I/O INTERFACE FOR TTL MAGNETIC STRIPE READERS

TECHNICAL REFERENCE MANUAL

Manual Part Number 99875148 Rev-6

DECEMBER 2003

MAGTEK®

REGISTERED TO ISO 9001:2000

20725 South Annalee Avenue
Carson, CA 90746
Phone: (310) 631-8602
FAX: (310) 631-3956
Technical Support: (651) 415-6800
www.magtek.com

REVISIONS

Rev Number	Date	Notes
1	22 Oct 99	Initial Release
2	12 Jun 00	Table of Contents corrected; Appendix A, Applications Note, Character Conversion 99875065-2 replaced -1 as update.
3	24 Jun 02	Sec 1: Updated MagTek Documents. Sec 4: Corrected Error Checking, Track 2; added Insertion Reader. Appendix A: Replaced 99875065-2 with –3. Appendix B: Replaced 99821066-8 with –10. Appendix C: Editorial corrections.
4	06 Aug 02	Front Matter: Title change; Sec 1, MagTek Documents, deleted Rev numbers and updated P/N 210065015/16 to /16 only, added 8 manuals to Reference Documents; Sec 2: corrected Error Checking, Track 2, added Insertion Readers; Sec 4: added Insertion Readers subsection; Appendix A : corrected character]ND and bumped rev; Appendix B: removed DIP IC 15, bumped rev; Appendix C: changed ANSI address, and dimensions on Trks 1, 2, 3.
5	07 Aug 03	Editorial and format corrections throughout. Front Matter: added ISO line to logo, changed Tech Support phone number. Sec 2: added note to Figure 2-2.
6	17 Dec 03	Appendix C, Standards, Card Data Format – Track 1: extended Data Character line to include Format Code (FC).

TABLE OF CONTENTS

FIGURES

SECTION 1. DECODING READER OUTPUT

The usual way to gather card data from a TTL (Transistor Transistor Logic) swipe magnetic card reader is as follows:

1. Connect the strobe line to an interrupt port on the CPU that is programmed to interrupt on the falling edge of the strobe.

2. Connect the data line to an ordinary CPU port pin.

3. Connect the card present line to another ordinary CPU port pin.

All three of the above signals are quiescently high (V_{cc}). When a card is swiped, the first signal to go low (active) is Card Present. It will go low after 14 to 15 head flux reversals have been read. These initial strobes are not output on the strobe line. After Card Present goes low, the strobes are output on the strobe line. Normally, storing data is not begun until the data line goes low (indicating a one) and the strobe line falls. When this occurs, the CPU starts to store the data in RAM (Random Access Memory) until the strobes stop because the card has cleared the head. Approximately 150ms later, Card Present returns to $V_{cc.}$ When this occurs, the CPU analyzes the data in RAM and checks for parity errors and LRC (Longitudinal Redundancy Check or check sum) error. If no errors are detected, the CPU outputs the data in some useful form, such as ASCII (American Standard Code for Information Interchange) or Keyboard Scan Code.

There are two methods of storing data, one way is to put each character into its own byte in RAM. The advantage of this method is that it is easy to debug the software. The disadvantages are that some bits are not used (wasted) and the card can only be read in the forward direction. The second method is to store all bits from the card into RAM (no unused bits). The only disadvantage to this method is that it is difficult to debug your software code. The reason being, in the case of Track 2 (5 bits per character), the first byte of RAM has the first character (5 bits) and 3 bits of the second character. The second byte of ram has 2 bits of the second character, 5 bits of the third character, and 1 bit from the fourth character, etc. When using In-Circuit Emulation (ICE) to decode the data in RAM, it is difficult to figure out what you have. Also, with both methods the data is read in backwards (Least Significant Bit First).

If you run out of interrupts, one way to create another interrupt is to set a counter to all one's. When the strobe falls, the counter increments causing a counter interrupt which can be used to collect the data bit and reset the counter to all one's again.

REFERENCE DOCUMENTS

43-millimeter Compatible, Swipe Reader, Technical Reference Manual, P/N 99875148
90-millimeter Compatible, Swipe Reader, Technical Reference Manual, P/N 99831083
100-millimeter Compatible, Swipe Reader, Technical Reference Manual, P/N 99831012
101-millimeter Compatible, Swipe Reader, Technical Reference Manual, P/N 99821101
Slim Profile Card Guide, Swipe Reader, Technical Reference Manual, P/N 99875147
MT-215, TTL, Half Card Insertion Reader, Technical Reference Manual, P/N 99875009
Model MT-215, TTL, Insertion Reader, Technical Reference Manual, P/N 99875042
IntelliStripe 50 Manual Insertion Reader Tech Reference Manual, P/N 99875079
Model MT-215 TTL, Single or Dual Head, Insertion Reader Technical Reference Manual, P/N 99875113
IntelliStripe 60, TTL Insertion Reader, Technical Reference Manual, P/N 99875135
Petrol (P-Series), TTL, Insertion Readers, Technical Reference Manual, P/N 99875139
Model MT-215, TTL, Single Head, 3-Track, Insertion Reader Technical Reference Manual, P/N 99875157

MAGTEK DOCUMENTS INCLUDED IN THIS MANUAL

The following MagTek documents are included as appendices in this manual:

Appendix A. *Character Conversion*, P/N 99875065
Appendix B. *Read IC 21006516 Spec Sheet*, P/N 99821066
Appendix C. *Magnetic Stripe Card Standards*, P/N 99800004
Appendix D. *Card Reader Interface*, P/N 99875063

TRACK 1 DATA FORMAT

The following is typical for Track 1 data.

The Track 1 data format includes the following items:

1. Approximately 62 leading clocking zeros
2. The Start Sentinel "%"
3. Up to 76 alphanumeric characters selected by the user
4. The end sentinel "?"
5. The LRC (longitudinal redundancy check character)
6. Trailing clocking zeros (minimum of 62) to the end of the card

Each data character is in a 7-bit (6 plus parity) alphanumeric format. The data is in the format "parity, 32, 16, 8, 4, 2, 1". Parity is odd (an odd number of "one's" in each character).

The character is written "backwards" on the card starting with the least significant digit and ending with the parity bit. The card data format is "1, 2, 4, 8, 16, 32, parity".

See Appendix C for ATM and credit card formats.

Example: The data message "AT3" is encoded on the card as follows:

1.	"000..00"	leading clocking zeros	(approximately 62)
2.	"1010001"	the Start Sentinel "%"	(1000101)
3.	"1000011"	the character "A"*	(1100001)
4.	"0010110"	the character "T"*	(0110100)
5.	"1100100"	the character "3"	(0010011)
6.	"1111100"	the End Sentinel "?"	(0011111)
7.	"0011100"	the character "<"**	(0011100) the LRC (check digit)
8.	"000..00"	trailing clocking zeros	(minimum of 62) to end of the card

*Use capital letters only.
**The character "<" is not defined by credit card standards but is a legitimate LRC character.

The binary values of all card data can be found in the MagTek publication, Application Note - Character Conversion, Part Number 99875065, see Appendix A.

See Appendix C for ATM and Credit Card Formats.

Calculating odd parity and LRC, for the message AT3:

		P	32 16 8 4 2 1	Hex Value, Less Parity	Decimal Value Less Parity
Start sentinel	%	1	0 0 0 1 0 1	05 hex	05 (4+1)
	A	1	1 0 0 0 0 1	21 hex	33 (32+1)
	T	0	1 1 0 1 0 0	34 hex	52 (32+16+4)
	3	0	0 1 0 0 1 1	13 hex	19 (16+2+1)
End sentinel:	?	0	0 1 1 1 1 1	1F hex	31 (16+8+4+2+1)
LRC	<	0	0 1 1 1 0 0	1C hex	28 (16+8+4)

Calculating odd parity: scan each horizontal row and enter a zero or one In the P column so that each row has an odd number of one's.

To find the LRC: disregard the parity bits, scan up each vertical column, and make sure that each column has an even number of one's in it. Enter a zero or one in the LRC position, so that it does. If you are using a computer to calculate the LRC, use the "Exclusive OR" function. Next, calculate odd parity for the LRC character, in the horizontal direction.

TRACKS 2 AND 3 DATA FORMAT

The following is typical for Track 2 and Track 3 data.

The Track 2 and 3 data formats include the following items:

1. Leading clocking zeros: Approximately 22 (Track 2) or 62 (Track 3)
2. The start sentinel (hexadecimal "B")
3. Up to 37 (track 2) or 104 (track 3) hexadecimal numeric characters selected by the user
4. The end sentinel (hexadecimal "F")
5. The LRC (longitudinal redundancy check character)
6. Trailing clocking zeros: A minimum of 22 (track 2) or 62 (track 3)
   to the end of the card

Each data character is in a 5-bit (4 plus parity) numeric format. The data is in the format "parity, 8, 4, 2, 1". Parity is odd (an odd number of one's in each character).

