

Dokumentacja Techniczna

„Symulator Pirat 2.0”

Gabriel Wójtowicz
Nikodem Ulrych
Michał Fleszar

Spis treści

1. Wstęp.....	4
2. Przyrządy pokładowe.....	4
3. Przyrządy pokładowe - Elektronika	15
4. Przyrządy Pokładowe - Efekt Końcowy.....	23
5. Drażek sterowy	27
6. Dźwignia wyczepu.....	30
7. System sterowania	33
8. Rama pod uchwyt monitorów	39
9. Wykończenie wizualne	41
10. Podsumowanie oraz Efekt Końcowy.....	45

1. Wstęp

Realizację projektu rozpoczęliśmy w maju 2022 roku. Za podstawę do modernizacji posłużył zbudowany przez nas symulator, który Posiadał układ przeniesienia ruchu sterów na potencjometry, współpracujący z symulatorem Condor 2.

Naszym celem było przystosowanie symulatora do nowoczesnej platformy symulacyjnej Microsoft Flight Simulator 2020, wykonanie oraz zaprogramowanie deski przyrządów, poprawa realizmu systemu sterowania, montaż nowego drążka sterowego oraz dodanie funkcji takich jak: wyczep, czy regulacja pedałów.

2. Przyrządy pokładowe

Pierwszym etapem prac było zaprojektowanie, wykonanie oraz zaprogramowanie deski przyrządów. Deskę oraz przyrządy zaprojektowaliśmy na wzór oryginalnych w programie Autodesk Fusion 360. Następnie po przygotowaniu w programie Prusa Slicer lub Cura Slicer wydrukowaliśmy je w technologii FDM z materiału PLA na drukarce Ender 3v2.

Przykładowy oryginalny układ przyrządów w szybowcu Pirat:



Tablica przyrządów składa się z:

- Prędkościomierza

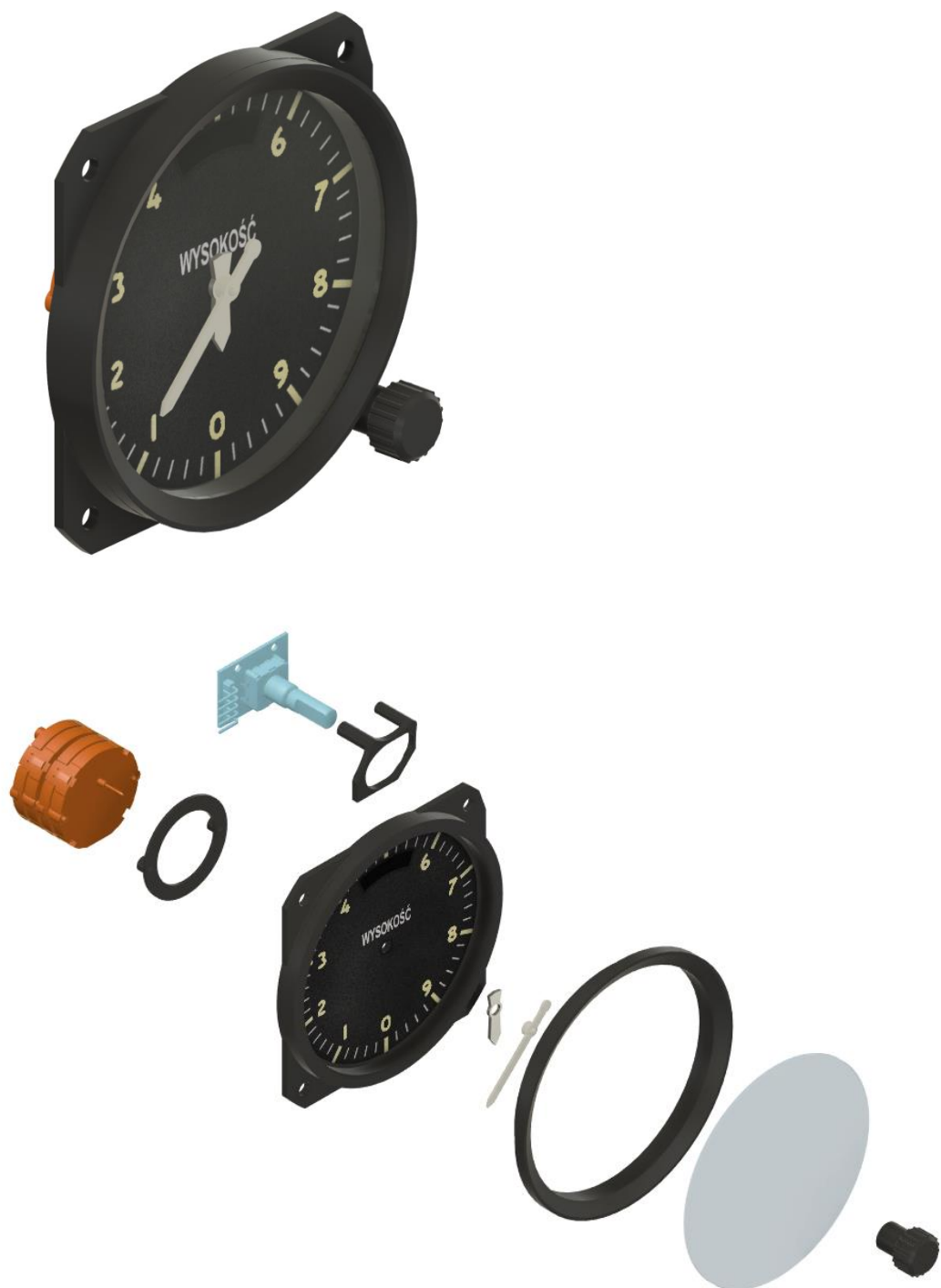




- Wariometrów: 5m/s i 30m/s



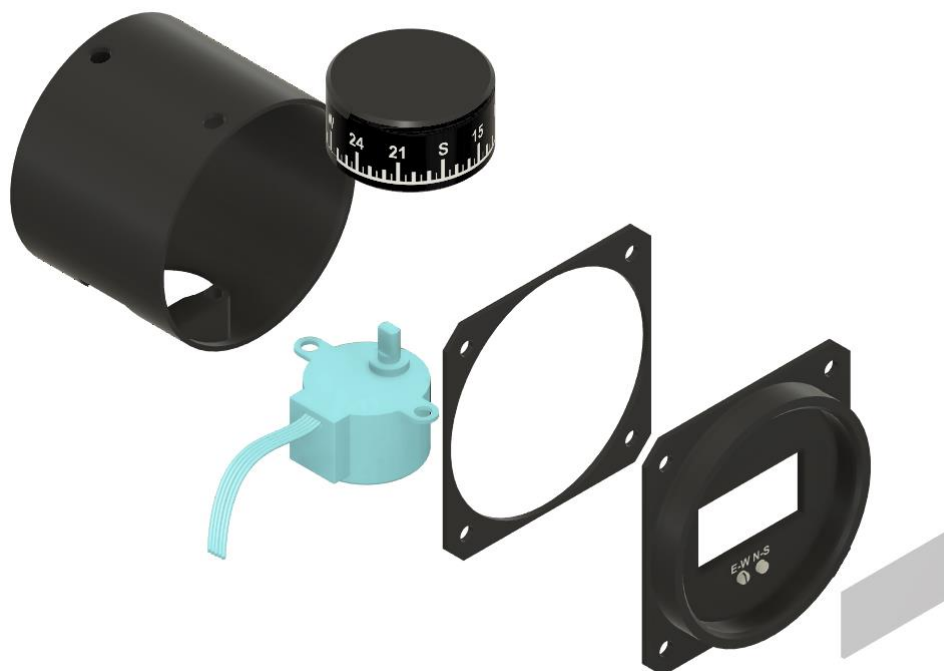
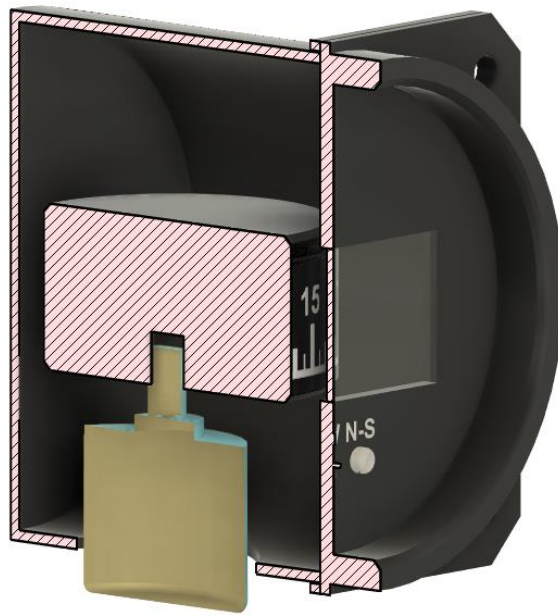
- Wysokościomierza



