Debreceni Egyetem
Műszaki Kar
Villamosmérnöki és Mechatronikai Tanszék

Parallax Penguin robot fejlesztése és programozása PBASIC nyelven

Témavezető:
Dr. Husi Géza
Tanszékvezető főiskolai docens

Készítette:
Ujvárosi Tamás János
Mérnök Informatikus BSc hallgató

Debrecen
2010

Tartalomjegyzék

„Minden igazságot könnyű megérteni, ha már felfedezték;

a cél felfedezni őket.”

Galileo Galilei

Szakdolgozatom témaválasztása

Harmad- és negyedéves, szakirányos tanulmányaim során egyre sűrűbben nyílt lehetőségem megismerkedni a robotprogramozással, illetve a robotikával a Debreceni Egyetem Műszaki Karán. Szakdolgozatom témaválasztása végül egy lépegető szerkezetre, a Parallax Penguin robotra esett a választható eszközök, illetve témák közül. A széleskörű választási lehetőség is jól jelzi, hogy a Debreceni Egyetemnek is hosszú távú célja a minél átfogóbb, magas színvonalú képzés biztosítása.

A hétköznapi életben is minden ember számára fontos, hogy megtalálja a megfelelő utat, amin céljait elérve végigmehet. A mindennapok során rövid és hosszú távú céljaink elérésékor akadályokba ütközünk, amiket észre kell venni és feloldani vagy elkerülni, hogy folytatni tudjuk célkitűzéseink megvalósítását. A céltalan kóválygás az emberek számára is kínzó és hasztalan állapot. Ugyanilyen fontos a robotoknál is, hogy legyen valamilyen céljuk, feladatuk, amit végrehajthatnak.

Szakdolgozatom és az azt megelőző kutatási munkám célkitűzése olyan mobil robot navigációs feladatok bemutatása és gyakorlati alkalmazása volt, amelyek kivitelezhetőek az általam választott Parallax Penguin robot segítségével.

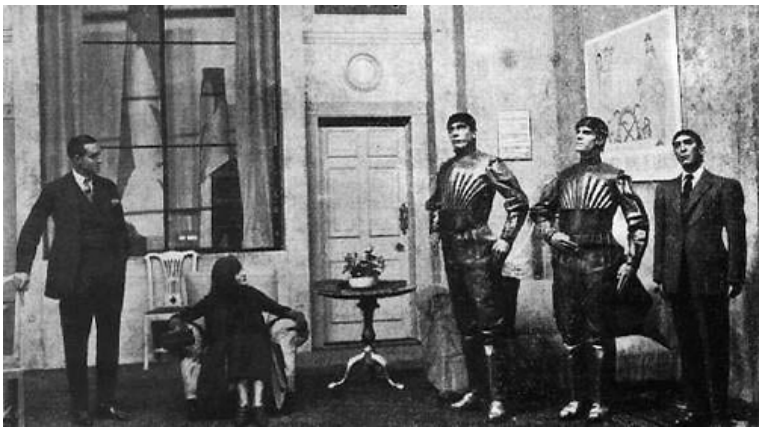
Robotika

A robot fogalma

A hétköznapi ember számára a robot kifejezés hallatán emberi alakot öltött beszélni és járni tudó intelligens gépek, - amelyek a sci-fi filmek világából csöppentek a való életbe - vagy a háztartási robotok jutnak az eszébe.

Mi alapján dönthetjük el, hogy valójában mi is az a robot? Megvizsgálhatjuk definíció szerint, de - mint ez majd lentebb látható -, a szakirodalom sem egységes a fogalommal kapcsolatban.

Először is a szó eredetét nézzük meg. A robot szó a szláv robota, magyarul a munka, szolgaság szavakból ered. A munka szóhoz társítva a fizikai munkavégzés juthat eszünkbe, a szolgaságról már nem is beszélve. A robot kifejezést először 1921-ben Karel Čapek cseh író használta Rossum Univerzális Robotjai (R.U.R) című utópisztikus drámájában, melyben az emberi felépítésű automaták fellázadtak az emberiség ellen és átvették az uralmat a Föld felett. [1.]



1. ábra, Jelenet a Rossum Univerzális Robotjai című drámából, 1921 (forrás [1.])

A robot definícióját a világon többféleképpen adja meg a szakirodalom. Néhány példa a robot definíciójára:

A Merriam Webster Online szótár három definíciót is ad a robot szóra.

- Az első szerint ***„egy gép, amely embernek néz ki és különféle összetett emberi cselekvéseket hajt végre (mint a járás vagy a beszéd)”***,
- a második alapján ***„egy eszköz, amely bonyolult és gyakran ismételt feladatok hajt végre”***,
- és a harmadik szerint ***„egy mechanizmus, amit automatikus vezérlés irányít”***. [2.]

Az International Organization for Standardization (Nemzetközi Szabványosítási Szervezet-ISO) 1994-ben a robotra adott definíciója az ISO 8373 számú szabvány:

- ***„egy automatikus irányítású, újraprogramozható, többcélú automatikus ipari feladatok elvégzésére használt manipulátor, három vagy több tengelye programozható, amik rögzítettek vagy mobilak lehetnek”*** [3.]

És végül egy érdekes meghatározás, a robotika atyjának is nevezett, az ipari robotok terén úttörő Joseph Engelberger-től, amikor azt kérdezték tőle, hogyan definiálja a robotot:

- ***„Nem tudom definiálni a robotot, de felismerem, ha látok egyet.”*** [4.]

2. ábra, Joseph F. Engelberger (forrás [12.])

Látható, hogy nincs egységes definíció arra, mi is a robot, azonban vannak bennük azonosságok.

- A robot automatikus, vagyis félig vagy teljesen önmagától végzi feladatait.
- A robot valamilyen mozgást végez és/ezért mechanikai, mozgó alkatrészeket, elemeket tartalmaz, és a mozgásban több szabadságfokkal rendelkezik
- A robot leghasznosabban ipari feladatok elvégzésére hasznosítható.

Észrevehető, hogy sok szerkezet nem tekinthető robotnak, például amelyek, csak egyetlen műveletet ismételgetnek, vagy teljesen külső irányítás alatt állnak.

Érdemes megvizsgálni, hogy egy robot valóban csak az előre megadott feladatokat végzi el, vagy figyel és reagál környezetére. Esetleg önállóan mérlegel a megtehető lehetőségek között vagy akár önállóan képes tanulni is? Sokkal ügyesebbek (és bonyolultabbak) és kevesebb külső beavatkozást igényelnek azok a robotok, amelyek különböző információkat gyűjtenek környezetük állapotáról, a feldolgozott információkat átszűrik, és ezek alapján hoznak döntéseket. A robotgenerációk fejezetben a különbségek később jól láthatóak lesznek.

A robotok általános feladatai

A robotok rendszerint olyan munkákat, feladatokat végeznek el,

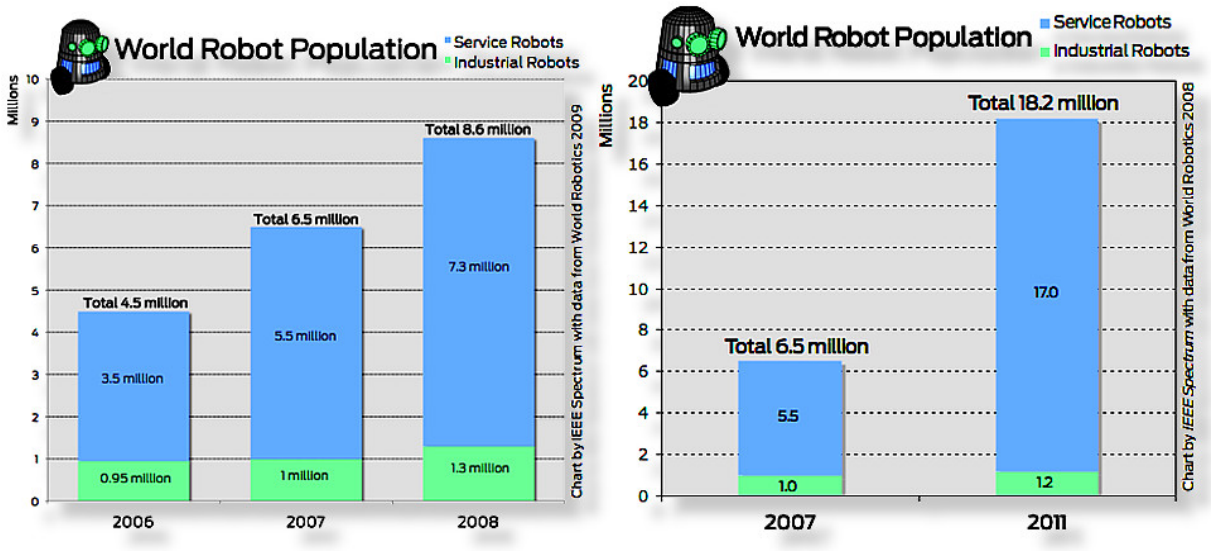
- ***amelyek túl veszélyesek vagy piszkosak***
- ***vagy napi 24 órában kell végezni***
- ***vagy túl nehezek az ember számára***
- ***vagy egyszerűen túl monoton, de nagy pontossággal végrehajtandó feladatok,***

ezért egy robot sokkal nagyobb biztonsággal képes elvégezni, mint az emberek. [5.]

A robotika

A robotika a robotokkal foglalkozó mérnöki tudományág és egyben gazdasági iparág, beleértve tervezésüket, gyártásukat és alkalmazásukat. [7.] A robotikához tartozik minden eddigi, a robotokkal kapcsolatban elért irodalmi, kutatási és fejlesztési eredmény és mind az a tudáshalmaz, amelynek köszönhetően alkalmazásuk mára széleskörűen elterjedt. A robotika fejlődése olyan nagy neveknek köszönhető, mint George Devol, Joseph F. Engelberger, Victor Scheinman vagy Hugh F. Durrant-Whyte.

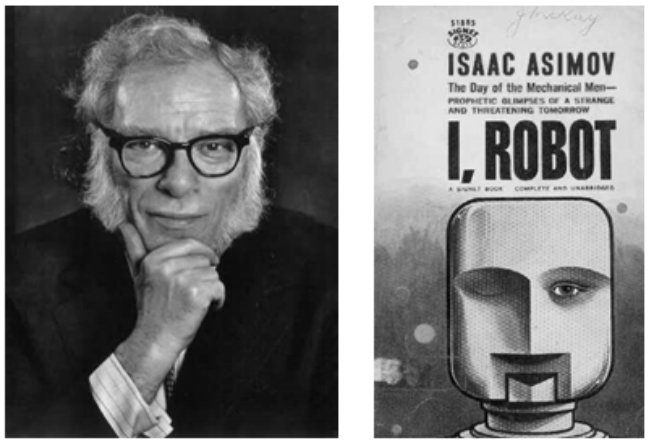
A robotika fejlődésének köszönhetően és az IFR (International Federation of Robotics), 2009. évi World Robotics kiadványának adatai alapján a világ robot állománya 2006-ban elérte a 4,5 milliós, 2007-ban a 6,5 milliós és 2008-ben a 8,6 milliós darabszámot. A 2008. évi adatok szerint a 8,6 millió robotból megközelítőleg 1,3 millió volt az ipari és 7,3 millió a szolgáltató robot. Becslések szerint 2011-re 18,2 millióra növekszik a világon használatban lévő robotok száma. [30.]



3. ábra, A világ robot populációja: balra konkrét adatok, jobbra prognózis a 2011. évre (forrás [30.])

A robotika kifejezés Isaac Asimov orosz-amerikai sci-fi írótól ered. A szó megalkotásakor még ő sem tudott róla, hogy ezzel egy új, korszakalkotó szót talált ki.

Asimov emellett megfogalmazott a robotok számára három fontos szabályt, alaptörvényt, amelyeket műveiben a robotoknak mindig be kellene tartaniuk. Asimov nem egyszerre fogalmazta meg ezeket a törvényeket, hanem köteteiben, novelláiban folyamatosan és tudatosan jelentek meg - miközben volt olyan, amelyikből kimaradt -, míg mindhármat először együtt az Én, a robot című kötetben található Körbe-körbe (1941) című novellájában olvashatták a sci-fi rajongók. Ezeket nevezzük a robotika három alaptörvényének.



4. ábra, Isaac Asimov, 1920-1992 és az Én, a robot című novellás kötetének borítója, 1950 (forrás [15.])

A robotika három alaptörvénye:

1. A robotnak nem szabad kárt okoznia emberi lényben, vagy tétlenül tűrnie, hogy emberi lény bármilyen kárt szenvedjen.
2. A robot engedelmeskedni tartozik az emberi lények utasításainak, kivéve, ha ezek az utasítások az első törvény előírásaiba ütköznének.
3. A robot tartozik saját védelméről gondoskodni, amennyiben ez nem ütközik az első vagy második törvény bármelyikének előírásaiba. [9.]

Asimov korában a robotika hívői hittek abban, hogy már csak pár év választja el őket az intelligens robotoktól, azonban az emberi gondolkodáshoz hasonlóval rendelkező robotok fejlesztése még napjainkban is eléggé gyerekcipőben jár. Érdemes azonban elgondolkodni azon, hogy a jövőben még jobban elterjedő, mesterséges intelligenciával rendelkező robotok számára ezek akár valóságos törvények is lehetnek.

Az ipari robotok

Látható, hogy nagy számban alkalmaznak robotokat az iparban, ezért mindenképpen meg kell említenem az ipari robotokat. Nagyon sok és sokféle munkát végeznek az emberi erőforrás helyett, ráadásul olyanokat is, amelyekre ember már képtelen lenne, mint például az

autógyártást óriási mennyiségben és minőségben vagy a mikroelektronikai áramkörök precíziós összeszerelését, amelyek gyakran nanométeres méretű komponensekből épülnek fel. Jelenleg olyan neves cégek gyártanak ipari robotokat, mint például az amerikai Adept Technology, a japán FANUC Ltd és a Kawasaki Heavy Industries Ltd, a német KUKA Roboter GmbH, a francia Stäubli Robotics vagy a svájci BlueBotics SA.

Az ipari robot definíciója: Az ipari robot olyan mechatronikai (mechanikai-elektromos-elektronikai) szerkezet, amely:

- ***nyílt kinematikai láncú mechanizmust,***
- ***intelligens vezérlést tartalmaz,***
- ***irányított mozgásokra képes,***
- ***automatikus működésre képes és***
- ***előírt, programozható feladatokat végez.*** [6.]

5. ábra, Egy IBM 7576 Scara robotkar, ami - az egyetemen is oktatott - AML nyelven programozható (forrás [14.])

A szolgáltató robotok

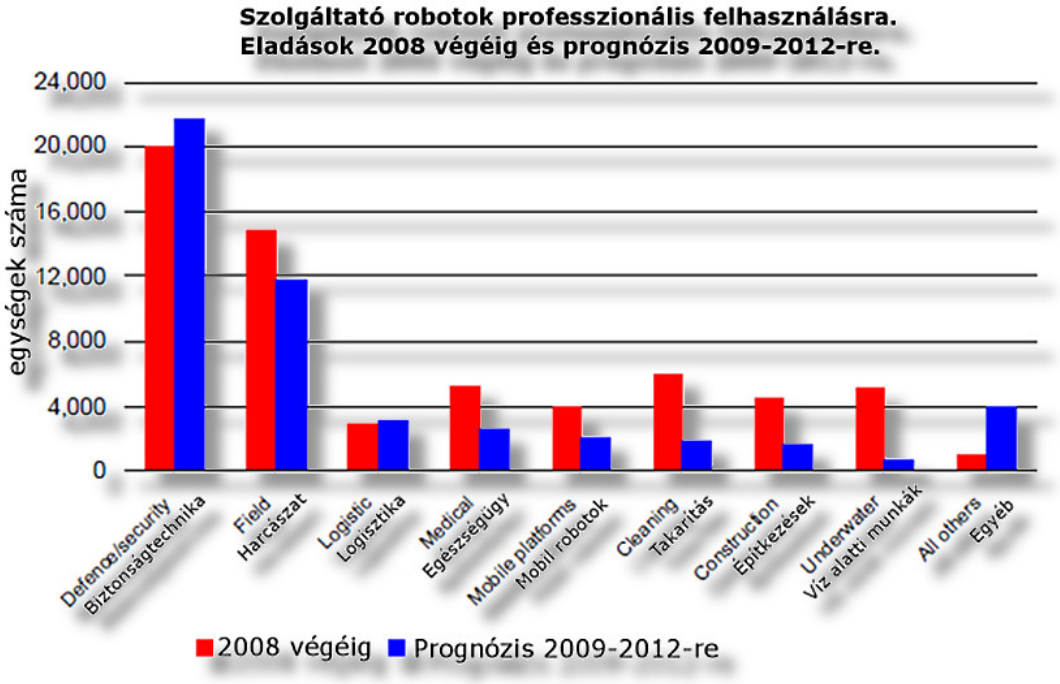
Manapság a robotok számának nagyobb hányadát az úgynevezett szolgáltató robotok teszik ki. A szolgáltató robotok feladata leggyakrabban olyan tevékenységekben az emberek segítése, mint a harcászat és a biztonságtechnika, de emellett alkalmazzák az egészségügyben, a mezőgazdaságban, a logisztikában, a víz alatti és takarítási munkák során és az építőiparban. Az ilyen szolgáltatásokat nyújtó robotokat professzionális szolgáltató robotoknak nevezik.

A szolgáltató robotok másik kategóriáját a háztartási robotok széles csoportja alkotja. A háztartási alkalmazások száma többszöröse a professzionális robotokéhoz képest. Az otthoni robotok feladata a különböző takarítási és kerti munkák mellett a szórakoztatás és újabban az idős és beteg emberek egyszerűbb szükségleteinek kiszolgálása. (Ez leginkább a robotok hazájának számító Japánban terjed.) A szórakoztató robotok közül a legnépszerűbbek a robot házi állatok, illetve az emberi viselkedést utánzó, beszélő, sétáló vagy táncoló humanoid robotok.

A szolgáltató robotok definíciója: A szolgáltató robotok olyan robotok,

- ***amelyek félig vagy teljesen önállóan képesek hasznos szolgáltatásokat nyújtani az embereknek, kivéve a gyártási műveleteket.*** [13.]

A szolgáltató robotok definíciója láthatóan átfedésben van az ipari robotokéval, hiszen léteznek nem gyártási műveletek végző ipari robotok is. Leginkább oly módon különböztethetőek meg az ipari robotoktól, hogy legtöbbször mobil felépítésűek, ritkán rendelkeznek robotkarokkal és csak ritkán vannak rögzítve feladataik elvégzésére. Többek között a Fraunhofer, a Fujitsu Laboratories, a Husqvarna, az iRobot, a Toyota, és a Sony gyárt különböző szolgáltató robotokat.



6. ábra, A professzionális szolgáltató robotok eladásai és előrejelzései tevékenységek szerinti eloszlásban (forrás [18.])

Robotgenerációk

Napjainkban, a robotikában három nagyobb robot generációt különböztethetünk meg egymástól. Mind a három generáció valamilyen forradalmi változást hozott.

Az első generációs robotok az 1960-s években jelentek meg, főleg mozgatási feladatokat végeztek, például rakodást, emelést vagy hegesztést. Egy előre megadott utasítássorozat alapján dolgoztak, környezetük jeleit nem tudták érzékelni így a rendkívüli változásokra sem tudtak reagálni.

A legtöbb első generációs robot rögzített, nehéz szerkezet volt. Alkalmazása az iparban terjedt el, mert az ember számára nehéz és veszélyes fizikai munkákat el tudták végezni. Jelenleg az iparban használt - például - emelő, festő, rakodó vagy hegesztő robotok közül egy részük felépítése és vezérlése alapján még most is az első generációs robotok szintjén van.

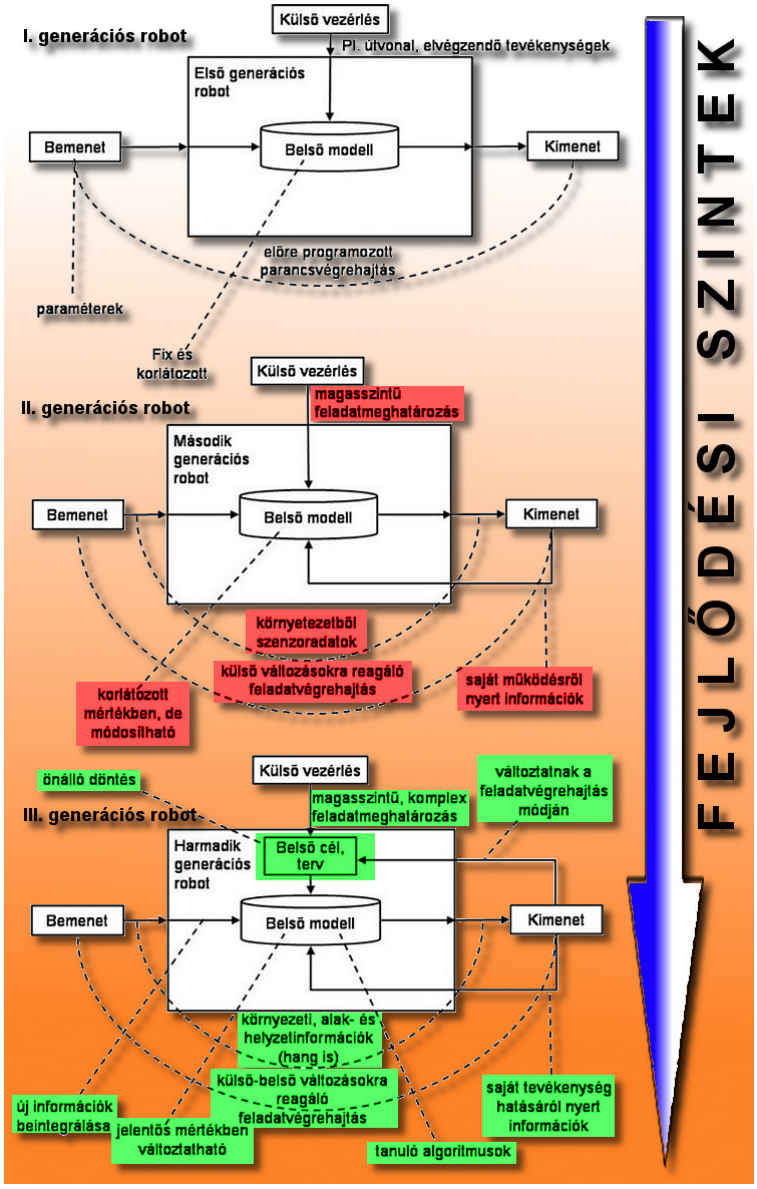
Az 1970-s évek technológiai fejlődésének köszönhetően a robotokat már felszerelték a környezet jeleinek észlelésére alkalmas érzékelőkkel, és ezeknek köszönhetően képessé tették őket viselkedésük korlátozott változtatására. Ezeket nevezzük második generációs robotoknak.

Az érzékelővel szerelt robotok változtatni tudják tevékenységüket az érzékelt információk alapján. A második generációs robotok másik újítása a magas szintű programozási nyelvek alkalmazása a programozásukhoz. Fő területük jelenleg a mikroelektronikai gyártás, ahol sokszor már emberi beavatkozásra sincs nagy szükség. Ilyen például az integrált áramkörgyártás összetett és nagy precízitást igénylő feladata.

