

# Специализированный семинар «День Технологий QNX»



Использование механизма  
публикации/подписки для  
межзадачного взаимодействия в QNX

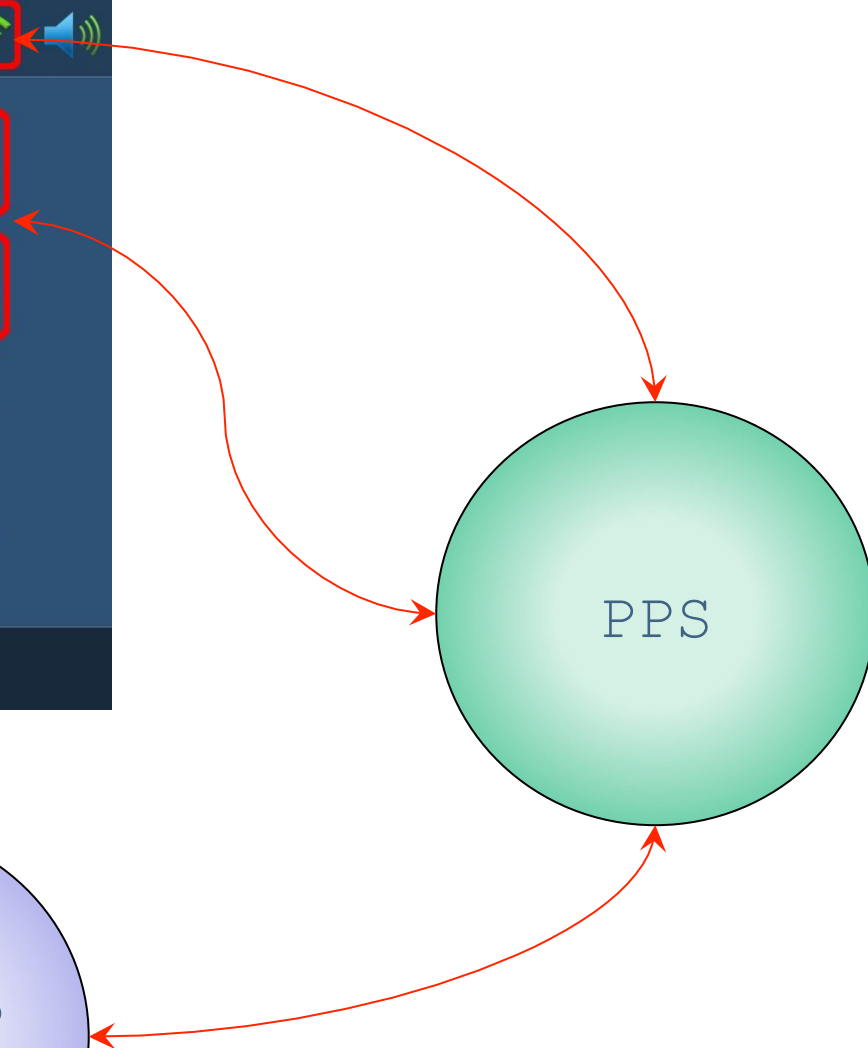