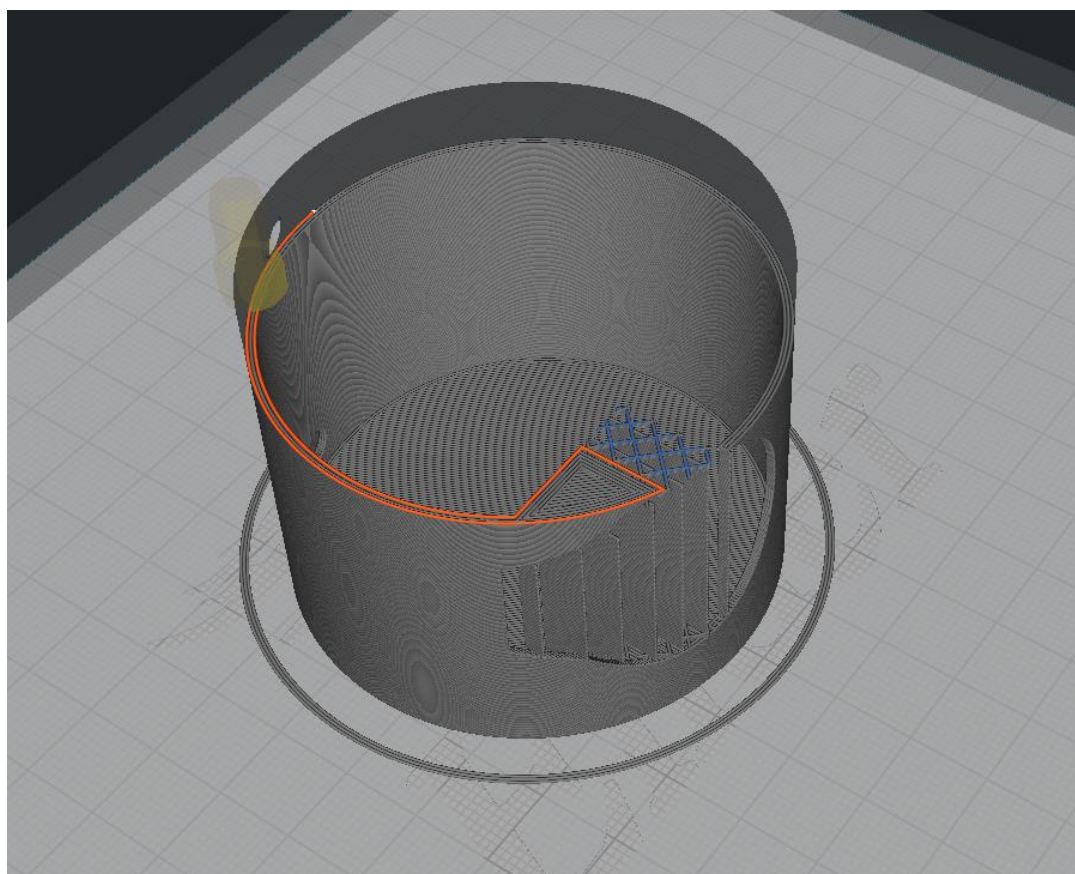
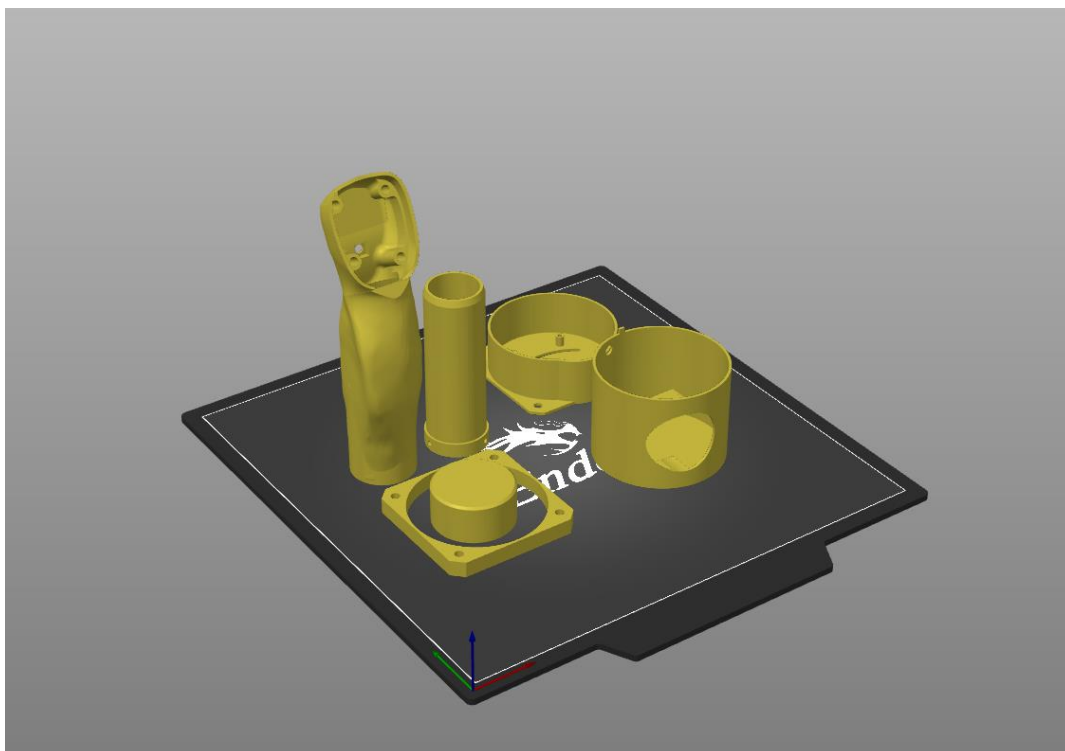
- Zakrętomierza z Chyłomierzem Poprzecznym

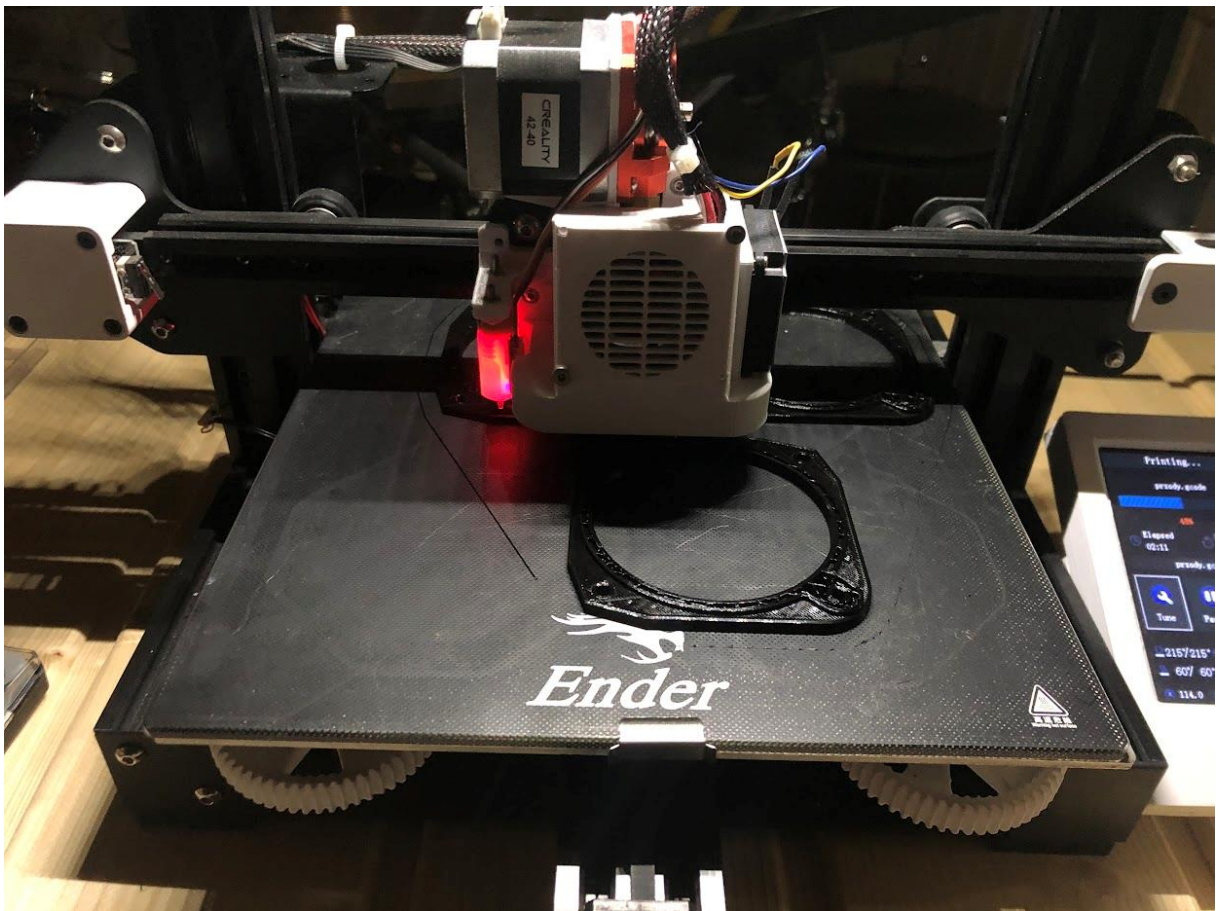
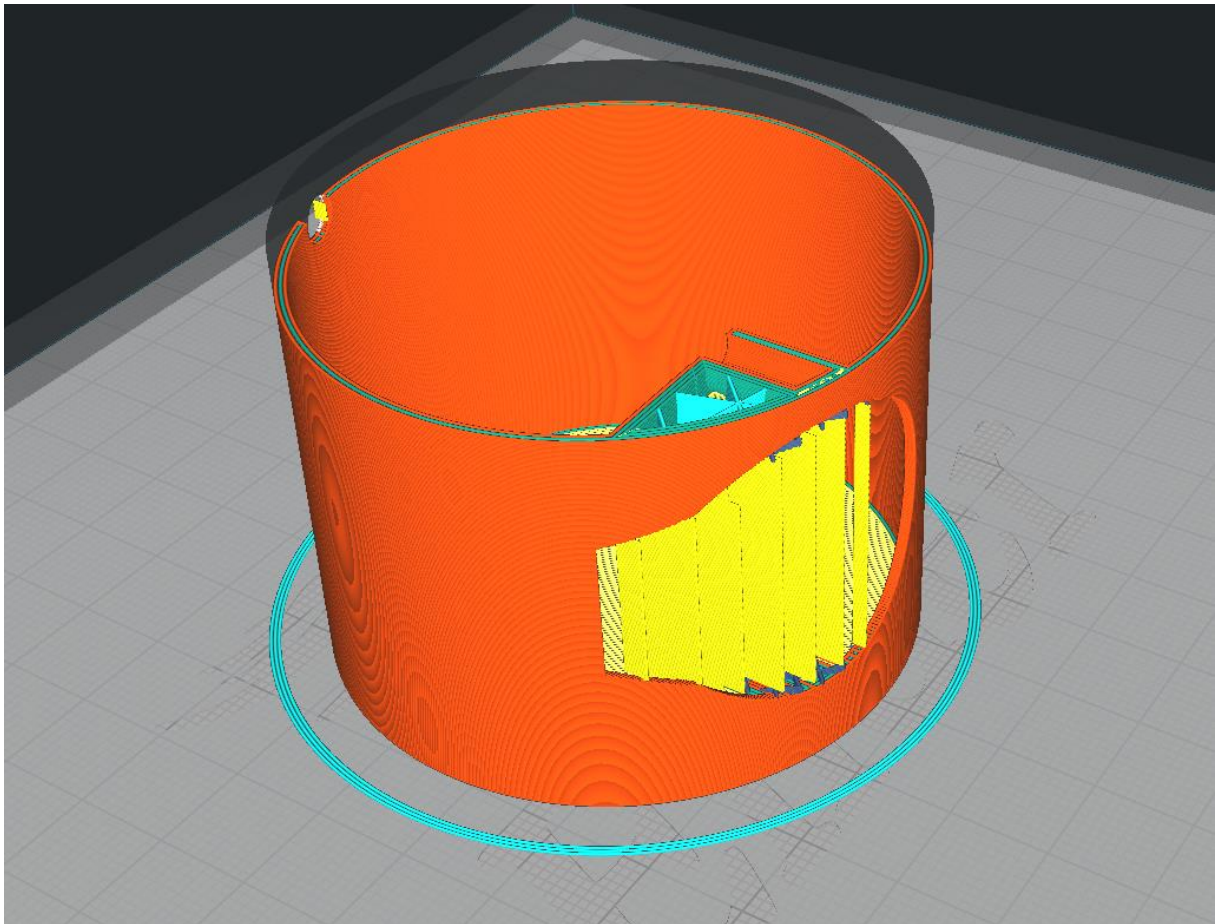


