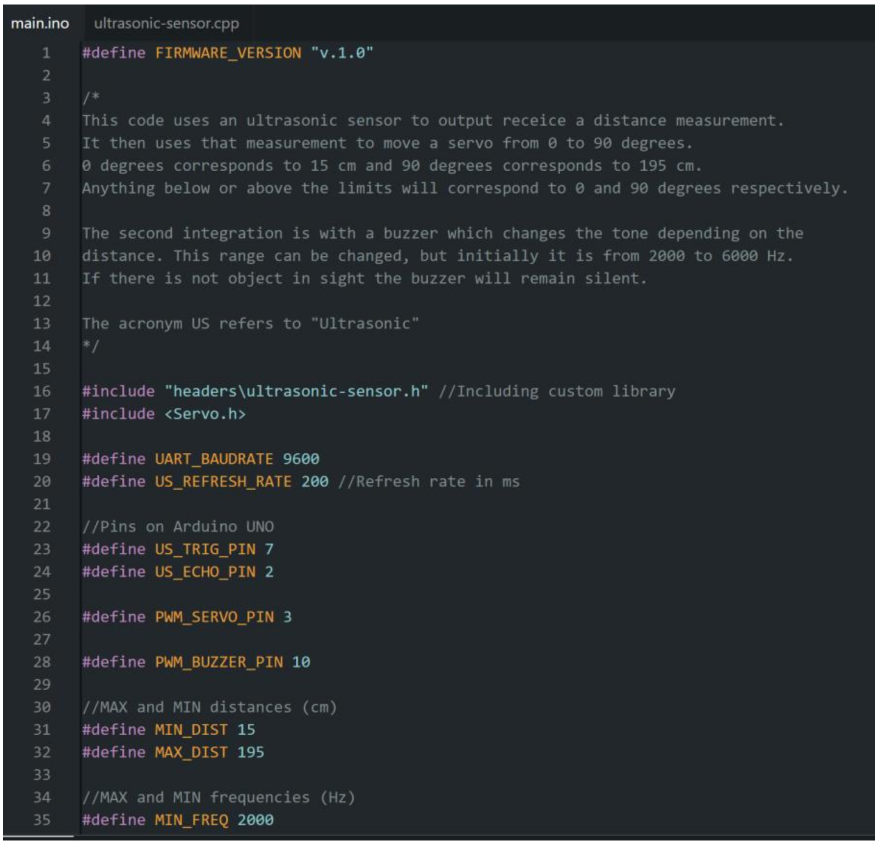
Link to video:

https://youtu.be/jQcl_k7-Zfg

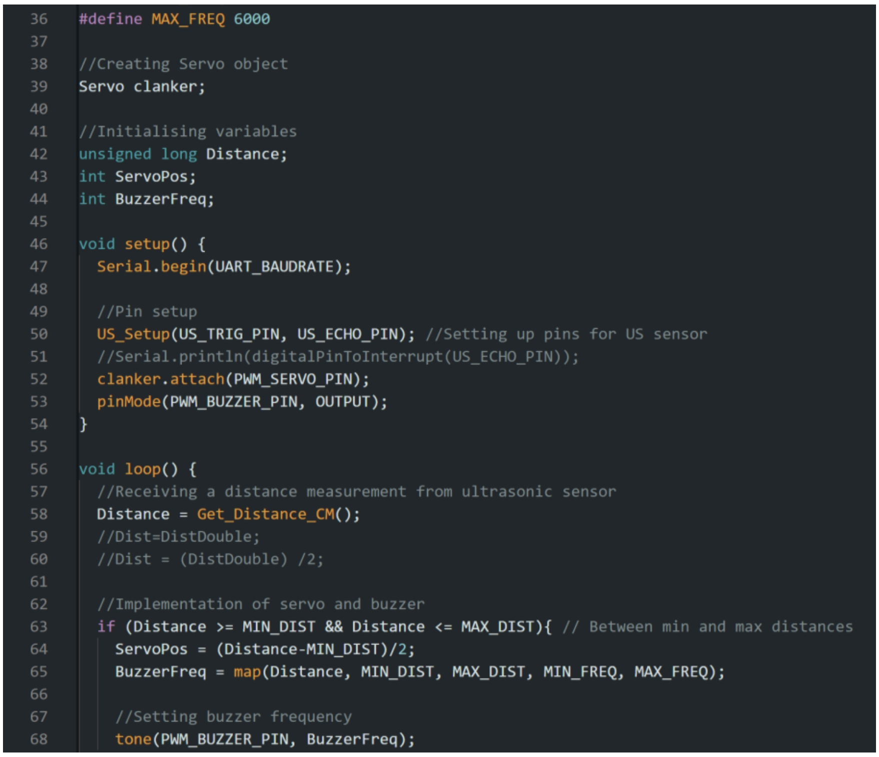
Link to GitHub:

https://github.com/Tottocs/Ultrasound-Sensor-Integration

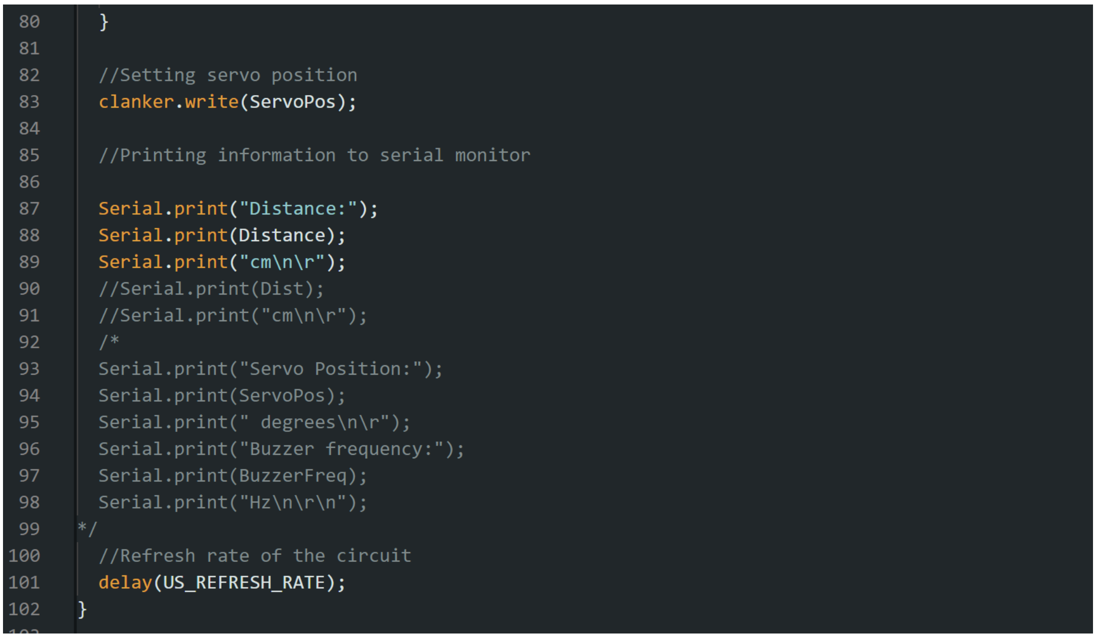
main.ino File



```
#define FIRMWARE_VERSION "v.1.0"

/*
This code uses an ultrasonic sensor to output receice a distance measurement.
It then uses that measurement to move a servo from 0 to 90 degrees.
0 degrees corresponds to 15 cm and 90 degrees corresponds to 195 cm.
Anything below or above the limits will correspond to 0 and 90 degrees respectively.

The second integration is with a buzzer which changes the tone depending on the
distance. This range can be changed, but initially it is from 2000 to 6000 Hz.
If there is not object in sight the buzzer will remain silent.

