



دانشکده مهندسی کامپیوتر

کنترل روبات خپرا ۳ از طریق شبکه بی سیم و پیاده سازی عبور روبات از ماز

پایان نامه برای دریافت درجه کارشناسی
در رشته مهندسی کامپیوتر گرایش سخت افزار

زینب ماپار

۸۶۵۲۱۴۱۸

استاد راهنما:

دکتر ناصر مزینی

آبان ماه ۱۳۹۰



دانشکده مهندسی کامپیوتر

کنترل روبات خپرا ۳ از طریق شبکه بی‌سیم و پیاده‌سازی عبور روبات از ماز

پایان نامه برای دریافت درجه کارشناسی
در رشته مهندسی کامپیوتر گرایش سخت افزار

زینب ماپار

۸۶۵۲۱۴۱۸

استاد راهنما:

دکتر ناصر مزینی

آبان ماه ۱۳۹۰

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

تأییدیهی هیأت داوران جلسه‌ی دفاع از پایان‌نامه

نام دانشکده: مهندسی کامپیوتر

نام دانشجو: زینب ماپار

عنوان پایان‌نامه: کنترل روبات خپرا ۳ از طریق شبکه بی‌سیم و پیاده‌سازی عبور روبات از ماز

تاریخ دفاع: ۹۰/۸/۱۱

رشته: مهندسی کامپیوتر

گرایش: سخت افزار

ردیف	سمت	نام و نام خانوادگی	مرتبه	دانشگاه یا مؤسسه	امضا
۱	استاد راهنمای	دکتر ناصر مزینی	استادیار	دانشگاه علم و صنعت	
۲	استاد مدعو داخلی	دکتر سید وحید ازهربی	استادیار	دانشگاه علم و صنعت	

تأییدیهی صحت و اصالت نتایج

باسمہ تعالیٰ

اینجانب زینب ماپار به شماره دانشجویی ۸۶۵۲۱۴۱۸ دانشجوی رشته مهندسی کامپیووتر مقطع تحصیلی کارشناسی تأیید می‌نمایم که کلیهی نتایج این پایان‌نامه حاصل کار اینجانب و بدون هرگونه دخل و تصرف است و موارد نسخه‌برداری شده از آثار دیگران را با ذکر کامل مشخصات منبع ذکر کرده‌ام. درصورت اثبات خلاف مندرجات فوق، به تشخیص دانشگاه مطابق با ضوابط و مقررات حاکم (قانون حمایت از حقوق مؤلفان و مصنفان و قانون ترجمه و تکثیر کتب و نشریات و آثار صوتی، ضوابط و مقررات آموزشی، پژوهشی و انصباطی ...) با اینجانب رفتار خواهد شد و حق هرگونه اعتراض درخصوص احراق حقوق مکتب و تشخیص و تعیین تخلف و محاذات را از خویش سلب می‌نمایم. در ضمن، مسؤولیت هرگونه پاسخگویی به اشخاص اعم از حقیقی و حقوقی و مراجع ذی‌صلاح (اعم از اداری و قضایی) به عهده‌ی اینجانب خواهد بود و دانشگاه هیچ‌گونه مسؤولیتی در این خصوص نخواهد داشت.

نام و نام خانوادگی:

امضا و تاریخ:

مجوز بهره‌برداری از پایان‌نامه

بهره‌برداری از این پایان‌نامه در چهارچوب مقررات کتابخانه و با توجه به محدودیتی که توسط استاد راهنما به شرح زیر تعیین می‌شود، بلامانع است:

- بهره‌برداری از این پایان‌نامه/رساله برای همگان بلامانع است.
- بهره‌برداری از این پایان‌نامه/رساله با اخذ مجوز از استاد راهنما، بلامانع است.
- بهره‌برداری از این پایان‌نامه/رساله تا تاریخ ممنوع است.

نام استاد یا اساتید راهنما:

تاریخ:

امضا:

تقدیم به پدر و مادر عزیزم

به پاس مهر بی بدیل و زحمات بی دریغشان

چکیده

روبات خپرا ۳ یک روبات متحرک چرخ دار است که در بسیاری از دانشگاه های دنیا، از آن برای انجام فعالیت های تحقیقاتی استفاده می شود. اجرای این پروژه بر روی روبات خپرا، در دو مرحله انجام شده است. در مرحله ای اول با بهره گیری از ارتباط بلوتوث و پروتکل ارتباطی مخصوص روبات، دستورات پایه را به روبات ارسال کرده و از این طریق به کنترل توابع روبات پرداخته شده است. سپس کارت wifi بر روی روبات راه اندازی و روبات به شبکه بی سیم متصل شده است. بعد از اتصال روبات به شبکه بی سیم با بهره گیری از پروتکل nfs ، کامپیوتر به سرور nfs و روبات به مشتری nfs تبدیل شده اند.

در مرحله دوم، کد «برنامه عبور روبات خپرا از مازهای ساده» به زبان C نوشته و بر روی روبات پیاده سازی شده است تا روبات بتواند با اجرای آن، به طور مستقل از مازهای ساده عبور کند.

واژه های کلیدی: روبات خپرا ۳، بورد II Korebot II، برنامه نویسی بر روی

فهرست مطالب

۱	فصل ۱: مقدمه
۲	۱-۱- مقدمه
۳	فصل ۲: آشنایی با روبات خپرا ۳
۴	۲-۱- تاریخچه
۴	۲-۲- معرفی روبات خپرا ۳
۶	۲-۲-۱- معرفی بورد Korebot II
۸	۲-۲-۲- سنسورهای مادون قرمز
۹	۲-۲-۲-۱- اندازه گیری نور محیط
۹	۲-۲-۲-۲- اندازه گیری نور بازتاب شده (نزدیکی به اجسام)
۹	۲-۲-۳- سنسورهای اولتراسونیک (ماوراء صوت)
۱۱	۲-۴-۱- موتورها و کنترل آنها
۱۳	۲-۴-۲-۱- حالت سرعت
۱۳	۲-۴-۲-۲- حالت موقعیت
۱۴	۲-۳-۱- پروتکل ارتباطی روبات خپرا
۱۴	۲-۳-۲- پروتکل ارتباطی
۱۵	۲-۳-۳- دستورات پروتکل
۱۶	۱-۲-۳-۲- دستور A, mode
۱۶	۲-۲-۳-۲- دستور B
۱۶	۳-۲-۳-۲- دستور C, array-index (8bits), array-value (16bits)
۱۷	۴-۲-۳-۲- دستور G, US-number
۱۷	۵-۲-۳-۲- دستور J, max-speed (16 bits), acceleration (8 bits)
۱۸	۶-۲-۳-۲- دستور F, target-position-motor-left (32 bits), target-position-motor-right (32 bits)
۱۸	۷-۲-۳-۲- دستور N
۱۸	۸-۲-۳-۲- دستور Z
۱۹	فصل ۳: ارتباطات بی سیم خپرا ۳
۲۰	۱-۳- مقدمه
۲۰	۲-۳- بلوتوث روبات
۲۱	۳-۳- محیط ویندوز

۲۱ Interface Khepera III - ۱-۳-۳
۲۲ RealTerm - ۲-۳-۳
۲۳ ۴-محیط لینوکس ۳
۲۳ ۱-چرا از لینوکس استفاده می کنیم؟ ۳
۲۴ ۲-برقراری ارتباط بلوتوث کامپیوتر و روبات در محیط لینوکس ۳
۲۵ Minicom - ۳-۴-۳
۲۷ ۴-برقراری ارتباط بلوتوث خپرا ۳ به همراه بورد Korebot II ۳
۲۸ ۱-فرستادن فایل به Korebot (آپلود) ۳
۲۸ ۲-فرستادن فایل به کامپیوتر (دانلود) ۳
۲۹ ۵-تکنولوژی wifi ۳
۲۹ ۱-راه اندازی کارت وایرلس ۳
۳۳ ۱-۱-ویرایشگر vi ۳
۳۳ ۲-ارتباط با Korebot با استفاده از ssh و wifi ۳
۳۴ ۳-انتقال فایل با استفاده از scp و wifi ۳
۳۴ ۶-پروتکل nfs و تنظیمات آن ۳
۳۵ ۱-۶-راه اندازی سرور nfs (کامپیوتر) ۳
۳۷ ۲-۶-راه اندازی مشتری (Korebot) nfs ۳

فصل ۴: برنامه نویسی بر روی خپرا ۳

۳۸ ۱-مقدمه ۴
۳۹ ۲-نصب Light toolchain ۴
۳۹ ۱-راهنمای نصب نرم افزار ۴
۴۰ ۲-نصب پوشه‌ی development ۴
۴۰ ۳-نصب (light toolchain) cross-compiler ۴
۴۱ ۴-نصب کتابخانه‌ی بورد libkorebot ۴
۴۲ ۵-نصب اسکریپت برای آپلود کرنل و سیستم فایل: ktboot: ۴
۴۲ ۳-کاربرد Light toolchain ۴
۴۲ ۱-توسعه‌ی نرم افزار ۴
۴۳ ۴-برنامه نویسی برای روبات خپرا ۳ ۴
۴۳ ۵-طرح مسئله ۴
۴۴ ۶-برنامه‌ی عبور خپرا از مازهای ساده ۴
۴۴ ۱-توضیح برنامه ۴

۴-۶-۲- کد برنامه ۴۵

۵۹	فصل ۵: جمع‌بندی و پیشنهادها
۶۰	۱- مقدمه ۵
۶۱	۲- جمع‌بندی ۵
۶۱	۳- پیشنهادها ۵

۶۲ مراجع

فهرست اشکال

..... شکل (۱-۲) روبات خپرا ۳	۵
..... شکل (۲-۲) نمای کلی سخت افزار Korebot از پایین	۶
..... شکل (۳-۲) نمای کلی سخت افزار Korebot از بالا.....	۷
..... شکل (۴-۲) سنسورهای مادون قرمز از نمای پایین.....	۸
..... شکل (۵-۲) فرستنده و گیرنده امواج ماوراء صوت	۱۰
..... شکل (۶-۲) نحوه شماره گذاری سنسورهای اولتراسونیک.....	۱۰
..... شکل (۷-۲) تغییر مقدار زمان روشن بودن به کل زمان تناوب، توان موتور را تعیین می کند.....	۱۲
..... شکل (۸-۲) منحنی سرعت بر حسب زمان و منحنی مکان بر حسب زمان برای روبات در حالت موقعیت ..	۱۴
..... شکل (۱-۳) برنامه‌ی Interface Khepera III	۲۲
..... شکل (۲-۳) RealTerm یک ترمینال سریال برای ارتباط با پورت‌های COM است.....	۲۳
..... شکل (۳-۳) برنامه blueman	۲۵
..... شکل (۴-۳) ارتباط بلوتوث در لینوکس مانند یک پورت سریال در نظر گرفته می شود.....	۲۵
..... شکل (۵-۳) برنامه minicom برای ارتباطات سریال در لینوکس	۲۶
..... شکل (۶-۳) ورود به سیستم عامل بورد Korebot	۲۷
..... شکل (۷-۳) کارت wifi بطور صحیح راه اندازی شده است.....	۳۱
..... شکل (۸-۳) برای دیدن IP کارت از دستور ifconfig -a استفاده می شود.....	۳۲
..... شکل (۹-۳) برای چک کردن اتصال به شبکه از دستور ping استفاده می شود.....	۳۲
..... شکل (۱۰-۳) انتقال فایل khepera3_test از کامپیوتر به بورد Korebot	۳۴
..... شکل (۱۱-۳) با دستور rpcinfo باید یک لیست از برنامه های در حال اجرا بینید.....	۳۶

فهرست جداول

٥	جدول (١-٢) مشخصات روبات خپرا
٧	جدول (٢-٢) مشخصات بورد Korebot

فصل ۱:

مقدمه

۱-۱- مقدمه

در عصر حاضر تلاش گسترده‌ای جهت خودکار نمودن فعالیت‌های انسانی و انجام آن توسط ربات‌های هوشمند در حال انجام است. از نمونه این کاربردها می‌توان استفاده از ربات‌ها در ایستگاه‌های فضایی برای پیمایش کرات، در صنعت برای جابجایی اجسام، در کشاورزی برای حمل محصولات کشاورزی و عملیات باگبانی، در میدان‌های جنگی برای پیدا کردن میین، در سوانحی چون آتش سوزی و زلزله برای امدادرسانی، در کمک به معلولین و غیره نام برد.

هدف اصلی محققان در زمینه ربات‌های متحرک، طراحی ربات‌هایی است که بتوانند در محیط‌های ناشناخته به طور خودکار و بدون نیاز به کنترل انسانی با توجه به اطلاعات حاصل از حسگرهای بدون برخورد به موانع حرکت نمایند. این زمینه تحقیقاتی ناوبری ربات نام گرفته است(۱).

فصل ۲:

آشنایی با روبات خپرا ۳

۱-۲- تاریخچه

خپرا^۱ یک روبات متحرک^۲ است که اولین نمونه‌ی آن در سال ۱۹۹۱ در آزمایشگاه LAMI^۳ دانشگاه EPFL^۴ سوئیس و تحت نظرارت پروفسور نیکود^۵ ساخته شد (۲). ابعاد کوچک روبات و قابلیت‌های متعدد آن باعث شد که این روبات در دانشگاه‌های مختلف دنیا برای انجام فعالیت‌های تحقیقاتی مورد استفاده قرار بگیرد.

با پیشرفت تکنولوژی و به وجود آمدن نسل جدید میکروکنترلرهای K-team^۶ سری‌های جدید روبات خپرا را عرضه کرد. روبات خپرا^۷ ۳ جدیدترین مدل روبات‌های خپرا است که در این پژوهه مورد استفاده قرار گرفته است.

۲-۲- معرفی روبات خپرا ۳

خپرا ۳ یک روبات متحرک چرخ دار استوانه‌ای به قطر ۱۳۰ میلی متر و ارتفاع ۷۰ میلی متر است (شکل ۲-۱). این روبات دارای پردازنده‌ی DsPIC30F5011 با فرکانس ۶۰ مگا هرتز، تعداد نه سنسور مادون قرمز^۸ و همچنین تعداد پنج سنسور اولتراسونیک^۹ برای تشخیص موانع می‌باشد. تعداد دو سنسور مادون قرمز دیگر برای دنبال کردن خط^{۱۰} در زیر روبات قرار دارند. ویژگی‌های دیگر این روبات در جدول ۱-۲ آمده‌اند (۳).

