Software module to enable a bank of software timers with a resolution time of
one msecond for each. The timers can be individually started, stopped, expired, etc.

NOTES: Module uses TIMERS for its interrupts.

TIMERS_TEST (in the .c file) conditionally compiles the test harness for the code.
Make sure it is commented out for module usage.

Created on November 15, 2011, 9:54 AM

---

#define TIMER_ACTIVE 1
#define TIMER_EXPIRED 1
#define TIMER_NOT_ACTIVE 0
#define TIMER_NOT_EXPIRED 0

---

@Function TIMERS_Init(void)

@param none

@return None.

@brief Initializes the timer module

@author Max Dunne 2011.11.15

---

@Function SetTimer(unsigned char Num, unsigned int NewTime)

@param Num - the number of the timer to set.
@param NewTime - the number of milliseconds to be counted

@return ERROR or SUCCESS

@brief sets the NewTime into the chosen timer and clears any previous event flag
and sets the timer active to begin counting.

@author Max Dunne 2011.11.15

---

@Function StartTimer(unsigned char Num)

@param Num - the number of the timer to start

@return ERROR or SUCCESS

@brief simply sets the active flag in TMR_ActiveFlags to restart a stopped timer.

@author Max Dunne 2011.11.15

---

@Function StopTimer(unsigned char Num)

@param Num - the number of the timer to stop.

@return ERROR or SUCCESS

@brief simply clears the bit in TimerActiveFlags associated with this timer. This
will cause it to stop counting.

@author Max Dunne 2011.11.15

---

@Function InitTimer(unsigned char Num, unsigned int NewTime)

@param Num -  the number of the timer to start
@param NewTime - the number of tick to be counted

@return ERROR or SUCCESS

@brief sets the NewTime into the chosen timer and clears any previous event flag
*and sets the timer active to begin counting.

@author Max Dunne 2011.11.15

---

@Function IsTimerActive(unsigned char Num)

@param Num - the number of the timer to check

@return ERROR or TIMER_NOT_ACTIVE or TIMER_ACTIVE

@brief used to determine if a timer is currently counting.

@author Max Dunne 2011.11.15

---

@Function IsTimerExpired(unsigned char Num)

@param Num - the number of the timer to check

@return ERROR or TIMER_NOT_EXPIRED or TIMER_EXPIRED

@brief used to determine if a timer is currently expired.

@author Max Dunne 2011.11.15

---

@Function ClearTimerExpired(unsigned char Num)

@param Num - the timer whose event flag should be cleared.

@return ERROR or SUCCESS

@brief simply clears the appropriate bit in Event Flags to show that the event
has been serviced.

@author Max Dunne 2011.11.15

---

@Function GetTime

@param None

@return FreeRunningTimer, the current value of the module variable FreeRunningTimer

@remark Provides the ability to grab a snapshot time as an alternative to using
The library timers can be used to determine how long between 2 events.

@author Max Dunne
@date 2011.11.15

unsigned int GetTime(void);

@endif