

РАЗРАБОТКА СРЕДСТВ ОБРАБОТКИ И ОТОБРАЖЕНИЯ ИНФОРМАЦИИ СИСТЕМЫ ДИАГНОСТИКИ ЭЛЕКТРОННЫХ МОДУЛЕЙ КОМБАЙНОВ ПО «ГОМСЕЛЬМАШ»

А. Д. Макаренко

*Учреждение образования «Гомельский государственный технический
университет имени П. О. Сухого», Беларусь*

Научный руководитель Э. М. Виноградов

Работа выполняется с целью разработки микропроцессорного модуля и написания программного обеспечения для него, которое будет выполнять прием данных из CAN сети, обрабатывать их и передавать на мобильное устройство. Также будет разработано программное обеспечение для мобильного устройства на основе системы Android, которое позволит в удобном для пользователя виде выводить сообщения, принимаемые от модуля. Данные нововведения помогут более оперативно подключаться к сети комбайна для проведения диагностики, а также позволят упростить процесс работы с протоколами данных, что, в свою очередь, позволит автоматизировать диагностику комбайнов.

Разработка состоит из двух частей: разработка электронного модуля и Android приложения. В первую часть входят выбор микроконтроллера и устройства передачи данных, написание для микроконтроллера программного обеспечения и разработка электронной схемы. В качестве микроконтроллера используется STM32F407VGT6. Для отладки использовалась плата STM32F4DISCOVERY, которая представлена на рис. 1. Связь с Android приложением будет осуществляться посредством Wi-Fi модуля.

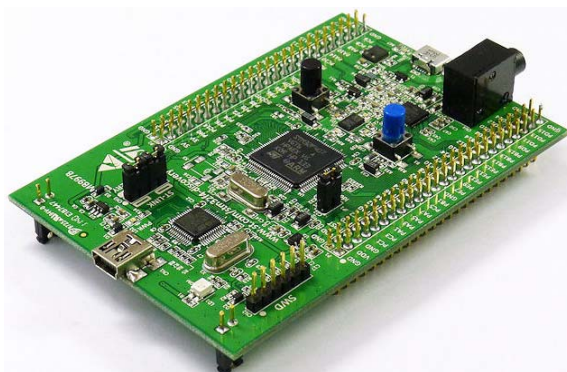


Рис. 1. Отладочная плата STM32F429 Discovery

Характеристики 32-битного ARM-микроконтроллера STM32F407VGT6 семейства Cortex-M4:

- I/O (макс.), 82 шт.;
- напряжение питания: 1,8–3,6 В;
- частота: до 168 МГц;
- память программ (Flash): 1 Мб;
- ОЗУ (RAM): 196 Кб;
- 4 Кбайт backup SRAM (аналог EEPROM);
- 16-битные таймеры;
- 2 32-битных таймера общего назначения;
- USB 2.0;

- 10/100 Ethernet MAC с отдельным DMA;
- 3x SPI (37,5 МБ/с
- 3x I2C с поддержкой SMBus/PMBus;
- 4x USART– 2x CAN (2.0B Active);
- SDIO (для SD-карт)
- аналоговый генератор случайных чисел;
- 82x GPIO (выводы I/O);
- RTC (Real-Time Clock);
- 3 12-битных АЦП, 2,4 млн выборок в секунду, 16 каналов;
- 2 12-битных ЦАП;
- возможность отладки по JTAG или SWD.

Для написания программного обеспечения для Android использовалась среда разработки Android studio с применением языка программирования Java.

Комбайн – это сложный механизм, состоящий из множества механических узлов, которые, в свою очередь, управляются электронными модулями. К каждому электронному модулю подключено множество датчиков (давления, оборотов, температуры, напряжения, датчики угла и пр.), модуль обрабатывает сигналы датчиков и соответствующим образом реагирует. Для взаимодействия модулей в комбайне используется CAN сеть (рис. 2).

