

# معرفی نرم افزار Qusber

## محصول جدید گروه NREC

نرم افزار کاسبر (Qusber) یک نرم افزار میانی مدیریت و کنترل دستگاه های جانبی (در حال حاضر برای دستگاه های USB) است که به صورت ویژه برای دستگاه های درون کار (Embedded Devices) طراحی و پیاده سازی شده است. این نرم افزار با برقراری ارتباط با دستگاه USB می تواند داده های مشخص از دستگاه را در زمان دقیق مقرر شده یا انتظار رویداد (Event driven) از دستگاه دریافت کرده و یا داده های مورد نظر کاربر را به روش دلخواه به دستگاه انتقال دهد. به طور کلی کاربرد این نرم افزار در دستگاه هایی است که توسط رایانه و سیستم های پردازشی با توان بالا کنترل می شوند. مانند: ربات ها، دستگاه های آزمایشگاهی، سیستم های اتوماسیون صنعتی، صنایع کنترل فرایند و ...

بزرگترین مزیت Qusber، ضریب امنیت بالا در عملکرد و سرعت است که این مهم به دلیل استفاده ی حداقلی از منابع داخلی سیستم عامل و همچنین عدم نیاز به درایور های پیچیده و همه منظوره (General Purpose) موجود در سیستم عامل ها پدید آمده است.

این نرم افزار در راستای تکمیل بخشی از "پروژه ی طراحی بازوی صنعتی ربات" به عنوان سیستم عامل آغاز به کار کرد، ولی در حال حاضر قابلیت نصب و راه اندازی روی انواع دستگاه های مهندسی پزشکی و یا انواع ربات ها با نیاز های پردازشی بالا را دارد.

نرم افزار Qusber از ابتدا و از کوچکترین موارد از مراحل نیازمندی و طراحی تا پیاده سازی و پشتیبانی و حتی مستندسازی به طول کامل از توان متخصصان گروه به همراه استفاده از نرم افزارها و ابزارهای کاملاً رایگان و یا متن باز (Open Source) بهره می برد.

در زیر بخشی از مشخصات کلی ابزارها و بخش‌های مورد استفاده می‌آید:

- زبان برنامه‌نویسی (*Programming Language*): زبان **C** ▲
- استاندارد کدنویسی (*Coding Standard*): **gnuc++0** ▲
- مترجم (*Compiler*): مترجم متن بازگنو **GCC** نسخه **4.5.3** ▲
- الگوی رشته (*Threading Model*): الگوی **Posix** ▲
- سکو (*Platforms*): سکوهای **Windows, GNU/Linux** ▲
- مجوز (*License*): مجوز خصوصی (*Propriety*) ▲
- برنامه‌نویس (*Programmer*): عرفان اسماعیلی برزی (*Erfan Esmayili Barzi*) ▲
- مشاور (*Consultant*): سید محسن طباطبایی فر (*Mohsen Tabatabayi Far*) ▲
- نرم‌افزار پارسنجی (*Profiler*): نرم‌افزار **Gprof** نسخه **2.21.0.20110327** ▲
- نرم‌افزار مهار نسخه (*CVS*): نرم‌افزار **Git** نسخه **1.7.4.1** ▲
- نرم‌افزار پردازش متن (*Text Processor*): نرم‌افزار **VIM** نسخه **7.3.35** ▲
- نرم‌افزار مستندسازی (*Document Tool*): نرم‌افزار **LibreOffice Writer** نسخه **3.3.2** ▲
- مستندساز (*Document Writer*): عرفان اسماعیلی برزی (*Erfan Esmayili Barzi*) ▲
- نسخه (*version*): نسخه **1.02** ▲
- تاریخ آخرین ویرایش (*Last Edition Date*): تاریخ **Tue Jun 14 5:17:02 IRDT 2014** ▲

## بخش (Module) های گوناگون قابل استفاده

### بخش پیکربندی (Configuration Module):

در این بخش کاربر می‌تواند تنظیمات از پیش تعیین شده مانند شماره‌ی Product ID دستگاه‌های مورد نظر و ... را در پرونده (File) مشخص ذخیره کند و از وارد کردن دوباره و یا چندباره‌ی آن جلوگیری به عمل آید. قالب این پرونده از روش‌های مختلف نوشتار پیکربندی از جمله پیکربندی تودرتو (Nested)، ایجاد فهرست (List) و ... پشتیبانی می‌کند.

