

UNIVERSITATEA POLITEHNICA TIMIȘOARA  
FACULTATEA DE MECANICĂ  
DEPARTAMENTUL DE MECATRONICĂ  
SPECIALIZAREA ROBOTICĂ

# **Robot Mobil Comandat De La Distanță Prin Raspberry Pi, Cu Redare Video**

Coordonator științific:

Șl.Dr.Ing. Cristian Moldovan

Absolvent:

Andreas Bissinger

TIMIȘOARA

2017

## CUPRINS

1. Introducere .....	3
1.1. Mecatronică și Robotică .....	3
1.1.1. Mecatronică .....	3
1.1.2. Robotică .....	4
1.1.2.1. Robotica Mobilă .....	5
1.2. Stadiul actual și realizările importante ale sistemelor robotice comandate de la distanță .....	7
1.2.1. Vehiculul spațial Spirit .....	7
1.2.2. Sistemele de avioane fără pilot .....	8
1.2.3. iRobot .....	10
1.2.4. Robotul ASIMO .....	12
2. Aplicabilitate .....	13
2.1. Aplicații ale sistemelor robotice comandate de la distanță .....	13
2.2. Flexibilitate .....	15
2.3. Viitoare îmbunătățiri .....	15
3. Alte variante de proiectare .....	16
3.1. Robot mobil cu Arduino, comandat din telefonul mobil folosind Bluetooth .....	16
3.1.1. Bluetooth .....	17
3.2. Robot mobil, comandat prin telecomandă IR .....	18
3.2.1. Infraroșu .....	18
3.3. Varianta aleasă – Robot Mobil Comandat De La Distanță Prin Raspberry Pi, Cu Redare Video .....	20
3.3.1. Wi-Fi .....	20
3.4. Diferențe între variantele de proiectare .....	22
4. Construcția robotului .....	23
4.1. Elemente componente, programe și limbaje de programare utilizate .....	23
4.1.1. Raspberry Pi 3 – Model B .....	23
4.1.1.1. Generalități .....	23
4.1.1.2. Instalarea sistemului de operare .....	25
4.1.1.3. Utilizarea plăcii Raspberry Pi 3 de la distanță .....	27
4.1.2. Camera Video Raspberry Pi .....	29
4.1.3. Kit Robot + Motoare Electrice .....	30
4.1.4. ULN2803 .....	32
4.1.5. Python .....	33
4.1.5.1. Generalități .....	33
4.1.5.2. Stil .....	34
4.1.5.3. Biblioteci și Extindere .....	34
4.1.6. HTML .....	35
4.2. Construcția propriu-zisă a robotului .....	37
5. Funcționare .....	43
6. Concluzii .....	47

## SCOPUL LUCRĂRII

Lucrarea are ca scop dezvoltarea unui robot mobil comandat de la distanță, capabil să se deplaseze în medii necunoscute sau periculoase pentru om, în timp ce redă imagini în timp real. Robotul va fi comandat printr-o aplicație web și printr-un modul Wi-Fi, integrat într-o placă Raspberry Pi, iar pentru deplasarea acestuia se vor folosi 4 motoare de curent continuu.

# 1. INTRODUCERE

## 1.1. Mecatronică și Robotică

### 1.1.1. Mecatronică

Termenul "mecatronică" a fost utilizat pentru prima dată în anul 1975 de către concernul japonez Yaskawa Electric Corporation, fiind o prescurtare a cuvintelor Mecanică-Electronică-Informatică.

La început, mecatronică a fost înțeleasă ca o completare a componentelor mecanicii de precizie, aparatul de fotografiat cu bliț fiind un exemplu clasic de aplicație mecatronică.

Cu timpul, noțiunea de mecatronică și-a schimbat sensul și și-a extins aria de definiție: mecatronica a devenit știința inginerescă bazată pe disciplinele clasice ale construcției de mașini, electrotehnicii, electronicii și informaticii. Scopul acestei științe este îmbunătățirea funcționalității utilajelor și sistemelor tehnice prin unirea disciplinelor componente într-un tot unitar.

Totuși, mecatronica nu este același lucru cu automatica sau cu automatizarea producției. Aceștia sunt termeni care apar și în afara domeniului mecatronicii, dar sunt și incluși în el. Mecatronica poate fi definită ca o concepție inovatoare a tehnicii de automatizare pentru nevoile ingineriei și educației.

Mecatronica s-a născut ca tehnologie și a devenit filosofie care s-a răspândit în întreaga lume. În ultimii ani, mecatronica este definită simplu: știința mașinilor inteligente.

Ca o concluzie, se poate spune, că mecatronica este o sferă interdisciplinară a științei și tehnicii care se ocupă în general de problemele mecanicii, electronicii și informaticii. Totuși, în ea sunt incluse mai multe domenii, care formează baza mecatronicii și care acoperă multe discipline cunoscute, cum ar fi: electrotehnica, energetica, tehnica de cifrare, tehnica microprocesării informației, tehnica reglării și altele.

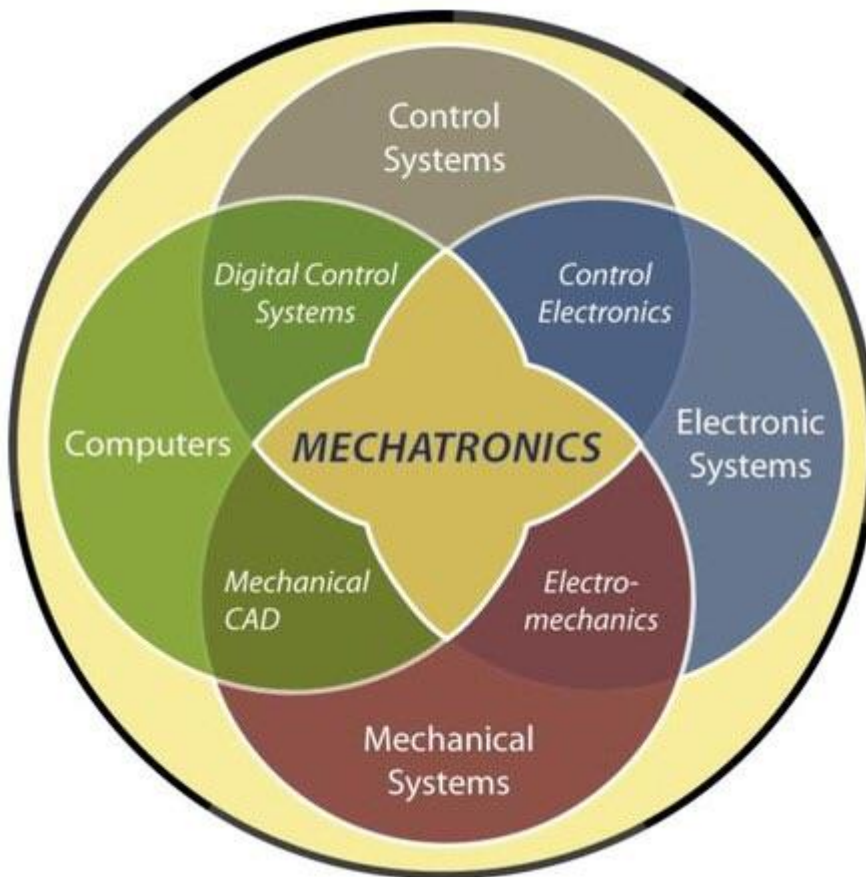


Fig. 1.1.1.1. Mecatronică

### 1.1.2. Robotică

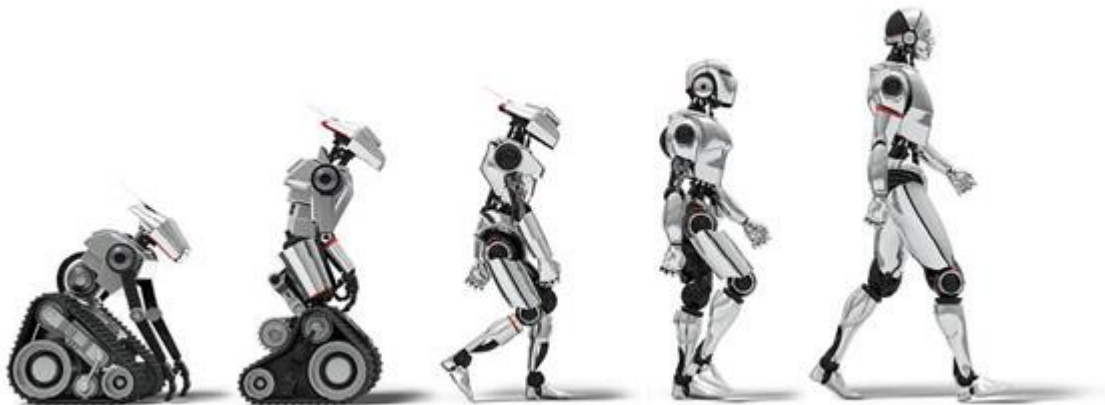
Robotica este știința care se ocupă cu tehnologia, proiectarea și fabricarea roboților. Robotica necesită cunoștințe de electronică, mecanică și programare, iar persoana care lucrează în acest domeniu a ajuns să fie cunoscută ca robotician sau inginer în robotică. Denumirea de robot a fost introdusă pentru prima oară de către Karel Čapek în anul 1921 în lucrarea sa "Roboții universali ai lui Rossum", în 1921 plecând de la cuvântul ROBOTA, muncă, activitate de rutină, preluat de către Isaac Asimov, în povestirea științifico-fantastică "Fuga în cerc" (1941). Robotii sunt mecanisme care îndeplinesc diferite sarcini, singure.

Roboții sunt realizați mai ales prin combinația disciplinelor: mecanică, electrotehnică și informatică. Între timp s-a creat din legătura acestora mecatronica. Pentru realizarea de sisteme autonome (care să găsească singure soluții) este necesară legătura a cât mai multor discipline de robotică. Aici se pune accent pe legătura conceptelor de inteligență artificială sau neuroinformatică (parte a informaticii) precum și idealul lor biologic biocibernetică (parte a biologiei). Din legătura între biologie și tehnică s-a dezvoltat bionica.

Cele mai importante componente ale roboților sunt senzorii, care permit mobilitatea acestora în mediu și o dirijare cât mai precisă. Un robot nu trebuie neapărat să poată să acționeze autonom, fapt pentru care se distinge între roboții autonomi și cei teleghidați.

Tipuri de roboți:

- Robot mobil
- Robot autonom
- Robot umanoid
- Robot industrial
- Robot de servicii
- Robot jucărie
- Robot explorator
- Robot pășitor
- Robot militar



*Fig. 1.1.2.1. Robotică*

### **1.1.2.1. Robotica Mobilă**

Robotica mobilă este o subcategorie a roboticii, care se axează pe realizarea, implementarea și îmbunătățirea roboților mobili. Spre deosebire de roboții normali, roboții mobili se pot deplasa prin diverse mijloace.

Primii roboți mobili au fost Elmer (1948) și Elsie (1949), construiți de William Grey Walter, roboții fiind capabili să caute sursa de lumină și să evite obstacole.

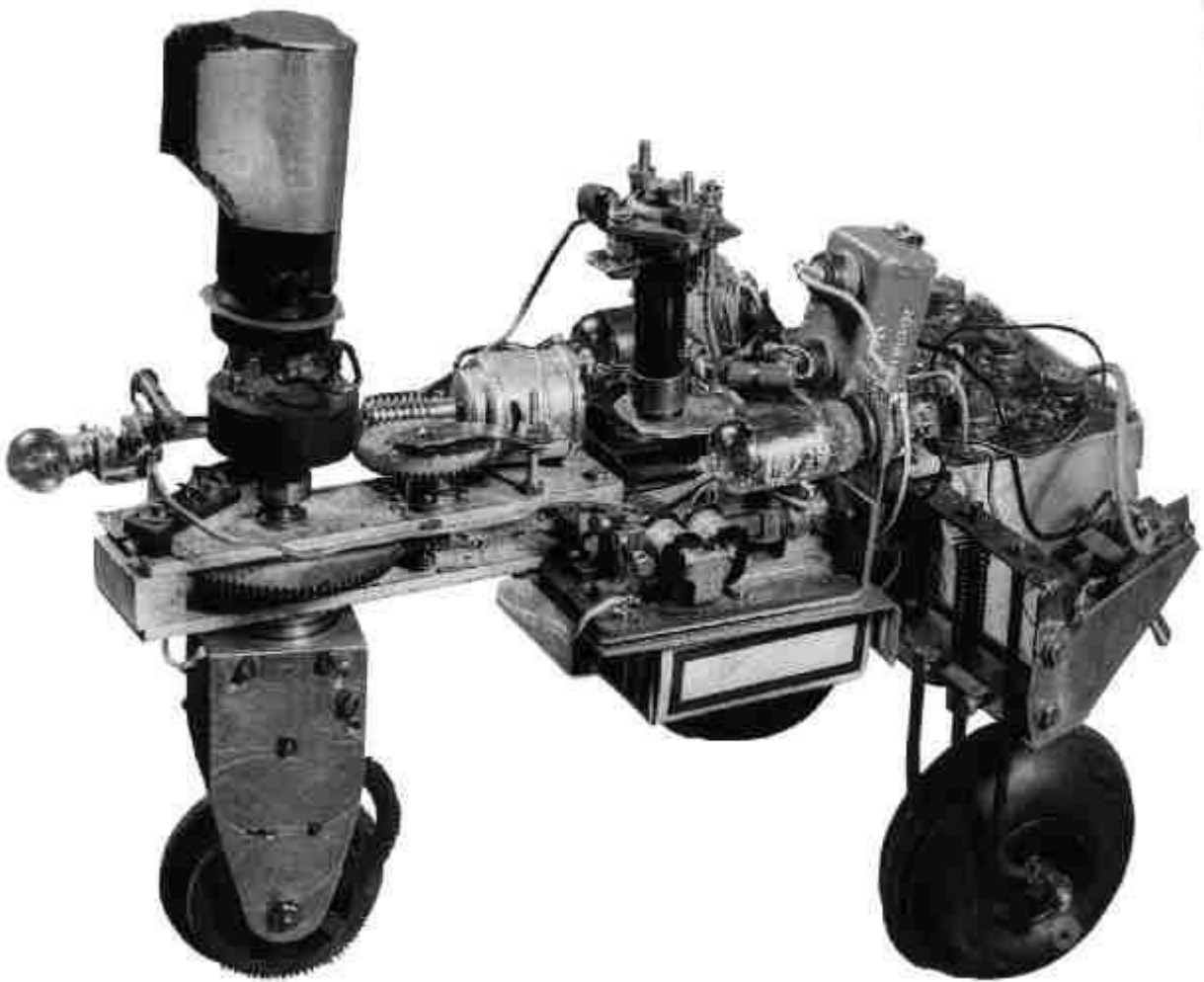


Fig. 1.1.2.1.1. Robotul Elsie

În 1951 Grey Walter prezintă la un festival din Marea Britanie un robot țestoasă, triciclu. Tot în 1951 Edmund Berkeley inventează un robot mic, “veverița”, capabil să adune nuci sau mingi de golf. Era primul robot cu un braț aflat sub control automat, primul robot automat care putea efectua o sarcină, alta decât a se îndrepta spre lumină.

Odată cu războiul rece, progresul tehnologic atinge limite nemaîntâlnite până atunci. Transpuse în lumea roboticii mobile, noile descoperiri aveau să dea lumii roboți ce foloseau ultrasunete pentru detecția obiectelor (1980), roboți cu mișcări mult mai precise (Zero-2, 1984) sau chiar roboți cu 6 picioare, capabili să susțină un om și având suspensii reglabile (Adaptive Suspension Vehicle, 1986).

