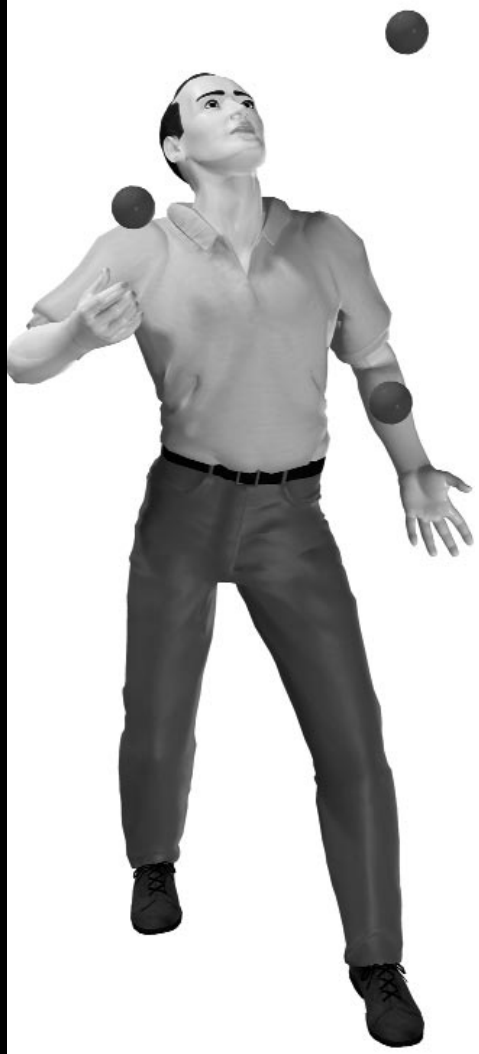
ROSS BANNATYNE

Time Triggered Protocol: TTP/C

Fault-tolerant, real-time performance is vital to the success of by-wire systems. Here is a solution that meets safety-critical requirements.



In this article I'll discuss an advanced serial communications protocol/system that has been developed for applications that require highly dependable or fault-tolerant operation. "TTP" stands for "Time Triggered Protocol," which describes the first fundamental property of this new communication protocol: access to the communication medium is granted to the computer nodes on a time-slot principle. The suffix "/C" indicates that this protocol conforms to so called class C applications, classified by the Society of Automotive Engineers, Inc. (SAE). Class C applications are all electronic systems of an automobile which are connected by a high-speed communication bus.

This article will discuss the beginnings of TTP/C, the requirement for such a solution, some explanation of why no other suitable solutions exist, the principles of the communications system, and details on the actual serial communications protocol message frames. Finally, I'll describe some typical applications of TTP/C.

TTP/C has been developed over the last 15 years from a research project at the Technical University of Vienna, directed by Professor Hermann Kopetz. The research project migrated into a European Community-funded scientific project by a consortium that included Daimler-Benz, Volvo, Ford, Bosch, and Magneti-Marelli. The standard is open and a significant amount of advanced development work has been undertaken using TTP/C. The references at the end of this article provide some more detailed writing on the subject matter.

Typical TTP/C applications would include automotive brake-by-wire or steer-by-wire systems, in which the systems must be "fail-operational," as the applications are safety-critical. "By-wire" systems transfer electrical signals down a wire instead of using a medium such as hydraulic fluid to transfer

Although several nontrivial challenges must be overcome before by-wire systems become the mainstream, many compelling reasons exist for the technology to be introduced.

muscular energy. A conventional antilock braking system (ABS) is considered "fail-silent"; if a fault in the electronic control system is detected, the control system is switched off, leaving the manual hydraulic back-up still operational. If no such hydraulic back-up is available (as in the case of a by-wire system), the system must continue to function in the event of a fault occurring.

The automotive industry has identified many good reasons to develop by-wire systems: reduction in parts count, removal of hydraulic system, improved maintenance, increased performance and functionality, increased passive safety by removal of mechanical linkages to passenger compartment, fuel economy, and so on. Although several nontrivial challenges must be overcome before by-wire systems become the mainstream, many compelling reasons exist for the technology to be introduced—so the challenges should be overcome relatively quickly. The TTP overcomes the challenge of fault-tolerant distributed embedded processing.

