TTC

An outline for a CAN Global Clock

by
Lars-Berno Fredriksson, Kvaser AB and Jacob Österling, Ericsson Radio System AB
March 13, 2000

The goal for TTC is to establish a TDMA protocol on top of the basic CAN protocol. The base for TTC is a common global clock, supported on the CAN Controller level. It would be an advantage if the CAN Global Clock (CGC) could be synchronized to other TDMA system clocks, e.g., in radio communication systems like GSM and Bluetooth. Some of these require extreme accuracy (1 microsecond or better) and resolution. The CGC may not require to meet this without adding external components.

There are basically two concepts for a Global Clock:

a) Each node has a Native Clock running, giving a Local Time independent of the Global Clock. Each exported or imported time stamps is recalculated to the Global Time or Local Time respectively.

b) Each node has a Native Clock running. A Local Clock is created by hardware adjustments that is synchronized with the Global Clock.

As we intend to standardize a low-level Global Clock for CAN Controllers, the concept b) is preferred as it opens for simple and low-cost devices. A principle sketch of the concept is attached.

Assumptions:

1. There is one and only one Time Master (TM) at a time.
2. Simple CAN Controllers may not have TM capability but only Time Slave (TS) capability.
3. Specific messages are used for time synchronization.
4. Nodes not supporting CGC should be allowed in a TTC system.

A CGC supporting chip should have the following features:

1. A 32 bit up-counter with a comparator for setting a wraparound value.

   32 bit is chosen to make time messages easy to handle in the CAN data field. The programmable wraparound comparator is included to support different time bases as GSM 19 bit and Bluetooth 28 bit counters.

2. The other features in the ISO 11898 revision.
3. A prescaler, programmable at least from 4 to 64.
4. Ability to insert the counter value, byte-wise selectable, captured at SOF of transmission into the last byte(s) of the data field of the message.
5. Ability to time stamp received messages.
6. A flag indicating that the local clock is synchronized with the master clock within a programmable limit.

7. A flag for setting the CAN controller into master or slave mode (only if TM capable)
8. Programmable Line Delay Compensation
9. Programmable CGC Phase Adjustment Value (a 32 bit signed integer). PAV is a correction of the static frequency difference between the Time Master and Time Slave.
10. Programmable Time Adjustment Value. TAV is a correction of the offset of the Local Clock in the Time Slave and the Time Master.
11. Optional: CGC phase error signal output. Might be required for a VCO phase lock loop to increase accuracy.
12. Provisions for transmitting Time Synchronization Messages
13. A programmable (1 to AA_h) one byte physical Time Node Address, default AA_h

    This address belongs to the time system and may or may not be the same as the control system node address.
14. The EAN/UPC code of the product and the serial # of the item (40bit integer)

    The reason fro this is that TTC is intended for use in safety critical systems. Then there has to be a standardized and simple way to detect if any node has been replaced in the system since the last system consistency test was made.

CGC protocol

1. Time Synchronization Messages

TSM has the control bits set to one. There are eight types of TSM, six from the Time Master and two from the Time Slaves:

Master Messages

a) MTM, Master Time Message (Mandatory for TM and SLIO)

The MTM has DLC 4 and contains the captured counter value at SOF of the transmission

b) DMTM, Direct Master Time Message (Mandatory)

The DMTM has DLC 5. The first byte is the node address and the remaining bytes are the captured counter value at the SOF of the transmission

c) SSM1, Slave Setting Message 1 (Mandatory)

The SSM1 has DLC 7. The first byte is the node address, the second is set to 1, the third is the Prescaler Value to be set and the last four bytes are the Wraparound Value.

d) SSM2, Slave Setting Message 2 (Mandatory for SLIO, Optional for others)

The SSM2 has DLC 6. The first byte is the node address, the second is set to 2 and the remaining four bytes contains the Time Adjustment Value (TAV).

e) SSM3, Slave Setting Message 3 (Mandatory for SLIO, Optional for others)

The SSM3 has DLC 6. The first byte is the node address, the second is set to 3 and the remaining four bytes contains the Phase Adjustment Value (PAV).

f) Master Clock Request (Mandatory for SLIO, Optional for others)
The MCR has DLC 2. The first byte is the node address, the second is set to 4. When a slave receives this message, it shall transmit SOM.

