IrDA Control Errata to IrDA Control Specification version 1.0

Last Modified: Oct 26, 1998

Following are a list of corrections/clarifications and suggested amendments or changes to the IrDA Control specification Version 1.0 dated June 30, 1998. The closed errata are organized by the date they were closed. Open errata are at the end.
The closed errata should be considered an integral part of the IrDA Control version 1.0 specification. No change to the version number is required based on these changes.
The points are classified according to the following scheme:

- CORRECTION: a change required to correct an error in the existing document corrections may change the specified behavior of the protocol to match that which was originally intended by the authors
- CLARIFICATION: textual enhancement that provides clearer explanation of a protocol element without changing any behavior
- MODIFICATION: a modification of the currently specified behavior that adds but does not delete any function from the protocol
- CHANGE: a modification of the currently specified protocol that may add and or delete function from the protocol
- PROBLEM: a known problem for which an alteration to the document has yet to be proposed.

1. July 1998 - Closed Errata

All errata in this section were approved and closed at the July 1998 IrDA meeting.

1.1 CRC Field (3.2.3)

CLARIFICATION

An example of VHDL implementation follows to describes the process.

For better understanding of CRC calculation algorithm, an example of VHDL implementation is described.

```
library IEEE;
use IEEE.std_logic_1164.all;
use WORK.all;

entity TxCRC is
  port(
    TxRESET :in          std_logic;
    TxLongFrame :in std_logic;
    TX_CRCCLK :in        std_logic;
    TXCRCIN :in          std_logic;
    ADDTXCRC :in std_logic;
    CRCOUT :out          std_logic
------------------------------------------------------------
-- TxRESET        ... Reset signal for this module
-- TxLongFrame    ... H for long frame, L for short frame
-- TX_CRCCLK      ... Clock signal for this module
-- TXCRCIN        ... Input data sequence for this module
-- ADDTXCRC       ... This signal is L while calculating CRC value and H when transmitting CRC value
-- CRCOUT         ... Output signal from this module
------------------------------------------------------------
);
end TxCRC;

architecture ARC of TxCRC is
  signal CRCFF          :std_logic_vector(15 downto 0);
  signal CRC_LONGIN     :std_logic;
  signal CRC_SHORTIN    :std_logic;

begin
------------------------------------------------------------
-- Compute CRC8   (x8+x7+x2+1) for Short Frame
-- Compute CRC16  (x16+x15+x2+1) for Long Frame
------------------------------------------------------------
  CRC_LONGIN    <= (CRCFF(15) xor TXCRCIN) and (not (ADDTXCRC));
  CRC_SHORTIN   <= (CRCFF(7) xor TXCRCIN) and (not (ADDTXCRC));

  process(TxRESET,TX_CRCCLK)
  begin
```

```
    if TxRESET='1' then
      CRCFF<="1111111111111111";
-------------------------------------------------------------
-- The CRC register is preset to all "1"s
-- prior to calculation of the CRC value.
-------------------------------------------------------------
    elsif TX_CRCCLK='1' and TX_CRCCLK'event then
      if TxLongFrame='1' then
            CRCFF(0)          <= CRC_LONGIN;
            CRCFF(1)          <= CRCFF(0);
        CRCFF(2)      <= CRCFF(1) xor CRC_LONGIN;
        CRCFF(3)      <= CRCFF(2);
        CRCFF(4)      <= CRCFF(3);
        CRCFF(5)      <= CRCFF(4);
        CRCFF(6)      <= CRCFF(5);
        CRCFF(7)      <= CRCFF(6);
        CRCFF(8)      <= CRCFF(7);
        CRCFF(9)      <= CRCFF(8);
        CRCFF(10)     <= CRCFF(9);
        CRCFF(11)     <= CRCFF(10);
        CRCFF(12)     <= CRCFF(11);
        CRCFF(13)     <= CRCFF(12);
        CRCFF(14)     <= CRCFF(13);
        CRCFF(15)     <= CRCFF(14) xor CRC_LONGIN;
      elsif TxLongFrame='0' then
        CRCFF(0)      <= CRC_SHORTIN;
        CRCFF(1)      <= CRCFF(0);
        CRCFF(2)      <= CRCFF(1) xor CRC_SHORTIN;
        CRCFF(3)      <= CRCFF(2);
        CRCFF(4)      <= CRCFF(3);
        CRCFF(5)      <= CRCFF(4);
        CRCFF(6)      <= CRCFF(5);
        CRCFF(7)      <= CRCFF(6) xor CRC_SHORTIN;
      end if;
    end if;
  end process;

  process(TxRESET,TX_CRCCLK)
  begin
    if TxRESET='1' then
      CRCOUT<='0';
    elsif TX_CRCCLK='1' and TX_CRCCLK'event then
      if ADDTXCRC='0' then
        CRCOUT<=TxCRCIN;
-------------------------------------------------------------
-- This module outputs the input data while ADDTXCRC is L
-------------------------------------------------------------
      else
        CRCOUT<= ( TxLongFrame and not(CRCFF(15)) ) or
              ( not TxLongFrame and not(CRCFF(7)) );
-------------------------------------------------------------
```

```
-- This module outputs the CRC value when ADDTXCRC is H
-- CRC value is inverted as it is sent out
-------------------------------------------------------------
      end if;
    end if;
  end process;

end ARC;
```

2. October 1998 - Closed Errata

All errata in this section were approved and closed at the October 1998 IrDA meeting.

2.1 Corrections for Figures and Tables

CORRECTION

The errors to be corrected in this section are all caused by operation for editing tables or diagrams. The specification Final version 1.0d does not have such errors.
However, the Final Specification (Final Revision 1.0 June 30, 1998) includes some errors such as wrong positioned pointers and unnecessary ruled lines, which are amended here. The tables and diagrams shown here as correct ones are identical to those found in Final version 1.0d.

Table 3.2 (3.2.2)
In the table of Specification 1.0, the positions of the pointers showing First chip and Last chip are incorrect. They should be corrected as shown below.

Field Name	Pulse Sequence
AGC	1 1 1 1
PRE	0 1 0 1 0 1 0 1 0 1
STS	0 1 1 0 1 1 0 1 0 0
STL	0 1 0 0 1 0 1 1 0 1
STO	0 1 0 0 1 0 1 1

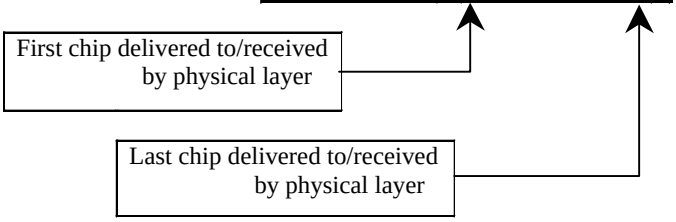


Table 3.2: Pulse Sequence Representation for AGC, PRE, STS, STL, and STO

Figure 3.4 (3.2.2)

In the table of Specification 1.0, the positions of the pointers showing First chip and Last chip are incorrect. They should be corrected as shown below.

Data Value (Byte)	Resulting DBSs								Resulting 16 PSM data symbol streams (chips and symbols transmitted from left to right for LSB first reception)
	MSB							LSB	
	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	
0x1B	0	0	0	1	1	0	1	1	
									00011110
									01010000
									00011110 01010000
0x0B	0	0	0	0	1	0	1	1	00011110 10100000
0xF5	1	1	1	1	0	1	0	1	00000101 11100001

First chip delivered to/received by physical layer

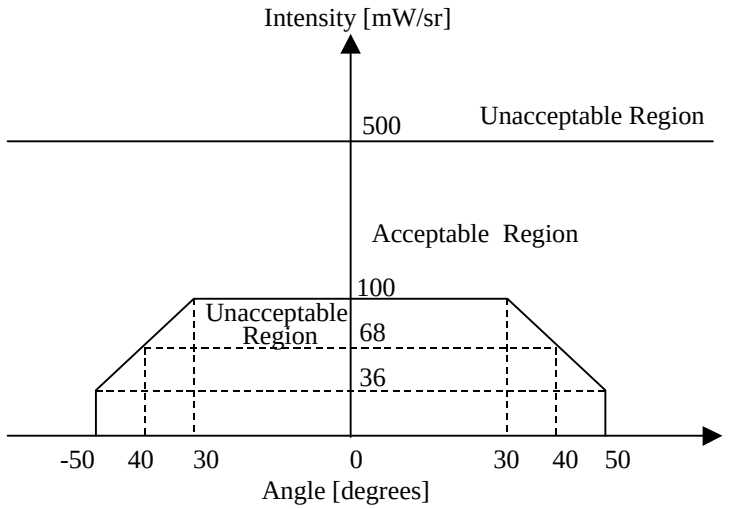
Last chip delivered to/received by physical layer

