

# Working with special hardware from China

*On developing software without a manual*

Willem L. Middelkoop

Dec. 31, 2020

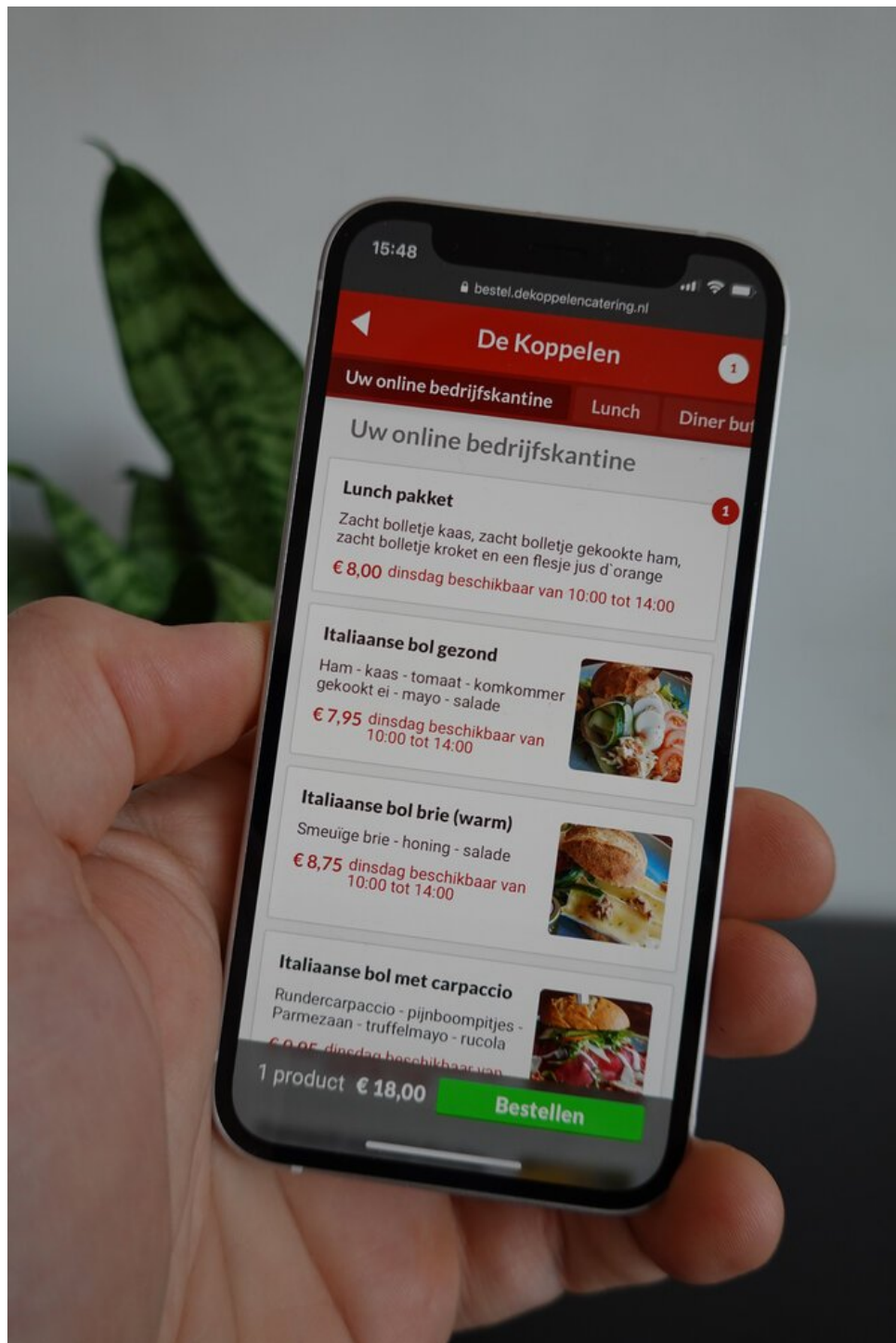


This month I received a shipment from China containing some special hardware, Android devices with embedded printers! To make life easier for the restaurant and business owners that use my food ordering app, I want these devices to automatically print new orders. If only I can get these machines to work with my software... challenge accepted!