¹ Khepera

² Mobile robot

³ Laboratoire de Micro-Informatique

⁴ Ecole Polytechnique Federale de Lausanne

⁵ Prof. Jean-Daniel Nicoud

⁶ Khepera III

⁷ Infrared

⁸ Ultrasonic

⁹ Line following

کنترل روبات خپرا ۳ از طریق شبکه بی سیم و پیاده سازی عبور روبات از ماز آشنایی با روبات خپرا ۳



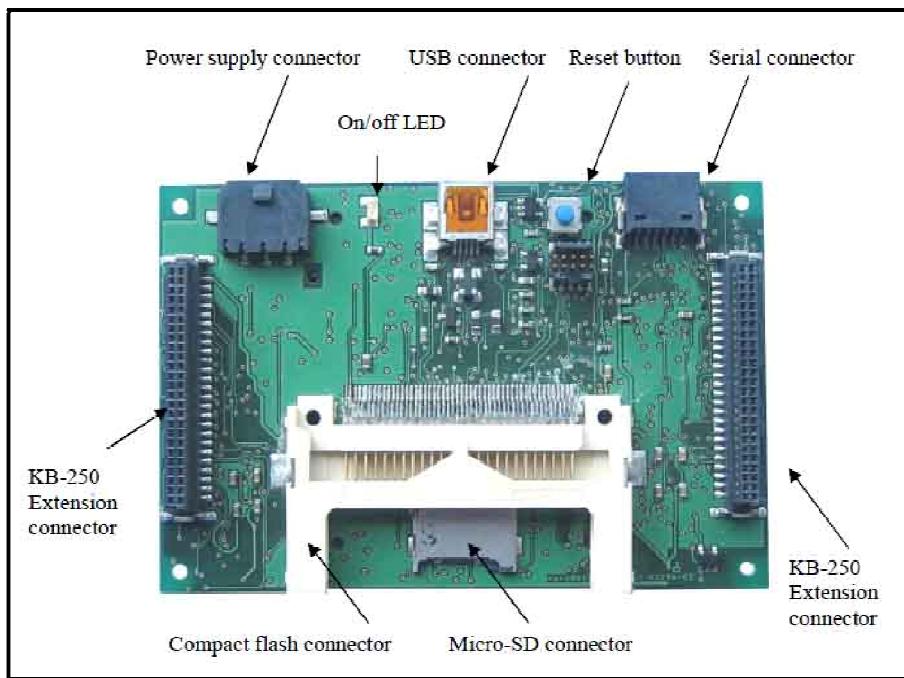
شکل (۱-۲) روبات خپرا ۳

جدول (۱-۲) مشخصات روبات خپرا ۳

پردازنده	DsPIC 30F5011 با فرکانس ۶۰ MHz
حافظه RAM	۴KB روی بورد Korebot ، DsPIC ۱۲۸MB
حافظه Flash	۳۲KB روی بورد Korebot ، DsPIC ۶۶KB
حرکت	۲ موتور DC با انگوشهای افزایشی (~ ۲۲ پالس در هر میلی متر از حرکت روبات)
سرعت	حداکثر: ۰/۵ m/s
سنسورها	۱۱ سنسور مادون قرمز و ۵ سنسور اولتراسونیک با بازه تشخیص ۲۰ cm تا ۱۶۰ cm
توان	بسطه باتری با پاور آدپتور لیتیوم پلیمر (۱۴۰۰ m Ah)
ارتباط	پورت سریال استاندارد، ارتباط USB تا سرعت ۱۱۵kbps ، ارتباط بی سیم با کارت WiFi و Korebot
ابعاد	قطر: ۷۰ mm ارتفاع: ۱۳۰ mm
وزن	۶۹۰ g

۱-۲-۲ - معرفی بورد Korebot II

ویژگی منحصر به فرد روبات خپرا، امکان ارتقای قابلیت‌های آن با استفاده از بورد Korebot می‌باشد. این بورد به واسطه‌ی گذرگاه^۱ KB-250 بر روی روبات نصب می‌شود. بورد Korebot دارای پردازنده PXA270 با فرکانس ۶۰۰ مگا هرتز و حافظه‌ی اصلی^۲ ۱۲۸ مگابایت است. به علاوه یک سیستم عامل لینوکس بر روی آن تعییه شده است. در شکل (۲-۲) و شکل (۳-۲) و ۳-۲ نمای کلی از سخت افزار Korebot نشان داده شده است و در جدول ۲-۲ مشخصات کلی بورد آمده است(4).

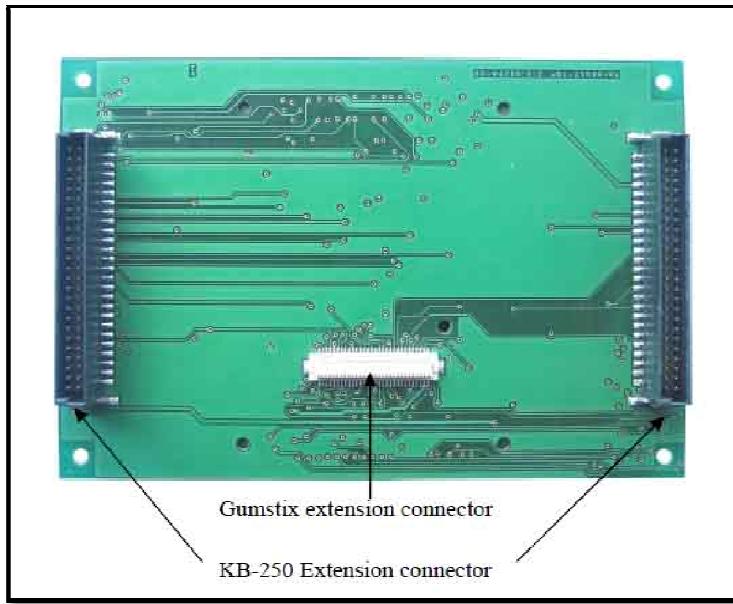


شکل (۲-۲) نمای کلی سخت افزار Korebot از پایین

¹ Bus

² RAM

کنترل روبات خپرا ۳ از طریق شبکه بی سیم و پیاده سازی عبور روبات از ماز آشنایی با روبات خپرا ۳



شکل (۳-۲) نمای کلی سخت افزار Korebot از بالا

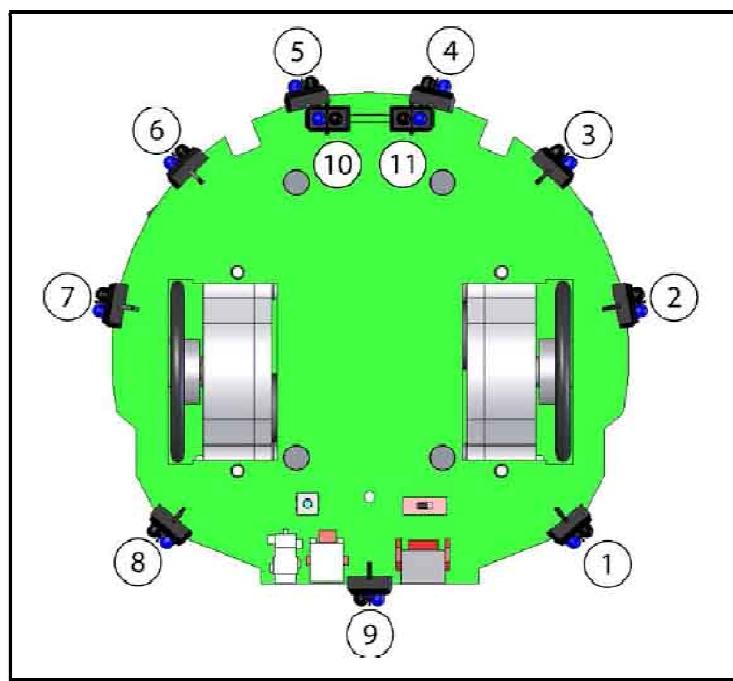
جدول (۲-۲) مشخصات بورد Korebot

پردازنده	پردازنده Marvell PXA270 به همراه XScale با فرکانس ۶۰۰ MHz
حافظة RAM	۱۲۸MB
حافظة Flash	۳۲MB
ویژگی ها	USB host signals Compact Flash connector Micro SD connector Mini USB connector (برای دستگاه ها، دوربین، USB key و غیره) سازگار با گذرگاه توسعه (I2C) Korebot پورت سریال
سیستم عامل	Angstrom distribution (OpenEmbedded tools)، سیستم عامل لینوکس، kernel 2.6.24
صرف توان	۵V و ۱۰۰mA (بدون هیچ گونه کارت توسعه)
ابعاد	عرض: ۳۳mm ارتفاع: ۵۷mm عمق: ۸۵mm
وزن	۵۲g

کنترل روبات خپرا ۳ از طریق شبکه بی سیم و پیاده سازی عبور روبات از ماز آشنایی با روبات خپرا ۳

۲-۲-۲- سنسورهای مادون قرمز

خپرا، تعداد نه سنسور مادون قرمز در اطراف بدنه و دو سنسور در زیر دارد که مانند شکل ۴-۲ شماره گذاری می‌شوند. هر سنسور شامل یک پخش کننده^۱ و یک دریافت کننده‌ی نور مادون قرمز است که با استفاده از آن‌ها می‌توان دو مقدار «نور محیط» و «تزوییکی به اجسام» را اندازه‌گیری کرد.



شکل (۴-۲) سنسورهای مادون قرمز از نمای پایین

¹ Emitter

۱-۲-۲-۱- اندازه گیری نور محیط^۱

این مقدار فقط با استفاده از قسمت دریافت کننده سنسور اندازه گیری می‌شود. خروجی سنسور در واقع میزان نور مادون قرمز محیط را تعیین می‌کند و به شدت تحت تأثیر محیط اطراف روبات است. بسته به نوع، رنگ و فاصله منبع نور، مقادیر سنسورها تغییر می‌کنند. مقادیر سنسورها به طور متوالی و با فاصله زمانی 3 میلی ثانیه خوانده می‌شوند، در نتیجه یک سنسور هر 33 میلی ثانیه یک بار خوانده می‌شود.^(۵)

۱-۲-۲-۲-۲- اندازه گیری نور بازتاب شده^۲ (نزدیکی به اجسام^۳)

زمانی که قسمت پخش کننده سنسور، نور مادون قرمز را در محیط پخش می‌کند، قسمت دریافت کننده میزان نور را می‌سنجد. اما مقداری که روبات بر می‌گرداند در واقع اختلاف بین اندازه گیری با استفاده از قسمت پخش کننده سنسور و اندازه گیری بدون آن (اندازه گیری نور محیط) است. در این روش، سنسورها برای تشخیص اجسام اطراف روبات (با فواصل کم) به کار می‌روند. خروجی سنسورها به رنگ، جنس و میزان صیقلی بودن سطح اجسام و همچنین شرایط نور محیط بستگی دارد. به همین دلیل ممکن است روبات در تشخیص نزدیکی به اجسام دچار اشتباه شود. اما به طور کلی، هرچه جسم به روبات نزدیک‌تر باشد، نور بیشتری را منعکس می‌کند، در نتیجه خروجی سنسورها بزرگ‌تر است.

در این بخش نیز مانند بخش قبل، تمام سنسورها در طول 33 میلی ثانیه خوانده می‌شوند.^(۵)

۱-۲-۳- سنسورهای اولتراسونیک (ماوراء صوت)

در اطراف بدنی روبات خپرا، تعداد پنج سنسور اولتراسونیک وجود دارد که هر سنسور از یک

^۱ Ambient

^۲ Reflected

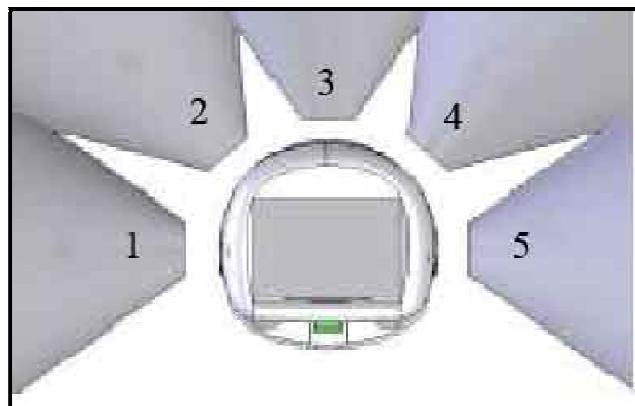
^۳ Proximity

کنترل روبات خپرا ۳ از طریق شبکه بی سیم و پیاده سازی عبور روبات از ماز آشنایی با روبات خپرا ۳

فرستنده^۱ و یک گیرنده امواج ماوراء صوت تشکیل شده است(شکل ۵-۲). فرکانس اسمی این سنسورها ۴۰ کیلو هرتز با تولورانس ۱ کیلو هرتز است. نحوه شماره گذاری و میدان دید این سنسورها در شکل ۶-۲ نشان داده شده است.



شکل (۵-۲) فرستنده و گیرنده امواج ماوراء صوت



شکل (۶-۲) نحوه شماره گذاری سنسورهای اولتراسونیک

پارامترهای مربوط به سنسورهای اولتراسونیک مانند حداکثر تعداد اکو^۲، تعداد سنسورهای فعال و غیره را می‌توان از طریق دستورات مخصوص خپرا تنظیم کرد. به طور پیش فرض، حداکثر تعداد اکوها مقدار ۳ می‌باشد و فقط سنسور جلویی (شماره ۳) فعال است. این سنسورها اجسام در فاصله‌ی ۲۰ سانتی متری تا حدود ۱۶۰ سانتی متری را تشخیص می‌دهند و برای اجسام خارج از

¹ Transmitter

² Echo

این محدوده مقدار صفر را برابر می‌گرداند.

آخرین پارامتر نشان می‌دهد که آیا بخش بالایی بدن‌هی خپرا بر روی آن قرار دارد یا خیر. وقتی که بدن‌هی بالایی خپرا نصب باشد، انعکاس امواج معاوِر^۱ صوت در درون روبات، روی مقدار خروجی سنسورها تأثیر می‌گذارد. بنابراین برای تشخیص دقیق‌تر موانع نزدیک (۲۰ تا ۴۰ سانتی متر) بهتر است بدن‌هی بالایی روبات را برداریم^(۵).

