Aktualności

To jest sekcja do której przemigorwałem wszystkie moje wpisy z bloga.

Najnowsze wpisy w tej sekcji

Ludzie którzy wpłynęli na kształt współczesnej informatyki


źródło: Wikimedia.org

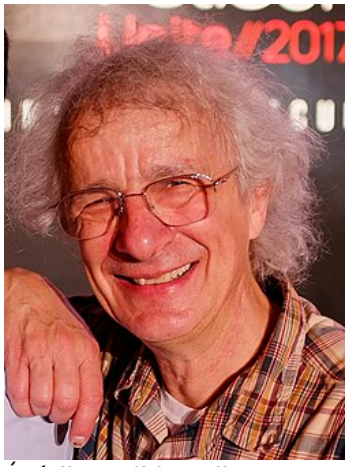
Ten artykuł jest moim subiektywnym spisem osób o których mało się mówi nauczając historii informatyki, lub mało się słyszy o tych osobach w mediach. To zestawienie to zbiór interesujących osób które moim zdaniem miały większy wpływ na informatykę niż się nam wszystkim wydaje. Wybrałem tutaj osoby o których swego czasu czytałem bardzo dużo i zaciekawiła mnie ich historia lub dokonania w świecie informatyki. Sporo z tych osób jest wymienione tutaj ze względu na swój wpływ na społeczność hackerską, sam jestem zwolennikiem kultury i zasad hackerskich o czym można przeczytać na stronie głównej tej wiki. Zbiór nie ma konkretnej kolejności, jest podzielony na kategorie tematyczne, które opisują na jaką sferę informatyki miały wpływ osoby w nich wymienione.

Świat kultury hackerskiej i cyberbezpieczeństwa

Clifford Stoll

Jest on osobą wyjątkową z punktu widzenia informatyki, gdyż nie jest informatykiem, czytelnik może teraz zadać pytanie to co w takim razie on tutaj w ogóle robi, zaraz wszystko opiszę. Zacznijmy od zainteresowań Clifforda. Jest on z wykształcenia Astronomem, aktualnie zajmuję się tematami związanymi z starymi kalkulatorami mechanicznymi i

elektromechanicznymi, matematycznymi zagadnieniami topologii dla kanału Numberphile na Youtube, natomiast w 1986 roku pracował w Lawrence Berkeley National Laboratory gdzie był administratorem systemów, i tutaj zaczyna się nasza historia.


Źródło: Wikimedia.org

Pewnego dnia w 1986 roku przełożony poprosił Clifforda aby ten sprawdził i zdiagnozował problem który powoduję błąd w zaksięgowaniu 75 centów przez system księgowy. Clifford wyśledził błąd do miejsca w którym znalazł wpisy o logowaniu nie autoryzowanego użytkownika, który logował się na komputery laboratorium i korzystał z nich przez 9 sekund i nie zapłacił za ten czas. Clifford po dalszej analizie znalazł że tym nie autoryzowanym użytkownikiem był ktoś z uprawnieniami super usera w systemie unix (odpowiednik administratora w systemach Windows), po dogłębnej analizie udało się wywnioskować że dostęp do takich uprawnień dostał poprzez wykorzystanie podatności systemu movemail pakietu GNU Emacs.

W roku 1986, nie było systemów wykrywania włamań do systemów ani nawet firewalli, system komputerowy w LBSL opierał się na wielkiej maszynie typu mainframe do której chodziło wiele linii szeregowy z modemów, do który wdzwaniali się użytkownicy chcący skorzystać z komputera. Clifford wpadł na pomysł aby zebrać z całego laboratorium wszystkie dalekopisy i podłączyć je do wszystkich wchodzących linii szeregowych do komputera. Clifford wziął wózek i w piątek wieczorem „pożyczył" od wszystkich pracowników ich dalekopisy, kolejno podłączył je do każdej z 50 linii szeregowych. Przygotował sobie śpiwór, kanapki oraz termos, a następnie położył się spać. Nagle w nocy obudził go charakterystyczny dźwięk drukarki która drukowała operacje wykonywane przez użytkownika. Jak możecie się domyślić współpracownicy w poniedziałek rano nie byli zadowoleni z zaistniałej sytuacji. Clifford następnie przeanalizował od którego z dostawców przychodzi ta linia, udało mu się znaleźć że tym dostawcą jest Tymnet z pomocą specjalistów z firmy Tymnet udało się zwęzić poszukiwania do call center w MITRE. Przez następne dziesięć miesięcy Clifford analizował wszystko co było możliwe natomiast z małym skutkiem, nie mógł znaleźć kim jest ten włamywacz. Udało się znaleźć Cliffordowi że łącze przez które łączy się nasz „bohater" ma prędkość 1200 baudów co oznacza że jest to prawdobnie linia telefoniczna przez którą wdzwania się intruder.

W poniedziałek rano Clifford oddał wszystkim ich dalekopisy natomiast zostawił jeden podłączony do linii przez którą łączył się hacker, tak żeby mógł obserwować wszystkie akcje przez niego podejmowane. Clifford zauważył że nieautoryzowany użytkownik próbuje się łączyć poprzez system LBSL do militarnych baz danych w Stanach Zjednoczonych, i wyszukuje w nich fraz „nuclear" albo „SDI". Przestępca również pobierał hasła użytkowników (prawdopodobnie aby stworzyć ataki słownikowe) i zostawiał konie trojańskie aby wykradać hasła. Clifford był zadziwiony tym że tak łatwo do wszystkich systemów mógł się ten przestępca dostać, wynikało to z tego że sporo z administratorów systemów nie zmieniało standardowych haseł w oprogramowaniu. Co bardziej zadziwiające cracker mógł czasami podłączyć się jako użytkownik gość bez hasła do baz militarnych.

Był to jeden z pierwszych – jeżeli nie pierwszy – przypadek nieautoryzowanego włamania do systemów komputerowych w historii. W dużym skrócie historia kończy się w ten sposób że Clifford zastawia pierwszy w historii honeypot (pułapkę) na crackera. Pułapka składa się z wpisów mówiących o tym że w LBSL założono nowy dział, który został założony z powodu kontraktu SDI (wszystko o czywiście fikcyjne). Pozwoliło to dowiedzieć się kim hacker jest oraz skąd się łączy okazało się że ma fikcyjne konto s systemach Niemieckiej poczty skąd się łączył, i że mieszka w Hanowerze.

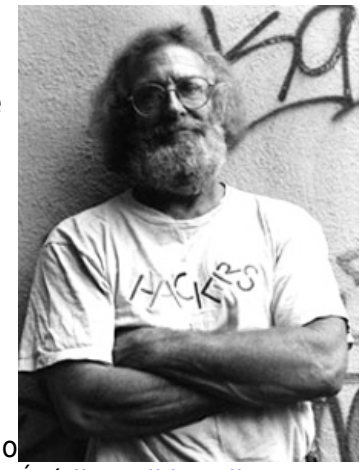
Jego imię i nazwisko to Markus Hess, i późniejsze śledztwo wykazało że od kilku lat jest zaangażowany w sprzedaż wyników swojego crackowania do ZSRR. Clifford poleciał do zachodnich Niemiec i złożył zeznania w procesie Markusa Hessa

Źródła:

- https://en.wikipedia.org/wiki/The_Cuckoo%27s_Egg_(book)
- https://en.wikipedia.org/wiki/Clifford_Stoll
- https://www.youtube.com/watch?v=1h7rLHNXio8
- stalking_the_wily_hacker.pdf

← techniczne szczegóły włamania

John Draper



Źródło: Wikimedia.org

Jest on osobą wyjątkową z punktu widzenia informatyki, ponieważ był jednym z pierwszych hakerów w historii, zanim jeszcze powstało to słowo w kontekście komputerów. Czytelnik może teraz zapytać: co takiego zrobił, że o nim piszemy? Sprawa zaczyna się bardzo nietypowo — od… gwizdka z pudełka po płatkach śniadaniowych.

John Draper, znany również pod pseudonimem Captain Crunch, był jedną z najbarwniejszych postaci wczesnych lat 70. XX wieku w świecie tzw. phreakingu — czyli hakowania systemów telefonicznych. Draper odkrył, że plastikowy gwizdek dołączony do pudełek płatków Cap'n Crunch wydaje dźwięk dokładnie o częstotliwości 2600 Hz. I teraz zaczyna się magia: dokładnie taki sygnał wykorzystywały amerykańskie centrale telefoniczne do oznaczania wolnego kanału. Dzięki temu, dmuchając w ten gwizdek przez słuchawkę, można było przejąć kontrolę nad systemem telefonicznym i dzwonić za darmo w dowolne miejsce na świecie.

Brzmi absurdalnie? Być może, ale tak właśnie było. Draper zaczął eksperymentować z tzw. blue boxami – elektronicznymi urządzeniami generującymi tony DTMF, które pozwalały manipulować centralami telefonicznymi. W praktyce pozwalało to dzwonić bez opłat międzynarodowo, łączyć się z numerami służbowymi, a nawet przejąć linię policyjną. To był czas, gdy systemy telefoniczne nie były w ogóle zabezpieczone przed tego rodzaju ingerencją – analogowe sieci opierały się w całości na zaufaniu do tonów i impulsów, które można było naśladować.

Draper nie tylko sam wykorzystywał blue boxy, ale także uczył innych, jak je budować. W tym miejscu warto wspomnieć, że wśród jego uczniów znaleźli się młodzi wówczas Steve Jobs i Steve Wozniak – założyciele Apple. Wozniak był tak zafascynowany możliwością budowy blue boxa, że razem ze Steve'em Jobsem stworzyli kilka takich urządzeń i sprzedawali je kolegom z uczelni. Jobs później przyznał, że gdyby nie przygoda z blue boxami, Apple mogłoby nigdy nie powstać.

Draper był postacią kontrowersyjną – z jednej strony traktowany jako geniusz i pionier, z drugiej jako przestępca i buntownik przeciw systemowi. Był kilkukrotnie aresztowany, spędził czas w więzieniu, ale nigdy nie przestał być aktywny w świecie technologii. W latach 80. pracował nad oprogramowaniem, m.in. dla Apple, a później zajmował się bezpieczeństwem systemów i telefonii VoIP.

Co ciekawe, John Draper przez całe życie miał problemy ze słuchem, a mimo to jego kariera kręciła się wokół… dźwięków i telefonii. Był także radiowcem, konstruktorem urządzeń elektronicznych, a później – aktywnym uczestnikiem konferencji hakerskich takich jak DEF CON czy HOPE, gdzie do dziś jest legendą.

Draper nie był inżynierem w tradycyjnym sensie. Był typem majsterkowicza, samouka i eksperymentatora. Pracował poza schematem, często balansując na granicy prawa. Ale to właśnie

tacy ludzie jak on – podobnie jak Clifford Stoll czy John Carmack – pokazali, jak ogromny wpływ na technologię mogą mieć jednostki z nietypowymi zainteresowaniami i odwagą do łamania reguł.

Źródła:

- https://en.wikipedia.org/wiki/John_Draper
- https://en.wikipedia.org/wiki/Phreaking

Kevin Mitnick


Źródło: Wikimedia.org

Jest on osobą wyjątkową z punktu widzenia informatyki, ponieważ przez lata był uważany za najbardziej poszukiwanego hakera na świecie, a jego historia brzmi jak scenariusz filmu sensacyjnego. Czytelnik może teraz zapytać: co takiego zrobił, że stał się ikoną cyberprzestępczości? Wszystko zaczęło się całkiem niewinnie – od autobusów w Los Angeles.

Młody Kevin Mitnick nauczył się w wieku kilkunastu lat, jak działa system biletowy lokalnej komunikacji miejskiej. Korzystając z kartonów po biletach, maszyny do ich kasowania i kilku rozmów z kierowcami, stworzył własny system do podróżowania za darmo. Później przesiadł się z autobusów na linie telefoniczne – był zafascynowany phreakingiem i dźwiękami w sieciach telekomunikacyjnych. Szybko odkrył, że potrafi przekonać ludzi do przekazania mu informacji, których normalnie nie powinni udostępniać. Tak zaczęła się jego droga do tzw. inżynierii społecznej (social engineering), z której zasłynął na całym świecie.

Kevin w latach 80. i 90. włamywał się do systemów takich firm jak Nokia, Motorola, Sun Microsystems, Fujitsu, a nawet Pacific Bell. Nie używał do tego zaawansowanych exploitów – jego główną bronią był telefon i… uprzejma rozmowa. Potrafił udawać administratora IT, inżyniera serwisu czy pracownika firmy telekomunikacyjnej, a następnie wyciągał hasła, numery identyfikacyjne i dostępy od niczego niepodejrzewających pracowników. Później wykorzystywał te informacje, by dostać się do systemów firm, pobierać ich kod źródłowy i badać, jak działają ich technologie.

W 1995 roku, po kilkuletnim pościgu, FBI w końcu go złapało. Mitnick ukrywał się pod fałszywym nazwiskiem, zmieniał miejsca zamieszkania i praktycznie nie zostawiał śladów. Co ciekawe, zatrzymano go na podstawie śledztwa prowadzonego przez innego hakera – Tsutomu Shimomurę, który osobiście postanowił odnaleźć Mitnicka po tym, jak ten włamał się na jego komputer. Po zatrzymaniu Mitnick został oskarżony o wiele przestępstw komputerowych i spędził ponad 5 lat w więzieniu, z czego 8 miesięcy w izolatce, bo – jak głosi legenda – władze obawiały się, że mógłby „uruchomić broń nuklearną, jeśli zadzwoni z telefonu-ślimaka”.

Po wyjściu z więzienia Kevin przeszedł niezwykłą transformację – został specjalistą ds. bezpieczeństwa, konsultantem, mówcą publicznym i autorem bestsellerów. Założył własną firmę – Mitnick Security, która doradzała największym korporacjom i instytucjom rządowym, jak chronić się przed atakami. Napisał również kilka książek, m.in. „Sztuka Podstępu” i „Sztuka Włamania”, w których opisywał techniki socjotechniczne i uczył, jak się przed nimi bronić.

Mitnick był jednym z pierwszych, którzy pokazali światu, że największym zagrożeniem dla bezpieczeństwa komputerowego nie jest kod, tylko… człowiek. Dzięki jego historii dziś cyberbezpieczeństwo to nie tylko firewalle i antywirusy, ale też szkolenia dla pracowników i lepsza świadomość zagrożeń.

Kevin Mitnick zmarł w 2023 roku po walce z nowotworem, ale jego legenda wciąż żyje. Został zapamiętany nie tylko jako przestępca, ale też jako nauczyciel i pionier myślenia o bezpieczeństwie informacji.

Źródła:

- https://en.wikipedia.org/wiki/Kevin_Mitnick
- https://www.mitnicksecurity.com/
- https://en.wikipedia.org/wiki/Social_engineering_(security)

Świat Systemów Operacyjnych i komputerów osobistych

Ken Thompson i Denis Ritchie


Źródło: Wikimedia.org

Jest to duet wyjątkowy z punktu widzenia informatyki, bo bez ich pracy trudno sobie dziś wyobrazić świat komputerów, systemów operacyjnych czy Internetu. Choć nie nosili czarnych golfów, nie wychodzili na scenę z błyskami reflektorów i nie krzyczeli „one more thing”, to ich wpływ na technologię był… większy niż kogokolwiek z wielkich firm Doliny Krzemowej. A mimo to, gdy Dennis Ritchie zmarł w październiku 2011 roku — w tym samym miesiącu co Steve Jobs — świat niemal tego nie zauważył. Media milczały. Twitter nie zalały wspomnienia. Być może dlatego, że geniusz Ritchiego i Thompsona był bardziej cichy, skromny i… wszechobecny.

Ken Thompson i Dennis Ritchie pracowali razem w laboratoriach Bell Labs, gdzie w latach 60. i 70. XX wieku rozwijano najbardziej zaawansowane technologie komputerowe swoich czasów. Ich wspólna historia zaczyna się od systemu Multics, który miał być nowoczesnym systemem operacyjnym, ale z powodu nadmiaru ambicji projekt utknął. Gdy projekt zamknięto, Thompson postanowił napisać własny system od zera – coś prostszego, eleganckiego i elastycznego. Tak narodził się UNIX.