În 1988 apare Shadow Walker, un robot humanoid, bazat pe mersul cu ajutorul unor motoare pneumatice.

În 1996 apare P2 de la Honda, primul robot cu autocontrol, humanoid. Putea merge pe cele două picioare, având 1,80 m și 210 kg, folosea tehnologie wireless, putea chiar să urce și să coboare scările, putând opera independent.

## 1.2. Stadiul actual și realizările importante ale sistemelor robotice comandate de la distanță

Oamenii au căutat întotdeauna metode prin care a-și face munca mai ușoară, mai sigură sau mai eficientă. Din această cauză au apărut roboții sau sistemele robotice. De exemplu, pentru a afla mai multe despre planeta Marte, oamenii au trimis un robot denumit Spirit.

### 1.2.1. Vehiculul spațial Spirit

Vehiculul spațial Spirit, denumit oficial Mars Exploration Rover este primul dintre cele două vehicule spațiale de tip „rover” ale misiunii Mars Exploration Rover a organizației NASA. „Rover” e un cuvânt englez cu traducerea „vagabond” și care aici se referă la un automobil pentru teren accidentat. A amartizat cu succes la ora 04:35 UTC în ziua de 4 ianuarie 2004. A fost urmat de „fratele său geamăn”, Opportunity, care după trei săptămâni a amartizat pe cealaltă parte a planetei.

Spirit a absolvit cu succes misiunea sa planificată de 90 de zile marțiene. Însă curățarea naturală a panourilor sale solare de către vânt a avut ca rezultat o producție sporită de energie electrică, astfel că a continuat să funcționeze de peste douăzeci de ori mai mult decât prevăzuseră cei de la NASA. Spirit s-a deplasat 10 km pe suprafața lui Marte, față de 1 km cât se planificase inițial, reușind să studieze extensiv rocile marțiene și geografia zonei. Rezultatele științifice inițiale ale primei faze a misiunii au fost publicate într-un număr special al revistei Science.

La 1 mai 2009 (5 ani, 3 luni, 27 de zile terestre după amartizare și de 21,6 ori mai mult decât durata planificată) Spirit s-a împotmolit într-un sol moale. Evenimentul se mai întâmplase și timp de opt luni NASA a analizat situația cu grijă, rulând simulări teoretice și practice pe Pământ, după care a programat în cele din urmă „Spirit” să efectueze manevre de eliberare. Eforturile au continuat până la 26 ianuarie 2010 (6 ani și 22 de zile după amartizare; de 24,6 ori mai mult decât durata planificată), când oficialii NASA au anunțat că Spirit este blocat fără ieșire în solul moale, dar că va continua să efectueze cercetări științifice din această poziție.

În iulie 2010 echipa Spirit continua să aștepte mesaje ale vehiculului, care însă nu mai transmisesse de la 22 martie 2010.



Fig. 1.2.1.1. Vehiculul Spațial Spirit



### 1.2.2. Sistemele de avioane fără pilot

Sistemele de avioane fără pilot au devenit, în ultima perioadă, o prezență tot mai importantă în activitățile militare și civile, științifice și economice, desfășurate de diferite state ale lumii.

Avioanele fără pilot sunt aeronave fără echipaj uman la bord, ce pot fi comandate la distanță sau pot efectua zboruri în regim autonom, având la bord senzori dedicați pentru diferite aplicații militare sau civile. Potrivit principiilor siguranței aeronautice, avioanele fără pilot sunt considerate acele aparate de zbor care sunt proiectate, construite, operate și menținute în exploatare, iar echipajele lor sunt de asemenea calificate, în concordanță cu prevederile relevante în materie aeronautică.

Întrebuințarea avioanelor fără pilot a fost statuată, pentru prima dată, prin prevederile Convenției de la Chicago, din 07 decembrie 1944, care în articolul opt prevede: „Nici o aeronavă care poate zbura fără pilot nu va putea survola, fără pilot, teritoriul unui Stat contractant fără a avea o autorizație specială a statului respectiv și în conformitate cu prevederile acestei autorizații. Fiecare Stat contractant se obligă să ia măsurile necesare pentru ca zborul fără pilot al unei asemenea aeronave în regiunile deschise aeronavelor civile să fie controlat, în așa fel încât să se evite orice pericol pentru aeronavele civile.”

Începând cu anul 1960, avioanele fără pilot au fost întrebuințate cu predilecție în domeniul militar, acestea acționând îndeosebi pentru supraveghere și recunoaștere. Ulterior, spre sfârșitul secolului XX și începutul secolului XXI, preocupările pentru cercetarea – dezvoltarea acestor tipuri de mijloace au dus la apariția unei adevărate industrii, în cadrul căreia au fost adoptate tehnologii noi, care au permis o diversificare mare a produselor în domeniu și o creștere permanentă a performanțelor lor. Au apărut, astfel, avioanele fără pilot specializate în culegerea de informații relevante, la nivel tactic, operativ și strategic, pentru descoperirea și marcarea țintelor și, mai nou, pentru lovirea diferitelor obiective.

Un număr însemnat de state europene folosesc avioane fără pilot, mai ales începând cu cea de a doua jumătate a deceniului opt al secolului XX. Acestea, pentru început, au importat astfel de mijloace, de la principalii producători – SUA și ISRAEL. În ultima perioadă, se remarcă o creștere a interesului statelor europene pentru dezvoltarea de industrii proprii în acest domeniu, axate pe producerea de drone de dimensiuni mici și mijlocii, sisteme de comandă-control, componente și diferiți senzori. Realizările în acest domeniu se bazează, în cea mai mare măsură, pe rezultatele cercetărilor efectuate cu posibilitățile proprii fiecărui stat.

De asemenea, există programe dezvoltate pe baza colaborării între marii producători din domeniu, cu precădere pentru producerea avioanelor fără pilot de altitudine, înălțime și anduranță mari, precum și a celor de luptă, în acest sens remarcându-se colaborarea franco-israeliană pentru realizarea avionului fără pilot HARFANG, cea pentru dezvoltarea sistemului nEUROn, dintre firme din Franța, Suedia, Grecia, Elveția, Spania și Italia. La acestea se adaugă acțiunile desfășurate, în Marea Britanie, de către compania BAE Systems, în vederea realizării dronei TARANIS.

În anul 1987, în înzestrarea Aviației Militare Române, a fost introdus sistemul de avioane de cercetare fără pilot VR-3, de producție sovietică, cu care a fost dotată o escadrilă de cercetare fără pilot.

Escadrila a fost prevăzută cu 12 avioane fără pilot, reactive, VR-3, având viteza de 950km/h, plafonul maxim de zbor de 5000m și raza de acțiune de 200km. Dintre acestea opt aparate puteau executa misiuni de fotogrametrie, iar patru aparate erau specializate pentru cercetare video, pentru care transmiterea informațiilor se realiza prin intermediul unei legături radio, către o stație de la sol.

Sistemul de avioane fără pilot VR-3 a fost complet autonom și dislocabil, personalul având la dispoziție întregul suport tehnologic necesar pregătirii sistemelor de comandă și a echipamentelor, întreținerii la sol, pregătirii pentru lansare, lansării, recuperării, transportului și întreținerii aparatelor, precum și pentru recepția, procesarea, interpretarea și transferul informațiilor către eșalonul superior, beneficiarul misiunilor de cercetare.

La începutul anilor 2000, escadrila de cercetare fără pilot a fost desființată, iar aparatele de zbor au fost stocate și ulterior casate, mai puțin un exemplar care, în prezent, se află la Muzeul Aviației Române.

În România au existat preocupări pentru construirea unor tipuri de avioane fără pilot, dar acestea, de cele mai multe ori, au rămas la stadiul de proiect, sau au fost realizate diferite prototipuri sau demonstratoare.

De regulă, activitățile de cercetare-dezvoltare s-au desfășurat în cadrul institutelor de cercetări de profil și al universităților și, mai nou, de companii private, pe proiecte finanțate din fonduri proprii.

Până în prezent nu s-au manifestat preocupări din partea autorităților guvernamentale române de dezvoltare a unor proiecte de sisteme de avioane fără pilot, deși evenimentele care au avut loc în ultima perioadă de timp pe plan mondial, au scos în evidență atât necesitatea cât și importanța folosirii acestor mijloace în vederea culegerii de informații relevante pentru numeroase domenii de activitate militare și civile.

În acest context, Institutul Național de Cercetări Aeronautice și Spațiale (INCAS) a realizat prototipul IAR-T, un mini avion fără pilot cu o greutate de 20kg, fiind echipat cu un sistem video în timp real, având viteza maximă de 180km/h, o anduranță de 30 de minute și raza maximă de acțiune de 10km. Acesta a fost conceput pentru a fi utilizat, mai ales, în domeniul cercetărilor științifice, dar din lipsa comenzilor nu a mai fost dezvoltat.



Fig. 1.2.2.1. IAR-T

La rândul său, Institutul Național de Aviație S.A., a realizat prototipurile a două avioane fără pilot ARGUS S și ARGUS XL, ambele fiind în faza de dezvoltare.

Avionul fără pilot ARGUS S a fost conceput pentru a efectua misiuni de supraveghere, fiind proiectat pentru a avea o greutate maximă de 140 kg și o viteză maximă de 240 km/h, având o rază de acțiune de 200 km în zbor autonom. Decolarea și aterizarea sunt radio-comandate de la sol, de un operator, în timp ce zborul de croazieră se bazează, în întregime, pe sistemele de bord.

Sistemul ARGUS XL este un avion fără pilot în configurație Canard, construit, în întregime, din materiale compozite. Aripile și ampenajele sunt detașabile, iar trenul de aterizare principal este o lamă elastică, echipată cu frâne mecanice și nu este retractabil. Trenul de bot este fix, cu direcție și cu piston oleo-pneumatic.

În cadrul Facultății de Științe și Ingineria Mediului a Universității "Dunărea de Jos" funcționează un Centru de Excelență pe Probleme de Mediu, unde a fost realizat un avion fără pilot, cu o autonomie de zbor de 150 km, care poate fi utilizat pentru supraveghere, recunoaștere, efectuarea de hărți cadastrale, observații de mediu și de biodiversitate.

Agencia de Cercetare pentru Tehnică și Tehnologii Militare, din Ministerul Apărării Naționale, în prezent, dezvoltă microsistemul de avioane fără pilot SACT5-BOREAL.

Realizările românești din ultima perioadă, cu pregnanță cele din domeniul privat, probează capacitatea specialiștilor români de a proiecta și de a produce sisteme de avioane fără pilot la un nivel tehnologic comparabil cu cel atins de marii producători de pe plan mondial, fiind create premisele dezvoltării unei noi subramuri a industriei aeronautice în România. Însă, posibilitățile de realizare a acestora sunt grevate, pe de o parte, de fondurile financiare limitate ale producătorilor și, pe de altă parte, de interesul scăzut al autorităților statului român de a achiziționa astfel de sisteme din producția internă.

### 1.2.3. iRobot

iRobot, un robot mic, foarte ușor, care ajută soldații să aibă o viziune de ansamblu în cadrul unei misiuni fără a fi descoperiți, a fost lansat de FirstLook. Acesta este dotat cu două brațe ce îl ajută pe robot să se poată ridica și să evite micile obstacole, sau să urce scări de 20 cm înălțime.

Dotat și cu o cameră video care oferă o imagine de 360 de grade, robotul poate transmite aceste imagini către un ecran cu touchscreen ce se montează pe încheietura mâinii, care are și rol de operator al unității de control.



Fig. 1.2.3.1. Interfață iRobot

Robotul are dimensiunile de 26 cm lungime, 23 cm lățime și 10 cm înălțime și cântărește sub 2.5 kg. Acesta poate evita obstacole de 20 cm, poate rezista la o cădere de la o înălțime de aproximativ cinci metri, poate sta sub apă până la o adâncime de un metru și poate atinge o viteză de 5.5 km/h.

Pe lângă echipamentul video montat pe micul robot, există și un echipament audio, care va ajuta soldații să asculte ceea ce se întâmplă în jurul robotului, dar și să poată trimite mesaje oamenilor de pe câmpul de luptă.

Robotul are autonomie timp de șase ore în care se poate deplasa oriunde este necesar, dar poate rezista până la zece ore în stare pasivă, în timp de observație.

Prețul oficial al iRobot nu a fost făcut public, dar se presupune că ar valora în jur de 100\$, iar FirstLook se crede că are mii de exemplare pregătite pentru a fi livrate.



Fig. 1.2.3.2. Robotul iRobot

#### 1.2.4. Robotul ASIMO

La ora actuală, ASIMO este printre cei mai inteligenți roboți din lume, având o inteligență artificială minimă, putând reține fețe și apoi recunoaște și chiar poate purta o conversație.

Momentan, nu poate fi achiziționat într-o variantă de uz casnic, dar pe termen lung are mari șanse să ne facă viața mai ușoară.

De la distanță, pare a fi un tânăr astronaut în costumul celor de la NASA. Are 1,3 metri înălțime și 50 de kilograme, iar numele său este o variantă prescurtată pentru „Pas avansat în mobilitate inovativă” (Advanced Step in Innovative Mobility). Arată uman și tocmai din acest motiv, în cadrul debutului său la New York, la un hotel din Manhattan, ASIMO a întâmpinat audiența cu mesajul „Hello New York! Thank you for coming today!”. Aceasta a fost o înregistrare audio ce folosea vocea unui copil.

ASIMO poate să meargă, să alerge și să urce pe scări. Viteza sa maximă de deplasare atinge o limită decentă de 9 kilometri la oră. Este suficient de flexibil și ponderat în manifestarea forței de care e capabil încât să poată deschide o sticlă și o poate turna într-un pahar. În ultima perioadă, a învățat să vorbească foarte bine în limbajul semnelor și tinde să devină din ce în ce mai autonom.

Dacă totul continuă să evolueze în același ritm, în doar câțiva ani va fi capabil să ajute un bătrân singur în casă la toate activitățile zilnice.

