PIAFS Microcode for MC68302(IMP) Family User's Manual Rev.1.0

m

PIAFS Controller

1. Overview

PIAFS (PHS Internet Access Forum Standard) is a data link layer protocol to perform data communication over the PHS (Personal Handy Phone System) channel. The PIAFS protocol is divided to two sublayers in order to optimize implementation on the MC68302 family (302 hereafter) devices. Refer to Appendix A-2 for applicable devices. The lower sublayer function is to be implemented by the CP (Communication Processor) and the upper one is to be done by the 68K core processor. The CP will run the proprietary microcode provided by Motorola for the PIAFS controller to implement the lower sublayer. In this implementation where the CP will do receive and transmit PIAFS frames, an external dedicated controller chip may be eliminated and it will bring much benefit to the user in such cost and board space. It will also load off the 68K core processor from those software programs for the overall PIAFS protocol.
This document will describe about operation mode of the CP with the PIAFS microcode (PIAFS controller hereafter).

Processed by the 68K core : Time management, control function such state control and upper layer protocol
Processed by the CP : Sequential processing over serial bit stream

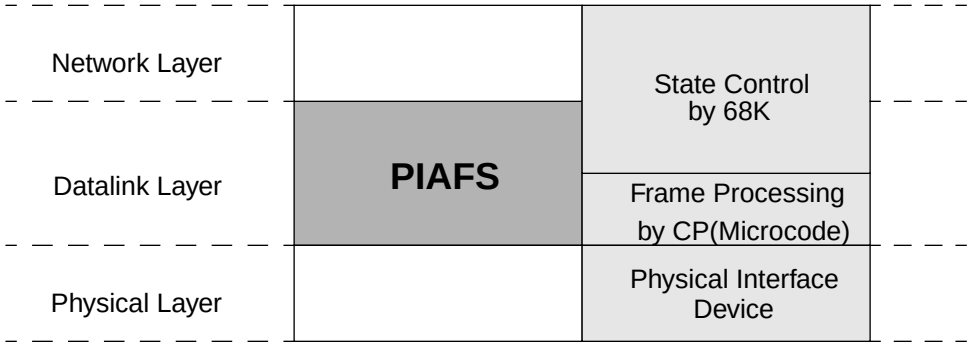


Figure 1. PIAFS Protocol

2. PIAFS Controller Frame Reception and Transmission Processing

Synchronous Frame

Data (24 bits)	SYNC (32 bits)	Data (552 bits)	CRC (32 bits)

Data/Control Frame

Data (608 bits)	CRC (32 bits)

Figure 2. PIAFS Frame

The PIAFS controller supports communication of 640 bits frame unit. It receives all frames of SYNC frame and the other Data/Control frame. It receives those frames assuming there are back-to-back frames incoming after synchronization is established. It searches for the SYNC pattern in a SYNC frame for frame synchronization. Once frame synchronization is established, the PIAFS controller checks the CRC and transfers entire bits of the receiving frame including the CRC to receiving buffer. If it detects a CRC error on the first incoming frame, the user should make it into the state to search for the SYNC pattern again. All received frames are transferred to receiving buffer transparently. Hence these fame types should be distinguished by user program.

The user prepares 608 bits data to transmit. The PIAFS controller transfers 640 bits frame consists of 608 bits data from transmit buffer and 32-bit CRC. Once transmit begins, each frame is assumed to be transmitted continuously. Hence the user needs to prepare an idle frame to sustain the link establishing status.

2.1. Reception Processing

The PIAFS controller searches for the SYNC pattern of 32 bits in receiving bit stream at first. When it detects the SYNC pattern, the following 584 bits(552 bits data and 32-bit CRC) are to be discarded as the first SYNC frame. The PIAFS controller checks CRC and transfers a frame to a buffer from the second frame. That is, the first SYNC frame is used only for establishment of frame synchronization. The PIAFS controller checks CRC on the second frame and continues frame reception under assumption that there are no bits between frames. If a CRC error is detected on the second frame, the user needs to issue ENTER_HUNT_MODE command to make the PIAFS controller into the initial state to search for the SYNC pattern from the first. From the second frame on, the PIAFS controller begins frame reception, checking 32-bit CRC and transferring those receiving data(even if the frame includes error bits) transparently to the data buffer linked to the RxBD. Single RxBD deals with one reception frame.