UNIX, napisany początkowo przez Kena Thompsona, był później przepisany na nowy język programowania — C, który stworzył Dennis Ritchie. Dzięki temu UNIX nie był już zależny od jednej maszyny — można go było przenieść na różne architektury sprzętowe, co było absolutnym przełomem. I tak UNIX zaczął się rozprzestrzeniać: do uczelni, firm, wojska, aż w końcu stał się fundamentem dzisiejszych systemów operacyjnych.

To właśnie na bazie UNIX-a powstały później takie systemy jak Linux, macOS, Android, a nawet systemy routerów, serwerów internetowych i urządzeń wbudowanych. Sam język C, zaprojektowany przez Ritchiego, do dziś pozostaje jednym z najważniejszych i najbardziej wpływowych języków programowania — fundamentem systemów operacyjnych, sterowników, baz danych i gier.

Mimo tego wszystkiego, Dennis Ritchie pozostał w cieniu. Nie był celebrytą. Nie zakładał firm. Nie sprzedawał komputerów. Tworzył narzędzia, które inni potem używali, często nie wiedząc, kto je zbudował. Gdy zmarł w październiku 2011, niedługo po śmierci Steve’a Jobsa, świat technologii pochylił się nad ikoną Apple, a o Ritchiem wspomniano tylko w kilku niszowych artykułach. To było jak utrata fundamentu domu, której nikt nie zauważył, dopóki dom się nie zachwiał.

Ken Thompson z kolei, choć dziś mniej aktywny, również odegrał kolosalną rolę – był twórcą nie tylko UNIX-a, ale i pierwszej wersji edytora grep, języka B (prekursora C), a także współtwórcą systemu Plan 9 i języka Go (we współpracy z Google). Thompson był zawsze zafascynowany minimalizmem i

prostotą kodu — jego filozofia „małe jest piękne” do dziś jest obecna w projektowaniu systemów.

Obaj – Thompson i Ritchie – otrzymali wiele nagród, w tym Nagrodę Turinga (najwyższe wyróżnienie w informatyce), National Medal of Technology, a także uznanie środowisk akademickich i inżynieryjnych na całym świecie. Ale ich największą nagrodą jest to, że niemal każdy smartfon, komputer czy serwer działający dziś na Ziemi nosi w sobie ślady ich kodu, ich myśli, ich sposobu rozwiązywania problemów.

Ich historia pokazuje, że czasem najwięksi bohaterowie technologii to ci, których nie widać na okładkach, ale których praca jest obecna wszędzie.

Źródła:

- https://en.wikipedia.org/wiki/Dennis_Ritchie
- https://en.wikipedia.org/wiki/Ken_Thompson
- https://en.wikipedia.org/wiki/Unix
- https://www.bell-labs.com

Susan Kare


Źródło: Wikimedia.org

Jest ona osobą wyjątkową z punktu widzenia informatyki, ponieważ nie była programistką, inżynierem ani projektantką systemów operacyjnych, a mimo to na zawsze zmieniła sposób, w jaki ludzie wchodzą w interakcję z komputerami. Czytelnik może zapytać: to co w takim razie ona tutaj robi? Już wyjaśniam. Susan Kare była artystką, która narysowała duszę komputerów Apple.

Susan z wykształcenia była historykiem sztuki i grafiką. W latach 80. dołączyła do zespołu pracującego nad projektem Apple Macintosh, czyli komputerem, który miał spopularyzować interfejs graficzny (GUI) i myszkę. W tamtych czasach komputery to były głównie czarne terminale z zielonym tekstem – bez ikon, bez okien, bez emocji. Kare miała sprawić, że ten nowy komputer będzie ludzki.

Pracując z ograniczeniem do siatki 32×32 piksele i czerni z bielą, Susan stworzyła pierwszy zestaw ikon dla Macintosha: nożyczki, foldery, kosz na śmieci, zegar, uśmiechniętą twarz „Happy Mac” witającą użytkownika po uruchomieniu komputera. Te pikselowe rysunki, mimo prostoty, były intuicyjne, estetyczne i pełne charakteru. Nie potrzebowały podpisów – każdy wiedział, co znaczą.

Ale Kare nie stworzyła tylko ikon. Zaprojektowała również pierwsze czcionki ekranowe, takie jak Chicago, Geneva, Monaco, czy New York, które zostały użyte zarówno w interfejsie graficznym, jak i w dokumentach. Jej podejście łączyło funkcjonalność z elegancją – litery miały być czytelne nawet na ekranie o bardzo niskiej rozdzielczości.

Ciekawostką jest, że swoje pierwsze projekty ikon rysowała… na papierze milimetrowym, zanim jeszcze dostała dostęp do komputera, bo zespół inżynierów był zbyt zajęty programowaniem systemu. Kare przekładała swoją wiedzę o kompozycji, proporcjach i symbolice graficznej na język komputerów – i robiła to z niesamowitym wyczuciem.

Po odejściu z Apple, Susan pracowała m.in. dla NeXT (firmy założonej przez Steve’a Jobsa po jego

pierwszym odejściu z Apple), a potem dla Microsoftu, gdzie stworzyła grafiki i czcionki do Windows 3.0. Pracowała również dla Facebooka, IBM, PayPal i wielu innych firm technologicznych.

Mimo że jej praca była często „niewidoczna” w sensie medialnym, to wszyscy ją znają — bo każdy, kto korzystał z komputera w latach 80., 90., a nawet dziś, widział jej ikony, kliknął jej przyciski, czytał litery jej czcionek. Można powiedzieć, że Susan Kare dała komputerom twarz – dosłownie.

W 2015 roku jej prace zostały włączone do stałej kolekcji w Museum of Modern Art w Nowym Jorku. Z małych pikseli zrobiła sztukę, a z komputerów – przyjazne narzędzia codziennego użytku.

Źródła:

- https://en.wikipedia.org/wiki/Susan_Kare

John Carmack


Źródło: Wikimedia.org

Jest on osobą wyjątkową z punktu widzenia informatyki, bo choć nie stworzył żadnego systemu operacyjnego czy protokołu sieciowego, to bez jego pracy dzisiejszy świat gier komputerowych wyglądałby zupełnie inaczej. Czytelnik może teraz zapytać: to co takiego zrobił? Już tłumaczę. Zainteresowania Carmacka od najmłodszych lat krążyły wokół elektroniki, komputerów i… włamań. Tak, jako nastolatek próbował dostać się do szkolnych systemów komputerowych i przez to trafił nawet do poprawczaka. Jednak to nie był koniec jego historii – był to dopiero początek.

W dorosłym życiu Carmack wykorzystał swoje zdolności programistyczne do rzeczy absolutnie przełomowych – był współzałożycielem legendarnego studia id Software, w którym razem z Johnem Romero stworzyli takie tytuły jak Wolfenstein 3D, DOOM, Quake i wiele innych. To właśnie Carmack odpowiadał za silniki graficzne tych gier – czyli to, co sprawia, że gra „działa”, wyświetla świat 3D, światła, cienie i pozwala graczowi się w nim poruszać.

W czasach, gdy komputery osobiste miały moc zbliżoną do dzisiejszych kalkulatorów, Carmack dokonywał cudów. Dla przykładu: Wolfenstein 3D powstał, zanim jeszcze popularne były karty graficzne 3D – cały silnik gry to sprytna symulacja trójwymiarowości za pomocą dwuwymiarowej matematyki. W DOOM poszedł jeszcze dalej, dodając tekstury, efekty świetlne i złożone mapy, które dały początek nowemu gatunkowi – tzw. FPS (First Person Shooter).

To właśnie DOOM był pierwszą grą, która była tak „modowalna”, że gracze mogli tworzyć własne poziomy i modyfikacje, co stworzyło ogromną społeczność twórców niezależnych. Mało kto wie, że Carmack udostępnił później kod źródłowy silnika DOOMa na wolnej licencji, co było nie lada wydarzeniem – był to gest, który zainspirował rzesze młodych programistów.

W 1996 roku Carmack stworzył Quake – pierwszą w pełni trójwymiarową grę akcji, z której silnika później korzystały takie produkcje jak Half-Life czy Call of Duty. Wiele firm kupowało licencje na silniki Carmacka, dzięki czemu powstał cały ekosystem gier zbudowanych na jego fundamentach.

Co ciekawe, John Carmack nie ograniczał się tylko do gier. Od 2000 roku interesował się tematyką lotów kosmicznych i założył firmę Armadillo Aerospace, która konstruowała rakiety i brała udział w konkursach NASA. Z biegiem lat Carmack zaczął także interesować się rzeczywistością wirtualną –

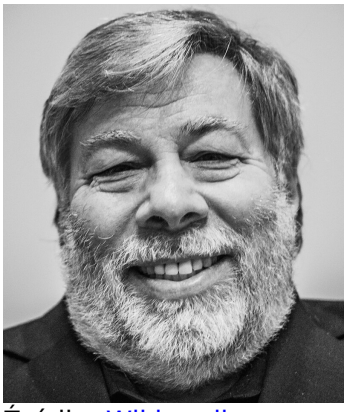
został CTO (Chief Technology Officer) w firmie Oculus VR, gdzie odpowiadał za rozwój gogli Oculus Rift. Można więc powiedzieć, że Carmack nie tylko stworzył nowoczesne gry komputerowe, ale także brał udział w tworzeniu nowoczesnej rzeczywistości wirtualnej.

W 2019 roku Carmack ogłosił, że odchodzi z pracy na pełny etat w Oculusie, by zająć się rozwojem ogólnej sztucznej inteligencji. Dziś prowadzi własną firmę Keen Technologies i pracuje nad budową AGI – sztucznej inteligencji o ogólnych zdolnościach poznawczych. Co przyniesie przyszłość? Nie wiadomo, ale znając Carmacka, prawdopodobnie znów przesunie granice tego, co możliwe.

Źródła:

- https://en.wikipedia.org/wiki/John_D._Carmack
- https://en.wikipedia.org/wiki/Id_Software

Steve Wozniak


Źródło: Wikimedia.org

Jest on osobą wyjątkową z punktu widzenia informatyki, ponieważ był nie tylko współzałożycielem Apple, ale również twórcą jednych z najważniejszych komputerów osobistych w historii. Czytelnik może zapytać: czy to nie Steve Jobs był tym geniuszem? Jobs miał wizję — Wozniak miał lutownicę. I to właśnie od tej lutownicy zaczyna się nasza historia.

Steve Wozniak od młodości interesował się elektroniką. Jako nastolatek projektował własne kalkulatory, konstruował gry logiczne i – co ważne – czytał dokumentacje sprzętowe dla zabawy. W czasach gdy dostęp do komputerów był luksusem dla instytucji naukowych i korporacji, Woz potrafił odtworzyć ich działanie z pamięci i uruchomić własne wersje na płytce prototypowej.

Na początku lat 70. poznał Steve’a Jobsa – młodszego kolegę z liceum i entuzjastę technologii. Ich wspólna przygoda zaczęła się od żartu. Zbudowali „blue boxa”, czyli urządzenie generujące sygnały DTMF, które pozwalało na darmowe wykonywanie połączeń telefonicznych. Wozniak zaprojektował obwody, Jobs znalazł klientów – i tak zaczęli sprzedawać nielegalne urządzenia. Ta wspólna „wpadka” była pierwszym pokazem ich uzupełniających się talentów.

Prawdziwa rewolucja zaczęła się w 1976 roku, gdy Wozniak zbudował komputer Apple I – pierwszy komputer osobisty z pełną klawiaturą i możliwością podłączenia do zwykłego telewizora jako monitora. Co istotne, Woz zrobił to samodzielnie: zaprojektował płytę główną, zbudował prototyp i zaprogramował system. Jobs, dostrzegając potencjał, zaproponował sprzedaż komputerów jako gotowych zestawów. Tak powstała firma Apple Computer, a ich pierwszym klientem był sklep Byte Shop, który zamówił 50 sztuk.

Ale to Apple II, zaprojektowany również przez Wozniaka, był prawdziwym przełomem. Komputer miał kolorową grafikę, dźwięk i możliwość rozszerzeń – był marzeniem każdego domowego użytkownika lat 70. i 80. Wozniak po raz kolejny zrobił coś niesamowitego: stworzył cały system komputerowy, który mógł być produkowany masowo, a przy tym był tani, stabilny i potężny.

W przeciwieństwie do Steve’a Jobsa, który później stał się ikoną biznesu i marketingu, Woz pozostał inżynierem z krwi i kości. Nie interesowały go zarządy i konferencje, bardziej bawiły go zloty hobbystów, pokazy robotów i edukacja technologiczna dzieci. W latach 80., po poważnym wypadku lotniczym, powoli wycofywał się z Apple, ale nigdy nie przestał być aktywny technologicznie. Zakładał

własne projekty, uczył informatyki w szkołach i wspierał społeczności makerów.

Wozniak był też znany z tego, że w przeciwieństwie do wielu innych liderów Doliny Krzemowej – nie traktował pieniędzy jako celu. Podczas IPO Apple oddał tysiące akcji swoim kolegom, którzy nie zostali ujęci w systemie opcji pracowniczych, bo uważał, że „to po prostu uczciwe”.

Do dziś Steve Wozniak jest żywą legendą informatyki i symbolem tego, że prawdziwa innowacja często rodzi się z pasji, a nie z chęci zysku. To dzięki niemu powstał jeden z pierwszych komputerów osobistych, który rzeczywiście każdy mógł mieć w domu. A wszystko zaczęło się od lutownicy, taniej pamięci RAM i... ogromnego zamiłowania do elektroniki.

Źródła:

- https://en.wikipedia.org/wiki/Steve_Wozniak
- https://www.woz.org
- https://pl.wikipedia.org/wiki/Apple_I
- https://pl.wikipedia.org/wiki/Apple_II

2025/05/11 12:21 · administrator · 0 Komentarz

Wyszkolenie własnego modelu AI

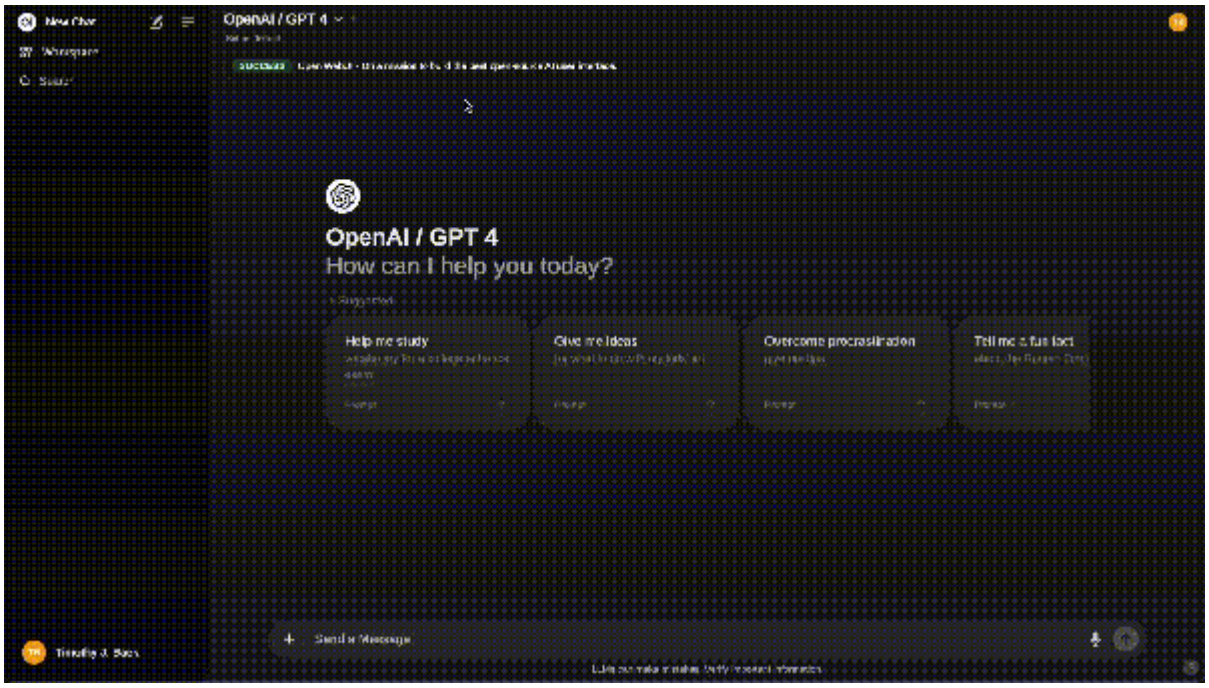
źródło: https://commons.wikimedia.org/wiki/File:Artificial-Intelligence.jpg

Wikipedia

> A large language model (LLM) is a type of machine learning model designed for natural language processing tasks such as language generation. LLMs are language models with many parameters, and are trained with self-supervised learning on a vast amount of text.