The character is written "backwards" on the card starting with the least significant bit and ending with the parity bit. The card data format Is "1, 2, 4, 8, parity".

Example: The data message "123" is encoded on the card as follows:

1. "00.00"	leading clocking zeros	Approximately 22 (tk2) or 62 (tk3)
2. "11010"	the character "hex B"	(01011) the Start Sentinel
3. "10000"	the character "1"	(00001)
4. "01000"	the character "2"	(00010)
5. "11001"	the character "3"	(10011)
6. "11111"	the character "hex F"	(11111) the End Sentinel
7. "00100"	the character "4"	(00100) the LRC (check digit)
8. "00.00"	trailing clocking zeros	A minimum of 22 (Tk2) or 62 (tk3) to the end of the card

Calculating odd parity and LRC for the message 123:

		P	8 4 2 1	Hex Value	Decimal Value
				Less Parity	Less Parity
Start Sentinel:	B	0	1 0 1 1	B	11 (8+2+1)
	1	0	0 0 0 1	1	1
	2	0	0 0 1 0	2	2
	3	1	0 0 1 1	3	3 (2+1)
End Sentinel:	F	1	1 1 1 1	F	15 (8+4+2+1)
LRC	4	0	0 1 0 0	4	4

Calculating odd parity: scan each horizontal row and enter a zero or one In the P column so that each row has an odd number of one's.

To find the LCR: disregard the parity bits, scan up each vertical column, and make sure that each column has a even number of one's in it. Enter a zero or one in the LCR position, so that it does. If you are using a computer to calculate the LCR, use the "exclusive or" function. Next, calculate odd parity for the LCR character, in the horizontal direction.

The Hex characters are usually displayed in their ASCII equivalent: Hex B as a ";" semicolon, Hex D as an "=" equal sign, and Hex F as a "?" question mark.

SECTION 2. ENCODING

There are several schemes used to record or encode ones and zeros in the computer industry. (We use the term record and encode synonymously.) In the card industry the International Organization for Standardization (ISO) has defined F2F as the encoding scheme for cards. F2F stands for frequency - double frequency, or for the purist - two frequency coherent phase recording. F2F encoding provides for self-clocking data. That is, the serial data stream consisting of one and zero bits provides the timing information for the reader to determine which bit is which.

The key feature of self-clocking data is that the data bits can be extracted from the serial data stream without the need to control the speed of the magnetic media past the encoding head. Thus, card swipe readers, where a human hand is passing the card through the read slot can work regardless of how fast or slow the card is passing through the slot. Serial data merely means that the one and zero bits, that form the desired characters, are stored on the same track, one bit after the next bit.

F2F is an encoding technique which places flux transitions on the magnetic stripe separated by a defined distance for zero bits and one half that distance for one bits.

The defined distance for Track 1 and Track 3 is 0.0047619 inches for zero bits. This value is the reciprocal of the bit density of Track 1 - 210 bits per inch. For one bits - 0.0023809 (1/2 the zero bit distance.)

The defined distance for Track 2 is 0.013333 inches for zero bits. This value is the reciprocal of the bit density of Track 2 - 75 bits per inch. for one bits - 0.0066665 inches (1/2 the zero bit density.)

Figure 2-1 illustrates this concept.

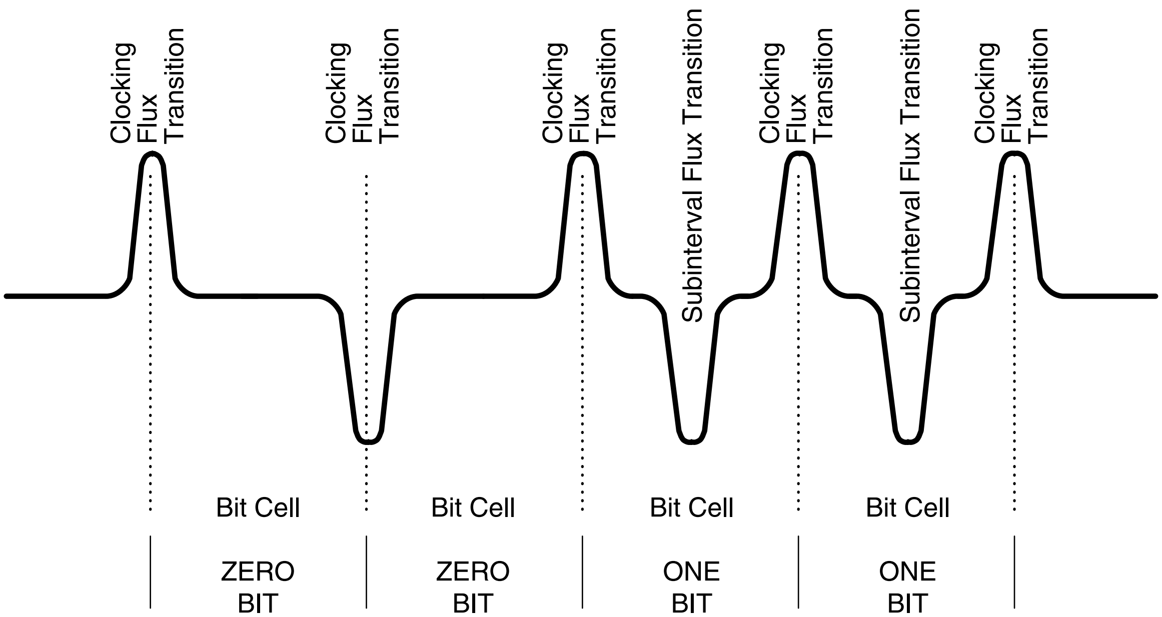


Figure 2-1. Bit Cell - Flux Transition

The encoding device has the responsibility to separate the flux transitions at the proper distance. Variations in the proper distance between flux transitions is what is commonly referred to as jitter.

Before the desired information can be stored (encoded) on the card, many things must happen. The information itself (the alphanumeric characters for Track 1 and the numeric characters for Tracks 2 and 3) needs to be coded. That is, the information must be converted to one and zero bits. The one and zero bit code for each character can be found in the Coded Character Set tables in ISO/IEC 7811-2, see Appendix A. These tables define the pattern of the one and zero bits for each character. There is a special bit that is added to each character code for the purpose of error detection. This special bit is called a parity bit. The parity bit is either a zero bit or a one bit depending on whether the number of one bits in the character is an even number or and odd number. The ISO standard has defined that for all tracks odd parity is maintained on each character.

This means that if the character has an odd number of one bits in the character code for that character, the parity bit is then a zero bit. On the other hand, if the character has an even number of one bits in the character code for that character, the parity bit is a one bit causing the total number of one bits to be odd. Thus the term odd parity.

Before the first character (or more specifically, the first series of bits that define the desired first character) can be encoded, a Start Sentinel character needs to be encoded. The Start Sentinel is a defined series of bits, which signal the start of the data bits. Track 1 Start Sentinel consists of seven bits (as all characters on Track 1). Track 2 and 3 Start Sentinel consists of five bits (as all characters on Track 2). As an example, the Track 2 Start Sentinel code is: 11010.
After the last data character is encoded, an End Sentinel must be encoded. The End Sentinel is a defined series of bits, which signal the end of the data bits. Immediately after the End Sentinel, a special error checking character is also encoded. This error checking character is called the LRC - Longitudinal Redundancy Check character. For further details see Section 3.

Before the Start Sentinel character and after the LRC character, "clocking bits" are encoded. These clocking bits provide card readers with timing to start their synchronization process before encountering the start sentinel (or end sentinel for reverse reading). Figure 2-2 illustrates the relative location of all the characters previously discussed.

This means that if the character has an odd number of one bits in the character code for that character, the parity bit is then a zero bit. On the other hand, if the character has an even number of one bits in the character code for that character, the parity bit is a one bit causing the total number of one bits to be odd. Thus the term odd parity.