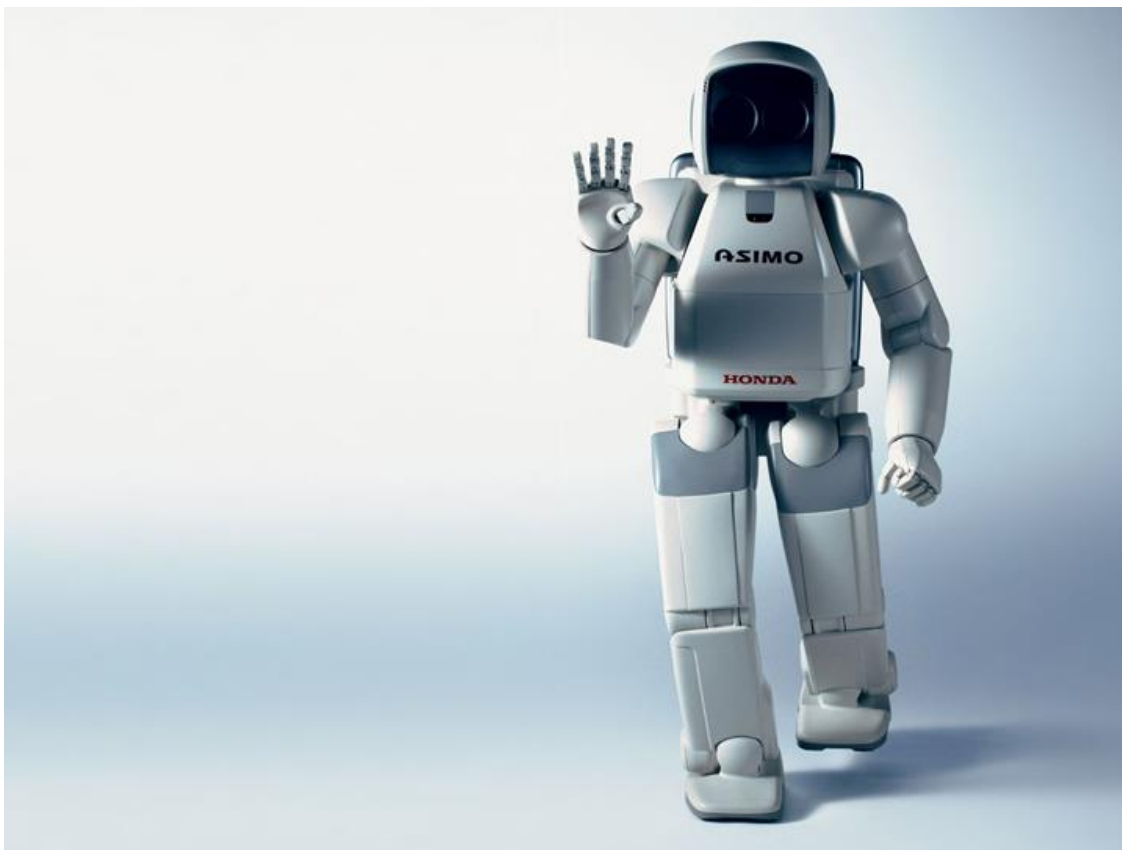


Fig. 1.2.4.1. Robotul ASIMO

## 2. APLICABILITATE

### 2.1. Aplicații ale sistemelor robotice comandate de la distanță

În general, roboții comandați de la distanță își au rolul în medii periculoase pentru om, cum ar fi mediile toxice, mediile cu temperaturi extreme, mediile cu radiații ș.a.

Medii greu accesibile sunt, de asemenea, medii în care roboții comandați de la distanță sunt eficienți. Ca exemplu, Vehiculul spațial Spirit (capitolul 1.2.1.).

Mediile industriale, spre deosebire de altele (extraterestre, subacvatice, urbane, agricole, forestiere etc.), au caracteristici specifice care au determinat folosirea unor structuri dedicate ale roboților industriali fiși și, în ultimul timp, induc personalizări specifice privind dezvoltarea și implementarea roboților mobili universali ca roboți mobili industriali. Printre caracteristicile și cerințele specifice ale mediilor industriale legate de implementarea roboților, în special, a celor mobili, se evidențiază: structurarea parțială a locațiilor de lucru, nivelul crescut de securitate, necesitatea unui sistem senzorial extern precum și necesitatea unor sistemele auxiliare.

La proiectarea, dezvoltarea și implementarea roboților industriali se urmărește, pe de-o parte, mărirea eficienței acțiunilor asupra mediilor tehnologice prin creșterea vitezei și preciziei de execuție și, pe de altă parte, înlocuirea operatorilor umani care execută operații simple repetitive sau care lucrează în medii austere (căldură/frig, lumină/întuneric, vibrații/zgomote etc.) și nocive (cu radiații, poluate etc.) sau scoaterea operatorului uman din mediul respectiv, controlând robotul de la distanță.

În perioada 2000-2003, numărul de unități de roboți pentru servicii este estimat la peste 49,400 unități, din care 40,000 sunt roboți domestici (excluzând cei de curățenie cu vacuum) și aproximativ 5,000 sunt roboți medicali. Roboții domestici destinați uzului casnic cu sistem de vacuum sunt introduși pe piață la sfârșitul anului 2000. De asemenea apar de la zi la zi roboți miniaturizați ce vor intra într-o mulțime de case și vor suplini o parte din activitățile umane. Mai mult de atât, tehnologia actuală poate fi folosită în scopul complementării asistenței umane în cazuri dificile prin intermediul roboților de asistență ce suplinesc cu succes asistenții persoanelor cu handicap. În cadrul sectorului roboți de serviciu, prețul destul de ridicat este încă un impediment, însă avantajele multiple care le oferă în timp justifică investiția. Exact cum ne folosim acum de telefoane mobile, PC-uri așa ne vom putea folosi de "bucătării inteligente" sau "case inteligente", în care diverse echipamente vor fi conectate la un PC care va coordona "muncile" în casă, prin intermediul roboților de serviciu. De exemplu : robot de curățenie, robot subacvatic, robot cositoare; robot de curățat cu vacuum. În funcție de serviciile pentru care au fost creați, din totalul analizat în 1999, la un număr de 6600 de unități s-au numărat 50%, roboți domestici, 14% roboți subacvatice, 12% roboți medicali, 6% roboți pentru curățenie și restul de 23% au fost în categoria "altele".



Fig. 2.1.1. Robot Casnic

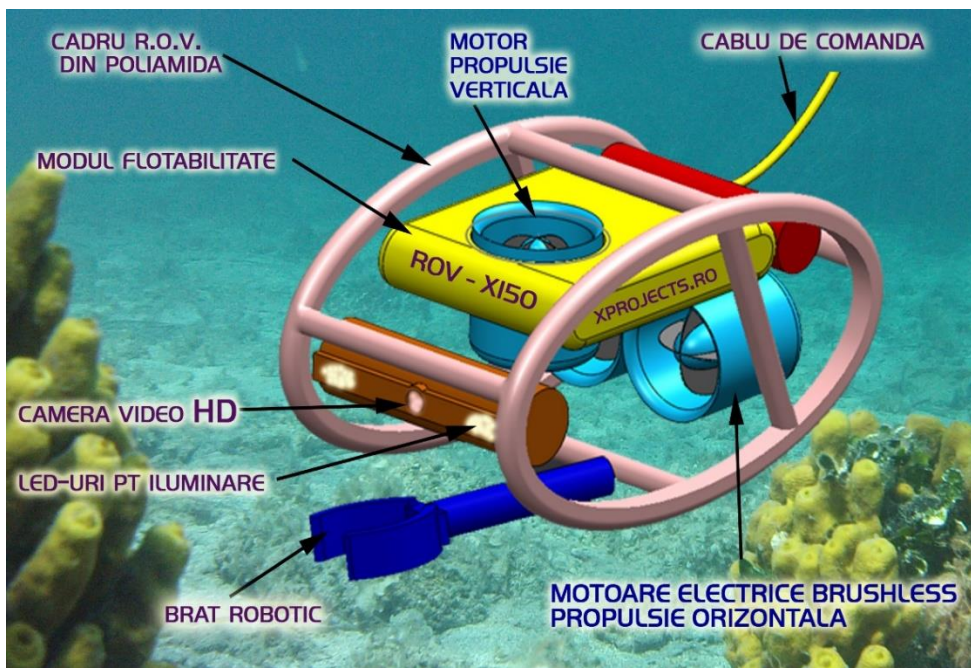


Fig. 2.1.2. Robot Subacvatic

## 2.2. Flexibilitate

În acest caz, prin flexibilitate înțelegem simplitatea prin care, cu mici modificări aduse robotului mobil, acesta să poată îndeplini mai multe funcții.

De exemplu, o dronă militară, dotată cu armament și muniție poate deveni foarte ușor o dronă pentru fotografiat / înregistrat, schimbându-i "efectorul final", mai exact, renunțând la armament, în favoarea unui aparat foto.

O echipă de ingineri de la Universitatea Harvard au creat primul „octobot”, un dispozitiv despre care specialiștii spun că reprezintă primul pas în producția unor noi tipuri de roboți avansați. Sistemul automatizat are opt „tentacule” și o înfățișare similară cu cea a unei caracatițe.

Robotul este parțial autonom și a fost produs, într-o mare măsură, cu ajutorul unei imprimante 3-D. Pentru a funcționa, sistemul trebuie alimentat cu un gaz obținut din peroxid de hidrogen și un catalizator al platinei.

„Una dintre principalele viziuni în domeniul roboticii este cea a producerii unor dispozitive în totalitate flexibile, însă problema a fost întotdeauna cea a înlocuirii componentelor rigide precum bateriile și sistemele electronice de control cu unele flexibile, care să poată fi mai apoi puse laolaltă. Această cercetare demonstrează faptul că putem fabrica ușor componentele-cheie ale unui robot simplu și în totalitate flexibil”, a declarat coordonatorul proiectului octobotului, profesorul Robert Wood, într-un comunicat de presă dat de Universitatea Harvard.

## 2.3. Viitoare Îmbunătățiri

În primul rând, pentru a mări aria aplicabilității, se poate achiziționa un IP public sau un domeniu, pentru a hosta aplicația web pe acel IP public sau domeniu. În acest fel, oricine cu o conexiune la internet ar avea acces la robotul mobil. Totuși, ca o măsură de siguranță, s-ar implementa un sistem de logare, astfel încât să se limiteze accesul doar pentru persoanele autorizate, din orice colț al lumii.

O altă îmbunătățire ar reprezenta un kit al robotului propriu-zis cu materiale cu o calitate superioară. Mai exact, motoare mai puternice, șasiu mai puternic, un sistem de suspensie bazat pe arcuri și telescoape.

De asemenea, datorită limitărilor shield-ului ULN2803, robotul se mișcă pe direcția înainte prin rotirea celor 2 roți din fața robotului, iar pe direcția înapoi prin rotirea celor 2 roți din spatele robotului. O îmbunătățire ar fi controlul sincron al celor 4 roți pe direcția pe care se dorește a se deplasa robotul. Acest lucru poate fi pus la punct prin alte shield-uri sau drivere pentru Raspberry Pi și motoare de curent continuu, cu un preț mult mai ridicat.

Fiind o aplicație web, aceasta poate fi accesată de pe mai multe tipuri de dispozitive. Însa pentru dispozitivele cu ecran tactil, codul sursă al aplicației trebuie reconfigurat și adaptat ecranului tactil.

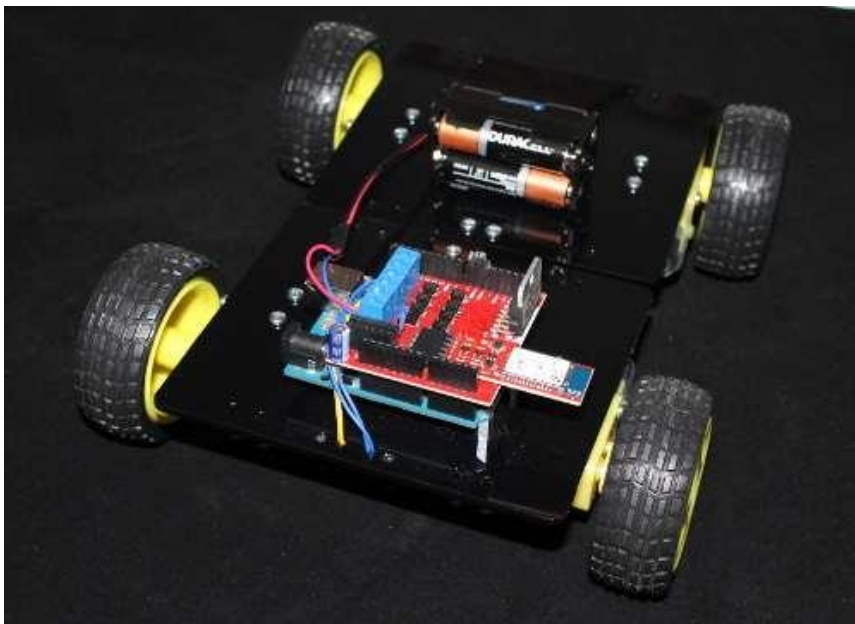


### 3. ALTE VARIANTE DE PROIECTARE

#### 3.1. Robot mobil cu Arduino, comandat din telefonul mobil folosind Bluetooth

Folosind același kit robot, dar aplicând alte principii, obținem un alt proiect. Este vorba despre un robot mobil cu Arduino, comandat din telefonul mobil, folosind Bluetooth.

Este un proiect mai simplist și în același timp mai limitat.



*Fig. 3.1.1. Robot mobil cu Arduino*



*Fig. 3.1.2. Robot mobil cu Arduino*

Practic, este tot un robot mobil, controlat de la distanță folosind telefonul mobil sau chiar laptop-ul, dar distanța în cauză este mult mai mică, datorită limitărilor modulului Bluetooth.

### 3.1.1. Bluetooth

Bluetooth este un set de specificații (un standard) pentru o rețea personală (engleză: personal area network, PAN) fără fir (wireless), bazată pe unde radio. Tehnologia Bluetooth a fost creată în 1994.

„Bluetooth” este o traducere în engleză a cuvântului scandinav Blåtand/Blåtann, cum era supranumit regele viking Harald I al Danemarcei din sec. al X-lea. Harald I a unit Norvegia și Danemarca; el era renumit ca fiind foarte comunicativ și se pricepea să îi facă pe oameni să comunice între ei. În română bluetooth s-ar traduce „dinte albastru”.

Specificația Bluetooth a fost formulată pentru prima dată de Sven Mattisson și Jaap Haartsen, muncitori în orașul Lund, Suedia, la divizia de telefonie mobilă a companiei Ericsson. La 20 mai 1998 a fost fondată gruparea Bluetooth Special Interest Group (SIG), care azi are rolul de a vinde firmelor tehnologia Bluetooth și de a urmări evoluția acestei tehnologii.

Printr-o rețea Bluetooth se poate face schimb de informații între diverse aparate precum telefoane mobile, laptop-uri, calculatoare personale, imprimante, camere foto și video digitale sau console video printr-o unde radio criptate (sigur) și de rază mică, desigur numai dacă aparatele respective sunt înzestrate și cu Bluetooth.

Aparatele care dispun de Bluetooth comunică între ele atunci când se află în aceeași rază de acțiune. Ele folosesc un sistem de comunicații radio, așa că nu este nevoie să fie poziționate față în față pentru a transmite; dacă transmisia este suficient de puternică, ele pot fi chiar și în camere diferite.



Fig. 3.1.1.2. Logo Bluetooth

### 3.2. Robot mobil, comandat prin telecomandă IR

La fel ca și varianta propusă anterior, acest proiect folosește același kit robot, dar diferă prin modul de a transmite informația. În acest caz este vorba despre mediul infraroșu.



Fig. 3.2.1. Robot mobil, comandat prin telecomandă IR

Chiar dacă este un proiect mai ușor de realizat, este, în același timp, un proiect mai limitat.

La fel ca și în cazul modulului Bluetooth, telecomanda cu infraroșu are un dezavantaj și anume distanța.

#### 3.2.1. Infraroșu

Radiația în infraroșu (IR) este o radiație electromagnetică a cărei lungime de undă este mai lungă decât cea a luminii vizibile (400-700 nm), dar mai scurtă decât cea a radiației terahertz (100  $\mu\text{m}$  - 1 mm) și a microundelor ( $\sim 30000 \mu\text{m}$ ). Majoritatea radiației termice emise de către obiectele aflate la temperatura camerei este în infraroșu.

Energia în infraroșu este emisă sau absorbită de molecule atunci când se schimbă mișcările de rotație - vibrație. Energia în infraroșu excita moduri de vibrație într-o moleculă printr-o schimbare de dipol, făcându-l interval de frecvență util pentru studiul acestor stări energetice pentru moleculele de simetrie corespunzătoare. Spectroscopia în infraroșu examinează absorbția și transmiterea de fotoni în intervalul energetic infraroșu.

