**Дніпровський ліцей інформаційних технологій**

**при Дніпропетровському національному університеті**

**імені Олеся Гончара**

**Випускна робота**

За темою:

**“****Роботизований автомобіль на базі управління ардуіно”**

**Виконавець:**

**ліцеїст ІІІ-Г класу**

**Котов Болеслав Вадимович Керівники роботи:**

**Чашка Ю. М. \_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Дніпро**

**2017**

**Зміст**

[Вступ 3](#_Toc499901775)

[Мета 3](#_Toc499901776)

[Завдання: 3](#_Toc499901777)

[Актуальність 3](#_Toc499901778)

[Основа частина 4](#_Toc499901779)

[Глава 1 4](#_Toc499901780)

[Arduino 4](#_Toc499901781)

[C++ 5](#_Toc499901782)

[Глава 2 6](#_Toc499901783)

[Опис проекту 6](#_Toc499901784)

[Принцип роботи 6](#_Toc499901785)

[План виконання проекту 6](#_Toc499901786)

[Хід виконання проекту 7](#_Toc499901787)

[Програмна частина 9](#_Toc499901788)

[Додаток 1 - Програма рукавички (передавача) 9](#_Toc499901789)

[Додаток 2 - Програма робоавто (одержувача) 11](#_Toc499901790)

[Висновок 15](#_Toc499901791)

[Програмно-апаратні вимоги 16](#_Toc499901792)

[Використані програмні засоби 16](#_Toc499901793)

[Список використаних джерел 16](#_Toc499901794)

# **Вступ**

Коли ми дивимося науково-фантастичні фільми з абсолютно новими, незнайомими для нас технологіями і пристроями, ми задаємося питанням про те, як це влаштовано, і що б ми зробили, якби у нас з'явився цей апарат. На основі одного з таких пристроїв я вирішив створити робо-рукавичку. А тепер можна поміркувати на тему: «де її можна застосувати в реальності?». У світі вже існують аналоги, які розрізняються чутливістю, вартістю, складністю виконання і, звичайно, завданнями, для яких вони розроблені. У моєму випадку, рукавичку можна використовувати для управління робо-автомобілем. Також існує безліч інших завдань, в яких застосування такого при строю помітно спростило б управління. Роборукавичку можна використовувати для управління дронами, гвинтокрилами, будівельними кранами, водним транспортом, робо-рукою і в багатьох-багатьох інших галузях.

Мета

Метою моєї роботи було створити робо-рукавичку для управління робо-автомобілем.

### **Завдання**:

1. Створення готової моделі рукавички
2. Складання робоавтомобіля
3. Написання програми
4. Тестування для виявлення несправностей

**Актуальність**

Необхідність спростити і полегшити управління робо-технічних систем.

# **Основа частина**

## **Глава 1**

### **Arduino**

Arduino - це одноплатний комп'ютер який має власний процесор і пам'ять. Для програмування використовується прощена версія С ++, відома так само як Wiring. Розробку можна вести як з використанням безкоштовної середовища Arduino IDE, так і за допомогою довільного C / C ++ інструментарію.

У проекті використовується Arduino UNO і Arduino nano.

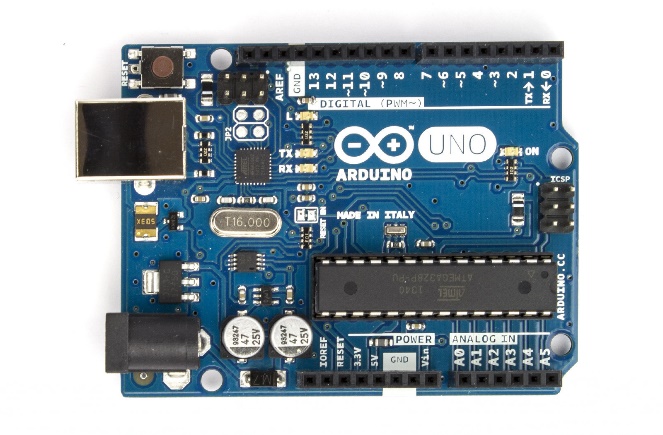


Рисунок 1 - Arduino Uno.