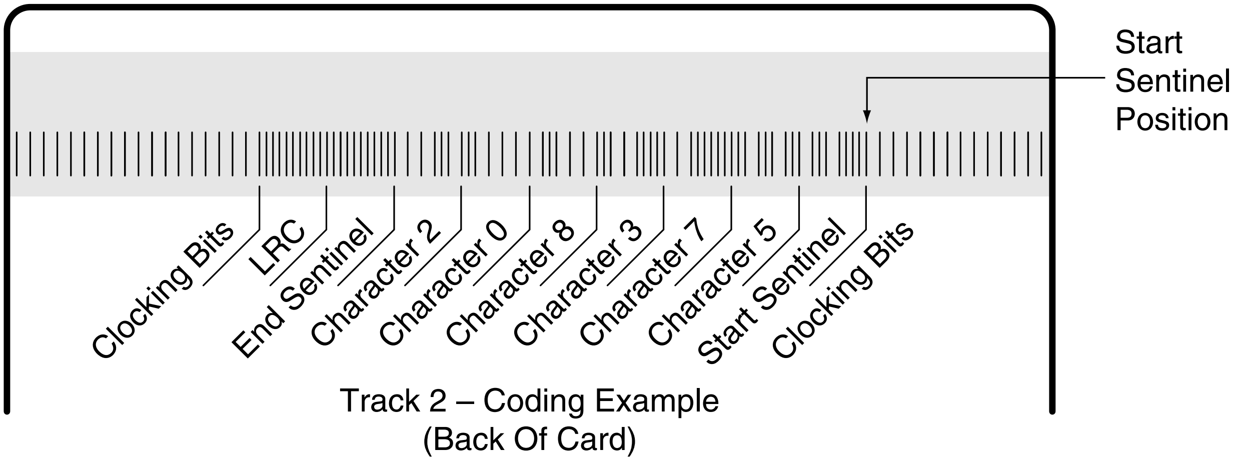


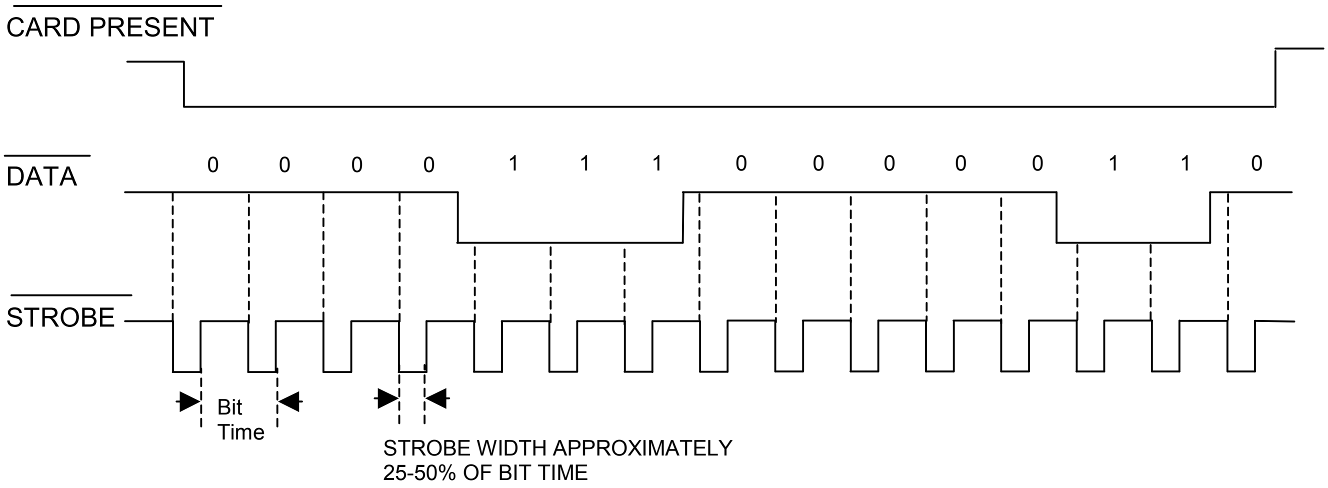
Figure 2-2. Card Coding

Note: In this example, Start Sentinel is Hex B, End Sentinel is Hex F, and LRC is Hex F.

Note that as you look at the diagram, while not to scale, the encoded information does travel from right to left along the length of the magnetic stripe. See Appendix A.

TIMING

The timing for Card Present, Strobe, and Data is as shown in Figure 2-3.



STROBE WIDTH APPROXIMATELY 25-50% OF BIT TIME

Notes:

1. Time out of the $\overline{\text{CARD PRESENT}}$ signal occurs approximately 150 ms after the last strobe transition.
2. $\overline{\text{DATA}}$ is valid 1.0µ sec before the negative edge of $\overline{\text{STROBE}}$.

Figure 2-3. Timing

DATA

The Data signal is valid while the strobe is low. If the Data signal is high, the bit is a zero. If the Data signal is low, the bit is a one.

STROBE

The Strobe signal indicates when Data is valid. It is recommended that Data be loaded by the user with the leading edge (negative) of the Strobe.

CARD PRESENT

Card Present will go low after 14/15 flux reversals from the head. Card Present will return high 150 milliseconds after the last flux reversal.

When no card is being moved through the unit, the Data, Strobe, and Card Present signals are high. The signal timing diagram shown above represents the data along with other signals that are generated during the reading process.

SECTION 3. LRC CALCULATION

The formula for calculating LRC below uses Track 2 as an example. Perform the following:

1. Write the value for each character (see examples below and Appendix A).

2. Do not use the odd parity bit (P) in the formula for calculating LRC calculation.

3. Count the number of one bits in each of the 4 vertical columns.

4. Enter a zero or one in the LRC position so that each vertical column has a even number of ones (not the parity column {P}).

5. After all 4 vertical columns have their LRC bit affixed, affix odd parity to the LRC character itself.

Examples:

	P	8	4	2	1
B	0	1	0	1	1
0	1	0	0	0	0
F	1	1	1	1	1
4	0	0	1	0	0

Where:
"B" is the start sentinel
"F" is the end sentinel
The LRC is under the line

	P	8	4	2	1
B	0	1	0	1	1
0	1	0	0	0	0
1	0	0	0	0	1
F	1	1	1	1	1
5	1	0	1	0	1

	P	8	4	2	1
B	0	1	0	1	1
0	1	0	0	0	0
1	0	0	0	0	1
2	0	0	0	1	0
3	1	0	0	1	1
F	1	1	1	1	1
4	0	0	1	0	0

Calculation of LRC for an encoder, using a Micro Processor (Intel 8751), is as follows:

START:

```
MOV R2,#0 ;ZERO REGISTER 2 (LRC ACCUMULATOR)
MOV R0,#START_SEN ;POINT AT FIRST CARD CHARACTER (START SENTINEL)
```

LOOP1:

```
MOVX A,@R0 ;PUT A CARD CHARACTER INTO THE ACCUMULATOR
ANL A,#0FH ;MASK OFF THE PARITY BIT
XCH A,R2 ;EXCHANGE THE ACCUMULATOR & REG 2
XRL A,R2 ;"EXCLUSIVE OR" ACC WITH R2
XCH A,R2 ;RUNNING TOTAL OF LRC IS NOW IN REG 2, CARD CHARACTER IN ACC.
INC R0 ;POINT AT THE NEXT CHARACTER
CJNE A,0FH,LOOP1 ;COMPARE, JUMP TO LOOP1 IF NOT EQUAL TO A END SENTINEL.
;WHEN THE PROGRAM FALLS THROUGH THE ABOVE INSTRUCTION THE FINAL LRC IS
;IN REGISTER 2, WE CALL THE ODD PARITY ROUTINE TO AFFIX ODD PARITY TO THE
;LRC AND WE ARE READY TO WRITE THE LRC ON A CARD.
```

SECTION 4. ERROR CHECKING

This section describes error checking by a typical single track reader and with and Intel 8751 CPU on a typical Track 2 reader.

ERROR CHECKS BY A TYPICAL READER

The following error checking is performed by a typical card reader:

1. While the card is being swiped no error checking is done. All of the micro processor unit (MPU) time is being devoted to storing the data being read off the card into memory (RAM). Normally the strobe output from the card reader is connected to an interrupt input on the MPU. The MPU is set so that the interrupt is triggered on the falling edge of the strobe. The data output line from the card reader goes to a normal MPU port input pin. On each Falling edge of the strobe, one bit of data is collected from the data input line and the interrupt routine is exited. The 'Card Present' output from the card reader goes to a normal MPU port Input pin. All outputs from the card reader are quiescently high (V_{cc}).

2. When 'card present' returns to high again the MPU knows that the card has cleared the read head and error checking can begin. The MPU now checks each character that it collected in step 1 for the following:

   A. Checks to ensure the first character is a 'Start Sentinel'. This will be a hex B (01011) for Track 2 or Track 3. For Track 1 it will be a percent sign (1000101).

   B. Checks each character for odd parity (an odd number of one bits).

   C. Updates the LRC total as each character is checked.

   D. Checks to see if the current character is an 'End Sentinel' (11111, 0011111). If no 'End Sentinel' is found, this is an error. When the 'end sentinel' is found, the MPU inputs the next character (LRC), checks its parity, and updates the LRC total. It then checks that the LRC total is zero. If it is not zero it is an error.

3. Some readers keep count of the number of characters checked, and if this number exceeds the maximum allowed, will give an error. The maximum allowed is: track 1 = 79; track 2 = 40; track 3 = 107.

