УДК 004.773

ПРИМЕНЕНИЕ CAN-ШИНЫ В СРЕДЕ «ИНТЕРНЕТ ВЕЩЕЙ»

Просвирин А.Д., Санкт-Петербургский политехнический университет Петра Великого, Санкт-Петербург, Россия

В данном докладе описан метод обеспечения взаимодействия нескольких устройств посредством проводной сети. Также показан физический и программный этапы реализации сети на основе CAN-шины.

Введение. В настоящее время активно развивается так называемое направление «Интернет вещей», включающее в себя работу с программными устройствами и реализацию схемотехнических решений. Одно из популярных направлений работы в этой области – это создание системы «Умный Дом». В ходе реализации данной системы передо мной возникла подзадача по организации сбора информации с множества датчиков и передача их на головное устройство.

Основная часть. В ходе исследований было выявлено, что более рационально использование проводных устройств, что в свою очередь было обусловлено условиями эксплуатации, а именно: необходимость снабжение устройств стабильным питанием, работа на больших расстояниях и предотвращение коллизий с существующими беспроводными сетями, такими как Wi-Fi, Bluetooth и т. д.

Все устройства системы удобнее всего подключить в общую шину, поэтому логично использовать готовый протокол общения. Однако, одновременно они общаться не могут, поэтому обычно используется понятие master-slave, где головное устройство определяет временное окно на общение с одним из устройств. В моем случае это не самое удачное решение, потому как датчики и исполнительные устройства имеют разный приоритет. Так, к примеру, датчики, отвечающие за безопасность, имеют более высокий приоритет, то есть требуют немедленной возможности передачи сигнала при срабатывании. А температурные наоборот, имеют более низкий. Таким образом, необходим иной подход для определения приоритета передачи сообщений. В итоге выбор пал на протокол CAN (Controller Area Network), который самостоятельно определяет приоритет сообщения и предполагает использование всего двух сигнальных проводов.

В протоколе описана структура сообщения, начинающегося с идентификатора устройства. Чем меньше числовое значение этого идентификатора, тем больший приоритет имеет передаваемое сообщение. То есть, если два устройства начнут отправку сообщения одновременно, то право передачи будет отдано устройству с меньшим значением идентификатора. Исходя из этого, каждому типу устройств был назначен свой диапазон идентификаторов, каждый из которых определяет важность соответствующего устройства.

По мимо архитектурного вопроса необходимо было решить задачу по физической реализации протокола. Здесь стоит уточнить, что лишь немногие промышленные контроллеры имеют встроенные CAN-трансиверы. С этой целью выгодно использовать трансиверы TJA 1050, выполненные в виде микросхемы поверхностного монтажа и имеющей сравнительно маленькие размеры. При этом в качестве контроллеров использовались STM32 F103C8T6 и Atmel ATmega328P. При этом, в ATmega328p нет встроенного формирователя кадров и поэтому было необходимо использовать дополнительную схему, а именно – MCP 2515. Для каждого из видов контроллеров были составлены принципиальные схемы подключения к шине, а затем разведены и созданы печатные платы. При этом, на всех платах были использованы удобные разъемы RJ-45 для упрощения процесса подключения и изменения мест подключения устройств.

Затем был реализован протокол общения более высокого уровня, обеспечивающий удобную передачу команд, получение данных и назначение идентификаторов устройствам динамически. Также стоит отметить, что CAN-шина предусматривает передачу не более 8 байт в одном сообщении, поэтому была внедрена поддержка передачи более длинных сообщений путем разбиения на фрагменты.

Выводы. Объединение устройств с помощью CAN-шины достаточно удобное и гибкое решение, не требующее больших затрат. При этом, помимо упомянутых вещей, протокол обеспечивает помехоустойчивое кодирование и сигнализирует о неполадках при приеме сигнала. Также благодаря дифференциальной передачи сигналов в кабеле возможна передача данных на расстояние больше 250 метров, а предусмотренное количество идентификаторов устройств (2^{29} штук $\approx$ 560 миллионов) покрывает большинство необходимых сценариев работы такой системы. А реализованный протокол взаимодействия нескольких устройств поверх протокола CAN обеспечил их надежное взаимодействие.