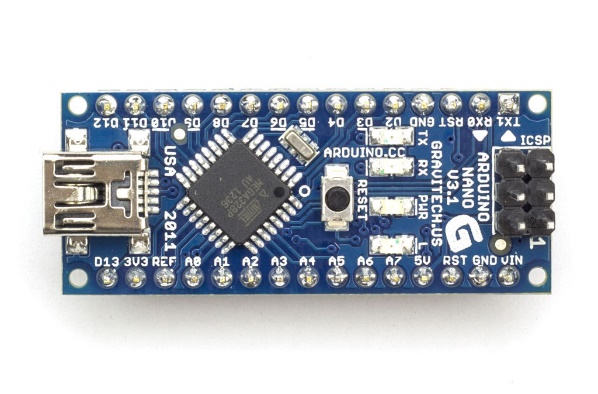


Рисунок 2 - Arduino nano.

Arduino Nano - це функціональний аналог Arduino UNO, але розміщений на мініатюрній платі. Рекомендується діапазон харчування: 7-12 В. UNO і nano виконані на базі процесора Atmega328p з тактовою частотою 16 МГц, що володіє пам'яттю 32 кБ і має 20 контрольованих контактів введення і виведення для взаємодії із зовнішнім світом.

### **C++**

C++ (Сі-плюс-плюс) — мова програмування високого рівня з підтримкою декількох парадигм програмування: об'єктно-орієнтованої, узагальненої та процедурної. Розроблена Б'ярном Страуструпом в AT&T Bell Laboratories (Мюррей-Хілл, Нью-Джерсі) 1979 року та початково отримала назву «Сі з класами». Згодом Страуструп перейменував мову на C++ у 1983 р. Базується на мові С. Вперше описана стандартом ISO/IEC 14882:1998, найбільш актуальним же є стандарт ISO/IEC 14882:2014

## **Глава 2**

Опис проекту

В основі робо-рукавички лежить датчик вигину. Конструкція цього датчика гранично проста. Такий датчик може зібрати будь-яка людина, що має навички в пайці, без будь-яких знань в області радіоелектроніки.

### **Принцип роботи**

Принцип роботи датчика схожий на принцип роботи оптопари. Датчик заснований на взаємодії фоторезистора і світлодіода. У протилежних кінцях трубки розташовані фоторезістр і світлодіод, коли трубка згинається, промені світла випромінюються світлодіодом і потрапляють на фоторезистор в меншому обсязі. Фоторезистр змінює свій опір, і як наслідок, змінюються вихідні дані на аналоговому порту. Всі отримані дані передатись через Wi-Fi передавач на робоавтомобіль. У робо-автомобілі отримані дані обробляються, і відповідно до значень відбувається регулювання швидкості обертання моторів.

План виконання проекту

1) Продумати конструкцію в загальних рисах

2) Зібрати всі необхідні матеріали

3) Спаяти пробний варіант

4) Написати пробну програму

5) Провести перевірку на коректність роботи всього пристроя

6) Усунути недоліки конструкції та програми

7) Багаторазове тестування для виявлення недоліків і їх усунення.

Хід виконання проекту

  У різні кінці гнуться трубки (через доступність і низьку ціну я взяв трубку від крапельниці) вставлений світлодіод і фоторезістер. Все з'єднане за схемою:

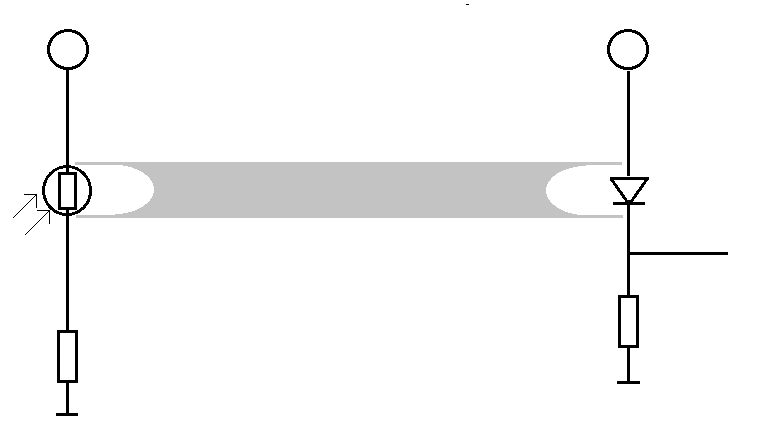


Рисунок 3 - схема трубки з світлодіодом та фоторезистором

Потім обмотуємо його термоусадкой для ізоляції від зайвого світла і спаюємо все. Вийшли ось такі датчики:



Рисунок 4 - датчик обмотаний термоусадкой

Після цього я зробив розводку для з'єднання з пинами Arduino.

В ході подальших перевірок на виявлення несправностей було виявлено, що дроти перетинаються, тим самими викликаючи некоректні значення. Для усунення цієї проблеми було вирішено залити клеєм проблемні ділянки.

Як вже було сказано раніше, при згинанні датчика знижується рівень сигналу. Якщо уявити значення при вигині у вигляді графіка, то він прийме подібний вид.