4. If for some reason you do not have enough interrupts available to use the preceding method, MagTek can provide an application note on how to do it without using interrupts. This is done by using a "USART" for each track along with an MPU. See Appendix D.

ERROR CHECKING, TRACK 2, WITH AN INTEL 8751

Example of error checking in a Track 2 reader, using an Intel 8751 CPU:

```
; INITAILIZE REGISTERS AND FLAG BITS.

CHECK: MOV R2,#0 ;ZERO REGISTER 2 (LRC ACCUMULATOR)
    CLR PARITY_FLAG ;CLEAR THE PARITY ERROR FLAG
    MOV R0,#START_SEN ;POINT AT FIRST CARD CHARACTER IN RAM (START SENTINEL)

; START PARITY AND LRC CHECK (LONGITUDINAL REDUNDANCY CHECK CHARACTER).

CK0:   CALL CK1 ;GO TEST A CHARACTER
    CJNZ A,0FH,CK0 ;COMPARE ACCUMULATOR TO END SENTINEL, GO TO CK0 IF NOT ES
             ;IF IT'S A END SENTINEL THE PROGRAM DROPS THROUGH
    CALL CK1 ;GO UPDATE THE LRC CHARACTER
    MOV A,R2 ;MOVE REGISTER 2 TO THE ACCUMULATOR, GET THE LRC SUM
    JNZ ERROR ;IF LRC SUM IS NOT ZERO, THIS EQUALS AN ERROR
    JB PARITY_FLAG,ERROR ;SKIP TO ERROR IF PARITY FLAG IS SET
    NOP ;IF THIS POINT IS REACHED THE CARD WAS READ CORRECTLY
      ;END OF TEST


; CK1--LRC SUBROUTINE
; THIS ROUTINE KEEPS THE RUNNING SUM OF THE LRC IN REGISTER 2

CK1:   MOVX A,@R0 ;PUT A CARD CHARACTER FROM RAM INTO THE ACCUMULATOR
    CALL PARITY_CK ;CHECK FOR ODD PARITY
    MOVX A,@R0 ;GET THE SAME CHARACTER AGAIN
    ANL A,#0FH ;ZERO THE FIRST 4 BITS, INCLUDING THE PARITY BIT
    XCH A,R2 ;EXCHANGE ACCUMULATOR AND REGISTER 2
    XRL A,R2 ;"EXCLUSIVE OR" THE ACCUMULATOR WITH REGISTER 2
    XCH A,R2 ;EXCHANGE ACCUMULATOR AND REGISTER 2
    INC R0 ;POINT AT THE NEXT CHARACTER
    RET ;RETURN, AT THIS TIME THE RUNNING LRC IS IN REGISTER 2
```

```
; PARITY--PARITY SUBROUTINE
; COUNTS THE NUMBER OF 'ONES' IN THE ACCUMULATOR.
; SETS THE 'PARITY ERROR FLAG' IF IT FINDS A CHARACTER WITH AN EVEN NUMBER OF ONE  BITS.

PARITY_CK:
      MOV R4,#0 ;INITIALIZE THE 'ONES' BIT COUNTER TO ZERO
PARITY1: CLR C ;CLEAR THE CARRY
PARITY2: JZ PARITY3 ;SKIP TO PARITY3 IF ACCUMULATOR ZERO, DONE, GO TO PARITY3
      RRC A ;ROTATE THE ACCUMULATOR RIGHT, THROUGH THE CARRY
      JNC PARITY2 ;IF THE CARRY IS 'ZERO', GO TO PARITY2
      INC R4 ;OTHERWISE INCREMENT THE '1' COUNTER & GO TO PARITY1
      JMP PARITY1 ;GO TO PARITY1 AND CONTINUE
PARITY3: MOV A,R4 ;PUT THE 'ONES' COUNT INTO THE ACCUMULATOR
      JB ACC.0 PARITY4 ;IF BIT 0 IS A 1 (ODD), CHARACTER IS OK, RETURN
      SETB PARITY_FLAG ;PARITY IS EVEN, SO SET ERROR FLAG
PARITY4: RET ;RETURN

ERROR:   ;THE READ ERROR ROUTINE IS ENTERED AT THIS POINT.
```

INSERTION READERS

While it is possible for the card reader to read data on either the insertion or withdrawal stroke, it should be noted that card reading is most reliable during the card withdrawal stroke. For this reason MagTek recommends that customer's software be designed to emphasize data capture during the card withdrawal stroke.

For the most reliable operation:

1. Read the card upon insertion. When the card hits the backstop (Card Present will go high), check the data for errors.

2. If the data contains no errors, output that data, start sentinel first, after the card has been withdrawn.

3. If an error was detected, clear that data and get ready to read the card again upon withdrawal.

4. After the card is removed (Card Present will go high), check the data for errors.

5. If no errors are found, output that data, start sentinel first.

6. If an error is detected, indicate to the host that an error was detected, indicate to the customer to insert the card again.

SECTION 5. BIDIRECTIONAL READING AND AMPLITUDE VARIATIONS

READING IN ONE DIRECTION VERSES BOTH DIRECTIONS

1. The simplest way to read a card is in one direction only, with the Start Sentinel read first. If this method is chosen, normally each character is stored in one byte of memory. During the debug operation this Is a simple and clear method to see what is happening using 'In Circuit Emulation' (ICE).

2. If reading in both directions is desired, the normal method is to scan the data in the forward direction and if no errors are found, then the data is sent. If an error is found, then the scan is done in the reverse direction. And if no error is found the data is sent in the forward direction (Start Sentinel first). If an error is found in both directions, then an error is Reported.

3. To be able to process the data in both directions, it is necessary to pack the data into RAM without any unused bits between characters. In the case of 5 bit data, the first byte will have the first character plus 3 bits of the Second character. The second byte will have 2 bits of the second character, 5 bits of the third character and one bit of the fourth character, etc. With the data not being on byte boundaries, it can be difficult to decode it and especially difficult when looking at the data using ICE.

4. When a card is read in the forward direction, the Start Sentinel is first, and the Start Sentinel was chosen so that the first bit would be a one. This makes it simple to tell when you have reached the end of leading clocking zeros and the first data bit. It also allows for the first bit to be on a Byte boundary.

   When a card is read in the reverse direction, there is no way to tell the boundary between the trailing zeros (which are now acting as the leading zeros) and the LRC character. For example, if the LRC is 1 (00001), then the first one bit will be encountered as the fifth bit of the message, and there is no easy way to tell when the trailing zeros stop and the LRC starts:

```
               B      5      F      1
000000000,11010,10101,11111,10000,00000000
```

which really looks like this:

```
00000000011010101011111110000000000000
```

As you can see if you are scanning right to left, it is not easy to tell where the clocking zeros stop and the message begins.

It is necessary to write a program that will:

1) go through memory, starting from the end that has the start sentinel, and

2) decode the characters (which may be forward or backwards depending on which way the card was swiped and not on a byte boundary), and

3) check for errors then transmit the characters forward back To the host.

Another problem is that the LRC will assume the same value as the Start Sentinel some of the time, and when it does, then the character adjacent to it has to be tested to see if it is an end sentinel. If it is an end sentinel, then this character is an LRC otherwise it is a Start Sentinel. In this way you can make sure which end is really the beginning. See Appendix D, Bidirectional Reading.

CARD READBACK AMPLITUDE VARIATIONS

Causes of readback amplitude variations are as follows:

- Speed of the card - the faster, the more amplitude
- Head contact, head pressure, smoothness of stripe, bumps
- Quality of oxide
- Orientation of oxide (polarization)
- Thickness of oxide
- Type of oxide, low or high coercivity

To check the amplitude variations, connect a scope between ground (pin 5) and drive or gain set (pin 9 or 11), on the F2F read/decode IC. Swipe an encoded card, and check the scope for the variations above.

The waveshape should be similar to Figure 2-1. For pin locations refer to Appendix B, Connection diagram (Pin Outs).

