

ZERVIDJANNA

Dokumentacja rozwoju głowy robota

Prototyp chwytaka szczękowego, widzenie maszynowe, sterownik serwa i testy integracyjne

Cel dokumentu

Zebranie w jednym miejscu procesu projektowania prototypowej głowy Zervidjanny: od koncepcji mechanicznej, przez przetwarzanie obrazu i rozpoznawanie obiektów, aż po fizyczny model szczęki oraz pierwszy sterownik serwomechanizmu. Tekst zachowuje luźny, warsztatowy charakter notatek z pracy nad projektem.

Status: prototyp głowy ze sklejki, testy widzenia maszynowego, sterowanie serwem oraz integracja z reakcją szczęki



| 0. Wstęp

Projekt głowy Zervidjanny wyrósł z połączenia kilku tematów: robotyki, elektroniki, widzenia maszynowego i mechaniki inspirowanej naturą. Założenie było proste tylko na papierze: stworzyć moduł, który będzie wyglądał jak głowa łani, a jednocześnie będzie potrafił rozpoznawać obiekty, reagować na otoczenie i pracować jako chwytak. Brzmi jak plan idealny do spokojnego wieczoru przy komputerze... czyli oczywiście plan, który bardzo szybko zaczął obrastać problemami technicznymi.

| 1. Przygotowanie zaplecza i zasady pracy

Na początku uporządkowałem zaplecze projektowe, założenia bezpieczeństwa i listę elementów, które będą potrzebne do dalszej pracy. W praktyce oznaczało to przejrzenie narzędzi, elektroniki, dostępnych modułów kamer, serwomechanizmów, stanowisk testowych oraz oprogramowania. Ten etap był mniej widowiskowy niż cięcie laserem czy uruchamianie kamery, ale bez niego późniejsze składanie głowy mogłoby zamienić się w poszukiwanie jednej śrubki przez pół dnia.

| 2. Koncepcja systemu głowy robota

Przyjąłem założenie, że większą wartość ma realizacja kompletnego mini-systemu niż tworzenie serii luźnych testów bez wspólnego celu. Taki projekt obejmuje cały cykl inżynierski: koncepcję, dobór technologii, modelowanie, prototypowanie, programowanie, testy i poprawki. Właśnie takie podejście najlepiej pasuje do Zervidjanny, bo głowa robota nie jest pojedynczym elementem, ale punktem styku mechaniki, elektroniki i oprogramowania.

Po analizie możliwych kierunków rozwoju uznałem, że dobrym celem będzie system do detekcji i interakcji z obiektami. Przykładowo może on rozpoznawać butelkę, kubek albo osobę, a następnie przekazać informację do mechanizmu szczęki. W wersji laboratoryjnej może to wyglądać jak zabawa w łapanie puszek, ale technicznie jest to już pierwszy krok do głowy robota, która wie, na co patrzy i próbuje na to zareagować.

Ponieważ równolegle rozwijany jest projekt robota kroczącego inspirowanego sylwetką zwierzęcia, naturalnym krokiem było połączenie systemu rozpoznawania obiektów z koncepcją głowy Zervidjanny. W rezultacie celem stało się zaprojektowanie modułu integrującego widzenie maszynowe, układ wykonawczy i elementy sensoryczne. Taka głowa jest trudniejsza niż klasyczne ramię z chwytakiem, ale też znacznie ciekawsza - bo musi być jednocześnie mechanizmem, czujnikiem i trochę „twarzą” robota.

Co istotne, przyjęta architektura systemu zakłada uniwersalność oprogramowania - bez względu na zastosowaną platformę (czy będzie to „głowa” robota, czy np. ramię manipulatora), algorytmy detekcji, rozpoznawania oraz interakcji z obiektami pozostają praktycznie niezmiennie. Różnice sprowadzają się jedynie do warstwy wykonawczej - zamiast segmentów szyi i szczęki można zastosować odpowiednik w postaci ramienia i chwytaka. Takie rozwiązanie niesie ze sobą szereg zalet:

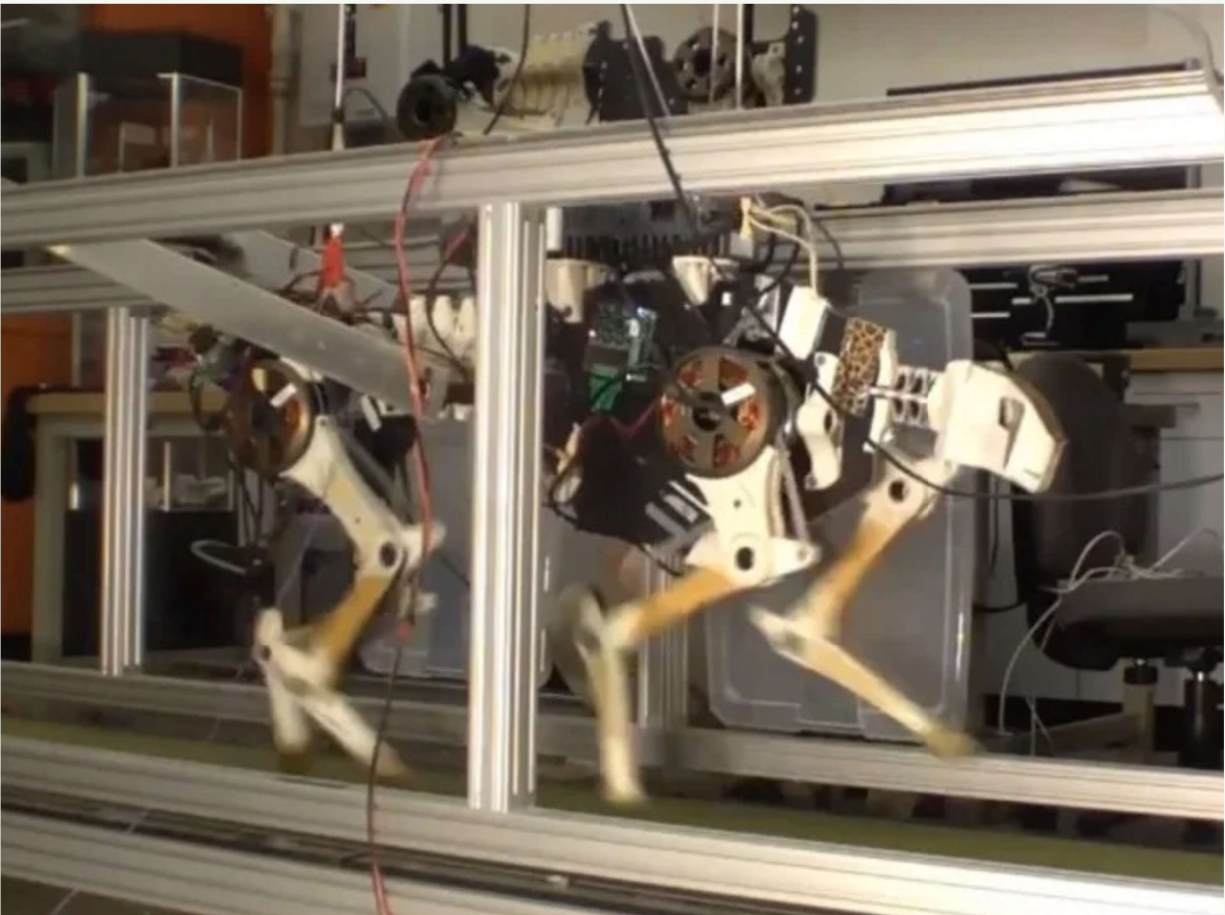
- Modularność i skalowalność - ten sam system może być wdrażany w różnych konfiguracjach sprzętowych, co zwiększa jego uniwersalność.
- Ponowne użycie kodu - algorytmy mogą być adaptowane do różnych aplikacji bez konieczności pisania ich od nowa.
- Testowalność i elastyczność - system można najpierw testować w wersji laboratoryjnej, a następnie przenieść na fizycznego robota bez większych zmian w kodzie.
- Potencjał komercjalizacji - możliwość łatwego dopasowania systemu do różnych potrzeb użytkowników zwiększa jego wartość rynkową.

| 3. W poszukiwaniu Świętego Graala

Ten etap poświęciłem niemal w całości na poszukiwanie istniejących rozwiązań technicznych, które mogłyby posłużyć jako baza lub inspiracja do systemu rozpoznawania i interakcji z obiektami. Chodziło o przeanalizowanie, jak podobne zagadnienia rozwiązyli inni twórcy - zarówno w środowiskach akademickich, hobbystycznych, jak i komercyjnych. Założenie niby proste: głowa robota ma przenosić rozpoznane obiekty i możliwie mocno nawiązywać do natury, aby kiedyś mogła trafić do zwierzopodobnego robota o nazwie Zervidjanna.

Choć początkowo wydawało się to prostym zadaniem - wystarczy przecież poszukać projektów typu „object detection + robotic arm” - rzeczywistość szybko zweryfikowała mój optymizm. Spędziłem kilka godzin przeglądając różne repozytoria na GitHubie, wpisy na forach technicznych, blogi i artykuły. Niestety, większość znalezionych projektów była albo bardzo uproszczona i edukacyjna, albo zbyt skomplikowana i nieudokumentowana w sposób umożliwiający ich praktyczne wykorzystanie jako punktu wyjścia. Co więcej żaden nie wyglądał nawet podobnie do robotycznej głowy zwierzęcia, z pominięciem kilku dużych projektów firm takich jak Festo czy Boston Dynamics, ale i w tym przypadku nacisk kładzie się na kinematykę i ruch a nie samą głowę. Najciekawszy dla mnie projekt jeśli chodzi o robota kroczącego do którego dokumentacji nie sposób dotrzeć załączam poniżej w postaci fotki z internetu.

Pomimo dużego zaangażowania i poświęcenia całego wieczoru na analizę dostępnych materiałów, nie znalazłem żadnego projektu, który spełniałby moje oczekiwania i mógłby posłużyć jako gotowa baza. Z jednej strony było to frustrujące, ale z drugiej - utwierdziło mnie to w przekonaniu, że warto stworzyć coś własnego, od podstaw, dopasowanego do założonego celu i jednocześnie innowacyjnego. Być może to właśnie moja koncepcja stanie się w przyszłości inspiracją dla innych...



Cheetah – robot z zaplecza MIT.

4. Fantazja górą

Ten dzień poświęciłem intensywnemu rozważaniu założeń projektowych dla części mechanicznej mojego systemu - a konkretnie głowy robota, która będzie nie tylko nośnikiem dla elementów wykonawczych, ale też jego że tak powiem osobowością.

Na wstępie przeanalizowałem funkcję, która ma być osią projektu, czyli szczękę - pełniącą rolę chwytaka do interakcji z obiektami. Ale szybko zrozumiałem, że jeśli to ma być głowa robota, trzeba pomyśleć znacznie szerzej i uwzględnić również inne funkcje, które mogą być potrzebne teraz lub w przyszłości. Na listę kluczowych modułów trafiły więc m.in.:

- moduł słuchowy (mikrofony kierunkowe, praca stereofoniczna, docelowo lokalizacja dźwięku),
- moduł wzroku (kamery stereo dla uzyskania głębi obrazu, oraz z możliwością sterowania kątem widzenia),
- głośnik (komunikacja głosowa lub dźwiękowa interakcja z otoczeniem),
- czujniki odległości (np. ToF lub podczerwieni do wykrywania przeszkód i bliskich obiektów, które są poza obszarem głównych kamer),
- czujniki gazów (dla celów bezpieczeństwa lub inspekcji, np. amoniak, dym, metan),
- żyroskop i akcelerometr (do stabilizacji pozycji lub wykrywania uderzeń, choć to pewnie lepiej umieścić docelowo w ciele robota, hmm...),
- miejsce na język (teoretycznie niepotrzebne, ale skoro już wszystko jest to niech i język może pokazać :)

Z biegiem dnia fantazja poniosła mnie jeszcze dalej - postanowiłem, że głowa mogłaby mieć system obronny, np. wysokonapięciowy generator podłączony do zębów szczęki. Taki system mógłby teoretycznie odstraszać intruzów, przeszkadzające zwierzęta domowe, albo... robić wrażenie na publiczności :). W ramach tego samego pomysłu powstała też koncepcja dodania elektrod do ładowania, tak by robot mógł przygryźć stację dokującą i uzupełnić energię.

Pod koniec tego etapu miałem już z grubsza określoną listę wymagań funkcjonalnych oraz pierwsze pomysły na rozmieszczenie modułów i ograniczenia przestrzenne. Było też jasne, że projekt nie będzie prosty i nie zakończy się na jednym eleganckim szkicu - ale za to na pewno nie będzie nudny. Złożoność zadań sprawia, że każdy kolejny krok daje nowe problemy, a więc również nowe okazje do nauki.

Wierzę, że właśnie takie zadania, wymagające ciągłego doskonalenia i wychodzenia ze strefy komfortu, są najlepszym sposobem na rozwój i zdobycie cennego doświadczenia, które może zaprocentować w przyszłości. (a chyba, raczej, na pewno zaowocuje słabnącym wzrokiem od komputerów i nieprzespanymi nocami :))

5. Co ma piernik do wiatraka

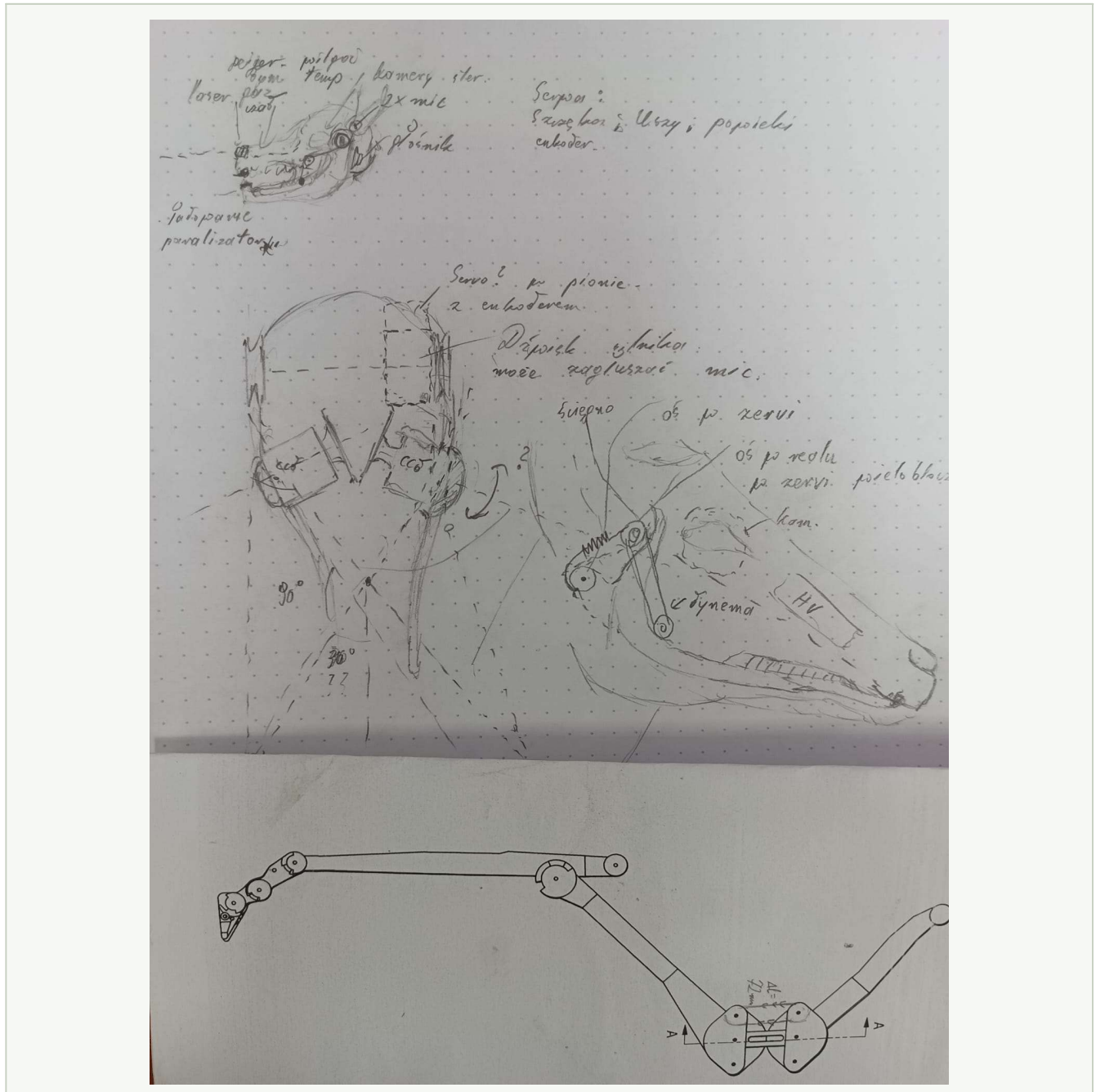
Po przespaniu się z dotychczasowymi pomysłami, dotarło do mnie coś istotnego - choć mój projekt brzmi momentami jak szalony, to jednak nie jest on nierealny. Wymaga on po prostu bardzo starannego planowania, priorytetyzacji i kreatywności w zagospodarowaniu przestrzeni.

Pierwotny projekt Zervidjanny zakładał stworzenie robota kroczącego o postaci łani daniela. Siłą rzeczy jego głowa musi być anatomicznie i wizualnie spójna z całością. Oznacza to szereg ograniczeń konstrukcyjnych, estetycznych i funkcjonalnych, które trzeba uwzględnić od samego początku.

Dobrze że mamy sobotę :) - Ten dzień poświęciłem więc na wykonanie wstępnego szkicu głowy, na którym umieściłem wszystkie największe gabarytowo planowane elementy czyli mikrofony oraz kamery. Sporo czasu zajęło także szukanie małych obiektywów kamer które mają małą dystorsję aby znać przybliżone wymiary, czy też mikrofonów które nie zbierają wszystkich śmieci z otoczenia, bo później trzeba to filtrować w oprogramowaniu. Od razu rzuciło mi się w oczy, że dostępne miejsce jest bardzo ograniczone, a każdy milimetr musi zostać wykorzystany z chirurgiczną precyzją.

Mimo że pierwszy prototyp głowy ogranicza się głównie do działania szczęki, już na starcie rozpocząłem analizę jej możliwości ruchowych. Od razu pojawiły się konflikty przestrzenne i pytania związane z momentami obrotowymi serwomechanizmów. Problemem okazały się także kamery, które zajmują cenne miejsce. Głowa ma być efektowna wizualnie, ale też zdolna do precyzyjnej pracy: chwytania, odchylania, a w przyszłości być może także ekspresji i wizualizacji mowy.

Na poniższym rysunku celowo umieściłem dawno temu zaprojektowaną część tylnej nogi Zervidjanny razem z rysunkiem głowy - pytanie tylko co ma noga do głowy? Jak widać na poniższym rysunku części nogi, w stawie kolanowym wykorzystałem dwustronny bloczek celem zwiększenia momentu silnika głównego odpowiedzialnego za poruszanie tym stawem, podobnie jak to się stosuje w dźwigach. Tą samą koncepcję chcę użyć do zaciskania szczęki, dzięki czemu silnik o mniejszym momencie obrotowym będzie w stanie dość mocno chwycić zadany obiekt. Część odpowiedzialna poglądowo za ten mechanizm przedstawiona jest na szkicu samej głowy w rzucie bocznym, z czego sprężyna utrzymuje naprężenie linki z dyneemy.



Szkic prototypu głowy robota która służyć ma docelowo jako chwytak, widoczny wielobloczek w stawie kolanowym wymyślony lata temu do starego projektu, tu jako koncepcja dla mechanizmu zamykania szczęki z większą siłą.

| 6. Nie-zabójcze ugryzienie

Ten etap w całości poświęciłem analizie szczęki robota - zarówno pod kątem zakresu ruchów, jak i funkcjonalności chwytaka. Zacząłem od geometrii działania: klasyczne otwieranie i zamykanie to jedno, ale pojawił się także pomysł niewielkiego ruchu bocznego, przypominającego naturalną pracę żuchwy u zwierząt. Taki ruch mógłby pełnić rolę dodatkowej interakcji lub ekspresji.

