

Instrukcja do ćwiczeń laboratoryjnych
z przedmiotu
Wyposażenie pokładowe.

SERIA I

- Lab. 1. Badanie wpływu temperatury na wskazania sensorów, kompensacja temperaturowa
- Lab. 2. Badanie czujników przyspieszenia – skalowanie, modelowanie
- Lab. 3. Badanie czujników prędkości kątowej – skalowanie, modelowanie
- Lab. 4. Badanie układów wizualizacji informacji pilotażowo nawigacyjnej

SERIA II

- Lab. 5. Projekt i badanie układu wyznaczania orientacji przestrzennej i kursu I
- Lab. 6. Projekt i badanie układu wyznaczania orientacji przestrzennej i kursu II

Dodatki

- Dodatek A. Obsługa programu KEIL (do lab. 2 i lab. 3)
- Dodatek B. Obsługa programu CAN_Monitor (do lab. 2 i lab. 3)
- Dodatek C. Obsługa programu PomiaryGyro (do lab. 1)
- Dodatek D. Modelowanie czujników (do lab. 2 i lab. 3)
- Dodatek E. Instalacja instalacji Simulink Dekstop Real Time

Laboratorium 1.

Badanie wpływu temperatury na wskazania sensorów, kompensacja temperaturowa.

1. Przebieg zajęć

- przygotowanie badanego układu sensorów i systemu rejestrującego, podłączenie zasilania, sprawdzenie poprawności komunikacji
- włączenie rejestracji
- ustawienie zadanej temperatury, chłodzenie/grzanie do osiągnięcia stabilnej temperatury, rejestracja danych (minimum 5 wartości)

UWAGA! Na zajęcia potrzebny jest przynajmniej jeden komputer z arkuszem kalkulacyjnym (np. Ms Excel).

2. Zadania do wykonania

- zaproponowanie kształtu i współczynników funkcji kompensującej
- wyznaczenie błędów wskazań w funkcji temperatury

3. Przygotowanie teoretyczne

- błędy sensorów lotniczych
- błędy powodowane wpływami środowiskowymi
- sposoby kompensacji wpływu temperatury na odczyt czujnika

W trakcie zajęć należy zbadać wpływ temperatury na wskazania czujnika prędkości kątowej. Należy zaproponować kształt i współczynniki funkcji kompensującej wpływ temperatury.

Laboratorium 2.

Badanie czujników przyspieszenia – skalowanie, modelowanie.

1. Przebieg zajęć

- przygotowanie badanego układu sensorów i systemu pomiarowego, podłączenie zasilania, sprawdzenie poprawności transmisji
- uruchomienie rejestracji danych
- zadawanie przyspieszeń wzorcowych na 3 osiach o wartościach +1g, 0g, -1g i odczyt wskazań sensorów w programie CAN_Monitor
- przetworzenie zarejestrowanych danych do formatu tekstowego w celu dalszej obróbki

UWAGA! Na zajęcia potrzebny jest przynajmniej jeden komputer z arkuszem kalkulacyjnym (np. Ms Excel), oraz komputer z pakietem Matlab/Simulink, z zainstalowanymi co najmniej dodatkami Aerospace Toolbox i Aerospace Blockset.

2. Zadania do wykonania

- zaproponowanie kształtu i współczynników funkcji skalującej
- wyznaczenie błędów wskazań
- stworzyć model przyspieszeniomierza w pakiecie Matlab/Simulink

3. Przygotowanie teoretyczne

- oznaczenia wartości przyspieszeń i prędkości kątowych
- pomiar przyspieszeń i prędkości kątowych
- sensory MEMS, budowa, zasada działania

W trakcie zajęć należy wykonać skalowanie czujników przyspieszenia. Zadane wartości przyspieszeń to : -1g , 0g, +1g.

Należy wyznaczyć kształt i współczynniki funkcji odwzorowującej wartość z przetwornika analogowo-cyfrowego na przyspieszenie.

Po wyznaczeniu funkcji skalującej należy ją wprowadzić do oprogramowania badanego urządzenia, a następnie ponownie dokonać porównania wartości zadanej z odczytaną i wyznaczyć różnicę.

Przedyskutować otrzymane wyniki.

W ramach zadania należy stworzyć model czujnika w oprogramowaniu Matlab/Simulink

Laboratorium 3.

Badanie czujników prędkości kątowej – skalowanie, modelowanie.

1. Przebieg zajęć

- przygotowanie badanego układu sensorów i systemu pomiarowego, podłączenie zasilania, sprawdzenie poprawności transmisji
- włączenie rejestracji danych
- zadawanie prędkości kątowych w zakresie +/- 300 st/s, w jednej osi, odczyt wskazań sensorów w programie CAN_Monitor
- przetworzenie zarejestrowanych danych do formatu tekstowego w celu dalszej obróbki

UWAGA! Na zajęcia potrzebny jest przynajmniej jeden komputer z arkuszem kalkulacyjnym (np. Ms Exel), oraz komputer z pakietem Matlab/Simulink, z zainstalowanymi co najmniej dodatkami Aerospace Toolbox i Aerospace Blockset.

2. Zadania do wykonania

- zaproponować kształt i współczynniki funkcji skalującej
- wyznaczyć błędy wskazań
- stworzyć model przyspieszeniomierza w pakiecie Matlab/Simulink

3. Przygotowanie teoretyczne

- oznaczenia wartości przyspieszeń i prędkości kątowych
- pomiar przyspieszeń i prędkości kątowych
- sensory MEMS, budowa, zasada działania
- giroskopy optyczne (FOG, RLG), budowa, zasada działania

W trakcie zajęć należy wykonać skalowanie czujników prędkości kątowej.

Badany układ należy umieścić na stole obrotowy. Następnie należy zadawać obroty stołu (w zakresie +/- 300 st/s) i dokonać odczytu wskazania z czujnika prędkości kątowej. Pozyskane dane mają posłużyć do wyznaczenia funkcji skalującej. Po wyznaczeniu funkcji skalującej należy wyznaczyć różnicę wskazań między stołem obrotowym a wyskalowanym czujnikiem. Przedyskutować otrzymane wyniki.

W ramach zadania należy stworzyć model czujnika w oprogramowaniu Matlab/Simulink

Laboratorium 4.

Badanie układów wizualizacji informacji pilotażowo-nawigacyjnych

W trakcie zajęć należy zapoznać się ze sposobem przekazywania informacji pilotażowo-nawigacyjnych na ekranach systemu EFIS

Laboratorium 5.

Projekt i badania układu orientacji przestrzennej i kursu I

1. Przebieg zajęć

- w zależności od potrzeb stworzyć schemat układu odczytywania i przetwarzania danych (rejestracja lub badanie on-line)
- uruchomienie symulacji, sprawdzenie uzyskanych wyników

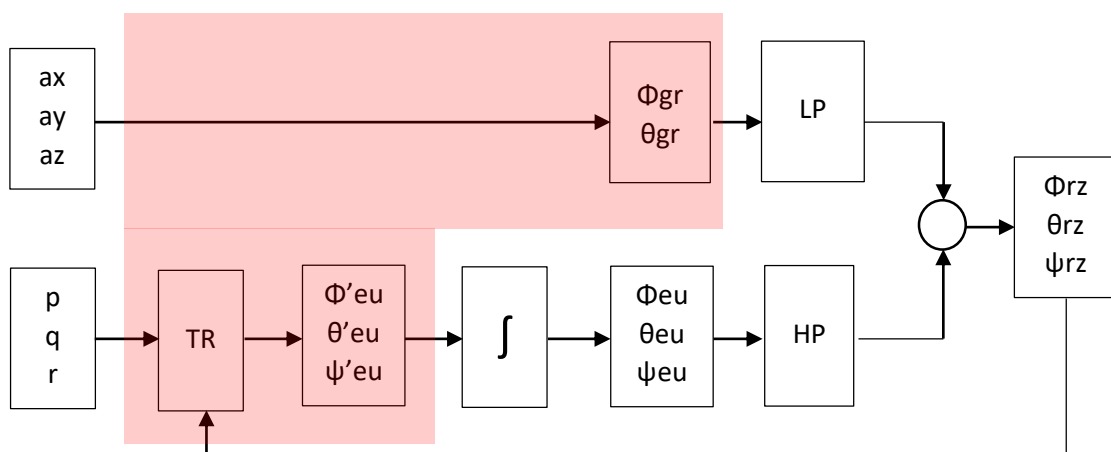
2. Zadania do wykonania

- zaprojektowanie układu AHRS zgodnego ze schematem
- przygotowanie funkcji wyznaczających kąty grawitacyjne
- przygotowanie funkcji przeliczających wskazania prędkości kątowych mierzone w układzie samolotu do wskazań w układzie Ziemi (przekształcenie Tait'a-Bryan'a)
- sprawdzenie poprawności działania zastosowanych przekształceń (przeliczeń).

3. Przygotowanie teoretyczne

- Matlab/Simulink – obsługa
- AHRS – budowa i zasada działania
- Układy odniesienia w lotnictwie, przejście między układami odniesienia

W trakcie zajęć należy w środowisku symulacyjnym Matlab/Simulink przygotować schemat pozwalający na importowanie danych, w tym przyspieszeń i prędkości kątowych mierzonych w układzie samolotu. W kolejnym kroku zadaniem jest przygotowanie elementów schematu pozwalających na wyznaczenie kątów grawitacyjnych oraz wyznaczenie prędkości kątowych mierzonych w układzie Ziemi (część schematu zaznaczona na czerwono).



