

SHARP PC-1600 Custom Barcode Reader

by Tom Stahl

This hack is about building a custom CE-1601N substitute including driver software for the SHARP PC-1600.



Hardware

- Datalogic P51 dumb wand (reader pen) with open collector output
- Custom build plug for SHARP 5-pin SIO port
- Custom circuit board with Atmel AVR microcontroller – Atmega328P - integrated in the plug
- SMD components (quartz, capacitors, resistors, inverter-IC, LED)

Introduction

Since I was not able to acquire an original barcode reader pen for the SHARP PC-1600 (the CE-1601N + CE-1F01A) yet, I had the idea of adopting a standard reader pen to the vintage pocket computer.

There are basically 3 types of wands (reader pens):

1. Dumb wand
This type has a 1-pin digital output (typically an open collector) which continuously indicates the wands actual reading state: black = 1, white = 0.
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 but leaves the actual decoding to the host computer.

Of course the simplest solution would be to buy a smart wand with RS-232 output and just solder it to a 15-pin plug for the pocket computer. But hey, that is not a hack and it's far away from the original CE-1601N, since the pocket wouldn't have anything relevant to do then :-(
Furthermore this solution would be limited to the barcode standards provided by the manufacturer (e.g. EAN) and it could not be used to scan other codes or completely different things (e.g. drawings).

So I thought about adopting a dumb wand to the SHARPs SIO port which is capable of 38400baud transfer rate. But there are several drawbacks:

First, even a scan-sample transfer rate of 38400baud *may* be insufficient for a reliable recognition. And second, it is very inefficient and complicated to build a hardware that "speaks" RS-232/UART without the use of a microcontroller. But if you incorporate one for the UART translation, it can do more – which leads to the approach of a semi-smart wand.

Top Level Design

The information contained in a barcode is encrypted within the width of the bars and the width of the spaces (white room between the bars).

For interfacing of a semi-smart wand (to be designed) with the PC-1600, a generic runlength encoded RS-232 data stream representing one scan line seems best. So the microcontroller shall not do any barcode specific interpretation, but for one scan line it shall send a sequence like this:

"SCAN, 50, 1033, 2345, 750, 3500, ..., 657, END"

| | | | | |
n b(1) w(1) b(2) w(2) ... b(n)

Where n is the total number of bars in the scan, $b(i)$ is the number of black raw-samples in the i -th bar and $w(i)$ the number of white raw-samples in the i -th space. So there is an alternation of $b(i)$ and $w(i)$ and after normalizing, the numbers represent a measure for the width of the bars and spaces.

There are only a few assumptions to be made:

- Continuous¹ black reads means 'idle'-state.
- The scan begins with a transition from continuous white read (i.e. 'ready'-state after 'idle') to black.
- The scan stops after a certain timeout, when no further transitions are detected.

The pocket computer then shall read those runlength encoded scan lines via RS-232 and do the actual decoding. So a specific PC-1600 driver can be provided for each barcode (e.g. an EAN-13 driver) – just like the original CE-1F01A does. Different applications can be supported in the same way, just by loading a different decoder-driver into the SHARP PC-1600.

