



UNIWERYTET RZESZOWSKI

Realizacja modelu podajnika taśmowego połączonego z sortownikiem opartym o płytkę Arduino oraz belkę tensometryczną.

Marcin Pater

Hubert Hara

Praca wykonana w ramach projektu z kursu podstaw programowania robotów

Cele projektu

Realizacja sterowania modelem podajnika taśmowego z wykorzystaniem płytki Arduino NANO oraz belki tensometrycznej. Celem projektu było wypracowanie rozwiązania które pozwoli na rozróżnianie oraz sortowanie elementów, gdzie czynnikiem decydującym jest masa elementu sortowanego.

Geneza projektu

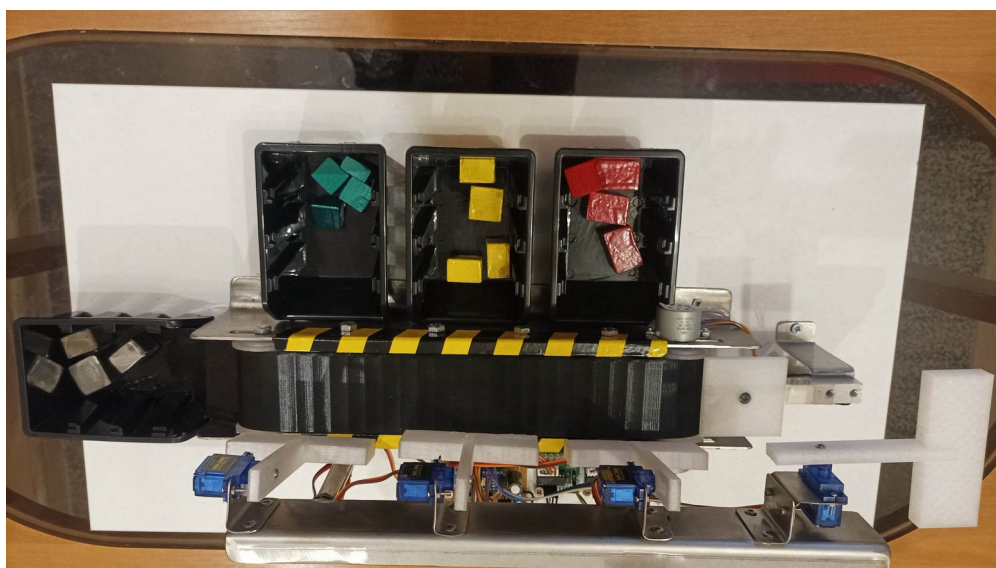
Z racji realizowanego kierunku studiów jakim jest Mechatronika oraz fakt iż mamy styczność z wieloma urządzeniami takimi jak sensory i akulatory oraz z zamykaniem do pracy z mikrokontrolerami takimi jak Arduino czy też ESP32 powstał pomysł budowy układu, który umożliwi sortowanie elementów przy rozpoznawaniu ich masy. W tym celu wykorzystano zjawisko zmiany rezystancji przy elementach jakimi są tensometry pod wpływem ich odkształcenia, wykorzystano gotowy element belki tensometrycznej. Element ten w połączeniu z płytką Arduino daje możliwość detekcji masy elementu. Elementy za pomocą taśmy napędzanej silnikiem krokowym jest przenoszony do odpowiedniej lokalizacji a następnie jest spychany przez serwomechanizm do odpowiedniego pojemnika. Kuwetki są dodatkowo oznaczone kolorami oraz elementy o masie do której są przypisane.

Elementy użyte do realizacji projektu

Element	liczba sztuk
Arduino Nano	1
Belka tensometryczna	1
Moduł HX711	1
Silnik krokowy	2
Serwomechanizm sg90	4
Regulator napięcia 7805	1
Dioda N1004	4
Wałki toczne	4
Wałek końcowy	1
Wałek napędowy	1
Taśma podajnika	1
Łopatkki spychające	4
Płyta do umocowania modelu	1
Zasilacz	1
Korytka na elementy	4

Konstrukcja układu

Model układu sortującego został umocowany na płycie meblowej co umożliwia jego łatwy transport oraz pozwala na wy pozycjonowanie elementów. Pierwsze co rzuca się w oczy to podajnik taśmowy(rys.1.), został on zrobiony wygiętych elementów wykonanych ze stali nierdzewnej które stanowią podstawę dla napędu taśmy oraz serwomechanizmów. Pomiędzy elementami szkieletu zamocowano wałki wykonane zostały z materiału ABS przy pomocy drukarki 3D (rys.2)



