

# AGH

## Złożone Systemy Cyfrowe

**Projekt:**

***Autorski robot piszący po klawiaturze***

**Autor:**

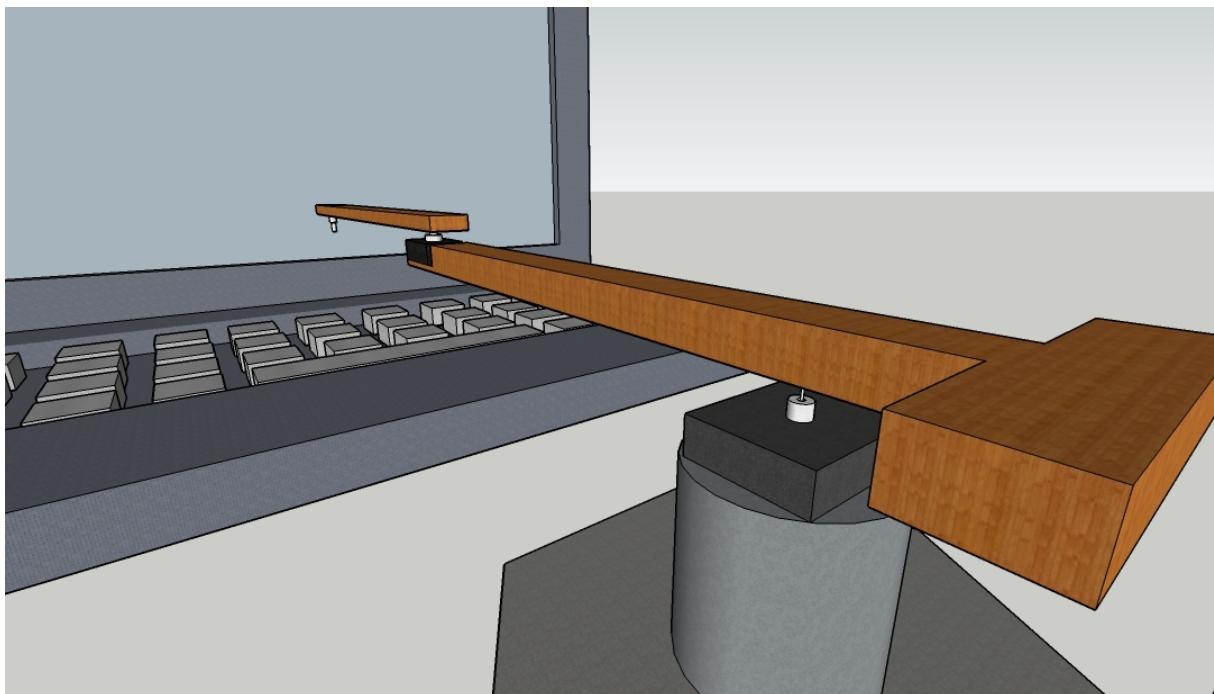
***Szymon Sukiennik***

# **SPIS TREŚCI**

- 1. Wstęp**
- 2. Opis działania sygnału PWM**
- 3. Opis działania serwomechanizmu**
- 4. Opis działania tworzonego robota**
- 5. Obliczenia/model dla kinematyki odwrotnej**
- 6. Wymagania projektowe**
- 7. Model starego prototypu**
- 8. Model aktualnego robota**
- 9. Harmonogram pracy**
- 10. Efekty pracy**
- 11. Bibliografia**

# **1. Wstęp**

Celem projektu jest zaprojektowanie i skonstruowanie robota zdolnego do pisania po określonej klawiaturze. Za sterowanie robotem będą odpowiedzialne serwomechanizmy sterowane odpowiednimi sygnałami PWM wysyłanymi przez mikrokontroler Arduino.



## **2. Opis działania sygnału PWM**

PWM (Pulse-Width-Modulation) - metoda modulacji szerokości impulsu napięciowego lub prądowego o stałej amplitudzie i częstotliwości. Jest to technika otrzymywania analogowych rezultatów przy cyfrowych wartościach.

PWM wykorzystywane są w przetwornicach impulsowych, w sterowaniu jasnością świecenia diod LED czy silnikami prądu stałego, także serwami. Właśnie to ostatnie zastosowanie jest wykorzystywane w tym projekcie.

Najważniejszym parametrem sygnału PWM jest współczynnik wypełnienia określany na podstawie poniższego wzoru:

$$k_w = \frac{T_{ON}}{T}$$

Okres sygnału (impulsu) jest sumą czasów wystąpień stanu niskiego i wysokiego.

$$T = T_{ON} + T_{OFF}$$

Gdzie:

$T_{ON}$  - czas stanu wysokiego sygnału

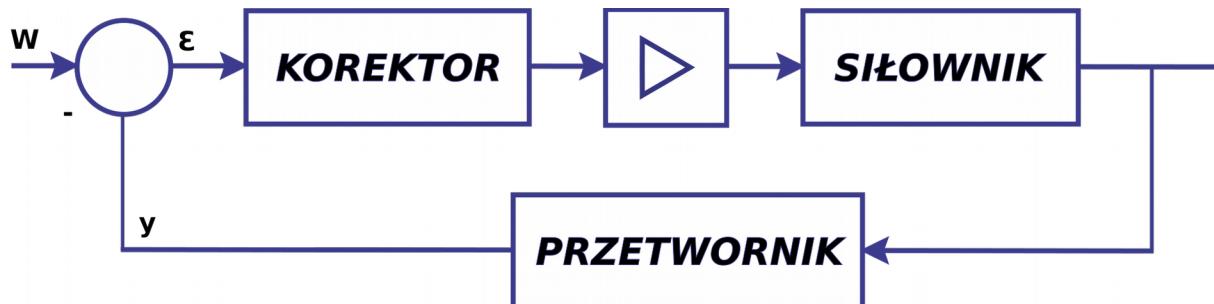
$T_{OFF}$  - czas stanu niskiego sygnału

$T$  - okres sygnału

Ostatecznie mnożąc współczynnik wypełnienia przez amplitudę otrzymamy średnią wartość napięcia. Zmieniając szerokość sygnału przy odpowiedniej częstotliwości jesteśmy w stanie uzyskać rezultat analogowy przy cyfrowych wartościach.

## **3. Opis działania serwomechanizmu**

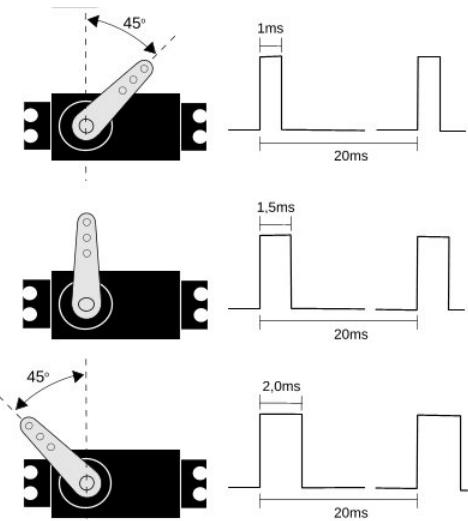
Serwomechanizm - zamknięty układ sterowania korzystający ze sprzężenia zwrotnego, w którym sygnałem wyjściowym jest jakaś dana, taka jak położenie, prędkość czy przyspieszenie. Często jest to po prostu przesunięcie.



Wartość wejściowa porównywana jest z przetworzonym przez przetwornik bieżącym sygnałem wyjściowym i powstały w ten sposób błąd podawany jest na człon korekcyjny, a dalej na wzmacniacz. Wzmocniony sygnał trafia do silownika, którego przemieszczenie jest wartością wyjściową układu. Ruch silownika do zadanej wartości zmniejsza błąd aż do jego zniwelowania.

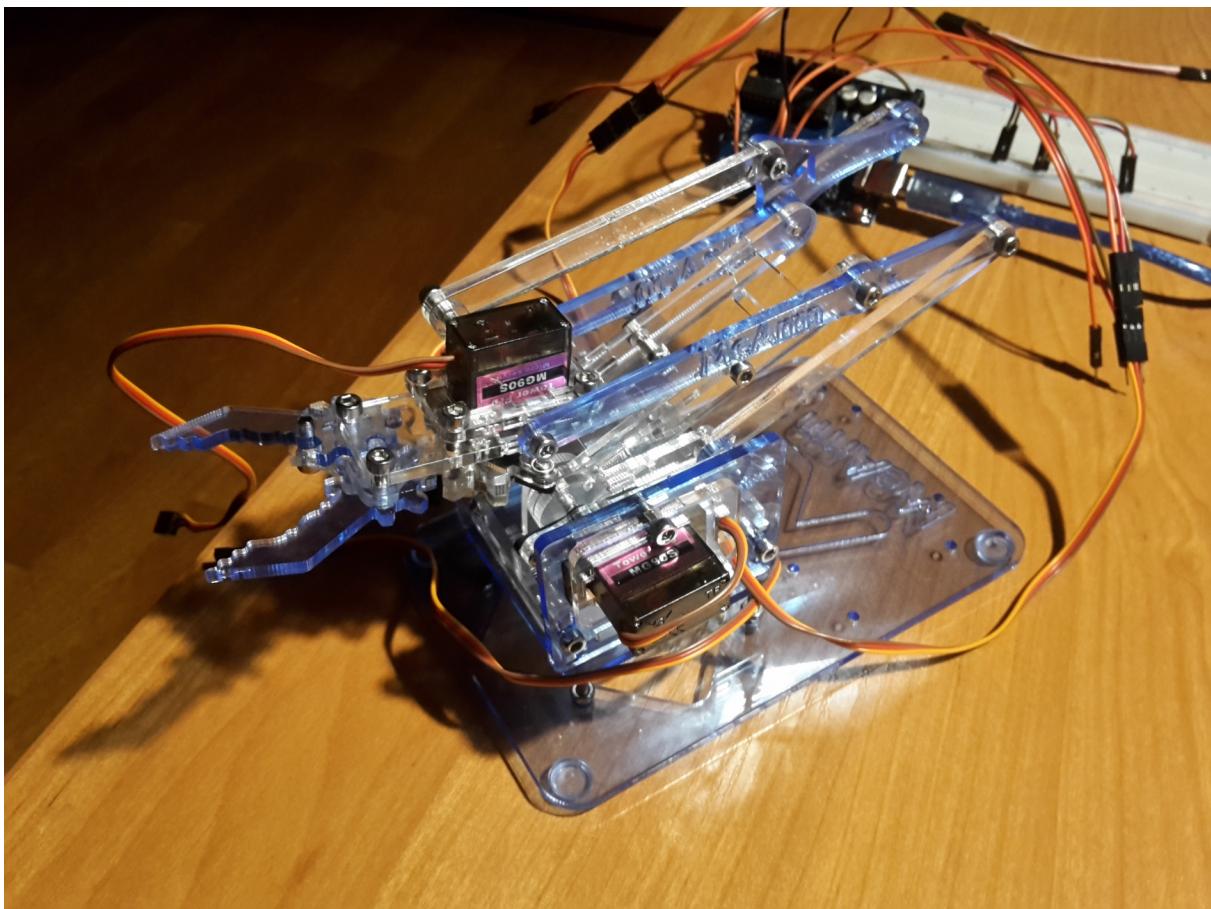
Zatem zadaniem serwomechanizmu jest likwidacja błędów regulacji powstających na skutek zmian wielkości wejściowej.

