



Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное
учреждение высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ «Специальное машиностроение»

КАФЕДРА «Колесные машины»

Отчёт о выполнении лабораторной работы №4
по курсу
«Управление техническими системами»
на тему
«Изучение принципов работы ультразвукового датчика
расстояния»

Студент СМ10-71

(подпись, дата)

В.Б. Сухоносенко

(Ф.И.О.)

Преподаватель

(подпись, дата)

А.А. Смирнов

(Ф.И.О.)

2025 г.

Содержание

1	Исходные данные	3
2	Выполнение работы	4
2.1	Тарировка датчика	4
2.2	Программа Arduino	6
2.3	Построение диаграммы направленности датчика	10
3	Вывод	14

1 Исходные данные

Задачей лабораторной работы №4 являлось:

1. Изучение теоретических основ,
2. Проведение тарировки ультразвукового датчика расстояния с использованием платы Arduino и АЦП L-Card E14-140M,
3. Разработка управляющей программы для индикации расстояния до преграды,
4. Написание управляющей программы для микроконтроллера,
5. Загрузка программы в микроконтроллер и ее тестирование,
6. Построение диаграммы направленности датчика для препятствия в виде столбика.

На рис. 1.1 показана плата индикации с подключенным модулем АЦП и процесс получения данных для диаграммы направленности датчика.

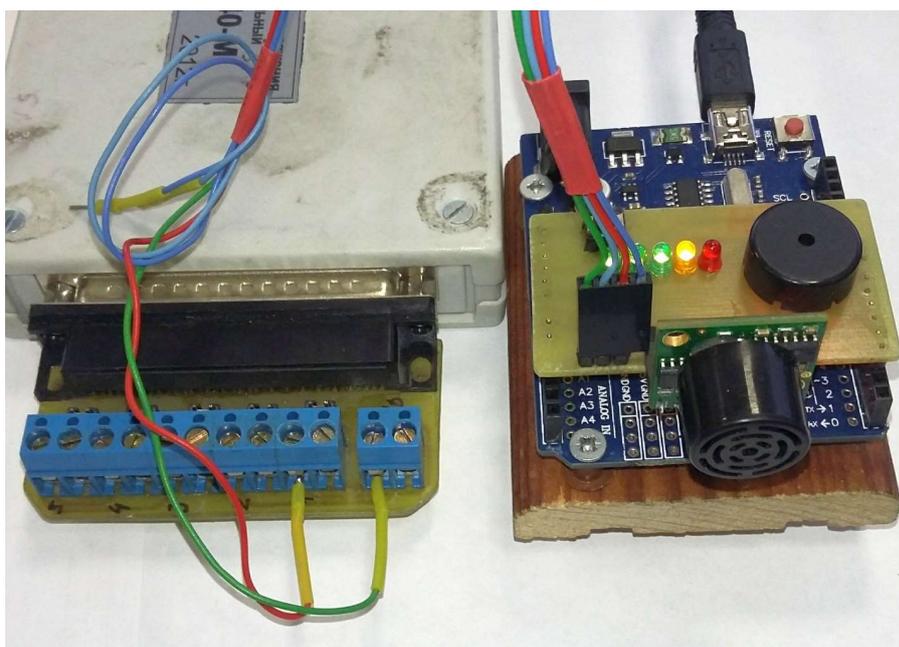


Рисунок 1.1 — Плата индикации

2 Выполнение работы

2.1 Тарировка датчика

Для платы Arduino написана программа, выводящая в последовательный порт длительность импульса, посылаемого датчиком, в микросекундах. Эта длительность пропорциональна расстоянию, которое регистрирует ультразвуковой датчик.

```
// Программа для работы с сонаром LV-MaxSonar-EZ1.  
// Измерение длины импульса  
int SonarPWPin = 2; //пин Arduino, подключенный к пину PW сонара  
  
void setup() {  
  pinMode(SonarPWPin, INPUT);  
  Serial.begin(9600); //инициализируем послед. порт передачи данных  
}  
  
void loop() {  
  unsigned long time_mks = pulseIn(SonarPWPin, HIGH); //длина импульса  
  //в мкс  
  //печатаем в последовательный порт ширину импульса  
  Serial.print("Pulse width is: ");  
  Serial.print(time_mks);  
  Serial.print(" mks\n"); //переход на новую строку  
  delay(500);  
}
```

Построена таблица Excel (см. рисунок 2.1) - для вручную выставленного с помощью рулетки расстояния между датчиком и стеной в диапазоне 0,5-3 м записывалось выдаваемое значение в микросекундах.

