

FIND3R 2.0

Opiekun: Grzegorz Nowik

Uczestnicy: Konrad Oborski, Maciej Bachurzewski

Szkoła: Liceum Ogólnokształcące Politechniki Białostockiej



FIND3R 2.0 to robot wykorzystywany do poszukiwań osób po tąpnięciach w kopalniach, bądź zawaleniach zabudowań. Robot może być wykorzystany również w sytuacjach możliwego zawalenia się stropu lub całego budynku, po to aby nie narażać ludzkiego życia. Jego kształt i budowa pozwala na pokonywanie wszelkiego rodzaju nierówności, takich jak stopnie schodów wzniesienia czy nierówne tereny, ponieważ może regulować wysokość przedniego i tylnego modułu. Moduły połączone są 2 serwomechanizmami, jeden z przodu długi z tyłu, podnoszącymi wybraną część po łuku do góry, analogicznie opuszczając do dołu. Cała konstrukcja składa się właśnie z 3 elementów. Po podniesieniu przedniej oraz tylnej części robota możliwy jest obrót o 360 stopni w miejscu, co daje robotowi większą mobilność. Urządzenie dzięki swoim niewielkim rozmiarom może wjechać w ciasne miejsca, gdzie człowiek lub zwierzę może nie być w stanie dotrzeć, co może być przyczyną złej diagnozy podczas poszukiwania osób na miejscu katastrofy. Dzięki podglądowi z małej kamery, która dzięki swoim rozmiarom nie doda znacznej wagi i sterowaniu bezprzewodowemu za pomocą zmodyfikowanego pada, ekipa ratunkowa może przez cały czas mieć kontakt z urządzeniem i na bieżąco oceniać stan sytuacji. Urządzenie porusza się na 6 gąsienicach dzięki czemu jest odporniejszy na problemy takie jak przebicie opony lub zablokowanie się jednego z kół.

Projekt został wymyślony po zakończeniu budowy pierwszego robota. Dzięki czemu wiedzieliśmy co robić, w jakiej kolejności i na co zwracać szczególną uwagę przy projektowaniu, konstruowaniu i programowaniu. Dzięki naszym umiejętnościom wydobyliśmy z robota najbardziej przydatne funkcje, ograniczyliśmy margines błędu. W trwających dalej testach staramy wyeliminować wszystkie nieprawidłowości programu, przez co robot staje się coraz lepszy.

Produkt został stworzony z myślą pomocy grupom ratowniczym, aby ułatwić im pracę i nie narażać ich na niepotrzebne zagrożenie. Nasz produkt dotrze we wszystkie trudno dostępne, bądź całkowicie niedostępne człowiekowi miejsca, co może spowodować zwiększenie ilości uratowanych osób. Robot może być również wykorzystywany w miejscach niedostępnych z powodu różnych niebezpiecznych substancji chemicznych czy biologicznych. Życie człowieka jest najważniejsze, a nasz robot ratuje je oraz chroni.

Nasz robot jest sterowany specjalnie zmodyfikowanym kontrolerem, przez co jest dokładny i możemy nim sterować ze znacznej odległości. Dodatkowo

możemy podglądać pracę robota na wyświetlaczu, dzięki kamerze zamontowanej na obudowie FIND3RA.

Podobne produkty są drogie i ciężko dostępne, przez co nie każda grupa ratownicza może z nich korzystać. Naszym założeniem jest stworzenie czegoś nowego, innowacyjnego, co byłoby o wiele tańsze, a nie traciłoby na jakości. Taki cel przyświecał nam od samego początku prac.