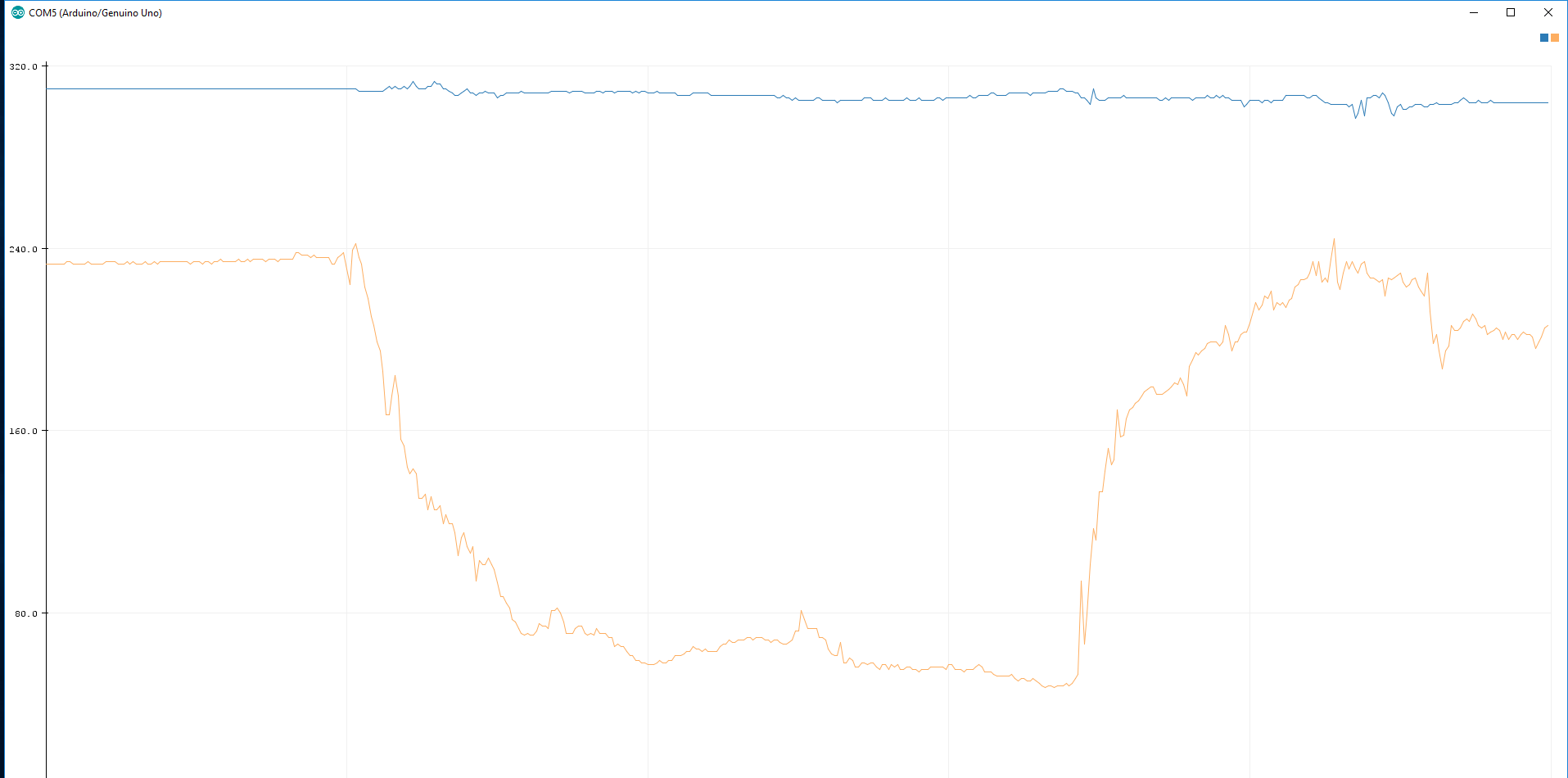


Рисунок 5 - датчик рівня сигналу

(Помаранчевий-вигнутий датчик; синій-випрямленний датчик)

Потім я взявся за вивчення Wi-Fi модуля Nrf24l01. Для спрощення роботи з датчиком я вибрав бібліотеку RF24 і nRF24l01. Щоб не псувати контакти модуля, я зібрав перехідник для з'єднання з пинами Arduino. Для згладжування радіо перешкод, між током і землею модуля, я, впаяв конденсатор наміналом 10 мкФ. Підключення модуля:

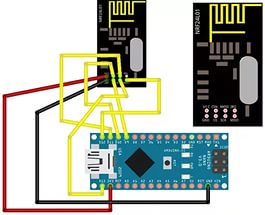


Рисунок 6 - підключення радіомодуля до arduino nano

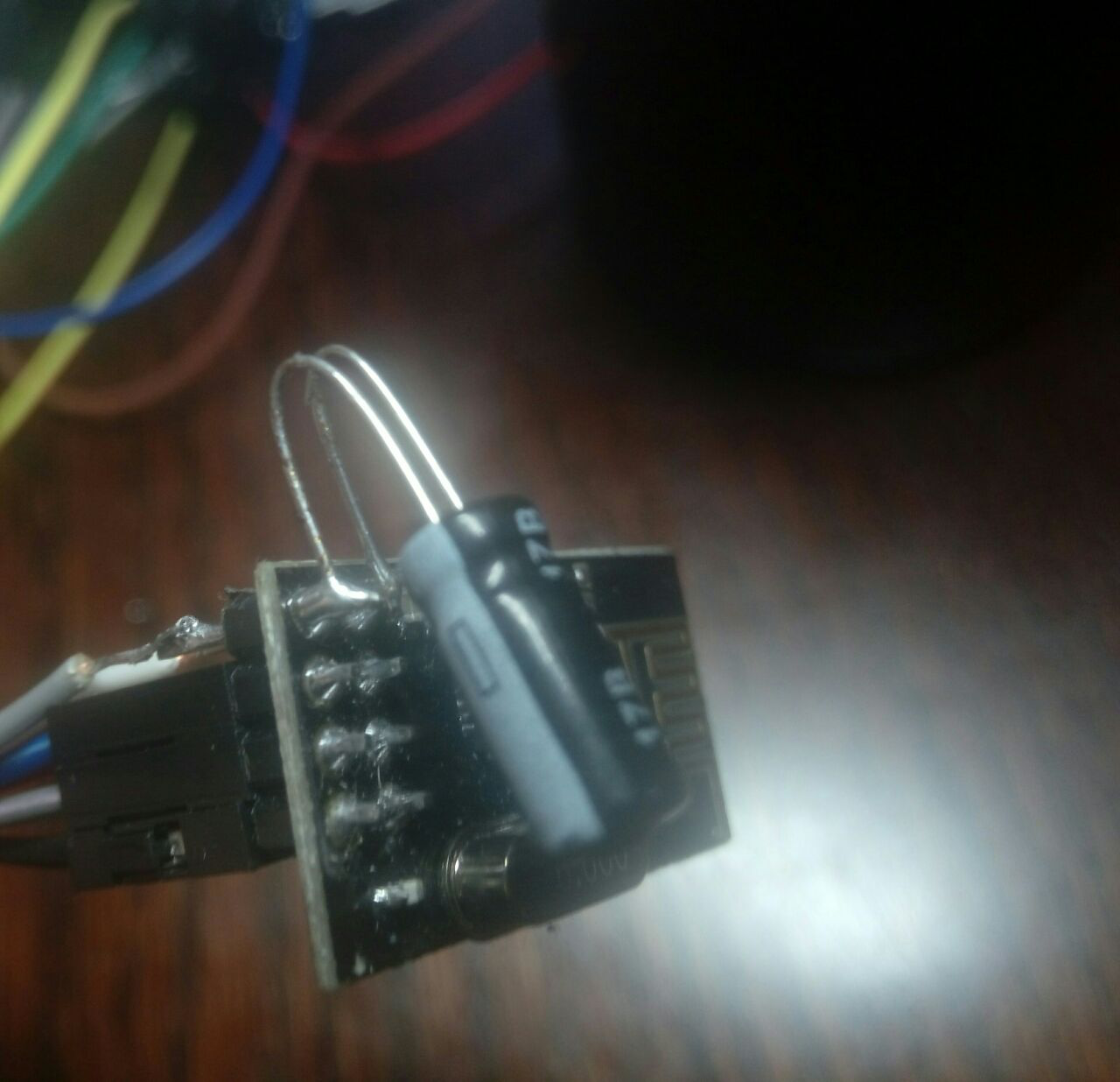


Рисунок 7 - припаяний конденсатор до радіомодуля

Аналлогічним чином я підключив Nrf24l01 до робо-авто.

Потім я перевірив все що вийшло на працездатність. Все запрацювало, але для подальшої настройки потрібно довести матеріальну систему до її кінцевого виду. Я, ушив датчики вигину в рукавичку і з'єднав всі компоненти. Машинка почала рухатися. Тепер необхідна програмна настройка.

# **Програмна частина**

Додаток 1 - Програма рукавички (передавача)

#include <SPI.h> // Підключаємо бібліотеку для роботи з шиною SPI

#include <nRF24L01.h> // Підключаємо файл настройок з бібліотеки RF24

#include <RF24.h> // Підключаємо бібліотеку для роботи з nRF24L01 +

int button = 8,button2=6;

RF24 radio(9, 10); // Створюємо об'єкт radio для роботи з бібліотекою RF24, вказуючи номери

int data[2]; // створюємо масив для прийому даних

void setup(){

radio.begin(); // Почнемо роботу nRF24L01 +

pinMode(button, OUTPUT);

pinMode(button2, OUTPUT);

radio.setChannel(5); // Вказуємо канал передачі даних (від 0 до 127), 5 - значить передача даних здійснюється на частоті 2,405 ГГц (на одному каналі може бути тільки 1 датчик і до 6 передавачів)

radio.setDataRate(RF24\_1MBPS); // Вказуємо швидкість передачі даних (RF24\_250KBPS, RF24\_1MBPS, RF24\_2MBPS), RF24\_1MBPS - 1 Мбіт / сек

radio.setPALevel(RF24\_PA\_HIGH); // Вказуємо потужність передавача (RF24\_PA\_MIN = -18dBm, RF24\_PA\_LOW = -12dBm, RF24\_PA\_HIGH = -6dBm, RF24\_PA\_MAX = 0dBm)

radio.openWritingPipe(0x1234567890LL); // Відкриваємо трубу з ідентифікатором 0x1234567890 для передачі даних (на одному каналі може бути відкрито до 6 різних труб, які повинні відрізнятися тільки останнім байтом ідентифікатора)

digitalWrite(button, HIGH);

digitalWrite(button2, HIGH);