APPENDIX A. CHARACTER CONVERSION

TRACK 1 CHARACTER SET

	Card Data							ASCII			
Character	P	B_6	B_5	B_4	B_3	B_2	B_1	P0	P1	EP	OP
Space	1	0	0	0	0	0	0	20	A0	A0	20
! (ND)	0	0	0	0	0	0	1	21	A1	21	A1
" (ND)	0	0	0	0	0	1	0	22	A2	22	A2
# (OG)	1	0	0	0	0	1	1	23	A3	A3	23
$	0	0	0	0	1	0	0	24	A4	24	A4
% (SS)	1	0	0	0	1	0	1	25	A5	A5	25
& (ND)	1	0	0	0	1	1	0	26	A6	A6	26
' (ND)	0	0	0	0	1	1	1	27	A7	27	A7
(	0	0	0	1	0	0	0	28	A8	28	A8
)	1	0	0	1	0	0	1	29	A9	A9	29
* (ND)	1	0	0	1	0	1	0	2A	AA	AA	2A
+ (ND)	0	0	0	1	0	1	1	2B	AB	2B	AB
, (ND)	1	0	0	1	1	0	0	2C	AC	AC	2C
-	0	0	0	1	1	0	1	2D	AD	2D	AD
.	0	0	0	1	1	1	0	2E	AE	2E	AE
/	1	0	0	1	1	1	1	2F	AF	AF	2F
0	0	0	1	0	0	0	0	30	B0	30	B0
1	1	0	1	0	0	0	1	31	B1	B1	31
2	1	0	1	0	0	1	0	32	B2	B2	32
3	0	0	1	0	0	1	1	33	B3	33	B3
4	1	0	1	0	1	0	0	34	B4	B4	34
5	0	0	1	0	1	0	1	35	B5	35	B5
6	0	0	1	0	1	1	0	36	B6	36	B6
7	1	0	1	0	1	1	1	37	B7	B7	37
8	1	0	1	1	0	0	0	38	B8	B8	38
9	0	0	1	1	0	0	1	39	B9	39	B9
: (ND)	0	0	1	1	0	1	0	3A	BA	3A	BA
; (ND)	1	0	1	1	0	1	1	3B	BB	BB	3B
< (ND)	0	0	1	1	1	0	0	3C	BC	3C	BC
= (ND)	1	0	1	1	1	0	1	3D	BD	BD	3D
> (ND)	1	0	1	1	1	1	0	3E	BE	BE	3E
? (ES)	0	0	1	1	1	1	1	3F	BF	3F	BF

	Card Data							ASCII			
Character	P	B_6	B_5	B_4	B_3	B_2	B_1	P0	P1	EP	OP
@ (ND)	0	1	0	0	0	0	0	40	C0	C0	40
A	1	1	0	0	0	0	1	41	C1	41	C1
B	1	1	0	0	0	1	0	42	C2	42	C2
C	0	1	0	0	0	1	1	43	C3	C3	43
D	1	1	0	0	1	0	0	44	C4	44	C4
E	0	1	0	0	1	0	1	45	C5	C5	45
F	0	1	0	0	1	1	0	46	C6	C6	46
G	1	1	0	0	1	1	1	47	C7	47	C7
H	1	1	0	1	0	0	0	48	C8	48	C8
I	0	1	0	1	0	0	1	49	C9	C9	49
J	0	1	0	1	0	1	0	4A	CA	CA	4A
K	1	1	0	1	0	1	1	4B	CB	4B	CB
L	0	1	0	1	1	0	0	4C	CC	CC	4C
M	1	1	0	1	1	0	1	4D	CD	4D	CD
N	1	1	0	1	1	1	0	4E	CE	4E	CE
O	0	1	0	1	1	1	1	4F	CF	CF	4F
P	1	1	1	0	0	0	0	50	D0	50	D0
Q	0	1	1	0	0	0	1	51	D1	D1	51
R	0	1	1	0	0	1	0	52	D2	D2	52
S	1	1	1	0	0	1	1	53	D3	53	D3
T	0	1	1	0	1	0	0	54	D4	D4	54
U	1	1	1	0	1	0	1	55	D5	55	D5
V	1	1	1	0	1	1	0	56	D6	56	D6
W	0	1	1	0	1	1	1	57	D7	D7	57
X	0	1	1	1	0	0	0	58	D8	D8	58
Y	1	1	1	1	0	0	1	59	D9	59	D9
Z	1	1	1	1	0	1	0	5A	DA	5A	DA
[(ND)	0	1	1	1	0	1	1	5B	DB	DB	5B
\ (ND)	1	1	1	1	1	0	0	5C	DC	5C	DC
] (ND)	0	1	1	1	1	0	1	5D	DD	DD	5D
^ (FS)	0	1	1	1	1	1	0	5E	DE	DE	5E
_ (ND)	1	1	1	1	1	1	1	5F	DF	5F	DF

TRACK 2 AND 3 CHARACTER SET

	Card Data					ASCII			
Character	P	B_4	B_3	B_2	B_1	P0	P1	EP	OP
0	1	0	0	0	0	30	B0	30	B0
1	0	0	0	0	1	31	B1	B1	31
2	0	0	0	1	0	32	B2	B2	32
3	1	0	0	1	1	33	B3	33	B3
4	0	0	1	0	0	34	B4	B4	34
5	1	0	1	0	1	35	B5	35	B5
6	1	0	1	1	0	36	B6	36	B6
7	0	0	1	1	1	37	B7	B7	37

	Card Data					ASCII				
Character	P	B_4	B_3	B_2	B_1	P0	P1	EP	OP	Hex Character
8	0	1	0	0	0	38	B8	B8	38	
9	1	1	0	0	1	39	B9	39	B9	
: (AS)	1	1	0	1	0	3A	BA	3A	BA	A
; (SS)	0	1	0	1	1	3B	BB	BB	3B	B
< (ND)	1	1	1	0	0	3C	BC	3C	BC	C
= (FS)	0	1	1	0	1	3D	BD	BD	3D	D
> (ND)	0	1	1	1	0	3E	BE	BE	3E	E
? (ES)	1	1	1	1	1	3F	BF	3F	BF	F

P = Parity
EP = Even Parity
OP = Odd Parity
P0 = Parity bit set to 0
P1 = Parity bit set to 1
SS = Start Sentinel
ES = End Sentinel
FS = Field Separator
AS = Account Separator (Track 3 only)
ND = Character Not Defined by Credit Card Standards
OG = Optional Graphic

MAGTEK®

REGISTERED TO ISO 9001:2000

20725 South Annalee, Carson, CA. 90746
Phone: (310) 631-8602, Help Line (651) 415-6800
Fax (310) 631-3956 http://www.magtek.com
P/N 99875065-4, 4/03Copyright 2000-2003, Mag-Tek®, Inc

APPENDIX B. READ IC 21006516 SPEC SHEET

MAGTEK®

21006516 SMD (SOIC)

Magnetic Stripe F/2F Read/Decode Integrated Circuit

General Information

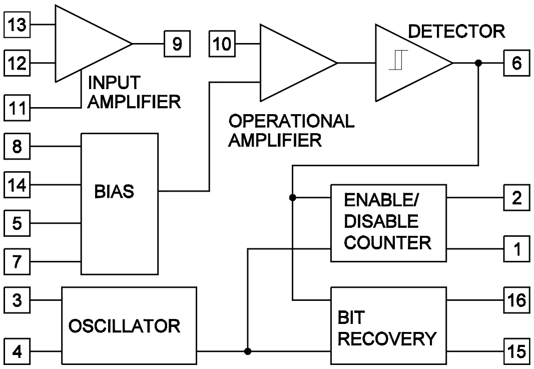
The F/2F Read/Decode Integrated Circuit in intended for use in recovering F/2F encoded data received from a magnetic head.

Features

- 150 to more than 12,000 F/2F bits per second.*
- Low power: Full operation from 2.7 to 5.5 volts. Current below 2.0 mA.
- Recovers Data with as much as 30% dropout of amplitude.*
- Accepts amplitudes from below 20% of ISO reference voltage to more than 250% of ISO reference voltage.*
- Meets or exceeds the requirement for:
  IEC 1000-4-2 ESD (Electro Static Discharge)
  IEC 1000-4-3 Radiated EMC Field (2x requirement)
  IEC 1000-4-4 Electrical Fast Transient Burst requirement (transmissions on I/O cable)

* When used with MagTek read heads and recommended circuit.

Functional Block Diagram



Functional Description

The F/2F Read/Decode I.C. will recover clock and data signals from an F/2F data stream generated from a magnetic head. The I.C. will function for data rates from 150 to more than 12,000 bits per second. Acquisition and tracking of the data within this range is automatic. The F/2F Read/Decode I.C. is composed of three functional sections:

- Signal Conditioning and Detection
- Bit Recovery
- Enable/Disable Counters

The signal conditioning and detection section amplifies and filters the signal from the magnetic head, rejects common mode noise, and detects the signal peaks. Other features include protection against certain waveform distortions that may be present in the signal.

The enable/disable counters provide initialization for the recovery section. These counters initialize both the bit recovery and the signal conditioning and detection sections.

The Bit Recovery section locks onto the data rate and performs the recovery of individual bits from the F/2F data stream.

