УДК 004

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ЛЕТАЮЩЕЙ РОБОТОТЕХНИЧЕСКОЙ ПЛАТФОРМЫ НА БАЗЕ ROBOT OPERATING SYSTEM

Ярошевич Е.В., Дубатовка В.В.
Белорусский национальный технический университет
Минск, Республика Беларусь

ROS (Robot Operating System) – это мета операционная система (ОС) с необходимым набором утилит (команд) для облегчения написания кода в робототехнике [1]. Она необходима для упрощения и гибкости написания умных алгоритмов, связанных с эксплуатацией робота, и не может существовать самостоятельно т.е. работает на базе другой ОС.

Сейчас в робототехнике активно изучается коллективная работа роботов, как пример – роевые полеты дронов. Таким образом, целью работы является разработка печатной платы и программного обеспечения с запуском роя дронов на базе ROS.

Для таких сложных систем настраивается общая сеть (сервер) и подключаются так называемые клиенты, тогда у каждого робота есть доступ к координатам других, поскольку все сложные операции обрабатываются на 1-м устройстве, но такая система не может работать вне зоны действия данного протокола (например Zigbee [2] до 100 м), где любая информация передается через каждого робота (клиента). Чтобы этого избежать используется другой тип связи (LoRa (Long Range) [3] до 20 км), работающий на дальних дистанциях. Используемый полетный контроллер (самый распространенный на STM32F427VIT6) не обладает такой возможностью, поэтому был взят за основу другой тип контроллеров, распаянных вместе: STM32WB55xx и STM32WL55JC. Такое решение позволяет еще отказаться от отдельного компьютера на корпусе робота (RaspberryPi), но значительно уменьшает его “умственные способности”.

Настройка сервера происходит с помощью сборки пакетов, установленных непосредственно на сервере и самих клиентов, включая *Chrony* [4] – альтернативный клиент и сервер протокола сетевого времени NTP. В качестве сервера может выступать любой компьютер, через который будут производится расчеты углов в двух системах координат: декартовой и полярной. Помимо этого нужно прописать независимые утилиты такие как: экстренная посадка (rosservice call /land), отслеживание углов (rostopic angle), индикация ошибки при полете (rosrun check selfcheck.py), запись их в check-файл и т.д.

В любой операционной системе при запуске различных программ они записываются в bag-файлы и занимают много места на жестком диске робота. При роевых полетах жесткий диск будет заполняться ненужными программами значительно быстро, чтобы это избежать нужно создать

скрипт bash, который будет удалять bag-файлы старше заданного периода (например: на диске осталось меньше 15% свободного места от общего объема), при этом настроить автоматическое периодическое выполнение данного скрипта, используя cron или systemd.

Выводы:

- вместо стандартного полетного контроллера в качестве роевых запусков выгодно использовать 2 контроллера с разным типом связи;
- для облегчения написание ПО на роботе использовалась мета-операционная система ROS;
- с целью безопасности и упрощения поиска информации на роботе создаются различного рода утилиты;
- для эффективной работы с памятью создаются специальные bash-скрипты.

1. ROS - Robot Operating System [Электронный ресурс]. – Электронные данные. – Режим доступа: https://www.ros.org/.
2. Zigbee — [Электронный ресурс]. – Электронные данные. – Режим доступа: https://ru.wikipedia.org/wiki/Zigbee.
3. LoRa (Long Range) — [Электронный ресурс]. – Электронные данные. – Режим доступа: https://habr.com/ru/companies/realtrac/articles/304312/.
4. Chrony — [Электронный ресурс]. – Электронные данные. – Режим доступа: https://redos.red-soft.ru/base/arm/arm-network/synchro-time/chrony/.