STEP TECHNICA CO., LTD.

STEP TECHNICA

CUnet
Introduction Guide
(A Guide to the CUnet Protocol)

Preface

This manual gives an overview of the concepts of the CUnet, and defines various terms used in the system. Be sure to read this manual before referring to each of *"manuals"* for the various ICs comprising the CUnet.

● Target Readers

This manual is for:

• Those who first build a CUnet

• Those who first use StepTechnica's various ICs to build a CUnet

● Prerequisites

This manual assumes that you are familiar with:

• Network technology

• Semiconductor products (especially microcontrollers and memory)

[Caution]

• **To users with** *"CUnet User's Manual"* **released before March, 2001**

Some terms in this manual have been changed to conform to International Standards.

■ This manual has been prepared based on Standard English^{TM} meeting the requirements of the International Organization for Standardization (ISO) and the American National Standards Institute (ANSI). This English manual is consistent with the Japanese document **"STD-CUSTU-V1.1J"**.

• Standard English is a trademark of Win Corporation.

CONTENTS

Figures & Tables

1 What is CUnet?

[Information Sharing at the Collective Unconscious]

Collective Unconscious Network: CUnet

This is a new concept of communication
that allows multiple stations
connected to network to share memory data and
send/receive mail of dataset, and allows the stations
to use both simultaneously.

"CUnet" is a coined word that combines "Collective Unconscious" (psychological term) and "Network".
"CUnet" is a registered trademark of StepTechnica Co., Ltd.

"CUnet" is a generic term for the whole network system including individual equipment with the **"CUnet IC"**.

This guide describes the CUnet protocol that defines the basic operation and functions of the CUnet, and its terminology.

2 Outline of CUnet

The following description is an outline of CUnet.

    (1)   A CUnet-dedicated IC with **CUnet protocol** (**CUnet IC**) is provided to configure a CUnet.

    (2)   The **CUnet IC** has memory.

    (3)   The **CUnet IC** has a **bus interface** (**BUS I/F**) connecting user CPU.

    (4)   User CPU can read/write access to memory embedded in the **CUnet IC** via the **BUS I/F**.

    (5)   The **CUnet IC** has a **network interface** (**network I/F**).

    (6)   The **Network I/F** connects to the **network**.

    (7)   Multiple **CUnet ICs** are connected via the **network**.

    (8)   The **CUnet protocol** copies data, which a user CPU writes to memory of the **CUnet IC**, to memories of all **CUnet ICs** connected via the **network**.

    (9)   The **CUnet protocol** also copies data, which a user CPU connected to other **CUnet IC** writes to memory of a **CUnet IC**, to memories of all the **CUnet ICs** connected via the **network**.

    (10) Data copied to memory of **CUnet IC** can be read by all user CPUs accessing the **CUnet IC**.

    (11) The **CUnet protocol** helps all the **CUnet ICs** connected via the **network** to recognize mutually data in memory of each **CUnet IC**.

    (12) This allows the user to build a **network** in which the entire system operates as if it **shares a piece of memory data**, although the memory data exists in each **CUnet IC**.

    (13) The **CUnet protocol** can **share memory data**. In addition, it also allows sending the dataset between specific **CUnet ICs** using **mail sending/reception** function. The dataset can be sent to a mail receive buffer of the destination **CUnet IC** by instructing **mail sending/reception** after writing dataset to mail send buffer that the **CUnet IC** controls.

● **Keyword**	**CUnet protocol**
	CUnet IC
	Bus interface (BUS I/F)
	Network interface (network I/F)
	Network
	Share memory data
	Mail sending/reception

3 CUnet Configuration

The CUnet consists of multiple **user equipments** with a CUnet-dedicated IC (CUnet IC) and a network connecting these equipments. The CUnet protocol defines **half-duplex multi-drop connection** as communication mode. If half duplex mode allowing sending and receiving of serial communication signals between the CUnet ICs is used, the CUnet can be used with many types of **network cables**. The user can choose between coaxial cable, optical fiber cable, etc., according to the purpose and environment. In addition to the **network cable**, the CUnet network requires **differential driver/receiver components (TRX)** that receive/send serial communication signal (Fig. 3).

The CUnet protocol defines each CUnet IC connected via the network and equipment with the CUnet IC as a **"station"**.

Each **station** connected via the network must be assigned an individual **"Station Address"**, which is often abbreviated as **"SA"** in the CUnet protocols.

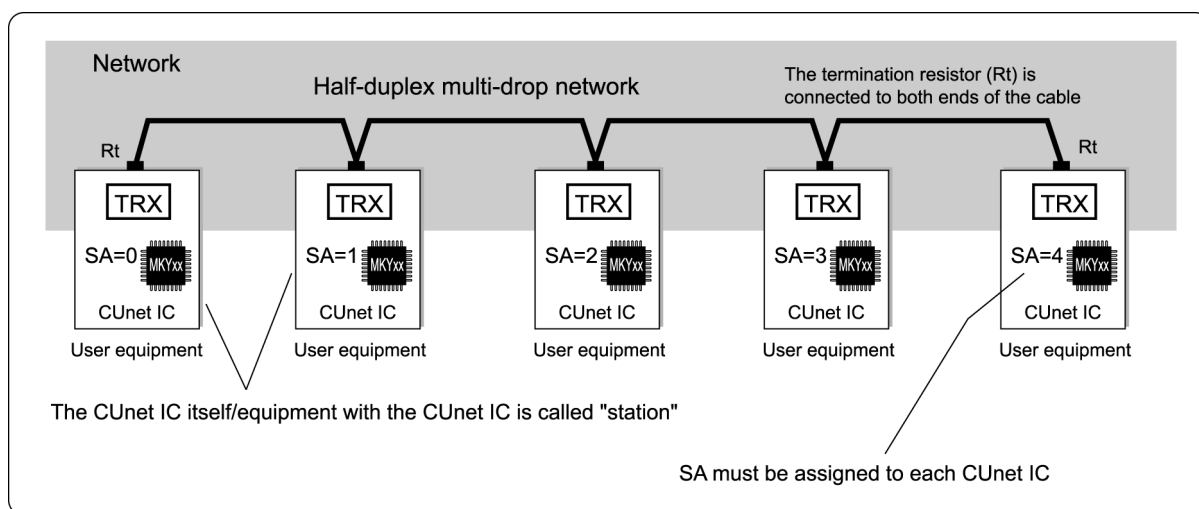Fig. 3   CUnet Configuration

● **Keyword**	**User equipment**
	Half-duplex multi-drop connection
	Network cable
	Differential driver/receiver components (TRX)
	Station
	Station Address (SA)

4 Cyclic Time-sharing

As mentioned in section **"2 Outline of CUnet"**, the CUnet protocol copies data, which a user CPU writes to memory of the CUnet IC, to memories of all CUnet ICs connected via the network. This "copying" is actually performed if the user CPU does not write data to the memory of the CUnet IC. In other words, the CUnet protocol copies the data in the memory of the CUnet IC connected via the network. For this "copying", the concept of cyclic time-sharing of the CUnet protocol is important.

This section describes the cyclic concept of cyclic time-sharing in items (1) to (4), and the concept of sharing memory data by time-sharing in items (5) and (6), and the concept of mail sending/reception in items (7) to (9).

