CS 225/226 - MINI PROJECT

# PART-2

## Mohit Kishore (1701CS28)

**AIM:** To construct a mini, compact sonar with display.

**MOTIVATION:** This experiment has been performed to familiarize one with the working of an existing electronics project.

**COMPONENTS USED:**

1. HC-SR04 Ultrasonic Sensor.
2. 1.8” ST7735 display.
3. SG90 Micro Servo Motor.
4. Arduino Nano.
5. Jumper Wires.

**CRITICAL ANALYSIS:**

This project has been made to understand the working of an electronics project. In this project:

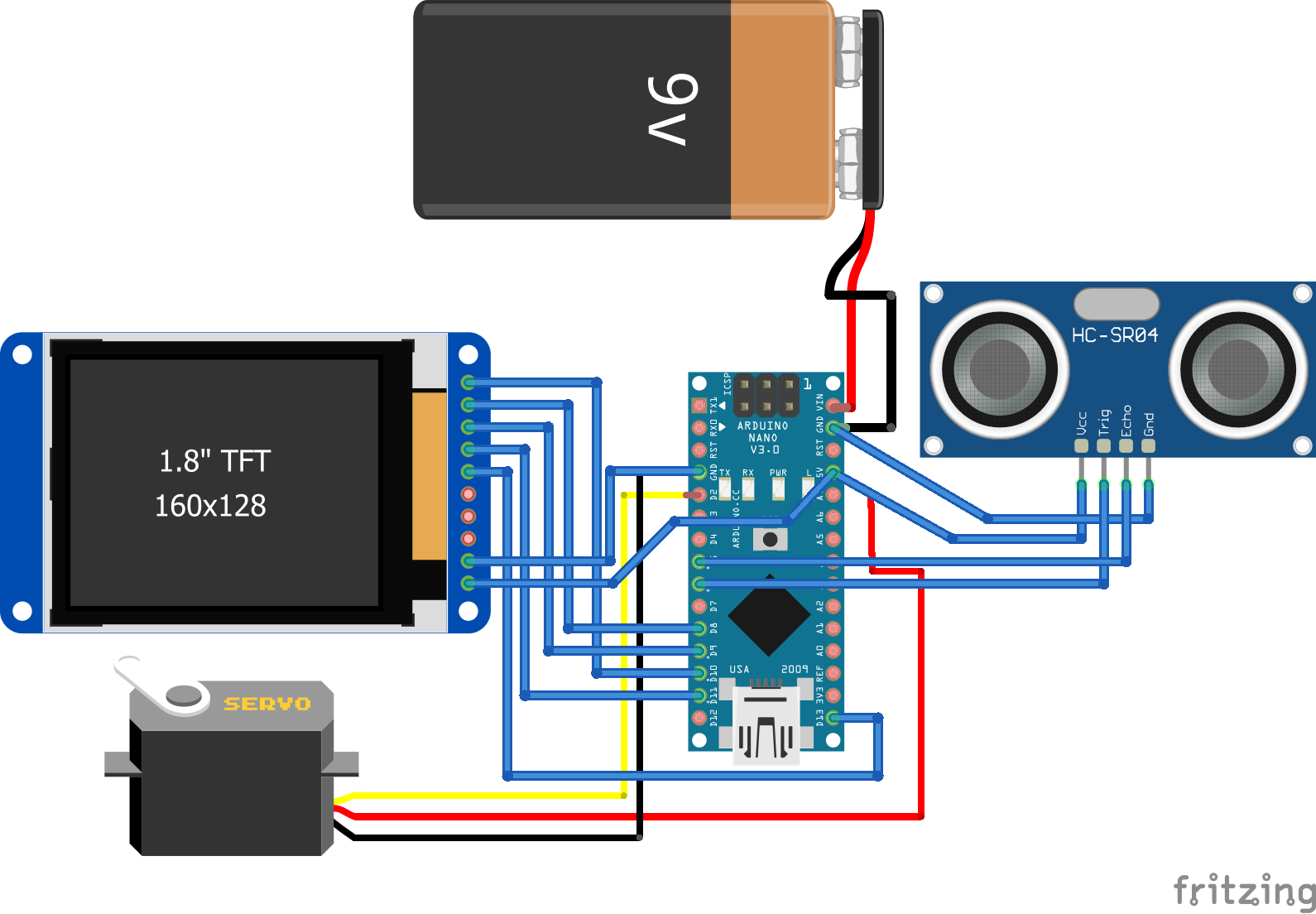
* Arduino Nano is used as the brains.
* HC-SR04 ultrasonic sensor is used to generate and detect ultrasonic waves. The reflected waves are detected to produce a value in proportion to evaluate the presence of an object.
* SG90 micro servo is used to rotate the ultrasonic sensor in 180 degrees.
* ST7735 display is the monitor used to visualise all data on the screen.