Od pewnego czasu chodzi za mną pomysł żeby wyszkolić jakichś otwarty model LLM np. Ollama, wyszkolić go wszystkimi materiałami z mojego dysku lub z tej wiki i potem opublikować go na mojej stronie jako chat bot którego można sie o wszystko zapytać.

Ollama + Openwebui



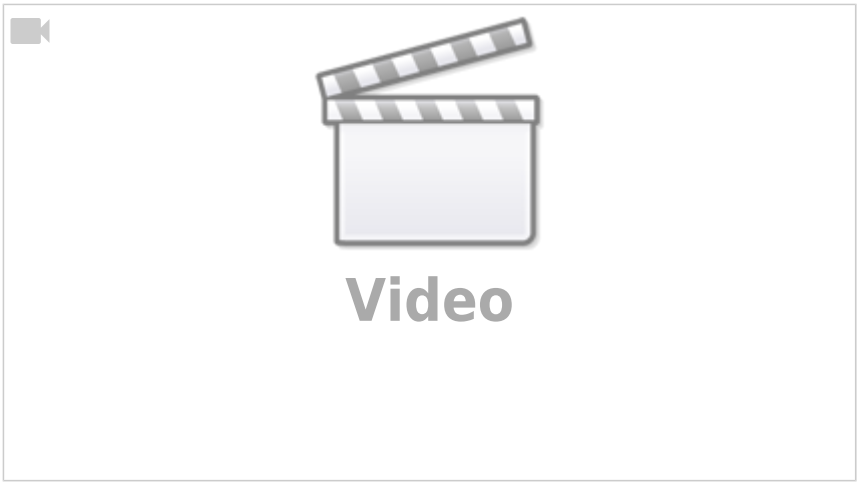
źródło: https://docs.openwebui.com/assets/images/demo-d3952c8561c4808c1d447fc061c71174.gif

Testowałem ten zestaw narzedzi i niestety ale wszystkie potrzebne pliki którymi chcielibyśmy uczyć AI trzeba wysłać przez panel webowy do modelu który trenujemy co jest strasznie mozolne. Potem model musi to wszystko przeczytać co powoduje że trwa to jeszcze dłużej. Nie jest to najlepsze rozwiązanie do takiego zastosowania jak wymieniłem we wstępie. Nie ma tutaj też możliwości podłączenia jakiegoś katalogu z plikami tak żeby AI sobie wszystko zaindeksowała a następnie odpowiadała na pytania zgodnie z tą wiedzą.

Zaleta tego rozwiązania jest jedna jest to program webowy można go otworzyć wszedzie oraz ma możliwość podłączenia różnych dostawców usług AI w jedno miejsce, co pozwala na porównywanie wyników różnych modeli AI.

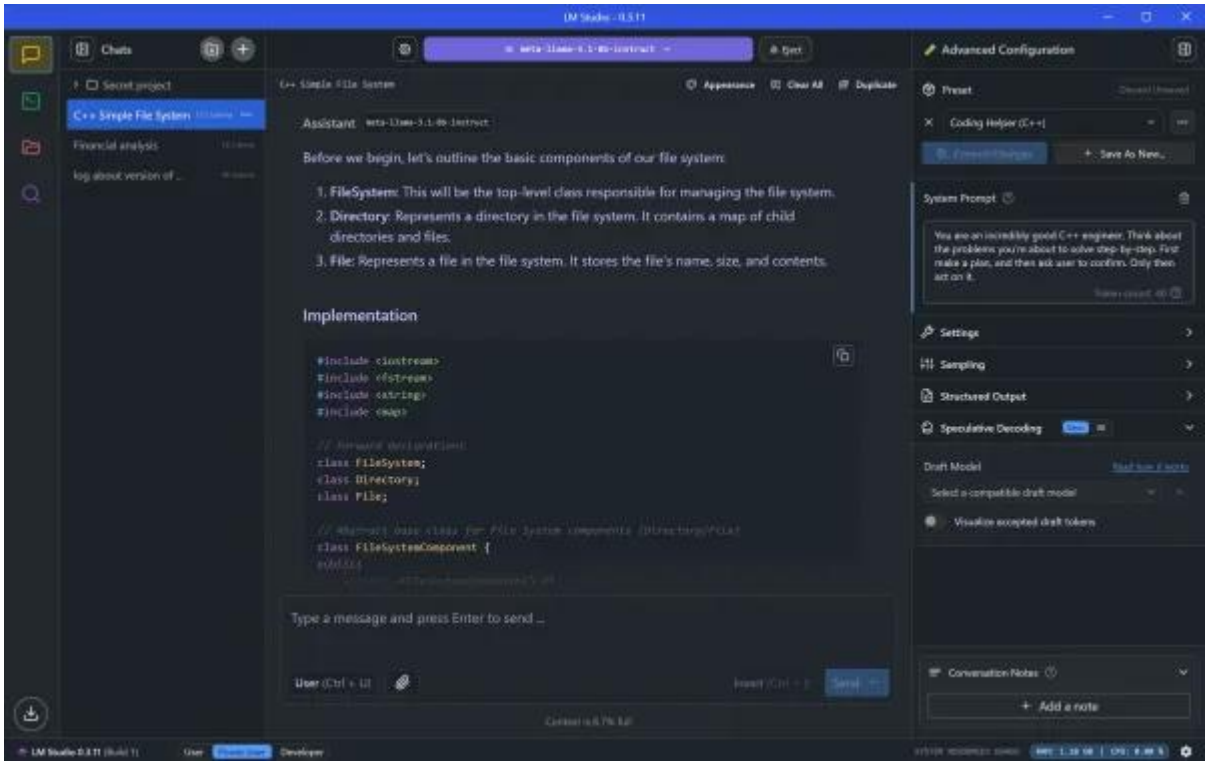
Poza tym ma fajne ustawienia uprawnień do modeli oraz promptów.

Sam okazjonalnie korzystam z tego narzędzia ponieważ bardziej opłaca się płacić za poszczególne żądania do API OpenAI niż za całą subskypcje ChatGPT PLUS albo PRO.



Fajny materiał omawiający OpenWebUi na YT

LMstudio

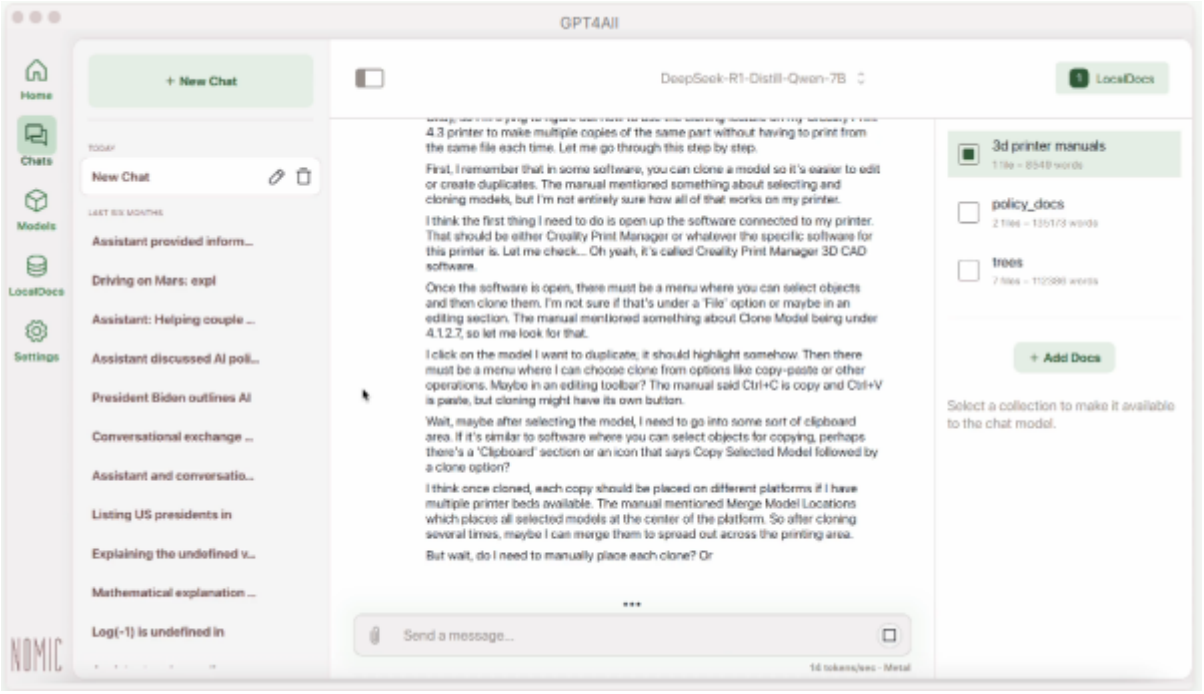


źródło:https://lmstudio.ai/_next/image?url=%2F_next%2Fstatic%2Fmedia%2Fhero-windows.2a9fa20d.webp&w=3840&q=75

To jest narzędzie które zainstalowałem na mojej głównej stacji roboczej. Jest to fajny program jeżeli chcecie na komputerze poczatować z AI na jakieś tematy poufne. Tutaj znowu jest ten sam problem jak ten z OpenWebUi, narzędzie to zachowuje się jak aplikacja Webowa przerobiona na aplikację desktop za pomocą pakietu Elektron. Ma to swoje wady, wrzucanie plików działa tak samo jak w OpenWebUi, powoduje to ten sam problem czyli nie możemy wysłać 100 plików bo cała aplikacja się zawiesi, nie ma również mozliwości podpięcia katalogu tak żeby AI go zindeksowała.

Jest to fajny program jak chcecie w łatwy sposób poeksperymentować z modelami LLM na swoim własnym komputerze.

GPT4ALL



źródło: https://www.nomic.ai/gpt4all

Od razu disclaimer nie testowałem jeszcze tego narzędzia. Natomiast ze strony możemy przeczytać że:

> Chat with Your Files Privately: Introducing LocalDocs Grant your local LLM access to your private, sensitive documents with LocalDocs. Your documents stay secure and private. Your local LLM can access your documents without an internet connection.

Czyli być może będzie to właśnie to o co mi chodzi.

W dokumentacji na stronie możemy przeczytać:

> How It Works. A LocalDocs collection uses Nomic AI's free and fast on-device embedding models to index your folder into text snippets that each get an embedding vector. These vectors allow us to find snippets from your files that are semantically similar to the questions and prompts you enter in your chats. We then include those semantically similar snippets in the prompt to the LLM. To try the embedding models yourself, we recommend using the Nomic Python SDK

Zapowiada się dobrze, z dokumentacji wynika że jest specjalne narzędzie od Nomic AI które indeksuje foldery pliki zostają zamienione na snipetty i każdy znich ma wektor zagnieżdzania, nastepnie te wektory pozwalają AI na znajdowanie snipettów z plików które są semantycznie podobne do pytań i promptów jakie podajemy modelowi. I potem te semantycznie podobne snipetty są dodawane do odpowiedzi.

Z tego co mogliśmy przeczytać jest to coś co pozwala najbliżej się zbliżyć do tego co chciałem osiągnąć.

W jednym z następnych postów będziemy robić testy tego rozwiązania i będziemy patrzeć jak sobie radzi z większą ilością danych oraz czy w ogóle się do czegoś nadaje.

— *Kacper Ostrowski 2025/05/09 11:08*

2025/05/09 10:49 · administrator · 0 Komentarz

Test generatora DDS FNIRSI

oryginalny wpis na blogu: lipca 12, 2023

Nie zawsze chce mi się wyciągać mój duży generator który waży około 20kg, ma on co prawda bardzo dobre parametry ale nie zawsze mi potrzeba 10MHz sinusoidy z współczynnikiem zniekształcenia mniejszym niż 1%. Czasami potrzebuję zwykłej sinusoidy ze stabilną częstotliwością. Siedząc kiedyś na allegro zauważyłem taki prosty generator DDS FNIRSI, malutki, lekki zasilany z zasilacza, pomyślałem czemu nie. Wiem że jeżeli jest jakaś przeszkoda stojąca pomiędzy mną a zrobieniem nowego projektu to nie rozpocznę pracy nad nowym projektem. I w taki sposób stałem się posiadaczem tego malutkiego generatora. W tym materiale postaram się sprawdzić czy taki generator będzie przydatny w warsztacie elektronika.

Wygląd:

Generatorek jest sprzedawany razem w zestawie z obudową z czarnego nie prześwitującego akrylu, moim zdaniem bardziej estetyczny niż te wszystkie przeźroczyste obudowy do innych tego pokroju urządzeń. Obudowa jest do własnoręcznego złożenia, montaż prosty chociaż chwilę mi zajęło dojść gdzie powinienem dać dystanse.

Interfejs:

Do wykorzystania generatora projektanci przeznaczyli 5 przycisków oraz 2 potencjometry. Potencjometry wykorzystujemy do regulacji Amplitudy przebiegu oraz składowej stałej. Przyciski natomiast służą do przełączania rodzaju przebiegu, włączania/wyłączania wyjścia oraz zmiany częstotliwości

Spostrzegawczy czytelnicy pewnie się zorientowal. Ale czemu tutaj są pokrętła do regulacji składowej oraz amplitudy? Już spieszę odpowiedzieć na to pytanie, wynika to z tego że ten generator jest swego rodzaju analogowo cyfrową hybrydą, cyfrowo odbywa się generowanie sygnału oraz regulacja częstotliwości natomiast składowa i amplituda jest regulowana poprzez dwa proste obwody wzmacniaczy operacyjnych na wyjściu.

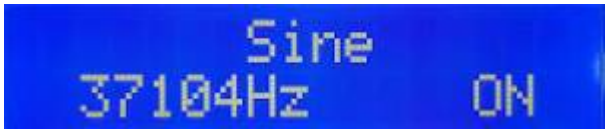
Parametry:

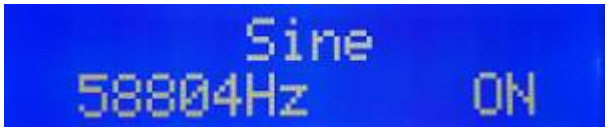
Poniżej parametry deklarowane przez producenta:

1. Kanały: 2
2. Zakres częstotliwości DDS (kanał 1): 1 Hz do 65534 Hz
3. Częstotliwość High Speed (kanał 2): 1 MHz, 2 MHz, 4 MHz i 8 MHz
4. Regulacja offsetów: potencjometr 0.5pp - 5Vpp
5. Regulacja amplitudy: potencjometr 0.5Vpp - 14Vpp
6. Sygnały DDS:
7. Sinusoida
8. Sygnał prostokątny
9. Sygnał trójkątny
10. Fala piłokształtna
11. Odwrotna fala piłokształtna
12. EKG
13. Szum losowy
14. Interfejs: 2 potencjometry i 5 przycisków
15. Wyświetlacz LCD 1602, 2 linie z podświetleniem
16. Regulacja częstotliwości: 1 Hz, 10 Hz, 10 0Hz, 1000 Hz i 10000 Hz
17. Impedancja wyjściowa 20 do 200 Ohm

Wiem wiem pasmo trochę słabe i przebiegi dziwne EKG !?!?!?