Laboratorium 6.

Projekt i badania układu orientacji przestrzennej i kursu II

1. Przebieg zajęć

- w zależności od potrzeb stworzyć schemat układu odczytywania i przetwarzania danych (rejestracja lub badanie on-line)
- uruchomienie symulacji, sprawdzenie uzyskanych wyników

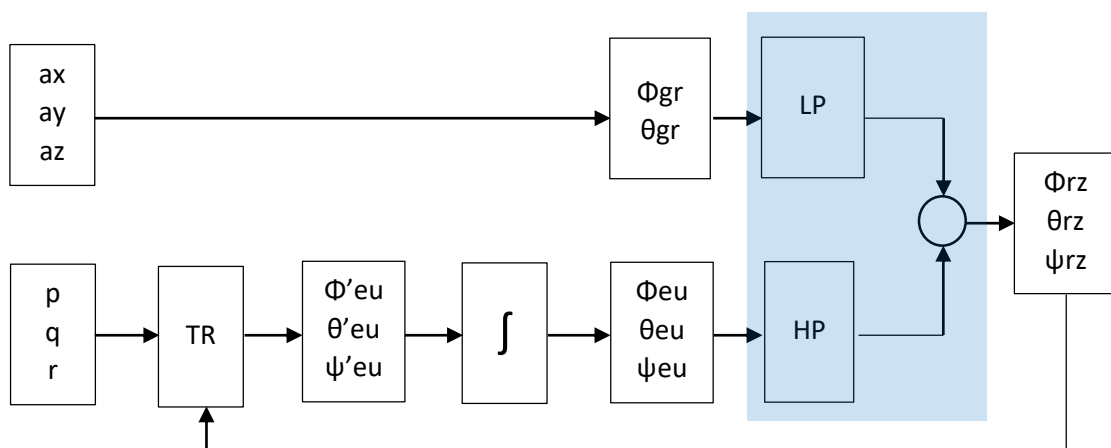
2. Zadania do wykonania

- zaprojektowanie układu AHRS zgodnego ze schematem
- budowa dyskretnego filtra komplementarnego
- dobór wartości stałych czasowych filtra komplementarnego
- odłączanie korekcji
- sprawdzenie poprawności działania zastosowanych przekształceń (przeliczeń).

3. Przygotowanie teoretyczne

- Matlab/Simulink – obsługa
- AHRS – budowa i zasada działania
- Układy odniesienia w lotnictwie, przejście między układami odniesienia
- Transmitancja operatorowe filtrów dolnoprzepustowego i górnoprzepustowego

W drugiej części zajęć należy w środowisku symulacyjnym Matlab/Simulink do wcześniejszego schematu dodać układy filtrów komplementarnych, dobrać wartości stałych czasowych dla tych filtrów, a także zaimplementować elementy układu odłączania korekcji w trakcie zakrętu (część schematu zaznaczona na niebiesko).



Laboratorium 7.

Kolokwium zaliczeniowe

- Prędkości i wysokości stosowane w lotnictwie
- Pomiar przyspieszeń i prędkości kątowych, budowa giroskopów MEMS, FOG, RLG
- Błędy sensorów pomiarowych, ich eliminacja
- Budowa i zasada działania układów orientacji przestrzennej
- Sensory MEMS - budowa, zasada działania
- Wskazania informacji z systemów pilotażowo-nawigacyjnych.

Literatura:

Polak Z., Rypulak A., Awioniki, przyrządy i systemy pokładowe

Bociek S., Gruszeck J., Układy sterowania automatycznego samolotem

Tomczyk A., Pokładowe cyfrowe systemy sterowania samolotem

Gosiewski Z., Ortyl A., Algorytmy inercjalnego bezkardanowego systemu orientacji i położenia obiektu i ruchu przestrzennym

Noty techniczne sensorów MEMS i sensorów ciśnienia (<https://lwalek.v.prz.edu.pl>)

Dodatek A. Obsługa programu KEIL

Podczas ćwiczeń będziesz korzystał z oprogramowania KEIL

Ikona programu znajdują się na pulpicie komputera



1. Uruchom środowisko KEIL, otwórz projekt zgodny z przedmiotem i rokiem studiów, otwórz plik „student.h”
2. Wykonaj modyfikacje pliku „student.h” zgodnie z zamieszczoną w nim instrukcją

```
20
21 //*****
22 //***** INSTRUKCJA UZYTEKOWANIA KODU *****
23 //*****
24 //-----
25 //----- NIE WYKONYWAC W KODZIE ZMIAN INNYCH NIZ OPISANE W INSTRUKCJI -----
26 //----- NIE MODYFIKOWAC LINII Z OPISAMI (NA ZIELONO) I Z GWAZDKAMI (*) -----
27 //----- ZANIM COS ZMIENISZ PRZECZYTAJ CALOSC INSTRUKCJI -----
28 //-----
29 // Aby zainicjowac poprawnie przygotowac badany ukklad do cwiczenia nalezy go //
30 // zaprogramowac. W programie musi byc umieszczona definicja grupy laboratoryjnej, //
31 // zgodnie z zamieszczonym nizej przykladem. W pliku moze byc zdefiniowana tylko //
32 // jedna grupa. Usun definicje inne niz przykladowa i definicje twojej grupy //
33 // Grupa laboratoryjna 4, podgrupa 6 - przyklad //
34 // #define _NR_GRUPY_ WP2018_L46_ //
35 //-----
36 // Zdefiniuj swoja grupe w kolejnej linijce (nr 37)
37 #define NR GRUPY_ WP2018_L11
38 //-----
39 // Zdefiniuj wartosc stalej czasowej (linia nr 43)
40 // stala czasowa filtru rekurencyjnego dla funkcji LiczACC i LiczGYRO (ms)
41 // dla cwiczenia 1, 2, 3 ustaw 2500
42 // dla cwiczenia 4 ustaw 20
43 #define TEES 1500
44 //-----
45 //-----
46 // 1. Zapisz aktualna wersje pliku (CTRL+S lub odpowiednia ikonka) //
47 // W tym pliku nie musisz dokonywac juz zadnych zmian //
48 // 2. Przekompiluj kod (skrót - F7) //
49 // 3. Uruchom DEBUGER (CTRL+F5), kod zostanie wgrany do pamieci procesora. //
50 // 4. Uruchom program (F5) //
51 // 5. Opusc tryb debugowania (CTRL+F5) //
52 // Odlacz przewod do programowania (zgodnie z zaleceniami prowadzacego zajecia) //
53 // Przejdz do wykonywania kolejnej czesci cwiczenia. //
54 // W dalszej czesci cwiczenia bedziesz korzystal z pliku student.c //
55 //-----
56
```

Modyfikuj tylko linie zaznaczone czerwoną ramką. Po zakończeniu modyfikacji umieść program w pamięci mikrokontrolera i uruchom go zgodnie z instrukcją. (Dodatkowa instrukcja obrazkowa znajduje się na końcu dokumentu).

3. Zbierz dane pomiarowe zgodnie z ćwiczeniem i zaleceniami prowadzącego. Dane odczytaj w programie CAN_Monitor, identyfikatory zgodnie z umieszczonymi w instrukcji w pliku „student.h”.
4. Dane wprowadź do arkusza kalkulacyjnego i wyznacz funkcje skalujące, zgodnie z zaleceniem prowadzącego.
5. Zapisz funkcje skalujące w odpowiedniej dla twojej grupy części pliku „student.c” – w przykładzie znajduje się to na żółtym polu.

```
20
21 //*****
22 //***** INSTRUKCJA UŻYTKOWANIA KODU *****//
23 //*****
24 //-----//
25 //----- NIE WYKONYWAC W KODZIE ZMIAN INNYCH NIŻ OPISANE W INSTRUKCJI -----//
26 //----- NIE MODYFIKOWAĆ LINII Z OPISAMI (NA ZIEŁONO) I Z GWIAZDKAMI (*) -----//
27 //----- ZANIM COS ZMIENISZ PRZECZYTAJ CAŁOŚĆ INSTRUKCJI -----//
28 //-----//
29 // Aby zainicjować poprawnie przygotować badany układ do ćwiczenia należy go //
30 // zaprogramować. W programie musi być umieszczona definicja grupy laboratoryjnej, //
31 // zgodnie z zamieszczonym niżej przykładem. W pliku może być zdefiniowana tylko //
32 // jedna grupa. Usun definicje inne niż przykładowa i definicje twojej grupy //
33 // Grupa laboratoryjna 4, podgrupa 6 - przykład //
34 // #define _NR_GRUPY_ _WP2018_L46_ //
35 //-----//
36 // Zdefiniuj swoją grupę w kolejnej linii (nr 37)
37 #define _NR_GRUPY_ _WP2018_L11_
38 //-----//
39 // Zdefiniuj wartość stałej czasowej (linia nr 43)
40 // stała czasowa filtra rekurencyjnego dla funkcji LiczACC i LiczGYRO (ms)
41 // dla ćwiczenia 1, 2, 3 ustaw 2500
42 // dla ćwiczenia 4 ustaw 20
43 #define TEES 1500
```

```

main.c student.c* student.h startup_stm32f4xx.s system_stm32f4xx.c
55 // sIMU.rawP - wartosc z czujnika predkosci katowej przechylenia, CAN_ID 1304 //
56 // sIMU.rawQ - wartosc z czujnika predkosci katowej pochylenia, CAN_ID 1303 //
57 // sIMU.rawR - wartosc z czujnika predkosci katowej odchylenia, CAN_ID 1305 //
58 //-----//
59 //-----//
60 //-----// #include "imu.h" //-----//
61 //-----// #include "adc.h" //-----//
62 //-----// #include "student.h" //-----//
63 //-----//
64 //-----// void WP2018 (unsigned char nrgrupy) ( //-----//
65 //-----// switch (nrgrupy) ( //-----//
66 //-----//
67 //-----//
68 //-----//
69 //-----// case _WP2018_L11_ : //-----//
70 //-----//
71 //-----// GRUPA L11 -----//
72 //-----// Ponizej wprowadzaj swój kod -----//
73 //-----//
74 //-----//
75 //-----//
76 //-----// break: // end of WP2018_L11 -----//
77 //-----//
78 //-----//
79 //-----//
80 //-----// case _WP2018_L12_ : //-----//
81 //-----//
82 //-----// GRUPA L12 -----//
83 //-----// Ponizej wprowadzaj swój kod -----//
84 //-----//
85 //-----//
86 //-----//
87 //-----//
88 //-----// break: // end of WP2018_L12 -----//
89 //-----//
90 //-----//
91 //-----//