Absolute Maximum Ratings (Non-Operating)

Supply Voltage	7.0 Volts
Voltage Input Range	0 to VCC
Output Sink Current	10 mA
Internal Power Dissipation	100 mW
Storage Temperature Range	-55 to 150°C
Lead Soldering (10 Sec.)	260°C

Electrical Characteristics

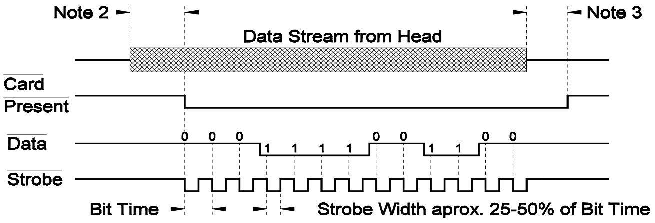
Unit operates from 2.7 to 5.5 Volts (VCC) and -30° to 70°C
Test conditions: VCC = 5.00 Volts, Ambient Temperature = 25°C

Table 1: Electrical Characteristics.

Parameter	Condition	Min.	Max.	Typ.	Units
Device Current			2		mA
Logic Low Out (VOL)	at +2.0 mA [1]		0.4		Volts
Logic High Out (VOH)	at -2.0 mA [1]	VCC - 0.5			Volts
Oscillator Frequency	([2])			1.2	MHz

Notes:
1. TTL/CMOS compatible. Outputs covered include the following: Card Present, Read Data, Strobe and F/2F.
2. Frequency measured using R2 and C3 component values. Refer to recommended circuit for required values.

Signal Timing Diagram



Notes:
1. Card Present, Data, and Strobe are negative true logic.
2. Card Present goes low after 14-15 head flux reversals.
3. Card Present returns to high level approximently 150mS after the last flux transition.
4. Data is valid 1.0µS (min.) before the leading negative edge of strobe and remains valid until approximently 1.0µS before the next STROBE.

DATA

The DATA signal is valid while the STROBE is low. If the DATA signal is high, the bit is a zero. If the DATA signal is low, the bit is a one.

STROBE

The STROBE signal indicates when DATA is valid. It is recommended that DATA be loaded by the user with the leading negative edge of the STROBE.

CARD PRESENT

CARD PRESENT will go low after 14-15 flux reversals from the head. It will return high if Reset or about 150 milliseconds after the last flux reversal. The CARD PRESENT signal can be tied together with other card present signals from more than one IC, however use only one pull-up resistor (R4). E.g., if this is a 3 track reader, all 3 CARD PRESENTS would be tied together using a single 10K resistor.

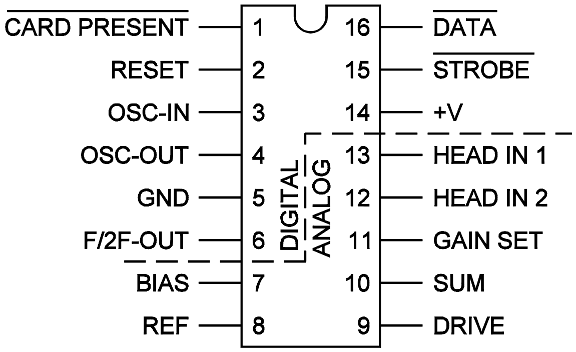
When no card is being moved through the unit, the DATA, STROBE and CARD PRESENT signals are high.

The signal timing diagram shown above represents the data along with other signals that are generated during the reading process.

Reset Feature

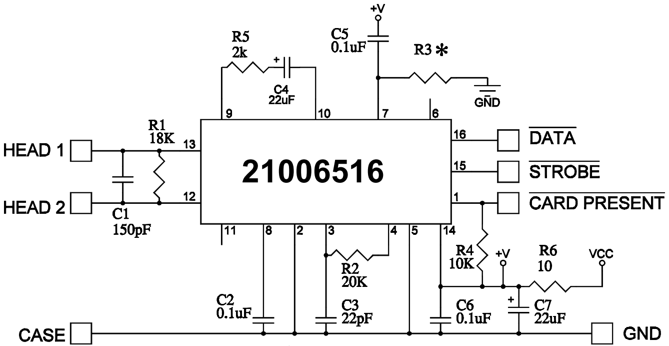
In normal operation, the I.C. resets itself approximately 150 milliseconds after the last flux reversal from the magnetic head. Reset may be forced by applying a pulse of 1.0 to 100 microseconds to the RESET pin. The positive edge of the pulse will reset the I.C.

Connection Diagram (Pin Outs)



Recommended Circuit

This circuit is intended for use in systems employing F/2F data such as ID cards that conform to ISO 7811-2 or 7811-6.



VCC May be +2.7V to +5.5V

Tolerances: Resistors ±5%; C1, C3 ±5% NPO; other capacitors non-critical.

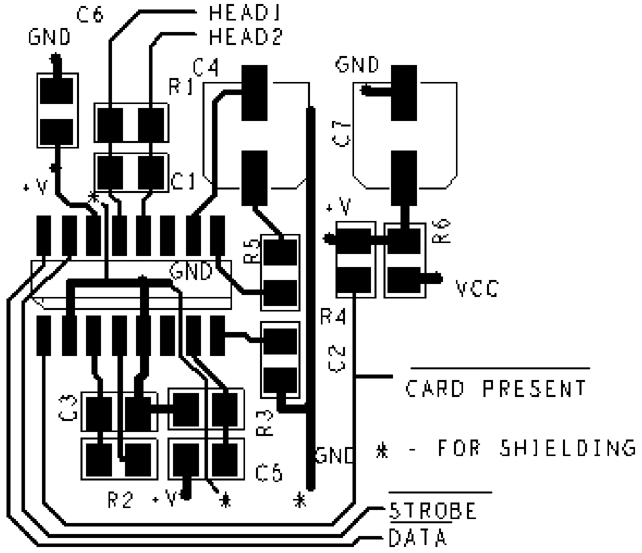
* R3 has different values for Rev B and Rev G. The table bellow shows the value of R3.

Table 2: Value of R3.

	Rev B	Rev G	Rev H
R3	470K	1.5M	1.5M

Printed Circuit Board Layout Requirements

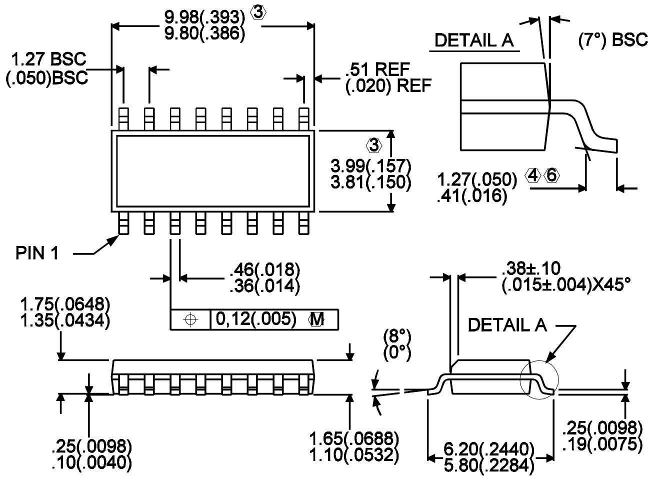
The I.C. is a combined digital and analog system. The analog signal pins are very low level. Good layout practice requires that there be a separation of head and other analog signals from the digital outputs. The digital signals are: DATA, STROBE, CARD PRESENT, F/2F OUT, OSC-OUT and OSC-IN. The analog signals in order of sensitivity are HEAD in 1&2, SUM, DRIVE, GSR and BIAS. The digital signals must not pass near the first four analog signals. The layout below illustrates the requirements.



Caution

Keep Data, Strobe, and Card Present signals away from Head 1 and Head 2 signals.

SOIC Physical Dimensions



Notes: Unless otherwise specified

1. Dimensions in mm (in.)
2. Dimension and tolerancing per ANSI Y14.5m-1982
3. Reference Datum (Mold flash not included).
4. Length of terminal for soldering.
5. Flammability rating UL 94V-0
6. Thermal resistance JA=225, JC=190

MAGTEK®

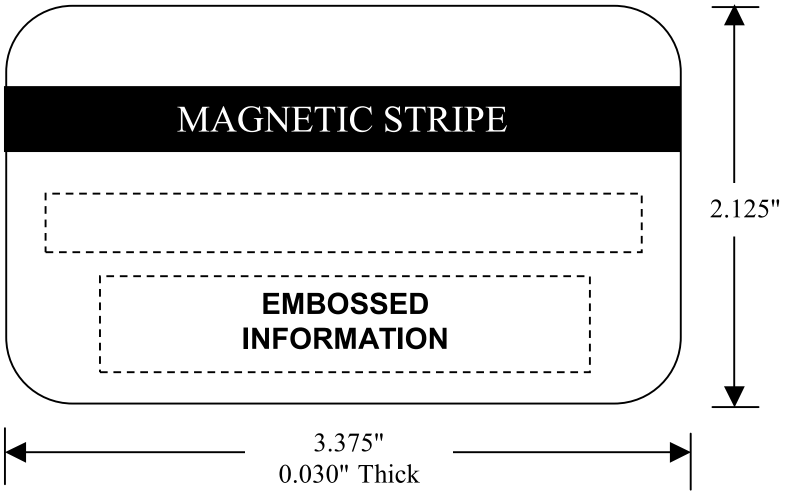
REGISTERED TO ISO 9001:2000

20725 South Annalee, Carson, CA. 90746
Phone: (310) 631-8602, Help Line (651) 415-6800
Fax (310) 631-3956
http://www.magtek.com
P/N 99821066-11, 6-24-03, Copyright© 1998 -2003,
MagTek, Inc.

