

# KindleBerry Pi

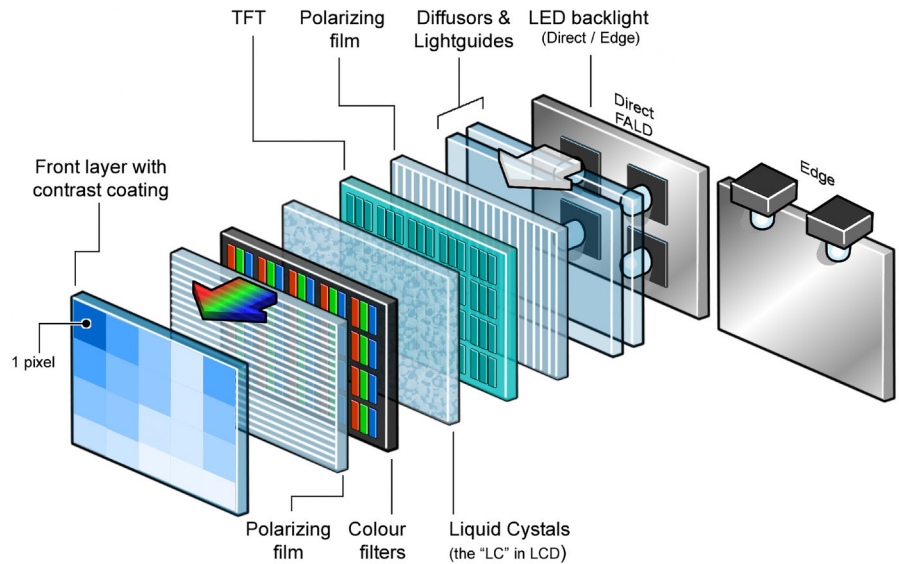
Lubicie czasem posiedzieć z laptopem w parku? Albo popracować zdalnie w barze przy plaży lub latem w kawiarni, w malowniczej miejscowości skąpanej w słońcu? Brzmi fajnie, nie? Niestety, ci co spróbowali, szybko odkryli, że słońce i laptop to nie jest dobre połączenie – klasyczne ekrany LCD nie nadają się do pracy przy dużym nasłonecznieniu. Gdyby tylko mieć laptop z ekranem EPD... W artykule pokazujemy, jak zbudować takie urządzenie w bardzo prosty sposób.

W większości laptopów wbudowany ekran LCD wymaga podświetlenia na tyle jasnego, by barwy prezentowane na ekranie mogły być jaśniejsze niż otoczenie. Wynika to z budowy takiego ekranu, tak jak zostało to pokazane na **rysunku 1**. Niestety, w miejscach mocno nasłonecznionych jasność otoczenia jest tak duża, że ekran nawet nowoczesnego laptopa przestaje być czytelny.

Zupełnie odwrotne cechy mają wyświetlacze EPD, czyli bazujące na elektronicznym papierze (**rysunek 2**). W ich przypadku, by prezentowana treść była czytelna, musi być oświetlona z zewnątrz – np. właśnie dzięki światłu słonecznemu. Niestety typowy czytnik taki jak Kindle nie jest komputerem i nie da się na nim normalnie pracować. Można jednak przygotować rozwiązanie łączące, w którym serce komputera stanowi Raspberry Pi, a wyświetlacz pochodzi z czytnika takiego jak Kindle.

## KindleBerry Pi – projekt bazowy

W artykule bazujemy na projekcie KindleBerry Pi (**fotografia 3**), opracowanym przez Studio Ponnuki na bazie czytnika Kindle 3. generacji oraz minikomputera Raspberry Pi generacji pierwszej. Zastosowanie nowszego czytnika nie zrobi większej różnicy, a może podnieść koszt wykonania projektu. Można jednak pokusić się o użycie czytnika z ekranem o większej przekątnej albo z szybszym odświeżaniem, dzięki czemu praca na komputerze będzie wygodniejsza. Natomiast zdecydowanie warto sięgnąć po nowszą wersję Raspberry Pi, która będzie nie tylko silniejsza, ale też będzie zawierała wbudowane więcej interfejsów, które przydadzą się w trakcie użytkowania komputera. Najlepiej wykorzystać model Raspberry Pi 3 B+,



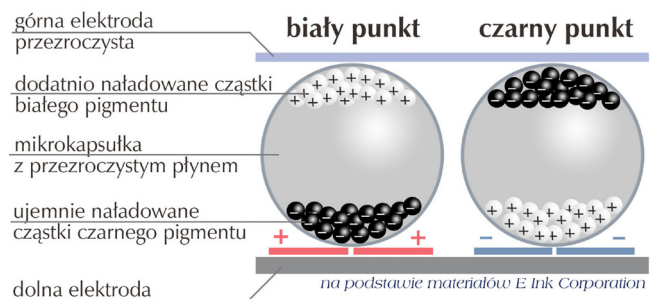
Rysunek 1. Budowa typowego ekranu TFT-LCD

w którym od ręki otrzymamy aż 4 porty USB, 1 GB pamięci RAM oraz obsługę Wi-Fi (**fotografia 4**).