- Busoli

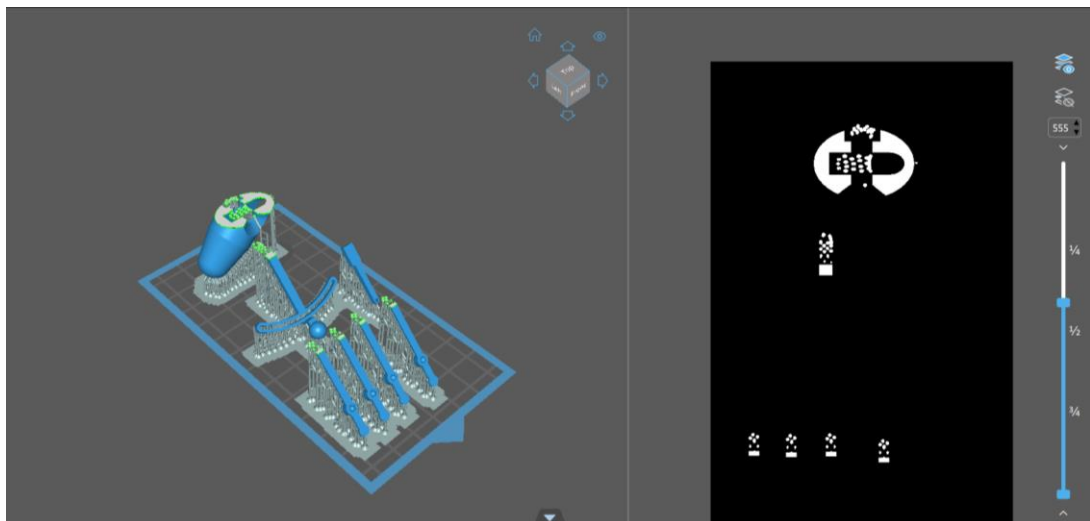
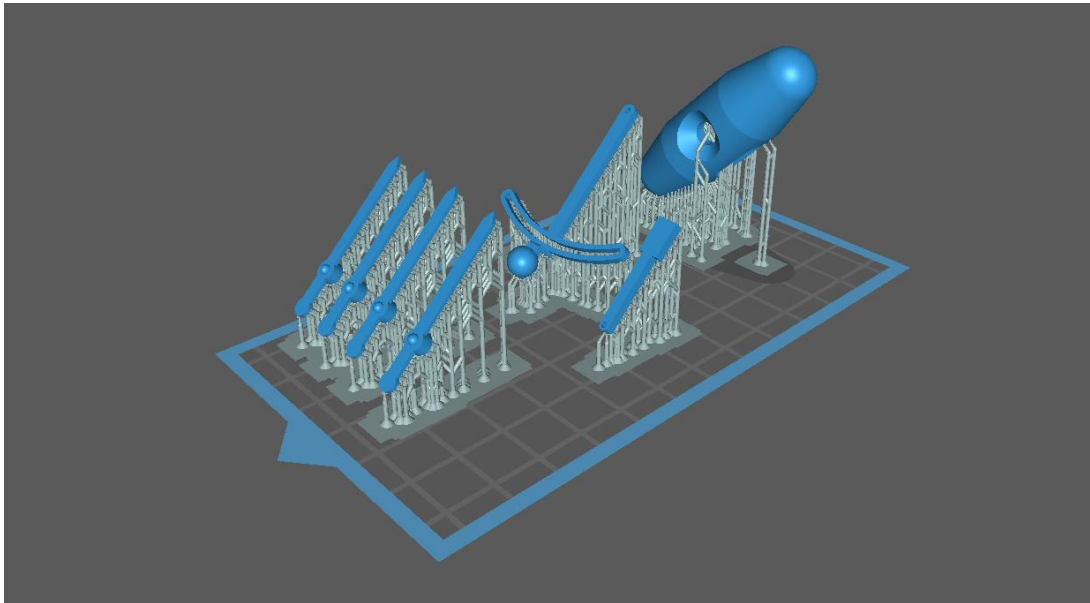


Zdjęcia poniżej przedstawiają proces przygotowania modelu 3D do druku, oraz druk

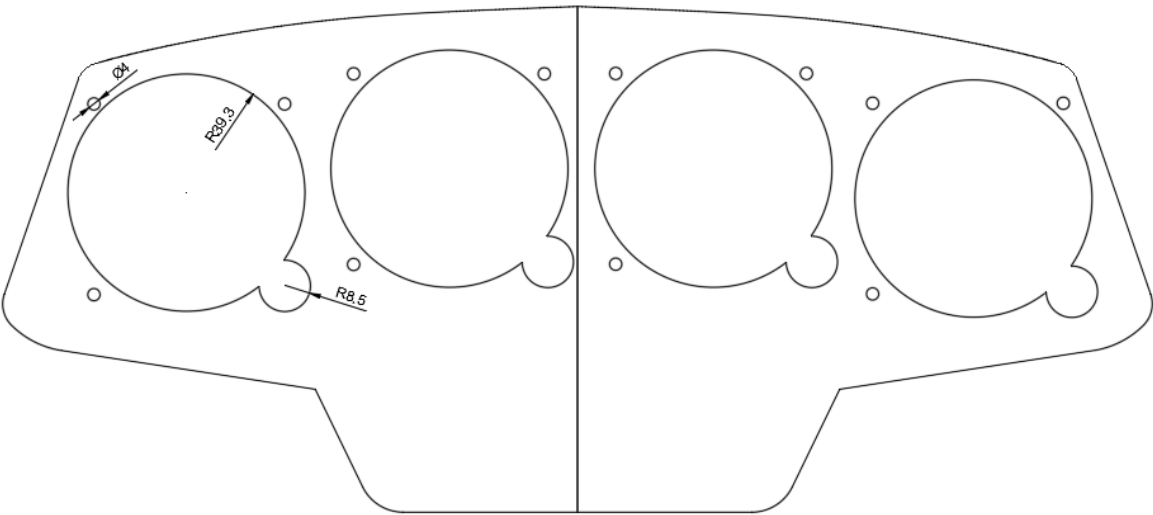




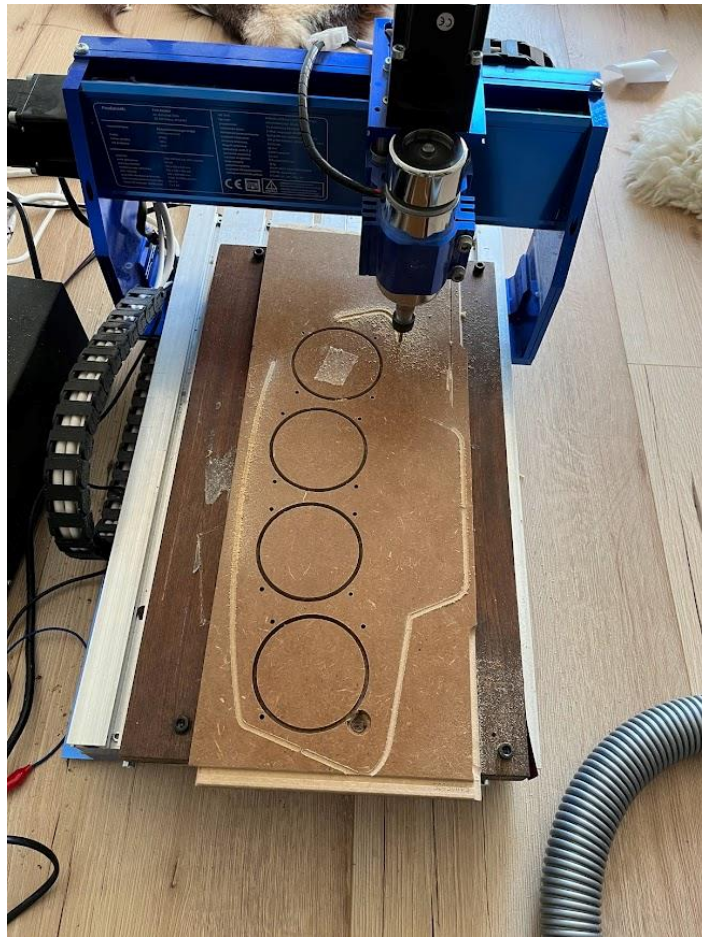
Wskazówki przyrządów zaprojektowaliśmy w programie Autodesk Fusion 360. Następnie po przygotowaniu programie Chitubox, wydrukowaliśmy w technologii druku żywicznego 3D, na drukarce Anycubic Foton S z materiału SLA. Poniższe zdjęcia przedstawiają ten proces.



Kolejnym etapem prac było zaprojektowanie oraz wykonanie w technologii CNC tablicy przyrządów. Tablica została zaprojektowana przez nas w programie Autodesk Fusion 360, do obsługi CNC posłużył program Mach 3.



