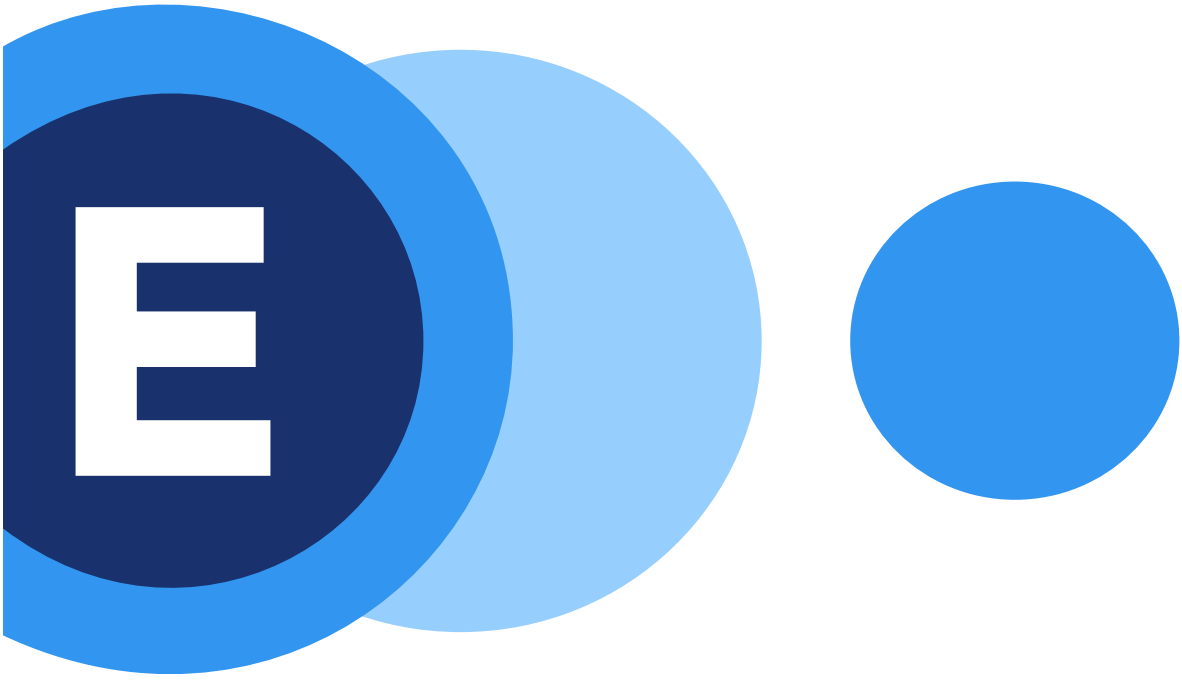
SallandElectronics

THE HACKER'S GUIDE TO THE WOTS BADGE

SPECIFICATION



Document information

Title	The hacker's guide to the WoTS Badge
Subject	Specification
Current version	0.2
Status	preliminary release
Author	Joost van Velzen
Project id	N/A
Customer	WoTS 2016
Filename	The hackers guide to the WoTS Badge.doc
Document Id	WOTS2016-BADGE1
Template	SE_EN_basic.dot
Last updated	04-Oct-16 08:26:00 by Joost van Velzen.
Number of pages	12, including front page and this information page.

Revision history

Date	Version	Status	Author	Description	Distribution
06-06-2016	0.1	Preliminary	JvV	First release	External
04-10-2016	0.2	Final	JvV	Added WotsBadgeUWP description	External