۴-۲-۲- موتورها و کنترل آن‌ها

робات خپرا ۳ دارای دو موتور DC است که با میزان کاهش $1 : 43/2$ به چرخ‌ها وصل شده‌اند. یعنی به ازای $43/2$ دور چرخش موتور، چرخ روبات یک دور می‌چرخد. از طرفی یک انگدر افزایشی^۱ روی محور^۲ هر موتور قرار دارد که به ازای هر بار چرخش موتور، ۱۶ پالس تولید می‌کند. بنابراین به ازای هر چرخش یک چرخ، انگدر $= 43/2 \times 16 = 691/2$ پالس تولید می‌کند. در حالت پیش فرض، تفکیک پذیری^۳ انگدر روبات در حالت چهار برابر است، یعنی هر پالس را به چهار واحد^۴ تقسیم می‌کند. بنابراین، یک چرخش چرخ معادل 2764 واحد است. از طرف دیگر، قطر یک چرخ 41 میلی متر است، در نتیجه در یک دور چرخش چرخ، روبات $128/8$ میلی متر را طی می‌کند.

$$1 \text{ دور چرخش چرخ} = 128/8 \text{ میلی متر} = 2764 \text{ واحد}$$

هر موتور با یک کنترل گر هدایت می‌شود که در یک ریزپردازنده^۵ PIC18F4431 پیاده سازی شده است. ریزپردازندهی PIC روی توان^۶ موتور کنترل مستقیم دارد و می‌تواند پالس‌های افزایش انگدر افزایشی را بخواند. قسمت کنترل کنندهی موتور مانند یک بردهی^۷ ارتباط i2c عمل می‌کند. وقتی روبات خپرا Korebot^۸ روی روبات نصب است، پردازندهی Korebot و در غیر این صورت پردازندهی اصلی خپرا،

¹ Incremental encoder

² Axis

³ Resolution

⁴ Measure

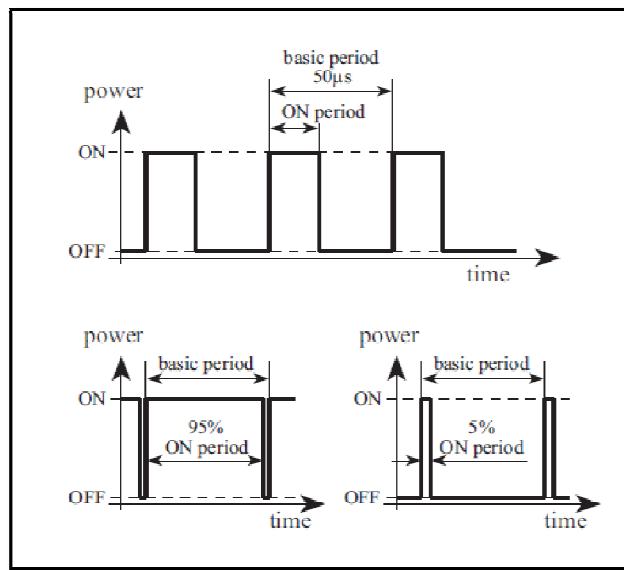
⁵ Power

⁶ Slave

کنترل روبات خپرا ۳ از طریق شبکه بی سیم و پیاده سازی عبور روبات از ماز آشنایی با روبات خپرا ۳

ارباب^۱ i2c می‌شود.

هر کنترل کننده موتور، با یک فرکانس خاص موتور را روشن و خاموش می‌کند تا مقدار توان مطلوبی را برای موتور فراهم کند. به این روش کنترل توان، مدولاسیون عرض پالس^۲ یا PWM می‌گویند. در روش PWM نسبت زمان روشن بودن به زمان دوره تناوب، توان موتور را معین می‌کند. هر چه زمان روشن بودن بیشتر باشد، توان موتور هم بیشتر است^(۶). همان طور که در شکل ۷-۲ مشاهده می‌شود، فرکانس سوئیچینگ پایه ثابت است و تغییر مقدار زمان روشن بودن به کل زمان تناوب، توان موتور را تعیین می‌کند.



شکل (۷-۲) تغییر مقدار زمان روشن بودن به کل زمان تناوب، توان موتور را تعیین می‌کند

هر کنترل کننده موتور با تنظیم مقدار صحیح PWM می‌تواند سرعت یا مکان موتور را کنترل کند.

هر دو موتور DC می‌توانند با یک کنترل کننده PID کنترل شوند. کنترل کننده های PID با سه مقدار ثابت K_p برای قسمت تنااسبی^۳ K_i برای قسمت انگرال گیر و K_d برای قسمت مشتق گیر

¹ Master

² Pulse Width Modulation

³ Proportional

کنترل روبات خپرا ۳ از طریق شبکه بی سیم و پیاده سازی عبور روبات از ماز آشنایی با روبات خپرا ۳

معین می‌شوند. کنترل کننده PID می‌تواند در دو حالت کنترلی استفاده شود: حالت سرعت یا حالت موقعیت. حالت کنترلی فعال، بر اساس دستور رسیده مشخص می‌شود. مثلاً اگر کنترل کننده، یک دستور کنترل سرعت دریافت کند، به حالت سرعت سوئیچ می‌کند. پارامترهای کنترلی K_p , K_i و K_d می‌توانند برای هر دو حالت تنظیم شوند(5).

۲-۲-۱-۴- حالت سرعت

در حالت سرعت، کنترل کننده مقدار سرعت موتور را از ورودی می‌گیرد، مقدار واقعی سرعت بر حسب میلی متر بر ثانیه را محاسبه کرده و موتورها را برای رسیدن به این سرعت تنظیم می‌کند. رابطه بین سرعت ورودی^۱ و سرعت واقعی^۲ در رابطه ۱-۲ نشان داده شده است.

(۱-۲)

$$Real\ Speed = \frac{Motor\ Speed}{K_{Speed}} \quad [\frac{mm}{s}]$$

اگر از تنظیمات پیش فرض استفاده شود، K_{Speed} برابر با $144/0.1$ خواهد بود. در حالت رگلاسیون، حداقل سرعت موتور 43000 (برابر با سرعت واقعی 298 میلی متر بر ثانیه) و حداقل سرعت آن 2000 (برابر با سرعت واقعی $13/9$ میلی متر بر ثانیه) است(5).

۲-۲-۲-۴- حالت موقعیت

در حالت موقعیت، کنترل کننده مقدار شتاب و حداقل سرعت از ورودی (با دستور J) خوانده، سپس با استفاده از این مقادیر، به چرخ‌ها شتاب می‌دهد تا به حداقل سرعت دست یابند و دوباره از سرعت چرخ‌ها می‌کاهد تا روبات به موقعیت هدف برسد. منحنی سرعت بر حسب زمان برای

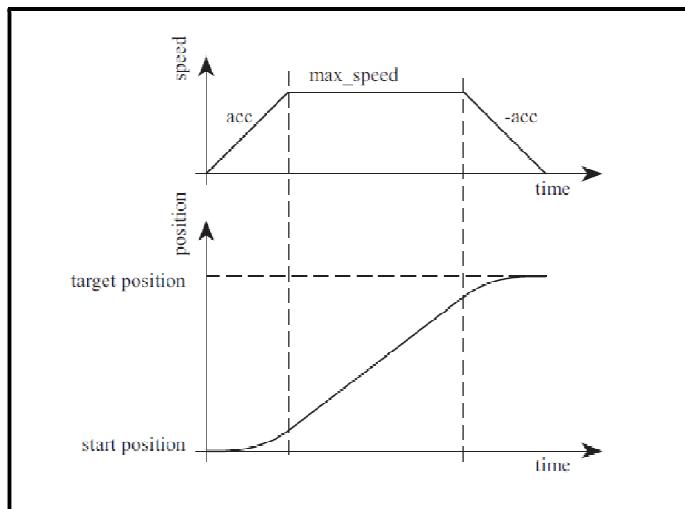
¹ Motor Speed

² Real Speed

کنترل روبات خپرا ۳ از طریق شبکه بی سیم و پیاده سازی عبور روبات از ماز آشنایی با روبات خپرا ۳

روبات در این حالت در نمودار شکل ۸-۲ نشان داده شده است.

کنترل کننده همچنین مکان هدف^۱ هر چرخ را بر حسب پالس دریافت می‌کند (با دستور F) و طبق منحنی‌های نشان داده شده در شکل ۸-۲ حرکت می‌کند تا روبات به موقعیت هدف برسد. با توجه به اینکه هر پالس معادل 0.47×10^{-6} میلی متر است، می‌توان تعداد پالس‌های لازم برای پیمودن فاصله‌ی مورد نظر را به دست آورد(5).



شکل (۸-۲) منحنی سرعت بر حسب زمان و منحنی مکان بر حسب زمان روبات در حالت موقعیت

۲-۳-۲- پروتکل ارتباطی روبات خپرا

۲-۳-۱- پروتکل ارتباطی

پروتکل ارتباطی بین روبات و کامپیوتر از کدهای ASCII استاندارد استفاده می‌کند و از طریق یک

¹ Target position

کنترل روبات خپرا ۳ از طریق شبکه بی سیم و پیاده سازی عبور روبات از ماز آشنایی با روبات خپرا ۴

پورت سریال، به کنترل کامل توابع روبات می‌پردازد. وسایل دارای بلوتوث می‌توانند با استفاده از پروتکل RFCOMM پورت‌های سریال را شبیه سازی کنند (7). بنابراین می‌توان از ارتباط بلوتوث به عنوان خط سریال بین روبات و کامپیوتر استفاده کرد.

بعد از برقراری ارتباط بلوتوث (طبق بخش ۲-۳) می‌توان طبق قواعد این پروتکل، دستورات را از کامپیوتر به روبات ارسال کرد و پاسخ‌های بازگشته از روبات را نیز مشاهده کرد. هر دستور با یک حرف بزرگ الفبا شروع می‌شود و در صورت نیاز، با پارامترهای عددی یا حرفی که با ویرگول از هم جدا شده‌اند، ادامه می‌یابد. در نهایت با یک line feed دستور خاتمه می‌یابد. هر پاسخ با همان حرف آغازین دستور متناظرش شروع می‌شود، با این تفاوت که حرف کوچک است. در صورت لزوم، پاسخ با پارامترهای عددی یا حرفی که با ویرگول از هم جدا شده‌اند، ادامه یافته و در انتها به یک line feed ختم می‌شود.

هنگامی که یک دستور، پارامترهای عددی بزرگ‌تر از هشت بیت (بزرگ‌تر از ۲۵۵) دارد، باید از پیشوندهای زیر استفاده کرد.

- d: برای مقادیر ۱۶ بیتی

- l: برای مقادیر ۳۲ بیتی

- f: برای مقادیر ممیز دار^۱

مثالاً برای تنظیم سرعت از دستور D, l20000, l20000 استفاده می‌شود و روبات پس از اجرای دستور، پاسخ d را به کامپیوتر ارسال می‌کند (5).

۲-۳-۲ - دستورات پروتکل

دستورات پروتکل ارتباطی به همراه نحوهٔ پارامتر دهی آن‌ها در راهنمای روبات خپرا آمده است. در اینجا به معرفی و نحوهٔ استفاده‌ی چند دستور می‌پردازیم.

¹ Float

کنترل روبات خپرا ۳ از طریق شبکه بی سیم و پیاده سازی عبور روبات از ماز آشنایی با روبات خپرا ۳

۱-۲-۳-۲ - دستور A, mode

این دستور روبات را به حالت برایتنبرگ^۱ می‌برد. در این حالت روبات با استفاده از سنسورهای مادون قرمز (mode=Q) یا سنسورهای اولتراسونیک (mode=I) از اجسام دوری می‌کند. با دستور A, 2 حالت برایتنبرگ متوقف می‌شود.

۲-۳-۲-۲ - دستور B

روبات در جواب به این دستور، نسخه‌ی بایاس‌اش را به کامپیوتر می‌فرستد.

۳-۲-۳-۲ - دستور C, array-index (8bits), array-value (16bits)

با استفاده از دستور C می‌توان تعداد سنسورهای مادون قرمز و اولتراسونیک فعال و غیره (داخل یک متن، به معنی «سایر موارد» یا «غیره» هرگز سه نقطه استفاده نمی‌کنند) را برای سیستم تنظیم کرد. دستور C, 0, d31 تمام سنسورهای اولتراسونیک روبات را فعال می‌سازد. در حالت پیش فرض فقط سنسور جلویی (сенسور شماره ۳) فعال است.

دستور C, 1, d3 یعنی حداکثر تعداد اکوها برابر ۳ باشد. با توجه به اینکه که گاهی چندین شیء در راستای امواج یک سنسور اولتراسونیک قرار می‌گیرند، روبات باید بداند چند اکو را در نظر بگیرد، یعنی فاصله‌ی چند شیء نزدیک به روبات را باید محاسبه کند.

دستور C, 6, d1 یعنی بدنه‌ی بالایی خپرا روی روبات قرار دارد. هنگامی که بدنه‌ی بالایی روی روبات نصب است، به دلیل انعکاس امواج درون بدنه‌ی روبات، نویز ایجاد می‌شود. برای اندازه گیری دقیق‌تر فواصل ۴۰ تا ۲۰ سانتی متر، باید بدنه‌ی بالایی خپرا را برداشته و با دستور C, 6, d0 این حالت را به خپرا اعلام کرد.

¹ Braitenberg

کنترل روبات خپرا ۳ از طریق شبکه بی سیم و پیاده سازی عبور روبات از ماز آشنایی با روبات خپرا ۳

۴-۲-۳-۲- دستور G, US-number

به طور کلی هر دستور G، تعداد اکوها بازگشتی، فاصله‌ی هر اکو بر حسب سانتی متر، دامنه‌ی هر کدام و زمان نسبی دیده شدن هر اکو را برمی‌گرداند.

دستور 1 G، مقادیر مربوط به سنسور اولتراسونیک شماره‌ی ۱ را از روبات می‌خواند.

برای خواندن مقادیر سنسور اول باید از قبل با دستور C, 0, value آن را فعال کرد.