Slave Messages

g) STM, Slave Time Message (Mandatory)
The STM is a response to the DMTM. It has DLC 8 and contains the captured counter value at the SOF of the received DMTM in the first four bytes and the captured counter value at SOF at transmission in the remaining bytes.

h) SOM, Slave Oscillator Message (Mandatory for SLIO, Optional for others)
The SOM has DLC 6. The first four bytes are the oscillator frequency expressed as a float32 (IEEE 754 single precision floating point) and the last two bytes are the crystal accuracy in ppm as a 16bit integer.

2. Phase adjustment

A phase adjustment of the Local Clock is done by altering the sequence of the prescaler. An adjustment is requested by one of the following states:

a) The TAV register is not zero.
While the TAV is not zero, one adjustment is carried out per prescaler wraparound period, and the TAV is decremented by one. An adjustment of the Local Clock of 4 prescaler clock periods for a prescaler counting to 8 will therefore take 32 prescaler clock periods.

b) The PAV counter has reached zero.
The PAV is decremented by one each prescaler wraparound period. When it reaches 0, a prescaler adjustment is made. The PAV is then reloaded to the PAV register value.

Phase adjustment can be executed by manipulating the two least significant bits of the prescaler in the following way:

When the local clock is faster than the master; 0,1,2,2,3

When the local clock is slower than the master; 0,1,3

This procedure is done at a rate given by a signed 32 bit integer (Phase Adjustment Value) that gives the increment for the procedure. A negative sign indicates that the local clock is faster than the master clock. The value 0 is interpreted as no adjustment should be done.

3. CGC adjustment sequence

A1) No MTM received yet

Step	Action
1.	Set the sync flag to 1, indicating unsync
2.	At reception of the first MTM, store the capture value at reception as t_{SR0}
3.	Register the CAN ID of the MTM and store the data as t_{MT0}
4.	Calculate $t_{MT0} - t_{SR0} = TAV$
5.	Write the TAV into the Time Adjustment Value Register
6.	Reception of a second MTM with the same CAN ID
7.	Calculate $t_{MT1} - t_{SR1} = TAV$
8.	Write the TAV into the Time Adjustment Value Register
9.	If TAV < [accuracy], set the sync flag to 0, indicating sync to TM
10.	Calculate $[(t_{MT1} - t_{MT0})/TAV]*prescaler = PAV$
11.	Write the PAV into the Phase Adjustment Value Register

A2) MTM received with the same CAN ID as earlier received

Step	Action
1.	At reception of the MTM, store the capture value at reception as t_{SRn}
2.	Calculate $t_{MTn} - t_{SRn} = TAV$
3.	If TAV < [accuracy], set the sync flag to 0 (if not already set) If TAV > [accuracy], set the sync flag to 1 (if not already set)
4.	Calculate $(t_{MTn} - t_{MTn-1})/[TAV+(t_{MTn} - t_{MTn-1})/PAV/prescaler]*prescaler = PAV$
5.	Write the PAV into the Phase Adjustment Value Register

A3)
MTM received with a different CAN ID
Follow from step 2 in alt. A1)

B) Line delay calculations.

In some time critical systems, the line delay may not be ignored. As any CAN message has a single source, each received CAN Id can be associated with a line delay time, unique for each node in the system. In most cases, the line delay can be assumed to be symmetric. If so, then it is enough if the receiving node is able to calculate the delay. However, as the basic concept of CAN is broadcasting without source and destination addresses, the scheme has to be left to the higher layer protocol.