A harmadik generációs robotok létrejötte a mesterséges intelligencia fejlődésének és gyakorlati alkalmazásának köszönhető. Napjaink robotkutatásai is ezen a területen folynak. A mesterséges intelligencia felhasználásával egyre több harmadik generációs robotot terveznek és készítenek.

A harmadik generációs robotok az érzékelt jelek feldolgozása után, egy mesterséges intelligenciára épülő rendszert (például szakértői rendszert) igénybe véve elemzik a gyűjtött az információkat. Tevékenységeikről magas szintű megfogalmazásokat adnak, önállóan hoznak döntéseket vagy módosítják viselkedésüket valós környezetben. Jelenleg még inkább

kutatási célokra alkalmazzák, nem tömeggyártásban az orvostudománytól kezdve az űrkutatásig, bár ipari kezdeményezések is találhatóak, például a mesterséges intelligenciával ellátott mobil szállító robotok (robot targoncák). Illetve a szolgáltató robotok közül leginkább a haditechnikában alkalmaznak harmadik generációs robotokat, mint például a robotrepülők vagy a felderítő és aknakereső robotok.



7. ábra, Az első, második és harmadik generációs robotok strukturális modelljei (átalakított forrás [10.])

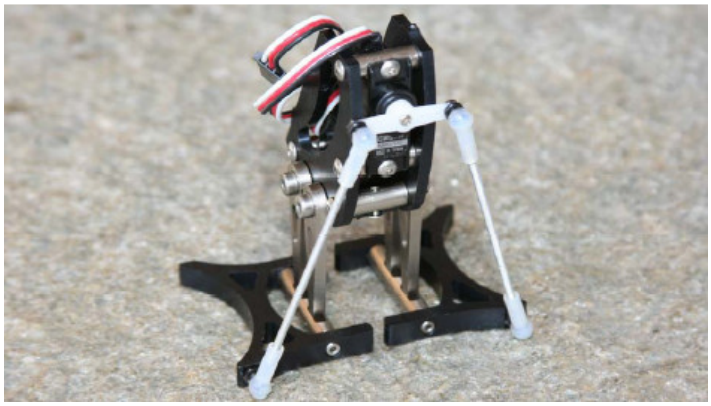
A robotok rendszerszervezése

A robotok legegyszerűbben és legpontosabban a rendszertervezés strukturált szemlélete alapján tervezhetőek és készíthetőek. A kész robotok pedig rendszerszintekre bontva adhatóak meg a legjobban. Ezt az elvet követve a Penguin robotnak is megvizsgáltam a rendszerszervezését. Alapjaiban véve három rendszert különböztetünk meg egymástól. [8.]

A mechanikai rendszer

A mechanikai rendszert felépítő főbb komponensek a szerkezeti váz és a működtető berendezések. A vázszerkezet mozgatható részei a működtető és mozgató berendezések segítségével végeznek mozgási, vagy mozgatási műveleteket. Ilyen berendezések többek között a szervo és léptető motorok, de akár a hidraulikus vagy pneumatikus berendezések is.

- Helyváltoztató berendezések: Olyan berendezések, amelyek a vázszerkezet aktív részeit mozgatják, aminek következményeképpen mozgást végez a környezetében.
- Manipulációs berendezések: Olyan mozgató berendezések tartoznak ide, amelyek objektumokkal vagy objektumokon hajtanak végre műveleteket, de a robot helyzetén nem változtatnak.



8. ábra, A Penguin robot mechanikai rendszere: a szervo motorok és vázszerkezet összeszerelve (forrás [21.])

A szervo motorok olyan impulzus vezérelt egyenáramú motorok, melyekkel legfeljebb 180 fokos kitérés érhető el. Az elektronika az impulzus egységeket szögekre számítja át, és ennek függvényében fordul el a motor. Méretükhöz képest nagy erő kifejtésére képesek. A szervo motorok alkalmasak bármilyen robot végtag mozgatására, mint például a Penguin robot lábai.

9. ábra, A Penguin robot mikro szervo motorjai (forrás [21.])

A szenzoros rendszer

A robotot körülvevő környezet egy hatalmas jelforrás, és ehhez a megfelelő szenzorokat kell összeválogatni. A szenzorok olyan érzékelő eszközök, amelyek segítségével a robot mérni vagy észlelni tudja a különböző környezeti jellemzőket. Feladatuk, hogy a környezetből nyert információt begyűjtsék, majd továbbítsák feldolgozásra a vezérlő rendszernek.

- **Belső állapot**: A szenzoros rendszer egyik feladata a robot belső állapotának az érzékelése, például a robot belső hőmérséklete, a mozgási sebessége vagy a szögelfordulás mérése a szervo motorokon.
- **Külső állapot**: Az érzékelő rendszer másik feladata a külső állapot, a tárgyaknak és a környezetnek az érzékelése, például a robot előtti akadályok, a fény, a távolság mérése vagy a tárgyak színének felismerése.

A működés szempontjából a szenzoroknak két típusát különböztethetjük meg.

- **Passzív szenzorok**: A passzív szenzorok természetes eredetű elektromágneses sugárzásokat, illetve fizikai érintkezést érzékelik, mérőjelet nem bocsájtanak ki. Ilyen passzív szenzorok például az infravörös, a látható fény, a mágnesesség, a hőmérséklet vagy a nyomás érzékelők, illetve a kapcsolók.
- **Aktív szenzorok**: Az aktív szenzorok saját maguk bocsájtanak ki egy bizonyos fizikai tulajdonságú mérőjelet, amelynek a visszaverődését várják. Ilyenek például az ultrahangos vagy lézeres távolságmérő eszközök és a radarok.

A szenzoros rendszer által nyerhető információk pontosságának és hatékonyságának növelése érdekében gyakran használnak szenzorfúziót. A különböző szenzorokból nyert információk feldolgozása és összevetése után pontosabb kép kapható a környezetről, mint az egyes érzékelők által nyert információk külön-külön történő kezelése révén.

A vezérlőrendszer

A vezérlőrendszer a mechanikai és szenzoros rendszer között teremti meg a kapcsolatot, lényegében ez a robot agya. A vezérlőrendszer feladata a robot szenzoros rendszere által nyert információk feldolgozása és ezek alapján a mechanikai rendszer műveleteinek irányítása.

A vezérlőrendszer rendszerint valamilyen digitális rendszerre épül, amely lehet egy általános célú személyi számítógép vagy például egy mikrokontrolleres beágyazott rendszer. A vezérlőrendszer - a hardver mellett - magába foglalja a vezérlési utasításokat tartalmazó programokat és a hozzájuk kapcsolódó összetett algoritmusokat.

A vezérlőrendszer hozza meg a döntést és adja ki az utasítást az adott mozgásról, mozgatásról vagy mozdulatról (egyszóval, az akcióról), a mechanikai rendszer belső és a környezet külső érzékelt jelei alapján. A modern robot-vezérlőrendszerek mögött egyre gyakrabban áll a mesterséges intelligencia és az arra épülő intelligens rendszerek, mint például a szakértői rendszerek, a szabályozáselmélet vagy a kibernetika.

Különböző mobil robot mozgatási rendszerek összehasonlítása

Szeretnék kitérni egy szerintem fontos témakörre, mégpedig a leggyakrabban előforduló mobil robot mozgatási rendszerek összehasonlítására.

A **kerekes mozgatás** során a robot leggyakrabban kettő vagy négy keréken gurulva mozog, de a tervezés összetettségétől függően egy, illetve kettőnél több kerékkel is fel lehet szerelni. A négy kerekes meghajtáshoz általában két (jobb esetben szervo) motort alkalmaznak, mozgásukban eltérés abból következhet, hogy a motorok tengelyére az első és a hátsó vagy a két oldalsó kereket kapcsolják. Az ilyen mobil robotok képesek méretüktől függően kisebb akadályokon akár nagy sebességgel is keresztülhaladni.

A **lánctalpas megoldás** hasonló a kerekes robotokéhoz, csak a motorok tengelyére kerék helyett fogaskerekekkel összeszerelt lánctalpakat szerelnek. A lánctalpas robotok nagy előnye, hogy egy helyben meg tudnak fordulni, illetve méretüktől függően nehéz terepviszonyok között (például homokban), meredek emelkedőn vagy akár nagyobb akadályokon is képesek keresztülhaladni. Speciális kialakítású lánctalpas robot akár még lépcsőzni is képes. Mozgási sebességük azonban kisebb, mint a kerekes robotoké.

A **lépegető robotoknak** több típusa is létezik. Rendszerint kettő, illetve ennél több (például rovar felépítésű robot) lábbal rendelkeznek. A két lábbal rendelkező robotok között a leggyakoribb megoldás a súlypontáthelyezéses lépegetés, amelyek különböző bonyolultságúak és szabadságfokúak lehetnek. Például, a Penguin robot mozgása is ilyen elven működik, egyszerű felépítéssel, de elég korlátozott körülmények között. A két lábon lépegető robotok többsége nem képes nagy sebességű haladásra és nagyrészt csak sík, akadálymentes felületeken tudnak mozogni. A kettőnél **több lábon lépegető robotok** sebessége és stabilitása nagyobb, mint kétlábú társaiké és közepes emelkedőn is képesek felmászni, illetve méretüktől függően közepes akadályokat átlépni.

Az 1. táblázatban négy Parallax mobil robot mozgatási rendszereit hasonlítottam össze.

1. táblázat, Mobil robot mozgatási rendszerek összehasonlítása (saját tapasztalatok alapján készített táblázat)

Lehetséges megoldások	Motor típusa	Mozgási lehetőségek	Sebesség	Stabilitás	Energiaigény	Programozási komplexitás
4 kerekű robot (Quad Rover) 	2 darab folyamatos forgásra képes szervo motor	Kisebb emelkedő, kisebb akadályok, könnyű fordulás	Nagy sebesség	Közepesen nagy stabilitás	Kis, a kerekek kis ponton érintkeznek a talajjal	Egyszerű
Lánctalpas robot (Boe-bot Tank Tread Kit) 	2 darab folyamatos forgásra képes szervo motor	Meredek emelkedő, nagyobb akadályok, nagyon egyszerű fordulás	Közepes sebesség, a lánctalp fogaskerék áttételei lassítják	Nagy stabilitás	Közepes, a lánctalp nagyobb felületen érintkezik a talajjal	Egyszerű
2 lábon billegve lépegető robot (Penguin) 	2 darab 180 fokos szögben elforduló szervo motor	Sík, akadálymentes felületek, nehézkes mozgás	Alacsony sebesség	Kis stabilitás, érzékeny a robot súlypontjára	Nagy, a talpakra és a motorokra eső terhelés nagy	Közepesen összetett
6 lábon lépegetés (Boe-bot Crawler Kit) 	2 darab folyamatos forgásra képes szervo motor	Közepes emelkedő, közepes akadályok	Közepes sebesség	Közepesen nagy stabilitás, lehetséges oldalirányú dőlés	Közepes, a lábakra és motorokra eső terhelés jobban eloszlik	Közepesen összetett

A táblázatból következtethető, hogy a leghatékonyabb mozgatási rendszer a lánctalpas megoldás, míg a Penguin robot lépegetése a legkevésbé hasznosítható a mobil robotoknál.

A Parallax és a Penguin robot

A Parallax cég és termékei

A kaliforniai Rocklin városában székelő, amerikai Parallax cég nem robotok készítésével kezdte. Az 1990-s évek elején dobták piacra első mikrokontrollerüket, ami jelenleg is a cég fő profilját alkotja, de pár éve termékpalettájukon már robotok is találhatóak, amelyeknek az irányítását Parallax mikrokontrollerek végzik.

A Parallax világszerte forgalmazza mikroszámítógépeit, így hazánkban is jelen vannak. Annak köszönhetik sikereiket, hogy termékeik értékteremtőek, innovatívak és viszonylag kedvező áron beszerezhetőek. Mikrokontrollereiken nagyon sok fiatal mérnök tanult és tanul jelenleg is. Széles termékpalettájuk rengeteg feladatra kínál megoldást az oktatástól kezdve, az ipari megoldásokig.

A Parallax robotok

A Parallax több különféle, főleg oktatási és hobbi célú mobil robotot készít. Legtöbbjük ezen kívül még teljesen át is alakítható, amihez a cég különféle készleteket is árul.

Amiben alapvetően mindegyik Parallax robot megegyezik:

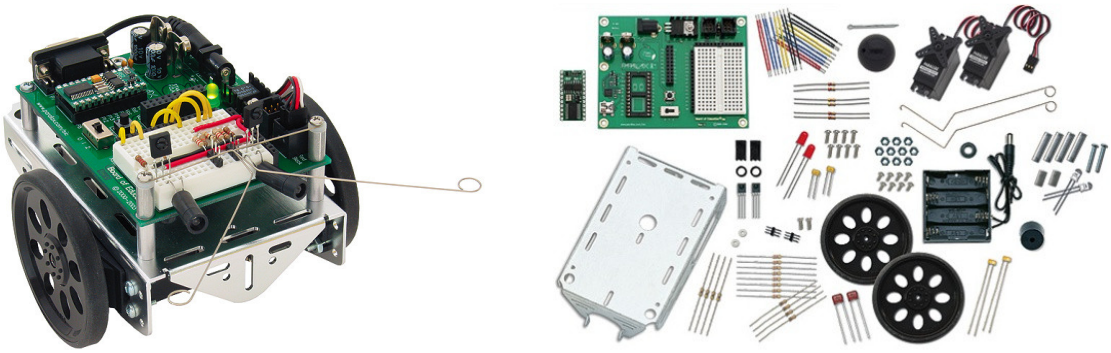
- gyári mikrokontrolleres vezérlés
- mobil robotok, saját tápellátással
- nagyfokú bővíthetőség és átalakíthatóság, moduláris felépítés
- univerzális szervo motorok, kommunikációs és egyéb eszközök
- univerzális szenzorok, szenzorcsomagok a különféle robotokhoz
- elektronikus terméktámogatás, fejlesztési tanácsadás és fórum

Jelenleg hat robotkészlet kapható, kezdve az egyszerűbbektől, - amikkel akár már kisebb gyermekek is próbálkozhatnak - az összetettebbekig, amelyekkel már komolyabb projektek is megvalósíthatóak (például mérnökök számára).

Scribbler Robot: Apró, háromkerekű, előre összeszerelt robot. Egyszerű felépítésű grafikus felületen, lépcsős elven programozhatóak, már 8 éves kortól ajánlják használatát, aminek köszönhetően a gyerekek hamar megismerkedhetnek a robotikával. Basic Stamp mikrokontroller vezérli.

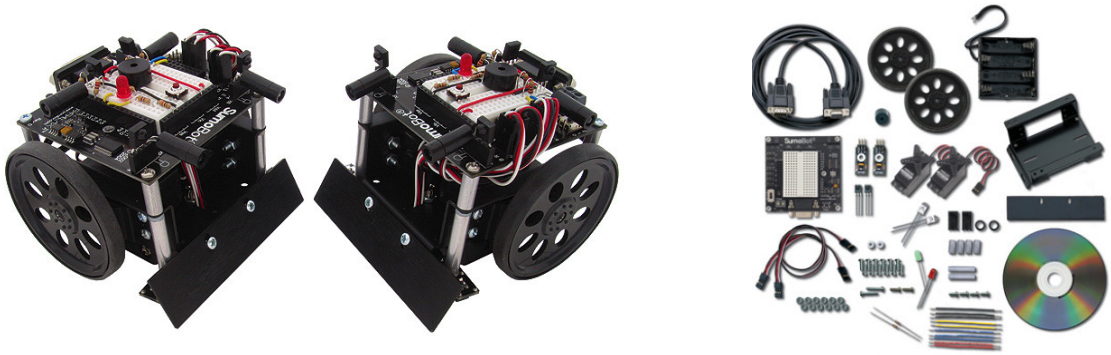
10. ábra, A Scribbler robot (forrás [19.])

Boe-Bot Robot: A Parallax legnépszerűbb, készletben kapható, háromkerekű, összeszerelhető robotja. Könnyen programozható, sokféleképpen átalakítható, rengeteg szenzorral és alkatrésszel bővíthető. Ezt a robotot is Basic Stamp mikrokontroller vezérli.



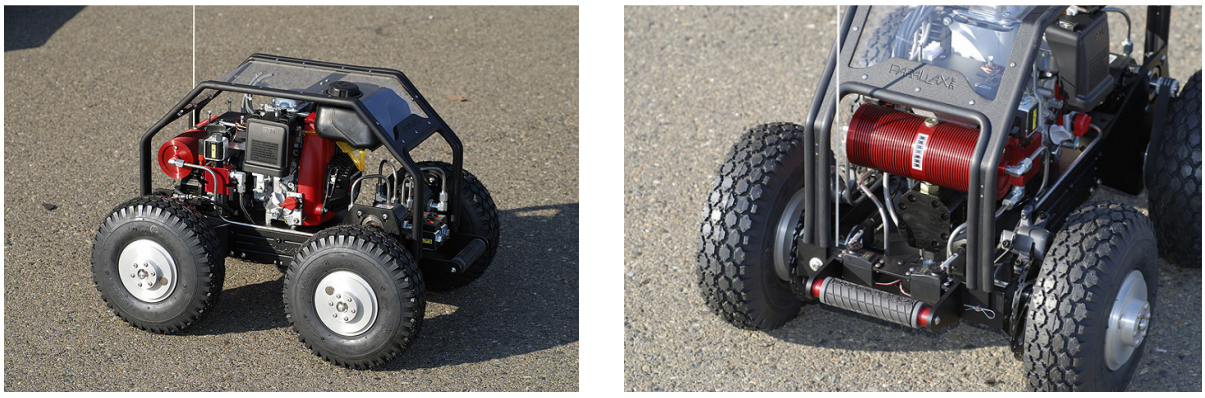
11. ábra, A Boe-Bot robot, egy ajánlott kialakításban (forrás [19.])

SumoBot Robot: Két darab szumó birkózó, Boe-Bot alapkészletre épülő, összeszerelhető robot alkotja. Ezzel a robottal akár nevezni lehet hivatalos robotbirkózó versenyekre is. Vezérlését szintén egy Basic Stamp mikrokontroller látja el.



12. ábra, A látványos és robosztus SumoBot robot, küzdő versenyekhez tervezve (forrás [19.])

Propeller QuadRover: Négykerekű gázmotoros terepjáró robot. Akár egy gépkocsit is képes elhúzni. Vezérlését multiprocesszoros Parallax Propeller mikrovezérlő látja el.



13. ábra, A nagyteljesítményű QuadRover robot (forrás [19.])

Stingray Robot: Háromkerekű, rengeteg szenzorral bővíthető, multiprocesszoros Propeller mikrovezérlővel ellátott mobil robot, haladó programozóknak és mérnököknek.

14. ábra, A Stingray robot (forrás [19.])

A Parallax lépegető robotja, a Penguin robot

A Penguin, a Parallax lépegető mobil robotja remek lehetőséget kínál a robotikával való ismerkedéshez. A robot készlet ára viszonylag magas, és a többi Parallax vagy más gyártók által kínált robotkészletekhez képest *szerintem* kissé rugalmatlan felépítésű és sajnos csak korlátozottan bővíthető. Mindezek mellett azonban a robot jól átalakítható.

15. ábra, A Parallax Penguin robot eredeti és átalakított formában (átalakított forrás [19.])

A Parallax-nak nem ez az első lépegető robotja. A Penguin robot elődje a Toddler volt. A Toddler és a Penguin lépegető robotok prototípusát Ken Gracey (a Parallax cég alelnöke) fejlesztette ki, több más saját készítésű robotja mellett. Tervei alapján a Toddler-t 2002-ben, a Penguin kódnevű robotot pedig 2007-ben kezdte tömegesen gyártani és forgalmazni a Parallax, több színvariációban. [32.] A Toddler nagyobb (25 cm magas) és nehezebb robot volt, mint a Penguin. A Toddler robot bővíthetőségét szolgáló kiegészítő modulok és a szenzorok a robot tetején található próbapanelre szerelhetőek. Vezérlését a Basic Stamp BS2 mikrokontroller látta el. A Toddler szó angolul a járni tanuló, totyogó kisgyermeket jelenti.

16. ábra, A Toddler és a Penguin méretarányos összehasonlítása (átalakított képek, forrás [20.])

A Penguin robot specifikációja (Rev.A)

BASIC STAMP 2 px (BS2px) beágyazott mikrokontroller,

CNC megmunkált 6061 alumínium alkatrészek fekete, piros, kék vagy nikkel színben,

egy Hitachi HM55B digitális iránytű,

2 db foto-ellenállás fényérzékeléshez,

2 db infravörös led és egy infravörös érzékelő,

kék színű, hét-szegmenses LED kijelző és a működést visszajelző kék LED,

2 db mikro szervo motor a billenéshez és lépéshez,

egy Piezo hangszóró (a hátsó panelen az elemtartó mögött),

egy bővítő port PING ultrahang szenzorhoz vagy egyéb opcionális komponensekhez,

az elemtartó mellett elhelyezett bekapcsoló gomb,

FTDI 232RL mini USB programozó port,

CR123-s méretű elemtartó (2 db CR123 lítium elemnek, a 6 Voltos tápellátáshoz).

A Penguin robot mechanikai rendszere

A Penguin robot gyári kialakítása

Figyelemre méltó, hogy a Penguin csak 10 cm magas. A robot két lába a szervo motorokat magába foglaló, a panelt és elemtartót is hordozó fémtörzshöz kapcsolódik. A lábmunkához a két mikro szervo motor működését kell összehangolni. Az alul felszerelt szervo motor előre és hátra mozgatja a lábakat, míg a felül elhelyezett motor billegteti a robotot.

A robot lábai csak a törzs mentén tudnak elmozdulni előre és hátra keskeny szögben. A lábak két rövid tengellyel kapcsolódnak a talpakhoz. A talpak a két tengely segítségével tudnak a billenés irányába dőlni. A billenés során hol az egyik, hol a másik talpra nehezedik a robot súlya, miközben a robot előre hátra lépteti a lábait. Az összekötő tengely segítségével a két mozgáselemből alakul ki a lépegetés.

17. ábra, A Penguin robot lábai tengellyel kapcsolódnak a talpakhoz (forrás [21.])

A természetfilmekben gyakran láthatóak a Föld déli féltekéjén élő különleges madarak, a pingvinek. Rövid, térd nélküli merev lábaikkal egy helyben állnak a havas és jeges talajon, járásuk pedig a billegés és a lépés kombinációjából jön ki.

