Atmel AVR 2004 Design Contest
Project No: A3616

Wiegand to RS232 Converter

or How to Connect to PC an Access Control Card Reader

Abstract

Wiegand Data Interface is one of the most widespread interfaces in access control industry but it is completely unknown in PC world. Consequently, experimenting on access control devices or even simple checking whether data preprogrammed by vendors in access control tags suit the purchaser's order needs at least an access controller if not the whole access control system. Wiegand to RS232 converter solves the problem enabling direct connection of the access control card reader to PC and thus allowing an engineer to play with access control devices on his desk.

Presented in the project Wiegand to RS232 converter obtains the following features:

- support of Wiegand 26 and Wiegand 37 data formats
- automatic recognition of Wiegand 26 and Wiegand 37 data format
- checking parity of Wiegand data block
- transmission via RS232 link card data and card format
- control of the reader's LEDs and beeper.

The block diagram of Wiegand to RS232 converter is shown in figure 1 while the full schematic is shown in figure 2. As it can be seen, the circuit is the minimalistic one. It contains only the AVR microcontroller, the rest of needed hardware (e.a. full-duplex UART and Wiegand input shift register) is emulated in software. The operation of the converter is straightforward. Received from the reader data are checked for errors and then retransmitted as ASCII characters via RS232 link. In opposite direction control commands are sent on receiving which the converter switches on and off the reader's LEDs and beeper.