Prodz	Nazwa przedmiotu		Data
1:1	Tablica przyrządów		
ZSTiH	Rysował		Nr rys.
	Sprawdził		



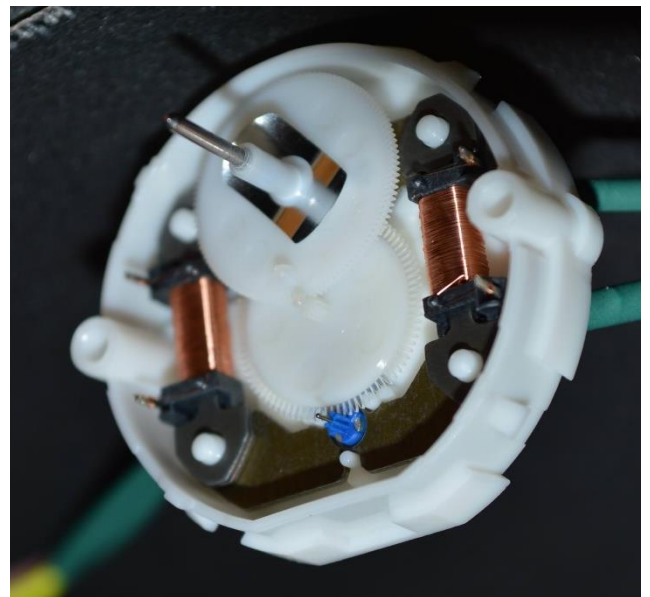
3. Przyrządy pokładowe - Elektronika

Kolejny etap prac to długotrwałe testy mające na celu wybór platformy symulacyjnej, wybór sposobu komunikacji SYMULATOR-PRZYRZĄDY, wybór urządzenia sterującego przyrządami oraz rodzaju napędu wskazówki (silniki krokowe różnego rodzaju lub serwomechanizmy). Ostatecznie zdecydowaliśmy o zastosowaniu:

- Symulatora Microsoft Flight Simulator 2020, jako programu (platformy) symulacji.
- Oprogramowania MobiFlight, jako mostu między symulatorem a Arduino.
- Arduino Mega 2560, jako płytki sterującej kompatybilnej z Mobi Flight
- 5 silników krokowych x27, silnika krokowego x40, silnika krokowego 28BYJ-48 ze sterownikiem ULN2003, oraz enkodera, przełącznika chwilowego oraz 5 przycisków tact switch

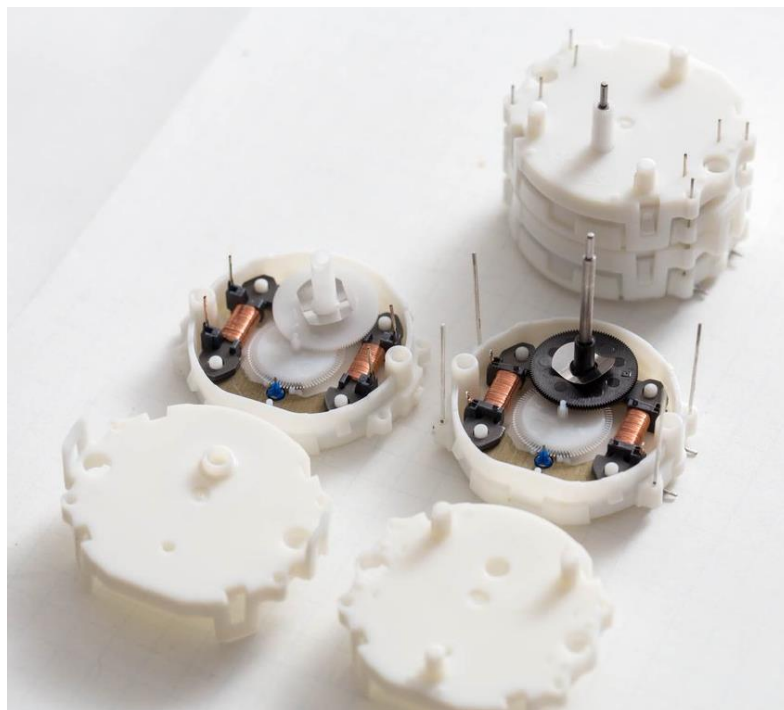
Opis użytych podzespołów:

- Silnik x27 – postanowiliśmy zastosować silnik x27.168 ze względu na jego niewielkie rozmiary. W przyrządach niewymagających obrotu 360°, silnik ten daje możliwość ustawienia wskazówki w pozycji 0 bez dodatkowych czujników (homingu). Silnik daje ponadto możliwość usunięcia ogranicznika, po czym posiada nieograniczony obrót 360°. X27 cechuje się również bardzo wysoką dokładnością, oraz nie wymaga sterownika, co w znaczący sposób pozwala ograniczyć ilość przewodów sterujących.



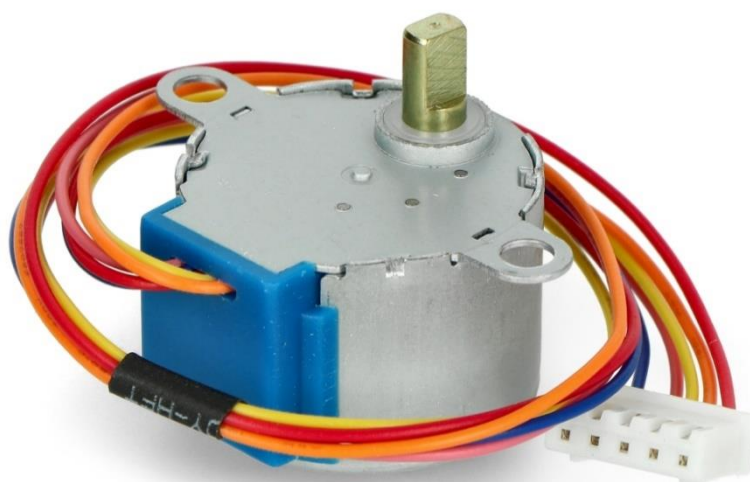
Zdjęcie 1 – Silnik Krokowy X27 (Źródło: <https://forum.arduino.cc/t/gauge-stepper-motor-longevity/488006/3>)

- Silnik x40 – posiada wszystkie zalety ww. jest zbudowany z dwóch x27. Zastosowaliśmy go w wysokościomierzu ze względu na konieczność sterowania dwiema wskazówkami.



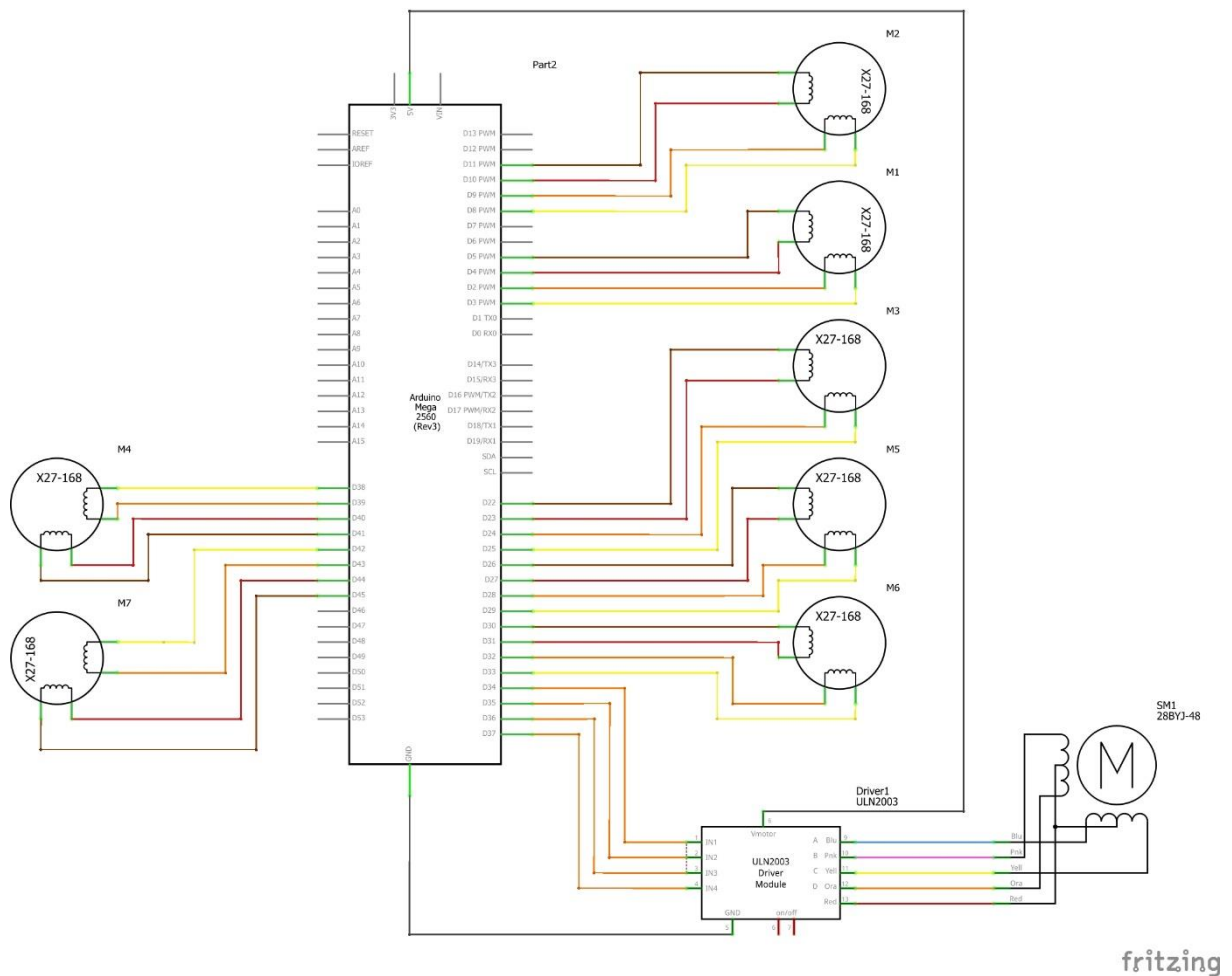
Zdjęcie 2 – Silnik X40 (Źródło: <https://twitter.com/GregDavill/status/1109043444574388225>)

- Silnik 28BYJ-48 – jeden z najbardziej popularnych silników krokowych. Zastosowaliśmy go w busoli ze względu na kształt wału napędowego, niską cenę oraz dużą dokładność. Silnik wymaga zastosowania sterownika, w naszym wypadku jest to sterownik ULN2003.



Zdjęcie 3 – Silnik 28BYJ-48 (Źródło: https://botland.com.pl/img/art/inne/03480_6.jpg)

Kolejnym krokiem było zaprojektowanie schematu podłączenia silników krokowych do Arduino. Użyliśmy do tego celu programu Fritzing.



Kolejnym krokiem było programowanie poszczególnych przyrządów oraz ich testy w programie Mobiflight. Mobiflight to oprogramowanie open source, które pozwala na przetwarzanie danych z symulatora na min. ruch silnika krokowego.