```

6. Zaktualizuj oprogramowanie mikrokontrolera

```

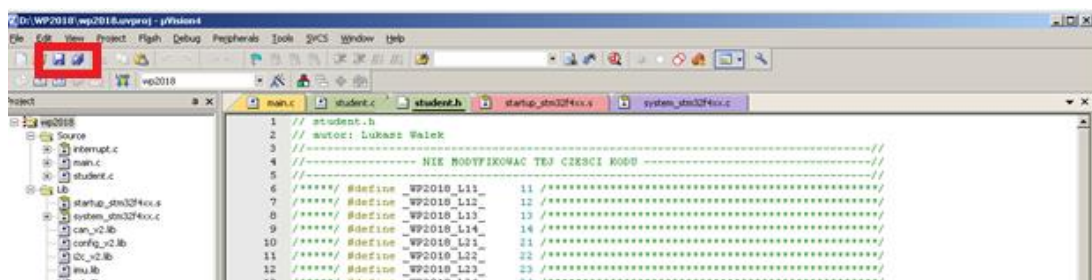
44 //-----//
45 //-----//
46 // 1. Zapisz aktualna wersje pliku (CTRL+S lub odpowiednia ikonka) //
47 // W tym pliku nie musisz dokonywac juz zadnych zmian //
48 // 2. Przekompiluj kod (skrót - F7) //
49 // 3. Uruchom DEBUGER (CTRL+F5), kod zostanie wgrany do pamieci procesora. //
50 // 4. Uruchom program (F5) //
51 // 5. Opusc tryb debugowania (CTRL+F5) //
52 // Odlacz przewod do programowania (zgodnie z zaleceniami prowadzacego zajecia) //
53 // Przejdz do wykonywania kolejnej czesci cwiczenia. //
54 // W dalszej czesci cwiczenia bedziesz korzystal z pliku student.c //
55 //-----//
56 //-----//

```

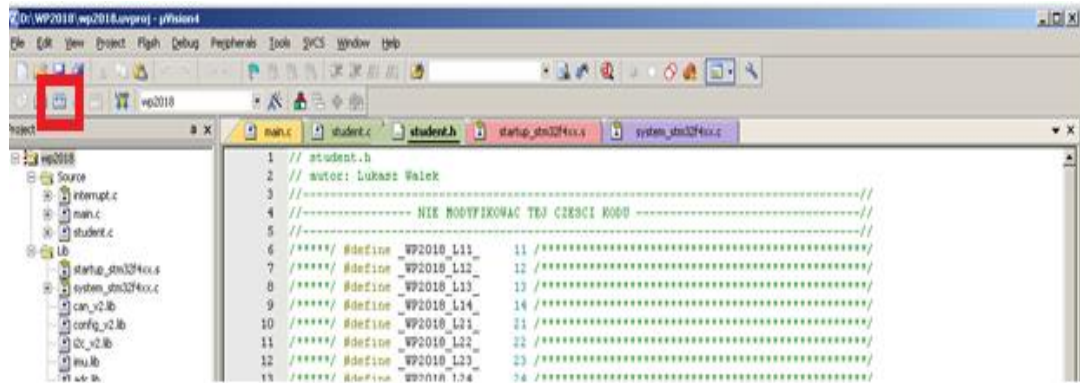
7. Wykonaj weryfikacje skalowania, zbierz wyniki pomiarów.

III. Programowanie mikrokontrolera

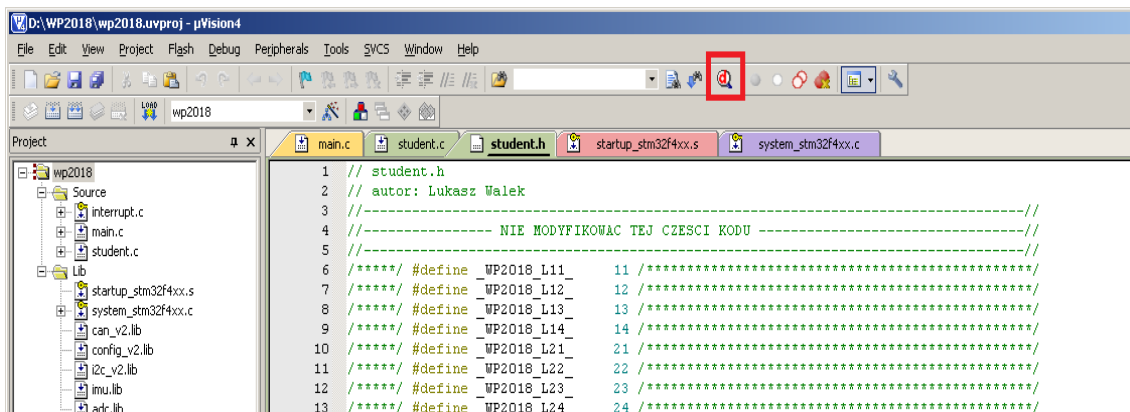
1. Zapisz zmodyfikowany kod (skrót klawiszowy CTRL + S lub ikona w czerwonej ramce)



2. Skompiluj kod (skrót klawiszowy F7 lub ikona w czerwonej ramce)



3. Podłącz programator do gniazda na płycie mikrokontrolera, zachowaj odpowiednia odległość przewodu i gniazda (biała strona wtyczki do białej strony gniazda)
4. Uruchom tryb DEBUG (skrót CTRL+F5 lub ikona w czerwonej ramce) – program ładuje się do pamięci mikrokontrolera.