APPENDIX C. STANDARDS

DIMENSIONS – FINANCIAL TRANSACTION CARDS

ISO	7810	Physical Characteristics 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



For Copies of Specifications contact:
AMERICAN NATIONAL STANDARDS INSTITUTE
25 W. 43rd Street, New York, New York 10036
(212) 642-4900

MAGNETIC STRIPE ENCODING – FINANCIAL TRANSACTION CARDS

0.223"	RECORDING DENSITY (bits per inch)	CHARACTER CONFIGURATION (including parity bit)	INFORMATION CONTENT (including control characters)
0.100" Track 1 IATA	210BPI	7 BITS PER CHARACTER	79 APHANUMERIC CHARACTERS
0.100" Track 2 ABA	75BPI	5 BITS PER CHARACTER	40 NUMERIC CHARACTERS
0.100" Track 3 Thrift	210BPI	5 BITS PER CHARACTER	107NUMERIC CHARACTERS

CARD DATA FORMAT - TRACK 1

76 ALPHANUMERIC DATA CHARACTERS

SS	FC	PAN	FS	NAME	FS	ADDITIONAL DATA	DISCRETIONARY DATA	ES	LRC

Primary Account Number (19 digits max.)

Name (26 alphanumeric characters Max.)

	No. of Characters
Expiration Date (YYMM)	4
Service Code	3

	No. of Characters
*PVKI	1
*PVV	4
*CVV or *CVC	3

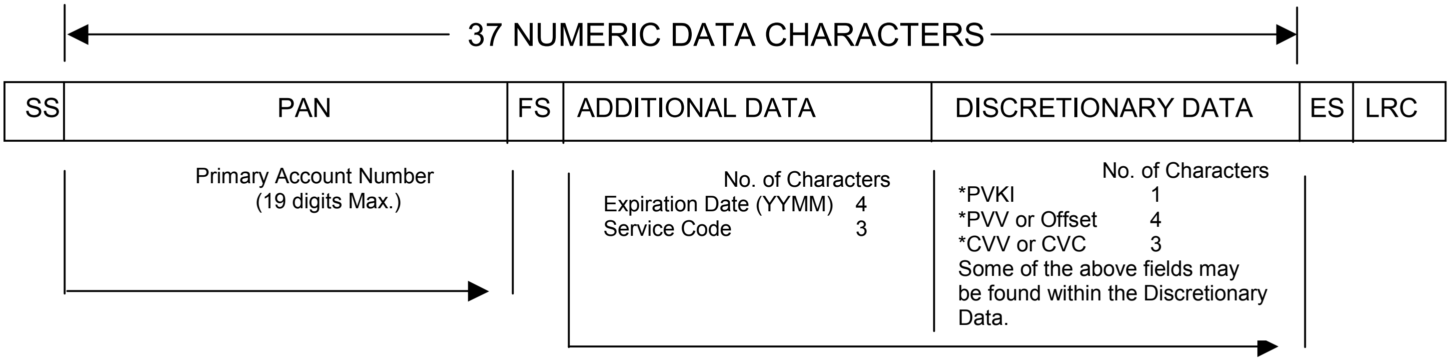
Some or all of the above fields may be found within the Discretionary Data.

Control Characters:
SS = Start Sentinel %
FS = Field Separator ^
ES = End Sentinel ?

FC = Format Code
LRC = Longitudinal Redundancy Check Character

*(PKIV) = PIN Verification Key Indicator
*(PVV) = PIN Verification Value
*(CVV) = Card Verification Value
*(CVC) = Card Validation Code

CARD DATA FORMAT – TRACK 2



	No. of Characters
Expiration Date (YYMM)	4
Service Code	3

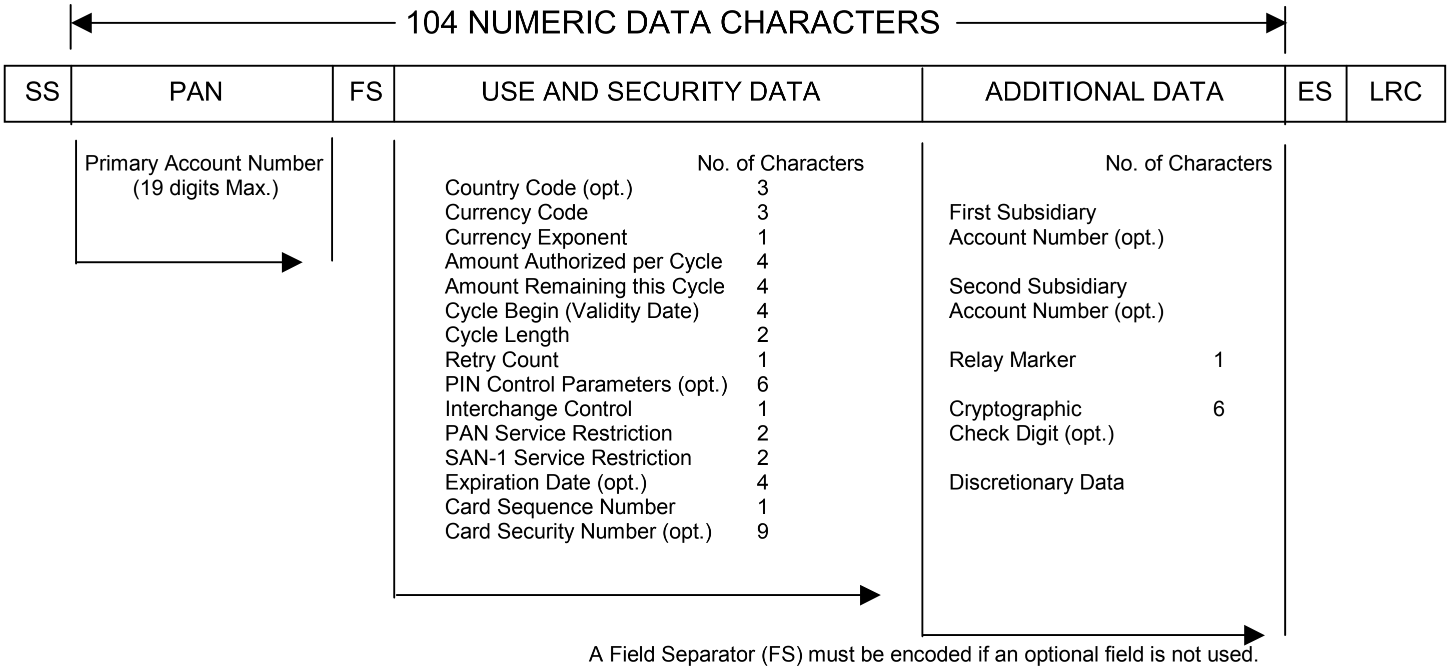
	No. of Characters
*PVKI	1
*PVV or Offset	4
*CVV or CVC	3

Some of the above fields may be found within the Discretionary Data.

SS = Start Sentinel	Hex B	;	ES = End Sentinel	Hex F	?
FS = Field Separator	Hex D	=	Longitudinal Redundancy Check Character		

*(PKIV) = PIN Verification Key Indicator
*(PVV) = PIN Verification Value
*(CVV) = Card Verification Value
*(CVC) = Card Validation code

CARD DATA FORMAT – TRACK 3 (ISO 4909)



	No. of Characters
Country Code (opt.)	3
Currency Code	3
Currency Exponent	1
Amount Authorized per Cycle	4
Amount Remaining this Cycle	4
Cycle Begin (Validity Date)	4
Cycle Length	2
Retry Count	1
PIN Control Parameters (opt.)	6
Interchange Control	1
PAN Service Restriction	2
SAN-1 Service Restriction	2
Expiration Date (opt.)	4
Card Sequence Number	1
Card Security Number (opt.)	9

	No. of Characters
First Subsidiary Account Number (opt.)	
Second Subsidiary Account Number (opt.)	
Relay Marker	1
Cryptographic Check Digit (opt.)	6
Discretionary Data	

A Field Separator (FS) must be encoded if an optional field is not used.

SS = Start Sentinel	Hex B	;	FC = Format Code (2 digits)*(PKIV) = PIN Verification Key
FS = FS = Field Separator	Hex D	=	LRC = Longitudinal Redundancy Check Character
ES = End Sentinel	Hex F	?	