Additional interest is expected in several other applications that require a high degree of dependability, particularly in the fields of aeronautics, military, and medical systems.

Requirements for safety-critical systems

Closed-loop control-orientated safety critical applications usually execute a control cycle in a pre-defined time period. For example an electronic braking system usually executes a control loop every 10ms or thereabouts. In this cycle, several sensor inputs are evaluated, an algorithm in which output control variables are calculated, then signals are sent to actuators at the wheel. The loop is then repeated. In this type of system, regularity of information transfer is critical to maintain control of the system.

The distributed embedded control world already supports several serial communications systems such as CAN (Controller Area Network), SAE J1850-DLC, and SAE J1850-HBCC specifications. Three categories of communications systems are classified by the SAE: Class A is for low-speed networks typically used in vehicle body controls; Class B is for high-speed networks but with no safety-critical requirement; and Class C systems require certain stringent safety-critical requirements. The existing communications protocols do not meet Class C requirements, hence the Time Triggered Protocol was developed. The additional requirements for Class C are that they must be deterministic with small and bounded latencies, all-fault scenarios must be accounted for with a safe alternative operating mode, distributed clock synchronization (global time) must be supported, and the bus is guarded against "babbling idiot" nodes.

The unsuitability of the existing communications protocols stems from the fact that they are "event-triggered," in that a precise moment in time when a message will be received isn't specified. A communications protocol can only be predictable if worst-case transmission time and jitter are known at the time of the design and meet the requirements of the application. Real-time control applications are very sensitive to jitter, and so it is an important parameter for developing real-time distributed systems. The time delay between presenting a message to be transmitted at the senders interface and receiving the message at the receivers interface is known as the transmission time. Jitter is defined as the variability of this transmission time (the minimum transmission timesubtracted from the maximum transmission time). The maximum jitter depends on the longest message that is possible to transmit.

The type of communications protocol most suitable for ensuring regularity of information transfer is TDMA (Time Division Multiple Access). Using a TDMA scheme ensures that nonpredictable message delays aren't possible, as message transmissions are scheduled at the time of the design. Each electronic control unit is assigned a time slot in which it's given exclusive access to the bus to send messages. As every control unit has its own time slot, collisions are impossible. Also, as each transmission has the same priority for bus access, worst-case jitter can be easily calculated.

In time-triggered systems all actions are derived from the progression of a globally synchronized time base accessible to all nodes, whereas in event-triggered systems, all actions are derived from the occurrence of events. Table 1 outlines the main differences between TTP/C and the CAN protocol.

TDMA-based systems transmit *state* messages—for example, a switch being either on or off. State messages can be observed for a longer period of time than an event and are transmitted periodically. No new value overwrites an old value until the next TDMA round, and the state information isn't consumed when it's read. In a typical distributed embedded control system in which a number of sensors are sampled or polled periodically during the control cycle, state messages prove to be the most suitable message type for closed-loop control applications. Events, on the other hand, contain information that is valid

RUPERT ADLEY

The resource requirements for a time-triggered system are determined before run time so the system will behave predictably and be able to handle peak load situations deterministically.

at a particular point of time (until an overriding event occurs). An example of such an event would be when a push-button switch is pressed and released. Event messages are typically queued for consumption and consumed when read. These event messages are more efficient in systems with sporadic or rare occurences that require observation.

The resource requirements for a time-triggered system are determined before run time so the system will behave predictably and be able to handle peak load situations deterministically. Event-triggered systems are usually more inefficient than time-triggered systems when the system is operating at less than peak loading, because the system must be designed to handle worst-case conditions that may rarely occur.

The Time Triggered Protocol was developed to meet the requirements of deterministic communications, as well as to support the fail-operational or fault tolerant requirements that are critical in systems which would otherwise exhibit catastrophic behavior in the event of a fault.

Time Triggered Protocol principles

A TTP/C-based network is shown in Figure 1. Four host controllers are shown. These hosts could be electronic control units in a vehicle network such as braking, steering, suspension, and powertrain. Each of the four nodes are composed of a host, CNI (controller network interface), and the TTP/C controller. Two buses are present to support redundancy; if a fault develops on one bus, the alternate bus is available.