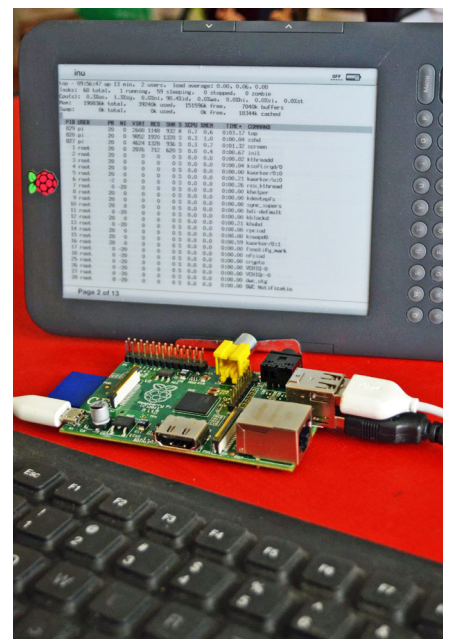
Ogólny projekt takiego przenośnego, miniaturowego komputera wygląda następująco (**rysunek 5**): Kindle pracuje jedynie jako wyświetlacz, podłączony do Raspberry Pi kablem USB. RPI podłączone jest do zasilania – czy to do zwykłego zasilacza 5 V (ładowarki do telefonu), czy do akumulatora. Zamiast ogniw i sterownika zasilania można po prostu użyć powerbanku. Do Raspberry Pi podłączamy także klawiaturę i myszkę, jeśli jest potrzebna. Można je podłączyć przewodami, do złączy USB, bezprzewodowo, poprzez jeden lub dwa odbiorniki radiowe podłączane do USB, albo – w przypadku modelu RPI 3B+ – przez Bluetooth, bez dodatkowych konektorów. Łącznie tymi wszystkimi akcesoriami zajmiemy maksymalnie trzy porty USB typu A, co oznacza, że w modelu Raspberry Pi 3B+ zostanie nam przynajmniej jeden wolny, a w przypadku modelu z dwoma portami możemy być zmuszeni do użycia dodatkowego huba USB.

Czy w ten sposób tworzymy faktycznie przenośny komputer? Pomyślmy o tym w następujący sposób, zakładając, że korzystamy z Raspberry Pi 3 model B+. Wyobraźmy sobie, że jedziemy na dłuższy wyjazd zagraniczny, np. do „ciepłych krajów”, w trakcie

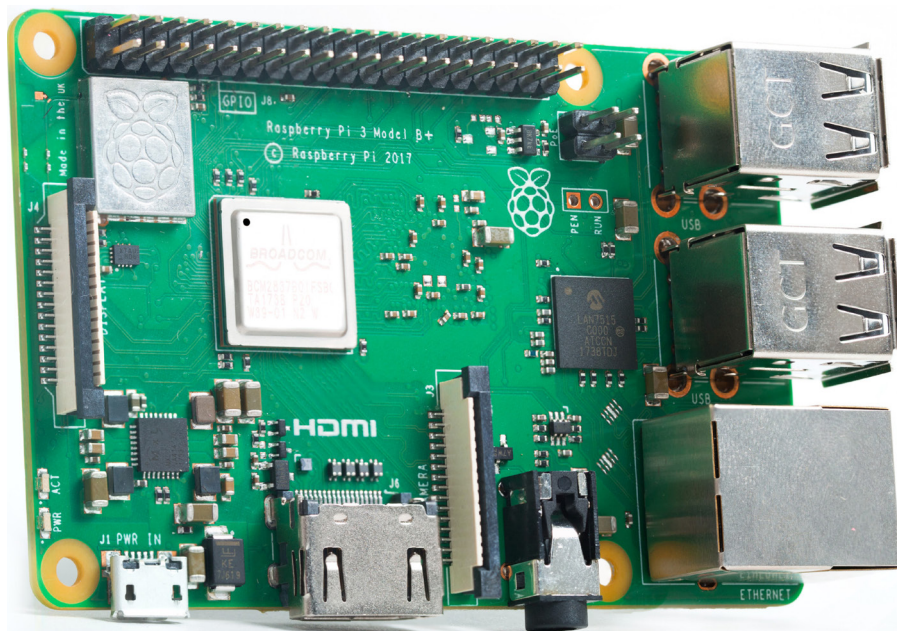
## Schemat działania e-papieru – przekrój