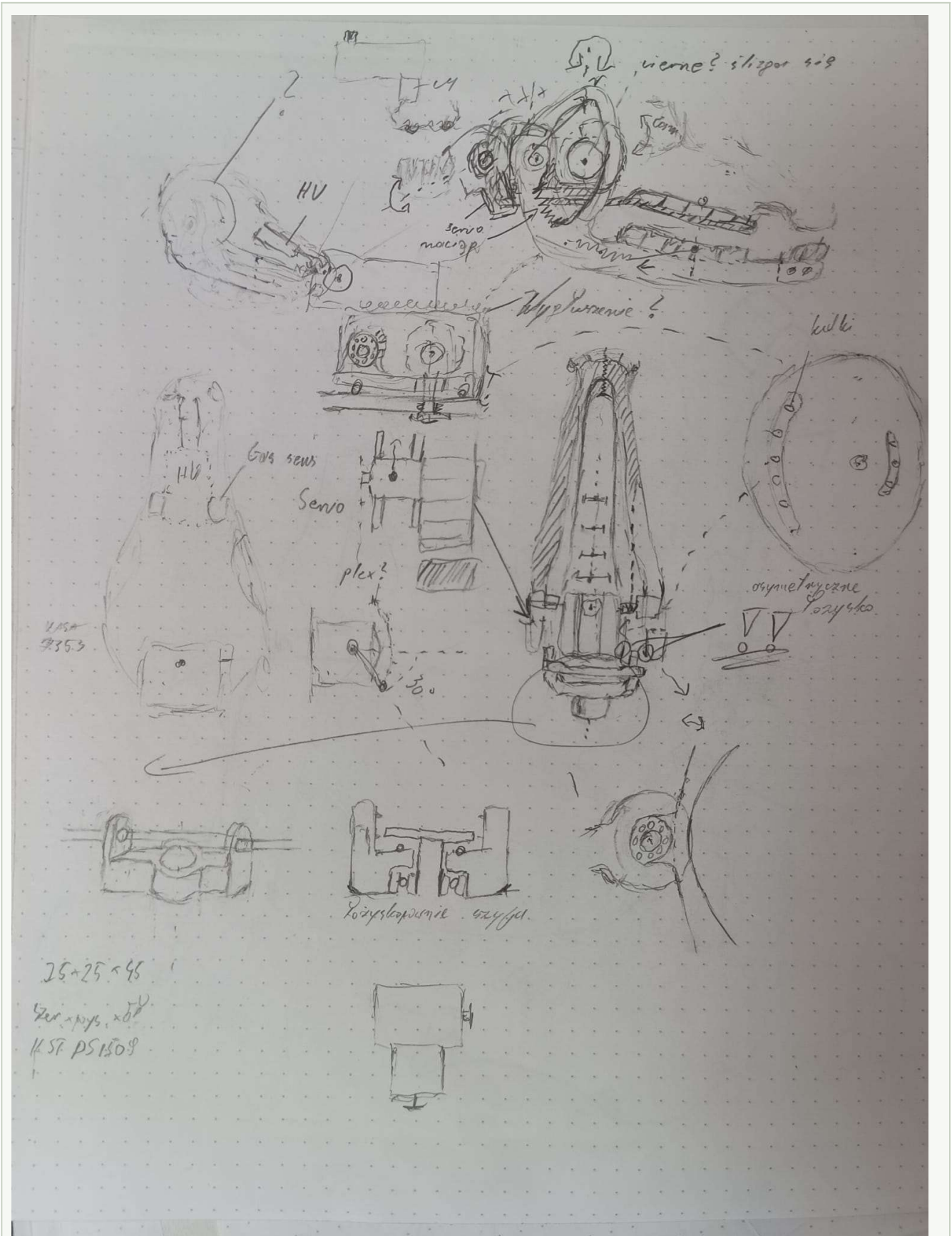
Jednak już na wczesnym etapie zdałem sobie sprawę, że pierwotne założenia konstrukcyjne nie wytrzymują konfrontacji z rzeczywistością. Mechanizm wielobloczkowy, który miał pierwotnie odpowiadać za zwiększenie siły zamykania szczęki, okaże się na pewno zbyt wolny w działaniu. To wyklucza jego zastosowanie w sytuacjach, gdzie szczeka miałaby pełnić funkcję ekspresyjną - np. przy symulacji mowy lub dynamicznej reakcji na otoczenie.

Dodatkowo okazało się, że rozwiązanie oparte na ścięgnach (prowadzenie linek z wykorzystaniem rolek i silników osadzonych w korpusie) w tej konkretnej lokalizacji - czyli w ciasnej przestrzeni głowy - jest bardzo trudne do zrealizowania. Brakuje po prostu miejsca na rolki, prowadnice i punkty kotwiczenia, a każda dodatkowa linka to potencjalny problem montażowy i serwisowy. W tej sytuacji musiałem stworzyć jakieś inne autorskie rozwiązanie, które spełniałoby wszystkie wymagania:

- możliwość szybkiego otwierania i zamykania szczęki,
- mechanizm zdolny do generowania silnego docisku,
- ochrona mechaniczna przed nadmiernym ściskaniem obiektu,
- maksymalna kompaktowość konstrukcji,
- potencjał do wykorzystania w przyszłości jako „narząd mowy” robota.

Po wielu szkicach i analizach wymyśliłem system napędowy z wykorzystaniem jednego serwomechanizmu (w sumie dwóch ale ten drugi to tylko zwiększenie momentu), który na osi posiadać będzie sprzęgło cierne, oraz małą rolkę na linkę. W podstawowym scenariuszu serwo obraca się i zamyka szczękę poprzez sprzęgło cierne. Jednak jeśli w szczęce znajdzie się obiekt, sprzęgło zacznie się ślizgać, co pozwoli uniknąć przeciążeń. W tym czasie jednak rolka naciągająca linkę z dyneemy zacznie zaciskać szczękę coraz mocniej - geometria zamocowania ścięgna spowoduje wykorzystanie większej dźwigni przy końcu szczęki. Tym samym zachowana zostaje funkcja delikatnego domknięcia z opcją mocnego ścisku - a wszystko to przy użyciu jednego napędu i bez zbędnych linek w przestrzeni roboczej.

Rozwiązanie to przedstawiłem w formie szkicu technicznego. Powstał też zarys koncepcji jakby to określić asymetrycznego łożyska - hmm, coś na styl zwykłego łożyska wzdłużnego oporowego ale z niesymetrycznie ustawioną osią, celem wykorzystania w bocznym ruchu żuchwy.



Szkic szczęki bez wielobloczka, wizualizacja pomysłu z sprzęgłem ciernym oraz asymetrycznym łożyskiem.

7. Pokaż, kotku, co masz w środku

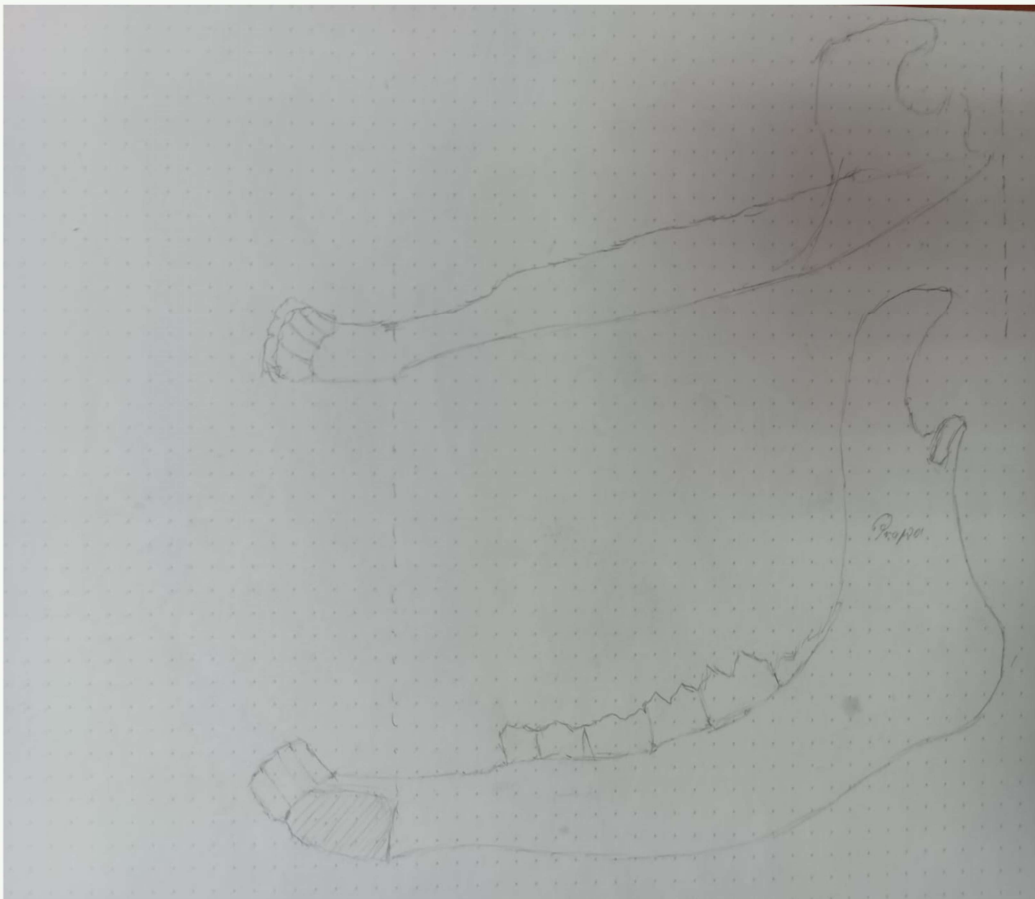
Po kilku etapach planowania, analiz i szkicowania nadszedł w końcu moment, kiedy mogłem powiedzieć: „czas zacząć zabawę”. Mając gotową koncepcję oraz wstępny projekt mechanizmu szczęki, przystąpiłem do bardziej precyzyjnych rozważań dotyczących rozmieszczenia serwomechanizmów i ich integracji z konstrukcją. Zanim jednak mogłem dobrać konkretne podzespoły, pojawiło się pytanie kluczowe: jak duża w rzeczywistości powinna być głowa robota?

Na szczęście los nieco uśmiechnął się do mnie kilka lat temu - pozyskałem wtedy kompletny szkielet danielicy z własnej hodowli, która niestety mi padła. Na początku miałem ja wypchać ale ostatecznie wygotowałem kości, oczyściłem i zachowałem szkielet na takie okazje. Żartuję oczywiście że na takie okazje, ale fakt faktem - przydał się przy tworzeniu modeli 3D do gier oraz tworzeniu animacji ruchu, lecz to inna historia.... Teraz dzięki niemu mogłem w rzeczywistej skali przyjrzeć się strukturze kości czaszki i dokładnie określić fizyczne ograniczenia przestrzenne dla planowanej mechaniki.

Wybrałem z zestawu odpowiednie kości - głównie szczękę i część czaszki - i postanowiłem wykonać rysunek w skali 1:1, który miał posłużyć jako baza do dalszego projektowania. Niestety, już na etapie rysowania okazało się, że odręczne przenoszenie kształtu kości na papier to wcale nie taka prosta sprawa - szczególnie bez idealnego kąta patrzenia i odpowiedniego oświetlenia. Rysunek udało się co prawda wykonać, ale szybko zdałem sobie sprawę, że jego dokładność jest zbyt niska, by mógł stanowić wiarygodną podstawę projektową.

Mimo to dołączam do dzienniczka rzuty prawej części szczęki - z boku i z góry - jako dokumentację etapów pracy, nawet jeśli nie zostaną one wykorzystane. To pokazuje ewolucję procesu projektowego i może co najwyżej służyć jako archiwalny zapis kierunku myślenia.

Albo straszenia dzieci że będą miały takie ząbki jak nie będą używać szczoteczki :)



Szkic obrazujący prawą część szczęki w rzucie 2D

8. Konsultacje i pomocna dłoń

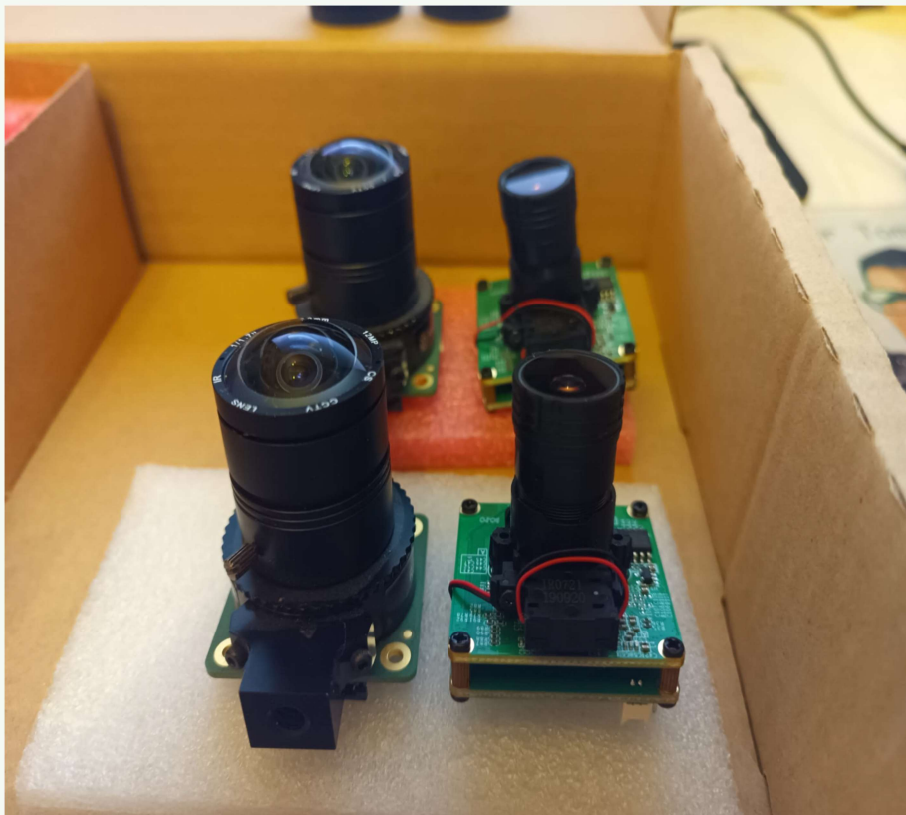
W tym etapie odbyła się szersza konsultacja techniczna dotychczasowych postępów, przemyśleń oraz problemów konstrukcyjnych. Rozmowa w firmie BTC Korporacja dotyczyła głównie idei głowy robota, mechanizmu szczęki, możliwych układów napędowych oraz ograniczeń wynikających z bardzo ciasnej przestrzeni wewnątrz czaszki.

Konsultacje okazały się bardzo pomocne. Spojrzenie z zewnątrz pozwoliło oddzielić pomysły, które warto rozwijać, od tych, które wyglądały dobrze tylko na kartce. Było trochę konstruktywnej krytyki, trochę motywacji i kilka nowych znaków zapytania - czyli dokładnie tyle, ile trzeba, żeby projekt mógł iść dalej zamiast utknąć w zachwycie nad własnym szkicem.

Miłym zaskoczeniem było też pozyskanie do testów ciekawej kamery z wymiennym szerokokątnym obiektywem. Jej kompaktowe wymiary i gotowa optyka mogą znacząco ułatwić integrację w ciasnej przestrzeni głowy.

Co więcej - ku mojemu ogromnemu zadowoleniu - otrzymałem dwa dodatkowe moduły kamer do testów, nieco mniejsze, z gotowym procesorem do postprocesingu obrazu. Dzięki temu mogę teraz prowadzić eksperymenty i prototypowanie i to na dwóch różnych zestawach kamer. To drobny gest, który jednak bardzo doceniam.

Na koniec dnia wspólnie przeanalizowaliśmy kolejne etapy projektowania części mechanicznej chwytaka - krok po kroku omówiliśmy możliwe ścieżki rozwoju, potencjalne problemy i metody ich obejścia. Spotkanie to okazało się nie tylko merytoryczne, ale i bardzo motywujące do dalszej pracy.



Kamerki otrzymane od BTC Korporacji

| 9. Gwałtownych uciech i koniec jest gwałtowny..

Weekend to tradycyjnie czas nadrabiania godzin - tym razem z ogromnym zapalem, bo mam kamerki, które wcześniej otrzymałem. Ciekawość oczywiście zwyciężyła - na pierwszy ogień poszły kamery z większym obiektywem, bo wyglądały najprofesjonalniej, miały większą rozdzielczość i wyglądały wizualnie najładniej w mojej ocenie :)

Szybko jednak okazało się, że nie będzie to „plug and play” - kamery korzystają z interfejsu CSI, czyli specjalistycznego złącza dla kamer Camera Serial Interface używanego np. w Raspberry Pi czy rozwiązaniach NVIDIA, skubaniec z tego interfejsu - może przesyłać 10Gb/s danych przy 4 linach magistrali. Poczytałem trochę o tym standardzie i dowiedziałem się, że istnieją konwertery CSI-USB, ale... przypomniałem sobie, że mój stary superkomputer NVIDIA Jetson Xavier NX, który trzymam na specjalne okazje i planowo miał służyć do kinematyki odwrotnej i sterowania silnikami BLDC w projekcie robota kroczącego, może mieć takie złącza - nos mnie nie zawiodł, posiada aż dwa dedykowane złącza CSI.

Znalazłem więc wszystko, co potrzebne do testów, w mojej warsztatowej graciarni. Euforia trwała niestety krótko - kamera nie zadziałała. Najpierw szok i niedowierzanie: może producent coś pomylił na złączu? Może pinout jest inny? Może... spaliłem coś przez złe podłączenie? Zacząłem szczegółowo analizować schematy, porównując piny zasilania i magistrali danych kamery oraz Xavier'a. Wszystko wyglądało OK - żadnych sprzeczności - więc teoretycznie powinno działać. Problem? Wciąż martwa cisza.

Po kilku godzinach kombinowania - zmiana BIOS-u, ponowna instalacja systemu (nawet kilkakrotnie, z różnymi konfiguracjami), aktualizacje - nic nie pomagało. Kamera była po prostu niewidoczna. Dopiero późnym wieczorem, po solidnym przeszukaniu internetu, trafiłem na bardzo istotny wątek na forum NVIDIA. Ktoś miał podobny problem z innym modelem kamery - okazało się, że nie BIOS, nie sterowniki, ale... kernel systemu Linux był winowajcą, w przypadku osoby z forum.

Domyślnie Jetson Xavier NX obsługuje kamery typu IMX219, a nie większe, bardziej wymagające modele przetwornika jak mój IMX477. Zgodnie z instrukcją przebudowałem kernel, skopiowałem pliki, wykonałem wszystkie polecenia... i... Nadal nie działa, sprawdziłem kolejny kernel i te same dwa kernele na starszej i jeszcze starszej wersji systemu operacyjnego Ubuntu i... cóż się mogło stać - a no nic, NADAL NIE DZIAŁA. O tak od razu lepiej, większy napis bardziej oddaje moje wkurzenie na tą nieszczęsną sytuację z kamerką:)

Opadły mi ręce. Dosłownie. Aż nie chce się mi pisać dalej tej dokumentacji, jak można przez 12 godzin, podłączać jedną kamerę CSI do Xaviera, a co lepsze bez pozytywnego skutku. Po tylu godzinach walki z czymś, co miało być prostym testem kamery, poszedłem spać pokonany...

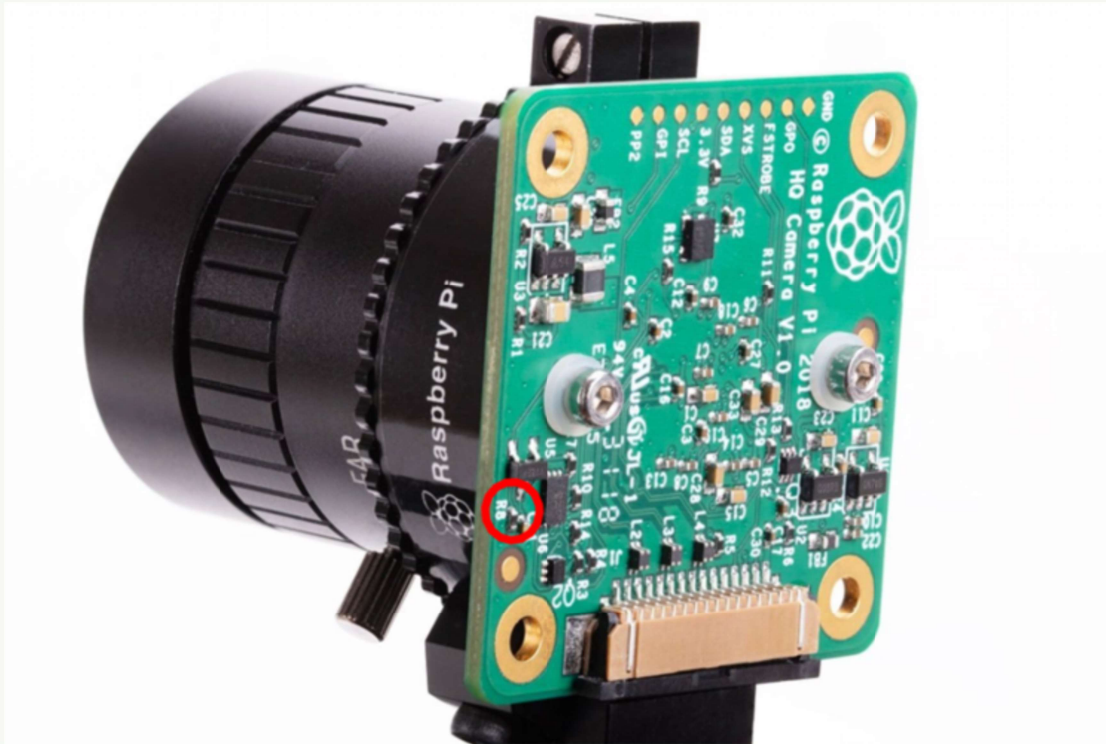
| 10. CSI: Xavier - po nitce do kłębka

Od rana kontynuowałem nierówną walkę z niepokorną kamerą IMX477. Co prawda miałem odpuścić temat, ale po tylu godzinach prób i zmianach systemu honor nie pozwalał się poddać. Postanowiłem sprawdzić, czy Jetson Xavier w ogóle próbuje rozmawiać z kamerą - do sprawdzenia było zaledwie kilka rzeczy: zasilanie, linie I2C, sygnał taktujący, magistrala CSI i kilka linii sterujących. Brzmi prosto? Teoretycznie. Niestety nie posiadam dobrego analizatora stanów logicznych, więc ratowałem się starym dobrym oscyloskopem analogowym C1-99, tak to ten sam który prawie pamięta czasy zimnej wojny :). Mimo wieku to nadal mój ulubiony sprzęt.

Po restarcie systemu zauważyłem, że na I2C jest aktywność - czyli coś tam Xavier gada. Sprawdziłem napięcia zasilania - wszystko się zgadzało. Ale coś mi nie grało z pinem nr 11 (Power Enable). Na jego wyjściu napięcie wynosiło 1.8V, podczas gdy kamerka zasilana była 3.3V. Teoretycznie mogło to działać, ale coś mi mówiło, że to potencjalna przyczyna problemu. Zacząłem analizować ścieżki na PCB i znalazłem winowajcę - sygnał „Power Enable” powinien mieć logikę 3.3V, ale z jakiegoś powodu tak nie było i kamera nie była w stanie się wybudzić, otrzymując z komputera sterowanie 1.8V. Chwila zastanowienia i chęć włączenia kamery na stałe - diagnoza: odlutować rezystor R8, który „blokował” kamerę. Po tej operacji kamera została wykryta w BIOSie - pojawiła się informacja o przetworniku obrazu IMX477.

