

## IKKI G`ILDIRAKLI TRANSPORT ROBOTLARI HARAKATINI DASTURLASH

**Ravshonbek Sultanov**

Toshkent viloyati Chirchiq davlat pedagogika instituti, o'qituvchi

[ravshanbek-1992@mail.ru](mailto:ravshanbek-1992@mail.ru)

**Muxabbat Xalmetova**

Toshkent viloyati Chirchiq davlat pedagogika instituti, o'qituvchi

[m.xalmetova@cspi.uz](mailto:m.xalmetova@cspi.uz)

### ANNOTATSIYA

Maqolada robototexnik qurilmalarni Arduino UNO yordamida dasturlashning amaliy ko'nikmalarini hosil qilish usuli keltirilgan. Ikki g'ildirakli transport robotining harakat dasturlari keltirilgan.

**Kalit so'zlar:** Arduino, UNO, sketch, interfeys, robot, datchik, PID kutubxonasi, sensor, robototexnika, transport.

## PROGRAMMING OF TWO-WHEEL TRANSPORT ROBOTS MOVEMENT

**Ravshonbek Sultanov**

Chirchik State Pedagogical Institute, Tashkent region, teacher

[ravshanbek-1992@mail.ru](mailto:ravshanbek-1992@mail.ru)

**Mukhabbat Khalmetova**

Chirchik State Pedagogical Institute, Tashkent region, teacher

[m.xalmetova@cspi.uz](mailto:m.xalmetova@cspi.uz)

### ABSTRACT

The article presents a method for developing practical skills in programming robotic services by Arduino UNO. The action programs of the two-wheeled transport robot are given.

**Keywords:** Arduino, UNO, sketch, interface, robot, sensor, PID library, sensor, robotics, transport.

### KIRISH

Mamlakatimizda kompyuter texnologiyasini o'rganishga oid darslik va qo'llanmalar istagancha topiladi. Biroq robototexnika fanini o'rganish hali yetarli darajada yo'lga qo'yilmagan. Holbuki, robototexnikani o'zlashtirmasdan dunyo taraqqiyotiga qo'shilib, uning yutuqlariga erishib bo'lmaydi. Shu boisdan ham umumiy o'rta ta'lim hamda o'rta maxsus va oliy ta'lim muassasalarida robototexnika alohida dars sifatida o'qitilishi zamon talabiga aylanib ulgurdi.

**Keywords:** Arduino, UNO, sketch, interface, robot, sensor, PID library, sensor, robotics, transport.

## ADABIYOTLAR TAHLILI VA METODOLOGIYA

Robototexnik qurilmalarni yig'ish bilan birga ularni dasturlash masalasi ham ancha murakkab jarayon hisoblanadi. Odatda robototexnik qurilmalarga dasturlar Arduino platasi yordamida yoziladi.

Arduino - bu o'ziga xos protsessor va xotiraga ega bo'lgan kichik plata bo'lib, professional bo'lmagan foydalanuvchilarga mo'ljallangan oddiy avtomatlashtirish va robototexnika tizimlarini yaratish uchun mo'ljallangan apparat va dasturiy ta'minot brendidir.

Dasturiy ta'minot qismi dasturlarni yozish, kompilyatsiya qilish va dasturiy ta'minot uchun bepul dasturiy qobiqdan iborat. Uskuna qismi rasmiy ishlab chiqaruvchilar tomonidan sotiladigan oldindan yig'ilgan bosma elektron platalar to'plamidir. To'liq ochiq tizim arxitekturasi sizga Arduino mahsulotlar qatoriga bema'lol nusxa ko'chirish yoki qo'shish imkoniyatini beradi.

Arduinodan avtomatlashtirilgan ob'ektlarni yaratish yoki standart simli va simsiz interfeyslar orqali kompyuterda dasturiy ta'minotga ulanish uchun foydalanish mumkin.