No dobrze ale przejdźmy do poważniejszych kwestii jak się korzysta z tego generatora. Powiem szczerze jestem zaskoczony jak dokładna jest częstotliwość wskazywana przez ten generator. Poniżej kilka pomiarów miernikiem częstotliwości. Niebieski wyświetlacz to wyświetlacz generatora czerwony wyświetlacz to wyświetlacz miernika.





Poniżej obliczenia błędu względnego przyjmując wartość podawaną przez miernik częstotliwości za wartość odniesienia.

ΔX - Różnica pomiędzy wielkością mierzoną oraz wielkością odniesienia
X - Wielkość odniesienia
X_0 - Wielkość wskazania
δ - Błąd względny

$$\delta = \frac{\Delta X}{X} \cdot 100\% = \frac{|X - X_0|}{X} \cdot 100\%$$

1 POMIAR

$X = 37105,326$ Hz
$X_0 = 37104$ Hz

$$\Delta X = |37105,326 - 37104| = 1,326$$

$$\frac{\Delta X}{X} = \frac{1,326}{37105,326} = 0,000035736$$

$$\frac{\Delta X}{X} \cdot 100\% = 0,0035736\%$$

$\delta_1 \approx 0,004\%$

2 POMIAR

$X = 58806,144$ Hz
$X_0 = 58804$ Hz

$$\delta = \frac{\Delta X}{X} \cdot 100\% = \frac{2,144}{58806,144} \cdot 100\%$$

$\delta_2 = 0,003645877\%$
$\delta_2 \approx 0,004\%$

3 POMIAR

$X = 706,98534$ Hz
$X_0 = 707$ Hz

$$\delta = \frac{\Delta X}{X} \cdot 100\% = \frac{0,01466}{706,98534} \cdot 100\%$$

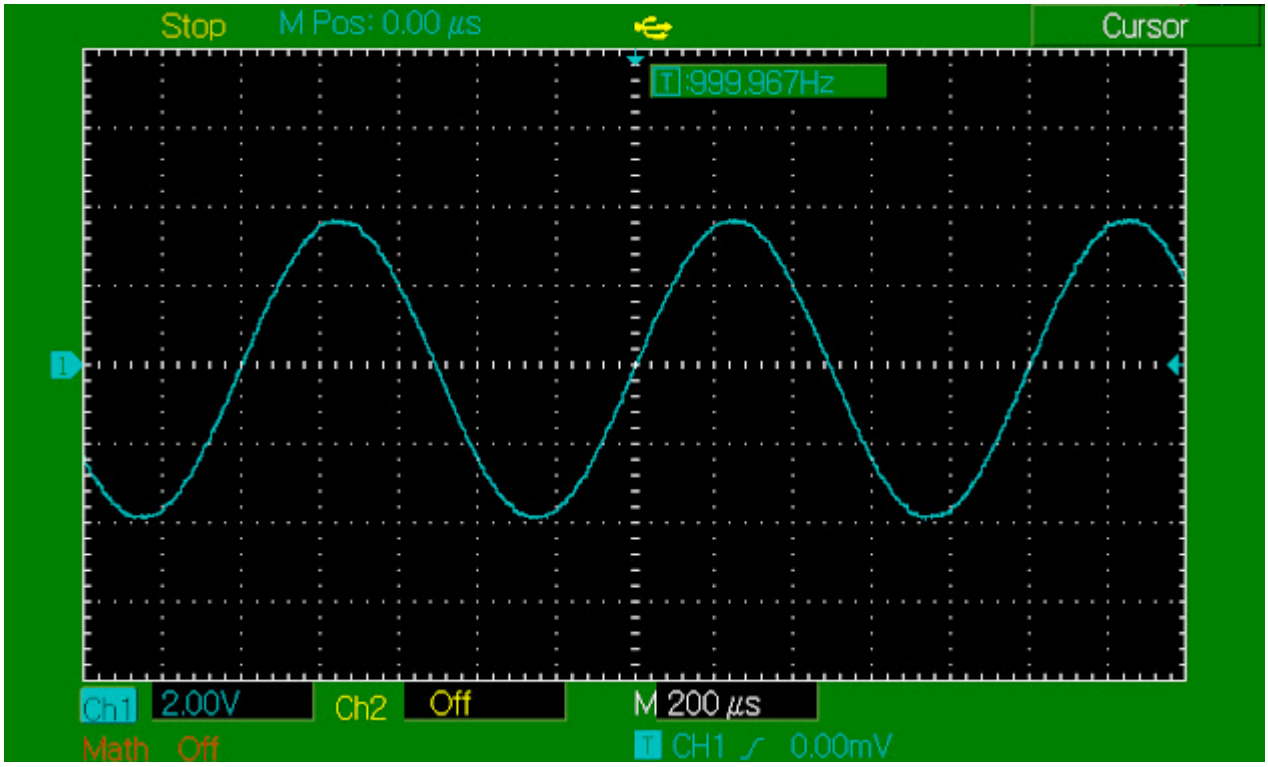
$\delta_3 = 0,020735393\%$
$\delta_3 \approx 0,02\%$

ŚREDNI BŁĄD

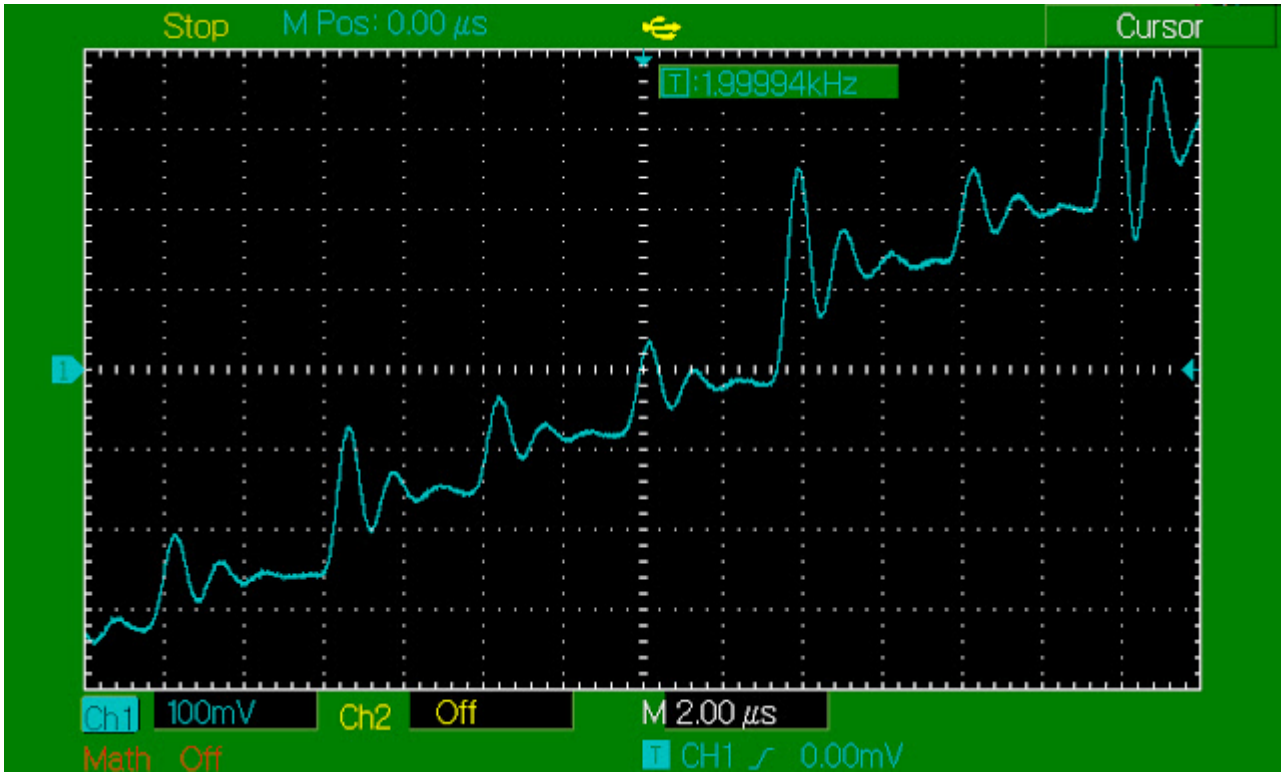
$$\overline{\delta} = \frac{\delta_1 + \delta_2 + \delta_3}{3} \approx 0,0093\%$$

Jak widać średni błąd częstotliwości wyświetlanej przez ten generator to mniej niż 1/100 procenta. WOW ?!?! Dobry wynik jak na urządzenia za około 100zł. Podsumowując wskazaniom częstotliwości tego generatora można ufać.

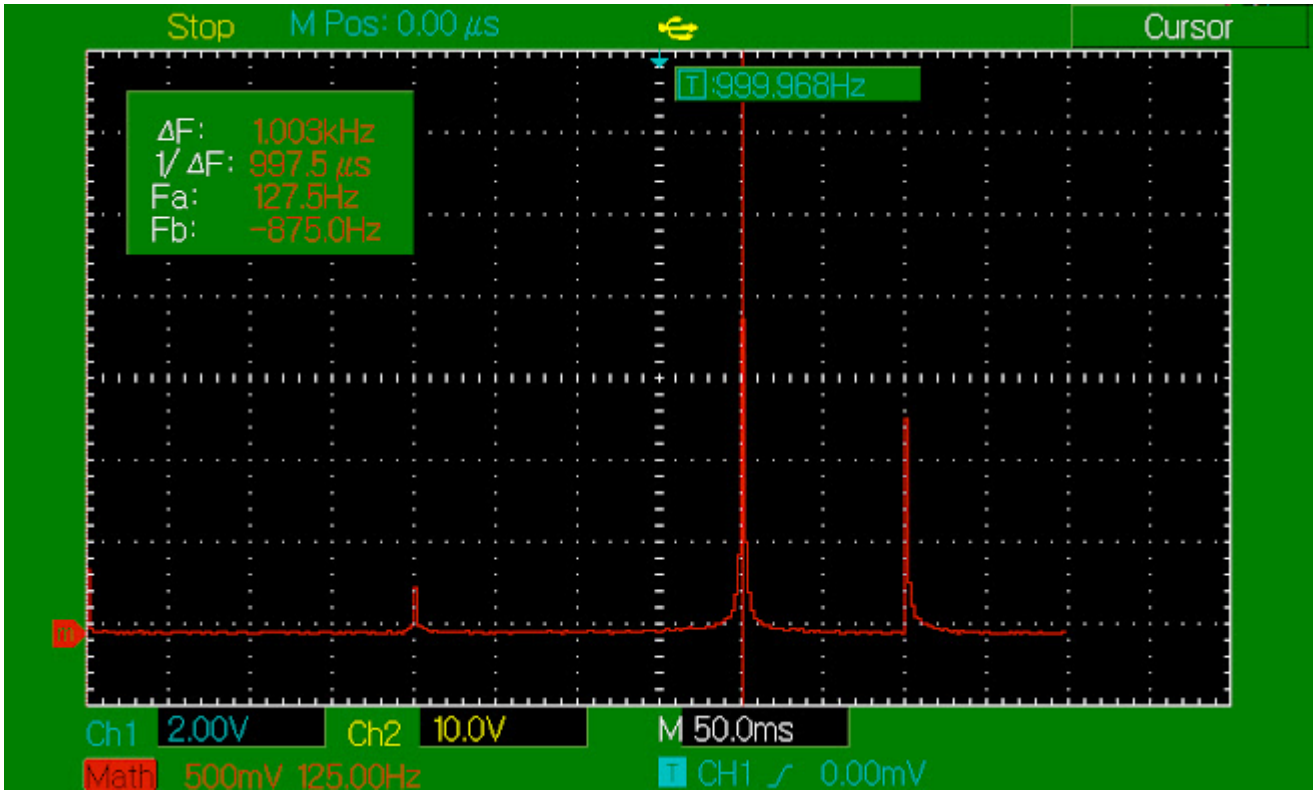
Natomiast co z kształtem przebiegów. Poniżej sinusoida o częstotliwości 1kHz.



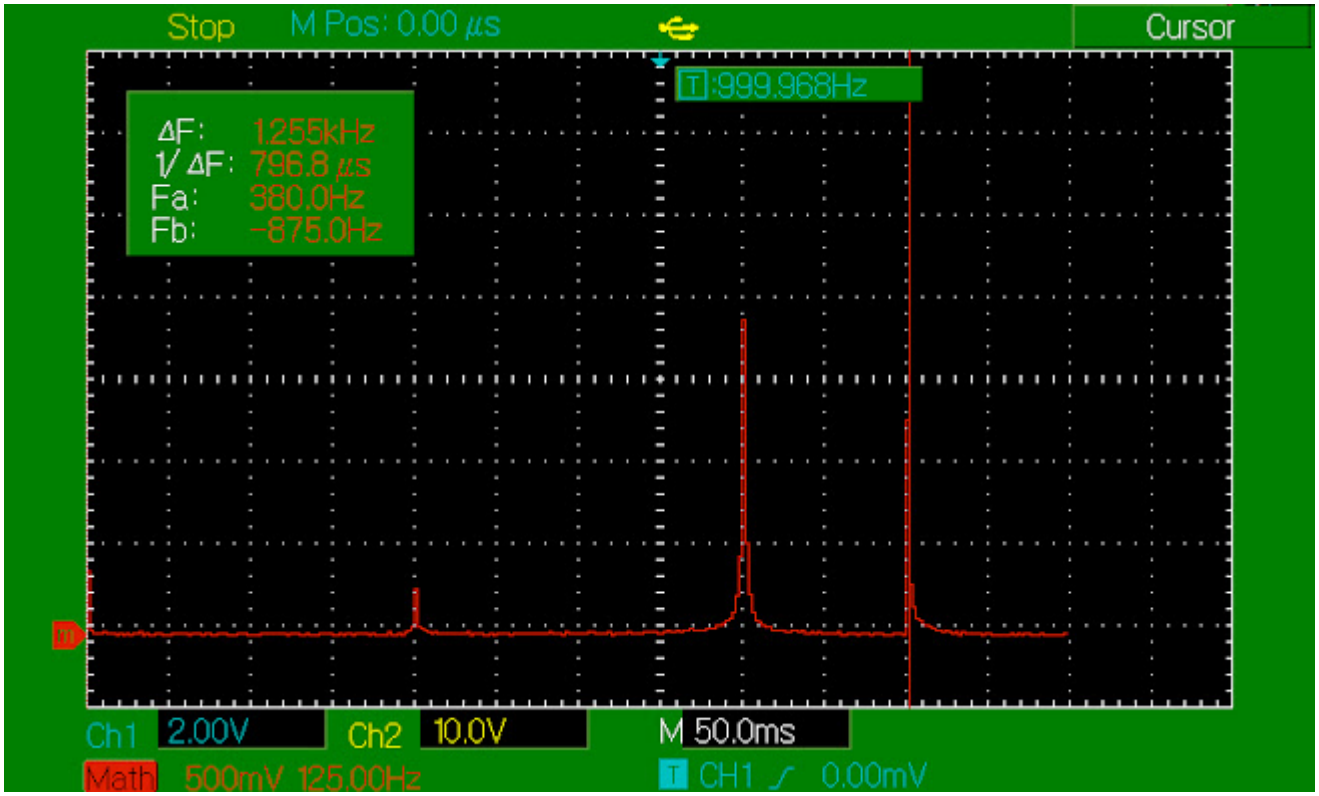
Wygląda ładnie nie widać zniekształceń. Teraz przyjrzyjmy się zboczu rosnącemu tej sinusoidy