The acronym US refers to "Ultrasonic"
*/

#include "headers\ultrasonic-sensor.h" //Including custom library
#include <Servo.h>

#define UART_BAUDRATE 9600
#define US_REFRESH_RATE 200 //Refresh rate in ms

//Pins on Arduino UNO
#define US_TRIG_PIN 7
#define US_ECHO_PIN 2

#define PWM_SERVO_PIN 3

#define PWM_BUZZER_PIN 10

//MAX and MIN distances (cm)
#define MIN_DIST 15
#define MAX_DIST 195

//MAX and MIN frequencies (Hz)
#define MIN_FREQ 2000
```

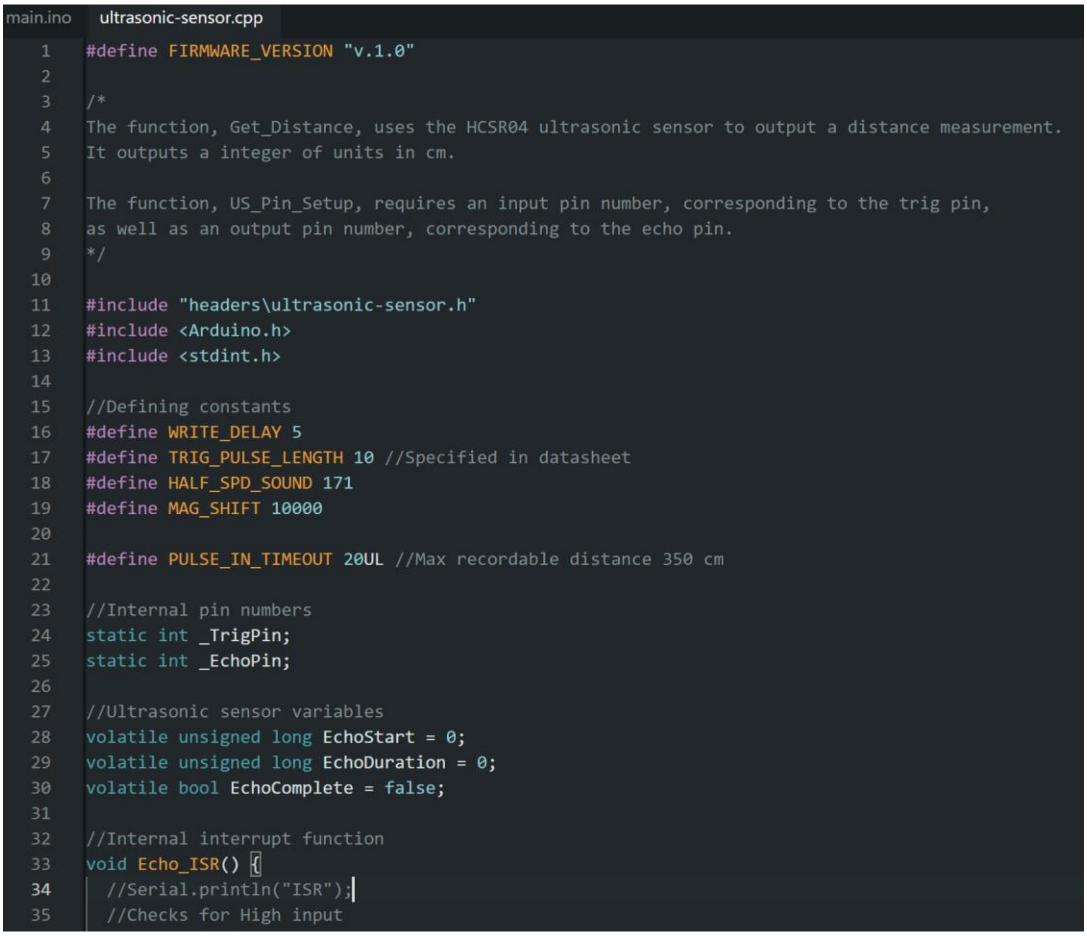


```
#define MAX_FREQ 6000

//Creating Servo object
Servo clanker;

//Initialising variables
unsigned long Distance;
int ServoPos;
int BuzzerFreq;

void setup() {
  Serial.begin(UART_BAUDRATE);

  //Pin setup