  (1) Standard unit of time
  (2) Standard unit of time-sharing
  (3) Cycle configuration
  (4) Cycle time
  (5) Partitioning and owning of memory
  (6) Data sharing by copying data in memory block
  (7) Practical side of mail sending/reception
  (8) Control of mail sending/reception
  (9) Time required to send/receive mail

4.1 Standard Unit of Time

The standard unit of time in the CUnet protocol is T_{BPS}. In the CUnet protocol, a signal consists of two values with a given time of Hi and Low of RZ (Return to Zero, or Manchester Code) that function in pairs (Fig. 4.1).

In the CUnet protocol, the **"baud rate"** refers to the **BPS (Bit Per Second)** unit representing how many times of the smallest unit (one Hi level or one Lo level) making up one pair occur in 1 second. T_{BPS} means the time width of one pulse. For example, if the **baud rate** is 10 Mbps, the time T_{BPS} is 100 ns, indicating that 10,000,000 Hi-level and Lo-level pulse signal strings propagate over the network in 1 second.
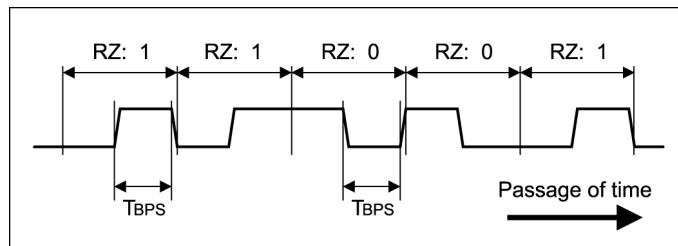
Fig. 4.1 RZ (Return to Zero), T_{BPS}

● **Keyword**	T_{BPS} (Standard unit of time)
	Baud rate
	BPS (Bit Per Second)

4.2　Standard Unit of Time-sharing

The CUnet protocol defines the standard unit of time-sharing as a **frame** and a required time of **one frame** as **"Frame Time"**.

The CUnet protocol also defines two types of positive-integer constants called **"frame length constant"** **(LOF: Length of Frame)** and **"Final Station"**.　The **Frame Time** can be calculated from the following equation 1 based on the **LOF**, **Final Station**, and T$_{\text{BPS}}$.　The **Final Station** is often called **"FS"** for short.

Equation 1: Frame Time = (LOF + FS + 1) $\times$ 2 $\times$ T$_{\text{BPS}}$ [sec]
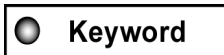
⚪ **Keyword**	**Frame**
	Frame Time
	Length of Frame: LOF
	Final Station: FS

4.3 Cycle Configuration

One **cycle** defined in the CUnet protocol consists of frame(s) corresponding to the number of FS(s) counted from "0", and frame(s) corresponding to the number of **Public Frame Count(s) (PFC)** defined in the CUnet protocol. To distinguish these frames, the latter is called **"Public Frame"**.

In the CUnet protocol, the **cycle** mentioned above is constantly repeated. This **cycle** can be compared to an ordinary clock. In the ordinary clock, a second hand goes around the face of a clock once in 60 seconds, so there are "1 to 59 seconds" at each position. The **cycle** in the CUnet protocol, on the other hand, a **Station Time (ST)** beginning with "0" is used to represent the time each time a frame, which is a standard unit of time-sharing, goes by. For example, a time after three frame times elapsed from "0" **ST** (ST = 0) is "3" **ST**, or **"ST = 3"**. Figure 4.3 shows a **cycle** configuration in which FS = 9, and **Public Frame Count (PFC)** = 2.
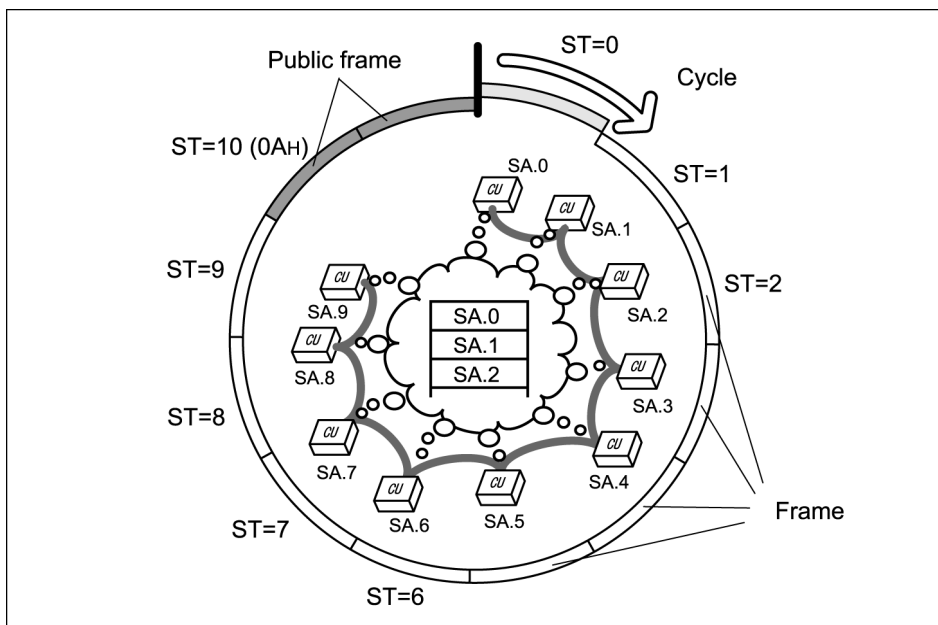
Fig. 4.3   Cycle Configuration

● **Keyword**	**Cycle** **Public Frame Count: PFC** **Public Frame** **Station Time: ST**

4.4 Cycle Time

In the CUnet protocol, a time required for one cycle is called **"Cycle Time"**. As can be seen from the cycle configuration in Figure 4.3, **Cycle Time** can be calculated from the following equation 2.

Equation 2: Cycle Time = Frame Time $\times$ (FS + PFC + 1) [sec]

◉ Keyword   Cycle Time

📖 **Reference**   The CUnet protocol (Version 1.00) defines PFC = 2, LOF = 151. When the final station in Figure 4.3 is 9 and TBPS is 12 Mbps ($\approx$83.3 n.sec), the frame time of the cycle can be calculated from $(151+9+1) \times 2 \times 83.3 \approx 26.83$ μs, resulting in a cycle time of 26.83 μs $\times$ 12 $\approx$ 322.0 μs.

4.5 Partitioning and Owning of Memory

Within the memory on the CUnet IC, memory area where memory data is shared is called **"Global Memory"**. The **Global Memory** is often called **"GM"** for short.

The **GM** is partitioned into some **Memory Blocks (MB)** counted sequentially from "0". In the CUnet protocol, the partitioned GM is corresponded to the station address (SA) by the **memory block** in accordance with the partitioning, just like the first MB = 0 corresponds to the SA = 0, and the next MB = 1 corresponds to the next SA = 1. The CUnet protocol calls this partitioning **"Owning"** (Fig. 4.5).

For example, in the CUnet protocol, "the station at SA = 0 **owns** the area at MB = 0 of GM," or "the **owned area** of the station at SA = 0 is GM area of MB = 0."