5. Uruchom program (skrót F5)
6. Opuść tryb debugowania (skrót lub ikona jak w punkcie 4), odłącz przewód programatora

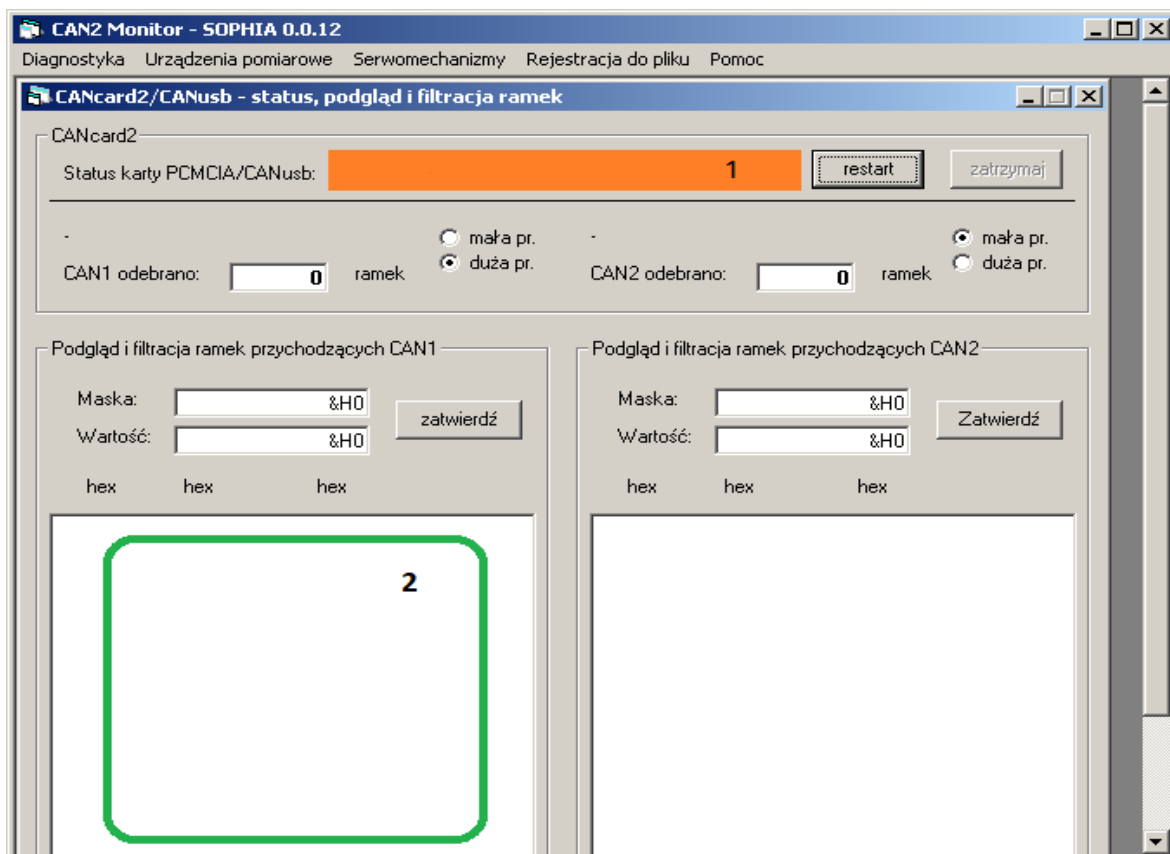
Dodatek B. Obsługa programu CAN_Monitor

Podczas ćwiczeń będziesz korzystał z oprogramowania CAN_Monitor

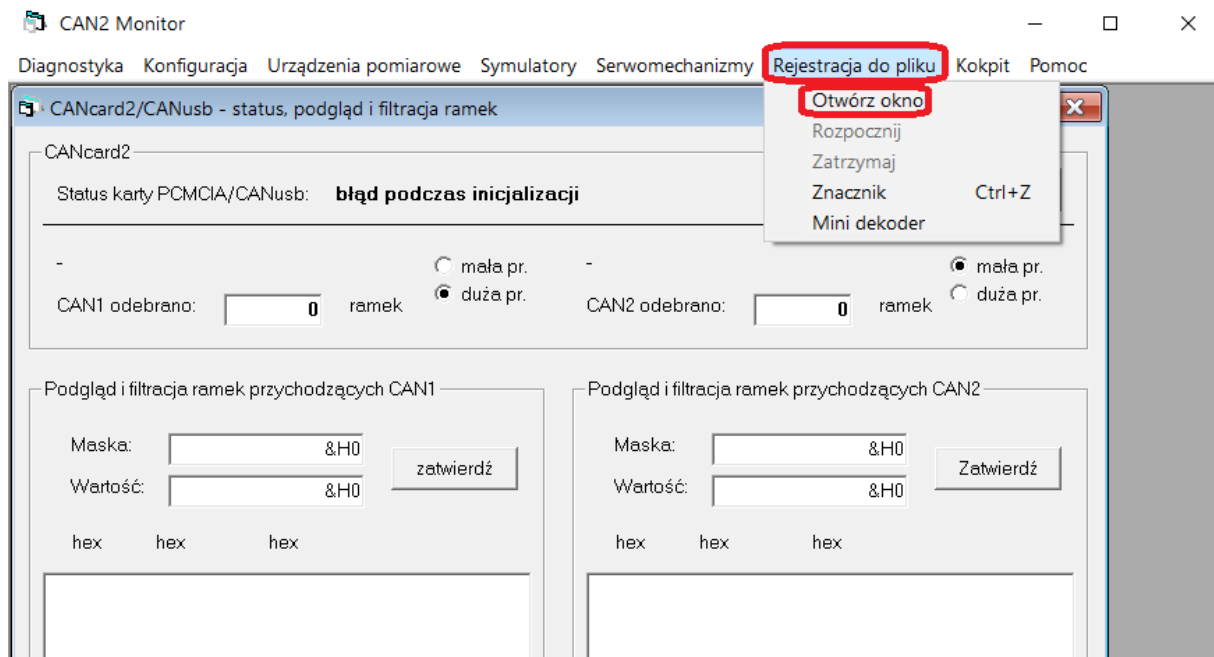
Ikona programu znajduje się na pulpicie komputera



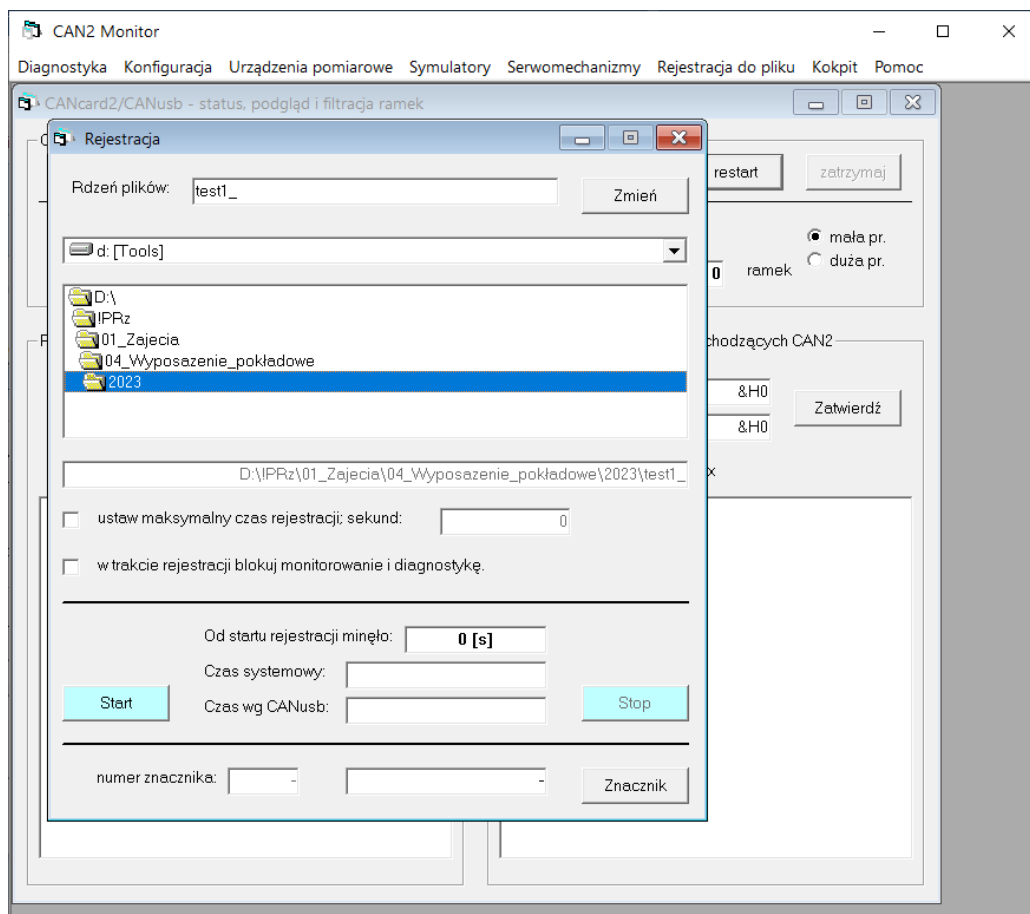
1. Uruchom program przy pomocy skrótu na pulpicie
2. Po uruchomieniu powinien być widoczny status „zainicjowano poprawnie” (pomarańczowa ramka1), oraz powinny być widoczne odbierane dane (zielona ramka2)



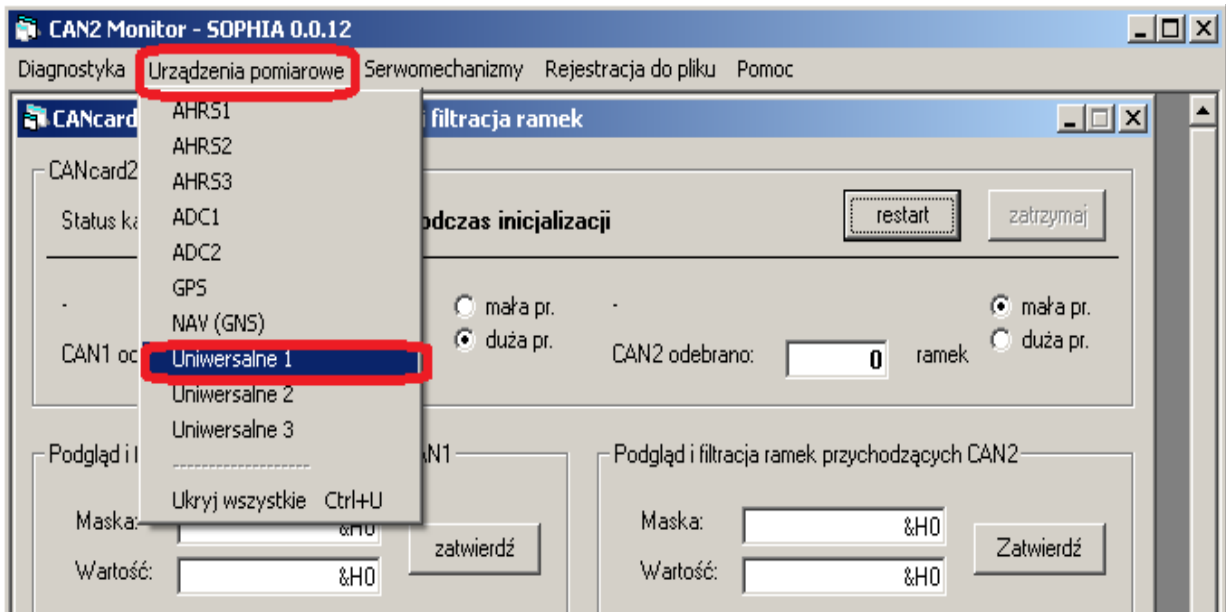
3. Wybierz zakładkę „Rejestracja do pliku” a następnie „Otwórz okno”