Robototexnik qurilmalarga quyidagi mikrokontrollerlarni misol qilishimiz mumkin: Arduino platalari, Father simlari, Breadboard taxtasi, PIR datchiklar, Svetodiodlar, Fotodiodlar, Sensor datchiklar va boshqalar. Arduino IDE ga yozilgan kodlar sketch deb ataladi. Arduinoga sketch yozish uchun uni kompyuterga Arduino USB orqali bog'lab olish kerak. Sketchlar orqali biz Arduino platalariga dasturlar yozib qurilmalar vazifalarini kiritishimiz mumkin.

## NATIJALAR

Arduino platalariga dastur yozishni "ikki g'ildirakli robot harakatini boshqarish" jarayonida ko'rib chiqamiz.

Dastlab biz ushbu dasturning ishlashi uchun zarur bo'lgan kutubxonalarni kiritamiz. Ular tarkibiga I2C kutubxonasi, PID kutubxonasi va MPU6050 kutubxonasi kiradi.

```
#include "I2Cdev.h"
#include <PID_v1.h> //From https://github.com/br3ttb/Arduino-PID Library/blob/master/PID_v1.h
#include "MPU6050_6Axis_MotionApps20.h" //https://github.com/jrowberg/i2cdevlib/tree/master/Arduino/MPU6050
```

Keyin MPU6050 sensoridan ma'lumotlarni olish uchun zarur bo'lgan

o'zgaruvchilarni e'lon qilamiz. Biz tortishish vektori va kvaternion qiymatlarini o'qiymiz, so'ngra botning yaw balandligi va rulon qiymatini hisoblaymiz. Ypr [3] suzuvchi massivi yakuniy natijaga ega bo'ladi.

```
// MPU control/status vars
bool dmpReady = false; // set true if DMP init was successful
uint8_t mpuIntStatus; // holds actual interrupt status byte from MPU
uint8_t devStatus; // return status after each device operation (0 =
success, !0 = error)
uint16_t packetSize; // expected DMP packet size (default is 42 bytes)
uint16_t fifoCount; // count of all bytes currently in FIFO
uint8_t fifoBuffer[64]; // FIFO storage buffer
// orientation/motion vars
Quaternion q; // [w, x, y, z] quaternion container
VectorFloat gravity; // [x, y, z] gravity vector
float ypr[3]; // [yaw, pitch, roll] yaw/pitch/roll container and gravi
ty vector
```

Keyinchalik kodning juda muhim segmenti keladi va bu erda siz to'g'ri qiymatlar to'plamini sozlash uchun uzoq vaqt sarflaysiz. Agar siz robot juda yaxshi tortishish markazi bilan qurgan bo'lsangiz va uning qismlari nosimmetrik tarzda joylashtirilgan bo'lsa (aksariyat hollarda bunday emas), u holda sizning belgilangan nuqtangiz qiymati 180 ga teng bo'ladi. Boshqa holatda sizning botingizni Arduino seriyali monitoriga ulang va uni yaxshi muvozanat holatini topasiz, ketma-ket monitorda ko'rsatilgan qiymatni o'qing va bu sizning belgilangan qiymatingiz bo'ladi. Kp, Kd va Ki qiymatlari sizning botingiz bo'yicha sozlanishi kerak. Kp, Kd va Ki qiymatlari bir xil ikkita botga teng bo'lmaydi, shuning uchun undan qochib bo'lmaydi.

```
/******Tune these 4 values for your BOT******/
double setpoint= 176; //set the value when the bot is perpendicular to gr
ound using serial monitor.
//Read the project documentation on circuitdigest.com to learn how to set
these values
double Kp = 21; //Set this first
double Kd = 0.8; //Set this second
double Ki = 140; //Finally set this
/******End of values setting******/
```

Keyingi satrda biz kirish, chiqish, sozlash nuqtasi, Kp, Ki va Kd o'zgaruvchilarini kiritish orqali PID algoritmini ishga tushiramiz. Ulardan yuqoridagi

kod parchasida Kp, Ki va Kd set-point qiymatlarini o'rnatamiz. Kirish qiymati MPU6050 sensoridan o'qiladigan yawning joriy qiymati bo'ladi va chiqish qiymati PID algoritmi bilan hisoblangan qiymat bo'ladi. Shunday qilib, asosan, PID algoritmi bizga kirish qiymatini belgilangan nuqtaga yaqinligini to'g'irlash uchun ishlatilishi kerak bo'lgan chiqish qiymatini beradi.

```
PID pid(&input, &output, &setpoint, Kp, Ki, Kd, DIRECT);
```

