OTGS
Reducing Energy Consumption of USB-connected Low-cost Sensors on Smartphones

in 't Veen, Ivar Jacobus Gerardus; Majid, Amjad; Pawelczak, Przemek

DOI
10.1109/DySPAN.2017.7920774

Publication date
2017

Document Version
Final published version

Published in
2017 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)

Citation (APA)
in 't Veen, I. J. G., Majid, A., & Pawelczak, P. (2017). OTGS: Reducing Energy Consumption of USB-connected Low-cost Sensors on Smartphones. In *2017 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)* (pp. 1-2). IEEE. https://doi.org/10.1109/DySPAN.2017.7920774

OTGS: Reducing Energy Consumption of USB-connected Low-cost Sensors on Smartphones

Ivar in 't Veen, Amjad Yousef Majid, and Przemysław Pawełczak
TU Delft, Mekelweg 4, 2628 CD Delft, The Netherlads; email: {a.y.majid, p.pawelczak}tudelft.nl

***Abstract*—This poster presents a hardware/software solution, called On-The-Go Switch (OTGS), enabling a smartphone to control the connection state of a USB-attached device. Through an example, we show how OTGS can reduce the energy consumption of portable spectrum sensing platform utilizing low-cost Software Defined Radio. We show that OTGS can reduce the energy consumption by 47%, compared to a baseline, which would be difficult with a software alone, e.g. kernel modification.**

I. Introduction

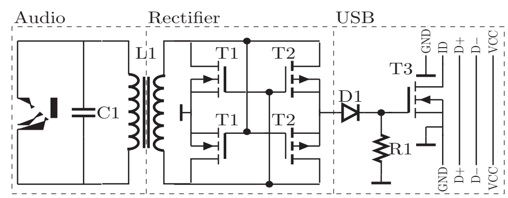
Many sensing applications use a combination of a smartphone and a USB-connected device, e.g. crowd-sourced radio spectrum sensing [1]. Unfortunately USB devices deplete the battery of the phone. According to the USB specification the USB device has to manage its own power consumption by going to a *suspended* state if possible or requested [2, Sec.7,9]. However, many USB devices do not support such state [3]. Even if supported, a *suspended* USB device is still allowed to consume energy [2, Sec.7]. Therefore, connecting a USB device to a smartphone, e.g. via a USB On-The-Go cable [4], can decrease a smartphone's battery life—both because USB device keeps the smartphone awake and because of USB device own power consumption.

To alleviate the problem we introduce On-The-Go Switch (OTGS): a hardware/software platform that enables a smartphone to control via the audio port the *logical* connection state of a *physically* connected USB device. This approach limits both the duration at which the smartphone needs to act as USB master and the duration at which the USB device is consuming energy. Our proposed OTGS could work on any smartphone with USB OTG capabilities and a headphones connector without relying on custom kernels, rooted/jailbraked firmwares or even the operating system. As a case study we show how OTGS can reduce the overall power consumption of a radio spectrum sensing platform.

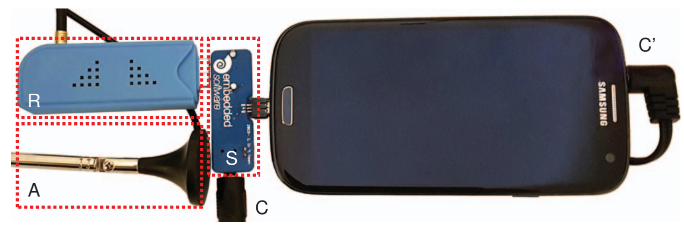
II. OTG Switch

Inspired by the design of [5] we propose OTG switch which replaces the OTG cable connecting a smartphone and USB device. Following the USB standard [2] OTG cables use a fifth connection (called `ID` connection) within the OTG connector to signal their presence to a USB master. The OTGS simulates the physical connection and disconnection of the OTG cable by switching this `ID` connection between a pull-down and its

Supported by the Dutch Technology Foundation STW under contract 12491.



(a) OTG switch, components: C1 = 1μF, **L1**: LPR6235, **T1**: ZSM61P03F, **T2**: ZSM61N03F, **D1**: DFLS120L-7, **T3**: FDV301N, R1 = 1MΩ (b) OTG switch



(c) PoSSP hardware components: **R**: RTL-SDR dongle, **S**: OTG Switch, **A**: Antenna. **C** and **C'** are connected with an audio cable (not fully shown).

Fig. 1. PoSSP platform: (a) OTGS schematics; (b) OTGS; and (c) system overview.

default pull-up state, see Fig. 1(a). The `ID`-connection, while in pull-down state, will enable the phones USB master mode to provide power and communication to the attached USB device. On a hardware level the switching is done by a FET transistor which is controlled from the phone's headphone output. To enable the OTGS the attached smartphone plays a audio signal through its headphone connector, this waveform is rectified on the OTGS to provide a suitable signal to switch the FET. The audio signal outputted by the phone is a 15 kHz sinusoid. The OTGS needs around one second to stabilize after changing state. The combination of the large drain resistor and parasitic input capacitance mitigates the need for an extra storage capacitor.

III. Case Study: Energy consumption of PoSSP

In this case study we introduce our Portable Spectrum Sensing Platform (PoSSP), analyze its power consumption and show how this can be greatly reduced by the OTGS.