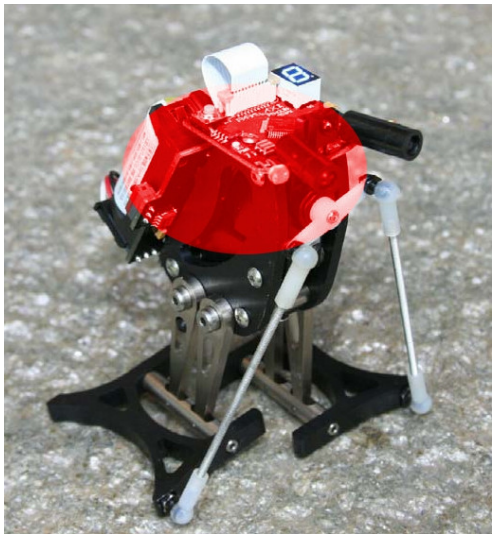
A lépegetések megvalósítására példaként bemutatok **egy előre lépést**, ami a *jobbra dől, bal lábbal előre lép, balra dől végül jobb lábbal előre lép*, illetve egy **tengely körüli jobbra fordulást**, ami a *jobbra dől, bal lábbal előre lép, középre dől, jobb lábbal előre lép, balra*

dől, bal lábbal előre lép, középre dől, végül jobb lábbal előre lép mozgáselemek kombinációiból jön ki.

A Penguin robot fém alkatrészei alumíniumból készültek és - a CNC szerszámgépes megmunkálásnak köszönhetően - pontosan illeszkednek egymáshoz.

A robot tetején található áramköri panel tartalmazza beágyazott mikrokontrollert, a szenzorokat és egyéb elektronikai alkatrészeket. A robot hátára szerelt panelen található az elemtartó, az USB csatlakozó (mini USB) és az USB vezérlő chip, egy piezo hangszóró és az elemtartó mellett a robot kétállású főkapcsolója.

Érdemes megvizsgálni a robot súlypontját. A nehezebb elemek felül találhatóak. Jóval a lábak fölött helyezkednek el a szervo motorok, az elektronika, az elemek és az egyéb alkatrészek. Emellett a robot nem teljesen a talpain áll, hiszen egy apró állítás a billenő szervo motoron középállásból máris szinte a talpak élére állítja a robotot.



18. ábra, Az alap kiépítésű Penguin robot súlypontját szemléltető kép (átalakított kép, forrás [21.])

Ennek következménye lehet, hogy az esetleges gyors vagy szélsőséges motor mozdulatok miatt, döntött felületen vagy apró akadályon a robot könnyen megakadhat vagy felborulhat.

A Penguin robot mérete, felépítése és tápellátása miatt nem képes nagy távolságok megtételére, mozgási sebessége és teherbírása alacsony. Járásának megtervezése és

megvalósítása sokkal kifinomultabb programozási módszereket igényel, mint a kerekes robotok mozgása, mivel a szervo motorok működését különleges módon kell összehangolni.

A gyalogló, két lábon járó robotok tervezése, építése és programozása viszonylag járatlan terület a robotikában, hiszen rengeteg korlátja van a lépegetés és járás megvalósításának.

Említettem korábban, hogy szerintem ez a Parallax robot kissé rugalmatlan felépítésű. A Penguin robot esetében ez annyit jelent, hogy nem tud se lépcsőzni, se emelkedőn vagy lejtőn lépegetni még kis mértékben sem, mert egyszerűen felborul. A szerkezeti kialakítás korlátai miatt csak vízszintes felületeken, (mint például padlón vagy asztalon) halad stabilan és önmagára nézve is biztonságosan. Mindezeket leszámítva, vízszintes, akadálymentes felületeken a robot megfelelően teljesíti a tőle elvárható feladatot, a lépegetést.

Az átalakított Penguin robot felépítése (*A söprögető pingvin*)

A roboton öt (kisebb-nagyobb) átalakítást végeztem:

- Az egyik lábszáron a csavar fejének helye nem volt jól bemarva, ezért a csavar állandóan kilazult, ami egy pontatlan gyári alkatrész megmunkálás miatt történt.
- Kiforrasztottam az infravörös ledeket az áramköri panelről és helyükre tördelhető IC foglalatot tettem, amibe hosszabbító kábellel bekötöttem a ledeket (az útvonalkövetéshez).
- Hosszas kísérletezést követően, műanyagból készítettem egy viszonylag merev, előre döntött, stabil tartószerkezetet az infravörös ledeknek (az útvonalkövetéshez).
- Az elemtartót úgy rögzítettem fel, hogy a digitális iránytű chiptől távolabb kerüljön, mert az elemek megzavarták az iránytű működését.
- Kiegészítő, házilag készített elemtartót szereltem fel a robotra az eredeti elemtartó fölé és ezt párhuzamosan kötöttem be, ügyelve a helyes polaritásra, ezzel növelve az elemek terhelhetőségét és a működési élettartamot.

Az átalakításoktól a Penguin robot tömege nőtt, súlypontja pedig a kiegészítő elemtartó miatt enyhén hátra tolódott. Az átalakításoknak köszönhetően a robottal olyan feladatok is végrehajthatóak, amelyek az alapkiépítésű robottal nem, például az útvonalkövetés.

A Penguin robot vezérlőrendszere

A mikrovezérlők és a BS2px mikrokontroller

A Penguin robot lelke egy beágyazott Parallax Basic Stamp 2px mikrokontroller. A Parallax robotoknál tény, hogy ahány robot, annyi mikrovezérlő. Mindegyik mikrokontroller különböző tulajdonságokkal és beállításokkal rendelkezik, nem úgy, mint az egységes vezérlővel ellátott robotoknál. Más típusok vezérlőrendszerei azonban leggyakrabban rejtett specifikációjúak.

A Parallax nem titkolt célja, hogy a programozó ne csak robotot, hanem mikrokontrollert is tanuljon programozni. Ez egy olyan széleskörű tudásszerzésre és szemlélet kialakítására ad lehetőséget, ami fontos lehet a későbbiekben és informatika más területien is jól hasznosítható.

A mikrovezérlő definíciója

A mikrovezérlők (vagy más néven mikroszámítógépek) a számítástechnika fejlődésének köszönhetőek. Az első mikrokontrollert 1974-ben mutatta be a Texas Instruments TMS 1000 néven, 28 évvel az első elektronikus számítógép (Eniac, 1946) és 3 évvel az első mikroprocesszor (Intel 4004, 1971) megalkotása után.

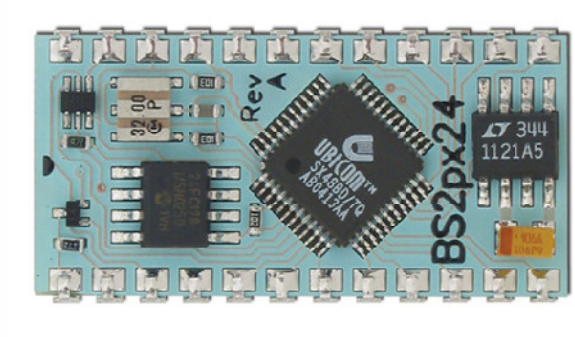
Vizsgáljuk meg először a definícióját. A mikrovezérlő vagy más néven mikrokontroller definíciója sokkal egységesebben megfogalmazott, mint a roboté.

A Merriam Webster Online szótár szerint a mikrovezérlő definíciója a következő:

- **„egy mikroprocesszor, ami irányítja néhány vagy minden funkcióját egy elektronikus eszköznek vagy rendszernek” [11.]**

A mikrovezérlő egy félvezető lapkán kiépített kisméretű integrált áramkör. A mikroprocesszor különböző utasítások végrehajtásával kezel bemeneteket és működtet

kimenetére csatlakozó eszközöket. A mikrovezérlő egy mini számítógép, ugyanazon tokban nem csak a mikroprocesszor kap helyet, hanem mellette különböző perifériák is és ezek együttesen alkotják a mikrokontrollert.



19. ábra, A Penguin roboton beágyazva megtalálható BS2px mikrovezérlő, az eredeti DIP tokozásában (forrás [22.])

Mikrovezérlők a robotok vezérlőrendszereként

Napjainkban a mikrovezérlők használata nagyon elterjedt, felhasználási területük széleskörű. Köszönhető ez kis méretüknek, moduláris felépítésüknek, alacsony fogyasztásuknak, kedvező áruknak és annak, hogy szinte bármit irányíthatunk megfelelően programozott mikrokontrollerrel.

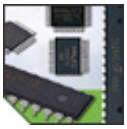
Ha lehet választani az általános célú személyi számítógépek, a PLC-k (Programozható logikai vezérlő), az FPGA-k (programozható kapuhálózat) vagy a mikrovezérlők hasonló célú felhasználása között - mint például egy robot vezérlése -, akkor sok esetben (kellemes tulajdonságai miatt) a mikrokontrollerekre esik a választás.

Nem véletlenül alkalmazzák az autóipartól kezdve, az orvosi eszközökön át az űrkutatásig. Az élet szinte minden területén találkozhatunk mikrovezérlővel szerelt készülékekkel, így nem véletlenül a robotika is kiemelten foglalkozik felhasználásukkal és a robotok vezérlésére alkalmazva jelentős szerepet töltenek be.

Parallax mikrokontroller családok

Jelenleg négy mikrokontroller családot forgalmaz a Parallax és ezek közül szerelik fel a különféle robotokat.

Javelin Stamp: Egyetlen mikrovezérlőről van csak szó, aminek a felépítése megegyezik a Basic Stamp 2px-el, a különbség annyi, hogy a Javelin Stamp PBASIC helyett Java nyelven programozható.



Sx: Nagy sebességű egyetlen processzormagos, 8 bites RISC (Csökkentett Utasítás Készletű Számítógép) mikrokontrollerek. Assemblyben (a gépi kódhoz legközelebb álló nyelven) és az Sx/B, BASIC alapú nyelven programozhatóak.

Propeller: A Parallax nem modul szintű chipjei. Nyolc darab független, 32 bites, RISC processzormagja van (a COG-ok), mind a négy forgalomban lévő Propeller mikrokontrollernek, amelyek akár 20 MIPS (millió utasítás per másodperc) sebességre is képesek. Videó jel feldolgozásra, vagy valós idejű rendszerek irányítására is alkalmas. A Spin nevű magas szintű objektum orientált nyelven programozhatóak.

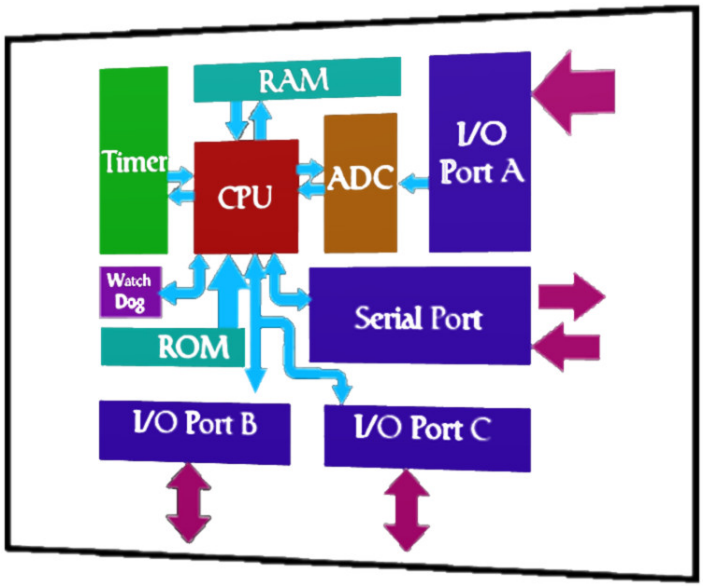
Basic Stamp: A Basic Stamp volt az első, már 1991 óta forgalmazott Parallax modul szintű mikrokontroller család. A Basic Stamp elnevezés azt jelenti, hogy egy BASIC nyelven programozható bélyeg (Stamp) méretű mikrovezérlő. A BS1 és a BS2-s még PIC mikrokontroller volt, azonban a többi Basic Stamp 2-s már Sx mikrovezérlő. Mindegyik Basic Stamp PBASIC nyelven programozható. A család legújabb és egyben leggyorsabb tagja a Penguin roboton is megtalálható BS2px mikrovezérlő.

A mikrovezérlők felépítése

A mikrokontrollerekben egyetlen integrált áramköri tokban van kialakítva hely a mikroprocesszornak (CPU) és a különböző perifériáknak. A mikroprocesszor mellett a félvezető lapkán leggyakrabban memóriaegységek, be- és kimeneti portok (I/O), órajel generátorok, soros kommunikációs interfészek, analóg-digitális átalakítók és találhatóak. A legelterjedtebbek a Harvard architektúrára épülő PIC-k (Programozható Integrált Áramkör), de léteznek a továbbfejlesztett, gyorsabb változatai is, az Sx mikroprocesszorok.

A Harvard architektúra lényege, hogy a számítógép a programutasításokat és az adatokat külön memóriákban tárolja. Ez a felépítés biztonságosabb és hatékonyabb is, mint a Neumann-elvet követő mikroprocesszoroké, miközben a Harvard architektúra nem sérti a Neumann-elvet, ezért gyakran alkalmazzák mikrovezérlők tervezésekor.

A mikroprocesszor feladata az utasítások végrehajtása. A mikroprocesszor egyetlen szilícium kristályon található több tíz vagy százmillió tranzisztorból felépülő digitális eszköz. Egy utasítást bizonyos idő alatt végez el, több utasítást az órajel generátor által előállított órajellel szinkronizálva tud végrehajtani. A processzorokat az adatbuszuk szélessége szerint osztályozzák. A mikrokontrollerek a 4 bitestől kezdve 32 vagy akár 64 bites adatokkal is tud műveleteket végezni, de felmérések alapján, a világon használatban lévő mikrovezérlők 55%-a jelenleg 8 bites. [16.] Az utasítások hossza (bitekben) eltérhet az adatok szóhosszától.



20. ábra, A mikrokontrollerek általános felépítése (forrás [23.])

A BS2px egy Ubicom SX48AC mikroprocesszorral szerelt mikrokontroller. Az órajel generátor 32 MHz-s órajellel szinkronizálja a processzort, ami így akár másodpercenkénti 19.000 PBASIC utasítás végrehajtására is képes. (Ez az eddigi leggyorsabb Basic Stamp)

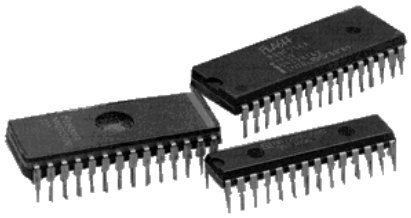
A memóriáknak két típusa található meg a mikrokontrollerekben. Az egyik a programutasítások tárolására szolgáló, a másik a futási idő alatt adatokat tároló memória.

- Kódmemória
- Adatmemória

A kódmemória ROM (Csak Olvasható Memória), EPROM (Törölhető, Csak Olvasható Memória), EEPROM (Elektromosan Törölhető, Csak Olvasható Memória) vagy Flash (hasonló az EEPROM-hoz, csak gyorsabb) memória. A ROM-t csak egyszer lehet felprogramozni és ez csak gyárilag történik. A másik három típusra többször is elmenthetjük (és törölhetjük) a programokat. A kódmemória a tartalmát a tápfeszültség kikapcsolása után is megőrzi (non-volatile: nem illékony). Gyakran adják meg ezeknek az élettartamát újraírhatósági élettartamban, ami általában pár százezer és több millió között mozgó szám és azt mutatja meg hányszor írható át a memória tartalma.

A kódmemóriában tárolódik a mikrovezérlő számára végrehajtandó utasításokat tartalmazó programkód. A kódmemóriákban tárolható programok mérete azonban korlátozott, általában 8 és 64 Kilobyte közötti memóriamodul található az integrált áramkörön. Ha mégis szükségünk lenne nagyobb tárterületre programkódjainknak, akkor számos bővítési megoldást találhatunk, persze ez nagyon megdrágítja a kiépítés árát és használhatósága függ a mikrokontroller típusától is. Ezért sokszor nélkülözhetetlen programkódjaink írásakor kódjaink optimalizálása, hatékonyságanalízise.

A BS2px mikrovezérlőben 16 Kbyte EEPROM kódmemória található, ami nyolc különálló 2 Kbyte-s részre, memóriarekeszre (slot-okra) van bontva.

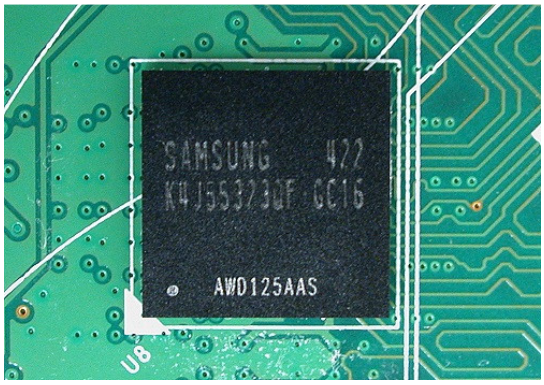


21. ábra, Különböző típusú ROM memóriachipek (forrás [24.])

Az adatmemória rendszerint RAM (Véletlen Elérésű Memória) memória. A véletlen elérés azt jelenti, hogy bármely, véletlenszerűen kiválasztott memóriacím eléréséhez ugyanannyi időre van szükség. Írható és olvasható memória, azonban a tápfeszültség kikapcsolása után elveszti tartalmát (volatile: illékony).

Az adatmemória ideiglenesen tárol adatokat a mikroprocesszor számára. Ezek az adatok lehetnek a mikroprocesszornak szánt utasítások, adatok és számítási eredmények. Az utasítások és adatok a kódmemóriában tárolt program gépi nyelvre történő fordítása révén kerülnek be az adatmemóriába.

A BS2px mikrokontrolleren 38 Byte (nagyon kevés) RAM memória használható a működés során, emellett 128 Byte úgynevezett ScratchPad RAM memória is található futtatás alatti adatmentéshez és a külön EEPROM memóriarekeszekre mentett programok egymás közötti adat és információ megosztásához.



22. ábra, Példa egy RAM memóriachipre (forrás [25.])

A be és kimeneti portok, vagyis az I/O-k segítségével tudjuk a mikrokontroller vezérlő lábaihoz bekötött külső elektronikai eszközöket irányítani, azokkal kommunikálni. A mikrovezérlő soros I/O-n vagy soros kommunikációs interfészen keresztül - különböző soros adatátviteli protokollok segítségével - kommunikál a perifériákkal. Soros interfész például az I^2C, Maxim/Dallas Semiconductor 1-Wire vagy az RS232.

A BS2px mikrovezérlőben 16 + 2 darab I/O található és az alap soros kommunikáción kívül a felsorolt protokollok is használhatóak.

A Penguin robot kapcsolói és visszajelző eszközei

A roboton az elemtartó mellett található egy kisméretű kétállású kapcsoló, ami a főkapcsoló funkcióját tölti be, ami az elektronika tápellátását kapcsolja be vagy szakítja meg.

A panel tetején található a felső sarokban egy apró RESET gomb. A RESET gomb megnyomásával megszakítható a bekapcsolt állapotban lévő roboton futó program működése és a gomb folyamatos nyomva tartásáig ez az állapot fennáll. A gomb felengedése esetén a program újraindul.

Az elemtartó mögött található egy szintén kisméretű elektronikai alkatrész, egy piezo hangszóró, vagy más néven piezo zümmögő. A piezo hangszóró a piezoelektromosság elvén működik. A benne elhelyezett kristályok alakja feszültség hatására deformálódik, amely az impulzussorozat frekvenciájával megegyező hanghatást vált ki. A piezokristályok elsősorban magasabb frekvenciás hangokat tudnak kiadni. Működtetéséhez a FREQOUT parancs használható.

A panel tetején a RESET gombbal ellentétes sarokban található egy hét szegmenses kijelző. A hét szegmenses kijelző decimális számokat tud hét vonallal kirajzolni 0 - 9-ig. Emellett a hexadecimális A - F értékeket is kirajzolja, különféle torzított alakzatokban. (Illetve az F esetén jelen esetben nem világít kijelző.) Az eredeti tervek alapján a kijelző feladata az egymással rádiófrekvenciás kapcsolatban lévő Penguin robotok megkülönböztetése lett volna. [32.] Igazából azonban használata nem indokolt és a hozzá lefoglalt négy I/O láb (a Rev.A kiadású Penguin roboton) - bővíthetőség szempontjából is - hátrányos.

23. ábra, Egy hét szegmenses kijelző kikapcsolt állapotban. (forrás [x.])

A Penguin robot szenzoros rendszere

A megfelelő tájékozódáshoz a robotnak a környezetéről és állapotáról információkkal kell rendelkeznie, ennek érdekében a Penguin roboton különböző beépített érzékelők, más néven szenzorok találhatóak. A szenzorok feladata, hogy fizikai mennyiségeket mérjenek és érzékeljenek, amelyeket a mikrokontroller saját maga számára érthető és felhasználható digitális információvá alakít. A Penguin roboton három különböző érzékelő már be van építve. Az infravörös érzékelő mellett két darab fényérzékelő és egy digitális iránytű található.

A Penguin robot beépített érzékelői

Infravörös szenzor

(Infrared receiver - Panasonic PNA4601M)

Az érzékelő akár 8 méteres hatótávolságból is érzékeli az infravörös fényt, az észlelést 1 biten tárolva (igen/nem), tehát az infravörös fény intenzitását nem tudja mérni. Az infravörös fény szabad szemmel nem látható elektromágneses sugárzás, amit minden test kibocsájt hő formájában (közép és hosszú hullámhosszú infravörös). Azt elkerülendő, hogy az érzékelő minden környezeti infravörös zajt (például izzólámpa) érézkeljen, csak a 38 KHz-s vivőfrekvenciára ültetett közeli infravörös tartományú, 940 nanométeres (közeli infravörös : 0,75-1,4 µm) hullámhosszúságú infravörös fényt érzékeli.

Fényérzékelők

(Photoresistors - PerkinElmer VT935G-B)

A fényérzékelők fotóellenállásokból állnak. A szenzor a fény jelenlétének és észlelésére és a lassú változások érzékelésére alkalmazható. Minél nagyobb intenzitású a fényérzékelőre eső fény, annál kisebb lesz az ellenállása és fordítva.

Digitális iránytű

(Digital Compass Sensor - Hitachi HM55B)

A szenzor segítségével meghatározható az északi irány. A digitális iránytű az érzékelő két tengelyére párhuzamosan eső mágneses erővonalak erősségét méri. A robot az irány érzékelésének képességével felismeri abszolút helyzetét, ezzel beállítható viszonylag nagy pontossággal bármilyen irányba. Használatakor figyelembe kell venni, hogy nagyon érzékeny a mágneses anyagokból készült eszközök, például a mágnesek, a motorok, a közelében lévő vezetékek vagy elemek által keltett zavaró mágneses kölcsönhatásokra is. Ha szükség van a pontos földrajzi helyzet meghatározására, akkor mindezek mellett még figyelni kell a mágneses deklinációra is.

A Penguin robothoz beszerezhető szenzorok és egyéb modulok

A Penguin robot alapfelszereltségben mindösszesen a fentebb ismertetett három beépített érzékelőt tartalmazza, azonban alkalmazásának bővítésére több különféle, a Penguin és egyéb Parallax robotokhoz és mikrokontrollerekhez szánt szenzor megvásárolható.