### بخش رابط کاربری (UI):

رابط کاربری سامانه به دو صورت پوسته‌ی دستوری (Command Shell) و پوسته‌ی گرافیکی در دسترس است. در رابط کاربری گرافیکی از کتابخانه (Library) ی GTK برای که تحت میزکار Gnome نسخه‌ی ۲ طراحی شده، اما در باقی میزکارها نیز قابل استفاده خواهد بود.



### بخش ثبت زمان (Timing Module):

به دلیل بحرانی بودن برخی کاربرهای این نرم‌افزار مانند کاربردهایی که زمان دقیق انجام کار یا دریافت رویداد با دقت نانو ثانیه مدنظر است، می‌توان با بهره‌گیری از این بخش، دقت زمانی ثبت رویدادها، گزارش‌گیری و یا انجام وظایف را به صورت دقیق‌تری پایان داد.

## بخش زمان بند (Scheduling Module):



این افزونه با به کارگیری امکانات سیستم عامل می تواند مدیریت توزیع رشته ها بر روی پردازنده ی مرکزی، الگوریتم زمان بندی قرارگیری فرایند برو روی پردازنده ، الویت بارگذاری فریایند و ... را مدیریت کند.

بدین وسیله می توان نسبت به مدیریت فرایند برنامه ی **Qusber** نسبت به دیگر فرایندها تصمیم گیری کرد تا در صورت نیاز به اولویت سنجی، از آن بهره برد.

## بخش گزارش گیری (Logging Module):



این افزونه با استفاده از افزونه ی زمان گیری می تواند از کوچکترین تا مهم ترین رویدادهای درون نرم افزار را با اولویت سنجی که توسط کاربری انجام می شود، در خروجی نرم افزار در صفحه ی نمایش یا محیط کاربری گرافیکی و یا درون یک پرونده ی گزارش گیری چاپ کند.

```
2014/5/10 10:32:0.492867567 $ DEBUG> Qdis:: Set recv buff size: done successfully.
2014/5/10 10:32:0.492887123 $ DEBUG> Qdis:: enabling udev event receiving: done successfully.
2014/5/10 10:32:0.492894526 $ DEBUG> Qdis:: locking mutex: done successfully.
2014/5/10 10:32:0.492900183 $ DEBUG> Qdis:: Else condition: done successfully.going to MONITOR_OK
2014/5/10 10:32:0.492906748 # DEBUG> Qdis:: Before start receiving: listening to proper signals em
2014/5/10 10:32:4.857744834 # INFO> Qdis:: A new event reached: remove.
2014/5/10 10:32:4.857766066 # DEBUG> Qdis:: Print udev event: Properties list entry.
2014/5/10 10:32:4.857794910 # INFO> Qdis::      name: ACTION.
2014/5/10 10:32:4.857803431 # INFO> Qdis::      value: remove.
2014/5/10 10:32:4.857811812 # INFO> Qdis::      name: DEVPATH.
2014/5/10 10:32:4.857818586 # INFO> Qdis::      value: /devices/LNXSYSTM:00/device:00/PNP0COA:00
2014/5/10 10:32:4.857825361 # INFO> Qdis::      name: SUBSYSTEM.
```

