Using UART in radio data transmission with the CDP-02 module

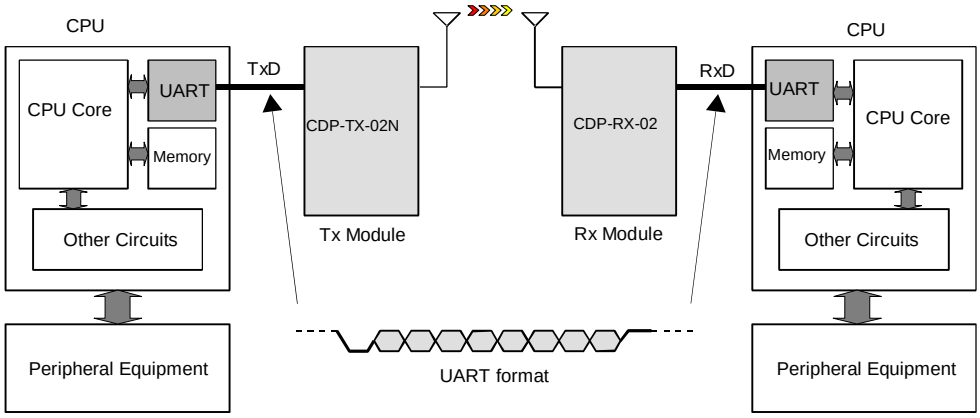
By Tomihiko Uchikawa

Abstract:
The first time a customer uses the CDP-TX-02N/RX-02N (called CDP-02 module) radio module, they are often uncertain how to format the data.
The CDP-02 handles serial digital data up to a data rate of 4800 bps. The data can be controlled BIT-by-BIT at this data speed, however this obviously places a burden on the CPU.
Normally, CPUs in communications equipment must perform main control of the communications equipment itself as well as of the radio module and the data transmission process. If specific hardware is designed for received data processing, the task load of the CPU can be reduced, however this is only possible for applications limited in size, cost and current etc.
Most of the CPUs currently available on the market have a UART interface and some of them have a multi-UART interface. This document explains how to connect the CDP-02 directly to a UART and the use of UART format for radio data communication.
When a UART interface is used for radio transmission, detecting the start of the data is important. In this document, we explain methods 1 and 2 where a delimiter is inserted to the beginning of the data packet at the transmitter unit and the receiver unit detects it easily.
Method 1 uses logic level - high as delimiter. Method 2 uses several bytes of characters as delimiter. The result is the same, but Method 1 requires time-control with timer interruption, but Method 2 does not.
Both methods are easy to implement in systems. We hope you find this document helpful in designing your radio equipment.

Connection of CDP-02 module:

As Figure 1 shows, the CDP-02 module is simply connected to the UART of a CPU and the data is processed according to the program.

Figure 1: Radio Transmission



The CDP-02 module input voltage range covers 2.4 to 12 V. The UART of the CPU operates with 3 or 5 V and can be connected to the CDP-module directly. The format of data transmitted between radio devices is configured as follows.
The following 2 states are considered in the radio module. Method 1 and 2 can be adapted to both states.

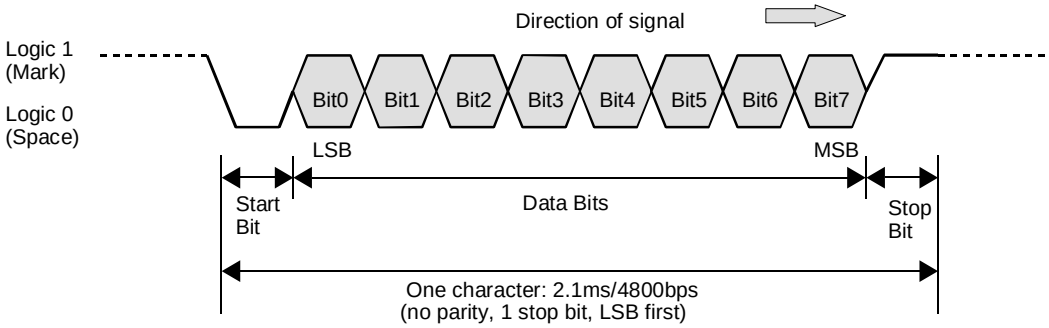
1. The power of CDP-02 module is normally off, the CDP-02 module turns on when data is sent. The power is turned off when transmission is finished.
2. The power of CDP-02 module is always on, and data is transmitted when the data is fed to CDP-module. (The carrier is emitted even without data input).