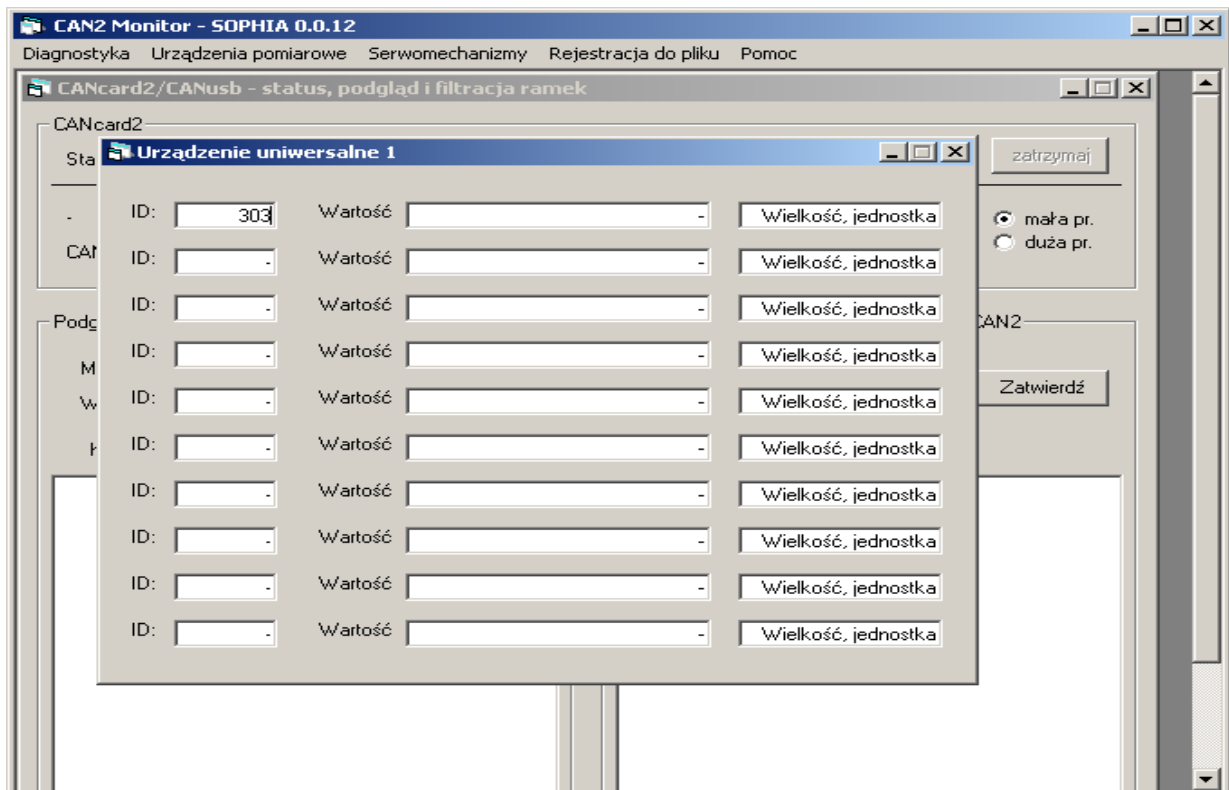
4. Wpisz nazwę rdzenia plików (nie może kończyć się cyfrą), następnie wciśnij przycisk zmień, w dalszej kolejności wybierz katalog do zapisu danych, włącz rejestrację przyciskiem „Start”.



- Wybierz zakładkę „Urządzenie pomiarowe”, a następnie „Uniwersalne 1”, w każdym oknie można wyświetlić 10 danych



- W polu po prawej stronie wpisz numery identyfikatorów zgodnie z wymaganiami przy ćwiczeniu, jeżeli dana jest przesyłana, to wyświetli się w oknie po prawej stronie



7. Numery identyfikatorów znajdziesz w pliku „student.c”

```
// Do poszczególnych ćwiczeń niezbędna będzie ci znajomość następujących zmiennych //  
// i identyfikatorów ich ramek z magistrali CAN (CAN_ID) //  
//-----//  
// CWICZENIE 1 - skalowanie czujników barometrycznych //  
// sADC.Pdiff - ciśnienie różnicowe, CAN_ID 325 //  
// sADC.Pstat - ciśnienie statyczne, CAN_ID 326 //  
// sADC.rawPdiff - wartość z czujnika ciśnienia różnicowego, CAN_ID 1304 //  
// sADC.rawPstat - wartość z czujnika ciśnienia statycznego, CAN_ID 1304 //  
//-----//  
// CWICZENIE 2 - skalowanie czujnika przyspieszenia //  
// sIMU.Ax - przyspieszenie podłużne, CAN_ID 300 //  
// sIMU.Ay - przyspieszenie poprzeczne, CAN_ID 301 //  
// sIMU.Az - przyspieszenie normalne, CAN_ID 302 //  
// sIMU.rawAx - wartość z czujnika przyspieszenia podłużnego, CAN_ID 1300 //  
// sIMU.rawAy - wartość z czujnika przyspieszenia poprzecznego, CAN_ID 1301 //  
// sIMU.rawAz - wartość z czujnika przyspieszenia normalnego, CAN_ID 1302 //  
//-----//  
// CWICZENIE 3 - skalowanie czujnika prędkości katowej //  
// sIMU.P - prędkość katowa przechylenia, CAN_ID 304 //  
// sIMU.Q - prędkość katowa pochylenia, CAN_ID 303 //  
// sIMU.R - prędkość katowa odchylenia, CAN_ID 305 //  
// sIMU.rawP - wartość z czujnika prędkości katowej przechylenia, CAN_ID 1304 //  
// sIMU.rawQ - wartość z czujnika prędkości katowej pochylenia, CAN_ID 1303 //  
// sIMU.rawR - wartość z czujnika prędkości katowej odchylenia, CAN_ID 1305 //  
//-----//
```

8. Zarejestrowane dane rozkoduj przy pomocy programu Dekoder_can, wybierz „Rozkodowanie indywidualne”(1), następnie wpisz wybrane identyfikatory (2) i wciśnij przycisk „Rozkoduj” (3).

Rozkoduj CANas

UWAGA! Wszystkie opcje wyboru dotyczą rozkodowania standardowego.
Rozkodowanie indywidualne posiada swoje własne okno.

AHRS
 AHRS1
 AHRS2
 AHRS3

ADC
 ADC1
 ADC2

NAV/GPS
 NAV
 GPS

CON
 CON1
 CON2

ZN
 ZN

KS
 KS1
 KS2
 KS3

Zapis identyfikatorów nierozkodowanych

Rozkodowanie standardowe

1
Rozkodowanie indywidualne

Rozkodowanie indywidualnego zestawu danych

1300	<input checked="" type="checkbox"/> ID	0	<input type="checkbox"/> ID 11
1301	<input checked="" type="checkbox"/> ID	0	<input type="checkbox"/> ID 12
1302	<input checked="" type="checkbox"/> ID	0	<input type="checkbox"/> ID 13
300	<input checked="" type="checkbox"/> ID	0	<input type="checkbox"/> ID 14
301	<input checked="" type="checkbox"/> ID	0	<input type="checkbox"/> ID 15
302	<input checked="" type="checkbox"/> ID	0	<input type="checkbox"/> ID 16
0	<input type="checkbox"/> ID 7	0	<input type="checkbox"/> ID 17
0	<input type="checkbox"/> ID 8	0	<input type="checkbox"/> ID 18
0	<input type="checkbox"/> ID 9	0	<input type="checkbox"/> ID 19
0	<input type="checkbox"/> ID 10	0	<input type="checkbox"/> ID 20