С помощью LGraph записывался график напряжения на пине, соответствующим выводу датчика.

В MATLAB построен график напряжения на этом пине и измерена погрешность, с которой Arduino выводит время в микросекундах по сравнению с тем, что посылает датчик на самом деле - рисунок 2.2.

Расстояние	Длина импульса		Погрешность %	Коэф-т пропорциональности мкс/м
	Arduino	АЦП+MATLAB		
1,00	5550	5750	-3,48	5550
1,50	8350	8750	-4,57	5567
2,00	11143	11500	-3,10	5572
2,50	14085	14000	0,61	5634
0,50	2750	3000	-8,33	5500
			Среднее значение	5564

Рисунок 2.1 — Расчет коэффициента пропорциональности в Excel

Программа для расчета длины импульса в MATLAB:

```

time=uts1m(:,1); %выделяем вектор времени (1-й столбец матрицы expe1)
U = uts1m(:,2); %выделяем вектор напряжений (2-й столбец матрицы expe1)
time=time*1e6;
figure()
tiledlayout('Flow')
nexttile;
%Строим графики
plot (time, U, 'k', 'LineWidth',1);
title('Напряжение, В');
xlabel('Время, мкс');
nexttile;
plot (time, U, 'k', 'LineWidth',1);
xlim([1.29e6 1.36e6])
title('(Приближено)');
xlabel('Время, мкс');
grid on;

time_impulse_mks= 1355000-1343250
% Полученная длина импульса
%в микросекундах (вариант 3, равна 11750)

```

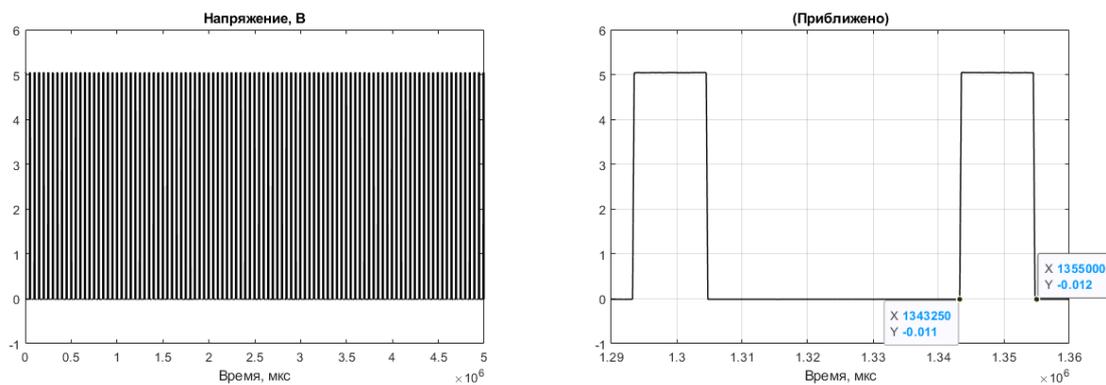


Рисунок 2.2 — Вычисление длины импульса в MATLAB

Среднее значение коэффициента пропорциональности между временем импульса и расстоянием - 5564 мкс/м.

2.2 Программа Arduino

Программа написана в соответствии с вариантом 3 (таблица 2.1) и алгоритмом, представленным на блок-схеме - рисунок 2.3.