Uruchomiłem przykładowy kod z pakietu NVIDIA Jetson służący do wyświetlenia obrazu z przetwornika kamery i... kamera zadziałała niemal natychmiast! Po drobnej regulacji przysłony i ostrości uzyskałem idealny obraz w 4K/30FPS - ostrość jak żyletka.

Załączam zdjęcie pokazujące miejsce na PCB z nieszczęsnym R8 - niech ten element będzie przestrogą dla innych. To był dzień pełen detektywistycznej pracy - ale opłaciło się, dalszą część dnia w sumie poświęciłem wyłącznie na zabawę z tą kamerką i testowanie dostarczonych przez producenta Xaviera programów testowych - jedne programowo poprawiają zniekształcenia soczewki, inne stabilizują obraz, wykrywają ruch itd. - jest tego całkiem sporo.



Zaznaczony skubaniec przez którego straciłem prawie dwa dni życia.

11. Wojna światów, czyli ja vs. Python

Po ostatnich bojach z kamerą postanowiłem dać mózgowi chwilę wytchnienia od sprzętu, kabli i oscyloskopu z epoki wczesnego Gagarina. W ramach resetu - przesiadka na programowanie. Oprogramowanie jest przecież integralną częścią głowy robota: to ono ma powiedzieć mechanice, na co patrzy, gdzie znajduje się cel i co warto z tym zrobić.

Chwilę biłem się z myślami - pisać to wszystko od razu na Jetsonie Xavier z Linuxem, czy może na mocniejszej komputerze z Windowsem, gdzie mam już gotowe środowisko i nie muszę bawić się w kombinowanie z zależnościami, wersjami bibliotek i innymi atrakcjami znanymi każdemu, kto kiedykolwiek próbował postawić Pythona + CUDA + cudaToolkit. Decyzja zapadła: Windows + zwykła kamera USB. Xavier i IMX477 wrócą do gry później, jak już wszystko będzie dopracowane.

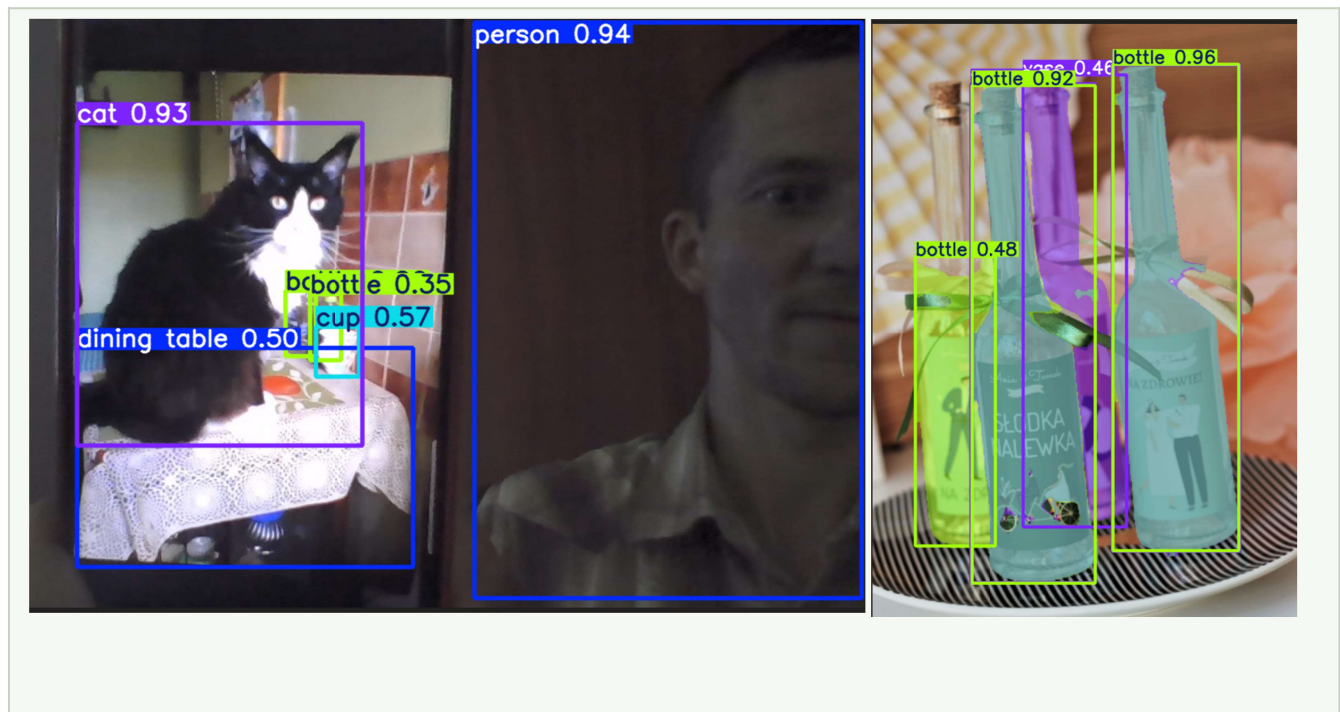
Pisanie oprogramowania samo w sobie nie jest dla mnie nowością - zawodowo tworzę oprogramowanie, więc nie spodziewałem się wielkiej rewolucji, ale mimo to potraktowałem to jako miłą odmianę. Do głowy wracał jednak cały czas temat mechanicznego chwytaka, który czeka na rozwiązanie kilku smakowitych problemów konstrukcyjnych... Ale zanim do tego wrócę, trzeba przygotować coś, co mu powie, co i gdzie chwycić lub ugryźć a może nawet porazić prądem...

Zabrałem się więc za pisanie pierwszej wersji algorytmu detekcji obiektów, klasyfikacji i wyznaczania ich współrzędnych. Na tym etapie nie interesują mnie jeszcze szczegóły typu sterownik, wyjście/wejście - to będzie później. Teraz chodzi tylko o to, żeby mój soft wiedział, na co patrzy. Poniżej ogólna koncepcja algorytmu, który zacząłem w dniu dzisiejszym implementować:

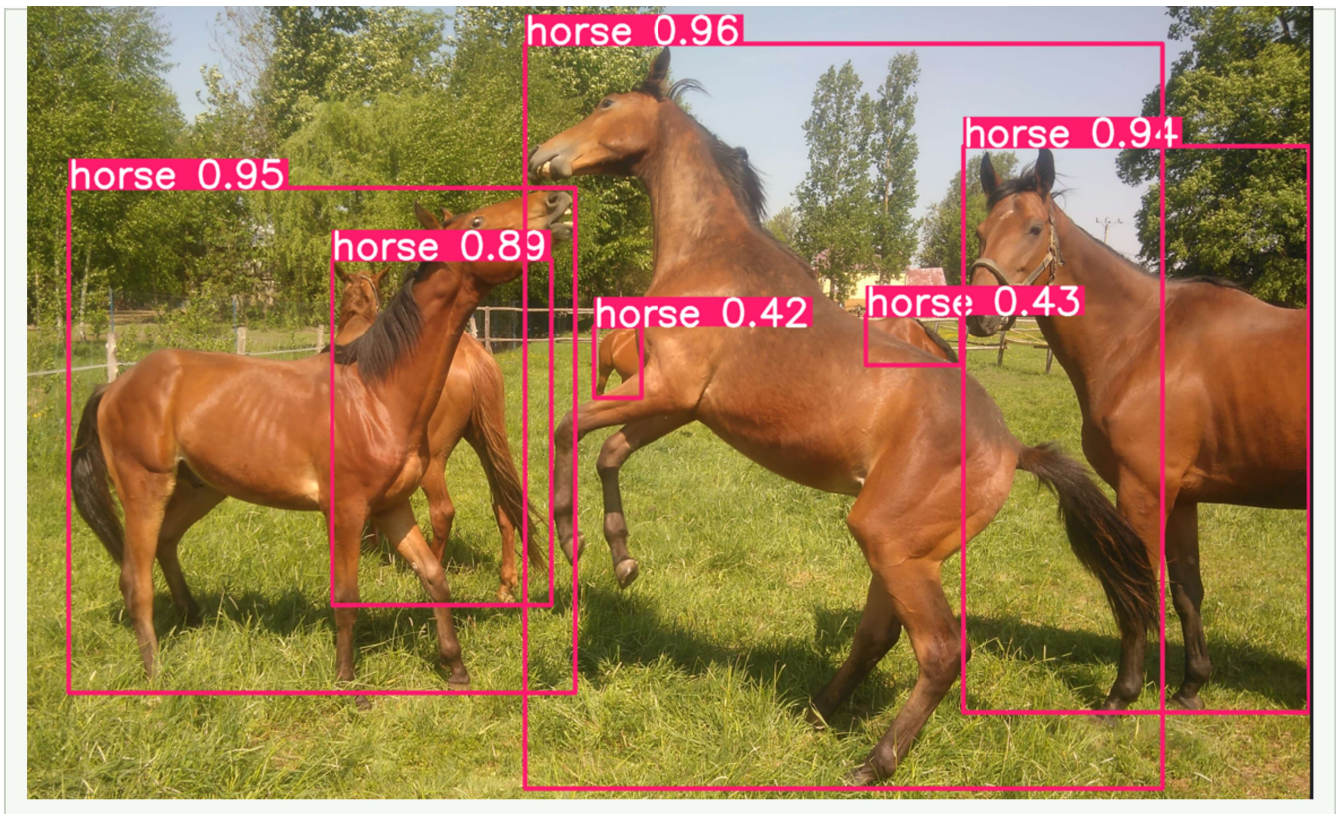
Algorytm lokalizacji i klasyfikacji obiektów (wersja: prowizorka v0.1):

- Pobranie klatki z kamery (stream lub na początek pojedyncza klatka z kamery USB albo obraz z pliku celem testu modelu sieci neuronowej).
- Wstępne przetwarzanie obrazu (konwersja do RGB, skalowanie do rozmiaru modelu, ewentualny blur i korekcja jasności).
- Wykrycie obiektów (test modeli np. YOLOv8 lub MobileNet lub nowsze, wskazanie obiektów z klasyfikacją, współrzędne ramki detekcji).
- Filtracja obiektów wg priorytetów i planu (np. tylko „bottle”, „cup”, itd., pierwszy plan, obiekt najbliższy).
- Wyznaczenie środka wykrytego obiektu oraz obliczenie kąta przechylenia od osi.
- Przekształcenie współrzędnych pikselowych na przestrzeń fizyczną w np. mm (skalowanie lub kalibracja kamery. I tu mały kłopot bo trzeba znać odległość do celu)
- Zapisanie obiektu i jego współrzędnych do bufora.
- (opcjonalnie) Wyświetlenie obrazu z naniesionymi ramkami, nazwami klas i współrzędnymi.

Z tak przygotowanymi danymi w przyszłości będzie można bezpośrednio sterować chwytakiem, niezależnie czy to przez UART, TCP czy Morse’em przez przekaźnik:, zależnie od potrzeb i napisanego oprogramowania sterownika urządzenia. Poniżej screen z działania programu, który świetnie radzi sobie z klasyfikacją obiektów i wyznaczaniem pozycji głównego celu.



Test modelu do segmentacji obiektów na przykładowych danych wejściowych. Działa całkiem dobrze..



A tak model widzi moje kopytne. Co ciekawe nawet wypatrzył te kobyłki o prawdopodobieństwie 0.42 i 0.43

12. Houston, mamy problem

Dzień zaczął się niewinnie - kontynuowałem przegląd modeli sieci neuronowych do klasyfikacji i segregacji obiektów. Wszystko szło jak po maśle. Modele wyglądały obiecująco, a ja już prawie widziałem tego mojego robota łaniopodobnego z gracją sięgającego po paczkę chipsów. Ale wtedy nagle... BUCH - jakby ktoś przywalił mi obuchem w łeb. Zdałem sobie sprawę, że cały misternie zaprojektowany chwytak, nawet jeśli będzie znał współrzędne i dzięki czujnikowi ToF obliczy odległość do obiektu... nie będzie miał zielonego pojęcia z jaką siłą ten obiekt złapać. A to dlatego, że zapomniałem o czujniku nacisku w fazie projektowania szczęki. No i po balu. Miało być pięknie, a tu kłops. Oczywiście - dorzucenie tensometru albo jakiegoś prostego enkodera ze sprężynką nie jest technicznie niewykonalne. Problem polega na tym, że wszystko - nawet najmniejszy dodatek - pożera czas, a tego mam jak na lekarstwo.

Próbowałem kombinować z pomiarem prądu w serwie, żeby jakoś oszacować siłę, ale to droga przez mękę - nieliniowość, mała precyzja i mnóstwo problemów. Z tyłu głowy miałem też cały czas myśl, że na koniec tego etapu chcę mieć coś fizycznego, cokolwiek, co się rusza, nawet jeśli będzie to tylko kawałek sklejk wyciętej laserem. Dlatego chwilowo ograniczyłem zakres: oprogramowanie ma rozpoznać cel i wyznaczyć współrzędne, a mechanika ma spróbować go chwycić. A jak zostanie czas, to się dopisze więcej.

W tym etapie przetestowałem kilka modeli nie tylko do klasycznej detekcji obiektów, ale też do segmentacji oraz rozpoznawania twarzy. Szczególnie przydatny okazał się embedding, czyli cyfrowy podpis twarzy zapisany jako wektor cech. Dla człowieka to tylko ciąg liczb, ale dla programu to coś w rodzaju odcisku twarzy w przestrzeni latentnej. Porównując takie wektory, np. przez odległość euklidesową, można ocenić, czy nowo wykryta twarz należy do znanej osoby, czy trzeba dodać ją jako nowy profil. To ważne, bo docelowo głowa Zervidjanna może wiedzieć, kto do niej podchodzi, i dodać tę informację do kontekstu rozmowy.

Na dziś starczy. Fajnie czasem popisać kod bez ryzyka spalenia przetwornika za kilkaset złotych. Jutro? Kto wie - może znowu wrócę do walki z mechaniką, tymczasem załączam postęp prac dotyczących oprogramowania, które chwilowo uznaję za zakończone jeśli chodzi o detekcję i wyznaczenie współrzędnych w czasie rzeczywistym, driver do sterowania serwami dopiszę jak skończę głowę, bo wtedy będę dopiero wiedział jakie wej./wyj. oprogramować.



Test na przykładowej fotografii. Całkiem nieźle „łapie” twarze.

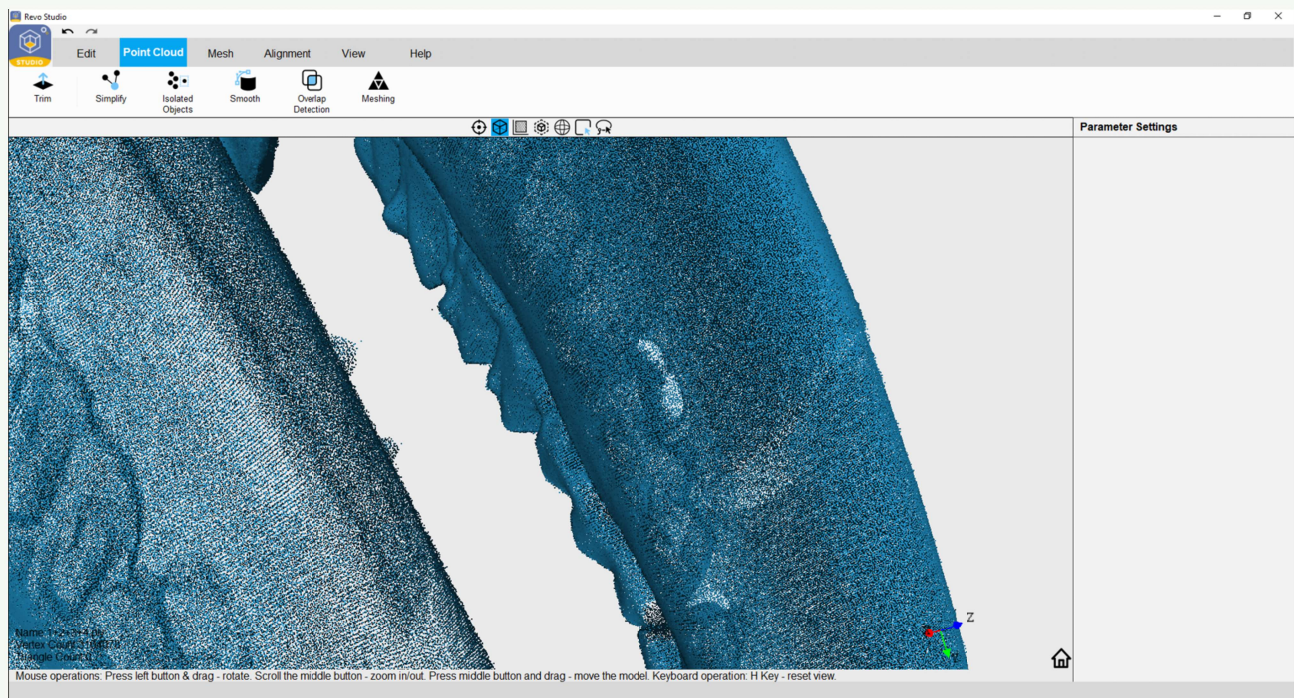
13. Powrót do przeszłości, aż po same kości

Choć lubię pisać oprogramowanie, to jednak brakuje w tym czegoś... namacalnego. Brakuje zapachu topionej kalafonii, przypalonych grotem palców i świecących z nadmiaru prądu rezystorów. Obcowanie z fizycznym układem ma po prostu „to coś”. Ale do rzeczy - 2D rysowane ołówkiem to już zdecydowanie za mało. Pora sięgnąć po cięższe uzbrojenie, czyli skaner 3D.

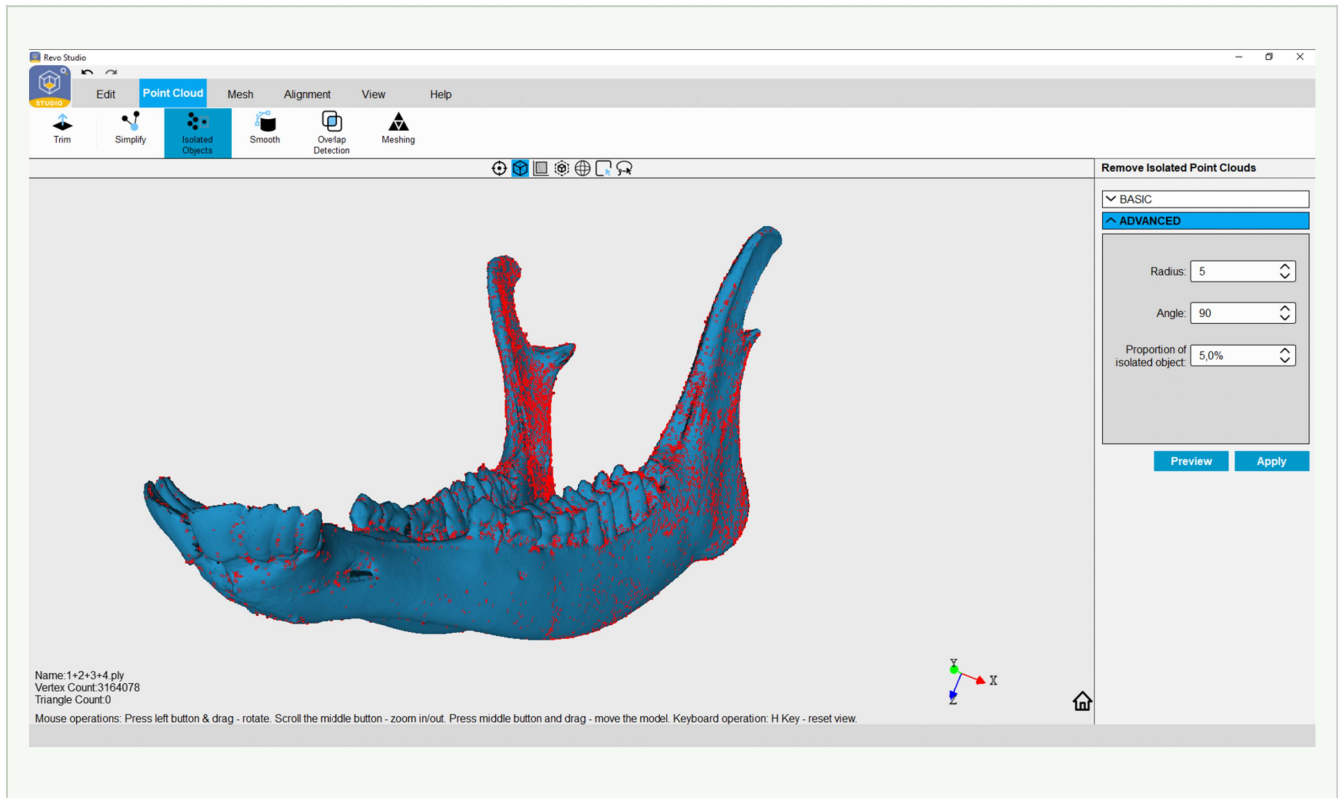
Tak, dziś realizowałem prace związane ze skanowaniem kości czaszki, szczęki oraz ostatnich kręgów kręgosłupa. Dla niewtajemniczonych - jeśli ktoś nigdy nie bawił się takim cudeńkiem, to zdradzę mały tip: warto skanowany obiekt pokryć kredą w sprayu. Dzięki temu oprogramowanie o wiele lepiej łączy pojedyncze skany w jedną, spójną chmurę punktów. Skaner przez chwilę się kurzył, więc zaczęło się klasycznie: aktualizacja softu. Zainstalowałem najnowszą wersję RevoScan - i oczywiście, jak zwykle, zapomniałem zapisać moje ulubione ustawienia. No to jazda z parametrami od nowa...gdzie ja mam głowę :) Po re-instalce i zeskanowaniu odpowiednich części szkieletu, przekształcałem chmurę punktów tzw. mesh w model 3D jako siatkę poligonów. Potem eksport do formatu .obj, żeby móc później wykorzystać możliwości Blendera w postprocessingu. Tym razem poszło gładko, nie jak z IMX477, zeskanowałem czaszkę, szczękę oraz dwa ostatnie kręgi. Na tym etapie powinno wystarczyć do dość dobrego odwzorowania wymiarów oraz jako podstawa do dalszych prac projektowych. Patrzę na ilość tekstu jaką napisałem i gdyby to nieco bardziej rozpisać i dodać do tego parę skomplikowanych wzorów i wykresów, najlepiej takich których nikt nie zrozumie - miałbym chyba pracę inżynierską za sobą :)