اگر قبلًا با دستور C, 1, d3 حداکثر تعداد اکوها، برابر با ۳ تنظیم شده باشد، ولی تنها ۲ شی، در مقابل سنسور باشند، روبات در پاسخ تعداد اکوها را برابر با ۲ می‌دهد (یعنی فقط ۲ شی را در مقابل سنسور تشخیص داده است) و فاصله‌ی روبات تا اشیا را هم مشخص می‌کند. Dist1 فاصله‌ی روبات تا نزدیک‌ترین شی، و Dist2 فاصله‌ی روبات تا دومین شی، نزدیک، بر حسب سانتی متر است.

توجه: اگر روبات تعداد اکوها را صفر برمی‌گرداند، دو حالت زیر محتمل است:

- فاصله‌ی مانع تا روبات کمتر از ۱۵ سانتی متر یا بیشتر از ۱۶۰ سانتی متر بوده است. به

طور کلی، محدوده‌ی قابل اندازه گیری با سنسورهای اولتراسونیک روبات خپرا از ۱۵ سانتی متر تا حدود ۱۶۰ سانتی متر است. حد بالای این سنسورها را به حدود ۱۶۰ سانتی متر محدود کرده‌اند تا زمان خواندن سنسورها کمتر شود.

- مانع خیلی کوچک است و سنسور مانعی را در مقابلش نمی‌بیند. لذا بهتر است از مانع‌های بزرگ استفاده شود تا اندازه گیری‌ها معتبر باشند.

آخرین مقداری که دستور G برمی‌گرداند، نویز سفید^۱ است که قبل از ارسال امواج اولتراسونیک اندازه گیری می‌شود. بنابراین دامنه‌ی واقعی هر اکو به اندازه‌ی تفاضل دامنه و نویز سفید است.

۴-۲-۳-۲-۵- دستور J, max-speed (16 bits), acceleration (8 bits)

دستور J حداکثر سرعت و شتاب موتور در حالت موقعیت را تعیین می‌کند (جهت یادآوری به بخش ۲-۲-۴-۲ مراجعه شود). در حالت پیش فرض حداکثر سرعت ۲۰۰۰۰ و مقدار شتاب ۶۴

^۱ White noise

کنترل روبات خپرا ۳ از طریق شبکه بی سیم و پیاده سازی عبور روبات از ماز آشنایی با روبات خپرا ۳

است.

۶-۲-۳-۲ - دستور F, target-position-motor-left (32 bits), target-position-motor-right (32 bits)

موقعیت هدف را برای چرخها و مشخص می‌کند. در این حالت جابجایی در سه مرحله انجام می‌شود: مرحله‌ی شتاب برای رسیدن به حداکثر سرعت (در صورت امکان)، مرحله‌ی حرکت با سرعت ثابت و مرحله‌ی کاهش سرعت قبل از رسیدن موقعیت هدف. شتاب و حداکثر سرعت را می‌توان از قبل با دستور J مشخص کرد. پارامترهای دستور J بر حسب پالس هستند (هر پالس معادل ۰/۰۴۷ میلی متر است).

۷-۲-۳-۲ - دستور N

دستور N مقادیر ۱۱ سنسور مادون قرمز، در حالت اندازه گیری نزدیکی اجسام را برمی‌گرداند و آخرین پارامتر آن زمان نسبی اندازه گیری است.

۸-۲-۳-۲ - دستور Z

شمارنده‌ی زمان نسبی را صفر می‌کند.

فصل ۳:

ارتباطات بی سیم خپرا ۳

۱-۳- مقدمه

این فصل به چگونگی کنترل روبات و انجام تنظیمات آن، از طریق بلوتوث و همچنین شبکه‌ی بی‌سیم می‌پردازد

۲-۳- بلوتوث روبات

روبات خپرا ۳ به دستگاه بلوتوث مدل WT12 از شرکت بلوگیگا^۱ مجهز است. ارتباط بلوتوث می‌تواند جایگزین مناسبی برای ارتباطات با سیم روبات و کامپیوتر باشد. در این حالت ارتباط بلوتوث از طریق پورت سریال /dev/rfcomm0 در لینوکس قابل استفاده است.

تنظیمات پورت سریال مربوط به بلوتوث به شرح زیر است:

- نرخ ارسال داده ۱۱۵۲۰۰ بیت بر ثانیه
- بیت داده ۸
- بدون بیت parity
- بدون بیت توقف^۲
- کنترل جریان سخت افزاری ندارد

1 Blue Giga

2 Stop bit

۳-۳- محیط ویندوز

- ﴿ روبات را روشن کنید، LED سبز رنگ باید روشن شود.
- ﴿ بلوتوث کامپیوتر را فعال کرده و با استفاده از گزینه Add a device روبات را جستجو کنید.
- ﴿ روبات با لوگوی کامپیوتراهای جیبی^۱ و نام KHIII به همراه شماره سریال خپرا، نمایش داده می‌شود. (شماره سریال خپرا روی برچسبی واقع بر مدار درون روبات آمده است.)
- ﴿ کد امنیتی "0000" را برای برقراری ارتباط بلوتوث، وارد کنید.
- ﴿ ارتباط باید امن باشد و در این مرحله یک پورت COM برای ارتباط سریال بلوتوث با روبات رزرو شود.

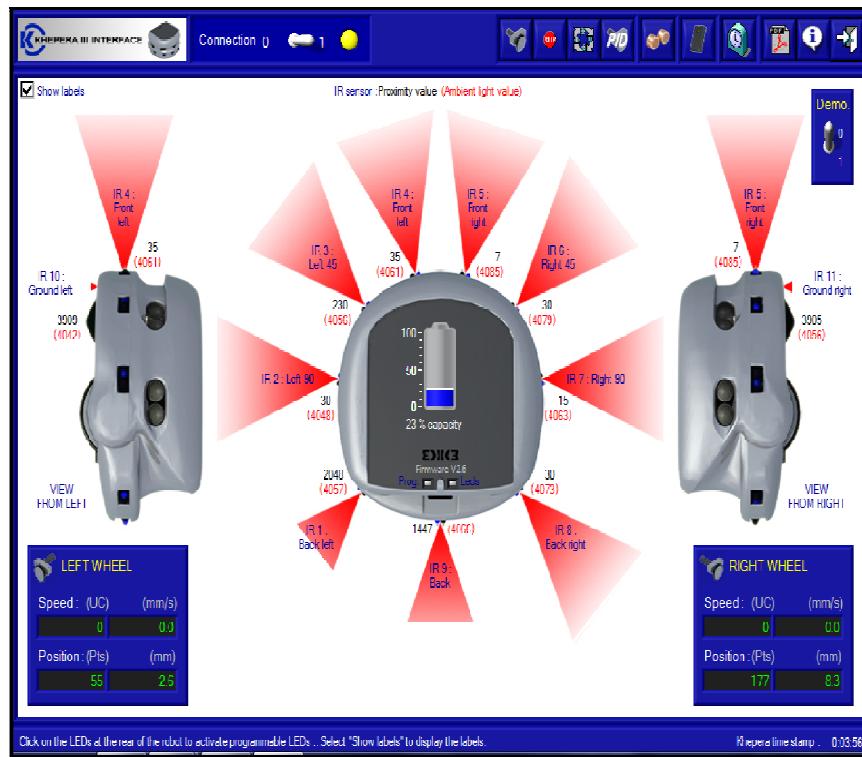
حال می‌توان از برنامه‌هایی که از پورت COM استفاده می‌کنند مانند III و Interface khepera کنترل از راه دور روبات استفاده کرد. RealTerm

Interface Khepera III - ۱-۳-۳

برنامه‌ی Interface Khepera III برنامه‌ای ساده و کاربرپسند^۲ برای آشنایی با ویژگی‌های روبات خپرا، دیدن دموها و کنترل روبات از طریق رابط گرافیکی است. برای استفاده از این برنامه، پس از برقراری ارتباط بلوتوث روبات و کامپیوتر، برنامه را اجرا کرده و با زدن دکمه‌ی Connection پورت COM مربوط به روبات را انتخاب کنید. شکل (۱-۳) محیط این برنامه را نشان می‌دهد.

¹ PDA

² User friendly



شکل (۱-۳) برنامه‌ی Interface Khepera III

RealTerm -۲-۳-۳

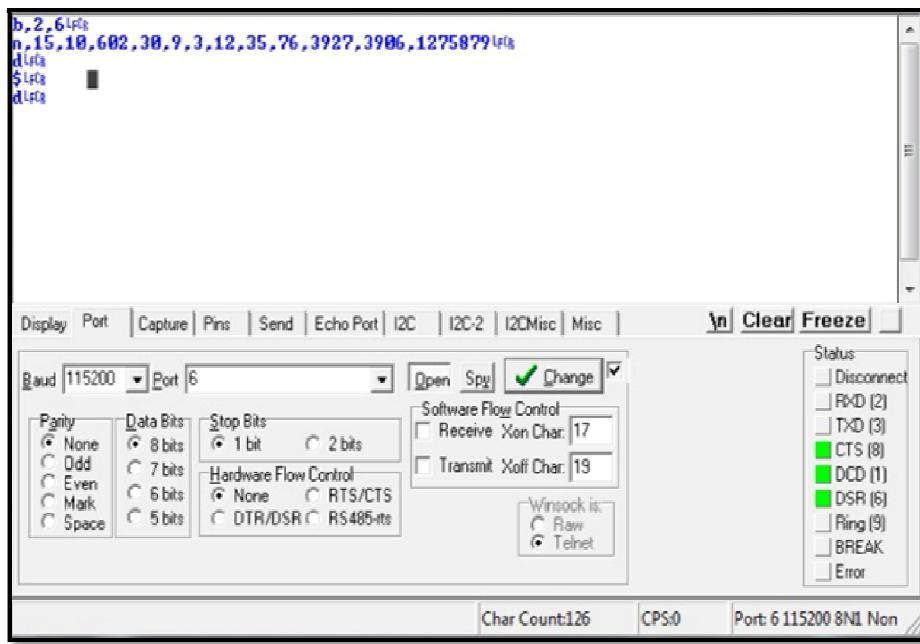
RealTerm یک ترمینال سریال برای ارتباط با پورت‌های COM است. این برنامه‌ی کد باز^۱، برای گرفتن^۲، کنترل کردن و اشکال زدایی داده‌های باینری و دیگر جریان‌های داده مناسب است که برخلاف برنامه‌ی HyperTerminal، مودم‌های شماره گیر^۳ را پشتیبانی نمی‌کند(8).

برای پیکربندی برنامه‌ی RealTerm، تب Port را انتخاب کرده و پورت سریال را همانند شکل (۳-۲) تنظیم کنید. سپس با زدن دکمه‌ی Change تغییرات را در برنامه اعمال کنید. برای فرستادن دستورات به روبات، باید از فرمت مخصوص دستورات که در بخش ۲-۳-۲ آمده، استفاده کرد. به عنوان مثال برای آگاهی از نسخه نرم افزار روبات، در ترمینال دستور B را ارسال کنید. باید توجه داشت که در این قسمت، فقط جواب‌های رسیده از روبات نمایش داده می‌شوند.

¹ Open source

² capturing

³ Dialing modem



شکل (۲-۳) یک ترمینال سریال برای ارتباط با پورت‌های COM است.

۴-۳- محیط لینوکس

۱-۴-۳- چرا از لینوکس استفاده می‌کنیم؟

با توجه به قابلیت‌های منحصر به فرد لینوکس و انعطاف پذیری آن نسبت به ویندوز، استفاده از سیستم عامل لینوکس برای کار با روبات خپرا توصیه می‌شود. از طرفی بورد Korebot خود دارای یک سیستم عامل لینوکس تعبیه شده است که برای دسترسی بهتر به آن، تغییر تنظیمات آن و همچنین برنامه نویسی و نصب برنامه‌ها و درایور دیگر مازول‌های روبات نیاز به استفاده از سیستم عامل لینوکس است.

در این پروژه از اوبونتو^۱ نسخه‌ی ۱۱/۰۴ برای ارتباط با روبات خپرا ۳ استفاده شده است.

¹ Ubuntu

۴-۲- برقراری ارتباط بلوتوث کامپیوتر و روبات در محیط لینوکس

برای سهولت در برقراری ارتباط بلوتوث با روبات، ابتدا با وارد کردن دستور زیر در ترمینال، برنامه blueman را نصب کنید.

Sudo apt-get install blueman

در محیط Ubuntu 11.4، از مسیر System>Preferences، برنامه Bluetooth Manager را اجرا کرده و جستجو کنید.

روبات با نام KHIII به همراه شماره سریالش نمایش داده می‌شود. در این حالت آدرس Mac بلوتوث روبات با فرمت استاندارد مشخص شده است.

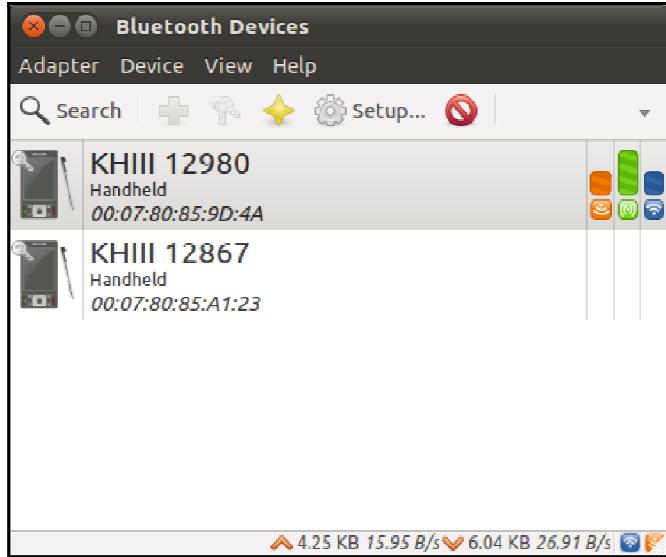
برای جفت ۱ شدن خپرا و کامپیوتر، کد امنیتی خپرا که «0000» است را وارد کنید.

با کلیک راست بر روی نام روبات، Setup را انتخاب کرده و گزینه Connect to Serial Port را انتخاب کنید. (شکل ۳-۳)

در صورتی که برقراری ارتباط بلوتوث موفقیت آمیز باشد، پیغامی مانند آنچه در شکل (۳-۴) آمده است ظاهر می‌گردد. همان طور که این پیغام بیان می‌دارد، ارتباط بلوتوث در لینوکس مانند یک پورت سریال در نظر گرفته می‌شود که از طریق /dev/rfcomm0 قابل دسترسی است.