Sposób przetwarzania informacji z symulatora:

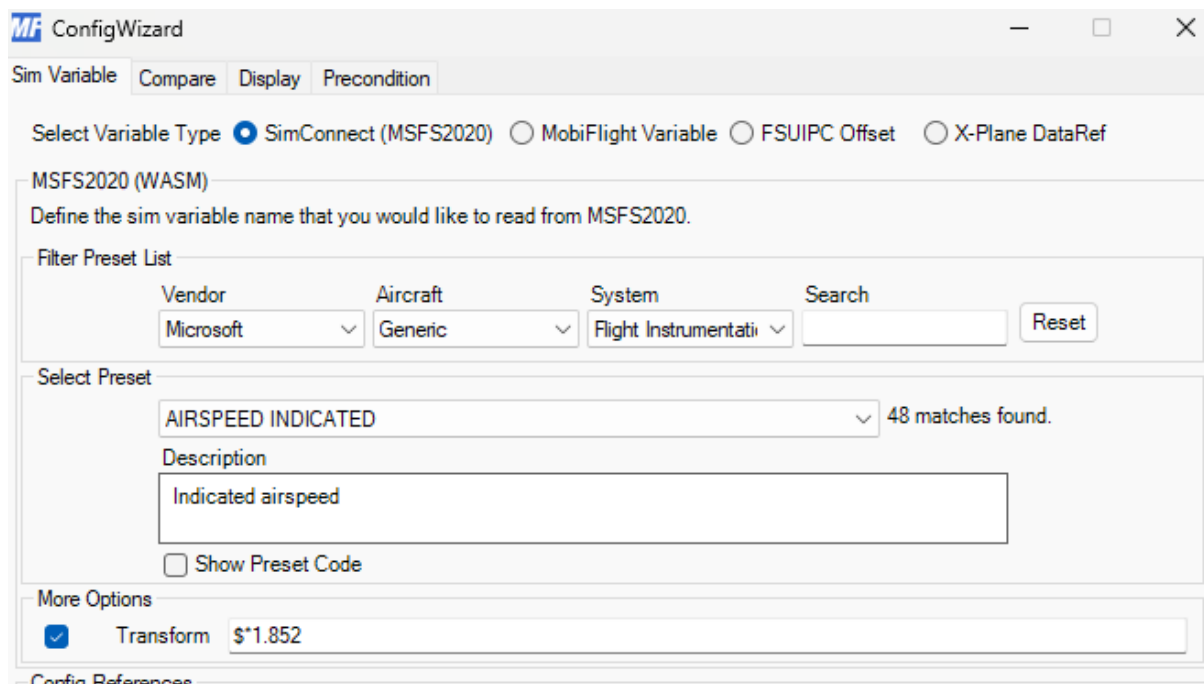
Na początku program poprzez wtyczkę FSUIPC7 pozyskuje z symulatora Microsoft Flight Simulator 2020 zmienne, w naszym wypadku:

- Prędkość [kt]
- Wysokość [ft]
- Prędkość pionową [ft/min]
- Odchylenie chyłomierza poprzecznego (Wartość 0-1, gdzie 1 oznacza pełne wychylenie)

- Kurs magnetyczny [°]

Następnie jednostki symulatora (imperialne) przeliczane są na metryczne (stosowane w polskich szybowcach)

- $kt * 1.852 = km/h$
- $ft * 0.3048 = m$
- $ft/min * 0.00508$



Zdjęcie 4 – zrzut ekranu sim variable z programu Mobiflight prezentujący wybraną zmienną (prędkość IAS) oraz proces jej konwersji z kt na km/h

Kolejnym etapem jest zamiana otrzymanej wartości na ruch silnika krokowego. W celu dobrania odpowiedniej liczby kroków należy przeprowadzić kalibrację. W naszym modelu zastosowaliśmy rozwiązanie polegające na ustawianiu pozycji wskazówki za pomocą stopni jej wychylenia. Program po wpisaniu liczby kroków odpowiadającej pełnemu obrotowi (1450) oraz skali oznaczającej ile jednostek (stopni) zawiera się w pełnym obrocie, pozwala na korzystanie z wygodniejszych niż kroki stopni.

MF ConfigWizard

Sim Variable Compare **Display** Precondition

Choose your display type which is used for output from the list below.

Display type

Choose Output Device ▾

Module MobiFlight Mega/ SN-87a-59c ▾

Use type of Stepper ▾

Display settings

Stepper Predkosc ▾

Sim settings

Display scale 360 steps per revolution

Compass Mode ☐ Enabled

Advanced stepper settings

Full revolution 1450

Max speed 1800

Acceleration 3600

Manual calibration

Move steps

-50 -10 -1 +1 +10 +50

Test settings

Test value -175

Test current settings

Następnie ustaliliśmy interpolacje, a więc przyporządkowaliśmy odpowiednim wartościom (np prędkości) liczbę stopni obrotu wskazówki. Dokonujemy tego na zmontowanym przyrządzie metodą empiryczną, a zatem wpisując kolejne testowe wartości prędkości, obserwujemy ruch wskazówki i dostosowujemy wartość obrotu w stopniach.

Interpolation Settings

☒ Apply interpolation to modify the current value

Assign various input / output value mappings. Linear interpolization will be used between the values.

Input Value	Output Value
0	0
50	40
60	65

Interpolacje dla wszystkich przyrządów naszego symulatora, które jej wymagają przedstawiono na poniższych zdjęciach.

```
<description>Predkosc</description>
<settings msdata:InstanceType="MobiFlight.OutputConfigItem, MFConnector, Version=9.7.0.0"
  <source type="SimConnect" VarType="CODE" Value="(A:AIRSPPEED INDICATED,Knots)" UUID="bb
  <modifiers>
    <transformation active="True" expression="$*1.852" substrStart="0" substrEnd="7" />
    <comparison active="False" value="" operand="" ifValue="" elseValue="" />
    <interpolation active="True">
      <value x="0" y="0" />
      <value x="50" y="40" />
      <value x="60" y="65" />
      <value x="70" y="90" />
      <value x="80" y="115" />
      <value x="90" y="143" />
      <value x="100" y="175" />
      <value x="110" y="205" />
      <value x="120" y="230" />
      <value x="130" y="255" />
      <value x="140" y="280" />
      <value x="150" y="305" />
      <value x="160" y="330" />
      <value x="170" y="355" />
      <value x="180" y="375" />
      <value x="190" y="395" />
      <value x="200" y="418" />
      <value x="210" y="438" />
      <value x="220" y="455" />
      <value x="230" y="475" />
      <value x="240" y="495" />
      <value x="250" y="513" />
      <value x="260" y="525" />
      <value x="270" y="537" />
      <value x="280" y="549" />
      <value x="290" y="561" />
      <value x="300" y="573" />
    </interpolation>
  </modifiers>
</settings>
```

Zdjęcie 5 – Interpolacja prędkościomierza

```
<description>Vario 5ms</description>
<settings msdata:InstanceType="MobiFlight.OutputConfigItem, MFConnector, Version=9.7.0.0, C
  <source type="SimConnect" VarType="CODE" Value="(A:VERTICAL SPEED,Feet per minute)" UUID=
  <modifiers>
    <transformation active="True" expression="$*0.00508 " substrStart="0" substrEnd="7" />
    <comparison active="False" value="" operand="" ifValue="" elseValue="" />
    <interpolation active="True">
      <value x="-5" y="-125" />
      <value x="-4" y="-100" />
      <value x="-3" y="-75" />
      <value x="-2" y="-50" />
      <value x="-1" y="-25" />
      <value x="0" y="0" />
      <value x="1" y="25" />
      <value x="2" y="50" />
      <value x="3" y="75" />
      <value x="4" y="100" />
      <value x="5" y="125" />
    </interpolation>
  </modifiers>
</settings>
```

Zdjęcie 6 – Interpolacja Wariometru 5m/s

```

<description>Vario 30ms</description>
<settings msdata:InstanceType="MobiFlight.OutputConfigItem, MFConnector, Version=9.7.0.0, Cu
  <source type="SimConnect" VarType="CODE" Value="(A:VERTICAL SPEED,Feet per minute)" UUID="
  <modifiers>
    <transformation active="True" expression="$*-0.00508 " substrStart="0" substrEnd="7" />
    <comparison active="False" value="" operand="" ifValue="" elseValue="" />
    <interpolation active="True">
      <value x="-30" y="130" />
      <value x="-28" y="114" />
      <value x="-25" y="-115" />
      <value x="-20" y="102" />
      <value x="-15" y="90" />
      <value x="-10" y="75" />
      <value x="-5" y="43" />
      <value x="-4" y="32" />
      <value x="-3" y="24" />
      <value x="-2" y="16" />
      <value x="-1" y="8" />
      <value x="0" y="0" />
      <value x="1" y="-8" />
      <value x="2" y="-16" />
      <value x="3" y="-24" />
      <value x="4" y="-32" />
      <value x="5" y="-43" />
      <value x="10" y="-75" />
      <value x="15" y="-90" />
      <value x="20" y="-102" />
      <value x="25" y="-115" />
      <value x="30" y="-130" />
    </interpolation>
  </modifiers>
</settings>

```

Zdjęcie 7 – Interpolacja Wariometru 30m/s

```

<description>Wysokosc1000</description>
<settings msdata:InstanceType="MobiFlight.OutputConfigItem, MFConnector, Version=9.7.0.0,
  <source type="SimConnect" VarType="CODE" Value="(A:INDICATED ALTITUDE,Feet)" UUID="3110
  <modifiers>
    <transformation active="True" expression="$*0.3048" substrStart="0" substrEnd="7" />
    <comparison active="False" value="" operand="" ifValue="" elseValue="" />
    <interpolation active="True">
      <value x="0" y="0" />
      <value x="1000" y="-37" />
      <value x="2000" y="-70" />
      <value x="3000" y="-110" />
      <value x="4000" y="-145" />
    </interpolation>
  </modifiers>
</settings>

```

Zdjęcie 8 – Interpolacja wskazówki tysięcy wysokościomierza

```

<description>Wysokosc100</description>
<settings msdata:InstanceType="MobiFlight.OutputConfigItem, MFConnector, Version=9.7.0.0
  <source type="SimConnect" VarType="CODE" Value="(A:INDICATED ALTITUDE,Feet)" UUID="311
  <modifiers>
    <transformation active="True" expression="$*0.304" substrStart="0" substrEnd="7" />
    <comparison active="False" value="" operand="" ifValue="" elseValue="" />
    <interpolation active="True">
      <value x="0" y="0" />
      <value x="100" y="40" />
      <value x="200" y="75" />
      <value x="300" y="110" />
      <value x="400" y="145" />
      <value x="500" y="180" />
      <value x="600" y="215" />
      <value x="700" y="250" />
      <value x="800" y="285" />
      <value x="900" y="320" />
      <value x="1000" y="360" />
      <value x="1100" y="395" />
      <value x="1200" y="430" />
      <value x="1300" y="465" />
      <value x="1400" y="500" />
      <value x="1500" y="535" />
      <value x="1600" y="570" />
      <value x="1700" y="605" />
      <value x="1800" y="640" />
      <value x="1900" y="675" />
      <value x="2000" y="715" />
      <value x="2100" y="755" />
      <value x="2200" y="790" />
      <value x="2300" y="825" />
      <value x="2400" y="860" />
      <value x="2500" y="895" />
      <value x="2600" y="930" />
      <value x="2700" y="965" />
      <value x="2800" y="1000" />
      <value x="2900" y="1035" />
      <value x="3000" y="1070" />
    </interpolation>
  </modifiers>
</settings>

