

Arbeiten mit spezieller Hardware aus China

Über die Entwicklung von Software ohne Handbuch

Willem L. Middelkoop

Dec. 31, 2020

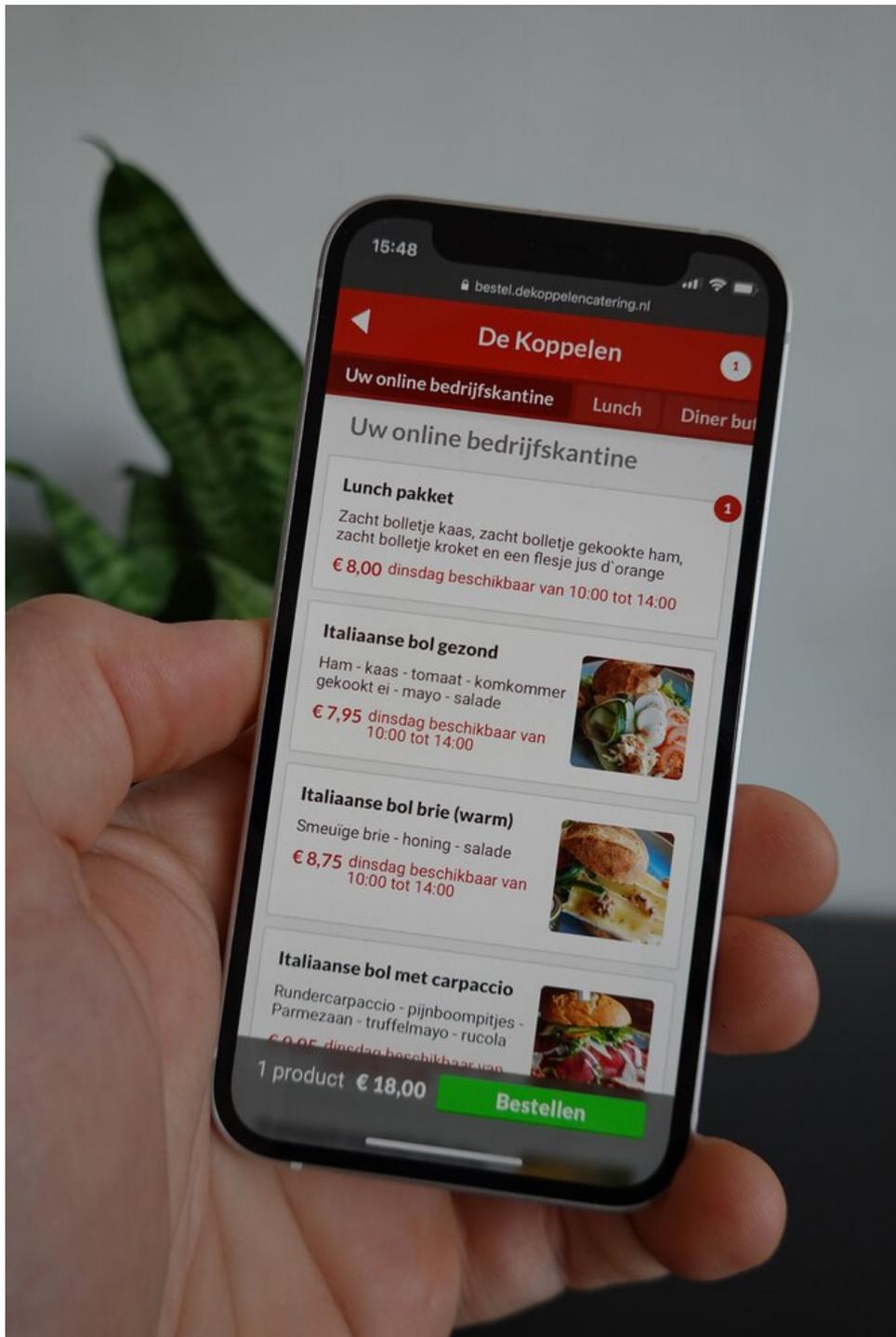


Diesen Monat habe ich eine Lieferung aus China mit spezieller Hardware erhalten, Android-Geräte mit eingebauten Druckern! Um den Restaurant- und Geschäftsinhabern, die meine Bestell-App verwenden, das Leben zu erleichtern, möchte ich, dass diese Geräte neue Bestellungen automatisch drucken. Wenn ich diese Maschinen nur mit meiner Software zum Laufen bringen könnte... Herausforderung angenommen!