Sajnos, a Parallax robotok közül a Penguin robot a legkevésbé bővíthető, mivel csak egyetlen szabad I/O port-tal rendelkezik. A B. kiadású (Rev.B) Penguin roboton már két szabad I/O port van és az akár már két darab egy csatornás (1 I/O láb) vagy egy darab két csatornás (2 I/O láb) modullal is bővíthető, például rádiófrekvenciás vagy Bluetooth kommunikációs, giroszkóp vagy gyorsulás- és elmozdulás mérő modulokkal. A Függelékben megtalálható a Rev.A és a Rev.B kiadású elektronikáknak a fényképe. Az itt felsorolt helymeghatározó rendszer modul és a szenzorok az A. kiadású (Rev.A) Penguin robothoz használhatóak.

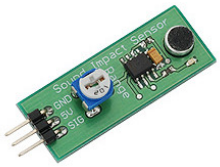
PING ultrahang szenzor modul

(PING UltraSonic Sensor) **FELHASZNÁLVA**

A PING szenzort a Penguin robothoz is ajánlja a Parallax. A PING egy könnyen kezelhető reflexiós elven működő ultrahangos távolságmérő (aktív) érzékelő eszköz. A szenzor a 2 - 330 centiméter közötti tartományban ad pontos

mérést. A szenzor piezokerámiából készült ultrahangos adó és vevőegységet tartalmaz, mely 40 KHz-s frekvencián működik. A mérés folyamatát a felül elhelyezett led világítása mutatja.

Hangérzékelő modul



(Sound Impact Sensor)

Az érzékelő a beépített mikrofonnal az olyan hangos zajokat észleli, mint például a tapsolás. Csak a változásokat érzékeli, az észlelést 1 biten tárolva (igen/nem), legfeljebb 3 méteres hatótávolságban, a hangerősséget azonban nem tudja mérni. A környezeti zajokból adódó téves érzékelés megelőzősére, az érzékenység egy apró kézi potenciométerrel állítható.

ColorPAL színfelismerő szenzor modul

(ColorPAL Sensor)

A ColorPAL szenzor egy szín- és fényérzékelő, amely az RGB színskála (Piros, Zöld, Kék) színei alapján meg tudja határozni a vizsgált objektum színét. A színérzékeléshez a beépített RGB LED színkomponensenként megvilágítja az objektum felületét közelről és külön-külön megméri a visszaverődött fényeket. A három mérés és a színkeverés pontosságát javító színreferencia algoritmusok (fekete, fehér és árnyékban mért) alapján kikeveri a leolvasott színt. A mért szín nagy felbontású (24 biten tárolva). Az érzékelő modulon található beépített memóriában az RGB méréshez és színkeveréshez tárolhatunk saját programokat.

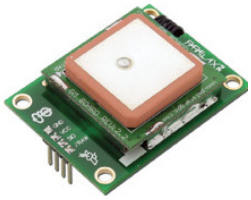
PIR mozgásérzékelő modul

(Passive Infra-Red Sensor)

A PIR szenzor egy olyan passzív piroelektromos érzékelő eszköz, ami a testek által kibocsájtott közép és hosszú hullámhosszú infravörös hősugárzás hatására ellenállását változtatja. Ezt a változást alakítja át az elektronika digitális jellé. Csak a változásokat észleli legfeljebb 6 méteres hatótávolságban és az észlelést 1 biten tárolja (igen/nem). A mozgásérzékelő szenzor a kibocsájtott infravörös hősugárzás intenzitását mérni nem tudja.

GPS vevő modul

(Parallax GPS Receiver Module)



A GPS vevő modul egy beépített antenna és egy társ-processzor segítségével képes akár 12 műhold sugárzott jelét is venni az NMEA 0183 kommunikációs protokoll szerinti adatmondatok vagy más, specifikus adatok formájában. Az NMEA 0183 egy nemzetközi hajózási elektronikai szabvány, amelyet GPS vevők, radarok, szonárok illetve számítógépes rendszerek összekapcsolásra alkalmaznak. Ez a GPS vevő legjobban a szabadban használható, amihez tiszta égboltra is szükség lehet a pontos lokalizációhoz.

Univerzális távirányító

(Universal Remote Controller) **FELHASZNÁLVA**

Az univerzális távirányítóval, egy kapcsolóegység segítségével rengeteg féle készülék (televízió, videó, DVD lejátszó, HIFI és egyéb eszköz) irányítható, a memóriájában eltárolt vezérlőkódok segítségével. Az univerzális távirányítóval - a Sony televíziókhoz és videó rögzítőkhöz alkalmazott utasításkészlettel lehet vezérelni a Penguin robotot is.

Rádiófrekvenciás adó és vevő

(Parallax 433 MHz RF Receiver and Transmitter)

A Penguin robotot távolról irányítani lehet egy rádiófrekvenciás vevő egységen keresztül, amire a rádiófrekvenciás adó küldi az adatokat és/vagy az utasításokat. Fordítva is lehetséges, ha az adó egység van a Penguin robothoz csatlakoztatva és a robot utasításait követi egy másik robot vagy robotok. A másik adó vagy vevő egységhez egy másik robot/próbapanel szükséges.

A beszerezhető szenzorok közül a PING szenzort és egy (nem Parallax) univerzális távirányítót használtam fel.

A programozási nyelv és a környezet

A PBASIC mikrokontroller programozási nyelv, alkalmazása és előnyei

A Penguin robot a Parallax által a Basic Stamp mikrokontrollerek programozására kifejlesztett PBASIC magas szintű programozási nyelven programozható.

Alapja a népszerű BASIC programozási nyelv. Azért volt a kezdetek óta szükség egy magas szintű felhasználó közeli programozás nyelvre, mert a gépi kódú és az ahhoz közeli assembly programozás nehezebben tanulható és tanítható.

A továbbiakban - a teljesség igénye nélkül - rövid áttekintést adok a PBASIC nyelv lehetőségeiről, tulajdonságairól és előnyeiről. Megjegyzem, a teljes PBASIC referencia könyv legfrissebb, 2.2-s bővített változata 504. oldal terjedelmű.

- Három kiadása készült eddig a PBASIC nyelvnek és ez mindegyik új forráskód írása előtt direktívaként kötelezően kiválasztandó, az adott Basic Stamp típusával együtt. Ez azért van így, mert a frissített kiadások az újabb, és újabb Basic Stamp mikrokontrollerekkel együtt új utasításokkal bővültek, a régebbi típusokhoz pedig az előző kiadások használhatóak. A Penguin robot programozásánál, a BS2px mikrovezérlőnek köszönhetően a használt verzió, a legfrissebb kiadás, a PBASIC 2.5

  ' {$STAMP BS2px}
  ' {$PBASIC 2.5}

- A PBASIC egyik előnye, hogy a BASIC alaputasításai megmaradtak. Az elágaztató (branch, if-else, select-case), az ismétlő (do-loop, for-next, do-while) és az ugró (goto, gosub) utasítások szerepelnek benne, ezek mellé kerültek be az EEPROM és RAM, az I/O kezelő, a szinkron és aszinkron soros és párhuzamos kommunikációs és egyéb speciális parancsok. Ilyenek például az időzítőket és hangokat kezelő speciális parancsok, az energiagazdálkodási és a hiba visszakeresési és ellenőrzési (Debug) utasítások.

- Konstansokat (CON), változókat (VAR) és akár I/O-kat (PIN) is definiálhatunk és nevesíthetünk. A változók és konstansok definiálásakor figyelni kell arra, hogy maximum 16 bites számokat tárolhatunk le, tehát 0 és 65535 között értéket adhatunk meg legfeljebb. Ez nem azt jelenti, hogy a mikrovezérlő mikroprocesszora nem tud ennél nagyobb számokkal is dolgozni. Az értékek több számrendszerben is megadhatóak, a bináris számrendszertől kezdve a hexadecimálisig.
- A programblokkok címkézhetőek, így például ugró utasításokkal elérhetőek vagy többször is felhasználhatóak.

A gyakorlati alkalmazás tárgyalásakor a kódrészletekben bővebben is bemutatom a PBASIC nyelv felhasznált elemeit.

A PBASIC egy interpretált nyelv, azaz a fordítási folyamat futásidőben történik. Az interpreter egy úgynevezett értelmező eszköz. Lehet szoftveres vagy hardveres felépítésű, a Basic Stamp hardveres interpreter-t használ, egy különálló integrált áramkört, a PBASIC Interpreter Chip-t.

A forráskód megírása és a szintaktikai ellenőrzés után egy tokenizálási folyamat történik, ami lényegében egy „nyelvészeti” eljárás. A tokenizáló egy tárgykódot generál. A tárgykódban nincs benne a kommentezés, az utasítások egyszerűbb utasításokra vannak szétszedve. A tárgykód feltölthető a mikrokontroller programkód memóriájába. Futtatás közben a PBASIC interpreter chip a tokenizált tárgykód utasításait, a felismert tokeneket lefordítja gépi kódra és a mikroprocesszor azonnal végrehajtja.

A forráskódban a ' karakterrel a sorok végén megjegyzések tehetőek és egyetlen sorba több utasítás egymás után a **:** karakterrel elválasztva adható meg.

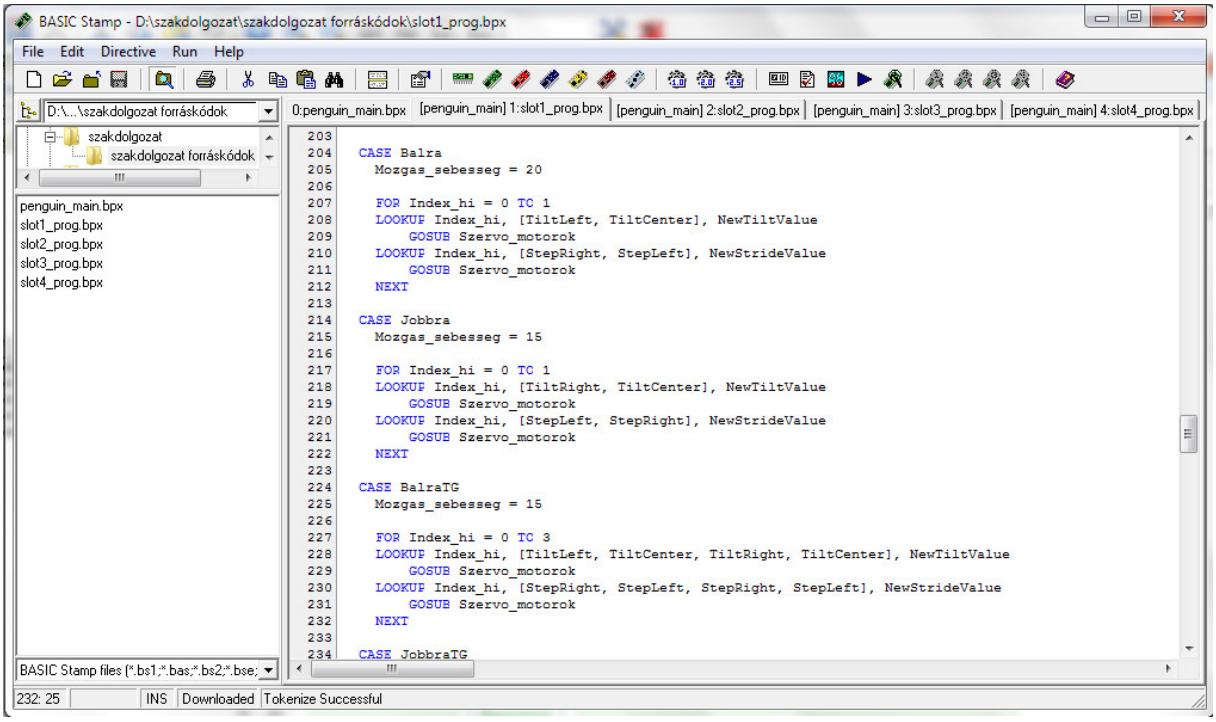
A programozási környezet

A Basic Stamp mikrokontrollerekhez a Basic Stamp Editor (továbbiakban BSE) szerkesztő programmal készíthetőek forráskódok, emellett a számítógép a BSE segítségével kommunikálhat a mikrokontrollerrel USB kapcsolaton keresztül.

A BSE minimális rendszer követelményei:

- Egy számítógép és egy BASIC Stamp mikrovezérlő,
- Microsoft Windows operációs rendszer (Windows 2000/XP vagy újabb),
- legalább 10 MB hely a merevlemezen (ennél többre nincs is szükség),
- a mikrokontroller csatlakoztatásához USB csatlakozó
- és az FTDI 232RL típusú USB vezérlőhöz telepítőprogram (driver).

A BSE Windows grafikus felületű, így a jól ismert környezetnek köszönhetően használata gyorsan elsajátítható.



24. ábra, A BASIC STAMP Editor Windows 7 alatt (forrás [x.])

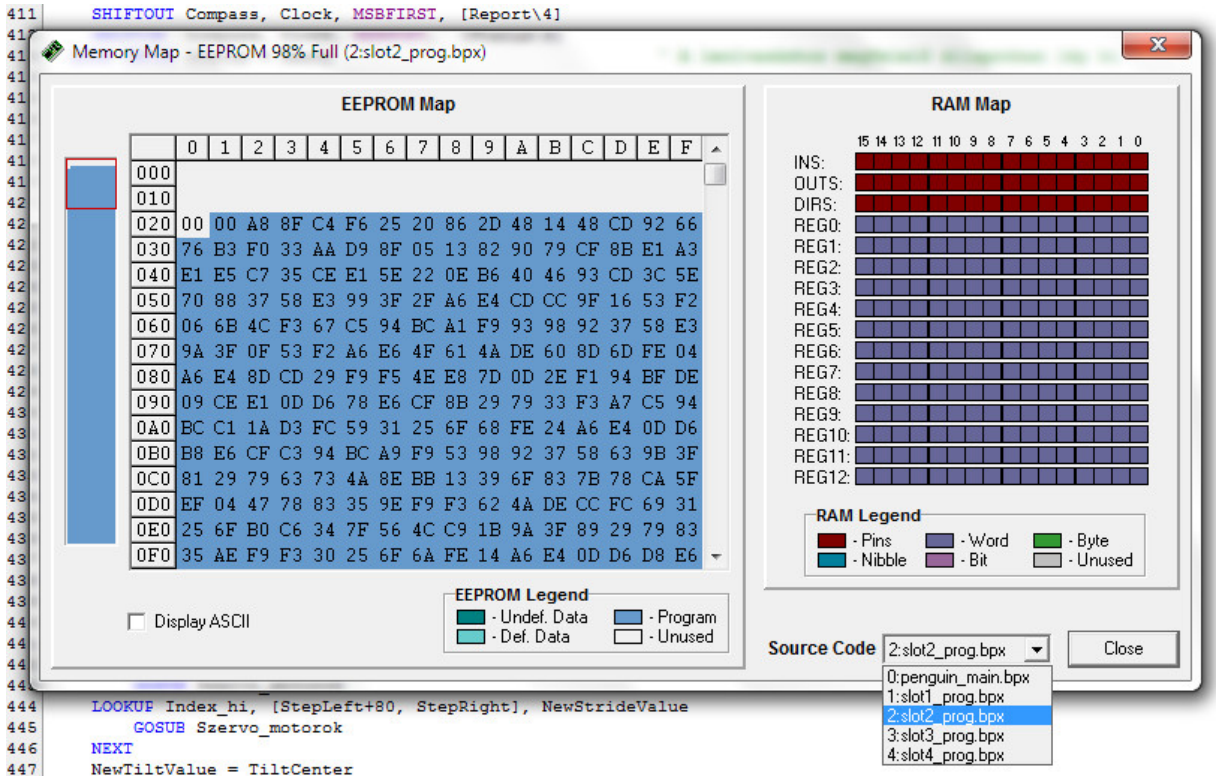
A közérthető szerkesztést és gyors visszakeresését elősegíti a szintaktikai szöveg kiemelés különböző színekben. A kód szerkesztéséhez kettéválasztható a szerkesztőablak, így ugyanannak a programkódnak az eleje és a vége is böngészhető és írható. A sorok számozhatóak és akár könyvjelzővel megjelölhetőek is. A nevesítések gyorsan, a forráskódban akár minden fellelhető helyen, egyszerre lecserélhetőek. A BSE ablak bal szélén

a Windows Intézőhöz hasonlóan kereshetünk a mappákban egyszerűen a forráskód fájlok között. A beépített formázási lehetőségeknek köszönhetően a forráskód kinézete könnyen alakítható. A felső ikonsoron választhatóak ki a fordítónak szánt PBASIC verzióra és Basic Stamp típusra vonatkozó direktívák. Emellett a szerkesztő programhoz és a PBASIC nyelvhez hasznos és bőséges súgó (PBASIC Syntax Guide) is rendelkezésre áll.

A portvezérlő segédprogram segítségével automatikusan felismertethető vagy kézzel beállítható, hogy melyik kommunikációs portra csatlakozzon a felismert Basic Stamp modul. A feltöltés aktuális állapotát folyamatablak mutatja.

A fejlesztők rendelkezésére áll a fejlett hiba visszakeresés/ellenőrzés, (Debug Terminal) amivel kommunikálni is lehet a mikrovezérlővel és ellenőrizni programjaink viselkedését.

A BSE-ben található egy Memória Térkép (Memory Map) nevű segédprogram is, amely az aktuálisan szerkesztett program memóriaigényét mutatja színes szemléltető ábrán. Külön látható a RAM és az EEPROM memóriák kihasználtsága (a Scratchpad RAM memória tartalma azonban nem). A memória térkép azonban csak virtuális memóriaigényt mutat, nem pedig a mikrokontroller memóriájából olvas!



25. ábra, A memória térképe az egyik navigációs programnak (forrás [x.])

Program- és forráskód optimalizáció és memóriagazdálkodás

A mikrovezérlő programozás során gyakran találkozhatunk olyan jelenségekkel, hogy

- gyorsan elfogyott a szabadon rendelkezésre álló adatmemória,
- vagy, nincs elég hely programunknak a programkód memóriában.

A Penguin robot mikrovezérlője esetén, programonként 2 Kbyte (a 16Kbyte nyolc részre bontva) felhasználható programkód memória és 32 Byte változóknak fenntartott RAM adatmemória található, ami legfeljebb 13 darab (9 Byte foglalt) szabadon felhasználható, 16 bites (és kisebb részekre bontható) változónak adhat helyet. Összetettebb programoknál a 26 Byte (a 13 darab Word változó) felhasználását jól át kell gondolni és a különböző szubrutinokban újrahasznosítani őket. A programkód memóriát ennél nehezebb teljesen lefoglalni, de például a DEBUG parancsok során kiíratott ASCII karakteres szövegek, a FREQOUT parancsok vagy a hosszan felsorolt ismétlődő utasítások nagyon sok helyet elfoglalnak.

Gyakran olvasható a különböző mikrokontroller programozással kapcsolatos szakirodalomban, hogy a modern számítógépeken nevelkedett programozóktól egy teljesen új szemléletet követel meg a memóriakezelés terén. Ez teljesen igaz, hiszen, míg egy számítógépes alkalmazásban *szinte* korlátlanul lehet gazdálkodni például változókkal a memóriában, addig a mikrovezérlők esetén ez a teljes projekt átgondolására kényszeríti a programozót.

A programok optimalizálhatóak és így végrehajtási sebességük is gyorsítható, ha rekurziót és ugró utasításokat használunk (GOSUB, GOTO) vagy például a változókat alias nevekkel foglaljuk le. Az ismétlődő utasításokat felsorolásuk helyett tehetjük inkább ciklusokba vagy például a LOOKUP parancs segítségével index szerint elágaztató parancsba is. Emellett a program-, illetve forráskód optimalizációnak köszönhetően egy nagyobb és bonyolultabb program is elfér kevesebb helyen. Példának az egyik forráskódomat mondanám, aminek, - kezdeti állapotának - mérete a memóriában (ha elfért volna 2 KByte-n) az optimalizált nyolc-kilencszerese volt.

Robot navigáció

A navigáció definíciója és célja

A szó a latin navigatio, eredeti jelentéséből a hajózásból ered. A navigációt és mögöttes módszereit a tengeri, a légi és a szárazföldi közlekedésben alkalmazzák tájékozódás céljából.

Lényege, - egyszerűen megfogalmazva - A pontból megfelelő navigáció segítségével eljutni, - lehetőleg a legrövidebb úton - B pontba. Napjainkban nagyon fontos a mobil robotok fejlesztésekor, hogy autonómak azaz, önállóak legyenek. Az ilyen típusú robotok nagyon jól hasznosíthatóak az iparban, a tudományos kutatásokban, vagy akár csak a mindennapokban.

A navigáció alapfeladatai:

- a pillanatnyi tartózkodási hely meghatározása,
- a rendeltetési helyet megközelítő optimális útvonal meghatározása,
- a jármű tervezett útvonalon való vezetése, irányítása. [26.]

A robotok számára a tájékozódás azonban nem olyan könnyű feladat, mint az embereknek. Az ember érzékszerveivel kevés információ alapján, nagyon rövid idő alatt, pontosan felméri a környezetét, míg egy gyors számítógép sok információból is, jóval lassabban, csekély pontossággal tájékozódik, sőt nem minden esetben megfelelően.

A mai mobil robotok különféle érzékelőket, helymeghatározó rendszert, képi látást megvalósító kamerákat is tartalmazhatnak, navigációjuk mégis nehézkes és sokszor bonyolult algoritmusokat igényelnek strukturálatlan környezetben. A strukturálatlan környezet, a robot navigációja szempontjából olyan környezet, ahol a terepviszonyok, a rajta lévő objektumok, fényviszonyok véletlenszerűen változnak (például a Mars bolygón lévő kutatórobotok esetében), és egy kötött pályás, rögzített lépésekből és mozdulatokból álló algoritmus nem használható. Ezen akadályok leküzdésében sokat lendítenek előre a robotika robot-navigációs területén folyó mesterséges intelligencia kutatások is. De nem szabad elfelejtkezni arról sem, hogy egy strukturált környezet sem jelent mindig garanciát arra, hogy a robot tökéletesen végrehajtsa benne feladatát.

Mobil (autonóm) robot navigáció

A mobil robotok szabadon mozoghatnak környezetükben, nincsenek lerögzítve valamilyen pozícióba. Munkavégzésük környezete lehet,

- a szárazföldi környezet, földfelszínen és föld alatt,
- a vízi környezet, a felszínen és a felszín alatt
- vagy a légi környezet, illetve a világűr.

A mobil robotok mindig saját tápellátással rendelkeznek és a mozgás szempontjából,

- kerekeken guruló robotok,
- lábakon lépegető vagy falmászó robotok,
- vízen úszó vagy merülő, illetve kétéltű (vízi-szárazföldi) robotok
- vagy repülő robotok lehetnek.

A mobil robotok navigáció és autonómitás szempontjából az alábbi csoportosítás szerint adhatóak meg:

Kézi vezérlésű robotok szenzorok nélkül

A kézi vezérlésű robot teljesen külső irányítás alatt áll. A külső vezérlő általában emberi személy vagy egy számítógép. Az irányító eszköz mobil robotoknál ritkább esetben vezetékkel összekötve, gyakoribb esetben valamilyen vezeték nélküli kommunikáció segítségével kommunikál a robottal, mint például a rádiós, a Bluetooth vagy az infravörös. A kézi irányítás előnye, hogy a robot pontosan oda megy, ahová a kezelője irányítja és csak a kezelő által kiadott utasításokat hajtja végre.

