

Комуникационна мрежа за разпределени системи за автономни авто-модели

Николай Браянов*, Георги Атанасов*, Анна Стойнова* и Стоян Браянов**

* Технически Университет София, Факултет по Електроника, София 100, България, бул. „Кл. Охридски“ 8, бл. 1, e-mail: nbrayanov@ecad.tu-sofia.bg, ava@ecad.tu-sofia.bg

** Технически Университет София, Филиал Пловдив, Факултет по Машиностроене и уредостроене, Пловдив 4000, България, ул. „Цанко Дюстабанов“ 25

***Резюме.** Настоящата публикация разглежда въпроса за осигуряване на надеждна комуникация между разпределени системи за автономни авто модели. Разглежда се работата на стандартните комуникационни интерфейси в негостоприемна среда както и решенията, приложени в сходна индустрия – автомобилната. Публикацията предлага комбинация от стандартен интерфейс и протокол, които са възможно решение на проблема. Налага се извода, че е необходимо допълнително изследване на предложението протокол с цел оптимизирането му спрямо спецификата на интерфейса.*

***Communication network for distributed systems for autonomous car model.**(Nikolay Brayanov, Georgi Atanasov, Anna Stoynova and Stoyan Brayanov). The present paper discusses the question for communication in distributed systems, applied in autonomous car model. It starts with an analysis of standard communication interfaces' and their behavior in unfriendly environment. Decisions for common problems are found in interface and protocols in a relevant industry – automotive. In result, the paper proposes a combination of standard interface and protocol that should be capable to handle communication problems in this context. As conclusion, it claims that optimization of communication protocol is needed so that is could fit to the interface.*

Увод

С развитието на електрониката, възможностите за измерване на различни параметри на управляваните системи нарастват, водейки до допълнителна необходимост от ресурси – памет, периферия, процесорно време. В сферата на мобилните приложения се налагат и допълнителни ограничения във връзка с габаритите и консумацията на изделията, както и работата им в негостоприемна среда. В този контекст, решение предлага използването на разпределени системи[1].

Комуникационната мрежа в превозните средства е базирана на концепцията за разпределен контрол [2]. Това дава възможност за интегриране на сложни функционалности, и такива изискващи паралелност в извършване на процесите [3], [4], [5]. В допълнение информацията от сензорите може да бъде

многкратно използвана, както се демонстрира в [6]. Универсалността в комуникационният интерфейс дава възможност за промяна на конфигурацията, по функционален, географски или друг признак[7] и изграждане на функционалности, като контрол на обратната информация, дискретен контрол и диагностика и сервиз[8].

В настоящата публикация се разискват възможностите за създаване на комуникационна мрежа за отделени системи за нуждите на моделистиката. Необходимостта идва от широкото разпространение на такъв тип устройства(например квадрокоптери), които могат да застрашат живота и здравето на хората и имат нужда от сигурна комуникация[9], [10]. Конкретно ще разгледаме дизайна на радио управляем автомобилен модел и превръщането му в автономна система, състояща се от по-прости възли.

Състояние на проблема

Целта на публикацията е да предложи дизайн на интерфейс за разпределени системи, използвайки възможностите на стандартни, разпространени и евтини технологии. За да разгледаме проблема, първо ще бъдат анализирани интерфейсите и възможностите с които стандартно разполагат контролерите. След това ще бъде почерпен опит от индустрия, пряко свързана с превозните средства – автомобилната. В опит за локализиране на ключовите параметри за комуникацията между разпределени системи, ще бъдат анализирани използваните интерфейси, причините за възникването им, техническите им параметри.

UART (универсален асинхронен приемопредавател) възниква през 60-те като комуникационен протокол между терминално и комуникационно оборудване (DTE и DCE)[11]. Той е универсален от гледна точка на това, че времена, напрежения, контрол на предаването и проверка за грешки могат да бъдат конфигурирани. Максималната скорост може да достигне до 921600 бита за секунда, въпреки че практиката показва че скорости над 460800 са трудно достижими[12]. Стандартният UART протокол не дава възможност за провеждането на комуникация между много устройства чрез една линия.