Radiațiile infraroșii sunt folosite în aplicații industriale , științifice sau medicale. Aparatele pentru vedere nocturna folosind iluminare infraroșie apropiată activă oferă observarea oamenilor și animalelor fără ca observantul să fie detectat. Astronomia în infraroșu folosește senzori echipați pe telescoape pentru a trece prin regiunile greu vizibile din spațiu precum norii moleculari, mai sunt folosiți pentru a detecta noi planete sau pentru a detecta traiectoria obiectelor în spațiu. Camerele cu detectoare infraroșii sunt folosite pentru a detecta pierderea de căldura din sisteme izolate, pentru a observa schimbările de traiectorie a sângelui în corpul uman și pentru a detecta aparate electrice care se supraîncălzesc.

Fotografierea în infraroșu este folosită în aplicațiile militare pentru achiziția de date dar și în scopuri industriale sau civile. Utilizările militare includ vederea nocturna, supravegherea pe timp de noapte, localizare și urmărire. Omul la temperatura normală a corpului radiază pe lungimea de undă de 10 micrometri. Utilizările civile includ analiza eficienței termale, monitorizarea mediului înconjurător, inspectarea uzinelor industriale, detectarea temperaturii la distanță, comunicațiile fără fir pe distanțe scurte, spectrografie și meteorologie.

Transmisiile de date în infraroșu sunt de asemenea utilizate în transmisiile de date între perifericele sistemelor complexe cât și o perioadă bună de timp pentru a transfera date între telefoanele mobile. Aceste dispozitive sunt utilizate conform standardelor publicate de IRDA (Infrared Data Association). Telecomenzile și celelalte dispozitive ce utilizează transmiterea de date în infraroșu folosesc LED-uri (light-emitting diodes) pentru a emite radiație infraroșie care este direcționată de o lentilă plasticată într-un fascicul îngust. Fasciculul este modulat și se deschide și închide pentru a cripta date. Receptorul utilizează o fotodiada din silicon pentru a converti radiația infraroșie în curent electric. Răspunde numai semnalului pulsatoriu creat de transmițător și filtrează schimbările radiației infraroșii din lumina ambientală. Comunicațiile prin infraroșu sunt folosite utilizărilor casnice în zonele cu densitate mare a populației. Infraroșu nu poate străpunge pereții și astfel nu poate interfera cu alte dispozitive din încăperi alăturate. Infraroșu este cea mai comună metodă folosită pentru telecomenzile ce comandă aparate simple. Telecomenzile prin infraroșu utilizează protocoale precum RC-5, SIRC, NEC etc. pentru a putea funcționa.

Comunicațiile optice în spațiul liber ce utilizează lasere prin infraroșu poate fi o metodă foarte convenabilă pentru a instala o legătură de comunicație în spațiul urban care să funcționeze până la 4 gigabit/s, în comparație cu îngroparea cablului de fibră optică.

Laserele infraroșii sunt utilizate pentru a asigura lumina pentru sistemele de comunicație prin fibră optică. Lumina infraroșie cu lungimea de undă în jurul valorii de 1330 nm sau 1550 nm sunt cele mai bune decizii pentru firele standard de silicon.

### 3.3. Varianta aleasă - Robot Mobil Comandat De La Distanță Prin Raspberry Pi, Cu Redare Video

Varianta aleasă reprezintă cea mai optimă variantă, când vine vorba de control de la distanță. Prin modulul Wi-Fi integrat pe Raspberry Pi, robotul se poate conecta la o rețea wireless, dat fiind faptul că în zilele noastre rețelele wireless sunt foarte des întâlnite. În caz că robotul se află într-o zonă fără vreo rețea wireless, cu ajutorul unui router portabil montat pe robot, a unei cartele GSM și a unui abonament de date, robotul nu mai cunoaște limite impuse de distanță, atâta timp cât ISP-ul dispune de semnal în zona în care se află robotul.

#### 3.3.1. Wi-Fi

Wi-Fi este numele comercial pentru tehnologiile construite pe baza standardelor de comunicație din familia IEEE 802.11 utilizate pentru realizarea de rețele locale de comunicație (LAN) fără fir (wireless, WLAN) la viteze echivalente cu cele ale rețelelor cu fir electric de tip Ethernet. Suportul pentru Wi-Fi este furnizat de diferite dispozitive hardware, și de aproape toate sistemele de operare moderne pentru calculatoarele personale (PC), routere, telefoane mobile, console de jocuri și cele mai avansate televizoare.

Standardul IEEE 802.11 descrie protocoale de comunicație aflate la nivelul gazdă-rețea al Modelului TCP/IP, respectiv la nivelurile fizic și legătură de date ale Modelului OSI. Aceasta înseamnă că implementările IEEE 802.11 trebuie să primească pachete de la protocoalele de la nivelul rețea (IP) și să se ocupe cu transmiterea lor, evitând eventualele coliziuni cu alte stații care doresc să transmită.

802.11 face parte dintr-o familie de standarde pentru comunicațiile în rețele locale, elaborate de IEEE, și din care mai fac parte standarde pentru alte feluri de rețele, inclusiv standardul 802.3, pentru Ethernet. Cum Ethernet era din ce în ce mai popular la jumătatea anilor 1990, s-au depus eforturi ca noul standard să fie compatibil cu acesta, din punctul de vedere al transmiterii pachetelor.[1]

Standardul a fost elaborat de IEEE în anii 1990, prima versiune a lui fiind definitivată în 1997. Acea versiune nu mai este folosită de implementatori, versiunile mai noi și îmbunătățite 802.11a/b/g fiind publicate între 1999 și 2001. Din 2004 se lucrează la o nouă versiune, intitulată 802.11n și care, deși nu a fost definitivată, este deja implementată de unii furnizori de echipamente.

Din punct de vedere al securității, IEEE și Wi-Fi Alliance recomandă utilizarea standardului de securitate 802.11i, respectiv a schemei WPA2. Alte tehnici simple de control al accesului la o rețea 802.11 sunt considerate nesigure, cum este și schema WEP, dependentă de un algoritm de criptare simetrică, RC4, nesigur.



Fig. 3.3.1.1. Logo Wi-Fi

Limitările standardului provin din mediul fără fir folosit, care face ca rețelele IEEE 802.11 să fie mai lente decât cele cablate, de exemplu Ethernet, dar și din folosirea benzii de frecvență de 2,4 GHz, împărțită în 12 canale care se suprapun parțial două câte două. Limitările date de consumul mare de energie, precum și de reglementările privind puterea electromagnetică emisă, nu permit arii de acoperire mai mari de câteva sute de metri, mobilitatea în cadrul acestor rețele fiind restrânsă. Cu toate acestea au apărut și unele tehnologii care permit legături fără fir bazate pe standardul 802.11 între două puncte fixe aflate la distanțe de ordinul sutelor de kilometri.

O limitare importantă a rețelelor Wi-Fi o constituie aria de acoperire. Ea depinde mult de capacitățile antenelor dispozitivelor și de topografia particulară a zonei pe care urmărește rețeaua să o acopere. Plantele absorb radiațiile electromagnetice, și astfel instalarea unei rețele într-o zonă împădurită (cum ar fi un parc) limitează aria de acoperire a acesteia. Pereții de beton reflectă puternic undele radio, instalarea unei rețele într-o clădire aducând astfel limitarea numărului de camere ce poate fi acoperit de o singură celulă. În interiorul clădirilor, un punct de acces cu o antenă de dimensiuni mici și un preț accesibil poate acoperi o rază de aproximativ 32 m, iar în exterior, același punct de acces poate ajunge la 95 m. Aria de acoperire poate fi și mai restrânsă în cazul folosirii benzii de 5 GHz în locul celei de 2,4 GHz (mai zgomotoasă, dar în care se poate acoperi o arie mai mare). Transmisiunea la cea mai mare distanță cu ajutorul unor dispozitive Wi-Fi a fost realizată, folosind antene puternice și semnale direcționate, de Ermanno Pietrosemoli de la Escuela Latinoamericana de Redes, care a transferat 3 MB de date între vârfulurile El Aguila și Platillon din Venezuela, aflate la o distanță de 382 km.

Dar, cum a fost specificat și anterior, Wi-Fi-ul este foarte "dinamic". Cu ajutorul unui router portabil atașat pe robot, limitele conexiunii devin limitele impuse de către furnizorul abonamentului, iar aceste limite sunt destul de largi.

Mai jos este reprezentată harta României cu acoperirea semnalului, pe care un important furnizor de servicii de telefonie și date mobile, le oferă în țara noastră. Pentru ca robotul să funcționeze optim și să ofere un timp de răspuns cât mai mic, ideal ar fi, ca tipul conexiunii să fie minim 3G.

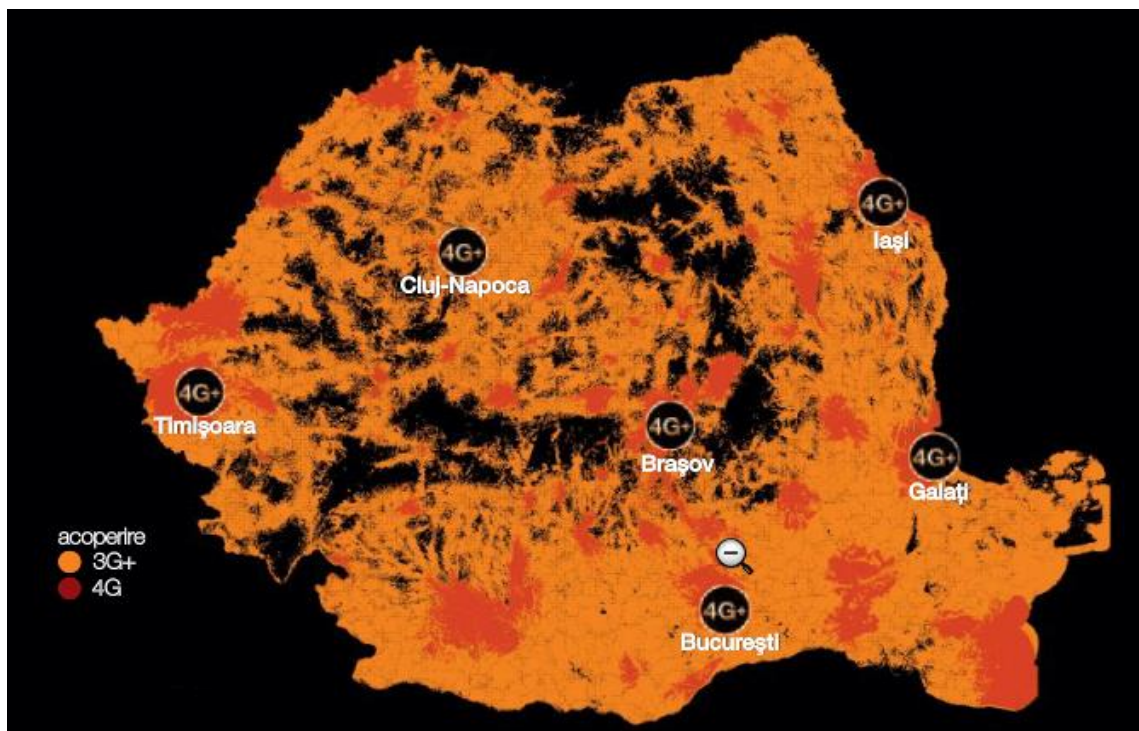


Fig. 3.3.1.2. Hartă acoperire semnal

### 3.4. Diferențe între variantele de proiectare

Cea mai importantă diferență dintre variantele de proiectare definite mai sus este modul de transmitere a datelor.

În primul caz, conexiunea prin Bluetooth oferă o conexiune stabilă, cu un consum redus de energie, dar și limitări mari în ceea ce privește distanța.

Modulul infraroșu are, la rândul său, o serie de avantaje : cost redus, simplitate, consum redus de energie; dar prezintă și o serie de dezavantaje, care, în acest caz, sunt indispensabile, mai exact, raza de acoperire foarte mică.

Modulul Wi-Fi consumă, într-adevăr, mai multă energie decât celelalte variante, dar, când vine vorba de distanța de acționare, este cel mai eficient. Pe lângă asta, poate fi accesat de orice dispozitiv cu un browser modern și o conexiune la internet.

## 4. CONSTRUCȚIA ROBOTULUI

### 4.1. Elemente componente, programe și limbaje de programare utilizate

#### 4.1.1. Raspberry Pi 3 - Model B

##### 4.1.1.1. Generalități

Raspberry Pi este o placă de dezvoltare de tip SBC (Single Board Computer) – un sistem de calcul nemodular implementat pe un singur cablaj electronic. Chiar dacă are dimensiuni reduse (85mm x 56mm), Raspberry Pi este un calculator complet permițând funcționalități obișnuite precum rularea unui sistem de operare (Linux sau Windows) și rularea de aplicații utilizator (jocuri, editoare de text, medii de programare, redarea de muzică și filme, aplicații de teleconferință, aplicații Internet). Diferențele între o placă Raspberry Pi și un calculator personal (PC) sau laptop constau atât în dimensiunea redusă a plăcii, cât și în puterea mai mică de calcul a acesteia – nu are aceleași performanțe de calcul precum un PC desktop care are un cost și o dimensiune de câteva ori mai mari. Putem compara placa Raspberry Pi cu o tabletă sau cu un sistem de tip NetBook, dar fără a dispune de ecran și tastatură. În plus, placa Raspberry Pi oferă posibilitatea de a conecta diverse componente electronice specifice sistemelor embedded: senzori, butoane, ecrane LCD sau pe 7 segmente, drivere de motoare, rele etc. Posibilitatea de a personaliza sistemele de programe (sistemul de operare, aplicațiile) și posibilitatea de interconectare cu alte componente electronice fac din placa Raspberry Pi un sistem de calcul ce poate sta la baza unor proiecte personale extrem de interesante și de puternice – un calculator ce poate fi integrat în sisteme electronice și mecanice proiectate și realizate de utilizator.



