

SHIFOXONADA XIZMAT KO'RSATISH UCHUN MO'LJALLANGAN AQILLI SHIFOKOR ROBOT

Jalolov Tursunbek Sadriddinovich

Osiyo Xalqaro Universiteti assistenti

Ramazonov Jahongir Abdurasul o'g'li

Osiyo Xalqaro Universiteti

Annotatsiya. *Shifokor robot loyihasi asosan yuqadigan kasalliklarni oldinini olish uchun ishlab chiqilgan. Shifoxonaning eshik qismidan kirib turgan insonlarni qo'lga dezinfeksiya qilib va temperaturasini ham o'lchab keyin o'tkazib yuboradi. Keyingi vazifasi bemor xonasiga kirib analiz olishi, bemorni kayfiyatini ko'tarish uchun suhbatlashish funksiyasi ham kiritilgan va robotni orqa qismiga joylashgan qurilma bilan shifokor bemorni davolashi uchun kerak bo'lgan asbob- uskunalarni va ovqat iste'mol qilish vaqtida oziq-ovqat, ichimlik bemorga keltirib beradi. Bu shifokor robot loyihasi masofadan telefon yordamida boshqariladi.*

Kalit so'zlar: *Shifokor robot, dezinfeksiya funksiyasi, temperaturasini o'lchash, robototexnika, arduino mikrokontrolleri.*

KIRISH

Shifokor robot loyihasi asosan kasalliklar insonlarga yuqishini oldinini olish uchun ishlab chiqilgan. Masalan: **covid-19** koronavirusi kasalligi ilk marotaba 2019-yilda Xitoyning Uxan shahrida aniqlandi va global miqyosda tarqalib, 2019–2020-yillardagi koronavirus pandemiyasini keltirib chiqardi. Kasallik yuqori harorat, yo'tal hamda nafas olishni mushkullashishi kabi simptomlarni keltirib chiqaradi. Oldinini olish uchun shifokorlar o'rniga bemorlardan odamsimon robotlar analiz olishi, temperaturasini o'lchashi va ko'plab shunga o'xshagan vazifalarni bajara oladi.

TADQIQOT MATERIALLARI VA METODOLOGIYASI

Tajriba o'tkazish uchun shifoxonaga bu loyiha olib boriladi. Robotimizga shifoxonaning barcha qismlarini o'lchamlarri kiritiladi. Chunki kiritilgan o'lchamlar bu robot uchun chegara bo'ladi. Shu chegarani ichki qismida robotimiz o'z vazifalarini bajaradi.

TADQIQOT NATIJALARI

Robotning kodi C++ dasturlash tilida yozilgan. Bu robot masofadan boshqariladi, aftomatlashtirilgan funksiyalari ham bu shifokor robotga kiritilgan. Ishlashi uchun Arduino, motorlarni boshqarilishi uchun drayver va masofaga ulab berish uchun NRF moduli ishlatilgan.



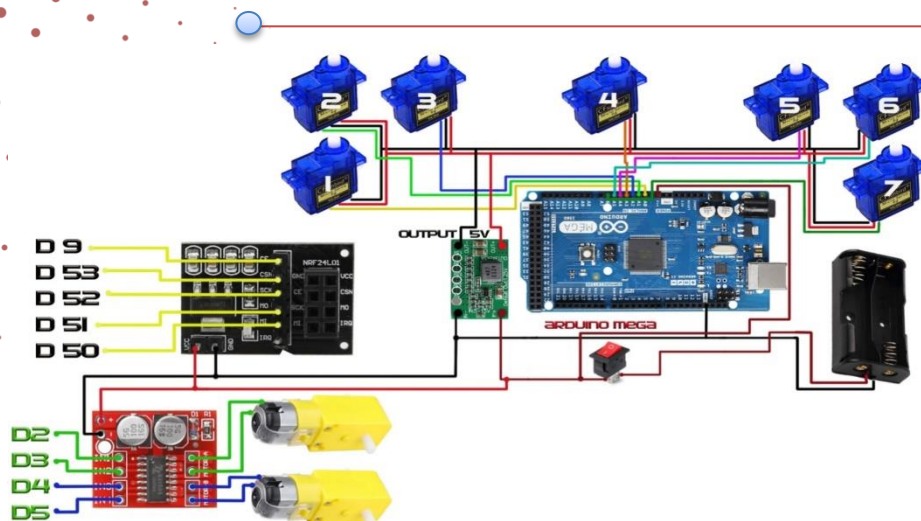
1-rasm. Shifokor robot loyihasining tayyor holati 1-versiya.