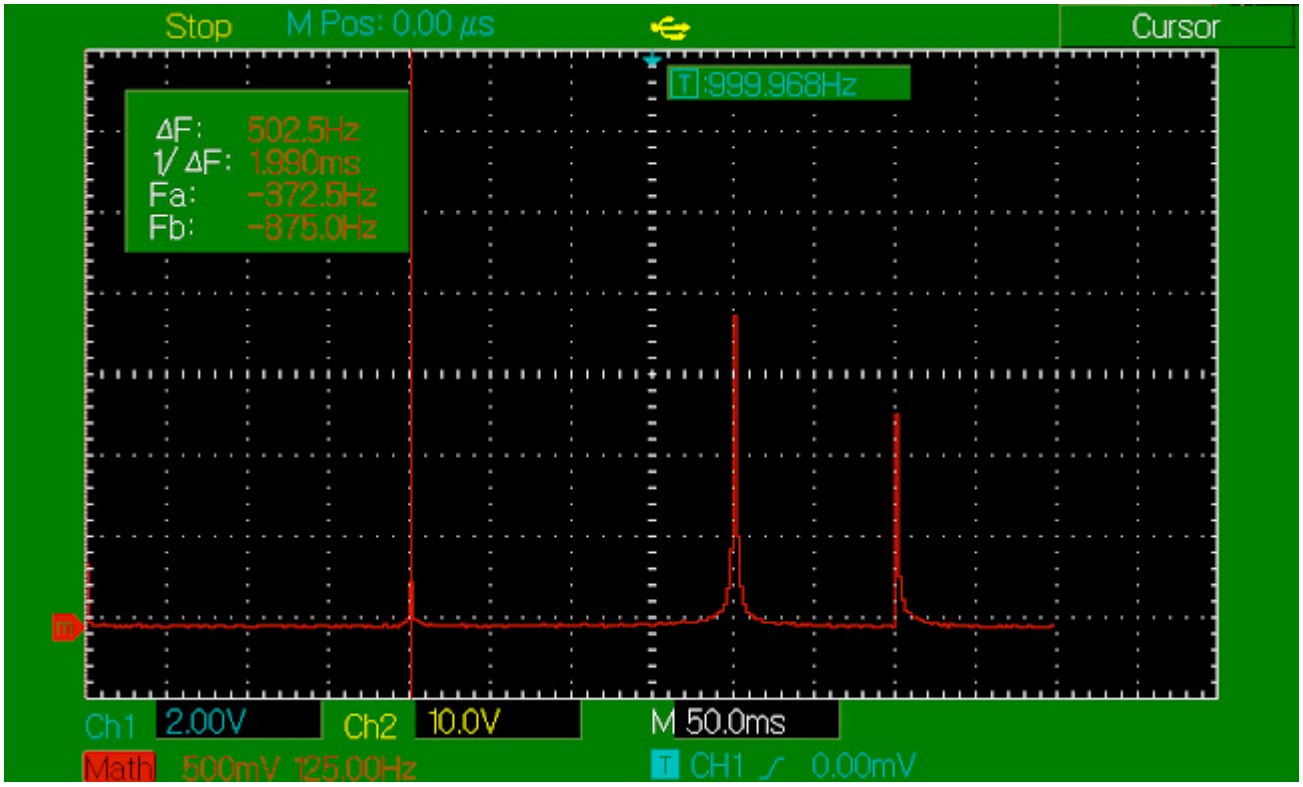
No i zaczyna nam się ujawniać wielka wada generatorów DDS. Mianowicie nie generują one częstotliwości z jakiegoś oscylatora ale wykorzystują jakiś sygnał pośredni np. PWM lub PCM a następnie filtrują go do postaci gładkiej sinusoidy pozostawia to oddźwięk taki jak widać powyżej. Poniżej widać analizę widma powyższej sinusoidy.



Mamy tutaj do czynienia z trzema składowymi częstotliwościami. Największa oczywiście z nich to częstotliwość ustawiona na generatorze.



Druga największa z nich to 1,255kHz

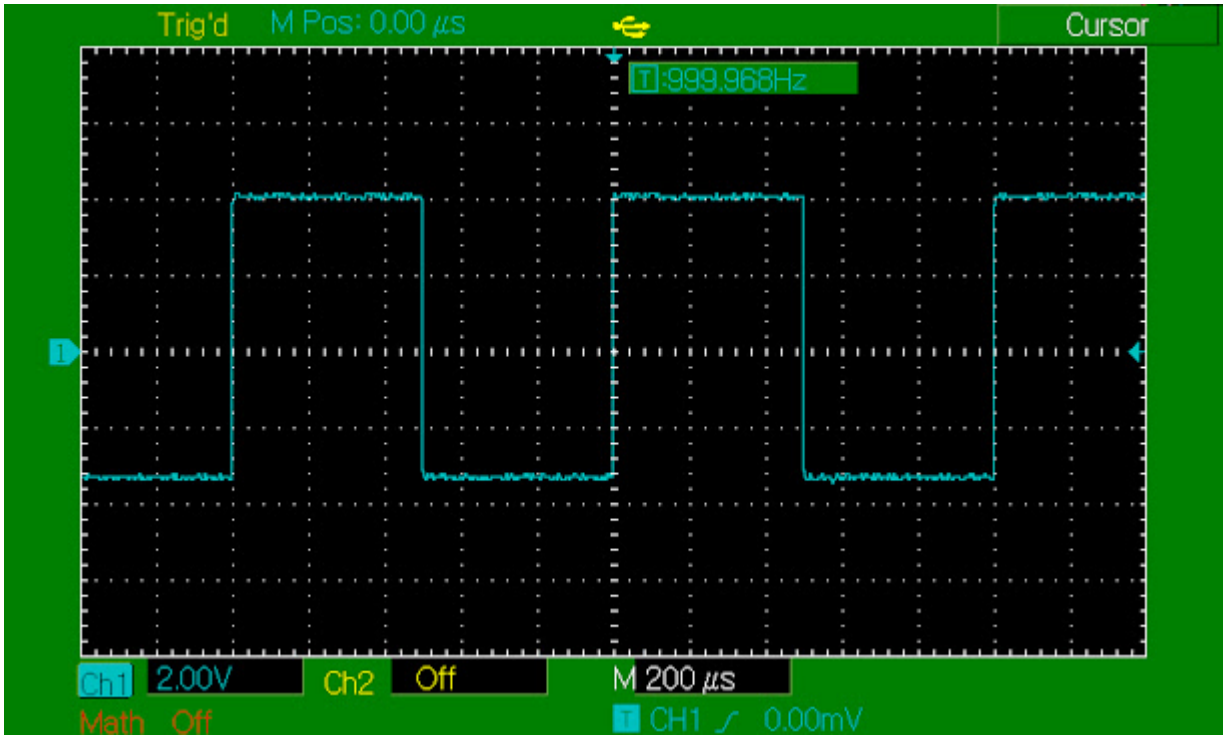


Trzecia najmniejsza z nich to 503Hz.

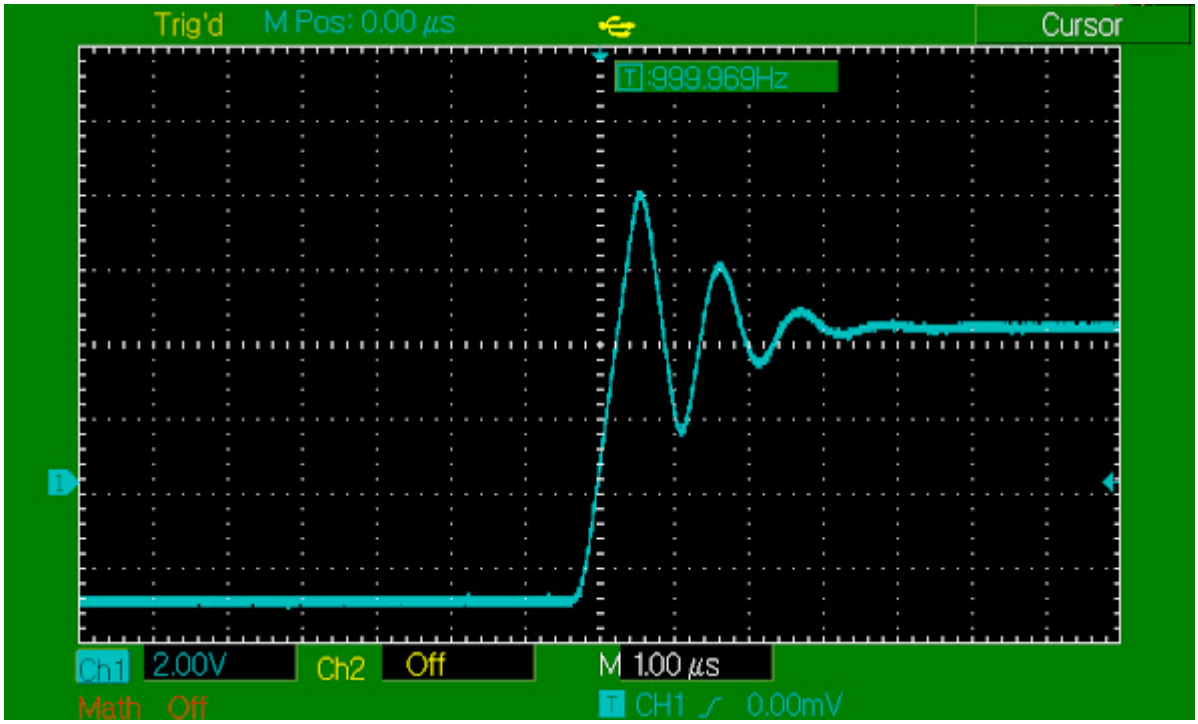
Są to dziwne częstotliwości natomiast najwidoczniej generator wykorzystuje je do stworzenia częstotliwości 1kHz.

Podsumowując częstotliwość stabilna, natomiast nie jest do końca czysta ale tak to już jest z tanią cyfrową technologią. Do prostszych pomiarów wystarcza.

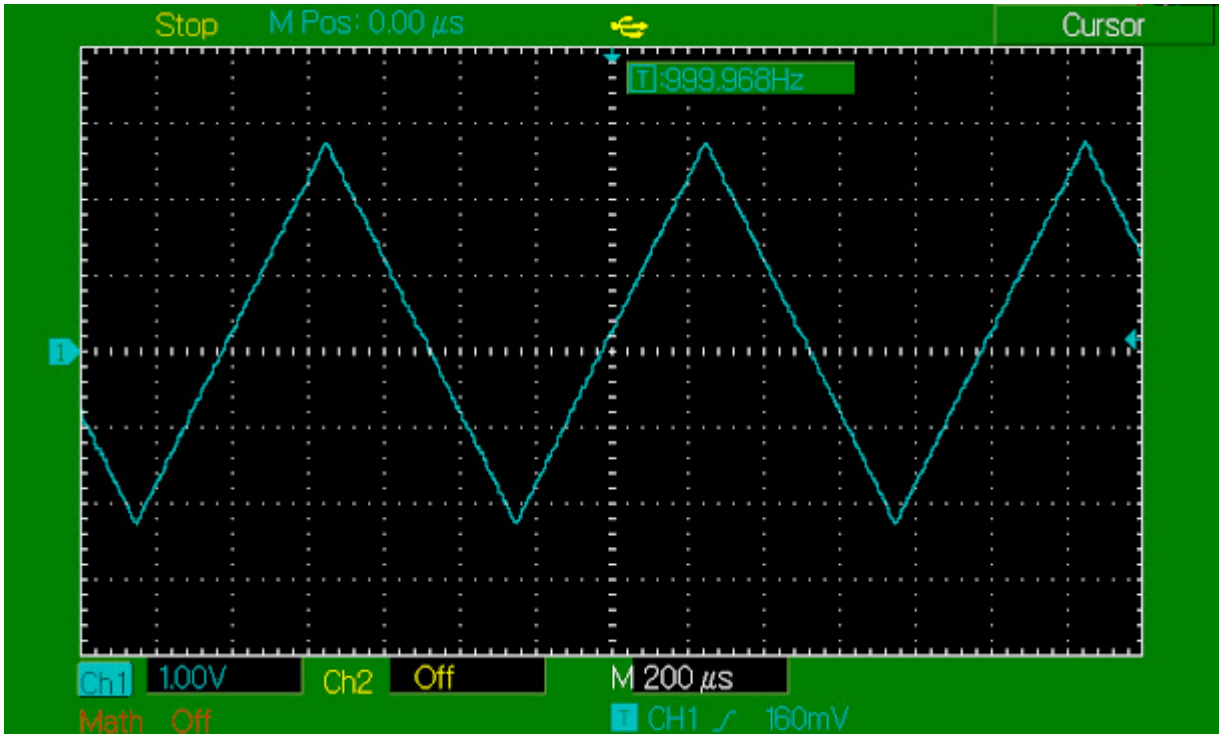
Przetestujmy w takim razie resztę przebiegów.



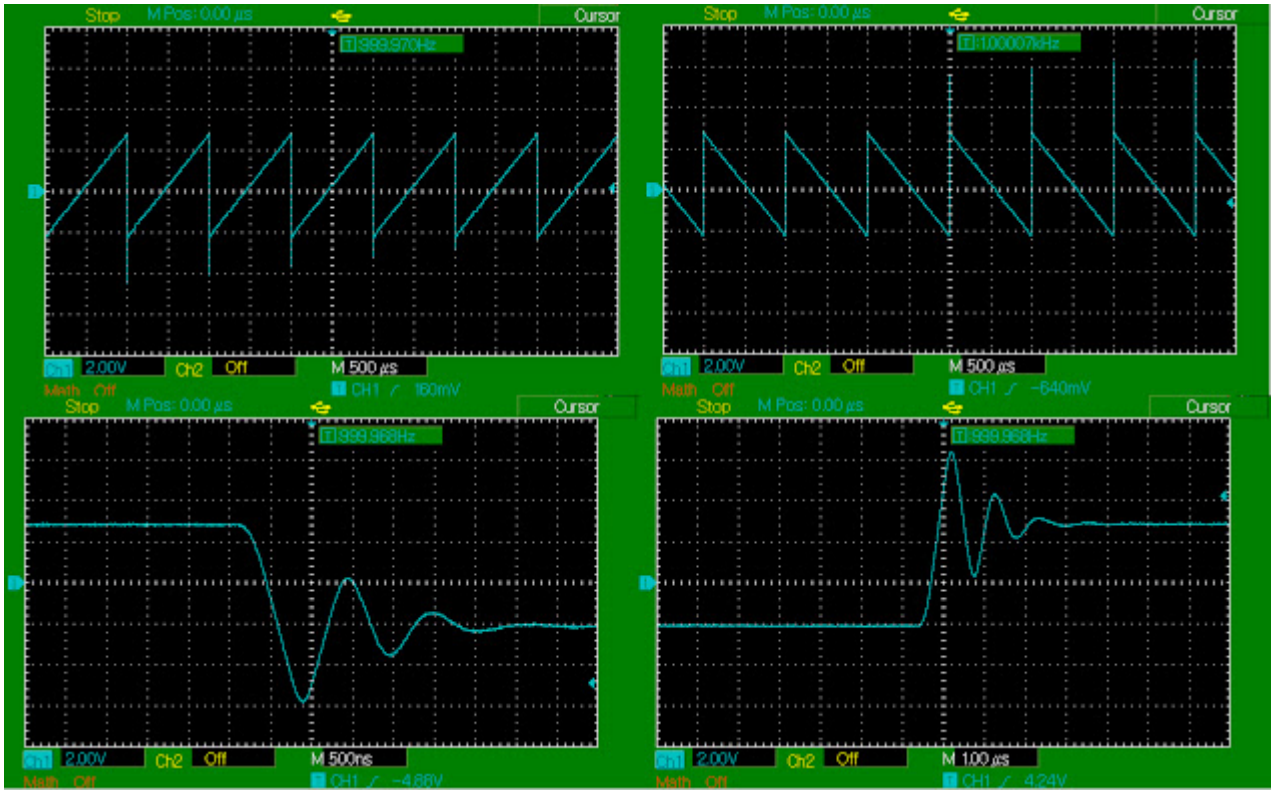
Prostokąt 1kHz wygląda dobrze ale spójrzmy na jego zbocze rosnące