Liczba ramek do rozkodowania: 6

Rozkodowanie "rozrzutne"
 Rozkodowanie bajtów

Czas początkowy [s]: 0 Czas końcowy [s]: 36000

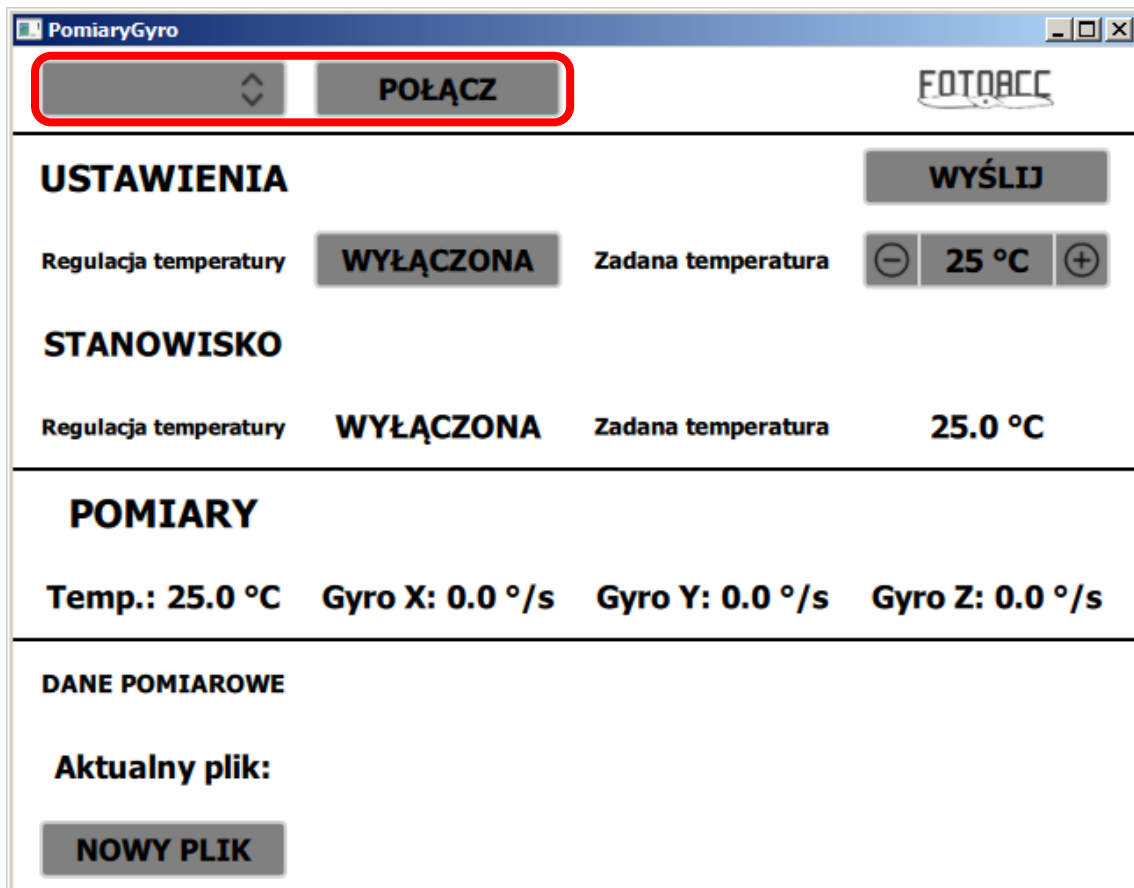
Zapisz ID Wczytaj ID Rozkoduj

Dodatek C. Obsługa programu PomiaryGyro

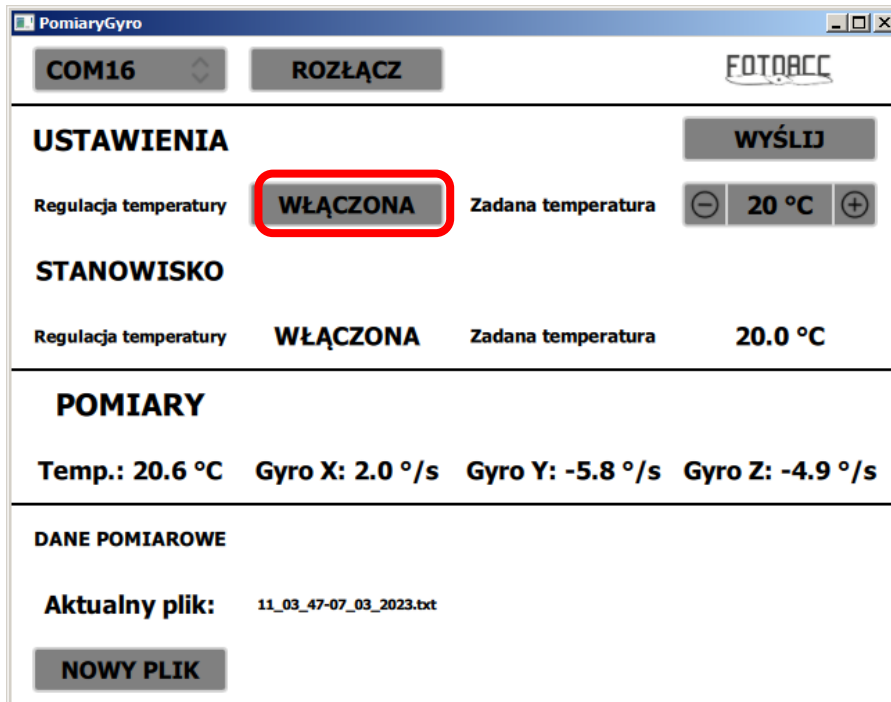
Podczas ćwiczeń będziesz korzystał z oprogramowania PomiaryGyro

Ikona programu znajduje się na pulpicie komputera

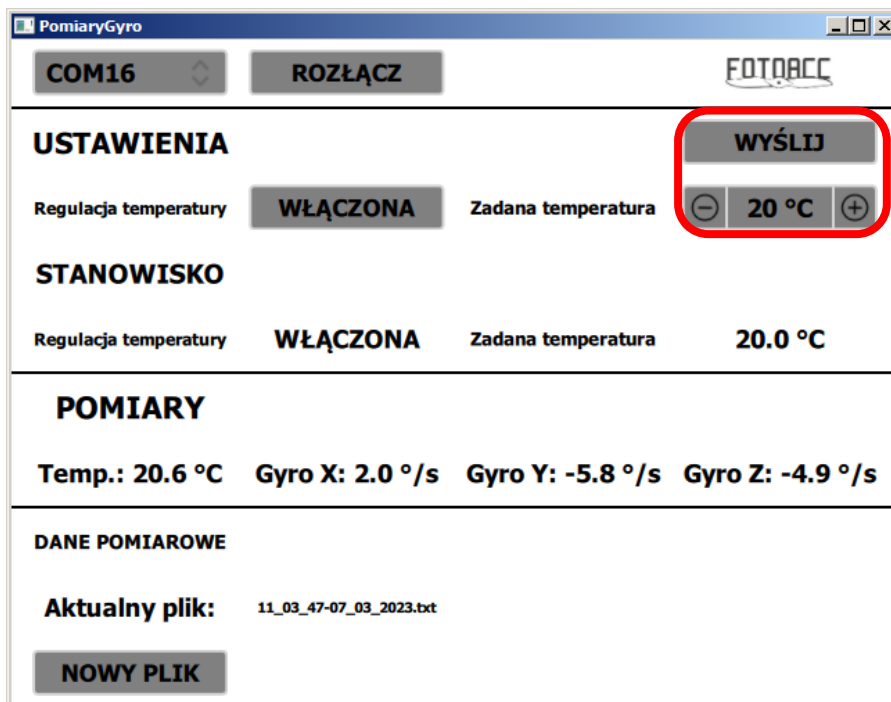
1. Uruchom program przy pomocy skrótu na pulpicie
2. Wybierz port COM i wśnij przycisk POŁĄCZ.



3. Przełącz na Regulację temperatury na WŁĄCZONA



4. Ustaw zadaną temperaturę i wciśnij przycisk WYŚLIJ



5. Obserwuj temperaturę (1), po jej ustaleniu odczytaj wartości prędkości kątowych (2).

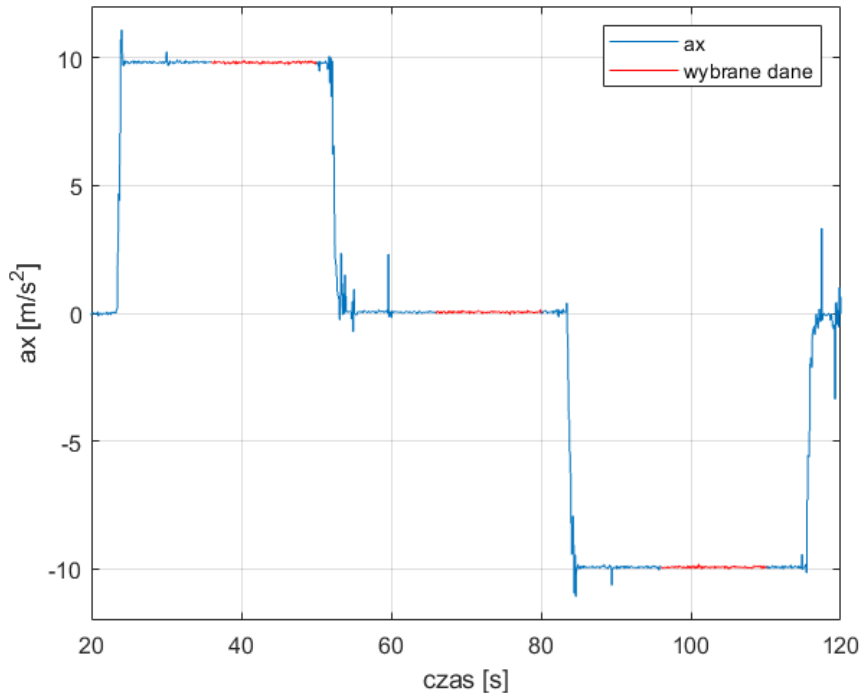
The screenshot shows the 'PomiaryGyro' software window. At the top, there are controls for 'COM16' and 'ROZŁĄCZ', and the 'FOTOBACC' logo. Below this is the 'USTAWIENIA' (Settings) section with a 'WYŚLIJ' button. The temperature regulation is set to 'WŁĄCZONA' (ON) with a target temperature of '20 °C'. The 'STANOWISKO' (Station) section also shows 'WŁĄCZONA' and '20.0 °C'. The '1 POMIARY' (Measurements) section displays: 'Temp.: 20.6 °C' (circled in red), 'Gyro X: 2.0 °/s', 'Gyro Y: -5.8 °/s', and 'Gyro Z: -4.9 °/s' (all circled in red). The 'DANE POMIAROWE' (Measurement Data) section shows 'Aktualny plik: 11_03_47-07_03_2023.txt' and a 'NOWY PLIK' button. A red '2' is placed to the right of the measurement data.