The direction of the object is shown as red lines in the sonar.

**CIRCUIT DIAGRAM**:

The circuit diagram is provided below.

NOTE: Instead of pin D2, D3 has been used to connect servo, as written in the code.



**APPLICATION:**

The concept of sonar finds application in the following places:

* It is used for estimation of range and direction of arrival of objects such as vessels or submarines. It basically functions as depth finder also.
* It is used for communication between sub-merged submarines or between a submarine and a surface vessel.
* It is used for locating mines and underwater hazards in order to achieve safe navigation.
* It is widely used by commercial fishermen for fish finding.
* It is used for seafloor mapping and seafloor imaging.
* SONAR concept is being applied to medical imaging. Here high frequency sound waves are subjected on body and echoes are recorded to create image called sonogram.

**SOURCE CODE:**

The source code of the program is provided below.

#include <Servo.h>.

#include <SPI.h>

#include "Ucglib.h"

const int trigPin = 6;

const int echoPin = 5;

int Ymax = 128;

int Xmax = 160;

int base = 8;

int pos = base+6;

int deg=0;

int x;

int val =200;

int j = 2;

Servo myServo;

long duration;

int distance;

int k;

Ucglib\_ST7735\_18x128x160\_HWSPI ucg(/\*cd=\*/ 9, /\*cs=\*/ 10, /\*reset=\*/ 8);

void setup(void)

{

delay(1000);

myServo.write(80);

// ucg.begin(UCG\_FONT\_MODE\_TRANSPARENT);

ucg.begin(UCG\_FONT\_MODE\_SOLID);

ucg.setFont(ucg\_font\_6x10\_tr);

ucg.clearScreen();

ucg.setRotate270();

pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output

pinMode(echoPin, INPUT); // Sets the echoPin as an Input

Serial.begin(9600);

myServo.attach(3); // Defines on which pin is the servo motor attached

}

void loop(void)

{

fix();

for ( x=80; x >= 10; x--){

distance = calculateDistance();

Serial.println(distance);

k = map(x, 80, 10, 15,165);

myServo.write(k);

if (distance < 30){

int f = x+6;

ucg.setColor(255, 0, 0);

ucg.drawLine(Xmax/2, pos, -val\*cos(radians(f\*2)),val\*sin(radians(f\*2)));

}

ucg.setColor(0, 207, 0);

ucg.drawLine(Xmax/2, pos, -200\*cos(radians(x\*2)),200\*sin(radians(x\*2)));

int d = x+1;

ucg.setColor(0, 207, 0);

//ucg.drawLine(Xmax/2, pos, -200\*cos(radians(d\*2)),200\*sin(radians(d\*2)));

int c = x+2;

ucg.setColor(0, 207, 0);

//ucg.drawLine(Xmax/2, pos, -200\*cos(radians(c\*2)),200\*sin(radians(c\*2)));

int b = x+3;

ucg.setColor(0, 102, 0);

//ucg.drawLine(Xmax/2, pos, -200\*cos(radians(b\*2)),200\*sin(radians(b\*2)));

int a = x+4;

ucg.setColor(0, 102, 0);