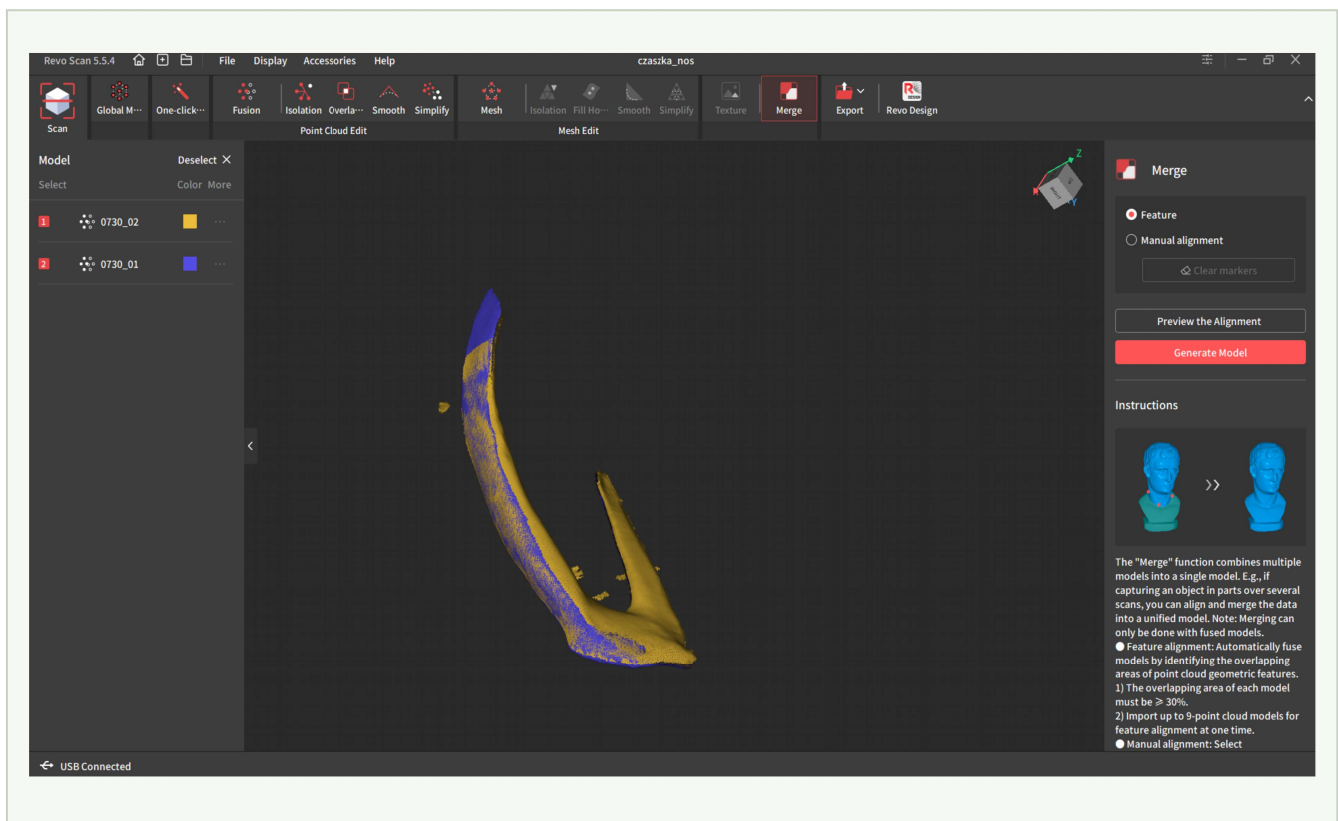
Stanowisko pracy.



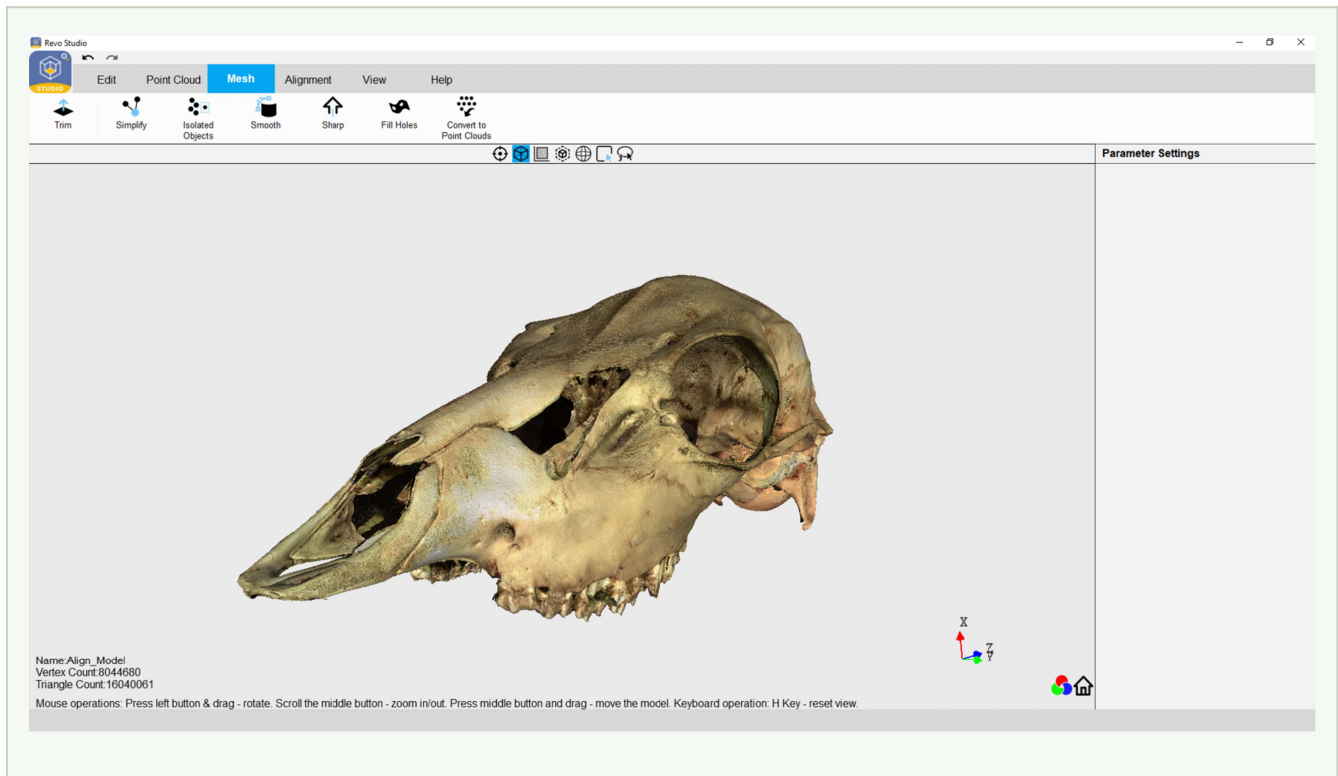
Tak zwana mesh - czyli chmura punktów w przestrzeni którą widzi skaner. (szczęka)



Zaznaczone i wstępnie usunięte artefakty, już na etapie tworzenia siatki poligonów z mesh.



Łączenie kilku osobnych perspektyw skanowania przed fuzją tysięcy klatek mesh w jedną. Tutaj część nosa.



A tak np. wygląda czaszka po obróbce wstępnej z naniesioną teksturą po zakończeniu skanowania. Na startcie model ma ok 16 milionów poligonów, a to znaczy że komputer musi być o konkretnej mocy obliczeniowej, bo inaczej samo otwarcie pliku w blenderze potrwa chwilę.

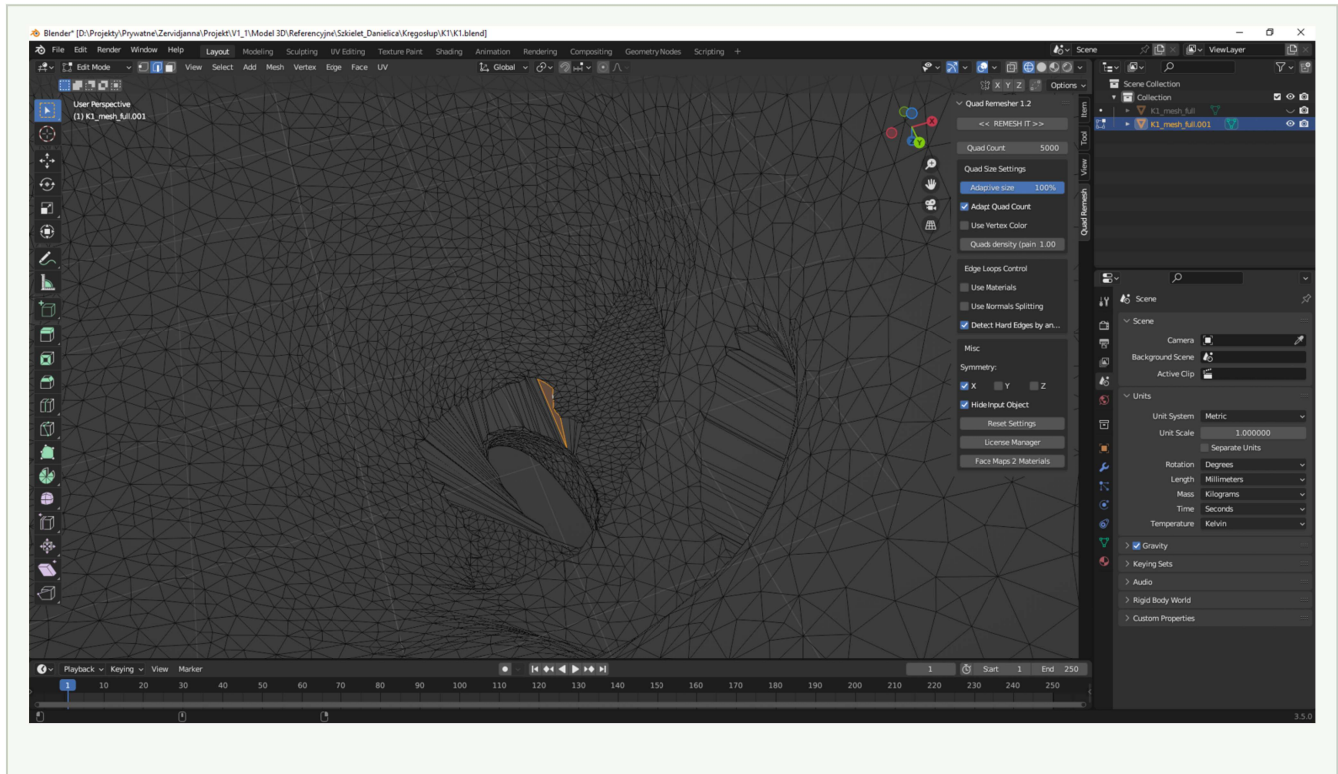
14. Topologia

Tym razem opiszę przykład zadania z kategorii: „nie trudne, ale czasochłonne”. Skupiłem się na ręcznej obróbce siatek 3D modeli, które wcześniej uzyskałem ze skanera. Niektóre etapy skanowania 3D musiałem wykonać ponownie. Modele pochodzące z laserowego skanowania - szczególnie te bardziej skomplikowane - mają niestety swoje artefakty, czyli różne błędy topologii, nieciągłości powierzchni, „dziury” czy przekrzywione vertexy. I trzeba je usunąć - ręcznie.

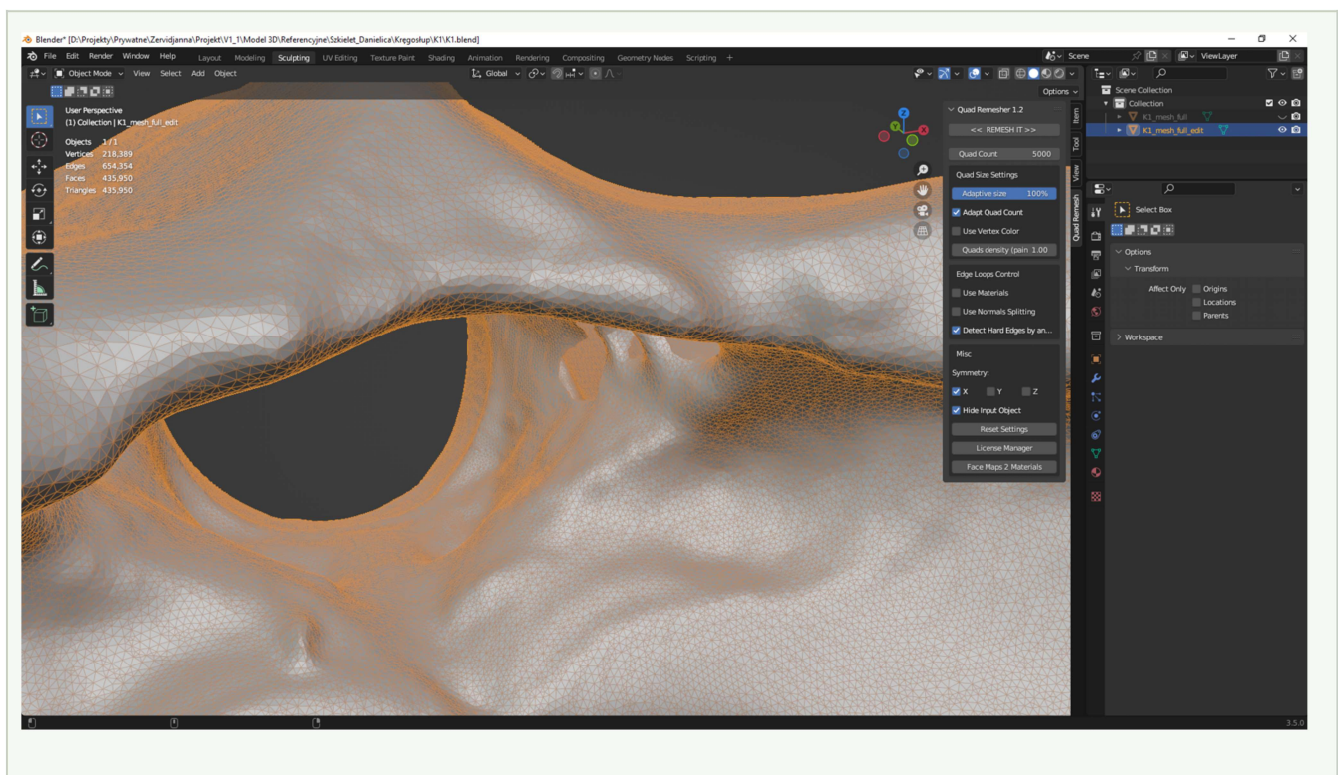
Dla samego prototypu nie jest to absolutnie konieczne, bo model 3D służy przede wszystkim jako baza do szacowania skali i rozmieszczenia elementów. Ale... no właśnie - chyba poświęciłem na to tyle czasu, bo jestem trochę pedantyczny w kwestiach projektowych. Ładnie obrobiony model można potem tekstuować, użyć w symulacji albo jako materiał prezentacyjny. Kolejny etap prac to postprocessing siatki ze skanera: cierpliwość, kawa, dobra muzyka i ręczne scalanie poligonów, czyli zestaw obowiązkowy.

Szczerze? Podziwiam zapał grafików 3D, którzy na co dzień tworzą z taką precyzją postacie do gier. Ja dziś tylko dłużej przy kościach i szczęce, a i tak chwilami miałem ochotę wrócić do kodu i debugowania algorytmów. Model wygląda teraz dużo lepiej i może posłużyć jako estetyczna podstawa pod przyszłe prace projektowe.

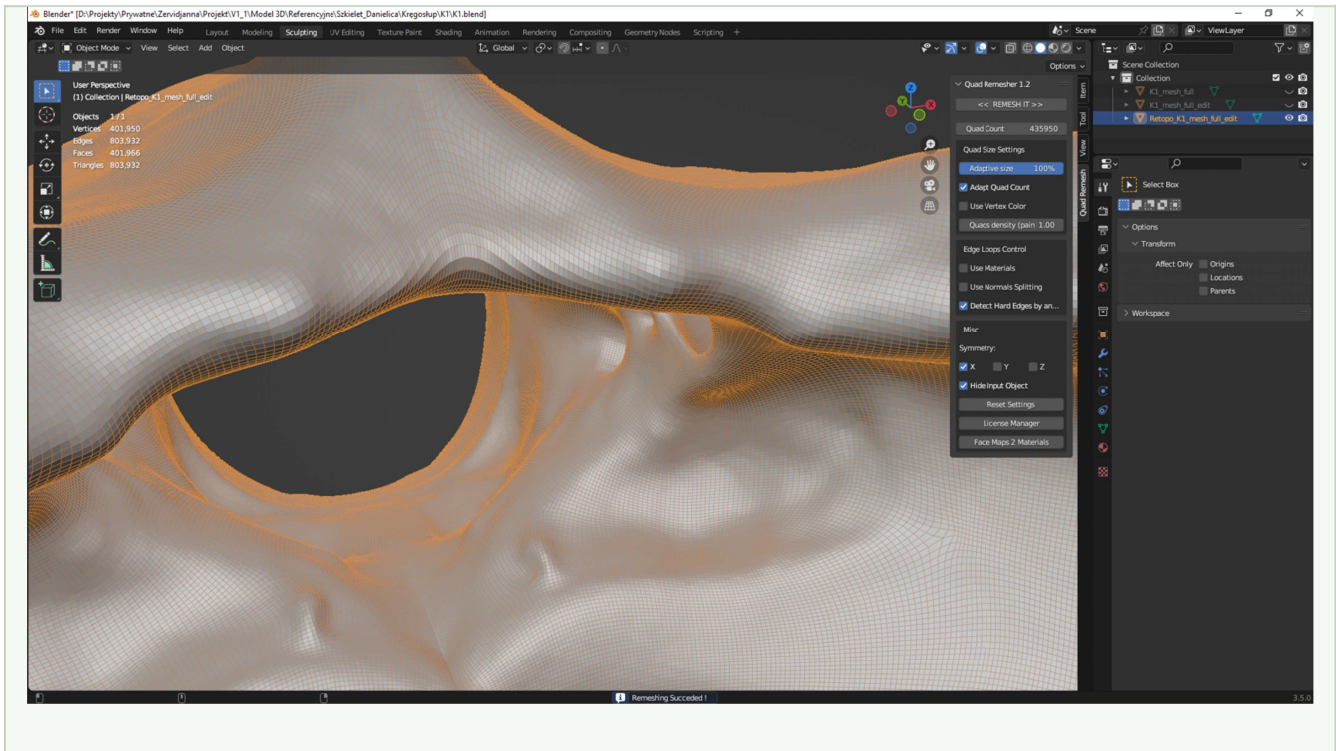
Tylko jak teraz pójść spać i wstać z wyrka rano, skoro jest już prawie trzecia nad ranem? Nieważne - jest satysfakcja, a to - bezcenne.



Ręczne łatanie dziur poprzez łączenie odpowiednich poligonów.



Utworzenie i lekkie wygładzenie siatki dynamicznej.

