MSRW-206 Manual Swipe Magnetic Card

Reader/Writer Encoder

High Coercivity / Low Coercivity

Dual Mode

PROGRAM MANUAL

VERSION 1.1.5 August 18th, 2007

INDEX

Chapter 1. Introduction

Thank you very much for purchasing the MSRW-206 product series of manual swipe magnetic card reader writer/encoder. It can be used in many business applications from Point of Sale system, custom membership card, to banking and credit card services, etc.

The MSRW-206 series is designed to read and write track 1/2/3 magnetic cards in Hi-Co or Lo-Co mode select by software command.

Features:

- Read/Write format conforms to ISO 7811/1~6
- Manual swipe to read and/or write card with RS232 output
- (Optional) USB Interface (purchase the USB-serial adapter)
- Read/Write High & Low Coercive force of magnetic stripe;
- 5~35ips operational swipe speed data write
- 5~55ips operational swipe speed data read
- +24VDC+/-10%, 2.0A Max, universal power adapter supplied.
- Write then verify single, dual track of data in one pass
- High/Low Coercivity encoding function

Chapter 2. Packing List

1. **MSRW-206 Hi-Co(2750 oe.) /Lo-Co (300 oe.) magnetic card reader / writer / encoder**
2. **AC/DC Switch power supply, AC 110~240V, 2A MAX. (NO Power Cord)****

   **** you can use a desktop computer type power cord.)**
3. **2 Blank Gemplus CR80 test card, Hi-Co 2750 oe.**
4. **Free software download link.**

http://www.msrw206.com/software/demo.exe

USB adapter driver: (only for notebook without serial port).

http://www.msrw206.com/software/usbdrv9720.rar

Chapter 3. Technical specifications

Specification	
Power Requirement:	+24VDC ±5%
Consumption:	Typical 450mA Max. 900mA plus for each writing track
Power supply:	External Switching power 24V/2.0A
Communication:	Standard RS232 signal levels
Body Material:	ABS
Dimension and Weight:	210.1 (L) x 60 (W) x 60 (H) mm. Approx. 3 LB including packing
Card Handling Method:	Manual swipe, Single direction
Environment:	Operation : -10 to +55°c, 10 to 85% humidity Storage : -30 to +70°c, 10 to 90% humidity
Coercive force:	Hi/Lo Read/Write 2750/300 Oe.
Read card:	Track 1:210 bpi, Track 2:75 bpi,
Read speed:	STD card : 5~55ips, Jitter ±15% : 5~50ips, Amp.60% : 5~50ips
Write jitter:	Interval < ±10%, Sub-interval < ±12%
Card thickness:	0.76 1.2mm
Error rate:	Read < 0.5%, Write < 0.8%
Head life:	Min. 500,000 swipes for both read/write head

Chapter 4. Setup

Because this card reader / writer has its internal CPU, it does not use any computer's CPU processing power, it only send/receive commands and data to/from computer via serial communication. It can be connected to many computer devices (a desktop, notebook, POS, Palm, even a dumb VT-100 terminal) with custom made software.

HARDWARE SETUP:

1. Power Off your computer.

2. 

Connect the card writer's RS-232 9-PIN female data connector to your computer's serial port COM:1/COM:2 (which is a 9-Pin male RS-232 connector.)

** If your computer has NO serial ports, you have to buy a USB-to-Serial adapter with the right software driver CD.

3. Connect power cord to the switching power supply and AC receptacle (110 V ~ 240 V).

5. Connect the switching power supply's output DC-24V plug to card writer's power block, the machine will do a power-on-test.

6. If the power-on-test finished okay, the Green LED on the card writer will become steady Green, it is ready to work.

OPERATING MODE:

-ISO **:** for all bank cards. Credit card, gift card, store card, membership card…

-RAW DATA **:** read/write data in HEX mode: door card, special card…

AAMVA: DMV California State driver's license mode.

DATA Limit:

◇max. number of characters recording ：

Track 1 76 characters

Track 2 37/104 characters (75 BPI/ 210 BPI), numbers only.

Track 3 104 characters, numbers only.

SOFTWARE SETUP:

Because this is an industry standard card writer, you can use several industry standard software to operate it. Please check with the software vendors directly.

Chapter 5. Software

There are some software on the market which may operate this machine, when you use these software, please select the model as “MSR206”.

** You must shutdown following software that using serial ports:

- Palm Sync
- GPS
- Modem/FAX.
- If “Not Found xxx206 !” appears, please check the serial port plug connection, and check the power supply.
- When the program is first executed, you can enter a password.
- In software’s main window you can execute all functions by clicking the appropriate buttons and following the popup messages.
- Use Hi-Co card in Hi-Co writing mode, please use Lo-Co card in Lo-Co writing mode. Otherwise, the write-operation will fail. All bank cards after 2002 are Hi-Co; Most membership cards are Lo-Co.

If you are a programmer, you can use following serial commands to control the card writer. You can use C, C++, VB…etc.