## Food ordering app

Through [an highly optimised webshop](#) people can order food for takeaway and delivery. My customers are restaurants and store owners that sell their products through the app.

Orders are forwarded to kitchen and delivery staff, making sure everything is prepared and delivered.



*People use the app to order food - orders are forwarded to kitchen and delivery staff*

### Need for a printer

Over the past months I have been [adding features](#) to help my customers manage their online stores better. A commonly heard request is to get rid of the computer or tablet in the kitchen that they now have to use to access online orders. Instead of a big and bulky

computer I searched for a compact alternative.

## Android with embedded printer

After some digging through various international market places, I was able to contact a Chinese hardware vendor that has created an Android device with an integrated printer! Few weeks later it arrived from China - what a wonderful world we live in!

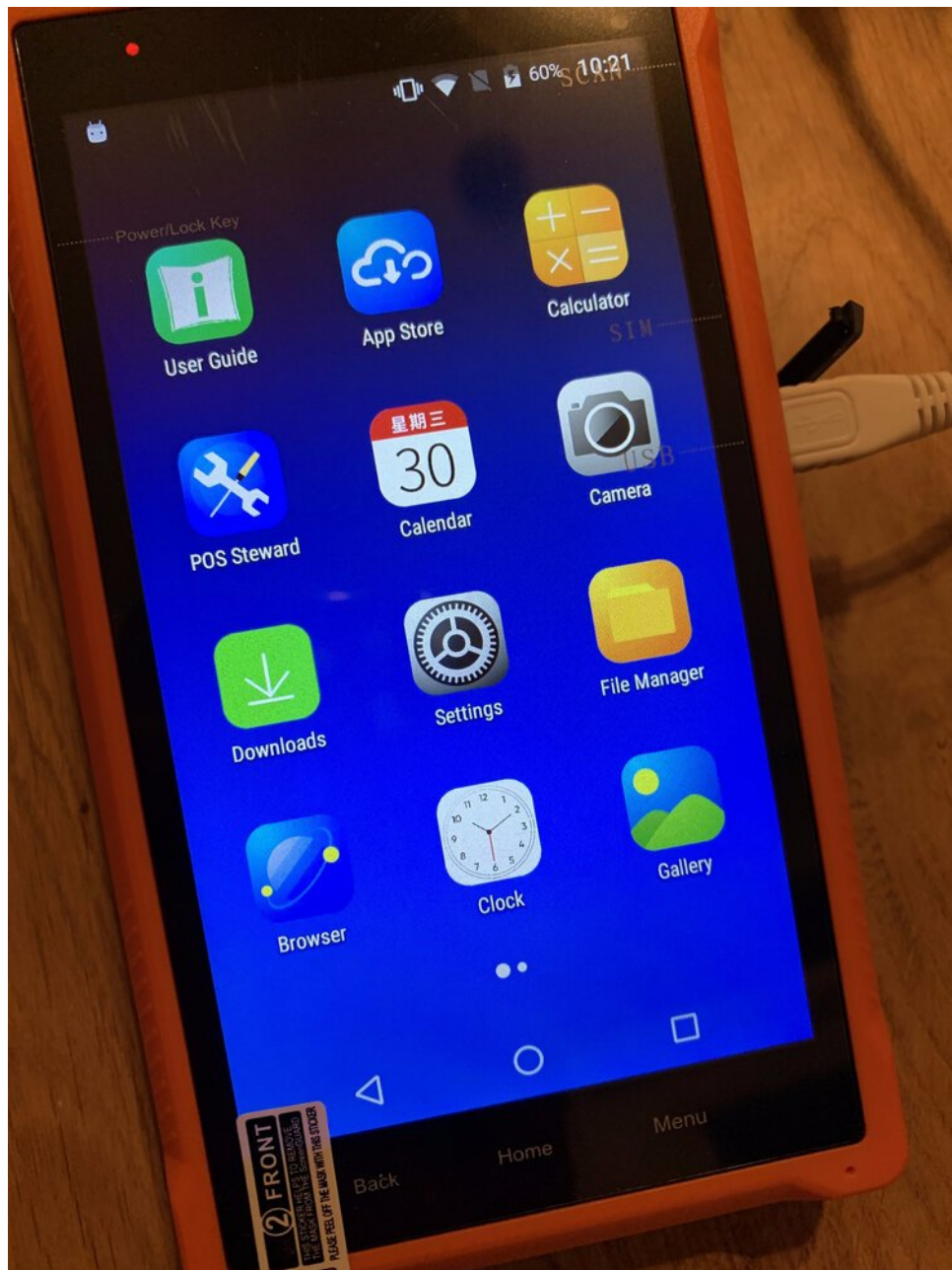


*Handheld Android Terminal - OEM hardware straight from China*

This is an interesting piece of hardware as it is:

- compact (compared to a conventional PC, tablet or laptop)
- it can run on batteries *and* on direct power
- works on WiFi *and* has the option for 4G mobile internet
- has a large touch screen
- it can run native Android apps
- it can access the web
- it has an integrated printer!





*It runs Android - with some Chinese sauce on it (figuratively speaking...)*

The integrated printer is super practical as it enables the device to become a total solution for handling online orders. Using the touch screen orders can be accessed and the device can be configured to automatically print them. This allows for easy forwarding to the kitchen and delivery staff.