CONTENTS

1 INTRODUCTION

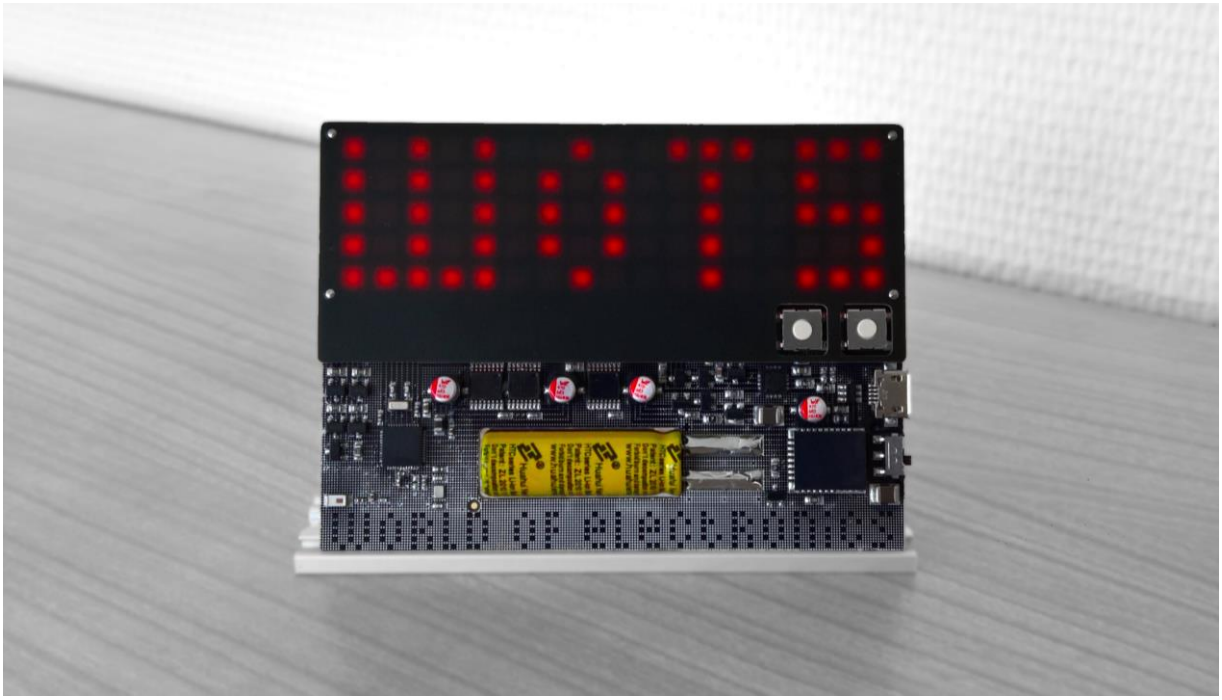
Dear hacker, let me introduce myself: I am Joost van Velzen. I run the innovation department at SallandElectronics, which means my job is all about creativity and inventing new technology. When we designed the WoTS Badge we decided to put all the best things of all the sponsors in it: The badge has a special battery from Elincom that looks like a capacitor, the very latest Bluetooth Smart system-on-chip from Nordic semiconductor and a very efficient power chip from Würth. The WoTS Badge was inspired by the 8Rlicht and got a lot of the software features from that smart taillight. The best feature it got is its hackability: You can fully control the badge either via USB or wireless: that ought to make your lives easier…

So, I wish you happy hacking and I hope that the WoTS Badge will be part of your smart home solution.

Good luck!

Joost van Velzen

2 OVERVIEW



The image shows the final version of the WoTS-Badge.

What you need to know is that there is a micro-usb port on the right, just above the on-off switch. The switch only disconnects the battery, so when the badge is tethered to a PC via USB it will work even with the switch set to OFF.

There are two buttons. Button 1 is the left one in the picture, button 2 is on the right. Button 1 start the wireless connection and button 2 cycles through display modes. There are three display modes: one is showing the design you uploaded (image, text or animation), one shows the current time and the last one shows the battery status.

The matrix size is 17x5, you can send images, animations and text and even use the alarm clock function to have the device time animations itself.

3 BASICS

First things first. In the following paragraphs you find the things you need to connect to the badge as well as some background to help you understand the rest of the document.