Fig. 4.5 Partitioning and Owning of Memory

◉ Keyword   Global Memory: GM
Memory Block: MB
Owning
Owned area

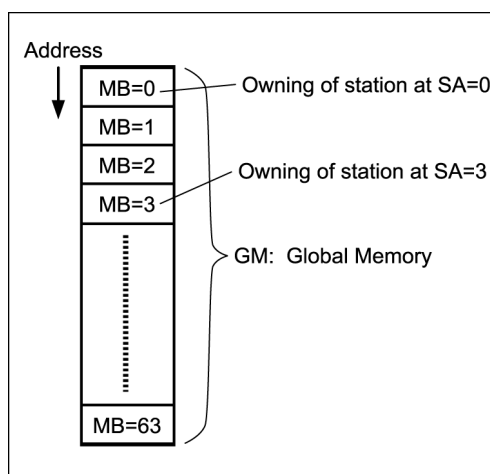## 4.6 Data Sharing by Copying Data in Memory Block

The CUnet IC, where the station address corresponding to the ST is set, packetizes the value of SA and the owned Memory Block data of Global Memory (GM) to send it (a packet) to the network. After the CUnet IC receives the data packet, it writes the received data to the MB of the GM of its self-station indicated by the SA value in the packet. This allows data in the MB to be copied from one CUnet IC to all the CUnet ICs connected via the network. The ST proceeds each time the frame goes by and thus data in the MB owned by each SA is copied between all the stations. When one cycle is completed, copying of data between all the stations is also completed, allowing each station connected via the network to share the memory data.

From Figure 4.6, when ST = 0, data at MB = 0 of GM is transmitted from the station at SA = 0 to the network. After the CUnet IC receives the data, the CUnet IC writes the received data to the area at MB = 0 of GM. In this way, the data at the owned area of GM (MB = 0) in the CUnet IC at SA = 0 is copied to the areas at MB = 0 of all the CUnet ICs. Similarly, when ST = 1, data at MB = 0 of GM is transmitted from the station at SA = 1 to the network and is copied to all the areas at MB = 1 on the CUnet. When ST = 2, the data from SA = 2 is similarly sent to be copied. When ST = 3, the data from SA = 3 is similarly sent to be copied. When ST = 63, the data from SA = 63 is also sent to be copied. The sequence of copying is executed in accordance with the passage of the frame. In this way, when one cycle is completed, GM data in all the CUnet ICs connected via the network becomes the same.

The CUnet cycle continues automatically, allowing the user program controlling the user CPU to share the data through the memory-like GM.
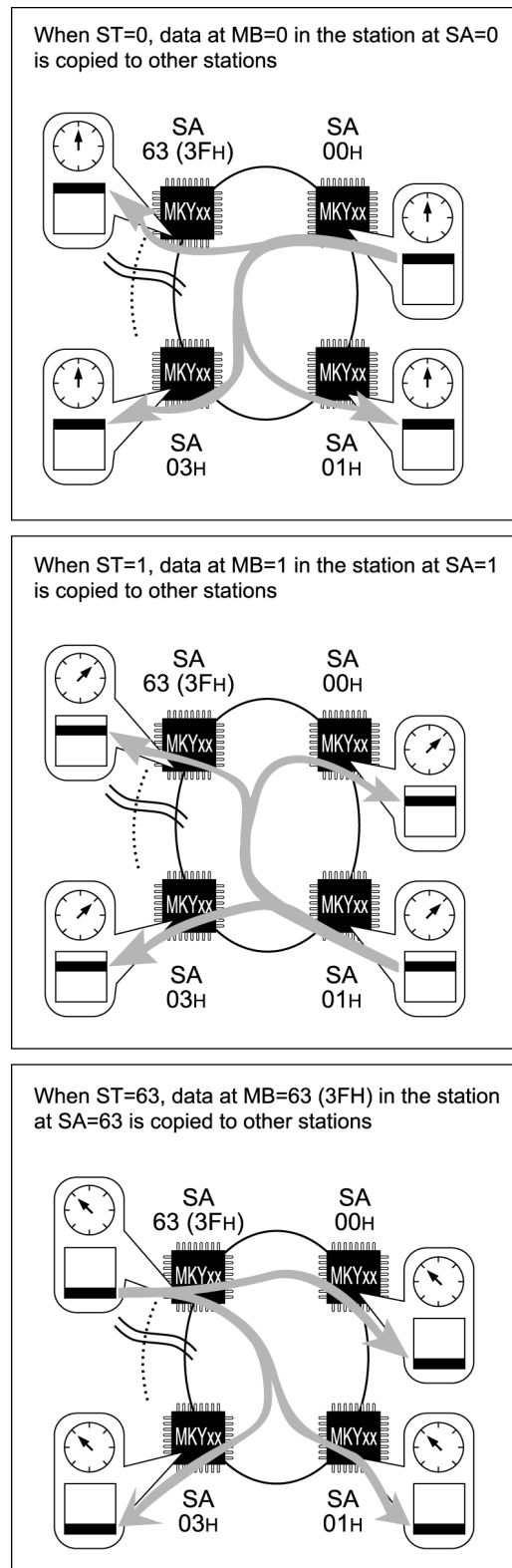
Fig. 4.6 Principle of Data Sharing

4.7 Practical Side of Mail Sending/Reception

The CUnet protocol can share memory data. In addition, it also allows sending/receiving the dataset between specific CUnet ICs using the **mail sending/reception** function.

In the CUnet protocol, the dataset is sent/received as follows: The dataset in the **mail send buffer** in the CUnet IC is divided into pieces of a size that the CUnet protocol defines as the dividing size. Then the divided dataset is packetized into packets with source and destination station addresses attached, and the packet is sent for each cycle. The destination CUnet IC receives the packets and stores them sequentially the **mail receive buffer**. When sending of all the divided dataset is complete, the **mail sending/reception** is complete.