TYPICAL CARD READER / WRITER COMMAND /RESPONSE

6.COMMAND and RESPONSE

This section gives detailed description of commands to the MSRW206 and the Corresponding response from MSRW206.

Notional conventions:

<ESC>	Control character named
[sname]	Special string named sname, meaning can be found in section 7. i.e.[Data Block] [Status Byte] [Select Byte] etc.
X	Standard ANSI character

Command Description:

1. Command **:RESET**
   Command code :<ESC>a
   Hex code :1B 61
   Response :none
   Description :This command reset the **MSRW206** to Initial state.

2. Command **:READ**
   Command code :<ESC>r
   Hex code :1B 72
   Response : [Data Block] <ESC> [Status Byte]
   Description :This command request **MSRW206** to read a card Swiped and respond with the data read.

3. Command **:WRITE**
   Command code :<ESC>w [Data Block]
   Hex code :1B 77 [Data Block]
   Response :<ESC> [Status Byte]
   Description :This command request **MSRW206** to write the Data Block into the card swiped.

4. Command **:Communication test**
   Command code :<ESC>e
   Hex code :1B 65
   Response :<ESC> y [1B][79]
   Description :This command is used to verify that the Communication link between computer and **MSRW206** is up and good.

5. Command **:AII LED off**
   Command code :<ESC><81>
   Hex code :1B 81
   Response :none
   Description :This command is used to turn all the LEDs.

6. Command **:AII LED on**
   Command code :<ESC><82>
   Hex code :1B 82
   Response :none
   Description :This command is used to turn on all the LEDs.

7. Command **:GREEN LED on**
   Command code :<ESC><83>
   Hex code :1B 83
   Response :none
   Description :This command is used to turn on Green LED.

8. Command **:YELLOW LED on**
   Command code :<ESC><84>
   Hex code :1B 84
   Response :none
   Description :This command is used to turn on the Yellow LED.

9. Command **:RED LED on**
   Command code :<ESC><85>
   Hex code :1B 85
   Response :none
   Description :This command is used to turn on the Red LED

10. Command **:Sensor test**
    Command code :<ESC><86>
    Hex code :1B 86
    Response :<ESC> 0 (1B 30) if test ok
    Description :This command is used to verify
    that the card sensing circuit of **MSRW206** is
    working **p**roperly **MSRW206** will not response until
    a card is sensed or receive a RESET command.

11. Command **:Ram test**
    Command code :<ESC><87>
    Hex code :1B 87
    Response : <ESC> 0 (1B 30) ram test ok
    <ESC> A (1B 41) ram test fail
    Description :This command is used to request **MSRW206** to
    Perform a test on its on board RAM.

12. Command **:Set leading zero**
    Command code :<ESC>z [leading zero of track 1&3][leading zero
    of track 2]
    Hex code :1B 7A [00~ff][00~ff]
    Response :<ESC> 0 (1B 30) set ok
    <ESC> A (1B 41) set fail
    Description :This command is used to set how many
    leading zeros will be written before the
    card data starts, and the space should
    calculated as
    [leading zero] X25.4/BPI (75or210) =mm
    Default setting of leading zero:[3D][16]
    TK1&TK3: [3D]means leading zero=61
    Tk2: [16]means leading zero=22

13. Command **:Check leading zero**
    Command code :<ESC>I
    Hex code :1B 6C
    Response : 1B [00~ff][00~ff]
    Description :This command is used to ask MSRW206
    The present setting number of leading zeros.

14. Command **:Erase card**
Command code :<ESC> c [Select Byte]
Hex code :1B 63 [Select Byte]
Response : <ESC> 0 [1B][30] command Select Byte ok
<ESC> A [1B][41] command Select Byte fail
Description :This command is used to erase the card data when Card swipe.
*[Select Byte]format :
00000000 : Track 1 only
00000010 : Track 2 only
00000100 : Track 3 only
00000011 : Track 1 & 2
00000101 : Track 1 & 3
00000110 : Track 2 & 3
00000111 : Track 1,2 & 3

15. Command **:Select BP I**(only for TK2)
Command code :<ESC> b [Density]
Hex code :1B 62 [D2 or 4B]
Response : <ESC> 0 [1B][30] select ok
<ESC> A[1B][41] select fail
Description :This command is used to select the density of TK2.
[D2] : TK2 BPI=210 [4B] :TK2 BPI=75

16. Command **:Read raw data**
Command code :<ESC> m
Hex code :1B 6D
Response :[Raw Data Block]<ESC>[Status Byte]
Description :This command requests MSRW206 to
Read a card swipe but send without ASCII decode.
Refer to [Raw Data Block] & [Raw Data] format.

17. Command **:Write raw data**
Command code :<ESC> n [Raw Data Block]
Hex code :1B 6E [Raw Data Block]
Response :<ESC> [Status Byte]
Description :This command requests MSRW206 to
Write raw Data Block into the card swiped.
Refer to [Raw Data Block] & [Raw Data] format.