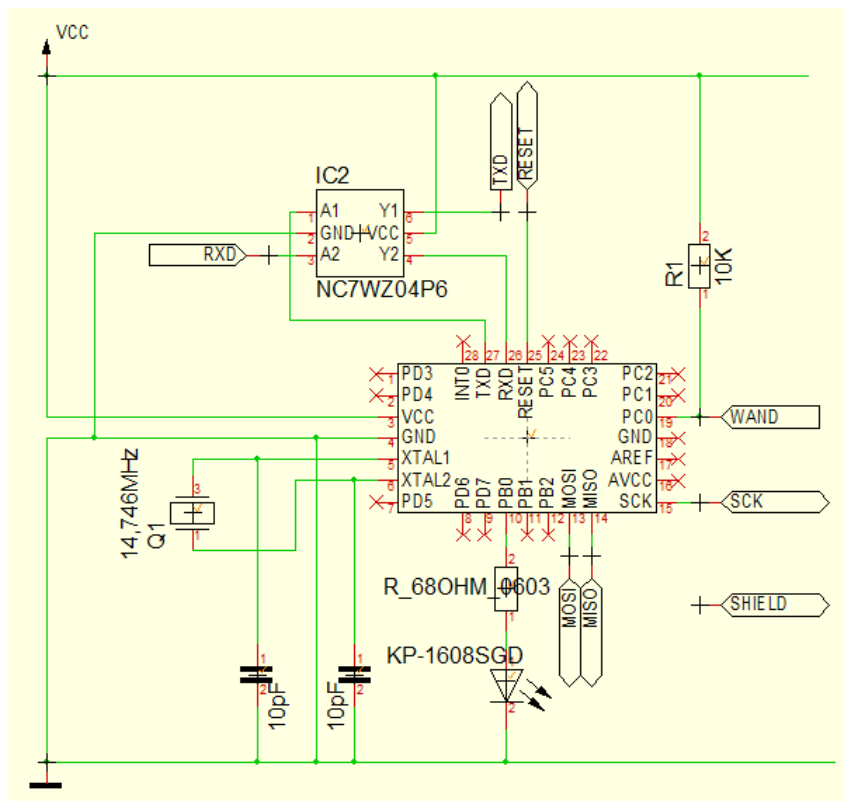
¹ Technically 'continuous' is defined by a certain minimum time threshold.

Technical Design

I chose a Datalogic P51 dumb wand and a Atmega328P microcontroller with integrated UART as the basis to build the semi-smart wand. In order to get close to the original CE-1601N, I wanted to use the 5-pin SIO jack of the PC-1600, which was intended by SHARP for optical data transfer.

In fact this port itself is *not* an optical port, but a normal TTL one with inverted UART RXD and SMD signals. The conversion to optical signals (and back to TTL) is done within the original SHARP plugs (CE-1600L and plug of CE-1601N) – so no problem here.

Here is the circuit diagram for the hardware to be integrated within a suitable custom plug:



The dumb wand has an open collector output, which is pulled-up by the 10K resistor and is directly connected to one of the microcontroller ports. The signal goes LOW on white.

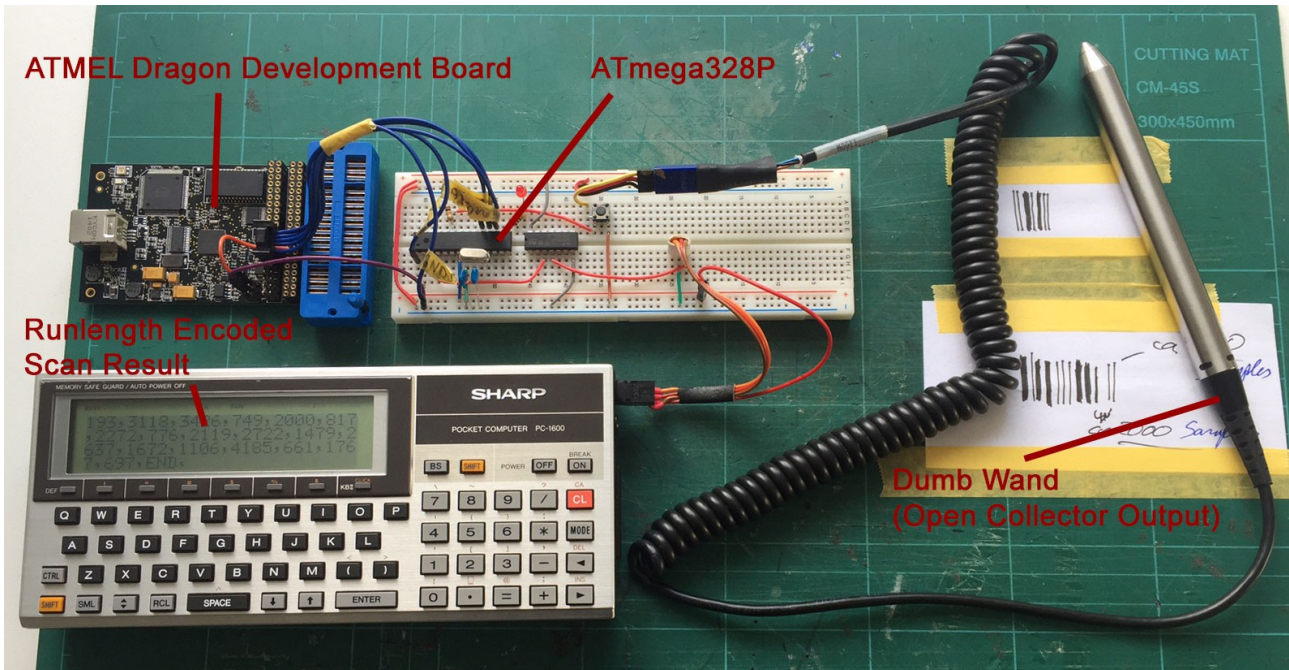
The logic inverter (IC2) is needed, because the PC-1600 signals have inverted UART TTL level.