In the CUnet protocol, the mail is sent using the public frame, which does not interfere with sharing memory data (Fig. 4.7).
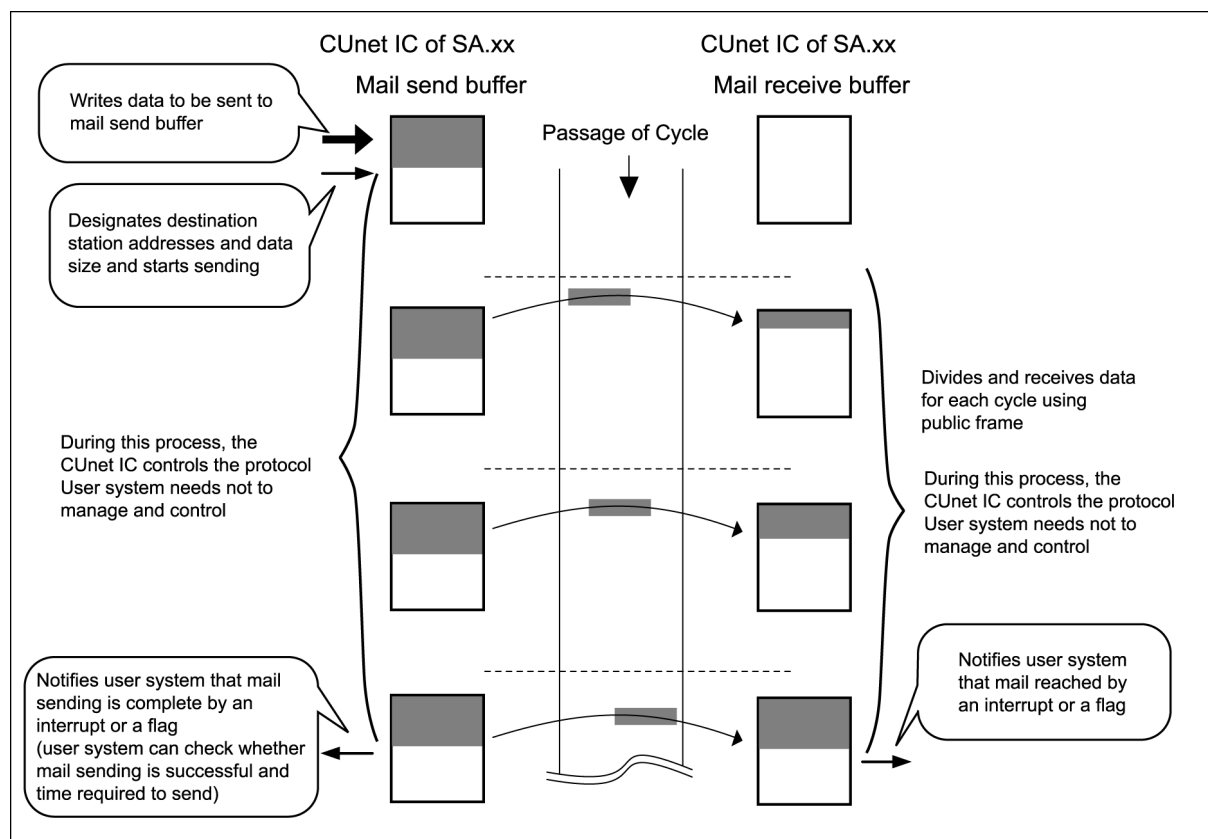
Fig. 4.7   Practical Side of Mail Sending/Reception

⬤ **Keyword**	**Mail Sending/Reception**
	Mail send buffer
	Mail receive buffer

4.8 Management of Mail Sending/Reception

The CUnet protocol defines that the CUnet IC must have all the management required to send/receive the mail as follows.

    (1) To manage division and concatenation of dataset

    (2) To manage use rights for public frame

    (3) To manage destination station, i.e. whether or not the station exists

    (4) To manage how much available memory space remains in destination station

    (5) To manage whether or not the sending/reception of dataset by packets is successful, retry, and the number of times of retry.

    (6) To manage time required to send/receive mail (i.e. setting and management of timeout, and report of the time required to transfer mail)

These help the user system using the CUnet use the mail sending/reception function simply (Fig. 4.7).

In the CUnet protocol, even if multiple stations start mail sending/reception around the same time, the CUnet IC can send/receive mail corresponding to the public frame counts in a cycle simultaneously. If the number of mail exceeds that of public frame counts, the CUnet IC starts sending from the station address (SA) with the minimum SA value.

Because this priority rotates, if the station with the minimum SA value continues to send mail, the station with a greater SA value needs not wait for sending mail endlessly.

In addition, if a station starts to send mail to an other station that is already sending mail, the mail from other station must be waited until the already sent mail is terminated.

4.9 Time Required to Send/Receive Mail

In the CUnet protocol, the time required to send/receive mail is calculated from the following equation 3. However, the following three items are not included, so the time obtained from the equation 3 should only be used as a guide when building a user system.

    (1) Wait time for obtaining use rights for public frames when stations exceeding the public frame counts start to send mail simultaneously

    (2) Retry time due to (5) in *"4.8 Management of Mail Sending/Reception"* in case when the packet to be sent at the timing of public frames, is discarded based on a test described later in *"6.1 Test at Packet Reception"*, because the packet is damaged by external noise or improper environment.

    (3) Wait time that station must wait for until the already mail is terminated, when it starts to send mail between other station that is already sending/receiving mail

Equation 3:   (((Byte count of dataset) + 7 ÷ 8) + 3) × cycle time [sec]

In a partial solution (underlined part), digits after the decimal point are dropped.

Reference For example, in a system (FS = 3) that is operated by four stations with a baud rate of 12 Mbps, the time required for sending/receiving of 250 bytes of mail can be calculated as follows:

$(((250 + 7) \div 8) + 3) \times 109 \ \mu s = 35 \times 109 \ \mu s = 3.82 \ ms$

5 Synchronization of CUnet IC

To implement "cyclic time-sharing" described in section 4, all the CUnet ICs connected via the network should synchronized.

The CUnet protocol defines four phases: **START phase**, **CALL phase**, **RUN phase**, **BREAK phase** to start the station, enter into network, leave network, and maintain cycle synchronization. These four phases assure discipline and synchronization of multiple CUnet ICs connected via the network. This section describes discipline that maintains synchronization in each phase. Figure 5 shows the phase transition.
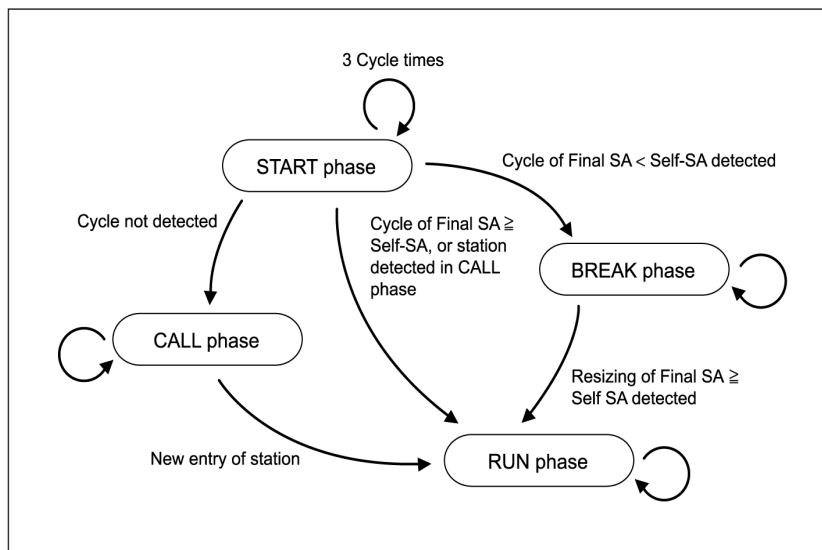
Fig. 5   Phase Transition in CUnet

⬤ **Keyword**	**START phase**
	CALL phase
	RUN phase
	BREAK phase

5.1 START phase

The CUnet protocol only receives for about three cycles right after startup in order to recognize the operating status of the CUnet protocol. This period is called "START phase".

If the CUnet protocol cannot confirm reception of a packet from other stations during the START phase, the entire network recognizes it as a new startup to transit to the CALL phase. If the CUnet protocol can confirm reception of a packet from other stations during the START phase, the network transits to the RUN phase.