```

Zdjęcie 9 – Interpolacja wskazówki setek wysokościomierza

```

<description>Kulka</description>
<settings msdata:InstanceType="MobiFlight.OutputConfigItem, MFConnector, Version=9.
  <source type="SimConnect" VarType="CODE" Value="(A:TURN COORDINATOR BALL,Position
  <modifiers>
    <transformation active="False" expression="$" substrStart="0" substrEnd="7" />
    <comparison active="False" value="" operand="" ifValue="" elseValue="" />
    <interpolation active="True">
      <value x="-1" y="-30" />
      <value x="0" y="0" />
      <value x="1" y="30" />
    </interpolation>
  </modifiers>
</settings>

```

Zdjęcie 10 – Interpolacja chyłomierza poprzecznego

W efekcie symulator posiada 6 funkcjonalnych przyrządów:

- Prędkościomierz wskazujący prędkość IAS* szybowca [km/h] w zakresie 0-300km/h

- Wariometr wskazujący prędkość pionową (wznoszenia lub opadania)[m/s] w zakresie $\pm 5\text{m/s}$
- Wariometr wskazujący prędkość pionową [m/s] w zakresie $\pm 30\text{m/s}$
- Wysokościomierz z dwiema wskazówkami wskazujący wysokość [m] w zakresie 0-3000m**
- Busolę wskazującą kurs magnetyczny [$^{\circ}$] w zakresie 360°
- Chyłomierz poprzeczny, wskazujący koordynację zakrętu szybowca (kąt pomiędzy wektorem przyspieszenia a osią szybowca)

* IAS – Indicated Airspeed, prędkość przyrządowa (wskazywana)

** Aktualnie uznaliśmy za nieuzasadnioną komplikację wpisywanie interpolacji dla wyższych wysokości, ze względu na to że nie przewidujemy symulowania lotów falowych. W przypadku chęci zasymulowania lotów na wyższych wysokościach, wystarczy wydłużyć zakres wskazywanych wartości poprzez dopisanie kolejnych interpolacji.

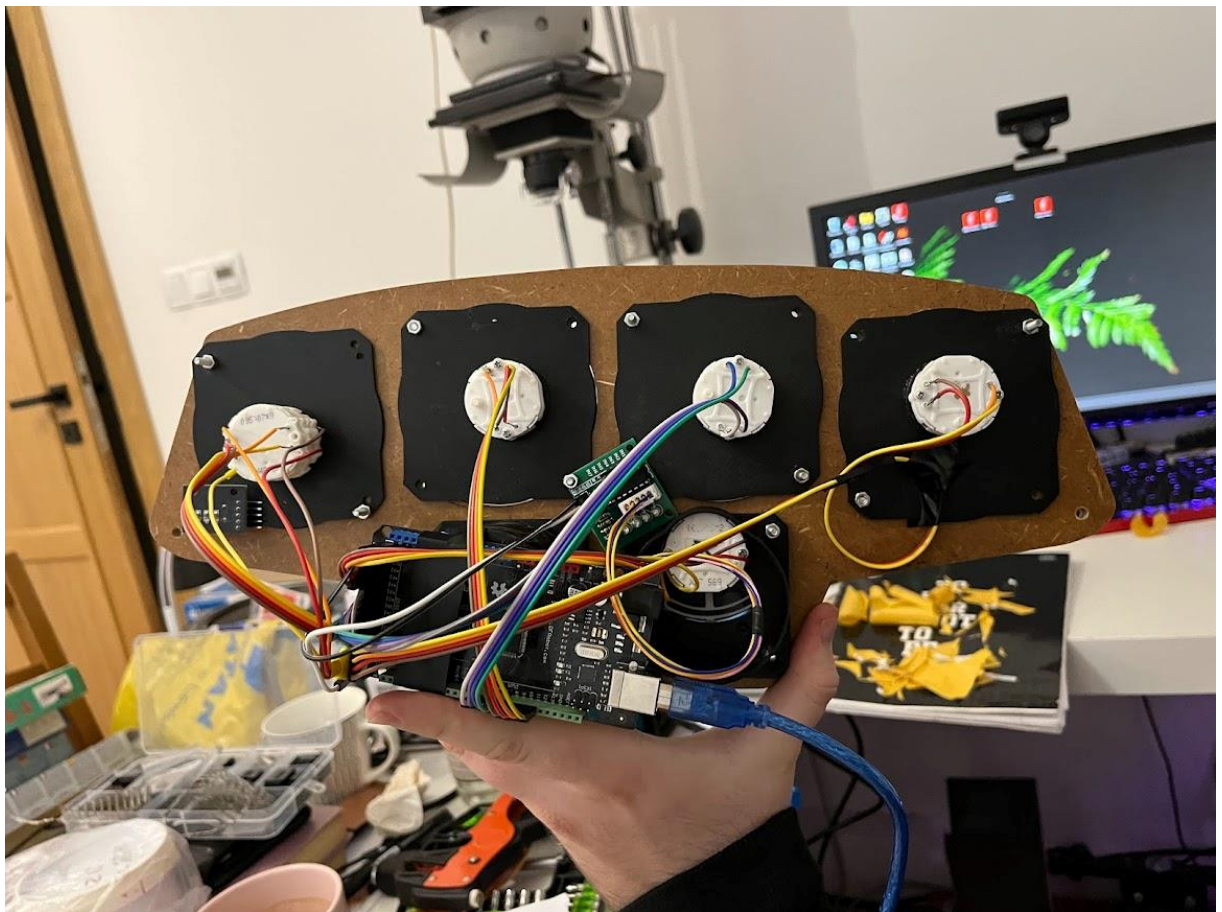
Aktualnie symulator nie symuluje zakrętomierza, ze względu na to iż nie występuje on w większości szybowców dostępnych w symulatorze Microsoft Flight Simulator 2020, a w realnych lotach szybowcowych nie jest często wykorzystywany ze względu na to iż nie jest przyrządem niezbędnym do wykonania lotów. Jeżeli jednak zaistniałaby potrzeba zasymulowania zakrętomierza, wystarczy podłączyć oraz skalibrować jego silnik.

4. Przyrządy Pokładowe - Efekt Końcowy

Końcowym etapem tworzenia deski przyrządów pokładowych było malowanie elementów, zaprojektowanie oraz wklejenie skali i oznaczeń oraz wykonanie z plexi „szkiełek” przyrządów przy pomocy wyfrezowanego na frezarce CNC kopyta.



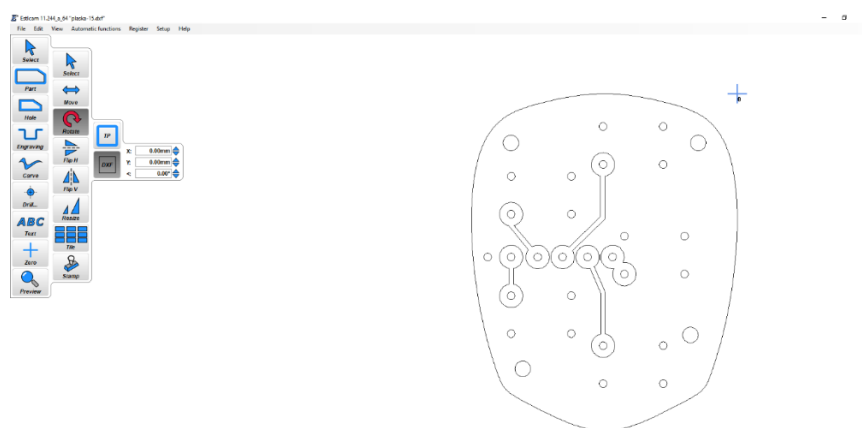




5. Drażek sterowy

W trakcie prac, postanowiliśmy zastąpić oryginalny drażek sterowy szybowca Pirat bardziej nowoczesnym, z wieloma przyciskami, które będą przydatne w symulatorze. Wykorzystaliśmy gotowy model drażka open vario z serwisu thingiverse (<https://www.thingiverse.com/thing:1818509>). Następnie został on przez nas wydrukowany w technologii FDM używając PLA.

Kolejnym krokiem było zaprojektowanie płytki z przyciskami tact switch oraz wyprowadzeniami na przewody. Płytką została zaprojektowana w programie Autodesk Fusion 360, a następnie przy pomocy programu MACH 3 wyfrezowana na frezarce CNC.