Figure 3.4: Example of packet generation procedure

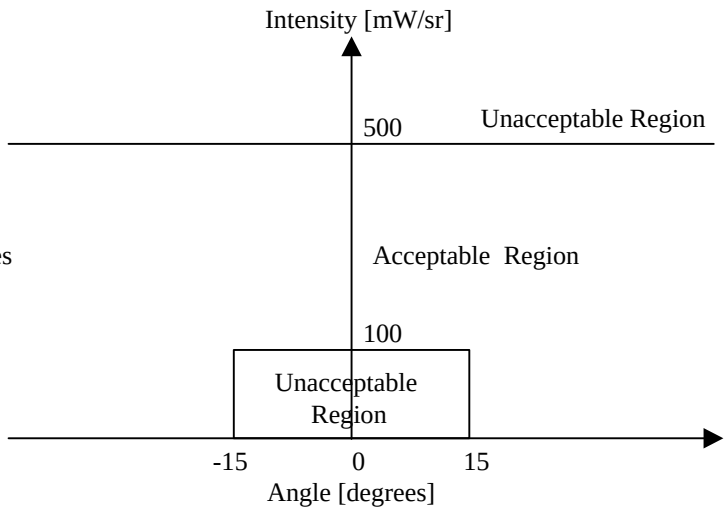
Figure 3.9 (3.5)

In the diagram of Specification 1.0, the area showing Unacceptable Region is missing. It should be corrected as shown below.

(a) Host Devices



(b) Peripheral Type 1 Devices



(c) Peripheral Type 2 Devices

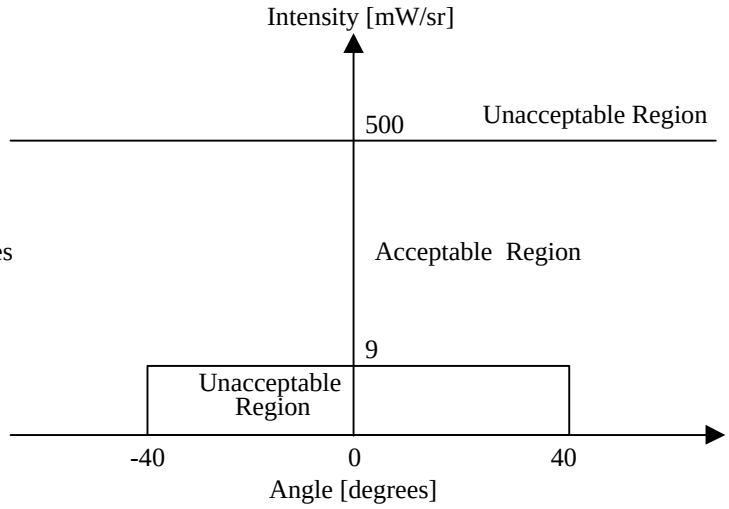
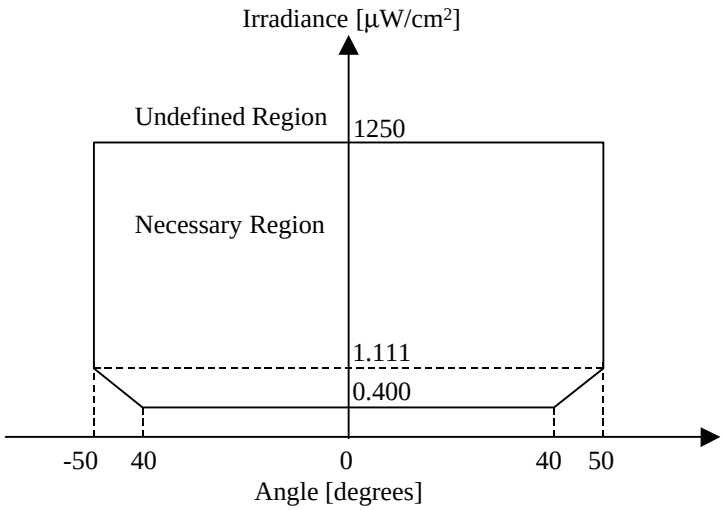


Figure 3.9 : Acceptable Intensity Region in horizontal angular range

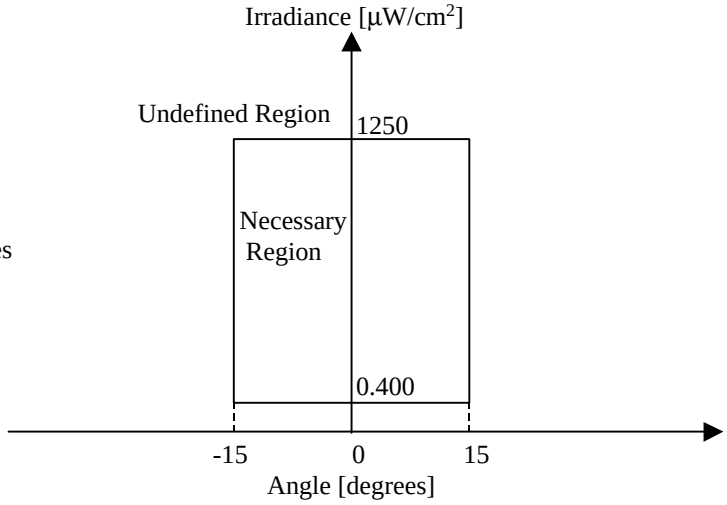
Figure 3.10 (3.6)

In the diagram of Specification 1.0, the area showing Necessary Region is wrong. It should be corrected as shown below.

(a) Host Devices



(b) Peripheral Type 1 Devices



(c) Peripheral Type 2 Devices

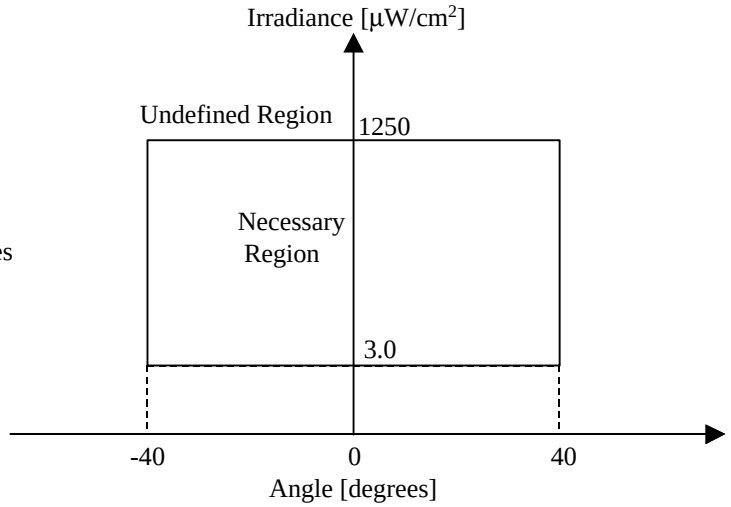


Figure 3.10 : Necessary Irradiance Region in horizontal angular range

Table 4.1(4.1)

The table of Specification 1.0 has unnecessary ruled-lines. It should be corrected as shown below.