5.2 CALL phase

The CALL phase means the stage in which the CUnet is waiting to be connected. During the CALL phase, any station repeatedly sends call packets an interval of two to three cycles so that the station in the START or CALL phase can be recognized. The CALL phase transits to the RUN phase after the self-station recognize the reception of the call packet from other stations.

5.3 RUN phase

The RUN phase means the stage in which the CUnet operates normally. The RUN phase continues until when any station cannot receive all the packets from other stations or when the user system instructs the network to stop.

5.4 BREAK phase

The BREAK phase is an exception state for the CUnet. During the START phase, when other stations are already in the RUN phase and the SA with values greater than those of the Final Station (FS) that defines a cycle is set, the START phase transits to the BREAK phase (because the station cannot enter the cycle).
The station in the BREAK phase sends a break packet at the timing of public frames in order to prevent interference with the RUN phase to demonstrate to other stations that there is a station in the BREAK phase.
The BREAK phase transits to the RUN phase when Final Station ≥ Station Address.

📖 **Reference** For the relationship of the BREAK phase and the user system, refer to *"7.1 Resizing"*.

5.5 Calibration of time

The CUnet protocol defines calibration of the cycle timing of all the CUnet ICs connected via the network with reference to the source Station Address attached to a packet received in the following four cases. This **calibration** allows cycle **synchronization** of all the CUnet ICs connected via the network to always be maintained.

(1) When any station receives a packet from an other station while it is in the START phase and then transits to the RUN phase.

(2) When any station receives a packet from an other station while it is in the CALL phase and then transits to the RUN phase.

(3) When any station receives a packet with a minimum station address while it is in the RUN phase.

(4) When any station receives a packet with a minimum station address while it is in the BREAK phase.

⚪ Keyword	**Synchronization**
	Calibration

6 Data Quality Assurance

The CUnet protocol assures data to be copied and transferred when sharing memory data and transferring the mail (dataset). This section describes the quality assurance defined in the CUnet protocol.

6.1 Tests at Packet Reception

The CUnet protocol requires that the CUnet IC **perform the following three tests concurrently** when receiving packets. This prevents garbage data that tends to occur in other communications.

(1) **Pattern format test:** This test examines, at the completion of packet reception, whether the structure system of a packet starting with a specified start pattern meets the CUnet protocol.

(2) **CRC-16 test:** Generally, in an Ethernet LAN and bulk communications, one "CRC-12" block check code (BCC) is used for an array of thousands to tens of thousands of bits. In contrast, the CUnet protocol uses CRC-16 BCC, which provides better performance than the CRC-12, for one packet (an array of hundreds to thousands of bits). This BCC is examined at the completion of packet reception.

(3) **Validity test of RZ signal format (RZ test):** The packet of serial communication signals consists of a pulse array for RZ (Return to Zero or "Manchester code") described in **"4.1 Standard Unit of Time"**. The CUnet protocol examines received packets in bit units to check that the RZ format is maintained.

⬤ **Keyword**	**Perform three tests concurrently** **Pattern format test** **CRC-16 test** **Validity test of RZ signal format (RZ test)**

📖 **Reference**	Generally, although the "test" and "correction" concepts are used in serial communications, "correction" is completely different from "test". "Correction" restores damaged packets. It is effective if data like consecutive voice data can have a certain error level (garbage data). However, it is not suitable for CUnet. The CUnet protocol only performs "tests" and discards all received packets failing the tests.

6.2 Handshake for Individual Stations

If a common and popular communication mode (i.e., the broadcasts on the Ethernet LAN) is used, data sent from one station can be received by other multiple stations. In this case, however, the user system cannot be assured that the multiple stations received the data correctly.

If the user system wants such an assurance in a common and standard communication mode, the user system must get an answer (Answerback) indicating that the data was received after being sent. To get the answerback individually, multiple stations must perform a **peer-to-peer handshake** communication individually. If **communication is multiple stations-to-multiple stations**, the **handshake** communication must be performed ((station count-1) × station count) times.

The CUnet protocol, on the other hand, assures a **multiple stations-to-multiple stations handshake communication**.

⬤ **Keyword**	**Peer-to-Peer**
	Handshake
	Multiple stations-to-multiple stations handshake communication

6.2.1 Receive Status and Link Status

The CUnet protocol requires that the CUnet IC have two types of status called **"Receive Status"** and **"Link Status"**. These statuses exist individually for final stations.

The **receive status** indicates that receiving of data from other stations is successful. The **link status** indicates that the previous data sent from the self-station reached other stations correctly. In other words, it indicates that the handshake is established.

The **receive status** and the **link status** are initialized at the timing of sending a packet described in *"4.6 Data Sharing by Copying Data in Memory Block"*, and then managed until the timing of sending a packet from the next self-station.

⬤ **Keyword**	**Receive Status**
	Link Status

6.2.2 Practical Side of Establishing Handshake

The CUnet protocol also requires that a packet to be sent as described in *"4.6 Data Sharing by Copying Data in Memory Block"*, contain a receive status.

A handshake allowing the self-station to recognize where the packet sent from the self-station is copied to can be established by determining the receive status contained in each packet to be received from the other station during the first sending of packet and the sending of packet in the next cycle. The CUnet protocol calls this a successful handshake by determining the receive status as "link established" and operates the link status. Figure 6 shows the concepts of the receive status and link status.
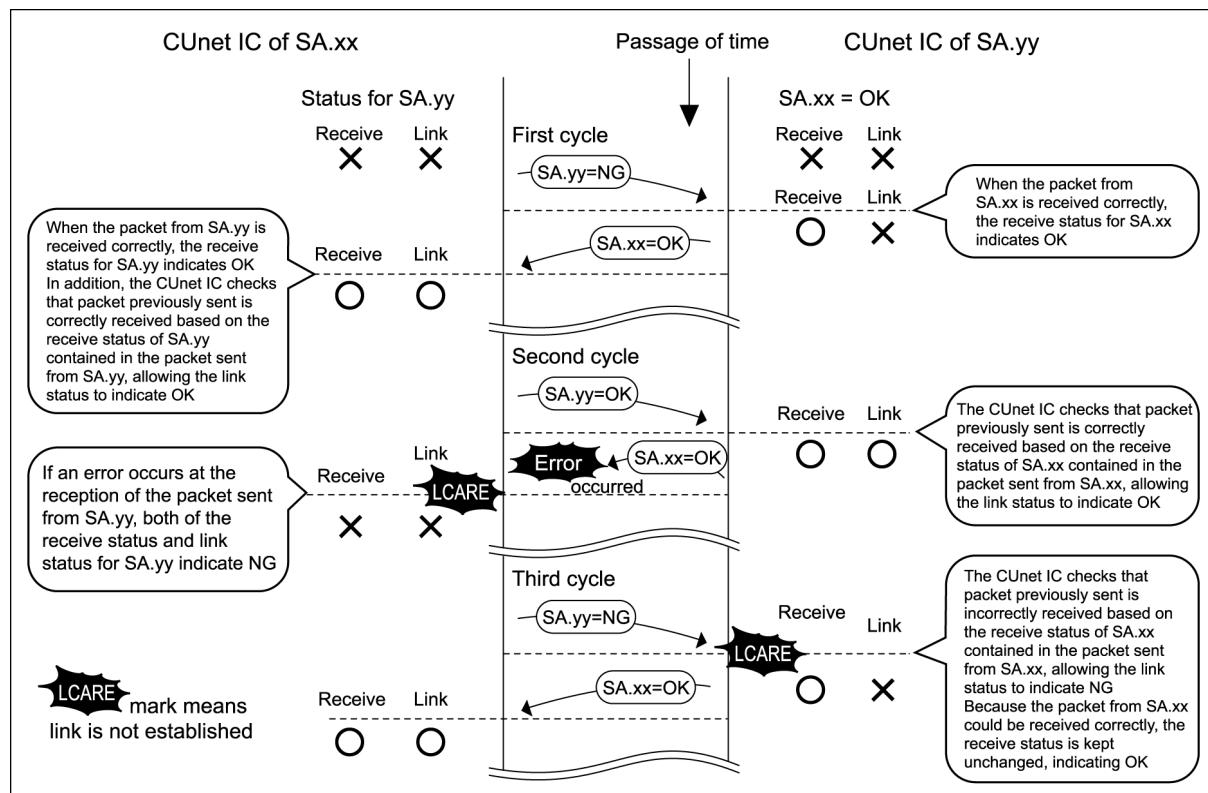