Void o'rnatish funksiyasi ichida biz MPU6050-ni DMP (Digital Motion Processor) ni sozlash orqali ishga tushiramiz. Bu bizga akselerometr ma'lumotlarini gyroskop ma'lumotlari bilan birlashtirishda yordam beradi va Yaw, Pitch va Rollning ishonchli qiymatini beradi. Qanday bo'lmasin, sozlash funksiyasida qidirishingiz kerak bo'lgan bitta kod segmenti - bu gyro ofset qiymatlari. Har bir MPU6050 datchigi o'zlarining ofset qiymatlariga ega, siz ushbu Arduino eskizidan o'zingizning datchikingizning ofset qiymatini hisoblashingiz va quyidagi satrlarni dasturingizda yangilashingiz mumkin.

```
// supply your own gyro offsets here, scaled for min sensitivity
mpu.setXGyroOffset(220);
mpu.setYGyroOffset(76);
mpu.setZGyroOffset(-85);
mpu.setZAccelOffset(1688);
```

Shuningdek, biz motorlarimizni ulash uchun foydalanadigan Raqamli PWM pinlarini ishga tushirishimiz kerak. Bizning holatda bu D6, D9, D10 va D11 lar hisoblanadi. Shunday qilib, biz ushbu pinlarni ishga tushiramiz, chunki chiqish pinlari ularni sukut bo'yicha past qiladi.

```
//Initialise the Motor output pins
pinMode (6, OUTPUT);
pinMode (9, OUTPUT);
pinMode (10, OUTPUT);
pinMode (11, OUTPUT);

//By default turn off both the motors
analogWrite(6,LOW);
analogWrite(9,LOW);
analogWrite(10,LOW);
analogWrite(11,LOW);
```

Asosiy sikl funksiyasi ichida biz MPU6050 ma'lumotlarini o'qishga tayyor yoki yo'qligini tekshiramiz. Agar ha bo'lsa, biz uni PID qiymatini hisoblash uchun

ishlatamiz va keyin PIDning qanday javob berishini tekshirish uchun ketma-ket monitorda PID kirish va chiqish qiymatini namoyish etamiz. So'ngra chiqish qiymatiga qarab bot oldinga yoki orqaga qarab harakatlanishi yoki bir joyda turishi kerakligiga qaror qilamiz.

### MUHOKAMA

Biz MPU6050 bot tik turganida 180 ni qaytaradi deb o'ylaymiz. Bot old tomonga tushganda biz tuzatish qiymatlarini ijobiy qabul qilamiz, agar bot orqaga tushayotgan bo'lsa, biz salbiy qiymatlarni olamiz. Shuning uchun biz ushbu holatni tekshiramiz va botni oldinga yoki orqaga qaytarish uchun tegishli funksiyalarni chaqiramiz.

```
while (!mpuInterrupt && fifoCount < packetSize)
{
    //no mpu data - performing PID calculations and output to motors
    pid.Compute();
    //Print the value of Input and Output on serial monitor to check how it is
    working.
    Serial.print(input); Serial.print(" =>"); Serial.println(output);
    if (input>150 && input<200){//If the Bot is falling
        if (output>0) //Falling towards front
        Forward(); //Rotate the wheels forward
        else if (output<0) //Falling towards back
        Reverse(); //Rotate the wheels backward
    }
    else //If Bot not falling
    Stop(); //Hold the wheels still
}
```