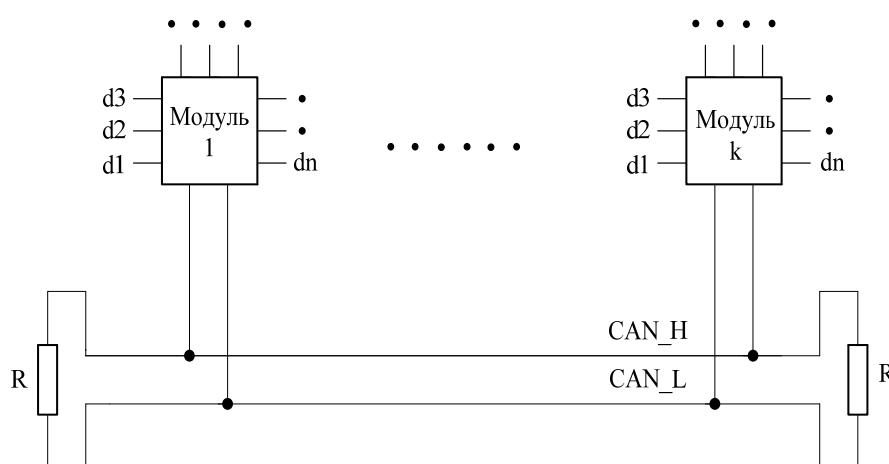


Рис. 2. Пример CAN сети комбайна

У CAN сети есть ряд преимуществ и недостатков.

К преимуществам CAN сети относятся:

- 1) возможность работы в режиме жесткого реального времени;
- 2) простота реализации и минимальные затраты на использование;
- 3) высокая устойчивость к помехам;
- 4) арбитраж доступа к сети без потерь пропускной способности;
- 5) надежный контроль ошибок передачи и приема;
- 6) широкий диапазон скоростей работы;
- 7) большое распространение технологии, наличие широкого ассортимента продуктов от различных поставщиков.

К недостаткам CAN сети относятся:

- 1) небольшое количество данных, которое можно передать в одном пакете (до 8 байт);
- 2) большой размер служебных данных в пакете (по отношению к полезным данным);

3) отсутствие единого общепринятого стандарта на протокол высокого уровня, однако же это и достоинство. Стандарт сети предоставляет широкие возможности для практически безошибочной передачи данных между узлами, оставляя разработчику возможность вложить в этот стандарт все, что туда сможет поместиться. В этом отношении CAN подобен простому электрическому проводу. Туда можно «затолкать» любой поток информации, который сможет выдержать пропускная способность шины.

В состав комбайна входят следующие модули:

- 1) питающий-измельчающий аппарат – отвечает за длину резки, обороты адаптера, измельчение растительной массы, вымолот зерна и пр.;
- 2) борт-информатор – собирает и обрабатывает информацию со всех датчиков комбайна;
- 3) трансмиссия – коробка передач, отвечает за движение комбайна;
- 4) автоматика – движение/поворот силопровода, системы копирования и пр.;
- 5) металлодетектор – уровень чувствительности детектора, срабатывание на металл и пр.;
- 6) CAN-панель – вывод в сеть статус всех кнопок;
- 7) терминал – вывод всей информации оператору в графическом виде.

Сообщения от модулей, датчиков выводятся в общую сеть. У каждого сообщения есть свой уникальный адрес. Все адреса по каждому модулю хранятся в протоколе обмена в бумажном виде. Это очень неудобно при диагностике (так как протокол обмена имеет большой объем), а также при написании ПО. Поиск нужного адреса занимает много времени, провоцирует ошибки и пр.

Для диагностики нам необходимо узнать состояние датчиков, для этого необходимо подключиться к CAN сети. Используют USB-CAN преобразователи. Специалист-наладчик с ноутбуком и данным преобразователем подключается к сети CAN комбайна, считывает информацию с датчиков, расшифровывает ее с помощью протокола обмена и анализирует, диагностируя возникшую проблему. Однако используемые сейчас устройства имеют ряд недостатков, затрудняющих работу, в ряде случаев приходится использовать одновременно несколько устройств, что также неудобно.

Наша разработка построена на основе микроконтроллера STM32. Она позволит решить ряд проблем:

- 1) дороговизна аппаратного и программного обеспечения диагностики узлов комбайна, что особенно важно ввиду внедрения на предприятиях политики импорта-замещения;
- 2) большие временные затраты для диагностики, обусловленные большим объемом и неудобством бумажной версии протокола передачи данных по CAN сети;
- 3) необходимость специально обученного персонала для обеспечения диагностики;
- 4) необходимость наличия нескольких модулей для решения поставленной задачи и, соответственно, необходимость обучения сразу нескольким продуктам программного обеспечения для USB-CAN преобразователей;
- 5) большая вероятность человеческой ошибки вследствие необходимости работы с большим объемом информации и недостаточной информативностью данных, предоставляемых ПО.