Rys.1. Układ sortowania



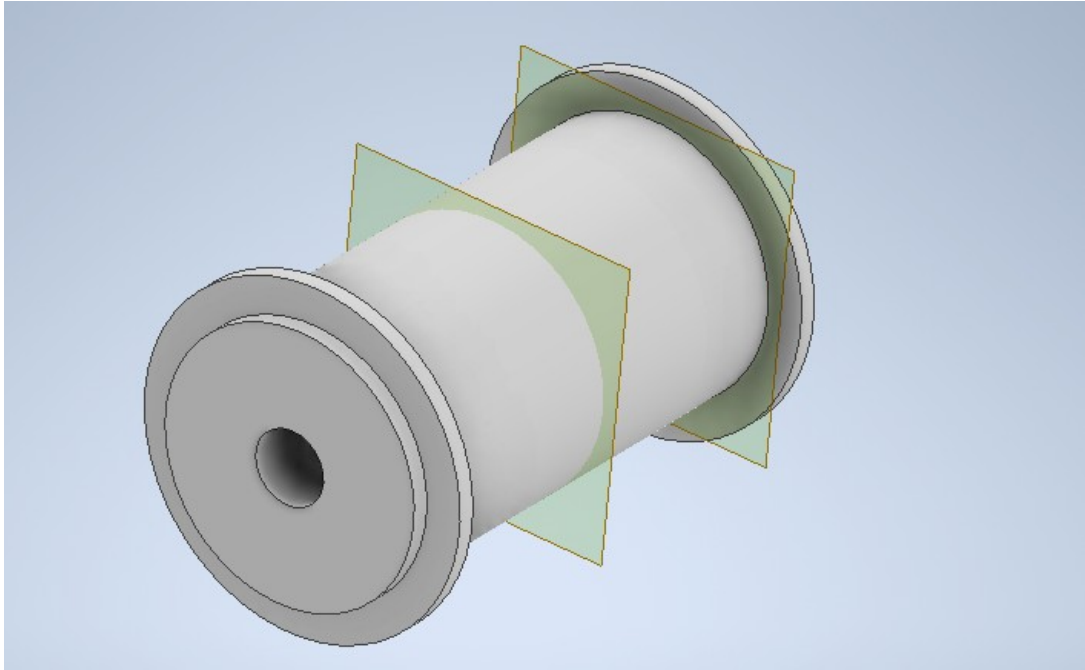
Rys.2 Proces drukowania elementów

W celu dobrania odpowiednich parametrów wydruku wykonano wiele prób aby dobrać parametry takie jak wypełnienie czy też prędkość wydruku.



Rys.3. próbki z różnym wypełnieniem.

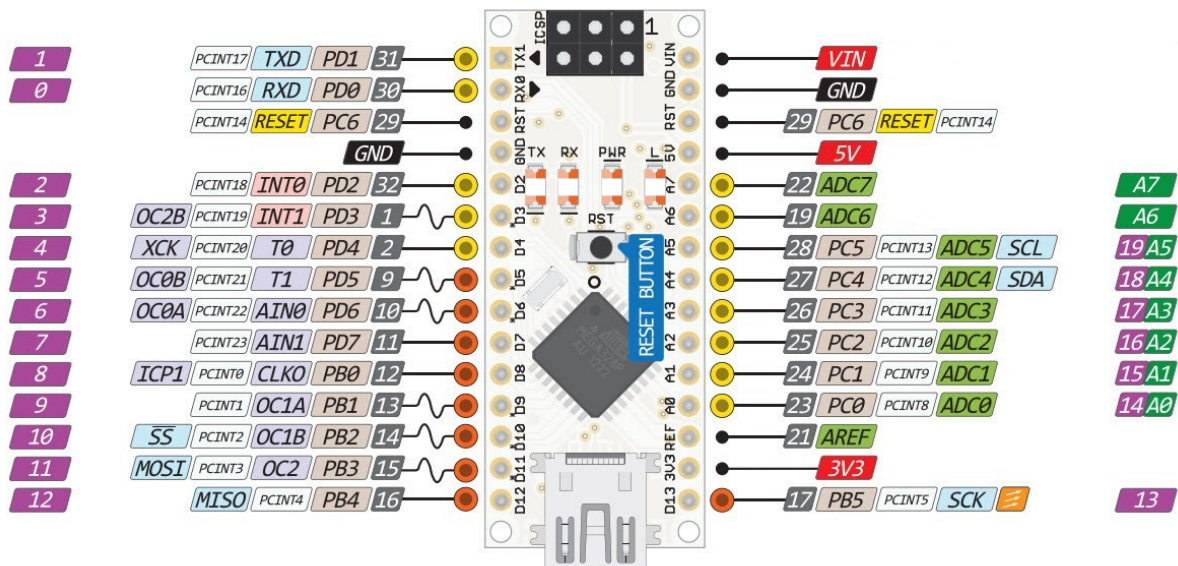
Po uzyskaniu akceptujących parametrów wydrukowano wałki oraz łopatki, Elementy te wymagały uprzedniego zaprojektowania w programie Autodesk Inventor (rys.4.).