Rysunek 2. Zasada działania elektronicznego papieru na przykładzie technologii e-ink



Fotografia 3. Podstawowa wersja KindleBerry



Fotografia 4. Raspberry Pi 3 Model B+

którego będziemy musieli co jakiś czas pracować, ale będziemy się starać jak najwięcej czasu spędzić na świeżym powietrzu. Jest to coraz bardziej popularny sposób spędzania zimy przez wielu polskich specjalistów. W taką podróż oczywiście zabieramy smartfon, a do niego mocną ładowarkę i powerbank. Bierzymy też czytnik książek elektronicznych, bo w czasie wolnym chcemy się zrelaksować przy ciekawej lekturze. Do czytnika i do telefonu potrzebne są przewody zasilające, a być może i druga ładowarka na wszelki wypadek. Oznacza to, że większość podzespołów KindleBerry Pi i tak bierzemy ze sobą! Potrzebne jest jeszcze tylko Raspberry Pi i klawiatura oraz ewentualnie myszka. Ta ostatnia może być miniaturowa, a i klawiatura może być niedużych rozmiarów – na pewno zajmie mniej miejsca niż laptop. Ba – komplet ten jest znacznie tańszy niż pełnoprawny, nawet niewielki laptop, więc w razie nieszczęśliwego zdarzenia utrata takiego zestawu będzie nas mniej bolała niż utrata całego notebooka. A przy okazji zyskujemy możliwość pracy w jasnym świetle.

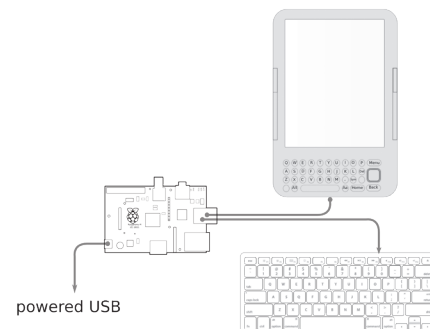
## Uwolnienie czytnika

Największym problemem, związanym z całym tym projektem, jest zdjęcie zabezpieczeń z czytnika. Urządzenia takie jak Kindle są zabezpieczone przed różnymi operacjami, by można było z nich korzystać tylko tak, jak producent to przewidział. Operacja usuwania tych zabezpieczeń nazywa się „jailbreaking” i sprawia, że użytkownik ma dostęp do praktycznie całego systemu operacyjnego czytnika, a więc może go przekonfigurować i doinstalować na nim dodatkowe rzeczy. Niestety, w praktyce oznacza to też utratę gwarancji na czytnik, a w przypadku popełnienia błędu po drodze urządzenie może ulec uszkodzeniu. Co więcej, w niektórych

krajach taka czynność może być nielegalna. Niemniej, im urządzenie jest bardziej popularne, tym większa szansa, że ktoś przygotował już narzędzia umożliwiające łatwe zdjęcie zabezpieczeń z takiego sprzętu. W przypadku Kindle, a więc najbardziej popularnego elektronicznego czytnika książek na świecie, nowe programy do zdejmowania zabezpieczeń pojawiają się dosyć szybko po premierze nowych generacji urządzeń. I choć sprzęt ten jest droższy niż rozwiązania konkurencyjne, to bardzo łatwo go dostać, nawet przeglądając pierwszy lepszy serwis z ogłoszeniami kupna i sprzedaży przedmiotów przez Internet.

Obecnie, szukając programu do zdjęcia zabezpieczeń, warto sięgnąć po jedno z dwóch narzędzi, zależnie od posiadanej wersji urządzenia i firmware'u. Dla urządzeń z firmwarem w wersji 2.x, 3.x i 4.x, a więc na czytnikach Kindle bez obsługi interfejsu dotykowego, takich jak Kindle 2, DX, DXG, 3 i 4, postępujemy zgodnie z zaleceniami podanymi pod adresem: <http://bit.ly/2YQy8RV>. W przypadku czytników z firmware'em w wersji od 5.0.x do 5.4.4.2, a więc urządzeń takich jak Kindle Touch, PaperWhite, PaperWhite2, Touch 2, Voyage, PaperWhite3, Oasis i Touch 3, aktualną wersję narzędzi znaleźć można pod adresem: <http://bit.ly/2VJvgUZ>. Można się spodziewać, że z czasem pod tym drugim adresem będą pojawiały się kolejne narzędzia do łamania zabezpieczeń nowszych generacji

```
Listing 1. Ustawienia umożliwiające komunikację IP za pomocą interfejsu USB
allow-hotplug usb0
mapping hotplug
script grep
map usb0
iface usb0 inet static
address 192.168.2.1
netmask 255.255.255.0
broadcast 192.168.2.255
up iptables -I INPUT 1 -s 192.168.2.1 -j ACCEPT
```