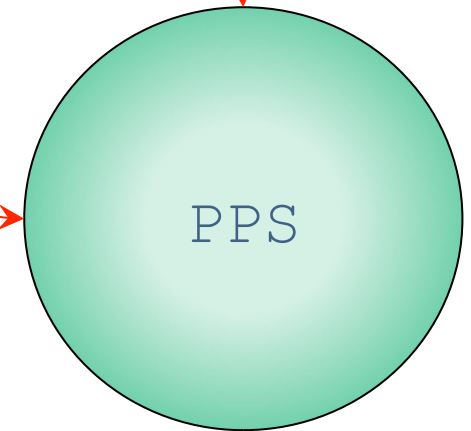
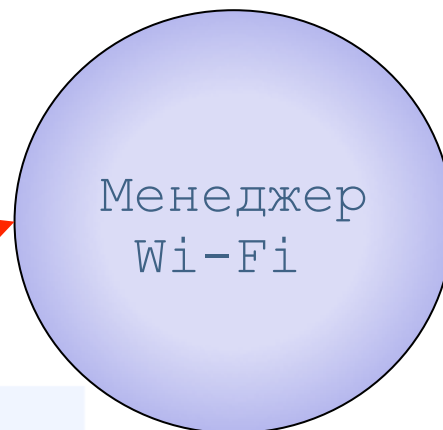
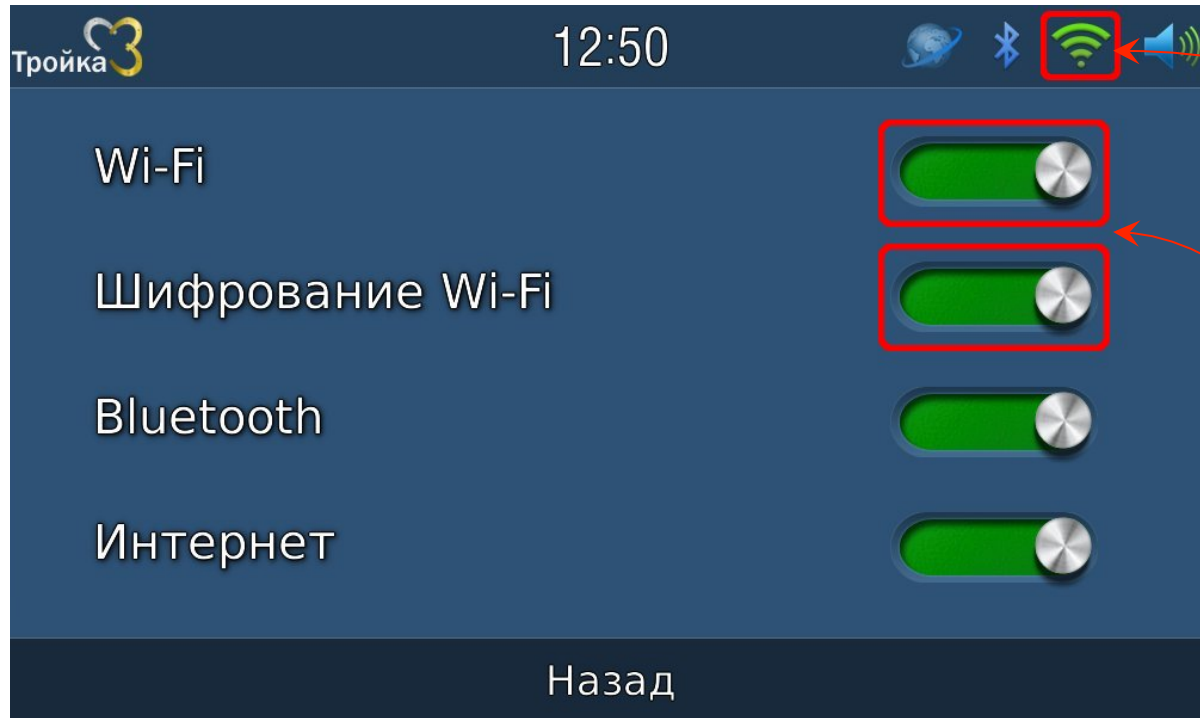
Олег Большаков, СВД Встраиваемые Системы