Kąt wychylenia jest determinowany przez szerokość sygnału PWM, który dociera na wejście. Serwomechanizm spodziewa się sygnału co 20ms (50Hz). Puls trwający 1.5ms ustawi silownik na pozycji 90 stopni (pozycja neutralna). Odpowiednio sygnał trwający 1 (minimum) oraz 2 (maksimum) ms odpowiada pozycji 0 i 180 stopni silownika. Te wartości mogą się różnić w zależności od marki oraz rodzajów serwomechanizmów. Zwykle pozycja neutralna oraz częstotliwość nie zmieniają się i są umowne jednak nie zawsze jest to pewne.



## 4. Opis działania tworzonego robota

Robot będzie składał się z dwóch ruchomych ramion (pozorna imitacja ręki i stawu łokciowego) poruszających się w dwóch płaszczyznach Y/Z oraz z podstawki, która będzie obracała robotem w płaszczyźnie X. Na końcu drugiego ramienia będzie przytwierdzone kolejne serwo odpowiadające za pracę chwytaka.



Długości ramion będą odpowiednio ustalone tak aby robot był w stanie ustawić się nad każdym (pesymistycznie: kilkoma) z klawiszy klawiatury. Całość zbudowana będzie z odpowiednio lekkich materiałów aby obciążenie serwomechanizmów było jak najmniejsze. Podstawa będzie umiejscowiona w strategicznym dla osiągalności, określonym miejscu.

Implementacja modułu sterowania odbywać się będzie w języku C z wykorzystaniem *Arduino IDE* i komplikacją przez *avr-gcc* (*GNU Toolchain*).

Do dyspozycji jest kilka podejść:

- Hard-coding klawiszy
  - $\{ „x”:(„y” \text{ def}, „z” \text{ def})=OK\}$

gdzie klawisz „**x**” będzie osiągany przy kątach wychylenia „**y**”/„**z**” odpowiednio pierwszego i drugiego ramienia.

- **Kinematyka odwrotna**

- o  $\{s': \{x', y', z'\} = [L1/L2/L3=const] = (m \text{ deg}, n \text{ deg}, o \text{ deg}) = OK\}$

gdzie klawisz „**s**” będzie miał zakodowaną lokację „**x**”/„**y**”/„**z**” i przy stałej określonej długości ramion **L1/L2/L3** mikrokontroler będzie obliczał kąty „**m**”/„**n**”/„**o**” ustawienia ramion (np. względem prostej równoległej do szerszej części klawiatury), przy których zostanie on osiągnięty

Jako, że operacja poruszania ramieniem jest **atomiczna i sekwencyjna** możliwy jest do wcisnięcia tylko jeden klawisz. Ogranicza to znaczaco zakres możliwości takiej konstrukcji. Mowa np. o:

- polskich znakach
- symbolach typu „€” itd.
- skrótach klawiszowych
- dużych literach bez użycia klawisza „Caps-lock”

## 5. Obliczenia/model dla kinematyki odwrotnej

KINEMATYKA ODWROTNIA

Model poogladowy

PO - punkt Odniesienia, PU - punkt Uchwytu

$L_1, L_2, L_{3a}, L_{3b}$  wektory łączące punkt odniesienia z punktem uchwytu

○ ━━○ - punkty obrotu

W celu uproszczenia rozwiązań danego modelu robiemy diagram 3D na dwa diagramy 2D. Problem polega na obliczeniu odpowiednich kątów serwo-mechanizmów na podstawie podanej lokalizacji w przestrzeni trójwymiarowej.

