



RoboCup (Tr)

Bilgisayarda Matlab runtime 9.7 2019b, Clojure, leiningen, emacs ve xctu xp software olması gerekir.

- 1) Github'tan neu-islanders'ı clonela.
- 2) Anteni bağla.
- 3) Terminali aç ve neu-islanders dosyasında make yap.
- 4) Eğer problem çıkarsa (pid-file'dan dolayı) make kill yap.
- 5) Veya pid-file'ı silmen gerekecek 'rm -rf pid-file'.
- 6) Sorunu çözünce make yap, robot ID'sini ve başka bir şeyleri gösteren bir pencere açılacak.

Emacs tarafı:

- 1) Emacs'i aç ve real-world programının bulunduğu dosyaya giderek real-world'ü aç.
- 2) kısayollar: ctrl + x + f ile dosyaları arayabilirsin, ctrl + x + 3 ekranı ikiye böler, ctrl + y yapıştır, ctrl + Alt + w kopyala, ctrl + x + r + l yaparak user=> kısmına geçiş yapabilirsin.
- 3) Ctrl + x + r + l (r ve l'yi iki tuştan sonra yap)yaparak user=>'a git ve oraya arada bir boşluk ile (in-ns 'neu-islanders.command.real-world) yaz.
- 4) Daha sonra buradan komutları test edebilirsin. Örnek: (motor 4 10) id'si 4 olan robotun motorlarına 10 birim güç verir ve çalıştırır. (motor 4 [10,0,0,0]) ile id'si 4 olan robotun sadece 1. motoruna 10 birim güç vererek sadece onu çalıştırır (eksi güç ters yönde hareketi belirtir). kick ve chip-kick fonksiyonları da aynı şekilde (kick 4 100) veya (chip-kick 4 30) test edilebilir.

1. parametre: fonksiyon adı

2. parametre: robot id

3. parametre: güç seviyesi

- Xctu'da channel'ı kontrol et 4'e 4 1 grup 10 diğeri 17 olması gerek. Baud rate de 57600 olması gerekir.

Testler ve değişiklikler tamamlandığında program düzgünce kapatılmalıdır.

Robocup Test 1	Tarih: 20/04/21
Robot ID	Test Sonucu
Home 0	Tüm fonksiyonlar çalışır
Home 1	Tüm fonksiyonlar çalışır
Home 2	Tüm fonksiyonlar çalışır
Home 4	Tüm fonksiyonlar çalışır
Home 6	Tüm fonksiyonlar çalışır
Home 8	Chip-kick'te problem var

RoboCup (En)

Computer must have Matlab runtime 9.7 2019b, Clojure, Leiningen, emacs and xctu xp software.

- 1) Clone the neu-islanders project from github.
- 2) Connect the Antenna.
- 3) Open the terminal and do 'make' in neu-islander folder.
- 4) If there is a problem (due to pid-file) do 'make kill'.
- 5) Or you can delete the pid-file using 'rm -rf pid-file'.
- 6) After you resolve the issues, do 'make' there should be a window showing the robot ID and other stuff.

Emacs side:

- 1) Open Emacs and the folder where the real-world program is, then open real-world.
- 2) shortcuts: ctrl + x + f to search the folders/files, ctrl + x + 3 divides the screen into 2, ctrl + y paste, ctrl + Alt + w copy, ctrl + x + r + l to go to user=> section.
- 3) Ctrl + x + r + l (r and l are after first two keys) by doing this you can go to user=>, then make sure there is a gap and write this (in-ns 'neu-islanders.command.real-world)
- 4) Then you can test the commands from here. E.g.: (motor 4 10) sends 10 units of power to the robot with id = 4 and motors start working. (motor 4 [10,0,0,0]) only sends 10 units of power to the first motor of the robot with id = 4 and only the first motor works (minus power indicates the opposite direction of the motion). kick and chip-kick functions work with the same principle (kick 4 100) or (chip-kick 4 30) can be tested.

1. parameter: function name

2. parameter: robot id

3. parameter: power level

- Check the channels from Xctu, there are 2 teams of 4 and 1 group should have channel id 10 and the other should have channel id 17. Baud rate has to be 57600.

After tests and modifications are done, shut the program cleanly.

Robocup Test 1	Date: 20/04/21
<i>Robot ID</i>	<i>Test Result</i>
Home 0	All functions work
Home 1	All functions work
Home 2	All functions work
Home 4	All functions work
Home 6	All functions work
Home 8	Problem with chip-kick

UAV (Tr)

Bilgisayarda Arduino, ve Arduino'nun içinde Teensy 3.2, 9DoF razor IMU kurulu olması gerekir.

- 1) Github'dan vecihi projesini clone'la.
- 2) Arduino'dan > preferences > external editor'ü işaretle.
- 3) İstedğin external editor ile programı düzenleyebilirsin.
- 4) Daha sonra değişiklikleri uçağın teensy board'una kablo ile aktarman gerekir. (hdg ve stb modları şimdilik manuel olarak değiştiriliyor.)

Uçuş ve Test:

Python3 yüklü olması gerek, daha sonra pip install ile ekstra kütüphaneleri yükle. Powershell kullanarak pip install yapılabilir. İndirilecek kütüphaneler: invoke, pyserial, matplotlib, PyQt5 (örnek: pip install invoke).

