НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

ФАКУЛЬТЕТ ІНФОРМАТИКИ ТА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

КАФЕДРА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

**Звіт**

з дисципліни **«**Системи обробки сигналів**»**

тема: «Проект Arduino. Створення далекоміру»

Виконала

Студентка групи ІО-64

Невмержицька А. Ю.

Перевірив:

ст. вик. Виноградов Ю. М.

Київ – 2018

**Опис проекту**

Ультразвуковий далекомір виготовлений на базі платформи Arduino Nano.

Для визначення місця розташування використовується модуль ультразвукового далекоміра, показання з нього фільтруються двома фільтрами, завдяки чому досягнуто точність ~ 1 мм.

Ультразвуковий детектор HC-SR04 працює за принципом відлуння-локації випромінює звукову хвилю і приймає відбитий луна-сигнал. Час між відправленим і прийнятим сигналом можна перетворити в відстань.

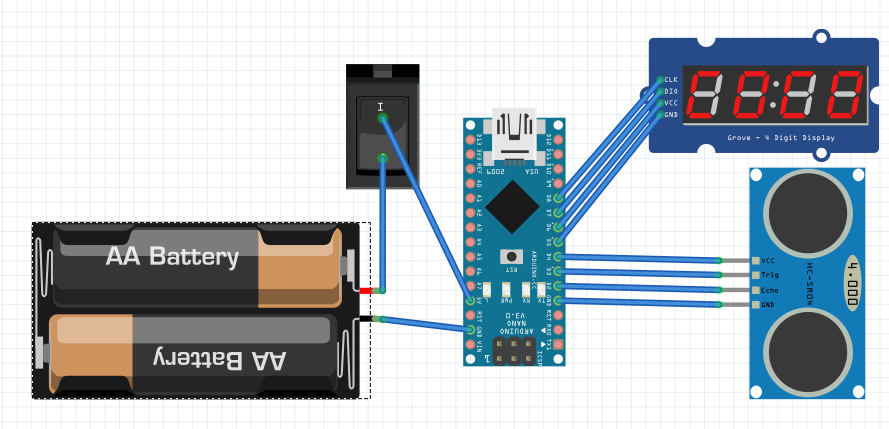
Для відображення відстані використовується дисплей на драйвері TM74HC59.

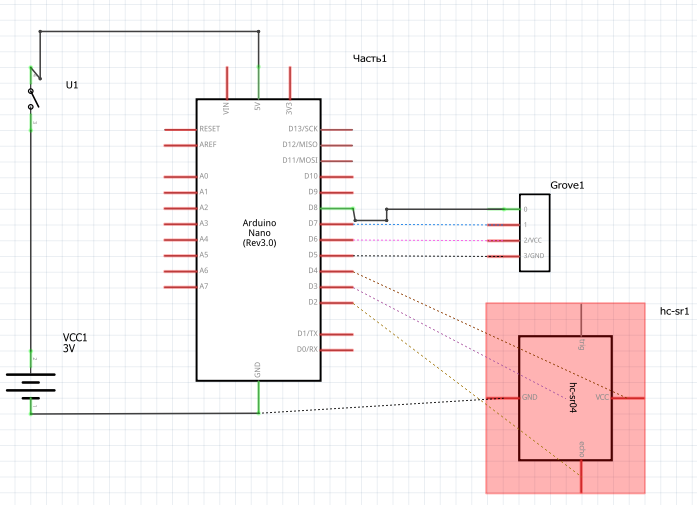
**Матеріали та компоненти**

* Arduino Nano
* Ультразвуковий далекомір HC-SR04
* Дисплей TM74HC59
* Батарейний відсік
* Пластиковий корпус
* Перемикач
* Провода
* Батарейки

Також знадобляться паяльник, дриль, ніж, надфілі, термоклей.

**Схема**





**Фото готового пристрою**



**Код програми**

#define ECHO 2

#define TRIG 3

#define sensVCC 4

#define dispGND 5

byte DIO = 6;

byte RCLK = 7;

byte SCLK = 8;

#define dispVCC 9

#include <TM74HC595Display.h>

#include <TimerOne.h>

TM74HC595Display disp(SCLK, RCLK, DIO);

unsigned char SYM[47];

#include <NewPing.h>

NewPing sonar(TRIG, ECHO, 400);

float dist\_3[3] = {0.0, 0.0, 0.0};

float middle, dist, dist\_filtered;

float k;

byte i, delta;

unsigned long dispIsrTimer, sensTimer;

void setup() {

Serial.begin(9600);

symbols(); // создать символы для отображения на дисплее

// настройка пинов

pinMode(sensVCC, OUTPUT);

pinMode(dispGND, OUTPUT);

pinMode(dispVCC, OUTPUT);

// подаём сигналы на пины

digitalWrite(sensVCC, 1);

digitalWrite(dispGND, 0);

digitalWrite(dispVCC, 1);

}

void loop() {

if (millis() - sensTimer > 50) {

if (i > 1) i = 0;

else i++;

dist\_3[i] = (float)sonar.ping() / 57.5; // получить расстояние в текущую ячейку массива

dist = middle\_of\_3(dist\_3[0], dist\_3[1], dist\_3[2]); // фильтровать медианным фильтром из 3ёх последних измерений

delta = abs(dist\_filtered - dist); // расчёт изменения с предыдущим

if (delta > 1) k = 0.7; // если большое - резкий коэффициент

else k = 0.1; // если маленькое - плавный коэффициент

dist\_filtered = dist \* k + dist\_filtered \* (1 - k); // фильтр "бегущее среднее"

disp.clear(); // очистить дисплей

disp.float\_dot(dist\_filtered, 1); // вывести

sensTimer = millis(); // сбросить таймер

}

if (micros() - dispIsrTimer > 300) {

disp.timerIsr();

dispIsrTimer = micros();

}

}

// медианный фильтр из 3ёх значений

float middle\_of\_3(float a, float b, float c) {

if ((a <= b) && (a <= c)) {

middle = (b <= c) ? b : c;

}

else {

if ((b <= a) && (b <= c)) {

middle = (a <= c) ? a : c;

}

else {

middle = (a <= b) ? a : b;

}

}

return middle;

}

// символы для дисплея

void symbols() {

SYM[0] = 0xC0; //0

SYM[1] = 0xF9; //1

SYM[2] = 0xA4; //2

SYM[3] = 0xB0; //3

SYM[4] = 0x99; //4

SYM[5] = 0x92; //5

SYM[6] = 0x82; //6

SYM[7] = 0xF8; //7

SYM[8] = 0x80; //8

SYM[9] = 0x90; //9

SYM[10] = 0b01000000; //0.

SYM[11] = 0b01111001; //1.

SYM[12] = 0b00100100; //2.

SYM[13] = 0b00110000; //3.

SYM[14] = 0b00011001; //4.

SYM[15] = 0b00010010; //5.

SYM[16] = 0b00000010; //6.

SYM[17] = 0b01111000; //7.

SYM[18] = 0b00000000; //8.

SYM[19] = 0b00010000; //9.

}