Rys.4. Szkic wałka końcowego

Taśma podajnika jest ciekawym elementem ponieważ został on również wydrukowany tym razem z gumy dzięki czemu wydruk jest elastyczny dobrze współpracuje z wałkami mi pozwala na przenoszenie elementów.

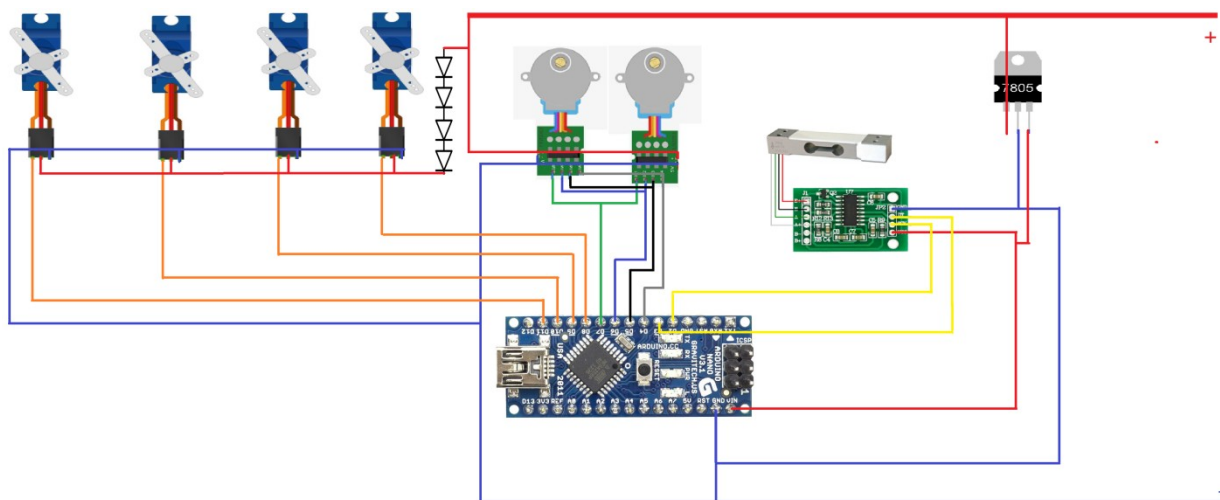
Kolejnym aspektem jest elektronika jaka została użyta w układzie. Płytką Arduino Nano(rys.5.) pełni role kontrolera urządzenia odpowiedzialna jest za pomiar wagiysterowanie silników krokowych oraz serwomechanizmów.