Fig. 4.1.1.1.1. Raspberry Pi



În ciuda dimensiunii reduse placa Raspberry Pi 3 dispune de periferice integrate numeroase acoperind complet funcționalitatea unui sistem de calcul (audio, video, porturi USB, conectivitate de rețea):

- Procesor SoC pe 64 de biți din familia ARMv8-A, Broadcom BCM2837, ce lucrează la o frecvență de 1.2GHz și dispune de 4 nuclee de tip ARM Cortex-A53;
- 1GB de memorie RAM (folosită și ca memorie video, partajată cu procesorul grafic);
- Procesor grafic Broadcoam VideoCore IV 3D integrat pe aceeași pastilă de siliciu ca și procesorul principal;
- Ieșire digitală video / audio HDMI;
- Ieșire analogică video (composite video) / audio mixtă prin intermediul unei mufe jack 3.5mm;
- Mufă de rețea RJ45 Ethernet 10/100 Mbit/s;
- Conectivitate WiFi 802.11n;
- Conectivitate Bluetooth 4.1 / BLE;
- 4 porturi USB 2.0;
- 40 de pini de intrare / ieșire (GPIO);
- Slot card de memorie microSD (utilizat pentru instalarea sistemului de operare);
- Conectori dedicați pentru cameră video (CSI) și afișaj (DSI);

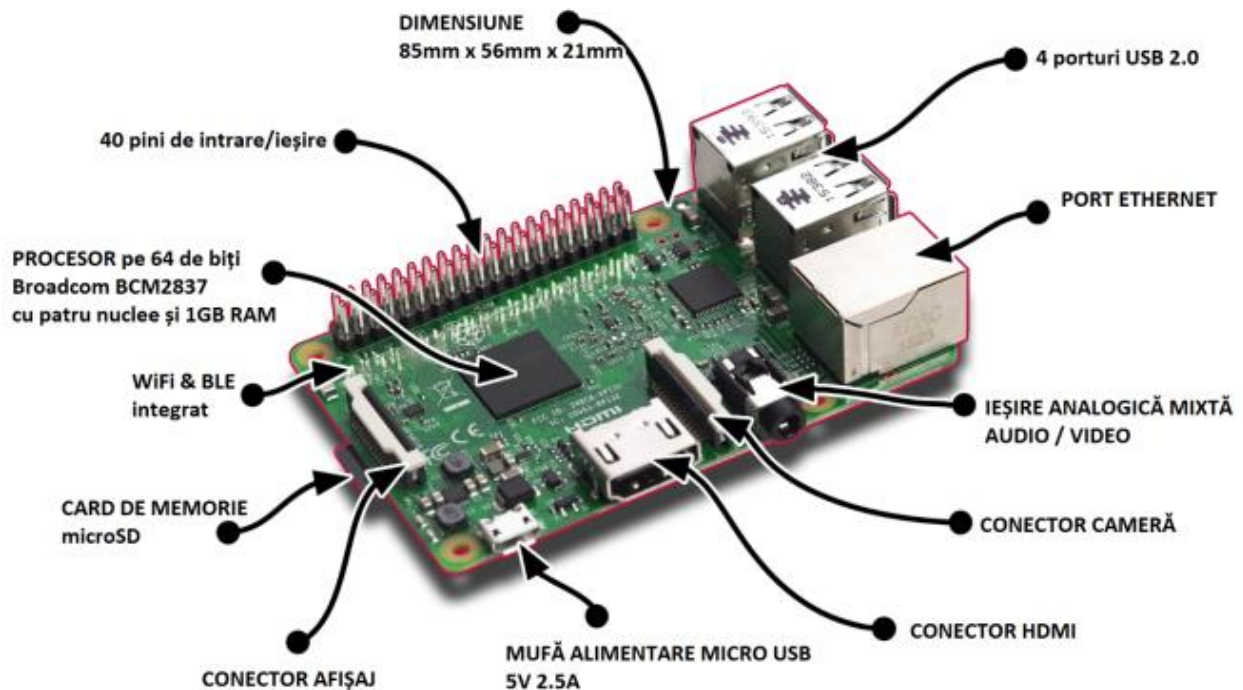


Fig. 4.1.1.1.2. Raspberry Pi

Pentru a pune în funcțiune placa Raspberry Pi 3 avem nevoie de următoarele componente suplimentare:

- Cablu HDMI și un monitor / televizor cu intrare HDMI. În cazul în care nu dispunem de un monitor / televizor cu intrare HDMI putem utiliza un adaptor HDMI-DVI sau un adaptor HDMI-VGA, depinde de intrarea monitorului pe care îl utilizăm.
- Alimentator de rețea cu ieșire de 5V, minim 2.5A și mufă microUSB. Este recomandată utilizarea alimentatorului oficial sau a unui alimentator de calitate care asigură o tensiune corectă și un curent suficient pentru alimentarea plăcii Raspberry Pi 3. În cazul în care utilizăm un alimentator ieftin putem distruge placa din cauza fluctuațiilor de tensiune sau putem întâmpina probleme în utilizare din cauza curentului insuficient debitat de sursa de tensiune.
- Tastatură și mouse USB. Sunt necesare pentru instalarea și configurarea inițială a sistemului. Dacă, ulterior, se utilizează sistemul de la distanță, tastatura, mouse-ul și monitorul nu mai sunt necesare. O variantă mai simplă, și mai comodă, este utilizarea unui dispozitiv mixt tastatură / touchpad fără fir.
- Card de memorie microSD, capacitate minimă 8GB (16GB recomandat), clasă de viteză 10. Cardul de memorie va stoca sistemul de operare la fel ca și hard-disk-ul în cazul unui sistem de calcul de tip PC sau laptop. Este foarte importantă utilizarea unui card microSD de calitate și de viteză adecvată din cauză că utilizarea unui card de calitate îndoielnică poate conduce la probleme de funcționare extrem de neplăcute: blocări în funcționare, resetarea aleatorie a sistemului, pierderi de date etc. Dacă este posibil, se recomandă utilizarea cardului microSD oficial.
- Dacă sistemul va fi utilizat într-o rețea locală pe cablu este necesar și un cablu de rețea UTP – patch-cord. Dacă se utilizează placa într-o rețea locală WiFi nu este necesar. Conectivitatea de rețea (conectivitatea Internet) nu este obligatorie pentru funcționarea plăcii, dar este recomandată deoarece altfel nu se pot realiza actualizările de securitate ale sistemului de operare, nu se pot menține corect setările de dată și oră și, bineînțeles, se pierde o funcționalitate importantă a sistemului.
- Opțional, dar recomandat, este utilizarea și a unei carcase pentru placa Raspberry Pi. Manipularea plăcii fără nici un fel de protecție poate conduce la deteriorarea acesteia datorită descărcărilor electrostatice (descărcarea sarcinii electrice statice a corpului uman în circuitul electronic prin atingerea componentelor conductoare a acestuia), a șocurilor mecanice și a murdăriei (praf, lichide, grăsimi).

În acest proiect, pentru a exclude firele, astfel încât robotul să fie independent, se va folosi o baterie externă Anker de 10Ah, cu parametrii de ieșire de 5V, respectiv 3A.

#### 4.1.1.2. Instalarea sistemului de operare

Următorul pas, necesar înainte de punerea în funcțiune a plăcii Raspberry Pi 3, este pregătirea cardului microSD pentru instalarea sistemului de operare. Placa Raspberry Pi 3 poate rula diverse distribuții ale sistemului de operare Linux și o versiune minimală a sistemului de operare Microsoft Windows 10 (prin versiune minimală înțelegem o variantă ce nu poate fi folosită ca platformă desktop ci doar ca platformă pentru aplicații – Windows 10 IoT Core). În continuare vom exemplifica instalarea

distribuției Linux Raspbian, sistemul de operare oficial al plăcii Raspberry Pi, sistem de operare foarte ușor de utilizat și recomandat pentru începători.

În cazul în care se utilizează cardul microSD oficial al plăcii Raspberry Pi, instalarea sistemului de operare este foarte simplă, deoarece cardul de memorie conține utilitarul NOOBS (New Out Of Box Software), ce facilitează instalarea diverselor sisteme de operare specifice plăcii Raspberry Pi. Pentru pornirea instalării sistemului de operare se inserează cardul în slotul microSD al plăcii și se pornește sistemul (toate echipamentele trebuie să fie conectate corect: tastatură, mouse, cablu HDMI). După inițializare, utilitarul NOOBS ne permite alegerea sistemului de operare pe care dorim să-l instalăm:

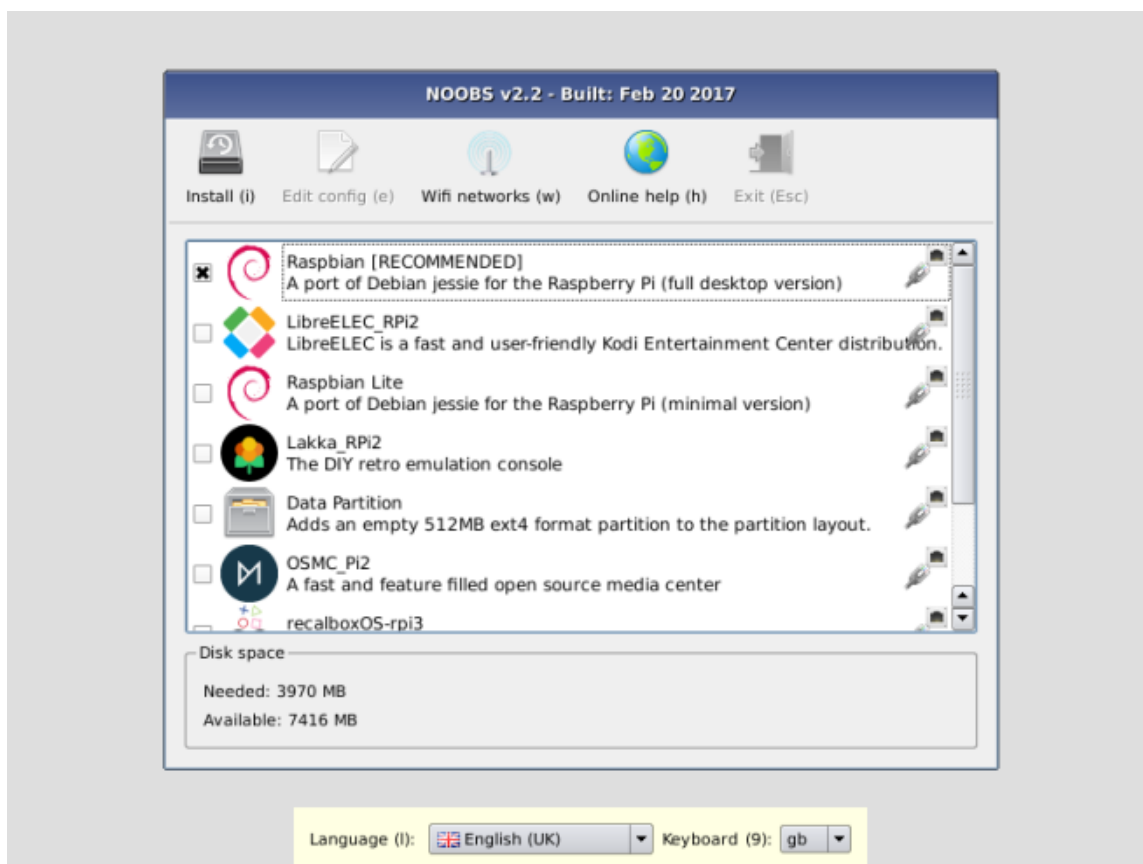


Fig. 4.1.1.2.1. NOOBS

În cazul în care placa nu este conectată la Internet (prin cablu sau prin WiFi), singura opțiune de instalare va fi sistemul de operare Raspbian, al cărui kit de instalare se află deja pe cardul de memorie. Toate celelalte sisteme de operare necesită conectivitate Internet pentru instalare. Conexiunea la Internet se poate realiza prin conectarea cu un cablu la o rețea ce oferă setări automate (DHCP) sau configurând accesul WiFi (opțiunea Wifi networks din partea de sus a ferestrei anterioare). După confirmarea sistemului de operat dorit, utilitarul NOOBS va instala automat sistemul de operare – acest proces durează câteva zeci de minute.

Dacă nu se dispune de un card microSD cu NOOBS preinstalat, putem utiliza un card microSD de calitate pe care să copiem utilitarul NOOBS – se downloadează și se dezarchivează pe cardul microSD. După această operație instalarea decurge ca în cazul precedent.

O alternativă la utilizarea programului NOOBS este copierea directă a sistemului de operare pe cardul microSD. Copierea nu se poate face direct, ca în cazul NOOBS, ci se realizează prin transferul unui fișier imagine cu ajutorul unui utilitar specializat, de exemplu: Etcher sau Win32DiskImager. Se downloadează imaginea de sistem de operare dorit (variantea LITE nu include interfața grafică) și se copiază pe cardul de memorie. După această operație cardul de memorie va conține sistemul de operare deja instalat, gata de utilizare. Cardul de memorie se poate introduce în slotul plăcii și se poate porni sistemul.

După finalizarea procesului de instalare sistemul va reporni și putem deja să lucrăm cu placa Raspberry Pi.

#### 4.1.1.3. Utilizarea plăcii Raspberry Pi 3 de la distanță

De multe ori este mai comod sau pur și simplu natura proiectului, cum se prezintă și cazul de față, impune accesul la distanță pentru utilizarea plăcii Raspberry Pi. Prin acces la distanță nu se înțelege neapărat faptul că placa se află la mare distanță, ci că nu este accesată de la o tastatură și un monitor direct conectate. În acest fel putem utiliza placa, fără a bloca un set de periferice dedicate.

Pentru a putea să ne conectăm la placa Raspberry Pi prin intermediul rețelei este necesar să cunoaștem adresa IP a acesteia. Dacă alocarea adreselor IP în rețeaua locală în care se utilizează placa se realizează în mod static, este suficient să ne notăm adresa alocată după instalarea sistemului de operare, utilizând comanda ifconfig în aplicația Terminal.

Dacă placa se conectează într-o rețea locală ce alocă dinamic adresele IP, asta înseamnă că la fiecare repornire este posibil ca adresa IP să se modifice și nu avem altă posibilitate decât să ”scanăm” rețeaua, pentru a descoperi ce adresă a fost alocată pentru placa Raspberry Pi. Scanarea se face cu aplicații specifice, de exemplu: Angry IP Scanner sau nmap.

După instalare, aplicația Angry IP Scanner permite verificarea întregii plaje de adrese dintr-o rețea locală și raportează ce adrese IP sunt active (sunt alocate) și ce servicii rulează sistemele de calcul respective.

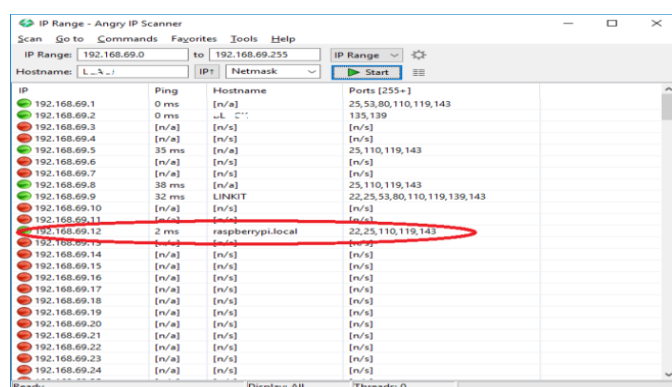


Fig. 4.1.1.3.1. Angry IP Scanner

Pentru accesul la distanță în linie de comandă se utilizează programul SSH, ce permite deschiderea unei sesiuni de tip terminal de pe un alt sistem de calcul. Este necesară instalarea unui client SSH pe sistemul de pe care se va face accesul. Cel mai cunoscut client SSH pentru Windows este putty, sistemele Linux au clientul SSH instalat implicit:

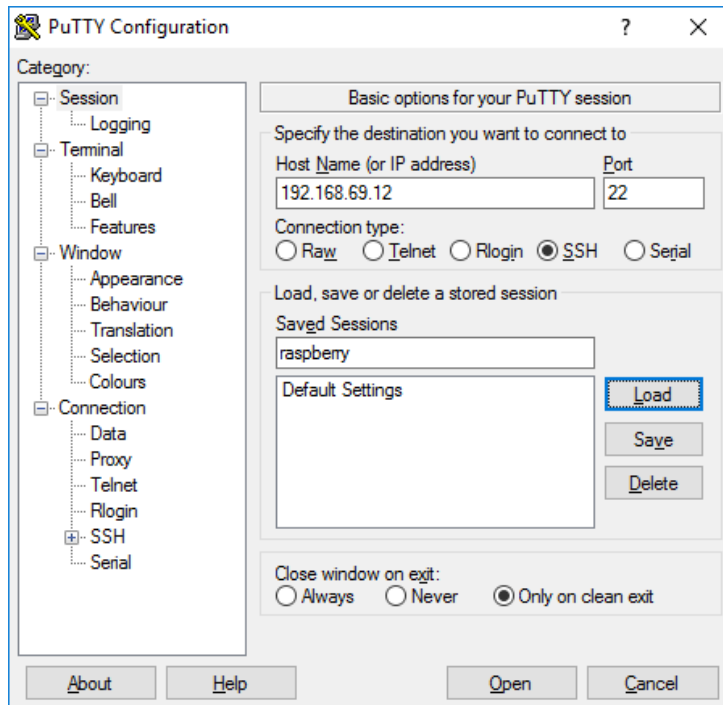


Fig. 4.1.1.3.2. PuTTY

Pentru accesul la distanță, beneficiind de interfața grafică pusă la dispoziție de sistemul de operare, vom utiliza programul VNC. La fel ca și în cazul SSH, este necesară instalarea pe sistemul de pe care se face accesul a unui client specific. Unul dintre cei mai cunoscuți clienți VNC este RealVNC:

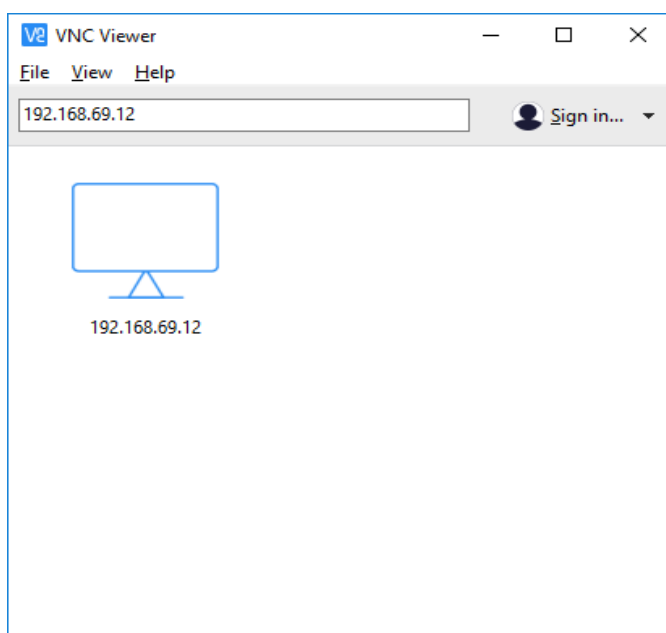


Fig. 4.1.1.3.3. VNC Viewer

### 4.1.2. Camera Video Raspberry Pi

Camera Video Raspberry Pi se folosește pentru a înregistra imagini și sunet la calitate HD, în timp ce poate fi folosită și ca aparat foto, permițând captura unor imagini fixe. Este foarte simplu de utilizat, dar oferă, în același timp, utilizatorilor experimentați o largă aplicabilitate. Sunt foarte multe exemple, în care această camera este folosită pentru înregistrări de tip time-lapse sau slow-motion. Fiind un produs oficial Raspberry Pi, se pot folosi librăriile puse la dispoziție de către fundație, pentru a crea diverse efecte.

Camera se conectează pe placa Raspberry Pi prin panglica specială cu care vine dotată. Mai exact, panglica respectivă se introduce în slot-ul de pe placa Raspberry Pi (situat între porturile Ethernet și HDMI), dedicat camerei.

Specificații:

- Senzor 8 MegaPixeli
- Foto: 3280 x 2464 px
- Formate: 1080p / 720p
- Dimensiuni: 25mm x 23mm x 9mm



Fig. 4.1.2.1. Cameră Raspberry Pi

### 4.1.3. Kit Robot + Motoare Electrice

Kit-ul ales este de fapt o platformă robotică foarte practică, simplă și extrem de flexibilă. Împreună cu cele 4 motoare de curent continuu, robotul (mașinuța) se deplasează având, practic, tracțiune integrală. Datorită roților mari și a faptului că șasiul devine flexibil, abordează foarte bine orice gen de suprafață.

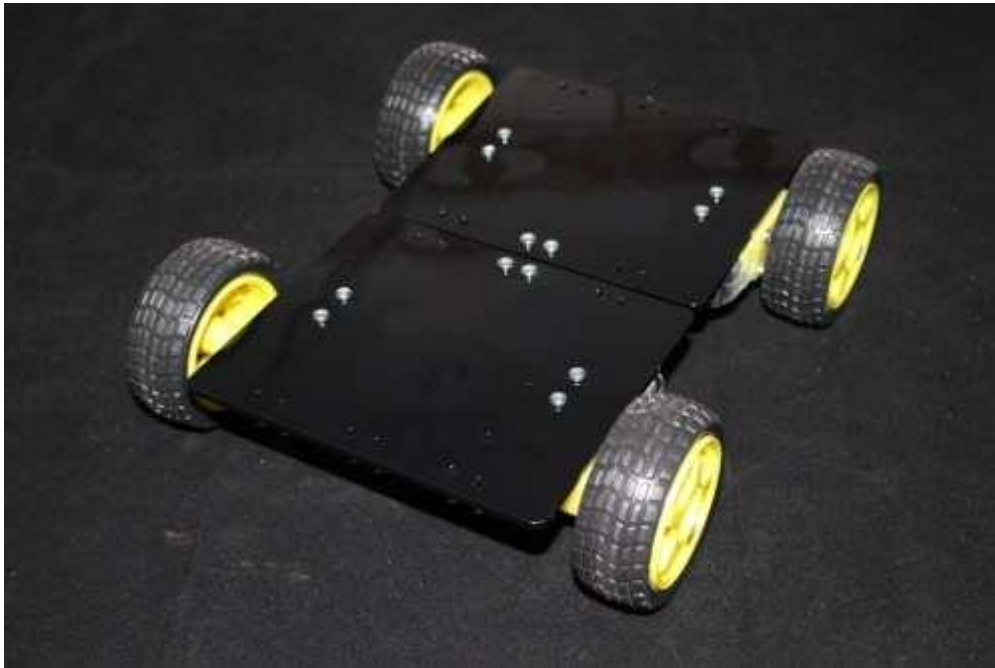


Fig. 4.1.3.1. Kit Robot

Kit-ul se compune din:

- patru motoare cu cutie de viteze
- patru roți din cauciuc
- două șasiuri din plexiglas
- 4 elemente de prindere pentru motoare, printate 3D
- 2 elemente de extindere, printate 3D
- 8 șuruburi M3 x 35 mm (pentru fixarea motoarelor pe suportți)
- 10 șuruburi M3 x 35 mm (8 pentru fixarea suportșilor de motor pe placa de plexiglas, 2 de rezervă)
- 4 șuruburi M3 x 18 mm
- 1 șurub M4 x 50 mm pentru fixarea celor două plăci (prin intermediul elementelor paralelipedice de fixare)
- 24 piulițe M3



Fig. 4.1.3.2. Kit Robot

Motoarele electrice din acest kit sunt motoare electrice de curent continuu, dotate cu câte o cutie de viteze cu raport 1:120 și prezintă următoarele specificații:

- Tensiune nominală: 6 V
- Curent nominal: < 300 mA
- Viteza de rotație: 100 RPM
- Cuplu: 1 kgf/cm



Fig. 4.1.3.3. Motor Electric



#### 4.1.4. ULN2803

În acest proiect se va utiliza un shield ULN2803, pentru a face legătura dintre motoarele de curent continuu și placa Raspberry Pi. Shield-ul acționează, practic, ca un amplificator, deoarece curentul de ieșire a plăcii Raspberry este mult prea mic pentru a acționa motoarele în cauză.

Acest shield permite controlul unor sarcini de putere mai mare, controlate folosind pinii GPIO ai plăcii Raspberry PI. Spre exemplu, se pot controla motoare de putere mică (până în 0.5 A / motor), sau bobinele unor relee.

Consumul general la nivelul plăcii este de 500 mA. Se pot controla până la 8 sarcini independent.

Pentru a controla o sarcină folosind placa ULN2803, se conectează sarcina între unul dintre pinii marcați "5V" și unul dintre pinii de deasupra (marcat "GPIO" urmat de un număr). Pinul marcat "GPIO" funcționează ca un întrerupător conectat între sarcină și GND. Atunci când, din codul sursă se activează pinul GPIO corespunzător, întrerupătorul se închide și permite trecerea curentului, închizând circuitul și alimentând sarcina conectată. Atunci când pinul GPIO este în LOW, întrerupătorul este deschis, curentul nu trece prin sarcină.

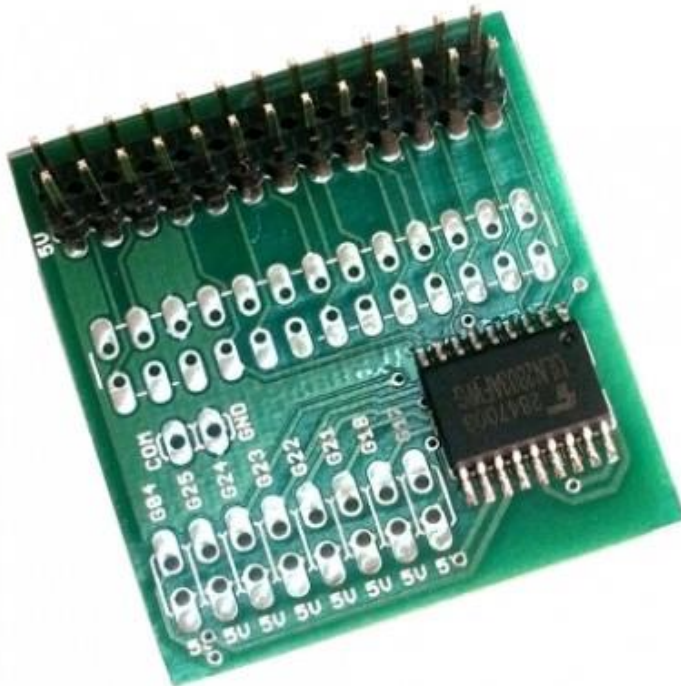


Fig. 4.1.4.1. ULN2803

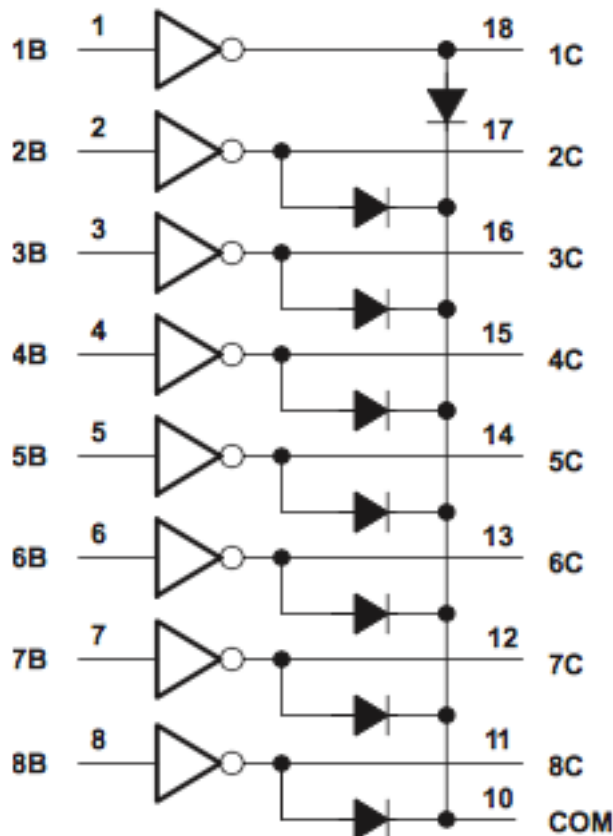


Fig. 4.1.4.2. ULN2803

## 4.1.5. Python

### 4.1.5.1 Generalități

Python este un limbaj de programare orientat pe obiecte, creat de către Guido Van Rossum în anul 1989. Guido Van Rossum lucrează și în ziua de azi la îmbunătățirea limbajului Python, limbaj care este folosit de către companii mari, cum ar fi Google sau Yahoo! pentru programarea aplicațiilor web.

Popularitatea în creștere, dar și puterea limbajului de programare Python au dus la adoptarea sa ca limbaj principal de dezvoltare de către programatori specializați și chiar și la predarea limbajului în unele medii universitare. Din aceleași motive, multe sisteme bazate pe Unix, inclusiv Linux, BSD și Mac OS X includ din start interpretatorul CPython.

Python pune accentul pe curățenia și simplitatea codului, iar sintaxa sa le permite dezvoltatorilor să exprime unele idei programatice într-o manieră mai clară și mai concisă decât în alte limbaje de programare ca C. În ceea ce privește paradigma de programare, Python poate servi ca limbaj pentru software de tipul object-oriented, dar permite și programarea imperativă, funcțională sau procedurală. Sistemul de tipizare este dinamic iar administrarea memoriei decurge automat prin intermediul unui serviciu „gunoier”

(garbage collector). Alt avantaj al limbajului este existența unei ample biblioteci standard de metode.

Implementarea de referință a Python este scrisă în C și poartă deci numele de CPython. Această implementare este software liber și este administrată de fundația Python Software Foundation.

#### 4.1.5.2. Stil

Python este un limbaj multi-paradigmă, concentrându-se asupra programării imperative, orientate pe obiecte și funcționale, ceea ce permite o flexibilitate mai mare în scrierea aplicațiilor. Din punctul de vedere al sintaxei, Python are un număr de construcții și cuvinte cheie cunoscute oricărui programator, dar prezintă și un concept unic: nivelul de indentare are semnificație sintactică. Blocurile de cod sunt delimitate prin simplă indentare.

În C un astfel de blocuri sunt deseori desemnate prin acolade, {<cod>}, dar în Python nu este nevoie de astfel de construcții. Nivelele de indentare îndeplinesc această funcție. Această importanță a indentării este foarte surprinzătoare pentru mulți utilizatori noi ai limbajului Python, chiar dacă sunt programatori cu experiență. Dar o astfel de utilizare a indentării permite codului să fie mai ușor de citit și mai compact. Programatorii cu experiență vor indenta implicit codul sursă, oricare ar fi limbajul, fiindcă astfel se permite structurarea codului sursă și evidențierea funcționalității. Python face din această deprindere folositoare în acest sens o cerință strictă.

O impunere similară există și în limbajul de programare Java, care forțează programatorii să delimiteze clasele în fișiere aparte, din motive de organizare și sporire a eficienței de scriere a softului în echipe.

#### 4.1.5.3. Biblioteci și Extindere

Includerea tuturor acestor structuri, precum și a funcțiilor ce permit manipularea și prelucrarea lor, precum și multe alte biblioteci de funcții sunt prezente datorită conceptului "Batteries Included", ce poate fi explicat prin faptul că Guido van Rossum și comunitatea ce s-a format în jurul limbajului cred că un limbaj de programare nu prezintă utilitate practică dacă nu are un set de biblioteci importante pentru majoritatea dezvoltatorilor.

Din acest motiv Python include biblioteci pentru lucrul cu fișiere, arhive, fișiere XML și un set de biblioteci pentru lucrul cu rețeaua și principalele protocoale de comunicare pe internet (HTTP, Telnet, FTP). Un număr mare de platforme Web sunt construite cu Python. Abilitățile limbajului ca limbaj pentru programarea CGI sunt în afara oricărui dubii. De exemplu YouTube, unul din site-urile cu cea mai amplă cantitate de trafic din lume, este construit pe baza limbajului Python.

Totuși, Python permite extinderea funcționalității prin pachete adiționale programate de terți care sunt axate pe o anumită funcționalitate. De pildă, pachetul wxPython conține metodele și structurile necesare creării unei interfețe grafice.

Popularitatea limbajului este în creștere începând cu anul 2000, datorită faptului că Python permite crearea mai rapidă a aplicațiilor care nu cer viteze înalte de procesare a datelor.

De asemenea este util ca limbaj de scriptare, utilizat în cadrul aplicațiilor scrise în alte limbaje. Modulele (bibliotecile) Python pot fi de asemenea scrise în C, compilate și importate în Python pentru a mări viteza de procesare.