18. Command **:Get device model**
Command code :<ESC>t
Hex code :1B 74
Response :<ESC>[Model] S
Description :This command is used to get the model of MSRW206.There are four models Model 1,2,3,&5

Models	Description
MSRW206-1	Track 2
MSRW206-2	Track 2 & 3
MSRW206-3	Track1, 2 & 3
MSRW206-5	Track 1 & 2

19. Command **:Get firmware version**
Command code :<ESC> v
Hex code : <ESC> 76
Response : <ESC> [version]
Description :This command can get the firmware version of MSRW206.

*[version] is a 5 bytes version number, format is “REV?X.XX”
MSRW206 ? = O
MSRW206HC ? = H
MSRW206HL ? = U

20. Command **:Set BPC**
Command code :<ESC> o [tk1bit][tk3bit]
Hex code : <ESC> 6F [05-08][05-08][05-08]
Response : <ESC> 30 [tk1bit][tk2bit][tk3bit]
Description :This command is used to set the bit per Character of every track.

21. Command **:Set Hi-Co**
Command code :<ESC> x
Hex code :1B 78
Response :<ESC>0
Description :This command is used to set MSRW206HL status to write Hi-Co card.

22. Command **:Set Low-Co**
Command code :<ESC> y
Hex code :1B 79
Response :<ESC> 0
Description :This command is used to set MSRW206HL status to write Low-Co card.

23. Command **: Get Hi-Co or Low-Co status**
Command code :<ESC> d
Hex code :1B 64
Response :<ESC> H ------to write Hi-Co
<ESC> L------to write Low-Co
Description :This command is to get the MSRW206 write status

7. DATA FORMAT

- **[Data Block]** format:

	Start Field	RW Data Field	Ending Field
Command code	<ESC> s	[Card data]	?<FS><ESC>[Status]
Hex code	1B 73	[Card data]	3F 1C 1B [Status]

- **[Card data]** format:

	Card Data
Char Code	<ESC>1[string1] <ESC>2[string2] <ESC>3[string3]
Hex Code	1B 01[string1] 1B02 [string2] 1B 03 [string3]

- **[Status Byte]** format:

Status	description	HEX	ASCII
OK	if read, write or command ok	30h	0
Error	Write or read error	31h	1
	Command format error	32h	2
	Invalid command	34h	4
	Invalid card swipe when in write mode	39h	9

*Note:

1. When [Status Byte] equal 39h means card moving error
2. None available and none data tracks will not be transmitted when swipe of card. For example, when read card with data encoded on track 2 only for MSRW206-5,it will transmit data like **1b 73 1b 01 1b 02 [string] 3f 1c,**for no data on track 1 so it shown 1B 01 only.

[Raw Data Block] Format:

	Start Field	R/W Data Field	Ending Field
Command code	<ESC> s	[Raw data]	?<FS><ESC>[Status]
Hex code	1B 73	[Raw data]	3F 1C 1B [Status]

- **[Raw Data]** Format:

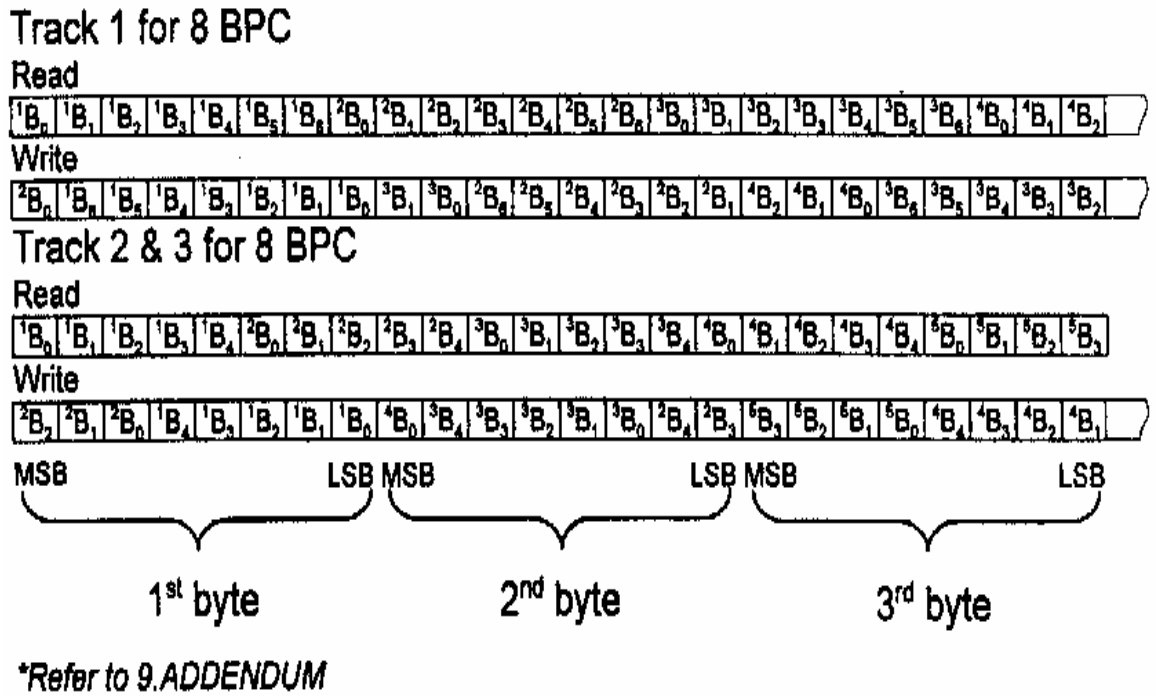
	Raw Data
Char Code	<ESC>1 [L1][string1] <ESC>2 [L2][string2] <ESC>3 [L3][string3]
Hex Code	1B 01 [L1][string1] 1B 02 [L2][string2] 1B 03 [L3][string3]