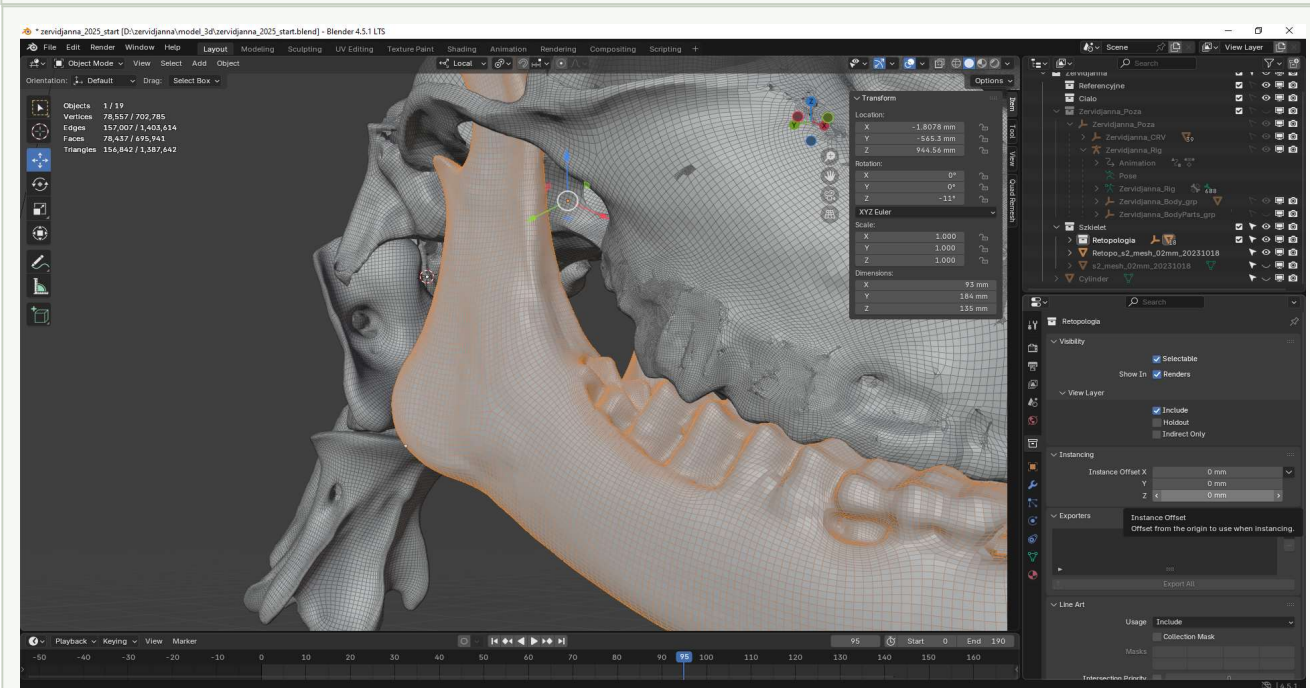
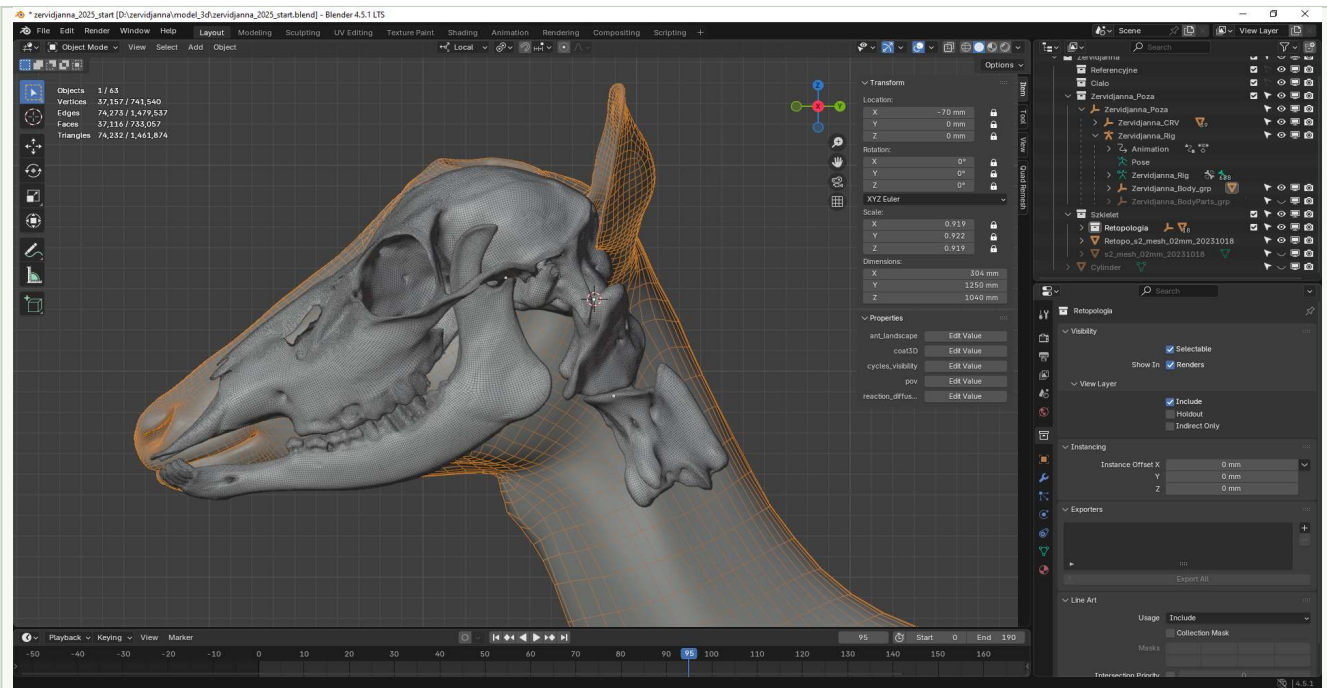


Zmiana topologii modelu i symetryzacja.

15. Tetris 3D

Zabrałem się za ustawianie wcześniej przygotowanych obiektów w Blenderze, żeby całość w końcu zaczęła przypominać coś konkretnego. Na zeskanowany wcześniej szkielet (3D + wstępna obróbka) nałożyłem siatkę ciała zwierzęcia, którą przygotowałem jeszcze kilka lat temu przy okazji projektu Zervidjanny - może w końcu doczeka się ona głowy:). Potem przyszła kolej na mozolne dopasowywanie wirtualnych kości tak, by tworzyły spójną anatomicznie strukturę i dobrze „wpasowały się” w model. Na ich podstawie ustawiłem osie obrotu i maksymalne zakresy ruchu dla poszczególnych segmentów - głównie po to, by sprawdzić, czy np. szczęka lub kręgosłup nie będą kolidować z innymi elementami i czy cały ruch wygląda naturalnie. Taka cyfrowa przymiarka to spore ułatwienie na późniejszych etapach - pomaga określić kąty i ograniczenia, które będzie trzeba przenieść do wersji fizycznej.

Dodałem tzw. rig, czyli wirtualny szkielet sterujący - dzięki niemu można symulować ruchy i testować mechanikę całej konstrukcji. Niestety, rigowanie w Blenderze to dla mnie nowość - ostatni raz miałem z czymś podobnym do czynienia lata temu w 3Ds Maxie, gdzie wyglądało to trochę inaczej. Przyznaję, szło mi to chwilami jak krew z nosa, więc nie udało się do końca dokończyć symulacji - ale na szczęście nie była ona konieczna, by ruszyć dalej. Na obecnym etapie mam już ładny rzut boczny w skali 1:1, z zaznaczonymi osiami obrotu i określonym zakresem ruchów poszczególnych stawów. To w zupełności wystarczy, by oprzeć na tym dalsze prace projektowe - w tym poglądowe szkice techniczne, które będą nie tylko odpowiednio wyskalowane, ale też możliwie bliskie anatomicznej rzeczywistości.



Dokładnie ułożone kości szkieletu wraz z naniesionymi osiami obrotu, celem oszacowania zakresu ruchu na podstawie geometrii stawów.

Po ustawieniu szczęki tak, by tylny zgryz był równy, a oś obrotu pozostała na miejscu, udało się odczytać kąt wychylenia w osi „Z” - tu wynosi on -10° , co daje zakres pracy szczęki w przybliżeniu $\pm 10^\circ$. W sam raz, żeby zrobić teatralny zgryz, przesunąć żuchwę z dezaprobatą... albo udawać że żuje się Mambę.

Z niektórymi stawami, jak np. szczeka, jest ten problem, że mechanicznie może się otwierać niemal bez końca - dopiero natura mówi „stop”, wrzucając mięśnie i ścięgna jako blokady. Ale ile to właściwie powinno być? Wstępnie przyjmuję 15 stopni... chociaż zaraz - przecież trzeba zmieścić co najmniej 7,5 cm! No dobra, zaokrąglamy do 8 cm - bo kto wie, może kiedyś robot będzie musiał przynieść komuś ęketuB awip. Trzeba być gotowym na wszystko :), czyli z 15 robimy ok 25 stopni.

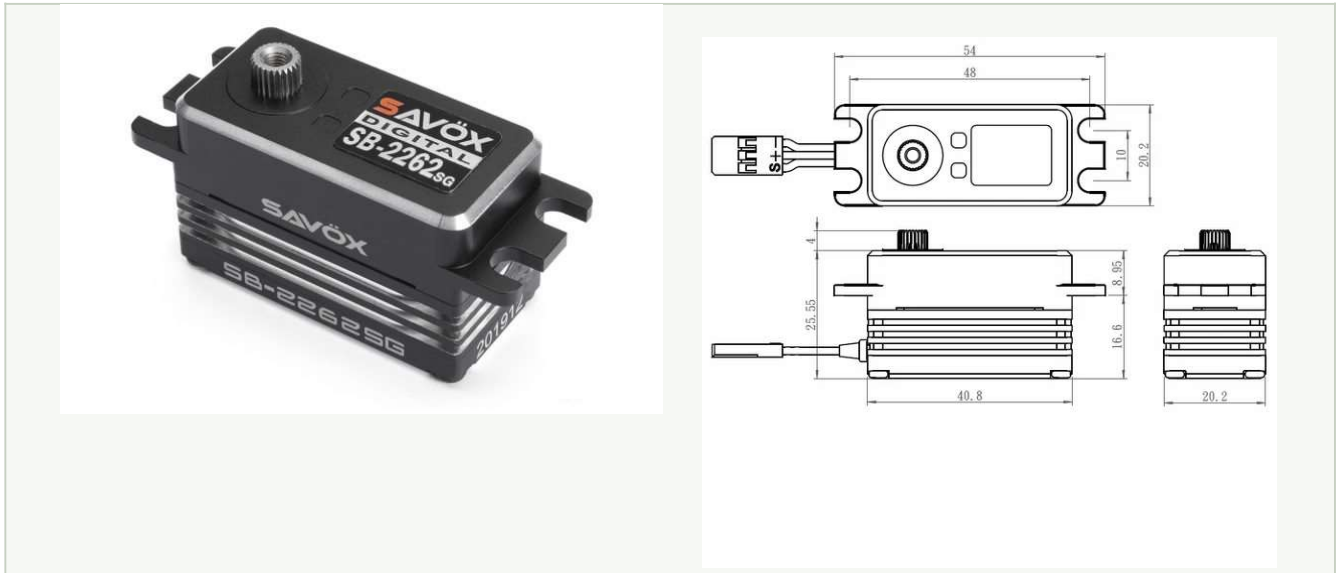
16. Mały, chudy, ale byk

Tym razem czas poświęciłem poszukiwaniom odpowiedniego serwomechanizmu dla kluczowych elementów konstrukcji czyli w tym przypadku szczęki. Kryteria były jasne i raczej bezlitosne:

- duży moment obrotowy,
- duża prędkość kątowna,
- kompaktowe wymiary - najlepiej niskoprofilowe,
- silnik bezszczotkowy dla dłuższej żywotności i większej sprawności,
- łożyska na wyjściu osi bo lubię jak coś jest „wieczne”,
- metalowa przekładnia gwarantująca wytrzymałość pod obciążeniem.

Przy okazji analizowałem też kwestie montażu: kształt obudowy, możliwe sposoby zamocowania oraz przestrzeń, którą serwo będzie zajmować w docelowej konstrukcji. Po odrzuceniu kilku modeli, które były „prawie dobre”, padł wybór na Savox SB2262SG - serwo o świetnym stosunku momentu do wymiarów, a do tego szybkie.

Na koniec jeszcze raz przeanalizowałem parametry: prędkość obrotu, moment przy różnych napięciach zasilania oraz odporność na przeciążenia. Wnioski? Teoretycznie powinno to działać perfekcyjnie... choć projektua potrafi psuć najlepsze plany. Klik i dwa takie cudenka miały dotrzeć na dniach.



Bardzo ciekawe serwomechanizmy biorąc pod uwagę ich moment obrotowy w stosunku do wielkości.

17. Czacha dymi

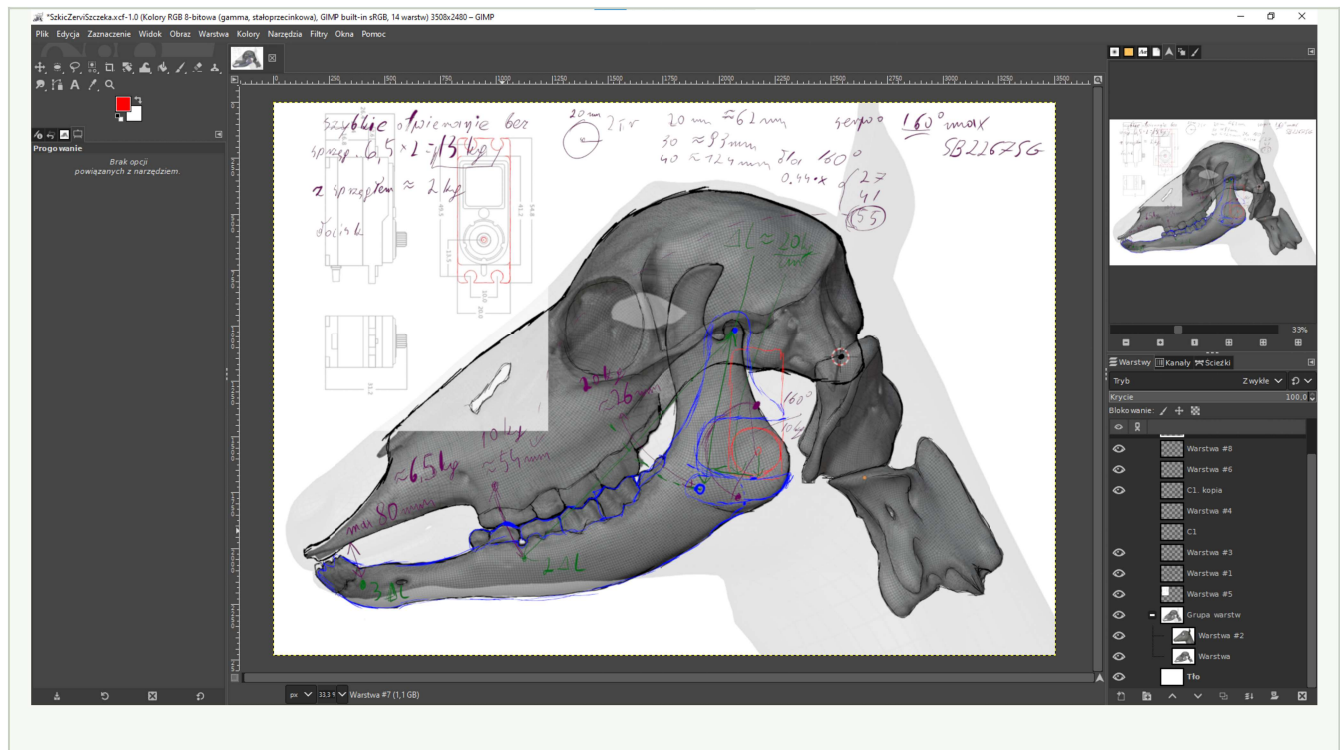
Na podstawie rzutu bocznego modelu ciała wraz z zaznaczonym szkieletem wykonałem za pomocą tabletu graficznego szkic konstrukcyjny - a może trafniej: szkic biomechaniczny. Rysunek uwzględnia skalę oraz przebieg osi obrotu poszczególnych segmentów, co jest kluczowe przy projektowaniu mechanicznych połączeń. Zazaczyłem też wstępnie lokalizację elementów, które planuję umieścić w prototypie.

Najwięcej czasu pochłonęła jednak próba dopracowania mojego pierwotnego pomysłu na cierne sprzęgło. Pomysł pojawił się już na początku projektu, ale - jak to zwykle bywa - w miarę upływu czasu zaczęły się mnożyć pytania: Czy będzie się ślizgać? Jak ustalić optymalny docisk? Czy nie zużyje się zbyt szybko? W głowie to zawsze działa idealnie... gorzej, gdy trzeba to narysować i policzyć.

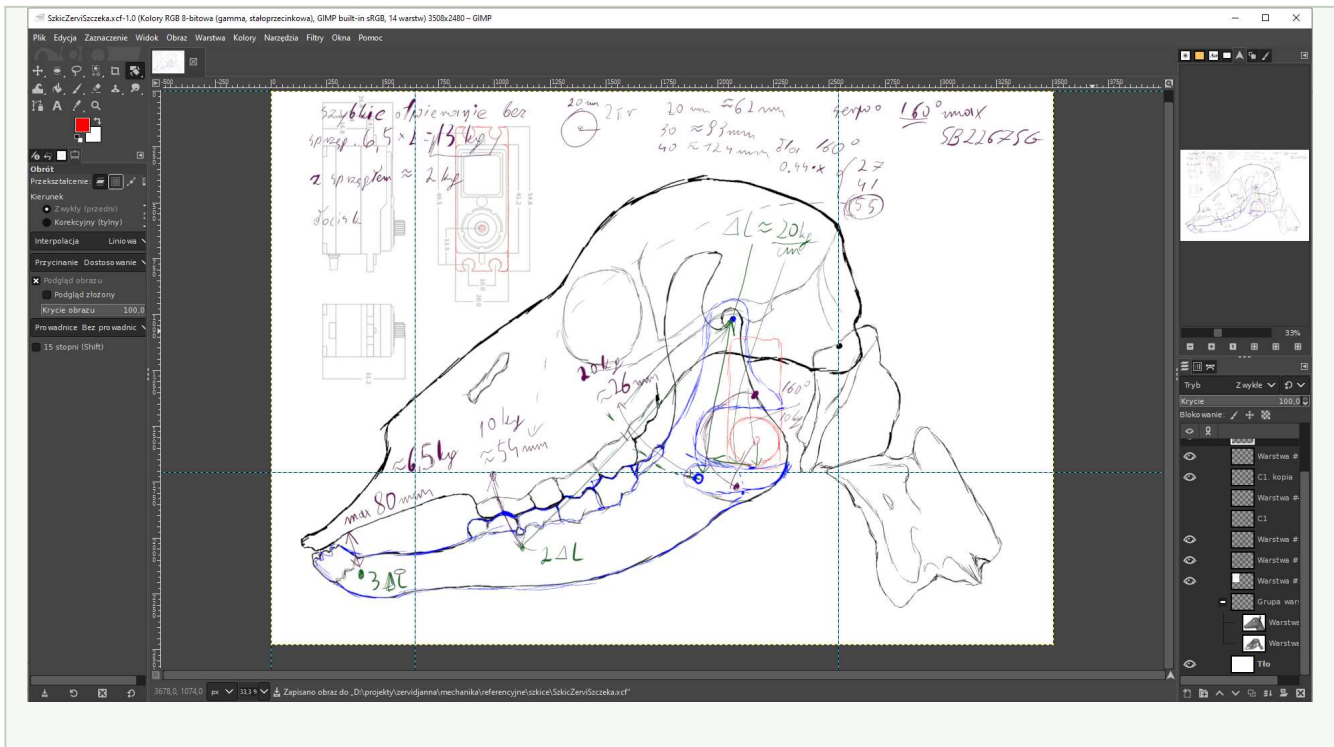
Zakładając serwomechanizm o momencie ok. 30 kg·cm, umieszczony bezpośrednio na osi obrotu szczęki, siła na jej końcu (przy długości 180 mm od osi do przednich zębów) byłaby ok. 18 razy mniejsza - czyli ok. 1,6 kg nacisku. Przy dwóch serwach daje to w teorii ~3 kg. Mało. Zakres pracy szczęki to ok. 25°, a serwo ma 160°, więc i tak nie wykorzystujemy jego pełnego potencjału w precyzyjnym ustawieniu rozwarcia.

Plan był taki: po złapaniu obiektu w zęby i zadziałaniu sprzęgła ciernego linka umieszczona 120 mm od osi miała nawijać się na dodatkową rolkę automatycznie zwiększając moment. Aby to działało, linka musiałaby skrócić się o ok. 54 mm co daje ok 8 cm na końcu szczęki zgodnie z pierwotnym założeniem - wymusza to zastosowanie rolki o średnicy ~40 mm (obwód 124 mm), żeby przy zakresie 160° uzyskać przesunięcie ~55 mm. Przy rolce o promieniu 20 mm moment serwa spada o połowę, co w teorii dawałoby ~6,5 kg ścisku dla jednego serwa (zakładając ograniczenie do 20 kg·cm na serwo). Dwa serwa = 12 kg - już nieźle, ale... złożoność mechanizmu rosła jak drożdże w ciepłej kuchni.

Po długich próbach upraszczania konstrukcji (nie dlatego, że się nie da, tylko dlatego, że miejsca mam tyle co kot napłakał) wyrzuciłem z projektu zarówno linkę, jak i ciernie sprzęgło. Zamiast tego umieściłem serwo tak, by pełny moment przekazywało w odległości 60 mm od osi obrotu. Rozwiązało to problem z dociskiem, mechanizmem linkowym i sprzęgłem, a siła z jaką robot mógłby chwycić obiekt została na poziomie 12kg na końcu dość długiej bo 180mm szczęki. Czyli parametry szybkości i mocnego docisku zostały zachowane, jednocześnie upraszczając konstrukcję i tworząc ją bardziej niezawodną.



Odrysowywanie geometrii na podstawie bocznego rzutu kośćca i modelu ciała.



Wstępny zarys usytuowania serwa (czerwone), zaznaczone (na oko) długości odległości od osi obrotu tj. 60, 120, 180mm (zielone), i szacowane wartości siły nacisku (fiolet).