//ucg.drawLine(Xmax/2, pos, -200\*cos(radians(a\*2)),200\*sin(radians(a\*2)));

int e = x+5;

ucg.setColor(0, 0, 0);

ucg.drawLine(Xmax/2, pos, -200\*cos(radians(e\*2)),200\*sin(radians(e\*2)));

ucg.setColor(255, 0, 0);

ucg.setPrintPos(160,0);

ucg.setPrintDir(2);

ucg.print("Deg :");

deg = map (x, 80, 10 , 0, 180);

ucg.setPrintPos(120,0);

ucg.setPrintDir(2);

ucg.print(deg);

ucg.setPrintPos(10,0);

ucg.print(distance);

ucg.setColor(0, 0, 255);

ucg.setPrintPos(90,38);

ucg.setPrintDir(2);

ucg.print("0.25");

ucg.setPrintPos(90,70);

ucg.print("0.50");

ucg.setPrintPos(90,110);

ucg.print("1.00");

}

// ucg.clearScreen();

fix();

for ( x=10; x <= 80; x++){

distance = calculateDistance();

Serial.println(distance);

k = map(x, 10, 80, 165,15);

myServo.write(k);

if (distance < 10){

int e = x-5;

ucg.setColor(255, 0, 0);

ucg.drawLine(Xmax/2, pos, -val\*cos(radians(e\*2)),val\*sin(radians(e\*2)));

}

ucg.setColor(0, 207, 0);

ucg.drawLine(Xmax/2, pos, -200\*cos(radians(x\*2)),200\*sin(radians(x\*2)));

int a = x-1;

//ucg.drawLine(Xmax/2, pos, -200\*cos(radians(a\*2)),200\*sin(radians(a\*2)));

int b = x-2;

ucg.setColor(0, 102, 0);

//ucg.drawLine(Xmax/2, pos, -200\*cos(radians(b\*2)),200\*sin(radians(b\*2)));

int c = x-3;

ucg.setColor(0, 102, 0);

//ucg.drawLine(Xmax/2, pos, -200\*cos(radians(c\*2)),200\*sin(radians(c\*2)));

int d = x-4;

ucg.setColor(0, 0, 0);

ucg.drawLine(Xmax/2, pos, -200\*cos(radians(d\*2)),200\*sin(radians(d\*2)));

ucg.setColor(255, 0, 0);

ucg.setPrintPos(160,0);

ucg.setPrintDir(2);

ucg.print("Deg :");

deg = map (x, 10, 80 , 0, 180);

ucg.setPrintPos(120,0);

ucg.setPrintDir(2);

ucg.print(deg);

ucg.setPrintPos(10,0);

ucg.print(distance);

ucg.setColor(0, 0, 255);

ucg.setPrintPos(90,38);

ucg.setPrintDir(2);

ucg.print("0.25");

ucg.setPrintPos(90,70);

ucg.print("0.50");

ucg.setPrintPos(90,110);

ucg.print("1.00");

}

//ucg.clearScreen();

}

void fix(){

ucg.setColor(255, 0, 0);

ucg.drawDisc(Xmax/2, base, 5, UCG\_DRAW\_LOWER\_RIGHT);

ucg.drawDisc(Xmax/2, base, 5, UCG\_DRAW\_LOWER\_LEFT);

ucg.setColor(225, 255, 50);

ucg.drawCircle(80, base, 115, UCG\_DRAW\_LOWER\_RIGHT);

ucg.drawCircle(80, base, 115, UCG\_DRAW\_LOWER\_LEFT);

ucg.drawCircle(80, base, 78, UCG\_DRAW\_LOWER\_RIGHT);

ucg.drawCircle(80, base, 78, UCG\_DRAW\_LOWER\_LEFT);

ucg.drawCircle(80, base, 40, UCG\_DRAW\_LOWER\_RIGHT);

ucg.drawCircle(80, base, 40, UCG\_DRAW\_LOWER\_LEFT);

ucg.drawLine(0, base, 160,base);

ucg.setColor(0, 0, 0);

ucg.drawBox(100, 0, 30, 8);

}

int calculateDistance(){

digitalWrite(trigPin, LOW);

delayMicroseconds(2);

// Sets the trigPin on HIGH state for 10 micro seconds

digitalWrite(trigPin, HIGH);

delayMicroseconds(10);

digitalWrite(trigPin, LOW);

duration = pulseIn(echoPin, HIGH); // Reads the echoPin, returns the sound wave travel time in microseconds

distance= duration\*0.034/2;

return distance;

}