1. Wyliczenie  $\phi$ :

I sp.

$$\frac{x}{y} = \tan(\phi)$$

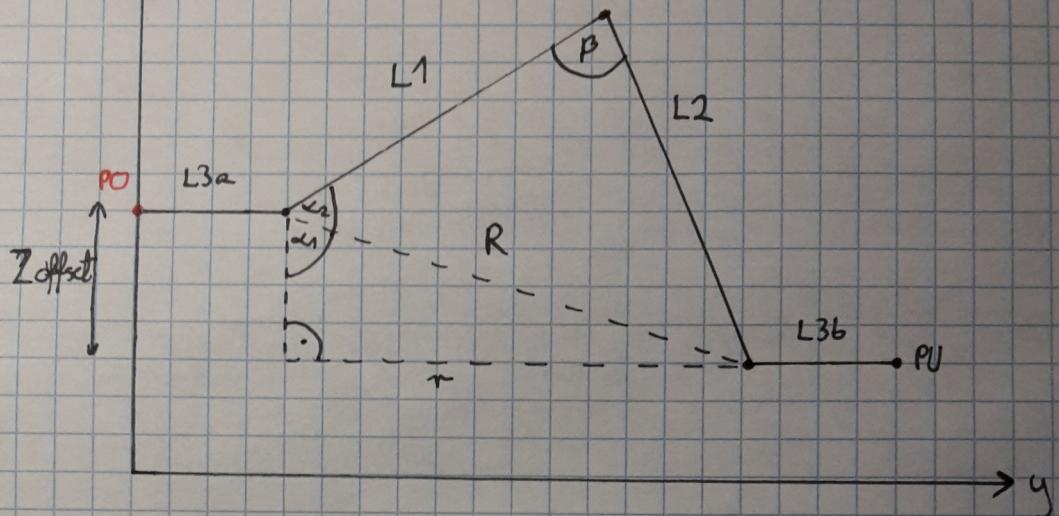
$$\cdot \phi = \tan^{-1}\left(\frac{x}{y}\right)$$

II sp.

$$r = \sqrt{x^2 + y^2}, \cos(\phi) = \frac{y}{r}$$

$$\cdot \phi = \cos^{-1}\left(\frac{y}{r}\right)$$

2.



$$R = \sqrt{Z_{\text{offset}}^2 + r^2}$$

$$\alpha_1 = \cos^{-1}\left(\frac{Z_{\text{offset}}}{R}\right)$$

ztr. cosinusör :  $c^2 = a^2 + b^2 - 2ab \cos(\gamma)$  many:

$$L2^2 = L1^2 + R^2 - 2L1 \cdot R \cdot \cos(\alpha_2)$$

$$\alpha_2 = \cos^{-1}\left(\frac{L2^2 - L1^2 - R^2}{-2L1 \cdot R}\right)$$

- $\alpha = \alpha_1 + \alpha_2$

$$R^2 = L1^2 + L2^2 - 2L1 \cdot L2 \cdot \cos(\beta)$$

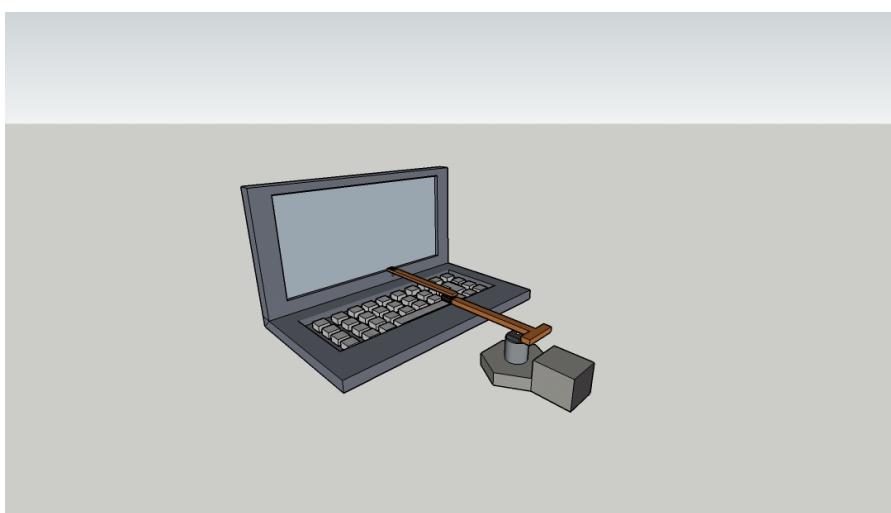
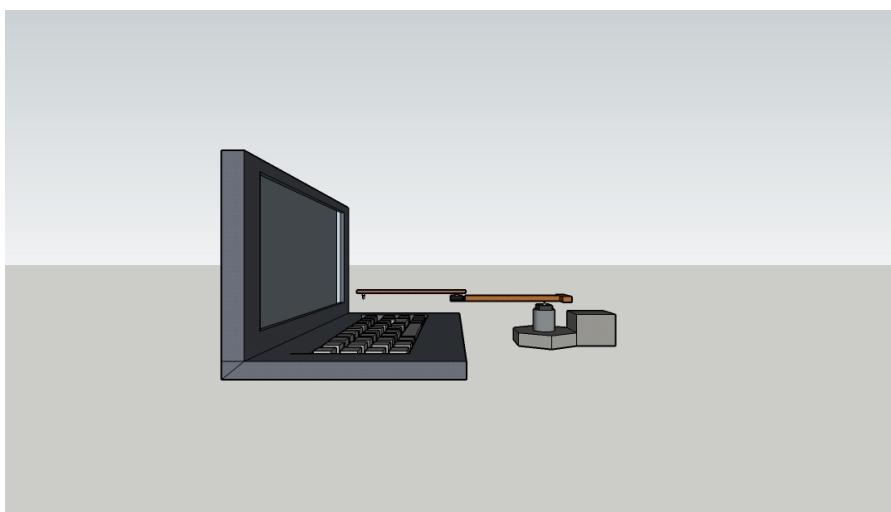
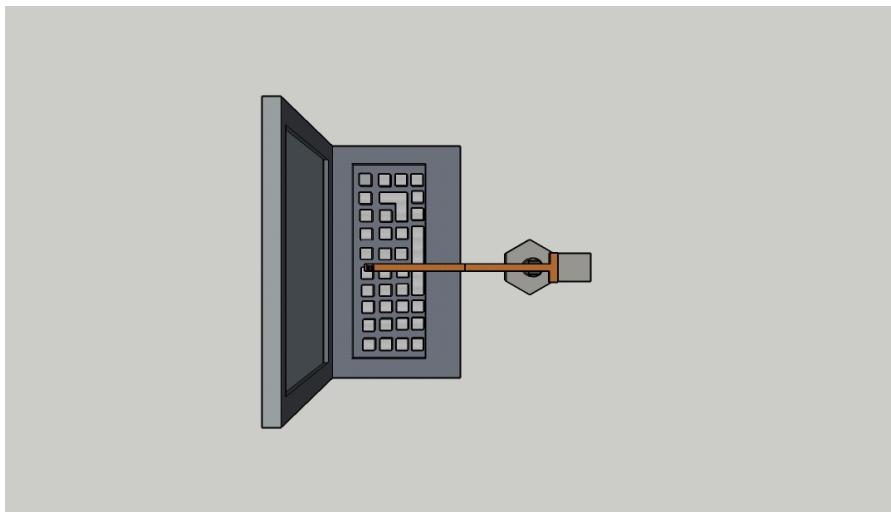
- $\beta = \cos^{-1}\left(\frac{R^2 - L1^2 - L2^2}{-2L1 \cdot L2}\right)$

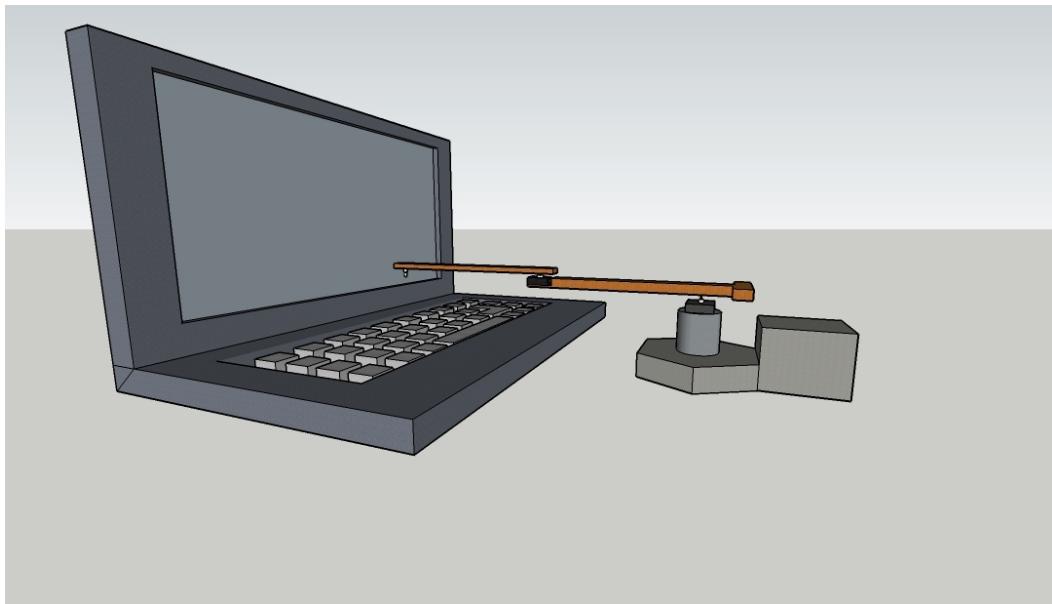
## **6. Wymagania projektowe**

- mikrokontroler Arduino
- 3 (4) serwomechanizmy
- siłownik lub serwomechanizm
- model ramienia i podstawka (z odpowiednim usztywnieniem, przeciwagą)
- laptop z klawiaturą (określony)
- określona ilość przewodów
- zasilanie zewnętrzne (np. od komputera) lub odpowiednio przystosowany magazyn energii (bateria)
- (ewentualnie) elementy niwelujące szum i przekłamania sygnałów generowanych przez mikrokontroler i same serwomechanizmy typu stabilizatory sygnału, kondesatory, rezystory itd. oraz element, który je zagreguje czyli płytka programowalna
- implementacja modułu sterującego robotem