## بخش کشف اتصال (Discovery Module):



این افزونه با برقراری ارتباط (link) با هسته (Kernel) سیستم عامل به تمامی رویدادهایی که مربوط به اتصال، تغییر و یا حذف دستگاه متصل شده می باشد، گوش فرا می دهد و در صورت ثبت کنترل کننده (Handler) برای آن می تواند به دسترسی سریع تر و با بار پردازشی کمتر نسبت به جست و جوی همهی دستگاهها فرامین کاربر را اجرا کند. به طور مثال کاربر می تواند با ثبت یک کنترل کننده برای یک دستگاه با مشخصات معین، به محض اتصال آن به ماشین، از آن مطلع شده و فرامین لازم را صادر کند. این عمل نسبت به جست و جوی همهی دستگاهها در زمانهای مشخص بار پردازشی بسیار کمتری را به ماشین و سیستم عامل تحمیل می کند.

```
[sudo] password for erfan:
2014/5/10 10:32:0.491696259 $ WARN> Qsig:: Registering SIGINT handler: done successfully.
2014/5/10 10:32:0.491777135 @ WARN> Qsig:: Registering SIGSTOP handler: failed unfortunately.
2014/5/10 10:32:0.491785307 @ WARN> Qsig:: Registering SIGKILL handler: failed unfortunately.
2014/5/10 10:32:0.491791872 $ WARN> Qsig:: Registering SIGTERM handler: done successfully.
2014/5/10 10:32:0.491797878 $ WARN> Qsig:: Registering SIGABRT handler: done successfully.
2014/5/10 10:32:0.491804164 $ WARN> Qsig:: Registering SIGQUIT handler: done successfully.
2014/5/10 10:32:0.491810100 $ WARN> Qsig:: Registering SIGSEGV handler: done successfully.
2014/5/10 10:32:0.491816316 $ WARN> Qsig:: Registering SIGILL handler: done successfully.
2014/5/10 10:32:0.491822113 $ WARN> Qsig:: Registering SIGHUP handler: done successfully.
2014/5/10 10:32:0.491828748 $ WARN> Qsig:: Registering SIGFPE handler: done successfully.
2014/5/10 10:32:0.491834545 $ WARN> Qsig:: Registering SIGTRAP handler: done successfully.
2014/5/10 10:32:0.491840551 # INFO> Main:: Initializing Program: Hurrrrra.
```

## بخش رشته‌بندی (Threading Module):



این افزونه با استفاده از الگوی رشته‌بندی POSIX سبب می‌شود تا بخش‌های مختلف برنامه به جای استفاده از فرایندهای مجزا، در یک فضای کاری مشترک با حافظه و بار پردازشی کمتری بر روی پردازنده‌ی مرکزی قرار گیرد تا تضمین سرعت و دقت برنامه تأمین شود.

از طرفی با بهره‌گیری از رشته‌های سطح پایین که نزدیک به سطح ماشین هستند، با وجود فضای کاری مشترک و جلوگیری از نیاز به ابزارهایی مانند Piping و ... برای ارتباط فرایندها، تقریباً به عنوان یک فرایند مجزا در نظر گرفته می‌شود تا با بهره‌گیری از مجزا بودن هر بخش یا تابع، انتزاع لازم برای اموری نظیر آزمایش و رفع اشکال برنامه را آسان‌تر کند