6. Zarejestruj plik w katalogu programu.

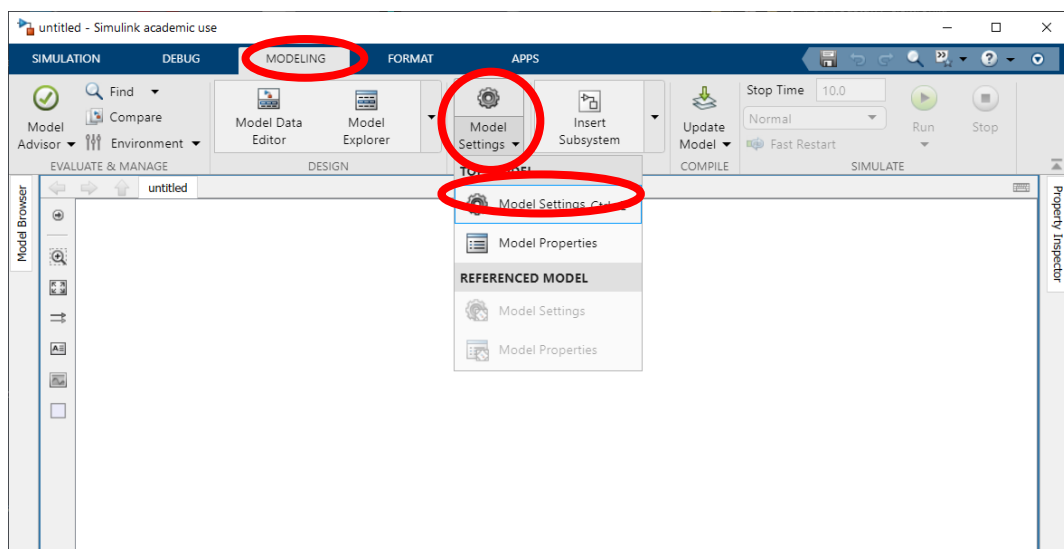
This screenshot is similar to the previous one, but the '1 POMIARY' section is now titled 'POMIARY' and the measurement values are displayed below the settings. The 'Aktualny plik:' field in the 'DANE POMIAROWE' section now contains the filename '11_03_47-07_03_2023.txt', which is circled in red. The red '2' is no longer present.

Dodatek D. Modelowanie czujników

1. Zimportuj dane z zarejestrowanego pliku do arkusza kalkulacyjnego, wybierz wyskalowane dane do przetwarzania, dla przyspieszeń (lab 2) po 15 sekund, dla prędkości kątowych (lab 3) po 10 sekund (patrz przykład).

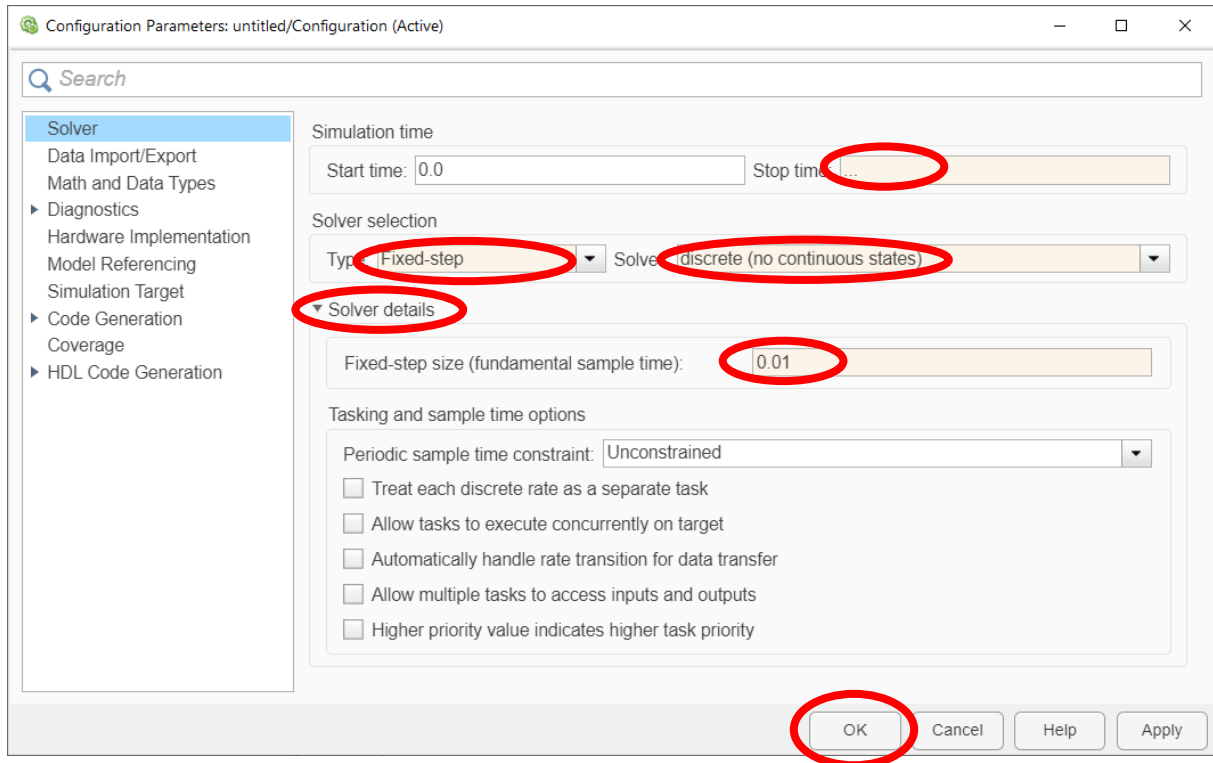


2. Dla wybranych zakresów wyznacz wartości średnie \bar{x} i wartości błędów $\Delta x = \bar{x} - x$ (wartość średnia – wartość oczekiwana)
3. Wyznacz wartości wariancji lub odchylenia standardowego $w = \sigma^2$
 - a. W przypadku przyspieszeń dla $a_x = 0$, $a_y = 0$ i $a_z = 1g$
 - b. W przypadku prędkości kątowej dla wybranej prędkości z zakresu $\langle -10 \frac{m}{s^2}, 10 \frac{m}{s^2} \rangle$
4. Uruchom MATLAB, następnie SIMULINK, a następnie utwórz nowy model. Otwórz okno parametryzacji (CTRL + E lub patrz na rysunek).

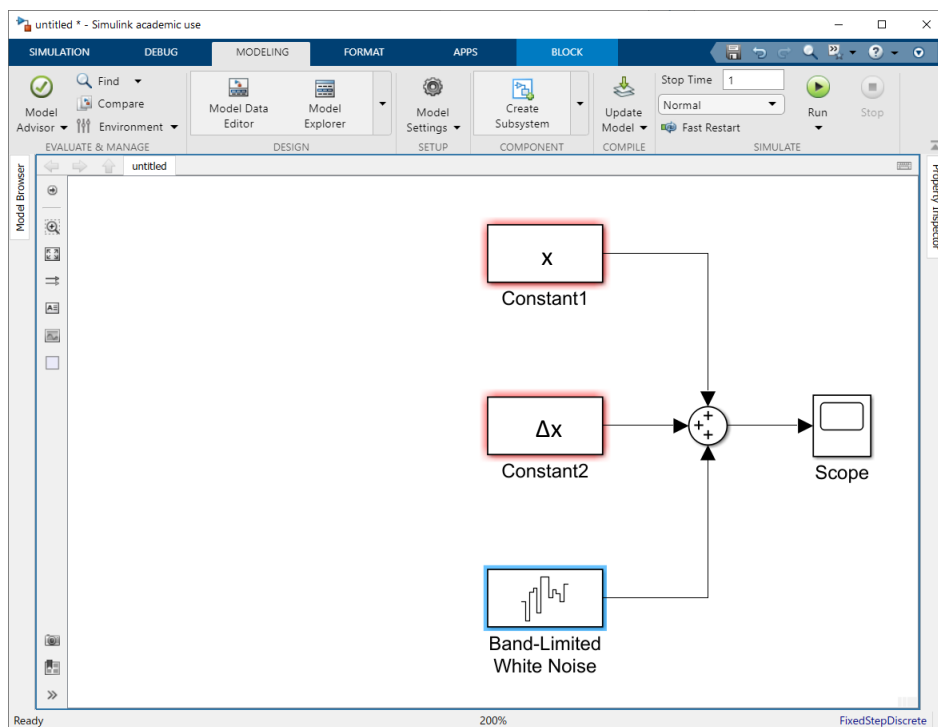


5. Ustaw następujące parametry (patrz rysunek):

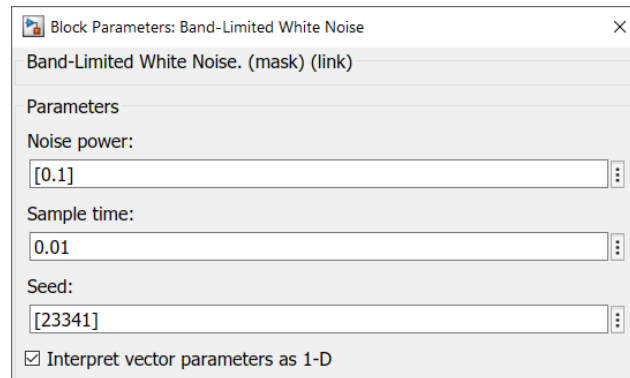
- a. Type: Fixed-step
- b. Solver: discrete (no continuous state)
- c. Rozwiń Solver details
- d. Fixed-step size: 0.01 s
- e. Stop time: 1 s