SPI е синхронен сериен интерфейс за комуникация, разработен от Моторола[13]. Той е пълен-дуплекс, т.е. данните могат да бъдат изпращани едновременно в двете посоки. Подходящ е за обмен на данни между устройства и може да достигне скорости над 10MHz. Повече от две устройства, могат да бъдат свързани по:

- класически метод - всички устройства са свързани паралелно и активното устройство за комуникация се избира чрез допълнителен сигнал
- верижен метод - устройствата са свързани последователно като изхода на всяко е свързан с входа на следващото. Активират се едновременно, като по този начин се избягва нуждата от много изходи на главното устройство и става възможна комуникацията между подчинените устройства под контрола на главното. Недостатъкът е, че при по-голям брой подчинени устройства във веригата се губи скорост.

I2C шина е създадена от компанията Филипс в началото на 80-те за комуникация между компоненти, част от една и съща платка[14].

Интерфейсът е полу-дуплексен, синхронен, сериен и дава възможност за комуникация между едно или повече главни устройства и едно или повече подчинени устройства. Максимална скорост е 3400Kbps, но тя е достижима единствено в High-Speed конфигурация, която не всички устройства поддържат. Поради тази причина, при необходимост от скорости на 1Mbps се препоръчва използването на SPI.

През 1983 Бош започват разработката на CAN. Интерфейсът използва серийна мулти-мастър комуникация със скорост до 1Mbps, данните се предават в съобщения(рамки) които могат да пренасят между 0 и 8 байта данни[15]. Протокола има следните качества:

- приоритизиране на съобщения
- гарантирани максимални закъснения
- гъвкавост на конфигуриране
- изпращане до повече от един получател и времева синхронизация
- повече от един мастър на комуникацията
- откриване на грешки при предаване
- автоматично повторно предаване на сгрешените съобщения
- различаване на временни грешки и постоянно прекъсване на комуникацията с участниците

В края на 90-те, пет автомобилни гиганта създават т.нар. LIN консорциум с цел по-ниска цена на комуникацията[16]. Протоколът дава възможност за изграждане на йерархични мрежи[17]. Основните качества на LIN са:

- Концепция за един мастър и множество подчинени
- Ниска цена на контролера, базиран на UART
- Самостоятелна синхронизация, без кварц
- Детерминистично предаване на сигнала, което може да бъде предварително изчислено
- Евтина имплементация чрез един проводник
- Скорости до 20 kbit/s.
- Предсказуемо поведение
- Възможност за конфигуриране
- Многослоен протокол, съдържащ диагностика

Комуникационни шини, от които зависи сигурността се използват за време критична комуникация на модули от които зависи безопасността, като например завиване по жица или спиране по жица. Те трябва да са с висока надеждност и гарантирана скорост на предаване. Поради тази причина това е синхронна комуникация за която не е необходима арбитрация. В допълнение тези интерфейси

предлагат възможност за времева синхронизация. Такива, активирани по време, мултиплексирани с време делене протоколи са FlexRay, TTP/C и TTCAN.

FlexRay е протокол, поддържащ комуникация със скорости до 10 Mbit/s [18]. Време интервалите са предефинирани. Те съдържат статичен и динамичен сегмент, време интервал за символи и време, в което мрежата не се използва (Фиг. 1). Това дава възможност чрез един протокол да бъдат имплементирани всички нужди за комуникация. Метода за изпращане на данни е подобен на този при CAN, но отношението полезни данни/изпратени данни е 11% до 96%, в зависимост от размера на полезните данни в един пакет от 1 до 254 байта.