Note:

1. [L1],[L2],[L3]is the length of [string1], [string12], [string3]
2. None available and none data tracks will not output when swipe of card.
   For example, when read card (encoded data on track 2 only) on MSRW206-5, it will transmit data like

   1B 73 1B 01 00 1B 02 [L2][string]3F 1C

- **[Raw Data]** bit orientation:



*Refer to 9.ADDENDUM

8.COMMUNICATION SEQUENCE

The examples below assumes data on track 1,2&3 to be 01,23,45 respectively

8.1 MSRW 206 INITIALIZATION

HOST	Direction	MSRW206
Command code:<ESC>a HEX code:[1B][61]	======➔	(Reset)
Command code:<ESC>e HEX code:[1B][65]	(Serial port test) ======➔	
	➔======	Command test ACK:<ESC>y HEX code:[1B][79]
Command code:<ESC>a HEX code:[1B][61]	======➔	(Reset)

8.2 WRITE DATA TO MSRW 206

HOST	Direction	MSRW206
Command code: <ESC>w<ESC>s<ESC>[01]01 <ESC>[02]23<ESC>[03]45?<FS> HEX code: [1B][77][1B][73][1B][01][30][31][1B] [02][32][33][1B][03][34][35][3F][1C]	(write command) ====➔	
	(status ACK) ➔====	(Wait until swipe card) Command ACK:<ESC><status> HEX code:[1B][status] Status=[30] no error Status=[31]~[3F] if error

8.3 READ DATA TO MSRW 206

HOST	Direction	MSRW206
Command code:<ESC>r HEX code:[1B][72]	====➔	
	(status ACK) ➔====	(Wait until swipe card) Command ACK: <ESC>s<ESC>[01]%01?<ESC>[02];23? <ESC>[03];45??<Fs><ESC>[status] HEX code: **[1B][73][1B][01][25][30][31][3F][1B][02][3B][32][33][3F][1B][03][3B][34][35][3F][3F][1C][1B] [status]** Status=[30] ok Status=[31]~[3F] if error

*[xx]=HEX Code XX

9.ADDENDUM

(I) WRITE DATA TO MAGNETIC CARD

THE WRITE command:

Command	WRITE
Command code	<ESC> w **[Data Block]**
Hex code	1B77 **[Data Block]**
Response	<ESC> [Status Byte]
Description	This command request MSRW106 to write the Data Block into the card swiped.

[Data Block] format:

	Start Field	R/W Data Field	Ending Field
Command code	<ESC>s	**[card data]**	?<FS>
Hex code	1B 73	**[card data]**	3F 1C

[card data] format:

	card data
Char. code	<ESC>[01][string] <ESC>[02][string] <ESC>[03][string3]
HEX code	1B 01 [string1] 1B 02 [string2] 1B 03 [string3]

As an example the following information will be write to the card:

Track1:%ABC123?
Track2:12345?
Track3:12345?

HOST	DIRECTION	MSR106/ MSRW206
Command code: <esc>w<ESC>s<ESC>[01]ABC123 <ESC>[02]12345<ESC>[03]12345?<FS> HEX code: [1B][77][1B][73][1B][01][41][42][43][31][32][33][1B][02][31][32][33][34][35][1B][03][31] [32][33][34][35][3F][1C]	(write command) ====➔	
After send command to MSRW 206	Yellow LED on, then swipe card	Write data to the magnetic card
	(Status ACK)	(Wait until swipe card) Command ACK:<ESC>[status] HEX code:[1B] [status] Status=[30] no error Status=[31] ~ [3F] if error

