Circle 522

Interface A TTL Magcard Reader To The PC Games Port

PATRICK GUEULLE
55, Rue de Richelieu, B.P. 279, 76055 Le Havre, Cedex, France;
e-mail: pgueulle%siam@cal.fr.

The connection of magnetic-card readers to PC-compatible computers is usually made through a serial port or a keyboard interface. In this configuration, the computer merely receives ASCII strings, but the decoding process must be supported by a microcontroller embedded into the rather expensive "intelligent" card reader. Another design option is to use a much cheaper "TTL" reader and write some PC software to process the raw binary data from the card.

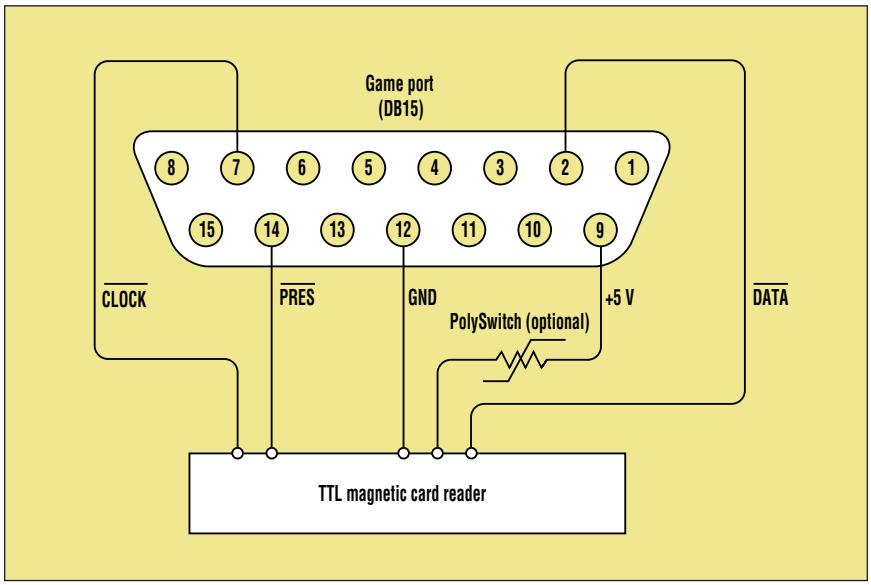
Common single-track magcard readers come with just five wires, two for the 5-V power supply and three for the logic outputs: "card present," "data" and "clock." The "games port" or "joystick connector" of most PCs is seldom used for "serious" tasks, but lends itself very well to the present application because it offers a good +5-V supply and four logic input lines. The necessary connections are shown, including an optional "PolySwitch" resettable fuse inserted into the +5-V line to protect the motherboard tracks against accidental overcurrent conditions *(see the figure)*.

The software program was written in Turbo-Pascal and proved to work satisfactorily, under MS-DOS, on CPUs running at 25 MHz or faster *(see the listing)*. Operation under Windows isn't recommended, at least on CPUs slower than 200 MHz, since the main routine is time-critical.

Once run, the program waits for a card being swept into the reader, then it displays the decoded contents of its ISO-2 track (the most commonly used one, carrying a maximum of 40 numeric characters). The ISO-3 (max. 107 numeric characters) and ISO-1 (max. 79 alphanumeric characters) tracks could be treated very similarly with appropriate readers.

The program assumes that the card is encoded in accordance with the 5-bit ANSI code, but it's written in such a way that it could easily be adapted, on a

```
(* TTL Magcard Reader              Copyright 1997 Patrick Gueulle *)
program magcard;
uses crt;

var       t: array[1..240] of byte;
          e,j,k: byte;

procedure read;
begin
    for k:=1 to 240 do
    begin
          repeat
          e:=port[513];
          until e and 32 = 0;
          t[k]:=e;
          repeat
          until port[513] and 32 = 32;
    end;
end;

procedure build;
begin
    clrscr;
    sound(880);delay(500);nosound;
    for k:=1 to 240 do
    begin
    t[k]:=t[k] and 16;
    if t[k]=0 then t[k]:=1;
    if t[k]=16 then t[k]:=0;
    end;
end;

procedure decode;
begin
     j:=0;
     repeat
     j:=j+1;
    until
    keypressed or((t[j]=1)and(t[j+1]=1)and(t[j+2]=0)and(t[j+3]=1)and(t[j+4]=0));
    repeat
    if (t[j]=1)and(t[j+1]=1)and(t[j+2]=0)and(t[j+3]=1)and(t[j+4]=0) then write(';');
    if (t[j]=1)and(t[j+1]=0)and(t[j+2]=1)and(t[j+3]=1)and(t[j+4]=0) then write('=');
    if (t[j]=1)and(t[j+1]=1)and(t[j+2]=1)and(t[j+3]=1)and(t[j+4]=1) then write('?');
    if (t[j]=0)and(t[j+1]=1)and(t[j+2]=0)and(t[j+3]=1)and(t[j+4]=1) then write(':');
    if (t[j]=0)and(t[j+1]=0)and(t[j+2]=1)and(t[j+3]=1)and(t[j+4]=1) then write('<');
    if (t[j]=0)and(t[j+1]=1)and(t[j+2]=1)and(t[j+3]=1)and(t[j+4]=0) then write('>');
    if (t[j]=0)and(t[j+1]=0)and(t[j+2]=0)and(t[j+3]=0)and(t[j+4]=1) then write('0');
    if (t[j]=1)and(t[j+1]=0)and(t[j+2]=0)and(t[j+3]=0)and(t[j+4]=0) then write('1');
    if (t[j]=0)and(t[j+1]=1)and(t[j+2]=0)and(t[j+3]=0)and(t[j+4]=0) then write('2');
    if (t[j]=1)and(t[j+1]=1)and(t[j+2]=0)and(t[j+3]=0)and(t[j+4]=1) then write('3');
    if (t[j]=0)and(t[j+1]=0)and(t[j+2]=1)and(t[j+3]=0)and(t[j+4]=0) then write('4');
    if (t[j]=1)and(t[j+1]=0)and(t[j+2]=1)and(t[j+3]=0)and(t[j+4]=1) then write('5');
    if (t[j]=0)and(t[j+1]=1)and(t[j+2]=1)and(t[j+3]=0)and(t[j+4]=1) then write('6');
    if (t[j]=1)and(t[j+1]=1)and(t[j+2]=1)and(t[j+3]=0)and(t[j+4]=0) then write('7');
    if (t[j]=0)and(t[j+1]=0)and(t[j+2]=0)and(t[j+3]=1)and(t[j+4]=0) then write('8');
    if (t[j]=1)and(t[j+1]=0)and(t[j+2]=0)and(t[j+3]=1)and(t[j+4]=1) then write('9');
    j:=j+5;
   until keypressed or (j>235);
end;

begin
     clrscr;
     writeln('Please Swipe The Card');
     read;
     build;
     decode;
     writeln;writeln;
end.
```



In this configuration, a magnetic-card reader is linked to the infrequently used "games port" of a PC.

bit-by-bit basis, to any proprietary code or to the 7-bit alphanumeric ANSI code.

It should also be noted that the parity of the data (the last bit of each character) isn't checked directly, but that the occurrence of any bit pattern other than the 16 ones listed here could easily be detected as a parity error. In addition, the last character of the track is usually an LRC, and could be used, at a later stage, to control the integrity of the data string.