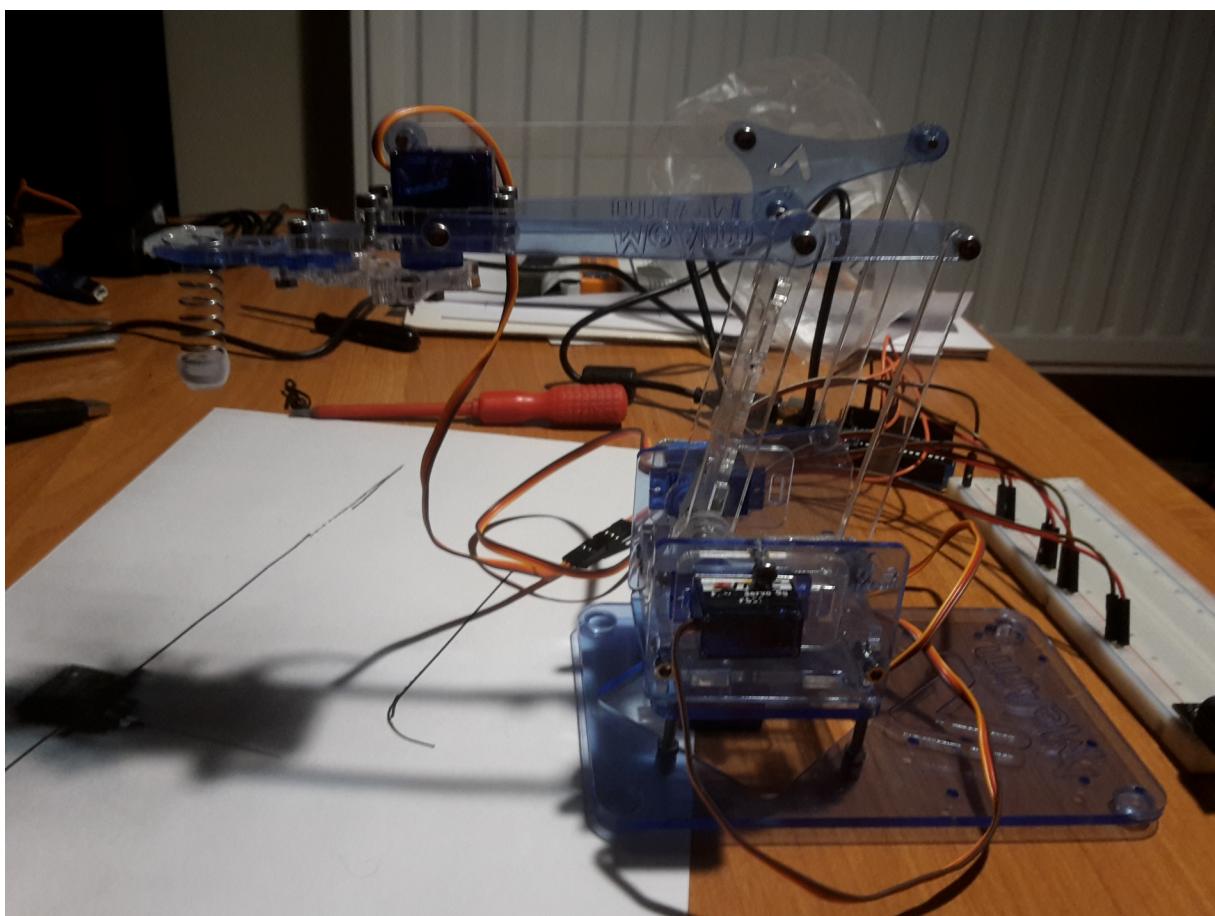
## 7. Model starego prototypu

Przykładowy model obrazujący wygląd, sposób działania robota.





## 8. Model aktualnego robota



## **9. Harmonogram pracy**

Wrzesień 2016

- ◆ przygotowanie wstępnej dokumentacji projektu
- ◆ wybranie mikrokontrolera do sterowania robotem (Arduino UNO)
- ◆ przygotowanie wstępnego projektu modelu robota oraz modelu poglądowego
- ◆ wybranie wstępnej metody sterowania robotem za pomocą mikrokontrolera

Listopad 2016

- ◆ implementacja komunikacji dwukierunkowej między mikrokontrolerem, a komputerem przez Serial Port
- ◆ przygotowanie szkieletu oprogramowania sterującego
- ◆ zakup mikrokontrolera Arduino UNO Rev3
- ◆ modyfikacja dokumentacji projektu
- ◆ pozyskanie serwomechanizmów
- ◆ rozpoczęcie budowy prototypu modelu

Grudzień 2016

- ◆ przebudowa dokumentacji projektu
- ◆ dodanie do dokumentacji informacji o sygnale PWM i sposobie działania serwomechanizmu
- ◆ wykonanie prototypu modelu robota z PCV (niepowodzenie)
- ◆ zmiana koncepcji modelu robota na prostego robota manipulującego
- ◆ zbudowanie prototypu modelu robota z części akrylicznych
- ◆ przygotowanie zewnętrznego zasilania dla serwomechanizmów
- ◆ aktualizacja dokumentacji
- ◆ zakup dodatkowych serwomechanizmów, ponieważ wcześniejsze uległy spaleniu (uszkodzeniu) przez niedopatrzenie. Niestety nowe są niemal o połowę słabsze i znacząco to wpływa na działanie robota.