Kézi vezérlésű robotok szenzorokkal

Ezek a robotok annyiban térnek el kézi irányítású társaiktól, hogy képesek érzékelni a robot közelében lévő akadályokat és figyelmeztetni a robotot irányító személyt. A kézi vezérlést ez a tulajdonság megkönnyíti, ezért az ilyen robotok sokszor olyan helyzetekben is felhasználhatóak, ha a robotot irányító személy nem ismeri a terepviszonyokat. Ilyen robot például a katonai felderítő és bombakereső robot, az iRobot's PackBot.

Vonalkövető robotok

A vonalkövetés során a robot egy felrajzolt, festett vagy ragasztott útvonalat követ, ami elágazásokat is tartalmazhat. A vonalkövető robotok irányítása a vonalakra van bízva, és ez a technika az iparban rendkívül jól bevált és gyakorta alkalmazott. A vonalkövető robot azonban az előtte lévő akadályt nem kerüli ki, - a vonalak miatt értelemszerűen- csak megáll előtte és az akadály eltávolításáig várakozik.

Nézzünk például egy anyagmozgató robotot (targonca), amelynek a gyár területén különböző helyekre kell raklapokon tárolt alkatrészeket szállítania. Nem kis logisztikai feladat ennek a megvalósítása. Ebben lehet nagy segítség, ha a robotot felszereljük érzékelőkkel és beprogramozzuk a gondosan megtervezett útvonalak követésére. Így emberi munkaerő igénybevétele nélkül tud egyik helyről a másikra anyagot szállítani. Ilyen robotok például az FMC vagy a Transbotics útvonalkövető AGV robotjai is. (Automated Guided Vehicles)

Autonóm véletlenszerűen mozgó robot

Ezen robotok a szenzorok által érzékelt információkra valós időben reagálva módosítják mozgásirányukat, így az véletlenszerűen változik a robot környezetétől függően. Autonómok, vagyis teljesen függetlenek az emberi vagy a külső számítógépes irányítástól. Az akadályokat és a falakat szenzorokkal érzékelik, de nem terveznek a nyert információk alapján útvonalat. Az érzékelők egyik típusa a fizikai, úgynevezett ütköző érzékelők, amik a falakkal vagy akadályokkal történő fizikai érintkezés során jeleznek vissza, míg másik típusa az érintkezés nélküli érzékelők, amelyek az általuk kibocsájtott mérőjel visszaverődése alapján navigálnak az objektumok vagy falak közelében, mint például az ultrahangos, lézeres vagy infravörös szenzorok (aktív szenzorok).

Autonóm útvonaltervező robot

Az autonóm útvonaltervező robot útitervet készít aktuális pozíciójától az elérendő célpozícióig a szenzorokból, kamerákból és helymeghatározó rendszerekből nyert információk alapján, - gyakran szenzorfúzió segítségével - kiegészítve speciális pozícionáló, illetve térképkészítő algoritmusokkal (háromszögelés, Monte Carlo/Markov lokalizáció).

Navigációjuk során először meghatározzák saját pozíciójukat, majd kiszámítják a cél vagy az útvonal menti jelzők helyzetét és odanavigálnak. Gyakran alkalmaznak egy, illetve kettő

(vagy több) kamerás rendszereket a gépi látásra, amelyekkel szinte korlátlan információ áll rendelkezésre a robot látóteréből. Autonómok, tehát külső irányítástól teljesen függetlenek, önállóan képesek feladataik elvégzésére, mint például a saját merülő elemeik esetén azok automatikus feltöltése a töltő bázisukon. A veszélyes szituációkra és akadályokra reagálnak, ha szükséges leállnak. Képesek akár saját maguk megjavítására (bizonyos szintig) és rendelkezhetnek a tanulás képességével is. Több robot esetén sokszor hálózatba is kapcsolódnak. Ilyen robotok például a MobilRobots Patrolbot vagy a takarító robot, az iRobot's Roomba.

Vegyünk két példát, amivel szemléltethető a különbség a véletlenszerűen mozgó és az útvonaltervező autonóm mobil robotok között.

- Egy véletlenszerűen mozgó autonóm mobil robot elé hirtelen akadály kerül, az ultrahang érzékelő ezt jelzi is, megadva a robot objektumtól való távolságát. Ezek után a robot elfordul az akadály elől és másik irányba indul.
- Egy útvonaltervező autonóm mobil robot a GPS rendszer segítségével meghatározza aktuális pozícióját, digitális kamerás képalkotó rendszerével képet rögzít környezetéről, - aminek a jellegzetes pontjait elmenti digitális térképében - majd egy lokalizációs algoritmus és addig megtanult helyismereti adatainak segítségével kiszámítja a célobjektumhoz vezető optimális útvonalat, - figyelembe véve a lehetséges akadályokat a térképen- végül ezek alapján megközelíti a célpozíciót.

Kevert autonómiájú robotok

Ezek a robotok mind az autonóm, mind a kézi vezérlésű és mind a vonalkövető robotok tulajdonságaival rendelkezhetnek. A robot vezérlését, ha szükséges, külső emberi személy vagy számítógép is átveheti, vagy követhet útvonalat is, az amúgy teljesen önállóan működő szerkezet. Erre a típusra rengeteg példa létezik, hiszen a külső vezérlés és az autonóm viselkedés összeolvasztása kényelmes és praktikus megoldás.

Az egyik, talán legismertebb példája a kevert autonómiájú robotoknak a Mars felszínére 2004-ben megérkezett két amerikai űrszonda, a MER-A és MER-B kódnevű, Spirit és Opportunity fantázianevű rover (vándorló) robotok. Ezek a robotok olyan, folyamatosan frissített szoftverekkel vannak ellátva, amelyeknek köszönhetően földi irányítás hiányában vagy kommunikációs zavar esetén is képesek önálló cselekvésekre és vizsgálatokra.

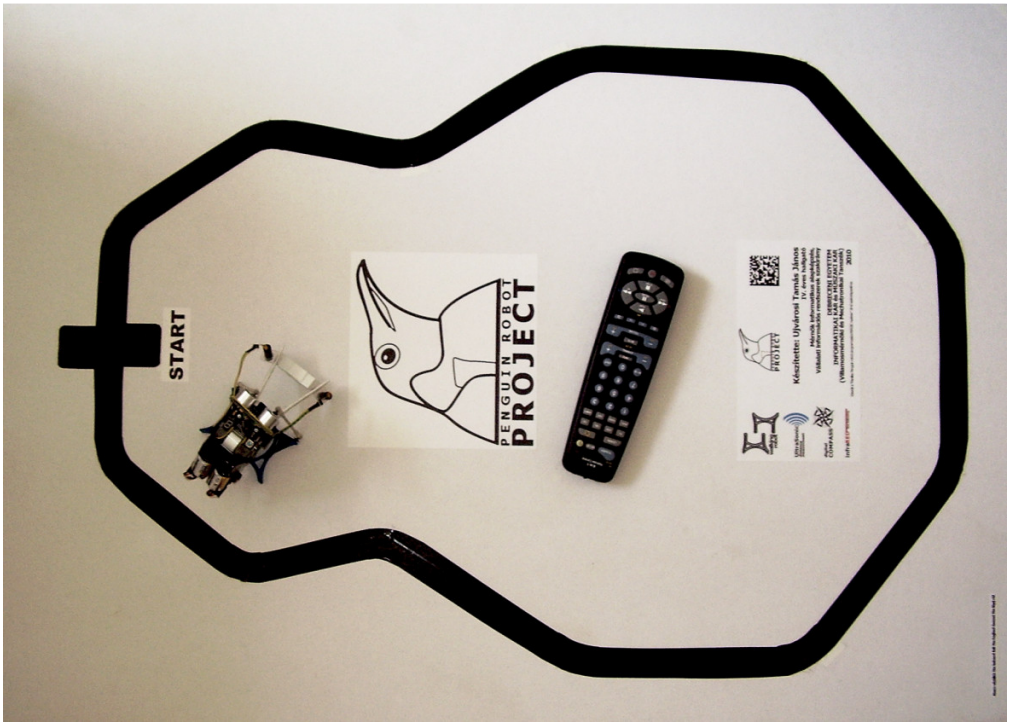
Gyakorlati megvalósítás a Penguin robottal

A gyakorlati részhez kapcsolódó kellékek és a megírt forráskódok a meglévő eszközök és lehetőségek szerint az általam megvalósított navigációs módszereket szemléltetik. A navigációs programok a Penguin robothoz készültek és azon működnek is. A Penguin robot a gyakorlati megvalósítás alapján robotgeneráció szempontjából második generációs, navigáció és autonómitás szempontjából kevert autonómiájú mobil robot.

Fontosnak tartom kiemelni, hogy a forráskódok és programok elkészítésében nagy segítséget nyújtott, hogy a Parallax rengeteg demonstrációs forráskódot elektronikusan közzétett.

A tesztpálya

A Penguin robot navigációjának demonstrálásához készítettem egy tesztpályát, amivel így egy strukturált terepet nyújtottam a robotnak. A programokban a tesztpályával kapcsolatos problémásabb igényeket ehhez igazítottam.



26. ábra, A tesztpálya felülnézetből, a START felirat mellett a robottal és az univerzális távirányítóval (forrás [x.])

A tesztpálya adatai:

- 100 cm hosszú, 70 cm széles karton
- tört fehér, sima felület,
- fekete szigetelő szalagból felragasztott ~25 mm széles csíkok útvonalkövetéshez.

A szervo motorokat kezelő szubrutin

A navigáláshoz az egyik alapvetően fontos programrész, a szervo motorokat pontosan kezelő szubrutin, ami több, különböző feladatokat végrehajtó programban is felhasználható. A robot bekapcsolásakor a szervo motorok mindig hirtelen megugranak, ezért nagyon fontos, hogy a Penguin robot motorjai bekapcsolás után középre legyenek állítva (centralizálva). Ez praktikussága mellett a robot szempontjából egy elővigyázatossági lépés is, azért hogy a motorok idő előtt ne menjenek tönkre.

Az alábbi forráskód részlet a mikro szervo motorokat kezelő szubrutint mutatja. A modul a két motor aktuális és új állapotának adatait és a sebességet várja bemenő adatként. Ezek alapján kiszámítja a motorok mozgatásának irányát (növekvő vagy csökkenő) és a sebességet figyelembe véve a lépésközt.

1.1 forráskód példa

```
StrideStep = NewStrideValue - OldStrideValue        ' léptető motor esetén
Stride_Elojel = StrideStep.BIT15

 StrideStep = ABS(StrideStep) / Mozgas_sebesseg     ' léptető motor lépéköz

IF Stride_Elojel THEN StrideStep = -StrideStep      ' léptető motor mozgásirány


TiltStep = NewTiltValue - OldTiltValue              ' billenő motor esetén

Tilt_Elojel = TiltStep.BIT15

 TiltStep = ABS(TiltStep) / Mozgas_sebesseg         'billenő motor lépésköz

IF Tilt_Elojel THEN TiltStep = -TiltStep            'billenő motor mozgásirány


FOR Index = 1 TO Mozgas_sebesseg                    ' a mozgássebesség a mérték
```

```
                OldStrideValue = OldStrideValue + StrideStep
                OldTiltValue = OldTiltValue + TiltStep
        PULSOUT Lepes_Szervo, OldStrideValue        ' léptető motor pulzusszélessége
        PULSOUT Billenes_Szervo, OldTiltValue       ' billenő motor pulzusszélessége
    PAUSE 15                                        ' lépésközök közötti szünet
NEXT
 NewStrideValue = OldStrideValue
 NewTiltValue = OldTiltValue
```

Ezt szubrutint építettem be és használtam fel a szervo motorok mozgatására mindegyik, a navigációs módszerekhez általam készített program forráskódjában.

Rögzített lépésekből álló mozgás

Az előre rögzített navigációs lépésekből álló program készítése igényli a legkevesebb és egyben a legtöbb információt is a környezetről. Míg a robot érzékelni nem képes semmit, addig a programozónak pontosan fel kell mérnie a robot helyett a terepet és olyan előre rögzített útvonalat készítenie, amin a robot végig tud haladni önmagára nézve biztonságosan.

A Penguin robottal kapcsolatban, a módszer alkalmazása során olyan problémák merültek fel ebben az esetben, mint például a talaj felszínének anyaga, a pályán elhelyezett objektumok és a pálya határai.

- A robot talpa csiszolt felületű festett fém, tehát a csiszolt, sima felületeken igen erősen csúszkálhat a robot, ráadásul két különböző anyag, mint például a csiszolt felület és a szigetelő szalag találkozásakor olyan mozdulatokat is tehet, amit a program nem tartalmaz. Ez abból következik, hogy a szigetelő szalagon viszonylag nagy a talpak tapadása, míg a csiszolt felületeken kicsi. Így a Penguin robot egyenes vonalú mozgásának iránya megtörhet a két felület határán. Ezen annyiban lehet segíteni, hogy lassabb haladási sebességet kell beállítani és az egyes talpakra eső súlyt kell növelni.

- A pályán elhelyezett objektumokat a robot előre programozott mozgás esetén nem érzékeli, így ha nekimegy valaminek, akkor elakad vagy módosul az útvonala.

- A pálya szélei ismeretlenek a robot előtt, mert azokat szenzorok nélkül nem tudja érzékelni. Ezért olyan lépéssorozatra van szükség, amivel biztosan nem lép vagy esik le a Penguin robot a pálya széleinél.

A pályához való alkalmazkodás szabályainak betartása mellett az egyszerűtől kezdve a bonyolultig bármilyen lépésekből álló útvonal elkészíthető.

Az alábbi forráskód részletben a robot hat lépést tesz előre, majd négyszer egy képzeletbeli tengely mentén kicsit jobbra fordul.

1.2 forráskód példa

```
FOR Index_hi = 0 TO 5
    Jaras_Irany = Elore :     GOSUB JarasIranyVegrehajtas      :   NEXT
FOR Index_hi = 0 TO 3
    Jaras_Irany = JobbraTG :   GOSUB JarasIranyVegrehajtas   :   NEXT
```

Útvonalkövetés infravörös szenzorral

A robottal a világos (pl. fehér) felületre vagy pályára felragasztott ellentétes színű csíkokat lehet követni. Az infravörös érzékelővel történő útvonalkövetés azon az elven alapszik, hogy az infravörös fény nem verődik vissza a fekete színű felületekről, mert a sötét színű anyagok elnyelik, míg a fehér/világos színűekről visszaverődik. A fényvisszaverődésen alapuló érzékelést reflexiós érzékelésnek nevezik. Az útvonalkövetést többféleképpen is meg lehet valósítani, azonban a Penguin robotnál több korlátot is figyelembe kellett vennem.

Az egyik korlát szerkezeti jellegű volt. A robot infravörös ledjeit az alap kiépítésben nem lehetett lehajlítani és elég közel helyezni a talajhoz. A másik korlát a Penguin robot mozgása, hiszen a lépegetés darabos mozgás, a robot nincs állandóan egyensúlyi helyzetben,

nem véletlenül az útvonalkövetés is leghatékonyabban kerekes robotokkal megvalósítható. A robot lépegetése miatt a mintavételezés nagyon körülményes.

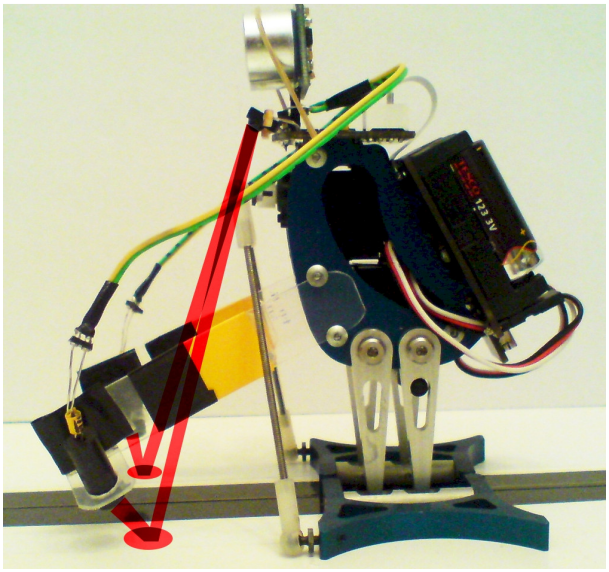
Az első problémát úgy oldottam meg, hogy a két infravörös ledet kiforrasztottam az áramköri panelről és helyükre tördelhető IC foglalatot tettem, amibe hosszabbító kábelekkel csatlakoztattam a ledeket, amik így már a talajhoz kellő közelségbe helyezhetőek lettek.

Ezek után már csak a ledeket tartó szerkezet elkészítése maradt hátra. Műanyagból készítettem egy könnyű és viszonylag rázkódásmentes tartószerkezetet az infravörös ledeknek, amiknek a foglalatát kis szögben lefelé hajlítva rögzítettem, hogy a ledek fényét rendesen észlelje az infravörös érzékelő. Fontos művelet volt emellett még a tartószerkezet és az infravörös ledek talajtól való távolságának pontos beállítása is.

A második korlát - a robot billegő járása miatt történő - pontatlan mérések kiküszöbölése volt. Ezt úgy oldottam meg, hogy minden egyes megtett lépés után a billenő szervo motort középre állítottam, amitől a robot vízszintes helyzetbe került, függetlenül a léptető motor aktuális pozíciójától és így a mérés alatt a robot infravörös ledjei is vízszintesen voltak, amitől a mérés minősége javult.

Korábban kísérleteztem az egyetlen infravörös ledes megoldással is. A robot mozgása azonban túlságosan cikk-cakkos volt, hiszen csak a vonal széléinek érintésére reagált és sajnos sokszor a billegések során elvesztette azt. Egyetlen leddel nem állapítható meg, hogy a robot milyen irányban tért le az útvonalról, ha az elkanyarodik.

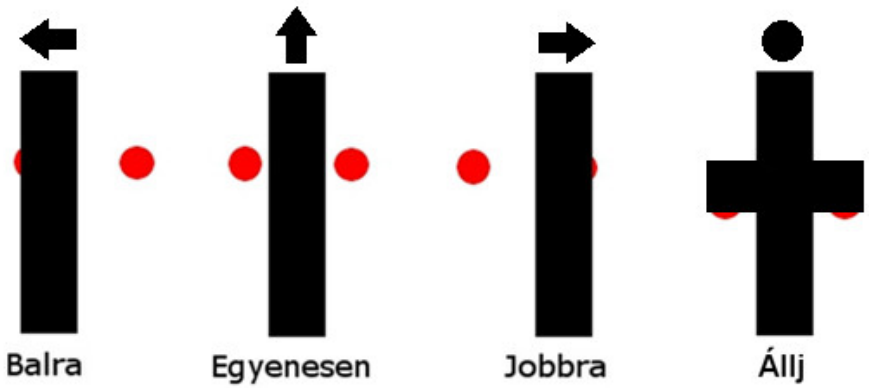
Az általam megvalósított két ledes megoldás esetén sokkal pontosabb az útvonalkövetés, mert a ledeket a ragasztószalag csíkok széleitől kintebb helyezve nem a fekete nyomvonalat, hanem a felület fehér részeiről történő visszaverődést vizsgálom. Ha az útvonal valamilyen irányban elkanyarodik, akkor egyszerűen az egyik led fénye nem verődik vissza és a robot ebbe az irányba fordul. Ha egyáltalán nincs visszaverődés, akkor a robot középre állítja a szervo motorokat és megáll. A vonal elhagyása (túl sok előre irányba tett lépés) és hosszabb idejű állás esetén a vezérlés egy idő után magától megszakítja az útvonal követési algoritmust.



27. ábra, Útvonalkövetés során a fehér felületről visszaverődik az infravörös fény (illusztráció, forrás [x.])

Abban az esetben, ha útvonalkövetés során a robot elé akadály kerül, akkor a robot megáll, és addig várakozik, míg az akadály előtte megszűnik. Az útvonal követő robotok az útvonalról nem térhetnek le, ezért kell várakozniuk. Az akadály észlelésére az ultrahang szenzort használtam, ami minden 15 cm-en belüli akadály esetén megállítja a robotot. Az ultrahang szenzor mérési folyamatáról és működéséről az Objektumérzékelés ultrahang szenzorral és digitális iránytűvel című fejezetben részletesebben is írok.

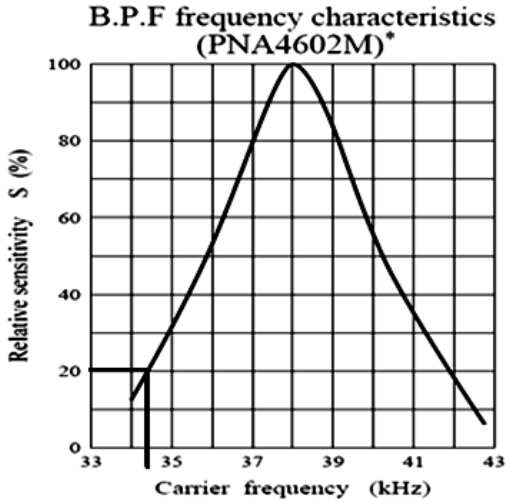
A lehetséges iránykombinációk száma 2^2, ezt az alábbi ábra pontosan szemlélteti. Tehát, ha valamelyik led fénye elnyelődik, akkor fordul, ha mindkettőé, akkor megáll, különben pedig előre lépked a robot. Attól függően fordul el, melyik oldali led fénye **NEM** verődik vissza.



28. ábra, Két infravörös ledes megoldás esetén a lehetséges haladási irányok (forrás [x.])

Az volt a tapasztalatom, hogy még két leddel is az útvonalkövetés megvalósítása nehéz feladat a Penguin robottal. Az optimális és ésszerű határok között követhető útvonal elkészítésénél több szempontot is figyelembe vettem.

- A pályán hegyes szögben kanyarodó útvonalról a robot könnyen lesétálhat.
- A robot a kidomborodó felületű csíkokon rosszul lépve elveszítheti az útvonalat.
- A tesztpálya és a csíkok anyagának minősége is rengeteget számít (szennyeződések)



29. ábra, Az infravörös érzékelő relatív érzékenysége a különböző hordozófrekvenciákra (forrás [17.])

Az alábbi forráskód részlet az útvonalkövetés irány és akadály ellenőrzési és döntési utasításait mutatja. A fenti karakterisztikán látható, hogy a 34,371KHz-s jelet csak ~20%-os érzékenységgel észleli az infravörös szenzor. Azért használtam kisebb frekvenciát a 38KHz-s vivőfrekvencia helyett, mert a nagyobb hordozófrekvenciájú infravörös fény sokszor a ragasztószalag fekete (de fényes) felületéről is érzékelhetően visszaverődött, míg a kisebb hordozófrekvenciájú már nem.