Food-Bestell-App

Über einen [hochoptimierten Webshop](#) können Leute Essen zum Mitnehmen und Liefern bestellen. Meine Kunden sind Restaurants und Ladenbesitzer, die ihre Produkte über

die App verkaufen. Bestellungen werden an Küchen- und Lieferpersonal weitergeleitet, um sicherzustellen, dass alles zubereitet und geliefert wird.



Leute bestellen Essen über die App - Bestellungen werden an Küche und Lieferpersonal weitergeleitet

Bedarf an einem Drucker

In den letzten Monaten habe ich [Funktionen](#) hinzugefügt, um meinen Kunden zu helfen, ihre Online-Shops besser zu verwalten. Eine häufig geäußerte Bitte ist, den Computer

oder das Tablet in der Küche loszuwerden, den sie jetzt benutzen müssen, um auf Online-Bestellungen zuzugreifen. Anstelle eines großen und sperrigen Computers suchte ich nach einer kompakten Alternative.

Android mit integriertem Drucker

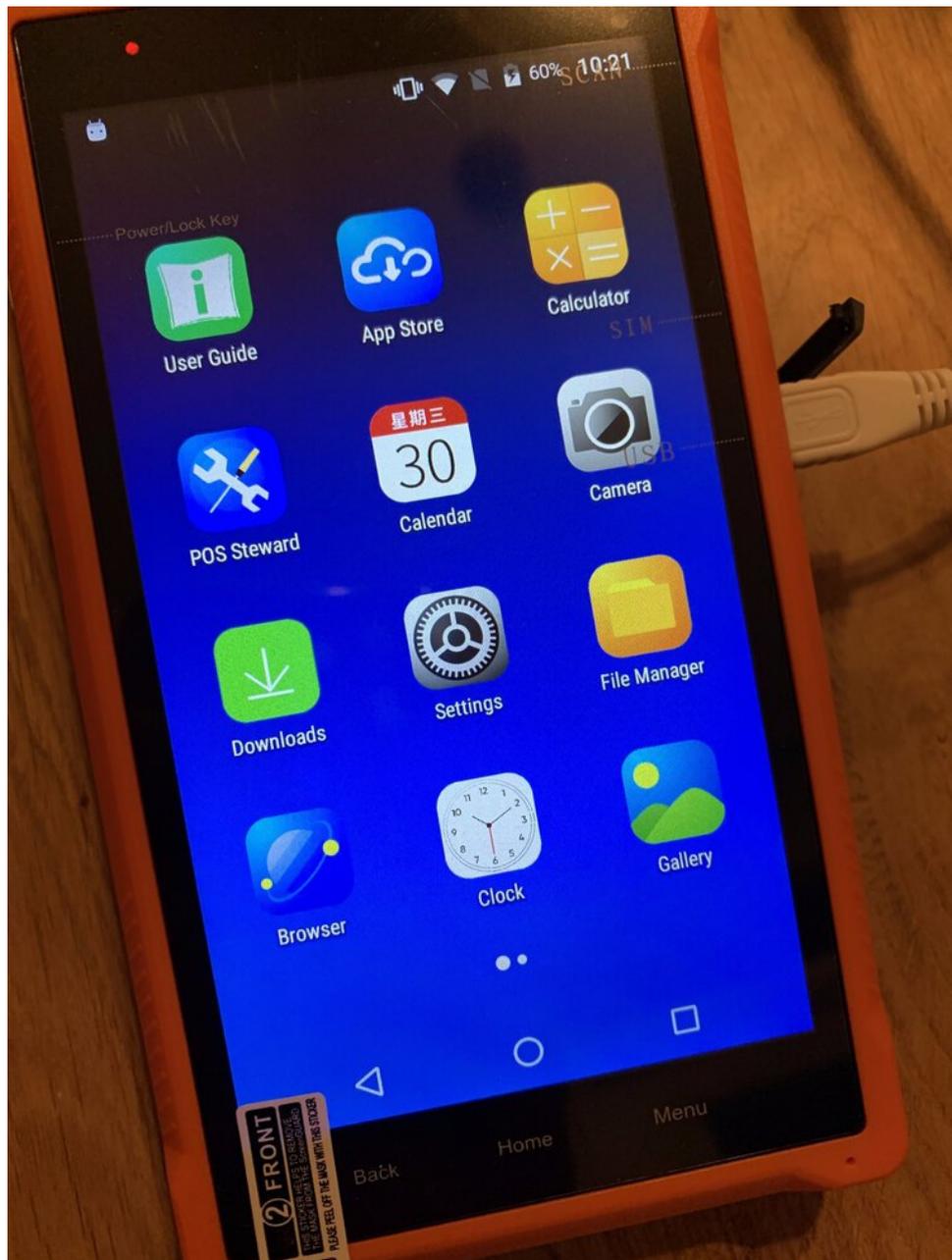
Nach einigem Stöbern auf verschiedenen internationalen Marktplätzen konnte ich einen chinesischen Hardware-Anbieter kontaktieren, der ein Android-Gerät mit integriertem Drucker entwickelt hat! Ein paar Wochen später kam es aus China an - in was für einer wundervollen Welt wir leben!