The host controller of each module runs the application software. The sending of messages is controlled by a scheduling table called the *message descriptor list*. This list contains the information that controls access to the bus in any particular time slot. The communications system and TTP/C controller will operate autonomously from the host software, using the message descriptor list which is stored in the CNI. Each node in the network is synchronized to a common global time. The CNI decouples the communication network from the host and provides a data-sharing interface between the host and the TTP/C controller. This is best physically implemented with dual port RAM that can be addressed by either the host or the TTP/C controller.

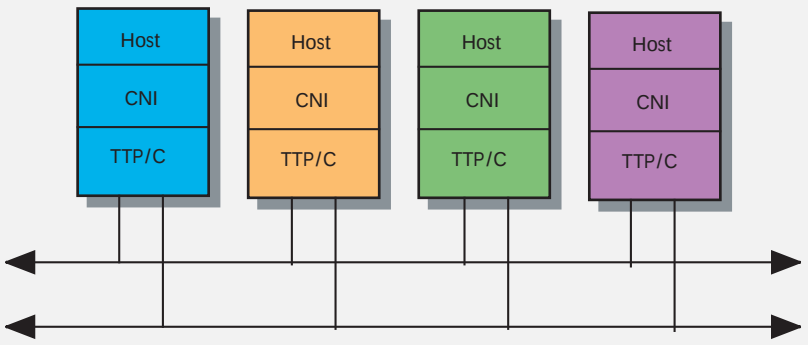
The third segment of the node is the actual TTP/C controller, which connects the node to the network. The TTP/C controller provides guaranteed transmission times with minimal latency jitter, fault-tolerant clock synchronization, and fast error detection. In support of fault tolerance, the TTP/C also supports replica determinism as well as a replicated communications channel.

The system is based on state message transmission; state messages can typically be observed over a longer period of time than an event message, which would change every time there is a new event, as opposed to periodically. State messages are well suited to closed-loop control type applications, in which inputs are usually required to be sampled once per control cycle. No queuing of messages occurs in the CNI, as a new version of the state message overwrites the old one every TDMA round. The Class B communications protocols, which we discussed previously, operate using event-based messaging.

TABLE 1 Differences between TTP/C and CAN

Function	TTP/C	CAN
Multi-Master Bus	✔	✔
Medium Access Control	TDMA	CSMA/CA
Flexible Bus Access	limited (modes)	4
Replicated Broadcast Buses	4	2 CAN controllers
Global Time Base	4	software
Membership Service	4	no
Bus Guardian	4	no
Replica Determinism	4	no
Composability	4	limited

FIGURE 1 Typical TTP/C-based system



The global time base is critical to the system, as the communications protocol depends on the knowledge of when every transmitted message is specified.

Replica determinism is implemented by duplicating nodes, so that if one node develops a fault, the signal from the node is replaced by a redundant node that broadcasts the same result in a different time slot. The main strategy for fault tolerance in the TTP/C system is fail-silence. A fail-silent architecture must deliver a correct output or no output at all. When no output is generated, the hardware has developed a fault. A number of error-detection strategies, both in hardware and software, must be employed in order to ensure fail-silence. The TTP/C controller uses watchdogs as well as a bus guardian which enables the bus driver only during the nodes transmission slot and disables it at all other times. This arrangement prevents the babbling idiot problem which can cause havoc in priority-based event triggered systems.

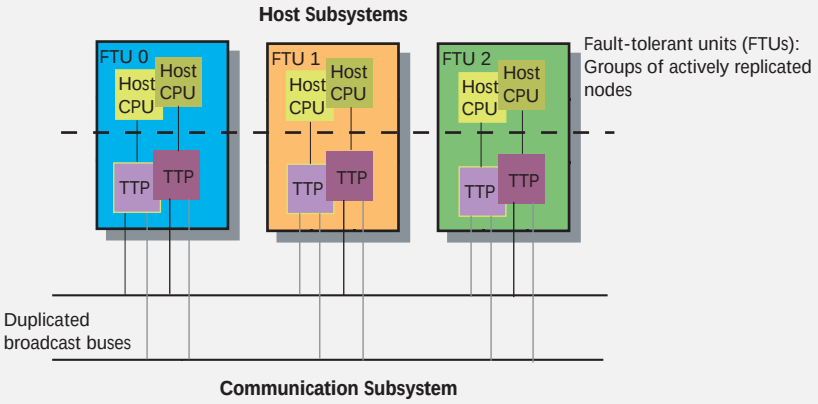
The second replica is grouped together in a cluster with the first replica, and is known as a fault-tolerant unit (FTU). A system consisting of three FTUs is shown in Figure 2.