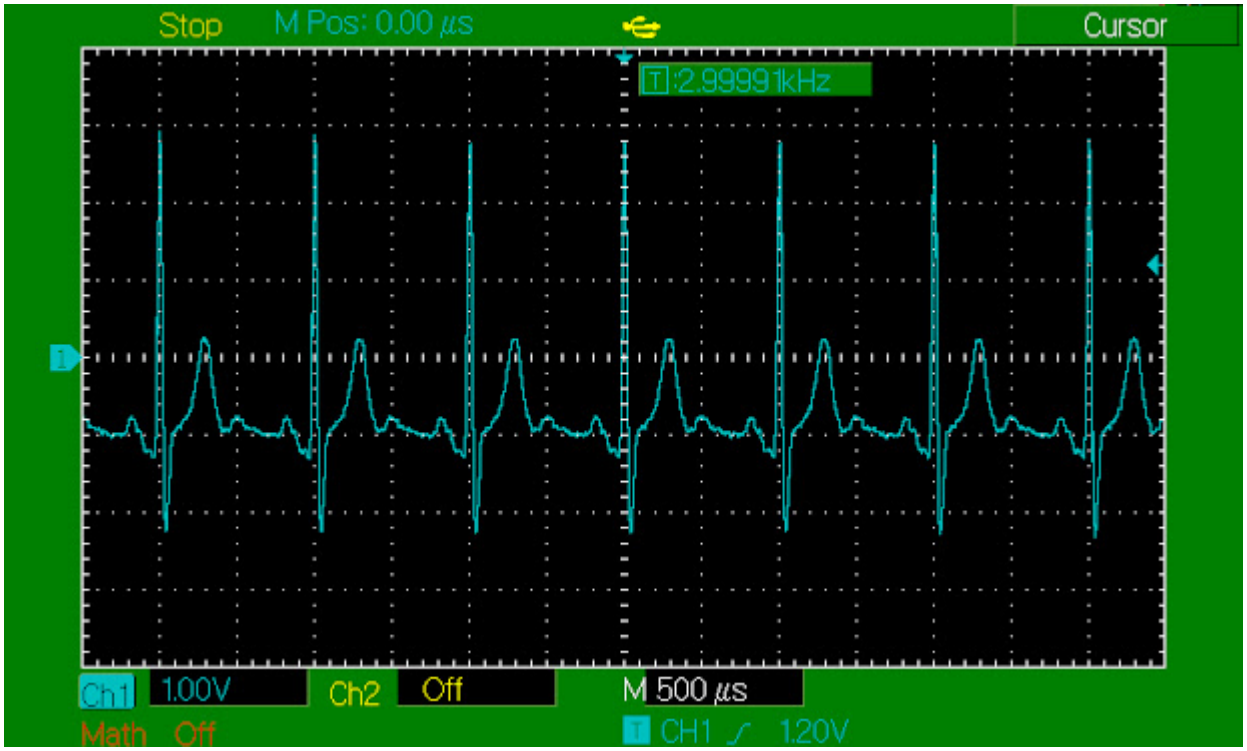
Zbocze rosnące prostokąta nie wygląda najlepiej mamy spore przestrzelenie oraz spore oscylacje gasnące



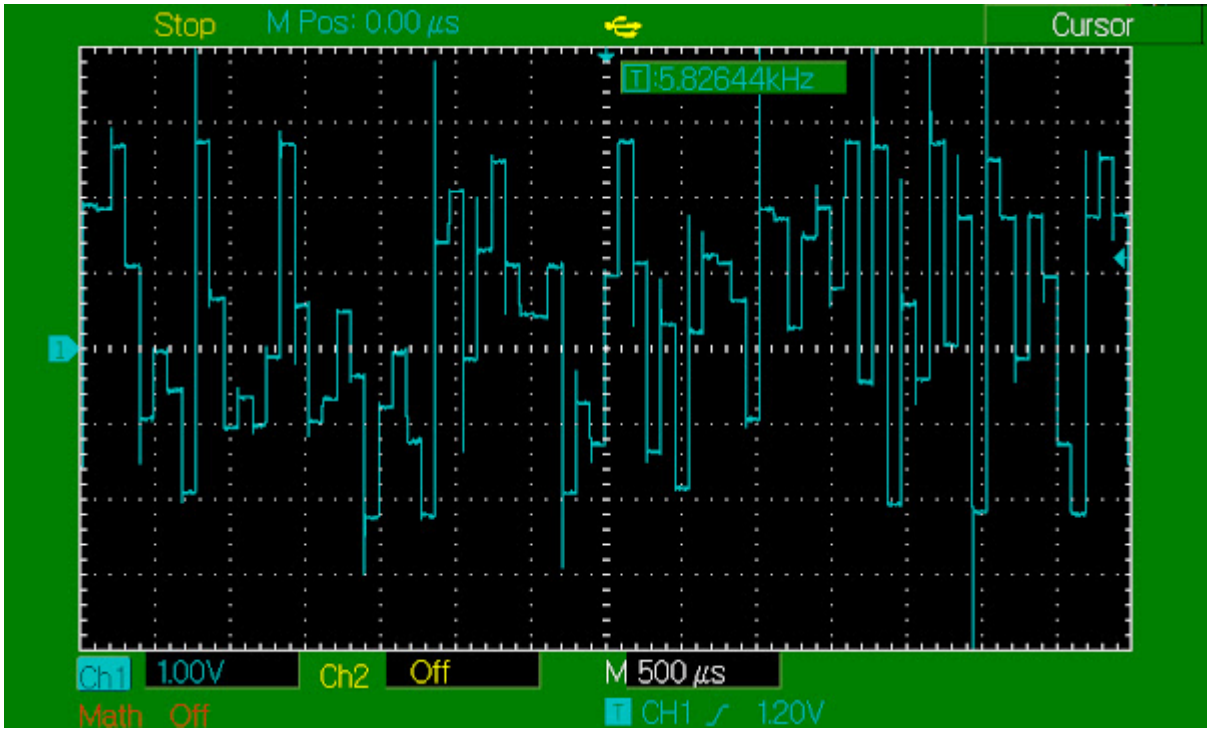
Trójkąt 1kHz (przyglądałem się zboczom i wyglądają tak samo jak przy sinusie)



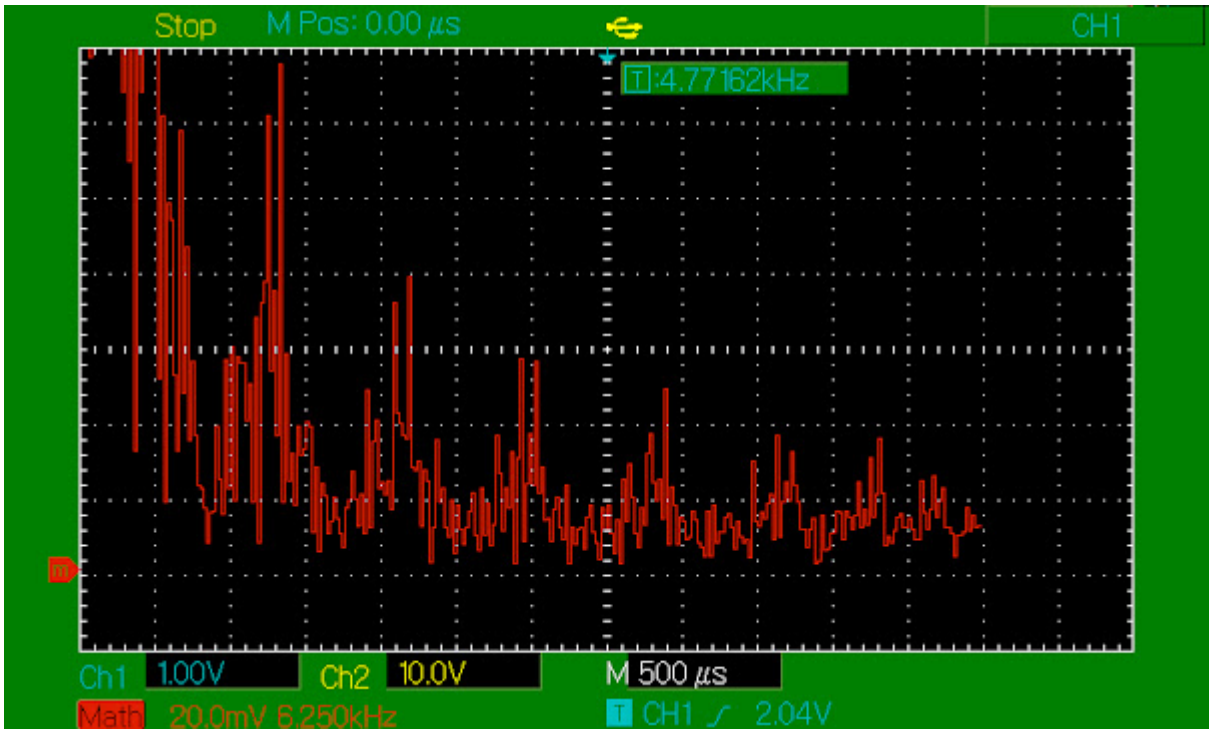
Zestawienie przebiegów piło kształtnych oraz ich zbocz rosnących i opadających



Przebieg ECG nie zajmuję się elektroniką medyczną więc nie wypowiem się na teamat tego przebiegu poza tym że oscyloskop nie wie jak ma mi zmierzyć jego częstotliwość



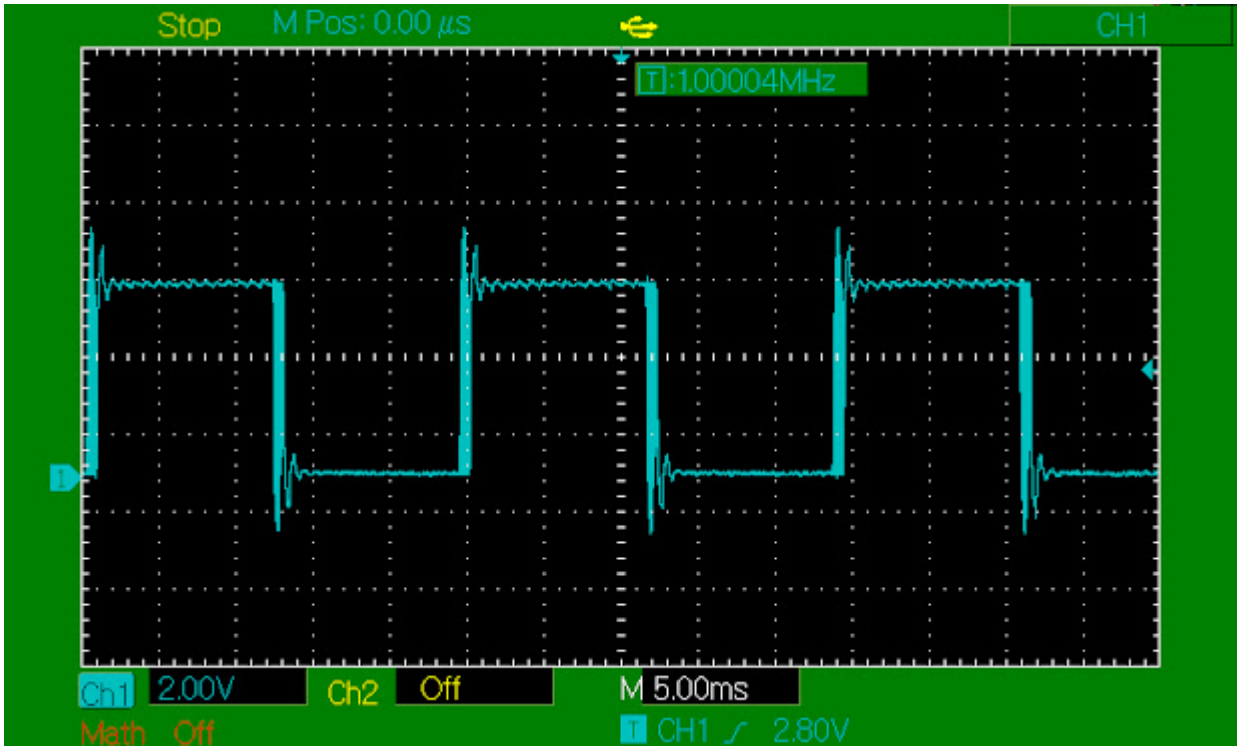
Szum losowy tak się oczywiście nazywa natomiast jak widać ma on sporo wspólnego z falą prostokątną. Poniżej analiza widmowa.



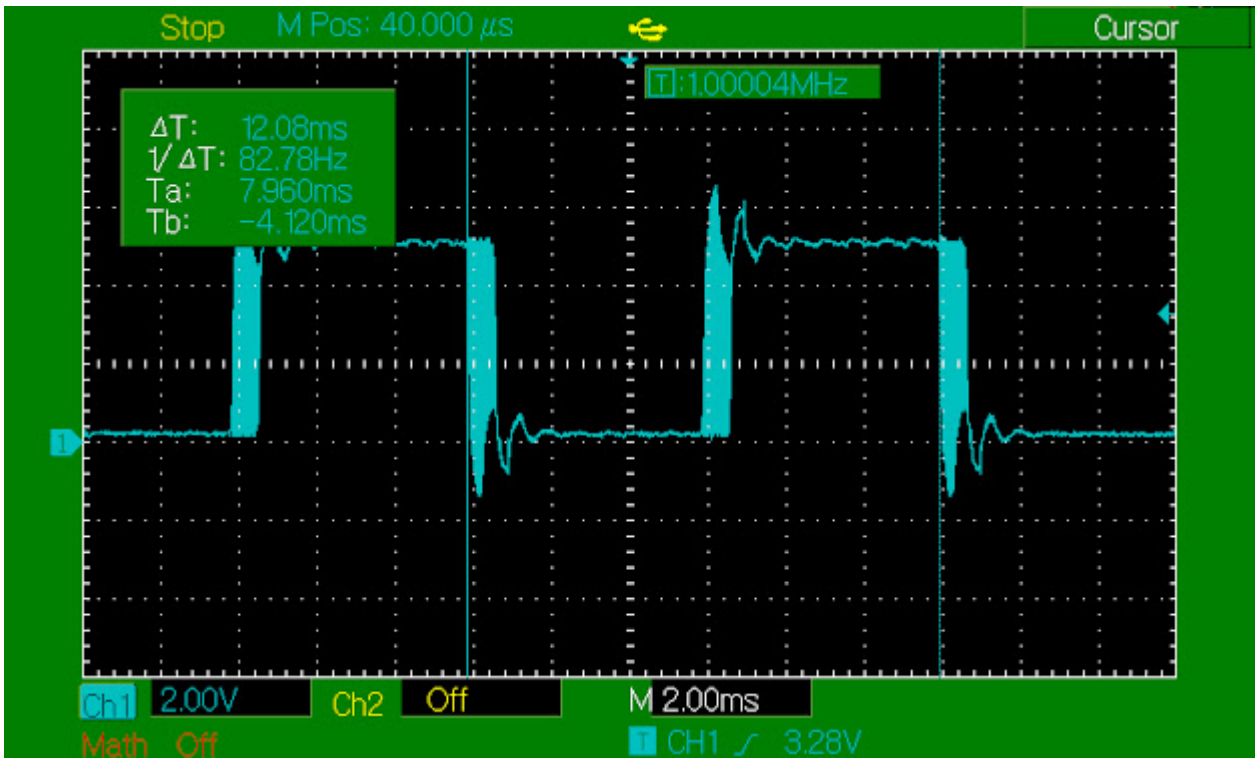
Jak widzimy FFT pokazuje że mamy tutaj sporo miejsc gdzie odznaczają nam się częstotliwości harmoniczne, spektrum prawdziwego szumu powinno być bardziej jednolite i przypominać wręcz wykres funkcji 1/x.

Przejdźmy do następnej funkcji tego generatora mianowicie tzw. „szybkiego” kanału na którym możemy uzyskać częstotliwości fali prostokątnej 1,2,4 i 8 MHz. Kanał ten pozwala tylko na uzyskanie przebiegu prostokąta oraz wybrać tylko pojedyncze częstotliwości nie pozwala na płynną regulację częstotliwości.