(II) WRITE RAW DATA TO MAGNECTIC CARD

Converting Card Data Information to Hexadecimal for the Binary Write Function

Converting track one ASCII information into HEX

						0	1	2	3
					B5	0	0	1	1
	B3	B2	B1	BO	B4	0	1	0	1
0	0	0	0	0		(sp)	0	2	P
1	0	0	0	1		!	1	A	Q
2	0	0	1	0		“	2	B	R
3	0	0	1	1		#	3	C	S
4	0	1	0	0		$	4	D	T
5	0	1	0	1		%	5	E	U
6	0	1	1	0		&	6	F	V
7	0	1	1	1		‘	7	G	W
8	1	0	0	0		(	8	H	X
9	1	0	0	1		)	9	I	Y
A	1	0	1	0		*	:	J	Z
B	1	0	1	1		+	;	K	[
C	1	1	0	0		`	<	L	\
D	1	1	0	1		,	=	M	]
E	1	1	1	0		.	>	N	^
F	1	1	1	1		/	?	O	-

Converting track two and three ASCII information into HEX

Data	p	B3	B2	B1	B0
0	1	0	0	0	0
1	0	0	0	0	1
2	0	0	0	1	0
3	1	0	0	1	1
4	0	0	1	0	0
5	1	0	1	0	1
6	1	0	1	1	0
7	0	0	1	1	1
8	0	1	0	0	0
9	1	1	0	0	1
:	1	1	0	1	0
:(*)	0	1	0	1	1
<	1	1	1	0	0
=	0	1	1	0	1
>	0	1	1	1	0
?(*)	1	1	1	1	1

As a example the following information will be write to the card:

Track1:%ABC123?
Track2:12345?
Track3:12345?
We use three different data bits to write raw data on the cards. The procedures are listed as below:

08,08,08 BITS

1. Set each track as 08.
   First of all, set BPC command:
   1B,6F,08,08,08

2. Present the information to the card encoder, as follows:

UIC	UIC
Start Field	1B6E 1B73
Track1 header	1B01
Length	08
Track1 data	C5B07814954E3E2A
Track header	1B02
Length	05
Track2 data	2B8849EAAF
Track3 header	1B03
Length	05
Track3 data	2B8849EAAF
Ending Field	3F1C

Transfer the track1 data to HEX under 08 bits:

	B0	B1	B2	B3	B4	B5	P
%	1	0	1	0	0	0	1
A	1	0	0	0	0	1	1
B	0	1	0	0	0	1	1
C	1	1	0	0	0	1	0
1	1	0	0	0	1	0	1
2	0	1	0	0	1	0	1
3	1	1	0	0	1	0	0
?	1	1	1	1	1	0	0
LRC	0	1	0	1	0	1	0

Calculate Odd Parity (P column)
If there are an Even NUMBER OF 1's in the row of data for each character, put a 1 in the P column. Otherwise, put a 0 in the column.

LRC: If there are an Even Number of 1's in the column of data for each character, put a 0 in the LRC row. Otherwise, put a o in the row. The last LRC will be considered as the parity rule of this row.

B0	B1	B2	B3	B4	B5	B6	B7	
1	0	1	0	0	0	1	1	
0	0	0	0	1	1	0	1	
0	0	0	1	1	1	1	0	
0	0	1	0	1	0	0	0	
1	0	1	0	1	0	0	1	
0	1	1	1	0	0	1	0	
0	1	1	1	1	1	0	0	
0	1	0	1	0	1	0	0	

B7	B6	B5	B4	B3	B2	B1	B0	HEX
1	1	0	0	0	1	0	1	C5
1	0	1	1	0	0	0	0	B0
0	1	1	1	1	0	0	0	78
0	0	0	1	0	1	0	0	14
1	0	0	1	0	1	0	1	95
0	1	0	0	1	1	1	0	4E
0	0	1	1	1	1	1	0	3E
0	0	1	0	1	0	1	0	2A

Transfer track 2 (track 3)data to HEX under 08 bits:

	B0	B1	B2	B3	P
;	1	1	0	1	0
1	1	0	0	0	0
2	0	1	0	0	0
3	1	1	0	0	1
4	0	0	1	0	0
5	1	0	1	0	1
?	1	1	1	1	1
LRC	1	0	1	0	1

B0	B1	B2	B3	B4	B5	B6	B7
1	1	0	1	0	1	0	0
0	0	0	1	0	0	0	1
1	0	0	1	0	0	1	0
0	1	0	1	0	1	1	1
1	1	1	1	0	1	0	1

B7	B6	B5	B4	B3	B2	B1	B0	HEX
0	0	1	0	1	0	1	1	2B
1	0	0	0	1	0	0	0	88
0	1	0	0	1	0	0	1	49
1	1	1	0	1	0	1	0	EA
1	0	1	0	1	1	1	1	AF

07,05,05 BITS

Set TK1 TK2 & TK3 AS 07,05,05

1. 1b,6F,07,05,05
   First of all, set BPI command:

2. Present the information to the card encoder, as follows:

UIC	
Start Field	1B6E 1B73
Track1 header	1B01
Length	09
Track1 data	456162235152131F2A
Track2 header	1B02
Length	08
Track2 data	0B01021304151F15
Track3 header	1B03
Length	08
Track3 data	0B01021304151F15
Ending Field	3F1C

Transfer the track1 data to HEX under 07 bits:

	B0	B1	B2	B3	B4	B5	P
%	1	0	1	0	0	0	1
A	1	0	0	0	0	1	1
B	0	1	0	0	0	1	1
C	1	1	0	0	0	1	0
1	1	0	0	0	1	0	1
2	0	1	0	0	1	0	1
3	1	1	0	0	1	0	0
?	1	1	1	1	1	0	0
LRC	0	1	0	1	0	1	0

Calculate Odd Parity (P column)
If there are number of 1'sin the row of data for each character, put a1 in the P column. Otherwise, put a 0 in the column.