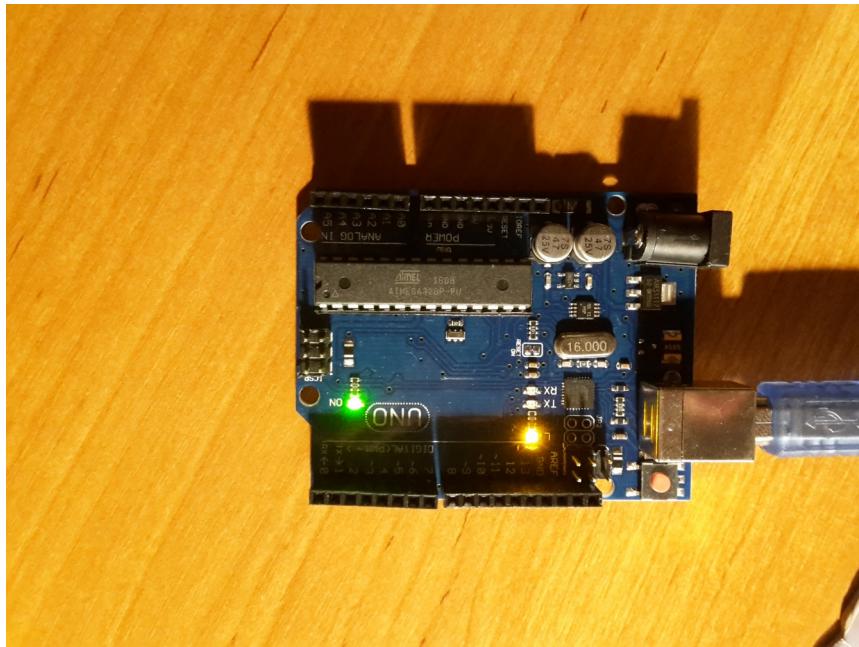
- ◆ rozpoczęcie pracy nad sterowaniem z wykorzystaniem kinematyki odwrotnej

## Styczeń 2017

- ◆ zakończenie implementacji biblioteki obsługującej ruch robota w wykorzystaniem kinematyki odwrotnej
- ◆ napisanie programu do pisania po klawiaturze z wykorzystaniem robota i wyżej wymienionej biblioteki
- ◆ hardcoding pozycji klawiszy
- ◆ testowanie
- ◆ aktualizacja dokumentacji

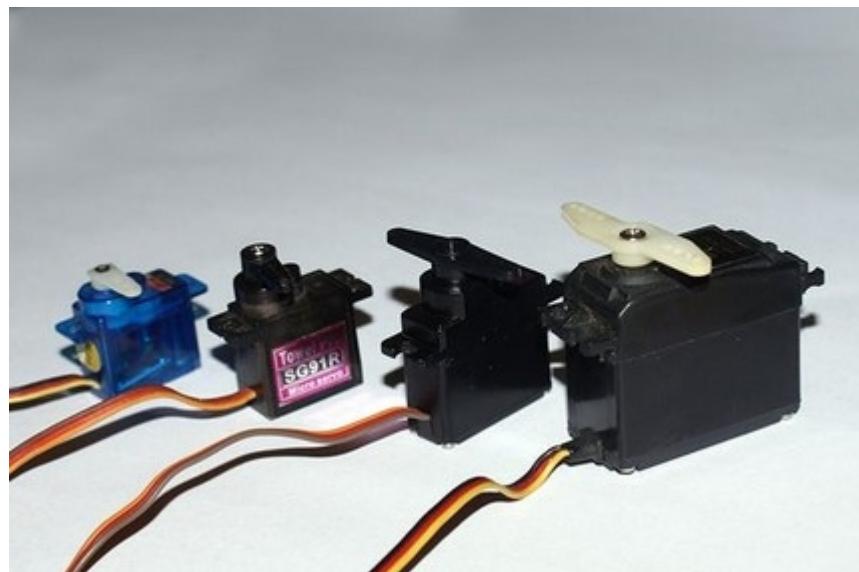
## 10. Efekty pracy

- Zakup Arduino UNO 3REV (ATMEGA328P)



- Pozyskanie kilku serwomechanizmów

- (od lewej) Hitec HS-55 - nieużyty
- TowerPro SG-91R - nieużyty
- TowerPro MG-90S - użyty 1 (1,4kg/cm(4,8V)-2,2kg/cm(6V) użyty 1
- Fitec FS-90 - [1,3kg/cm(4.8V)-1,5kg/cm(6V) użyte 3



- ◆ Pozyskanie przewodów, płytka stykowej, zasilacza, przełączki od zasilacza itd.



## ■ Implementacja komunikacji między Arduino, a laptopem przez Serial Port

Komputer wysyła na mikrokontroler znak. Arduino go odczytuje i zwraca to co odczytał.