The same message is sent on both channels by the first replica at a particular pre-defined timeslot and then re-sent by the second replica at a later TDMA slot. The second replica is completely physically separate from the first replica.

Error detection is achieved at the receiver side, since the arrival time of all messages in the system is known at design time. If a message isn't received at the expected time, this is regarded as a transmission error by all receivers. The global time base is thus critical to the system, as the communications protocol depends on the knowledge of when every transmitted message is specified. A synchronization algorithm is executed by each of the controllers in the network so that clock correction is possible and each node in the system will always have an identical notion of global time.

The concept of fail-silence in the communications system means that no voting system by several (minimum three) components is required, as in traditional fault-tolerant computer architectures. This concept is important, as typical voting schemes involving three or more CPUs are expensive to implement in lower-cost applications (such as the automotive industry). Each node focuses only on

FIGURE 2 Fault-tolerant units



detecting faults within its own entity; if a fault is detected, it switches itself off.

TTP/C message frame types

Two types of frame exist in TTP/C: initialization frames (I-frames) and normal frames (N-frames). These frames are indicated in Figure 3.

N-frames are transmitted periodically during normal operation of the system and contain application data. Three fields are present in the frame: a control field, data field, and cyclic redundancy check (CRC) field. Clock synchronization occurs just prior to the control field. It is inevitable that local timebases drift apart; therefore, a resynchronization strategy is implemented using the control field. The control field in the N-frame consists of an initialization bit which indicates that it is a normal frame. The mode bits are also contained in the control field, and indicate the operating mode of the system. The next field in the N-frame is the data field, which can contain up to 16 bytes of application data, depending on the operating mode. Finally, a CRC field consists of two bytes. The CRC is a slightly different calculation for the N- and I-frames, and makes it posssible for the receiver of the frame to detect errors in transmission. A normal frame is accepted only if the receiver and sender agree on the mode, global time, and node membership (which nodes are active or inactive, and which have a bit set to 1 or 0).

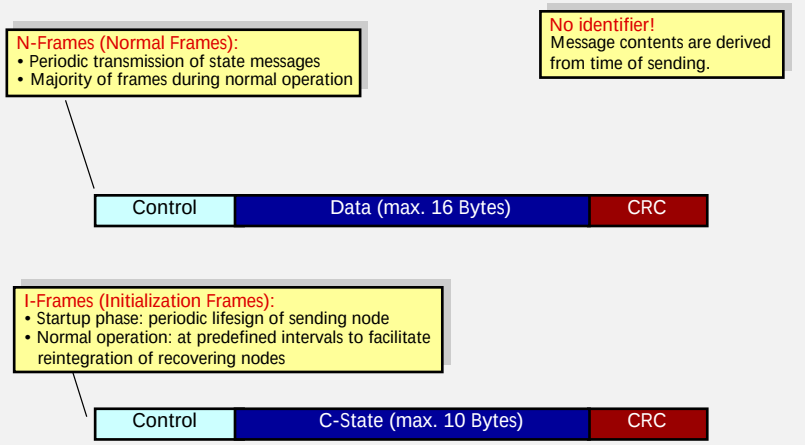
Neither the I-frame nor the N-frame have any identifier to indicate from which node they were transmitted. The message sender is implied from the time of sending.

I-frames are used for system initialization and contain data on the internal state of the TTP controller for its associated node in its data field. This information is known as the C-State (controller state). In TTP/C, all nodes are forced to implicitly agree on their C-states. The C-State contains information about the current operating mode, TDMA slot, global time, and the membership status. If the C-state of the sender isn't identical to the C-state of a receiver, the message will be disregarded by the receiver, due to the different CRC.