Data format between radio devices: * The data rate of the following example is 4800 bps.

The UART data format that is used to communicate between radio devices is shown in figure 2. Set the control registers as shown below and form 10 bits with a start bit and stop bit.

1. Data is configured as 8 bits, which enables handling of binary data.
2. Set non-parity and 1 bit for a stop bit to ensure efficient data transmission.

Figure 2: Data format between radio modules
Asynchronous serial logic waveform



When CDP-02 operates at 4800 bps, 1 bit width is approx. 210 us, so it takes 2.1 ms to transmit 1 character (1 byte) of data i.e. a total of 10 bits. It should be noted when you observe the data with an oscilloscope that the UART normally outputs the data with LSB first.

Example of data frame configuration:

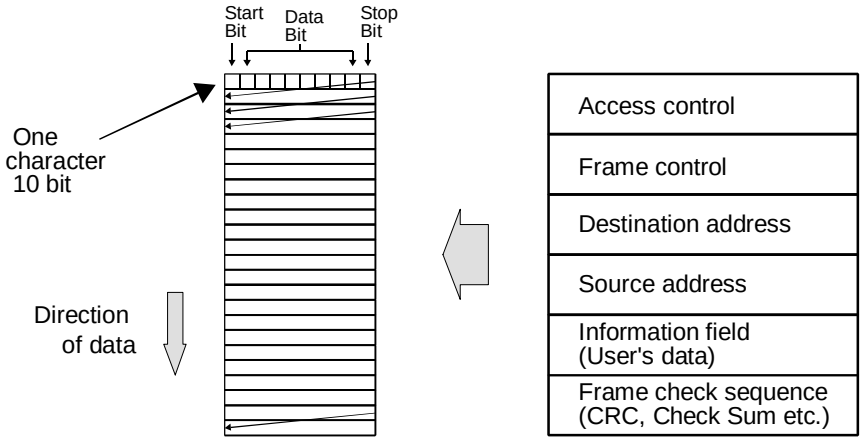
The product must decide the destination of the data, error handling and communication control for radio data communication. A frame configuration that contains data and a number of fields is often used for data communication, and data framing is performed at the transmitter unit.

An example of frame configuration is shown in Figure 3.

The frame is made up of a number of fields consisting of the required number of characters. The characters comprise 10 bits as shown in Figure 2.

An error checksum is added to the end of the frame on the transmitter side. The checksum can be used for simple error checking but the CRC-code is used for secure communication. The code is processed at the receiver unit. As the CDP-02 is a radio module for one-way communication, frames that include data errors are ignored. Therefore it is best to send the same data frame two or three times.

Figure 3: Example: Information frame configuration



Transmission method between radio devices:
As described above, the data is transmitted in the frame configuration as shown in Fig. 4 and 6. You need to make a program, which performs transmission of dummy data when no information (data) frame is input. It takes approx. 90 ms to stabilize the transmission frequency of the CDP-02 module when it is first turned on, therefore dummy data is fed as a preamble to the UART during the start up period for radio transmission. Dummy data is necessary to make the receiver data slicer capable of receiving the subsequent data. Method 1 and Method 2 use different codes for dummy data.
In Method 1, any data can be used, but CC [hex] or 33 [hex] which is often used for frame synchronization is recommended. In Method 2, FF [hex] is used.

Frame header configuration and actual transmitter and receiver procedures

Method 1. The use of logic high level as frame head identification:
A close up of the information frame is shown in figure 4.
Frame Identification is performed at the receiver unit by detecting the 15 ms high level delimiter which is added in front of the data frame at the transmitter unit. When the receiver unit detects the delimiter, the data of the pre-set frame length is written in a receiver buffer and is processed as necessary. The radio transmission method using UART has the advantages that the frame head can be detected in a short period of time and binary data transmission is possible by controlling the transmission and reception with the frame length.
The following explains transmission and reception procedures when the power of CDP-02 module is always on and radio communication is performed as needed. The same procedure can be applied to cases where the power of the CDP-module turns on at each transmission. However, please keep in mind that CDP-02 has an unstable period of approx. 90 ms just after the power is turned on, thus dummy data (CC [hex]) should be sent to UART for at least 100 ms. Please refer to Figure 5 which shows an outline of the program.