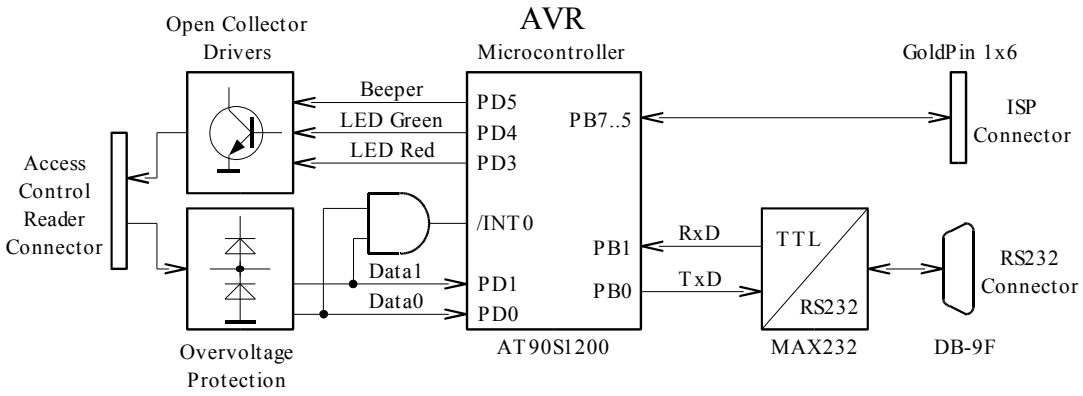

Figure 1. Block diagram of Wiegand to RS232 converter

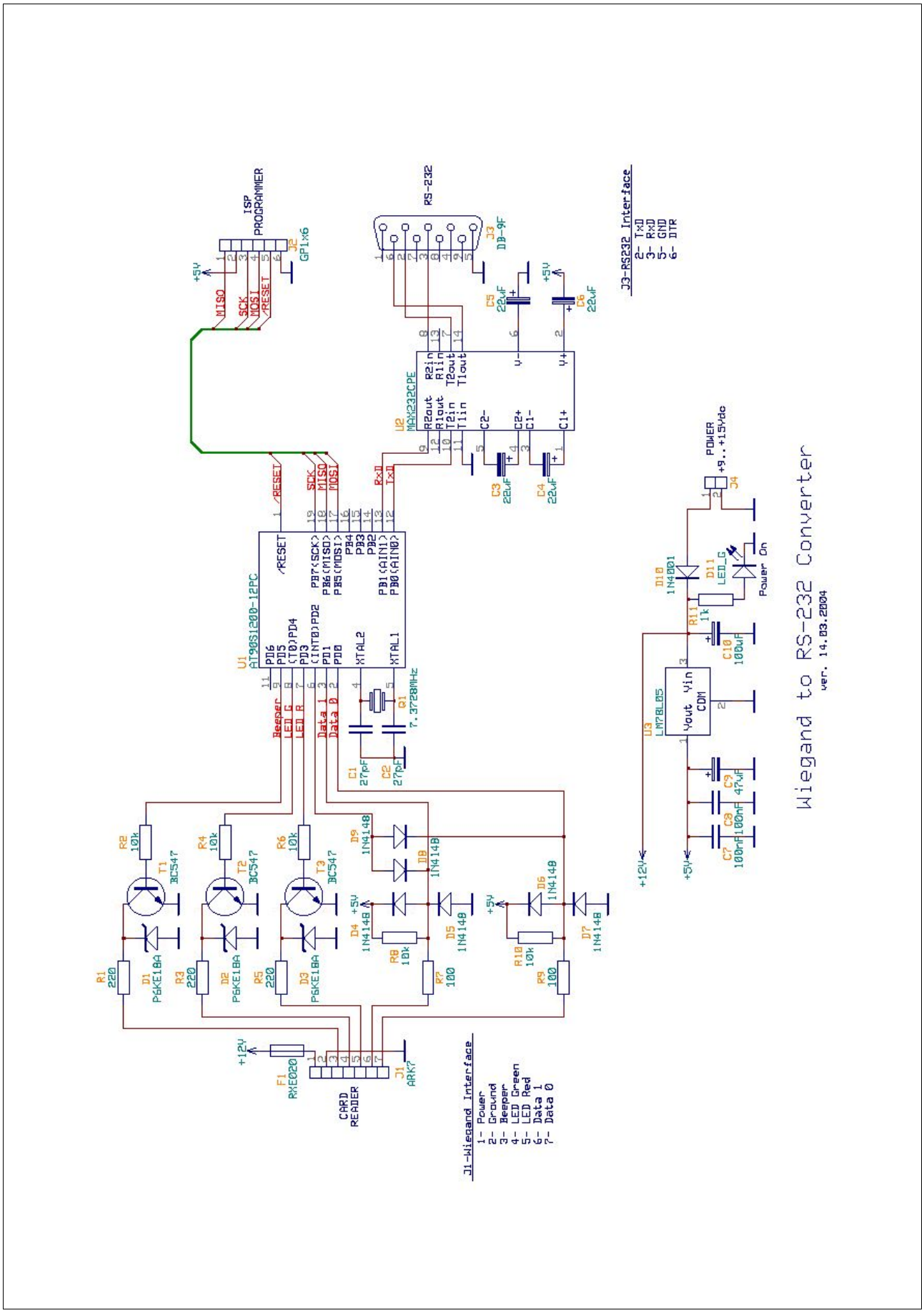

Figure 2. Schematic of Wiegand to RS232 converter

Unlike hardware, software of Wiegand to RS232 converter is rather complicated, mainly due to limited resources of the microcontroller. Nevertheless, both Wiegand and RS232 interfaces have been fully implemented. An example snippet of the converter's code is shown in figure 3.

```
;---------------------------------------------------------------------
;       External Interrupt 0 Service Subroutine (vector 0x01)
;---------------------------------------------------------------------
; Read bits are stored in Wiegand buffer Wiegand_B0..Wiegand_B4 as:
;      Wiegand_B0:   b19 b20 b21 b22 b23 b24 b25 b26(odd)
;      Wiegand_B1:   b11 b12 b13 b14 b15 b16 b17 b18
;      Wiegand_B2:   b3  b4  b5  b6  b7  b8  b9  b10
;      Wiegand_B3:   x   x   x   x   x   x b1(even) b2
;      Wiegand_B4:   x   x   x   x   x   x   x   x
;        b1  - even parity bit for bits b2..b13
;        b26 - odd parity bit for bits b14..b25
; for Wiegand26 data format and as:
;      Wiegand_B0:   b30 b31 b32 b33 b34 b35 b36 b37(odd)
;      Wiegand_B1:   b22 b23 b24 b25 b26 b27 b28 b29
;      Wiegand_B2:   b14 b15 b16 b17 b18 b19 b20 b21
;      Wiegand_B3:   b6  b7  b8  b9  b10 b11 b12 b13
;      Wiegand_B4:   x   x   x b1(even) b2 b3 b4  b5
;        b1  - even parity bit for bits b2..b19
;        b37 - odd parity bit for bits b19..b37
; for Wiegand37 data format
; The end of the data frame is detected as time_Wgnd time-out
;
ext_INT0: in      SREGcopy,SREG        ; save SREG
          in      tempINT,PIND         ; read Wiagand inputs
          andi    tempINT,0b00000011   ; extract Data_1 and Data_0 lines
;         cpi     tempINT,0x00         ; Data_1Data_0=00?
          breq    extERROR             ; error - both lines can't be low
          cpi     tempINT,0x03         ; Data_1Data_0=11?
          breq    extERROR             ; error - one line should be low
          ror     tempINT              ; read Wiegand bit (C=0 when Data_0=0
                                       ;                   C=1 when Data_1=0)
          rol     Wiegand_B0           ; \
          rol     Wiegand_B1           ; |
          rol     Wiegand_B2           ; | complete Wiegand data block
          rol     Wiegand_B3           ; |
          rol     Wiegand_B4           ; /

          inc     Wgnd_count           ; count read bits
          ldi     time_Wgnd,Wgnd_tout  ; load time-out timer

extERROR: out     SREG,SREGcopy        ; restore SREG
          reti
```