Rysunek 5. Połączenie przewodowe Kindle

czytników Kindle, aż do momentu, gdy firma Amazon wprowadzi w swoich produktach jakieś rewolucyjne zmiany. W niniejszym artykule nie opisujemy kolejnych kroków związanych z samym zdjęciem zabezpieczeń, gdyż będą się one różnić szczegółami w zależności od wersji posiadanego czytnika. Na koniec warto doinstalować narzędzie ułatwiające wgrywanie nowych aplikacji, takie jak LaunchPad.

## Instalacja oprogramowania na Kindle

Po „uwolnieniu” czytnika możemy zabrać się do instalowania na nim narzędzi, potrzebnych do realizacji projektu. W pierwszej kolejności będzie to emulator terminalu. Dobrym przykładem będzie myts-6, bazujący na programie kiterm, ale nieco usprawniony pod kątem pracy na Kindle. Jego opis można znaleźć pod adresem: <http://bit.ly/2WmvQMM>, skompilowany program, zgodny z LaunchPadem, pobrać z adresu: <http://bit.ly/2K0hzP7>, a kod źródłowy został udostępniony pod adresem: <http://bit.ly/2YOwlwn>.

Bardzo ważne jest także, by zmodyfikować działanie interfejsu USB czytnika, tak by mógł służyć jako interfejs sieciowy. W tym celu korzystamy z narzędzia UsbNetwork. Jest ono dostępne pod adresem: <http://bit.ly/2YQy8RV>, przy czym konkretna wersja, jaką należy pobrać, będzie zależać od wersji posiadanego czytnika i jego firmware'u.

Po upewnieniu się, że w czytniku funkcja UsbNetwork jest włączona, należy podłączyć czytnik przez USB do Raspberry Pi. W tym momencie czytnik będzie na swoim porcie USB widział sieć ethernetową i miał tam skonfigurowany adres 192.168.2.2. Oznacza to, że na Raspberry Pi należy skonfigurować adres 192.168.2.1 jako automatycznie przypisywany do portu USB. By to zrobić, należy

w pliku /eth/network/interfaces wprowadzić ustawienia pokazane na **listingu 1**.

Teraz musimy upewnić się, że Raspberry Pi będzie automatycznie uruchamiać się z zalogowanym użytkownikiem. Bez tego nie dojdzie do momentu, w którym czytnik będzie można kontrolować z użyciem Raspberry Pi.

Aby Raspberry Pi miało automatycznie zalogowanego użytkownika, należy edytować plik /etc/inittab i wycommentować linijkę: #1:234:respawn:/sbin/getty 3840 tty1

A po niej dodać:  
[prettify]1:2345:respawn:/sbin/getty -a YOUR\_USER\_NAME -8 -s 38500 tty1 linux

Powyższe działanie sprawi, że użytkownik będzie automatycznie się logował. Warto też upewnić się, że login do systemu w Kindle jest identyczny.

Druga sprawa to taka by nie otwierać dwóch sesji na Kindle, tylko dołączyć do zaczątej. Dlatego w pliku .bash\_profile można wpisać: if [ -z "\$STY" ]; then exec screen -XR fi

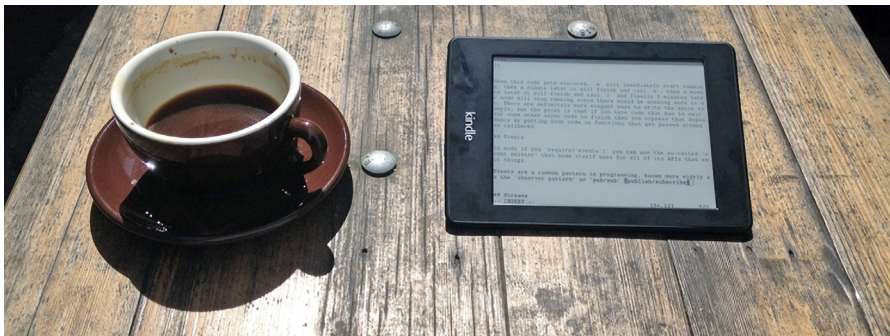
To już właściwie wszystko w podstawowej wersji. Teraz korzystanie z komputera będzie odbywać się poprzez połączenie SSH z poziomu Kindle z Raspberry Pi. Na ekranie Kindle będzie pojawiać się to, co widać w sesji, do której się dołączyło, a więc w tej, w której poruszać się można klawiaturą podłączoną do Raspberry Pi. Efekt takiej konfiguracji można zobaczyć na wideo pod adresem:

<http://bit.ly/2JCbjh4>, a opis całego projektu znajduje się pod adresem: <http://bit.ly/2wjqQtu>.