  US_Setup(US_TRIG_PIN, US_ECHO_PIN); //Setting up pins for US sensor
  //Serial.println(digitalPinToInterrupt(US_ECHO_PIN));
  clanker.attach(PWM_SERVO_PIN);
  pinMode(PWM_BUZZER_PIN, OUTPUT);
}

void loop() {
  //Receiving a distance measurement from ultrasonic sensor
  Distance = Get_Distance_CM();
  //Dist=DistDouble;
  //Dist = (DistDouble) /2;

  //Implementation of servo and buzzer
  if (Distance >= MIN_DIST && Distance <= MAX_DIST){ // Between min and max distances
    ServoPos = (Distance-MIN_DIST)/2;
    BuzzerFreq = map(Distance, MIN_DIST, MAX_DIST, MIN_FREQ, MAX_FREQ);

    //Setting buzzer frequency
    tone(PWM_BUZZER_PIN, BuzzerFreq);
```

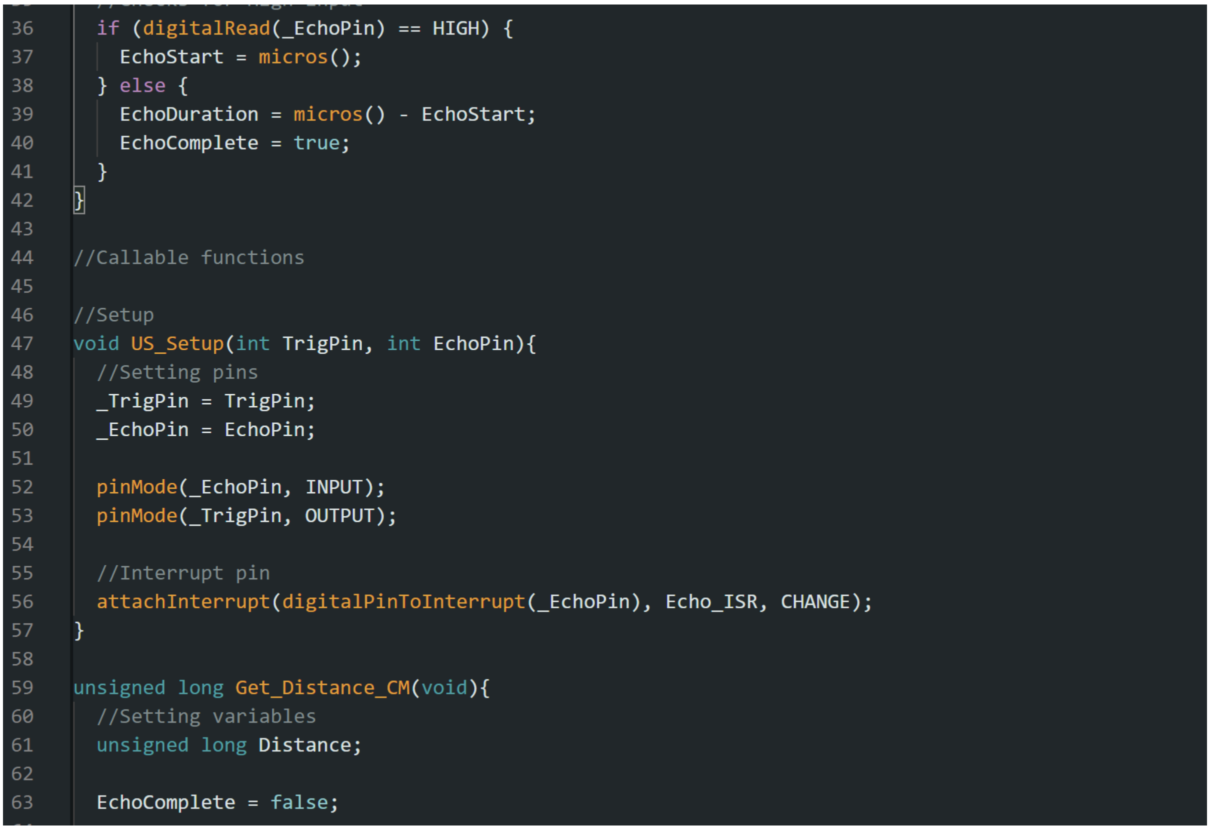


```
  }

  //Setting servo position