Calculus:

Line delay d

$$d = [(t_{MRk} - t_{MTk}) - (t_{STn} - t_{SRn})]/2$$

C) SLIO setting

SLIO clocks have to be set by the TM by the following procedure:

1. If the SLIO timer is unknown, the TM transmits an MCR.

2. If the node address matches either 0 or the SLIO’s node address, it responds by transmitting SOM.

3. The TM calculates the prescaler value and transmits an SSM1

4. The SLIO sets the prescaler and responds by transmitting an STM as ACK

5. The TM transmits a DMTM

6. The SLIO responds by transmitting an STM

7. The TM calculates the required Time Adjustment Value and transmits an SSM2

8. The SLIO adds the Time Adjustment Value and responds by transmitting an STM

9. The TM transmits a DMTM and stores the transmission time

10. The SLIO with matching node address responds by an STM

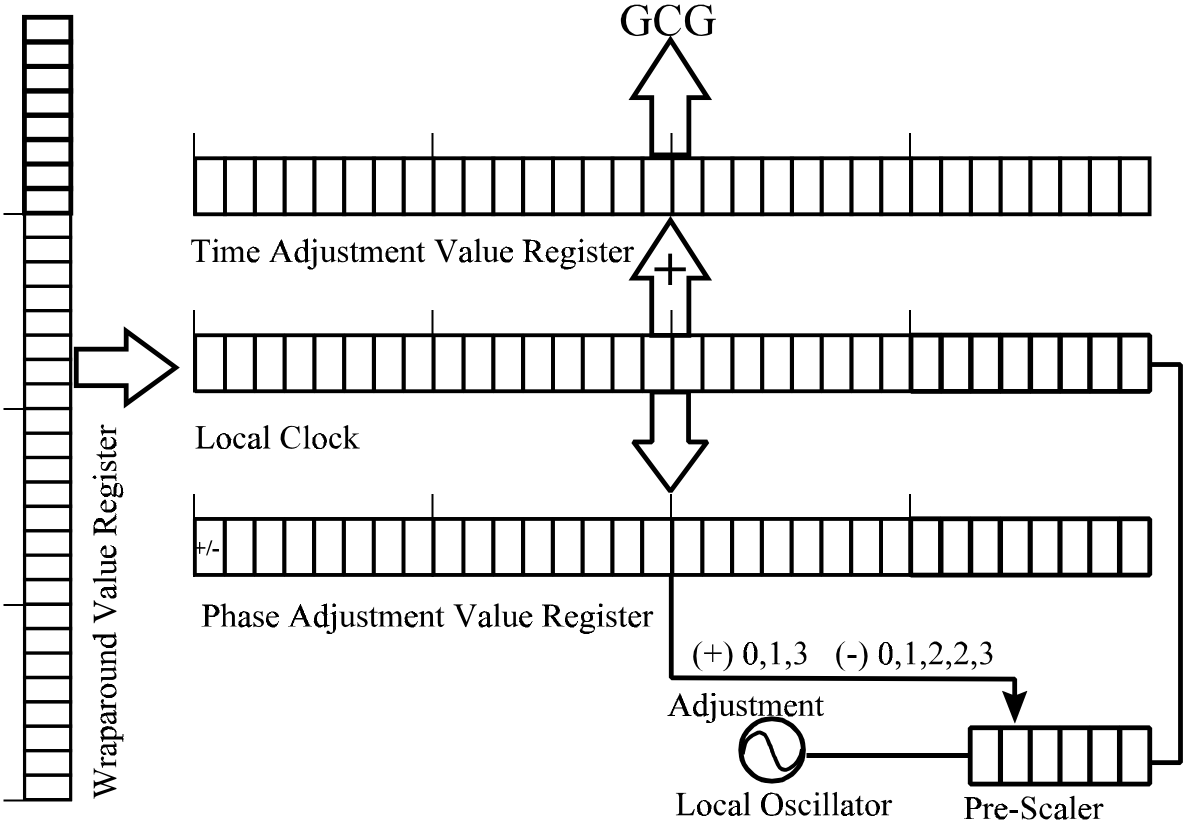
11. The TM checks the received values and judges whether the setting was successful or not.

All SLIOs known and set

12. The TM transmits an MTM

13. The SLIO checks whether the local clock is synchronized within limits or not. If the clock is not synchronized within the limit, the SLIO transmits an STM

14. When the TM receives an STM as above, it calculates a proper value for the phase error compensation and offset and transmits SSM2 and SSM3


GCG
Time Adjustment Value Register
+
Local Clock
+/-
Phase Adjustment Value Register
(+) 0,1,3 (-) 0,1,2,2,3
Adjustment
Local Oscillator
Pre-Scaler
Wraparound Value Register