### Wersja bezprzewodowa z Kindle PaperWhite

Projekt można by pozostawić na aktualnym etapie, ale nie spełnia on jeszcze postawionych założeń i opiera się na starych komponentach. Jego usprawniona wersja została opisana przez Maxa Odgena (pod adresem <http://bit.ly/2K0pXyc>), który w trakcie spotkania programistów w Australii miał okazję przez kilka dni doświadczać problemów w pracy z laptopem na otwartym powietrzu. Postanowił zbudować swoją wersję KindleBerry, z założeniem że będzie ona mogła długo pracować bez dostępu do zasilania i świetnie sprawdzi się np. w trakcie korzystania z komputera w lesie lub na łące (**fotografia 6**). Dodatkowo, zdecydował że pomiędzy klawiaturą, wyświetlaczem i pozostałą częścią komputera ma nie być żadnych kabli.

W swoim rozwiązaniu zastosował Raspberry Pi Model B z obudową wydrukowaną na drukarce 3D. System Raspbian został zainstalowany na 8-gigabajtowej



**Fotografia 6. Kindle bez przewodów w pełnym świetle i z bezprzewodowym podłączeniem do Raspberry Pi**

karcie pamięci SD. Wybrany model Kindle Paperwhite był w tamtym czasie najszybszym z dostępnych, a do tego ma wbudowane podświetlenie, dzięki czemu ekran jest czytelny także gdy w otoczeniu jest bardzo ciemno, czyli np. wieczorami. Użycie tej wersji Kindle sprawiło, że odświeżanie wyświetlacza następuje z opóźnieniem ok. 200 ms, co w zupełności wystarcza do pracy w trybie tekstowym, szczególnie jeśli użytkownik umie pisać na klawiaturze bezwzrokowo, a więc gdy dzięki doświadczeniu nie musi sprawdzać, czy dobrze trafił w klawisz.

Jako klawiaturę, autor wykorzystał model Apple Wireless Keyboard (**fotografia 7**), a więc przenośną wersję na Bluetooth, o układzie identycznym jak ten, z jakiego korzysta na co dzień na swoim laptopie. Ponieważ autor zapragnął by jak najbardziej wyeliminować kable, w projekcie użył nie tylko modemu Bluetooth na USB (model IOGEAR GBU421, **fotografia 8**), ale też karty Wi-Fi na USB (Edimax EW-7811Un, **fotografia 9**). W przypadku użycia nowszej wersji Raspberry Pi, oba te dodatki nie będą potrzebne.

Do zapewnienia komunikacji bezprzewodowej pomiędzy Raspberry Pi i Kindle posłużył przenośny router firmy TP-LINK, model TL-WR702N (**fotografia 10**). To element, który nie jest konieczny, ale zwiększa wygodę, przy czym dosyć mocno podnosi koszt realizacji całego projektu.

Jako źródło zasilania wykorzystany został powerbank USB, a uzupełnieniem jest 4-portowy hub USB, który ułatwia zarządzanie kablami.

Warto dodatkowo omówić kwestię zasilania. Sam Kindle, dzięki swojemu wyświetlaczowi, może pracować dosyć długo bez ładowania. Fakt, że będzie miał włączone Wi-Fi skróci ten czas. Natomiast z różnych analiz wynika, że Raspberry Pi jest w stanie działać ponad 28 godzin na powerbanku o pojemności 16 Ah, w trakcie strumieniowania wideo wysokiej rozdzielczości, co widać na filmiku pod adresem: <http://bit.ly/2M6y-ZML>. W przypadku jeśli praca ma się odbywać jedynie w terminalu, pobór mocy jest znacznie mniejszy i komputer będzie w stanie pracować jeszcze dłużej. Można



**Fotografia 7. Klawiatura Apple Wireless Keyboard**



**Fotografia 8. Miniatury modemu Bluetooth IOGEAR GBU421 na USB**

**Fotografia 9. Miniatury karty sieciowej Edimax EW-7811Un na USB**



**Fotografia 10. Przenośny router Wi-Fi TP-Link TL-WR702N**



**Fotografia 11. Rozstawienie KindleBerry w wersji bezprzewodowej jest szybkie, łatwe i wygodne**



Fotografia 12. Połączenia pomiędzy elementami KindleBerry w wersji bezprzewodowej

też sięgnąć po większe powerbanki – z łatwością da się znaleźć modele o pojemności 26 Ah, a największe modele konsumenckie dochodzą do 50 Ah. W tym kontekście ograniczeniem staje się raczej czas pracy routera Wi-Fi, który na szczęście też można zasilac z powerbanku.