Host Status		From Host to Peripheral (MAC payload length)	From Peripheral to Host (MAC payload length)
To unenumerated peripherals		Short (0 - 9 bytes)	Short (0 - 9 bytes)
To unbound peripherals in Mode-1/2		Short (0 - 9 bytes)	Short (0 - 9 bytes)
To bound peripherals	Mode-0	-	Short (0 - 9 bytes)
	Mode-1	Short (0 - 9 bytes)	Short (0 - 9 bytes)
		Short (0 - 9 bytes)	Long (0 - 97 bytes)
		Long (0 - 97 bytes)	Short (0 - 9 bytes)
	Mode-2	Short (0 - 9 bytes)	Short (0 - 9 bytes)

Table 4.1 Possible Short/Long frame combination

Table 4.2 (4.1)

Host mode	IrDA 1.1 device	Peripherals at the CL polling rate	Peripherals at the NCL polling rate		Entire Polling Cycle Time
		Short	Long	Short	
Mode-1	0	0	0	8	8 T_{SS}
			1	7	$1T_{SL}$ + 7 T_{SS}
			2	6	$2T_{SL}$ + 6 T_{SS}
			3	5	$3T_{SL}$ + 5 T_{SS}
			4	4	$4T_{SL}$ + 4 T_{SS}
			5	1	$5T_{SL}$ + 1 T_{SS}
		1	0	7	10 T_{SS}
		2		6	12 T_{SS}
		3		5	20 T_{SS}
		4		1	17 T_{SS}
Mode-2	1	0	0	2	T_{IrDA} + 2 T_{SS}

The table of Specification 1.0 has unnecessary ruled-lines. It should be corrected as shown below.

Table 4.2 Maximum number of peripherals that can be accommodated in a Host mode

Table4.4 (4.2.3)

The table of Specification 1.0 has unnecessary ruled-lines. It should be corrected as shown below.

		Frame from Host device		
		Meaning	1	0
MAC Control field	**D7**	Packet direction	1 (Host is sending)	
	D6	Bind timer restarted	Restarted	Not Restarted
	D5	Long frame enable	Enable	Disable
	D4	Device hailing	Hailing	Not hailing
Peripheral Address field	**D3**	(Peripheral Address)	-	-
	D2		-	-
	D1		-	-
	D0		-	-

		Frame from Peripheral device		
		Meaning	1	0
MAC Control field	**D7**	Packet direction	0 (Peripheral is sending)	
	D6	Polling Request	Request	No Request
	D5	(Reserved)	(Reserved)	
	D4	(Reserved)	(Reserved)	
Peripheral Address field	**D3**	(Peripheral Address)	-	-
	D2		-	-
	D1		-	-
	D0		-	-

Table 4.4 MAC Control field

Table 4.6(4.5.1)

The table of Specification 1.0 has unnecessary ruled-lines. It should be corrected as shown below.

Bits	Denotes	Meaning
D0	Upper Layer Protocol	0 – Host does not support HID-IrDA Control Bridge 1 – Host supports HID-IrDA Control Bridge
D1		0 – Host does not support HA-LLC 1 – Host supports HA-LLC
D2		Reserved (for the 3rd protocol)
D3		Reserved (for the 4th protocol)
D4-D15	Pre-assigned number or a pseudo random number	

Table 4.6 HostID field Definitions

Table 4.7(4.5.1)

The table of Specification 1.0 has unnecessary ruled-lines. It should be corrected as shown below.

Bits	Denotes	Meaning
D0	Upper Layer Protocol	0 – Host does not support HID-IrDA Control Bridge 1 – Host supports HID-IrDA Control Bridge
D1		0 – Host does not support HA-LLC 1 – Host supports HA-LLC
D2		Reserved (for the 3rd protocol)
D3		Reserved (for the 4th protocol)
D4	Support for Long frames from host	0 – Host does not support long frames from host 1 – Host supports long frames from host
D5	Support for Long frames to host	0 – Host does not support long frames to host 1 – Host supports long frames to host
D6-D15	Reserved	

Table 4.7 HostInfo field Definitions

Table 4.9(4.5.2)

The table of Specification 1.0 has unnecessary ruled-lines. It should be corrected as shown below.

Bits	Denotes	Meaning
D0	Upper Layer Protocol	0 – Peripheral does not support HID-IrDA Control Bridge 1 – Peripheral supports HID-IrDA Control Bridge
D1		0 – Peripheral does not support HA-LLC 1 – Peripheral supports HA-LLC
D2		Reserved (for the 3rd protocol)
D3		Reserved (for the 4th protocol)
D4	Support for Long frames from host	0 – Peripheral does not support long frames from host 1 – Peripheral supports long frames from host
D5	Support for Long frames to host	0 – Peripheral does not support long frames to host 1 – Peripheral supports long frames to host
D6	Max. Polling Rate	0 – NCL Polling Rate 1 – CL Polling Rate
D7-D15	Reserved	

Table 4.9 PeripheralInfo field Definitions

2.2 DC-Biased Scheme

ADDITION

Following descriptions are about the method of reducing the interference on legacy remote control systems (the DC-Biased scheme).
The 16-Pulse Sequence Modulation (16PSM) scheme of IrDA Control is designed to decrease IR energy on the frequency band 33kHz - 40kHz, however, it still has influence on receivers of legacy remote control systems.

As shown in Appendix H, the DC-Biased scheme can obtain a great improvement for the reduction of the interference, moreover it has the interoperability with the original 16PSM scheme. Therefore, the DC-Biased scheme is one of the solutions for the coexistence with legacy remote control systems and it should be appended to the Final Specification 1.0. The descriptions are as follows.

The following paragraph needs to be appended next to the second paragraph in section 3.1(Overview), in order to give an introduction to Appendix described below.

“However, receivers of many legacy remote control systems also detect other frequencies from 33kHz - 40kHz band and therefore the 16PSM scheme still has influence on such remote control systems. So, for those systems, which operate under the existence of legacy remote control systems, it is highly recommended to implement the DC-Biased scheme described in Appendix H.”

In addition, the following appendix needs to be added, since the IrDA-Control can be greatly improved in the interference on legacy remote control system with this scheme.

Appendix H. Reduction of interference on legacy remote control systems

H.1 Introduction

As shown in Figure A.1, the spectrum of the 16PSM signal defined in chapter 3 can be divided into two components. One is a pass band component around 1.5MHz, and another is a base band component around 0Hz.

The pass band component is essential, while the base band component is not necessary for communication.

The 16-Pulse Sequence Modulation (16PSM) scheme of IrDA-Control is designed so that its base band component minimizes the interference on the frequency band used by legacy remote control systems. However, it still has influence on receivers of those legacy remote control systems, which also detect other frequencies from 33kHz - 40kHz band.

The DC-Biased scheme eliminates the unnecessary base band component of 16PSM signal and widely decreases the interference on legacy remote control systems at the cost of 2.66 times of IR power.

For those IrDA-Control systems, which operate under the existence of legacy remote control systems, it is highly

recommended to implement the DC-Biased scheme described in this section.

H.2 DC-Biased Scheme

The energy of base band component caused by the alternation of the existence of the SEPC

Figure H.1 shows the original IrDA-Control signal defined in chapter 3 and Figure H.2 shows the DC-Biased IrDA-Control signal. In the DC-Biased scheme, IR is continuously emitted at half intensity of the peak of 1.5MHz sub-carrier pulse, among every SEPC. This scheme keeps the DC level in the packet constant and eliminates the base band component.

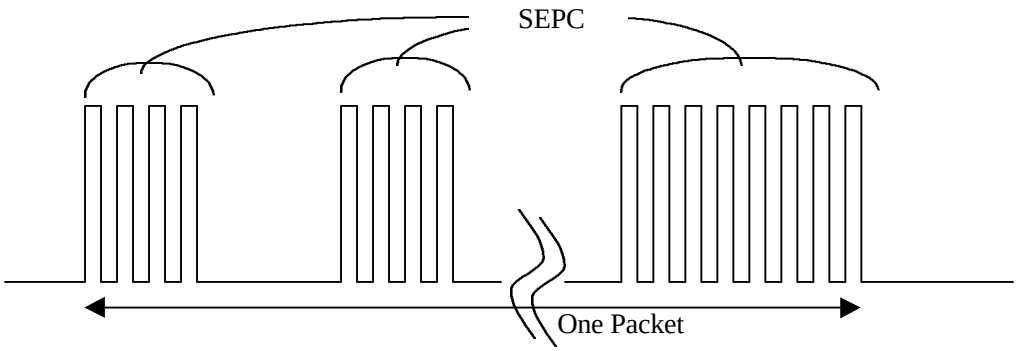


Figure H.1 Original IrDA-Control Signal

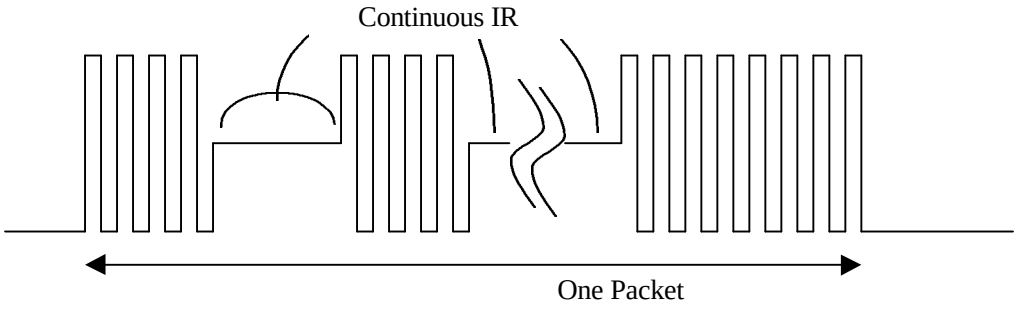


Figure H.2 DC-Biased IrDA-Control Signal

The performance of this scheme highly depends on the intensity of the continuous IR emission part. Therefore, it is preferable that the IR intensity of the continuous emission part is precisely same as the average IR intensity of SEPC.