3.1 USB connection

Set the serialport to 460800, 8N1, no flow control.

In case you are a .Net fan, forget about Microsoft's SerialPort implementation and use the FTDI drivers directly instead. To enable communication via USB you need to send the following command followed by either CR+LF or only LF:

```
CONNECT()
```

This returns OK or NO followed by CR+LF. The USB cannot function when the wireless is active, so in that case you get NO.

```
OK
```

```
NO
```

Over the USB connection the badge will respond to every other valid command with OK.

The USB connection uses flow control on large commands. If you send commands of more than 256 characters, then the badge will send an XOFF so it can process the buffer and will send XON when it is done and you can continue sending. This is by no means a full XON/XOFF flow control scheme and if you only use live mode you don't need to worry about it.

3.2 Wireless connection

To start the wireless connection, press button 1. The badge will show a four digit code on the display for about three minutes or until you connect. The code is the last part of the advertised name of the device. E.g. if the badge displays 1234 then the name is WoTS-1234. Our own app uses this to connect to the correct device and you can do the same.

To create a serialport over the wireless connection we use the UART service from Nordic (UUID=6E400001-B5A3-F393-E0A9-E50E24DCCA9E). You can write messages of at most 20 bytes to the TX characteristic (UUID=6E400002-B5A3-F393-E0A9-E50E24DCCA9E). If your message is larger than 20 bytes, then you need to split it into multiple messages. The badge does not send anything back.

A command must always start in a new message, you do not need to send CR+LF.

3.3 Command formats

Commands are case sensitive and do not support additional spacing, tabs, line breaks etc. The only place where a space char is accepted in a command is when it is part of a text parameter.

3.3.1 Duotrigesimal / Base32

We use a specific Base32 alphabet to encode columns of pixels in a single char. We have chosen a set that extends hexadecimal, so you can play around with the bottom 4 lines of the display until you finally master the art of simply visualizing the image in duotrigesimal format in your mind. If you haven't mastered the art of thinking in hex yet, but you can count to seven, then you can play with the bottom 3 lines using the numbers 0 to 7.

Thus it takes 17 chars to define an image that will fill the display, and no, there are no halftones.

The 32 characters are:

0123456789ABCDEFGHJKLMNPQRSTUVWX

X = 31, note that I and O are skipped to avoid confusion with 1 and 0.

Note that only uppercase is supported. Sending ‘a’ instead of ‘A’ will NOT work.

3.3.2 We start on Monday

In time related functions Monday always has index zero. Sunday is 6.

3.3.3 Time formatting

The time is always expressed as a 5 character sequence consisting of two digits for the hours, a colon and two digits for the minutes: “23:59”.

4 MODES

The WoTS Badge can operate in several modes which can be controlled by button 2 or by sending SETSTATE via USB or Wireless.

```
SETSTATE(D)
```

The blue D in this case stands for Design. You can switch to Clock mode by sending ‘C’ instead and battery mode by sending a ‘B’.

4.1 Design mode

In this mode your design is shown on the badge. You can upload a design using the live, text, image or animation commands.

4.1.1 Live

This commands loads an image directly into the framebuffer. The image is shown on the display and not stored in the flash memory. Powercycling the device will make the last image, text or animation reappear.

```
LIVE(0123456789ABCDEFGH)
```

In the example above 0123456789ABCDEFGH is a string of at most 17 duotrigesimal characters that together form exactly one frame. The frame will remain on the display until the next frame is sent. So yes, you can do streaming video ☺.

4.1.2 Upload new text

Send a new text and store it in flash (frequent use of this function wears out the flash).

```
TEXT(look Ma, I created a scrolling text!)
```

The text you send may have the following characters: a-z, A-Z, 0-9 and the following symbols: +-.()[]/\=|@^%&±#!?;:$_{}<>.