	Add	p	B5	B4	B3	B2	B1	B0	HEX
%	0	1	0	0	0	1	0	1	45
A	0	1	1	0	0	0	0	1	61
B	0	1	1	0	0	0	1	0	62
C	0	0	1	0	0	0	1	1	23
1	0	1	0	1	0	0	0	1	51
2	0	1	0	1	0	0	1	0	52
3	0	0	0	1	0	0	1	1	13
?	0	0	0	1	1	1	1	1	1F
LRC	0	0	1	0	1	0	1	0	2A

HEX

	B3	B2	B1	B0
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
A	1	0	1	0
B	1	0	1	1
C	1	1	0	0
D	1	1	0	1
E	1	1	1	0
F	1	1	1	1

Transfer tack2 (Track3) data to HEX under 05 bits:

	B0	B1	B2	B3	P
;	1	1	0	1	0
1	1	0	0	0	0
2	0	1	0	0	0
3	1	1	0	0	1
4	0	0	1	0	0
5	1	0	1	0	1
?	1	1	1	1	1
LRC	1	0	1	0	1

	Add 0	Add 0	Add 0	P	B3	B2	B1	B0	HEX
;	0	0	0	0	1	0	1	1	0B
1	0	0	0	0	0	0	0	1	01
2	0	0	0	0	0	0	1	0	02
3	0	0	0	1	0	0	1	1	13
4	0	0	0	0	0	1	0	0	04
5	0	0	0	1	0	1	0	1	15
?	0	0	0	1	1	1	1	1	1F
LRC	0	0	0	1	0	1	0	1	15

06,05,06 BITS

3. Set TK1 ,TK2 & TK3 as 06,05,06
   First of all, set BPI command:
   1b,6F,06,05,06

4. Present the information to the card encoder, as follows:

UIC	
Start Field	1B6E 1B73
Track1 header	1B01
Length	09
Track1 data	052122231112131F2A
Track2 header	1B02
Length	08
Track2 data	0B01021304151F15
Track3 header	1B03
Length	08
Track3 data	0101020304051F1F
Ending Field	3F1C

Transfer track1 data to HEX under 06 bits:

	B0	B1	B2	B3	B4	B5
%	1	0	1	0	0	0
A	1	0	0	0	0	1
B	0	1	0	0	0	1
C	1	1	0	0	0	1
1	1	0	0	0	1	0
2	0	1	0	0	1	0
3	1	1	0	0	1	0
?	1	1	1	1	1	0
LRC	0	1	0	1	0	1

	Add 0	Add 0	B5	B4	B3	B2	B1	B0	HEX
%	0	0	0	0	0	1	0	1	05
A	0	0	1	0	0	0	0	1	21
B	0	0	1	0	0	0	1	0	22
C	0	0	1	0	0	0	1	1	23
1	0	0	0	1	0	0	0	1	11
2	0	0	0	1	0	0	1	0	12
3	0	0	0	1	0	0	1	1	13
?	0	0	0	1	1	1	1	1	1F
LRC	0	0	1	0	1	0	1	0	2A

Transfer tack2 data to HEX under 05 bits:

	B0	B1	B2	B3	P
;	1	1	0	1	0
1	1	0	0	0	0
2	0	1	0	0	0
3	1	1	0	0	1
4	0	0	1	0	0
5	1	0	1	0	1
?	1	1	1	1	1
LRC	1	0	1	0	1

	Add 0	Add 0	Add 0	P	B3	B2	B1	B0	HEX
;	0	0	0	0	1	0	1	1	0B
1	0	0	0	0	0	0	0	1	01
2	0	0	0	0	0	0	1	0	02
3	0	0	0	1	0	0	1	1	13
4	0	0	0	0	0	1	0	0	04
5	0	0	0	1	0	1	0	1	15
?	0	0	0	1	1	1	1	1	1F
LRC	0	0	0	1	0	1	0	1	15

Transfer tack3 data to HEX under 06 bits:

	B0	B1	B2	B3	B4	B5
!	1	0	0	0	0	0
1	1	0	0	0	0	0
2	0	1	0	0	0	0
3	1	1	0	0	0	0
4	0	0	1	0	0	0
5	1	0	1	0	0	0
?	1	1	1	1	1	0
LRC	1	1	1	1	1	0