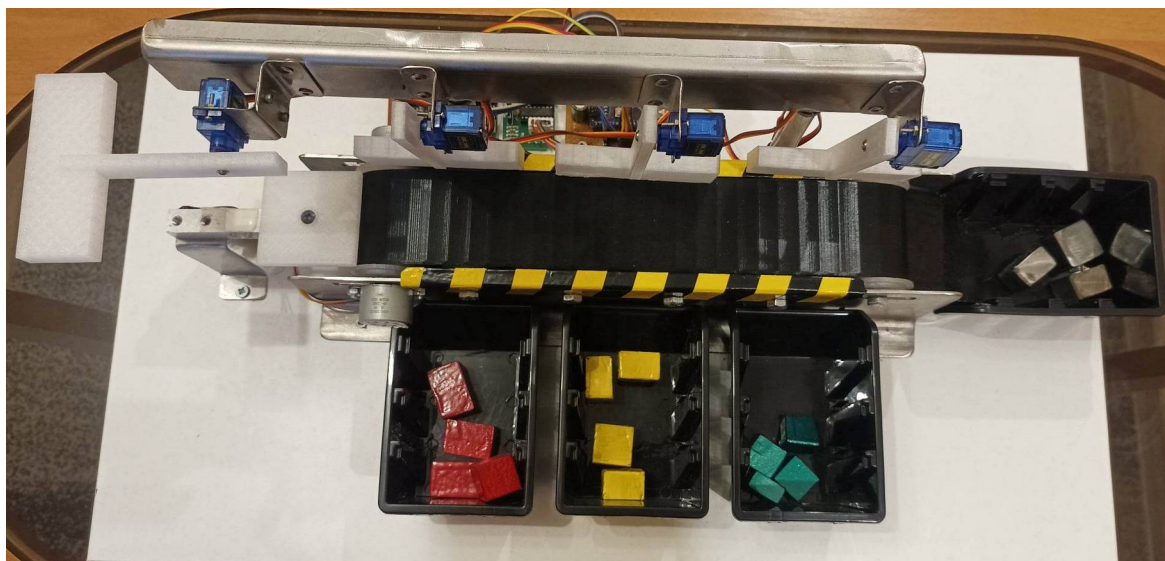
Rys.5. Płytką Arduino NANO

Układ zasilany jest z zasilacza 7V dla serwomechanizmów napięcie jest obniżane za pomocą diod N1004 jako że na każdej diodzie spadek wynosi 0,5v to wynikowo serwomechanizmy dostają 5V

silniki krokowe są zasilane bez obniżania napięcia a płytka przez regulator zabieg ten został zastosowany z powodu ograniczonej wydajności prądowej płytki Arduino. Schemat połączenia przedstawiono na (rys.6.)



Rys.6. Schemat połączenia układu



Rys.7. Gotowy układ sortownika

Gdy cała konstrukcja była gotowa napisano program do obsługi peryferiów podpiętych do mikrokontrolera. Kod zrealizowano w języku C. Układ po podaniu napięcia bądź po resecie wykonuje kalibrację belki tensometrycznej następnie pozycjonowane są serwomechanizmy po tym procesie układ jest gotowy do pracy. Po ułożeniu elementu na belce tensometrycznej wykonywane są trzy pomiary aby wyeliminować błąd wyciągana jest średnia z pomiarów. Następnie serwomechanizm spycha element ze stolika ma taśmę i krokiem kolejnym jest uruchomienie silników krokowych oraz ich praca aż do momentu który wynika z warunku jaki spełnia segregowany przedmiot. Gdy taśma się zatrzymuje zostaje wysterowane serwo które dokonuje zrzucenia elementu z podajnika do pojemnika. Zostaje również zwiększona wartość w ewidencji która jest wystawiana na porcie