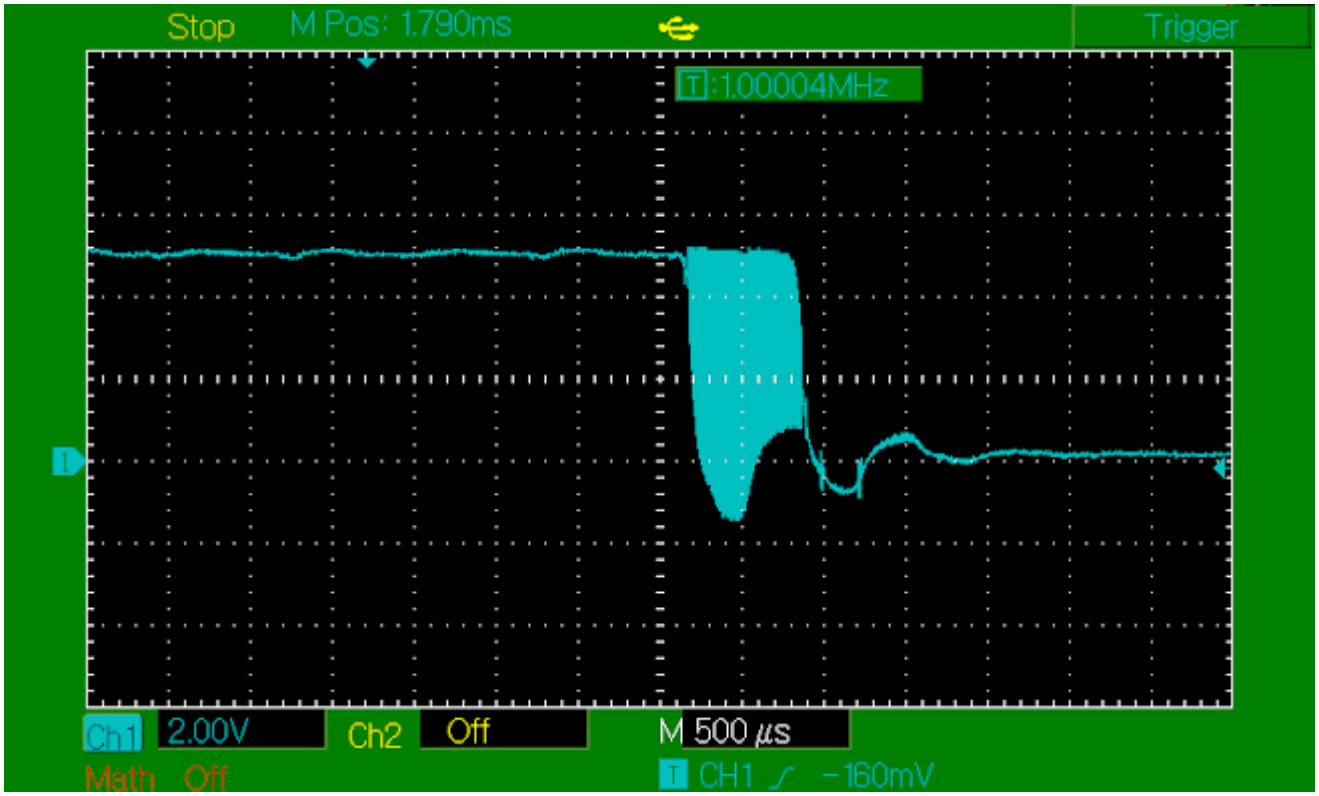
Poniżej prostokąt jaki udało mi się uzyskać z tego kanału.



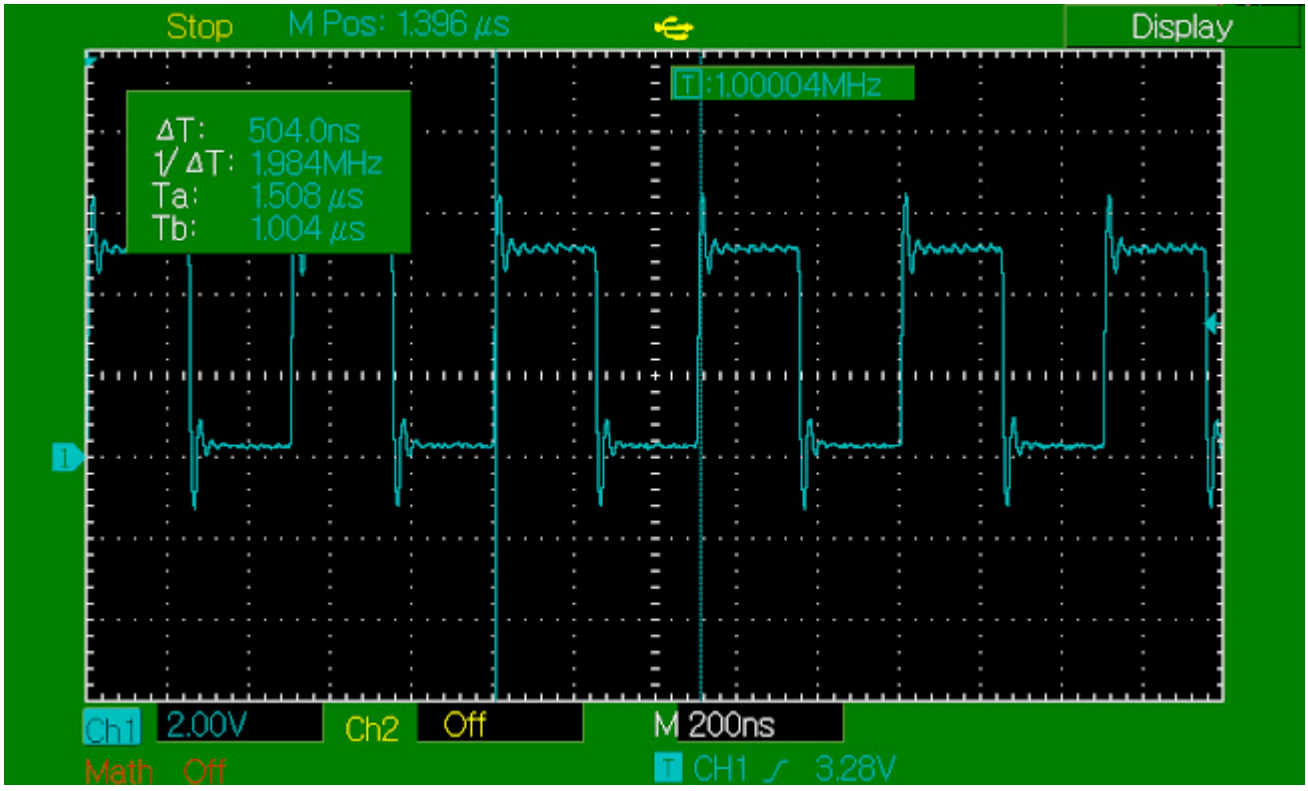
Wygląda on trochę podejrzanie. Miernik częstotliwości pokazuje 1 MHz ale patrząc na podziałkę na osi czasu widzimy że coś tutaj jest nie tak. Zmierzmy tą częstotliwość za pomocą kursorów. Najpierw oczywiście przybliżając dla dodatkowej dokładności



No i proszę bardzo ... to nie ta częstotliwość o którą chodzi. No ale nadal nie wiemy co to są za oscylacje przy zboczach prostokąta. Przybliżmy i zobaczmy co to jest.



Coraz bliżej ...



No i proszę bardzo ... jest i nasz przebieg tutaj akurat pomiar był wykonywany dla częstotliwości 2 MHz. Co spostrzegawczy pewnie się domyślają. Tak każda częstotliwość na „szybkim kanale” tego generatora tak wygląda. Kompletnie się nie nadaje do jakiegokolwiek wykorzystania.

Podsumowując Generator jest dobry jak chcecie coś podręcznego do wygenerowania prostych przebiegów do wykorzystania w obwodach analogowych. Natomiast do wykorzystania w obwodach logicznych nie próbowałbym.

Ja będę okazjonalnie korzystał z tego generatora głównie na szkoleniach oraz do przeprowadzania prostych testów takich jak ten ;) na mojego bloga.

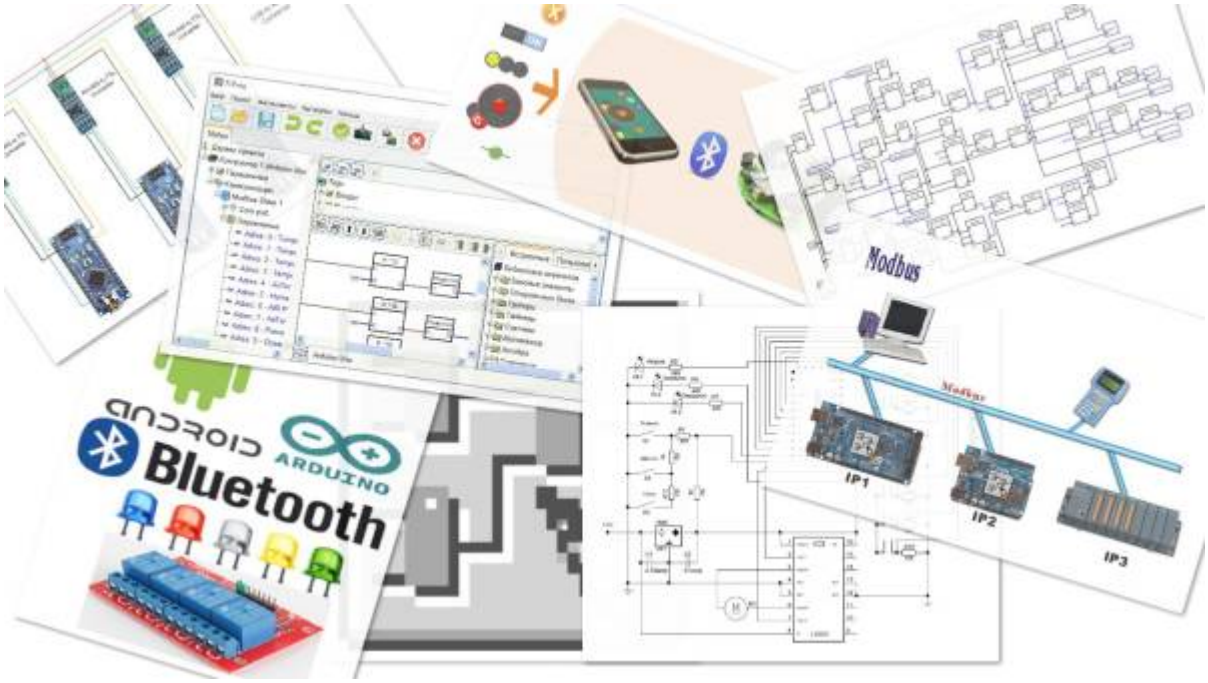
Nie testowałem tutaj obciążalności wyjścia tego generatora gdyż nie do tego służą generatory, oraz nie ma co testować składowej stałej jak i sterowania amplitudą ponieważ są to czysto analogowe obwody i nie mam im nic do zarzucenia.

Dla początkujących elektroników funkcje oferowane przez ten generator będą świetne: wiele różnych przebiegów, cyfrowa nastawa częstotliwości, stabilna częstotliwość, stosunkowo małe zniekształcenia sinusoidy itp. Cena bardzo dobra jakość wykonania również natomiast generowane przez niego przebiegi nie są najlepsze.

2025/05/09 10:39 · administrator · 0 Komentarz

FLprog - Programowanie Arduino dla opornych, Wstęp

oryginalny wpis na blogu: sierpnia 27, 2022



Jest wiele różnych programów pozwalających na programowanie graficzne arduino tj.: ardublock, XODide, MyOpenLab. Natomiast ten program w porównaniu do innych jest zewnętrzny nie jest dodatkiem do Arduino IDE tak jak ardublock. Nie jest zupełnie nowym językiem programowania z bardzo wysokim progiem wejścia, ani również nie wymaga podłączenia do komputera tak jak MyOpenLab. FLprog jest to kompletnie zewnętrzny program który generuje kod który możemy wgrać do arduino. Po zinterpretowaniu naszych graficznych wypocin program po prostu otwiera okno arduino IDE z naszym kodem.

Na stronie autora https://flprog.ru/en/ w zakładce projects możemy zobaczyć wiele rzeczy które społeczność programu zrobiła. Można tam znaleźć projekty sterowników do pieca grzewczego aż po programiki do małych zabawek i robocików.

Program opiera się na bardzo prostej zasadzie mianowicie mamy bloki/klocki z których układamy nasz program, następnie łączymy je przewodami/łącznikami co pozwala na przepływ danych lub stanu

logicznego z jednego bloku do drugiego. Pozwala to tak naprawdę na natychmiastową realizację jakiekolwiek pomysłu z praktycznie zerowym progiem wejścia.

Mi udało się do tej pory w tym programie stworzyć coś na wzór czujnika parkowania. Jest to urządzenie które za pomocą czujnika ultra dźwiękowego wydaje periodyczne sygnały dźwiękowe których okresowość jest zależna od dystansu przed czujnikiem ultradźwiękowym. Mówiąc prościej czujnik piszczy częściej jeżeli coś się do niego zbliża.

Poniżej przedstawiam proces tworzenia takiego projektu.

Najpierw musimy znaleźć lub zakupić części, w tym projekcie wykorzystano poniższe części:

- Czujnik zbliżeniowy HCSR04
- Mikrokontroller Arduino UNO

FLprog wspiera Arduino Uno, Arduino Leonardo, Arduino MEGA i wiele innych

- Moduł Wzmacniacza Audio 18W TDA2030 18W

Buzzery do arduino mają bardzo drażniący dzwięk to pozwoli na lekkie zniwelowanie tego efektu

- Jakiś Stary głośnik

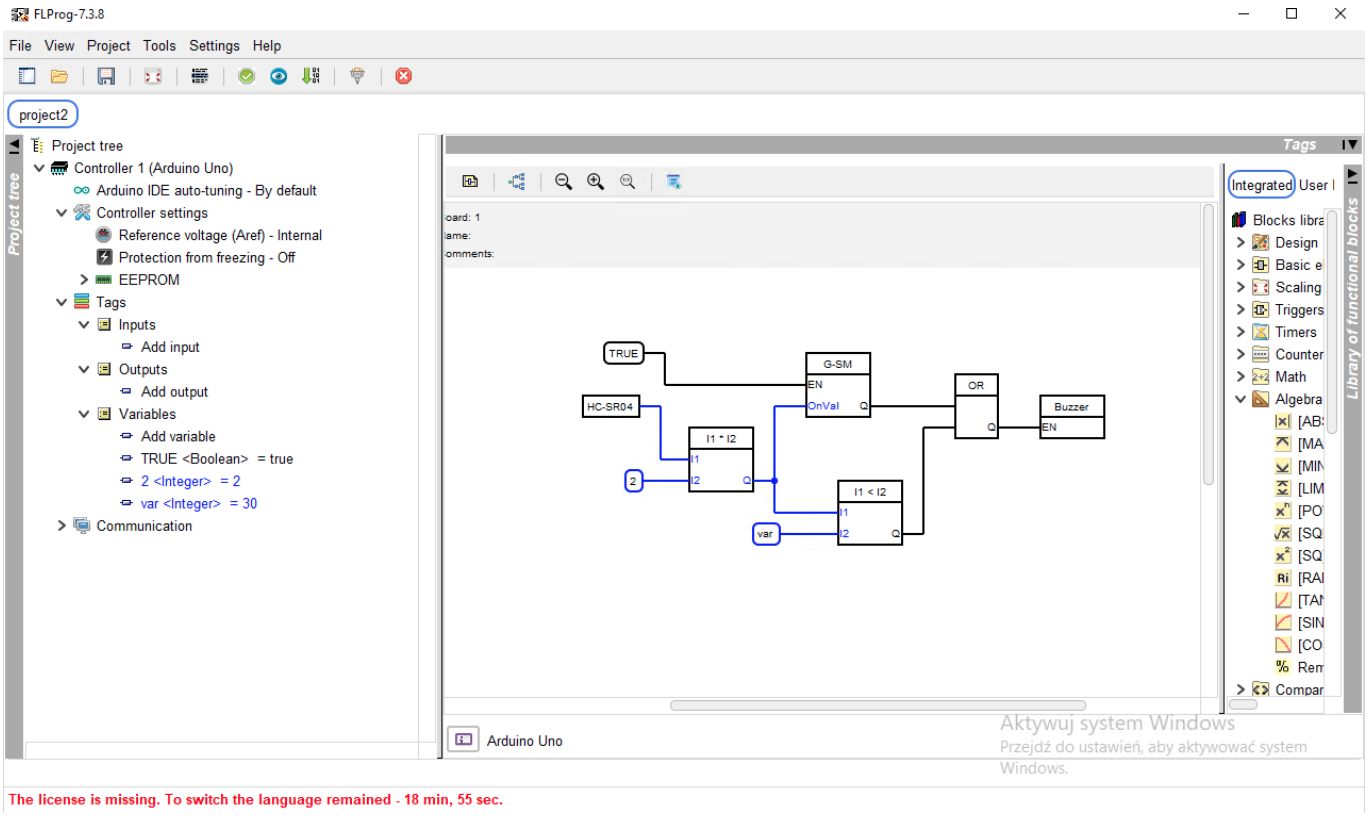
impedancji od 3 ohm do 16 ohm

- Zródło zasilania

W moim projekcie arduino jest zasilane z zasilacza warsztatowego Natomiast wzmacniacz audio jest zasilany z linii 5V z arduino gdyż nic tutaj nam nie będzie pobierać zawrotnych wartości prądu

Potem należy wszystko podłączyć. Czujnik Podłączamy do pinów arduino nie jest istotne które konkretnie ważne aby miały znaczek ~ bo to oznacza że wpierają PWM a to jest dla nas istotne, w programie potem można skorygować numer wykorzystanego pinu. Następnie podłączamy wzmacniacz, podłączamy jego zasilanie podłączamy głośnik oraz na wejście wzmacniacza podłączmy jeden z pinów.