2.2. Transmission Processing

The user prepares transmit data of 608 bits in the buffer linked to the TxBD. The PIAFS controller calculates 32-bit CRC for the 608 bits data and transmit a frame of 640 bits consists of 608 bits data and the following 32-bit CRC. It continues to transmit frames without inserting idle bits between the frames. So the user should prepare a data buffer and a TxBD for an idle frame even when there is no data to transmit. When receiving the STOP_TRANSMIT command, the PIAFS controller will abort the current frame being transmitted. And when it is given the RESTART_TRANSMIT command, it resumes transmission. Single TxBD deals with one transmission frame.

3. SCC Parameter RAM Memory Map

3.1. Common SCC Parameter RAM

Contents of the parameter RAM are described in the following table. Parameters from $80 to $9B are SCC parameters common to all protocols. Parameters from $9C to $BF are protocol-specific ones for the PIAFS mode. Refer to the *MC68302 User's Manual* for the common SCC parameters. Parameters from $80 to $83 are not used by the PIAFS controller and may not be initialized for the PIAFS mode by the user.

Table 1. Common SCC Parameter RAM Memory Map

Address	Name	Width	Description
SCC Base + $80	RFCR	Byte	Rx Function Code
SCC Base + $81	TFCR	Byte	Tx Function Code
SCC Base + $82	MRBLR	Word	Maximum Rx Buffer Length
SCC Base + $84##		Word	Rx Internal State
SCC Base + $86##		Byte	Reserved
SCC Base + $87##	RBD	Byte	Rx Internal Buffer Number
SCC Base + $88		Long Word	Rx Internal Data Pointer
SCC Base + $8C		Word	Rx Internal Byte Count
SCC Base + $8E		Word	Rx Temp
SCC Base + $90##		Word	Tx Internal State
SCC Base + $92##		Byte	Reserved
SCC Base + $93##	TBD	Byte	Tx Internal Buffer Number
SCC Base + $94		Long Word	Tx Internal Data Pointer
SCC Base + $98		Word	Tx Internal Byte Count
SCC Base + $9A		Word	Tx Temp

##Modified by the CP following a CP or system reset.

3.2. PIAFS-Specific Parameter RAM

Table 2. PIAFS-Specific Parameter RAM Memory Map

Address	Name	Width	Description
SCC Base + $9C##	RCRC_L	Word	Temp Receive CRC Low
SCC Base + $9E##	RCRC_H	Word	Temp Receive CRC High
SCC Base + $A0##	C_MASK_L	Word	CRC Constant Low
SCC Base + $A2##	C_MASK_H	Word	CRC Constant High
SCC Base + $A4##	TCRC_L	Word	Temp Transmit CRC Low
SCC Base + $A6##	TCRC_H	Word	Temp Transmit CRC High
SCC Base + $A8	RES	Word	Reserved
SCC Base + $AA#	**CRCEC**	Word	CRC Error Counter
SCC Base + $AC	RES	Word	Reserved
SCC Base + $AE#	**FRLEN**	Word	Frame Length($50)
SCC Base + $B0#	**FSTSS**	Word	Constant($49)
SCC Base + $B2	RES	Word	Reserved
SCC Base + $B4	RES	Word	Reserved
SCC Base + $B6	RES	Word	Reserved
SCC Base + $B8	RES	Word	Reserved
SCC Base + $BA#	**SYNC2**	Word	SYNC Pattern 2
SCC Base + $BC#	**SYNC3**	Word	SYNC Pattern 3
SCC Base + $BE#	**SYNC4**	Word	SYNC Pattern 4

Initialized by the user program
##Modified by the CP following a CP or system reset.