FORMAT:

```
Sending <wysyłany znak>
<kod ASCII>
Readback <odebrany znak> at <czas> :: <numer wysyłanego znaku>
```

OUTPUT:

```
Opening serial port
Opening text file
Sending SPACE
32
Readback SPACE at 2016-12-14 01:37:01.433 :: 1
Sending SPACE
32
Readback SPACE at 2016-12-14 01:37:02.434 :: 2
Sending "a"
97
Readback "a" at 2016-12-14 01:37:03.435 :: 3
Sending "b"
98
Readback "b" at 2016-12-14 01:37:04.437 :: 4
Sending "c"
99
Readback "c" at 2016-12-14 01:37:05.438 :: 5
Sending "d"
100
Readback "d" at 2016-12-14 01:37:06.439 :: 6
Sending "e"
101
```

```
Readback "e" at 2016-12-14 01:37:07.440 :: 7
Sending "f"
102
Readback "f" at 2016-12-14 01:37:08.442 :: 8
Sending "g"
103
Readback "g" at 2016-12-14 01:37:09.443 :: 9
Sending "h"
104
Readback "h" at 2016-12-14 01:37:10.443 :: 10
Sending "i"
105
Readback "i" at 2016-12-14 01:37:11.444 :: 11
Sending "j"
106
Readback "j" at 2016-12-14 01:37:12.446 :: 12
Sending "k"
107
Readback "k" at 2016-12-14 01:37:13.447 :: 13
Sending "l"
108
Readback "l" at 2016-12-14 01:37:14.448 :: 14
Sending "m"
109
Readback "m" at 2016-12-14 01:37:15.450 :: 15
Sending "n"
110
Readback "n" at 2016-12-14 01:37:16.451 :: 16
Sending "o"
111
Readback "o" at 2016-12-14 01:37:17.452 :: 17
Sending "p"
112
Readback "p" at 2016-12-14 01:37:18.453 :: 18
Sending "q"
113
Readback "q" at 2016-12-14 01:37:19.455 :: 19
Sending "r"
114
Readback "r" at 2016-12-14 01:37:20.456 :: 20
...
...
```

Nawet dla pliku o długości 10 000 działa bezbłędnie przy różnych drukowalnych znakach.

### ■ Budowa prototypu modelu robota z PCV

Niestety mechanizm okazał się zbyt ciężki i cechował się zbyt dużą bezwładnością przez co wprawienie go w kontrolowany ruch stało się bardzo ciężkie, w istocie niemożliwe.

W kolejnym podejściu należy wykorzystać lżejsze materiały jak model skonstruowany z lekkich deseczek, płyty pilśniowej.

- Budowa manipulatora z części akrylicznych

Budowa powiadła się i robot działa wg zamierzonego sposobu. Niestety uległy spaleniu dość mocne jak na swoją wagę (13g - micro) serwa i musiałem ratować się jedynymi jakie znalazłem, niestety o dużo mniejszym momencie. Powoduje to dosyć wysokie stopy błędu i dość częste przekłamania przy pracy.

- Implementacja sterowania robotem z użyciem kinematyki odwrotnej

Omówienie API:

arm::goDirectlyTo(x,z,y) – przemieszczamy się płynnie do wybranego punktu (jeśli to możliwe). Dystans pokonujemy w krokach z małym opóźnieniem i co każdy krok wyliczana trasa może ulegać zmianie, na lepszą.

arm::goToPoint(x,y,z) – przemieszczamy się do wybranego punktu (jeśli to możliwe) natychmiastowo, najkrótszą drogą.

arm::isReachable(x,z,y) – sprawdzamy czy dany punkt jest osiągalny z naszej pozycji

arm::openGripper() - otwieramy chwytak

arm::closeGripper() - zamykamy chwytak

arm::getX/Y/Z() - sprawdzamy położenie na danej płaszczyźnie.

## **11. Bibliografia**

<https://www.arduino.cc/en/Tutorial/PWM>

<http://starter-kit.nettigo.pl/2012/01/co-to-jest-pwm/>

<http://www.mikrokontrolery.org/artykuly/elektronika/130-pwm-modulacja-szerokoci-impulsu>

<https://pl.wikipedia.org/wiki/Serwomechanizm>

[https://en.wikipedia.org/wiki/Servo\\_control](https://en.wikipedia.org/wiki/Servo_control)

<http://hobby.abxyz.bplaced.net/index.php?pid=3&aid=20>

[http://avrkwiat.nstrefa.pl/omnie/index.php?option=com\\_content&view=article&id=244&Itemid=272](http://avrkwiat.nstrefa.pl/omnie/index.php?option=com_content&view=article&id=244&Itemid=272)

<http://www.forbot.pl/forum/topics20/mechanika-serwa-modelarskie-w-robotyce-amatorskiej-kompendium-vt3902.htm>

<https://oscarliang.com/inverse-kinematics-and-trigonometry-basics/>

<http://www.learnaboutrobots.com/inverseKinematics.htm>

<http://www.robotyka.com/teoria.php/teoria.76>