Continuous clock synchronization without any overhead to the frame length is acheived by executing an averaging algorithm periodically at each node. The node is given access to a global time base transmitted in the I- and N-frames. The receiver knows apriori the sending time of each frame from each node, so the disparity between the specified send time and the observed receive time indicates the time difference between sending and receiving nodes. Thus the appropriate distributed time bases can be tweaked to ensure uniform global time.

Because real-time distributed systems typically have different operating modes such as start-up, normal operating, emergency, and so on, TTP/C supports rapid mode changing. At any given time, the ensemble of nodes in

FIGURE 3 Frame types



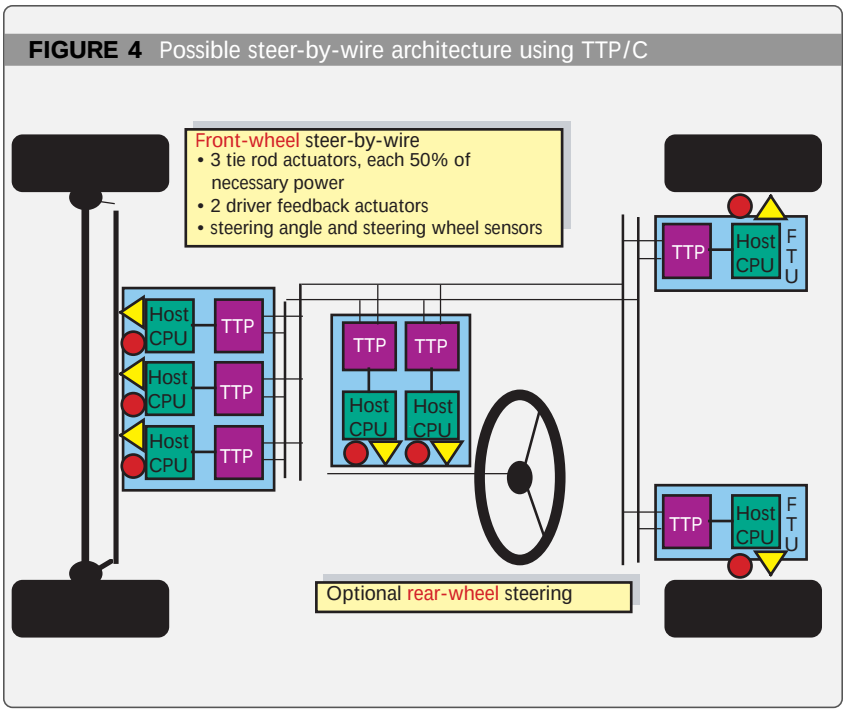


FIGURE 4 Possible steer-by-wire architecture using TTP/C

the system will be operating in a particular mode. A mode change is permitted when any node indicates that a mode change should occur using the mode bit in its control field.

Applications

A number of by-wire projects, mainly in the automotive industry, are being developed using the TTP/C protocol. TTP/C has been shown to be a suitable communications protocol for such applications because it satisfies the requirements of safety-critical communications systems by being deterministic, providing redundancy, and guarding against a fault which results in a single node monopolizing the bus. The architecture is also composable, which allows the behavior of an overall system to be predicted from the subsystem properties. Therefore,

A high level of redundancy is anticipated to be required on a steer-by-wire system, as no direct mechanical connection will exist between the driver and the wheels.