CRCEC: CRC Error Counter for frame reception. This 16 bits counter is maintained by the CP. This may be initialized to $0 by the user while the channel is disabled.

FRLEN: Byte length of the PIAFS frame. The user should set this $50 for 80 bytes.

FSTSS: Number of bytes from the first SYNC pattern to the head of second frame. The user should set this $49 for 73 bytes.

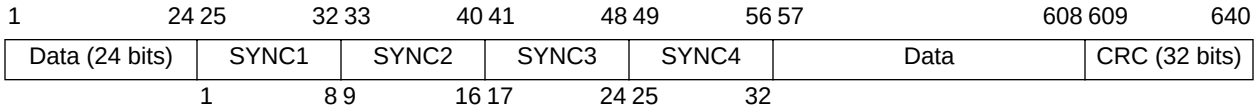
SYNC2: SYNC pattern 2. The SYNC2 should be set upper 8 bits from the 9th bit to the 16th bit of overall 32 bits SYNC pattern. The lower 8 bits should be set to $0.

SYNC3: SYNC pattern 3. The SYNC3 should be set upper 8 bits from the 17th bit to the 24th bit of overall 32 bits SYNC pattern. The lower 8 bits should be set to $0.

SYNC4: SYNC pattern 4. The SYNC4 should be set upper 8 bits from the 25th bit to the 32nd bit of overall 32 bits SYNC pattern. The lower 8 bits should be set to $0.

4. Definition of the SYNC pattern in a SYNC frame

The SYNC pattern of 32 bits is defined by each 8 bits respectively as figured below.



Location to set the SYNC pattern bits

SYNC1 ---> DSRx

SYNC2 ---> Parameter RAM and DSRx

SYNC3 ---> Parameter RAM

SYNC4 ---> Parameter RAM

	15 — 8	7 — 0
DSRx	Bit16 — Bit9	Bit8 — Bit1
	SYNC2	SYNC1

	15 — 8	7 — 0
SYNC2	Bit16 — Bit9	All 0
SYNC3	Bit24 — Bit17	All 0
SYNC4	Bit32 — Bit25	All 0

5. PIAFS Command Set

5.1. Transmit Command

STOP_TRANSMIT command:

After a hardware or software reset and the enabling of the channel in the SCC mode register, the channel is in transmit enable mode and starts polling the first TxBD in the table. The channel STOP_TRANSMIT command disables the transmission of frames on the transmit channel. If the PIAFS controller receives this command during frame transmission, the frame of that message is aborted. The Ready bit in TxBD is not cleared and TBD is not advanced. PIAFS controller stops transmitting any frames after it finishes to transmit all data already transferred to FIFO.

RESTART_TRANSMIT command:

RESTART_TRANSMIT command re-enables the transmission on the transmit channel that has been stopped by STOP_TRANSMIT command. PIAFS controller enables transmission from the present TxBD.

5.2. Receive Command

ENTER_HUNT_MODE command:

After a hardware or software reset and the enabling of the channel by its SCC mode register, the channel is in the receive enable mode and will use the first RxBD in the table. ENTER_HUNT_MODE command is generally used to force the PIAFS receiver to abort reception of the current frame and enter the hunt mode. In the hunt mode, the PIAFS controller continually scans the input data stream searching for the sync pattern. After receiving the command, the cur-

rent receive buffer is closed, and the CRC is reset. Further frame reception will use the next RxBD.

6. PIAFS Error-Handling Procedure

6.1. Transmission Errors

Transmitter Underrun:

When this error occurs, the channel terminates buffer transmission, closes the buffer, sets the underrun(UN) bit in the TxBD, and generates the TXB interrupt (if enabled). The channel will resume transmission after the reception of the RESTART_TRANSMIT command.

TxBD Shortage:

This error occurs when there is no TxBD whose ready bit is set during transmission. The transmission is stopped and the SHT bit in the event register is set.