H.3 Effectiveness of DC-Biased Scheme

Figure H.3 and Figure H.4 show the spectrums of the original IrDA-Control signal and the DC-Biased IrDA-Control signal respectively. As shown in those figures, the energy besides the range of 33kHz to 40KHz observed in Figure H.3 almost disappears in Figure H.4, because this scheme eliminates the base band component.

Figure H.5 shows the experimental environment of measurement of the effectiveness of the DC-Biased scheme. As shown in the figure, an IrDA-Control signal transmitter is placed in front of the IR receiver of a TV (SHARP Network Vision 32C-PC1[1]) at a distance of 1m. And the probabilities of the TV controlled successfully in 20 times trials are measured at each distance (a) between the TV and the remote-controller while the following 3 signal patterns are transmitted at 160mW/sr from the IrDA-Control signal transmitter.

1. Original IrDA-Control Signal
2. DC-Biased IrDA-Control Signal
3. No Signal

One short packet[2] signal is transmitted in every 27.6ms in condition 1 and 2. This simulates the situation where an active peripheral is existing in front of the TV.

Table H.1 shows the results of this experimental test on the TV and its remote controller.

As shown in Table H.1, a large improvement can be observed by applying the DC-Biased scheme. While there is no distance where the probability become more than 60% when the original IrDA-Control signal is being transmitted, the performance is almost the same as no ambient noise when the DC-Biased IrDA-Control signal is being transmitted.

[1] TSOP 5000 series (TEMIC) is used for remote controller receiver in this TV.

[2] The packet length is 11 byte from Host Address to CRC flag.

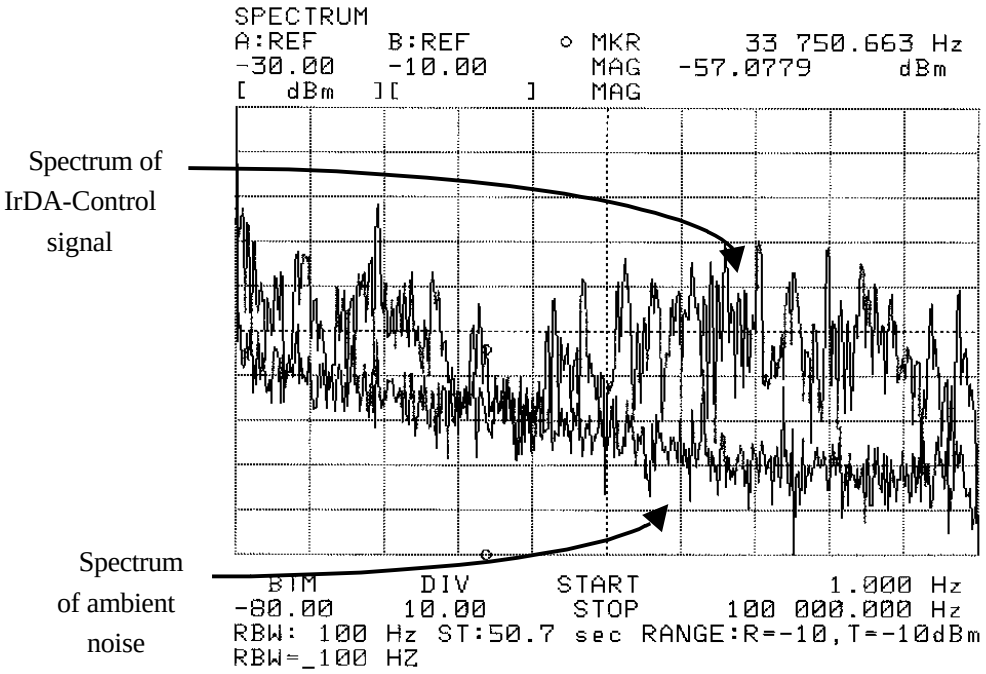


Figure H.3 Spectrum of original IrDA-Control Signal

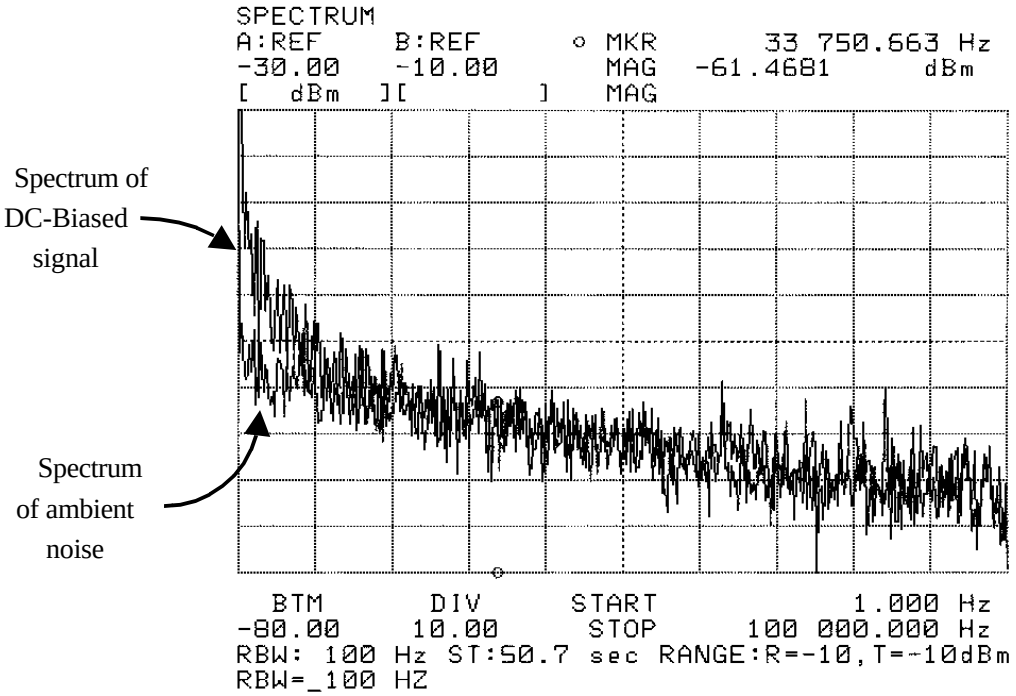


Figure H.4 Spectrum of DC-Biased IrDA-Control Signal

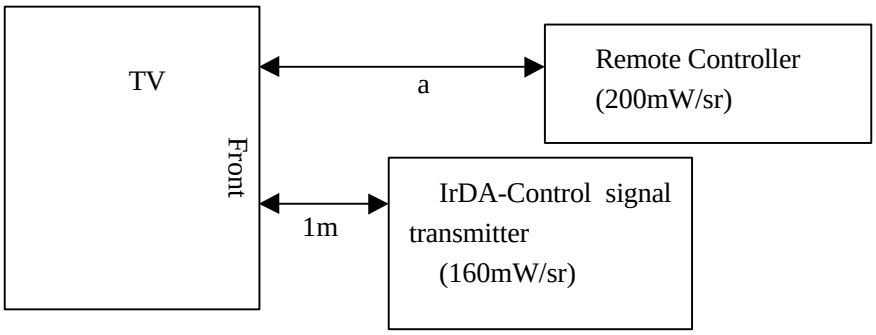


Figure H.5 Environment of the Experiment

Probability of TV controlled successfully

Distance(m)	1 IrDA Control	2 DC-Biased Scheme	3 No ambient noise
1	35%	100%	100%
2	45%	100%	100%
3	45%	100%	100%
4	60%	100%	100%
5	35%	100%	100%
6	35%	100%	100%
7	35%	100%	100%
8	30%	100%	100%
9	45%	100%	100%
10	40%	100%	100%
11	40%	100%	100%
12	5%	100%	100%
13	0%	90%	100%
14	0%	90%	100%
15	0%	80%	95%

Table H.1

3. Open Errata

3.1 Appendix D

MODIFICATION

The errors to be modified in this section are all in relation to Appendix D, in which some examples of packet traffic profiles based on MAC rules are illustrated. However these examples in Final Specification 1.0 contain some errors in both sentences and figures because the descriptions in Appendix D have not completely been updated since the Specification draft version 0.8. These are to be amended here.
Additionally, the Final Specification 1.0 includes some errors, which are caused by operation for editing, such as caption lacking. These are also to be amended.