*Its party trick is the embedded printer - super useful in the context of handling (online) orders*

## Running something / software

In order to make all of this happen, I need to be able to run custom software on it. Although it runs Android, the devices came without any manuals, guidance or software development kit. This means I had to reverse engineer my way into this device.

The best thing you can then do is to try something you know well, see if you can get another of your apps running on the device. I connected the device to [my Linux tablet](#) and started tinkering! I happen to have created a [little Android app](#) before... that seemed like the perfect candidate to test my development uplink to the device.



*Snake '97 on the Lemmid Printer - ha!*

The advantage of running something well known on unknown hardware is that you are able to establish some kind of a reference to what the hardware capabilities are. It turns out that the Android printer is quite capable!

## **No documentation or SDK**

To fully leverage the hardware's capabilities I needed to figure out how to program the integrated printer. The Chinese hardware vendor integrated the printer in a non-standard Android way (for as much as there is *a standard way* on Android..). The devices I received came without any software or source code, a real bummer!

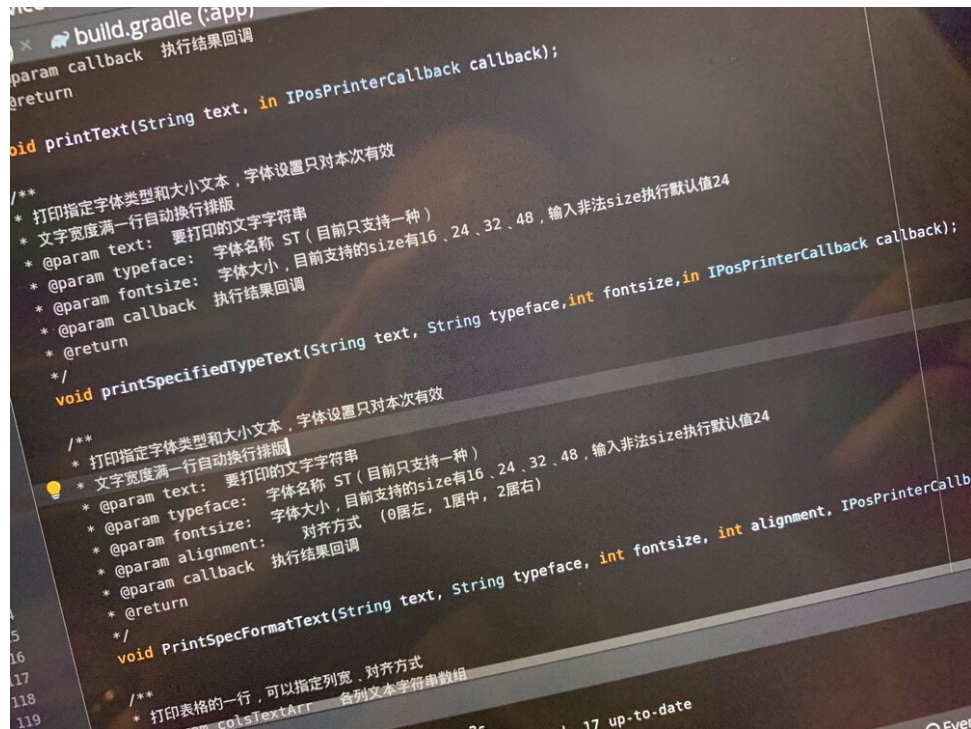
I investigated the serial and model numbers on the OEM device. I looked for any clue I could find and I searched the internet for similar devices. Perhaps that I could find