4.1.3 Upload new image

Send a new image and store it in flash (frequent use of this function wears out the flash).

You can also send an image that is wider than 17 columns, in this case the image will start scrolling.

```
IMAGE(00000000000000000123456789ABCDEFGHJKLMNPQRSTUVWX00000000000000000)
```

The example above shows the entire duotrigesimal alphabet scrolling by over and over again.

4.1.4 Upload new animation

A very cool feature is that you can also upload animations. Animations are played at 10fps. The data of an animation consists of 17 duotrigesimal characters per frame and each frame is separated by a colon. If you use only 10 characters in a frame the badge will append the frame with zeroes.

```
ANIMATION(1248G8421248G8421:G1248421G1248421G:8G12421G8G12421G8:48G121G848G121G84:248G1G84248G1G842)
```

The above animation is a saw line with 45 degree angles that scrolls down twice per second.

Animations wear out the flash, same as texts and images.

4.2 Clock mode

The badge can function as a digital clock with visible notifications.

The clock requires some configuration, the most obvious of which is the time.

4.2.1 Set the time

Set time and day of the week. Note that hours and minutes always require two digits. The last parameter is the day of the week, which is Monday (0).

```
SETTIME(09:01,0)
```

The clock has a bit of drift, which the app automatically corrects whenever it is connected.

4.2.2 Set energy friendly clock mode

The clock has a power saving feature and will always go off after one minute. Press button 2 again to make screen go on. You can choose to force the clock on during a specific period of the day on specific days with the CLOCKON command:

```
CLOCKON(09:00,17:00,1111100)
```

The parameters are ‘start time’, ‘end time’, ‘weekday mask’.

Sending the above command will force the display on between 9:00 and 17:00 on Monday till Friday. The first time value must always be earlier than the second time value. The second time may actually be 24:00 to be able to specify that the clock must be on the entire day. Note that on Saturday and Sunday the clock is off as the last two bits of the weekday mask are zero.

To set the clock to always on you can send:

```
CLOCKON(00:00,24:00,1111111)
```

To set the clock to always off you can send:

```
CLOCKON(00:00,24:00,0000000)
```

This setting is obviously the best if you use the badge in clock mode running from the battery.

4.2.3 Setting notifications

You can configure up to 10 timers. You cannot make changes to timers you have already set, but you can clear all timers using CLEARTIMERS and send them again.

```
CLEARTIMERS()
```

To set a timer you use one of the “TIMED” commands: TIMEDTEXT will show a (scrolling) text as a notification and TIMEDIMAGE and TIMEDANIMATION will show an image or animation respectively. The format of the content is the same as when you upload a design.

```
TIMEDTEXT(17:00,0000100,WEEKEND!!!!!!)
```

The above command will cause the text “weekend!!!!!!” to appear on Friday at 17:00.

```
TIMEDTEXT(12:30,1111100,Lunch!)
```

The above command will cause the text “Lunch!” to appear on Monday to Friday at 12:30.

```
TIMEDIMAGE(09:00,1111100,XXXXXXXXXXXXXXXX)
```

The above command will cause all leds to go on at 9:00 on working days.

```
TIMEDANIMATION(17:05,0000100,XXXXXXXXXXXXXXXX:0)
```

The above command causes the screen to strobe when you're still at work on a Friday at five minutes past 17:00.

Each of the TIMED commands consists of a time, a weekday mask and appropriate content.

4.3 Battery mode

In battery mode the display shows a graphical presentation of the battery capacity.