Transmitter unit procedure
1. Dummy data is transmitted as CC [hex] while waiting for a transmission. This is necessary for the CDP-receiver module to output the right data immediately when actual data is received.
2. The data is prepared in the transmitter buffer. Pre-calculated code is added as a frame check sequence to the end of the frame.
3. Approx. 15 ms high level is inserted as a delimiter prior to transmitting the information frame. First transmission interruption is prohibited. The transmission stops at the high state because the dummy data being output ends at the stop bit = high state. A timer counts 15 ms then transmission interruption is permitted and the first byte of the frame data is fed to a transmitter buffer of the UART. Note that more than a 20 ms long high should be avoided to prevent unstable processes in the CDP-receiver module.
4. Transmission interruptions happen continuously during the processing of 3 above. All the data in the frame is sent in the transmission interruption routine. Dummy data is then sent again after the information frame transmission and then the transmission unit is set to transmission waiting status.
5. The next frame is sent as needed. However, processing time in the receiver unit should be taken into consideration.

Receiver unit procedure
1. A 10 ms timer and a receiver buffer is provided.
2. The timer is reset each time the interruption routine of the UART occurs. If there is no information (data) frame, the dummy data is always sent from the transmitter and the timer is reset each time (only 2.1 ms for one character is counted).
3. The timer starts counting when the delimiter inserted before the information frame at the transmitter unit is detected. A flag that indicates the elapse of 10 ms is set when the timer reaches the set value.
4. The next receiving data after the flag is on is the first data of the information frame, therefore the data is acquired in the length of the frame in a receiver buffer.
5. The data in the buffer is read and a frame check is conducted, then the received data is processed as necessary.