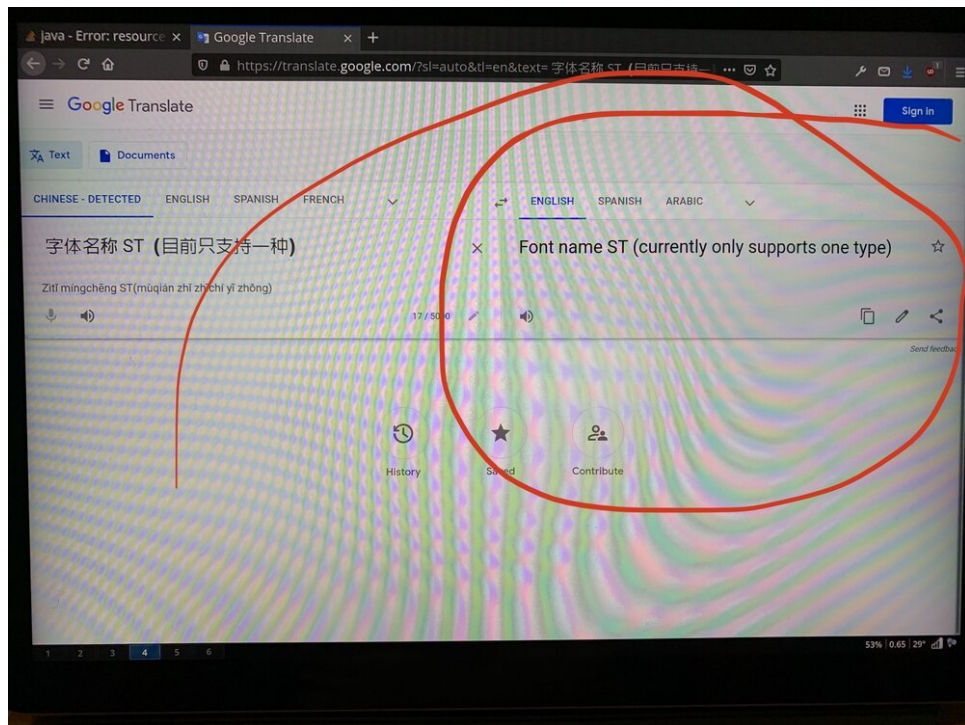
some piece of (related) source code that would give me a starting point to work on this device.

## Reverse engineering Java code

After hours of research, I found a package on a remote part of the internet that contained some source code for a similar device! The problem was... it was all in Chinese!