  clanker.write(ServoPos);

  //Printing information to serial monitor

  Serial.print("Distance:");
  Serial.print(Distance);
  Serial.print("cm\n\r");
  //Serial.print(Dist);
  //Serial.print("cm\n\r");
  /*
  Serial.print("Servo Position:");
  Serial.print(ServoPos);
  Serial.print(" degrees\n\r");
  Serial.print("Buzzer frequency:");
  Serial.print(BuzzerFreq);
  Serial.print("Hz\n\r\n");
*/
  //Refresh rate of the circuit
  delay(US_REFRESH_RATE);
}
```

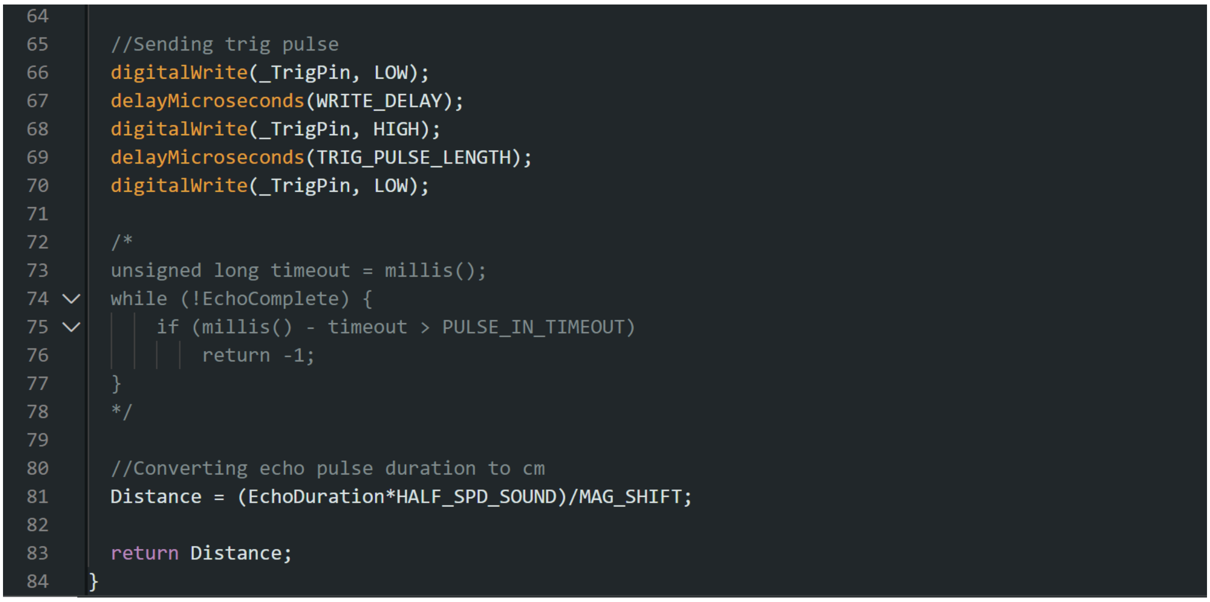
ultrasonic-sensor.cpp File



```
#define FIRMWARE_VERSION "v.1.0"

/*
The function, Get_Distance, uses the HCSR04 ultrasonic sensor to output a distance measurement.
It outputs a integer of units in cm.