6. Zbuduj model zgodnie z poniższym rysunkiem



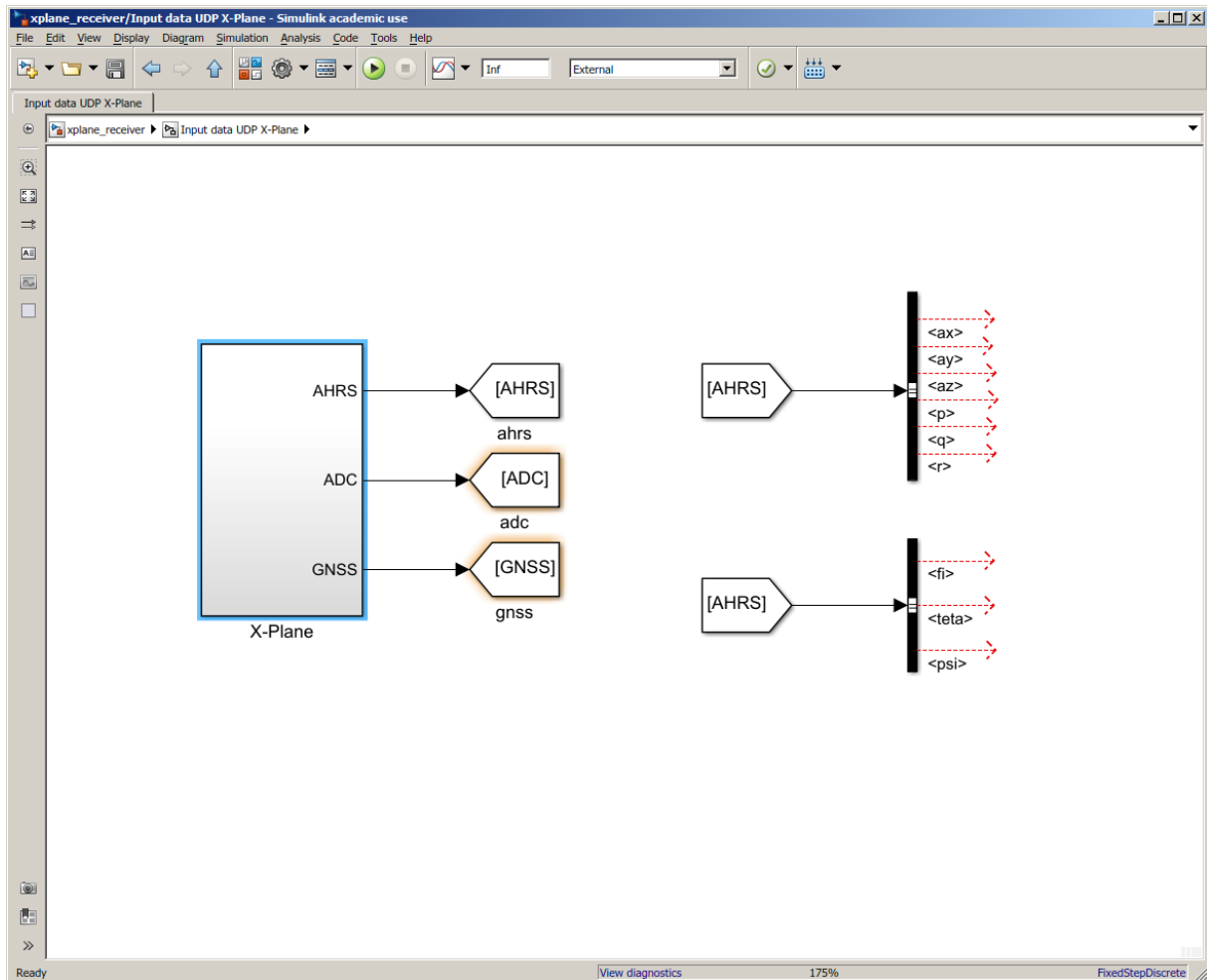
7. Uzupełnij parametry w poszczególnych elementach
 - a. Constant1 – wartość oczekiwana x
 - b. Constant2 – błąd $\Delta x = \bar{x} - x$ (wartość średnia – wartość oczekiwana)
 - c. ...White Noise – patrz pkt 8.
8. Otwórz i uzupełnij parametry bloku Band-Limited White Noise
 - a. Noise power: wt – iloczyn wariancji i kroku symulacji (fixed-step size:0.01 s),
 - b. Sample time: 0.01 s,
 - c. Seed: pierwszy model - suma numerów indeksów członków zespołu, każdy następny model – zwiększ wartość o 1.



9. Wykonaj symulację, zapisz wyniki w postaci wykresu (0-1s)
10. Porównaj z wynikami (1s) z wybranego zakresu z pkt 1. Przedstaw wyniki w sprawozdaniu (na jednej stronie, 2 wykresy po 1s, pierwszy z danymi rzeczywistymi, drugi z danymi z symulacji, wykresy wyskaluj tak, aby początek oraz czas 1 s były w podobnym położeniu i podobnej długości).

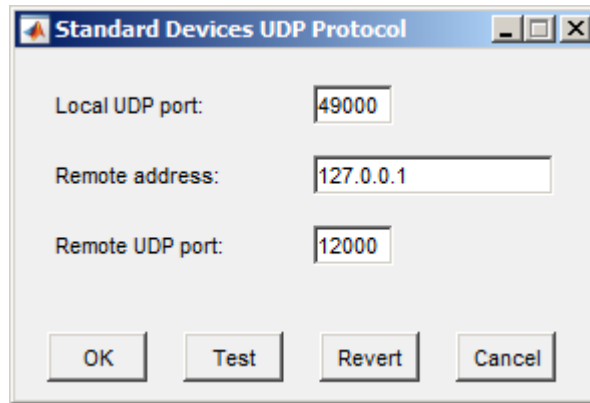
Dodatek E. Instrukcja do Matlab/Simulink Desktop Real Time

1. Zainstaluj Matlab/Simulink w wersji co najmniej 17a, zainstaluj aktualizacje (ważne)
2. Zainstaluj kernel Desktop Real-Time, polecenie: `sldrtkernel -install`
Instrukcja:
<https://www.mathworks.com/help/sldrt/ug/real-time-windows-target-kernel.html>
3. Otwórz model: `xplane_receiver_ver2017a.slx`

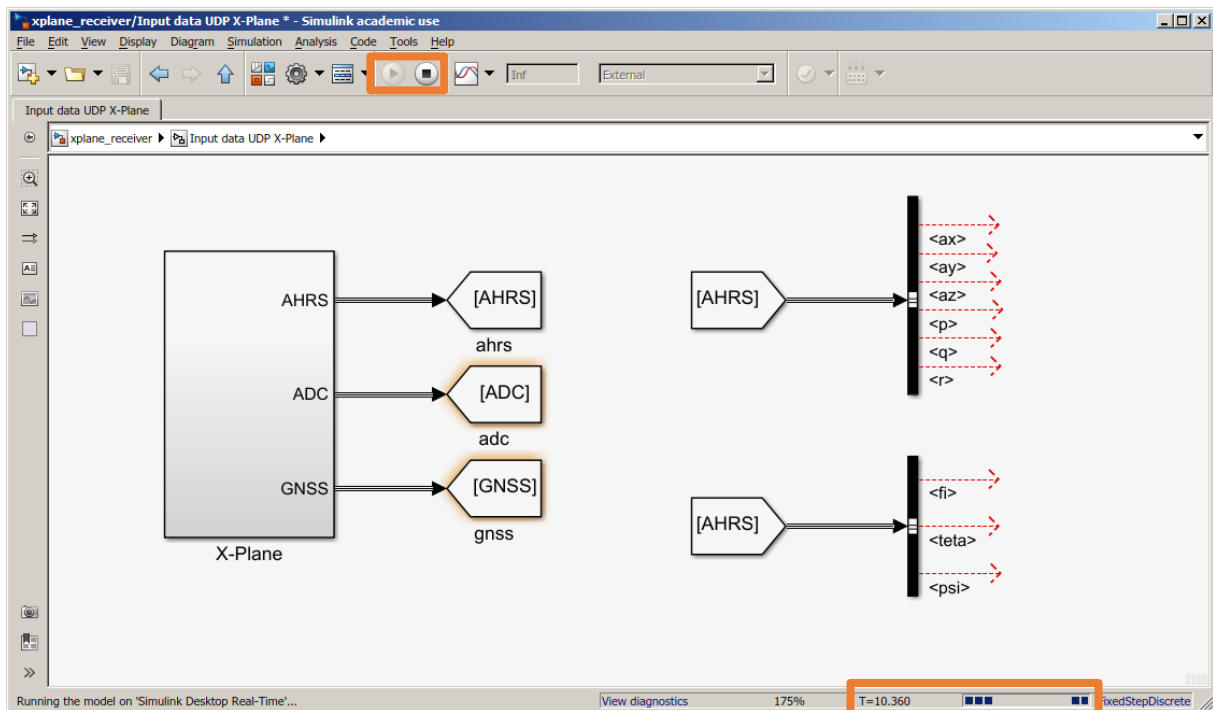


4. Sprawdź nr IP swojego komputera (karty bezprzewodowej), powinien być w tej samej puli adresów co komputer X-Plane

- Do testów ustaw „Remote address” na „127.0.0.1” i „Local UDP Port” na „49000”. Na zajęciach spytaj prowadzącego o numer „Remote address”



- Wydź na główną stronę modelu
- Uruchom symulację przyciskiem RUN, w poprawnie działającej symulacji będzie się zwiększał czas (na dolnym pasku)



- Gratulacje, twój komputer jest gotowy do zajęć laboratorium Wyposażenia pokładowego.

