CMPE-663 / EEEE-663 GRADUATE PROJECT
ULTRASOUND

Instructor: Weil
12/9/2019
Vinita Narayanamurthi - vn3128@rit.edu

Area of Focus

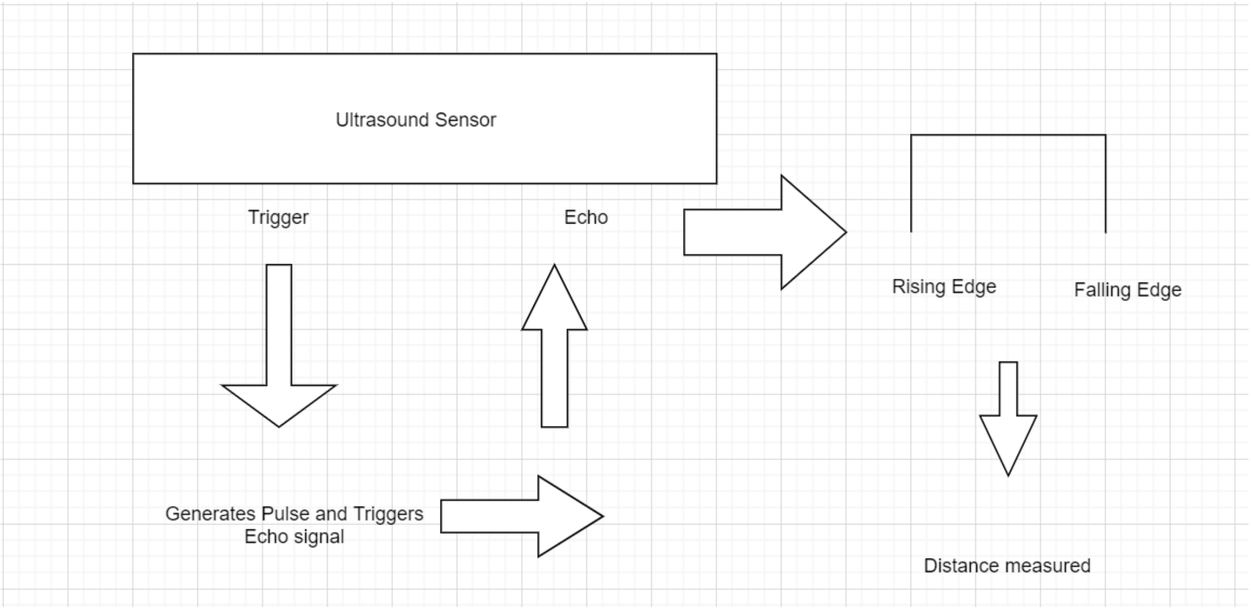
Member	GPIO Trigger Pulse and Echo	Timer – Input Capture	Post	User Interrupts	Distance Measurement	
Vinita	x	x	x	x		

Analysis / Design:

The project proposes to get the distance of the object using ultrasonic senor. The ultrasonic sensor consists of transmitter Trigger pulse, which should be triggered in order to start the process of distance measurement. The trigger pulse is given by a GPIO general purpose output pin which is kept high for atleast 10 micro seconds. This helps in the triggering of the echo pulse which starts shooting high. The echo pulse touches the object and returns the signal back to the receiver. The up shooting and the receiving back of the signal gives the round trip time taken to reach the object. The distance is calculated using the formula

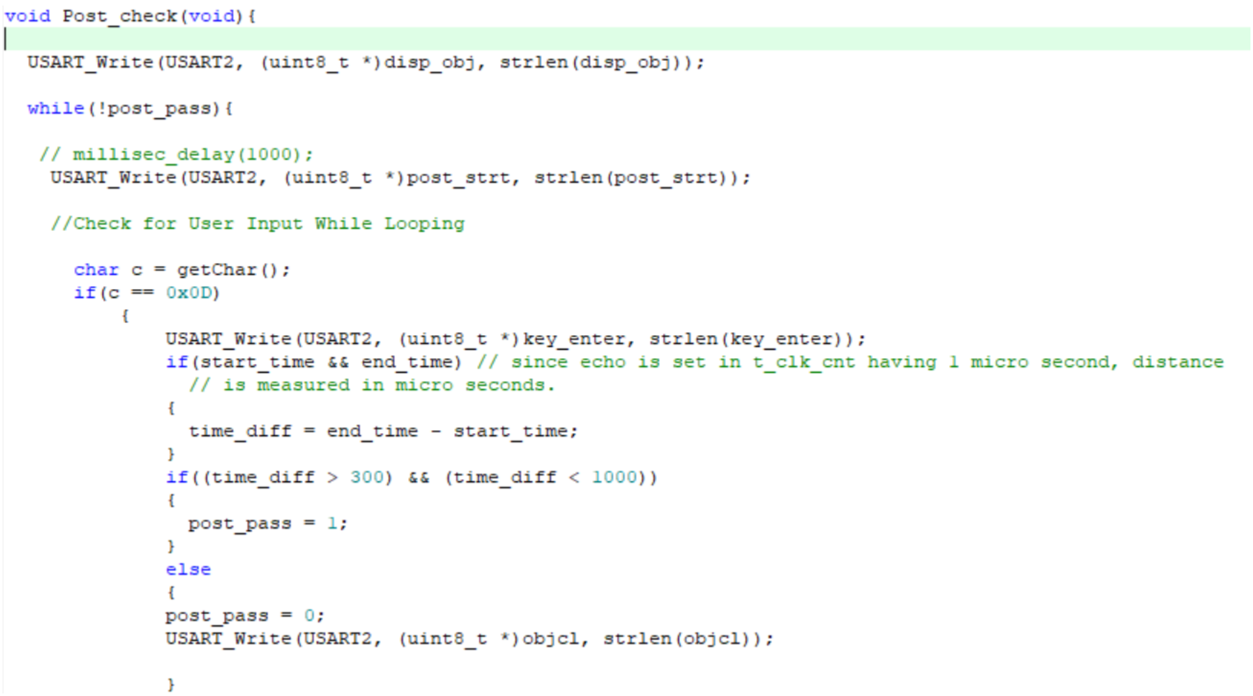
Distance = (Round trip time taken/2)*Speed of sound.