1.3 forráskód példa

```
DO
  GOSUB TaviranyitoEllenorzes                          ' távirányító ellenőrzése
  GOSUB UtvonalKovetes                                 ' döntés  a lehetséges haladási irányokról
    IF SzabadVar.LOWNIB=15 THEN RobotTaviranyitas  ' biztonsági számláló sok előre lépéskor
    IF   SzabadVar.NIB1 = 10 THEN RobotTaviranyitas   ' ÁLLJ biztonsági számláló álló helyzethez
Meres:
```

```
        GOSUB UltrahangosTavolsagMeres          ' akadály észlelése
                GOSUB TaviranyitoEllenorzes     ' távirányító ellenőrzése
         IF cm < 15 THEN Meres                  ' 15 cm-en belüli akadály esetén
LOOP
UtvonalKovetes:                                 ' döntés a lehetséges haladási irányokról
GOSUB Utvonal_Ellenorzes                        ' ledek ellenőrzése
 IF LeftIr = 1 AND RightIr = 1 THEN             ' Mindkét led fénye visszaverődik
    SzabadVar.LOWNIB = SzabadVar.LOWNIB + 1     ' biztonsági számláló növelése
    Jaras_Irany = Elore                         ' Előre
 ELSEIF LeftIR = 0 AND RightIr = 1 THEN         ' csak a bal oldali fénye nem verődik vissza
    SzabadVar.LOWNIB = 0                        ' biztonsági számláló 0
    Jaras_Irany = Balra                         ' Balra
 ELSEIF LeftIr = 1 AND RightIr = 0 THEN         ' csak a jobb oldali fénye nem verődik vissza
    SzabadVar.LOWNIB = 0                        ' biztonsági számláló 0
    Jaras_Irany = Jobbra                        ' Jobbra
 ELSE                                           ' egyik fénye sem verődik vissza
   SzabadVar.LOWNIB = 0                         ' biztonsági számláló 0
   SzabadVar.NIB1 = SzabadVar.NIB1 + 1          ' ÁLLJ biztonsági számláló növelése
   Jaras_Irany = STOPS                          ' ÁLLJ!
 ENDIF
GOSUB JarasIranyVegrehajtas                     ' a kiválasztott lépés végrehajtása
RETURN

Utvonal_Ellenorzes:
FREQOUT Lemitter, 2, 5700: LeftIr =IrInput^1   ' bal led világít 0,332 ms-ig 34,371KHz-n és az észlelés
FREQOUT Remitter, 2, 5700: RightIr=IrInput^1 ' jobb led világít 0,332 ms-ig 34,371KHz-n és az észlelés
RETURN
```

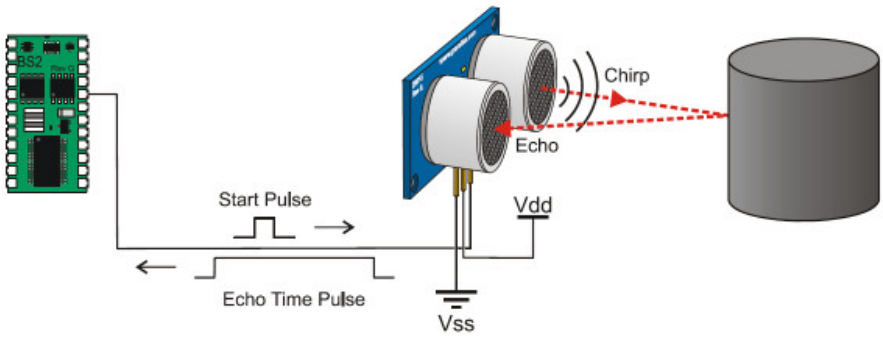
A láthatatlan fal

A láthatatlan fal navigáció az útvonalkövetés elvén működik. A robot csak a tesztpályán található útvonallal körbehatárolt területen vagy egyéb fekete szélekkel

körbehatárolt területen mozoghat, és ha az infravörös érzékelőből nyert információk alapján észleli a fekete ragasztó szalagból készült útvonalat vagy annak belső szélét, akkor ellentétes irányba fordul vagy visszalép, ha egyik infravörös led fénye sem verődik vissza. Azért hasonlítom láthatatlan falhoz, mert a robot úgy fordul vagy lép vissza, mintha valóban egy falat észlelt volna maga előtt. A láthatatlan falhoz hasonló az asztal széleinek észlelése, amikor a lefele irányba világító infravörös ledek fénye elvész az asztal széleinél. Ez azonban az elég törékeny robot szempontjából sokkal veszélyesebb mutatvány. A forráskód az útvonalkövető algoritmusból került átalakításra, a távolságmérés kihagyásával.

Objektumérzékelés ultrahang szenzorral és digitális iránytűvel

Egyszeri körbefordulás után a robot a legközelebbi objektum irányába fordul és megközelíti. A megközelítés előtt kötelezően kiválasztandó, hogy az objektum megközelítése automatikus vagy kézi vezérlésű módon történjen. Ehhez a feladathoz az ultrahang szenzort és a digitális iránytűt használtam. A robot (amennyire lehetséges) egy képzeletbeli tengely mentén lépkedve körbefordul és minden egyes megtett lépés után az ultrahang szenzorral távolság- és a digitális iránytűvel iránymérést végez. A memóriában a legközelebbi objektum távolsága és iránya tárolódik. Ha a körbefordulás-mérés közben az előzőnél kisebb távolságban is található objektum, akkor annak adataival felülíródnak az addig mentett értékek.



30. ábra, A PING ultrahang szenzor működési elve és bekötése (forrás [29.])

Az ultrahang az emberi fül számára nem hallható, 20000 Hz-nél magasabb rezgésszámú hang. Közismert, hogy többek között a denevérek és a delfinek is ultrahanggal

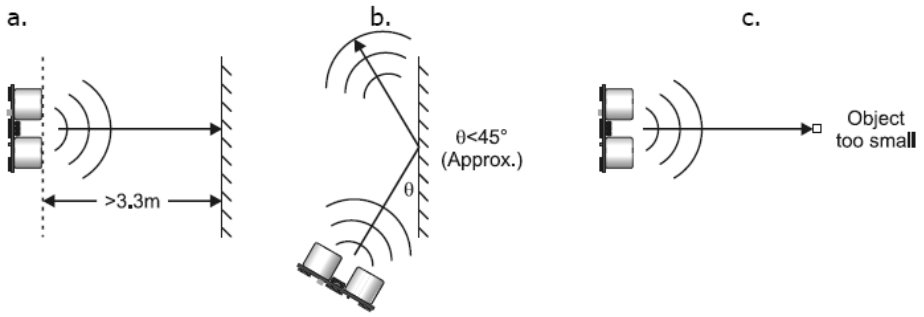
tájékozódnak. Az ultrahang szenzor egy ultrahangot kibocsájtó hangszóróból és egy ultrahangra érzékeny mikrofonból épül fel. A hangszóró kibocsájt egy - az észleléshez és a PULSIN parancs futtatásához elegendő idejű - 40 KHz-s ultrahangimpulzust (a Chirp). A visszaverődött ultrahangot a mikrofon észleli (az Echo) és ezzel a mérés befejeződik. A kibocsájtás és az észlelés között eltelt idő fele a visszaverődési idő, az úgynevezett echo idő.

A visszaverődés idejéből és az ultrahang levegőben mért terjedési sebességéből könnyen kiszámolható a távolság. Az alábbi képlettel a hőmérsékletet is figyelembe véve megadható a terjedési sebesség egy szobahőmérsékletű, 25° Celsius fokos szobára vagy teremre.

$$C_{levegő} = 331.5 + (0.6 \times T_C)\ m/s \quad [29.]$$

Egy szobahőmérsékletű teremben az ultrahang terjedési sebessége a levegőben 346,5 m/s.

Fontos tényező még a mérés során az objektumok alakja, mérete és távolsága. A szenzor használatakor be kell tartani a specifikációban megadott információkat, tehát csak a 2 és ~334 centiméter közötti tartományban mérhetünk (a. szituáció), ezen kívül értelmetlen eredményt kapunk. A másik hiba, ha túl nagy a beesési szög, akkor nem verődik vissza semmi és szintén értelmetlen eredménye lesz a mérésnek (b. szituáció). Hasonlóan hibás eredményt kapunk, ha az objektum mérete túl kicsi. Ez halmozottan igaz a Penguin robot esetében, ahol a PING szenzor 12 cm magasságban helyezkedik el a robot tetején, tehát csak az ennél kicsit kisebb és a magasabb objektumokról tud ultrahang visszaverődni (c. szituáció).



31. ábra, A PING szenzor és a hibás mérési helyzetek (forrás [29.])

Az ultrahang mérési programrészlet forráskódja viszonylag egyszerű. Az ultrahangimpulzus kibocsájtása, a visszaverődés mérése és a centiméterre történő átváltás mindössze öt sor.

1.4 forráskód példa

```
Ping = 0                                                    ' alacsony jel impulzus (0)
PULSOUT Ping, Trigger                                       ' aktív 13 usec-ig, magas jel impulzus (1)
PULSIN  Ping, 1, NyersAdatTavolsag                          ' az echo impulzusszélessége (1->0)=(0)
NyersAdatTavolsag = (NyersAdatTavolsag */ Scale) / 2        ' µsec-ba konvertálás és időfelezés
cm = NyersAdatTavolsag ** 2257          ' 1 / 29.034 (a ** a 32 bites érték felső 16 bitjét adja vissza)
```

A digitális iránytű feladatai a navigációs programban:

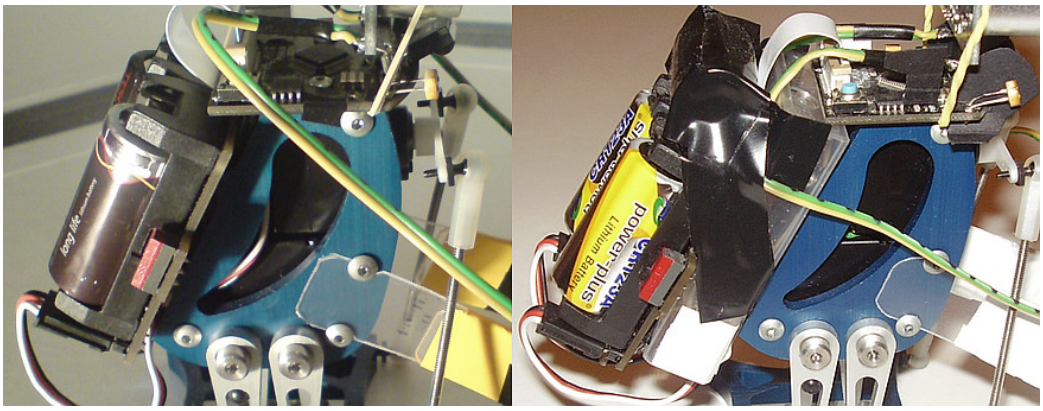
- A kezdő-, az aktuális és az objektum irány mérése,
- a robot egyszeri körbefordulásának észlelése,
- a legközelebbi objektum irányába történő fordulás ellenőrzése,
- és az automatikus megközelítés során az objektum irányának megtartása.

Sajnos, az iránytű két mérése közötti eltérés igen nagy lehet, ugyanabban a mérési pozícióban. Az eltérés akár 15° is lehet, de általában ennél kevesebb 4 - 6° az ingadozás. Ez többek között a robotot körülvevő - főként elektromos - eszközök által kibocsájtott mágneses zaj miatt van, mint például a robot közelében elhelyezkedő nagyfeszültségű vezetékek, motorok, vagy a tápellátást biztosító elemek nagy permeabilitású acél fémburkolata. Ezen problémák következménye lehet, hogy a robot nem teljesen az objektum pontos irányába áll be a mérések befejeződésével és emiatt az automatikus megközelítés során nem mindig pontosan az objektum valós irányába halad. Emellett a körbefordulás-mérés közben a robot talpainak csúszkálása is ront az automatikus megközelítésen. Az iránytű pontatlansága valamelyest azonban csökkenthető a többször kimért irányértékek - nagy részben - egyszerű átlagolásával. Az alábbi forráskód részlet az általam készített átlagolási algoritmust mutatja be.

1.5 forráskód példa

```
IranyAtlagolas:                         ' az iránytű mérési értékeinek átlagolása pontosítás céljából
    FOR Index_hi = 0 TO 8                           ' irány leolvasása és mentése 9-szer
      GOSUB Iranytu_leolvasasa                      ' irány leolvasása
        PUT Index_hi, Irany                         ' irány mentése az SP RAM-ba
    NEXT
          Irany_atlag = 0
```

```
IF Irany > 8 AND Irany < 248 THEN        ' korrigáló átlagolás 12°-347° közötti tartományban
   FOR Index_hi = 0 TO 8                          ' 9 irány érték átlagolása
      GET Index_hi, Irany                         ' irány betöltése változóba az SP RAM-ból
      Irany_atlag = Irany_atlag + Irany           ' irányok összegzése
   NEXT
     Irany = Irany_atlag / Index_hi               ' átlagszámítás
ELSE                                             ' közelítő átlagolás a 348° - 11° közötti tartományban
        Irany = 0  :  Index = 0
   FOR Index_hi = 0 TO 8                         ' 9 irány 0°-tól számított távolságának átlagolása
      GET Index_hi, Irany                         ' irány betöltése változóba az SP RAM-ból
      IF Irany > 235 AND Irany <= 255 THEN       ' HA: elméletileg:348° - 360°,gyakorlatilag:331° - 360°
           Irany = ~ Irany                       ' irányérték invertálása távolságnak pl: ~249 --> 6
           Index = Index + 1                     ' felső tartomány számláló
        ENDIF                                    ' HA vége, Különben: 0° - 11° tartománybeli kicsi érték
        Irany_atlag = Irany_atlag + Irany        ' 0 számított távolságok összege
   NEXT
        Irany_atlag = Irany_atlag / Index_hi     ' átlagszámítás
              IF Index > (Index_hi/2) THEN       ' ha több a felső tartománybeli (348-360) érték, akkor
                   Irany = 0 - Irany_atlag       ' 0 - átlag
              ELSE                               ' különben
                   Irany = 0 + Irany_atlag       ' 0 + átlag
              ENDIF
 ENDIF
RETURN
```



32. ábra, A Penguin roboton az elemtartó eredeti és az általam módosított pozíciója (forrás[x.])

Az elemek fémburkolata által keltett mágneses zaj azért is nagyon zavaró, mert nem tudtam könnyen megoldani ezt az akadályt. A probléma forrása az, hogy elemek burkolatának nagy a permeabilitása. (Sokszorosa a levegőének.) Emiatt az elemek a mágneses erővonalakat eltérítik az iránytűtől. Ezért az elemtartó és a robot törzse közé egy műanyag dobozt szereltem, így az elemtartó hátrébb és lentebb, lényegében az iránytűtől távolabb került, és ami igazán érdekes, hogy az elemtartó tetejére - a passzív mágneses árnyékolás tesztelésének céljából - egy keresztbe fektetett használt CR123-s elemet rögzítettem. Ezek után, tapasztalataim szerint a mérés pontossága nagy mértékben javult.

A digitális iránytű a Föld mágneses erővonalainak az érzékelő (x és y) tengelyeire párhuzamosan eső komponenseinek erősségét adja meg µTesla érzékenységben, egy 11 bites változóban letárolva. Az érték pozitív lesz, ha északi irányba mutat az iránytű, és negatív, ha a déli irányba. Egy 16 bites változót tehát negatív érték esetén maszkolni kell 11 bitesre. Ezután az x és az y tengely értékéből az arkusztangens függvény segítségével meghatározható az irány. Az arkusztangens függvény az arkusztangens (arc tg) általánosítása és arra alkalmas, hogy egy síkvektor y és x koordinátáiból - mindenképpen fordított sorrendben - kiszámítsa a vektor irányszögét (azaz az Északi irányba mutató erővonalakkal bezárt szögét jelen esetben), nulla és 2π között. Az irányszög kiszámítási képlete a következő:

arc tg(-y,x) és PBASIC parancsban x ATN -y

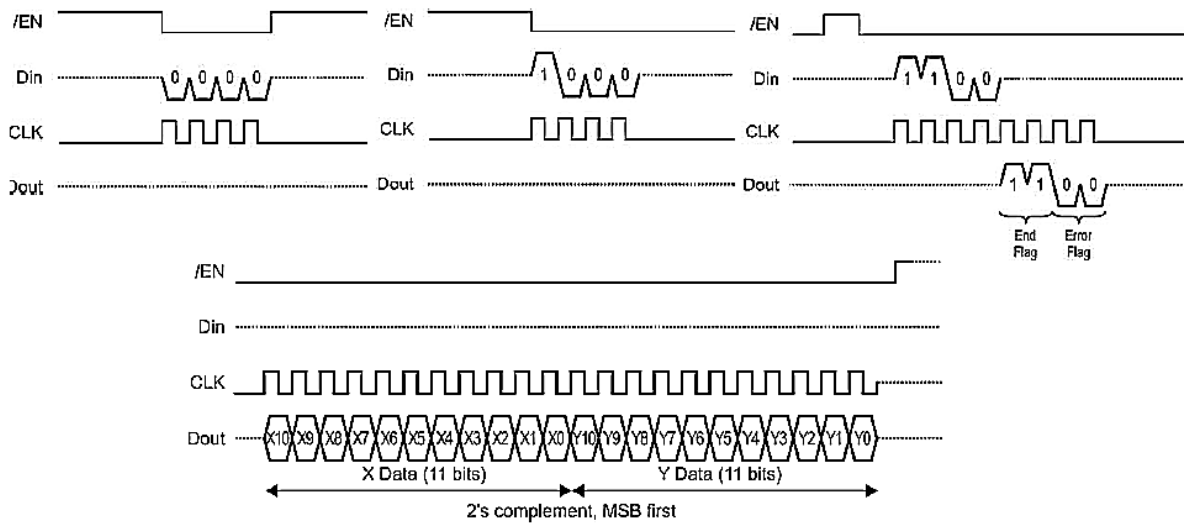
A kapott érték 8 bites lesz, a PBASIC-ben található ATN parancs végrehajtása után. A kimért irányt is így egy 8 bites változóban tárolom. Ez 0-255-ig terjedő skálát eredményez, ami nem a megszokott 0-360-ig terjedő fokrendszer. Erre azért volt szükség, mert a kör (folytonosság) tulajdonsága legjobban a változó túlcsordulással implementálható. Végül egy PBASIC bináris operátorral (a */ operátor, ami megfelel a (360/256) x 8_bites_irány képletnek) a 8 bites érték átváltható a 360°-s fokrendszerbe (az eredmény egy 16 bites szám lesz!!!).

Az alábbi forráskód részlet az iránytű leolvasási szubrutinját mutatja.

1.6 forráskód példa

```
HIGH Enable                                  ' magas impulzus - iránytű RESET
        LOW Enable                           ' alacsony impulzus
SHIFTOUT Compass, Clock, MSBFIRST, [Reset\4] ' %0000 küldése - reset jelző
HIGH Enable                                  ' magas impulzus
```

```
LOW Enable                                          ' alacsony impulzus - mérés kezdete
SHIFTOUT Compass, Clock, MSBFIRST, [Measure\4]      ' %1000 küldése - mérés kezdete jelző
status = 0

DO                                                  ' mérés állapotának ellenőrzése
  HIGH Enable                                       ' magas impulzus
        LOW Enable                                  ' alacsony impulzus
  SHIFTOUT Compass, Clock, MSBFIRST, [Report\4]     ' %1100 küldése, első bittől kezdve
  SHIFTIN  Compass, Clock, MSBPOST,  [Status\4]     ' Mérési státusz jelző érték fogadása, míg a
' mérés be nem fejeződik, tehát míg %1100 nem lesz a status, utolsó bittől kezdve
LOOP UNTIL status = Ready                           ' mérés kész, status = %1100

SHIFTIN Compass,Clock,MSBPOST, [x\11,y\11]     '  az értékek fogadása (22 bit), utolsó bittől kezdve
HIGH Enable                                         ' magas impulzus -- a mérés lezárva
IF (y.BIT10=1) THEN y=y | NegMask : IF (x.BIT10=1) THEN x=x | NegMask ' negatív értékek maszkolása
Irany = ( x ATN -y ) */ 360                         ' irányszög számítás és átváltás a 360°-s rendszerbe
```



33. ábra, A digitális iránytű mérési folyamatát szemléltető diagramok (balról jobbra sorrendben) (forrás[28.])

Az ultrahangos mérés során nem adódtak problémáim, pontosnak találtam a mért értékeket. Az iránytű mérési ingadozása ugyanakkor megnehezíti az objektum pontos megközelítését. Az alábbi forráskód részlet a teljes objektumkeresési algoritmust mutatja be,

kezdve a robot egyszeri körbefordulásától, a legközelebbi objektum távolságának és irányának rögzítésétől, majd annak irányába történő fordulásán át az objektum választható megközelítéséig.

1.7 forráskód példa

```
GOSUB IranyAtlagolas                                    ' kezdőirány leolvasása
        Irany_temp = Irany                              ' kezdőirány mentése változóba
                PUT 44, Irany_temp                      ' kezdőirány mentése SP RAM-ba
DO
  GOSUB TaviranyitoEllenorzes                           ' távirányító ellenőrzése
        GOSUB UltrahangosTavolsagMeres                  ' távolságmérés
                GOSUB IranyAtlagolas                    ' aktuális irány leolvasása
  IF cm < cm_temp THEN                                  ' legkisebb észlelt távolsággal felülírás
   PUT 77, Word cm, Irany                               ' mentés a SP RAM-ba
    cm_temp = cm  :  ENDIF                              ' változó felülírása

 IF Index_lo > 8 THEN                                   ' biztonsági számláló, a túl korai leállás ellen
 GET 44, Irany_temp                                     ' kezdőirány betöltése

 IF (Irany > Irany_temp - 4 AND Irany < Irany_temp + 4) THEN      ' ha a robot egyszer körbefordult
    GET 77, Word cm_temp, Irany_temp
  DO
      GOSUB TaviranyitoEllenorzes                       ' távirányító ellenőrzése
      GET 77, Word cm_temp, Irany_temp                  ' kezdeti körbefordulási irány újraírása
      GOSUB IranyAtlagolas                              ' aktuális irány leolvasása

      IF (Irany > Irany_temp - 4 AND Irany < Irany_temp + 4) THEN MegkozelitesValaszto
' miután elérte az objektum irányát, választható, az automatikus vagy kézi vezérlésű megközelítés
      GOSUB IranyDontes  :  GOSUB JarasIranyVegrehajtas
   LOOP
  ENDIF
 ENDIF
   Jaras_Irany = JobbraTG                               ' KÜLÖNBEN, forduljon körbe tovább
   GOSUB JarasIranyVegrehajtas
```

```
    Index_lo = Index_lo + 1                                  ' biztonsági számláló növelése
LOOP

MegkozelitesValaszto:                                        ' automatikus vagy kézi megközelítés
          GOSUB IR_TaviranyitoKod                            ' távirányítókód ellenőrzése
 SELECT remoteCode
  CASE VolUp   :  GOTO AutoMegkozelites                      ' AUTO megközelítés
  CASE VolDn   :  GOTO RobotTaviranyitas                     ' KÉZI vezérlésű megközelítés
 ENDSELECT
 GOTO MegkozelitesValaszto          ' különben a megközelítés választás újraellenőrzése

AutoMegkozelites:                                            ' célobjektum megközelítése
          Index_lo = 0              ' biztonsági számláló, ha a robot elvesztené az objektumot
  DO
    IF Index_lo = 30 THEN FOPROGRAM
    GOSUB TaviranyitoEllenorzes                              ' távirányító ellenőrzése
    GOSUB UltrahangosTavolsagMeres                           ' távolság mérés
    IF cm <= 15 THEN RobotTaviranyitas                       ' ha elérte az objektumot
    GOSUB IranyAtlagolas                                     ' aktuális irány meghatározása
    GET 79, Irany_temp                                       ' objektum irányának betöltése
      IF (Irany > Irany_temp - 2 AND Irany < Irany_temp + 2) THEN  ' a megfelelő irányba tartson
          Jaras_Irany = Elore                                ' egyenesen előre lépkedve közelíti meg
          Index_lo = Index_lo + 1                            '  2. biztonsági számláló növelése
      ELSE
          GOSUB Iranydontes                                  ' optimális fordulási irány az objektum felé
      ENDIF
    GOSUB JarasIranyVegrehajtas  :  PAUSE 150                ' lépés végrehajtása
  LOOP

IranyDontes:                                                 ' optimális fordulási irány az objektum felé
 IF (Irany_temp >= Irany) THEN                       ' ha nagyobb az objektum iránya, mint az aktuális
          IF ABS(Irany_temp - Irany) < 128 THEN              ' és különbségük kisebb, mint a félkör
           Jaras_Irany = JobbraTG                            ' akkor jobbra fordulj
```

```
        ELSE                                        ' különben
         Jaras_Irany = BalraTG                      ' balra
        ENDIF
  ELSE                                  ' ha kisebb az objektum iránya, mint az aktuális
        IF ABS(Irany_temp - Irany) < 128 THEN       ' és különbségük kisebb, mint a félkör
          Jaras_Irany = BalraTG                     ' akkor balra
        ELSE                                        ' különben
          Jaras_Irany = JobbraTG                    ' jobbra
        ENDIF
  ENDIF
RETURN
```

Akadály elkerülés

Ez a faladat mutatja be legjobban, hogyan képes a Penguin robot önálló navigációra (autonóm véletlenszerűen mozgás). A robot feladata, hogy a megfelelő irányba haladva kikerülje a vele szembe lévő akadályokat, majd tovább folytassa útját, addig, amíg el nem ér a pálya végéig. Az akadályokból a Penguin számára is könnyen bejárható labirintus készíthető.