¹ Pair

کنترل روبات خپرا ۳ از طریق شبکه بی سیم و پیاده سازی عبور روبات از ماز ارتباطات بی سیم خپرا ۳



شکل (۳-۳) برنامه blueman



شکل (۴-۳) ارتباط بلوتوث در لینوکس مانند یک پورت سریال در نظر گرفته می شود.

﴿ اگر برقراری ارتباط بلوتوث موفقیت آمیز نبود، با دستور زیر مازول بلوتوث کامپیوتر را ریست نموده، و مراحل قبل را تکرار کید.

Sudo /etc/init.d/Bluetooth restart

Minicom -۳-۴-۳

برنامه minicom یک برنامه متن محور^۱ برای ارتباطات سریال است که برای سیستم عامل های

^۱ Text-based

کنترل روبات خپرا ۳ از طریق شبکه بی سیم و پیاده سازی عبور روبات از ماز ارتباطات بی سیم خپرا ۳

لینوکس طراحی شده است(9).

برای فرستادن دستورات و فایل‌ها به روبات، از طریق پورت سریال rfcomm0 باید برنامه minicom را با دستورات زیر نصب و اجرا کنیم.

Sudo apt-get install minicom

Sudo minicom

﴿ دکمه های Ctrl و A را همزمان فشرده و پس از رها کردن آن‌ها دکمه Z را بزنید تا لیست دستورات و تنظیمات minicom نمایش داده شود.

﴿ برای انجام تنظیمات، Serial Port Configure Minicom را انتخاب کرده و سپس وارد Setup شوید.

﴿ وسیله سریال را با آدرس /dev/rfcomm0 مشخص کنید.

﴿ دقت کنید که ویژگی ارسال داده‌ها 8N1 115200 باشد و کنترل جریان نرم افزاری و سخت افزاری انتخاب نشده باشند.

﴿ با انتخاب save setup as dfl، تغییرات را ذخیره نمایید.

﴿ حال می‌توان همانند شکل ۵-۳ در صفحه اصلی minicom دستوراتی مانند B و N را با فرمت مخصوص خپرا ارسال کرد و جواب آن‌ها را در minicom مشاهده نمود.

```
Welcome to minicom 2.5

OPTIONS: I18n
Compiled on Feb  5 2011, 06:31:35.
Port /dev/rfcomm0

Press CTRL-A Z for help on special keys

b,2,6
n,1,6,36,1,28,46,70,21,39,3930,3932,4453614
d
a
```

شکل (۳-۵) برنامه minicom برای ارتباطات سریال در لینوکس

۴-۴-۳- برقراری ارتباط بلوتوث خپرا ۳ به همراه بورد Korebot II

- « تمام مراحل برقراری ارتباط بلوتوث روبات به همراه بورد Korebot به کامپیوتر، همانند حالت قبل است فقط در مرحله‌ی آخر در صفحه‌ی اصلی minicom زده و چند دقیقه منتظر بمانید تا لینوکس تعییه شده^۱ موجود روی بورد Korebot راه اندازی شود.
- « برای ورود به سیستم عامل بورد Korebot، از نام root استفاده کرده و در قسمت پسورد بزنید. Enter
- « حال به عنوان root@korebot2 به بورد دسترسی دارید (شکل ۶-۳).

```
Welcome to minicom 2.5
```

```
OPTIONS: I18n
Compiled on Feb 5 2011, 06:31:35.
Port /dev/rfcomm0
```

```
Press CTRL-A Z for help on special keys
```

```
# 
# Password: 
root@korebot2:~$
```

```
OpenEmbedded Linux korebot2 ttyS1
```

```
Angstrom 2007.9-test-20090708 korebot2 ttyS1
```

```
korebot2 login: root
Password:
```

```
Welcome to Korebot2!
```

```
root@korebot2:~$
```

شکل (۶-۳) ورود به سیستم عامل بورد Korebot

- « برای خروج از سیستم عامل بورد Korebot، در واقع باید از minicom خارج شوید. برای این کار دکمه‌های Ctrl A و پس از رها کردن آن‌ها دکمه X را بزنید.

¹ Embedded Linux

۳-۴-۱- فرستادن فایل به Korebot (آپلود)

- ﴿ بعد از برقراری ارتباط بلوتوث و اجرا کردن برنامه‌ی minicom، دکمه‌های Ctrl و a را فشرده و پس از رها کردن آن‌ها، دکمه‌ی s را فشار دهید.
- ﴿ zmodem را انتخاب کنید.
- ﴿ حال باید فایل مورد نظر را برای ارسال به Korebot انتخاب کنید. با دکمه‌های جهت دار جابجا شوید، برای تغییر دایرکتوری دو بار دکمه‌ی فاصله و برای انتخاب فایل یک بار دکمه‌ی فاصله را بزنید.
- ﴿ با انتخاب گزینه OK فایل ارسال می‌شود.
- ﴿ فایل ارسالی به آخرین دایرکتوری که minicom در آن بوده است، انتقال می‌یابد. با استفاده از دستور ls می‌توانید از وجود فایل مطمئن شوید.

۳-۴-۲- فرستادن فایل به کامپیوتر (دانلود)

- ﴿ در ترمینال Korebot ابتدا به دایرکتوری مورد نظر رفته، سپس دستور زیر را وارد کنید. به جای FILE_NAME اسم فایلی را که می‌خواهید از Korebot به کامپیوتر منتقل کنید بنویسید.

lsz FILE_NAME

- ﴿ برای یافتن مکان فایل ارسالی در کامپیوتر، می‌توان از دستور زیر استفاده کرد.
find / -name FILE_NAME

۵-۳- تکنولوژی wifi

بلوتوث و wifi تکنولوژی‌های متفاوتی برای ارتباط بی سیم هستند. بلوتوث برای انتقال اطلاعات در فواصل کوتاه (۱۰ متر) و بین دو تا هفت وسیله است، در حالی که تکنولوژی wifi تا محدوده‌ی ۱۰۰ متر را پوشش می‌دهد و وسائل زیادی می‌توانند از طریق wifi به یک شبکه متصل شوند و با هم ارتباط داشته باشند. بنابراین راه اندازی ارتباط wifi روبات و کامپیوتر از اهمیت ویژه‌ای برخوردار است و می‌توان با استفاده از یک شبکه بی سیم چندین روبات و کامپیوتر را به هم مرتبط کرد(10).

۱-۵-۳- راه اندازی کارت وایرلس

نحوه نصب و راه اندازی کارت فلش فشرده‌ی بی سیم^۱ مدل Ambicom WL5400G-CF در ادامه آمده است.

﴿ هنگامی که روبات خاموش است، کارت wifi را درون Korebot قرار دهید. دقت کنید که برای اضافه کردن ماذول‌ها یا جدا کردن آن‌ها، حتماً باید روبات و ماذول خاموش باشند. در غیر این صورت موجب آسیب دیدن روبات و ماذول می‌گردد..

﴿ برای Load شدن ماذول wifi به طور اتوماتیک، باید pxa2xx_cs را به فایل /etc/modules بورد Korebot اضافه کنید.

echo pxa2xx_cs >>/etc/modules

﴿ تنظیمات مربوط به شبکه بی سیم:
۱. برای شبکه‌های بدون رمز گذاری^۲، باید فایل /etc/network/interfaces بورد را اصلاح کنید. برای این کار باید از یک ویرایشگر متن^۳ موجود در لینوکس Korebot، به نام vi استفاده کنید (برای اطلاع از دستورات و نحوه کار با vi به بخش ۹-۳ مراجعه نمایید).

¹ Wireless compact flash card

² Encryption

³ Text-editor

vi /etc/network/interfaces

کد زیر را طبق تنظیمات شبکه^۵ بی سیم، تغییر داده و به این فایل اضافه کنید.

****** /etc/network/interfaces ******

The loopback interface

#

auto lo

iface lo inet loopback

#

Wireless interfaces

#

auto eth0

#iface eth0 inet dhcp

iface eth0 inet static

wireless_mode managed

wireless_essid YOUR_SSID_OF_NETWORK

address YOUR_IP_ADDRESS

netmask YOUR_NETMASK

gateway YOUR_GATEWAY_IP

SSID نام شبکه بی سیمی است که می‌خواهید روبات به آن متصل شود. •

دقت کنید نام شبکه باید یک کلمه‌ای و بدون استفاده از کاراکتر فاصله

باشد.

اگر dhcp server مسیریاب^۱ فعال نباشد، روند IP دهی به دستگاه‌های

درون شبکه، به طور استاتیک می‌گیرد. پس باید به جای

your_IP_Address، به صورت دستی IP روبات را مشخص کنید.

¹ Router

- کد های اضافی موجود در فایل را با به کارگیری کاراکتر # به توضیح^۱ تبدیل کنید.

برای شبکه هایی با رمزگذاری WEP و WPA و غیره، به راهنمای Korebot .ii مراجعه نمایید.

با دستور زیر مأذول Network روبات را مجدداً راه اندازی کنید.

/etc/init.d/networking restart

اگر مراحل قبلی به درستی انجام شده باشد، پس از راه اندازی مجدد پیغام done می دهد؛ شکل (۷-۳) . در غیر این صورت پیغام خطای حاوی ifdown و ifup می دهد که برای تصحیح آن باید به فایل interfaces مراجعه کنید.

```
auto eth0
iface eth0 inet static
    wireless_mode managed
    wireless_essid test
    address 192.168.1.33
    netmask 255.255.255.0
    gateway 192.168.1.1
root@korebot2:~$ /etc/init.d/networking restart
Reconfiguring network interfaces... done.
root@korebot2:~$
```

شکل (۷-۳) کارت wifi بطور صحیح راه اندازی شده است.

پیشنهاد می شود که برای راه اندازی اولیه کارت wifi، مودم را روی حالت بدون رمزگذاری و بدون پسورد پیکربندی کنید و dhcp server آن را غیرفعال نمایید. در صورتی که کارت wifi بدون مشکل راه اندازی شود، چراغ های روی کارت روشن می شوند. برای

^۱ Comment

کنترل روبات خپرا ۳ از طریق شبکه بی سیم و پیاده سازی عبور روبات از ماز ارتباطات بی سیم خپرا ۳

دیدن IP کارت از دستور ifconfig -a (شکل ۸-۳) و برای چک کردن اتصال به شبکه از دستور ping استفاده می شود. در شکل ۹-۳ روبات، کامپیوتر با IP ۱۹۲.۱۶۸.۱.۳ را ping کرده است.

```
root@korebot2:~$ ifconfig -a
eth0      Link encap:Ethernet HWaddr 00:1A:6B:0C:34:F4
          inet addr:192.168.1.33 Bcast:192.168.1.255 Mask:255.255.255.0
                  UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
                  RX packets:48 errors:0 dropped:0 overruns:0 frame:0
                  TX packets:42 errors:0 dropped:0 overruns:0 carrier:0
                  collisions:0 txqueuelen:1000
                  RX bytes:5831 (5.6 KiB) TX bytes:6806 (6.6 KiB)

lo      Link encap:Local Loopback
          inet addr:127.0.0.1 Mask:255.0.0.0
                  UP LOOPBACK RUNNING MTU:16436 Metric:1
                  RX packets:32 errors:0 dropped:0 overruns:0 frame:0
                  TX packets:32 errors:0 dropped:0 overruns:0 carrier:0
                  collisions:0 txqueuelen:0
                  RX bytes:2336 (2.2 KiB) TX bytes:2336 (2.2 KiB)
```

شکل (۸-۳) برای دیدن IP کارت از دستور ifconfig -a استفاده می شود.

```
root@korebot2:~$ ping 192.168.1.3
PING 192.168.1.3 (192.168.1.3): 56 data bytes
64 bytes from 192.168.1.3: icmp_seq=0 ttl=64 time=2.8 ms
64 bytes from 192.168.1.3: icmp_seq=1 ttl=64 time=1.7 ms
64 bytes from 192.168.1.3: icmp_seq=2 ttl=64 time=1.7 ms
64 bytes from 192.168.1.3: icmp_seq=3 ttl=64 time=1.8 ms
64 bytes from 192.168.1.3: icmp_seq=4 ttl=64 time=8.0 ms
64 bytes from 192.168.1.3: icmp_seq=5 ttl=64 time=2.8 ms
64 bytes from 192.168.1.3: icmp_seq=6 ttl=64 time=5.5 ms
64 bytes from 192.168.1.3: icmp_seq=7 ttl=64 time=2.1 ms
64 bytes from 192.168.1.3: icmp_seq=8 ttl=64 time=1.8 ms
64 bytes from 192.168.1.3: icmp_seq=9 ttl=64 time=2.6 ms
64 bytes from 192.168.1.3: icmp_seq=10 ttl=64 time=1.8 ms

--- 192.168.1.3 ping statistics ---
11 packets transmitted, 11 packets received, 0% packet loss
round-trip min/avg/max = 1.7/2.9/8.0 ms
```

شکل (۹-۳) برای چک کردن اتصال به شبکه از دستور ping استفاده می شود.

۳-۱-۱-۱- ویرایشگر vi

vi یک ویرایشگر غیر گرافیکی معروف، قدرتمند و غیر کاربرپسند است که برای تغییر بعضی فایل‌های پیکربندی^۱ تنها ویرایشگر موجود است(11).

vi در دو حالت اصلی کار می‌کند، یکی برای دادن دستورات و دیگری برای تغییر متن. هنگام شروع، برنامه در حالت «فرمان»^۲ است. در این حالت می‌توان مکان نما^۳ را با دکمه‌های جهت دار^۴ یا دکمه‌های i و k و j و h جابجا کرد، متن را کپی، پیس^۵ و حذف کرد، تغییرات را ذخیره نمود و از برنامه خارج شد.

برای تغییر دادن به حالت درج^۶ از دکمه I استفاده می‌شود. در این حالت، فشردن کاراکترها باعث اضافه شدن آن‌ها در موقعیت مکان نما می‌شود و دکمه Backspace برای حذف کاراکترها غیرفعال است. برای بازگشت دوباره به حالت «فرمان» از Esc استفاده می‌شود(12).