This section is derived from P/N 99800004. Copies may be obtained from MagTek.

The track formats used in this document are based on ISO Standards, however, other formats may be used. Contact your card issuer for your exact requirements.

APPENDIX D. CARD READER INTERFACE

INTERFACING THE CARD READER TO A MICROPROCESSOR

The Mag-Tek Card Reader may be interfaced to a microprocessor unit (MPU) in a number of ways. Selection of the most suitable method will depend on the system requirements and the MPU capabilities. The two most common methods are:

1. Single-bit input programming.
2. USART (Universal Synchronous/Asynchronous Receiver - Transmitter such as NEC 82C51).

SINGLE - BIT INPUT PROGRAMMING

This method of interface does not require any external chip to implement serial data communication between the Card Reader and an MPU. This function is done through a software program that allows the MPU to transmit and receive data. This process requires some very time-critical programming. A disadvantage of this approach is that, while the processor is receiving data serially, it must totally dedicate itself to the task. Accurate timing can only be maintained if the program remains in a tight wait loop without being diverted to other functions. When programming the MPU, the timing loops required for receiving data cannot exceed the time period of the incoming data bits. Table 1 shows the data rates at card speed of 50 inches per second (ips).

	TK1	TK2	TK3
Bit Rate (bits/sec.)	10,500	3,750	10,500
Character Rate (char/sec.)	1,500	750	2,100

Table 1 data rates @ 50ips

USART

The primary advantage of the USART method is that the MPU is relieved of the critical time-dependent programming.

The CARD PRESENT signal can be connected to an input pin on the USART. The DSR input of the USART is suggested as an input of the CARD PRESENT signal to the MPU. The serial data input (RXD) of the USART receives data from the Card Reader. The clock input (RXC) of the USART is connected to the read STROBE output of the Card Reader. The USART must be set up to operate in the synchronous mode with a single sync character. This sync character must be equivalent to the Start Sentinel character of the track that is being read.

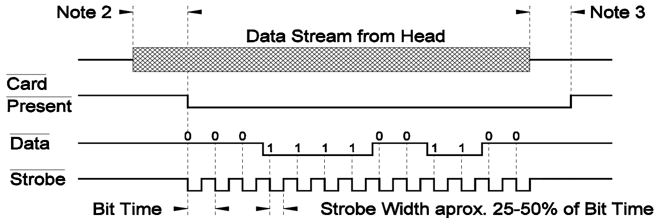
When reading Track 1, the user may set the parity on the USART to ON or OFF. If parity is set to Enabled, then the word size must be set to 6 bits. In this case the USART checks the character for parity error. If the user prefers the parity to be checked by the MPU and not the USART then parity must be set to Disabled and word size must be set to 7 bits. When reading Track 2 or Track 3, parity must be set to OFF. This is because data characters encoded on these two tracks are in 5 bit words, including parity. The USART is limited to a minimum word size of 5 bits only when parity is OFF. In this case, the USART treats the parity bit just like any other data bit, and the MPU should check for correct parity on each character.

In operation, the USART remains inactive until it recognizes the Start Sentinel character. Then it becomes active and collects the data characters, frames the data, and presents it to the MPU. (In some applications, this may not be suitable for reliable Start Sentinel detection; see the *Detecting Start Sentinel* discussion below.)

When using either method, after the CARD PRESENT signal indicates that the card is gone, the MPU should perform error detection by checking both the parity of each character and of the Longitudinal Redundancy Check (LRC) character. To perform the LRC calculation, each bit of each character excluding the parity bit should be exclusive ORed with the respective bit of all characters, including Start Sentinel and End Sentinel. Exclusive ORing the parity bit of all characters does not generate the parity bit for the LRC character; it is the parity bit for the LRC character.

The Start Sentinel and the End Sentinel characters frame data. The first bits encountered by the Card Reader are the leading zeros. They indicate to the Reader the presence of an encoded magnetic stripe card, and allow the Reader to synchronize itself with the incoming data bits. The leading zeros are followed by the Start Sentinel character, which indicates the beginning of data on a track. The characters following the Start Sentinel represent the data. The End Sentinel Character indicates the end of data. After the End Sentinel is the Longitudinal Redundancy Check (LRC) character. Trailing zeros follow the LRC and fill the remainder of the track.

NOTE: Characters are encoded on the magnetic stripe with the least significant bit recorded first. The Start Sentinel character on Track 2 is recorded as 11010. The bit pattern is B_0 B_1 B_2 B_3 P. The least significant bit is B_0. B_3 is the most significant bit. P is the parity bit. As defined in the ANSI x4.16 1983, ODD parity is required. The conventional representation of the Start Sentinel would be 01011 (P B_3 B_2 B_1 B_0) or 0Bh (hex).



Notes:

1. Card Present, Data, and Strobe are negative true logic.
2. Card Present goes low after 14-15 head flux reversals.
3. Card Present returns to high level approximately 150mS after the last flux transition.
4. Data is valid 1.0µS (min.) before the leading negative edge of strobe and remains valid until approximately 1.0µS before the next STROBE.

DETECTING START SENTINEL

In some cases, random noise on a blank track (especially High Coercivity media) may be detected as a Start Sentinel. If a Start Sentinel is erroneously detected then parity and/or LRC checking would undoubtedly indicate that the track, although blank, actually has an error on it. For this reason, it is suggested that some number of the leading zeros be included in the detection of Start Sentinel. The location of the Start Sentinel and the number of zeros required to synchronize the read circuitry determine the number of detectable zeros prior to the Start Sentinel. In the worst case, there may be no more than four leading zeros on track 2 and no more than nine on tracks 1 and 3. In other words, the Start Sentinel on track 2 should be considered as a 9-bit value: 4 bits of zero and 5 bits of Start Sentinel.

BIDIRECTIONAL READING

In many of today's magnetic stripe card applications, reading the card in both directions is necessary. Programming for bidirectional reading has its own special requirements. First, the data acquired from the reader must be treated in memory as a collection of bits (see note below). Since the direction is not known, the program must search the bits looking for the Start Sentinel. If the first character is not the Start Sentinel, then you must move to the other end of the memory and begin searching the bits in reverse order for the Start Sentinel. However, if a Start Sentinel is found, it is possible that the character is LRC. You may check for this by determining if the next character is the End Sentinel. If the next character is the End Sentinel, then you must move to the other end of memory and begin searching the bits in the reverse order as discussed above. The important point is to realize that the LRC could appear to be a false Start Sentinel.

If the message does not meet the requirements for a correct recording (the sentinels, party & LRC) then try to interpret the message in the other direction. The card is unreadable only after the software has tried both directions.

Note: For bidirectional reading, a USART cannot be used to identify the Start Sentinel. The USART must be programmed to begin capturing data based on the first binary 1 that is found. Set the word length to one byte (8 bits) and disable parity checking; for track 2, the word size should be no more than 5 bits.

CONVERTING CARD DATA TO ASCII DATA

The data transmitted or received by the user's device is in ASCII. The data encoded on the cards, however, is in a different format called *Card Data*. All RS-232 Mag-Tek equipment automatically converts transmitted and received data from ASCII to Card Data. The procedure for converting ASCII to Card Data is described bellow.

TRACK 1

Card data on Track 1 consists of six binary bits and an odd parity bit for each character. A method for converting ASCII characters to six-bit Card Data (the parity bit is not included in the calculation) is to subtract 20h (hex) from the equivalent 0 parity ASCII character (see Character Conversion Chart). For example, the ASCII character that represents the percent sign (%) is 25h. Subtract 20h from 25h and the result is 05h, which represents the six-bit portion of the card data code for the percent sign. An odd parity bit must be added to the six-bit portion of the character to complete the Card Data code. A method for converting card data to ASCII characters is to remove the parity bit from the Card Data code, then add 20h to the remaining six-bit portion of the character. The result will be the 0 parity ASCII character.

TRACK 2 & 3

Card data on Track 2 & 3 consists of four binary bits and an odd parity bit for each character. A method for converting ASCII characters to four-bit Card Data (again the parity bit is not included in the calculation) is to subtract 30h from the equivalent ASCII character. For example, the ASCII character that represents the number 7 is 37h. Subtract 30h from 37h and the result is 07h, which represents the four-bit portion of the card data code for the number 7. An odd parity bit must be added to the four-bit portion of the character to complete the Card Data code. A method for converting card data to ASCII characters is to remove the parity bit from the Card Data code, then add 30h to the remaining four-bit portion of the character. The result will be the 0 parity ASCII character.

MAGTEK®

REGISTERED TO ISO 9001:2000

20725 South Annalee, Carson, CA. 90746
Phone: (310) 631-8602, Help Line (651) 415-6800
Fax (310) 631-3956
http://www.magtek.com
P/N 99875063-4, 4/03 Copyright 1997-2003, Mag-Tek