*Chinese code to make the printer work... how can one possibly understand this?*



*Google Translate to the rescue!*

I analysed the source code and quickly figured out how the basics worked. Using Google Translate I discovered that some parts of the source code were limited or not fully supported. But at least it offered me a starting point! Eventually I was not only able to read the Chinese source code, I was able to write software for it, too!

```

fiedTypeText("欢迎使用智能POS机数据终端\n", "ST", 32, callback);
Lines(1, 8, callback);
fiedTypeText("智能POS 数据终端 智能POS\n", "ST", 32, callback);
Lines(1, 8, callback);
fiedTypeText("#POS POS ipos POS POS POS ipos POS POS ipos\n", "ST", 32, callback);
Lines(1, 8, callback);
kLines(1, 16, callback);
map(1, 6, mBitmap, callback);
kLines(1, 16, callback);
fiedTypeText("开启打印测试\n", "ST", 32, callback);
fiedTypeText("*****", "ST", 24, callback);
fiedTypeText("这是一行16号字体\n", "ST", 16, callback);
fiedTypeText("这是一行24号字体\n", "ST", 24, callback);
fiedTypeText("这是一行32号字体\n", "ST", 32, callback);
fiedTypeText("这是一行48号字体\n", "ST", 48, callback);
fiedTypeText("ABCDEFGHIJKLMNOPQRSTUVWXYZ01234\n", "ST", 16, callback);
fiedTypeText("abcdefghijklmnopqrstuvwxyz56789\n", "ST", 24, callback);
fiedTypeText("αβγδεζηθικλμνξοπρστυφ\n", "ST", 24, callback);
fiedTypeText("κρχκμνκλρκνκμρτυφ\n", "ST", 24, callback);
fiedTypeText("http://www.baidu.com\n", 4, 3, callback);
fiedTypeText("16", 16, callback);
fiedTypeText("11", 11, callback);

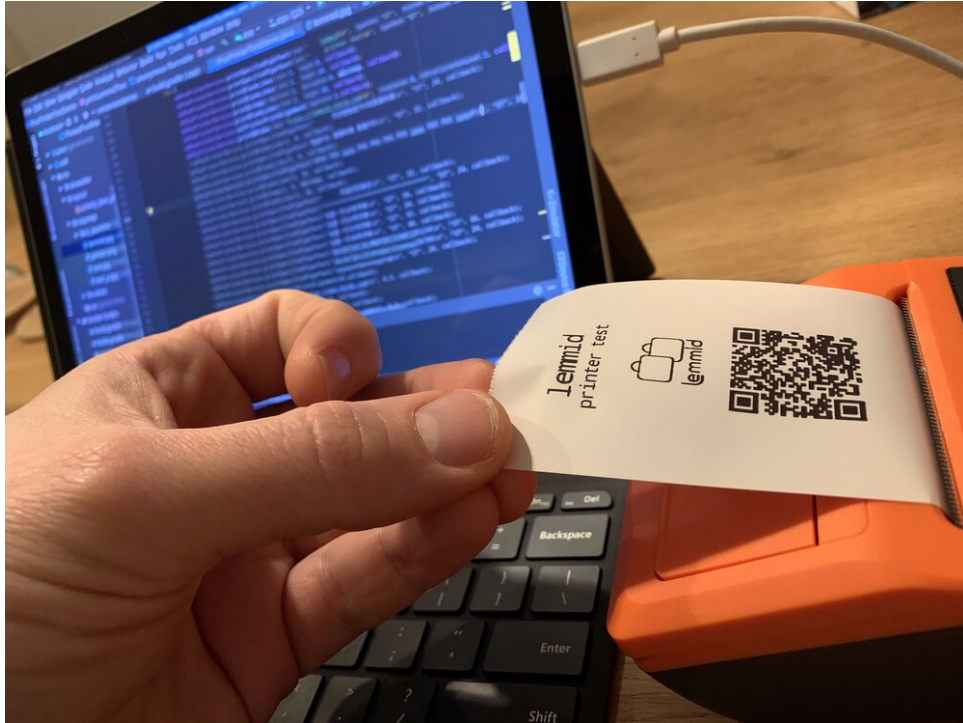
```

*Java code to drive the printer*

The next step was to create a test program that could test if the adapted Chinese



source code would actually work. If I would succeed this would enable me to start writing real software for the Android printer, software that will eventually be used by my customers. Well... low and behold, it worked! The printer started making some noise and voila: a Lemmid Logo appeared!



*YES! The printer prints the "Lemmid" logo on a ticket!*

## Conclusion

This is brilliant stuff! I am really happy that I found a way to write software for this special hardware. It enables me to integrate the hardware into the platform that I am building for my customers.

If there is one thing to takeaway from this post is that you sometimes simply need to try and see if you can figure out a way to make things work. Because only if you don't try, you can be really certain it won't work!