Figure D.9 (D.2)
According to the description of Table 4.2 (in section 4.2), the maximum number of peripherals at the NCL polling rate is seven under existing only one CL polling rate peripheral.
Therefore, change the first sentence of the paragraph from – When the number of CL peripherals at the CL polling rate is one, the maximum number of peripherals at the NCL polling rate, which can be bound, is twelve – to "When the number of CL peripherals at the CL polling rate is one, the maximum number of peripherals at the NCL polling rate, which can be bound, is seven".
For conformance with above modification, replace with the following figure.

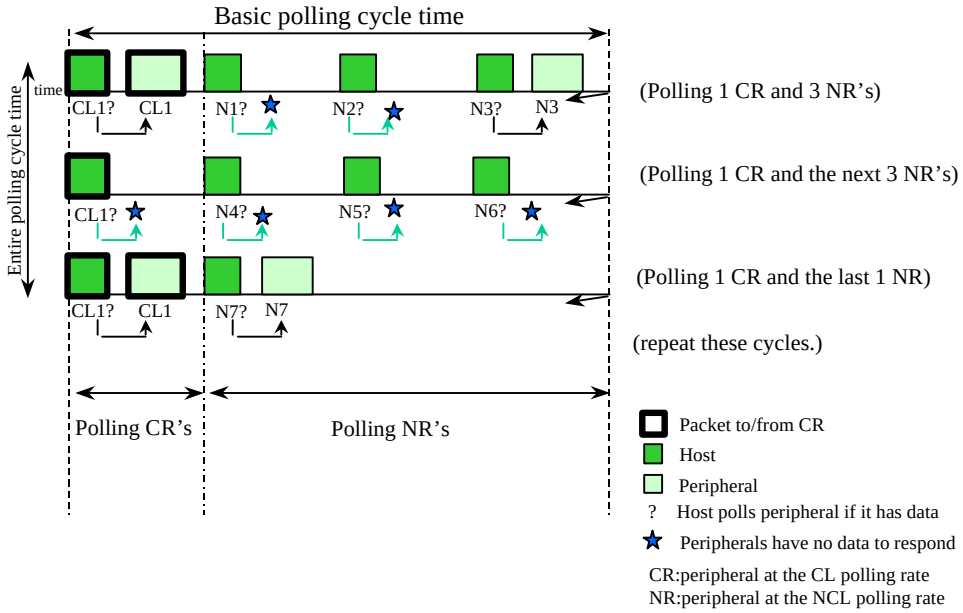


Figure D.9 Example of packet traffic in Mode-1 (1 CL and maximum NCL)

Figure D.10 (D.2)

According to the description of Table 4.2 (in section 4.2), the maximum number of peripherals at the NCL polling rate is six under existing two CL polling rate peripherals.
Therefore, change the first sentence of the paragraph from – When the number of CL peripherals at the CL polling rate is two, the maximum number of peripherals at the NCL polling rate, which can be bound, is eight – to "When the number of CL peripherals at the CL polling rate is two, the maximum number of peripherals at the NCL polling rate, which can be bound, is six".
For conformance with above modification, replace with the following figure.

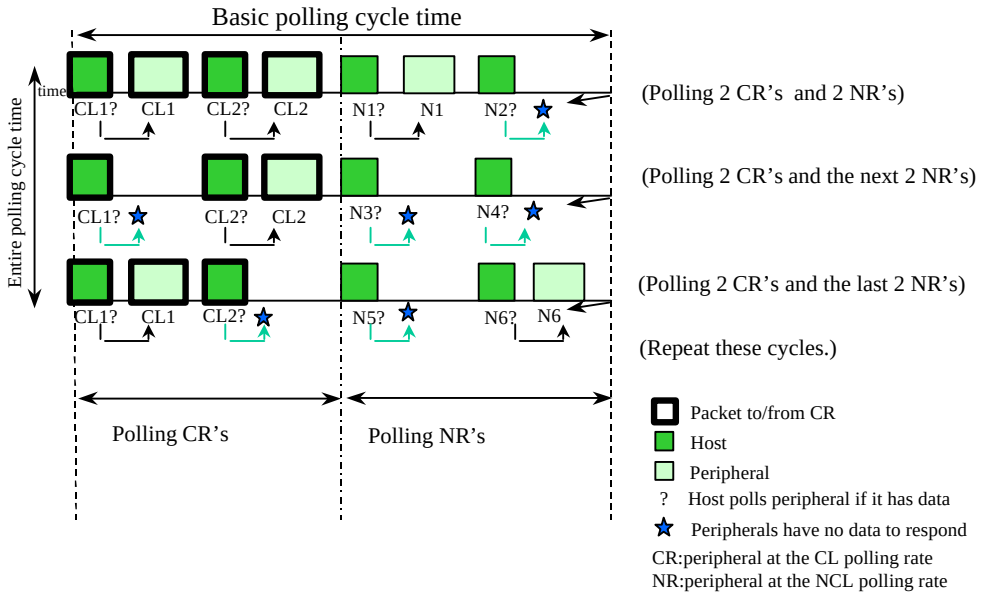


Figure D.10 Example of packet traffic in Mode-1 (2CL and maximum NCL)

Figure D.11 (D.2)
According to the description of Table 4.2 (in section 4.2), the maximum number of peripherals at the NCL polling rate is five under existing three CL polling rate peripherals.
Therefore, change the first sentence of the paragraph from – When the number of CL peripherals at the CL polling rate is three, the maximum number of peripherals at the NCL polling rate, which can be bound, is four – to "When the number of CL peripherals at the CL polling rate is three, the maximum number of peripherals at the NCL polling rate, which can be bound, is five".
For conformance with above modification, replace with the following figure.

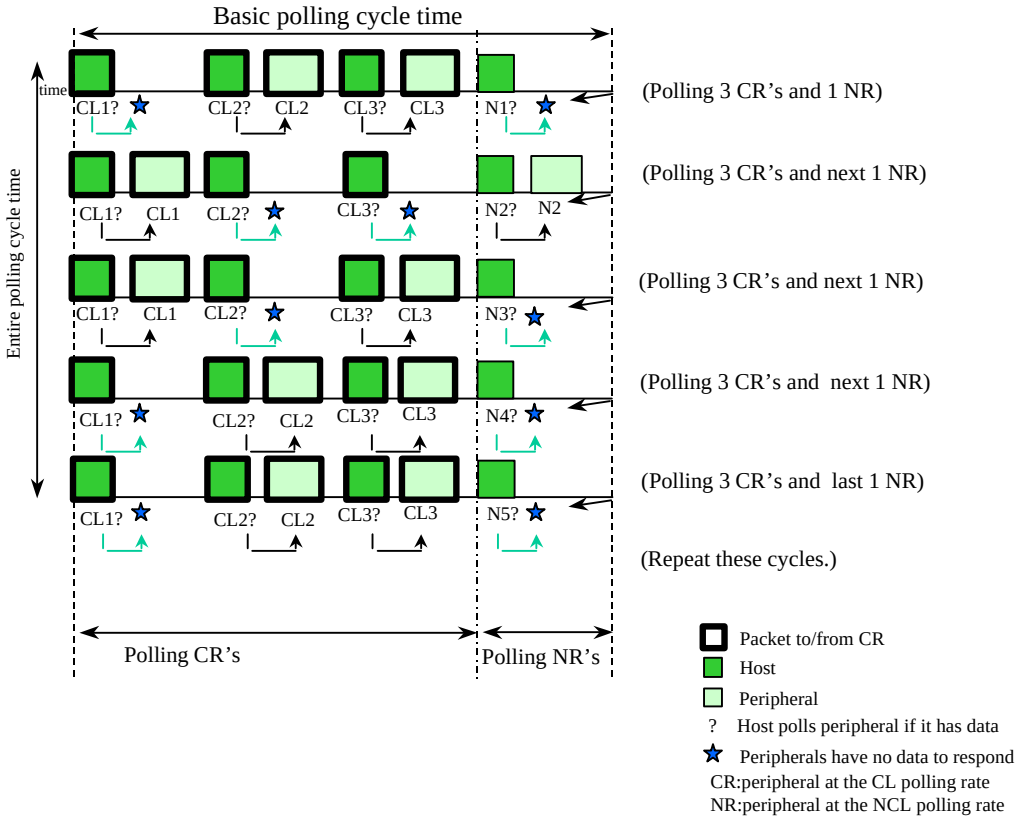


Figure D.11 Example of packet traffic in Mode-1 (3CL and maximum NCL)