MUHOKAMA

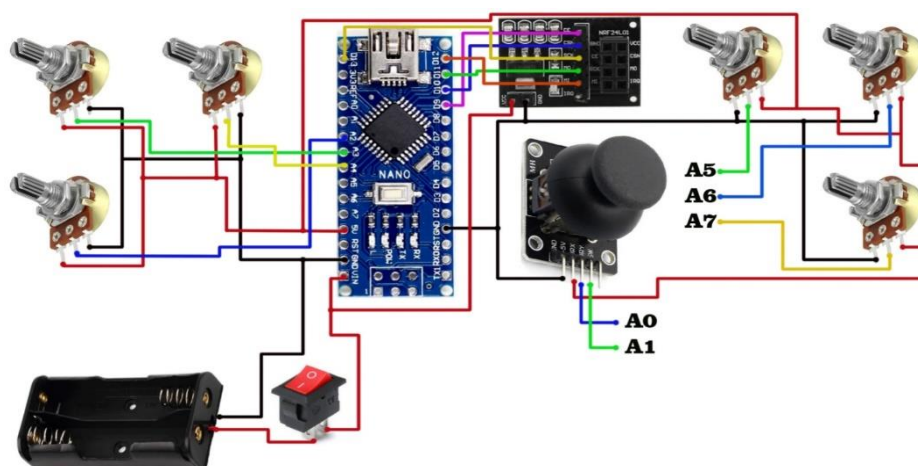
Elektron xisoblash mashinalari (EXM) yuqorida aytilganidek mantiqiy sxemalar va ular asosidagi mantiqiy elementlarni jamlash, turli kombinatsiyalar hosil qilish orqali yig'iladi.

Shulardan kelib chiqib, mantiqiy elementlarni talaba hamda o'quvchilarga o'qitish juda muhimdir. Bu mavzular faqatgina Elektronika, Radioelektronika va Sxematexnika fanlaridagina emas balki Axborot texnologiyalarini o'qitishda ham juda ahamiyatli sanaladi. Men mana shunday sabablardan kelib chiqib ushbu mavzuni hozirgi kunda jadal rivojlanib, turmush tarzimizga singib borayotgan Robototexnikaning asosi bo'lgan Arduino mikrokontrolleri orqali tushuntirish yanada qiziqarli hamda samarali deb o'ylayman. Quyida ushbu mavzuni Arduino platformasi yordamida tushunishga harakat qilamiz.

Arduino — professional bo'lmagan robototexnika ishqibozlari (foydalanuvchilar) uchun dasturiy va texnik taminotlar yig'indisi. Arduino mutloq ochiq platforma hisoblanadi. Undagi barcha loyihalar qismlari (Asosan dasturiy taminot) ochiq tarqatiladi. U o'zining platasi va IDE dasturlash muhitiga ega.



2-rasm. Shifokor vazifasini bajarishi uchun Arduino platasida yig'ilgan sxema



3-rasm. Robot masofadan boshqarilishi uchun pultini yig'ilgan sxema

XULOSA

Arduino robototexnika va elektronikiga qiziquvchi va izlanuvchi yoshlarga juda qo'l keladi. Biz o'z tajribalarimizda Arduino Uno turidan foydalandik. Uning boshqa turlaridan farqi protsessori, mikrokontrolleri, raqamli va analog chiqishlarning ko'p yoki kamligi bilan farqlanadi. Arduinodan foydalanayotgan kishi unga har xil elektr komponentalar va modullarni ulash imkoniyatiga ega bo'ladi, masalan: led chiroqlar, datchiklar, rele modullari tarmoq (Wi-fi, Bluetooth, Ethernet) modullari, sensorlar, motorlar, magnit eshik qulflari va elektr energiyasi bilan ishlaydigan barcha narsalar. Arduino texnik va dasturiy qismni birlashtirib beruvchi qurilma. Arduino uchun dasturlar odatiy C++ da

yoziyadi, kontaktlarda I / O (Input-kirish, Output-chiqish) ni boshqarish uchun oddiy va tushunarli algoritmlar va dasturlar tuziladi.