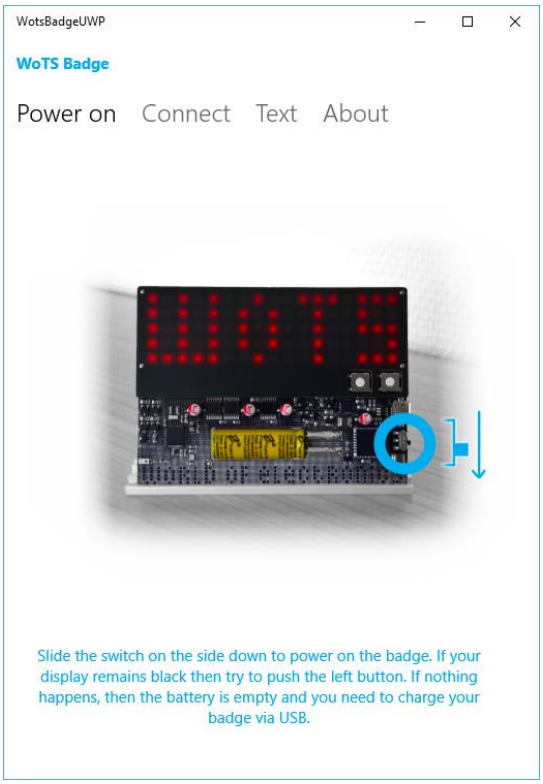
- When the battery is full all but two pixels are on.
- When the battery is empty, all you see is the outline of a battery.
- When the battery isn't empty or full then the filled area (leds on) represents the remaining capacity.
- When the battery is charging then the status will constantly switch between the current capacity and the next step, causing a blinking effect.

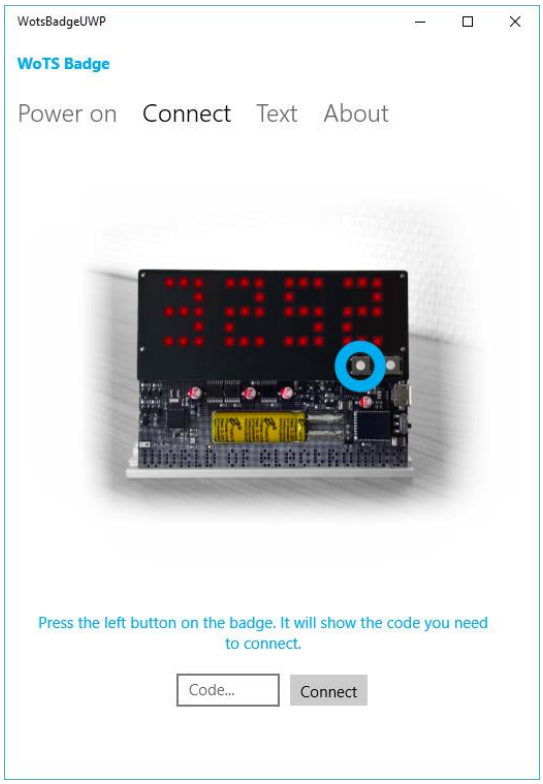
5 SAMPLE CODE

5.1 Uart code

Check out the code for the WaveNed sensornetwork on the hackathon github.

5.2 Windows / Windows phone app (UWP)





The Windows phone app was not finished as it was not possible to get it completely tested and debugged and then in the Windows Store in time for the WoTS. When it would be in the Windows Store we would have another problem as this is not the recommended app to use on a windows machine that can run desktop apps. You should use the EuroCircuits app instead. Also the API's used in this app are only available on the very latest version of Windows 10: The anniversary update. Thus we have a lot of good reasons, to not put this in the Windows Store. We will probably not cause a major inconvience to users as it is extremely likely that they will have access to a Windows laptop considering they have a Windows phone.

We are sharing the code of this app with you, because it is good enough to build upon for the Hackathon. Once a device is paired the code works relatively well so this is a nice code base to control a WoTS-Badge wirelessly from your Windows 10 laptop, phone or tablet.

Install Visual Studio Community edition on a Windows 10 laptop with anniversary update. Go to new project and select UWP to install all the UWP stuff, then open the project.

You can find the project WotsBadgeUWP on the hackathon github.

P.s. This specification is subject to change without prior notice (you'll probably notice when you start testing).