The function, US_Pin_Setup, requires an input pin number, corresponding to the trig pin,
as well as an output pin number, corresponding to the echo pin.
*/

#include "headers\ultrasonic-sensor.h"
#include <Arduino.h>
#include <stdint.h>

//Defining constants
#define WRITE_DELAY 5
#define TRIG_PULSE_LENGTH 10 //Specified in datasheet
#define HALF_SPD_SOUND 171
#define MAG_SHIFT 10000

#define PULSE_IN_TIMEOUT 20UL //Max recordable distance 350 cm

//Internal pin numbers
static int _TrigPin;
static int _EchoPin;

//Ultrasonic sensor variables
volatile unsigned long EchoStart = 0;
volatile unsigned long EchoDuration = 0;
volatile bool EchoComplete = false;

//Internal interrupt function
void Echo_ISR() {
  //Serial.println("ISR");
  //Checks for High input
```

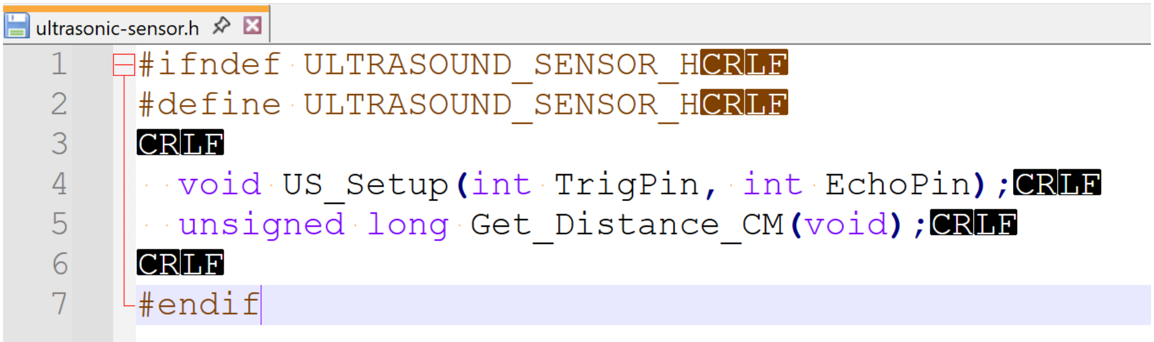


```
  //Checks for high input
  if (digitalRead(_EchoPin) == HIGH) {
    EchoStart = micros();
  } else {
    EchoDuration = micros() - EchoStart;
    EchoComplete = true;
  }
}

//Callable functions

//Setup
void US_Setup(int TrigPin, int EchoPin){
  //Setting pins
  _TrigPin = TrigPin;
  _EchoPin = EchoPin;

  pinMode(_EchoPin, INPUT);
  pinMode(_TrigPin, OUTPUT);

  //Interrupt pin
  attachInterrupt(digitalPinToInterrupt(_EchoPin), Echo_ISR, CHANGE);
}

unsigned long Get_Distance_CM(void){
  //Setting variables
  unsigned long Distance;

  EchoComplete = false;
```



```

  //Sending trig pulse
  digitalWrite(_TrigPin, LOW);
  delayMicroseconds(WRITE_DELAY);
  digitalWrite(_TrigPin, HIGH);
  delayMicroseconds(TRIG_PULSE_LENGTH);
  digitalWrite(_TrigPin, LOW);

  /*
  unsigned long timeout = millis();
  while (!EchoComplete) {
      if (millis() - timeout > PULSE_IN_TIMEOUT)
          return -1;
  }
  */

  //Converting echo pulse duration to cm
  Distance = (EchoDuration*HALF_SPD_SOUND)/MAG_SHIFT;

  return Distance;
}
```

Header file



```
#ifndef ULTRASOUND_SENSOR_H
#define ULTRASOUND_SENSOR_H

  void US_Setup(int TrigPin, int EchoPin);
  unsigned long Get_Distance_CM(void);

#endif
```