Figure 4: Radio transmitting Method 1

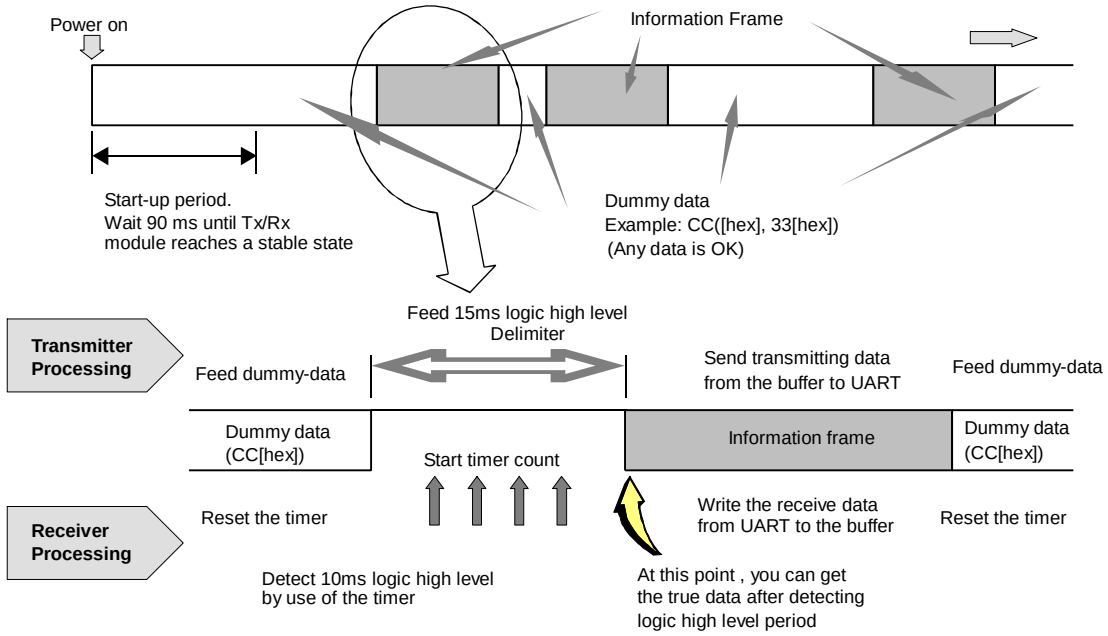
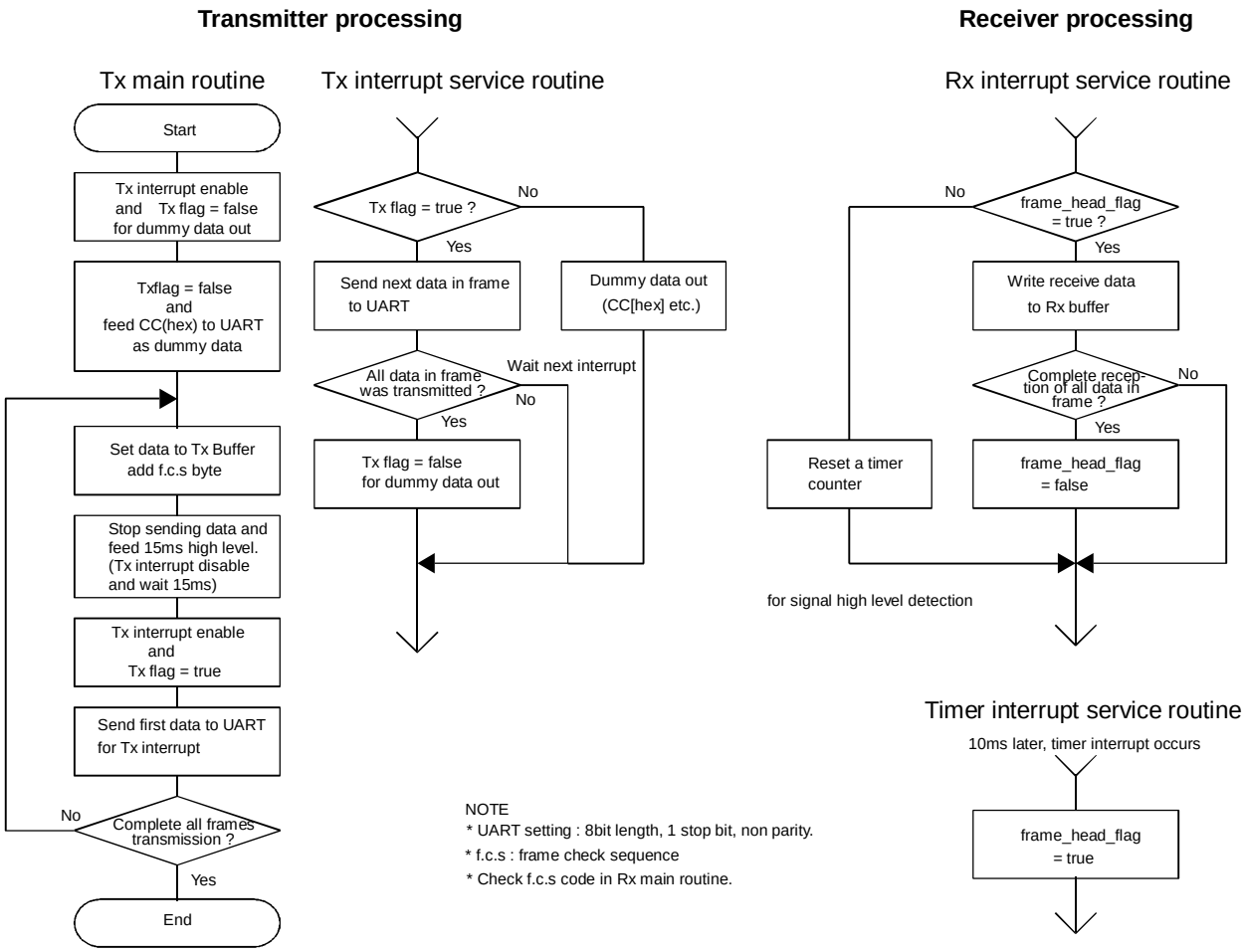


Figure 5: General flow chart of Method 1



Method 2. Use of character string as frame head identification

A close up of the information frame is shown in Figure 6.
Several bytes of delimiter (an identification character string like '@FrameHead") is inserted as the frame head at the transmitter unit, and the receiver unit detects the delimiter.
When the receiver unit detects the delimiter, the data of pre-set frame length is written in a receiver buffer and the data is processed as necessary.
The UART of the receiver unit is synchronized with the transmitted signal by sending FF [hex] as the dummy data, and it enables the receiver unit to detect the frame head without fail.
The character string as the delimiter is compared each time when the receiver interruption occurs. When all the character bytes are matched, the flag is set and subsequent data is stored.
As mentioned in Method 1, binary data transmission is possible by controlling transmission and reception with frame length. Also the procedure for transmission when the CDP module power turns on is the same as Method 1. Please refer to Figure 7, which shows an outline of the program.