Fig. 4.1.5.3.1. Logo Python

#### 4.1.6. HTML

HyperText Markup Language (HTML) este un limbaj de marcare utilizat pentru crearea paginilor web ce pot fi afișate într-un browser (sau navigator). Scopul HTML este mai degrabă prezentarea informațiilor – paragrafe, fonturi, tabele, etc. – decât descrierea semanticii documentului.

Specificațiile HTML sunt dictate de World Wide Web Consortium (W3C).

HTML este o formă de marcare orientată către prezentarea documentelor text pe o singură pagină, utilizând un software de redare specializat, numit agent utilizator HTML, cel mai bun exemplu de astfel de software fiind browser-ul web. HTML furnizează mijloacele prin care conținutul unui document poate fi adnotat cu diverse tipuri de metadata și indicații de redare. Indicațiile de redare pot varia de la decorațiuni minore ale textului, cum ar fi specificarea faptului că un anumit cuvânt trebuie subliniat sau că o imagine trebuie introdusă, până la scripturi sofisticate, hărți de imagini și formulare. Metadatale pot include informații despre titlul și autorul documentului, informații structurale despre cum este împărțit documentul în diferite segmente, paragrafe, liste, titluri etc. și informații cruciale care permit ca documentul să poată fi legat de alte documente pentru a forma astfel hiperlink-uri (sau web-ul).

HTML este un format text proiectat pentru a putea fi citit și editat de oameni utilizând un editor de text simplu. Totuși scrierea și modificarea paginilor în acest fel solicită cunoștințe solide de HTML și este consumatoare de timp. Editoarele grafice cum ar fi Macromedia Dreamweaver, Adobe GoLive sau Microsoft FrontPage permit ca paginile web să fie tratate asemănător cu documentele Word, dar cu observația că aceste programe generează un cod HTML care este de multe ori de proastă calitate.

HTML se poate genera direct utilizând tehnologii de codare din partea serverului cum ar fi PHP, JSP sau ASP. Multe aplicații ca sistemele de gestionare a conținutului, wiki-uri și forumuri web generează pagini HTML.

HTML este de asemenea utilizat în e-mail. Majoritatea aplicațiilor de e-mail folosesc un editor HTML încorporat pentru compunerea e-mail-urilor și un motor de prezentare a e-mail-urilor de acest tip. Folosirea e-mail-urilor HTML este un subiect controversat și multe liste de mail le blochează intenționat.

Paginile HTML sunt formate din etichete sau tag-uri și au extensia „.html” sau „.htm”. În marea lor majoritate aceste etichete sunt pereche, una de deschidere <eticheta> și alta de închidere </eticheta>, mai există și cazuri în care nu se închid, atunci se folosește <eticheta />. Navigatorul web interpretează aceste etichete afișând rezultatul pe ecran. HTML-ul este un limbaj care nu face deosebire între litere majuscule și minuscule.

Pagina principală a unui domeniu este fișierul „index.html” respectiv „index.htm”. Această pagină este setată a fi afișată automat la vizitarea unui domeniu.



Fig. 4.1.6.1. HTML

## 4.2. Construcția propriu-zisă a robotului

Kit-ul robotului se assemblează în următorii pași:

- Elementele de prindere pentru motoare se fixează de șasiurile din plexiglas cu 8 șuruburi M3 x 35 mm și 8 piulițe M3.

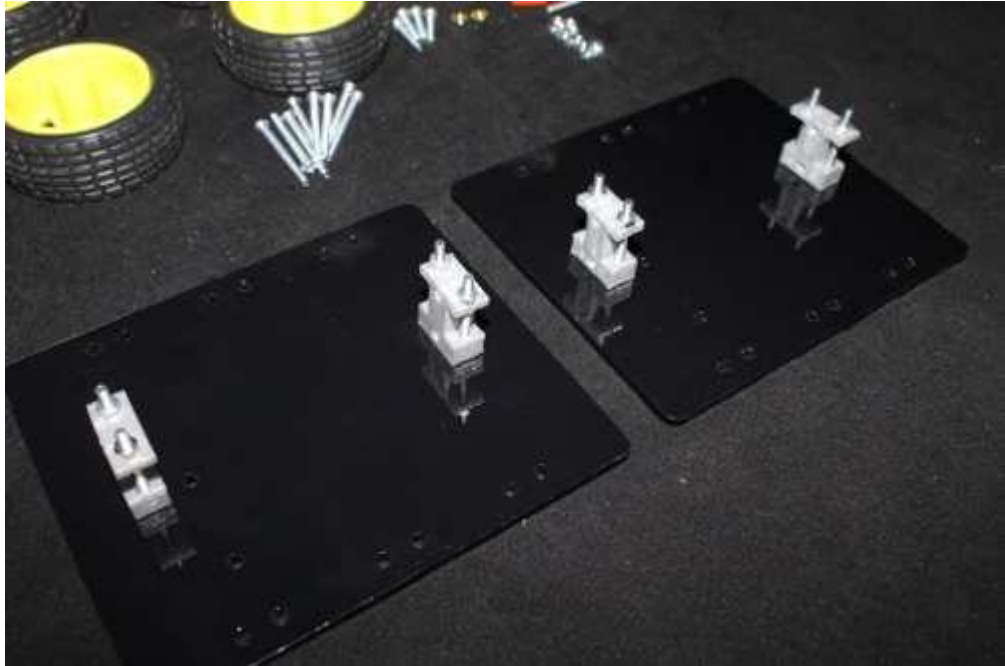


Fig. 4.2.1. Pas1

- Motoarele cu reductor se fixează cu 8 șuruburi M3 x 35 mm (câte 2 pentru fiecare motor).

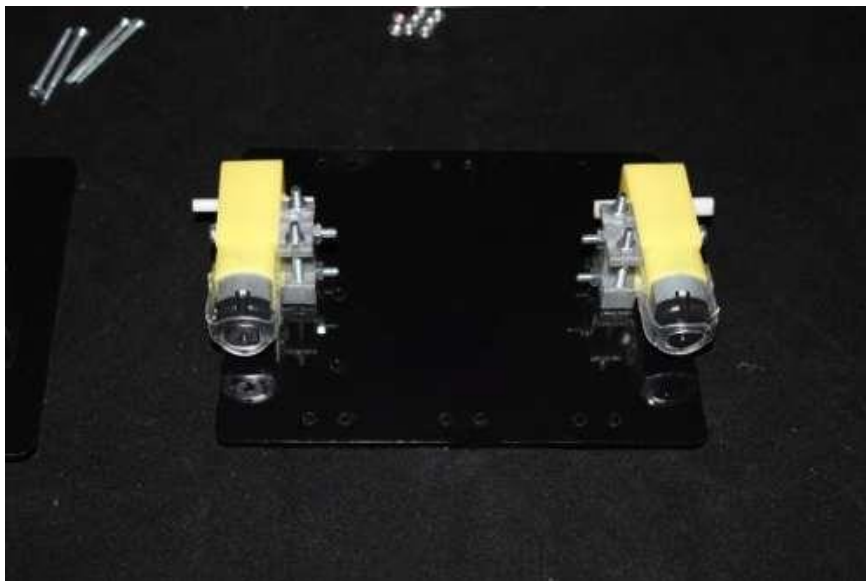


Fig. 4.2.2. Pas2



Fig. 4.2.2. Pas2b

- Elementele de extindere pentru plăcile din plexiglas se fixează cu 4 șuruburi M3 x 18 mm.



Fig. 4.2.3. Pas3

- Cele două șasiuri din plexiglas se atașează împreună printr-un șurub M4 x 45 mm.

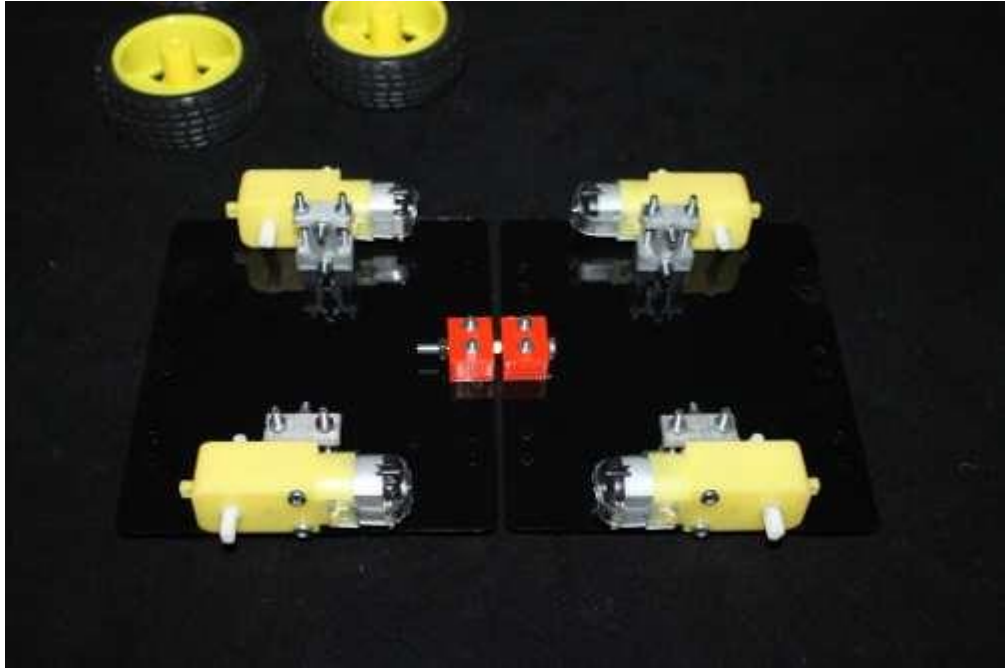


Fig. 4.2.4. Pas4

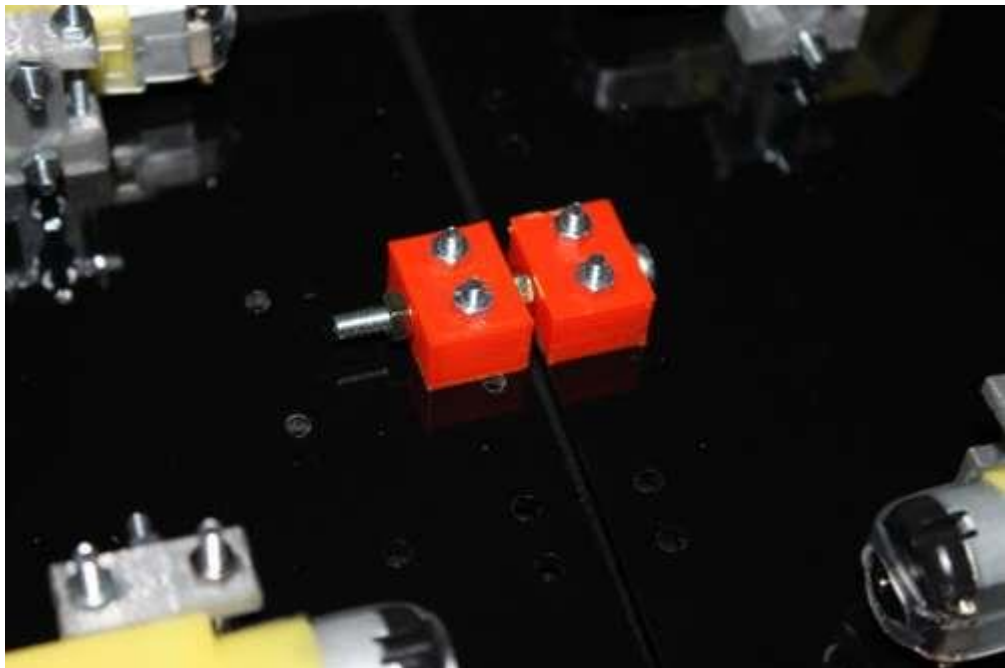


Fig. 4.2.5. Pas4b



- Se montează roțile.



*Fig. 4.2.6. Pas5*

- În final, platforma Flexybot va arăta ca în imaginile de mai jos.



*Fig. 4.2.7. Pas6*

După asamblarea kit-ului, se montează placa Raspberry Pi, camera Raspberry Pi, bateria externă, shield-ul ULN2803 (direct pe placa Raspberry Pi) și se fac conexiunile între motoare și shield.