szeregowym i może być wyświetlana na monitorze komputera za pomocą monitora portu szeregowego.

```
#include "Servo.h" //Biblioteka służąca do obsługi serwomechanizmu
#include "HX711.h"

// HX711 circuit wiring
const int LOADCELL_DOUT_PIN = 2;
const int LOADCELL_SCK_PIN = 3;

HX711 scale;
long reading = 0;
long calibration = 0;

Servo myservo; // Utworzenie obiektu myservo
int pos = 0; //Początkowa pozycja serwa
Servo myserv1; // Utworzenie obiektu myservo
int pos1 = 0; //Początkowa pozycja serwa
Servo myserv2; // Utworzenie obiektu myservo
int pos2 = 0; //Początkowa pozycja serwa
Servo myserv3; // Utworzenie obiektu myservo
int pos3 = 0; //Początkowa pozycja serwa
int pojemnik_0 = 0;
int pojemnik_1 = 0;
int pojemnik_2 = 0;
int pojemnik_3 = 0;

void setup() {
  //tenso calibration
  Serial.begin(9600);
  scale.begin(LOADCELL_DOUT_PIN, LOADCELL_SCK_PIN);
  delay(3000);
  if (scale.is_ready()) {
    reading = scale.read();
  }
  delay(300);
  if (scale.is_ready()) {
    reading += scale.read();
  }
  delay(300);
  if (scale.is_ready()) {
    reading += scale.read();
  }
  calibration = (reading / 3);
  Serial.println("calibration ");
  Serial.println(calibration);

  //tenso end
  pinMode(4, OUTPUT);
```



```

pinMode(5, OUTPUT);
pinMode(6, OUTPUT);
pinMode(7, OUTPUT);

digitalWrite(4, LOW);
digitalWrite(5, LOW);
digitalWrite(6, LOW);
digitalWrite(7, LOW);
myserv.attach(8); //Przyłączenie serwomechanizmu do pinu 8 w Arduino
myserv1.attach(9); //Przyłączenie serwomechanizmu do pinu 9 w Arduino
myserv2.attach(10); //Przyłączenie serwomechanizmu do pinu 10 w Arduino
myserv3.attach(11); //Przyłączenie serwomechanizmu do pinu 11 w Arduino

// procedura kalibracji
for (pos = 0; pos <= 125; pos += 1)
{
    myserv.write(pos); //ustawienie zadanej pozycji
    delay(5); //Czas na zmianę pozycji
}
delay(200);
for (pos = 125; pos >= 0; pos -= 1)
{
    myserv.write(pos); //ustawienie zadanej pozycji
    delay(5); //Czas na zmianę pozycji
}

delay(200);
for (pos = 0; pos <= 30; pos += 1)
{
    myserv1.write(pos); //ustawienie zadanej pozycji
    delay(5); //Czas na zmianę pozycji
}
delay(200);
for (pos = 30; pos >= 0; pos -= 1)
{
    myserv1.write(pos); //ustawienie zadanej pozycji
    delay(5); //Czas na zmianę pozycji
}
delay(200);
for (pos = 0; pos <= 30; pos += 1)
{
    myserv2.write(pos); //ustawienie zadanej pozycji
    delay(5); //Czas na zmianę pozycji
}
delay(200);
for (pos = 30; pos >= 0; pos -= 1)
{
    myserv2.write(pos); //ustawienie zadanej pozycji
    delay(5); //Czas na zmianę pozycji
}
delay(200);

```

```

for (pos = 0; pos <= 30; pos += 1)
{
    myserv3.write(pos); //ustawienie zadanej pozycji
    delay(5); //Czas na zmianę pozycji
}
delay(200);
for (pos = 30; pos >= 0; pos -= 1)
{
    myserv3.write(pos); //ustawienie zadanej pozycji
    delay(5); //Czas na zmianę pozycji
}
delay(3000);
//koniec kalibrowania serwomechanizmów
wypisz_ile();
}

void wypisz_ile()
{
    int print_clear = 0;
    for (print_clear = 0; print_clear < 20; print_clear++)
    {
        Serial.println("");
    }
    Serial.print("pojemnik_0 ");
    Serial.println(pojemnik_0);
    Serial.print("pojemnik_1 ");
    Serial.println(pojemnik_1);
    Serial.print("pojemnik_2 ");
    Serial.println(pojemnik_2);
    Serial.print("pojemnik_3 ");
    Serial.println(pojemnik_3);
    for (print_clear = 0; print_clear < 9; print_clear++)
    {
        Serial.println("");
    }
}

int wait_step_motor = 4;
int i = 0;
void step_motor(int steps)
{
    for (i = 0; i < steps; i++)
    {
        digitalWrite(4, HIGH);
        delay(wait_step_motor);
        digitalWrite(4, LOW);
        digitalWrite(5, HIGH);
        delay(wait_step_motor);
        digitalWrite(5, LOW);
        digitalWrite(6, HIGH);
        delay(wait_step_motor);
    }
}