18. Sekundowa robota

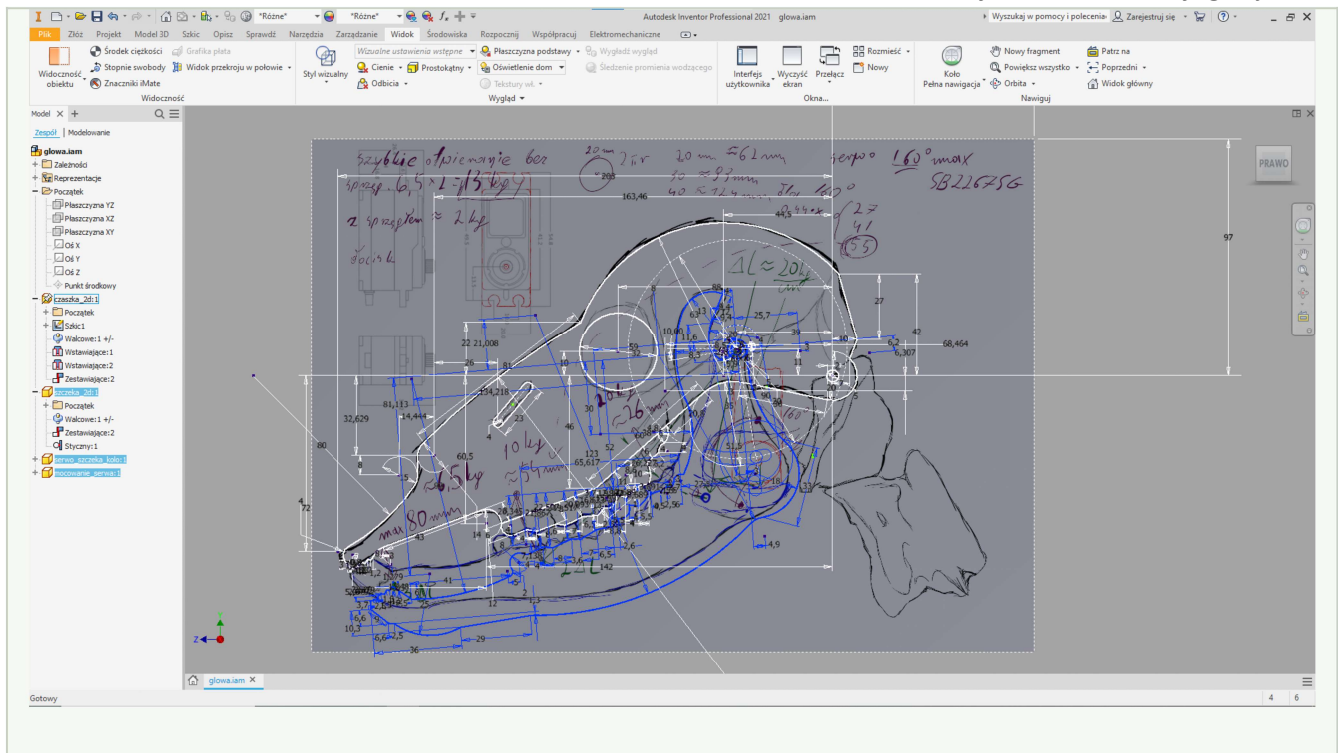
Na podstawie wcześniej przygotowanego rysunku biomechanicznego zrobiłem w Autodesk Inventor szkic, który pozwolił utworzyć proste, trójwymiarowe elementy głowy. Celem było sprawdzenie w praktycznym teście, czy mój pomysł z osadzeniem serwa niżej faktycznie ma sens.

Proces był prosty choć w praktycznym teście sporo czasu zajęło przeniesienie rysunku na jego zwymiarowaną i osadzoną wersję w Inventorze. Efekt końcowy? Całość prezentuje się obiecująco i według mojej oceny działałaby bez zarzutu... gdyby nie te nieszczęsne „ale” które widać nawet nie zastanawiając się długo. A nawet widać na starcie dwa problemy:)

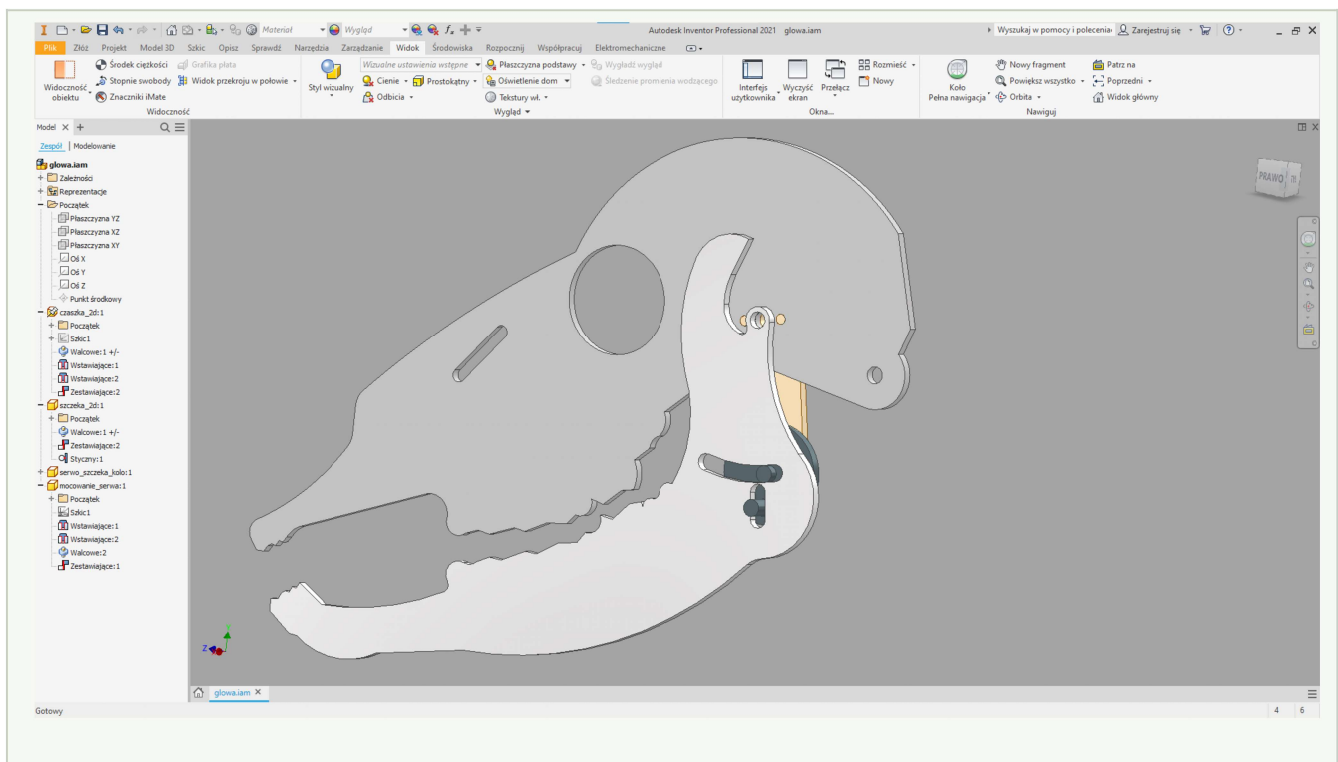
Serwo które nie ma przekładni samohamownej naturalnie wymaga prądu do utrzymania zadanej pozycji tym większego im większy musi utrzymać moment obrotowy na wyjściu swojej osi. Serwa które zamówiłem nie mogą utrzymać dużego momentu przez czas dłuższy niż sekundę. W moim przypadku oznacza to ograniczenie do około 6 kg*cm (czyli 1:5 jego maksymalnych możliwości) przy obecnej konfiguracji szczęki a to bardzo kiepsko. Pomysł z obniżeniem serwa jest dobry, ale teraz priorytetem staje się odciążenie serwa, niezależnie od tego, jak duży obiekt „sarniuk” będzie miał w pysku. Nawet jeśli kiedyś głowa będzie pełnić funkcję przenośnika np. butelek, to wymaga solidnego i znacznie dłuższego chwytu niż marna 1 sekunda.

Dodatkowy ale bardzo prosty do rozwiązania kłopot który widać na załączonych zrzutach ekranu to mała powierzchnia styku osi roboczej orczyka serwomechanizmu z dźwignią szczęki. Przy nawet ograniczonym momencie pojedynczego serwa do 20kg*cm cała ta siła naciska przy obecnej konstrukcji na powierzchnię wycinka obwodu 5mm osi orczyka i jej wysokości, dajmy na to 5mm. Nawet gdyby powierzchnia styczna do osi miała 0.5mm to widać że mamy do czynienia z naciskiem rzędu 20kg na (5mm*0.5mm) czyli to tak jakbyśmy położyli - uwaga ok 800kg na cm². Takiego nacisku to nawet Rosyjska stal typu gniotsja nie łamiotsja nie wytrzyma za długo.

Tak więc... projekt idzie do przodu, ale serwo w aktualnej konfiguracji zachowuje się jak pracownik na minimalnej krajowej - potrafi sporo, ale tylko przez chwilę, a potem chce przerwy.



Sokole oko jedynie dostrzeże wszystkie te wymiary, ale zapewniam że nie warto się wpatrywać :) To jest jedynie screen ukazujący ilość wymiarów jakie musiałem podać aby ze zdjęcia zrobić szkic konstrukcyjny. Jak dostanę oczopląsu to będę wiedział przynajmniej dlaczego.



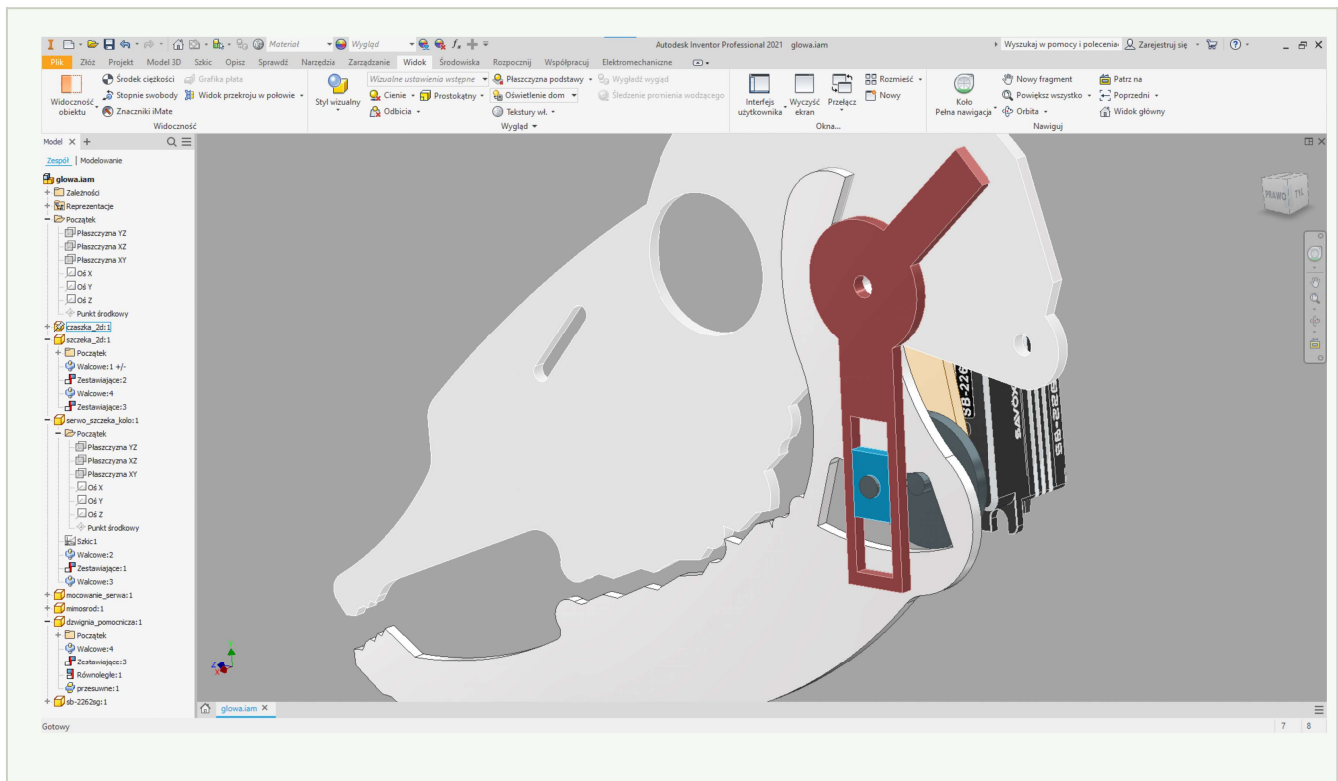
Po ukryciu całej tej bazgraniny z wymiarami i niedużym „wyciągnięciu” odpowiednich warstw oraz dodaniu odpowiednich zależności, coś w końcu normalnego widać. Wersja mechanizmu zaciskania szczęki z umieszczonym serwow tak by moment z orczyka przekazywany był na dźwignię szczęki 60mm od jej osi obrotu.

19. Sprężyna młodości

Po wczorajszym odkryciu czekających mnie problemów, dziś wpadłem na rozwiązanie, które może zrewolucjonizować cały układ. Zamiast katować oś orczyka i liczyć na to, że rosyjska stal „gniotsja nie łamiotsja” jednak się nie ugnie, postanowiłem, że serwo dostanie godniejsze życie aby nie starzało się tak szybko jak ja :) - niech tylko pracuje tyle, ile trzeba, a całą resztę roboty wykona za niego mechanika.

Pomysł jest taki: Serwo za pomocą orczyka będzie ciągnąć dodatkową dźwignię, a ta w końcowej fazie automatycznie się zablokuje. W czasie pracy dodatkowej dźwigni pałeczkę przejmuje sprężyna, która dociska szczękę z odpowiednią siłą i nie ma znaczenia, czy w pysku będzie listek, czy butelka po oranżadzie. Nawet jeśli serwo się wyłączy, szczęka nadal będzie zaciśnięta - czyli dokładnie to, o co mi chodziło. Dzięki temu serwo nie musi już trzymać całego obciążenia przez czas nieograniczony, a co za tym idzie - nie będzie ani się grzać, ani palić, ani strajkować po kilku sekundach pracy, co więcej dużo dłużej zachowa swoją „młodość”, a i trybki nieco mu odpoczną. To trochę tak, jakby zatrudnić kogoś tylko do dokręcenia słoika - raz zakręci, a resztę trzyma gumka recepturka :)

Zmodernizowany mechanizm wrzucam w załączniku jako zrzut ekranu - wygląda zdecydowanie bardziej „inżyniersko”, a w głowie mam już pomysły, jak ten system rozwinąć dalej. Oczywiście każde rozwiązanie ma swoje minusy, więc jaki jest minus tym razem? A więc rodzi się pytanie: Gdzie i jak umieścić sprężynę, a dodatkowy problem jaki już widzę to to że sprężyna będzie w zależności od wielkości obiektu mniej lub więcej go ścisnąć - hmm...



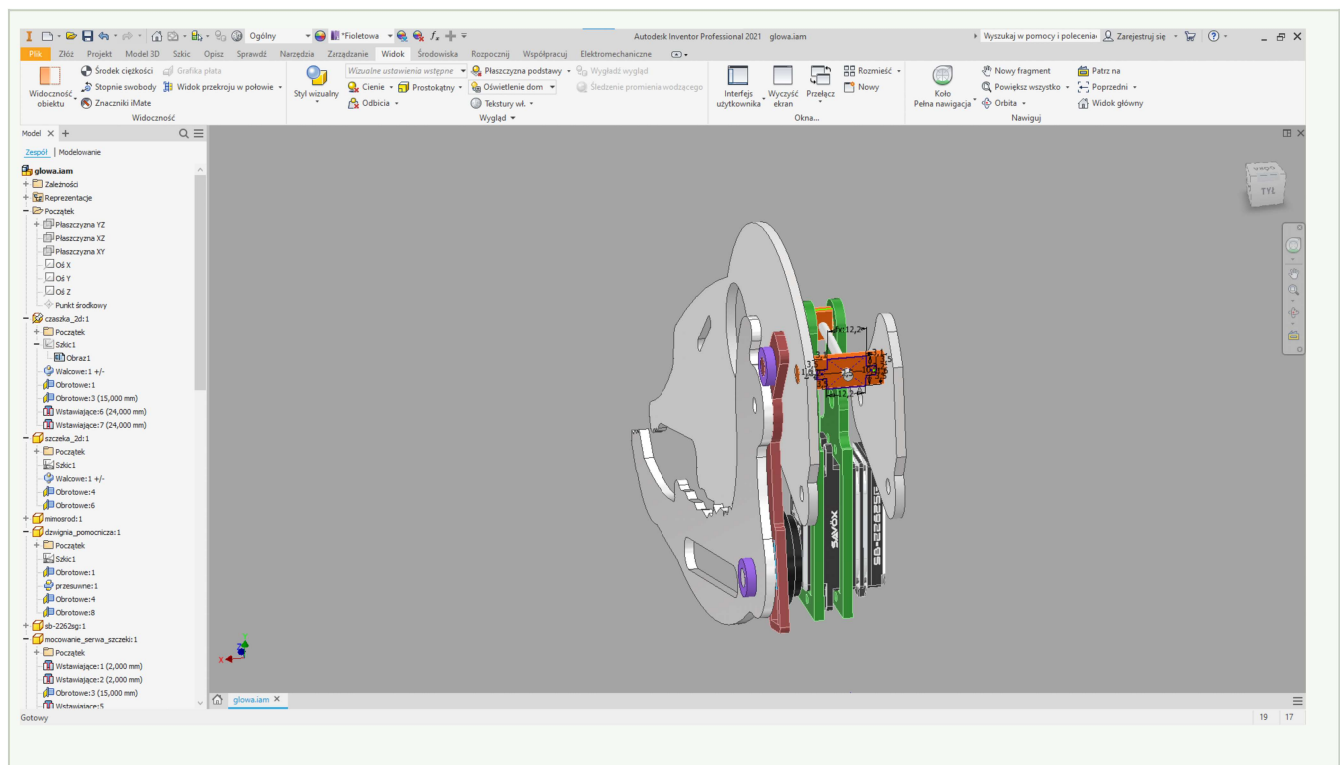
Zmodernizowany mechanizm zamykania szczęki. Od firmy Savox otrzymałem model 3D serwa co nieco ułatwiło pracę.

20. Złomowisko

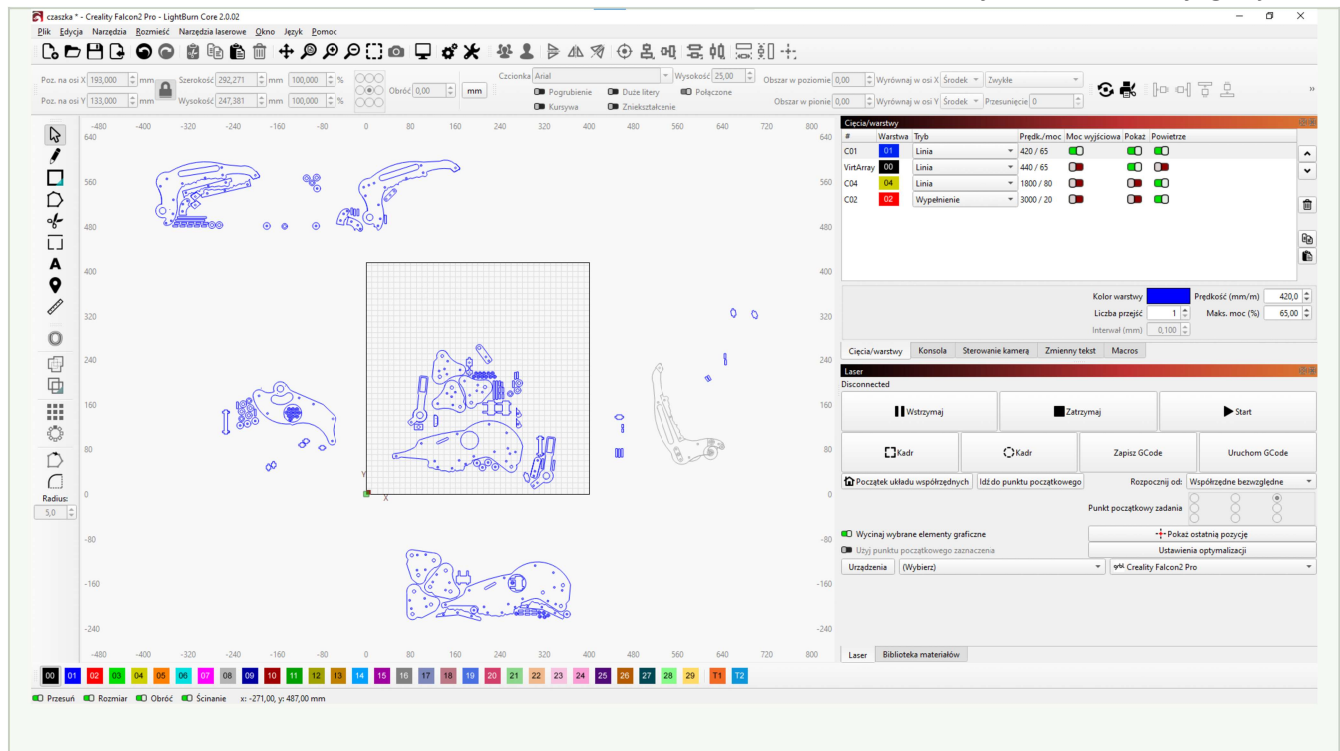
Stwierdziłem, że serwomechanizmowi trzeba zrobić nie tylko pomoc w postaci pomocnej sprężyny, ale i porządną naciąg stanowiący automatycznie regulację kąta serwa w stosunku do czaszki. W Autodesk Inventor doszukiwałem do modelu kolejne elementy tak, żeby wszystko tworzyło spójną całość. Skupiłem się także na takim zaprojektowaniu elementów aby można było je docelowo wyciąć na laserze w skleje, po to aby coś w końcu zrobić fizycznie. Pojawiły się więc nowe mechaniczne połączenia, trochę wzmocnień, no i najważniejsze: śrubowy naciąg serwomechanizmu. Po co? A no dlatego, że nie mam bladego pojęcia, jaką sprężynkę wygrzebię na moim złomowisku.

A jak wiadomo - złom rządzi się swoimi prawami: dzisiaj znajdziesz amortyzator od malucha, a jutro sprężynkę od długopisu. Dlatego trzeba było zrobić mechanizm regulowany, żeby serwo i sprężyna żyły w zgodzie, niezależnie od tego, z jakiego „rodzaju stali szlachetnej” (czytaj: złomu) będzie zrobiona. A tak bardziej na poważnie, to mocno się to przyda do regulacji która skompensuje niedoskonałości mechanizmu, gdyż w tym przypadku nie muszę i nie chcę wszystkiego obliczać, bo i tak jest to tylko test działania wymyślnego mechanizmu a nie gotowy model głowy czy chwytaka. Więc zamiast liczyć wszystko co do Newtona, buduję regulację, która sama „przyjmie” różnice i niedoskonałości. Po co inżynieria kosmiczna, skoro można zrobić prosty patent, który tu na pewno zadziała?.