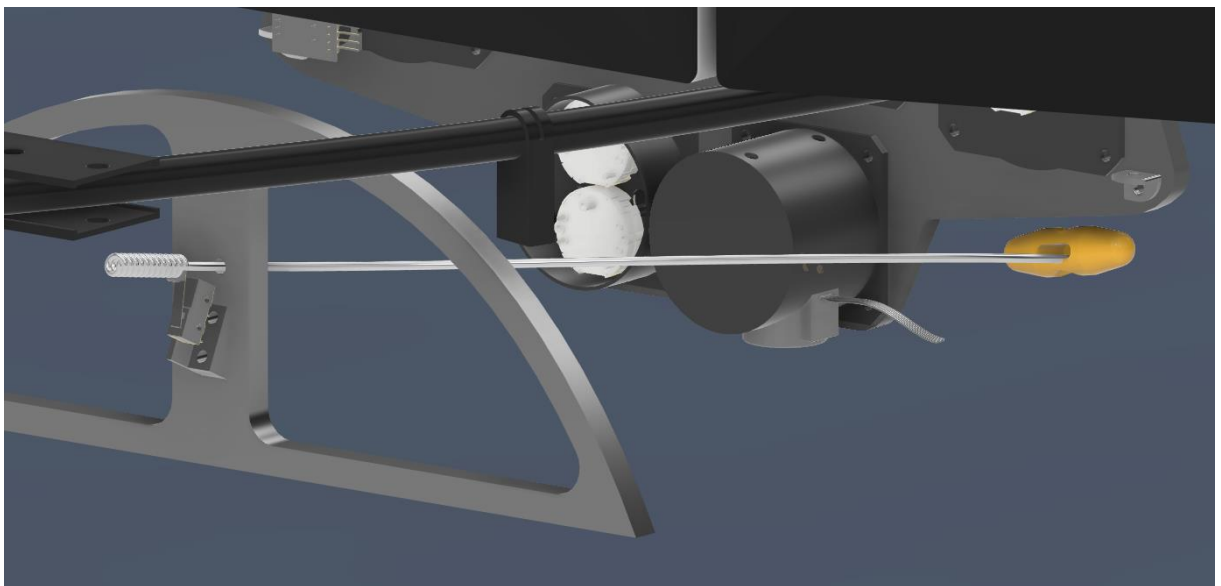


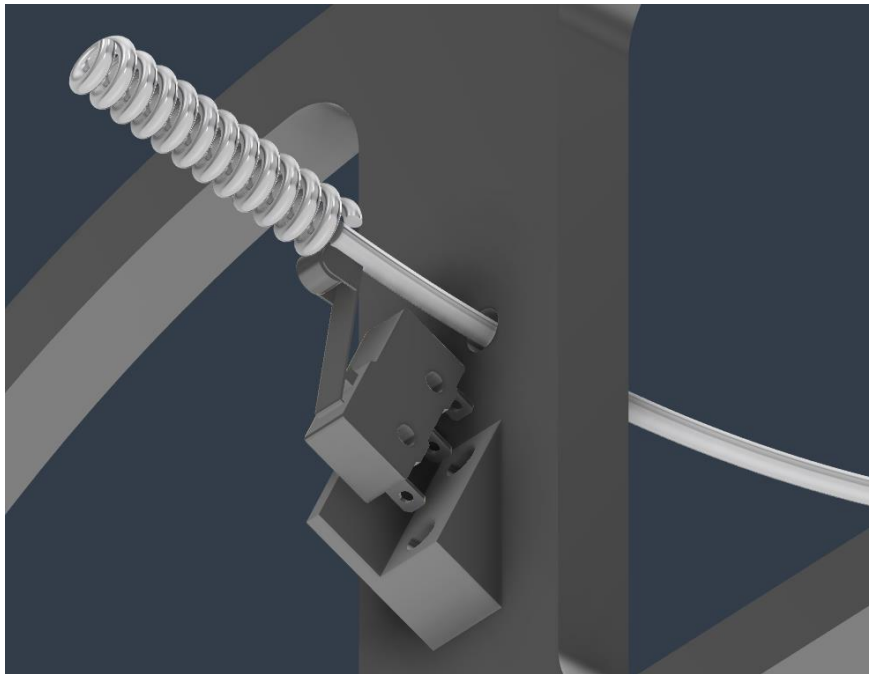




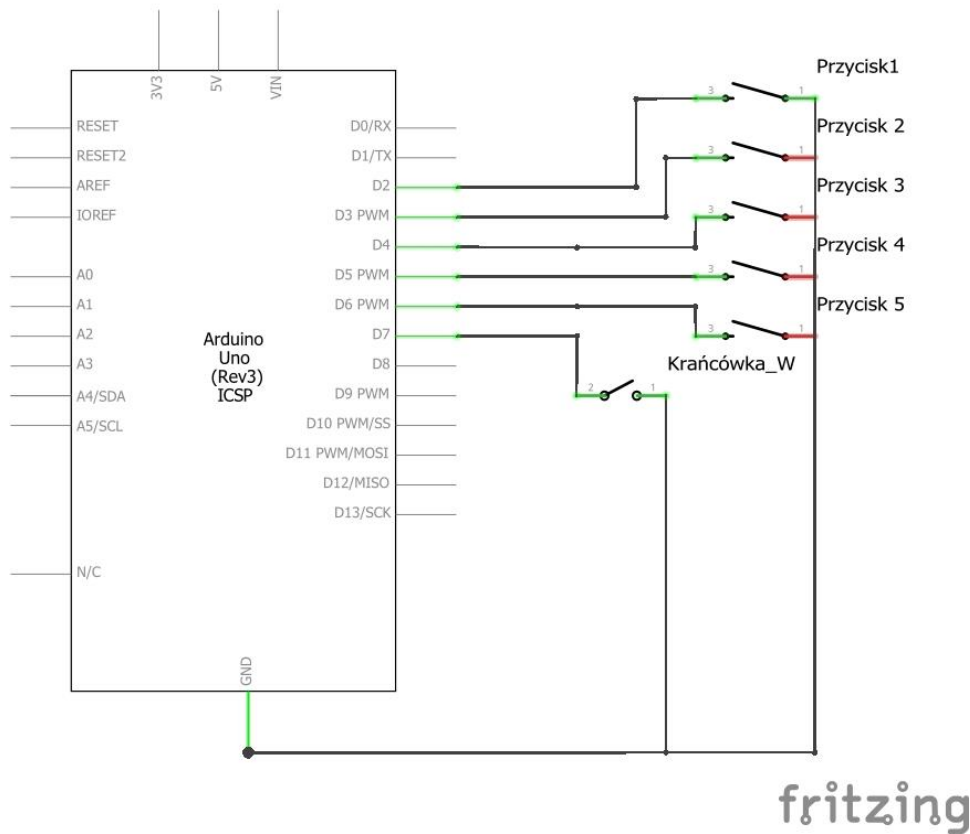
6. Dźwignia wyczepu

System dźwigni wyczepu szybowca oparliśmy na oryginalnym. Sprężyna zwrotna aktywuje włącznik krańcowy, który jest zamocowany do wręgi kadłuba nr 2.





Przyciski umieszczone na drążku sterowym, wraz z włącznikiem krańcowym wyczepu są podłączone pod Arduino Uno. Schemat podłączenia przedstawiono na poniższym schemacie.



Do przekazywania informacji z arduino wykorzystaliśmy program mobiflight. Po dodaniu w nim poszczególnych przycisków digital input, przesterowaliśmy każdy przycisk w taki sposób aby był możliwy do przypisania w symulatorze, korzystaliśmy przy tym także z oprogramowania vjoy. Aktualnie każdemu przyciskowi można w ten sposób przypisać dowolną funkcję, co pozwala na pełną personalizację.

Włącznik	Przycisk	Przycisk	Przycisk
☒	Wyczep	MobiFlight Uno	Button
☒	Joystick_btn_1	MobiFlight Uno	Button 1
☒	Joystick_btn_2	MobiFlight Uno	Button 2
☒	Joystick_btn_3	MobiFlight Uno	Button 3
☒	Joystick_btn_4	MobiFlight Uno	Button 4
☒	Joystick_btn_5	MobiFlight Uno	Button 5

Input settings

On Press On Release

Action Type MobiFlight - Virtual Joystick input (vJk) Copy Paste

vJoy virtual Joystick Settings

Joystick ID 1

Button Nr 1

Axis --

Button ☒ on ☐ off

Value 0

Supports input value (@) and placeholders (\$, #, etc.)

Input settings

On Press On Release

Action Type MobiFlight - Virtual Joystick input (vJk) Copy Paste

vJoy virtual Joystick Settings

Joystick ID 1

Button Nr 1

Axis --

Button ☐ on ☒ off

Value 0

Supports input value (@) and placeholders (\$, #, etc.)

7. System sterowania

System sterowania został przez nas usprawniony w porównaniu do poprzedniej wersji.

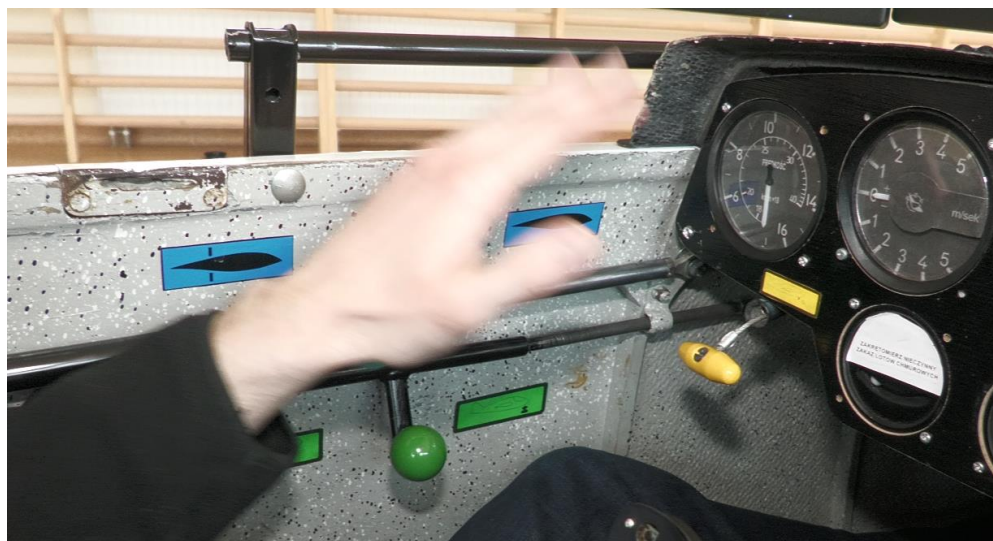
Zastosowaliśmy sprężyny poprawiające realizm sił działających na ster. Zdjęcie poniżej przedstawia nowy system (lewy) oraz dotychczasowy (prawy).



Poprzez regulację sterów oryginalnymi śrubami udało nam się praktycznie w 100% wyeliminować luzy na drążku.

W celu zapewnienia jeszcze większej imersji, oraz poprawy realizmu sterowania, dokonaliśmy dalece idących zmian w działaniu popychacza hamulców aerodynamicznych, poprzez zastosowanie siłownika symulującego siłę potrzebną do odblokowania hamulców oraz „wyssania” przez strugi powietrza opływające skrzydło.





Dopracowaliśmy również działanie pedałów orczyka. Poprzez regulację udało nam się uzyskać efekt powrotu do neutrum, który w realnym szybowcu jest wymuszony opływem strug powietrza.

Przywróciliśmy do działania oryginalną regulację pedałów orczyka co pozwala na dostosowanie ich odległości do wzrostu pilota.



