CMPE-663 / EEEE-663 GRADUATE PROJECT
ULTRASOUND

Instructor: Weil
12/11/2019
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.

Trigger Pin PWM – Output Compare:

For the trigger pin, PWM signal is used in output compare mode. The prescalar and the CCR register value are given to be 79 and 10 respectively such that PWM signal has 100% duty cycle. For each occurrence of the PWM on signal, the transmitter pings a high pulse. This pulse triggers the echo pin which starts the echo pulse generation. The echo pulse hits the objects and comes back to the receiver. The time taken for the echo pulse to return back gives the round trip period. The distance is then calculated using the above formula. The GPIO and timer configuration for the input capture and output compare mode is configured.

Echo Pulse – Input Capture:

After the trigger pulse is emitted, the echo pulse is starts to charge. This gives us the rising edge. The return of the echo pulse to the receiver gives us the falling edge. We need to capture both the rising as well as the falling edge in order to calculated the round trip time. We need to enable both CC1P and CC1NP of CCER register, to capture both rising and falling edges. Whenever an edge occurs (rising edge or falling edge) , SR as well as CC1IF flag is set. Reading the value of capture compare register CCR1 resets both the flag. We are thus reading two consecutive values from CCR1 register. The first one is start time (rising edge) and the second one is end time(falling edge)

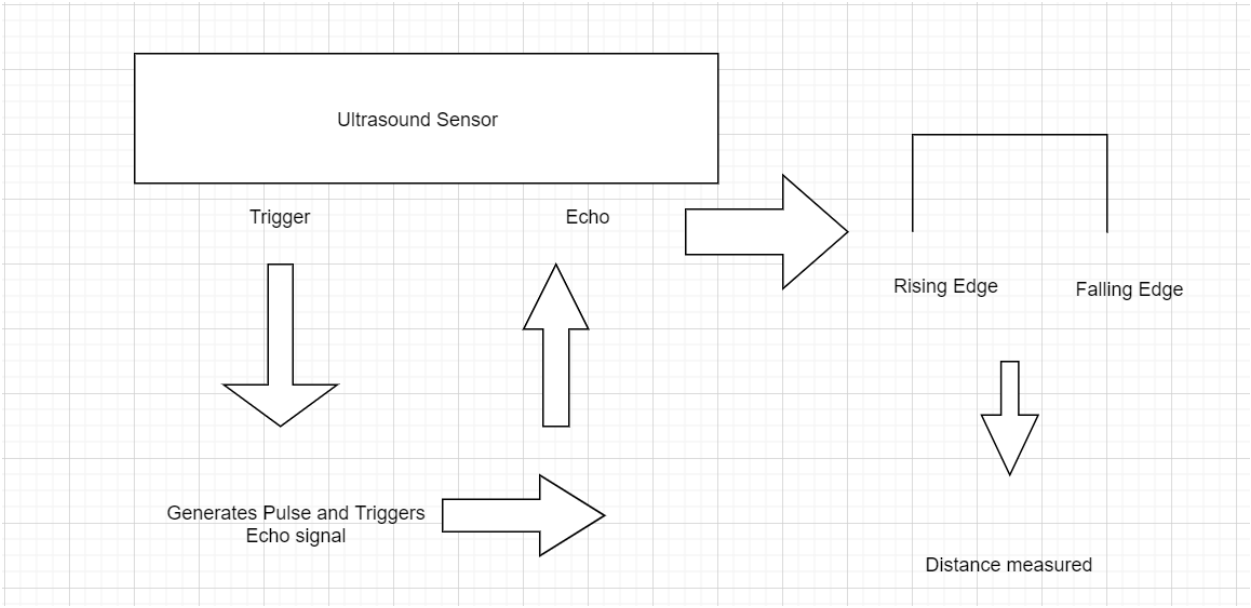

Figure 1: Working of Ultrasonic sensor

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 4: Measure 100 readings  */
uint32_t readings = 0;
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
  {
    if(  TIM2->SR &(TIM_SR_CC1IF))
        {
        startTime =TIM2->CCR1;
        }
    if(TIM2->SR &(TIM_SR_CC1IF))
        {
        endTime = TIM2->CCR1;
        }
    distance_check = ((end_time - start_time)*0.343)/2;
    readings ++;

   if((distance_check <= 50) || (distance_check>= 1000))// Check for outliers
   {


   while(distance_check > 50 && distance_check <1000)  // To correct the outliers
   {

     distance_check =0;
     start_time = 0;
    end_time = 0;
    if(  TIM2->SR &(TIM_SR_CC1IF))
     {
     startTime =TIM2->CCR1;
     }
    if(TIM2->SR &(TIM_SR_CC1IF))
     {
     endTime = TIM2->CCR1;
     }
     distance_check = ((end_time - start_time)*0.343)/2;
   }

 }
 (distance_measurement[readings] = ((end_time - start_time)*0.343)/2);


  start_time = 0;
  end_time = 0;
  distance_check =0;
}
```

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

Lessons Learned:

I learnt the significance of combing the input capture and output compare mode. 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.