}

void loop(){

data[0] = analogRead(A0); // перший датчик

data[0] = map(data[0], 0, 355, 255, 0);

//if (data[0] < 110)data[0] = 0; data[0] = data[0] \* 10 + 1;

data[1] = analogRead(A6); // другий датчик

data[1] = map(data[1], 0, 129, 255, 0);

//if (data[1] < 110)data[1] = 0; data[1] = data[1] \* 10 + 2;

radio.write(&data, sizeof(data));

Serial.print("data[0]=");

Serial.println(data[0]);

Serial.print("data[1]=");

Serial.println(data[1]);

}

Додаток 2 - Програма робоавто (одержувача)

#include <SPI.h> // Підключаємо бібліотеку для роботи з SPI

#include "nRF24L01.h" // Підключаємо файл настройок з бібліотеки RF24

#include "RF24.h" // Підключаємо бібліотеку для роботи з nRF24L01 +

RF24 radio(9, 10); // Створюємо об'єкт radio для роботи з бібліотекою RF24

int data[2];

int input = 0, ost;

int massiv[1];

// визначаємо піни управління правим мотором

int IN1 = 5;

int IN2 = 4;

// визначаємо піни управління лівим мотором

int IN3 = 7;

int IN4 = 8;

// визначаємо піни управління швидкістю правого мотора

int ENA = 6;

// визначаємо піни управління швидкістю лівого мотора

int ENB = 3;

void setup()

{

Serial.begin(9600);

delay(1000); // Почнемо роботу індикатора

radio.begin(); // Почнемо роботу nRF24L01 +

radio.setChannel(5); // Вказуємо канал прийому даних

radio.setDataRate(RF24\_1MBPS); // Вказуємо швидкість передачі даних

radio.setPALevel(RF24\_PA\_HIGH); // Вказуємо потужність передавача

radio.openReadingPipe(1, 0x1234567890LL); // Відкриваємо 1 трубу з ідентифікатором 0x1234567890 для прийому даних

radio.startListening();

pinMode(IN1, OUTPUT);

pinMode(IN2, OUTPUT);

pinMode(IN3, OUTPUT);

pinMode(IN4, OUTPUT);

pinMode(ENA, OUTPUT);

pinMode(ENB, OUTPUT);

}

void testA(int i) // управління швидкістю правого мотора

{

digitalWrite(IN2, HIGH);

digitalWrite(IN1, LOW);

analogWrite(ENA, i);

}

void testB(int i) // управління швидкістю лівого мотора

{

digitalWrite(IN4, HIGH);

digitalWrite(IN3, LOW);

analogWrite(ENB, i);

}

void loop()

{

if (radio.available()){

radio.read(massiv, sizeof(massiv));

//input = massiv[0];

//ost = input % 10;

//input = (input - ost) / 10;

//

//if (ost == 1) testA(input);

//else testB(input);

input = data[0];

ost = data[1];

if (input<0)

input \*= (-1);

if (ost<0)

ost \*= (-1);

testA(input);

testB(ost);

Serial.print("data[0]=");

Serial.println(data[0]);

Serial.print("data[1]=");

Serial.println(data[1]);

}

}

Програма вийшла невелика, але щоб підібрати оптимальні параметри різкості пішло чимало часу.

І так! Все працює!Далі я намагався усунути недоліки рукавички.