Figure D.12 (D.2)

Replace Figure D.12, which lacks caption and some words in the Final Specification 1.0, with the following figure.

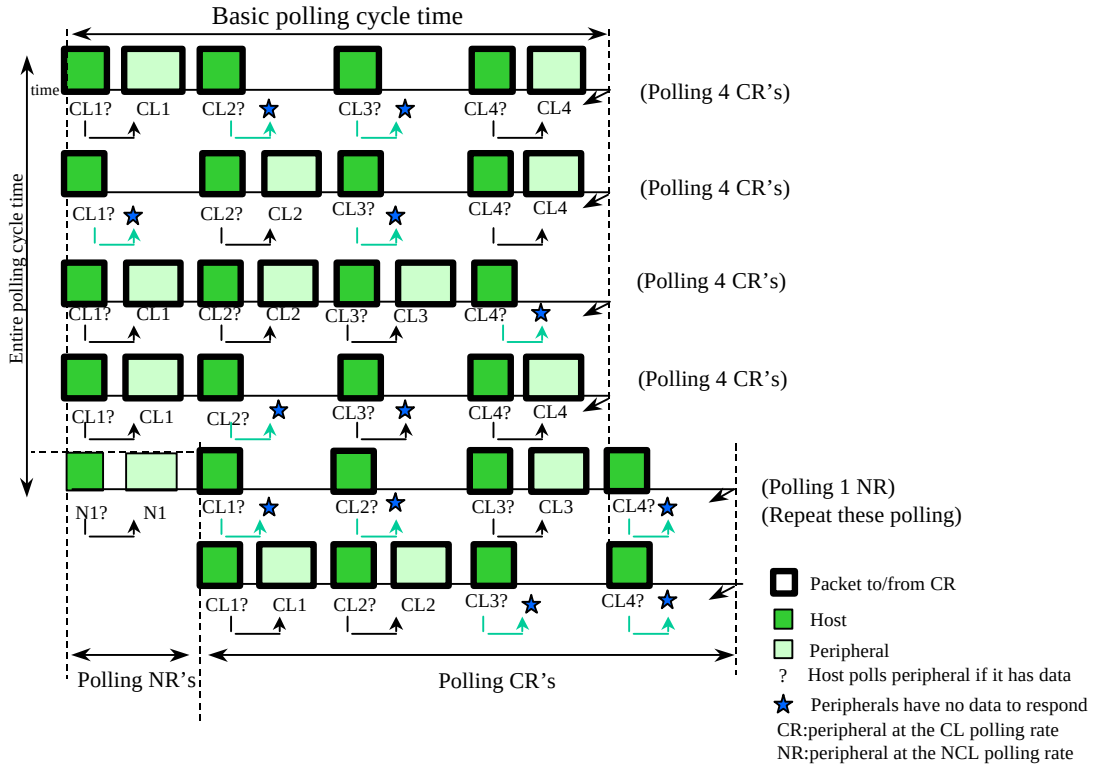


Figure D.12 Example of packet traffic in Mode-1 (4CL and maximum NCL)

Mode-2 (D.3)

Part of the text is lacking in the Final Specification 1.0. The description should be as follows, which are identical to that found in Final version 1.0d, including mention of Figure D.13 with changes from "IRBus" to "IrDA Control".

D.3 Mode-2

In Mode-2, the traffic of IrDA SIR Ver1.1 and the traffic of IrDA Control are performed by turns. Figure D.13 shows packet traffic in Mode-2.

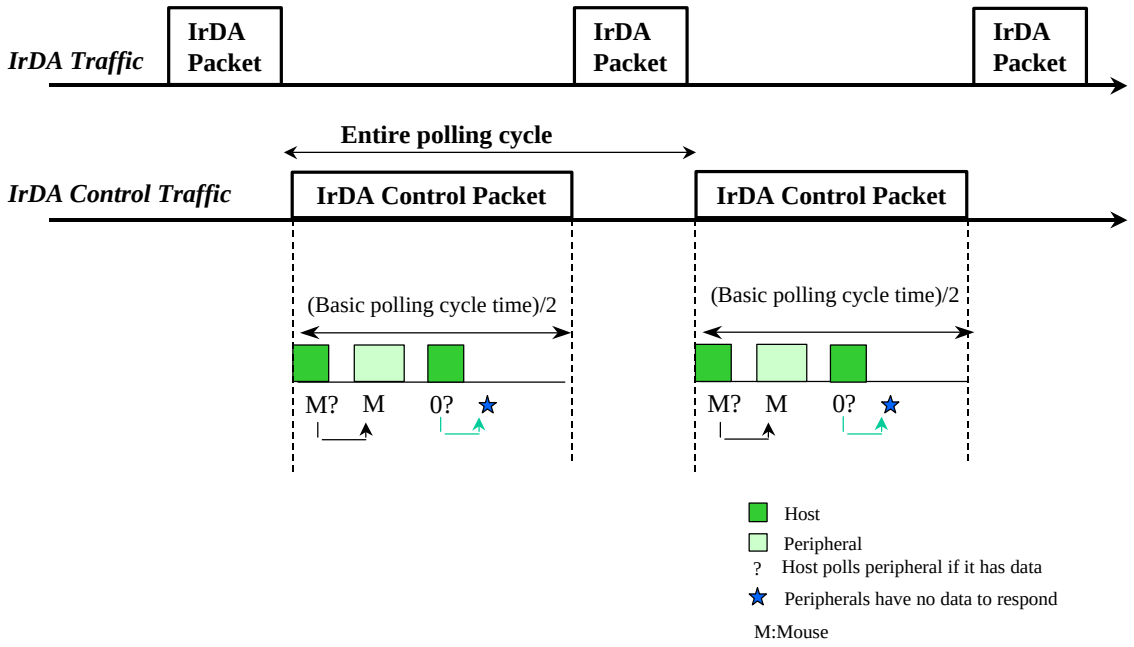


Figure D.13 Example of packet traffic in Mode-2

3.2 Figure E.2

CORRECTION

Figure E.2 (in Appendix E) is to be modified since part of information is lacking in the Final Specification 1.0. Replace with the following figure with changes from “IrBus” to “IrDA Control”.

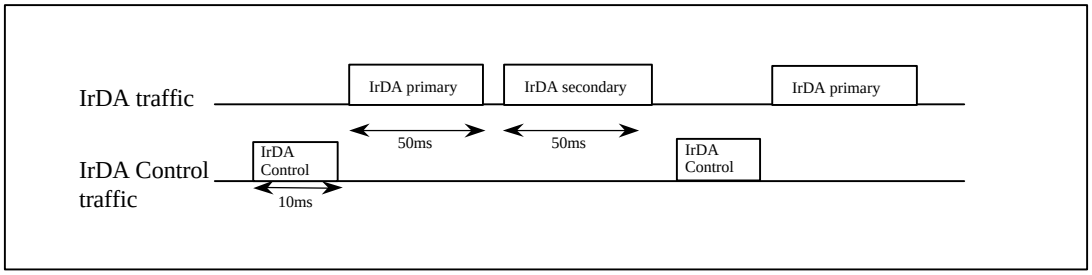


Figure E.2 IR Traffic profile in NRM

3.3 Figure 4.4 (4.3.2)

CORRECTION

According to the description of the item 6 (Binding of Enumerated Peripheral (Inactive Host)), an unbound peripheral waits for up to 69ms for the host hail.
Therefore, change the description in Figure 4.4 (Case where host is sleeping) from - Media Sense for 64 ms - to "Media Sense for 69 ms".