- ✓ Большая гибкость IPC
- ✓ Слабое связывание
- ✓ Сохранение данных между перезагрузками
- ✓ Удобочитаемые человеком объекты
- ✓ Простая отладка
- ✓ Независимость от языка программирования



# Пример использования PPS





pps [аргументы]

- b запуск в интерактивном режиме
- l способ загрузки объектов:
  - 1 – каталоги и объекты по требованию
  - 2 – объекты по требованию
  - 3 – каталоги и объекты при старте
- m точка монтирования
- p каталог сохранения объектов
- t период сохранения объектов (мс)
- v увеличение уровня подробности



# Свойства (атрибуты) объектов

```
attrname:encoding:value\n
```

attrname имя атрибута

encoding кодировка (необязательное поле)

value значение свойства

PPS никак не интерпретирует значение `encoding`

`::` обычный текст

`:c:` C-строка (`\t`, `\n`)

`:b64:` кодировка `base64`



## ➤ Общие принципы работы с PPS

Режим подписки

Функция `ppsparse()`

Практические примеры





```
# touch /pps/figure
```

```
...  
fd = open("/pps/figure", O_CREAT|O_RDWR);  
close(fd);  
...
```

```
# cat /pps/figure  
@figure
```



# Добавление/изменение свойств объекта

СВД Встраиваемые Системы

```
# echo "form::rectangle" >> /pps/figure  
# echo "color::red" >> /pps/figure
```

```
...  
char *form = "form::rectangle\n";  
char *color = "color::red\n";  
write(fd, form, sizeof(form));  
write(fd, color, sizeof(color));  
...
```

```
# cat /pps/figure  
@figure  
color::red  
form::rectangle
```





# Удаление свойства объекта

```
# echo "-color" >> /pps/figure
```

```
...  
char *color = "-color\n";  
write(fd, color, sizeof(color));  
...
```

```
# cat /pps/figure  
@figure  
form::rectangle
```



```
# rm /pps/figure
```

```
...  
unlink("/pps/figure");  
...
```

```
# cat /pps/figure  
/pps/figure: No such file or directory
```



## Общие принципы работы с PPS

### ➤ Режим подписки

Функция `ppsparse()`

Практические примеры





## 1. Функция обработки события `ionotify()`

```
while (ionotify (fd, _NOTIFY_ACTION_POLLARM,  
               _NOTIFY_COND_INPUT, &event) &  
       _NOTIFY_CONT_INPUT)  
{  
    if (read (fd, buf, sizeof (buf)) > 0)  
        /* обработка данных */  
        process (buf);  
}
```



## 2. Функция `select()`

```
int fd1, fd2, r;
fd_set readfds;
...
FD_ZERO(&readfds);
FD_SET(fd1, &readfds);
FD_SET(fd2, &readfds);
r = select(fd2 + 1, &readfds, NULL, NULL, NULL);
if (r > 0) {
    if (FD_ISSET(fd1, &readfds)) {
        /* обработка данных объекта 1*/
    }
    if (FD_ISSET(fd2, &readfds)) {
        /* обработка данных объекта 2*/
    }
}
```



## Перевод PPS в режим блокирующего чтения

### 1. Использование суффикса `?wait`

```
# cat /pps/figure?wait
```

### 2. Сброс флага `O_NONBLOCK`

```
flags = fcntl(fd, F_GETFL);  
flags &= ~O_NONBLOCK;  
fcntl(fd, F_SETFL, flags);
```

**ИЛИ**

```
int i = 0;  
ioctl(fd, FIONBIO, &i);
```



# Режим дельты

/pps/figure

```
write(fd, "form::circle", ...);  
write(fd, "color::red", ...);
```

```
read(fd, ...);
```

```
@figure  
color::red  
form::circle
```

/pps/figure?delta

```
read(fd, ...);
```

```
@figure  
form::circle
```

```
read(fd, ...);
```

```
@figure  
color::red
```



# Изменение нескольких атрибутов

СВД Встраиваемые Системы

```
char *buf =  
    "form::triangle\n  
    color::blue\n";  
write(fd, buf, ...);
```

```
read(fd, ...);
```

```
@figure  
form::triangle  
color::blue
```

```
char *buf1 =  
    "form::triangle\n";  
write(fd, buf1, ...);  
char *buf2 =  
    "color::blue\n";  
write(fd, buf2, ...);
```

```
read(fd, ...);
```

```
@figure  
form::triangle
```

```
read(fd, ...)
```

```
@figure  
color::blue
```





# Фильтр уведомлений

```
echo color::red >>  
echo size::small >>  
echo action::stop >>
```

```
cat ?wait,delta,f=size+color
```

```
@figure  
color::red  
size::small  
action::stop
```

```
echo color::blue >>
```

```
@figure  
color::blue
```

```
echo action::run >>  
echo size::large >>
```

```
@figure  
size::large
```