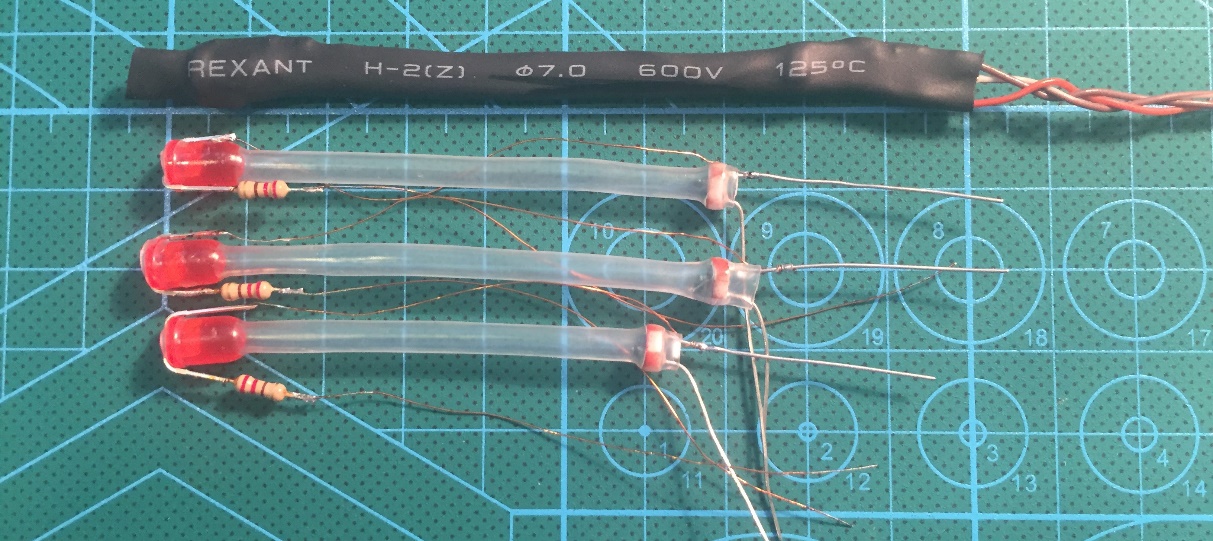
Я вирішив замінити трубку від крапельниці на прозору термоусадку й обмотати ізоляційною стрічкою.

Рисунок 8 - термоусадка

# **Висновок**

Проект закінчений успішно, всі завдання виконані. За час виконання проекту я дізнався багато нового. Можна бути впевненим в тому, що такі пристрої будуть широко застосовуватися в майбутньому. Управління машинкою таким способом-це тільки початок. Скільки ще можна придумати способів застосування? Воно може бути затребуване як в високоточних галузях, так і для простих завдань. Його можна застосувати, як для оброблення різного рівня складності, так і для веселощів. Також стало ясно, що для того щоб отримати найкращий результат, необхідно виконати заново одну і ту ж роботу кілька разів. Я буду продовжувати роботу над удосконаленням рукавички, так як це дуже перспективний проект і до того ж мені подобається розвивати свої вміння в області програмування і робототехніці.

# **Програмно-апаратні вимоги**

Для повноцінної роботи програми необхідно задовольнити наступні вимоги:

Системні вимоги:

* Arduino IDE;
* Windows 8.1 або пізніші версії.

Апаратні вимоги:

* Intel Pentium-4560 або вище;
* монітор з роздільною здатністю 1280 х 768;
* 800 MB вільного місця на жорсткому диску.

**Використані програмні засоби**

* Arduino IDE
* Adobe Photoshop
* Sublime text 3
* Paint
* Microsoft Word 2016
* Processing-3.3.5

**Список використаних джерел**

<https://ru.wikipedia.org>

<https://www.arduino.cc/>

<http://arduino.ua/>

<http://amperka.ru/>

<https://www.youtube.com/>

<http://arduino-diy.com/>

<http://cxem.net/>

<https://geektimes.ru/>

<https://habrahabr.ru/>

<https://github.com/>