An external quartz is provided, because the internal R/C oscillator of the Atmega wouldn't be precise and stable enough for a RS-232 communication (e.g. voltage dependent frequency).

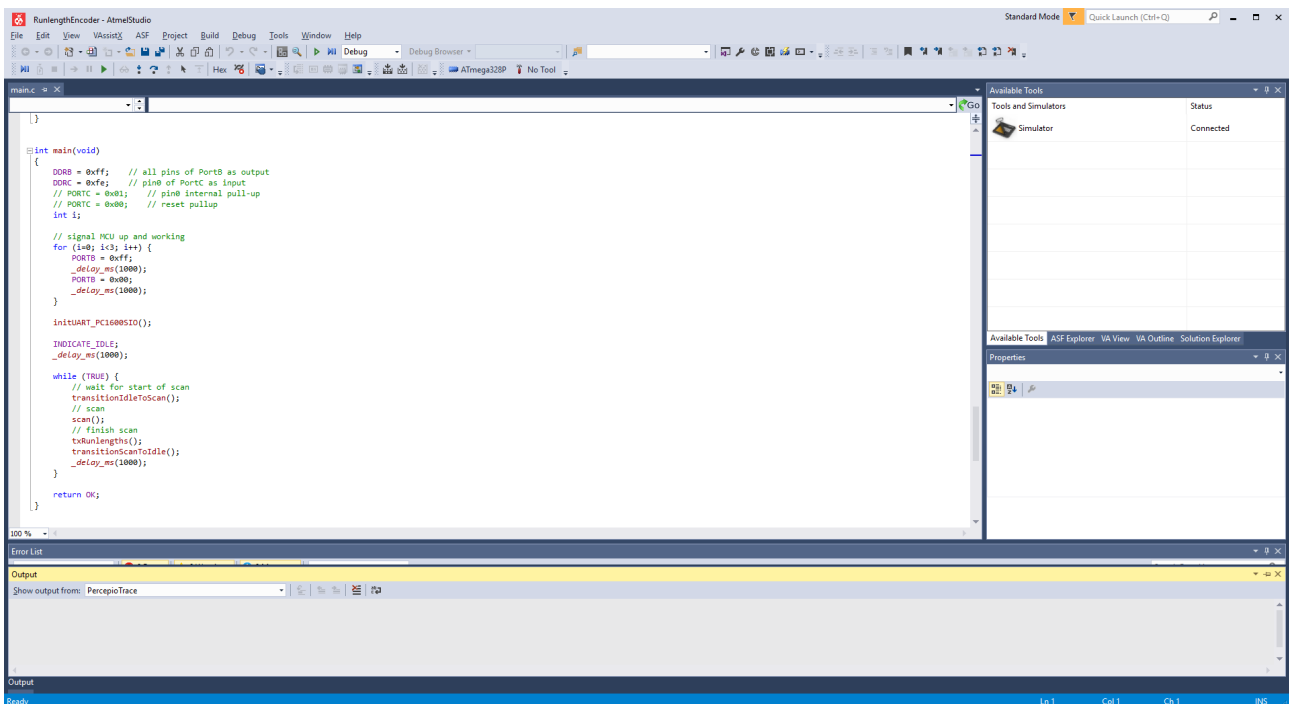
The MISO, MOSI, SCK, RESET signals together form the programming interface of the Atmega (SPI-interface for on board programming).

