Up/Down Timer

Design Document

# Algorithm:

UPTIMER:

1. Enter the preset value.
2. Press start button.
3. Start the timer.
4. When timer reaches the preset value, stop the timer.
5. Turn ON hooter.
6. If hooter reset PB is pressed, turn OFF hooter.
7. Repeat from Step (1) to Step (6).

DOWN TIMER:

1. Enter the preset value.
2. Press start button.
3. Start the timer.
4. Subtract the Preset value with the timer value.
5. When result becomes 0, stop the timer.
6. Turn ON hooter.
7. If hooter reset PB is pressed, turn OFF hooter.
8. Repeat from Step (1) to Step (7).

# 

# 

# Design Goals

* Up or Down counting
* Display formats:
  + hh:mm:ss
  + mm:ss
* Hooter or without hooter on reaching limits.

# Data Structures

## APP

STATE state //

UINT8 buffer[MAX\_BUF\_SIZE] // contains the timer data to be displayed

UINT8 presetBuffer[MAX\_BUF\_SIZE] // used to store preset value on which timer to be halted

UINT8 presetValue // used to store converted preset value

UINT8 rtcValue // used to store converted rtc value

UINT8 blinkIndex // used to switch to the next digit during set state

typedef enum

{

HALT\_STATE,

COUNT\_STATE,

SETTING\_STATE

}STATE;

# Interface

## config.h

#define HH\_MM\_SS

#define MM\_SS

#ifdef HH\_MM\_SS

#define MAX\_BUF\_SIZE 6

#elif MM\_SS

#define MAX\_BUF\_SIZE 4

#endif

## Functions

/\*------------------------------------------------------------------------------

\* void APP\_updateEEPROM (UINT8 from, UINT8\* buffer)

\*

\* Function to update the EERPOM with preset or current value of timer

\*

\* Input : from - Starting index from which data to be updated

Buffer - pointer to the data buffer

\* return value: none.

\*

\*------------------------------------------------------------------------------\*/

void APP\_updateEEPROM (UINT8 from, UINT8\* buffer)

/\*------------------------------------------------------------------------------

\* void APP\_readEEPROM (UINT8 from, UINT8\* buffer)

\*

\* Function to read from the EERPOM to get preset or timer value

\*

\* Input : from - Starting index from which data to be retrieved

Buffer - pointer to the data buffer where data to be stored

\* return value: none.

\*

\*------------------------------------------------------------------------------\*/

void APP\_updateEEPROM (UINT8 from, UINT8\* buffer)

/\*------------------------------------------------------------------------------

\* void APP\_handleHours (void)

\*

\* Function used to handle the hours when it crosses 23hrs in HH:MM:SS

\*

\*

\* return value: none.

\*

\*------------------------------------------------------------------------------\*/

void APP\_handleHours (void)

/\*------------------------------------------------------------------------------

\* void APP\_handleMinutes (void)

\*

\* Function used to handle the minutes when it crosses 59min in MM:SS

\* display

\*

\* return value: none.

\*

\*------------------------------------------------------------------------------\*/

void APP\_handleMinutes (void)

/\*------------------------------------------------------------------------------

\* void APP\_handleHooter (void)

\*

\* Function used to turn ON and turn OFF hooter

\*

\* return value: none.

\*

\*------------------------------------------------------------------------------\*/

void APP\_handleHooter (void)

/\*------------------------------------------------------------------------------

\* void APP\_handleHooter (void)

\*

\* Function used to turn ON and turn OFF hooter

\*

\* return value: none.

\*

\*------------------------------------------------------------------------------\*/

void APP\_handleHooter (void)

/\*------------------------------------------------------------------------------

\* void APP\_handleRTCdata (void)

\*

\* Function used to separate higher and lower nibbles of hours,

\* minutes, seconds and store it into the buffer

\*

\* return value: none.

\*

\*------------------------------------------------------------------------------\*/

void APP\_handleRTCdata (void)

/\*------------------------------------------------------------------------------

\* void APP\_resetPresetBuffer(void)

\*

\* Function used to reset Preset buffer with ASCII ‘0’

\*

\* return value: none.

\*

\*------------------------------------------------------------------------------\*/

void APP\_resetPresetBuffer(void)

/\*------------------------------------------------------------------------------

\* void APP\_resetDisplayBuffer(void)

\*

\* Function used to reset app display buffer with ASCII ‘0’

\*

\* return value: none.

\*

\*------------------------------------------------------------------------------\*/

void APP\_resetPresetBuffer(void)