# Подписка на все объекты в директории

СВД Встраиваемые Системы

```
/pps/figures/circle  
    /triangle
```

```
fd = open("/pps/figures/.all");
```

```
@circle  
color::red  
size::medium
```

```
fd = open("/pps/figures/.all?delta");
```

```
@circle  
color::red  
@triangle  
color::green
```



1. Открыть объект `/pps/.notify` и получить идентификатор группы

```
char noid[16], buf[128];
int notify_fd, fd1, fd2;

notify_fd = open("/pps/.notify", O_RDONLY);
read(notify_fd, &noid[0], sizeof(noid));
```

2. Открыть объект с опцией `?notify=group:value`

<code>group</code>	Идентификатор группы
<code>value</code>	Произвольный идентификатор объекта



# Группы уведомлений

```
sprintf(buf, "/pps/circle?notify=%s:f1", noid);  
fd1 = open(buf, O_RDONLY);  
sprintf(buf, "/pps/triangle?notify=%s:f2", noid);  
fd2 = open(buf, O_RDONLY);  
  
while(read(notify_fd, &buf, sizeof(buf)) > 0) {  
    printf("Notify %s\n", buf);  
    /* обработка данных */  
}
```

```
Notify 8:f1  
Notify 8:f2  
Notify 8:f2
```

```
Notify -8:f1
```



## Общие принципы работы с PPS

### Режим подписки

- **Функция** `ppsparse()`

### Практические примеры





```
#include <sys/pps.h>
```

```
extern pps_status_t ppsparse( char **ppldata,  
    const char * const *objnames,  
    const char * const *attrnames,  
    pps_attrib_t *info,  
    int parse_flags );
```



# Структура `pps_attrib_t`

<code>char *obj_name</code>	Указатель на текущий объект
<code>int obj_index</code>	Индекс объекта в <code>objs</code>
<code>char *attr_name</code>	Указатель на текущее свойство
<code>int attr_index</code>	Индекс свойства в <code>attrs</code>
<code>char *encoding</code>	Указатель на кодировку
<code>char *value</code>	Указатель на значение свойства
<code>int flags</code>	Флаги разбираемой строки
<code>int options</code>	Параметры строки
<code>int option_mask</code>	Маска параметров строки
<code>int quality</code>	Не используется
<code>char *line</code>	Указатель на текущую строку



```
static const char *const objs[] = {  
    "circle",  
    "triangle",  
    NULL  
};
```

```
static const char *const attrs[] = {  
    "color",  
    "size",  
    NULL  
};
```





## Функция ppsparse ()

```
char ppsdata[1024];  
pps_status_t rc;  
pps_attrib_t info;  
  
memset(&info, 0, sizeof(info));  
n = read(fd, ppsdata, sizeof(ppsdata) - 1);  
ppsdata[n] = '\\n';  
while ((rc = ppsparse(&ppsdata, objs,  
                    attrs, &info, 0))) {  
    /* обработка данных */  
}
```



Общие принципы работы с PPS

Режим подписки

Функция `ppsparse()`

➤ Практические примеры





/pps/org/proj/dev/settings

/action

/status

org	Организация
proj	Проект или программный комплекс
dev	Менеджер устройства, датчика
settings	Настройки
action	Действие
status	Текущее состояние



```
/pps/kpda/troika/wifi/settings
```

```
ssid::troika  
password::1234  
dhcp::no  
ip::192.168.1.1
```

```
/pps/kpda/troika/wifi/action
```

```
start::yes
```