6.2. Reception Errors

Overrun Error:

The PIAFS controller maintains an internal FIFO for receiving data. When a receive FIFO overrun occurs, the channel writes the received data byte to the internal FIFO over the previously received byte. The previous data byte and the frame status are lost. The channel closes the buffer with the overrun (OV) bit in the RxBD set and generates the RXF interrupt (if enabled). RBD is not advanced. If the E bit in the corresponding RxBD is set, the receiver enters the hunt mode and resumes frame reception.

CRC Error:

CRC is checked upon each frame reception. When this error occurs, the channel sets the CR bit in the RxBD, and generates the RXF interrupt (if enabled). The channel also increments the CRC error counter (CRCEC). Even if the channel receives a frame with a CRC error, it continues to receive the next frame.

Busy Condition:

This error occurs when a frame was received and discarded due to lack of buffers. When this error occurs, the BSY bit in the event register is set and generates the interrupt (if enabled). The reception is stopped.

7. PIAFS Mode Register

The PIAFS mode is function mode only for the PIAFS protocol. Specific bits in the PIAFS mode register should be set as described below. And all the other bits should be set to zero.

15	14	13	12	11	10	9	8
-	-	-	PIAFS1	-	-	PIAFS0	-
7	6	5	4	3	2	1	0
-	-	DIAG1	DIAG0	ENR	ENT	MODE1	MODE0

PIAFS1 = Set to "1"
PIAFS0 = Set to "1"
DIAG1-0 = Refer to *MC68302 User's Manual*.
ENR, ENT = Refer to *MC68302 User's Manual*.
MODE1-0 = Set to "11"

8. PIAFS Receive Buffer Descriptor (RxBD)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OFFSET + $0	E	-	W	-	-	-	-	-	-	-	-	-	-	OV	CR	-
OFFSET + $2	Data Length															
OFFSET + $4	-	FC			All 0				Rx Buffer Pointer (Upper 8 bits)							
OFFSET + $6	Rx Buffer Pointer (Lower 16 bits)															

E - Empty

0 = The data buffer reception associated with this RxBD has been filled with received data, or data reception has been aborted due to an error condition. The user is free to examine or write to any fields of the RxBD.

1 = The data buffer associated with RxBD is empty. This bit signifies that the RxBD and its associated buffer are available to the PIAFS controller. The user should not write to any fields of this RxBD after it sets this bit. The empty bit will remain set while the PIAFS controller is currently filling the buffer with received data.

W - Wrap

0 = This is not the last RxBD in the table.

1 = This is the last RxBD in the table. After this buffer has been used, CP will receive incoming data into the first RxBD in the table. The numbers of RxBD in the table can be programmed to maximum eight.

CR - CRC Error

This frame contains a CRC error. Both receiving data and its CRC are always written on receiving buffer.

OV - Overrun

A receiver overrun occurred during frame reception.

14,12,11,10,9,8,7,6,5,4,3,0 - Reserved for future use.

Data Length

The data length is the number of octets written to this RxBD's data buffer by PIAFS controller. It is written by CP once as the RxBD is closed. In PIAFS mode, because single RxBD is used for one frame reception, $50 is written by CP. The data length includes 32-bit CRC.

Rx Buffer Pointer

The receive buffer pointer always points to the first location of the associated data buffer. Buffer may reside in either internal or external memory but the internal memory access is also through 68K data bus. Rx buffer pointer must be even. Among 32bits, the lower 24 bits are effective for address and the upper 3 bits are used for setting the function code of bus cycle.

9. PIAFS Transmit Buffer Descriptor (TxBD)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OFFSET + $0	R	-	W	-	-	-	-	-	-	-	-	-	-	-	UN	-
OFFSET + $2	Data Length															
OFFSET + $4	-	FC			All 0				Tx Buffer Pointer (Upper 8 bits)							
OFFSET + $6	Tx Buffer Pointer (Lower 16 bits)															

R - Ready

0 = This buffer is not currently ready for transmission. If this bit is one, the user is free to manipulate this TxBD (or its associated buffer). The CP clears this bit after the buffer has been fully transmitted or after an error condition has been encountered.