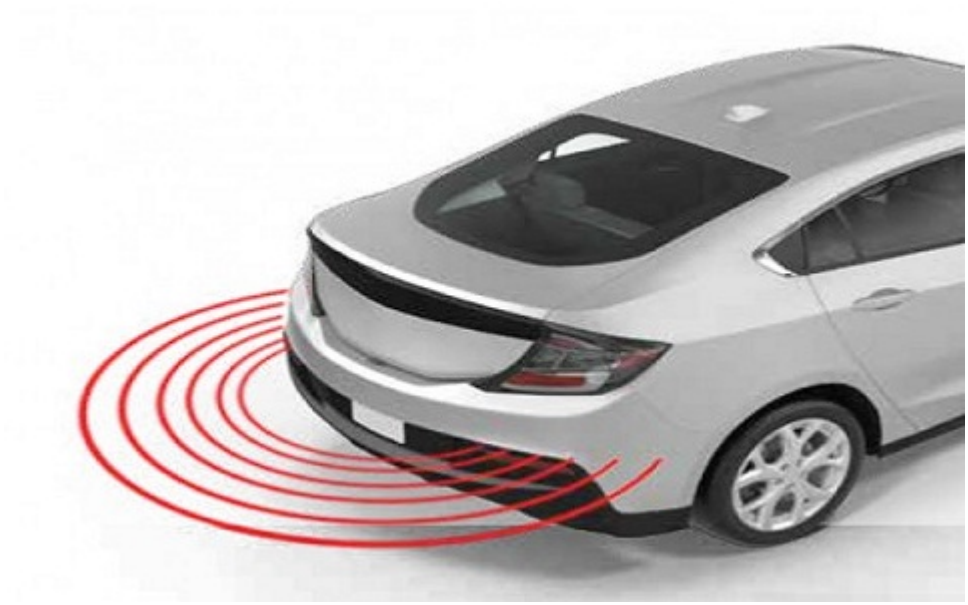
Następnie otwieramy program i zaczynamy zabawę. Poniżej mamy już cały wykonany projekt. Przy tworzeniu projektu naciskamy File>New>New Project for the controller a następnie w dodatkowym okienku wybieramy opcję FBD (function block diagram) czyli właśnię tą opcję o którą nam chodzi. Z listy wybieramy mikrokontroller który nas interesuję.



W ten sposób stworzyliśmy nowy pusty projekt.

Teraz warto wyjaśnić co mamy na ekranie bo nie jest tego mało. Po prawej stronie interfejsu mamy dostępny bloki które pozwalają nam tworzyć program. Po lewej stronie mamy widoczne nasze: zmienne, wejścia i wyjścia arduino, porty komunikacyjne, pamięć wewnętrzną oraz zabezpieczenie przez zacięciem mikrokontrollera. Następnie żeby zacząć prace normalnie musieli byśmy dodać wejścia i wyjścia natomiast w tym przypadku będziemy korzystać z gotowych bloków przystosowanych do naszych elementów. Zaczynamy od przeciągnięcia z listy z prawej strony bloku naszego czujnika HC-SR04 zakładka sensors>distance sensor. Klikając dwa razy w te blok możemy ustawić na których pinach podłączyliśmy sygnały Echo i Trig naszego sensora. W nocie katalogowej możemy wyczytać że sensor ten maksymalnie jest w stanie wykryć obiekty oddalone o 4. W tym oknie ustawiamy również ten dystans. Teraz mamy blok funkcyjny który na wyjściu podaje nam w formacie integer wartość odległości obiektu od sensora. FLprog podświetla nam takie połączenia jak i też zmienne integer na niebiesko. Teraz z racji tego że nasza liczba z sensora raczej będzie mała ponieważ interesują nas obiekty blisko, z zamysłem że budujemy czujnik prakowania. W związku z tym tworzymy zmienną integer o wartości 2 oraz pobieramy blok mnożenia math>MUL(*). Łączymy wszystko tak jak na obrazku, mnożenie jest przemienne więc nie ważne do którego wejścia bloku podłączymy co. Następnie chcemy generować impulsy od długości proporcjonalnej do odległości. Pobieramy z zasobnika blok generatora Timers>Generator. Następnie w parametrach tego bloku ustawiamy nasz generator jako symetryczny multiwibrator a następnie wybieramy że długośc impulsów jest zależna od zewnętrznego parametru. Generator musimy również włączyć więc tworzymy zmienną boolean o wartości true i łączymy ją do wejścia enable. Następnie musimy stworzyć warunek że poniżej pewnej odległości głośnik zacznie piszczeć w sposób ciągły. Wykorzystujemy do tego blok comparison>comparator. I wybieramy odpowiednie porównanie tworzymy zmienną dla minimalnej odległości i łączymy wszystko w odpowiedni sposób i teraz nasz komparator będzie miał wyjście w stanie HIGH wtedy kiedy odległość spadnie poniżej około 30cm. Następnie musimy w jakiś sposób połączyć te dwa sygnał w taki sposób że każdy z nich jest równoważny czyli jak potrzeba to generator włącza głośnik. Osoby które znają bramki logiczne powiedzą ... tutaj pasuje OR i tak to prawda właśnie o taką bramkę chodzi, podłączamy ją. I teraz zostaje ostatni blok other>Piezo Speaker, dodajemy go do projektu podłączamy do wyjścia bramki OR

i ustawiamy parametry, w ustawieniach ustawiamy pin który podłączyliśmy do wejścia wzmacniacza, wybieramy również opcję continuosly opcja jest wyjaśniona w okienku, a następnie częstotliwość jaką chcemy żeby generował, ja ustawiłem na 1kHz uważam że jest stosunkowo mało irytujący taki dzwięk.



W ten sposób mamy czujnik parkowania co prawda bardzo prosty natomiast działa. Zapraszam do sprawdzenia innych moich projektów oraz polecam samemu zacząć przygodę z arduino i FLprog.

2025/05/09 10:24 · administrator · 0 Komentarz

Aktualności - blog

Najnowsze wpisy:

Ludzie którzy wpłynęli na kształt współczesnej informatyki

źródło: Wikimedia.org

Ten artykuł jest moim subiektywnym spisem osób o których mało się mówi nauczając historii informatyki, lub mało się słyszy o tych osobach w mediach. To zestawienie to zbiór interesujących osób które moim zdaniem miały większy wpływ na informatykę niż się nam wszystkim wydaje. Wybrałem tutaj osoby o których swego czasu czytałem bardzo dużo i zaciekawiła mnie ich historia lub dokonania w świecie informatyki. Sporo z tych osób jest wymienione tutaj ze względu na swój wpływ na społeczność hackerską, sam jestem zwolennikiem kultury i zasad hackerskich o czym można przeczytać na stronie głównej tej wiki. Zbiór nie ma konkretnej kolejności, jest podzielony na kategorie tematyczne, które opisują na jaką sferę informatyki miały wpływ osoby w nich wymienione.

→Czytaj więcej...

2025/05/11 12:21 · administrator · 0 Komentarz

Wyszkolenie własnego modelu AI

źródło: https://commons.wikimedia.org/wiki/File:Artificial-Intelligence.jpg

Wikipedia

> A large language model (LLM) is a type of machine learning model designed for natural language processing tasks such as language generation. LLMs are language models with many parameters, and are trained with self-supervised learning on a vast amount of text.

Od pewnego czasu chodzi za mną pomysł żeby wyszkolić jakichś otwarty model LLM np. Ollama, wyszkolić go wszystkimi materiałami z mojego dysku lub z tej wiki i potem opublikować go na mojej stronie jako chat bot którego można sie o wszystko zapytać.

→Czytaj więcej...

2025/05/09 10:49 · administrator · 0 Komentarz

Test generatora DDS FNIRSI

oryginalny wpis na blogu: lipca 12, 2023

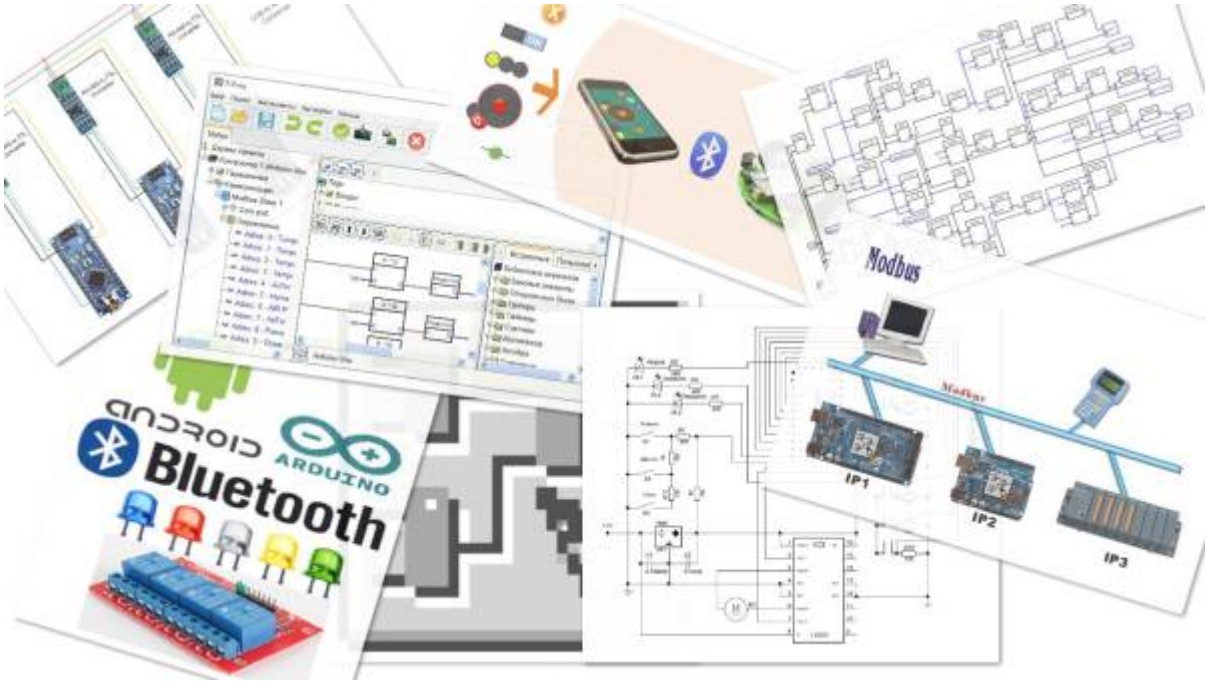
Nie zawsze chce mi się wyciągać mój duży generator który waży około 20kg, ma on co prawda bardzo dobre parametry ale nie zawsze mi potrzeba 10MHz sinusoidy z współczynnikiem zniekształcenia mniejszym niż 1%. Czasami potrzebuję zwykłej sinusoidy ze stabilną częstotliwością. Siedząc kiedyś na allegro zauważyłem taki prosty generator DDS FNIRSI, malutki, lekki zasilany z zasilacza, pomyślałem czemu nie. Wiem że jeżeli jest jakaś przeszkoda stojąca pomiędzy mną a zrobieniem nowego projektu to nie rozpocznę pracy nad nowym projektem. I w taki sposób stałem się posiadaczem tego malutkiego generatora. W tym materiale postaram się sprawdzić czy taki generator będzie przydatny w warsztacie elektronika.

→Czytaj więcej...

2025/05/09 10:39 · administrator · 0 Komentarz

FLprog - Programowanie Arduino dla opornych, Wstęp

oryginalny wpis na blogu: sierpnia 27, 2022



Jest wiele różnych programów pozwalających na programowanie graficzne arduino tj.: ardublock, XODide, MyOpenLab. Natomiast ten program w porównaniu do innych jest zewnętrzny nie jest dodatkiem do Arduino IDE tak jak ardublock. Nie jest zupełnie nowym językiem programowania z bardzo wysokim progiem wejścia, ani również nie wymaga podłączenia do komputera tak jak MyOpenLab. FLprog jest to kompletnie zewnętrzny program który generuje kod który możemy wgrać do arduino. Po zinterpretowaniu naszych graficznych wypocin program po prostu otwiera okno arduino IDE z naszym kodem.

→Czytaj więcej...

2025/05/09 10:24 · administrator · 0 Komentarz
2025/05/07 16:04 · administrator · 0 Komentarz

Ciekawe tematy do realizajci w przyszłości

Teoria Węzłów

Wikipedia

> In topology, knot theory is the study of mathematical knots. While inspired by knots which appear in daily life, such as those in shoelaces and rope, a mathematical knot differs in that the ends are joined so it cannot be undone, the simplest knot being a ring (or „unknot”). In mathematical language, a knot is an embedding of a circle in 3-dimensional Euclidean space, $\displaystyle \mathbb {E} ^{3}$. Two mathematical knots are equivalent if one can be transformed into the other via a deformation of $\displaystyle \mathbb {R} ^{3}$ upon itself (known as an ambient isotopy); these transformations correspond to manipulations of a knotted string that do not involve cutting it or passing it through itself.

Może by coś takiego zasymulować na komputerze albo zrobić program który pokaże kombinacje węzłów dla pozycji elektronów w atomie zgodnie z modelem kwantowym atomu?

Wyszkolenie własnego modelu AI

Wikipedia

> A large language model (LLM) is a type of machine learning model designed for natural language processing tasks such as language generation. LLMs are language models with many parameters, and are trained with self-supervised learning on a vast amount of text.

Od pewnego czasu chodzi za mną pomysł żeby wyszkolić jakichś otwarty model LLM np. Ollama, wyszkolić go wszystkimi materiałami z mojego dysku lub z tej wiki i potem opublikować go na mojej stronie jako chat bot którego można sie o wszystko zapytać.

Sentient AI: Czy to możliwe?

Wikipedia

> Artificial consciousness,[1] also known as machine consciousness,[2][3] synthetic consciousness,[4] or digital consciousness,[5] is the consciousness hypothesized to be possible in artificial intelligence.[6] It is also the corresponding field of study, which draws insights from philosophy of mind, philosophy of artificial intelligence, cognitive science and neuroscience.

Chciałbym się dowiedzieć czy matematycznie możliwe jest obliczyć jaka sieć neuronowa i jakie moce obliczeniowe byłyby potrzebne do wymodelowanie ludzkiego mózgu jako sieć neuronowa. Jeżeli coś takiego jest możliwe to chciałbym napisać program który obliczy ile neuronów taka sieć neuronowa by miała oraz ile lat zajęłoby jej uczenie żeby była na zadowalającym poziomie wiedzy.