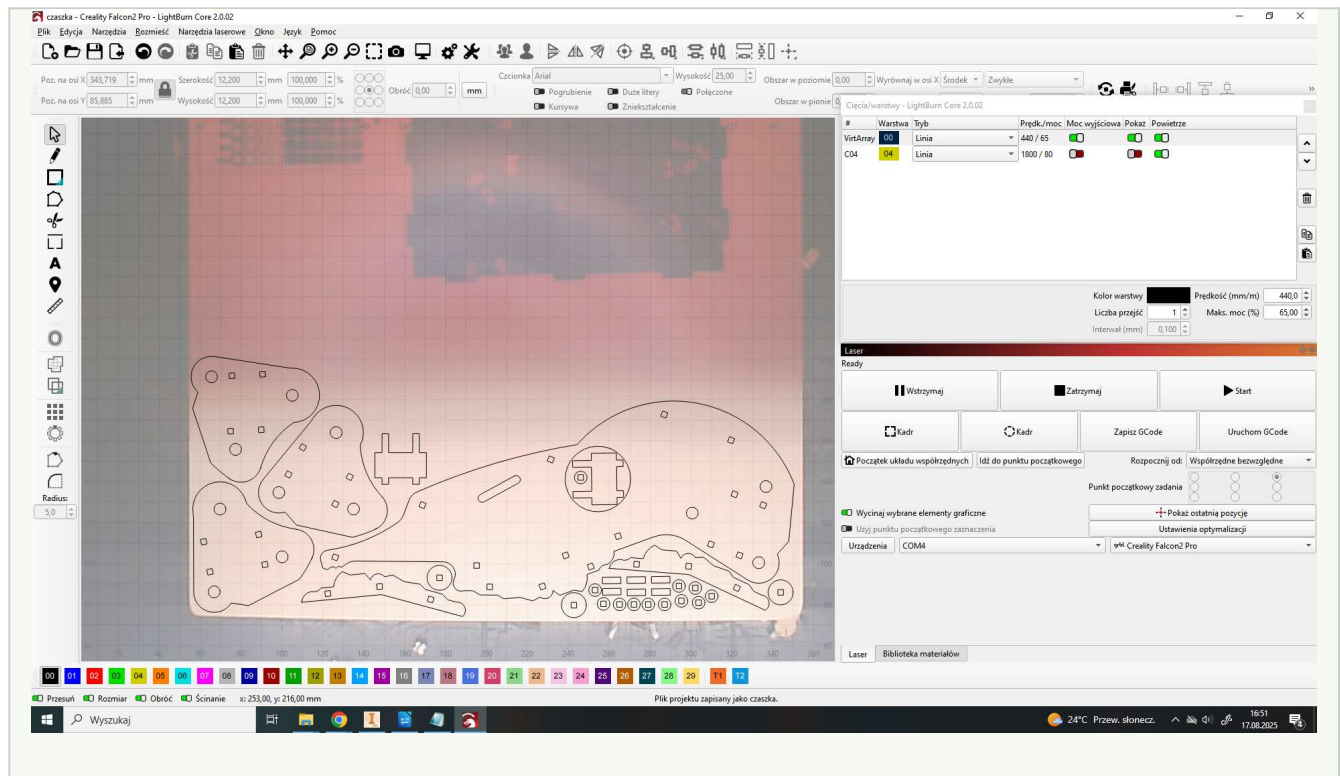
Również w czasie rozważań nad naciągami ubzdurałem sobie miejsce na sprężynę - w szczęce. Jak wiadomo czaszka ma swoje wymagania i musi jeszcze pomieścić inne bajery. Żeby więc to wszystko pogodzić, musiałem dodać cały system rolek i spiąć to z pomocniczą dźwignią. To właśnie te rolki prowadzą siłę sprężyny tam, gdzie trzeba. Trochę liczenia przełożeń, trochę kombinowania - i powstał układ, który w głowie wygląda zaskakująco obiecująco.



Dodane elementy do naciągu serwa, oraz rolki dla odpowiedniego przekazania siły sprężyny, która będzie umieszczona w szczęce.



LightBurn po zaimportowaniu odpowiednich części z Inventora



A tutaj umieszczanie elementów tak by zużyć jak najmniej sklejk.

22. Wyjście z Matrixa

Dziś nastąpił przełom - wyszedłem z cyfrowego Matrixa i w końcu przeszedłem do świata fizycznego. Koniec patrzenia w ekran, kombinowania w Inventorze i poprawiania wymiarów w LightBurn. Przyszedł czas na to, co da się lubić bardziej - smród palonej sklejkki i prawdziwe mechaniczne dłubanie.

Pierwsze cięcia poszły gładko. Zaczynałem od montażu pod serwo i śruby naciągowej, bo bez tego reszta nie miałaby sensu. Potem na warsztat poszła baza, czyli czaszka z elementami usztywniającymi. Kiedy już miałem solidny „fundament”, dorzuciłem zęby - bo bez nich czaszka wyglądałaby, jakby była po wizycie u kiepskiego dentysty. Na deser została szczeka, która jak i inne elementy po raz pierwszy trafiła z wirtualnego modelu do realnego świata.

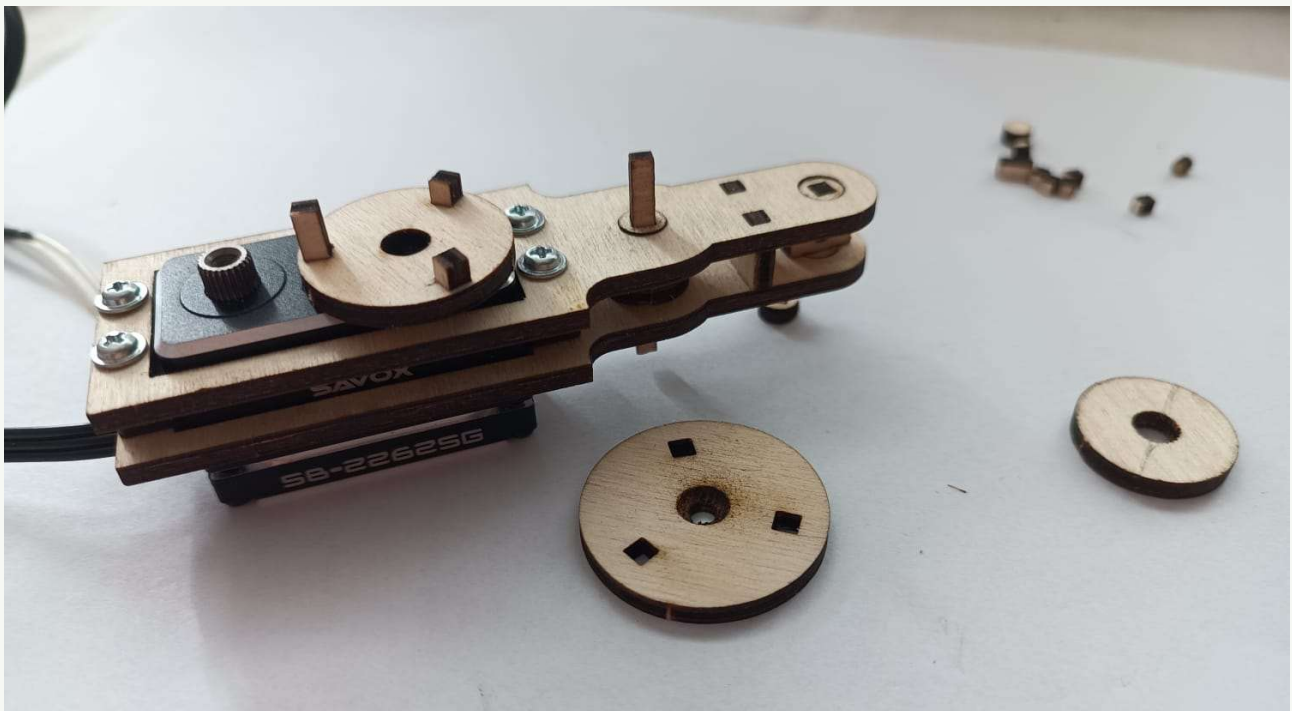
Wrażenia? Ależ tak i to kilka:

Zapach wypalanej sklejkki - niby drażniący, ale dla mnie był to zapach postępu.

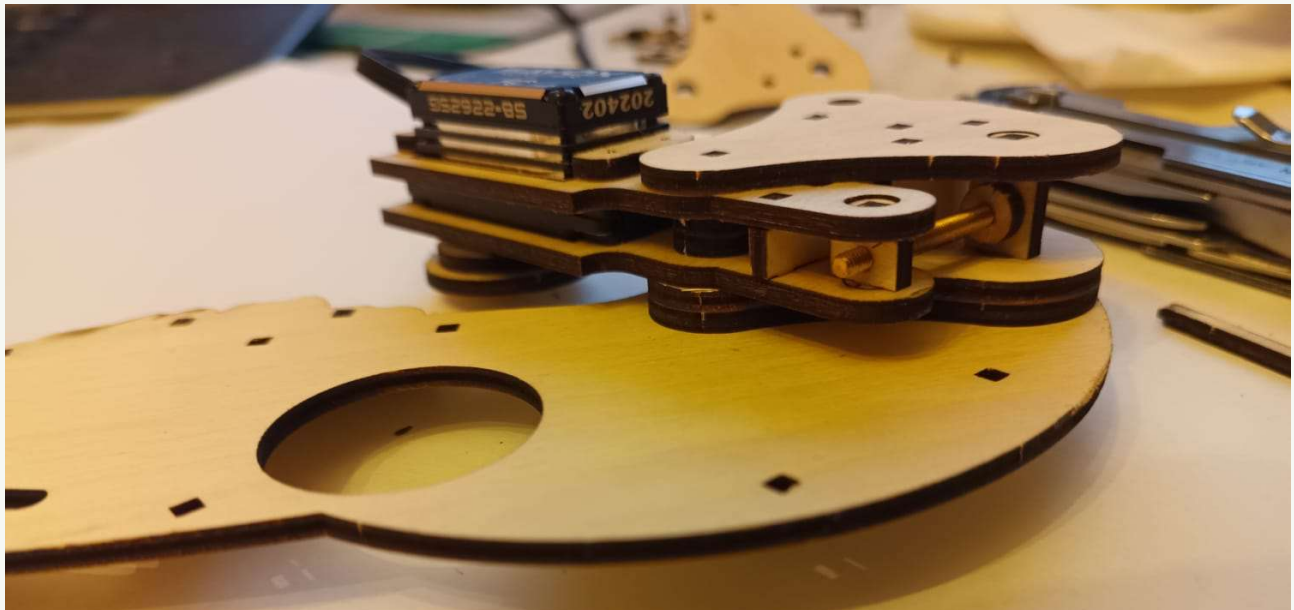
Precyzyjnie spasowane wpusty montażowe - aż przyjemnie się je wkłada (jak by to nie zabrzmiało:), nie ruszają się i trzymają solidnie elementy pozycjonując je na swych miejscach. Warto było poświęcić sporo czasu na bardzo precyzyjne wymiarowanie części.

Pierwsze realne zarysy konstrukcji - nagle coś, co było tylko w komputerze, zaczęło istnieć na biurku. Niby nic szczególnego ale jednak jak zwykle cieszy.

To był dzień, w którym projekt zaczął naprawdę żyć - i choć to dopiero kropla w morzu porównując do tego co chciałbym zbudować, to i tak powolutku całość nabiera charakteru.



Obudowa dla serwa wraz z częścią mechanizmu napinającego. Orczyk, aby śruba się schowała został w centralnej części 3-krotnie wygrawerowany - co przepaliło sklejkę o ponad 2mm umożliwiając schowanie łba śruby.



Mechanizm naciagowy - jak ukończę jakimś cudem ten projekt, będzie służył do regulacji siły zgryzu i będzie sterowany innym serwem. Teraz regulacja manualna śrubokrętem musi wystarczyć.



Artystyczny nieład i składanie puzzli.



Gotowy łaniogłowy chwytak :)

23. Sklejka to nie stal

Jak to zwykle bywa - teoria teorią, a prototypowania zaraz weryfikuje wszystkie piękne założenia. Okazało się, że 3-milimetrowa sklejka nie ma w sobie ducha stali i przy większym naprężeniu linki, osie po prostu odmawiały współpracy. Wystarczyło kilka ruchów mechanizmu, żeby pokazały, kto tu rządzi - i nie były to one.

Nie było więc innej opcji, jak zabrać się za przeróbki. Najwięcej pracy wymagały:

- szczęką, która dostała nową, mocniejszą geometrię przy osi,
- dźwignia pomocnicza, bo to ona przenosiła znaczną część obciążeń,
- rolki, które dostały solidne wzmocnienia, aby linka mogła pracować bez rozrywania konstrukcji.

Ostatecznie powstała wersja, która w końcu mnie satysfakcjonuje. Sporo elementów trzeba było wyciąć i poskładać na nowo, więc zabawa w składanie zaczęła się od początku. Jednak efekt końcowy wynagradza wszystko - mechanizm działa, wygląda stabilnie i przede wszystkim przestał się buntować przy mocniejszym naciągu. Sterowanie serwem na potrzeby testów odbywało się za pomocą modelarskiego odbiornika RC.



Ulepszona wersja głowy.

| 24. Walka z bykiem

Dziś na warsztat trafił ten nasz przysłowiowy mały chudy ale byk w postaci serwomechanizmu modelarskiego, który - jak się okazało - wcale nie zachowuje się jak zwykle, leniwe serwo 50 Hz z klasycznym sygnałem 1-2 ms i kątem rzędu 90 stopni. Tutaj mamy trochę inny świat:

- ~140° ruchu przy szerokości impulsu PWM 800-2200 μ s
- 90° przy 1000-2000 μ s
- 1500 μ s - pozycja neutralna
- pasmo martwe: 3-5 μ s (!), czyli bardzo czułe na sterowanie
- zakres częstotliwości: od 50 aż do 333 Hz

Wniosek? Ten byczek daje o wiele więcej niż standardowe serwo - wystarczy umieć to wykorzystać. Ponieważ nie ma co się bawić w półśrodki i szukać sterowników które umożliwią mi pełną kontrolę, postanowiłem stworzyć własny sterownik, który pozwoli wyciągnąć z serwa pełne możliwości, a w przyszłości oprogramowanie może posłużyć do sterowania wieloma serwami jednocześnie. Ale... że lenistwo to najlepszy motor postępu, zamiast wymyślać tylko na chwilę koło na nowo, nie będę projektował od zera płytki PCB, gdyż czas mnie goni, a jedynie sięgnę po moją starą, ukochaną płytkę ZL27ARM. To na niej pisałem swoje pierwsze programy na STM-a, a pinout znam prawie na pamięć - więc idealna baza do szybkiego prototypu sterownika który spełni poniższe życzenia:

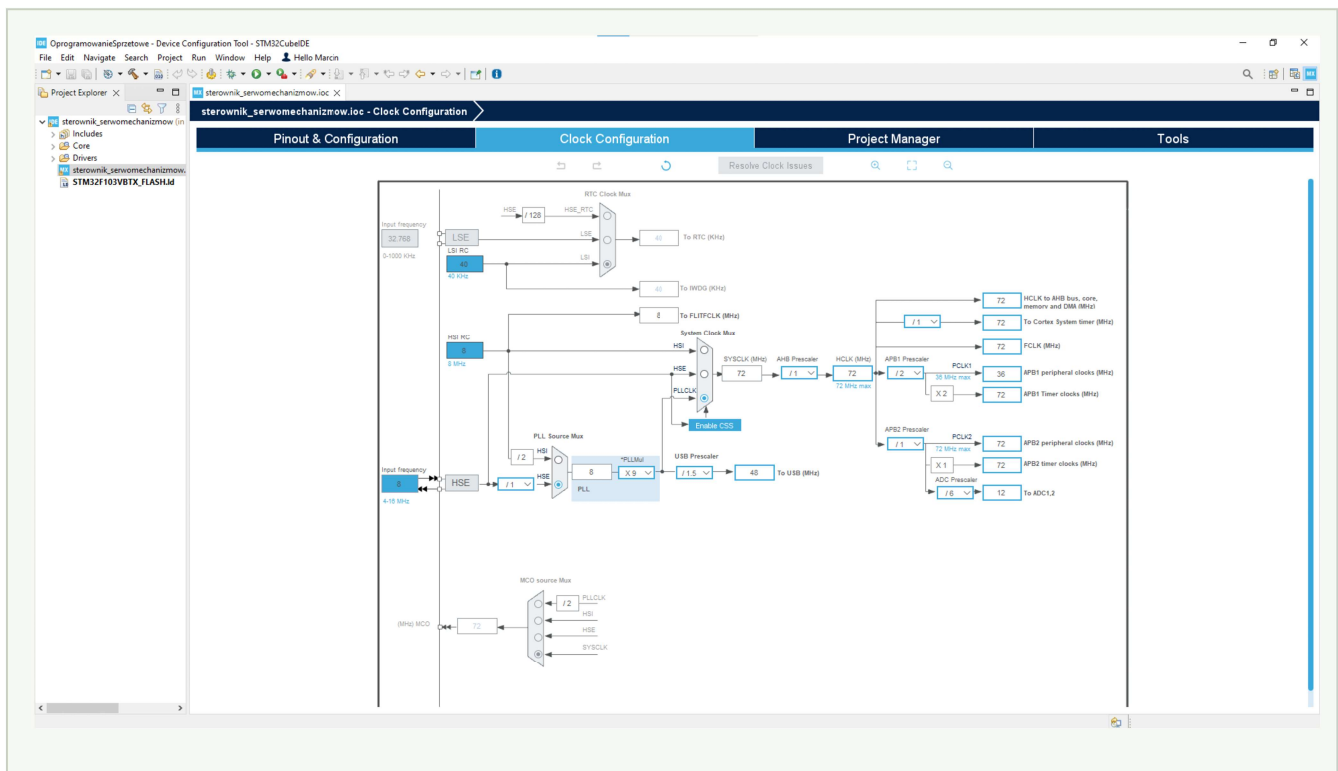
- Wyjście: generacja PWM dla serwa z możliwością ustawienia częstotliwości i szerokości impulsu zgodnie z dokumentacją serwa.
- Wejście: USB, bo łatwo sprzęgnąć to z PC na którym będzie program sterujący.
- Komunikacja: jednostronna (PC → sterownik). Na potrzeby prototypu bez CRC i bez szyfrowania, bo nie ma sensu komplikować pierwszej działającej wersji.
- Logika działania:
 - oczekuj na przerwanie od sprzętowego USB jeśli ma coś w buforze wejściowym.
 - zdekoduj „tajne znaki” z informacji wejściowej.
 - przelicz dane na odpowiednie wartości rejestrów dla PWM.
 - wypływ PWMa dla serwomechanizmu.

Dziś zrobiłem pierwszy krok jeśli chodzi o stworzenie oprogramowania sterownika - wymyśliłem wymagania i zarys sterownika który ma być zgodny z specyfikacją serwa. Zaktualizowałem oprogramowanie STM32CubeIDE. Teraz czas na implementację na STM-ie i sprawdzenie, czy byk rzeczywiście da się ujarzmić.

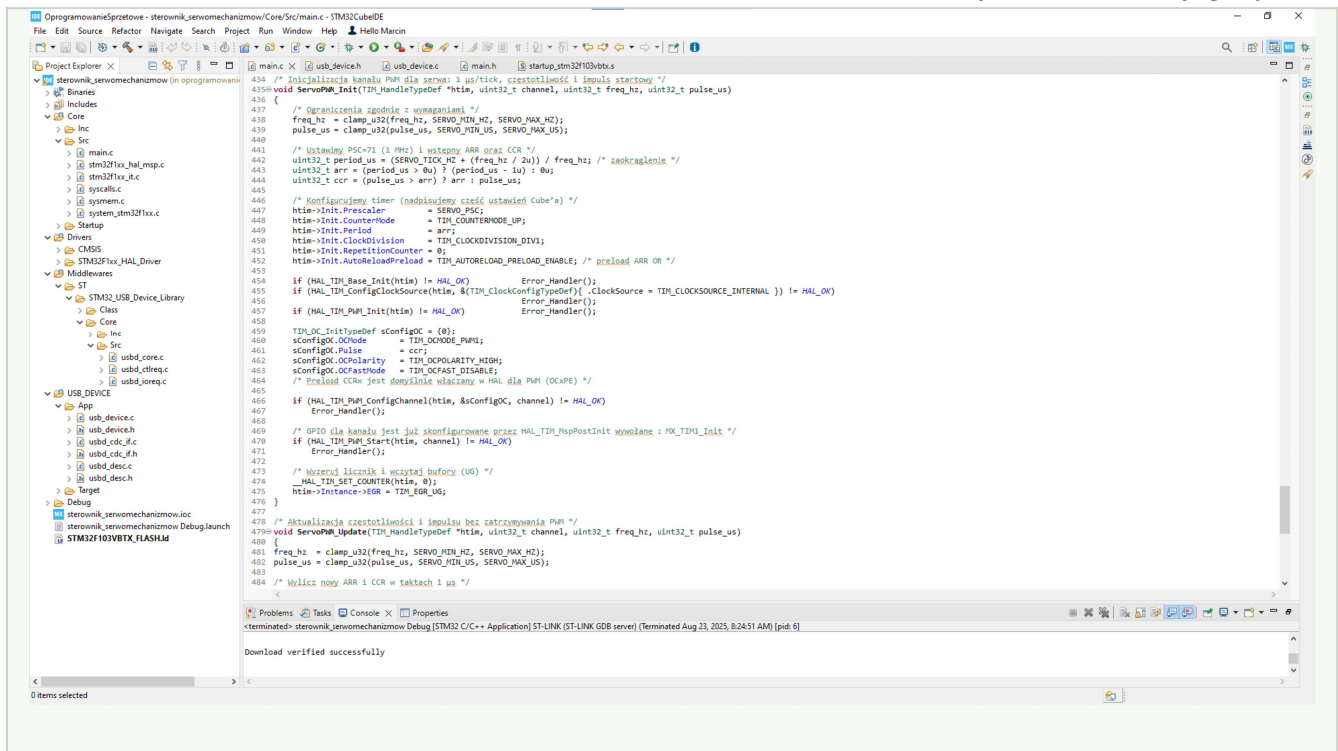
25. Treser

Z dokumentacji wynika, że serwo może pracować maksymalnie przy częstotliwości 333 Hz, czyli minimalny okres sygnału PWM to 3 ms. Impuls musi się mieścić w widełkach 800 - 2200 μ s, a punkt neutralny wypada przy 1500 μ s, co daje około ± 70 stopni wychylenia. Jako poganiacza wybrałem starego znajomego - stm32 z kwarcem 8 MHz i PLL podkręconym do 72 MHz. Tak wiem że 72MHz to nieco przewymiarowane, trochę jak zakładanie turbiny do malucha - niepotrzebne do samego sterowania serwem, ale fajnie mieć zapas, gdyby wpadło mi do głowy liczyć w locie jakieś kinematyki czy inne wygibasy w samym sterowniku.