Our PoSSP platform setup consists of a *Samsung Galaxy S3* running *Android 4.3.1*, our OTGS, an RTL-SDR dongle [6], and antenna, see Fig. 1(c). Source code for the PoSSP, i.e., sensing software, PCB design and experiment results, are available upon request or via [7].

978-1-5090-2830-6/17/$31.00 ©2017 IEEE

TABLE I
POSSP POWER CONSUMPTION PER COMPONENT

	Wave†	Audio‡	USB	Dongle	Power (mW)
$P_{\text{p,on}}$	—	—	—	—	789
$+P_{\text{wave}}$	yes	headphones	—	—	870
$+P_{\text{otgs}}$	yes	OTGS	OTGS	—	1015
$+P_{\text{usb}}$	yes	OTGS	OTGS	connected	1557§

† Is waveform generated by the smartphone
‡ Type of connection to the smartphone's audio port
§ Average power consumption of four tested RTL-SDR dongles

TABLE II
POWER CONSUMPTION OF VARIOUS COMMERCIALLY AVAILABLE RTL-SDR DONGLES

Type	Tuner	Chipset	No.	P_{usb} (mW)	$\sigma_{P_{\text{usb}}}$
USB HDTV Stick	E4000	RTL2832U	1	454	—
ezcap	FC0013	RTL2832U	1	434	—
NooElec	R820T2	RTL2832U	4	662	18
DVB-T+DAB+FM	R820T	RTL2832	4	615	12

A. Energy Consumption of PoSSP Components

The power consumption of the PoSSP components is measured using a Monsoon power monitor [9] set to a voltage of $4.19\,\text{V}$. The device under test is an Android smartphone in *airplane mode* with the screen enabled. This phone is running a custom app, generating a $15\,\text{kHz}$ audio signal.

Table I shows power measurements for different combinations of PoSSP components. These results show that the audio generator consumes $81\,\text{mW}$ ($P_{\text{wave}} - P_{\text{p,on}}$) while the OTGS adds another $145\,\text{mW}$ ($P_{\text{otgs}} - P_{\text{wave}}$). This increase in energy consumption while the switch is enabled is lower than the amount of energy otherwise wasted by keeping the radio dongle continuously enabled. Furthermore, measurements show that not all RTL-SDR dongles have the same energy consumption, see Table II.

B. Demonstration of Energy Consumption Reduction with PoSSP

In this experiment we use NEAT [8] which is a high resolution, coin sized, and self powered power meter. NEAT connects to the vibration module of the smartphone to synchronize the measured power consumption trace of the smartphone with an activity trace of an application running on the same smartphone. Then, we used an freely available tool, `rtl_power` [10], to instruct the RTL-SDR dongle to sense the radio spectrum.

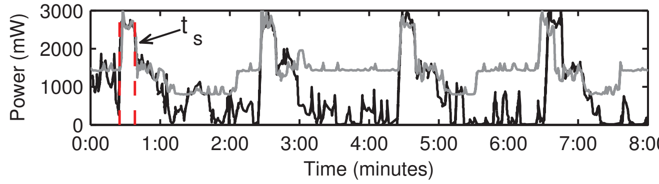


Fig. 2. Example NEAT [8] power traces of PoSSP, top (light) line shows consumption without, bottom (dark) line shown with OTGS. The sensing period for both traces is marked with t_s, see Section III-B for details.

Fig. 2 shows the power traces of PoSSP while doing a periodic sensing operations every two minutes with, and without OTGS. A sensing operation takes $\approx 20\,\text{s}$, and it consumes $\approx 34\,\text{J}$. OTGS, on one hand, adds two extra joules to the cost of a sensing operation. On the other hand, it reduces the energy consumption of a non-sensing period from $130\,\text{J}$ to $51\,\text{J}$. Moreover, OTGS will save more energy if the interval between two sensing operations is increased.

IV. CONCLUSION

This paper proves how a simple On-The-Go switch (OTGS) can reduce the energy consumption of a USB host, i.e a smartphone, without affecting the performance of a system. Furthermore, we introduce a smartphone-based Portable Spectrum Sensing Platform, which utilizes OTGS, aimed at prolonging smartphone battery life.

REFERENCES

[1] A. Nika, Z. Zhang, X. Zhou, B. Y. Zhao, and H. Zheng, "Towards commoditized real-time spectrum monitoring," in *Proc. ACM HotWireless '14*.
[2] "USB specification version 2.0." [Online]. Available: http://www.usb.org/developers/docs/usb20_docs
[3] "Linux kernel archives." [Online]. Available: https://www.kernel.org/doc/Documentation/usb/power-management.txt
[4] "USB OTG." [Online]. Available: http://www.usb.org/developers/onthego/
[5] Y.-S. Kuo, S. Verma, T. Schmid, and P. Dutta, "Hijacking power and bandwidth from the mobile phone's audio interface," in *Proc. ACM DEV '10*.
[6] "RTL SDR." [Online]. Available: http://sdr.osmocom.org/
[7] (2016) Experiments source code. [Online]. Available: http://www.es.ewi.tudelft.nl/papers/2017-Veen-DySPAN_source_code.zip
[8] N. Brouwers, M. Zuniga, and K. Langendoen, "NEAT: A novel energy analysis toolkit for free-roaming smartphones," in *Proc. ACM SenSys '14*.
[9] "Monsoon power monitor." [Online]. Available: https://www.msoon.com/LabEquipment/PowerMonitor/
[10] "RTL-power." [Online]. Available: http://kmkeen.com/rtl-power/