Taka konfiguracja sprawia, że komputer jest lekki, łatwy do przenoszenia i nie

plącze się – łatwo go rozstawić nawet w miejscu o małej przestrzeni (fotografia 11). Może stanowić dodatkowy komputer, noszony oprócz laptopa. Router Wi-Fi i Raspberry Pi mogą być na stałe podłączone do huba USB. Sam Hub można podłączać do powerbanku jednym kablem, gdy tylko chce się użyć komputera. W efekcie rozstawienie całości wymaga jedynie wyjęcia z torby

#### Listing 2. Lista skrótów Kindle Paperwhite

```

";dm" : "/usr/bin/dm.sh",
";dmcc" : "/usr/local/bin/dmcc.sh",
";dh" : "/usr/bin/dh.sh",
";dt" : "/usr/bin/dt.sh",
";Wi-Fipopup" : "/usr/local/bin/Wi-Fipopup.sh",
";sandbox" : "/usr/local/bin/sandbox.sh",
";sbox" : "/usr/local/bin/sandbox.sh",
";shpm" : "/usr/sbin/shipping_mode",
";lzzl" : "/usr/sbin/shipping_mode",
";urst" : "/usr/sbin/userstore_reset",
";usbnetwork" : "/usr/local/bin/usbnetwork.sh",
";un" : "/usr/local/bin/usbnetwork.sh",
";di" : "/usr/local/bin/indexerdisable.sh",
";stopIndexing" : "/usr/local/bin/indexerstop.sh",
";startIndexing" : "/usr/local/bin/indexerenable.sh",
";disableIndexing" : "/usr/local/bin/indexerdisable.sh",
";indexStatus" : "/usr/local/bin/indexstatus.sh",
";ddc" : "/usr/local/bin/dynconfig.sh",
";resetConfig" : "/usr/local/bin/resetConfig.sh",
";rc" : "/usr/local/bin/resetConfig.sh",
";twoFingerChromeOn" : "/usr/local/bin/altChrome.sh 1",
";homeKeyChromeOn" : "/usr/local/bin/altChrome.sh 2",
";normalChrome" : "/usr/local/bin/altChrome 0",
";debugOn" : "/usr/bin/debugOn.sh",
";debugPaint" : "/usr/bin/debugPaint.sh",
";debugOff" : "/usr/bin/debugOff.sh",
";debugPerf" : "/usr/bin/debugPerf.sh",
";dP" : "/usr/bin/debugPerf.sh",
";311" : "/usr/bin/311.sh",
";411" : "/usr/bin/411.sh",
";611" : "/usr/bin/611.sh",
";711" : "/usr/bin/711.sh",
";setTime" : "/usr/bin/dateTime.sh",
";st" : "/usr/bin/dateTime.sh",
";-ds" : "/usr/bin/ds.sh",
";toggleLight" : "/usr/bin/light.sh",
";fc-cache" : "/usr/bin/fc-cache.sh",
";htmlViewer" : "/usr/local/bin/htmlViewer.sh",
";installHtml" : "/usr/local/bin/installHtmlViewer.sh",
";merchant" : "/usr/local/bin/merchant.sh",
";updateCamp" : "/usr/bin/updateCamp.sh"

```

klawiatury, czytnika, a następnie podłączenia do powerbanku kabla USB prowadzącego do huba USB, a pozostałe komponenty same się włączą. Połączenia zostały zaprezentowane na **fotografii 12**.

Ten projekt nie jest też trudny od strony programowej. Naturalnie trzeba mieć odblokowanego Kindle i trzeba na nim zainstalować SSH do pracy przez Wi-Fi. Zazwyczaj sprowadza się to do wprowadzenia zmiany w pliku /mnt/us/usbnet/etc/config, w którym opcję USE\_FIWI należy ustawić na true. Jeśli nie chcemy korzystać z USB over Ethernet, a w tym przypadku USB nie będzie potrzebne, ustawiamy też opcję USE\_Wi-Fi\_SSHD\_ONLY na true. Aktualny adres IP tak zmodyfikowanego Kindle możemy szybko poznać korzystając ze skrótu „;711”, który można wpisać w pole wyszukiwania w czytniku. Listę skrótów dostępnych na Kindle Paperwhite prezentujemy w **listingu 2**.