Prototyping

First, I built a prototype to validate the technical design:



I used the Atmel Dragon together with Atmel Studio 7.0 for the microcontroller programming (C-Code):



It's working well :-)

PCB Design and Production

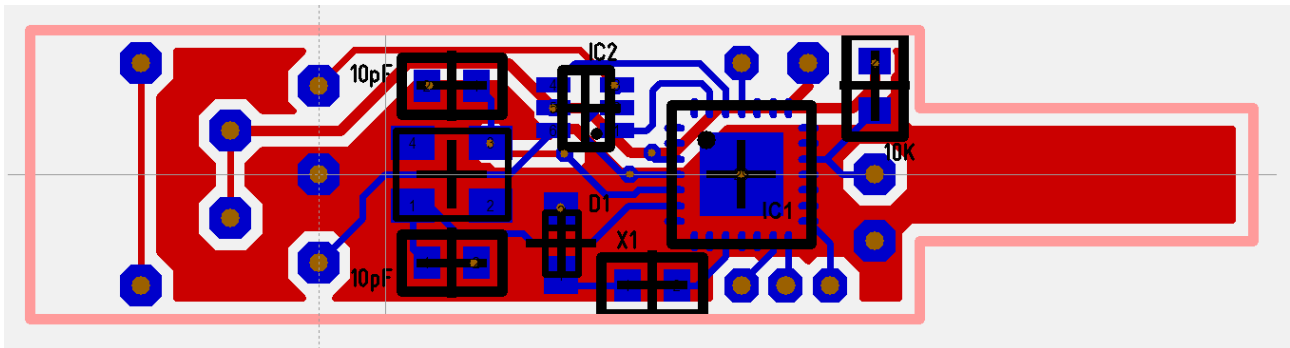
Since the SIO port of the PC-1600 is SHARP proprietary, there is no suitable plug available on the market (including ebay). Therefore I decided to customize a standard guitar plug, which looks nice.

The challenge now was to design and weld a circuit board that is small enough to fit into this plug!

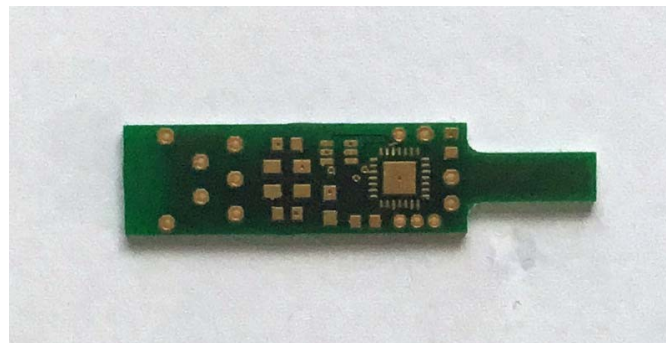
(ok, this a bit crazy but the jouney is the reward :-))



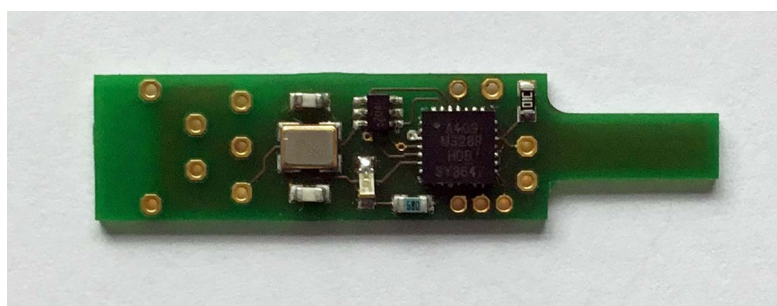
Here is the PCB-layout ...