* Dummy data
Other data like 00 [hex], 80 [hex] and FE [hex] can be used as dummy data in Method 2. However this code is limited to types in which the falling edge does not appear in UART format other than the start bit (as LSB first). The start bit cannot be detected with code 55 [hex] or most other code.

Transmitter procedure:
1. Dummy data FF [hex] is transmitted during the transmission waiting status. This is necessary because it helps the CDP-receiver module to shift to data processing immediately.
2. Data is prepared in the buffer of the transmitter unit. Pre-calculated code is added as the frame check sequence to the end of the frame.
3. The first byte of the frame data is fed to the transmission buffer of the UART.
4. Transmission interruptions occur continuously during the processing of 3 above. All the data in the frame is sent in the transmission interruption routine. The dummy data is sent after the transmission and sets the transmission unit to the transmission waiting status.
5. The next frame is transmitted as needed. However, processing time in the receiver unit should be taken into consideration.

Receiver unit procedure:
1. All receiver processes are performed in the interruption routine. The program code that writes the data from the next reception interrupt goes into a buffer when the delimiter is detected, is written first. (See figure 7). Then the program code for delimiter detection is written. This is because the data following the final character of the detected delimiter is the first data of the information field.
2. Check the data of the delimiter character string each time when the reception interruption occurs. For example, if '@FrameHead' is used, a counter is incremented when the first byte is compared and matched with '@', if it does not match then clear the counter. This procedure is repeated. A flag is set to store the next reception interruption data when the value of the counter reaches 10 which means that the delimiter has been detected.
3. When the flag is on, the data is acquired in preset lengths between the transmitter unit and the receiver unit into a buffer, and the data is processed as necessary.