Handheld Android Terminal - OEM-Hardware direkt aus China

Dies ist ein interessantes Stück Hardware, da es:

- kompakt ist (im Vergleich zu einem herkömmlichen PC, Tablet oder Laptop)
- mit Batterien *und* mit Netzstrom betrieben werden kann
- über WLAN *und* optional über 4G-Mobilfunkinternet funktioniert
- einen großen Touchscreen hat
- native Android-Apps ausführen kann
- auf das Internet zugreifen kann
- einen integrierten Drucker hat!



Es läuft Android - mit etwas chinesischer Sauce drauf (im übertragenen Sinne...)

Der integrierte Drucker ist super praktisch, da er das Gerät zu einer Gesamtlösung für die Bearbeitung von Online-Bestellungen macht. Über den Touchscreen kann auf Bestellungen zugegriffen werden und das Gerät kann so konfiguriert werden, dass sie automatisch gedruckt werden. Dies ermöglicht eine einfache Weiterleitung an das Küchen- und Lieferpersonal.



Der Clou ist der eingebaute Drucker - super nützlich im Kontext der Bearbeitung von (Online-)Bestellungen

Ausführen von etwas / Software

Um all dies zu ermöglichen, muss ich in der Lage sein, benutzerdefinierte Software darauf auszuführen. Obwohl es Android ausführt, wurden die Geräte ohne Handbücher, Anleitungen oder Software Development Kit geliefert. Das bedeutet, ich musste mich per Reverse Engineering in dieses Gerät einarbeiten.

Das Beste, was man dann tun kann, ist, etwas auszuprobieren, das man gut kennt, und zu sehen, ob man eine andere seiner Apps auf dem Gerät zum Laufen bringen kann. Ich habe das Gerät mit meinem [Linux-Tablet](#) verbunden und angefangen zu tüfteln! Ich habe zufällig schon einmal eine kleine [Android-App](#) erstellt... die schien der perfekte Kandidat zu sein, um meine Entwicklungsverbindung zum Gerät zu testen.



Snake '97 auf dem Lemmid Drucker - ha!

Der Vorteil, etwas Bekanntes auf unbekannter Hardware auszuführen, besteht darin, dass man eine Art Referenz für die Hardwarefähigkeiten erstellen kann. Es stellt sich heraus, dass der Android-Drucker ziemlich leistungsfähig ist!

Keine Dokumentation oder SDK

Um die Fähigkeiten der Hardware voll auszuschöpfen, musste ich herausfinden, wie man den integrierten Drucker programmiert. Der chinesische Hardware-Anbieter hat den Drucker auf eine nicht standardmäßige Android-Weise integriert (soweit es *eine Standardweise* auf Android gibt..). Die Geräte, die ich erhalten habe, kamen ohne Software oder Quellcode, eine echte Enttäuschung!