Таблица 2.1 — Варианты индикации светодиодов в зависимости от расстояния

Цветовая индикация	Расстояние до препятствия, м					Диапазон
	Вариант 1	Вариант 2	Вариант 3	Вариант 4	Вариант 5	
○ ○ ○ ○ ○	> 3,5	> 4,0	> 3,0	> 2,8	> 3,2	D1
● ○ ○ ○ ○	3,5 ... 2,5	4,0 ... 3,0	3,0 ... 2,0	2,8 ... 2,2	3,2 ... 2,2	D2
● ● ○ ○ ○	2,5 ... 1,5	3,0 ... 2,0	2,0 ... 1,0	2,2 ... 1,2	2,2 ... 1,2	D3
● ● ● ○ ○	1,5 ... 1,0	2,0 ... 1,0	1,0 ... 0,5	1,2 ... 0,6	1,2 ... 0,8	D4
● ● ● ● ○	1,0 ... 0,3	1,0 ... 0,5	0,5 ... 0,2	0,6 ... 0,3	0,8 ... 0,5	D5
● ● ● ● ●	< 0,3	< 0,5	< 0,2	< 0,3	< 0,5	D6

Программа представлена ниже. Закомментированные строки с тоном - возможность установить разный тон для каждого из диапазонов расстояний.

```
//VOLODYA ARTYOM 2025
```

```
int SonarPWPin = 2; //пин Arduino, подключенный к пину PW сонара
int SpeakerPin = 3; //пин для подключения пьезодинамика
int LED1Pin = 4; //пин светодиода №1
```

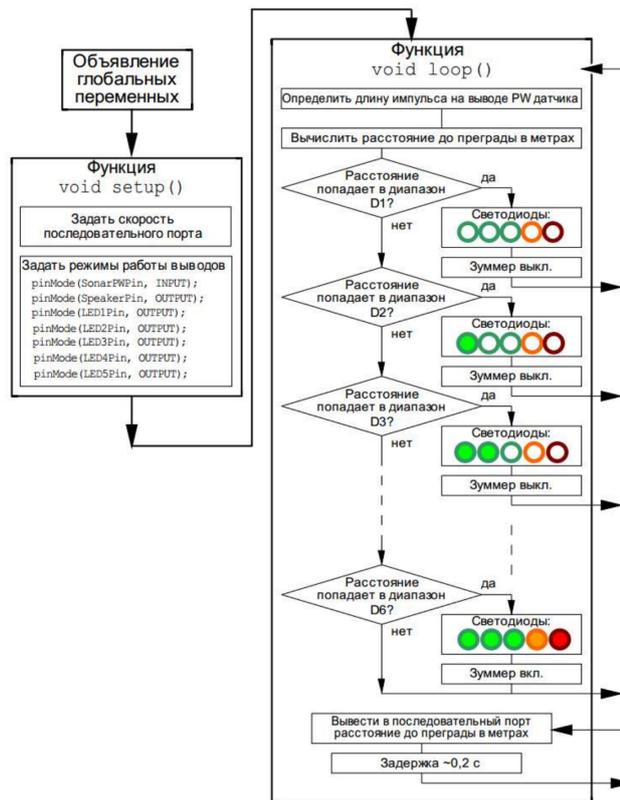


Рисунок 2.3 — Блок-схема алгоритма программы индикации

```

int LED2Pin = 8; //пин светодиода №2
int LED3Pin = 5; //пин светодиода №3
int LED4Pin = 7; //пин светодиода №4
int LED5Pin = 6; //пин светодиода №5
float distance = 0;
float scale = 5564; //коэф. пропорциональности [мкс/м]

```

```

void setup() {
pinMode(SonarPWPin,INPUT);
Serial.begin(9600); //инициализируем послед. порт передачи данных
pinMode(SpeakerPin,OUTPUT);
pinMode(LED1Pin,OUTPUT);
pinMode(LED2Pin,OUTPUT);
pinMode(LED3Pin,OUTPUT);
pinMode(LED4Pin,OUTPUT);
pinMode(LED5Pin,OUTPUT);
}