Figure 6: RF Transmitting Method 2

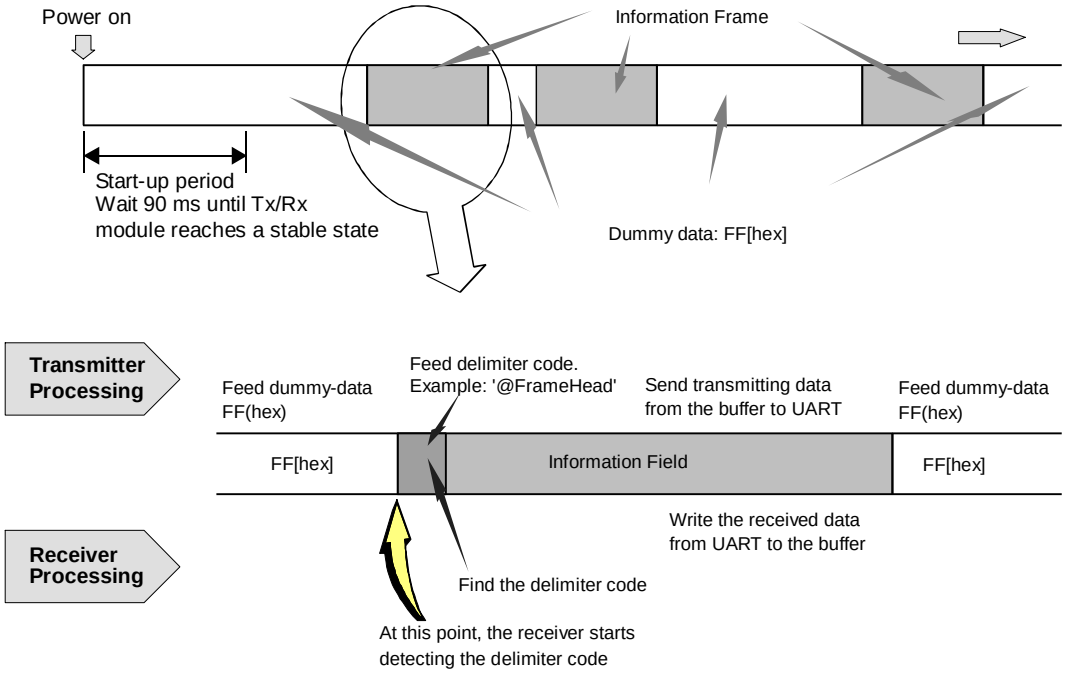
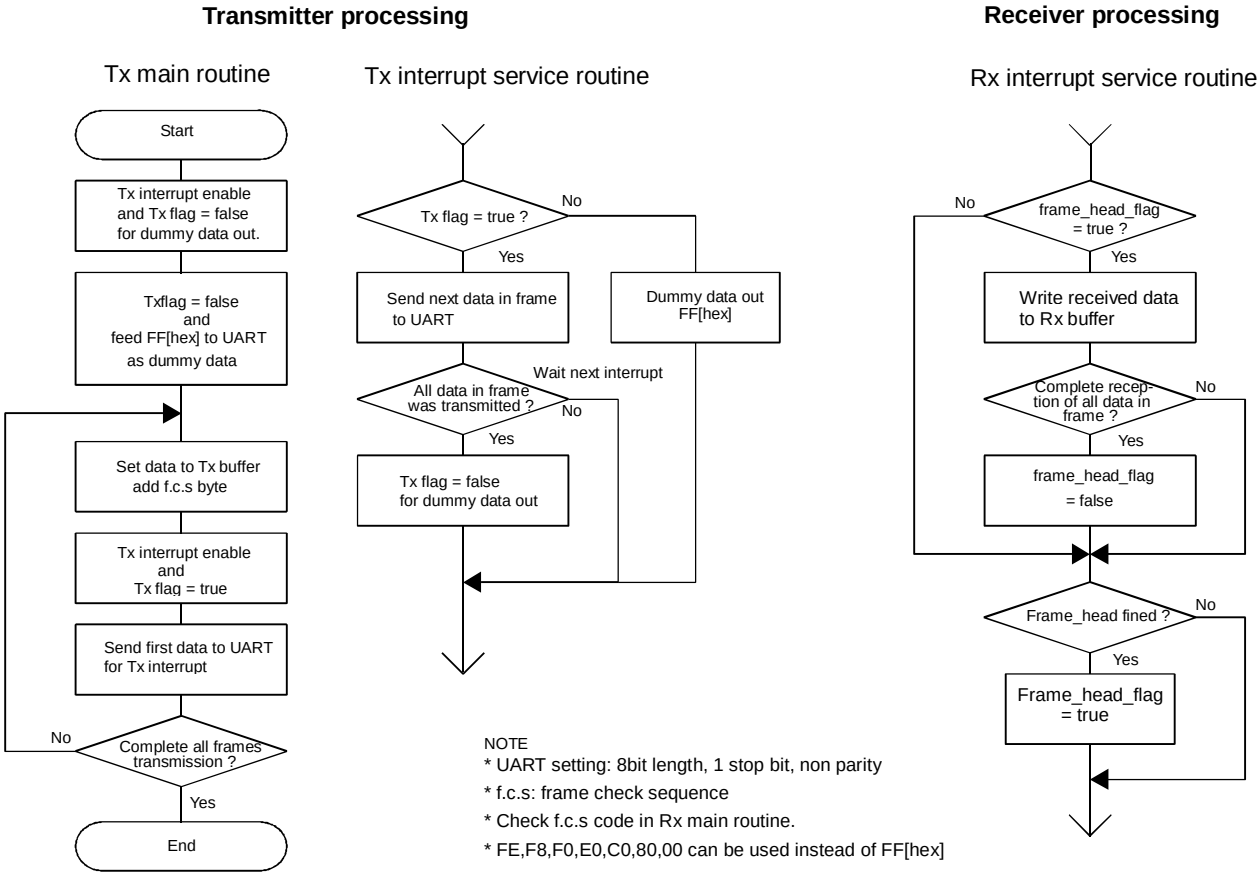


Figure 7: General flow chart of method 2



Remarks on the program
*The program is designed for one-way communication. When the data frame is transmitted continuously, the data processing time in the receiver units should be taken into consideration before sending the next data frame.
*Error checking is performed frame by frame. Parity check of UART is not necessary.
***Important!** In Method 1, the receiver detects 10 ms consecutive high level. Therefore the data in the information frame has to be transmitted continuously without a longer pause.

Operating environment

- Radio module: Circuit Design CDP-TX-02N and CDP-RX-02N
- CPU: Mitsubishi microcomputer: M16C/62A Group M30620FCAFP
- CPU development language C: NC30 Debugging environment: KD30
- OS: Windows 2000, Windows Me
- Compatible Windows program: Original development application (use Delphi ver.6.0, Serial Control VCL)

If you need further information about our company and products, please visit http://www.circuitdesign.jp or contact our local distributor.