1 = The data buffer, which has been prepared for transmission by the user, has not been transmitted yet or is now being transmitted. No fields of this TxBD may be written by the user once this bit is set.

W - Wrap

0 = This is not the last TxBD in the table.

1 = This is the last TxBD in the table. After this buffer has been used, CP will return to the first TxBD. The numbers of TxBD in the table can be programmed to maximum eight.

UN - Underrun

The PIAFS controller encountered a transmitter underrun condition while transmitting the associated data buffer.

14,12,11,10,9,8,7,6,5,4,3,2,0 - Reserved for future use.

Data Length

Transmit data length should be set to $4C excluding its CRC.

Tx Buffer Pointer

The transmit buffer pointer always points to the first location of associated data buffer. Buffer may reside in either internal or external memory but the internal memory access is also through 68K data bus. Rx buffer pointer must be even. Among 32 bits, the lower 24 bits are effective for address and the upper 3 bits are used for setting the function code of bus cycle.

10. PIAFS Event Register

PIAFS event register is a memory-mapped register that may be read at any time. A bit is cleared by writing a one while writing a zero does not affect a bit value. More than one bit may be cleared at a time. This register is cleared at reset.

7	6	5	4	3	2	1	0
-	-	-	SHT	-	BSY	TXB	RXF

SHT - Shortage

Transmission was stopped due to lack of transmit buffers.

BSY - Busy Condition

A frame was received and discarded due to lack of buffers.

TXB - Tx Buffer

A buffer has been transmitted on the PIAFS channel.

RXF - Rx Frame

A frame has been received on the PIAFS channel. Receiving status is written on RxBD.

11. PIAFS Mask Register

The SCC mask register (SCCM) is referred to as the PIAFS mask register when the SCC is operating as PIAFS controller. It is an 8-bit read/write register that has the same bit formats as the PIAFS event register. If a bit in the PIAFS mask register is a one, the corresponding interrupt in the event register will be enabled. If the bit is zero, the corresponding interrupt in the event register will be masked.This register is cleared at reset.

12. PIAFS Microcode From RAM Initialization Sequence

1. Load the PIAFS microcode(S-record file) into the internal DPRAM.
2. Set the microcode enable bit. The bit location varies on each device.
   - 302,EN302 — Write $0001 to address $F8 in supervisory space.
   - PM302,LC302 — Set the bit 23 of the SCR(System Control Register) at $F4.
3. Write a software reset command to the CR(CP Command Register).
4. Continue with the normal initialization sequence.

Appendix

A-1. Option of Physical Interface according to each application

It is expected two cases for being connected with the PIAFS controller. One is for wireless interface being connected with PHS baseband chip and the other is for ISDN B-ch interface. A 302-based system is applicable for the both interface.

A-1.1. PHS Wireless Interface

There are some interfaces to connect with PHS. In the case of connection with a baseband chip directly, that is 32kbps continuous communication, NMSI mode should be used along with CTS signal control as transmission timing. Refer to *MC68302 User's Manual* for the timing in detail. In the case of connection with time-division multiplexing interface by 4 bits, the multiplex interface mode should be used. PCM, IDL or GCI is used suitable for timing specification. SIMASK register can be used for IDL or GCI interface in the same way as ISDN B-channel interface as described below.

A-1.2. ISDN B-channel Interface

IDL or GCI mode can be used for ISDN B-ch interface. 4 bits of 8 bits (B-Channel) are used for communication channel by SIMASK register configuration.

A-2. Applicable Devices for PIAFS Microcode

PIAFS microcode for the 302 family is designed to be used on MC68302, MC68LC302, MC68PM302, MC68EN302, MC68356, MC68DP356. This microcode can't be used on MC68SC302 and MC68QH302's QH mode. Operation of the PIAFS controller has not been looked into on MC68QH302's normal mode yet.