Warto też zainstalować oprogramowanie Kindle Unified Application Launcher (KUAL) oraz narzędzia Extend i terminal KTerm. Zostało to opisane pod adresem: <http://bit.ly/2X48PLD>.

### Dodatkowe ustawienia

W trakcie pracy z Kindle warto wyłączyć wygaszacz ekranu. Można to zrobić wpisując „~ds” w pasku wyszukiwania. W trakcie uruchamiania terminala można spróbować wymusić ukrycie klawiatury ekranowej (parametr „-k0”), ale nie zawsze działa to poprawnie. Zamiast tego można po prostu – po uruchomieniu Kindle, dotknąć ekranu dwoma palcami i wybrać opcję „ukryj klawiaturę”. Dotknięcie dwoma palcami pozwoli też przełączać inne rzeczy lub wyłączyć sam terminal.

Pewnym ograniczeniem jest możliwość przełączania okien. Użycie skrótu „CTRL+A” spowoduje problem, bo o ile Raspberry Pi przelazę się do innej sesji, terminal Kindle tego nie zrobi i przez to przestanie pokazywać to, co się aktualnie wpisuje. Dlatego ten tryb pracy wymusza stosowanie pojedynczej konsoli, a jeśli zachodzi potrzeba

REKLAMA

Specjalistyczne szkolenia dla elektroników i automatyków



**TECHDAYS**

techdays@techdays.pl  
TECHDAYS.PL

CERTYFIKOWANY PARTNER SZKOLENIOWY  
life.augmented



Fotografia 13. Kindle z okienkami, w roli wyświetlacza od laptopa

przełączania się pomiędzy programami, warto stosować komendę „CTRL+Z” i komendę „fg %X”.

Problematyczny może być rozmiar terminala, szczególnie jeśli zmieni się rozmiar czcionki. Do dostosowania rozmiaru warto użyć komendy „stty columns xx”, gdzie xx to liczba, która najlepiej odpowiada używanemu ekranowi przy aktualnym rozmiarze czcionki. Liczbę tę należy dobrać metodą prób i błędów.

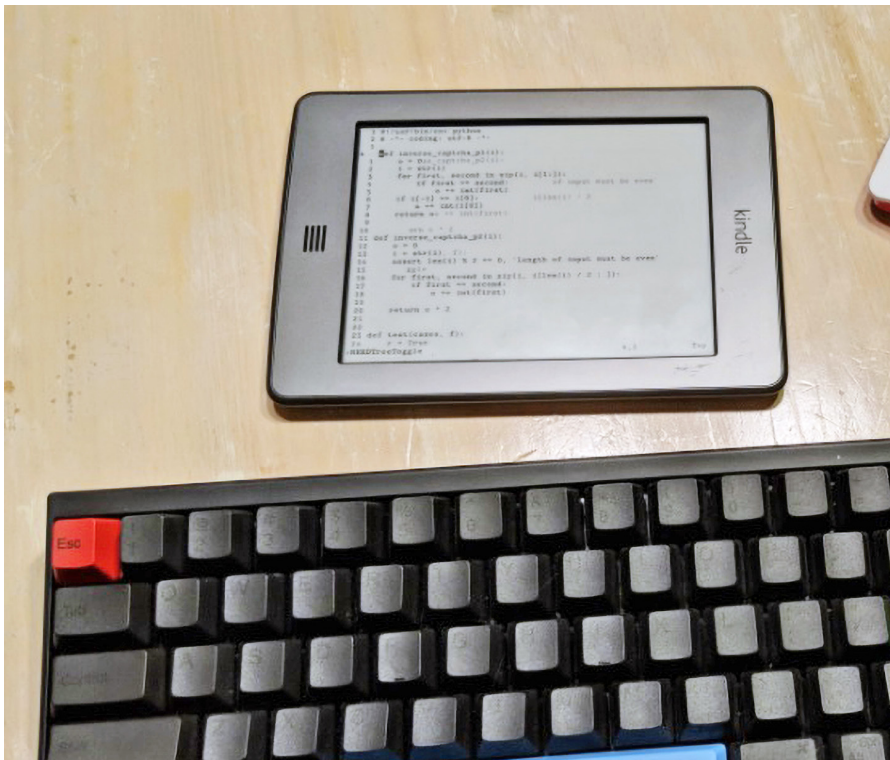
W końcu można też wyłączyć sprawdzanie połączenia internetowego. W tym celu należy utworzyć pusty plik /mnt/us/Wi-Fi\_NO\_NET\_PROBE i zrestartować usługę Wi-Fid. Można to zrobić za pomocą następujących komend:

```
touch /mnt/us/Wi-Fi_NO_NET_PROBE
restart Wi-Fid
```