در حالت «فرمان» برای پاک کردن کاراکتر زیر مکان نما^۷، برای باز کردن خط جدید^۸، برای حذف یک خط dd، برای خنثی کردن^۹ آخرین تغییر ii استفاده می‌شود. در این حالت برای ذخیره تغییرات و خارج شدن از برنامه از دستور ZZ یا wq: و برای خروج از برنامه بدون اعمال تغییرات روی فایل از !q: استفاده می‌شود(13).

باید توجه کرد که لینوکس و vi هر دو حساس به اندازه‌ی حروف^{۱۰} هستند.

۳-۵-۲- ارتباط با Korebot با استفاده از wifi و ssh

بعد از راه اندازی کارت wifi، می‌توان برای ارتباط با بورد Korebot، دستور زیر را در یک ترمینال لینوکس اجرا کرد که در آن به جای KOREBOT_IP، IP خپرا قرار گیرد.

¹ Configuration files

² Command

³ Cursor

⁴ Arrow keys

⁵ Paste

⁶ Insert

⁷ Undo

⁸ Case Sensitive

کنترل روبات خپرا ۳ از طریق شبکه بی سیم و پیاده سازی عبور روبات از ماز ارتباطات بی سیم خپرا ۳

ssh root@KOREBOT_IP

بعد از اجرای این دستور، ترمینال لینوکس به کنسول Korebot تبدیل می‌شود.

۳-۵-۳- انتقال فایل با استفاده از wifi و scp

بعد از اتصال روبات به شبکه، برای انتقال فایل از کامپیوتر به بورد Korebot دستور زیر را در ترمینال لینوکس وارد کنید که در آن به جای file/path آدرس فایل در کامپیوتر قرار می‌گیرد. به عنوان مثال در شکل ۱۰-۳ فایل برنامه‌ی اجرایی khepera3_test به روبات ارسال شده است.

scp file/path root@KOREBOT_IP :/home/root

```
comi@comi-VGN-Z530N:~$ scp Documents/khepera3_test root@192.168.1.33:/home/root
root@192.168.1.33's password:
khepera3_test                                         100% 2866      2.8KB/s   00:00
```

شکل (۱۰-۳) انتقال فایل khepera3_test از کامپیوتر به بورد Korebot

۶-۳- پروتکل nfs و تنظیمات آن

پروتکل سیستم فایل شبکه^۱ یا به اختصار nfs، از طریق شبکه این امکان را فراهم می‌کند که کامپیوترهای لینوکس قسمتی از دیسک سخت‌شان^۲ را روی یک کامپیوتر لینوکس دور^۳ قرار دهند^۴ به نحوی که گویی دیسک سخت محلی آن کامپیوتر است(14).

با استفاده از پروتکل nfs کامپیوتر به سرور nfs و Korebot به مشتری^۵ nfs تبدیل می‌شوند تا از طریق لینوکس موجود بر Korebot بتوان به داده‌های ذخیره شده روی کامپیوتر دسترسی داشت و از آن‌ها استفاده کرد.

¹ Network file system

² Hard Disk

³ Remote Linux machine

⁴ Mount

⁵ Client

کنترل روبات خپرا ۳ از طریق شبکه بی سیم و پیاده سازی عبور روبات از ماز ارتباطات بی سیم خپرا ۳

راه اندازی پروتکل nfs شامل دو بخش است: راه اندازی سرور nfs و راه اندازی مشتری nfs

۱-۶-۳- راه اندازی سرور nfs (کامپیوتر)

﴿ برای نصب بسته های^۱ مورد نیاز، دستور زیر را در ترمینال وارد کنید.

```
apt-get install nfs-kernel-server nfs-common portmap
```

﴿ در هنگام پیکربندی portmap آن را به loopback محدود نکنید! اما اگر اشتباهًا portmap را محدود نمودید، برای تصحیح آن دستور زیر را اجرا کنید.

```
sudo dpkg-reconfigure portmap
```

﴿ سرویس ها را مجدد راه اندازی کنید.

```
sudo /etc/init.d/nfs-kernel-server restart
```

﴿ برای شروع مجدد portmap می توانید از یکی از دو دستور زیر استفاده کنید.

```
sudo /etc/init.d/portmap restart
```

```
sudo service portmap restart
```

﴿ با دستور rpcinfo چک کنید که nfs به درستی اجرا شده باشد. باید یک لیست از برنامه های RPC در حال اجرا را ببینید که شامل mountd, portmapper, nlockmgr و nfs باشد(۱۵). یک نمونه از این لیست در شکل ۱۱-۳ نشان داده شده است.

```
rpcinfo -p localhost
```

¹ Packages

² Bind

```
comi@comi-VGN-Z530N:~$ sudo service portmap restart
portmap start/running, process 680
comi@comi-VGN-Z530N:~$ rpcinfo -p localhost
program vers proto port
 100000  2   tcp    111  portmapper
 100000  2   udp    111  portmapper
 100024  1   udp   58521  status
 100024  1   tcp   44593  status
 100003  2   tcp   2049  nfs
 100003  3   tcp   2049  nfs
 100003  4   tcp   2049  nfs
 100227  2   tcp   2049
 100227  3   tcp   2049
 100003  2   udp   2049  nfs
 100003  3   udp   2049  nfs
 100003  4   udp   2049  nfs
 100227  2   udp   2049
 100227  3   udp   2049
 100021  1   udp  41765  nlockmgr
 100021  3   udp  41765  nlockmgr
 100021  4   udp  41765  nlockmgr
 100021  1   tcp  46824  nlockmgr
 100021  3   tcp  46824  nlockmgr
 100021  4   tcp  46824  nlockmgr
 100005  1   udp  55013  mountd
 100005  1   tcp  40274  mountd
 100005  2   udp  55013  mountd
```

شکل (۱۱-۳) با دستور rpcinfo باید یک لیست از برنامه های در حال اجرا بینید.

حال باید دایرکتوری یا فایل هایی که می خواهید با روبات به اشتراک بگذارید را در فایل /etc(exports) اعلام کنید. ابتدا دایرکتوری nfsarm را در قسمت mnt کامپیوتر بسازید، سپس خط زیر را به فایل exports اضافه کنید.

vi /etc(exports)

/mnt/nfsarm KOREBOT_IP(rw, no_root_squash, sync)

در خط بالا، عبارت *rw* به معنی این است که به روبات اجازه خواندن و نوشتمن در دایرکتوری nfsarm را می دهد. از طریق عبارت *no_root_squash* اجازه می دهد که کاربر root در روبات، بتواند مانند root سرور عمل کند و در نهایت، عبارت *sync* به معنای همزمانی است(16).

☞ هرگاه فایل exports /etc(exports را تغییر می‌دهد باید دستور زیر را اجرا کنید.

exportfs -av

اگر بعد از اجرای این دستور، اخطاری مبنی بر عدم وجود دایرکتوری دریافت کردید، باید آدرس دایرکتوری در فایل exports را کامل‌تر بدهید.

sudo /etc/init.d/nfs-kernel-server restart

۳-۶-۲- راه اندازی مشتری (Korebot) nfs

☞ حال باید دایرکتوری‌ای که کامپیوتر به اشتراک گذاشته است را روی سیستم فایل قرار دهید.^۱ ابتدا دایرکتوری /mnt/nfs را روی Korebot ساخته و سپس دستور mount را اجرا کنید.

makedir /mnt/nfs

mount -t nfs -o nolock COMPUTER_IP:/mnt/nfsarm

اگر دستور بالا با شکست مواجه شد و پیغام permission denied داد، یعنی روزی کامپیوتر در حال اجرا نیست. باید ابتدا در ترمینال کامپیوتر nfs-kernel-server را مجدد راه اندازی نمود و دوباره دستور را اجرا کرد.

حال اگر در لینوکس Korebot به پوشه^۲ /mnt/nfs بروید و سپس دستور ls را اجرا کنید، می‌توانید محتويات دایرکتوری /mnt/nfsarm کامپیوتر را ملاحظه کنید و در صورت نیاز فایل‌ها را خوانده و تغییر دهید.

¹ Mount

² Folder

فصل ۴:

برنامه نویسی بر روی خپرا ۳

۱-۴- مقدمه

در این فصل به بررسی نحوه نصب light tool chain و استفاده از این مجموعه ابزار برنامه نویسی، برای ساخت یک برنامه اجرایی پرداخته خواهد شد. در ادامه، کد «برنامه عبور روبات خپرا از مازهای ساده» بررسی می شود.

۲-۴- نصب Light toolchain

راهنمای نصب نرم افزار مورد نیاز برای استفاده از بورد و ابزار توسعه در این بخش شرح داده شده است.

۱-۲-۴- راهنمای نصب نرم افزار

در زیر لیست فایل های مورد نیاز برای نصب نرم افزار light toolchain را مشاهده می کنید. این فایل ها علاوه بر اینکه در بسته DVD موجود می باشند، می توان آن ها را از آدرس اینترنتی <http://ftp.k-team.com/KorebotII/software> نیز دریافت نمود.

	: light_toolchain/	← از پوشه
korebot2-oetools-light-1.0-	: Cross-compiler light	•
	kb1.tar.bz2	
	: common_files/	← از پوشه
development_k2_v1.0.tar.bz2	: Development folder	•
libkorebot-1.11-kb1.tar.bz2	: Board library sources	•
	Script for uploading	•
ktboot-2.0_20090416.tar.bz2	: kernel and file system	

۴-۲-۲- نصب پوشه‌ی development

پوشه‌ی development اصلی برای توسعه نرم افزار می‌باشد که حاوی لینک و اسکریپت‌هایی برای به کارگیری cross-compiler به منظور ساخت برنامه‌ها می‌باشد.

۱. فایل development_k2_v1.0.tar.bz2 را در پرونده‌ی home خود از حالت فشرده خارج کنید. سپس به پوشه‌ی development_k2_v1.0 که به تازگی ایجاد شده وارد شوید. این پوشه از این پس پرونده‌ی توسعه‌ی شما خواهد بود. با فرض اینکه کنسول را به کار می‌برید، فرمان‌های زیر قابل استفاده هستند:

```
tar -xjf development_k2_v1.0.tar.bz2 -C ~/  
cd ~/development_k2_v1.0
```

۲. متغیر KTEAM_HOME در فایل env.sh را به گونه‌ای تغییر دهید که به پرونده‌ی توسعه اشاره کند. برای این منظور فیلد YOUR_USERNAME را با حساب کاربری سیستم عامل خود جایگزین کنید.

```
KTEAM_HOME=/home/YOUR_USERNAME/development_k2_v1.0
```

۴-۲-۳- نصب (light toolchain) cross-compiler

۱. فایل korebot2-oetools-light-1.0-kb1.tar.bz2 را در آدرس /usr/local به وسیله‌ی فرمان زیر از حالت فشرده خارج کنید:

```
sudo tar -xjf korebot2-oetools-light-1.0-kb1.tar.bz2 -C /usr/local
```

توجه: شما باید در حالت root بوده و یا از sudo استفاده کنید.

۲. به وسیله‌ی فرمان زیر یک سیمبلینک^۱ در پرونده‌ی توسعه‌ی خود ایجاد کنید.

```
ln -s /usr/local/korebot2-oetools-1.0/tmp/cross ~/development_k2_v1.0/cross
```

۳. برای کسب اطمینان از اینکه نصب درست انجام گرفته است، می‌بایست

¹ symlink

را اجرا کنید. قبل از هر چیز دسترسی به متغیرهای محیطی^۱ را برقرار کرده و نسخه cross-compiler را چک کنید.

source env.sh

arm-angstrom-linux-gnueabi-gcc -version

فرمان آخر باید خروجی زیر را بدهد.

arm-angstrom-linux-gnueabi-gcc (GCC) 4.1.2

۴-۲-۴- نصب کتابخانه‌ی بورد libkorebot

کتابخانه هم اکنون بر روی korebot نصب شده است. برای نصب کتابخانه بر روی سیستم توسعه‌ی خود دستورالعمل زیر را دنبال کنید.

۱. کتابخانه‌ی libkorebot-1.11-kb1.tar.bz2 را در پوشه‌ی توسعه‌ی خود با دستور زیر از حالت فشرده خارج کنید.

tar -xjf libkorebot-1.11-kb1.tar.bz2 -C ~/development_k2_v1.0

۲. با اجرای فرمان‌های زیر می‌توان کل کتابخانه را در پوشه‌ی libkorebot-1.11-kb1 دوباره کامپایل کرد.

source ./env.sh

make clean

make

نسخه‌ی جدید libkorebot را می‌توان از سایت ftp زیر دانلود کرد:

<http://ftp.k-team.com/korebot/libkorebot>

^۱ Environment variables

۴-۲-۵- نصب اسکریپت برای آپلود کرنل و سیستم فایل: ktboot

برای مقداردهی تنظیمات پیش فرض korebot لازم است به وسیله اسکریپت زیر فایل ktboot-2.0_20090416.tar.bz2 را در پوشه‌ی توسعه‌ی خود از حالت فشرده خارج کنید.

```
tar -xjf ktboot-2.0_20090416.tar.bz2 -C ~/development_k2_v1.0
```

۴-۳- کاربرد Light toolchain

نرم افزار light toolchain برای ساخت برنامه‌های قابل اجرا بر روی روبات به کار می‌رود.

۱-۳-۴- توسعه‌ی نرم افزار^۱

یک قالب^۲ برنامه به نام prog-template.c در کتابخانه‌ی بورد libkorebot، در پوشه‌ای به آدرس libkorebot-1.11-kb1/template موجود است. می‌توان از این قالب برنامه، برای شروع کد نویسی و از دستورات زیر برای ساخت برنامه استفاده کرد.

ابتدا در ترمینال با دستور cd وارد پوشه‌ی libkorebot-1.11-kb1/template شوید و سپس دستورات زیر را وارد کنید.

```
source ../../env.sh
```

```
make
```

دستور اول متغیرهای محیطی^۳ را در دسترس سیستم می‌گذارد. دستور دوم اسکریپت^۴ را اجرا می‌کند تا برنامه‌ی اجرایی^۵ کامپایل و ساخته شود.