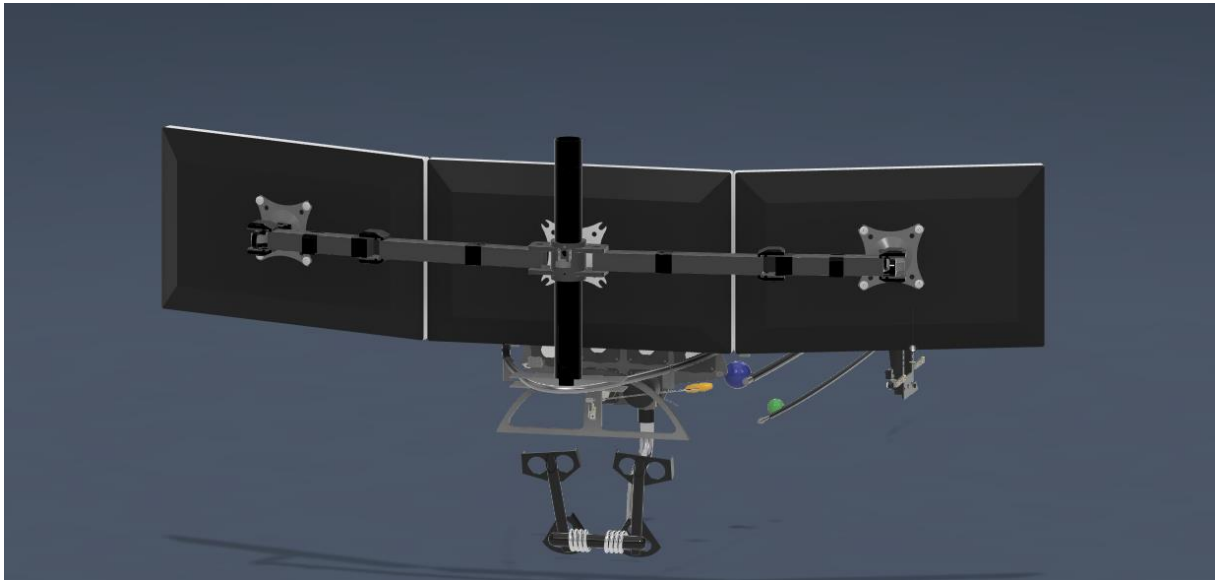
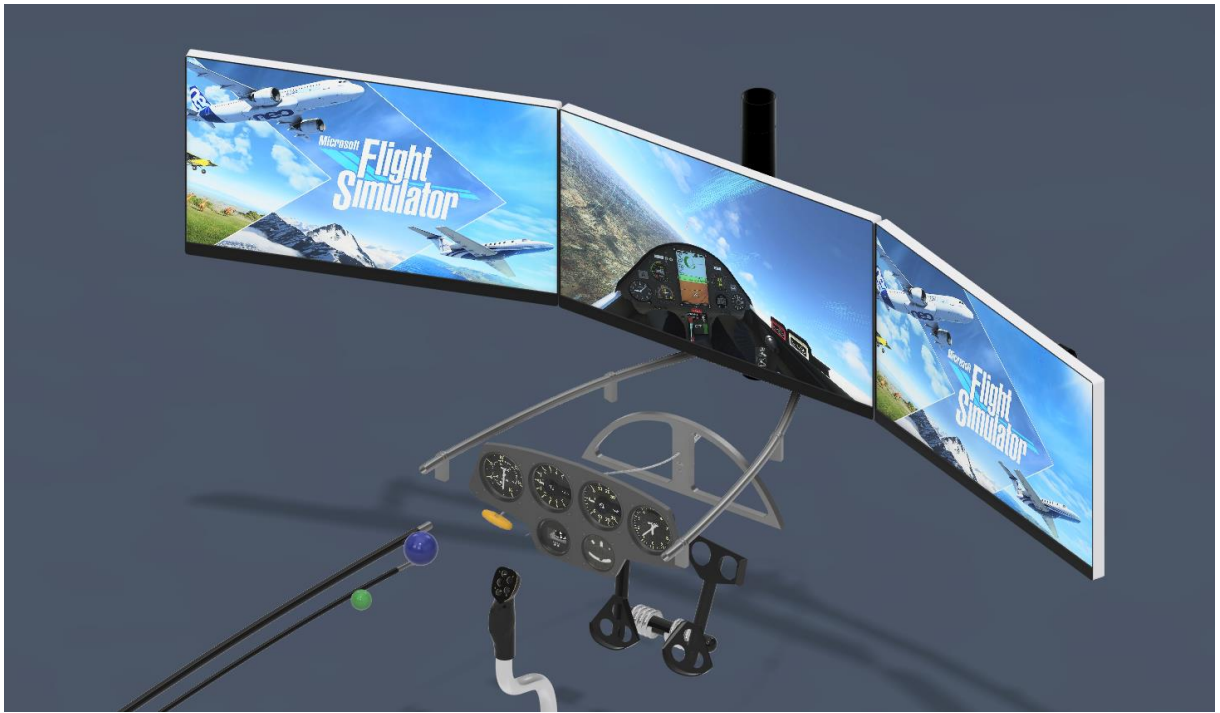
Poprawiliśmy również działanie trymera. Aktualnie jego zakres wychyleń oraz sposób działania odzwierciedla jego realny odpowiednik.



8. Rama pod uchwyt monitorów

Wykonaliśmy oraz zamocowaliśmy do wręg 2 i 3 ramę uchwytu na monitory, przez co są one stabilne i równomiernie obciążają konstrukcję.





9. Wykończenie wizualne

Dokonałiśmy znacznych poprawek lakierniczych, w wielu miejscach kadłub został pomalowany na nowo.



Przywróciliśmy również oryginalne napisy „Pirat”. Naklejki zostały wykonane na nasze zamówienie, a następnie precyzyjnie przez nas przyklejone.



W naszym projekcie zadbaliśmy o najdrobniejsze detale min. poprzez umieszczenie na kadłubie oryginalnych naklejek oraz informacji eksploatacyjnych, odpowiednio je również postarzyliśmy aby wyglądały naturalnie. Wydrukowaliśmy także w technologii SLA oraz pomalowaliśmy nowe kulki trymera i hamulców aerodynamicznych a także uchwyt wycze-
pu.

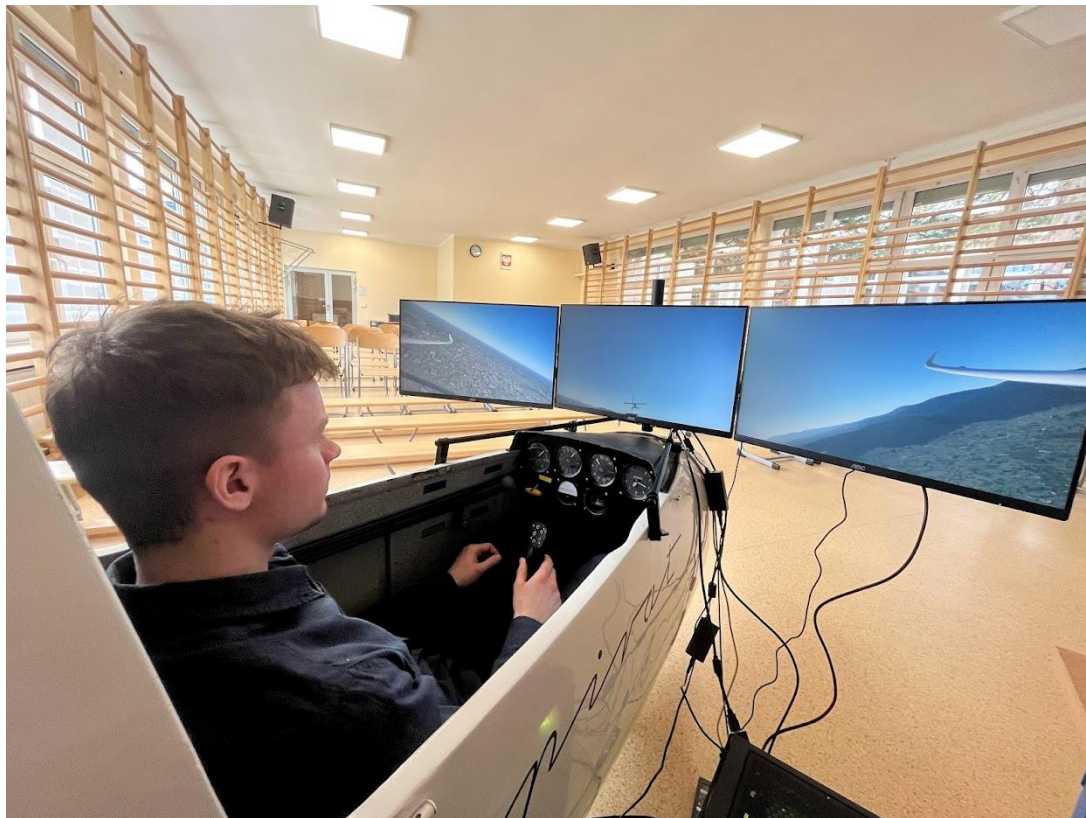


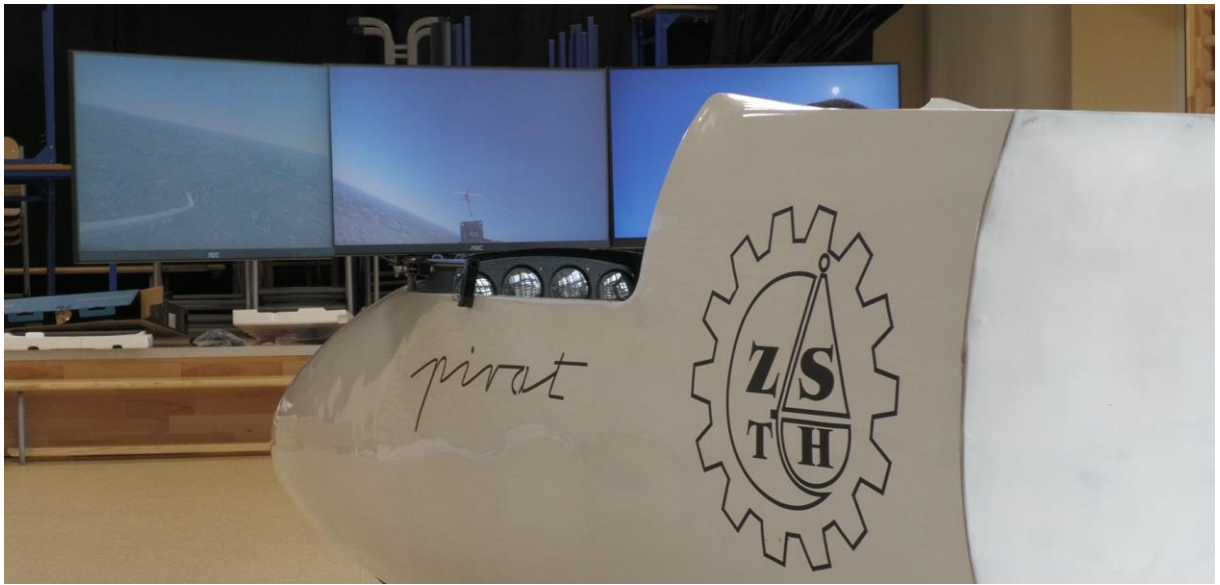
Zadbaliśmy także o komfort osób użytkujących symulator, min. poprzez wykonanie nowej tapicerki dzięki uprzejmości firmy SZD Allstar. Aktualnie tapicerka jest taka sama jak w nowoczesnych polskich konstrukcjach SZD takich jak szybowce Junior, Perkoz lub Nexus.



10. Podsumowanie oraz Efekt Końcowy

Poniższe zdjęcia przedstawiają końcowy wygląd naszego symulatora.





Link do prezentacji wideo:

bit.ly/Symulator_Pirat