A robot a program elején megméri a kezdőirányt a korábban részletezett digitális iránytűvel, majd elindul előre. Feladata, hogy amíg el nem ér a pálya végéig, addig tartsa ezt az irányt. Az akadályoknál, - amelyeket az ultrahang szenzorral észlel - el kellett döntenem, hogy milyen irányból kerülje ki a Penguin. Mivel az ultrahang szenzort nem tudom forgatni, ezért a robotnak kell elfordulnia. Alapértelmezett esetben a robot jobbra fordul, majd ellenőrzi, hogy abban az irányban van-e vele szemben akadály, ha nincs, akkor pár lépést tesz előre, ezzel kikerülve az akadályt. Végül az iránytűvel a program elején mért kezdőirányba visszafordul és halad tovább abba az irányba előre.

Azonban, ha a robot előtt a jobb oldali ellenőrzéskor újabb akadály található, akkor a Penguin először visszafordul a kezdőirányba, majd balra fordul, ahol, ha nincs előtte akadály, akkor előre lépegetve megkezdi az elkerülést. Ha bal oldalt is található akadály, akkor a robot

zsákutcába került, ahonnan hátrafele lépegetéssel tud kijönni, majd újra belekezd az oldalirányú akadályok ellenőrzésébe.

A pálya végén található keresztben egy széles fekete csík, ami a robot számára jelzi, hogy befejezheti az akadálykerülgetést. Ezt az infravörös érzékelővel tudja észlelni az útvonalkövetéshez hasonlóan. A program forráskódját az objetumkereső program algoritmusából alakítottam át. Az alábbi rövid részlet a kikerülési algoritmust mutatja.

1.8 forráskód példa

```
        GOSUB UltrahangosTavolsagMeres                          ' távolsámérés
IF cm < 14  THEN                                                ' akadály közvetlenül a robot előtt
  ElsoIrany:                                                    ' jobb oldali ellenőrzés
     FOR Index_lo = 0 TO 3                                      ' jobbra fordulás
      LOOKUP Index_lo, [STOPS, JobbraTG, JobbraTG], Jaras_Irany  :  GOSUB JarasIranyVegrehajtas
     NEXT
      GOSUB UltrahangosTavolsagMeres  :  IF cm > 16 THEN Kikerul        ' ellenőrzés
  MasikIrany:                                                   ' akadály volt jobb oldalt – bal oldali ellenőrzés
      GOSUB IranyAtlagolas : GET 44, Irany_temp                 ' iránymérés és kezdőirány beolvasás
   IF (Irany > Irany_temp - 5 AND Irany < Irany_temp + 5) THEN          ' megfelelő irány
      FOR Index_lo = 0 TO 3                                     ' balra fordulás
     LOOKUP Index_lo, [STOPS, BalraTG, BalraTG ], Jaras_Irany  :  GOSUB JarasIranyVegrehajtas
      NEXT
   ELSE                                                         ' kezdőirányba állás
      GOSUB IranyAtlagolas  :  GET 44, Irany_temp               ' iránymérés és kezdőirány beolvasás
      GOSUB IranyDontes  :  GOSUB JarasIranyVegrehajtas         ' fordulás a kezdőirányba
      GOTO MasikIrany                                           ' újraellenőrzés balra fordulásoz
   ENDIF
       GOSUB UltrahangosTavolsagMeres  :  IF cm < 17 THEN Zsakutca       ' zsákutca
  Kikerul:                                                      ' kikerülés szabad irányba
    FOR Index_lo = 0 TO 3                                       ' előre lépegetés és közben ellenőrzés
         GOSUB UltrahangosTavolsagMeres  :  IF cm < 10 THEN Akadalykerulgetes
      LOOKUP Index_lo, [Elore, Elore, Elore, Elore], Jaras_Irany  : GOSUB JarasIranyVegrehajtas
    NEXT
ENDIF
```

Fényforrás követése fényérzékelőkkel

A Penguin robot a két darab fényérzékelője segítségével a legerősebben világító fényforrást keresi és halad annak irányába. A program szobai világítási körülményekhez van beállítva. Ezen kritériumhoz igazodva a fényforrás lehet akár egy zseblámpa is, annyi kikötéssel, hogyha túl sötét (például: lekapcsolt lámpák) vagy ha túl világos (például: erősen besüt a nap) van a helyiségben, akkor a robot automatikusan megáll. A programba be van építve egy biztonsági számláló is, aminek az a feladata, hogy a robot egy bizonyos lépésszám után magától is befejezze a keresést.

A roboton található fényérzékelők fénytől függő fotóellenállások. Minél több fényt kapnak az érzékelők, annál kisebb lesz az ellenállásuk. Azért, hogy minél jobban megkülönböztethető legyen, hogy milyen irányból érkezik több fény, apró, fekete kartonlapokat rögzítettem az érzékelők mellé. A kapott értékek egymáshoz viszonyított mértékéből meghatározható, hogy melyik irányba forduljon a robot. Azonban, ha különbségük abszolút értéke elhanyagolható, vagyis az úgynevezett holttéren belül van az eredmény, akkor a robot egyenesen halad előre, megközelítve a fényforrást.

Két módon követheti a fényforrást a robot:

- véletlenszerűen, a környezeti fényektől függően,
- irányítottan, egy fényforrással szándékosan rávilágítva és azzal vezetve.

Az alábbi forráskód részlet ennek a navigációs feladatnak a mérési és döntési részét mutatja.

1.9 forráskód példa

```
DO
IF Index_lo > 240 THEN RobotTaviranyitas                    ' biztonsági számláló
        GOSUB TaviranyitoEllenorzes                         ' távirányító ellenőrzése
HIGH Bal_LDR  :  HIGH Jobb_LDR  :  PAUSE 3                  ' RC áramkörök előtöltése
RCTIME Bal_LDR, 1, BalLDR : RCTIME Jobb_LDR, 1, JobbLDR     ' ellenállások mérése
  IF ABS(BalLDR - JobbLDR) > (BalLDR + JobbLDR) ** Deadzone AND BalLDR > JobbLDR  THEN
        Jaras_Irany = Jobbra  :  Index_lo = Index_lo + 1    ' holt zónán kívüli érték és jobbra
  ELSEIF ABS(BalLDR - JobbLDR) > (BalLDR + JobbLDR) ** Deadzone AND BalLDR < JobbLDR THEN
        Jaras_Irany = Balra  :  Index_lo = Index_lo + 1     ' holt zónán kívüli érték és balra
```

```
  ELSE                                                    ' holt zónán belüli, elhanyagolható
        Jaras_Irany = Elore   :   Index_lo = Index_lo + 3      ' előre
  ENDIF
IF BalLDR >= 6000 AND JobbLDR >= 6000 OR BalLDR < 100 OR JobbLDR < 100 THEN ' túl sötét/világos
        Jaras_Irany = STOPS   :   Index_lo = Index_lo + 5      ' állj
ENDIF   :   GOSUB JarasIranyVegrehajtas                   ' lépések végrehajtása
LOOP
```

Robot távirányítás univerzális távirányítóval

Az univerzális távirányító és az infravörös érzékelő segítségével távirányítható a Penguin robot. A távirányítás során a robottal végrehajthatóak különféle mozgások és ezen felül kiválaszthatóak a navigációs programok. Emellett csak a programozó fantáziájára van bízva, hogy a gombkombinációkból milyen funkciókat valósít meg.

Az univerzális távirányítón először ki kell választani a Sony TV-khez használható (vezérlőkódot) módot a kódlista alapján. A távirányító az infravörös ledjét villogtatva 38,5 KHz-es vivőfrekvencián küldi a lenyomott gombok vezérlőkódjait. A ledek a pulzus szélesség modulált jelküldés miatt villognak bizonyos időközönként, bizonyos ideig. A távirányító egymás után 12 bitet sugároz, ebből a 2,4 ms idejű START jelzés után az első hét bit az utasítás, a lenyomott gomb kódja, a maradék öt bit azonban a robot szempontjából elhanyagolható. A bináris 0 időtartama 0,6 ms, a bináris 1-é ennek a duplája, 1,2 ms. A hosszan nyomva tartott gomb kódját 20-30 ms időközönként a távirányító újraküldi. [31.] Szerencsére, az infravörös érzékelő még szélsőséges helyzetekben is észleli a távirányító innen-onnan visszaverődő infravörös fényét. Így a Penguin robotot még távolról és mögötte elhelyezkedve is lehet távvezérelni.

Az alábbi forráskód részlet a távirányítón lenyomott gomb kódját olvassa le. Ez a mód azonban közvetlenül a szubrutinra ugorva megakasztja a program működését, hiszen addig várakozik, míg kellő pulzusszélességű jelimpulzust nem mér. Ezért a leolvasás és a főprogram közé be kellett helyeznem egy olyan szubrutint is, ami csak akkor indítja el a leolvasási

algoritmust, ha a távirányító START impulzusához hasonló hosszúságú impulzust mér az infravörös érzékelőn. Ezt a kódrészletet mindegyik navigációs programomba beépítettem.

1.10 forráskód példa

```
remoteCode = 0
DO
      PULSIN IrInput, 0, irPulse              ' pulzus szélesség mérése
 LOOP UNTIL irPulse > KuszobStart        ' 2400 µs hosszúságú szünettel kezdődik a kódküldés
FOR Index = 0 TO 6                            ' csak az első 7 bit a lenyomott gomb kódja !!!
   PULSIN IrInput, 0, irPulse   :  IF irPulse > KuszobPulse THEN remoteCode.LOWBIT(Index)=1    ' 1250
NEXT
 IF (remoteCode < 10) THEN remoteCode = remoteCode + 1 ' A kódhoz a megfelelő számokat rendeli.
 IF (remoteCode = 10) THEN remoteCode = 0
```

A távirányító jelenlétét érzékelő algoritmus többféleképpen is megoldható. A fentebb leírt, az impulzusszélességet mérő forráskód részlet nagyon egyszerű. A módszer előnye, hogy a távirányító START impulzusához van igazítva, így a véletlenszerű, rövid időtartamú infravörös impulzusokra nem érzékeny.

1.11 forráskód példa

```
    PULSIN IrInput, 0, IrPulse                      ' impulzusszélesség mérés az infravörös érzékelőn
    IF IrPulse > KuszobPulse THEN RobotTaviranyitas     ' valóban távirányító jele
```

A probléma másik megközelítéseként bemutatnék egy szintén nagyon egyszerű megoldást. Ez azon alapszik, hogy az infravörös érzékelőnek az infravörös fény észleléskor mérhető ellenállása lesz. Ez a módszer azonban gyakrabban eredményez téves észlelést.

1.12 forráskód példa

```
    RCTIME IrInput, 0, IrRes                    ' ellenállás mérése az infravörös érzékelőn
    IF IrRes > 0 THEN RobotTaviranyitas         ' valóban távirányító jele
```

A Függelékben megtalálható a navigációs programjaimhoz használható gombok képe az univerzális távirányítón.

Tápellátás

A tápellátás fontosságáról viszonylag kevés szó esik, pedig ez az egyik nélkülözhetetlen komponense a mobil robotoknak. Az ideális tápellátásnak egy mobil robot esetében az alábbi fontos tulajdonságokkal kell rendelkeznie:

- hosszú élettartamú
- könnyű elemcserét tesz lehetővé, egy jó helyre szerelt elemtartóval
- elérhető áron beszerezhető (az akkumlátorok hosszabb távon előnyösebbek)

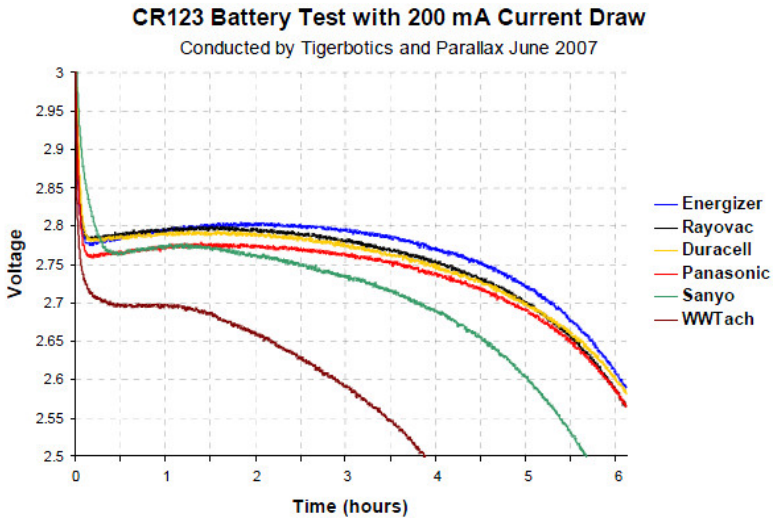
A Penguin robot tápellátását két darab CR123A típusú 3 Voltos lítium elem adja. A CR123A egy nagy kapacitású kifejezetten digitális kamerákhoz és fényképezőgépekhez készített elem típus. Kiskereskedelmi áruk - több beszerzési helyet és az Internetes kínálatot is figyelembe véve - elég magasnak mondható, kezdve az 500 Ft/darab ártól, az 1250 Ft/darab árig. Az általam kipróbált és felhasznált különböző márkájú CR123A típusú elemek:

- **Duracell Ultra M3**: Magas ár/magas élettartam. 2 darab/csomag
- **Rayovac**: Közepes ár/magas élettartam. Alapelem a készletben. 2 darab/csomag
- **Tesco Everyday**: Alacsony ár/közepes élettartam. 1 darab/csomag
- **Energizer Lithium Photo**: Közepes ár/magas élettartam. 1 darab/csomag
- **Power-Plus**: Alacsony ár/rövid élettartam. 1db/csomag

34. ábra, Az általam kipróbált CR123A elemek (forrás [x.])

A Parallax készített egy elemtesztet különböző gyártmányú CR123-s elemekkel és a Penguin robottal. Az elemteszt kiértékeléséhez hasonló eredményre jutottam én is. Nem feltétlenül szükséges a legdrágább elemet megvenni, ha a közel fél áron megvásárolható márkájú is hosszú élettartamú. A hivatalos tesztben nem szereplő, de általam használt Tesco Everyday

elemeket a Parallax teszten körülbelül a Sanyo elemek, míg a Power-Plus elemeket a WWTach elemek méréseredményei mellé sorolnám. (Tapasztalati úton!)



35. ábra, A Parallax által végzett terhelés teszt, különböző márkájú CR123A elemeken (forrás [21.])

Az elemek állapotáról a robot semmilyen figyelmeztető jelzést nem ad, ha azok a lemerülés határán vannak. Sok apró jelenséggel találkoztam a programozás során a merülő elemek miatt, amelyek sokszor idegesítő és értelmetlen dolgokat eredményeznek, mint például a hirtelen újraindulásokat, amikről az ember azt hiheti, hogy elrontott valamit a programozás során. A Függelékben található egy érdekes listakép a különböző feszültség szinteken jelentkező hibajelenségekről.

Az elemek élettartamának minimális meghosszabbítása érdekében használhatóak a programozás során energiagazdálkodási parancsok. Ilyenkor a mikrovezérlő áramfelvétele a működés közbeni 55mA helyett 450µA lesz. Az elemek élettartama meghosszabbítható továbbá még a roboton lévő visszajelző eszközök csökkentett használatával is, például a hét szegmenses kijelző állandó használatának mellőzésével. A működés hosszabbtávú folyamatosságának biztosítása céljából felszereltem a robotra egy kiegészítő elemtartót is, amelynek pólusait párhuzamosan kötöttem be az eredetivel. Így a robot továbbra is 6V feszültséget kap, de a dupla terhelhetőséggel hosszabb ideig képes üzemelni újraindulás és hibamentesen. Az egyetlen nehézség a pót elemtartó felszerelése volt, mert a robot két oldalára a digitális iránytű mérési hibái, az elejére pedig az infravörös ledek miatt nem lehetett felszerelni. Így végül az eredeti elemtartó fölé rögzítettem, úgy, hogy felbillentve lehet az eredeti tartóba betenni az elemeket.

Összefoglalás

Szakdolgozatom célja különböző navigációs módszerek és ötletek bemutatása és ezek közül néhány gyakorlati megvalósítása volt az általam választott Parallax Penguin robot segítségével. Ezen célkitűzésemet úgy érzem sikerült megvalósítanom.

A PBASIC programozási nyelv és a Penguin robot segítségével olyan navigációs programokat készítettem, amelyeknek köszönhetően a robot teljes mértékben kielégíti a második generációs robotokkal szemben támasztott követelményeket. Mindemellett a mobil robotok több típusát is demonstrálják az általam készített navigációs programok, bár összességében kevert autonómiájú mobil robotnak tekinthető a Penguin robot.

Az első programban a robot előre rögzített lépéseket hajt végre, miközben egyik érzékelőjét sem használja. A második program feladata szerint a robot egyszeri körbefordulással felméri a terepet, közelében lévő objektumokat keresve, majd a keresés után a hozzá legközelebb eső irányába fordul, és - az automatikus vagy a kézi vezérlést kiválasztva - megközelíti azt. A harmadik feladatban a robot akadályokat kerül ki, majd a pálya végén elhelyezett jelzésnél megáll. Ehhez a két programhoz az ultrahang, az infravörös szenzort és a digitális iránytűt használtam fel. A negyedik programban a Penguin robot a tesztpályán található fekete színű ragasztószalagból készített útvonalat követi. A következő programban a robot az útvonal által határolt területen mozog, úgy, mintha láthatatlan falakba ütközne. Ez utóbbi programokhoz az útvonalkövető tartószerkezetre rögzített infravörös ledeket és az infravörös érzékelőt hívtam segítségül. És végül, a hatodik navigációs programban a Penguin fényforrások felé lépeget. A feladathoz a roboton található fényérzékelőket használtam. Emellett mindegyik navigációs program - beleértve a főprogramot is - tartalmazza a távirányítóval történő kézi vezérlés lehetőségét, ezzel modellezve a külső beavatkozást. Így, mindegyik programban az alkalmazott szenzorok mellet az infravörös érzékelőt is használtam.

A programozás során törekedtem a minél ésszerűbben optimalizált forráskódok, illetve programok kivitelezésére, illetve az elemek élettartamának hosszabb távú fenntartására.

A navigációs feladatok elkészítése nálam a *tervezés* - *megvalósítás* - *tesztelés* fázisokból épült fel. A tervezés során felvázoltam, hogy milyen feladatot hajtson végre a robot és ehhez milyen programot készítsek, illetve, ha szükséges volt, akkor mit alakítsak át a roboton az

adott feladathoz, alkalmazkodva más feladatok és a komplex projekt igényeihez. A forráskódokban található verziószámok a programok nagyobb átdolgozásainak a számát jelölik. Például az útvonalkövetést vagy az objektum keresést rengetegszer át kellett gondolnom és párszor az elejétől kezdve is újraterveznem a felmerült akadályok miatt. A megvalósítás során próbáltam minél jobban követni a tervezés alatt felvázolt célokat, amennyiben megvalósíthatóak voltak. Emellett, a programozásban a kezdetek óta nagyon nagy segítségemre volt a szerkesztő program súgója is, mert a különböző PBASIC parancsok rengeteg specifikus tulajdonsággal rendelkeznek. A harmadik fázis, a tesztelés volt a leginkább időigényes, hiszen a Penguin robot mozgása elég lassú és a tesztelési fázisok monotonok, de mégis ez érte meg a legjobban, hiszen itt derültek ki a programban, a szerkezetben vagy akár a tervezésben rejlő hibák.

Összefoglalva, sikerült megvalósítanom néhány mobil robot navigációs feladatot ezzel az apró lépegető robottal, a dolgozatban rámutatva a felmerült problémákra és azok lehetséges megoldásaira. Emellett próbáltam minél részletesebben és közérthetőbben bemutatni a robotok és a robotika világát mind műszaki, mind történeti oldalról megközelítve, mindezeken felül egy kis kitekintést adva a mikrovezérlőkre és a robotikában betöltött szerepükre.

A robotika számomra egy nagyon izgalmas érdeklődési és egyben kutatási terület is. A robotprogramozás mellett, remélem a robotok szerkezeti tervezésével, építésével és átalakításával is lesz lehetőségem foglalkozni és még többet tanulni róluk a későbbiekben. A Penguin robot programozásával, illetve átalakításával eltöltött időt élveztem és mindenképpen hasznosnak tartottam.

A robotikát, a robotprogramozást és a Parallax robotokat őszintén tudom ajánlani, mind hobbi szinten, mind szakmai tudás megalapozásának céljából. Szerintem, ez a témakör rengeteg lehetőséget, tudást, munkát, késő éjszakai ébrenlétet és rengeteg örömet rejt magában, főleg, ha önmagunknak is sikerül valami újat és addig ismeretlent felfedezni benne.

Fejlesztési lehetőségek

Természetesen számos további fejlesztési lehetőség adódik a robottal kapcsolatban. Mindezek előtt, amit személy szerint ajánlanék, - általánosságban a robotfejlesztésekkel kapcsolatban - hogy legjobban egy csapat keretein belül lehet a fejlesztéseket végrehajtani, mert sokkal többre lehet jutni, mint ha az ember egymagában áll neki.