Ich untersuchte die Serien- und Modellnummern auf dem OEM-Gerät. Ich suchte

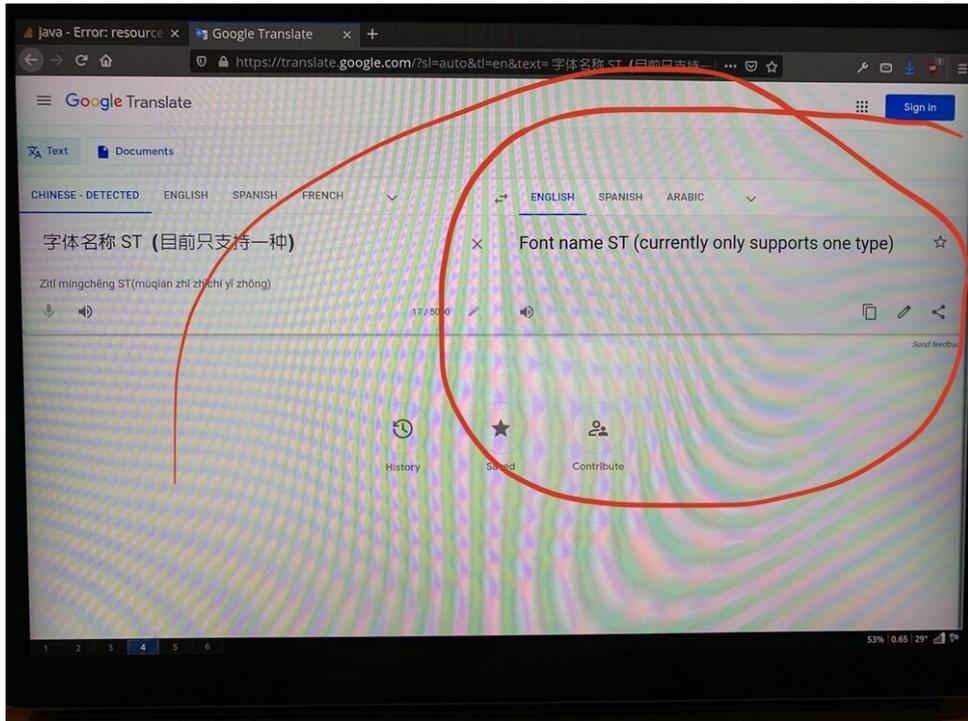
nach jedem Hinweis, den ich finden konnte, und durchsuchte das Internet nach ähnlichen Geräten. Vielleicht könnte ich ein Stück (verwandten) Quellcode finden, der mir einen Ausgangspunkt für die Arbeit an diesem Gerät geben würde.

Reverse Engineering von Java-Code

Nach stundenlanger Recherche fand ich ein Paket auf einem abgelegenen Teil des Internets, das Quellcode für ein ähnliches Gerät enthielt! Das Problem war... es war alles auf Chinesisch!



Chinesischer Code, um den Drucker zum Laufen zu bringen... wie kann man das nur verstehen?



Google Translate zur Rettung!

