: calling this function can serve an interrupt handler without knowing if interrupt signal actually happened in hardware

### **-void MCAL\_PIC\_DirectHighISR(void);**

Description:Serves the reason of interrupt by searching for it and assigning the default

high priority interrupt handler to it's handler then clears the interrupt flag.

Input Value: void

Return Value:void

### **-void MCAL\_PIC\_DisINT(u8\_t INTID);**

Description:not needed for current application yet it should disable an Interrupt by it's id

Input Value: void

Return Value: void

### **-void MCAL\_PIC\_xxxxredirect(void (\*ptr) (void));**

Description: these functions are used for interfacing with files that need to use an external interrupt

the function links the interrupt handler to the inputed function location

interrupt handler are always called interrupt functions should not be called directly

Input Value: a pointer to a function that takes void and returns void

Return Value:void

### **-void MCAL\_PIC\_InitTimer0(u8\_t mode,u16\_t prescaler);**

Description:initialize Timer0 in predefined mode that has different levels of configuration

Input Value: 1-unsigned 8bit char for possible modes :

<b>Title</b>	GScrubber Class Documentation	<b>Revision</b>	A
		<b>Date</b>	03.01.2020

bit8\_OV: configures timer0 to work as 8bit timer that generates an interrupt on Overflow (256)

sys\_clk: configures timer0 to work an bit8\_OV and configures that to provide system time functionality through time structure global SysTime

Return Value:void

**-void MCAL\_PIC\_Timer0SetPS(u16\_t prescaler);**

Description:sets the prescale value of Timer0

Input Value: void

Return Value:void

**-void MCAL\_PIC\_Timer0\_GetTick(void);**

Description: optimizes timer0 prescaler to generate an interrupt every 1 ms based on frequency settings & and prescaler old values uses MATH.c for rounding

Input Value: void

Return Value:void

**-void MCAL\_PIC\_Timer0SysClkHandler(void);**

Description:default function to be called for SysClk interrupt handling , just increases time

Input Value: void

Return Value:void

Note : systime is a global value and it's alive form the start of the PIC till the end of the program

it's a local systime for the purpose of the application , do not mix with the concept of unix systime

**-Time\* MCAL\_PIC\_GetTime(void);**

Description:a getter to access systime

<b>Title</b>	GScrubber Class Documentation	<b>Revision</b>	A
		<b>Date</b>	03.01.2020

Input Value: void

Return Value: copy of SysTime structure is returned

**-void MCAL\_PIC\_ResetTime(void);**

Description:resets SysTime global to 0

Input Value: void

Return Value: copy of SysTime structure is returned

**-void MCAL\_PIC\_SysClkDef(void);**

Description:default space for SysClk interrupt

Input Value: void

Return Value:void

## Revision History

Rev	Date	Reason / Changes	Approval
A	03.01.2020	Initial release	RE