SHARP CE-1601N Barcode Reader
Custom Replacement
*** Fully Compatible ***

(c) spellbound

Introduction

In early 2017 I was lucky enough to acquire an original SHARP CE-1601N barcode reader plus driver disk CE-1F01A. Over the last year I spent some time with disassembling and analyzing the driver code and deducing the hardware design from there, since I was interested in building a fully compatible custom replacement.

During analysis the original CE-1601N gave me some surprises:

In general there are three types of barcode readers (also called wands):

1. Dumb wand
   This type has a 1-pin digital output, which continuously indicates the wands actual reading state (black or white).
2. Smart wand
   This type does all the barcode decoding and just transfers the final result via USB or RS-232 to the host computer.
3. Semi-smart wand
   This type offers an intermediate protocol. It provides some generic pre-processing (i.e. runlength encoding) but leaves the actual decoding to the host computer.

I thought the CE-1601N might be a semi-smart wand with optical cable and RS-232 communication at 38400 baud. But in fact that's totaly wrong: The CE-1601N turned out to be a dumb wand with absolutely low level communication (1-bit direct port read by the PC-1600, no RS-232 communication at all). Furthermore the CE-1601N does not have an optical cable (like the CE-1600L). But on the other hand it is bidirectional: The driver software at the PC-1600 is able to signal a successful read, and the CE-1601N indicates that via an internal buzzer plus LED.

For those who are interested in the details, I provide my original analysis log in the appendix.

For the replacement I chose the Datalogic P51 dumb wand. It is available on ebay, reliable quite elegant and robust. In other words: A perfect fit.

As with an earlier semi-smart wand prototype I wanted to hide the hardware that is necessary to adapt the P51 to the PC-1600 into a nice custom plug (the original SHARP plug for the 5-pin SIO port was completely proprietary and there was/is no compatible 3rd party product). So after prototyping, I designed a PCB version of my adapter circuit that fits into a small, modified guitar plug. Also I was able to find the right materials to build a reliable custom 5-pin SIO-plug for the PC-1600.

Usage

Since this replacement is fully compatible with the original SHARP CE-1601N and its driver CE-1F01A, there is little to say about the usage:

Plug it into the SIO port, switch on the PC-1600 and load the original driver software.

For EAN-code this would be:

```
LOAD"<drive-letter>:EAN"
RUN
```

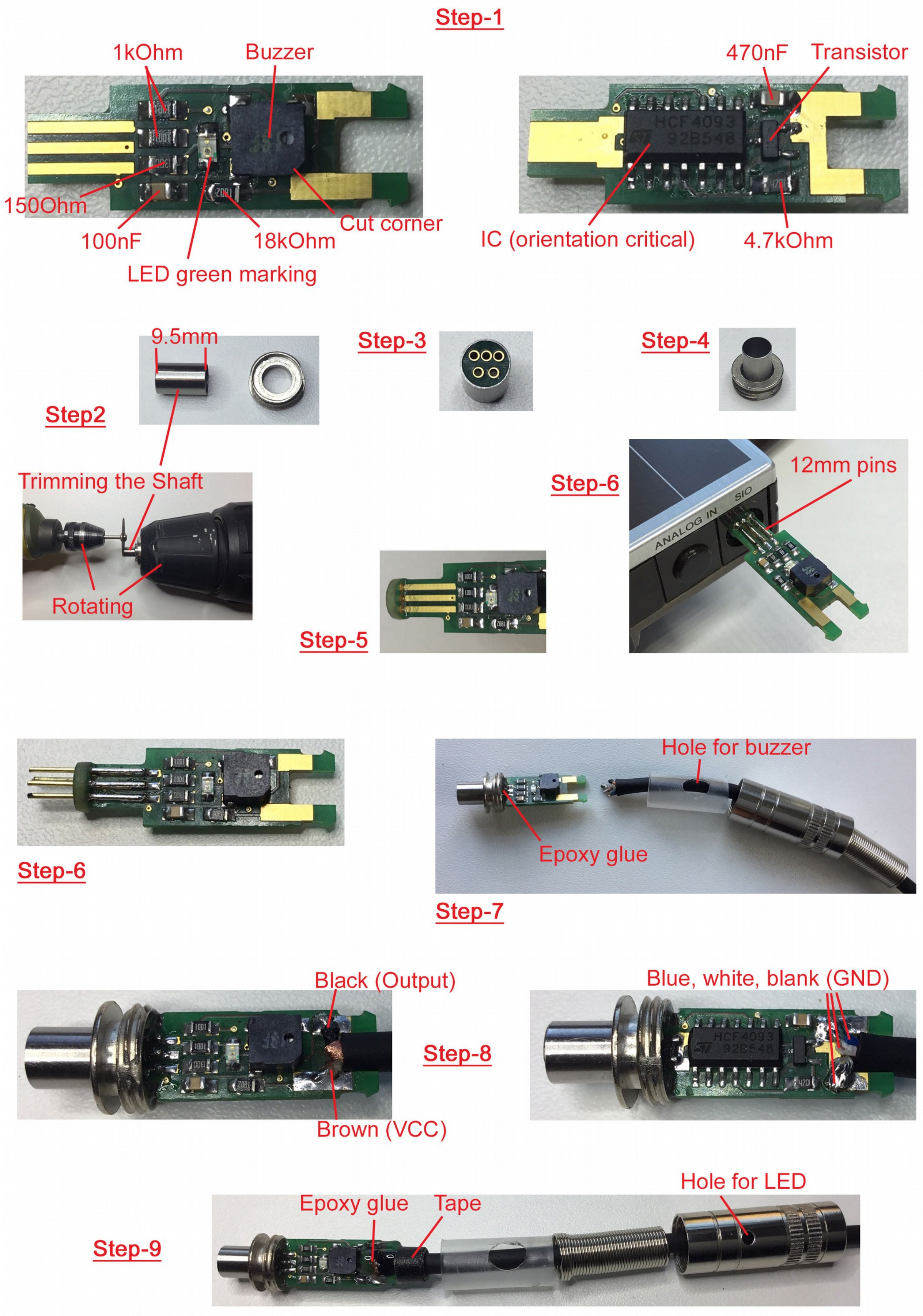
There are basically two usage modes for the CE-1F01A driver:

1. Explicit call: `CALL&FF48,<variable>$`
   This CALL blocks, until a barcode has been successfully read and decoded, or ON/BREAK has been pressed.
   On a successful read the pen (more specific: the plug of the pen) gives an optical (LED) and acoustic (buzzer) indication.
   Then the CALL terminates and the decoded string is returned into the variable specified as CALL-argument.
2. Keyboard emulation mode: CTRL $
   This key combination deactivates the keybord and takes input strings from the barcode reader instead. (CTRL % enables both keyboard and pen).

For the usage details see the original CE-1F01A manual, which is public available - e.g. here:

http://www.sharp-pc-1600.de/Download_Bedienung.html

Kit Assembly



Step-1: Main PCB Assembly

Solder all components in place as shown. Be careful not to bridge the buzzer with the 18kOhm resistor. The LED, buzzer and IC are orientation critical. Place the buzzer towards the PCBs center, so that it does not collide with the plugs shell later on.

Step-2: Shaft Trimming

Trim the shaft for the 5-pin SIO plug to 9.5mm length like shown. The combination of a battery screwdriver and a micro tool with cutting disc yields a perfect result. Additionally trim one end in an angle of 45°, so that it fits straight (90°) into the base.

Step-3: Pin-PCB Trimming

Carefully trim the pin holder PCB with a fine rasp, so that it fits into the shaft. Remove it.

Step-4: Base Assembly

Glue the shaft into the plug base with thin superglue.

Step-5: PCB Glueing

Carefully roughen the surface of the pin holder PCB between the two rows of pin holes. Then secure the pin holder PCB to the main PCB with superglue as shown.

Step-6: Pin Trimming and Adjustment

Cut five 12mm pins from the 0.6mm brass rod provided with the kit and sand them, so that there is no sharp rim. Make sure, that all pins have identical length. Push one pin into the middle upper pin jack of the SIO-port as far as possible – this will be the reference, since it is the least deep pin jack. Plug all other pins in and adjust them to the reference pin. Now attach the PCB-assembly and solder the pins to both PCBs like shown. The three top side pins must not be bridged (the two bottom side pins are bridged).

Step-7: PCB/Shaft Assembly and Wand Cable Preparation

Attach the PCB/pin subassembly to the base/shaft subassembly with 2-component epoxy glue. If the pin holder PCB has a tight fit in the shaft, you can perform this step while the whole assembly is mounted to your PC-1600, so that the plug base has a perfect fit to the computers housing. Alternatively ensure the right adjustment of the two subassemblies before the glue hardens. Line up the remaining plug components on the cable like shown. Consider drilling the LED hole into the shell now (see Step-9).

Step-8: Wand Cable Mounting

The Datalogic P51 wand has an open collector output. This output signal is normally color coded in black. The VCC input for the wand is coded brown. All other wires have to be connected to GND (main PCB bottom side). Be careful not to bridge different signals.

Step-9: Finish

Secure the cable to the main PCB, e.g. by epoxy glue. Apply some tape to the cable behind the main PCB, so that the spring gently presses against it, when the plug is closed. Close the plug and mark the LED position on the shell. Open the plug again and carefully drill a hole for the LED.

Post Assembly Tests

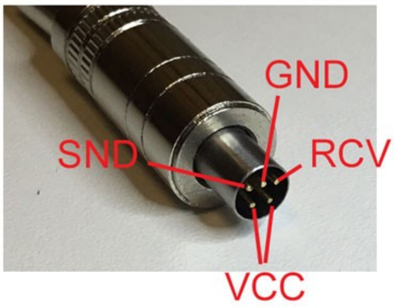
With a multimeter or similar equipment assure that there is no bridging of different input/output signals, VCC, GND and the frame-ground (shaft or shell of the plug).