3.4 Unbinding (4.3.3)

CORRECTION

In the description of Host's binding/Unbinding rule, there are two sentences of "5 seconds plus 69ms for NCL peripheral, 30 seconds plus 69ms for NCL peripheral".
These should be "5 seconds plus 69ms for NCL peripheral, 30 seconds plus 69ms for CL peripheral".

Similarly, in the description of Peripheral's binding/Unbinding rule, there is a sentence of "5 seconds for NCL peripheral, 30 seconds for NCL peripheral".
This should be "5 seconds for NCL peripheral, 30 seconds for CL peripheral".

3.5 Sleep Mode (4.4.1)

CORRECTION

The following paragraph in Final Specification 1.0 contains some errors.

Change to Mode-0 from another mode:
- If a host in Mode-1 has received no frames from any NCL peripherals for 5 seconds and no frames from any CL peripherals for 30 seconds.
- If a host in Mode-2 has finished an IrDA SIR ver1.1 data communication, and has received no frame from any NCL peripherals within 5 seconds.

Above paragraph should be as follows.

Change to Mode-0 from another mode:
- If a host in Mode-1 has received no frames from any NCL peripherals for 5 seconds plus 69ms and no frames from any CL peripherals for 30 seconds plus 69ms.
- If a host in Mode-2 has finished an IrDA SIR ver1.1 data communication, and has received no frame from any NCL peripherals within 5 seconds plus 69ms.

3.6 Host Requirements (4.5.1) / Peripheral Requirements (4.5.2)

CORRECTION

The following sentence in Final Specification 1.0 contains an error (a conflict).
All fields are stored in little endian byte order (Most significant byte first) in the MAC payload.

Above sentence should be as follows.
All fields are stored in little endian byte order (Least significant byte first) in the MAC payload.

3.7 Table5.3 (5.3)

CORRECTION

LLC control bit assignment for SET_MODE in Table 5.3 is different from that one in Table 5.2. LLC control bit assignment for SET_MODE in Table 5.3 should be “1010”.

3.8 Example Descriptors (F.9)

CORRECTION

The example descriptors in Final Specification 1.0 contain two errors. In the "Descriptor ID Codes" definition, there is the following description.

idDEVICEEQU 01h

This should be as follows.

idDEVICE EQU 01h

In the IrDA Control Descriptor (IrBus Descriptor), there is the following description.

DB 10000000b; bmLogDevAttributes_1

Because this example is based on HID, this should be as follows.

DB 10001000b; bmLogDevAttributes_1

3.9 Remarks for multiple hosts

CLARIFICATION

Not enough descriptions in relation to multiple hosts are given in the Final Specification 1.0, some remarks to be added. Add the following remark to the end of the first paragraph of section 4.1 (Overview) to clarify the relation with the descriptions in Appendix G.

“In this chapter, basic MAC rules which don’t use the multi-host algorithms are defined only. Detailed multi-host algorithms are described in Appendix G. ”

Additionally in the descriptions of multiple host algorithms, the information how changes the MAC rules under that algorithms is needed. Add the following sentences to the end of Appendix G.4 (Implementation Notes).

“In case this dithering scheme is used, every regulation of “69ms” in chapter 4 is to be changed into “69ms plus 12ms”. 69ms is the time period in which all the bound peripherals can be polled at least once by the host, and there is a possibility that a maximum of 12ms (random value between 0 and 12) delay may be added to the period by applying the scheme.”

3.10 MAC Control field (4.2.3)

CLARIFICATION

There is no description of a peripheral's Bind Timer restarting rule in "MAC Control field (Frame from Host device)". For conformance with the description of the item 1 in "Peripheral's binding/Unbinding rule" (section 4.3.3), add the following sentence for D6 (Bind timer restarted) descriptions in "MAC Control field (Frame from Host device)".

"The peripheral that received polling packet with this bit set from the host device restarts Bind Timer."

3.11 Enumeration (4.3.1)

MODIFICATION

According to the description of the second item of the procedure in section 4.4.2 (Normal Mode (Mode-1)), a host in Mode-1 is guaranteed to poll a bound NCL peripheral and poll to PADD='0x0' at least once every 69ms.
Since the value of 69ms is sufficient, change the second sentence in the first item of the rule in "IrDA Control Enumeration (Inactive Host)" from - On user activity, an unenumerated peripheral waits for up to 1 second - to "On user activity, an unenumerated peripheral waits for up to 69ms ".

3.12 Figure 4.5 (4.5)

MODIFICATION

Figure 4.5 has not been updated since the Specification draft version 0.9, therefore, it does not reflect the MAC rules defined in the Final Specification 1.0 correctly. For conformance with the descriptions of peripheral's MAC rules in the Final Specification 1.0, replace with the following figure.

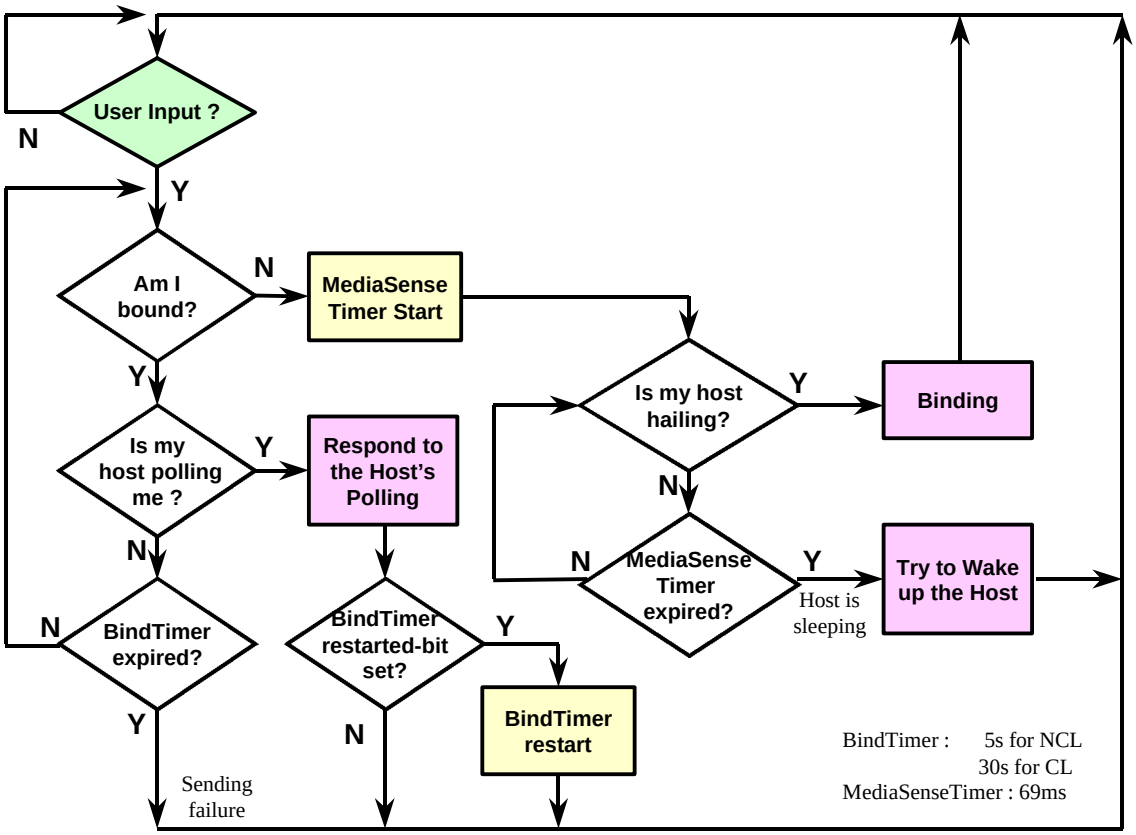


Figure 4.5 Algorithms of IrDA Control Peripheral devices

3.13 Host MAC Protocol Machine (4.7.3)

MODIFICATION

Considering the upper layer protocol (HID LLC) behavior, one response from the peripheral occurs for two polls from the host typically (IN - DATA0/1 - ACK). This means the counter value of response from the peripheral will be half even if the peripheral returns all responses to its polls. Consequently, threshold value related that response counter is to be change to its half to reflect HID LLC characteristics as follows.