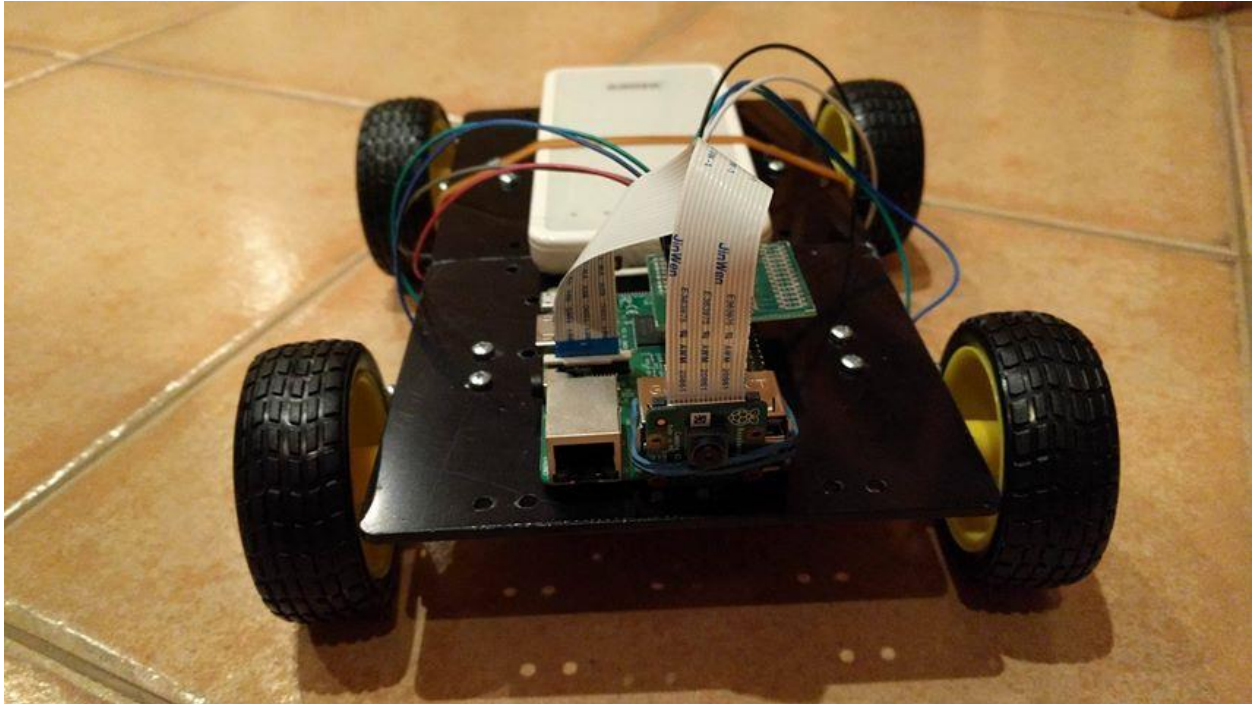


Fig. 4.2.8. Robot Mobil



Fig. 4.2.9. Robot Mobil

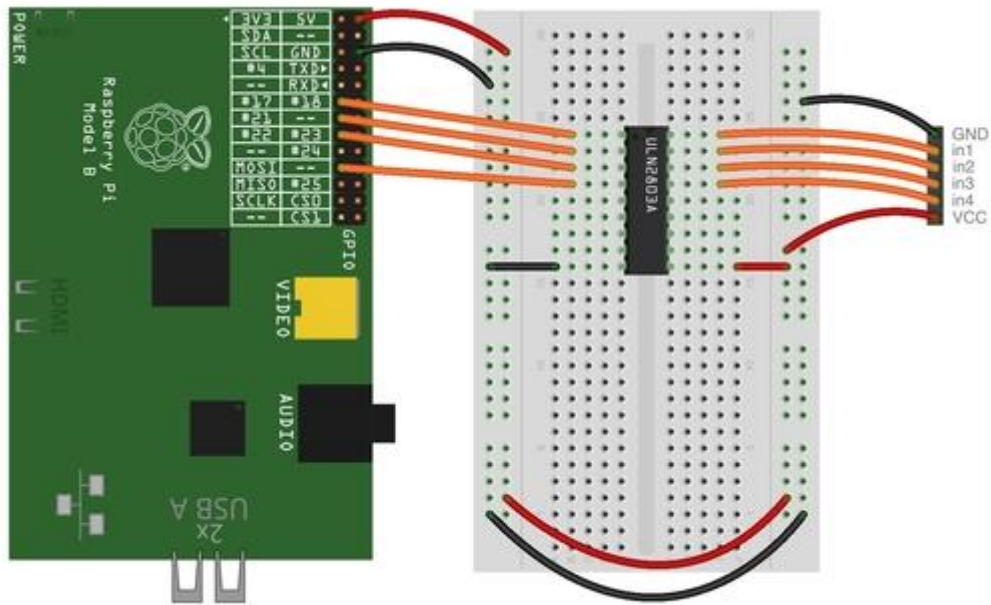


Fig. 4.2.10. Conexiune Raspberry-ULN2803

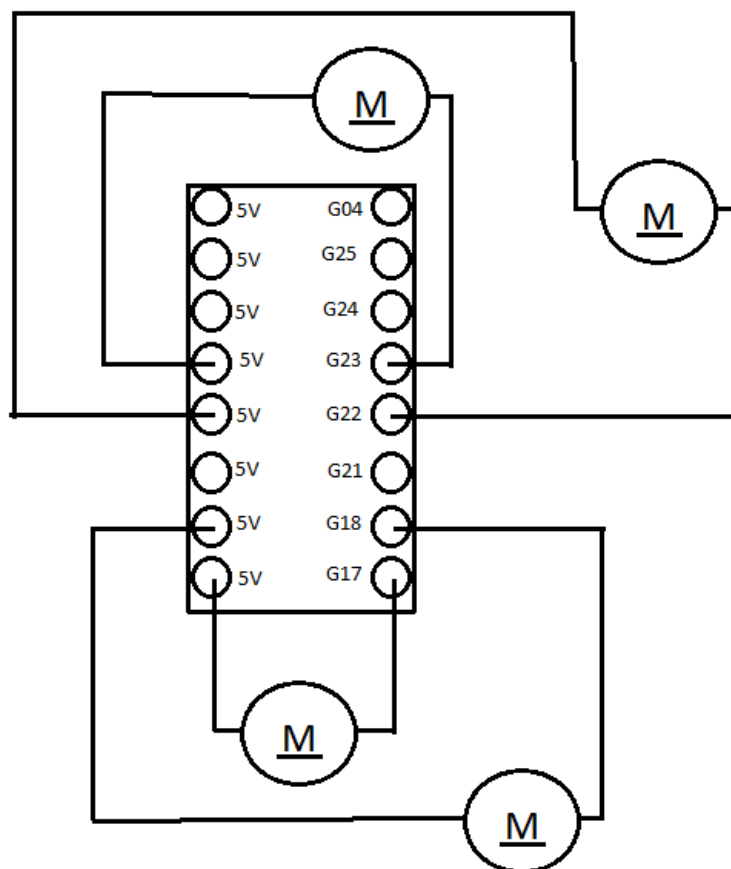


Fig. 4.2.11. Conexiune Motoare-ULN2803

## 5. FUNCȚIONARE

Robotul mobil se deplasează cu ajutorul motoarelor de curent continuu. Motoarele primesc curent de la shield-ul ULN2803, care la rândul său este conectat la placa Raspberry Pi. Când unul dintre butoanele de pe aplicația web este apăsat, robotul mobil se deplasează în direcția aferentă butonului. Cât timp butonul este ținut apăsat, robotul își continuă deplasarea, iar când butonul este eliberat, robotul se oprește.

Fiind o aplicație web, nu are importanță dispozitivul de pe care este accesată aplicația, atâta timp, cât dispozitivul respectiv are o conexiune la internet și se regăsește în aceeași rețea cu Raspberry-ul.

Cărui motor și când anume să transmită placa Raspberry Pi curent, este stabilit în codul sursă.

Pentru început se importă librăriile necesare:

```
from flask import Flask, render_template, Response
from picamera import PiCamera
from io import BytesIO
from time import sleep
from robo_car import *
```

Cu aceste comenzi din aplicația principală definim rezoluția camerei și setăm pinii pentru controlul motoarelor ca fiind 17, 18, 22 și 23:

```
camera.resolution = (500, 400)
buffer = BytesIO()
car = Car(17, 18, 22, 23)
```

În aplicația robo\_car.py este definită o clasă denumită Car, care definește mișcarile robotului:

```
class Car:
    def __init__(self, left_down, left_up, right_down, right_up):
        self.left_down = left_down
        self.left_up = left_up
        self.right_down = right_down
        self.right_up = right_up
        gpio.setmode(gpio.BCM)
```

```
self._init_motors()

def forwards(self):
    gpio.output([self.right_up, self.left_up], True)
    sleep(0.09)
    gpio.output([self.right_up, self.left_up], False)

def backwards(self):
    gpio.output([self.left_down, self.right_down], True)
    sleep(0.09)
    gpio.output([self.left_down, self.right_down], False)

def turn_left(self):
    gpio.output([self.right_up, self.left_down], True)
    sleep(0.09)
    gpio.output([self.right_up, self.left_down], False)

def turn_right(self):
    gpio.output([self.left_up, self.right_down], True)
    sleep(0.09)
    gpio.output([self.left_up, self.right_down], False)

def _init_motors(self):
    self._pins
    [self.left_down, self.left_up, self.right_down, self.right_up] =
    gpio.setup(self._pins, gpio.OUT)

def cleanup(self):
    gpio.cleanup(self._pins)
```

În fișierul index.html se regăsește codul pentru poziționarea în pagină a ferestrei cu redarea video de la camera Raspberry Pi și a butoanelor de comandă:

```
video_stream {

position:fixed;
margin-top:50px;
margin-left:200px;

}

.ctrl_buttons {

position:fixed;
margin-top:130px;
margin-left:750px;
width:400px;
height:400px;
border: 2px solid black;

}

#btn_up {

position:absolute;
margin-left:150px;
width:100px;
height:100px;
font-size:20px;

}
```

```
#btn_down {  
  
position:absolute;  
left:150px;  
bottom:0;  
width:100px;  
height:100px;  
font-size:20px;  
  
}
```

```
#btn_left {  
  
position:absolute;  
left:0px;  
top:150px;  
width:100px;  
height:100px;  
font-size:20px;  
  
}
```

```
#btn_right {  
position:absolute;  
right:0px;  
top:150px;  
width:100px;  
height:100px;  
font-size:20px;
```

Întregul cod sursă, cât și aplicațiile care fac posibilă comanda robotului mobil se regăsesc în Anexa 1.

## 6. CONCLUZII

S-a realizat un robot mobil comandat de la distanță, capabil să preia informații dintr-un mediu virtual, transformând aceste informații în mișcări fizice.

Folosindu-se de un IP privat, din rețeaua locală la care este conectat Raspberry-ul, aplicabilitatea acestui robot se rezumă doar la dispozitivele (telefoane, laptop-uri, tablete) conectate la aceeași rețea. Dar, cum a fost specificat și în capitolul "Viitoare Îmbunătățiri", aplicabilitatea robotului se poate mări foarte ușor cu achiziția unui IP public sau a unui domeniu public.

Mișcările robotului sunt limitate de citirea stării butonului apăsat de mai multe ori pe secundă, iar acest lucru face ca mișcarea robotului să nu pară foarte fluentă.



---

## BIBLIOGRAFIE

Dolga, V., "Proiectarea Sistemelor Mecatronice", Editura Politehnica, Timișoara, ISBN 978-973-625-573-1, 2007.

Maniu, I., Dolga, V., Ciupe, V. Bogdanov, I. Rădulescu, C., Varga, Ș., "Robotică. Sistemul de acționare", Vol.II, Editura Politehnica, ISBN: 978-973-625-996-8, 2007.

Rădulescu C., Vatau, S., "Roboți Mobili - Vehicule Ghidate Automat", Vol.1, Editura Politehnica, ISBN 978-973-625-664-6, 2008.

Allen B. Downey, "Think Python: An Introduction to Software Design", 2002.

[http://www.mecatronica.ro/definitie\\_mecatronica.html](http://www.mecatronica.ro/definitie_mecatronica.html)

<http://www.scientia.ro/stiinta-la-minut/istoria-ideilor-si-descoperirilor-stiintifice/2500-introducere-in-istoria-roboticii.html>

<http://www.rumaniamilitary.ro/sistemele-de-avioane-fara-pilot-in-romania>

<https://playtech.ro/2014/noul-robot-umanoid-honda-asimo-tinde-sa-fie-de-un-real-ajutor/>

<http://webbut.unitbv.ro/teze/rezumat/2011/rom/DuguleanaMihai.pdf>

[http://www.imst.pub.ro/Upload/Studenti/SSS\\_2016/lucrarile\\_sesiunii\\_stud\\_2016/IMPLEMENTAREA\\_ROBOTILOR\\_INDUSTRIALI.pdf](http://www.imst.pub.ro/Upload/Studenti/SSS_2016/lucrarile_sesiunii_stud_2016/IMPLEMENTAREA_ROBOTILOR_INDUSTRIALI.pdf)

<http://www.descopera.ro/stiinta/15616273-cercetatorii-de-la-harvard-au-fabricat-primul-robot-flexibil-ce-animal-a-inspirat-crearea-lui-video>

<https://www.raspberrypi.org/documentation/usage/python/>

<http://python.net/~gherman/propaganda/tut-ro/tut-ro.pdf>

<https://www.python.org/>

<https://ro.wikipedia.org/>

<https://www.robofun.ro/>

<https://www.w3.org/TR/html5/>

<http://www.drogoreanu.ro/tutorials/html.php>

<https://tutorialehtml.com/ro/introducere-in-html/>

---

## ANEXA 1

```
app.py
{
from flask import Flask,render_template,Response
from picamera import PiCamera
from io import BytesIO
from time import sleep
from robo_car import *

app = Flask(__name__)
camera = PiCamera()
camera.hflip = True
camera.vflip = True
camera.resolution = (500, 400)
buffer = BytesIO()
car = Car(17,18,22,23)

def move_car(car,btnid):
    if btnid == "btn_up":
        car.forwards()
    elif btnid == "btn_down":
        car.backwards()
    elif btnid == "btn_left":
        car.turn_left()
    elif btnid == "btn_right":
        car.turn_right()

def gen(camera,buffer):
    for frame in
camera.capture_continuous(buffer,format='jpeg',use_video_port=True):
```

```
        buffer.seek(0)
        yield (b'--frame\r\n'b'Content-Type: image/png\r\n\r\n'
+ buffer.read() + b'\r\n')
        buffer.seek(0)
        buffer.truncate()

@app.route("/")
def index():
    return render_template("index.html")

@app.route("/camera_stream")
def camera_stream():
    return Response(gen(camera,buffer),mimetype='multipart/x-
mixed-replace; boundary=frame')

@app.route("/btn/<id>",methods=["POST"])
def btn(id):
    print(id);
    move_car(car,id)
    return id;

if __name__ == "__main__":
    app.run(host='0.0.0.0', threaded=True)

}
```

```
robo_car.py
{
import RPi.GPIO as gpio
from time import sleep

class Car:
    def __init__(self, left_down, left_up, right_down, right_up):
        self.left_down = left_down
        self.left_up = left_up
        self.right_down = right_down
        self.right_up = right_up
        gpio.setmode(gpio.BCM)
        self._init_motors()

    def forwards(self):
        gpio.output([self.right_up, self.left_up], True)
        sleep(0.09)
        gpio.output([self.right_up, self.left_up], False)

    def backwards(self):
        gpio.output([self.left_down, self.right_down], True)
        sleep(0.09)
        gpio.output([self.left_down, self.right_down], False)

    def turn_left(self):
        gpio.output([self.right_up, self.left_down], True)
        sleep(0.09)
        gpio.output([self.right_up, self.left_down], False)

    def turn_right(self):
        gpio.output([self.left_up, self.right_down], True)
        sleep(0.09)
```

```
        gpio.output([self.left_up,self.right_down],False)

    def _init_motors(self):
        self._pins =
[self.left_down,self.left_up,self.right_down,self.right_up]
        gpio.setup(self._pins,gpio.OUT)

    def cleanup(self):
        gpio.cleanup(self._pins)

if __name__ == "__main__":
    pins = [17,18,22,23]
    car = Car(17,18,22,23)
    car.forwards()
    car.backwards()
    car.turn_left()
    car.turn_right()
    car.cleanup()
}
```

```
index.html
{
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>TestWebPage</title>

<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.m
in.js"></script>

<style>

.video_stream {

position:fixed;
margin-top:50px;
margin-left:200px;

}

.ctrl_buttons {

position:fixed;
margin-top:130px;
margin-left:750px;
width:400px;
height:400px;
border: 2px solid black;
```

```
}
```

```
#btn_up {
```

```
position:absolute;
```

```
margin-left:150px;
```

```
width:100px;
```

```
height:100px;
```

```
font-size:20px;
```

```
}
```

```
#btn_down {
```

```
position:absolute;
```

```
left:150px;
```

```
bottom:0;
```

```
width:100px;
```

```
height:100px;
```

```
font-size:20px;
```

```
}
```

```
#btn_left {
```

```
position:absolute;
```

```
left:0px;
```

```
top:150px;
```

```
width:100px;
```

```
height:100px;
```

```
font-size:20px;
```

```
}
```



```
#btn_right {  
  
position:absolute;  
right:0px;  
top:150px;  
width:100px;  
height:100px;  
font-size:20px;  
  
}  
  
</style>  
  
</head>  
<body>  
  
<div class='ctrl_buttons'>  
  
<button type='button' id = "btn_up">  
Forward  
</button>  
  
<button type='button' id = "btn_down">  
Back  
</button>  
  
<button type='button' id = "btn_left">  
Left  
</button>  
  
<button type='button' id = "btn_right">  
Right
```

```
</button>
```

```
</div>
```

```
<div class='video_stream'>
```

```
<h1>Video Stream</h1>
```

```

```

```
</div>
```

```
<script>
```

```
var _init = function() {
```

```
    let btns = $(".ctrl_buttons").children();
```

```
    let timeout;
```

```
    let btn_names = ["up", "down", "left", "right"];
```

```
    btn_names.forEach(function(name) {
```

```
        let btn = $("#btn_"+name)
```

```
        let btn_id = btn.attr('id');
```

```
        btn.mousedown(function() {
```

```
            timeout = setInterval(function() {
```

```
                $.ajax({
```

```
                    url: '/btn/'+btn_id,
```

```
                    success: function(data) {
```

```
                        console.log(data);
```

```
                    },
```

```
                    type: 'POST'
```

```
                });
```

```
            }, 100);
```

```
});  
  
    btn.mouseup(function() {  
        clearInterval(timeout);  
    });  
  
});  
  
}  
  
_init();  
  
</script>  
  
</body>  
</html>  
}
```