Figure 3. Wiegand interface transmission service subroutine

The prototype of Wiegand to RS232 converter has been assembled on two perf-boards, one of which contains signal conditioning and protection circuits of card reader interface while the second one carries the microcontroller, RS232 driver and voltage regulator. Originally the converter worked with MiniProx™ access control proximity card reader made by HID Corporation and such test circuit is shown in figure 4.

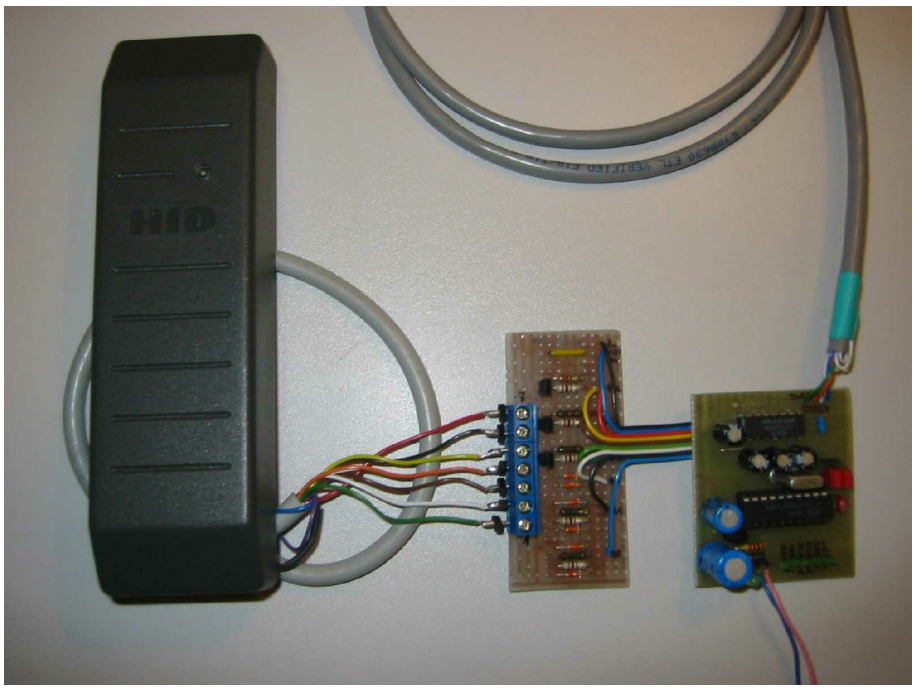

Figure 4. Complete circuit: the card reader and Wiegand to RS232 converter

As an example application of Wiegand to RS232 converter, the simple PC program called *Wiegand Card Reader Test* has been written. It was designed mainly for visualising programed data in access control tags (e.a. data format, facility code and card number), but it can be used for testing access control readers themselves. In figure 5 there is shown the screen shot of *Wiegand Card Reader Test* program during read-out of Wiegand 26 card.

Figure 5. Read-out of Wiegand 26 card data