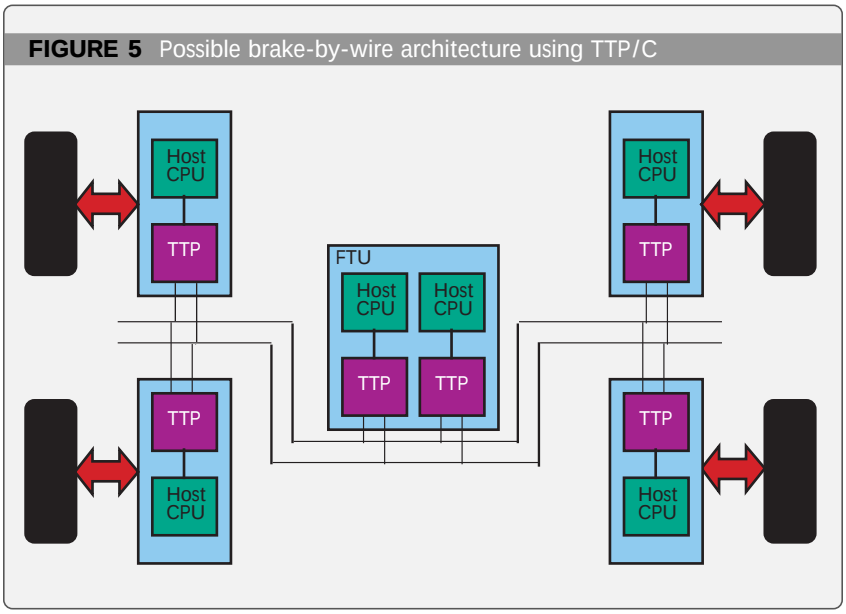
independent development, testing, validation, and certification of subsystems (nodes) may be accomplished.

A possible steer-by-wire architecture is shown in Figure 4. The TTP/C communications network is used to connect the steering actuators (motor controllers) at the front of the vehicle, the steering control unit mounted near the steering wheel, and the actuator units on the rear wheels (motors used for four-wheel steering). Three replicated nodes are present at the front actuator. These nodes receive information on intended steering angle from the main control unit and drive motors, which control the angle of the wheels. Feedback on angle and motor torque is returned to the main controller, and additional actuators are used to provide a comfortable level of steering wheel feedback to the driver. The main controller also consists of two nodes because steering is a safety-critical application.

A high level of redundancy is anticipated to be required on a steer-by-wire system, as no direct mechanical connection will exist between the driver and the wheels.

Brake-by-wire is another application that can be developed using the TTP/C communications protocol. An example of a brake-by-wire architecture is shown in Figure 5.

FIGURE 5 Possible brake-by-wire architecture using TTP/C



The system illustrates wheel nodes that control actuation of braking motors as well as providing the interface with the wheel speed sensors. It may be the case that a fault-tolerant unit isn't required at the wheel node, because a catastrophic event may not occur if one of the wheel nodes inhibits itself. It should be possible to brake the vehicle to rest safely with any three wheel nodes operational. The main central control unit consists of two replica controllers. The main control unit must be redundant because if a fault develops, a catastrophic situation could occur.

Automotive and beyond

TTP/C wasn't developed to compete with existing serial communications protocols; rather, it was developed to address new, advanced systems which will emerge and possess requirements that cannot be met by today's popular event-triggered Class B protocols. Both Class B and Class C systems will coexist in modern vehicles with a communications gateway that will allow them to share information. Although the initial applications are likely to be in the automotive market, many unrelated fields may require robust network operation that can continue to operate as normal if a node stops working correctly.

As with other popular serial communications protocols, the TTP/C controller module is planned to be integrated along with other functions on microcontrollers or as a standalone entity that can be designed into nodes in a given system. **esp**

Ross Bannatyne graduated from the University of Edinburgh, Scotland, with honors in Electrical Electronic Engineering and is currently a systems engineering manager for Motorola's Transportation Systems Group in Austin, TX.

Recommended Reading

Daimler-Benz AG, B.Hedenetz, and R. Belschner, "Brake-by-wire without Mechanical Backup by Using a TTP-Communication Network," *SAE Congress Conference Proceedings*, 1998.

Kopetz, H., "Fault Management in the Time Triggered Protocol (TTP)," *SAE Congress Conference Proceedings*, 1998.

Koptez, Hermann. *Real-Time Systems: Design Principles for Distributed Embedded Applications*. Dordrecht, The Netherlands: Kluwer Academic Publishers, 1997.

Kopetz, H., "Should Responsive Systems Be Event-Triggered or Time-Triggered?," *IEICE Transactions on Electronics*, November 1993.

Robert Bosch GmbH, E. Dilger, T. Fuhrer, B. Muller, and S. Poledna, "The X-By-Wire Concept: Time-Triggered Information Exchange and Fail Silence Support by New System Services," *SAE Congress Conference Proceedings*, 1998.