NCLPR_TO_CLPR_THRESHOLD	45 (formerly 90)
CLPR_TO_NCLPR_THRESHOLD	35 (formerly 70)

3.14 Protocol Peripheral Machine (4.7.4)

MODIFICATION

There is no proper state transition rule when MACEVT_DATA_REQ event occurs in BOUND state.
Therefore, add MACEVT_DATA_REQ event condition and its handler on MACSTATE_BOUND in MAC_Peripheral_dispatch() as follows.

```
MAC_Peripheral_dispatch(int evtid, void *evtdata)
{
  switch(State){
  …
  case MACSTATE_BOUND:
       switch(evtid){
       case MACEVT_DATA_REQ:
              Action31(evtdata);
              /* Same State*/
              break;
       case PHYEVT_DATA_IND:
              State = Action21(evtdata);
              break;
       case TIMEVT_EXPIRE_IND:
       …
       }
       break;
```

3.15 GET_DESCRIPTOR (5.3.1)

MODIFICATION

Not enough descriptions of the GET_DESCRIPTOR are given in the Final Specification 1.0.
To complement information necessitated at the time of bridging between IrDA Control and USB, append the following descriptions to the end of the paragraph of GET_DESCRIPTOR (LLC Code 0x9).

When using this code, Get_Descriptor ID shown in Table 5.5 must be set at the first byte of LLC payload, whereby the type of Descriptor is discriminated.
In addition, in case of the GET_DESCRIPTOR for String Descriptor and Report Descriptor, the below-listed argument data must be set also at the second byte and following bytes of the LLC payload in order to give the index to appropriate information.

[String Descriptor]

offset	field	size	Value
1	DescriptorIndex	1	Number
3	LangID	2	Number (Lower byte first)

These arguments are the same ones that necessitated when issuing Standard Request of Get_Descriptor (String) on USB, so that HID LLC can use them as well.

[Report Descriptor]

offset	field	size	Value

1	HID LLC Endpoint	1	Number

This argument corresponds to the HID LLC Endpoint number. This is needed to get the proper Report Descriptor associated with HID LLC Endpoint of devices using multiple endpoints in particular.

3.16 SET_MODE (5.3.1) / Figure 5.4 (5.3.3) / (Appendix F.8)

MODIFICATION

Few descriptions of SET_MODE are given in the Final Specification 1.0.
To complement information necessitated at the time of bridging between IrDA Control and USB, specify the behavior of SET_MODE as below.

SET_MODE (5.3.1)
In the Final Specification 1.0, the description of "SET_MODE (LLC Code 0xA)" in section 5.3.1 (LLC Control Field Packet Type Descriptions) is shown below.

SET_MODE (LLC Code 0xA)
This code is used to send commands to a device. Setting the state of a keyboard LED is an example of when it would be used. See Appendices for specific usage.

Above descriptions should be modified as follows.

SET_MODE (LLC Code 0xA)
This code is used to send commands to a device.
When using this code, Report ID at the first byte of LLC payload, Report Type at the second byte and Report Data at the following bytes must be set so that Set_Report request on USB can be properly mapped to HID LLC. The values of these fields depend on underlying USB Class definition.
Setting the state of a keyboard LED is an example of when it would be used. See Appendices for specific usage.

Figure 5.4
For conformance with above modification, replace Fig 5.4 in section 5.3.3 with the following figure.

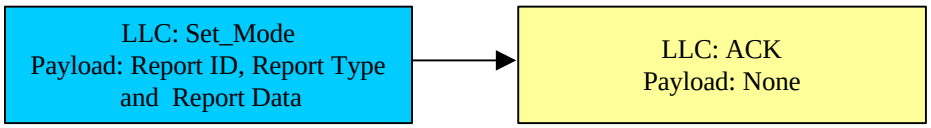


Figure 5.4 Set_Mode sequence

Appendix F.8
For conformance with above modification, replace the packet format of Set_Mode in Appendix F.8 (Additional USB-HID IrDA Control Commands) with the following format.

Source	HADD	MACC	PADD	LLC	DATA

Host	Host Addr	0xC	0xN	0x0A	0x00(Report ID), 0x02(Report Type) ,(LED data)

3.17 GET_STATUS (5.3.1) / Figure 5.3 (5.3.3) / (Appendix F.8)

MODIFICATION

Few descriptions of GET_STATUS are given in the Final Specification 1.0.
To complement information necessitated at the time of bridging between IrDA Control and USB, specify the behavior of GET_STATUS as below.

GET_STATUS (5.3.1)
In the Final Specification 1.0, the description of "GET_STATUS (LLC Code 0xC)" in section 5.3.1 (LLC Control Field Packet Type Descriptions) is shown below.

GET_STATUS (LLC Code 0xC)
This code requests the status of a device. See Appendices for specific usage.

Above descriptions should be modified as follows.

GET_STATUS (LLC Code 0xC)
This code requests the status of a device.
When using this code, Report ID at the first byte of LLC payload and Report Type at the second byte must be set so that Get_Report request on USB can be properly mapped to HID LLC. The values of Report ID and Report Type field are depend on underlying USB Class definition. See Appendices for specific usage.

Figure 5.3
For conformance with above modification, replace Fig 5.3 in section 5.3.3 with the following figure.

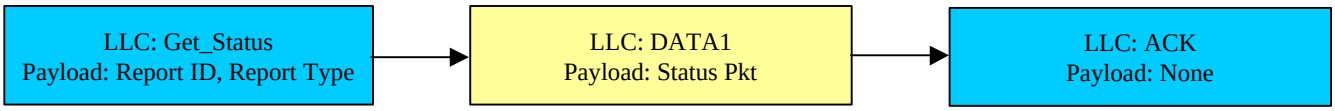


Figure 5.3 Get_Status sequence

Appendix F.8
For conformance with above modification, replace the packet format of Get_Status in Appendix F.8 (Additional USB-HID IrDA Control Commands) with the following format.

Source	HADD	MACC	PADD	LLC	DATA
Host	Host Addr	0xC	0xN	0x0C	0x00(Report ID), 0x02(Report Type)

3.18 IrDA Control Descriptor / Table5.6 (5.3.5)

MODIFICATION

Below-listed new items are to be added at the offset 17 and following bytes in IrDA Control Descriptor. Those items exist within HID Descriptor originally, this is performed to complement information necessitated at the time of bridging between IrDA Control and USB.
These data complements to IrDA Control descriptor must be needed for a peripheral having multiple endpoints in particular.
(Related to Appendix F.9 as well)

offset	field	size	Value	Description
17	bCountryCode_1	1	Number	Hardware target country 1
18	bCountryCode_2	1	Number	Hardware target country 2
19	bCountryCode_3	1	Number	Hardware target country 3
20	wDescriptorLength_1	2	Number	Length of Report Descriptor 1 (Lower byte first)
22	wDescriptorLength_2	2	Number	Length of Report Descriptor 2 (Lower byte first)
24	wDescriptorLength_3	2	Number	Length of Report Descriptor 3 (Lower byte first)

3.19 Configuration Descriptor (F.9)

CLARIFICATION

The field for wTotalLength of Configuration Descriptor on IrDA Control indicates the combined length of Configuration and Report Descriptor. When multiple Report Descriptors exist, total length of Report Descriptors and Configuration itself should be included. The line for it in the part of Configuration Descriptor in Appendix F.9 should be noted as follows:

```
DW    cCONFIG_REQ     ;wTotalLength (Configuration + Report )
```

3.20 Dithering Host Algorithm (G.5)

MODIFICATION

Considering the upper layer protocol (HID LLC) behavior, one response from the peripheral occurs for two polls from the host typically (IN - DATA0/1 - ACK). This means the value for errorcount will be half compared to the value for responsecount even if the peripheral returns all responses to its polls. Consequently, threshold value defined in the Final Specification 1.0 plus 50 % is appropriate to reflect HID LLC characteristics. Change MACRO definitions as follows.

```
DITHER_LOW_THRESHOLD        55 (formerly 5)
DITHER_HIGH_THRESHOLD       75 (formerly 25)
```