### Kindle z okienkami

Oba opisane projekty bazują na trybie tekstowym. Ogranicza on mocno zakres zadań, jakie można wykonywać na takim komputerze, ale niektórzy ludzie chwalą sobie brak okazji do rozpraszania uwagi, jakie wprowadza środowisko graficzne. Niemniej, można czytnik EPD wykorzystać także w środowisku graficznym, w Raspbianie.

W tym celu należy zainstalować na odbezpieczonym czytniku program Kindle VNC viewer. Można go pobrać z adresu: <http://bit.ly/2YKZ858>. Jest to narzędzie, które pozwala z poziomu Kindle podłączyć się do serwera VNC. W wersji obecnie dostępnej nie pozwala na zdalne sterowanie urządzeniem, ale nie ma to dla nas znaczenia, gdyż sterować będziemy za pomocą klawiatury i myszki.



Fotografia 14. KindleBerry można też zbudować z użyciem Raspberry Pi Zero

Na samym Raspberry Pi zalecany serwerem jest Real VNC. Żeby go zainstalować można skorzystać z gotowych pakietów za pomocą poleceń:

```
sudo apt-get update
sudo apt-get install realvnc-vnc-server realvnc-vnc-viewer
```

Pomijając parametr realvnc-vnc-viewer nie zainstalujemy klienta VNC.

Następnie by włączyć serwer można wpisać komendę sudo raspi-config i przejść do opcji Interfacing Options, a wtedy wybrać VNC → Yes. Alternatywnie można skorzystać ze środowiska graficznego, w którym wchodzimy do Menu → Preferences → Raspberry Pi Configuration → Interfaces i upewniamy się, że VNC jest włączone.

Gdy wszystko jest skonfigurowane, na Kindle należy uruchomić sieć przez USB lub Wi-Fi, w zależności od wybranej metody połączenia. W wersji z USB można skorzystać ze skrótów wpisywanych w oknie wyszukiwania. Wpisujemy kolejno „;debugOn”, „;usbNetwork” (lub „;~usbNetwork” – zależnie od wersji czytnika) oraz „;debugOff” (bez cudzysłowów). Następnie przez ssh lub telnet łączymy się z poziomu Raspberry Pi z Kindlem. np. tak:

```
telnet 192.168.2.2
/etc/init.d/netwatchd stop
/etc/init.d/powerd stop
/mnt/us/kindlevncviewer/
kvncviewer.sh 192.168.2.1:1 &
```

Pierwsze z poleceń spowoduje zalogowanie się na Kindle z poziomu Raspberry Pi. Drugie zatrzymuje działanie usługi monitorującej sieć. Trzecie zatrzymuje działanie

usługi zarządzającej poborem mocy. Czwarte powoduje włączenie klienta VNC, przy czym podany adres IP należy zastąpić adresem urządzenia, z którym ma się komunikować Kindle. W przypadku połączenia przez USB, będzie to właśnie 192.168.2.1.

Aby zakończyć pracę należy z poziomu Raspberry Pi nacisnąć kombinację „CTRL+C” i wpisać:

```
Killall kindlevncviewer
```

Można też z powrotem włączyć usługi netwatchd i powerd.

### Podsumowanie

Opisany projekt to świetna okazja do wyjęcia starego czytnika z szuflady i przygotowania go pod kątem zbliżającego się lata. Wdrożenie projektu jest szybkie, a największą trudność może stanowić ewentualny problem ze zdjęciem zabezpieczeń z czytnika, szczególnie jeśli ten jest nowym modelem. Projekt ten nie wymaga też znajomości elektroniki – wystarczy podpiąć kilka kabli i dokonać konfiguracji programowej. A ci, którzy niekoniecznie chcą nosić ze sobą kilka oddzielnych elementów i zamiast tego woleliby coś w postaci laptopa, gdzie klawiatura, myszka i źródło zasilania są w jednej obudowie, mogą zamiast Raspberry Pi użyć laptopa (fotografia 13). Opisane w artykule metody konfiguracji podziałają także ze zwykłym komputerem, a nie tylko z Raspberry Pi. Jeśli jednak chcemy w ogóle zminimalizować rozmiary urządzenia, można sięgnąć po Raspberry Pi Zero, czego przykładem jest fotografia 14.

Marcin Karbowniczek, EP