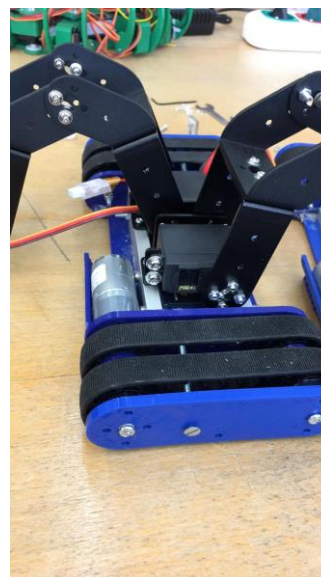
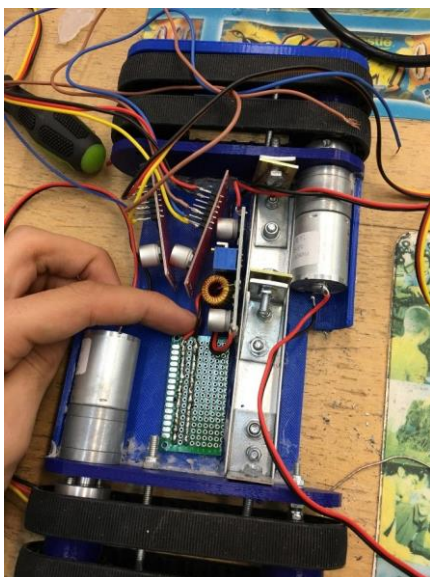
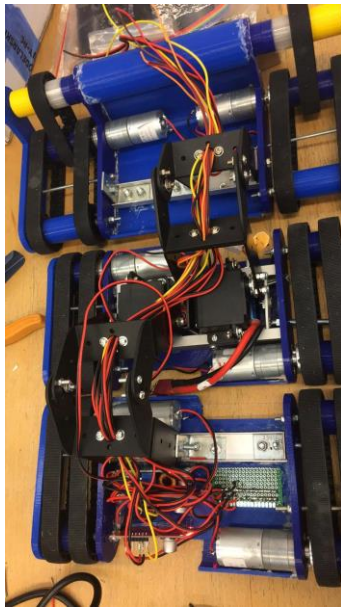
Zastosowany materiał musi być lekki, aby robot mógł poruszać się nawet po grząskich terenach, a także wytrzymały, aby w trakcie prac maszyna nie uległa zniszczeniu. Do budowy wykorzystaliśmy technologię druku 3D, dzięki czemu mogliśmy uzyskać odpowiednie kształty, a także jest to o wiele tańszy sposób.

Robot jest napędzany energią elektryczną, przez co nie zatrać środowiska spalinami i innymi szkodliwymi substancjami. Użytkownik projektu nie musi martwić się o niepożądane działanie robota. Zainstalowaliśmy liczne czujniki i zabezpieczenia przeciwawaryjne.

Własną pracę uważamy za bardzo dobrze i skrupulatnie wykonaną. Mimo, że spotykaliśmy się wyłącznie raz w tygodniu, na zajęciach robotycznych w naszej szkole, dawaliśmy z siebie wszystko i wykorzystywaliśmy cały możliwy czas. Utrudnieniem było także to, że każdy z nas pochodzi z innego miasta, więc często nie mogliśmy się zgrać, aby wszyscy byli na zajęciach. Wadami produktu jest na pewno druk 3D, który nie jest tak trwały, jak np. blacha, nieperfekcyjne wykonanie, spowodowane presją czasu, opóźnienia w pracach, problemy na drodze łączności. Zalet produktu jest mnóstwo, najważniejszymi są tania produkcja, łatwość wykonania, innowacyjność oraz prostota obsługi.

Nad projektem pracowaliśmy kilka miesięcy. Przez pierwsze miesiące skupiliśmy się na projektowaniu i drukowaniu części i obudowy, przez kolejne składaliśmy robota, lutowaliśmy przewody, organizowaliśmy miejsce w robocie, kupowaliśmy potrzebne produkty, a także modyfikowaliśmy kontroler. Robot wciąż potrzebuje kilku poprawek jednak przez panującą sytuację nie mamy możliwości dalszej pracy.

Zdjęcia wykonane przez autorów:



Kod robota:

```
void setup() {  
  pinMode(3,OUTPUT);  
  pinMode(5,OUTPUT);  
  pinMode(2,OUTPUT);  
  pinMode(9,OUTPUT);  
  pinMode(6,OUTPUT);  
  pinMode(8,OUTPUT);  
}
```

```
void loop() {  
  lewo();  
}
```

```
void tyl(){  
  analogWrite(3,200);  
  digitalWrite(32,LOW);  
  digitalWrite(42,HIGH);  
  analogWrite(7,200);  
  digitalWrite(36,HIGH);  
  digitalWrite(38,LOW);  
}
```

```
void przod(){  
  analogWrite(3,200);  
  digitalWrite(32,HIGH);
```

```
digitalWrite(42,LOW);  
analogWrite(7,200);  
digitalWrite(36,LOW);  
digitalWrite(38,HIGH);  
}
```

```
void pravo(){  
    analogWrite(3,200);  
    digitalWrite(32,HIGH);  
    digitalWrite(42,LOW);  
    analogWrite(7,200);  
    digitalWrite(36,HIGH);  
    digitalWrite(38,LOW);  
}
```

```
void lewo(){  
    analogWrite(3,200);  
    digitalWrite(32,LOW);  
    digitalWrite(42,HIGH);  
    analogWrite(7,200);  
    digitalWrite(36,LOW);  
    digitalWrite(38,HIGH);  
}
```

```
void STOP(){  
    analogWrite(3,200);
```

```
digitalWrite(32,LOW);  
digitalWrite(42,LOW);  
analogWrite(7,200);  
digitalWrite(36,LOW);  
digitalWrite(38,LOW);  
}  
}
```