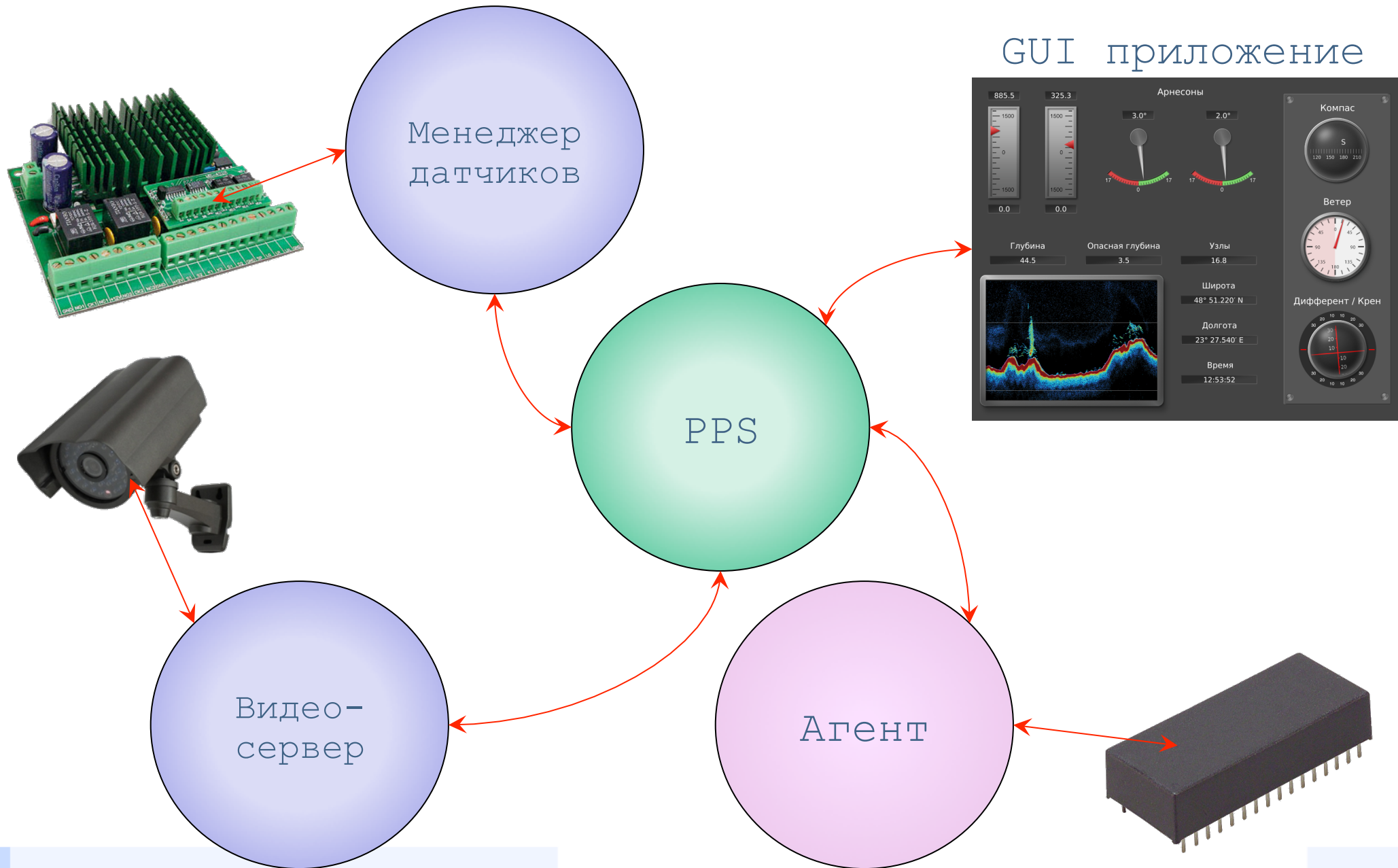
```
/pps/kpda/troika/wifi/status
```

```
connected::yes  
ip::192.168.1.1
```



# Собственный агент сохранения объектов

СВД Встраиваемые Системы





# Спасибо за внимание.

**СВД Встраиваемые Системы**

[www.kpda.ru](http://www.kpda.ru)    [forum.kpda.ru](http://forum.kpda.ru)

[sales@kpda.ru](mailto:sales@kpda.ru)    [support@kpda.ru](mailto:support@kpda.ru)

## **Центральный офис:**

196066 Санкт-Петербург

Московский проспект, 212А

тел.: (812)373-41-17

факс:(812)373-19-07

## **Технический офис:**

191014 Санкт-Петербург

ул.Госпитальная, д.3

тел./факс:(812)578-02-45