A general purpose GPIO output pin is set for the trigger pulse. This pin is kept on for 10 microseconds. The pulse from the trigger pin is responsible for triggering the echo pulse having 8 cycles of 40 KHZ frequency. The echo pulse GPIO is set as alternate input function mode which is connected to timer 2 channel 1 for input capture. The figure below shows the GPIO setup.



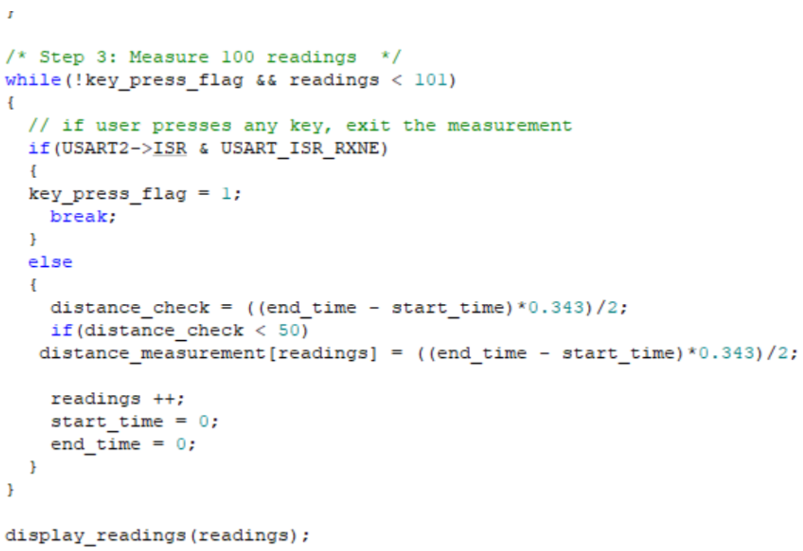
POST SETUP:

The post function test whether the object is in measurable distance of 300 to 1000 mm. If not the post is repeated until the requirement is met.



```
void Post_check(void){

  USART_Write(USART2, (uint8_t *)disp_obj, strlen(disp_obj));

  while(!post_pass){

   // millisec_delay(1000);
    USART_Write(USART2, (uint8_t *)post_strt, strlen(post_strt));

    //Check for User Input While Looping

      char c = getChar();
      if(c == 0x0D)
          {
              USART_Write(USART2, (uint8_t *)key_enter, strlen(key_enter));
              if(start_time && end_time) // since echo is set in t_clk_cnt having 1 micro second, distance
                // is measured in micro seconds.
              {
                time_diff = end_time - start_time;
              }
              if((time_diff > 300) && (time_diff < 1000))
              {
                post_pass = 1;
              }
              else
              {
              post_pass = 0;
              USART_Write(USART2, (uint8_t *)objcl, strlen(objcl));

              }
```

The post test is followed by taking 100 readings and checking the readings for the outliers.



```
;

/* Step 3: Measure 100 readings  */
while(!key_press_flag && readings < 101)
{
  // if user presses any key, exit the measurement
  if(USART2->ISR & USART_ISR_RXNE)
  {
  key_press_flag = 1;
    break;
  }
  else
  {
    distance_check = ((end_time - start_time)*0.343)/2;
    if(distance_check < 50)
   distance_measurement[readings] = ((end_time - start_time)*0.343)/2;

    readings ++;
    start_time = 0;
    end_time = 0;
  }
}

display_readings(readings);
```

The values are then sorted and min, max and running average are found and the measurements are recorded in a csv file.

Project Results:

Lessons Learned:

The GPIO and timer configuration of both the trigger as well as echo pins though used previously, required a deeper insight for understanding the exact process running in the background. The ultrasonic distance measurement is a new lesson learnt in this project. We are still facing some issues while capturing the value of CCR1 register during both the rising and falling edges of the echo pulse.