```

```

//В мкс

}

void loop() {
unsigned long time_mks = pulseIn(SonarPWPin, HIGH); //длина импульса

distance = time_mks/scale;
if (distance > 3)
{
digitalWrite(LED1Pin, LOW);
digitalWrite(LED2Pin, LOW);
digitalWrite(LED3Pin, LOW);
digitalWrite(LED4Pin, LOW);
digitalWrite(LED5Pin, LOW);

noTone(SpeakerPin);
}
else if (distance >2 && distance<= 3)
{
digitalWrite(LED1Pin, HIGH);
digitalWrite(LED2Pin, LOW);
digitalWrite(LED3Pin, LOW);
digitalWrite(LED4Pin, LOW);
digitalWrite(LED5Pin, LOW);

noTone(SpeakerPin);
// tone(SpeakerPin,round(440*(1+2*3/3)));
}
else if (distance >1 && distance<= 2)
{
digitalWrite(LED1Pin, HIGH);

```

```

digitalWrite(LED2Pin, HIGH);
digitalWrite(LED3Pin, LOW);
digitalWrite(LED4Pin, LOW);
digitalWrite(LED5Pin, LOW);

noTone(SpeakerPin);
// tone(SpeakerPin,round(440*(1+2*2/3)));
}
else if (distance >0.5 && distance<= 1)
{
digitalWrite(LED1Pin, HIGH);
digitalWrite(LED2Pin, HIGH);
digitalWrite(LED3Pin, HIGH);
digitalWrite(LED4Pin, LOW);
digitalWrite(LED5Pin, LOW);

noTone(SpeakerPin);
// tone(SpeakerPin,round(440*(1+2*1/3)));
}
else if (distance >0.2 && distance<= 0.5)
{
digitalWrite(LED1Pin, HIGH);
digitalWrite(LED2Pin, HIGH);
digitalWrite(LED3Pin, HIGH);
digitalWrite(LED4Pin, HIGH);
digitalWrite(LED5Pin, LOW);

noTone(SpeakerPin);
// tone(SpeakerPin,round(440*(1+2*0.5/3)));
}
else if (distance<= 0.2)
{
digitalWrite(LED1Pin, HIGH);

```

```

digitalWrite(LED2Pin, HIGH);
digitalWrite(LED3Pin, HIGH);
digitalWrite(LED4Pin, HIGH);
digitalWrite(LED5Pin, HIGH);

tone(SpeakerPin,440);
//tone(SpeakerPin,round(440*(1+2*0.2/3)));
}
// tone(SpeakerPin,round(440*(1+2*distance/3)));
Serial.print("Distance: ");
Serial.print(distance);
Serial.print(" m\n");//переход на новую строку
//Serial.print("Pulse width is: ");
//Serial.print(time_mks);
//Serial.print(" mks\n");//переход на новую строку
Serial.print("freq ");
Serial.print(round(440*(1+2*distance/3)));
Serial.print(" mks\n");//переход на новую строку
delay(200);
}

```

2.3 Построение диаграммы направленности датчика

Проект Arduino загружен на микроконтроллер. В зону действия датчика подводился столбик на разных расстояниях в соответствии со схемой эксперимента.

Зарегистрированы расстояния от столбика до оси датчика, при котором загораются светодиоды.

Результаты записаны в Excel и затем обработаны в MATLAB для получения трехмерной области, показывающей зону, в которой датчик регистрирует наличие объектов.

```

% 3Д зона направленности
p.z = [0 250 500 750 1000 1250 1500 1750 2000];

```

```

p.x = [0 150 320 395 440 465 540 550 760];

z_fine = linspace(min(p.z), max(p.z), 50);
x_interp = spline(p.z, p.x, z_fine);

[Xcyl, Ycyl, Zcyl] = cylinder(x_interp, 50);
Zcyl = Zcyl * (max(p.z) - min(p.z)) + min(p.z);

figure;
hold on;
mesh(Zcyl, Xcyl, Ycyl, 'EdgeColor', 'k', 'LineWidth', 0.5, 'FaceAlpha', 0);
axis equal; xlabel('X'); ylabel('Y'); zlabel('Z');
pbaspect([1 1 1])
view(145, 20); lighting gouraud; camlight;
hold on
theta_orig = 0;
X_orig = p.x' .* cos(theta_orig);
Y_orig = p.x' .* sin(theta_orig);
Z_orig = repmat(p.z', length(theta_orig), 1)';
scatter3(
Z_orig(:), X_orig(:), Y_orig(:), 10,
'r', 'filled', 'MarkerEdgeColor', 'red');
grid minor

