MAGTEK

Application Note - Magnetic Stripe Reader

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. UART (Universal Asynchronous Receiver - Transmitter).

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

UART

The primary advantage of the UART 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 UART. The DSR input of the UART is suggested as an input of the CARD PRESENT signal to the MPU. The serial data input (RXD) of the UART receives data from the Card Reader. The clock input (RXC) of the UART is connected to the read STROBE output of the Card Reader. The UART 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.

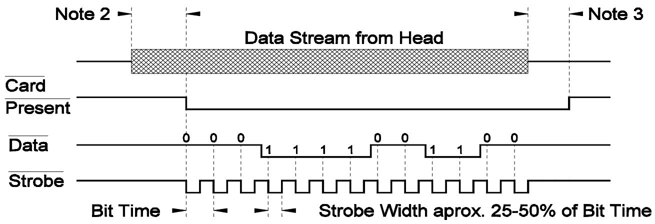
When reading Track 1, the user may set the parity on the UART to ON or OFF. If parity is set to Enabled, then the word size must be set to 6 bits. In this case the UART checks the character for parity error. If the user prefers the parity to be checked by the MPU and not the UART then parity must be 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 UART is limited to a minimum word size of 5 bits only when parity is OFF. In this case, the UART treats the parity bit just like any other data bit, and the MPU should check for correct parity on each character.

In operation, the UART 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.

Bi-directional Reading

In many of today's magnetic stripe card applications, reading the card in both directions is necessary. Programming for bi-directional 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 this 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. 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 can be considered unreadable if the message fails after the software has tried both directions.

Note: For bi-directional reading, a UART cannot be used to identify the Start Sentinel. The UART 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
20725 South Annalee, Carson, CA. 90746
Phone: (310) 631-8602, Help Line (888) 624-8350
Fax (310) 631-3956
http://www.magtek.com
P/N 99875063-2, 7/97 © Copyright 1997, Mag-Tek, Inc.