Endi mantiqiy amallarni Arduino platformasida tushuntirish uchun quyidagicha tajriba o'tkazamiz. 4-rasmda keltirilgan sxemani tuzamiz. Bu yerda 2 ta yashil LED chiroqlari jadvaldagi A va B kiruvchi signallarni anglatadi. Chiroq yonsa, (1) ni va

```

RJR_robot | Arduino 1.8.13
Файл Правка Скетч Инструменты Помощь
RJR_robot
#include <SoftwareSerial.h>
#include <DFPlayer_Mini_Mp3.h>
unsigned long T;
#include <SPI.h>
#include "nRF24L01.h"
#include "RF24.h"
#include <Servo.h>
//RF24 radio(9, 10);
RF24 radio(9, 53);

byte recieved_data[20];
byte motor1 = 2;
byte motor2 = 3;
byte motor3 = 4;
byte motor4 = 5;
byte servo1 = A0;
byte servo2 = A1;
byte servo3 = A2;
byte servo7 = A3;
byte servo4 = A4;
byte servo5 = A5;
byte servo6 = A6;
Servo myservo1;
Servo myservo2;
Servo myservo3;
Servo myservo4;
Servo myservo5;
Servo myservo6;
Servo myservo7;
byte address[][6] = {"1Node", "2Node", "3Node", "4Node", "5Node", "6Node"};

void loop() {
  if (recieved_data[4] < 5) {
    myservo7.write(170);
  } else if (recieved_data[9] > 175) {
    myservo7.write(10);
  } else {
    myservo7.write(90);
  }
  byte pipeNo;
  while ( radio.available(&pipeNo) ) {
    radio.read( &recieved_data, sizeof(recieved_data) );
    digitalWrite(motor1, recieved_data[0]);
    digitalWrite(motor2, recieved_data[1]);
    digitalWrite(motor3, recieved_data[2]);
    digitalWrite(motor4, recieved_data[3]);
    myservo1.write(recieved_data[4]);
    myservo2.write(recieved_data[5]);
    myservo3.write(recieved_data[6]);
    myservo4.write(recieved_data[7]);
    myservo5.write(recieved_data[8]);
    myservo6.write(recieved_data[9]);
  }
}

```

```

void setup() {
  Serial.begin(9600);
  pinMode(motor1, OUTPUT);
  pinMode(motor2, OUTPUT);
  pinMode(motor3, OUTPUT);
  pinMode(motor4, OUTPUT);

  myservo1.attach(servo1);
  myservo2.attach(servo2);
  myservo3.attach(servo3);
  myservo4.attach(servo4);
  myservo5.attach(servo5);
  myservo6.attach(servo6);
  myservo7.attach(servo7);
  myservo2.write(0);
  myservo5.write(180);
  myservo7.write(90);
  radio.begin();
  radio.setAutoAck(1);
  radio.setRetries(0, 15);
  radio.enableAckPayload();
  radio.setPayloadSize(32);
  radio.openReadingPipe(1, address[0]);
  radio.setChannel(0x65);
  radio.setPALevel(RF24_PA_LOW);
  radio.setDataRate(RF24_250KBPS);
  radio.powerUp();
  radio.startListening();
}

```

rasm. Robot funksiyalari kodi

Endi kompyuter yordamida ishni bajarish uchun zaruriy dastur tuziladi. 5-rasmda ajratib ko'rsatilgan "&" belgi "VA" mantiqiy amalini bajaradi. Dasturni yuklagach, 4-rasmda yig'ilgan sxema orqali 1-jadvaldagi amallarni ketma-ketlik asosida bajarib ko'rish mumkin. Bunda "YOKI" hamda "EMAS" va boshqa kombinatsiyadagi amallarni quyidagi 2-jadvalda keltirilgan belgilardan foydalanish mumkin.