فایل template ساخته شده، در واقع برنامه‌ی قابل اجرا بر روی روبات است. حال می‌توان این

¹ Application Development

² Template

³ Environment variables

⁴ Script

⁵ Executable program

کنترل روبات خپرا ۳ از طریق شبکه بی سیم و پیاده سازی عبور روبات از ماز برنامه نویسی بر روی خپرا ۳

برنامه را از طریق بلوتوث (بخش ۳-۴-۱ آپلود کردن فایل) یا از طریق ارتباط بی سیم (بخش ۳-۵ دستور scp یا بخش ۳-۶ پروتکل nfs) به Korebot منتقل کرد و با دستور زیر آن را اجرا نمود.

/template

تعدادی برنامه‌ی آماده‌ی اجرا در بخش scr/tests از پوشه‌ی Korebot هستند که می‌توانید آن‌ها را بر روی روبات اجرا کنید.

توجه: اگر نام برنامه را عوض کردید، باید قبل از کامپایل آن، تکرارهای نام برنامه در اسکریپت Makefile را نیز عوض نمایید.

۴-۴- برنامه نویسی برای روبات خپرا ۳

زبان برنامه نویسی برای روبات خپرا، زبان C است. برای شروع برنامه نویسی، توصیه می‌شود از طریق آدرس زیر به کتابخانه‌ی KoreBot رفته، فایل kb-khepera3.c و نیز برنامه‌ی khepera3- test.c به همراه توضیحات مربوطه مطالعه شوند. بدین ترتیب می‌توان با توابع کتابخانه‌ی KoreBot و نحوه استفاده از آن‌ها آشنا شد.

<http://ftp.k-team.com/korebot/libkorebot-doc/files.html>

نحوه کامپایل برنامه و ساخت فایل اجرایی آن در ۴-۳-۱- (توسعه‌ی نرم افزار) توضیح داده شده است.

۴-۵- طرح مسئله

مهم‌ترین بخش در این پروژه، نوشتمن یک برنامه و پیاده سازی آن به طور عملی بر روی روبات خپرا است؛ برنامه‌ای که بتواند چگونگی به کارگیری قابلیت‌های اساسی روبات را به نمایش بگذارد. بدین ترتیب، روبات می‌تواند با استفاده از اطلاعات دریافتی از محیط واقعی، به طور مستقل تصمیم گیری کند و عکس‌العمل مناسب نشان دهد.

۶-۴- برنامه‌ی عبور خپرا از مازهای ساده

نوشتن «برنامه‌ی عبور خپرا از مازهای ساده بدون برخورد به دیواره‌ها» به خوبی می‌تواند نحوه‌ی به کارگیری قابلیت‌های پایه‌ی ای خپرا را نشان دهد. در این برنامه برای حرکت صحیح روبات در ماز، باید از طریق سنسورهای اولتراسونیک فاصله‌ی روبات تا دیواره‌ها را به دست آورد و بر اساس مقادیر آن‌ها موتورها را کنترل کرد. برای نوشتن این برنامه باید از نحوه‌ی تنظیم و عملکرد سنسورهای اولتراسونیک و کنترل کننده‌های موتور خپرا اطلاع داشت. دستورات پروتکل ارتباطی خپرا و کامپیوتر در بخش ۲-۳ و توابع کتابخانه‌ای متناظر با آن‌ها در کتابخانه‌ی Korebot و برنامه‌ی khepera3-test.c آمده‌اند.

به طور کلی برنامه از سه بخش اصلی تشکیل شده است:

۱. خواندن سنسورهای اولتراسونیک و به دست آوردن فاصله
۲. حرکت کردن روبات به طور مستقیم و به اندازه‌ی مناسب (جلو یا عقب)
۳. چرخیدن روبات در جای خود به اندازه‌ی ۹۰ درجه (چپ یا راست)

۶-۱- توضیح برنامه

تابع main با فراخوانی تابع initKH3() روبات خپرا را مقداردهی اولیه می‌کند. در صورتی که تابع initKH3() با موفقیت انجام شود، در ادامه سنسورهای اولتراسونیک را پیکربندی می‌کند. در این مرحله تمام پنج سنسور اولتراسونیک فعال و تعداد اکوها یک می‌شود.

در حلقه‌ی اصلی برنامه، روبات ۷ بار مقادیر تمام سنسورهای اولتراسونیک را خوانده و به ازای هر سنسور، مقدار فاصله با بیشترین دامنه را به عنوان بهترین جواب در آرایه best_US ذخیره می‌کند. سپس بر اساس وجود یا عدم وجود مانع در مقابل روبات و همچنین فاصله‌ی روبات با دیواره‌ها، روبات یا بطور مستقیم حرکت می‌کند و یا در جای خود ۹۰ درجه می‌چرخد.

برنامه بر اساس مقادیر سنسور جلو(شماره ۳)، سنسور چپ(شماره ۱) و سنسور راست(شماره ۵) تصمیم می‌گیرد. اگر فاصله سنسورهای چپ و راست هر دو برابر مقدار صفر باشند، بدین معنا است

کنترل روبات خپرا ۳ از طریق شبکه بی سیم و پیاده سازی عبور روبات از ماز برنامه نویسی بر روی خپرا ۳

که روبات در مکان فعلی نمی‌تواند فاصله تا دیواره‌ها را بطور صحیح اندازه بگیرد، پس روبات برای اطلاع یافتن از فواصل به اندازه ۴ سانتی‌متر به عقب می‌رود تا مکانش را بهبود بخشد.

اما در غیراین صورت، فاصله روبروی خپرا بررسی می‌شود، اگر فاصله زیاد (بیش از ۴۰ سانتی‌متر) باشد، خپرا با گام‌های ۱۵ سانتی‌متری جلو می‌رود. اگر فاصله متوسط (بین ۳۰ تا ۴۰ سانتی‌متر) باشد، خپرا با گام‌های ۵ سانتی‌متری مستقیماً جلو می‌رود. اما اگر فاصله کم (کمتر از ۲۵ سانتی‌متر) باشد، یعنی خپرا به دیوار ماز نزدیک شده است و حال باید با بکارگیری مقادیر سنسورهای چپ و راست، تصمیم بگیرد که به کدام سمت بچرخد. اگر فاصله مربوط به سنسور سمت راست زیادتر بود، روبات در جای خود ۹۰ درجه به راست می‌چرخد و در غیر این صورت به چپ می‌چرخد.

در صورت فشردن هر دکمه‌ای اجرای برنامه متوقف می‌شود.

۴-۶-۲- کد برنامه

```
/* Simple Maze Solving program for Khepera 3
Zeinab Mapar - 86521418          ABAN 1390 */

#include <korebot/korebot.h>
#include <stdlib.h>
#include <stdio.h>

#define STEP 15 //cm
#define SPEED 20000
#define MAXBUFFERSIZE 100
#define PI 3.14159265358979
#define WHEEL_DISTANCE 88.41 //mm
#define MM_PER_PULSE 0.0466
```

```
/*! handle to the various khepera3 devices (knet socket, i2c
mode) */

static knet_dev_t * dsPic;

static knet_dev_t * mot1;

static knet_dev_t * mot2;

/*-----*/
/*! initMot initializes then configures the motor control
* unit.

* \return A value :
*         - 1 if success
*         - 0 if any error
*/
int initMot(knet_dev_t *hDev)
{
    if(hDev)
    {
        kmot_SetMode( hDev , kMotModeIdle );
        kmot_SetSampleTime( hDev , 1550 );
        kmot_SetMargin( hDev , 6 );
        if(hDev == mot1)
            kmot_SetOptions( hDev , 0x0 , kMotSWOptWindup |
kMotSWOptStopMotorBlk | kMotSWOptDirectionInv );
        else
            kmot_SetOptions( hDev , 0x0 , kMotSWOptWindup |
kMotSWOptStopMotorBlk );
        kmot_ResetError( hDev );
        kmot_SetBlockedTime( hDev , 10 );
        kmot_SetLimits( hDev , kMotRegCurrent , 0 , 500 );
        kmot_SetLimits( hDev , kMotRegPos , -10000 , 10000 );

        /* PID */
    }
}
```

```
kmot_ConfigurePID( hDev ,kMotRegSpeed,620,3,10 );

kmot_ConfigurePID( hDev,kMotRegPos,600,20,30);

/*maxspeed=15000 accelerate=30 */

kmot_SetSpeedProfile(hDev,15000,30);

return 1;

}

else

{

printf("initMot error, handle cannot be null\r\n");

return 0;

}

}

/*-----*/
/*! initKH3 initialize various things in the kh3 then

* sequentially open the various required handle to the three
*i2c devices on the khepera3 using knet_open from the knet.c
*libkorebot's modules. Finally, this function initializes then
*configures the motor control unit.

* \return A value :
*         - 0 if success
*         - <0 if any error
*/
int initKH3( void )

{

/* This is required */

kh3_init();

/* open various socket and store the handle in their
respective pointers */

dsPic = knet_open( "Khepera3:dsPic" ,KNET_BUS_I2C,0,NULL );
mot1  = knet_open( "Khepera3:mot1" ,KNET_BUS_I2C,0,NULL );
mot2  = knet_open( "Khepera3:mot2" ,KNET_BUS_I2C,0,NULL );

if(dsPic!=0)
```

کنترل روبات خپرا ۳ از طریق شبکه بی سیم و پیاده سازی عبور روبات از ماز برنامه نویسی بر روی خپرا ۳

```
{  
    if( (mot1 !=0) && (mot2 !=0) )  
    {  
        initMot(mot1);  
        initMot(mot2);  
        return 0;  
    }  
    else  
        return -1;  
}  
return -2;  
}  
/*-----*/  
/*! configureOS configures various parameters inside the kh3  
*firmware using kb_khepera3.c library. */  
  
int configureOS( int argc, char * argv[], void * data)  
{  
    char Buffer[MAXBUFFERSIZE]; /* buffer that handle the  
returned datas from kh3 */  
  
    short index; /* variable that handles index */  
    short value; /* variable that handle value */  
  
    /* Retrive the arguments from the parameter */  
  
    index = atoi(argv[1]);  
    value = atoi(argv[2]);  
  
    /* Configure */  
  
    if(kh3_configure_os((char *)Buffer, index, value, dsPic))  
        printf("\r\n%c\r\n", Buffer[0]);  
    else  
        printf("\r\nnc, error... \r\n");  
}  
/*-----*/
```

کنترل روبات خپرا ۳ از طریق شبکه بی سیم و پیاده سازی عبور روبات از ماز برنامه نویسی بر روی خپرا ۳

```
void move_forward_p(float pos_cm)
{
    float fpos;
    long lpos,rpos;
    // read old encoder position
    lpos = kmot_GetMeasure(mot1, kMotRegPos);
    rpos = kmot_GetMeasure(mot2, kMotRegPos);

    fpos = pos_cm*10.0/MM_PER_PULSE;//(1pulseunit= 0.0466mm)
    printf("moving      forward      cm=      %.1f      in      pulses=
%.1f\n",pos_cm,fpos);

    /* tell motor controllers to move K3 forward, using speed
and accel profile */
    kmot_SetPoint(mot1, kMotRegPosProfile, lpos+(long)fpos);
    kmot_SetPoint(mot2, kMotRegPosProfile, rpos+(long)fpos);
}

/*-----*/
void move_backward_p(float pos_cm)
{
    float fpos;
    long lpos,rpos;
    // read old encoder position
    lpos = kmot_GetMeasure(mot1, kMotRegPos);
    rpos = kmot_GetMeasure(mot2, kMotRegPos);

    fpos = pos_cm*10.0/MM_PER_PULSE;//(1pulseunit= 0.0466mm)
    printf("moving      BACKward      cm      =      %.1f      in      pulses=
%.1f\n",pos_cm,fpos);

    /* tell motor controllers to move K3 backward, using speed
and accel profile */
```

کنترل روبات خپرا ۳ از طریق شبکه بی سیم و پیاده سازی عبور روبات از ماز برنامه نویسی بر روی خپرا ۳

```
kmot_SetPoint(mot1, kMotRegPosProfile, lpos-(long)fpos);
kmot_SetPoint(mot2, kMotRegPosProfile, rpos-(long)fpos);

}

/*-----*/
void rotate_right_p(float degree)
{
    float fpos;
    long lpos,rpos;
    lpos = kmot_GetMeasure(mot1, kMotRegPos);
    rpos = kmot_GetMeasure(mot2, kMotRegPos);
    fpos = degree/360.0*PI*WHEEL_DISTANCE/MM_PER_PULSE; // from degree to mm to pulse

    printf("rotating right degree = %.1f in pulses=% .1f\n",degree,fpos);

    /* tell motor controllers to rotate K3, using speed and accel profile */
    kmot_SetPoint(mot1, kMotRegPosProfile, lpos+(long)fpos);
    kmot_SetPoint(mot2, kMotRegPosProfile, rpos-(long)fpos);
}

/*-----*/
void rotate_left_p(float degree)
{
    float fpos;
    long lpos,rpos;
    lpos = kmot_GetMeasure(mot1, kMotRegPos);
```

```
rpos = kmot_GetMeasure(mot2, kMotRegPos);

fpos = degree/360.0*PI*WHEEL_DISTANCE/MM_PER_PULSE; // from degree to mm to pulse

printf("rotating    left    degree    =    %.1f    in    pulses= %.1f\n",degree,fpos);

/* tell motor controllers to rotate K3, using speed and accel profile */

kmot_SetPoint(mot1, kMotRegPosProfile, lpos-(long)fpos);

kmot_SetPoint(mot2, kMotRegPosProfile, rpost+(long)fpos);

}

/*-----*/
/*! measureUS retrieves ultrasonic measure from a given transceiver and fill response[3] array with ultrasonic measures.

 * \param 1st first param (us_index) is the us number to read from (1 to 5)
 */

int measureUS( int us_index,int response[3])

{

char Buffer[MAXBUFFERSIZE];

int i;

//short usnoise; /* Noise on the given adc pin when no us is received */

short echonbr; /* Number of echo part of this us measure */

short usvalue; /* Variable that handle distances */

short usampl; /* Variable than handle amplitudes */

short argument;

argument = us_index;

if(kh3_measure_us((char *)Buffer, argument, dsPic))

{

/* We guess the echo number ( how many echos have been received from a captor ) */

}
```

```
echonbr = (Buffer[1] | Buffer[2]<<8);

response[0]=echonbr;

/* Loop as may time it is required */

if(echonbr < 10)

{

    for(i = 0; i < echonbr ; i++)

    {

        /* Get the distance measure from one echo */

        usvalue = (Buffer[i*8+3] | Buffer[i*8+4]<<8);

        usamp1 = (Buffer[i*8+5] | Buffer[i*8+6]<<8);

        /* fill response array wiht the result */

        response[1]=usvalue;

        response[2]=usamp1;

    }

}

else

    printf("read error\r\n");

}

else

    printf("\r\ng, error...");

}

/*-----*/
/*! Test if anykey was pushed
 * \return -1 if error occured
 *         >=0  number of characters to read
```