Фиг. 1. Структура на FlexRay протокол

В TTP/C комуникацията е организирана в TDMA (време делене за многократен достъп) сесии (Фиг. 2). TDMA секцията е разделена на слотове. Всеки участник в комуникационната система има отделен слот за изпращане и трябва да изпраща рамки през всяка сесия. Размера на рамките може да е между 2 и 240 байта, като всяка рамка в повечето случаи носи няколко съобщения. Цикълът на комуникацията е повтаряща се поредица от TDMA секции. В различните секции могат да бъдат пренасяни различни съобщения, но във всеки цикъл трябва да бъдат изпратени всички съобщения. Данните се защитават с 24 битова CRC. Последователността се съхранява в описанието на съобщенията (MEDL) в контролера на комуникацията.

За реализацията на протокола се изисква синхронизация на часовниците и графика на съобщенията. Всеки участник в комуникацията измерва разликата между доставеното съобщение и времето, в което то се очаква да пристигне [19], [20].



Фиг. 2. Структура на TTP/C протокол

TTCAN се базира на CAN интерфейса и премахва закъсненията за да осигури

детерминистична комуникация. При този протокол не е нужно всеки участник в комуникацията да е наясно с всички съобщения, които се предават. Контролера взема единствено нужната му информация - за синхронизация и за изпращане/получаване на съобщения [21], [22]. Специфичността на устройство, участващо в комуникационната TTCAN мрежа е запазена минимална, в сравнение с другите протоколи с време делене. Периодичността на TTCAN съобщенията се управлява чрез „Reference message“. Периода между две такива съобщения се нарича базов цикъл и се състои от няколко различни по размер времеинтервала. Той може да бъде използван както за периодични, така и за спонтанни съобщения. Периодичните съобщения се изпращат в т.нар. „exclusive“ прозорци (Фиг. 3). В случай че мрежата не е конструирана правилно и възникне конфликт, CAN протоколът ще го арбитражира. Пропуснатото съобщение няма да бъде изпратено отново, затова TTCAN дава възможност то да бъде многократно изпращано в един базов цикъл, гарантирайки че той ще бъде изпратен. Времевия прозорец за асинхронни съобщения се нарича „arbitration“. През този прозорец арбитражът на TTCAN решава кое съобщение ще бъде изпратено.

Поради практическата нужда да бъдат изпращани различни по цикличност съобщения TTCAN добавя системна матрица, състояща се от различни базови цикли.



Фиг. 3. Структура на TTCAN протокол

Идея

Тази публикация е проект за реализация на интерфейс за разпределени системи, използвайки възможностите на стандартни, евтини и разпространени технологии.

Анализа показва, че комуникацията между разпределени системи с приложение в индустрия с изисквания за безопасността има нужда от поне два типа комуникация:

- асинхронна комуникация
- синхронна комуникация с времеделене, гарантираща синхронизация във времето, в която всеки от участниците може да иницира комуникация

Чрез стандартните интерфейси, следва да бъдат

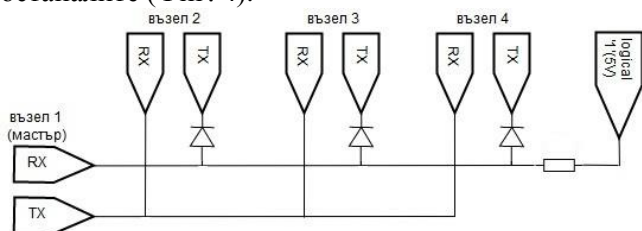
реализирани нужните свойства:

- Синхронна и асинхронна комуникация
- Адресиране
- Проверовъчна сума
- Синхронизация
- Работа в реално време
- Диагностика на комуникацията

Детайли

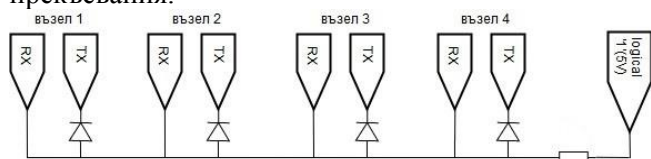
В тази част на публикацията ще бъде представен подхода, чрез който чрез стандартни интерфейси са реализирани двата необходими типа комуникация.