Talaba va o'quvchilar har bir amal uchun dasturiy qismni o'zgartirib, Arduino mikrokontrolleriga yuklasa, 4-rasmda yig'ilgan sxema ishga tushadi. Shu bilan ular rostlik jadvalidagi qiymatlarni amalda ko'zlari bilan ko'rishadi. Bu bilan ularning bilimlari yanada mustahkamlanadi, fanga bo'lgan qiziqishi ortadi kompyuter bilan ishlash qobilyati rivojlanadi va boshqa ko'plab samaralarni berishi mumkin.

```
RJR_robot | Arduino 1.8.13
Файл Правка Скетч Инструменты Помощь

RJR_robot $
#include <SPI.h>
#include "nRF24L01.h"
#include "RF24.h"
RF24 radio(9, 10);
//RF24 radio(9, 53);
byte address[][6] = {"1Node", "2Node", "3Node", "4Node", "5Node", "6Node"};
byte potent0 = A0;
byte potent1 = A1;
byte potent2 = A2;
byte potent3 = A3;
byte potent4 = A4;
byte potent5 = A5;
byte potent6 = A6;
byte potent7 = A7;
byte transmit_data[20];
byte latest_data[20];
boolean flag;
byte flag1 = 0;
byte flag2 = 0;

void setup() {
  radio.begin();
  radio.setAutoAck(1);
  radio.setRetries(0, 15);
  radio.enableAckPayload();
  radio.setPayloadSize(32);
  radio.openWritingPipe(address[0]);
  radio.setChannel(0x65);
  radio.setPALevel(RF24_PA_LOW);
  radio.setDataRate(RF24_250KBPS);
  radio.powerUp();
  radio.stopListening();
}

void loop() {
  flag1 = map(analogRead(potent0), 0, 1023, 0, 4);
  flag2 = map(analogRead(potent1), 0, 1023, 0, 4);
  if (flag1 == 0) {
    transmit_data[0] = LOW;
    transmit_data[1] = HIGH;
    transmit_data[2] = HIGH;
    transmit_data[3] = LOW;
  } else if (flag1 == 4) {
    transmit_data[0] = HIGH;
    transmit_data[1] = LOW;
    transmit_data[2] = LOW;
    transmit_data[3] = HIGH;
  } else if (flag2 == 0) {
    transmit_data[0] = LOW;
    transmit_data[1] = HIGH;
    transmit_data[2] = LOW;
    transmit_data[3] = HIGH;
  } else if (flag2 == 4) {
    transmit_data[0] = HIGH;
    transmit_data[1] = LOW;
    transmit_data[2] = HIGH;
    transmit_data[3] = LOW;
  } else {
    transmit_data[0] = LOW;
    transmit_data[1] = LOW;
    transmit_data[2] = LOW;
    transmit_data[3] = LOW;
  }
}
```

```
transmit_data[4] = map(analogRead(potent2), 0, 1023, 0, 180);
transmit_data[5] = map(analogRead(potent3), 0, 1023, 0, 180);
transmit_data[6] = map(analogRead(potent4), 0, 1023, 0, 180);
transmit_data[7] = map(analogRead(potent5), 0, 1023, 0, 180);
transmit_data[8] = map(analogRead(potent6), 0, 1023, 0, 180);
transmit_data[9] = map(analogRead(potent7), 0, 1023, 0, 180);
for (int i = 0; i < 20; i++) {
  if (transmit_data[i] != latest_data[i]) {
    flag = 1;
    latest_data[i] = transmit_data[i];
  }
}
if (flag == 1) {
  radio.powerUp();
  radio.write(&transmit_data, sizeof(transmit_data));
}
}
```

5-rasm. Robot pultining kodi

REFERENCES:

1. Jalolov, T. S. (2024). PYTHONNING MATEMATIK KUTUBXONALARINI O'RGANISH: KENG QAMROVLI QO'LLANMA. *BIOLOGIYA VA KIMYO FANLARI ILMIY JURNALI*, 2(5), 71-77.
2. Jalolov, T. S. (2023). PARALLEL PROGRAMMING IN PYTHON. *TECHNICAL SCIENCE RESEARCH IN UZBEKISTAN*, 1(5), 178-183.
3. Jalolov, T. S. (2023). PYTHON DASTUR TILIDADA WEB-ILOVALAR ISHLAB CHIQUISH. *TECHNICAL SCIENCE RESEARCH IN UZBEKISTAN*, 1(5), 160-166.