کنترل روبات خپرا ۳ از طریق شبکه بی سیم و پیاده سازی عبور روبات از ماز برنامه نویسی بر روی خپرا ۳

```
*/  
  
int kbhit(void)  
{  
    int cnt = 0;  
    int error;  
    static struct termios Ottty, Ntty;  
  
    tcgetattr( 0, &Otty);  
    Ntty = Ottty;  
  
    /*  
     * Set up the terminal to raw mode.  
     */  
    Ntty.c_iflag = 0; /* input mode */  
    Ntty.c_oflag = 0; /* output mode */  
    Ntty.c_lflag &= ~ICANON; /* raw mode */  
    Ntty.c_cc[VMIN] = CMIN; /* minimum time to wait */  
    Ntty.c_cc[VTIME] = CTIME; /* minimum characters to  
    wait for */  
  
    if (0 == (error = tcsetattr(0, TCSANOW, &Ntty))) {  
        error += ioctl(0, FIONREAD, &cnt);  
        error += tcsetattr(0, TCSANOW, &Otty);  
    }  
  
    return ( error == 0 ? cnt : -1 );  
}  
  
/*! Change terminal mode for getchar to return immediately  
*  
* \param dir 1= mode changed to non-blocking, 0 mode  
reverted to previous
```

کنترل روبات خپرا ۳ از طریق شبکه بی سیم و پیاده سازی عبور روبات از ماز برنامه نویسی بر روی خپرا ۳

```
* \return none
*/
void change_term_mode(int dir)
{
    static struct termios oldt, newt;

    if ( dir == 1 )
    {
        tcgetattr( STDIN_FILENO, &oldt);
        newt = oldt;
        newt.c_lflag &= ~( ICANON | ECHO );
        tcsetattr( STDIN_FILENO, TCSANOW, &newt);
    }
    else
        tcsetattr( STDIN_FILENO, TCSANOW, &oldt);
}

/*-----*/
int main(int argc, char *argv[])
{
    short rc,i,j;
    char Buffer[MAXBUFFERSIZE];
    long lpos,rpos;
    int best_US[5][2]; //a 2D array that maintain distance & amplitude of 5 US!
    int temp[3]; // return distance & amplitude of a US sensor!

    // Set the libkorebot debug level
    kb_set_debug_level(2);
```

```
// Init the korebot library

if((rc = kb_init( argc , argv )) < 0 )

    return 1;

printf("\n Khepera3 Maze Solver program \n  Zeinab Mapar
Aban 1390 \r\n");

if(!initKH3())

{

    printf("\n Init ok... \r\n");




    kh3_revision((char *)Buffer, dsPic);

    printf("\r\n  %c,%4.4u,%4.4u  =>  Version  =  %u,
Revision = %u\r\n",
           Buffer[0],          (Buffer[1]
Buffer[2]<<8), (Buffer[3] | Buffer[4]<<8),
           (Buffer[1]          |      Buffer[2]<<8),
(Buffer[3] | Buffer[4]<<8));



    printf("\n \n");

    // configure to use all the US

    kh3_configure_os((char *)Buffer, 0, 31, dsPic);




    // configure to have the first echo only

    kh3_configure_os((char *)Buffer, 1,1, dsPic);


    sleep(3);




    // loop until any key is pushed

    while (!kbhit())


{
```

```
//initiate best_us array with 0!
for(i=0;i<5;i++)
{
    for (j=0;j<2;j++)
    {
        best_US[i][j]=0;
    }
}

/*-----reading US sensors 7 times-----*/
for(i=0;i<7;i++)
{
    // j= number of US sensor
    for (j=1;j<=5;j++)
    {
        measureUS(j,temp);
        if(temp[0]==1) //echonumber=1
        {
            if(temp[2]>best_US[j-1][1])
            {
                best_US[j-1][0]=temp[1];
                best_US[j-1][1]=temp[2];
            }
        }
    }
}

// print out all US measures
for(i=0;i<5;i++)
{
```

کنترل روبات خپرا ۳ از طریق شبکه بی سیم و پیاده سازی عبور روبات از ماز برنامه نویسی بر روی خپرا ۳

```
printf("US sensor= %d: distance=%d      amplitude=%d      \n",i,
best_US[i][0],best_US[i][1]);  
}  
  
/*-----Decision Making base on Distance-----*/  
/*if right & left US sensors return distance=0, measures are  
*not true!so move backward 4cm to get a better US measures.*/  
  
if(best_US[4][0]==0 && best_US[0][0]==0)  
{move_backward_p(4);}  
  
else  
{  
    if(best_US[2][0]>40)  
    {  
        move_forward_p(15);  
    }  
  
    else if(best_US[2][0]<=40 && best_US[2][0]>30)  
    {  
        move_forward_p(5); // move 5 cm forward  
    }  
  
    /* if ahead distance (for US.num=3) is shorter than 25 cm,  
     * use right & left sensors */  
    else  
{  
    if(best_US[4][0]>best_US[0][0])  
        rotate_right_p(90);  
  
    else if(best_US[0][0]>best_US[4][0])  
        rotate_left_p(90);  
    else
```

کنترل روبات خپرا ۳ از طریق شبکه بی سیم و پیاده سازی عبور روبات از ماز برنامه نویسی بر روی خپرا ۳

```
move_backward_p(2);  
}  
}  
sleep(3);  
}  
printf("\n End\n");  
}  
return 0;  
}
```

فصل ۵:

جمع‌بندی و پیشنهادها

۱-۵- مقدمه

در انجام این پروژه دو هدف کلی مد نظر بوده است. هدف اول کنترل روبات و انجام تنظیمات آن، از طریق بلوتوث و همچنین شبکه‌ی بی‌سیم است. هدف دوم نوشتن برنامه‌ای برای عبور روبات از یک ماز ساده و در انتهای پیاده‌سازی آن به طور عملی بر روی روبات خپرا است. با توجه به اینکه این پروژه دو هدف کلی و متوالی دارد مراحل انجام کار نیز به دو بخش تقسیم می‌شوند.

مراحل انجام بخش اول پروژه: ابتدا ارتباط بلوتوث روبات و کامپیوتر را در محیط‌های ویندوز و لینوکس برقرار کرده و از آن به عنوان یک خط سریال برای انتقال اطلاعات استفاده شده است. در مرحله‌ی بعد با بکار گیری پروتکل ارتباطی مخصوص روبات، می‌توان دستورات پایه را از کامپیوتر به خپرا ارسال کرد و پاسخ‌های دریافتی از روبات را نیز مشاهده کرد و از این طریق به کنترل توابع روبات پرداخت.

در ادامه نصب و راهاندازی کارت wifi بر روی روبات بررسی شده و به چگونگی اتصال روبات به شبکه‌ی بی‌سیم پرداخته شده است. بعد از اتصال روبات به شبکه‌ی بی‌سیم با بهره‌گیری از پروتکل nfs، کامپیوتر را به سرور nfs و روبات را به مشتری nfs تبدیل می‌کنیم به طوری که کامپیوتر بتواند قسمتی از هارددیسک خود را روی لینوکس روبات قرار دهد. در این صورت روبات از طریق لینوکس موجود در برد korebot می‌تواند به داده‌های ذخیره شده روی کامپیوتر دسترسی یابد و برنامه‌های اجرایی موجود در هارد دیسک کامپیوتر را مستقیماً اجرا کند.

مراحل انجام بخش دوم پروژه: در این بخش ابتدا نحوه‌ی نصب light tool chain و سپس استفاده از این مجموعه ابزار برنامه‌نویسی، برای ساخت یک برنامه‌ی اجرایی بررسی شده است. در ادامه، کد «برنامه‌ی عبور روبات خپرا از مازهای ساده» را به زبان C نوشته و بر روی روبات پیاده‌سازی می‌کنیم تا روبات بتواند با اجرای آن، به طور مستقل از مازهای ساده عبور کند.

۲-۵- جمع‌بندی

با انجام مراحل مذکور، اهداف اساسی این پروژه محقق شده‌اند. درواقع می‌توان از طریق بلوتوث یا شبکه‌ی بی‌سیم به کنترل روبات و انجام تنظیمات آن و همچنین تبادل فایل بین کامپیوتر و روبات پرداخت.

استفاده از کارت WiFi و اتصال به شبکه‌ی بی‌سیم نسبت به ارتباط بلوتوث سه مزیت عمده دارد: اول، بعد از راهاندازی اولیه‌ی کارت WiFi روبات همواره به محض روشن شدن، به طور خودکار به شبکه‌ی بی‌سیم مورد نظر وصل می‌شود.

دوم، پوشش شبکه‌ی بی‌سیم از محدوده‌ی بلوتوث بسیار بیشتر است، پس برای کنترل روبات از فواصل دور اتصال آن به شبکه‌ی بی‌سیم مناسب‌تر است.

سوم، با استفاده از شبکه‌ی بی‌سیم و پروتکل nfs روبات به قسمتی از هارددیسک کامپیوتر مانند حافظه‌ی محلی خود دسترسی می‌یابد.

از طرفی پیاده سازی «برنامه‌ی عبور روبات از ماز‌های ساده» بطور عملی روی روبات خپرا موفقیت آمیز بوده است. روبات خپرا با استفاده از اطلاعات دریافتی از محیط واقعی بطور مستقیم تصمیم گیری می‌کند و بدون برخورد به دیواره‌ها از ماز عبور می‌کند.

۳-۵- پیشنهادها

با توجه به اینکه در این پروژه اتصال روبات به شبکه بی‌سیم و برنامه نویسی بر روی آن انجام شده است، می‌توان پروژه‌های بعدی را در راستای ارتباط و همکاری بین چند روبات خپرا تعریف نمود.

مراجع

مراجع

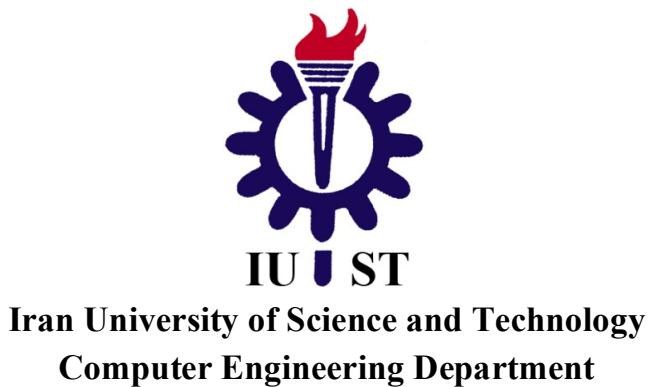
- [1] *A survey of socially interactive robots.* **T. Fong, I. Nourbakhsh, and K. Dautenhahn.** s.l. : Robotics and Autonomous Systems, 2003.
- [2] Khepera III. *K-Team.* [Online] [Cited: 11 1, 2011.] <http://www.k-team.com/mobile-robotics-products/khepera-iii>.
- [3] **Bernabeu, David Ar'an.** Multi-Robot Behavioural Algorithms Implementation in Khepera III Robots. *final project degree in industrialengineering.* 2010.
- [4] **J.Tharin.** KoreBot II ver 1.3. *user manual.* 2011.
- [5] **F.Lambercy.** Khepera III robot v1.3. *user manual.* 2011.
- [6] Introduction to Pulse Width Modulation (PWM). *Netrino.* [Online] [Cited: 11 1, 2011.] <http://www.netrino.com/Embedded-Systems/How-To/PWM-Pulse-Width-Modulation>.
- [7] RFCOMM Protocol. [Online] [Cited: 11 1, 2011.] <http://www.palomewireless.com/infotooth/tutorial/rfcomm.asp>.
- [8] RealTerm. [Online] [Cited: 11 1, 2011.] <http://realterm.sourceforge.net/>.
- [9] Minicom. [Online] [Cited: 11 1, 2011.] <http://searchenterpriselinux.techtarget.com/definition/minicom>.
- [10] wifi Vs. bluetooth. [Online] [Cited: 11 1, 2011.] <http://www.freewimaxinfo.com/wifi-vs-bluetooth-comparison.html>.
- [11] vi editor. [Online] [Cited: 11 1, 2011.] <http://linux.about.com/library/bl/open/newbie/blnewbie7.1vi.htm>.
- [12] vi editor. [Online] [Cited: 11 1, 2011.] <http://www.linuxclues.com/articles/08.htm>.
- [13] vi commands. [Online] [Cited: 11 1, 2011.] http://www.yolinux.com/TUTORIALS/LinuxTutorialAdvanced_vi.html.
- [14] nfs protocol. [Online] [Cited: 11 1, 2011.] http://nfs.sourceforge.net/nfs-howto/ar01s02.html#whatis_nfs.
- [15] File sharing with nfs. [Online] [Cited: 11 1, 2011.] <http://www.techienote.com/2010/10/file-sharing-in-ubuntu-with-nfs.html>.
- [16] nfs in linux. [Online] [Cited: 11 1, 2011.] http://www.dba-oracle.com/oracle_tips_linux_nfs.htm.

Abstract:

KheperaIII is a mobile robot with wheels which is in use in many universities in the world for researches. This project has been done in two stages. In the first stage, using Bluetooth and communication protocol, basic commands are sent to the robot to control its functions. Then a WIFI card has been installed on the robot and it has been connected to the wireless network. After connecting to the Wireless network, computer and robot are developed to NFS server and NFS client by applying NFS protocol.

In the second stage, a program, “passing Khepera through simple mazes”, has been written in C programming language and implemented on the robot so that the robot be able to pass through simple mazes independently.

Keywords: KheperaIII robot, KorebotII, korebotII programming



Controlling KheperaIII Robot via Wireless Network and Implementing of Passing It through a Maze

**A Thesis Submitted in Partial Fulfillment of the Requirement for the
Degree of Bachelor of Science in Computer Engineering**

**By:
Zeinab Mapar**

**Supervisor:
Dr.Mozayani**

November 2011