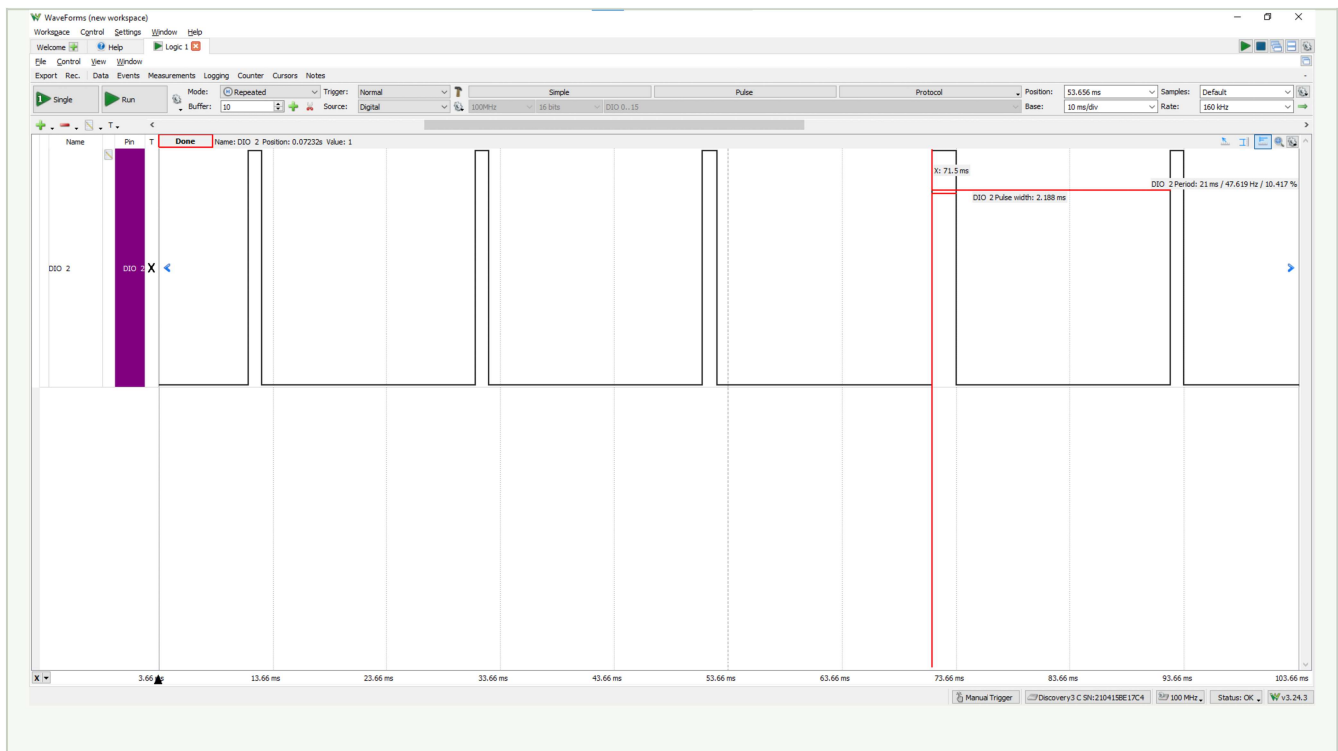
Poustawiałem więc rejestry tak, żeby timer na dzień dobry wypływał 50 Hz i impuls 1500 μ s. To taki uniwersalny sygnał powitalny: „Dzień dobry serwomechanizmie, oto twoja pozycja zero”. Dzięki temu każde, nawet najbardziej marudne serwo, powinno stanąć grzecznie w neutralnej pozycji. Napisałem wstępny kod do obsługi PWM i... nie do końca zadziałał. Sygnał powitalny działa super ale przy aktualizacji długości impulsu procesk zmienia na czas jednego okresu samą częstotliwość sygnału i co gorsze wypełnienie, więc szykuje się mała poprawka.



Ustawione zegary taktujące rdzenia STM32 oraz jego peryferii, w tym Timera i interfejsu USB



Część kodu odpowiedzialna za wstępną inicjalizację Timera 1 w trybie PWM.



Jak widać z wykresu popełniłem jakiś błąd - w czasie zmiany wypełnienia (czerwona pionowa linia) jeden kolejny okres sygnału oraz jego wypełnienie zmienia się co jest niedopuszczalne.

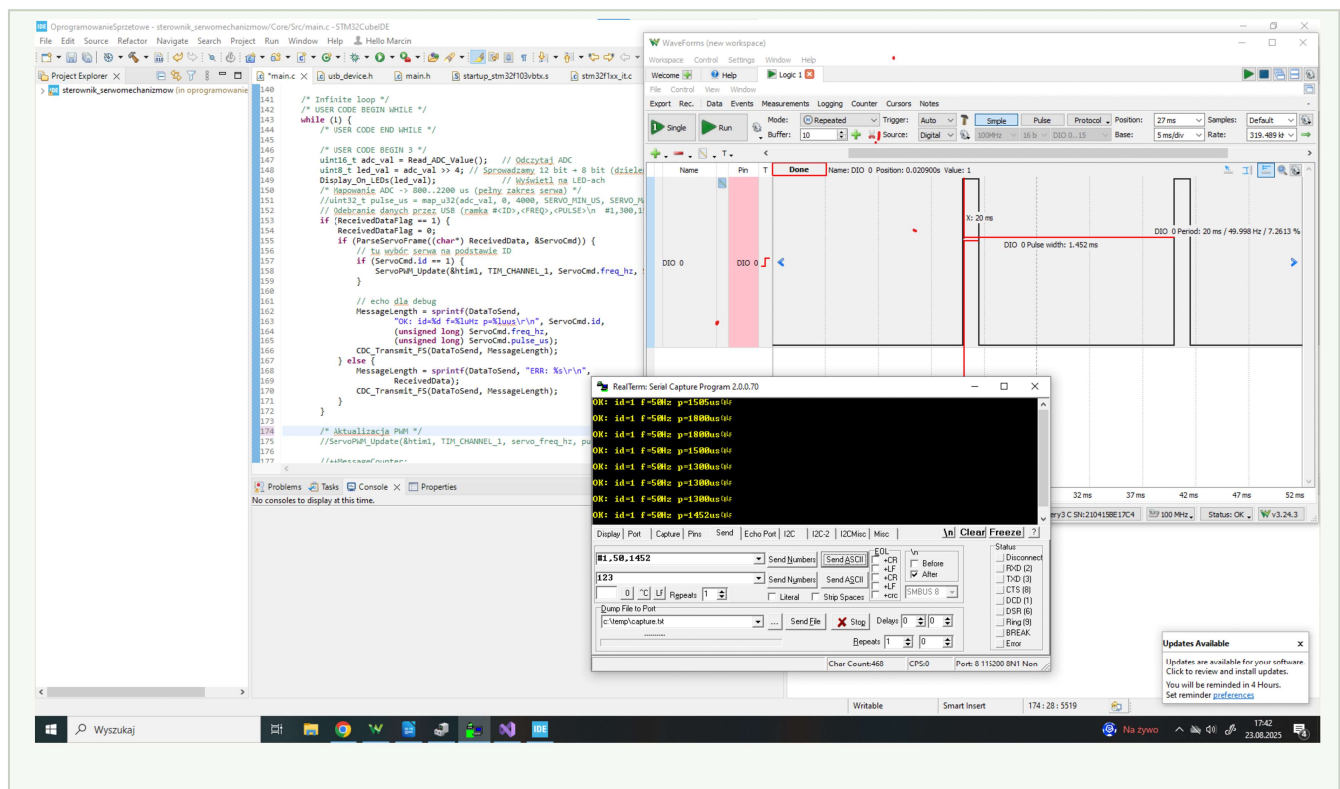
26. Debugowanie

Na początek mała dygresja - wcześniej oczywiście testowałem serwo na zwykłym odbiorniku RC. Problem w tym, że taki odbiornik daje klasyczny sygnał: 50 Hz i impuls w zakresie 1-2 ms. To działa, ale zupełnie nie wykorzystuje możliwości mojego serwa, które przecież potrafi znacznie więcej.

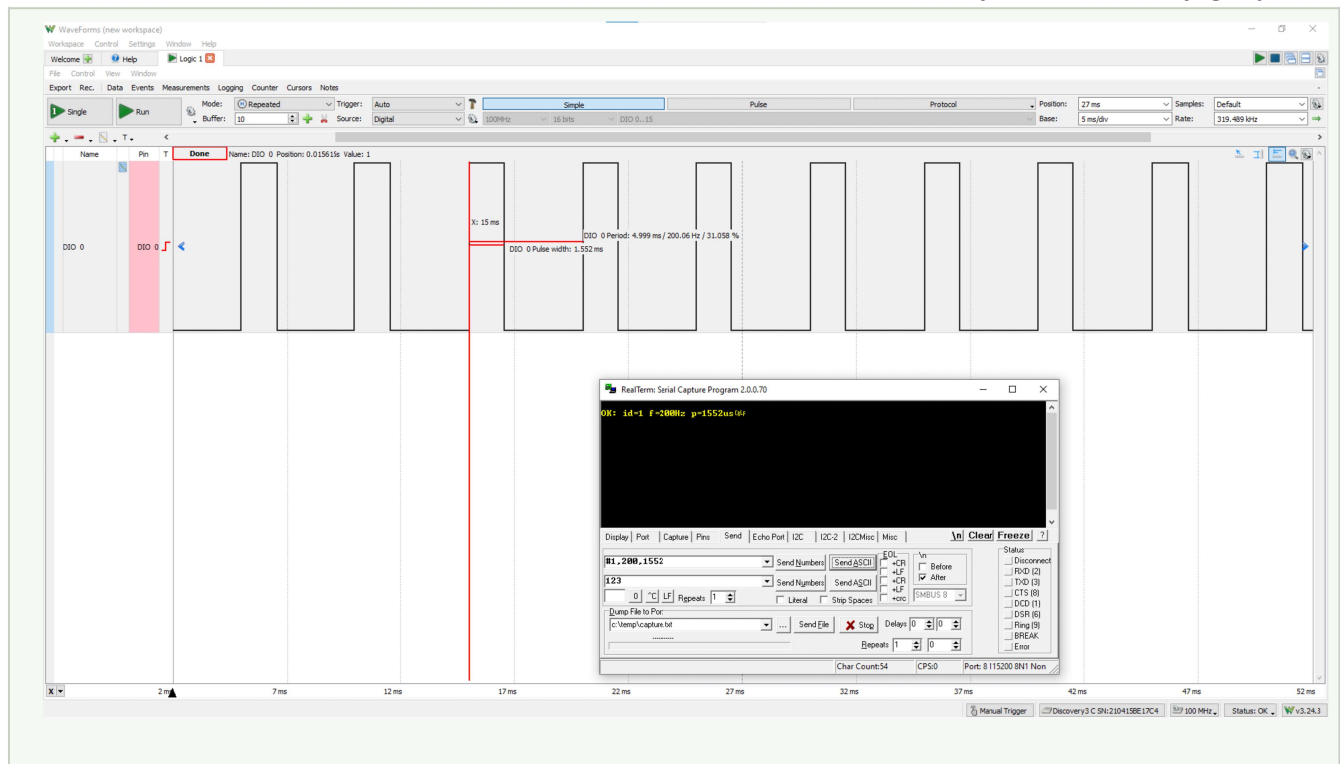
Dzisiaj poprawiłem błąd w kodzie, przez który przy nadpisywaniu rejestrów timera zmieniał się nie tylko okres, ale i wypełnienie PWM. Do tego dorzuciłem odczyt z ADC (na razie testowy), odpaliłem USB w trybie VCP i zrobiłem porządną refaktoryzację sterownika.

Komunikacja PC-sterownik opiera się na prostej wymyślonej na poczekaniu ramce protokołu RS232: #<nr_serwa>,<czestotliwosc>,<szer_impulsu>/n - komputer mówi wprost, a sterownik wie, co robić, bez zbędnych interpretacji. Tak wiem że nie ma CRC i dane są w zwykłym ASCII ale szkoda mi teraz czasu na zabawy z wymyślaniem lepszego protokołu komunikacyjnego.

Efekty? Sterownik trzyma częstotliwość i szerokość impulsu idealnie, a do tego rozdzielczość sterowania i responsywność serwa zauważalnie wzrosły. W porównaniu do odbiornika RC różnica jest ogromna - wreszcie mogę wykorzystać pełne możliwości serwa. Oczywiście, pasowałoby w przyszłości zmienić ramkę na binarną, żeby dane szybciej przelatywały przez RS-a, a tak naprawdę to pewnie zmienię protokół na CAN lub I2C ale to już małe piwo i temat na później.



Przykład z utrzymywaniem 1452us i 50Hz na PWM-ie



I kolejny screen z pomiaru generowanego sygnału na podstawie zadanego z USB

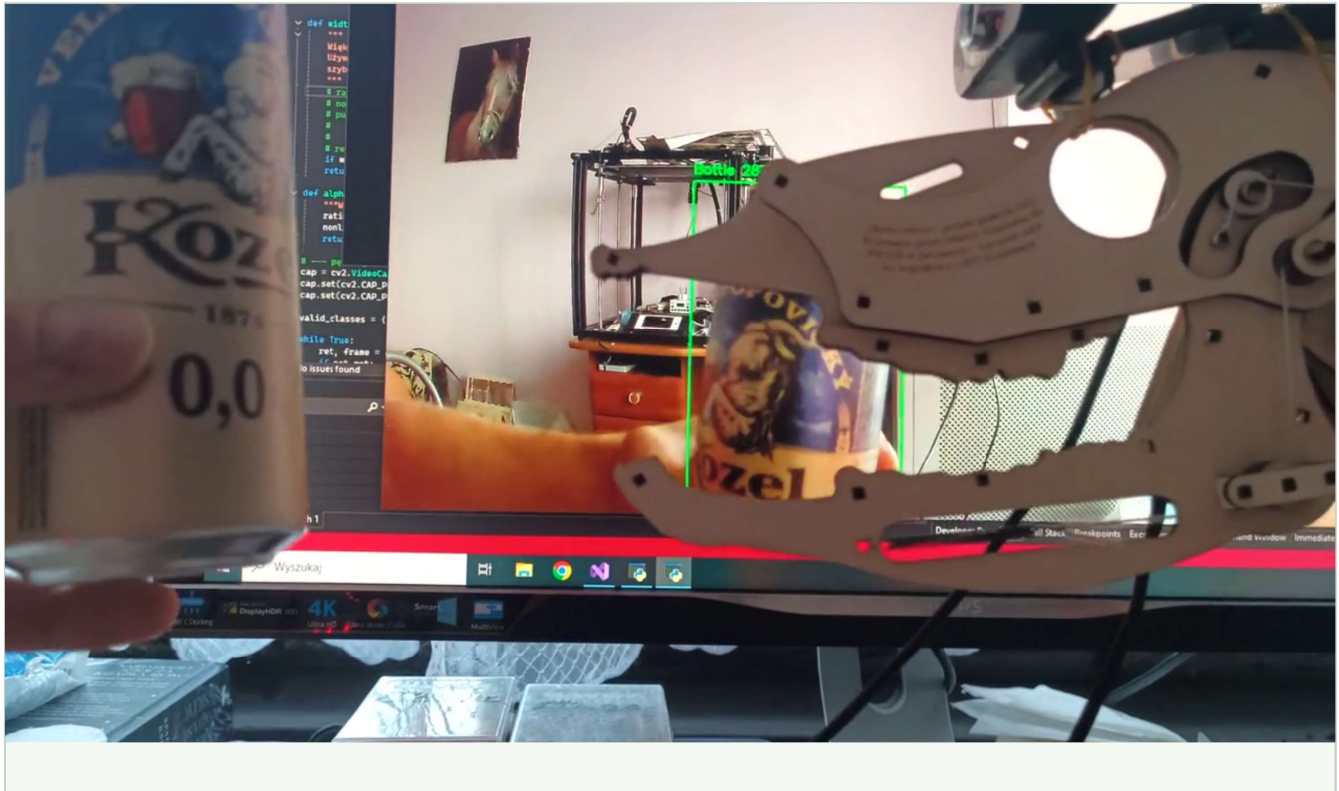
Na powyższych fotkach widać działanie analizatora Analog Discovery 3 - świeża zabawka w warsztacie, która szybko udowodniła, że potrafi robić za pół laboratorium: analizator stanów logicznych, oscyloskop, programowany zasilacz i generator w jednej czarnej skrzynce pod USB. Muszę z niechęcią przyznać, że mój ukochany oscyloskop z czasów Gagarina, C1-99, chyba częściej będzie teraz zbierał kurz niż dane.

27. Pijaczka gadaczka

W końcu nadszedł ten moment, kiedy połączyłem oprogramowanie do detekcji obiektów typu butelka z nowo powstałym sterownikiem szczęki. Algorytm - tak jak pisałem wcześniej - analizuje obraz z kamery, wyszukuje zadany obiekt i w zależności od odległości odpowiednio szeroko otwiera szczękę. Im bliżej obiekt, tym szerzej paszcza się rozdziawia :). Pod ręką miałem puszkę po piwie i muszę przyznać, że wyszło całkiem nieźle - tym bardziej że model sieci do rozpoznawania obiektów jest jednym ze słabszych i wcale nie był uczony na puszkach. A jednak pysk robota w miarę sensownie reagował na pojawiającą się puszkę i „łapał” ją. Tego jeszcze nie grali, żeby robota poić piwem. Może Kozel w ramach reklamy wyśle mi kontener puszek - tyle że pełnych - przydałyby się na żniwa!

Spory kawałek dnia poświęciłem też na inne zastosowanie szczęki - już nie jako chwytaka, ale jako głowy do projektu Zervidjanny, ruszającej paszczą w takt mówienia. I tu pojawiły się prawdziwe schody: przetwarzanie w czasie rzeczywistym tablic audio, uśrednianie, dobór parametrów... słowem cała orkiestra rzeczy, które trzeba było ogarnąć, żeby efekt nie wyglądał jak discopolowe karaoke. Ostatecznie wyszło zaskakująco dobrze - szczególnie jak na kod klepany „na kolanie”.

Nie ma jednak sensu udawać, że wszystko da się zrobić od razu. Zarys projektu uznaję za zamknięty sukcesem: powstał prototyp głowy, działa podstawowa reakcja na obiekt, a szczęka może pracować także jako element wizualizacji mowy.



28. Wniosek taki

Na koniec muszę przyznać, że cały etap był dla mnie sporym zaskoczeniem. Wiele projektów robiłem wcześniej, ale rzadko który powstał tak szybko i łączył w sobie tak pełną ścieżkę: od pomysłu, przez dobór komponentów, skanowanie 3D, modelowanie, cięcie laserem, programowanie, aż po spójny efekt końcowy.

Dużą różnicę zrobił brak sztywnego schematu. Był tylko zarys i ogólny kierunek, a reszta zależała od kolejnych problemów technicznych. Dzięki temu powstało rozwiązanie, które jest jednocześnie dziwne, praktyczne i bardzo „zervidjannowe”. Projekt łączy temat rozpoznawania obiektów z mechaniką głowy robota, więc przyda się bezpośrednio w dalszej pracy nad Zervidjanną.

Podsumowując - z prostego prototypu powstała mała przygoda inżynierska, z której zostało sporo wiedzy, doświadczenia i satysfakcji. Przy okazji powstały prototypy chwytaka oraz sterownika, które - choć robione „na kolanach” - działają i pokazują, że kierunek myślenia nie był wcale taki zły.

Dziś roboty mogą w locie łapać butelki, a wkrótce mogą zacząć same dopominać się o piwo. Wtedy trzeba będzie opracować system antyalkoholowy dla Zervidjanny - żeby przypadkiem nie zniknął cały zapas przeznaczony na żniwa :)

Skrót techniczny aktualnego prototypu

Obszar	Rola w prototypie
Mechanika	głowa ze sklejk, mechanizm szczęki, sprężynowe wspomaganie docisku, regulacja naciągu i testy wytrzymałości elementów.
Widzenie maszynowe	kamera, detekcja obiektów, segmentacja, punkty charakterystyczne twarzy oraz przygotowanie danych do sterowania chwytakiem.
Uczenie osób	embedding twarzy jako wektor w przestrzeni latentnej; porównywanie profili przez odległość euklidesową i aktualizacja bazy znanych osób.
Sterowanie	sterownik STM32 z PWM dla serwa, komunikacja z PC przez USB/VCP i możliwość dalszego rozwoju protokołu.
Integracja	reakcja szczęki na obiekt z kamery oraz próba ruchu paszczy w rytm mowy syntetycznej TTS.