Az egyik lehetséges fejlesztési irány lehetne, ha például az ultrahang szenzort egy szervo motor tetején forgatnánk körbe, így gyorsítva és pontosítva a keresést vagy távolságmérést. Másik távlati irány lehetne a gyorsított útvonalkövetés vagy a robot rádiófrekvenciás kommunikációja. Esetleg a tápellátását is le lehetne cserélni olcsóbb energiaforrásra vagy a robot mozgását jobban segítő kerekes szervo motorokat lehetne felszerelni. Az internetes videómegosztókon is található pár érdekes példa átalakított Penguin robotokra.

Emellett, - az előbb említett fejlesztőcsapat keretein belül - a robothoz készülhetnének mesterséges intelligenciát is tartalmazó navigációs és egyéb algoritmusok vagy több Parallax robottal is összekapcsolható lehetne a Penguin robot.

Irodalomjegyzék

Utolsó ellenőrzés 2010. 10. 20.

[1.] R.U.R. (Rossum's Universal Robots) Wikipédia angol nyelvű

http://en.wikipedia.org/wiki/R.U.R._(Rossum's_Universal_Robots)

[2.] Merriam Webster Szótár angol nyelvű - Robot

http://www.merriam-webster.com/dictionary/robot

[3.] ISO 8373 szabvány, ipari robot definíciója, angol nyelvű

http://www.dira.dk/Portals/0/Robotter/robotdef.pdf

[4.] Tech Bytes - Your View: How would you define a robot?

http://www.cbc.ca/technology/technology-blog/2007/07/your_view_how_would_you_define.html

[5.] Robot - Wikipédia

http://hu.wikipedia.org/wiki/Robot

[6.] Kecskeméti Péter - Robottechnika

http://www.kando-kkt.sulinet.hu/brain/aut/robottech.ppt

[7.] Robotics - Wikipedia angol nyelvű

http://en.wikipedia.org/wiki/Robotics

[8.] Magyar Attila - Robotika

http://www.aut.vein.hu/oktatok/MagyarA/Rob1.pdf

[9.] A robotika három törvénye - Wikipédia

http://hu.wikipedia.org/wiki/A_robotika_három_törvénye

[10.] Kömlődi Ferenc (2007) - Autonóm mobil robotok

http://www.nhit.hu/data/101419/mobilrobotika3.2.doc

[11.] Merriam Webster Szótár - Microcontroller angol nyelvű

http://www.merriam-webster.com/dictionary/microcontroller

[12.] Industry Week - Joseph Engelberger fényképe

http://www.industryweek.com/slideshows/HallofFame2009/Joseph-Engelberger.jpg

[13.] International Federation of Robotics: Service Robots, angol nyelvű

http://www.ifr.org/service-robots/

[14.] IBM 7576 ipari robotkar képe

http://www.antenen.com/htdocs/robots/products/prodimages/ibm_7576_2.jpg

[15.] znn2010 blog - Isaac Asimov és budapestlight blog képek

http://znn2010.freeblog.hu/files/2010/01/Isaac_Asimov.jpg

http://www.tomandmaria.com/st197/images/asimov cover.jpg

[16.] Microcontroller - Wikipedia, angol nyelvű

http://en.wikipedia.org/wiki/Microcontroller

[17.] PNA4601M Series Infrared Receiver Documentation, angol nyelvű

http://www.parallax.com/Portals/0/Downloads/docs/prod/audiovis/PNA4601M.pdf

[18.] IFR Statistical Department, 2009.09.30. diagramok, angol nyelvű

http://www.ifr.org/uploads/media/Charts_30_09_2009_01.pdf

[19.] Parallax Robot képek: Scribbler, Boe-bot, SumoBot, QuadRover, Stingray, Penguin

http://www.parallax.com/Portals/0/Images/Prod/2/281/28136b.gif

http://www.parallax.com/Portals/0/Images/Prod/2/281/28136-L.jpg

http://www.parallax.com/Portals/0/Images/Prod/2/288/28832a-L.jpg

http://www.parallax.com/Portals/0/Images/Prod/2/288/28832-L.jpg

http://www.parallax.com/Portals/0/Images/Prod/2/274/27402a-L.jpg

http://www.parallax.com/Portals/0/Images/Prod/2/274/27400-L.jpg

http://www.parallax.com/Portals/0/Images/Prod/Q/QuadRover02_L.jpg

http://www.parallax.com/Portals/0/Images/Prod/Q/QuadRover05_L.jpg

http://www.parallax.com/Portals/0/Images/Prod/2/289/28980-L.jpg

http://www.parallax.com/Portals/0/Images/Prod/2/289/28980a-L.jpg

http://www.parallax.com/Portals/0/Images/Prod/2/273/27314-L.jpg

[20.] Botmag és Media Digikey Toddler robot képek, Alphaworks Penguin robot kép

http://www.botmag.com/issue7/images/middlepic2.jpg

http://media.digikey.com/photos/Parallax Photos/MFG_27316.jpg

http://alphaworks.ish-lyon.cnrs.fr/Documents/Digital_Media/3D_Models/Toddler/Toddler_3D_preview.png

[21.] Parallax Penguin Robot Documentation v 1.0-1.4, angol nyelvű

http://www.parallax.com/Portals/0/Downloads/docs/prod/robo/27313-6 PenguinDoc-v1.4.pdf

[22.] Parallax Basic Stamp BS2px modul képe

http://www.parallax.com/Portals/0/Images/Prod/B/BS2PX-IC-M.jpg

[23.] Pictutorials - What is microcontroller kép

http://www.pictutorials.com/microcontroller_architectur.gif

[24.] ROM Chip képe

http://doit.ort.org/course/procmem/images/379.gif

[25.] Hardwarezone - RAM chip képe

http://www.hardwarezone.com/img/data/articles/2004/1280/ram-chip.jpg

[26.] Navigáció - Wikipédia

http://hu.wikipedia.org/wiki/Navigáció

[27.] Vogel Miklós - *Ipari robotjárművek helymeghatározó rendszerei* - Műszerügyi és Méréstechnikai Közlemények 68. száma

http://www.muszeroldal.hu/MMK/nr68/pic/vogel2.gif

[28.] Parallax HM55B Digitális iránytű dokumentáció, angol nyelvű

http://www.parallax.com/Portals/0/Downloads/docs/prod/compshop/HM55BModDocs.pdf

[29.] Parallax PING Szenzor dokumentáció, angol nyelvű

http://www.parallax.com/Portals/0/Downloads/docs/prod/acc/28015-PING-v1.6.pdf

[30.] IEEE Spectrum, angol nyelvű

http://spectrum.ieee.org/automaton/robotics/industrial-robots/041410-world-robot-population

http://spectrum.ieee.org/automaton/robotics/robotics-software/world_robot_population_reaches_6_and_half_million

[31.] IR Remote Application Kit dokumentáció, angol nyelvű

http://www.parallax.com/Portals/0/Downloads/docs/prod/compshop/IRremoteAppKit.pdf

[32.] David Buckley weboldala - Ken Gracey lépegető robotjai, angol nyelvű

http://www.davidbuckley.net/DB/inspired/KenGracey/KenGracey.htm

[x.] Saját készítésű fényképek, ábrák és képernyőkép mentések

Felhasznált irodalom

Utolsó ellenőrzés 2010. 10. 20.

DR. HUSI GÉZA oktatási segédanyagai - Debreceni Egyetem Műszaki Kar Villamosmérnöki és Mechatronikai Tanszék

Hivatalos Parallax weboldal, angol nyelvű

http://www.parallax.com

Mobil Robot - Technology of Robotics - angol nyelvű

http://robotechno.us/mobile-robot.html

Penguin Robot Society - *How to program the Parallax Penguin Robot*

http://www.p-robot.com/

BRUNO SICILIANO, OUSSAMA KHATIB (2008) - *Handbook of Robotics* - Springer - ISBN: 978-3-540-23957-4

BRUNO SICILIANO, LORENZO SCIAVICCO - *Robotics: Modelling, Planning and Control* - Springer - ISBN 978-1-84628-641-4

ANDY LINDSAY (2004) - *Robotics with the Boe bot: Student Guide*: Version 2.2 - Parallax Press - ISBN 1-928982-03-4

http://www.parallax.com/dl/docs/books/edu/roboticsv2_2.pdf

BILL WONG (2003) - *Advanced Robotics with the Toddler: Student Guide*: Version 1.2 - Parallax Press - ISBN 1-928982-14-X

http://www.parallax.com/dl/docs/books/edu/toddler.pdf

JEFF MARTIN, JON WILLIAMS (1994 - 2005) - BASIC *Stamp Syntax and Reference Manual* Version 2.2 - Parallax Press - ISBN 1-928982-32-8

http://www.parallax.com/dl/docs/prod/stamps/web-BSM-v2.2.pdf

JON WILLIAMS (2005) - *StampWorks* Version 2.1 - Parallax Press - ISBN 1-928982-35-2

http://www.parallax.com/dl/docs/books/sw/Web-SW-v2.1.pdf

TRACEY ALLEN (1998-2003)- EME Systems - *Tracey Allen's BASIC Stamp app-notes*

http://www.emesystems.com/BS2index.htm

JONATHAN DIXON, OLIVER HENLICH (1997) - *Mobil Robot Navigation*

http://www.doc.ic.ac.uk/~nd/surprise_97/journal/vol4/jmd

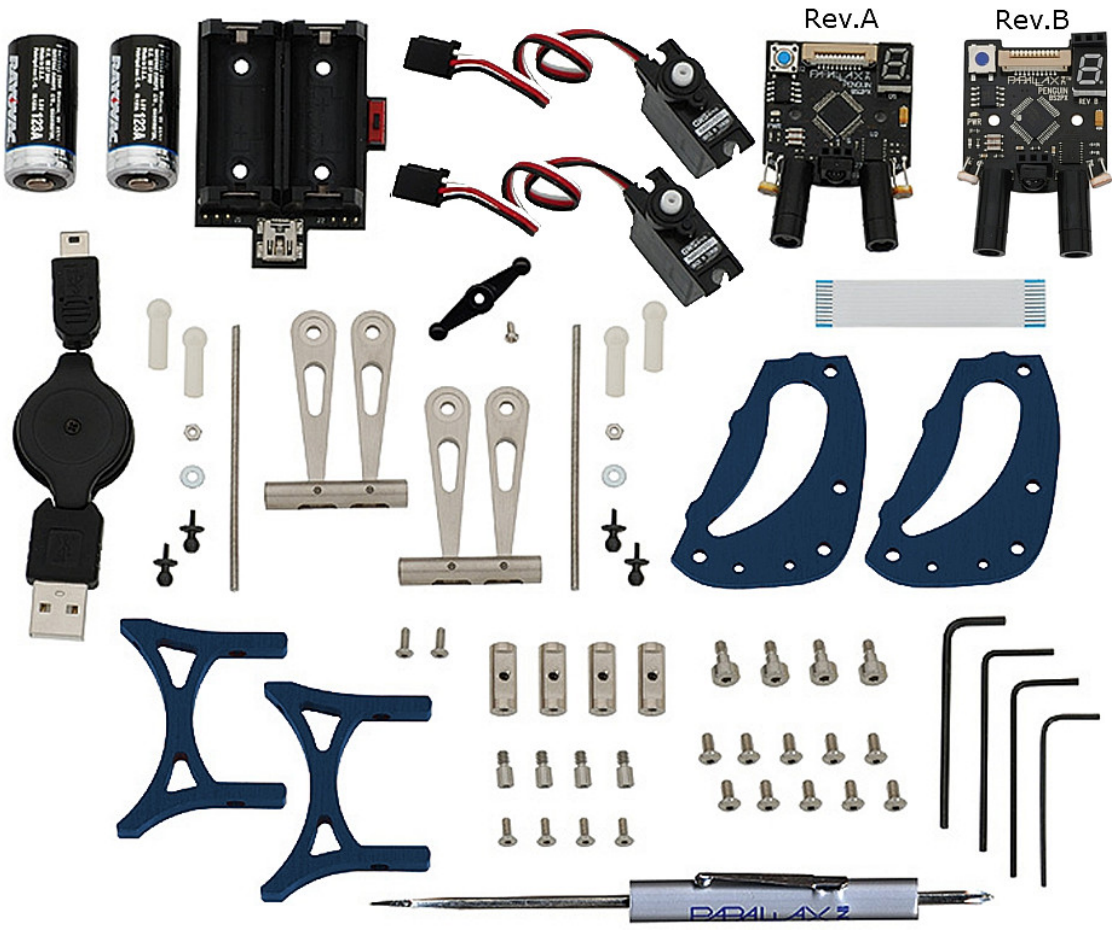
WERNER ÁGNES (2007) - *Robotika - A mesterséges intelligencia alkalmazásának egyik legfontosabb és leglátványosabb területe* (rt.pdf)

http://virt.uni-pannon.hu/index.php/component/docman/doc_download/230-rt

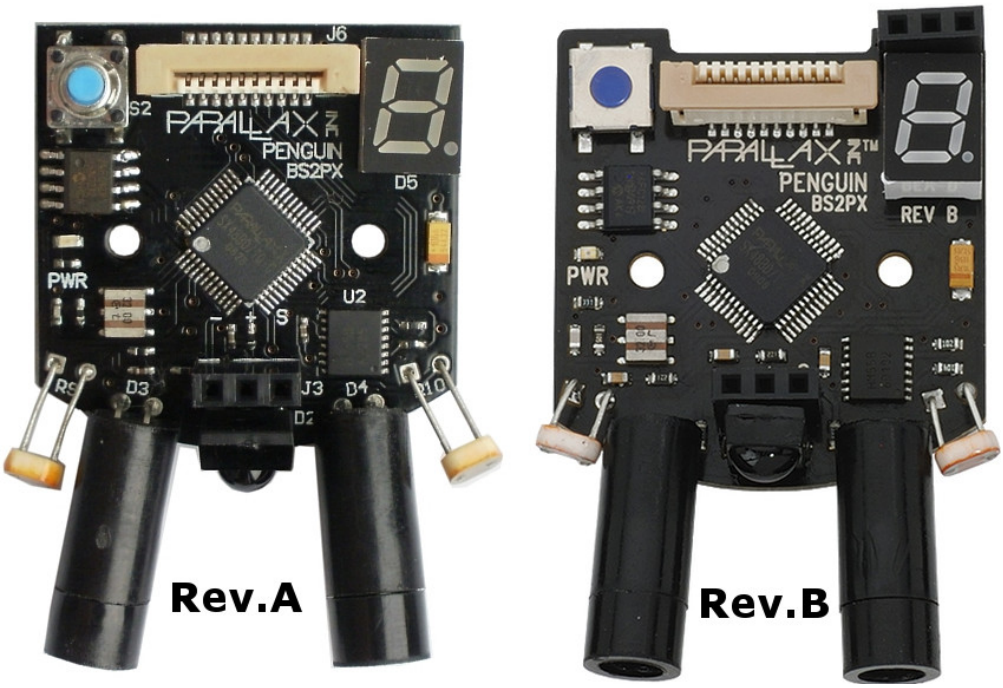
Függelék

```
Battery Threshold
-----------------
Generally, nominal CR123A battery voltages above 5-volts are
acceptible for normal operating. Below 5-volts and the batteries
should be immediately replaced.

Battery Too Low
---------------
If a bad battery set is placed into Penguin, and if the voltage is
very low, around 4.8 volts or less, previously loaded programs will
fail.

-------------------------------------------------------
rc  volts  observations
-------------------------------------------------------
103  6.00  rc samples can vary on different penguins
114  5.00  below 5 volts and replace battery
121  4.60  program errors, fails, erratic behavior
           program won't load properly, display fails
126  4.43  battery fluctuating 4.45, 4.44, 4.43
           program going crazy, recyles the reset
           cannot reload program, no basic stamps found
           led is still on and dim
           turn off and on, still cannot find stamp
130  4.00
     3.90  No useable serial ports found
     3.85  USB device not recognized
     3.49  LED very dim
     3.66  no useable serial ports found, led dim
```

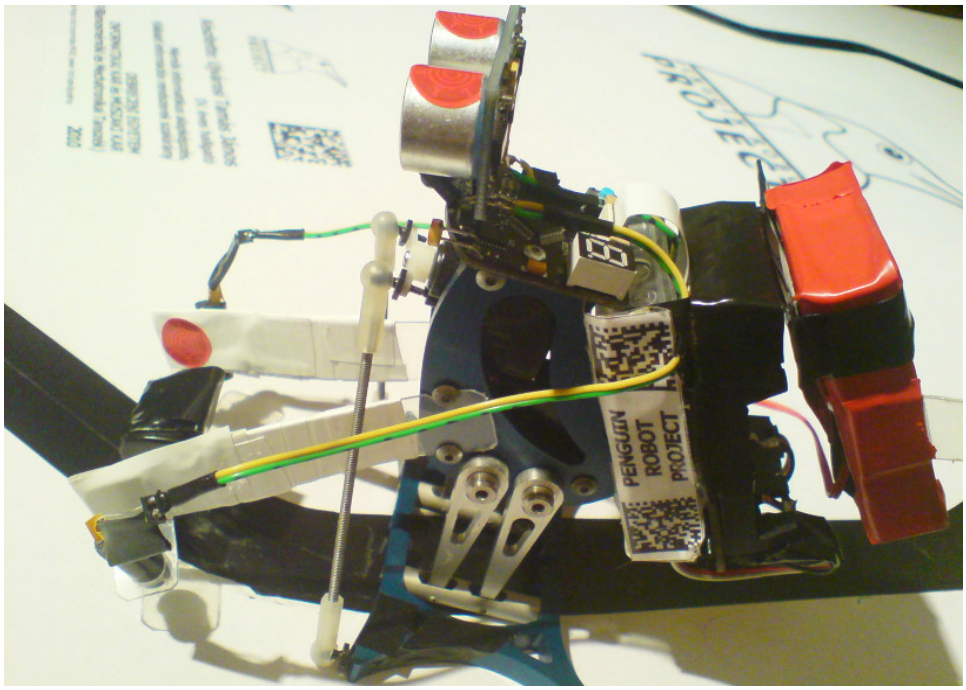
A merülő elemek különböző feszültségszinteken jelentkező hibajelenségei (forrás: Penguin Robot Society)



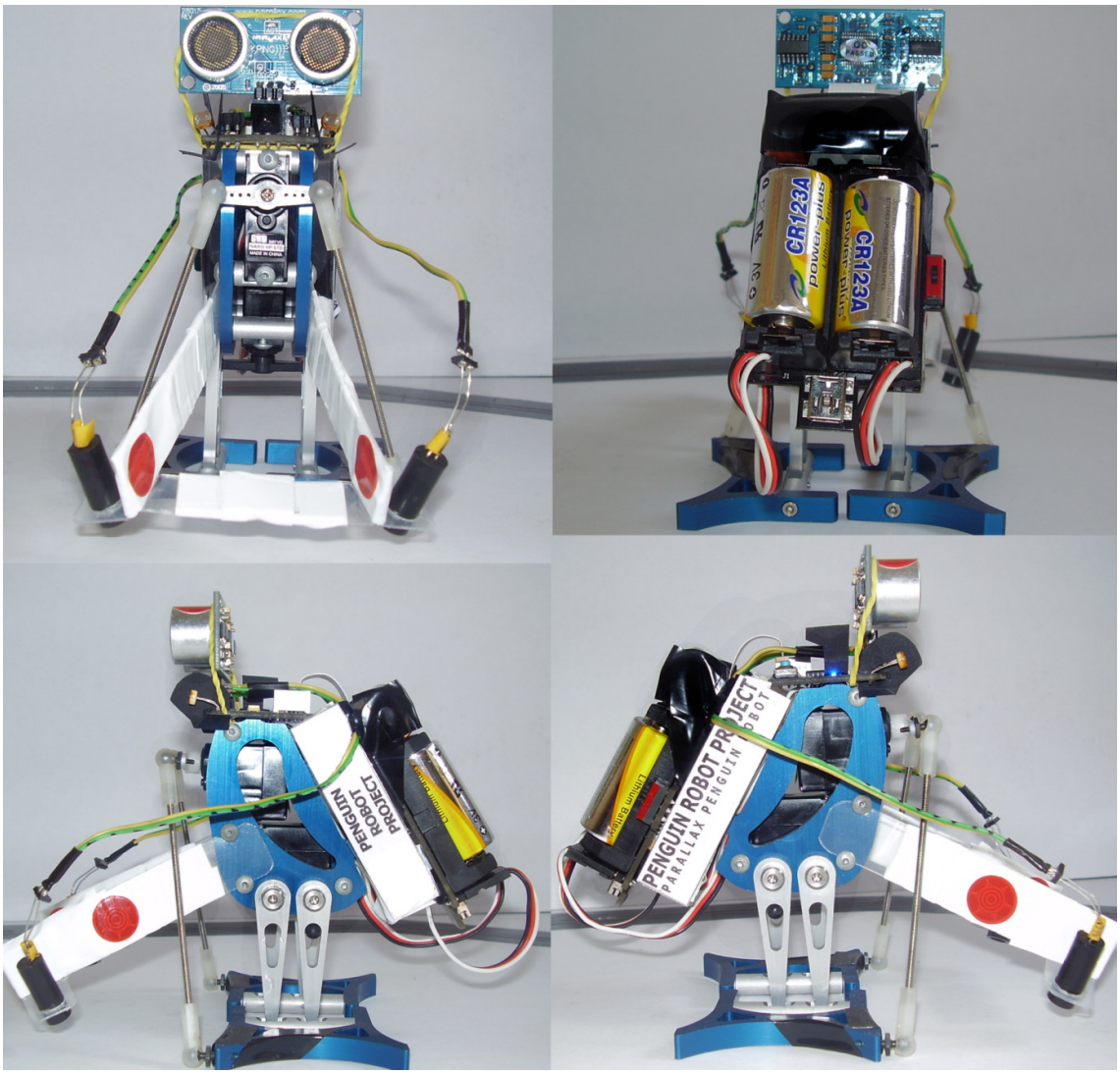
Parallax Penguin robot készlet Rev.A vagy Rev.B kiadású áramkörrel (forrás: www.parallax.com)



A Rev.A és a Rev.B kiadású áramkörök (forrás: www.parallax.com)



Az átalakított Penguin robot a kiegészítő elemtartóval (saját készítésű kép)



Az átalakított Penguin robot különböző nézetekből

A navigációs programokhoz felhasználható távirányítógombok (zöld színnel kiemelve)

Köszönetnyilvánítás

Szeretnék köszönetet mondani témavezetőmnek, Dr. Husi Gézának, a Debreceni Egyetem Műszaki Karának Villamosmérnöki és Mechatronikai Tanszékének tanszékvezető docensének a dolgozat kezdeti feltételeinek megteremtéséért, a dolgozat megírásában nyújtott segítségéért, tanácsaiért és ötleteiért, illetve az általam választott Parallax Penguin robot rendelkezésemre bocsájtásáért. Továbbá, hogy támogatásával bemutathattam szakdolgozatom gyakorlati oldalát a XVI. Épületgépészeti, Gépészeti és Építőipari Szakmai Napok nemzetközi konferencia és kiállítás keretében.

Illetve, óriási köszönet illeti kisfiamat és feleségemet bátorításukért és kitartó támogatásukért.