```



```

digitalWrite(6, LOW);
digitalWrite(7, HIGH);
delay(wait_step_motor);
digitalWrite(7, LOW);
}
}
void loop() {
  //tenso rading
  if (scale.is_ready())
  {
    long reading = 0;
    for (int y = 0; y < 3; y++)
    {
      reading += ((scale.read() - calibration)) / 2;
      delay(300);
    }
    reading = reading / 3;
    //Serial.print("HX711 reading: ");
    //Serial.println(reading);

    if (reading > 3000)
    {
      //servo 0
      for (pos = 0; pos <= 125; pos += 1) //Zmiana pozycji (dodawanie) do 180 stopni
      {
        myserv.write(pos); //ustawienie zadanej pozycji
        delay(4); //Czas na zmiane pozycji
      }
      delay(1000);
      for (pos = 125; pos >= 0; pos -= 1) //Zmiana pozycji (dodawanie) do 180 stopni
      {
        myserv.write(pos); //ustawienie zadanej pozycji
        delay(2); //Czas na zmiane pozycji
      }
      delay(1000);
    }
    if (reading > 3000 && reading < 7000)
    {
      //servo 1
      step_motor(150);
      delay(2000);
      for (pos = 0; pos <= 35; pos += 1) //Zmiana pozycji (dodawanie) do 180 stopni
      {
        myserv1.write(pos); //ustawienie zadanej pozycji
        delay(5); //Czas na zmiane pozycji
      }
      delay(2000);
      for (pos = 35; pos >= 0; pos -= 1) //Zmiana pozycji (dodawanie) do 180 stopni
      {
        myserv1.write(pos); //ustawienie zadanej pozycji

```

```

    delay(5); //Czas na zmianę pozycji
}
pojemnik_0++;
wypisz_ile();
delay(3000);
}
if (reading > 7000 && reading < 18000)
{
    //serwo 2
    step_motor(600);
    delay(2200);
    for (pos = 0; pos <= 35; pos += 1) //Zmiana pozycji (dodawanie) do 180 stopni
    {
        myserv2.write(pos); //ustawienie zadanej pozycji
        delay(5); //Czas na zmianę pozycji
    }
    delay(2000);
    for (pos = 35; pos >= 0; pos -= 1) //Zmiana pozycji (dodawanie) do 180 stopni
    {
        myserv2.write(pos); //ustawienie zadanej pozycji
        delay(5); //Czas na zmianę pozycji
    }
    pojemnik_1++;
    wypisz_ile();
    delay(3000);
}
if (reading > 18000 && reading < 30000)
{
    //serwo 3
    step_motor(1200);
    delay(2000);
    for (pos = 0; pos <= 35; pos += 1) //Zmiana pozycji (dodawanie) do 180 stopni
    {
        myserv3.write(pos); //ustawienie zadanej pozycji
        delay(5); //Czas na zmianę pozycji
    }
    delay(2000);
    for (pos = 35; pos >= 0; pos -= 1) //Zmiana pozycji (dodawanie) do 180 stopni
    {
        myserv3.write(pos); //ustawienie zadanej pozycji
        delay(5); //Czas na zmianę pozycji
    }
    pojemnik_2++;
    wypisz_ile();
    delay(3000);
}
if (reading > 30000)
{
    step_motor(1500);
    pojemnik_3++;
    wypisz_ile();
}

```

```
}  
  
}  
else  
{  
  Serial.println("HX711 not found.");  
}  
delay(500);  
}
```

Obsługa układu

W celu uruchomienia układu należy podłączyć zasilacz po podaniu napięcia rozpocznie się proces kalibracji urządzenia. W tym czasie nie wolno dotykać belki tensometrycznej nie może się na niej również znajdować żaden element aby kalibracja została poprawnie wykonana. Gdy proces kalibracji zakończy się oraz na serwomechanizmy będą ustawione w pozycjach wyjściowych można umieścić element na stoliku belki układ automatycznie wykona trzy pomiary i na ich podstawie zdecyduje gdzie dany element ma trafić. Zostanie uruchomiona łopatką która umieści element na taśmie a następnie włączy się przesuw taśmy. Gdy element osiągnie wymagany punkt układ się zatrzyma i włączy się łopatką spychającą. Element zostanie umieszczony w pojemniku a w programie zostanie zaksięgowane miejsce gdzie jest zamieszczony element. Każdy element umieszczony na belce zostanie potraktowany identycznie gdy element jest za ciężki zostanie umieszczony w pojemniku na końcu taśmociągu. Nie zostanie zepchnięty do pojemników przez serwomechanizmy.

Literatura

[1] [https:// docs.arduino.cc/hardware/nano](https://docs.arduino.cc/hardware/nano)

[2] Arduino Nano (V2.3) User Manual.pdf

[3] <https://botland.com.pl/content/161-wykorzystanie-belki-tensometrycznej-za-posrednictwem-arduino>

[4] <https://strefainzyniera.pl/artukul/1010/arduino-i-serwomechanizm--kurs-arduino-cz-5>