Fig. 6   Concepts of the Receive Status and Link Status

6.2.3 Disclosure of Status

The CUnet protocol defines that the CUnet IC must disclose the receive status and the link status using a flag bit so that the user CPU can read the statuses. By reading the status flag bit in the CUnet IC, the user system using CUnet can recognize the latest status of memory data sharing, and also can determine whether or not data quality is assured by the CUnet protocol as follows.

(1) **Flag bit indicating the receive status is "ON":** The flag bit assures that memory block (MB) data corresponding to the status of global memory (GM) is received by the latest cycle.

(2) **Flag bit indicating the link status is "ON":** The flag bit assures that MB data corresponding to the status of GM linked by the latest cycle. The flag also assures that MB data of the self-station is copied correctly to the station corresponding to the status.

6.2.4 Status in Practical User System

In the CUnet, once a stable environment is provided, **"link is unestablished"** is unlikely.

A **link is unestablished** when a packet that the CUnet IC received is discarded according to the test(s) due to an external noise or improper environment. This **"unestablished link"** is also described as **"link is broken"**.

If the **link is broken**, the status flag that the user system references will show either both of the receive and link status flag bits as "OFF" (refer to the second cycle of SA.xx side in Fig. 6); or the receive status flag bit as "ON" and but the link status flag bit as "OFF" (refer to the third cycle of SA.yy side in Fig. 6). These show the **link is unestablished**.

⬤ **Keyword**	**Link is unestablished/Link is broken**

6.2.5 Recovery

In common and popular communications, when a packet is prevented from being sent/received due to external noise or an improper environment, the packet can be **recovered** using a retry. In this case, a retry is necessary because if a packet is lost, communication itself is lost.

In contrast, the CUnet protocol has no retry concept (reforwarding) because it uses a cyclic time-sharing operation that allows communications to be performed continuously and cyclically. In CUnet, if the link is broken, the lost packet is always **recovered** at the next cycle.

In the mail sending/reception by the CUnet protocol, if a mail send packet is lost during transfer due to external noise or an improper environment, the mail send packet is **recovered** at the next cycle using the retry mentioned above, which reforwards the previously sent mail packet.

⬤ **Keyword**	**Recovery**

6.3 Packet Collision

In a communication network that sends packets via a network cable, packet collisions must be avoided and communications must be system-intensive.

The CUnet IC, on the other hand, uses a cyclic time-sharing operation to synchronize the timings of all the CUnet ICs connected via the network. Therefore, in the RUN phase, packet collisions never occurs. However, the following may cause packet collisions.

(1) Call packet collision in CALL phase

(2) Jammer (refer to *"6.3.2"*)

In case (1), the call packet collision is a transient event that does not affect network communication. In case (2), jammer is an exception event caused by a defect or hardware failure. Whatever the case CUnet will not affect the user system.

6.3.1 Call Packet Collision in CALL Phase

The sending interval of the call packet in the CALL phase is defined by the CUnet protocol according to the station address (SA) set for the CUnet ICs. This prevents packets from colliding consecutively even if one packet collision occurs.

For example, if the CUnet ICs at $SA = 1$ and $SA = 7$ send call packets simultaneously, packet collision occurs once but never occurs subsequently because the sending interval of the call packet of each CUnet IC differs. In this case, because there are two stations on the network, if a call packet can be transmitted to the network only once, the two stations transit to the RUN phase in which the CUnet normally operates.

6.3.2 Jammer

If there is a station that "can send a packet but cannot receive a packet" due to a failure, even if the network is in the RUN phase, the network cannot recognize the stations, the station that "can send a packet but cannot receive a packet" continues the CALL phase. In this case, even though a packet of other station in the RUN phase is being sent, a call packet is send and, thereby resulting in packet collisions.

The CUnet protocol calls such a station that "can send a packet but cannot receive a packet due to a failure" a **"Jammer"**. Because the **Jammer** cannot receive a packet, it cannot be forced to stop from the network side. In this case, the user must remove the **Jammer** or repair the failure. The CUnet protocol defines that a CUnet IC detect a **Jammer** and **warn the user system(s) about it**.

⬤ **Keyword**   **Jammer**
                 Warn about Jammer

📖 **Reference**   A jammer may exist whatever communication media including CUnet is used. For example, a station that cannot receive a packet in RS232C/RS458 or Ethernet LAN communications sends a packet without detecting that other communication is running, it can disturbs the network.

7 Increased Practicality

To increase the practicality of CUnet, the CUnet protocol defines "resizing", "expansion of owned area", and "CUnet-dedicated I/O-IC (CUnet I/O-IC)" .

7.1 Resizing

The CUnet protocol provides an initial value for the constant called "Final station" (FS) described in *"4.2 Standard Unit of Time-sharing"*. For example, the initial value in the CUnet protocol version 1.00 is 63 (3FH). In a user system with only two stations at SA = 0 and SA = 1 the network is not used during frames from SA = 2 to SA = 63 that configure a cycle.