	Add 0	Add 0	B5	B4	B3	B2	B1	B0	HEX
!	0	0	0	0	0	0	0	1	01
1	0	0	0	0	0	0	0	1	01
2	0	0	0	0	0	0	1	0	02
3	0	0	0	0	0	0	1	1	03
4	0	0	0	0	0	1	0	0	04
5	0	0	0	0	0	1	0	1	05
?	0	0	0	1	1	1	1	1	1F
LRC	0	0	0	1	1	1	1	1	1F

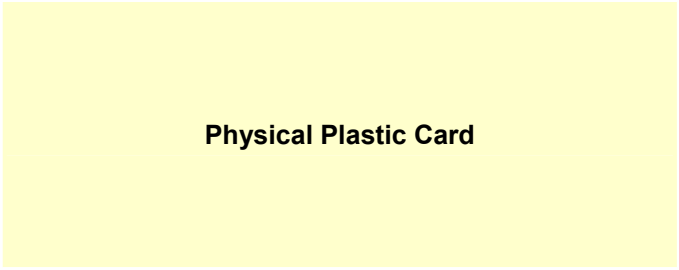
ISO Magnetic Stripe Card Standards

The majority of magnetic cards used in the UK, Europe and USA confirm to the following ISO standards for magnetic cards.

ISO Number	Description of Standard
7810	Physical Characterisitics of Credit Card Size Document
7811-1	Embossing
7811-2	Magnetic Stripe - Low Coercivity
7811-3	Location of Embossed Characters
7811-4	Location of Tracks 1 and 2
7811-5	Location of Track 3
7811-6	Magnetic Stripe - High Coercivity
7813	Financial Transaction Cards

Full copies of these standards can be pruchased from www.iso.org and www.ansi.org. The information below is abstracted from these standards.

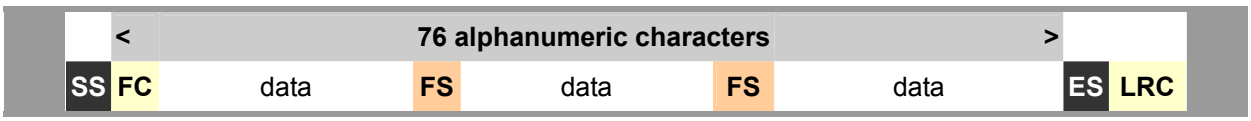
Physical Dimensions of Cards:



Characteristics of Tracks:

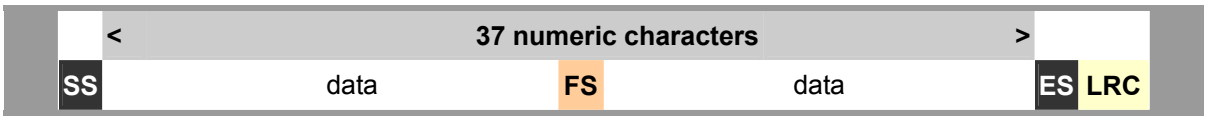
Position	Track Number	Recording Density (bits per inch)	Character Configuration (including parity bit)	Information Content (including control characters)
0.223" (5.664mm) from card edge				
0.110" (2.794mm)	Track 1	210BPI	7 bits per character	79 Alphanumeric characters
0.110" (2.794mm)	Track 2	75BPI	5 bits per character	40 Numeric characters
0.110" (2.794mm)	Track 3	210BPI	5 bits per character	107 Numeric characters

Track One:



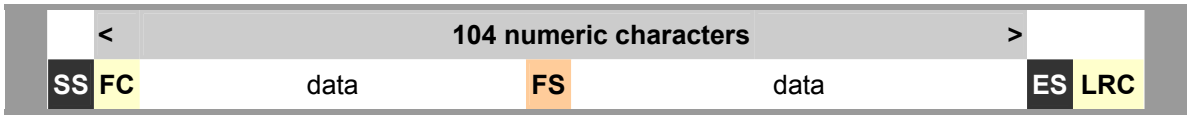
	Description	Value
SS	Start Sentinel	05h
FC	Format Code	
FS	Field Seperator	3Eh
ES	End Sentinel	1Fh
LRC	Longitudinal Redunancy Check Character	

Track Two:



	Description	Value
SS	Start Sentinel	0Bh
FS	Field Seperator	0Dh
ES	End Sentinel	0Fh
LRC	Longitudinal Redunancy Check Character	

Track Three: (ISO 4909)



	Description	Value
SS	Start Sentinel	**0Bh**
FC	Format Code	2 digits
FS	Field Seperator	**0Dh**
ES	End Sentinel	**0Fh**
LRC	Longitudinal Redunancy Check Character	

Character Sets

Data stored on magnetic stripes does not use the standard ASCII character set as used on PCs. Below are the two ANSI/ISO character sets used on magnetic stripes.