В [23], авторите предлагат метод за създаване на мрежа от устройства свързани чрез UART. Идеята е приложима чрез добавяне на специфична информация в използвания протокол за комуникация. Като недостатък на метода може да бъде посочена, че архитектурата не позволява свързване на всеки от участниците в мрежата към останалите (Фиг. 4).



Фиг. 4. UART архитектура за "multi-drop" комуникация с мастър

В [24] се предлага друг протокол. Въпреки че се използва RS485, идеите могат да бъдат реализирани и върху UART. В допълнение, възможно е и "multi-drop" свързване по подобие на RS485 (Фиг. 5), то обаче ще доведе до усложнение в схемата или генериране на допълнителни прекъсвания.



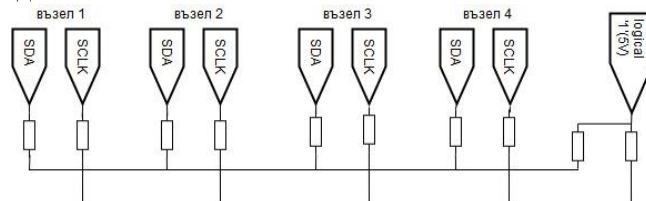
Фиг. 5. UART архитектура за "multi-drop" комуникация без мастър

При всяка от изброените конфигурации UART интерфейсът използва два проводника, независимо от броя на свързаните устройства. Максималната скорост от 460kb ще бъде редуцирана в случай на "multi-drop" свързване в следствие по-голям брой колизии.

След обзор на методите за създаване на мрежа на база SPI [25], [26], интерфейсът беше отхвърлен като възможност за имплементация на който и да е от двата типа комуникация. Причини за това са

големия брой на необходимите проводници за свързване, растящ линейно с броя на устройствата в мрежата. Допълнително затруднение е и липсата на гъвкавостта в архитектурата на мрежа и невъзможността лесно да се прибавят нови устройства. От друга страна верижното свързване води до сложен протокол, зависещ от броя на свързаните устройства, както и закъснения при предаване на информацията.

За I2C интерфейсът са необходими три проводника, независимо от броя на свързаните устройства, при това осигурява скорост до 3400Kbps. Свързване в мрежа и адресиране е част от стандартната му дефиниция. В допълнение се дава възможност за комуникация с много мастър устройства [27], [28], (Фиг. 6). Като недостатък може да се посочи, че интерфейсът е създаден за свързване между контролери на една платка или на малки разстояния, като нарастването на дължината на проводниците и съответно капацитета им ще доведе до грешки при комуникацията и намаление на скоростта на предаване.



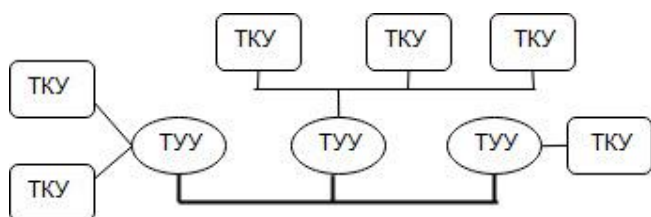
Фиг. 6. I2C "multi master" архитектура

UART и I2C интерфейсите са способни да се справят с реализацията на нужните протоколи. За да бъде избран най-правилния интерфейс, ще бъдат анализирани недостатъците им.

- I2C
 - По-голям брой проводници – реално с един повече. Този параметър не дава предимство.
 - По-ниска устойчивост на шум – I2C се тригерира по ниво и това води до по-ниската му устойчивост на шум. В предвид по-високата скорост на предаване, допълнителна проверка с цел откриване на грешки би редуцирала този проблем, като реалната скорост може да остане по-висока. Това следва да е тема на бъдещи изследвания
- UART
 - Не поддържа адресация – въпреки че интерфейсът не поддържа адресация, може да бъде осигурена поддръжка от страна на протокола.