In this case, if the value of the final station is changed to "1", the user can use CUnet with the most effective cycle. As just mentioned, changing the value of the FS is called **"resizing"** in the CUnet protocol. The frame time varies depending on this **resizing**, as shown in equation 1 in *"4.2 Standard Unit of Time-sharing"* (Fig. 7.1).
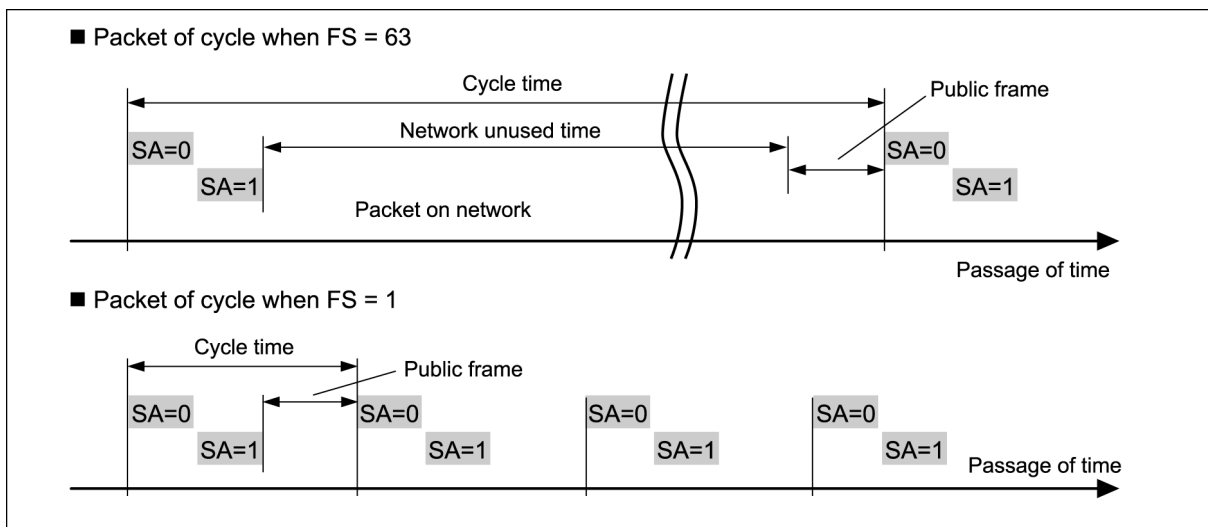

Fig. 7.1   Concept of Resizing

The CUnet protocol also defines that **resizing** be possible. In the CUnet protocol, changing the value of the FS to smaller one is called **"reduced resizing"**, and changing the value to a greater one is called **"increased resizing"** or **"expanded resizing"**.

In a CUnet with three stations at SA = 0, SA = 1, and SA = 7, if the user system executes **reduced resizing** for FS = 1, the station at SA = 7 cannot be given a timing for sending a packet in a cycle and is forced to leave the network to stop. If the station at SA = 7 starts again, it cannot enter the network because it cannot given a timing for sending a packet in a cycle. At this time, the station at SA = 7 transits to the BREAK phase. To make the station reenter the network, the station at SA = 0 or SA = 1 must execute **expanded resizing** of FS = 7 or greater.

⬤ Keyword	**Resizing**
	Reduced resizing
	Increased resizing/Expanded resizing

7.2 Expansion of Owned Area

In CUnet, one CUnet IC has one SA and owns one memory corresponding to the SA described in *"4.5 Partitioning and Owning of Memory"* and *"4.6 Data Sharing by Copying Data in Memory Block"*.

The CUnet protocol simultaneously allows CUnet to own one memory block corresponding to the SA and associated multiple memory blocks. This is called **"expansion of owned area"**.

The CUnet IC enabling **expanding owned area** has a function to set the **own width (OWN width)**, meaning to set how many areas the CUnet IC owns.

Figure 7.2 shows an example of the CUnet IC in which a station at SA = 1 has OWN = 3. The station at SA = 1 sends MB data at MB = 1 to the network when the station time (ST) is 1, and sends MB data at MB = 2 to the network when ST = 2, and sends MB data at MB = 3 to the network when ST = 3 to each network. In this way, the **expansion of owned area** can be achieved by a mechanism in which the station at SA = 1 sends data to the network as if it were the station at SA = 2 and SA = 3. Therefore an other CUnet IC cannot be set to the station address (SA = 2 and SA = 3 in Fig. 7.2) superseded by the **expansion of owned area**.

In addition, when sending mail between stations, the station address (SA = 2 and SA = 3 in Fig. 7.2) that is superseded by the **expansion of owned area** cannot be specified as a destination because there is no mail receive buffer(s) in the SA that is superseded by the **expansion of owned area**.
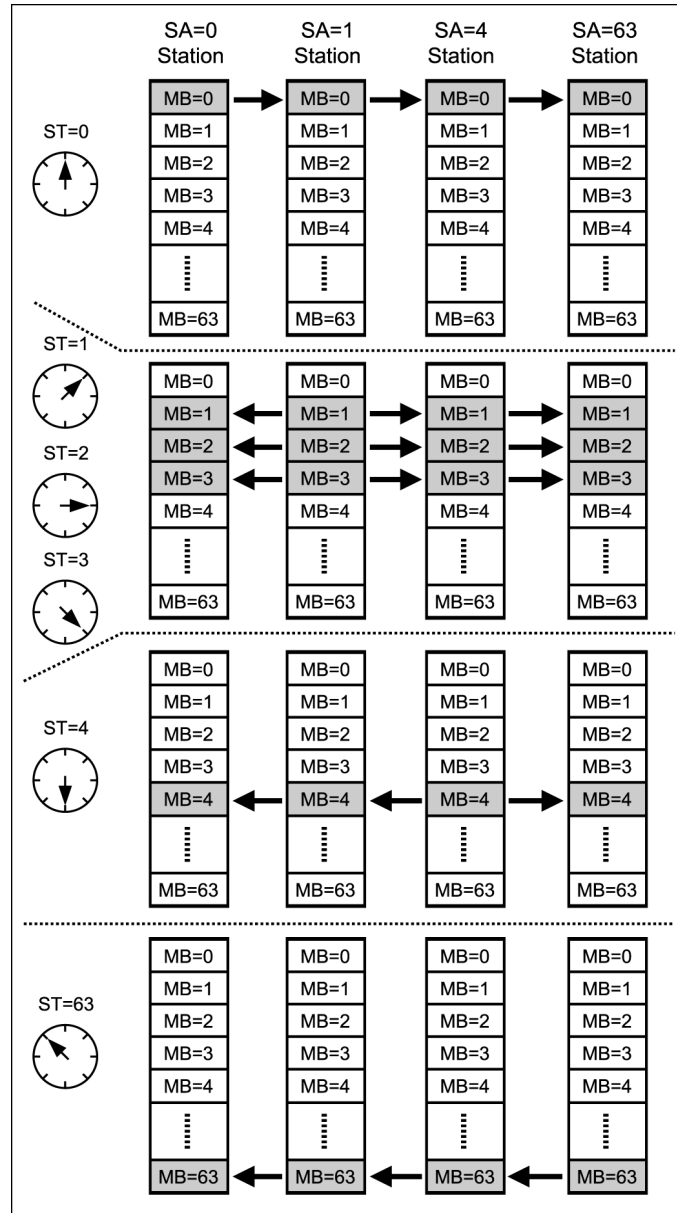

Fig. 7.2   Expansion of Owned Area

⬤ **Keyword**	**Expansion of owned area**
	OWN width (OWN)

7.3 CUnet I/O-IC

CUnet has the **CUnet-dedicated I/O-IC (CUnet I/O-IC)** enabling output of global memory (GM) data to the IO pin, or input of the state of IC pin as GM data.

The CUnet protocol requires the **CUnet I/O-IC** to have the following functions and capabilities:

    (1) Global memory or a register corresponding to it.

    (2) The capabilities other than mail sending/reception function.

    (3) The same functions and capabilities to support the phase transition and synchronization as those of the CUnet IC.

    (4) The same functions and capabilities to assure the data quality as those of the CUnet IC.