ANSI/ISO ALPHA Data Format
(Track 1)

The ANSI/ISO ALPHA format is 7 bit, 6 data bits + 1 parity bit (odd). The data is read least significant bit first.
The character set contains 64 characters, 43 alphanumeric, 3 framing/field characters and 18 control/special charact

Data bits							Character	Value	Function
b1	b2	b3	b4	b5	b6	b7		(Hex)	
0	0	0	0	0	0	1	space	00	Special
1	0	0	0	0	0	0	!	01	Special
0	1	0	0	0	0	0	"	02	Special
1	1	0	0	0	0	1	#	03	Special
0	0	1	0	0	0	0	$	04	Special
1	0	1	0	0	0	1	%	05	**Start Sentinel**
0	1	1	0	0	0	1	&	06	Special
1	1	1	0	0	0	0	'	07	Special
0	0	0	1	0	0	0	(	08	Special
1	0	0	1	0	0	1	)	09	Special
0	1	0	1	0	0	1	*	0A	Special
1	1	0	1	0	0	0	+	0B	Special
0	0	1	1	0	0	1	,	0C	Special

1	0	1	1	0	0	0	-	0D	Special
0	1	1	1	0	0	0	.	0E	Special
1	0	0	1	0	0	1	/	0F	Special
0	0	0	0	1	0	0	0	10	Data
1	0	0	0	1	0	1	1	11	Data
0	1	0	0	1	0	1	2	12	Data
1	1	0	0	1	0	0	3	13	Data
0	0	1	0	1	0	1	4	14	Data
1	0	1	0	1	0	0	5	15	Data
0	1	1	0	1	0	0	6	16	Data
1	1	1	0	1	0	1	7	17	Data
0	0	0	1	1	0	1	8	18	Data
1	0	0	1	1	0	0	9	19	Data
0	1	0	1	1	0	0	:	1A	Special
1	1	0	1	1	0	1	;	1B	Special
0	0	1	1	1	0	0	<	1C	Special
1	0	1	1	1	0	1	=	1D	Special
0	1	1	1	1	0	1	>	1E	Special
1	1	1	1	1	0	0	?	1F	**End sentinel**
0	0	0	0	0	1	0	@	20	Special
1	0	0	0	0	1	1	A	21	Data
0	1	0	0	0	1	1	B	22	Data
1	1	0	0	0	1	0	C	23	Data
0	0	1	0	0	1	1	D	24	Data
1	0	1	0	0	1	0	E	25	Data
0	1	1	0	0	1	0	F	26	Data
1	1	1	0	0	1	1	G	27	Data
0	0	0	1	0	1	1	H	28	Data
1	0	0	1	0	1	0	I	29	Data
0	1	0	1	0	1	0	J	2A	Data
1	1	0	1	0	1	1	K	2B	Data
0	0	1	1	0	1	0	L	2C	Data
1	0	1	1	0	1	1	M	2D	Data
0	1	1	1	0	1	1	N	2E	Data
1	1	1	1	0	1	0	O	2F	Data

0	0	0	0	1	1	1	P	30	Data
1	0	0	0	1	1	0	Q	31	Data
0	1	0	0	1	1	0	R	32	Data
1	1	0	0	1	1	1	S	33	Data
0	0	1	0	1	1	0	T	34	Data
1	0	1	0	1	1	1	U	35	Data
0	1	1	0	1	1	1	V	36	Data
1	1	1	0	1	1	0	W	37	Data
0	0	0	1	1	1	0	X	38	Data
1	0	0	1	1	1	1	Y	39	Data
0	1	0	1	1	1	1	Z	3A	Data
1	1	0	1	1	1	0	[	3B	Special
0	0	1	1	1	1	1	\	3C	Special
1	0	1	1	1	1	0	]	3D	Special
0	1	1	1	1	1	0	^	3E	**Field Separator**
1	1	1	1	1	1	1	_	3F	Special

ANSI/ISO BCD Data Format

(Tracks 2 and 3)

The ANSI/ISO BCD format is 5 bit, 4 data bits + 1 parity bit (odd). The data is read least significant bit first.

The character set contains 16 characters, 10 alphanumeric, 3 framing/field characters and 3 control/special characters.

Data bits					Character	Value	Function
b1	b2	b3	b4	b5		(Hex)	

0	0	0	0	1	0	00	Data
1	0	0	0	0	1	01	Data
0	1	0	0	0	2	02	Data
1	1	0	0	1	3	03	Data
0	0	1	0	0	4	04	Data
1	0	1	0	1	5	05	Data
0	1	1	0	1	6	06	Data
1	1	1	0	0	7	07	Data
0	0	0	1	0	8	08	Data
1	0	0	1	1	9	09	Data
0	1	0	1	1	:	0A	Control
1	1	0	1	0	;	0B	**Start Sentinel**
0	0	1	1	1	<	0C	Control
1	0	1	1	0	=	0D	**Field Separator**
0	1	1	1	0	>	0E	Control
1	1	1	1	1	?	0F	**End Sentinel**

----------END----------