## بخش مدیریت سیگنال (Signal Handling):



در این بخش با استفاده از کتابخانه‌ی مدیریت سیگنال، تمامی سیگنال‌های شناخته‌شده‌ی سیستم‌عامل لینوکس و ویندوز مدیریت و دریافت شده و بسته به پیکربندی کاربر می‌تواند عکس‌العمل مناسب پاسخ مناسب دهد. به عنوان نمونه سیگنال INT با شماره‌ی ۲ می‌تواند به عنوان پایان‌دهنده‌ی کلی نرم‌افزار محسوب شود.

```

2014/5/10 10:46:7.133558715 # INFO> Qdis:: ... Value: (null).
2014/5/10 10:46:7.133606467 # DEBUG> Qdis:: Print udev event: Devlinks list entry.
2014/5/10 10:46:7.133614010 # DEBUG> Qdis:: Print udev event: device tags list entry.
*****
2014/5/10 10:46:7.133624137 $ DEBUG> Qdis:: Unlocking mutex: done successfully.End of the loop
2014/5/10 10:46:7.133631959 $ DEBUG> Qdis:: locking mutex: done successfully.
2014/5/10 10:46:7.133639292 $ DEBUG> Qdis:: Else condition: done successfully.going to MONITOR_OK
2014/5/10 10:46:7.133646556 # DEBUG> Qdis:: Before start receiving: listening to proper signals emi
qqqq^CInterrupt
2014/5/10 10:46:12.843366559 # WARN> Qsig:: Receiving SIGINI: (null).
^CInterrupt
2014/5/10 10:46:13.573333824 # WARN> Qsig:: Receiving SIGINI: (null).
^CInterrupt
2014/5/10 10:46:13.783319675 # WARN> Qsig:: Receiving SIGINI: (null).
^CInterrupt

```

## بخش هسته (Core Part):



این نرم افزار قابلیت انتقال داده به/از دستگاه را به دو شکل کلی همزمان (Synchronous) و ناهمزمان (Asynchronous) و چهار روش انتقال وقفه (Interrupt)، تکه (Bulk)، مهار (Control) و ناهمزمان استاندارد (Isochronous) دارا می باشد. همچنین امکان جست و جو برای دستگاهایی که با درگاه USB به سیستم عامل متصل هستند (USB Devices) به دو روش سعی و خطا برای اتصال (Exhaustive Search) و یا با انتظار برای رسیدن علامت مجزا (Signal) از طریق فعال سازی Discovery Module فراهم می باشد.

```

+++++
endp addr = 1
MaxPacketSize for endp 1= 8
endpoint Descr type=5
endpoint attr type=3
+++++
2014/5/4 13:12:12.623610507 $ DEBUG> Qdump:: Opening device and get device handle: done successfully.
2014/5/4 13:12:12.623634044 # DEBUG> Qdump:: Kernel driver is active: Try to detach it.
2014/5/4 13:12:12.623679720 $ DEBUG> Qdump:: Claiming interface: done successfully.
Qdump> 2 bytes read. usbData[0] = 2.
Qdump> 2 bytes read. usbData[1] = 69.
Qdump> 2 bytes read. usbData[2] = 0.

```

پیرامون ارسال یا دریافت داده ها، به دلیل استفاده از الگوی رشته بندی سطح پایین (LOW Level Threading)، قابلیت استفاده مجزای هر رشته (Thread) از هسته های متمایز از پردازنده های مرکزی (CPU) وجود دارد که این قابلیت توسط بخش زمان بند برای کل نرم افزار آماده سازی می شود.

در صورت فعال بودن بخش گزارش گیری نرم افزار، تمامی بخش های آن بر اساس پیکربندی

اعمال شده توسط کاربر، امکان گزارش‌گیری از بخش هسته تا تمامی زیر بخش‌ها مهیا شده است.

جهت بالا بردن قابلیت بهنگام‌بودن (Real-time) بخش هسته به هنگام استفاده از دستگاه، می‌تواند راه‌انداز خود سیستم‌عامل را برای دستگاه‌های USB غیر فعال کند و یا با استفاده از بخش Signaling علامات وارده را کنترل و مدیریت نماید.

این نرم‌افزار در بخش آزمایش و بررسی، با نرم‌افزار Gprof تحت شرایط مختلف و زمان‌های کاری زیاد پاسخ مثبت و قابل قبولی از خود نشان داده است.

به طور کلی این نرم‌افزار می‌تواند در صنایع مختلف به ویژه در رباتیک به صورت یک سامانه‌ی یک‌پارچه‌ی امن در قالب یک راه‌انداز یا رابط برای دستگاه‌های USB نقش ایفا کند که با استفاده از محصولات جانبی مانند سامانه‌های Real-time می‌توان توان امنیتی آن را به صورت فزاینده افزایش داد.

**پایان**