... and the final PCB:

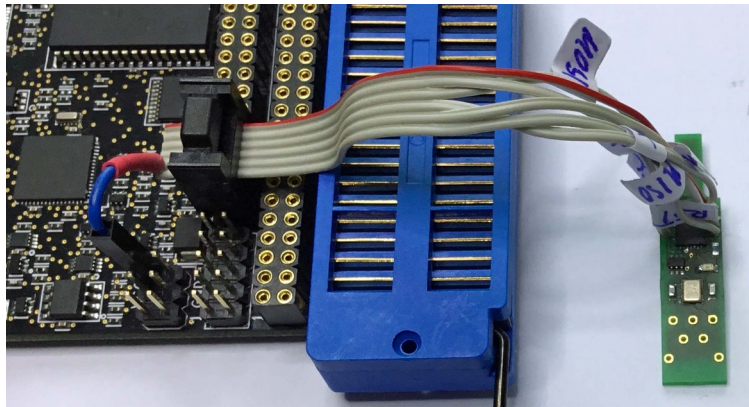


The smallest available version of the Atmega328P is a VFQFN-28 package with 4x4mm size and pins underneath the IC. It can only be welded with a reflow station. The capacitors, resistors and LED are of size 603.



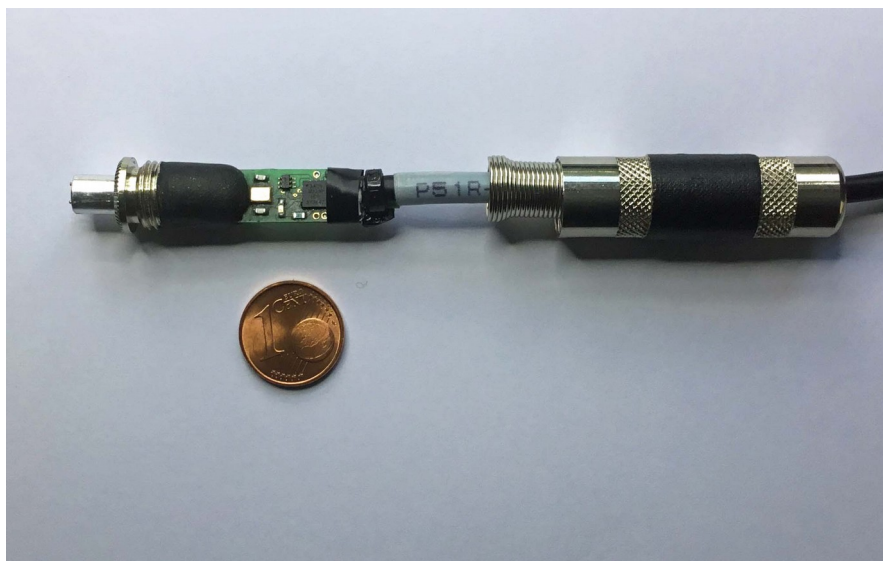
This is by far the tiniest and most compact board I've ever welded. I really can tell, that this was at the border of feasibility!

The board provides pads for connecting the 6pin SPI programming interface:



Custom Plug

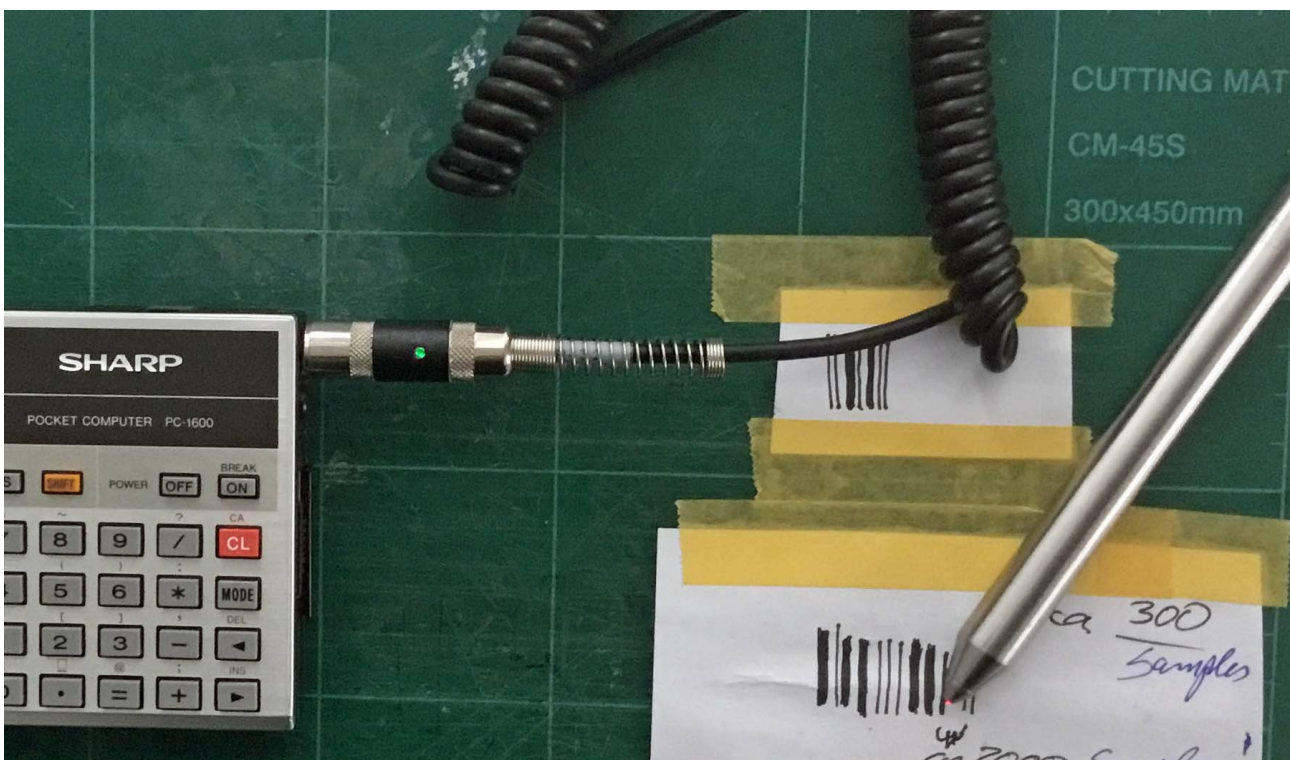
Building the plug turned out to be a similar challenge as the board. I skip the details and just show the result:



Operation and Tests

The circuit board in the plug provides an LED that is used for indication of the wands state. For instance, it flashes 3x when the pocket computer (and thus the microcontroller) is powered on and it lights up when the wand is put on a white surface to indicate 'ready for scan' to the operator.

If you take a closer look at the tip of the wand, you can see the light from the wands red LED that is powered directly by the PC-1600 SIO port. A photo diode that is incorporated in the wands tip too, detects if there is a reflection from ground (white) or not (black). Within the wand this signal goes through an adaptive gain control (AGT), some digital trigger elements and finally to the open collector output pin of the dumb wand.



The Atmega runs at 14,7Mhz and a quick scan operation results in a runlength of about 1200 to 2000+ samples for the last thin line in my hand drawn test chart! So there should definitely be no problem with precision/resolution.

It makes me smile a bit, that the plug is actually faster than the pocket computer ...

Decoding

Now this is an elegant and working semi-smart wand for the SHARP PC-1600. The pocket computer for now just prints the received runlength encoded scans on the display – just for test/validation.

The next step will be to write a first decoder/driver for the EAN-13 barcode standard running on the PC-1600 (C or Assembler).

But that's for sure: The hard part is already done :-)

References

- The original SHARP CE-1601N barcode reader pen for the PC-1600:
<http://pocket.free.fr/> -> SHARP -> CE-1601N
- A good source for inspiration was this academic laboratory course project - but it takes a different design approach (smart wand):
<https://people.ece.cornell.edu/land/courses/ece4760/FinalProjects/s2004/mtf23/BarcodeScanner/index.htm>