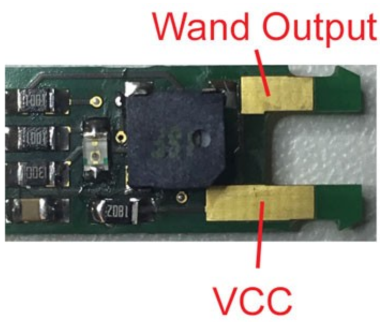
Especially check that there is no short circuit between VCC and GND.

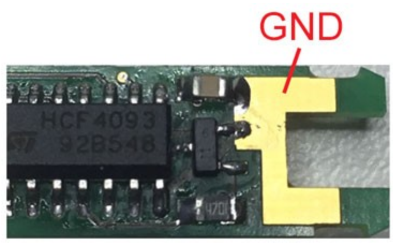
Verify that the P51 is connected to VCC and GND correctly and that the wand output is not shorted to VCC or GND.

In order to perform these tests, measure at the pins of the plug and at the PCB pads for the wires of the wand.

Here are the relevant pinouts for testing:







ATTENTION:

Do not mount the wand to the PC-1600, if any of these tests fail – otherwise your PC-1600 and/or the wand circuit may be harmed!

Appendix: Log of my CE-1601N/CE-1F01A Analysis

- 1st goal: decode the protocol between the CE-1601N and PC-1600.
- Aproach: try to read the output string of the wand via BASIC COM2:
- Result: protocol error, reproducable - no data readable from BASIC-SIO!
- Question: does the driver do a special SIO setup? if so: which one?
- Approach: disassemble driver and get some hints, before using the logic analyzer
- First findings:
  - BCR does only allocation/deallocation and driver wrapping, nothing special.
  - As expected, the actual driver sits in EXROM3 (part of the RAM, that can be reserved as extension pseudo ROM, protected from BASIC, NEW0 etc.).
  - The PC-1600 provides only one ROM-routine for UART (RS-232/SIO) communication: 01D8 and the code uses only ONE function of it: CSBRK (send a number of BRK chars).
  - NO UART DATA-BYTE RECEIVE OR TRANSMIT VIA HIGH-LEVEL ROM-CALLs. Instead the actual read access is absolutely low level via the UART STATUS port 22h: IN A,(22).
  - THE DATA PORT (20h) OF THE UART IS NOT USED.
  - THERE IS NO WRITE-OP TO THE STATUS PORT, SO NO CHANGE OF THE RS-232 PARAs (baud-rate, parity etc.).
  - ONLY BIT7 OF THE STATUS PORT IS EVALUATED (RLA, JR NC).
  - THERE IS NO CODE FOR DATA-TRANSFER VIA THE RS-232 PROTOCOL FROM CE-1601N TO PC-1600.
- Hypothesis:
  - Bit7 of port 22 is not documented in any available manual/system handbook. It seems as if it represents directly or indirectly the raw state of the SIO.RXD line.
  - The communication from PC-1600 to CE-1601N (for buzzer and LED indication) seems to be implemented by sending breaks in sequence - so very low level too.
  - The CE-1601N seems to be a DUMB wand and the PC-1600/CE-1F01A-driver seems to be able to raw-read the wands state (0 or 1) via the UART status port.

- Next goal: verify or falsify the hypothesis
- Tests and results:
    - CE-1601N connected to a breadboard, powered with 5V DC and RXD line connected to a LED.
    - Battery power supply on and wand switched on => tip illuminates
    - Wand in the air or on a black surface => LED = off
    - Wand on white surface => LED = on
    - Running a simple BASIC prog instead of driver:
      10 CLS:WAIT0
      20 B=(INP(&22) AND 128)/128
      30 PRINT B: GOTO 20
      Result:
      Pen on black surface => program prints '0', pen on white surface => program prints '1'
      The reading does not change over time, when the wand is constantly put on a white or a black surface
    - Running a simple BASIC prog:
      10 SETDEV"COM2:",PO
      20 LPRINT"this is a long string to test how the wand reacts when sent"
      Result:
      The pen's buzzer makes a short "crispy" sound, similar to that of a prog save on tape. The LED of the pen lights up and slightly flickers exactly for the same time that the buzzer sounds. The length of the sent string directly controls the length of the sound/light at the wand.
    - Observe SIO.TXD with digital analyzer.
      Result:
      During confirmation SIO.TXD goes HIGH.
- Conclusion:
    - The wand has a continuous raw output to the SIO.RXD line: BLACK = LOW = 0, WHITE = HIGH = 1.
    - Port 22h,bit7 provides a raw read of the SIO.RXD line.
    - The wand directly uses the SIO.TXD UART-signal to control the internal buzzer and the LED. There is no UART/RS-232 protocol interpretation in between.
    - Sending about 350 serial breaks (i.e. not an ASCII-char) in sequence via UART results in SIO.TXD going HIGH, which lights up the LED and drives the buzzer through an internal generator of the CE-1601N.
    - So, there is clear evidence, that
      THE CE-1601N IS A BIDIRECTIONAL DUMB-WAND WITHOUT A MICROCONTROLLER OR UART.