```

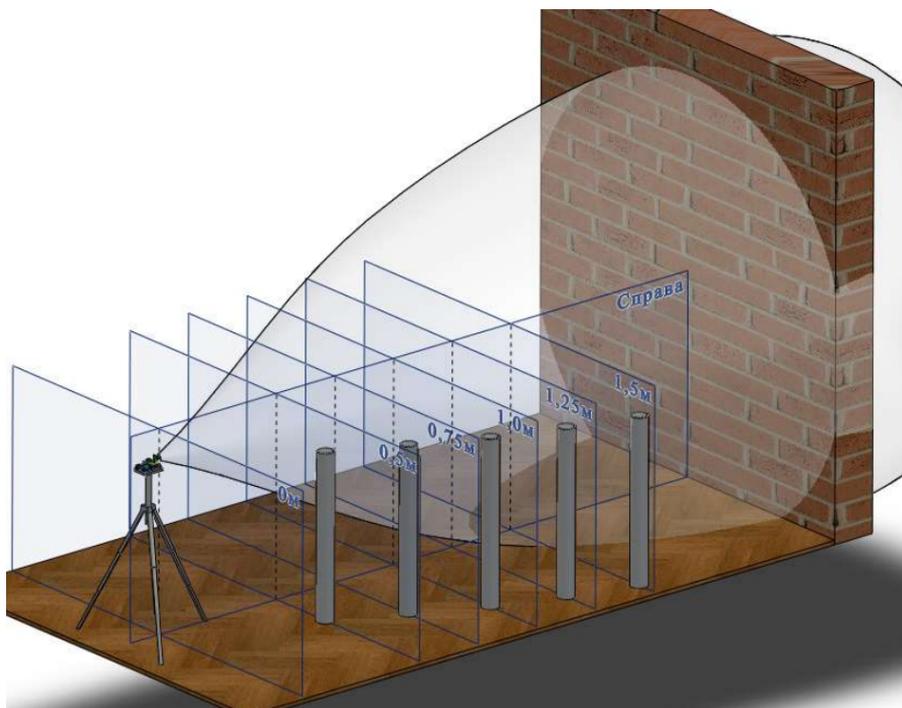


Рисунок 2.4 — Схема проведения эксперимента для диаграммы направленности

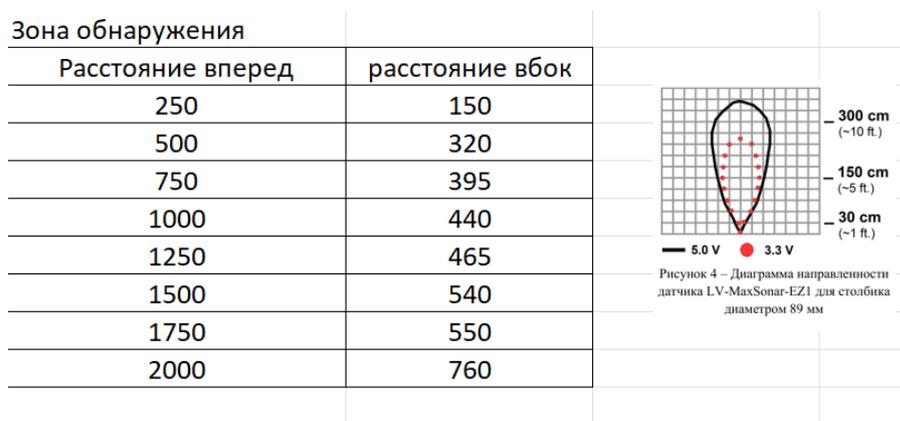


Рисунок 2.5 — Запись результатов эксперимента в Excel

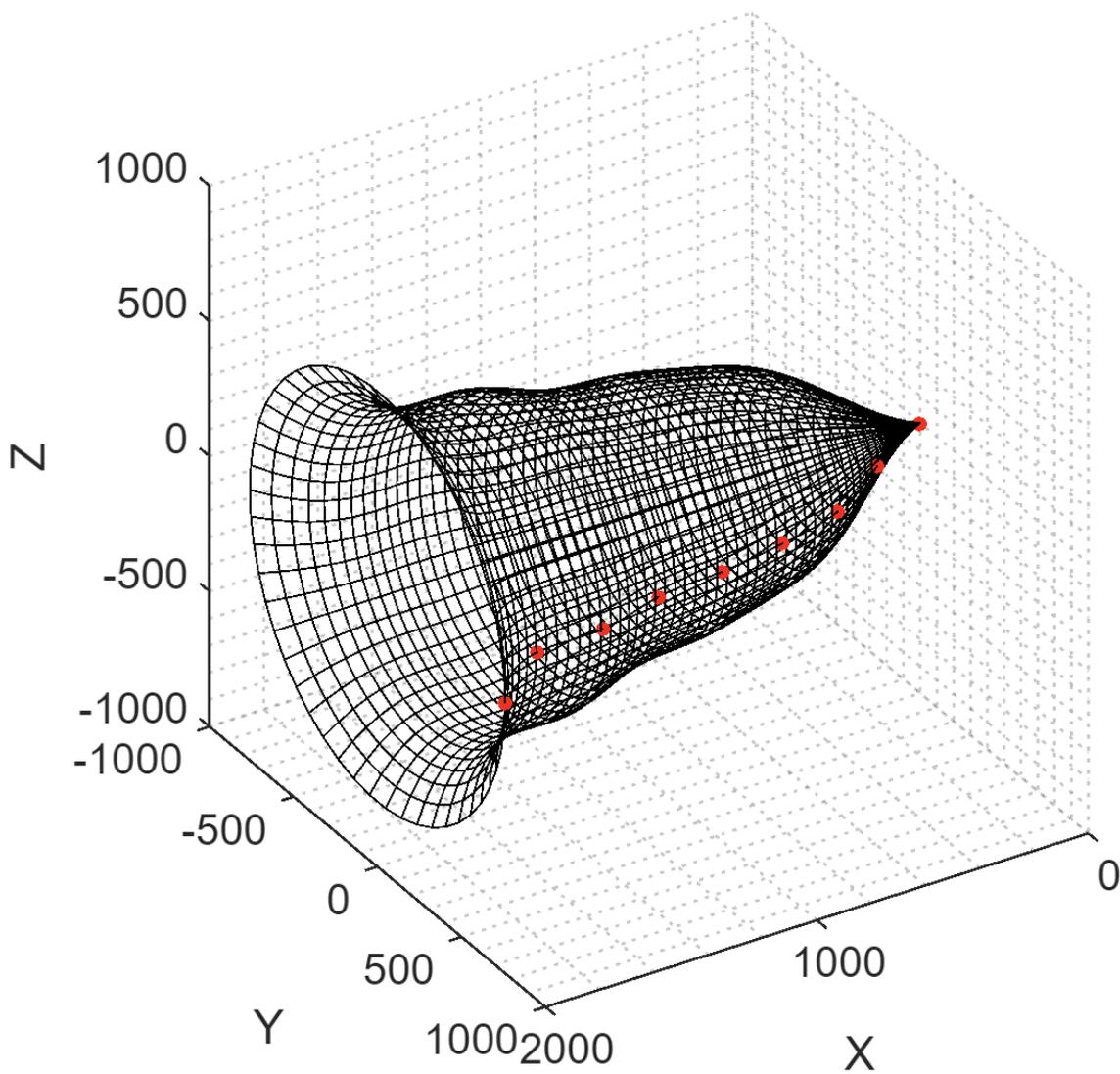


Рисунок 2.6 — Область срабатывания датчика

3 Вывод

В ходе выполнения лабораторной работы были изучен принцип работы ультразвукового датчика расстояния. Проведена тарировка датчика и разработана программа для микроконтроллера Arduino, обрабатывающая данные с датчика, зажигающая светодиоды и включающая динамик в зависимости от вычисленного расстояния до преграды.

Проведён эксперимент для определения диаграммы направленности датчика для препятствия в виде столбика. Результаты использованы для построения области, в которой датчик регистрирует объекты.