PID chiqishi o'zgaruvchisi, shuningdek, dvigatelni qanchalik tez aylantirish kerakligini hal qiladi. Agar bot yiqilib tushmoqchi bo'lsa, biz g'ildirakni asta-sekin aylantirib, kichik tuzatishlarni amalga oshiramiz. Agar ushbu kichik tuzatish ishlasa va bot tushib qolsa, biz motorning tezligini oshiramiz. G'ildiraklarning qanchalik tez aylanishini PI algoritmi hal qiladi. Teskari funksiya uchun biz salbiy qiymatni musbatga aylantirishimiz uchun mahsulot qiymatini -1 ga ko'paytirganimizga e'tibor bering.

```
void Forward() //Code to rotate the wheel forward
{
    analogWrite(6,output);
    analogWrite(9,0);
    analogWrite(10,output);
    analogWrite(11,0);
}
```

```
    Serial.print("F"); //Debugging information
}
void Reverse() //Code to rotate the wheel Backward
{
    analogWrite(6,0);
    analogWrite(9,output*-1);
    analogWrite(10,0);
    analogWrite(11,output*-1);
    Serial.print("R");
}
void Stop() //Code to stop both the wheels
{
    analogWrite(6,0);
    analogWrite(9,0);
    analogWrite(10,0);
    analogWrite(11,0);
    Serial.print("S");
}
```

## XULOSA

Harakatni boshqarish robotlari kelajakning ko'zga ko'ringan suniy intellektlaridan biri sanaladi. Shu asosida o'zini o'zi muvozanatlaydigan bir qancha loyihalar yaratilgan. Bu esa robototexnikaning shiddat bilan rivojlanayotganiga bir misoldir. Robototexnika – har qanday bola uchun eng yangi va istiqbolli mashg'ulotlaridan biri hisoblanadi. Ushbu kursda o'quvchilarga robotlarni va boshqa obyektlarni qurish va yaratish to'g'risidagi bilimlar beriladi. Bu Internet-texnologiyalari yo'nalishidagi sohadir.

Shuni qayd etish lozimki, robototexnika fanini ta'lim tizimiga izchillik bilan joriy etish kelajakda yoshlarning zamon talablariga mos fikrlash doirasining shakllanishiga zamin yaratadi. Qolaversa, nafaqat robototexnika sohasi, balki iqtisodiyotning boshqa turli tizimlari rivojlanishiga ham kuchli turtki beradi.

## REFERENCES

1. Жўраева, Н. В., Султанов, Р. О., Абдуллаева, С. А., Рахимжонов, В. А. (2020). Systematization of word combinations in the uzbek language. Наука и Мир, 2(6), 65-68.
2. Sultanov R. O., Yusupov M. R. (2020). Ta'limda matematika fanini o'qitishdagi muammolar va ularning yechimida axborot kommunikatsiya texnologiyalarining ahamiyati. O`zMU xabarlari, 2(1/2/1), 144-147.

3. Султанов, Р. О. (2020). Idea блокчи шифрлаш алгоритмини такомиллаштириш методлари. *Academic Research in Educational Sciences*, 1(3), 397-404.
4. Kamolov, E. R., Raximov, S. M., Sultanov, R. O., Maxmudov, M.A., (2021). Innovative method of developing creative thinking of students. *Экономика и социум*, 1(80).
5. Хуррамов, А. Ж., Комолов, Э. Р., Разработка алгоритма управления с учетом трудноформализуемой информации // *Academic research in educational sciences*, (2020). Volume 01, Issue 03, -pp: 240-247.
6. Gulbaev, N. A., Xalmetova, M. X., Sobirova, S. R. (2020). Building models of territorial distributed systems. *International Journal for Innovative Engineering and Management Research*, Volume 09, Issue 10, Pages: 194-198.
7. Ахмедов, Б. А. (2021). Задачи обеспечения надежности кластерных систем в непрерывной образовательной среде. *Eurasian Education Science and Innovation Journal*, 1(22), 15-19.
8. Akhmedov, B. A., Xalmetova, M. X., Rahmonova, G. S., Khasanova, S. Kh. (2020). Cluster method for the development of creative thinking of students of higher educational institutions. *Экономика и социум*, 12(79), 588-591.
9. Akhmedov, B. A., Makhkamova, M. U., Aydarov, E. B., Rizayev, O. B. (2020). Trends in the use of the pedagogical cluster to improve the quality of information technology lessons. *Экономика и социум*, 12(79), 802-804.