Kod do pada:

```
#include <SPI.h>

#include <Servo.h>

Servo myservo, myservo2;

const int slaveSelectPin = 13;

unsigned char PS2buf[10];

unsigned char i;

void setup()
{
    Serial.begin(9600);

    Serial.println("Inization...");

    pinMode(slaveSelectPin,OUTPUT);
    digitalWrite(slaveSelectPin,HIGH);

    SPI.begin();

    SPI.setBitOrder(LSBFIRST);

    SPI.setDataMode(SPI_MODE3);

    SPI.setClockDivider(SPI_CLOCK_DIV64);

    delay(20);

    Serial.println("Inizatin finished.");

    pinMode(3,OUTPUT);

    pinMode(5,OUTPUT);

    pinMode(2,OUTPUT);

    pinMode(9,OUTPUT);
```

```
pinMode(6,OUTPUT);  
pinMode(8,OUTPUT);  
myservo.attach(13);  
myservo2.attach(11);  
myservo2.write(57);  
myservo.write(30);  
}
```

```
void loop()  
{  
  if(Get_PS2Dat(PS2buf))  
    for(i=0;i<9;i++)  
    {  
      Serial.print(PS2buf[i]);  
      Serial.print(" ");  
  
      if(i==3 && PS2buf[i]==239)  
        {przod();}  
      if(i==3 && PS2buf[i]==255)  
        {STOP();}  
      if(i==3 && PS2buf[i]==191)  
        {tyl();}  
      if(i==3 && PS2buf[i]==127)  
        {lewo();}  
      if(i==3 && PS2buf[i]==223)  
        {prawo();}
```

```
if(i==4 && PS2buf[i]==239)
{myservo.write(140);
delay(3000);// sets the servo position according to the scaled value
myservo2.write(110);}
if(i==4 && PS2buf[i]==127)
{myservo.write(30);
myservo2.write(57);}
}
```

```
Serial.println();
delay(250);
}
```

```
unsigned char PS2_RWByte(unsigned char dat)
{
    SPI.transfer(dat);
    //while(!(SPSR&0x80));
    return(SPDR);
}
```

```
unsigned char Get_PS2Dat(unsigned char *buf)
{
    unsigned char i;
    digitalWrite(slaveSelectPin,LOW);
    delayMicroseconds(15);
    buf[0]=PS2_RWByte(0x01);delayMicroseconds(15);
```

```

    buf[1]=PS2_RWByte(0x42);delayMicroseconds(15);
    buf[2]=PS2_RWByte(0x00);delayMicroseconds(15);
    buf[3]=PS2_RWByte(0x00);delayMicroseconds(15);
    buf[4]=PS2_RWByte(0x00);delayMicroseconds(15);
    buf[5]=PS2_RWByte(0x00);delayMicroseconds(15);
    buf[6]=PS2_RWByte(0x00);delayMicroseconds(15);
    buf[7]=PS2_RWByte(0x00);delayMicroseconds(15);
    buf[8]=PS2_RWByte(0x00);delayMicroseconds(15);
    digitalWrite(slaveSelectPin,HIGH);
    if((buf[0]==0xff)&&(buf[1]==0x41)&&(buf[2]==0x5a))
    return 1;
    if((buf[0]==0xff)&&(buf[1]==0x73)&&(buf[2]==0x5a))
    return 2;
    return 0;
}

```

```

void tyl(){
    analogWrite(5,200);
    digitalWrite(3,LOW);
    digitalWrite(2,HIGH);
    analogWrite(9,200);
    digitalWrite(8,HIGH);
    digitalWrite(6,LOW);
}

```

```

void przod()

```

```
{  
  analogWrite(5,200);  
  digitalWrite(2,LOW);  
  digitalWrite(3,HIGH);  
  analogWrite(9,200);  
  digitalWrite(6,HIGH);  
  digitalWrite(8,LOW);  
}
```

```
void prawo(){  
  analogWrite(5,200);  
  digitalWrite(3,HIGH);  
  digitalWrite(2,LOW);  
  analogWrite(9,200);  
  digitalWrite(6,HIGH);  
  digitalWrite(8,LOW);  
}
```

```
void lewo(){  
  analogWrite(5,200);  
  digitalWrite(3,LOW);  
  digitalWrite(2,HIGH);  
  analogWrite(9,200);  
  digitalWrite(6,LOW);  
  digitalWrite(8,HIGH);  
}
```

```
void STOP(){  
    analogWrite(5,20);  
    digitalWrite(2,LOW);  
    digitalWrite(3,LOW);  
    analogWrite(9,200);  
    digitalWrite(6,LOW);  
    digitalWrite(8,LOW);  
}
```