- Ниска скорост – въпреки ниската скорост, при система с високи нива на шума UART може да се окаже интерфейса с по-висока скорост на предаване на полезна информация
- При реализация на мулти мастър, скоростта на предаване драстично намаля – този негатив може да се избегне, като не се използва мрежа с такава архитектура. Алтернативата, "Multidrop serial", не предлага връзка между подчинените устройства.

Опростена схема на мрежа от разпределени системи се състои от точки за управление на услугата(ТУУ) и точки за контрол на услугата(ТКУ) (Фиг. 7). Вторите се състоят от сензори и актуатори и са подчинени на първите. В този случай не е нужен обмен на информация между всички участници в мрежата. Връзката между точките за управление на услуги е симетрична – всяка от точките могат да отправят заявка към друга точка. В този случай използването на мулти мастър архитектура е задължително[29].



Фиг.7. Опростена схема на мрежа от разпределени системи

Възможно е и използването на централизирана мрежа фигура. Такъв тип мрежа е с повишено натоварване в централния контролер и отнема възможността за реализация на различни архитектури.

Устройства, съдържащи електромагнитни модули(като електромотор) и използвани в незащитена среда следва да бъдат устойчиви на шум. За съжаление използването на шумоустойчивия интерфейс(UART) би довело до лимитации във възможните архитектури. От друга страна използването на различни интерфейси в различните нива на комуникация е ненужно усложняване. В допълнение, I2C може да бъде използвано с по-висока скорост, поддържа адресиране и broadcast, което допълнително би олекотило необходимия протокол. Поради тези

причини за реализация на мрежата ще бъде използван I2C интерфейс.

Практично би било изграждането на единствен, универсален протокол с цел улеснение и повторно използване. В допълнение, изследването показва, че протоколите, мултиплексирани по време предлагат в някои от случаите по-добри резултати от асинхронните.

Поради по-ниската устойчивост на шум на I2C интерфейса, е необходима по-честа синхронизация. В допълнение, за да се намали риска от грешно предаване на пакет, което да доведе до липса на данни, съобщенията трябва да бъдат изпращани повече от веднъж. Тези две изисквания свързват необходимия протокол с решението, предоставено от TTCAN. Това решение е и по-динамично и гъвкаво, позволява по-лесна промяна в мрежата.

В следващо изследване ще бъде изследвана приложимостта на този TTCAN върху I2C. Протоколът TTCAN поддържа всички необходими функционалности. За прилагането му върху I2C той може да бъде оптимизиран, в предвид по-ниската скорост на предаване и по-ниската устойчивост на шум.

Заклучение

В публикацията бяха разгледани стандартните интерфейси за комуникация. След анализ на възможностите за свързване на множество разпределени системи се наложи извода, че подходящи интерфейси са UART и I2C.

Сравнението на методите за свързване на тези интерфейси в мрежа показва, че при реализация на мулти мастър комуникация чрез UART, скоростта на предаване намалява. Това, както и вградената възможност за адресация наложи използването на I2C.

I2C шина е създадена за комуникация между компоненти, част от една и съща платка и не е шумоустойчив. Протоколът, който следва да се приложи следва да отговаря на тази му особеност. В тази връзка, като най-близък до нуждите беше посочен TTCAN, заради възможността за честа синхронизация и многократно изпращане на едно и също съобщение.

Необходима е допълнителна работа за практическото изследване на тази комуникация и възможността TTCAN протоколът да бъде променен за нуждите на I2C, в предвид различията в скорост на предаване и шумоустойчивост.