(setup.py kullanarak da kurulum yapılabilir.)

Uçuşta ve test esnasında değişkenleri takip etmek için uygulanması gereken adımlar:

- 1) Anten bilgisayara bağlanır ve hangi port olduğu control edilir.
- 2) Powershell açıp oradan vecihi dosyasına gir (clone'ladığın yere).
- 3) Dosyanın içinde invoke capture --port COMX (X yerine sayı gelecek).
- 4) Programın UI tarafı açılacak ve buradan değerlerde değişiklikler yapılabilir.

Not: UI tarafı her açıldığında anlık değerler yeni bir .json dosyasına aktarılmaya başlar ve kapatıldığında ise .json dosyası kaydedilir. Eğer bir uçuş boyunca değerleri almak istiyorsanız UI'ı uçuş başlangıcında açıp bittiğinde kapamayı unutmayın. Her dosya otomatik olarak gün ve saat ismiyle kaydedilir. Bu dosyalar vechini klasöründe bulunabilir.

Load: Uçaktan anlık değerleri alır.

Save: Yazdığımız veya load'dan aldığımız değerleri uçağa geri gönderir.

Off-set: Eğer uçağın kalibrasyonu bozulursa su terazisiyle ayarlar yapıldıktan sonra tekrardan kalibre etmek için bu tuşa basılır. Onun dışında kullanılmaz.

UAV (En)

Make sure there is Arduino with Teensy 3.2 and 9DoF razor IMU on your computer.

- 1) Clone the vecihi project from Github.
- 2) From Arduino > preferences > external editor, tick the box for external editor.
- 3) You can use any external editor to edit the code.
- 4) If there are any changes, you have to connect the micro usb cable to the teensy, and send them to the plane. (hdg and stb can be changes manually right now.)

Flight and Test:

You have to have Python3, then use pip install to get the extra libraries. You can use the Powershell for pip installations. Libraries to download: invoke, pyserial, matplotlib, PyQt5 (for example: pip install invoke).

(setup.py can be used to add the libraries.)

During the flight and test, apply these steps to follow the changes in variables:

- 1) Connect the antenna to the computer and check the port.
- 2) Open powershell, go to vecihi folder (wherever you cloned it).
- 3) In that folder do: invoke capture --port COMX (X will be a number).
- 4) The UI side of the software will open and you can modify the variables from here.

Note: Whenever the UI window is opened, instant values are written in a .json file and when you shut the window the .json file is saved. If you want to get the values during a flight, don't forget to open the UI window at the start of the flight and shut it at the end of it. Each file is saved with the date and time as file name. The files can be found in vecihi folder.

Load: Takes the instant values of the variables from the plane.

Save: Sends the values we modified or loaded, back to the plane.

Off-set: If there is a calibration issue with the plane, everything has to be set up again using the spirit level after that, click on this button to re-calibrate the plane. Don't use it otherwise.

Temizlik Robotu (Tr)

Bilgisayarda python3, ROS (noetic) ve WeBots olması gerekir. (linux)

Simülasyonu çalıştırmak için:

- 1) Github'tan cleaning-machine projesini clone'la.
- 2) Terminalden ros'u aç: \$roscore
- 3) Daha sonra yeni terminal tab'ından cleaning-machine > build folder'ına gir.
- 4) Build folder'ında \$make sim komutunu yaz.
- 5) Sonra cleaning machine ana folder'ına geri dön ve ./launch sim komutunu yaz. (Eğer webots açılmaz diye bir sorun olursa “sudo apt-get install ros-noetic-webots-ros” komutunu gir ve tekrar dene.)
- 6) Simülasyon açıldıktan sonra manuel olarak navigasyon yapmak için ./launch man komutunu girmek gerekir. (Navigasyon için açılan pencerenin seçili olduğu zamanlarda w,a,s,d tuşları ile hareket sağlanabilir.)
- 7) Otomatik Slam navigasyonu için ./launch nav komutu kullanılır, şu an için manueldir.

ROS komutları hakkında bilgi için, [ROS Tutorials](#) linkini kullanabilirsiniz.

Webots için [Webots Tutorials](#) linkini kullanabilirsiniz.

Cleaning Machine (En)

Computer must have python3, ROS (noetic) and WeBots. (linux)

Starting the simulation:

- 1) Clone the cleaning-machine project from Github.
- 2) From terminal open ros: \$roscore
- 3) Then open new terminal tab go to cleaning-machine > build folder.
- 4) In build folder, enter this command: \$make sim
- 5) Go back to the main folder cleaning-machine, and write command: ./launch sim (If webots doesn't open and there is an error in the terminal due to this, insert this command and try again: sudo apt-get install ros-noetic-webots-ros)
- 6) After opening the simulation, to navigate manually insert this command: ./launch man (When the navigation window opens, you can use w,a,s,d keys to move.)
- 7) For automatic slam navigation, write ./launch nav command, it is manuel right now.

To get more information about ROS commands go to [ROS Tutorials](#).

For webots, you can use [Webots Tutorials](#).