Building a CUnet by combining the CUnet IC with the **CUnet I/O-IC** can achieve **remote I/O** via GM data, enabling all stations with CUnet ICs to read the GM to recognize the state of the IC pins of the stations with the **CUnet I/O-IC**. Combining CUnet ICs with the **CUnet I/O-IC** to build the CUnet increases the practicality and compatibility with the user system (s) (Fig. 7.3).
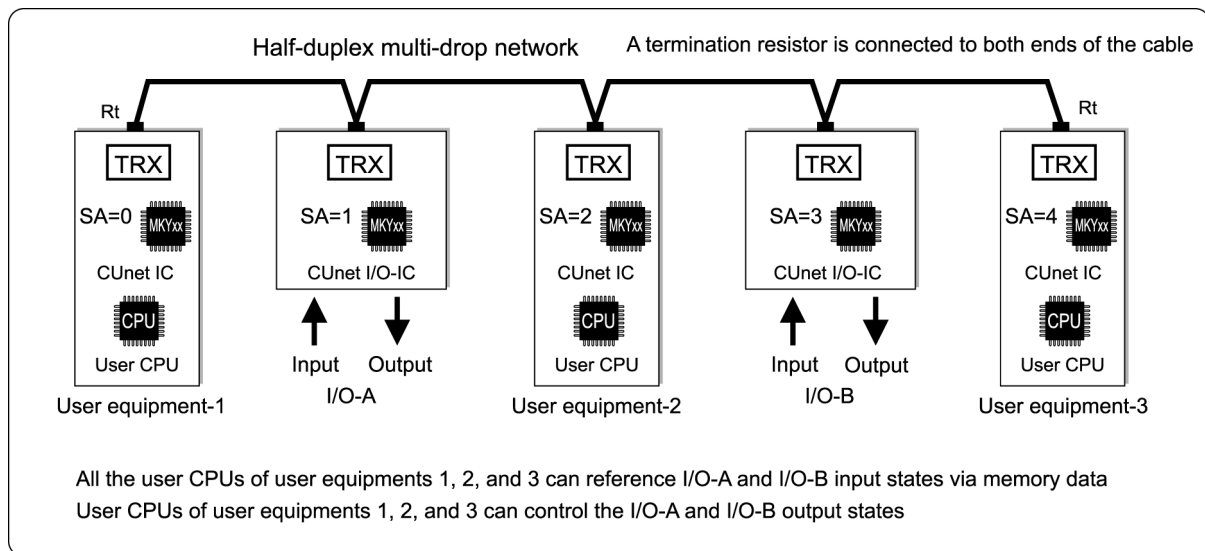

Fig. 7.3   Using CUnet I/O-IC

⚫ **Keyword**	**CUnet I/O-IC** **Remote I/O**

8 Recommended Network Cables and Baud rate

In the CUnet protocol, as described in *"3 CUnet Configuration"*, if the half-duplex communication mode allowing sending and receiving of serial communication signals between CUnet ICs is used, CUnet can be used with many types of network cables and differential driver/receiver components (TRX) comprising networks.

For practicality and economic efficiency, StepTechnica recommends the following network:

(1) **Network cables:** Cables equivalent to or better than LAN cables (10BASE-T, Category 3 or higher) and shielded network cable. Use one pair of conductors (leave the rest open).

(2) **TRX (driver/receiver components):** Differential driver/receiver (based on RS-485) components and pulse transfer for electrical isolation.

(3) **Baud rate:** Select from 12, 6, and 3 Mbps. These **baud rates** are usually selected depending on the network length (**network cable length**) (Table 1).

Each connection point of a multi-drop network cable is called a "branch". Table 1 indicates the network cable length for the CUnet when using the network with 32 or less branches.

The recommended differential driver/receiver is an RS-485-based driver/receiver.   Therefore, the branch count "32" stipulated in the RS-485 specification is used as a guide in Table 1.

Table 1 Baud Rates and Network Cable Lengths

Baud rate	Network cable length
12 Mbps	to 100 m
6 Mbps	to 200 m
3 Mbps	to 300 m

Up to 64 CUnet stations can be connected to the CUnet, enabling connection of "64" branches. This recommended network is isolated electrically by a pulse transformer and the format of signals propagated through the network is RZ (Return to Zero). Consequently, "64" branches can be connected using a standard RS-485-based driver/receiver without using DC component signals. In this case, the cable length is likely to be shorter than the value in Table 1 (due to increase of dispersion of propagated signal energy).

Caution   The **network cable lengths** in *"Table 1"* are provided as a guide for each **baud rate**, but performance is not guaranteed.

Reference
(1) A driver/receiver (based on RS-485) is commercially available.
(2) LAN cables are available, but the user can buy recommended shielded, and flexible LAN cables from us.
(3) A pulse transfer suitable for 12, 6, and 3 Mbps can be purchased from us.
(4) Network cable length can be extended by adding HUB(s). For details, refer to *"User's Manual"* for HUB-IC (MKY02).

Background information to help build a network is described in the *"CUnet Technical Guide"*. For more information about how to select components or to get recommended components, visit our Web site at **www.steptechnica.com/**

9 Summary of this Guide

CUnet is a new communication concept that enables sharing of memory data and sending/receiving of mail including dataset simultaneously. Communication using the CUnet IC with CUnet protocol provides user systems with the following benefits:

(1) Assured data quality at lower cost

(2) Constant cycle time calculated from formulae (Constancy)

(3) Real-time parallel processing

(4) Hot swap

This guide describes the basic functions and capabilities of CUnet by explaining the CUnet protocol.

When actually using the CUnet IC, it is important to understand the functions and capabilities by referring to *"CUnet IC Manual"*, *"CUnet Technical Guide"*, and *"Let's Try! CUnet"* (booklet), etc.

"Manual" updates, product information, technical reports, etc., are available at our Web site:

http://www.steptechnica.com/

Appendix   CUnet protocol Version 1.00

This section describes the parameter specifications for the CUnet protocol version 1.00 used in the MKY40, or the CUnet IC.

Network system: Cyclic time-sharing
  Network signal system: RZ signal system (Manchester Code)
  Network communication system: Half-duplex multi-drop
  Quality assurance at packet reception: Format test, CRC-16, and RZ test
  Length of frame (LOF): 151 (at normal), 256 (at frame option)
  Public frame count (PFC): 2
  Final station (initial value): 63
  Final station value (variable or fixed): Variable
  Allowable final station value: 1 to 63
  Phase count: 4 (START, CALL, RUN, BREAK)
  Number of connectable CUnet ICs: 64 max.
  Supported station address range: 0 to 63 (00H to 3FH)
Global memory size: 512 bytes
  Memory block size: 8 bytes (64 bits)
  Memory block count: 64
Maximum size of dataset at mail sending/reception: 256 bytes
  Count of mail send buffer: 1 (256 bytes)
  Count of mail receive buffer: 2 (256 bytes each)
  Partition size of dataset: 8 bytes
  Management of queue for mail sending rights: Priority rotation
  Mail sending/reception protocol control: REQ, ACK, NAK, EOT handshaking, and retry count (3 times)

CUnet Introduction Guide
(A Guide to the CUnet Protocol)