Благодарности

Изследването е извършено с финансовата подкрепата на НИС при ТУ - София по проект №162ПД0022-03 на тема "Изследване на възможностите за моделно базирана разработка на вграден код съгласно изискванията в автомобилната индустрия"

Литература

[1] Coulouris G., Dollimore J., Kindberg T., Blair J.; "Distributed Systems. Concept and Design"; ISBN 10: 0-13-214301-1; ISBN 13: 978-0-13-214301-1; 1.3Trends in distributed systems

[2] He J., Li J., Zhang L. (2016) A Distribution-Based Model for Electric/Electronic Architectures of Automotive. In: Society of Automotive Engineers of China (eds) Proceedings of SAE-China Congress 2015: Selected Papers. Lecture Notes in Electrical Engineering, vol 364. Springer, Singapore

[3] Axelsson J., Fröberg J., Hansson H., Norström C., Sandström K., Villing B., "A Comparative Case Study of Distributed Network Architectures for Different Automotive Applications"; Technical Report, No 478; MRTC, 2003-01-28

[4] Ataide F., Santos M. M., "A Comparison of the Communication Impact in CAN and TTP/C networks when supporting Steer-by-Wire Systems"; <https://repositorio-aberto.up.pt/bitstream/10216/69375/2/69520.pdf>

[5] Grillinger P., Herout P.; "SIMULATION TOOL FOR FUNCTIONAL VERIFICATION OF TTP/C-BASED SYSTEMS"; <http://www.kiv.zcu.cz/~pgrillin/download/ess03.pdf>

[6] Brayonov N., Stoyanova A. "Multi sensor control based on fuzzy logic"

[7] Daniels B., "Safety and reliability of programmable electronic systems"; DOI:10.1007/978-94-009-4317-9

[8] Suwatthikul, J., "Fault detection and diagnosis for in-vehicle networks"; DOI: 10.5772/56395

[9] Tomasello F., Ducci M., "Safe integration of drones into airspace"; ISBN 978-92-846-0175-2

[10] Carr E., "Unmanned Aerial Vehicles: Examining the Safety, Security, Privacy and Regulatory Issues of Integration into U.S. Airspace"

[11] https://en.wikipedia.org/wiki/Universal_asynchronous_receiver/transmitter

[12] <http://blog.mbedded.ninja/electronics/communication-protocols/uart-protocol>

[13] https://en.wikipedia.org/wiki/Serial_Peripheral_Interface_Bus

[14] <https://www.i2c-bus.org/i2c-bus/>

[15] Bosch, "CAN Specification Version 2.0"

[16] https://en.wikipedia.org/wiki/Local_Interconnect_Network

[17] "LIN Specification Package Revision 2.1"; www.lin-subbus.org

[18] Kern A., "Ethernet and IP for Automotive E/E Architectures – Technology Analysis, Migration Concepts and Infrastructure"

[19] Elmenreich W., Ipp R., "Introduction to TTP/C and TTP/A"

[20] Bannatyne R., "Time Triggered Protocol: TTP/C"; <http://kurser.iha.dk/eit/tidrts/SupplementingCourseMaterial/TimeTriggeredProtocol-TTP-C.pdf>

[21] Führer T., Müller B., Dieterle W., Hartwich F., Hugel R., Walther M., "Time Triggered Communication on CAN (Time Triggered CAN- TTCAN)"; http://www.bosch-semiconductors.de/media/ubk_semiconductors/pdf_1/canliteratur/cia2000paper_1.pdf

[22] Fredriksson L., "TTCAN explained"; <http://www.kvaser.com/wp-content/uploads/2014/08/ttcan-explained.pdf>

[23] <http://blog.thegaragelab.com/uart-based-networking-for-microcontrollers/>

[24] Butler J., "Networking microcontrollers with serial ports"; http://m.eet.com/media/1159045/esc_1993_vol1_page25_butler_networking%20microcontrollers%20with%20serial%20ports.pdf

[25] <http://sfe.io/t16>

[26] <http://www.intersil.com/content/dam/Intersil/documents/an13/an1340.pdf>

[27] <http://www.byteparadigm.com/applications/introduction-to-i2c-and-spi-protocols/>

[28] <http://www.embedded.com/design/connectivity/4023975/Serial-Protocols-Compared>

[29] Han K., Cho J., "A Distributed Service Architecture for Networked Automotive E/E System"; <http://worldcomp-proceedings.com/proc/p2012/PDP4345.pdf>