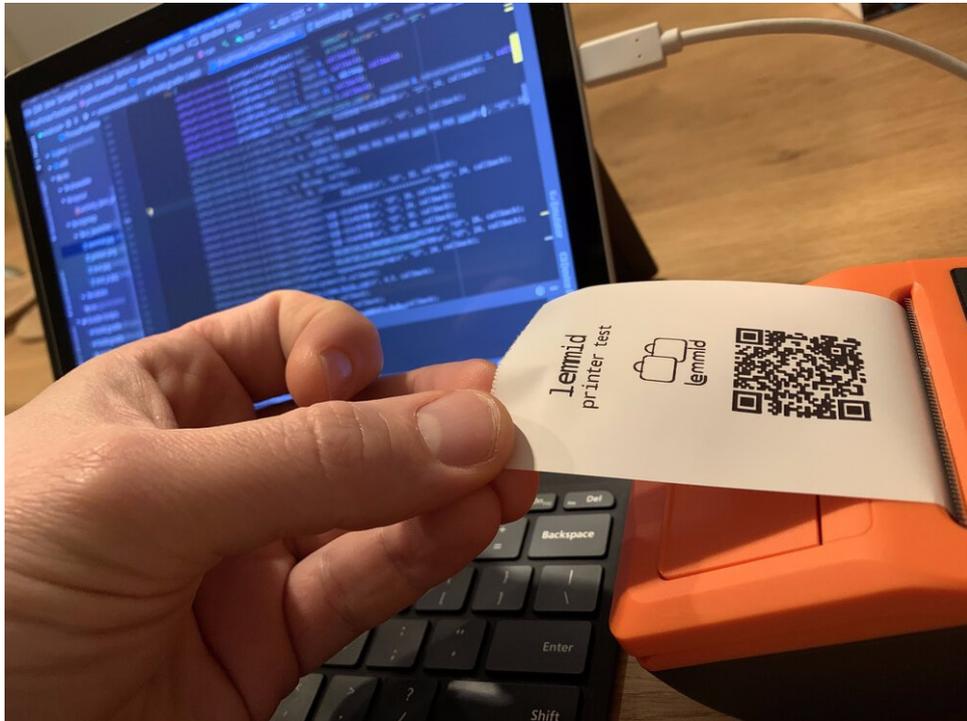
Ich analysierte den Quellcode und fand schnell heraus, wie die Grundlagen funktionierten. Mit Google Translate entdeckte ich, dass einige Teile des Quellcodes eingeschränkt oder nicht vollständig unterstützt wurden. Aber zumindest bot es mir einen Ausgangspunkt! Schließlich war ich nicht nur in der Lage, den chinesischen Quellcode zu lesen, sondern auch Software dafür zu schreiben!

```

fiedTypeText("欢迎使智能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);
kLines(1, 8, callback);
fiedTypeText("#POS POS ipos POS POS POS ipos POS POS ipos\n", "ST", 32, callback);
kLines(1, 16, callback);
hap(1, 6, mBitmap, callback);
hkLines(1, 16, callback);
ecifiedTypeText("开启打印测试\n", "ST", 32, callback);
ecifiedTypeText("*****\n", "ST", 24, callback);
ecifiedTypeText("这是一行16号字体\n", "ST", 16, callback);
ecifiedTypeText("这是一行24号字体\n", "ST", 24, callback);
ecifiedTypeText("这是一行32号字体\n", "ST", 32, callback);
ecifiedTypeText("这是一行48号字体\n", "ST", 48, callback);
ecifiedTypeText("ABCDEFGHIJKLMNOPQRSTUVWXYZ01234\n", "ST", 16, callback);
ecifiedTypeText("abcdefghijklmnopqrstuvwxyz56789\n", "ST", 24, callback);
ecifiedTypeText("κρχκμνκλρκνκμτφ\n", "ST", 24, callback);
nterPrintAlignment(0, callback);
RCode("http://www.baidu.com\n", 4, 3, callback);
  
```

Java-Code zur Ansteuerung des Druckers

Der nächste Schritt war die Erstellung eines Testprogramms, mit dem getestet werden konnte, ob der angepasste chinesische Quellcode tatsächlich funktionieren würde. Wenn mir das gelänge, könnte ich anfangen, echte Software für den Android-Drucker zu schreiben, Software, die schließlich von meinen Kunden verwendet werden würde. Nun... siehe da, es funktionierte! Der Drucker fing an, Geräusche zu machen, und voila: ein Lemmid-Logo erschien!



JA! Der Drucker druckt das "Lemmid"-Logo auf ein Ticket!

Fazit

Das ist brillantes Zeug! Ich bin wirklich froh, dass ich einen Weg gefunden habe, Software für diese spezielle Hardware zu schreiben. Es ermöglicht mir, die Hardware in die Plattform zu integrieren, die ich für meine Kunden baue.

Wenn es eine Sache gibt, die man aus diesem Beitrag mitnehmen kann, dann ist es, dass man manchmal einfach versuchen muss, einen Weg zu finden, die Dinge zum Laufen zu bringen. Denn nur wenn man es nicht versucht, kann man wirklich sicher sein, dass es nicht funktionieren wird!